Embedded controller for a fully suspended active magnetic bearing system

D.E. Vogel B.Eng.

Dissertation submitted to

The School of Electrical, Electronic and Computer Engineering

in partial fulfilment of the requirements for the degree

Master of Engineering

in Computer and Electronic Engineering

at North-West University

Supervisor:	Prof. G van Schoor
Assistant Supervisor:	Mr. E.O. Ranft

June 2006

Potchefstroom Campus

Declaration

I hereby declare that all the material incorporated in this dissertation 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: ————————
Deon Edward Vogel

Summary

The industrial application of active magnetic bearings is expanding. This expansion is a driving force in the integration of AMBs. The Magnetic Bearing Modelling and Control (MBMC) research group in the School of Electrical, Electronic and Computer Engineering, North-West University is accordingly compelled to expand their research to the application of embedded control systems. The aim of this study is to develop an embedded controller for an active magnetic bearing in order to establish a DSP platform for future research in embedded control systems.

The embedded controller developed during this study is required to be capable of actively controlling a spindle with a rotaional speed of 60 000 rpm. It is further required that the embedded controller is capable of stand-alone operation, scalable in terms of the number of axes controlled and flexible in terms of the control algorithm implementation.

A TMS320F2812 DSP is selected for its processing speed, on-chip peripherals and available development tools such as the eZdsp® TMS320F2812 DSP Starter Kit, VisSim® Embedded Controls Developer and Code Composer Studio®. The interface of the embedded controller is designed for an existing double radial AMB model, which allows for the performance of the embedded controller to be compared to the existing PC-based controller.

The AMB system exhibits a slightly higher second order equivalent stiffness and damping when using the embedded controller as opposed to the existing PC-based controller. The AMB system is also slighlty less sensitive when using the embedded controller.

This embedded controller establishes a DSP platform which can be used for further research into embedded control systems and advanced control algorithms. The knowledege gained and controller developed for this study serves as essential stepping stones towards the ultimate goal of AMB integration through the progression from a DSP to an FPGA and eventually an ASIC.

Opsomming

Die industriële aanwending van aktiewe magnetiese laers (AMLs) is besig om uit te brei. Hierdie uitbreiding dien as 'n dryfveer vir die integrasie van AMLs. Die Magnetiese Laer Modellering en Beheer navorsingsgroep in die Skool vir Elektriese, Elektroniese en Rekenaar Ingenieurswese aan die Noordwes Universiteit is dienooreenkomstig gemotiveer om hulle navorsing uit te brei na die toepassing van ingebedde beheerstelsels. Die doel van hierdie studie is dan om 'n ingebedde beheerder vir 'n AML te ontwikkel en so 'n DSP platform vir navorsing in ingebedde beheerstelsels daar te stel.

Die ingebedde beheerder wat gedurende hierdie studie ontwikkel word moet daartoe in staat wees om 'n spil met 'n rotasiespoed van 60 000 opm aktief te beheer. Dit moet verder daartoe in staat wees om selfstandig bedryf te word, skaleerbaar te wees in terme van die aantal asse wat beheer word en aanpasbaar wees in terme van die beheeralgoritme implementering.

'n TMS320F2812 DSP is gekies op grond van die verwerkingspoed, ingebedde randapparatuur en die ontwikkellingshulpmiddels wat daarvoor beskikbaar is soos die eZdsp® DSP Starter Kit, VisSim® Embedded Controls Developer en Code Composer Studio®. Die koppelvlak van die ingebedde beheerder is ontwerp vir 'n bestaande dubbel radiale aktiewe magnetiese laer, wat dit moontlik maak om die gedrag van die ingebedde beheerder te vergelyk met die van die bestaande rekenaar-gebasseerde beheerder.

Die AML stelsel toon 'n geringe toename in tweede orde ekwivalente styfheid en demping wanneer die ingebedde beheerder gebruik word in vergelyking met die rekenaar-gebasseerde beheerder. Die AML stelsel is ook 'n bietjie minder sensitief wanneer die ingebedde beheerder gebruik word.

Die ingebedde beheerder slaag in die doel om 'n DSP platform daar te stel vir verdere navorsing in ingebedde beheerstelsels en gevorderde beheeralgoritmes. Die kennis opgedoen tydens hierdie studie dien as noodsaaklike boustene vir die oorkoepelende doel van AML integrasie, deur die progressie van 'n DSP na FPGA en uiteidelik 'n ASIC.

Acknowledgements

Firstly I would like to thank THRIP and M-Tech Industrial for their sponsorship of this study. Without their support, it would certainly not have been possible.

I would also like to extend my utmost appreciation and thanks to the following people:

- My supervisors, for their guidance and motivation throughout this study.
- My parents, for their unconditional love and support.
- My friend André for proofing this dissertation and much needed coffee breaks.
- My friend Morné for his ideas and meaningful inputs.
- My friends Kenny, Kobus, Markus, Sois and Charl.

This dissertation is typeset with LaTeX using the LyX document processor in conjunction with a BibTeX reference library.

Hofstadter's Law:

It always takes longer than you expect, even when you take into account Hofstadter's Law.

- Douglas Hofstadter, *Gödel, Escher, Bach*, 1979.

Contents

Declaration

I hereby declare that all the material incorporated in this dissertation 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: ——————————
Deon Edward Vogel

List of Figures

List of Tables

List of Abbreviations

AAF	Anti-Aliasing Filter
AC	Alternating Current
ADC	Analogue to Digital Converter
AIF	Anti-Imaging Filter
AMB	Active Magnetic Bearing
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital to Analogue Converter
DC	Direct Current
DOF	Dergree Of Freedom
DSP	Digital Signal Processor / Processing
EMC	Electromagnetic Compatability
ESD	Electrostatic Discharge
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
I/O	Input/Output
LED	Light-Emitting Diode

LSB	Least Significant Bit
MFLOPS	Million Floating Point Operations Per Second
MIPS	Million Instructions Per Second
Mbps	Mega bits per second
Msps	Mega samples per second
PC	Personal Computer
PID	Proportional-Integral-Derivative
PWM	Pulse Width Modulation
RAM	Random Access Memory
rpm	Revolutions per minute
SPI	Serial Peripheral Interface
TTL	Transistor-Transistor Logic
XINT	External Interrupt
ZOH	Zero-Order Hold

List of Symbols

A_{max}	Maximum passband attenuation (dB)
A_{min}	Minimum stopband attenuation (dB)
b_{eq}	Second order equivalent damping (Ns/m)
f_a	Signal bandwidth (Hz)
f_s	Sampling frequency (Hz)
FS	ADC full scale converted code
K_d	PID derivative gain
k_{eq}	Second order equivalent stiffness (N/m)
K_i	PID integral gain
K_p	PID proportional gain
PO	Percentage overshoot
T	Sampling period (s)
T_d	Damped oscillation period (s)
V_{in}	Input voltage
$V_{in_{max}}$	Maximum input voltage
V_{out}	Output voltage
$V_{out_{max}}$	Maximum output voltage
ω_d	Damped oscillation frequency (rad/s)
ω_P	Passband frequency (rad/s)
ω_n	Natural frequency (rad/s)
ω_S	Stopband frequency (rad/s)
ω_{wp}	PID differentiator pole frequency (rad/s)

Chapter 1

Introduction

This chapter provides some basic information on the operation of AMBs, as well as some introductory information on digital control and the motivation behind an embedded controller for active magnetic bearings. It also includes the problem statement and a description of the research method to be followed.

1.1 Background on AMBs

Active magnetic bearings are used by industry to suspend high-speed rotating machines. Applications of AMBs range from energy storage flywheels to high-speed spindles and turbo machinery. AMBs are primarily used for these applications because of their contact free operation. AMBs are further capable of dynamically adjusting their characteristics such as the stiffness and damping.

1.1.1 Basic operating principle

Active magnetic bearings operate on the principle that an attraction force is generated in the air gap of an electromagnet. The force attempts to reduce the reluctance by closing the air gap and is nonlinear in nature [1]. Closed-loop control is therefore needed to realise stable, contact free suspension of the rotor in mid-air.

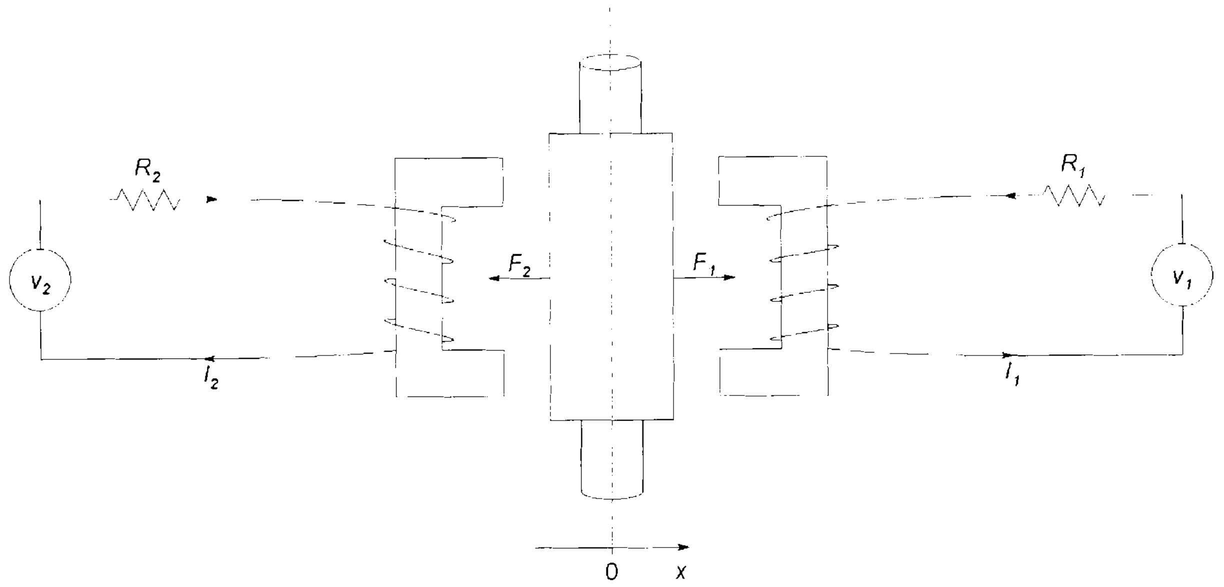


Figure 1.1: A simple 1 DOF AMB [3]

A simple one degree of freedom AMB is shown in Figure 1.1 in order to demonstrate the use of the electromagnets. By using two electromagnets opposing each other, the AMB is capable of producing both positive and negative forces on the rotor. This enables the force of the AMB to be linearised if the magnets are operated in differential mode [2] and improves the controllability of the AMB.

1.1.2 Digital control

Traditionally, AMBs were controlled by means of analogue PID controllers. These controllers had to be tuned for each application [4]. Digital control is now possible through advances in DSP technology, facilitating the implementation of advanced control methods that are more flexible, as some algorithms can even incorporate on-line training [5].

Digital controllers have various advantages. They enable designers to implement flexible control systems with optimal control strategies in order to obtain a higher stiffness than was previously achievable with analogue PID controllers [6].

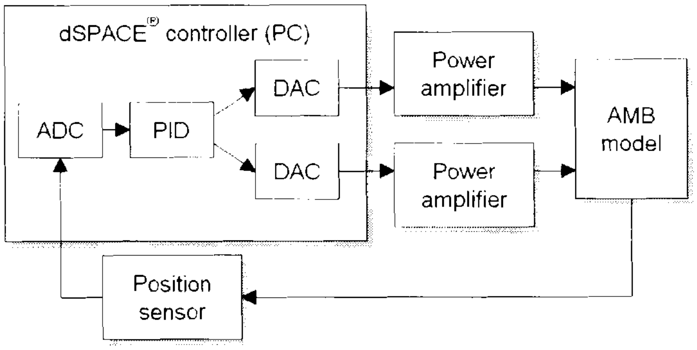


Figure 1.2: Current PC-based AMB controller

By implementing digital controllers even the non-linearity of the system can be compensated for. The force of the magnetic bearing can thus be linearised over virtually the entire working range of the AMB, without using large bias currents [7].

Digital controllers have also realised the possibility of condition monitoring. The AMB can then diagnose the entire rotor's integrity as well as the integrity of the bearing itself [8].

1.1.3 Integration

Digital control is currently implemented with component-based systems [9]. The AMB and the controller are therefore two separate units. An example of this is the existing double radial AMB model in the School for Electrical, Electronic and Computer Engineering at North-West University [10]. Figure 1.2 shows a block diagram of the PC-based controller which utilises a dSPACE® expansion card. The expansion card includes an on-board DSP as well as multiple digital and analogue inputs and outputs. The controller's software is generated using Matlab® Simulink®, which simplifies code development. Although the controller is very versatile, it is also a expensive piece of equipment.

Greater integration will enable AMBs to be more flexible in their application and will therefore expand the use of AMBs. Implementing an embedded control system using a DSP-based controller is the first step to a higher level of integration [7].

1.1.4 DSP technology

DSP technology has now evolved to such a point that it can be used to implement complex motor control algorithms within a graphical programming language [11]. This enables the developer to develop complex, optimised, reliable DSP code without an extensive knowledge of assembler or even C programming.

Research into this field is therefore necessary to establish expertise in the field of DSP which will aid in the development of an integrated active magnetic bearing.

1.2 Problem statement

The purpose of the study is to realise an experimental development model of a controller with the following functionality for AMB control applications:

1. Embedded DSP-based implementation
2. Standalone operation
3. Scalable platform in terms of the number of axes to be controlled
4. Flexible in terms of the control algorithm used and specific application
5. High bandwidth
6. Capable of implementing additional functions such as condition monitoring

There are two specific applications for which the embedded controller is earmarked, the first being the implementation of the current PID control algorithm on the double radial AMB model. Secondly, the controller should also be capable of accurately controlling other high-speed AMBs such as high-speed spindles. The major difference between the mentioned applications, are the required bandwidth. For the double radial AMB model, the system will need to actively control disturbances with a maximum frequency of 500 Hz, while the system will need to actively control disturbances with a maximum frequency of 1 kHz for the high-speed spindle.

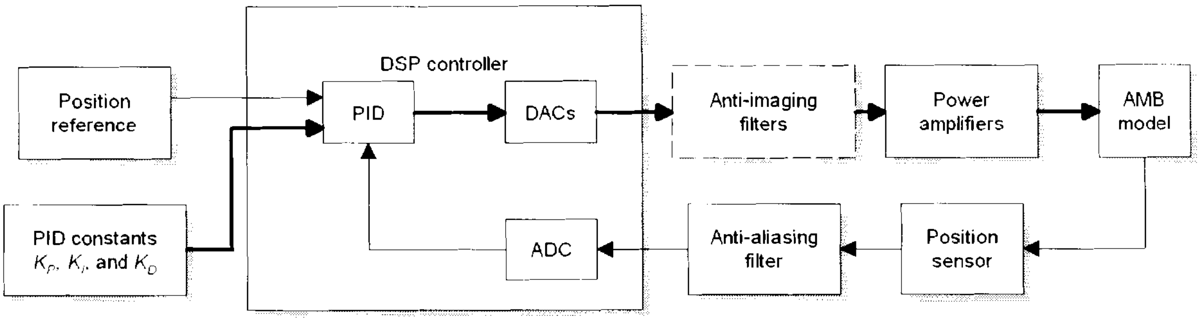


Figure 1.3: Embedded AMB controller

The embedded controller will thus greatly contribute to the AMB research infrastructure of the School for Electrical, Electronic and Computer Engineering and plays a crucial role in future high-speed AMB development.

1.3 Issues to be addressed and methodology

The first step in developing the embedded controller is compiling a detailed specification, listing all the relevant requirements and constraints of the controller. The appropriate DSP platform, chip, development board and programming environment can then to be selected. Once this is done, the interfacing requirements between the current model and embedded controller can be investigated and satisfied.

A block diagram of the embedded controller is shown in Figure 1.3. It shows the anti-aliasing filter used to bandwidth-limit the input signal, as well as an anti-imaging filters which can be used to bandwidth-limit reference signals for the power amplifiers if necessary. An AMB's transfer function is inherently low-pass in nature, which usually makes the use of an anti-imaging filter unnecessary [2].

The interface circuits have to scale the signals to the appropriate ranges in order to maximise the resolution of the signals and enable the controller to interface with the double radial AMB model. The transfer characteristics of the input and output circuits have to be verified before implementing them in the AMB model.

Once the embedded controller can be safely interfaced with the double radial AMB model, the existing PID control algorithm can be implemented using the chosen development environment and compiled for the chosen DSP chip. The computational requirements of the control algorithm can then be determined. When the embedded controller is fully integrated, the performance of the system can practically be verfied.

Each of the mentioned aspects are now discussed in detail.

1.3.1 Specification

Firstly the relevant requirements and constraints of the embedded controller are specified, including required system bandwidth. This specification will be used to evaluate the successful implementation of the controller.

1.3.2 Platform selection

The appropriate DSP platform is selected with the specifications in mind. This is done by selecting the appropriate DSP manufacturer, DSP series, DSP chip, development board and programming environment.

For DSP selection, processing speed, word length, peripherals, cost and availability has to be considered. In selecting the programming environment, cost, availability, features and ease of integration have to be evaluated while keeping the selected DSP development board in mind. The selected hardware and software is then sourced.

1.3.3 Interfacing

Both the input and output interfacing requirements need to be considered. Each is subsequently discussed.

1.3.3.1 Input

As previously stated, the DSP must be safely interfaced with the position sensors of the double radial AMB model. This is accomplished by scaling the input signals to match the full scale of the DSP's A/D converter in order to minimise the effect of quantisation errors. The DSP also needs to be protected from over-voltage and ESD phenomena.

Further, the input from the position sensors have to be band-limited in order to prevent aliasing. This is accomplished by designing the filter so as to provide adequate attenuation at the Nyquist frequency. The filter also cannot distort the control signal.

1.3.3.2 Output

The double radial AMB model's power amplifiers are designed to generate their own PWM signal and need only a current reference from the controller. The selected DSP therefore has to incorporate an DAC to generate the necessary current reference. As various motor-control DSPs do not incorporate DACs, it should also be considered to low-pass filter a PWM output of the DSP, which would therefore necessitate the use of an anti-imaging filter.

Further, the output has to be scaled to match the current reference levels expected by the power amplifiers.

1.3.4 DSP algorithm

The next step would be to implement the existing control algorithm on the embedded controller. This has to be done by firstly verifying the AMB's dynamic performance by simulation and finally implementing the control algorithm on the DSP itself.

1.3.4.1 Simulation

In order to verify the influence of the filters on the control algorithm, the AMB must first be simulated with the analogue filters, ADC, DAC, etc. implemented in software. Upon successful verification of the system's dynamic performance, the control algorithm can be implemented.

1.3.4.2 DSP implementation

After successful simulation, the control algorithm can be implemented on the embedded controller with relative ease. Once the control algorithm is implemented on the DSP, the performance of the AMB system can be verified.

1.3.5 System verification

The verification of the system should be done incrementally. Firstly, the transfer characteristics of the interface should be verfied before the embedded controller is intergrated. If the transfer characteristics are acceptable, the embedded controller can be integrated.

If the embedded controller has been fully integrated, the closed-loop system performance of the embedded controller can be verfied by determining the system sensitivity, as well as the second order equivalent stiffness and damping.

1.4 Dissertation overview

Chapter 2 contains a detailed literature study on the aspects involved in designing a digital control system. It starts off with a discussion of digital control systems in general and proceeds to focus on digitally controlled AMBs specifically. It then discusses various issues regarding analogue to digital conversion, digital to analogue conversion, controller cycle time, digital number representation and filter selection.

In Chapter 3, a detailed description of the embedded controller's design is given. It starts off with a discussion of the design process followed and then discusses the system specification. Hereafter the DSP platform and development tools are selected. The system interfaces are then designed and the AMB system is simulated in its entirety.

The design implementation is discussed in Chapter 4, starting with a description of the interface implementation. The DSP firmware implementation is also discussed in detail.

The overall system performance is evaluated in Chapter 5 by verifying the interface characteristics and analysing the system's sensitivity and step response. From the step response, the equivalent second order stiffness and damping are determined. The system performance in also compared to the existing dSPACE® controller.

Chapter 6 gives a critical account of the insights gained during the development of the embedded controller. It discusses aspects such as interfacing and firmware implementation. Some ares are then identified which require further investigation.

Chapter 2

Literature Study

This chapter covers various issues regarding digital control systems. It starts off with a basic, generic digital control system and then discusses issues regarding analogue to digital conversion, digital to analogue conversion, the processor itself and concludes with a section on filter selection.

2.1 A digital AMB control system

A basic, generic digital control system consists of an analogue to digital converter (ADC), a digital processor, a digital to analogue converter (DAC) and a feedback sensor, as depicted in Figure 2.1. The ADC converts the analogue output of the feedback sensor into a digital format which can be used by the digital processor. The processor compares the desired (reference) state of the process to the actual state, as converted by the ADC. It then determines the action required to bring the process to the desired state, according to the control strategy used. This action is then communicated to the process using the DAC and the result is measured by the feedback sensor.

An AMB system is no different to the generic control system described above, although it imposes some unique requirements. Usually the time constant of a plant is relatively long, so the control system is not affected by the time delays imposed by the controller. An AMB system, on the other hand, is very sensitive to time delays in the control loop [2, 12]. In some instances, even a time delay of 120 μs cannot be tolerated. The AMB system is therefore classified as a real-time system.

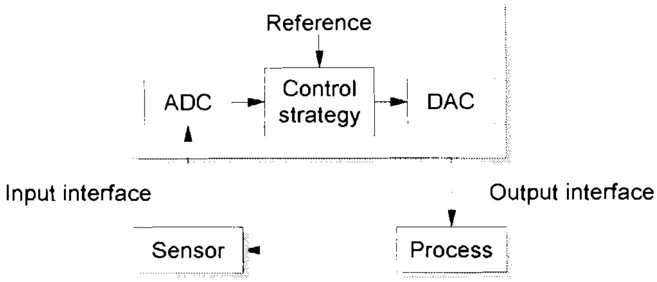


Figure 2.1: A simple digital control system

In order to successfully implement a digital control system, the following aspects have to be considered carefully [2]:

- Analogue to digital conversion,
- Digital to analogue conversion,
- The control law,
- Sampling rate, and
- Number representation within the processor.

