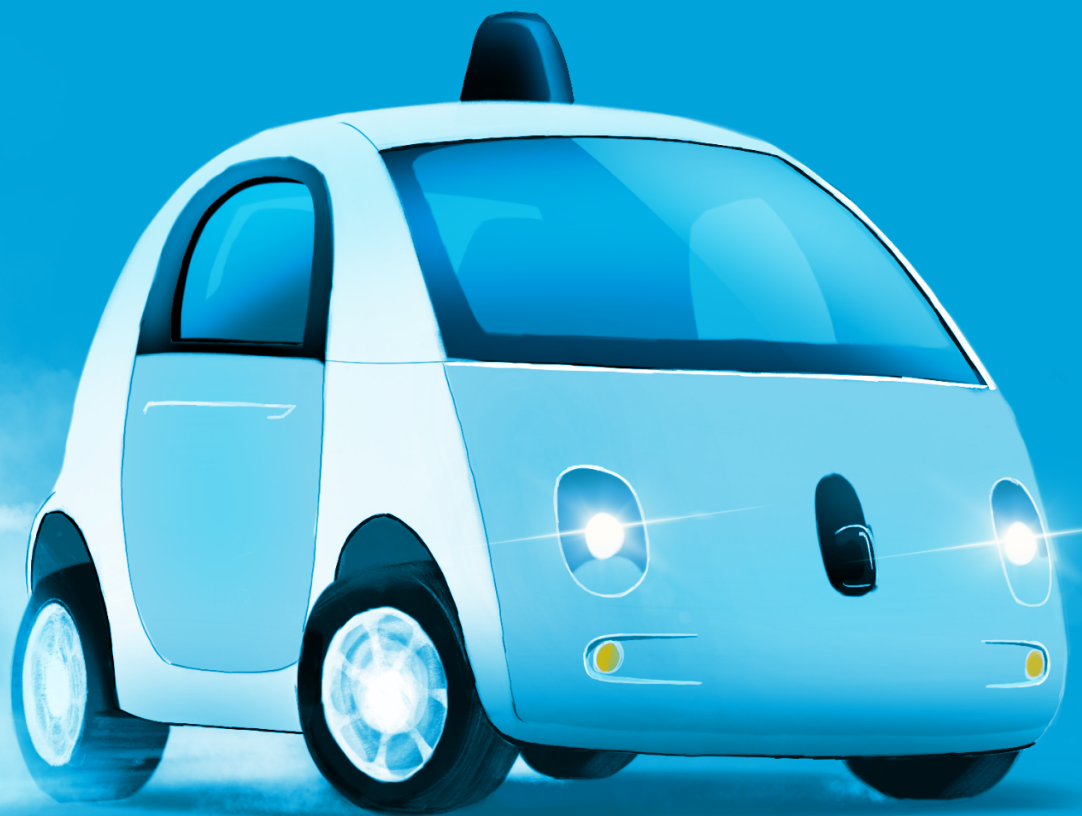


# Sensor Fusion of Computer Vision and Stock Sensors for Vehicle Dynamic Sideslip Estimation

Providing reliable state estimates with on-board sensors to support autonomous vehicle drift control

C.B. Kuyt

Master of Science Thesis





# **Sensor Fusion of Computer Vision and Stock Sensors for Vehicle Dynamic Sideslip Estimation**

**Providing reliable state estimates with on-board sensors  
to support autonomous vehicle drift control**

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering  
at Delft University of Technology

C.B. Kuyt

June 30, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE)  
Delft University of Technology



**POLITECNICO**  
MILANO 1863

DIPARTIMENTO DI ELETTRONICA  
INFORMAZIONE E BIOINGEGNERIA

The work in this thesis was supported by the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico Milano. Their cooperation is hereby gratefully acknowledged.



Copyright © Delft Center for Systems and Control (DCSC)  
All rights reserved.



---

# Abstract

Vehicle control automation is the next step in the development of vehicle active safety systems. One of the challenges in this development is to extend the operational range of the control systems beyond stable behaviour of the vehicle to handle extreme emergency situations. Drift motion is a vehicle state outside the stable regime. Automated drift control system design for autonomous vehicles is a recently investigated topic in the field of vehicle engineering. Stability analyses revealed the existence of unstable drift equilibria, for which control methodologies have been developed by multiple research groups around the world.

A lack of investigation was noticed in the literature into feasible sideslip estimation techniques for implementation in a production setting, using solely on-board stock sensors. The aim of this thesis was to develop a state estimation technique for reliable sideslip estimation to support drift control using only on-board stock sensors: encoders (ENC), an inertial measurement unit (IMU) and a camera. A computer vision algorithm (CVA) has been developed that is able to track features on the ground surface with the camera. A small-scale vehicle has been built to serve as an experimental testbed for state estimation. The vehicle motion was tracked with high precision with an external camera which served as a benchmark for the state estimator.

The raw signals of the sensors provided sideslip estimates with accuracies of 70-80% (IMU) and 80-90% (CVA). However, the estimates based on the IMU are subject to drift while the CVA shows significant outliers. To improve the estimates, two sensor fusion approaches were developed: one where the CVA measurements were processed individually (individual approach) and another where they were combined (combined approach). Fusing the measurements improved the estimates significantly (above 90%). The combined method proved to be superior to the individual method. The combined method was tested against validation experiments and obtained the same accuracy (90-100%), minimising the probability of over-fitting the algorithm parameters.



---

# Table of Contents

<b>Preface</b>	<b>xv</b>
<b>Acknowledgements</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1-1 Automated Driving . . . . .	1
1-1-1 Origin in Vehicle Safety . . . . .	1
1-1-2 State of the Art and Future Challenges . . . . .	2
1-1-3 Vehicle Control at the Handling Limits . . . . .	3
1-2 Tire and Vehicle Dynamics . . . . .	4
1-2-1 Tire-Road Interaction . . . . .	4
1-2-2 Cornering Motion . . . . .	5
1-2-3 Stability Analysis . . . . .	7
1-3 Drift Motion and Control . . . . .	8
1-3-1 Drift Motion Analysis . . . . .	8
1-3-2 Race Driver Behaviour Studies . . . . .	9
1-3-3 Automated Drift Control Development . . . . .	10
1-4 Vehicle Dynamic State Estimation . . . . .	12
1-4-1 Drift Control Literature Evaluation . . . . .	12
1-4-2 Established Methodologies . . . . .	13
1-4-3 State Estimation in Intelligent Vehicles . . . . .	14
1-5 Conclusion . . . . .	15
1-5-1 Problem Formulation . . . . .	15
1-5-2 Thesis Objectives . . . . .	16
1-5-3 Project Outline . . . . .	16

<b>2</b>	<b>Sideslip Estimation Methodology</b>	<b>17</b>
2-1	Introduction . . . . .	17
2-2	On-Board Sensor Suite . . . . .	18
2-2-1	Rotary Encoders . . . . .	18
2-2-2	Inertial Measurement Unit . . . . .	19
2-2-3	Camera with Computer Vision . . . . .	20
2-3	Dynamic State Observer . . . . .	21
2-3-1	Process and Measurement Models . . . . .	21
2-3-2	Kalman Filter . . . . .	21
2-3-3	Sensor Fusion Design Proposal . . . . .	22
2-4	Experimental Platform . . . . .	23
2-4-1	Scaled Vehicle Architecture . . . . .	23
2-4-2	Vehicle Parameter Estimation . . . . .	24
2-4-3	Motion Capture System . . . . .	25
2-5	Conclusion . . . . .	25
<b>3</b>	<b>Motion Capture System</b>	<b>27</b>
3-1	Introduction . . . . .	27
3-2	External Camera Data . . . . .	28
3-2-1	Marker Tracking . . . . .	28
3-2-2	Position Data . . . . .	28
3-3	Signal Processing . . . . .	30
3-3-1	Time Derivatives . . . . .	30
3-3-2	State Transformations . . . . .	31
3-4	Datasets . . . . .	34
3-4-1	Benchmark . . . . .	34
3-4-2	Validation . . . . .	34
3-5	Conclusion . . . . .	35
<b>4</b>	<b>Computer Vision Algorithm</b>	<b>37</b>
4-1	Introduction . . . . .	37
4-2	On-Board Camera Calibration . . . . .	38
4-2-1	Distortion . . . . .	38
4-2-2	Homography . . . . .	39
4-3	Algorithm Stages . . . . .	41
4-3-1	Image Processing . . . . .	41
4-3-2	Optical Flow Estimation . . . . .	42
4-3-3	Egomotion Projection . . . . .	44
4-4	Parameter Tuning . . . . .	45
4-4-1	Default Parameters . . . . .	45
4-4-2	Sensitivity Analysis . . . . .	46
4-5	Conclusion . . . . .	46

<b>5</b>	<b>Sensor Fusion Algorithm</b>	<b>49</b>
5-1	Introduction . . . . .	49
5-2	Signal Conditioning . . . . .	50
5-2-1	Characteristics . . . . .	50
5-2-2	Processing . . . . .	50
5-3	Algorithm Design . . . . .	53
5-3-1	Model Discretisation and Linearisation . . . . .	53
5-3-2	Sensitivity Analysis . . . . .	55
5-4	Results . . . . .	56
5-4-1	Approach Comparison . . . . .	56
5-4-2	Validation . . . . .	56
5-4-3	Critical Evaluation . . . . .	57
5-5	Conclusion . . . . .	57
<b>6</b>	<b>Conclusion</b>	<b>59</b>
6-1	Summary . . . . .	59
6-2	Recommendations . . . . .	61
<b>A</b>	<b>Vehicle Hardware Specifications</b>	<b>63</b>
<b>B</b>	<b>Computer Vision Function Parameters</b>	<b>65</b>
<b>C</b>	<b>Sensor Fusion Results</b>	<b>69</b>
C-1	Sensitivity Analysis . . . . .	69
C-1-1	Individual Processing . . . . .	69
C-1-2	Combined Processing . . . . .	71
C-2	Additional Estimation Result: Rear Wheel Longitudinal Slip . . . . .	72
	<b>Bibliography</b>	<b>73</b>
	<b>Glossary</b>	<b>79</b>
	List of Acronyms . . . . .	79
	List of Symbols . . . . .	80
	<b>Index</b>	<b>81</b>



---

# List of Figures

1-1	<b>State of the Art Technologies and corresponding Challenges and Solutions.</b> Autonomous vehicle operation comprehends three stages. Perception (green) is currently achieved with extensive sensor suites but they should be simplified to be implemented on large-scale. Decision (blue) currently requires powerful computers because of the complex human actors that are involved in traffic. The execution stage (red) is how the system acts on the decisions, with mostly electronic actuators. Electrical energy storage is difficult to implement on a large scale because of the lack of infrastructure. Also, the bottleneck of advanced vehicle control is usually the vehicle model of which the identification is still difficult in a mass production setting. . . . .	2
1-2	<b>Obstacle avoidance at high velocity.</b> The green area represents the set of potential paths on which the vehicle would have small slip angles. Outside of this region, the steep steering angle input would lead to high amplitude sideslip, in which the vehicle becomes unstable. The dotted black line represents the path on which the vehicle would just barely avoid a collision. This situation thus requires the vehicle to be handled in the unstable regime. . . . .	3
1-3	<b>Tire Model Parameters: Definitions.</b> Subfigure a) shows the forces that are exerted on the road surface at the tire-road contact point, b) and c) show a side- and top-view to visualise the dynamic parameters used to describe the tire behaviour. . . . .	4
1-4	<b>Tire Model Parameters: Dynamic Relationships.</b> It can be clearly observed in the right plot that the lateral friction decreases with longitudinal slip. This is an important phenomenon in drifting, since it implies that the lateral motion of the vehicle can be indirectly influenced by a longitudinal input on the wheels. . . . .	5
1-5	<b>Steady-State Cornering Motion.</b> The vehicle corners around a centre of rotation $CR$ at a cornering radius $\rho$ . . . . .	6
1-6	<b>Friction Limit Circle.</b> The outer red circle describes the theoretical maximum forces the tire can transmit between the ground and the vehicle. In longitudinal direction the practical maximum (yellow line) is limited by the maximum drive force that can be effectively delivered by the powertrain (driving). In lateral direction (cornering) the vehicle geometry limits the possible acceleration as well. For braking however, the limit is equal to the tire maximum, since the wheels can be locked and the tires will slide over the surface inducing the maximum deceleration to the vehicle. In a drift the rear tires move outside of the circle. . . . .	6

1-7	<b>Steady-State Cornering Motion.</b> The grey area represents a corner with a constant radius to visualise the differences between over- and under-steer in terms of that variable. The yellow line visualises an increasing cornering radius and the red line a decreasing radius. This phenomenon is leveraged by professional rally drivers to take sharper corners. . . . .	7
1-8	<b>Steady-State Cornering Motion.</b> Subfigure (a) shows stable cornering motion while Subfigure (b) shows unstable cornering (drifting) where $\beta$ is large. Also, (b) shows a counter-steer configuration, where $\delta$ is opposite to $\omega_z$ . . . . .	8
1-9	<b>Empirical Schematic Description of the Trail-Braking Manoeuvre.</b> At point 1 the driver initiates the corner after having stepped on the brakes for an amount of time depending on the length of the straight before the corner. The brake is released slowly until point 4 (the corner apex). While the steering angle is applied as a sinusoid, representing 'sliding mode control' behaviour. The throttle is gently introduced in the corner exit. . . . .	9
1-10	<b>Drift Equilibria Examples.</b> Subfigure (a) shows two linear sets of equilibria, one for a positive, the other for negative sideslip angle. The trajectory indicated with the arrow in (a) at the right-top corner corresponds to the left-top corner in (b). A counter-steer drift configuration as visualised in Figure 1-8 (b) is defined as a state where the sign of $\delta$ is opposite to the sign of $\beta$ . This can be clearly be observed in Subfigure (a). . . . .	10
1-11	<b>Sideslip Estimation Methodology Categories</b> . . . . .	13
2-1	<b>Sideslip Estimation Process</b> (a) Shows a general schematic of how process states that can not be directly measured can be estimated by measuring inputs and outputs with sensors and a state observer. (b) Shows the process to be estimated: the $CG$ -states of a drifting vehicle on a global $X - Y$ -plane. . . . .	17
2-2	<b>Rotary Encoder Schematic.</b> (a) Shows the placement of the encoders on the front- and rear axles. (b) Shows a sideview and the directly measured quantity: the wheel angular rate $\omega_w$ . (c) Shows the front view of the encoder. The displayed disk is located on the centreline. . . . .	18
2-3	<b>Inertial Measurement Unit Schematic.</b> (a) Shows the placement of the sensor housing at the (approximate) $CG$ of the vehicle. (b) Shows the linear accelerations measured by the accelerometer triad, (c) shows the the angular velocity sensor triad outputs. . . . .	19
2-4	<b>On-Board Camera Schematic.</b> (a) Shows the placement of the camera on the vehicle in a side view. (b) Shows an abstract representation of the image. The CVA detected a feature in the previous image (yellow point location), and tracked it into the current image (green point location). The algorithm projects the image point locations $p$ in pixels to ground surface locations $q$ in meters. (c) Shows for a point $q_n$ a representation of the measured quantities. . . . .	20
2-5	<b>Vehicle Hard- and Software Architecture</b> . . . . .	23
2-6	<b>Geometric Vehicle Properties</b> . . . . .	24
2-7	<b>Motion Capture System.</b> Subfigure (a) shows a schematic side-view of the setup: the ceiling holds the MCS camera 3 m above the test surface. Subfigure (b) shows the top view MCS camera image. The red markers were used to calibrate the homography. The green markers on the vehicle were used to track its position and heading. The phone on the floor shows a stopwatch which is also visible on the on-board camera image. This was used to align the MCS with the on-board data. . . . .	25
3-1	<b>Drift Experiment: Drive Inputs.</b> Subfigure (a) shows the motor torque step input to 100%, (b) shows the steering step input to 15 deg. The wheels are turned completely before the vehicle starts moving because the response of the motor is slower than the servo. . . . .	27



3-2	<b>Tracking Markers</b> Subfigure (a) shows the numbering of the markers to distinguish them for the calculations in Subfigure (b), where the $CG$ location is deduced. . .	28
3-3	<b>Drift Experiment: Position Data.</b> The left figure shows the $X$ - $Y$ -trajectory which results from plotting $X_{CG}(k)$ against $Y_{CG}(k)$ for each time sample $k$ . The red squares are the locations of the homography calibration markers. The experiment started at the bottom left, and ended at the top middle. The steep change of the curve at the end is a good example of how a drift can result in a much narrower cornering radius, confirming the theory described in Section 1-2-3. The right figure shows the yaw (heading) $\psi_{CG}(k)$ . For visualisation purposes, the domain of the yaw angle is defined as $[0, 2\pi]$ . The vehicle came at an abrupt standstill at the end, which the curve demonstrates with the steep change at 3.5 s. The plots show the raw data, neither of the curves were smoothed. . . . .	29
3-4	<b>Drift Experiment: Absolute <math>X</math>- and <math>Y</math>-Velocity Derivation.</b> The global $(X, Y)$ -position of the $CG$ is shown as the dotted lines. The blue lines are the discrete derivatives as defined in (3-6), the solid black lines are the filtered derivatives (3-7). . . . .	30
3-5	<b>Drift Experiment: Derived <math>CG</math>-States.</b> The yaw $\psi_{CG}(k)$ of the vehicle is shown as the dotted line in the left plot. The blue line is the discrete derivative as defined in (3-6), the solid black line is the filtered derivative (3-7). The black line in the right plot shows the longitudinal velocity derived from (3-8). The blue line shows the lateral velocity from (3-9). . . . .	31
3-6	<b>State Transformation Scheme.</b> Subfigure (a) shows how the absolute velocities in terms global coordinates are transformed to the relative velocities using the yaw. The obtained relative velocities were used to obtain the axle velocities (b) and relative $CG$ accelerations (c). . . . .	31
3-7	<b>Drift Experiment: Axle Velocities.</b> The left plot shows the rear axle velocity $V_r$ . The right plot shows the front axle velocity in the steering direction $V_\delta$ from (3-12). . . . .	32
3-8	<b>Drift Experiment: Relative <math>CG</math>-Accelerations.</b> The left plot shows the relative longitudinal acceleration derived from (3-13). The right plot shows the relative lateral acceleration derived from (3-14). The sudden stop of the motor that lead to a standstill can be clearly seen. The accelerations did not go completely to zero due the fact that the external CVA was constantly tracking the points and even the slightest (observed, not actual) difference in marker positions would lead to a small value for the acceleration. . . . .	33
3-9	<b>Drift Experiment: Sideslip Angle.</b> The curve followed from (3-15). The data was very useful in the sense that both a positive and negative sideslip angle was obtained, as well as a short sustain of a high amplitude at 3 s. . . . .	33
3-10	<b>Validation Dataset: Longitudinal Motion</b> . . . . .	35
3-11	<b>Validation Dataset: Lateral Motion</b> . . . . .	35
4-1	<b>On-board camera placement.</b> The white part is a 3D-printed mount to place the camera high above the ground, facing downward. The material is stiff enough to prevent relative motion of the camera thus ensuring a fixed geometry. . . . .	37
4-2	<b>Image Aberration.</b> The on-board camera used in this project suffered heavily from image aberration. Consider the homogeneous image grid in Subfigure (a). Subfigure (b) shows what barrel distortion does to the grid, (c) shows the effect of perspective. . . . .	38
4-3	<b>Planar Projection.</b> The camera observes a point $q$ in three-dimensional (3D) space on a flat surface (plane) with defined $(x, y)$ coordinates in $[m]$ , at a depth $z$ from the camera. The image formed on the sensor is two-dimensional (2D) and the corresponding point $p$ has $(u, v)$ coordinates in $[px]$ . . . . .	39

4-4	<b>On-Board Homography Estimation.</b> Subfigure (a) shows the calibration image. Additional red markers were placed near the camera, to increase the accuracy of the estimation in this critical area where more pixels were available per meter. Subfigure (b) shows how the coordinate system was defined, the origin was assumed to be the location of the IMU. . . . .	40
4-5	<b>RGB-Colour Image.</b> A digital RGB image is a 3 channel array of $u \times v$ pixels, with the origin at the top left corner (a). Each colour channel contains intensity an intensity value from 0 to 255 (b) for each element in the array (c). The resulting example image is shown in (d). . . . .	41
4-6	<b>Image Processing Steps.</b> Step 1: correcting for the distortion by using the calibration parameters. Step 2: Applying red colour threshold. Step 3: Greyscale conversion. . . . .	41
4-7	<b>Detection-Tracking Cycle.</b> The features that the camera tracks may disappear from the image. Therefore, the detected points were tracked into the next frame, wherein a new set of points was detected for tracking into the next frame and so on. . . . .	42
4-8	<b>Corner Definition.</b> A corner in a greyscale image is defined as a pixel location where the gradient in both the $u$ - and $v$ -direction is large relative to the surrounding pixels. Consider the image array given in (a). Subfigure (b) shows the resulting pixel colours. The blue dot is what could be perceived as a 'corner'. The yellow dot is what the algorithm defines as the corner location in pixels (in this case (2,1)). The red arrows identify in which direction the gradient is high, while the green arrows show the direction where the gradient is low. . . . .	42
4-9	<b>Green Colour Threshold.</b> This figure shows the variation of the green colour threshold maximum $G_{max}$ , and how it negatively affected the accuracy when it was not suppressed. It shows an increase in noise (more extreme outliers) for a higher value of $G_{max}$ . . . . .	45
4-10	<b>Red Colour Threshold</b> The tuned parameter is the red colour threshold minimum $R_{min}$ . The optimum was found to lie between 75 and 125, depending on the other parameters. . . . .	47
4-11	<b>Green Colour Threshold</b> The tuned parameter is the green colour threshold maximum $G_{max}$ . The optimum was found to lie between 50 and 100, depending on the other parameters. . . . .	47
4-12	<b>Blue Colour Threshold</b> The tuned parameter is the blue colour threshold maximum $B_{max}$ . The optimum was found to lie between 150 and 200, depending on the other parameters. . . . .	47
4-13	<b>Minimum Distance.</b> The tuned parameter is the minimum distance between points to be detected $D_{min}$ . The optimum was found to be around 10 px. . . . .	48
4-14	<b>Number of Points.</b> The tuned parameter is the maximum number of points to be detected $N_p$ . The optimum heavily depended on the choice of $Q_{min}$ . . . . .	48
4-15	<b>Minimum Quality.</b> The tuned parameter is the minimum quality of the points to be detected $Q_{min}$ . The optimum heavily depended on the choice of $N_p$ . . . . .	48
5-1	<b>ENC: Axle Velocities.</b> The rate of acquisition is 50 Hz, but the rate at which the calculation provides a value depends on the velocity amplitude and can be lower. The block-shape of the signals is also explained by the nature of the calculation (see Section 2-2-1 for more details). . . . .	51
5-2	<b>IMU: Linear Accelerations.</b> The raw signals showed very noisy behaviour and even after filtering, the acceleration measurement still oscillates. Although the VAF of the longitudinal measurement is 0, it can still be observed that the signal somewhat follows the benchmark. . . . .	51

5-3	<b>IMU: Yaw Angular Rate.</b> In the most extreme part of the drift the markers tracked by the MCS were very vague in the picture which influenced the yaw rate measurement. The error between 2.5 and 3.5 s was thought to be an error of the MCS, not the on-board sensor. . . . .	51
5-4	<b>CVA: CG-Velocities.</b> The above plots show the measurements from the tuned CVA. The grey dots are the individual measurements. The green line is the filtered median over these points. The zero measurements are automatically discarded by the algorithm. . . . .	52
5-5	<b>IMU: Kinematic Model Result.</b> The above plots show the output of the kinematic model given in (5-1) and (5-2) for the inputs shown in Figure 5-2 and 5-3. . . . .	52
5-6	<b>Absolute Estimation Error for IMU and CVA.</b> This figure shows the estimation errors of the signals in Figures 5-4 and 5-5. The signal drift of the IMU with kinematic model can clearly be seen towards the end, as well as the outliers in the CVA signal. . . . .	52
5-7	<b>Sensor Fusion Overview.</b> The ENC and IMU provided measurements at 50 Hz and were processed by the LPF from (4-18). The CVA needed the $\omega_z$ value from the IMU for the calculations in (4-14) and (4-15). In this diagram the combined approach is shown. In the individual approach the average and LPF blocks in the CVA section did not exist. The hold block synchronised the CVA to the 50 Hz of the IMU, ENC and Kalman filter. The prediction-correction cycle produced the final outputs $V_x$ and $V_y$ . . . . .	54
5-8	<b>Sideslip Estimation Approach Comparison.</b> The IMU signal is the output from the kinematic model, which showed significant drift. The CVA showed noisier behaviour. Fusion was the best solution, with the combined approach being superior. . . . .	56
5-9	<b>Combined Fusion Approach Validation.</b> The left plot shows the estimation of the longitudinal velocity for the experiment shown in Figure 3-10. The right plot shows the estimation of the sideslip angle for the experiment shown in Figure 3-11 . . . . .	57
A-1	<b>Vehicle Hardware Overview.</b> Part (1) is the chassis including the motor (2), ESC (3) and servo (4). Part (5) is the Arduino. The red lines indicate analog interfaces. Parts (6) indicate the encoders. Part (7) is the IMU, (8) is the camera. The blue lines indicate digital interfaces. Part (9) is the Ordoid. . . . .	63
C-1	<b>Process Noise Covariance on <math>V_x</math></b> . . . . .	69
C-2	<b>Process Noise Covariance on <math>V_y</math></b> . . . . .	69
C-3	<b>Measurement Noise Covariance on <math>V_\delta</math></b> . . . . .	70
C-4	<b>Measurement Noise Covariance on <math>V_{xn}</math></b> . . . . .	70
C-5	<b>Measurement Noise Covariance on <math>V_{yn}</math></b> . . . . .	70
C-6	<b>Process Noise Covariance on <math>V_x</math></b> . . . . .	71
C-7	<b>Process Noise Covariance on <math>V_y</math></b> . . . . .	71
C-8	<b>Measurement Noise Covariance on <math>V_\delta</math></b> . . . . .	71
C-9	<b>Measurement Noise Covariance on <math>V_{xN}</math></b> . . . . .	72
C-10	<b>Measurement Noise Covariance on <math>V_{yN}</math></b> . . . . .	72
C-11	<b>Drift Experiment: Rear Wheel Longitudinal Slip: <math>\lambda</math></b> . . . . .	72



---

# List of Tables

1-1	Contemporary Automated Drift Control Literature Survey Overview . . . .	11
5-1	Individual Fusion Algorithm Parameters . . . . .	55
5-2	Combined Fusion Algorithm Parameters . . . . .	55
A-1	Vehicle Component Specifications . . . . .	64
B-1	Computer Vision Algorithm Parameters for External Camera. . . . .	66
B-2	Computer Vision Algorithm Parameters for On-Board Camera. . . . .	67



---

# Preface

This thesis has been written for a Master of Science in Mechanical Engineering at the Delft University of Technology, in particular as part of the track Vehicle Engineering [1]. As a child I always loved drawing vehicles of all sorts, and this master programme drew my attention. Vehicle engineering encompasses the fields of automotive, aerospace, rolling stock and marine engineering. It encompasses elements of mechanical, electrical and software engineering as applied to the design, manufacture and operation of all sorts of vehicles and their respective engineering subsystems. Automotive engineering in particular, is vehicle engineering applied to automotive systems such as motorcycles, passenger cars and trucks. Some of the engineering attributes and disciplines that are of importance to this field are, for example: kinematics, mechatronics, aerodynamics, materials science, and ergonomics.

During my master programme I became especially interested in control systems design and state estimation. Professor Hellendoorn encouraged me to specialise in this field. Then I came into contact with Professor Corno, and we initiated a joint project between Delft and Milan. Preceding the writing of this thesis, a project was carried out at the Politecnico di Milano where Matteo was my daily supervisor together with Federico Roselli from October 2016 until March 2017. This thesis was written between March 2017 and June 2017 in Delft, where Hans and Matteo continued to provide me with feedback in the writing process. What really pleases me is to have inspired a bachelor student project to replicate the small scale vehicle that I used in my project. They improved its design significantly. Both Professor Hellendoorn and Professor Corno have the intention to continue collaborating on future projects regarding vehicle engineering.

Delft, June 30, 2017





---

# Acknowledgements

I would like to thank **Matteo Corno** for being a very involved supervisor, who was always available and provided me with help on a daily basis while in Milan. He truly encouraged me to make the most out of this project and his feedback was very constructive and helpful.

I would like to thank **Hans Hellendoorn** for motivating me to specialise in this challenging subject, after all I am very happy with this decision. He also provided clear guidelines, and his sober approach is very pleasing to work with.

I would like to thank **Federico Roselli** for his kind cooperation during my project in Milan. He brought the parts for the experimental vehicle from the USA and explained to me the working principles which saved me a lot of time and effort.

Finally I would like to thank **Vasco Previtali**, who was very helping in the laboratory. He made adjustments to the vehicle possible with his skills and experience with electronic hardware.



This thesis is dedicated to my grandfather, Dr. Onno Fokkens (1930-2017),  
who inspired me to become an engineer.



