

High Resolution Readout of Vibrating Island Accelerometer Using Digital Receiver

by

Lawrence Kon-Luen Chang

Submitted to the Department of Electrical Engineering
and Computer Science in partial fulfillment of the require-
ments for the degree of

Masters of Engineering in Electrical Engineering and
Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 26, 1998

© Lawrence K. Chang, 1998. All Rights Reserved.

The author hereby grants to MIT
permission to reproduce and to
distribute publicly paper and
electronic copies of this
document in whole or in part.

Author

✓ / Electrical Engineering and Computer Science
May 26, 1998

Certified by

Richard Gardner
Technical Supervisor, Draper Laboratory

Certified by

✓ Professor James L. Kirtley Jr.
Thesis Supervisor

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY Accepted by

✓ Arthur C. Smith

Chairman, Department Committee on Graduate Theses

JUL 14 1998

LIBRARIES

Eng

THE UNIVERSITY OF CHICAGO
LIBRARY
540 EAST 57TH STREET
CHICAGO, ILL. 60637
TEL: 773-936-3200

High Resolution Readout of Vibrating Island Accelerometer Using Digital Receiver

by

Lawrence Kon-Luen Chang

Submitted to the
Department of Electrical Engineering and Computer Science

May 26, 1998

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and
Computer Science

Abstract

The Vibrating Island Accelerometer (VIA) is targeted for $1\mu\text{g}$ accuracy and requires both a precise transducer element and a high resolution readout scheme. The quartz crystal transducer angle-modulates two stable oscillators, and is analogous to an FM/PM transmitter. The readout scheme can either phase demodulate the oscillator waveforms to yield velocity, or frequency demodulate to read acceleration. First, this thesis derives an equation that relates velocity resolution to acceleration resolution and digital integration error, and then estimates the resolution performance of the analog phase-locked loop, period readout scheme, and a conventional cross-differentiate multiply (CDM) FM demodulator. For a readout time of 360 seconds, the CDM receiver requires a fourth-order delta-sigma modulator to achieve the same velocity resolution as the phase readouts. Second, this thesis designs the quadrature demodulation cross-differentiate multiply (QCDM) digital receiver to readout the VIA. The FM bandwidth required for the VIA to achieve the desired accuracy is found to be twice the Carson Rule estimate and is used to design the FIR filters. It is also discovered that highly accurate numerical integration using Simpson's Rule can be implemented using a polyphase decimator. Using Matlab simulations, for a DSP input SNR of 122 dB, the designed QCDM demodulator achieves $8\mu\text{g}$ resolution.

Thesis Supervisor: James L. Kirtley
Title: Professor

Acknowledgments

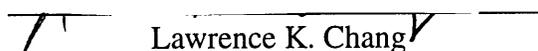
May 26, 1998

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under

IR&D 13315.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Permission is hereby granted by the Author to the Massachusetts Institute of Technology to reproduce and distribute publicly paper and electronic copies of this thesis and document in whole or in part, and to grant others the right to do so.

 Lawrence K. Chang

I would like to thank Dick Gardner for the opportunities and flexibility and support that he has given me during my time at Draper.

Thanks to Jim Scholten, Mike Luniewicz, Paul Ward, Steve Finberg, Ed Foster, Pete Welling, Carlo Venditti, Ken Kaiser, and everyone in E31 and on the VIA project who has helped me out along the way.

Thanks to Professor Kirtley for advising and for reviewing the drafts, and thanks to Vivian Mizuno for all of her help and cheerfulness.

Special thanks go out to Bob DiMatteo for always giving insightful advice, Jeanette Cahill for always having answers for my questions, Jake Morzinski for the Frame help, Ien Cheng for the vitamin C, Shan Wang for all of the lunches, my roommate, Vinay Mohta, for the papers, the textbooks, and the thesis discussions, and CF'ers for their spiritual support.

Thanks to my father, Peter, and my mother, Johanna, for everything, especially for putting me through MIT and their love. Thanks to Mary, Margaret, and Stephen for their continued encouragement over the years.

Table of Contents

1	Introduction	11
2	Vibrating Island Accelerometer Background	15
3	Comparison of Velocity and Acceleration Readout Schemes	23
3.1	Relationship Between Acceleration and Velocity Resolution	24
3.1.1	Digital Integration Error	25
3.1.2	Amplitude Quantization Error	29
3.2	Velocity Readout Schemes	31
3.2.1	Analog Phase-Locked Loop	32
3.2.2	Period Readout Scheme	34
3.3	Motivations for Reading Acceleration	36
3.3.1	Signal Bandwidth and Processing Speed Considerations	37
3.3.2	Utilizing Amplitude Information	42
3.4	Acceleration Readouts	43
3.4.1	Four Digital FM Demodulators	43
3.4.2	CDM Demodulation with Traditional A/D Conversion	45
3.4.3	Oversampling Delta-Sigma Modulator A/D Conversion	54
3.5	Summary	61
4	Acceleration Readout Design and Simulation	63
4.1	System Description	64
4.1.1	Functional Description	64
4.1.2	Detailed Description	68
4.2	Subsystem Design and Simulation	71
4.2.1	Discrete-Time Differentiator Design	71
4.2.2	Cross-Differentiation Multiply Structure	83
4.2.3	FM Bandwidth Requirements	87
4.2.4	Low Pass Filter Design	95
4.2.5	Numerical Integration	101
4.3	System Simulation	109
4.4	Further Tests and Design Directions	122
5	Conclusion	125
Appendix A Simulations		129
A.1	SIM1	129
A.2	SIM2	131
A.3	SIM3	133
A.4	SIM4	135
A.5	SIM5	137
A.6	SIM6	139
A.7	SIM7	141
A.8	SIM8	142
A.9	SIM9	145
A.10	QCDM	146

List of Figures

Figure 2.1: Draper Laboratory VI-A Quartz Crystal Design.....	16
Figure 2.2: Block diagram of Vibrating Island Accelerometer.	18
Figure 3.1: Digital integration error for a 100-Hz sinewave sampled at 1-kHz. (a) Continuous-time acceleration. (b) Continuous-time integrated velocity. (c) Sampled acceleration. (d) Sampled velocity. (e) Digitally integrated acceleration. (f) Digital integration error.	27
Figure 3.2: Probability density functions of random variables (a) a and (b) v	29
Figure 3.3: Analog phase-locked loop readout.	33
Figure 3.4: (a) Period readout scheme (b) Period readout with capacitor interpolation.	35
Figure 3.5: Spectrum of $s_1(t)$ for $f_1=10$ kHz and maximum acceleration of (a) 1 g, (b) 10 g, and (c) 20 g.	38
Figure 3.6: Beating down oscillator signals. (a) Multiplying each input by a Local Oscillator (LO). (b) Multiplying two inputs together for common mode reduction.	40
Figure 3.7: Simplified CDM Frequency Demodulator.....	45
Figure 3.8: Eigenfunction and eigenvalues of LTI systems.	47
Figure 3.9: Discrete-Time Differentiator.	49
Figure 3.10: Error model of simplified CDM frequency demodulation method.	50
Figure 3.11: Acceleration resolution versus number of A/D bits from (3.34).....	53
Figure 3.12: (a) Traditional A/D converter and (b) delta-sigma A/D converter.....	55
Figure 3.13: First-order sigma delta modulator.	56
Figure 3.14: Second-order sigma-delta modulator.	58
Figure 3.15: Quantization noise shaping. Solid line represents no noise shaping. Dashed-dotted line represents first-order noise shaping. Dashed line is second-order shaping.	59
Figure 4.1: Functional block diagram of readout scheme.....	65
Figure 4.2: Detailed block diagram of digital QCDM demodulator readout.....	69
Figure 4.3: (a) Direct form, (b) transposed direct form realization of an FIR system....	74
Figure 4.4: Low pass filter design specifications.....	77
Figure 4.5: Differentiator block.	78
Figure 4.6: Input $i(t)$ and its magnitude response.	79
Figure 4.7: Differentiator characteristics for $M=26$	80
Figure 4.8: Differentiated $i(t)$	82
Figure 4.9: Amplitude error introduced by differentiator.....	82
Figure 4.10: Cross-Differentiate Multiply demodulator block diagram.....	84
Figure 4.11: Ideal acceleration output for (4.23) and (4.24).....	85
Figure 4.12: Actual acceleration output for (4.23) and (4.24).	86
Figure 4.13: Acceleration output error.....	86
Figure 4.14: Input ideally-bandlimited to the Carson Limit. $N_i=6$, $f_a=100$	91
Figure 4.15: Acceleration error for Carson Limit bandwidth.	91
Figure 4.16: Input ideally bandlimited to 2400 Hz. $N_i=12$, $f_a=100$	92
Figure 4.17: Acceleration error for bandwidth of 2400 Hz.	92
Figure 4.18: FMBR relationship for the CDM demodulator.	94
Figure 4.19: Low pass filter blocks cascaded with CDM demodulator.....	95
Figure 4.20: Order-45 low pass filter.....	97
Figure 4.21: Distortion of $i(t)$ due to low pass filter.	98

Figure 4.22: Acceleration error for $M=26$, and order-45 low pass filter.	99
Figure 4.23: Narrowband filter hbnf of order 65.	100
Figure 4.24: Distortion of acceleration signal due to narrowband filter.	101
Figure 4.25: Acceleration error after narrowband-filtering stage.	102
Figure 4.26: Acceleration output after narrowband-filtering stage.	102
Figure 4.27: Integrator block	105
Figure 4.28: Digital filter implementation of numerical integration methods.	106
Figure 4.29: Simplified block diagrams.	106
Figure 4.30: Maximum accumulator error.	107
Figure 4.31: Maximum bilinear transform filter error.	108
Figure 4.32: Maximum polyphase decimator filter error.	108
Figure 4.33: M-filter polyphase decimator.	109
Figure 4.34: System simulated with QCDM.	111
Figure 4.35: Simulation with QCDM for $t=10$ seconds, input='con', and $Q=1e-6$	114
Figure 4.36: Simulation with QCDM for $t=10$ seconds, input='con', and $Q=1e-6$	114
Figure 4.37: Simulation of QCDM with $t=10$ seconds, input='con', and $Q=1e-6$	115
Figure 4.38: Simulation with QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$	115
Figure 4.39: Simulation of QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$	116
Figure 4.40: Simulation of QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$	116
Figure 4.41: Simulation with QCDM for $t=1$ second, input='cos', and $Q=1e-1$	121
Figure 4.42: Simulation with QCDM for $t=1$ second, input='cos', and $Q=1e-1$	121

List of Tables

Table 2.1: Vibrating Island Accelerometer parameters	20
Table 3.1: Theoretical Delta Sigma Modulator Parameter Values	60
Table 4.1: SIM1(M) results for M=22, 26, and 28	83
Table 4.2: SIM2(M) results for M=22, M=26, and M=28.....	87
Table 4.3: SIM4(M,NSB) for M=22, 26, 28, and NSB=18	93
Table 4.4: SIM6(M,Fp,Fs,Dp,Ds) for tests 1., 2., and 3	98
Table 4.5: Simulation Results from QCDM	118
Table 4.6: Simulation Results from QCDM (continued).....	119

Chapter 1

Introduction

analogy *n., pl. -gies*. 1. Correspondence in some respects between things otherwise dissimilar. 2. An inference that if two things are alike in some respects they must be alike in others [20].

The spark that ignites both the technical and the artistic imagination is often a simple analogy. When applied to solving an engineering problem, an analogy that highlights correspondence between the problem at hand and other problems, can infer possible solutions that otherwise might never have been considered. The exploitation of an analogy is at the origin of this thesis, which began with the observation that the quartz crystal transducer of the Vibrating Island Accelerometer behaves much like a frequency modulation (FM) transmitter in an analog communication system.

Accelerometers are inertial instruments that are used to guide aircraft, satellites, missiles, and unmanned vehicles, such as autonomous submarines. By integrating acceleration measurements over time, an Inertial Navigation System (INS) can determine instantaneous velocity and, after a second integration, position information. The accuracy of this position information depends, in part, on the bias stability of the accelerometer, with smaller bias stabilities reflecting smaller measured position error and thus better performance. Commercial applications require bias stability's of about 10,000 μ g, aircraft demand 100 μ g, and strategic missile applications need approximately 1 μ g bias stability [21].

For strategic applications, the best accelerometer to date has been the Pendulous Integrating Gyro Accelerometer (PIGA), an electromechanical design. However, since PIGA is very costly to build and to maintain, there has been an interest in looking at solid state

devices that promise to be smaller, cheaper, and easier to manufacture. Among the emerging technologies are silicon micromechanical accelerometers and the Vibrating Island Accelerometer [21].

At the heart of the Vibrating Island Accelerometer (VIA) is a quartz crystal that is the resonant frequency determiner of two oscillator circuits. Crystal oscillators are often used as timing devices, and are generally designed to be insensitive to external parameters such as temperature and force. The VIA crystal, on the other hand, is designed to be as sensitive to force, and thus acceleration, as possible. Therefore, in the midst of an acceleration field, the VIA oscillators will modulate their frequencies by an amount proportional to the magnitude of the acceleration. The crystal for this device is expected to linearly relate acceleration to frequency change, and thus can be viewed as an acceleration-controlled oscillator. Much like a voltage-controlled oscillator in a simple radio transmitter, the VIA transducer generates an FM modulated signal.

One half of the VIA is the quartz crystal transducer and its associated electronics. The second half of the instrument must extract acceleration information from the angle-modulated oscillator output. The portion of the inertial instrument that achieves this is called the *readout* of the accelerometer. Not simply a digital or liquid crystal display of some sort, as the name might suggest, the readout scheme is analogous to a radio receiver that demodulates an FM signal. In fact, for this particular device, the readout scheme can be implemented exactly as an FM demodulator or receiver.

The greatest challenge in designing the readout electronics for the VIA is to meet the high resolution requirements that it faces. Intended to be used for strategic military applications, the instrument is desired to have an acceleration resolution that would rank it among the most highly accurate in the world. The repercussions of this system-level design goal affect even the most detailed levels of design, for to achieve high resolution

collectively, each component in the accelerometer must introduce extremely small amounts of error to the acceleration signals. Therefore, not only must the VIA quartz crystal oscillators have extremely high stability waveforms (1 part in 10^{11}), but the readout scheme must not contaminate the acceleration signal while extracting it from the transducer.

It is the high resolution requirement of the accelerometer that motivates making the analogy between the accelerometer and a communication system. The hope is that in the vast, mature field of communications, where frequency modulation is a ubiquitous technique, there exists technology that can lend insight into finding a solution to the high resolution readout problem. The analogy leads naturally to three questions that this thesis will attempt to answer.

1. *What insight to the general nature of the problem can the analogy lend?* For example, communication theory should be able to reveal how the bandwidth of an FM signal translates into the signal-to-noise ratio of a demodulated output.

2. *Can a digital FM receiver theoretically achieve the resolution requirements of the VIA?* Despite the analogy relating the accelerometer to a communications system, a number of differences do exist, nevertheless, which might preclude an exact transfer of technology from one field to the other. In addition, although there may not be an off-the-shelf digital receiver that can be used directly to achieve the resolution requirements of the VIA, perhaps there are various emerging subsystem communication technologies that might be useful components in the readout scheme.

3. *What motivations are there for researching the implementation of a frequency demodulator over other schemes?* Other readouts, such as phase readout and period readout methods, exist for other accelerometers. How might they perform for this application?

Chapter 2 will give an overview of the VIA project in greater detail, constructing the communication-system analogy more clearly. The quartz crystal design will be shown, but not discussed in detail. The expected system specifications will be disclosed and defined, as well.

Chapter 3 will address Question 3 by quantifying the inherent challenges to implementing a frequency readout. It will then review two common phase readout methods proposed to demodulate a similar accelerometer, deriving expressions to estimate their theoretical performance. Next, consideration of frequency readout methods will discuss a few possible advantages to performing frequency demodulation. A specific FM demodulator, called the Cross-Differentiate Multiply demodulator, will then be compared to the aforementioned phase readouts. The thesis will not, however, perform a more detailed comparison between phase and frequency readout schemes.

Detailed analysis of the Quadrature Cross-Differentiate Multiply demodulator will take place in Chapter 4, seeking to answer Question 2 through the design and simulation of a digital FM receiver that can achieve the desired resolution requirements of the VIA. A detailed subsystem design and simulation with Matlab 5.1 is followed by a system simulation that studies the performance of the system in the presence of noise. Two of the subsystem analyses, in particular, look at the necessary FM bandwidth requirements to achieve high resolution, and the implementation of numerical integration methods using digital filters. The simulations are described in detail, and results are reported to show that the resolution can in fact be achieved in theory.

All the while, throughout this thesis, the answer to Question 1 is addressed. For by the end of this text, the reader will hopefully have observed and learned about the fruits of the central analogy that drives this thesis forward.

Chapter 2

Vibrating Island Accelerometer Background

The existence of quartz crystal accelerometers date back to at least as early as the 1960's, when IBM designed a unit targeted for inertial instrument applications. The most important advantages of these devices have traditionally been their implementation of a solid state rather than a mechanical transducer, a readout scheme that does not need an analog-to-digital (A/D) converter, and relatively simple readout electronics. These characteristics enabled the accelerometers to be low-cost, highly reliable devices. Although the feasibility of high resolution applications were cited as early as in 1971, no research has pursued the development of this application. Nor has any research fully explored the radiation hardness potential that crystal quartz accelerometers exhibit, an attribute that would deem them extremely useful for strategic military applications. The Vibrating Island Accelerometer being developed at Draper Laboratory is different from previous quartz crystal accelerometers in that it specifically targets the high resolution, radiation hard application. [30]

Since the high resolution application has never been built before, many uncertainties enshroud the VIA research effort with regards to the feasibility of its goals, including its intended linearity and reduced common mode effects. The efforts of this thesis focus on the exploratory study of a readout scheme that must likely be more complex than previous quartz accelerometers in order to achieve high resolution performance. One consideration, in the following chapters, will mention the reintroduction of an analog-to-digital converter into the readout scheme to improve upon the resolution performance of past devices. Significant advances in oversampling A/D conversion technology in recent years gives promise that this can be done. In parallel with the readout study, a novel transducer

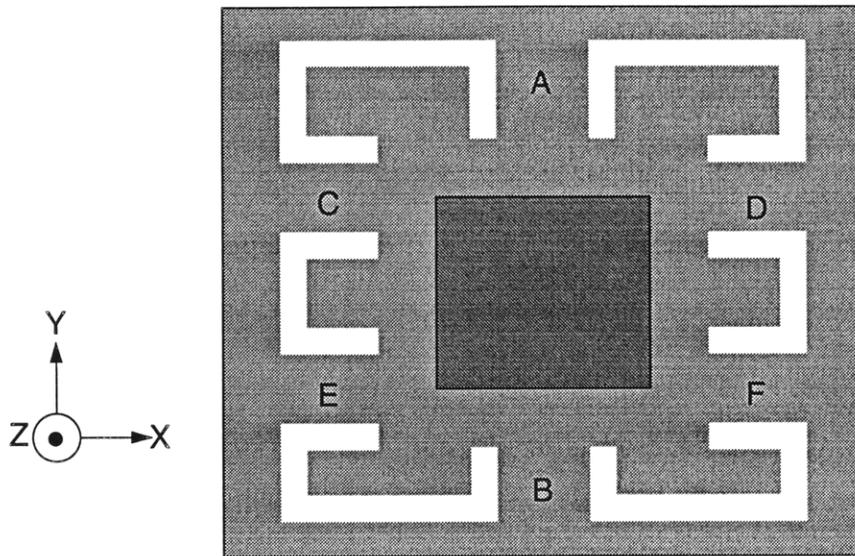


Figure 2.1: Draper Laboratory VI-A Quartz Crystal Design is being designed and analyzed that implements an AT crystal cut. The preliminary design of the quartz crystal is shown in Fig. 2.1.

Electrodes sandwich the crystal at points A and B, with each sandwich connected to a separate oscillator circuit. Points A, B, C, D, E, and F are joints that suspend the proof mass, located at the center of the crystal like an “island,” from the rest of the quartz. Acceleration is measured by taking the difference between the frequencies of the two oscillator circuits. When the instrument is accelerated in the X or Z direction, the joints A and B are tensed and stressed symmetrically so there should be no difference in the frequencies of oscillation. However, when the quartz is accelerated in the Y direction, points A and B are stressed differently, resulting in disparate operating frequencies from which a difference is obtained. This difference indicates the acceleration along the inertial axis of the instrument, which is the Y-axis in this case.

Actual implementation of the device will require that the zero g resonant frequencies of oscillator A and B be offset to avoid nonlinear performance that might occur near zero g, when the oscillators may tend to synchronize. In addition, by simply taking the difference between oscillator frequencies centered around identical carriers, it is impossible to distinguish between positive and negative acceleration along the inertial axis. Chapter 4, however, will show that quadrature demodulation can achieve sign resolution. The advantage of running two oscillators from the same crystal is that external parameters, such as temperature fluctuation and acceleration along the X and Z axes, should affect the oscillator signals in a similar fashion. The common modes from these errors are expected to cancel one another when differencing and demodulating the two oscillator signals.

The oscillator circuits themselves have a modified-Meacham structure and are implemented with high bandwidth operational amplifiers. An automatic gain control circuit is required to stabilize the amplitude of the oscillator, which impacts the stability of the oscillator frequency. As mentioned earlier, the frequency of the oscillators will be directly related to the sensed acceleration. Thus, the transducers are basically acceleration-controlled oscillators, and like voltage-controlled oscillators, what they produce are frequency modulated signals. However, frequency modulation (FM) and phase modulation (PM), although two distinct schemes of angle modulation, are closely related by an integration [3]. In defining frequency to be analogous to acceleration, then accordingly, in a linear system, phase will be analogous to velocity. So the outputs of the oscillators can be viewed as two angle modulated waves, either frequency modulated by acceleration, or phase modulated by velocity. The mathematical model of these signals will be described shortly. The functionality of the VIA oscillator can be modeled as in the block diagram of Fig. 2.2, and is analogous to a transmitter in a communication system.

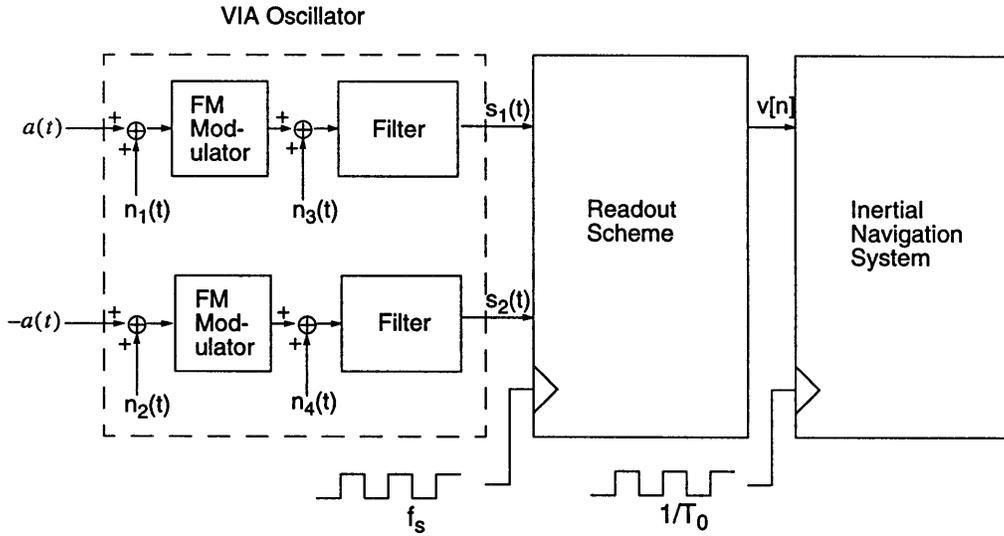


Figure 2.2: Block diagram of Vibrating Island Accelerometer.

It is important to accurately model the noise characteristics of an oscillator because the noise levels primarily dictate its stability. Different types of noise include random walk of frequency, flicker of frequency, white frequency, flicker of phase, and white of phase [22]. The sources of the noise include the quartz crystal transducer itself, the oscillator electronics, and external noise such as Johnson thermal noise. It is important to categorize the noise of an oscillator as either modulation, such as $n_1(t)$ and $n_2(t)$, or additive carrier, such as $n_3(t)$ and $n_4(t)$ because FM demodulators can shape the noise added to the carrier signal (Section 4.3). A detailed noise model of the VIA oscillator has not been constructed yet. The simulations in Chapter 4 will model the noise as zero-mean, independent, and wide-sense stationary.

While the input to the readout scheme consists of two analog angle-modulated waveforms, the output is expected to be a digital word representing velocity, $v[n]$, pro-

duced at a rate specified by the Inertial Navigation System (INS) (Fig. 2.2). The resolution of the readout is the value, in units of velocity, assigned to one quantum of the accelerometer's discretized output. The value of a single quantum is also related to the measure of the peak magnitude of noise, or error, that has accumulated through the system in one period of measurement. For an average acceleration error of $e_a(t)$ over time t , the resulting average velocity error due to $e_a(t)$ is

$$e_v(t) = e_a(t) \cdot t . \quad (2.1)$$

The corresponding position error is

$$e_p(t) = \frac{1}{2} e_a(t) t^2 = \frac{1}{2} e_v(t) t . \quad (2.2)$$

Therefore, for a position error of $e_p(360)=3.6$ ft for a flight of $t=360$ seconds, the velocity error is

$$e_v(t) = 2 \cdot e_p(t) \cdot \frac{1}{t} \quad (2.3)$$

$$e_v(360) = 2 \cdot 3.6 \cdot \frac{1}{360} = 0.02 \frac{ft}{sec} .$$

The acceleration error is

$$e_a(t) = e_v(t) \cdot \frac{1}{t} \quad (2.4)$$

$$e_a(360) = 0.02 \frac{ft}{sec} \cdot \frac{1}{360} \approx 5.56 \times 10^{-5} \frac{ft}{sec^2} \approx 1 \mu g .$$

These first order calculations are the basis for the first two fields of the following VIA accelerometer specifications. A similar derivation will be treated in more detail in Section 3.1.2. The oscillator frequency and scale factor are parameters of the quartz crystal. The range and bandwidth of the measured acceleration are estimates. The INS data rate is an approximation based on systems in use at Draper.

Table 2.1: Vibrating Island Accelerometer parameters.

Acceleration resolution (Δa)	1 μg
Velocity resolution (Δv)	0.02 ft/sec
Oscillator carrier frequencies (f_1, f_2)	24 MHz
Scale factor (k_1, k_2)	50 Hz/g
Bandwidth of acceleration signal (f_a)	100 Hz
Range of measured acceleration	0-20 g
INS required data rate ($1/T_0$)	500 Hz

Modeling the oscillator outputs as angle-modulated waves, they can be described as follows:

$$s_1(t) = S_1 \cos\left(2\pi f_1 t + 2\pi k_1 \int_0^t a(t) dt\right) = S_1 \cos(2\pi f_1 t + 2\pi k_1 v(t)) = S_1 \cos\phi_1(t) \quad (2.5)$$

$$s_2(t) = S_2 \cos\left(2\pi f_2 t - 2\pi k_2 \int_0^t a(t) dt\right) = S_2 \cos(2\pi f_2 t - 2\pi k_2 v(t)) = S_2 \cos\phi_2(t). \quad (2.6)$$

In these equations, the analogy between an FM transmitter and the accelerometer has been decomposed into an analogy between acceleration, $a(t)$, and frequency, velocity, $v(t)$, and phase. Signals $s_1(t)$ and $s_2(t)$ can be described as either phase-modulated by $v(t)$ or frequency-modulated by $a(t)$. Throughout this thesis, from this point onward, "phase readout" and "velocity readout" will be synonymous, as will "frequency readout" and "acceleration readout."

Using values from Table 2.1, the frequency stability requirement of the oscillator can be derived as follows:

$$stability = \frac{\Delta a \cdot k_1}{f_1} = \frac{(1\mu g)\left(50\frac{Hz}{g}\right)}{24MHz} = 2.08 \times 10^{-12}. \quad (2.7)$$

The signal-to-noise ratio of the oscillator output that corresponds to this level of stability, can be calculated from the following equation that approximates FM demodulator error as a function of carrier noise [17]:

$$SNR_{in} = \frac{4\pi^2 f_a^3}{3k_1^2 \cdot \Delta a^2} = \frac{4\pi^2 (100)^3}{3(50)^2 (1 \times 10^{-6})^2} = 5.26 \times 10^{15} \quad (2.8)$$

$$SNR_{in} = 10 \log_{10}(5.26 \times 10^{15}) = 157 \text{ dB}.$$

The largest desired signal-to-noise ratio of the acceleration readout corresponds to the maximum acceleration signal and acceleration resolution specified:

$$SNR_{out} = \frac{20g}{1\mu g} = 2 \times 10^7 \quad (2.9)$$

$$SNR_{out} = 20 \log_{10}(2 \times 10^7) = 146 \text{ dB}.$$

Equations 2.5-2.8 characterize what the readout scheme receives as input from the oscillator. The goal of the readout design is to extract high resolution velocity information from these inputs. Therefore, the readout could either frequency demodulate $s_1(t)$ and $s_2(t)$ and integrate the resulting acceleration, or it could phase demodulate $s_1(t)$ and $s_2(t)$ to get velocity directly. (Note that despite whether the demodulator directly reads acceleration or velocity, the instrument is still called an “accelerometer” since the quartz crystal transducer can measure both.) The challenge of the readout design is to meet the velocity and acceleration resolution requirements, which translates to achieving (2.9) for the acceleration reading. Different readout methods vary in how accurately and efficiently they can produce the measurements. Chapter 3 will discuss five different schemes and Chapter 4 will design and analyze the Quadrature Cross-Differentiate Multiply FM demodulator.

Having drawn a parallel between the VIA readout scheme and an analog communication receiver, a few words should be said about the possibility of borrowing techniques from digital communication schemes. Although the input and output of a digital commu-

nication system is a digital signal, usually a bit-stream, it must be transformed into an analog waveform before it can be transmitted over a real-world channel, which is inherently analog. The digital receiver must filter and resample the received waveform after the symbol timing is recovered, and identify the detected symbol. Digital systems are measured by how closely they can approach the Shannon Limit [23], which provides an upper bound to the quantity of information that can be transmitted over a channel having a given signal-to-noise ratio. High resolution digital systems that approach this limit exploit channel coding schemes for which the digital receiver has specific knowledge of the transmitter encoding technique. For this particular accelerometer application, the transmitter is inherently an analog modulator. Therefore, while the design of the receiver can be optimized by considering as much knowledge of the transmitter as possible, since the VIA oscillator does not implement any convolutional coding, viewing the oscillator as a digital system transmitter, like an M-ary Phase Shift Keying (MPSK) device, gives little if no advantage to viewing the oscillator as an analog frequency modulator. The important distinction, then, is that the analogy here involves an analog communication system rather than a digital communication system. The proposed QCDM receiver in Chapters 3 and 4 will implement digital signal processing to solve an analog communication problem. The demodulator will be called a digital receiver to imply that it implements digital signal processing, but not to imply that it employs techniques specific to digital communications.

Chapter 3

Comparison of Velocity and Acceleration Readout Schemes

The desire for a single parameter drives the design and function of all navigation systems, inertial or otherwise. Everything else is like an oyster shell or a strand of copper wire, all other quantities can be discarded once the pearl, the electrical signal, or the *position* is obtained. Gyroscopes, accelerometers, the Global Positioning System, an explorer's compass and map, all are used to derive position information for a given time. Therefore, although the ensuing discussion focuses on velocity and acceleration measurement, the context of the problem is that the INS ultimately calculates position information.

Chapter 3 will seek to answer Question 3 from the introduction by comparing common phase readouts to frequency readout methods. First, it will show that for general accelerometer design, digital integration error and the integration of quantization error are penalties of using frequency rather than phase demodulation. Then two phase readout schemes are reviewed to obtain benchmarks against which the feasibility of acceleration readout methods can be evaluated. In Section 3.3, the specific VIA characteristics and design features will be looked at to quantify performance measures of the benchmark phase readout systems. Considering the bandwidth requirements of the VIA, the possibility of heterodyning the oscillator outputs, and making use of oscillator output amplitude information to improve the readout resolution will be shown to be additional incentives to implement something other than a phase readout. In Section 3.4, four possible frequency readouts are reviewed, and the conventional cross-differentiate multiply demodulator is analyzed in greater detail. Using a conventional A/D conversion strategy with the CDM demodulator will be shown to result in very little improvement over the phase readout schemes. However, Section 3.4.3, will show that using delta-sigma modulation with the CDM demodula-

tor and frequency down-conversion provide flexibility in trading off sampling rate for amplitude quantization resolution. For a time of 360 seconds, the frequency readout can attain the same resolution as the velocity readout schemes using only one hundredths of the phase readout sampling rate.

3.1 Relationship Between Acceleration and Velocity Resolution

Most accelerometers and velocimeters have analog transducers that measure acceleration and velocity, quantities which are digitized and then converted into a position measurement by an INS microprocessor. Mathematically, the conversion is accomplished by integrating acceleration over time to get relative velocity, and integrating velocity over time to get relative position. Assuming that an analog transducer element has infinite resolution, the digitizing of the analog signal introduces quantization error. A second source of error, which is dependent on the sampling rate, surfaces in digital integration of the quantized signal. Assuming an accelerometer transducer physically contains quantities of both velocity and acceleration to infinite resolution, and assuming that the same noise power results from quantizing either of the two, digitizing the velocity signal and integrating once to get position will introduce less error than digitizing the acceleration signal and integrating twice. In the former case, the quantization error is digitally integrated once in calculating position, while in the latter case, the quantization error is digitally integrated twice. For these fundamental reasons, without considering implementation issues, reading out velocity directly from a transducer is generally preferred to reading out acceleration. If, however, the acceleration quantization noise power is smaller than the velocity quantization noise power, then the preferred readout method depends on the time period over which readings are taken. If the digital integration of acceleration error over a specified time period is smaller than the direct quantization of velocity, then it is preferable, from a mathematical standpoint, to read acceleration from the transducer.

Since acceleration and velocity can be read directly from the VIA oscillator by either frequency demodulating or phase demodulating, respectively, design of the best readout scheme must select between the two. In order to compare the theoretical performance of each scheme, an analytical expression relating digital integration error, acceleration quantization error, velocity quantization error, and time-of-reading, will now be derived.

3.1.1 Digital Integration Error

The derivation begins by considering an acceleration signal from a general accelerometer transducer:

$$a(t) = A \sin(2\pi f_a t). \quad (3.1)$$

The resulting velocity at time t , assuming $v(0)=0$, can be expressed as the definite integral of the acceleration.

$$v(t) = \int_0^t a(t) dt = -\frac{A}{2\pi f_a} \cos(2\pi f_a t) + \frac{A}{2\pi f_a} \quad (3.2)$$

The continuous-time acceleration and velocity signals are plotted in Fig. 3.1 (a) and (b) for $A=1$, $f_a=100$ -Hz.

Next, the continuous-time definite integral can be expressed as a Riemann sum with infinitely many area terms, and can be rewritten as follow:

$$v(t) = \int_0^t a(t) dt = \lim_{dt \rightarrow 0} \sum_{i=0}^{\frac{t}{dt}} a(i dt) dt = \lim_{T \rightarrow 0} T \sum_{i=0}^{\frac{t}{T}} a(iT), \text{ for } T=dt. \quad (3.3)$$

Assuming no amplitude quantization, sampling both $v(t)$ and $a(t)$ faster than the Nyquist rate ($f_s > 2f_a$), the following relationship between discrete-time acceleration, $a[n]$, and discrete-time velocity, $v[n]$, falls out from (3.3) by setting $t=nT$:

$$v[n] = v(nT) = \lim_{T \rightarrow 0} T \sum_{i=0}^n a[i], \text{ for } T=1/f_s. \quad (3.4)$$

From (3.4), it is apparent that in order for discrete-time accumulation to be equivalent to a continuous-time integration, the sampling period must approach zero and the sampling frequency must approach infinity. Therefore, sampling $a(t)$ at a *finite* frequency, f_s , accumulating the samples over time, and scaling the sum by the sampling period, $T=1/f_s$, will yield a discrete-time representation of velocity, $v_a[n]$, that is *not* equal to $v[n]$, the velocity $v(t)$ sampled at f_s . The two digital representations of velocity differ by a digital integration error.

For example, sampling $a(t)$ from (3.1) and $v(t)$ from (3.2) at a sample rate of $f_s=1$ -kHz ($T=0.001$ -sec), gives $a[n]$ and $v[n]$ as shown in Fig. 3.1 (c) and (d). The dashed lines show $a(t)$ and $v(t)$ as references, and the “staircase” waveforms are the sampled discrete-time signals. Discrete-time integrating $a[n]$, that is, accumulating $a[n]$ and multiplying by T , the discrete-time representation of velocity, $v_a[n]$, results, as shown in Fig. 3.1 (e). Notice that $v[n]$ and $v_a[n]$ are different. The digital integration error, $e_i[n]$, is the difference between them and is plotted in Fig. 3.1 (f).