Analogue to digital conversion and digital to analogue conversion are discussed separately in the following sections. This is followed by a discussion of the control law, sampling rate, and number representation in the section covering processor concerns.

2.2 Analogue to digital conversion

When an analogue signal is to be converted into a digital, sampled signal, it is important to consider the impact the conversion will have on both the amplitude and frequency content of the signal [13]. This determines the signal conditioning necessary at the input interface and determines the accuracy to be expected of the analogue to digital conversion process.

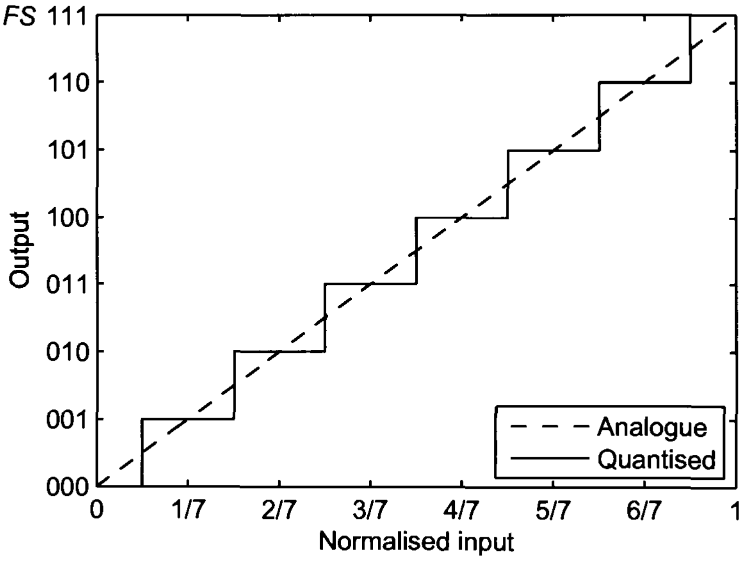


Figure 2.2: Quantisation of an ideal 3-bit ADC

2.2.1 Amplitude

In order to illustrate the effect of analogue to digital conversion on the amplitude of the signal, the output of an ideal 3-bit unipolar ADC is shown in Figure 2.2. An ideal unipolar ADC makes it first transition at $\frac{1}{2}$ LSB and at every LSB after that until it has reached full scale (FS). This causes a quantisation error of maximum $\frac{1}{2}$ LSB [13].

In an AMB system, the $\frac{1}{2}$ LSB error is usually negligible when a 12-bit ADC is used. However, the error is amplified if the converted signal's voltage range does not match the ADC's voltage range. It is therefore important to match the maximum input voltage to the FS voltage of the ADC [2].

There are four types of DC errors which could also severely increase the quantisation error; each discussed in turn in the following paragraphs [13, 14].

Gain error: The gain error causes the real converted value to be a constant times the ideal converted value if it is the only error which occurs.

Offset error: The offset error causes the real converted value and ideal converted value to differ by a constant value if it is the only error which occurs.

Integral linearity error: The integral linearity error is analogous to the linearity error of an amplifier. It occurs as the real transfer characteristic of the ADC differs from a straight line.

Differential nonlinearity error: The differential nonlinearity error relates to the linearity of the code transitions of the ADC. This type of error occurs when a change of 1 LSB in the analogue signal causes the converted value to stay the same, to skip a code or even to go back to a smaller code.

The gain and offset errors can be alleviated by the user through compensation in the DSP software, but the integral linearity and differential nonlinearity errors are intrinsic to the ADC and should be specified by the manufacturer as it determines the performance that can be expected from the ADC [13].

2.2.2 Frequency content

The effect of analogue to digital conversion on the frequency content of the signal is now considered. In this section it is assumed that no quantisation errors occur. Suppose an analogue signal has a bandwidth of f_a, having no frequency component higher than f_a, and is sampled at a frequency of f_s. According to the Nyquist sampling theorem [15], all the information within the signal can only be preserved if

$$f_s \geq 2 \times f_a \tag{2.1}$$

An effect called aliasing occurs if the signal is sampled at a frequency f_s with $f_s < 2 \times f_a$, as shown in Figure 2.3. From the figure it can be seen that the high-frequency signal is now perceived to be a signal with a frequency less than $\frac{f_s}{2}$. It is therefore necessary to limit the bandwidth of the analogue signal to prevent aliasing. This can be done using a combination of the following approaches, namely:

- Increasing the ADC's sampling rate,

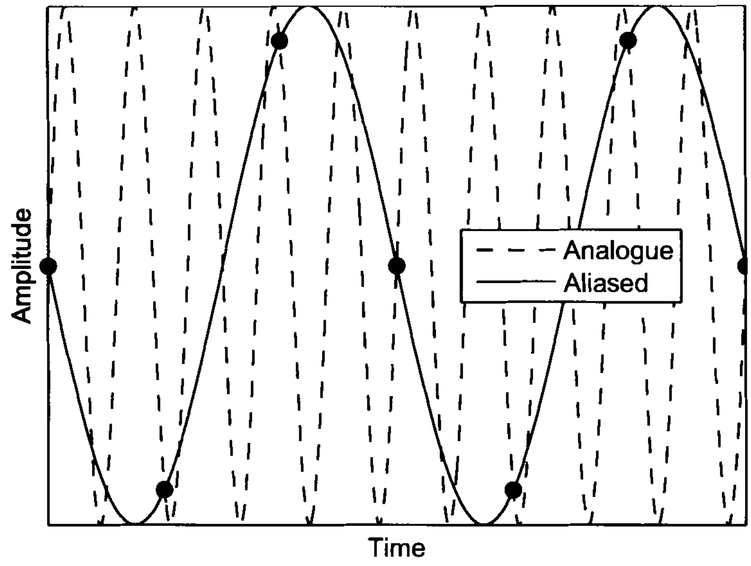


Figure 2.3: Aliasing caused by undersampling

- Synchronous sampling, and
- Analogue filtering.

Increasing the ADC's sampling rate increases the bandwidth of the signal which can be digitised. Sometimes the sampling rate of the ADC can be far greater than $\frac{f_a}{2}$. The signal can then be filtered digitally and decimated in time. As the filter output and decimation can take long to compute, this approach can pose a problem in real-time applications like the AMB system and is therefore not considered further [16].

Synchronous sampling can only be used if the frequency components higher than $\frac{f_s}{2}$ occur at specific intervals, as is illustrated in the following example:

Figure 2.4(a) shows the switching noise of a bi-state power amplifier controlled by a DSP's PWM outputs. The switching noise has a 10 MHz frequency component occurring at a frequency of 100 kHz (the PWM frequency). If the DSP's ADC is used to sample the signal at 20 kHz and the ADC is not synchronised with the PWM, the switching noise will appear as an aliased signal as shown in Figure 2.4(b). If the DSP's ADC is synchronised with the PWM, the ADC can then

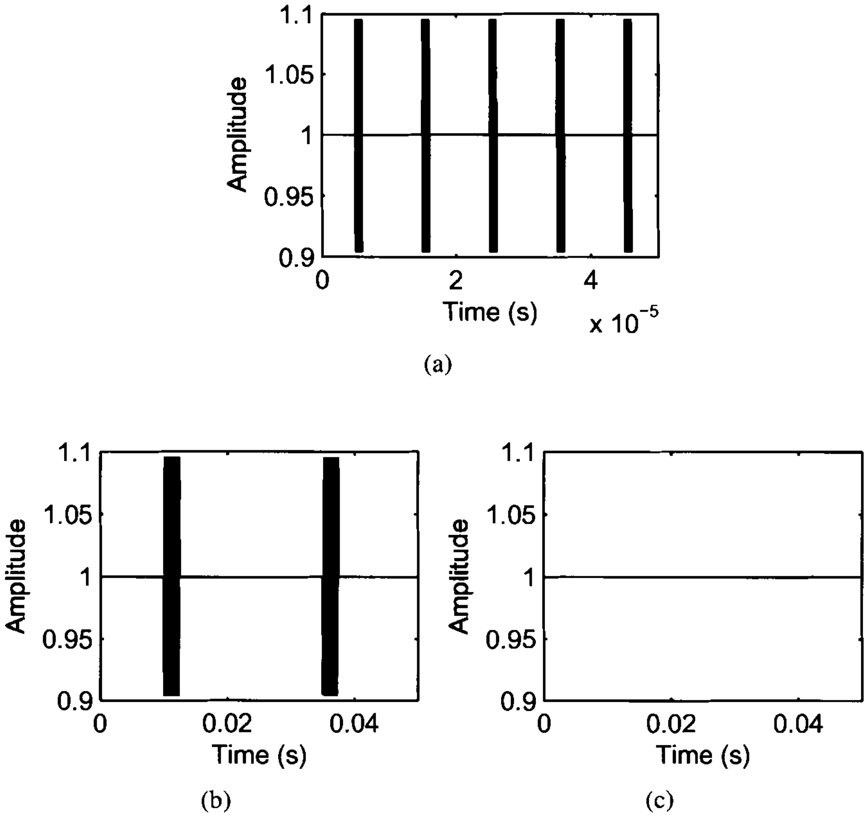


Figure 2.4: Synchronous sampling

(a) Signal with 10 MHz noise (b) Asynchronous sampling
(c) Synchronous sampling

sample between the occurrences of the switching noise. The effect of the switching noise is then effectively eliminated in the time domain as seen in Figure 2.4(c).

Whenever aliasing signal components occur randomly in time, the only way to proceed is to eliminate them in the frequency domain. This is done by analogue low-pass or band-pass filtering [17]. Usually, a high-order (8-12 poles) "brick-wall" filter is used, but the group delay of the high-order filter can cause the AMB system to become unstable. A lower-order (1-2 poles) filter, with the same passband as the high-order filter, is then used and some aliasing of the input is tolerated [18]. As there is always a random noise component evident in real-world signals, it is prudent to use

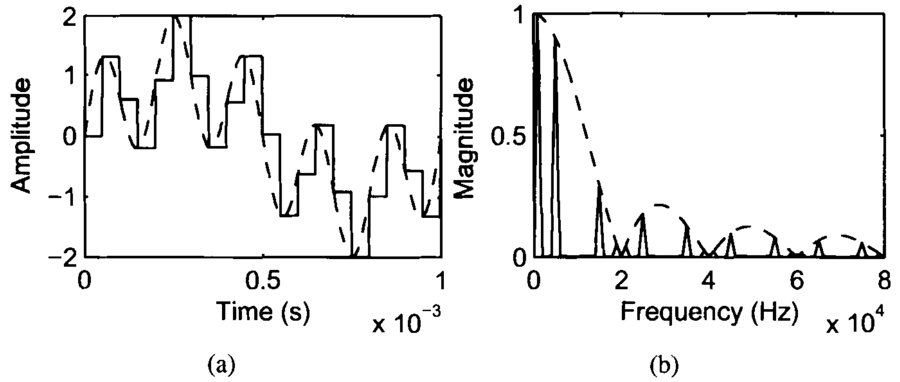


Figure 2.5: DAC conversion

(a) DAC input and output (b) FFT of DAC output

an analogue filter to avoid aliasing, even if a synchronous sampling scheme is used. Analogue low-pass filters are discussed in Section 2.5 as they are relevant to the application concerned.

2.3 Digital to analogue conversion

In order to interface with a process, the DSP's digital control signals have to be converted into analogue signals. Figure 2.5(a) shows a signal consisting of 1 kHz and 5 kHz components, both with an amplitude of 1, which is sampled using a zero-order-hold (ZOH) DAC at a rate of 20 kHz. The frequency content of the reconstructed analogue signal is shown in Figure 2.5(b). The amplitude of the frequency components are changed according to (2.2) [13]:

$$H(f) = \left| \frac{\sin\left(\frac{\pi f}{f_s}\right)}{\frac{\pi f}{f_s}} \right| \tag{2.2}$$

It can also be seen that the frequency components are mirrored and that they repeat at multiples of f_s. Usually it is necessary to limit the bandwidth of the DAC signal at the output interface by using an analogue low-pass filter, called an Anti-Imaging Filter (AIF), in order to recover the original

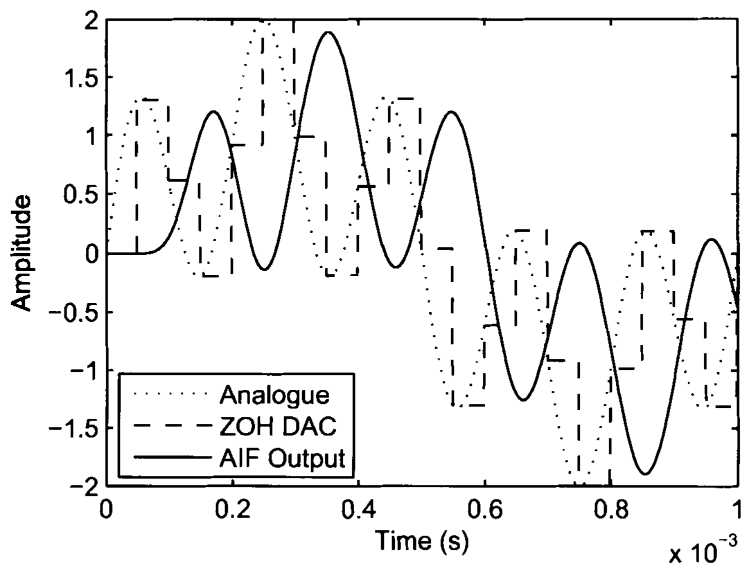


Figure 2.6: Analogue signal reconstruction

signal as shown in Figure 2.6. In AMB systems, however, the control plant usually has a low-pass characteristic, making an AIF unnecessary [2].

From Figure 2.6 it can also be seen that the reconstructed analogue signal's amplitude is reduced because of the nature of the DAC signal. This is predicted by (2.2) and seen in Figure 2.5(b). The reconstructed signal is also time delayed due to the filter and ZOH DAC's time delay.

This can have a drastic effect on the behaviour of a closed-loop control system as the high-frequency components are attenuated and introduces a time-delay in the control loop. The amplitude distortion can, however, be corrected by adjusting the AIF's transfer function [15].

2.4 Processor concerns

There are mainly three concerns when considering the processor, as mentioned in Section 2.1, namely the control law, sampling rate and number representation within the processor. Each of these is now discussed.

2.4.1 The control law

The control law, presumed to be a PID controller for the purpose of this study, has to be converted from a continuous controller into a discrete controller by describing the controller with a set of difference equations [2, 18], written using the z transform in order to integrate and differentiate numerically.

A classical PID controller has the following transfer function [17]:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s \tag{2.3}$$

where K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain.

In the w plane the transfer function of the PID control law can also be written as [19]:

$$D(w) \quad = \quad K_P + \frac{K_I}{w} + K_D w \tag{2.4}$$

The gain of the PID transfer function in (2.4) increases as the frequency increases, which can cause the system to become unstable. This problem is alleviated by introducing a pole (ω_{wp}). The transfer function then becomes [19]:

$$D(w) = K_P + \frac{K_I}{w} + \frac{K_D w}{1 + w/\omega_{wp}} \tag{2.5}$$

If the pole is chosen to be at [19] :

$$\omega_{wp} = -\frac{2}{T} = -\frac{\omega_s}{\pi} \tag{2.6}$$

it is usually far beyond the bandwidth of the system and will have very little impact on the system response. Here ω_s is the sampling rate in rad/s and T is the sampling interval.

The transfer function of the PID controller can then be written as [19]:

$$D(z) = K_P + K_I \frac{T}{2}\left[\frac{z+1}{z-1}\right] + K_D \left[\frac{z-1}{Tz}\right] \tag{2.7}$$

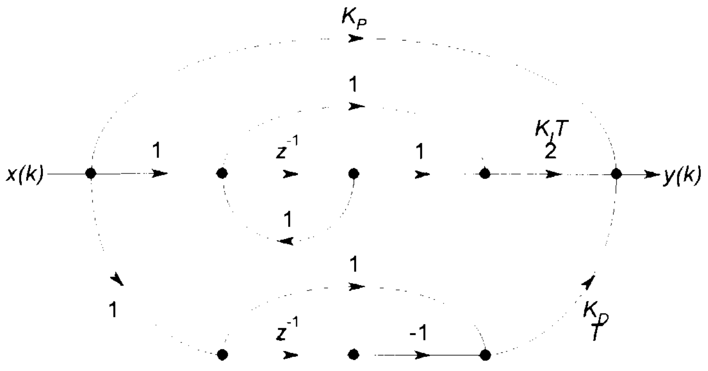


Figure 2.7: PID signal flow [19]

and implemented as shown in Figure 2.7. It may, however, be beneficial to write the transfer function as a standard second-order function. It can then be implemented using one of the standard structures used to realise second-order filters, which will help to avoid coefficient sensitivity problems. If the PID transfer function is written in the form [19]:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{2.8}$$

then

$$\begin{aligned}
a_0 &= K_P + \frac{K_I T}{2} + \frac{K_D}{T} \\
a_1 &= -K_P + \frac{K_I T}{2} - \frac{2K_D}{T} \\
a_2 &= \frac{K_D}{T} \\
b_1 &= -1 \\
b_2 &= 0
\end{aligned} \tag{2.9}$$

One of the structures with the least amount of operations needed to implement the transfer function is the 3D direct second order filter structure shown in Figure 2.8. Its structure prohibits the unbounded escalation of the node values, which therefore makes it ideal for fixed-point implementation.

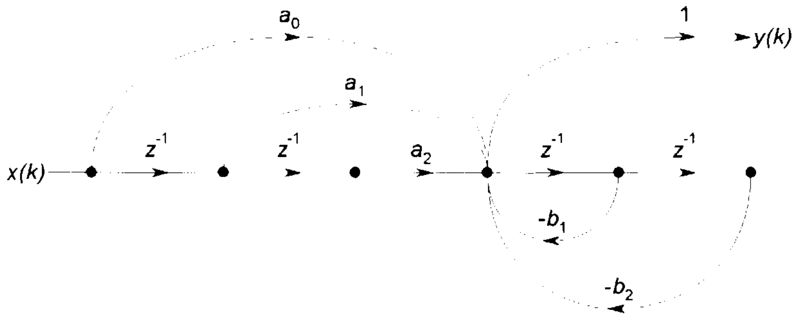


Figure 2.8: 3D direct second order filter structure [19]

2.4.2 Sampling rate

For digital control systems, it is assumed that the control loop is executed once for every sample. The sampling rate therefore determines the rate at which the input is sampled and the rate at which the output is updated.

The selection of the sampling rate depends on the time constant of the system, and should as a rule of thumb be chosen so that there are at least five samples per time constant [2, 18, 19]. If the sampling rate is chosen too high, the calculated derivatives could have a considerable error, which would make the control system susceptible to noise. If it is chosen too low, the control system would be incapable of actively controlling high-frequency disturbances [2].

2.4.3 Number representation

Numbers within digital systems are stored using a digital code with a finite resolution. The code closest to the number is then used to represent it. This is done in one of two ways, namely [15]:

- Floating Point, and
- Fixed Point.

Floating point numbers are represented by three parts. For 32-bit floating point (single precision) numbers, bits 0 to 22 are used for the mantissa (*M*), bits 23 to 30 for the exponent (*E*) and bit 31 for the sign (*S*). The value of the floating point number (*v*) can then be determined using (2.10) [15]:

$$v = (-1)^S \times M \times 2^{E-127} \tag{2.10}$$

keeping in mind that some number codes are reserved for $\pm\infty$, ± 0 and some other codes, referred to as NANs (Not A Number).

Although the floating point numbers are very good at representing large and small numbers [15], the DSPs capable of floating-point calculations are considerably slower than their fixed-point counterparts. For instance, Texas Instrument's TMS320C6x family of DSPs are capable of up to 8000 MIPS (fixed point), but only up to 1800 MFLOPS (floating point) [20]. Generally, floating-point DSPs also do not have peripherals such as ADCs, DACs and PWM controllers on-chip [20] and therefore need more external peripherals to perform the task of digital control.

Fixed-point numbers lack the precision of floating point numbers, as a bit may be used for the sign, a certain number of bits, called the radix, are then used for the integer part of the number and the rest of the bits are used to represent the fraction part of the number as close as possible. Number range and scaling then becomes part of the design process, complicating algorithm implementation [15].

The precision of the number representation will affect the accuracy of the numerical integration and differentiation used, as both the value and the time-step will have numerical errors. A very short sample time will further compound the error as integration and differentiation are done more regularly [2].

2.5 Interface filters

As anti-aliasing and anti-imaging are done using low-pass filters, low-pass filters are discussed in detail in this section. First the different filter approximations are discussed, and then the impact of the filters on the closed-loop control is discussed.

