Design and implementation of control and actuation for an over-actuated research vehicle

Petter Tomner

Vehicle Dynamics
Aeronautical and Vehicle Engineering
Royal Institute of Technology

Master Thesis

ISSN 1651-7660
TRITA-AVE 2015:04

Postal address	***Visiting Address***	***Telephone***	***Telefax***	***Internet***
KTH Vehicle Dynamics SE-100 44 Stockholm, Sweden	Teknikringen 8 Stockholm	+46 8 790 6000	+46 8 790 9290	www.kth.se

Abstract

The RCV is a four wheel drive and steer electrical vehicle developed and built by the Transport Lab at KTH Royal Institute of Technology. It is fully steer-by-wire and each wheel can be individually controlled, both with regard to steering angle, as well as camber and driving torque.

The RCV is planned to be used as a common platform for different fields of research, as a rolling laboratory to implement and evaluate research with.

In this report the specification and functionalities of the RCV are reviewed and its data collection capabilities are validated and evaluated through classic vehicle dynamic analyses such as circle tests, step steer and roll out tests. Also, some more experimental functionalities such as simple torque vectoring, toe sweep and steering by joystick, as proof of concept of the RCV as a research and prototyping platform.

Finally, some suggestions for further developments for the RCV platform are presented.

Contents

1 Introduction

As of recent developments in the vehicle industry towards so called alternative fuels and electric propulsion, there is a need for alternatives to the classic vehicle powertrain in which a combustion engine transfers power to the wheel through clutch, gearbox, drive shaft, etc.

One alternative power-train approach is the autonomous corner module (ACM), in which the wheel together with steering actuation, suspension and gear are integrated in one modular unit mounted in the four corners of the vehicle. Propulsion is preferably electric and the actuation steer-by-wire. This allows for four wheel drive and four wheel steering, controlling the forward speed and yaw rate of the vehicle, as well as lateral movement indirectly, by simply turning all the wheels in the same direction. This is commonly referred to as an over actuated system. As each wheel can be steered individually, parameters like toe in and out, Ackermann angle, can be tuned dynamically by the actuators, providing a greater number of degrees of freedom compared to a classical vehicle set-up. Further more, due to the distributed wheel hub motors, the torque on and rotational speed of each wheel can be chosen arbitrary, as there is no relational restrictions from differential couplings, common axis, etc.

Given a four wheel drive and steer vehicle, it has a theoretical advantage with regard to vehicle dynamic characteristics compared to one axis drive one axis steer ditto.

This report evaluates and explores some of the characteristics of such an over-actuated vehicle, the Research Concept Vehicle (RCV) of KTH Transport Lab.

1.1 Outline of the thesis

This report is a documentation on some of the author's work on the RCV as well as documentation and reviews of other's work on the RCV.

Firstly, in the Theory section, fundamental definitions and vehicle models are introduced.

Secondly, in the Implementation section, the RCV and its subsystem are presented.

Thirdly, in the Prior work section, prior work on the RCV, as well as some similar research done at other universities, is reviewed.

Fourthly, in the Experimental results section, data from real world test runs are analysed, evaluated and presented.

Lastly, in the Conclusion and suggestion for further developments sections, concluding remarks and recommendations are presented.

In the appendix, some RCV contextual relevant know-how topics are covered.

2 Theory

The coordinate system in this report is under ISO 8855:2011. Refer to Figure 1.

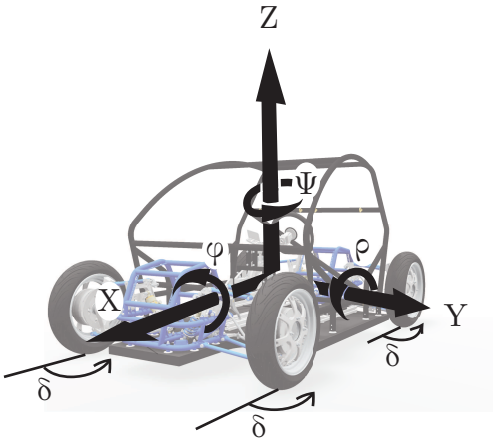

Figure 1: The body fixed reference system. Origo is Centre of Mass. ρ is pitch, Ψ is yaw, ϕ is roll. δ is the wheel turning angle, which is defined to be in the same direction for all four wheels.

2.1 General definitions

Over actuation An over actuated system could be defined as a system in which there are more control variables than relevant independently controlled generalized coordinates.

E.g. a three degrees of freedom, longitudinal, lateral and yaw, ship, with rudder and a fixed axis propeller is able control longitudinal movement and, but not independently of longitudinal movement, yaw, and not lateral movement at all, and is thus under actuated. A ship with two propellers such that each could be turned to point towards any point of the horizon and has individual torque distribution, is able to control lateral and longitudinal movements and yaw, and is thus fully actuated. The same ship with four propellers would be over actuated.

Introducing some relevant abbreviations for actuation; Four wheel steer is abbreviated 4WS, two wheel steer 2WS, four/two wheel drive 4WD/2WD, or any combinations of these, e.g. 4WDS, 2WS4WD, etc.

2.2 Ackermann steering geometry

To avoid skidding while cornering the wheels have to be aligned such that they follow their specific trajectory around the turning point. Given vehicle base W and length between front and rear axes L and the equivalent turning radius R with positive direction to the left of the vehicle, the steering angle δ_i of the wheels is given by:

$$\delta_1 = \arctan \frac{L}{R - \frac{W}{2}} \tag{1}$$

$$\delta_2 = \arctan \frac{L}{R + \frac{W}{2}} \tag{2}$$

where δ_1 is the steering angle of the front left wheel and δ_2 is ditto for the front right wheel.

Analogous, the geometric relations for 4WS is compiled, with a turning point perpendicular to the centre of the vehicle.

$$\delta_1 = \arctan \frac{L}{2(R - \frac{W}{2})} \tag{3}$$

$$\delta_2 = \arctan \frac{L}{2(R + \frac{W}{2})} \tag{4}$$

$$\delta_3 = -\arctan \frac{L}{2(R - \frac{W}{2})} \tag{5}$$

$$\delta_4 = -\arctan \frac{L}{2(R + \frac{W}{2})} \tag{6}$$

where δ_3 is the steering angle of the rear left wheel and δ_4 is ditto for the rear right wheel.

In effect, a vehicle with symmetric 4WS has $\frac{1}{2}$ as long minimum turning radius as it would with only two wheel steering.

Analogous, with the 4WS geometries, the turning point can be arbitrary moved along a line perpendicular to the vehicle's longitudinal direction as well as parallel. The vehicle's rotational point T as a quota of the distance of a point on a line from rear to front steering axis through the centre of the vehicle, and the total distance between the axes, give the following geometric relationships:

$$\delta_1 = \arctan \frac{L \cdot (1 - T)}{R - \frac{W}{2}} \tag{7}$$

$$\delta_2 = \arctan \frac{L \cdot (1 - T)}{R + \frac{W}{2}} \tag{8}$$

$$\delta_3 = -\arctan \frac{L \cdot T}{R - \frac{W}{2}} \tag{9}$$

$$\delta_4 = -\arctan\frac{L \cdot T}{R + \frac{W}{2}} \tag{10}$$

The equivalent turning radius R is given by the relationship:

$$R = \frac{L}{\tan\Delta} \tag{11}$$

, where Δ is some steering wheel input. This allows for a similar turning radius, given a steering wheel angle, for both four wheel steering and movable turning point steering, as if the vehicle would have been two wheel steered. This is most desirable when, e.g., the turning point position is programmed to be speed dependent, as the driver do not have to compensate when the vehicle accelerates in a corner. Refer to Figure 2.

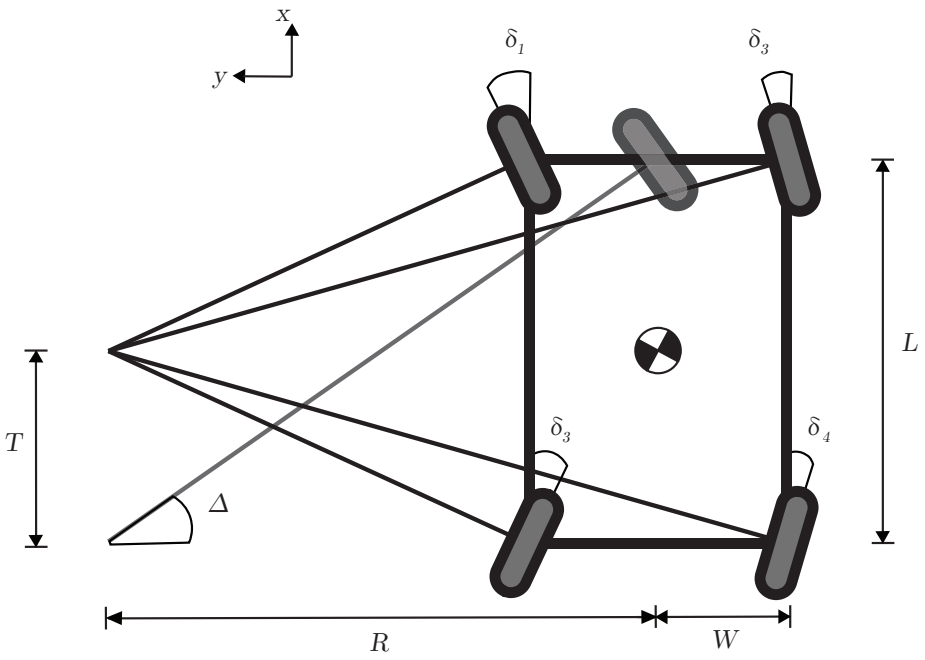

Figure 2: Schematic figure of the 4WS case of Ackermann steering geometry. Equivalent turning radius angle Δ (also known as the Ackermann angle) seen from imaginary middle wheel.

In a more special case, the vehicle has a heading that differs from the direction of travel, e.g. with the parallel steering angle δ_p common to all the four wheels. From this state, an Ackerman steering could be superimposed allowing for skidding free rotation. These steering angles are used when "crab steering" with a joystick.

2.3 Four wheel steer bicycle model

One common way [1, ch.7.1] to model a four wheeled vehicle is to approximate it as a two wheel vehicle – hence, "bicycle model", refer to Figure 3.

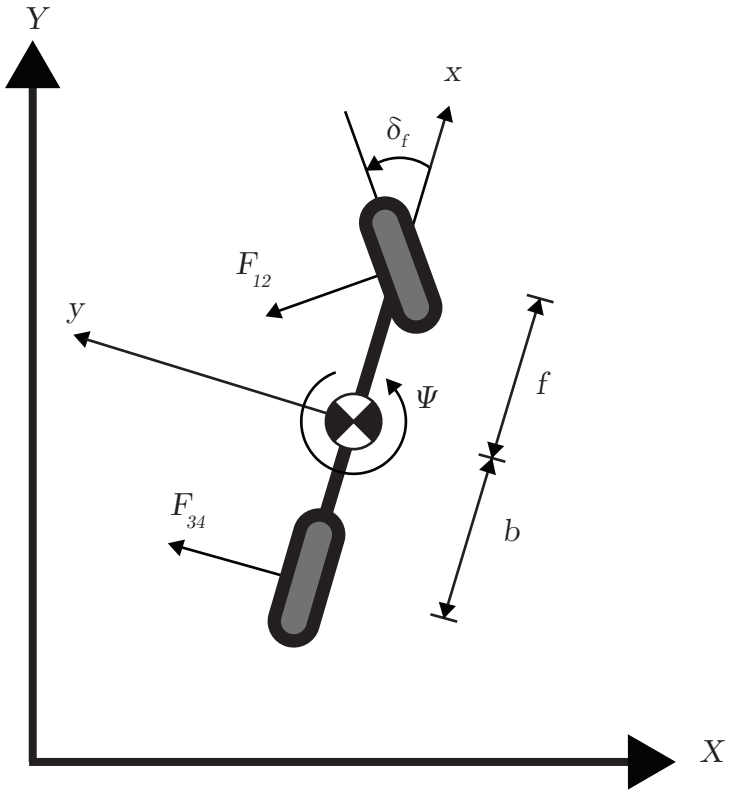

Figure 3: The steering angle δ_f is also the Ackermann angle.

The force balance on the COG in vehicle coordinates is as follows:

$$x \uparrow: m(\ddot{x} - \dot{\psi}\dot{y}) = -F_{12}\sin\delta_f \tag{12}$$

$$y \leftarrow: m(\ddot{y} + \dot{\psi}\dot{x}) = F_{34} + F_{12}\cos\delta_f \tag{13}$$

$$z \circlearrowleft: J_z\ddot{\Psi} = fF_{12}\cos\delta_f - bF_{34} \tag{14}$$

where m is the vehicle mass,J_z is the moment of inertia around the z-axis. Introducing *cornering stiffness* C:

$$F_{12} = -C_{12}\alpha_{12} \tag{15}$$

$$F_{34} = -C_{34}\alpha_{34} \tag{16}$$

where α is the angle between the wheels heading and the path of its centre, as given by:

$$\alpha_{12} = \arctan \frac{\dot{y} + \dot{\Psi} f}{\dot{x}} - \delta_f \tag{17}$$

$$\alpha_{34} = \arctan \frac{\dot{y} - \dot{\Psi} b}{\dot{x}} \tag{18}$$

(Note that δ is the angle between the heading of the vehicle and the heading of the wheel, and α is in other words the angle between where the wheel is pointing and where it is going). Assuming small angles and no forward acceleration; $a_x = 0$ and $\delta, \alpha_{12}, \alpha_{34} \ll 1$, which gives $\cos\delta \approx \delta$ and $\sin\delta \approx 0$,

and by changing notation from $\dot{x}$ to v_x, Equation (12 – 18) translates to the following matrix form:

$$\begin{pmatrix} mD + \frac{C_{12}+C_{34}}{v_x} & mv_x + \frac{fC_{12}-bC_{34}}{v_x} \\ \frac{fC_{12}-bC_{34}}{v_x} & J_z D + \frac{f^2 C_{12}+b^2 C_{34}}{v_x} \end{pmatrix} \begin{pmatrix} v_y \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} C_{12} \\ fC_{12} \end{pmatrix} \delta_f \tag{19}$$

where D is the derivative operator.

It is possible to approximate a 4WS vehicle with bicycle model, refer to Figure 4.

The force balancing equations now become:

$$x \uparrow: m(\ddot{x} - \dot{\psi}\dot{y}) = -F_{12}\sin\delta_f - F_{34}\sin\delta_r \tag{20}$$

$$y \leftarrow: m(\ddot{y} + \dot{\psi}\dot{x}) = F_{34}\cos\delta_r + F_{12}\cos\delta_f \tag{21}$$

$$z \circlearrowleft: J_z\ddot{\Psi} = fF_{12}\cos\delta_f - bF_{34}\cos\delta_r \tag{22}$$

With a new definition for the rear wheel slip angle α_{34}:

$$\alpha_{34} = \arctan \frac{\dot{y} - \dot{\Psi} b}{\dot{x}} - \delta_r \tag{23}$$

,the matrix form becomes:

$$\begin{pmatrix} mD + \frac{C_{12}+C_{34}}{v_x} & mv_x + \frac{fC_{12}-bC_{34}}{v_x} \\ \frac{fC_{12}-bC_{34}}{v_x} & J_z D + \frac{f^2 C_{12}+b^2 C_{34}}{v_x} \end{pmatrix} \begin{pmatrix} v_y \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} C_{12} & C_{34} \\ fC_{12} & -bC_{34} \end{pmatrix} \begin{pmatrix} \delta_f \\ \delta_r \end{pmatrix} \tag{24}$$

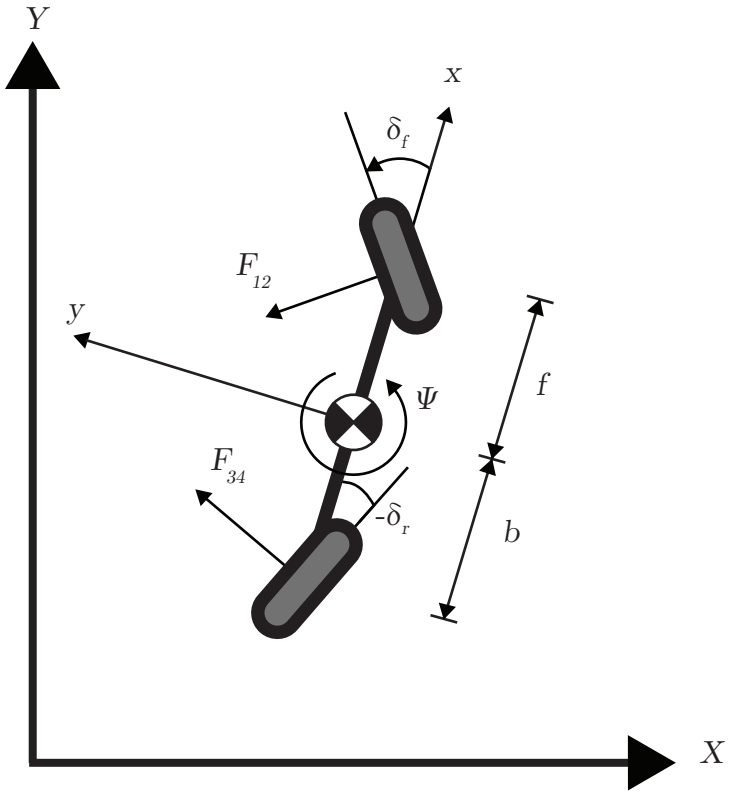

Figure 4: N.B: Turning the rear (or front) wheel to the right is a negative angle.

2.4 Four wheel steer and drive bicycle model

A 4WD vehicle with individual torque allocation on each wheel can generate a arbitrary body torque M.

Introducing F_y as the sum of the y-components of the forces which generate M, such that:

$$M = \vec{e_z} \cdot \sum_{i=1}^{4} \vec{F_i} \times \vec{r_i} \tag{25}$$

$$F_y = \vec{e_y} \cdot \sum_{i=1}^{4} \vec{F_i} \tag{26}$$

where $\vec{r_i}$ is the distance vector from the centre of mass to the i:th wheel contact point, $\vec{F_i}$ is the road force acting on the wheel contact point and $\vec{e_j}$ is the j:th axis normalized unit vector.