For a finite sampling frequency, then, equation 3.4 must be adjusted to account for digital integration error, $e_{dti}[n]$:

$$v[n] = e_{dti}[n] + v_a[n], \text{ where} \quad (3.5)$$

$$v_a[n] = \frac{1}{f_s} \sum_{i=0}^n a[i].$$

For a sinusoidal acceleration, the integration error is a function of n , as can be seen in Fig. 3.1 (f), as well as the amplitude of the integrand and the sampling frequency. As the sampling frequency gets larger and larger, the Riemann sum is a better and better approximation for the integral. Approximating the maximum magnitude of the error as,

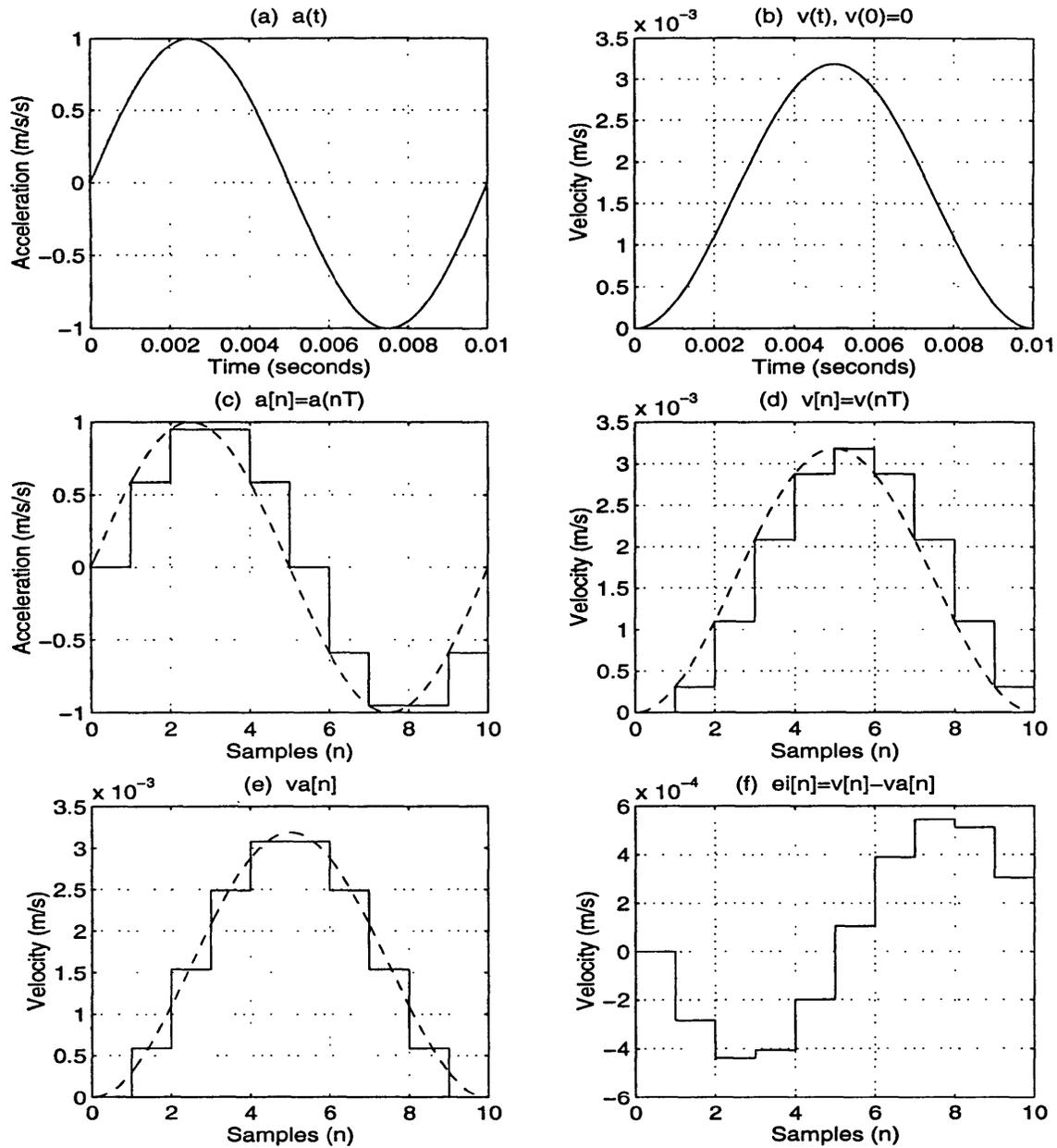


Figure 3.1: Digital integration error for a 100 Hz sinewave sampled at 1 kHz. (a) Continuous-time acceleration. (b) Continuous-time integrated velocity. (c) Sampled acceleration. (d) Sampled velocity. (e) Digitally integrated acceleration. (f) Digital integration error.

$$|e_{di}[n]| \leq \alpha \frac{A}{f_s}, \quad (3.6)$$

numerical analysis (Section 4.2.5, Fig. 4.30) shows that $\alpha \approx 1$ for sinusoidal $a(t)$ with $0 < f_a < 1 \text{ kHz}$. Therefore,

$$|e_{dti}[n]| = |v[n] - v_a[n]| \leq \frac{A}{f_s}. \quad (3.7)$$

Since the acceleration is sinusoidal, note that $e_{dti}[n]$, is periodic with essentially the same period as the acceleration, resulting in two significant implications. First, the maximum value of the digital integration error is independent of time and sample as seen in (3.6). So for larger and larger n , the integration error will be no greater than $\frac{A}{f_s}$. The second implication is that the deterministic qualities of the digital integration error with knowledge of the accuracy of integrand suggests that error correction can improve upon digital integration. Section 4.2.5 will, in fact, show that two other numerical integration methods, the Trapezoidal Rule and Simpson's Rule, produce smaller errors than the Riemann Sum.

What is the digital integration error for an arbitrary bandlimited acceleration signal? To apply the preceding development would require knowledge of the spectrum, the DFT, of the expected acceleration signal. The digital integration error for each discrete frequency could be estimated by (3.6), where $\frac{A}{f_s}$ would be the discrete Fourier series coefficient corresponding to a discrete acceleration frequency, f_a . The errors for each discrete frequency of the DFT could then be summed to get an estimate for the total digital integration error of the bandlimited acceleration signal. Moreover, if each error is in fact sinusoidal with known frequency and amplitude, it is likely that a very good estimate of the error could be obtained. Section 4.2.5 will show how the integration rules can be approximated as digital filters, for which frequency responses could be computed. However, at this point, having obtained a bound on digital integration error using accumulation, attention turns now to considering amplitude quantization error.

3.1.2 Amplitude Quantization Error

In analog to digital conversion, not only is the time axis discretized into periods of time but the amplitude axis is discretized into quanta as well. In the preceding section, discretizing the time axis contributed to the digital integration error in (3.6). This section will reveal how amplitude quantization introduces error.

Continuing to refer to ideal unquantized acceleration and velocity (unquantized in amplitude) as $a[n]=a(nT)$ and $v[n]=v(nT)$, respectively, the following notation is now used to represent amplitude-quantized acceleration for an acceleration readout and amplitude-quantized velocity for a velocity readout:

$$\hat{a}[n] = a[n] + a_n \quad (3.8)$$

$$\hat{v}[n] = v[n] + v_n \quad (3.9)$$

The quantization noise, according to [5], can be denoted by a_n and v_n , samples of independent random variables identically distributed between $\pm\frac{\Delta_a}{2}$ and $\pm\frac{\Delta_v}{2}$, respectively, where Δ_a is one discrete quantum of acceleration and Δ_v is one discrete quantum of velocity. Each quantum is the value assigned to the least significant bit of the digital word. Therefore, Δ_a is acceleration resolution and Δ_v is velocity resolution. The probability density functions of a and v are uniform distributions and are shown in Fig. 3.2.

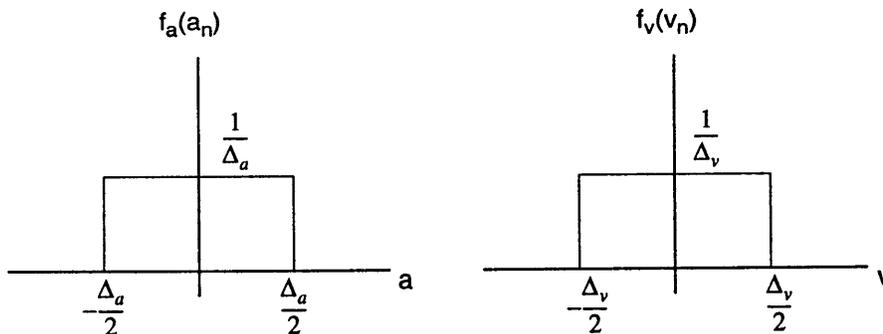


Figure 3.2: Probability density functions of random variables (a) a and (b) v .

The subscript, n , of random variable samples a_n and v_n in (3.8) and (3.9) denote that each sample, n , corresponds to a unique sample of the random variables having experi-

mental values within the distribution of a and v . Therefore, the set or experiment, $\{a_0, a_1, a_2, \dots, a_n\}$, contains $n+1$ samples of random variable a , each having an experimental value within the distribution of a , with each value having no other dependence on n except for distinction.

To compare acceleration resolution and velocity resolution, the accelerometer readout output, $\hat{a}[n]$, must be integrated to yield the following derived estimate of $v[n]$, denoted as $\hat{v}_a[n]$:

$$\hat{v}_a[n] = \frac{1}{f_s} \sum_{i=0}^n \hat{a}[i].$$

By (3.8), this can be rewritten as

$$\hat{v}_a[n] = \frac{1}{f_s} \sum_{i=0}^n a[i] + \frac{1}{f_s} \sum_{i=0}^n a_i.$$

As in (3.5), the first term is the ideal accumulated acceleration. So,

$$\hat{v}_a[n] = v_a[n] + \frac{1}{f_s} \sum_{i=0}^n a_i.$$

Also by (3.5), the derived estimate can be written in terms of the integration error and the ideal velocity:

$$\begin{aligned} \hat{v}_a[n] &= v[n] - e_{dii}[n] + \frac{1}{f_s} \sum_{i=0}^n a_i \\ \hat{v}_a[n] - v[n] &= -e_{dii}[n] + \frac{1}{f_s} \sum_{i=0}^n a_i = e_{va}[n]. \end{aligned} \quad (3.10)$$

Comparing (3.10) to (3.9), it is evident that for an acceleration readout to achieve better resolution than a velocity readout, the following must hold true:

$$\max |e_{va}[n]| = \max \left| -e_{dii}[n] + \frac{1}{f_s} \sum_{i=0}^n a_i \right| < \max |v_n| = \frac{\Delta v}{2}.$$

By the triangle inequality,

$$\left| -e_{dii}[n] + \frac{1}{f_s} \sum_{i=0}^n a_i \right| < |-e_{dii}[n]| + \left| \frac{1}{f_s} \sum_{i=0}^n a_i \right|.$$

Therefore,

$$\max \left| -e_{dti}[n] + \frac{1}{f_s} \sum_{i=0}^n a_i \right| < \max |e_{dti}[n]| + \frac{n\Delta a}{f_s^2}.$$

Since $t = \frac{n}{f_s}$,

$$\max |e_{dti}[n]| + \frac{t\Delta a}{2} < \frac{\Delta v}{2}. \quad (3.11)$$

Equation 3.11 is a fairly simple expression that is a slight modification of (2.1). It is a first order metric by which the resolution of frequency and phase readouts for the VIA can be compared. For an acceleration readout scheme to outperform a velocity readout scheme over a period of time, it must depend on accurate digital integration. The accuracy of integration using an accumulator depends on the sampling rate and acceleration amplitude.

Having obtained the metric in (3.11), the next step is to determine the fundamental limitations on Δv and Δa for various readout schemes. First, Section 3.2 will consider some traditional schemes that have been used or proposed to read phase in the Vibrating Beam Accelerometer and other Quartz Resonant Accelerometers. Then, motivation for implementing a frequency demodulator to read acceleration will be discussed. Lastly, a first-order comparison between phase and frequency demodulation will be derived to demonstrate that in theory, the fundamental limitations of an accelerometer readout warrant its consideration as a viable readout scheme. Note that purely functional analyses are developed here, while a noise model of an acceleration readout will be discussed in the next chapter.

3.2 Velocity Readout Schemes

While a detailed comparison between specific velocity and acceleration readout schemes is beyond the scope of this thesis, taking some time to look briefly at two common phase readout approaches as points of reference will be imperative to understanding the motiva-

tion for designing an acceleration readout.

3.2.1 Analog Phase-Locked Loop

The analog phase-locked loop (PLL) readout [1][2], shown in Fig. 3.3, functions like a standard PLL that accurately tracks the phase of its input. In fact, the PLL functions exactly like PLLs in common timing circuit frequency synthesizers that output high frequency signals coherent to lower frequency inputs. The only difference between frequency multipliers and the analog PLL readout is that the outputs are taken from different points in the loop. Whereas the output of a frequency synthesizer loop is taken from the voltage-controlled oscillator (VCO) output, the phase-locked loop velocity readout is taken from the N-bit counter. Understanding what the N-bits represent is the key to understanding how this readout works.

The N-bit counter in Fig. 3.3 is enabled to count from zero to 2^N continuously during the operation of the device. Each time the counter output reaches 2^N , it resets or overflows and begins counting from zero again. The output is clearly periodic, and in fact can equivalently be viewed as counting from zero to 2π repeatedly. The least significant bit of the N-bit counter represents $2\pi/2^N$ radians, which is the value of one discrete quantum and the resolution of the phase reading. The feedback loop drives the most significant bit (MSB) to follow the input, $s_1(t)$ (as in equation 2.5), so that the error between them is zero. Note that in Fig. 3.3, [N-1] denotes a 1-bit-wide word that is the MSB of the N-bit counter outputs. Any error between the two is integrated by the filter and subsequently adjusts the frequency of the VCO output until phase-lock between the N-bit counter MSB and $s_1(t)$ is regained.

At any time $t=t_0$, the N-bits represent the fraction of the period of the input at time $t=t_0$ relative to $t=0$. In other words, the N-bit counter indicates $[\phi(t_0)\text{mod}(2\pi)]$, the

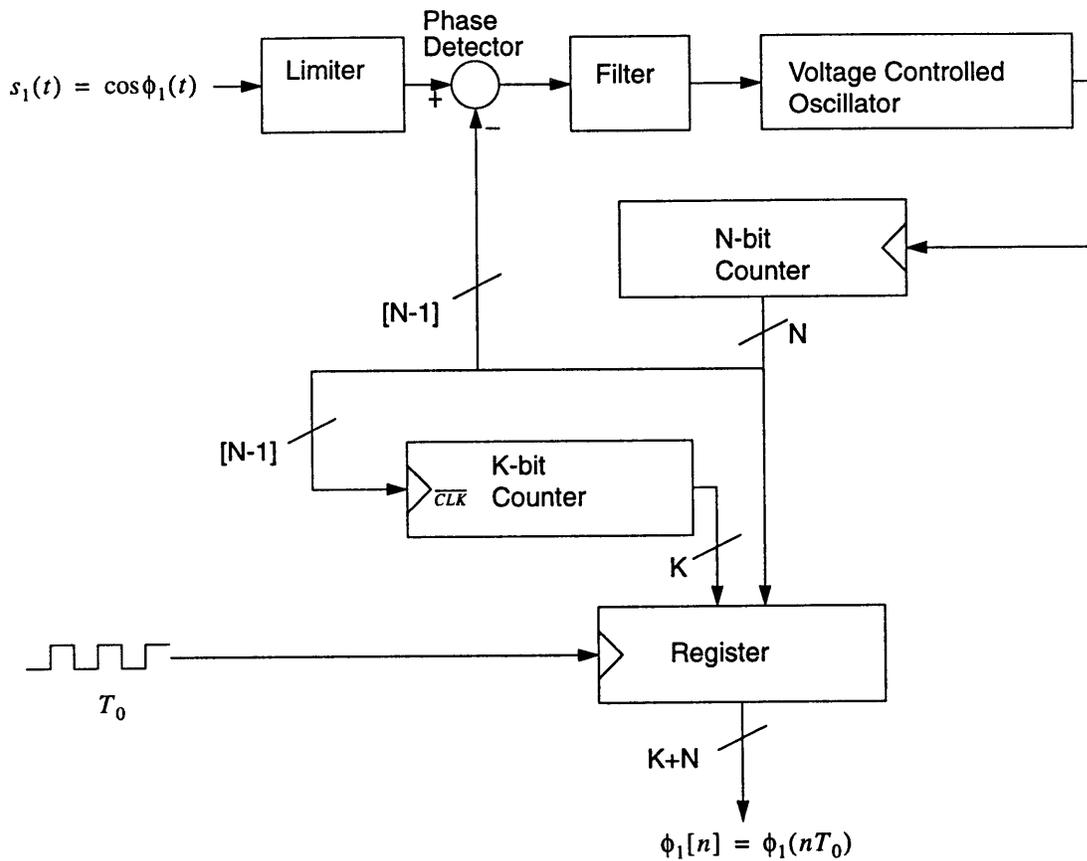


Figure 3.3: Analog phase-locked loop readout.

instantaneous phase at t_0 modulo 2π . To develop an intuitive understanding about how the readout indicates instantaneous velocity, Fig. 3.3 includes a K-bit counter which outputs, for any time $t=t_0$, the number of full 2π cycles completed between $t=0$ and $t=t_0$. Both counter outputs are latched every T_0 . Therefore, the register output is a digital representation of the phase of $s_1(t)$ which can be further processed by a digital signal processor (DSP). If the DSP clock frequency, T_0 , and the instantaneous frequency of $s_1(t)$ fall within relative ranges of each other, actual implementation can exclude the K-bit counter, leaving accumulation of total phase for the DSP. Note that implementing a readout for the VIA might use two PLLs, one for both $s_1(t)$ and $s_2(t)$.

The resolution of this readout is determined by the size of N and the maximum output frequency of the VCO, f_{VCO} , which are generally related by the Nyquist frequency, f_N , of the input, $s_1(t)$, as follows:

$$f_{VCO} \geq 2^N \times f_N$$

Assuming that the phase detector, limiter, and filter are ideal components, and that infinite-bit counters exist, the fundamental limits of velocity resolution depend on a high VCO frequency. Since phase resolution is $2\pi/2^N$ radians, velocity resolution, where k_p is the scale factor in units of rad/(m/s), is:

$$\Delta v_0 \geq \frac{2\pi}{k_p} \times \frac{f_N}{f_{VCO}} \text{ m/s} \quad (3.12)$$

3.2.2 Period Readout Scheme

Like the analog phase-locked loop readout, the period readout accomplishes input phase digitization using a high resolution counter clocked by a high frequency signal. In the PLL readout, the VCO output essentially serves as a variable high frequency clock for the N-bit counter. For the period readout, however, the counter clock signal is a fixed frequency at $f_{clk}=1/T_1$, as shown in Fig. 3.4(a). The M-bits of the counter are latched by a register that is clocked by a squarewave form of the input signal. Although the high speed clock is fixed in this case, an accurate phase word is available irregularly, only at rising edges of the input. That is, the sample rate of the output varies and depends on the frequency of $s_1(t)$

$$\frac{1}{T_s} = \frac{d\phi_1(t)}{dt}.$$

Whereas in the PLL readout the clock T_0 could read instantaneous phase from the counters, the period readout scheme requires interpolation to derive instantaneous phase at a constant rate. Such interpolation methods use software to turn the period readout into a virtual phase-locked loop and rely heavily on digital filters to improve upon resolution [1].

Other schemes can be employed to improve upon performance of the period readout. One technique uses capacitor interpolation (Fig. 3.4(b)) to improve upon resolution and

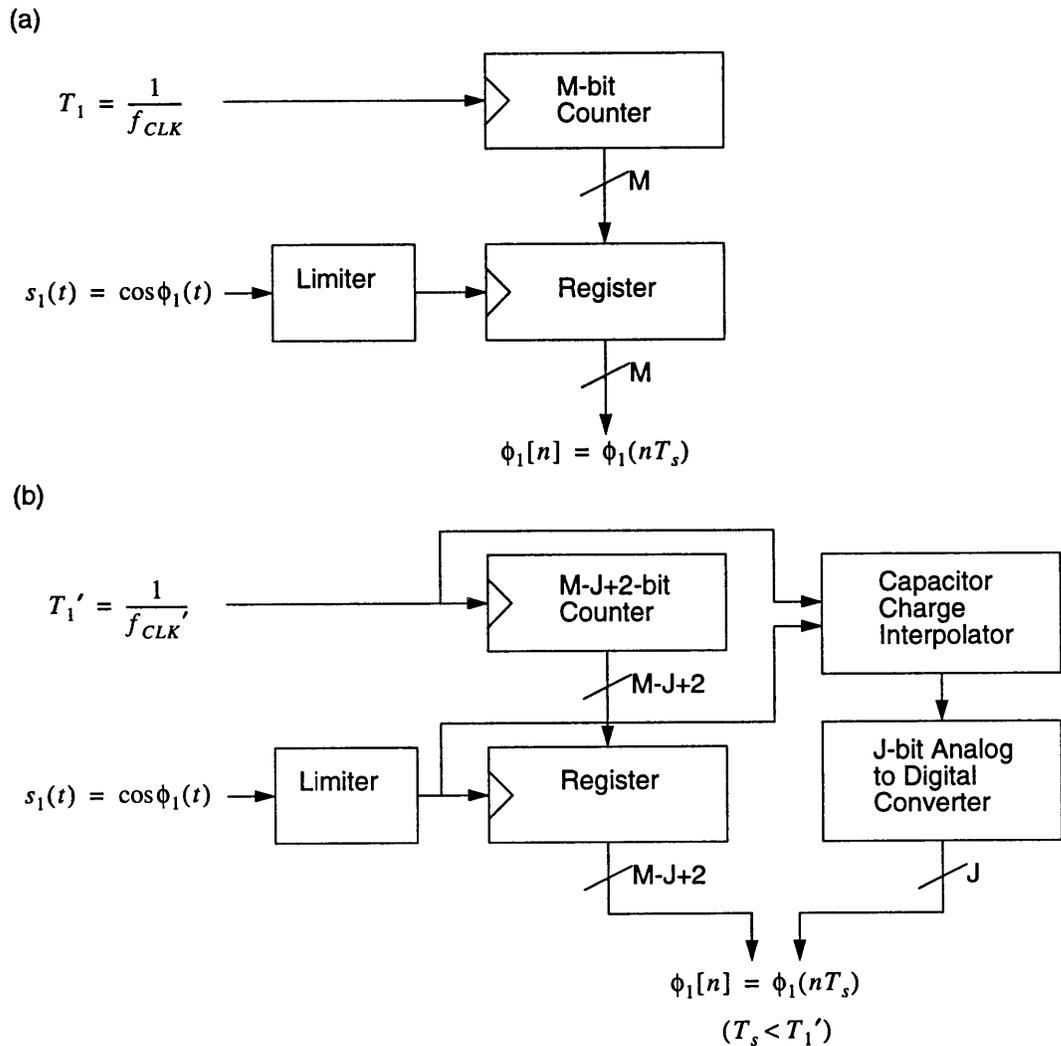


Figure 3.4: (a) Period readout scheme (b) Period readout with capacitor interpolation effectively makes clocks seem faster than they really are. The paper in which it is described claims that a system that uses a 10-MHz clock with capacitor charge interpolation can read phase as if using a virtual 7-GHz clock [1].

Again, assuming ideal components and not considering digital filtering, a first-order approximation to the fundamental limit of resolution depends on the clock frequency and the bandwidth of the input signal. In this readout, each binary value of the counter represents a quantum of time and not phase. Therefore, the phase resolution will vary and

depend on frequency of $s_1(t)$. The worst case, and thus the limit of resolution, occurs when the input frequency is as large as possible.

$$\Delta v_1 \geq \frac{2\pi}{k_p} \times \frac{f_N}{f_{CLK}} \text{ m/s} \quad (3.13)$$

The period readout scheme also has a lower bound to the input frequency that depends on the rate at which a velocity readout is desired. Even with interpolation methods, the minimum input frequency must be greater than or equal to $1/T_0$, the output rate of the readout to an inertial navigation system which is 500 Hz for the VIA

$$\min\left(\frac{d\phi_1(t)}{dt}\right) \geq \frac{1}{T_0}. \quad (3.14)$$

3.3 Motivations for Reading Acceleration

Now, the requirements and unique specifications of the Vibrating Island Accelerometer will be considered. In addition, the advantages of reading acceleration will be discussed.

Recall from Chapter 2 that the VIA crystal quartz transducer is expected to output two angle modulated signals, defined again here for convenience:

$$s_1(t) = S_1 \cos\left(2\pi f_1 t + 2\pi k_1 \int_0^t a(t) dt\right) \quad (3.15)$$

$$s_2(t) = S_2 \cos\left(2\pi f_2 t - 2\pi k_2 \int_0^t a(t) dt\right) \quad (3.16)$$

The carrier frequencies, f_1 and f_2 , are expected to be on the order of 24 MHz, the bandwidths of $a_1(t)$ and $a_2(t)$, f_a , are expected to be 100 Hz. The scale factor is 50 Hz/g, and the maximum acceleration to be measured is 20 g. Total accumulated velocity is expected to be output at $1/T_0=500$ Hz to a guidance computer.

3.3.1 Signal Bandwidth and Processing Speed Considerations

Section 3.2 concluded that approximations to the velocity resolution of both the analog PLL and period readouts depended on the maximum frequency, f_N , of the input signals. Therefore, this critical parameter must be derived from the above specifications. As discussed in Chapter 2, because the VIA is expected to linearly relate acceleration to frequency change, the oscillator outputs can be viewed as angle modulated waves as described in (3.15) and (3.16). Thinking of $s_1(t)$ and $s_2(t)$ as FM signals, analog communications theory approximations will be used to estimate their necessary FM bandwidth for this first-order analysis.

First, consider the modulation index, m_f , of the frequency modulated signals. The modulation index is a ratio of the peak carrier frequency deviation to the modulating signal frequency, and is a measure of the bandwidth of the modulated signal centered at the carrier frequency.

$$m_f = \frac{\max(\Delta f_1)}{f_a} \quad (3.17)$$

The signal spectrum behaves as a Bessel function of the first kind, with dependence on m_f . For $m_f < 0.25$, the signal is categorized as narrowband FM (NBFM), and its spectrum is a single-carrier with double-sideband, precisely like an amplitude modulated waveform except the sideband is 90 degrees out of phase with the carrier [3][4]. Thus, the NBFM signal has a bandwidth of $f_B = 2f_a$ around the carrier. For wideband FM (WBFM), where $m_f > 0.25$, the often used Carson Rule states that 99% of the signal power will fall within the frequency band defined as the following [3][4]:

$$f_B = 2f_a(1 + m_f) \quad (3.18)$$

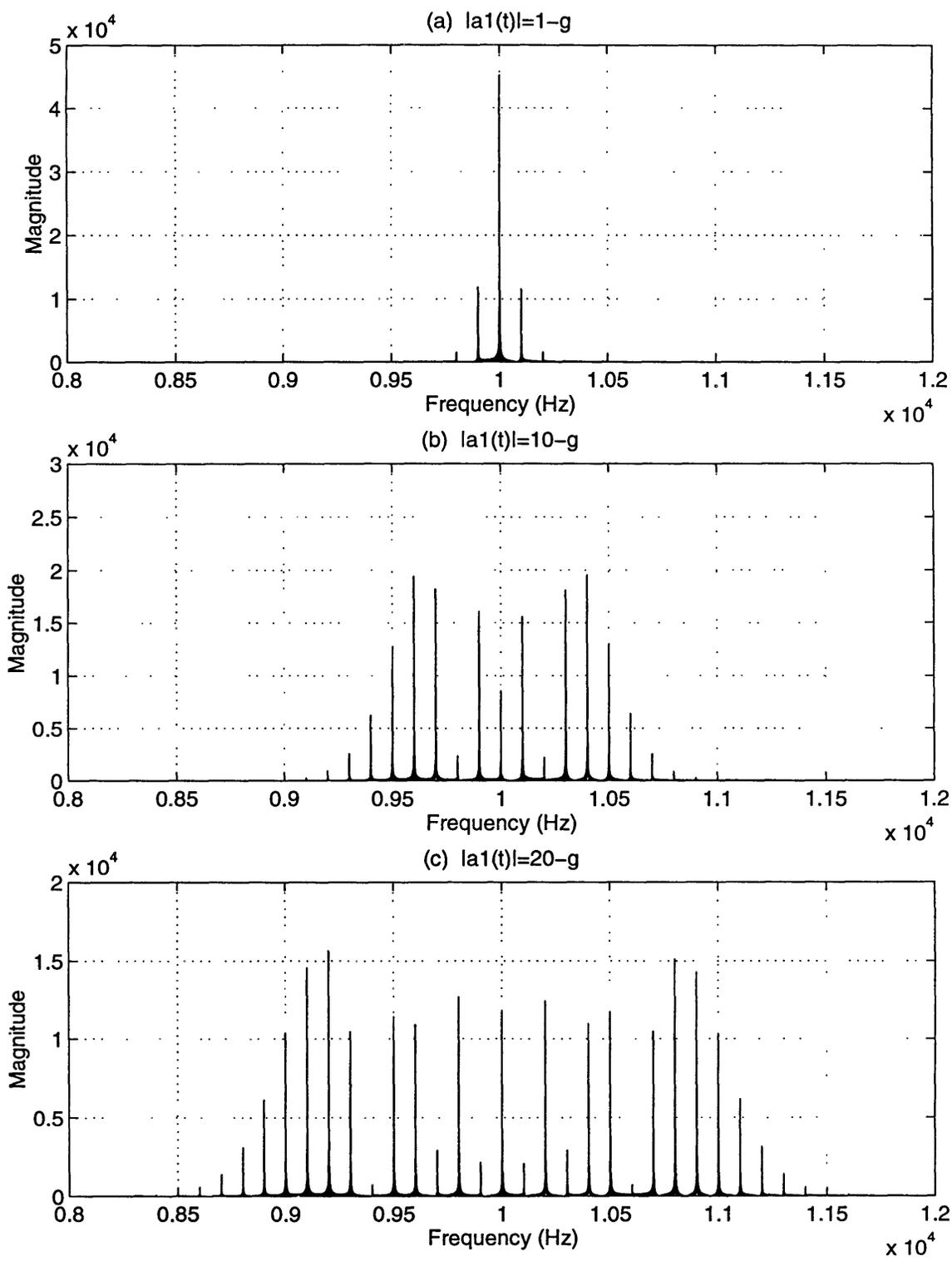


Figure 3.5: Spectrum of $s_1(t)$ for $f_1=10$ kHz and maximum acceleration of (a) 1 g, (b) 10 g, and (c) 20 g.

For the VIA parameters, the following modulation index is calculated:

$$m_f = \frac{(20g)\left(50\frac{\text{Hz}}{g}\right)}{(100\text{Hz})} = 10$$

Since $m_f > 0.25$, Carson's Rule approximates the bandwidth around f_1 as,

$$f_B = 2(100)(1 + 10) = 2.2\text{kHz} \quad (3.19)$$

Fig. 3.5 shows the spectrum of $s_1(t)$ for $S_1=1$, $k_1=50$ Hz/g, $f_1=10$ kHz, and $a_1(t)=A\cos(2\pi 100t)$, with $A=\{1, 10, 20\}$. For a carrier frequency of $f_1=24$ MHz, the spectrums would be identical to those shown only centered around 24 MHz rather than 10 kHz (10 kHz is used here simply to get a clearer graphic from Matlab). Notice that each pair of bands is separated by the acceleration frequency of 100 Hz, which is consistent with Bessel function theory. For a maximum acceleration of 1 g (Fig. 3.5(a)), $m_f=0.5$ and the spectrum resembles a single carrier, double sideband picture. Lastly, observe that for 20 g acceleration (Fig. 3.5(c)), the 99%-power bandwidth as calculated using the Carson Rule (3.19), lies between 8.9 kHz and 11.1 kHz. In this chapter, the Carson Rule is used only as an approximation of the signal bandwidth. The bandwidth required to meet the resolution specifications of the instrument is a more stringent figure, and is considered much more carefully in Chapter 4.

Having used Carson's Rule to estimate the FM bandwidth around the carrier, the Nyquist frequency of $s_1(t)$ can be approximated by adding in the carrier, f_1 :

$$f_N \approx f_1 + \frac{f_B}{2} \quad (3.20)$$

Now it is possible to get a first order estimate on the performance requirements of possible velocity readout schemes using equations 3.12 and 3.13. Taking the phase sensitivity to be $k_p=314$ rad/(m/s) and the desired velocity resolution to be $\Delta v=0.02$ ft/s, as discussed

in Chapter 2, for f_N approximately equal to 24 MHz, f_{VCO} and f_{CLK} are likewise approximately 24 MHz.

While this is certainly within the range of available VCO and clock frequencies, since velocity readout is only required at 500 Hz, and moreover the acceleration signal is only 100 Hz itself, one design issue that must be addressed is the possibility of obtaining the desired resolution using slower clocks.

Heterodyning

By multiplying the inputs with a local oscillator, the input can be modulated down to

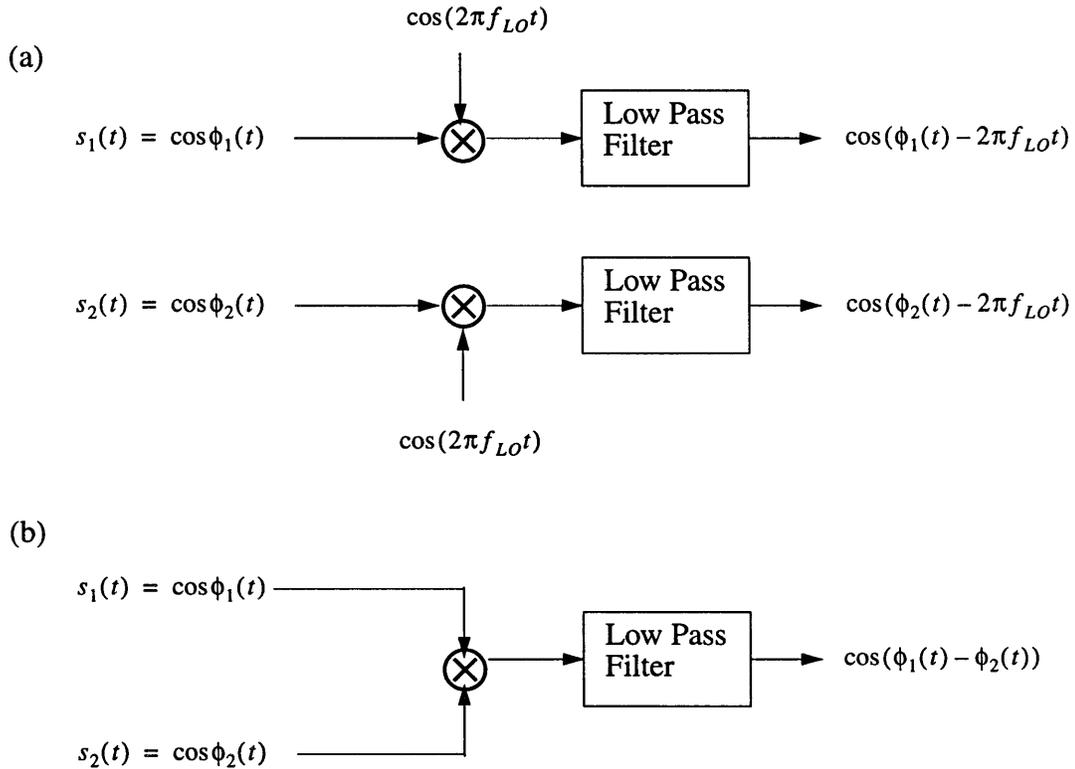


Figure 3.6: Beating down oscillator signals. (a) Multiplying each input by a Local Oscillator (LO). (b) Multiplying two inputs together for common mode reduction.

lower frequencies, reducing f_N , f_{VCO} and f_{CLK} by a proportional amount. In fact, the lower the f_N the better. Fig. 3.6(a) shows how this might be done.

Although at first this seems like a good idea, phase readout schemes may have trouble reading $\cos(\phi_i(t) - 2\pi f_{LO}t)$. The period readout scheme has a lower bound on the input frequency, as seen in (3.14). Furthermore, if the scale factor is increased, the Nyquist frequency, f_N , can conceivably increase until the dynamic range is larger than an interpolating period readout scheme can handle. Regarding the PLL readout, most VCO's have lower bounds to their frequency outputs, as well, that would not be able to handle a higher scale factor and wider band FM signal, despite its being beaten down to lower frequencies.

Rather than mixing the FM inputs with local oscillators, mixing them with each other (Fig. 3.6(b)) would be a particularly desirable possibility, and would be the ideal approach. One of the principle attributes of the VIA, as discussed in Chapter 2, is that the two oscillators on the same crystal will be similarly affected by temperature transients and other environmental factors. Beating the two oscillator signals together should therefore allow common modes to cancel out in looking at the difference frequency between $s_1(t)$ and $s_2(t)$. However, if $|f_1 - f_2| < \frac{1}{T_0}$, for a zero acceleration signal, beating $s_1(t)$ with $s_2(t)$ would produce a low frequency signal that would be below the dynamic range of the period readout. In order to use the period readout, the product of the two inputs would have to be mixed with another local oscillator to raise the beat note into the dynamic range of the readout. Ideally, a readout scheme would be able to directly process the beat note of $s_1(t)$ and $s_2(t)$.

In summary, analog PLL and period readout schemes for the VIA seemingly require unnecessarily high clock speeds. It is desirable to run the phase-readout schemes at lower frequencies if possible. However, reducing the velocity readout clock requirements by beating down the oscillator outputs may potentially be difficult due to inherent dynamic range resolutions of the devices.

Next, one further observation is made on possible shortcomings of phase readout schemes for the VIA.

3.3.2 Utilizing Amplitude Information

Both the analog PLL and period readout methods are essentially zero-crossing detectors that digitize the time axis. Both methods saturate the angle modulated oscillator outputs with limiters to virtually eliminate amplitude noise. However, in eliminating the amplitude noise, a significant amount of amplitude information has been reduced to containing only one bit of information! The amplitude is essentially digitized into 1-bit, while the time axis is digitized into N- or M-bits, depending on the readout scheme. Therefore, both phase readout methods, as seen by (3.12) and (3.13), have only one degree of freedom, one fundamental variable that can be adjusted. For the PLL, the f_{OSC} can be adjusted, and for the period readout, f_{CLK} is the sole parameter. A very intriguing possibility begins to unfold. Increasing the amplitude resolution to be greater than 1-bit by sampling the amplitude and retaining its information, can reduce the required resolution of the time axis, thereby reducing the necessary readout clock speed. Instead of one degree of freedom, there would be two degrees to work with, for more design flexibility. As mentioned in Chapter 2, previous accelerometers were not used for high resolution applications and the avoidance of using A/D conversion could simplify the readout scheme. However, for a high resolution application, use of an A/D converter may be worth the extra cost of implementation. Sampling the amplitude would also allow the usage of digital signal processing techniques which, after suffering quantization noise in the conversion, can be designed to have good noise performance, hinting at the possibility of achieving high resolution.

From a purely conceptual viewpoint, the desire, then, is to use amplitude information of $s_1(t)$ and $s_2(t)$ to help extract information about the phases of $s_1(t)$ and $s_2(t)$. This is another opportunity to borrow ideas from analog communication theory. Implementing a traditional phase demodulator would closely parallel the analog PLL readout scheme previously discussed, and be bounded by similar limitations. The desire to make use of amplitude information leads to the possibility of using a digital FM receiver to obtain instantaneous frequency information from the amplitude by differentiating $s_1(t)$ and $s_2(t)$.

3.4 Acceleration Readouts

As in the rest of this chapter, up to this point, this section's analysis makes a number of assumptions in proceeding to uncover first-order theoretical results. In seeking to make some telling first comparisons between reading out phase and reading frequency, too much detail, at this point, would further hinder progression to the more detailed treatments contained in the next chapter. In this section, a few standard FM digital demodulation schemes will be briefly discussed, one of which involves digital differentiation. For that demodulation scheme, called cross-differentiate multiply demodulation, using an extreme number of bits at the minimum sampling rate in a traditional analog-to-digital (A/D) conversion process will be considered. Then, Section 3.4.3 will consider how using sigma-delta modulation in the A/D conversion allows the trade-off of sampling frequency for bits required in the quantizer.

3.4.1 Four Digital FM Demodulators

The fundamental limits of reading out frequency are limited by at least two factors: analog-to-digital conversion and choice of demodulation scheme. Since various FM demodulators take advantage of different A/D conversion strategies, four distinct varieties of digital FM demodulation will be briefly considered, first.

The digital phase-locked loop system is first and foremost a coherent phase detecting scheme. The implementation is similar to the phase readout scheme in Section 3.2.1. Again, the scheme does not make use of carrier amplitude information, and is probably best suited for narrowband applications.

