



SEBASTIAN FRANZ
MICHAEL HOFMANN

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Active Steering Dolly for Long Combination Vehicles

Design of a Real-Time Control Interface for a Steerable Dolly

SEBASTIAN FRANZ
MICHAEL HOFMANN

Department of Applied Mechanics
Division of Vehicle Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

Active Steering Dolly for Long Combination Vehicles
Design of a Real-Time Control Interface for a Steerable Dolly
SEBASTIAN FRANZ
MICHAEL HOFMANN

© SEBASTIAN FRANZ, MICHAEL HOFMANN, 2015

Master's thesis 2015:01
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Dynamics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Some explanation

Chalmers Reproservice
Göteborg, Sweden 2015

Active Steering Dolly for Long Combination Vehicles
Design of a Real-Time Control Interface for a Steerable Dolly
Master's thesis in Automotive Engineering
SEBASTIAN FRANZ
MICHAEL HOFMANN
Department of Applied Mechanics
Division of Vehicle Dynamics
Chalmers University of Technology

ABSTRACT

Keywords: Some stuff, More stuff, Stuff

CONTENTS

Abstract	i
Contents	iii
1 Introduction (3 Seiten)	1
1.1 Purpose	1
1.2 Objectives	1
1.3 Limitations	2
1.4 Structure of this thesis	2
2 Overview (6-7 Seiten)	4
2.1 Status quo	4
2.2 Ongoing research	4
2.3 Market overview for existing solutions	5
2.4 Rapid Control Prototyping	5
3 Steering Model (3 Seiten)	8
3.1 Overview of the model	8
3.2 Input parameters	8
3.3 Real-Time implementation	8
3.4 Interface with Real-Time environment	8
4 Hardware Setup (9 Seiten)	9
4.1 Utilized dolly system	9
4.2 Real-Time Environment	9
4.2.1 CAN-bus extension	10
4.3 Interfaces with dolly	11
4.4 Interfaces with truck	12
4.5 Measurment Setup	12
4.5.1 On-board sensors	12
4.5.2 Inertial measurement unit	12
4.5.3 Arduino Due	13
5 Software Setup (8 Seiten)	15
5.1 Matlab/Simulink environment	15
5.1.1 FMI toolbox	15
5.1.2 dSpace RTI-blockset	15
5.1.3 Volvo Truck Model	15
5.2 ControlDesk monitoring environment	16
5.2.1 Maneuver control	16
5.2.2 Monitoring and logging	16
5.3 Arduino IDE and applications	16

6 Processing Time evaluation (3-4 Seiten)	17
6.1 Background	17
6.2 Measured input delay	17
6.3 Computational delay	17
7 Fault detection and system ability (5 Seiten)	18
7.1 Failure Mode and Effects Analysis (FMEA)	18
7.2 Safety concepts	18
7.3 Maximum capabilities of the system	18
7.4 Warning and state-info system	19
8 Testing (2-3 Seiten)	20
8.1 Overview	20
8.2 Bench-Testing	20
8.2.1 ECU-setup	20
8.2.2 VTM maneuver verification	20
8.2.3 CAN verification	20
8.2.4 Fault detection system verification	20
8.3 Vehicle testing	20
8.3.1 System calibration	20
8.3.2 Actuator tests	20
8.3.3 Algorithm evaluation	20
8.3.4 Sensor testing	20
8.4 Track testing	21
8.4.1 Testmaneuvers	21
8.4.2 Testenvironment AstaZero	21
8.4.3 Testmatrix	21
8.4.4 Test setup and instrumentation	21
9 Discussion (3-4 Seiten)	22
9.1 Results from bench testing	22
9.2 Results from in vehicle testing	22
9.3 Results from on-track testing	22
9.4 Comparrison	22
10 Conclusion (2 Seiten)	23
10.1 Recommendation	23
10.2 Future Work	23
References	24

1 Introduction (3 Seiten)

1.1 Purpose

Heavy goods-transport on the road has constantly increased over the last decades. Coupled with the stricter environmental regulations concerning CO₂-emissions and pollution, the call for more economical transport solution has led to the wider introduction of long combination vehicles. Those truck-trailer combinations have a longer history in geographical areas with low population density, mining and transport within factory sites where rail-road transport is not a viable option but transportation of large volumes and tonnages are called for. The prospects of saving costs on driver's salaries, reduced fuel consumption and decreased costs suggests the introduction of those combinations in other environments as well. Introduction of a new vehicle class leads to many challenges in safety, research and development, and legislation.

The driving behaviour of LCVs is in many ways different to that of standard trucks and needs to be researched in great detail to gain an understanding of the vehicle's dynamic properties, that is equally detailed as it is for other vehicle classes. This will lead to development of better safety and assistance systems and thus reduce threat potential, accidents and fatalities involving this emerging mode of transportation. Different usage patterns of LCVs have to be considered as well, when developing functions for LCVs. For example inner-city use is no prevalent use-case for LCVs, whereas highway safety features and handling properties at higher speeds are prime goals due to high percentage of highway-driving for LCVs. Nevertheless maneuverability for docking is also a development goal.

Besides the technical implementation, socio-economic aspects have to be considered. Legislation has to be adjusted to allow for longer vehicle classes, including new certification processes and driver training. Furthermore infrastructure might have to be modified or reviewed to accommodate the needs and dimensions of extended truck combinations.

The research project in which this thesis is embedded aims to develop an active dolly, meaning that steering will be autonomously conducted by the dolly based on the driving situation at hand and various vehicle parameters (e.g. speed, steering wheel angle). Furtheron braking capabilities are to be implemented to act in a similar fashion as an electronic stability control system (ESC) by creating a yaw-moment countering undesired vehicle movements. This counter-steering will be achieved through wheel-individual brake-application.

This high-level control algorithm will be executed on a rapid-prototyping system which is linked to and controls the dolly. To supply this connection between the hardware and control-algorithm implemented in the modeling-environment Simulink is the main-task of this thesis.