Table 2.1: Filter specifications

Parameter	Value
Passband frequency (ω_P)	6283.2 rad/s (1 kHz)
Stopband frequency (ω_S)	125663.7 rad/s (20 kHz)
Maximum passband attenuation (A_{max})	0.5 dB
Minimum stopband attenuation (A_{min})	80 dB

2.5.1 Filter transfer function

There are mainly four types of approximations which are used to derive a suitable transfer function for the realisation of low-pass filters, namely [21] :

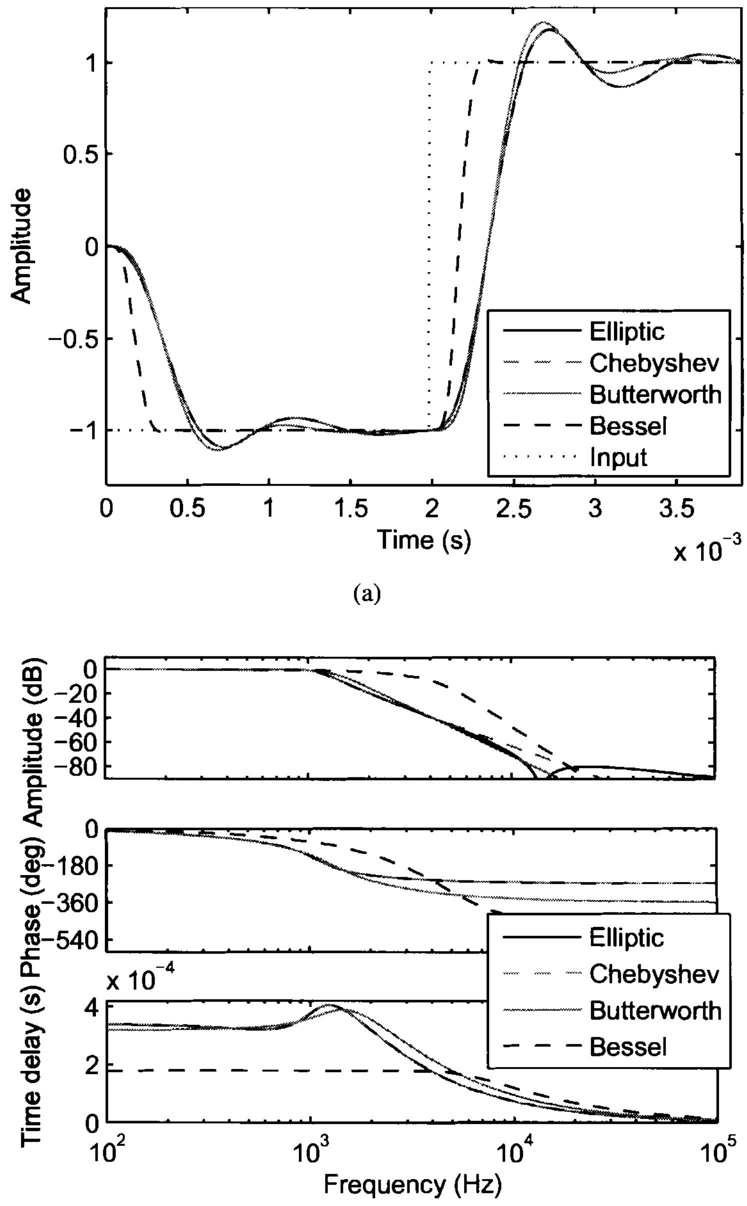
- Elliptic
- Chebyshev
- Butterworth
- Bessel

Each approximation uses other criteria, resulting in transfer functions which differ significantly in their amplitude and phase characteristics. A filter is now designed with each of the above mentioned approximations using the specifications given in Table 2.1.

The step and frequency responses of the designed filters is shown in Figures 2.9(a) and (b) respectively.

2.5.1.1 Elliptic

The elliptic approximation uses a rational function with finite poles and zeros in order to provide the best possible fit to the loss requirements of the filter. This generally results in the lowest order approximation of all the approximation methods discussed here [21].



(a)

(b)

Note that the responses of the elliptic and Chebyshev filters are nearly identical.

Figure 2.9: Analogue filter responses

(a) Step response (b) Frequency response

The approximation has distinguishing poles of attenuation in the stopband and equiripple in the passband. The mathematical development of the approximation is quite complex, so the design is done using tabulated normalised transfer functions like those shown in Table 2.2 [21].

In the tables the frequency (Ω) is normalised to the passband frequency. Therefore [21]

$$\Omega_S = \frac{\omega_S}{\omega_P} \tag{2.11}$$

The functions are denormalised by replacing [21]

$$s \text{ with } \frac{s}{\omega_P} \tag{2.12}$$

When a low-pass filter is designed using the elliptic approximation and the specification given in Table 2.1, the filter is a 3rd order filter with the following transfer function:

$$H_F(s) = \frac{22.8018s^2 + 1.5557 \times 10^{-10}s + 1.7797 \times 10^{11}}{(s^3 + 7.8679 \times 10^3 s^2 + 6.0585 \times 10^7\ s \\ + 1.7797 \times 10^{11}\ )} \tag{2.13}$$

2.5.1.2 Chebyshev

The Chebyshev approximation has the aim of attaining the best possible attenuation in the stopband, while allowing a certain amount of equiripple in the passband. The more ripple is allowed, the better attenuation is achieved in the stopband for a given filter order [21].

The *loss* transfer function of the Chebyshev approximation is described by [21]:

$$|H_L(j\omega)| = \left|\frac{V_{in}(j\omega)}{V_{out}(j\omega)}\right| = \sqrt{1 + \varepsilon^2 C_n^2\left(\frac{\omega}{\omega_P}\right)} \tag{2.14}$$

where ε is a constant with

$$\varepsilon = \sqrt{10^{0.1A_{max}} - 1} \tag{2.15}$$

Table 2.2: Normalised elliptic loss functions for $A_{max} = 0.5$ dB [21]

$\Omega_S = 1.5$

n	Denominator Constant K	Denominator of $H(s)$	Numerator of $H(s)$	A_{min} (dB)
2	0.38540	$s^2 + 3.92705$	$s^2 + 1.03153s + 1.60319$	8.3
3	0.31410	$s^2 + 2.80601$	$(s^2 + 0.45286s + 1.14917)(s + 0.766952)$	21.9
4	0.015397	$(s^2 + 2.53555)(s^2 + 12.09931)$	$(s^2 + 0.25496s + 1.06044)(s^2 + 0.92001s + 0.47183)$	36.3
5	0.019197	$(s^2 + 2.42551)(s^2 + 5.43764)$	$(s^2 + 0.16346s + 1.03189)(s^2 + 0.57023s + 0.57601)(s + 0.42597)$	50.6

$\Omega_S = 2.0$

n	Denominator Constant K	Denominator of $H(s)$	Numerator of $H(s)$	A_{min} (dB)
2	0.20133	$s^2 + 7.4641$	$s^2 + 1.24504s + 1.59179$	13.9
3	0.15424	$s^2 + 5.15321$	$(s^2 + 0.53787s + 1.14849)(s + 0.69212)$	31.2
4	0.0036987	$(s^2 + 4.59326)(s^2 + 24.22720)$	$(s^2 + 0.30116s + 1.06258)(s^2 + 0.88456s + 0.41032)$	48.6
5	0.0046205	$(s^2 + 4.36495)(s^2 + 10.56773)$	$(s^2 + 0.19255s + 1.03402)(s^2 + 0.58054s + 0.52500)(s + 0.392612)$	66.1

$\Omega_S = 3.0$

n	Denominator Constant K	Denominator of $H(s)$	Numerator of $H(s)$	A_{min} (dB)
2	0.083974	$s^2 + 17.48528$	$s^2 + 1.35715s + 1.55532$	21.5
3	0.063211	$s^2 + 11.82781$	$(s^2 + 0.58942s + 1.14559)(s + 0.65263)$	42.8
4	0.00062046	$(s^2 + 10.4554)(s^2 + 58.471)$	$(s^2 + 0.32979s + 1.063281)(s^2 + 0.86258s + 0.37787)$	64.1
5	0.00077547	$(s^2 + 9.8955)(s^2 + 25.0769)$	$(s^2 + 0.21066s + 1.0351)(s^2 + 0.58441s + 0.496388)(s + 0.37452)$	85.5

and $C_n(\Omega)$ is calculated using

$$C_n(\Omega) = 2\Omega C_{n-1}(\Omega) - C_{n-2}(\Omega) \tag{2.16}$$

where

$$\begin{aligned} C_1(\Omega) &= \Omega & (2.17)\\ C_0(\Omega) &= 1 & (2.18) \end{aligned}$$

The *normalised loss* transfer function $H_L(s)$ is given by [21]

$$H_L(s) = \prod_j \frac{1}{K}(s - s_j) \tag{2.19}$$

with s_j is the left-hand plane roots of

$$1 + \varepsilon^2 C_n^2\left(\frac{s}{j}\right) = 0 \tag{2.20}$$

and is denormalised by replacing

$$s \text{ with } \frac{s}{\omega_P} \tag{2.21}$$

K is adjusted in order to provide 0 dB loss at the passband minima [21].

When a low-pass filter is designed using the Chebyshev approximation and the specification given in Table 2.1, the filter is a 3rd order filter with the following transfer function:

$$H_F(s) = \frac{1.7753 \times 10^{11}}{(s^3 + 7.8723 \times 10^3 s^2 + 6.0595 \times 10^7\ s + 1.7753 \times 10^{11}\ )} \tag{2.22}$$

2.5.1.3 Butterworth

The Butterworth approximation has the characteristic that its slope is close to zero at DC [21]. The amplitude of the transfer function is then optimally flat throughout the passband, as seen in

Figure 2.9(b). The Butterworth approximation is therefore used where the magnitude of the signal throughout the passband must be preserved in the frequency domain.

All the transfer functions discussed so far, including the Butterworth approximation, do not have a linear phase response and therefore causes the output to ring when a step input is applied to the filter, as seen in Figure 2.9(a).

The *loss* transfer function for the Butterworth approximation is described by [21]

$$|H_L(j\omega)| = \left|\frac{V_{in}(j\omega)}{V_{out}(j\omega)}\right| = \sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_P}\right)^{2n}} \tag{2.23}$$

where ε a constant given by

$$\varepsilon = \sqrt{10^{0.1A_{max}} - 1} \tag{2.24}$$

and the order *n* calculated with

$$n = \frac{\log_{10}\left(\frac{10^{0.1A_{min}-1}}{\varepsilon^2}\right)}{\log_{10}\left(\frac{\omega_S}{\omega_P}\right)^2} \tag{2.25}$$

The *normalised loss* transfer function $H_L(s)$ is described by [21]

$$H_L(s) = \prod_j (s - s_j) \tag{2.26}$$

with s_j the left half plane roots of

$$1 + (-s^2)^n = 0 \tag{2.27}$$

and is denormalised by replacing

$$s \text{ with } s\left(\frac{\varepsilon^{1/n}}{\omega_P}\right) \tag{2.28}$$

The *gain* transfer function $H_F(s)$ is then given by

$$H_F(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{H_L(s)} \tag{2.29}$$

When a low-pass filter is designed using the Butterworth approximation and the specification given in Table 2.1, the filter is a 4th order filter with the following transfer function:

$$H_F(s) = \frac{4.4618 \times 10^{15}}{\begin{array}{c}(s^4 + 2.1357 \times 10^4\ s^3 + 2.2806 \times 10^8\ s^2 \\ +\,1.4266 \times 10^{12} s\ + 4.4618 \times 10^{15}\ )\end{array}} \tag{2.30}$$

2.5.1.4 Bessel

The Bessel approximation is used to approximate a loss transfer function that has a delay characteristic which is optimally flat throughout the passband. This results in a loss transfer function which has a linear phase characteristic throughout the passband. The Bessel approximation is therefore used where the phase of signal components must be preserved [21].

As seen in Figure 2.9(a), the filter has the best step response of all the approximations, but at a cost. The magnitude of the frequency response shows a very slow transition from the passband to the stopband and therefore the transfer function order is the highest of all the approximations.

The higher order allows the knee frequency of the Bessel approximation to be much higher than the other approximations, resulting in the shortest group delay of all the approximations. If, however, the group delay is compared to the group delay of the same order transfer functions designed with the other approximations, using the same knee frequency, the Bessel approximation will have the longest group delay of all.

The *normalised loss* transfer function for the Bessel approximation is given by [21]

$$H_L(s) = \frac{V_{in}(s)}{V_{out}(s)} = \frac{B_n(s)}{B_n(0)} \tag{2.31}$$

with $B_n(s)$ calculated using the following recursive formula:

$$B_n(s) = (2n-1)B_{n-1}(s) + s^2 B_{n-2}(s) \tag{2.32}$$

where

$$B_1(s) = s + 1 \tag{2.33}$$

$$B_0(s) = 1 \tag{2.34}$$

The transfer function is denormalised by replacing [21]

$$s \text{ with } sT_0 \tag{2.35}$$

where

$$T_0 = \frac{\Omega}{\omega} \tag{2.36}$$

with Ω the normalised frequency and ω the denormalised frequency.

The *gain* transfer function $H_F(s)$ is then given by

$$H_F(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{H_L(s)} \tag{2.37}$$

When a low-pass filter is designed using the Bessel approximation and the specification given in Table 2.1, the filter is a 6th order filter with the following transfer function:

$$\begin{aligned} H_F(s) \quad = \quad & \frac{3.2404 \times 10^{26}}{\begin{aligned}(s^6 + 1.1781 \times 10^5\ s^5 + 6.6091 \times 10^9\ s^4 \\ + 2.2246 \times 10^{14} s^3 + 4.6800 \times 10^{18} s^2 \\ + 5.7761 \times 10^{22} s\ + 3.2404 \times 10^{26}\ )\end{aligned}} \end{aligned} \tag{2.38}$$

2.5.2 Impact on AMB performance

According to [22] the addition of filters within the control loop of the AMB has the effect of increasing the natural frequency (ω_n) of the closed-loop system. This has the effect of increasing the second order equivalent stiffness (k_{eq}) of the system, while the increase in (b_{eq}) is less prominent. The static stiffness of the AMB remains unchanged as it is dependent on the DC characteristics of the AMB. Interface filters also have the effect of reducing the dynamic stiffness of the AMB at high frequencies.

2.6 Conclusion

There are various issues which have to be addressed when designing a digital AMB controller, some of which are interfacing, algorithm implementation and sampling rate.

The interfacing consists of two components, namely the AAF and AIF. The AAF is used to bandwidth limit the input to the controller in order to prevent aliasing, while the AIF is used to bandwidth limit the controller's output in order to prevent imaging. As AMB systems generally have a low-pass transfer characteristic, an AIF is usually unnecessary.

The PID control law is transformed into a discrete second order transfer function and can then be implemented using the 3D direct filter structure as discussed. The 3D filter structure aids coefficient insensitivity and is computationally efficient.

The controller's sampling rate is of utmost importance as it determines the controller's bandwidth. It also determines the frequency content of the controller's input and therefore influences the AAF design.

When selecting a filter approximation, it is important to consider the phase characteristic of the approximations, as the controller is highly dependant on the amplitude and phase of the input signal.

Chapter 3

System Design

This chapter discusses the design and specification of the embedded AMB controller. It starts off with the design process, which is followed by the detail system design and is concluded with detailed simulations of the closed-loop system including interface filters and DSP firmware.

3.1 Design process

Figure 3.1 illustrates the design process for the development of the embedded AMB controller. As the design process is iterative in nature, only the final iteration is discussed in this chapter. The process is summarised as follows:

1. The performance and interface requirements of the controller is specified.
2. The appropriate DSP platform and development tools are selected.
3. The interface circuitry is designed with the system and DSP interface requirements in mind. The closed-loop system response is then simulated using an analogue PID controller. The interface circuitry is implemented once the performance is acceptable.

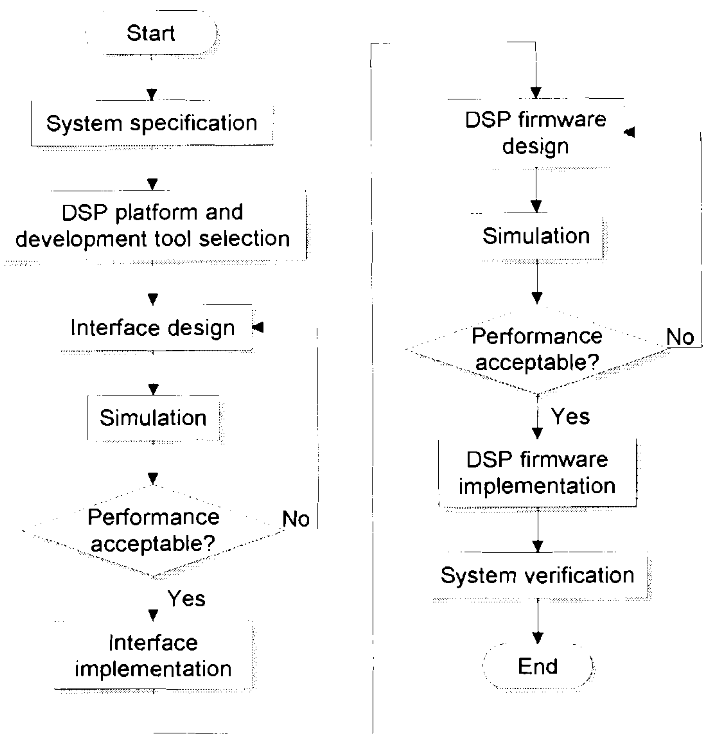


Figure 3.1: Controller design process

4. The DSP firmware is designed with the controller requirements in mind. The closed-loop system response is simulated including the firmware. The firmware is implemented if the controller meets the specified requirements.

5. The overall system performance is verified.

Steps 1 and 2 as well as the design and simulation of steps 3 and 4 are discussed in detail in the following sections. The implementation of steps 3 and 4 is discussed in Chapter 4 and step 5 is discussed in Chapter 5.

3.2 System specification

The embedded controller is specified according to the system's control parameters and the system interfaces, namely:

- AMB interface,
- PC interface,
- User interface, and
- Mains interface.

A visual representation of the system interfaces is shown in Figure 3.2. Each aspect of the system specification will now be discussed in detail. A summary of the system specification is given at the end of the section in Table 3.1.

Control parameters: The embedded controller is intended to fully suspend and actively control an active magnetic bearing system with rotor speeds of up to 30 000 rpm. It will be used to suspend an existing double radial AMB model, which requires four axes to be controlled with a PID control law. There should, however, be adequate processing speed to control a fully suspended AMB system with five axes.

Closed-loop system performance: The static stiffness and equivalent second order stiffness of the embedded controller shall be compared with the existing dSPACE® controller. A maximum deviation of 20 % will be tolerated.

AMB interface: The AMB interface consists of three types of signals, namely the current reference output, the power amplifier synchronisation output and the rotor position input. The nature of these signals are determined by the interface of the existing double radial AMB model. The controller must therefore have the following interface signals [10]:

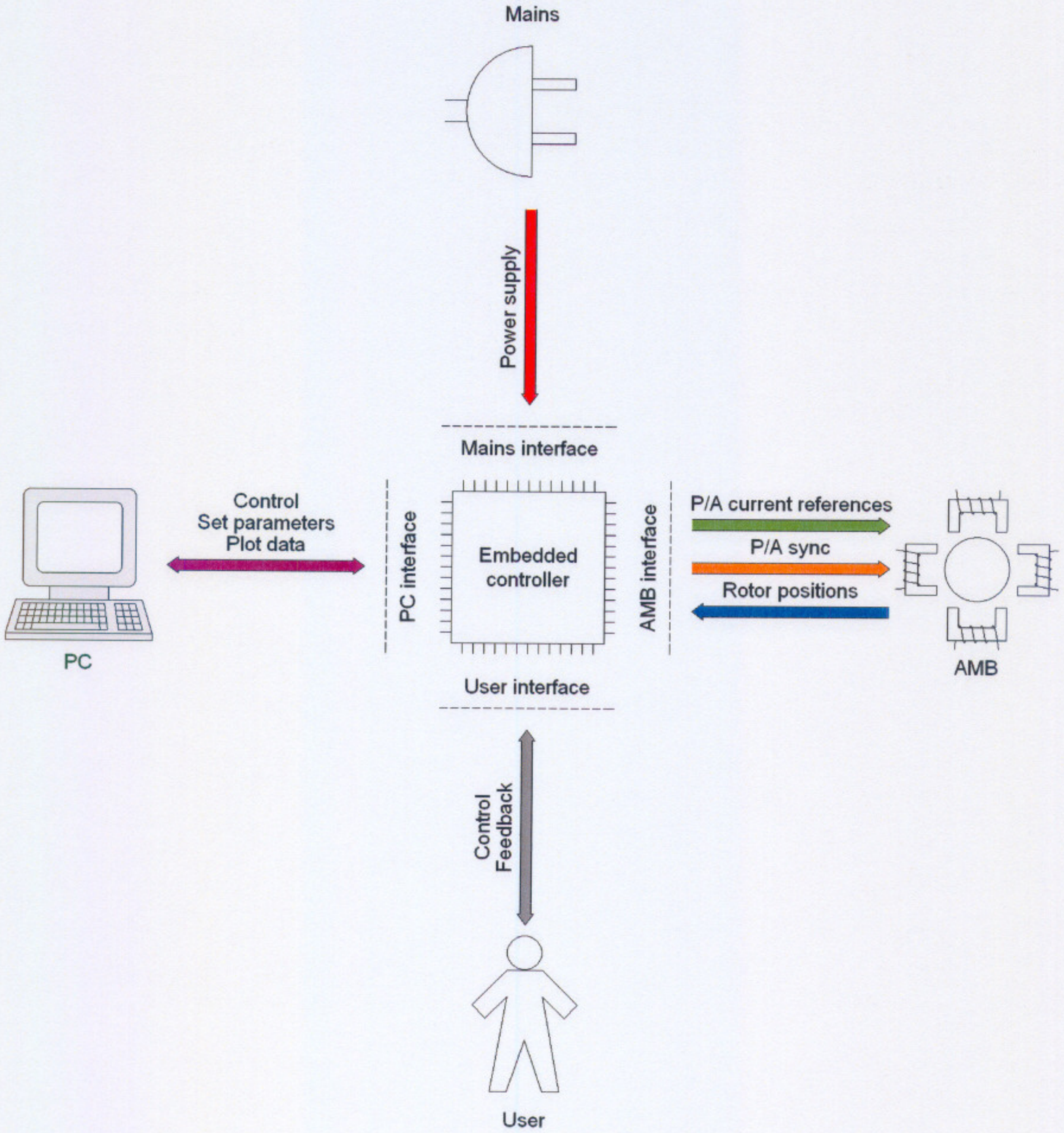


Figure 3.2: System interfaces

- Two current reference outputs for each axis with a range of *approximately* 2.5 to 10 V, representing a current range of 0 to 10 A.
- A single square wave with a voltage range of 0 to 5 V, a frequency of 100 kHz and a variable pulse width is used to synchronise the power amplifiers.
- One rotor position input per axis, having a sensitivity of 7.87 mV/μm with a DC offset, resulting in a range of *approximately* 0 to -10 V for 600 to -600 μm.

PC interface: The PC interface enables real-time performance analysis and control of the AMB through a high-speed digital communications link. In order to perform performance analysis, the user must be capable of setting the position reference for a given axis for every control cycle and be able to plot the rotor position using a graphical interface. The user must also be able to update the PID constants and turn the AMB on and off using the graphical interface. The embedded controller should be able to operate in a stand-alone mode and can therefore not be dependent on the PC interface for normal operation.

User interface: Through the user interface, the user must be capable of turning the AMB on and off by using a pushbutton, as well as verify the correct operation of the controller by means of an activity LED.

