

CHALMERS



Motor Control System Design for Electric Vehicles Master of Science Thesis in Embedded Electronic System Design

VICTOR PÅSSE

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, June 2014

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Developing the systems needed for an electric vehicle

The design, implementation and testing of the hardware and software needed to propel an electric vehicle with respect to efficiency, performance and safety.

VICTOR PÅSSE
© VICTOR PÅSSE, June

Examiner: Sven Knutsson

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover: Picture of modified Sinclair C5 about which the thesis is based
Department of Computer Science and Engineering

Göteborg, Sweden June 2014

Abstract

The use of efficient vehicles is crucial to meet the populations mobility demands. It is therefore important to conduct research and development on, for example, electric vehicles to contribute to the continued evolution of these vehicles. In this master thesis an electric vehicle has been fitted with developed hardware, such as motor drivers and control computers with the purpose of controlling the vehicle with respect to performance, safety and efficiency. Several software features, such as torque vectoring and regenerative braking, has been implemented and evaluated. The resulting hardware and software works as designed, performs well and can control the vehicle correctly and safely. The results of this report can and has been used to further develop other electric vehicles.

Acknowledgements

I, Victor Påsse, a student at Chalmers Tekniska Högskola, am extremely grateful to ÅF for the confidence bestowed in me by entrusting and financing my project entitled "Motor Control System Design for Electric Vehicles". I would also like to express my gratitude to my external supervisor Martin Lörne who has helped me through this project and guided me when problems has occurred. I would also like to thank many of the employees at ÅF for their help as well as my internal supervisor Sven Knutsson for his guidance.

Victor Påsse, Göteborg 14/05/28

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Vehicle	1
1.2	Aims	2
2	Method	3
2.1	System design	3
2.1.1	System layout	3
2.1.2	Operational margin	5
2.1.3	Fault containment	5
2.1.4	Safety	5
2.1.5	Legal	6
2.1.6	CAN bus protocol	6
2.2	Hardware	7
2.2.1	Hardware design tools	7
2.2.2	MotorDriver	8
2.2.3	ControlComputer	8
2.2.4	FrontNode	8
2.3	Software	9
2.3.1	Software design tools	9
2.3.2	MotorDriver	9
2.3.3	Anti Spin and Anti Lock Braking	11
2.3.4	Electronic Stability Program	11
2.3.5	Torque Vectoring	11
2.3.6	Regenerative Braking	11
3	Results	14
3.1	Hardware	15
3.1.1	MotorDriver	15
3.1.2	ContolComputer	22

3.1.3	FrontNode	24
3.1.4	SoftStart	27
3.1.5	Sensors	29
3.1.6	SoundRacer	29
3.2	Software	30
3.2.1	Stability Program	30
3.2.2	Torque Vectoring	32
3.2.3	Regenerative Breaking	33
3.2.4	Anti Lock Breaking	33
3.2.5	Anti Spin	33
4	Discussion	34
5	Conclusion	36
	Bibliography	37

Abbreviations

BEMK	Back ElectroMotive Force
BLDC	BrushLess Direct Current
CAN	Controller Area Network
EMI	ElectroMagnetic Interference
ESD	Electro Static Discharge
ESR	Equivalent Series Resistance
IMU	Inertial Measurement Unit
PCB	Printed Circuit Board
PID	Proportional Integral Derivative
PWM	Pulse Width Modulated
RPM	Revolutions Per Minute
RTOS	Real Time Operating System
SD	Secure Digital, memory card manufacturer
VGS	Voltage from Gate to Source

1

Introduction

This master thesis has been executed at ÅF which is a consult company with consultants in many different technical fields. The reason for ÅF to request this master thesis is to use the resulting vehicle as advertisement for the company at various gatherings and use it as an educational development platform for the employees.

1.1 Background

Electric vehicles are slowly replacing their internal combustion based counterparts and may one day be the standard. It is therefore important to develop methods for controlling the motors in electric vehicles in an efficient and safe way. This must be done from the lowest level of motor drivers to the highest level of the control systems to guarantee the safety and efficiency of the whole motor control system.

1.1.1 Vehicle

ÅF currently have a modified Sinclair C5[1], which has been fitted with BLDC motors in the rear wheels and a LiPo battery pack. An original Sinclair C5 is shown in Figure 1.1 for reference.



Figure 1.1: A Sinclair C5, the mechanical base for this master thesis.

ÅF wants a new drive system for the vehicle that demonstrates some of the functionality that could be implemented on a full-size electric vehicle such as different forms of vehicle stabilization systems and regenerative braking. The vehicle will, after completion, be displayed in official events and be used as an educational platform for employees and also for further development.

1.2 Aims

This master thesis aimed to develop the whole chain of both hardware and software components needed to drive and control the electric motors located in the rear wheels of a modified Sinclair C5 electric vehicle with regards to efficiency, performance and safety. This was done by developing and constructing two motor drivers, a central vehicle control computer and several sensors. These units were connected via a local network. Software has been developed to control the hardware and to aid the driver with vehicle handling enhancement functionality.

2

Method

2.1 System design

When designing a complex system it is important that the system is well structured and designed in a way which allows it to be expanded to include more subsystems if needed. Because of this, the system is designed to be modular, with each subsystem forming a node which is connected to the rest of the system via a bus. This design methodology allows for more subsystems to be added to the system without making any changes to the other subsystems. This also allows each subsystem to be independently modified as long as it is backwards compatible with respect to communication as the subsystems only share the bus and power.

2.1.1 System layout

The complete system is broken up into subsystems which reduces the complexity of the system. The subsystems are represented by the graph in Figure 2.1.

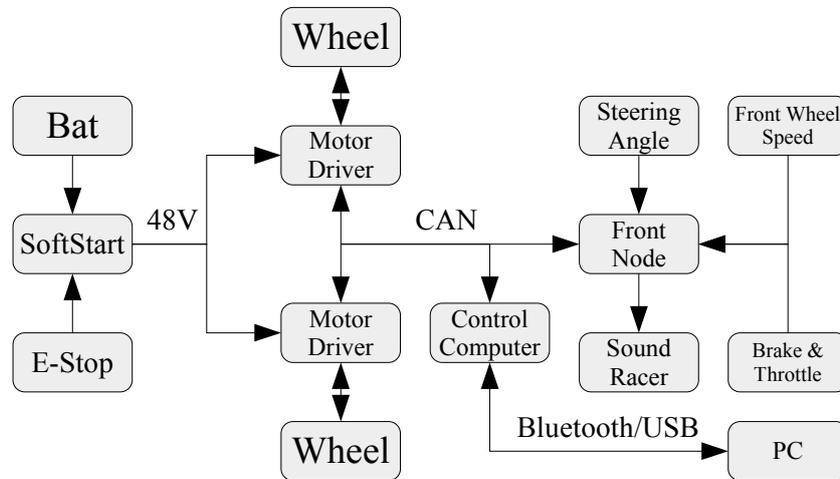


Figure 2.1: High level hardware system description

The main nodes in the system are

- **MotorDriver**
Based on speed and torque values from the control computer, the motor driver directs power from the battery to a motor. It also measures state data such as motor rotational speed, motor current and battery voltage and sends it to the control computer.
- **ControlComputer**
The control computer calculates target torque values for the motors using a set of vehicle control programs based on vehicle state. It also sends vehicle state to a PC or other external device via Bluetooth or a serial link. The control computer also has the ability to log vehicle state to SD card and contains a local IMU to measure vehicle orientation and external forces.
- **FrontNode**
The front node uses sensors to measure parameters such as throttle and brake pedal angle, steering angle and front wheel speed and sends these to the control computer. It also controls the SoundRacer unit based on data from the control computer.
- **SoundRacer**
This node designed and provided by ÅF generates the sound of an internal combustion engine based on a RPM value input. This unit will not be covered extensively in this report as it does not serve any functional purpose.
- **SoftStart**
The soft start limits the maximum current from the battery for the first second

after start to reduce current inrush as an effect of the large decoupling capacitors located in the motor drivers. It also reads the dead-man's handle and start/stop button and connects or disconnects the battery from the system accordingly.

2.1.2 Operational margin

One way to improve the reliability of a system is to design the given system with a high operational margin, this means that all components should be able to survive a harsher environment than they will experience during normal operation. This method can be applied to, for example, the components in the motor current path by increasing the maximum tolerated current above the maximum expected current. The same method can also be applied to software by adding code to cope with events that should not happen when operating normally but that could appear in case of faults or failures. Many features like the one described has been implemented in the hardware and software of the vehicle.

2.1.3 Fault containment

If one node fails other nodes that don't depend on the failed nodes services should not fail. This is called fault containment. As there are only small amounts of redundancy in the system many of the faults will lead to a system failure. This is however not true for the motor drivers as there are two motor drivers. The services provided by the motor drivers are power supply for the CAN bus, wheel state information and power to the motor. Some of these services such as supplying the CAN bus with power can be moved to the other motor driver in case of failure. This is implemented in the design.

2.1.4 Safety

One ever increasing design concern regarding vehicles is safety. As the vehicle can control its own speed and brakes it is important that this is done in a safe way and according to what the driver wants. Certain measures have been taken to decrease the probability that the vehicle misinterprets the drivers commands or fails due to errors. These measures include:

- Giving the brake pedal precedence over the throttle pedal to allow the driver to brake if the throttle pedal jams in an activated position.
- Detecting the failure of a sensor based on its reading.
- Shutting down the power to all subsystems if the driver falls of the vehicle.
- Defaulting sensitive data, such as throttle and torque, to a safe state if no new data is present on the CAN bus for a certain amount of time.