---

# Chapter 1

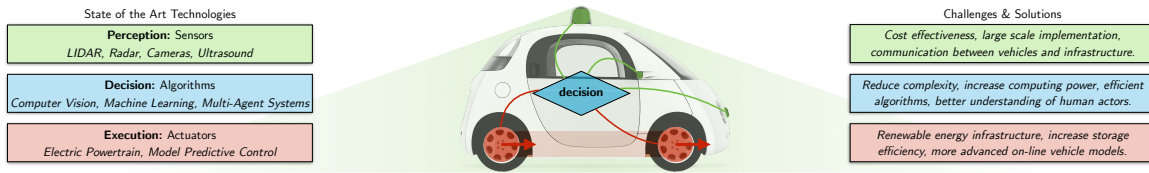
---

## Introduction

### 1-1 Automated Driving

#### 1-1-1 Origin in Vehicle Safety

Road traffic accidents account for more than a million deaths per year globally [2]. To tackle this problem, car manufacturers heavily invest in research and development in the area of vehicle safety. Vehicle safety encompasses passive, active, as well as pre- during- and post-crash elements to minimise the probability of an accident as well as the severity of injury [3]. Especially the active component of vehicle safety system development has lead to a better understanding of vehicle dynamics and the introduction of vehicle dynamic control systems. The anti-lock braking system (ABS) in 1978 [4] was one of the first vehicle active safety systems that reached the market. In the past 40 years innovation has continued to further improve vehicle active safety. Motorsport has always been a contributor to the understanding and mastery of all performance related vehicle elements as well. Vehicle dynamic control systems have not been an exception when it came to the improvement of vehicle performance. Traction- [3] and stability control [5] are examples of what motorsport has brought to improve road safety by leveraging vehicle dynamic control. As the innovation in vehicle active safety continues, it becomes evident that at some point the human driver, who is in many cases the cause of an accident [6], should be assisted or even be entirely replaced by the active safety system. This innovation within vehicle engineering is called vehicle control automation, or better known in popular terms as autonomous driving, and is currently of particular interest to the general public. The field of vehicle control automation is a fusion between vehicle engineering and computer science as it involves the use of mechatronics, artificial intelligence, and multi-agent systems to replace the human vehicle operator [7].

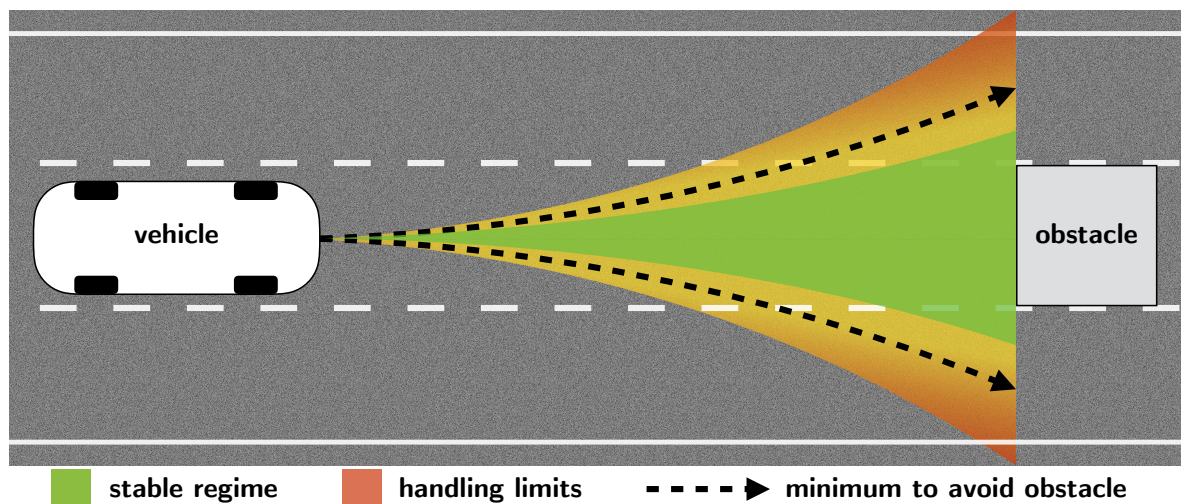


**Figure 1-1: State of the Art Technologies and corresponding Challenges and Solutions.** Autonomous vehicle operation comprehends three stages. Perception (green) is currently achieved with extensive sensor suites but they should be simplified to be implemented on large-scale. Decision (blue) currently requires powerful computers because of the complex human actors that are involved in traffic. The execution stage (red) is how the system acts on the decisions, with mostly electronic actuators. Electrical energy storage is difficult to implement on a large scale because of the lack of infrastructure. Also, the bottleneck of advanced vehicle control is usually the vehicle model of which the identification is still difficult in a mass production setting.

### 1-1-2 State of the Art and Future Challenges

Vehicle operation in general could be divided in three stages [8], as shown in Figure 1-1. To perceive the environment an intelligent vehicle needs advanced sensors. Current autonomous vehicles are equipped with very extensive sensor suites that are not yet ready for mass production due to cost and complexity. With the perceived information from the sensors the software needs to analyse and interpret the situation and make decisions on what control actions to take. This is referred to as *high-level control*. Automating this stage is challenging, since it requires some form of intelligence. The resulting product is therefore called an *intelligent vehicle*. There is an entire field of computer science research focussed on artificial intelligence in general, of which the resulting insight can be applied to tackle this problem. Once the decisions are made, the control actions need to be executed by actuators, which need to be individually controlled as well, referred to as *low-level control*. This field within vehicle engineering is more mature since it has been applied in the earliest active safety systems. Still, difficulties remain in pushing the boundaries of the vehicle when it comes to executing on control actions.

The operation differs significantly between the urban- and highway environment. In urban environments the velocities are low, there are many different road-users and the traffic structure is less strict. The problem here is that the implementation of intelligent vehicle systems requires the software to recognise the intentions of complex road users such as pedestrians and cyclists to make high-level control decisions [9]. The solutions to these problems lie more in the realm of computer science and human behaviour studies than that of vehicle engineering, as it applies to a multitude of actors and involves the application of artificial intelligence [10]. The highway environment on the contrary, is different: at the one hand the velocities are high, at the other hand there are both fewer and more homogeneous road users, who adhere to a clear traffic structure. In an emergency situation on the highway, where an obstacle needs to be avoided for example, it is challenging for the intelligent active safety system not to lose control over the vehicle. The vehicle should be operated at the handling limits if the collision is to be avoided in such a situation [11], as visualised in Figure 1-2. This topic is strongly related to vehicle dynamics because it requires a thorough understanding of tire- and vehicle dynamic behaviour to develop control systems for the limits of vehicle handling.



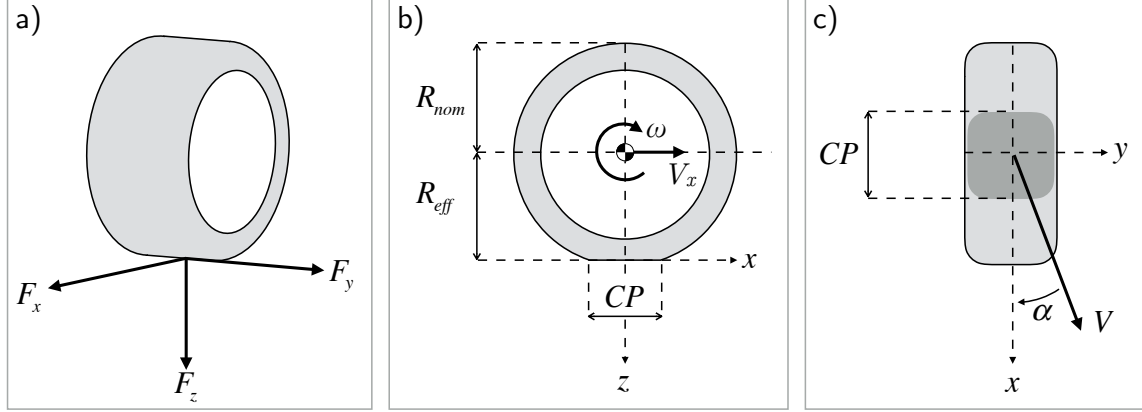
**Figure 1-2: Obstacle avoidance at high velocity.** The green area represents the set of potential paths on which the vehicle would have small slip angles. Outside of this region, the steep steering angle input would lead to high amplitude sideslip, in which the vehicle becomes unstable. The dotted black line represents the path on which the vehicle would just barely avoid a collision. This situation thus requires the vehicle to be handled in the unstable regime.

### 1-1-3 Vehicle Control at the Handling Limits

The handling limits of a vehicle are defined by its tire-road interaction, which is the main interaction between a vehicle and its environment. In stable operation, the tires adhere to the road and the vehicle is easily controllable. The handling limits of a vehicle are reached when the tires lose grip at a point where they start sliding over the road surface. This phenomenon is referred to as tire slip. Once the tire slip is at its maximum, there is hardly any adhesion between the tires and the road and this causes the vehicle behaviour to become highly unstable [12]. Historically, this vehicle behaviour has been undesirable and to be avoided with a human driver behind the wheel. The problem is that an average human driver is not capable of controlling the vehicle in this unstable operational regime [13]. Many vehicles on sale today are capable of assisting a human driver during highly dynamic manoeuvres to minimise the slip and maximise the vehicle handling stability. This has also been the main purpose of research and development in vehicle dynamic control in the past (such as in [14] and [15]).

Vehicle instability on itself *can* be desirable however, especially where it provides higher manoeuvrability in some situations, as the one described in Figure 1-2. Experienced race drivers have developed the skills to control vehicles in the unstable operation regime and ‘drifting’ is an established form of motorsport. Drifting is not for showing off; rally drivers in the world rally championship [16] have proven for years that drifting through some very sharp corners is faster than driving with minimal slip. The rise of vehicle control automation opens up new possibilities to expand the vehicle operation regime, provided a computer has the ability to achieve vehicle control with far better precision and aggression [17] than a human. In the past decade, research has been conducted on both human- and computer controlled drifting to better understand vehicle control at the handling limits [18] [19] [20]. Expanding the operating region of an intelligent vehicle opens up possibilities for avoiding collisions by leveraging the full capacity of the tires and beyond.

## 1-2 Tire and Vehicle Dynamics



**Figure 1-3: Tire Model Parameters: Definitions.** Subfigure a) shows the forces that are exerted on the road surface at the tire-road contact point, b) and c) show a side- and top-view to visualise the dynamic parameters used to describe the tire behaviour.

### 1-2-1 Tire-Road Interaction

To understand the handling limits of vehicles and to support the motivation behind the study of drift control, the tire-road interaction needs to be described. Whereas aerodynamics are important when driving in a straight line, the tire-road interaction is the most significant interaction between the vehicle and its environment in cornering and thus essential to understand in the context of drift control. The understanding of the basic principles of tire- and vehicle behaviour originates from modelling the fundamental dynamics, as done by Hans Pacejka [12]. Figure 1-3 shows definitions of dynamic parameters of a tire that are important within the scope of this thesis. The tire with nominal radius  $R_{nom}$  presses upon the ground surface due to the vertical load  $F_z$  resulting in a contact patch  $CP$  and an effective rolling radius  $R_{eff}$ . The vertical load  $F_z$  is induced by the weight of the body, which is dynamically distributed over the wheels. The longitudinal and lateral forces,  $F_x$  and  $F_y$ , are the main forces that influence the planar vehicle body motion. An important parameter in a tire model is the friction coefficient  $\mu$  which defines how the longitudinal and lateral forces relate to the load:

$$\mu_x = \frac{F_x}{F_z}, \mu_y = \frac{F_y}{F_z} \quad (1-1)$$

The tire travels at a velocity  $V$ , not necessarily parallel to the longitudinal vehicle axis  $x$ , resulting in lateral slip, defined by the sideslip angle  $\alpha$ . The wheel has an angular velocity  $\omega$  around its central axis. The longitudinal response of a tire is mainly generated by a relative motion between tire elements in contact with the road surface and tire body which causes slippage between the tire and the road; thus  $\omega$  is not necessarily analogous to the longitudinal velocity  $V_x$ . This results in the longitudinal tire slip  $\lambda$  in (1-2) given by  $\lambda_a$  for accelerating where  $\omega R_{eff} > V_x$  and  $\lambda_b$  for braking where  $\omega R_{eff} < V_x$ .

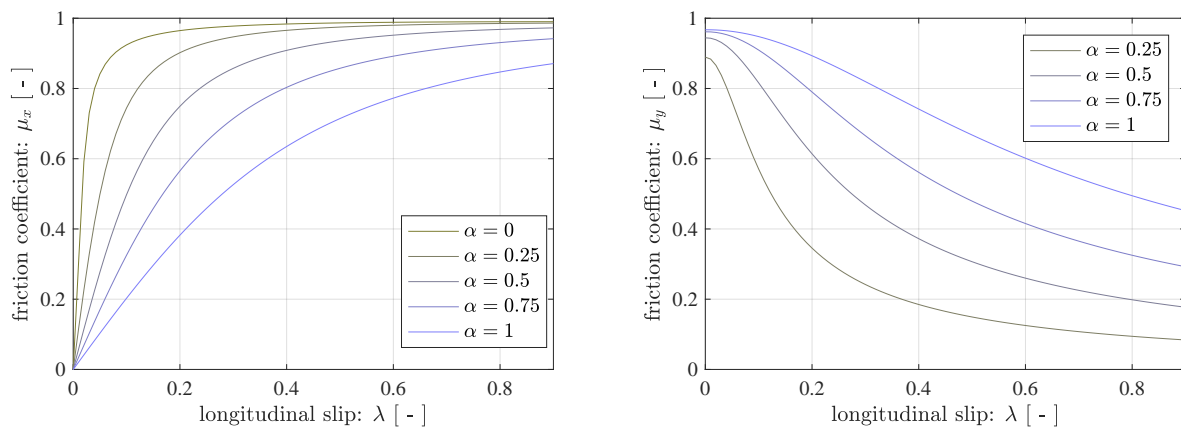


$$\lambda_a = \frac{\omega R_{eff} - V_x}{\omega R_{eff}}, \lambda_b = \frac{\omega R_{eff} - V_x}{V_x} \quad (1-2)$$

There are many different tire models, and depending on the application they can have a few to thousands of parameters. The models always contain however, either directly or indirectly, the dynamic parameters that have just been given. The resulting tire forces are thus a function of the previously defined inputs:

$$[F_x, F_y] = \mu(F_z, \alpha, \lambda) \quad (1-3)$$

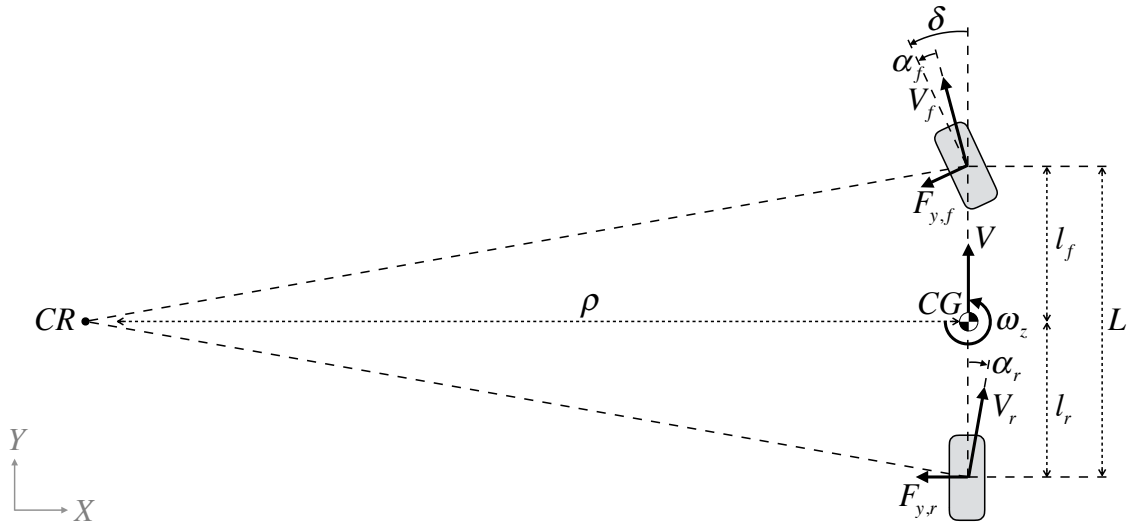
A model that has been developed by Howard Dugoff [21] will be referred to here as an example to visualise the dynamic relationship between the slip parameters  $\alpha$  and  $\lambda$  and the friction parameters  $\mu_x$  and  $\mu_y$ . Figure 1-4 shows the model output for a range of  $\alpha$  and  $\lambda$ .



**Figure 1-4: Tire Model Parameters: Dynamic Relationships.** It can be clearly observed in the right plot that the lateral friction decreases with longitudinal slip. This is an important phenomenon in drifting, since it implies that the lateral motion of the vehicle can be indirectly influenced by a longitudinal input on the wheels.

### 1-2-2 Cornering Motion

The lateral tire forces are the most dominant factor in cornering motion because they provide the only resistance to the centrifugal force on the centre of gravity  $CG$ . The previously described tire models can be used in a vehicle motion model to analyse its cornering stability. The simplest kinematic description of planar vehicle motion that captures the lateral dynamics well enough to understand stability is the single-track model, which is also known as the ‘bicycle model’ [22]. It is used in Figure 1-5 to provide insight into cornering motion of vehicles and the related stability.

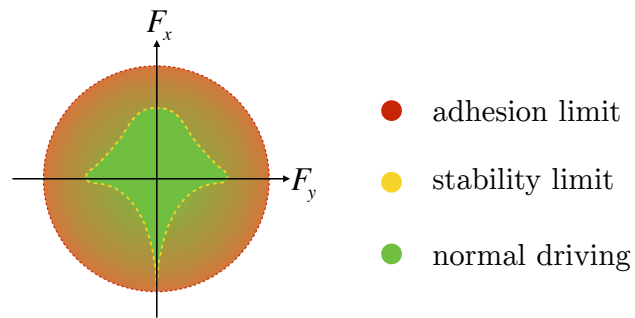


**Figure 1-5: Steady-State Cornering Motion.** The vehicle corners around a centre of rotation  $CR$  at a cornering radius  $\rho$ .

The bicycle model contains a mass  $m$  at the  $CG$ . The front axle is located at a longitudinal distance of  $l_f$  from the  $CG$ , the rear axle at  $l_r$ . The front wheel is oriented under a steering angle  $\delta$ . The cornering at velocity  $V$  results in a yaw angular rate  $\omega_z$ . As can be seen in Figure 1-4, tire behaviour is linear for small slip values. This linear behaviour can be approximated by the cornering stiffness given in (1-4).

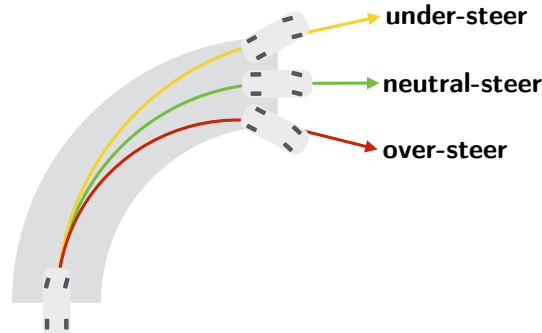
$$C_\alpha = \left. \frac{\partial F_y}{\partial \alpha} \right|_{\lambda=0, \alpha=0} \quad (1-4)$$

The cornering stiffness is defined by the tire adhesion limit. The maximum tire force limits velocities and accelerations in any planar direction of motion, not only cornering. This can be insightfully visualised using a friction ellipse [23], shown in fig. 1-6.



**Figure 1-6: Friction Limit Circle.** The outer red circle describes the theoretical maximum forces the tire can transmit between the ground and the vehicle. In longitudinal direction the practical maximum (yellow line) is limited by the maximum drive force that can be effectively delivered by the powertrain (driving). In lateral direction (cornering) the vehicle geometry limits the possible acceleration as well. For braking however, the limit is equal to the tire maximum, since the wheels can be locked and the tires will slide over the surface inducing the maximum deceleration to the vehicle. In a drift the rear tires move outside of the circle.

### 1-2-3 Stability Analysis



**Figure 1-7: Steady-State Cornering Motion.** The grey area represents a corner with a constant radius to visualise the differences between over- and under-steer in terms of that variable. The yellow line visualises an increasing cornering radius and the red line a decreasing radius. This phenomenon is leveraged by professional rally drivers to take sharper corners.

The cornering stiffness given in (1-4) can be used to describe cornering stability by means of the under-steer gradient, from the SAE definition in [24]:

$$K_{US} = \left( \frac{F_z}{C_\alpha} \right)_f - \left( \frac{F_z}{C_\alpha} \right)_r \quad (1-5)$$

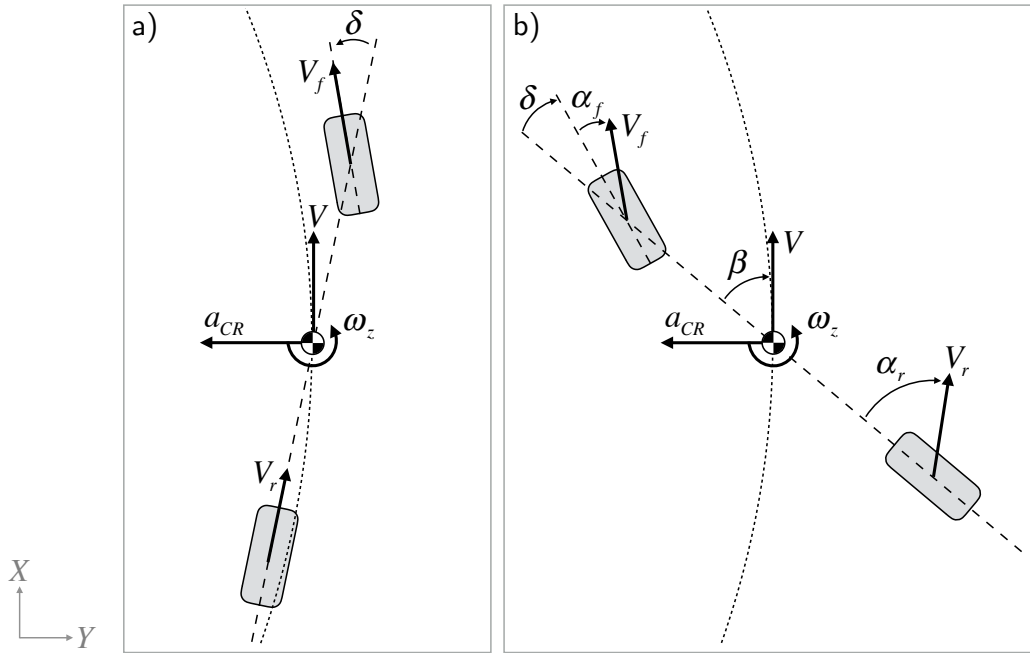
The cornering stability can be defined as: under-steer  $US$  for  $K_{US} > 0$ , neutral-steer  $NS$  for  $K_{US} = 0$  and over-steer  $OS$  for  $K_{US} < 0$ . Figure 1-7 shows the vehicle response for a vehicle that accelerates through a corner while maintaining a constant steering angle  $\delta$ . The response differs for the understeer gradient as follows:

- neutral-steer: constant cornering radius, equal slip angles
- under-steer: increasing cornering radius, larger front slip angle
- over-steer: decreasing cornering radius, larger rear slip angle

Basic stability analysis performed by Winkler (1998) [25] revealed a stable steady-state cornering motion for vehicles with under-steer characteristics. For vehicles with over-steer, stability is lost for:

$$V > V_{\text{crit}} = \sqrt{\frac{-L}{K_{US}}} \quad (1-6)$$

### 1-3 Drift Motion and Control



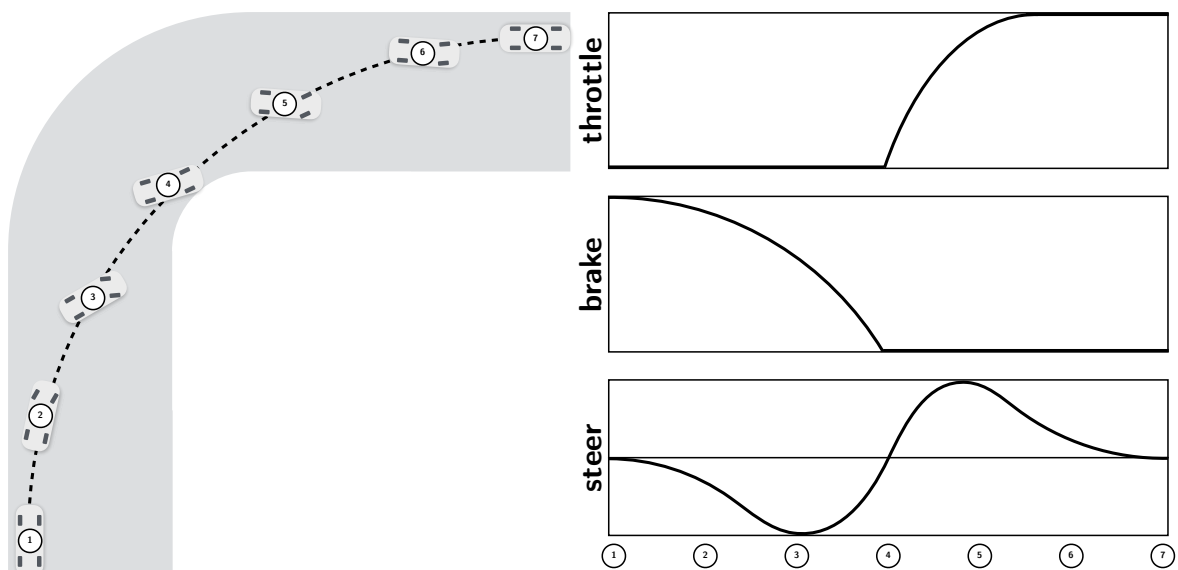
**Figure 1-8: Steady-State Cornering Motion.** Subfigure (a) shows stable cornering motion while Subfigure (b) shows unstable cornering (drifting) where  $\beta$  is large. Also, (b) shows a counter-steer configuration, where  $\delta$  is opposite to  $\omega_z$ .

#### 1-3-1 Drift Motion Analysis

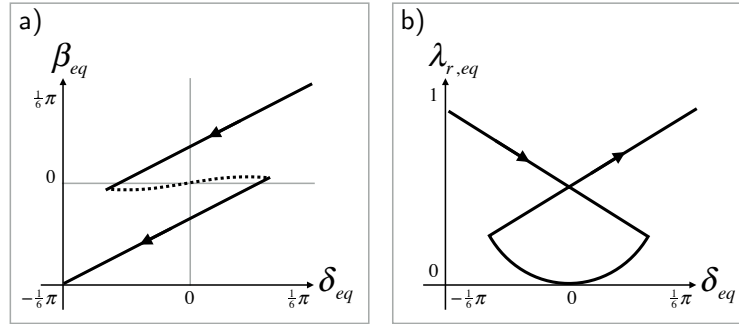
In Section 1-1-3 it was described how in some emergency situations an obstacle can only be avoided when an oversteer vehicle would move outside of the stable operating regime. As the previous section showed, the boundary of the stability region is where the tires lose grip and start sliding, a phenomenon referred to as drifting. Abdulrahim (2006) [18] explored physical parameters affecting drift motion through simulation and experimentation. Edelmann et al. (2009) [26] performed similar research as Abdulrahim and discussed handling characteristics of vehicle drift motion, and validated the theory in an experiment. Figure 1-8 shows a schematic representation of the difference between stable and unstable cornering motion. Both Abdulrahim and Edelmann concluded that equilibria exist, in both case a) with minimal slip, and in case b) with high tire slip  $\alpha$  and a large vehicle sideslip angle  $\beta$ . The difference between the two equilibria is the stability: a) is stable, b) is unstable. Another difference, apart from the large slip angles, is the longitudinal slip saturation in the rear tires, which causes the lateral friction from (1-1) to decrease significantly allowing for the lateral motion of the rear tires (as was explained in Section 1-2-1). Looking at the slip curves in Figure 1-4, it can be seen that the lateral force decreases with an increasing longitudinal slip. Therefore the lateral friction of (1-1) decreases, which destabilises the equilibrium.

### 1-3-2 Race Driver Behaviour Studies

When inexperienced drivers encounter a situation where they have to control their vehicle at a large sideslip angle  $\beta$ , they usually lose control and are not able to follow their desired cornering trajectory. Experienced race-drivers have developed the skills to control the vehicle in the unstable region of high sideslip. While Edelmann [26] and Abdulrahim [18] provided fundamental insight into the dynamics of a drifting vehicle, Dr. E. Velenis (with international collaborators) at the School of Engineering and Design, Brunel University, United Kingdom, approached the problem from a motorsports perspective. He compared numerical optimal corner solutions with expert race driver control inputs in simulation environments (CarSim, [27]). The work started with a mathematical analysis in 2005 [28]. Two rally-racing techniques for high-speed cornering were analysed: the trail-braking (see Figure 1-9, taken from [29]) and the pendulum-turn. Although these techniques were extensively used since the 1980's in rally racing, until the publication of this paper it had not been proven that these are in fact the optimal cornering solutions for certain situations. A single-track model was incorporated in a numerical optimisation scheme which was compared with minimum-time cornering problems 'solved' by expert rally drivers in action. The numerical optimal solution results were similar to the trail-braking technique for single 90 degree turns. The pendulum-turn technique would be better at coping with consecutive corners on low friction surfaces.