Mains interface: Power is supplied to the embedded controller through the mains interface. It should be able to connect to the power grid using a 220 V AC outlet. The low-voltage side should also be isolated.

3.3 DSP platform and development tool selection

The selected DSP will be used in a research environment which will pose stringent requirements to the DSP. The selected DSP's clock speed should therefore be as high as possible, while still being reasonably priced and readily available. The DSP should also incorporate peripherals such as high-speed ADCs, DACs, PWM controllers and high-speed digital communication modules.

Table 3.1: System specification

Parameter	Specification
Control	
Control law	PID
No. of axis to be controlled	4 (max 5)
Maximum rotor speed	60 000 rpm
Closed-loop performance	
Maximum deviation tolerated	20 %
AMB interface	
Current reference output	0 to 10 A (0 to 10 V) 2 per axis
Power amplifier synchronisation output	100 kHz, 0 to 5 V square wave adjustable duty cycle
Rotor position input	600 to -600 μm (approx. 0 to -10 V) 7.87 mV/μm sensitivity 1 per axis
PC interface	
Control	On/off
Plot data	Rotor positions
Set parameters	PID constants K_p, K_i and K_d Position references
Update rate	Once every cycle
User interface	
Control	On/off
Feedback	Activity LED
Mains interface	
Power supply	220 V AC input Low-voltage side isolated

Texas Instruments' TMS320C2000® DSP family is therefore selected as the DSP platform for the embedded controller, as these DSPs are developed especially for motor control and are the fastest DSPs available that include peripherals such as ADCs and PWM controllers on-chip. The specific chip selected is the TMS320F2812 as it is the fastest within the TMS320C2000® family and has very good development tools available [20, 23].

The features of the TMS320F2812 include [24]:

- 150 MHz operation (6.7 ns cycle time)
- 32-bit fixed-point processor
- 1.9 V Core, 3.3 V I/O
- 16 × 12-bit ADCs
- 16 × PWM outputs
- SPI high-speed communication

It should be noted that the DSP does not include a DAC module. This is addressed in Section 3.4.1.2.

Development tools available for the TMS320F2812 include:

TMS320F2812 eZdsp® DSK : The eZdsp® DSP Starter Kit (DSK) (shown in Figure 3.3) is a low-cost development tool incorporating a TMS320F2812 DSP, external RAM and IEEE 1149.Q JTAG emulation, enabling real-time debugging. The starter kit operates from a single 5 V supply, provides access to all the DSP's I/O features and includes Texas Instruments' C compiler and development environment, Code Composer Studio® [25].

Figure 3.3: TMS320F2812 eZdsp® DSK [25]

VisSim® Embedded Controls Developer: VisSim® Embedded Controls Developer is a high-level graphical programming language, enabling rapid development and simulation of embedded software as demonstrated in [11]. It supports fixed-point mathematical calculation, with an auto-scaling feature, as well as the on-chip peripherals of the TMS320F2812. VisSim® is capable of generating C code which is easily read and interpreted and has specific support for the TMS320F2812 eZdsp® DSK, which includes hardware-in-the-loop simulation capability and a plug-in for Code Composer Studio® [26].

The TMS320F2812 is therefore chosen for its speed, on-chip peripherals and available development tools. A TMS320F2812 eZdsp® DSK is sourced for the controller along with VisSim® Embedded Controls Developer, in order to facilitate the hardware and software development process. With the DSP selected and controller interfaces specified, the interfaces can be designed, as discussed in the following section.

3.4 Interface design

The design of each interface is now discussed in the following subsections, keeping in mind the DSP's interface requirements and the requirements specified in Section 3.2.

3.4.1 AMB interface

In order to design the AMB interface, the bandwidth of the control signals is first determined. With a maximum rotational speed of 60 000 rpm, the maximum frequency of the disturbance due to the rotation is 1 kHz, which is then also the system bandwidth.

3.4.1.1 Power amplifier synchronisation output

The power amplifier synchronisation output is a 5 V, 100 kHz square wave with a variable pulse width. The signal is therefore generated using a PWM output of the TMS320F2812 and converted from a 3.3 V signal to a 5 V signal by means of a buffer circuit.

3.4.1.2 Current reference output

As specified in Section 3.2, the power amplifiers require a current reference with a voltage range of 2.5 to 10 V that represents currents from 0 to 10 A. As the TMS320F2812 does not have DACs, the PWM outputs of the DSP are used and low-pass filtered by an AIF [27]. The PWM frequency is also chosen at 100 kHz in order to simplify the configuration of the DSP's event manager. The DAC would then have a resolution of approximately 10 bits. The corner frequency of the AIF is chosen to be 5 kHz in order to minimise the phase lag and attenuation introduced by the filter. The Bessel approximation is chosen for the filter transfer function as it will not distort the phase of the current reference signals.

The AIF has to have a gain (G) of

$$\begin{aligned} G &= \frac{V_{out_{max}}}{V_{in_{max}}} \\ &= \frac{10}{3.3} \\ &= 3.0303 \end{aligned} \tag{3.1}$$

in order for the maximum output voltage ($V_{out_{max}}$) to be 10 V for a maximum input voltage ($V_{in_{max}}$) of 3.3 V. The minimum voltage of 2.5 V is achieved by adjusting the PWM duty cycle with a factor of 0.75 and adding an offset.

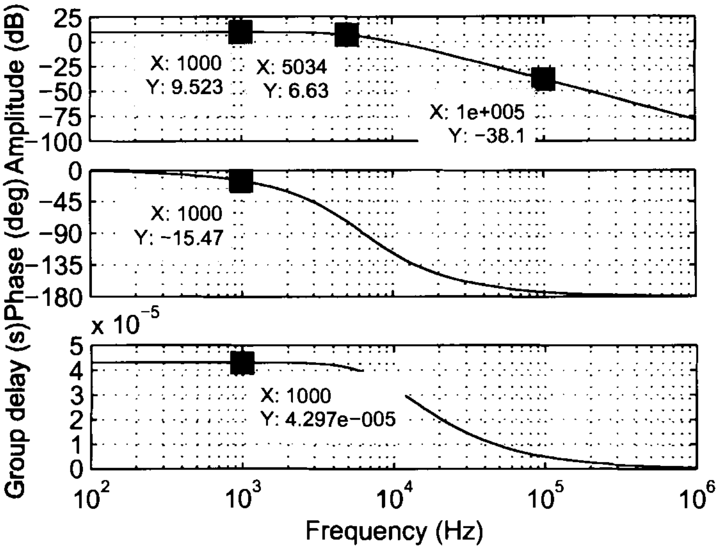


Figure 3.4: AIF frequency response

In order to minimise the group delay of the AIF, a maximum filter order of 2 is used with the bandwidth as high as possible. The transfer function is approximated using the description in Section 2.5.1.4 with a bandwidth of 5 kHz ($A_{max} = 3$ dB) and found to be

$$\begin{aligned} H_F(s) &= G \cdot \frac{1.6246 \times 10^9}{s^2 + 6.9813 \times 10^4 s + 1.6246 \times 10^9} \\ &= \frac{4.9231 \times 10^9}{s^2 + 6.9813 \times 10^4 s + 1.6246 \times 10^9} \end{aligned} \quad (3.2)$$

The AIF transfer function realises an attenuation of 38.1 dB at 100 kHz, as seen in Figure 3.4, with the -3 dB frequency of the filter just above 5 kHz.

The PWM frequency at 100 kHz would then cause a voltage ripple of 41 mV because of the 38.1 dB attenuation of the AIF at 100 kHz. This can easily be tolerated by the AMB, as it would cause a current ripple of approximately 30 mA.

3.4.1.3 Rotor position input

In order to design the rotor position input interface, three issues have to be addressed. Firstly, the ADC sample rate must be chosen, which then determines the cutoff frequency of the AAF. The filter gain must also be calculated from the input and output voltage ranges whereafter the AAF filter can be designed.

Sampling rate: As stated earlier, the maximum frequency of the disturbance to be compensated for is 1 kHz. In Section 2.4.2 it is said that the sampling rate is chosen, as a rule of thumb, to be at least five times the system bandwidth. In [28], however, it is recommended that the sampling rate be six to ten times the system bandwidth. The sample rate is therefore chosen to be 10 kHz, as this is ten times faster than the fastest disturbance.

AAF gain: As specified in Section 3.2, the input voltages of the AAF have a range of 0 to -10 V while the ADC of the TMS320F2812 can only convert voltages from 0 to 3 V [29]. The filter gain (G) must therefore be

$$\begin{aligned} G &= \frac{V_{out_{max}}}{V_{in_{max}}} \\ &= \frac{3}{10} \\ &= 0.3 \end{aligned} \tag{3.3}$$

It must be noted that the output of the filter must be inverted when the filter is implemented in order to get a positive output voltage for a negative input voltage.

With the sample rate and filter gain determined, the AAF can be designed. An ideal design for an AAF is first considered, whereafter a practical AAF is designed.

Ideal AAF specification: For an n-bit ADC to have no aliasing evident in the converted signal, the attenuation of the AAF with a gain of G must be

$$A_{min} = 20\log(2^n - 1) - 20\log(1/G) \tag{3.4}$$

If A_{min} is calculated for the DSP's 12-bit ADC and the filter gain $G = 0.3$, A_{min} is calculated to be 61.79 dB at the Nyquist frequency. The AAF's attenuation should therefore be a minimum of A_{min} dB at the Nyquist frequency in order to prevent aliasing. If a Bessel approximation is used with $A_{max} = 3$ dB at 2.5 kHz and a 20 kHz sample rate is assumed, resulting in a Nyquist frequency of 10 kHz, the filter will need to be a 15th order function.

It would be very unpractical to implement such a filter as it would introduce a very long group delay.

ADC synchronisation: As explained in Section 2.2.2, the aliasing can also be avoided if the noise only occurs at specific intervals and these intervals are controlled by the DSP. This is indeed the case, as the DSP controls the switching of the power amplifiers by means of the power amplifier synchronisation output. If the ADC is synchronised with the synchronisation output, the requirements of the AAF can greatly be relaxed.

AAF design: The AAF is specified as a second-order filter in order to minimise its group delay. The AAF also uses the Bessel approximation for its phase characteristics. The gain (G) must be 0.3 as determined earlier. The normalised filter should provide no more than 0.5 dB attenuation at 1 kHz. The filter transfer function is therefore given by

$$\begin{aligned} H_F(s) &= G \cdot \frac{3.6453 \times 10^8}{s^2 + 3.3069 \times 10^4 s + 3.6453 \times 10^8} \\ &= \frac{1.0936 \times 10^8}{s^2 + 3.3069 \times 10^4 s + 3.6453 \times 10^8} \end{aligned} \tag{3.5}$$

The frequency response of the designed AAF filter is shown in Figure 3.5. As seen in the figure, 20.89 dB attenuation is achieved at the Nyquist frequency. This should provide enough attenuation of very high frequency components without attenuating the input to the controller, although some level of aliasing may still be allowed to occur. From the figure it can also be seen that the attenuation of the filter is 10.95 dB at 1 kHz, which is due to the filter gain and the 0.5 dB attenuation of the normalised transfer function.

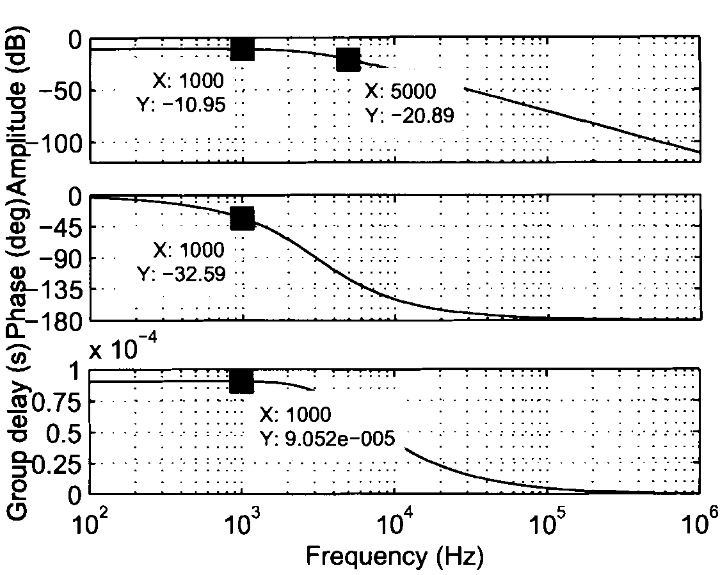


Figure 3.5: AAF frequency response

3.4.2 PC interface

The function of the PC interface is to monitor and control the AMB. Monitoring is constituted by plotting of the actual rotor positions and by displaying the control parameters and device status. Control is done by updating the control parameters, consisting of the PID constants, position references and bias currents, and by turning the AMB on or off.

Raw data will be sent to the PC and pre-processed data will be received from the PC in order to minimise the processing required of the embedded controller. The rotor position and position reference telemetry will be sent to the PC at the end of every control cycle. The PC can also request additional telemetry data, such as the control parameters, and the device status.

Communication channel selection: The communication channel selected is the SPI interface, as this is a high-speed interface available on both the embedded controller and the dSPACE® controller. The two SPI transceivers operate full-duplex in a master-slave mode with the master initiating all data transfers. Data is transferred in one of three modes [30]:

- Master sends data, slave sends data.

- Master sends dummy data, slave sends data.
- Master sends data, slave sends dummy data.

The software should determine whether the sent data is valid or not.

As the embedded controller will continuously send telemetry data, it will be used as the master device. This will also ensure synchronisation of updated data with the program cycle.

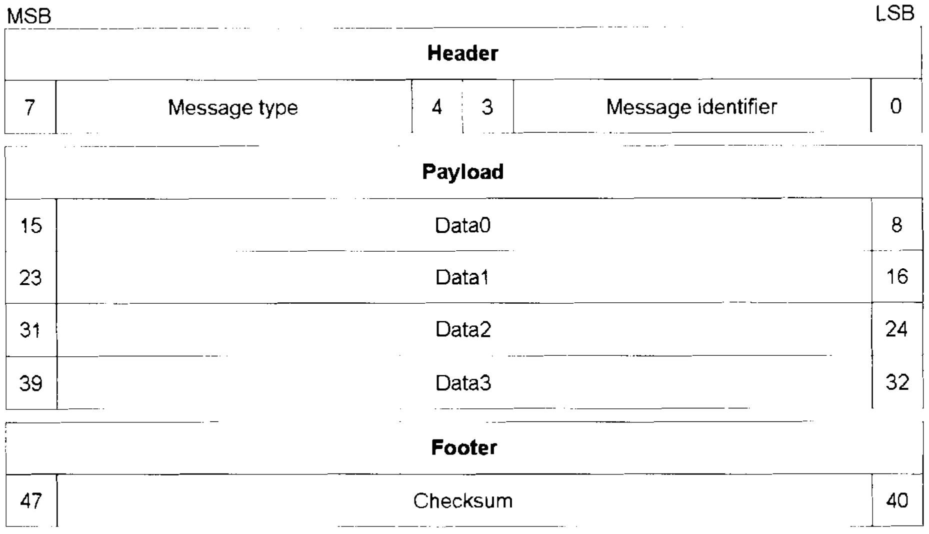
The TMS320F2812 has 32 16-bit first-in-first-out (FIFO) registers available of which 16 can be used for receiving data and 16 can be used for transmitting data. This is not available on the dSPACE® controller, which necessitates the implementation of a software buffer on the slave device.

The bit rate is limited to 1.25 Mbps by the capability of the dSPACE® controller's slave DSP. As the slave DSP is only capable of 8-bit data transfers, it also limits the data transfers to 8 bit characters.

Protocol definition Data sent over the PC interface will be formatted using the frame shown in Figure 3.6. The frame consists of three sections, namely:

- Header,
- Payload, and
- Footer.

The frame header contains message type and message identifier fields, which are used to determine the interpretation of the data frame. This is followed by the payload section, which is used to send data if necessary. If a payload field is not used, it is padded with 0xAA. The final section is the footer, which consists of the frame checksum. The checksum is calculated by determining the sum of all bytes in a frame, except for the checksum byte itself. Any overflow is ignored.



Numbers represent the bit number

Figure 3.6: Packet frame definition

The frame has a fixed size in order to ensure that the master and slave use the same frame size in a single transmission. Because of the frame size and maximum bit rate, only two frames can be transmitted during a single control cycle.

There are four types of messages defined, as outlined in Table 3.2. Request messages are used to request telemetry, update messages to update control parameters or the device status, telemetry messages to send telemetry data and dummy messages are used to pad data transfers. The message identifier is used to determine the type of data carried in the payload of the frame and is interpreted as outlined in Table 3.3.

The master device operates in the following manner: A rotor position telemetry frame is sent to the slave by default. A request or update is then received and serviced by the master. A reply or dummy frame is then transmitted. An update or request is again received and serviced, but no further frames are transmitted. This process is repeated after every control cycle if a previous transmission is not still active.

It is important to note that any request received during the second frame's transmission cannot be replied to. Also, as the position telemetry frame can only accommodate one axis, the left vertical

Table 3.2: Message type definition

Message type	Interpretation
0x1	Update frame
0x2	Request frame
0x3	Telemetry frame
0xF	Dummy frame

rotor position is sent by default. Another axis is selected by sending a rotor position telemetry request frame to the master.

The slave device operates in the following manner: The slave device waits until its send buffer is empty, whereafter it processes received telemetry. It then buffers two frames to be sent to the master during the next transmission. These may be request, update or dummy frames, although request frames are only buffered as a first frame.

Figures 3.7 and 3.8 show the flow diagrams of the protocol implementation on the master and slave devices respectively.

Graphical interface: The graphical interface is a virtual instrumentation panel with which the user can monitor and manipulate the embedded controller. It is the visual front-end for the PC interface and will be implemented using the dSPACE® ControlDesk® software package, as it can seamlessly exchange data with the SPI interface of the dSPACE® controller and easily visualise and capture data.

The graphical display will be similar to the existing double radial AMB interface software as it will perform the same function as the current software.

3.4.3 User interface

The user interface is used to switch the AMB on or off and verify the correct operation of the embedded controller. The user interface therefore consists of two LEDs and a pushbutton. One

Table 3.3: Message identifier definition

Message identifier	**Message interpretation**	**Payload fields***	
0x1	Rotor position	Data0(3:0)	Axis**
		Data0(7:4)	Unused
		Data2:Data1	Value
0x2	Position reference	Data0(3:0)	Axis**
		Data0(7:4)	Unused
		Data2:Data1	Value
0x3	Bias current	Data0(3:0)	Axis**
		Data0(7:4)	Unused
		Data2:Data1	Value
0x4	PID parameter	Data0(3:0)	Axis**
		Data0(7:4)	Parameter***
		Data4:Data1	Value
0xE	Status	Data0	On / off
0xF	Dummy		

* Unused payload fields are padded with 0xAA.

** Axis values are allocated as follows:

Left horizontal	0x1
Left vertical	0x2
Right horizontal	0x3
Right vertical	0x4
Axial	0x5

*** Parameter values are allocated as follows:

a_0	0x1
a_1	0x2
a_2	0x3

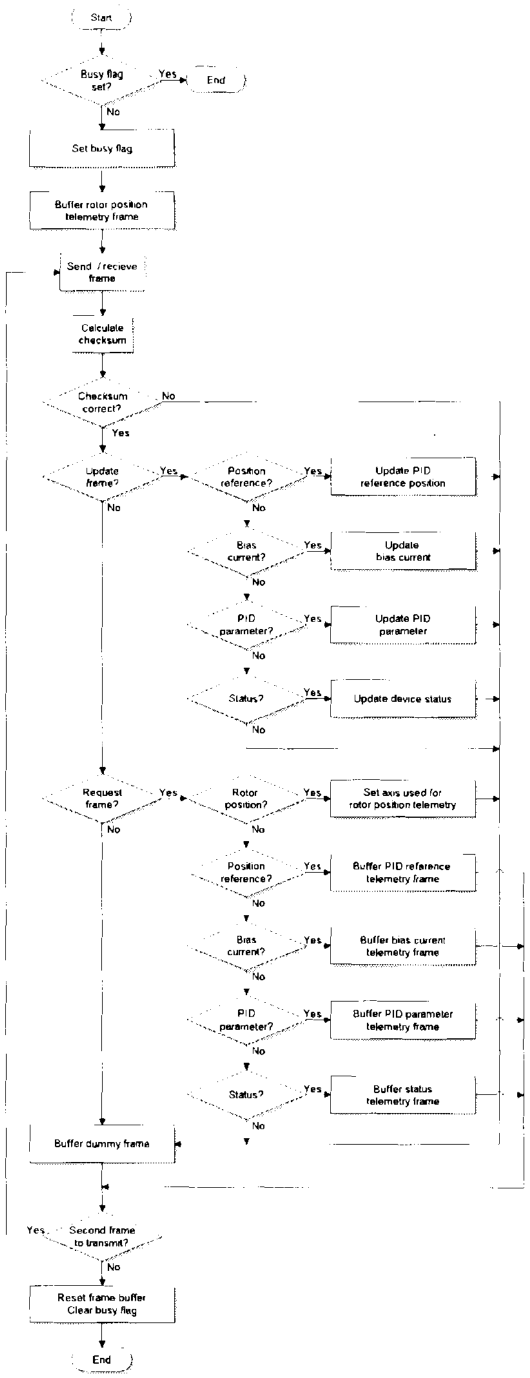


Figure 3.7: Protocol flow diagram: Master

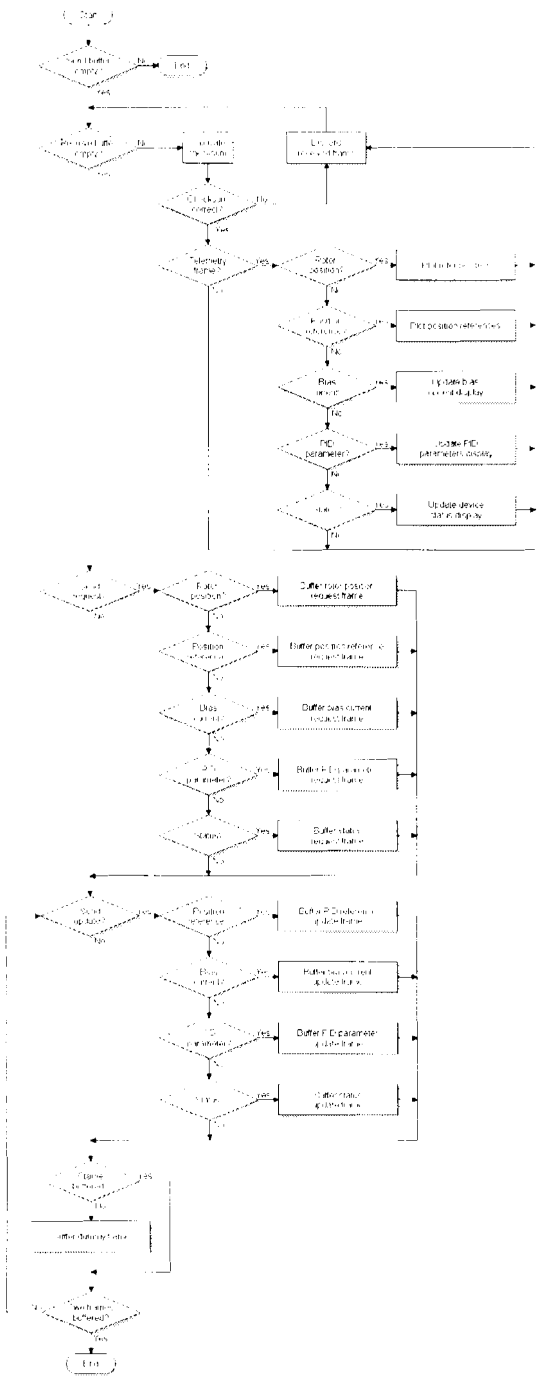

Start
No
End
Yes
No
Checksum correct?
No
Yes
Telemetry frame?
Yes
Rotor position?
Yes
No
Yes
Plot position references
No
Bias
Yes
Update bias current display
No
PID parameter?
Yes
Update PID parameters display
No
Yes
Update device status display
No
Yes
Rotor position?
Yes
Buffer rotor position request frame
No
Position
Yes
Buffer position reference request frame
Bias
Yes
Buffer bias current request frame
parameter?
Yes
Buffer PID parameter request frame
No
Status?
Yes
Buffer status request frame
Send update?
Yes
Position reference
Yes
No
Bias current?
Yes
Buffer bias current update frame
parameter?
Yes
Buffer PID parameter update frame
Yes
Buffer status update frame
Frame buffered
Yes
No
buffered?
Yes
End

Figure 3.8: Protocol flow diagram: Slave

LED, called the Activity LED, will flash at at frequency of 2 Hz in order to indicate that the embedded controller is running. The other LED, called the Status LED, will indicate whether the AMB is turned on or off. The pushbutton will be used to turn the AMB on or off.