1.2 Objectives

The main-goals that are supposed to be achieved within this thesis' scope of work are:

- supply a software interface for the high-level Simulink control algorithm to be run on a rapid-prototyping system to control the steering system on an actual dolly. The implementation of brake actuation and air-suspended leveling system will be explained in detail, but not carried out in practice.

- set up the physical hardware interface with the dolly; establish a suitable environment connection for the rapid-prototyping system on-board of the dolly
- come up with a measuring solution to determine the processing delays in the sensing system as well as the delays introduced by computation and actuator reaction times. Determine and try to minimize these occurring delays.
- verification of designed systems through different stages of hardware-in-the-loop tests (HIL)
- supply a safety system that continuously monitors the active steering system and triggers necessary warnings.
- supply an interface that allows to continuously determine the system's maximum actuation capabilities depending on the system's current properties (loaded weight, speed, steering angle, yaw-behaviour). Utilize a vehicle dynamics model to find out the critical boundaries for LCV's behaviour.
- prepare on-track testing (road-ready measuring equipment, start-up procedure, repeatable pre-recorded maneuver description for automated testing)

1.3 Limitations

The actual high-level algorithm to compute the desired angle for the dolly's steerable axles is not in the scope of this thesis. Nevertheless to establish an easier insight into the interfaces' parameters, an overview of the structure, in- and outputs of the underlying computational steering model is needed and shall be presented in chapter 3.

The hardware- and low-level control-system of the hydraulic actuators is in place already and thus will not be part of this thesis. It is supplied as turn-key software by the manufacturer and readily available on the dolly's electrical control units (ECU). Still some modifications are necessary to achieve the desired goals. The ECU's software version will be available fully calibrated and parametrized for the dolly at hand and thus provide a reliable working base to build upon.

1.4 Structure of this thesis

In the first two sections of this thesis a brief overview of the legal situation concerning LCV for different countries, the current state of the art and ongoing research in the field of LCV shall be presented (chapter 2). Furthermore an introduction to the model, that will be run on the rapid-prototyping system will be given (chapter 3). Those two chapters are meant to give an introduction into the matter and are mainly based on literature review.

In the succeeding chapters the conducted work will be described in detail. Starting with a description of the utilized hardware-systems and their interconnections in chapter 4, followed by detailing the different software-tools and environments running on those hardware-platforms in chapter 5. In chapter 6 the measuring concepts and theoretical details for the determination of the overall processing delays in the control-chain will be discussed. As at the planned high speeds and great inertia for testing safety is a major concern, extensive safety functions will be

implemented and systematically evaluated. This will be outlined and discussed in chapter 7. After evaluating the safety of the system subsequently testing and validation is conducted and discussed in the following chapter 8.

The work closes with a discussion of the results collected during testing and a conclusion where the authors will try to give recommendation for practical implementation and outline future research work in the field.

2 Overview (6-7 Seiten)

2.1 Status quo

- Verbreitung in anderen Ländern

Sweden between 2003 and 2012: share of 46% for LCV in traveled km [3]

- Führerscheinbestimmungen?

In Europe a person who drives a LCV is required to have the CE drivers license, which allows you to drive a tractor with a mass over 3.5 tonnes with a trailer, whose mass is over 750 kg. To be able to acquire that drivers license the person has to be over 21 years old. **EU-Führerscheinquelle** On top of that, there have been more requirements for the drivers that took part in trials of LCV in European countries. For example for the trials in the Netherlands drivers were required to have the drivers license for at least five years, haven't been in an accident in the last three years and they had to undertake a safe driving training. **Eurocombiquelle**

- Road wear

Road wear is a big concern when it comes to road transport. The main factor of road wear is the axle load of the vehicles. The axle load for a vehicle is calculated as follows:

$$F_{axl} = \frac{m_{tot}}{N_{axl}} \quad (2.1)$$

A standard tractor-semitrailer combination has five or six axles and a maximum mass of 40 tonnes. With equation (2.1) the axle load, if fully loaded, for that combination is 8 tonnes/axle respectively 6.67 tonnes/axle. An A-double-combination, which has ten or eleven axles and a maximum mass of 80 tonnes, also reaches a maximum axle load of 8 tonnes/axle respectively 7.27 tonnes/axle. For a 25.25 m combination **NAME?**, which has nine or ten axles and a maximum mass of 60 tonnes, the maximum axle load is even lower than for the standard combination. The maximum axle load for this case is 6.67 tonnes/axle respectively 6 tonnes/axle.

- besserer Titel fuer Überschrift

In countries of the EU the permitted maximum length of a truck-trailer combination is 18.75 m and its maximum weight is 44 tonnes. There is however the possibility for countries to make exceptions from that rule.[23] For example in Sweden and Finland road trains can be up to 25.25 m long with a maximum weight of 60 tonnes.[24] In several other countries of the EU LCV are allowed on certain roads for testing purposes. Table 2.1 shows the maximum length and weight of LCV in different countries.

2.2 Ongoing research

Zwei Seiten Text produzieren.

- LVC advantages/disadvantages (Baltin)

rearward amplification

Table 2.1: LCV in different countries[24][22][1][9][4]

Country	Max. Length [m]	Max. Weight [t]
Sweden	25.25	60.00
Finland	25.25	60.00
Australia	53.50	132.00
USA(trailers without truck)	26.07	59.86
Canada	36.88	63.50
Mexico	31.00	75.50

wendekreis

überholvorgaenge

- safety aspect

The analysis of crash data from the Swedish Traffic Accident Data Acquisition showed that there is no higher rate of fatal or severe crashes per traveled vehicle km for LCV compared to standard vehicle combinations. [3]

- Environmental aspects

With the use of LCV the environmental impact of goods transportation can be reduce significantly for high loading factors. The fuel consumption is about 15% lower compared to the standard vehicle combinations used in the EU. In the number of trips a reduction of 32% can be seen. This reduction is achieved because two LCV can carry the same amount of goods as three standard vehicle combinations. [2] However, the environmental impact of LCV can become negative if due to a high penetration of LCV goods transportation is shifted from rail to the road. [7]

- development of saftey systems

2.3 Market overview for existing solutions

- Fahrzeugwerk Bernard Krone:

mechanical steering of front axle with adjustable steering ratio; input through movable drawbar.

