Digitally Controlled Class-D Audio Amplifier

by

Carel van der Merwe

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. H. du T. Mouton

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2017 .....

Abstract

Digitally Controlled Class-D Audio Amplifier

CA. van der Merwe

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (Elec)

December 2016

Class-D audio amplifiers have become increasingly popular due to the fact that they use transistors as switches to amplify audio and do not operate them in their linear region, as is the case with other classes of amplifiers. This ensures that class-D amplifiers have very high efficiencies, making them a lot smaller than their counterparts.

Traditionally, class-D amplifiers have been controlled using analogue circuits. This thesis will discuss the digital control of a class-D amplifier. The goal is to implement the amplifier using only a switching output stage, demodulation filter, simple analogue-to-digital converter and an FPGA with peripheral components. This will make it possible for further work to reduce the amplifier to a single integrated circuit and output stage, making it even more compact than its analogue-controlled counterpart while maintaining equivalent performance.

The controller design is done in the z-domain with the PWM modelled as a sampling operation. A mathematical expression is obtained to determine the PWM input signal from which the comparator small-signal gain is calculated. Ripple compensation is implemented to ensure that the comparator small-signal gain remains constant. The main challenge in the controller design is adequately attenuating the quantization noise, which is induced into the system by the digital PWM and the analogue-to-digital converter. This is done by ensuring that the system has a high gain across the audio band (20 Hz to 20 kHz). Simulations are done in an environment emulating that of the FPGA. VHDL is used to practically implement the controller. A system setup is constructed using pre-designed hardware and experimental results are presented.

Uittreksel

Digitaal Beheerde Klas-D Klank Versterker

("Digitally Controlled Class-D Audio Amplifier")

CA. van der Merwe

Departement Elektriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MIng (Elek)

Desember 2016

Klas-D klank versterkers het onlangs baie populêr begin raak te danke aan die feit dat hulle seine versterk deur transistors as skakelaars te gebruik. Ander klasse van versterkers dryf gewoonlik transistors aan in hul lineêre gebied. Klas-D versterkers is dus baie meer effektief as ander versterkers wat veroorsaak dat hulle baie kleiner gemaak kan word as die van ander klasse.

Klas-D klank versterkers word tradisioneel beheer deur analoog bane. Hierdie tesis behels die digitale beheer van 'n klas-D klank versterker. Die doel is om die versterker te implementeer deur net 'n uittreestadium, demodulasie filter, analoog-na-digitaal omsetter en FPGA te gebruik. Dit sal dit moontlik maak om in toekomstige werk die versterker te implementeer deur slegs 'n uittreestadium en 'n enkele geïntegreerde stroombaan te gebruik. Hierdie sal die versterker nog kleiner as sy analoog beheerde eweknie maak, terwyl dit ekwivalente verrigting handhaaf.

Die beheerder ontwerp is in die z-vlak gedoen waar die PWM gemodelleer word as 'n monster operasie. 'n Wiskundige uitdrukking is afgelei om die PWM intreesein te bereken. Hierdie uitdrukking word dan gebruik om die kleinseinaanwins van die vlak-vergelyker te bereken. Riffelkompensasie word geïmplimenteer om te verseker dat die kleinseinaanwins konstant bly. Die hoof uitdaging van die beheerder ontwerp is om die kwantiseringsruis, wat deur die digitale PWM en die analoog-na-digitaal omsetter veroorsaak word, genoeg te onderdruk. Hierdie word bereik deur te verseker dat die sisteem

'n hoë aanwins het oor die hele klank spektrum (20 Hz - 20 kHz). Simulasies word gedoen in 'n omgewing wat die van die FPGA naboots en VHDL word gebruik om die beheerder prakties te implementeer. Die sisteem word gebou uit voorheen ontwikkelde hardware en eksperimentele resultate word getoonset.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations

- God for carrying me through the project.
- Professor Mouton for his intelligent teaching.
- Fellow students in the Power Electronics Group.
- My friends and family for their support.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AC	alternating current
ADC	analogue-to-digital converter
DC	direct current
EMI	electromagnetic interference
FFT	fast Fourier transfer
FPGA	field-programmable gate array
IC	integrated circuit
I/O	input/output
LUT	lookup table
MOSFET	metal-oxide-semiconductor field-effect transistor
NTF	noise-transfer function
PLL	phase-locked loop
PTE	pulse timing error
PWM	pulse-width modulation/modulator
RMS	root mean square
SPI	serial peripheral interface
STF	signal transfer function
THD	total harmonic distortion
VHDL	VHSIC hardware description language

Variables

A_{c_s}	carrier signal amplitude
A_r	reference signal amplitude
f_s	sampling frequency

K_{ss}	small-signal gain
m_a	amplitude modulation index
T_s	sampling time
t_d	time delay

Chapter 1

Introduction

1.1 Background

Class-D audio amplifiers have become increasingly popular due to the fact that they can produce high quality audio while operating at very high efficiencies. These high efficiencies can be achieved due to the fact that in a class-D amplifier, the transistors in the output stage are used as switches and not operated in their linear region as is the case with class-A, class-AB and other, more traditional, amplifier topologies. In an ideal output stage, this results in the transistors never having current flowing through them while having a voltage across them, resulting in theoretical power losses of zero. The high efficiencies of class-D amplifiers give them the advantage of being a lot smaller physically since they do not require the large heat sinks that are necessary with other classes of amplifiers.

Class-D amplifiers are divided into two categories according to the way they obtain their carrier signal. The amplifier is either self-oscillating, where feedback from the output is used to create the amplifier's own carrier signal by operating in a limit cycle [1], or externally clocked, where external circuitry is used to create a carrier signal. One of the biggest drawbacks of self-oscillating class-D amplifiers is, however, that the oscillation frequency varies along with the modulation index [2]. This can potentially cause audible beat tones in a multi-channel audio system.

Traditionally, both self-oscillating and externally clocked amplifiers have been controlled using anologue circuits. The following work will however discuss a digitally controlled class-D amplifier which will consist of an output stage, FPGA and a simple analogue-to-digital converter (ADC). A field-programmable gate array (FPGA) is used in order to utilize VHDL coding which will enable future work to implement the control of the amplifier, including the ADC, into a single integrated circuit (IC). This will enable the amplifier to be even smaller than its analogue controlled counterpart.

1.2 Basic Operation Overview

The basic principles upon which a class-D amplifier operates are very similar to that of a DC to AC converter. A basic half-bridge configuration is shown in Figure 1.1. The two MOSFET switches, S_1 and S_2, are switched complimentary by gate signals generated using pulse-width modulation (PWM). The PWM signal is obtained by comparing a high-frequency carrier signal, typically a triangle or sawtooth waveform, to a low-frequency reference signal. Given a sinusoidal reference signal the output voltage V_o is described by

$$V_o(t) = m_a V_s sin(w_1 t). \tag{1.1}$$

It is clear that the amplitude modulation index m_a and V_s, which is equal to half of the bus voltage, have a significant influence on the gain of the system. The amplitude modulation index is defined by

$$m_a = \frac{A_r}{A_{c_s}}, \tag{1.2}$$

where A_r is the amplitude of the reference signal and A_{c_s} the amplitude of the carrier signal. The frequency modulation index is defined by

$$m_f = \frac{f_r}{f_{c_s}}, \tag{1.3}$$

where f_r is the frequency of the reference signal and f_{c_s} the frequency of the carrier signal (which is equal to the switching frequency).

Using a triangle waveform as the carrier can result in pulse skipping if the system has a very high gain. A sawtooth carrier $c_s(t)$ is thus commonly used in class-D amplifiers, along with an audio signal as the reference signal $r(t)$. The use of an audio signal as

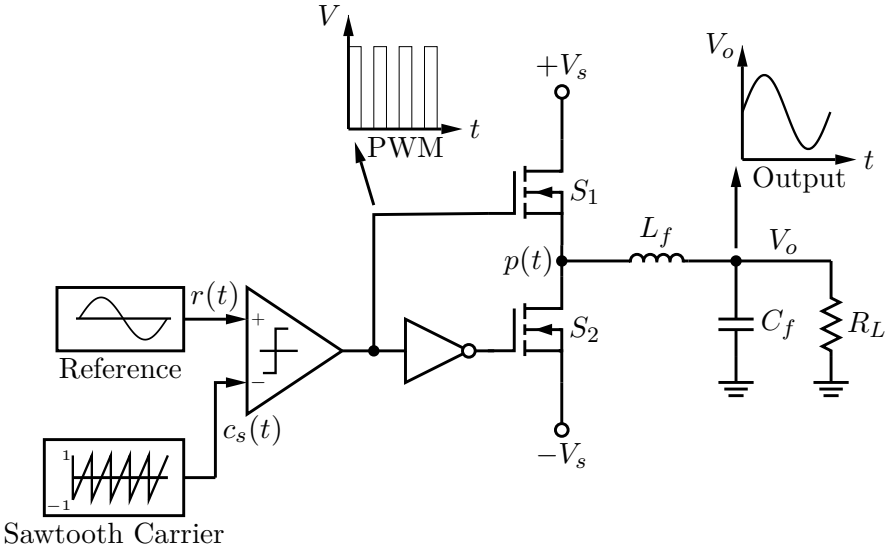


Figure 1.1: Basic Class-D configuration

the reference signal is thus the defining difference between a DC to AC converter and a class-D amplifier as this will result in time-varying amplitude and frequency modulation indexes.

A demodulation filter is used to remove the high-frequency components, which are present at harmonics of the switching frequency and their associated side-bands, from the amplified PWM signal $p(t)$. An amplified version of the reference signal is also contained within $p(t)$ [3]. The demodulation filter is a second order low-pass filter and the high-frequency harmonics exhibit a first order decrease with frequency [4]. The corner frequency of the demodulation filter is therefore chosen to adequately attenuate all frequencies above the audio band. The audio band is from 20 Hz to 20 kHz. The switching frequency needs to be notably higher than 20 kHz to avoid aliasing.

The basic class-D topology does, however, have several drawbacks. The open-loop performance of the system is limited by the non-linearities and imperfections introduced by the output stage and demodulation filter. These aspects include pulse timing and amplitude errors, electromagnetic interference (EMI) and power supply rejection which is essentially zero due to the amplitude of the output signal being modulated by the power supply. The demodulation filter leads to the frequency response of the amplifier being load dependent. The inductor's non-linear characteristics also contribute distortion and increase the amplifier's output impedance [4, 5]. The best way to overcome these drawbacks is to implement a global negative feedback loop. Global feedback is implemented by closing a loop around all the stages of the amplifier as a whole, in effect subtracting the output from the input of the system. Local feedback in turn, is when a feedback loop is only closed around a section of the amplifier. Global feedback does, however, give rise

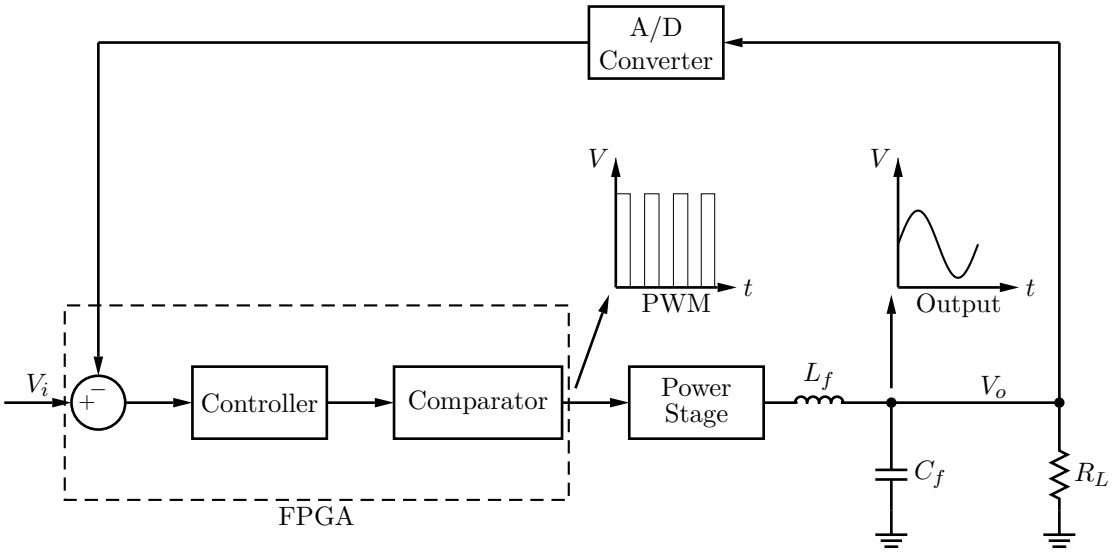


Figure 1.2: Digitally controlled Class-D setup

to high-frequency components, including quantization noise, being aliased into the audio band since the comparator acts as a sampling operation [1].

It is important to note that the common term *"digital amplifier"*, which is often used to describe class-D amplifiers, is deceiving. The power stage as seen in Figure 1.1 is clearly analogue. All class-D amplifiers are inherently analogue but can be classified as either analogue or digitally controlled. Figure 1.2 illustrates the configuration of a digitally controlled class-D audio amplifier. Here it can be seen that the basic elements as seen in Figure 1.1 are all present with the addition of a controller and a global feedback loop containing an ADC. The comparator and controller is implemented using an FPGA and the ADC is used to digitise the feedback signal. The biggest drawback of the digitally controlled configuration is that it induces quantization noise into the system. The choice of ADC is thus very important as its performance can be detrimental to the system if it does not exceed the performance requirements which were initially set for the amplifier.

1.3 Objectives

The main goal of this thesis is to practically implement a digitally controlled Class-D amplifier. This implies that the system has to consist of only an output stage, analogue-to-digital converter (ADC) and an FPGA. The FPGA must be used to implement the controller using VHDL and the ADC to digitise the output signal for feedback purposes. The system must be designed to implement and accommodate the following aspects:

- Ripple compensation.
- An anti-windup scheme.
- Accurate discrete-time domain comparator model.
- Control topology which can be implemented in an FPGA.
- A sigma-delta analogue-to-digital converter to digitise the output signal for global feedback.
- An open-loop controller gain of 50 dB or more across the audio band (20 Hz - 20 kHz).

The focus of the project must, however, remain on the modelling and implementation of the control system and not the output stage. It is desirable for the amplifier to provide state-of-the-art audio performance. For this to be achieved, the following objectives were set for the closed-loop performance of the amplifier:

- A flat frequency response, which does not vary more than 0.15 dB across the audio band.
- A THD+N measurement of less than 0.006 % in the audio band.
- A stable system with both 4.1 Ω and 8.2 Ω loads.

1.4 Thesis Outline

Chapter 2 covers the theoretical background which forms the foundation of the thesis. Distortion mechanisms, comparator models, ripple compensation and controller topologies are discussed. Anti-windup schemes are also investigated.

Chapter 3 discusses the design and modelling of the system. The stability of the system is determined using a root-locus plot and it is determined whether the designed system meets the fundamental operation requirements. An equation for the PWM input gradient is derived and the system's noise-transfer function is analysed.

Chapter 4 presents the simulation setup which is used to verify the operation of the controller. The simulation is done using Simulink. The results are analysed and confirms that the controller's performance meets the desired requirements.

Chapter 5 discusses the practical implementation of the controller using an FPGA. The hardware setup of the total system is discussed along with the operating frequencies of the different components. The system flow is also discussed.

Chapter 6 presents the practical measurements of the amplifier. The practical results are compared to the simulated results and proof of concept is confirmed as the system functions correctly. Practical results, however, are not of the same standard as the simulated results and the amplifier does not provide the required state-of-the-art performance.

Chapter 7 discusses the conclusions which were made following the work done in the thesis. Recommendations regarding further work are made.

Chapter 2

Theoretical Background

2.1 Distortion Mechanisms

It is shown in [3] that an ideal class-D output stage only has one source of distortion in the audio band which is the side-bands of the carrier signal. This is not the case in a practical circuit where the output stage is not ideal. Along with the several distortion mechanisms of the basic class-D configuration already mentioned in Section 1.2, the rest of the mechanisms are discussed in depth in [3] and [5]. Of these, pulse-timing errors are responsible for the most distortion of the output signal. Pulse-timing errors are caused by dead time [3]. Most of these distortion mechanisms are mitigated by implementing a global negative feedback loop, however, dead time still has a big influence on the distortion of the output signal along with the aliased high-frequency components and quantization noise induced by the global negative feedback loop.

2.1.1 Dead Time

The basic operation of a class-D amplifier, which was discussed in Section 1.2, illustrates that the system operates in a similar way to a DC to AC converter and that the two MOSFET switches are switched complimentary. In a practical circuit it is, however, not possible for a MOSFET to switch on or off instantaneously and there will be a short time during which it will operate in its linear region. This entails that the switch will have a voltage across it, along with current running through it. This could potentially cause that both the switches conduct current simultaneously. This will in turn short circuit the DC bus capacitors and a large amount of shoot-through current will flow through the device, potentially damaging the device. In order for the system to prevent this from happening, the modulator waits for a time period t_{dt} after turning a switch off before turning on the other switch. The time period when both switches are off, is called dead

time. Figure 2.2 illustrates the waveforms of a basic power stage with output filter. The

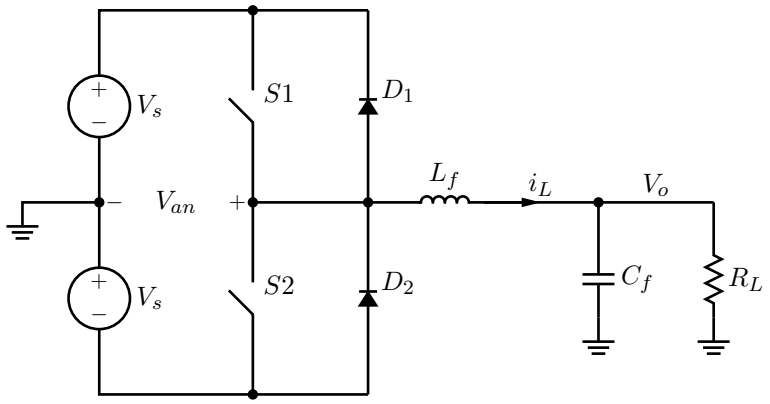


Figure 2.1: Power switching stage with output filter.

gate signals of S_1 and S_2 are shown along with the unfiltered output voltage V_{an}. The ideal waveforms are visible in grey while the waveforms when dead time is implemented are visible in black. It is important to note that the dead time was exaggerated in this example to illustrate the concept and is much shorter in a practical circuit. The first transition which is observed is when $S1$ switches on and S_2 switches off. In the dead time before S_1 switches on, diode D_2 will conduct current if $i_L > 0$ and the output voltage V_{an} is equal to V_s. It is important to notice that the ideal voltage is equal to $-V_s$ in this case. When $i_L < 0$, diode D_1 will conduct during the dead time and the $V_{an} = V_s$ which is the

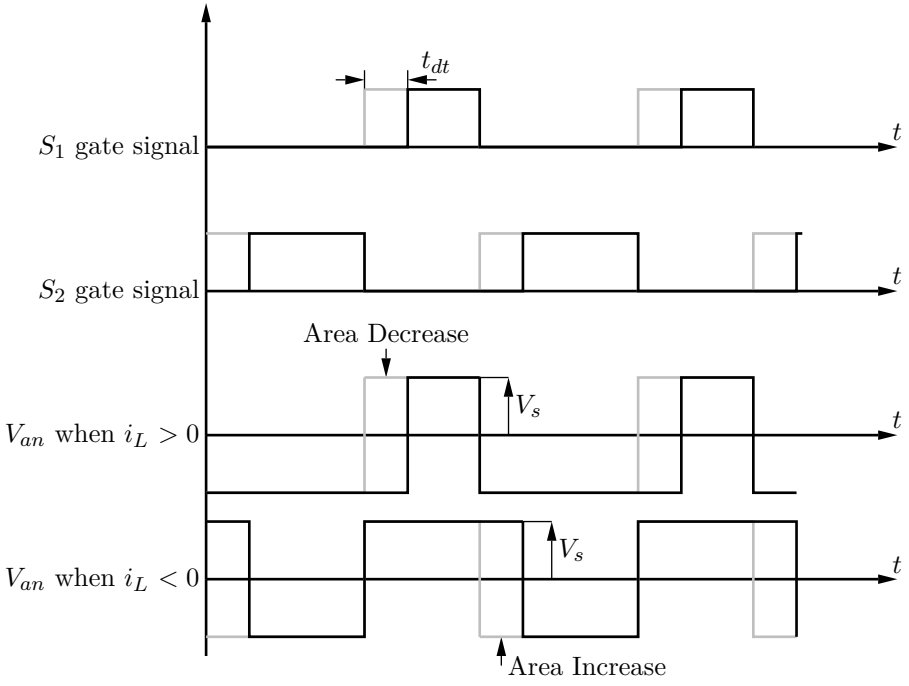


Figure 2.2: Converter waveforms with dead time adapted from [5].

same as the ideal output voltage. The area of the output voltage pulse is thus decreased in case of $i_l < 0$. The second transition is the opposite of the first, meaning that S_1 now switches off with S_2 switching on. Here it can be seen that the area of the output voltage V_{an} will increase in case of $i_L < 0$. The effect of dead time on the output voltage of the converter is thus clearly illustrated.

When the amplitude modulation index m_a of the system is small, dead time only causes a time delay of t_{dt} in the system. This happens due to the double polarity change that occurs every switching cycle in the inductor current. This is, however, not the case with large values of m_a where the polarity of the inductor current is constant for several switching cycles and is essentially determined by that of the reference signal. The unfiltered voltage error is thus dependent on the polarity of the reference signal. Along with the odd harmonic distortion in the audio band caused by dead time, it was also shown in [3] that the magnitude of the side-band switching harmonics increases in the audio band for large values of the amplitude modulation index m_a. It is important to note that pulse-timing errors increase when the switching frequency of the system is increased. The distortion of the output signal also increases with longer dead times.

2.1.2 Quantization Noise

When analysing an analogue signal, it is possible to attain an exact amplitude value for every instance in time. Digital signals are obtained from their analogue counterparts by sampling a series of values at certain instances in time. During this process, sampling represents the time and quantization the value of the measurement. The sampling time determines the bandwidth and quantization the resolution of the signal characterization. For a digital signal to exactly replicate it's analogue counterpart, an infinite amount of samples will have to be taken and every sample will need an infinite word length. This is not practically possible. A measuring error is thus introduced since the system's resolution is limited by the quantization. This induces quantization noise into the system [6].

Figure 2.3 illustrates the basic quantization process. It can be seen that the infinite amount of amplitude values of the analogue signal is converted to digital values by rounding up or down to the closest available digital value and retaining that value for the duration of the sampling period. The quantization error is the difference between the actual analogue value and the rounded digital value. This process is known as uniform quantization as the amplitudes of the analogue signal is mapped into words with equal bit sizes. Increasing the word length of the quantization levels will improve the quality of the digital signal and decrease the quantization error. This is due to the fact that a longer word length provides more amplitude levels, in turn increasing the resolution of the signal characterization. Increasing the sampling frequency will also reduce the quan-

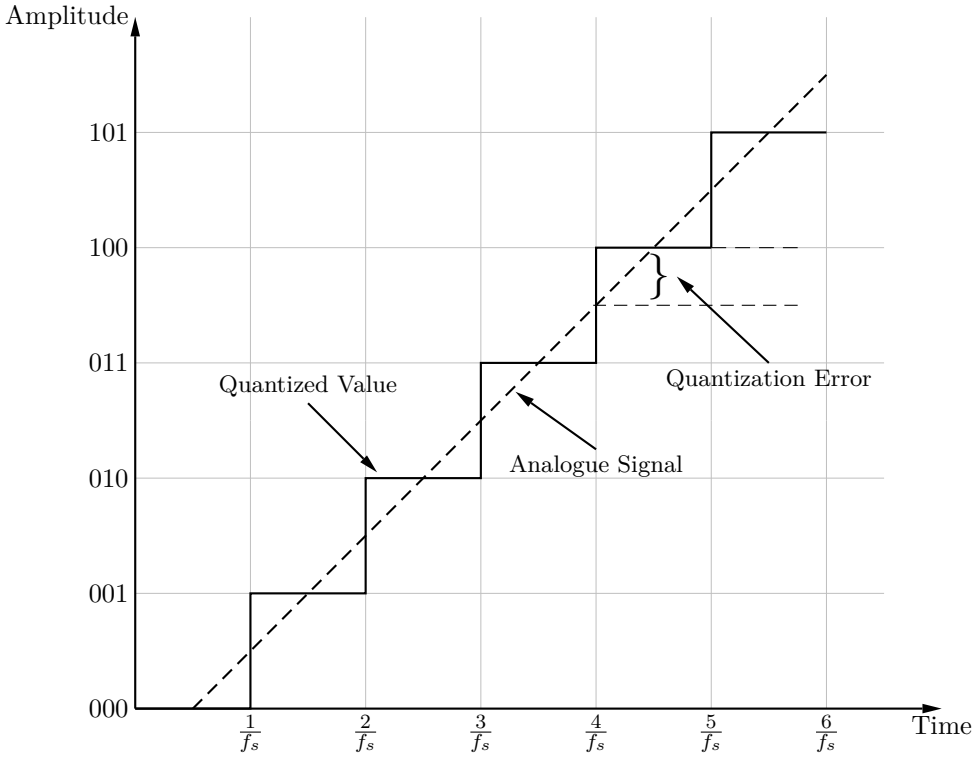


Figure 2.3: Quantization Example.

tization error as this will result in the digital value being updated more frequently and thus also increase the bandwidth.

2.2 Modelling of Pulse-Width-Modulator Loops

The pulse-width modulator is an integral part of the system. Traditionally, the pulse-width modulator's only non-linear element, which is the comparator, is modelled as an equivalent gain [7]. In [1] and [8] this traditional model is discussed at length and it is shown that modelling the PWM as an equivalent gain results in inaccurate stability margins as the high-frequency behaviour of the comparator is not accounted for. An accurate model of the PWM, which accounts for the non-linear effects such as aliasing and the emergence of image components, is also presented. The following section will discuss the model in detail and the integration thereof into closed-loop systems.

2.2.1 Comparator Small-Signal Models

The small-signal model describes the response deviation of the system when a small-amplitude perturbation signal is added to the input. This is done with the system linearised around a steady-state operating point. When a double-edged modulator is used,

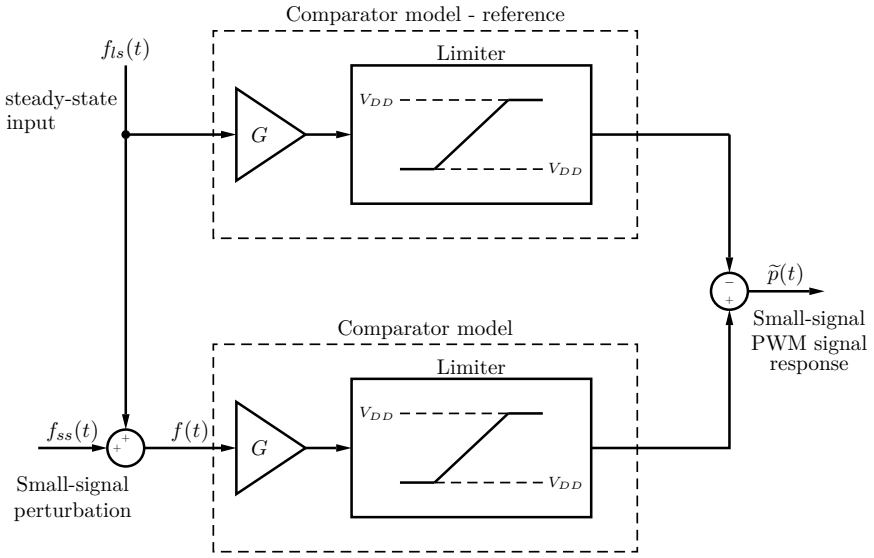


Figure 2.4: Small-signal comparator model [1].

steady-state operation can only be achieved when the duty cycle is equal to 50 %. This is achieved by given the system an input of zero and is one of the major disadvantages of a double-edged modulator. The small-signal model of a single-sided modulator is, however, linear and time invariant as long as the duty cycle remains constant. In [1] a conceptional small-signal model of an ideal comparator is presented for a generalised carrier. This model can be seen in Figure 2.4. The comparator is modelled by the gain G, which is followed by the power stage - implemented by saturation to the supply voltage V_{DD}. By letting G tend towards infinity, an ideal comparator model is obtained. The model consists of two ideal comparators of which one receives the large steady-state signal $f_{ls}(t)$ as input, referred to as the reference system. The other comparator also receives $f_{ls}(t)$, but a small-amplitude perturbation signal $f_{ss}(t)$ is superimposed and is referred to as the perturbed system. The small-signal response is obtained by subtracting the output of the reference system from the output of the perturbed system.

Figure 2.5 illustrates the effect the small-signal component will have on a single-edge naturally-sampled pulse-width modulator [1]. It is assumed that the PWM has control logic which ensures that the system only responds to the first crossing between the input signal and the carrier signal. The amplitude of the carrier is also assumed to be equal to 1. It is clear that the large steady-state input signal $f_{ls}(t)$ and $f(t)$, which is obtained by adding the small perturbation signal $f_{ss}(t)$ to $f_{ls}(t)$, intersects the sawtooth carrier at different points in time. This results in the PWM output being amended by a narrow rectangular pulse with a duration of $\triangle T$ and an amplitude of 2. The small-signal PWM response $\widetilde{p}(t)$ can be described as a train of impulses, as a narrow rectangular pulse will be present in every switching cycle. The small-signal PWM response is thus equivalent

to a Dirac comb with a frequency of f_s. The comparator can effectively be modelled as a sampling operation since, when working in the time-domain, multiplication with a Dirac comb is identical to sampling.

To calculate the small-signal gain, it is necessary to determine the area of these pulses. Figure 2.6 shows a zoomed view of the section surrounded by the ellipse in Figure 2.5. The gradient of the carrier is $\frac{2}{T_s}$. Basic geometry is used to derive

$$\frac{2}{T_s} \triangle T = f_{ls}(t_{sp} + \triangle T) + f_{ss}(t_{sp} + \triangle T) - f_{ls}(t_{sp}). \tag{2.1}$$

The pulse duration can then be determined by

$$\triangle T \approx \frac{f_{ss}(t_{sp})}{\frac{2}{T_s} - \dot{f}(t_{sp})}. \tag{2.2}$$

The distance between the steady-state signal $f_{ls}(t)$ and $f(t)$ at the sampling point t_{sp} is denoted by $f_{ss}(t_{sp})$. T_s is the switching period and $\dot{f}(t_{sp})$ is the gradient of $f(t)$ at t_{sp}. The area of the rectangular pulse in Figure 2.5 is given by

$$A = 2 \triangle T. \tag{2.3}$$

The strength of each impulse can be determined by

$$\text{Strength of impulse} = \left[\frac{2}{2f_s - \dot{f}(t_{sp})}\right] f_{ss}(t_{sp}). \tag{2.4}$$

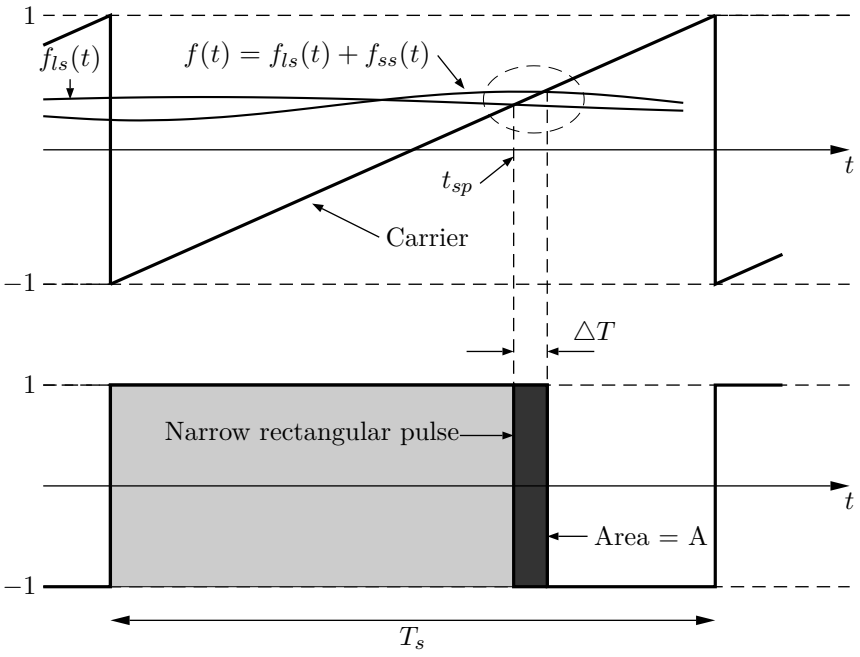


Figure 2.5: Small-signal carrier crossing.

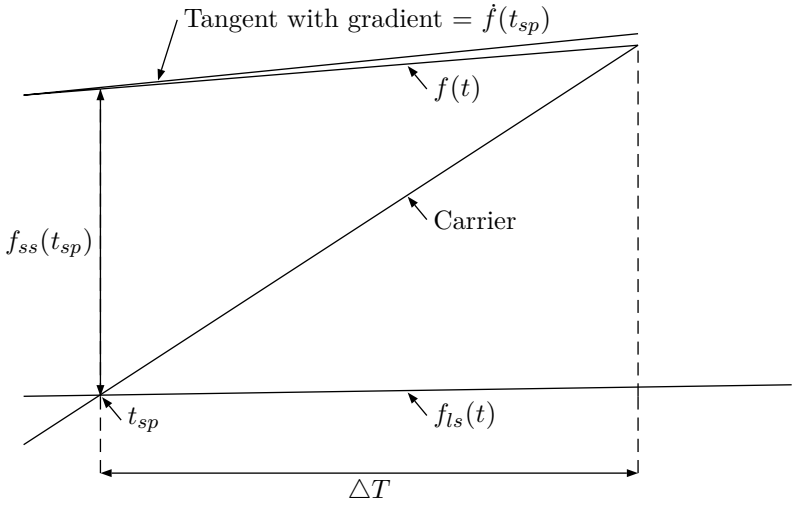


Figure 2.6: Small-signal transition point analysis.

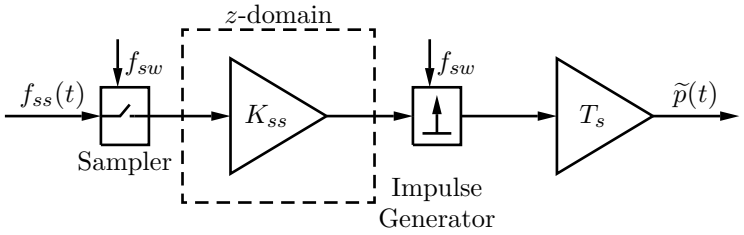


Figure 2.7: PWM Small-signal model.

At this point it is important to note that the strength of each impulse is dependent on the gradient of $f(t)$ and proportional to f_{ss} at the sampling point t_{sp}. Equation 2.4 can then be written as

$$A \approx K_{ss} T_s f_{ss}(t_{sp}), \tag{2.5}$$

where the small-signal gain K_{ss} is given by

$$K_{ss} = \left[\frac{2f_s}{2f_s - \dot{f}(t_{sp})} \right]. \tag{2.6}$$

A block diagram of the PWM small-signal model can be seen in Figure 2.7. It is modelled as a sampling operation with the sampling frequency equal to the switching frequency of the system. This is followed by the equivalent small-signal gain K_{ss} and an impulse generator. A sample is taken where the carrier signal and the input signal dissect each other. The sample is then multiplied by K_{ss} after which the impulse generator will generate a pulse with a strength of $f_{ss}K_{ss}$. Thereafter the signal is multiplied by the switching period T_s. The previously mentioned controller logic ensures that this process only happens once per switching period.

When integrated into a basic system, as seen in Figure 2.8a, the PWM output signal will be passed through a low-pass filter. A feedback loop will be implemented using

the output of the low-pass filter as the source of the feedback. The signal then also passes through a compensator before reaching the comparator input. This means that the signal will have a periodic ripple component when reaching the comparator input. In [1] it is observed that the comparator gain is effectively reduced, compared to open-loop operation, by the ripple feedback component. This implies that the ripple component, and thus K_{ss}, is dependent on the duty cycle of the system. In a practical system this makes it impossible to accurately compensate for K_{ss}, as it is constantly changing. Section 2.4 discusses ripple compensation techniques that makes the ripple gradient independent of the duty cycle, causing K_{ss} to be more constant and making it possible to accurately compensate for K_{ss}.

2.2.2 Integrating the Small-Signal Model into a Closed-Loop System

A basic closed-loop system can be seen in Figure 2.8a. Global negative feedback is implemented along with a basic compensator $G_c(s)$. The time delay function accounts for any practical time delay the system might have while the power stage is simply modelled as the gain V_d. An output filter $F(s)$ is also included. The traditional PWM model, where the comparator is simply modelled by a unity gain, is used and the open-loop transfer function is depicted by

$$G_{OL} = G_c(s)V_d e^{-st_d}F(s). \tag{2.7}$$

Figure 2.8b shows a system where the PWM small-signal model discussed earlier in this section is implemented. A small-signal compensation term, of which the value is the inverse of K_{ss}, is added to negate the effect of K_{ss}. It can be seen that the area between the sampling operation and the impulse generator is in the discrete-domain. Using block diagram manipulation Figure 2.8b is used to obtain Figure 2.8c. The stability of the system is now solely dependent on $G_c(z)$, which is obtained from [1] and given by

$$G_c(z) = \mathcal{Z}\{G_c(s)F(s)K_cV_d e^{-st_d}\}. \tag{2.8}$$

The $\mathcal{Z}$ symbol refers to the z-transform using the impulse invariance method discussed in section 2.3. Given that the loop stability analysis can now be done in the discrete-time domain, the accuracy of the loop stability analysis will improve in comparison with the traditional average model. This will also make it possible to model the frequency response of the comparator accurately up to the switching frequency [5].

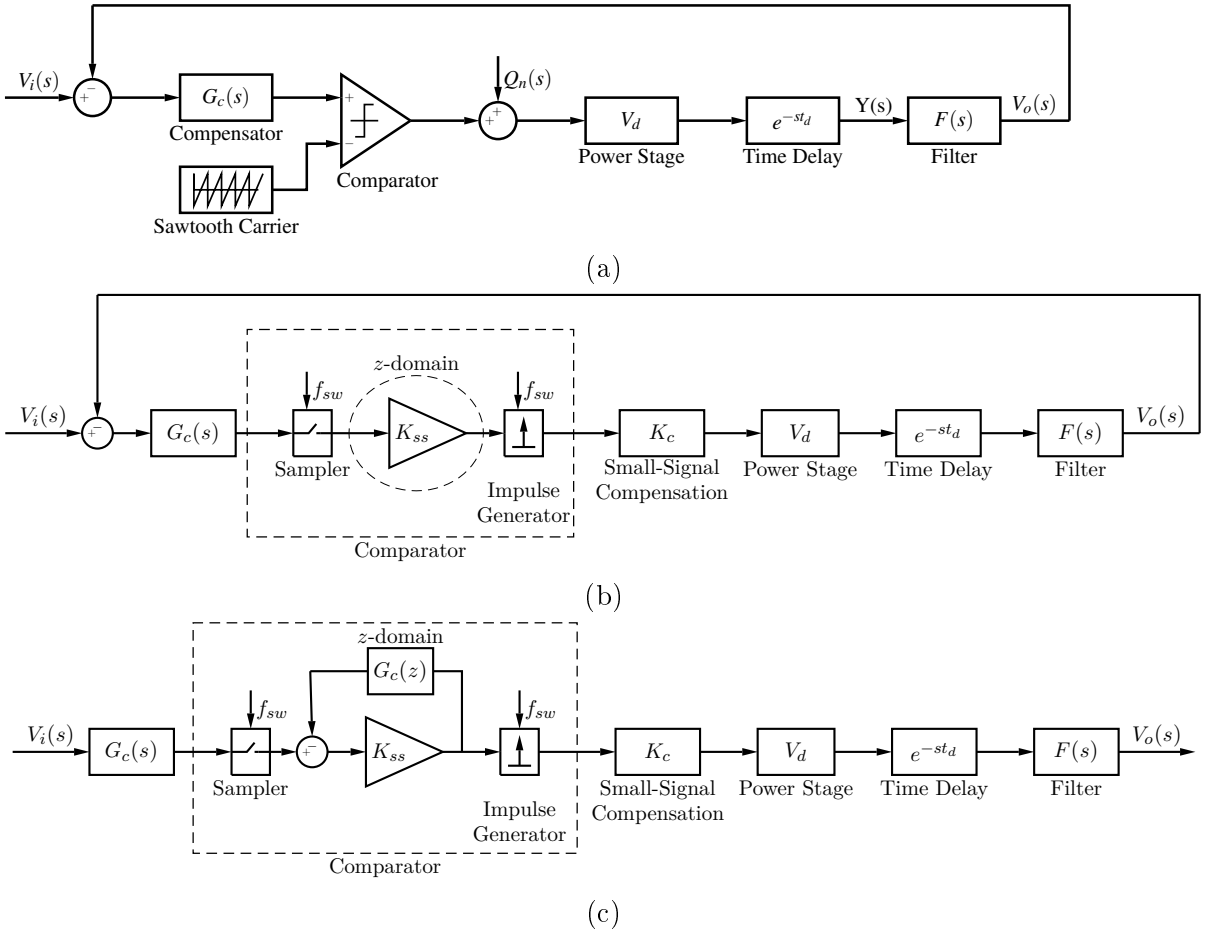


Figure 2.8: System models with traditional and z-domain comparator models.