The digital arctangent PM/FM demodulator [7][8] separates an angle modulated signal into its in-phase and quadrature components, A/D converts them, and then divides the former by the latter. The arctangent of the result is either computed or found using a ROM look-up table. The measured phase could then be differentiated to complete FM demodulation, allowing it to serve as a PM or FM demodulator. Despite its function, however, the method requires a jump detector to handle the numerical division by zero. For the VIA readout, this scheme would be implemented as a phase readout that takes advantage of the amplitude information of the modulated carrier. Additionally, it has been shown to be suitable for both narrowband and wideband FM, and can achieve signal-to-noise ratios of 70-dB [7]. In order to obtain a high resolution readout, however, the arctangent method would require a large amount of memory. Nevertheless, as memories get faster and larger, this scheme shows promise as a potential phase-readout strategy for the VIA.

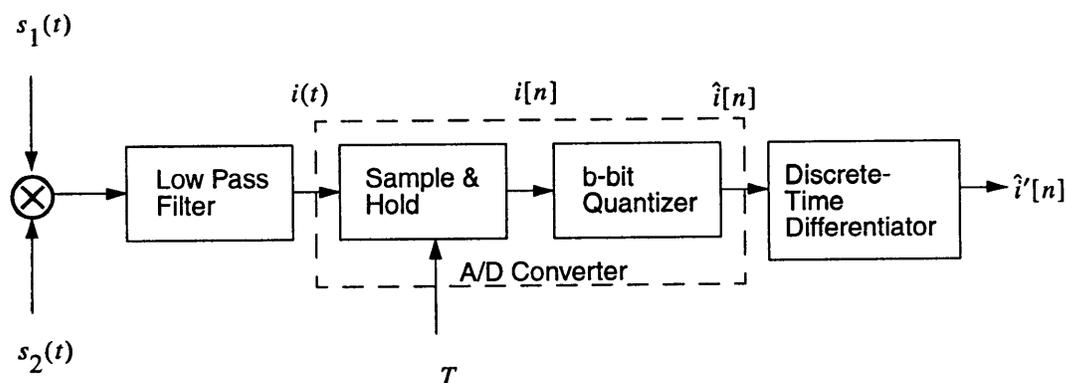
The oversampling delta-sigma frequency discriminator [9][10][11], the most recently discovered of the four detection schemes considered here, directly quantizes instantaneous frequency using delta-sigma modulation. Delta-sigma modulation has been studied extensively in the context of A/D conversion. Section 3.4.3 will, in fact, touch upon the basics and apply certain results to A/D conversion for the VIA. The delta-sigma frequency discriminator shows potential in a report that demonstrates how it can achieve signal-to-quantization noise ratio of 115-dB [10]. However, while this scheme seems simple and efficient, two primary concerns exist regarding performance for the VIA. First, the achievable resolution of delta-sigma frequency discriminator is limited by an analog phase

detector, which has been implemented using a standard D-flip flop. Second, the reported work suggests that this scheme is best suited for narrowband FM, and it is unclear how it might perform for WBFM.

The conventional cross-differentiate multiply (CDM) FM digital receiver [12][13] separates the in-phase and quadrature components of an FM modulated signal like the arctangent method, A/D converts, then differentiates the branches, and then cross multiplies the various terms to reject the AM components from the differentiation. Although the scheme requires a relatively large number of mathematical operations, its functional implementation is very straight forward, as shall be seen in Chapter 4, and it has been used to demodulate wideband FM, with reported signal-to-distortion ratios of up to 90 dB [12]. For these reasons, this scheme has been elected as the most suitable for theoretical analysis and comparison with phase demodulation methods.

3.4.2 CDM Demodulation with Traditional A/D Conversion

Chapter 4 will focus on the CDM in much greater detail and discover that two



$$i(t) = S_1 S_2 \cos \left(2\pi \left(k_1 \int_0^t a_1(t) dt + k_2 \int_0^t a_2(t) dt \right) \right) = S_1 S_2 \cos \phi_i(t)$$

Figure 3.7: Simplified CDM Frequency Demodulator

fundamental constraints on its performance are the resolution attainable by analog-to-digital conversion and the accuracy of discrete-time differentiation. Consider the simplified, incomplete CDM demodulator in Fig. 3.7. Analyzing only this portion of the scheme will yield the necessary first-order acceleration resolution estimate. Section 4.2.2 will show that the output $\hat{i}'[n]$ is proportional to acceleration.

One of the advantages of this scheme is that it naturally functions as a baseband demodulator. Therefore, the inputs can be mixed to eliminate common mode and the resulting beat note can be processed.

The velocity readout schemes attained resolution in sampling the phase directly with a high sampling rate, while relying on only 1-bit of amplitude intelligence. Considering these two parameters to be f_{\max} and b_{\min} , the desire now is to achieve the same velocity resolution by sampling the angle modulated input at a smaller frequency than f_{\max} by relying on amplitude discretization great than b_{\min} bits. However, first, having already defined a lower limit on amplitude resolution and an upper bound on sampling frequency, complete definition of the playing field occurs by defining an upper limit on amplitude resolution and a lower bound on sampling frequency. Only thereafter will other considerations be placed into meaningful perspective.

The lower limit of sampling frequency is ultimately dictated by the Nyquist theorem, which ensures that aliasing can be avoided if the sampling rate is at least twice the maximum frequency of the analog signal to be sampled. For the VIA readout parameters, however, the minimum sampling frequency will also depend on the frequency at which the velocity reading is to be output to the inertial navigation system, which could be above the Nyquist frequency for certain parameters.

The maximum number of bits used to quantize the amplitude ultimately would be dictated by the state of A/D converter technology. For this theoretical analysis, it is simply

denoted as b_{\max} , and will be determined in deriving the fundamental resolution limit for the CDM demodulator.

For linear and time-invariant (LTI) systems, the system frequency response reacts to sinusoidal and complex exponential inputs in a very special and powerful way. Sinusoids and complex exponentials are *eigenfunctions* of LTI systems, and when they serve as the input they emerge at the output of the system scaled in magnitude and shifted in phase. The scale factor and phase shift depend solely on the frequency of the eigenfunction and the frequency response of the LTI system. Fig. 3.8(a), shows that if the input to an LTI system is a sinusoid, the output is also a sinusoid, but scaled and shifted by a phase and magnitude determined by the instantaneous frequency of the sinusoid and the frequency response of the system. To be consistent with the development thus far, the instantaneous frequency of $i[n]$, from Fig. 3.7, is defined as

$$\omega_0 = \frac{d}{dn}\phi_i[n] = 2\pi(k_1a_1(nT) + k_2a_2(nT)) \quad (3.21)$$

where $\phi_i[n]$ is the phase of $i[n]$. When the LTI system is a differentiator, as in Fig. 3.8(b), the eigenvalue magnitude is merely the instantaneous frequency of the input and the eigenvalue phase is $\frac{\pi}{2}$ radians, resulting in $i'[n]$, the derivative of $i[n]$, as it should be.

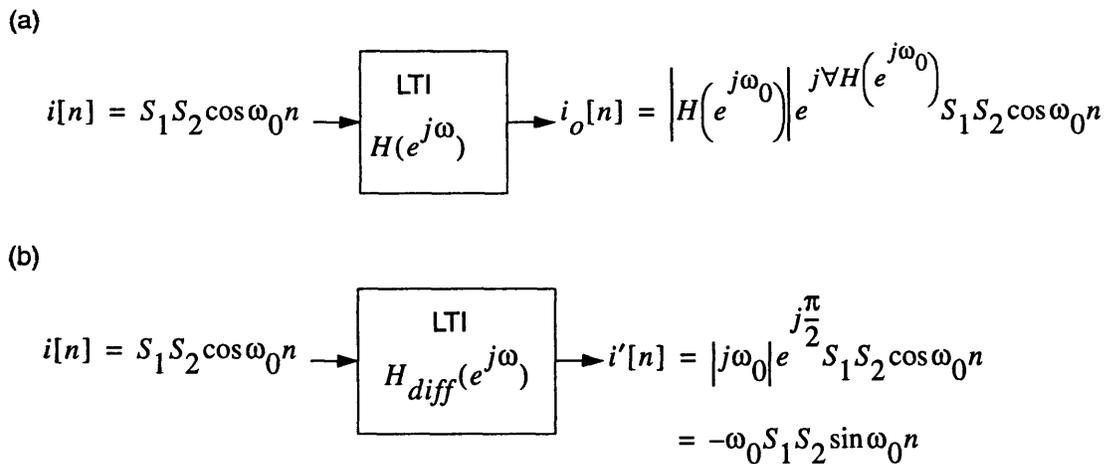


Figure 3.8: Eigenfunction and eigenvalues of LTI systems.

The ideal discrete-time differentiator is commonly defined as follows [14]:

$$H_{diff}(e^{j\omega}) = \begin{cases} \frac{j\omega}{\pi} & 0 \leq \omega \leq 2\pi F_p \\ \frac{j(2\pi - \omega)}{\pi} & 2\pi(1 - F_p) \leq \omega \leq 2\pi \end{cases} \quad (3.22)$$

F_p is the maximum frequency of the signal to be differentiated. DSP texts commonly express F_p as a frequency that is normalized to the sampling frequency. That is, F_p is expressed as a decimal number in the range $0 < F_p < 0.5$, for an unaliased spectrum. On the normalized frequency axis, 0.5 corresponds to the Nyquist frequency and 1.0 represents the sampling frequency. Since the discrete-time Fourier Transform is periodic in 2π , the region from zero to one on the normalized frequency axis represents the entire spectrum.

Like digital integrations differ from ideal integrations, actual discrete-time differentiators differ from the ideal differentiator described in (3.21). However, well characterized discrete-time differentiators with low error can be designed using the Remez Exchange Algorithm for optimal FIR filter design [14]. Achieving low magnitude error in the differentiator requires low F_p and a high order, or large length, impulse response. Low F_p means the sampling frequency is large compared to the maximum input frequency.

Fig. 3.9 gives an example of a 17-point discrete-time differentiator and its magnitude error. For a maximum input frequency of 1 kHz sampled at 10 kHz, $F_p=0.1$. Fig. 3.9, shows that this results in a maximum amplitude error of approximately $e_{diff}[n]=1 \times 10^{-7}$. The ideal magnitude response is plotted as a dashed line in the Magnitude Response plot. Since this is a finite impulse response filter, the phase is linear, resulting in a delay as the only phase distortion. Therefore, the phase error is negligible.

Consider the input to be a sinusoid, as in Fig. 3.8(b), the magnitude error of the differentiator would contribute to the scaling of the eigenfunction at the output. A second

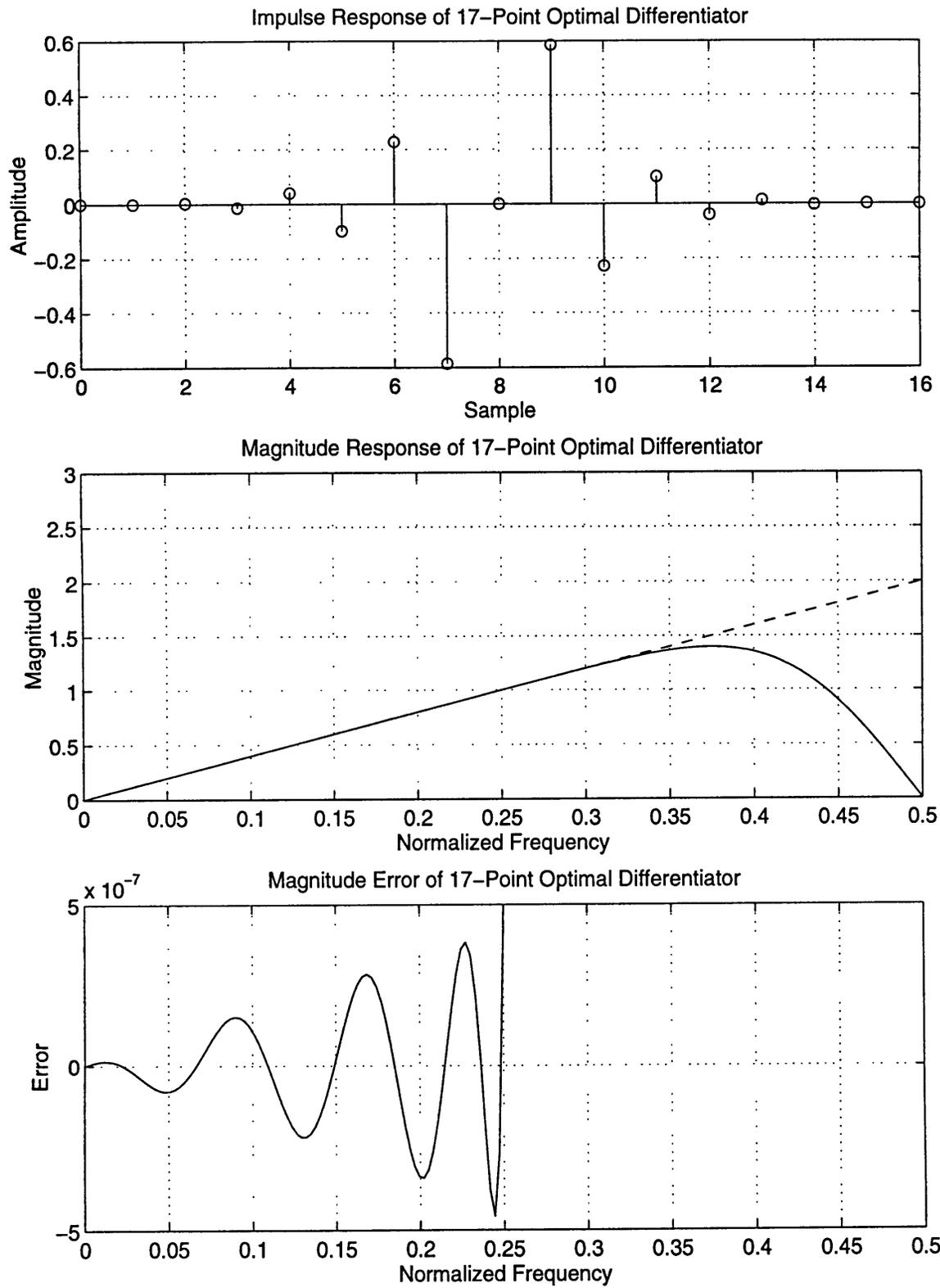


Figure 3.9: Discrete-Time Differentiator.

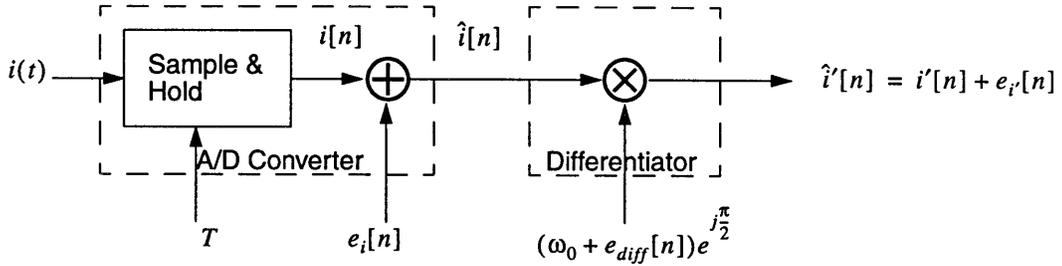


Figure 3.10: Error model of simplified CDM frequency demodulation method. major source of error at the output is the quantization noise contributed by the A/D converter. As in equations 3.8 and 3.9 in Section 3.1.2, quantization noise can be modeled as additive white noise, in the case of rounding, with a variance, or power, dependent on the number of bits used in the conversion and the maximum signal amplitude. Therefore, to get a first order estimate on the fundamental limitations of the CDM demodulator resolution, two primary error sources can be modeled as in Fig. 3.10.

Now, an estimate can be derived for the acceleration resolution achievable by the CDM receiver. The output of the differentiator can be expressed as the following:

$$\hat{i}[n] = \hat{i}[n](|j\omega_0| + e_{diff}[n])e^{j\frac{\pi}{2}}. \quad (3.23)$$

Since $\hat{i}[n] = i[n] + e_i[n]$,

$$\hat{i}[n] = (i[n] + e_i[n])(\omega_0 + e_{diff}[n])e^{j\frac{\pi}{2}}. \quad (3.24)$$

Expanding the product on the right-hand side of the equation,

$$\hat{i}[n] = \omega_0 i[n] e^{j\frac{\pi}{2}} + (i[n] e_{diff}[n] + \omega_0 e_i[n] + e_i[n] e_{diff}[n]) e^{j\frac{\pi}{2}}. \quad (3.25)$$

The term $e_i[n]e_{diff}[n]$ is negligible and is dropped. Meanwhile, writing the output as $\hat{i}[n] = i[n] + e_{i'}[n]$, and substituting this into the above equation,

$$i[n] + e_{i'}[n] = \omega_0 i[n] e^{j\frac{\pi}{2}} + (i[n]e_{diff}[n] + \omega_0 e_i[n]) e^{j\frac{\pi}{2}} \quad . \quad (3.26)$$

Separating the ideal output from the error terms,

$$i'[n] = \omega_0 i[n] e^{j\frac{\pi}{2}} \quad (3.27)$$

$$e_{i'}[n] = (i[n]e_{diff}[n] + \omega_0 e_i[n]) e^{j\frac{\pi}{2}} \quad (3.28)$$

Now, taking the magnitude of (3.28),

$$|e_{i'}[n]| = |i[n]e_{diff}[n] + \omega_0 e_i[n]| \quad (3.29)$$

By the triangle inequality,

$$|e_{i'}[n]| \leq |i[n]| |e_{diff}[n]| + \omega_0 |e_i[n]| \quad (3.30)$$

Considering the maximum quantization and differentiator error magnitudes,

$$|e_{i'}[n]| \leq |i[n]| |e_{diff}[n]| + \omega_0 |e_i[n]| \leq |i[n]|_{max} |e_{diff}[n]|_{max} + \omega_0 |e_i[n]|_{max} \quad (3.31)$$

Since the amplitude of $i[n]$ is $S_1 S_2$, and the maximum quantization error is $S_1 S_2 / 2^b$, the maximum error of (3.28) can be denoted as follows:

$$|e_{i'}[n]|_{max} = S_1 S_2 \left(|e_{diff}[n]|_{max} + \frac{\omega_0}{2^b} \right) \quad (3.32)$$

Substituting back in the expression for instantaneous frequency from (3.21),

$$|e_{i'}[n]|_{max} = S_1 S_2 \left(|e_{diff}[n]|_{max} + \frac{2\pi(k_1 |a_1(nT)|_{max} + k_2 |a_2(nT)|_{max})}{2^b} \right) \quad (3.33)$$

Since $s_1(t)$ and $s_2(t)$ are unit-less, and $a_1(nT)$ and $a_2(nT)$ have units of acceleration, the expression in (3.33) is a quantity of acceleration. In fact, Chapter 4 will show that $i'[n]$ is proportional to $a[n]$, the demodulated acceleration. Accordingly, the maximum error of $i'[n]$ is equivalent to the maximum acceleration scaled by a constant. For now, assuming that the scale factor is unity, (3.33) is taken as an expression of acceleration resolution for the CDM demodulator.

$$\Delta a = S_1 S_2 \left(|e_{diff}[n]|_{max} + \frac{2\pi(k_1 |a_1(nT)|_{max} + k_2 |a_2(nT)|_{max})}{2^b} \right). \quad (3.34)$$

For $S_1=S_2=1$, $|e_{diff}[n]|_{max} = 0.1\mu g$, $k_1=k_2=50$ Hz/g, $|a_1(nT)|_{max}=|a_2(nT)|_{max}=20$ g, and $b=32$ bits, the CDM demodulator resolution, or error, is $\Delta a = 1.56\mu g$.

For the phase readout schemes, equations 3.12 and 3.13 indicate that the velocity resolutions are dependent on either a high clock or VCO frequency. In this frequency readout scheme, given angle modulated inputs as in (3.15) and (3.16), and given a particular discrete-time differentiator, the acceleration resolution depends on the number of bits, b , of the A/D converter. Equation 3.34 also shows that the full-scale range of the A/D converter should be dictated by not only the amplitudes of $s_1(t)$ and $s_2(t)$, but also the largest frequency of $s_1(t)$ and $s_2(t)$. Frequency is, after all, the parameter that is digitized, so it makes sense that frequency helps determine the resolution of the A/D converter in this case. Additionally, one motivation, as described in Section 3.3.2, was to make use of the amplitude information of $s_1(t)$ and $s_2(t)$, which is also happening here in (3.34). As stated a few paragraphs ago, the amplitude of the angle-modulated waves are unit-less, and it is in fact the frequency of the waveforms that give units to the digitized values.

Equation 3.34 confirms that the acceleration resolution is dependent on both differentiator error (first term) and quantization error (second term). Assuming the differentiator error can be made arbitrarily small by designing it with an arbitrarily large length (in practice, the DSP processing speed and the sampling frequency would combine to limit the filter length), enabled looking solely at the dependence of resolution on the number of bits of the A/D converter. For the parameters listed above, and ignoring the differentiator error, the acceleration resolution as a function of the number of bits in the quantizer is plotted in Fig. 3.11. The next section will discuss how oversampling decreases the number of bits required in the A/D converter.

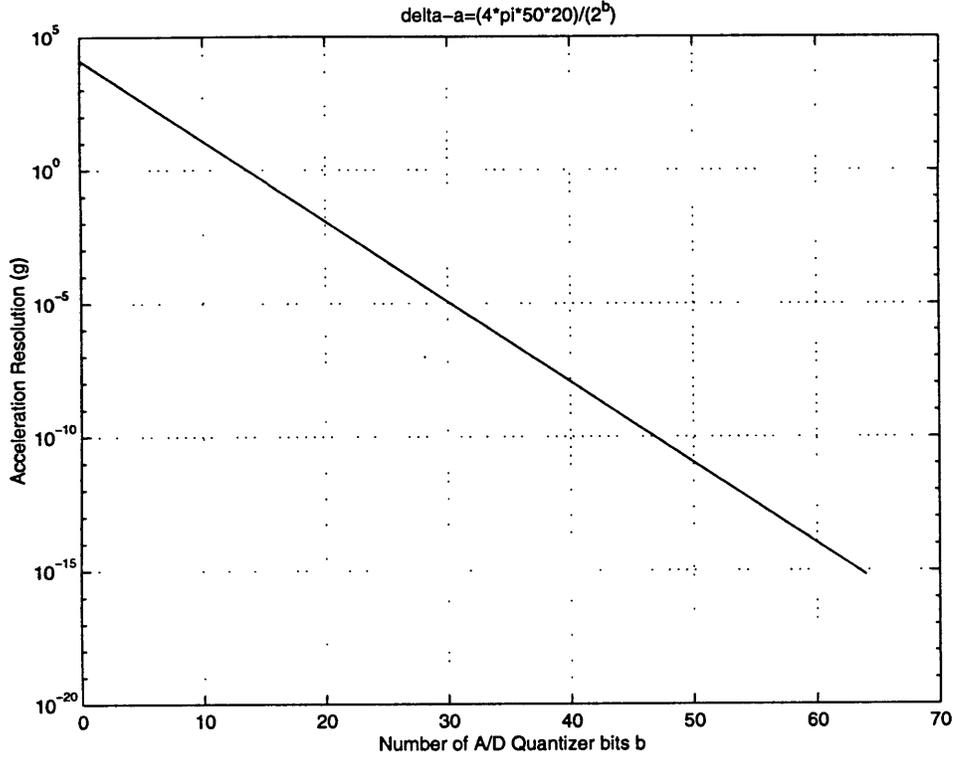


Figure 3.11: Acceleration resolution versus number of A/D bits from (3.34).

While this thesis will not further prove the results of (3.12) and (3.13), it *will* attempt in later chapters to verify the validity of (3.34) through more detailed analysis and simulation of the CDM FM demodulation scheme.

Now, to calculate instantaneous velocity from the acceleration measurement, the acceleration must be integrated from time zero to the time at which the velocity is desired. As seen in equation 3.10, the error of the derived discrete-time velocity, $e_{va}[n]$, will be attributed to a discrete-time integration error, $e_{dti}[n]$, and the accumulation of acceleration error over time. The metric used to compare acceleration to velocity resolution (3.11) is repeated here:

$$\max|e_{dti}[n]| + \frac{t\Delta a}{2} < \frac{\Delta v}{2} \dots \quad (3.35)$$

Using Riemann Sum integration, the maximum error (3.11) can now be calculated as,

$$e_{dii}[n] = \frac{S_1 S_2}{f_s} = 1 \times 10^{-4} \frac{m}{s} \quad (3.36)$$

Therefore, for the acceleration resolution found using (3.34), and for the digital integration error in (3.36), for a duration of $t=360$ seconds,

$$\max|e_{dii}[n]| + \frac{t\Delta a}{2} = (1 \times 10^{-4}) + \left(\frac{360}{2}\right) \left(\frac{32 \frac{m}{\text{sec}^2}}{1g}\right) (1.56 \mu g) = 0.01 \frac{ft}{\text{sec}} \quad (3.37)$$

Comparing (3.37) to the velocity readout schemes of Section 3.2, this resolution is on the same order of magnitude as the desired velocity resolution for the VIA (0.02-ft/sec). Section 3.3.1 estimated that without beating down the oscillator outputs, the period readout scheme would need a 24 MHz clock to achieve $\Delta v = 0.02$ ft/sec. Although the acceleration readout, in beating down the signals, gets close to this, achieving (3.38), it needs a 32-bit A/D converter to do so! So while the velocity readout schemes need fast clocks to achieve low velocity resolution, the cross-differentiate and multiply frequency demodulation scheme needs large A/D converters. And since a 24-MHz clock is much more feasible than a 32-bit A/D converter with 0.01-msec conversion time, the acceleration readout scheme does not seem promising.

However, the next section will show how sigma-delta modulation enables the trading-off of A/D converter bits with oversampling ratio to achieve the desired velocity resolution.

3.4.3 Oversampling Delta-Sigma Modulator A/D Conversion

In this section, trading in the traditional A/D converter (Fig. 3.12(a)) studied in the previous section for a delta-sigma-modulator-(DSM)-based A/D converter (Fig. 3.12(b)) will be shown to relax the required resolution of the quantizer by oversampling the filtered beat note of the oscillator outputs. After a brief description of DSM's, the noise shaping

capabilities of first and second order DSMs will be compared. This will provide intuition regarding why they are able to achieve good resolution. Then, attention will focus on how decimation can turn the oversampled modulator output into a multi-bit digital word, with the number of output bits dependent on the oversampling ratio. Lastly, equation 3.34 will be revised to consider DSM implementation

As seen in Fig. 3.12(b), A/D conversion using DSM's begins with a sample and hold block, with an oversampling ratio, N . DSM's usually use 1-bit quantizers whose output are fed back to be compared to the sampled analog input. The output of the DSM is a 1-bit, clocked signal and its average tracks the analog input. In this sense, delta-sigma modulators differ from delta-modulators (DM). DM's have 1-bit outputs whose average tracks the derivative of the analog input. Therefore, to match the DSM's application, the DM output must be digitally integrated. Furthermore, the DM does not shape the quantization noise in

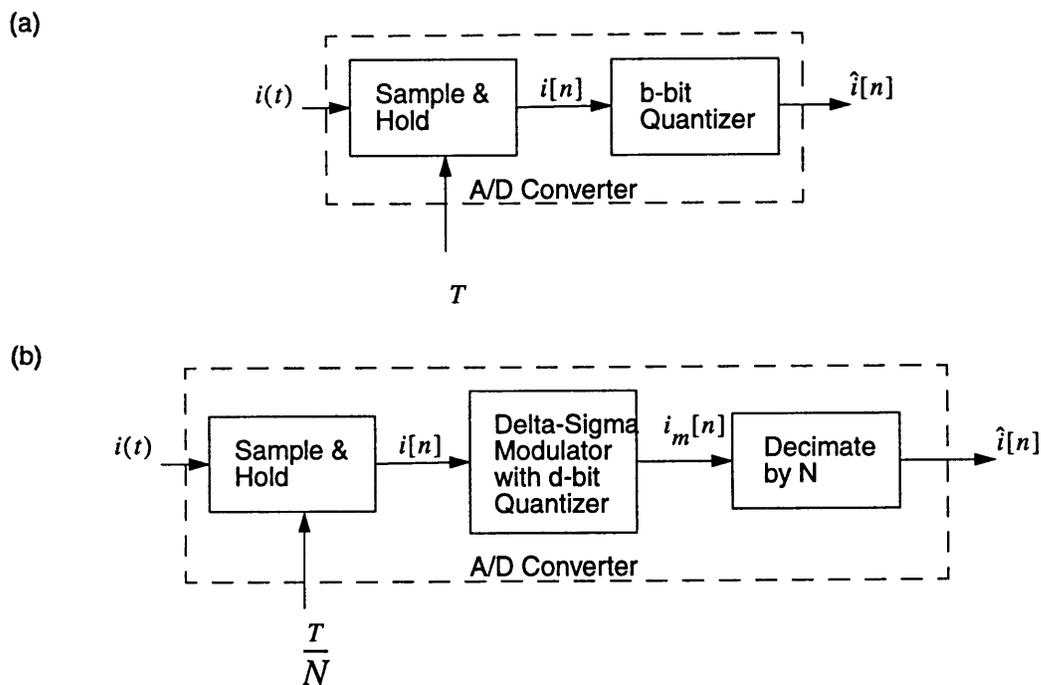


Figure 3.12: (a) Traditional A/D converter and (b) delta-sigma A/D converter.

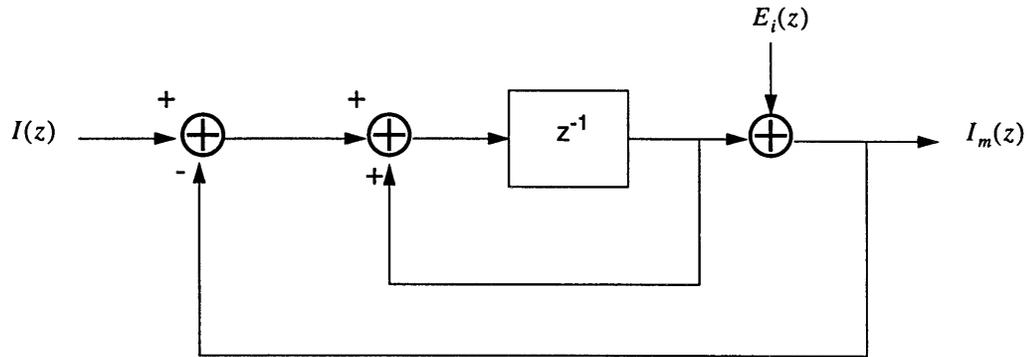


Figure 3.13: First-order sigma delta modulator.

the same way that a DSM does. So while DM's are useful for encoding signals at reduced bit rates, they do not function well in A/D conversion schemes [15].

Fig. 3.13 displays a z-transform representation of a first order DSM [15][16]. The summation of $E_i(z)$ represents a d-bit quantizer, similar to the b-bit quantizer representation in Fig. 3.10, where the quantization noise is modeled as a uniform distribution with average power dependent on the number of quantization levels. The input, $I(z)$, is the z-transform of $i[n]$ in Fig. 3.12, which is the oversampled filtered beat-note, $i(t)$ (Fig. 3.7), of the oscillator outputs. The modulator output, $I_m(z)$, can be described as follows:

$$I_m(z) = z^{-1}I(z) + (1 - z^{-1})E_i(z). \quad (3.38)$$

This equation shows that the DSM output is simply the input delayed by one sample plus an error term,

$$I_m(z) = z^{-1}I(z) + E_{i_m}(z) \quad (3.39)$$

where

$$E_{i_m}(z) = (1 - z^{-1})E_i(z). \quad (3.40)$$

In Fig. 3.10, the quantization error propagates through the system exactly as the input, $i[n]$, does. In (3.40), while the input is merely delayed, the quantization error term, $E_i(z)$, is filtered and thus “shaped.”

Assuming the quantization noise is wide-sense stationary, the power spectral density of the modulator output noise can be written as a function of the modulator quantization noise as follows:

$$\Phi_{e_{im}}(f) = |1 - e^{-j2\pi f}|^2 \Phi_{e_i}(f). \quad (3.41)$$

This expression can be further simplified in the following three steps:

$$\Phi_{e_{im}}(f) = |e^{-j\pi f} (e^{j\pi f} - e^{-j\pi f})|^2 \Phi_{e_i}(f) \quad (3.42)$$

$$\Phi_{e_{im}}(f) = |e^{-j\pi f} 2j \sin(\pi f)|^2 \Phi_{e_i}(f) \quad (3.43)$$

$$\Phi_{e_{im}}(f) = 4(\sin(\pi f))^2 \Phi_{e_i}(f). \quad (3.44)$$

From (3.44), it is apparent that the flat spectrum of $\Phi_{e_i}(f)$ is attenuated at low frequencies and accentuated at higher frequencies by the squared sinusoid.

Now, consider the second-order DSM in Fig. 3.14 [15][16]. Again, the quantizer is simply represented as additive white noise, $E_i(z)$. Solving for $I_m(z)$,

$$I_m(z) = z^{-1}I(z) + (1 - z^{-1})^2 E_i(z). \quad (3.45)$$

The modulator output is again the sum of the unit-sample-delayed input and the filtered quantization noise. The difference, now, is that the noise is shaped by a second order transfer function:

$$I_m(z) = z^{-1}I(z) + E_{i_m}(z) \quad (3.46)$$

where

$$E_{i_m}(z) = (1 - z^{-1})^2 E_i(z) \quad (3.47)$$

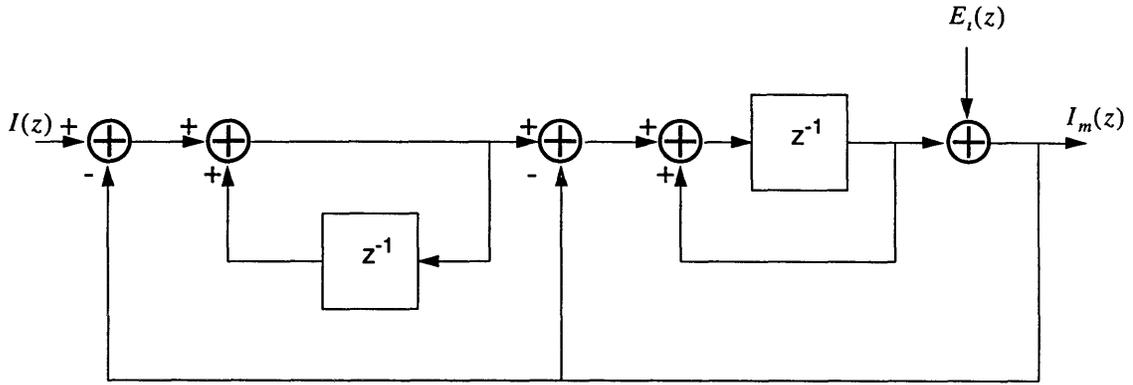


Figure 3.14: Second-order sigma-delta modulator.

so,

$$E_i(z) = (1 - 2z^{-1} + z^{-2})E_i(z). \quad (3.48)$$

Again, assuming that the noise is wide-sense stationary, the power spectral density of the quantization noise, $\Phi_{e_i}(f)$, can be related to the power spectral density of the DSM output noise, $\Phi_{e_{im}}(f)$:

$$\Phi_{e_{im}}(f) = |1 - 2e^{-j2\pi f} + e^{-j4\pi f}|^2 \Phi_{e_i}(f). \quad (3.49)$$

Further simplifying (3.49),

$$\Phi_{e_{im}}(f) = |-2e^{-j2\pi f} + 1 + e^{-j4\pi f}|^2 \Phi_{e_i}(f), \quad (3.50)$$

$$\Phi_{e_{im}}(f) = |-2e^{-j2\pi f} + e^{-j2\pi f}(e^{j2\pi f} + e^{-j2\pi f})|^2 \Phi_{e_i}(f), \quad (3.51)$$

$$\Phi_{e_{im}}(f) = |-2e^{-j2\pi f} + 2e^{-j2\pi f} \cos(2\pi f)|^2 \Phi_{e_i}(f), \quad (3.52)$$

$$\Phi_{e_{im}}(f) = |-2e^{-j2\pi f}(1 - \cos(2\pi f))|^2 \Phi_{e_i}(f), \quad (3.53)$$

and finally,

$$\Phi_{e_{im}}(f) = 4(1 - \cos(2\pi f))^2 \Phi_{e_i}(f). \quad (3.54)$$

As in (3.44), (3.54) shows that the noise spectrum, $\Phi_{e_i}(f)$, is attenuated at low frequencies, as well. Fig. 3.15 shows a plot of (3.44), (3.54), as well as the unshaped quantization

noise spectrum. As the input frequency lowers, the signal-to-noise ratio improves dramatically as the noise power in the baseband approaches zero. Without noise shaping, the quantization noise spectrum is uniform at all frequencies, as seen by the solid line in Fig. 3.15.

Interesting to note is that the shaped quantization noise spectrum is very similar to the noise spectrum of demodulated FM signals [17]. That is, white noise added to the carrier of a frequency modulated waveform gets shaped by the FM demodulation process in an analogous fashion to the shaping of quantization noise using delta-sigma modulation. This insight helps explain the function of delta-sigma discriminator described in Section 3.4.1.

The output of the DSM, $i_m[n]$ (Fig. 3.12(b)), is the same number of bits, d , as the DSM quantizer. The words are clocked at a high rate, T/N . The decimator converts the high rate,

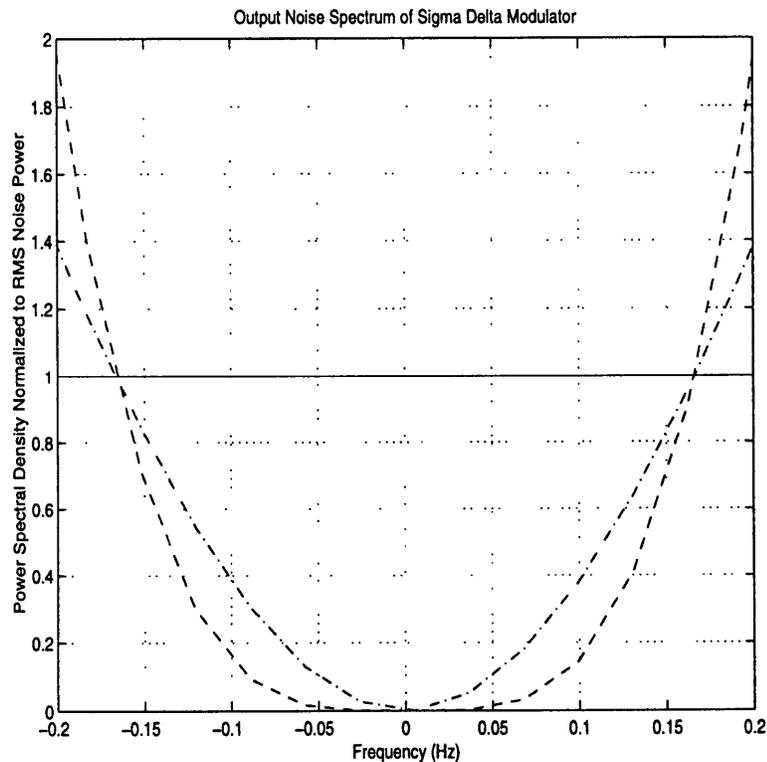


Figure 3.15: Quantization noise shaping. Solid line represents no noise shaping. Dashed-dotted line represents first-order noise shaping. Dashed line is second-order shaping.

d -bit data, $i_m[n]$, into low-rate, $(d+k\log_2N)$ -bit data, $i[n]$, where k is the order of the decimator which is set to be the same order (number of feedback loops) as the DSM [16].

