Single board computer based control of an active magnetic bearing

---

A dissertation presented to

The School of Electrical, Electronic and Computer Engineering

North-West University

In partial fulfilment of the requirements for the degree

Magister Ingeneriae

in Electrical and Electronic Engineering

by

Dewald Herbst

Supervisor: Prof. G. van Schoor
Assistant-Supervisor: J. Jansen van Rensburg

July 2008

Potchefstroom Campus

Declaration

I hereby declare that all the material incorporated in this thesis is my own original unaided work, except where specific reference is made by name or in the form of a numbered reference. The work herein has not been submitted for a degree at another university.

Signed: _____ Date: _____

Dewald Herbst

Acknowledgements

Firstly I want to thank M-Tech Industrial for supplying me with the necessary funds to continue my studies in engineering. Also for the funds supplied to buy all the required hardware to complete my project.

Secondly, I want to thank the following individuals for their involvement throughout the course of this project:

- Professor George van Schoor, my supervisor.
- Jacques Jansen van Rensburg, my assistant-supervisor.
- Doctor Eugén Ranft, the project manager.

"*Do everything without complaining or arguing, so that you may become blameless and pure, children of God without fault in a crooked and depraved generation, in which you shine like stars in the universe*..."

Philippians 2:14, 15

Summary

The McTronX Research group at the North-West University is currently conducting research in the use of active magnetic bearings (AMBs) for a flywheel energy storage system (FESS). Together with this, the aim of this project is to take the level of control of such AMB systems to an industrial level. Instead of using a rapid prototype dSPACE® controller, a single board computer (SBC) is proposed. Issues to be addressed, includes: SBC overview, control system specifications, skills development for SBC, implementation and evaluation.

All the digital and analogue input/output signal requirements for the FESS are determined prior to specifying an SBC. Six different SBCs were compared and in the end the SBC6713eII from Innovative Integration (II) was chosen and sourced. The SBC6713eII complies with all the requirements specified by the client.

Two interface boards were used to integrate the SBC with the FESS. The first board contained all the protection circuitry to protect the controller from faults that could occur on the sensor and actuator side of the FESS and is used to connect the dSPACE® to the FESS without the SBC. After the hardware was integrated, the software/firmware integration started. On the SBC, the PD control was implemented for the AMBs as well as the voltage over frequency control for the PMSM. A graphical user interface (GUI) was further developed on a host computer to monitor the FESS.

Four tests were done on the integration of the SBC with the FESS. Firstly the performance of the controller with regard to the control algorithms was tested. The stability and sensitivity analyses of the system followed and ended with the PMSM start-up test. The control algorithm execution time was longer than expected and adjustments to the sampling time had to be made. Stability tests showed a decrease in bearing stiffness and damping, which was due to low pass filters on the analogue to digital converter board. The sensitivity of the system also increased due to the effect of the filters on the system.

The inconsistency in bearing damping and stiffness, obtained from the stability tests was verified by adding the filters to the simulation. These filters caused an attenuation of less than 1 dB, but resulted in a phase shift of -36.3° in the control loop.

Industrial control was realised using an SBC, but further work is still necessary. The areas identified for future work is: non linear control algorithms, low noise digital power amplifiers, speed sensor and the PMSM drive.

Table of contents

List of figures

List of tables

List of abbreviations and acronyms

ADC	Analogue to Digital Converter
ALU	Arithmetic Logic Unit
AMB	Active Magnetic Bearing
bps	bits per second
CCS	Code Composer Studio
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
dc	direct current
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
GUI	Graphics User Interface
I/O	Input/Output
ISO	International Standards Organization
kbps	kilo bits per second
ksps	kilo samples per second
LQ	Linear Quadratic
MBps	Mega Bytes per second
MFLOPS	Million Floating point Operations per Second
MIPS	Million Instructions per Second
NCS	Network-based Control System
OSI	Open Systems Interconnection
PA	Power Amplifiers
PBMR	Pebble Bed Molecular Reactor
PLC	Programmable Logic Controller
PMSM	Permanent Magnet Synchronous Machine
PC	Personal Computer
PD	Proportional plus Derivative
PID	Proportional plus Integral plus Derivative
PWM	Pulse Width Modulation
rad/s	Radians per second
rpm	Revolutions per minute
RTD	Resistance Temperature Detector

SBC	Single Board Computer
TI	Texas Instruments
TTL	Transistor-Transistor Logic
VDU	Visual Display Unit

1
Chapter

Introduction

This chapter motivates the project and gives a brief description of active magnetic bearings (AMBs). An introduction to the current and the proposed control system follows thereafter. The issues to be addressed and the research methodology followed, is the discussed. The thesis overview is finally given.

1.1 Background

The McTronX research group at the Potchefstroom campus of the North-West University is currently busy conducting research on active magnetic bearings (AMBs). These AMBs are used in high speed applications such as helium blowers in industry. The level of industrial control of AMBs is therefore of great importance for the realization of a helium blower system in industry. This project is all about the realization of the control for high speed AMBs on an industrial level.

1.1.1 Active magnetic bearings (AMBs)

Active magnetic bearings (AMBs) are unique due to the fact that there is no physical contact between the rotor and the bearing. AMBs are used to suspend high-speed rotors to overcome the limitations posed by conventional bearings. Very fast rotational speed up to the limit of material strength is possible. There is no wear and no need for lubrication. The support dynamics and rotor position are refined in the controller according to the specific needs of each application [1].

Main application areas are [1] [2]:

- High vacuum
- Machining, machine tools
- Turbo compressors, -generators
- Cryogenics
- Clean rooms
- Electric drives

- Textile machinery
- Energy storage
- Vibration isolation
- Applications in space and physics

AMBs have many uses, but to control them, a better understanding is needed of their operation.

1.1.1.1 AMB operating principle

AMBs make use of the basic principle of an electromagnet. When an electric current flows through a conductor, a magnetic field is imposed perpendicular to the flow of current. The magnitude of the force acting on the rotor from a single electromagnet is given by (1.1) [1]:

$$f = \frac{B^2}{\mu_0} A \tag{1.1}$$

$$B = \frac{\mu_0 NI}{x_0} \tag{1.2}$$

f is the electromagnetic force, μ_0 the permeability of free space and A the total area of effect between the two surfaces. B is the flux density in the air gaps. In (1.2), N is the number of turns on the coil, I the current and x_0 the distance between the two surfaces. The larger the area of effect, the higher the attracting force would be. For the electromagnet to retain its force acting on the magnetic material as the distance x_0 increases, as shown in figure 1.1, the flux density should also be retained. To do this, the current should be increased to produce a higher flux. The current should be proportional to the square of the distance to keep the flux density the same. This is because the current is directly proportional to the flux density, given by (1.2), and the force is directly proportional to the square of the flux density, given by (1.1).

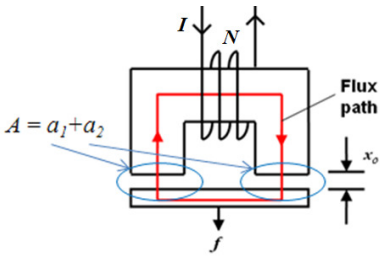


Figure 1-1: Force illustration of an electromagnet [1], [3]

The electromagnets are placed around the rotor to form a radial bearing as well as perpendicular to a disc to form an axial bearing. By applying current to the specific electromagnet, a force is generated, which will attract the rotor. By actively controlling the currents in the electromagnets, the position of the rotor can be controlled.

1.1.1.2 Components of an AMB

A complete AMB consists of 5 main components, as shown in figure 1-2:

- Sensor(s)
- Controller
- Power amplifiers(s)
- Electromagnet(s)
- Rotor

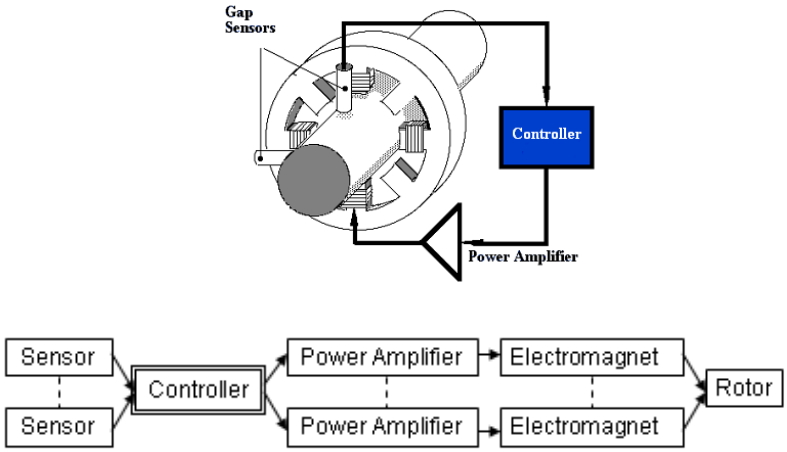


Figure 1-2: The 5 main components of an AMB [2]

The contact free sensors are used to detect the position of the rotor. Typical sensors used, are eddy-current displacement sensors. These sensors make use of high frequency alternating current that is applied to an air coil cast in a housing [2]. A voltage proportional to the clearance between the coil and the magnetic material (the rotor in this case) is generated by an eddy current sensor with a bandwidth of 10 kHz.

The most critical components of the system are the controller and the power amplifiers (PAs), because of the power requirements for frequencies from dc

to the kHz range [1]. The controller can be further divided into 5 components:

- Analogue to digital converters (ADCs)
- Anti-aliasing filters
- Digital controller
- Low pass filters
- Digital to analogue converters (DACs)

These five components of the controller are shown in figure 1-3.

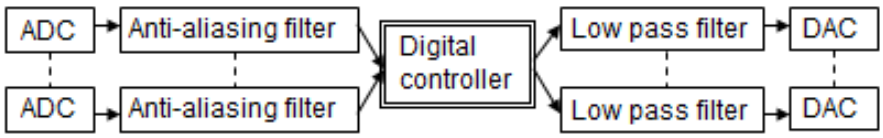


Figure 1-3: Components of the controller

The analogue signal from the sensor is converted to a digital value, filtered to remove the high frequency components and processed by a digital controller, which could either be a digital signal processor (DSP), central processing unit (CPU) or a field programmable gate array (FPGA). A combination of these can also be used to decrease response time. To avoid the scaling problems during the experimental work, a floating point signal processor is a good choice [1] [5]. A digital algorithm is then used to calculate an output value. Most commonly used digital control algorithms are the proportional plus derivative (PD) or proportional plus integral plus derivative (PID) and sometimes the more complex linear quadratic (LQ) [2] [4]. PD control is natural for AMBs. The proportional feedback manifests itself simply as proportional to mechanical stiffness and the differential feedback coefficient, as proportional to the mechanical damping. Stiffness to static load change can be increased drastically by adding an integral term [1]. The output value is then low-pass filtered and converted back to an analogue reference.

The PAs use a voltage reference to control the electromagnets with a current proportional to that voltage. Switch-mode PAs are used due to improved efficiency when compared to analogue amplifiers.

Both the electromagnet and rotor are built up with laminated sheets to reduce the eddy current losses which will result in higher operating temperatures. Higher operating temperatures will affect the magnetic properties of the electromagnet and rotor and in effect cause a lower stiffness of the rotor.

For a single rotor to be suspended, two radial- and one axial bearing are needed. This includes 4 electromagnets per radial bearing and 2 electromagnets per axial bearing. In the case of the contact free sensors, one sensor is needed for each of the six mechanical degrees of freedom (x, y, z, θ_x, θ_Y, θ_Z) to levitate a single piece of material. In the case of a rotor, a rotation will occur in one of the axes (θ_x or θ_Y or θ_Z), which means that one less sensor is needed [4]. Therefore 10 electromagnets, 10 PAs, 5 eddy-current sensors, 1 controller with five control loops and 1 rotor are needed. The controller should have at least 5 ADCs and 10 DACs. The basic bearing layout is displayed in figure1-4.

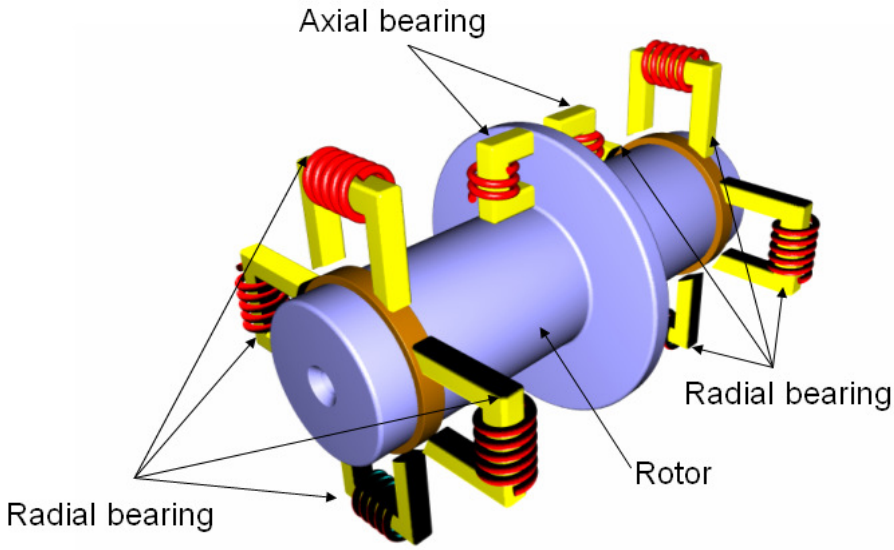


Figure 1-4: A basic AMB with 2 radial- and 1 axial bearing

The basic operation and components of an AMB were discussed in this section. Before a problem statement can be made regarding industrial control, the current control system has to be reviewed.

1.1.2 Current control system

The current system makes use of a personal computer (PC) with an operating system (OS), a stand-alone dSPACE® controller and MATLAB®. The

dSPACE® controls the complete AMB system, while the PC is only used to program the controller and display the stratus of the system. This is shown in figure 1-5. In this configuration, each system would make use of a PC with a dSPACE® controller. The cost of such a system is very high due to the fact that this is a high-tech stand-alone development system. The dSPACE® controller was perfect for the application at the start, but a more industrialised solution was needed.

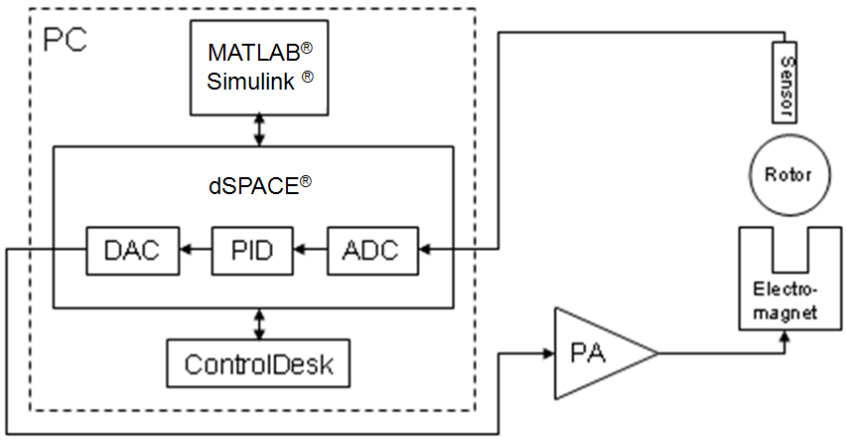


Figure 1-5: Current AMB system configuration

The current dSPACE® controller costs roughly R 280 000. Together with the PC and optical fibre communication between the PC and dSPACE® controller, this would average about R 300 000 for hardware that is not on an industrial level.

1.2 Problem statement

The purpose of this project is to take the level of control one step closer to industrial system specifications. A complete stand-alone dSPACE® controller will be replaced by a single board computer. This will mainly be developed for a flywheel system.

A single board computer was proposed, which is a digital controller board with certain input/outputs (I/O). It is mostly used and developed for industrial applications.

The minimum SBC specifications are as follows:

- 10 DACs (≥20 kHz)
- 5 ADCs (≥20 kHz)

- Fit in a double euro enclosure (6U)
- Ability to use a GUI on a desktop PC to adjust values and give commands to the SBC.

The proposed system consists of a PC with Visual Studio (VC8) and code composer studio (CCS). This is only used to program the single board computer (SBC) during the development phase. The PC can be disconnected after programming or Visual Studio can be used to create a graphical user interface (GUI) to adjust certain values and even to monitor the status of the AMB.

The SBC consists of ADCs, DACs, a digital controller and an Ethernet port used by the GUI. The configuration is shown in figure 1-6.

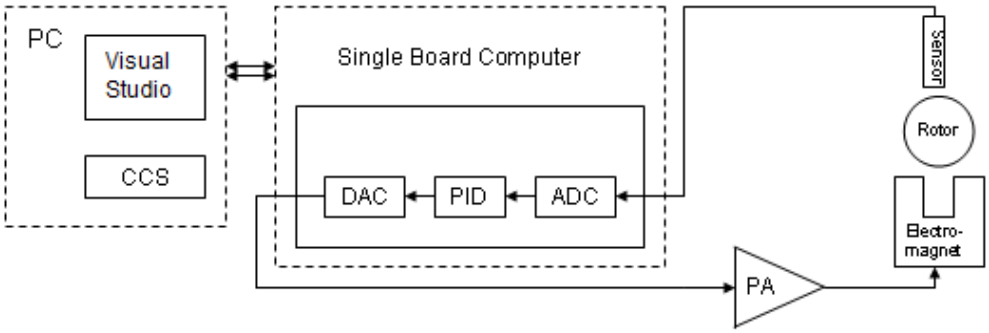


Figure 1-6: Proposed AMB system configuration

This system depends totally on the code on the DSP and the SBC itself. If this code is not well developed, a system failure can occur, causing AMB failure.

This system will be used to suspend the flywheel project of the McTronX Research Group. The flywheel project is an energy storage system, which makes use of a mechanical flywheel to store the energy. The momentum of the flywheel is then used to generate the electricity during power failure.

Every project has issues of its own. To be able to complete this project, these issues should be addressed. The following two sections will explain the issues and the associated research methodology.

1.3 Issues to be addressed

The high speed AMB used in the flywheel project has unique control issues. The main issues with regard to the implementation of a high speed AMB on a SBC are listed below.

1.3.1 Overview on SBCs

A better understanding of SBCs is necessary. This includes: technology used, available modules, cost, industrial usage etc. The industrial standards, to which these SBCs comply, are also very important.

1.3.2 Control system specifications

Before ordering a SBC, a more complete specification is needed. This should include the technical specification of the analogue input/output, digital input/output, processing power, communication, etc. This will be drawn up and used to identify the SBC. A specification regarding the GUI and communication aspect will be compiled as well.

1.3.3 Skills development for SBC

The specific SBC has its own architecture and programming software. This software should be obtained and loaded on a PC. Skills have to be developed so that it can be used to write and compile the code for the SBC.

1.3.4 Implementation/Integration

The SBC will initially be used to control an existing AMB model in the laboratory. The control system will be tested and improved where necessary. After completion of the test, the system will be integrated with the flywheel project.

1.3.5 Evaluation

After implementation, the performance of the SBC will be evaluated and compared to the results of the existing dSPACE® system. This includes the stability tests on all 5 AMBs as well as a sensitivity analyses.

1.4 Research methodology

The methodology used to address the issues mentioned in section 1.3, are discussed in this section.

1.4.1 Overview on SBCs

Different SBCs will be considered. Each SBC will then be characterised in terms of its cost, industrial application, processor(s) and its fulfilment to the existing specifications.

1.4.2 Control system specifications

The SBCs that meet the terms of the existing specifications will then be fully characterized in terms of performance, size, cost and availability. The extra functionality will then be discussed with the client to decide which is best for their application. After this, a more complete SBC specification can be drawn up and used to choose a specific SBC. The industrial standards as well as the tests done on the specific SBC will also be considered.

1.4.3 Skills development for SBC

Tutorials of the new programming language will be used as the introduction to the new language. The help files will then be used for further clarification. Online help and FAQ's will be used as well, if necessary.

1.4.4 Implementation/Integration

In order to improve the control, the system will be implemented on the existing flexible rotor AMB in the laboratory. The control algorithm will then be improved by adjusting variables and modifying methods. When this process is completed successfully, the SBC will be integrated with the new flywheel project for standalone operation.

1.4.5 Evaluation

With the use of measurements, the current dSPACE® system will be compared to the new SBC system. The measurements from the existing flexible rotor AMB project controlled by dSPACE® will be compared to the new measurements obtained from the SBC control method. The results of the

stability and sensitivity analyses are used to objectively evaluate and compare performance.

1.5 Thesis overview

The thesis constitutes of six chapters:

Chapter 2 contains the detailed literature study of all the aspects relevant to the successful completion of this project. The information given in this chapter will enable the designer to make an educated decision on aspects regarding the specification of the SBC.

The system requirements and single board computer (SBC) specification is done in the first part of chapter 3. The second section deals with the sourced SBC model.

Now that the SBC model is known, the system integration can be done. Chapter 4 includes the interface boards required to connect the SBC to the flywheel energy storage system (FESS). Thereafter the firmware designs for the SBC as well the software for the graphical user interface (GUI) on the PC, is done.

Chapter 5 includes the system verification. It consists of four tests; firstly the performance of the AMB control algorithm, secondly the stability tests, thirdly the sensitivity analyses and lastly the control of the permanent magnet synchronous machine. Some conclusions regarding the SBC, will also be discussed here.

Finally, chapter 6 gives an overall conclusion on the control performance of the SBC. This will be followed by some recommendations to improve this project and future work to be done.

Appendix A presents information about physical communication medium types e.g. fieldbus and profibus. This information is used to make a decision on a communication medium for industrial use. A system specification for the single board computer is added in Appendix A and Appendix C contains the firmware- and software code as well as the information used in this project, presented on a DVD, included in this thesis.

Chapter 1 started off with background on AMBs, their operating principles and requirements. It was followed by the current control system and a problem statement. The issues to be addressed and the research methodology were also presented and lastly an overview of the thesis was given.

2
Chapter

Literature study

To be able to specify a single board computer for industrial applications, a better understanding of industrial control architectures is needed. This chapter starts off with industrial control architectures and then describes the main differences between industrial and commercial applications. This is followed by the network and communication aspects regarding industrial control. The term single board computer is then described and different processors used in single board computers are discussed. Furthermore the control strategies for active magnetic bearings, as well as the permanent magnet synchronous machine are explained.

2.1 Industrial control architectures

Control can be seen as many different actions. Switching a light on or off is a means of control. Three main components are needed for control to take place; an operator, control system and a substation. This is shown in figure 2-1. In the case of the light that is switched on or off; the light is the substation, the switch is the control system and the human is the operator.

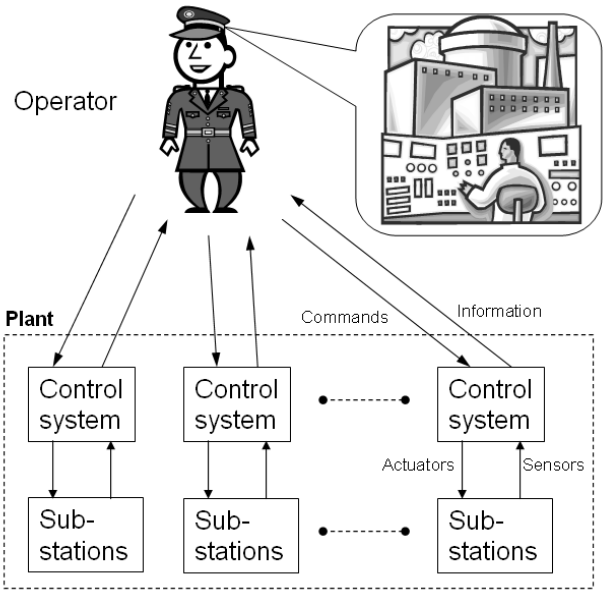


Figure 2-1: Simple control system

The control system can be a simple object like a switch or a more complicated and intelligent controller, like a programmable logic controller (PLC) or even a single board computer (SBC). The substation can also be a small plant or a smaller part of a bigger plant. Different control architectures exist between commercial and industrial applications. The main differences will be discussed in the section that follows.

2.1.1 Commercial vs. industrial architectures

Commercial and industrial applications need control by means of a computer, but due to the great difference in application, the controller requirements differ significantly. These differences are the following:

- Commercial applications make use of non-real time applications whilst industrial applications are highly focused on real time control. Waiting 3 to 4 seconds before an operation is executed, is unacceptable for industrial control.
- Commercial applications usually make use of a visual display unit only, (VDU) screen, keyboard, mouse and printer, while industrial applications have added digital input and output signals as well as analogue input and output signals, as shown in figure 2-2.