Assuming that the torque M is generated in such a way the the sum of the forces in the x-direction is zero, thus $a_x = 0$, the wheel motor torque allocation can be introduced in the bicycle model.

Given:

$$F_x = \vec{e_x} \cdot \sum_{i=1}^{4} \vec{F_i} = 0 \tag{27}$$

and $\delta \ll 1$; the force balance equations become:

$$x \uparrow: m(\ddot{x} - \dot{\psi}\dot{y}) = -F_{12} \sin \delta_f - F_{34} \sin \delta_r \tag{28}$$

,which approximately zeros out, and for the other axes:

$$y \leftarrow: m(\ddot{y} + \dot{\psi}\dot{x}) = F_{34} \cos \delta_r + F_{12} \cos \delta_f + F_y \tag{29}$$

$$z \circlearrowleft: J_z \ddot{\Psi} = f F_{12} \cos \delta_f - b F_{34} \cos \delta_r + M \tag{30}$$

which gives the matrix:

$$\mathbf{Ax} = \begin{pmatrix} C_{12} & C_{34} \\ f C_{12} & -b C_{34} \end{pmatrix} \begin{pmatrix} \delta_f \\ \delta_r \end{pmatrix} + \begin{pmatrix} F_y \\ M \end{pmatrix} \tag{31}$$

where $\mathbf{A}$ and $\mathbf{x}$ are identical to the left-hand side matrix and vector in the bicycle models presented earlier above. A simplified implementation of this model, *simple torque vectoring*, is presented in subsection 3.2.2, paragraph "Wheel hub motor torque allocation".

3 Implementations

With interdisciplinary studies and research in mind the KTH of Transport Lab built a modular vehicle, the Research Concept Vehicle (RCV), as a common platform for different departments and fields of research.

In the following section, the main characteristics of the RCV, and in the subsequent section, the controller software of the RCV, are reviewed.

3.1 Main RCV specification

The RCV is a two seat 4-wheel steer-by-wire electric vehicle.

It has two batteries; one feeding the wheel hub motors and one lower voltage auxiliary battery for the other systems. The motor battery is of lithium-polymer type, with 14 cells, made by Dow Koham (prod.nr.:SLPB100216216H), in series providing a peak current of 400 A at a nominal potential of 52 V. The operating potential is 38 to 58.5 V. A commercial BestTechPower HCX-D131 circuit protection is used on the main battery. It is rated for 50 to 80 A continuous current.

The auxiliary battery is a commercial 24 V 10 Ah Lithium iron phosphate battery (Celltech). The main battery charges the auxiliary battery through a DCDC converter.

Each wheel has a 1.8 kW Heinzmann PRA-230 wheel hub motor, of permanent magnet synchronous type, with a stall torque at 150 Nm and a rated maximum speed of 520 rpm, giving a maximum vehicle speed of approximate 55 $^{km}/_{h}$ with a two person load depending on tire radius and battery circuit power cap.

The wheels are mounted in left-right pair on a frame. Each wheel is controlled in regard of steering and camber by linear electric actuators, mounted on the frame. The camber is actuated by a Thomson Linear P-264 and the steering actuated by a Thomson Linear Max Jac.

Two frames are mounted on the vehicle, on which the steering and suspension is mounted. Both frames are identical and are easily replaceable with the modular approach.

All components, except the hydraulic brakes, are by-wire, refer to Figure 5 for an overview picture. A detail review of the different subsystems follows.

3.1.1 RCV electrical control units

The main component of the vehicle's electrical control system is the dSPACE 1401/1501 MicroAutoBox (MABX) computer. The MABX processes all signals from actuators, motors and driver inputs, then acting accordingly, sending orders to the different parts of the system.

To connect the different parts of the system, two buses of Controller Area Network (CAN), are used. One bus is assigned to the motor drivers, while the rest of the systems share the other bus. This due to technical limitations, the

Figure 5: A picture of the RCV driving on the KTH Campus. Author in driver's seat.

motor controllers being able to handle not as high baud rate on the CAN as the other components. The MABX is connected on both buses.

Some components though, explicitly the gas pedal, horn, tie rod force sensors and the instrument board, are connected to the MABX through ADC and bit I/O.

The actuators are controlled by motor controllers through the CAN buses. Nine Maxon Motor EPOS 70/10 controllers are used for the 4 camber, 1 steering wheel and 4 wheel steering actuators, and 4 Kollmorgen ACD 4805 controllers are used for the 4 wheel hub motors.

MicroAutoBox The MABX is a stand alone computer supposedly designed with fast idea-to-prototype development in mind. It has a PowerPC processor, a generous set of analogue to digital and digital to analogue converters and bit I/O, two CAN connections and one RS232 connection, along with a Local Interconnect Network (LIN) connection, etc.

The processor is a Motorola/IBM PPC603 at 200 MHz together with 2 MB RAM. The device supports timer and hardware interrupts and is thus multi tasking capable.

Using a proprietary serial interface, in it self undocumented and unspecified, the MABX can be connected to a x86 compatible PC running Microsoft Windows together with dSPACE's ControlDesk software.

Through ControlDesk it is possible to directly read and write to the memory

of the MABX during operation and with the software's standard tools it is trivial to make an intuitive and user friendly graphical user interface (GUI). E.g., an interface that reads the speed or current of a wheel motor stored in a memory block in the MABX and then shows it on the PC screen, or allows the PC operator to change variables in the MABX's memory, through a GUI of virtual buttons, levers, etc.

The software is able to log data and export it in various formats, notably Matlab's .mat-format. Data acquisition for logging purposes as well other reading and writing operations on the MABX memory through ControlDesk, is managed by a background task running on the MABX.

ControlDesk also functions as the device programmer, with compiled binaries either uploaded directly into the working memory of the MABX, or flashed to the persistent memory for stand alone booting.

Coding of the main program running on the MABX is done in either C or in MathWorks' Simulink and then converted to C by Simulink's pre-compiler. Due to the low level nature of C programming, Simulink is the language of choice, which, thanks to an extensive high level hardware library provided by dSPACE and the graphical nature of Simulink, makes coding accessible for less experienced programmers and shortens the start up time for a programmer new to a project, as well as the overall development time, which is suitable for the academic interdisciplinary nature of the RCV.

Controller for the camber, steering wheel and steering actuators The EPOS 70/10 is a motor controller made for smaller motors at a nominal peak current of 25 A and a continues current of 10 A at 70 V. The controller of the camber and steering works in positional mode, meaning that an internal PI-controller is given a reference value, in the case of the RCV through CAN from the MABX. The positional controller error is then calculated from the reference value and the positional value given by encoders connected to a Hall sensor on the actuators. The output of positional controller is then used by another internal PI-controller, controlling electrical current forcing the actuator into the desired position.

The controller of the steering wheel's motor, which is used for haptic feedback functionality, works in current mode, meaning that a reference current is provided directly by the MABX to an internal PI-controller. The motor position is used as a measurement of steering wheel angle and forwarded to the MABX.

As the Hall sensors only measures change, not position, the motor controller needs to keep track of the motors actual position by dead reckoning. To do this, it need to know the initial position of the actuator at power up. The initial position is determined by 'homing'; forcing the actuators to its inner most position, which is then designated position value zero. The haptic feedback motor in the steering wheel has no end position, and therefore can't home. In this case, the initial position is always set as zero at power up and it is up to the operators to align the steering wheel to the right position.

Table 1: Steering and camber actuators. Actuator data[3][4]

Max Jac	Stroke length, 100 mm Max current, 4 A Type, ball screw Max static load, 100 - 350 N Max dynamic load, 800 N
P-264	Stroke length, 150 mm Max current, ca. 8 A Type, acme screw Max dynamic load, 3375 N Max staftic load, 6750 N

Wheel hub motor controller Four Kollmorgen ACD 4805 motor controllers are used to feed the wheel hub motors of the RCV. The motor controller utilises pulse width modulation (PWM) to mimic a sinusoidal three phase alternating current.

The controller is connected to the main 52 V battery and is rated for a power output of 9.3 kVA for 2 minutes and 3.8 kVA for one hour continues use.

The four controllers are connected to a common CAN bus together with the MABX at a bitrate of 250 kbit/s. The controller support CAN speeds of up to 500 kbit/s, which is lower than the CAN standard's maximum bitrate of 1000 kbit/s. Therefore the two CAN buses on the RCV are run at different rates, as the EPOS 70/10 controllers connected to the other bus, together with the MABX, support the maximum bitrate.

The ACD 4805 controllers support velocity mode and current mode, and are configured for the latter, the MABX requesting reference currents through the CAN.

According to measurements done by the Transport Lab [2], the efficiency of the motor controller is 0.94 at a motor speed of 520 rpm and torque of 33 Nm.

3.1.2 Actuators

The RCV has a high degree of freedom actuation of its wheels. Each of the four wheels, sharing no common axis or actuator, can all be actuated individually. Four Thomson Linear Max Jac linear actuator actuates the wheel steering angle. Four Thomson Linear P-264 linear actuators actuates the camber of the wheels. The actuators are each controlled by one EPOS 70/10 motor controller. The controllers, as specified on p. 12, work in position mode.

The steering actuators can turn the wheels around 20 degrees inwards and

Table 2: Specifications of the four PRA-230 wheel hub motors.

Power	1,8 kW
Voltage	48V
Current	41A
Rated speed	520 rpm
Rated torque	33 Nm
Peak torque	150 Nm
Weight	13,8 kg
Diameter	255 mm
Depth (Width)	87 mm

15 degrees outwards in reference to the car body, which is suboptimal in regard of Ackermann steering. The camber actuators can tilt the wheels 10 degrees outwards (positive camber) and 15 degrees inwards.

The Max Jac actuators' motors have been found to be somewhat too weak for the steering. During testing of the vehicle four actuator motors have had their windings short circuited, most likely due to overheating. A software side limit of the actuator position extremes was then implemented to ensure power usage at suitable levels.

3.1.3 Wheel hub motors

Each wheel has a directly mounted gearless wheel hub motor. The motor, the Heinzmann PRA-230, is specified at 1.8 kW with a nominal voltage of 48 V. It has a stall torque at 150 Nm and a rated maximum speed of 520 rpm.

The motor is of permanent magnet synchronous type. The angular change of the rotor is measured by a 8-bit encoder. A temperature sensor measures the temperature of the stator. The measured values are sent to motor controller, for further relaying to the MABX by CAN.

According to measurements [2], the efficiency of the motor is 0.86 at the rated speed and rated load of 33 Nm at 520 rpm.

According to a stall drag test done by the Transport Lab, the effective traction force of one motor, as measured by a dynamo-meter fixed to the car body and ground, is 400 N, which is equivalent to 120 Nm with tyre radius 0.3 m. This test did not adjust for wheel friction.

3.1.4 Suspension

The RCV has camber actuated double wishbone suspensions. The scrub radius is about 25 mm.

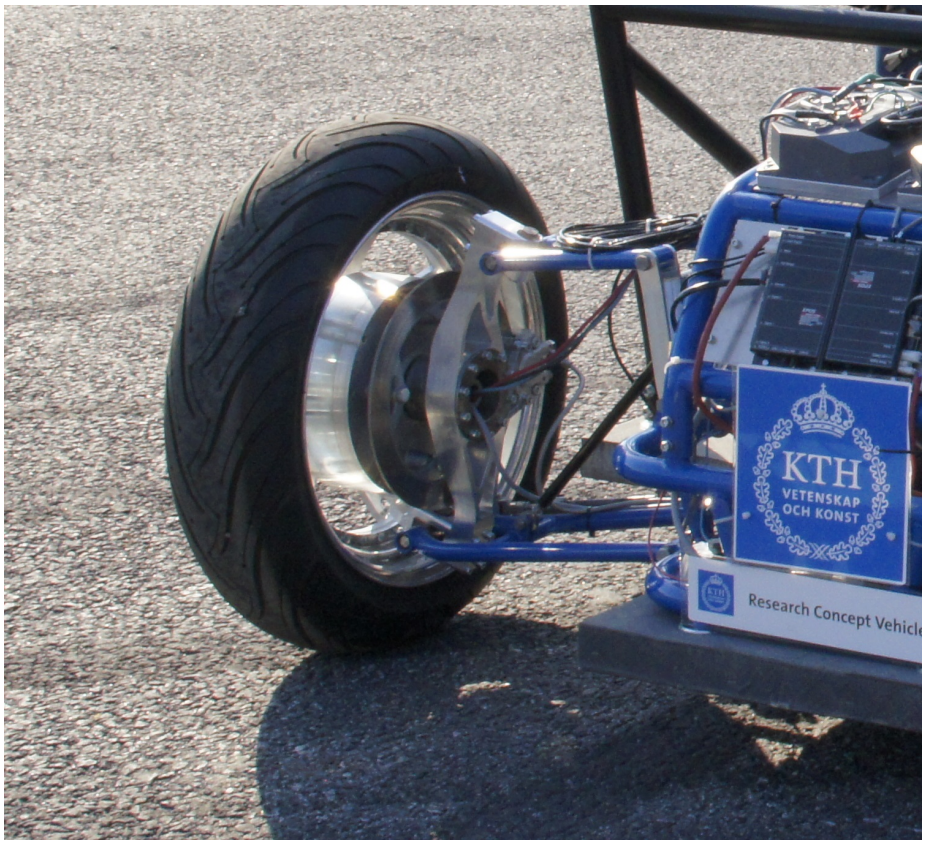

Figure 6: A picture of the front right wheel double wishbone suspension of the RCV.

The springs are made by Lesjöfors. From an original length of 150 mm, with a compression of 70 mm, the force is 3500 N (50 $^{\mathrm{N}}/_{\mathrm{mm}}$). The dampers, donations from Öhlins, have adjustable bump and return for both high and low speed and high grade.

The suspension is mounted on the bottom plate with aluminium inserts. Refer to Figure 6 for a picture of the suspension and to Figure 7 for a schematic representation of the linkage mechanism.

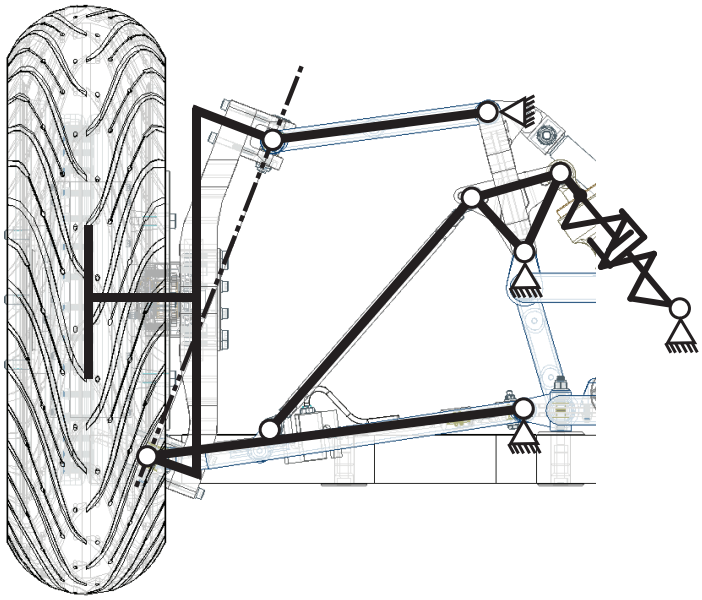

Figure 7: A schematic presentation of the upright mounting with static camber. The camber actuator connection point (upper most ground symbol), can be moved by the actuator in orbit around the ground symbol directly beneath it, to adjust camber angle.

3.1.5 Steering wheel

The steering wheel is essentially a wheel that is connected to an electric DC motor through a harmonic gear box with a 50-to-1 gear ratio. The motor has a position change encoder and is connected to an EPOS 70/10 motor driver.

The steering wheels position output is sent via CAN to the MABX and used as driver input. The MABX, in turn, sends torque requests to the steering wheel controller as haptic feedback. The steering wheel is capable of delivering 8 Nm of torque and can handle rotational rate up to 20 rad/s.

The haptic feedback model is as simple as a virtual torsion spring, which is winded up as the steering wheel is rotated. Also, data from the steering tie rod force sensors is used to get an actual road feedback into the steering wheels.

3.1.6 Bottom plate

The bottom plate of the RCV is the carrying structure of all components. It is a 5 over 5 under layer carbon fibre structure with a 50 mm sandwich laminate core (Divinycell H80) in the suspension mount areas and a 5 and 5 layer carbon fibre structure with a 50 mm PET core (Armacell ac115) in the roll cage area. The total mass of the plate is 34 kg.

The plate is specified to deflect no more then 5 mm when a 2 kN load over an 0.04 m^2 area is applied to the dead centre of the plate which rests on the suspension areas. According to simulations [5] the rated load gives a maximum deflection of 2 mm. The plate is specified not to fail up to a load of 6 kN in the centre.

3.2 The RCV firmware

As a by-wire vehicle the RCV is totally relying on its computers for operation. A by-wire vehicle have many theoretical advantages over a vehicle mechanically controlled directly by the operator, as computers allows for processing of the operator's inputs and decisions-making independent of the operator.

However, any software or general electrical error could have dire consequences, which in turn necessitate strict robustness of the systems for safe operation. The RCV, being a research prototype, is in its nature subject to experimental changes, and it is up to the operators the ensure safe operation given the set up as is.

The software controlling the RCV is centralized to the MABX, with a dashboard and a PC as an external interface. In the following subsections this software will be reviewed in detail. Refer to Figure 8 for a schematic overview.