In Section 3.4.2, the conventional A/D converter needed a 32-bit quantizer to achieve the desired velocity resolution. Therefore, employing Fig. 3.12(b) as the A/D converter, $d+k\log_2N=32$ bits were necessary to achieve the same velocity resolution. Table 3.1 lists different combinations of d , k , and N that can result in 32-bit DSM A/D conversion. For a modulator loop order of $k=2$, using a 12-bit quantizer requires an oversampling ratio of 1024. Therefore, for a sampling rate of 10 kHz, then an oversampled frequency of approximately 10 MHz would be necessary. Recall that using only 1-bit of oscillator output amplitude information, the PM demodulator schemes in Section 3.2 could achieve 0.02-ft/sec velocity resolution by sampling at 24 MHz. Therefore, using the DSM to quantify the amplitude with 12-bits instead of 1-bit only decreases the necessary sampling rate by approximately one-half, from 24 MHz to 10 MHz.. However, looking at the case in Table

Table 3.1: Theoretical Delta Sigma Modulator Parameter Values

modulator loop order k	over-sampling ratio N	bits in quantizer d	modulator loop order k	over-sampling ratio N	bits in quantizer d
2	64	20	4	16	16
2	128	18	4	32	12
2	256	16	4	64	8
2	512	14	4	128	4
2	1024	12	4	256	1

3.1 where $k=4$, $N=64$, and $d=8$, a fourth-order DSM A/D converter, oversampled by $N=64$, results in a sampling frequency of about 640 kHz, where the minimum sampling rate is again assumed to be 10 kHz. By quantizing the amplitude of the filtered beat note by 8-

bits instead of 1-bit with the PM demodulators, the necessary sampling rate can be reduced from 24 MHz to 640 kHz, nearly a factor of 100 improvement. This improvement is what was hoped for in Section 3.3.2, and it is clear that using a DSM is necessary for implementing a CDM FM receiver as an accelerometer readout.

Equation 3.34 can be rewritten as follows:

$$\Delta a = S_1 S_2 \left(|e_{diff}[n]|_{max} + \frac{2\pi(k_1|a_1(nT)|_{max} + k_2|a_2(nT)|_{max})}{2^{d+k\log_2 N}} \right) \quad (3.55)$$

Using a DSM A/D converter, in theory, has the flexibility to trade off sampling rate with quantizer resolutions to achieve the desired acceleration and velocity resolution. Oversampling in the A/D conversion does not affect the differentiator error $e_{diff}[n]$, because the decimator reduces the sampling frequency so that the differentiator filters the data at the same rate as when using conventional A/D conversion. Increasing N also does not affect the maximum acceleration signals $|a_i(t)|_{max}$ and $|a_i(t)|_{max}$. Therefore, (3.55) is exactly (3.34), except with $b=d+k\log_2 N$.

3.5 Summary

Chapter 3, primarily addressed Question 3 of the Introduction and performed first-order analyses to estimate the performance of the device. It was estimated that using a fourth-order DSM in conjunction with the CDM demodulator could theoretically attain 0.02 ft/sec velocity and $1\mu g$ acceleration resolution over a duration of 360 seconds.

This chapter contains purely theoretical developments and results. Very few implementation issues were considered. For instance, delta-sigma modulators have many practical limitations, especially for high resolution A/D conversion. For high precision instruments, often the performance of certain electronic components provide the fundamental constraints on the instrument. Nevertheless, theoretical results provide a bound on

the best resolution achievable, and are an indication of what design goals should be set for implementation efforts.

While further analysis of the phase demodulators and velocity readouts is beyond the scope of this thesis, further scrutiny of the CDM demodulation scheme will follow, as well as consideration of practical issues with regards to its design and implementation.

Chapter 4

Acceleration Readout Design and Simulation

Can a digital FM receiver achieve the readout resolution desired of the VIA? While this is the central question that drives Chapter 4, it has an overly simplified phrasing that demands so much more than a simple answer. A mere “yes” or “no” response immediately prompts the questions “how?” and “why not?” which will lead inevitably to statements that must be proven and doubted and questioned, repeatedly, until the final answer is nothing less than a sixty-page exposition that renders a binary response to the central question virtually meaningless. For, in answering the initial question and examining the validity of the derivations in Chapter 3, many unanswered issues arise. At once the value and beauty of a Master’s thesis, and at once its plague, this spawning of questions sheds new light on the research matter by opening and closing both old and new doors in a struggle to tame the question. While addressing the facets of design and feasibility of using the QCDM demodulator as a VIA readout, Chapter 4 seeks not only to provide answers, but to raise the central question to a new level.

While Chapter 3 discussed a variety of phase and frequency readout methods, Chapter 4 focuses entirely on the design and simulation of a single frequency demodulator. Since much of the system is implemented with digital signal processing, the design and simulation of the subsystems are intimately connected, and simulations will often be used extensively to design the subsystems. First, Section 4.1 will briefly give both a functional and a detailed overview of the entire Vibrating Island Accelerometer system. Then, in Section 4.2, the subsystems will be designed and simulated in a cumulative fashion. That is to say, the first subsystem will be simulated by itself; the second subsystem will be analyzed in isolation, then simulated together with the first; the third will be designed independently,

then tested collectively with the first two; and so on and so forth until Section 4.3, which simulates the entire system. Attention is paid to the design and analysis of the subsystems to facilitate the understanding of how individual parameters limit the overall system. Additionally, since the simulations primarily make repeated use of one set of assumed VIA performance specifications, if the specifications were to change during the course project, the readout would have to be redesigned and resimulated. Thus, the scrutiny of the subsystems will hopefully make their redesign much easier. Lastly, Section 4.4 summarizes the effects of each parameter on the readout resolution, and discusses additional implementational issues.

The response to “how?” commences next.

4.1 System Description

4.1.1 Functional Description

The cross-differentiate multiply structure of the baseband FM demodulator, shown in boxed section of Fig. 4.1, has been developed in the context of FM and televisions receivers. Song and Lee [12] have simulated an all-digital implementation of the demodulator, and discussed how hi-fi FM receivers are concerned with introducing low phase distortion, as is the VIA readout scheme. However, while conventional FM broadcasting receivers are particularly concerned with intermediate frequency (IF) bandpass filtering to reject adjacent channels, the VIA readout does not need to protect against distortion from other channels. Another difference lies in the demodulation scheme. In an FM broadcasting receiver, a signal is received and then demodulated into its inphase and quadrature components with a local oscillator, in a superheterodyning implementation. For the VIA readout, instead of one FM signal to demodulate, two signals, $s_1(t)$ and $s_2(t)$, defined as in (3.15) and (3.16) are available to the receiver. Furthermore, as seen in Chapter 2, it is known that the signals

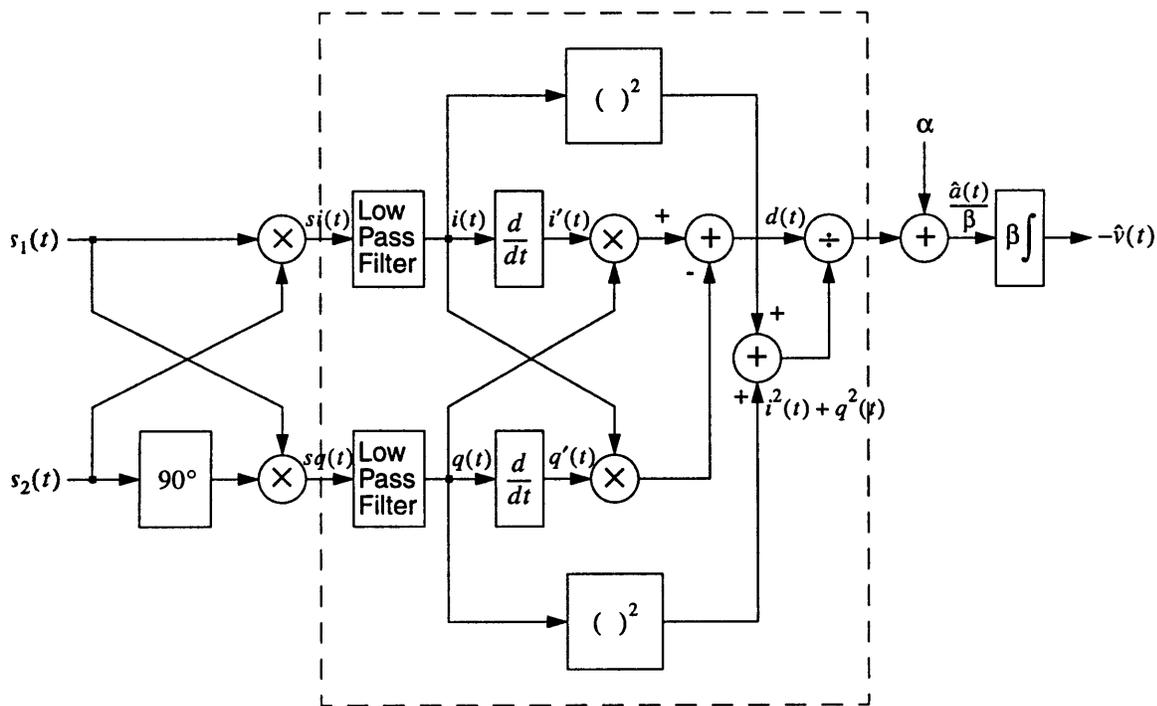


Figure 4.1: Functional block diagram of readout scheme.

are inversely modulated by the same acceleration signal. These observations lead, in part, to one of the most intriguing attributes of this design: the quadrature demodulation.

One of the reasons why quadrature demodulation is used often in communications systems is that it allows a receiver to distinguish the sign of a demodulated signal. For the VIA, it seems, at first, that the oscillator carrier frequencies must be offset from one another to eliminate the ambiguity of the sign of the beat note between them. If the carrier frequencies, f_1 and f_2 , were identical, acceleration in one direction and acceleration of the same magnitude in the opposite direction, would produce the same beat note. This is one of the reasons why the carrier frequencies would have to be offset from one another if the demodulation scheme were implemented as in Fig. 3.6b. The quadrature demodulation scheme in Fig. 4.1, on the other hand, allows the carrier frequencies to be the same because it does not require an offset between them to distinguish between positive and

negative acceleration. There are a couple of reasons for offsetting the carrier frequencies, as discussed in Chapter 2. The important point here is that the quadrature scheme eliminates one of these reasons, and takes one step closer to relaxing the requirements placed on the VIA crystal oscillator.

A second attribute of the quadrature demodulation is that it eliminates the need for a local oscillator. Since the VIA oscillators must be highly stable to achieve good resolution, a local oscillator used to demodulate $s_1(t)$ and $s_2(t)$ would have to be highly stable as well. Since the quadrature demodulation in Fig. 4.1 uses the oscillator outputs to demodulate themselves, a third highly stable oscillator is unnecessary.

Since the readout scheme includes a quadrature demodulation stage cascaded with the traditional cross-differentiate multiply demodulation scheme, it will be called the Quadrature demodulation and Cross-Differentiate Multiply (QCDM) demodulator.

The functional analysis of the QCDM demodulator is fairly straightforward, and is performed here for the standard model of the oscillator outputs described in (3.15) and (3.16). Assume, then, that the oscillator outputs are defined as follows:

$$s_1(t) = S_1 \cos\left(2\pi f_1 t + 2\pi k_1 \int_0^t a(t) dt\right) \quad (4.1)$$

$$s_2(t) = S_2 \cos\left(2\pi f_2 t - 2\pi k_2 \int_0^t a(t) dt\right) . \quad (4.2)$$

After the quadrature demodulation stage,

$$\begin{aligned} si(t) &= S_1 \cos\left(2\pi f_1 t + 2\pi k_1 \int_0^t a(t) dt\right) S_2 \cos\left(2\pi f_2 t - 2\pi k_2 \int_0^t a(t) dt\right) \\ &= \frac{S_1 S_2}{2} \cos(2\pi(f_1 + f_2)t) + \frac{S_1 S_2}{2} \cos\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right) \end{aligned} \quad (4.3)$$

$$\begin{aligned}
sq(t) &= -S_1 \cos\left(2\pi f_1 t + 2\pi k_1 \int_0^t a(t) dt\right) S_2 \sin\left(2\pi f_2 t - 2\pi k_2 \int_0^t a(t) dt\right) \\
&= -\frac{S_1 S_2}{2} \sin(2\pi(f_1 + f_2)t) + \frac{S_1 S_2}{2} \sin\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right). \quad (4.4)
\end{aligned}$$

Filtering out high frequency components yields the in-phase and quadrature signals:

$$i(t) = \frac{S_1 S_2}{2} \cos\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right) \quad (4.5)$$

$$q(t) = \frac{S_1 S_2}{2} \sin\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right). \quad (4.6)$$

The derivatives of $i(t)$ and $q(t)$ are

$$i'(t) = -2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1 S_2}{2} \sin\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right) \quad (4.7)$$

$$q'(t) = 2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1 S_2}{2} \cos\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right). \quad (4.8)$$

Cross multiplying and then differencing the products,

$$\begin{aligned}
d(t) &= -2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1^2 S_2^2}{4} \left(\sin\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right) \right)^2 \\
&\quad - 2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1^2 S_2^2}{4} \left(\cos\left(2\pi(f_1 - f_2)t + 2\pi(k_1 + k_2) \int_0^t a(t) dt\right) \right)^2 \\
&= -2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1^2 S_2^2}{4} [(\sin(\))^2 + (\cos(\))^2] \\
&= -2\pi[(f_1 - f_2) + (k_1 + k_2)a(t)] \frac{S_1^2 S_2^2}{4}. \quad (4.9)
\end{aligned}$$

The sum of the squares of $i(t)$ and $q(t)$ is

$$\begin{aligned}
i^2(t) + q^2(t) &= \frac{S_1^2 S_2^2}{4} ((\cos(\))^2 + \sin(\))^2 \\
&= \frac{S_1^2 S_2^2}{4}. \quad (4.10)
\end{aligned}$$

Therefore, after the division the acceleration term appears as

$$\frac{d(t)}{i^2(t) + q^2(t)} = -2\pi(f_1 - f_2) - 2\pi(k_1 + k_2)a(t). \quad (4.11)$$

In the calibration of the accelerometer, α and β can be measured for zero acceleration.

For

$$\begin{aligned} \alpha &= 2\pi(f_1 - f_2) \\ \beta &= \frac{-1}{2\pi(k_1 + k_2)}, \end{aligned} \quad (4.12)$$

$$\hat{a}(t) = a(t) \quad \text{and} \quad \hat{v}(t) = \int_0^t a(t) dt. \quad (4.13)$$

Note that in Section 4.2, the carrier frequencies f_1 and f_2 will be assumed to be equal to simplify the simulations. It is assumed that offset between them could be accounted for by adding α as in Fig. 4.1.

4.1.2 Detailed Description

In Chapter 2, a block diagram for the VIA oscillator was disclosed and discussed. In Section 4.1.1, a functional block diagram of the QCDM was the topic of interest. Now, the two blocks are juxtaposed in Fig. 4.2 to reveal the VIA system block diagram.

In Section 4.1.1, a mathematical analysis was performed on an infinite precision continuous-time model of the QCDM demodulator. Fig. 4.2 shows the additional considerations necessary for a practical digital implementation of the scheme. First, the analog-to-digital conversion must take place, which necessitates the inclusion of an anti-aliasing filter, $H_{aa}(j\omega)$. Second, implementation of a digital system requires consideration of the sampling rate, f_s , and changes in the sampling rate. Third, the mathematical operations of differentiation and integration in the continuous-time domain are implemented in the discrete-time domain with digital filters $H_{diff}(e^{j\omega})$ and $H_{int}(e^{j\omega})$. Fourth, as will be

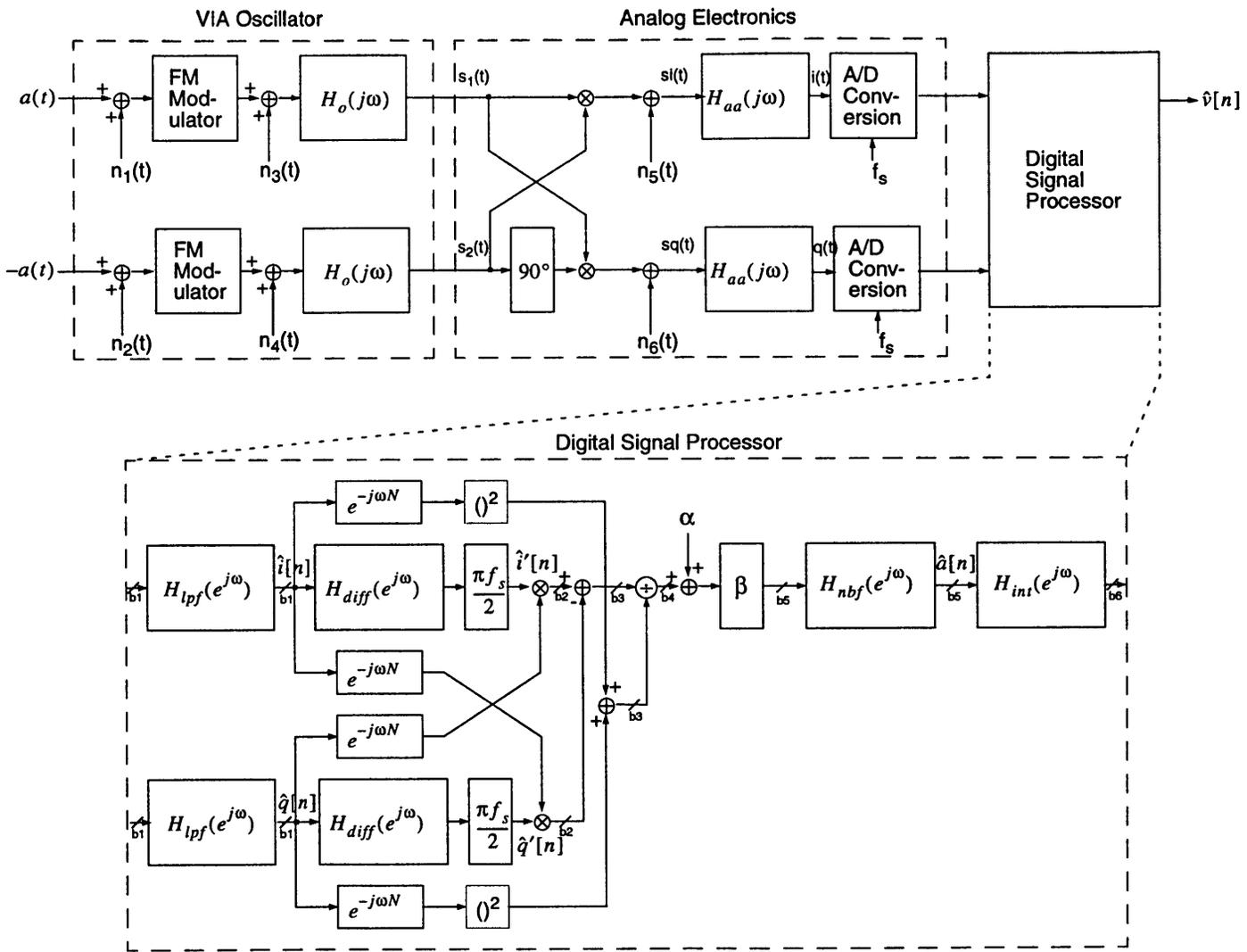


Figure 4.2: Detailed block diagram of digital QCCDM demodulator readout.

discussed in Section 4.2.1, since digital FIR filters introduce time delays, non-filtered data paths must be delayed with $e^{-j\omega n}$ blocks. Lastly, the wordlengths of the data paths are an important issue for fixed point implementations. Although some of these details are obvious, while others are less so, they are highlighted here simply to bridge the gap between the mathematically demonstrated scheme in Fig. 4.1 and the equivalent analog and digital design in Fig. 4.2.

The important system-level design consideration is the determination of the sampling rate, f_s . The constraint on f_s comes from $i(t)$ and $q(t)$ in (4.5) and (4.6), which are the signals that are A/D converted. Section 4.2.3 will show that for a high resolution readout, the Nyquist frequency, f_N , of $i(t)$ and $q(t)$, must be $f_N=2.4$ kHz for $f_1=f_2$. Therefore, the minimum sampling rate to avoid aliasing is $\min\{f_s\}=4.8$ kHz. To leave room for carrier offsets, where $f_1 \neq f_2$, the sampling rate is chosen to be $f_s=10$ kHz. For this sampling frequency, the maximum offset of the carriers is

$$|f_1 - f_2| < \left| \frac{f_s}{2} - f_N \right| = |5\text{kHz} - 2.4\text{kHz}| = 2.6\text{kHz}. \quad (4.14)$$

In the simulations, it will be assumed that $f_1=f_2$ so that $\alpha=0$. Note, also, that high frequency acceleration signals detected by the transducer, such as vibration transients, will be filtered out by the anti-aliasing filter, and then further attenuated by $H_{lpf}(e^{j\omega})$. Finally, since the acceleration and velocity signals are expected to be bandlimited to $f_a=100$ Hz, the integrated acceleration can be downsampled to output $\hat{v}[n]$ at least 500 Hz.

Section 4.2 will discuss the design and analysis of the demodulator subsystems independently, along with increasingly large groupings of the subsystems. First, the design of the differentiator is considered. Then, the CDM structure is simulated. Analysis and study of the constraints on the FM bandwidth of $\hat{i}[n]$ and $\hat{q}[n]$ comes next, and is followed by the design of the low pass filter $H_{lpf}(e^{j\omega})$ and narrowband filter, $H_{nbf}(e^{j\omega})$. Then, the numer-

ical integration filter, $H_{int}(e^{j\omega})$, is designed and simulated. Lastly, the oscillator and analog component noise models are added to the simulation.

This thesis project has designed, on a system level, the proposed QCDM demodulator readout scheme for the VIA. The analysis and simulation, however, are concentrated on the Digital Signal Processing block of the system. The VIA oscillators and the analog electronics are treated and simulated with relatively simple noise models and surface-level designs. In particular, the Delta-Sigma Modulator of the A/D converter is a significant factor in constraining the overall readout performance. Section 4.3 will only suggest possible directions for performing a more detailed analysis and simulation of DSMs.

4.2 Subsystem Design and Simulation

4.2.1 Discrete-Time Differentiator Design

This section begins by discussing general digital filter design issues. Then, discussion turns to the design of the discrete-time differentiators of the readout.

In designing the digital filters for the demodulator, including the differentiator discussed in this section, three key decisions must be made. First, whether to design an infinite impulse response (IIR) or finite impulse response (FIR) filter must be considered. Then, the implementational structure of the filter, whether direct form I or II, cascade, or parallel, has to be decided upon. Lastly, the type of coefficient quantization, whether truncation or rounding, should be determined, as well as the type of binary arithmetic implemented, fixed point or floating point.

A digital filter can be described in the discrete-time domain as either a linear constant coefficient difference equation or a state space representation [5]. This section considers the former description, which can be expressed mathematically as follows:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{l=0}^M b_l v[n-l] \quad a_0 = 1. \quad (4.15)$$

The corresponding transfer function in the z-transform domain can be written as a ratio of polynomials:

$$H(z) = \frac{\sum_{l=0}^M b_l z^{-l}}{1 + \sum_{k=1}^N a_k z^{-k}}. \quad (4.16)$$

Another z-transform description (among many) of a difference equation can be expressed in terms of the transfer function poles and zeros:

$$H(z) = b_0 \cdot \frac{\prod_{l=1}^M (1 - z_l z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})}. \quad (4.17)$$

Equation 4.17 begins discussion of the differences between FIR and IIR filters. The key distinction between the two categories of filters is that FIR filters do not have poles, and so the denominator of (4.17) is unity and the a_k 's of (4.16) are zero. Therefore, whereas stability is an important design constraint for IIR filters, FIR filters are inherently stable. However, IIR filters generally perform better for real-time applications because they usually can be implemented more efficiently, requiring lower filter order and thus fewer coefficients than an FIR filter having similar magnitude performance. While this is an important consideration for the VIA readout, another significant consideration is the phase distortion, or group delay, of the filter, which directly affects the resolution of the accelerometer. While a zero-phase filter response for this application is ideal but impossible for causal systems, FIR filters can have linear phase which results in constant group delay. Therefore, while FIR filters introduce time delays, they do not nonlinearly distort

the phase of an input signal as most IIR filters do. Choosing between implementing FIR or IIR filters, then, amounts to making a choice between minimum allowable processing speed and maximum possible resolution. Since the latter result is of particular interest for this theoretical study, FIR filters are used in the design of the QCDM demodulator.

In addition to the polynomial (4.16) and pole-zero (4.17) expressions, two other mathematical descriptions of digital filters are the partial fraction expansion of (4.17) and the aforementioned state space form. Each one of these four mathematical descriptions of a digital filter leads to a different structural implementation of the filter. The polynomial expression naturally describes what are called the direct form structures I and II, the pole-zero description leads to the cascade structure, the partial expansion form reflects the parallel structure, and the state space representation can be used to conveniently design structures based on state variables and linear transformations.

For IIR systems, the cascade, parallel, and normalized lattice implementations are generally more robust to errors from fixed point coefficient quantization, a very practical consideration, whereas the direct forms are less so and rarely used to implement filters with order larger than two. With regard to IIR digital filter roundoff noise, which stems from the quantization of products, the cascade structure has the best performance because it can be implemented with fewer multipliers. For FIR systems, the direct form (direct form I and II turn out to be the same structure for FIR filters, but not for IIR filters) is used most often because it is least sensitive to coefficient quantization; the quantized zeros are uniformly spread in the z -plane and are less sensitive to the quantization noise. Furthermore, the direct form more easily preserves the linear phase characteristic of FIR filters. For these reasons, and the simplicity of direct form analysis in Matlab, this structure was chosen to implement the digital filters in the design and simulation of the QCDM demodula-

tor. A much more detailed analysis can be done to justify this choice and might be useful for future efforts.

For FIR filters, setting $a_k=0$ for $k>0$, the difference equation in (4.15) reduces to

$$y[n] = \sum_{l=0}^M b_l v[n-l]. \quad (4.18)$$

Making a substitution of variables, a system output, $y[n]$, can be written as a convolution of an impulse response, $h[n]$, and an input, $x[n]$:

$$y[n] = \sum_{k=0}^M h[k]x[n-k] \quad (4.19)$$

Figure 4.3 shows a flowgraph representation of an infinite precision direct form realization of an FIR filter. The ‘filter’ command in Matlab implements the transposed form illustrated in Fig. 4.3(b), which is functionally equivalent to Fig. 4.3(a).

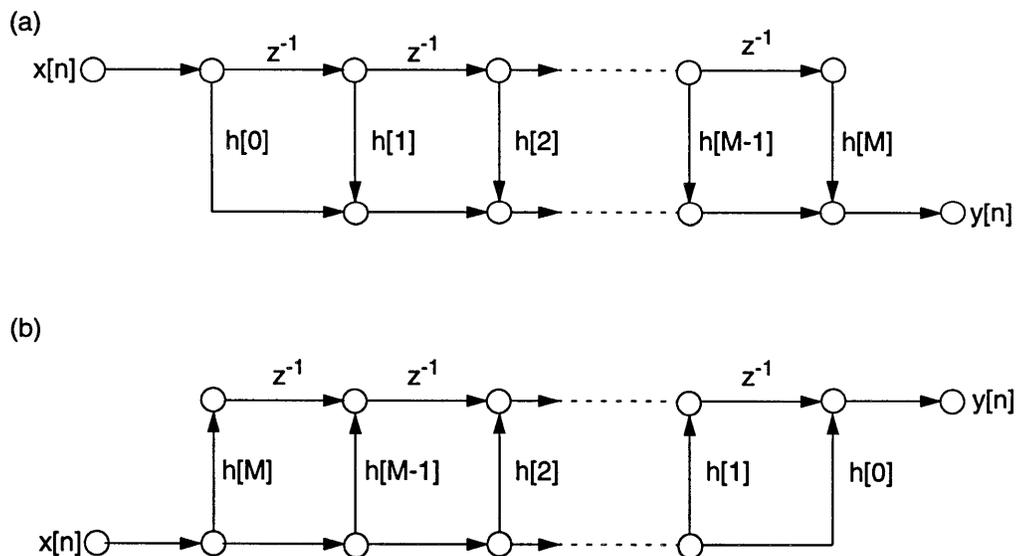


Figure 4.3: (a) Direct form, (b) transposed direct form realization of an FIR system.

Fixed-point arithmetic is usually implemented with 2's complement numbers. The value represented by a fixed-point number is normally scaled so that it is a fraction represented by a fixed, finite wordlength. Therefore, the product of two fixed-point numbers would also be a fraction, and will never be greater than one. Rounding off a fixed-point number eliminates the least significant bits, and introduces quantization error that can be modeled as white noise that is independent of the signal. Summing two fixed-point numbers, however, can introduce register overflow, which can be avoided by appropriately scaling the input. For direct form realizations of FIR filters using fixed-point, only the output of the system needs to be prevented from overflowing, because all other sums are partial sums and their overflowing does not affect the output. To achieve wider dynamic range and lower quantization error simultaneously, the wordlength must be increased.

Floating-point arithmetic represents a number by quantifying its mantissa and an appropriate scale factor. One of the greatest advantages of floating-point realizations is that it has high dynamic range. But while overflow is rare, rounding errors accrue after both addition and multiplication. Furthermore, floating-point systems are more difficult to analyze because the quantization noise is no longer white and independent of the signal.

For the QCDM, only FIR filters are implemented and no feedback is employed. Therefore, the overflow problem can be avoided, without too much difficulty, by proper scaling. Additionally, the dynamic range of the device will be known, and a fixed level of resolution is desired throughout the range; note that $i(t)$ and $q(t)$ in (4.5) and (4.6) are periodic and have limited amplitudes. Whereas the quantization error of floating-point numbers scale up and down with signal level, fixed-point numbers have the same quantization error, and thus resolution, throughout its dynamic range. For these reasons, a fixed-point realization is used for this theoretical study and implementation of the QCDM demodulator. Coefficient quantization and round off noise will be considered further in Section 4.3.

Having just chosen to design FIR filters and to implement them with direct form realization and fixed-point arithmetic, attention turns now to the selection of digital filter design technique. Two common FIR filter design methods are window-based and optimum equiripple design techniques.

In digital signal processing terminology, windowing is the act of truncating an infinite-length discrete-time signal by multiplying it by a finite length signal called a window. Calling the window length “finite” can be somewhat deceiving, for the window sequence is, in theory, an infinite length sequence consisting of all zero-valued samples outside of a group of adjacent nonzero samples. The number of nonzero samples is generally equivalent to the length of the window, and it is normally only the group of nonzero samples that are referred to as the “window.” In short, the zeros are ignored, but assumed to cancel out samples of the infinite-length discrete-time signal through multiplication.

In the window-based design for low pass filters, an ideal impulse response, which is a sinc in the time domain, is multiplied by a window. In the frequency domain, this corresponds to convolving the ideal impulse with the Fourier transform of the window. The convolved waveform is subtracted from the ideal frequency response to derive a measure of error. Minimizing the mean square error is the design objective, and can be accomplished for a certain window length constraint by varying the shape of the window. Common window types are rectangular, Bartlett, Hamming, Hanning, Blackman, and Kaiser, which has variable parameters.

For a high resolution application, the minimum mean square error criterion is not ideal because it does not say anything about the maximum error of the designed filter. The optimum equiripple filter design, on the other hand, is based on the minimization of maximum error criterion. This design technique ensures that the filter error is smaller than a specified

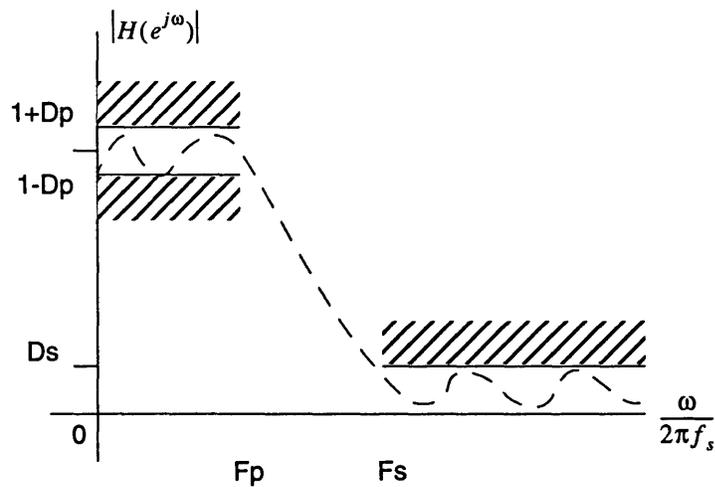


Figure 4.4: Low pass filter design specifications.

level, which makes it particularly suitable for designing high resolution systems. For this reason, the optimum approximation technique is used to design the QCDM FIR filters.

Optimum filter design is based on the Parks-McClellan Algorithm [5], an iterative algorithm that designs the minimum length filter for specified passband (F_p) and stopband (F_s) cutoff frequencies, as well as maximum allowable passband (D_p) and stopband (D_s) amplitude ripple. Fig. 4.4 shows the filter specifications that the algorithm requires for the design of a low pass filter. Section 4.2.4, will demonstrate how to generate the filter coefficients of a low pass filter from the above specifications using Matlab's Remez Algorithm. The Remez Algorithm can also be used to design discrete-time differentiators, as seen in Section 3.4.2, using the following command:

```
hdiff=remez(N,[FL FH],[AL AH], 'differentiator');
```

In this case, N is the desired filter order number, FL is the minimum frequency, FH is the maximum frequency for which a desired level of resolution must be obtained, and AL and AH are the desired filter amplitudes corresponding to FL and FH . The specified frequencies are normalized to the sampling frequency. Therefore, for successively larger

sampling frequencies, FH is successively lower, and the filter order is smaller. This will be elaborated on with the help of an example.

First, consider the differentiator block shown in Fig. 4.5. The objective of this section turns now to the characterization of the differentiator block and to design a discrete-time differentiator having low magnitude error. Note that the scale factor block is necessary to achieve the correct derivative from differentiator, which is designed using a Remez script.

SIM1(M), in Appendix A.1, designs a discrete-time differentiator of length $M+1$ using the Remez Algorithm. The function plots the differentiator impulse response, magnitude response, and magnitude error. Additionally, the function returns the maximum magnitude error of the differentiator for a given bandwidth and sampling frequency of an input signal. Lastly, the simulation also examines a test case by differentiating an FM input signal, $i(t)$, based on the VIA oscillator parameters. By subtracting the filtered output, $i'(t)$, from the ideal filtered output, a measure of error can be derived. The simulation is repeated with different values of M until the maximum output error is less than $10 \mu g$, $1 \mu g$, and $0.1 \mu g$. It is assumed, at this point, that these resolutions will be preserved by the rest of the CDM structure. The next section will look at how good an assumption this is. The differentiators generated for these error levels are the differentiators that will be used in the rest of the

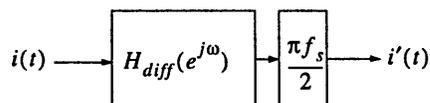


Figure 4.5: Differentiator block.

simulations. The test case is chosen to model the VIA oscillator signals using parameters, listed in Section 3.3, that will presumably result in the largest acceleration error. The assumption is that if the differentiators are designed using this worst-case test scenario, the differentiator will perform at least as well and, in most cases, better, than it performs using the test case. This assumption will be validated in Section 4.3.

The input signal, $i(t)$, is defined in SIM1(M) as follows:

$$i(t) = S_1 S_2 \cos\left(\left(\frac{k_1 A}{f_a}\right) \sin(2\pi f_a t)\right) = S_1 S_2 \cos\left(2\pi k_1 \int_0^t a(t) dt\right) \quad \text{where} \quad (4.20)$$

$$a(t) = A \cos(2\pi f_a t)$$

Note that $i(t)$ here is in the same form as $i(t)$ in (4.5). The parameter values are $S_1=S_2=1$, $A=20$ g, $k_1=50$ Hz/g, $f_a=100$ Hz, $f_1=f_2$, and $f_s=10$ kHz. The input is shown in Fig. 4.6a., and it's spectrum is show in Fig. 4.6b. Note that the sidebands are located every $2f_a=200$ Hz, whereas the sidebands in Fig. 3.5 were spaced only 100 Hz apart.

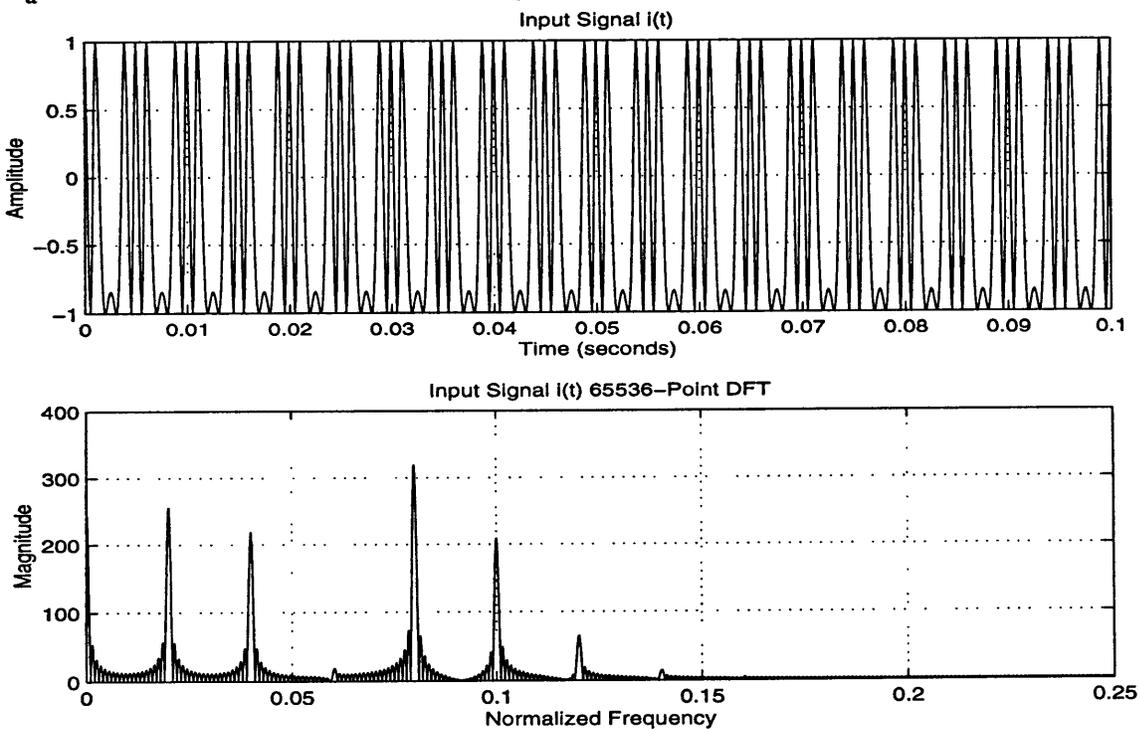


Figure 4.6: Input $i(t)$ and its magnitude response.