- Schmitz Cargobull:

front axle steerable (no information how)

2.4 Rapid Control Prototyping

In automotive software development standardized processes are applied to assure structured work. The V-model which derives its name from the characteristic shape constitutes one graphic representations of the sequence of steps towards the finished overall system. It also gives an approximate impression about the level of detail of each of the steps on the vertical axle, as well as chronological progress on the horizontal. A simplified version of this V-model can be seen in figure 2.1. First the user or customer needs are analyzed and put into a schematic formulation

of the system showing the logic dependencies and underlying physical working principles. This is usually done in a semi-standardized fashion utilizing Unified Markup Language (UML) or similar diagrams, preventing non-specific prose to ensure less room for mis-communication and better readability. Technical decisions and limitations are not yet considered. After the logical system and dependencies have been established they will be analyzed and broken down to actual technical system descriptions defining also which subsystem will take over which function, what parts are to be realised as software or hardware (e.g. filtering, safety functions). The different parts of the technical system are then specified in detail, defining interfaces between different software sub-system, operational states and distribution of functions over sub-systems. The fourth step will define the specifications for the software in details for example: data types, control sequences, decision structures, real-time behaviour. Finally the actual design step will implement these specification with regards to the hardware limitations (RAM/ROM, processing power), computation methods, data handling (detailed format, variable types, utilization of parameters or variables). To compile the system from the component-level sub-system a similar hierarchy is used in inverse. The corresponding steps of the project definition and specification phase (left side of the schematic) are supposed to provide test-procedures and goals. If necessary after insufficient test outcomes, another iteration of the previous step will be performed (vertical iterations). A similar sequence of development steps is applied to develop the hardware platform in parallel.[17]

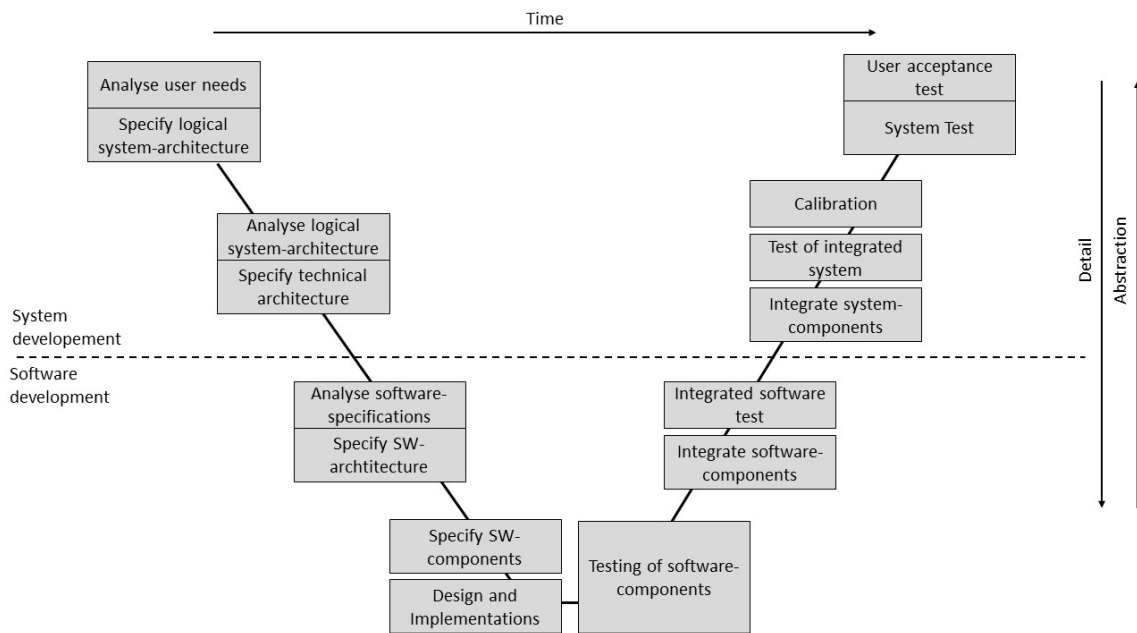


Figure 2.1: Overview for a process in automotive software development after the V-model

The increasing complexity of mechatronic systems and shorter product life-cycles lead to new development methods that made it possible to take 'short-cuts' in the established V-model process. This methods are summarized under the term rapid prototyping - in the case of software functions for mechatronic systems the term rapid control prototyping (RCP) was coined. It is possible to conduct testing and verification with early soft- and hardware versions which are still under development by having the remaining system and environmental influences simulated by the RCP-platform. In utilizing these new methods such as rapid

control prototyping and software/hardware-in-the-loop testing, a tremendous decrease in development time can be achieved. It is also possible to validate specifications early in the process, eliminating possible cost-intensive changes in later stages.[16] For the research project in which this theses is embedded it even opens up the possibility of on-road testing, as it eliminates the low-level development steps of the actual implementation and some of the detailed design work. The implementation of the steering algorithm (see section 3) on a stand-alone ECU, hydraulic actuator control, hardware layout, etc. would by far exceed the dimensions of a research project. Applying rapid control prototyping is the only feasible way to establish a functioning on-road prototype vessel.

- V-model
- real-time capability
- abstract high level model to programm ==> focus on function development and modelling, no low level coding needed
- time critical processes
- extensive lecture notes from [16]

3 Steering Model (3 Seiten)

3.1 Overview of the model

- based on lit from MI-paper
- overview graph of model
- single track model
- filtering of inputs?
- feedback-loop
- start condition
- what can be concluded with the model?

