CEDES

Cost Efficient Dependable Electronic Systems

Microprocessor control of a permanent-magnet motor with fault-detection, suited for an electromechanical brake system

Author	Magnus Källvik and Jonas Eriksson
Document Id	014
Date	15 June 2006
Availability	Public
Status	Final

CHALMERS

Microprocessor control of a permanent-magnet motor with fault-detection, suited for an electromechanical brake system

MAGNUS KÄLLVIK

JONAS ERIKSSON

Master's Thesis

Electrical Engineering Program

CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Division of Computer Engineering
Göteborg 2006

Abstract

In this thesis two different processor boards from the GAST project, with either a HCS12 or both a HCS12 and a MPC565, has been tested to see how well they would act as nodes in a distributed brake-by-wire system with electromechanical brakes. A brushless DC motor has been used as an actuator during the tests and studies in the area of suitable actuators for an electromechanical brake has been made. The conclusions from the tests are that the MPC565 would suite very well as a node. Even though its a general purpose processor it handles control tasks very well and has capability of running both applications and controlling an actuator, especially with help of its Time Processor Unit(TPU).

A brushless DC motor probably has to much ripple in its produced torque and the most suited actuator in an electromechanical brake is probably a Fault-tolerant Synchronous machine controlled with sensor less control.

Fault-detection has is important in brake-by-wire systems and the tested Model Based fault-detection did not manage to fulfill the requirements from such systems. This is probably due to the simplicity of the used algorithm used for estimation of system parameters and huge changes in the system during actuation.

Contents

Acknowledgments

We would like to thank our Examiner and Supervisor Roger Johansson for his help during our work and Håkan Sivencrona at Mecel for giving us the opportunity to do this thesis.

Magnus Källvik and Jonas Eriksson
Gothenburg, Sweden
May, 2006

1 Introduction

The complexity of technology used in automotives is constantly increasing with the goal of making cost savings and interesting products. More and more mechanical system has been replaced by electronic systems but the most safety-critical systems have been left out because of its complexity. But even more complicated system like the brake-system has been coming closer to a replacement of the old mechanical system to an electric system, a so called brake-by-wire-system. By-wire systems are commonly used in the aero industry but the requirements on low costs in the automotive industry has kept back the use of such systems. But dramatic development in electronics the last years of both price and products has made it more interesting with by-wire systems for the automotive industry. But its still a great challenge to implement a brake-by-wire system in series produced automotives with its high demands on low cost and still fulfil safety requirements.

As a part of this development the CEDES (Cost Efficient Dependable Systems) project has started which aims to research technologies that will provide dependable and reliable electronics systems that are affordable and possible to use in series production in the automotive industry.

GAST, another project led by Chalmers, has developed hardware in the form of processor boards and communication controller boards which can be used to simulate real ECUs (Electrical control units) and networks of ECUs in vehicles. These boards are built using standard components and open specifications, thereby providing a development and research platform.

This thesis is a part of CEDES and aims to look in to how a brake-by-wire system with electromechnical brakes can be used in series produced automotives. Focus is on suitable electric actuators and how these can be controlled with available hardware from the GAST project. As the available hardware is built-up by general purpose components this is also a test of how well such components can handle the task of running applications and controlling a suitable actuator. The actuators requirements on its surroundings to fulfil its duty will also be discussed briefly.

Used for this research is a Brushless DC motor and two different processor boards from the GAST project with either a HCS12 from Motorola or both a HCS12 and MPC565 processor on it.

2 Background

2.1 Dependable systems

Safety-critical system like e.g. a brake-by-wire systems have very high requirements on reliability. Normally the same requirements of those in the aero industry are wanted in such systems due to the great number of cars that is used around world. The aero industry requirements on failures can according to [5] be summarized

- No single fault will cause hazardous failure
- Any fault which will cause an in-flight shut down must have a failure rate of less than 10^{-7} failures per hour.
- Undetectable single faults which could, in combination with a subsequent fault, cause an in-flight shut down must have a failure rate of less than 10^{-8} failures per hour.

To completely fulfil these requirements no single fault in the entire system should cause hazardous failures. An easy but expensive way of solving this is by using a simple redundancy by duplicating all components that can fail. This costly solution though is probably not the way to reach the safety requirements in mass produced automotives where more cost effective solutions is seeked. Therefore solutions like Distributed Systems can be used making redundancy possible without duplicating the number of hardware units.

2.1.1 Distributed Systems

One way to solve the demands of future dependable systems without the extra wiring and processors is to use distributed systems. A way to use a distributed system for a brake-by-wire system can be seen in figure 2.1.

In figure 2.1 the rectangular boxes are nodes and the round and diamond shaped boxes connected to them are sensors and actuators. Every node can have several actuators and sensors connected to it. A node is supposed to be able to control several different tasks like steering and braking. Every node should also be able to handle applications like ABS. protection.

The black wires that connect the front nodes with each other are cross connections used in something called functional distribution. Functional distribution is a way to create redundancy by connecting sensors and actuators to at least to nodes. Thus if one node fails in some way, another node can perform

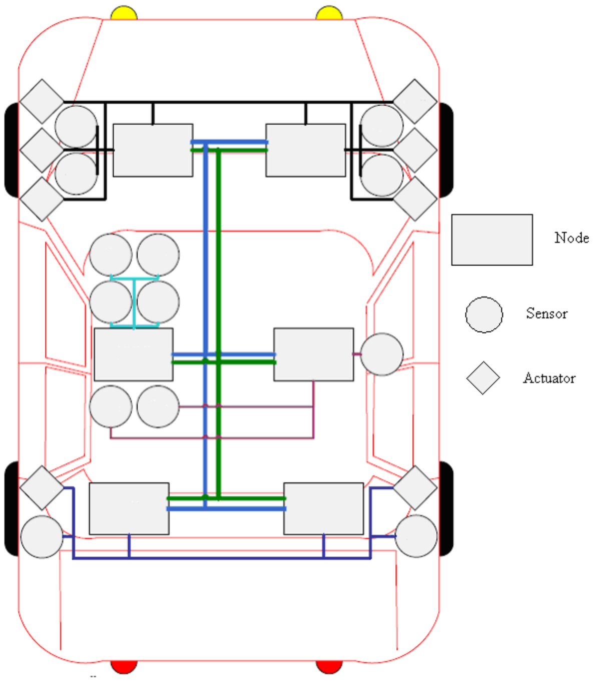

Figure 2.1: An example of a distributed system in a car

the faulted nodes tasks. This gives a fault tolerant system were a single fault does not result in heavily decreased operation or no operation at all.

The communications between different nodes can be handled with modern time triggered communication protocols like TTCAN, Flexray or TTP/C. All of these protocols are developed to work in complicated environments like in a car with high electrical and mechanical disturbances. The protocols are time triggered to guarantee that information from one node arrives to another node within a certain time. which is very important in brake-by-wire or steer-by-wire systems.

2.2 Brake-By-Wire

X-By-Wire is a commonly used way to name modern/future systems in automotives that don't have a mechanical coupling between actuator and the driver. E.g. steering systems that ever since the automotives were invented has been purely mechanical were the driver when rotating the steering wheel directly has been affecting the angle of the front wheels through mechanical couplings, although sometimes with help of different servo systems. In X-by-wire systems this mechanical coupling between the driver and the things the driver wants to effect is replaced by an electrical system where an electrical actuator builds up the mechanical force instead of the driver.

2.2.1 Electromechanical brakes

In comparison to conventional brake systems in cars where the braking force is built up by a hydraulic pressure created by the driver when pressing the brake pedal Brake-By-Wire build up its brake force with electric actuators.

There are different types of Brake-By-Wire systems it can be either purely electrical (Electro-Mechanical brake, EMB) or partly hydraulically and partly electrical (Electro-hydraulic brake, EHB). In an EHB-system a desired brake force is read from the brake pedal by a control unit which then commands and electrical actuator to build up oil pressure to activate the brake callipers. It either activates all wheels at the same time or each wheel independent of each other. In Electro-Mechanical Brake the braking force is generated directly at each wheel by electric motors that actuate the brake callipers. An overview of a brake-by-wire system with electromechanical brakes can be seen in figure 2.2. Here a wheel brake module consisting of a mechanical system with brake disc and callipers etc., an electric actuator and a control module. The control module achieves a braking torque command from the ECU and controls the actuator actuate the callipers.

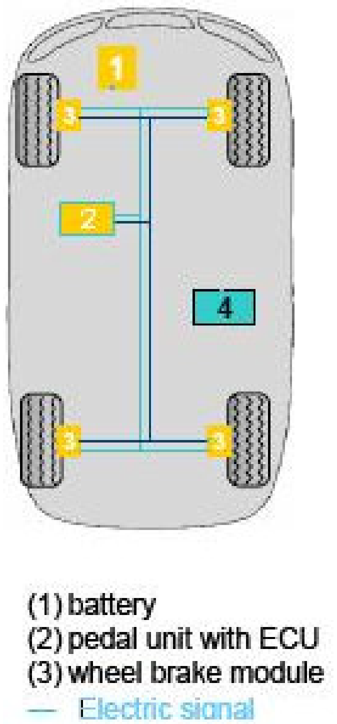

Figure 2.2: Overview of a electromechnical brake system. Courtesy Freescale Semiconductor

There are several advantages with an EMB-system in comparison to the old fashion hydraulic system. To name a few advantages for EMB [12] :

- Shorter stopping distances and optimized stability

- More comfort and safety due to adjustable pedals
- No pedal vibration in ABS mode
- Virtually silent
- Environmentally friendly with no brake fluid
- Improved crash worthiness
- Saves space and uses fewer parts
- Simple assembly
- Capable of realizing all the required braking and stability functions, such as ABS, EBD, TCS, ESP, BA, ACC, etc.
- Can easily be networked with future traffic management systems
- Additional functions, such as an electric parking brake, can easily be integrated.
- Weight reduction, especially for heavy duty vehicles that uses air-brakes

Mechanical system for an Electromechanical brake

The mechanical system of an electromechanical brake will not be handled in this report. In order to create realistic simulations and to suggest improvements an electro mechanical brake similar to the solution in figure 2.3 was assumed.

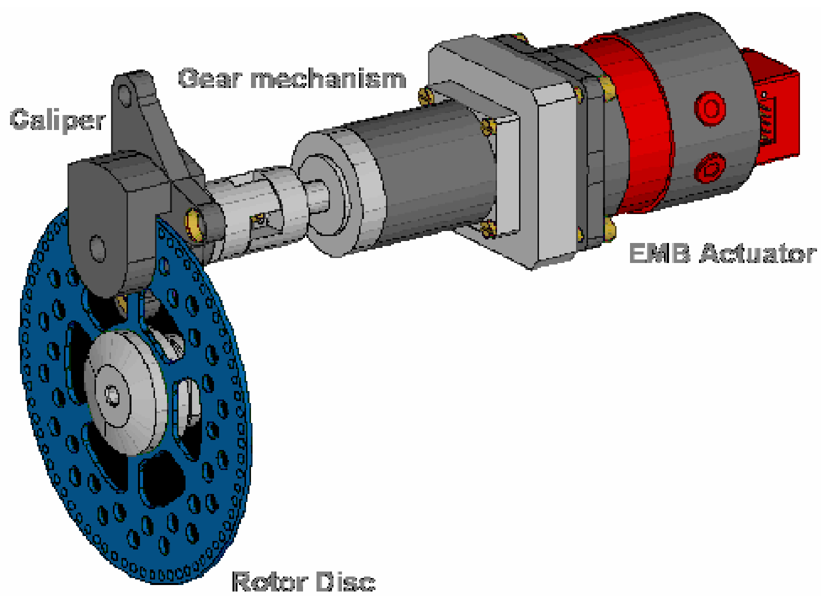

Figure 2.3: Assumed Mechanichal design of the electromechanical brake. Courtesy Freescale Semiconductor

The EMB-actuator is the only component creating brake power in an electromechanical brake. To transform the weak force from the motor into a force large enough to create sufficient break force a gear is used. The gear contains of two parts, a traditional gear and a ball screw to convert the radial movement into a movement pushing the two callipers together.