2.3 Impulse Invariance Method

There are several methods which can be used to convert a continuous-time transfer function $G(s)$ to the discrete-time domain transfer function $G(z)$. The small-signal model which is discussed in Section 2.2 demonstrates that both the input and the output of the loop filter are sampled signals due to the sampling nature of the comparator. The impulse invariance method is therefore appropriate to convert $G(s)$ to $G(z)$ since the behaviour of the comparator is very similar to that of the impulse invariance discretisation method. This method entails determining the z-transform of the sampled impulse-response of the system [5]. This method will, however, result in a non-zero value for $g_i(k = 0)$, which is the first sample of the impulse response. Due to the fact that a practical system is causal, the system will always have a propagation delay of at least one sample [1]. A possible solution to this problem is to simply remove the response at time zero after performing the z-domain transform as can be seen in

$$G(z) = \widehat{G}(z) - \widehat{g}_i(k = 0), \tag{2.9}$$

where $\widehat{G}(z)$ is the z-transform of $G(s)$ using the standard impulse invariance method.

An alternative method, which utilises the standard impulse invariance method and accounts for the propagation delay of the system, is proposed in [5]. Figure 2.9 illustrates the basic principle of the proposed method. The time delay of the system is denoted by t_d.

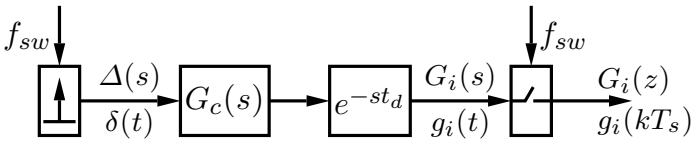


Figure 2.9: Simplified block diagram illustrating the impulse invariance method.

It is assumed that the form of the transfer function $G(s)$, which is obtained by expanding $G_c(s)$ into partial fractions and adding the time delay of the system, is equivalent to

$$G(s) = e^{-st_d}\frac{1}{s+p}. \tag{2.10}$$

The continuous-time domain impulse response is obtained from [9] as

$$g_i(t) = e^{-p(t-t_d)}\mu(t-t_d), \tag{2.11}$$

where the step-function $\mu(\tau)$ is defined as

$$\mu(\tau) = \begin{cases} 0 & \text{if} \quad \tau < 0 \\ 1 & \text{if} \quad \tau \geq 0 \end{cases}. \tag{2.12}$$

The continuous-time impulse response $g_i(t)$ can then be discretised by letting $t = kT_s$ where T_s is the sampling time and k the number of the sample. The discrete-time impulse response is given by

$$g_i(kT_s) = e^{-p(kT_s-t_d)}\mu(kT_s-t_d), \quad k = 0, 1, 2, ..., N, \tag{2.13}$$

where N is the number of samples taken.

It is clear that the first sample of the discrete-time impulse response will be zero. It is assumed that t_d is shorter than T_s and will therefore only influence the first sample. The z-transform of $g_i(kT_s)$ can be calculated and manipulated, as in [5], to obtain the s-domain to z-domain transformation

$$G_i(z) = e^{pt_d}\frac{e^{-pT_s}}{z-e^{pT_s}}. \tag{2.14}$$

The z-transform is essentially a summation of the individual terms and the total z-transform is therefore given by

$$G(z) = T_s\sum_{n=1}^{N} A_n e^{p_n t_d}\frac{e^{-p_n T_s}}{z-e^{-p_n T_s}}. \tag{2.15}$$

The transform is multiplied by T_s in order to ensure that the transfer function has equivalent DC gains in the two different domains. A_n is a constant coefficient obtained through partial fraction expansion of the s-domain transfer function. It can also be seen that the propagation delay of the system will only affect the location of the z-domain zeros.

According to [10] any proper transfer function, a transfer function where the order of the numerator is lower than that of the denominator, with distinct poles can be expanded into partial fractions. Equation 2.15 can thus be used to determine the z-transform of any transfer function which adheres to the above mentioned criteria and is expanded into the form of

$$G(s) = \sum_{n=1}^{N} \frac{A_n}{s + p_n}. \tag{2.16}$$

2.4 Ripple Compensation

As discussed in Section 1.2, implementing a global feedback loop is the best way to overcome several drawbacks which are present in an open-loop configuration. This, however, introduces new problems to the system. The ripple signal which is present on the output of the amplifier causes low-frequency distortion of the output signal even if the output stage is ideal. Due to the sampling nature of the comparator, the high-frequency components, which are present at the harmonics of the switching frequency and its side-bands, will be aliased into the audio band and will manifest as harmonic distortion should the aliased components be harmonically related to the input signal [1, 5, 11, 12].

In [11] two distortion mechanisms were identified which are caused by the feedback ripple aliasing. Firstly, a DC non linearity is caused due to the distortion of the pulse width. The second distortion mechanisms is due to phase modulation, which is in essence a non-linear time shift of the PWM pulses [4, 11]. The feedback ripple component also causes a change in the small-signal PWM gain, which can result in instability of the system as discussed in Section 2.2. Various methods of negating these drawbacks have been proposed. A minimum aliasing error filter, which is implemented in a class, is

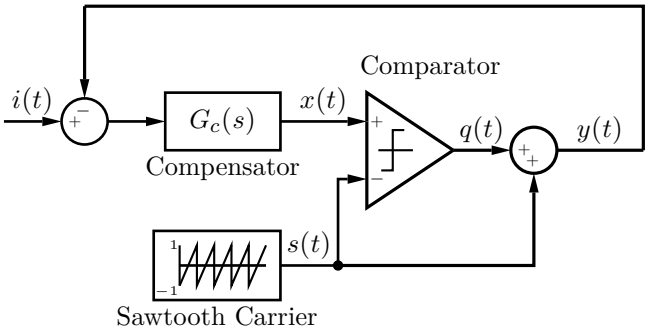


Figure 2.10: Simplified PWM feedback loop with ripple compensation [4].

presented in [11] and in [12] a method is proposed where the carrier signal is modulated to modify the symmetry thereof. The simplest, and most effective, way of solving the ripple feedback problem is presented in [13]. A basic implementation of this scheme can be seen in Figure 2.10. The method proposes cancelling the unmodulated edges of the PWM signal. This is done by adding the sawtooth carrier $s(t)$ to the PWM output signal $q(t)$ as can be seen in Figure 2.10.

The method was implemented in [4] and the digitally controlled amplifier produced immaculate results. The method is thus very suitable for this project and the basic operation thereof will be discussed further. Adding the sawtooth carrier $s(t)$ to the the PWM output signal $q(t)$ will result in the output signal $y(t)$ having a shape similar to that of the sawtooth carrier $s(t)$, as can be seen in Figure 2.11. The time average of $y(t)$ is equal to that of the modulator output $p(t)$ over one switching period. The unmodulated edges of $q(t)$ are effectively replaced by linear slopes thereby making the comparator input $x(t)$ mostly independent of the duty cycle of $p(t)$. The result being a DC offset to be the only remaining effect of the ripple feedback [4]. This minimises the frequency components which are aliased into the audio band to only the components associated with then sawtooth carrier signal [5].

The loop filter $G(s)$ will typically have a high gain throughout the audio band and a gain smaller than unity at the switching frequency. This will improve error rejection in the audio band and attenuate the amplitude of the switching frequency components. A typical loop filter waveform $x(t)$ can be seen in Figure 2.11. Due to the feedback loop, which ensures that the control loop accurately tracks the input signal $x(t)$, the crossings of $x(t)$ and $s(t)$ will correspond with those of $i(t)$ and $s(t)$. The ripple component of

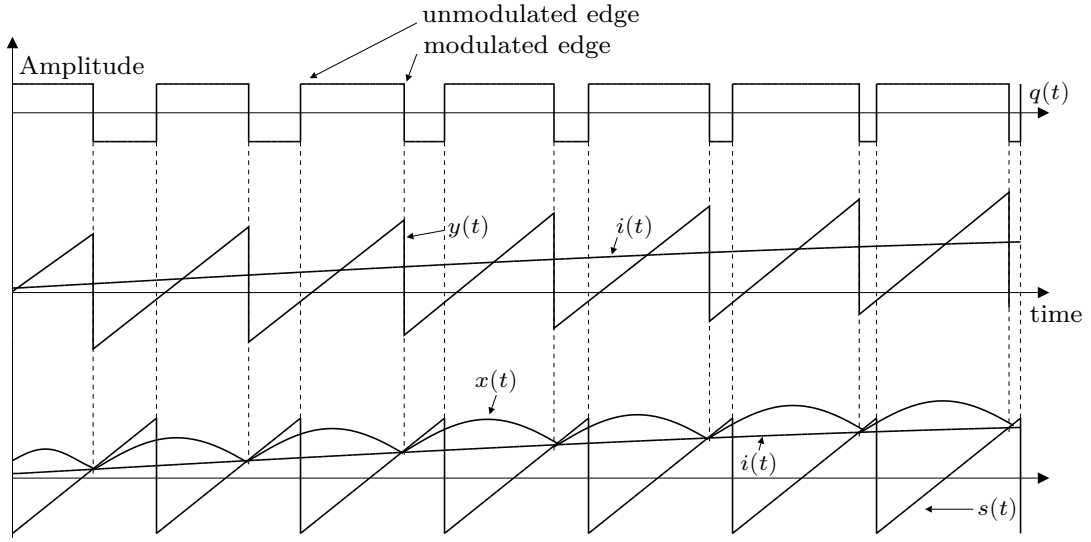


Figure 2.11: Ripple Compensation waveforms.

$x(t)$ is predominantly independent of the mean value of $x(t)$ with the effect of minimising the non-linearities associated with ripple feedback. Ripple compensation also ensures the equivalent comparator gain K_{ss} is largely independent of duty cycle since K_{ss} is a function of the gradient of the ripple feedback signal as discussed in Section 2.2.

The next step is to investigate ways to implement the chosen ripple compensation technique in a switching amplifier that includes a demodulation filter. Figure 2.12 illustrates three different implementations. For the time being, the power stage is assumed to be ideal and modelled by a gain which is represented by A. If a half-bridge output stage

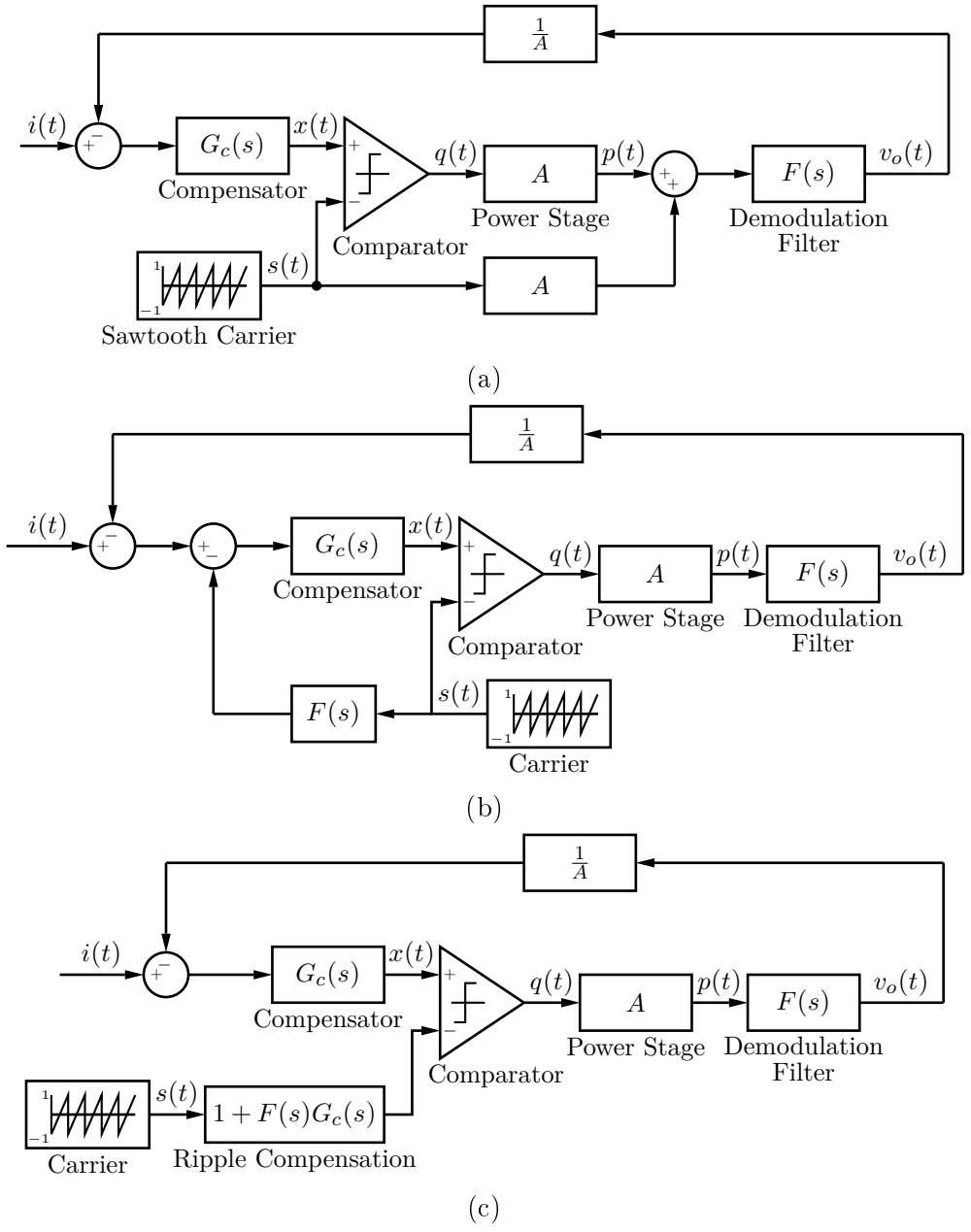


Figure 2.12: Different implementations of the ripple compensation technique [4].

is used, A will be equal to half of the bus voltage while if a full-bridge configuration is used, A will be equal to the bus voltage. It is clear that the configuration in Figure 2.12a is not practical as it will require an amplified version of the carrier signal to be added to $p(t)$ before the demodulation filter. Figures 2.12b and 2.12c are obtained through block diagram manipulation. In Figure 2.12b the sawtooth carrier is passed through a low-pass filter which is equivalent to the output demodulation filter. The signal is then subtracted from the compensator input. It can be seen in Figure 2.12c that the ripple compensation technique is essentially equal to pre-distorting the sawtooth carrier signal. This makes this particular technique even more attractive for digitally controlled amplifiers since the pre-distortion of the carrier can be done off-line and stored in a lookup table. At frequencies higher than the switching frequency, the frequency response of the demodulation filter is largely independent of the output load resistance . This will result in the ripple compensation technique having a low sensitivity with regards to the exact matching of the mathematical function $F(s)$ to the practical demodulation filter.

2.5 Controller Topologies

A controller, also known as a loop filter, is necessary to increase the loop gain in a certain frequency range. This is required to ensure that the error rejection of the system is adequate in the specified frequency range, thus enabling the system to provide high quality output. When implementing a controller in digital form, it is important to consider the device and system constraints. The two constraints that often influence the digital implementation of a control system are memory and timing requirements. An FPGA for instance, only has a limited amount of multipliers which can be used to implement the control system. It is also important that the timing requirements of the system are met, meaning the computation of the controller's output signal must be done within the timing constraints of the system. In [14] various block diagram configurations, known as *filter structures*, are discussed to implement transfer functions in the form of

$$H(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{1 + b_1 z^{-1} + \cdots + b_n z^{-n}}. \tag{2.17}$$

The real coefficients of the numerator and denominator are described by a_i and b_i respectively. The maximum order of the polynomials is given by n. Theoretically, an infinite amount of possibilities exist in which this standard transfer function could be implemented. The main structures including direct-form structure, second order modules, cascaded modules, paralleled modules and laddered structures are discussed in [14]. The structures are compared according to the amount of time-delay elements, multipliers,

summing junctions and signal distribution points they require to implement the transfer function. A structure is chosen to meet the specific constraints of the particular system.

Higher order loops which require a high gain across a certain bandwidth are, however, usually implemented using a chain of integrators [4, 5, 16]. This configuration in it's simplest form is a sole integrator. It is necessary to expand the loop filter since a single integrator has limited gain at the higher end of the audio band, which is the frequency band of interest. The audio band is from 20 Hz to 20 kHz. Pairs of integrators are thus added to improve the gain across the audio band. The chain of integrators is then modified to move the open-loop poles along the imaginary axis, away from DC. This is done by closing a negative feedback loop around a pair of integrators to form resonator loops [15]. The gain γ, embedded into the feedback loop, effectively determines the position of the open-loop poles. The value of γ can be determined using

$$\gamma = (2\pi f_p)^2, \tag{2.18}$$

where f_p is the frequency of the complex pole pair. The transfer function of two integrators with feedback is given by

$$\frac{Q(s)}{R(s)} = \frac{1}{s^2 + \gamma}. \tag{2.19}$$

Figure 2.13 illustrates a configuration with one resonator loop and the initial integrator, forming a third-order controller topology. It is important to notice that the open-loop poles become the noise-transfer function zeros when the loop is closed. Feedforward summation can also be observed, with the output of every integrator being multiplied by a gain k_i before being added together to form the output of the controller. The k_i coefficients can be determined by using the transfer function

$$\frac{Y(s)}{X(s)} = \frac{k_1 s^2 + k_2 s + k_3 + k_1 \gamma}{s(s^2 + \gamma)}, \tag{2.20}$$

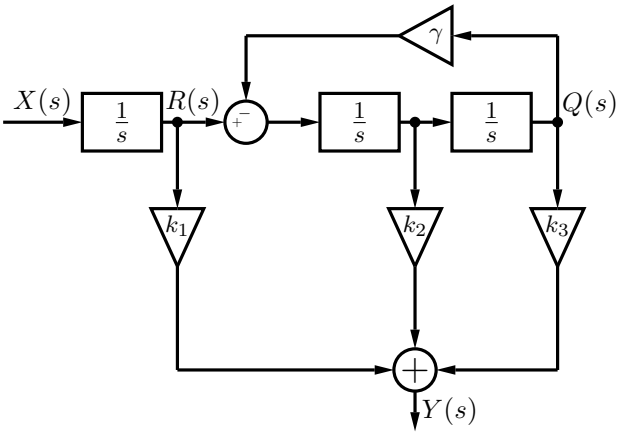


Figure 2.13: Chain of integrators with feedforward summation and local resonator feedbacks adapted from [15].

and equating it to the desired controller transfer function.

The major limiting factor of this configuration is, however, that the complex pole pairs always stay on the imaginary axis. This configuration can be translated to the z-domain by replacing each integrator with $1/(z-1)$. This term is the traditional z-transform of an integrator with an added time delay of z^{-1}. The time delay which is present in the integrators causes the poles to move away from the real axis on a vertical line which starts at $(1, 0j)$ when considering the root locus of the resonator. This in turn prohibits the resonator from having infinite gain at the resonance frequency. A high controller gain can, however, still be achieved as long as the shift of the poles are kept to a minimum. The necessary stability margins and the order of the loop filter are the two factors which limit the maximum obtainable loop gain. The configuration can be expanded by adding more resonator loops to the system seen in Figure 2.13, which was adapted from [15], to counteract these limiting factors.

2.6 Control System Stability

When designing high-gain control loops, it is important to consider the limiting factors. Firstly, high-gain control loops are inclined to be conditionally stable. The amplitude and rate of change of the controller are also restricted when implemented in a practical system. These limiting factors can severely influence the performance of the system and even make the it unstable in certain cases.

Windup is a result of the pulse-width modulator input reaching a saturation limit and the system integrator continuing to integrate even though the input is restricted [17]. This leads to over-modulation and causes the integrator values to increase to intolerably large values, thus saturating. When the system is over modulated, the gain collapses causing the conditionally stable loop to become unstable [5]. Windup also degrades the transient response of the system. When implementing a digital controller, even though the range of the input signal is known and can therefore be limited to a known value, wind-up is still a problem as the behaviour of the controller when returning from an over-modulated condition must still be controlled.

According to [17], protection against wind-up is achieved by ensuring that the states of the controller have two properties. The actual controller input should be used to drive the controller and must have a stable actualization when this is the case. An anti-windup scheme is also presented and can be seen in Figure 2.14. The loop filter is rearranged into a direct feedthrough term and a strictly proper transfer function which is then implemented in feedback form. Limiting circuits are also included to compensate for wind-up. These circuits can be saturation or slew rate limiters, or both depending on

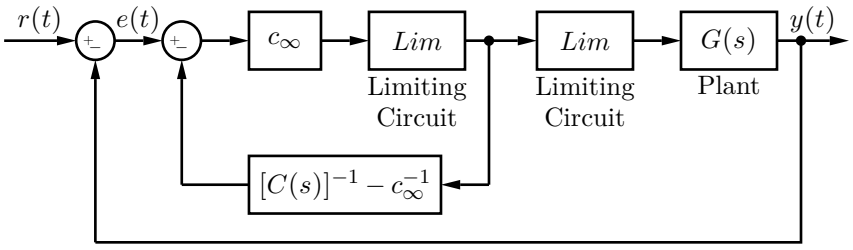


Figure 2.14: Control loop with anti wind-up implemented.

the system constraints. It is shown that this method is particularly easy to implement if the controller is biproper and minimum phase.

Another method was, however, proposed in [18] and discussed in [5] where the control loop is modified to become unconditionally stable when over modulated. This is achieved by reducing the loop order when the loop is over modulated. This method is very applicable to controllers which are implemented using a chain of integrators. This is due to the fact that when an integrator saturates, the order of the system is essentially reduced. The method can thus be implemented by designing the controller to be unconditionally stable when over modulated and by limiting the values of the integrators to return the system to a stable operating condition when coming out of over-modulation. When this method is used, the integrators do, however, tend to saturate during normal operation of the system as well. This reduces system performance and is caused by the fact that the output of an integrator in a stable system under maximum modulation could potentially be larger than the output of an integrator in an unstable system with no input. This means the method will only be successful if the output of the integrators in a stable system are limited to well below the output level in an unstable system [18].

It is, however, possible to stabilize a conditionally stable system without reducing the performance by saturating the integrator when not in over modulation. This is done by ensuring that the controller output is independent of the input signal. This method was initially proposed in [18] for a self-oscillating class-D amplifier, but was adapted in [5] for externally clocked amplifiers.

Figure 2.15a illustrates a basic system which includes a controller, output stage (which is inverting) and a feedback loop [5]. An error term $E(s)$ is also added to account for the output stage errors. The controller output is given by

$$R(s) = \frac{G_c(s)}{1 - G_c(s)F(s)}[V_i(s) + E(s)]. \tag{2.21}$$

It is clear that both the error term $E(s)$ and the input signal $V_i(s)$ have an influence on the output signal. It is also visible that the input signal $V_i(s)$ will have a dominating effect on the controller output for high modulation indexes.

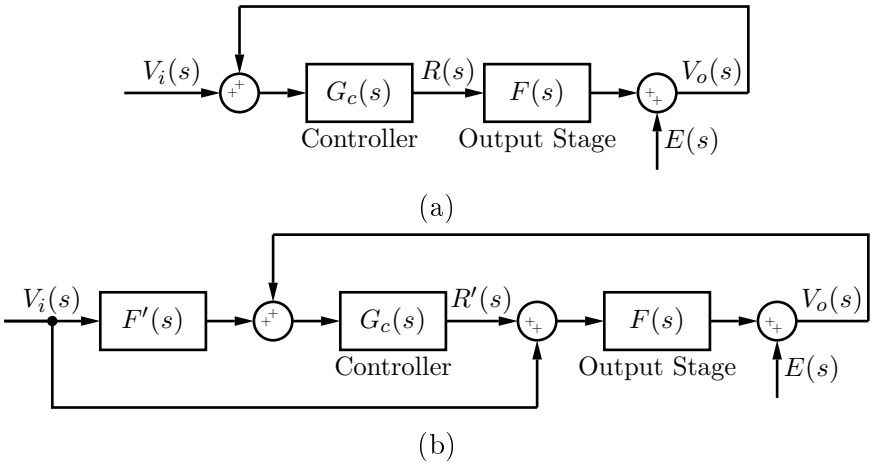


Figure 2.15: Generic system with output stage and system with deviation detection filter adapted from [5].

A deviation detection filter can be implemented to keep the controller output values very small during normal operation which will keep the integrators from saturating. Figure 2.15b illustrates the implementation of the filter. The filter $F'(s)$ approximates the output stage $F(s)$ resulting in the only difference in their outputs during stable operation being the minor error term $E(s)$. Therefore making the controller output virtually independent of the input signal $V_i(s)$. The controller output is in this case given by

$$R'(s) = \frac{G_c(s)}{1 - G_c(s)F(s)}\Big[V_i(s)[F(s) - F'(s)] + E(s)\Big]. \tag{2.22}$$

It is clear that when the system is unstable or in a state of over-modulation, the difference in output between $F(s)$ and $F'(s)$ will be large, causing the controller to saturate. When the controller is saturated, $F(s)$ receives the input signal $V_i(s)$ directly, meaning the controller essentially operates as an open circuit. As the system returns from instability or over-modulation, the difference in output between $F(s)$ and $F'(s)$ will decrease and the controller will return to normal operation.

2.7 Conclusion

This section covered the theoretical background which is necessary for the rest of the project. The different distortion mechanisms which are present in the system were discussed. The modelling of the comparator was investigated and the traditional average model was compared to a more accurate model. Ripple compensation was discussed along with several different implementations thereof. Different controller topologies were investigated along with stabilising techniques to ensure that conditionally stable systems

remain stable under all circumstances. The next chapter will discuss the design and modelling of the system.

Chapter 3

Modelling and Design

3.1 Introduction

This section will discuss the modelling and design of the system. Each component of the system is investigated and the appropriate method of design is chosen. An expression for the gradient of the controller input is determined along with the system's noise transfer function. When designing the controller, the constraints of the practical system is incorporated and the controller is designed to ensure that the system meets the pre-determined requirements. The stability and noise transfer function of the system is then analysed.

3.2 System Model

The basic system model was briefly discussed in Section 2.2.2. The focus there, however, was on the implementation of the accurate small-signal model in the closed-loop system. The system model as a whole is shown in Figure 3.1. The system contains a low-pass filter $F(s)$ at the output to attenuate the high-frequency components in the output of the power stage. A time delay e^{-st_d} is included to account for the delays of the different components in the system. This includes the delays of the ADC in the feedback loop, gate drivers and the digital PWM. The power stage is modelled using the gain V_d which is equal

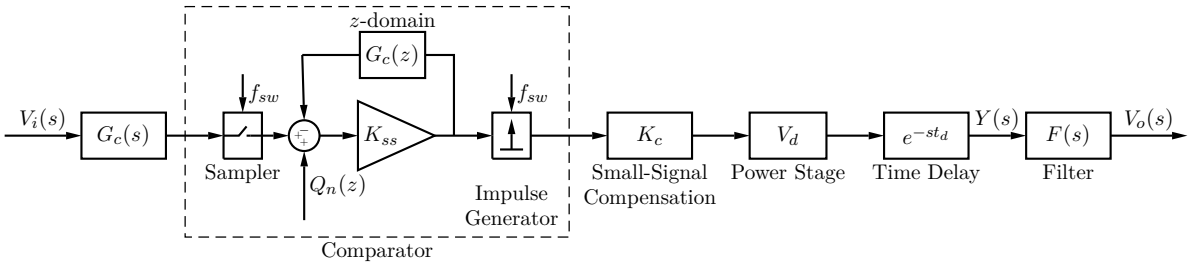


Figure 3.1: System model with accurate small-signal comparator model.

to the rail voltage and a small-signal compensation term K_c is included. The controller $G_c(s)$ along with the accurate small-signal model is also visible. The quantization noise which is induced into the system by the pulse-width modulator is represented by $Q_n(z)$.

3.3 PWM Input Gradient

As previously mentioned in Section 2.2, the gradient of the PWM input signal influences the small-signal gain K_{ss} of the pulse-width modulator. It therefore also effects the small-signal compensation term K_c since it is the inverse of K_{ss}. When looking at

$$K_{ss} = \left[\frac{2f_s}{2f_s - \dot{f}(t_{sp})}\right], \tag{3.1}$$

it is clear that the gradient of the PWM input signal at the transition points with the PWM sawtooth carrier $\dot{f}(t_{sp})$ must be determined to obtain the value of K_c. Figure 3.2 illustrates a typical PWM input signal along with the sawtooth carrier. The transition points of the two signals, which are the points of interest, are marked with dots.

In order to derive a mathematical expression for $\dot{f}(t_{sp})$, a couple of assumptions have to be made. Firstly, it will be assumed that the input of the system is a DC signal. This will ensure a constant duty cycle. Secondly, it is assumed that the PWM input signal is periodic. Figure 3.3 illustrates a simplified system model with ripple compensation. It can be seen that the input of the demodulation filter $X(s)$ will be a sawtooth waveform. This is due to the ripple compensation which is added to the PWM output after passing

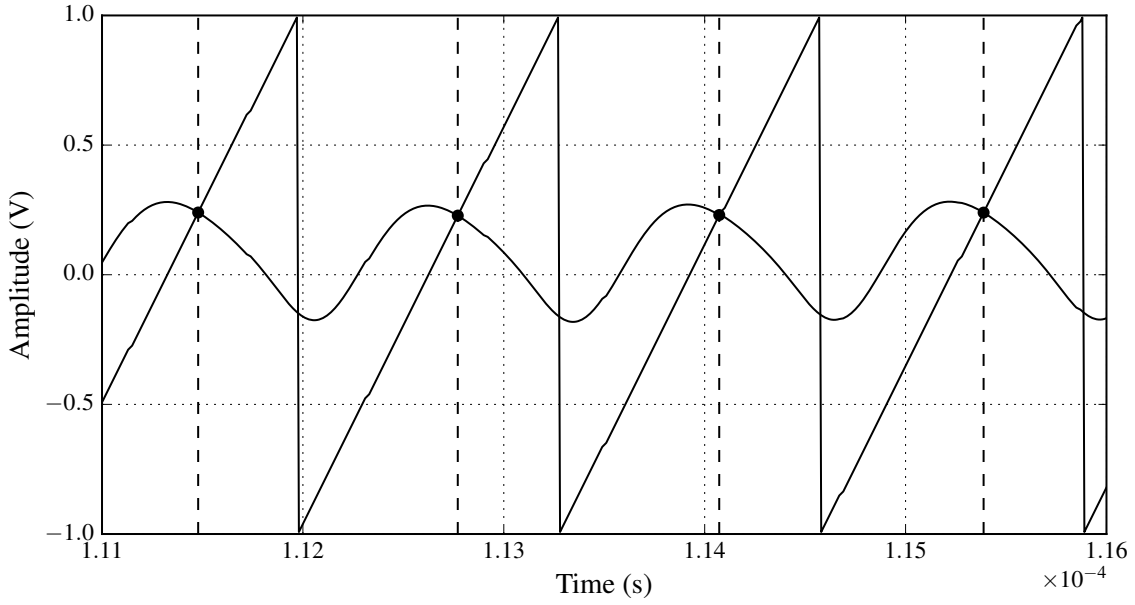


Figure 3.2: Typical PWM input signal with sawtooth carrier.

through the power stage. The demodulation filter input $X(s)$ can thus be described in the time domain by

$$x(t) = \frac{2V_d}{T_s}t, \quad \text{for } \frac{-T_s}{2} \leq t < \frac{T_s}{2}, \tag{3.2}$$

where T_s is the switching period and the DC component of the signal is ignored. Since an expression for $y'(t)$ is required, and an expression for $x(t)$ is known, the system in Figure 3.3 can be simplified to that seen in Figure 3.4. The output of the controller is described by

$$Y(s) = \frac{-X(s)F(s)G_c(s)}{V_d}. \tag{3.3}$$

It is clear that the V_d term is nullified by the V_d term in $X(s)$. When combining $F(s)$ and $G_c(s)$ into a single transfer function $H(s)$, the output of the controller is described by

$$Y(s) = -X(s)H(s). \tag{3.4}$$

The concept of superposition will be used to determine a time-domain expression for $Y(s)$. The transfer function $H(s)$ is decomposed into partial fractions in the form of

$$H_n(s) = \frac{k_n}{s - a_n}. \tag{3.5}$$

The differential equation for a single term can then be solved and applied to every partial fraction component. The derivatives of every output of the partial fraction components can then be added together to produce the total output of the controller. The concept is illustrated in Figure 3.5. The differential equation

$$u_i'(t) - a_i u_i(t) = -\frac{2k_i}{T_s}t, \tag{3.6}$$

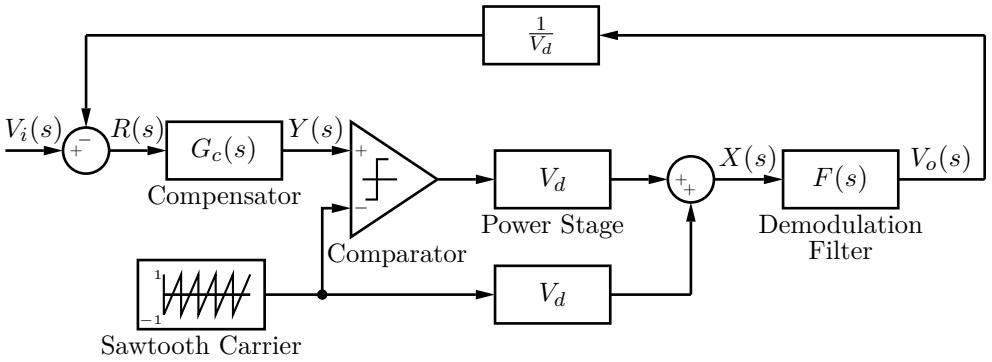


Figure 3.3: Simplified system with Ripple Compensation.

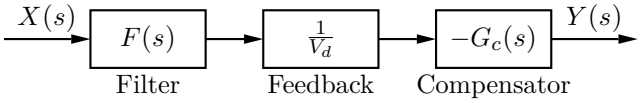


Figure 3.4: Simplified system for PWM input gradient calculation.

needs solving to obtain an expression for $u_i'(t)$. To solve this equation, the particular and homogeneous solutions are obtained and combined to yield

$$u_i(t) = c_1 e^{a_i t} + \frac{2k_i}{a_i T_s} t + \frac{2k_i}{a_i^2 T_s} + c_2, \tag{3.7}$$

where c_1 and c_2 are integration constants. There are not enough initial conditions to determine the value of c_2, it is, however, not needed when calculating the PWM input gradient as it is a constant which will equate to zero when the derivative is taken. The value of c_1 can be determined by assuming $u_i(t)$ is periodic, thus

$$u_i\left(\frac{-T_s}{2}\right) = u_i\left(\frac{T_s}{2}\right). \tag{3.8}$$

Solving for c then results in:

$$c_1 = \frac{2k_i}{a_i\left[e^{\frac{-a_i T_s}{2}} - e^{\frac{a_i T_s}{2}}\right]}. \tag{3.9}$$

The time-domain output of every partial fraction can therefore be determined using

$$u_i(t) = \frac{2k_i e^{a_i t}}{a_i\left[e^{\frac{-a_i T_s}{2}} - e^{\frac{a_i T_s}{2}}\right]} + \frac{2k_i}{a_i T_s} t + \frac{2k_i}{a_i^2 T_s} + c_2. \tag{3.10}$$

The derivative is obtained and yields

$$u_i'(t) = \frac{2k_i e^{a_i t}}{\left[e^{\frac{-a_i T_s}{2}} - e^{\frac{a_i T_s}{2}}\right]} + \frac{2k_i}{a_i T_s}. \tag{3.11}$$

An expression for the controller input derivative $y'(t)$ can now be obtained by summing the output derivatives of the individual partial fraction components. The gradient of the

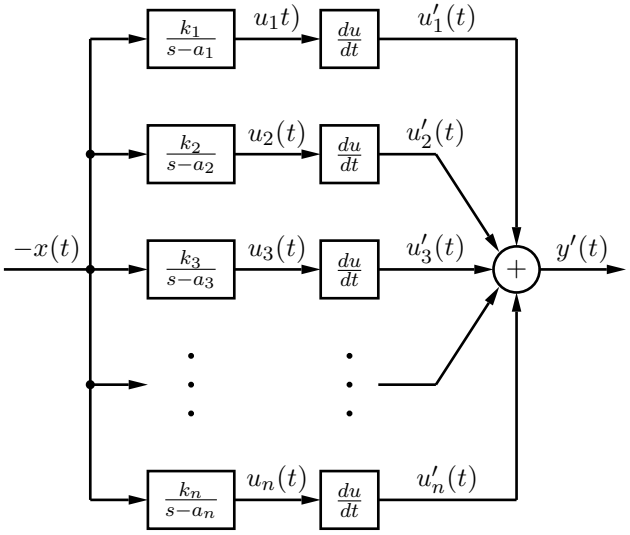


Figure 3.5: Superposition concept used to determine PWM gradient expression.

PWM input signal can thus be determined using

$$\frac{dy}{dt} = \sum_{n=1}^{N} \left(\frac{2k_n e^{a_n t}}{e^{\frac{-a_n T_s}{2}} - e^{\frac{a_n T_s}{2}}} + \frac{2k_n}{a_n T_s} \right), \tag{3.12}$$

where N is the number of partial fractions which the combined transfer function in Figure 3.4 was expanded into. The constant duty cycle of the PWM output makes it possible to determine the gradient at the transition points of the PWM carrier and the controller output.