3.2.1 General software set up

The RCV has software components subject to change and those that could be considered static from a programmer's point of view. The MABX, the PC interface and the IMU are highly customizable and reprogrammable. The actuator controllers, on the other hand, run proprietary firmwares which are configurable

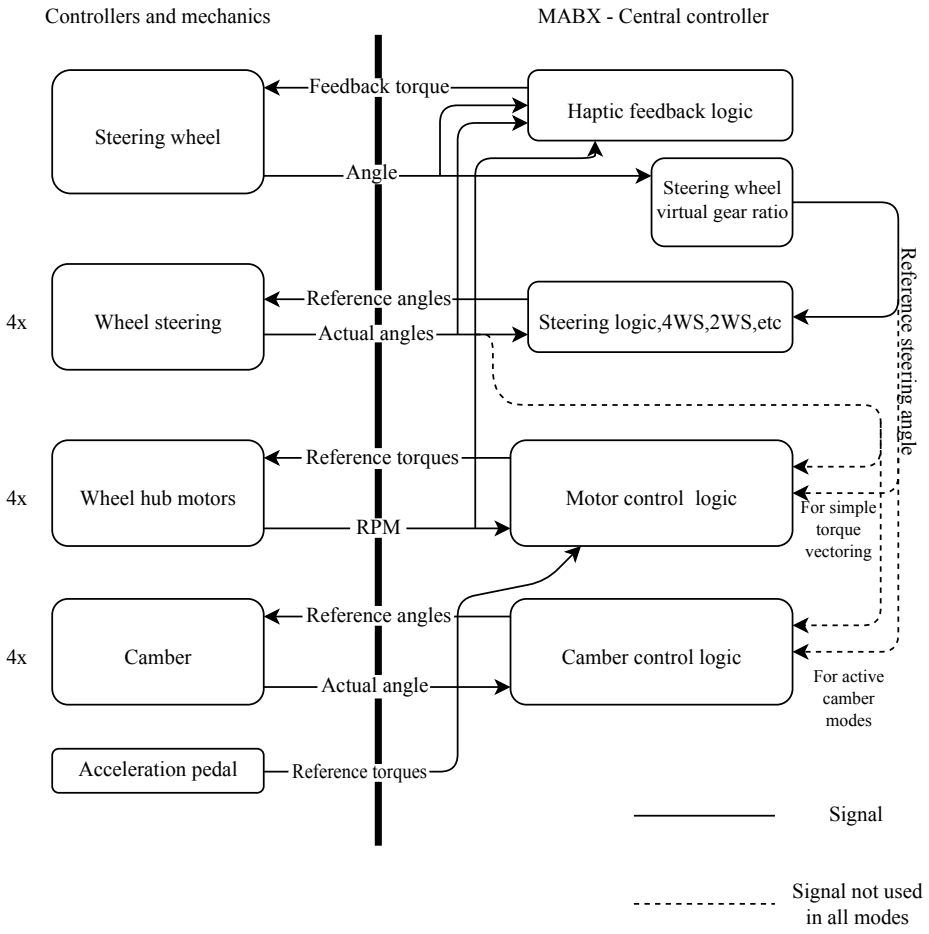

Figure 8: Schematic of the logical flow in the RCV actuation system. The schematic is not comprehensive. Signals such as motor temperature and actuator currents etc. are omitted.

but not reprogrammable, other than any firmware updates provided by the manufacturer.

MABX firmware The firmware of the MABX handles the overall operation of the vehicle. It receives the input from the operator , i.e. throttle, steering wheel angle, and inputs from the sensors, processes them and then gives orders to the actuators and motors, accordingly. Refer to Table 3 for a compilation of the input and outputs.

The firmware is programmed in Mathworks graphical language Simulink and then translated to C code with Simulink's pre-compiler. Then, the Microtech compiler and assembler compiles the C code to PowerPC-binaries.

The firmware consists of three abstract logical parts. The initialization, executed once, and the controller logic and the output, which are continuously executed during runtime and triggered by timer interrupts.

In the initialization phase, the MABX configures its own system, namely the two CAN buses as well as the RS232 serial interface, and starts the background task responsible for communicating with the PC.

In the input phase, the program reads the receiver buffer of the CAN and RS232 interfaces and collects values from the analogue to digital converters (ADC).

In the controller logic phase the program processes the inputs and, given the settings set by the operator. The logic for the camber, steering wheel, motors and wheel steering are handled by separate functions. These functions decide the torque and position requests and general commands that should be sent to the external controllers.

In the output phase the requests from the controller logic phase are sent to the corresponding controller via CAN. Also, in this phase, any error management or fail safe mechanism, such as over speed and over current protection for the wheel hub motors, are executed.

The conversion of the Simulink model to C code gives somewhat bloated code – in part totally unnecessary code. E.g. a CAN transmission block generates the check clause given below and depending on whether logging is enabled in the block (lower if-clause) or not (upper if-clause), the Simulink precompiler generates unnecessary checks, which might or might not be removed by the compiler.

```
//Part of code for one transmitter block
if (can_type1_msg_M1[CANTP1_M1_C2_TX_STD_0X621]->timestamp > 0.0) {
    }//This doesn?t do anything
[...]
//Part of code for another transmitter block
if (can_type1_msg_M1[CANTP1_M1_C2_TX_STD_0X621]->timestamp > 0.0) {
    rcv_B.SFunction1_gk = (real_T)//Here is does something
can_type1_msg_M1[CANTP1_M1_C2_TX_STD_0X621]->processed;
    }
```

The firmware of the MABX is easily reprogrammed via the PC interface.

PC interface The MABX has a serial interface which allows for connecting a PC. The PC is then able to access the memory of the MABX during runtime. The access is either realised through the dSPACE ControllDesk software, or with a supplied library for the Matlab scripting language “m” or C. As the serial interface is proprietary and undocumented, and the libraries precompiled, the interface is not very flexible.

The ControlDesk software functions both as a virtual instrument panel and a data logger. With a GUI of levers, buttons, numerical inputs, etc., the operator is able to change the memory of the MABX. E.g. an operator input could be the

Table 3: A tabular representation of the inputs and outputs of the program running on the MABX. The (optional) PC i/o interface is excluded.

Number of i/o	Name	Descr. of signal	Interface type
Inputs:			
4	Tie rod force sensors	Raw force values from in tensile sensor through an instrumental amplifier	Analogue value
1	Pedal is engaged sensor	When the operator pushes to accelerator pedal the signal goes high	Digital value
1	Pedal position sensor	How much the pedal has been pressed down, by potentiometer	Analogue value
4	Wheel steering actuator controller	Error codes from the controller and current, position from the wheel steering actuators	CAN
4	Wheel camber actuator controller	Error codes from the controller and current, position from the wheel camber actuators	CAN
4	Wheel hub motor controller	Error codes from the controller and current, voltage, temperature, RPM from the wheel hub motors	CAN
1	Steering wheel controller	Error codes from the controller and current, position of the steering wheel haptic feedback motor. This position is used as steering wheel angle	CAN
2	Dashboard direction selector	Button with states for forward, neutral and reverse	Digital value
12	Dashboard mode selector	Nub button with 12 states for choosing 4WS or 2WS etc	Digital value
1	Dashboard button	Multi purpose button	Digital value
1	Inertial measurement unit (IMU)	Measurements of 3DOF acceleration, 3DOF angular velocities, GPS position	RS232
Outputs:			
4	Wheel steering actuator controller	Requested position, initializing configuration	CAN
4	Wheel camber actuator controller	Requested position, initializing configuration	CAN
4	Wheel hub motor controller	Requested torque, initializing configuration	CAN
1	Steering wheel controller	Requested torque for haptic feedback	CAN
3	Dashboard warning lamp	Either green, orange or red light	Digital value

numerical value of a speed cap or a selection of the different operating modes of the controller logic. The flashing of the program to the MABX is done through ControlDesk.

It is possible to connect a generic joystick/gamepad to the PC and then use it together with ControlDesk to relay the inputs to the MABX.

The live vehicle data logged by ControlDesk can be exported to the Matlab .mat binary file format. Then, the data can be post-processed and visualized with a graphical user interface written in m.-script.

A more in detail explanation of the ControlDesk GUI, Matlab post-processor and of the RCV PC interface is available in the appendix.

3.2.2 Vehicle dynamics control

The steer-by-wire implementation of the RCV with no common physical axes makes the dynamics of the vehicle more influenced by software than traditional vehicles. E.g. the steering geometries can be arbitrarily chosen within mechanical limits and the wheel hub motors can be used as a kind of virtual differentials.

The steering actuation on the RCV use the Ackermann steering geometries for both two wheel steer and four wheel steer, with the option of moving the turning point with regard of the vehicle velocity.

Wheel steering with steering wheel The vehicle has tree main modes for steering the wheels with the steering wheel. Two wheel steer, four wheel steer and movable turning point steering. The turning point, being the point around which the vehicle rotates at steady state cornering.

With two wheel steer, as given in equations 1 and 2, the angle of the front wheels are set in such a way that the wheels follows their trajectory on the cornering circle, with the rear wheels fixed. Analogous, equations (3 – 6) gives the corresponding for four wheel drive with the rotational point in the centre of the vehicle.

In the case of movable turning point steering, the wheel angles follows equation (7 – 10) and the turning point is moving on a line perpendicular to, and between, the rear and front axes parallel, of the vehicle. A speed dependent way, such that that the turning point crawls to the rear as the speed increases, starting halfway between the steering axes at stand still.

Each of these modes are implemented in the Simulink model as separate triggered blocks, which are enabled by a case switch.

Joystick steering Given the over actuation qualities of the four wheel steering, the traditional driver input through a steering wheel, which in information space could be visualized by a line, is not able to, in it self, make the most out of the dynamic possibilities the set-up provides, if not by computer aided driving, like, for example, Electronic Stability Control, etc.

With two joysticks, the operator of the vehicle can steer the vehicles with four degrees of freedom (two separate planes in space), which in theory is enough to fully control all wheels, although highly impractical.

In the RCV joystick controller mode, the operator is using two joysticks, only using one left-right axis on each stick, one in the right and one in the left hand, to control firstly ordinary steering angles, as the joystick would have been a steering wheel. With the other joystick the operator steers the wheels in such a way that they all steer in the same direction, which is, as the operator moves the joystick to the left, all wheels steer to the left.

Thus, the vehicle is able the move to the left and right without rotating.

Wheel hub motor torque allocation The in wheel hub motors are controlled using different modes of operation, namely uniform torque distribution, simple torque vectoring, and "tank steer".

In uniform torque mode, the operator input through the throttle pedal is translated to a torque that is uniformly distributed among the four wheels.

In simple torque vectoring, the operator input through the throttle pedal is firstly, as in the previous described mode, translated to a torque. Then, depending on the steering wheel angle, a proportion of the torque is subtracted, for the inner wheels in the cornering, and added, for the other wheels. Refer to Equation 31.

The tank steering mode works in a similar way to simple torque vectoring, but, in addition to the relative gain, that's proportional to the throttle, an absolute term is added for the outer wheels, and subtracted for the inner, in such a way, that the vehicle would turn around even if standing still; enough force allocated.

The gains and offsets of these modes are configurable during runtime through the ControllDesk interface.

For all modes, regenerative braking is triggered when the pedal position is bellow a certain limit, by default 15 %. When under 15 %, a torque in opposite the driving direction is applied to the wheels, proportional to the driving speed.

3.2.3 Laboratorial modes

For the purpose of experiments and measurements the RCV has modes specially designed for vehicle dynamics testing. Bellow the most important are listed.

The step steer mode, is a mode in which the vehicle will have zero steering reference angle until the operator turns the wheel 45 degrees to the left or the right, by which time the vehicle will do a steep steer in that direction by an arbitrary angle, until the operator returns the steering wheel bellow 45 degrees. This, to e.g. measure yaw response.

The toe sweep mode, is a mode in which the front wheels of the vehicle will follow a saw tooth angle reference signals in opposite sign of each other. I.e. when the front left wheel steers left, the right wheel will steer right and vice versa. In this way, with the help of force sensors in the tie rods, some of the wheel forces can be estimated.

The camber step mode, is identical to step steer mode, but instead of a step in steering angle there will be a step in camber angle, in such a way, that all wheels tilt to the left if the operator steer 45 degrees to the left, and vice versa.

The roll out mode, is a mode to test the rolling and drag resistance. When a key on a joystick connected to the host PC is pressed, all motor torques are set to zero. Because of the regenerative braking the roll out test would not work by just disengaging the throttle pedal.

3.2.4 Inertial measurement unit

The RCV is equipped with an in house built[6] Inertial Measurement Unit (IMU). The IMU provides measurements of 3DOF acceleration, 3DOF angular velocities, 3DOF magnetic fields and GPS position. The data is sent to the MABX via RS232 serial connection and then used for state estimation and logging.

The hardware is Arduino-based with off-the-shelf components, i.e., a GPS receiver chip, EM-506A, and IMU-chip, Razor, with magnetic field sensors.

The main microcontroller of the IMU is the Atmel 8-bit ATmega2560, which is programmed in a C++-like language. Internally, the ATmega2560 communicates with an Atmel 8-bit ATmega328 microcontroller, also programmed in house, which operates the IMU- and magnetic field sensor-chips. The communication between the microcontrollers is via a Serial Peripheral Interface (SPI).

The serial output from the IMU can be configured to be either in ASCII text or as binaries. The binary message frame is presented in Table 4. Apart from transmitting the measured data to the MABX, the data is also available on a LCD on the IMU.

The magnetic readings are adjusted for roll ρ and pitch γ according to:

$$\begin{bmatrix} M_{x.adj.} \\ M_{y.adj.} \\ M_{z.adj} \end{bmatrix} = \begin{bmatrix} \cos\rho & 0 & \sin\rho \\ \sin\gamma\cos\rho & \cos\gamma & \sin\gamma\cos\rho \\ -\cos\gamma\sin\rho & -\sin\gamma & \cos\gamma\cos\rho \end{bmatrix} \cdot \begin{bmatrix} M_{x.norm.} \\ M_{y.norm.} \\ M_{z.norm.} \end{bmatrix} \tag{32}$$

where $M_{i.norm.}$ is the i:th axis' magnetic field strength normalized by the euclidean distance $\|\mathbf{M}\|$, and roll ρ and pitch γ are given by:

$$\rho = \arcsin \ddot{x} \tag{33}$$

$$\gamma = \arcsin \frac{-\ddot{y}}{\cos\rho} \tag{34}$$

When the car accelerates, $\mathbf{M_{ajd.}}$ will be inaccurate due to erroneous roll or pitch. Heading is thus rather derived from extrapolating GPS positions when the IMU is moving 'fast'. In fact, the accuracy of a compass on a electric vehicle is very deficient when moving at all, due to high torque currents and resulting magnetic fields. Refer to Figure 34 in the Appendix C.

Note that, as the Atmel chips are little endian byteorder, and the IBM/-Motorola PowerPC chip in the MABX is big endian (Motorola byte order), the incoming bytes has to be swapped in the MABX, e.g. the byteorder of a float32 from the IMU, 0-1-2-3, is converted to order 3-2-1-0 in the MABX.

The data output from the IMU is unfiltered but calibrated. Refer to appendix C for details about how the IMU should be calibrated and how the raw

Table 4: A tabular representation of the outputs of the IMU. The proceeding number is byte number. The total length of the message is 54 bytes plus a 2 byte hash. The communication interface is RS232.

0:System time ms uint32	4:Satellites in view int16	6:Latitude $degrees$ float32
10:Longitude $degrees$ float32	14:Compass $radians$ float32	18: Acc. x m/s^2 float32
22: Acc. y m/s^2 float32	26: Acc. z m/s^2 float32	30: Roll rate (x-axis) rad/s float32
34: Pitch rate (y-axis) rad/s float32	38:Yaw rate (z-axis) rad/s float32	42:Magnetic field x float32
46:Magn. field y float32	50:Magn. field z float32	54:hash int8x2

magnetometric data should be converted to bearing and refer to appendix D for instruction on how the error hash is calculated.

3.3 Prior work

The planning for the RCV started in late 2012 and the construction commenced early in 2013.

Some fundamental design choices build on concept work laid down by KTH researcher S. Zetterström [9], namely, the so called autonomous corner module (ACM). Refer to Figure 9. The design differs somewhat from the implementation in the RCV, notably, there are no active suspension, and the lower wishbone arm is not replaced by dual linear steering actuators.

Zetterström proposes that a "vehicle equipped with the arrangement according to the invention may comprise a control device of the 'joystick' type or similar, replacing a conventional steering-wheel." — a feature actually implemented in the RCV, to make the most of the manoeuvrability the autonomous corner modules offer.

Further research has been done at KTH concerning the ACM. E.g. M. Jonasson [10] argues that ACM:s could be used to improve vehicle performance as well as safety features and industrial design simplifications in regards of manufacturing. The authors lists increased unsprung mass as drawbacks of an ACM.

The steering wheel of the RCV is a research item made by Malmquist [8]. It is designed to minimize the total volume of the haptic feedback motor and harmonic drive gearbox. This, according to the authors models, prerequisites a gear ratio of about 60.

A great deal of future research is planned for the RCV as it is beginning to take shape of a mature research platform. Refer to O. Wallmark et al. [11] for an extensive review.

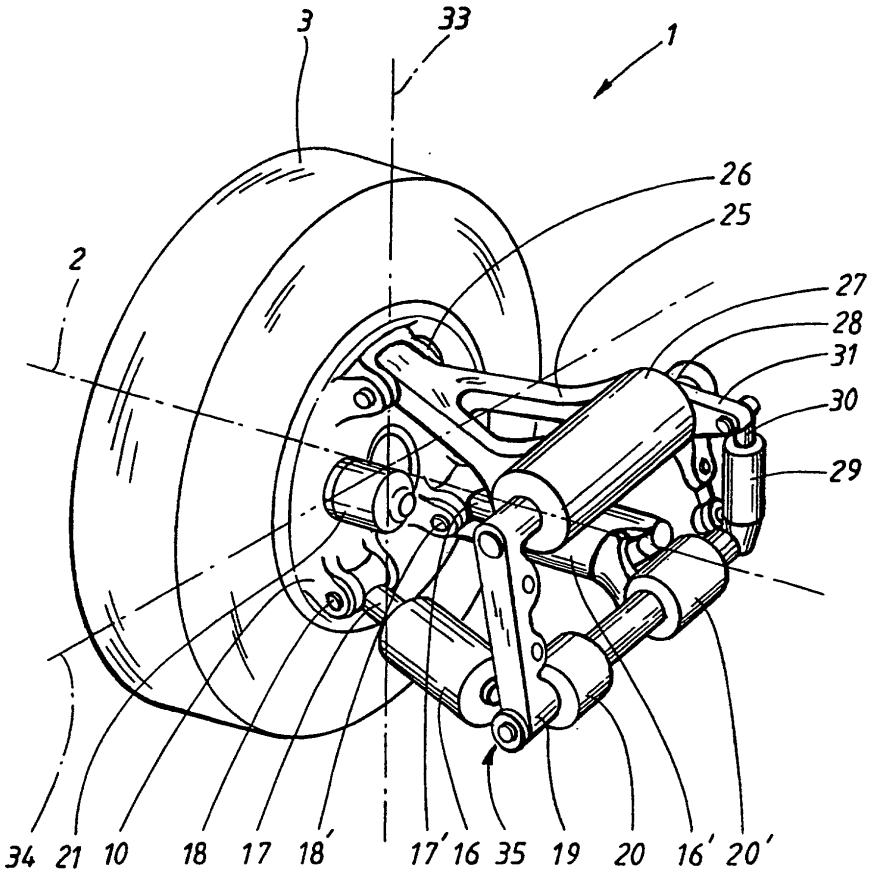

FIG. 2

Figure 9: The "autonomous corner module". Figure from patent EP1144212. The wheel has active suspension as well as steering and camber actuators.