**Figure 1-9: Empirical Schematic Description of the Trail-Braking Manoeuvre.** At point 1 the driver initiates the corner after having stepped on the brakes for an amount of time depending on the length of the straight before the corner. The brake is released slowly until point 4 (the corner apex). While the steering angle is applied as a sinusoid, representing 'sliding mode control' behaviour. The throttle is gently introduced in the corner exit.



**Figure 1-10: Drift Equilibria Examples.** Subfigure (a) shows two linear sets of equilibria, one for a positive, the other for negative sideslip angle. The trajectory indicated with the arrow in (a) at the right-top corner corresponds to the left-top corner in (b). A counter-steer drift configuration as visualised in Figure 1-8 (b) is defined as a state where the sign of  $\delta$  is opposite to the sign of  $\beta$ . This can be clearly be observed in Subfigure (a).

### 1-3-3 Automated Drift Control Development

A research group at Stanford was the first to implement a drift controller on experimental vehicles. These vehicles were designed and built by researchers at Stanford. Unstable drift equilibria have been numerically solved for models of the setups, of which an example is given in Figure 1-10. Voser et al. (2010) [20] had the objective to stabilise their experimental vehicle at the found unstable sideslip equilibria. A consecutive analysis by Kritayakirana et al. (2012) [17] focused on designing a control system that could perform minimum-time cornering (and thus operate at the tire limits, see Figure 1-6). Hindiyeh et al. (2014) [30] continued with the work performed in [20]. The study had the same objective to stabilise the vehicle at a desired slip angle, but a secondary objective was added: to control the longitudinal velocity around a desired value simultaneously. Goh et al. (2016) [31], essentially applied the same technique on a different experimental setup, now including weight transfer on a two-track control model for their control design to achieve very long sustained circular drifts. All of the above experimental setups had the following in common: the tire-surface interaction dynamics were empirically identified and incorporated in the models, and the control systems used measurements from external sensors (differential GPS) to estimate the sideslip angle. Another research group that experimentally evaluated drift controllers were from Tohoku University, Japan. Nakano et al. (2014) [32] (2015) [33] developed a trajectory tracking drift control algorithm for a 1:10 scale vehicle with independently driven wheels. A state-feedback approach similar to that of the methods developed by Stanford was utilised to control the longitudinal velocity, sideslip angle, and yaw angular rate of the vehicle. As with Stanford, the models used surface friction data as well as external sensors, in this case a motion capture system (MCS) to estimate the vehicle states. A research group at BMW approached drift control from an industry point of view. Werling et al. (2015) [34] proposed a steering/throttle control strategy, which approximated the travelling direction of a reference point on the vehicle by neglecting the comparably small slip angles at the front wheels of the drifting vehicle. This simplified the input-output dynamics and made the derived control strategy robust to changing surface conditions, tyre parameters, and vehicle load. The main advantage of the proposed algorithm was that only geometric vehicle parameters and measurements of the standard stock sensors were required, which is suitable for a production setting. The presented literature is summarised in Table 1-1.

Table 1-1: Contemporary Automated Drift Control Literature Survey Overview

Publication	Scale	Sensors	Estimation	Actuators	Objectives	Model	Control	Validation	Results
Vöser, C., Hindiyyeh, R.Y. and Gerdes, J.C., 2010. <i>Analysis and control of high sideslip manoeuvres.</i>	1:1	DA-GPS, INS	DA-GPS, INS sensor fusion with Kalman filters.	RWD EM, FWS	Stabilisation to a constant $\beta$ and $V_x$	Bicycle w/ Fiala Lateral	State Feedback	✓	Stabilisation at $\beta = 25$ deg and $V_x = 8 \text{ m s}^{-1}$
Velenis, E., Katzourakis, D., Frazzoli, E., Tsiotras, P., and Happee, R., 2010. <i>Stabilization of Steady-State Drifting for a RWD Vehicle.</i>	1:1	None, simulation	None, simulation	RWD, FWS	Stabilisation to a constant $\beta$ and $V_x$	2-track, Pacejka	Linear controller for delta, sliding mode controller for rear slip	Simulation	Stabilisation at $\beta = 30$ deg and $V_x = 8 \text{ m s}^{-1}$
Kritayakirana, K. and Gerdes, J.C., 2012. <i>Autonomous vehicle control at the limits of handling.</i>	1:1	DGPS, INS	DGPS, INS sensor fusion	4WD ICE, FWS	Path tracking at minimum corner time	Bicycle w/ Friction Limit Circle	Feedforward and feedback using g-g diagram	✓	Path tracking at $V_x = 20\text{-}30 \text{ m s}^{-1}$
Hindiyyeh, R.Y. and Gerdes, J.C., 2014. <i>A Controller Framework for Autonomous Drifting: Design, Stability, and Experimental Validation.</i>	1:1	DA-GPS, INS	DA-GPS, INS sensor fusion with Kalman filters.	RWD EM, FWS	Stabilisation to a constant $\beta$ and $V_x$	Bicycle w/ Fiala Lateral	Dynamic surface w/ nested loop	✓	Stabilisation at $\beta = 20$ deg and $V_x = 8 \text{ m s}^{-1}$
Nakano, H., Kingawa, J. and Kosuge, K., 2014. <i>Control of a four-wheel independently driven electric vehicle with a large sideslip angle.</i>	1:10	MCS, IMU	MCS, IMU, observer with internal model requiring surface friction data	4W-ID EM, 4WS	Stabilisation to a constant $\beta$ and $V_x$	2-track, Linear Tire	State Feedback	✓	Stabilisation at $\beta = -40$ and $V_x = 2 \text{ m s}^{-1}$
Werling, M., Reinisch, P. and Gröll, L., 2015. <i>Robust power-slide control for a production vehicle.</i>	1:1	IMU	Simplified observer model assuming zero $a_f$ and constant drift geometry (initiated by human driver)	RWD ICE, FWS	Stabilisation of $\beta$ and tracking of $x$ - $y$ course	Bicycle w/ simplified non-linear Bakker tire model	First-order sliding mode	✓	Stabilisation at $\beta = 30$ deg and $V_x = 15 \text{ m s}^{-1}$
Nakano, H., Kingawa, J. and Kosuge, K., 2015. <i>Trajectory Tracking Control of a Vehicle with a Large Sideslip Angle</i>	1:10	MCS, IMU	MCS, IMU, observer with internal model requiring surface friction data	4W-ID EM, 4WS	Tracking of $x$ - $y$ course at a desired $\beta$ and $V_x$	Quarter-car model with linear tire for each independent driven wheel	State Feedback	✓	Path tracking at $\beta = -30$ deg and $V_x = 1\text{-}3 \text{ m s}^{-1}$
Goh, J.Y. and Gerdes, J.C., 2016. <i>Simultaneous stabilization and tracking of basic automobile drifting trajectories.</i>	1:1	DGPS, INS	DGPS, INS sensor fusion	RWD EM, FWS	Stabilisation to a constant $\beta$ and $V_x$ , tracking of circular reference	Bicycle w/ Fiala Lateral	State Feedback	✓	Circular path tracking at $\beta = -40$ and $V_x = 7 \text{ m s}^{-1}$

## 1-4 Vehicle Dynamic State Estimation

### 1-4-1 Drift Control Literature Evaluation

The literature survey on drift control presented in Section 1-3 shall be evaluated to identify potential issues. The studies showed promising results, but a critical element seems to be missing. The research lacks an emphasis on the state-estimation process and focusses merely on the control design and its results. However, state estimation is essential for the control designs to be meaningful and implementations to be possible, not something to be overlooked. This section reviews the drift control literature regarding the utilised state estimation methodologies.

The work by Velenis et al. was carried out in a simulation environment, so state estimation did not apply there directly. The work from Stanford on the other hand, which could be considered the largest contribution to automated drift control with the most promising results, included approaches that relied heavily on differential GPS-based slip angle estimates and fairly precise inverse models of the wheels. The experimental setups used a sensor fusion of a differential global positioning system (DGPS) [35] and inertial navigation system (INS) [36], which allowed for a down-sampled controller at 200 Hz. These methods were based on work by Ryu et al. (2004) [37]. Also, tire properties were known for the systems and real-time friction estimation was possible in this research, with previous work by Hsu et al. (2014) [38] making this possible. Beal et al. 2011 [39], also at Stanford, performed prior work on some of the used state estimation techniques as well. Important to note is that the vehicles used in these studies were designed and built at Stanford and thus, all parameters for the vehicles were known and evaluated again and again over the years. These kind of real-time measurements and (partly empirically-based) models are not available for production vehicles on changing road conditions. The work by Nagano et al. was performed on a test surface with known friction parameters, which is almost impossible from an implementation point of view. Also, the motion capture system fulfilled the same function as the differential GPS did in the experimental setups of the Stanford group.

The work by Werling et al. at the BMW lab showed results that could be useful from an implementation point of view. There are some issues, however, that should be discussed regarding the state estimation. The state estimation used the output of a single-track model to estimate the sideslip assuming a zero sideslip angle at the front wheels. The tire model for the single-track state space used estimates of the lateral tire force. Since the experiments were performed on a wet surface, it can be questioned whether the simplified tire model and assumptions would hold on a dry surface with higher or unpredictable friction (referring to [26], where the authors showed the highly non-linear tire behaviour in drift states). Also, the conducted drift experiment was of very short duration. This allowed for a state estimation technique based on integrals of measured accelerations. These estimates would drift away from the actual value if there was no way of validating the integral estimate over time. The results in the paper seem to show that after 10 seconds, the sideslip angle still deviates from the desired angle with about 10-15 deg. The authors also do not show their actual estimated  $\beta$  in the results.

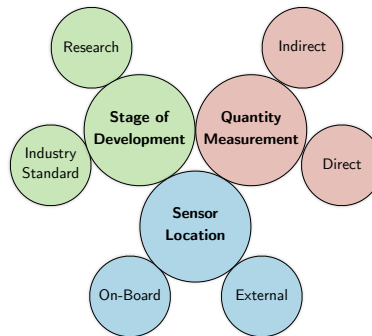


### 1-4-2 Established Methodologies

In control system design, it is essential to know the state of the system to apply appropriate control inputs and to verify the resulting response. To obtain the states, vehicle control systems rely on sensor measurements. Some states can be measured directly, others must be estimated based on a model or pre-defined relationship between measurements and states (indirect approach). Classical vehicle dynamic control design methods were usually based around the single-track model which needed state estimates of only the longitudinal velocity and the steering angle. Rotary encoders [40] were available sensors to measure the steering angle position and the wheel velocity (of individual wheels, if needed). When the encoders were appended with an inertial measurement unit (IMU) [41] to measure accelerations and yaw velocity, the sideslip angle could be estimated using a static model or a simple linear dynamic model in normal stable driving conditions where  $\beta \leq 2 \text{ deg}$  [42]. In extreme situations however, the coupling of the vehicle sideslip angle to various measured quantities in production vehicles is lost. Especially when the vehicle sideslip angle becomes large and sustained, knowledge about it is essential for a proper description of vehicle behaviour. Recall the definition of the sideslip angle:

$$\beta = \tan^{-1} \left( \frac{V_y}{V_x} \right) \quad (1-7)$$

The longitudinal velocity  $V_x$  can be measured with the encoders at low longitudinal slip values, however, as was described in Section 1-3-1, the longitudinal slip is very high at the rear wheels during drifting. In addition the sideslip angle requires a measurement of the lateral velocity  $V_y$  which can not be achieved with most vehicle stock sensors. The estimation of the lateral velocity has been a widely investigated topic in literature, even outside of the scope of drift control. Accurate online estimation of the lateral velocity enabled the improvement of existing-, and the development of new automotive safety systems. The studies conducted on sideslip estimation can be categorised depending on the utilised type of sensor measurements. The distinctions are summarised in Figure 1-11.



**Figure 1-11: Sideslip Estimation Methodology Categories**

The first distinction of studies is the development stage of the utilised technologies. The studies (such as [43], [44], [45] and [46]) utilised sensors that are considered ‘stock’ sensors: low-cost industry standard and present in all modern commercial vehicles. Examples of these sensors are wheel angular velocity encoders [40], steering angle encoders, IMU’s [41] and (low-cost) GPS receivers <sup>1</sup>. Other studies (such as [47], [48], [49], and [50]) used advanced additional sensors that are not present in normal commercial vehicles, and are merely used in research. examples of these sensors are: optical sensors [51], tire force sensors [52] [53] [54], high fidelity GPS receiver arrays [55] and DGPS setups [35]. The reason why these sensors are not industry standard (yet) is not only because of the prohibitively expensive costs for commercial use, but mainly because of the excessive complexity they introduce to the vehicle design which is problematic for mass-production vehicles. Another distinction between studies is the direct and indirect measurement of the sideslip angle. The former being an expensive and advanced technique while the latter is a more feasible technique for commercial implementation at the cost of less reliability. The final distinction is the sensor location: on-board or external. The DGPS setups utilise devices external from the vehicle, which requires an advanced smart infrastructure. This is definitely not available in all situations. This is why it would be preferred to be able to precisely estimate the sideslip exclusively with on-board sensors. Optical sensors are becoming more common in vehicle engineering research for sideslip estimation with on-board sensors. They are used in the same fashion as in computer mice. By pointing down on the ground surface, they capture the motion of features on the ground and by doing so deduce the vehicle motion. However, these sensors cope with implementation complexities (they do not work on wet surfaces for example) and have not penetrated the industry.

### 1-4-3 State Estimation in Intelligent Vehicles

Since automated drift control is applied in the context of intelligent vehicles, advanced on-board sensors will be a given for the reasons described in Section 1-1-2. Intelligent vehicles which are already on the market use cameras for advanced lane-keeping assistance (LKA) [56] and adaptive cruise control (ACC) [57]. Autonomous vehicle prototypes are equipped with extensive on-board sensor suites including radars, stereo vision, lasers and even combinations such as light detection and ranging (LIDAR) systems [58]. These sensors are meant to detect other road users and obstacles in sufficient detail to make tactical driving decisions in traffic [59]. However, these sensors could also be used to support the state estimation process of the vehicle stability control system. Especially the data coming from the cameras can be used to detect features in the surroundings of the vehicle and thus extract motion information. An interesting research topic would be to investigate whether a camera could be fused with traditional sensors such as encoders and an IMU to estimate the vehicle states for the use in automated drift control development. The very sensor that detects the obstacle to be avoided could also be used to provide state estimates for the control system that avoids the obstacle. This thesis will propose a state estimation technique that is a fusion between the old and the new.

---

<sup>1</sup>GPS is becoming more common in production vehicles for navigation, not necessarily for control purposes

## 1-5 Conclusion

### 1-5-1 Problem Formulation

To improve traffic- and vehicle safety and reduce related deaths, active safety systems have been developed. The most recent development in this area is vehicle control automation. One of the challenges is to maximise the operational regime of the control systems to be on the edge or even outside of the stable envelope of the tires and vehicle to be able to avoid accidents in extreme emergence situations. The analysis of drift motion in the past decade led to a better understanding of both the dynamics as well as its potential practical use in vehicle stability control design. Increasing the operational regime of automated vehicle control systems, as well as understanding vehicle behaviour in regions outside of stability is very useful. Section 1-3-3 discussed the current state-of-the-art in automated drift control, Table 1-1 summarises the results of the different studies. The survey revealed a lack of emphasis on implementation-feasible sideslip state estimation techniques in the most recent literature. The sideslip angle is a very important state to describe vehicle behaviour at the handling limits and real-time estimation of its state is necessary for automated drift control. Section 1-4 presented research focussed on sideslip estimation. The current state of the art research on automated drift control relies heavily on a direct measurement approach with very costly setups including external sensors. The presented methods also employ semi-empirical tire models which further complicates (maybe even prohibits) the potential implementation in production vehicles. This is a problem that needs to be solved to lay the groundwork for future implementation of drift control on production vehicles that do not have any external sensors available.

### 1-5-2 Thesis Objectives

The previous sections motivated the intention to start researching the topic of vehicle sideslip estimation with on-board sensors. This is considered to be a very relevant topic regarding the current developments in the field of vehicle engineering. As the literature survey showed, there seems to be a lack of attention given to the feasible implementation of sideslip estimation in the context of automated vehicles. This thesis should provide a solution to this issue. As was stated, it would be interesting to investigate the possibilities of fusing on-board stock sensors (encoders and an inertial measurement unit) with a camera, given that a camera is a standard sensor for an intelligent vehicle. This results in the following objectives:

- Develop a methodology to estimate the vehicle sideslip with encoders, an inertial measurement unit and a camera.
- Write a computer vision algorithm that can extract motion data from the camera.
- Develop a sensor fusion algorithm that fuses the measurements of the three sensors to estimate the sideslip.
- Design an experimental setup with a motion capture system that can externally validate the estimation.
- Implement the algorithm on an experimental vehicle and conduct validation experiments.

### 1-5-3 Project Outline

The Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano has provided a laboratory where the project to support this thesis has been carried out. The previously mentioned objectives have been successfully completed during the project and this thesis will provide the results as follows:

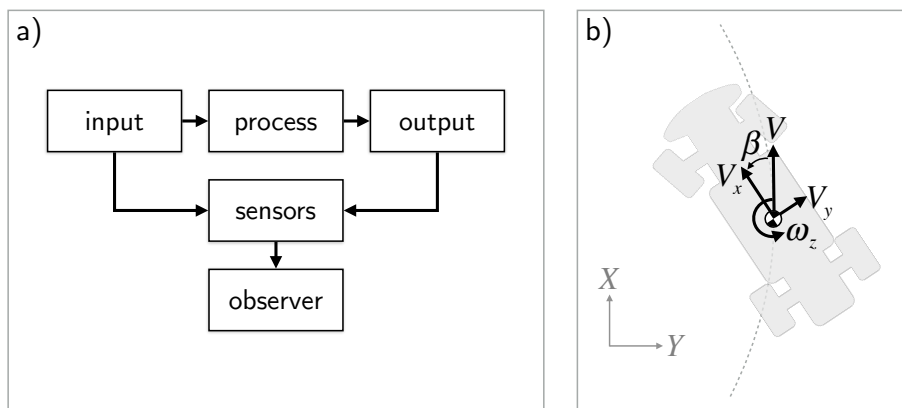
- In Chapter 2 the methodology is presented as well as the experimental testbed for validation data gathering.
- The processing of the gathered data is explained in Chapter 3.
- The on-board computer vision algorithm is presented in Chapter 4.
- Chapter 5 discusses the sensor fusion design and its results.

This thesis concludes with recommendations for follow-up joint research projects between the Delft University of Technology and the Politecnico di Milano.

# Sideslip Estimation Methodology

## 2-1 Introduction

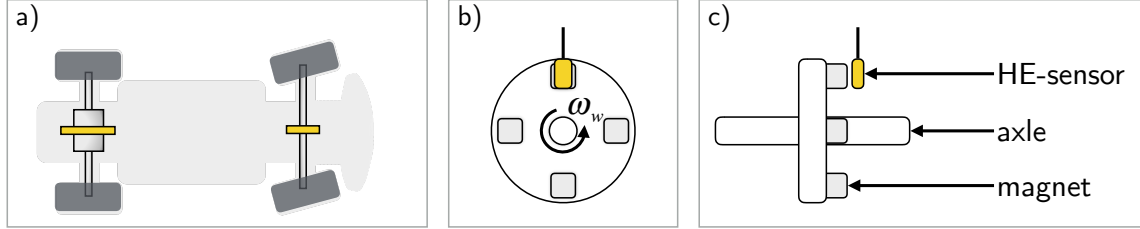
One of the objectives of this thesis, as it was explained in Section 1-5-3, was to estimate the sideslip angle of a vehicle exclusively with on-board sensors, namely: rotary encoders, an inertial measurement unit and a camera. This chapter presents the methodology to achieve that objective. The estimation process is given in Figure 2-1 a). The on-board sensor suite is laid out in Section 2-2 by explaining the basic working principles and relating the measured quantities of the sensors to the  $CG$ -states. The state observer design is explained in Section 2-3 by defining the process, inputs and outputs based on the available measurements. The experimental setup is presented in Section 2-4, in which the state estimation technique has been implemented and validated.



**Figure 2-1: Sideslip Estimation Process** (a) Shows a general schematic of how process states that can not be directly measured can be estimated by measuring inputs and outputs with sensors and a state observer. (b) Shows the process to be estimated: the  $CG$ -states of a drifting vehicle on a global  $X - Y$ -plane.

## 2-2 On-Board Sensor Suite

### 2-2-1 Rotary Encoders



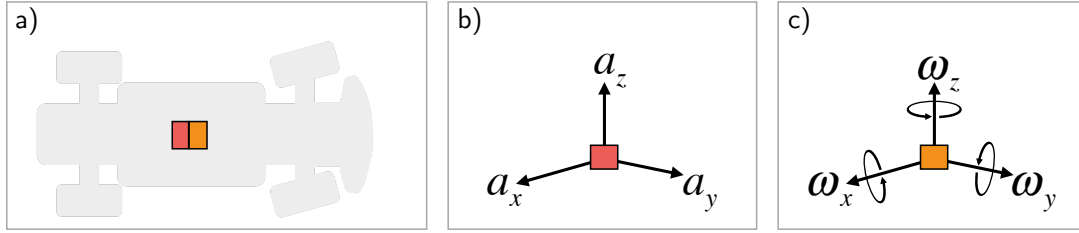
**Figure 2-2: Rotary Encoder Schematic.** (a) Shows the placement of the encoders on the front- and rear axles. (b) Shows a sideview and the directly measured quantity: the wheel angular rate  $\omega_w$ . (c) Shows the front view of the encoder. The displayed disk is located on the centreline.

A rotary encoder (ENC) is an electro-mechanical device that converts the angular motion of an axle to an analog code. The type that was used in this project is of the incremental, or relative variety. The raw output of the ENC provides information about the motion of the axle, which is further processed by software code into velocity data. The incremental measurements come from a Hall-effect (HE)-sensor that is attached along the axle. A HE-sensor is a transducer that varies its output voltage in response to a magnetic field. Four magnets are equally distributed over a disk attached to the axle, as shown in Figure 2-2. The rotation of the disk causes the magnets to pass by the HE-sensor four times per revolution. The angular rate of the wheel  $\omega_w$  can be calculated by either taking the time  $dt$  between two pulses, or by taking the average over multiple pulses for a less noisy (but more delayed) result. With a known nominal wheel radius the velocity can be calculated. The front wheel velocity is denoted as  $V_\delta$ , since the wheels under the steering angle actually travel in the direction of  $\delta$  and this should be accounted for. The ENC measurement of the front axle velocity can be written in terms of the  $CG$  states as (2-2), substituting (2-1). The rear wheel velocity  $V_r$  should be equal to  $V_x$  in case of minimal slip, as the vehicle is assumed to be a rigid body. The difference between the measured  $V_r$  and  $V_x$  can provide an estimate of the rear longitudinal slip (see Section 1-2-1).

$$V_{yf} = V_y - l_f \omega_z \quad (2-1)$$

$$V_\delta = \sqrt{V_x^2 + V_{yf}^2} \cos \left( \tan^{-1} \left( \frac{V_{yf}}{V_x} \right) - \delta \right) \quad (2-2)$$

### 2-2-2 Inertial Measurement Unit

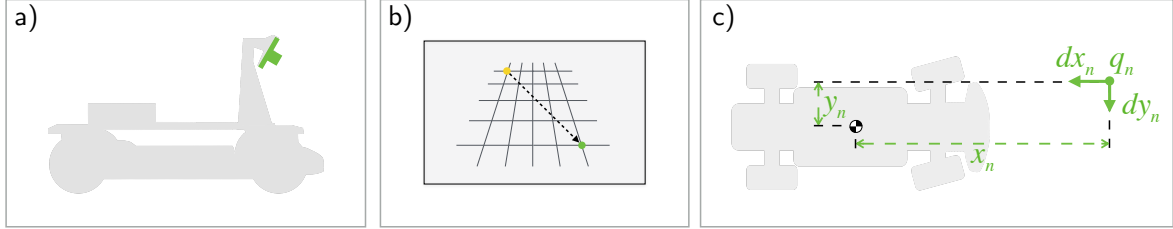


**Figure 2-3: Inertial Measurement Unit Schematic.** (a) Shows the placement of the sensor housing at the (approximate) *CG* of the vehicle. (b) Shows the linear accelerations measured by the accelerometer triad, (c) shows the the angular velocity sensor triad outputs.

The inertial measurement unit (IMU) is an electronic device which collects angular velocity and linear acceleration data. See Appendix A for technical specifications on the sensor that was used in this project. The IMU housing contains two separate parts, see Figure 2-3. The first part is an accelerometer triad. It generates three analog signals describing the accelerations along each of its axes produced by, and acting on the vehicle. The most significant of these sensed accelerations is caused by gravity. The second part is an angular velocity sensor triad. It also outputs three analog signals. These signals describe the vehicle angular velocity about each of the sensor axes. For this project the IMU was assumed to be located at the vehicle centre of mass. In fact, since the implementation was on a small scale, the actual difference between the *CG*- and IMU locations was small indeed. The accelerometer triad, and angular rate sensors within the IMU were mounted in such a way that their sensor coordinate axes were not perfectly aligned with those of the vehicle. This is due to the fact that the two sensors in the IMU were mounted in two different orientations in the housing, along with the fact that the axes of the IMU were not aligned with the vehicle axes. This was easily be accounted for in the signal processing after acquisition. The work from Selmanaj et al. (2016) [60] provided a useful example of a kinematic model that relates  $(a_x, a_y)$  to their respective velocity derivatives  $(\dot{V}_x, \dot{V}_y)$  and the yaw angular rate  $(\omega_z)$ , all of which could be measured directly by the IMU.

$$\begin{cases} a_x(t) = \dot{V}_x(t) - \omega_z V_y(t) \\ a_y(t) = \dot{V}_y(t) + \omega_z V_x(t) \end{cases} \quad (2-3)$$

### 2-2-3 Camera with Computer Vision



**Figure 2-4: On-Board Camera Schematic.** (a) Shows the placement of the camera on the vehicle in a side view. (b) Shows an abstract representation of the image. The CVA detected a feature in the previous image (yellow point location), and tracked it into the current image (green point location). The algorithm projects the image point locations  $p$  in pixels to ground surface locations  $q$  in meters. (c) Shows for a point  $q_n$  a representation of the measured quantities.

The final sensor was a camera that was placed at the front of the vehicle, as shown in Figure 2-4. The placement on the mounting structure was such that the lens pointed as much vertically down as possible, to see only floor from above with the least perspective. The mount was designed in such a way that it could also be used for other studies on the vehicle, such as the design of an LKA system, which requires a view far ahead. In that case the camera can be tilted upward. The on-board software made use of a custom designed computer vision algorithm (CVA) to detect and track features in the images. Considering the scope of this thesis, the tracking problem was simplified by placing red markers on the test surface in the lab. This allowed for a less computationally heavy algorithm because less pixels (only the red ones) needed to be searched for features to track. Several algorithms have been developed that are able to detect and track features in complex images, such as [61], and applying these algorithms to this project would be an interesting follow-up study. In this project however, a simpler CVA was designed. The CVA as it was designed in this project measured for a number of  $N$  points: the distance of the detected point  $n$  from the  $CG$  ( $x_n, y_n$ ) [ m ] and the displacement over time with respect to the previous frame ( $dx_n/dt, dy_n/dt$ ) [ m s<sup>-1</sup>]. The CVA deduces the  $CG$ -states from these measurements. This process is called *egomotion* estimation. The CVA measurements can be written in terms of the  $CG$  states by (2-4) and (2-5). As can be seen in the equations, the  $\omega_z$  measurement from the IMU was needed as well to operate properly. A detailed description of the design and the tuning of the parameters in the algorithm is presented in Chapter 4.

$$V_{xn} = -(V_x + y_n \omega_z) \quad (2-4)$$

$$V_{yn} = -(V_y - x_n \omega_z) \quad (2-5)$$



## 2-3 Dynamic State Observer

### 2-3-1 Process and Measurement Models

The on-board sensors presented in the previous section could not measure the  $CG$ -states directly. This required the design of a dynamic state observer (Figure 2-1 (a)). Many approaches to observer design are based on a state-space process model [62]. The process model describes the transformation of the  $CG$ -states ( $x$ ) subjected to the input ( $u$ ). This can be represented as the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2-6)$$

In addition, a measurement model describes the relationship between the  $CG$ -states and the on-board sensor measurements ( $z$ ). The model can be represented with the linear expression:

$$z_k = Hx_k + v_k \quad (2-7)$$

The terms  $w_k$  and  $v_k$  are stochastic variables representing the process- and measurement noise respectively. They are assumed to have probability distributions described by the process noise covariance  $Q$  and measurement noise covariance  $R$ .

### 2-3-2 Kalman Filter

The tool that was used in this project for the  $CG$  state estimation from the on-board sensor measurements is the Kalman filter [63] [64], due to its simplicity and robust nature. The filter estimates the  $CG$  states while obtaining feedback in the form of measurements. The filter operates in two stages: a time update and a measurement update. The time update equations are responsible for projecting the current state and error covariance estimates forward in time to obtain the a priori estimates for the next time step (prediction):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (2-8)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2-9)$$

The measurement update equations incorporate a new measurement to improve the a posteriori estimate (correction):

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2-10)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (2-11)$$

$$P_k = (I - K_k H) P_k^- \quad (2-12)$$

The in- and output measurements were supplied by the on-board sensor suite as described in Section 2-2. The Kalman filter was used to combine or fuse the measurements from the ENC, IMU and CVA. By weighing the different sensor measurements in the areas where they each perform best (by changing the  $Q$  and  $R$  matrices), the filter provided more accurate and stable estimates than a system based on any one sensor alone. The described process is therefore referred to as *sensor fusion*, and its application was essential to reliably estimate the vehicle sideslip angle.