The construction is also assumed to be made in such way that when e.g. the actuator is broken the system becomes passive and does not affect the system by applying an undesired brake force.

Actuators for Electromechanical brakes

Electro-mechanical brakes needs an electric motor actuator to convert electrical energy to mechanical energy. In automotive brakes, cost, weight and reliability is very important when choosing actuator. Permanent magnetized (PM) machines are therefore a good choice due to its very high efficiency and power density [12]. There are two principal classes of PM-machines, Permanent Magnet Synchronous Motors (PMSM) and brushless DC (BLDC)-motors. The primary differences between these to principles is that the PMSM are sinusoidal excited and the BLDC-motors is trapezoidal excited. These actuators is discussed further in chapter 3

An alternative to the use of the classic brushless dc motors or synchronous motors, fault-tolerant machines can be used increasing the reliability of the actuator considerably. Hence this machine is able to continue operation after a single fault in e.g. a phase winding or in a power device feeding the motor. This type of fault forces the classic motors to a shut down resulting in a heavily reduced brake capacity. Using this type of machine the safety requirement that no single fault should couse an in-flight shut down used in the aero industry mentioned earlier in the chapter can also be fulfilled if referring an "in-flight shut down"' to a shut down of an actuator in a brake system. Principles of fault-tolerant machines is described in 3.3.

Control Module

The Electric motor actuator is controlled by a Control module consisting of a microcontroller that with help power electronics, to feed the motor with power, controls the actuator. The microcontroller also receives the torque command from the driver and transmits diagnostic information about the brake system.

In contrast to hydraulic brakes, where the clearance of brake pads adjusts automatically when the hydraulic pressure is released; electro-mechanical brakes must actively adjust the clearance by actuator rotor movement. It is therefore very important to keep track of the rotor position at all time.

In this thesis it is assumed that there is a measured brake force for control purpose according to figure 2.4. This to get a more exact brake force hence it is important to have same brake force at wheels on the same wheels-axis in order to remain stability when a brake force is applied.

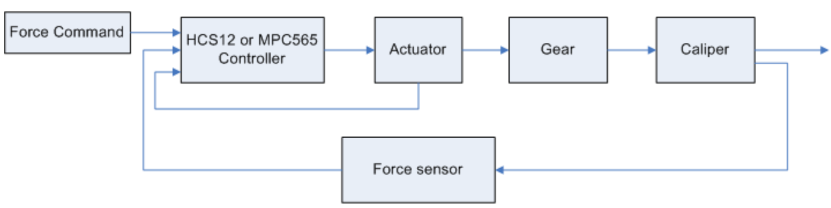

Figure 2.4: EMB control system structure

2.3 GAST

The GAST-project(General Application Development Boards for Safety Critical Time-Triggered Systems) is a project within the CEDES-project(Cost Efficient Dependable Electronic Systems). The background for the GAST-project is to produce cost-effective hardware to test and develope communication protocols used in by-wire applications. The boards also have multiple in- and outputs to control actuators and read sensed values. GASTs objectives are to deliver two development boards(G1 and G2) used for testing within the CEDES-project and the financiers. Other objectives of the project are to produce basic software drivers and project documentation in terms of technical reports for both G1 and G2. The project will continue until May 2006 [4].

2.3.1 G1

The G1 board was the first board developed within GAST-project. The heart of the board is a 16-bit Motorola HCS12 processor working at 8 MHz. This processor has been used in the automotive industry for many years and is a fairly simple processor. The board have multiple in- and outputs for communication, PWM-signals and A/D converters. The board has a large backplane connector used for different communication protocols. This connector hasn't been used in this report and will therefore not be further described. two connectors with more significance in this report can be found. The Automotive IO connector(upper) contains power supply, serial communication in- and outputs, HDO/HDI/AD(selected with jumpers), LDO/LDI/AD(selected with jumpers) and VRH and VRL to the AD-converters. The G1 specific IO connector(lower) contains V_{cc} and a number of digital inputs and AD-converters. Both connectors will be frequently used in this report.

2.3.2 G2

The other board within the GAST-project is G2 that is a far more complicated card then G1. The G2 board has two processors. The main processor is a 32-bit MPC565 Power PC processor from Freescale. The other processor is a 16-bit HCS12 from Freescale, the same kind that can be found on the G1 board. On this board HCS12 can be used a monitoring processor.

This board has the same backplane connector used on the G1 board. On the other side of the board a similar Automotive IO can be found but also a bigger connector from the same family. Both these connectors are packed with IO connections. The same high side and low side power switches that can be found on G1 can also be found on G2. The difference between the boards when it comes to IO-connections is that there are more connections on G2 and they are connected in a different way, especially the TPU-connections on the bigger connector. More about TPU and the big connector can be read in chapter 6.2.4.

2.4 Thesis Objectives

This thesis was done as a part of the CEDES project,[2]. The main objectives for the thesis were:

- Put together demands for brake-by-wire
- Test available hardware
- Study how fault in an electromechanical brake can be detected
- Suggest improvements on the basis of the tests on the available hardware

The purpose for this was to come a step closer to the implementation of a brake-by-wire system that the CEDES project are working on to setup in the future. This system is to be build with, among other, hardware from the GAST project [4] for the purpose to test how such a system can be implemented fulfilling safety requirements in a cost efficient way.

3 Electric motor actuator

In automotives cost, weight and reliability is very important. Permanent-Magnet Synchronous Machines(PMSM) is therefore a good choice due to its high power density and reliability [13]. This machines is classified depending on the wave shape of their induced emf (electro mechanic force) i.e., sinusoidal and trapezoidal. The sinusoidal type is called Synchronous machines and the trapezoidal type Brushless DC-machines. This chapter will describe the principles of this two types mathematical models will be developed and strategies for control will be discussed to show the differences between these to types of machines. Hence, fault-tolerant capability is required by an electromechanical brake system different strategies to achieve this using a special type of PMSM called Fault-tolerant machine will be discussed.

3.1 Brushless DC motors(BLDCm)

A brushless DC motor is basically a normal DC motor turned inside out. The stator in a normal DC motor creates a static electric field that interacts with a rotating magnetic field in the rotor. To transfer current to the rotor windings and to create the rotating field a normal DC motor uses brushes. The brushes are fastened at the stator and pressed against the rotor. The rotor has several isolated contact points that changes the current flow and the magnetic field in the rotor.

In a brushless DC motor the magnetic field in the stator is rotating due to an electronic commutation and the rotor has a permanent magnetic field created by permanent magnets why no physical connection to the rotor through brushes needed. To be able to create a rotating and torque producing magnet field the rotor position has to be known which is normally solved by using Hall-sensors or a resolver.

The main advantage of the brushless DC motor is its reliability hence there is no brushes to wear out. The brushes in a normal DC motor also create dust that can damage both the motor and the surroundings of the motor. The absence of vibrations and noise created by the brushes is also an advantage for the brushless motor.

The main disadvantage of the brushless DC motor is the need of a controller. Controlling the motor means additional costs and more complicated electronics.

3.1.1 The Principle

A brushless DC motor has, as has been said earlier, permanent magnets on the rotor. This magnets induces a trapezoidal emf in the stator windings when rotating, see fig. 3.1 where the waveforms of a 3-phase 8-pole machine is shown. To get a uniform power output one wishes to excite the windings that at the moment is at the top or bottom of its trapezoidal waveform. To achieve this currents in the stator-phases has to be switched every 15 degrees to achieve the shape of the currents in fig. 3.1. Each phase then contributes with the power $i_{xs}e_{xs}$. Which two phases that contributes with power depends on the angle of the rotor.

Currents cannot rise and fall in the phase windings in zero time, hence in actual operation there are power pulsations during turn off and turn on of phase currents. This pulsation is a disadvantages compared to synchronous motors where such pulsations is absent due to its sinusoidal emf.

The stator consists of different

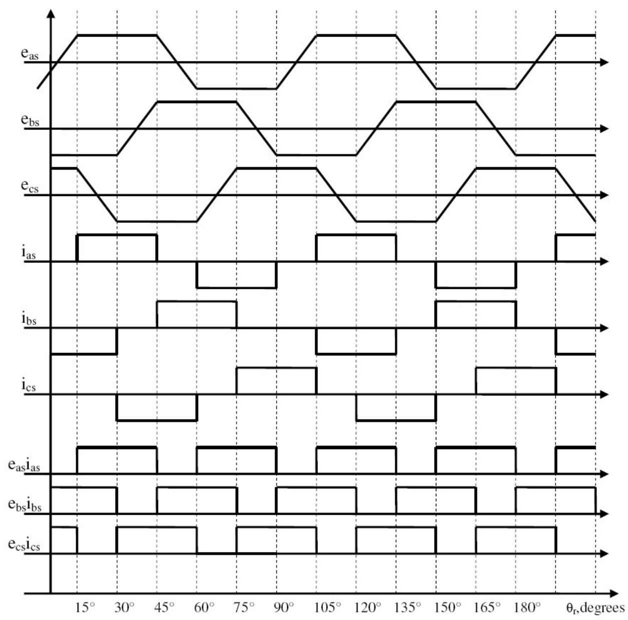

Figure 3.1: Brushless DC motor waveforms for three-phase, eight-pole machine

Hall sensors and switching scheme

To be able achieve the wave forms in 3.1 the angular position of the rotor has to be known. This is normally done with hall sensors sensing the magnet field created by the permanent magnets in the rotor or a resolver on the motor shaft. As can be seen in 3.1 only six discrete positions have to be known in each electric cycle. How many mechanical degrees an electric cycle corresponds to depends on the number of poles. That only six discrete positions has to be known results in major cost saving compared to a synchronous machine which requires continuous and instantaneous rotor position. Only three Hall sensors, displaced by 120 electrical degrees, are needed for a three-phase machine. Hall sensors however are very sensitive and are often together with the power electronics considered to be the most sensitive part [3].

For a three-phase, eight-pole motor the electric cycle corresponds to 90 degrees and a switching scheme can look like that in fig. 3.2 which is the used switching scheme.

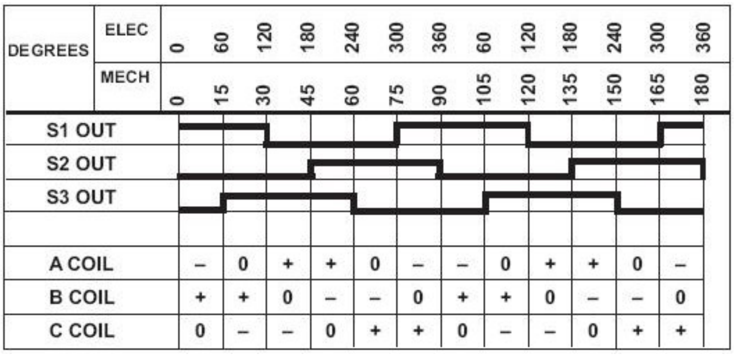

Figure 3.2: Switching Scheme

3.1.2 Modelling of BLDCm

In this section a mathematical model for a Litton,BN23-28CC-02LH, brushless DC motor is developed. The derived model is for a three phase motor but the derivation process is valid for any number of phases. The derivation of this model i based under the assumption that the induced currents in the rotor due to stator harmonic fields are neglected and that iron and stray losses also is neglected, [11].