4 Experimental results

In this section some experimental set-ups and live so called hardware-in-the-loop runs with the RCV as a lab on wheels will be reviewed.

Firstly, in the subsection Data logging, the logging and measurements of data will be evaluated and the IMU will be validated. In the sub-sequential sections some examples of what the data could be used for is presented.

4.1 Data logging

All data measured during a run is logged via the MABX to PC interface and the ControllDesk software. The variables to be logged are selected in ControllDesk and it is possible to log variables corresponding to most of the more simple blocks available in the Simulink model as well as those signals that are terminated by a termination block or not removed during optimisation of the code, during the conversion between Simulink and C by the Simulink precompiler.

The data is logged by a background task of the firmware running on the MABX at a sampling rate of 0.002 seconds. This gives a data rate of about 260 kB per second. The sampling rate is well below the internal task timer rate of relevant timer tasks. E.g. the main control logic runs at a period of 0.03 seconds and measurement data are requested from the actuators and motors every other 0.03 seconds or every 0.03 seconds, depending on priority, as only two outgoing CAN messages with the same ID destination can be queued at the same time due to limitations in the MABX. Refer to Figure 10 for an example of a measurement of current on a wheel hub motor from a test run.

During a test run an industrial grade VBOX IMU with GPS was fitted to the RCV to be able to validate the RCV's own IMU. As seen in Figure 11 the GPS data of the RCV's IMU is accurate compared to the reference GPS receiver, but the lower sampling rate of the RCV's, which is 1 Hz compared to 10 Hz, is clear. For the same run the accelerometer data is compared. See Figure 12. As seen, the RCV IMU data is accurate compared to the reference IMU. The same is true for angular velocities (not shown).

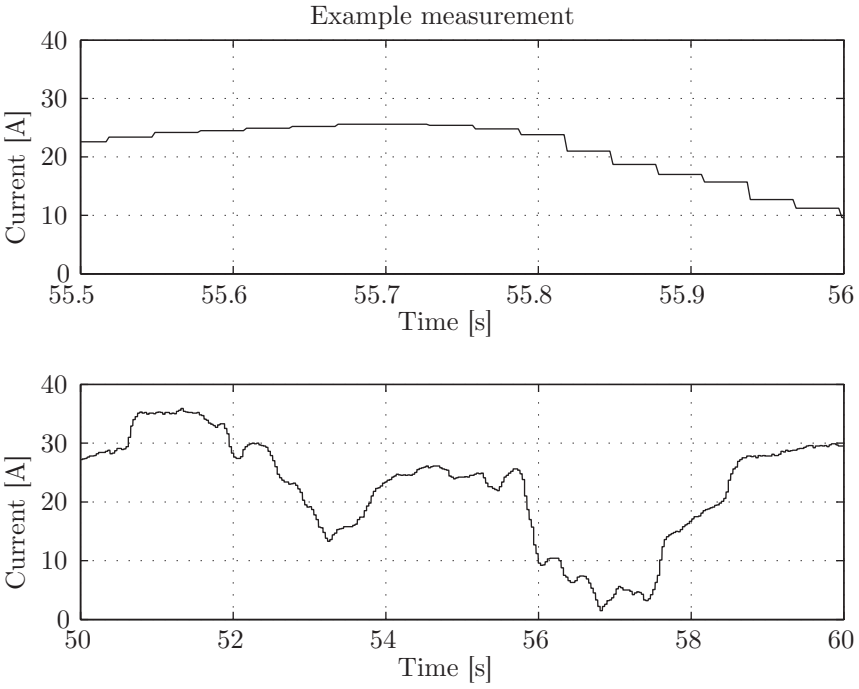

Figure 10: An example of current data from a wheel hub motor. The sampling time via the CAN-bus of 0.03 is clearly visible in the top most zoomed in plot.

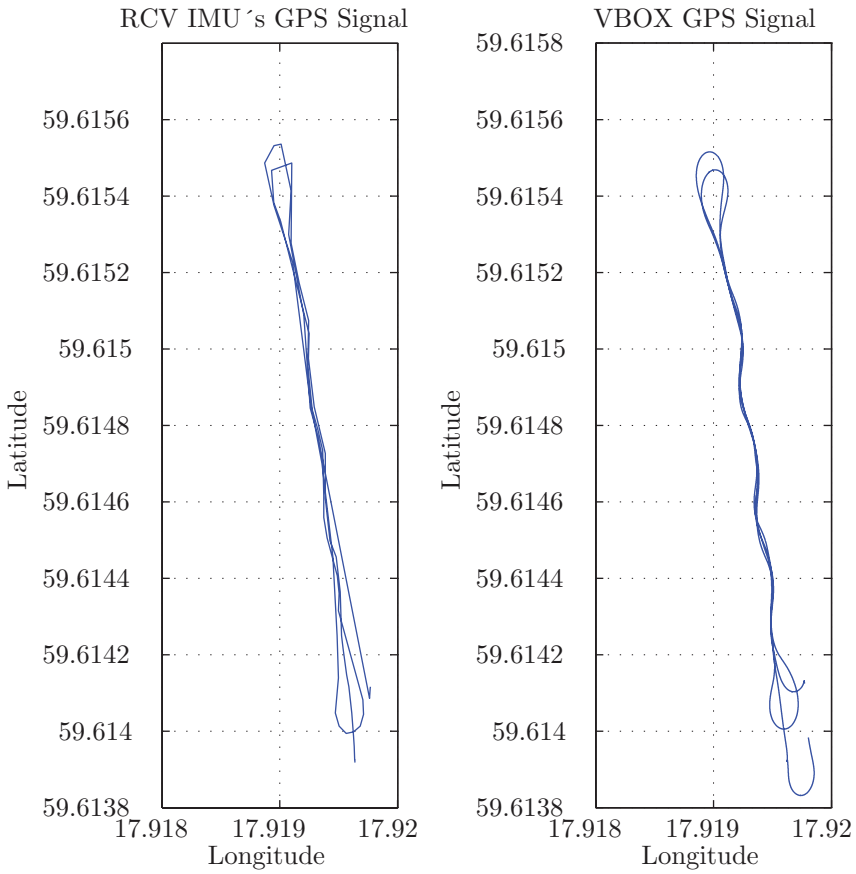

Figure 11: An example of positional data from the GPS receivers during a slalom track test. The lower sampling time of the RCV's GPS is visible.

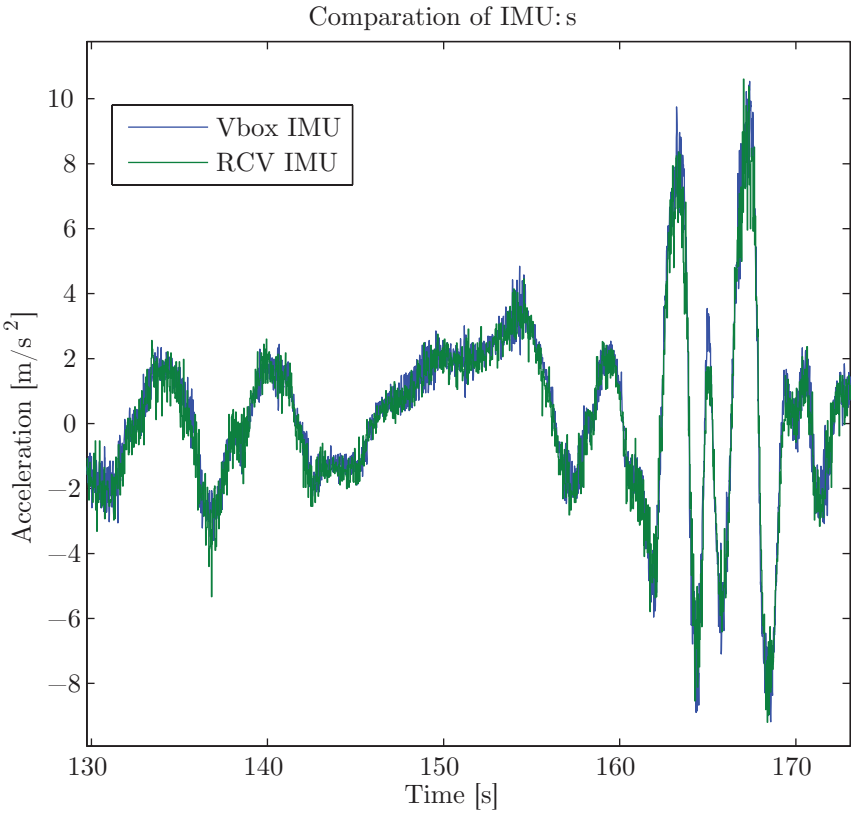

Figure 12: An example of acceleration data from the RCV's IMU and the reference IMU, with good correspondence.

4.2 Circle test

In test runs the RCV is driven counter clockwise a circle with a radius of approximately 11 meters, visualized by GPS data shown in Figure 13. This is done with different set-ups, namely two wheel steer, four wheel steer, and ditto but also with active camber. Given steering angle, Ackermann angle derived from circle radius and lateral acceleration, the under-steer gradient can be calculated according to:

$$\delta = \delta_a + K_{us} \cdot \ddot{y} \tag{35}$$

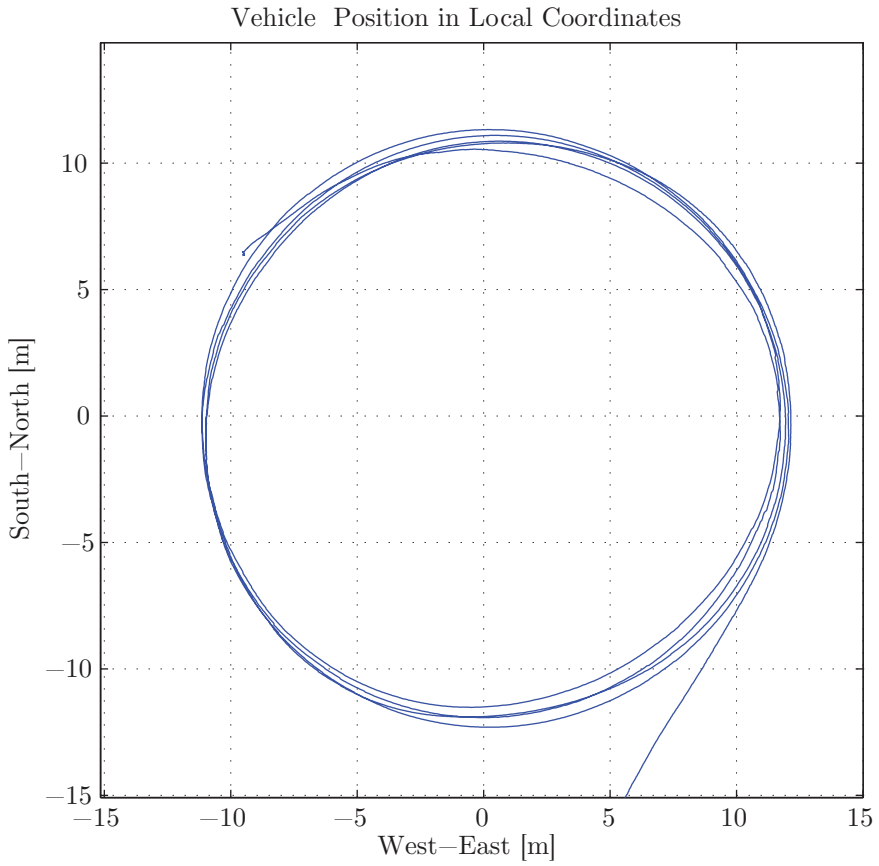

Figure 13: GPS coordinates from a circle test translated to absolute position in meter relative a point on the road. Radius being fairly constant during the run.

Assuming small angles, given an Ackermann angle of $L/R = 0.18$ for two wheel steer and $L/(2R) = 0.09$ for four wheel steer, with eq. 35 the understeer gradient K_{us} are derived for the different set-ups. The steering angle δ is in this

case defined as the arithmetic average of the steering angle of *all* the wheels and is shown, for the four wheel steer test run, in Figure 14.

The test cases are four wheel steer, two wheel steer, four with steer with active camber, and two wheel steer with active camber. K_{us} is calculated with a least mean square linear regression. The derived K_{us} is presented in Table 5 and in Figure 15 to 17.

Table 5: A tabular representation the understeer gradient for different set-ups.[1]Brushed data, refer to Figure 15

Set-up	$K_{us}[rad/\frac{m}{s^2}]$	$K_{us}[^\circ/g]$	Conf. Int. [95%]
4WS zero camber	-0,0014	-0,76	$K_{us} \in_{p0,95} [-0,0014 - 0,0013]$
4WS zero camber[1]	-0,0009	-0,51	$K_{us} \in_{p0,95} [-0,001 - 0,001]$
2WS zero camber	-0,0005	-0,28	$K_{us} \in_{p0,95} [-0,0005 - 0,0005]$
4WS active camber	-0,0026	-1,5	$K_{us} \in_{p0,95} [-0,0027 - 0,0026]$
2WS active camber	-0,0008	-0,42	$K_{us} \in_{p0,95} [-0,0007 - 0,0008]$

According the computed K_{us} it is clear that the RCV is over steered. 4WS makes for more oversteer than 2WS, with a factor of approximately 3 times as big K_{us}.

Observe that, when interpreting the data, as the steering angle δ used is an average of all the four wheels angles, the K_{us} need to be multiplied by 2 when comparing to 2WS, where rear wheel steering angle is 0, to the K_{us} of other 2WS vehicles, where δ is the average of only the two front wheels. Accordingly, 4WS K_{us} is approx. 3 times larger then 2WS on the RCV, the attribution to the understeer gradient, for each degree steering from each wheel actually steering, is approx. only 3/2 times as large for 4WS compared to 2WS, which is arguably a more appropriate comparison.

A critical speed is the theoretical velocity by which an over-steered vehicle will spin out in steady state cornering.

$$\delta = \delta_{Ack.} + a_y K_{us} \tag{36}$$

, substitute $\delta_{Ack.}$ with L/R, a_y with v_x^2/R, and rearrange:

$$\sqrt{\frac{\delta R - L}{K_{us}}} = v_{x,crit} \tag{37}$$

The speed for which δ is zero is the critical speed. To find the critical speed, steering angle is plotted versus lateral acceleration for a 2WS circle test. See Figure 19.

According to the regression, the K_{us} is 0.0009, which is the same as in Table 5, but multiplied by two, as in this case the wheel angle of the two front wheels are used, instead of all four.

The acceleration for which δ is zero in Figure 19 can be calculated to 206 $^{m}/_{s^2}$, which then translates to critical speed by: $a_y = v_x^2/R \Rightarrow v_x = 46$ $^{m}/_{s}$.

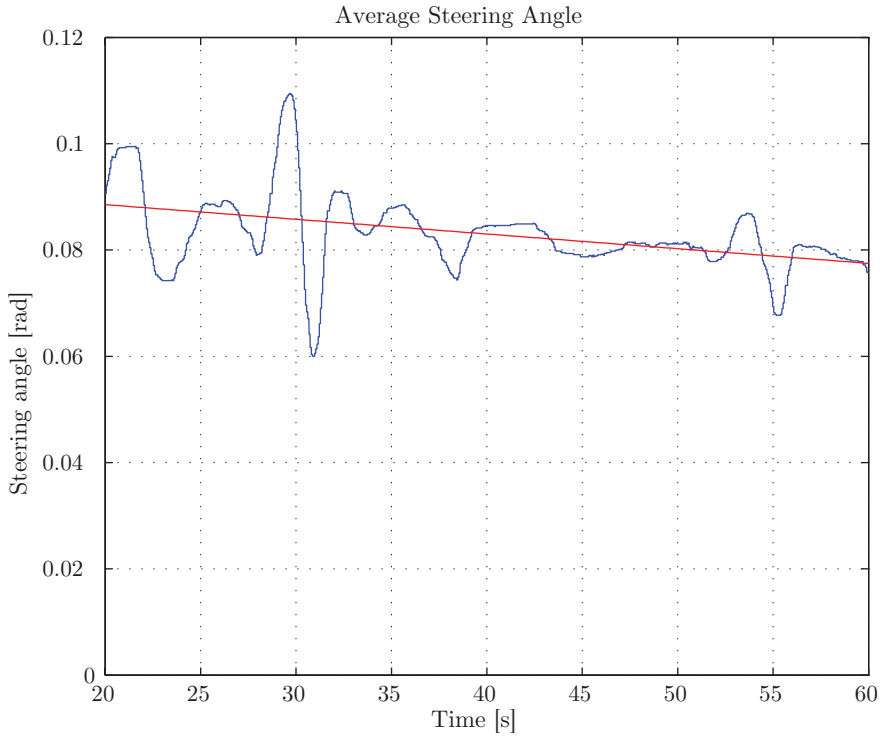

Figure 14: The average of the steering angle of the four wheels during a circle run. Linear trend shown.

The critical speed can also be derived from the K_{us} Table 5. A K_{us} of -0,0005 for 2WS zero camber, is multiplied by two, then eq. 37 gives the critical speed to 47 $^{m}/_{s}$, which complies well with the the value of 46 $^{m}/_{s}$ derived from the regression.

As above, the critical speed for the understeer gradients presented in Table 5 are computed and presented in Table 6. Note that when calculating the critical speed for 2WS the understeer gradient is multiplied by 2.