3.4 Noise Transfer Function

The noise transfer function of a system conveys important information regarding the performance thereof. In this case, it is of interest to determine the amount with which the quantization noise in the system is attenuated. The noise transfer function of the system with the average PWM model included, as seen in Figure 2.8a, is given by

$$NTF(s) = \frac{Y(s)}{Q_n(s)} = \frac{V_d e^{-st_d}}{1 + F(s)G_c(s)V_d e^{-st_d}}. \tag{3.13}$$

It is clear that the transfer function is defined with the term $Q_n(s)$ as input and the input of the output filter $Y(s)$ as output. This causes the output filter $Y(s)$ to effectively be eliminated from the NTF. This is done to clarify the attenuation ability of the controller. Once the accurate PWM small-signal model is included in the system, as seen in Figure 3.1, the expression for the noise transfer function must be adapted. The first step is to calculate the small-signal closed-loop transfer function of the system, and is given by

$$V_o(\omega) = V_i(\omega) \left(\frac{G_c(j\omega)K_c V_d e^{-j\omega t_d} F(j\omega)}{1 + G(e^{j\omega T_s})} \right) + Q_n(\omega) \left(\frac{K_c V_d e^{-j\omega t_d} F(j\omega)}{1 + G(e^{j\omega T_s})} \right), \tag{3.14}$$

where

$$G(e^{j\omega T_s}) = G(z) = \mathcal{Z}\{G_c(s)F(s)K_c V_d e^{-st_d}\}. \tag{3.15}$$

The z-transform using the impulse invariance method, discussed in section 2.3, is denoted using the $\mathcal{Z}$ symbol. Since we are trying to determine whether the quantization noise is adequately attenuated, only the second term of (3.14) is relevant. When substituting $V_o(\omega)$ with $Y(\omega)F(j\omega)$, the noise transfer function simplifies to

$$NTF(\omega) = \frac{Y(\omega)}{Q_n(\omega)} = \frac{K_c V_d e^{-j\omega t_d}}{1 + G(e^{j\omega T_s})}. \tag{3.16}$$

3.5 Controller design

This section will discuss the requirements, constraints and design of the controller. The system will be digitally controlled and implemented using an FPGA. Quantization noise will therefore have the biggest influence on the system with regards to noise. The system will have two sources of quantization noise. Both the pulse-width modulator and the ADC in the feedback loop will induce noise into the system. The choice of ADC is very important as it will essentially determine the overall performance quality of the system. A multi-bit sigma-delta ADC is chosen as it complies with the need of having a high sampling frequency and the signal-to-noise ratio is also satisfactory. Along with the quantization noise of the ADC, which is aliased into the audio band due to the sampling nature of the PWM, the ADC also causes a time delay in the feedback loop. The delay is minimised by using the modulator output of the ADC. This bypasses the built-in low-pass FIR filters and a minimum phase IIR filter can then be used to attenuate some of the quantization noise contributed by the ADC [4]. The delay does, however, still present a notable challenge with regards to the design of the controller.

Since the system is an audio amplifier, the aim of the controller is to adequately attenuate the quantization noise in the audio band (20 Hz to 20 kHz). There are multiple noise shaping techniques which can be used to reduce the effect of quantization noise as discussed in [19]. In the following work, however, an approach is chosen in which the controller design is used to adequately attenuate the quantization noise. To achieve this, the controller must have an open-loop gain of 50 dB or more in the audio band [5]. By ensuring that the quantization noise is adequately attenuated, the system obtains the desired flat frequency response across the audio band and a very high quality output signal.

3.5.1 Design Approach and Constraints

Before designing the controller, it is important to consider the constraints which will be placed upon the controller when practically implemented. High quality audio is commonly classified as audio with a bit depth of 24 bits which is sampled at 192 kHz. This requires the FPGA to use a 24-bit counter for the system, which is used to generate the PWM sawtooth carrier. The required FPGA clock frequency can be determined using

$$f_{FPGA} = f_{audio} \cdot 2^{\text{Counter bits}}. \tag{3.17}$$

It is clear that if the audio input has a bit depth of 24, and is sampled at 192 kHz, the required clock speed of the FPGA is more than 3 THz. It is not possible to clock an FPGA at such a high clock frequency.

This problem is overcome by increasing the switching frequency, and thus the sampling rate, of the system. This will reduce the requirements of the quantizer. This process is known as oversampling [10]. When selecting the switching frequency of the system, several factors need to be considered. As previously stated, it is impossible to clock the FPGA at a high enough speed if the switching frequency is too low since this will increase the required bit depth of the quantizer. If the switching frequency is too high, the switching losses will increase dramatically and other factors like EMI will have a significant influence on the system. The switching frequency is therefore chosen as 768 kHz as it is a good midway point. The controller will be implemented using an FPGA which is clocked at 98.304 MHz. The system will thus have a 7-bit pulse-width modulator. The system will be designed in the continuous-time domain using a root-locus approach, but stability will be assessed in the discrete-time domain to ensure accuracy. Simulations will be done in an environment which emulates the FPGA environment. The system requires a gain margin of at least 3 dB to accommodate for any component inaccuracies in the analogue part of the system. As discussed in Section 2.2.2, the stability of the system is solely dependent on

$$G_c(z) = \mathcal{Z}\{G_c(s)F(s)K_cV_de^{-st_d}\}. \tag{3.18}$$

The continuous-time controller $G_c(s)$ will consist of three different sections. A pole cancellation filter, low-pass filter and a chain of integrators will be combined to ensure the system has the required gain across the audio band while remaining stable.

3.5.2 Pole Cancellation Filter

A second order filter is used for the cancellation of the output low-pass filter poles. A complex zero pair is placed in close vicinity to these poles. Even though this will not provide exact cancellation of the poles, it will ensure that the system remains stable. It is, however, not possible to practically implement a filter with only a complex pair of zeros. A high-frequency complex pole pair is therefore added to attenuate the quantization noise of the ADC. The filter now has a biproper transfer function, since the degrees of the denominator and numerator are equal, and can be written in the form

$$G_{pc}(s) = \frac{a_0s^2 + a_1s + a_2}{s^2 + b_1s + b_2}. \tag{3.19}$$

Due to the noise profile of the ADC, which is discussed in Chapter 5, the frequency of the complex pole pair is chosen as 1.88 MHz. The damping ratio ζ is 0.99991. With the output stage low-pass filter having a cut-off frequency of 24.4 kHz, as discussed in Section 3.6, the complex zero pair is placed at 23.2 kHz with ζ equal to 0.97023. This

will result in a transfer function of

$$G_{pc}(s) = \frac{6295.0713(s^2 + 2.83 \cdot 10^5 s + 2.127 \cdot 10^{10})}{s^2 + 2.362 \cdot 10^7 s + 1.395 \cdot 10^{14}}. \tag{3.20}$$

The magnitude and phase response of the filter can be seen in Figure 3.6 and 3.7 respectively.

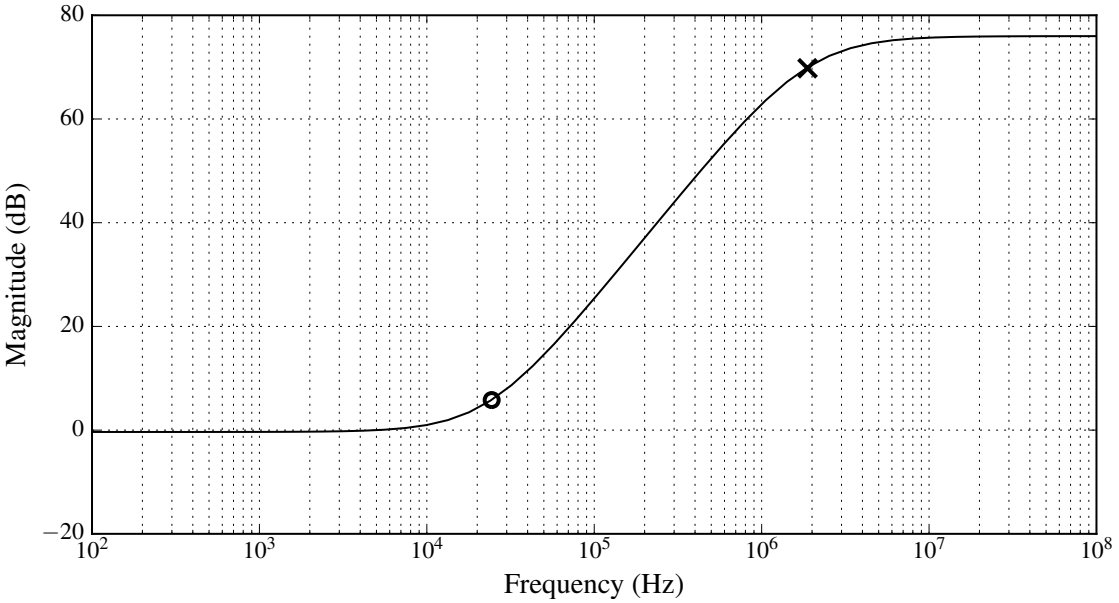


Figure 3.6: Magnitude response of pole cancellation filter.

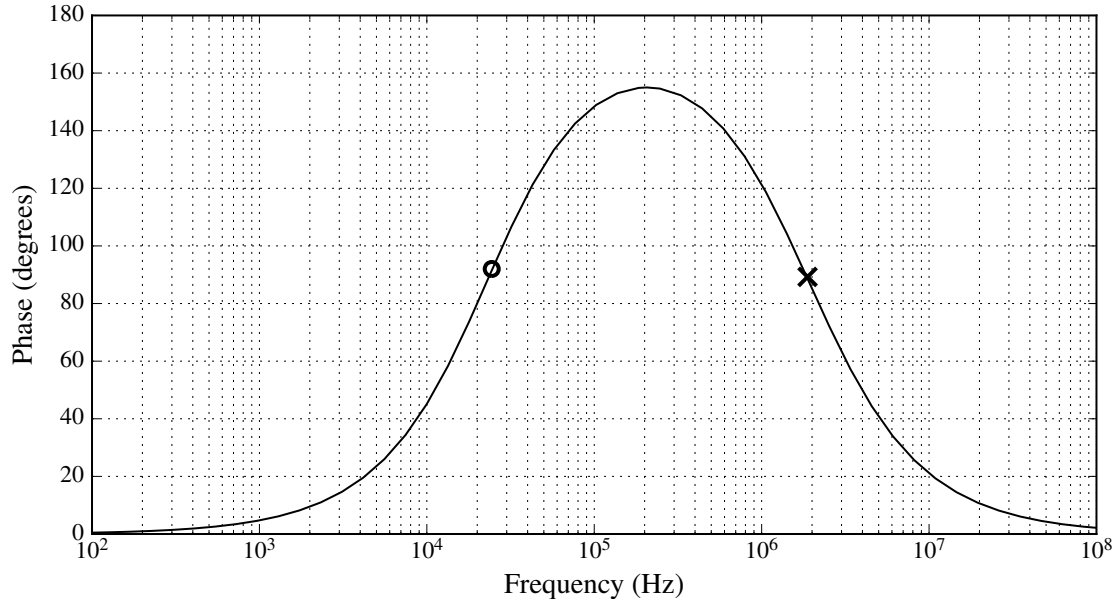


Figure 3.7: Phase response of pole cancellation filter.

3.5.3 Low-Pass Filter

The second section of the controller will be a second order low-pass filter. The goal of this section is to attenuate the quantization noise of the ADC. A complex pole pair is therefore introduced to the system, along with a high frequency complex zero pair. Even though the zero pair potentially increases quantization noise in surrounding frequencies, it is necessary to ensure stability of the system. The zero pair will also be placed in a frequency range which is not of interest when analysing the performance of the system. The filter will be implemented in the form of (3.19). The complex pole pair is placed at 1.15 MHz with a damping ratio of 0.3808. The choice of frequency is due to the fact that the quantization noise of the ADC starts to increase at approximately 1.5 MHz. The complex zero pair is placed at 30.4 MHz with a damping ratio of 0.99958. The filter transfer function is

$$G_{ADLPF}(s) = \frac{0.0014332(s^2 + 3.822 \cdot 10^8 s + 3.655 \cdot 10^{16})}{(s^2 + 5.503 \cdot 10^6 s + 5.221 \cdot 10^{13})}. \tag{3.21}$$

The magnitude and phase response of the filter can be seen in Figure 3.8 and 3.9.

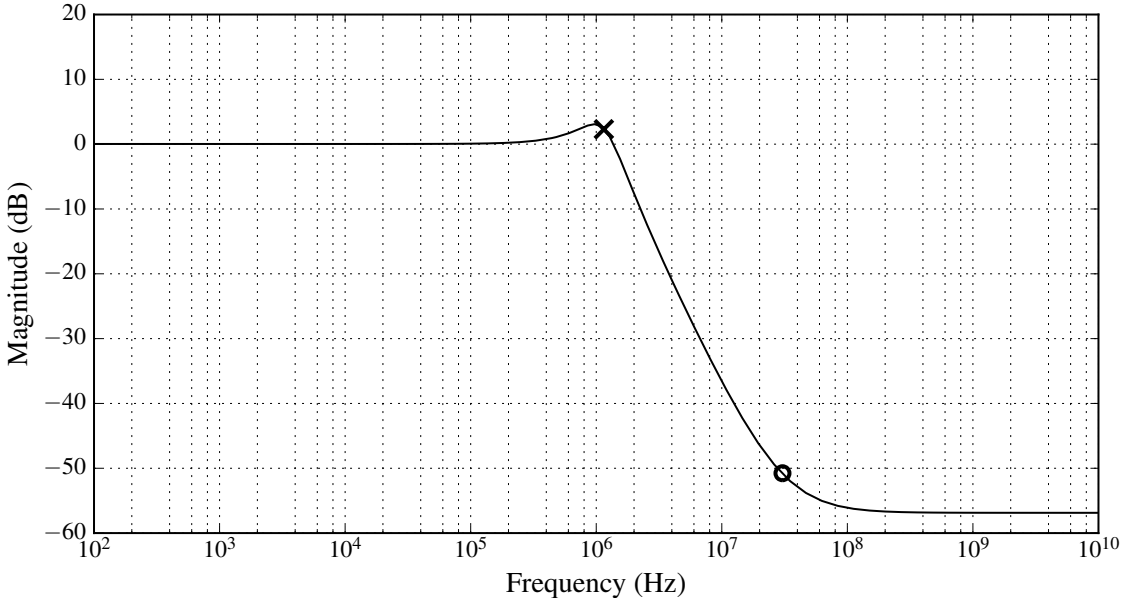


Figure 3.8: Magnitude response of ADC compensation low-pass filter.

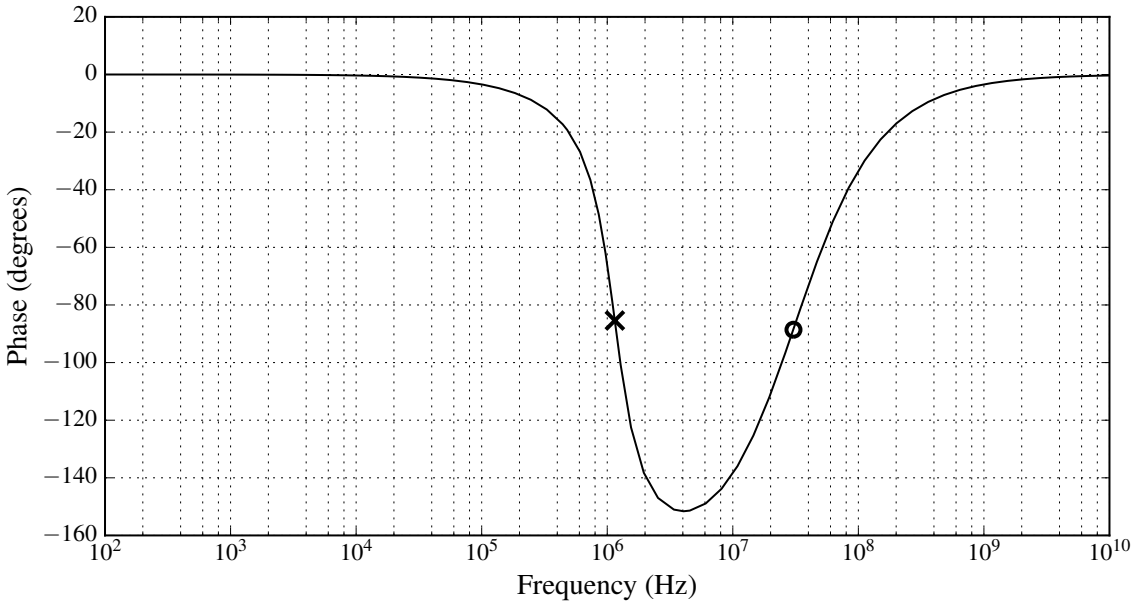


Figure 3.9: Phase response of ADC compensation low-pass filter.

3.5.4 Chain of Integrators

The first two sections of the controller were designed to compensate for quantization noise and stability issues caused by the external system components such as the ADC and output stage. The third section, the chain of integrators, is used to provide the desired gain across the audio band. Different controller topologies were discussed in Section 2.5. The chain of integrators with feed forward summation and local resonator feedback loops topology was chosen due to the fact that it is important to minimise the propagation delay of the system. It is desirable to ensure that the controller has an open-loop gain of 50 dB or more to adequately attenuate the quantization noise throughout the audio band [5]. To achieve this, the controller configuration in Figure 2.13 will be expanded by adding another pair of integrators with a local resonator feedback loop and feed forward summation to the chain. This will enable the chain of integrators to contribute an integrator, two complex pole pairs, two complex zero pairs and enable the gain of the system to be implemented using the feed forward summation factors k_i. Figure 3.10 illustrates the configuration. The values of γ_1 and γ_2 will determine the locations of the two complex pole pairs. The locations of the two complex zero pairs is determined by the feed forward summation coefficients k_i. To determine the values of k_i which implements the required transfer function, it is first necessary to determine an expression for the output $Y(s)$. The outputs of the integrators are respectively defined

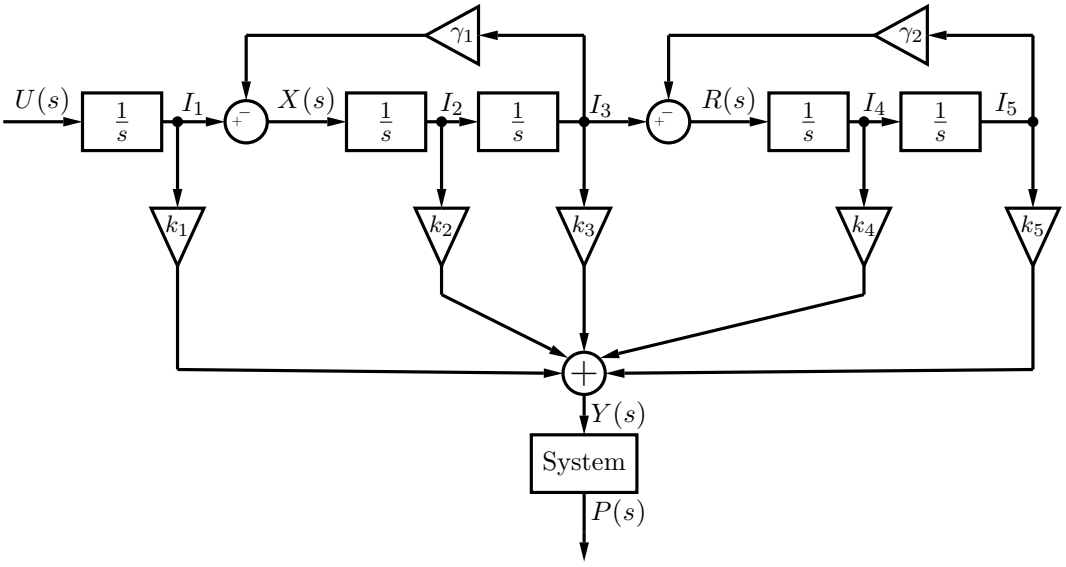


Figure 3.10: Expanded chain of integrators with feed forward summation and local resonator feedback loops.

as

$$
\begin{aligned}
I_1(s) &= \frac{U(s)}{s}, \\
I_2(s) &= \frac{U(s)}{s^2+\gamma_1}, \\
I_3(s) &= \frac{U(s)}{s(s^2+\gamma_1)}, \\
I_4(s) &= \frac{U(s)}{(s^2+\gamma_1)(s^2+\gamma_2)}, \\
I_5(s) &= \frac{U(s)}{s(s^2+\gamma_1)(s^2+\gamma_2)}.
\end{aligned}
\tag{3.22}
$$

When substituting

$$
\begin{aligned}
a_1 &= s^2+\gamma_1, \\
a_2 &= s^2+\gamma_2,
\end{aligned}
\tag{3.23}
$$

the transfer function from the input $U(s)$ to the output $Y(s)$ can be written as

$$
\frac{Y(s)}{U(s)} = \frac{k_1}{s} + \frac{k_2}{a_1} + \frac{k_3}{sa_1} + \frac{k_4}{a_1a_2} + \frac{k_5}{sa_1a_2}. \tag{3.24}
$$

The equation is then simplified to have a common denominator which results in

$$
\frac{Y(s)}{U(s)} = \frac{k_1a_1a_2 + k_2a_2s + k_3a_2 + k_4s + k_5}{sa_1a_2}. \tag{3.25}
$$

It is clear that the denominator of (3.25) is only dependent on the γ_i values of the controller which can be determined using (2.18). To determine the values of k_i, it is first necessary to determine the transfer function describing the position of the controller zeros

$$G_{zeros}(s) = (s - z_1)(s - z_2)(s - z_3)(s - z_4), \tag{3.26}$$

where z_i is the continuous-time domain position of the respective zero. Equation (3.26) can then be equated to the numerator of (3.25) to determine the k_i coefficients.

To obtain enough gain across the audio band, the two complex pole pairs are placed at 10.6 kHz and 17.8 kHz. The value of γ_1 is therefore equal to $4.4358 \cdot 10^9$ and γ_2 is equal to $1.2508 \cdot 10^{10}$. The associated zeros are located at 28.393 kHz and 43.035 kHz with damping ratios of 0.77431 and 0.83204 respectively. The transfer function describing zero positions is thus

$$G_{zeros}(s) = s^4 + c_1 s^3 + c_2 s^2 + c_3 s + c_4, \tag{3.27}$$

where

$$\begin{aligned} c_1 &= 7.262 \cdot 10^5, \\ c_2 &= 2.293 \cdot 10^{11}, \\ c_3 &= 3.452 \cdot 10^{16}, \\ c_4 &= 2.327 \cdot 10^{21}. \end{aligned} \tag{3.28}$$

After equating the numerator of (3.25) to (3.26), the following relationships are obtained which can be used to determine the values of k_i:

$$\begin{aligned} c_1 &= \frac{k_2}{k_1}, \\ c_2 &= \gamma_1 + \gamma_2 + \frac{k_3}{k_1}, \\ c_3 &= \frac{k_2}{k_1}\gamma_2 + \frac{k_4}{k_1}, \\ c_4 &= \gamma_1\gamma_2 + \frac{k_3}{k_1}\gamma_2 + \frac{k_5}{k_1}. \end{aligned} \tag{3.29}$$

The coefficient k_1 is chosen as $9.153 \cdot 10^4$ to ensure the controller will obtain the required 50 dB gain across the audio band. Since all the other coefficients are dependent on k_1, this value can be tweaked to optimise the system gain to ensure stability. The k_i coefficient

values are initially calculated as

$$\begin{aligned} k_1 &= 9.1530 \cdot 10^4, \\ k_2 &= 6.6473 \cdot 10^{10}, \\ k_3 &= 1.9433 \cdot 10^{16}, \\ k_4 &= 2.3282 \cdot 10^{21}, \\ k_5 &= -3.5164 \cdot 10^{25}. \end{aligned} \tag{3.30}$$

This results in (3.25) simplifying to

$$G_{ci} = \frac{9.153 \cdot 10^4 s^4 + 6.647 \cdot 10^{10} s^3 + 2.098 \cdot 10^{16} s^2 + 3.16 \cdot 10^{21} s + 2.13 \cdot 10^{26}}{s^5 + 1.694 \cdot 10^{10} s^3 + 5.549 \cdot 10^{19} s}. \tag{3.31}$$

The magnitude response of $G_{ci}(s)$, which is shown in Figure 3.11, illustrates that the controller gain is slightly lower in the higher range of the audio band, albeit very close, compared to the required 50 dB. The gain is initially kept slightly lower to ensure stability upon initial implementation. Once the amplifier is operating in a stable region, the gain will be optimized to ensure a 50 dB gain across the whole audio band. The phase response is shown in Figure 3.12.

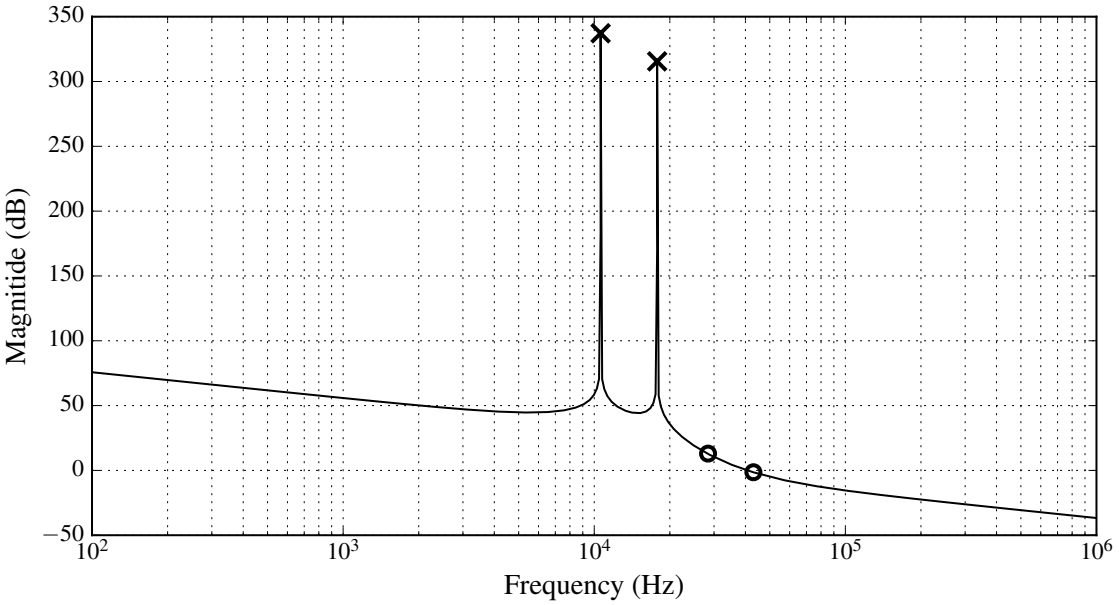


Figure 3.11: Magnitude response of the chain of integrators.

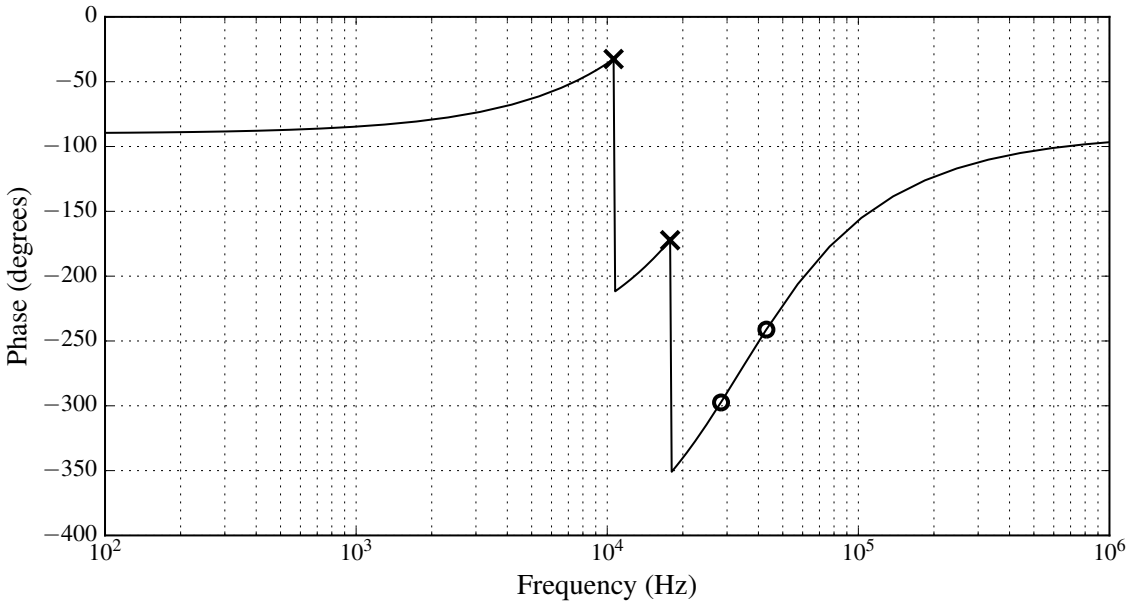


Figure 3.12: Phase response of the chain of integrators.

3.6 Output Stage Low-Pass Filter

The demodulation filter $F(s)$ is used to attenuate the high-frequency components of the PWM output. The corner frequency of $F(s)$ is usually chosen to be around 30 kHz. This will ensure that $F(s)$ does not affect the system's response in the audio band, but adequately attenuates high-frequency components. Figure 3.13 illustrates the simple low-pass filter with accompanying waveforms. The speaker is initially be modelled as a simple resistor R_L with a value of 8.2 Ω. The transfer function of the low-pass filter is

$$\frac{V_o(s)}{V_i(s)} = \frac{\frac{1}{L_f C_f}}{s^2 + \frac{1}{R_L C_f} s + \frac{1}{L_f C_f}}. \tag{3.32}$$

The hardware for the output stage is not developed in this project. The values of the inductor L_f and capacitor C_f, which are 20.83 μH and 2.04 μF respectively, are therefore

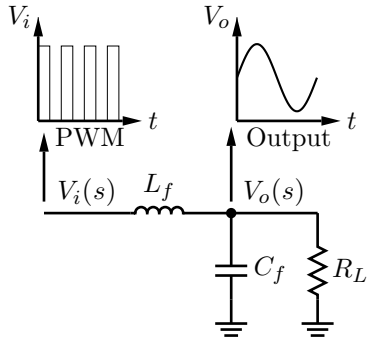


Figure 3.13: Basic Output Filter.

obtained from the pre-developed hardware. The cut-off frequency is determined using

$$f_c = \frac{1}{2\pi\sqrt{L_f C_f}}. \tag{3.33}$$

The transfer function is then calculated to be

$$F(s) = \frac{2.353 \cdot 10^{10}}{s^2 + 5.978 \cdot 10^4 s + 2.353 \cdot 10^{10}}, \tag{3.34}$$

where the filter has a cut-off frequency of 24.415 kHz and a damping ratio ζ of 0.19484. The magnitude response of the low-pass filter is shown in Figure 3.14.

3.7 System plots

In the following section the system will be analysed as a whole. The stability, noise transfer function and PWM input gradient of the system will be determined. Analysis will initially be done in the continuous-time domain after which the system will be converted to the discrete-time domain. The continuous-time open-loop transfer function of the system is

$$G_{OL} = G_c(s) V_d e^{-st_d} K_c F(s), \tag{3.35}$$

where

$$G_c(s) = G_{pc}(s) G_{ad}(s) G_{ci}(s).$$

Note that when (3.35) is used, the accurate PWM small-signal model from Section 2.2.2 is not included since the analysis is done solely in the continuous-time domain. The rail

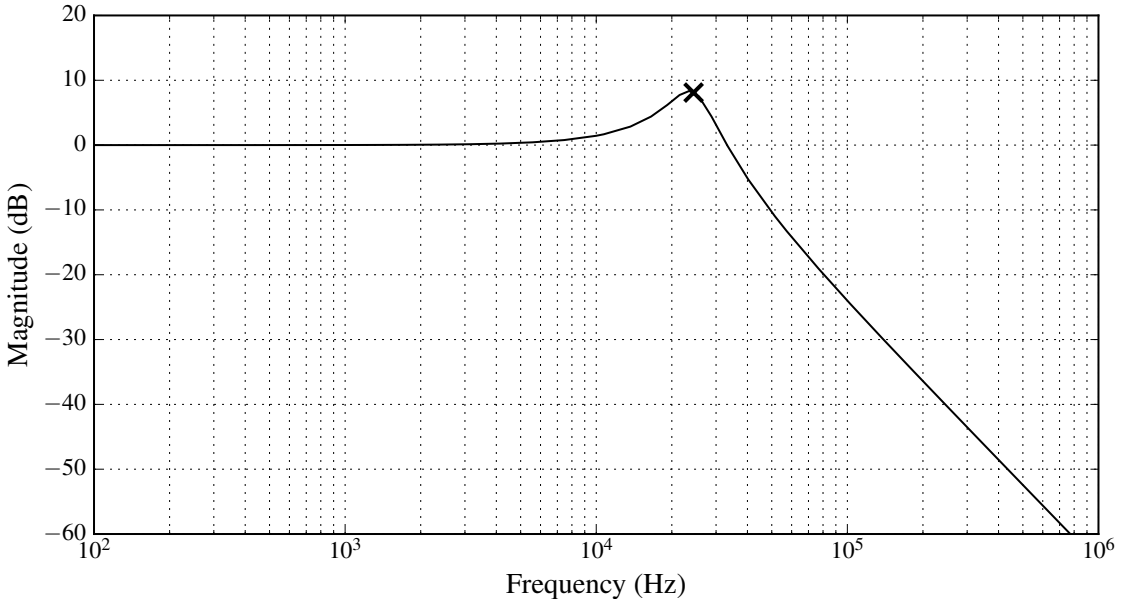


Figure 3.14: Magnitude response of output filter.

Table 3.1: Complex zero and pole positions of complete system.

Section	Pole Frequency	Damping Ratio	Zero Frequency	Damping Ratio
$G_{pc}(s)$	1.88 MHz	0.99991	23.212 kHz	0.97023
$G_{ADLPF}(s)$	1.15 MHz	0.3808	30.427 MHz	0.99958
$G_{ci}(s)$	0 Hz (Integrator)	-1	28.393 kHz	0.77432
	10.6 kHz	0	43.035 kHz	0.83204
	17.8 kHz	0		
$F(s)$	24.415 kHz	0.19484		

voltage V_d is chosen as 15 V, while the propagation delay t_d is equal to 650 ns (calculation of this value is discussed in Chapter 5). The small signal compensation term K_c will initially be assumed to be equal to 1. The complex zero and pole positions, along with their respective damping ratios, of the combined system are shown in Table 3.1. The magnitude of the continuous-time open-loop bode plot is shown in Figure 3.15. When considered together with the phase response, which is shown in Figure 3.16, the gain and phase margins can be determined as 8.52 dB and 31.4 degrees respectively. This indicates that the closed-loop system will be stable. Conclusions regarding the stability of the system can, however, not be made according to continuous-time domain bode plots as the controller will be implemented in the discrete-time domain. Examining the root-locus of the system in the z-domain will provide an accurate indication of the system's stability. The next step is to analyse the stability and performance of the system in the

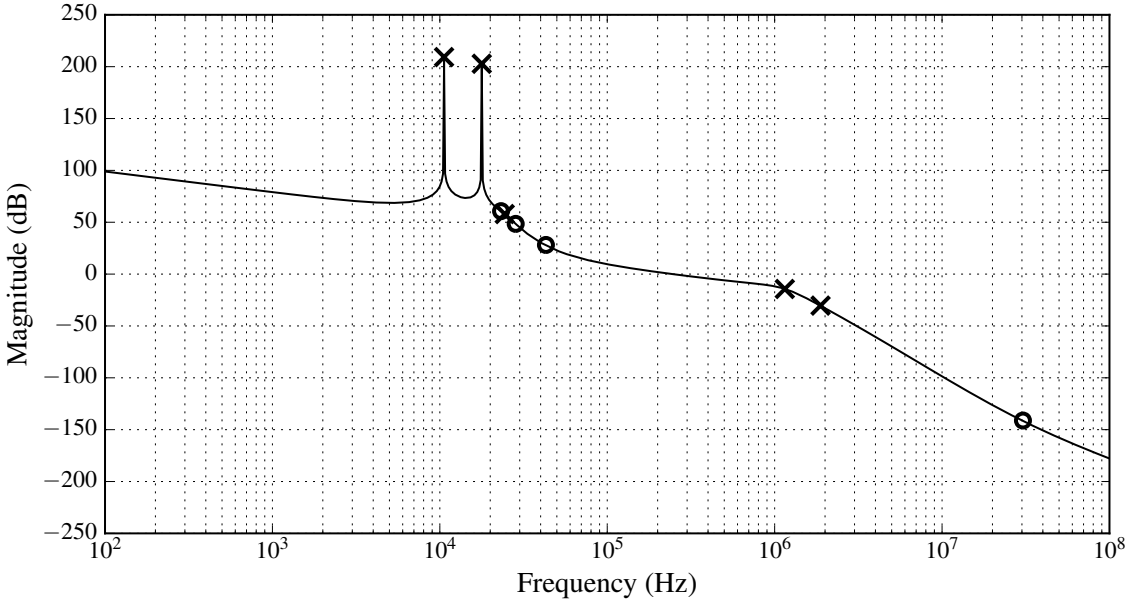


Figure 3.15: Magnitude of Continuous-time Open-Loop Bode Plot.

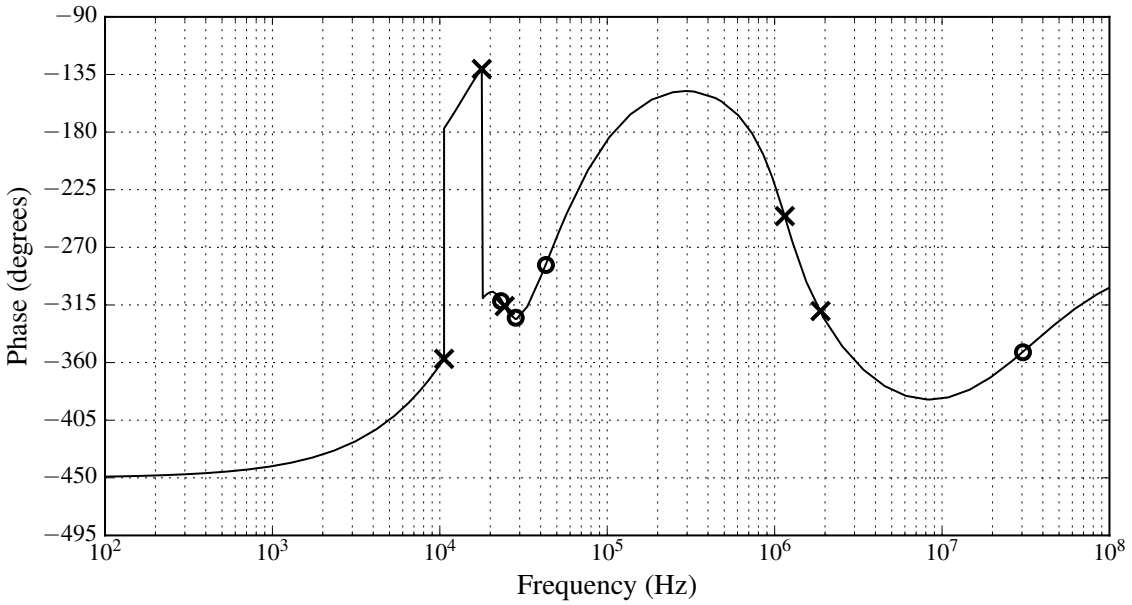


Figure 3.16: Phase of Continuous-time Open-Loop Bode Plot.

discrete-time domain using

$$G_c(z) = \mathcal{Z}\{G_c(s)F(s)K_cV_de^{-st_d}\}. \tag{3.36}$$

This will then include the accurate PWM model from Section 2.2.2 and therefore improve the accuracy of the analysis. The impulse invariance method was used to convert the system to the discrete-time domain with a sampling frequency of 768 kHz. The magnitude response can be seen in Figure 3.17. The response is plotted up to 384 kHz which is the Nyquist frequency. The Nyquist frequency is the minimum sampling frequency which can be used before the signal will be polluted with aliasing errors. Traditionally, this is half of the sampling frequency.