2.1.5 Legal

There are laws and regulations regarding vehicles like the one in this report. The vehicle is classified as a "Bike without pedals" and must therefore comply with, among others, the following rules according to TSFS 2010:144 chapter 3 [2]:

- 2§ The vehicle should be steerable
- 3§ The vehicle should be safe and the driver should have a good view of the surrounding
- 4§ The vehicle should be easy to operate
- 5§ The vehicle should be able to be turned on and off
- 9§ The vehicle should be at most 1.6x0.75m in area
- 11§ The vehicle should be able to decelerate with a retardation of at least $3\frac{m}{s^2}$
- 12§ The vehicle should brake if the throttle is released
- 13§ The vehicle should have a parking brake that is able to hold the vehicle at a standstill on a inclination of 15°

There are other regulation as well but only a subset of the rules will be covered in this report. Most of the rules are basic and the vehicle comply with those rules automatically, these rules are for example paragraph 2-5. The vehicle have been designed to comply with the rules 11 and 13 by ensuring that the brakes are strong enough and that some object, for example a battery, can be removed and used as a parking brake by putting it in front of a tire. The measurements of the braking force is covered in 3.1.1. There are two paragraphs that the vehicle does not comply, namely 9 and 12, the vehicle is 1.8x0.8m and does not brake when the throttle is released. Compliance with paragraph 12 can be easily implemented in software but it would not be suitable for this type of vehicle as the throttle and brake should behave as they do in a car. Applying the brakes when the throttle is released could result in a vehicle that is difficult to operate and the vehicle would thereby break paragraph 4. All things considered, the vehicle is not legally allowed to operate on public roads as it is too large and does not engage the brakes when the driver releases the throttle.

2.1.6 CAN bus protocol

All nodes need to either send or receive vehicle state data to or from other nodes, this is done by using a CAN bus. CAN only specify the lowest level of the communication standard and there exist several protocols for use on a CAN bus. This project uses no previously existing protocol but instead utilizes the fact that the only data to be sent on the CAN bus are scalars describing some form of sensor reading or signal to an actuator. A CAN frame contains, among other parts, an id field and a payload section which can

be up to 8 bytes long. Each scalar that are to be sent are given a unique identifier which is used as the id when sending that scalar on the CAN bus. The scalar itself is placed in the payload section and the format for the scalar can be signed or unsigned 8, 16 or 32bit fixpoint or a IEEE 754 float. Several scalars can be placed in the same package if the size does not exceed 8 bytes. This should however only be done if the scalars are closely related otherwise the readability could be degraded. There are many other standards to use but most of them rely on a state full system to send the CAN packages eg. several CAN frames are needed to make up a single data transmission. By implementing a state less CAN transmission system the drawbacks of a state full system, such as complexity and liability to crash or hang, can be mitigated.

2.2 Hardware

The hardware design process starts with the specifications of the system and results in finished hardware that fulfills the specifications. The process can be split up in to several steps such as

- High level system design, the entire system is broken up in to self contained subsystems that interact with each other.
- Low level system design, the components needed in each subsystem are identified.
- PCB schematic design, the schematics of PCBs in each subsystem are designed.
- PCB layout design, the layout of PCBs in each subsystem are designed.
- Subsystem testing, each subsystem is tested to ensure that it works according to the specifications.
- System testing, the final system is tested to ensure that it works according to the specifications.

If some subsystem does not fulfill the specifications when tested, there are three methods that can be used to make the system fulfill the specifications. The subsystem can be modified, redesigned or the specifications can be changed. To execute the PCB related design steps a PCB design tool is needed, the other steps does not require any special design tools.

2.2.1 Hardware design tools

The PCBs covered in this report has been designed in CadSoft EAGLE [3], which can be used in a free and limited mode to design non-commercial 2 layer PCBs that are smaller than 100x80mm. There are other similar tools such as KiCad [4] as well as more professional tools that requires a purchased license such as Altium Designer [5]. The decision to use EAGLE is based on previous knowledge of that software.

2.2.2 MotorDriver

The motor driver hardware subsystem takes power from the battery and drives the motor based on data coming from the CAN bus. The motor driver is implemented on two PCBs, the power driver and the control board. The design decision to separate the board is taken to increase the distance between the power driver, which generates high amounts of EMI, and the control board, which is sensitive to EMI. This separation also makes it easier to insert ferrite material around the cable that connect the PCBs to limit the amount of conducted high frequency noise. The power driver contains all the high current components such as MOSFETs, MOSFET drivers, DC/DC converters, fuses, current sensor and so on. The control board contains all the control related components such as the Cortex M3 based microcontroller, CAN transceiver and filters. The power driver PCB uses a thicker copper layer of 105 μ m compared to the 35 μ m of the control board, this is to reduce the inductance and resistance of the high current traces to reduce the EMI and power loss. The two PCBs are stacked and mounted in a water tight box with space for a fan to increase the cooling if needed.

2.2.3 ControlComputer

The control computer subsystem is the main computer which executes all the vehicle stability software. It is implemented on a single PCB that contains the Cortex M4 based microcontroller, an IMU to measure force and rotation rate of the vehicle, power supply to supply power locally in the node, micro SD card to log data on, bluetooth to send logged data via and CAN transceiver to interface to the CAN bus. There are some points of concern on this PCB such as the decoupling for the microcontroller as it runs at a high frequency of 168MHz and requires high current peaks. Another point of concern is the bluetooth antenna which requires a special shape of the ground plane for proper operation.

2.2.4 FrontNode

The front node hardware subsystem has two different purposes. It reads sensor outputs and reports their value on the CAN bus and it reads engine RPM on the CAN bus and sends it to the SoundRacer subsystem. The SoundRacer contains a CAN transceiver, however no code for utilizing it is currently available, this is the reason for connecting the SoundRacer via the front node. The front node subsystem is implemented on a single PCB on which a power supply to supply power locally in the node, microcontroller to run the software and input filters for all the sensors are mounted. The input filters are designed in such a way that they can be implemented as a lowpass filter, highpass filter, voltage divider or any combination out of those depending on the type of sensor connected to the front node. For example the reading from a sensor with a 0-10V signal output could be scaled to 0-3V, the ADC range, and lowpass filtered to avoid aliasing of high frequency noise. The front node is designed to be a general purpose unit with 8 analog inputs, 12 digital inputs/outputs, 2 open-drain outputs and a CAN bus.

2.3 Software

Most of the subsystems in the vehicle needs software to define their behaviour. The different purposes of the subsystems results in different demands of the software running on the subsystems. For example the motor driver needs the software to meet tight deadlines for the commutations to run smoothly, these deadlines are in the order of microseconds. The control computer needs many pieces of software such as vehicle control tasks and communications tasks to run independently of each other with quite loose deadlines, typically a few milliseconds. The requirements of the control computer allows for the use of a RTOS to simplify the software development while the motor driver would not get the benefits of a RTOS.

2.3.1 Software design tools

There are two types of software related to the vehicle, the software running on the vehicle and the software running outside of the vehicle. The software running on the vehicle includes the code in the motor driver, control computer and front node which are written in C using the "CooCox" IDE [6] whereas the software running outside of the vehicle includes the monitoring program running on a PC which is written in C# [7] using VisualStudioExpress C# [8]. C was chosen for all the on vehicle software as it is the largest industry standard and supported by most IDEs targeting the microcontrollers used in the vehicle. Instead of using CooCox, Eclipse, arm-gcc and openOCD can be used to achieve the same results. C# was chosen for the PC based program because of previous knowledge and familiarity with the syntax and IDE. The control computer runs the real time operating system CoOs [9], there are many alternatives such as chibios and freeRTOS but CoOs was chosen because of previous familiarity with that RTOS.

2.3.2 MotorDriver

The motor drivers software is 100% interrupt driven, meaning that when an event occurs that needs CPU time that event is executed if no other event of higher priority is currently executed. This allows for some form of priority based execution of code but if the CPU burden is too high then events may be delayed. There are several types of events that can occur, for example the Hall-effect sensors, reading the angle of the rotor in the motor will trigger the highest priority interrupt when the motor driver needs to do a commutation while some events are time driven and of lower priority such as sending data to the CAN bus.

With the equation 2.5, derived from equation 2.4, 2.3, 2.2 and 2.1, it is shown that the torque of the motor can be controlled by choosing a duty cycle based on the battery voltage, motor RPM and K_v . K_v is a constant and the battery voltage and RPM are measured. This allows the motor driver to receive an input torque value and drive the motor accordingly for both positive and negative values of input torque. A positive torque would result in acceleration and using power from the battery while a negative

torque would result in a deceleration and charging of the battery using power from the motor.

$$\tau_{Motor} = K_t * I_{Motor} \quad (2.1)$$

in which

T_{Motor} Motor Torque
 K_t Nm per A Motor Constant
 I_{Motor} Motor Current

$$I_{Motor} = \frac{V_{Driver} - V_{BEMK}}{R_{Motor}} \quad (2.2)$$

in which

I_{Motor} Motor Current
 V_{Driver} Driver Output Voltage
 V_{BEMK} Motor Back ElectroMotive Force
 R_{Motor} Motor Resistance

$$V_{Driver} = DutyCycle * V_{Battery} \quad (2.3)$$

in which

V_{Driver} Driver Output Voltage
 $DutyCycle$ Driver Duty Cycle
 $V_{Battery}$ Battery Voltage

$$V_{BEMK} = RPM * K_v \quad (2.4)$$

in which

V_{BEMK} Motor Back ElectroMotive Force
 RPM Motor Revolutions Per Minute
 K_v V per RPM Motor Constant

$$\tau_{Motor} \propto (DutyCycle * V_{Battery}) - (RPM * K_v) \quad (2.5)$$

in which

T_{Motor} Motor Torque
 $DutyCycle$ Driver Duty Cycle
 $V_{Battery}$ Battery Voltage
 RPM Motor Revolutions Per Minute
 K_v V per RPM Motor Constant

2.3.3 Anti Spin and Anti Lock Braking

Anti spin and anti lock braking are forms of traction control systems. Traction control tries to avoid losing traction by reducing the torque signal to the motors when traction is lost. This is done by monitoring the front wheel speed and the rear wheel speeds, if there is a large enough difference between the speed values then traction is assumed to be lost as any of the wheels are rotating too fast or slow. When this happens the torque signal to that wheel is reduced which allows the tires to slow catch up with the vehicle speed grip the ground more effectively.