3.4.4 Mains interface

The mains interface provides the power necessary to operate the embedded controller. The mains interface connects to a 220 V AC power outlet and provide isolated voltage supplies to the analogue input circuit, the digital output circuit and the DSP in order to prevent ground loops.

3.5 DSP firmware

As specified in Section 3.2, a PID control law is implemented on the embedded controller using only integer calculations. This is done using the 3D second order filter structure discussed in Section 2.4.1 as the filter structure is simple to compute and is less sensitive to its coefficients.

The PID filters' input is scaled to the ADC's output range (0 - 4095) [29] and their output is scaled to the PWM controller's period timer value (0 - 1500 for 100 kHz PWM frequency) [31]. This will ensure that the PID filters are insensitive to integer multiplication and division as they will only be implemented with integer values. This will significantly reduce the number of program cycles needed to compute the PID filters' output.

The PID control loop is run at 10 kHz, as this is the same as the sampling rate specified in Section 3.4.1.3. In order to minimise the processing delay, an axis's position signal will be sampled, its control loop executed and its outputs updated before proceeding to the next axis.

3.6 Simulation

This section discusses the simulations used to verify the closed-loop system performance during the design process. As the development is done using VisSim®, the existing Matlab® simulation

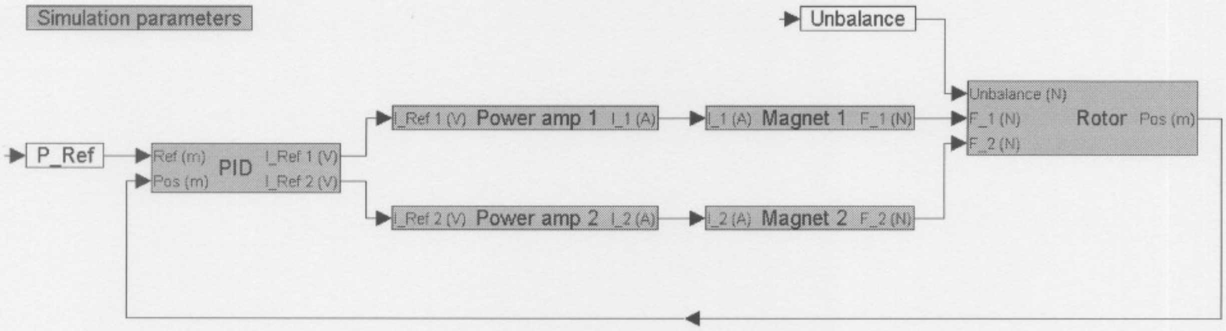


Figure 3.9: Base VisSim® model

model of the double radial AMB system [10] is converted to a VisSim® simulation model. The simulation results of the VisSim® and Matlab® simulation models are compared in order to verify the accuracy of the VisSim® simulation model. With the model verified, the interface filters and DSP firmware is modeled and simulated.

3.6.1 VisSim® simulation model verification

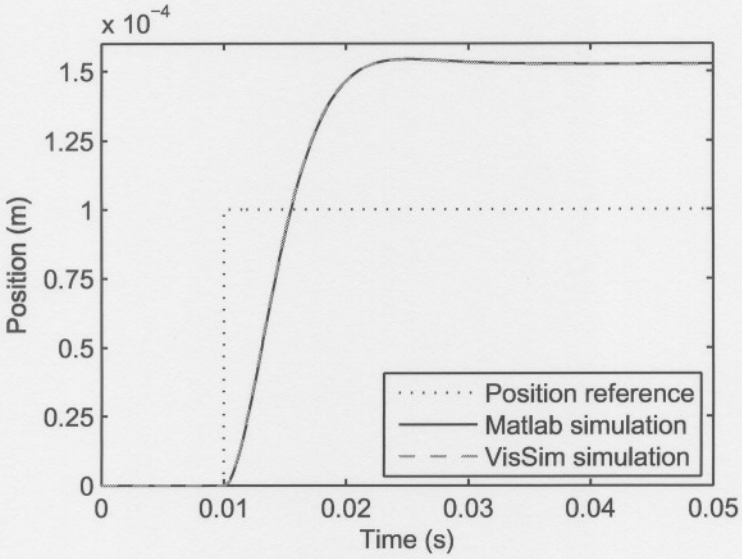
Figure 3.9 shows the VisSim® simulation model of a horizontal axis of the double radial AMB model. The simulation is run at 10 MHz and includes a classical PID controller which is run at 20 kHz as it is done in Matlab® [10].

The results of the Matlab® and VisSim® simulations are shown in Figure 3.10. As seen in the figure, the results are practically identical, which verifies the accuracy of the VisSim® model.

3.6.2 Interface filters

The interface filters designed in Sections 3.4.1.2 and 3.4.1.3 are included in the simulation of the left horizontal axis of the double radial AMB model, as shown in Figure 3.11. The simulation also includes a model of the position sensor, including the sensor voltage gain and offset.

For the simulation, the system was first given some time to reach steady state before the step is applied to the position reference. This is done to eliminate transients caused by the initialisation of



100 μm Horizontal step response

Figure 3.10: VisSim® model verification

the filter transfer functions. The result of the simulation is shown in Figure 3.12. As seen from the figure, the time delay introduced by interface filters have the effect of slightly increasing the slope of the rotor position, which translates into a slightly higher equivalent second order stiffness.

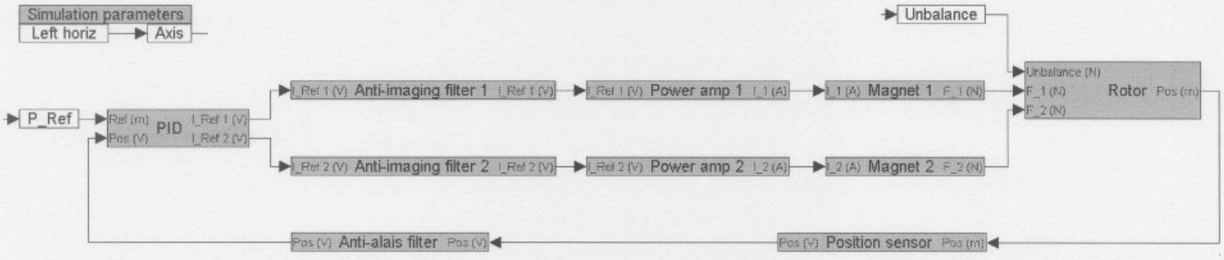
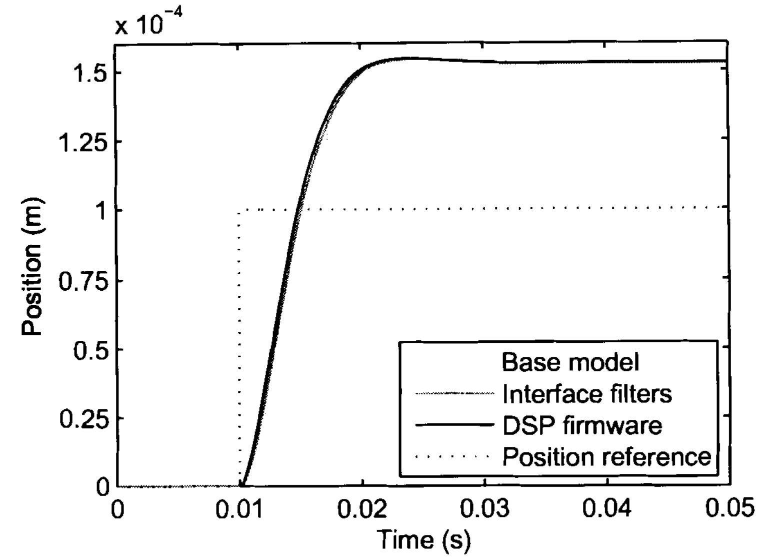

Simulation parameters
Left horiz
Axis
Unbalance
P_Ref
Ref (m)
Pos (V)
PID
I_Ref 1 (V)
I_Ref 2 (V)
I_Ref 1 (V) Anti-imaging filter 1 I_Ref 1 (V)
I_Ref 1 (V) Power amp 1 I_1 (A)
I_1 (A) Magnet 1 F_1 (N)
I_Ref 2 (V) Anti-imaging filter 2 I_Ref 2 (V)
I_Ref 2 (V) Power amp 2 I_2 (A)
I_2 (A) Magnet 2 F_2 (N)
Unbalance (N)
F_1 (N)
F_2 (N)
Rotor Pos (m)
Pos (V) Anti-alais filter Pos (V)
Pos (V) Position sensor Pos (m)

Figure 3.11: Interface simulation model



100 μm Horizontal step response

Figure 3.12: Interface filter and DSP firmware simulations

3.6.3 DSP firmware

In order to simulate the effect of the DSP firmware implementation as specified in Section 3.5, the PID block of the AMB simulation is updated. This enables the simulation to additionally include the following:

- The PID filter structure, including the differentiator pole,
- Quantisation effects on the PID filter coefficients,
- Quantisation effects of the fixed-point arithmetic on the PID filter,
- Input scaling of the PID filter,
- Output scaling of the PID filter,
- Quantisation effects of the 12-bit ADC,
- Quantisation effects of the PWM used as a DAC, and

- The PID algorithm running at 10 kHz.

Again the system is first allowed to reach steady state before the step is applied to the horizontal position reference. As seen in Figure 3.12, the DSP firmware has little effect on the simulated horizontal step response of the double radial AMB model.

3.7 Conclusion

The embedded controller is based on a TMS320F2812 and will execute five PID control loops at a rate of 10 kHz. The PID loops will be implemented using a 3D direct filter structure in order to minimise coefficient sensitivity, using VisSim® Embedded Controls Developer.

The interface of the embedded controller consists of the AAFs and AIFs, which are designed as 2nd order Bessel low-pass filters. The simulation of the double radial AMB predicts that the interface filters and firmware implementation will have the effect of increasing the system's equivalent second order stiffness.

Chapter 4

Design implementation

This chapter covers the implementation of the embedded controller design. Firstly, the implementation of each system interface of the design is discussed, whereafter the firmware implementation is described in detail.

4.1 Interface implementation

The interface implementation of the embedded controller is limited to two axes, allowing a single radial active magnetic bearing to be controlled. This would sufficiently demonstrate the operation of the embedded controller.

The implementation of each interface is now discussed.

4.1.1 AMB interface

4.1.1.1 Power amplifier synchronisation output

The power amplifier synchronisation output consists of a 5 V square wave with variable pulse width as mentioned in Section 3.4.1.1. The synchronisation signal is generated by buffering a

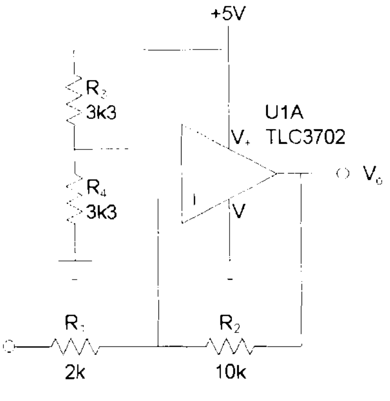


Figure 4.1: 5 V Buffer circuit

PWM output of the TMS320F2812. In order to prevent comparator chatter, a non-inverting Schmitt trigger circuit is implemented [32, 33]. The Schmitt trigger ensures that the output of the buffer only switches to 5 V after the input is greater than 3 V and then only switches to 0 V after the input is below 2 V.

An analogue voltage comparator is used to implement the buffer circuit as shown in Figure 4.1. The comparator introduces a slight delay which can be compensated for by adjusting the duty cycle of the PWM output. The simulated response of the buffer is shown in Figure 4.2.

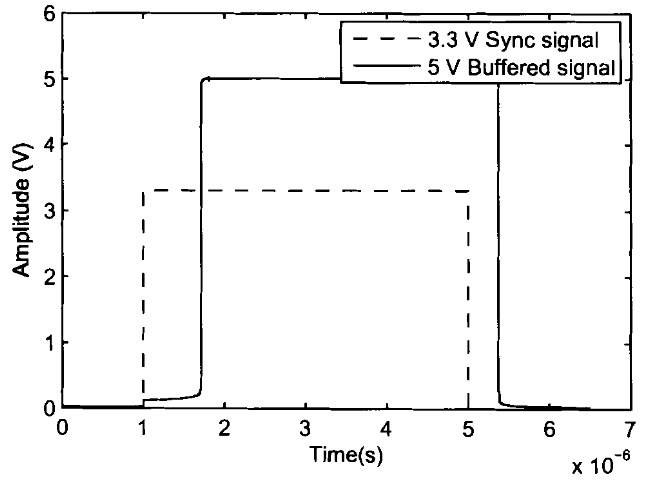


Figure 4.2: 5 V Buffer simulation

4.1.1.2 Current reference output

The anti-imaging filter transfer function given by (3.2) is implemented with the multiple feedback circuit shown in Figure 4.3 [34]. FilterPro® [34], a low-pass filter design program from Texas Instruments, is used to realise the filter circuit. The transfer function of the circuit is given by

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{-\frac{1}{RR_2C_1C_2}}{s^2 + \frac{RC_2+2R_2C_2}{RR_2C_1C_2}s + \frac{1}{RR_2C_1C_2}} \tag{4.1}$$

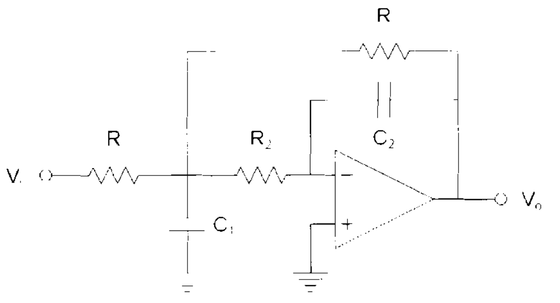


Figure 4.3: Multiple feedback circuit [34]

As seen from (4.1), the transfer function has a unity gain and inverted output. The gain required by the anti-imaging filter is realised by adding an inverting amplifier stage, as the transfer function requires the anti-imaging filter to be non-inverting. Component tolerances may cause a non-unity gain for the filter stage, but this is compensated by allowing the gain of the amplifier stage to be adjusted. The complete anti-imaging filter circuit is shown in Figure 4.4.

The anti-imaging filter circuit is simulated using OrCad® in order to verify its frequency response, the results of which is shown in Figure 4.5. From the figure it can be verified that the filter's simulated response is very close to the filter's designed response.

4.1.1.3 Position reference input

The anti-aliasing filter transfer function given by (3.5) is also implemented using the multiple feedback circuit shown in Figure 4.3 [34]. This is done because of the circuit's superior frequency

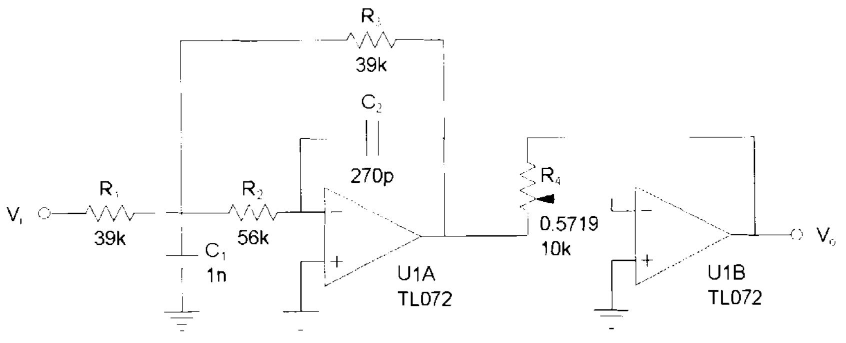


Figure 4.4: Anti-imaging filter circuit

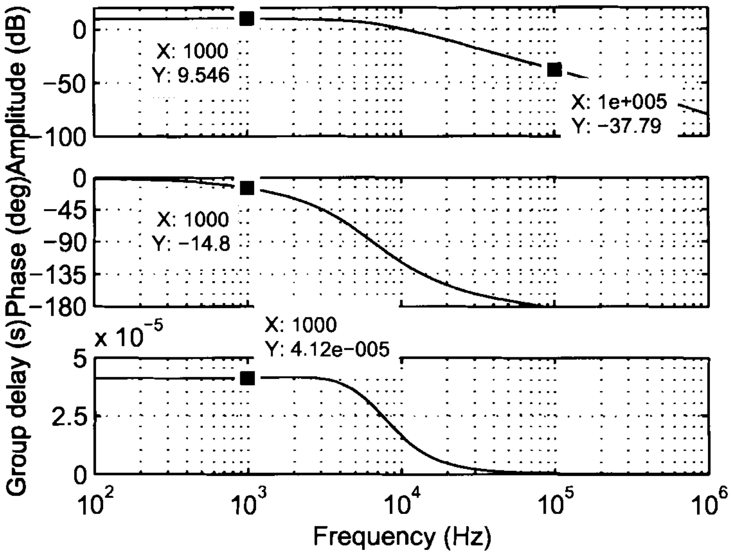


Figure 4.5: Anti-imaging filter simulation

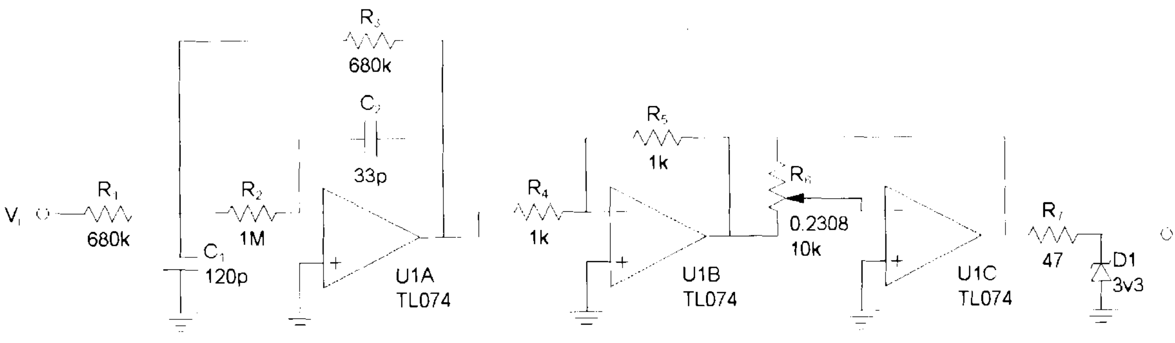


Figure 4.6: Anti-aliasing filter circuit

response. FilterPro® [34] is also used to realise the filter circuit.

The gain required by the filter transfer function is also realised by adding an inverting amplifier stage, as the required gain is less than unity. As the transfer function requires the anti-aliasing filter to be inverting, the addition of another inverting amplifier with unity gain is necessitated. Gain errors caused by component tolerances are also compensated by calibrating the gain of the amplifier stage.

In order to minimise the loading effect of the anti-aliasing filter on the dSPACE® controller, components are chosen in such a way that the input impedence of the anti-aliasing filter is very high. The complete anti-aliasing filter circuit is shown in Figure 4.6.

The anti-aliasing filter circuit is also simulated using OrCad® in order to verify its frequency response, the results of which is shown in Figure 4.7. Form the figure it can also be verified that the filter's simulated response is very close to the filter's designed response.

4.1.2 PC interface

The PC interface is designed to use a high-speed digital communication channel in order to eliminate the noise introduced by analogue signals. The communication channel selected is the SPI communication interface, as it is available on both the DSP and the dSPACE® controller [35].