The phase response is shown in Figure 3.18. It can be seen that although the poles translated to the same frequencies as that of the continuous-time domain poles, the locations of the zeros differ. This is due to the time delay which influences the location of the zeros when converted to the z-domain (as discussed in Section 2.3). The complex zero pair of G_{pc} remained at 23.2 kHz. Figure 3.17 further illustrates a real zero at 18.7 kHz and complex zero pairs at 32.6 kHz and 137 kHz. The gain margin in the discrete-time domain is calculated as 3.01 dB along with a phase margin of 6.03 degrees. The root locus of the system, shown in Figure 3.19 where an 8.2 Ω load is used, confirms that the system is stable as the closed-loop poles are all situated inside the unit circle. It is, however, important to note that the system is conditionally stable. This implies that the system could become unstable should the gain collapse or increase too much. Even though the

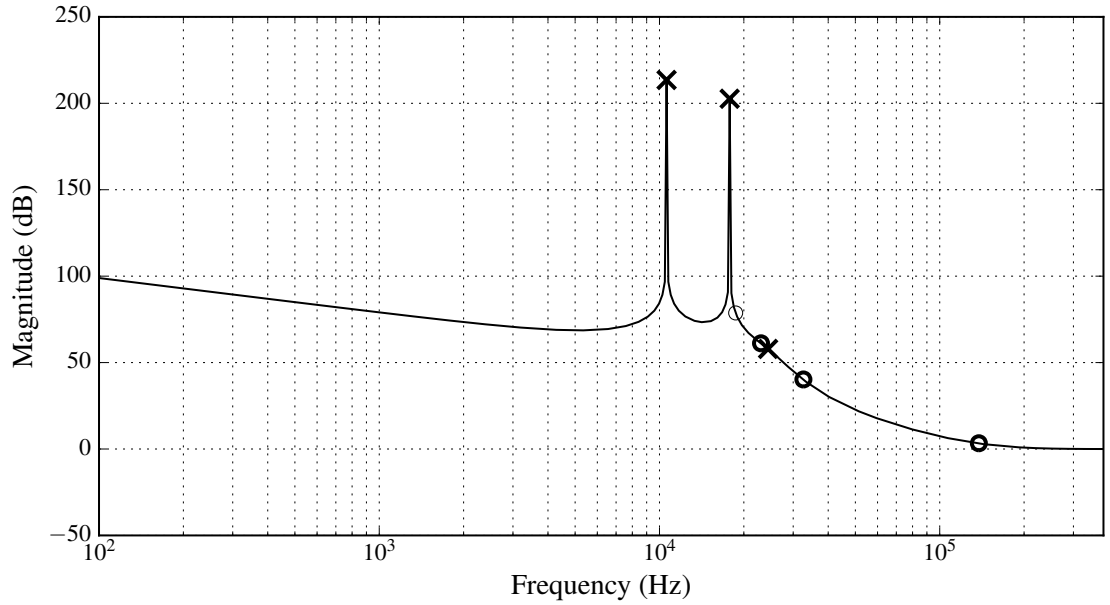


Figure 3.17: Magnitude of discrete-domain Open-Loop Bode Plot.

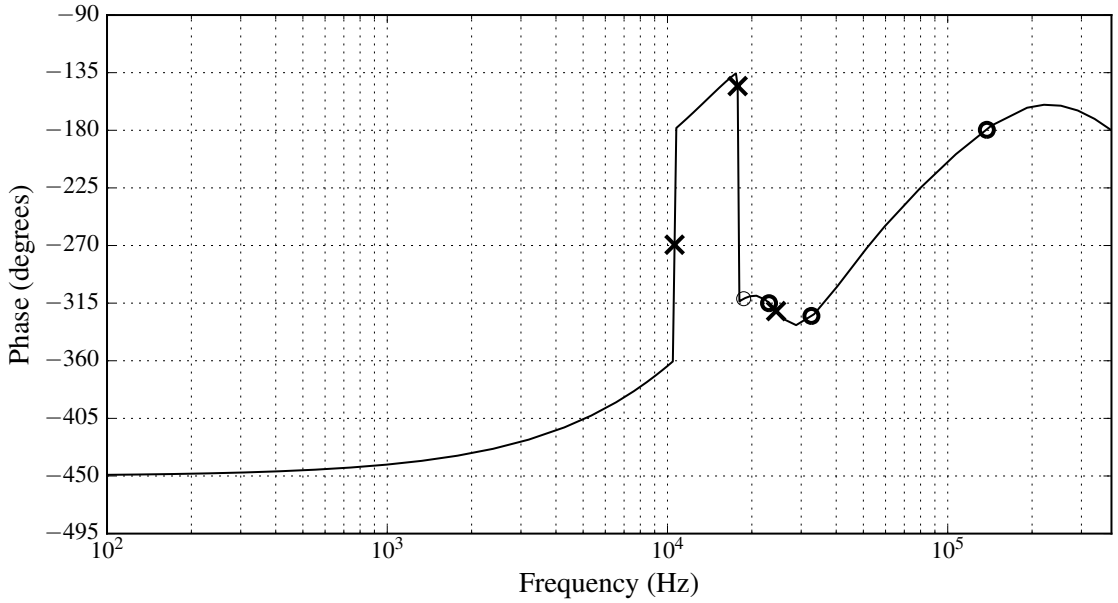


Figure 3.18: Phase of discrete-domain Open-Loop Bode Plot.

system was designed for an 8.2 Ω load, the system is stable when a 4.1 Ω load is attached. The stability margin is, however, reduced to 1.8 dB.

The noise-transfer function, calculated using (3.16), can be seen in Figure 3.20. It is clear that the controller will suppress the quantization noise generated by the PWM by almost 50 dB across the audio band. As previously stated, the target of 50 dB attenuation will be achieved once gain optimisation is done on a stable system. This is in line with the

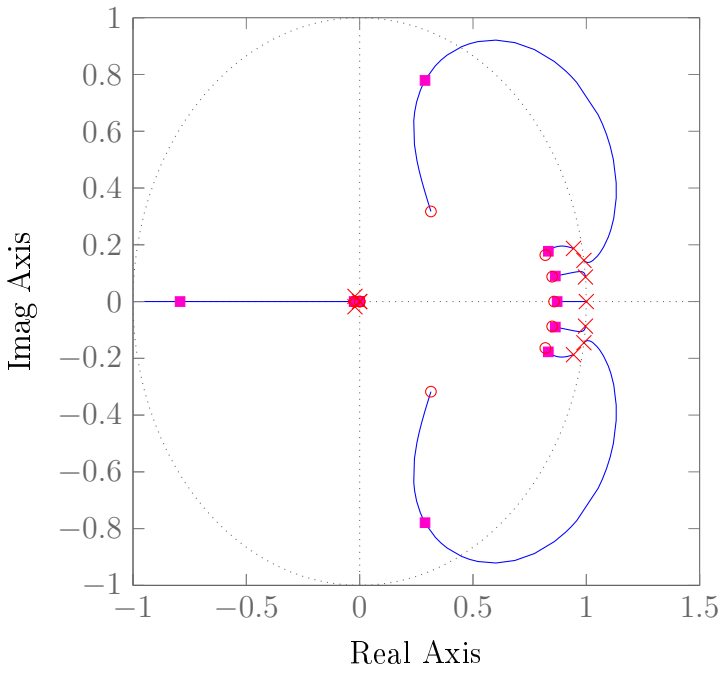


Figure 3.19: Root Locus in discrete-time domain with an 8.2 Ω load.

design requirements which were initially stipulated. When analysing the rapid decline in attenuation ability in the noise-transfer function at frequencies above the audio band, it is clear that quantization noise can not be eliminated. The noise is can merely be shaped away from the frequency band of interest. The downward spikes at 10.6 kHz and 17.8 kHz confirm that the open-loop poles of the system translate to the zeros of the NTF.

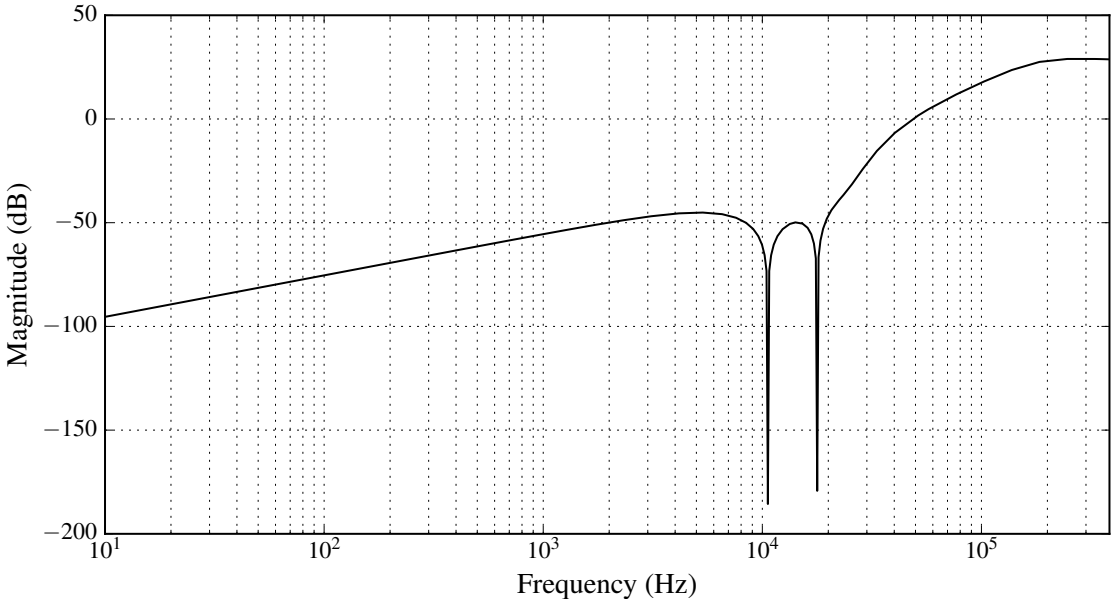


Figure 3.20: Noise-transfer function.

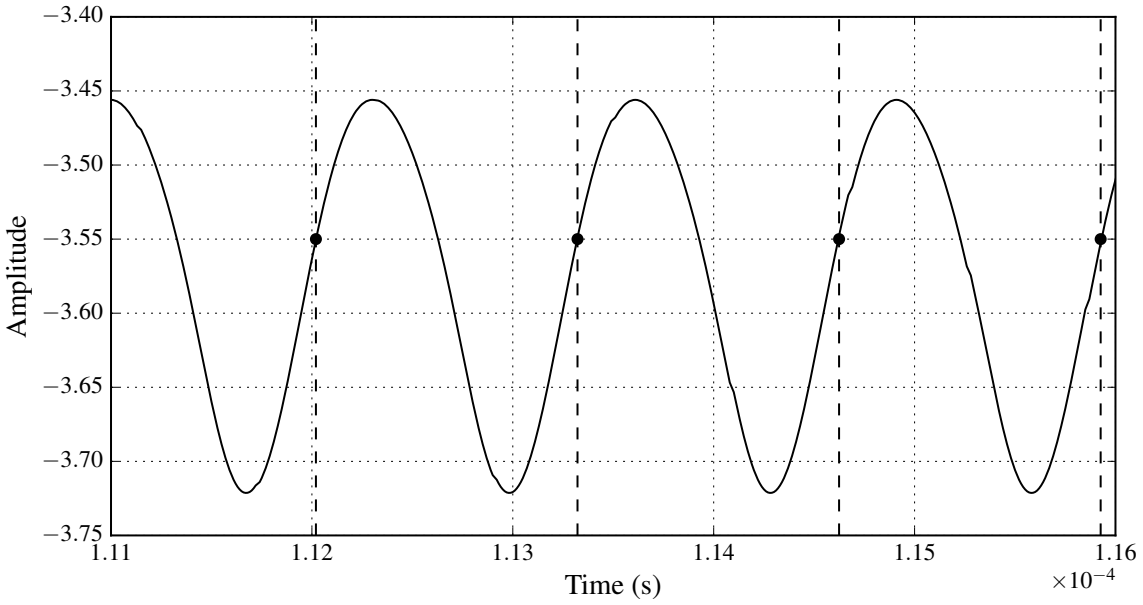


Figure 3.21: Calculated PWM Input signal.

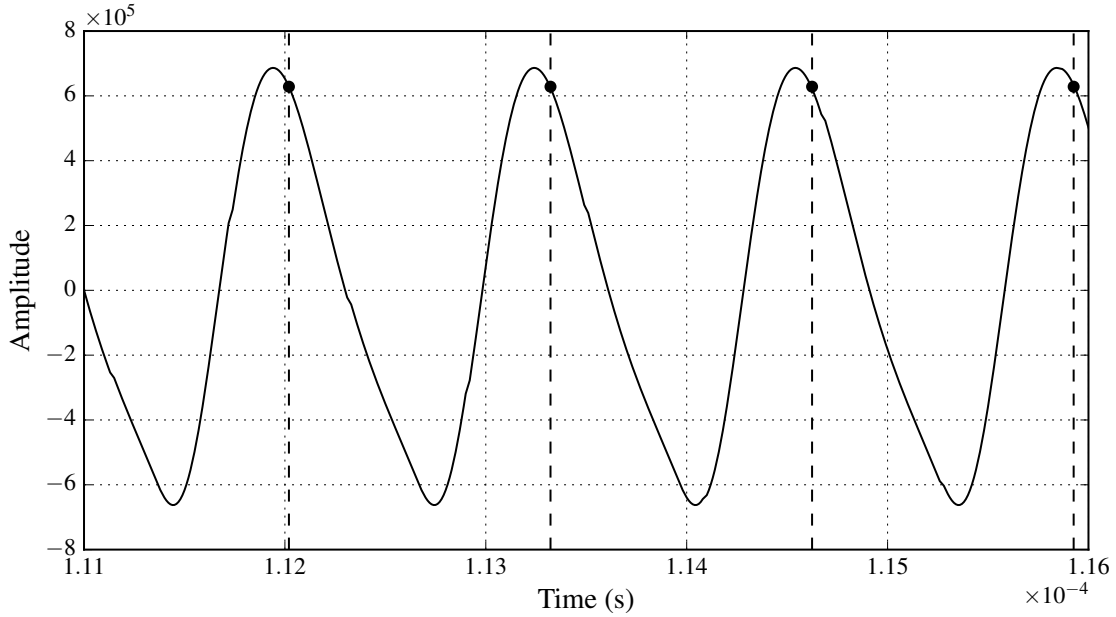


Figure 3.22: Calculated Gradient of PWM Input signal.

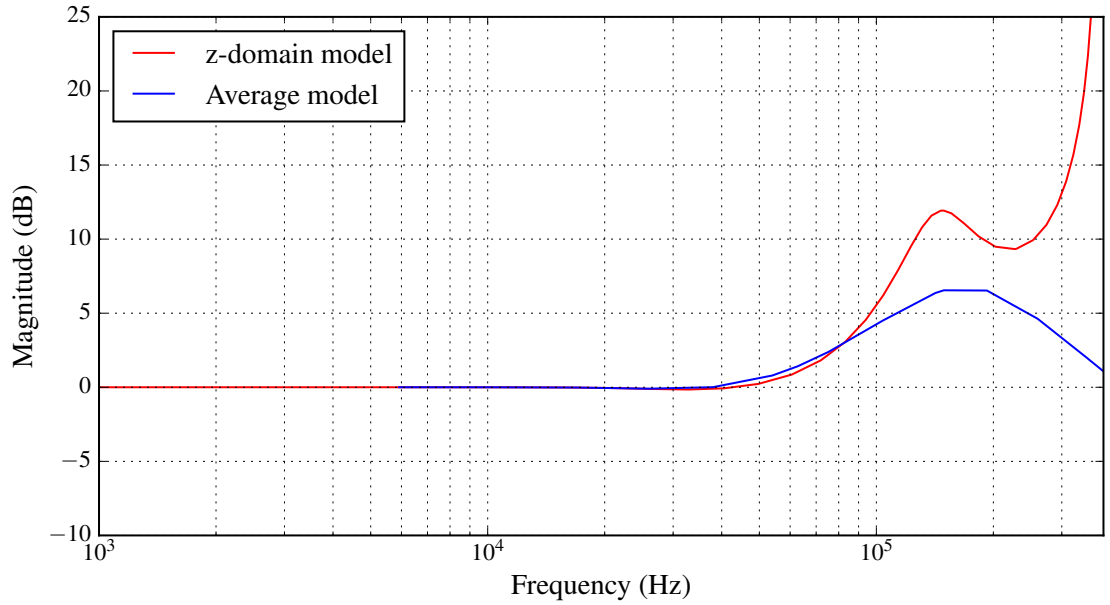
Since the design of the controller has been completed, it is possible to use (3.10) to determine the PWM input signal. The signal is illustrated in Figure 3.21. The DC offset of this signal is not accurate as the the c_2 term is not known. This does not affect the gradient of the signal. The PWM input gradient can be seen in Figure 3.22. It is important to note that only the value of the PWM gradient at the transition point of the PWM input and PWM carrier is of interest as this value will influence the value

of K_{ss}. Due to the fact that the input is assumed to have a constant duty cycle, it is possible to determine the location of these transition points as they will remain periodic. Equation 3.2 describes the demodulation filter input over a period of $\frac{-T_s}{2}$ to $\frac{T_s}{2}$. It can therefore be assumed that the falling edge of the PWM output signal, and thus the transition point of the sawtooth carrier and the PWM input signal, is situated at

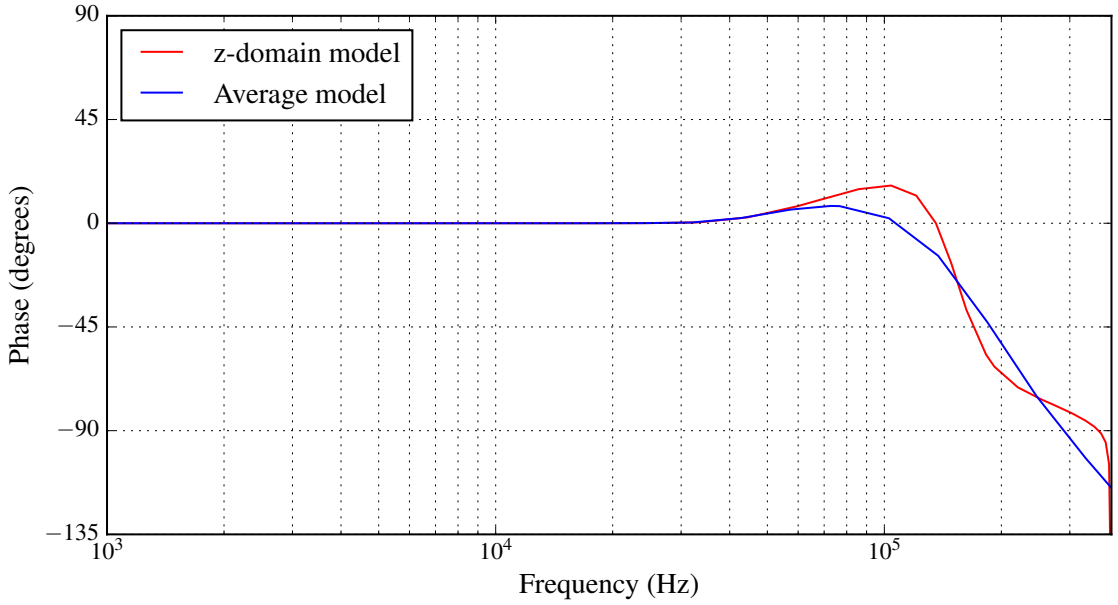
$$\text{Transition Point} = \frac{T_s}{2} + t_d + nT_s, \tag{3.37}$$

where n is the number of switching periods. The propagation delay of the system is also taken into account and the PWM gradient at the transition point is calculated as $6.2835 \cdot 10^5$. This produces a K_{ss} value of 1.6922. The transition points are marked in Figure 3.21 and 3.22.

The closed-loop response of the system is shown in Figure 3.23. The blue line indicates the closed-loop response calculated with the traditional average model of the PWM, while the red line illustrates the closed-loop response of the system including the accurate z-domain model. It can be seen that the closed-loop response is flat throughout the audio band in both cases. At frequencies higher than 40 kHz the models begin to differ. The phase responses of the two different models remain the same up until 60 kHz, as shown in Figure 3.23b. The closed-loop magnitude and phase responses indicate that the system might become unstable if the input reference has a frequency of more than 60 kHz. This is acceptable due to the fact that the audio band is only up until 20 kHz. The desired closed-loop response is clearly achieved, as the magnitude response is flat throughout the audio band.



(a) Closed-loop magnitude response.



(b) Closed-loop phase response.

Figure 3.23: Closed-loop magnitude and phase response of system.

3.8 System Stabilisation

As discussed in Section 2.6, integrator wind-up leads to over modulation of the amplifier resulting in a collapse of the loop gain. As shown in Figure 3.19, the system is conditionally stable. This means that the amplifier will become unstable should the system enter over modulation. It is therefore necessary to implement the estimation filter discussed in

Section 2.6. The filter must approximate the output stage in the audio band. A low-pass filter with a cut-off frequency of 100 kHz will therefore be implemented. The transfer function is

$$F'(s) = \frac{3.9478 \cdot 10^{11}}{s^2 + 9.348 \cdot 10^5 s + 3.948 \cdot 10^{11}}. \tag{3.38}$$

The reference input signal will also be added to the controller output as in Figure 2.15b.

3.9 Conclusion

In this chapter the design and modelling of the system was discussed. A full system model was presented and every section of the controller was discussed and designed. Bode, along with root-locus plots were used to determine that the system is stable and meets the fundamental requirement of having an open-loop gain of 50 dB or more across the audio band. The noise-transfer function was determined and confirmed that the system will adequately attenuate the quantization noise. An equation for the PWM input gradient was also determined making it possible to calculate the small-signal gain K_{ss} of the comparator. An estimation filter is implemented to counter the effects of integrator wind-up. The next chapter will discuss the simulation of the system.

Chapter 4

Simulation

4.1 Introduction

In this chapter the simulation process and results will be discussed. The first step will be to set up a simulation environment modelling the FPGA environment in which the system will be implemented. The system's closed-loop output, along with the FFT thereof, will be analysed with and without ripple compensation. The influence ripple compensation has on the small-signal gain of the comparator will also be investigated. The PWM input signal will also be measured and compared to that determined in Section 3.7. The stability of the system will be analysed using a root-locus diagram as well as a bifurcation diagram. It will then be possible to assess the system's overall performance and determine whether the initial system requirements are met.

4.2 Simulation Setup

The simulation is done using Simulink. It is important to note that the simulation must emulate the FPGA environment in which the controller will be implemented. The simulation is therefore clocked at two different clock frequencies. The PWM is clocked at 98.304 MHz, which is the clock speed of the FPGA. The controller is clocked at 19.6608 MHz, which is also the sampling rate of the ADC. A fixed-step size is used to ensure that the FFT of the output can be determined accurately. The sawtooth carrier frequency is equal to the switching frequency of the system. The simulation diagram is shown in Figure 4.1, where the clock frequency of every section is indicated. The sawtooth carrier is simulated by a 7-bit counter to obtain the required 768 kHz signal. The signal is scaled to provide values between -1 and 1. The input signal is selected as a 1 kHz sinusoid with an amplitude of 0.8. The small-signal gain is measured in the simulation by obtaining the derivative of the PWM input signal. The signal is then passed through

a sample-and-hold block which is triggered on the falling edge of the PWM output. The comparator output is a value between 0 and 1 which is scaled to provide a value between -1 and 1.

Since the output signal of the controller is sampled at 19.6608 MHz and the PWM operates at 98.304 MHz, a simple averaging filter is added to the simulation to ensure a smooth PWM input signal. The filter is depicted by $M'(s)$ in Figure 4.1 and is implemented by simply determining the average value of the last 8 input samples. This will prohibit sudden jumps in the PWM input signal. The loop is also rearranged to implement the anti-windup scheme proposed in Section 2.6. The estimation filter is depicted by $F'(s)$ and the input reference signal is added to the controller output.

The power stage is modelled as a simple gain with a value of half the DC-bus voltage (15 V) since a half-bridge configuration is used. The time delay block represents the total propagation delay of the system. The simulation model of the ADC was obtained from [20]. The model is essentially a noise shaper with a loop function which shapes the noise power spectrum to make it equivalent to the noise profile in the device's datasheet. Dither noise is also added to match the total noise power of the simulation to that of the practical ADC. Finally, it is important to remember that the ADC has a pipeline delay. This delay is added to the time delay block which will therefore consist of the delay caused by the FPGA calculations, the output stage delay and the pipeline delay of the ADC. This amounts to a total propagation delay of 650 ns. The noise profile of the ADC is discussed in depth in Section 5.6. The system is initially simulated without ripple compensation. The system is then simulated for a second time, with ripple compensation included, and the effect thereof is then discussed.

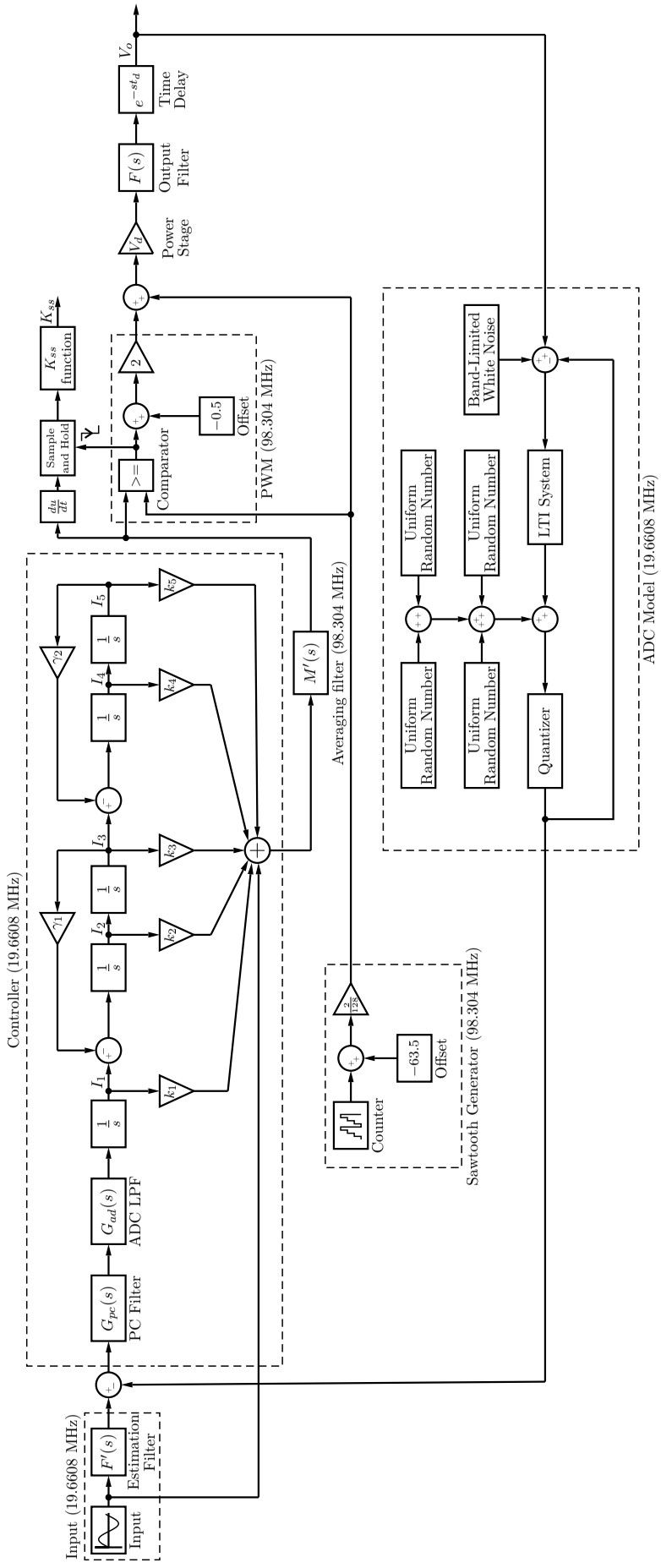


Figure 4.1: Simulation diagram.

4.3 Simulation Results

The first simulation is done without ripple compensation. The system was initially unstable without ripple compensation. The gain of the system was slightly lowered($-1.8\,\mathrm{dB}$) and the propagation delay was shortened to 500 ns to obtain stability. It is therefore clear that the design will not be stable without ripple compensation when practically implemented. The simulation without ripple compensation and new parameters is, however, still of value to analyse the effect ripple compensation.

Figure 4.2 shows the output signal of the system. It is clear that the system tracks the reference signal accurately. The high-frequency switching ripple is clearly visible. The FFT, seen in Figure 4.3, provides more information on the system's performance. All the FFTs calculated in this section makes use of a Hanning window during calculation. The FFT will have 1024 samples. The fundamental 1 kHz component of the sinusoid reference signal is clearly visible along with a large second harmonic component. The open-loop poles of the system are clearly visible at 10.6 kHz and 17.8 kHz. It can also be seen that the noise is adequately attenuated in the audio band and the amplitude of the noise only starts to increase at frequencies above 20 kHz.

Before analysing the PWM input signal, it worth investigating the effect that the averaging filter $M'(s)'$ has on the PWM input signal. As previously stated in this section, the controller is clocked at 19.6608 MHz and the PWM at 98.304 MHz. An averaging filter, which determines the mean value of the last 8 samples, is used to rectify the sudden changes in amplitude caused by the lower sampling time of the controller output.

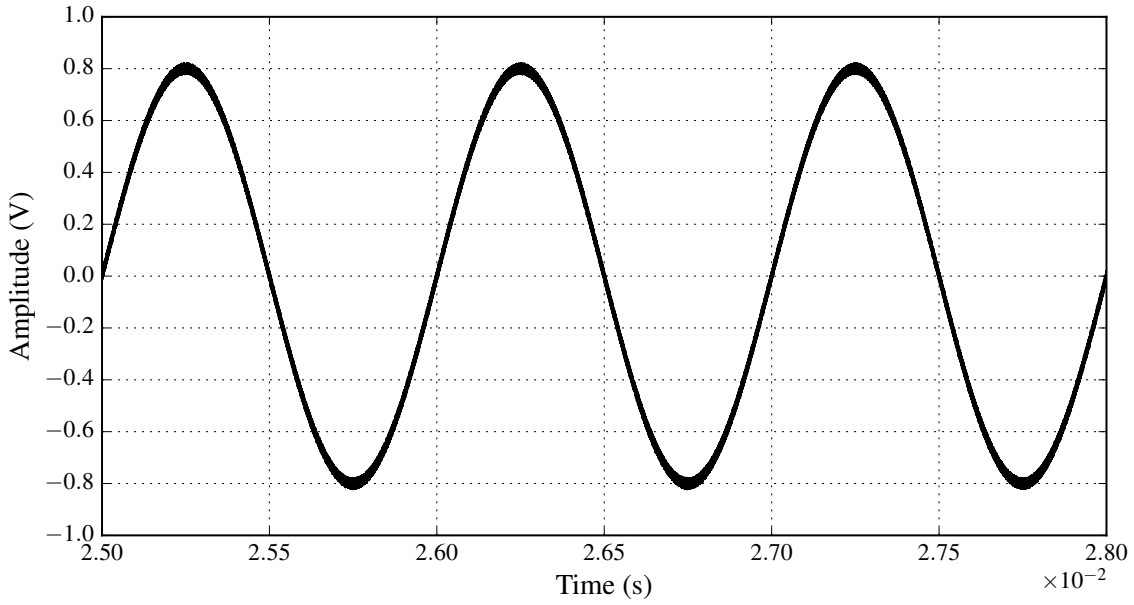


Figure 4.2: Output signal without ripple compensation (1 kHz).

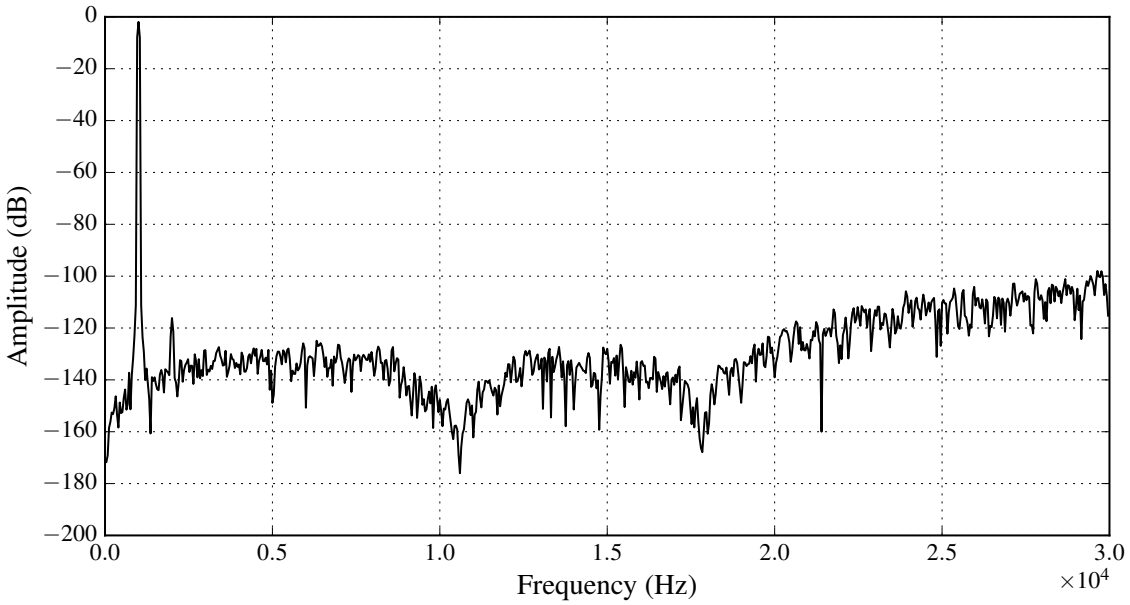


Figure 4.3: FFT of output signal without ripple compensation (1 kHz).

Figure 4.4 shows a typical controller output signal in blue, and the averaging filter output in black. It is evident that the filtered signal will provide a much more accurate PWM input signal. The filter does, however, contribute a time delay to the system as can be seen in the phase difference of the two signals shown in Figure 4.4. The PWM input signal, along with the sawtooth carrier, is shown in Figure 4.5a. As previously stated,

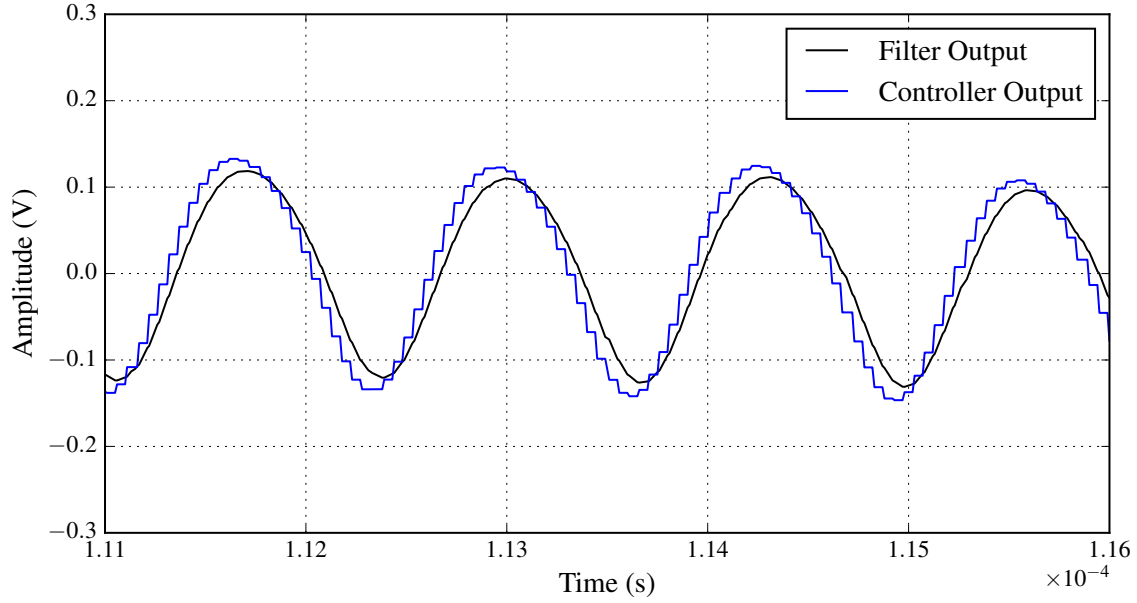
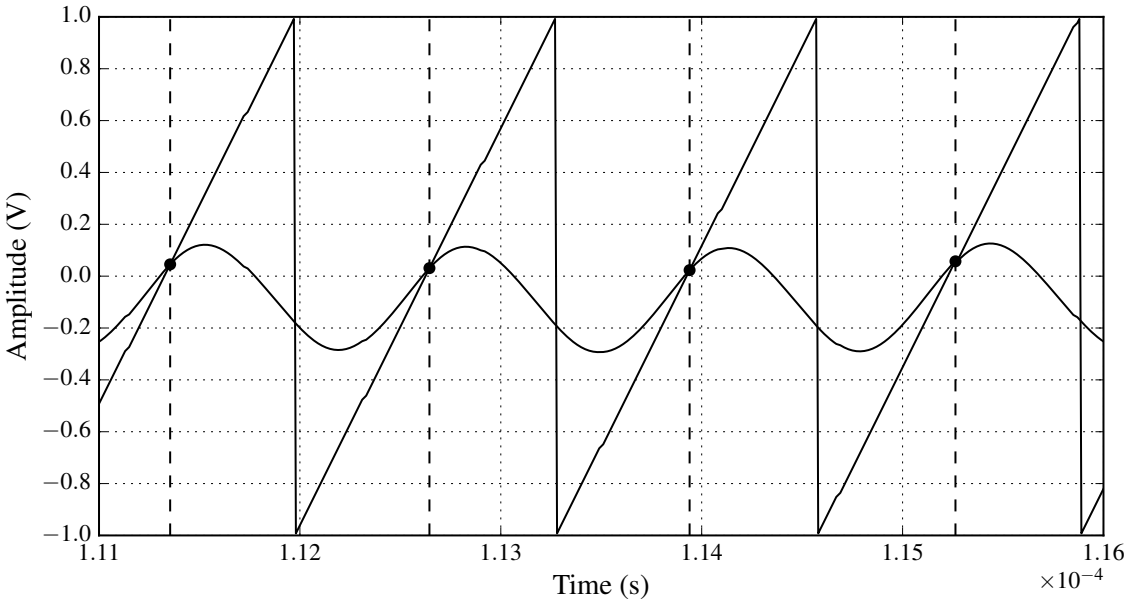
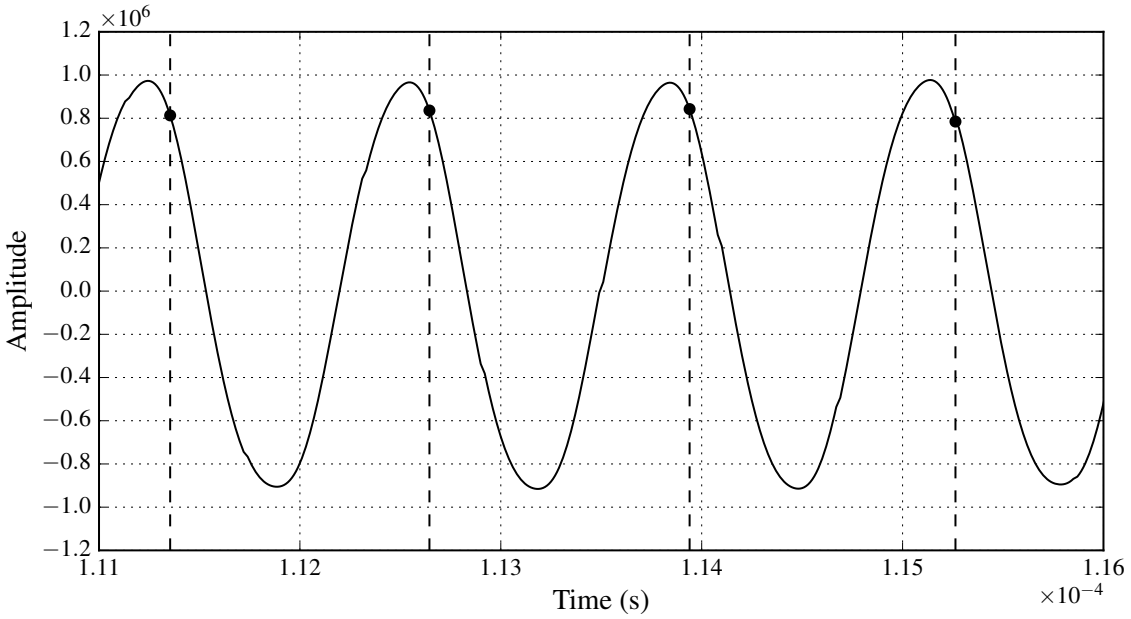


Figure 4.4: Input and output of averaging filter.