2.3.4 Electronic Stability Program

The driver commanded yaw rate of the vehicle is a function of the steering wheel deflection angle and the vehicle speed. In reality this is not the case as the vehicle can understeer and oversteer. This means that the vehicle is turning less or more than the driver wants. The yaw rate of the car can be measured as well as the speed and steering wheel angle to calculate if the car is steering too much or too little and the differential mode signal to the rear wheel torque values can be used to correct the unwanted yaw rate. This is done by the ESP or Electronic Stability Program.

2.3.5 Torque Vectoring

To enhance the steering performance of the vehicle the rear wheels can be used to increase or decrease the yaw rate by adding a differential mode torque signal to the rear wheels. The magnitude of the torque signal can be a function of any vehicle state information such as steering angle deflection and speed. By implementing Torque Vectoring the control envelope of the vehicle can be increased.

2.3.6 Regenerative Braking

To lower the speed of the vehicle some of its kinetic energy must be transformed to either heat which is the conventional form of braking or chemical energy in the battery which is referred to as regenerative braking. Regenerative braking stores the energy of the vehicle in the batteries and that energy can later be used to accelerate the vehicle. This form of braking is preferred over conventional braking as it is more efficient and produces less heat. The voltage generated by the motor, V_{BEMK} , as it spins can be estimated by the equation 2.6. This voltage will normally be lower than the battery voltage which means that the voltage generated from the motor needs to be boosted to a higher voltage than that of the battery in order to charge the battery. This can be accomplished by using the coils in the motor as the inductors in a multiphase boost topology power converter.

$$V_{BEMK} = \frac{RPM}{Kv} \quad (2.6)$$

in which

V_{BEMK} Motor Back ElectroMotive Force

RPM Motor Revolutions Per Minute

K_v V per RPM Motor Constant

The motor driver can be modelled by 6 switches as shown in Figure 2.2. This model will be used to describe the process of boosting the voltage from the motor to charge the battery which can be broken down to the following two states.

- State 1. Close switch S4 and S2, current will build up in the coils connected to S4 and S2 given that the motor spins.
- State 2. Close switch S4 and S1, the current that has been built up in the coils will circulate in the loop connecting S1, the battery, S4 and the coils themselves. This will charge the battery with the energy previously stored in the coils.

The ratio of the time spent in the respective states controls the amount of current used to charge the battery. The maximum charge current is bounded by the rotational speed of the motor according to equation 2.7 in which R_{MOTOR} is the resistance of a motor coil, R_{DRIVER} is the total resistance of the motor driver and $R_{BATTERY}$ is the resistance of the battery. To brake with a force over the regenerative bound is possible by applying the mechanical brakes. This can however only be done by the driver as the vehicle cannot control the mechanical brakes by itself.

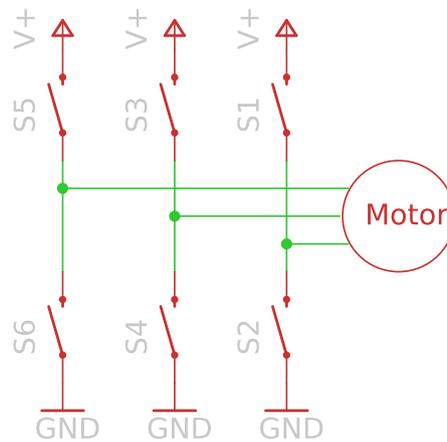


Figure 2.2: Model of motor driver for regenerative braking

$$I_{Max} = \frac{V_{BEMK}}{2 * (R_{Motor} + R_{Driver} + R_{Battery})} \quad (2.7)$$

in which

I_{Max} Maximum Brakeing Current

V_{BEMK} Motor Back ElectroMotive Force

R_{Motor} Motor Resistance

R_{Driver} Driver Resistance

$R_{Battery}$ Battery Resistance

3

Results

The completed vehicle is shown in Figure 3.1. In this final configuration all of the electronics are mounted, the chassis has been reworked to accommodate the motors and the body has been painted.



Figure 3.1: The final Vehicle

Determining whether or not the goals has been fulfilled requires evaluation of the results with respect to the projects aim. The results can be broken down in several segments such as hardware, software and control systems and the method for evaluating

the success of the results depends on the type of system being evaluated.

Hardware can typically be evaluated based on the specifications and whether or not the hardware passes each part of the specification. To give a more complete picture of the performance of the hardware, tests need to be carried out in different types of environments by for example varying the supply voltage, temperature and amount of vibration to simulate a range of operating conditions. The methods used to test the hardware on the vehicle are depending on the subsystem and are covered in the result section for the particular subsystem.

Software can be tested by creating test vectors for the functions and comparing the observed output with the desired output for the given vectors. Another, similar, method is simulating sensor inputs to excite the system and observing the outputs, this is basically the same thing as mentioned previously but the complete system is tested instead of subsystems of the software. The methods used for testing the software running on the vehicle are quite basic and not formally complete. The general method used could be described as a empirical test method or "if it seems to work, it is deemed working".

The control systems has been evaluated by driving the vehicle with and without them and comparing logs of sensor reading to determine if any gain in performance has been achieved.

3.1 Hardware

All of the hardware mentioned in section 2.2 has been fully implemented and tested. The different hardware subsystems has been tested to varying degrees. Simple hardware subsystems such as the sensors and front node has been verified to work but not tested for a range of different environmental parameters. More complex hardware subsystems such as the motor driver subsystem has been rigorously tested using a mechanical motor brake bench and tested in the full input voltage range and temperature range from about -20°C to 40°C ambient temperature.

3.1.1 MotorDriver

The motor driver is the most stressed piece of hardware as it has to control a high amount of power in a controlled and correct way for the vehicle to function. Any design flaws in the critical parts of the motor driver will have a high risk of resulting in a failure of the motor driver as many components are highly loaded and slight timing variations can result in cross conduction in MOSFETS and thereby a critical failure. Because of this sensitivity to errors it is important to test and verify the motor driver thoroughly.

The schematics and PCB layouts for the driver and controlBoard is shown in Figure 3.4, 3.5, 3.2 and 3.3 respectively.

The schematics for the controlBoard, seen in Figure 3.2, contains several sections such as CAN transceiver, power regulation, Hall sensor filtering and the microcontroller. By splitting the schematic into different parts the complexity of each part is lowered making the reading and understanding of the schematics easier.

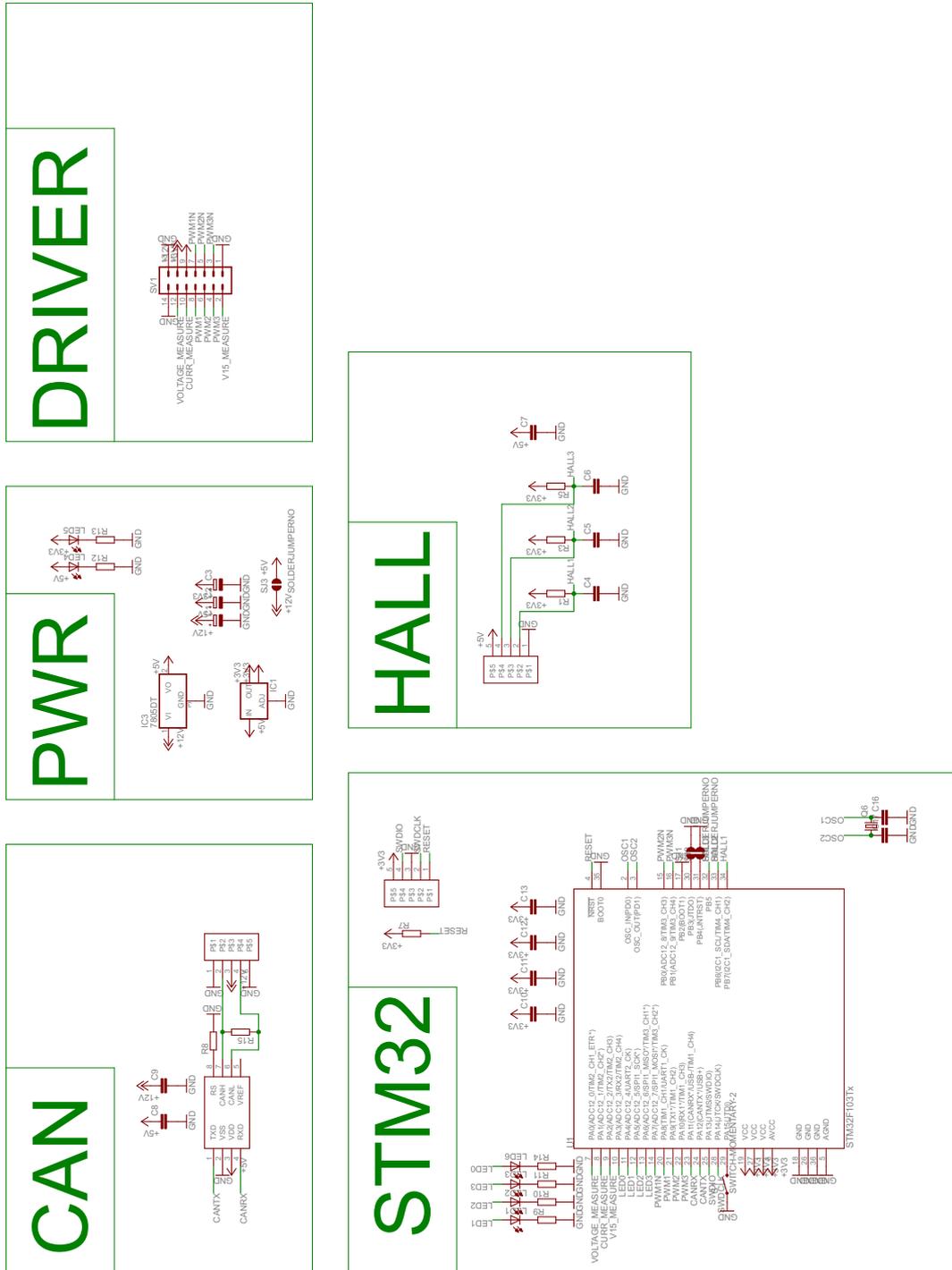


Figure 3.2: Motor driver ControlBoard schematics

The layout of the the control board, seen in Figure 3.3, did not require any special care except for keeping the linear voltage regulators at a low temperature. This was accomplished by placing solid copper planes connected via several thermally conductive vias to the pins of the voltage regulators that are the carrier for the silicon substrate and thereby lowering the thermal resistance from silicon die to ambient air. This design method for keeping the components generating a lot of heat at a low temperature is present through out all of the PCB designs.