The circuit equations of the stator windings can be written

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_a & 0 & 0 \\ 0 & L_b & 0 \\ 0 & 0 & L_c \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$

Where $R_a = R_b = R_c = R_s$ which is the stator resistance per phase and $L_a = L_b = L_c$ is the inductance per phase. e_a, e_b and e_c are the induced emf all assumed to be trapezoidal, as shown in fig. 3.1, with a peak value $E_p = k_E \omega_r$. The electromagnetic torque is given by

$$T_e = \left[\ e_a i_a + e_b i_b + e_c i_c\ \right] \frac{1}{\omega_r} (Nm)$$

For simulation purpose the instantaneous induced emfs can be written

$$e_a = f_a(\theta_e) k_E \omega_r \tag{3.3}$$

$$e_b = f_b(\theta_e) k_E \omega_r \tag{3.4}$$

$$e_c = f_c(\theta_e) k_E \omega_r \tag{3.5}$$

Where $f_a(\theta_e), f_b(\theta_e)$ and $f_c(\theta_e)$ have the same trapezoidal shape as e_a, e_b and e_c with a maximum amplitude of ± 1. However, in reality the induced emfs do not have sharp corners as shown in fig. 3.1, but rounded edges and fringing which makes the functions smooth with no abrupt edges. See section 3.1.5 to see how emf from the used BLDCm looks like. The electromagnetic torque can then be written

$$T_e = \left[\; f_a(\theta_e) k_E i_a + f_b(\theta_e) k_E i_b + f_c(\theta_e) k_E i_c \; \right] (Nm)$$

The equation of motion for the motor with inertia J, and viscous friction $b\omega_r$ and the load torque T_{load} can be written

$$J\dot{\omega}_r = T_e - b\omega_r - T_{load} \tag{3.7}$$

The relation between the electrical speed and the mechanical speed is dependent on the number of poles and can be written

$$\dot{\theta}_e = \frac{P}{2}\omega_r \tag{3.8}$$

where P is the number of poles.

The system can then be written in state-space form

$$= \mathbf{Ax} + \mathbf{bu} \tag{3.9}$$

where

$$x = \left[\; i_a \quad i_b \quad i_c \quad \omega_r \quad \theta_e \; \right]^T$$

$$\mathbf{A} = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & -\frac{k_T}{L} f_a(\theta_e) & 0 \\ 0 & -\frac{R}{L} & 0 & -\frac{k_T}{L} f_b(\theta_e) & 0 \\ 0 & 0 & -\frac{R}{L} & -\frac{k_T}{L} f_c(\theta_e) & 0 \\ -\frac{k_T}{J} f_a(\theta_e) & -\frac{k_T}{J} f_b(\theta_e) & -\frac{k_T}{J} f_c(\theta_e) & \frac{-b}{J} & 0 \\ 0 & 0 & 0 & \frac{P}{2} & \end{bmatrix}$$

$$b = \begin{bmatrix} \frac{1}{L} & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 \\ 0 & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & \frac{-1}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$u = \left[\; v_a \quad v_b \quad v_c \quad T_{load} \; \right]^T$$

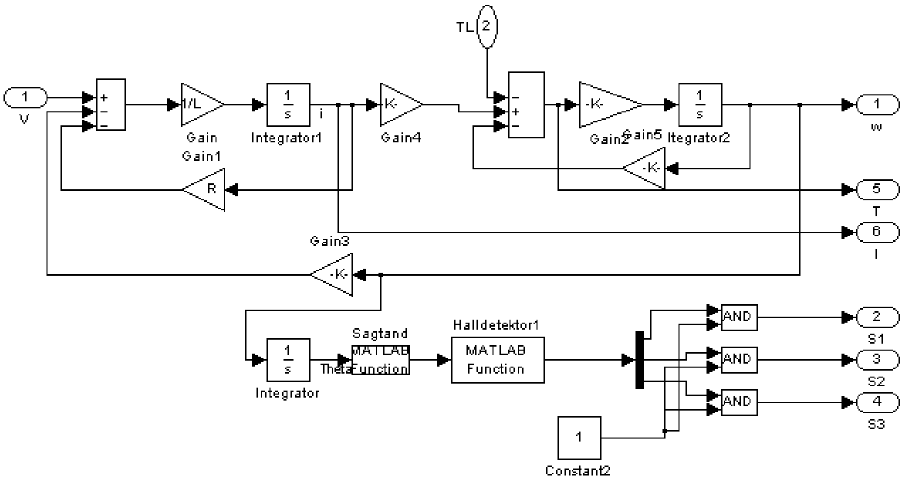

Figure 3.3: Motor model from simulink workspace

3.1.3 Simulation of a BLDCm

The simulation tool used to simulate the motor was matlab simulink. In simulink it is easy to create models with the help of mathematical blocks. The motor can be divided into three different blocks. The three different blocks are the electrical model, the mechanical model and a model for the three hall-sensors. Figure 3.3 is a workspace screenshot from the whole motor.

The mathematical model used to create the electrical model of the motor is the same used in chapter 3.1.2. In figure 3.3 is the part between the two summations. This part converts the incoming voltage into a torque.

The upper right part of the workspace contains the mechanical part with moment of inertia and the mechanical losses in the motor due to friction.

The matlab function box in the bottom of the workspace was made to recreate the hall-sensors in the motor. Depending of the rotors position the three signals S1, S2 and S3 are either zero or five volts according to the manual of the motor. The reason to recreate the signals from the motor instead of using the correct position was to see if the resolution of 15 degrees from the hall-sensors could be used in speed calculations.

3.1.4 Torque control of BLDCm

The produced electrical torque from a BLDCm is given by the equation

$$T_e = \left[\ f_a(\theta_e)k_E i_a + f_b(\theta_e)k_E i_b + f_c(\theta_e)k_E i_c\ \right] (Nm)$$

Only two phases conduct at the same time and with the windings in Y-connection the current in the second excited phases has the same magnitude but with the opposite signs. The rotor-position-dependent functions have the same sign as the currents in motoring mode. Thereby the produced torque can be written

$$T_e = 2k_T i_s (Nm) \tag{3.15}$$

This relationship makes it very easy to control the produced torque by controlling the stator current with the stator-current command

$$i_s^* = \frac{T_e^*}{2k_T}(Nm) \tag{3.16}$$

3.1.5 Measured emf from the used BLDCm

In chapter 3.1.1 the emf is shown to have the shape of a trapezoid. Since the trapezoid shape of the emf is very important to remain the linear relation between the current and the produced torque, the emf of the used motor was measured. To measure the emf, the voltage from the three phases when rotating the motor was measured. The result from the measurement from one of the windings can be seen in figure 3.4.

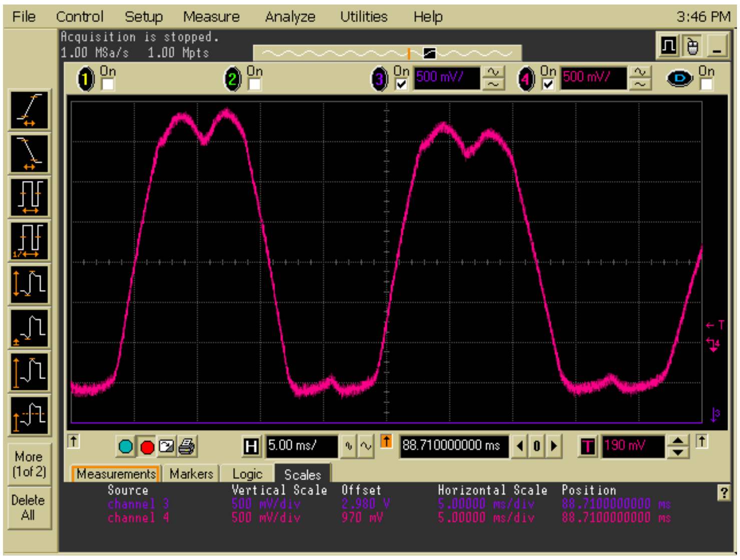

Figure 3.4: Measured EMF from a BLDCm

The figure shows that the produced emf does not have the shape of a trapezoid. The consequences will be a nonlinear relation between current and produced torque and result in a ripple in produced torque.

3.2 Permanent-Magnet(PM) Synchronous Machines

Since permanent magnet synchronous machines(PMSM) has a sinusoidal back-emf vector control is often used to control this type of machine. Vector control is more complicated then the control of e.g. a DC-motor hence it involves vector transformations like that in equation 3.17. This type of transformations is used to make it easier to control the motor and results in a control similar to that of a separately excited motor where you control to currents, a torque producing current and a magnetizing current.

$$\begin{bmatrix} i^r_{qs} \\ i^r_{ds} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\omega_r t & \cos(\omega_r t - \frac{2\phi}{3}) & \cos(\omega_r t + \frac{2\pi}{3}) \\ \sin\omega_r t & \sin(\omega_r t - \frac{2\phi}{3}) & \cos(\omega_r t + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}$$

The advantages with a sinusoidally excited motor is the absent of ripple in the produced torque.

3.3 Fault tolerant Permanent Magnet motors

A safety critical system like an electromechanical brake has high demands on safety. If same safety is required as in the aero industry the failure rate for the actuator has to be less then 10^{-7} failures per hour, see Chapter 2.1.This can not be achieved with a classic design of the PM-machine which has a electrical failure rate of $18 * 10^{-6}$ failures per hour [5]. A possible solution to achieve acceptable failure rate is to use a switched reluctance drive. But using a modified version of the conventional brushless machine proposed by [5], called modular permanent magnet brushless machine or fault tolerant permanent magnet brushless machine, gives a higher power density and a better output and with this type of machine a failure rate of 10^{-7} can be achieved.

In this chapter the above mentioned modular permanent magnet machine is discussed.

3.3.1 The principles of the fault tolerant machine

In the modular fault tolerant motor each phase is designed such that a fault in one phase does not directly affect healthy phases thus enabling the machine to continue with a faulted phase. This is provided by keeping phases electrically, physically and magnetically isolated from each other. The machine stator depicted in fig 3.5 where it can be seen that each phase is formed from a single coil and each coil is wound around an alternate single tooth to isolate the phases from each other. To remain isolation between phases outside the machine each phase is feed separately in comparison to the regular brushless machine where the phases are Y-connected and two phases are excited with the same current.

Phases are designed to have a relatively high reactance to limit currents in a short circuited phase. A design with 1.0 per unit reactance in each phase is seeked so the current driven in a short circuited phase at rated speed doesn't exceed rated current. Thus enabling the machine to continue operates without reducing the lifetime of other phases by raised operating temperature. Because current in a shorted phase is reactance limited it is almost exactly 90 degrees out phase with the back emf and is therefore producing very little torque when faulted. With this construction the fault tolerant machine can continue operating with one phase faulted. Thus two faults has to occur to stop the machine with hits design a failure rate of 10^{-7} per hour can reached. Thus it requires a failure rate per phase to be $\leq \sqrt{10^{-7}} = 316 * 10^{-6}$. This can be compared to a typical drive in industrial environment which has an electrical failure rate of $18 * 10^{-6}$ failures per hour [5].