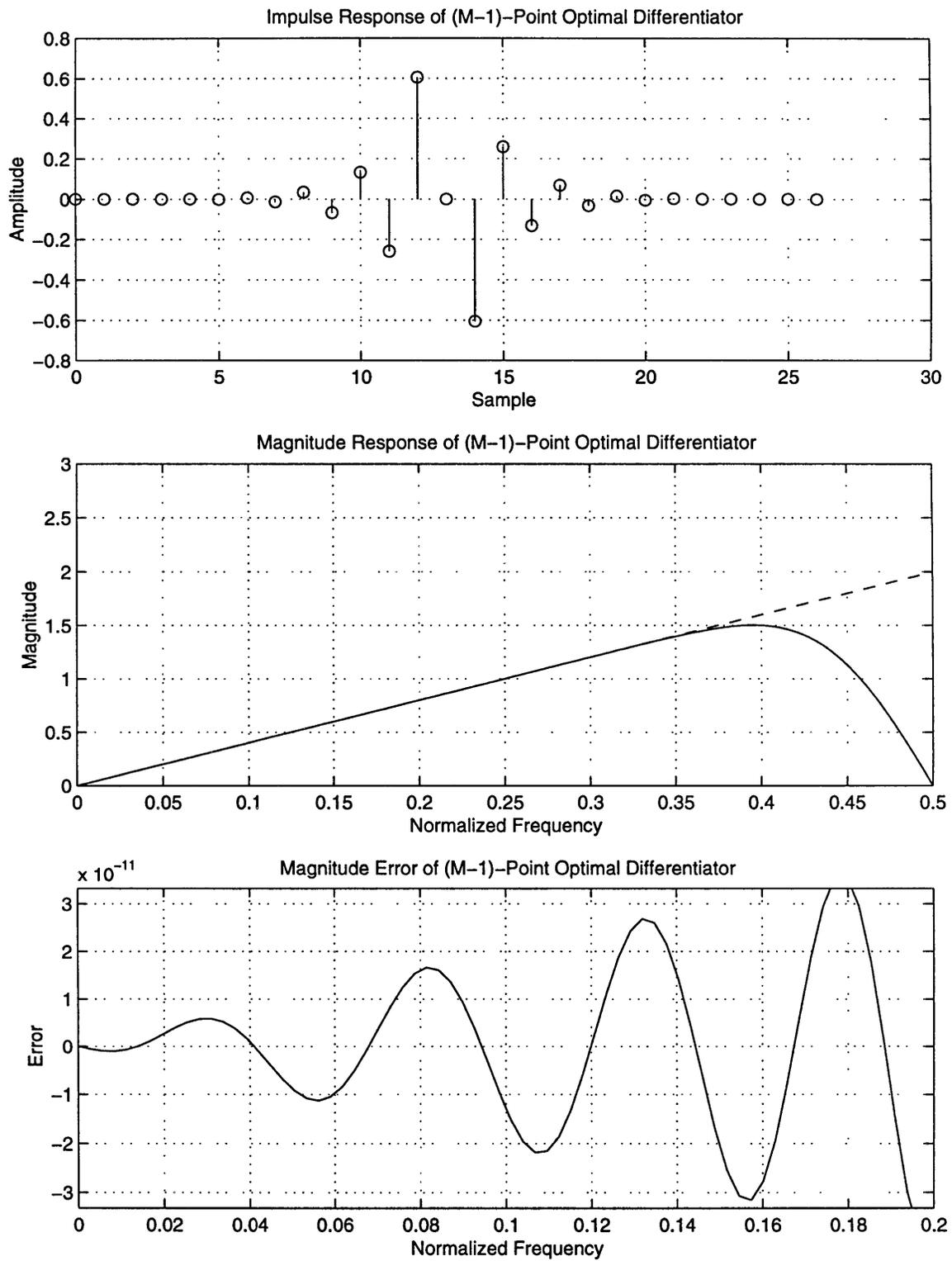


Figure 4.7: Differentiator characteristics for $M=26$.

The difference is that $i(t)$ is at baseband, while $s_1(t)$ was centered around carrier frequency $f_1=10$ kHz.

A differentiator, $H_{diff}(e^{j\omega})$, of order $M=26$ was found to meet the $1\mu g$ resolution level. The impulse response, $h_{diff}[n]$, the magnitude response, and magnitude error are show in Fig. 4.7. Since M is even, the differentiator is a Type III FIR filter, and has an integer delay $M/2$. By the Carson Limit calculated in Section 3.3.1, the baseband Nyquist frequency, f_N , for $s_1(t)$ is approximately 1.1 kHz. The parameters used to calculate the Carson Limit are the same for $i(t)$, and so it is expected that the largest frequency component of interest in $i(t)$, f_N , is roughly 1.1 kHz. Since the sampling frequency is 10 kHz, $f_N/f_s=0.11$. Fig. 4.7c. shows that for a normalized input frequency of 0.11, the magnitude error is approximately 2×10^{-11} . Scaling this error by $\frac{\pi f_s}{2}$, an estimate for $|e_{diff}[n]|_{max}$ can be obtained:

$$|e_{diff}[n]|_{max} = \left(\frac{\pi f_s}{2}\right) \cdot (1.660 \times 10^{-11}) = 2.6078 \times 10^{-7}. \quad (4.21)$$

Simulating the direct form implementation of $H_{diff}(e^{j\omega})$, $i(t)$ yields the output, $i'(t)$, displayed in Fig. 4.8. The ideal derivative of (4.20) is

$$i'_{ideal}(t) = (-Ak_1 2\pi) \cos(2\pi f_a t) \left[S_1 S_2 \sin\left(\left(\frac{k_1 A}{f_a}\right) \sin(2\pi f_a t)\right) \right]. \quad (4.22)$$

The difference between $i'_{ideal}(t)$ and $i'(t)$ is the error shown in Fig. 4.9, where the maximum error is 2.3790×10^{-7} . Note that the large output error between time $t=0$ and $t=0.005$ seconds is the result of the start-up transient that would occur harmlessly during the calibration of the instrument, and is not included in the maximum error estimate. The simulation results for $10\mu g$ and $0.1\mu g$ resolution are shown in Table 4.1.

The results suggest that the output error of a discrete-time differentiator can be approximated by finding the differentiator magnitude error corresponding to the Nyquist

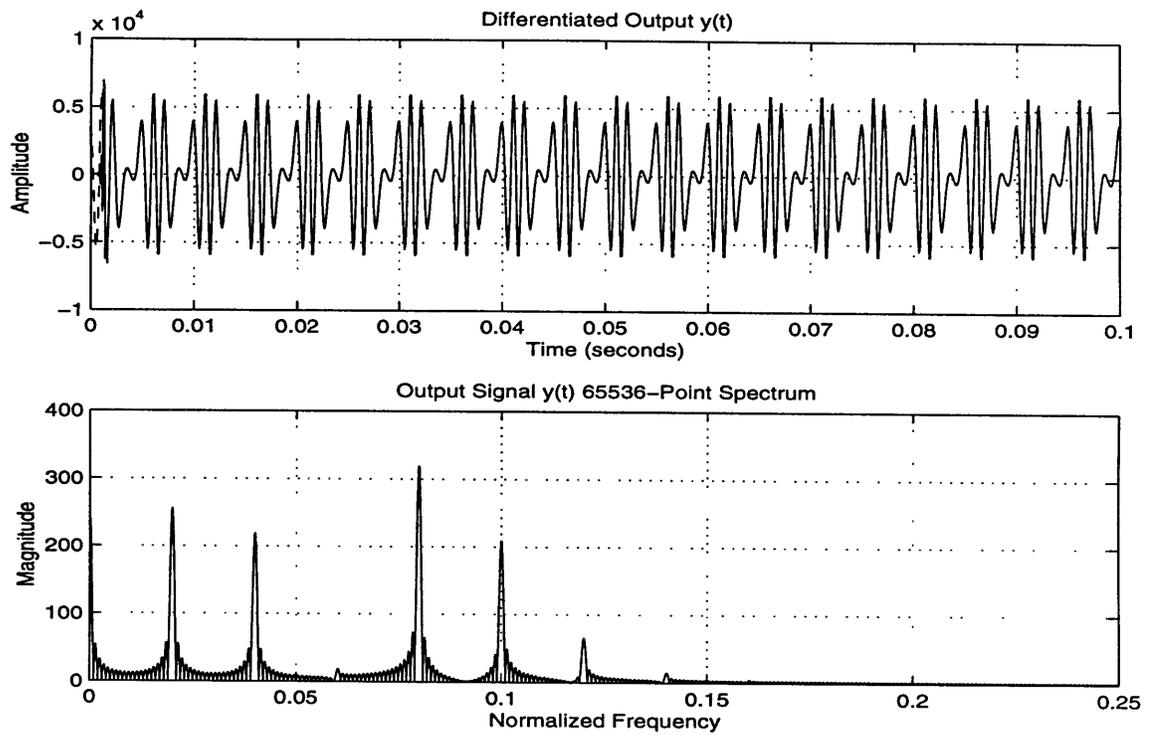


Figure 4.8: Differentiated $i(t)$.

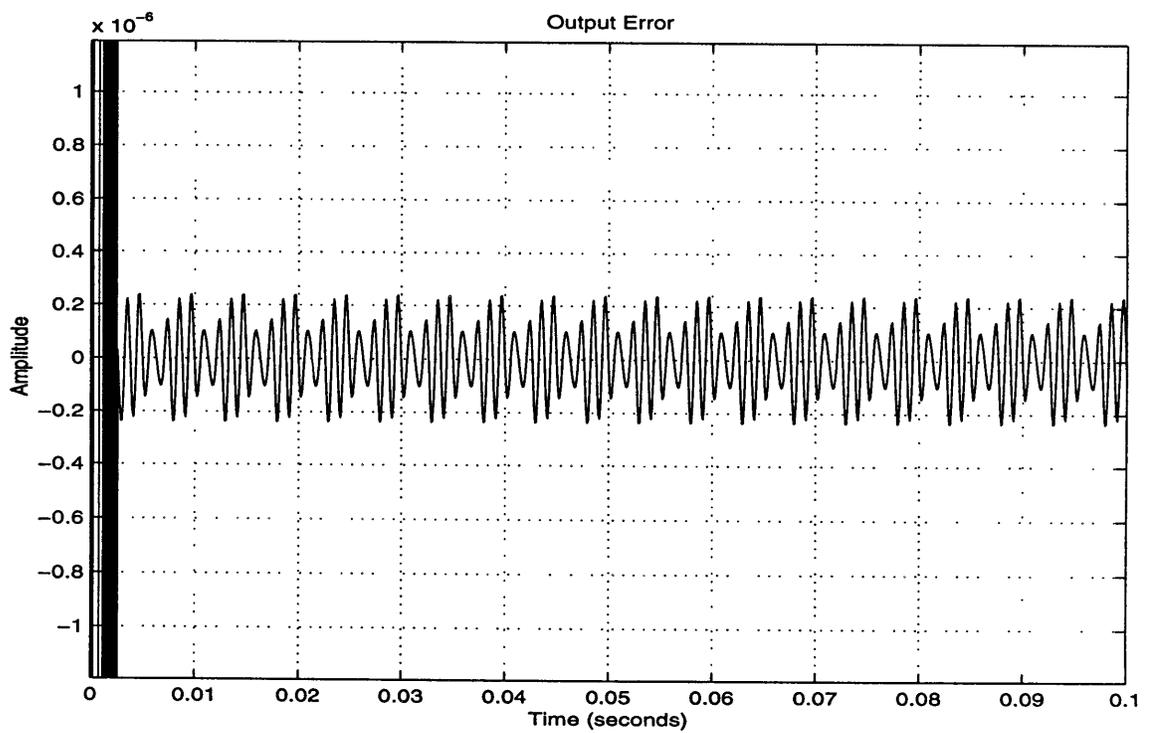


Figure 4.9: Amplitude error introduced by differentiator.

Table 4.1: SIM1(M) results for M=22, 26, and 28

M	$H_{\text{diff}}(e^{j\omega})$ Max. Error	Scaled Max. Error	Max. Accel. Error (g)
22	7.14×10^{-10}	1.12×10^{-5}	6.73×10^{-6}
26	1.66×10^{-11}	2.61×10^{-7}	2.38×10^{-7}
28	2.71×10^{-12}	4.26×10^{-8}	4.89×10^{-8}

frequency component of the input, using a graph like Fig. 4.7c., and scaling this error by $\frac{\pi f_s}{2}$. Simulated results support the derived estimate in Section 3.4.2.

This is, of course, only a means of approximation and a rule of thumb at best. The accuracy of the approximation depends on the accuracy of f_N estimate, which was based on the Carson Limit. In Section 4.2.3 the Carson Limit will be shown, in fact, to be an inadequate indicator of necessary FM bandwidth for a high resolution readout.

The recommended design procedure is to estimate the differentiator performance by using its magnitude error, and to use SIM1(M) to design the differentiator and to better predict its performance.

4.2.2 Cross-Differentiation Multiply Structure

In the previous section, the differentiator block was characterized with differentiators of order 22, 26, and 28 shown to introduce errors on the order of $10 \mu\text{g}$, $0.2 \mu\text{g}$, and $0.04 \mu\text{g}$. This section simulates the cross-differentiate multiply structure of the demodulator, seeking to verify the mathematical analysis in Section 4.1.1, while demonstrating how much the differentiator error propagates through multipliers and adders to affect the final acceleration readout (round off errors will be considered in Section 4.3). Up until this point, the acceleration resolution was estimated by the error at the output of the differenti-

ator, both in Sections 3.4.2 and 4.2.1. It was assumed that this error could be representative of the actual acceleration resolution. This section will finally look at the actual acceleration resolution of the demodulated acceleration signal itself to reveal how good the previous assumptions actually were.

Fig. 4.10 shows the system blocks that are simulated in this section. Notice that Fig.10 has a few more blocks separating $i(t)$ and $q(t)$ from $\hat{a}(t)$ than the corresponding section of the functional block diagram in Fig. 4.1. Creating the simulation for this section revealed the necessity of including the extra blocks in the design. First, notice the blocks containing $e^{-j\omega N}$. These are ideal delays with the delays set equal to the differentiator delay, so that $N=M/2$ (for even M). The absence of these delay blocks significantly distorts the acceleration output, $\hat{a}(t)$. Second, the $\frac{\pi f_s}{2}$ blocks are necessary to complete the differentiator operation, as discussed in Section 4.2.1. Lastly, for the expected oscillator outputs, the $\beta = \frac{1}{\pi(k_1 + k_2)}$ block was found to be a necessary scale factor for the oscillator signals.

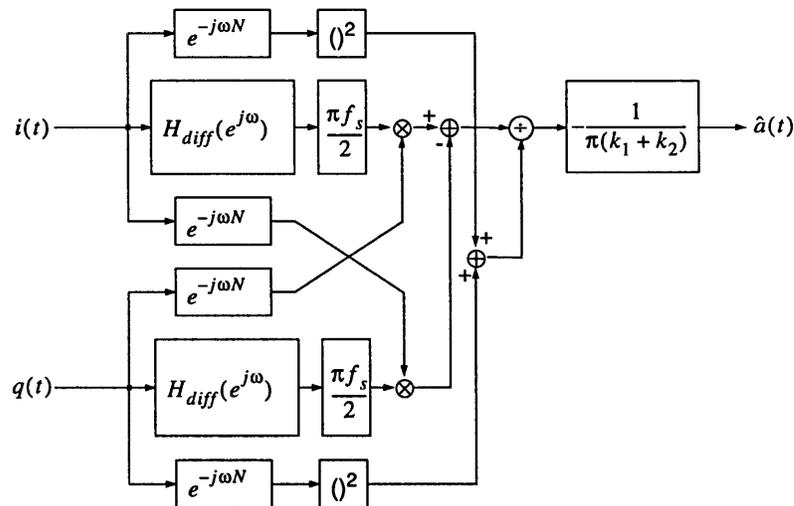


Figure 4.10: Cross-Differentiate Multiply demodulator block diagram.

SIM2(M) (Appendix A.2) implemented Fig. 4.10 with the following $i(t)$ and $q(t)$:

$$i(t) = S_1 S_2 \cos\left(\left(\frac{k_1 A}{f_a}\right) \sin(2\pi f_a t)\right) = S_1 S_2 \cos\left(2\pi k_1 \int_0^t a(t) dt\right) \quad \text{and} \quad (4.23)$$

$$q(t) = S_1 S_2 \sin\left(\left(\frac{k_2 A}{f_a}\right) \sin(2\pi f_a t)\right) = S_1 S_2 \sin\left(2\pi k_2 \int_0^t a(t) dt\right) \quad \text{where}$$

$$a(t) = A \cos(2\pi f_a t). \quad (4.24)$$

The parameters are exactly as in Section 4.2.1: $S_1=S_2=1$, $A=20$ g, $k_1=k_2=50$ Hz/g, $f_a=100$ Hz, and $f_s=10$ kHz. Equation (4.24) is the input acceleration to the entire accelerometer (see Fig. 4.2) and so ideally, for this subsystem, $\hat{a}(t)$ should be $a(t)$ delayed by the differentiator delay:

$$\hat{a}_{ideal}(t) = A \cos\left(2\pi f_a \left(t - \frac{M}{2f_s}\right)\right) \quad (4.25)$$

where f_s is the sampling frequency and $M/2$ is the differentiator delay. This ideal output is plotted in Fig. 4.11 along with its spectrum.

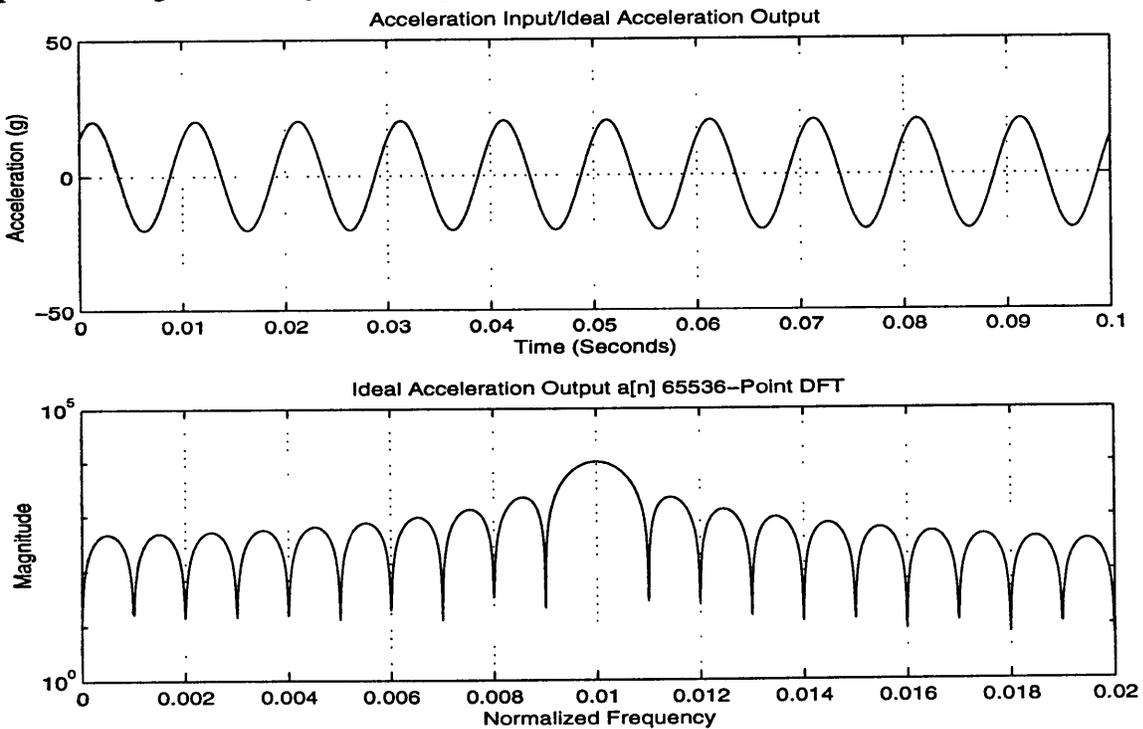


Figure 4.11: Ideal acceleration output for (4.23) and (4.24).

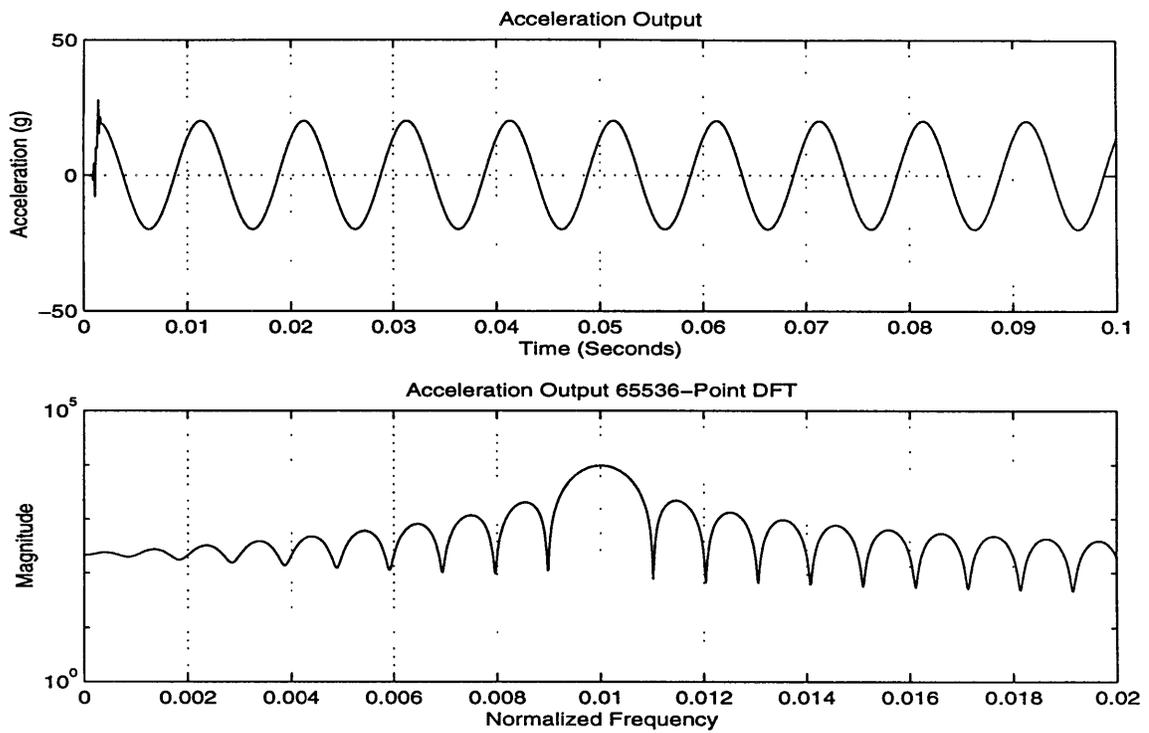


Figure 4.12: Actual acceleration output for (4.23) and (4.24).

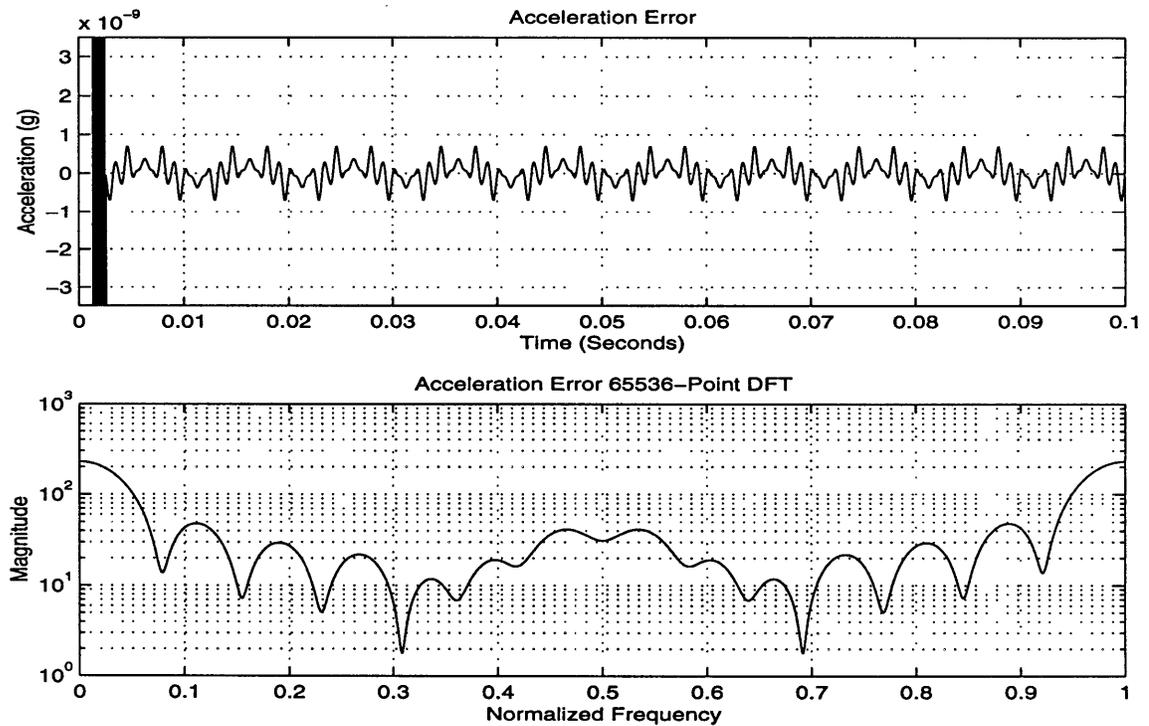


Figure 4.13: Acceleration output error.

Running SIM2(26), the demodulator output, $\hat{a}(t)$, is shown in Fig. 4.12. The acceleration error is displayed in Fig. 4.13, and the maximum error, excluding the start-up transient, is 7×10^{-10} g, or $0.0007 \mu\text{g}$! The performance is three orders of magnitude better than the originally intended $1 \mu\text{g}$ resolution. The pleasant surprise comes from, in fact, the scaling of the error by $\frac{1}{\pi(k_1 + k_2)}$. Already, it is evident that perhaps the $\frac{\pi f_s}{2}$ and $\frac{1}{\pi(k_1 + k_2)}$ blocks can be combined, and would be required in a practical implementation to prevent overflow using fixed arithmetic.

The simulations for $M=22$, $M=26$, and $M=28$ are summarized in Table 4.2

Table 4.2: SIM2(M) results for $M=22$, $M=26$, and $M=28$.

M	Acc. Err. (g)
22	3.00×10^{-8}
26	6.98×10^{-10}
28	8.35×10^{-11}

Apparently, the differentiators were designed with higher order than necessary and the error estimate (4.7) would do better by multiplying $|e_{\text{diff}}[n]|_{\text{max}}$ by $\frac{1}{\pi(k_1 + k_2)}$.

These exceedingly accurate differentiators will continue to be used in the simulations, however, for at some point, degradation will occur as more and more blocks in Fig. 4.2 are included in the simulation.

4.2.3 FM Bandwidth Requirements

Section 3.3.1 estimated the bandwidth of the oscillator FM signals by using Carson's Rule (3.18), which approximated the width of the frequency band that contained 99% of the signal power. Bandwidth was critical in Section 3.3 because the minimum clock speed and velocity resolution of the phase readouts depended on the maximum frequency of the FM signal (3.12) and (3.13). The theoretical analyses of Chapter 3, however, did not con-

sider, at any point, the repercussions of bandlimiting the FM signal, such as degradation of the signal and decreased resolution. These effects must be considered in a more detailed system analysis because a practical system includes low pass filters necessary to eliminate high frequency noise. Even the differentiator error discussed in Section 4.2.1 was a function of FM signal bandwidth. If the passband cutoff frequencies of the QCDM demodulator are set according to the Carson Rule, then 1% of the FM signal power would be discarded. For a high resolution system requiring an SNR of 146 dB (Chapter 2), the Carson Limit seems an inadequate estimate of FM bandwidth necessary to achieve the desired resolution. The objective of this section, then, is to determine the minimum bandwidth of an FM signal that can achieve a desired level of demodulator output resolution. The results will be used in the next section to design the low pass filters of the system.

Obtaining a closed form analytical expression relating FM bandwidth and demodulator resolution would be difficult, and Chapter 3's exclusion of such an expression in its first order theoretical analyses was not unreasonable. For perhaps the most straightforward approach to estimating this FM bandwidth-resolution (FMBR) relationship is through simulation, which is performed for a specific FM demodulation method. Therefore, only after obtaining adequate knowledge of a particular demodulator can the FMBR relationship be estimated. How well the estimate holds for other FM demodulators is unclear. The results of this analysis, then, pertain specifically to the CDM readout in question, and can be used merely as a benchmark for other systems.

The results of Sections 4.2.1 and 4.2.2 provide the necessary background to approximate the FMBR relationship for the CDM demodulator. Again, the analysis in this section is of the block diagram in Fig. 4.10. In the previous section, SIM2(M) used inputs (4.23) and (4.24) and compared the output, $\hat{a}(t)$, to the ideal output in (4.25). In this section, SIM3(M,Ni) (Appendix A.3) uses, instead, ideally bandlimited inputs, then compares the

output $\hat{a}(t)$ once again, to the ideal output in (4.25). The variable M is the differentiator order, and N_i is the number of sidebands contained in the input, as will be discussed in the next paragraph. $SIM2(M)$ simulates how the differentiator limits the acceleration readout resolution, while $SIM3(M,N_i)$ simulates how both the differentiator and FM signal bandwidth together restrict the acceleration resolution. By comparing the two simulation results, an estimate of the FMBR relationship can be obtained. Presumably, in starting with $N_i=1$ and increasing it for successive simulations, eventually the acceleration resolution will converge to the values in Table 4.2, at which point the differentiator is the dominant limiter of resolution.

An angle modulated signal can be approximated by a series of modified Bessel functions of the first kind as follows [3][4]:

$$A_1 \cos(B_1 \sin X_1) = \lim_{N_i \rightarrow \infty} \left(A_1 J_0(B_1) + A_1 \sum_{k=1}^{N_i} 2J_{2k}(B_1) \cos(2kX_1) \right) \quad (4.26)$$

$$A_2 \sin(B_2 \sin X_2) = \lim_{N_i \rightarrow \infty} \left(A_2 \sum_{k=0}^{N_i} 2J_{2k+1}(B_2) \sin((2k+1)X_2) \right) \quad (4.27)$$

$SIM3(M,N_i)$ uses the above descriptions to generate the ideally bandlimited FM inputs by defining the parameters A_1 , B_1 , X_1 , A_2 , B_2 , and X_2 , as follows:

$$i(t) = A_1 J_0(B_1) + A_1 \sum_{k=1}^{N_i} 2J_{2k}(B_1) \cos(2kX_1) \quad \text{where} \quad (4.28)$$

$$A_1 = S_1 S_2 = 1$$

$$B_1 = \frac{k_1 A}{f_a}$$

$$X_1 = 2\pi f_a t$$

and

$$q(t) = A_2 \sum_{k=0}^{Ni} 2J_{2k+1}(B_2) \sin((2k+1)X_2) \quad \text{where}$$

$$A_2 = S_1 S_2 \tag{4.29}$$

$$B_2 = \frac{k_2 A}{f_a}$$

$$X_2 = 2\pi f_a t$$

Comparing equations 4.28 and 4.29 to equations 4.23 and 4.24, it is clear that as N_i gets large, the ideally bandlimited inputs approach the ideal FM inputs used in SIM2(M). Notice in (4.28) and (4.29) that each sideband increases the baseband Nyquist frequency of the FM signal by $2f_a$ Hz. Therefore, the maximum frequency generated in the ideally bandlimited signal is approximately

$$f_N = 2 \cdot N_i \cdot f_a. \tag{4.30}$$

For $N_i=6$, $f_N=1200$ Hz, which is approximately the Carson Limit for $i(t)$ and $q(t)$. Executing SIM3(26,6), the Bessel-function generated FM signal is shown in Fig. 4.14a. Note that the spectrum in Fig. 4.14b. is similar to the spectrum of the ideal FM input shown in Fig. 4.6b, but has only six sidebands.

The resulting acceleration error at the output of the demodulator is show in Fig. 4.15b, with a maximum error, after the start-up transient, of 0.5996 g. As expected, the Carson Limit bandwidth fails to achieve the desired resolution of 1 μ g. In fact, it does not even come close!

Running the simulation for $N_i=12$, or $f_N=2400$ Hz, the generated FM signal is shown in Fig. 4.16a. The spectrum in Fig. 4.16b. should have six more sidebands than in Fig. 4.14b, but the upper sidebands are too small to be seen (non-rectangular windowing would help). Their effect, however, is apparent as the acceleration error is shown to be much larger in Fig. 4.17a. The maximum error, after the start-up transient, is 0.14 μ g, which achieves the desired resolution. These simulations suggest that to achieve 1 μ g resolution,

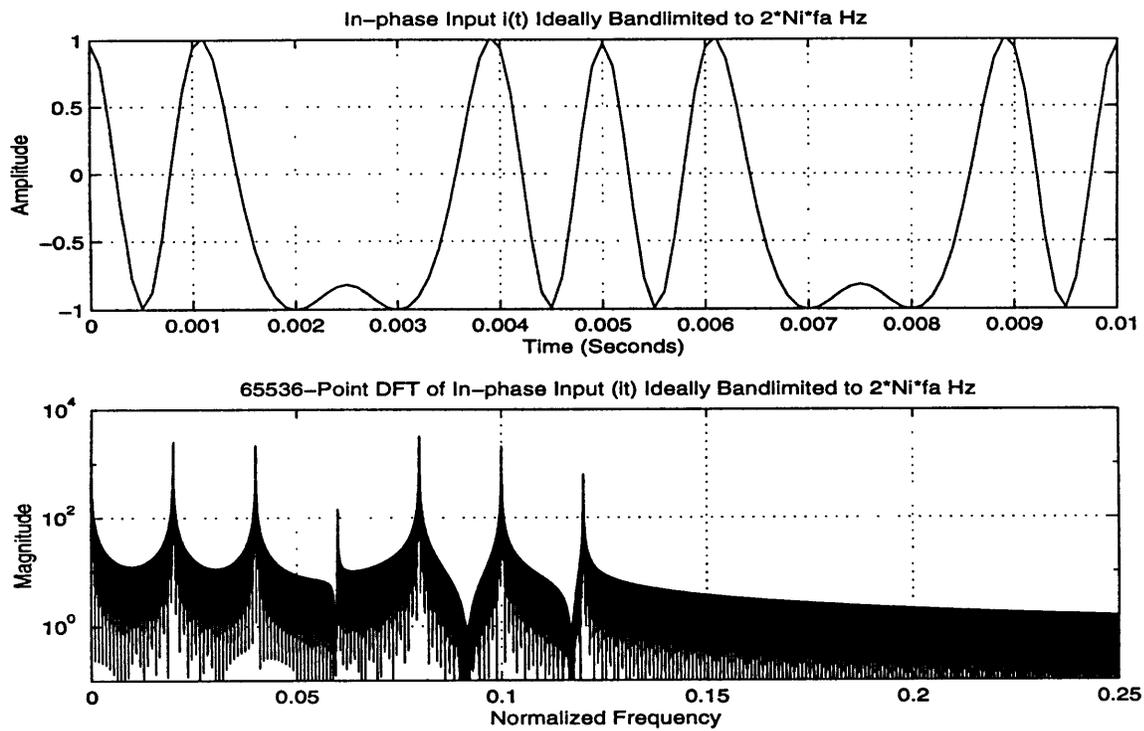


Figure 4.14: Input ideally-bandlimited to the Carson Limit. $N_i=6$, $f_a=100$.

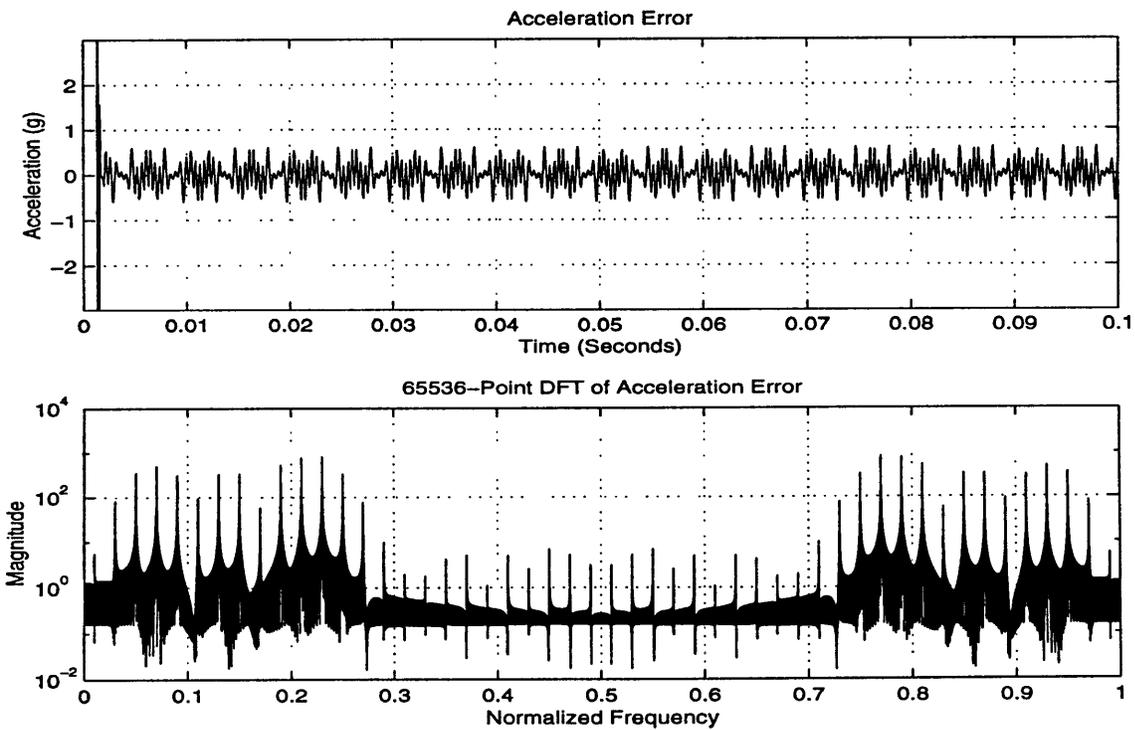


Figure 4.15: Acceleration error for Carson Limit bandwidth.