3.2 Input parameters

- what parameters are used as inputs? dimensions, min/max
- explain feedback/inverse path

3.3 Real-Time implementation

- what had to be changed to allow for MABII execution?
- how will feedback loop be handled? item incorporate measurings?
- simulation step size?
- utilized computational method

3.4 Interface with Real-Time environment

- capabilities of system (see section 7.3)
- signals passed forward from environment + restriction
- actuator signals
- signal modification (correct frequency for CAN)
- safety flags/signals

4 Hardware Setup (9 Seiten)

4.1 Utilized dolly system

The utilized dolly by Parator Industri AB (Parator) is equipped with two steerable axles. They are controlled by an after-market solution called ETS Electronic Trailer Steering System (ETS) supplied by V.S.E. Vehicle Systems Engineering B.V. (VSE), of which figure 4.1 gives an overview. Their product includes sensors, ECUs and hydraulic systems which come in a ready-to-mount housing, which is placed on the trailer/dolly. This solution is usually sold as a low-speed active steering system for truck-trailer combinations to provide better maneuverability at low speeds in inner-city areas. Besides electrical power and compressed-air (see "2" in the figure) supply there is no connection with the truck in place. This allows for use with many different truck/trailer original equipment manufacturers (OEM), as no insight into proprietary Controller Area Network communication (CAN) is needed. In the original VSE system the two parameters that influence the actuation of the dolly's steering are vehicle speed ("4") and kingpin-deflection ("3"). This deflection is the angle between the truck and trailer, which is measured by an additional kingpin-angle sensor supplied by VSE and mounted in the eye of the kingpin hub. Furthermore every steering-knuckle of the dolly is equipped with an angle sensor to provide appropriate wheel-individual feedback for the VSE control-system. A diagnosis screen is available in the VSE-unit, which allows for relatively simple set-up, calibration and parametrization to be done.[5]

VSE provides assisted steering up to a speed of 25km/h over which intervention until it reaches zero at 55km/h. At this threshold the steerable axles are locked and thus behave like normal rigid axles. This according to the manufacturer is to ensure stability at higher speeds.[5] Locking the steering at higher speeds leads to a more predictable behaviour for the user and system robustness. However, performance during high speed maneuvers can be improved by uniformly steering the dolly as well.[10] The desired demonstration of the algorithm presented in chapter 3 will as outlined in the introduction to this thesis, include steering at higher speeds, thus a work-around had to be established.

Short overview for the dolly, including:

- mechanical properties in short (weight, turning radius, max. tonnage)
- description of function (brake, steer, countersteer, lock at highspeed)
- difference low \Leftrightarrow high speed
- control system by VSE, diagnosis, display connection with truck
- system overview picture/schematics

4.2 Real-Time Environment

- mechanical properties of box (dimension, currents, mounting points in dolly)
- computational power/limitations

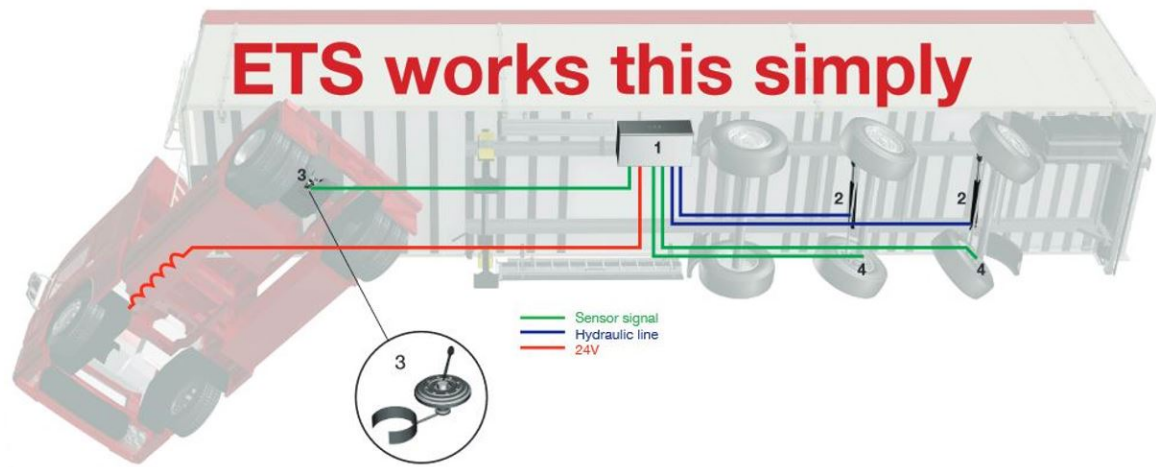


Figure 4.1: Active dolly legacy steering system supplied by VSE[5]

- explain interfaces with truck/dolly (abstract)
- explain technical realisation of HW interface (ZIF)
- explain rapid-prototyping
- robustness
- programming with software ref to 5.1
- runtime interface ref to 5.2

4.2.1 CAN-bus extension

The MABII comes with a preset number of in- and output ports. The maximum number of CAN-buses, that can be connected to the MABII's native controllers is limited to six. As there is no off-the-shelf solution by dSpace for extending the available bus-connections, it was necessary to come up with a gateway solution that allows to patch the needed amount of additional CAN-buses through to the ethernet connection which is also available on the MABII.

To do so, a gateway from CAN-protocol to the standardized User Datagram protocol (UDP) used for communication on ethernet infrastructure was implemented. It is a very light-weight protocol, that is straight-forward to implement and runs well, even with limited processing power on a microcontroller. For future purposes the broadcasting capabilities of UDP might also prove useful, as many nodes could be connected to this CAN-to-UDP gateway for example for visualization on different computers or additional logging outside of the MABII environment. One limitation that was decided on, is to have receiving capabilities only for the MABII to eliminate the need for extensive computation and CAN-matrix handling on the Arduino.

Excursus: The CAN-protocol is the most widespread protocol to allow for communication between different ECUs in the automotive field. Development began at the Robert Bosch GmbH, but is now standardized and enhanced and adjusted for special purposes internationally. Messages are broadcasted by the bus-participants and stamped with their unique identifier, which also doubles as an arbitration token to handle message prioritization. Each bus-participant can listen to all available messages and "only picks from the bus what he needs". The standard message has a size of eight byte à eight bit, transmitted after the identifier and followed by an ending sequence. Hardware-wise CAN-communication relies on only a pair of twisted wires, where a very robust voltage difference signal is transmitted. [6]

By utilizing the UDP-protocol to acquire data into the simulation, the MABII environment's very convenient possibility to incorporate CAN database files (.dbc-file)¹, which is the usual way to exchange information and instructions for CAN-networks is no longer available. To decompose the UDP packets into the original signals it was necessary to implement the function of a .dbc-file in the underlying Simulink-model.