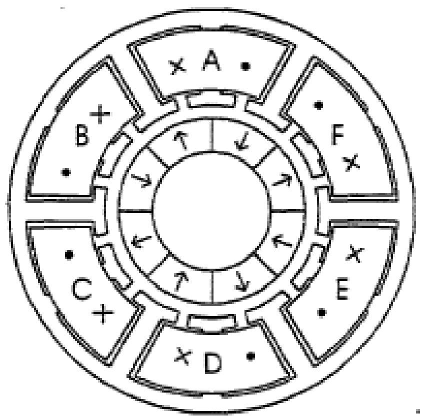

Figure 3.5: A machine stator for a fault tolerant machine designed to achieve isolation between its phases

To maintain performance after a fault has occurred on one phase the machine has to be over rated hence a loss of a phase demands the remaining phases to produce more torque to reach the demanded torque. The over rating factor, which tells how much a machines rated torque should be multiplied with to be able to maintain performance with a faulted phase, is 1.5 for a three-phase, 1.25 for a four-phase and 1.2 for five-phase machine.

Keeping phases isolated in the power electronics has the drawback that it each phase has to be feed separately which doubles the numbers of Mosfets in the power electronics. Doubling the number Mosfets doesn't mean duplicating the cost thus each Mosfet don't need to have as high rated power. In fact [1] calculates the rated Volt-Amps needed is approximately 15 percent higher for power electronics feeding a classic PM-motor then for power electronics feeding a modular motor. The differences in layout for power electronics feeding a regular respectively a modular motor can be seen in fig. 3.6. Where the upper figure shows a 3-phase inverter is used to feed a classic PM-motor and the lower a 3-phase H-bridge converter with each phase brought out of the motor and connected to a separate H-bridge.

To reduce the kneed of duplicating the number of Mosfets the neutral point can be drawn out form the motor thus making it possible to draw current in both directions with a faulted phase. This motor still has to have the same construction as the fault-tolerant machine mentioned above with isolated windings so a fault in one phase can't affect the healthy phases. This type of machine has to be overrated with a factor of $\sqrt{3}$[10]. In figure 3.7 this solution shown with help of an example where one phase i faulted(phase a) and isolated.

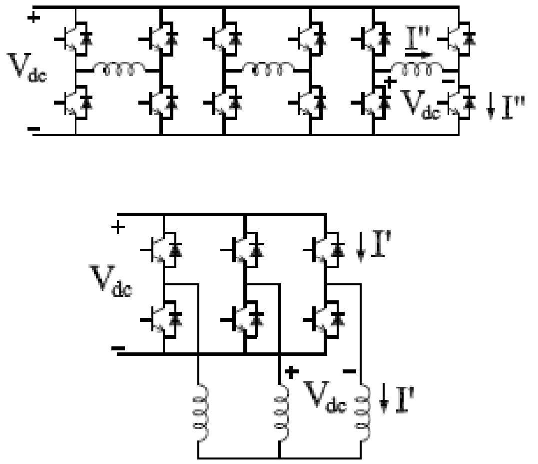

Figure 3.6: The upper figure shows an inverter used to feed classic PM-motor and the lower figure shows a H-bridge converter used to feed a fault-tolerant machine showing the need of duplicating the number of Mosfets

3.4 Sensor less Control

In the automotive industry as few parts as possible is often wanted to reduce costs and the number of parts that can break. A control system without a need of hall sensors or a resolver to keep track of the rotors absolute position is good example of this. Hence excluding hall sensor both reduces the number of parts and removes a very sensitive part of the system.

The most common way to keep track of the rotor position is to measure the induced emf and thereby, e.g. in the unexcited phase hence only two phases are excited at the same time. An example of this the zero-crossing technique where the emf is measured and commutation occurs when the induced emf changes sign. This is very easy to implement and an example of implementation is shown in figure 3.8 where an op-amps that changes sign when the measured emf changes sign is used to keep track of the rotor position.

All sensor less control that uses the back-emf to keep track of rotor position has the disadvantages that it can't be used at standstill or at low-speed hence that there is none or very little induced emf when operating in this regions. Which makes it difficult to use in an electromechanical brake system where low-speed and standstill performance is needed. But new methods to cover up these regions have been developed. An example of this is Signal Injection where a high frequency signal is feed to the motor and the response from this signal is measured and used to calculate the position of the rotor. In [14] this is described in detail. The drawback of this is that it has high losses at higher speeds and therefore a combination of back-emf sensing and signal injection is to prefer.

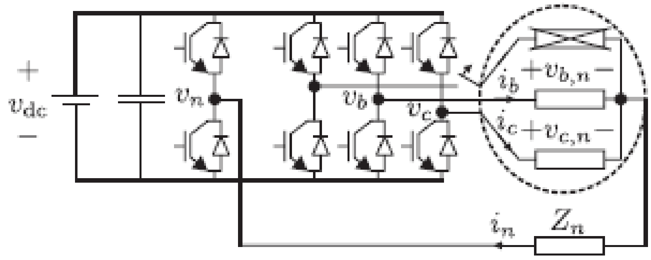

Figure 3.7: A Fault-tolerant drive (in fault mode where the faulted phase a is isolated) with the neutral drawn out and connected to Mosfets

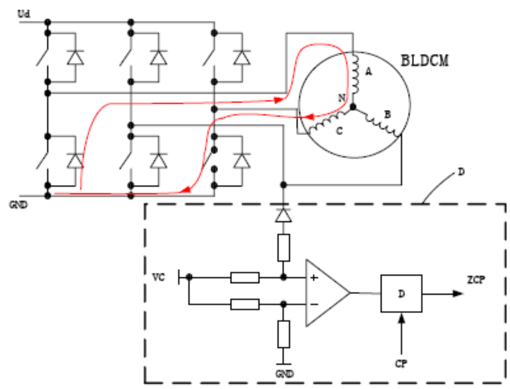

Figure 3.8: Electrical scheme of a circuit to measure induced emf in unexcited winding

4 Actuator control

It is assumed that the EMB(electro-mechanical brake) control module gets a desired braking force from the Control Unit hosting an application demanding a braking force, e.g. ABS or a stability control. There is also a measurement of the break force according to chapter 2.2.1.

The motor phase current in a BLDCM(Brushless DC-motor) is proportional to the generated torque, see chapter 3.1.4. A closed-loop control of the phase current is therefore an easy way to achieve high performance torque control. Current control also has the advantages that it can help protect driver circuits and motor from dangerous current peaks.

4.1 Torque control

As described earlier the motor phase current in a BLDCM(Brushless DC-motor) is proportional to the generated torque so torque means controlling the stator current. Current control also has the advantages that it can help protect driver circuits from dangerous current peaks.

To control the phase current a PI-controller was used because of its flexibility and that it is relatively easy to tune. The following controller was used

$$u(t) = K_c \left[e(t) + \frac{1}{T_i} \int_0^{\tau} e(\tau) d\tau \right] \tag{4.1}$$

which in the discrete time domain, using Backward Euler for the integration, becomes

$$u(k) = K_c e(k) + u_I(k-1) + K_c \frac{T}{T_i} e(k) \tag{4.2}$$

4.2 Anti-windup

Due to limited DC-link voltage the output of the controller, i.e. the commanded duty cycle of the PWM-signal, must be limited. However this causes the integral part of the PI-controller to accumulate the control error. This typically results in large overshoots because the integral part of the controller will still be high when the control error becomes small. This phenomenon is known as integrator windup. One way to solve this is to keep to integrator from being updated when the control signal has reached its maximum or minimum. Letting the controller act as normal when the control signal is within its limits.

4.3 Pulse Width Modulation(PWM)

PWM (Pulse With Modulation) is a common way to control the speed of DC motors. The principle is simple and contains two variables, the time of one period(T) and the duty cycle(D). The duty cycle describes the amount of time the signal is on in percent. When the signal is on the voltage is the source voltage, U and when it's of the signal voltage i zero. In this way the motor can be controlled with different duty cycles, e.g. a 50% duty cycle is equal to applying half the source voltage on the motor

When controlling a motor with PWM considerations to ripple in the produced torque has to be taken. A long period(T) gives a higher ripple in produced torque hence the stator current, which is directly proportional to the torque in a Brushless DC motor, gets a greater ripple. In figure 4.1 a graph over such ripple can be seen. Here the phase voltages is measured, on the connections to the motor, which is shown in the upper waveforms and the current which is shown in the lower waveform

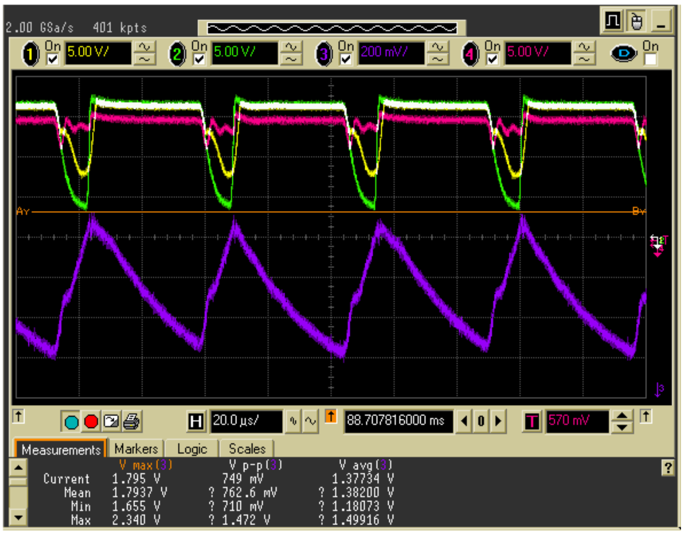

Figure 4.1: Waveforms of a BLDCm feed with PWM. The lower waveform is the measured phase current and the upper waveforms is the phase voltages

Nevertheless a shorter period gives more work for the power electronics which has to switch off and on faster with decreased period. This result in more switching losses hence motors is inductive loads which prevents fast changes. A to long period corresponding to a frequency within the audible region can a results in loud acoustic noise from the motor which has to considered when a silent actuator i wanted

A benefit with PWM is that there are no power consuming resistors needed when operating at lower speeds.

4.4 Filter

To get better performance of the controller the measured current was filtered with a digital filter. The used filter was a FIR windowed Kaiser filter with cut off frequency $\omega_c = 500Hz$ and $\beta = 6$.

4.5 Simulation and Experimental Evaluation

To test the controller designs before implementing them in software, they were first tested in matlab. In figure 4.2 the matlab workspace can be seen. The system contains of four different boxes. The motor box contains all motor parameters and is further described in chapter 3.1.3.