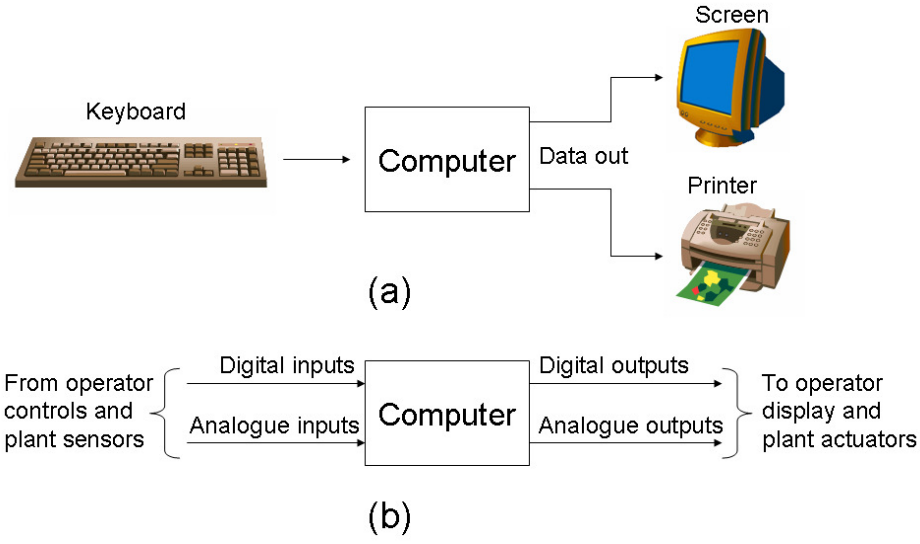


Figure 2-2: Difference between commercial and industrial applications a) Commercial computer, b) Industrial control computer

- The environmental conditions differ. Commercial applications normally operate in a controlled environment while industrial controllers are subjected to extreme temperature changes, high humidity as well as dust and dirt. Industrial computers should therefore be more robust, (mechanically and electrically) than commercial computers.
- Conventional commercial computers are connected by means of a switched Ethernet connection and are mainly used for file and printer sharing. Industrial control makes use of a hierarchical structure as shown in figure 2-3. The computers at the top still make use of an Ethernet connection, but the lower part of the hierarchy makes use of a fieldbus. The top part of the hierarchy is mostly for data/information transfer. The lower the level in the hierarchy, the higher the amount of connections and control orientation.

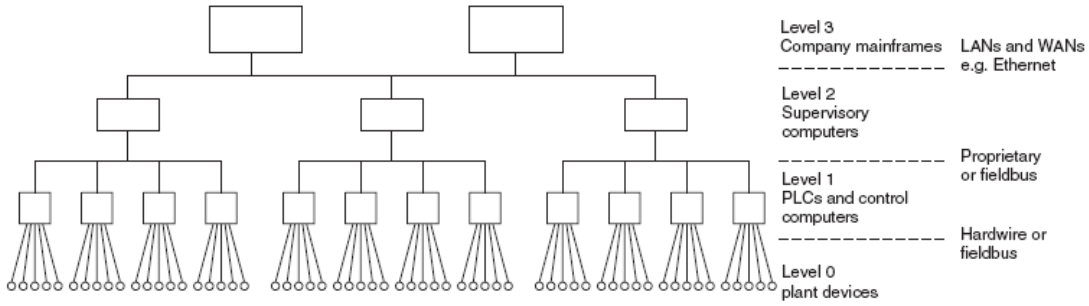


Figure 2-3: A simple communication hierarchy [18]

Before looking at the communication mediums, a better understanding of the open systems interconnection (OSI) model is needed. This would give an understanding of which procedure is needed to send data from one application to another.

2.1.2 OSI model

The International Standards Organization (ISO) developed the OSI model with reason. The main objectives of the (OSI) model are: [19]

- Allow manufacturers of different systems to interconnect their equipment through standard interfaces.
- Allow software and hardware to integrate well and be portable on differing systems.
- Create a model which all the countries of the world use.

The OSI model consists of seven layers. Each layer at the transmission has a direct relationship with the same layer at the receiving end [17]. Data passes from the top layer of the sender in system 1 to the bottom and then up from the bottom layer to the top on the recipient in system 2 [19]. The data flow from the sender to the recipient is shown in figure 2-4.

The functions of each layer are summarized in table 2-1: [17], [19]

Table 2-1: OSI layer functions

Layer	Function
Application	This links the user program to the communication process and determines what functions are required.
Presentation	This layer changes the data to a standard format. It uses a set of translations that allows the data to be properly interpreted. It can also add data encryption for security purposes.
Session	It provides the function to set-up, maintain, and close a session. It should also re-establish communication if there are problems with the link.
Transport	This layer provides error detection and correction for the whole message and controls message flow to prevent overrun at the receiver. It also allows the transmission of multiple streams from a single computer.
Network	It routes data frames through a network. It may split data for transmission and re-assemble it upon reception.
Data link	This layer ensures transmitted bits are received in a reliable way. This is achieved by adding extra bits such as start-, stop-, and error detection/correction bits. It also ensures integrity and controls the access to the network ensuring that multiple nodes do not attempt to access a common communication channel at the same time.
Physical	It does the coding and physical transmission of the message. Requirements such as transmission speed, voltage levels, connector types and cabling are covered.

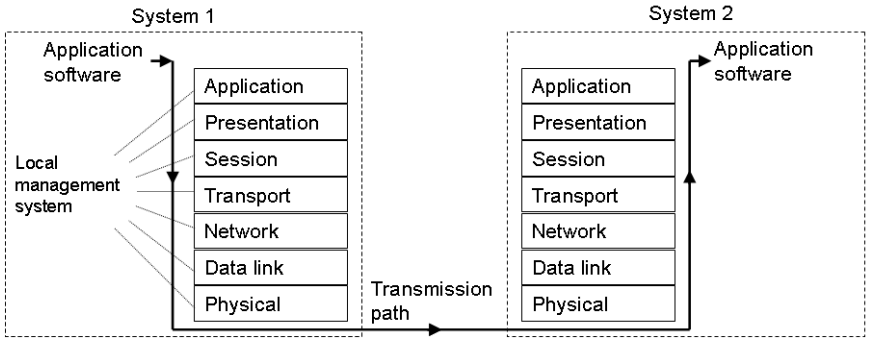


Figure 2-4: The OSI model

Communication through the seven layers can be explained in terms of placing an order by telephone. This will be explained for both sides of the OSI model.

1. Physical link layer – the phone is lifted and connected to the network. A dial tone is heard.
2. Error detection and control – a clear dial tone is heard with no noise.
3. Network layer – the number is dialled, area code, etc. Phone rings on the other side
4. Transport layer – the telephone is lifted at the receiving end. A switchboard picks up and requests you to hold. A minute later it puts you through to an operator. The operator asks if he/she can help.
5. Session layer – you give the order with the order and account number. The operator takes note in case the call is broken prematurely.
6. Presentation layer – you confirm where your order number came from.
7. Application layer – you give the exact items and quantity required as stipulated on the order. The operator confirms and completes the order.

At any stage, the lower layers can interact. A burst of noise on the line, for example, will cause the transport layer to ask for a repeat of the last message.

It can be seen that layers (1) to (4) are concerned with the communication and layers (5) to (7) are concerned with processing functions for the particular applications [17].

The physical layer can have many different standards. The main focus here is the industrial network; also commonly known as a fieldbus.

2.1.3 Fieldbus types

An industrial network is known as a fieldbus. Normally serial communication is employed due to cabling cost. Compared to parallel communication, serial communication has the disadvantage of lower speed, noise immunity, safety and program comprehensibility [17]. The speed of serial lines is slower than parallel communication by the factor of the number of parallel lines. Serial lines usually make use of low voltages in the order of 10 V, hence the noise immunity problem. The lower in the communication hierarchy shown in figure 2-3, the stricter the communication is on time limits. This is also known as a real-time network-based control system (NCS) or remote controller [20].

Remote controllers have three advantages:

- It reduces cabling costs. Long runs between the remote controller and the control room will only need the communication cables, not the control cables e.g. the sensor- and actuator cables.
- It allows complete units to be built, wired and tested prior to delivery and installation.
- Makes fault detection easier.

The IEC61158-2 standard of the International Electrotechnical Commission (IEC) is used to assist the interconnection of automation system components by fieldbus networks. According to the IEC61158-2, there are eight different physical layers used. The different layer types, a description of how they operate, and possible physical mediums and the maximum transmission speed in bits per second (bps) are presented in table A-1 in appendix A.

Each fieldbus type uses its own physical communication medium. The advantages and disadvantages of the mediums will be compared in the following section.

2.1.4 Communication mediums

The communication medium is the physical piece of wire and the connectors used with it. The wire can vary from a twisted pair cable, coaxial cable to even a fibre optical cable. Connectors can be anything from a DB9 connector, RJ45 to an SMA connector. The main considerations of choosing a communication medium is the transmission speed as well as the number of

allowable stations. Table A-1 summarises the physical mediums used by fieldbus. The medium IEC1158-2 has the worst data rate for a very short maximum distance and will therefore not be considered. Table 2-2 summarises the rest of these mediums and some additional considerations.

Table 2-2: Communication mediums [17], [19], [22], [23].

	Data rates	Maximum distance	No of drivers	No of receivers	No of conductors per signal
RS232	20 kbps	15 m	1	1	1
RS423	300 kbps	1200 m	1	10	2
RS422	10 Mbps	1200 m	1	10	2
RS485	10 Mbps	1200 m	32	32	2
USB	480 Mbps	5 m	127	127	2
Ethernet	100 Mbps	500 m	1024	1024	2
IEEE 1394 "Firewire"	400 Mbps	4.5 m	63	63	2

It is clear from the above table that RS232 will not be sufficient for industrial control applications due to the limit in drivers/receivers as well as the low data rate. RS422, -423, -485 is an improvement on RS232. Greater data rates and improved range is obtained with an increase in the number of conductors needed. RS485 is capable of up to 32 drivers and receivers without a repeater. This would be sufficient for a small plant or a section of a large plant. USB and Firewire have very high data rates, but very short cable length limits. This will work for the computers higher up in the hierarchy, but not for the controllers in the field. The Ethernet protocol is the most suitable for fieldbus applications. It allows for adequate cable lengths, drivers/receivers and data rates.

More detail about the RS485 and Ethernet communication mediums are needed before an informed decision can be made for industrial application. Some advantages and disadvantages of these two mediums will be highlighted in the following section.

2.1.4.1 RS485

RS485 makes use of a balanced differential circuit as shown in figure 2-5. No direct common ground or return signal is sent over the transmission line, but both sides should be connected to a common ground to prevent too large

potential differences between the transmitter and receiver. This is done to reduce the noise at the receiver.

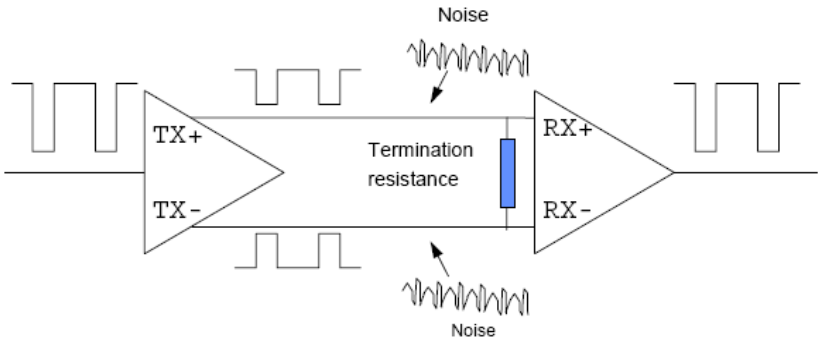


Figure 2-5: Balanced differential RS 485 [19]

There is no connector, cable or protocol specification for RS485. The IEC61158-2 standard for fieldbuses should be used for the specifications. RS485 is used as universal asynchronous receiver/transmitter (UART) for low speed communication on aircrafts, large sound systems, building automation as a simple bus and for video surveillance cameras. The advantages and disadvantages of using RS485 are as follows:

Advantages of RS485

- Bi-directional half-duplex operation
- Multipoint applications
- Low cost
- Noise immunity

Disadvantages of RS485

- Only one driver can be active at any one time
- More cabling required than RS232
- Not as common

RS485 is used in industrial fieldbuses because of its noise immunity and low cost.

2.1.4.2 Ethernet

Ethernet was first used only in offices for high-speed data transfer between computers, but recently it penetrated the industrial control market due to added advantages. These advantages and disadvantages include:

Advantages of Ethernet [24]:

- Widespread usage
- Lower cost due to higher volumes
- De facto Standard (already used higher up in the hierarchy)
- Most computers include Ethernet support.

Disadvantages of Ethernet [24]:

- It is nondeterministic (it is based on collision detection and avoidance that slows down the response of the network as traffic increases)
- Data collisions affect the bandwidth (all traffic is seen at every node)
- Lack of industrial-grade components.

The fact that Ethernet is nondeterministic makes it less useful lower down in the control hierarchy. For real-time control, the time delay for the data transfer should be fixed and known where possible. This is not the case with Ethernet. Although Ethernet is collision detection based, it is still much faster than RS485.

In 1999 Ethernet was used by about 50%, Profibus 26%, ControlNet 14% and Interbus-S and Foundation fieldbus 7% of the industry. The total is more than 100% due to firms that supports more than one bus. As seen in figure 2-6, Ethernet is becoming the new industrial standard for fieldbus control. It is mostly used higher up in the control hierarchy, but recently found its use lower in the hierarchy.

This is due to faster Ethernet rates of up to 1000 Mbps and more intelligent routers. The routers now know where the data should go and thus all nodes on the network do not see the data anymore.

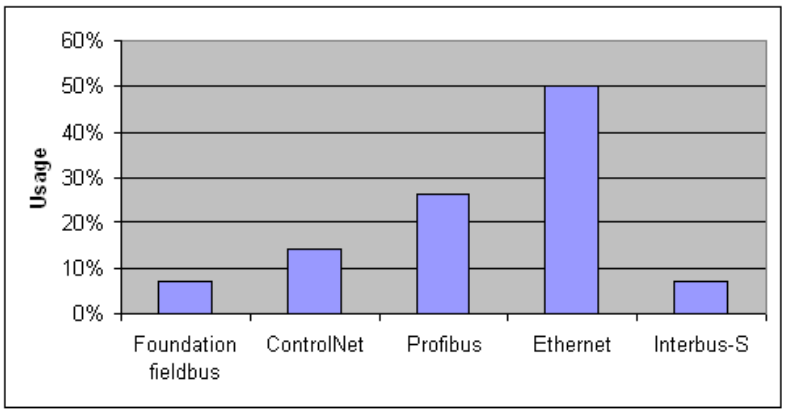


Figure 2-6: Fieldbus type usages (1999) [25]

The 2 main communication mediums to be considered are Ethernet and RS485. To continue with the research on industrial control on AMBs, the term single board computer (SBC) should be understood so that a well knowledge base decision can be made when specifying the board.

2.2 Single board computers

A single board computer (SBC) is difficult to define. Any printed circuit board that contains a digital controller, memory and input/output peripherals can be defined as a single board computer. Even your normal desktop computer is a single board computer. The main difference between consumer computers and single board computers is the fact that single board computers are much more rugged for industrial use [35]. Our interest is more related to industrial embedded controllers. Some examples of these industrial SBCs can be seen in figure 2-7.

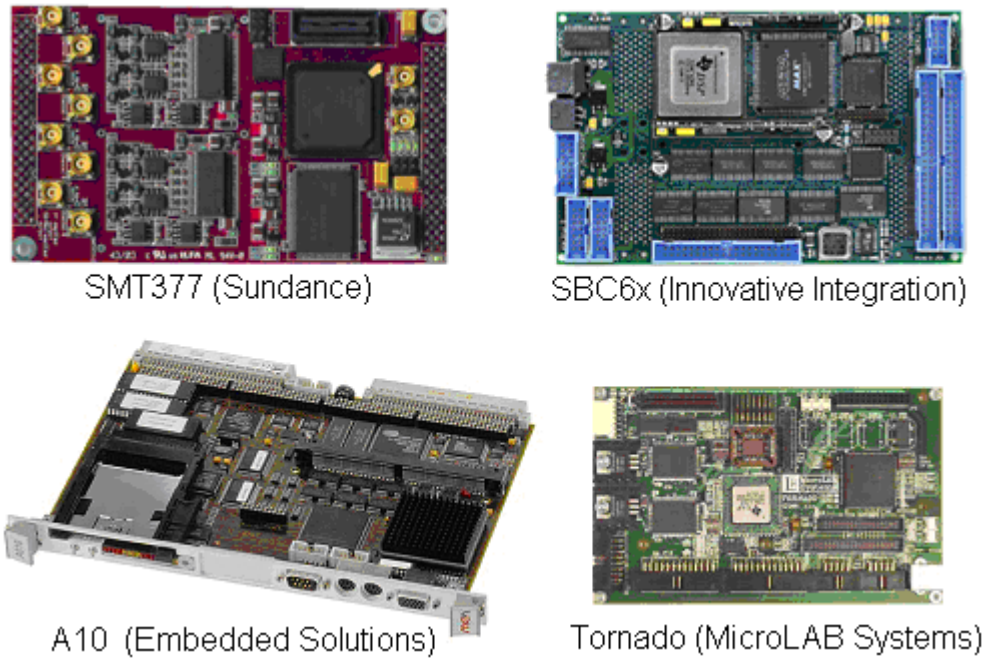


Figure 2-7: Industrial Single board computers

These SBCs normally make use of a field programmable gate array (FPGA) or a digital signal processor (DSP). These two processors will be discussed in the following section.

2.2.1 FPGA or DSP

Which is the best, FPGA or DSP? There is no true answer. Each has its place in the industry, and performs better in different situations. The features of each processor are listed below. [25]

DSP

- Operate at high rates, but is limited to few operations at a time
- Excellent floating point operations for increased accuracy/dynamic range (complex calculations, matrix inverse/division)
- Good for back-end use (Main processor which deals with all the commands to and from the co-processors)
- Supports C/assembly programming [26]
- Can make use of large data sets due to optimization for the use of external memory
- Easy to re-use processing units
- Branching and decision making is easy
- Major context switch, DSP branching (It is easy for a DSP to jump from the code of one application to the code of another application without restarting)
- System starts as block diagrams. It is difficult to change the block diagrams into C/assembly language where simultaneous operations should take place at once.

FPGA

- Operate at lower rates, but almost unlimited simultaneous operations
- Excellent fixed point operations and parallelism (filtering)
- Useful as front-end processor (co-processor)
- "Sea of gates" Use VHDL to create own multipliers, registers, adders [26]
- Limited internal storage and thus can make use of small data sets only.
- Difficult to re-use processing units due to the fact that branching is not possible. Therefore the unit has to be duplicated.
- Branching and decision making is difficult
- Major context switch, FPGA large reconfiguration (takes time)

- System start as block diagrams, easy to convert to FPGA
- Maintain high rates of I/O

To decide which processor to use, the following questions can be asked [26]:

1. What is the sampling rate of this part of the system? If it is more than a few MHz, an FPGA is the natural choice.
2. Is your system already coded in C? If so, a DSP may implement it directly. It may not be the highest performance solution, but it will be quick to develop.
3. What is the data rate of the system? If it is more than perhaps 20-30 MBps, then an FPGA will handle it better.
4. How many conditional operations are there? If there is none, an FPGA is perfect. If there are many, a software implementation may be better.
5. Does your system use floating point? If so, this is a factor in favour of the programmable DSP. None of the Xilinx cores support floating point today, although you can construct your own.
6. Are libraries available for what you want to do? Both DSP & FPGA offer libraries for basic building blocks like FIRs or FFTs. However, more complex components may not be available, and this could sway your decision to one approach or the other.

To conclude the comparison between the FPGA and DSP, the main criteria are summarized in table 2-3.

Table 2-3: Summary of FPGA vs. DSP performance [25], [26]

Performance	DSP	FPGA
Programmability	√	
Parallelism		√
High I/O data rates		√
Development time	√	
Available skills	√	
Cost	√	
Power consumption	√	
Floating point	√	
Fixed point		√

Included in the performance of the processors, is the aspect of choosing between a floating point processor and a fixed point processor. Again both has its advantages and disadvantages. These will be discussed in section 2.2.2.

2.2.2 Floating point vs. fixed point

The main difference between the floating point and fixed point processors is the numeric representation. Fixed point processors can only do integer arithmetic, while the floating point processor can do integer and real arithmetic.

When the first digital processors were developed, it was only available as fixed point processors. To add the floating point capability, the physical package had to be enlarged, which increased the cost of the chip. Due to this, the fixed point processor was favoured for high volume applications. This further reduced the cost of the fixed point processor.

Due to improved technology, the floating point package is now the same size as the fixed point processor and there is a small difference in price. Fixed point processors had to be programmed using assembly language, while the floating point processors could be programmed using C language. This favoured the floating point processors above the fixed point processors.

Today fixed point and floating point processors could be programmed using the same C compiler. The main consideration now is the processing speed, accuracy and dynamic range.

For the same chip, fixed point would have a faster processing speed, but the floating point processor would have an increased accuracy and dynamic range. The processor should be chosen purely on the application and not on the cost or ease of programming. If the application requires high accuracy, choose a floating point processor. If the application requires fast integer arithmetic, choose a fixed point processor.

Table 2-3 summarises some of the DSP's available from Texas Instruments. The format and accuracy of each are compared.

Table 2-4: Texas Instruments DSPs [12]

		Word width		
TI DSP(s)	**Format**	**Signal I/O (bits)**	**Coefficient (bits)**	**Intermediate result (bits)**
C25x™	fixed	16	16	40
C5x™/C62x™	fixed	16	16	40
C64x™	fixed	8/16/32	16	40
C3x™	floating	24 (mantissa)	24	32
C67x™(Single precision)	floating	24 (mantissa)	24	24/53
C67x™ (Double precision)	floating	53	53	53

When fixed point multiplication occurs, the result is equal to the sum of the signal width and the coefficient width plus an additional overflow width. In the C25x™ processor, the overflow width is 8 bits, therefore the intermediate result is 40 bits (16+16+8 = 40). In comparison with fixed point, floating point only needs the signal width (signal- and coefficient width should be the same) and the overflow width. This is because of the scientific format of the floating point processor. For instance, with fixed point calculations, the representation of the calculation would be as follows:

$$\begin{aligned} 15_{10} &\qquad 1111_2 \\ \times 15_{10} &\qquad \times 1111_2 \\ = \underline{255}_{10} &\qquad = 11100001_2 \end{aligned}$$

With floating point calculations, the representation would be more like this:

$$\begin{aligned} & 1.50_{10} \times 10^1 \\ \times & 1.50_{10} \times 10^1 \\ = & \underline{2.55}_{10} \times 10^2 \end{aligned}$$

This causes the floating point calculation to use less width than a fixed point calculation. Thus floating point processors are the preferred processor, but if calculation speed is more important than accuracy, fixed point processors should be used.

All the hardware aspects, for example the communication medium and the processor for the control of the AMBs in an industrial application, have been

discussed. Now all that is left is the control of the AMBs as well as an introduction to the flywheel energy storage system (FESS). Section 2.3 deals with the AMB control, and section 2.4 with the FESS.

2.3 Active magnetic bearing control

An active magnetic bearing operates similar to a conventional bearing, except that there is no physical contact between the rotor and the electromagnets. AMBs have some advantages over conventional bearings [27]:

- no mechanical wear and friction
- low drag torque
- no lubrication
- low energy consumption
- higher circumferential speeds
- operation in severe environments

As described in chapter 1, AMBs consist of 5 main components, the controller, power amplifiers, electromagnets, and rotor and position sensors. These components are shown in figure 2-8.

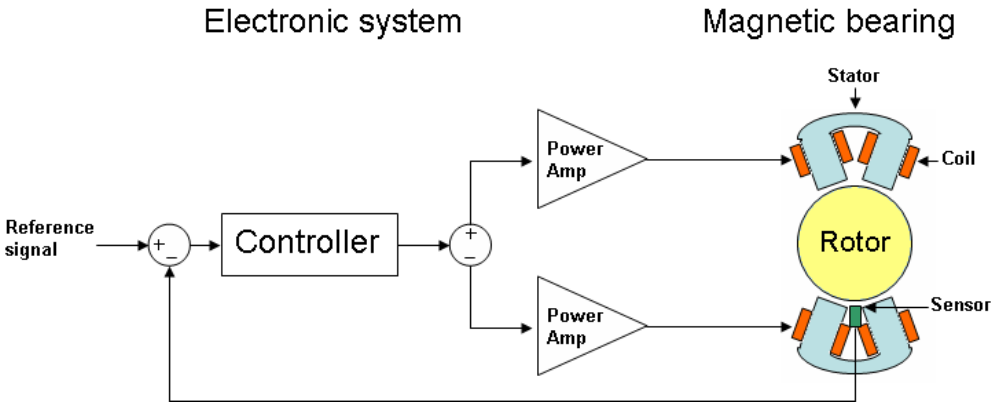


Figure 2-8: AMB system

The position offset of the rotor is obtained from the position sensor. This offset signal is then fed into the digital controller and an output signal is produced by the specific algorithm used. The current system makes use of the proportional-derivative (PD) algorithm. The output signal is amplified and applied to the electromagnets. In return, the electromagnets apply forces to the rotor, proportional to the output signal. These forces keep the rotor levitated. The control loop then restarts with the offset signal from the position sensor.