All messages that the CAN-gateway is supposed to handle and forward are put into one UDP frame in succession with respective CAN-identifiers and additional spacer bytes between signals. The UDP frame is broadcasted every time a new CAN-message reaches the gateway on one of its CAN-buses. Messages that were not updated with the incoming CAN-message will be kept at their previous value. This is called for, as UDP is a connection-less protocol without any loss-prevention mechanisms like acknowledgment-handling or retransmission of messages. Holding the values, if not available ensures, that at least some value is available on the bus. UDP also doesn't have native

- Skalierbarkeit
- daten-konvertierung und packaging zu UDP und von UDP zu Daten in CD!!
- Skizze des Systems
- verwendete bibliotheken
- verwendete Controller
- I2C-anbindung und bus-layout

4.3 Interfaces with dolly

To control and measure signals of the dolly a connection to the dollies ETS- and EBS-system had to be established. To achieve this, different kinds of CAN-buses had to be used. That was necessary because the various systems of the dolly use different CAN-protocols. In total five CAN-buses connect the MABII with the dolly. Two for each of the ETS-ECUs and one for the ASF- and EBS-Signal.

BREAKOUTBOX-SCHEMA

¹Correlates physical human readable/understable signals with units to the actual distribution over the different bytes of a CAN-frame. It allows to "decrypt" the information which is available on a CAN-bus and allows to code and send messages in the respective format that the other bus-participants (ECUs, sensors) expect.

- private CANbus with AngleSensor for kingpin
- Vehicle CAN (ISO 11992, connector ISO 7638-2)
 - brake by-wire
 - EBS, ABS
 - sensors for EBS, ABS
 - signals/msgs on vehCAN
- physial interface => diagnosis outlet
- distinguish between HIL/bench-testing and track-testing!!

4.4 Interfaces with truck

- CAN communication with truck
- steering system connction

4.5 Measurment Setup

4.5.1 On-board sensors

Sensor	Measureings	Type	Mounting point
Kingpin angle sensor	Kingpin angle	External	TBD!!!
	Temperature		
EBS	wheel speed	External	wheel brakes
Steering angle sensor	steering angle (first, second axle)	External	steering knuckles

Table 4.1: Available sensors of VSE's ETS

4.5.2 Inertial measurement unit

To determine the processing delays in the control chain (refer to chapter 6) as well as logging implementation for verification and analyses purposes a number of inertial measurement units (IMU) where utilized throughout this thesis' work. The system at hand combined a gyroscope (L3GD20H), and an accelero- and magnetometer (LSM303D) into an IMU put on one circuit board.[13] This one-chip solution allowed for a convenient access to the sensor measurements, as the sensor outputs could be received via Inter-Integrated Circuit-protocol (I²C) which eliminates the need for transducer. Furthermore a high-pass filter is integrated into the IMU's accelerometer, which leads to simpler compensation of the immanent drift.

These units supply the measurements for three axes each at a maximum frequency of 1600Hz for the accelerometer and 757.6Hz for the gyroscope. [20][19]

4.5.3 Arduino Due

To parse the IMU's sensor output (see also section 4.5.2) and convert it into a measurable format and for the implementation of the CAN-to-UDP gateway, it was decided to rely on the cost-efficient and flexible Arduino microcontroller platform. The Arduino platform series was developed by SmartProjects company in Italy and consists of a micro-controller board with several in- and output pins, and a set of programming tools to program said micro-controller (see also section 5.3). Hardware layouts as well as software of this solution are released under an open-source licence, so changes at any level of detail can be made and many other suppliers are now offering solution that tie in with the Arduino. Furthermore this lead to a broad user base and availability of vast supporting resources.

SmartProjects offers various Arduino boards mainly differing in number of communication pins, memory size, clock frequency and physical size. It was decided to utilize the to date most powerful Arduino Due, offering 84 MHz of clock-speed on a ARM Cortex-M3 processor, 512kB flash memory for programm-code and a 96kB working SRAM. It is the first Arduino with 32-bit architecture. This was done in order to allow for future use of this platform in other projects and having enough space for different sub-programs on the controller as well as enough processing power to deal with basic signal filtering and parsing at appropriate speeds.

The Due has a native I²C interface, which was used to gather the measurings from the IMU. It also has a Serial Peripheral Interface interface (SPI) which is needed to access the ethernet controller necessary for the CAN-UDP-gateway (refer to section 4.2.1), serial communication is also handled on hardware level. The availability of these ports in hardware form allow for robust systems and eliminate the need to implement those protocols on software level (bit-banging), which frees up memory resources. In addition to this digital communication possibilities the Due offers an abundance of 54 digital and 12 analog freely configurable I/O-pins. The Due is also the first Arduino to host an on-board CAN-controller, which in this thesis will be used for measurement transfer to the logging system (MABII), eliminating the need for additional hardware for CAN-bus interfacing on the Arduino side.

Besides the power-supply and the IMU-chip a MCP2551 CAN-transceiver by Microchip Technology Inc. was incorporated to take care of the pyhsical layer of CAN-bus communication by converting the digital signals from the Due's CAN-controller to the standardized voltage levels of the CAN. The MCP2551 is capable of different CAN standards and fully ISO-11898 compatible, which makes future use in different environments or as unit for in-vehicle CAN-interfacing feasible.

The Arduino Due was used to determine the delays in the system (see section 6.2) as well as a light-weight solution to quickly read CAN-outputs of the various systems during this thesis' work. This proofed to be an easy debugging solution. Additionally it was used to trigger some error codes to test the developed safety mechanisms (see section 8.2.4).