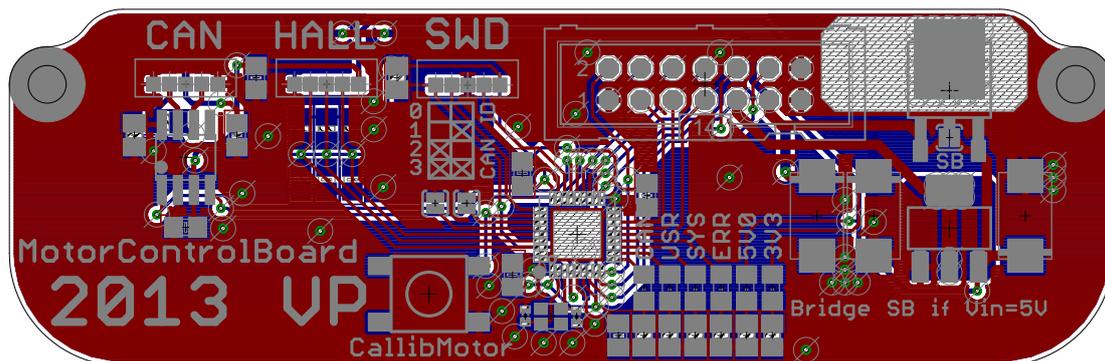


Figure 3.3: Motor driver control board layout

The schematic for the driver board, seen in Figure 3.4, contains several sections, just as the control board schematic. The more interesting sections are the MOSFETs section which contains the motor driving MOSFETs and the drivers section which contains the MOSFET drivers and their decoupling. It is important that the MOSFET drivers are correctly decoupled as they drive high peak currents into the gate of each MOSFET. The MOSFET drivers supply voltage must not exceed 14V as the drivers would then be destroyed or fall below 8V as the MOSFETs would not turn on completely. The driver board supplies power to all other boards on the vehicle, this is done by using switch regulators to lower the main battery voltage of about 48V to 12V for the gate drivers and 5V for all the other nodes. Another important feature of the driver board schematics is the presence of low impedance ceramic capacitors and bulk electrolytic capacitors, these are added to minimize the current running along the supply wires to the driver at high frequencies. Most of the high frequency power needed to drive the MOSFETs on and off quickly are supplied by these ceramic capacitors. The medium frequency power is supplied by the large electrolytic capacitors and the low frequencies are supplied by the batteries. This allows the current running on the cables between the batteries and drivers to be frequency limited and thereby reducing the EMI of the cables.

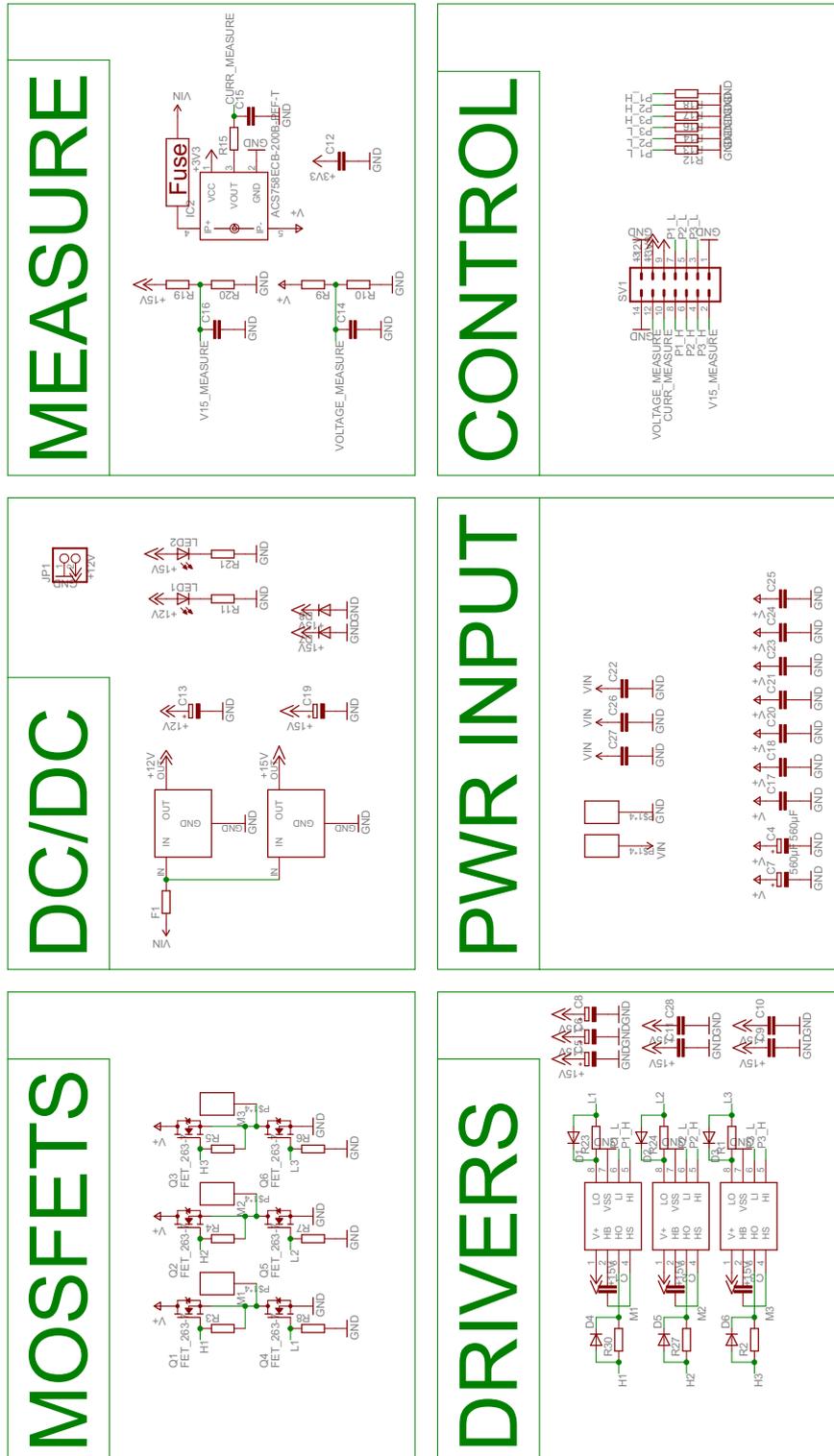


Figure 3.4: Motor driver bottom board schematics

The layout of the the driver board, seen in Figure 3.5 contains several important features. The main objective of the driver board is to supply the motor with power in a controlled fashion without producing too much EMI or heat. This was done by minimizing the resistance and inductance of the traces that carries the high currents. The highest current is present on the traces that connect the MOSFETs to the motor windings, this current can be much higher than the current from the battery as the driver and motor operates as a buck converter. This is especially true when the motor is running at low speeds as the BEMK of the motor is low. The motor driver and motor can be seen as a buck converter driving the BEMK and when the BEMK is low and duty cycle high the current on the traces running from the MOSFETS to the motor can reach several hundred amperes. The simplest way of minimizing the resistance and inductance of a trace is to make the cross sectional area of the trace as large as possible and the trace length as short as possible, this has been done by using a thick copper layer of $105\mu m$ and a trace width of about $10mm$. The traces connecting the MOSFETs to the main power input are also as wide, thick and short as possible, some parts of these traces have their stop-mask removed and a thick coating of solder added to further reduce the resistance and inductance. Another trick to lower the resistance and inductance of a trace is to route the same trace on several layers of the board, only two layers can be used on this board as it only contains two layers, routing traces on both sides of the board has been done for some high current segments. Another important aspect of the PCB layout is to place the ceramic and electrolytic capacitors close to the MOSFETs and power terminals to reduce the path length of the high frequency currents as those generates high amounts of EMI. Having a too low impedance path to the low impedance ceramic capacitors can however be bad as the currents will have higher slew rates which also generates EMI. There are another design concern regarding the layout of the driver board, namely thermal performance. The MOSFETs, electrolytic capacitors and to some extent the MOSFETs drivers generates heat which must be dissipated in some way. The motor driver node is enclosed in a watertight box which restricts the possibility of cooling the components. The only way to cool it is to try to keep the thermal resistance of the driver board as low as possible to allow heat to go from the heat generating components to the entire PCB and use the larger surface are of the PCB to radiate and conduct the heat to the enclosure and thereby keeping the temperature of the hottest components as low as possible. The temperature in the enclosure is monitored and the highest noted temperature inside the box is only a few degrees higher than that of the outside. The mounting holes on the PCB are positioned to allow the PCB to be mounted in a specific box.