Currently the McTronX research group at the North-West University is conducting research into high speed flywheel energy storage systems (FESS). A common FESS will be described in more detail in section 2.4.

2.4 Flywheel energy storage system

The mechanical assembly of a high speed flywheel energy storage system (FESS) consists of 3 main components. The motor/generator, the rotor and the bearings as shown in figure 2.9. The rotor is suspended by two radial- and one axial active magnetic bearing. PD control is used to control the position of the rotor by means of the AMBs. The motor/generator is a 3-phase permanent magnet synchronous motor (PMSM). The permanent magnets are glued to the rotor, while the coils are wound through a bobbin situated in the FESS housing. A 3-phase bridge is used to drive the motor. This bridge requires two pulse width modulated (PWM) signals per phase. The second PWM signal should be the inverse of the first signal.

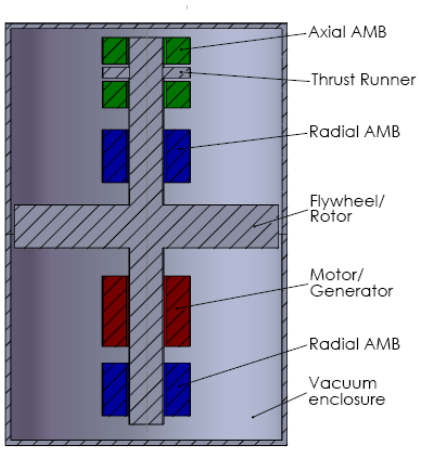


Figure 2-9: Flywheel energy storage system [30]

The PMSM in motor mode is used to spin the rotor up to the desired rotational speed. It is then kept at that speed until power failure. The PMSM then changes to generator mode and the kinetic energy stored in the rotor is then used to drive the generator.

The requirements for the control component for a basic FESS are normally at least 5 position sensors, 2 radial- and 1 axial AMB, 6 PWM signals for the PMSM and a few temperature sensors for system monitoring and protection. (The current FESS used by the McTronX research group will be discussed in chapter 3) Most of these signals are analogue signals that are converted to

digital signals for the controller. The analogue to digital sample theorem will be discussed in section 2.4.1. This is to motivate the sampling rate required by the PD control algorithm for the AMBs. Furthermore, the control of the PMSM will be discussed.

2.4.1 Analogue sampling theorem

Analogue to digital conversion for digital controllers, samples a continuous signal into a discrete time signal. The controller then only sees the sampled values. If the sampling rate is too low, information about the continuous signal is lost. If the sampling frequency is less than twice the sampled signal's frequency, aliasing occurs. Aliasing is when the controller observes a lower frequency than what the real continuous signal is. The effect of aliasing can be seen in figure 2-10. According to Shannon's sampling theorem, the sampling rate of the controller should at least be double the bandwidth of the signal [18]. Normally a sampling rate between 5-10 times the bandwidth of the signal, is a good choice [18], [17].

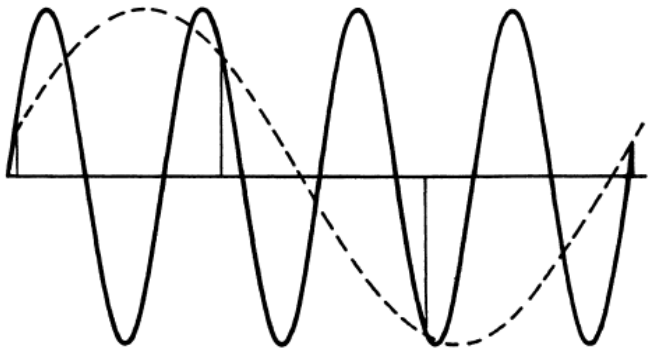


Figure 2-10: Aliasing [18]

The bandwidth of the FESS is 2 kHz; therefore a sample frequency of 10 ksps would be sufficient.

To control the AMBs, the analogue signal is converted to a digital value and used by the processor to calculate an output value for the power amplifiers which is connected to the AMB coils. The processor makes use of proportional-derivative (PD) control, to control the AMBs. Section 2.4.2 discusses the PD control algorithm.

2.4.2 Proportional-derivative control

PD control is known to be sufficient for many high speed AMB applications [14]. Proportional-derivative (PD) control is a linear control strategy that consists of only two parts, the proportional part and the derivative part. This algorithm makes use of an error signal, the error between the reference position and the actual position. This is given by (2.1):

$$e(t) = x_{ref}(t) - x_{actual}(t) \tag{2.1}$$

The output of the PD algorithm is then indicated by (2.2):

$$F(t) = K_p \times e(t) + K_d \times \frac{d}{dt} e(t) \tag{2.2}$$

where K_p is the proportional constant, K_d the derivative constant and t the instantaneous time.

PD control is very useful [2]:

- the stiffness of the bearing is determined in the proportional part
- the damping of the bearing is determined in the derivative part

Linear control operates well when the rotor is in a nominal position, but lacks performance in four other areas [5]:

- Large position variation is a problem. It is difficult for linear control to lift the stator from rest position, whereas with nonlinear control this is not a problem.
- Sinusoidal references cause a large delay between the reference and the real position.
- The change in stable loop gain for linear control is much less than it is for nonlinear control.
- Current consumption is more due to the required bias current needed for linear control. The bearing stiffness is related to the magnitude of the bias current.

In an AMB system, the position should have a very small variation and a constant reference. The loop gain of an AMB is also kept constant, therefore only the current consumption affects an AMB system. Although PD control has these disadvantages, it is adequate for rotor position control in an AMB system.

2.4.3 Permanent magnet synchronous motor (PMSM) control

Open-loop control is the simplest sensor-less control scheme used on synchronous motors. This is very useful on high speed PMSMs, because no mechanical speed sensor is needed. The rotation speed of the rotor is always locked to the excitation frequency. There are three commonly used control schemes:

- Constant V/f control
- Vector control
- Hybrid voltage-vector control

Constant V/f control is the easiest to implement and allow very high operating speeds. The only disadvantage is the trail-and-error process used to determine an optimal V/f ratio. The constant V/f control is described by figure 2-11.

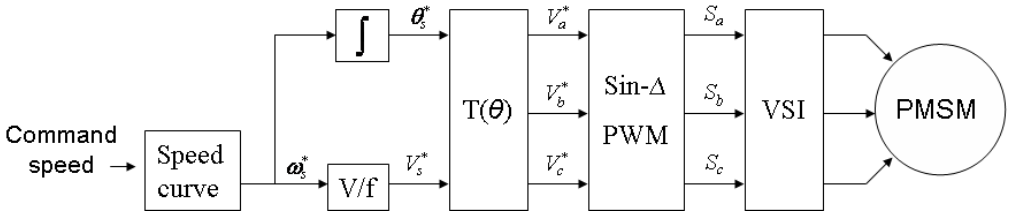


Figure 2-11: Constant V/f control [28]

The control scheme waits for a command speed given by the operator. It then uses the speed curve to ramp up or down the current speed to the desired speed. The instantaneous speed given by the speed curve is then used to obtain the rotational angle, as well as the maximum voltage supplied to the coils of the PMSM. To calculate the angle, (2.3) is used [28].

$$\theta_s^* = \int_0^t \omega_s^*(\tau) d\tau \tag{2.3}$$

$\theta_s{}^*$ is the instantaneous angle, $\omega_s{}^*$ is the rotational speed in rad/s and τ is the time.

To determine the maximum voltage $V_s{}^*$, the current (i_{as} and i_{bs}) in two of the PMSM coils is used. First the magnitude of the instantaneous current i_s is calculated using (2.4), then the magnitude of the instantaneous power factor angle ($i_s \cos \phi$) using (2.5). The maximum voltage can then be calculated using (2.6) [32].

$$i_s = \sqrt{\frac{1}{3}(i_{as} + 2i_{bs})^2 + (i_{as})^2} \tag{2.4}$$

$$i_s \cos\phi = \frac{2}{3}\left[i_{as} \cos\theta_s^* + i_{bs} \cos(\theta_s^* - \frac{2\pi}{3}) - (i_{as} + i_{bs}) \cos(\theta_s^* + \frac{2\pi}{3}) \right] \tag{2.5}$$

$$V_s^* = (i_s \cos\phi) r_s + \sqrt{(\omega_s^* \lambda_m)^2 + (i_s \cos\phi)^2 r_s^2 - i_s^2 r_s^2} \tag{2.6}$$

λ_m is the rotor permanent magnet flux and r_s is the stator winding resistance per phase.

The instantaneous voltage of each phase of the PMSM can then be calculated making use of (2.7) [28]. The phases are spaced 120° apart.

$$\begin{aligned} v_a^* &= V_s^* \sin\theta_s^* \\ v_b^* &= V_s^* \sin(\theta_s^* + \frac{2\pi}{3}) \\ v_c^* &= V_s^* \sin(\theta_s^* - \frac{2\pi}{3}) \end{aligned} \tag{2.7}$$

The instantaneous voltages are then converted to pulse-width modulated (PWM) signals with the use of a delta-sigma function.

All the relevant literature regarding the industrial control of AMBs has been discussed. This included the use of an SBC with a PD control algorithm to control the AMBs and an open-loop voltage over frequency control for the PMSM. This information will be used in chapter 3 and 4 to assist in specifying an SBC and doing the software integration. In chapter 3, the SBC to be purchased, will be specified.

3
Chapter

Single board computer

This chapter begins with a more detailed explanation of the flywheel energy storage system (FESS) which was developed by the McTronX research group. The control requirements for the single board computer (SBC) are extracted from the system analyses. The minimum performance requirements of the SBC are determined further on in this chapter, which is followed by the sourcing of the board thereafter.

3.1 System requirements

In Chapter 2, the standard architecture of an AMB was described. Figure 3-1 gives an overview of the functional units of the complete FESS.

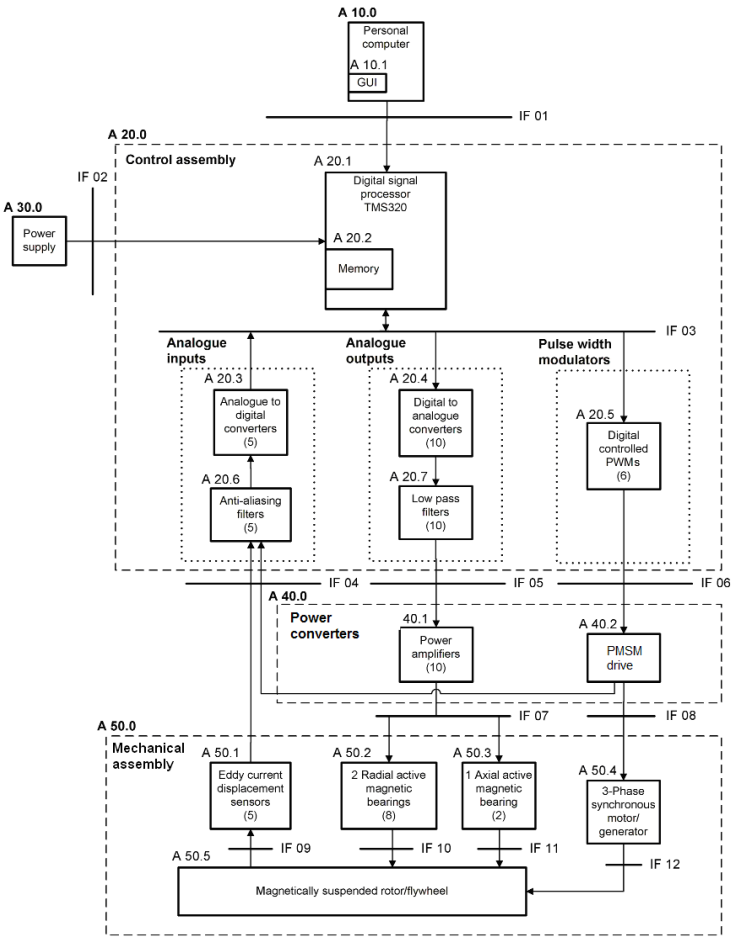


Figure 3-1: Architectural flow

The FESS consists of three main parts; the control assembly, power converters and the mechanical assembly which includes the rotor, AMBs, permanent magnet synchronous machine (PMSM) and the eddy current displacement sensors. This can be seen in figure 3-1. The system specification of architectural unit A20 is given in Appendix B. Between each architectural unit (A10-A50) there are interfaces (IF01-IF12). These interfaces are wires in the electronic components and a magnetic interface between the AMBs and rotor. In the current FESS, the McTronX research group makes use of a dSPACE® system for architectural unit A20. The proposed system will make use of a SBC to do the control on the FESS. Before a specific SBC can be chosen and sourced, a sub-system requirement is needed. The requirements of the SBC (A 20.0 in figure 3-1) will be included in the following section.

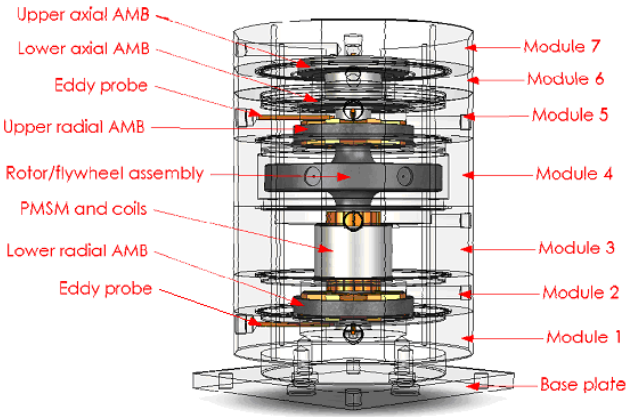


Figure 3-2: FESS model [31]

3.1.1 Input/output signals

To be able to specify a controller, all input and output signals have to be determined. All the input and output signals, digital and analogue, required by the system will be determined in this section.

As shown in figure 3-1 and 3-2, the system consists of two radial bearings and one axial bearing. Each radial bearing consists of four coils, two for each degree of freedom (x, y) and two position sensors, one for each degree of freedom. The axial bearing consists of two coils, one on both sides of the thrust runner, and one position sensor. This can be seen in figure 3-2. This adds up to 5 sensor inputs and 10 coil outputs.

The 10 coil outputs make use of power amplifiers (PAs) from Advanced Motion Control. These PAs has TTL fault outputs which are monitored. This adds 10 digital inputs.

For the system to operate under safe conditions, four resistance temperature detector (RTD) sensors have been added to monitor the temperature of the AMB coils. This is critical, because too high temperatures reduce the magnetic stiffness of the bearing, which could result in an unstable rotor that might damage the system.

The permanent magnet synchronous motor/generator (PMSM) consists of 3 phases. For each phase the voltage and current (LEM sensor) are measured. This adds 3 voltage sensors and 3 LEM sensors.

To drive the PMSM a 3-phase bridge is used. This bridge requires 6 pulse width modulated (PWM) signals, 1 signal to synchronise the switched mode supply (flyback) on the PMSM drive board with the analogue sampling, and 1 temperature switch to monitor the temperature status of the 3-phase bridge.

Three pickup coils are included in the system to obtain the rotational speed of the rotor. The system is operating in a vacuum, therefore 1 vacuum sensor and 2 relay outputs, (1 for the vacuum pump and 1 for the valve) is needed.

An additional relay is needed for the over-speed protection circuit. This is to ensure the safe operation of the FESS.

This adds up to 16 analogue inputs, 10 analogue outputs, 14 digital inputs and 10 digital outputs. When a controller is specified, this would be the minimum input/output signal requirements as shown in table 3-1.

Table 3-1: Controller signals

Analogue inputs	Analogue outputs	Digital inputs	Digital outputs
5 position sensors	10 PA (coils)	10 PA fault	6 PWM
4 RTD sensors		3 Pickup coils	3 Relays
3 LEM sensors		1 Temp switch	1 Flyback sync
3 Voltage sensors			
1 Vacuum sensor			
∑ = 16	**∑ = 10**	**∑ = 14**	**∑ = 10**

The digital input/output signal should all be 5 V transistor-transistor logic (TTL) levels. The analogue signals requirements will be determined by the eddy-current sensors and the power amplifiers. The other sensors (RTDs, voltage, vacuum and LEM) can be scaled to fit these specifications.

The CMSS 665 eddy-current sensor from SKF has a bandwidth of 10 kHz and output voltage ranges of -18 V to 0 V. These voltages will be limited to between -12 V and 0 V by the interface board, which includes interfaces IF04, IF05 and IF06. The 12A8 servo amplifier from Advanced Motion Control has a bandwidth of 2.5 kHz and input voltage range of ±15 V.

There is no specific requirement for the connectors of any of the input/output signals.

The final requirement to be specified is the communication to the computer. With the research done in chapter 2, Ethernet is the most commonly used in commercial and industrial networks. Therefore it is required that the SBC should have an Ethernet 10/100 Mbps base connection.

3.1.2 Signal processing requirements

In Chapter 2 the issue of fixed- and floating point operation has been discussed. It was concluded that the processor should be able to do floating point operations. The processing power needed will be estimated by means of million floating point operations per second (MFLOPS).

The current PD algorithm will be analyzed to predict a performance requirement. Figure 3-3 will be converted to pseudo code, which will then be used to calculate the MFLOPS.

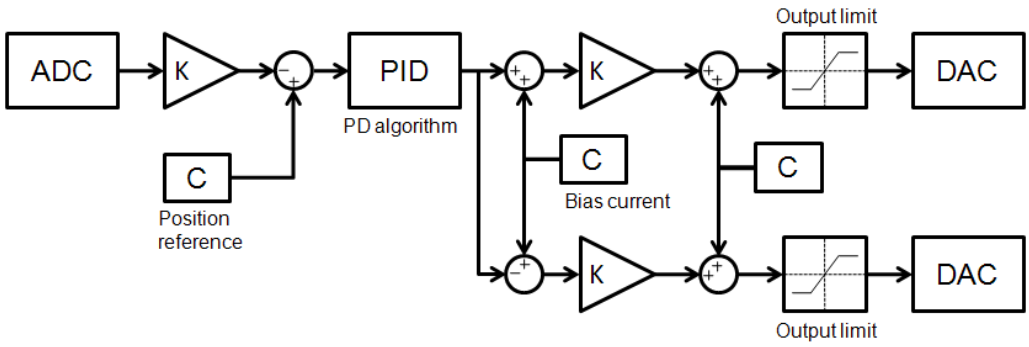


Figure 3-3: AMB control algorithm

The required cycles for the PD control algorithm are estimated at 58 cycles as shown in table 3-2. This is only for one degree of freedom (DOF) (one AMB).

There are five DOF, two for each radial AMB and one for the axial AMB. Currently the ADCs and DACs have a sampling frequency of 20 kHz. This would require at least 58x5x20000 = 5.8 MFLOPS.

For best performance of the AMBs, the delay between the read of the ADC, calculation of the PD control and the write of the DAC, should be as short as possible. A large delay could result in an unstable system. The higher the MFLOPS of a controller, the lower the delay would be between the input and output signal and the more stable the AMB control would be. It was decided for this project that the delay should not exceed one tenth of the sampling period. To compensate for this, the MFLOPS is multiplied by 10. This increases the minimum requirements to 58 MFLOPS.

Table 3-2: Control requirements for the PD algorithm [29]

Instruction	Mnemonic	No. of executions	Cycles	
Read the ADC	IN	1	2	
Multiply with scaling factor	MAC	1	3	
Subtract reference	SUB	1	2	
Calculate P	SUB	1	2	
Calculate D	SUB	1	2	
	MAC	1	3	
Multiply Kp	MAC	1	3	
Multiply Kd	MAC	1	3	
Add result	ADD	1	2	
Add/subtract bias constant	ADD	1	2	
	SUB	1	2	
Multiply with scaling factor	MAC	2	6	
Add amplifier linearity	ADD	2	4	
Check range limit	BCND	2	8	TRUE
		2	4	FALSE
Set limit	LAR	2	4	
Write to DAC	OUT	2	6	
		Total	58	

This is only for the levitation of the rotor, which is the most important function of the controller. The MFLOPS for the PMSM is not calculated due to the fact that the calculation intensity for a square root, sinus and co sinus depends on the processor as well as the libraries that are used to do this. The MFLOPS estimation of the PMSM control would therefore be very inaccurate. The control of the PMSM is of lower priority and would easily execute in the remaining nine tenths of the sampling period as shown in figure 3-4.

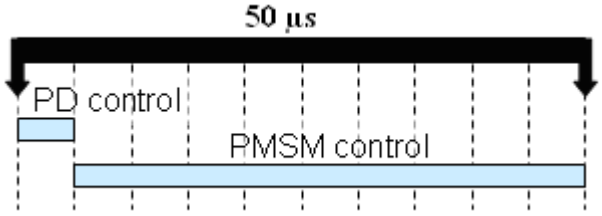


Figure 3-4: Control algorithm distribution

The TMS320C6000 series has 2 fixed-point arithmetic logic units (ALUs), 4 fixed- or floating-point ALUs and 2 multipliers. The ALU is the unit within a controller that does the calculations and decision making. This will allow up to 8 x 32 bit operations per cycle. Due to the ability of multi operations per cycle, the MFLOPS required by the PD algorithm, can easily be met.

3.1.3 Memory

No data logging will be done, meaning no physical data will be stored on the controller, except for the control algorithms, PD- and position reference constants. All data will be sent to the monitoring computer by means of an Ethernet connection. Therefore a programming memory of 64 kB will be sufficient.

3.1.4 Other

Other requirements include the physical and environmental specifications as well as the interfacing to the FESS. This SBC should comply with the industrial standards.

3.1.4.1 Physical

The only physical requirement is the size. The size of the SBC should not exceed the double Euro size (6U). This is 460 mm (width) x 330 mm (height) x 210 mm (depth).

3.1.4.2 Environmental

The SBC will be situated in a controlled enclosure with regulated temperature and humidity. The SBC should be able to operate in temperatures ranging from 0℃ to 73℃.

3.1.4.3 Interfacing

An interface board should be developed to connect the SBC to the current FESS. This interface board should not replace the existing interface board between the dSPACE® and FESS, but rather allow the FESS to be controlled by any one of the control systems.

3.2 Board sourcing

Possible SBCs, adequate for the application, were narrowed down to six. These six boards were further evaluated in terms of analogue I/O, digital I/O, controller precision, memory, size and temperature.

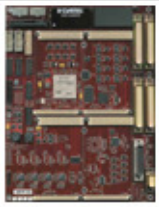
3.2.1 Existing models

The six SBCs have been summarised in table 3-3. All the specifications of the various boards which do not comply with the required specifications were highlighted in table 3.3. The SBC6713e and Lpdas are the only ones that fully comply with the specifications; the other four have slight deviations from the required specifications given in the first column.

The SMT148 and the Tornado TE6203 has only 8 analogue input and 8 analogue output channels. The required analogue channels are 16 for inputs and 10 for outputs. In contrast to that, the bandwidth of the ACR9000 is too low. The A10 almost comply with the specifications, the only drawback is the fixed point processor and the maximum voltage range of the analogue I/O.

The main decision is between the SBC6713e and the Lpdas. Both have very good properties, but the main difference is the size and the processing power.

Table 3-3: Available SBCs that comply with the given specifications

Criteria	Required	SMT148	SBC6713e	Lpdas	ACR9000	Tornado TE6203	A10, M35N, M62
							
Analogue input	# ≥ 16	# = 8	# = 16	# = 16	# = 32	# = 8	# = 16
	≥ 10 kHz	400 kHz	100 kHz	250 kHz	≈ 500 Hz	200 kHz	≈ 120 kHz
	± 12v	± 10v	±15v	±15v	± 10v	± 10v	± 10v
Analogue output	# ≥ 10	# = 8	# = 16	# = 16	# = 32	# = 8	# = 16
	≥ 2.5 kHz	200 kHz	100 kHz	250 kHz	≈ 500 Hz	200 kHz	≈ 60 kHz
	± 15V	± 10v	±15v	±15v	± 10v	± 10v	± 10v
Digital input	# ≥ 14	# = 24	# = 32	# = 56	# = 512	# = 8	# = 32
	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v
Digital output	# ≥ 10	# = 24	# = 32	# = 56	# = 512	# = 8	# = 32
	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v	TTL, 5v
Ethernet	Yes	Yes	Yes	Yes	Yes	No	Yes
Precision	floating-	fixed-	floating-	floating-	floating-	fixed-	fixed-
Processing power	≥ 96 MIPS\MFLOPS	(FPGA)	=1800 MIPS =1350 MFLOPS	=760MIPS	=150 MFLOPS	=2400 MIPS	≥100 MIPS
Program memory	≥ 64 kB	Not specified	32 MB flash	64 MB flash	8 MB flash	8 MB flash	8 MB flash
Size	(6U) 460mm(w) x 330mm(h) x 210mm(d)	40mm x 250mm x 200mm	(3U) 40mm x 160mm x 100mm	≤ (6U)	127mm x 267mm x 135mm	(3U) 40mm x 160mm x 100mm	(6U) 460mm x 330mm x 210mm
Temperature	0°C - 70°C	0°C - 85°C	0°C - 90°C	0°C - 70°C	Not specified	0°C - 90°C	0°C - 60°C

3.2.2 Chosen model

According to table 3-3, the SBC6713e and the Lpdas are the only two SBCs that fully comply with the given specifications. The SBC6713e has a much better processor performance and is half the size of the Lpdas. Therefore the SBC6713e will be used for the realization of the control of the FESS.