### 2-3-3 Sensor Fusion Design Proposal

The Kalman filter as it is presented in Section 2-3-2 requires the definition of a process- and measurement model. These models can take different forms and a decision needed to be made which one would be worked out in this thesis. The state vector of the model were defined by the longitudinal- and lateral velocity at the  $CG$ , which were to be estimated because they can not be measured directly with the available sensors:

$$x = \begin{pmatrix} V_x \\ V_y \end{pmatrix} \quad (2-13)$$

These states can be predicted by means of accelerations that can be directly measured. Therefore the input vector was defined by the linear accelerations at the  $CG$ , measured by the IMU:

$$u = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (2-14)$$

The process was defined by a state space model based on the kinematic description given in (2-3):

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \end{bmatrix} = \begin{bmatrix} 0 & \omega_z(t) \\ -\omega_z(t) & 0 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \quad (2-15)$$

The ENC and CVA measurements were used to define the measurement model. The rear ENC measurement was subject to significant longitudinal slip so it was not be used in the measurement model. The CVA outputs for a number of  $N$  points the deduced longitudinal ( $V_{xn}$ ) and lateral velocity ( $V_{yn}$ ) components at the  $CG$  (left measurement vector in (2-16)). These measurements can also be combined by taking their median, resulting in  $V_{xN}$  and  $V_{yN}$  (right measurement vector in (2-16)). Both approaches are discussed in Chapter 5.

$$z = \begin{pmatrix} V_\delta \\ V_{xn} \\ V_{yn} \end{pmatrix}, \quad z = \begin{pmatrix} V_\delta \\ V_{xN} \\ V_{yN} \end{pmatrix} \quad (2-16)$$

## 2-4 Experimental Platform

### 2-4-1 Scaled Vehicle Architecture

For the sake of efficiency and simplicity, a 1 : 10 scaled vehicle model was developed that is capable of drifting. The vehicle was a modified exemplar of the Berkeley Autonomous Race Car (BARC), which is an autonomous vehicle development platform developed at the University of California, Berkeley [65]. The main mechanical hardware components of the vehicle include the chassis, suspension, motor, transmission system, and wheels. The chassis is suspended on a double wishbone structure with coil springs on four wheels with rubber tires. The front tires were standard while the rear tires were slick for better drifting. The vehicle is rear-wheel driven by a brushless motor with an electronic speed control (ESC) that sets the target rotation speed of the motor. The ESC also sends signals to the servo that sets the steering angle of the front wheels. A lithium polymer battery supplies power to the on-board electronic components. An aluminium frame was built on top of the chassis for the purpose of holding electronic hardware.

The frame was equipped with the sensor suite described in Section 2-2. To establish a connection between the sensors, actuators and the on-board processor, an Arduino Nano was placed on the frame together with an USB hub. All signals are sent to the central processing unit, an ODROID XU4 which runs Lubuntu 14. The platform uses the robot operating system (ROS) [66], an open-source library and set of tools for programming robots. The ROS infrastructure is build on the concepts of nodes, topics and messages. A node is a program that processes data. That data is then packaged into a *message* format and published onto a *topic* for other nodes to subscribe to. The above hardware- and software architecture is summarised in Figure 2-5. Technical specifications of the used hardware can be found in Appendix A.

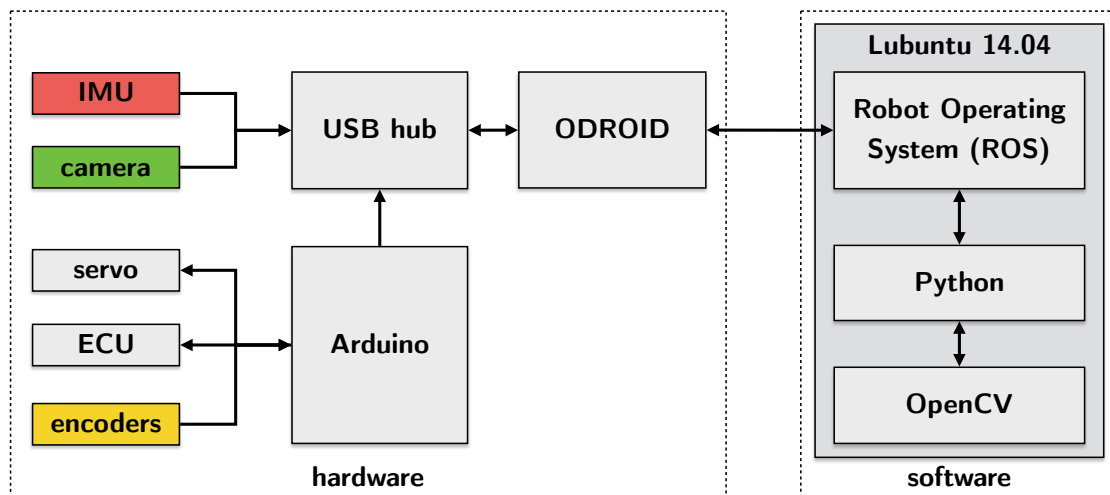


Figure 2-5: Vehicle Hard- and Software Architecture

### 2-4-2 Vehicle Parameter Estimation

Some vehicle parameters needed to be estimated to observe the states properly. The mass was weighed using a digital hanging scale with an accuracy of  $\pm 0.1$  kg. The total mass was estimated to be 2 kg and when weighing the vehicle with the front and rear axle suspended the mass distribution was estimated to be equal, which puts the *CG* in the middle of the vehicle. The dimensions were estimated using a vernier calliper with an accuracy of  $\pm 0.001$  m, with the following results:

- wheelbase length:  $L = 0.260$  m
- track width:  $T = 0.160$  m
- nominal tire diameter:  $R_{nom} = 0.072$  m

During the experiments, the steering angle  $\delta$  needed to be measured for the state observation as well. However, it could not be measured directly. What would be measured was the control signal that was sent to the servo. The servo (controlled system) was modelled from its pulse-width-modulation (pwm) input signal to the output  $\delta$ . Only the steady state response was modelled in this project for the sake of simplicity. During the experiments a constant steering angle was applied, so transient effects during the experiment could be neglected. From empirical measurements, the following linear function was acquired:

$$\delta = 0.7267U_S - 61.5189 \quad (2-17)$$

This is for the domain of  $\delta = \{-15, 15\}$  deg, outside of this domain the angle is constant.

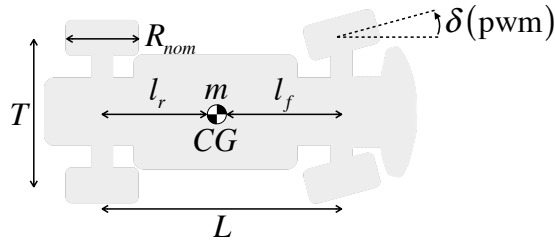
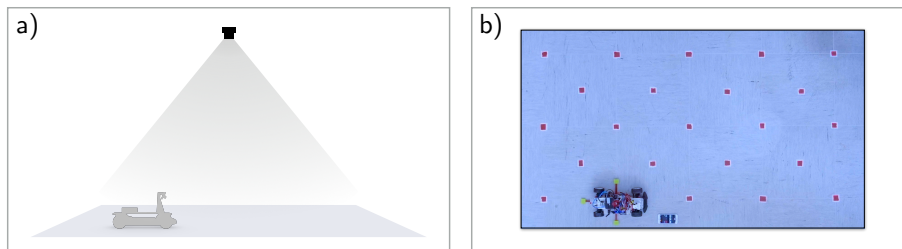


Figure 2-6: Geometric Vehicle Properties

### 2-4-3 Motion Capture System

A lab was available with a  $2 \times 2$  m low-friction surface that served as a test ground for drift experiments. The surface consists of smooth ceramic tiles so a friction coefficient of well below 0.5 should be expected, considering the slick tires of the vehicle. A motion capture system (MCS) was specifically designed for this thesis based on the computer vision software that would also be used on the vehicle. The MCS consisted of a high quality camera attached to the ceiling facing down on the test surface. The vehicle was equipped with green markers that the computer vision software would detect and track throughout the experiment, while the test surface contained red markers that the on-board camera could track for state estimation. Red is easily distinguishable from green by the computer vision functions [67] so the tracking of the vehicle by the MCS did not get disturbed by the red tracking markers for the on-board camera.



**Figure 2-7: Motion Capture System.** Subfigure (a) shows a schematic side-view of the setup: the ceiling holds the MCS camera 3 m above the test surface. Subfigure (b) shows the top view MCS camera image. The red markers were used to calibrate the homography. The green markers on the vehicle were used to track its position and heading. The phone on the floor shows a stopwatch which is also visible on the on-board camera image. This was used to align the MCS with the on-board data.

## 2-5 Conclusion

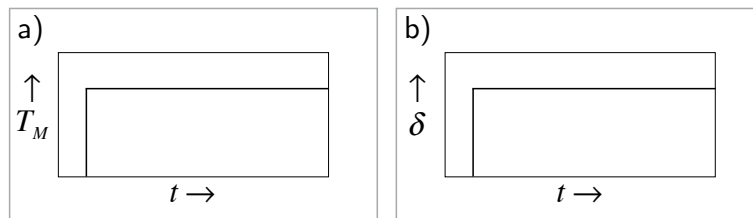
This chapter presented a state estimation methodology for the sideslip of a vehicle that exclusively utilises on-board sensors, namely: rotary encoders, an inertial measurement unit and a camera. The on-board sensor suite was laid out in Section 2-2 by explaining the basic working principles and relating the measured quantities of the sensors to the  $CG$ -states. These quantities were used in a Kalman filter by defining the process, inputs and outputs. The experimental setup was presented in Section 2-4 on which the state estimation technique was implemented and validated. The next chapter provides the benchmark data to which the methodology was validated.



# Motion Capture System

### 3-1 Introduction

The design of the on-board computer vision and sensor fusion algorithms required tuning of various parameters. A benchmark dataset was needed to tune the parameters (offline). Additional data was needed to verify the performance of the algorithm after it was tuned. Experiments were conducted using the experimental setup described in Section 2-4. The external camera acquired images during the experiments that served as an input to the motion capture system (MCS). This chapter explains how the raw data from the external camera was processed by the MCS to produce meaningful results that served as the benchmark and validation datasets. A more in-depth theoretical explanation of the working principle behind the computer vision functions is given in Chapter 4. Multiple experiments (with step- and ramp inputs at different time instances) were conducted, but a single experiment was selected to serve as the benchmark. In the chosen experiment the vehicle received the step inputs given in Figure 3-1, resulting in an unstable drift response. The processing of the external camera data is described step by step in the following sections, referring to the benchmark experiment as an example. The same steps apply to the processing of the datasets from the validation experiments.



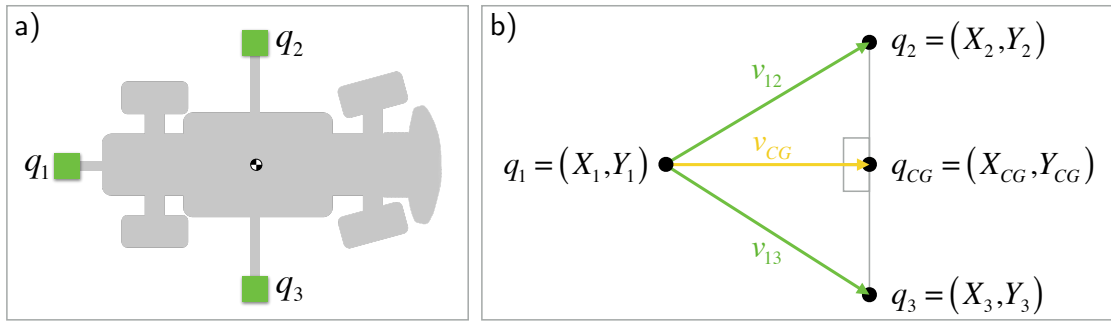
**Figure 3-1: Drift Experiment: Drive Inputs.** Subfigure (a) shows the motor torque step input to 100%, (b) shows the steering step input to 15 deg. The wheels are turned completely before the vehicle starts moving because the response of the motor is slower than the servo.

## 3-2 External Camera Data

### 3-2-1 Marker Tracking

The vehicle was tagged with three green markers (very well distinguishable from the red ones on the floor by computer vision functions [67]) to be tracked by the MCS. The reason to track three markers, is because this provides information on not only the location of the *CG*, but also the heading  $\psi$  of the vehicle with respect to the global coordinates  $X$  and  $Y$ . The MCS detects the markers in the initial frame by applying a green colour range threshold and detecting three corner points with a Shi-Tomasi edge detector. By specifying a minimum distance between the markers, a medium minimal quality and a maximum of 3 points, it was assured that the three markers were detected correctly. The computer vision function parameters of the MCS are given in Table B-1. Once detected, the markers are numbered clockwise as shown in Figure 3-2 (a). In the subsequent frames a Lukas-Kanade optical flow tracker tracked the detected points. The resulting output was a vector containing for each marker the image location  $p$  at every time sample throughout the experiment.

### 3-2-2 Position Data



**Figure 3-2: Tracking Markers** Subfigure (a) shows the numbering of the markers to distinguish them for the calculations in Subfigure (b), where the *CG* location is deduced.

The marker tracking stage outputs for each image frame the  $(u, v)$  image locations for  $q_1$ ,  $q_2$  and  $q_3$ . A linear mapping was established between the external camera image coordinates  $p$  in pixels and the globally defined coordinates  $q$  on the test surface to obtain the marker locations in meters. Elementary linear algebra was applied to describe the geometric relationship between the marker locations and the location and heading of the *CG*. Figure 3-2 (b) shows how this relationship was derived. Vectors  $v_{12}$  and  $v_{13}$  define the positions of  $q_2$  and  $q_3$  relative to  $q_1$  by (3-1) and (3-2) respectively. The vector between  $q_1$  and the *CG* location is derived as (3-3). Finally, the *CG* location was found via (3-4).

$$v_{12} = (x_2 - x_1, y_2 - y_1) \quad (3-1)$$

$$v_{13} = (x_3 - x_1, y_3 - y_1) \quad (3-2)$$



$$v_{CG} = \frac{1}{2}(v_{12} + v_{13}) \quad (3-3)$$

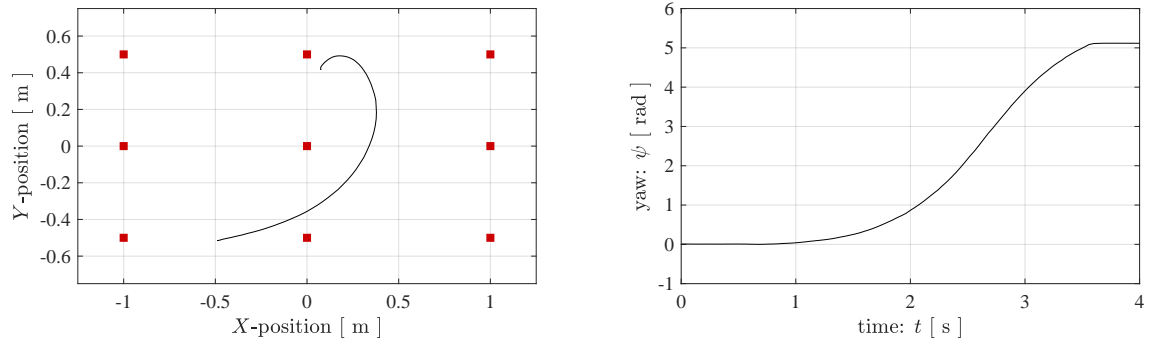
$$q_{CG} = q_1 + q_{CG} \quad (3-4)$$

Since the vector  $v_{CG}$  intrinsically inferred the heading of the vehicle  $x$ -axis (centreline), it served to deduce the vehicle yaw with respect to the global  $X$ -axis. The cross- and dot product with the unit vector of the global  $X$ -axis ( $v_X$ ) was calculated and substituted in an arctangent, which provided the yaw (3-5).

$$\psi = \tan^{-1} \left( \frac{v_{CG} \times v_X}{v_{CG} \cdot v_X} \right) \quad (3-5)$$

The above calculations were performed for each frame. After processing the images as described in Section 3-2-1, the following data was available for each sample  $k$  (given in Figure 3-3):

- timestamp at sample:  $t(k)$  [ ms ]
- $X$ -location of the  $CG$ :  $X_{CG}(k)$  [ m ]
- $Y$ -location of the  $CG$ :  $Y_{CG}(k)$  [ m ]
- heading relative to the  $X$ -axis (yaw):  $\psi_{CG}(k)$  [ rad ]



**Figure 3-3: Drift Experiment: Position Data.** The left figure shows the  $X$ - $Y$ -trajectory which results from plotting  $X_{CG}(k)$  against  $Y_{CG}(k)$  for each time sample  $k$ . The red squares are the locations of the homography calibration markers. The experiment started at the bottom left, and ended at the top middle. The steep change of the curve at the end is a good example of how a drift can result in a much narrower cornering radius, confirming the theory described in Section 1-2-3. The right figure shows the yaw (heading)  $\psi_{CG}(k)$ . For visualisation purposes, the domain of the yaw angle is defined as  $[0, 2\pi]$ . The vehicle came at an abrupt standstill at the end, which the curve demonstrates with the steep change at 3.5 s. The plots show the raw data, neither of the curves were smoothed.

### 3-3 Signal Processing

#### 3-3-1 Time Derivatives

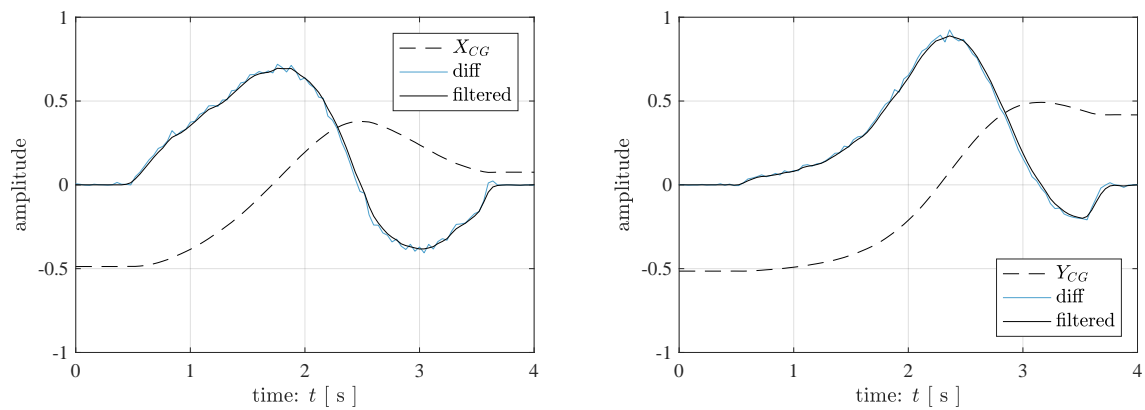
From the signals  $X_{CG}(k)$ ,  $Y_{CG}(k)$  and  $\psi_{CG}(k)$  combined with the information from  $t(k)$ , velocity- and accelerations were deduced by calculating the signal time-derivatives. A discrete-time derivative of a signal  $s$  as given in (3-6), where  $k$  is the sample and  $T_s$  is the sampling time, would not be the optimal solution to find the derivative because it amplifies noise in the measurements. This is especially important because the accelerations that were to be calculated required a second-order time derivate and also an inclusion of the yaw angular velocity which is itself is a derivative as well. Calculating this with the discrete method would result in propagating noise and may also amplify incorrect information in the signals.

$$V[k] = \frac{s[k] - s[k-1]}{T_s} \quad (3-6)$$

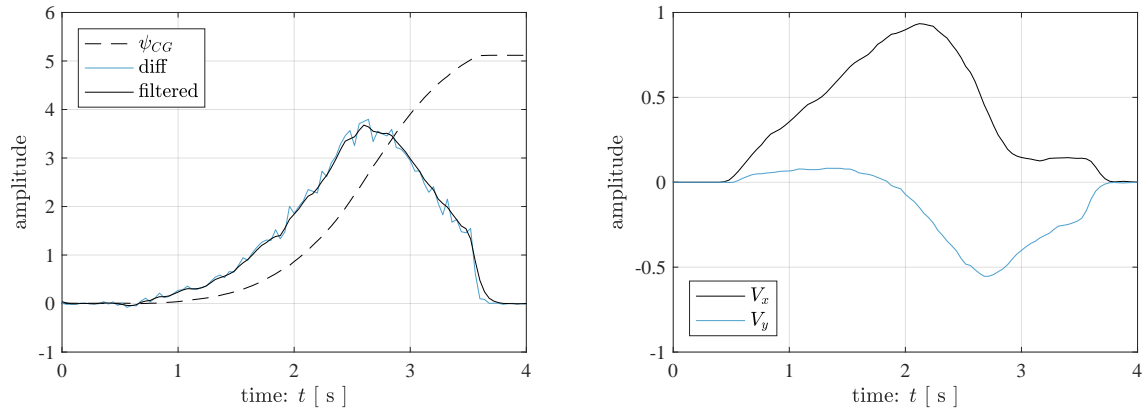
Since post-processing was applied, a linear a-causal approach was used to find the derivatives of the signals. By applying a differentiator with a low-pass filter, as given in (3-7), where  $s$  is the Laplace operator and  $\tau$  is the filter time constant, a result was obtained that contained less noise and only emphasised the frequencies of interest present in the signals.

$$F(s) = \frac{s}{(1 + \tau s)^2} \quad (3-7)$$

The value of  $\tau$  was a tuned parameter that remained the same when filtering each signal. It was found through trial- and error that  $\tau = 0.04$  s produced desirable results (Figure 3-4, and Figure 3-5). The CVA data was captured at a sampling rate of 30 Hz. The filter has two poles at 4 Hz, well under the Nyquist frequency of the signals of  $\frac{1}{2}f_s = 15$  Hz, thus satisfying the criterion. The filter was discretised using the Tustin method.



**Figure 3-4: Drift Experiment: Absolute X- and Y-Velocity Derivation.** The global  $(X, Y)$ -position of the  $CG$  is shown as the dotted lines. The blue lines are the discrete derivatives as defined in (3-6), the solid black lines are the filtered derivatives (3-7).

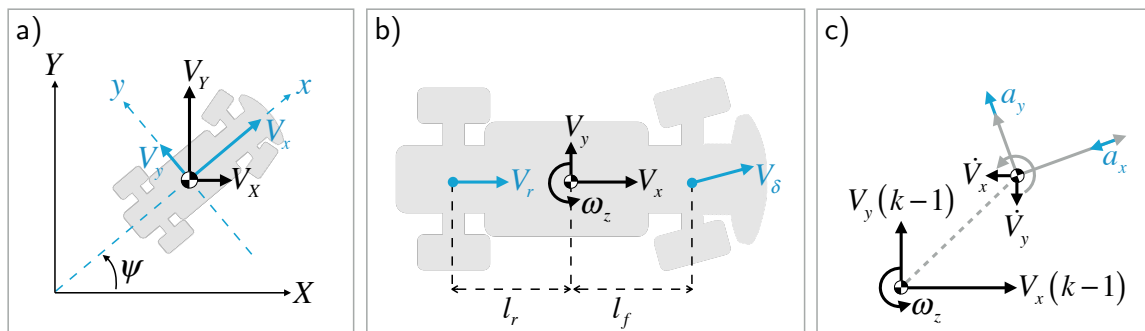


**Figure 3-5: Drift Experiment: Derived  $CG$ -States.** The yaw  $\psi_{CG}(k)$  of the vehicle is shown as the dotted line in the left plot. The blue line is the discrete derivative as defined in (3-6), the solid black line is the filtered derivative (3-7). The black line in the right plot shows the longitudinal velocity derived from (3-8). The blue line shows the lateral velocity from (3-9).

It can be observed in the left plot of Figure 3-5 that the yaw signal becomes noisier at high velocities compared to the position signals. The steep change at 3.5 s occurred because the motor stopped driving the wheels and the longitudinal slip dropped significantly. This increased the lateral friction resulting in a sudden drop in yaw angular velocity, confirming the theory in Section 1-2. What can be seen in the right plot on the other hand, is that the low friction at the rear wheels resulted in a positive lateral velocity at the beginning of the experiment.

### 3-3-2 State Transformations

Figure 3-4 shows the absolute velocities of the  $CG$  on the globally defined coordinates. From a vehicle dynamics point of view it was interesting to know the velocity components relative to the vehicle  $x$ - $y$ -coordinate system, as depicted in Figure 3-6 (a). By applying the state transformations (3-8) and (3-9) the relative components of the  $CG$  velocity were obtained, of which the result is shown in the right plot of Figure 3-5.



**Figure 3-6: State Transformation Scheme.** Subfigure (a) shows how the absolute velocities in terms global coordinates are transformed to the relative velocities using the yaw. The obtained relative velocities were used to obtain the axle velocities (b) and relative  $CG$  accelerations (c).

$$V_x = V_X \cos(\psi) + V_Y \sin(\psi) \quad (3-8)$$

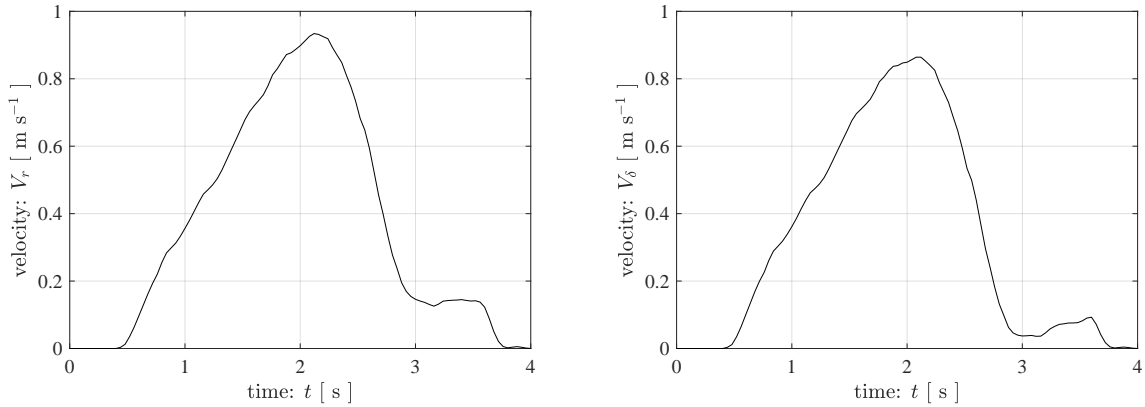
$$V_y = -V_X \sin(\psi) + V_Y \cos(\psi) \quad (3-9)$$

With  $V_x$  and  $V_y$  at the  $CG$  known, the velocities the the axles were deduced. These values were interesting since they were needed to verify the measurements of the wheel encoders. See Figure 3-6 (b). Naturally, the rear-axle velocity  $V_r$  is equal to  $V_x$  since a rigid body was assumed. The front axle velocity (in the steering direction  $\delta$ , since the wheel velocity was measured)  $V_\delta$  was obtained through (3-12), substituting (3-12) and (3-12). The results are shown in Figure 3-7.

$$V_{x,f} = V_x \quad (3-10)$$

$$V_{y,f} = V_y - l_f \omega_z \quad (3-11)$$

$$V_\delta = \sqrt{V_{x,f}^2 + V_{y,f}^2} \cos \left( \tan^{-1} \left( \frac{V_{y,f}}{V_{x,f}} \right) - \delta \right) \quad (3-12)$$



**Figure 3-7: Drift Experiment: Axle Velocities.** The left plot shows the rear axle velocity  $V_r$ . The right plot shows the front axle velocity in the steering direction  $V_\delta$  from (3-12).