Table 6: A tabular representation the criticall speed for different understeer gradiens. [1]Brushed data, refer to Figure 15

Set-up	$K_{us}[rad/\frac{m}{s^2}]$	v_{crit} m/s
4WS zero camber	-0,0014	38
4WS zero camber[1]	-0,0009	47
2WS zero camber	-0,0005	45
4WS active camber	-0,0026	28
2WS active camber	-0,0008	35

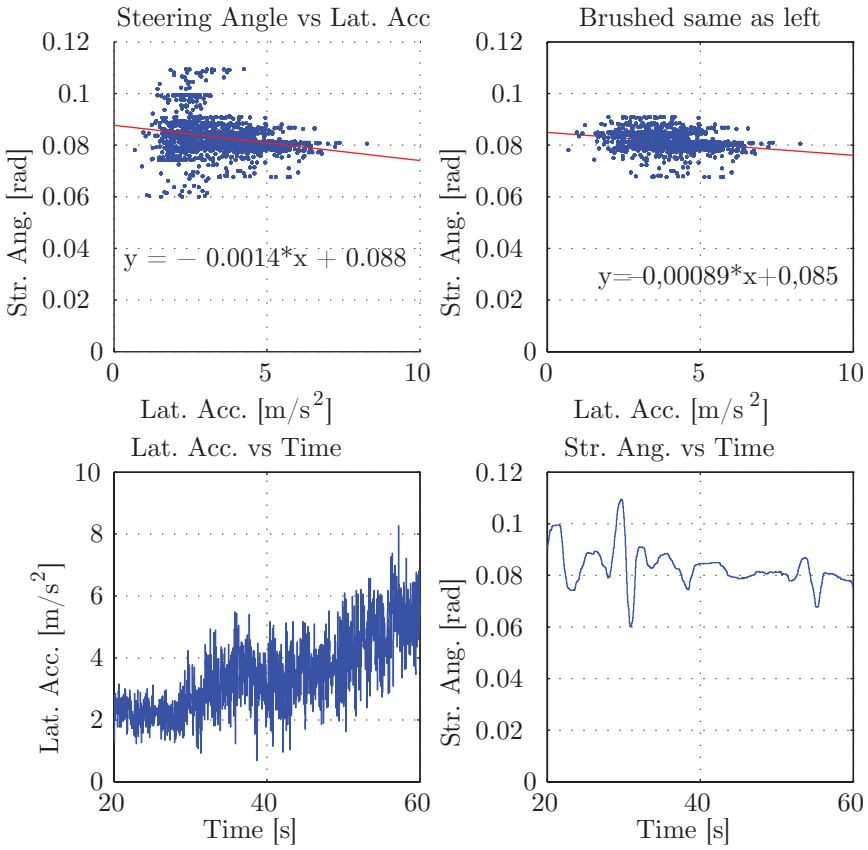

Figure 15: Data and results from the circle test with constant radius and four wheel steer. In the top right plot data from 34 seconds and earlier have been discarded, as there, as seen in the bottom right plot, were some jerky steering actuation at that time.

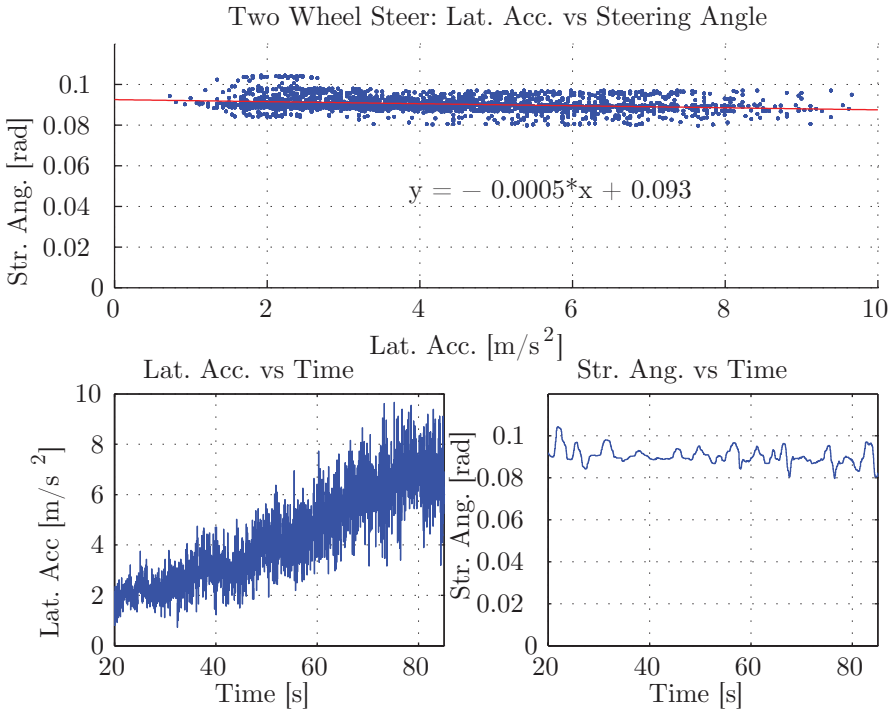

Figure 16: Data and results from the circle test with constant radius and two wheel steer.

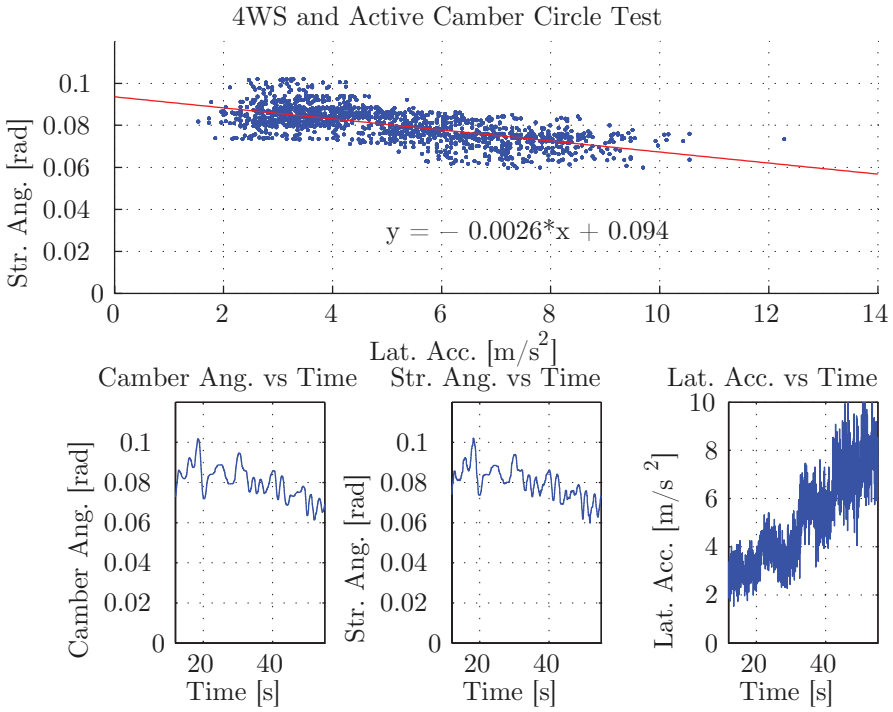

Figure 17: Data and results from the circle test with constant radius, two wheel steer and a camber gain proportional to the *steering wheel* steering angle.

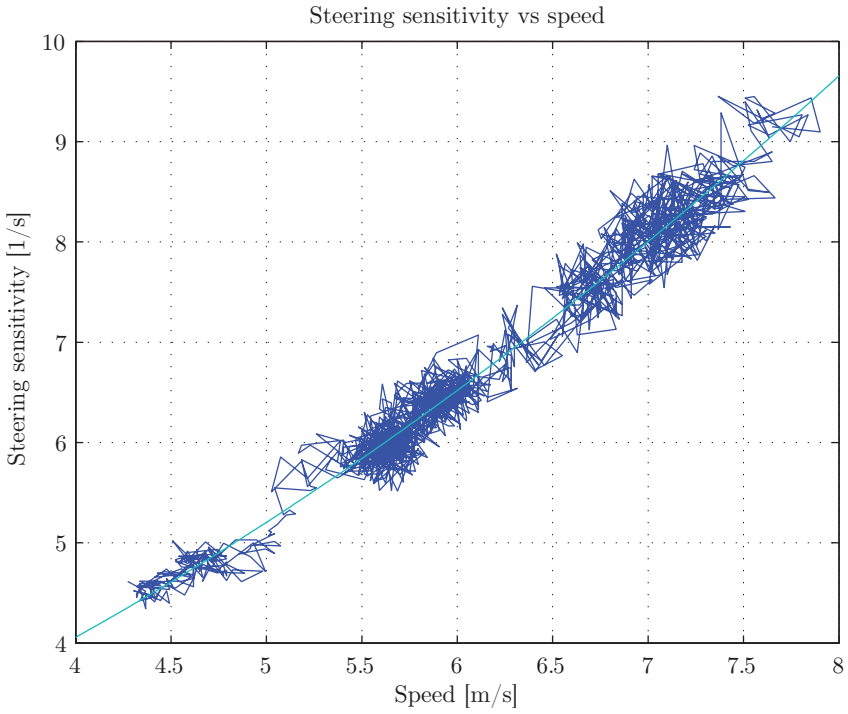

Figure 18: Steering sensitivity, Ψ/δ, versus speed, for a 2WS no camber circle test.

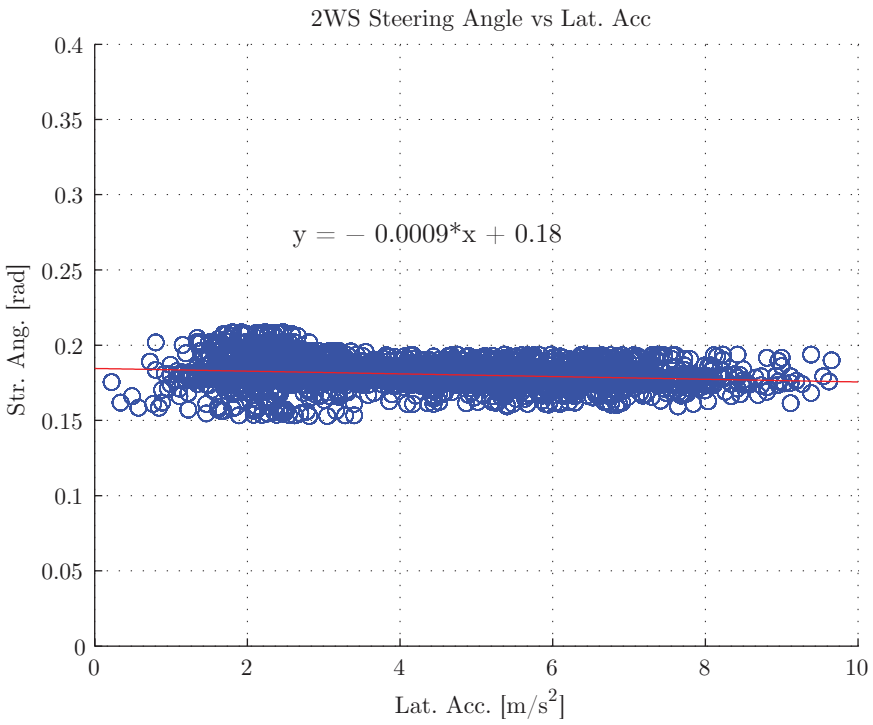

Figure 19: Data and results from a circle test with constant radius, two wheel steer and zero camber. For the purpose of calculating critical speed, the steering angle is instead defined as the average of the two front wheels, as compared with the earlier figures, where, for the sake of comparison, the steering angle is defined as the average of all wheels. The offset of 0.18 corresponds well with the theoretical Ackermann angle of $R/L = 0.18$ for radius $11m$ and length $2m$

4.3 Acceleration and torque test

On a straight the RCV is driven with full throttle to estimate acceleration and torque characteristics.

The AC Motor's DC current is estimated by the motor controllers. Motor power is estimated by:

$$P = UI \tag{38}$$

U being volt and I being current. The kinetic energy of the vehicle is estimated as:

$$E = \frac{mv_x^2}{2} + \frac{J_{wheels}\omega^2}{2} \tag{39}$$

where ω is the angular velocity of the wheels, J_{wheels} being the moment of inertia of the wheels, v_x being forward velocity and m mass. The effective power $P_{eff.}$, not including rolling resistance, drag, etc, is derived by $\dot{E} = P$, gives:

$$P_{eff.} = mv_x a_x + J_{wheels}\omega\dot{\omega} \tag{40}$$

Equivalent effective torque, is then derived by:

$$\tau = P_{eff.}/\omega \tag{41}$$

I_{wheels} are unknown but estimated as a 4 pieces of 0.3 meter radius 18 kg cylinders, giving $I_z = 4 \times mr^2/2 = 3$ kgm^2. Compared to the RCV's mass, with an equivalent moment of inertia of $r^2m = J_{eqv} = 54$ kgm^2, the term from moment of inertia to power could be considered small and is disregarded in the calculations.

Applying Equation (38 – 41) to a test run, renders Figure 20 and 21. The total power peak is at 16 kW, where battery protection cuts off. At this point during the test runs, the effective power is about 7 kW – the conversion from the power input to the motor controller, to translational acceleration, being at best 0.5.

It is possible to estimate cable and battery losses, by assuming a battery model:

$$U_{out} = U_{internal} - R_{internal}I_{out} - R_{cable}I_{out} \tag{42}$$

An estimate of the sum of the resistances of the battery and the cable is given by a least mean square regression of Equation 42, to 18m Ω with $p_{0.95}$ interval of [17.8m Ω 17.9m Ω]. At 300 A this gives an power of 1.5 kW in cable and battery losses, according to the model in Equation 42. Refer to Figure 22.

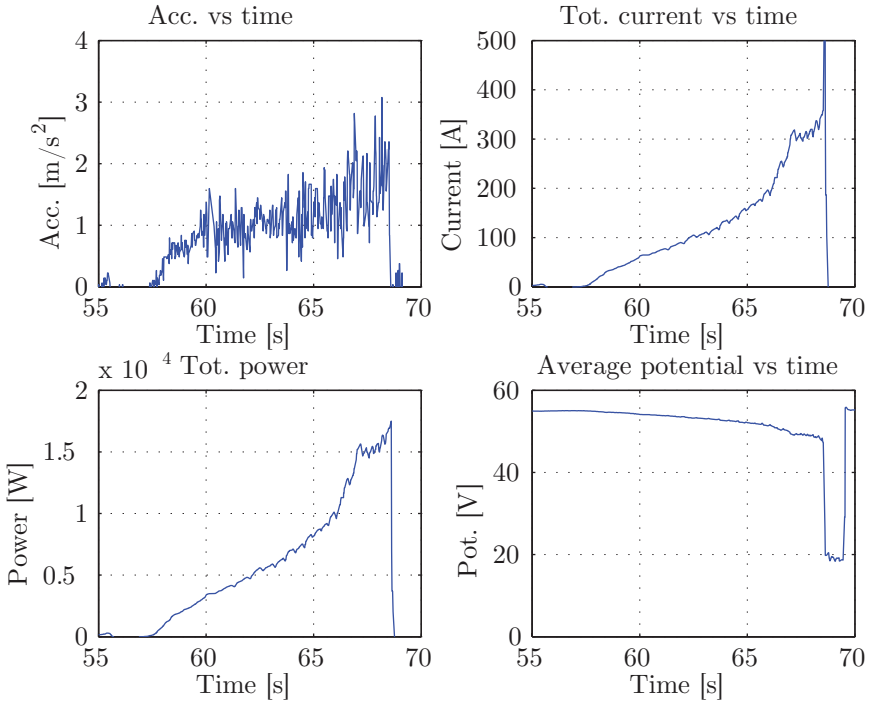

Figure 20: Data from one acceleration test run. Power is electric power, UI, not derived power from a_x and v_x.

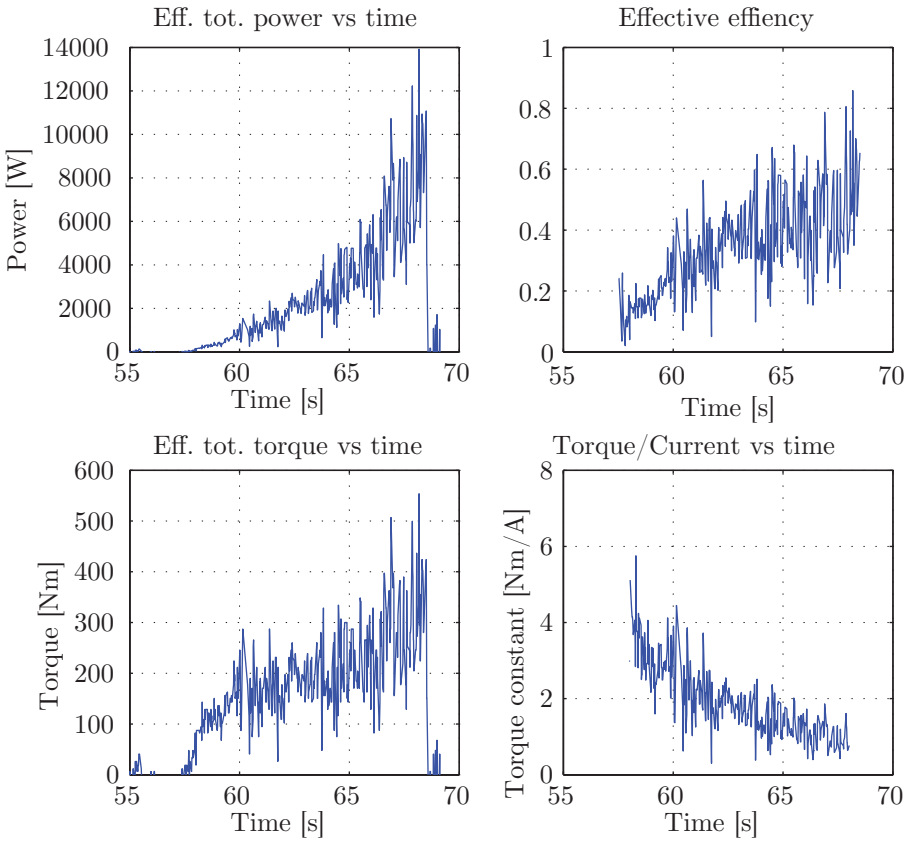

Figure 21: Derived data from one acceleration test run. Power is effective power, i.e. derived from acceleration and mass. Effective efficiency is $P_{eff.}/P_{tot}$, where P_{tot} is UI.