The following section will give more detail about the specific SBC6713e.

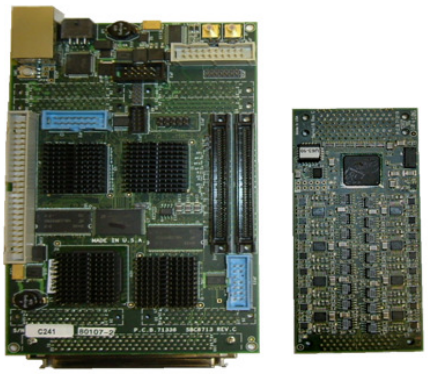


Figure 3-5: SBC6713e and SERVO16 analogue interface board

3.2.3 SBC6713e

The SBC6713e was purchased from Innovative Integration at a cost of R 54 492. It consists of a main processor (TMS320C6713), a co-processor (DM642) for the Ethernet control and a FPGA (Spartan-IIE, 600 Kgates). This is all in order to improve data flow and signal processing for real time applications.

The SERVO16 analogue card consists of 16 differential analogue inputs and 16 analogue outputs with simultaneous sampling. The card has a maximum sampling frequency of 100 kHz and filters on the input and output signals. The input has a 4-pole elliptic filter with the -3dB set at 30 kHz and the output has a single pole filter with the -3dB set at 50 kHz. Error correction offset and gain adjustment and decimation can be done within the FPGA to reduce processing time.

The rest of the detail is available in the product manuals presented in Appendix C.

Chapter 3 started off with a description on the current FESS. That information was used to extract the specifications for the SBC. Six SBCs were identified, and compared. One of them was chosen and sourced. Chapter 4 will deal with the interface board between the SBC, dSPACE® and the current FESS. Furthermore, the firmware development for the SBC as well as the software for the graphical user interface will be explained.

4
Chapter

System integration

The integration of the single board computer (SBC) and flywheel energy storage system (FESS) is discussed in this chapter. First the interfaces between the different hardware are described followed by the firmware for the SBC. The SBC firmware includes the active magnetic bearing (AMB) control as well as the permanent magnet synchronous machine (PMSM) control. Lastly, the software for the graphical user interface (GUI) is discussed.

4.1 Interface board

There are three hardware systems that should be interconnected. The dSPACE® controller as well as the single board computer (SBC) should be able to connect to the flywheel energy storage system (FESS).

For the dSPACE® controller to connect to the FESS, an interface board is required. Functions of the board include: voltage range limiting, protecting the analogue to digital converter (ADC) inputs of the controller and to do the signal conditioning required by the resistance temperature detectors (RTDs). The resistive values of the RTDs are converted to a voltage for measurement. The main purpose is however to connect the dSPACE® to the FESS and simultaneously protect the dSPACE® controller circuitry. This includes all digital inputs/outputs as well as the analogue inputs/outputs.

Another interface board was required to connect the SBC as well as the dSPACE® controller to the FESS. Figure 4-1 shows how these interface boards and controllers interconnect. Without the second interface board, dSPACE® would connect directly to the protection interface board as indicated in (a). The second interface board allows the SBC and the dSPACE® controller to connect to the FESS simultaneously as indicated in (b). The digital I/O of either the dSPACE® or the SBC can be used, but not simultaneously.

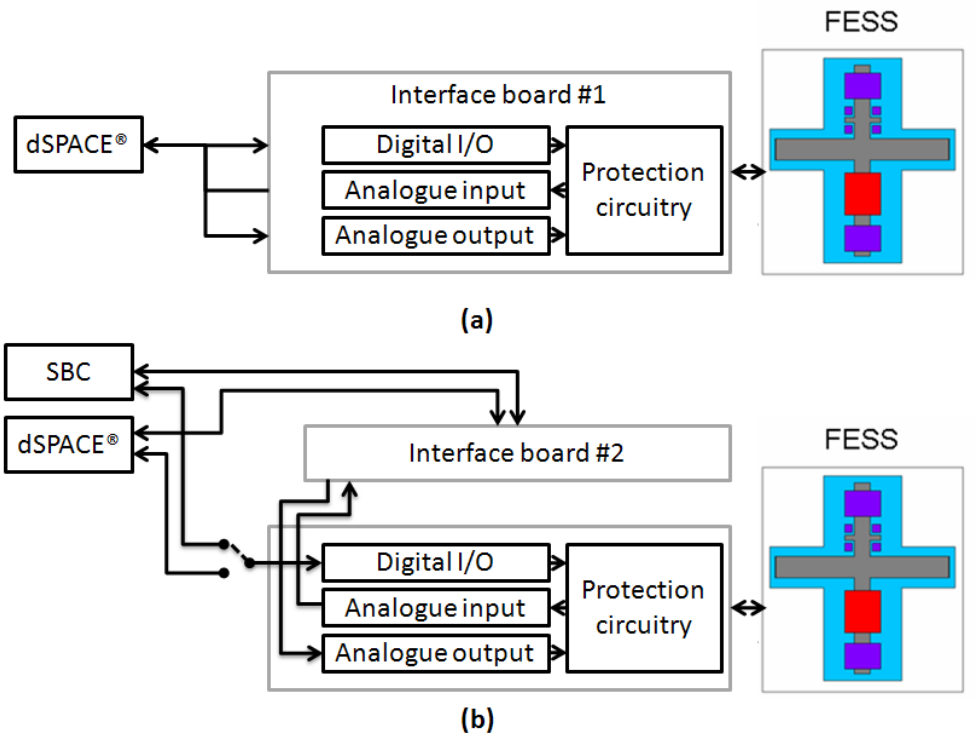


Figure 4-1: Interface board connections

Table 4-1 presents all the I/O control signals. The signals are divided into 3 main categories; the digital I/O, analogue input and the PA I/O. These three categories connect to the dSPACE® by means of a DB50 connector each and one MDR100 connector to the SBC. Table 4-1 shows each signal description with the pin number on the DB50 connector that correlates with the pin number on the MDR100 connector. This information was used to create the interface board for the SBC. The interface board for the SBC was designed to piggy-back onto the protection interface board. By doing this, the FESS could be controlled using the current dSPACE® system or the SBC system. The protection interface board between the FESS and dSPACE® is shown in figure 4-3.

One of the main aspects regarding the design of the interface boards was the grounding and shielding to minimise the noise in the system. To prevent the generation of ground loops, it was decided that all the ground signals will be grounded on the protection interface board. The protection interface board was divided into three zones namely: the analogue output, analogue sensing input and the digital I/O with a ground plane for each. All three planes will be connected to the chassis and all the shields are connected to the protection interface board as shown in figure 4-2.

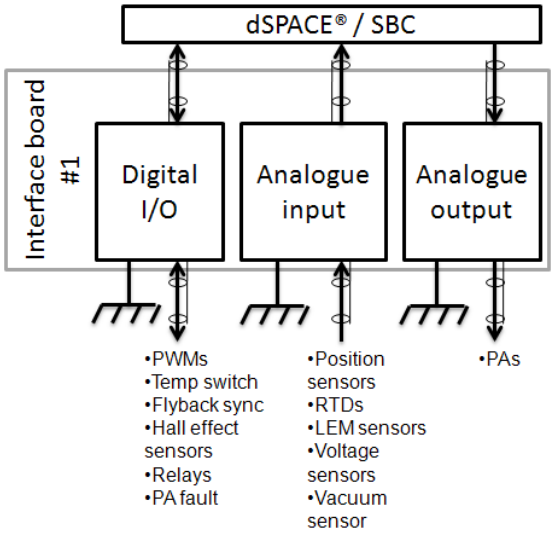


Figure 4-2: Gounding and shielding of FESS

With the second interface board in place, all the shielding is still kept the same except for the analogue input and analogue output grounds which come together.

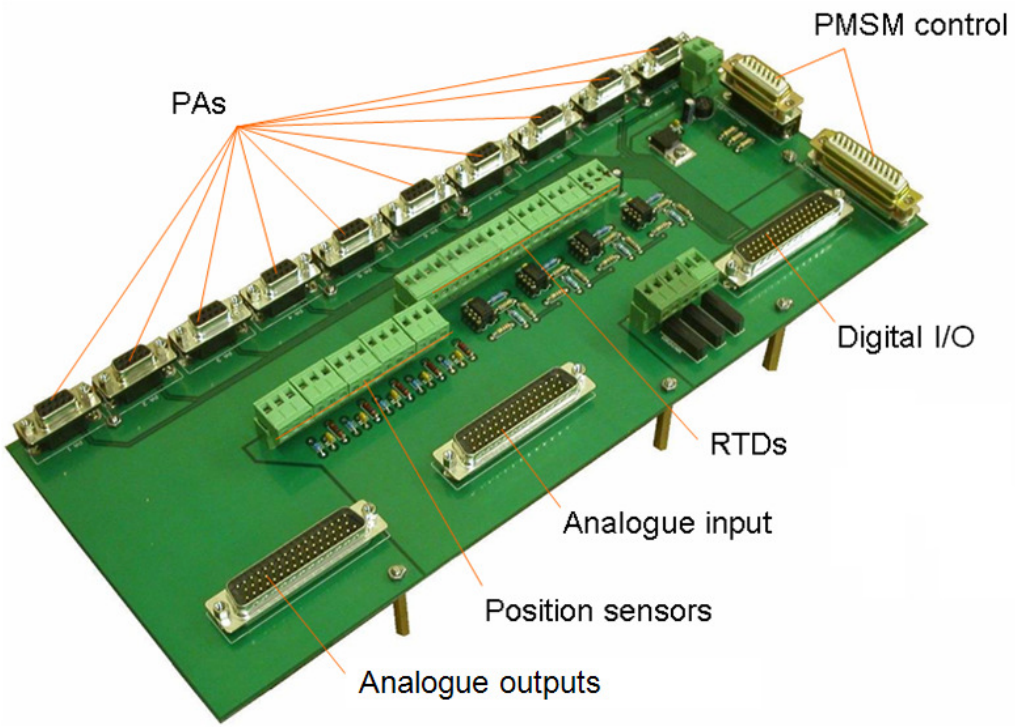


Figure 4-3: Protection interface board between dSPACE® and FESS

Table 4-1: Interface board connections

Digital IO			
Description	Use	DB50	IDC50
PWM 1+	3-Phase motor control	17	1
PWM 1 -		16	50
PWM 2+		15	2
PWM 2 -		14	50
PWM 3+		13	3
PWM 3 -		12	50
PWM 4+		11	4
PWM 4 -		10	50
PWM 5+		9	5
PWM 5 -		8	50
PWM 6+		7	6
PWM 6 -		6	50
Temp switch +	Power board protection	5	7
Temp switch -		4	50
5 v		NC	NC
Sync +	Flyback sync	2	8
Sync -		1	50
Pickup ground	Hall effect sensors	18	50
Pickup 1		34	9
Pickup 2		35	10
Pickup 3		36	11
Relay 3	Relays	50	12
Relay 2		49	13
Relay 1		48	14
Relay ground		47	50
PA 1, fault	Power amplifier fault	46	15
PA 2, fault		45	16
PA 3, fault		44	17
PA 4, fault		43	18
PA 5, fault		42	19
PA 6, fault		41	20
PA 7, fault		40	21
PA 8, fault		39	22
PA 9, fault		38	23
PA 10, fault		37	24

Analogue input			
Description	Use	DB50	MDR100
Input 1 +	Radial bearing position 1	41	32
Input 1 -		42	94
Input 2 +		43	81
Input 2 -		44	44
Input 3 +	Radial bearing position 2	45	31
Input 3 -		46	93
Input 4 +		47	80
Input 4 -		48	43
Input 5 +	Axial bearing	49	30
Input 5 -		50	92
Input 6 +	RTDs	34	79
Input 6 -		38	42
Input 7 +		35	29
Input 7 -		38	91
Input 8 +		36	78
Input 8 -		38	41
Input 9 +		37	36
Input 9 -		38	98
Input 10 +	Volt	7	85
Input 10 -		4	48
Input 11 +		6	35
Input 11 -		4	97
Input 12 +		5	84
Input 12 -		4	47
Input 13 +	LEM sensors	13	34
Input 13 -		12	96
Input 14+		11	83
Input 14 -		10	46
Input 15 +		9	33
Input 15 -		8	95
Input 16 +	Vacuum	39	82
Input 16 -		38	45

Analog output			
Description	Use	DB50	MDR100
Output 1 +	PA 1	1	27
Output 1 -		2	26
Output 2 +	PA 2	5	28
Output 2 -		6	26
Output 3 +	PA 3	9	87
Output 3 -		10	26
Output 4 +	PA 4	13	86
Output 4 -		14	26
Output 5 +	PA 5	17	89
Output 5 -		32	26
Output 6 +	PA 6	50	90
Output 6 -		49	26
Output 7 +	PA 7	46	50
Output 7 -		45	26
Output 8 +	PA 8	42	49
Output 8 -		41	26
Output 9 +	PA 9	38	39
Output 9 -		37	26
Output 10 +	PA 10	34	40
Output 10 -		19	26

The interface board for the SBC is shown in figure 4-4. The SBC interface board piggy-backs directly onto the interface board with the protection circuitry. This board allows the FESS to be controlled by either the SBC or the dSPACE® controller.

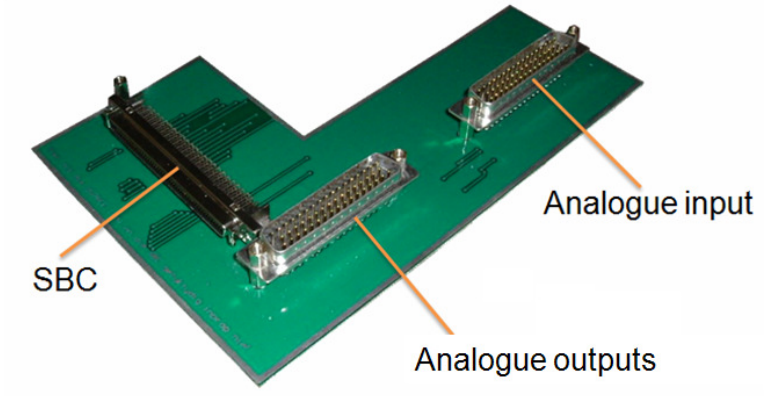


Figure 4-4: SBC interface board

4.2 Control algorithms

As mentioned at the end of chapter 2, PD control will be used to control the AMBs and voltage over frequency control will be used to drive the PMSM. The PD control algorithm as well as the voltage over frequency algorithm needs to be modified to be able to implement it in code. Section 4.2.1 and 4.2.2 explain how the algorithms are translated into equation form which can be coded more easily.

4.2.1 PD control

In chapter 2, (2.1) and (2.2) would be used to calculate the output of the PD control algorithm. Due to noise problems on the previous projects, it was decided to add a pole to the derivative part of the algorithm. The pole in the derivative part will be realised by making use of a first order low-pass filter. The PD control algorithm makes use of (2.2) as the input, as shown in figure 4-5.

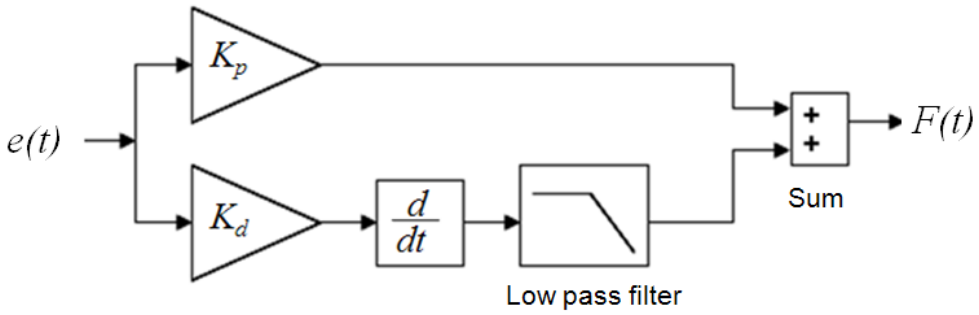


Figure 4-5: Simplified PD control

The first order low pass filter added to the algorithm can be realized by means of a capacitor and a resistor. The configuration shown in figure 4-6 will be used to derive the equations needed for the derivative part of the algorithm including the low pass filter.

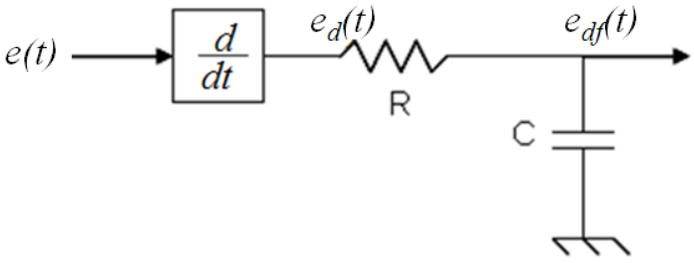


Figure 4-6: Differential part including a low pass filter

The output of PD control is given by (4.1).

$$PD(t) = K_p e(t) + K_d e_d(t) \tag{4.1}$$

The error position $e(t)$ is the difference between the reference position and the current position. This is given by (4.2).

$$e(t) = x_{ref} - x_{true}(t) \tag{4.2}$$

To calculate the derivative of the error, numeric differentiation in the digital domain is used. This is given by (4.3).

$$e_d(k) = \frac{e(k) - e(k-1)}{T} \tag{4.3}$$

where T is the sampling time in the digital domain and k is the k^{th} sampled element.

The transfer function of the first order low-pass filter is then calculated. The current in the capacitor is given by (4.4) and the current in the resistor is given by (4.5).

$$\begin{aligned} i_c(t) &= C\frac{d}{dt}e_{df}(t) \\ \frac{i_c(t)}{C} &= \frac{d}{dt}e_{df}(t) \end{aligned} \tag{4.4}$$

$$i_R(t) = \frac{e_d(t) - e_{df}(t)}{R} \tag{4.5}$$

The current in the capacitor is equal to the current in the resistor and it can be assumed that no current is drawn from the $e_{df}(t)$ node. By setting (4.4) equal to (4.5), (4.6) it results in:

$$\begin{aligned} \therefore \frac{d}{dt}e_{df}(t) &= \frac{1}{RC}\left(e_d(t) - e_{df}(t)\right) \\ &= \omega_p\left(e_d(t) - e_{df}(t)\right) \end{aligned} \tag{4.6}$$

The pole frequency is given by (4.7).

$$\omega_p = \frac{1}{RC} \tag{4.7}$$

The derivative can then be calculated by integrating (4.6) as shown in (4.8).

$$\begin{aligned} e_{df}(t) &= \int\left(\frac{d}{dt}e_{df}(t)\right)dt \\ &= \int\left(\omega_p\left(e_d(t)-e_{df}(t)\right)\right)dt \\ &= e_{df}(t)+T\cdot\frac{d}{dt}e_{df}(t) \end{aligned} \tag{4.8}$$

This concludes the derivation of the equations needed for the PD algorithm. Equations (4.1), (4.2), (4.6) and (4.8) are still in the continuous time domain and should be converted to the discrete digital domain. Equation (4.9) to (4.12) represents the converted equations which will be implemented directly into software including (4.3) to obtain the output for the PD control algorithm.

$$PD(k) = K_p e(k) + K_d e_d(k) \tag{4.9}$$

$$e(k) = x_{ref} - x_{true}(k) \tag{4.10}$$

$$\dot{e}_{df}(k) = \omega_p\left(e_d(k)-e_{df}(k-1)\right) \tag{4.11}$$

$$e_{df}(k) = e_{df}(k-1)+T\cdot\dot{e}_{df}(k) \tag{4.12}$$

4.2.2 PMSM control

Equations (2.3) to (2.7) are used to generate the 3-phase output. The only difficulty is calculating the integral of the speed. Due to the high sampling speed and low accuracy, a simplified method will be used. The integral over a specific time interval is equal to the area under the function during that time. This will be approximated by a rectangle with height equal to the value of the function at the specific time and width equal to the change in time as given by (4.9).

$$\theta_s^*(k) \approx \omega_s^*(k)\times T \tag{4.13}$$

Voltage outputs of (2.7) should then be converted to pulse-width modulated (PWM) signals. The true PWM signal (A1, B1, and C1 in figure 4-7) as well as the complementary PWM signal (A2, B2, and C2 in figure 4-7) should be generated for the 3-phase bridge. PWM signals will switch at a frequency of 50 kHz with a dead-band of 1 µs.

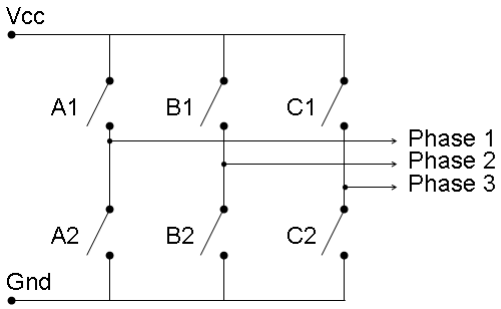


Figure 4-7: 3-Phase bridge

The TMS320C6713 DSP does not have on-chip PWM outputs, but could be generated in the software. By comparing a triangular wave with the reference signal, which is a sinusoidal wave in this application [33], a PWM signal can be generated. The dead-band ensures that the 3-phase bridge does not switch the high and low side of a phase simultaneously. This is implemented by producing another triangular wave, but with an offset as shown in figure 4-8.

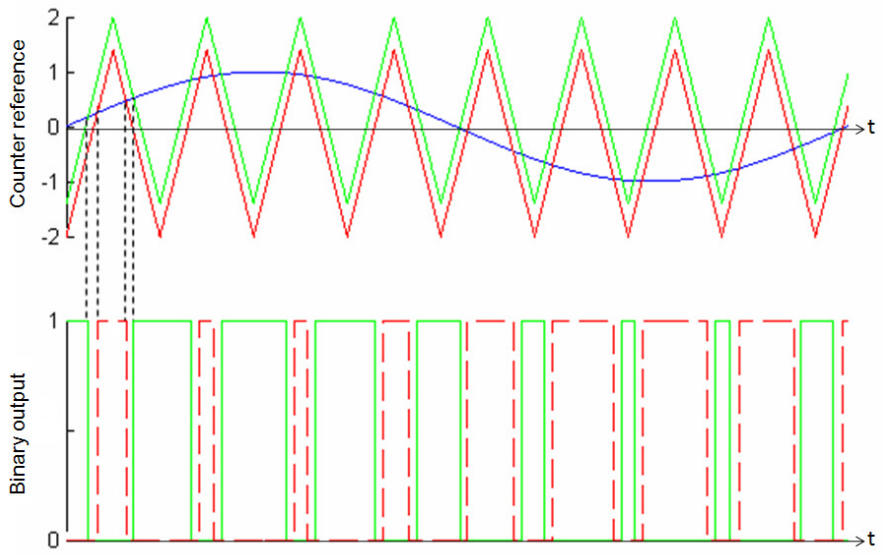


Figure 4-8: PWM signal generator

The blue line is the reference sinusoidal wave, green is the PWM signal and red is the inverted PWM signal.

A triangular wave will be generated by means of an up/down counter in software. The offset values will be added and subtracted within the compare function which will then generate the PWM outputs. A resolution of 100 points is added to the PWM signal, which means that the PWM signal should be updated at an frequency of 50 kHz x 100 points = 5 MHz.

An interrupt was enabled on the SBC to update the PWM signals at a rate of 5 MHz, but it was not possible for the SBC6713e. The SBC was kept busy by the interrupt for the complete cycle and did not get to the AMB and PMSM control at all. The resolution was reduced to 10 points, but it was still not enough to give time for the AMB and PMSM control.

This problem was overcome by doing all the calculations as planned, except for the conversion of the voltages to PWM signals. The SBC6713e still had 6 unused analogue outputs; therefore it was decided to write the three calculated voltages out on the DACs and make use of a single dSPACE® RT1104 card to convert the voltages to PWM signals.

4.3 SBC firmware

The SBC has three main pieces of code, namely the AMB-, PMSM control and the TCP/IP communication. This is also the order in which they would execute. For the AMB- and PMSM control, the SERVO16 analogue add-on board is required. This board deals with the entire analogue to digital as well as digital to analogue conversions. To communicate with this add-on board, Malibu software libraries from Innovative Integration (II) are used which was sourced with the SBC. These libraries include the setup and initialisation of the SERVO16 as well as libraries for the software communication interfaces on the SBC and the host PC.