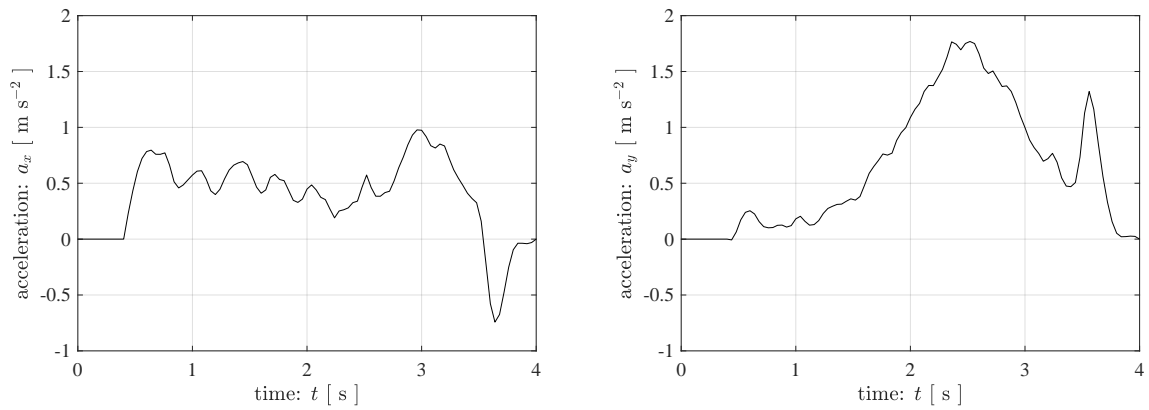
The final important states that were deduced from the obtained velocities are the linear accelerations at the  $CG$ . These were interesting because they were needed to verify the IMU measurements. The first step was to differentiate the  $CG$  velocities over time with the filter from (3-7), yielding  $\dot{V}$ . The influence of  $\omega_z$ , which is a rotation of the relative coordinate system to which the acceleration is measured, was taking into account as well:

$$a_x = \dot{V}_x - \omega_z V_y \quad (3-13)$$

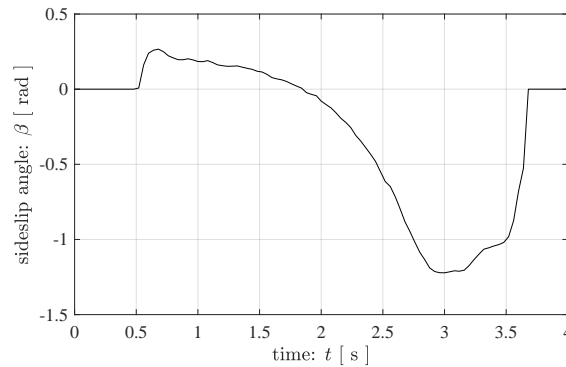
$$a_y = \dot{V}_y + \omega_z V_x \quad (3-14)$$

The results of (3-13) and (3-14) are shown in Figure 3-8. The sideslip angle was calculated as well to serve as the benchmark for the output of the estimation process, By applying (3-15) to the signals shown in the right plot of Figure 3-5, the result in Figure 3-9 was obtained.

$$\beta = \tan^{-1} \left( \frac{V_y}{V_x} \right) \quad (3-15)$$



**Figure 3-8: Drift Experiment: Relative CG-Accelerations.** The left plot shows the relative longitudinal acceleration derived from (3-13). The right plot shows the relative lateral acceleration derived from (3-14). The sudden stop of the motor that lead to a standstill can be clearly seen. The accelerations did not go completely to zero due the fact that the external CVA was constantly tracking the points and even the slightest (observed, not actual) difference in marker positions would lead to a small value for the acceleration.



**Figure 3-9: Drift Experiment: Sideslip Angle.** The curve followed from (3-15). The data was very useful in the sense that both a positive and negative sideslip angle was obtained, as well as a short sustain of a high amplitude at 3 s.

## 3-4 Datasets

### 3-4-1 Benchmark

The most important state in this project was the sideslip angle, which needed to be estimated with the on-board sensors. The drift dataset was chosen therefore to be the benchmark for the sensor fusion design. From the drift dataset followed the curve in Figure 3-9. The data was consistent with what could be interpreted visually in the video from the external camera. The angle was both negative and positive, which ensured no mistakes were made with signs. The short sustain between 3 and 3.5 s was also a good test for the algorithm because here the IMU accuracy would not be sufficient. Finally, the fact that the experiment ended at a standstill was also a good way to verify if the algorithm could correct itself to come back to zero. In the following chapters the accuracy of the on-board signals and estimates is established by comparing them to the signals presented in this chapter. Referring to the signals from the MCS as  $s$ , and the measured/estimated as  $\hat{s}$  the variance accounted for (VAF) in (3-18) was established for each signal, where  $V(s)$  is the signal variance (3-17) and  $\mu(s)$  is the signal mean (3-16).

$$\mu(s) = \frac{1}{N} \sum_{k=1}^K s_k \quad (3-16)$$

$$V(s) = \frac{1}{K-1} \sum_{k=1}^K |s_k - \mu|^2 \quad (3-17)$$

$$VAF(\hat{s}) = \frac{V(s - \hat{s})}{V(s)} \quad (3-18)$$

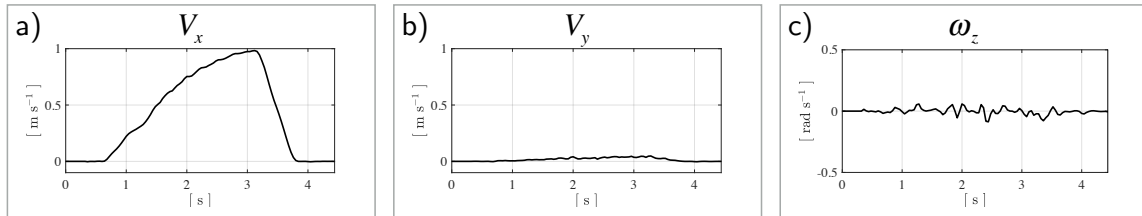
### 3-4-2 Validation

It is not desirable to tune an algorithm on a single dataset because this could lead to overfitting and does not ensure robustness. To evaluate the algorithm performance, it was tested against two other validation datasets which are extreme and opposite scenarios:

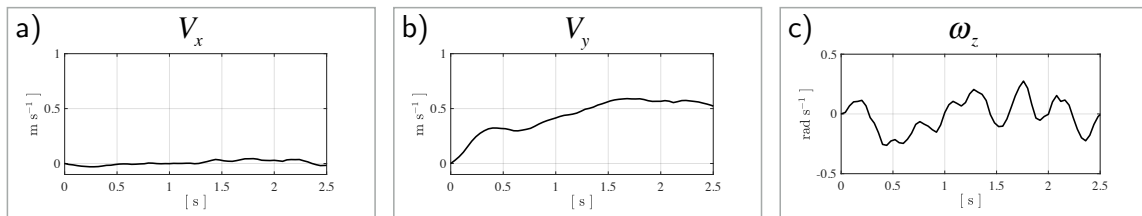
- pure longitudinal motion:  $V_x > 0, V_y \approx 0$  (Figure 3-10)
- pure lateral motion (manually induced)  $V_x \approx 0, V_y > 0$  (Figure 3-11)

In the first dataset the vehicle has a minimum sideslip ( $\approx 0$  deg). This was achieved by sending the motor a moderate constant input with zero steering angle. Maximum sideslip was achieved in the second dataset by sending zero inputs, and manually moving the vehicle sideways, resulting in maximum sideslip ( $\approx 90$  deg).

The acquisition and processing of the data from the experiments mentioned above was the same as was described in Sections 3-2 and 3-3. The pure lateral experiment was carried out manually, hence the chattery yaw angular rate (see Figure 3-11 (a)).



**Figure 3-10: Validation Dataset: Longitudinal Motion**



**Figure 3-11: Validation Dataset: Lateral Motion**

## 3-5 Conclusion

This chapter explained the data processing for the MCS, and how its output served as benchmark and validation datasets for the state estimation design process. The benchmark dataset was an unstable drift resulting from the step inputs given in Figure 3-1. Each step of the MCS data processing was described referring to the benchmark experiment as an example. In Section 3-2 a method was presented to extract the location and heading of the vehicle throughout the experiment. Section 3-3-1 explained how these signals were differentiated over time using a differentiator with low-pass filter. Section 3-3-2 presented a technique by which the obtained states were transformed to the desired outputs that were needed for benchmarking and validation. Finally, Section 3-4-1 discussed the outcome of the experiments.





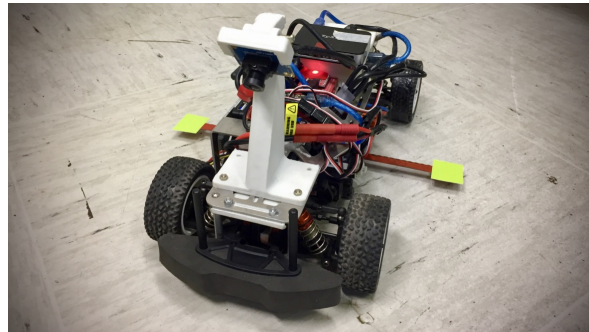
# Computer Vision Algorithm

## 4-1 Introduction

In Section 2-2-3 the function of the camera within the state estimation methodology has been described. The camera served as one of the sensors to provide a measurement of the lateral- and longitudinal velocity components of points on the surface in front of the vehicle, to deduce the *CG*-states. Figure 4-1 shows how the camera was mounted on the vehicle. This chapter presents the design of a computer vision algorithm (CVA) that can run on board the vehicle. The CVA translates the camera images into *CG*-velocity measurements. The benchmark dataset presented in Section 3-4-1 will be the reference for tuning the CVA parameters. The CVA consists of the following stages:

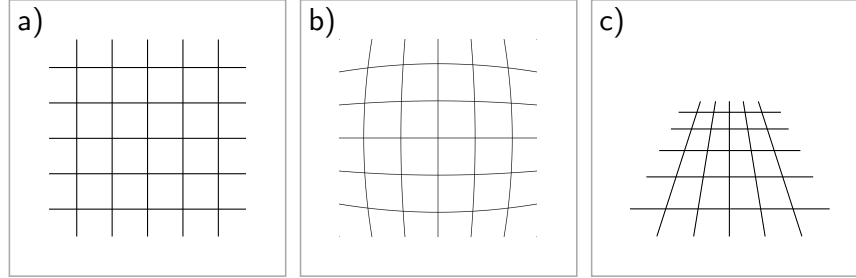
1. image acquisition and correction
2. feature detection and tracking
3. projection and egomotion estimation

The above stages are described in Section 4-3. The acquisition of the camera calibration parameters is explained in Section 4-2.



**Figure 4-1: On-board camera placement.** The white part is a 3D-printed mount to place the camera high above the ground, facing downward. The material is stiff enough to prevent relative motion of the camera thus ensuring a fixed geometry.

## 4-2 On-Board Camera Calibration



**Figure 4-2: Image Aberration.** The on-board camera used in this project suffered heavily from image aberration. Consider the homogeneous image grid in Subfigure (a). Subfigure (b) shows what barrel distortion does to the grid, (c) shows the effect of perspective.

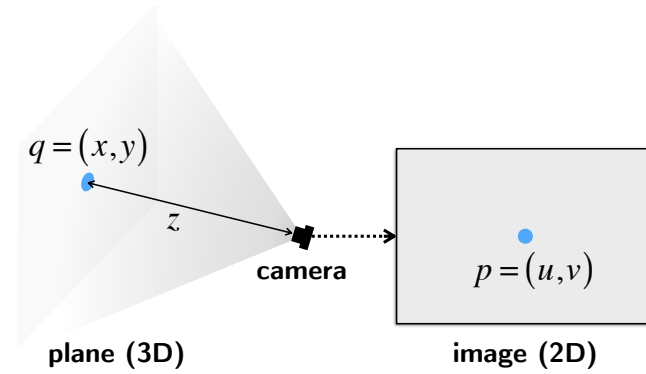
### 4-2-1 Distortion

In a theoretical ideal optical system, all rays of light coming from a point in the observed plane would converge to the same point in the image plane, forming a clear image. The influences which cause different rays to converge to different points are called aberrations. Examples of aberrations are field curvature, chromatic aberration and distortion. Barrel distortion, as visualised in Figure 4-2 b), is an important aberration to account for with small and cheap cameras like the one used on the vehicle in this project. Due to the distortion, straight lines will appear curved. Its effect is increases further away from the image centre. The distortion is constant and can be corrected for by (4-1). The variable  $r$  is given by (4-2), where  $(u_d, v_d)$  are the distorted image points,  $(u_u, v_u)$  are the undistorted image points and  $(u_c, v_c)$  is the image centre.

$$\begin{aligned} u_{corr} &= u_d \left( 1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \dots \right) \\ v_{corr} &= v_d \left( 1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \dots \right) \end{aligned} \quad (4-1)$$

$$r = \sqrt{(u_u - u_c)^2 + (v_u - v_c)^2} \quad (4-2)$$

To find the radial distortion coefficients ( $K_1, K_2$  etc.), calibration experiments were conducted where multiple images of chessboards were taken with the camera. The chessboard squares have known dimensions which allowed the algorithm to find the mathematical relationship between the image points and real-world points. The above technique is based on the work by Zhang [68].



**Figure 4-3: Planar Projection.** The camera observes a point  $q$  in three-dimensional (3D) space on a flat surface (plane) with defined  $(x, y)$  coordinates in [ m ], at a depth  $z$  from the camera. The image formed on the sensor is two-dimensional (2D) and the corresponding point  $p$  has  $(u, v)$  coordinates in [ px ].

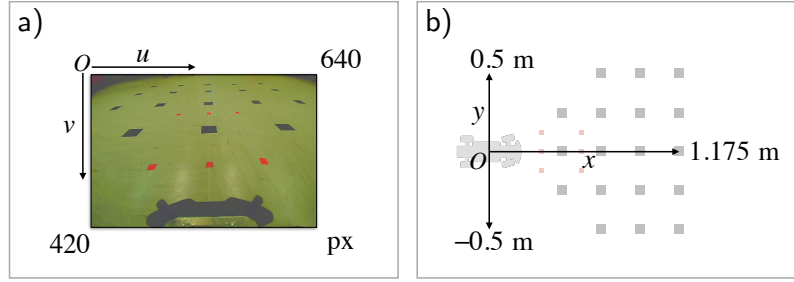
### 4-2-2 Homography

The points tracked by the algorithm are pixel locations in a digital image array, while the desired outputs are to be in  $\text{m s}^{-1}$  on the actual observed plane. A translation from pixel locations to the corresponding points on the observed plane was needed, which is referred to as a planar *homography*, a projection from one plane to another [69]. This projection simultaneously accounts for that fact that the on-board camera is also looking at the ground surface under an angle and from a certain distance which leads to perspective in the image [70], visualised in Figure 4-2 c). The homographic projection determines the relationship between an image coordinate  $p = (u, v)$  in pixels [ px ] and its corresponding point on the observed plane  $q = (x, y)$  in meters [ m ] (Figure 4-3). The following assumptions were made regarding the projection:

- only the plane to be observed was visible in the image
- the plane was level at all of the observed points
- the camera had a constant position relative to the plane

Under the assumptions given above, it was deduced that the transformation between the image and observed plane points is projective linear. Homographic transformation makes use of homogeneous Cartesian coordinates. Every point  $q$  on the plane is observed by the camera from a certain depth,  $z$ . However, the formed image is two-dimensional without depth. To project the two-dimensional image points onto a three-dimensional plane relative to the camera, the image coordinates were appended with a depth of 1, which is a relative parameter and thus will not affect the value of the non-homogeneous coordinates, which yields:

$$\tilde{p} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{pmatrix} \quad \tilde{q} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} \quad (4-3)$$



**Figure 4-4: On-Board Homography Estimation.** Subfigure (a) shows the calibration image. Additional red markers were placed near the camera, to increase the accuracy of the estimation in this critical area where more pixels were available per meter. Subfigure (b) shows how the coordinate system was defined, the origin was assumed to be the location of the IMU.

The projection was achieved with the dot product of the homographic transformation matrix  $H$  and the homogeneous image point  $\tilde{p}$ , obtaining the homogeneous point in the observed plane  $\tilde{q}$ :

$$\tilde{q} = H\tilde{p} \quad (4-4)$$

The homogeneous point in three-dimensional space can be transformed to its corresponding non-homogeneous planar two-dimensional coordinate through:

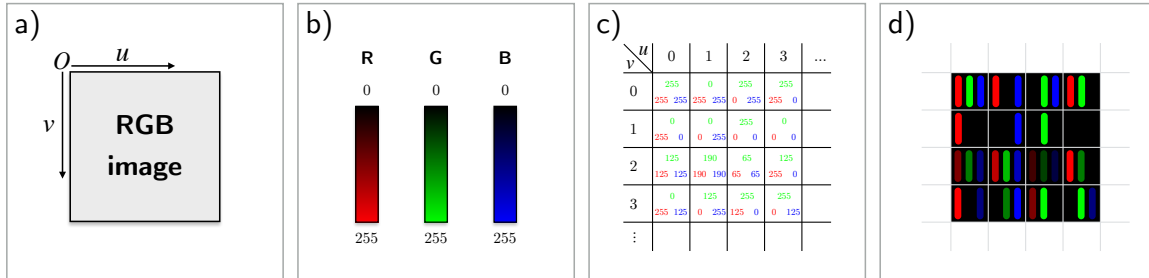
$$q = \begin{pmatrix} \tilde{x}/\tilde{z} \\ \tilde{y}/\tilde{z} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \quad (4-5)$$

The homographic transformation matrix  $H$  in (4-4) was estimated in an experiment. The camera was placed in the position in which it eventually would operate. A Cartesian coordinate system was defined for the to be observed plane with the origin at the  $CG$ , the  $x$ - and  $y$ -axis parallel to the vehicle axes (Figure 4-4 (b)). Markers were placed on the plane of which the  $(x, y)$  coordinates were determined. Since the markers were visible in the calibration picture taken by the camera, the respective  $(u, v)$  pixel locations could be derived from it. The experiment provided the following two arrays:

$$P = \begin{bmatrix} u_1 & \dots & u_N \\ v_1 & \dots & v_N \end{bmatrix} \quad Q = \begin{bmatrix} x_1 & \dots & x_N \\ y_1 & \dots & y_N \end{bmatrix} \quad (4-6)$$

The previously described methods were derived from the work from Corke (2011) [71], who developed a machine vision toolbox for Matlab. A particular function within this toolbox was used to find the homography matrix, using the arrays from (4-6) as inputs.

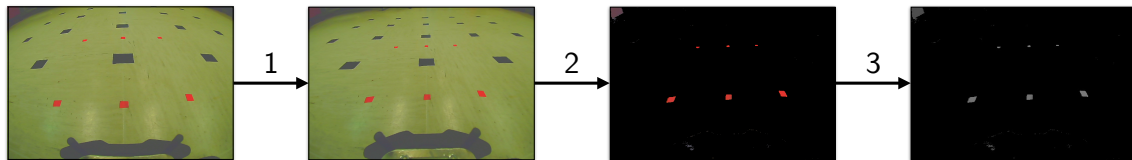
## 4-3 Algorithm Stages



**Figure 4-5: RGB-Colour Image.** A digital RGB image is a 3 channel array of  $u \times v$  pixels, with the origin at the top left corner (a). Each colour channel contains intensity an intensity value from 0 to 255 (b) for each element in the array (c). The resulting example image is shown in (d).

### 4-3-1 Image Processing

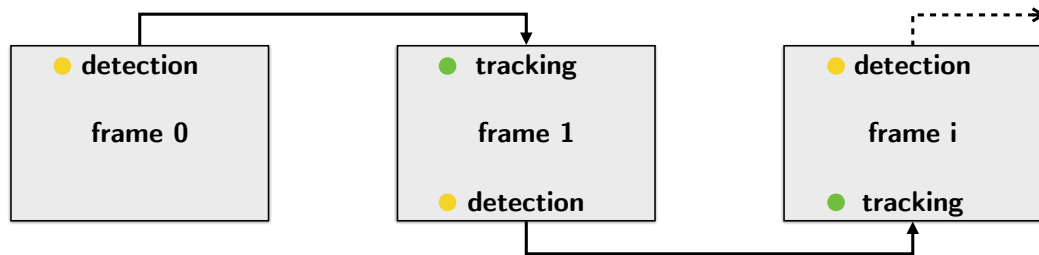
The raw images were acquired as digital RGB-colour images, see Figure 4-5. After acquisition, the images were processed in 3 steps, as shown in Figure 4-6. In the first stage the distortion was removed. A small part of the vehicle (the front bumper) was visible in the picture, while in the back of the image the parts of the environment could be seen that were not part of the ground surface. A region of interest (ROI) was defined so that these parts of the image were omitted. Red pixels within the ROI were extracted in step 2. The 3 image channels of an RGB image contain colour intensities for red, green and blue (RGB). A range of values for each channel have been defined. The filter let through all values within the range, while all values outside of the range were returned as zeros (resulting in black pixels). The final step in the image processing stage was the conversion from RGB-colour to greyscale. This was mainly for programming reasons, because the computer vision function that was used could only process greyscale images. Incidental to this approach though, was that it is in general more efficient to detect features in a greyscale image, since only a single channel needs to be investigated (thus resulting in at least three times less effort compared to detection in a RGB-colour image). The colour intensities in each channel were combined by taking the average value of the three.



**Figure 4-6: Image Processing Steps.** Step 1: correcting for the distortion by using the calibration parameters. Step 2: Applying red colour threshold. Step 3: Greyscale conversion.

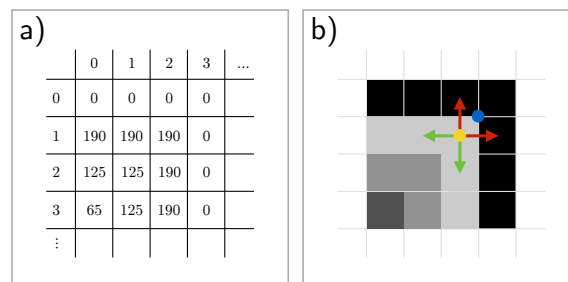
### 4-3-2 Optical Flow Estimation

The main goal of the algorithm was to extract motion data from the images. Motion of pixels throughout video frames is referred to as *optical flow*. To estimate the optical flow, features needed to be detected in an initial frame, and tracked into the next. These steps were performed simultaneously, as shown in Figure 4-7.



**Figure 4-7: Detection-Tracking Cycle.** The features that the camera tracks may disappear from the image. Therefore, the detected points were tracked into the next frame, wherein a new set of points was detected for tracking into the next frame and so on.

The detection was performed with the widely used Shi-Tomasi corner detection method [72]. The method was implemented as a `cv` function called: `goodFeaturesToTrack` from [73]. The function finds the most prominent corners in the image or in the specified image region, see Figure 4-8 for the definition of a ‘corner’. It then calculates the corner quality measure at every source image pixel using either a specified minimum eigenvalue  $\lambda$  for each corner or a Harris combined corner- and edge detector [74]. The corners not adhering to a specified minimum were rejected. The remaining corners are sorted by the quality measure in descending order. The function also rejects each corner for which there is a stronger corner at a distance less than a predefined minimum distance.



**Figure 4-8: Corner Definition.** A corner in a greyscale image is defined as a pixel location where the gradient in both the  $u$ - and  $v$ -direction is large relative to the surrounding pixels. Consider the image array given in (a). Subfigure (b) shows the resulting pixel colours. The blue dot is what could be perceived as a ‘corner’. The yellow dot is what the algorithm defines as the corner location in pixels (in this case (2, 1)). The red arrows identify in which direction the gradient is high, while the green arrows show the direction where the gradient is low.

Once the features were detected in an initial frame, they were tracked along the next frames until the detection was refreshed. The optical flow calculation presented in this section works on several assumptions:

- The pixel intensities of an object did not change between consecutive frames.
- Neighbouring pixels had similar motion.

Consider a pixel  $I(u, v, t)$  in the initial frame. The pixel moved by distance  $(du, dv)$  into the consecutive frame in  $dt$  time. Since the pixels were the same and their intensity did not change, it can be stated that:

$$I(u, v, t) = I(u + du, v + dv, t + dt) \quad (4-7)$$

The Taylor-series approximation of the right side of this expression with the removal of common terms and divided by  $dt$ , yields the following equation:

$$\frac{\partial I}{\partial u} \frac{\partial u}{\partial t} + \frac{\partial I}{\partial v} \frac{\partial v}{\partial t} + \frac{\partial I}{\partial t} = 0 \quad (4-8)$$

The above equation is called the optical flow equation. The terms in (4-9) are the image gradients. Similarly, the term in (4-10) is the gradient along time. The terms in (4-11) are the unknowns.

$$I_u = \frac{\partial I}{\partial u}, \quad I_v = \frac{\partial I}{\partial v} \quad (4-9)$$

$$I_t = \frac{\partial I}{\partial t} \quad (4-10)$$

$$\frac{\partial u}{\partial t}, \quad \frac{\partial v}{\partial t} \quad (4-11)$$

An equation with two unknowns is difficult to solve. A method to solve this problem was developed by Lucas and Kanade (1981) [75]. It was assumed that all the neighbouring pixels will have similar motion. The Lucas-Kanade method takes a  $3 \times 3$  patch around each found corner point  $n$ . The resulting 9 points are assumed to have the same motion, and thus  $(f_x, f_y, f_t)$  can be found for these 9 points. A solution is obtained with a least-squares fit, given below:

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \begin{bmatrix} \sum_n I_{u_n}^2 & \sum_n I_{u_n} I_{v_n} \\ \sum_n I_{u_n} I_{v_n} & \sum_n I_{v_n}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_n I_{u_n} I_{t_n} \\ -\sum_n I_{v_n} I_{t_n} \end{bmatrix} \quad (4-12)$$

### 4-3-3 Egomotion Projection

The optical flow estimation process described in the previous subsection outputs point locations  $p_n(u, v)$  in the current frame and their location in the previous frame. The locations were projected to ground point locations  $q_n(x, y)$  using the obtained homography matrix  $H$  by (4-4). The displacement  $(dx_n, dy_n)$  and the elapsed time  $dt$  between the frames resulted in planar velocity components for a total amount of  $N$  tracked points:

$$\left\{ \frac{dx_n}{dt}, \frac{dy_n}{dt} \right\}_N \quad (4-13)$$

Using the current location  $q_n(x, y)$  as well as the displacement (4-13) of each point, the egomotion was calculated accordingly, as was described in 2-2-3:

$$V_{xn} = - \left( \frac{dx_n}{dt} - \omega_z y_n \right) \quad (4-14)$$

$$V_{yn} = - \left( \frac{dy_n}{dt} + \omega_z x_n \right) \quad (4-15)$$

The above measurements were individual estimates of the  $CG$ -velocities and are referred to as the ‘raw’ CVA measurements. They were processed in two different ways in the sensor fusion algorithm. In the first approach they were processed individually. In the second approach the measurements were ‘combined’ by taking the median over each time sample as follows:

$$V_{xN} = \widetilde{\{V_{xn}\}_N} \quad (4-16)$$

$$V_{yN} = \widetilde{\{V_{yn}\}_N} \quad (4-17)$$

The resulting signals were filtered with the same low-pass filter as the benchmark data (given in (3-7), without the differentiator, where  $\tau = 0.04$  s):

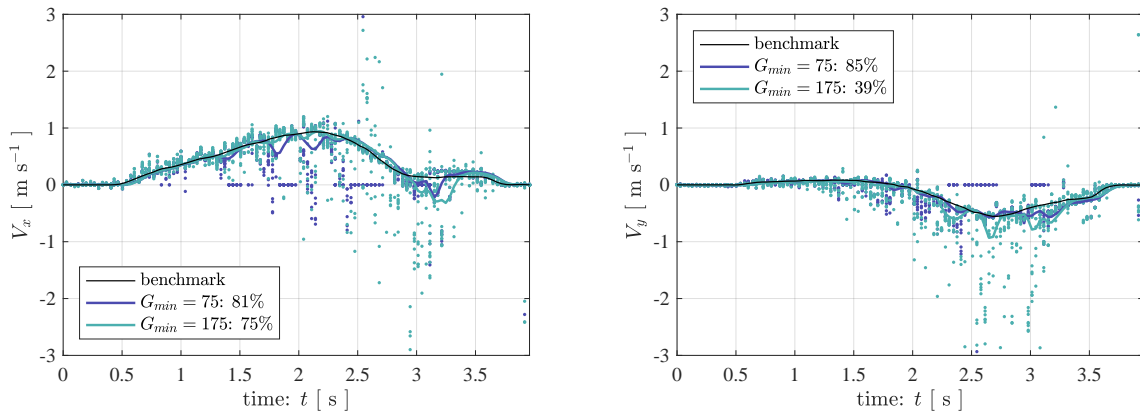
$$F(s) = \frac{1}{(1 + \tau s)^2} \quad (4-18)$$



## 4-4 Parameter Tuning

### 4-4-1 Default Parameters

The CVA contained various parameters that needed to be tuned. For each parameter a default value was established before a sensitivity analysis was performed. The surface area in front of the vehicle which was being observed with sufficient accuracy was roughly  $1 \times 1$  m. The area within the image could only contain as much as 4 markers. Each marker had 4 corners, so in an ideal case 16 points could be detected which was the default value for  $N_p$  (maximum number of points). The value of  $D_{min}$  (minimum distance between the points) would be greater than 0 in an ideal case, to account for the distance between the corners of each marker. However, the markers located further away were observed at a lower pixel density which meant that the pixel distance between corners differed throughout the image which made this value difficult to determine. Putting it at zero did not affect the detection of the best corners within the available pixels, so this was chosen as the default value.  $Q_{min}$  (minimum quality of detected points) should be close to zero for the default value because increasing it may result in the loss of detected points. With increasing velocity, the quality of corners decreased. At high velocities the image became vague, and this decreased the probability of a detection. A  $Q_{min}$  larger than 0 as a default value did not make sense, because first the colour threshold needed to be established before quality became an issue. With the default detection parameters established, the sensitivity analysis as well as the tuning process of the colour threshold could begin.