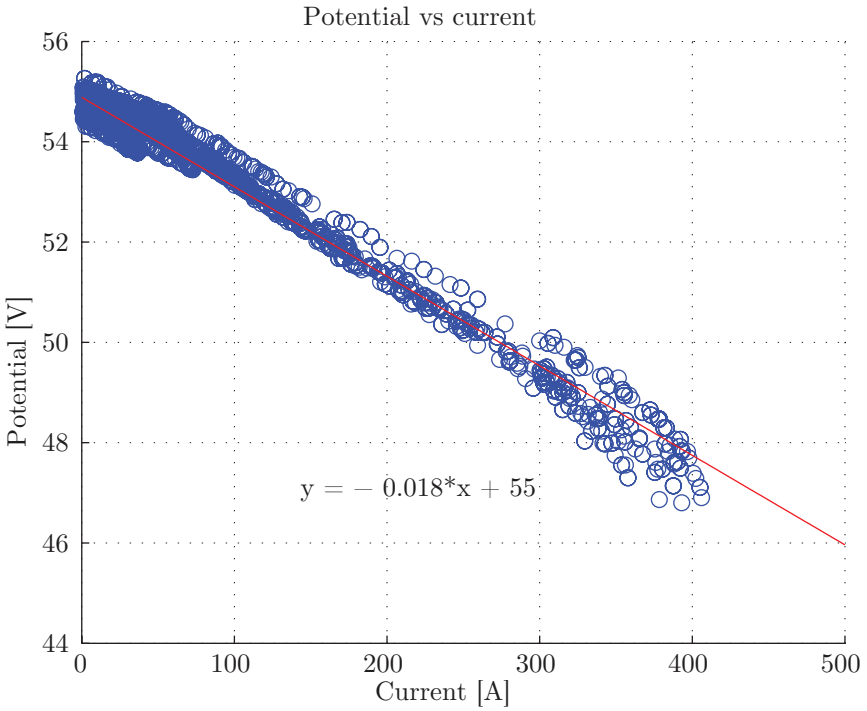

Figure 22: Scatter plot of the average voltage over the motor controllers as function of total current.

4.4 Step steer

During a test run the RCV had enabled the so called step steer steering, by with which the steering does a step response to the left or right as the operator turns the wheel more then 45 degrees to the left or right.

The amplitude of the step can be set arbitrary. When the operator turns the steering wheel back below 45 degrees, the reference angle goes back to zero.

The well known bicycle model:

$$\begin{pmatrix} mD + \frac{C_{12}+C_{34}}{v_x} & mv_x + \frac{fC_{12}-bC_{34}}{v_x} \\ \frac{fC_{12}-bC_{34}}{v_x} & J_zD + \frac{f^2C_{12}+b^2C_{34}}{v_x} \end{pmatrix} \begin{pmatrix} v_y \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} C_{12} \\ fC_{12} \end{pmatrix} \delta_f \tag{43}$$

,can be extended, as shown in subsection 2.3, to include a non zero rear steering angle, given front angle δ_f and rear angle δ_r; accordingly:

$$\begin{pmatrix} mD + \frac{C_{12}+C_{34}}{v_x} & mv_x + \frac{fC_{12}-bC_{34}}{v_x} \\ \frac{fC_{12}-bC_{34}}{v_x} & J_zD + \frac{f^2C_{12}+b^2C_{34}}{v_x} \end{pmatrix} \begin{pmatrix} v_y \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} C_{12} & C_{34} \\ fC_{12} & -bC_{34} \end{pmatrix} \begin{pmatrix} \delta_f \\ \delta_r \end{pmatrix} \tag{44}$$

Solve eq. 44 for $\dot{\psi}$. Apply Laplace transform. Let $s \to 0$, which gives:

$$\dot{\psi} = \frac{\mathrm{C_{12}\,C_{34}\,v_x\ (b+f)\ (d_f - d_r)}}{\mathrm{C_{12}\,C_{34}\,b^2 + 2\,C_{12}\,C_{34}\,b\,f + C_{34}\,m\,b\,{v_x}^2 + C_{12}\,C_{34}\,f^2 - C_{12}\,m\,f\,{v_x}^2}} \tag{45}$$

Substitute $b + f$ with L, divide numerator and denominator with $LC_{12}C_{34}$, set $\delta = \delta_f = -\delta_r$, as of the 4WS steering constrains, giving:

$$\dot{\psi} = \frac{2\delta v_x}{L + \frac{mv_x^2(bC_{34}-fC_{12})}{LC_{12}C_{34}}} = \frac{2\delta v_x}{L + K_{us}v_x^2} \tag{46}$$

The value of Equation 46 is the magnitude of the step response after the settling time.

With a step size of 0.125 radians and 4WS the RCV is driven at different speeds, triggering the step. The $\dot{\psi}$ is calculated with Equation 46, $K_{us} = -0,0012$ and compared to the actual yaw rate that was measured by the IMU.

As the operator was not able to steer the RCV during the test, there were some yaw rate present before the step. The *change* in yaw rate during the step is thus used for reference. Refer to Table 7 and Figure 23.

Clearly, the theoretical results corresponds well with the measured ditto.

Table 7: Tabular presentation of data from step steer runs.

Speed $[m/s]$	IMU $\dot{\psi}[rad/s]$	Derived $\dot{\psi}[rad/s]$
3,0	-0,33	-0,38
6,0	-0,74	-0,76
7,2	-0,95	-0,93

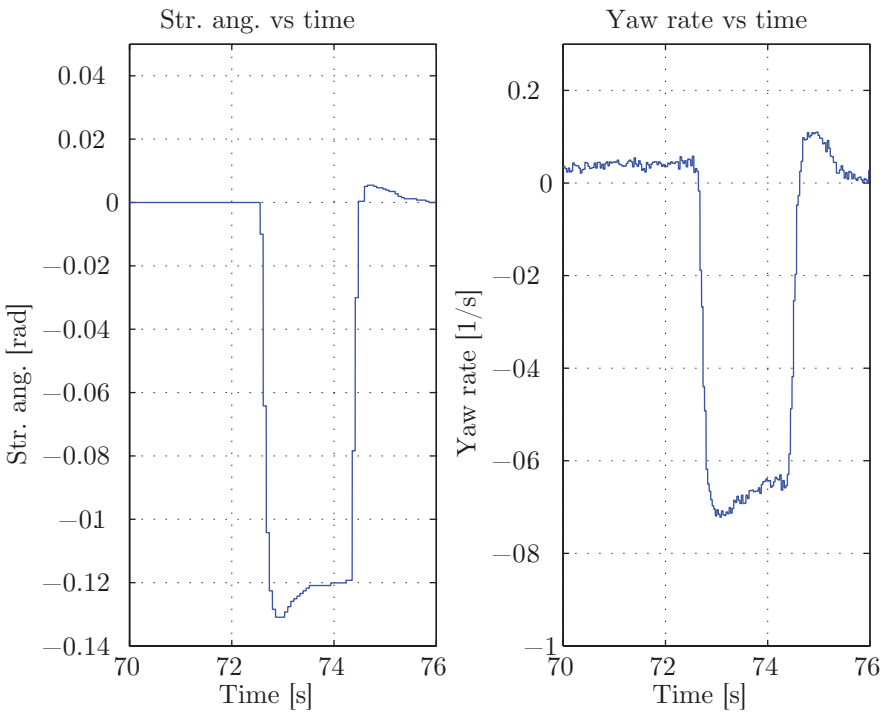

Figure 23: Steering angle of one of the wheels and yaw rate of the body during a step steer of 0.125 radians at a speed of 6 $^{m}/_{s}$.

4.5 Toe sweep

By letting the wheel reference angle follow a saw wave, in opposite signs left and right, so that the right wheel turns right when the left wheel turns left and vice versa, the dynamic properties of the wheels can be examined.

First, a theoretical analysis of the four bar linkage steering mechanism is made. The steering mechanism projected on the XY-plane is presented in Figure 24. In the figure the point of instantaneous rotation, IP, for the push rod is marked.

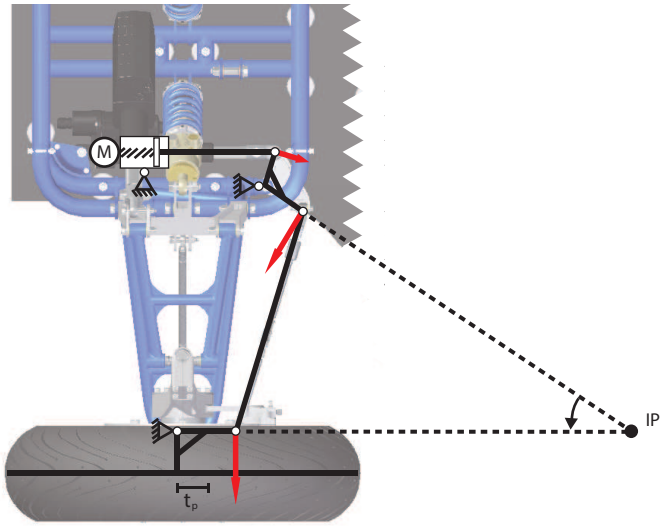

Figure 24: The steering geometry of the RCV front left wheel. All wheel have the same configuration, but mirrored on the right side of the vehicle, and with the push rod in front of the wheel instead of behind, in the rear. t_p is the pneumatic trail of the tire. IP is the instantaneous point of turn for the push rod. Red vectors are velocity.

The velocity of a point can be described by the equation:

$$\vec{v_A} = \vec{v_B} + \vec{\omega} \times \vec{r_{BA}} \tag{47}$$

This holds for all points in the mechanism. Although a trivial geometric problem, an analytical solution according to Equation 47 solving for steering angle as a function of actuator displacement could be cumbersome, therefore the system is simulated in MSC Adams.

The gear ratio is derived with help of the *Law of Power*, ignoring any internal friction, mass or moment of inertias; gives:

$$P = constant = \omega_{wheel} M_{wheel} = v_{actuator} F_{actuator} \tag{48}$$

which gives the gear ratio as:

$$\frac{\omega_{wheel}}{v_{actuator}} = \frac{F_{actuator}}{M_{wheel}} = n \ {}^{1}\!/_{\mathrm{m}} \tag{49}$$

Note that the dimension of the "gear ratio" is *per length unit* .

Refer to Figure 25 for the gear ratio for input of change in actuator position and the corresponding change in wheel angle.

As seen, the difference in gear ratio between the extremes is quite substantial. The simulated wheel angles are between the theoretical maxima given the mechanism's constraints. Still, for realistic angles, i.e. from -0.44 radians to 0.44 radians, the difference in gear ratio is 3.6 times. This means that one mm change in actuator position when the wheel is pointing far outward from the vehicle body, corresponds to a wheel angle change of 0.02 radians, and when the wheel is pointing far inwards, the wheel angle change for one mm actuator change is 0.006 radians.

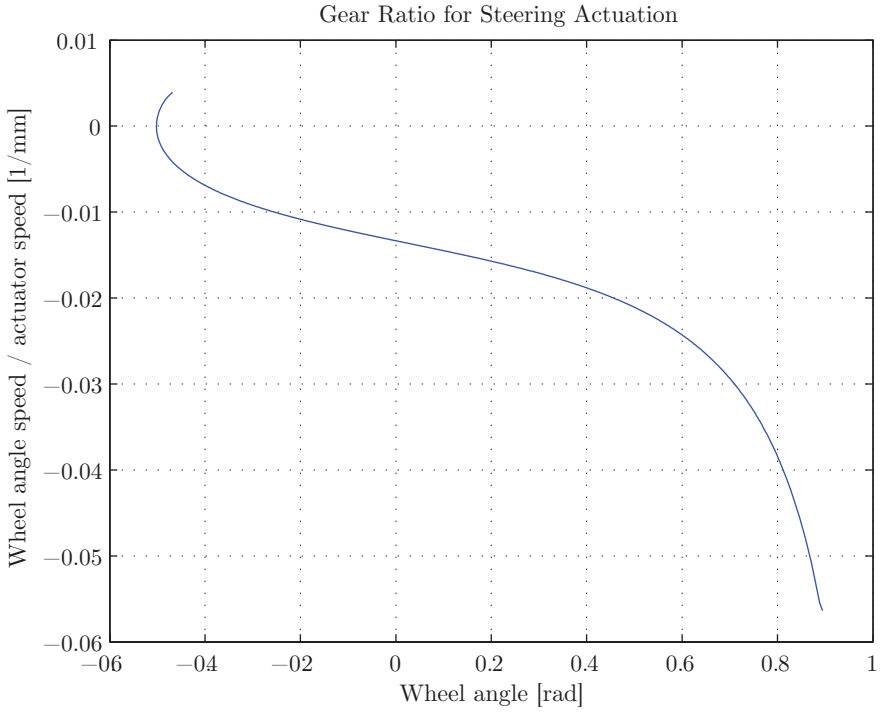

Figure 25: Change in wheel angle per change in actuator position as function of wheel angle. Data from MSC Adams simulation. Higher angle is outwards from the vehicle body.

In tests, the front wheel pair of the RCV was programmed to follow a triangle wave with magnitudes 0.09 radians and 0.15 radians. The forces in the tie rod

was measured with load sensors. Refer to Figure 26 for a run with a sweep magnitude of 0.09 radians and an initial speed of 1 $^{m}/_{s}$.

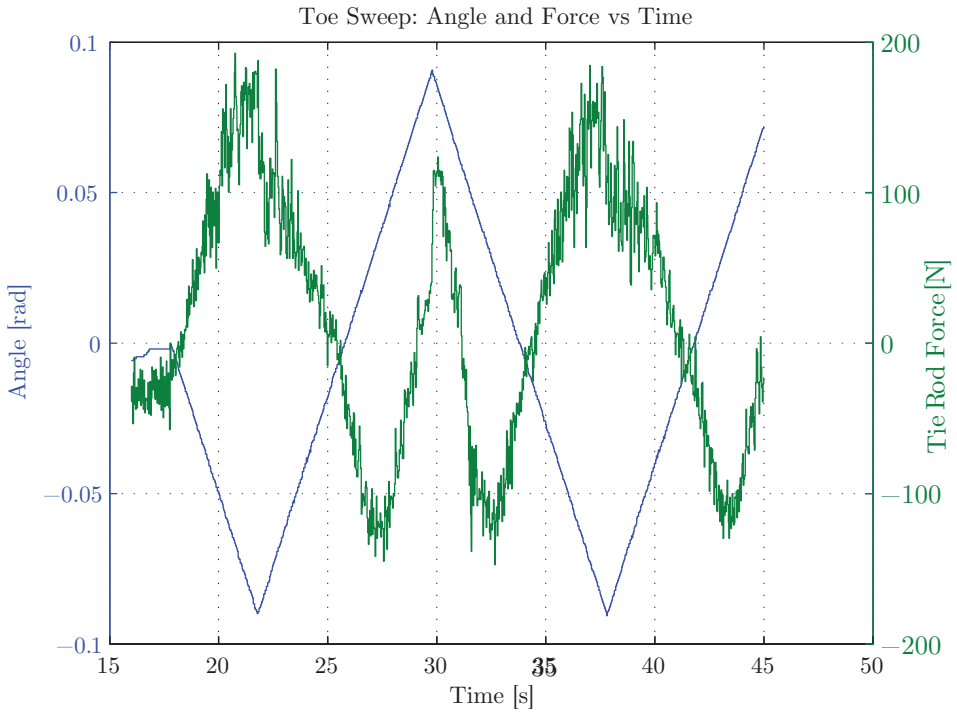

Figure 26: Toe sweep of 0.09 radians of the front right wheel. Angle blue, push rod force green. Positive angle is to the left inwards in respect of the vehicle body. Negative force is the rod pulling the wheel rear half inwards (trying to steer more right); positive is the rod pushing the wheel rear half outwards (trying to steer more left).

At a wheel angle of 0.06 radians inwards towards the RCV body the force on the push rod is zero and for higher angles the force in the tie rod is pulling. The Millikens [7, p. 31] states that for a car with no mechanical trail, the aligning torque can become negative. This could explain this behaviour. Due to the steer-by-wire steering, how ever, the operator will not notice this.

4.6 Roll out test

To measure the retardation forces acting on the RCV a roll out test is made. The vehicle is driven on a straight with no toe in on course asphalt and at 30 $^{km}/_{h}$ all motors are disengaged instantly by a press on a button on the joystick connected to the host PC.

The vehicle is brought to a halt and the procedure is repeated the opposite direction to even out some of test disturbances due to wind and road inclination.

The retardation forces acting on a free rolling vehicle on a flat surface with negligible slip can be describes by:

$$F_R = gmk_1 + \frac{\rho}{2}k_2\dot{x}^2 \tag{50}$$

where k_1 is the rolling coefficient and k_2 is the drag area. Drag area is defined as the drag coefficient times the front area. The rolling coefficient is depending on speed, but for speeds bellow 80 $^{km}/_{h}$, the coefficient can be regarded as constant [13, p. 114].

Newtons second law gives:

$$F_R = m\ddot{x} + I\alpha/r \tag{51}$$

where $\alpha r = \ddot{x}$. In accordance with subsection 4.3 the moment of inertia is estimated to 3 kgm^2, which is an equal mass of 33 kg - 6% of the total loaded vehicle mass of 600 kg.

Solving for $\ddot{x}$ gives:

$$\ddot{x} = \frac{gmk_1 + \frac{\rho}{2}k_2\dot{x}^2}{m + I/r^2} \tag{52}$$

To calculate the retardation, the signal from the wheel hall sensors are used, rather then the accelerometer signal, as the accelerometer measurements are hard to distinguish from gravitation at lower accelerations, even though the inclination is small.

To derive the retardation, a non-causal least mean square regression filter is used. At each velocity data point, a LMS regression gives the derivative over a time window taking into account both future and past data (i.e. non-causal). This approach gives lesser phase shift then a causal filter. Matlab code for this derivation method is supplied in Appendix E.

Refer to Figure 27 and Figure 28 for the unfiltered and filtered velocities.