The SBC will start with the SERVO16 initialisation, and then continue with the interrupt service routine, which executes every 50 μs. Within this routine, the AMB control and the PMSM control would be executed. After completion of the control algorithms, it will send the necessary information to the GUI and jump back to the main program. In the main program it will iterate in a while loop which waits for information from the GUI. It continues this while loop until the next interrupt service routine is called. The interrupt is serviced and it jumps back to the while loop. The timeline for the SBC code execution is shown in figure 4-9 and the complete SBC firmware flow is shown in figure 4-10.

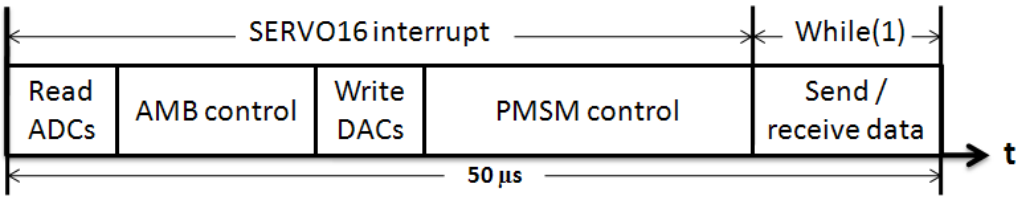


Figure 4-9: SBC code execution timeline

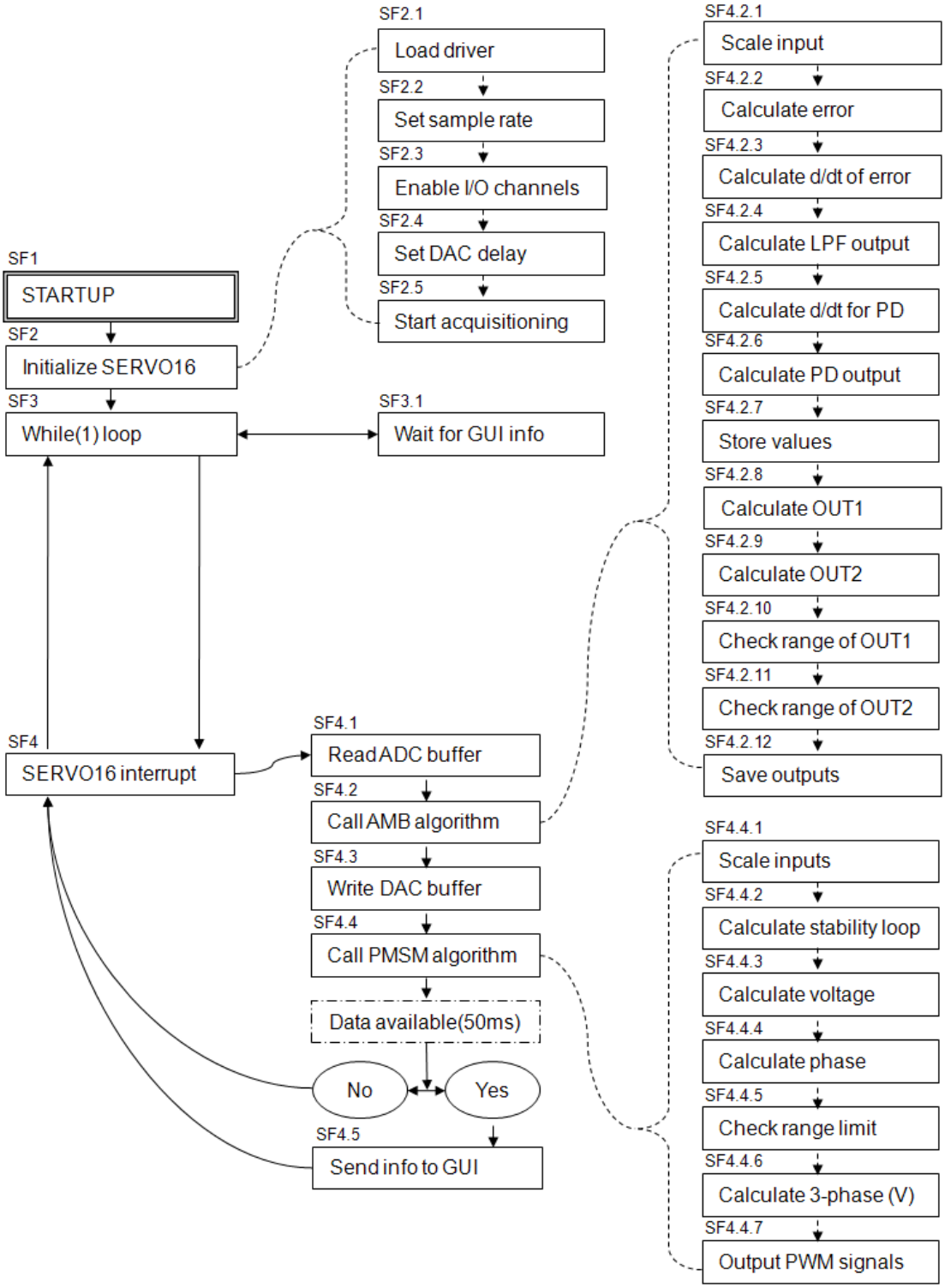


Figure 4-10: SBC firmware flow diagram

The SBC starts with the start-up procedure (software function 1, SF1) and continues with the initialisation of the SERVO16 (SF2). It first loads the SERVO16 add-on board drivers, and then starts to set the parameters. Parameters that should be set are the sample rate, number of channels and

the delay time between the ADC and DAC. The purpose of the delay time is to give the PD control algorithm time to finish before the outputs to the PAs are written. If this time is too short, the values of the previous samples will be written out.

The following section describes the code segments with reference to figure 4-10. Each code segment contains a reference number, for example "(SF2.1)", which correlates to the software function in figure 4-10. (All text to the right of a double forward slash should be seen as comments and not as part of the code that is executed.)

The following code shows the setup and the start of the SERVO16 add-on board from software functions (SP2.1) to (SP2.5).

```
// Load SERVO16 board driver                    (SF2.1)

LoadModule(Omnibus::mSite0, Omnibus::mtSERVO16);
```

```
// Set sample rate (Hz)                         (SF2.2)

ServoIo.Clock().Rate(20000);
```

```
// Enable all analogue I/O channels             (SF2.3)

ServoIo.InputChannels().EnableChannels(16);

ServoIo.OutputChannels().EnableChannels(16);
```

```
// Set delay of DAC clock (ns)                  (SF2.4)

ServoIo.Delay(20000);
```

```
// Start SERVO16 acquisitioning                 (SF2.5)

ServoIo.Start();
```

This completes the setup of the SERVO16 board. The SERVO16 will now call an interrupt after each ADC acquisition. The AMB- and PMSM control

algorithms will then be executed from the interrupt. From the set-up above, the control algorithms would execute 10000 times per second.

The AMB control is of higher priority, therefore the interrupt starts off with it. All sampled data are stored in a buffer. To prevent a buffer overflow, all samples written by the ADC should be read, and after completing the execution of the control algorithms, the newly calculated values should be written again. To do this the following code is used.

```
// Read available analogue values                    (SF4.1)
for (int i=0; i<inputs; ++i)
     ADC[i] = event[i];
```

```
// Write available analogue values                   (SF4.3)
for (int i=0; i<outputs; ++i)
     event[i] = DAC[i];
```

When the interrupt is called, all available analogue values are retrieved from the buffer, the AMB algorithm is executed and the analogue values are written. The PMSM control has a lower priority and therefore its algorithm would be executed after the DACs are written. The result of this would be that the PMSM algorithm is one sample time behind. In other words, the PMSM output would have a delay of 50 μs due to the sampling frequency, plus the DAC delay of 20 μs which adds up to 70 μs, while the delay for the AMB control is only the DAC delay of 20 μs.

The AMB algorithm is called 5 times, once for each axis. Each axis results in two outputs, therefore 5 inputs should result in 10 outputs, one for each PA. The AMB algorithm function takes 2 parameters, the position sensor value obtained from the ADC as well as the number of the axis as shown in the following code.

```
// Call the AMB function class                          (SF4.2)
for (int k=0; k<5; ++k)
     {
        algor(ADC[k],k);
        DAC[2*k] = TEMP[0];
        DAC[2*k+1] = TEMP[1];
     }
```

In the algorithm itself, some scaling is done first and then (4.2) is used to obtain the error value. (Refer to figure 3-3 for a visual representation of the algorithm). The derivative of the error is then calculated using (4.3) as well as the output of the first order low-pass filter with (4.6). Finally the derivative part for the PD algorithm is calculated using (4.8) and the PD output, using (4.1).

```
// Scale input values to meters                         (SF4.2.1)
input = ADC[k]*1.27e-3/2^15;
```

```
// Calculate the error from reference                   (SF4.2.2)
e1 = reference[k] - input;                          // (4.2)
```

```
// Calculate the derivative of the error                (SF4.2.3)
ed1 = (e1-e[k]*10000);                              // (4.3)
```

```
// Calculate the low-pass filter(LPF) output            (SF4.2.4)
epdd1 = (ed1-epd[k])*pole;                          // (4.6)
```

```
// Get the result of the LPF and derivative             (SF4.2.5)
epd1 = (epdd1/10000)+ epd[k];                       // (4.8)
```

```
// Calculate PD control output                          (SF4.2.6)
PD = Kp[k]*e1 + Kd[k]*epd1;                         // (4.1)
```

Before continuing with the AMB control, all necessary values should be stored for use in the following interrupt cycle. These values are the following:

```
// Store PD control values                                    (SF4.2.7)
e[k]= e1;
ed[k]= ed1;
epd[k]= epd1;
epdd[k]= epdd1;
```

The PD control output is then split into two components, one for each actuator per axis control. First the bias current is added/subtracted for each actuator output, and then scaled to the DAC range. The calculation of the outputs 'OUT1' and 'OUT2' are shown in the following code.

```
// Calculate OUT1                                             (SF4.2.8)
OUT1 = bias_current+PID;
     OUT1 *= amp_gain;
     OUT1 *= scaling;
```

```
// Calculate OUT2                                             (SF4.2.9)
OUT2 = bias_current-PID;
     OUT2 *= amp_gain;
     OUT2 *= scaling;
```

Before these values can be written to the DACs, it should be checked that the values are in the acceptable current range of the PAs. The PAs cannot receive a negative value and the current is also limited to 5 ampere maximum.

```
// Check range of OUT1                                    (SF4.2.10)
if (OUT1<0)            // Check if OUT1 is negative
     OUT1 = 0;
else if (OUT1>16384)   // Limit OUT1 to 5 ampere
OUT1 = 16384;
```

```
// Check range of OUT2                                    (SF4.2.11)
if (OUT2<0)            // Check if OUT2 is negative
     OUT2 = 0;
else if (OUT2>16384)   // Limit OUT2 to 5 ampere
     OUT2 = 16384;
```

If all is in range, the outputs are stored and the call to the algorithm writes these values to the required variables.

```
// Save AMB outputs                                       (SF4.2.12)
TEMP[0] = OUT1;
TEMP[1] = OUT2;
```

This concludes the AMB control algorithm. The PMSM control algorithm starts executing directly after all the DAC values are written into the buffer. This algorithm starts similar to the AMB control algorithm. First some scaling is done on two of the current sensor inputs:

```
// Scale inputs for PMSM to ampere                        (SF4.4.1)
ADC[12] *= 1000/33/scaling;
ADC[13] *= 1000/33/scaling;
```

These current sensor inputs are then used to calculate the voltage and phase of the voltage over frequency control method. A stability function was added to this control scheme. This was to overcome the moment of inertia of the rotor at standstill until 10 radians per second (rad/s).

```
// Get the result of the stability loop          (SF4.4.2)
if (Wo<(20*pi))
     Dwe = 0;
else
     Dwe = -0.1/Wo;
Dwe *= 3/2*Vs*IscosT;
```

Equations (2.4), (2.5) and (2.6) are then implemented to obtain the voltage and phase.

```
// Calculate the instantaneous current            (SF4.4.3)
Is = 2*ADC[13];                                  // (2.4)
     Is += ADC[12];
     Is = pow(Is,2);
     Is /= 3;
     Is += pow(ADC[12],2);
     Is = pow(Is,0.5);
// Calculate the instantaneous phase
IscosT = ADC[12]*cos(T);                         // (2.5)
     IscosT += ADC[13]*cos(T-2*pi/3);
     IscosT -= (ADC[12]+ADC[13])*cos(T+2*pi/3);
     IscosT *= 2/3;
// Calculate the instantaneous voltage
Vs = pow(2*pi*fo*Lm,2);                          // (2.6)
     Vs += pow(IscosT,2)*pow(rs,2);
     Vs -= pow(Is,2)*pow(rs,2);
     Vs = pow(Vs,0.5);
     Vs += IscosT*rs;
     Vs += 15;
```

```
 // Calculate the phase                                    (SF4.4.4)
T += (Wo-Dwe)/SampleRate;
T = fmod(T,2*pi);
```

Again some scaling is done and the range is limited for the PWM from 5% to 95%.

```
// Check voltage for duty cycle range                     (SF4.4.5)
Vs /= 240;
if (Vs > 0.45)
     Vs = 0.45;
else if (Vs < -0.45)
     Vs = -0.45;
```

The PMSM is a 3-phase motor with the phases distributed evenly; therefore the instantaneous voltage of each phase can be calculated as follows:

```
// Calculate 3-phase voltages                             (SF4.4.6)
Va = Vs*sin(T);
Vb = Vs*sin(T-2*pi/3);
Vc = Vs*sin(T+2*pi/3);
```

These three voltages are then converted to analogue values, fed to dSPACE® and converted to PWM signals and routed to the PMSM driver which makes use of a 3-phase bridge to drive the motor. This concludes the PMSM control algorithm.

The third piece of the SBC code is the communication between the SBC and host PC which is done via Ethernet (TCP/IP). The GUI on the host PC is only there to monitor the system. Not all the sampled data will be sent to the GUI or else this will occupy the network the whole time. A person's reaction time from seeing something and reacting on that is in the order of one to two seconds. Fifty samples per second of all the sampled data would then be sufficient for the operator to see what happens and still leave enough time for other applications to use the network. Data will flow bi-directional. Information will be

sent to the GUI and information (commands or variables) will be send back to the SBC. Data sent to the GUI includes all the analogue sampled values, while data sent to the SBC includes all the K_p and K_d constants of the PD control, offsets of the rotor positions, PMSM rate and the start/stop commands of the AMB and PMSM algorithms.

4.4 GUI

A GUI was designed taking human sight limitations into consideration. These sight limitations include the ability for a human to see different colours better at different angles. From the centre, moving outwards, red and green are most visible, while yellow and white is still visible on the outside [34]. Data from [34] was used to create the colour usage image as shown in figure 4-10.

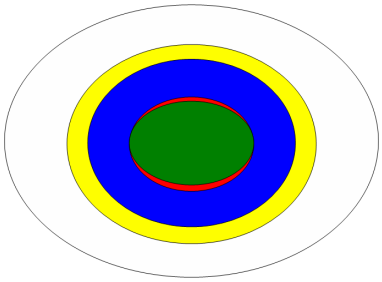


Figure 4-11: Colour usage

The control screen will be divided into three main parts. The critical information in the centre of the screen, the less critical information on the border of the screen, and the controls grouped on one side of the screen. Figure 4-11 shows the screen usage of the three main parts.

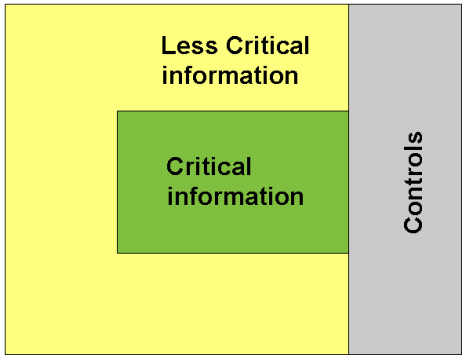


Figure 4-12: Screen usage

The critical information will comprise of the three rotor levitation status graphs, one for each AMB. The graphs will show the real-time position of the rotor. The less critical information is the status on the PMSM, the rotor speed, etc.

Another three graphs will be displayed, but with the status on the current and voltage in each phase of the PMSM. The rotor speed will be displayed as a number. The controls include the levitation of the rotor and the start/stop of the PMSM.

The host PC software includes only the GUI. As with the SBC firmware, the Malibu libraries are again included, but this time only for the communication. The SCB and the host PC makes use of a shared library to create a common interface which both would understand. All the data sent and received are defined within these libraries, as well as the methods to communicate.

No control algorithms are executed on the host PC allowing the FESS to operate as a stand-alone system. The GUI has 2 tabs, the first tab is to connect to the SBC and download the program when necessary. As shown in figure 4-12, this tab consists of the IP address of the SBC, a connection button, a browser button to select the SBC program file and an event log to see the current status of operation.

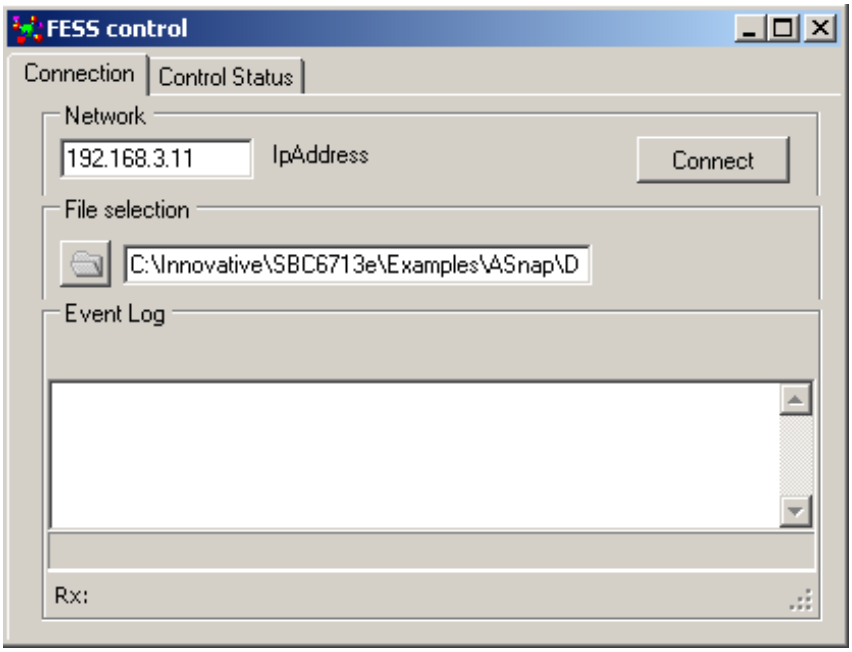


Figure 4-13: GUI layout: Connection tab

The second tab is the control status tab. This tab layout differs from the planned screen usage shown in figure 4-11. It was decided that for ease of use, the controls of each AMB should be situated close to the real-time status graph as shown in figure 4-13. All the "stop/start" buttons are situated at the top of the screen, in close proximity to each other. Furthermore, the temperature

and PMSM current and voltages were grouped together on one side of the screen and the AMBs on the right side of the screen.

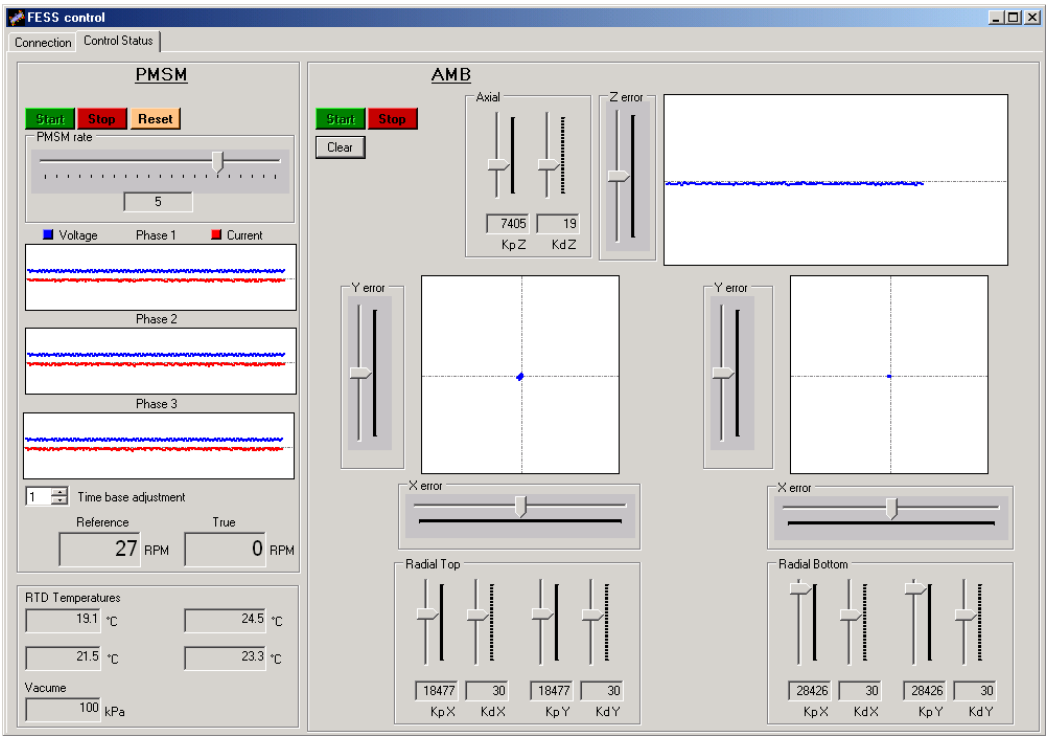


Figure 4-14: GUI layout

A software lock was implemented on the AMB- and PMSM start/stop buttons. The software lock is to ensure that the AMBs are not turned off while the rotor is still turning at a high speed. Also, the PMSM cannot be started before the AMBs are started. This software lock only allows the user to operate the FESS under safe operating conditions, thus omitting failure due to human error.

The PMSM should be able to reach the maximum speed of 30,000 rpm within 15 minutes. It is calculated by (4.10) and (4.11) that the required rate should be at least 3.49 radians per second squared (rad/s^2).

$$\omega_{\max} = \frac{30000\text{rpm} \times 2\pi\text{rad}}{60} = 3141.59\,\text{rad/s} \tag{4.14}$$

$$rate = \frac{\omega_{\max}}{(15 \times 60)\text{seconds}} = 3.49\,\text{rad/s}^2 \tag{4.15}$$

When the AMBs are started, and the PMSM start button is pressed, the PMSM will start to spin-up at a rate of 5 rad/s^2. This can then be adjusted by means of the slider. A maximum rate of 10 rad/s^2 is allowed by the slider. To stop the PMSM from accelerating, the slider can be moved to zero, and to decelerate the PMSM, the slider can be moved to the negative half of the slider. When the PMSM stop button is pressed, the slider will automatically jump to -5 rad/s^2 and start to decelerate. The reset button disables the PMSM control and will result in the rotor turning free. The currents and voltages of the 3-phases of the PMSM will be displayed just below the slider as shown in figure 4-13.

The screen division for the AMBs consists of three graphs with each its own set of controls. The top-left corner contains the axial AMB status plot. This plot is the axial displacement against time. To the left of the graph is the offset adjustment slider as well as the proportional and derivative constant adjusters. This allows the user to adjust the offset position as well as the PD control constants real-time.

To the bottom of the screen, are the two XY-plots of the top- and bottom radial AMBs. Again the position offset sliders are situated right next to the graphs. The PD constants of each AMB are situated beneath its graph. These can also be adjusted real-time. The temperatures of the AMBs, measured by the RTDs, are shown on the left-bottom side of the screen together with the vacuum pressure inside the enclosure.

To start the FESS, the sequence shown in figure 4-14 would normally be followed.

All the yellow action blocks in figure 4-14, are sliders in the GUI and the blue action blocks are press buttons. To start the FESS, open the GUI application. Both the SBC and the host PC with the GUI should be connected to a local network. The SBC's IP address is entered in the space provided. The default setting is '192.168.3.11', but can be altered when the IP differs.