(a) PWM input signal and sawtooth carrier.



(b) PWM input signal gradient.

Figure 4.5: Simulated PWM input signal and PWM gradient waveforms without ripple compensation.

the comparator small-signal gain is dependent on the gradient of the PWM input signal at the transition points of the sawtooth carrier and the PWM input signal. Figure 4.5b shows the gradient of the PWM input signal with the transition points marked. During simulation, a sample-and-hold procedure is used to measure the value of the gradient at the transition point, and keep that value until the next transition point. Even though it

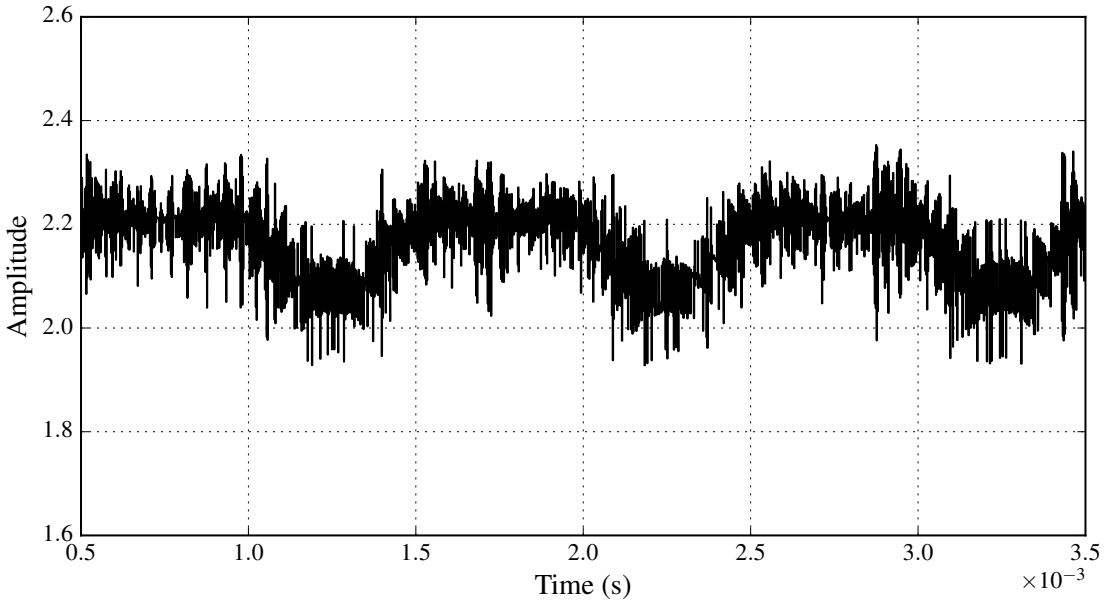


Figure 4.6: Small-signal gain without Ripple Compensation.

seems as if the value of the measured gradient almost remains constant, it is important to note that Figure 4.5b shows a very small section in time to illustrate the measuring procedure.

When the sample-and-hold procedure of the PWM input gradient is executed over a longer time period and the small-signal gain for those values is calculated, the signal in Figure 4.6 is obtained. It is clear that the small-signal gain K_{ss} is dependent on the sinusoidal reference signal. This makes it impossible to accurately compensate for K_{ss} by adding a constant gain term because K_{ss} is constantly varying. It is clear from Figure 4.6 that the total gain of the system can vary by a factor of up to 1.25 depending on the sinusoidal reference signal. This is the reason as to why the system is unstable without ripple compensation, as will be seen later in the section when the bifurcation diagram of the initial design is analysed.

The second simulation is done with ripple compensation implemented. The gain and the propagation delay is reset to the values which were initially used during design. Figure 4.7 shows the output signal, where it is once again clear that the reference signal is accurately tracked. It can be seen that the high-frequency switching ripple has a slightly smaller amplitude than the output signal in Figure 4.2.

The FFT of the output signal shown in Figure 4.8 illustrates that the noise is adequately attenuated over the audio band and the two open-loop poles at 10.6 kHz and 17.8 kHz are once again visible. It is, however, important to notice that the harmonics of the fundamental 1 kHz sinusoidal component are not visible. This indicates that ripple

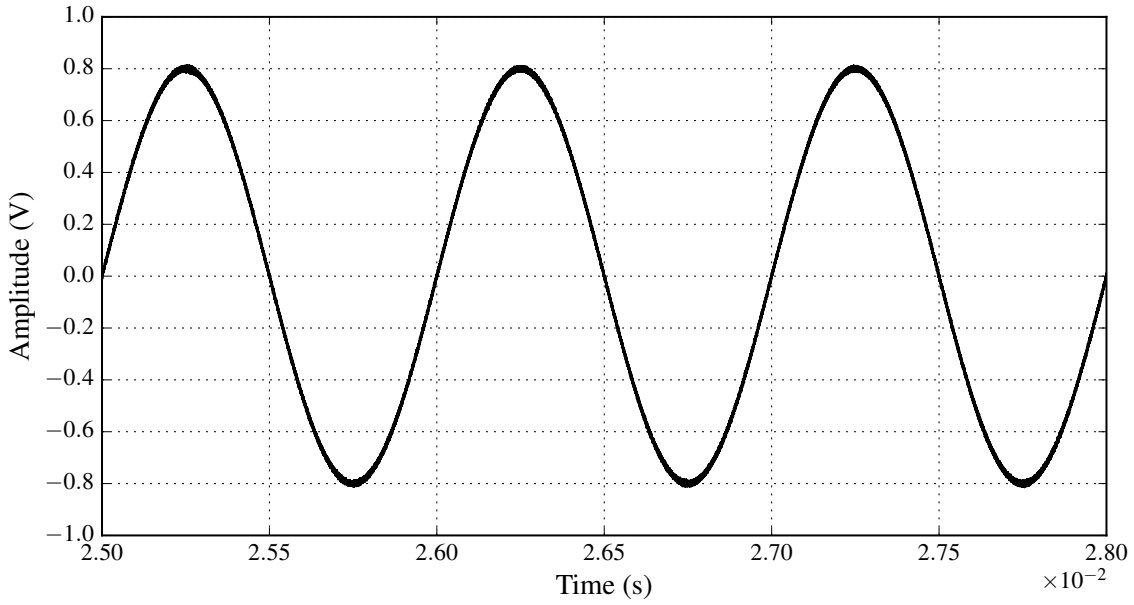


Figure 4.7: Output signal with Ripple Compensation (1 kHz).

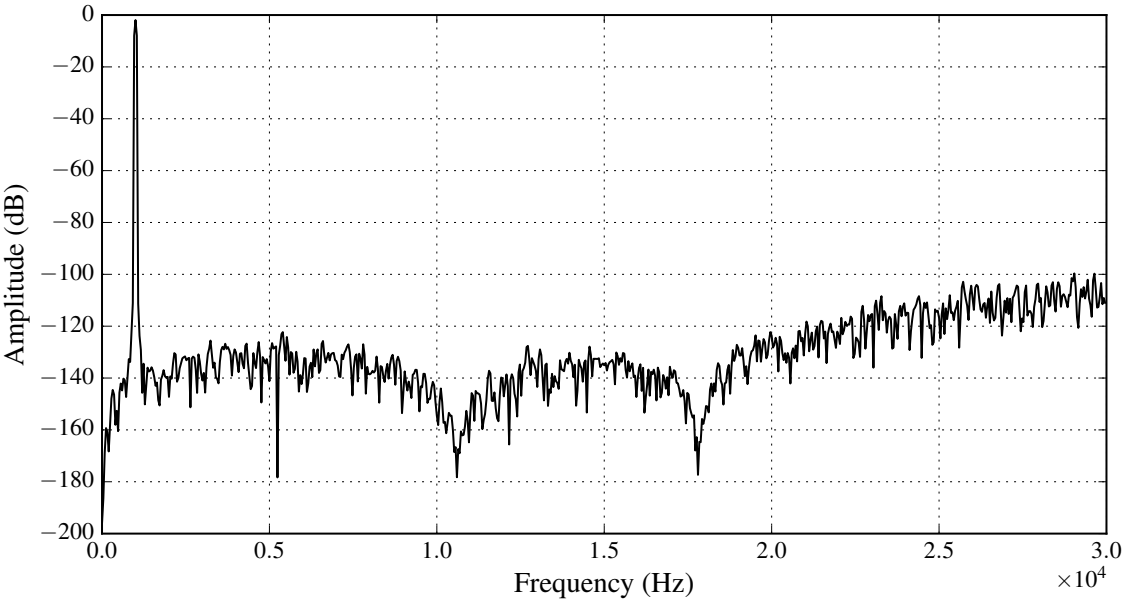


Figure 4.8: FFT of Output signal without Ripple Compensation (1 kHz).

compensation decreases the harmonic distortion in the system. It is important to remember that the noise floor seen when calculating the FFT does not necessarily coincide with that of the practical system. This is due to the fact that the amount of samples in the FFT has an influence on the noise floor amplitude of the FFT. The FFT does, however, still provide a good estimate of the performance of the system in terms of noise attenuation.

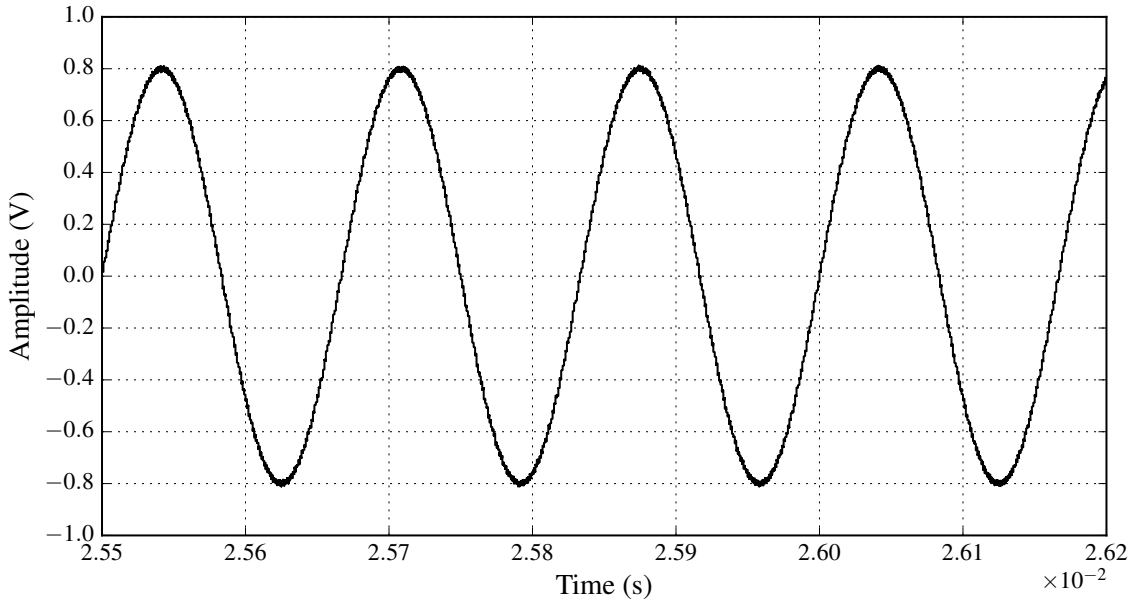


Figure 4.9: Output signal with Ripple Compensation (6 kHz).

To further verify the correct operation of the system, another simulation is done and the frequency of the reference signal is altered. Figure 4.9 and 4.10 illustrate the waveforms produced using a reference signal with a frequency of 6 kHz (amplitude remains 0.8). It is once again seen that the system tracks the reference signal accurately and the quantization noise, including harmonics, is adequately attenuated. Figure 4.11 illustrates

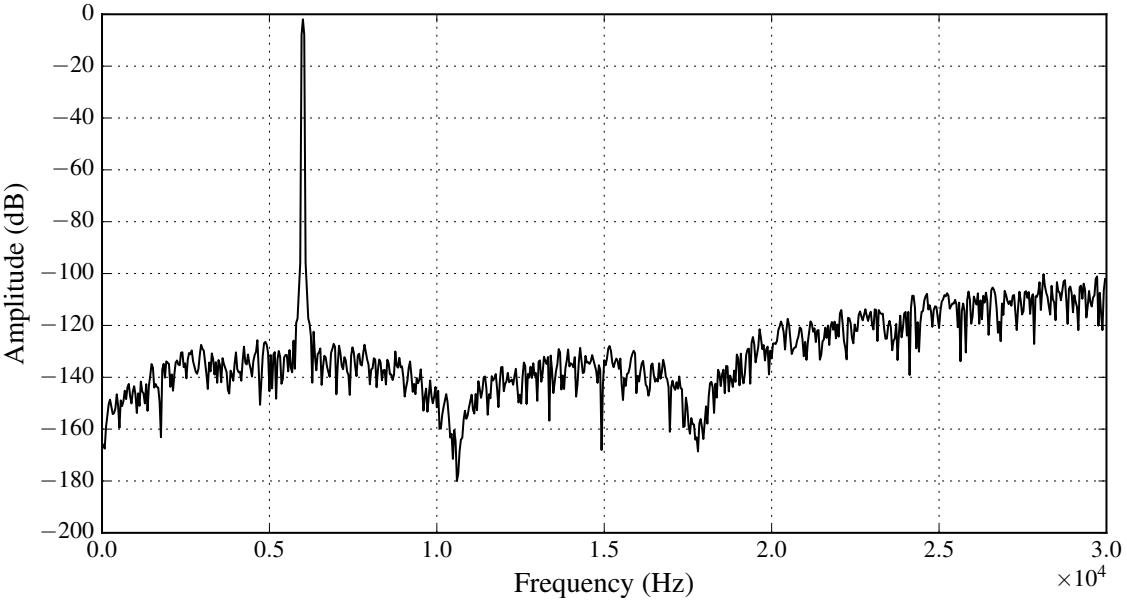


Figure 4.10: FFT of Output signal with Ripple Compensation (6 kHz).

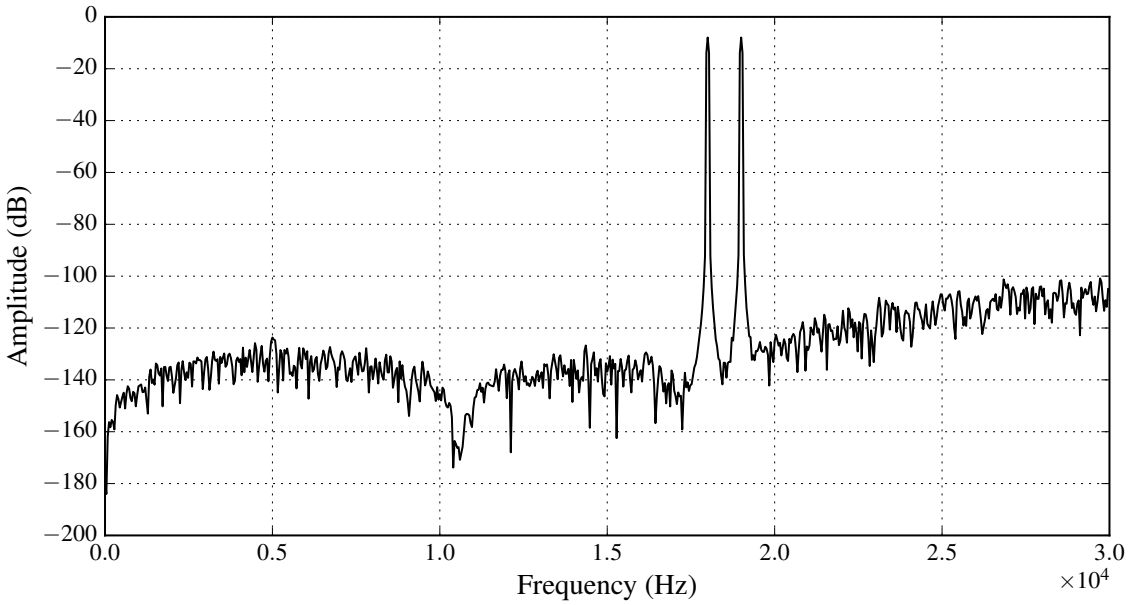
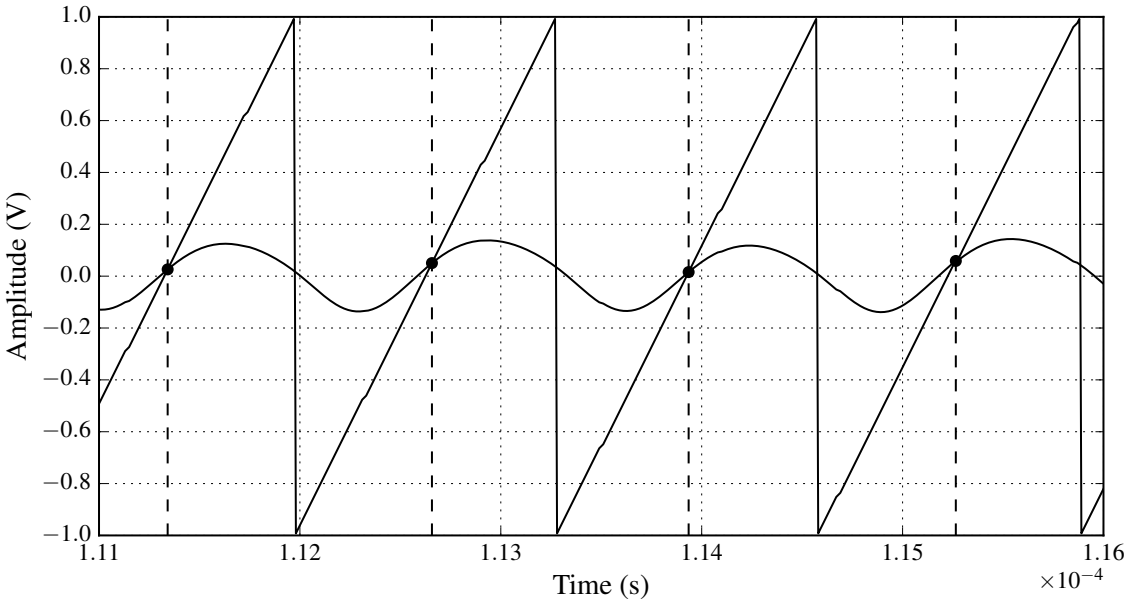


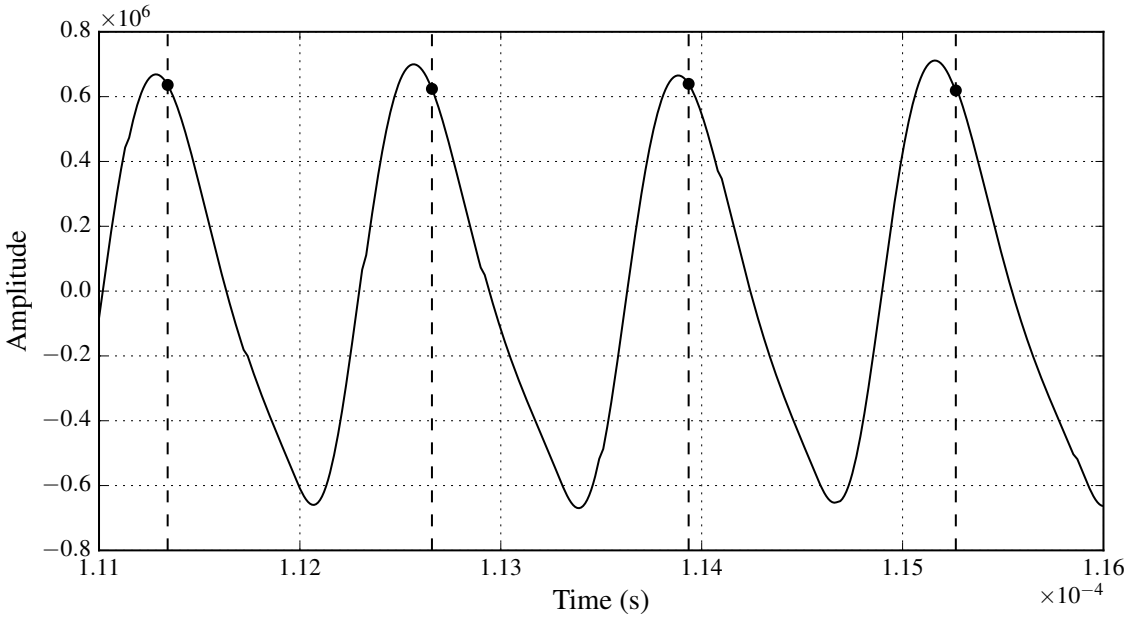
Figure 4.11: FFT of output signal with two-tone input (18 kHz and 19 kHz).

the FFT of the output signal with a two-tone input consisting of 18 kHz and 19 kHz components.

Figure 4.12a once again shows the PWM input signal along with the sawtooth carrier signal. The gradient of the PWM input signal, with the transition points with the sawtooth carrier marked, is seen in Figure 4.12b. It is clear that the form of the signal is very close to that of the theoretically calculated one seen in Figure 3.22. As with the previous simulation without ripple compensation, it seems as if the value of the gradient at the sample points remains constant. When ripple compensation is implemented, this is much closer to the truth as can be seen in Figure 4.12a, where the sample-and-hold operation is once again executed over a longer time period. It is clear that the small-signal gain K_{ss} is independent of the reference signal as the variation of K_{ss} is minimal. This makes it possible to accurately compensate for K_{ss}, which has an average value of 1.6770, by adding a gain of 0.5963 (the inverse of K_{ss}) to the system and therefore improving stability margins and performance. This value is slightly different, albeit very close, from the one calculated in Section 3.7 (1.6922). This is due to the fact that the theoretical calculation does not account for the quantization noise in the system. Should the system be simulated at higher frequencies, the theoretical K_{ss} and simulated K_{ss} will converge as the quantization noise decreases with an increase in simulation frequency.



(a) PWM input signal and sawtooth carrier.



(b) PWM input signal gradient.

Figure 4.12: Simulated PWM input signal and PWM gradient waveforms with ripple compensation.

The next step is to use a bifurcation diagram to determine whether the z-domain design, which was done in Chapter 3, accurately predicts the stability margins of the system. Figure 4.14 shows the bifurcation diagram of the system. The modulator output is plotted as a function of K_a, which is the additional gain added to the system. It is clear that the modulator output starts to bifurcate when K_a has a value of 1.235. This

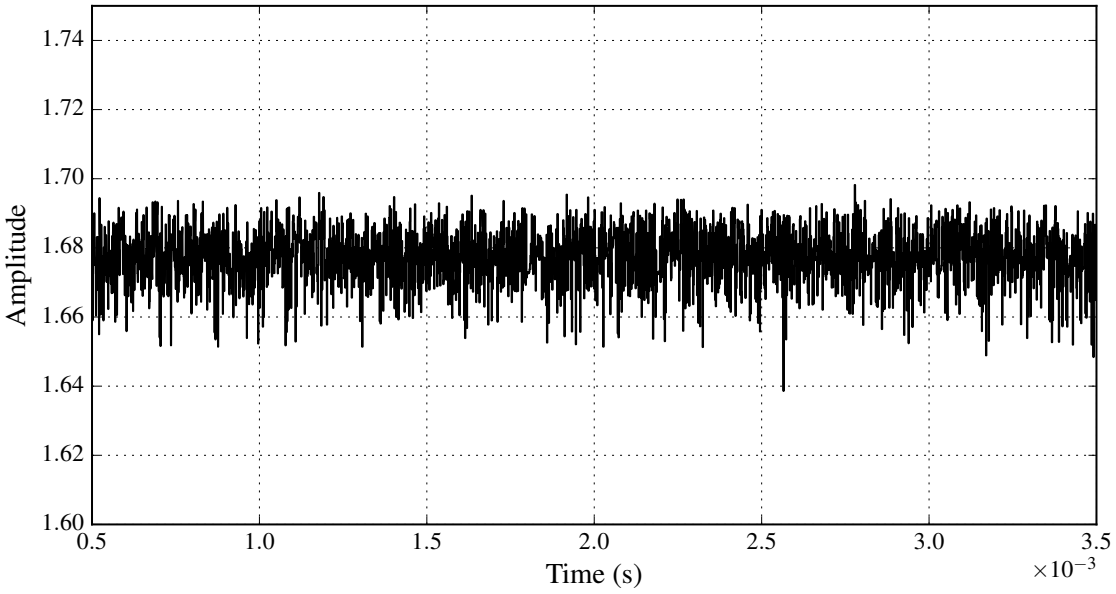


Figure 4.13: Small-signal gain with Ripple Compensation.

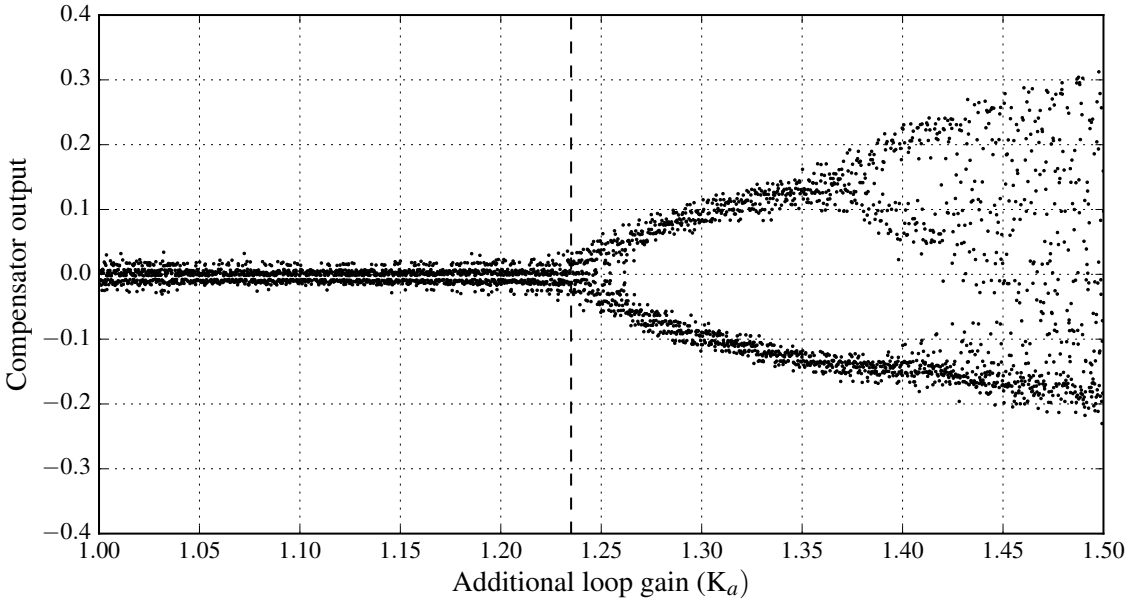


Figure 4.14: Bifurcation Diagram.

confirms that the system will not be stable without ripple compensation as it was seen in Figure 4.6 that the small-signal gain can vary with a factor of up to 1.25.

Figure 4.15 shows the normalised actual loop gain as a function of K_a. The actual loop gain is defined as the product of K_a and the small-signal gain K_{ss}. The gain margin of the system was determined in Chapter 3 as 3.01 dB. The critical value of the gain margin is thus 1.414 and can be seen in Figure 4.15. It is clear that the actual loop gain intersects

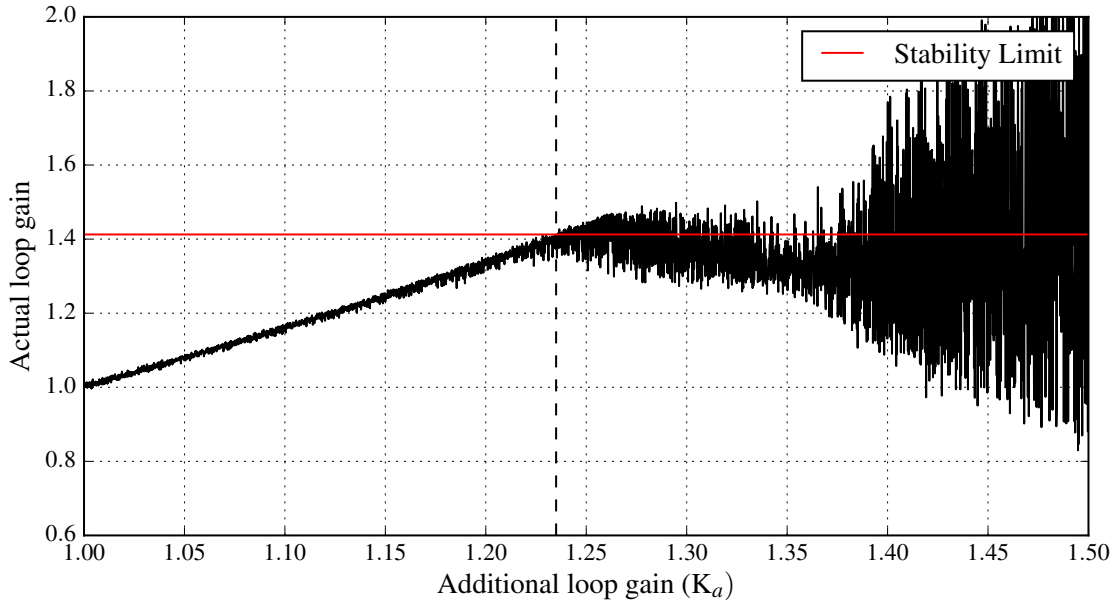


Figure 4.15: Actual loop gain as function of K_a.

the stability limit very close to the bifurcation value of K_a, which was determined in the bifurcation diagram. This confirms that the z-domain design accurately predicts the stability margin of the system.

4.4 Conclusion

In this section the system was successfully simulated. The simulation method and setup was discussed and it was confirmed that once ripple compensation is implemented, the system is in fact stable and will accurately track the reference signal. It was also seen in the FFT of the output signal that the system will adequately attenuate the noise in the audio band. It was observed that ripple compensation improves the performance of the system and makes the small-signal gain independent of the reference signal. This makes it possible to accurately compensate therefore by adding a constant gain to the system. The bifurcation point of the system was determined and it was found that the root-locus design in Chapter 3 predicts the stability margins of the system with high accuracy. The next step will be to practically implement the system.

Chapter 5

Practical Implementation

5.1 Introduction

The following section will cover the practical implementation of the system. The PWM evaluation board developed in [4] is used to test the control strategy. Along with an FPGA, the board contains several other pieces of hardware needed to implement the amplifier. This includes a low-jitter multi-output clock distributor, a sigma-delta ADC to digitise the output signal and an asynchronous sample rate converter with a digital audio interface. A separate output stage is used to amplify the PWM output signal received from the evaluation board. The hardware setup will be discussed, along with the basic logic which is used to implement the controller in an FPGA. Block diagrams will be used to illustrate the flow of the system. The integration of the different components' clock speeds with the system and the timing requirements will also be discussed.

5.2 Clock Distribution and System Flow

The clock distribution is done by an Analog Devices AD9512 IC. This is a multi-output clock distributor which provides low-jitter clock outputs and minimises phase noise therefore increasing data converter performance [21]. This is specifically important in this application as the system performance is dependant on the performance of the sigma-delta ADC [4]. The AD9512 will provide a 19.6608 MHz clock which will be used as the main clock for the ADC. A 98.304 MHz clock is also provided which is used as the FPGA clock and the external reclocking of the PWM output.

Figure 5.1 provides a detailed illustration of the system configuration. Dashed lines are used to show clock signals, while solid lines are used to depict data signals. The digital audio source provides the sample rate converter with a 24-bit audio signal. The signal then enters the FPGA where it is upsampled to produce the reference signal $R(z)$ which is a

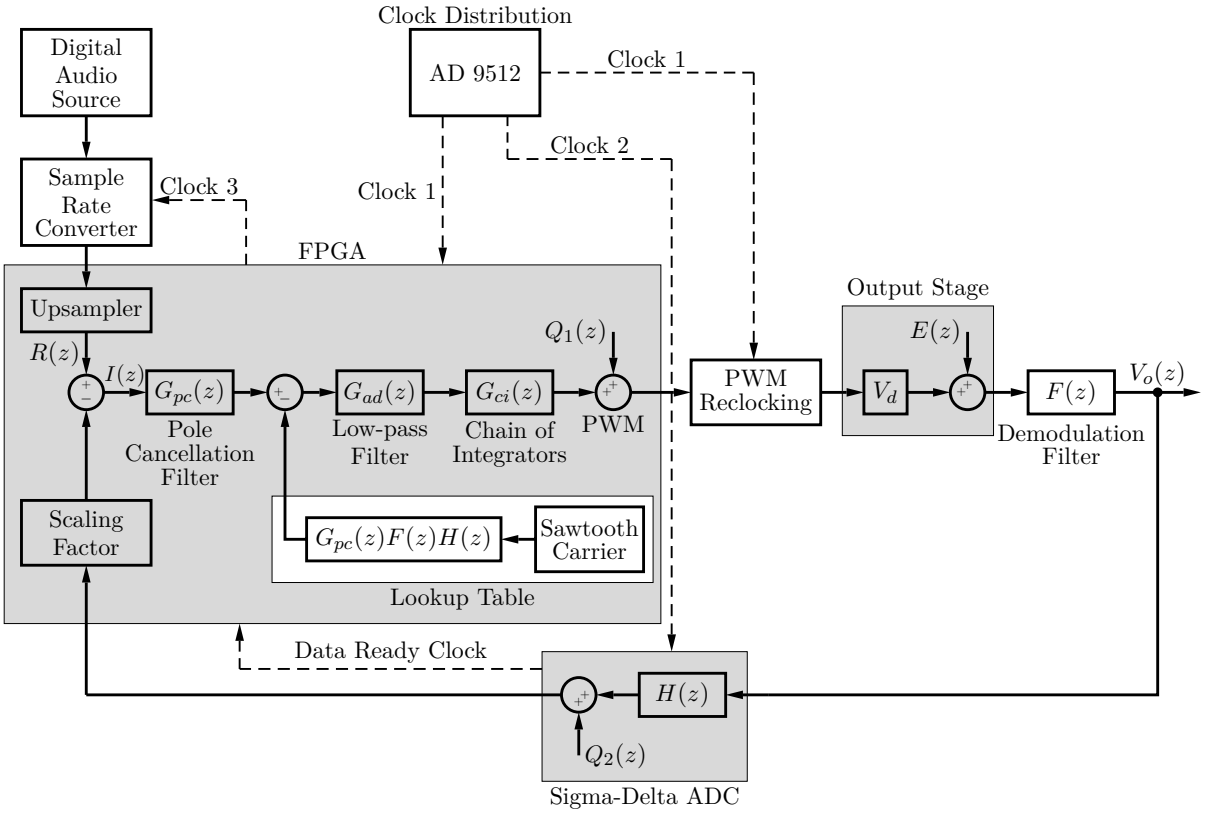


Figure 5.1: Practical implementation of system.

24-bit signal sampled at 19.6608 MHz. The output feedback is then subtracted to produce the error signal $I(z)$, which is then passed through $G_{pc}(z)$. The ripple compensation value for the specific sample is then obtained from a lookup table and subtracted before the signal is passed through $G_{ad}(z)$ and $G_{ci}(z)$. The quantization noise added by the 7-bit PWM is depicted by $Q_1(z)$. The PWM output which is provided by the FPGA is then reclocked by external flip-flops and sent to the output stage where V_d has a value of half the DC-bus voltage. The gain V_d is equal to the DC-voltage bus when a full-bridge output stage is used. The error signal $E(z)$ represents any imperfections the output stage might have. The output $V_o(z)$ is then obtained by passing the switched signal from the output stage through a demodulation filter. The output signal $V_o(z)$ is then digitised by the ADC before being sent to the FPGA where it is scaled to reflect the correct output signal relative to the reference signal $R(z)$. Figure 5.2 shows a photo of the development board used to implement the system. The output stage, which includes the demodulation filter, is shown in Figure 5.3.

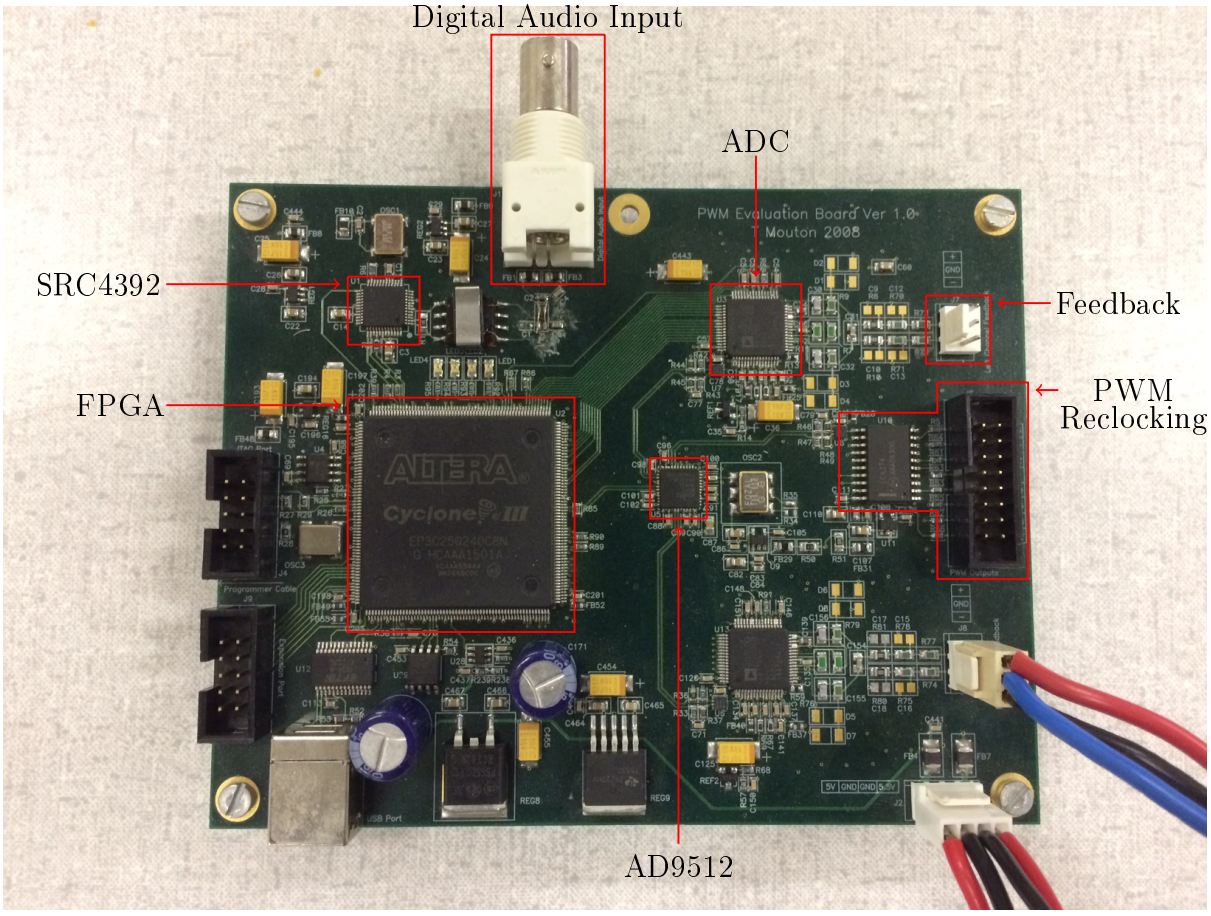


Figure 5.2: Development board used to implement the system.