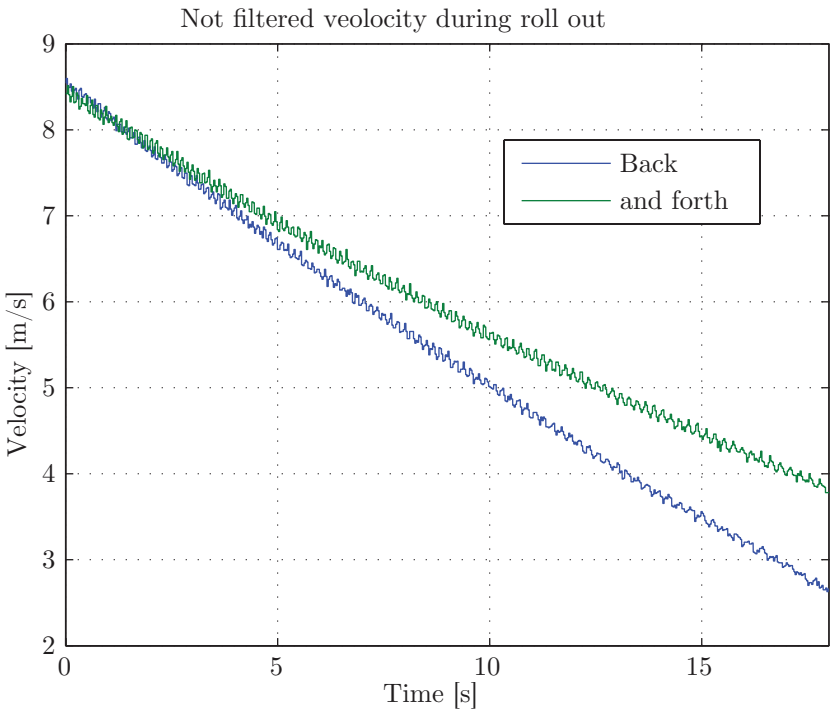

Figure 27: The non filtered velocity of the roll out test, in both directions. As seen, the retardation is higher when going back.

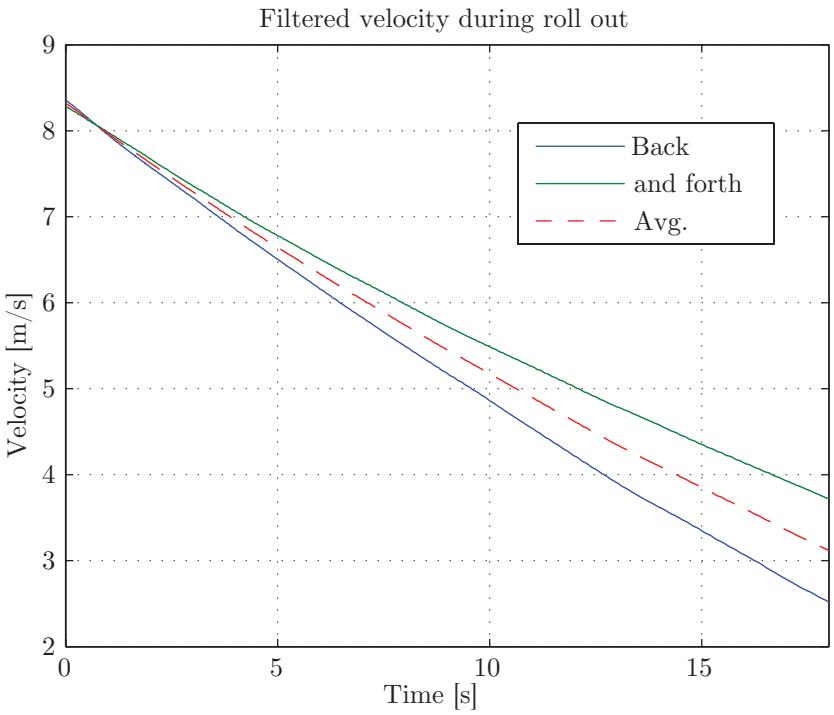

Figure 28: The filtered velocity of the roll out test. The average in between.

The acceleration, i.e. the derivative of the velocity, is estimated as the slope of the least mean square regression lines in each velocity data point.

Refer to Figure 29 for the derived acceleration during the roll out test.

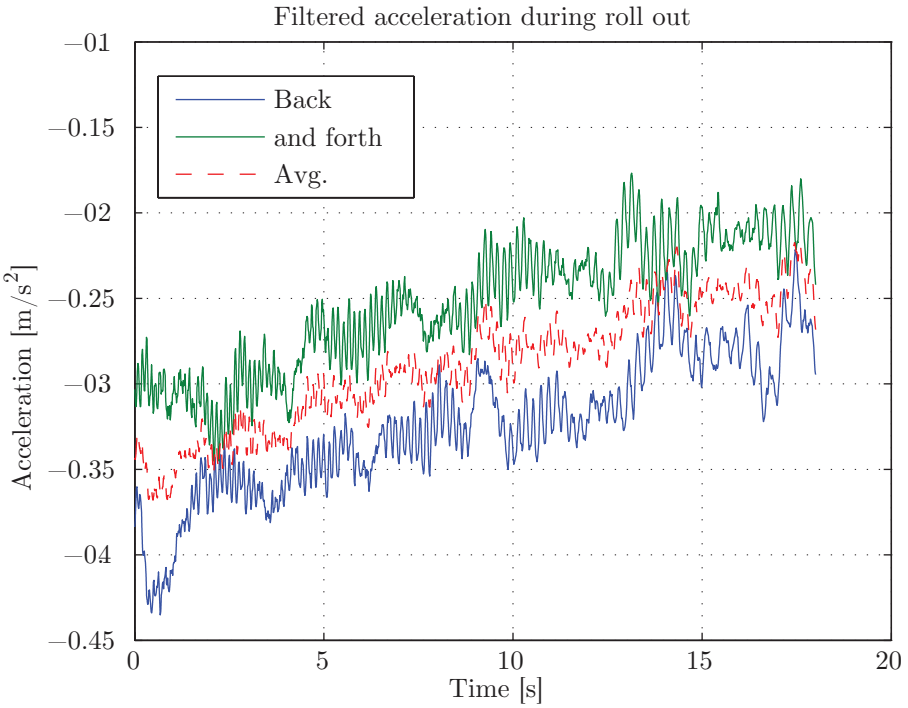

Figure 29: The filtered acceleration of the roll out test. The average in between.

To estimate the coefficients according to Equation 52, velocity squared is plotted versus acceleration and then a linear LMS regression is made. Refer to Figure 30.

The coefficient derived from the linear least mean square regression is p_{95} to two digits. The rolling coefficient k_1 is then calculated to 0.024 and the drag area coefficient is calculated, with the air density estimated to about 1,23 $^{\mathrm{kg}}/_{\mathrm{m}^3}$, to 2.1 m^3, which is fairly high, although expected due to the RCV non-streamlined design. A crude estimate of the frontal area is 1.73 m^2, which would give a drag coefficient of 1.21, which is about the same drag coefficient a standing person has [12] or about 3 to 4 times the drag coefficient of a production car.

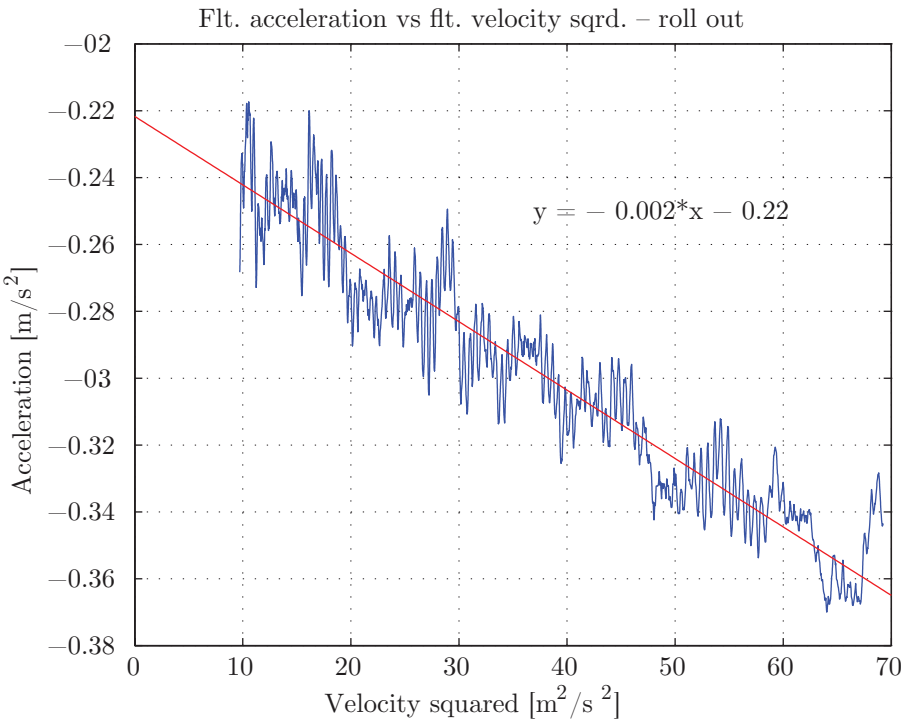

Figure 30: The filtered acceleration versus the velocity squared. The linear regression also shown.

5 Conclusion

The RCV proves it self to be a suitable platform for interdisciplinary research. There are good means for measurements together with data logging and subsequent post-processing. The RCV's PC interface is such that it offers flexibility as development environment for the software running the x-by-wire systems, as well as a debugging tool, and a GUI during runtime.

As a platform for vehicle dynamic research the RCV offers opportunities to do standard dynamic tests like cornering, roll out test, etc. in a more precise way. E.g. while performing a step steer test, the step can be triggered with a push off a button, and logged instantly. If the same test would be made in a non-drive-by-wire car, there would be a larger time lag then necessary, and, the driver would have difficulties to set the magnitude of the step correctly, which aggravates re-creation of the test.

Further-more, the RCV offers opportunities for dynamics testing not realizable with a 2WS or 2WD vehicle, like, crab steering, individual wheel torque allocation, toe sweep tests and so on.

As shown in subsection 4.5, the steering mechanism requires more force from the actuator when steering outwards then inwards and the caster changes sign with wheel angle change. Overall, the most problematic part of the RCV has been the steering.

As of the corner module design, not tempering with the paradigm, there are no rack connecting the tie rods of the wheels. This allows for arbitrary steering actuation with, e.g., optimal Ackermann steering, which, on a rack and pinion set-up, is suboptimally implemented as an approximation, which diverges from the optimal geometry as the steering angle increases due to geometrical constraints. However, a rack between the wheels balances any uniform torque on the steering coming from the scrub radii. A steering actuator of a cornering module need to be dimensionalized for this. The Max Jack actuators of the RCV are sufficient in speed and force, but lack thermal capacity and dissipation for continuous high power steering. New ground breaking design is, off course, associated with great challenges – as well as opportunities.

The RCV is maturing as a research platform – in the following concluding section, some possible enchantments are reviewed.

6 Suggestions for further developments

There are some main points of interest of which further refinement would be beneficial to the RCV as a research platform.

Higher grade GPS The EM-506A GPS chip currently fitted to the RCV's IMU has a too slow update rate; 1 Hz. There are widely available commercial grade chips with a update rate of up to 10 Hz, which, would greatly improve the means to do state estimations, e.g. body slip.

Steering actuation As of the summer 2014 rev. of the RCV, the steering actuation is the major bottleneck in dynamics performance.

First, the set-up of the actuators require a homing synchronization, that is, the actuator forces the wheel to the mechanical end position and notices when the actuator current goes over a threshold. In some situations, where the road is rough, etc., the current threshold might be exceeded before the actual mechanical end is reached, and thus the actuator will be misaligned. Another possible source of homing errors are bounces, as the actuator bounces of the mechanical end just before the current thresh-hold is reached, also resulting in misalignment.

A potentiometer or functionally equivalent to measure the position in absolute values rather then offset would be advisable.

Secondly, the motors driving the actuators are too weak, headmost in regard of thermal capacity. Even-though it also would be beneficial to increase the available torque and power, of more concern is the perseverance.

Thirdly, the steering angle limits of around 16 degrees inwards and 20 degrees outwards are suboptimal, both in regard of Ackermann geometry, in which the maximal inwards angle preferably is greater then the outward. With 2WS the turning radius is less then common vehicles, adjusted for vehicle size. With 4WS, however, the manoeuvrability is good compared to common vehicles.

There are some approaches to address these limitations that do not require a complete redesign of the suspension.

- The problem with Ackermann geometry could trivially be corrected by flipping the suspension so that the left side is right, and vice versa.
- The inwards steering angle of the wheel is constrained by the stop ring of the actuator ram hitting some armature. Redesigning the ring, e.g. cut off a relevant edge, would improve inwards steering.
- Another approach could be changing dimensions of the rocker.
- The Max Jack steering actuators could be enchanted to better suit requirements. A change to different actuators would most likely require a redesign of the suspension. During the fall of 2014, the Transport Lab will work with replacing the motors in the actuators with better suited ones.

Fourthly, the steering mechanism could be optimized for uniform gear ratio. Moving of joints, or the actuator, might solve the question.

Wheel caster The peculiar behaviour of aligning torque of the wheels, which changes from positive to negative during an inward turn of the wheel, as shown in subsection 4.5 on page 45, could render a wheel unstable, if a steering actuator breaks down. To comply with a specification of zero mechanical trail for modularity reasons (bi-directionality of the axes), this behaviour might be hard to mitigate. A mechanism to set caster dynamically could be beneficial, or, more straight forward, the steering actuators could be made self locking, which the Max Jacks are not.

However, practical tests show that the wheels maintain their general steering angles when power is cut to the steering actuators in cornering as well on a straight.

Mechatronical redundancy As a steer-by-wire vehicle, the RCV is totally dependent on its electrical, digital as analogue, subsystems.

While the hydraulic brakes always are accessible, effort should be put in ensuring redundancy in critical systems, as the MABX, the relays and the steering wheel. The steering wheel, for example, could have a total of three encoders connected to three CAN-drivers, to ensure that even if one encoder or controller fails, the MABX will still be able to tell what the steering input is. The relays could, for example, be duplicated into separate circuits in parallel, to ensure continuous operation even though one fails.

The MABX, which could be considered to expensive to have redundant duplicates of, could, for example, be complemented by a cheaper micro-controller, which listens on the CAN-bus for commands to the steering actuators, and if no such are present on the bus for what ever time limit, reads the steering wheel input on the bus and sends corresponding commands to the steering. Another possibility is off-course to replace the MABX with multiple cheaper systems working in parallel.

Steering angle calibration mechanism For proper calibration data, there is a need for a custom made mechanism aiding when calibrating the wheel angles to actuator positioning lookup tables. This mechanism could be as simple as a rod connected to the rim. The rod would then slide on another rod in front or in back of the vehicle, perpendicular to the vehicle's x-axis. On the rods there would be distance scales. From this, wheel angles could be calculated.

Oversteer compensation The RCV is greatly over-steered, especially in 4WS mode. This is mitigated to some point by movable Ackermann point steering, where the rotational centre of the vehicle is moved backwards as velocity is increasing. Further effort could be put into this.

Dashboard The dashboard, which is very minimalistic, lacks a speedometer which is in view of the driver. One would be beneficial for vehicle dynamics testing, as well as law abiding.

References

[1] E. Wennerström, S. Nordmark, B. Thorvald, 1999. *Fordonsdynamik*. Kungliga Tekniska högsklan, avd. Fordonsteknik, Inst. för Farkostteknik. Book.

[2] L. Camacho Silva *Modeling and Design of the Electric Drivetrain for the 2013 Research Concept Vehicle* KTH Royal Institute of Technology, Stockholm 2013

[3] Thomson Linear *Pro Series Electromechanical Linear Actuator* Installation and Operation Manual, 2010

[4] Thomson Linear *Max Jac Electric Linear Actuators* Product sheet/specification, 2013

[5] L.D. Reinius and M. Zeeshan, 2012.*Research Concept Vehicle Chassis Designing*. Student project report.

[6] M. Nybacka *RCV GPS/IMU Description* Product sheet/specification, KTH Royal Institute of Technology, Stockholm 2014

[7] W. F. Milliken and D. L. Milliken *Race car vehicle dynamics* Book, PA. USA, 1995

[8] D. Malmquist and J. Wikander *Optimal Design of Harmonic Drive Servo* IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2013, Article

[9] S. Zetterstrom *Vehicle wheel suspension arrangement* Patent EP1144212, 1998.

[10] M. Jonasson, S. Zetterström and A. S. Trigell *Autonomous corner modules as an enabler for new vehicle chassis solutions* FISITA Transactions 2006, paper F2006V054T, 2006 Part of Doc. thesis.

[11] Oskar, W. et al., 2014. *Design and Implementation of an Experimental Research and Concept Demonstration Vehicle*. In Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE. Coimbra, Portugal, pp. 1–6.

[12] A. D. Penwarden, P. F. Grigg and R. Rayment, 1978. *Measurements of Wind Drag on People Standing in a Wind Tunnel*. Building and Environment vol. 13, p. 75 - 84.

[13] Thomas D. Gillespie. *Fundamentals of vehicle dynamics*. SAE, Inc. 1992. Book.

A ControlDesk and its GUI for the RCV

Through ControlDesk the operator is able to access the memory of the dSPACE MABX at runtime. The connection between the MABX and the PC is through a proprietary serial interface, not Ethernet, even-though it is a standard RJ45 connector. A dSPACE Link Board or equivalent has to be used on the PC side to convert the signal.

To start ControlDesk, a hardware license key (a green USB-stick) has to be connected to the PC. The RCV GUI can be loaded by simply opening the project file.

Once loaded, the ControlDesk has to connect to the MABX. Given proper communication set-up, the GUI will go live if the "Start Online Calibration" button is pressed. If the firmware on the MABX differs from the firmware included in the ControlDesk project, you will be prompted whether the firmware should be uploaded to the MABX memory and restart the device with the new firmware. This will not flash the firmware. If the firmware do not differ, the MABX will not restart when the PC connects.

To flash a new firmware to the MABX right-click the "Platform" under "Hardware Configurations" in the "Project" controlbar. Then choose "Load Real-time Application to Flash".