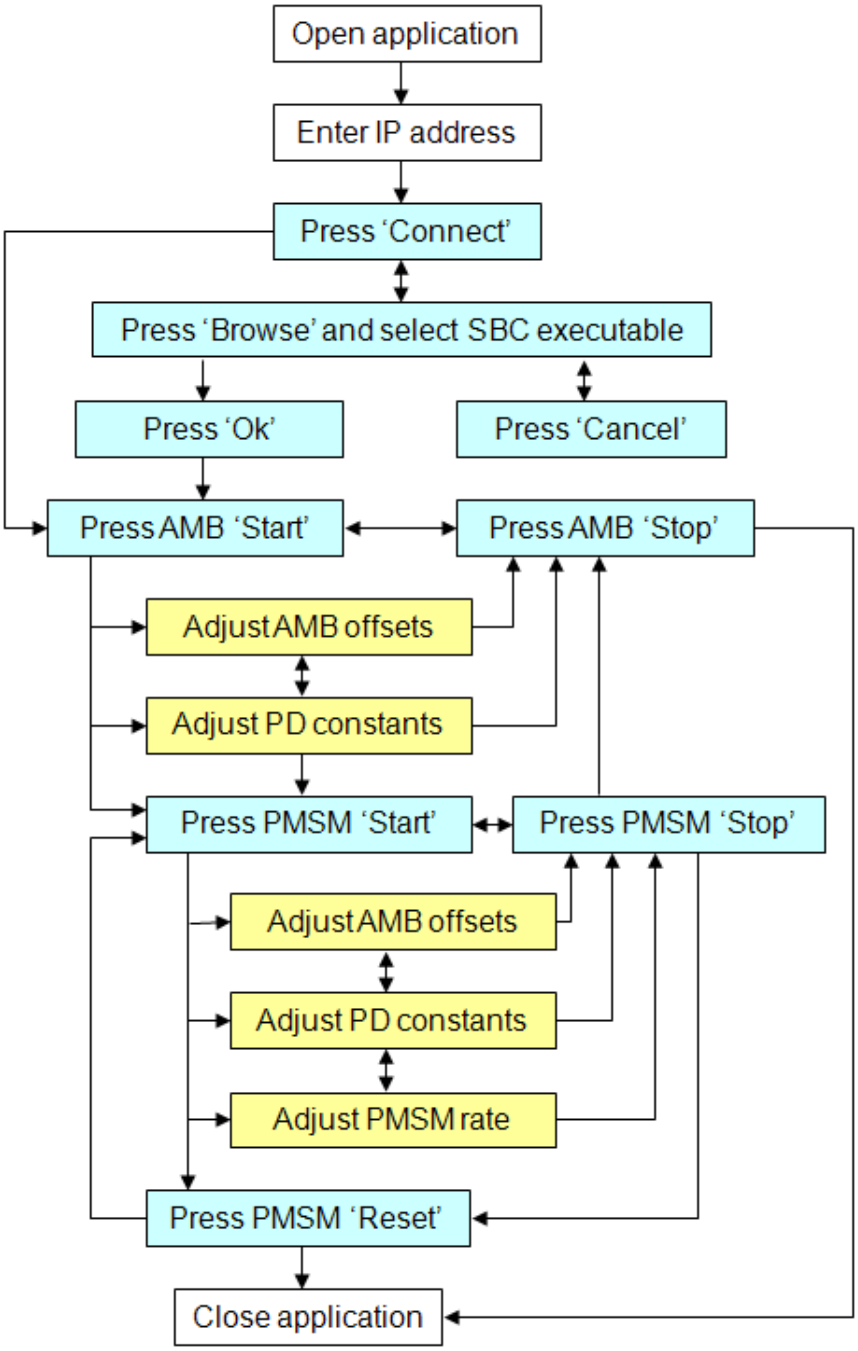


Figure 4-15: GUI start-up sequence

Now there are two possibilities. If the executable is not loaded on the SBC, press the browse button, select the executable and press 'ok', else continue to the 'control status' tab. Now the AMBs can be started, offsets adjusted, PD constants adjusted and the PMSM started. The AMBs can be stopped at anytime, as long as the PMSM is not rotating.

If the PMSM is started, the AMBs' offsets, PD constants and PMSM rate could still be changed. The PMSM reset can also be pressed anytime, irrespective of

the status of the PMSM rate. The application can also be closed at any time. When the application closes, the PMSM and AMBs will be stopped automatically.

Chapter 4 presented the two interface boards required by the FESS, the firmware design and software considerations for the GUI. The actual firmware and software descriptions followed as well as the start-up sequence for the GUI application. Evaluation topics in Chapter 5 will include: controller performance testing, stability analyses, sensitivity analyses and PMSM control.

5 Chapter

Testing and evaluation

Chapter 5 provides the testing procedure and results followed by an evaluation of the results. This includes 4 tests, namely: controller performance, stability, sensitivity as well as the PMSM tests.

5.1 Testing procedure

Four main tests were done on the system. First the performance of the controller was tested to confirm that it complies with specifications given. Secondly the system stability, thirdly the system sensitivity tests were done and lastly the permanent magnet synchronous machine (PMSM) control was tested.

5.1.1 Controller performance

To test the performance of the controller, the required control algorithms were implemented on the single board computer (SBC). The time taken to complete each algorithm was recorded and analysed. As shown in figure 5-1, the execution of the AMB and PMSM control algorithm took a little longer than the expected 50 μs. This resulted in the adjustment of the sampling frequency from 20 kHz as specified, to 10 kHz. This would still leave enough time for the data transfer over the ethernet cable for the GUI.

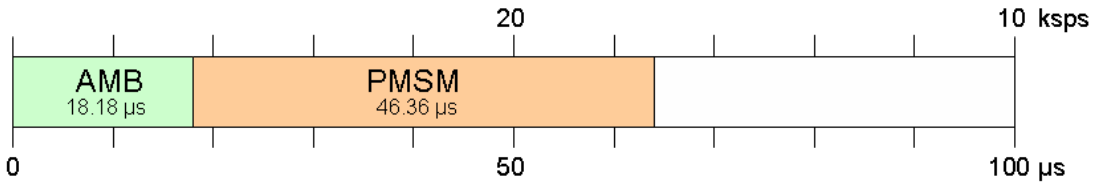


Figure 5-1: Control algorithm execution timing

A control cycle of 10 kHz proved to be adequate on other systems; therefore the 10 kHz ought to be sufficient for the control of the FESS.

5.1.2 System stability

To test the system stability, a step response test was done on all five degrees of freedom. A step of 10 μm is inserted in the reference and the response was recorded. The actual response of the system was then compared to the simulated response as well as the response obtained from the dSPACE® system. Figure 5-2 shows where the step error is inserted into the control system. The SBC records the data and sends it to the GUI as a string of values. These values are then imported into MATLAB®® and plotted against the simulated results.

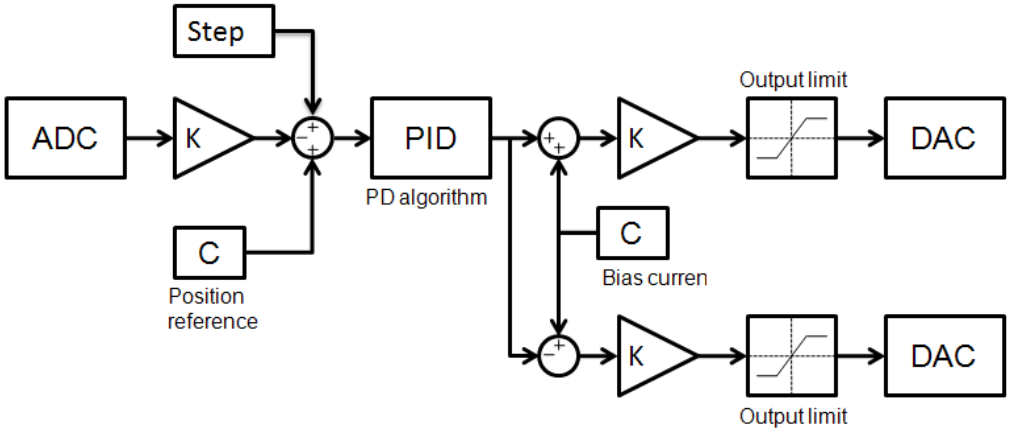


Figure 5-2: Step input on AMB control algorithm

This graph is then used to obtain values for the settling time (T_s) as well as the percentage overshoot ($P.O.$). The damping ratio (ζ), natural frequency (ω_n), equivalent stiffness (k_{eq}) and equivalent damping (b_{eq}) are then calculated with (5.1) – (5.4) respectively [30].

$$P.O. = 100e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{5.1}$$

$$\omega_n = \frac{4}{T_s\zeta} \tag{5.2}$$

$$k_{eq} = \omega_n^2 m \tag{5.3}$$

$$b_{eq} = \zeta \cdot 2 \cdot \sqrt{k_{eq} \cdot m} \tag{5.4}$$

where m is the equivalent mass of the rotor. The rotor mass is 18.3 kg, but the equivalent mass of the rotor for the bottom radial AMB is 5.38 kg and 13.22 kg for the top AMB.

Figure 5-3 shows the bottom radial x-axis response. There was a lot of noise on the sensor of the bottom radial AMB. A filter was implemented to help obtain realistic values for the settling time and the percentage overshoot, but by doing this, not very accurate results are obtained. More accurate results can be obtained by identifying the source of the noise and removing it.

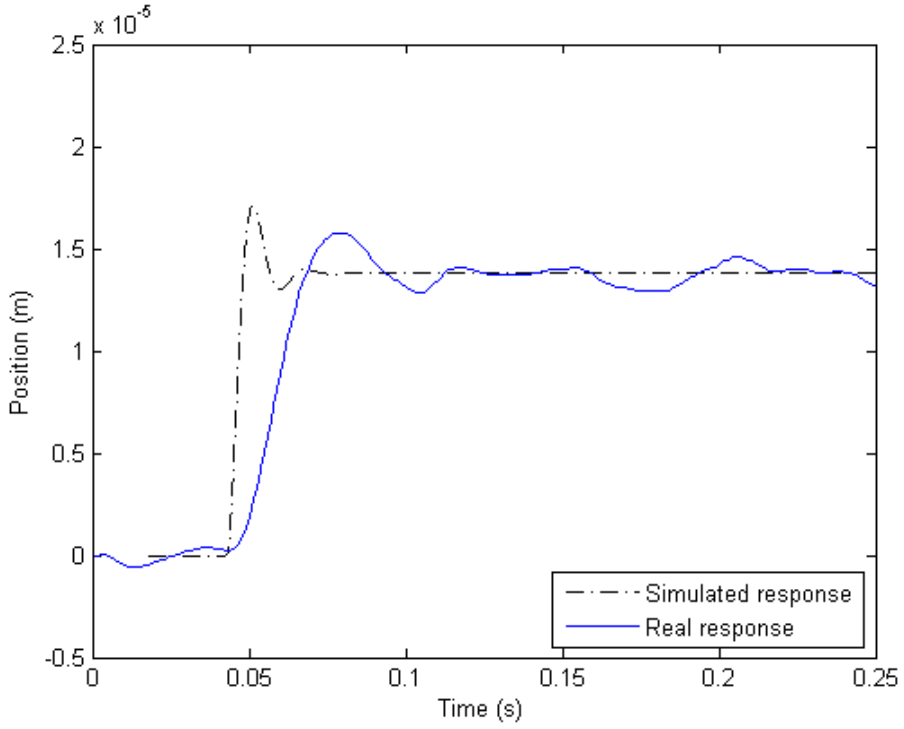


Figure 5-3: Bottom radial AMB x-axis step response

Table 5-1 shows the results obtained for the bottom radial x-axis. The step response results on the SBC are worse than the dSPACE® system, the equivalent stiffness and the damping is less than the dSPACE® system and the simulated results. This is due to the settling time that is much longer.

Table 5-1: Bottom x-axis control comparison

	Simulated	dSPACE®	SBC
$P.O.$ (%)	24.0	27.2	25.7
T_s (s)	0.0206	0.0208	0.1570
ζ	0.4138	0.3825	0.3969
ω_n (rad/s)	468.3	503.7	64.2
k_{eq} (N/m)	1,180,000	1,365,000	22,163
b_{eq} (N.s/m)	2,085	2,073	274

The same noise was obtained in the bottom radial y-axis response as in the x-axis response. The step response for the y-axis after filtering is shown in figure 5-4.

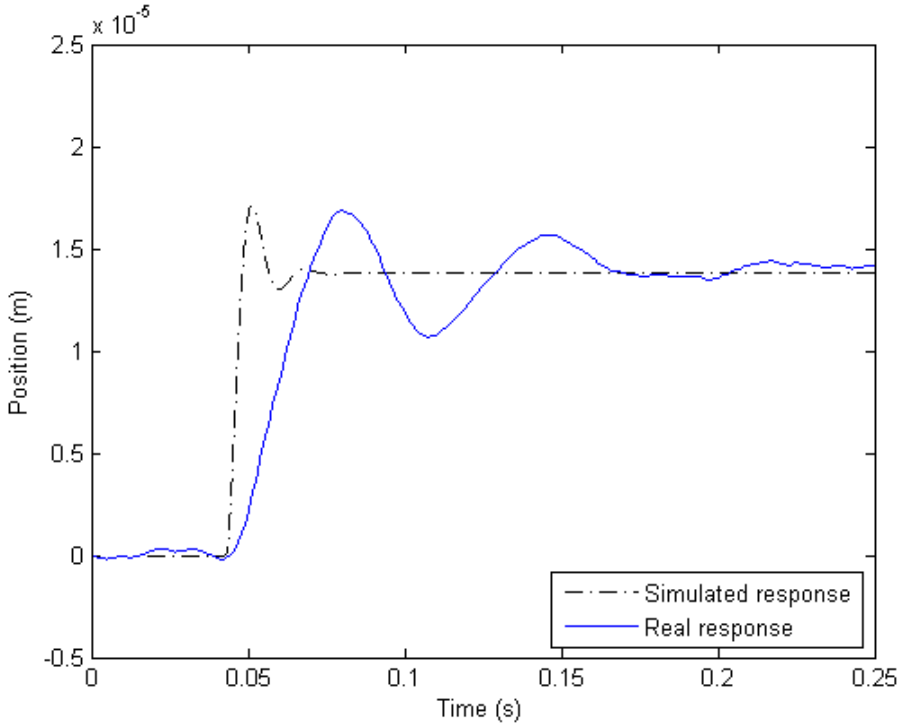


Figure 5-4: Bottom radial AMB y-axis step response

The step response in the bottom y-axis had a shorter settling time, but the equivalent stiffness and damping is still worse than the values obtained by the dSPACE® system for the y-axis. The results for the step response are summarized in table 5-2.

Table 5-2: Bottom y-axis control comparison

	Simulated	dSPACE®	SBC
$P.O.$ (%)	24.0	28.5	26.4
T_s (s)	0.0206	0.0220	0.1510
ζ	0.4138	0.3712	0.3903
ω_n (rad/s)	468.3	488.7	67.9
k_{eq} (N/m)	1,180,000	1,285,000	24,782
b_{eq} (N.s/m)	2,085	1,952	285

The result for the top radial x-axis has a lower percentage overshoot than the results for the bottom radial AMB. Figure 5-5 confirms this. The noise is very little compared to the step signal, thus much more accurate results are obtained from this data.

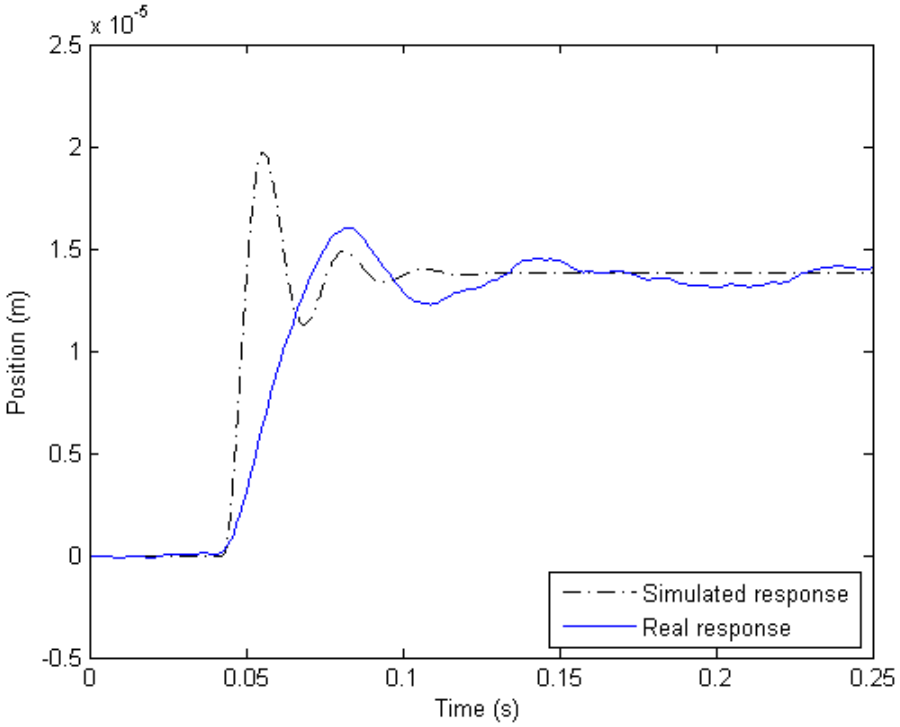


Figure 5-5: Top radial AMB x-axis step response

The percentage overshoot was less than the dSPACE® system and the simulated result, and the settling time was still longer than both dSPACE® and simulated results as shown in table 5-3. This caused the equivalent damping and stiffness to be lower than expected.

Table 5-3: Top x-axis control comparison

	Simulated	dSPACE®	SBC
$P.O.\ (\%)$	39.6	44.8	23.2
T_s (s)	0.0651	0.0736	0.1550
ζ	0.2830	0.2474	0.4217
ω_n (rad/s)	217.2	219.7	61.2
k_{eq} (N/m)	623,500	638,100	49,512
b_{eq} (N.s/m)	1,625	1,437	682

Again the noise is much less in the top radial y-axis than in the bottom radial AMB. Figure 5-6 shows the result obtained for the top radial y-axis.

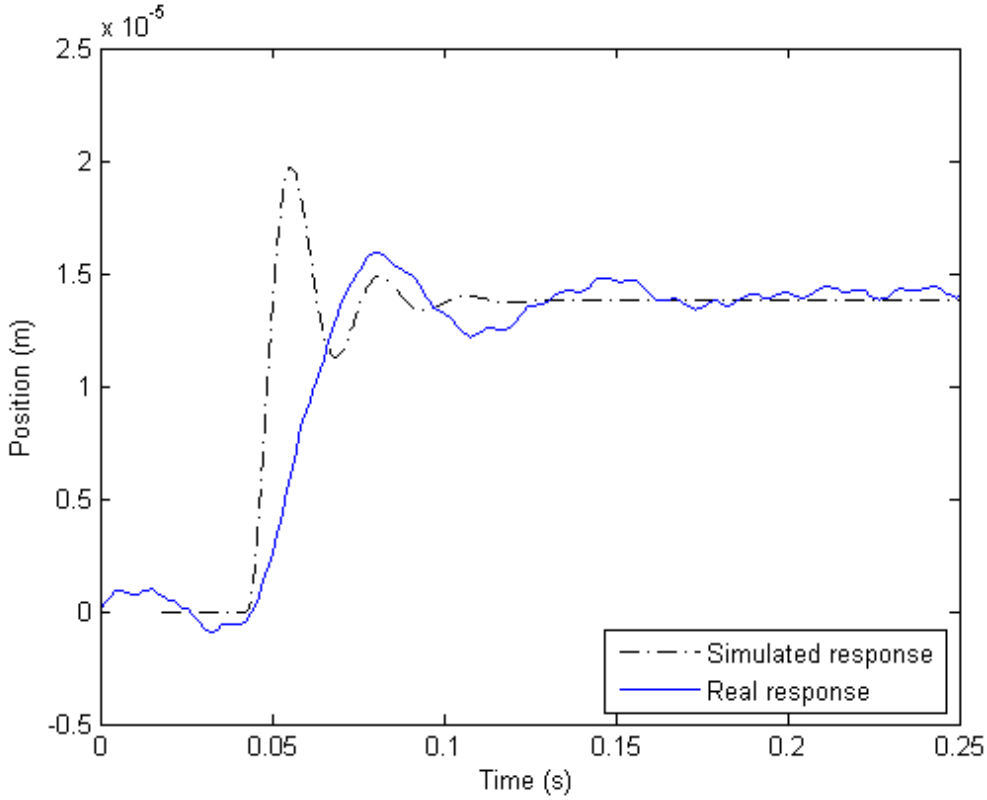


Figure 5-6: Top radial AMB y-axis step response

As with the top radial x-axis, the percentage overshoot is less than the dSPACE® and simulated results. Table 5-4 present the data obtained for the top radial y-axis. Again the equivalent damping and stiffness is less than the results obtained by the dSPACE® system as well as the simulated results.

Table 5-4: Top y-axis control comparison

	Simulated	dSPACE®	SBC
$P.O.$ (%)	39.6	46.0	23.7
T_s (s)	0.0651	0.0716	0.1520
ζ	0.2830	0.2403	0.4166
ω_n (rad/s)	217.2	232.7	63.2
k_{eq} (N/m)	623,500	715,600	52,749
b_{eq} (N.s/m)	1,625	1,478	696

Figure 5-7 shows the step response result for the axial AMB.

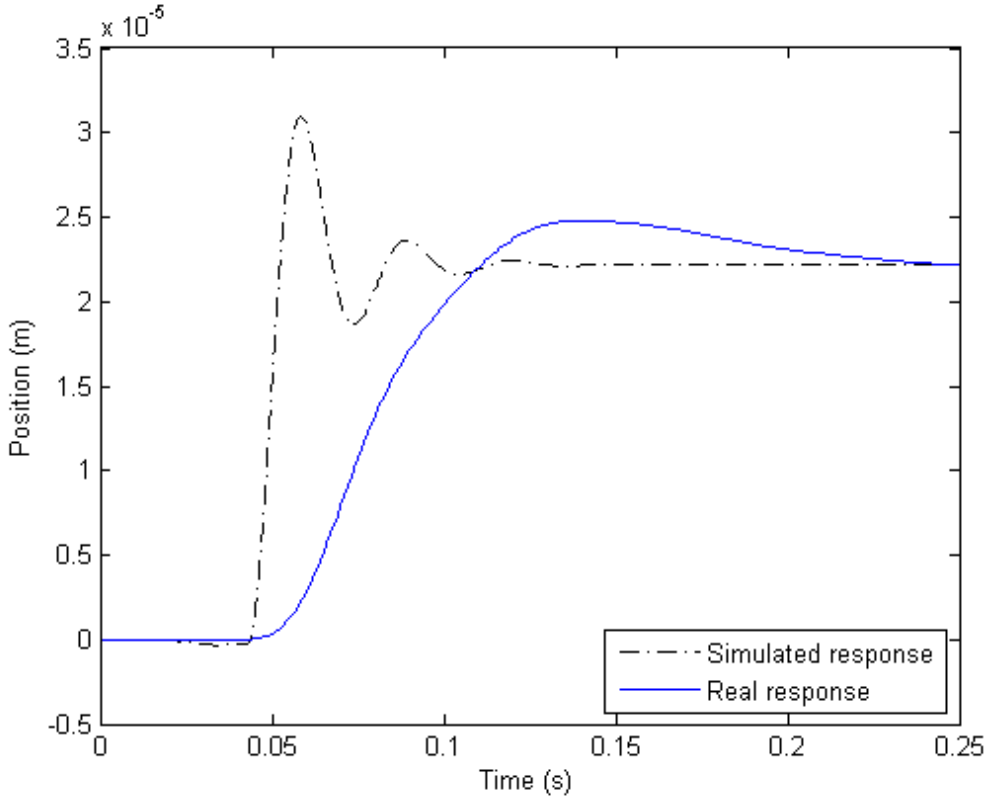


Figure 5-7: Axial AMB step response

Finally the step response result for the axial AMB is displayed in table 5-5. The percentage overshoot is much less than expected, and the settling time is also much longer. Therefore this results in a much lower equivalent damping and stiffness.

Table 5-5: Axial control comparison

	Simulated	dSPACE®	SBC
$P.O.$ (%)	39.1	37.3	9.3
T_s (s)	0.0661	0.0627	0.1510
ζ	0.2866	0.2996	0.6031
ω_n (rad/s)	211.2	213.0	43.9
k_{eq} (N/m)	829,400	844,200	35,887
b_{eq} (N.s/m)	2,251	2,374	985

In all 5 step response results, the equivalent stiffness is much less than expected, while the equivalent damping is also less. The cause of this large variation in results obtained should be clarified. The only big difference between the dSPACE® system and the SBC is the filters on the analogue to digital converter (ADC) and digital to analogue converter (DAC) board. There is a 4-pole elliptic filter, with the -3 dB attenuation set on 30 kHz, on the analogue inputs and a single pole filter, with the -3 dB attenuation set on 50 kHz, on the analogue outputs. The dSPACE® system contains no filter on the DACs and ADCs.

By adding these filters to the simulation, the results obtained from the simulation correlated more closely to the results obtained from the FESS. Figure 5-8 shows the results obtained from the adapted simulation compared to the actual results of the bottom radial AMB respectively.

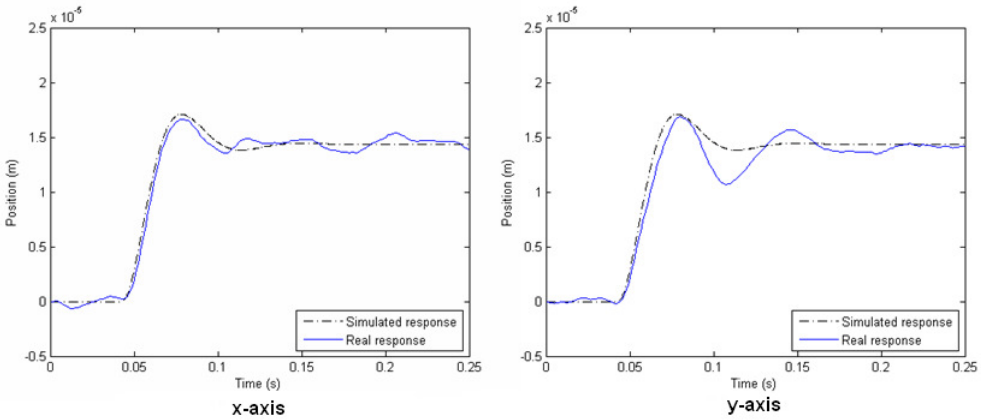


Figure 5-8: Bottom radial AMB step response

The equivalent damping and stiffness of the AMB are as expected. Table 5-6 compare these results.