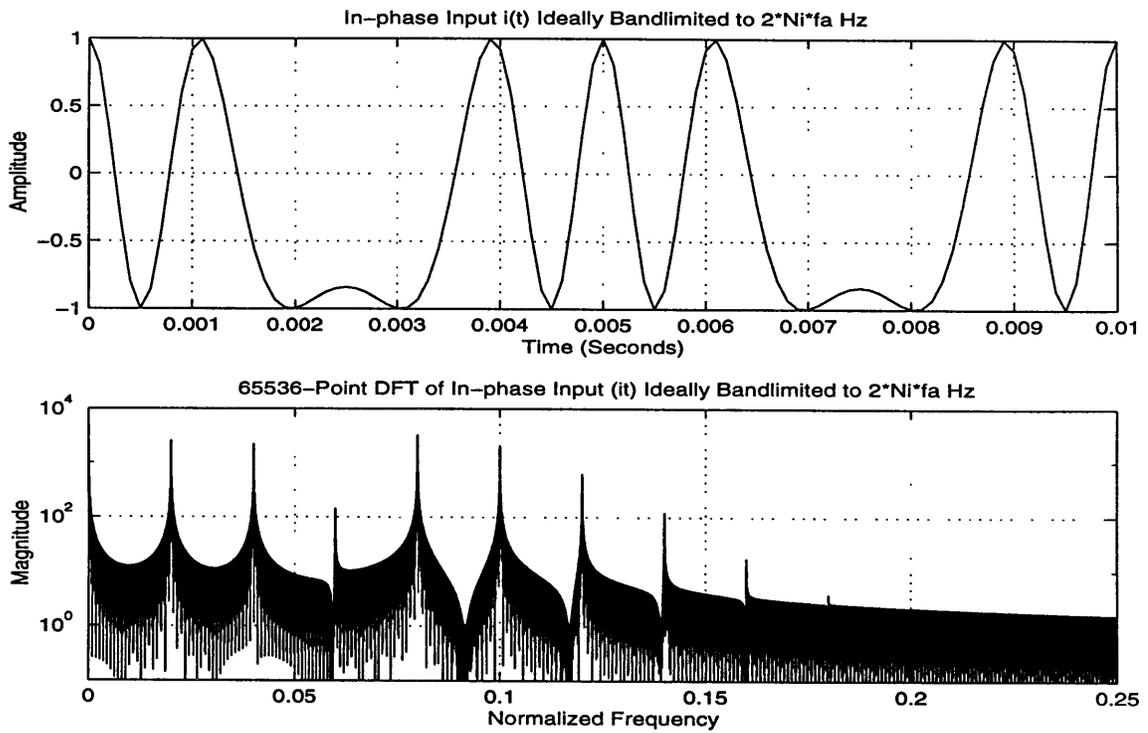


Figure 4.16: Input ideally bandlimited to 2400 Hz. $N_i=12$, $f_a=100$.

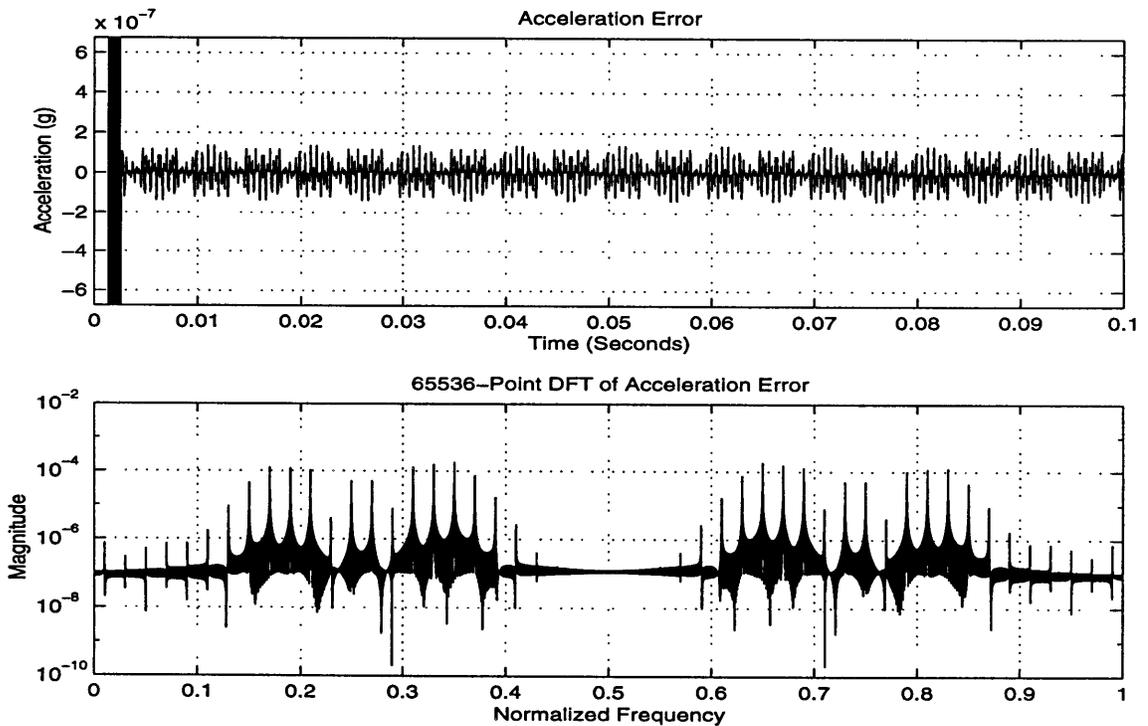


Figure 4.17: Acceleration error for bandwidth of 2400 Hz.

the Nyquist frequency needs to be at least twice the Carson Limit at baseband.

SIM4(M,NSB) (Appendix A.4) is an extension of SIM3(M,Ni). For a differentiator of order M, as designed in SIM1(M), this simulation computes the acceleration error for FM bandwidths

$$f_N = 2 \cdot Ni \cdot f_a \quad 1 \leq Ni \leq NSB, \quad (4.31)$$

thereby giving data from which to determine the FMBR relationship. SIM4(M,NSB) was executed for (M=22,NSB=18), (M=26,NSB=18), and (M=28, NSB=18), and the results

Table 4.3: SIM4(M,NSB) for M=22, 26, 28, and NSB=18.

Ni	f_N (Hz)	Acc. Err. (g) for M=22	Acc. Err. (g) for M=26	Acc. Err. (g) for M=28
1	200	16.8	16.8	16.8
2	400	13.4	13.4	13.4
3	600	101	101	101
4	800	9.63	9.63	9.63
5	1000	2.40	2.40	2.40
6	1200	6.00×10^{-1}	6.00×10^{-1}	6.00×10^{-1}
7	1400	8.90×10^{-2}	8.90×10^{-2}	8.90×10^{-2}
8	1600	1.00×10^{-2}	1.00×10^{-2}	1.00×10^{-2}
9	1800	8.06×10^{-4}	8.06×10^{-4}	8.06×10^{-4}
10	2000	5.83×10^{-5}	5.83×10^{-5}	5.83×10^{-5}
11	2200	3.06×10^{-6}	3.06×10^{-6}	3.06×10^{-6}
12	2400	1.30×10^{-7}	1.35×10^{-7}	1.35×10^{-7}
13	2600	3.09×10^{-8}	5.86×10^{-9}	5.51×10^{-9}
14	2800	3.00×10^{-8}	7.84×10^{-10}	2.02×10^{-10}
15	3000	3.00×10^{-8}	6.98×10^{-10}	8.45×10^{-11}
16	3200	3.00×10^{-8}	6.97×10^{-10}	8.30×10^{-11}
17	3400	3.00×10^{-8}	6.97×10^{-10}	8.31×10^{-11}
18	3600	3.00×10^{-8}	6.97×10^{-10}	8.31×10^{-11}

are displayed in Table 4.3. Note that as the bandwidth increases, the resolution for each M converges to the differentiator limited resolutions listed in Table 4.2, as expected. Therefore, prior to convergence, the resolution is limited by the FM bandwidth. After convergence, the resolution is limited by the differentiator error.

Fig. 4.18 shows a plot of the FMBR relationship for the CDM. Again, as the FM bandwidth increases, the resolution improves until leveling off at the resolution limits determined by the differentiator.

SIM4(M,NSB), therefore, is a useful design tool that relates both FM bandwidth and differentiator order to the acceleration resolution for the CDM demodulator and the expected worst-case parameters of the VIA. SIM4(M,NSB) computes the best possible theoretical performance of the CDM based on the two parameters. As more and more blocks are included in the simulation, still more parameters will be taken into consideration.

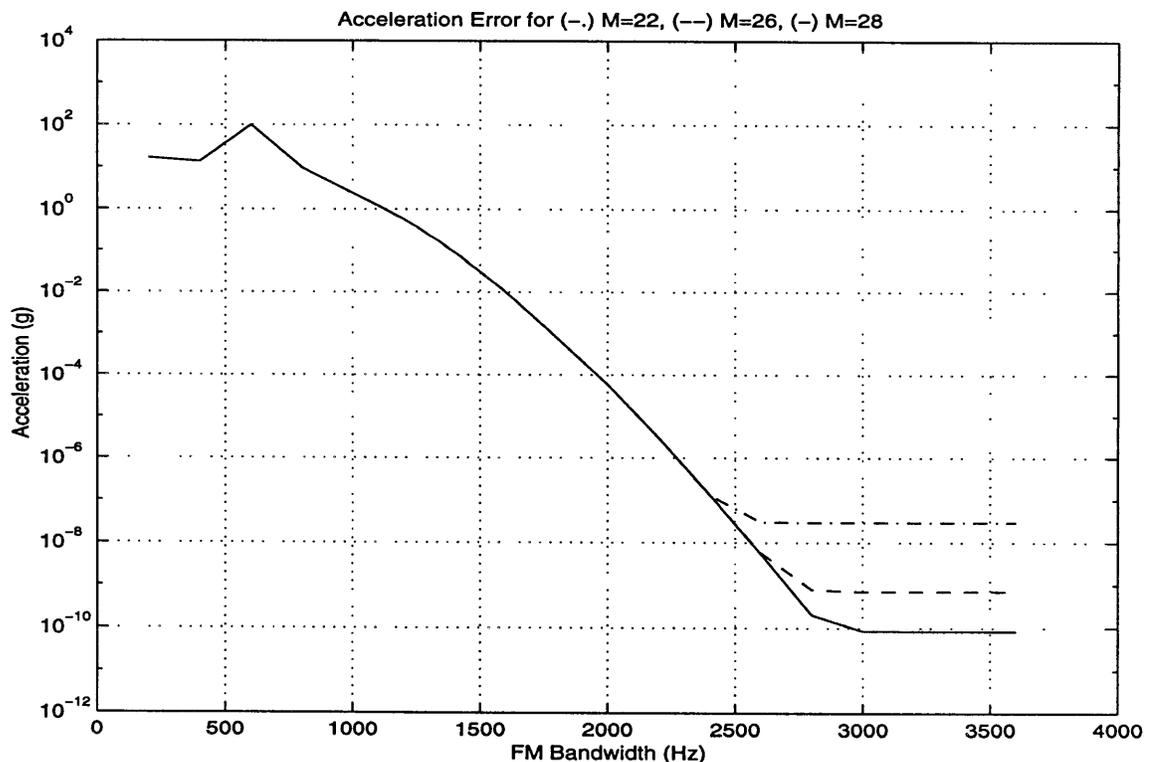


Figure 4.18: FMBR relationship for the CDM demodulator.

ripple amplitude, D_p and D_s . First, to ensure that the desired resolution can be achieved, the F_p must be at least $2 \cdot N_i \cdot f_a$ for the resolution achieved in the previous section. Second, the ripple magnitudes need to be at least as small as the desired resolution, so that they do not distort the FM signal by more than the intended resolution. Lastly, the F_s will primarily be chosen to balance the trade-off between minimizing the filter length and maximizing the high frequency noise attenuation. The lower F_s is, the higher the order of the filter, but the better the noise rejection. The higher F_s is, the lower the order of the filter, but the out-of-band noise attenuation is worse.

`SIM5(F_p, F_s, D_p, D_s)` designs an FIR filter that minimizes the filter order while meeting the specifications. There are limitations, however, and the Remez Algorithm fails to converge for certain specifications. The function then filters the ideal input FM signal, $i(t)$, defined in (4.23), at a 10 kHz sampling rate, to observe the distortion introduced by the filter.

Executing `[hlpf, N]=SIM5(2.4e3, 4.5e3, 1e-8, 1e-8)`, the function designs an order-45 filter with magnitude response shown in Fig. 4.20a. and passband and stopband ripples given in Fig. 4.20b. and c. The ripples are less than 1×10^{-8} , as specified for D_p and D_s , and the passband is sized correctly. The distortion introduced in filtering $i(t)$ with $H_{lpf}(e^{j\omega})$ is shown in Fig. 4.21. The error is less than $D_p=D_s=1 \times 10^{-8}$ as expected.

`SIM6(M, F_p, F_s, D_p, D_s)` does as much and more than `SIM3(M, N_i)`. By setting $F_p = 2 \cdot N_i \cdot f_a$, the bandwidth of the FM input signal can be controlled. Specifying M sets the differentiator order, as before. The three new parameters, F_s , D_p , and D_s , now give greater freedom in designing the filters, and can be adjusted in future simulations. The following three test were done using `SIM6`:

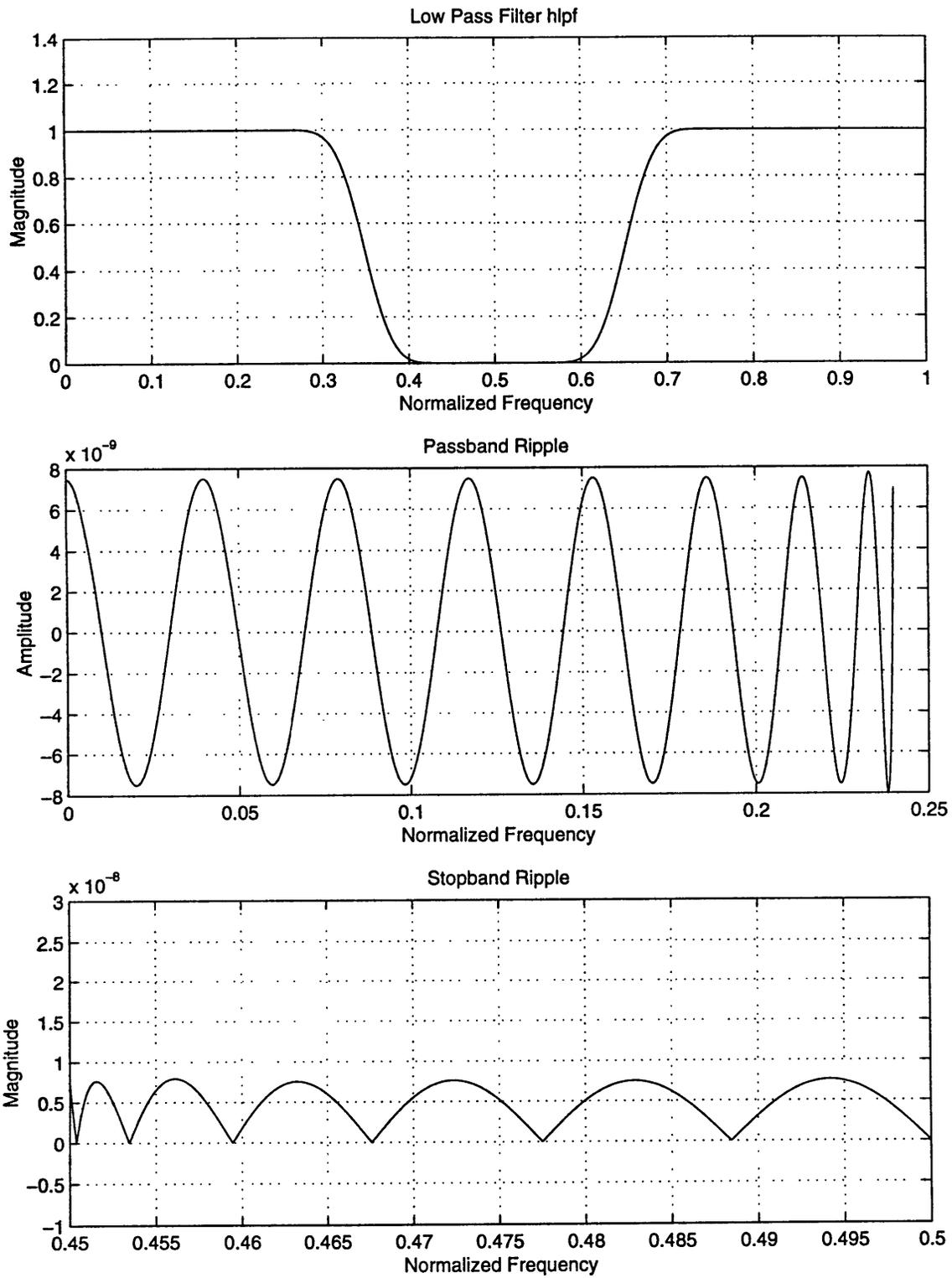


Figure 4.20: Order-45 low pass filter.

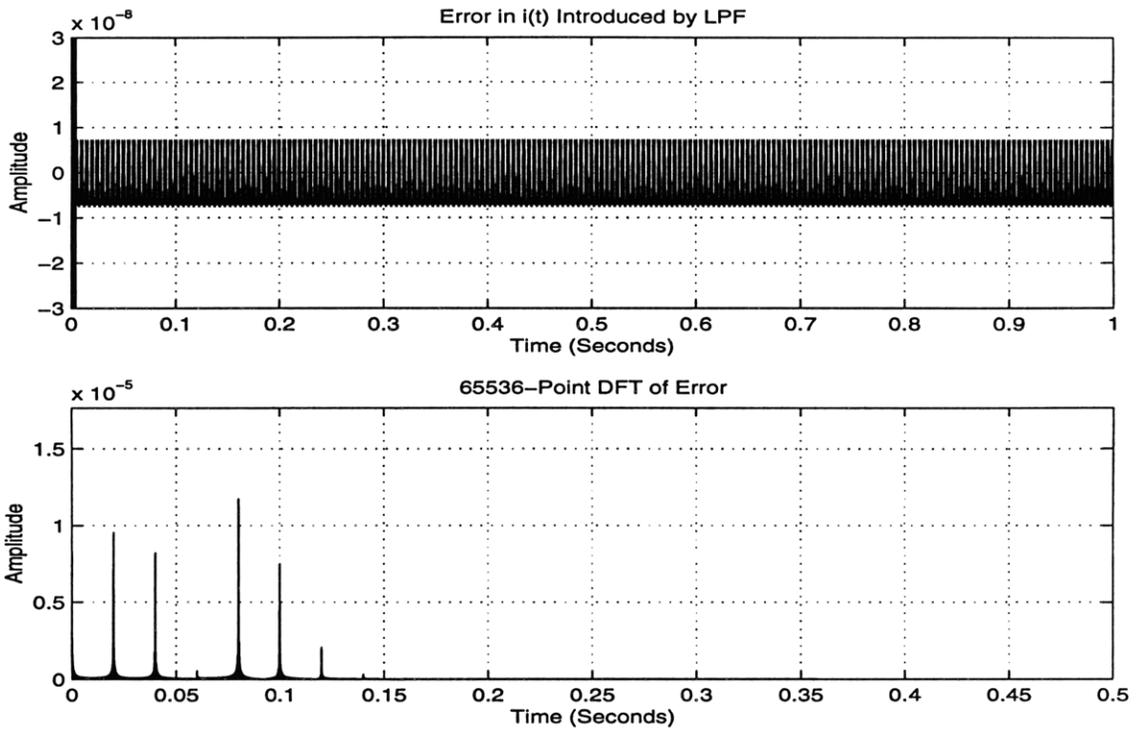


Figure 4.21: Distortion of $i(t)$ due to low pass filter.

1. [hlpf1,N1]=SIM6(22,2.4e3,4.5e3,1e-7,1e-7);
2. [hlpf2,N2]=SIM6(26,2.4e3,4.5e3,1e-8,1e-8);
3. [hlpf3,N3]=SIM6(28,2.4e3,4.5e3,1e-9,1e-9);

While F_p and F_s are kept the same in each test, D_p and D_s are set to be one hundredth of the desired resolutions $10 \mu g$, $1 \mu g$, $0.1 \mu g$. The acceleration error for $M=26$ is shown in Fig. 4.22, and the maximum errors, excluding the start-up transients, for the other test are included in Table 4.4. As D_p and D_s decrease, the order of the designed low pass filter increase, as expected.

Table 4.4: SIM6(M,F_p,F_s,D_p,D_s) for tests 1., 2., and 3.

M	Filter Order	Acc. Err. (g)
22	39	8.47×10^{-6}
26	45	1.39×10^{-7}
28	50	3.86×10^{-8}

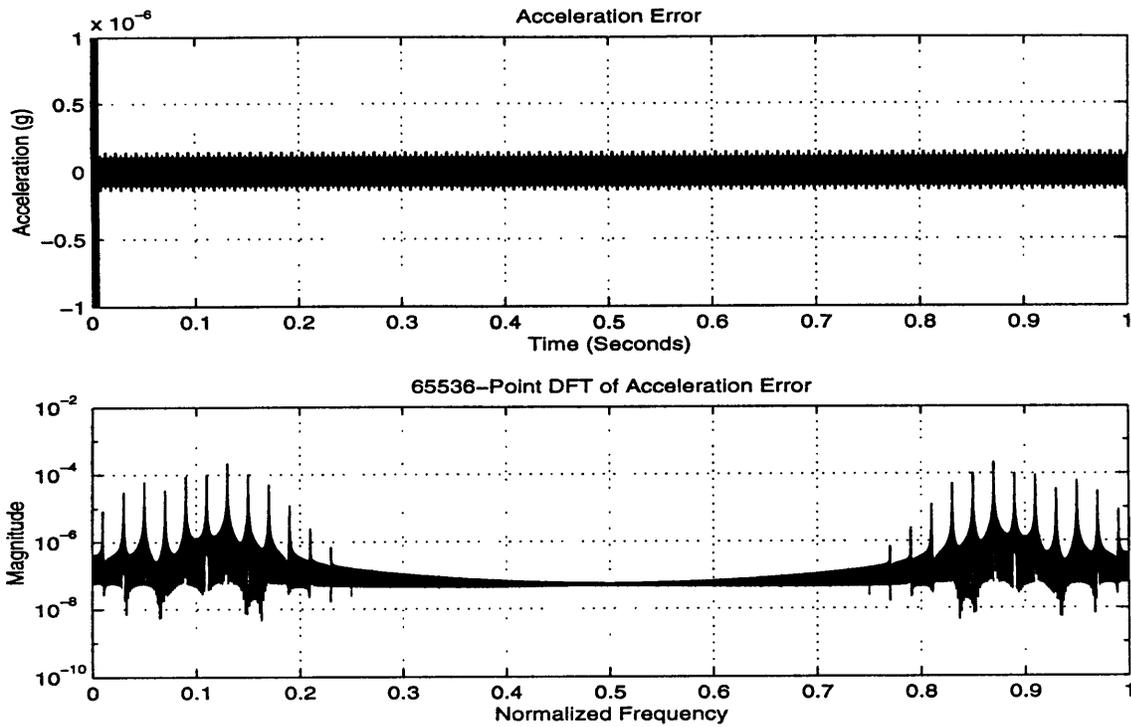


Figure 4.22: Acceleration error for $M=26$, and order-45 low pass filter.

The narrowband filter is designed in a similar way, using $\text{SIM7}(Fp2, Fs2, Dp2, Ds2)$. Since the maximum acceleration frequency is expected to be 100 Hz, $Fp2$ is set to be 200 Hz. The passband and stopband ripple are set at $Dp2=Ds2=1e-7$, and the stopband cutoff is set to $Fs2=1.5e3$. As in the design of the LPF, decreasing $Fs2$ will increase the order of the filter while increasing $Fs2$ will lower the length of the optimal filter. For these values, SIM7 designs the narrowband filter shown in Fig. 4.23.

For a test acceleration signal, where

$$a(t) = 20\cos(2\pi 100t), \quad (4.32)$$

the distortion introduced by the filter is shown in Fig. 4.24. The primary purpose of this filter is to eliminate noise outside of the band dictated by the maximum acceleration signal. The effects of this filter will be seen more clearly in Section 4.3.

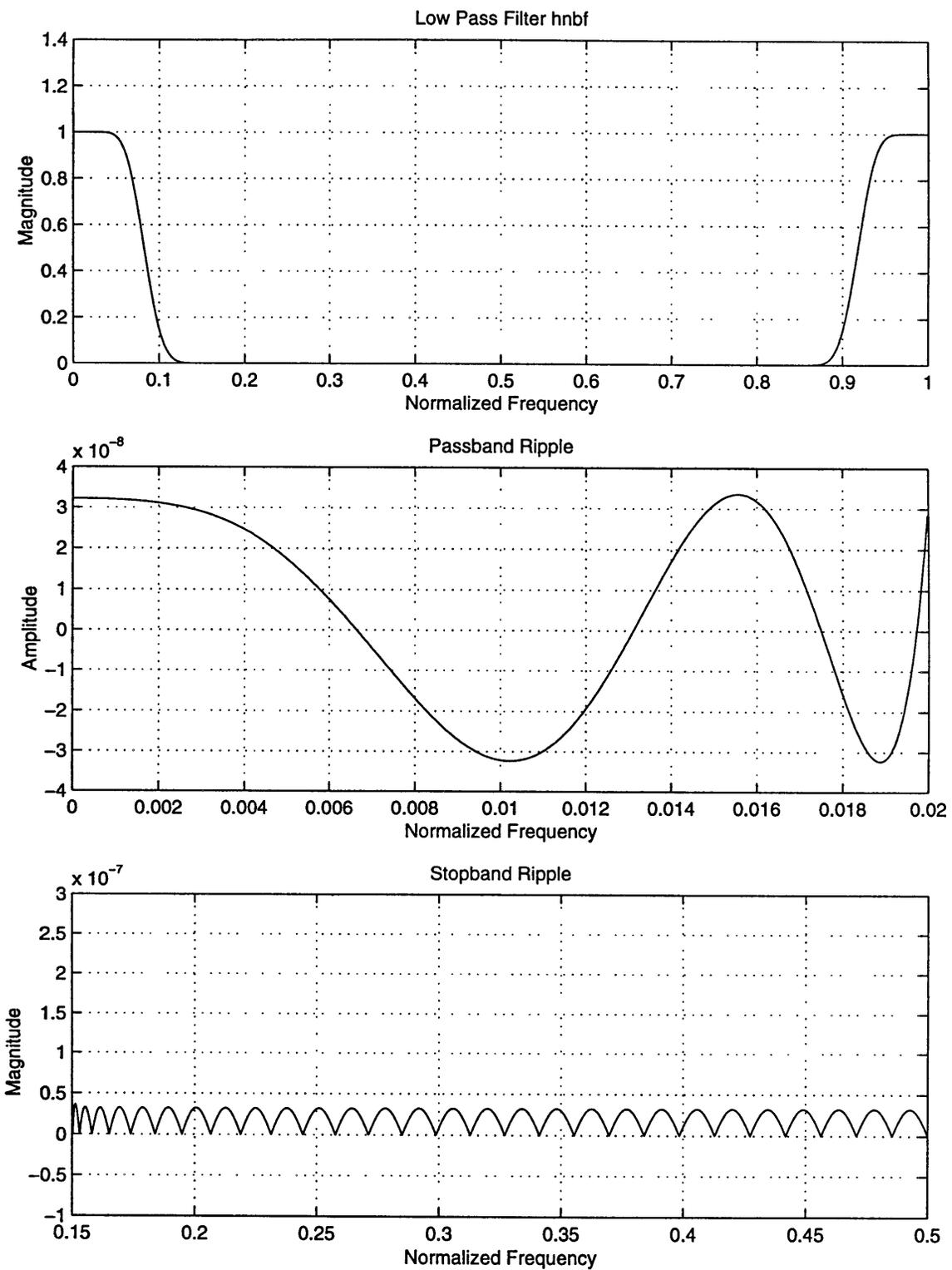


Figure 4.23: Narrowband filter hbnf of order 65.

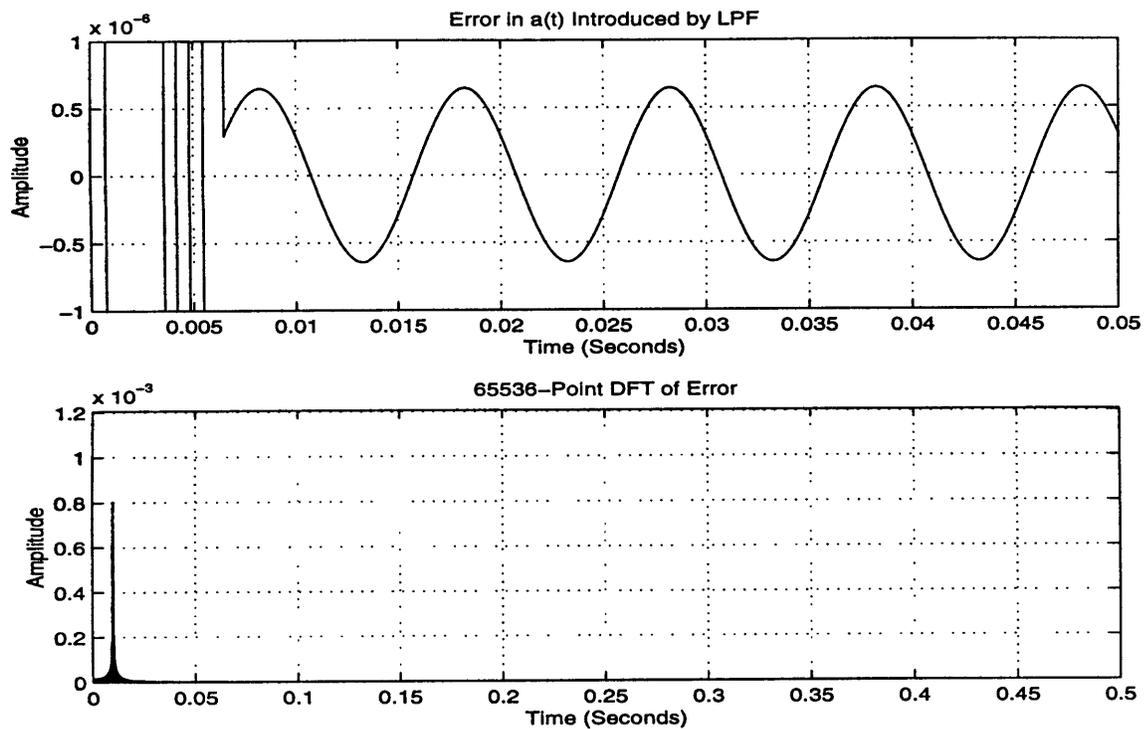


Figure 4.24: Distortion of acceleration signal due to narrowband filter.

SIM8(M,Fp,Fs,Dp,Ds,Fp2,Fs2,Dp2,Ds2) cascades the narrowband filter to the back end of the CDM demodulator. For the input signals defined in (4.23) and (4.24), SIM8(26,2.4e3,4.5e3,1e-8,1e-8,2e2,1.5e3,1e-7,1e-7) produces the results in Figures 4.25 and 4.26. Comparing Fig. 4.25b. to Fig. 4.22b., note that the output acceleration signal is degraded slightly by the narrowband filter, but the error spectrum has been further band-limited. This will be an important means of rejecting high frequency error, as previously stated. Fig. 4.26 shows the acceleration output after narrowband filtering

4.2.5 Numerical Integration

Section 3.1.1 discussed in detail the error introduced by digital integration, and showed that to obtain high resolution the sampling frequency needs to be high for Riemann Sum integration. This section reveals how the integration resolution can be

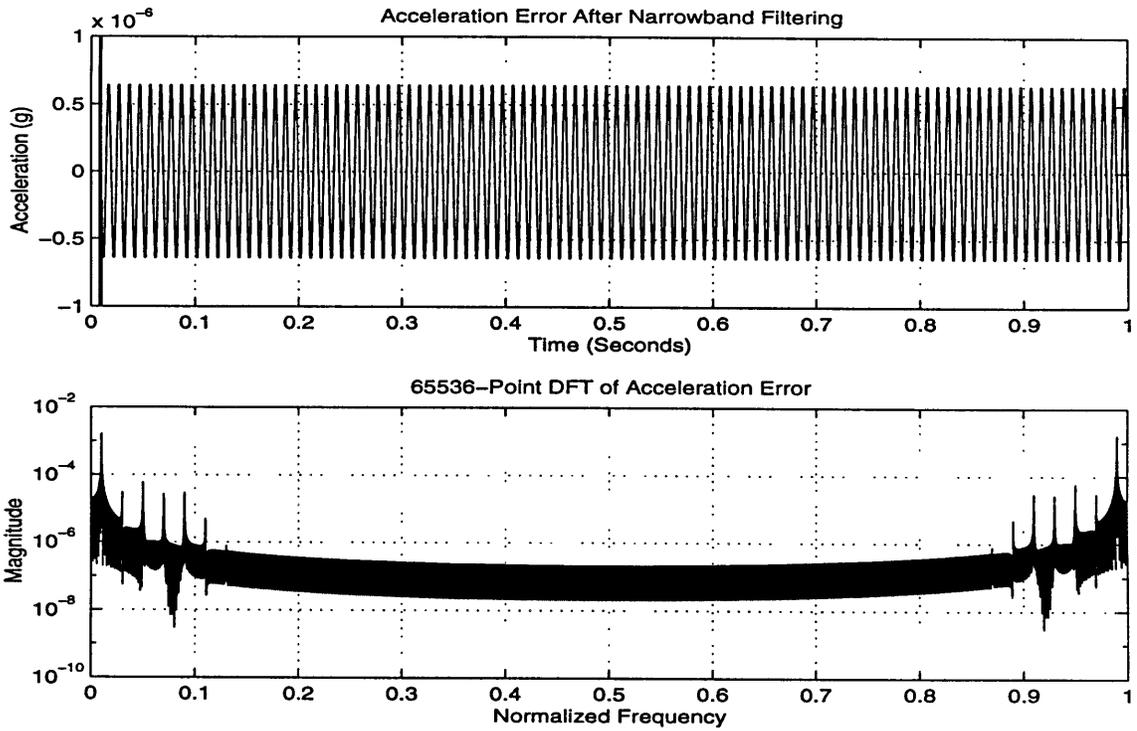


Figure 4.25: Acceleration error after narrowband-filtering stage.

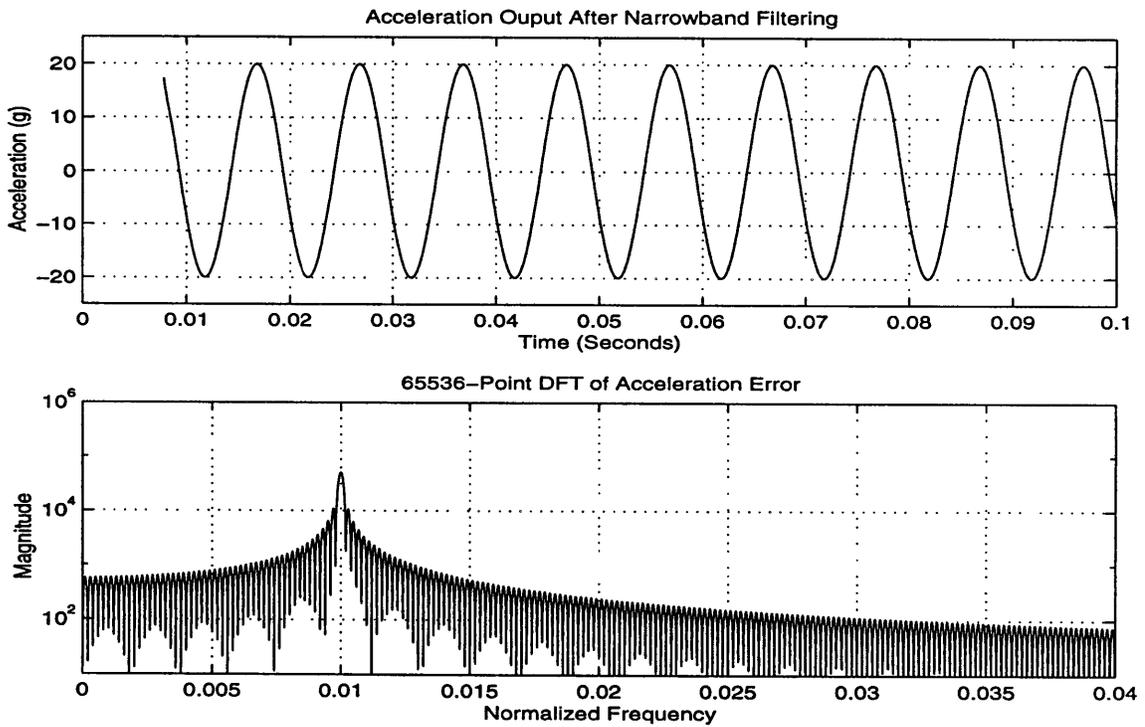


Figure 4.26: Acceleration output after narrowband-filtering stage.

improved by decreasing the sampling rate rather than increasing it. It will show that polyphase decomposition might be a method of achieving arbitrarily accurate integration without relying solely on a high sampling frequency. The most fascinating aspect of this section, however, is the description of three analogies that intimately connect numerical methods to digital filter structures.

Numerical methods are often employed when certain expressions or experimental data cannot be easily described with closed form analytical functions. In such cases, rather than abandoning the data because processing it cannot yield exact solutions and descriptions, more often than not, numerical methods are used to come up with answers, where none would exist otherwise. The answers, however, have varying degrees of accuracy depending on the numerical method employed. The errors are smaller for more computationally intensive strategies, and the trade-off that must be made is one of computational efficiency versus accuracy.

In Section 4.2.1, the design of discrete-time differentiators was treated. High accuracy was dependent on the sampling frequency, the Nyquist frequency of the input, and the filter length. Designing highly accurate digital integrators, however, does not seem to be as straightforward, and is less commonly treated in introductory DSP texts. An ideal integrator is an IIR filter, whereas the digital filters designed for the CDM are FIR, and therefore optimal filter design with the Remez Algorithm is not possible. Since the integrator instills a significant limitation on frequency readout schemes, a highly accurate scheme is necessary. Turning to calculus and numerical methods sheds much insight onto the design of an accurate digital integrator.

In two dimensions, integration can be viewed, in a graphical perspective, as the determination of the area under a continuous time curve. All numerical methods inherently involve discretization, and integration is no different. Numerical integration, or discrete-

time integration in the time domain, sums up small, elemental, discrete areas to approximate the larger area under the curve. The three numerical methods described here vary in the shape of the discrete elements used in the summation. The Riemann Sum, as seen in Section 3.1.1, uses rectangles to approximate the area of the elements, the Trapezoidal Rule uses trapezoids, and Simpson's Rule constructs parabolas, with each successive method outperforming its predecessor.