The project in ControlDesk is linked to the Simulink project where the compiled binaries and variable descriptions will be stored after a compilation of the Simulink model. Mostly, ControlDesk will recognize when the Simulink project has been updated, but sometime, e.g. if the ControlDesk GUI was live during compilation of the Simulink model. To update the ControlDesk project manually, right-click the .sdf-file under the "Platform" and then press "Reload Variable Description". If the path to the Simulink project is changed, you'll need to update the ControlDesk project by instead choosing "Replace Variable Description".

The RCV GUI consists of multiple tabs. Each tab has a set-up of virtual so called instruments. To add an instrument, drag and drop it from the "Instrument Selector" controlbar. You will need to know what is the name and location in the Simulink model of the variable you want to connect the instrument to. Drag and drop the variable from the "Variables" controlbar to the instrument.

Data logging is done by adding the relevant variables to the so called recorder. The data polling should be handled by the background task of MABX, and not asynchronously by the PC (the data will have huge variation in sampling time if the PC is polling). The exact layout of the tabs are under constant change and will not be reviewed in further detail here.

B Post processing app. with GUI for Matlab

With a custom made application with a graphical user interface it is possible to process the exported data from the ControllDesk environment in Matlab. The application lists all the measured variables and the user can plot the variables, apply filters, scale them or provide an offset for calibration. The application is to be executed in a Matlab environment.

It is possible to export variables from the application to the Matlab workspace for further analysis.

The data in a .mat-file can also be accessed directly, without this application, by just loading it and then manually access the elements of the structure. Much of the functionality of the GUI can, of course, just as well be done via the Matlab interpreter.

The application interface is shown in Figure 31 and an explanation follows.

N.B.: The filters (except "FIR filter no time sync") and the differentiator assumes constant sample rate.

N.B.: When using the differentiator, any filter if chosen in drop-down-list will be applied before the differentiators own filter. The integrator does not have its own filter.

N.B.: Any offset is added after the scaling multiplication.

X-axis Chose one variable to plot on the x-axis in the list, or chose "Use time for x-axis" to use time for x-axis.

Y-axis Chose one or more variables to plot on the y-axis. Select more variables by holding down the ctrl-key.

Plot XY Plots the selected variables on the same y-axis versus a variable or the time on the x-axis.

Plot XYY Plots two selected y-variables on separate axes versus a variable or the time on the x-axis.

Plot GPS Plots the logged GPS coordinates converted to meters with north in positive y-direction and east in positive x-direction.

Plot Map Plots the logged GPS coordinates on a map. This feature requires an internet connection to download the map. Save your work before using this function as Matlab might crash.

Export vars Export the selected variables together with time to the Matlab workspace. The variables will have the prefix "export__". The filter and time window will be applied if enabled.

Plot derivative Calculate and plot the derivative of the selected y-variables. The variables together with time are exported to workspace with the prefix "export__d_". Note that the calculation of the derivative takes some time.

Plot integral Calculate and plot the integral of the selected y-variables. The variables together with time are exported to workspace with the prefix "export__i_".

Time window Select which part of the measurement that should be plotted or exported. The format is "startime stoptime". Enable the radiobutton "$-t_0$" to set the first time vector element as a new timestamp zero.

Variable scaling Chose whether there should be any scaling of the variables before plotting or export. The format is "$(k_{x1})k_{y1}k_{y2}...k_{yn}$" and k_{x1} should only be present if "Use time for x-axis" is not selected.

Variable offset Chose whether there should be any (calibration e.g.) offset on the variables before plotting or export. The format is "$(o_{x1})o_{y1}o_{y2}...o_{yn}$" and k_{o1} should only be present if "Use time for x-axis" is not selected.

Filter window time The time window to filters should smooth the data over. The filter of choose is chosen in the list directly to the right.

- No filter: Means, no filter. This is default.
- Non-causal FIR filter: This is a FIR filter that takes the average of both future and past signals around each data point within the time window. This is the recommended filter.
- FIR filter: This filter takes the average of past data points within the time window.
- FIR filter no time sync: Same as above, but, the filter does not assume that the sampling rate is constant. If the sampling rate is not constant, this filter is the only one that is usable.

Auto create new figures Select this if a new figure should be created each time a plot-button is pressed. Deselect this if a subplot is to be created.

Create\select subplot Creates a subplot or select a plot in a subplot. Specify rows and columns in the prompt window. Remember to deselect "Auto create new figures".

Select Figure Selects a figure with figure number n. That figure will then be subject to any changes made through the application. When a new figure is created it is automatically selected. To the right the presently selected figure number is shown.

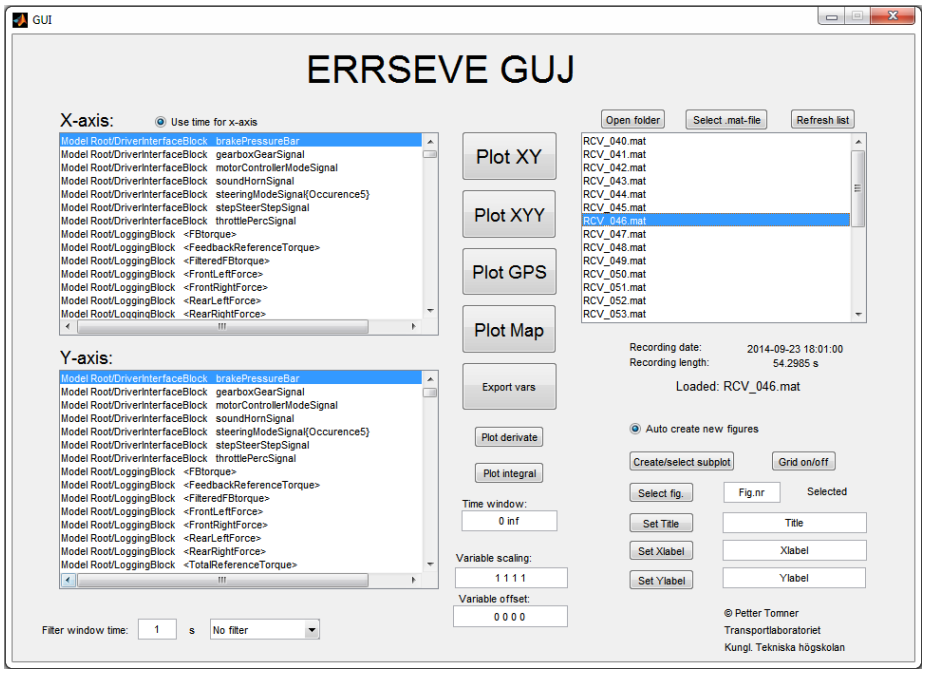

Figure 31: An overview of the application.

Set ... Sets the corresponding attributes of the selected figure.

Grid on\off Toggles the grid on the selected figure.

Open folder Select a working folder with recorded measurements. Then the measurement files can be selected and loaded in the list bellow. It is also possible to transverse folders. ".." is the parent folder of the current folder.

Select .mat-file Select a specific .mat-file with measurements.

Refresh list Refreshes the view of the current working folder.

C IMU calibration and bearing calculation

The RCV IMU accelerometer is calibrated in the following way. Assure that the IMU is on a flat surface and completely still. Press the joystick to the left and then press it (into the IMU, like a button). During a period of time, specified by IMU_N_CALIBRATION_TIME in imu.ino when the firmware was compiled, at default 3.2 seconds, the IMU will calculate the average of the signals of the gyro and accelerometer axes. This average will then be subtracted from the measurements before they are sent via the RS232 serial connection to what ever receiver.

The magnetometer is calibrated in the following way. First, recompile the IMU Mega2560 software with COMPASS_CAL set to "true" in the main file. The IMU will then send raw magnetometer data for the x,y and z-axis through the USB port as newline row terminated ASCI. The default baudrate for the USB is 57600 $1/s$. Place the IMU in the space in which it will operate.

It is possible to do either 3D or 2D calibration of the magnetometer. For a 2D calibration rotate the IMU around it is XY-axis at least two turns. Then use the x- and y- data to fit an ellipse, on the form:

$$1 = (x - o_x)^2/a^2 + (y - o_y)^2/b^2 \tag{53}$$

,where o_i is the offset from origo, a and b are axes with the unit . This could be solved as a linear least square problem; a solution for a 2D calibration is presented in Figure 32. The ratio between the small and big axis in the best fit is 0.97.

In a similar way the 3D problem of fitting the data to an ellipsoid can be solved given equation:

$$1 = (x - o_x)^2/a^2 + (y - o_y)^2/b^2 + (z - o_z)^2/d^2 \tag{54}$$

,which applied to calibration data gives a best fit as shown in Figure 33.

A 3D calibration is required for being able to compensate for roll or pitch according to eq. 32. The 2D calibration is easier to perform, as the IMU can be steadily rested on a flat surface, where as in the case of 3D calibration nicety is limited by the more or less firm hand of the person holding it. Some sort of rotating device is probably needed for high precision 3D calibration.

Due to the large and dynamic internal magnetic fields in an electrical vehicle, the compass bearing shouldn't be used for anything critical. Heading should instead be calculated using GPS position extrapolation. Refer to Figure 34 for an example on how high current disturbs the compass.

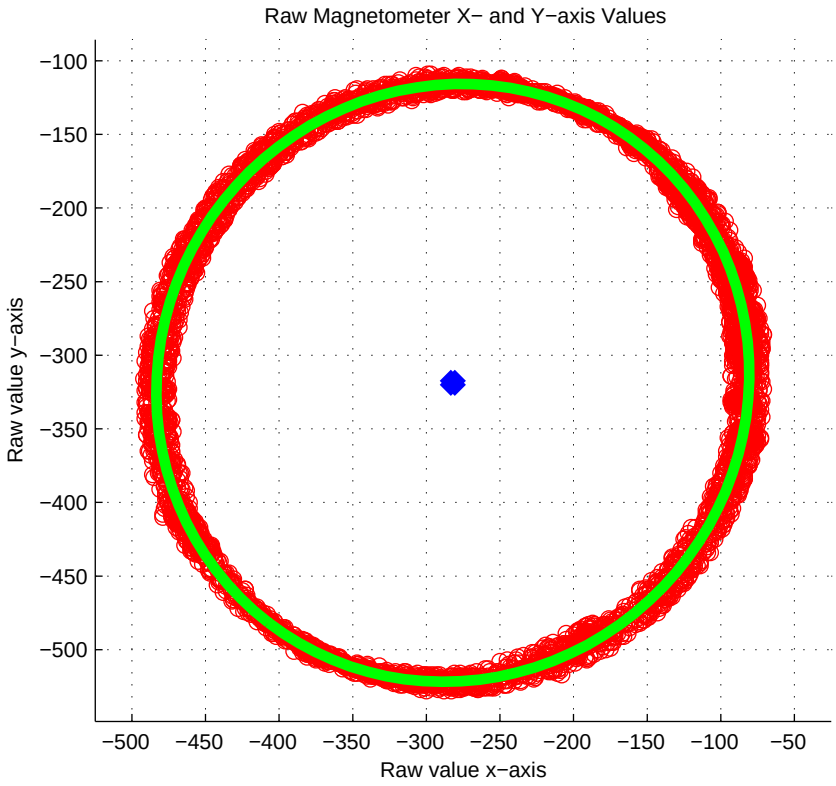

Figure 32: A best fit of an ellipse to the raw values of the magnetometer x- and y-axis data for some rotations in the xy-plane – centre of ellipse shown in the middle. 5 000 data points used.

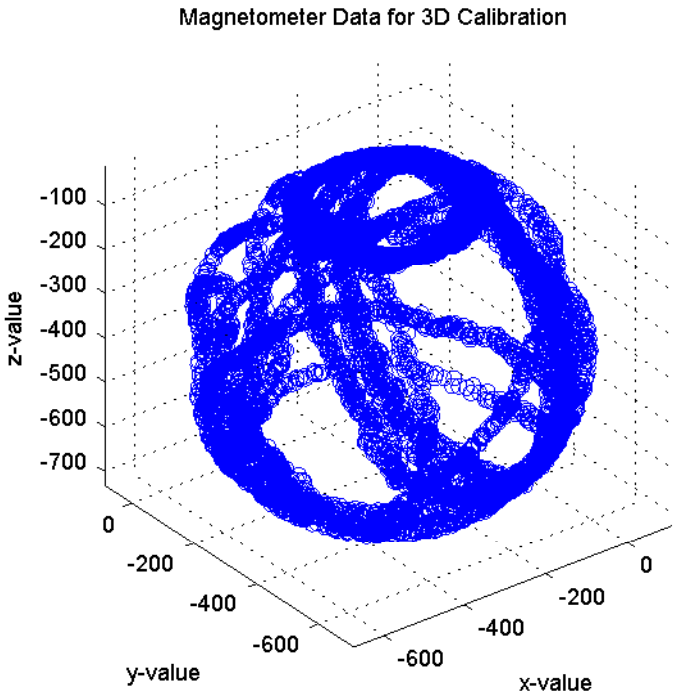

Figure 33: A best fit of an ellipsoid to the raw values of the magnetometer x- and y- and z-axis data for some rotations in space. 10 000 data points used.

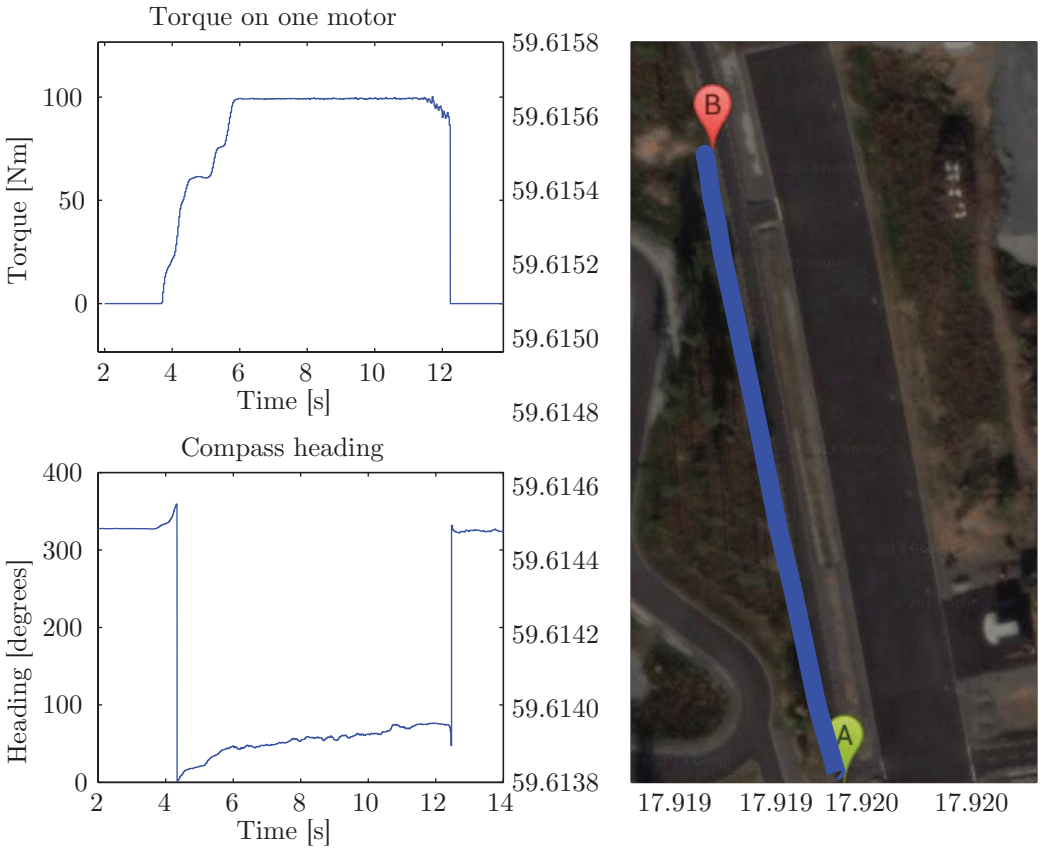

Figure 34: The compass heading changes 80 degrees eastwards as the motors are engaged, clearly showing why a compass should not be used on an electric vehicle. The RCV travels in a straight path, as seen on the GPS-map.

D IMU hash validation

The RS232 connection between the IMU and the MABX has proved to be somewhat noisy and to prevent undiscovered transmission errors a simple hash function has been implemented.

The message broadcasted from the IMU is specified in Table 4. The hash is calculated in the following way:

- For each byte 0 to 53, sum them and overflow to 0 from 256 for each addition (still add the remainder to the 0 after the wrap around). This sum is the first hash byte.
- Add the first hash byte to it self with an overflow to 0 from 256 for each addition. This sum is the second hash byte.

This hash functions is guaranteed to discover one faulty byte in the received message.

E Non-causal least mean square filter

A implementation of a non-causal least mean square filter is presented bellow:

```
function [dyfilt, yfilt,tfilt] = NonCausalLinReg(y,t,td)
%%%%%%%%%%%%%%%%% Non causual linear regression filter %%%%%%%%%%%%%%%%%%%%
% The filter takes data from "the future" when calculating the smoothing.
% Thus a non-causal filter.
%
% Outputs:
% dyfilt - filtered derivative
% yfilt - filterer value
% tfilt - corresponding time to filtered value
%
% Inputs:
% y - vector of data to be filtered
% t - vector of the time points where y were sampled
% td - the time window to smooth the signal over
%
% y has to be sampled at a somewhat equal pace, otherwise you'll have to
% rewrite the script with interp1().
%
% td is the timewindow to smooth over. The closest odd multiple of the mean
% sample time will be choosen as actual time window.
%
% (c) Petter Tomner, 2014
% Transport Lab, Kungliga Tekniska h'o'gskolan
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

it = length(y);
mdt = mean(diff(t));
n = round(td/mdt);
if mod(n,2) == 0 %want to have odd n
    n=n+1;
end

h = waitbar(0,'Processing non-causal Linear LMS regression derivation');

yfilt=zeros(1,it-1*n);
tfilt=zeros(1,it-1*n);
dyfilt=zeros(1,it-1*n);
for i=1:it-1*n
    tmp = polyfit(t(i:n+i)',y(i:n+i)',1);
    %tmp = fit(t(i:n+i)',v(i:n+i)','poly1'); %this one is really slow
    yfilt(i)=tmp(1)*(t(i+round(n/2)))+tmp(2) ;
    tfilt(i)=t(i+round(n/2));
    dyfilt(i)=tmp(1);
    if mod(i,100)==0
        waitbar(i/(it-n));
    end

end

close(h);
```