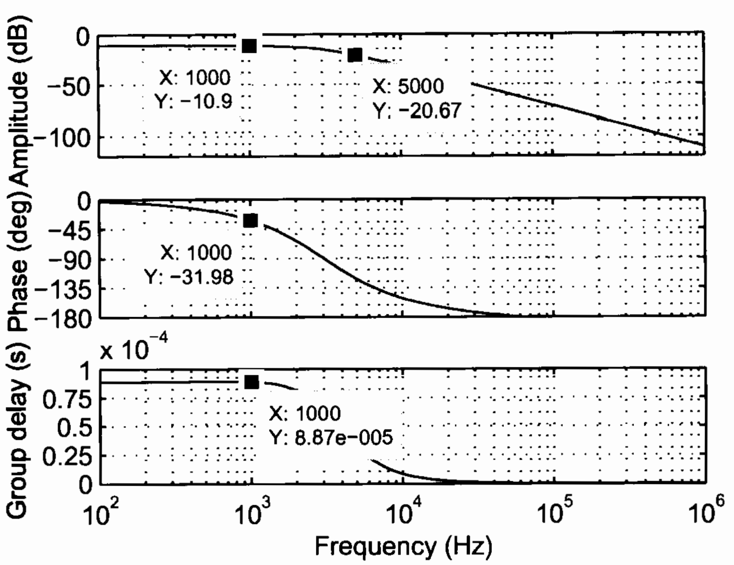


Figure 4.7: Anti-aliasing filter simulation

With the embedded controller as master device, the SPI communication channel consists of four outputs signals from the embedded controller and one input signal [30] as an additional signal is used as an user interrupt to interrupt the dSPACE® controller. The slave DSP employed by the dSPACE® controller is TTL-based, using 5 V I/O [35]. The TMS320F2812, however, is CMOS based, with 3.3 V I/O which is not 5 V tolerant [24].

The output logic levels of 3.3 V CMOS devices is compliant with the input logic levels of 5 V TTL devices [36], enabling the outputs of the embedded controller to be directly connected to the input of the dSPACE® controller [37]. The output of the dSPACE® controller is connected to the TMS320F2812 via a 10 kΩ series resistor, as this limits the current to an acceptable level. This, in turn, limits the voltage applied to the DSP's input [37].

The communication protocol is implemented on the dSPACE® controller by adding user code to the Simulink® model shown in Figure 4.8, using the RTLib® library provided by dSPACE® [38]. The user code is included on the data CD. The use of the SPI interface on the dSPACE® controller has, however, proved to be problematic. The biggest problem encountered is that communication delay between the master and slave DSP on the dSPACE® controller greatly limits the data rate of the SPI interface.

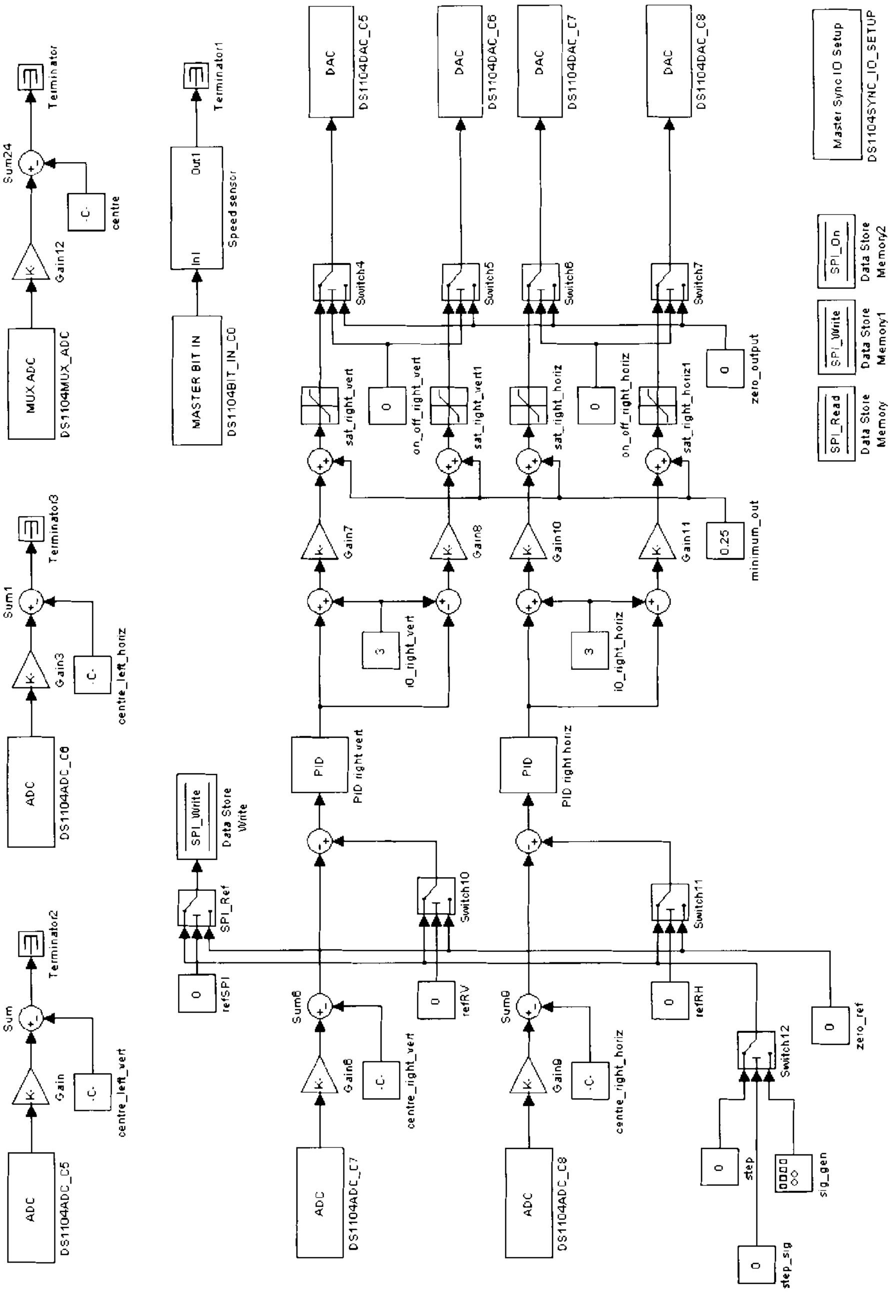


Figure 4.8: dSpace® controller Simulink® model

The bulk of the communication protocol is therefore abandoned in order to reduce the amount of data to be transferred. The data frame is reduced to 12 bits, and can therefore not include any error checking or message identification. The interpretation of received data, as well as the control parameter to be reported to the PC, is compiled into the embedded controller's firmware and can therefore not be changed during run-time.

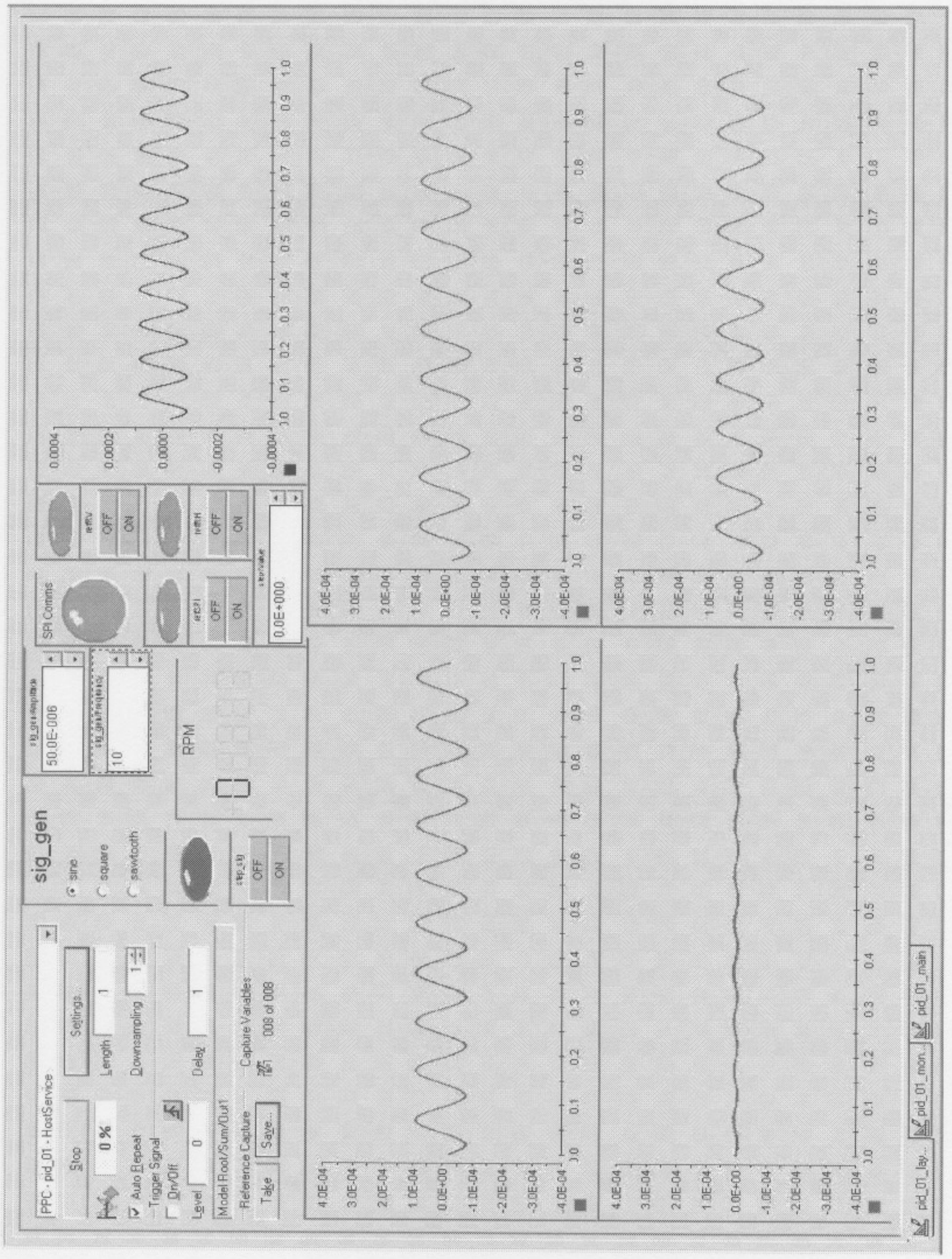
Six data bits are transmitted roughly every $50\,\mu$s using a baud rate of approximately 1.1 Mbps, as this is the highest baud rate at which the data is transmitted error free, in order to allow the dSPACE® controller adequate time for processing the data communications.

ControlDesk® is used to capture the reference signals and rotor position signals, and to control the generation of the reference signals. The ControlDesk® interface is practically identical to the existing interface, with only the controls for the left bearing omitted. Figure 4.9 shows the data capture screen of the ControlDesk® interface.

The data capture screen consists mainly of five graphs. The bottom four are from left to right, top to bottom: Left vertical position, right vertical position, left horizontal position and right horizontal position. These positions are measured by the dSPACE® controller itself. The right topmost graph is a graph of the data received from the embedded controller via the SPI communication channel. The data capture screen also has controls for generating a reference signal. This reference can be channeled to either axis of the right AMB or to the embedded controller via the SPI communication channel. Figure 4.9 shows a 50 μm sinusoidal reference which is channeled to both axes of the right AMB as well as to the embedded controller. As the embedded controller is configured to interpret received data as a position reference for the vertical axis, the reference is also applied to the left vertical axis.

4.1.3 User interface

As specified in Section 3.4.3, the user interface consists of two LEDs and a pushbutton. The activity and status LEDs are connected to general purpose input/output (GPIO) pins of the DSP (GPIOF4 and GPIOF5 respectively [39]) via $1\,\text{k}\Omega$ resistors. This effectively limits the output current of each I/O pin to roughly 1 mA, which is within the DSP's safe operating limits [24].



PPC - pid_01 - HostService

0 %

Length 1

☑ Auto Repeat

Downsampling 1

Trigger Signal

□ On/Off

Level 0

Delay 1

Model Root/Sum/Out1

Capture Variables 008 of 008

sig_gen

● sine

○ square

○ sawtooth

sig_gen/Amplitude 50.0E-006

sig_gen/Frequency 10

0.0E+000

Figure 4.9: ControlDesk® data capture window

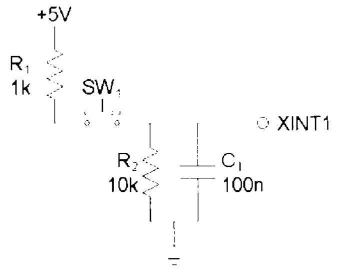


Figure 4.10: Pushbutton circuit

The pushbutton is connected to the first external interrupt (XINT1) of the DSP via the circuit shown in Figure 4.10, which limits the input current to roughly 350 μA. This enables the pushbutton to be software polled via GPIOE0 or to provide an external interrupt source [39].

4.1.4 Mains interface

The mains interface, which interfaces with the 220V AC mains power grid, comprises three separate isolated power supplies, namely the DSP power supply, analogue circuit power supply and digital circuit power supply.

The 5 V power supply provided with the DSP starter kit is used to power the DSP, as the power supply meets the DSP's power demands and is isolated from the mains grid. The analogue circuit's power supply is a regulated ±15 V supply, which includes a LED to show that the power supply is turned on and utilises a 12 V, 1.5 VA dual transformer. The ground of the analogue power supply is connected to analogue ground. The digital circuit's power supply is identical to the analogue power supply as it supplies ±15 V to the AIF and in addition includes a 5 V regulated power supply to power the Schmitt-trigger buffer circuit and the pushbutton. The digital circuit's ground is connected to digital ground. Analogue and digital ground is connected at the ADC in order to prevent noise. This is also shown in Figure 4.11.

4.2 System integration

Figure 4.11 shows how the individual system interfaces are integrated into a single unit. The figure shows how the outputs of the existing dSPACE® controller are disconnected and how signals are fed to the embedded controller, AMB interface box and dSPACE® controller.

The existing signal bus consists of two 50-wire ribbon cables, designated Port A and Port B. The signals are injected into the existing signal bus by inserting two pieces of 50-wire ribbon cable where the dSPACE® controller connects to the AMB interface box, using a DB-50 plug to connect to the AMB interface box and a DB-50 socket to connect to the dSPACE® controller. The pin assignment of the connectors to the AMB interface box and dSPACE® controller are shown in Figures 4.12(a) and (b) respectively.

The grounding policy of the embedded controller is designed to minimise noise effects. As seen in Figure 4.11, analogue and digital ground is seperated from each other and only connected at the ADC in order to prevent ground loops. The power amplifier current references are also disconnected from dSPACE® and paired with the specific signal ground used in the AMB interface box. As the current reference is sensed via an optocoupler on the power amplifiers, the effect is therefore to transmit the current reference differentially. This minimises the effect of common mode noise on the current reference. Each signal and its ground return path are grouped in Figures 4.12(a) and (b).

Figure 4.13 shows a photograph of the embedded controller. The 50-wire ribbon connectors are clearly visible, connecting to the digital circuit on the left hand side, as well as to the analogue circuit on the right. Both the analogue and digital circuits are then connected to the DSP development board via ribbon cables. The layout of the digital and analogue circuits are similar to the layout shown of Figure 4.11.

4.3 DSP firmware implementation

The DSP firmware is first compiled using the VisSim® model in order to aid in rapid prototyping of the control algorithm. The implementation of the SPI communication channel is very limited in

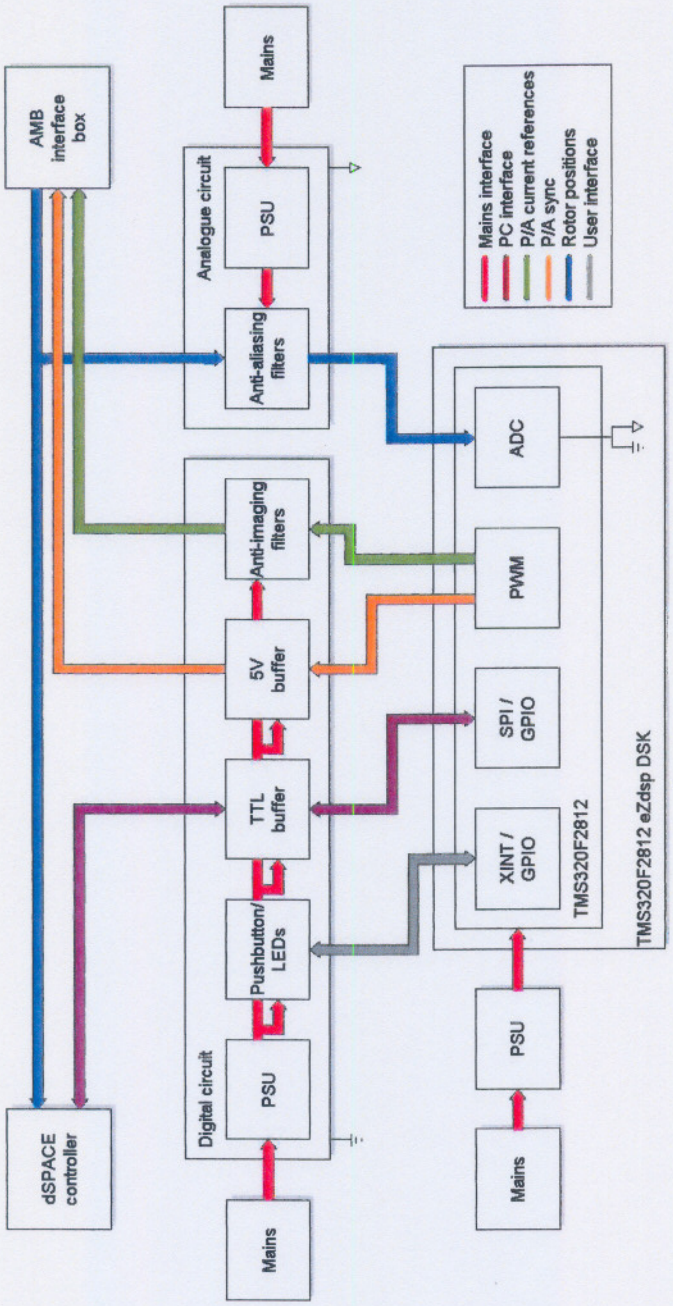
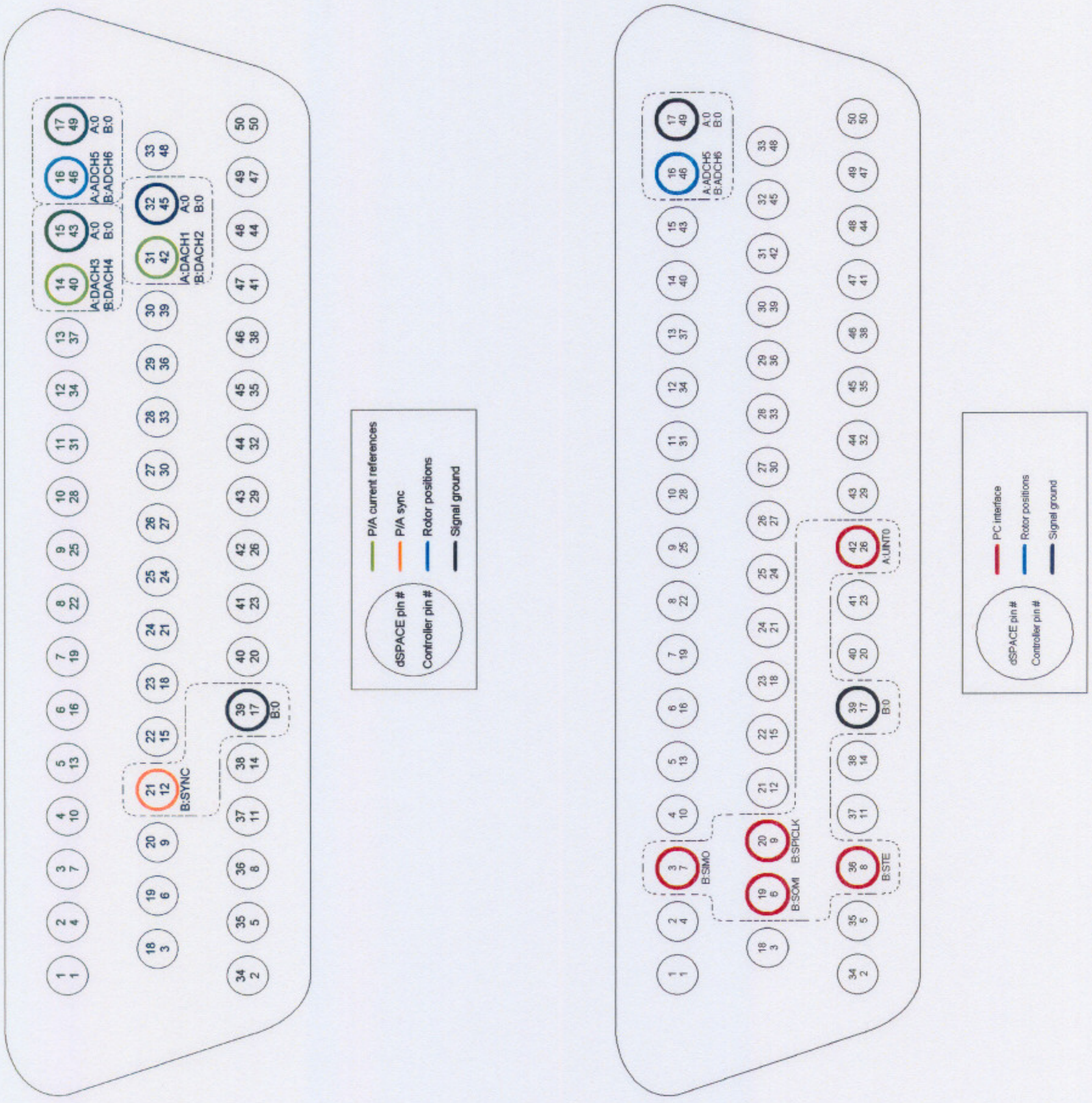


Figure 4.11: System integration



(a) (b)

Figure 4.12: System connectors

(a) AMB interface box (b) dSPACE controller

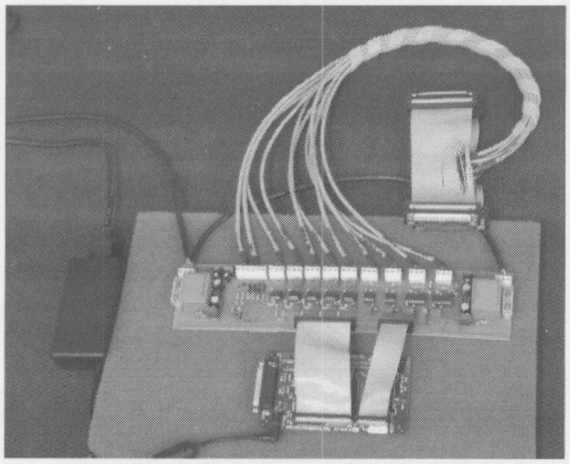


Figure 4.13: Photograph of embedded controller

VisSim® and overheads caused the cycle time to be very low, which necessitated the decision to implement the DSP firmware using an object-orientated C approach [40].