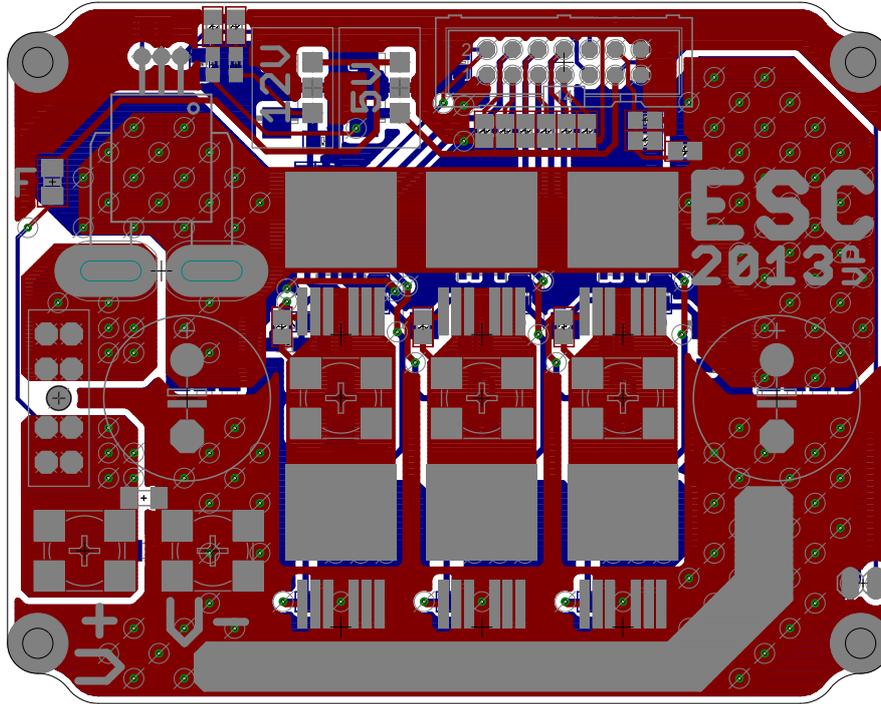


Figure 3.5: Motor driver driver board layout

Testing and verification of the hardware was done by first testing every feature without any load such as the sensors, MOSFET drivers, commutation timing and MOSFET gate rise/fall time. The most critical things to test are the MOSFET gate rise/fall curve, if the gate drive circuitry is not correctly designed the MOSFETs can either turn on/off too slow and thereby be operating in the resistive region and waste power or turn on and off several times resulting in excessive ringing on the VGS voltage. It is also important to have the right amount of dead-time when changing the active transistor in each half-bridge, if the dead-time is lower than the rise time and the reverse recovery time of the body diodes in the MOSFETs cross conduction will occur, potentially destroying the driver, and if the dead-time is too high excessive amounts of power will be dissipated in the drivers as the efficiency is lowered.

The performance of the motor driver perceived by the user can be split in two - acceleration and retardation. The retardation has been measured by logging key vehicle state data such as speed, pedal engagement and acceleration while braking the vehicle from full speed to a standstill at a reasonable pedal engagement. The results of this test are shown in Figure 3.6. The minimum retardation according to the law is marked to show that the retardation performance is as required. The legal aspects are covered in 2.1.5.

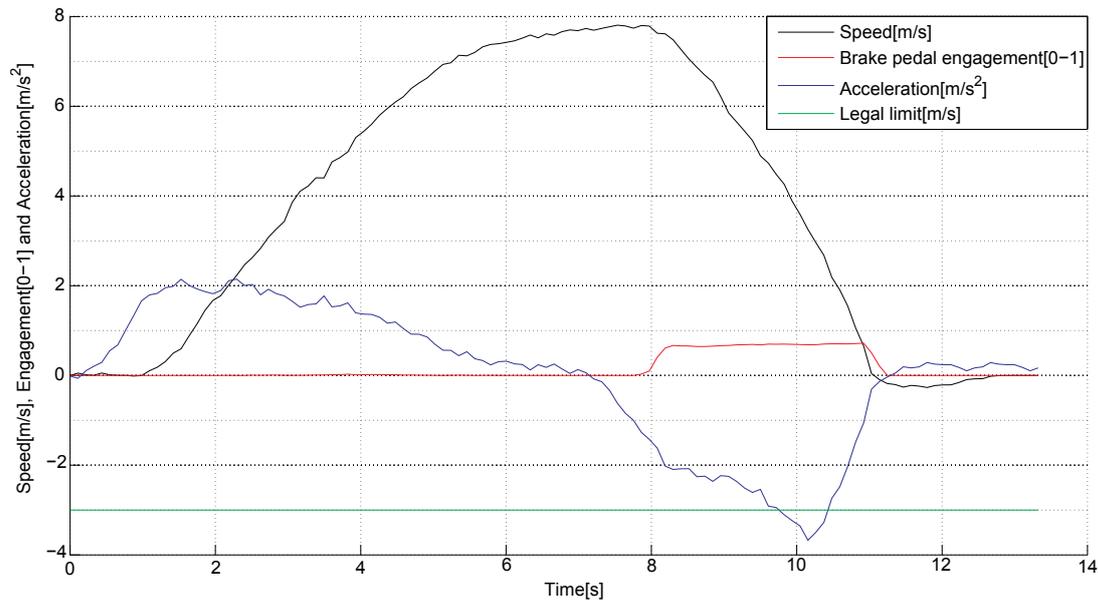


Figure 3.6: Plot of vehicle state while breaking from full speed

As seen in Figure 3.6 the time it takes to brake the vehicle from its top speed of about $8 \frac{m}{s}$ is about 3s which gives a mean retardation of about $2.7 \frac{m}{s^2}$ and peak retardation of about $3.5 \frac{m}{s^2}$ which is sufficient as any higher levels of retardation would not feel comfortable in the vehicle, nor is it required by law.

The acceleration has also been measured by logging key vehicle states such as speed, pedal engagement and acceleration while accelerating the vehicle from a standstill to full speed at a reasonable pedal engagement. The results of this test are shown in Figure 3.7. The acceleration ramps up with the pedal engagement and evens out at about $2 \frac{m}{s^2}$ until the continued acceleration would require a higher battery voltage and as a result of the limited battery voltage the acceleration falls down to zero as the vehicles speed evens out at the maximum speed of about $8 \frac{m}{s}$. An acceleration of about $2 \frac{m}{s^2}$ is sufficient but higher acceleration can be attained by changing the maximum current allowed to the motors, the motor drivers has been tested up to about three times higher current than those attained in this test.

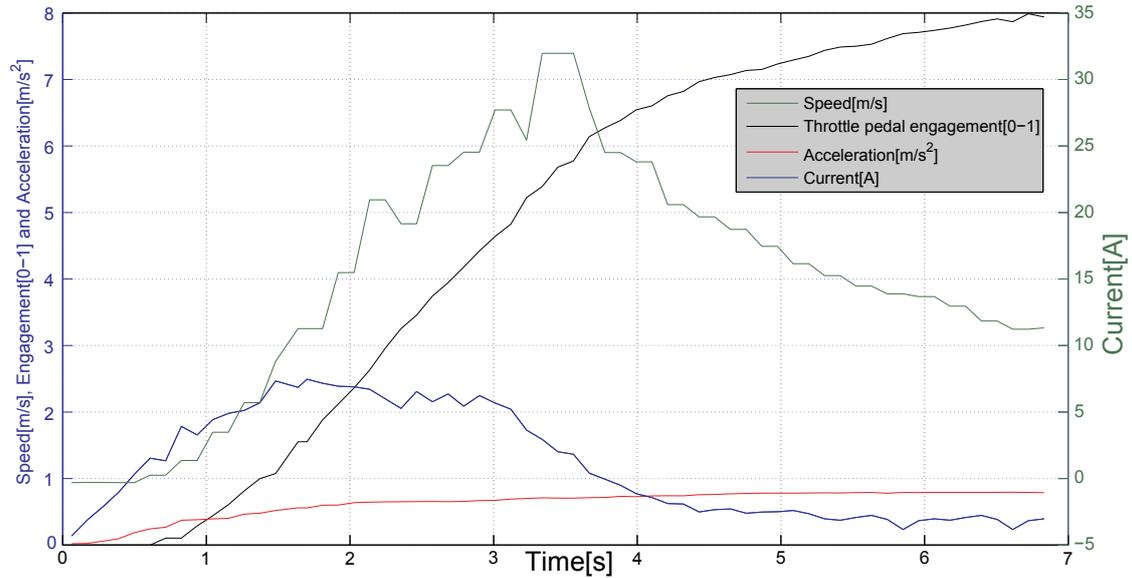


Figure 3.7: Plot of vehicle state while accelerating to full speed

3.1.2 ContolComputer

The Control computer contains a powerful, floating point capable Cortex M4[10] based microcontroller from ST Microelectronics, IMU(gyro and accelerometer), SD card socket, bluetooth module, CAN transceiver and power regulation as well as some other miscellaneous parts. A powerful microcontroller is needed in the control computer node to execute the vehicle control programs in floating point math at high speed and to be capable of running potential future software.

The schematics of the control computer is shown in Figure 3.8. The microcontroller has many unused pins. This is because most microcontrollers with a CPU running at high speed are packaged in high pin count packages. The control computer needs much computing power but only a few GPIOs which leads to many unused pins as there were no microcontrollers available from ST with the M4 cortex core and few GPIOs at the time when the control computer was designed.