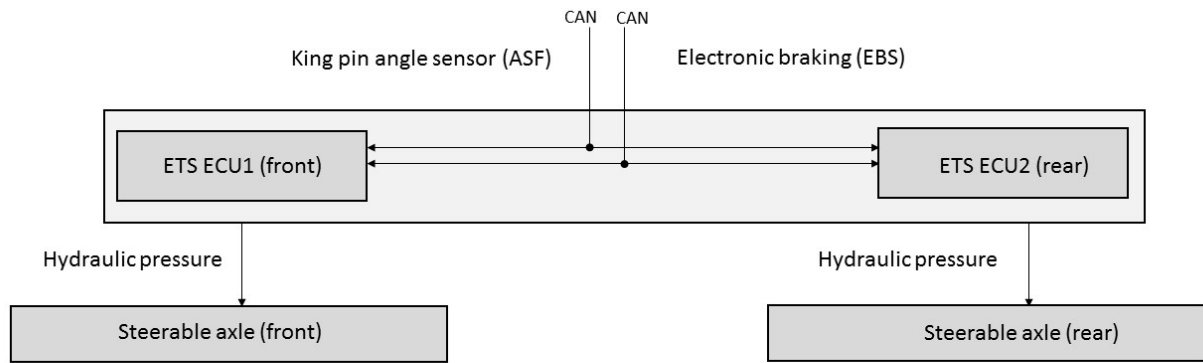


Figure 4.2: System overview for sensors and signal path for ETS legacy system by VSE

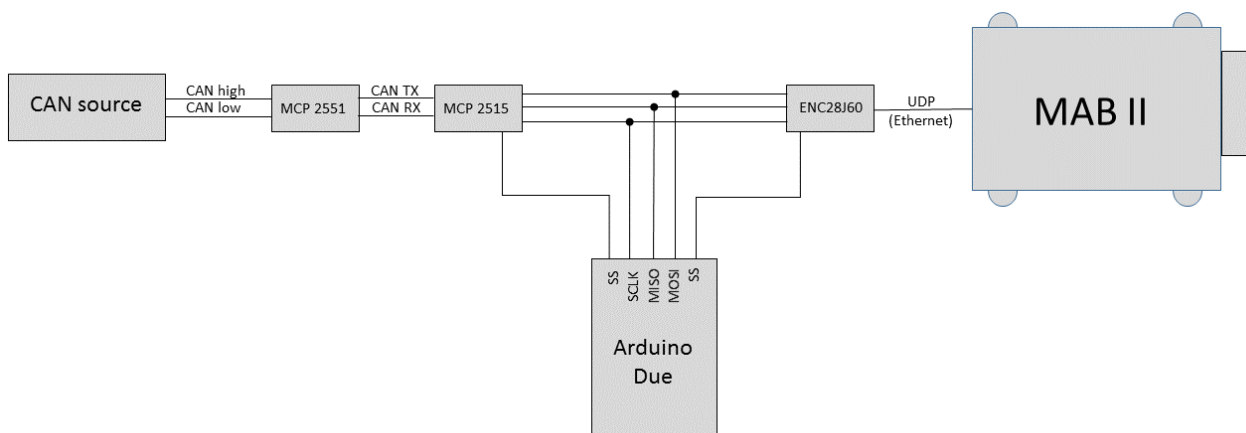


Figure 4.3: Extension of MABII CAN-buses (hardware overview)

5 Software Setup (8 Seiten)

5.1 Matlab/Simulink environment

5.1.1 FMI toolbox

5.1.2 dSpace RTI-blockset

The dSpace RTI-Blockset (Real-time interface) is a plug-in for MATLAB/Simulink that allows you to connect a simulink-model to the different inputs and outputs of the MABII. There are RTI-blocks for CAN, Ethernet, LIN, FPGA and the analog and digital outputs of the MABII. In this project only the RTI CAN MultiMessage Blockset was used. These blocks establish an interface between the physical CAN-Buses of the MABII and the simulink-model running on the MABII. There are four different blocks in this blockset:

In the "GeneralSetup" block the paths of the model root and the destination folder for generated files are set.

In the "ControllerSetup" block first the name of the controller has to be set, then the physical CAN-Bus, that should be used, is set by choosing a module number and a controller number. How the module- and controller numbers have to be set for the different CAN-buses of the MABII is shown in 5.1.

Table 5.1: CAN-layout MABII

CAN	Module number	Controller number	ZIF-Pin CAN-High	ZIF-Pin CAN-Low
CAN1	1	1	c2	c3
CAN2	1	2	b2	b3
CAN3	2	1	B2	B3
CAN4	2	2	A2	A3
CAN5	3	1	P2	P3
CAN6	3	2	N2	N3

After setting the module- and controller number the identifier format has to be set to either standard or extended format, the transceiver type must be chosen between ISO11898-2 and ISO11898-6. ISO11898-2 is used for a high-speed medium access unit and ISO11898-6 for the selective wake-up functionality of a high-speed medium access unit. If needed, a termination resistance of 120 Ohms can be set in the block as well. As a last step the Baud rate of the CAN-bus has to be defined.

In the "MainSetup" block a dbc-file is connected to one of the controller blocks, that were created before. A "ControllerSetup" block can only be connected to one "MainSetup" block at a time. When the dbc-file is loaded in the "MainSetup" block, the different messages and signals of the dbc-file can be chosen as inputs and/or outputs to the simulink-model.

5.1.3 Volvo Truck Model

The Volvo Truck Model (VTM) is a computational framework used within Volvo Trucks to simulate the dynamic behaviours of trucks and combinations. As a library it extends Simulink, where maneuvers, track layout and the trucks kinematic and dynamic properties are linked

together and then computed. This toolbox was used as a base for the simulation of the LVC including the dolly for this project. It is possible to run simulation offline with a predefined maneuver and a given environment as well online where these parameters are fed into the calculation live or as measurements from the real-world environment. For online use all relevant parameters and states can be accessed in Simulink or in case of execution on the MABII through ControlDesk as their respective representation of the Simulink variable.