Table 5-6: Bottom radial AMB with filters

	Simulated	SBC (X)	SBC (Y)
$P.O.$ (%)	26.0	25.7	26.4
T_s (s)	0.1500	0.1570	0.1510
ζ	0.3940	0.3969	0.3903
ω_n (rad/s)	67.7	64.2	67.9
k_{eq} (N/m)	24,658	22,163	24,782
b_{eq} (N.s/m)	287	274	285

Figure 5-9 shows the results obtained for the top radial AMB, again with the filters added to the simulation.

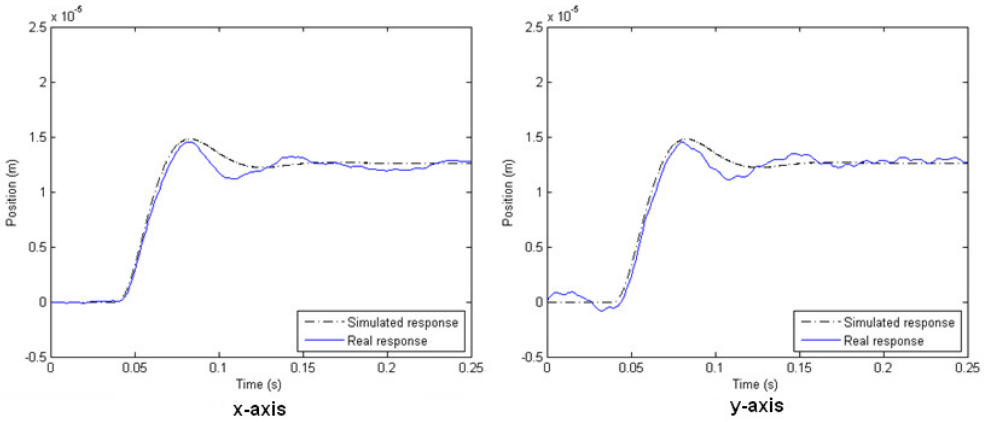


Figure 5-9: Top radial AMB step response

The equivalent damping and stiffness is a little less than expected, but still acceptable. Table 5-7 shows the top radial AMB's step response results.

Table 5-7: Top radial AMB with filters

	Simulated	SBC (X)	SBC (Y)
$P.O.$ (%)	24.1	23.2	23.7
T_s (s)	0.1500	0.1550	0.1520
ζ	0.4122	0.4217	0.4166
ω_n (rad/s)	64.7	61.2	63.2
k_{eq} (N/m)	55,314	49,512	52,749
b_{eq} (N.s/m)	705	682	696

The axial AMB's simulated and actual results correlated closely. The actual response had a larger percentage overshoot, but a shorter settling time as shown in figure 5-10.

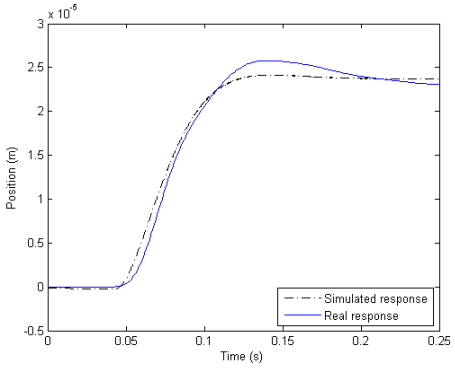


Figure 5-10: Axial AMB

The stiffness and damping for the axial AMB was a bit higher than expected. Table 5-8 shows the results for the axial AMB.

Table 5-8: Axial AMB with filters

	Simulated	SBC (axial)
$P.O.\ (\%)$	7.8	9.3
T_s (s)	0.1517	0.1510
ζ	0.6306	0.6031
ω_n (rad/s)	41.8	43.9
k_{eq} (N/m)	32,525	35,887
b_{eq} (N.s/m)	981	985

The filters were neglected when the SBC was sourced due to the high cut-off frequencies. One filter had a cut-off frequency of 30 kHz and the other 50 kHz. The system operates at 10 kHz, and therefore the filters should not affect the system negatively. The orders of the filters were overlooked. The phase shift of these filters caused the lowered stiffness and damping of the AMBs.

Although the stiffness and damping of the AMBs are not the same anymore, the sensitivity tests were still performed to observe the effect that these filters have.

5.1.3 System sensitivity

Compared to passive bearings, e.g. ball bearings or oil-film bearings, magnetic bearings have a negative stiffness due to static magnetic forces, resulting in magnetic bearings being unstable. Feedback control is required in active magnetic bearings to provide a positive stiffness and damping [36]. The sensitivity of an AMB system is specified by ISO CD 14839-3. In this standard, the sensitivity of a system is defined as the ratio of the response of the system to the input of a disturbance signal given by (5.5). The sensitivity of a system is given by the worst rating obtained from all the AMBs in the system.

$$G_s(s) = 20 \cdot \log\left(\frac{V_R(s)}{V_D(s)}\right) \tag{5.5}$$

Systems can be grouped into one of four zones [36].

Zone A: The sensitivity functions of newly commissioned machines would normally fall within this zone.

Zone B: Machines with the sensitivity functions within this zone are normally considered acceptable for unrestricted long-term operation.

Zone C: Machines with the sensitivity functions within this zone are normally considered unsatisfactory for long-term continuous operation. Generally, the machine may be operated for a limited period in this condition until a suitable opportunity arises for remedial action.

Zone D: The sensitivity functions within this zone are normally considered to be sufficiently severe to cause damage to the machine.

Table 5-9 presents the peak sensitivity for each zone. The FESS falls into zone A, which results in a maximum allowable peak sensitivity of 8 dB.

Table 5-9: Peak sensitivity ate zone limits [36]

Zone	Peak sensitivity level
A/B	8 dB
B/C	12 dB
C/D	14 dB

ISO CD 14839-3 defines the sensitivity as the ratio of V_R over V_D, where V_D is a sinusoidal disturbance with varied frequency [36]. The input of V_D and the reference V_R is shown in figure 5-11.

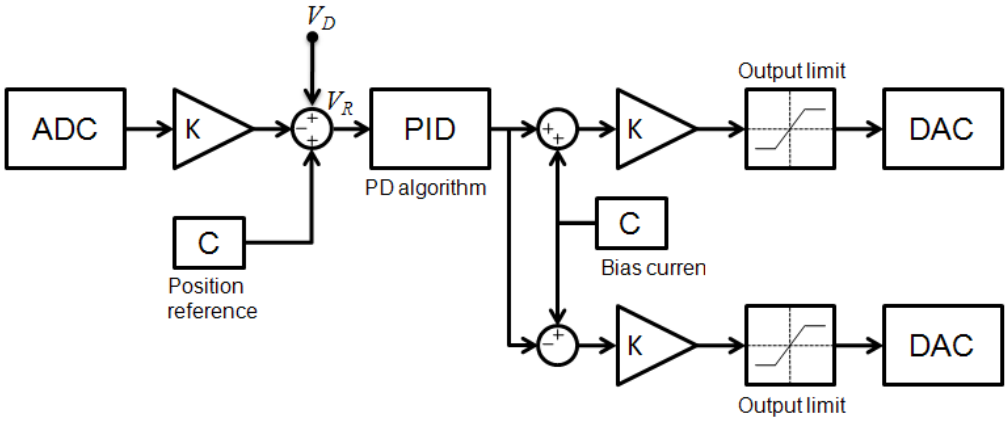


Figure 5-11: Disturbance input for sensitivity measurements

To calculate the sensitivity of the system, (5.5) was used with frequencies ranging from 0 to 2000 Hz [30].

A 1 μm sinusoidal reference was inserted into the system with increments of 1 Hz per second. The fast Fourier transform (FFT) was further used to obtain the frequency spectrum of the measured data.

Figure 5-12 shows the results obtained for the bottom radial AMB. The first three critical frequencies are clearly visible. The first is at 49 Hz (2,940 rpm), the second is at 191 Hz (11,460 rpm) and the third at 736 Hz (44,160 rpm). The sensitivity of the critical frequencies is 2.1 dB, 11.98 dB and 14.61 dB respectively. The third critical frequency provided the worst rating of 14.61 dB.

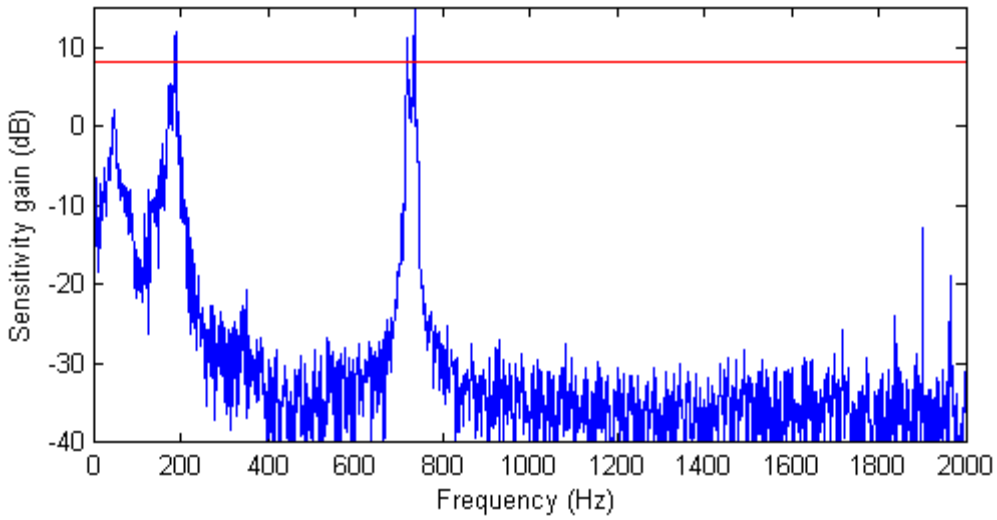


Figure 5-12: Bottom radial AMB sensitivity

The sensitivity for the top radial AMB was measured similar to the bottom AMB. A sinusoidal reference of 1 μm was inserted and the system response was recorded and analyzed. Figure 5-13 presents the results obtained for the top radial AMB. Again the first three critical frequencies are visible. The sensitivity of each critical frequency is 13.72 dB, 2.49 dB and -0.54 dB respectively.

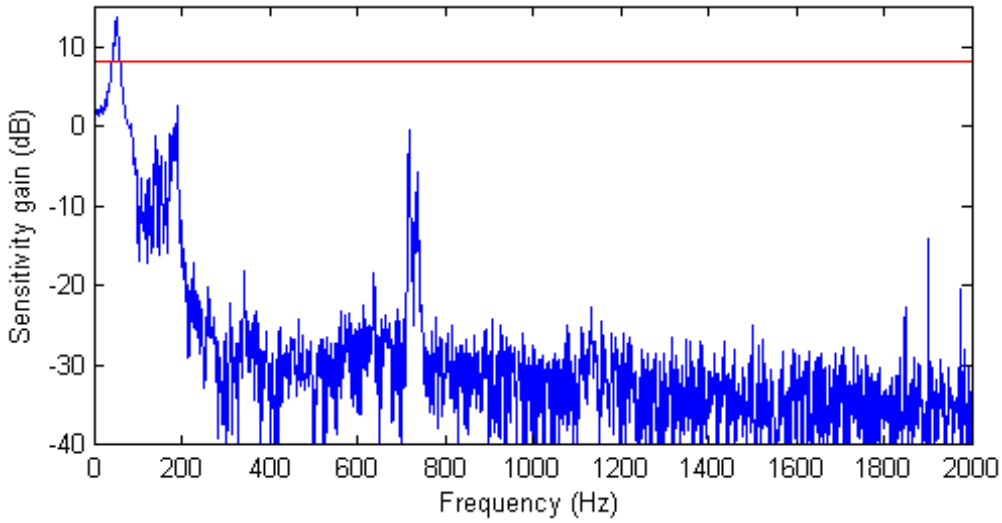


Figure 5-13: Top radial AMB sensitivity

The difference in sensitivity peak amplitudes in the top radial AMB and bottom radial AMB is due to the difference in equivalent mass. The top radial AMB experience an equivalent rotor mass of 13.22 kg and the bottom radial AMB an equivalent mass of 5.38 kg. The difference in sensitivity peaks is due to the difference in inertia of the rotor at the top and the bottom.

Figure 5-14 shows the sensitivity results obtained for the axial AMB.

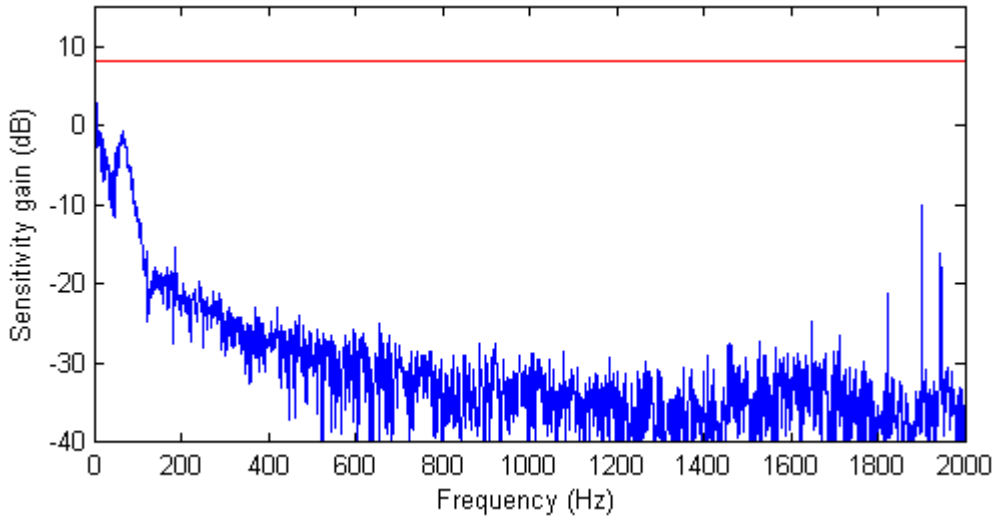


Figure 5-14: Axial AMB sensitivity

The rotor has an axial critical frequency at 65 Hz (3900 rpm) at which the sensitivity of the axial AMB is -0.73 dB.

The AMB system degraded from zone A/B with a maximum sensitivity of 8 dB, to Zone C/D with a maximum sensitivity of 14 dB. This is due to the

lowered damping and stiffness of the AMBs resulting from the 4 pole elliptic filter on the analogue to digital converter board. The damping and stiffness of the AMBs should be increased to obtain a maximum sensitivity of 8 dB to maintain the zone A/B according to ISO CD 14839-3.

5.1.4 PMSM start-up

The PMSM control was also tested and a difficulty was experienced with the generation of the PWM signals. To generate the PWM signals at a rate of 50 kHz with a resolution of 100 units, the processor should run the PWM interrupt at a frequency of 5 MHz. This was a problem for the DSP. It was mainly busy servicing the interrupt and did not get to the control of the AMBs. The problem was overcome by writing the duty cycle of each phase as a voltage on the DACs. These voltages were then converted to PWM signals, using an RT1104 dSPACE® card. The rest of the algorithm remained the same.

A ramp input of 5 rad/s^2 was inserted and the 3-phase outputs were monitored against time. Figure 5-15 shows the results obtained for the PMSM control algorithm.

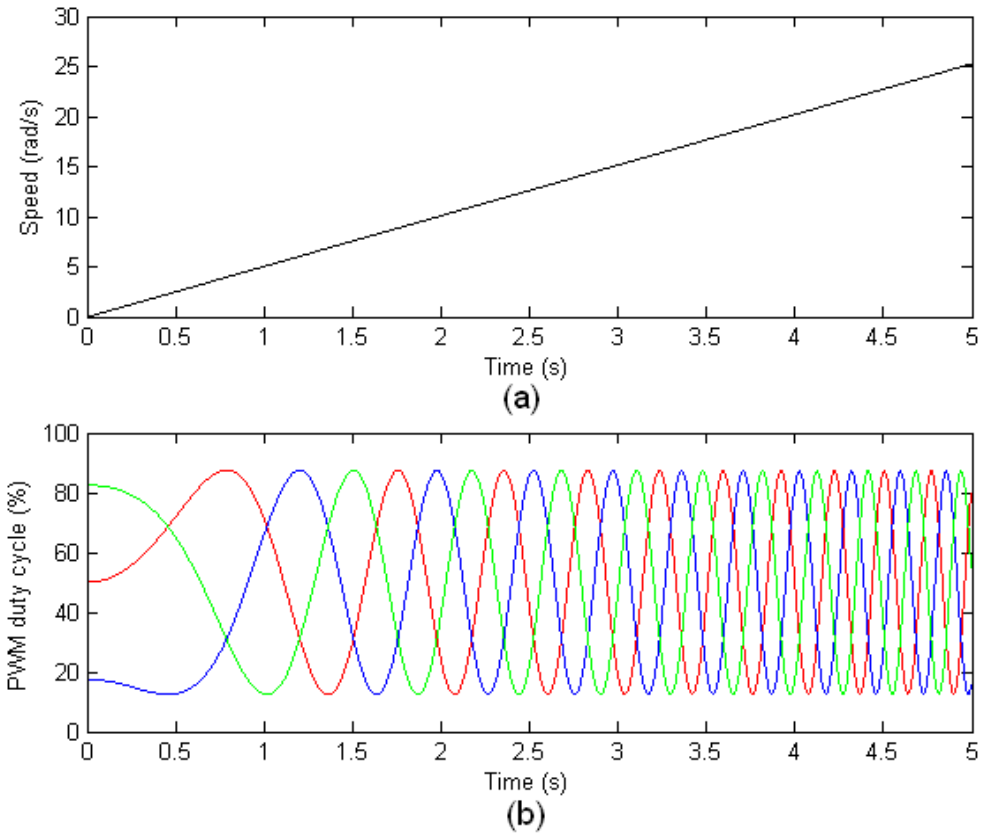


Figure 5-15: PMSM start-up curve

As the ramp input increases in figure 5-15 (a), the frequency of the three phases increases (b) and in effect the PMSM speed increases. Figure 5-15 (b) represents the duty cycle of the PWM signals fed to the 3-phase bridge.

5.2 Conclusion

The control algorithm execution time exceeded the required 50 μs, which resulted in the sample frequency adjustment from 20 kHz to 10 kHz. A 4-pole elliptic filter on the ADC board affected the stability of the system significantly. A reduction in bearing stiffness and damping occurred, which was confirmed by adding the filter to the simulation. Due to the reduction of bearing stiffness and damping, the results obtained for the sensitivity were much higher. The critical frequencies remained the same, but the maximum sensitivity increased from 8 dB to just above 14 dB. Lastly, the PMSM control worked well. The rotor's actual speed increased as the reference speed increased.

Chapter 5 included all the testing done on the SBC and the FESS. First the AMB control algorithm and SBC performance was tested. It was followed by the step response testing of the AMBs to verify the stability of each. The sensitivity analysis of the AMBs was done, followed by the PMSM control testing. Chapter 6 will further conclude on the testing results obtained and highlight the areas of future studies for this project.

6
Chapter
Conclusion and recommendation

Chapter 6 starts off with an explanation for the inconsistency in the stiffness and damping of the active magnetic bearings. A conclusion on the single board computer as well as the industrial control aspect follows. The areas for future work are then identified.

6.1 Stiffness and damping inconsistency

The stiffness and damping inconsistency on the active magnetic bearings (AMBs) is due to the 4-pole elliptic filter on the analogue to digital converters (ADCs) and the single pole filter on the digital to analogue converters (DACs). The attenuation of these 2 filters at 10 kHz is 0.98 dB as shown in figure 6-1(a). The inconsistency is caused by the attenuation, but rather the phase shift of -36 degrees at 10 kHz as shown in figure 6-1(b).

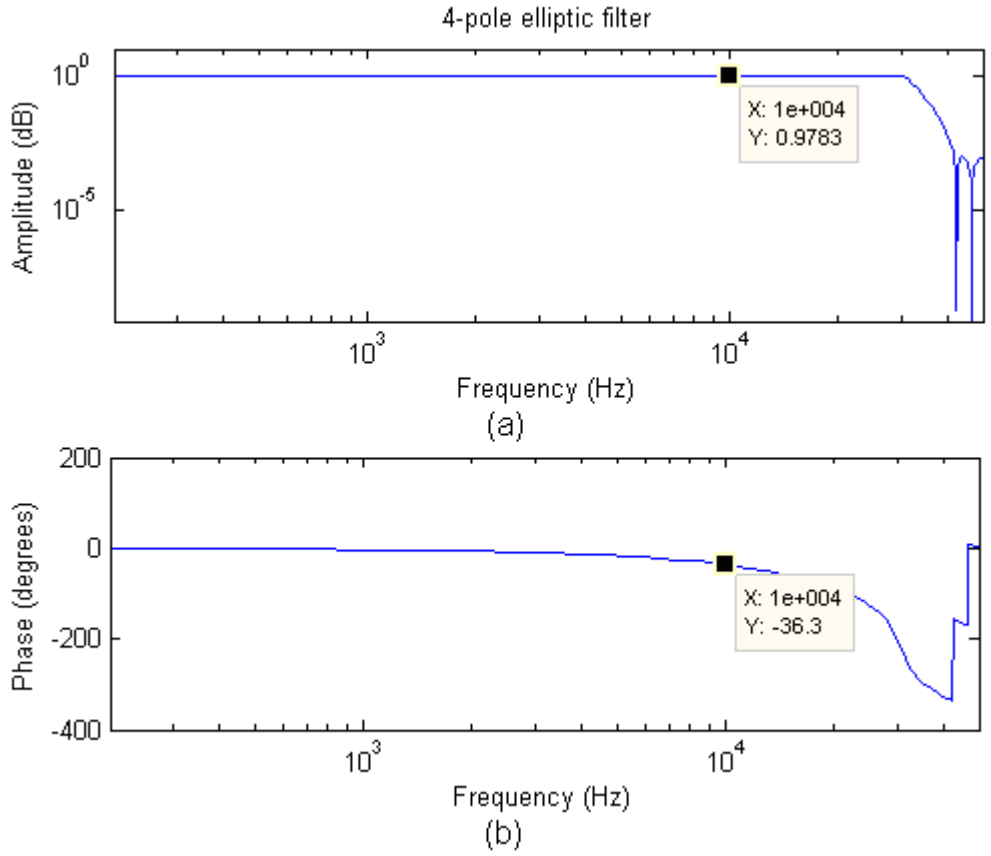


Figure 6-1: Bode diagram for the 4-pole elliptic filter

To correct the inconsistency, the order of the 4-pole elliptic filter should either be reduced or the cut-off frequency should be increased. The ADC and DAC board was sourced together with the single board computer (SBC). The filters are implemented in the analogue circuitry and not digitally, therefore the filters cannot be altered.

The actual results and simulated results correlated closely after the filters were added to the simulation. To maintain the original damping and stiffness of the AMBs, the simulation can be used to determine new values for the proportional and derivative constants. The bias current can also be altered to reduce the sensitivity of the system.

6.2 Summary

A SBC was specified and sourced in this project with the aim of controlling the AMBs. The control of the AMBs should be on an industrial level of control. The conclusion of these two factors is given in section 6.2.1 and section 6.2.2.

6.2.1 Single board computer

The SBC performed well for the application given. The performance of the digital signal processor (DSP) was tested by measuring the execution time of the control algorithms. The results verified that the performance of the processor is sufficient for the AMB control and partly for the PMSM control. To maintain the sampling frequency of 20 kHz as specified in chapter 3, part of the PMSM control algorithm could be moved to the field programmable gate array (FPGA) on the SBC. This was not included in this project due to the high level of programming complexity of VHDL to program the FPGA.

6.2.2 Industrial control

An industrial level of control was achieved by replacing the dSPACE® development control system with an SBC. The SBC made use of an ethernet connection to transfer the data from the SBC to the host PC to display the status of the flywheel energy storage system (FESS) in the control room. Control commands could also be sent back to the SBC to change the control constants for example.

The industrial control operated well for this application. The status was displayed in real-time by sending the information to the control room every 20 ms. The only disadvantage of this control architecture is that all the

sampled data cannot be sent to the control room in real-time. This would only be necessary if the user wanted to do analyses on the data, but it was not required for this project.

6.3 Future work

The control of the FESS with a SBC to achieve an industrial level of control, operated well, but there are still room for improvement on this project. The main areas for future work identified, are: the SBC, nonlinear control algorithms, digital power amplifiers, PMSM drive and the speed sensor.