The firmware implementation is based on the VisSim® simulation, including the signal scaling as shown in Figure 4.14. This reduces problems associated with number scaling and representation. In order to further aid in the C implementation, examples provided by Texas Instruments were used as a base for code development.

In order to demonstrate the capacity of the DSP to perform the necessary operations, five PID controllers are implemented in the software, as well as the SPI communication.

In order to minimise the processing delay introduced by the controller, a single axis's position is sampled, its PID loop executed and its output updated before the next axis's position is sampled [2, 12]. In order to further improve the response of the embedded controller, the current transfer characteristic of each of the four power amplifiers used are determined and used to correct gain and offset errors with the controller firmware.

The different aspects of the peripheral configuration are discussed in the following paragraphs. The firmware source code is included on the data CD.

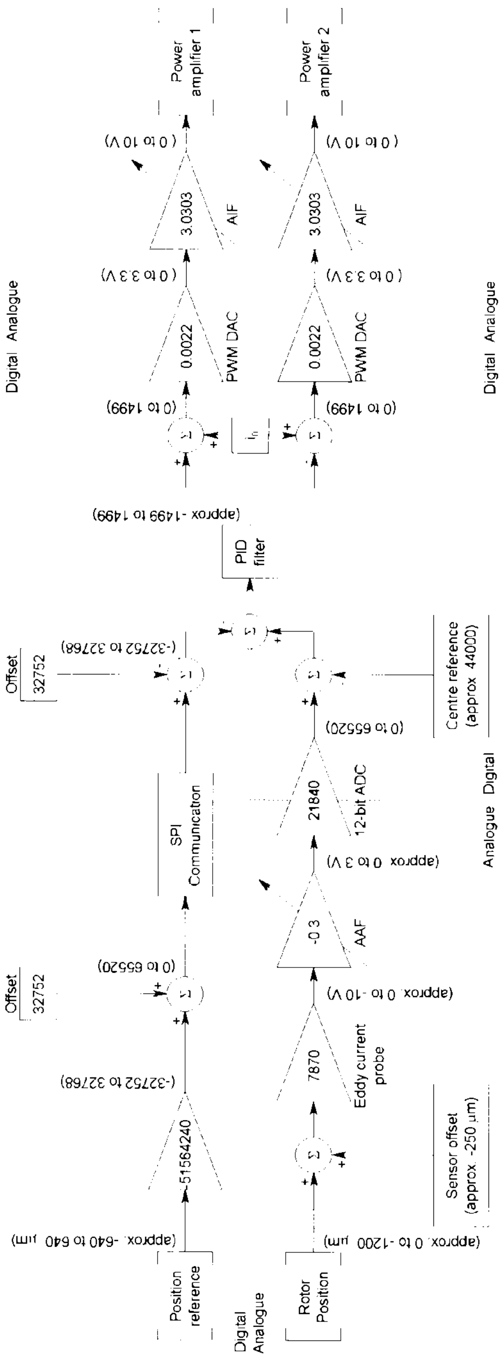


Figure 4.14: Signal scaling

4.3.1 CPU timer

The CPU timer is configured to overflow and generate a CPU interrupt at a rate of 10 kHz and is used to toggle the activity LED. The startup of the PID filter is also timed to enable the integrator for five seconds during startup, if desired, in order to centre the rotor, whereafter it is disabled for normal operation. The timer is used to re-enable the ADC start of conversion at a rate of 10 kHz, timing the control loops to be executed at a rate of 10 kHz.

4.3.2 General purpose input/output

All I/O pins of the DSP are configured as GPIO inputs in order to put the pins in a high-impedance state, except for the I/O pins used for the LEDs, the pin used to interrupt the dSPACE® controller and the PWM, SPI and first external interrupt which are selected as peripheral pins. Input qualification is enabled for the external interrupt in order to limit contact bounce from the pushbutton [39].

4.3.3 Event manager

The event manager is used to generate the generate the PWM signals based on the configuration of four timers. The timers are configured to continuously increment up to a fixed period of 1499 and then roll-over to 0, which results in a PWM frequency of 100 kHz [31].

The event manager is configured in such a way that the four timers operate in pairs of two, with two timers starting simultaneously with the same period [31]. One timer is used to generate the PWM signal, while the other is used to trigger the ADC start of conversion. The timer used to trigger the ADC is preloaded with a value in order to offset the ADC trigger with respect to the PWM signal.

4.3.4 ADC

The ADC is configured to convert a single analogue channel at 2.8 Msps after a start of conversion trigger is received from the event manager [29]. The ADC is configured to generate an interrupt at the end of each conversion sequence. This interrupt is used to process the analogue value sampled and run the control loop if necessary. Further ADC conversions are disabled before the control loop is executed, and re-enabled at the end of the control loop if a sequence of five control cycles have not been completed yet.

Two precision external voltage references are used to correct the ADC gain and offset errors. They are sampled at the beginning of every sequence of control loops. A running average of the references' converted values are then updated and the ADC calibration parameters calculated. These parameters are then used to adjust the converted position values.

4.3.5 SPI

The SPI is configured as master device to transmit 6 bit data at a rate of 1.1 Mbps, using the transmit and receive FIFO buffers with both the transmit and receive FIFOs generating a processor interrupt [30]. This enables the DSP to send an entire frame without software overheads.

Data transmission is started at the end of each control loop, whereafter received data is processed every 6 bits. Upon processing the received data, the dSPACE® controller is interrupted in order to process its received data and buffer data for the next transmission.

4.3.6 External interrupt

The push-button connected to the first external interrupt is configured to generate an interrupt on the rising edge of each pulse. The push-button is used to change the state of the controller from

- Off, to
- On without the integrator, to
- On with the integrator, to
- Off, etc.

4.4 Conclusion

The embedded controller is implemented using a TMS320F2812 DSP. The DSP is capable of controlling a 5 DOF active magnetic bearing, while the interface is only implemented for a 2 DOF active magnetic bearing. The embedded controller utilises SPI as a digital communication channel between itself and the dSPACE® controller reporting the rotor position of a preselected axis and receiving a position reference for the same axis.

The firmware for the embedded controller is implemented with Code Composer Studio® using an object-orientated C approach. The firmware is interrupt driven ensuring accurate timing of the control loops. The firmware is also structured to be time-critical, ensuring that the best possible cycle time can be realised for each axis's control loop.

Chapter 5

System verification

This chapter discusses the verification of each system interface and the firmware execution. Hereafter it proceeds to discuss the verification of the complete AMB system by determining the system sensitivity and equivalent stiffness and damping, which is compared to the simulated results and the existing dSPACE® controller.

5.1 Interface verification

5.1.1 AMB interface

As the AMB interface translates the control signals, the correct operation of the interfaces is critical to the performance of the AMB system. The sub-interfaces are now verified.

5.1.1.1 Power amplifier synchronisation output

As mentioned in Section 4.1.1.1, the power amplifier synchronisation circuit is implemented using a Schmitt-trigger circuit. Figure 5.1 shows the output waveform of the buffer circuit. As seen from the figure, the analogue comparator introduces a time delay into the buffered synchronisation signal. The duty cycle of the input signal is therefore increased to approximately 60 % in order to generate the required duty cycle of 40 %.

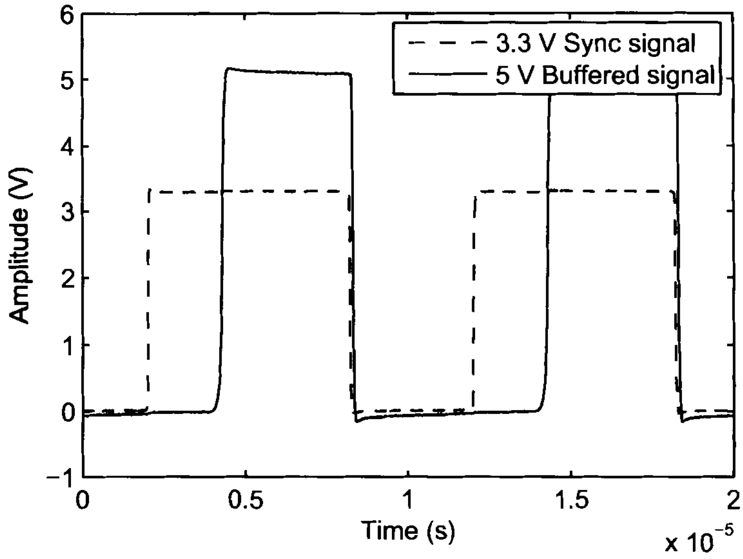


Figure 5.1: Power amplifier synchronisation output

5.1.1.2 Current reference output

The current reference output circuit consists of the anti-imaging filters. In order to verify the implementation of the designed filter circuit, the frequency response of the filter is measured, the result of which is shown in Figure 5.2.

If the realised response is compared to the simulated response, it can be seen that the responses correlate well, as summarised in Table 5.1. The group delay displays a slight deviation from the expected result at very low frequencies, but this can be attributed to inaccurate measurement as it is very difficult to accurately determine the zero crossing of the slow-changing input and output voltages.

The realised gain is slightly less than the designed value as the filter gain is calibrated to have a DC gain of exactly 3.0303 for an input of 3.3 V in order to compensate for the voltage offset introduced by the operational amplifier.

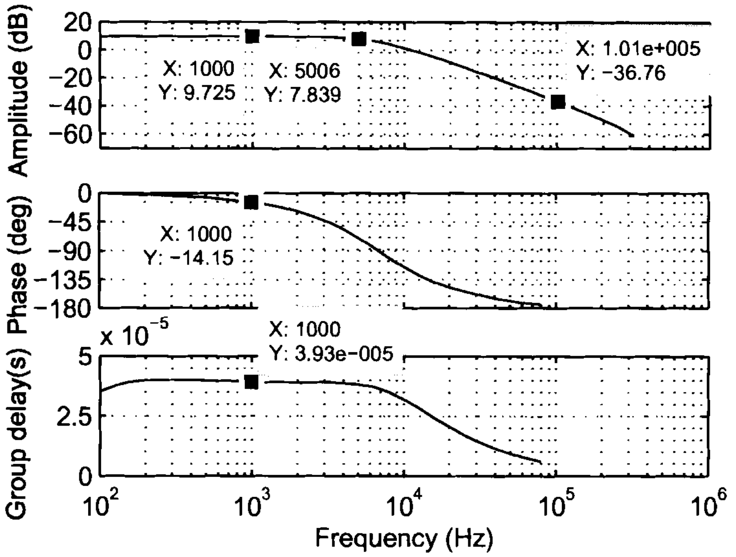


Figure 5.2: Anti-imaging filter response

Table 5.1: Anti-imaging filter realisation

	Designed response		**Simulated response**		**Realised response**	
Frequency (kHz)	**Amplitude (dB)**	**Group delay (μs)**	**Amplitude (dB)**	**Group delay (μs)**	**Amplitude (dB)**	**Group delay (μs)**
1	9.523	42.97	9.546	41.2	9.725	39.3
5	6.63		6.996		7.839	
100	-38.1		-37.788		-36.76	

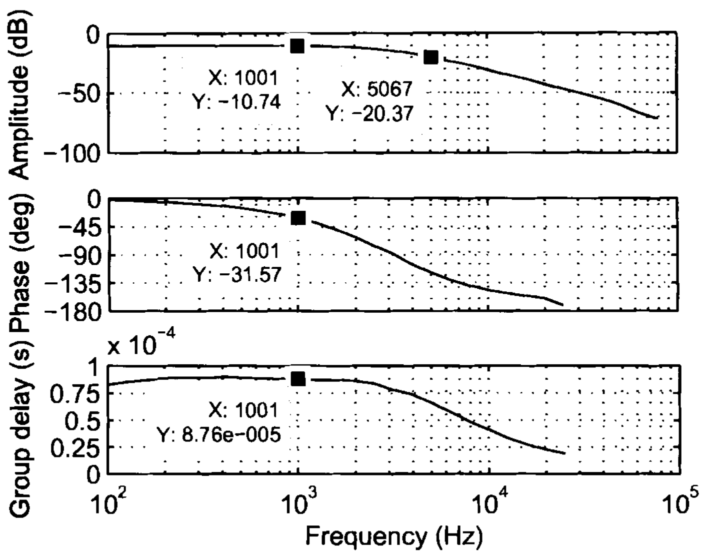


Figure 5.3: Anti-aliasing filter response

5.1.1.3 Position reference input

The position reference input circuit consists of the anti-aliasing filters. In order to verify the implementation of the designed filter circuit, the frequency response of the filter is measured, the result of which is shown in Figure 5.3.

If the realised response is compared to the simulated response, it can be seen that the responses correlate well, as summarised in Table 5.2. The group delay also displays a slight deviation from the expected result at very low frequencies, but again this can be attributed to inaccurate measurement. The realised gain is also slightly lower than the designed value as the filter gain is calibrated to have a DC gain of exactly -0.3 for the centre rotor position input in order to compensate for the voltage offset introduced by the operational amplifier.

5.1.2 PC interface

The PC interface is verified by sending a square wave to the embedded controller and echoing it back to the PC. The time difference between the sent and received signals can then easily be

Table 5.2: Anti-aliasing filter realisation

	Designed response		Simulated response		Realised response	
Frequency (kHz)	**Amplitude (dB)**	**Group delay (μs)**	**Amplitude (dB)**	**Group delay (μs)**	**Amplitude (dB)**	**Group delay (μs)**
1	-10.95	90.52	-10.90	88.65	-10.74	87.6
5	-20.89		-20.73		-20.37	

measured and represents the round-trip time of the SPI communication. Figure 5.4 displays a typical delay, which is determined to be 500 μs. This time delay is used to correct the time difference between the reference and position signals when analysing the step response of the AMB system.

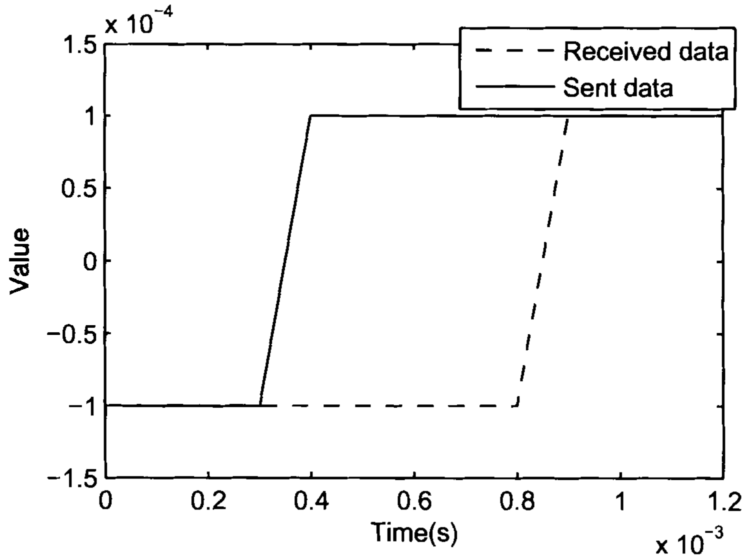


Figure 5.4: Round-trip time measurement

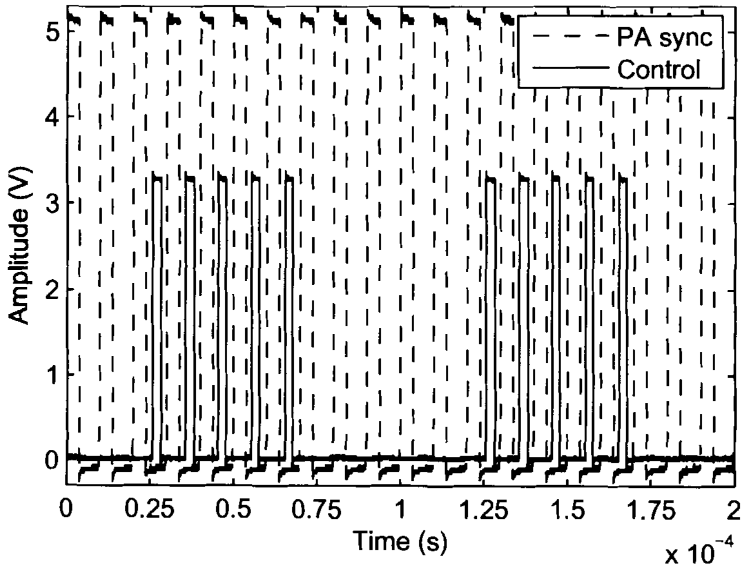


Figure 5.5: Firmware execution measurement

5.2 Firmware execution verification

The execution of the embedded controller's control loops are verified by toggling a general purpose input/output pin upon the execution of each control loop. The output is measured along with the power amplifier synchronisation output as reference. The result is shown in Figure 5.5.

From the figure it can be seen that five control loops are executed every 100 μs, synchronised with the falling edges of the power amplifier synchronisation signal. The time delay between a falling edge of the power amplifier synchronisation signal and the start of a control loop's execution is mostly accounted for by the acquisition time of the ADC. The maximum duration between the start of conversion of the ADC and the end of the control loop is measured to be approximately 5 μs. The timely execution of the control loops are therefore verified.

5.3 Overall system verification

The performance of the embedded controller is verified by determining the system sensitivity, static stiffness and equivalent stiffness and damping. The PID constants are the same as those used

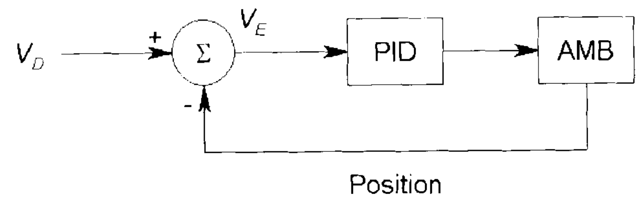


Figure 5.6: Sensitivity gain measurement

for the dSPACE® controller, namely: $K_p = 12500$, $K_i = 0$, $K_d = 37.5$.

5.3.1 System sensitivity

The sensitivity of the AMB system utilising the embedded controller is measured using the measurement procedure proposed in [41]. The sensitivity gain is determined by injecting a disturbance frequency (V_D) into the controller whilst keeping the position reference zero, as shown in Figure 5.6. The error signal as seen by the PID controller (V_E) is then measured. The sensitivity gain is then given by the ratio of the signals in the frequency domain, as seen in (5.1).

$$G(s) = \frac{V_E(s)}{V_D(s)} \tag{5.1}$$

In order to conduct the test, the disturbance is digitally sent from the PC to the embedded controller. The embedded controller is then configured to report its calculated error signal back to the PC. Both signals are stored on the PC for later analysis. The maximum frequency used for the test is determined by the larger of three times the rated speed and 2 kHz. For digital controllers the maximum frequency also cannot exceed half of the sampling frequency, as stated by the Nyquist theorem discussed in Section 2.2.2 [41].

A frequency analysis of the error signal is done in order to determine the amplitude of the frequency component corresponding to the frequency of the injected disturbance. $G(s)$ is then calculated for each frequency.

The results of the system sensitivity test conducted for the vertical and horizontal axes are shown in Figures 5.7(a) and (b) respectively. The peak sensitivity of the system is then determined to be

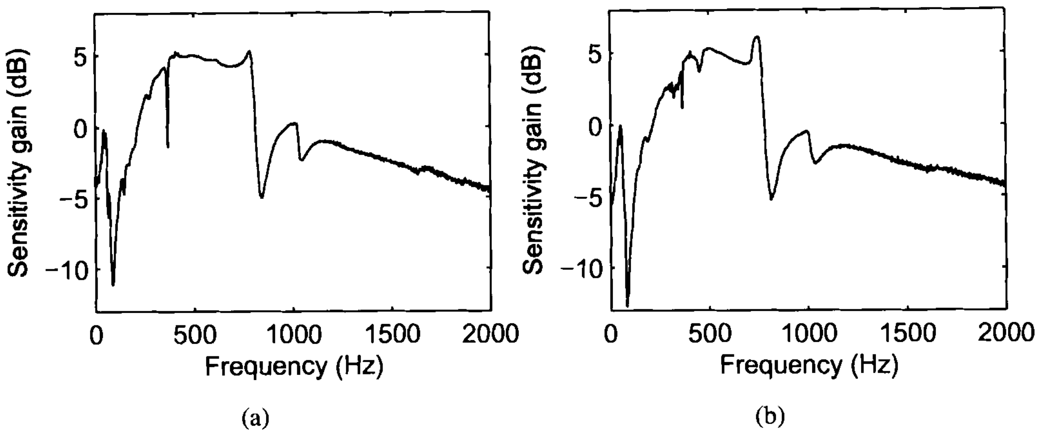


Figure 5.7: Embedded controller system sensitivity

(a) Left vertical (b) Left horizontal

6.085 dB. As the Zone A [41] requirements for AMBs are that the peak sensitivity of any controlled axis cannot exceed 8 dB, the system falls well within the Zone A requirements.

When the sensitivities of the left horizontal axis, while using the embedded controller and the dSPACE® controller shown in Figure 5.8 are compared, it can be seen that the peak sensitivity of the embedded controller is less than that of the dSPACE® controller. It can also be seen that the embedded controller does not reveal a sensitivity peak at 375 Hz as the case with the dSPACE® controller. The embedded controller does however reveal a sensitivity peak at 800 Hz unlike the dSPACE® controller due to the time delay introduced by the AAF and AIF.