5.3 Asynchronous Sample Rate Converter

In order for the system to be able to interpret a digital audio input signal, the system features an asynchronous sample rate converter with a digital audio interface. The Texas Instruments SRC4392 IC is used in this case. The device will upsample the 24-bit audio input to 196.6 kHz. A 24.575 MHz clock from the FPGA is used to operate the device (Clock 3 in Figure 5.1).

5.4 FPGA

The FPGA used to implement the control system and PWM is an Altera EP3C25Q240C8 Cyclone III. The device has 24624 logic elements, 608 Kb memory, 149 I/O pins and will be configured using a JTAG interface [22]. The code for the device will be written in VHDL. Quartus software is used to compile and load the code onto the FPGA. The fixed-point package is used to store variables.

The FPGA will operate with a 98.304 MHz low-jitter clock which is received from the

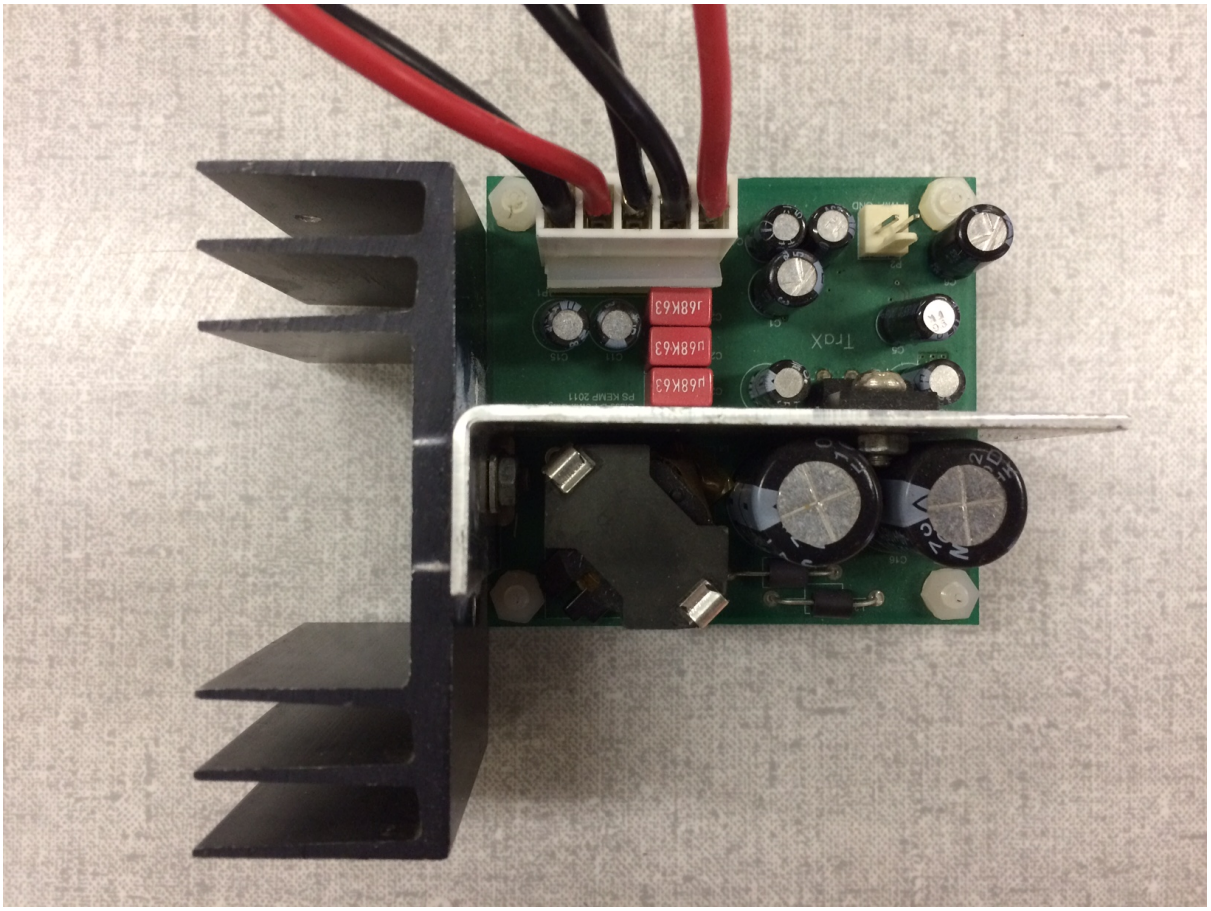


Figure 5.3: Output stage.

AD9512 clock distributor (Clock 1 in Figure 5.1). The digital feedback loop operates at a clock speed of 19.6608 MHz. The clock is received from the data ready pin of the ADC (Data Ready Clock in Figure 5.1). The pulse-width modulator will, which has 7 bits, operates at 98.304 MHz. The PWM input signal will therefore be updated every fifth cycle of the 98.304 MHz clock. In [4] it was found that this specific clock ratio of 5:1 does not cause measurable distortion of the output signal in the audio band. This is, however, subject to adequate low-pass filtering of the ADC input. It was found that the FPGA contributed approximately 200 ns to the propagation delay of the system due to the internal calculations. The FPGA uses a phase-locked loop (PLL) to obtain a 1.024 MHz which is used to configure the clock distribution IC using SPI communication. A second PLL is used to obtain a 24.575 MHz clock which is used to operate the SRC4392 IC.

5.4.1 Upsampler

The reference input received from the sample rate converter is sampled at 196.61 kHz. As previously mentioned, the loop filter operates at 19.6608 MHz. An upsampler is therefore required to increase the sampling frequency of the reference input.

The first step of the upsampling is to zero pad the signal with $N-1$ zeros between each sample. The signal is then interpolated by passing it through a low-pass filter. A low-pass filter is also required to remove the aliased high-frequency components in the frequency band of interest, which are present due to upsampling, and interpolate the signal. A basic example of upsampling is illustrated in Figure 5.4 to clarify the concept. Figure 5.4a shows a simple 1 kHz sinusoid sampled at 10 kHz. To increase the sampling by a factor $N = 3$, thus to a frequency of 30 kHz, two zeros are added between every sample as seen in Figure 5.4b.

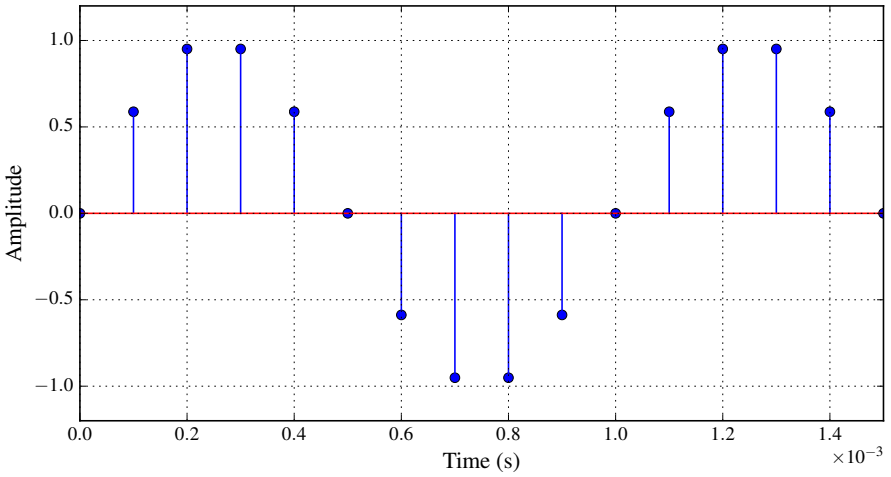
When viewing the signal in the frequency domain, upsampling essentially contracts the frequency axis by a factor N [23]. As the signal has a bandwidth of 15 kHz, the cut-off frequency of the low-pass filter will be chosen as (7.5 kHz) divided by N, thus having a value of 2.5 kHz. The output of the filter (the interpolated signal) is seen in Figure 5.4c.

This project requires the input signal to be upsampled with a factor of $N = 100$. The low-pass filter is chosen as a sixth order Butterworth filter to ensure a high quality interpolated signal. Since the signal has a bandwidth of 9.8304 MHz, the cut-off frequency will be 49.152 kHz. This will ensure the unwanted frequency components present due to upsampling are adequately attenuated. The interpolation low-pass filter has a transfer function of

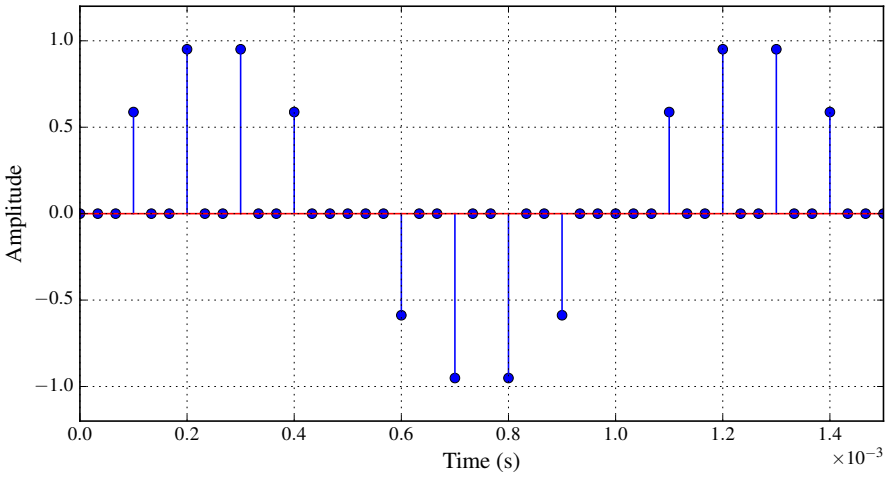
$$G_{int}(z) = \frac{c_1 z^6 + c_2 z^5 + c_3 z^4 + c_4 z^3 + c_5 z^2 + c_6 z + c_7}{z^6 - 5.939 z^5 + 14.7 z^4 - 19.4 z^3 + 14.4 z^2 - 5.704 z + 0.9411}, \tag{5.1}$$

where

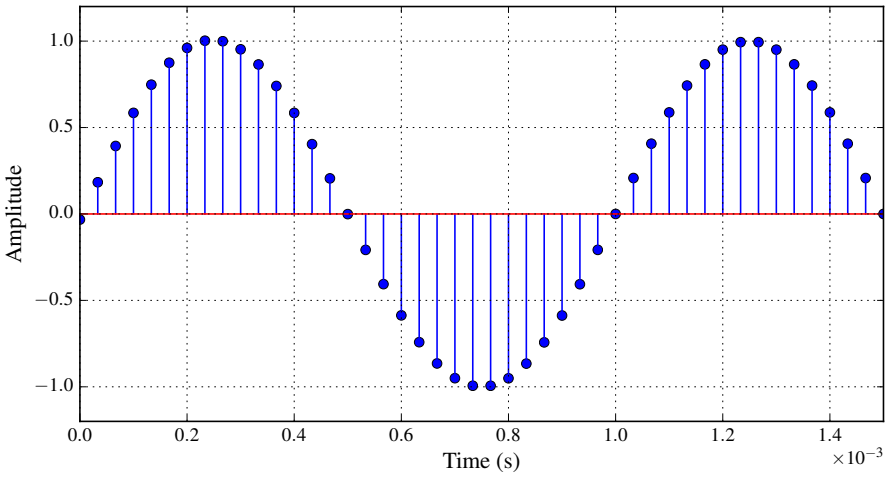
$$\begin{aligned} c_1 &= 2.277 \cdot 10^{-13}, \\ c_2 &= 1.366 \cdot 10^{-12}, \\ c_3 &= 3.416 \cdot 10^{-12}, \\ c_4 &= 4.554 \cdot 10^{-12}, \\ c_5 &= 3.416 \cdot 10^{-12}, \\ c_6 &= 1.366 \cdot 10^{-12}, \\ c_7 &= 2.277 \cdot 10^{-13}. \end{aligned} \tag{5.2}$$



(a) Sampled sinusoidal signal.



(b) Zero padded signal.



(c) Upsampled signal passed through interpolation filter.

Figure 5.4: Basic Upsampling Process.

5.4.2 Digital Filters

The system contains three digital filters. A pole cancellation filter, a low-pass filter and an averaging filter. The averaging filter $M'(s)$ is simply implemented by adding the last 8 samples together and using a bit-shift operation to determine the average, which is used as the filter output. Both the pole cancellation filter $G_{pc}(s)$ and the low-pass filter $G_{ad}(s)$, which were designed in Chapter 3, are digitally implemented as IIR filters. The continuous-time domain filters will be transformed to the z-domain at the same sampling frequency as that at which the feedback loop operates (19.6608 MHz). The filters are implemented in the following form:

$$\frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{5.3}$$

When implemented digitally, the output is determined using

$$y(k) = a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) - b_1 y(k-1) - b_2 y(k-2), \tag{5.4}$$

where $x(k)$ and $y(k)$ are the current input and output samples respectively. The previous input and output samples are denoted by $x(k-1)$ and $y(k-1)$ with the notation following the same trend with samples further back in time. The filter coefficients are shown in Table 5.1.

5.4.3 Chain of Integrators

In order to implement the chain of integrator in the FPGA, it is first necessary to determine the discrete-time domain transfer function of the chain of integrators seen in Figure 3.10. This will then be used to determine the γ_{zi} values in the resonator feedback loops (which determine the locations of the poles) and the feedforward coefficients (which determine the gain and the location of the zeros). The transfer function is depicted by

$$\frac{Y(z)}{U(z)} = \frac{k_{z1}a_{z1}a_{z2} + k_{z2}(z-1)a_{z2} + k_{z3}a_{z2} + k_{z4}(z-1) + k_{z5}}{(z-1)a_{z1}a_{z2}}, \tag{5.5}$$

where

$$a_{z1} = (z-1)^2 + \gamma_{z1},$$
$$a_{z2} = (z-1)^2 + \gamma_{z2}.$$

Table 5.1: Discrete-time domain Filter coefficients.

Filter	a_0	a_1	a_2	b_1	b_2
$G_{pc}(z)$	6295	$-1.177 \cdot 10^4$	5477	-1.097	0.3008
$G_{ad}(z)$	0.001433	0.08216	0.03329	-1.639	0.7559

The next step is to calculate the required transfer function, which is done by determining the z-transform of $G_{ci}(s)$ (calculated in Chapter 3) with a sampling frequency of 19.6608 MHz. This will produce $G_{ci}(z)$ where the numerator is depicted by

$$N_{ci}(z) = c_{z1}z^4 + c_{z2}z^3 + c_{z2}z^2 + c_{z3}z + c_{z4}. \tag{5.6}$$

The z-domain coefficients have the following values:

$$\begin{aligned} c_{z1} &= 0.004742, \\ c_{z2} &= -0.01879, \\ c_{z3} &= 0.02793, \\ c_{z4} &= -0.01845, \\ c_{z5} &= 0.00457. \end{aligned}$$

The numerator of (5.5) can then be equated to (5.6), which yields the following relationships to determine the feedforward gain values k_{zi}:

$$\begin{aligned} k_{z1} &= c_{z1}, \\ k_{z2} &= c_{z2} + 4k_{z1}, \\ k_{z3} &= c_{z3} - k_{z1}(6 + \gamma_{z1} + \gamma_{z2}) + 3k_{z2}, \\ k_{z4} &= c_{z4} - k_{z1}(-4 - 2\gamma_{z1} - 2\gamma_{z2}) - k_{z2}(3 + \gamma_{z2}) + 2k_{z3}, \\ k_{z5} &= c_{z5} - k_{z1}(1 + \gamma_{z1} + \gamma_{z2} + \gamma_{z1}\gamma_{z2}) - k_{z2}(-1 - \gamma_{z2}) - k_{z3}(1 + \gamma_{z2}) + k_{z4}. \end{aligned}$$

The feedforward coefficients have the following values:

$$\begin{aligned} k_{z1} &= 0.00474, \\ k_{z2} &= 1.7474 \cdot 10^{-4}, \\ k_{z3} &= 2.5850 \cdot 10^{-6}, \\ k_{z4} &= 1.5637 \cdot 10^{-8}, \\ k_{z5} &= -1.2909 \cdot 10^{-11}. \end{aligned} \tag{5.7}$$

The denominator of (5.5) is only dependant on the integrator and the positions of the two complex pole pairs. As discussed in Section 2.5, the poles are placed on the imaginary axis in the continuous-time domain. This means the poles will be placed on the unit circle in the discrete-time domain. The frequencies of the poles are determined by γ_{z1} and γ_{z2}. These values are obtained by squaring the imaginary part of the pole coordinate in the z-domain. The complex pole pairs at 10.6 kHz and 17.8 kHz translate to γ values of $1.1475 \cdot 10^{-5}$ and $3.2359 \cdot 10^{-5}$ respectively. Listing 5.1 contains VHDL code which can be used to implement the chain of integrators as shown in Figure 3.10.

Listing 5.1: VHDL implementation of Chain of Integrators

```
process
begin

wait until rising_edge(ADCClock);
if Reset = '0' then
ControlIn := LPFOut;

Integrator5 := resize(Integrator5+Integrator4,
                      Integrator5'high, Integrator5'low);

Integrator4 := resize(Integrator4+Integrator3-gamma2*Integrator5,
                      Integrator4'high, Integrator4'low);

Integrator3 := resize(Integrator3+Integrator2,
                      Integrator3'high, Integrator3'low);

Integrator2 := resize(Integrator2+Integrator1-gamma1*Integrator3,
                      Integrator2'high, Integrator2'low);

Integrator1 := resize(Integrator1+ControlIn,
                      Integrator1'high, Integrator1'low);

Sum := resize(Integrator1*kz1+Integrator2*kz2+Integrator3*kz3+
              Integrator4*kz4+Integrator5*kz5, Sum'high, Sum'low);

ControlOut <= resize(Sum, ControlOut'high, ControlOut'low);

end if;
end process;
```

The low-pass filter output `LPFOut` is assigned as the controller input `ControlIn`. The integrators are implemented as accumulators. `Integrator1` is simply an accumulation of the controller input signal. `Integrator2` accumulates `Integrator1` and `Integrator2` while receiving negative feedback in the form of `Integrator3` multiplied by `gamma1`. The same thought process is followed for the rest of the integrators. Each integrator output is then multiplied by its feedforward factor and summed together to form the controller output term `Sum` and resized to form the controller output `ControlOut`.

5.4.4 Pulse-width modulator

As previously stated, the 7-bit pulse-width modulator will operate at 98.304 MHz. To obtain a sawtooth carrier with a frequency of 768 kHz, a counter is implemented which increases up to 127 before resetting to 0. The PWM input is therefore scaled to a value between 0 and 127 before being compared to the sawtooth carrier. If the PWM input is bigger than the sawtooth carrier, the PWM output is set to 1, else the PWM output is set to 0. The PWM contains a flag which detects when the input intersects the sawtooth carrier. This enables the system to prohibit the PWM from switching more than once per switching period. The PWM output is reclocked internally, before being reclocked by flip-flops externally using the 98.304 MHz low-jitter clock obtained from the clock distributor (Clock 1 in Figure 5.1). This is done to ensure the accuracy of the PWM output before sending it to the output stage.

5.4.5 Ripple Compensation

Ripple compensation is implemented using the configuration seen in Figure 2.12b. In this specific system, the sawtooth carrier is pre-distorted by passing it through the transfer function

$$G_pre(z) = G_{pc}(z)F(z)H(z), \tag{5.8}$$

where $G_{pc}(z)$ is the pole cancellation filter, $H(z)$ the time delay of the ADC and $F(z)$ a discrete-time domain filter approximating the analogue demodulation filter. The transfer function of $F(z)$ is

$$F(z) = \frac{3.041 \cdot 10^{-5}z + 3.038 \cdot 10^{-5}}{z^2 - 1.997z + 0.997}. \tag{5.9}$$

It is obtained by determining the continuous-time domain function of the analogue filter and then transforming it to the discrete-time domain using a sample rate of 19.6608 MHz.

The biggest advantage of this method is that the sawtooth carrier signal is passed through (5.8) off-line using MATLAB. A memory initialization file (.mif) is then created containing the values of a distorted switching cycle. This file is then loaded onto the FPGA and used as a lookup table from which the ripple compensation values are obtained. Each value in the file has a unique address which is updated on every clock cycle to ensure that the correct value is used. This minimises calculation time in the FPGA as the value can simply be recalled from the lookup table before being subtracted from the pole cancellation filter output as seen in Figure 5.1. A simulated version of the pre-distorted sawtooth carrier used for ripple compensation is shown in Figure 5.5.

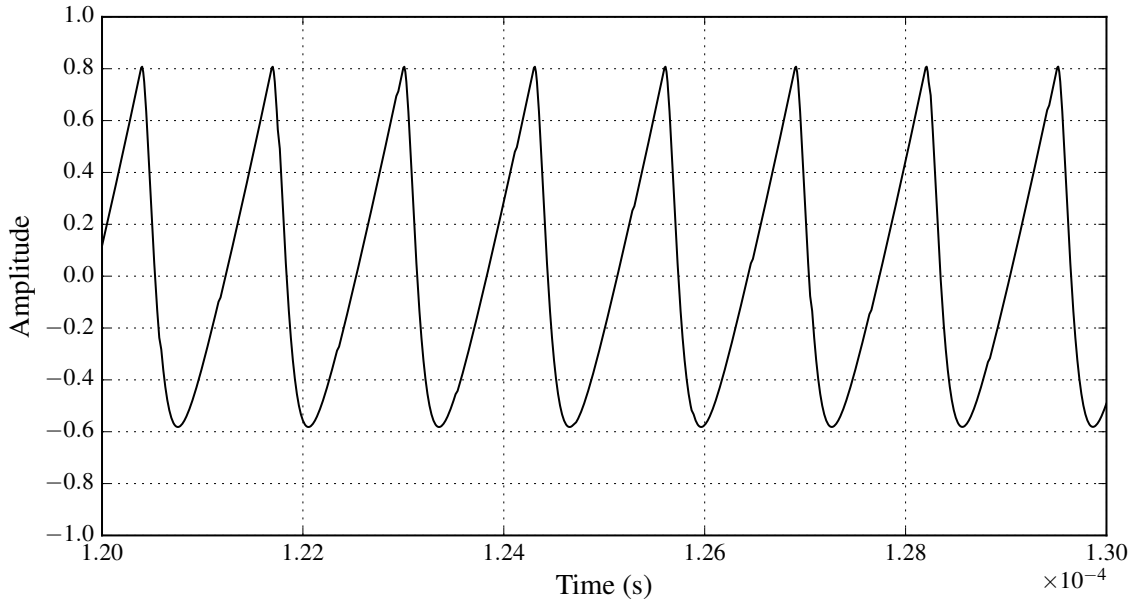


Figure 5.5: Pre-distorted sawtooth carrier used for ripple compensation.

5.5 Output Stage

The half-bridge output stage which was developed in [5] is used for this project. The demodulation filter has an inductor value of 20.83 μH and a capacitor value of 2.04 μF. This results in an ideal resonant frequency of 24.4 kHz and a Q factor of 2.57 when an 8.2 Ω load is attached. It is shown in [5] that the frequency response of the demodulation filter remains very similar when a 4.1 Ω load is attached. The output stage is operated at DC-bus voltages of 15 V and -15 V.

5.6 Sigma-Delta Analogue-to-Digital Converter

The system contains an Analog Devices AD7760 sigma-delta ADC in the feedback loop which will be used to digitise the output signal. It is desirable to have the sampling frequency of the ADC as high as possible to ensure a high quality signal. The maximum sampling frequency of the ADC is 20 MHz [24]. The sampling frequency is therefore chosen as 19.6608 MHz and can be achieved by clocking the ADC using the 19.6608 MHz clock received from the clock distributor (Clock 2 in Figure 5.1). In Figure 5.1 the ADC is modelled with the transfer function

$$H(z) = z^{-n}, \tag{5.10}$$

where n is equal to the amount of sample delays the ADC contains. The shaped quantization noise which is added to the system by the ADC is depicted by $Q_2(z)$.

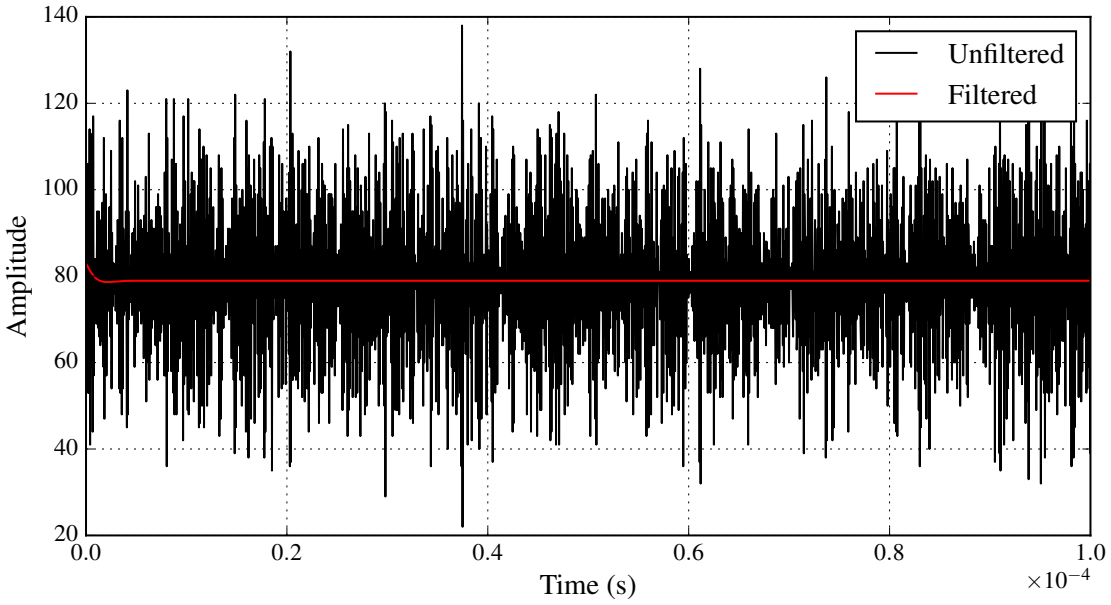


Figure 5.6: ADC Output with 5 V input.

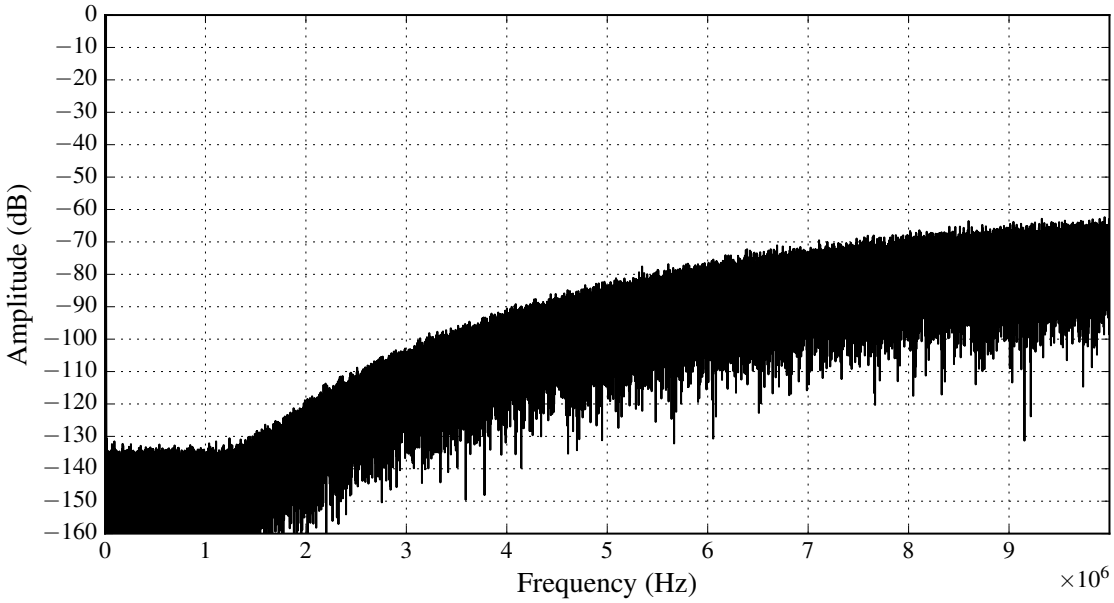


Figure 5.7: FFT of Data Output in Modulator Output Mode [24].

As mentioned in Section 3.5 the ADC will be operated in modulator output mode, bypassing the built-in low-pass FIR filters to minimise the time delay of the ADC. The output format in modulator mode is a 16-bit twos complement number. The value is, however, scaled to 15 bits and bits 0 to 3 are always zero [24]. The ADC thus effectively provides a 12-bit signal. A resistive divider is used to scale the amplifier output signal to within the maximum input ranges of the ADC. It is clear that the output will have

to scaled to reflect the correct output signal of the amplifier. Figure 5.6 illustrates the ADC output when a 5 V DC signal is applied to the input. The signal contains a lot of high-frequency noise, making it impossible to accurately determine the necessary scaling factor. After passing the signal through a low-pass filter it is, however, clear that the output value is 78.19 when applying 5 V to the input and a scaling factor of 0.064 will have to be implemented in the FPGA. This is in addition to the $\frac{1}{V_d}$ scaling factor which needs to be implemented to compensate for the output stage gain of the system.

The shaping of the quantization noise done by the ADC can be seen in Figure 5.7. It is clear that the noise profile is relatively flat up until approximately 1.5 MHz. The quantization noise does, however, rise at higher frequencies with an increase of around 60 dB at half the clock frequency. It will therefore be necessary to implement a low-pass filter in the FPGA to attenuate the quantization noise above 1.4 MHz as the unattenuated high-frequency noise will result in fold back distortion. It is important to remember that the ADC has an 8 cycle delay therefore contributing 407 ns to the total propagation delay of the system [20].

5.7 Conclusion

This chapter covered the practical implementation of the system. The different hardware components and their specifications were discussed and an overview of the system operation was given. The implementation of the digital control system in an FPGA was done and the next step will be to take measurements and determine the amplifier's performance.

Chapter 6

System Validation and Performance Analysis

6.1 Introduction

This section will cover the validation of the system operation and experimental results will be presented. A Tektronix TDS 3014C oscilloscope, together with the SignalTap function of Quartus, is used to perform basic signal measurements which will provide insight into the operation of the different components of the system. The amplifier's performance will be analysed using an Audio Precision SYS-2722 audio analyser.

6.2 System Validation

6.2.1 Open-Loop Operational Tests

The first step in validating the system's operation is to verify that the fundamentals of the system function correctly. The PWM operation will therefore be tested first. A sinusoid input signal with an amplitude of 0.8 and a frequency of 10 kHz is applied directly to the PWM input. The sawtooth carrier signal, which is produced by the FPGA, is shown in Figure 6.1. It is clear that the sawtooth carrier is generated correctly as the value increases up to 127 (7-bit signal) before being reset to 0. The sawtooth carrier signal also clearly has a frequency of 768 kHz, which is the required switching frequency.

The digital input signal is initially generated by the Audio Precision SYS-2700 audio analyser. The signal is then upsampled by the SRC4392 sample rate converter. The output of the sample rate converter, which is a 24-bit signal sampled at 196.601 kHz, is shown in Figure 6.2. It is clear that the value of the signal changes at a rate of

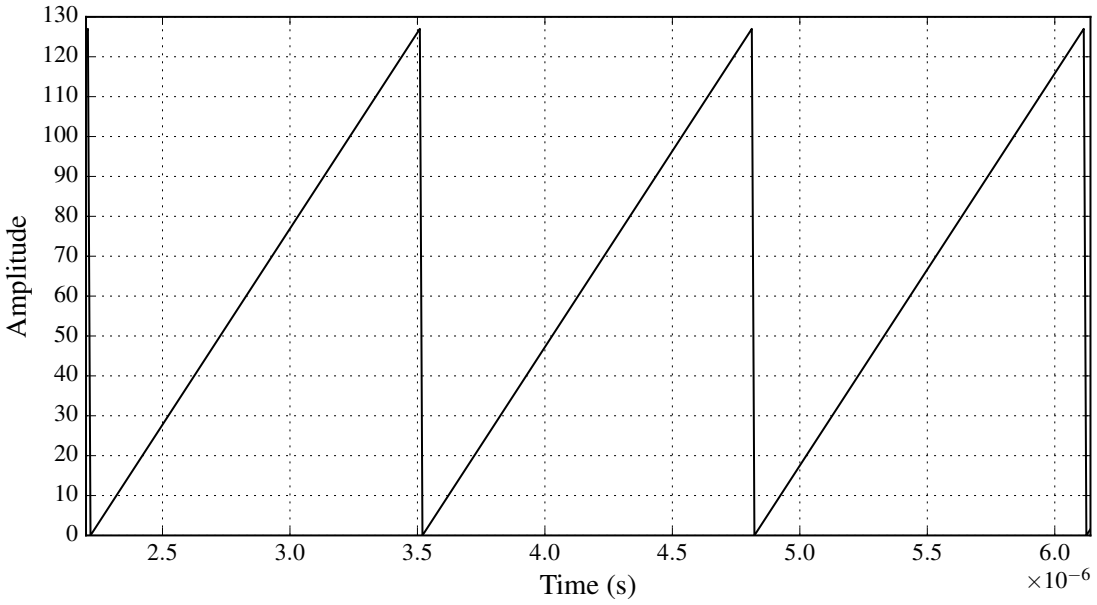


Figure 6.1: Sawtooth carrier produced by the FPGA.

196.601 kHz and the values coincide with the expected values, which are the values of a sinusoid multiplied by 0.8, fitted to a 24-bit signal.

The signal is then upsampled, as discussed in Chapter 5, to produce the input reference signal which has a sampling frequency of 19.6608 MHz. The output of the upsampler, and thus the PWM reference signal, is shown in Figure 6.3. The reference signal must,

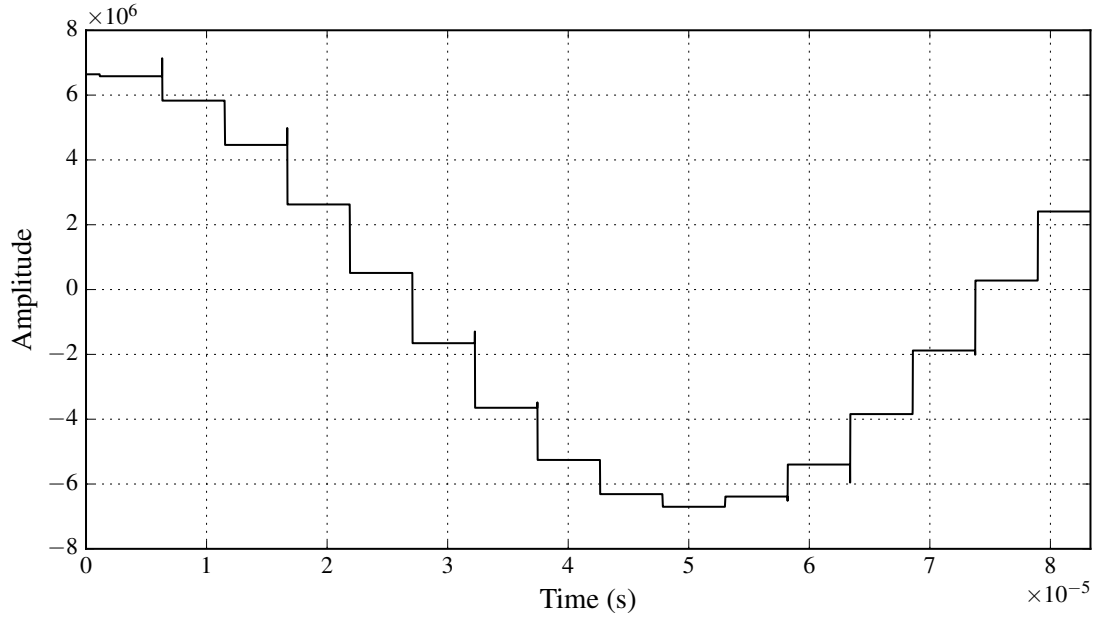


Figure 6.2: Input received from sample rate converter.

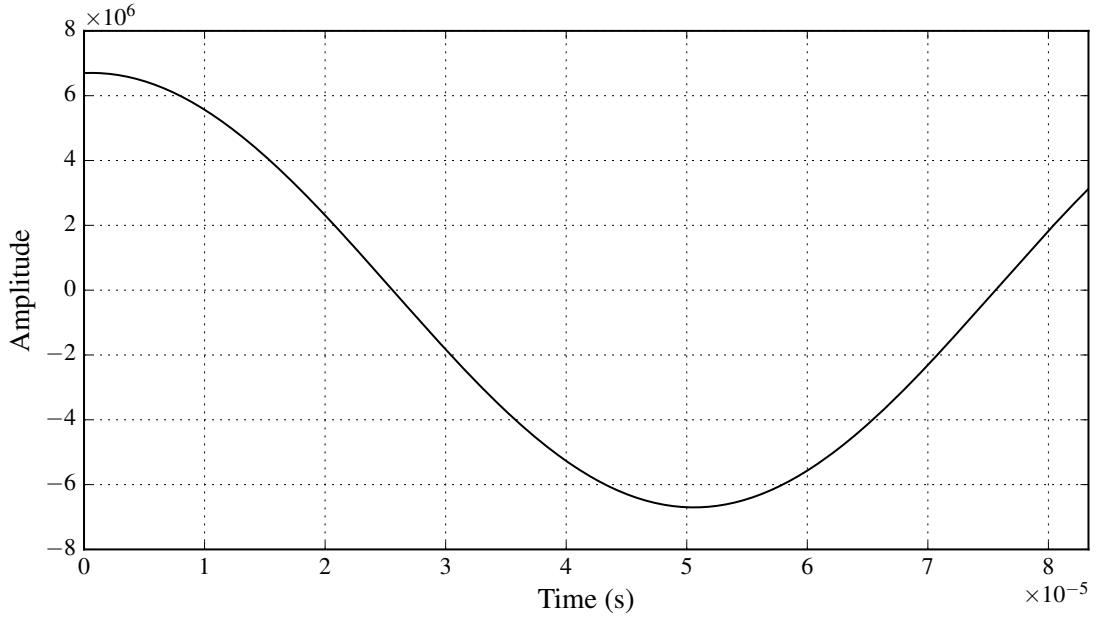


Figure 6.3: PWM reference signal.

however, be rescaled to a value between 0 and 127 before being compared to the sawtooth carrier. The scaled version of the PWM input signal is shown in Figure 6.4.