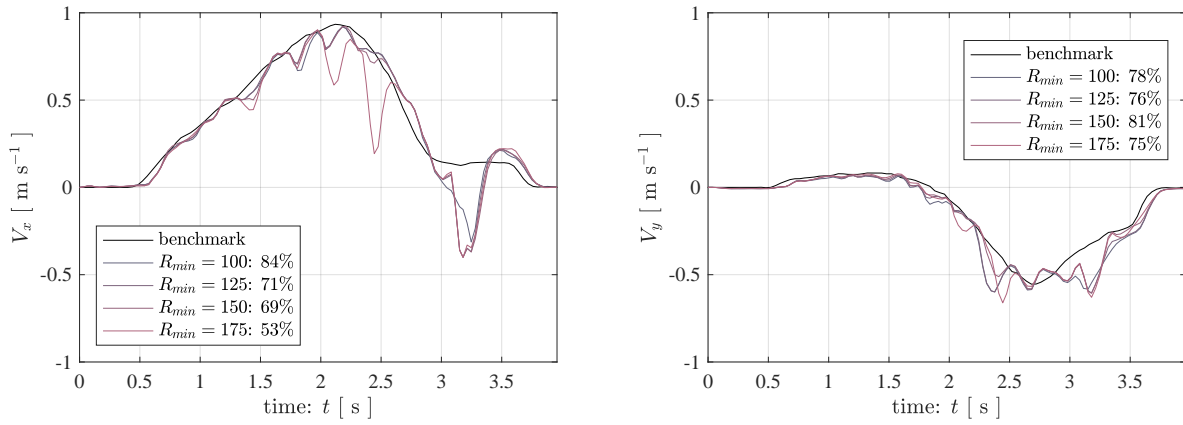
**Figure 4-9: Green Colour Threshold.** This figure shows the variation of the green colour threshold maximum  $G_{max}$ , and how it negatively affected the accuracy when it was not suppressed. It shows an increase in noise (more extreme outliers) for a higher value of  $G_{max}$ .

### 4-4-2 Sensitivity Analysis

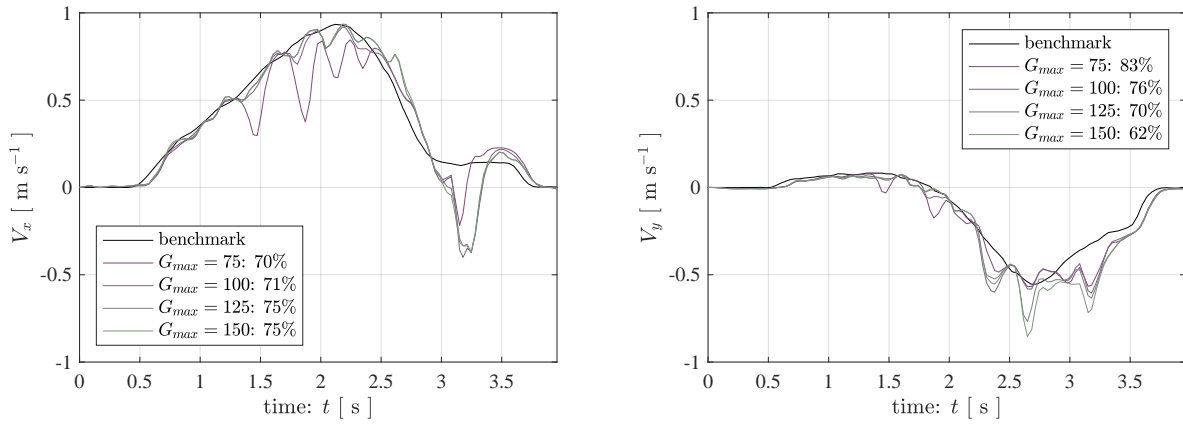
The colour of the markers is red. The red pixels needed to be extracted from the image so the red channel was prioritised. Prioritising means restricting a minimum with no maximum in this case. The other channels of green and blue were suppressed, meaning restricting a maximum with no minimum. This put  $R_{max}$  at 255,  $G_{min}$  and  $B_{min}$  at 0. Now the tuning of  $R_{min}$ ,  $G_{max}$  and  $B_{max}$  remained. If a too strict threshold was applied, e.g. with  $R_{min}$  at 255,  $G_{max}$  and  $B_{max}$  at 0, the algorithm did not find any points because such bright red pixels were not present in the image. So the optimum lies somewhere in between 255 and 0 for these values. Green is the ‘inverse’ colour of red, so its maximum was restricted more than blue. During the second part of the experiment, the camera looked in the direction of the windows and the reflection of the beautiful blue Italian sky was seen on the floor which ‘interfered’ with the red marker colour (making them seem more blue). Therefore blue had a priority over green. Also, the floor contained a lot of green which should be suppressed anyway, since only the trackers should be extracted. The results of the variation of these parameters is given on page 47. Additionally, Figure 4-9 shows why the green colour should be suppressed. This figure also illustrates the raw data-points that resulted in the curves after averaging and filtering. The following colour threshold parameters were selected from the analysis:  $R \in \{75, 255\}$ ,  $G \in \{0, 75\}$ ,  $B \in \{0, 175\}$ . The detection parameters were varied to establish the sensitivity of each. These were strongly related parameters because increasing the number of available points only had an influence if more points of the same quality could be detected, while the minimum distance again prohibited these points to be detected in the first place. The results of the analysis are shown on page 48. The tracking parameters were evaluated for accuracy but no difference was found by changing the parameters. The difference is expected to be in computational load but this was difficult to establish, also because this algorithm was not evaluated on the platform it would eventually operate on. This could be the aim of future studies. The values were nonetheless chosen such that they were expected to result in the least computational load.

## 4-5 Conclusion

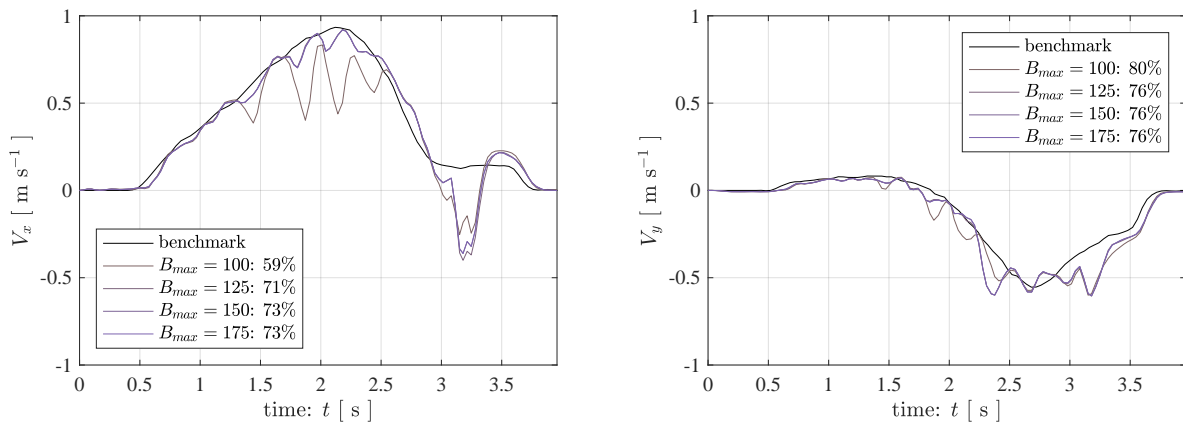
This chapter described the design of a computer vision algorithm that used the on-board camera to measure the longitudinal- and lateral velocity. The image was processed by correcting for distortion and isolating the red coloured markers in an ROI. After applying a greyscale conversion to the colour channels, the image was ready for the detection- and tracking stage. Corners of the markers were detected in the greyscale image by a Shi-Tomasi detector. A Lukas-Kanade optical flow function tracked the points through consecutive frames. Each tracked point location was obtained in pixels, so a conversion to actual coordinates in meters was needed for the final measurement output of the algorithm. This was achieved through homographic projection. The benefit of this approach is that it simultaneously removed perspective from the image.



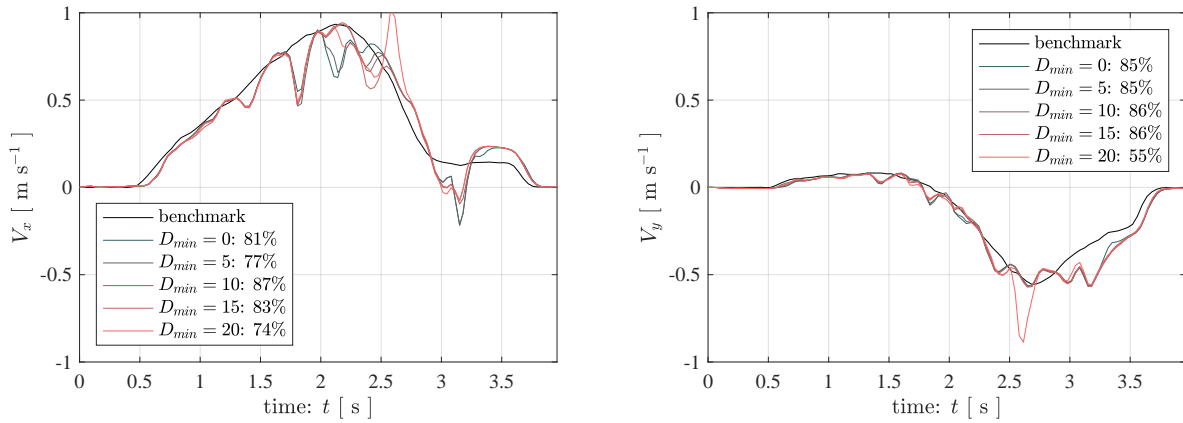
**Figure 4-10: Red Colour Threshold** The tuned parameter is the red colour threshold minimum  $R_{min}$ . The optimum was found to lie between 75 and 125, depending on the other parameters.



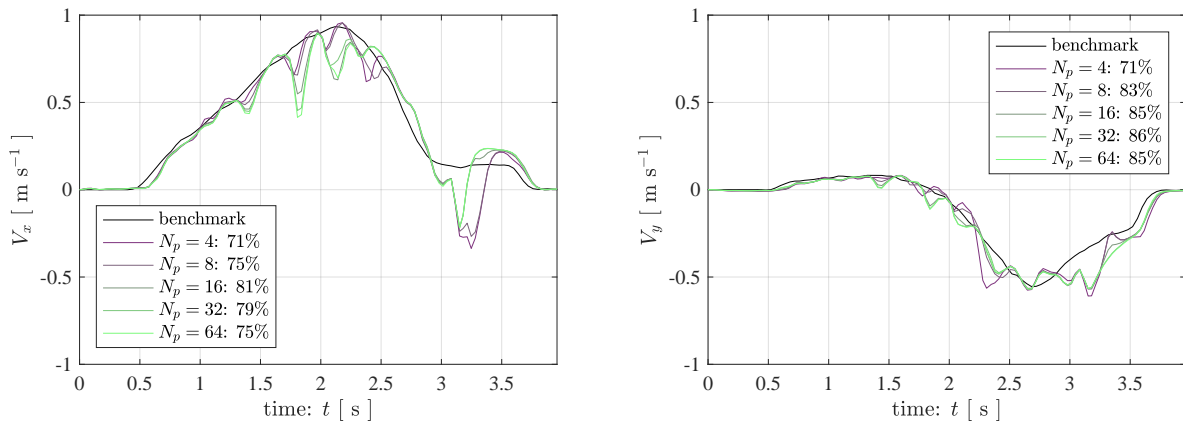
**Figure 4-11: Green Colour Threshold** The tuned parameter is the green colour threshold maximum  $G_{max}$ . The optimum was found to lie between 50 and 100, depending on the other parameters.



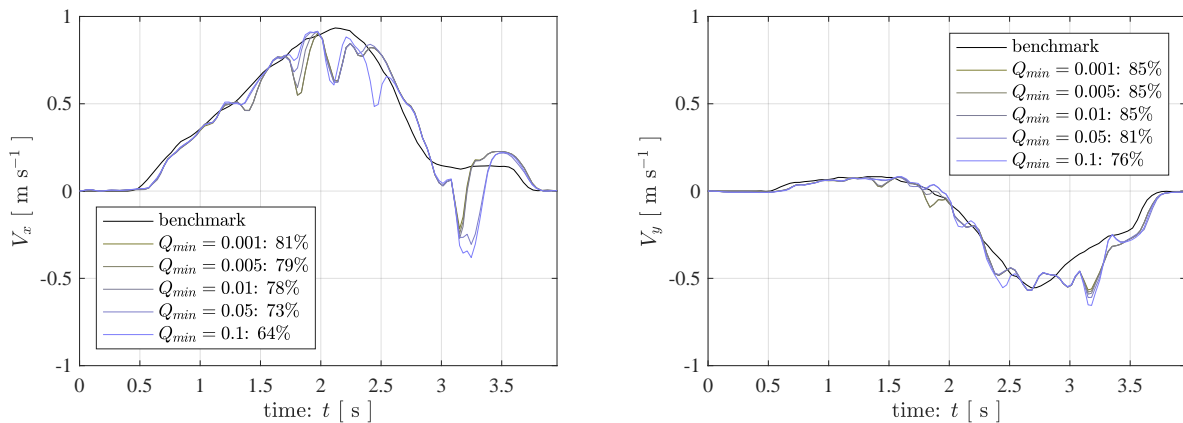
**Figure 4-12: Blue Colour Threshold** The tuned parameter is the blue colour threshold maximum  $B_{max}$ . The optimum was found to lie between 150 and 200, depending on the other parameters.



**Figure 4-13: Minimum Distance.** The tuned parameter is the minimum distance between points to be detected  $D_{min}$ . The optimum was found to be around 10 px.



**Figure 4-14: Number of Points.** The tuned parameter is the maximum number of points to be detected  $N_p$ . The optimum heavily depended on the choice of  $Q_{min}$ .



**Figure 4-15: Minimum Quality.** The tuned parameter is the minimum quality of the points to be detected  $Q_{min}$ . The optimum heavily depended on the choice of  $N_p$ .

# Sensor Fusion Algorithm

### 5-1 Introduction

This chapter presents two fusion approaches that provided an estimate of the sideslip angle with the measurements of the ENC, IMU and CVA. In the first approach the CVA measurements were processed individually, this will be referred to as the *individual* fusion approach. In the second approach the median was taken over the CVA measurements which was then put through a low-pass filter as described in Section 4-3-3. This will be referred to as the *combined* fusion approach. Section 5-2 presents the raw signals of the different sensors, as well as the processed signals with their respective accuracies compared to the MCS measurements. To substantiate the necessity of applying sensor fusion in the first place, the results of sideslip estimation with the kinematic model from (2-15) that used only the IMU measurements (Figure 5-5), as well as a method that used only the output of the median filtered CVA (Figure 5-4) are presented as a baseline. In Section 5-3 the sensor fusion algorithm design is described more in detail for both fusion approaches, as well as the tuning of the various parameters in the algorithm. In Section 5-4 the results of the tuning are presented and the two approaches are compared. The best of the two approaches was selected and tested against the validation datasets of which the results are presented in Section 5-4-2. The conclusions drawn in this chapter are critically evaluated in Section 5-4-3.

## 5-2 Signal Conditioning

### 5-2-1 Characteristics

The following measurements were available to estimate the sideslip angle (Figures 5-1, 5-2, 5-3 and 5-4 show the raw measurements of the drift experiment in grey.):

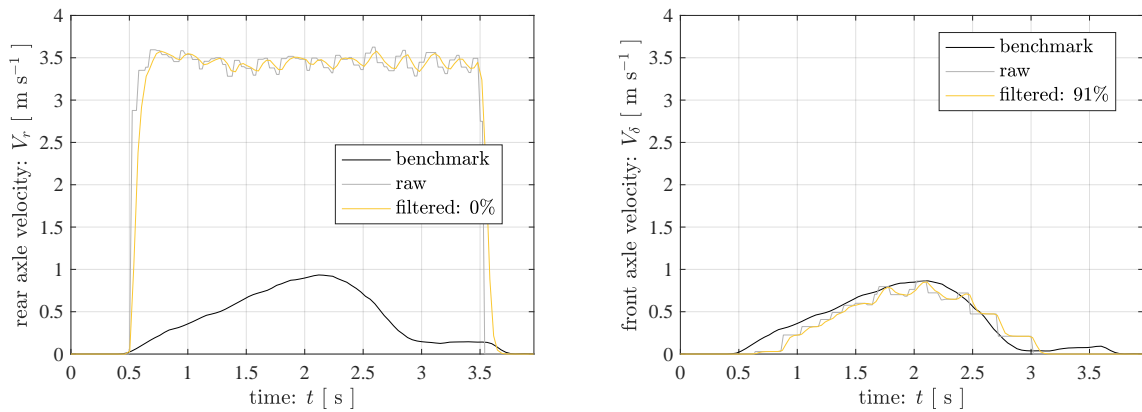
- ENC:  $V_r$ ,  $V_\delta$  (Figure 5-1)
- IMU:  $a_x$ ,  $a_y$ ,  $\omega_z$  (Figure 5-2 and Figure 5-3 )
- CVA:  $V_{xn}$ ,  $V_{yn}$  (Figure 5-4)

The sensors on their own were not able to provide a reliable estimate of the sideslip angle for the following reasons:

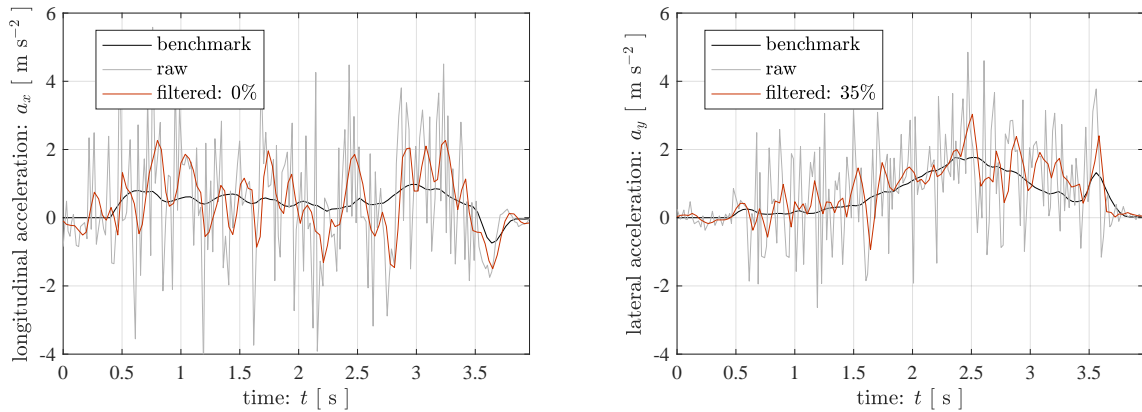
- The rear ENC measured under very high longitudinal slip conditions, so this measurement was not useful for velocity calculation (though it provided an estimate of the longitudinal slip, which could be useful for tire model based control). The front ENC measured under better conditions but had a large delay and was especially inaccurate at low velocities.
- The IMU provided an estimate that is an integral of its measurements based on the kinematic model as was explained in Section 2-2-2. Figure 5-5 shows the result of the sideslip estimation with this model for the drift experiment. As can be seen in the figure, this method is subject to signal drift bias and initial conditions, thus not reliable for estimation over longer time periods.
- The CVA could be very accurate but had extreme unpredictable outliers, even after filtering. The camera that was used was cheap and had a low resolution. Increasing the performance of the CVA was possible but it was expected to come at the cost of a very high computational load. It was more efficient to fuse the measurements for reliability. Figure 5-4 shows the result of the state estimation with only the CVA.

### 5-2-2 Processing

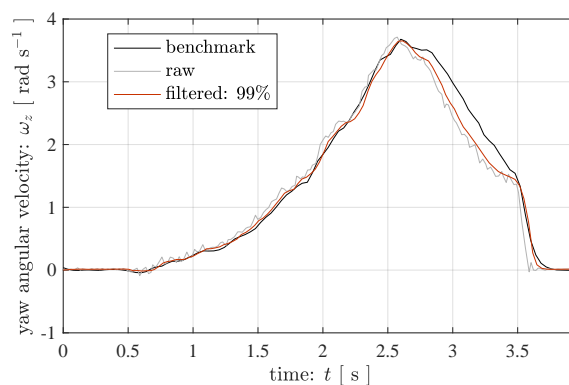
The IMU and ENC acquired their measurements at 50 Hz, the CVA at 30 Hz. The measurements were simultaneously processed at each iteration of the algorithm (50 Hz). The CVA measurement was held between acquisitions to produce an output at 50 Hz that was synchronous with the ENC and IMU. All measurements were put through a low-pass filter (LPF) given in (4-18) to prevent aliasing and to suppress high frequency noise. Only in the combined approach the CVA measurements were processed by the LPF, as was explained in Section 4-3-3. An overview of this process is given in Figure 5-7. Figures 5-1, 5-2, 5-3 and 5-4 show the filtered measurements of the drift experiment in blue. The benchmark dataset was considered to closely approximate reality. This meant that the accuracy of each individual on-board measurement could be established by comparing them against the respective MCS measurement. The VAF of each filtered measurement was calculated with respect to the MCS outputs and the values are provided in the corresponding legends of Figures 5-1, 5-2, 5-3 and 5-4.



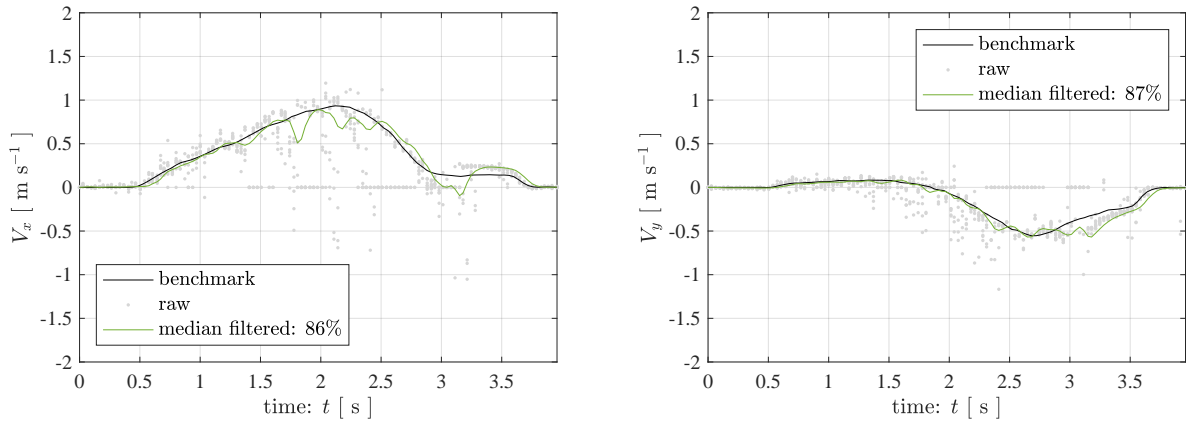
**Figure 5-1: ENC: Axle Velocities.** The rate of acquisition is 50 Hz, but the rate at which the calculation provides a value depends on the velocity amplitude and can be lower. The block-shape of the signals is also explained by the nature of the calculation (see Section 2-2-1 for more details).



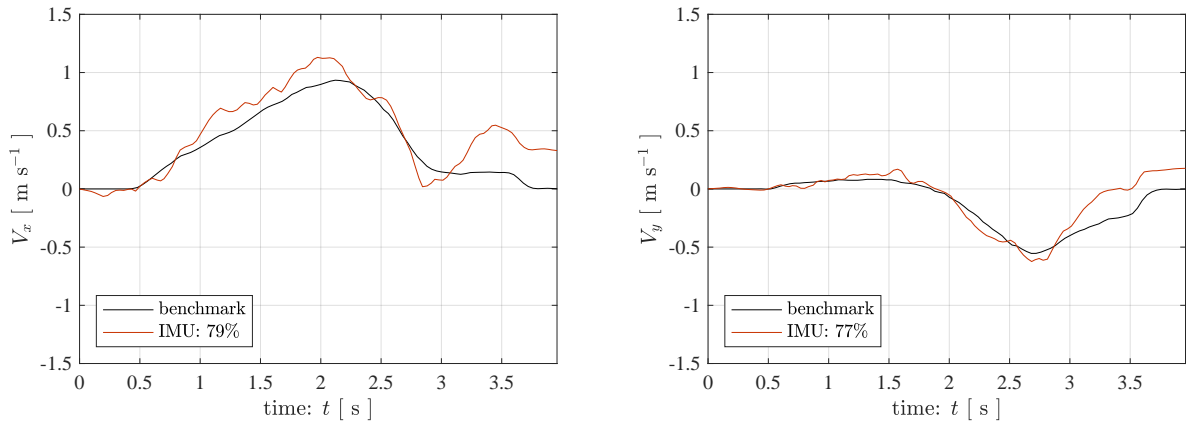
**Figure 5-2: IMU: Linear Accelerations.** The raw signals showed very noisy behaviour and even after filtering, the acceleration measurement still oscillates. Although the VAF of the longitudinal measurement is 0, it can still be observed that the signal somewhat follows the benchmark.



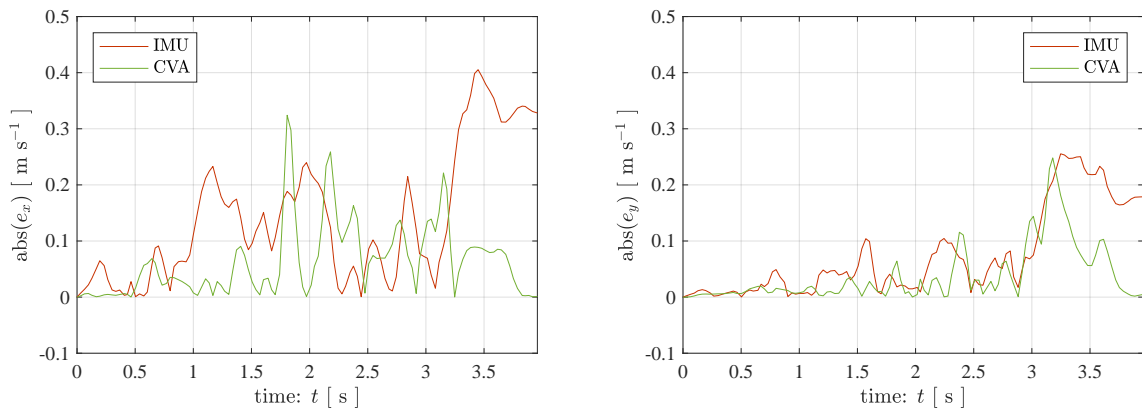
**Figure 5-3: IMU: Yaw Angular Rate.** In the most extreme part of the drift the markers tracked by the MCS were very vague in the picture which influenced the yaw rate measurement. The error between 2.5 and 3.5 s was thought to be an error of the MCS, not the on-board sensor.



**Figure 5-4: CVA: CG-Velocities.** The above plots show the measurements from the tuned CVA. The grey dots are the individual measurements. The green line is the filtered median over these points. The zero measurements are automatically discarded by the algorithm.



**Figure 5-5: IMU: Kinematic Model Result.** The above plots show the output of the kinematic model given in (5-1) and (5-2) for the inputs shown in Figure 5-2 and 5-3.



**Figure 5-6: Absolute Estimation Error for IMU and CVA.** This figure shows the estimation errors of the signals in Figures 5-4 and 5-5. The signal drift of the IMU with kinematic model can clearly be seen towards the end, as well as the outliers in the CVA signal.