5.2 ControlDesk monitoring environment

5.2.1 Maneuver control

5.2.2 Monitoring and logging

- data-format
- frequency
- synchronizing over different CANs

5.3 Arduino IDE and applications

The Arduino system provides an integrated development environment (IDE) written in Java, providing cross-platform support. It is used to develop the code as well as compiling the code and subsequently uploading it into the microcontroller via the computers serial interface. Within the IDE it is also possible to load some of the officially supported libraries directly. It conveniently is possible to monitor the serial interface as well, which is the most practical way to monitor and debug code that is executed on the Arduino.

Library name	Purpose	Comment
LSM303	read magnetometer on IMU via I2C	[15]
L3G	read gyro & accelerometer on IMU via I2C	[14]
UIPethernet		[12]
TinyGPSPlus	acquire and parse GPS signal from EM-506 via serial	v0.94b[11]
mcp_can	implement CAN via MCP2515 and MCP2551 via SPI	[8]

Table 5.2: List of utilized libraries on the Arduino platform

6 Processing Time evaluation (3-4 Seiten)

6.1 Background

The desired solution is supposed to operate at any speed. For high speeds a quick processing and transmission time is required to ensure prompt and realtime intervention of the control system based on the measured input signals. If the delays induced by the different components in the complete system are known or can be estimated, they can be compensated for in the steering-algorithm running on the rapid-prototyping system.

The dolly is equipped with a system to determine the deflection angle of the drawbar. This is measured at the kingpin. The sensor's raw signal is then parsed and filtered in a low-level system which feeds the filtered signals to the CAN-bus, where it is picked up by the MABII. The filtering operation takes a certain time and thus induces a delay.

Furthermore the model running on the rapid-prototyping system needs a certain time to calculate the current desired steering angle for the dollies' wheels. This has to be determined as well. The steering mechanisms on the dolly are also a delay-inducer due to the inertia in the hydro-mechanic system. This is as well unavoidable, but when measured can as well be compensated for.

In this chapter the accuracy and delays in the Arduino measuring set-up, see also section 4.5, is outlined as well. Which is important for determining the hardware-induced delays.

6.2 Measured input delay

- critical transmission time CAN (worst case)
- I²C delay
- IMU-frequency
- msg-frequency from vehicle/dolly

Two bus-protocols are used to aquire the data measured from the IMU and transmit to the MABII. The data0 is send via I²C-protocol to the Arduino Due, where it is processed and then send out (to the MABII) on a private CAN-bus.

6.3 Computational delay


- filtering on arduino
- averaging
- drift correction

7 Fault detection and system ability (5 Seiten)

7.1 Failure Mode and Effects Analysis (FMEA)

FMEAs were first used by NASA in the apollo project in the 1960s. A FMEA is used to detect possible failures before they appear. Therefore the FMEA is done in a early stage of a project in order to be able to take the results of the FMEA into consideration when developing a system. In the process every possible failure of a system are taken into consideration.

In a first step a block diagram of the system with all of its inputs, outputs and subsystems was created to gain a complete understanding of the system. Every subsystem was broken down to the lowest level and for those subsystems block diagrams with all the components, inputs and outputs were created as well.

Starting from the block diagrams all potential failure modes were determined for every component of the system respectively subsystem. As a next step the potential effects of these failure modes were determined and the severity of these effects were evaluated on a scale from one to ten, with 1 being the lowest severity and ten being the highest severity. For the evaluation a table that shows how different severities correlate with the numbers was used.  Following this, potential causes for every failure mode were defined. Then the probability of occurrence for each cause was evaluated, also using a scale from one to ten. After that the current control mechanisms, that detect the failure when it should appear, were listed for each failure mechanism and the detectability of the failure mechanism was evaluated using a scale from one to ten. After the severity, probability and detectability were evaluated, a risk priority number is calculated as follows:

$$RPN = severity * probability * detectability$$

This risk priority number is used to identify the failure mechanisms that need to be addressed. In table 7.1 it is shown how the risk priority number For the failure mechanisms that need to

Table 7.1: Risk priority number

RPN	Action
$0 < RPN < 40$	No action needed
$40 < RPN < 100$	Decide if action is needed after review
$RPN > 100$	Action needed

be addressed actions were decided, that either lower the severity, probability or detectability of them or more than one of those factors. After the performance of these actions the severity, probability and detectability were evaluated again and a new RPN was calculated. If the RPN still is to high a iteration of the process was done.

7.2 Safety concepts

7.3 Maximum capabilities of the system

- give indicator of maximum angle/angle rate

- describe "algorithm"/lookuptable
- explain underlying physical correlation (ref to MA from)

7.4 Warning and state-info system

- warnings from dolly ECU

In the original state, the ETS-ECU monitors all safety critical components of the dolly. In case of a malfunction, an error code is sent via the CAN-Bus and the error message is visible on the diagnose displays of the dolly. Due to the required modification of the ASF- and speed-signal, the inbuilt supervision of the ETS-ECU is no longer fully functioning. Therefore a additional supervision of the system has to be implemented in the Simulink-model.

- warnings from EBS
- warnings from vehicle
- 'own' error codes and warnings (e.g. logging, MABII related, arduino-IMU related)

8 Testing (2-3 Seiten)

8.1 Overview

In different states of the project different kind of tests were performed. As a first step bench tests were done to verify the developed software. Following that, several tests on the actual dolly, with the dolly in standstill and no trailers connected, were made.

8.2 Bench-Testing

8.2.1 ECU-setup

8.2.2 VTM maneuver verification

- include simulation results for maneuver of controller being run inside VTM
- perhaps include some picture of the video-output, with the actuated truck
- plots for maneuver (steering input, speed over time)

8.2.3 CAN verification

- check output from ASF with arduino
- check ECU CANs, make sure they differ in output

8.2.4 Fault detection system verification

- send faulty inputs
- show plot of faulty input and system reaction

8.3 Vehicle testing

8.3.1 System calibration

8.3.2 Actuator tests

8.3.3 Algorithm evaluation

8.3.4 Sensor testing

- test angle sensor offline
- zero IMUs, make sure paired sensors show the same outputs

8.4 Track testing

8.4.1 Testmaneuvers

- lit research for standard maneuvers
- sine-wave
- outline critical parts of maneuver
- figure with SA over time
- expected behaviour from simulation

8.4.2 Testenvironment AstaZero

- overview of AZ
- map in appendix?
- restrictions of environment

8.4.3 Testmatrix

- checklist for launch
- parameters that very varied
- different runs
- planned maneuvers

8.4.4 Test setup and instrumentation

- detailed description of placement of sensors, wiring, logging-PC

9 Discussion (3-4 Seiten)

9.1 Results from bench testing

- lessons-learned?
- adaptation for future projects
- what was taken over for further tests?
- what couldnt be simultaед?