The definite integral of a function, $f(x)$, between points $x=a$ and $x=b$ of the x -axis, can be expressed in the following three ways [18]:

$$\text{Riemann Sum: } \int_a^b f(x) dx \approx \Delta x \sum_{i=0}^n f_i, f_0=f(a), f_n=f(b). \quad (4.33)$$

$$\text{Trapezoidal Rule: } \int_a^b f(x) dx \approx \frac{\Delta x}{2} \left(f_0 + f_n + 2 \sum_{i=1}^{n-1} f_i \right), f_0=f(a), f_n=f(b). \quad (4.34)$$

$$\text{Simpson's Rule: } \int_a^b f(x) dx \approx \frac{\Delta x}{3} \left(f_0 + f_n + 4 \sum_{i=1}^{\frac{n}{2}} f_{2i-1} + 2 \sum_{j=1}^{\frac{n}{2}-1} f_{2j} \right), f_0=f(a), f_n=f(b), n \text{ is even.} \quad (4.35)$$

The derivation of each of these approximations stems from the Taylor Series expansion for $I(x + \Delta x)$ and $I(x - \Delta x)$, where $I(x) = \int_a^b f(x) dx$. The derivatives of $f(x)$ in the Taylor Series are replaced by finite difference approximations using $f(x)$. In short, Simpson's Rule outperforms the other two methods because it uses more points in the approximation of higher order derivatives of $f(x)$, that is, it uses more knowledge of $f(x)$ to calculate each elemental area. Notice, too, that Simpson's Rule works only for even n . This can be viewed as downsampling the x -axis by two, or using knowledge of $f(x)$ at two points to calculate each point of the integral.

To develop the analogous digital filter implementations of (4.33), (4.34) and (4.35), first define a digital system, shown in Fig. 4.27, where the output, $\hat{v}[n]$, is an

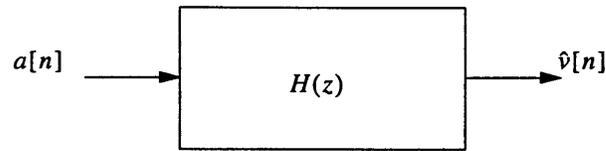
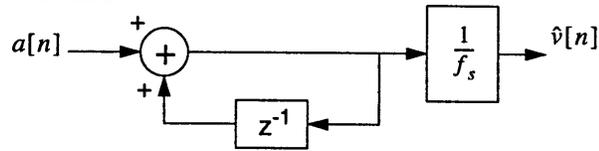


Figure 4.27: Integrator block

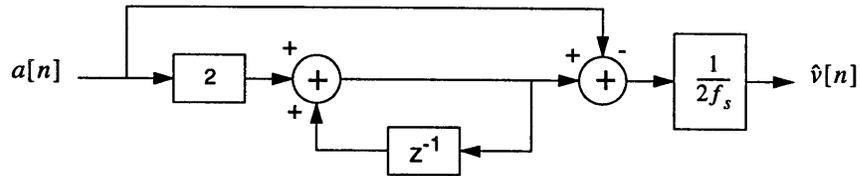
approximation of the integral of the input $a[n]$ sampled at frequency f_s . The input is analogous to the samples of $f(x)$, such that $a[n] = f_i$ for $a \leq i \leq b$ where i is an integer. The sampling rate is inversely proportional to the stepsize, such that $\frac{1}{f_s} = \Delta x$. The system output is analogous to the series integration approximation, while the digital system is analogous to a chosen numerical method. The initial conditions are considered to be zero, with $a[0] = f_0 = 0$. The analogous digital systems are then implemented in block-diagram form by inspection of (4.33), (4.34), and (4.35). The block diagrams are then simplified, combining blocks into transfer functions. The digital filter implementations of the numerical integration methods are shown in Fig. 4.28 and the simplified diagrams are shown in Fig. 4.29.

The most fascinating aspect of this development, lies in the interpretation of the resulting digital filters in Fig. 4.29. The Riemann Sum is implemented with a standard accumulator, the method of digital integration analyzed in Section 3.1.1 The Trapezoidal Rule is implemented with the bilinear transform, which maps the imaginary axis of the Laplace Transform domain to the unit circle of the z-transform domain [5]. In the Laplace domain, integration is equivalent to multiplying by $\frac{1}{s}$. Using the bilinear transform to map s to z , $\frac{1}{s}$ maps to the filter in Fig. 4.29b. The most interesting and least intuitive of the three analogies is the implementation of Simpson's Rule. Fig. 4.29(c) is a polyphase implementation of a decimator filter, that uses the bilinear transform as one of its filters and an accumulator as the other! The polyphase structure is used often in filter banks, which are multirate digital systems that process many different bands of a signal spectrum separately in

(a) Riemann Sum Implementation



(b) Trapezoidal Rule Implementation



(c) Simpson's Rule Implementation

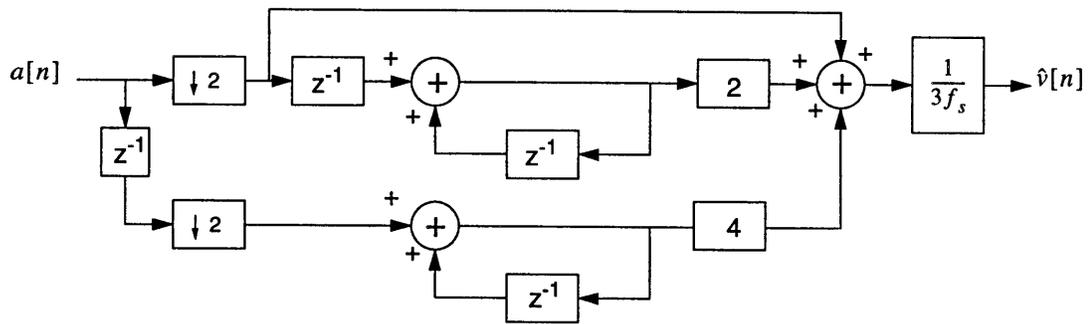
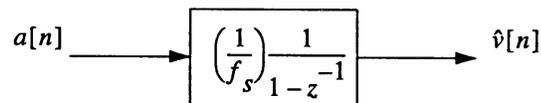
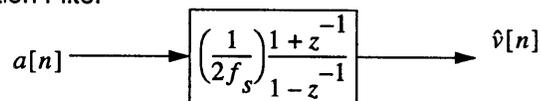


Figure 4.28: Digital filter implementation of numerical integration methods.

(a) Accumulator



(b) Bilinear Transformation Filter



(c) Polyphase Decimator

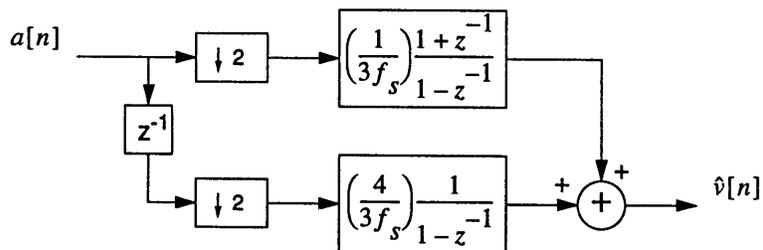


Figure 4.29: Simplified block diagrams.

parallel channels. Multirate digital systems have received much attention lately in the fields of communications, speech processing, and image compression.

SIM9(Fs,A,fa,input) (Appendix A.9) implements each of the three integration filters and outputs the velocity error for the following input, which is specified by 'cos':

$$a(t) = A \cos(2\pi f_a t). \tag{4.36}$$

The simulation was run a number of times for sampling frequency of 10 kHz and various values of A and f_a . The results are plotted in Figures 4.30, 4.31, and 4.32, with the decimator performing the best out of the three.

For the VIA readout, the DSP sampling frequency is limited on the front end by the necessary FM bandwidth, as seen in Section 4.2.3. After demodulation, however, the velocity signal can be output at approximately 500 Hz, depending on the inertial navigation in use. Therefore, the acceleration signal can be downsampled in the integration to the

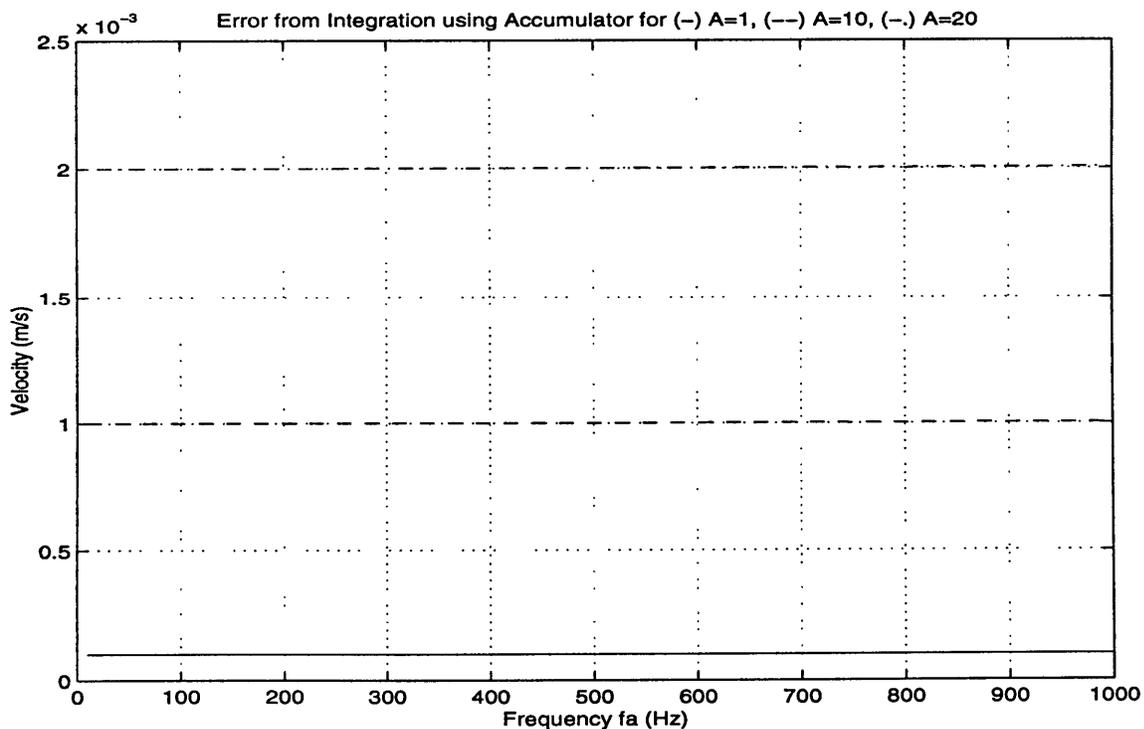


Figure 4.30: Maximum accumulator error.

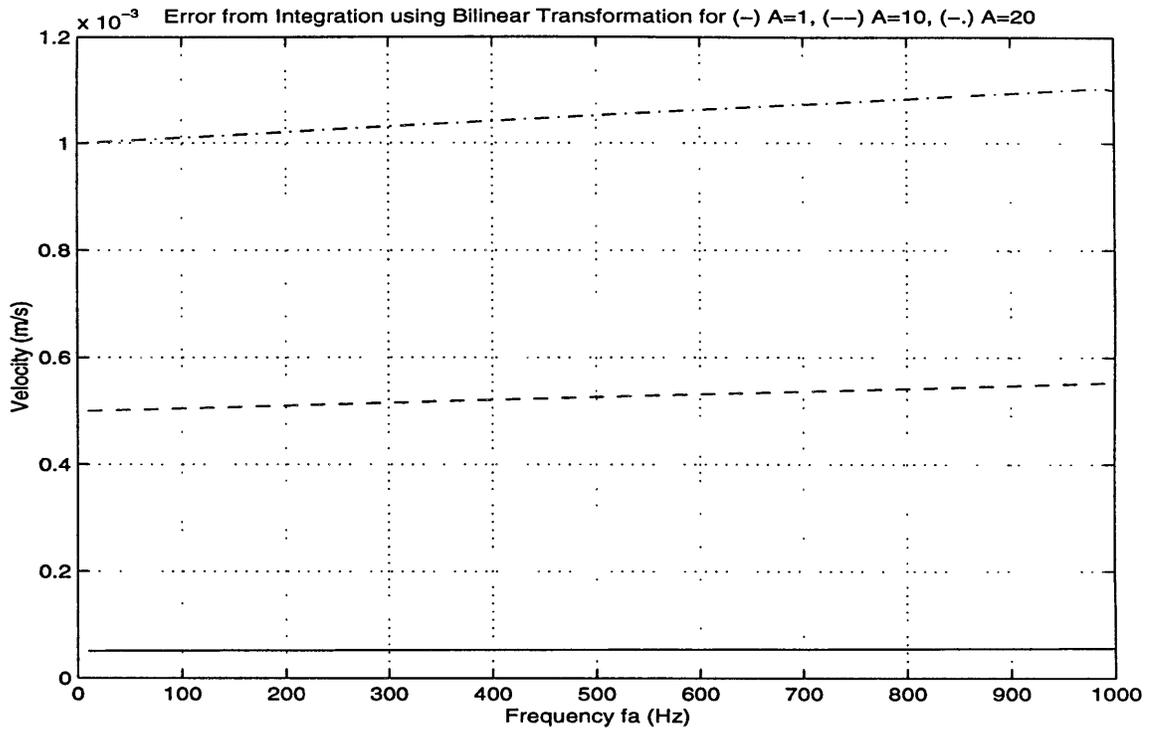


Figure 4.31: Maximum bilinear transform filter error.

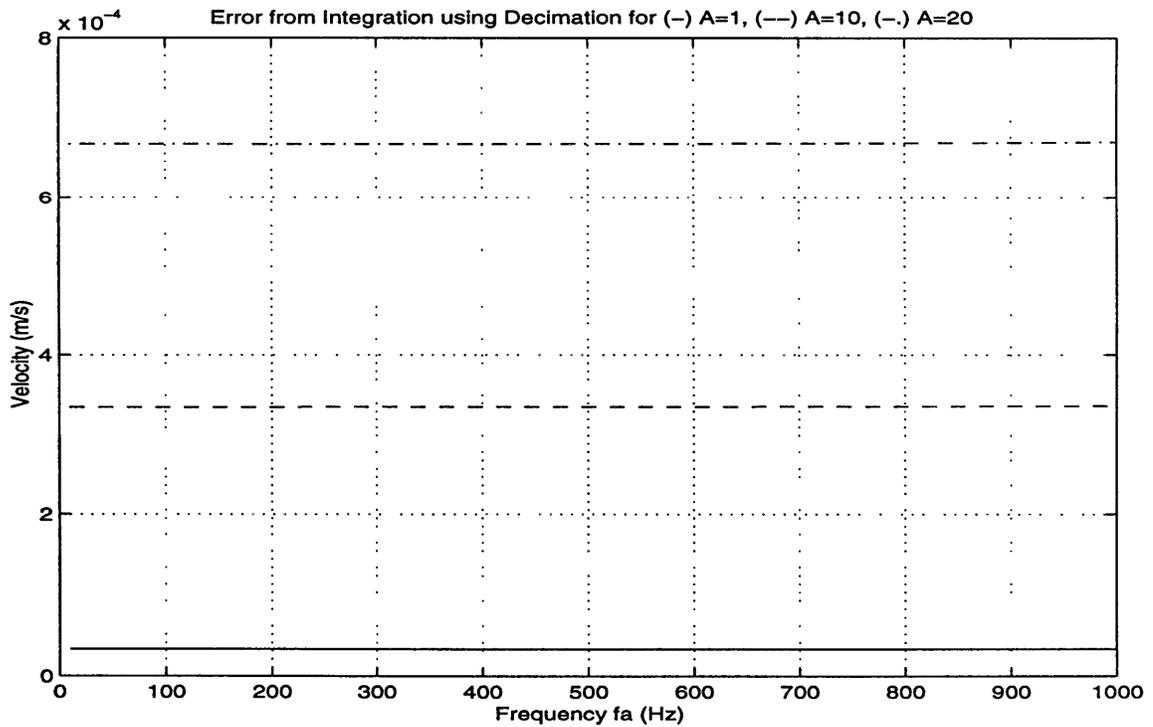


Figure 4.32: Maximum polyphase decimator filter error.

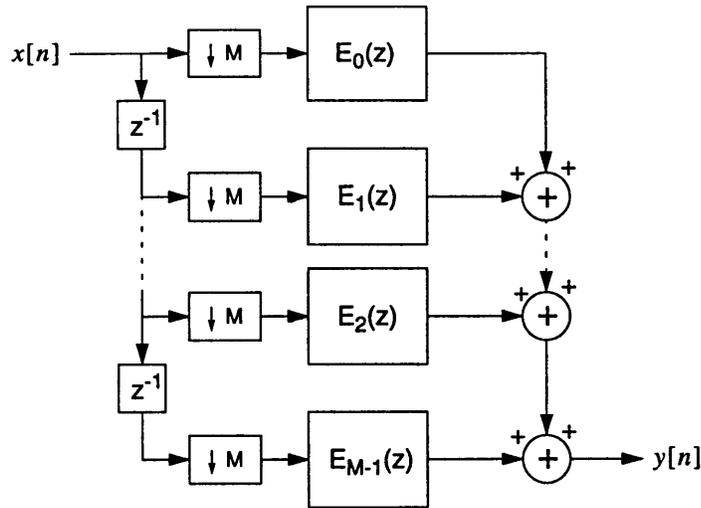


Figure 4.33: M-filter polyphase decimator.

velocity output. This makes the implementation of Simpson's Rule particularly attractive. Simpson's Rule, however, only decimates by a factor of two, whereas the sampling rate reduction of 10 kHz to 500 Hz is a factor of 20. The most useful aspect about the polyphase description in Fig. 4.29(c) is that it hints at how even greater integrator resolution can be obtained by further downsampling. Fig. 4.33 shows a general implementation of a decimation filter with M separate branches, where M is the downsampling ratio. The implementation of Simpson's Rule suggests that better integrators can be designed by downsampling by more than two with the appropriate choice of $E_0(z)$, $E_1(z)$, ... $E_{M-1}(z)$. Although this is a terribly intriguing possibility, Simpson's Rule suffices for this application, and attention turns now to the system analysis of the QCDM.

4.3 System Simulation

The significance of Section 4.2 is that it demonstrated the theoretical resolution that can be achieved by QCDM demodulator. Having done this, in addition to having designed the system, the results of Section 4.2 provide a valuable reference by which performance

degradation due to additive white noise can be measured. There are many noise sources in the system found in the transducer, the analog electronics, and the digital system itself, as partially shown in Fig. 4.2. This section studies how the QCDM Demodulator performs in the presence of noise, to determine under what conditions it can achieve the readout resolution desired of the VIA.

Fig. 4.2 showed that the noise is generated by the oscillator, the analog mixers, and the phase shifter. Section 3.1.2 discussed how quantization noise can degrade a signal. Within the DSP, there can be roundoff noises after multiplication and overflow following additions. Each noise source can have different statistics, and accurately modeling each one would be a tedious task, albeit an essential one for the future. At the present, this first analysis of the QCDM receiver vastly simplifies the system noise models by assuming that they are all wide-sense stationary, independent random processes. Then, all of the noise sources are modeled as a single noise source that has a variance equal to the sum of all the individual noise-source variances. As seen in Fig. 4.34, this single source, $n_1(t)$, is injected at the input of the DSP, added to $i(t)$ and $q(t)$, which are defined as in (4.23) and (4.24). The noise source has a uniform probability density distribution, zero-mean, and can produce values in the range of $-\frac{Q}{2} \leq n(t) \leq \frac{Q}{2}$. Therefore, its variance, or power, is $\frac{Q^2}{12}$.

In simulating the DSP system in Fig. 4.34 and its response to additive noise, the validation of three key results is sought. First, one of the most important specifications of the VIA readout scheme is that its electronics must add as little noise as possible to the highly stable oscillator waveforms. Therefore, the signal-to-noise ratio (SNR) at the input of the readout must be roughly preserved by the electronics. Second, FM signals exhibit better noise immunity than AM signals as a result of what is often called the “FM Advantage [17].” This advantage comes from the noise shaping that occurs in general FM demodulation: white noise added to an FM signal will be shaped into a triangular spectrum after

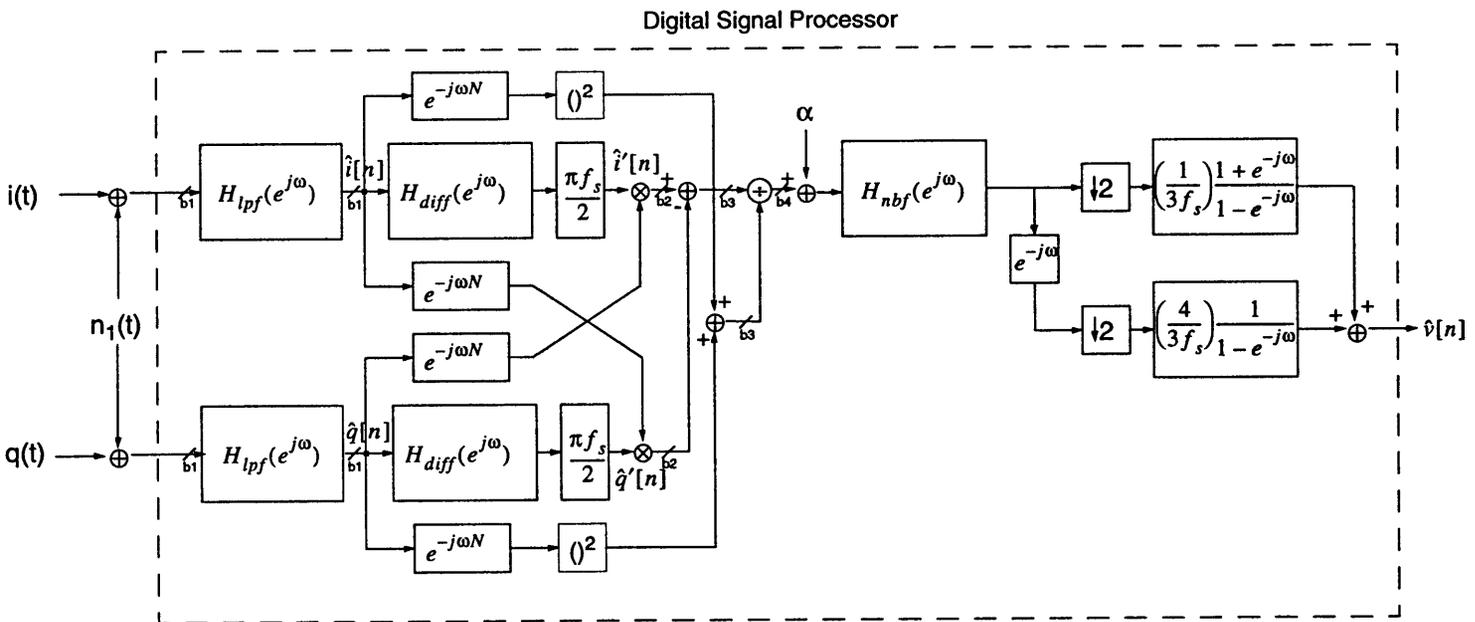


Figure 4.34: System simulated with QCDM.

demodulation. For a low frequency modulating signal, the SNR at the output can be high if the shaped noise is filtered out. The resolution of a demodulated FM signal can be approximated by (2.8) as [17]

$$\Delta a = \frac{2\pi f_a}{k_1} \sqrt{\frac{f_a}{3SNR}}, \quad (4.37)$$

where Δa is the acceleration resolution, f_a is the acceleration bandwidth, k_1 is the scale factor, and SNR is the signal-to-noise ratio of the FM signal. One goal of this simulation is to see how well the QCDM demodulator performs in comparison to (4.37). Thirdly, the simulations will seek to verify the validity of (3.34) and (3.35).

The Matlab function created to simulate Fig. 4.34 was called QCDM (Appendix A.10). This function is the culmination of all the subsystems designed in Section 4.2. However, the function is not restricted to simulating the system designed in the previous section, because the parameters of the system are left to be defined by the arguments of QCDM. Calling of the function takes the following form:

```
[names,values]=QCDM(M,Fp,Fs,Dp,Ds,Fp2,Fs2,Dp2,Ds2,Q,input,A,fa,fs,t);
```

Most of the arguments should be familiar by now, as most were seen in previous simulations. The variables M, Fp, Fs, Dp, Ds, Fp2, Fs2, Dp2, and Ds2 are defined here exactly as in Section 4.2.4. The variables A, fa, fs, and t are familiar, too, although they were not available to be freely defined in the calling of the other functions. They represent acceleration amplitude (g), acceleration frequency (Hz), sampling frequency (Hz), and the duration of the flight or simulation (seconds), respectively. The new arguments are Q and input. The variable Q determines the noise power of $n_1(t)$ that is added to $i(t)$ and $q(t)$. Since S1 and S2 in (4.23) and (4.24) are coded as unity in this simulation, Q also sets the SNR of the FM signals. The argument, input, can be specified as either 'cos' or 'con,'

which sets the input acceleration equal to either a cosine with amplitude A and frequency f_a , or a constant with amplitude A .

The simulation was run for a number of different test cases. For all of the simulations, the filters used were the ones designed in Section 4.2. Therefore, the first arguments were defined as $M=26$, $F_p=2.4e3$, $F_s=4.5e3$, $D_p=1e-8$, $D_s=1e-8$, $F_{p2}=2e2$, $F_{s2}=1.5e3$, $D_{p2}=1e-7$, $D_{s2}=1e-7$. Recall that the low pass filter designed by these specifications had order 45, and the narrowband filter had order 65. For each simulation, the amplitude of acceleration was $A=20$, the acceleration frequency for cosine input was $f_a=100$ Hz, and the sampling frequency was $f_s=10$ kHz. All other parameters not explicitly mentioned, were defined as in Section 4.2.2. The three parameters that varied between tests were Q , input, and t .

The function `QCDCM` returns in *values* the input SNR, the low pass filter order, the acceleration signal power, the acceleration error variance and maximum value prior to narrowband filtering, the narrowband filter order, the acceleration error and variance and maximum value after narrowband filtering, the output acceleration SNR, the maximum velocity error and the variance of the velocity, and finally, the velocity output SNR. The function also plots the acceleration and velocity outputs. The results of the simulation are summarized in Figures 4.35-4.42, and Tables 4.5 and 4.6.

The first impetus in analyzing the data is to observe whether or not the device is performing functionally as an accelerometer. That is, having set the acceleration signal at the input of the system, it is known precisely what the readout should output. As in Section 4.2, this knowledge is used to determine the readout error and resolution. Given a constant 20 g input, the demodulator is expected to produce $\hat{a}[n] = 20$ and a velocity $\hat{v}[n]$ that is a ramp with slope 20 g. Defining a cosine acceleration input, the same signal is the expected acceleration output, along with a sinusoidal velocity. Fig. 4.35-4.37 show the

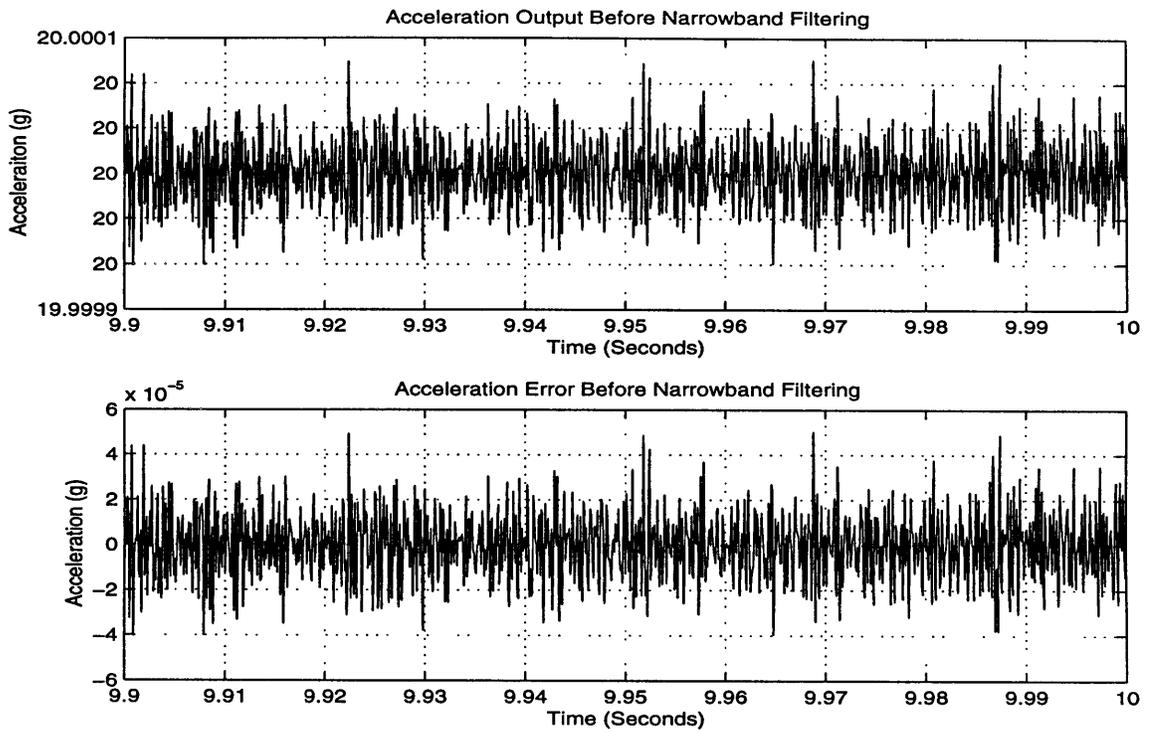


Figure 4.35: Simulation with QCDM for $t=10$ seconds, input='con', and $Q=1e-6$.

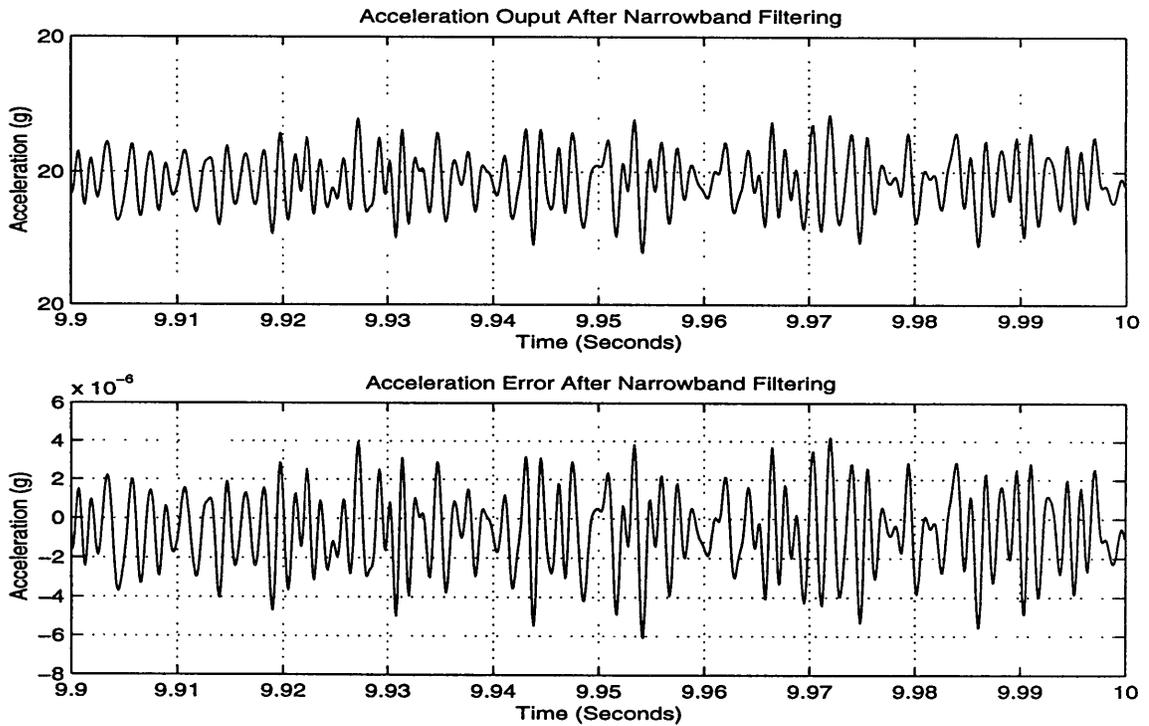


Figure 4.36: Simulation with QCDM for $t=10$ seconds, input='con', and $Q=1e-6$.

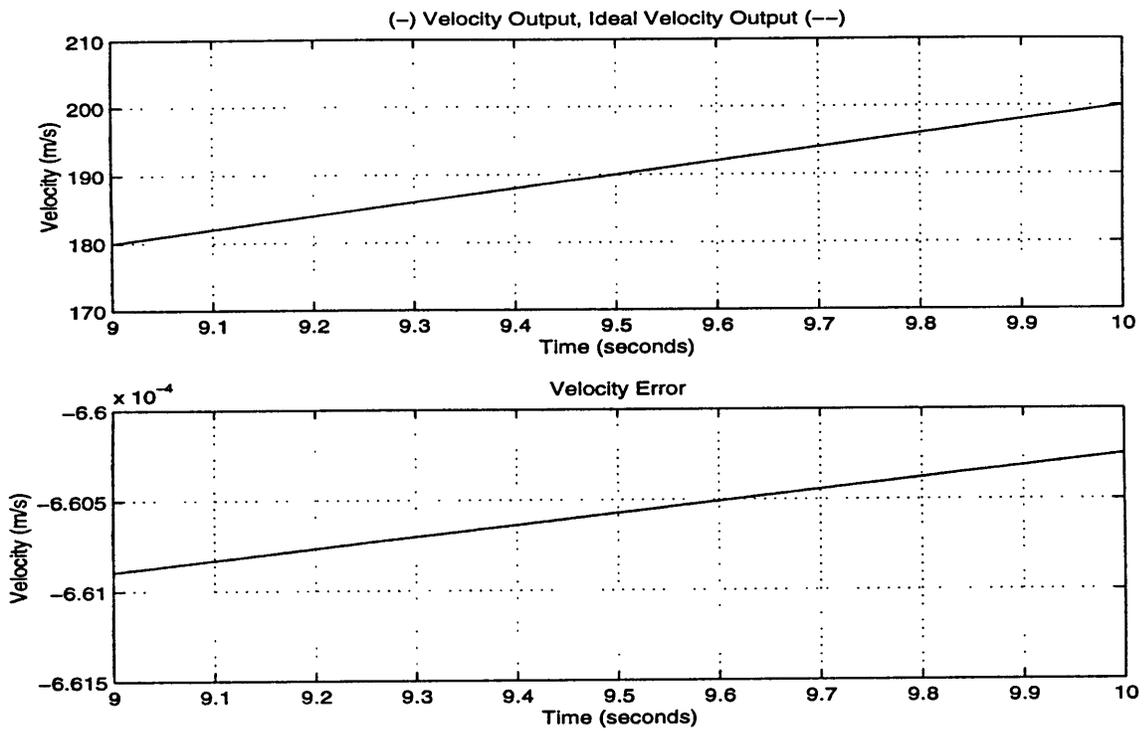


Figure 4.37: Simulation of QCDM with $t=10$ seconds, input='con', and $Q=1e-6$.

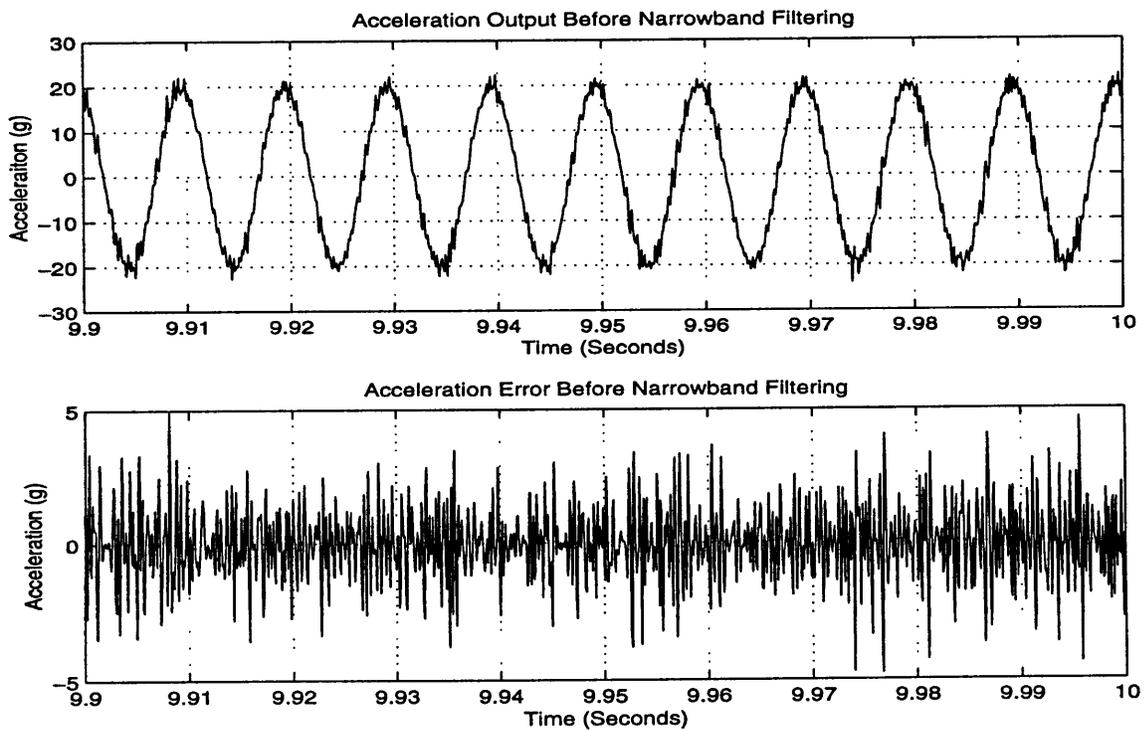


Figure 4.38: Simulation with QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$.