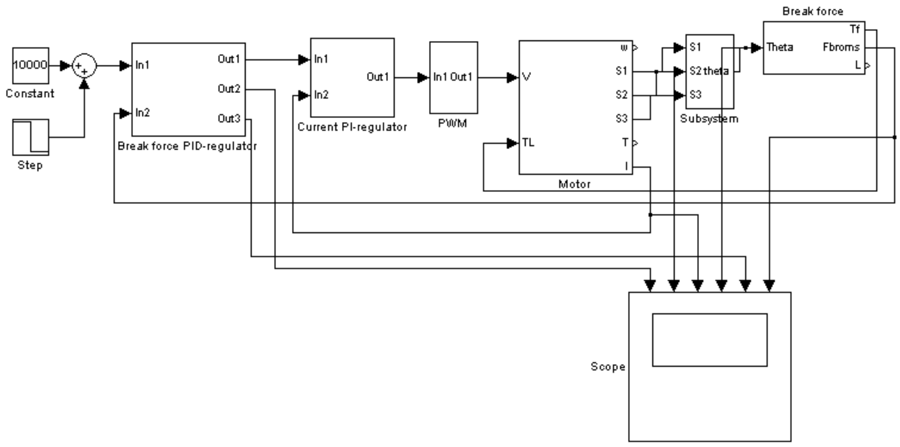

Figure 4.2: Screenshot from matlab simulink showing control strategy

The PWM box contains matlab functions converting a voltage into PWM-signals.

Right of the motor there are two blocks converting the hall sensor signals into an angle and one block simulating the brake mechanism. The brake force block converts the angle into a brake force and a torque load fed back to the motor. The brake force is fed back to the PID-controller.

On the left side of the workspace there are two different controllers. The first controller is a PID-controller and it controls the break force which is proportional to the current. The second controller is a PI-controller and it controls the current. They are both in closed-loop. The important results from these simulations was not to find the right parameters for the controllers, it was to find the right control strategy.

The PI-controller that controls the current in the above simulations were implemented and tested with the available hardware controlling the available Brusless DC-motor. In figure 4.3 the result from a step change in current command applied to the implemented PI-controller can be seen. The settle time is about $100ms$ and the sample time used in these tests were $4kHz$.

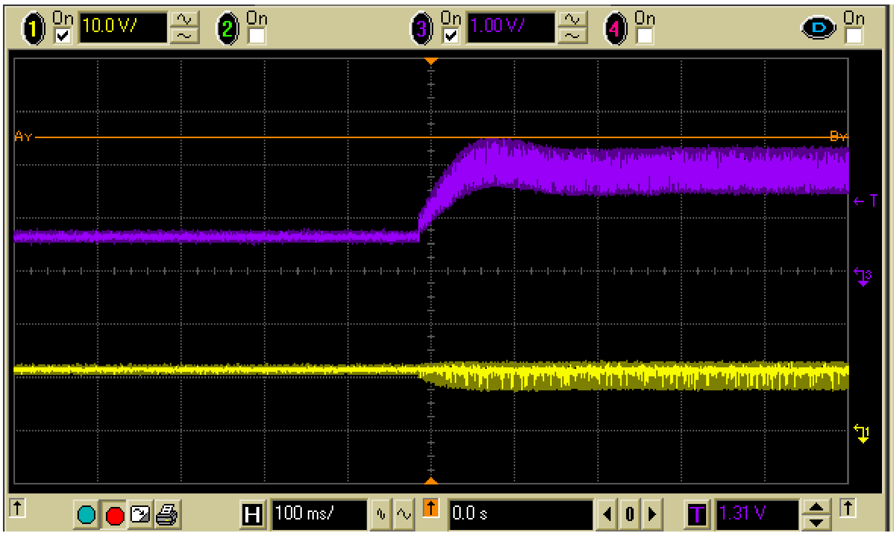

Figure 4.3: A step change applied to the implemented PI-controller. The upper waveform is the measured stator current

5 Fault detection

In safety-critical systems like a brake-by-wire system it's very important to quickly detect faults to be able to take appropriate actions in presence of a fault. To be able to take appropriate actions the diagnostic system not only has to be able detect a fault but also in which part of the system that the fault has occurred. In this chapter a diagnostic system with model-based fault detection is discussed.

5.1 Model Based fault detection

Model Based fault detection is a technique were system parameters is estimated to detect changes in the system that is supervised. This technique requires a model of the supervised system, in this case a brushless DC motor(BLDCm), and an algorithm for estimation of system parameters. In the following sections a model of the used BLDCm used for parameter estimation is developed and an algorithm for estimation of these parameters, called Linear Mean Square, is explained.

5.1.1 A simple model of a Brushless DC motor used for parameter estimation

The motor model developed is a general model for 3-phase brushless DC motor. The phases are Y-connected and feed with a PWM-signal through an H-bridge.A mathematical model of a brushless DC motor can be divided into to two subsystems, the electric and the mechanical.

Electrical subsystem

The phases in the modelled motor is Y-connected so a simple electric model where the inductance, induced currents in the rotor due to stator harmonic fields and iron and stray losses is neglected, can be represented by the scheme in fig. 5.1.

Where R_i is the phase resistance and e_i is the induced emf, $k_{Ei}\omega_r(t)$. With the definitions in 5.1 the mathematical model of one phase can be represented by

$$v_i - v_n = R_i i_i(t) + k_{Ei} w_r(t) \tag{5.1}$$

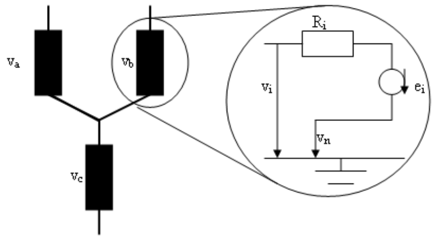

Figure 5.1: Scheme of phases in Brushless DC motor with the phase inductance neglected

where v_i is the phase voltage and v_n is the voltage in the star point Hence, for all phases the system of equations holds

$$\begin{bmatrix} v_a - v_n \\ v_b - v_n \\ v_c - v_n \end{bmatrix} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \begin{bmatrix} k_{Ea} \\ k_{Eb} \\ k_{Ec} \end{bmatrix} \omega_r(t)$$

As one of the coils is always open, e.g. only a and b is conducting, the equation becomes more simple and one can eliminate v_n. With only a and b conducting the following conditions occur, $v_a = v_{pwm}, v_b = 0, i_b = -i_a$ and $i_c = 0$. Substituting this into eq. 5.2 gives

$$v_{pwm} = (R_a + R_b)i_1(t) + (k_{Ea} + k_{Eb})\omega_r(t) \tag{5.3}$$

Where v_{pwm} is the pulse width modulated signal $v_{pwm} = v_s dty$, v_s is the source voltage that should be constant and dty is the duty cycle, $dty \in [-1, 1]$. Equivalent equations can be evolved for the five other cases that can occur, e.g. phase a and c conducting and so on.

To provide a cheaper and simpler solution for the model based fault detection a simpler model is looked for to avoid measuring of all three phases. Only the source voltage,v_s,source current i_s and the angular velocity ω_r are measured so a model with only this signals as input i looked for. Setting up the power balance gives

$$v_s i_s(t) = vspwm(t)\bar{i}_s(t) \tag{5.4}$$

Hence equation 5.3 can be written

$$v_s dty = (R_a + R_b)\bar{i}_s(t) + (k_{Ea} + k_{Eb})\omega_r(t) \tag{5.5}$$

Using this to form the average of equation 5.2 for all six cases one get

$$v_s dty = \frac{2}{3}(R_a + R_b + R_c)\bar{i}_s(t) + \frac{2}{3}(k_{Ea} + k_{Eb} + k_{Ec}\omega_r(t) \tag{5.6}$$

Substituting $\frac{2}{3}(R_a + R_b + R_c)$ with R and $\frac{2}{3}(k_{Ea} + k_{Eb} + k_{Ec}$ with k_E gives

$$v_s dty = R\bar{i}_s(t) + k_E\omega_r(t) \tag{5.7}$$

Here the voltage drop in the power electronics is assumed to be constant and subtracted from the measured supply voltage.

Mechanical subsystem

So far only the mathematical model only models the electric part but a model for the mechanical part is also wanted to be able to detect faults in the mechanical part of the system.

The torque in a brushless DC motor is proportional to the supply current $\bar{i}_s(t)$ and the magnetic flux linkage.

$$T_{el} = k_T \bar{i}_s(t) \tag{5.8}$$

For an ideal brushless dc motor $k_T = k_E$. Assuming that losses only occurs from viscose friction, $b\omega_r$ the torque balance can be written

$$J\dot{\omega}_r = T_{el} - b\omega_r k_T - T_{load} = k_E \bar{i}_s(t) - b\omega_r k_T - T_{load} \tag{5.9}$$

5.1.2 Linear Least Square Algorithm

Let a system be given by the differential equation

$$y(t) + a_1 y^{(1)}(t) + ... + a_n y^{(n)}(t) = b_0 u(t) + b_1 u^{(1)}(t) + ... + b_m u^{(m)}(t) \tag{5.10}$$

with the input u(t), the output y(t) and $y^n(t) = d^n y(t)/dt$. The coefficients represent the systems physical parameters which can be determined by parameter estimation. The equation 5.10 can also be written in vector form:

$$y(t) = \mathbf{x}^T(t)\theta \tag{5.11}$$

with

$$\theta^T = [-a_1 ... - a_n \; b_0 ... b_m] \tag{5.12}$$

$$\mathbf{x}^T(t) = [y^1(t)...y^n(t) \; u^(t)...u^m(t)] \tag{5.13}$$

To estimate the unknown parameters in θ an error is introduced as fallows

$$\hat{e} = y(t) - \mathbf{x}^T(t)\theta \tag{5.14}$$

where $\hat{\theta}$ is a vector of supposed parameters and $\hat{e}$ is an error in modelling at the moment t.

When sampling with the discrete time $k = t/T_s = 0, 1, 2, 3...N$, where T_s is the sampling time, the result can be shown in vector form

$$\begin{bmatrix} \hat{e}(1) \\ \hat{e}(2) \\ . \\ . \\ . \\ \hat{e}(N) \end{bmatrix} = \begin{bmatrix} y(1) \\ y(2) \\ . \\ . \\ . \\ y(N) \end{bmatrix} - \begin{bmatrix} \mathbf{x}^T(1) \\ \mathbf{x}^T(2) \\ . \\ . \\ . \\ \mathbf{x}^T(N) \end{bmatrix} \hat{\theta}$$

To solve this type of equations in practice Least Squares algorithm is often used, as in this paper. Least Squares algorithm means minimizing the sum if square errors

$$J = \sum_{t=1}^{N} \hat{e}^2 = \hat{e}^T \hat{e} \tag{5.16}$$

To conclude a vector of the supposed parameters, $\hat{\theta}$ the seeked solution is

$$\frac{\partial J}{\partial \hat{\theta}} = 0 \tag{5.17}$$

Equation 5.16 and 5.15 gives the solution of equation ref($ls_{s}olution$)

$$\hat{\theta} = [X^T(t)X(t)]^{-1}[X^T y] \tag{5.18}$$

In practice when trying to minimize J and this is needed to be done whenever new information is available e.g. whenever new input/output samples is available, which is with the rate of T_s. It's not practical to save huge amounts of data and solve an equation system each time. Instead a technique called Recursive Least Squares Method(RLS) can be used. Here a supposed model with the parameters, $\theta(t \hat{-} 1)$, is used to calculate the error in equation 5.14. To reach the solution in 5.17 the supposed parameters are updated following the negative gradient $-\frac{\partial J}{\partial \hat{\theta}}$ which, in a singular variable case, is

$$J = \sum_{t=1}^{N} \hat{e}^2 \Rightarrow \frac{\partial J}{\partial \hat{\theta}} = 2\hat{e}\frac{\partial \hat{e}}{\partial \hat{\theta}} \tag{5.19}$$

and equation 5.14 gives

$$\Rightarrow \frac{\partial J}{\partial \hat{\theta}} = 2\hat{e}\mathbf{x} \tag{5.20}$$

The parameters are then updated according to

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \gamma x(t+1)e(t+1) \tag{5.21}$$

Where γ defines the rate of which the parameters will be updated. It can either be a constant or, which is often used, the covariance matrix, P(t+1), which is defined by letting

$$P(t) = [X^T(t)X(t)]^{-1} \tag{5.22}$$

which by some matrix algebra gives

$$P(t+1) = P(t)[I - \frac{x(t+1)x^T(t+1)P(t)}{1 + x^T(t+1)P(t)x(t+1)}] \tag{5.23}$$

5.2 Using Model based Fault Detection on a Brushless DC-motor

The simple model developed in 5.1.1 can be written in the vector form eq. 5.24

$$y(t) = \mathbf{x}^T \theta \tag{5.24}$$

where

$$\theta^T = [R\ ] \tag{5.25}$$

$$\mathbf{x}^T = [i \; \omega] \tag{5.26}$$

To use modelbsed fault detection the current i and the speed ω has to be measured and sent to the LMS-algorithm fore estimation of the system parameters R and k_τ. And fault can then be detected when analysing changes in these parameters or the variance of the same.