5.3.2 Static stiffness

The static stiffness of the AMB is measured by determining the deflection of the rotor due to a static force exerted on the rotor. In this case, the defection of the rotor due to gravity is measured on the vertical axis. Since there are two AMBs opposing gravity, half the rotor mass is used for the calculation. The static stiffness is calculated using (5.2).

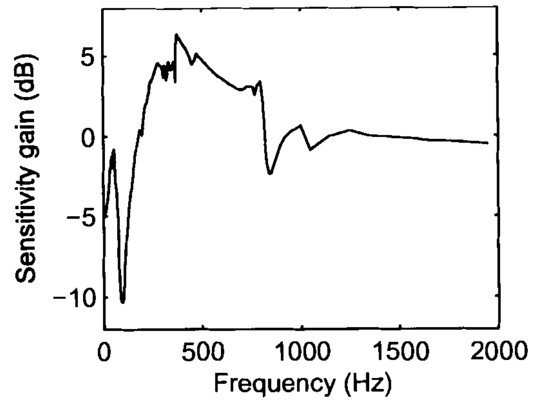


Figure 5.8: dSPACE® controller sensitivity - Left horizontal

$$\begin{aligned} k_s &= \frac{F}{x} \\ &= \frac{mg}{x} \text{ N/m} \end{aligned} \tag{5.2}$$

The deflection due to gravity is measured to be 80.1 μm, which results in a static stiffness k_s of 472.74 $\times$ 10^3 N/m. The simulated deflection is 76.8 μm, which yields a static stiffness of 493.05 $\times$ 10^3 N/m. The experimental and simulated results therefore differ by 4.1%.

There are two major causes for the difference between the experimental and simulated results. The first is possible gain and offset errors introduced by the interfacing circuits, while the second is the offset of the magnetic centre of the AMB with respect to the sensor's centre position which causes the stiffness of the AMB to be asymmetrical.

5.3.3 Equivalent stiffness and damping

In [10] it is discussed how an AMB system is linearised about its working point, which enables the AMB system to be modeled using a second order transfer function. The equivalent second order stiffness and damping of the AMB system can be approximated by analysing a step response of the AMB. In order to eliminate the effect of gravity, the analysis is done using the horizontal axis.

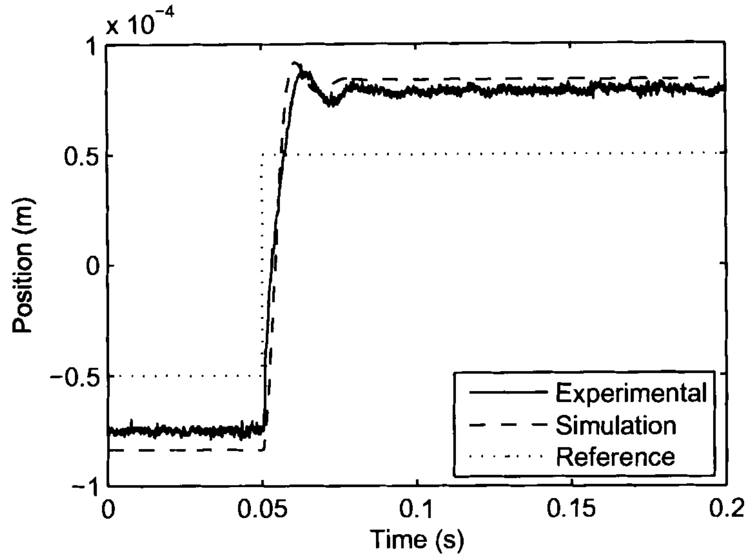


Figure 5.9: Embedded controller 100 μm horizontal step

Figure 5.9 shows the step response of the AMB system using the embedded controller for a 100 μm step on the horizontal axis. Firstly, it must be noted that the steady-state error is not the same in the positive and negative direction. This can be explained by the fact that sensor's centre position does not coincide with the magnetic centre as the forces exerted by the electromagnets are non-linear in nature. This results in the system's stiffness being unsymmetrical about the sensor's centre.

The noise in the step response prohibits the direct determination of the percentage overshoot (PO) and damped oscillation period (T_d) from the step response. The data is therefore averaged and PO and T_d is determined to be 4.97 % and 0.0160 s respectively.

In order to determine the equivalent second order stiffness k_{eq} and damping b_{eq}, the following equations are used [10, 42]:

$$PO = 100e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{5.3}$$

$$\omega_d = \frac{2\pi}{T_d} \tag{5.4}$$

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} \tag{5.5}$$

$$k_{eq} = \omega_n^2 m \tag{5.6}$$

$$b_{eq} = 2\zeta\sqrt{k_{eq}m} \tag{5.7}$$

The equivalent second order stiffness and damping is then determined to be 1.1389 × 10^6 N/m and 2.8971 × 10^3 Ns/m respectively.

The results of the simulation used for the firmware design does not display the level of overshoot seen in the practical results. The simulation is therefore extended to include other known sources of dynamics in the AMB system such as the RC filters in the position sensors and power amplifiers. However this still does not reproduce the overshoot displayed by both the embedded and dSPACE® controllers. One of the only further possible causes for the overshoot can be the rotor dynamics, which is not included in the simulation. A first-order approximation of the rotor dynamics is included in the simulation by the addition of a pole and zero, which yields the simulation results shown in Figure 5.9.

The step response of the AMB simulation is analysed and PO and T_d is determined to be 4.21 % and 0.0174 s respectively, which results in a second order equivalent stiffness and damping of 1.0153 × 10^6 N/m and 2.8115 × 10^3 Ns/m respectively.

A step response is also measured on the horizontal axis using the dSPACE controller, which is shown in Figure 5.10. The step response is analysed and PO and T_d is determined to be 4.10 % and 0.0178 s respectively. This results in an equivalent second order stiffness of 0.9783 × 10^6 N/m and equivalent damping of 2.7710 × 10^3 Ns/m. It should be noted that the dSPACE® controller is compensated to control about the magnetic centre of the AMB, resulting in a symmetrical steady-state error.

When compared to the dSPACE® controller, the embedded controller's equivalent stiffness and damping is slightly more than that of the dSPACE® controller. The simulation's second order

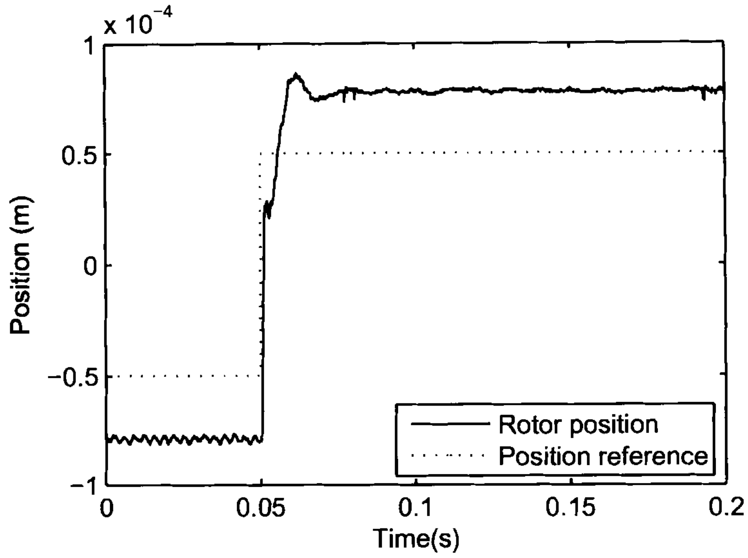


Figure 5.10: dSPACE® controller 100 μm horizontal step

equivalent stiffness and damping also shows good correlation with the practical results when the rotor dynamics are included.

5.4 Conclusion

The embedded controller is successfully verified, and conforms to the Zone A requirements for AMB controllers. The performance of the embedded controller shows a good correlation with that of the dSPACE® controller, as well as the simulated results. The equivalent stiffness and damping of the embedded controller is slightly higher than the dSPACE® controller's, as is predicted by the simulation.

The difference between the simulated and experimental results may partly be attributed to the inaccuracy of the simulation model, as the simulation model does not include an accurate model of the rotor dynamics. Other possible sources for the difference between the simulated and experimental results may be inaccurate filter gains, as well as non-linearities and offsets introduced by non-ideal operational amplifiers.

Chapter 6

Conclusions and recommendations

6.1 Development tools

The choice of the TMS320F2812 as DSP platform proved to be correct as the on-chip peripherals are adequate for the suspension of the double radial AMB. The DSP has also proved to have enough processing power to run five PID loops and manage the high-speed digital communication with the PC.

The eZdsp® F2182 DSK also aided in the development of the embedded controller as it is supported by both VisSim® and Code Composer Studio®. It also provided easy access to the peripherals of the DSP and the on-board JTAG emulator proved to be invaluable during code development.

Although it was hoped that VisSim® would provide a high-level programming interface for the DSP, it proved to be too limited in the configuration and use of the on-chip peripherals and its software overheads extended the cycle time of the embedded controller to such an extent that two PID loops could barely be executed at a rate of 10 kHz. It did, however, prove to be invaluable during the simulation of the firmware in order to verify the fixed-point scaling of the PID filter.

6.2 Interfacing

Interfacing proved to be one of the biggest concerns regarding the embedded controller, especially since the controller operates in a very noisy environment and the input voltage is very small. The embedded controller is then very sensitive to offset voltages, introduced by non-ideal operational amplifiers, as well as any slight deviation in filter gains, caused by non-linearities and temperature drift.

Although care was taken to scale the position input to the full range of the ADC, a DC offset was not used. It is predicted that the noise can be halved if the scaling utilises a DC offset, as the effective range used by the ADC is roughly 1.25 to 2.75 V, where the available range is 0 to 3 V. The centre position of each axis is then set externally via the DC offset, which would fix the centre offset used by the firmware to $1/2\ FS$ (full scale). As the gain and offset of the input filter are then set separately, the DC offset introduced by the operational amplifier can be compensated for.

6.3 Firmware implementation

The firmware for the embedded controller was successfully implemented with Code Composer Studio® using an object-orientated C approach, which simplified the firmware conceptualisation. The firmware only uses integer multiplication and power-of-two division in order to get the cycle time as short as possible.

A single axis's position is sampled, its PID loop executed and its current reference updated in less than 5 μs, which is short compared to the 130 μs time delay introduced by the interface filters. This would not have been possible if the firmware was not coded using C.

Because of the optimised C code, the embedded controller is capable of suspending a 5 DOF AMB and of the implementation of more advanced control algorithms.

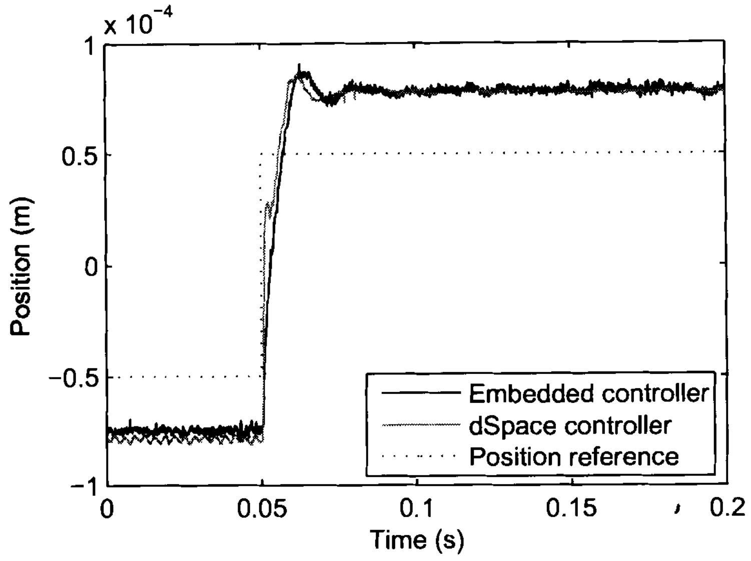


Figure 6.1: 100 μm step response, embedded controller and dSPACE® controller

6.4 Future work

6.4.1 Simulation refinement

As mentioned is Section 5.3.3, there is a major difference in the level of overshoot predicted by the simulation model and the amount of overshoot observerd in both the responses of the embedded controller and the dSPACE® controller seen in Figure 5.9.

As the simulation already includes all dominant dynamics such as the controller pole, interface filters, power amplifiers, magnets, and magnetic forces on the rotor, the only other unmodelled non-linearity is the rotor dynamics. A crude attempt was made to model the rotor dynamics by introducing a pole and zero into the system, but the result is still unsatisfactory. Further investigation is therefore definitely needed to develop an accurate model of the rotor dynamics.

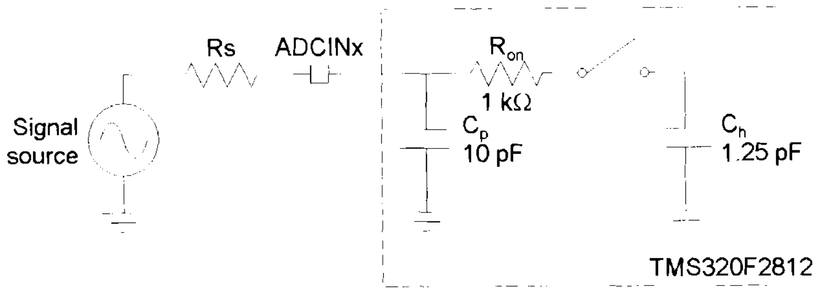


Figure 6.2: ADC analogue input impedance model [24]

6.4.2 Interfacing

Although great care was taken to select a filter topology that would even attenuate frequency components beyond the bandwidth of the operational amplifiers, there is still a noticeable amount of noise evident in the position signal measured by the embedded controller. This is most probably due to noise being superimposed on the position signal as it is relayed from the analogue interface circuit to the DSP.

It should therefore be investigated whether an AAF is necessary at all. If synchronised sampling is used and the position signal is scaled to fully utilise the ADC range, it could be less susceptible to noise. The delay caused by the AAF could be eliminated by eliminating the AAF as a whole, which could greatly improve the controller's response. If an AAF is then deemed to be necessary, it could be possible that only a simple RC filter would suffice. It could also be possible to construct this RC filter using the internal capacitance (C_p) of the ADC converter (see Figure 6.2), which could minimise the noise on the converted signal.

If an active filter is necessary, a detailed study should be conducted on the realisation of the active filter. This should include factors such as wire-wound vs. carbon resistors, the capacitors used, op-amp selection and even component placement and signal routing. It is also recommended that a PCB layout is done which incorporates the DSP, the necessary power supplies and the interface filters, utilising a multi-layer PCB with a ground plane. If the design is done keeping EMC in mind, it should also prove to be less sensitive to radiated noise.

The use of conventional DACs should also be investigated, as the AMB system does not need an AIF due to its transfer function. The AIF was necessitated here because the PWM signals were used as DAC outputs. If both the AAF and AIF could be eliminated, the overall time delay of the controller could be in the range of 5 μs, which would greatly improve the controller's response.

6.4.3 Firmware

As the object orientated C approach proved to be very effective, it is recommended that a library of object-orientated C code is designed and developed. This would greatly simplify the C coding of the DSP and make the C implementation available to the entire MBMC research group.

Although the embedded controller is capable of stand-alone operation, it has to be programmed using a PC after power-up. This can be corrected by porting the in-memory application to a flash application. The embedded controller would then automatically boot its firmware at startup and would therefore operate without being connected to a PC.

The amount of cycle time available on the embedded controller makes it ideal for the implementation of advanced control algorithms, which would greatly improve the system performance. Although it may be necessary to change the firmware to be less time-critical if the cycle time nears 10 μs, it would still be a good area for further investigation.

6.5 Conclusion

The main aim of this project was to develop an embedded controller for an AMB system. The embedded controller is successfully implemented for a single radial AMB using a TMS320F2812 DSP while it is capable of suspending a 5 DOF AMB.

The embedded controller paves the way for the development of embedded control systems in the MBMC research group and can be used to investigate various aspects of embedded digital control.

References

[1] M. S. Sarma, *Electric Machines Steady-State Theory and Dynamic Performance*, 2nd ed. Boston, Massachusetts: PSW Publishing Company, 1996.

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

[3] S. Pradhananga, "Experimental validation of a smart-bias active magnet bearing controller," Master's thesis, Louisiana State University and Agricultural and Mechanical College, Baton Rouge, LA, August 2004.

[4] Y. Miyagawa, M. Taniguchi, and H. Ueyama, "Development of active magnetic bearing system with digital control," *KOYO Engineering Journal English Edition*, no. 155E, pp. 34–41, 1999.

[5] 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, no. 1, pp. 37–49, January 2001.

[6] S. Liu, Z. Bian, and C. Yin, "The application of high speed and optimal control in electromagnetic bearing system based on dsp," in *ISCORMA 2*, Gdansk, Poland, August 2003, pp. 1–4.

[7] F.-M. Renner, K.-J. Hoffmann, R. Markert, and M. Glesner, "Design methodology of application specific integrated circuits for mechatronic systems," *Microprocessors and Microsystems*, vol. 24, no. 2, pp. 95–103, 17 April 2000.

[8] H. Ueyama, "Recent trends in development of active magnetic bearings," *KOYO Engineering Journal English Edition*, no. 163E, pp. 24–31, 2003.

[9] N. Skricka and R. Markert, "Improvements in the integration of active magnetic bearings," *Mechatronics*, vol. 12, no. 8, pp. 1059–1068, 2002.

[10] E. O. Ranft, "The development of a flexible rotor active magnetic bearing system," Master's thesis, School of Electrical and Electronic Engineering, North-West University Potchefstroom Campus, Potchefstroom, May 2005.

[11] J. N. D. Ireland, B. Burton, and R. G. Harley, "Implementation of induction motor FOC with embedded DSP interface and software for rapid prototyping electric motor traction systems," in *SAUPEC 2005*, Durban, South Africa, January 2005.

[12] H. Bleuler, C. Gähler, R. Herzog, R. Larsonneur, T. Mizuno, and R. Siegwart, "Application of digital signal processors for industrial magnetic bearings," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 280–289, December 1994.

[13] W. Kester and J. Bryant, "Sampled data systems," in *Mixed-Signal and DSP Design Techniques by the technical staff of Analog Devices*, W. Kester, Ed. Burlington, Massachusetts: Newnes, 2003.

[14] A. C. Fischer-Cripps, *Newnes Interacing Companion.* Oxford: Newnes, 2002.

[15] S. W. Smith, *Digital Signal Processing a Practical Guide for Engineers and Scientists.* Burlington, Massachusetts: Newnes, 2003.

[16] W. Kester and J. Bryant, "ADCs for DSP applications," in *Mixed-Signal and DSP Design Techniques by the technical staff of Analog Devices*, W. Kester, Ed. Burlington, Massachusetts: Newnes, 2003.

[17] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing a Practical Approach*, 2nd ed. Harlow, Essex: Prentice Hall, 2002.

[18] E. Maslen, "Digital controller hardware," in *Magnetic Bearing Short Course.* Alexandrea, Virginia: University of Virginia, August 19-20 1997.

[19] C. L. Philips and H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed. Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[20] Texas Instruments, "DSP selection guide," 2Q 2006.

[21] G. Daryanani, *Principles of Active Network Synthesis and Design*. New York: Wiley, 1976.

[22] C. H. Marais, "High speed flexible rotor active magentic bearing control," Master's thesis, School of Electrical, Electronic and Computer Engineering, North-West University, Potchefstroom Campus, May 2006.

[23] Texas Instruments, "Motor control overview," 3Q 2004.

[24] *TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors Data Manual*, Texas Instruments, December 2004, Literature number SPRS174L.

[25] Texas Instruments, "TMS320F2812 eZdsp DSP starter kit (DSK) product bulletin," 2002.

[26] Visual Solutions, "VisSim Embedded Controls Developer."

[27] D. M. Alter, "Using PWM output as a digital-to-analog converter on a TMS320C240 DSP," Texas Instruments, Application report SPRA490, November 1998.

[28] I. Ahmed, "Implementation of PID and deadbeat controllers with the TMS320 family," Texas Instruments, Application report SPRA083, 1997.

[29] *TMS320x281x DSP Analog-to-Digital Converter (ADC) Reference Guide*, Texas Instruments, November 2004, Literature number SPRU060B.

[30] *TMS320x281x, 280x DSP Serial Peripheral Interface (SPI) Reference Guide*, Texas Instruments, November 2004, Literature number SPRU059B.

[31] *TMS320x281x DSP Event Manager (EV) Reference Guide*, Texas Instruments, November 2004, Literature number SPRU065C.

[32] S. Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, 2nd ed. Boston, Massachusetts: McGraw-Hill, 1998.

[33] D. A. Neamen, *Electronic Circuit Analysis and Design.* Boston, Massachusetts: McGraw-Hill, 1996.

[34] J. Bishop, B. Trump, and R. M. Stitt, "FilterPro MFB and Sallen-Key low-pass filter design program," Texas Instruments, Application report SBFA001A, November 2001.

[35] *DS1104 R&D Controller Board Hardware Installation and Configuration For Release 4.1*, dSPACE, March 2004.

[36] W. Kester, "Hardware design techniques," in *Mixed-Signal and DSP Design Techniques by the technical staff of Analog Devices*, W. Kester, Ed. Burlington, Massachusetts: Newnes, 2003.

[37] Z. Yu, "3.3V DSP for digital motor control," Texas Instruments, Application report SPRA550, June 1999.

[38] *Real-Time Interface (RTI and RTI-MP) Implementation Guide For Release 4.1*, dSPACE, March 2004.

[39] *TMS320x281x DSP System Control and Interrupts Reference Guide*, Texas Instruments, March 2005, Literature number SPRU078C.

[40] M. Neser, "Object-orientated embedded C," June 2006.

[41] Technical Committee ISO/TC 108, "Mechanical vibration - vibration of rotating machinery equipped with active magenetic bearings - part 3: Evaluation of stability margin," International Organization for Standardization, Committee draft ISO/CD 14839-3, November 2003.

[42] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 9th ed. Upper Saddle River, New Jersey: Prentice Hall, 2001.

Appendix A

Contents of data CD

The data CD included contains the following:

- Matlab® filter design code
- The Base Matlab® AMB model simulation
- VisSim® AMB model simulations
- A VisSim® simulation viewer
- OrCad® simulation of the interfacing circuits
- C code for the embedded controller's firmware implementation in Code Composer Studio®
- User C code and Simulink® model used for the dSPACE® controller
- An electronic copy of this document