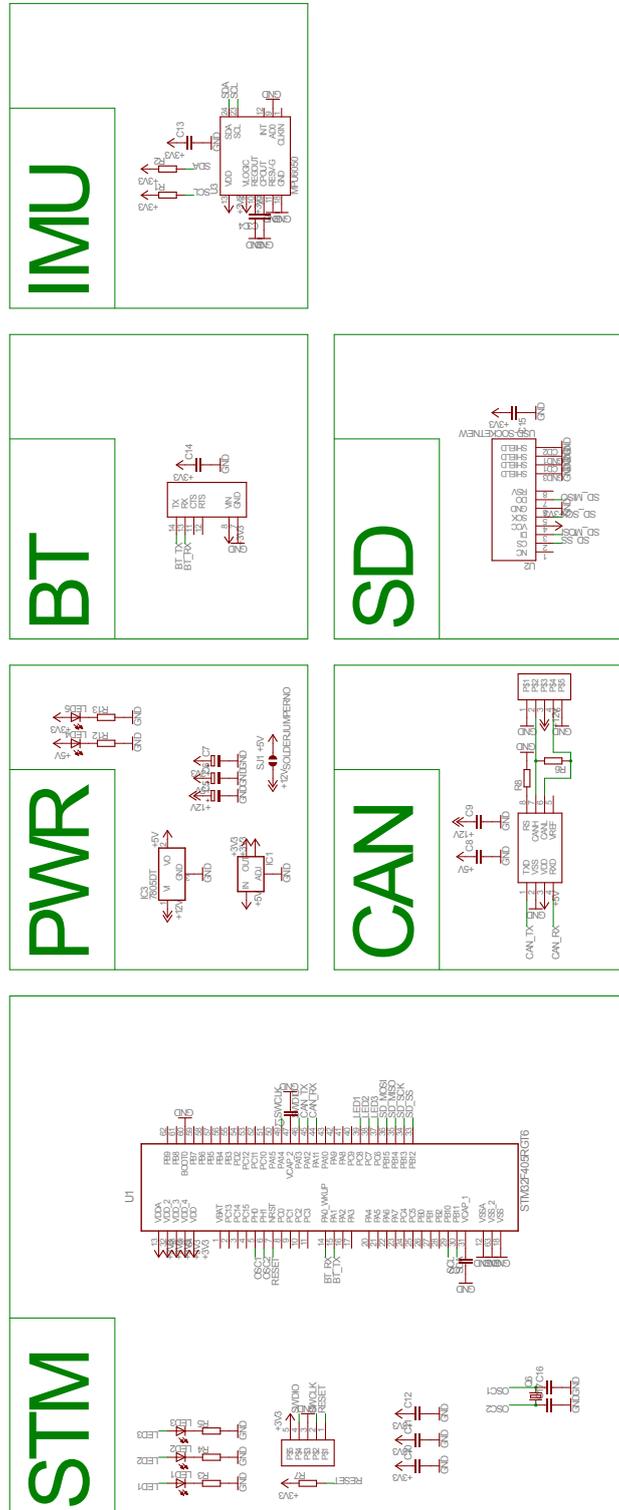


Figure 3.8: Control computer schematics

The PCB layout of the control computer is shown in Figure 3.9. There are some design decisions behind the PCB layout. These include removing the ground planes around the antenna of the bluetooth module, placing the IMU at right angles relative to the enclosure as the enclosure is placed at right angles relative to the lateral and longitudinal directions of the vehicle which removes the need of applying any coordinate transform to the data from the IMU and placing many decoupling capacitors close to the microcontrollers voltage supply pins. There are many LEDs located on the PCB. This is to make debugging code and hardware easier as the microcontroller can show the state of different variables or which part of the code that is currently being executed with the LEDs. Two LEDs are connected to the voltage rails. This allows the user to quickly evaluate if the voltage rails are at the correct voltage or not. Most of the PCBs in the more complex nodes have many LEDs to ease debugging. The mounting holes on the PCB are positioned to allow the PCB to be mounted in a specific box.

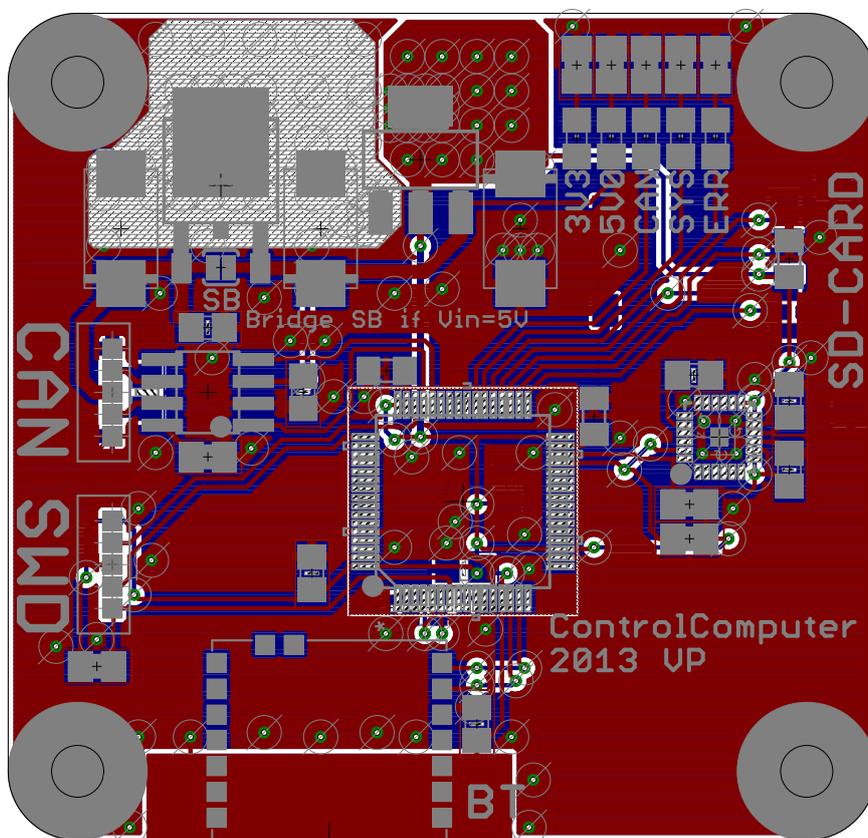


Figure 3.9: Control computer layout

3.1.3 FrontNode

The front node module acts as a hub between the CAN bus and all the sensors in the front of the vehicle and the SoundRacer node. Some sensors have an analog output

signal and some have a digital output signal. These signals are routed to the front node, filtered to remove any noise and measured by the microcontroller in the front node using either the ADC or GPIO peripheral. Analog signals comes from, for example, the steering angle sensor or pedal engagement sensors whereas digital signals comes from, for example, the front wheel speed sensor. The values from the sensors are then scaled and sent on the CAN bus. Any packages on the CAN bus containing information regarding the SoundRacer node are received by the front node and the signals to the SoundRacer are changed accordingly.

The schematics of the front node is shown in Figure 3.10. The schematic is, as the previous schematics, broken up in subsections to increase the readability. The sections include the microcontroller, CAN, power regulation and input/output filtering parts. The front node and Motor driver are based on the same microcontroller to simplify the reuse of code between the subsystems. By sharing parts between several nodes the workload can be lowered. Each input channel can be filtered by a lowpass filter with a gain of ≤ 1 by configuring passive components connected to the inputs. These filters can for example be used to scale the reading of a 5V sensor to 3.3V, which is the maximum voltage that should be applied to the microcontroller, and lowpass filter the signal to prevent high frequency noise being aliased down to low frequencies when sampled.

The PCB layout of the front node is shown in Figure 3.11. The layout shares many features with the motor driver and the control board seen in Figure 3.4 and 3.2 such as the CAN transceiver, voltage regulation and microcontroller section. The voltage regulation components can get hot so the same precautions as in the motor driver control board has been taken by placing large copper pours thermally connected to the regulators to act as heat sinks. The mounting holes on the PCB are positioned to allow the PCB to be mounted in a specific box.

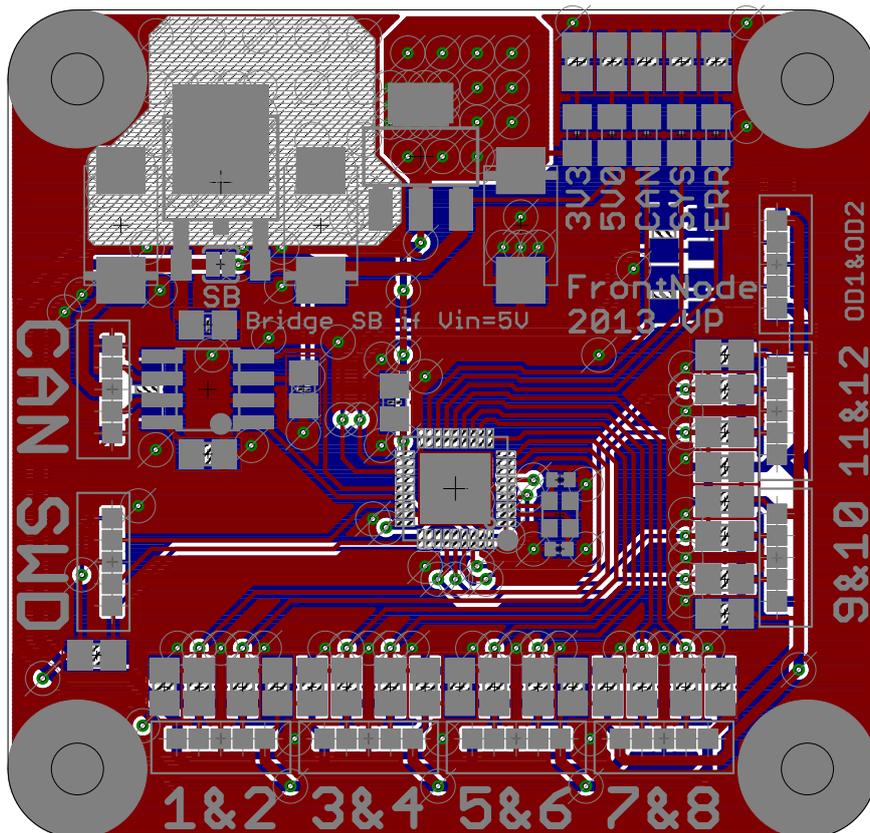


Figure 3.11: Front node layout

3.1.4 SoftStart

The soft start module limits the inrush current from the batteries by first connecting a resistor in series with the batteries to slowly charge the bulk capacitors on the motor drivers and after a fixed time connect the batteries directly to the motor drivers. The soft start module also connects or disconnects the battery based on the start/stop button and dead-man's handle.

The schematics of the soft start module is shown in Figure 3.12. The schematics mostly consists of a pair of relays to connect the batteries via a resistor or directly to the motor drivers, a RC filter to set the delay before the charge resistor is shorted, a charge

resistor and a discharge resistor. One important feature of the soft start module is to disconnect the power quickly when the user either disconnects the dead-man's handle or presses the power off switch, this is done by placing a diode, seen in the section labeled "SlowStart" in Figure 3.12, across the R in the RC net to lower its discharge time and thereby disconnecting the batteries quickly.