6.3.1 Single board computer

The SBC worked well for the AMB control, but the performance was not sufficient for the PMSM control together with the AMB control. Specifically the DSP struggled to generate the pulse width modulated (PWM) signals required by the PMSM drive. The PWM signal could easily be generated with the FPGA, but this would require the programmer to learn VHDL.

Instead of using a DSP, a digital controller featuring PWM outputs could be used, for example the TMS320F2812. Within the McTronX Research group, dSPACE® was used to control the AMB system with 5 degrees of freedom and a PMSM. The SBC was the first new type of controller used for this AMB application on an industrial level. Although the SBC sourced, was designed to be used in many different applications, the McTronX group should develop their own SBC which would specifically fit their application.

6.3.2 Non linear control algorithms

As stated in section 2.4.2, the stable gain range for nonlinear control is much higher, the current consumption is much less and nonlinear control deals very well with large position variations. This motivated the use of nonlinear control algorithms above linear control algorithms like proportional-derivative (PD) control. A nonlinear control algorithm should be implemented on the FESS to verify a better stability as well as a sensitivity analyses.

6.3.3 Power amplifiers

One of the major problems within the FESS is the noise levels. The power amplifiers (PAs) in the system, switch 60 V, 5 A at a frequency of 36 kHz. There are 10 PAs, each generating switching noise at a frequency of 36 kHz

with no specific time reference between the PAs' noise. The noise can be reduced by synchronising the (PAs) and sampling the ADCs in synchronism. The only problem with this is that the current PAs do not have the functionality to synchronise them.

Another suggestion is to move away from the concept of sending an analogue reference from the controller to the PAs. Digital PAs could be used, which would immobilize the noise on the output channels.

6.3.4 Permanent magnet synchronous machine drive

The PMSM drive is a first generation prototype within the McTronX Research group. The reliability of the drive is still unknown. As with the PAs, the use of a digital drive should be investigated. Instead of sending 6 PWM signals to the drive, 1 digital speed reference could be sent. This would reduce the processing power required for the main controller, and the need for a digital processor that features PWM outputs. The drive would then contain all the intelligence and the PWM generators for powering the PMSM.

6.3.5 Speed sensor

The speed sensor currently used in the FESS has no connection to either the dSPACE® or the SBC. An RS485 interface is required by the speed sensor, which none of the dSPACE® or SBC feature. The SBC has an RS422 add-on board, which can be used to realise the sensing of the true speed, but was not included in the objectives of the project scope. Currently the true speed of the rotor cannot be compared to the reference speed generated by the controller. Future projects should include the sensing of the true speed via the RS422 port so that the actual rotor speed can be compared to the reference speed and a motivation on the accuracy of the voltage over frequency control can be derived.

6.4 Closure

The aim of this project was to replace the current dSPACE® control system used on the flywheel energy storage system (FESS), with an industrial level of control featuring a single board computer (SBC). This SBC had to control 5 active magnetic bearings (AMBs) and one 3-phase permanent magnet synchronous machine (PMSM). The results in chapter 5 verify that the goals for this project have been reached. The system is fully functional, except for the

sensitivity of the system that is higher than expected. Due to this, the rotor was not rotated up to its limit of 30,000 rpm. This is the first project for the McTronX Research group that is about an industrial level of control. The project will serve as a baseline for further research on industrial control of high-speed AMB systems.

References

All references used in this document are listed below.

[1] H Bleuler, C. Gähler, R. Hertzog, et al, “Application of Digital Signal Processors for Industrial Magnetic Bearings,” *IEEE transactions on control system technology*, vol. 2, no. 4, 1994, pp. 280-289.

[2] G. Schweitzer, H. Bleuler, and A. Traxler, “Active Magnetic Bearings Basics, Properties and Applications of Active Magnetic Bearings,” Zürich: Author’s Reprint, 2003.

[3] M.S. Sarma, “Electric Machines Steady-State Theory and Dynamic Performance,” 2nd ed., PSW Publishing Company, Boston, 1996, pp. 34-73.

[4] H Kim, H.C. Kim, “Modeling and control of a magnetic bearing system for the magnetically suspended centrifugal blood pump,” *The international journal of artificial organs*, vol. 23, no. 10, 2000, pp. 47-51.

[5] J.Y. Hung, N.G. Albritton, F. Xia, “Nonlinear control of a magnetic bearing system,” *Mechatronics*, vol. 13, 2003, pg. 621-637.

[6] P. Schroder, B. Green, N. Grum, and P. J. Fleming, “On-line evolution of robust control systems: an industrial active magnetic bearing application”, *Control Engineering Practice,* vol. 9, 2001, pp. 37-49.

[7] H. Chang, S. C. Chung, “Integrated design of radial active magnetic bearing system using genetic algorithms”, *Mechatronics,* vol. 12, 2002, pp. 19-36.

[8] F. M. Renner, K. J. Hoffmann, R. Markert, M. Glesner, “Design methodology of application specific integrated circuits for mechatronic systems”, *Microprocessors and Microsystems,* vol. 24, no. 2, 2000, pp. 95-103.

[9] N. Skricka, R. Markert, “Improvements of the integration of active magnetic bearings”, *Mechatronics,* vol. 12, no. 8, 2002, pp. 1059-1068.

[10] C. R. Knospe, “Active magnetic bearings for machining applications”, Control Engineering Practice, vol. 15, 2007, pp. 307–313.

[11] P. Barney, J. Laufier, J. Redmond, W. Sullivan, “Adaptive spindle balancing using magnetically levitated bearings”.

[12] Texas Instruments, Digital Signal Processing, [On-line] Available at: http://focus.ti.com/lit/ml/spry061/spry061.pdf, [Date of access 8 March 2007].

[13] Chan-Tang Hsu, Shyh-Leh Chen, “Nonlinear control of a 3-pole active magnetic bearing system”, Automatica, vol. 39, 2003, pp. 291 – 298.

[14] Thomas R. Grochmal, Alan F. Lynch, “Experimental comparison of nonlinear tracking controllers for active magnetic bearings”, Control Engineering Practice, vol. 15, 2007, pp. 95–107.

[15] Ha-Yong Kim, Chong-Won Lee, “Design and control of active magnetic bearing system with Lorentz force-type axial actuator”, Mechatronics, vol. 16, 2006, pp. 13–20.

[16] M. Parnichkun, C. Ngaecharoenkul, “Kinematics control of a pneumatic system by hybrid fuzzy PID”, Mechatronics, vol. 11, 2001, pp. 1001-1023.

[17] E. A. Parr, "Progammable Controllers: An engineer's guide", 3 ed, 2003.

[18] M. A. Laughton, D. J. Warne, "Electrical Engineer's Reference Book", 16 ed, 2003.

[19] W. Buchanan, "Computer Busses: Design and application", 1 ed, 2000.

[20] K. C. Lee, S. Lee, H. H. Lee, "Implementation and PID tuning of network-based control systems via Profibus polling network", Computer Standards & Interfaces, vol. 26, 2004, pp. 229-240.

[21] R. Hüsemann, C. E. Pereira, "A multi-protocol teal-time monitoring and validation system for distributed fieldbus-based automation applications", Control Engineering Practice, vol. 15, 2007, pp. 955-968.

[22] M. Tooley, S. Winder, "Newnes data communications pocket book", 4 ed, 2002.

[23] A. C. Fischer-Cripps, "Newnes Interfacing companion", 1 ed, 2002.

[24] C. LeBlanc, "The future of industrial networking and connectivity", Dedicated systems magazine, 2000.

[25] P. Cavill, "FPGA or DSP for military applications? Both have their place", DSP-FPGA.com Product Resource Guide, 2005.

[26] P. Warner, "Choosing DSP or FPGA for your Application", Hunt Engineering, 2002.

[27] R. Larsonneur, “Design and control of active magnetic bearing systems for high speed rotation”, PH.D Thesis, Swiss Federal Institute of Technology, Z¨ urich, March 1990.

[28] L. Xu, “Implementation and Experimental Investigation of Sensorless Control chemes for PMSM in Super-High Variable Speed Operation” IEEE, 1998.

[29] Texas Instruments, "TMS320F/C24x DSP Controllers Reference Guide: CPU and Instruction Set", SPRU160C, June 1999.

[30] S. Myburgh, "The development of a fully suspended AMB system for a high-speed flywheel application", masters, 2008.

[31] J. J. Janse van Rensburg, "Development of a flywheel energy storage system –uninterrupted power supply (FLY-UPS)", masters, 2008.

[32] A. de Klerk, "Drive implementation of a permanent magnet synchronous motor", masters, 2008.

[33] S. Jeevananthan, R. Nandhakumar, P. Dananjayan, "Inverted sine carrier for fundamental fortification in PWM inverters and FPGA based implementations", Serbian journal of electrical engineering, vol. 5, Nov 2007, pp. 171-187.

[34] B. S. Blanchard, W. J. Fabrycky, "System engineering and analyses", 4 ed, 2006.

[35] Community wireless solutions, [On-line] Available at: http://www.cuwin.net/hardware/ruggednodes, [Date of access 23 April 2008].

[36] ISO TC108/SC2/WG7, "Active magnetic bearing – Evaluation of stability margin", ISO CD148393-3, Oct. 7, 2003.

A

Appendix

Additional information

All additional information used within this thesis will be presented here.

Table A-1: Physical medium types [21]

Type	Description	Physical medium	Max Speed
Foundation Fieldbus	It is an all-digital, serial, two-way communication system. Software modules called function blocks are used. Each function block processes input data according to a specified algorithm and an internal set of control parameters and produces output data. Examples of function blocks are: analogue input/output, digital input/output and PID control.	IEC1158-2	31.25 kbps
Control Net	Makes use of concurrent time domain multiple access (CTDMA) as medium access strategy. This strategy divides time in repeating network update times (NUTs). These NUTs can be configured between 2 and 100 ms, based on the time critical nature of the data. All nodes are continuously synchronized so they know when they may access the media to broadcast their data.	Coaxial with transformer isolation	5 Mbps
Profibus	It uses a hybrid medium access mechanism with master/slave hierarchy. IEC61158 supports 2 versions: Profibus DP and PA. DP is used for interconnecting programmable logic controllers (PLCs) and peripheric devices. PA is specially designed for process control applications, allowing operation in classified areas.	RS485/ IEC1158-2	12 Mbps/ 31.25 kbps
P-Net	This industrial protocol also uses a master/slave communication scheme. It uses a medium access strategy denominated virtual token passing. It also allows a multi-net topology, where an additional network layer is specified to the protocol.	RS485	76.8 kbps
High Speed Ethernet (HSE)	Provides specifications for a cost-effective, high-speed, plant-wide network for process control using commercial off the shelf ethernet hardware and software, which include support to standard Ethernet protocols like TCP/IP, DHCP, SNTP, and SNMP.	Ethernet	100 Mbps
SwiftNet	This fieldbus uses the producer/consumer communication paradigm, allowing synchronous data transfer up to 120,000 samples of 16 bits/s. The medium access strategy is Time Division Multiple Access (TDMA). It uses synchronization messages that provide time marks to all network nodes.	RS485	5 Mbps
WorldFIP	It also uses the producer/consumer model in industrial and building automation applications. WorldFIP admits two communication forms: configured, which supports synchronous data, and on demand, that handles asynchronous events.	IEC1158-2	31.25 kbps
Interbus-S	It uses a physical connection based on ring topology, seeking cyclical in/out data exchange among field level devices. Each device occupies a predefined message slot reserved to send and receive data. The transmission in a physical medium RS485 is full duplex, allowing concurrent messages sending and receiving.	RS485	500 kbps

B

Appendix

System specification

The system specification of the flywheel energy storage system (FESS) is given in this appendix

Table of contents

List of figures

List of tables

List of abbreviations and acronyms

A	Architecture
ADC	Analogue to Digital Converter
AMB	Active Magnetic Bearing
DAC	Digital to Analogue Converter
EMC	Electro-Magnetic Compatibility
FESS	Flywheel Energy Storage System
GPIO	General Purpose Inputs/Outputs
IF	Interface
LPF	Low Pass Filter
PA	Power amplifier
PWM	Pulse Width Modulator
QA	Quality Assurance
SBC	Single-Board Computer
SP	Software Procedure
UPS	Uninterrupted Power Supply

1. Document management

1.1 Contributors

Checked by party	Individual name	Signature	Date
Author:	Mr Dewald Herbst		
Quality Assurance:	Mr Eugén Ranft		
Technical Approval:	Mr Eugén Ranft		
Project Manager:	Prof George van Schoor		

1.2 Version control

Project Title:	Single board computer based control of an AMB
Document Number:	FESS-TYPEBI-200-VA (Project-type-reference-revision)
System / Subsystem Title:	FESS (Flywheel Energy Storage System)
Document Issue Date:	2007-07-09
Client:	Prof Johann Holm
Client Reference:	FESS-TYPEBI-200-VA

1.3 Acceptance

Checked by	Name	Signature	Date
Accepted by:			

1.4 Security levels and restrictions

Level	Description	Applicable level
1	Strictly confidential – not to be distributed	
2	Company confidential – distributed inside company	X
3	Client confidential – distributed to limited clients and contractors	
4	Public domain – distributed freely	

2. Development specification scope

2.1 Identification

This specification establishes the performance, design, development and test requirements for the single-board computer used with the FESS.

2.2 Subsystem overview

2.2.1 Primary mission statement

The primary mission of the single-board computer within the flywheel energy storage system (FESS) is to actively control the position of the magnetically suspended rotor of the synchronous motor/generator.

2.2.2 Operational architecture

The single-board computer in this specification is an embedded digital signal processor (DSP) with advanced signal processing techniques. The DSP receives analogue electronic signals from the position sensors on the AMB, process the signals, and produce the necessary analogue control signal to control the position of the rotor.

The operator can only adjust the position of the rotor by means of a graphical user interface (GUI) through the personal computer. The operator should be able to view the status of the rotor via the GUI and shut down the FESS if necessary.

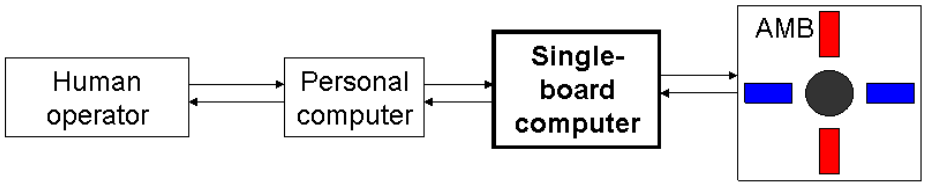


Figure 2-1: Concept diagram of the FESS

3. Applicable documents

The following documents of the exact issue shown, form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

[1] "MIL-STD 490X, Specification practices"

4. Single-board computer requirements

4.1 Single-board computer definition

A single-board computer (SBC) consists of a digital processor, digital signal converters, programmable memory, digital interface and digital input/output

signals. The composition of these components will be discussed in the following section.

4.1.1 Functional architecture and physical interfaces

The SBC consist of nine architectural functions namely: DSP, ADC, DAC, GPIO, PWM, anti-aliasing filter, LPF, JTAG, and TCP/IP. Refer to figure 4-1. The function of each will be discussed further.

- A 20.1 – The function of the DSP is to generate the necessary control signals for the ADC, DAC, GPIO, PWM, JTAG, and TCP/IP. It should also receive signals from the ADC, process the signals, and feed the result to the DAC and PWM. The DSP should be able to receive commands through the TCP/IP and JTAG.
- A 20.3 – The function of the ADC is to convert the analogue signal received from the position sensor to an acceptable digital signal. This digital signal should then be passed to the DSP.
- A 20.4 – The function of the DAC is to convert the digital signal received from the DSP to an acceptable analogue signal. This signal should then be passed on to the power amplifiers to control the rotor position.

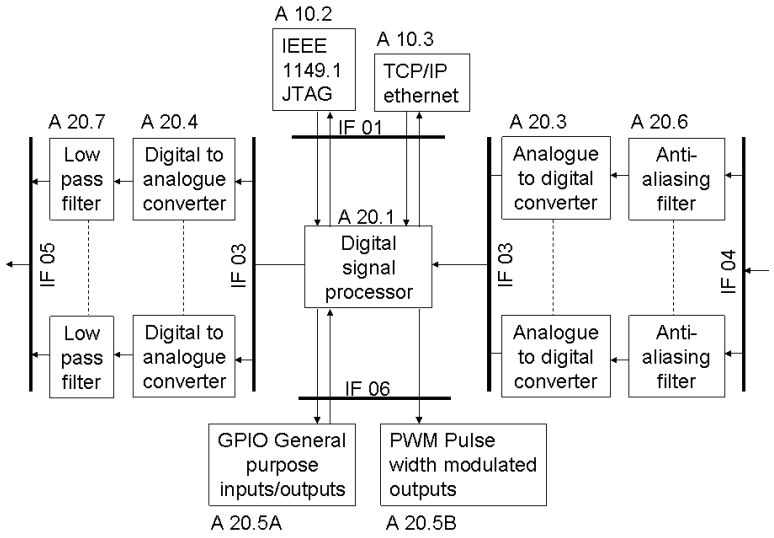


Figure 4-1: Functional architecture of a single-board computer

- A 20.5A – The function of the GPIO is to give the ability to synchronize processes and to view the status by means of LEDs.
- A 20.5B – The function of the PWM is to drive the synchronous motor/generator. Signals will be received from the DSP and converted to

an acceptable PWM signal. This signal will be passed to the synchronous motor/generator driver.

- A 20.6 – The function of the anti-aliasing filter is to filter out high frequency noise within the sensor probes. This is to ensure an accurate measurement of the rotor position. The signal from the sensors will be filtered and passed to the ADC.
- A 20.7 – The function of the LPF is also to filter out high frequency noise due to high switching frequencies within the system. The analogue signal from the DAC will be filtered and passed to the amplifiers.
- A 10.2 – The function of the JTAG is to evaluate the system if a failure occurred. This should be able even if the DSP has crashed or is in stop mode. All registries should be visible.
- A 10.3 – The function of the TCP/IP is to allow an operator to control the reference position of the rotor and shut down the system for maintenance from a control room. The operator should also be able to view the status of the rotor via the GUI on the computer. Signals will flow from the computer via the TCP/IP to the DSP and back.

4.1.2 Component list

The following table shows the components that should be included in the single-board computer. Refer to figure 1-2 for more detail.

Table 4-1: Component list

Reference	Quantity	Description
A 20.1	1	Digital signal processor
A 20.4	10	Digital to analogue converters
A 20.7	10	Low pass filters
A 20.3	10	Analogue to digital converters
A 20.6	10	Anti-aliasing filters
A 20. 5B	6	Pulse width modulated outputs
A 10.3	1	TCP/IP ethernet port
A 10.2	1	JTAG port
A 20.5A	10	General purpose inputs/outputs

4.2 Characteristics

The minimum characteristics of the components listed in the previous section, will be stated below.

4.2.1 Performance characteristics

Table 4-2: Component performance characteristics

Reference	Description	Characteristic
A 20.1	Digital signal processor	32-Bit floating point DSP, ≥58 MIPS
A 20.4	Digital to analogue converters	16-Bit, ≥ 10 ksps, +/-15 volt output, ≥ 10 mA per output
A 20.7	Low pass filters	-3dB set at 50 kHz, single pole filter, S/N ≥ 40 dB
A 20.3	Analogue to digital converters	16-bit, ≥ 10 ksps, +/-15 volt differential, ≥ 5 MΩ||5 pF input impedance
A 20.6	Anti-aliasing filters	3dB set at 30 kHz, elliptic filter, S/N ≥80 dB
A 20. 5B	Pulse width modulated outputs	0/5 volt, ≥ 20 mA per output, ≥10 MHz cycle time
A 10.3	TCP/IP Ethernet port	10BASE-T/100BASE-T standard ethernet protocol, RJ-45 connector
A 10.2	JTAG port	Standard JTAG 1149.1 compliant emulation port debugger support.
A 20.5A	General purpose inputs/outputs	0/5 volt, ≥ 20 mA per output, ≥ 5 MHz switching time

4.2.2 Physical characteristics

The SBC should consist of all the components listed. The complete SBC size should be a single euro size (3U 160 mm x 100 mm). The weight should not exceed 500 g. Refer to table 4-3 for more detail.

4.2.3 Reliability

The SBC should be operational with a total down time of less than 3 days per annum.

4.2.4 Maintainability

The SBC should have maintenance down time of less than 8 days per annum for all types of maintenance. Firmware maintenance should be possible without shutting the SBC down.

4.2.5 Environmental conditions

The SBC should operate in the following environmental conditions:

- -10 to 55ºC
- 10 to 65% humidity

- Withstand shocks of up to 5G

Refer to table 4-3 for more detail.

Table 4-3: Component physical characteristics

Reference	Description	Physical		Environmental		
		Weight	Size	Temperature	Humidity	Shock
A 20.1	Digital signal processor	20 g	20 mm x 20 mm	5ºC - 57ºC	10% - 55%	5G
A 20.4	Digital to analogue converters	10 g	10 mm x 10 mm	5ºC - 51ºC	10% - 65%	6G
A 20.7	Low pass filters	20 g	10 mm x 20 mm	5ºC - 65ºC	7% - 60%	4G
A 20.3	Analogue to digital converters	10 g	10 mm x 10 mm	5ºC - 51ºC	10% - 65%	6G
A 20.6	Anti-aliasing filters	30 g	10 mm x 30 mm	5ºC - 65ºC	7% - 60%	4G
A 20. 5B	Pulse width modulated outputs	50 g	10 mm x 40 mm	0ºC - 70ºC	5% - 80%	7G
A 10.3	TCP/IP Ethernet port	50 g	20 mm x 30 mm	0ºC - 65ºC	7% - 75%	6G
A 10.2	JTAG port	30 g	10 mm x 30 mm	0ºC - 65ºC	7% - 80%	7G
A 20.5A	General purpose inputs/outputs	50 g	10 mm x 40 mm	0ºC - 70ºC	5% - 85%	8G

4.3 Design and construction requirements

4.3.1 Materials, processes, and parts

Materials and components shall be used that are commonly obtainable from secondary suppliers within South Africa.

4.3.2 Electromagnetic radiation

The EMC should comply with the government regulations of South Africa.

4.3.3 Nameplates, and product marking

No specific requirements.

4.3.4 Workmanship

No specific requirements.

4.3.5 Interchange ability

No specific requirements.

4.3.6 Safety

High voltage and high current carrying conductors shall be isolated to prevent shock from direct touch.

4.4 Documentation

A circuit layout, test plan procedure, maintenance plan as well as an installation instructions document will be drawn up.

4.5 Maintenance and logistics

4.5.1 Maintenance

Maintenance and repairs shall be performed by the McTronX research group at the NWU. If maintenance and repairs requires unavailable skill and equipment, help shall be sought from technical documentation from section 1.3.4.

4.5.2 Supply

The subsystem shall be in use for no less than a period of 10 years before significant phase-out or upgrades may be performed. As a result:

- All components in the system must be supported over a minimum period of 10 years.
- New components may be introduced over time. Provision shall be made for new items to replace obsolete items.
- Provision shall be made for supply of components by means of a third party manufacturer and complete manufacturing data pack, should the initial supplier not be able to proceed with supply.

4.5.3 Major component characteristics

Refer to section 4.2.1

Quality assurance

5.1 General

The quality assurance (QA) will be done by an external company. All given specifications will be verified.

5.1.1 Responsibility for tests

No specific requirements.

5.1.2 Special tests and examinations

Table 5-1: Acceptance test matrix

Requirement	Acceptance requirement					Method and criteria
	1	2	3	4	5	
Architecture in 1.3.1	x			x		Refer to method/value
Characteristics in 1.3.2	x	x		x		Refer to method/value
Construction in 1.3.3	x	x				Refer to method/value
Documentation in 1.3.4				x		Refer to method/value
Maintenance in 1.3.5				x		Refer to method/value

The following acceptance methods shall be applicable:

1. Demonstration: The operation of the system, or a part of the system, that relies on observable *functional* operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
2. Test: The operation of the system, or a part of the system, using instrumentation or other special test equipment to collect data for later *analysis and reporting*.
3. Analysis: The *processing* of accumulated data obtained from other qualification methods. Examples are reduction, interpolation, or extrapolation of test results.
4. Inspection: The *visual* examination of system components, documentation, etc.
5. Special qualification methods. Any special *qualification* methods for the system, such as special tools, techniques, procedures, facilities, acceptance limits, use of standard samples, pre-production or periodic production samples, pilot models, or pilot lots.

   The following sub-categories may be applicable:

   A. Computer software test

B. Internal qualification test (at factory, not acceptance test)

C. External qualification test (at independent third party, not acceptance test)

5.2 Quality conformance inspection

No specific requirements.

C

Appendix

CD

All data sheets and product manuals are available on the CD.

C.1 SBC6713e datasheets

C.2 SBC firmware

C.3 GUI software

C.4 Measurements

C.5 References

C.6 Photos

C.7 Dissertation