## 5-3 Algorithm Design

### 5-3-1 Model Discretisation and Linearisation

The process model as it was defined in (2-15) contained the states that are the central reference points to all other measured quantities. The model, that related the state-transition  $\dot{x}$  to the state  $x$  and the input  $u$ , was defined in continuous time and was discretised at a sampling time  $T_s$ , resulting in (5-1) and (5-2).

$$\Phi(k) = e^{AT_s} = \begin{bmatrix} \cos(\omega_z(k) T_s) & \sin(\omega_z(k) T_s) \\ -\sin(\omega_z(k) T_s) & \cos(\omega_z(k) T_s) \end{bmatrix} \quad (5-1)$$

$$\Gamma(k) = \int_0^{T_s} e^{As} ds B = \frac{1}{\omega_z(k)} \begin{bmatrix} \sin(\omega_z(k) T_s) & 1 - \cos(\omega_z(k) T_s) \\ \cos(\omega_z(k) T_s) - 1 & \sin(\omega_z(k) T_s) \end{bmatrix} \quad (5-2)$$

The measurement model related the state vector  $x$  to the output vector  $y$ . The model differed for the two filter approaches that were developed. In the first approach each point velocity  $V_{xn}$  and  $V_{yn}$  from the CVA was processed individually. This resulted in the process model that can be written as a set of equations: (5-3) for the front axle velocity from the ENC, (5-4) for the longitudinal point velocities and (5-5) for the lateral point velocities from the CVA for each point  $n$ .

$$h_1(\hat{x}, \omega_z, \delta) = \sqrt{\hat{x}_1^2 + (\hat{x}_2 - l_f \omega_z)^2} \cos\left(\tan^{-1}\left(\frac{(\hat{x}_2 - l_f \omega_z)}{\hat{x}_1}\right) - \delta\right) \quad (5-3)$$

$$h_{2n}(\hat{x}_1, y_n, \omega_z) = -(\hat{x}_1 - y_n \omega_z) \quad (5-4)$$

$$h_{3n}(\hat{x}_2, x_n, \omega_z) = -(\hat{x}_2 + x_n \omega_z) \quad (5-5)$$

The above equations were linearised to obtain the measurement model matrix  $H$ , which is the Jacobian matrix of partial derivatives of  $h$  with respect to  $\hat{x}$ :

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}} \quad (5-6)$$

The above resulted in:

$$\frac{\partial h_1}{\partial \hat{x}_1} = \frac{\hat{x}_1 \cos(\delta) \sqrt{\frac{\hat{x}_1^2 + (\hat{x}_2 - l_f \omega_z)^2}{\hat{x}_1^2}}}{\sqrt{\hat{x}_1^2 + (\hat{x}_2 - l_f \omega_z)^2}} \quad \frac{\partial h_1}{\partial \hat{x}_2} = \frac{\hat{x}_1 \sin(\delta) \sqrt{\frac{\hat{x}_1^2 + (l_f \omega_z - \hat{x}_2)^2}{\hat{x}_1^2}}}{\sqrt{\hat{x}_1^2 + (l_f \omega_z - \hat{x}_2)^2}} \quad (5-7)$$

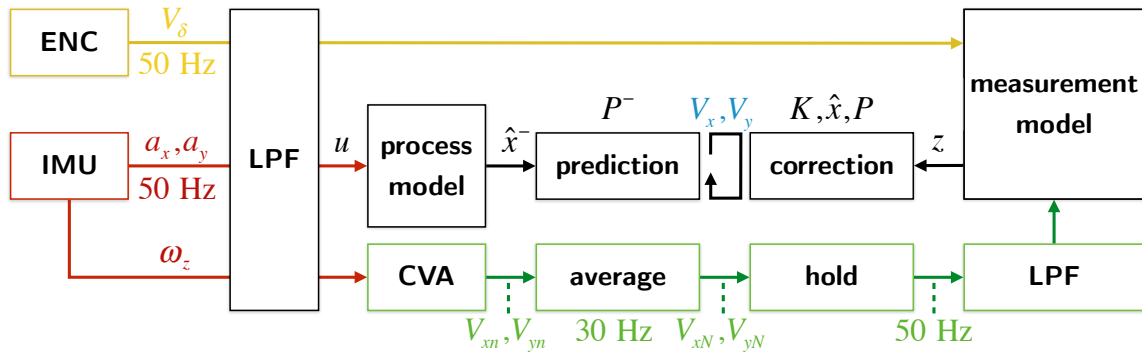
$$\frac{\partial h_2}{\partial \hat{x}_{1n}} = -1 \quad \frac{\partial h_2}{\partial \hat{x}_{2n}} = 0 \quad \frac{\partial h_3}{\partial \hat{x}_{1n}} = 0 \quad \frac{\partial h_3}{\partial \hat{x}_{2n}} = -1 \quad (5-8)$$

In some images the points tracked by the CVA disappeared or were discarded due to insufficient quality, as was explained in Chapter 4. This caused the measurement vector  $z$  to become shorter in length, which influenced the length of vectors  $y$ ,  $z$  and matrices  $H$ ,  $K$  and  $R$ . However, due to the definition of the Kalman filter, this was not a problem. The following definitions show that the arrays may vary in size:

- state-transition model:  $x \in \mathbb{R}^2, \Phi \in \mathbb{R}^{2 \times 2}, u \in \mathbb{R}^2, \Gamma \in \mathbb{R}^{2 \times 2}$
- estimation error covariance:  $P \in \mathbb{R}^{2 \times 2}$
- ENC and CVA measurements of  $N$  tracked points (multiple, varying):  $z \in \mathbb{R}^{1+2N}$
- Non-linear prediction of ENC and CVA measurements (length varying):  $y \in \mathbb{R}^{1+2N}$
- Linearised output (length varying):  $H \in \mathbb{R}^{(1+2N) \times 2}$
- Kalman gain (length varying):  $K \in \mathbb{R}^{2 \times (1+2N)}$
- Measurement noise covariance (length varying):  $R \in \mathbb{R}^{(1+2N) \times (1+2N)}$
- Process noise covariance:  $Q \in \mathbb{R}^{2 \times 2}$

In the second fusion approach the CVA measurements were processed as a single velocity component at the  $CG$  for each axis  $V_{xN}$  and  $V_{yN}$ . The first two entries of  $H$  were given by (5-7), while the 4 other entries were given by:

$$\frac{\partial h_2}{\partial \hat{x}_{1n}} = 1 \quad \frac{\partial h_2}{\partial \hat{x}_{2n}} = 0 \quad \frac{\partial h_3}{\partial \hat{x}_{1n}} = 0 \quad \frac{\partial h_3}{\partial \hat{x}_{2n}} = 1 \quad (5-9)$$



**Figure 5-7: Sensor Fusion Overview.** The ENC and IMU provided measurements at 50 Hz and were processed by the LPF from (4-18). The CVA needed the  $\omega_z$  value from the IMU for the calculations in (4-14) and (4-15). In this diagram the combined approach is shown. In the individual approach the average and LPF blocks in the CVA section did not exist. The hold block synchronised the CVA to the 50 Hz of the IMU, ENC and Kalman filter. The prediction-correction cycle produced the final outputs  $V_x$  and  $V_y$ .

### 5-3-2 Sensitivity Analysis

Both algorithms as they were described in the previous subsection contained 5 tuning parameters. The first two given in (5-10) described the process noise covariance of each state, where  $q_1$  represented the noise covariance on  $x_1$  ( $V_x$ ) and  $q_2$  represented that of  $x_2$  ( $V_y$ ).

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \quad (5-10)$$

The other three parameters given in (5-11) described the measurement noise covariance of each measured quantity. Parameter  $r_1$  was the measurement noise of the ENC axle velocity  $V_\delta$ . In case of the individual CVA point measurements fusion algorithm,  $r_2$  and  $r_3$  are diagonal  $N \times N$  matrices with equal values on each diagonal element for the noise covariances on  $V_{xn}$  and  $V_{yn}$  respectively. In case of the single combined CVA measurement fusion algorithm,  $r_2$  and  $r_3$  were scalars describing the noise covariances on  $V_{xN}$  and  $V_{yN}$  respectively.

$$R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \quad (5-11)$$

The influence of each of the parameters on the estimation accuracy was analysed. Figures C-1, C-2, C-3, C-4, and C-5 in Appendix C-1-1 show the sensitivity analysis results for the individual approach. Figures C-6, C-7, C-8, C-9, and C-10 in Appendix C-1-2 show the sensitivity analysis results for the combined approach. From these analyses optimal values were selected, presented in Tables 5-1 and 5-2.

**Table 5-1: Individual Fusion Algorithm Parameters**

Description	Abbreviation	Value
Process noise covariance on $V_x$	$q_1$	0.01
Process noise covariance on $V_y$	$q_2$	0.01
Measurement noise covariance on $V_\delta$	$r_1$	0.05
Measurement noise covariance on $V_{xn}$	$r_2$	30
Measurement noise covariance on $V_{yn}$	$r_3$	5

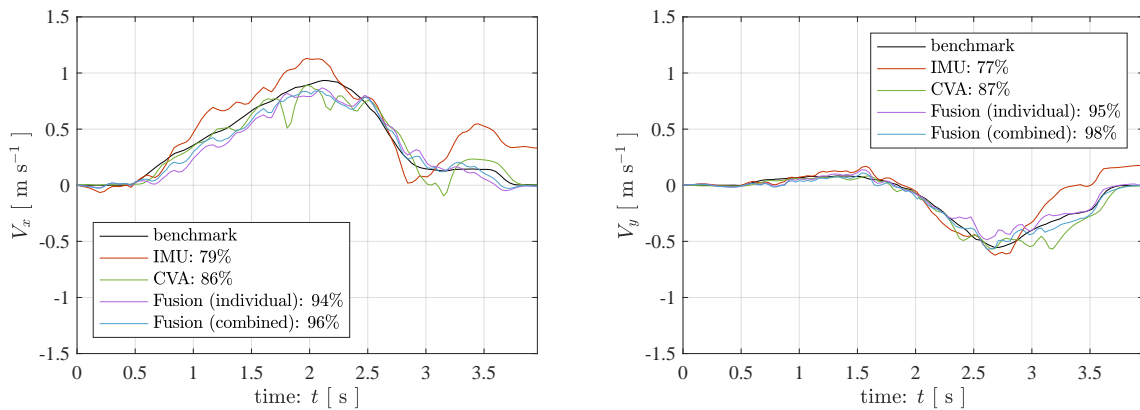
**Table 5-2: Combined Fusion Algorithm Parameters**

Description	Abbreviation	Value
Process noise covariance on $V_x$	$q_1$	0.075
Process noise covariance on $V_y$	$q_2$	0.1
Measurement noise covariance on $V_\delta$	$r_1$	0.075
Measurement noise covariance on $V_{xN}$	$r_2$	2
Measurement noise covariance on $V_{yN}$	$r_3$	2

## 5-4 Results

### 5-4-1 Approach Comparison

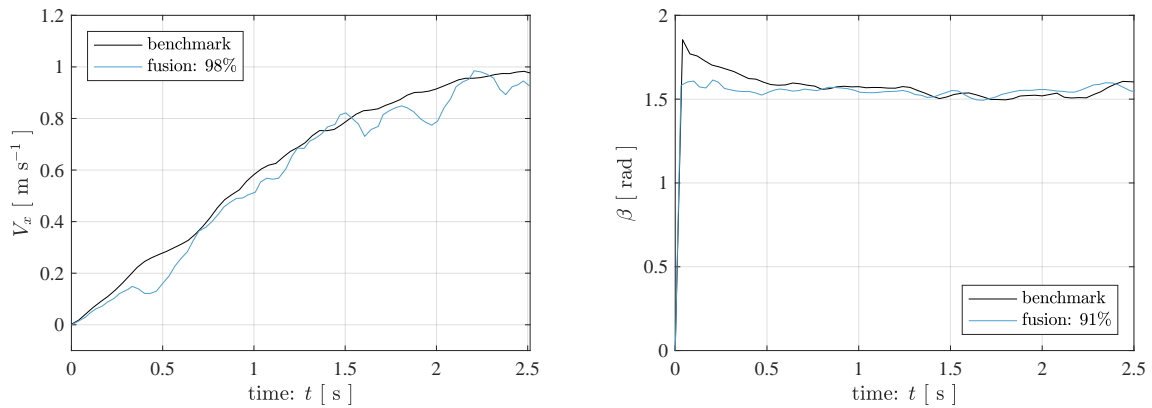
The sensitivity analyses provided optimal values for both fusion approaches. The different approaches were compared with each other. Figure 5-8 shows the accuracies of the different approaches. The first approach was an established technique where the kinematic model and IMU are used to estimate the sideslip. This resulted in an accuracy below 80% for both velocity components and significant drift near the end of the experiment. The CVA developed in Chapter 4 was more accurate, with a VAF of above 80% and no drift. But fusion was even better, surpassing 90% accuracy. Of the two fusion approaches developed in this thesis, it seemed that the combined fusion approach was the best option, with slightly better results. What cannot be shown here is how the computational load differs between the approaches. It was assumed that the combined approach would be computationally lighter. Also, the fusion model was less complicated in case of the combined processing, as the explanations in the previous sections substantiated.



**Figure 5-8: Sideslip Estimation Approach Comparison.** The IMU signal is the output from the kinematic model, which showed significant drift. The CVA showed noisier behaviour. Fusion was the best solution, with the combined approach being superior.

### 5-4-2 Validation

It was concluded in the previous subsection that the combined fusion approach was superior in the drift experiment. However, to minimise the risk of over-fitting the filter parameters, the algorithm was tested on the validation datasets from Section 3-4-2. The algorithm was not adjusted in any way for the tests. In the first experiment pure longitudinal motion was induced, resulting in a zero sideslip angle. During the experiment the lateral motion was not exactly zero however, which made establishing the accuracy of a zero sideslip angle difficult. The left plot in Figure 5-9 therefore shows the estimation accuracy of the longitudinal motion, since this accuracy could be easily established. In the pure lateral motion experiment it was easier to establish the accuracy of the sideslip estimation, of which the results are shown in the right plot of Figure 5-9. It was concluded that the developed algorithm was valid in simulation and the aim of future work could be to validate it in an experiment by running the code on-board.



**Figure 5-9: Combined Fusion Approach Validation.** The left plot shows the estimation of the longitudinal velocity for the experiment shown in Figure 3-10. The right plot shows the estimation of the sideslip angle for the experiment shown in Figure 3-11

### 5-4-3 Critical Evaluation

In this section the conclusions drawn in this chapter are critically evaluated. The first obvious point of critique is that the previously presented validation was the result of a simulation, not an actual test on the experimental vehicle. The main concern with this is that the processing unit on the vehicle might not be able to process the images fast enough to produce the results without a significant delay. The potential consequences are unknown. Another point is that the individual CVA measurements could have been processed in a separate Kalman filter prior to fusion. This could potentially improve the individual fusion approach, which now suffers heavily from its outliers. Another improvement that should have been investigated is tracking of points through multiple frames instead of a single frame in the CVA algorithm. This could significantly decrease the ‘chatter’ behaviour of the CVA point velocity estimation. A fusion approach that was not evaluated in this chapter is the fusion between the ENC and the IMU, without the CVA. This could potentially remove the drift from the IMU measurement, eliminating the need for the complex CVA solution altogether. Finally, the ‘IMU only’ based approach was not tuned, which could have potentially improved its result. On the other hand, the signal drift would still make this approach undesirable, even with better tuning.

## 5-5 Conclusion

This chapter provided two fusion approaches to estimate the sideslip angle with the measurements of the ENC, IMU and CVA. The raw signals of these sensors provided sideslip estimates with accuracies of 70-80% (IMU) and 80-90% (CVA). Fusing the measurements improved the estimates significantly to above 90%. Of the two approaches that were developed in this thesis, the combined method proved to be superior to the individual method. The combined method was tested against validation experiments and was proven to obtain the same accuracy which minimises the probability of over-fitting the filter parameters.



---

## Chapter 6

---

# Conclusion

### 6-1 Summary

#### **Vehicle Active Safety: Autonomous Driving at the Handling Limits**

Car manufacturers heavily invest in research and development in the area of vehicle safety. A widely discussed innovation within vehicle safety engineering is autonomous driving. An autonomous vehicle should be able to avoid obstacles in emergency scenarios. This requires vehicle operation at the handling limits to maximise the performance and safety of the autonomous vehicle. This is considered to be one of the big development challenges in the coming years in the field of vehicle engineering.

#### **Automated Drift Control Development: Feasible State Estimation**

The limits of vehicle handling are reached when the tires lose grip and start slipping, known as drift motion. Drift motion has been investigated in the past decade and analyses revealed that equilibria exist in this motion state. Race driver behaviour has been studied which inspired novel design approaches for automated drift controllers. The literature on this topic showed that controlling drift motion for autonomous vehicles is possible, but feasible state estimation methodologies have not yet been proposed or tested. Current control methodologies rely on external sensors and very precise vehicle models using tire force calculations. This thesis had the objective to develop and test a state estimation methodology that only relied on simple on-board sensors and did not require a dynamic vehicle model to estimate the sideslip angle.

## Sideslip Estimation Methodology

Chapter 2 presented a state estimation methodology for the sideslip of a vehicle that exclusively utilised on-board sensors, namely: rotary encoders, an inertial measurement unit and a camera. The on-board sensor suite was laid. The basic working principles were explained and the measured quantities of the sensors were related to the *CG*-states. A Kalman filter was defined with a process model with inputs and outputs that related to the measured quantities from the sensors. An experimental setup was presented on which the state estimation technique was implemented and validated.

## Motion Capture System

Chapter 3 explained the motion capture system that provided benchmark and validation datasets for the state estimation design process. The benchmark dataset was an unstable drift resembling the experiments carried out in drift control development. Each stage of the MCS data processing was described referring to the benchmark experiment as an example. A method was presented to extract the location and heading of the vehicle throughout the experiment. These signals were differentiated over time using a differentiator with low-pass filter. A technique was presented by which the obtained states were transformed to the desired outputs that were needed as benchmark and validation.

## Computer Vision Algorithm

Chapter 4 described the design of a computer vision algorithm that used the on-board camera to measure the longitudinal- and lateral velocity. The image was processed by correcting for distortion and isolating the red coloured markers in an ROI. After applying a greyscale conversion to the colour channels, the image is ready for the detection- and tracking stage. Corners of the markers are detected in the greyscale image by a Shi-Tomasi detector. A Lukas-Kanade optical flow function tracks the points through consecutive frames. Each tracked point location was obtained in pixels, so a conversion to actual coordinates in meters was needed for the final measurement output of the algorithm. This was achieved through homographic projection. The benefit of this approach was that it removes any perspective from the image without requiring additional steps.

## Sensor Fusion Algorithm

The raw signals of the sensors provided sideslip estimates with accuracies of 70-80% (IMU) and 80-90% (CVA). However, the estimates based on the IMU were subject to drift while the CVA showed significant outliers. To improve the estimates Chapter 5 provided two approaches to fused the measurements of the ENC, IMU and CVA. In the first approach the CVA measurements were processed individually. In the second approach the median of the CVA measurements was taken, which was then put through a low-pass filter. Fusing the measurements improved the estimates significantly (above 90%). The combined method proved to be superior to the individual method. The combined method was tested against validation experiments and obtained the same accuracy (90-100%) which excluded the possibility of over-fitting the filter parameters.



## 6-2 Recommendations

### Improved Hardware

The experimental platform that has been used in this project has been replicated and improved by a group of B.Sc. students at the TU Delft [76]. This allows for more advanced projects in the future that could use the state estimation methodology proposed in this thesis. The recommendations presented next both apply to this vehicle and the vehicle that has been developed in Milan.

### On-Line Validation

The next step in the development of the methodology presented in this thesis would be to validate the algorithm on-line. The script is already written in Python which allows it to run on the Ubuntu platform. By validating the algorithm on-line the computational effort can be determined which would be a basis for an optimisation in that area.

### System Identification and Control Design

Once a lightweight, optimal state estimator is in place, control designs can be proposed in a simulation environment that allow the vehicle to make autonomous drifts. This would require modelling of the vehicle dynamics in more detail. The MCS can be used to identify the system dynamics to support the control design process, but the state estimator could serve this role as well, as the accuracy is above 90%.

### Marker-Less Computer Vision

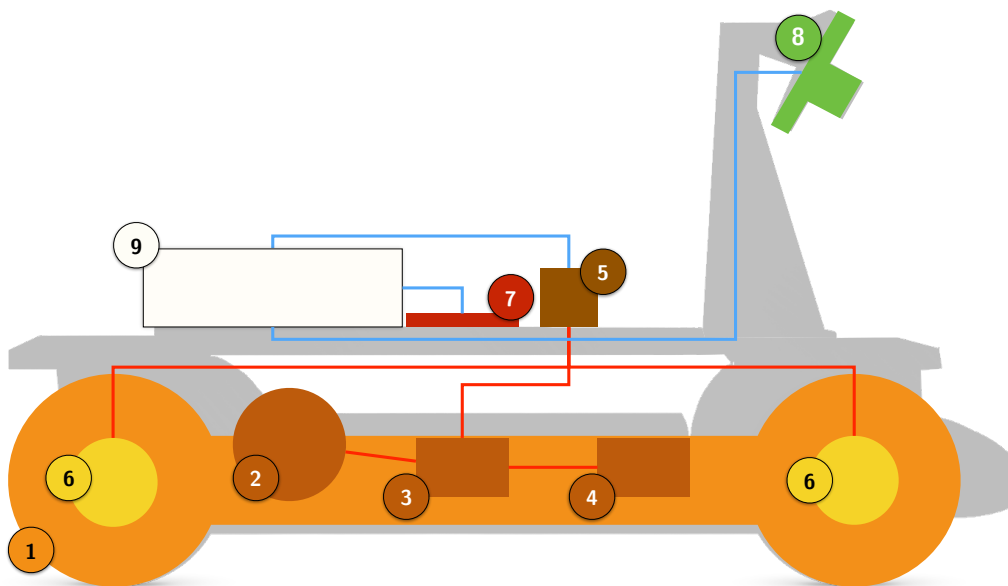
The CVA algorithm could be improved as well, to be able to work without markers on the ground. Or it could use markers that are common on the highway (white stripes). The basis of the algorithm could be used for more purposes than drift control, such as lane-keeping. This would simply be a matter of changing the colour thresholds and processing of the egomotion calculation.

### Collaboration between Delft and Milan

Finally, this project is one of hopefully many more collaborations between the TU Delft and the Politecnico di Milano on autonomous vehicle control development. Now that both universities have working test platforms using the same hard- and software it is easy to share algorithms and data. The platform is also suitable for research at all levels (B.Sc, M.Sc, Ph.D. and Postdoc). Studies could also be performed simultaneously on the platform, as the vehicle frame has a modular structure. This allows the removal and placement of many different sensors. An important notice to researchers in Delft: do not forget to adjust the computer vision algorithm, the algorithm was tuned for Italian blue sky interference, which is unfortunately not common in Delft.



# Vehicle Hardware Specifications



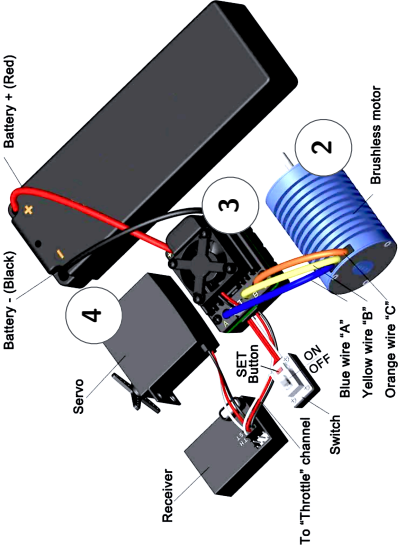
**Figure A-1: Vehicle Hardware Overview.** Part (1) is the chassis including the motor (2), ESC (3) and servo (4). Part (5) is the Arduino. The red lines indicate analog interfaces. Parts (6) indicate the encoders. Part (7) is the IMU, (8) is the camera. The blue lines indicate digital interfaces. Part (9) is the Ordoid.

Table A-1: Vehicle Component Specifications

#	Component	Type	Comments	url
1	Chassis	RC car chassis - Basher RZ-4 1/10 scale RTR	Includes suspension, motor (2), ESC (3), servo (4), transmission and wheels.	<a href="https://hobbyking.com/en_us/rz-4v2kit.html">https://hobbyking.com/en_us/rz-4v2kit.html</a>
5	Hardware Interface	Mini Nano V3.0 ATmega328P Microcontroller Board	Including all necessary cables and structural components.	<a href="https://www.amazon.com/ATmega328P-Microcontroller-Board-Cable-Arduino/dp/B00N1AMS9C/ref=sr_1_2?ie=UTF8&amp;id=1442638774&amp;sr=8-2&amp;keywords=arduino">https://www.amazon.com/ATmega328P-Microcontroller-Board-Cable-Arduino/dp/B00N1AMS9C/ref=sr_1_2?ie=UTF8&amp;id=1442638774&amp;sr=8-2&amp;keywords=arduino</a>
6	Encoders	Hall Effect Sensor - US1881	3D printed disk on axle keeps sensor in place, magnets included.	<a href="https://www.sparkfun.com/products/9312">https://www.sparkfun.com/products/9312</a>
7	IMU	myAHRS+ (Attitude Heading Reference System)	Communication and configuration are enabled via UART / USB	<a href="http://amerindroid.com/products/myahrs-attitude-heading-reference-system">http://amerindroid.com/products/myahrs-attitude-heading-reference-system</a>
8	Camera	ELP 2.1mm Lens 1080p HD Free Driver USB Camera Module for Linux	3D printed mount on vehicle to face downwards on ground surface.	<a href="https://www.amazon.com/ELP-Driver-Camera-Module-ELP-USB-FHD01M-L21/dp/B00KATWSSU">https://www.amazon.com/ELP-Driver-Camera-Module-ELP-USB-FHD01M-L21/dp/B00KATWSSU</a>
9	Microcontroller	ODROID-XU4 Single Board Computer with Quad Core 2GHz A15, 2GB RAM	Including 3D printed enclosure, mounting structure, USB port, Wi-fi router.	<a href="https://www.amazon.com/ODROID-XU4-Single-Board-Computer-Gigabit/dp/B0163GEA64">https://www.amazon.com/ODROID-XU4-Single-Board-Computer-Gigabit/dp/B0163GEA64</a>



1



2 Brushless motor

3

4 Servo

Battery - (Black)

Battery + (Red)

Receiver

SET Button

To "Throttle" channel

Switch

Blue wire "A"

Yellow wire "B"

Orange wire "C"



7



8



9

---

## Appendix B

---

# **Computer Vision Function Parameters**

Table B-1: Computer Vision Algorithm Parameters for External Camera.

Parameter	Symbol	Value	Unit	Motivation
Acquisition	frame width	$W_f$	1920 px	The external camera is attached to the ceiling ( $> 3$ m above ground) and tracks markers less than 2 cm in diameter. It is therefore necessary to choose a high resolution, to make sure the markers are captured with sufficient accuracy. The chosen resolution results in a frame rate of 25 frames per second by default, which is 5 frames per second less than the rate of the on-board algorithm which it needs to verify. This is acceptable however for the following reasons: the external camera is fixed and the markers are always in the image, and all other parameters have no restrictions in terms of computational load. Both conditions contribute to a more accurate measurement, so 5 frames per second does not make a difference in terms of accuracy when compared to the on-board algorithm.
	frame height	$H_f$	1080 px	
	frame rate	fps	25 s <sup>-1</sup>	
Region of Interest	top edge	$e_T$	0 px	The most important parameter here is the left edge crop. The raw image contains surfaces that are not part of the experimental setup, but do contain some green colour in them and thus distract the search algorithm from finding the markers. The right edge does only contain the test surface, but a darker part of it which is partly in a shadow and the same holds as for the left edge, it does contain some green colour.
	right edge	$e_R$	100 px	
	bottom edge	$e_B$	0 px	
	left edge	$e_L$	300 px	
Color Range	red channel	$R$	{0, 160}	The raw image coming from the external camera is edited first before entering the algorithm. A blue temperature filter is applied to remove the green colour from the test surface, which interferes with the markers on the vehicle. To remove this blue, the upper limit of the blue channel is restricted to 140. To emphasize the green of the markers, the lower limit of the green channel is set to 100. The red channel limit of 160 is relatively high but during tuning it was concluded that it needed to be at least 160. This is probably because the green vehicle markers are moving just above the red markers on the floor, sometimes interfering with each other.
	green channel	$G$	{100, 255}	
	blue channel	$B$	{0, 140}	
Detection	max number of points	$N_p$	3 -	There are only three markers placed on the vehicle to track, therefore a maximum of 3 points will be detected. It has been measured on the initial calibration picture that the markers are between 80 and 90 px away from each other, so the minimum is set to 80 to make sure all three markers are detected individually instead of multiple corners on a single marker. It has been observed that the quality level does not matter. In fact, a too low value could result in the absence of one or more detected markers.
	min distance	$D_{\min}$	80 px	
	min quality	$Q_{\min}$	0.25 -	
Tracking	window width	$u_W$	100 px	Counterintuitively, it seemed better to increase the window size to extend beyond the distance between the markers. Apparently the algorithm is able to distinguish the corners better when taking into account more information around the original point. Increasing the pyramid level was also found to be better, because the lower resolution resulted in less possibilities to mistake one point for another. So what in general is a more accurate solution does not hold in this case. The minimum quality was set at 0.01, below this value the performance did not increase. Same goes for the number of iterations, for which a higher amount than 100 did not change the result. All of the above occlusions were drawn based on looking at the result by eye, because there is not other quantitative data to compare it against.
	window height	$v_W$	100 px	
	max pyramid level	$L_{\text{pyr}}$	3 -	
	min quality	$\epsilon$	0.01 -	
	max number of iterations	$N_{\max}$	100 -	

Table B-2: Computer Vision Algorithm Parameters for On-Board Camera.