An example of the PWM output, which is provided by the FPGA and then reclocked by external flip-flops, is shown in Figure 6.5. It is clear that the '1' output provided by the FPGA translates to a 3.3 V flip-flop output. This PWM output is then used as input

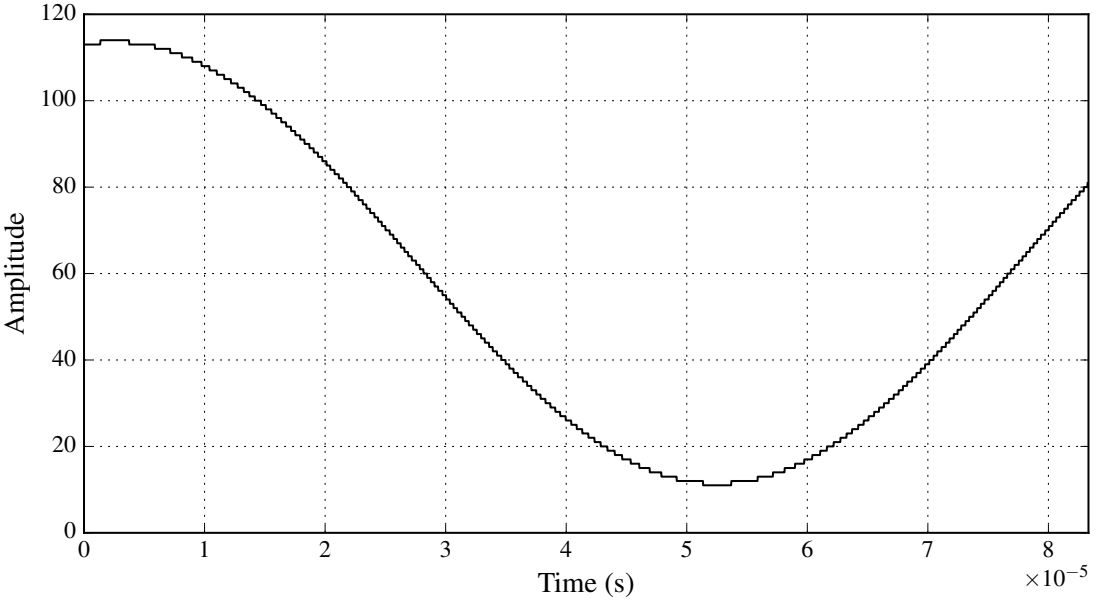


Figure 6.4: Scaled PWM reference signal.

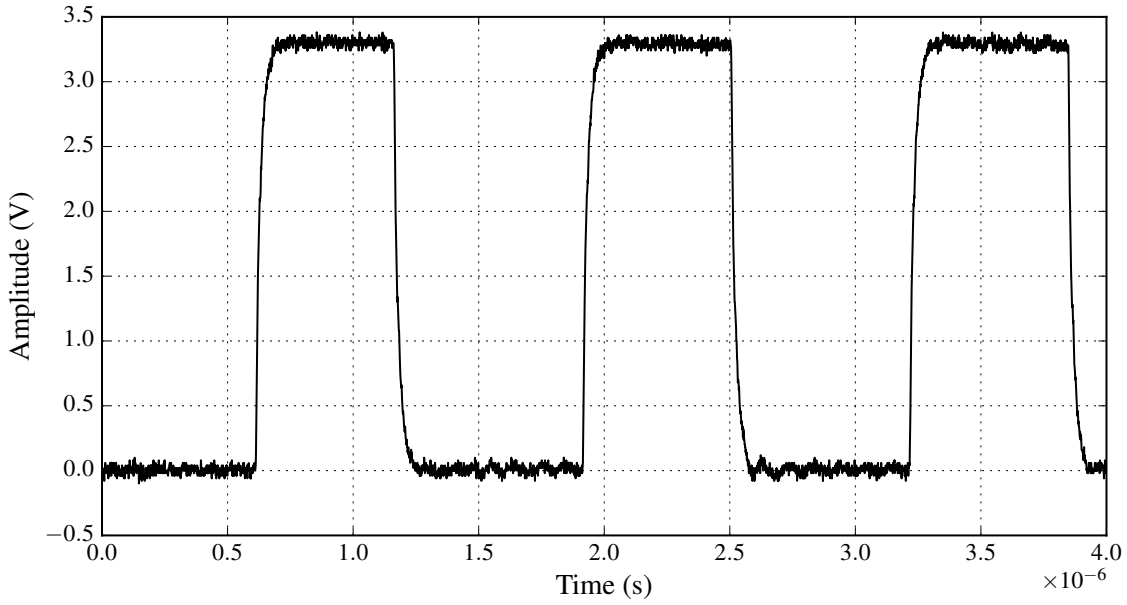


Figure 6.5: PWM output signal example.

to the power switching stage, of which the output is shown in Figure 6.6. The output produces a sinusoidal signal with the same frequency as the input signal (10 kHz). The high-frequency voltage ripple, which is present due to the switching of the amplifier, is clearly visible. The signal also has a small DC offset, which is to be expected as it is currently an open-loop system with no controller added.

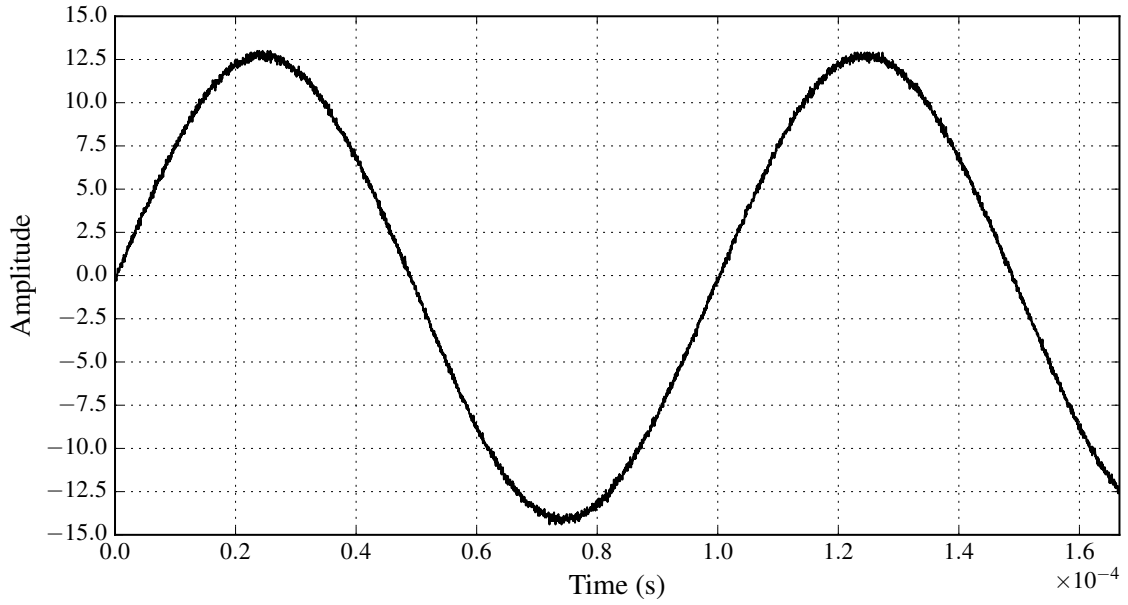


Figure 6.6: Open-loop output signal with 10 kHz input signal.

It is clear that the basic system consisting of a digital input signal, upsampler, PWM and output stage operates correctly. The next step is to implement a closed-loop system by using the ADC to provide output feedback to the FPGA.

6.2.2 Closed-Loop Operational Tests

The system will now be expanded by including a feedback loop. The ADC will be used to digitise the output signal making it possible to compare the output signal to the input signal and determine an error signal. As discussed in Section 5.6, the noise floor of the ADC starts to rise at approximately 1.5 MHz. The low-pass filter discussed in Section 3.5.3 is therefore included to attenuate the high-frequency noise induced by the ADC.

The system is once again tested with a sinusoidal input signal with a frequency of 10 kHz and amplitude of 0.8. The 12-bit digitised signal provided by the ADC is shown in Figure 6.7. The signal contains a notable amount of high-frequency noise, as expected. The feedback signal is then scaled and subtracted from the input signal to determine an error signal. The error signal is shown in Figure 6.8. The blue signal illustrates the error signal before the low-pass filter and the red signal shows the output of the low-pass filter. It is clear that the low-pass filter attenuates the high-frequency noise of the ADC. The amplifier output, measured with an oscilloscope, is shown in Figure 6.9. The high-frequency voltage ripple is once again evident. There is also an amplitude discrepancy. The expected output is a sinusoid with a maximum value of 12 V, while the output

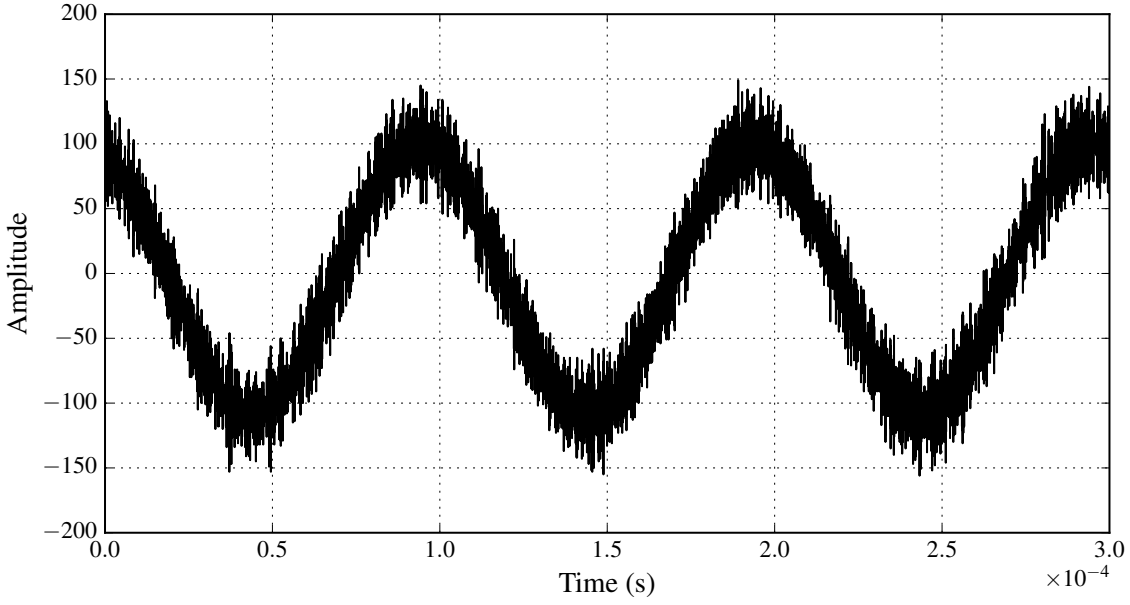


Figure 6.7: Output signal measured by ADC (10 kHz).

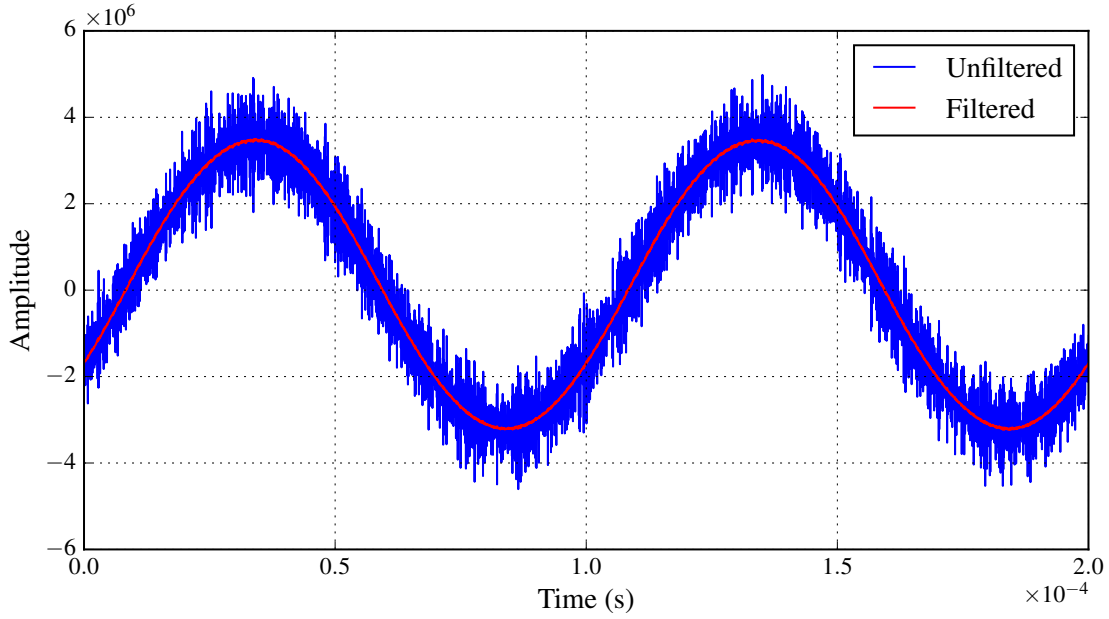


Figure 6.8: Filtered and unfiltered error signal.

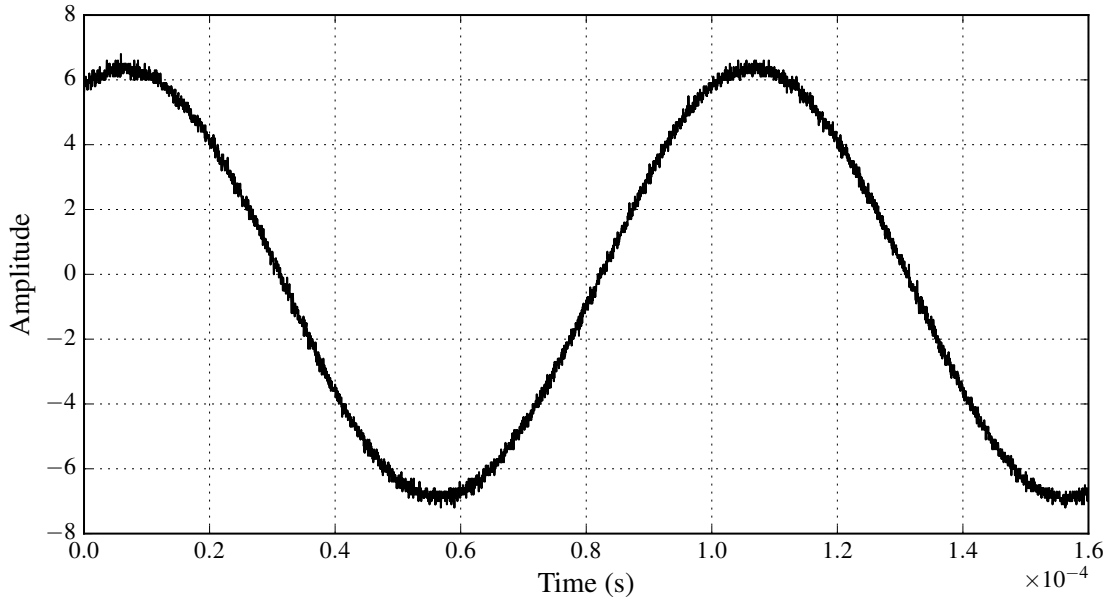


Figure 6.9: Output of the closed-loop amplifier with a 10 kHz sinusoid input.

currently has a maximum value of approximately 6 V with a small DC offset. This is once again due to the fact that the system does not contain a controller and therefore does not track the reference signal accurately. It is clear that the basic amplifier operates correctly. The infrastructure is thus firmly in place to implement a controller. The performance of the amplifier will then be determined using the SYS-2722 audio analyser.

6.3 Performance Analysis

In this section the performance of the amplifier is analysed. This is done obtaining a range of different measurements using the Audio Precision SYS-2700 audio analyser. This instrument specifically focusses on the audio band (20 Hz to 20 kHz) as these are the only frequencies audible to the human ear. The output of the amplifier is passed through an AUX-0025 switching amplifier measurement filter before presented to the audio analyser. Unless stated otherwise, the measurements are conducted with a sinusoid input with an amplitude of 0.8 and a frequency of 1 kHz.

The output of the open-loop system, where the reference signal is directly compared to the sawtooth carrier as discussed in Section 6.2, is shown in Figure 6.10. It is important to note that the high-frequency voltage ripple is no longer visible and that the system does not have a DC offset (as was the case in Figure 6.6). This is due to the bandwidth filtering which is done by the audio analyser. The FFT of the output is shown in Figure 6.11. The fundamental component is clearly visible at 1 kHz, with plenty of higher order harmonics also visible. The noise floor is situated at approximately −80 dBV. This is expected as the system currently has no controller or feedback loop. The system has a THD+N measurement of 0.6 %.

The closed-loop system output, with a low-pass filter included as discussed in Section 6.2, is shown in Figure 6.12. It is once again clear that the maximum amplitude of

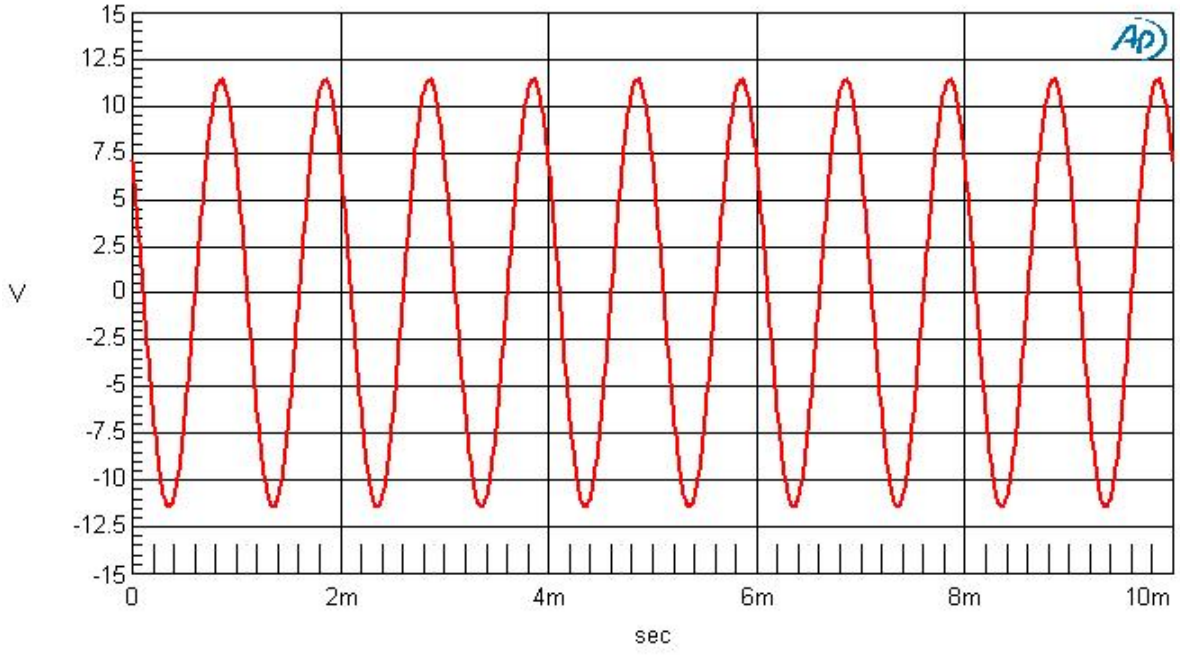


Figure 6.10: Open-loop output with 1 kHz input.

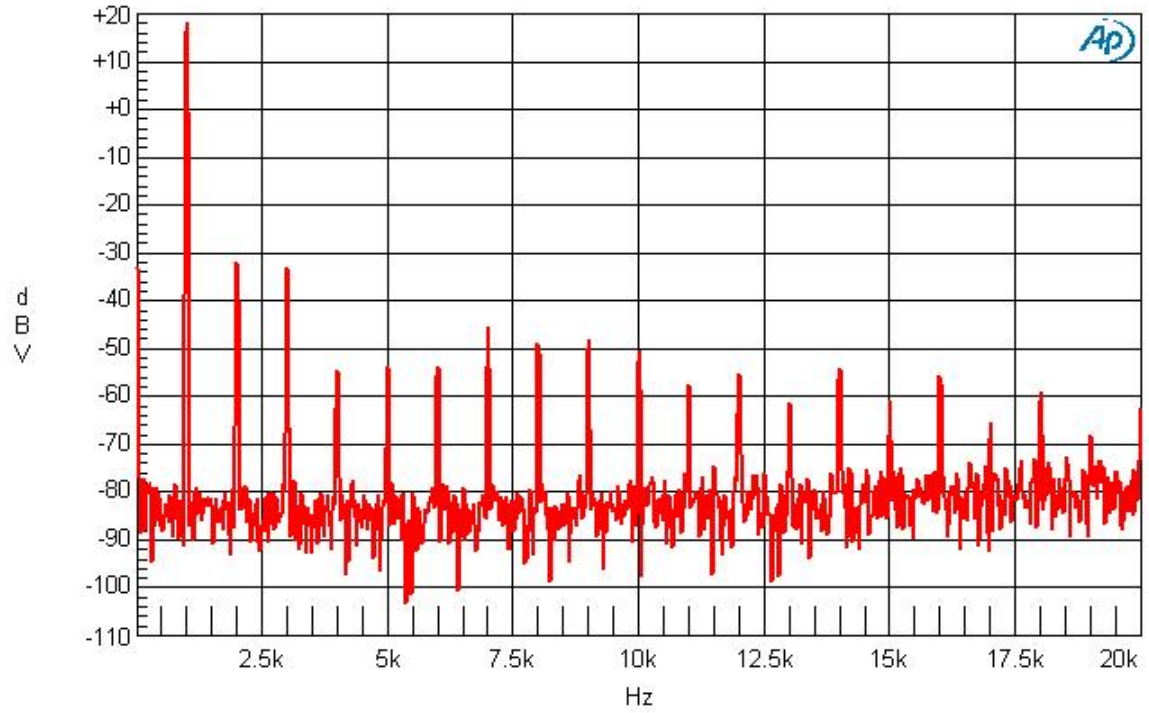


Figure 6.11: FFT of open-loop output with 1 kHz input.

the output is smaller than the expected value of 12 V. The FFT, as shown in Figure 6.13, illustrates that the higher-order harmonics of the system are a lot smaller than that of

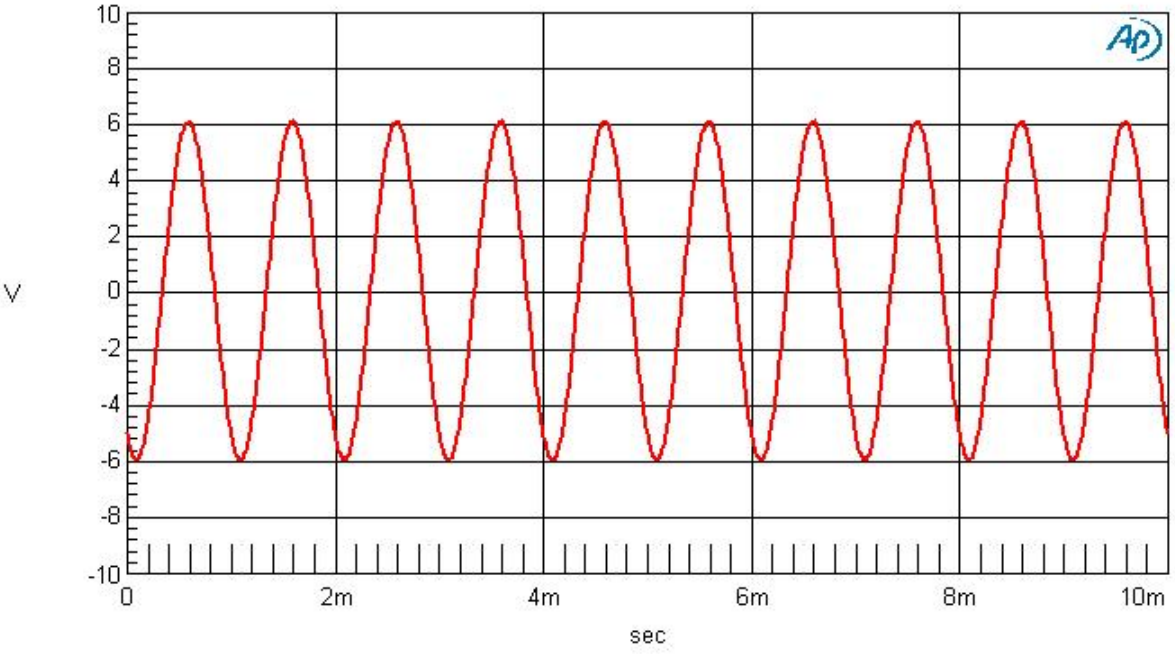


Figure 6.12: Closed-loop output with 1 kHz input.

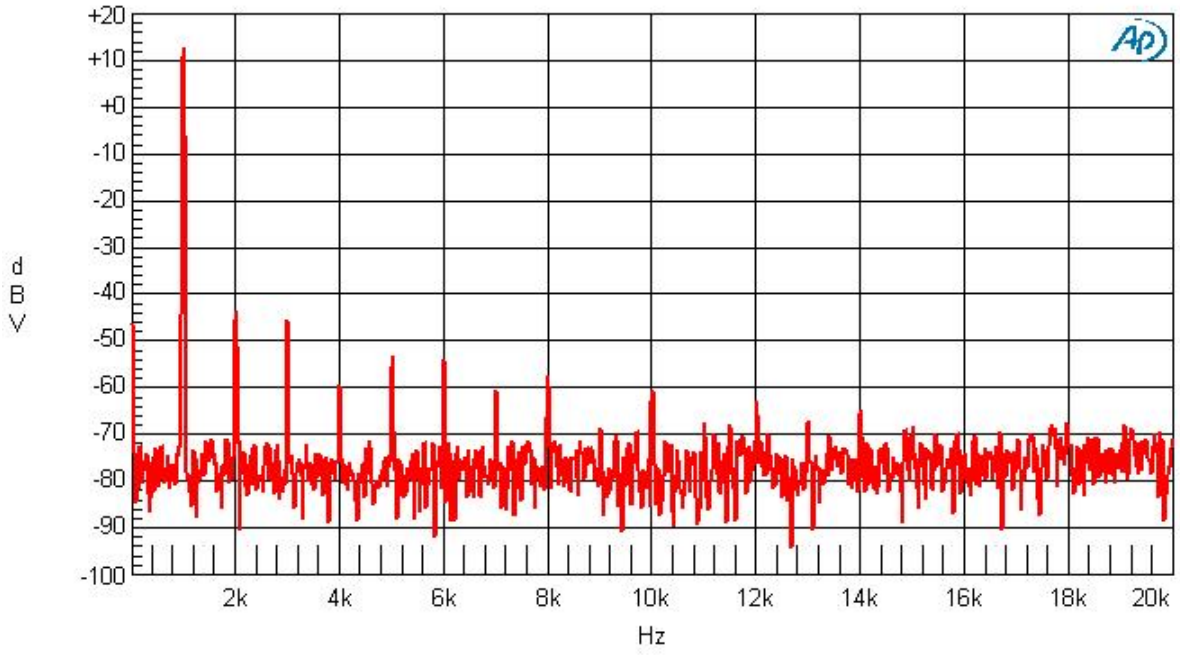


Figure 6.13: FFT of closed-loop output with 1 kHz input.

the open-loop system. This is confirmed when looking at the THD+N measurement of the system, which is now 0.5 %. The noise floor is situated at −75 dBV. This is slightly higher than the open-loop system and is due to the noise induced by the ADC.

The first controller to be implemented will be an unconditionally stable one. The poles of the system remain in the same location, but the zero positions are slightly altered to ensure that the system remains stable at a low gain. This will make it possible to test the system with a gain which starts low, and is then ramped up to obtain the optimal gain for operation. The main goal of this is to verify that the method used to implement the controller is correct. The closed-loop magnitude response of the system with the low-gain controller is shown in Figure 6.14. It can be seen that certain frequencies will be dampened, while other will be amplified. The output of the amplifier, with a 1 kHz sinusoidal input, is shown in Figure 6.15.

It is noted that the peak amplitude is much lower than what would be expected with a high-gain controller. This is in agreement with what is shown in Figure 6.14 in terms of the controller damping certain frequencies. In the FFT of the output signal, as illustrated by Figure 6.16, there are two downward spikes at 10.6 kHz and 17.8 kHz. This confirms the correct placement of the two complex pole pairs at those frequencies and coincides well with the FFTs determined during simulation. The correct implementation of the controller is further verified by changing the frequency of the reference input signal to 15 kHz. The output, where the amplitude of the signal once again coincides with the

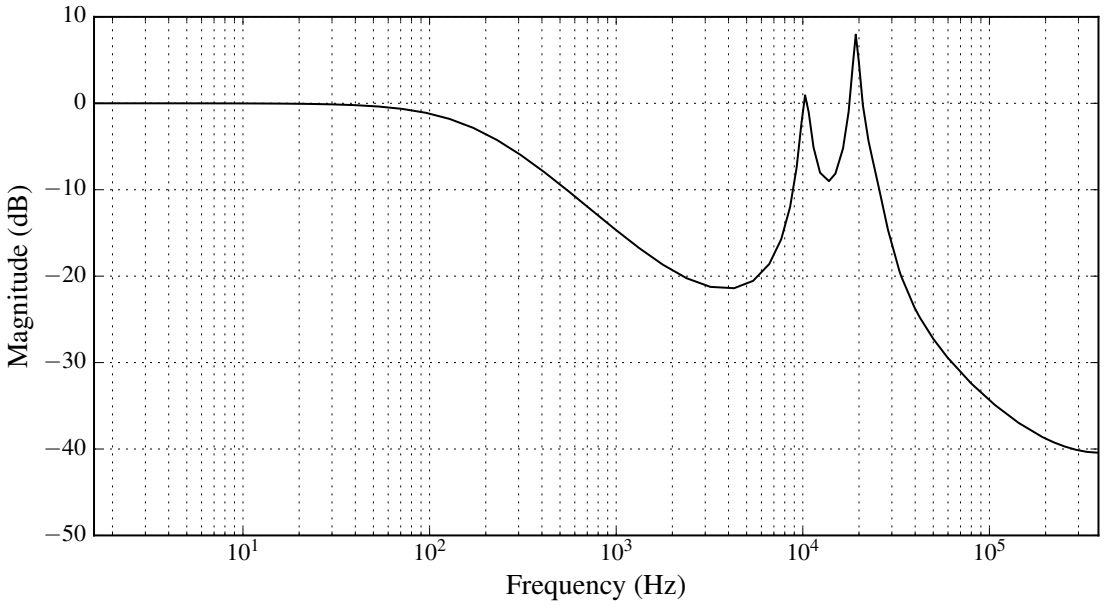


Figure 6.14: Closed-loop magnitude response of system with low-gain controller.

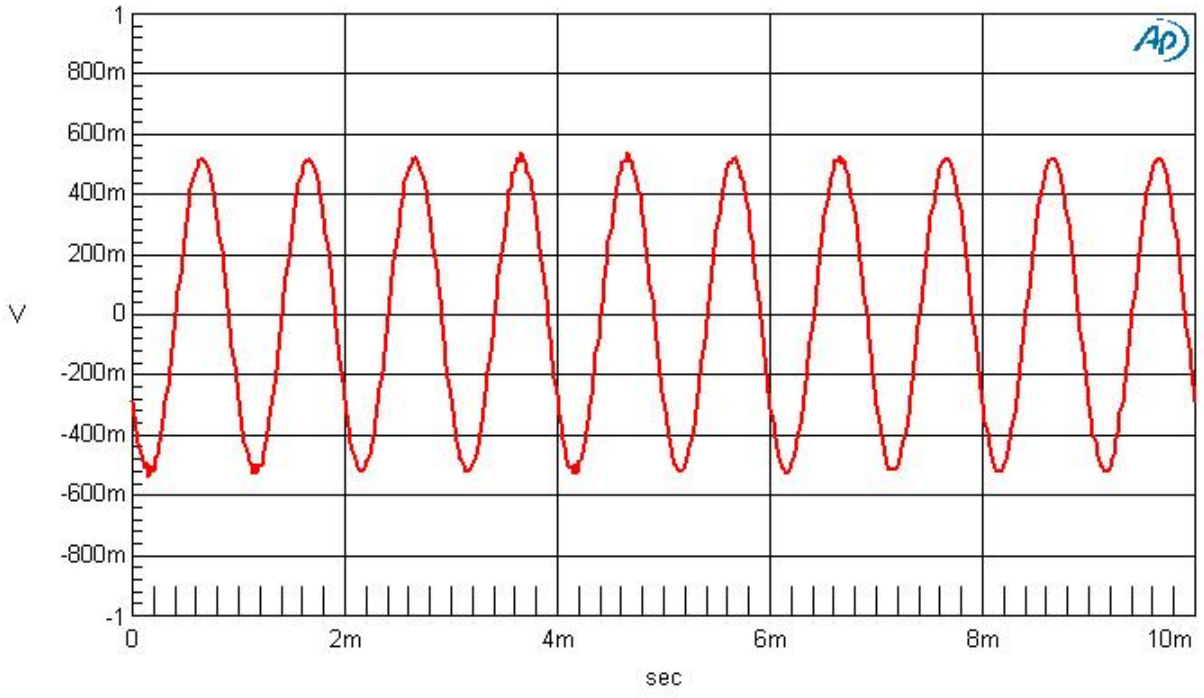


Figure 6.15: Amplifier output with low-gain controller (1 kHz input).

magnitude response of the system, is shown in Figure 6.17. The frequency response of the system is shown in Figure 6.18. This also coincides with the designed closed-loop magnitude response of the system as the frequencies in the lower half of the audio band

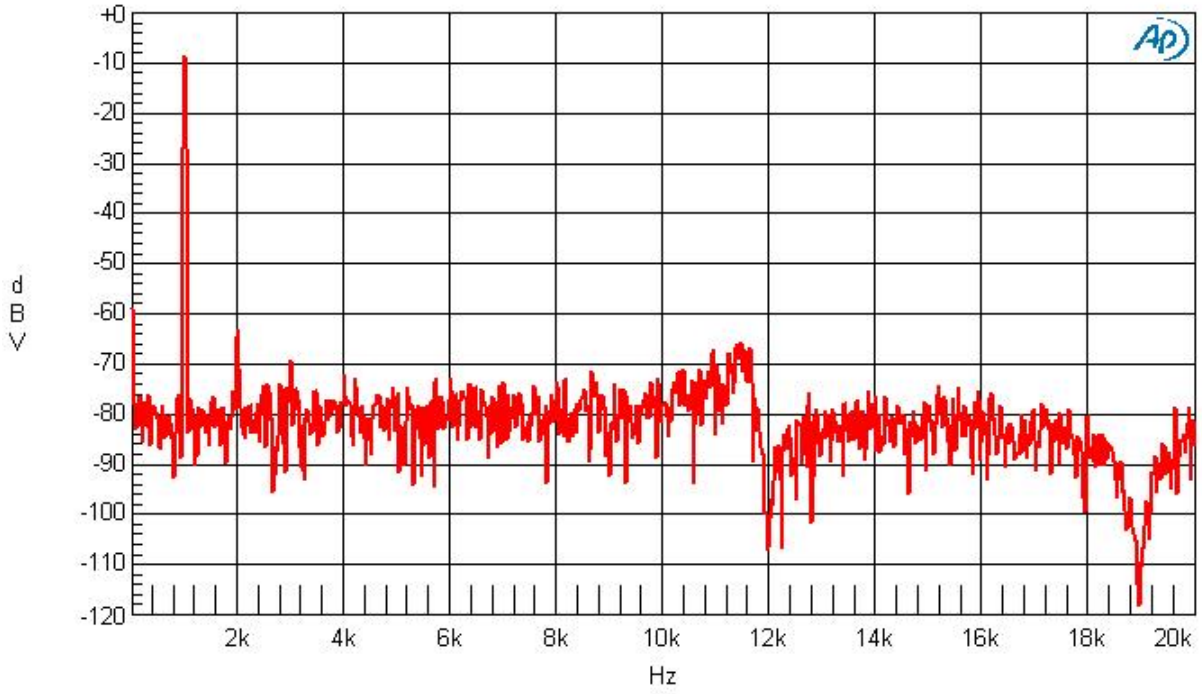


Figure 6.16: FFT of amplifier output with low-gain controller (1 kHz input).

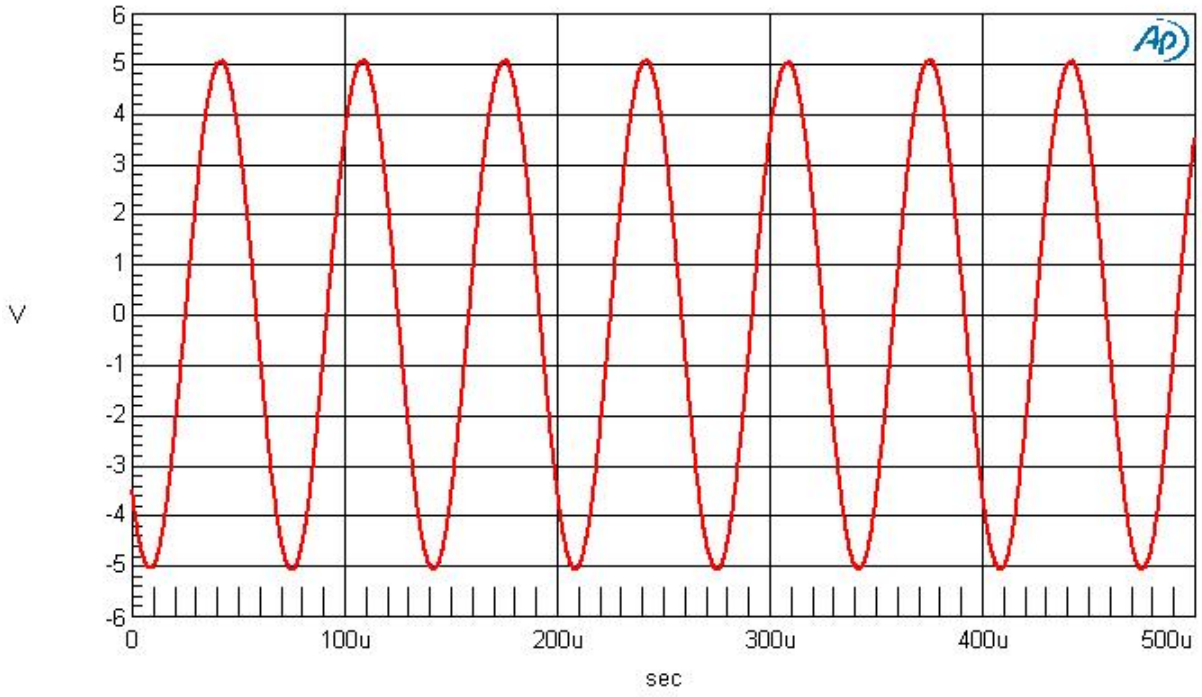


Figure 6.17: Amplifier output with low-gain controller (15 kHz input).

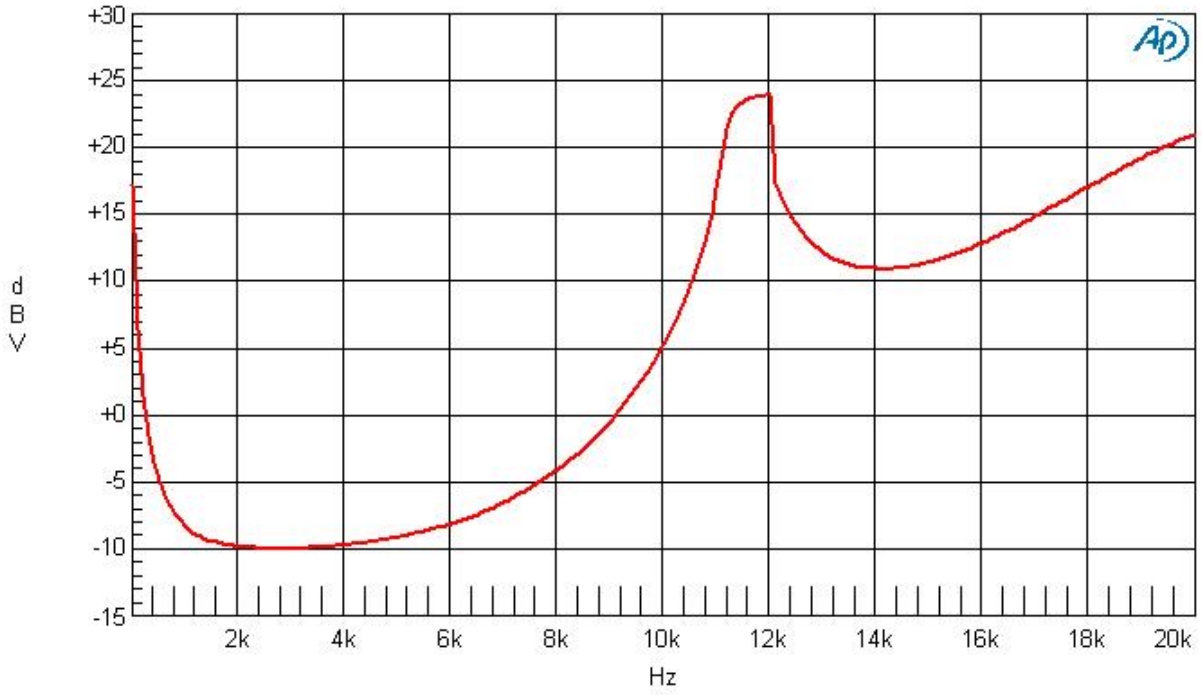


Figure 6.18: Frequency response of low-gain controller.

are more attenuated than those in the upper half of the audio band. The influence that the two complex pole pairs in the audio band has on the amplitude of the output is also evident.