9.2 Results from in vehicle testing

- measured delay
- CAN-analysis
- robustness?
- reliability of safety features

9.3 Results from on-track testing

- what went good/bad?
- measuring problems?
- short descr. of accumulated data
- discussion of noise, filtering, systematic errors

9.4 Comparrison

- VTM \Leftrightarrow testing

10 Conclusion (2 Seiten)

10.1 Recommendation

10.2 Future Work

References

- [1] U. E. P. Agency. *Longer combination vehicles. a glance at clean freight strategies*. 2010. URL: <http://www.epa.gov/smartway/forpartners/documents/trucks/techsheets-truck/420f10053.pdf> (visited on 02/09/2015).
- [2] H. Backman and R. Nordström. Improved performance of European long haulage transport. *Transport Research Institute (TfK), Stockholm* (2002).
- [3] A. Bálint et al. “Correlation between truck combination length and injury risk”. *Australasian College of Road Safety Conference, 2013, Adelaide, South Australia, Australia*. 2013.
- [4] T. R. Board. *REVIEW OF MEXICAN EXPERIENCE WITH THE REGULATION OF LARGE COMMERCIAL MOTOR VEHICLES*. Oct. 2011. URL: http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rrd_362.pdf (visited on 02/09/2015).
- [5] V. V. S. E. B.V. *Product information ETS for trailers*. 2014.
- [6] S. Corrigan. Introduction to the controller area network (CAN). *Texas Instrument, Application Report* (2008).
- [7] C. Doll et al. Long-Term Climate Impacts of the Introduction of Mega-Trucks, Study to the Community of European Railways and Infrastructure Companies (CER). *Fraunhofer ISI (Study Co-Ordinator, Karlsruhe) TRT (Milan), NESTEAR (Gentilly), Fraunhofer-ATL (Nuremberg), Fraunhofer-IML (Dortmund), Karlsruhe* (2009).
- [8] GitHub. *Seed-Studio/CAN_BUSshield*. 2015. URL: https://github.com/Seed-Studio/CAN_BUS_Shield/.
- [9] A. Infrastructure and Transportation. *Highway Provider View of Long Combination Vehicles*. Mar. 2005. URL: <http://www.transportation.alberta.ca/Content/docType59/Production/FAQs%28US%29.pdf> (visited on 02/09/2015).
- [10] M. S. Kati et al. “Performance Improvement for A-double Combination by introducing a Smart Dolly”. *13th International Heavy Vehicle Transport Technology Symposium, San Luis, Argentina*. 2014.
- [11] mikalhart/*TinyGPSPlus*. 2013. URL: <https://github.com/mikalhart/TinyGPSPlus/>.
- [12] ntruchsess/*arduino_uip*. 2014. URL: https://github.com/ntruchsess/arduino_uip.
- [13] PololuCorporation. *AltIMU-10 v4 Gyro, Accelerometer, Compass, and Altimeter (L3GD20H, LSM303D, and LPS25H Carrier)*. 2015 (accessed February 12, 2015). URL: <https://www.pololu.com/product/2470/>.
- [14] pololu/*l3g-arduino*. 2015. URL: <https://github.com/pololu/l3g-arduino>.
- [15] pololu/*lsm303-arduino*. 2015. URL: <https://github.com/pololu/lsm303-arduino>.
- [16] “Rapid Control Prototyping”. German. *Rapid Control Prototyping*. Springer Berlin Heidelberg, 2006, pp. 295–318. ISBN: 978-3-540-29524-2. DOI: 10.1007/3-540-29525-9_7. URL: http://dx.doi.org/10.1007/3-540-29525-9_7.
- [17] J. Schäuffele and T. Zurawka. *Automotive software engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. Springer-Verlag, 2012.
- [18] P. Steenhof, C. Woudsma, and E. Sparling. Greenhouse gas emissions and the surface transport of freight in Canada. *Transportation Research Part D: Transport and Environment* **11.5** (2006), 369–376. ISSN: 1361-9209. DOI: <http://dx.doi.org/10.1016/j.trd.2006.07.003>. URL: <http://www.sciencedirect.com/science/article/pii/S1361920906000411>.

- [19] STMicroelectronics. *L3GD20H - MEMS motion sensor: three-axis digital output gyroscope: Datasheet - production data*. 2013.
- [20] STMicroelectronics. *LSM303D - Ultra compact high performance e-Compass 3D accelerometer and 3D magnetometer module: Datasheet — preliminary data*. 2012.
- [21] K. Tindell, H. Hansson, and A. J. Wellings. “Analysing real-time communications: controller area network (CAN)”. *Real-Time Systems Symposium, 1994., Proceedings*. IEEE. 1994, pp. 259–263.
- [22] D. of Transport and M. Roads. *Guideline for Multi-combination Vehicles in Queensland*. July 2013. URL: http://www.tmr.qld.gov.au/~media/busind/Heavyvehicles/guidepermits/Guideline_mcv_form1.pdf (visited on 02/09/2015).
- [23] T. C. O. T. E. UNION. *Council Directive 96/53/EC*. July 25, 1996. URL: <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:31996L0053&from=en> (visited on 02/05/2015).
- [24] Vägverket. *Weight and Dimensions for Road Traffic*. URL: <http://www.unece.org/fileadmin/DAM/trans/wp24/wp24-presentations/documents/pres08-04.pdf> (visited on 02/05/2015).