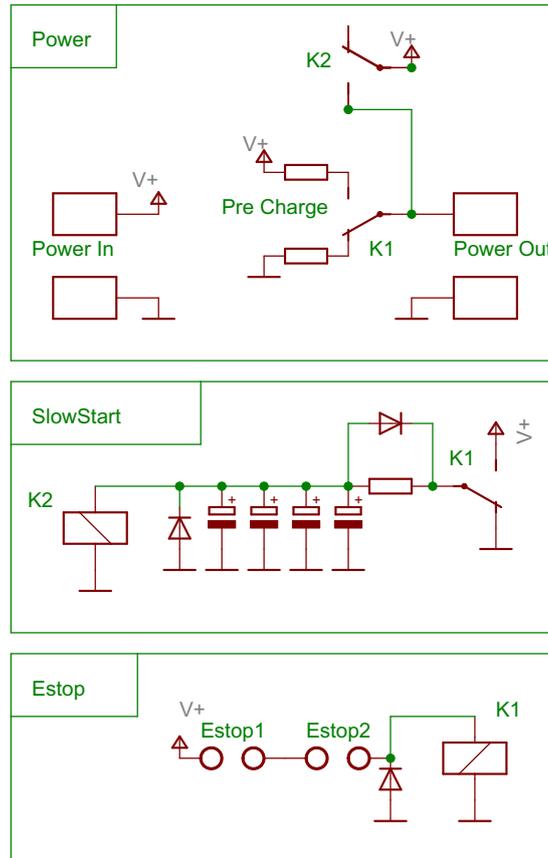


Figure 3.12: Soft start schematics

The layout of the soft start module, seen in Figure 3.13, is simple and only contains a few important design decisions. The most important design aspect of the PCB layout is to minimize the power loss, this is done by using a large copper thickness of $105\mu m$, wide and short traces and adding solder on the traces. The mounting holes on the PCB are positioned to allow the PCB to be mounted in a specific box.

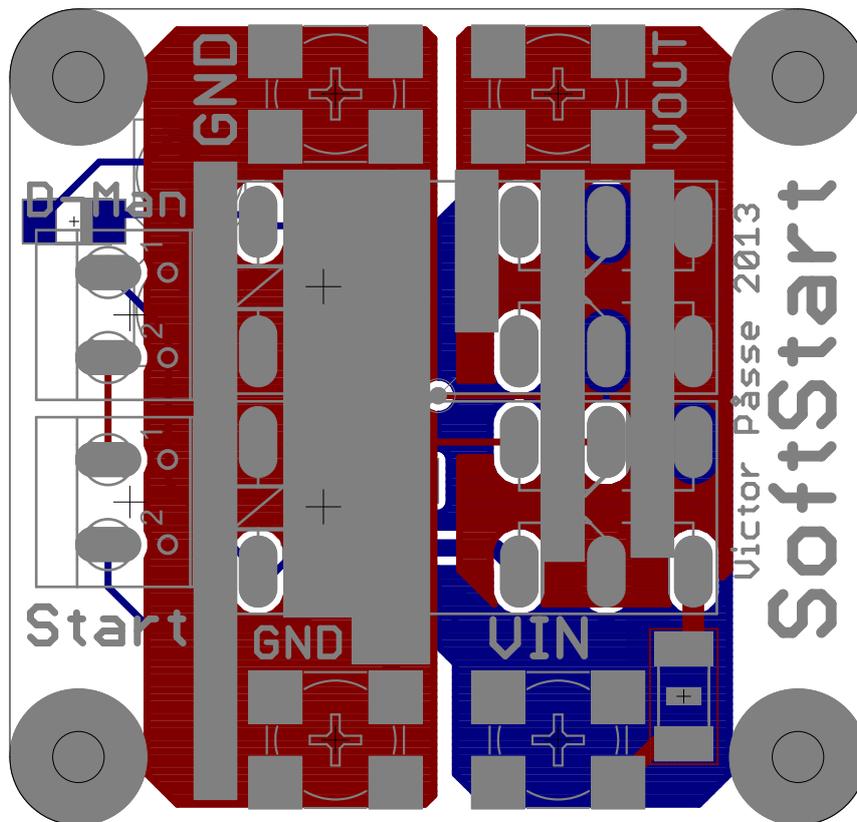


Figure 3.13: Soft start layout

3.1.5 Sensors

Two sensors types have been developed for use on the vehicle; a Hall effect based angular sensor and an optical rotational encoder with quadrature output. The optical sensor was not used in the final setup of the vehicle as the front wheel was changed and the optical sensor removed. A Hall based magnetic sensor was used instead as an optical sensor could be more prone to failure due to dirt or bright sun light. These sensors are connected to the pedals, steering rack and front wheel. The schematics and layouts of the sensors are elementary and are not included in this report.

3.1.6 SoundRacer

The SoundRacer unit has been connected to the front node and is controlled via a PWM signal that controls the virtual engine RPM. The duty cycle of the PWM signal is proportional to the throttle pedal engagement, this could be implemented in a nicer way to properly simulate a combustion engine by for example implementing a complete virtual drivetrain.

3.2 Software

All of the software mentioned in 2.3 has been either fully or partially implemented and tested. The testing has been done empirically and not by formal verification. This decision has been made partly because the main aims of this project are not dependent of any formal verification and because the time needed to formally verify all code could be used for tasks that have higher priority. The software running on the front node, motor driver or the operating system running on the control computer are not covered in this section, only the vehicle enhancement software subsystems. The resulting code and test results are covered in the subsections below for each software subsystem.

3.2.1 Stability Program

The purpose of the stability program is to measure the true yawrate and calculate target yawrate based on steering angle and vehicle speed. The true yawrate is measured in the IMU on the control computer and the target yawrate can be calculated using equation 3.1 and equation 3.2, measuring the front wheel speed and the steering angle. The yawrate error is feed in to a PID regulator whose output is added to the differential mode torque signal to the rear wheels. Equation 3.1 and equation 3.2 was developed using simple trigonometry.

$$TurnRadius = \frac{WheelSeparation}{\tan(SteeringAngle)} \quad (3.1)$$

in which

TurnRadius Turn Radius

WheelSeparation Wheel Separation between Front and Rear Wheels

SteeringAngle Steering Angle

$$YawRate = \frac{FrontWheelSpeed}{TurnRadius} \quad (3.2)$$

in which

YawRate Vehicle YawRate

FrontWheelSpeed Front-Wheel Speed

TurnRadius Turn Radius

The model for estimating the yawrate has been tested by driving the vehicle at different speeds and steering angles and logging vehicle state data such as speed, steering angle and yawrate to calculate the estimated yawrate based on the logged speed and steering angle and comparing the true yawrate with the estimated yaw rate. The data of this test is shown in Figure 3.14 and the results are that the model is sufficiently accurate. Any discrepancies between the measured yawrate and the estimated yawrate is attributed to non linearity in the steering and measurement errors.

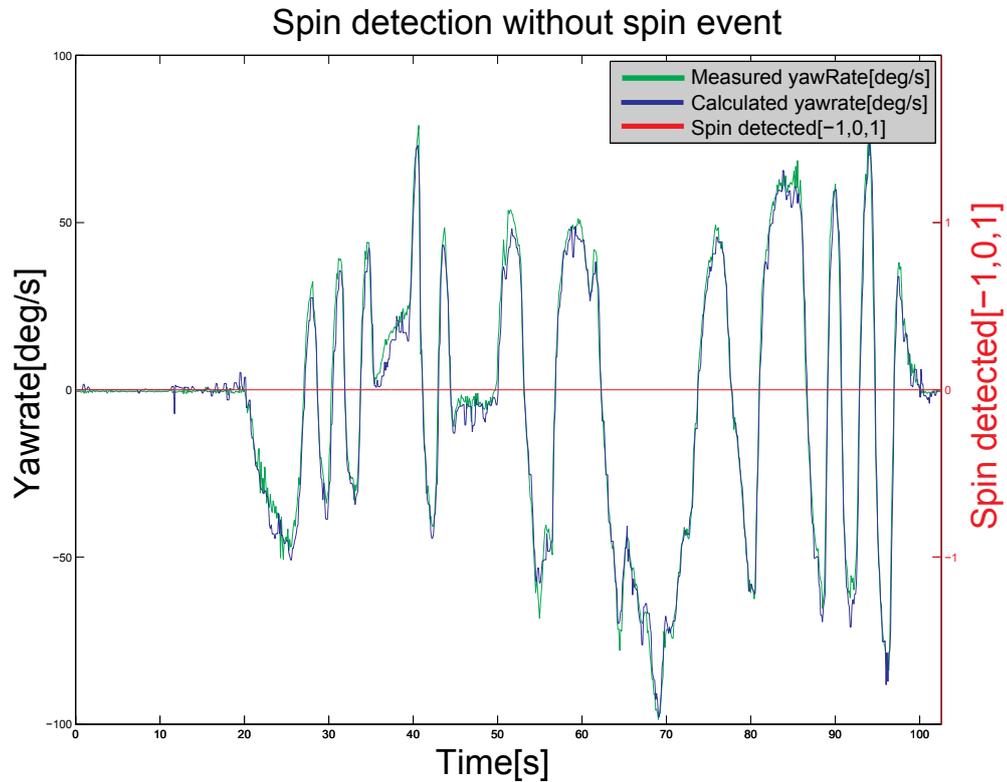


Figure 3.14: Stability program test without spin event

The measured yawrate and the estimated yawrate are compared and the difference is filtered and then compared to a threshold of $50^\circ/s$, if the difference is higher than the threshold a clockwise spin is detected and if the difference is lower than the negative threshold a counter clockwise spin is detected. The output of the spin detection algorithm during a spin event is shown in Figure 3.15 which shows the same data as in Figure 3.14 but with a spin event present. A large discrepancy between the measured and estimated yawrate can be seen from 43 seconds in to the test until 44 seconds in to the test. This shows that the spin detection algorithm works and can detect spin events.