The validation of system operation and controller implementation is now complete. The next step is to attempt to implement the high-gain controller designed in Chapter 3, which will ensure the amplifier provides high-quality audio.

During the implementation of the low-gain controller, it was seen that as the gain was increased, stability became a problem. The system became unstable at lower gain values than expected. It was also found that the variable sizes of the digital integrators (implemented as accumulators) played a massive role in the stability of the system. If the variable sizes were too large the system would often become unstable, while making the variable size too small would cause the integrators to saturate. Saturated integrators reduce the amplifier's performance.

It was seen in Chapter 4 that the designed system is not stable without ripple compensation. The pre-distorted sawtooth carrier, which is used to implement the ripple compensation, was measured on the FPGA using the SignalTap function and can be seen in Figure 6.19. The signal is almost identical to the simulated pre-distorted sawtooth carrier seen in Figure 5.5. The only difference being the amplitude scaled to the appropriate level inside the FPGA. When subtracting the pre-distorted sawtooth carrier from

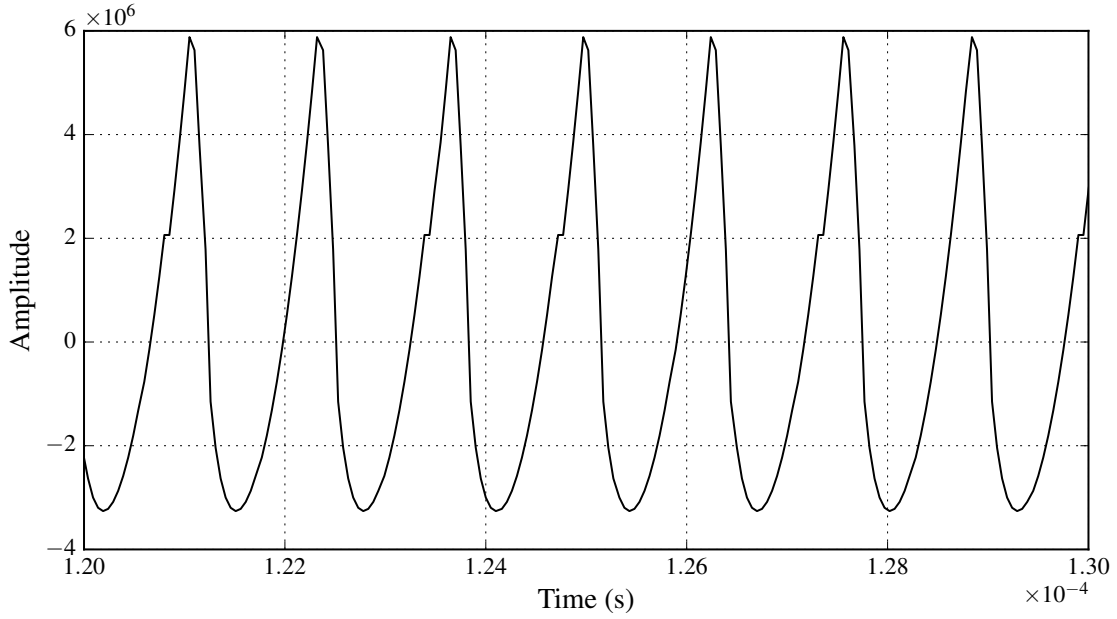


Figure 6.19: Pre-distorted sawtooth carrier used for ripple compensation.

the pole-cancellation filter output, it is important to incorporate the time delay of the system to ensure the signals are in phase. Otherwise, the ripple compensation will have minimal to no effect and could potentially cause the system to become unstable.

When an attempt was made to implement the high-gain controller designed in Chapter 3, the system was found to be unstable even with ripple compensation. It was found that the system is only stable once the integrator variable values were decreased, therefore saturating the integrators. This was still the case even after the anti-windup scheme discussed in Section 2.6 was implemented.

The possibility that the amplifier becomes unstable during startup, and then remains in that state, was also investigated. This possibility was nullified by implementing an activate button for the amplifier. The control system does therefore not receive any feedback or produce any PWM output until the startup procedures of the amplifier are complete, after which the user can activate the amplifier using an external button. A soft-start was also implemented with the user having control over the gain of the system via another external button. The other potential cause of instability, is the time delay of the system. The propagation delay has an influence on the z-domain pole mapping. The system can therefore potentially be unstable to the design not compensating for the correct propagation delay. The design of the system was altered to compensate for both longer and shorter propagation delays with no success.

In spite of all these attempts to stabilise the system, it remained unstable. The design was optimised by slightly changing the zero positions and altering the size of the integrator

variable sizes to provide the best possible results, given that the system remains stable. The rest of the chapter will discuss the stable amplifier's performance.

The optimised amplifier is found to be stable with and without ripple compensation. Figure 6.20 illustrates the FFT of the output signal without ripple compensation, given a sinusoid with a frequency of 1 kHz (amplitude remains 0.8). The fundamental component is clearly visible with multiple harmonics also present. These harmonics are decreased when ripple compensation is added as shown in Figure 6.21, where the FFT of the output signal is shown with ripple compensation implemented.

To further demonstrate the correct operation of the amplifier, the frequency of the input signal is changed to 6 kHz. The FFT of the output signal is shown in Figure 6.22. The fundamental component is clearly visible with the same amplitude as when the reference signal had a frequency of 1 kHz. Higher order harmonics are also visible. A two-tone signal, with frequencies of 18 kHz and 19 kHz, was also supplied as a reference signal. The FFT of the output is shown in Figure 6.23.

In all four the above mentioned FFTs, the noise floor is situated at approximately −75 dB. This is a lot higher than the desired noise floor. This is due to the fact that the system becomes unstable once the gain is notably increased. The system does therefore not possess the gain to adequately attenuate the noise in the audio band. The frequency response is shown in Figure 6.24. The frequency response shows a sharp drop at fre-

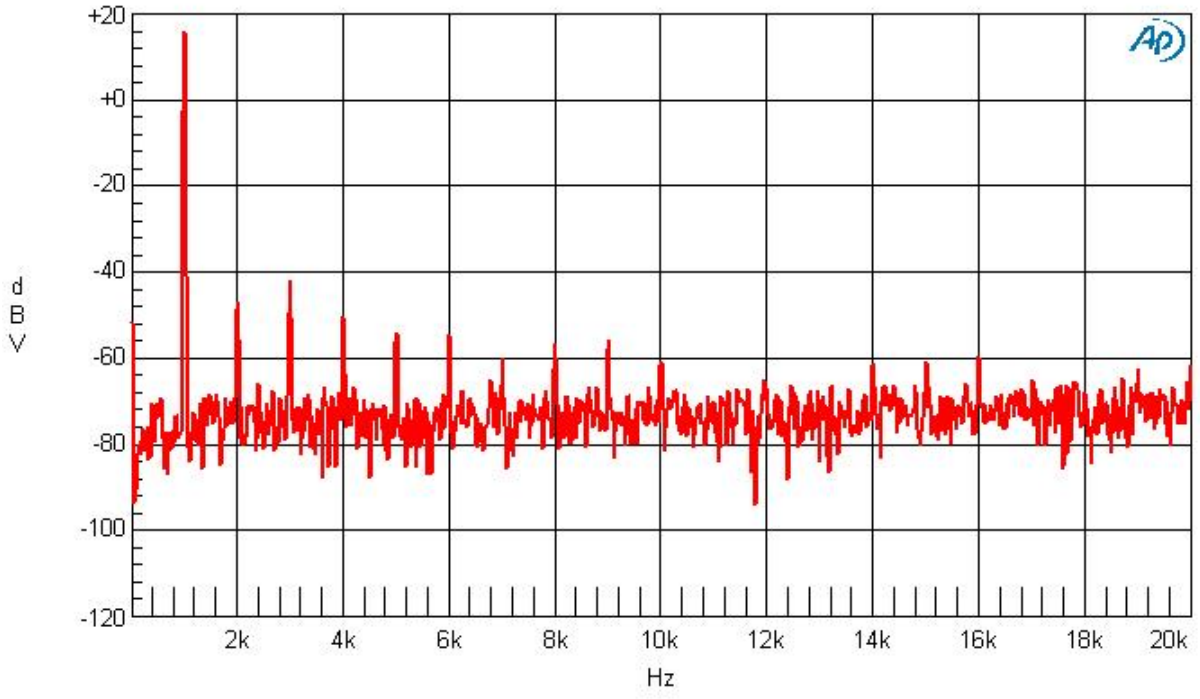


Figure 6.20: FFT of output signal without ripple compensation with 1 kHz input.

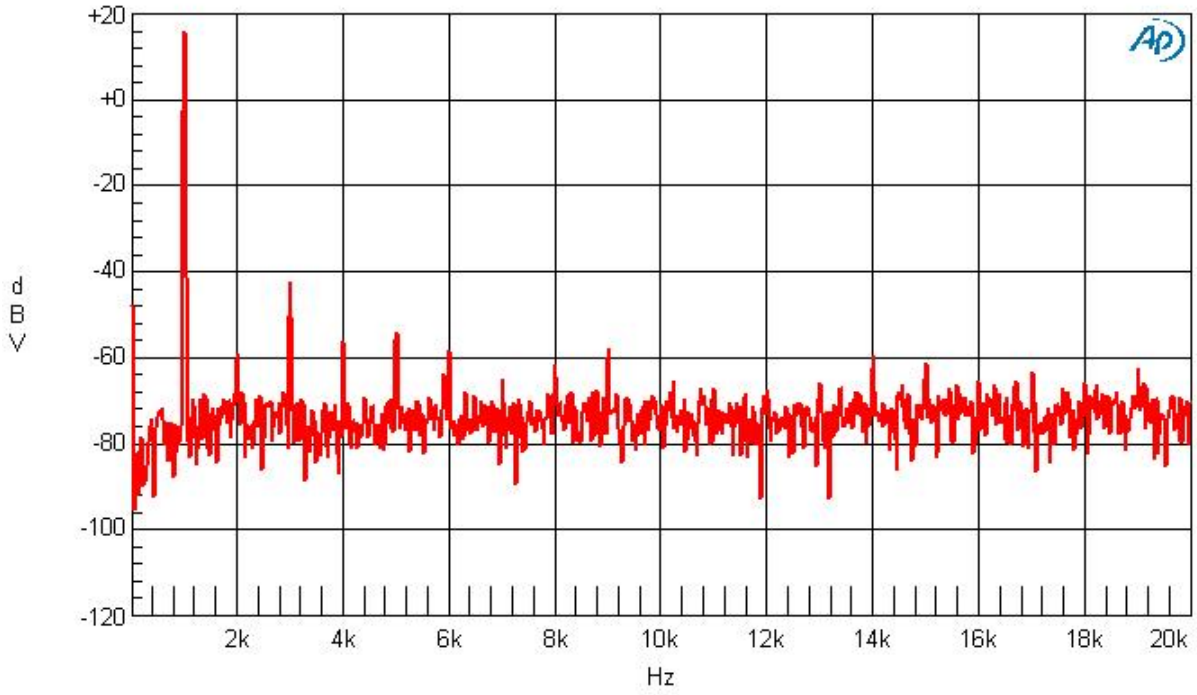


Figure 6.21: FFT of output signal with ripple compensation with 1 kHz input.

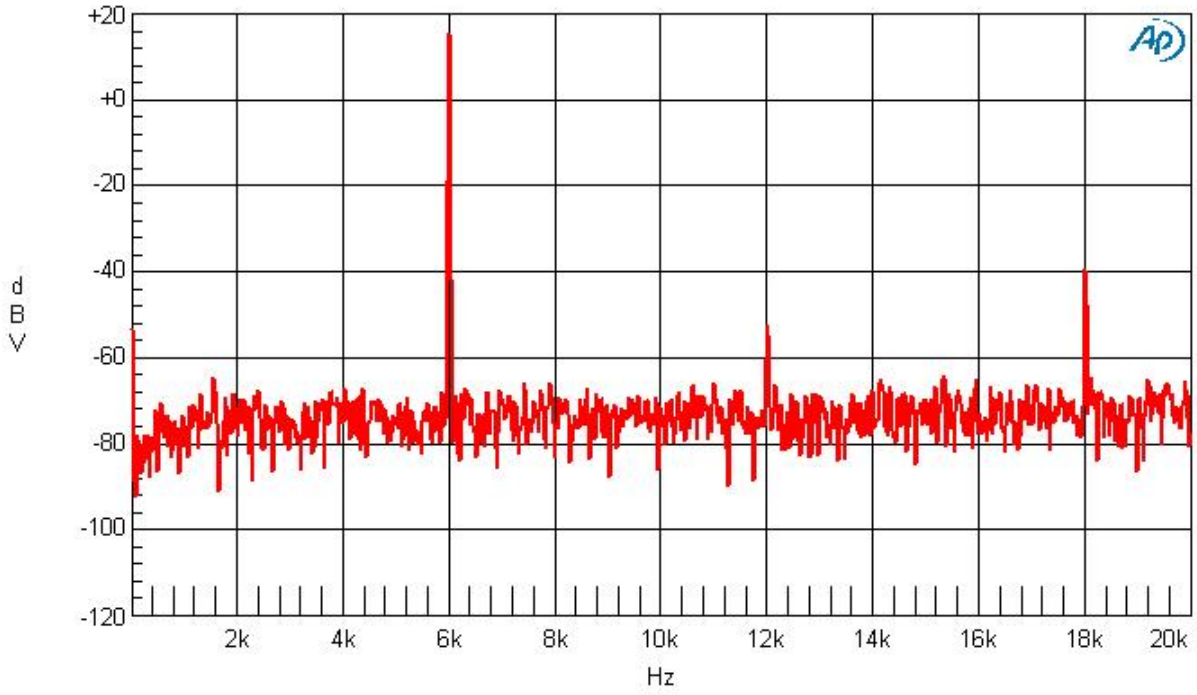


Figure 6.22: FFT of output signal with ripple compensation with 6 kHz input.

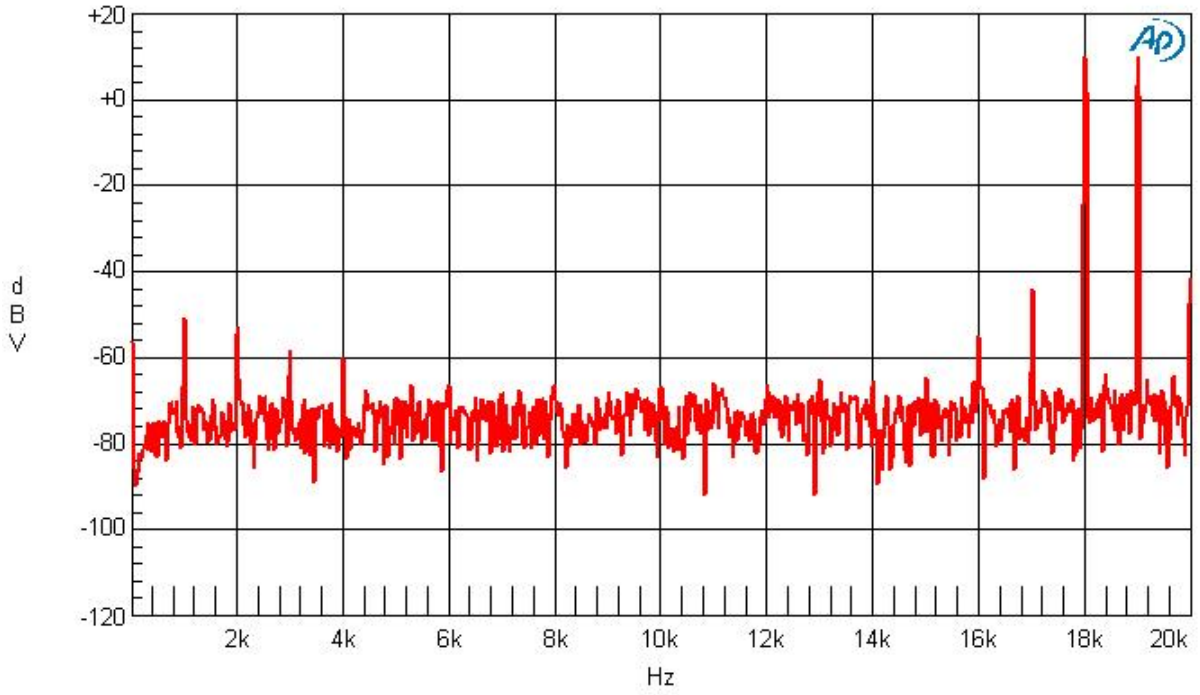


Figure 6.23: FFT of output signal with two-tone input signal (18 kHz and 19 kHz).

quencies below 1 kHz. This is expected as the noise floor of the FFTs also illustrate a decreased level at these low-frequencies, meaning the amplifier has a slightly higher gain in this region. The frequency response then slowly increases along with frequency, resulting in a variation of approximately 1 dBV from frequencies of 1 kHz and higher.

The THD+N of the system is firstly measured as a function of frequency and is shown in Figure 6.25. The THD+N value remains below 0.5 % up until approximately 6 kHz, where it starts to increase dramatically and decreases again below 0.5 % at approximately 17 kHz. The sharp peak at roughly 14 kHz is due to large second-order harmonics, which are not within the audio band.

The THD+N is secondly measured as a function of power. The load has a value of 8.2 Ω. It is clear that as the power, and thus the amplitude of the input signal, increases, the THD+N decreases. This is to be expected as an increase in the input signal will lead to an increased fundamental component of the output. The THD+N starts at 1.4 % at low power, before decreasing to below 0.4 % at 6 W.

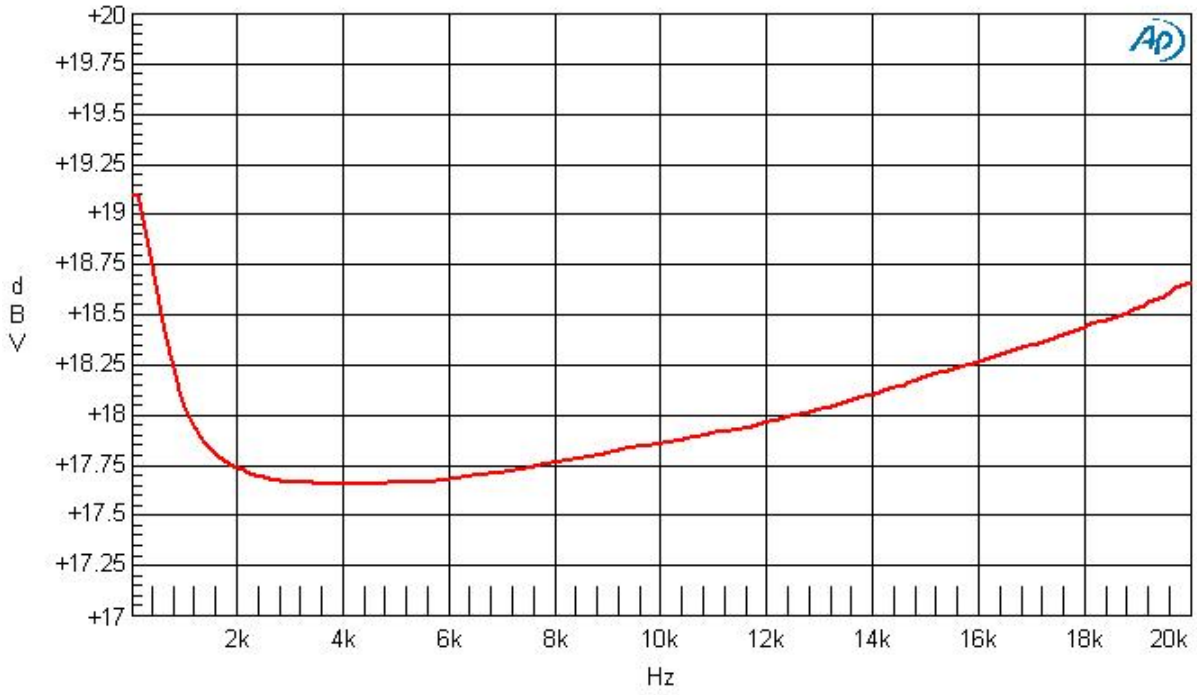


Figure 6.24: Frequency response.

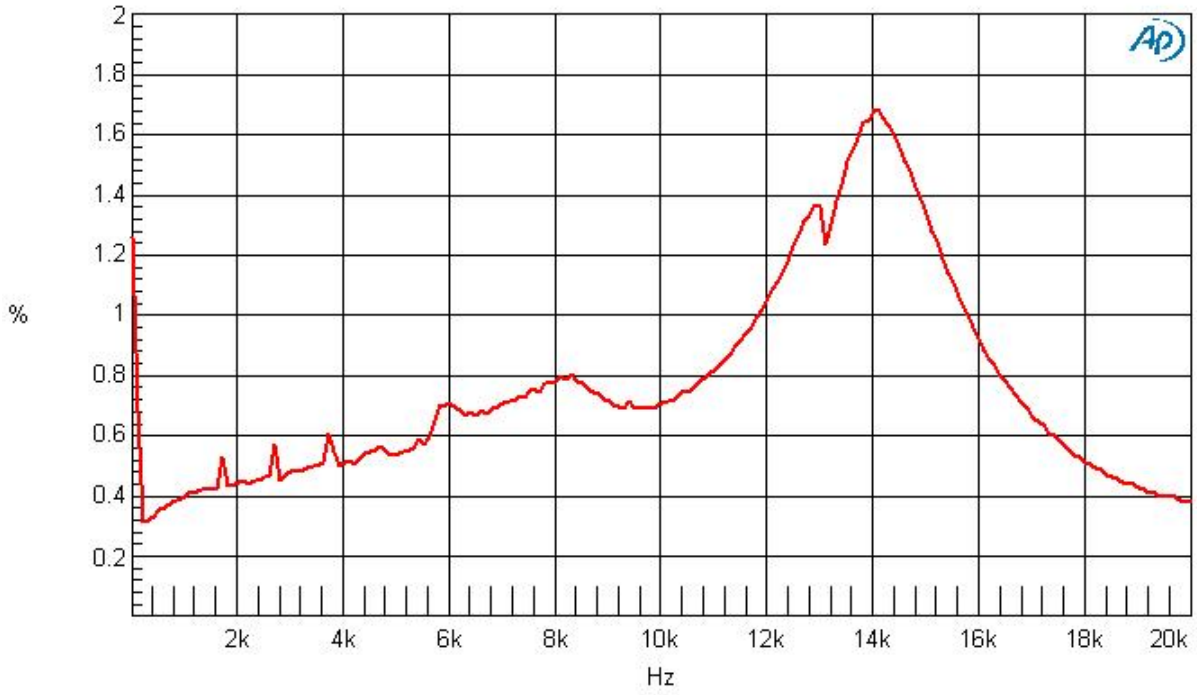


Figure 6.25: THD+N as a function of frequency with an input amplitude of 0.8.

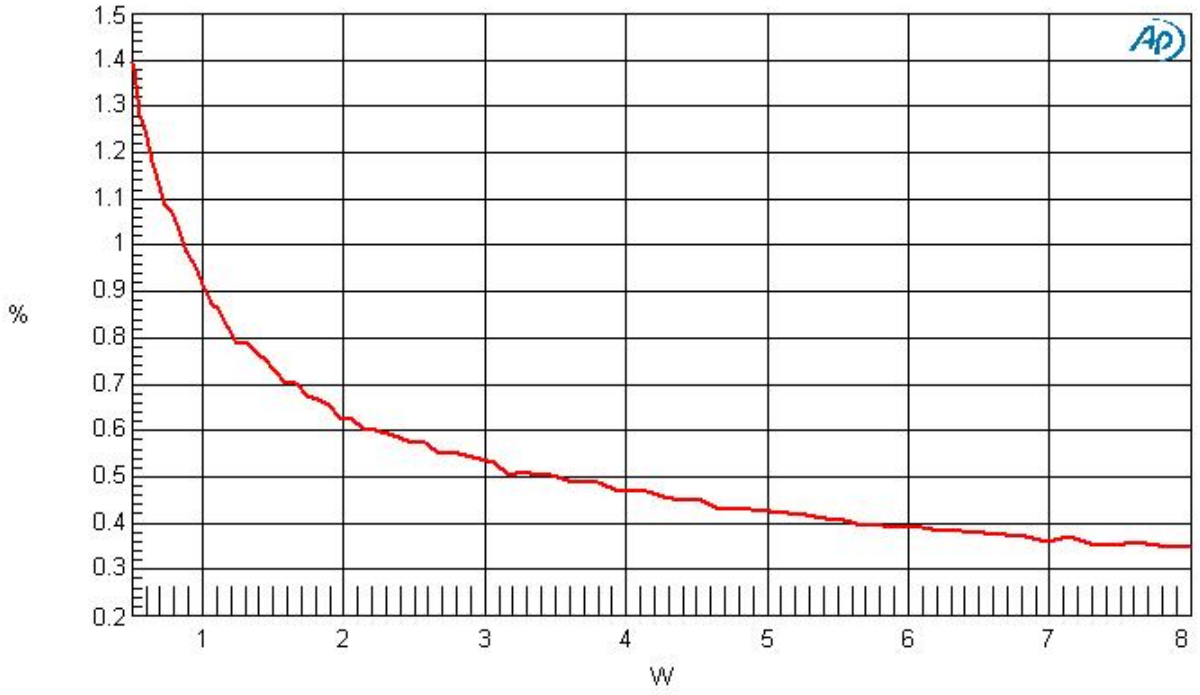


Figure 6.26: THD+N as a function of power with 1 kHz input and an 8.2 Ω load.

6.4 Conclusion

This section discussed the validation of the system operation and analysed the amplifier's performance. A low-gain controller was initially implemented to verify the correct operation of the system. This was successful and confirmed that the baseline infrastructure of the practical implementation in terms of hardware and software is correct.

The high-gain controller designed in Chapter 3 proved to be unstable. Various attempts were made to stabilise the system to no avail. The system was then slightly altered to ensure stable operation. Measurements were taken to determine the amplifier's performance. The amplifier's performance did not come close to the desired state-of-the-art performance. This is largely due to the fact that the integrators saturated or the system became unstable when the gain was increased to high levels, resulting in poor performance. This occurred most likely due to an error in the digital scaling of the system gain. This implies that there is a discrepancy between the gain thought to be implemented, and the actual implemented gain. A high system gain is key to the amplifier's performance as this ensures that the noise is adequately attenuated. The instability could also potentially have occurred to the incorrect calculation of the total propagation delay of the system. The exact calculation of the system propagation delay will have to be investigated in further work.

Chapter 7

Conclusions

7.1 Overview

7.1.1 Design

In Chapter 3, the system was successfully designed ensuring that the system met the design requirements. These were a stability margin of at least 3 dB and an open-loop controller gain of 50 dB or more throughout the audio band. An accurate discrete-time domain comparator model was successfully used during the design. The closed-loop response of the system was analysed and found to be adequate. A mathematical expression for the PWM input signal was derived. This was used to determine the small-signal comparator gain. The optimal controller topology for the specific project was also chosen.

7.1.2 Simulation

The system was simulated in Chapter 4. The Simulink simulation model included the sigma-delta analogue-to-digital converter and ripple compensation was implemented to ensure a constant small-signal gain. This ensured that the system remained stable as the small-signal gain could accurately be compensated for. Using a bifurcation diagram, it was found that the root-locus design predicted the stability margin of the system correctly. The FFT of the output signal clearly illustrated that the controller adequately attenuated the noise in the audio band. It can therefore be assumed that the design and simulation of the system was successful and met all pre-determined requirements.

7.1.3 Practical Implementation and Measurements

Chapter 5 and 6 discussed the practical implementation of the system and its performance. The system was implemented as set out in the initial objectives, using only an

FPGA (with peripheral components), a sigma-delta analogue-to-digital converter and an output stage. The basic system infrastructure was successfully implemented. The high-gain controller, designed in Chapter 3, proved to be unstable. The system remained unstable with different designs and proved to be unstable at unexpected system gain values. This was potentially due to a discrepancy in the gain thought to be implemented and the actual implemented gain. This is possibly due to an error in the digital scaling of the signals inside the FPGA. The instability could potentially also be caused by the incorrect calculation of the propagation delay of the system as this influences the placement of the z-domain zeros. The state-of-the-art audio performance, stipulated in Section 1.3, was therefore not achieved.

An amplifier with lower performance standards was, however, successfully implemented confirming the validity of the design approach and implementation method. The amplifier produced low audio performance due to not having enough gain to adequately attenuate the noise in the audio band. The performance of the system was also decreased by the saturation of the controller integrators.

The successful design, simulation and to a lesser extent implementation of the digitally controlled class-D amplifier proves that with further work the amplifier can be improved to provide state-of-the-art audio performance and eventually be implemented in a single IC.

7.2 Future work and Improvements

In future work it will be necessary to investigate the digital scaling of the signals. A potential option is to implement the MATLAB fixed-point tool during simulation. This will provide a more accurate presentation of the digital variable sizes and different scaling coefficients present during the practical implementation. This could potentially solve the gain implementation problem.

The implementation of the anti-windup scheme and digital filters should also be investigated further in order to minimise the time delay which they contribute to the system. A method can also be devised to ensure that the propagation delay of the system is calculated accurately as this could also be a reason for the instability of high-gain systems. Once the amplifier achieves state-of-the-art performance, the VHDL code can be used to manufacture a single IC. This will include both the controller and the ADC, ensuring the amplifier consists of a single IC and an output stage.

Bibliography

[1] L. Risbo, "Discrete-Time Modeling of Continuous-Time Pulse Width Modulator Loops," in *AES 27th Internation Conference*, 2005.

[2] P. Kemp, T. Mouton, and B. Putzeys, "High-Order Analog Control of a Clocked Feedback Using Z-Domain Methods," in *Audio Engineering Society Convention 131*, 2011.

[3] F. Koeslag, "A detailed analysis of the imperfections of pulsewidth modulated waveforms on the output stage of a class D audio amplifier," PhD dissertation, Stellenbosch University, 2008.

[4] T. Mouton and B. Putzeys, "Digital control of a PWM switching amplifier with global feedback," in *AES 37th International Conference*, 2009.

[5] P. S. Kemp, "The Design of an Analogue Class-D Audio Amplifier using Z-Domain Methods," MSc, Stellenbosch University, 2012.

[6] K. C. Pohlmann, *Principles of Digital Audio*, 4th ed. McGraw Hill, 2000.

[7] N. Mohan, T. Undeland, and W. Robbins, *Power Electronics: Converters, Applications and Design*, 3rd ed., B. Zobrist, Ed. John Wiley& Sons, Inc, 2003.

[8] L. Risbo, M. Høyerby, and M. Andersen, "A versatile discrete-time approach for modeling switch-mode controllers," in *PESC Record - IEEE Annual Power Electronics Specialists Conference*, 2008, pp. 1008–1014.

[9] M. Gopal, *Digital Control and State Variable Methods*, 2nd ed. McGraw Hill, 2004.

[10] J. Proakis and D. Monolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 4th ed. Prentice Hall, 2007.

[11] L. Risbo and C. Neesgard, "PWM Amplifier Control Loops with Minimum Aliasing Distortion," in *Audio Engineering Society Convention 120*, 2006.

[12] B. H. Candy and S. M. Cox, "Improved Analogue Class-D Amplifier with Carrier Symmetry Modulation," in *Audio Engineering Society Convention 117*, 2004.

[13] B. Putzeys, "Simple, Ultralow-Distortion Digital Pulse Width Modulator," in *Audio Engineering Society Convention 120*, 2006.

[14] C. Phillips and T. Nagle, *Digital Control System Analysis and Design*, 3rd ed. Prentice Hall, 1995.

[15] S. Norsworthy, R. Schreier, and G. Tmes, *Delta-Sigma Data Converters*. A John Wiley & Sons, 1997.

[16] P. Midya, B. Roeckner, and T. Paulo, "Digital control of a PWM switching amplifier with global feedback." in *AES 37th International Conference*, 2009.

[17] G. Goodwin, S. Graebe, and M. Salgado, *Control System Design*. Prentice-Hall, 2000.

[18] B. Putzeys, "Self-Oscillating class D amplification device, European Patent," 2010.

[19] M. Gerzon and P. Craven, "Optimal noise shaping and dither of digital signals," in *Audio Engineering Society Convention 87*, 1989.

[20] B. Putzeys, "Personal Communication via e-mail," 2016.

[21] Analog Devices, "AD9512 Data sheet." [Online]. Available: www.analog.com/media/en/technical-documentation/data-sheets/AD9512.pdf

[22] Altera, "Cyclone III Device Handbook Volume 2." [Online]. Available: www.altera.com/content/dam/altera-www/global

[23] T. Niesler, "Digital Signal Processing : Signals and signal analysis," 2014.

[24] Analog Devices, "AD7760 Data Sheet." [Online]. Available: www.analog.com/media/en/technical-documentation/data-sheets/AD7760.pdf

Appendices

MATLAB design scripts

Basic System Initialisation

```
s = tf('s');
%  Output stage
Lo = 20.83e-6;
Co = 2.04e-6;
Ro = 8.2;
Vd = 15;
Tdelay = 650e-9;
Kss = 1.679;

% Output filter model
Hfilter = (1/(Lo*Co))/(s^2 + s/(Ro*Co) + 1/(Lo*Co));
Hfilterz = c2d(Hfilter,(1/19.6608e6));
[numest, denest] = tfdata(Hfilterz,'v');

% LC Pole Cancellation
CancFil = 6295.0713 *(s^2 + 2.83e005*s + 2.127e010)
        / (s^2 + 2.362e007*s + 1.395e014);
CancFilz = c2d(CancFil,(1/19.6608e6));
[numCFil, denCFil] = tfdata(CancFilz,'v');

% ADC low-pass filter
ADLPFs = 0.0014332* (s^2 + 3.822e008*s + 3.655e016)
       / (s^2 + 5.503e006*s + 5.221e013);
ADLPFz = c2d(ADLPFs,(1/19.6608e6));
[numADL, denADL] = tfdata(ADLPFz,'v');

Sisteem = Hfilter*Vd;
Gain = 68;
Total = Sisteem*Gmanual*ADLPFs*CancFil*Gain*Kss;

% Impulse invariance transform
Ts = 1/(768e3);
```

```
[num, den] = tfdata(Total,'v');
[r, p, k] = residue(num, den);

pdel = exp(-p*Tdelay);
pd = exp(p*Ts);
rd = Ts*pd.*r.*pdel;

[numz, denz] = residue(rd, pd, 0);
Gz = tf(real(numz),real(denz),Ts);
```

Controller design

```
clear a1 a2 a3 a4 a5 a1r a2r a3r a4r a5r;
Ts = 1/(19.6608e6);
z = tf('z',Ts);
s = tf('s');

poolfrek1 = 10.6e3;
poolfrek2 = 17.8e3;

gamma1 = (2*pi*poolfrek1)^2;
gamma2 = (2*pi*poolfrek2)^2;

% s-domain zero positions
zero1 = -1.3814e005 + 1.1289e005*j;
zero2 = -1.3814e005 - 1.1289e005*j;
zero3 = -2.2498e005 + 1.4999e005*j;
zero4 = -2.2498e005 - 1.4999e005*j;

% determine equation for desired zero positions
WantedEq = (s - zero1)*(s - zero2)*(s  - zero3)*(s- zero4);
[num3, den3] = tfdata(WantedEq,'v');

coef1s = num3(1);
coef2s = num3(2);
coef3s = num3(3);
coef4s = num3(4);
coef5s = num3(5);

a1 = 1;
a2 = coef2s*a1;
a3 = (coef3s - gamma1 - gamma2)*a1;
a4 = (coef4s - (a2/a1*gamma2))*a1;
```

```
a5 = (coef5s - (gamma1*gamma2) - (a3/a1*gamma2))*a1;

k1c = s^2 + gamma1;
k2c = s^2 + gamma2;

% s-domain controller transfer function
Gmanual = (a1*k1c*k2c + a2*s*k2c + a3*k2c + a4*s + a5)/(s*k1c*k2c);

% Determine desired z-domain controller transfer function
Gcontrol = c2d(Gmanual*Gain,(1/(19.6608e6)));
[num2, den2] = tfdata(Gcontrol,'v');

% Placing z-domain poles
gamma1z = 1.147515625e-005;
gamma2z = 3.235903225e-005;

k1 = (z-1)^2 + gamma1z;
k2 = (z-1)^2 + gamma2z;

coef1z = num2(2);
coef2z = num2(3);
coef3z = num2(4);
coef4z = num2(5);
coef5z = num2(6);

% determine coefficients to implement controller in chain of integrator form
a1r = coef1z;
a2r = coef2z + 4*a1r;
a3r = coef3z - a1r*(6 + gamma1z + gamma2z) + 3*a2r;
a4r = coef4z - a1r*(-4-2*gamma1z - 2*gamma2z) - a2r*(3 + gamma2z) + 2*a3r;
a5r = coef5z - a1r*(1 + gamma1z + gamma2z + gamma1z*gamma2z)
      - a2r*(-1 - gamma2z) - a3r*(1 + gamma2z) + a4r;

% verify calculation of coefficients
Gzman = (a1r*k1*k2 + a2r*(z-1)*k2 + a3r*k2 + a4r*(z-1) + a5r)/((z-1)*k1*k2);
[num1, den1] = tfdata(Gzman,'v');
```

PWM input gradient calculation

```
s= tf('s');
Ts = 1/(768e3);
A = zeros(1,98304);
B = zeros(1,98304);
```

```
% System definition
Gtot = Hfilter*Gmanual*CancFil*ADLPFs*Vd*Kss;

% Obtain data from TF
[num1, den1] = tfdata(Gtot,'v');
% Expand into partial fractions
[r1, p1, k1] = residue(num1, den1);

yuit = 0;
yafuit = 0;
k = r1;
a = p1;
kint = k(11);
i = length(a);
% Integration constant of output
c = (2*k)./(a.*(exp(-a.*Ts/2)-exp(a.*Ts/2)));
% Integration constant of gradient
f = (2*k)./(exp(-a.*Ts/2)-exp(a.*Ts/2));

 % Time vector
t = (-Ts/2):1/(98.304e6):(Ts/2-1/(98.304e6));

for n = 1:i-1
y = c(n)*exp(a(n)*t) + (2*k(n)*t)/(a(n)*Ts)+ (2*k(n))/(a(n)^2*Ts) ;
yuit = yuit + y;
yaf = f(n)*exp(a(n)*t) + 2*k(n)/(a(n)*Ts);
yafuit = yafuit + yaf;
end

yint = -kint*t.^2/Ts;
yafint = -2*t.*kint/Ts;
yuit = yuit + yint;
yafuit = yafuit + yafint;

for r = 0:767
for p = 1:128
A(128*r+p) = yuit(p);
B(128*r+p) = yafuit(p);
C(128*r+p) = yafuit(69);
end
end
```