5.3 Simulation and Experimental Evaluation

To test the LMS-algorithm simulations were made in matlab. A matlab function was made and was initialized in the same system used in chapter 4.5. To test the algorithm two different faults were inserted. The first fault is a step change of the resistance in the motor windings from 1 Ω to 0.85 Ω. The second fault is also a step change but this time a change of k_τ with 50. Figure 5.2 shows a simulation with a step change in parameter k_τ and figure 5.3 shows a simulation with a step change in parameter R.

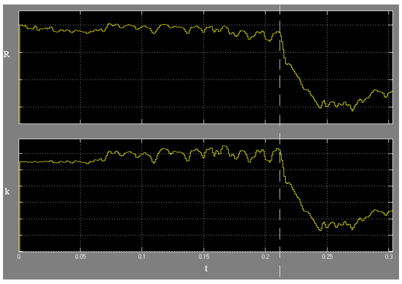

Figure 5.2: Step change in parameter k_τ

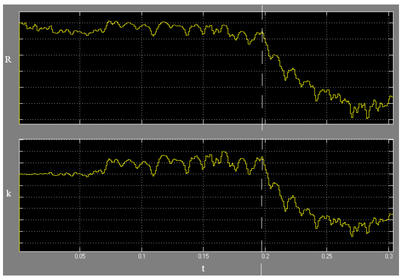

Figure 5.3: Step change in parameter R

The figures shows that the problem will be detected but it is impossible to see if it is a change in resistance or in k_τ. This is not a problem if the goal only is to detect faults. If the goal is to isolate the failure and be able to perform appropriate actions to maintain full operation of the actuator, this algorithm can not be used.

Experiemental evaluation with implementation on the Gast G1 processor board showed the same result as from the simulation at small currents. At larger currents great losses accured in the power electronics cousing the temperature to rise in those compenents which resulted in huge variations in the ciruit resistance. This high temperatures also made the in-built protection in the Mosfet:s to shut down the circuit. The fault detection treated this changes and shut downs a faults.

The tested fault detection worked very badly at low-speed due to the bad resolution in speed measurement with the hallsensors that only can detect whenever the rotor posistion has changed 15 degrees.

5.4 Alternative Methods

An alternative way to detect changes in the motor is to examine the current waveform during one pwm-cycle hence this waveform contains information about the winding inductance. By determining how much the slope of the current during the time that the PWM-signal is high the winding inductance can be calculated. This requires a very good resolution in the current measurement and a very fast sampling. One of the most common electrical failures is breakdown of the winding insulation causing bigger currents which leads to breakdown of the motor [9].

The used model based fault-detection described earlier in the chapter didn't have the ability to detect where in the system a fault occurred. Although this fault-detection could be applied to each phase separately having an algorithm for each phase running when the corresponding phase is excited the experiments showed that the algorithm was not good enough. An alternative solutions could be to replace the constant γ with a more complicated filter like a Discrete Square Root Filter used in [9] which might give a better estimation.

6 Implementation

6.1 Hardware

To test solutions described in earlier chapters a test-system was implemented. In this chapter the hardware of this test-system is described.

For controlling a BLDCM(brushless DC-motor) as described in chapter 3.1.1 you either have to have an external circuit that tells the system how to commutate the currents or the microcontroller can take care of this with a software solution, although it increases the load on the microcontroller. In this test-system the microcontroller was chosen to commutate the currents in order to get as few components as possible that could fail. The Hall-sensors was therefore connected directly to I/O-ports on the G1-card so the commutation could be done by the microcontroller. The used I/O-ports were three A/D inputs that were enabled to work as digital inputs.

To control the motor phase currents with the microcontroller the ports HDO(High Digital Out) and LDO(Low digital out) were connected to the motor phases, see ref(fig:hardware). Each phase was connected to both HDO and LDO were HDO, when turned on, served a phase with voltage and LDO grounded a phase . It was therefore very important to control that a HDO- and a LDO-port connected to the same phase never was turned on at the same time thus causing a short-circuit.

6.1.1 Additional current measurement

The torque created by the motor is proportional to the current flowing through the motor. Therefore the objective to control the torque can be made by controlling the current. Unfortunately the G1 card is very limited when it comes to current measurements, only voltage measurements are available. An additional card measuring the current and converting it to a voltage

The first objective was to find well suited components to the demands. The most important component to find were current sensors operating within the range of the current from the motor and the voltages that is able to be sensed with the G1 card. The ideal sensor should be able to sense currents from zero to 5 amperes, that is the nominal current of the motor and give an output voltage from zero to 5 volts, which is the voltage range that can be sensed by the processor. This would be the ideal case but since the output circuits on the G1 card can handle currents from 0 to approximately 1,5 amperes the resolution would be better with components that are able to sense currents from 0 to 1,5 or maybe 2 amperes.

Three different components close to the demands were found and all three were decided to be used on the additional card. The components were MAX4173T and MAX4173F from Maxim and ACS706ELC-05C by Allegro Microsystems. The components from Maxim need a resistor to be able to sense the current. The difference between the voltages on either side of the resistor is amplified by an OP-amp and gives an output linear to the current flowing through the resistor. The difference between MAX4173T and MAX4173F is the gain out from the OP-amp. MAX4173T has a gain of 20 and MAX4173F has a gain of 50. The voltage V_{out} can be calculated by the formula described in eq. 6.1 [7].

$$V_{out} = (\text{GAIN}) * R_{Sense} * I_{Load} \tag{6.1}$$

ACS706ELC-05C by Allegro Microsystems uses a different technology. It has a hall-sensor inside to sense the magnetic field created by the current. V_{out} is then given by eq. 6.2 [8].

$$V_{out} = \frac{V_{cc}}{2} + 0.133 * I \tag{6.2}$$

When the important circuits where found some estetic improvements where made by including components for hall-sensors, VRH(Voltage Reference High), VRL(Voltage Reference Low)and voltage measurements. To be able to easily connect the G1 card with the additional card the same connectors were used.

Since the performance of the different current sensors where unknown and had to be tested, jumpers where connected. The jumpers changes which current sensing component to use according to table 6.1. The fourth component is directly connected to pin 24 on the P1 connector.

Jumper	Pin 1,2	Pin 2,3
J1	MAX4173T(2)	ACS706ELC-05C(2)
J2	MAX4173T(3)	ACS706ELC-05C(3)
J3	MAX4173T(1)	ACS706ELC-05C(1)
J4	V_{cc}	VRH_{in}
J5	GND	VRL_{in}

Table 6.1: Jumpers

The rest of the pins on connector P1 and P2 where assigned according to table 6.2 and table 6.3. In the tables IP+ and IP- is in and out for current measurements with ACS706ELC-05C and RS+ and RS- is in and out for current measurements with MAX4173T. S1, S2, S3 are connections made for the three hall-sensors. NC means Not Connected.

The reason there are two sets of outputs from the hall-sensors are that the processor(HCS12) only can generate interrupts from either positive or negative slopes of the signal. To solve this every hall-sensor output is connected to two digital inputs on the processor. One channel gives interrupts when the signal is changed from 0 to 1 and the other when the signal changes from 1 to 0.

Pin	Function
1	GND
2	GND
3	$S1_{out}$
4	$S2_{out}$
5	$S3_{out}$
6	$S1_{out}$
7	$S2_{out}$
8	$S3_{out}$
9	NC
10	NC
11	NC
12	NC
13	NC
14	NC
15	NC
16	NC
17	$S1_{out}$
18	$S2_{out}$
19	$S3_{out}$
20	U_{out}
21	$J1_{out}$
22	$J2_{out}$
23	$J3_{out}$
24	$J4_{out}$
25	V_{cc}
26	V_{cc}

Table 6.2: Connector P1

Pin	Function
1	1IP-
2	1IP+
3	2IP-
4	2IP+
5	3IP-
6	3IP+
7	1RS-
8	1RS+
9	2RS-
10	2RS+
11	3RS-
12	3RS+
13	4RS-
14	4RS+
15	U_{in}
16	$S1_{in}$
17	$S2_{in}$
18	$S3_{in}$
19	VRH_{in}
20	VRL_{in}
21	VRH_{out}
22	VRL_{out}

Table 6.3: Connector P2

6.1.2 Protection of I/O-pins and A/D-converter

To protect the sensitive inputs to the A/D-converter and the I/O-pins the circuit in figure 6.1 was added to each input. The used protection is basically the same as that used for the pins that is protected on the G1 and G2-boards and limits the input voltages and currents.

6.1.3 Eagle

The program used to create the board was Eagle. Eagle is a simple and easy to use PC-program created by Cadsoft. Besides beeing easy to use another benefit with Eagle is that it can be downloaded for free on cadsofts homepage(www.cadsoft.de). The version that is free is called light edition and is limited to 2 layers and can only make boards up to 100 x 80 mm. The limitations are not a problem in this project since the additional card doesn't need more space or more layers.

The boards layout in Eagle is first made in a schematics workspace. The schematics for the additional card can be seen in Appendix A . As can be seen

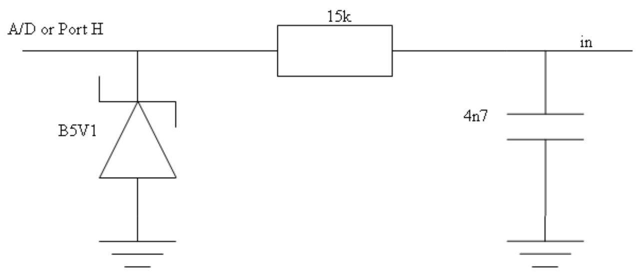

Figure 6.1: Protection for A/D-converter and I/O-pins

in the schematics several of the components needed additional resistors and capacitors to work properly.

When the schematics are done Eagle automatically transfers the components to a board workspace. All the components are then placed by hand on the board but all the wiring can be made automatically by Eagle. To avoid too many via-holes different thickness of the wiring were used. The current going through the current sensors demands thicker wiring then Eagles standard thickness. But if all wires should have the same thickness the amount of via-holes and the complexity of the card would be huge. Very little time was used to optimize the card because the only thing that was important was that the card worked. A picture over the board can be seen in Appendix A.