Parameter	Symbol	Value	Unit	Motivation
Acquisition	frame width	$W_f$	640 px	This is the lowest resolution at which the on-board camera can operate (without hacking the default settings). Increasing the resolution is not necessary since the camera is close to the ground and increasing it would unnecessarily increase the computational load. The default frame rate that is optimal for the chosen resolution is 30 fps based on the camera.
	frame height	$H_f$	480 px	
	frame rate	fps	30 s <sup>-1</sup>	
Color Range	red channel	$R$	{75, 255}	The on-board camera needs to track red markers, thus the range should emphasise the red channel. Clearly, 255 is the upper limit for the red colour channel. The lower limit of 75 was found through the sensitivity analysis. Green is the inverse colour to red, and thus has a lower maximum than blue in this case. The lower limit is 0, of course. The upper limit was found through the sensitivity analysis.
	green channel	$G$	{0, 75}	
	blue channel	$B$	{0, 175}	
Detection	max number of points	$N_p$	8 -	The tuning of the maximum number of points is a trade-off between accuracy and computational load. A higher value increases accuracy and load. This will therefore be tuned later when comparing results. The minimum distance is not as relevant for the on-board camera as for the external camera because the points of interest have no specified distance from each other but the accuracy increases by choosing 10 px. For the minimum quality level the same trade-off holds as for $N_p$ .
	min distance	$D_{\min}$	10 px	
	min quality	$Q_{\min}$	0.001 -	
Tracking	window width	$u_W$	25 px	The window size is not necessarily a trade-off between load and accuracy. A large window is sometimes less accurate because the algorithm may find other points that are similar to the points it is tracking. The pyramid level is a tuning parameter where the maximum pyramid level is 1, the minimum 3. Level 1 is heavy and accurate, 3 is lighter but less accurate (in general). Minimum is the case of the quality level in qualitative terms means maximum, since the quality is higher for a lower value of $\epsilon$ . A higher $\epsilon$ decreases accuracy and load. A higher number of iterations increases accuracy and computational time.
	window height	$v_W$	25 px	
	max pyramid level	$L_{\text{pyr}}$	3 -	
	min quality	$\epsilon$	0.001 -	
	max number of iterations	$N_{\max}$	1 -	





---

## Appendix C

---

# Sensor Fusion Results

### C-1 Sensitivity Analysis

#### C-1-1 Individual Processing

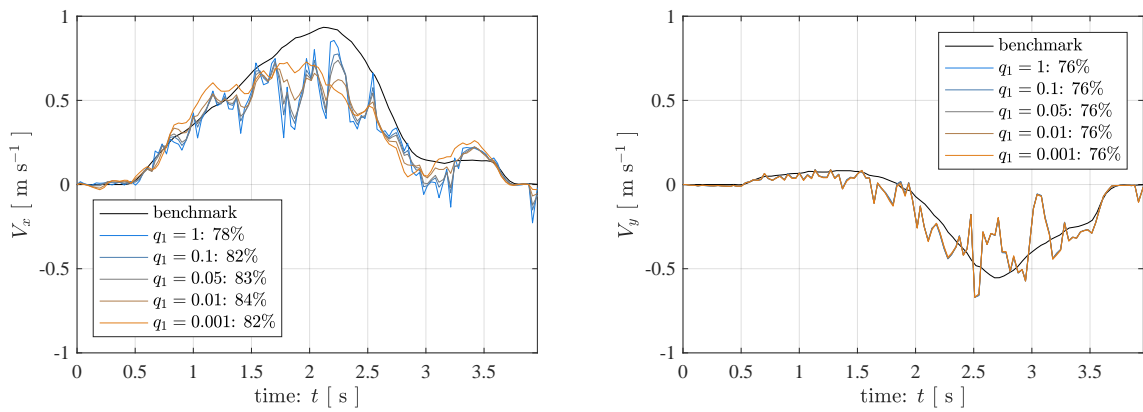


Figure C-1: Process Noise Covariance on  $V_x$

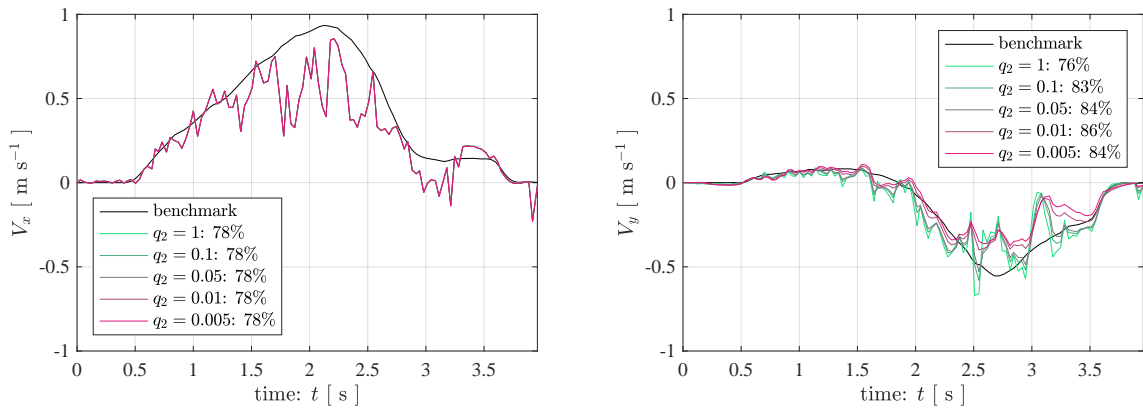
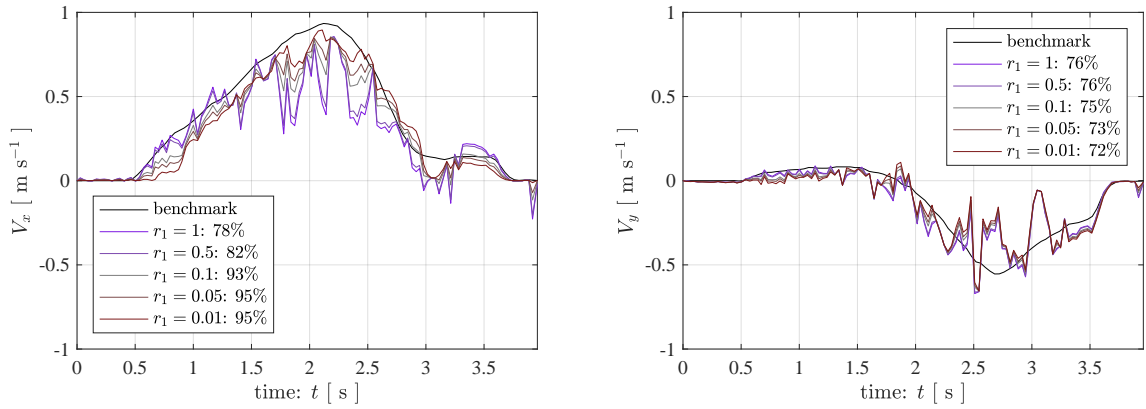
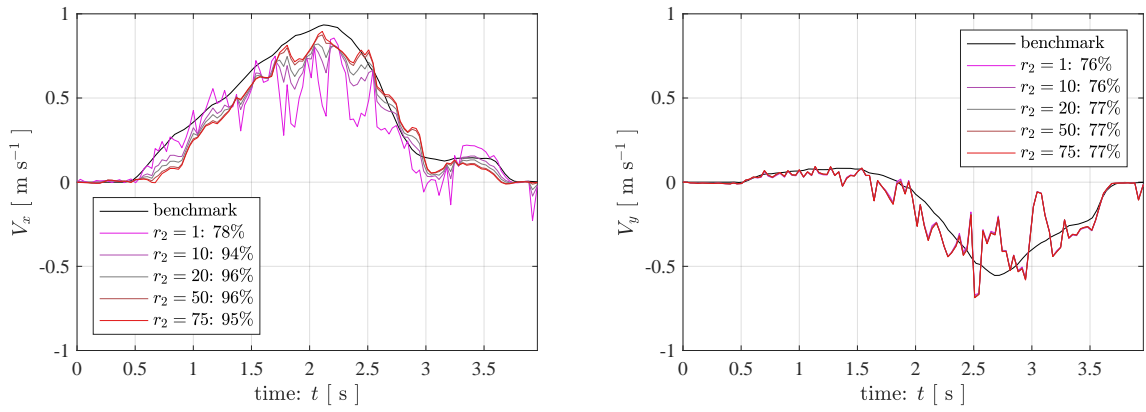
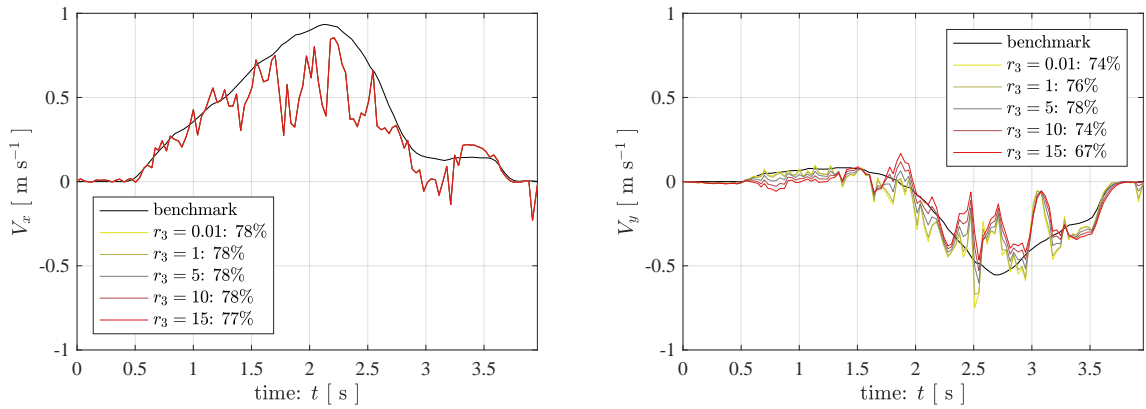


Figure C-2: Process Noise Covariance on  $V_y$

Figure C-3: Measurement Noise Covariance on  $V_\delta$ Figure C-4: Measurement Noise Covariance on  $V_{xn}$ Figure C-5: Measurement Noise Covariance on  $V_{yn}$

### C-1-2 Combined Processing

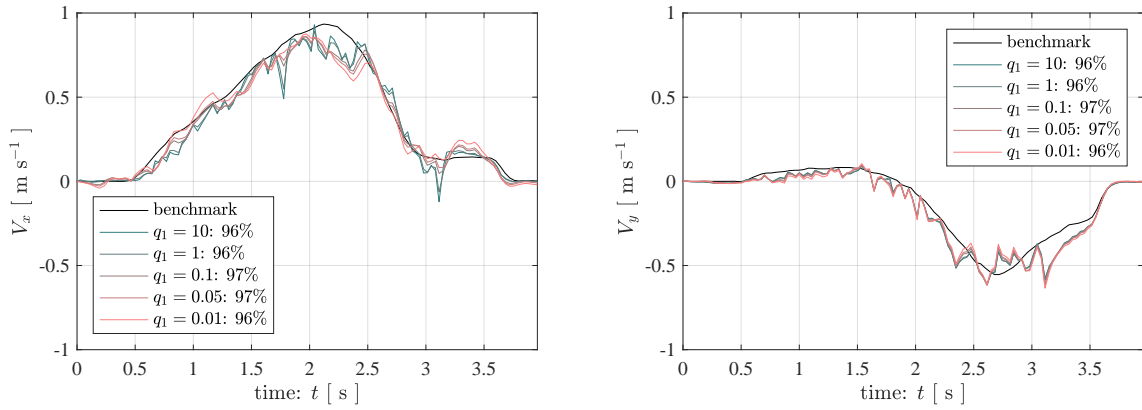


Figure C-6: Process Noise Covariance on  $V_x$

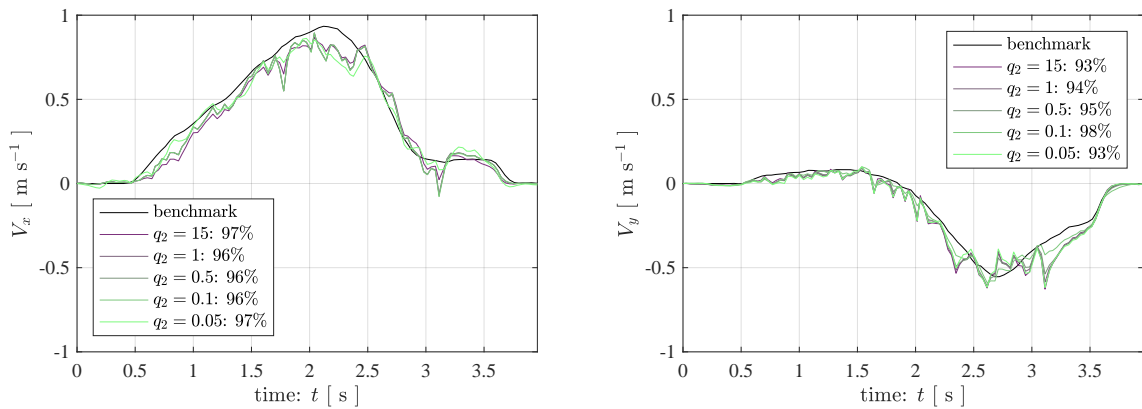


Figure C-7: Process Noise Covariance on  $V_y$

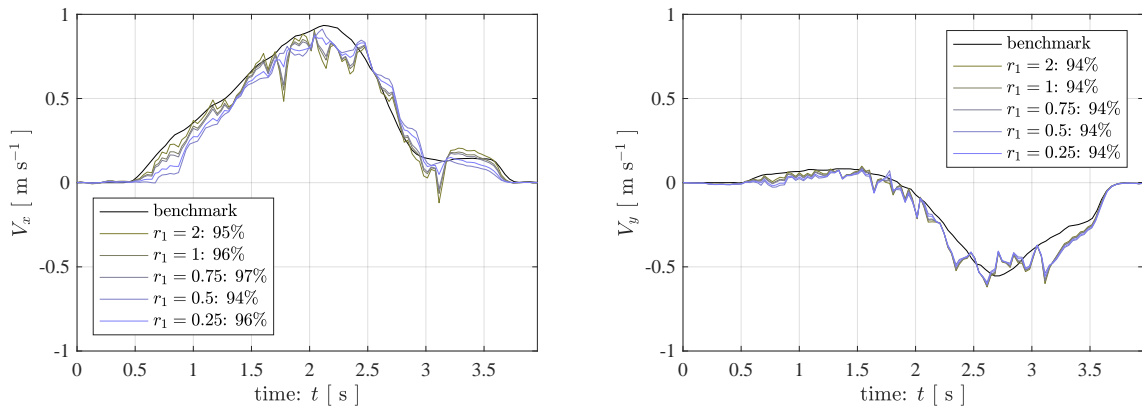
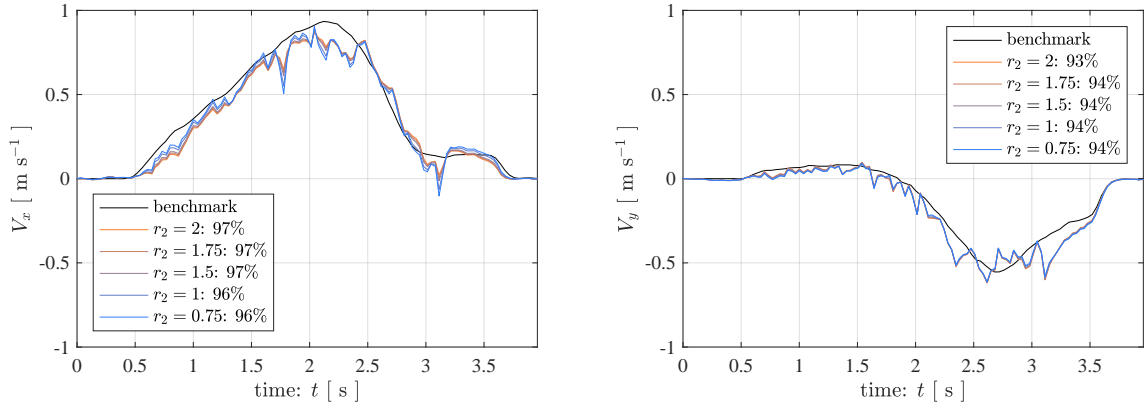
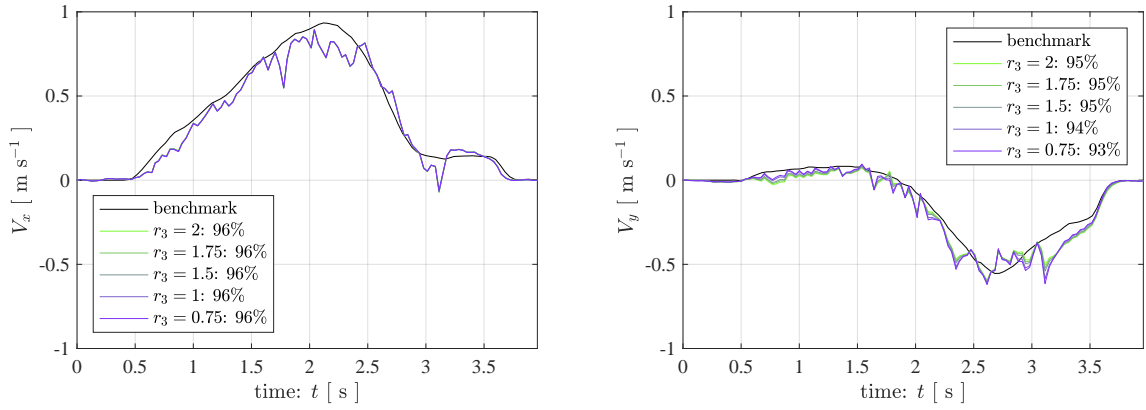
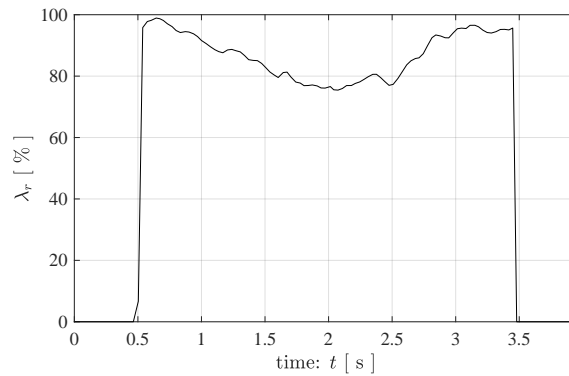


Figure C-8: Measurement Noise Covariance on  $V_\delta$

Figure C-9: Measurement Noise Covariance on  $V_{xN}$ Figure C-10: Measurement Noise Covariance on  $V_{yN}$ 

## C-2 Additional Estimation Result: Rear Wheel Longitudinal Slip

Figure C-11: Drift Experiment: Rear Wheel Longitudinal Slip:  $\lambda_r$

---

# Bibliography

- [1] *TU Delft, Master Track Vehicle Engineering.* [Online]. Available: <http://www.tudelft.nl/en/study/master-of-science/master-programmes/mechanical-engineering/tracks-overview/vehicle-engineering/>
- [2] “Global health observatory (gho) data,” *World Health Organization*, 2015.
- [3] S. P. Baker and W. Haddon Jr, “Reducing injuries and their results: the scientific approach,” *The Milbank Memorial Fund Quarterly. Health and Society*, pp. 377–389, 1974.
- [4] D. Skoyles, “Anti-lock brake systems,” 1973, uS Patent 3,724,914.
- [5] T. Ehret and U. Hartmann, “Method for controlling motor vehicle stability,” 1995, uS Patent 5,402,342.
- [6] S. A. Shappell and D. A. Wiegmann, *A human error approach to aviation accident analysis: The human factors analysis and classification system.* Ashgate Publishing, Ltd., 2012.
- [7] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, “Autonomous driving: technical, legal and social aspects,” 2016.
- [8] P. Ioannou and A. Bose, “Automated vehicle control,” 2005.
- [9] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Puetzold, and C. Wohler, “Autonomous driving goes downtown,” *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 6, pp. 40–48, 1998.
- [10] S. Munder, C. Schnorr, and D. M. Gavrila, “Pedestrian detection and tracking using a mixture of view-based shape–texture models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 333–343, 2008.

- [11] M. Alirezaei, S. T. Jansen, A. J. Schmeitz, and A. K. Madhusudhanan, "Collision avoidance system using state dependent riccati equation technique: An experimental robustness evaluation," in *Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16), September 13-16, 2016, Munich, Germany*. CRC Press/Balkema, Taylor & Francis Group, 2016.
- [12] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.
- [13] B. Heissing and M. Ersoy, "Driving dynamics," in *Chassis Handbook*. Springer, 2011, pp. 35–147.
- [14] M. Itoh, M.-P. P. Lemoine, F. Robache, and H. Morvan, "An analysis of driver's avoiding maneuver in a highly emergency situation," *SICE Journal of Control, Measurement, and System Integration*, vol. 8, no. 1, pp. 27–33, 2015.
- [15] R. Rajamani, "Electronic stability control," in *Vehicle Dynamics and Control*. Springer, 2012, pp. 201–240.
- [16] *World Rally Championship*. [Online]. Available: <http://www.wrc.com>
- [17] K. Kritayakirana and J. C. Gerdes, "Autonomous vehicle control at the limits of handling," *International Journal of Vehicle Autonomous Systems*, vol. 10, no. 4, pp. 271–296, 2012.
- [18] M. Abdulrahim, "On the dynamics of automobile drifting," SAE Technical Paper, Tech. Rep., 2006.
- [19] E. Velenis, P. Tsiotras, and J. Lu, "Optimality properties and driver input parameterization for trail-braking cornering," *European Journal of Control*, vol. 14, no. 4, pp. 308–320, 2008.
- [20] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Vehicle System Dynamics*, vol. 48, no. 1, pp. 317–336, 2010.
- [21] H. Dugoff, P. Fancher, and L. Segel, "An analysis of tire traction properties and their influence on vehicle dynamic performance," SAE Technical Paper, Tech. Rep., 1970.
- [22] E. NATHMAN and C. SMITH, "An analytical and experimental study of automobile dynamics with random roadway inputs," 1977.
- [23] D. F. Moore, *The friction of pneumatic tyres*, 1975.
- [24] SAEJ670, "Vehicle dynamics terminology," 2008.
- [25] C. B. Winkler, "Simplified analysis of the steady-state turning of complex vehicles," *Vehicle System Dynamics*, vol. 29, no. 3, pp. 141–180, 1998.
- [26] J. Edelmann and M. Plöchl, "Handling characteristics and stability of the steady-state powerslide motion of an automobile," *Regular and Chaotic Dynamics*, vol. 14, no. 6, pp. 682–692, 2009.
- [27] *CarSim*. [Online]. Available: <https://www.carsim.com/products/carsim>

- 
- [28] E. Velenis and P. Tsiotras, "Minimum time vs maximum exit velocity path optimization during cornering," in *2005 IEEE international symposium on industrial electronics*, 2005, pp. 355–360.
  - [29] T. O'Neil, "Rally driving manual," *Team O'Neil Rally School and Car Control Center, Dalton, OH*, 2006.
  - [30] R. Y. Hindiyeh and J. C. Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, p. 051015, 2014.
  - [31] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 597–602.
  - [32] H. Nakano, J. Kinugawa, and K. Kosuge, "Control of a four-wheel independently driven electric vehicle with a large sideslip angle," in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*. IEEE, 2014, pp. 265–270.
  - [33] —, "Trajectory tracking control of a vehicle with a large sideslip angle," in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2015, pp. 971–976.
  - [34] M. Werling, P. Reinisch, and L. Gröll, "Robust power-slide control for a production vehicle," *International Journal of Vehicle Autonomous Systems*, vol. 13, no. 1, pp. 27–42, 2015.
  - [35] B. Churcher, "Differential global positioning system," 2011.
  - [36] E. Hose, "Inertial navigation system." DTIC Document, Tech. Rep., 1978.
  - [37] J. Ryu and J. C. Gerdes, "Integrating inertial sensors with global positioning system (gps) for vehicle dynamics control," *Transactions-American Society for Mechanical Engineers: Journal of Dynamic Systems Measurement and Control*, vol. 126, no. 2, pp. 243–254, 2004.
  - [38] Y.-H. J. Hsu, S. M. Laws, and J. C. Gerdes, "Estimation of tire slip angle and friction limits using steering torque," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 896–907, 2010.
  - [39] C. E. Beal, S. P. Boyd, J. C. Gerdes, and S. Beiker, *Applications of model predictive control to vehicle dynamics for active safety and stability*. Stanford University, 2011.
  - [40] S. Ito, M. Nagao, K. Toki, and K. Morita, "Magnetic rotary encoder for detection of incremental angular displacement," Mar. 9 1982, uS Patent 4,319,188.
  - [41] M. M. Morrison, "Inertial measurement unit," Dec. 8 1987, uS Patent 4,711,125.
  - [42] A. T. Van Zanten, "Bosch esp systems: 5 years of experience," SAE Technical Paper, Tech. Rep., 2000.

- [43] J. Farrelly and P. Wellstead, "Estimation of vehicle lateral velocity," in *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on.* IEEE, 1996, pp. 552–557.
- [44] M. Hiemer, A. von Vietinghoff, U. Kiencke, and T. Matsunaga, "Determination of the vehicle body side slip angle with non-linear observer strategies," SAE Technical Paper, Tech. Rep., 2005.
- [45] L. Imsland, H. F. Grip, T. A. Johansen, T. I. Fossen, J. C. Kalkkuhl, and A. Suissa, "Nonlinear observer for vehicle velocity with friction and road bank angle adaptation-validation and comparison with an extended kalman filter," SAE Technical Paper, Tech. Rep., 2007.
- [46] G. Baffet, A. Charara, and D. Lechner, "Estimation of vehicle sideslip, tire force and wheel cornering stiffness," *Control Engineering Practice*, vol. 17, no. 11, pp. 1255–1264, 2009.
- [47] D. M. Bevly, R. Sheridan, and J. C. Gerdes, "Integrating ins sensors with gps velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 1. IEEE, 2001, pp. 25–30.
- [48] W. Deng and H. Zhang, "Rls-based online estimation on vehicle linear sideslip," in *American Control Conference, 2006.* IEEE, 2006, pp. 6–pp.
- [49] J. Ryan, D. Bevly, and J. Lu, "Robust sideslip estimation using gps road grade sensing to replace a pitch rate sensor," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on.* IEEE, 2009, pp. 2026–2031.
- [50] K. Nam, H. Fujimoto, and Y. Hori, "Lateral stability control of in-wheel-motor-driven electric vehicles based on sideslip angle estimation using lateral tire force sensors," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 1972–1985, 2012.
- [51] M. Litzenberger, B. Kohn, A. Belbachir, N. Donath, G. Gritsch, H. Garn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.* IEEE, 2006, pp. 653–658.
- [52] K. Nam, S. Oh, H. Fujimoto, and Y. Hori, "Vehicle state estimation for advanced vehicle motion control using novel lateral tire force sensors," in *American Control Conference (ACC), 2011.* IEEE, 2011, pp. 4853–4858.
- [53] A. Kunnappillil Madhusudhanan, M. Corno, and E. Holweg, "Sliding mode-based lateral vehicle dynamics control using tyre force measurements," *Vehicle System Dynamics*, vol. 53, no. 11, pp. 1599–1619, 2015.
- [54] M. Corno, M. Gerard, M. Verhaegen, and E. Holweg, "Hybrid abs control using force measurement," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1223–1235, 2012.
- [55] D. D. Hrovat, H. E. Tseng, and J. L. Yester, "Vehicle dynamics measuring apparatus and method using multiple gps antennas," Dec. 30 2003, uS Patent 6,671,587.



- 
- [56] S. Ishida and J. E. Gayko, "Development, evaluation and introduction of a lane keeping assistance system," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 943–944.
  - [57] P. I. Labuhn and W. J. Chundrlik Jr, "Adaptive cruise control," Oct. 3 1995, uS Patent 5,454,442.
  - [58] J. Turner, *Automotive sensors*. Momentum Press, 2009.
  - [59] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.
  - [60] G. P. Donald Selmanaj, Matteo Corno and S. M. Savaresi, "Vehicle sideslip estimation: A kinematic based approach," 2016.
  - [61] A. Jaegle, S. Phillips, and K. Daniilidis, "Fast, robust, continuous monocular egomotion computation," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 773–780.
  - [62] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall Upper Saddle River, NJ, 2000, vol. 1.
  - [63] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
  - [64] P. S. Maybeck, "Stochastic models, estimation, and control," 1979.
  - [65] *Berkeley Autonomous Race Car*. [Online]. Available: <http://www.barc-project.com>
  - [66] *Robot Operating System*. [Online]. Available: <http://www.ros.org>
  - [67] J. R. Parker, *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.
  - [68] Z. Zhang, "A flexible new technique for camera calibration," vol. 22, 2000, p. 1330–1334.
  - [69] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques," *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.
  - [70] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, "A robust visual odometry and precipice detection system using consumer-grade monocular vision," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3421–3427.
  - [71] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.
  - [72] C. Tomasi and J. Shi, "Direction of heading from image deformations," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*. IEEE, 1993, pp. 422–427.

- [73] J. Shi *et al.*, “Good features to track,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on.* IEEE, 1994, pp. 593–600.
- [74] C. Harris and M. Stephens, “A combined corner and edge detector.” in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.
- [75] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” 1981.
- [76] T. Kalidien, P. van der Burg, A. Mulder, E. Rietveld, and M. Vonk, “Design and development of the delft scaled vehicle: A platform for autonomous driving tests,” 2017.

---

# Glossary

## List of Acronyms

<b>ABS</b>	anti-lock braking system
<b>MCS</b>	motion capture system
<b>DGPS</b>	differential global positioning system
<b>INS</b>	inertial navigation system
<b>IMU</b>	inertial measurement unit
<b>LKA</b>	lane-keeping assistance
<b>ACC</b>	adaptive cruise control
<b>LIDAR</b>	light detection and ranging
<b>ENC</b>	rotary encoder
<b>HE</b>	Hall-effect
<b>CVA</b>	computer vision algorithm
<b>BARC</b>	Berkeley Autonomous Race Car
<b>ESC</b>	electronic speed control
<b>ROS</b>	robot operating system

## List of Symbols

$\alpha$	Tire sideslip angle
$\beta$	Vehicle sideslip angle
$\delta$	Front wheel steering angle
$\lambda$	Longitudinal tire slip
$\mu$	Friction coefficient
$\omega_z$	Yaw angular rate
$CG$	Centre of gravity
$F_x$	Longitudinal tire force
$F_y$	Lateral tire force
$F_z$	Vertical load
$l_f$	Longitudinal distance between front axle and $CG$
$l_r$	Longitudinal distance between rear axle and $CG$
$NS$	Neutral-steer
$OS$	Over-steer
$R_{nom}$	Nominal tire radius
$US$	Under-steer
$V_x$	Longitudinal velocity
$V_y$	Lateral velocity
$x$	Longitudinal vehicle axis

---

# Index

*CG*-states, 17

accuracy, 34

active safety, 1

autonomous driving, 1

bicycle model, 5, 6

cornering stiffness, 6

drift equilibria, 10

drifting, 3

encoders, 18

external sensors, 14

handling limits, 2

high-level control, 2

inertial measurement unit, 19

intelligent vehicle, 2

Kalman filter, 21

kinematic model, 19

low-level control, 2

measurement model, 53

on-board sensors, 14

process model, 53

stability, 7

stock sensors, 14

tire slip, 4

tire-road interaction, 4

under-steer gradient, 7

vehicle sideslip angle, 13