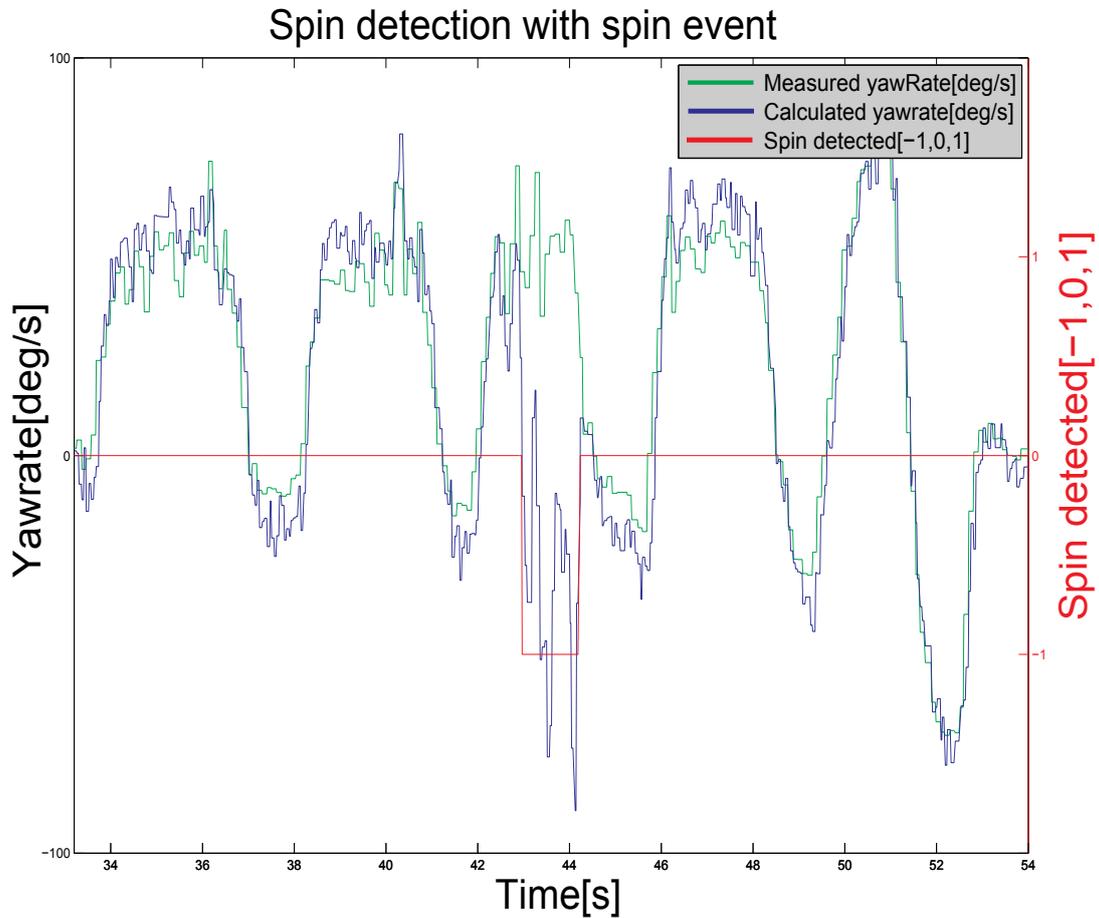


Figure 3.15: Stability program test with spin event

The PID regulator was never implemented on the vehicle as there was no available location suitable to test the stability program nor any time to do so.

3.2.2 Torque Vectoring

Torque vectoring has been implemented by adding a differential mode signal to the torque value to the motor drivers based on the steering angle. One problem with the vehicle is the missing suspension, this makes the vehicle lift the inner rear wheel if turned sharply. By implementing torque vectoring the same total amount of torque is applied independently of whether a wheel is off the ground or not by increasing the torque to the wheel on the ground and reducing the torque to the with less weight on it. The benefits are several such as lowered power consumption as less power is send to any wheel in the air, the wheel not contacting the ground does not build up speed which would be lost when ground contact occurs and thereby increasing the wear on the tire. Another benefit of the torque vectoring is that the torque developed by each motor is more proportional

to the load on that wheel. This lowers the risk of spinning as the torque is more prone to be lower than the static/dynamic friction limit.

3.2.3 Regenerative Breaking

The regenerative breaking system has been implemented as discussed in section 2.3.6. The regenerative breaking system has been tested by simply breaking the vehicle with the regenerative breaking enabled. The motor driver has been tested with breaking currents up to 25A at voltages up to 70V which is more than the battery is designed to be charged with. The regenerative breaking system is therefore deemed as working and not limiting in any way.

3.2.4 Anti Lock Breaking

The anti lock breaking system has been implemented and tested. The system works by measuring the front wheel speed and the rear wheel speeds and comparing these. If any rear wheel is rotating slower than the front wheel by a fixed amount while breaking, that rear wheel is considered locked and the breaking force is reduced to regain traction. This implementation requires the front wheel speed to be equal to the vehicles true speed which may not be true as there is a mechanical brake present on the front wheel designed to be used in emergency situations. If the front brake is applied, the anti lock breaking system will not function properly and may increase the breaking distance. Because of this the anti lock breaking system was removed and not implemented in the final version but it is, however, functional.

3.2.5 Anti Spin

The anti spin system is implemented by comparing the true vehicle speed to each rear wheel speed and if any rear wheel is rotating quicker than the true vehicle speed plus a threshold then the torque to that wheel is lowered which makes that rear wheel slow down and stop spinning. There was no available location suitable to test the anti spin system as spinning only occurs if the friction to the ground is very low as most of the weight of the vehicle is on the rear wheels.

4

Discussion

Since all subsystems, both hardware and software, work as designed and the vehicle is fully operational and performs well, the main aims of this project are fulfilled. The results of this project shows that it is feasible to build a electric propulsion system from ground and up within a highly limited period of time and still maintain performance and safety of said system.

Most of the subsystems can be reused for any electric vehicle that has one or more electric motors. The design of the motor driver node has been slightly modified and reused on a electric bike. The regenerative anti lock braking software subsystem was moved to the motor driver instead of running on the control computer node in the bike as no control computer node was present. This shows that the hardware and software are modular and the configuration of these can easily be modified to serve many purposes.

Several parts of the project can be worked on more to increase the value and usefulness of the vehicle, for example the pieces of software that was never fully implemented or tested such as stability program and anti spin or adding hardware nodes such as a dashboard. A dashboard will be added by ÅF in the form of a tablet connected to the vehicle via a bluetooth or USB link.

There are parts of the subsystems that has flaws, some examples of these are listed below:

- The main bulk electrolytic capacitors has a too high ESR which limits their ripple current handling capability, this could lead to excessive heat and EMI noise.
- Some of the inputs on the nodes are not protected very well for noise or ESD.
- The relays in the soft start node are not rated for the surge current that they could experience due to the design of the soft start

Most systems have flaws and it is important to find out all of them so that they can be fixed or avoided. The mentioned flaws has not been fixed as they have not resulted in

any problems as of yet. By testing the systems to their limits, flaws that can result in problems or errors can be found. It is therefore important to test all systems to their limits. The vehicle has been driven in several types of terrain, such as wet, uneven and loose terrain, as well as by several drivers to find flaws on the vehicle. The only failures that has occurred during testing are fuses which have blown and have been replaced by fuses with higher current rating. This is however a good form of failure to have as it shows that the motor drivers can handle more current than the fuses. After the final fuses of 30A per battery and 40A per motor driver was mounted no fuses have blown.

It is important to note that the hardware and software developed in this project is not, in any way, tested for compliance with the rules and regulations of automotive systems and would to all probability fail most of the harsh tests designed to keep the vehicles on the roads safe and reliable. The results of this project are for purely academical purposes and should be treated accordingly.

The methodology used in this report is not the same as those used by large vehicle manufacturers who usually use a more model based approach by modeling the vehicle carefully and design their software and hardware using higher abstraction layers. Examples of this include using simulink or similar programs to develop and maintain the control systems used in the vehicles for everything from climate control to engine control.

5

Conclusion

All of the hardware works as designed and performs very well according to all tests. Most of the software has been implemented and tested successfully while some software was never completed as it was too complex and the need of them was not as great as predicted in the beginning of the project.

To conclude this project, it is clear that developing the whole chain of hardware and software needed to control an electric vehicle can be quickly and easily done and still function in a satisfying manner.

Bibliography

- [1] N/A, Sinclair C5, http://en.wikipedia.org/wiki/Sinclair_C5, [Online; accessed 28-May-2014] (2014).
- [2] Transportstyrelsens, Transportstyrelsens författningssamling, https://www.transportstyrelsen.se/tsfs/TSFS%202010_144.pdf, [Online; accessed 03-May-2014] (2010).
- [3] Farnell, CadSoft Eagle PCB Design Software, <http://www.cadsoftusa.com/>, [Online; accessed 03-May-2014] (2011).
- [4] J.-P. Charras, KiCad EDA Software Suit, <http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>, [Online; accessed 03-May-2014] (2014).
- [5] Altium, EDA Software, <http://www.altium.com/>, [Online; accessed 03-May-2014] (2014).
- [6] Farnell, Free ARM Cortex MCU IDE, http://www.coocox.org/CooCox_CoIDE.htm, [Online; accessed 03-May-2014] (2011).
- [7] Microsoft, Visual C#, <http://msdn.microsoft.com/en-us/library/kx37x362.aspx>, [Online; accessed 03-May-2014] (2014).
- [8] Microsoft, Visual Studio Express, <http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>, [Online; accessed 03-May-2014] (2014).
- [9] Farnell, Free and open ARM Cortex MCU RTOS, <http://www.coocox.org/CoOS.htm>, [Online; accessed 03-May-2014] (2011).
- [10] S. Microelectronics, STM32F4 Series, <http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577>, [Online; accessed 28-May-2014] (2014).