6.1.4 Hardware Issues

On the HDO and LDO-outputs there are power switches connected to be able to connect bigger loads to the card. The HDO power switch is a BTS712N1 from Infineon. It's designed to handle nominal load currents up to 1,9 A and can be fed with voltages from 5-34 V. The LDO-pins on the processor are connected to a low side power switch called BSP75N also manufactured by Infineon. This component can handle nominal load currents up to 0,7 A and voltages up to 60 V. A problem with both these components is the unlinear behaviour when the temperature of the drivers increases. The resistivity between small currents and the nominal current is in worst case more then doubled from approximately 700 mΩ to 1350 mΩ.

6.2 Software

6.2.1 Development environment

Within the GAST-project many drivers and sample programs have been developed for both G1 and G2. A problem is that they have been developed in different environments. Since most of the drivers and sample files for G1 is made in GMV XCC12 and in Eclipse for G2 it was easier to use them.

Development with XCC12 for G1

The development for the G1 board was made in GMV XCC12. GMV XCC12 includes a text editor for code writing, a cross compiler and terminal to read output from the serial interface between the computer and the card. It is also easy to upload s-records with GMV XCC12.

Development with Eclipse for G2

All development for G2 was made in Eclipse with the gcc-compiler. Eclipse was setup with compiler and writing of plug-ins by Petter Uvesten and Mattias Petterson. More information about this environment can be read in their master thesis report [6]. This environment supports code writing, compiling, a terminal and a plug-in to upload s-records to the board.

6.2.2 Motor Control Software

The software design started with an overview of what features and fault detection mechanisms the program should have. When all features were lined up some kind of prioritation list had to be made. The following prioritation list was made:

1. Commutation of the motor.
2. Keeping track of the motors position.
3. Having a reliable time base.
4. Communication.
5. Controlling the motor.
6. Fault detection.

Many of the things from the list are dependent of each other. To be able to regulate the motor or be able to detect faults, the first three things on the list have to work. An important thing to point out is that the breaks would not even work if any of the top five on the list completely fail.

Commutation and update of position

The easiest way to be sure that a processor makes a certain operation when it is needed, is to use interrupts. If the processor generates an interrupt when it is time to commutate it is easy to guarantee that it would be made within a certain time. The commutation depends on the hall-sensors and interrupts were made every time any of the three hall-sensors changed state by using digital inputs.

The first thing that happens when an interrupt is generated from the digital inputs is that the commutation is performed according to figure 3.2. The next thing that happens is that the position is updated. The function that updates the position always has a saved copy of the state before the interrupt. The new state is compared with the old state and the two new states that are possible. The only two states possible is if the motor has turned forwards or backwards. If the new state is not one of the two possible states an error message is displayed. If it is one of the two states the position is updated with the old position plus one or minus one. The whole sequence performed when an interrupt from the digital inputs is performed can be seen in figure 6.2.

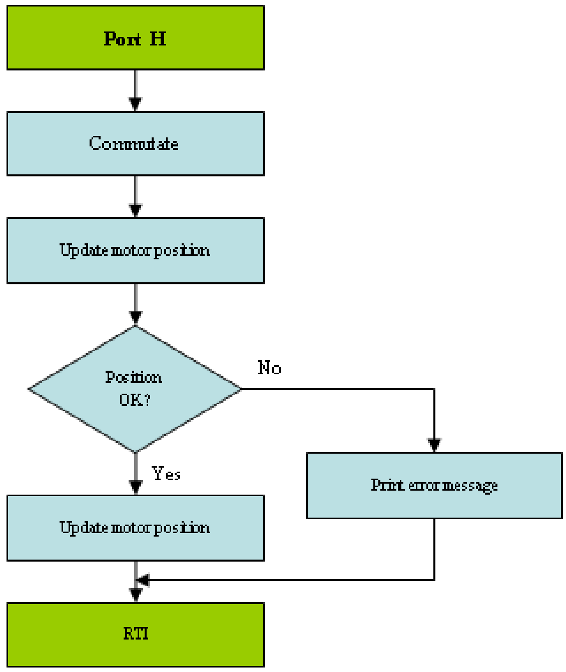

Figure 6.2: Flowchart when interrupts occurs on portH

Periodic interrupts

The third most important thing according to the prioritation list is a reliable time base. The reason why it is important with a reliable time base is that there are several important calculations made were time is an important factor such as velocity-, integration- and filter-calculations. An easy way to get a reliable time base is to generate interrupts at fixed time steps. Both MPC565 and HCS12 have periodic interrupts called Real Time Interrupt(RTI) on HCS12 and Periodic Interrupt Timer(PIT) on MPC565 that both can be used to perform periodic interrupts. The whole sequence performed when a periodic interrupt occurs can be seen in figure 6.3.

Communication

In a real application a communication protocol also have to be handled by the processor. There are several different communication protocols that are recommended for automotive applications like TTCAN, TTP/C and Flexray. At this moment software drivers are beeing created for all of these protocols for both G1 and G2, and when they are done it would be easy to implement one of these protocols in this program. One problem with G1 is that there is not much time left for the processor to handle the communication. With the recommended response time of the system and the demanding program it is not sure that the processor will be able to handle the communication and keep the same

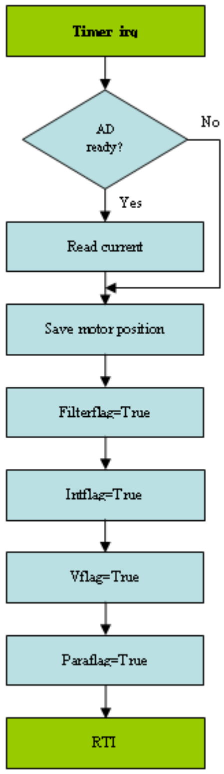

Figure 6.3: Flowchart when periodic interrupt occurs

performance.

Control software

The control of the motor is dependent on several different steps to work. To implement these in software the most important thing is to have a reliable time base. The PI-regulator needs an integrator to work and the integrator is very time dependent. Since the measured current is very noisy the regulator needs to filter the current. The main programs flowchart can be seen in figure 6.4.

The filter is also time dependent. The filter is according to table 6.4 and 6.5 the most time demanding operations. The difference between the two different processors is very significant when it comes to calculate the filtered values.

The G1 board with the HCS12 processor have no float processing unit which means that every float operation takes a lot more time. Since the program is very time demanding float operations could not be used during this project. This problem caused a lot of extra work since no values can be transformed to physical values like current or torque and scaling factors had to be used frequently. Another problem is that the processor is occupied with filter calculations almost half the time. This prevents future applications like sensor less control, membership or other systems implemented to detect failures. To run

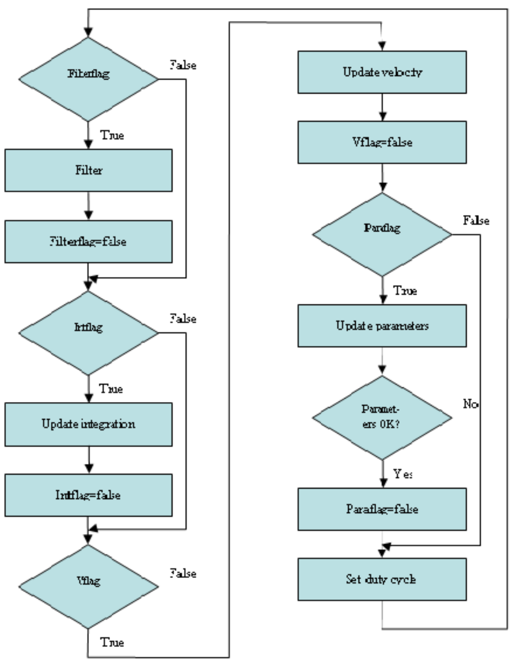

Figure 6.4: Flowchart for the main program loop

another application on the processor at the same time is also impossible.

The MPC565 processor has a floating point unit and is much faster when it comes to filter calculations. The filter calculations are done within 8 μs. This gives more time for other applications, bigger filters or higher sampling frequencies.

Fault detection

This part of the program constantly tries to detect fault by using an LMS-algrorithm. The whole theory about fault detection used in this program can be found in chapter 5. The physical values calculated with this function are compared with values that are needed for problem free operation. If these values go outside the limits an error message is displayed.

Rules for interrupts

The periodic interrupt is only periodic if the actions performed inside the interrupt always take the same time. The clock does not work while the interrupt is performed. The interrupts can not be interrupted by another interrupt. If another interrupt with higher priority occurs during the periodic interrupt it will not take effect until the periodic interrupt is done. On HCS12 This means that the commutation-interrupt with a higher priority can not be guaranteed to happen momentarily. The maximum delay is the same time it takes to run the periodic interrupt which can be found in table 6.4 and 6.5.

On MPC565 an external interrupt can not be made when the MSR[EE]-bit is cleared. This bit is cleared automatically when an external interrupt occurs. This prevents other interrupts before the important registers have been saved. If another interrupt occurs while MRS[EE] is cleared it will be delayed. If multiple interrupts occur during an interrupt the interrupt with the highest priority will be next to be performed. On MPC565 the MSR[EE]-bit can be set manually which means that interrupts can be allowed to happen during other interrupts. In this case it is not necessary since the TPU takes care of the commutation and there is more then enough time to update the position. A complete table with program times can be found in table 6.5 and table 6.4.

6.2.3 Timings on G1 and G2

To get an idea on how strained the processors are when running the program time measurements were made.

Function	Clock-cycles	Time(μs)
Filtering	1863	466
Digital interrupt	302	76
Control	58	15
timer$_irq$	3990	998
Parameter estimation	255	64

Table 6.4: Timings for functions on G1

6.2.4 Time Processor Unit(TPU)

The TPU is a feature on the MPC565 processor found on the G2 board. The TPU is an intelligent semi-autonomous microcontroller operating simultaneously with the CPU. The TPU is controlled by the CPU but can be programmed individually and be run without any strain on the CPU. There are three TPU-units found on the processor named A, B and C. They all have 15 IO-pins that can be assigned to different assignments. These three modules can handle a special type of code called micro-instructions. They can all work independent and have their own memory. There is 15 different TPU-functions burned into a ROM but it is also possible to write own micro-instruction-code to customize own TPU-functions. More information on how to use the TPU can be found in appendix B.

The TPU is designed to handle inputs and outputs without any strain on the processor. Typical TPU-functions are motor control or PWM signals. In

Function	PIT Clock-cycles	Time(μs)
Filtering	75	5
Control	58	15
timer$_irq$	3990	998
Parameter estimation	255	64

Table 6.5: Timings for functions on G2

this project two different TPU-functions were used, PWM and HALLD. Both are TPU-functions burned into the ROM. These functions were used to create interrupts from the hall-sensors and to create PWM signals to drive the motor. There are TPU-functions specially made to control BLDC motors but they need additional hardware and could therefore not be fitted in to this project. An interesting future work to do within this field would be to write a TPU-function that can both commutate the motor with PWM signals and keep track of the motors position without involvement of the CPU.

7 Conclusions

7.1 Summary

In this report two processor boards from the GAST project has been used to control a Brushless DC-motor(BLDCm). The used processor boards has either a PowerPC-processor(MPC565) and a HCS12 or only a HCS12. The purpose of the tests has been to see how well these processor boards could act as wheel-nodes in a distributed brake-by-wire system.

Both processors have the ability to control the motor in a way acceptable for use in an electromechanical brake. But the need of a fast time-triggered communication, e.g. Flexray, in distributed system makes the HCS12 come a little bit short with its old architecture and its eight MHz. The MPC565 on the other hand has great capabilities of both controlling an actuator like a brushless DC-motor and a Synchronous motor and at the same time handle a quick time-triggered communication. In the implemented control of a brushless DC-motor, which had a sampling rate of 4 kHz, only 10% of the CPUs resources was used. By implementing more of the control functions, then was done in this thesis, in to the MPC565 co-processors called Time Processor Unit(TPU) the processor can be relieved even more. Tests has shown that the PowerPC even though its a general purpose processor would without any problem handle control of an actuator, a fast communication and still be able to run quite heavy application like ABS.

The MPC565 has the disadvantages of being approximately six times more expensive then a HCS12 but the falling prices and the great flexibility of a MPC565 speaks for the use such processors.

Brushless DC motors is a quite good choice for use in an electromechanical brake due to its reliability and that it's easy to control. But the ripple in the produced torque from the tested motor speaks for the use of the synchronous motor instead where such ripple is absent. Even though this type of motor is more complicated to control, hence vector control is needed which requires more heavy calculations then the control of a Brushless DC motor, which is similar to that of a classic DC motor. If a MPC565 is used these extra calculations won't affect the burden of the CPU thus this can be taken care of by the TPU. The best actuator to use could be a Fault-Tolerant Permanent Magnet synchronous motor due that a failure rate of less then 10^{-7} failures per hour can be reached with this type of machine. Which meets the safety requirements in the aero industry.

Sensor less control of the actuator is probably a good choice hence it would reduce the number of components used to control the motor. Reducing the number of components is both cost-saving and increases the reliability of the control system thus there are less components that can break. An actuator in electromechanical brake must be able to work at both standstill and low-speed and the most common way of sensor less control which i based on back-emf sensing i therefore not possible. But a combination of this technique and a technique called Signal Injection described in [14] would be able control the actuator in all speeds. Sensor less control can also be used as a redundancy to a sensor controlled motor.

An electromechanical brake needs to have a quick and accurate fault-detection system. The tested Model based fault-detection based on a simplified model of the BLDCm and a Linear Mean Square(LMS) estimate did not fulfil these needs. Probably due to the simplicity of the LMS algorithm. There were also great problems with huge changes in resistance in the used power electronics, with up to 100% changes in resistance. This though, can probably be solved by measuring the voltage drop over the MOSFETs to see changes in resistance. This would also make it possible to a fault-detection in the power electronics. To be able to use fault-tolerant machines a fault-detection able of detecting which phase that is broke is needed. This to be able to continue operation after a single fault which is needed to reach the safety requirements.

7.2 Future work

A future work to test if its possible to use a fault-tolerant machines with sensor less control in an electromechanical brake would be interesting. Is it cost-efficient to use this type of machine and exactly how good performance can be achieved with sensor less control?

To do these tests it would be interesting to have a relevant mathematical model of the mechanical system of a possible electromechanical brake to be able to do a more accurate research in matlab/simulink. And, there implement and test the above mentioned ideas of sensor less control, the use fault-tolerant machines and develope a better fault-detection.

Simulink is great tool to for easy tests of ideas but to make these testing relevant very good models of used components is a must. A way to get a good model of e.g. the actuator is to buy the actuator and use the Matlabs System Identification Toolbox to get calculate a model that can be used for simulations and make the step from simulation to reality as small as possible. Making this step as small as possible would probably also make it easier to use the built-in code-generator in simulink to generate code for testing on a real system. Problems with this sequence of work are to get use of the co-processor on the MPC565. But still, it would be a quick way to work and such modifications can be made later when a good control system i designed.

Appendix A

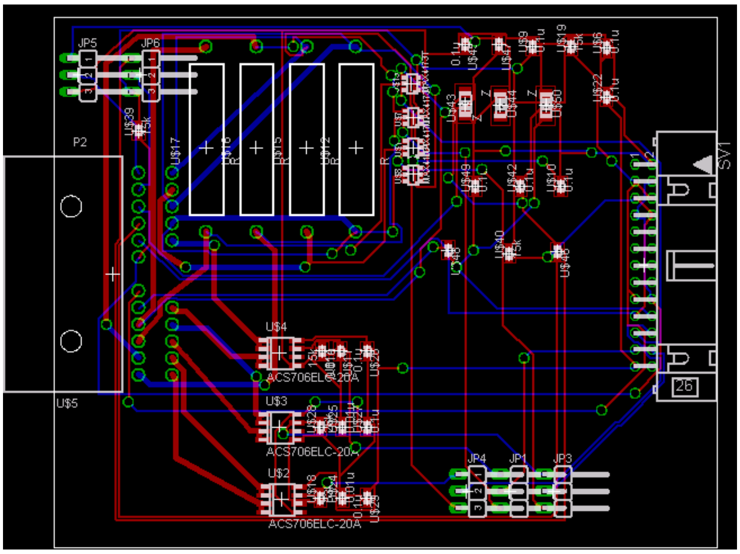

Figure 1: This the layout of the contructed board for among other things additional current measurement made in Eagle

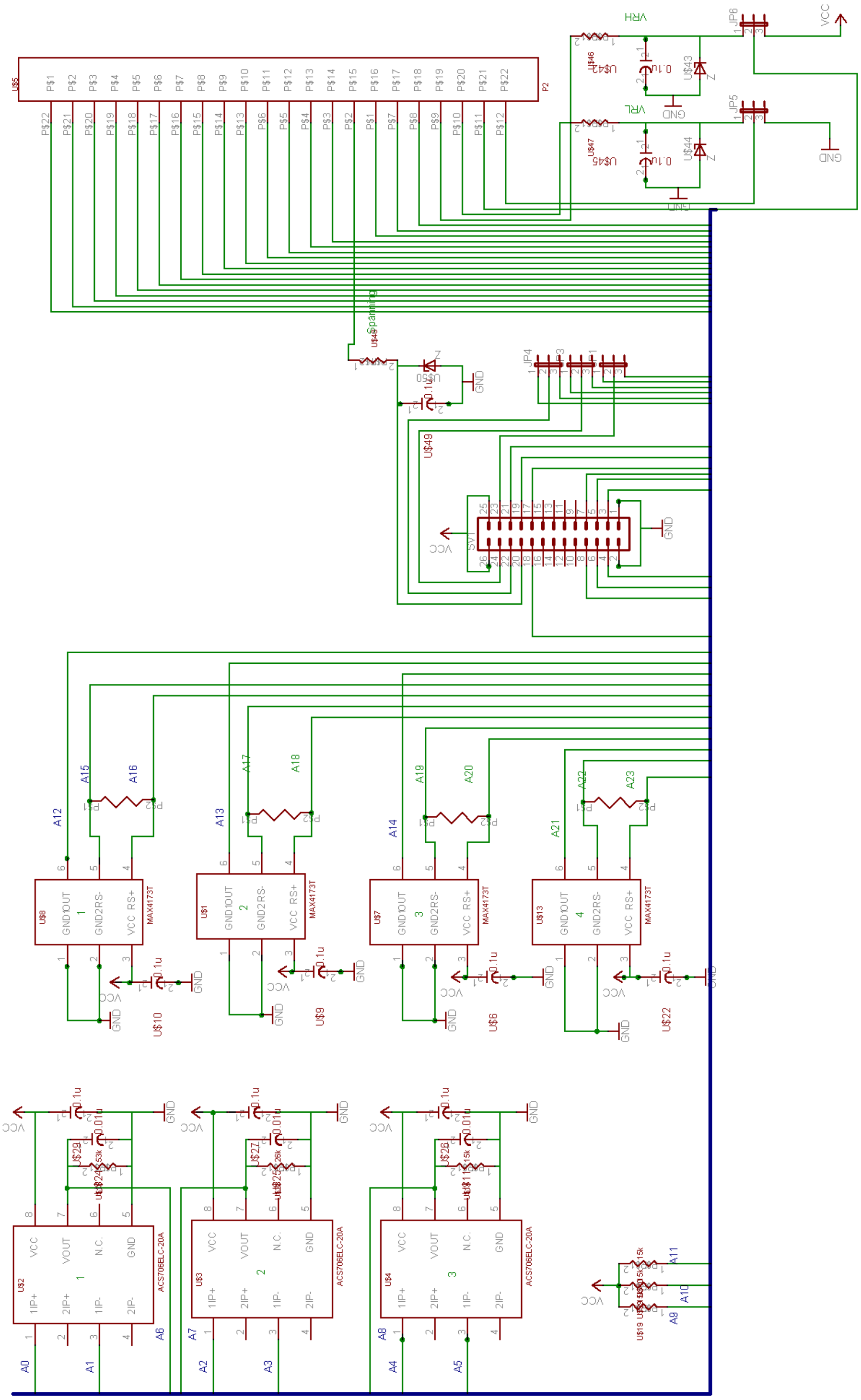

Figure 2: Schematic of the constructed board for additional current measurement made in Eagle

Bibliography

[1] Nicola Bianchi and Silverio Bolognani. Faulttolerant pm motors in automotive applications. Technical report, Department of Electrical Engineering, University of Padova, 2005.

[2] CEDES. The cedes project homepage. *http://www.cedes.se/*.

[3] Yi L. Murphey(2) Abul Masrur(1) ZhiHang Chen(2) and BaiFang Zhang(2). A fuzzy system for fault diagnostics in power electronics based brake-by-wire system. Technical report, (1)U. S. Armny RDECOM-TARDEC, Warren, M148397-5000) (2)Departmnent of Electrical and Computer Engineering, The University of Michigan-Dearborn Dearborn, MI 48127, USA, 2005.

[4] GAST. The gast project homepage. *http://www.chl.chalmers.se/gast*.

[5] J. A. Haylock B. C. Mecrow A. G. Jack S. Green and D. J. Atkinson. The need for fault tolerance in an aeroengine electric fuel control system. Technical report, University of Newcastle, UK, 1999.

[6] Petter Uvesten Mattias Pettersson. A distributed FlexRay-based research platform. Technical report, Department of Computer Science and Engineering, Chalmers University of Technology, 2005.

[7] Maxim. *Maxim MAX4173T/F/H Manual*, December 2004. Download from http://pdfserv.maxim-ic.com/en/ds/MAX4173F-MAX4173T.pdf.

[8] Allegro MicroSystem. *Allegro MicroSystem ACS706ELC-05C Manual Rev.1*, 2005. Download from http://www.allegromicro.com/datafile/0706-05C.pdf.

[9] Olof Moseler and Rolf Isermann. Application of model-based fault detection to a brushless dc motor. Technical report, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 47, NO. 5, October 2000.

[10] Lennart Harnefors(2) Oskar Wallmark(1) and Ola Carlsson(1). Control algotihms for a fault-tolerant pmsm drive. Technical report, (1)Department of Energy and Enviroment, Chalmers University of Technology, Gothenburg, Sweden (2)ABB Power Technologies, Ludvika, Sweden, 2005.

[11] R.Krishnan. *ELECTRIC MOTOR DRIVES, Modeling, Analysis and Control*. Prentice Hall, Inc., 2001.

[12] Freescale Semiconductor. 56f8300 hybrid controller used in control of electro-mechanical brake. Technical report, Department of Linguistics, Uppsala University, 2000.

[13] Freescale Semiconductor. 56f8300 hybrid controller used in control of electro-mechanical brake. Application Note(AN1999), Rev.0, Download available at www.freescale.com, Nowember 2004.

[14] Oskar Wallmark. On control of permanent-magnet synchronous motors in hybrid-electric vehicle applications. Technical report, Department of Electric Power Engineering,Chalmers University of Technology, 2004.