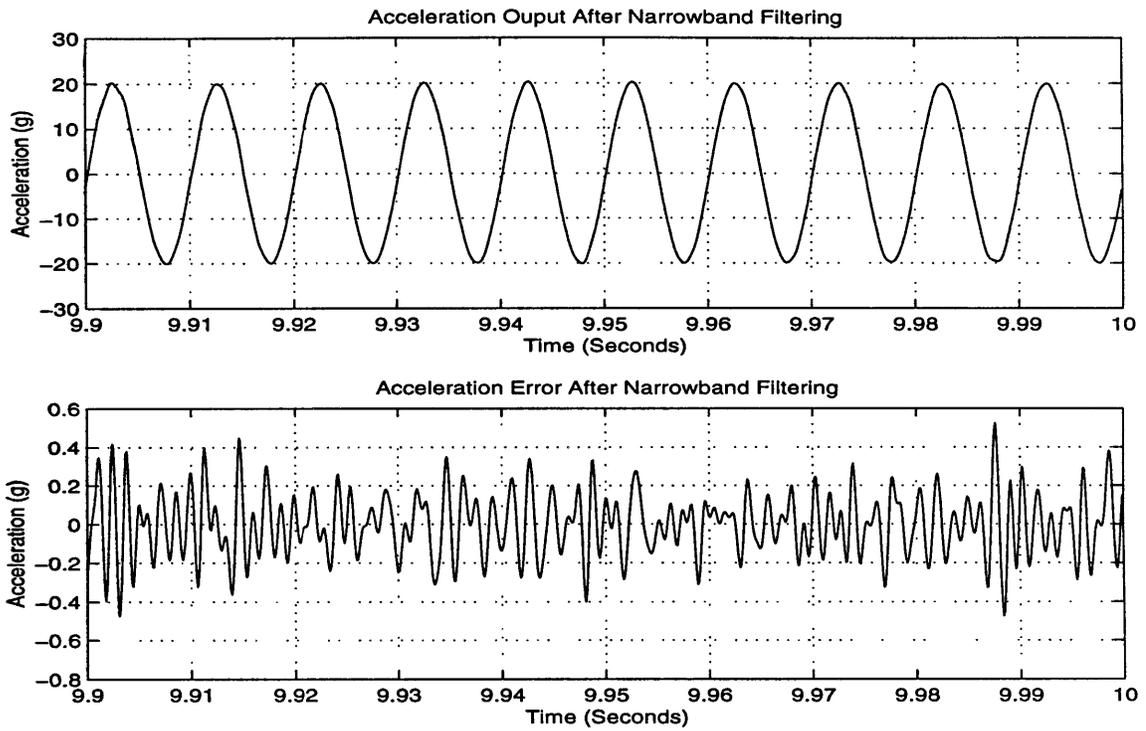


Figure 4.39: Simulation of QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$.

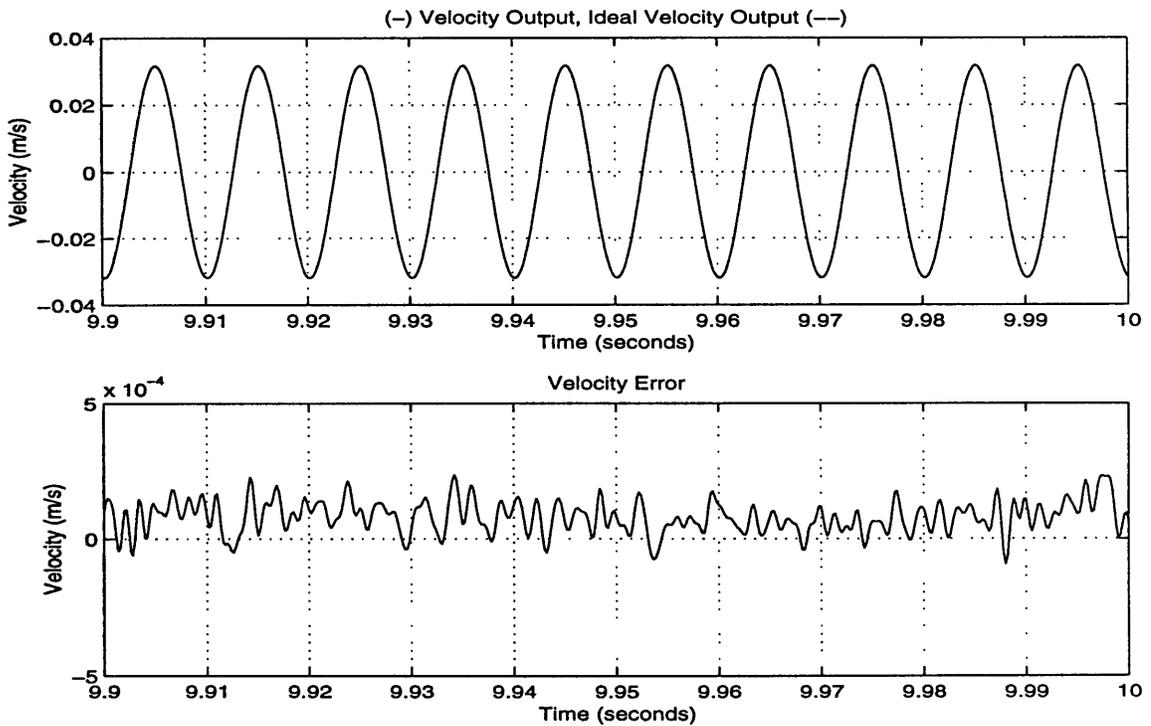


Figure 4.40: Simulation of QCDM for $t=10$ seconds, input='cos', and $Q=1e-1$.

results for $t=10$ seconds, $\text{input}='con'$, and $Q=1e-6$. The acceleration output is 20 g, as expected, and the velocity is a ramp with slope 20 g. Fig. 4.38-4.40 show the results for $t=10$ seconds, $\text{input}='cos'$, and $Q=1e-1$. The acceleration is a delayed cosine and the velocity is a delayed sinusoid. Note that due to the start-up transient from the filters, Matlab returns some NaN (Not a Number) values. The simulator, QCDM, zeros out these values and sets the initial condition of the integrator based on the ideal, expected output. This procedure must also happen in the calibration of the actual instrument.

Having seen that the simulation is functionally correct, a number of other observations are now highlighted. Figures 4.35, 4.36, 4.38, and 4.39 demonstrate the direct impact the narrowband filter has on the resolution of the acceleration signal. Its effectiveness is accentuated by the noise shaping characteristics of the demodulator, as will be discussed shortly. For both the constant and cosine inputs, the narrowband filter improves the resolution by at least one order of magnitude. Changing the parameters $Fp2$, $Fs2$, $Dp2$, and $Ds2$ will have a noticeable effect on the resolution. By limiting the acceleration bandwidth, $Fp2$ and $Fs2$ can be lowered, and the resolution should improve. Columns 4 and 5 of Table 4.5 show that the order of magnitude improvement was consistent throughout the tests.

Table 4.5 and Figures 4.35, 4.36, 4.38, and 4.39 also indicate the resolution of the QCDM demodulator for the various input noise levels. If the noise is assumed to be completely composed of A/D quantization noise and nothing else, the results can be compared to the derivations of Chapter 3, in which acceleration resolution was predicted using differentiator and quantization error. In equations 3.34, it was shown that the acceleration resolution could be approximated by the sum of the differentiator error and the quantization noise. Since the differentiator error for $M=26$ is $2.38e-7$ g (from Table 4.1), equation 3.34 predicts that for quantization error much larger than $2.38e-7$ g, the acceleration resolution is proportional to the quantization error. Comparing Q with the maximum

Table 4.5: Simulation Results from QCDM.

t (s e c)	Q	a(t) type	Maximum Acceleration Error before Narrowband Filter (g)	Maximum Acceleration Error after Narrowband Filter (g)	Maximum Output Velocity Error (gxsec)
1	1e-6	con	5.68790531971e-5	7.56501597365e-6	6.66361612468e-4
1	1e-6	cos	5.28181406391e-5	7.404855194792e6	8.35664734880e-5
1	1e-3	con	0.05739293792060	0.00601655061161	6.68549886203e-4
1	1e-3	cos	0.05817972520395	0.00772063397553	8.59779883802e-5
1	1e-1	con	5.18689127586051	0.70810531667502	8.76597497494e-4
1	1e-1	cos	5.36282253487642	0.70687902126487	2.88852863730e-4
30	1e-6	con	6.37566251775e-5	8.71279739911e-6	6.57034938001e-4
30	1e-6	cos	7.14743347458e-5	9.01011290466e-6	8.35665337903e-5
30	1e-3	con	0.06185275495234	0.00769001909055	6.57940850260e-4
30	1e-3	cos	0.06491357060339	0.00873317936462	8.64822805700e-5
30	1e-1	con	6.29793117540774	0.82625664119261	9.72919373680e-4
30	1e-1	cos	6.64586007000836	0.95952948548596	3.59174166181e-4

acceleration error after narrowband filtering in Table 4.5, it is evident that the quantization resolution is roughly preserved by the demodulator, but degraded by nearly one order of magnitude. This result suggests that as (3.34) predicts, one fundamental limitation of the readout resolution is the quantization in the A/D conversion. Section 4.2 showed how the acceleration error is also dependent on the differentiator error as (3.34) suggests. However, Section 4.2 showed that the resolution depended significantly on the FM bandwidth of the signal, and by Table 4.5, the bandwidth of the acceleration signal, too. Furthermore, when other noise sources are considered, the contribution of quantization error to the size of Q decreases in proportion, but (3.34) does not include these other considerations. Equation 3.34 is clearly an inadequate approximation of the actual resolution of the QCDM demodulator. The significance and utility of (3.34), however, is that it does indeed provide

Table 4.6: Simulation Results from QCDM (continued).

time (sec)	Q	a(t) type	Acc Ave Pow	FM SNR in (dB)	Acc Err Var	Acc SNR out (dB)	Vel Err Var	Vel SNR out (dB)
1	1e-6	con	400	122	2.44e-10	140	2.16e-3	41.9
1	1e-6	cos	200	122	1.98e-10	137	3.83e-5	11.3
1	1e-3	con	400	61.7	2.37e-4	80.5	2.13e-3	41.9
1	1e-3	cos	200	62.0	2.09e-4	76.9	3.83e-5	11.3
1	1e-1	con	400	21.8	2.54	39.9	2.16e-3	41.9
1	1e-1	cos	200	22.0	2.04	37.0	3.82e-5	11.3
30	1e-6	con	400	122	2.38e-10	140	7.34e-5	86.1
30	1e-6	cos	200	122	2.04e-10	137	1.30e-6	25.9
30	1e-3	con	400	61.8	2.38e-4	80.2	7.34e-5	86.1
30	1e-3	cos	200	62	2.03e-4	77.2	1.30e-6	25.9
30	1e-1	con	400	21.8	2.38	40.3	7.34e-5	86.1
30	1e-1	cos	200	22.0	2.05	37.2	1.30e-6	25.9

a minimum bound on the readout error, as was originally intended. That is, for a given differentiator and quantization error, (3.34) indicates the best possible resolution attainable. The data in Table 4.5 support this conclusion about (3.34).

In (3.35), the velocity error was predicted to be the sum of a discrete-time integration error, $e_{dti}[n]$, and the accumulation of acceleration error over time, $e_{va}[n]$. Fig. 3.37b. shows that the slope of the error is approximately

$$\frac{(-6.6025 \times 10^{-4})g - (-6.608 \times 10^{-4})g}{1 \text{ sec}} = 5.5 \times 10^{-7} g.$$

This should give an indication of the acceleration error. In Table 4.5, for $Q=1e-6$, the maximum acceleration errors are about $8e-6$ g. The average acceleration error would probably be less, and closer to $5.5e-7$ g, but the collected data is inconclusive in this regard. The dif-

ference between the tests done for $t=1$ and $t=30$ seconds give some unexpected results, too. For instance, the maximum acceleration error multiplied by the duration of the flight, t , should give an indication of the minimum velocity error as in (3.35). The data does not show this. The results are inconclusive in this regard, as well, and simulations should be run for larger durations than $t=30$ seconds. This time period was chosen due to the limitations of the Sun machines used to run the simulations.

Using Tables 4.5 and 4.6, the expression (4.37) can be tested. For $f_a=100$, and $k_1=50$ Hz/g, the expected demodulator resolution for SNR's equal to 122 dB, 62 dB, and 22 dB, are $5.76e-5$ g, $5.76e-2$ g, and 5.76 g, respectively. From Table 4.6, the FM SNRs 122 dB, 62 dB, and 22 dB correspond to the Q values $1e-6$, $1e-3$, and $1e-1$. Table 4.5 shows that these Q values correspond to acceleration resolutions of $7.40e-6$ g, $7.72e-3$ g, and $7.07e-1$ g, respectively. In each case, the QCDM receiver resolution is better than the resolutions predicted by (4.37). Interesting to note, however, is that (4.37) actually does much better job of predicting the acceleration signal before the narrowband filtering: $5.38e-5$ g, $5.82e-2$ g, and 5.36 g. The derivation of (4.37) in [17], however, seems to suggest that it should predict resolution for *after* narrowband filtering.

Fig. 4.41 and Fig. 4.42 show the noise shaping effects of the demodulator. For $t=1$, $Q=1e-1$, and input='cos,' the spectrum of $i(t)$ corrupted with noise is shown in Fig. 4.41a. Comparing this plot with Fig. 4.16b, shows clearly the effect the noise has on the spectrum. Fig. 4.41b. shows the spectrum of the acceleration output before narrowband filtering. Notice that the lower frequency noise has been shifted to the higher frequencies, creating a triangularly-shaped noise spectrum, similar to the quantization noise shaping of delta-sigma modulators (Section 3.4.3). Subtracting out the 100 Hz sinusoid signal would show this more clearly, although a plot of this is not included here. The attenuation of the spectrum at normalize frequency 0.4 results from the low pass filtering done at the front

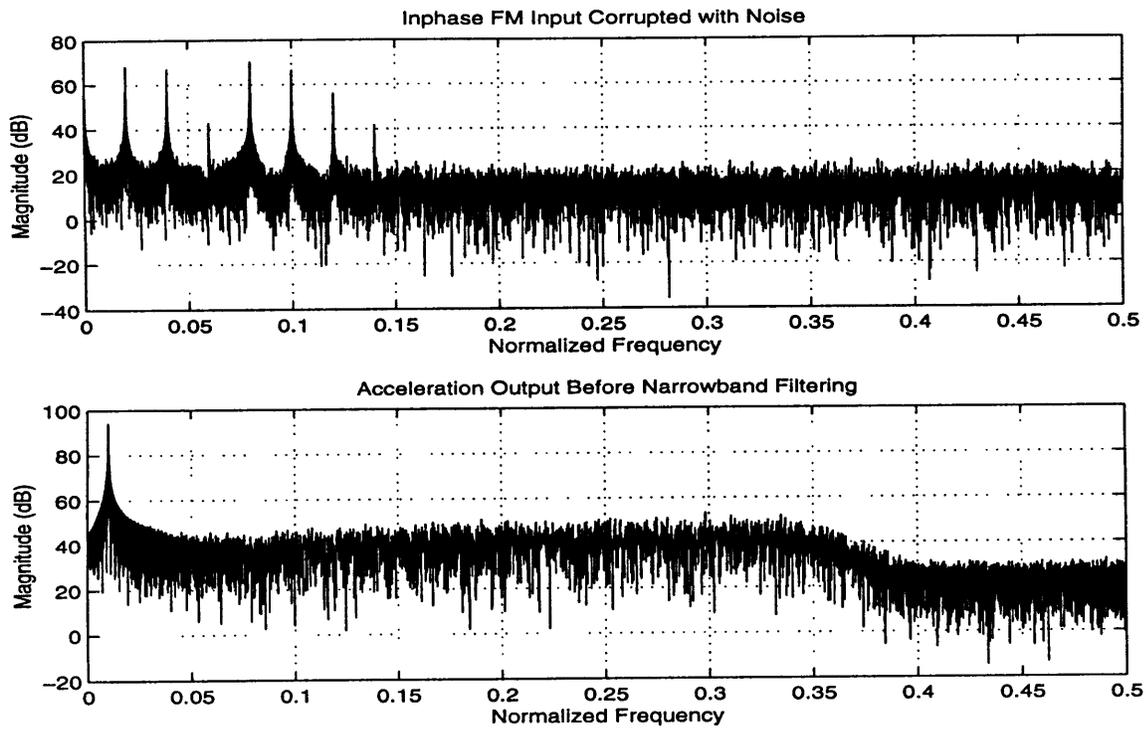


Figure 4.41: Simulation with QCDM for $t=1$ second, input='cos', and $Q=1e-1$.

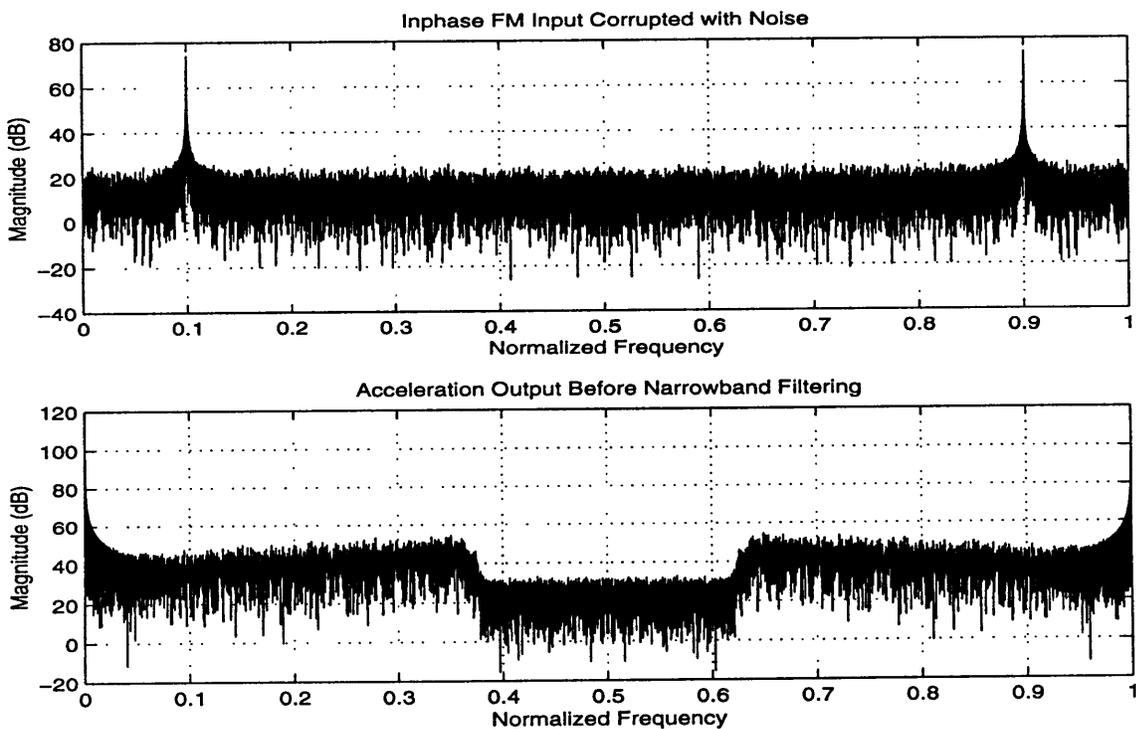


Figure 4.42: Simulation with QCDM for $t=1$ second, input='cos', and $Q=1e-1$.

end of the DSP. Fig. 4.42 shows similar results for input='con.' The significance of the noise shaping characteristics of the FM demodulator, is that by further bandlimiting the acceleration signal, the narrowband filter F_{p2} can be decreased (increasing the filter order, of course), and the acceleration resolution should be improved.

In the same way that the DSP input noise could be viewed as purely quantization noise, it could also be viewed as the noise from a number of different sources, separately or collectively, including the oscillator, roundoff noise within the DSP, mixer and phase-shifter noise, and quantization noise. These simulations have shown that if the sum of all of these noise sources results in an SNR of 122 dB at the DSP input, then the QCDM demodulator designed in Section 4.2 can achieve $8\mu g$ resolution.

4.4 Further Tests and Design Directions

A number of further tests should be done using QCDM. To improve the demodulator noise performance, the response of each subsystem to noise should be looked at individually to see where possible gains could be obtained. The simulations should also be run for longer durations of time on a machine that has more memory available than the ones used for this thesis. Next, better and more accurate noise models should be used in the simulation. Decomposing the noise sources might be a significant step in reducing the degradation they cause. For example, the FM noise shaping affects noise that is added to the carrier of the FM signal. It is not clear whether or not the demodulator would shape any noise added to the modulating signal itself in the oscillator (such as $n_1(t)$ and $n_2(t)$ in Fig. 4.2). Furthermore, the oscillator, as discussed in Chapter 2, is expected to have extremely high frequency stability. How good the SNR measure in (2.8) is should be checked. Still further steps would require that the VIA oscillator parameters be better understood. That is, the scale factors, the acceleration bandwidth, the maximum acceleration, the oscillator carrier frequencies and their offset, should be settled upon. Equally as important, perhaps,

is obtaining knowledge about the specific inertial navigation system the readout scheme would operate with. This will significantly impact such readout system parameters such as sampling rate, and output digital word length and format.

The most important theoretical task that lies ahead is research on the delta-sigma modulator used for the A/D conversion. As seen in Section 3.4.2 and Section 4.3, the A/D quantization places one of the most significant restrictions on the readout resolution. Much research is currently being done on high resolution DSMs and bandpass DSMs, and the 32-bit resolution required for this application will seemingly soon be much more common. A fourth-order DSM running at 2.56 MHz, with oversampling ratio (OSR) equal to 64, was simulated to achieve SNDR=94.5 dB for a 20 kHz input sinusoid [24]. A single loop, multi-bit DSM sampling at 32 MHz, was simulated to achieve 16-bit accuracy, or SNR=85 dB, for a 500 kHz input signal [25]. Tan and Eriksson [26] cascaded two second-order stages to create a fourth-order DSM, and simulated the sampling of a 1 kHz input at a rate of 1.024 MHz for SNRs beyond 150 dB. Fischer and Davis [27] simulated a third-order second-order cascade DSM by sampling a 320 kHz input at 20.48 MHz to achieve an SNR=136 dB. Still another effort studied a fourth-order, single-loop, single-bit DSM architecture and reported that sampling a 600-700 Hz signal at 256 kHz, SNRs could achieve 130 dB [28]. The literature seems to suggest that DSMs are worth looking at for the VIA application.

The next design iteration of the QCDM receiver should include a bandpass DSM as described in [12]. This scheme would perform digital quadrature demodulation, and eliminate the need for using noisy analog mixers and phase shifters. The phase shifter could be implemented with the Hilbert Transform in the digital domain. The DSM required for the QCDM readout certainly pushes the envelope of available technology, and a DSM specific

to this application would probably have to be designed. Looking at oversampling delta-sigma frequency discriminators mentioned in Section 3.4.1 would be a good idea.

A number of practical implementational issues need to be considered, as well. Implementing the readout with a DSP would require efficient coding that would dictate the maximum possible sampling rate. In addition, selection of an appropriate processor would be necessary. Implementing the readout with an ASIC would require prototyping with a Field Programmable Gate Array, and would first have to be simulated. The different possible implementations should be compared and contrasted. Consideration of other multirate digital signal processing techniques should be considered, including the use of efficient sinc filters and differentiators as described in [29].

In summary, the analysis of this chapter has shown the digital system parameters necessary for the QCDM FM demodulator to theoretically achieve acceleration resolutions on the order of $1\ \mu\text{g}$.

Chapter 5

Conclusion

Why should the frequency demodulator be considered in the design and implementation of the VIA readout? This thesis attempted to address this inquiry with two responses. First, the entire essence of Chapter 3 was devoted to answering this question as it derived analytical expressions with which comparison of phase and frequency readout methods was made possible. One key issue that was raised described how the phase readout schemes can be viewed as performing 1-bit digital quantization of the oscillator outputs at high sampling rates. Implementing a multi-bit A/D convertor could theoretically achieve higher resolution by extracting more intelligence from the oscillator signal amplitude at lower sampling rates. The implementation of multi-bit quantizers pointed naturally to reading out frequency because instantaneous frequency can be derived by taking the derivative of an angle modulated signal, such as those generated by the VIA oscillator. Later in Chapter 3, four possible frequency readout schemes were described, with ensuing analysis of the CDM demodulator in particular. The CDM receiver was compared with two common phase readouts, and no single method proved to be superior than the others. Since frequency readouts have seemingly received less attention for inertial instrument applications, focusing on them in this research endeavor was particularly attractive.

A second general response to the initial question stems from the high resolution requirement of the VIA. Often times, the only way to solve a difficult, complex problem is to decompose it into smaller problems, each of which can be solved independently. The high resolution goal can be seen as a complex problem, and frequency discrimination methods decompose the problem into finer components than phase demodulators do. For example, a high resolution acceleration readout could achieve much better velocity resolu-

tion than a velocity readout for short time periods. The acceleration readout has an extra dependence on time, that corresponding velocity readouts do not. One of the main benefits of decomposing a problem is the general increase of understanding that can happen. For instance, this thesis looked carefully at three pieces of the high resolution problem that revealed some general results about the nature of the problem. First, the importance of accurate digital integration for frequency readouts was discussed, and three numerical integration methods were explored to solve the problem. Second, the relationship between FM bandwidth and the receiver resolution was studied, showing that for this high resolution application, the required bandwidth exceeded twice the Carson Limit. Lastly, it was learned that the resolution of the readout depended highly on the quantization of the analog oscillator outputs. A first order theoretical analysis showed that a fourth order delta-sigma modulators might be capable of achieving the necessary resolution.

The thesis also addressed the following question: what can be learned by exploiting the analogy of the FM transmitter in a digital communication system? The thesis described that little could be obtained from looking at digital transmission systems in the virtual bit-link sense, but much could be gained by looking at analog communication theory and digital signal processing techniques. First, potential benefits of using quadrature demodulation were described as an ideal front end to the CDM demodulator. Secondly, the thesis discussed the appropriateness of using digital finite impulse response filters because of their linear phase characteristics. One of the most fascinating aspects of the thesis was the discovery that highly accurate digital integration could be achieved by implementing Simpson's Rule with a polyphase decimator, which hinted at a means of designing even better digital integrators. Lastly, although this thesis did not, unfortunately, analyze delta-sigma modulators in depth, it was discovered in the literature

searches that bandpass delta-sigma modulation is an emerging technology that could be particularly useful for this application.

The most important benefit of exploring the communication system analogy, however, was not simply the discovery of useful techniques that could theoretically work to achieve a high resolution readout. Bandpass delta-sigma modulators and polyphase filterbanks are currently active areas of research in their respective fields. And although there may not exist a DSM right now that can achieve the exact, desired resolutions for the VIA, it is likely that it can and will be developed somewhere in the near future. Since the Vibrating Island Accelerometer is a long-term project, it might be worthwhile to design and research readout schemes that implement forward-looking technology.

Finally, can an FM receiver theoretically achieve the high resolution required of the VIA readout? In theory, the answer, of course, is "yes." The more important contribution of this thesis is that it showed how it might be achieved with the design and simulation of a specific frequency demodulator: the QCDM discriminator. A Matlab function, QCDM, was written that enables easy design and simulation of the frequency readout scheme by specifying a number of pertinent parameters. Preliminary tests were done on the readout designed in Section 4.2. The data supported some of the derivations in Chapter 3, and was inconclusive regarding others. The simulations did successfully demonstrate the noise shaping characteristics of the FM demodulator, while showing that the QCDM structure performs better than a common metric (4.37) used to predict the resolution of FM receivers. Lastly, Section 4.4 suggested further QCDM tests that could be performed, and recommended direction for future theoretical and practical work to follow up the research described in this now-concluded thesis.

Appendix A

Simulations

A.1 SIM1

```
function [y,e]=sim1(M)
%
% SIM1(M) designs a discrete-time differentiator of length M+1 using the
% the Remez Algorithm, plots its impulse response, frequency response,
% and magnitude error, and then plots the results of differentiating an
% example input. The differentiator is designed for the VIA QCDDM
% demodulator acceleration readout scheme. The function also returns the
% maximum differentiator magnitude error for a given input bandwidth and
% sampling frequency, as well as the error in differentiated example.
%
% If M is even, the differentiator, Hdiff, is a Type III FIR filter with
% integer delay M/2. If M is odd, then Hdiff is a Type IV FIR filter with
% non-integer delay M/2.
%
% The sampling frequency, fs, is 10-kHz. The input signal, i(t), used in
% the example plots is defined as in Fig. 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0
% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)))
%
% These parameters result in the maximum expected modulation index of the
% VIA oscillator outputs.
%
% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
fs=10e3, % fs=10-kHz

% DIFFERENTIATOR
hdiff=remez(M,[0 0.5],[0 1],'differentiator');% Differentiator
hdiff_dft=fft(hdiff,356);
N=0:1/356:(1-(1/356));
hideal=4*N,
error=hideal-abs(hdiff_dft);

figure(1)
n=0:1:length(hdiff)-1;
subplot(3,1,1); stem(n,hdiff); grid;
title('Impulse Response of (M-1)-Point Optimal Differentiator')
xlabel('Sample'); ylabel('Amplitude');

subplot(3,1,2); plot(N,abs(hdiff_dft),'-'.N,hideal,'-');
axis([0 0.5 0 3]); grid
title('Magnitude Response of (M-1)-Point Optimal Differentiator')
xlabel('Normalized Frequency'); ylabel('Magnitude')

subplot(3,1,3); plot(N,error);
%axis([0 0.2 -5e-11 5e-11]);
grid
title('Magnitude Error of (M-1)-Point Optimal Differentiator');
xlabel('Normalized Frequency'); ylabel('Error')

%***** INPUT DEFINITION *****
f=1e3;% Approximate Input Bandwidth from Carson Limit
% calculated in Section 3.3.1.

% MAXIMUM DIFFERENTIATOR ERROR
% Maximum differentiator error for fs=10-kHz and input bandwidth=1-kHz
% occurs at 1-kHz/10-kHz=0.1, which is the (0.1*356)th sample of hdiff.
max_hdiff_error=max(abs(error(1:floor((f/fs)*356))))
axis([0 0.2 (-2*max_hdiff_error) (2*max_hdiff_error)])
```

```

% CONTINUOUS TIME AXIS
t=0:1/fs:10*(10/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.

% DISCRETE TIME AXIS
L=10*(10/f)*fs,
n=0:1:L;

% ***** FM INPUT *****
fa=100;% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
A=20;% A=20 g <-- Maximum acceleration, for max mf.
S1=1;% S1=1 unitless, voltage.
i=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));

% INPUT SPECTRUM
L2=2^16;
N=0:1/L2:(1-(1/L2));
i_dft=fft(i,L2);

figure(2)
subplot(2,1,1)
plot(t,i)
grid;
title('Input Signal i(t)')
xlabel('Time (seconds)')
ylabel('Amplitude')
subplot(2,1,2)
plot(N,abs(i_dft));
grid;
axis([0 0.25 0 400])
title('Input Signal i(t) 65536-Point DFT')
xlabel('Normalized Frequency')
ylabel('Magnitude')

% IDEAL OUTPUT
nd=M/2;% Delay
yi=(-A*k1*2*pi*cos(2*pi*fa*((n-nd)/fs)).*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));

% ***** FILTERED OUTPUT *****
a=1,
b=hdfff;
yi=(pi*fs/2)*filter(b,a,i);% Notice that the output needs to be scaled
% by (pi*fs/2) to be consistent with the
% traditional definition of ideal discrete-
% time differentiators. See Oppenheim and
% Schafer p. 96

figure(3)
subplot(2,1,1)
plot(t,yi,'-',t,yii,'--')
grid;
title('Differentiated Output y(t)')
xlabel('Time (seconds)')
ylabel('Amplitude')
subplot(2,1,2)
yi_dft=fft(i,L2);
plot(N,abs(i_dft));
grid;
axis([0 0.25 0 400])
title('Output Signal y(t) 65536-Point Spectrum')
xlabel('Normalized Frequency')
ylabel('Magnitude')

% ***** MAGNITUDE ERROR *****
error_i=yii-yi;
error_max=max(abs(error_i(floor(L/2):L)))% Look later samples.

figure(4)
subplot(1,1,1)
plot(t,error_i)
axis([0 (100/f) (-5*error_max) (5*error_max)])
grid;
title('Output Error')
xlabel('Time (seconds)')
ylabel('Amplitude')

```

```

y=yi;
e=error_i;

```

A.2 SIM2

```

function [y]=sim2(M)
%
% SIM2(M) designs discrete-time differentiators of length M+1, just as
% in SIM1(M), and then implements the cross-differentiate multiply scheme
% using infinite precision floating point mathematics. This simulation
% demonstrates that the method works to reject the amplitude modulation
% that occurs during the differentiation.
%
% This simulation shows that the error in the differentiator is
% proportional to the acceleration resolution, as stated in Section 3.4.2
% of the thesis. This verifies the portion of Equation 3.34 that indicates
% dependency of acceleration resolution on the differentiator error.
%
% This simulation also shows the importance of the necessary delays
% included in the non-differentiated branches of the demodulator. If the
% delays are not accounted for exactly, there is added error to the
% acceleration readout.
%
% Lastly, the simulation helped to discover the necessity of certain scale
% factors that are now included in this code.
%
% Again, the sampling frequency, fs, is 10-kHz. The input signal, i(t),
% used in the example plots is defined as in Fig. 3.7 of this thesis, with
% the following parameter values:
%
%   a1(t)=A*cos(2*pi*fa*t)
%   a2(t)=A*cos(2*pi*fa*t)
%   k1=k2=50-Hz/g
%   A=20-g
%   fa=100-Hz
%   S1=S2=1
%   i=S1*cos((k1*A/fa)*sin(2*pi*fa*t))
%   q=S2*sin((k2*A/fa)*sin(2*pi*fa*t))
%
% These parameters result in the maximum expected modulation index of the
% V1A oscillator outputs
%
% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
fs=10e3, % fs=10-kHz

%***** INPUT DEFINITION *****
f=1e3;% Input frequency

% CONTINUOUS TIME AXIS
t=0:1/fs:10*(10/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.

% DISCRETE TIME AXIS
L=10*(10/f)*fs;
n=0:L,

% ***** FM INPUT *****
fa=100,% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
A=20;% A=20 g <-- Maximum acceleration, for max mf.
S1=1;% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

si=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));
sq=S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs)));

% DIFFERENTIATE
hdiff=remez(M,[0 0 5],[0 1],'differentiator');% Differentiator
a=1,
b=hdiff,

```

```

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq),

% IDEAL DIFFERENTIATED OUTPUT
nd=M/2;% Delay
sidi=(-A*k1*2*pi)*(cos(2*pi*fa*((n-nd)/fs))).*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));
sqdi=(A*k2*2*pi)*(cos(2*pi*fa*((n-nd)/fs))).*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));

% DELAYED (IDEAL) INPUT
side=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqde=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% SQUARED DELAYED INPUT
sis=side *side;
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sidi.*sqde)-(sqdi.*side);

% IDEAL OUTPUT
a=A*cos(2*pi*fa*((n-nd)/fs));
figure(1)
subplot(2,1,1)
plot(t,a)
axis([0 0.1 -50 50])
title('Acceleration Input/Ideal Acceleration Output')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
grid
subplot(2,1,2)
L2=2^16;
N=0:L2:(1-(1/L2));
a_dft=fft(a,L2);
semilogy(N,abs(a_dft));
grid
axis([0 0 02 1e0 1e5])
title('Ideal Acceleration Output a[n] 65536-Point DFT')
xlabel('Normalized Frequency')
ylabel('Magnitude')

% ACCELERATION OUTPUT
acceleration_out=(dividend./divisor)/(-2*pi*k1);
figure(2)
subplot(2,1,1)
plot(t,acceleration_out)
title('Acceleration Output')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
axis([0 0.1 -50 50])
grid
subplot(2,1,2)
acceleration_out_dft=fft(acceleration_out,L2);
semilogy(N,abs(acceleration_out_dft));
grid
axis([0 0.02 1e0 1e5])
title('Acceleration Output 65536-Point DFT')
xlabel('Normalized Frequency')
ylabel('Magnitude')

% ACCELERATION ERROR
error_acceleration=(acceleration_out)-a;
error_max=max(abs(error_acceleration(floor(L2)/2:1:L)))

figure(3)
subplot(2,1,1)
plot(t,error_acceleration)
axis([0 0.1 (-5*error_max) (5*error_max)])
grid;
title('Acceleration Error')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
subplot(2,1,2)

```

```

error_acceleration_dft=fft(error_acceleration,L2);
semilogy(N,abs(error_acceleration_dft));
grid
%axis([0 0.02 1e0 1e5])
title('Acceleration Error 65536-Point DFT')
xlabel('Normalized Frequency')
ylabel('Magnitude')

```

A.3 SIM3

```

function [error_max]=sim3(M,Ni)
%
% SIM3(M,Ni) generates ideally bandlimited spectra of the frequency
% modulated input using bessel function formulation. The ideally
% bandlimited FM signal is then demodulated by the CDM demodulator, as
% described in SIM2(M), and the acceleration error is determined. This
% function plots the bandlimited input and its spectrum, and plots the
% error and the error spectrum.
%
% M is the differentiator order, and Ni is the number of sidebands used in
% the bessel function to generate the FM input. The bandwidth of the
% FM input is 2*Ni*fa, where fa is the acceleration frequency. Bandwidth,
% in this case, refers to the width of the band of positive frequencies of
% the spectrum centered at DC. The total input bandwidth, including
% negative frequencies, would be 4*Ni*fa.
%
% Using SIM2, the resolution limitation due to the differentiator can be
% determined. SIM3 determines the resolution limitations due to both
% the differentiator and the FM bandwidth. Therefore, by comparing the
% results of SIM3 to SIM2, it is possible to estimate the effect of FM
% bandwidth on the theoretical CDM demodulator acceleration resolution.
%
% The function returns error_max, the maximum acceleration error due to
% the parameters M and Ni.

% SAMPLING FREQUENCY
fs=10e3, % fs=10 kHz

%***** INPUT DEFINITION *****
f=1e3;% Input frequency

% CONTINUOUS TIME AXIS
t=0/1/fs:10*(100/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.

% DISCRETE TIME AXIS
L=10*(100/f)*fs;
n=0.1:L;

% ***** FM INPUT *****
fa=100;% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
A=20;% A=20 g <- Maximum acceleration, for max mf.
S1=1;% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

% THESE ARE THE FUNCTIONS I NEED TO APPROXIMATE WITH BESSEL:
% si=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));
% sq=S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs)));

% GENERATE BANDLIMITED si
Bi=k1*A/fa;
si=S1*besselj(0,Bi)*ones(1,L+1);
for i=1:Ni
    si=si+(2*S1*besselj(2*i,Bi)*cos(2*i*2*pi*fa*(n/fs)));
end;

```

```

% GENERATE BANDLIMITED sq
Nq=Ni,% Number of sidebands. Bandwidth if fa*Nq.
Bq=k2*A/fa;
sq=zeros(1,L+1);
for i=0:Nq
    sq=sq+(2*S2*besselj(((2*i)+1),Bq)*sin(((2*i)+1)*2*pi*fa*(n/fs)));
end;

%***** FROM sim2.m *****

% DIFFERENTIATE
M=26;
hdiff=remez(M,[0 0.5],[0 1],'differentiator');% Differentiator
a=1;
b=hdiff;

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq);

% IDEAL DIFFERENTIATED OUTPUT
nd=M/2;% Delay
sidi=(-A*k1*2*pi)*(cos(2*pi*fa*((n-nd)/fs))).*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));
sqdi=(A*k2*2*pi)*(cos(2*pi*fa*((n-nd)/fs))).*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));

% DELAYED (IDEAL) INPUT
% side=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
% sqde=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% GENERATED DELAYED BANDLIMITED INPUT side
side=S1*besselj(0,Bi)*ones(1,L+1);
for i=1:Ni
    side=side+(2*S1*besselj(2*i,Bi)*cos(2*i*2*pi*fa*((n-nd)/fs)));
end;

% GENERATED DELAYED BANDLIMITED INPUT sqde
sqde=zeros(1,L+1);
for i=0:Nq
    sqde=sqde+(2*S2*besselj(((2*i)+1),Bq)*sin(((2*i)+1)*2*pi*fa*((n-nd)/fs)));
end;

% SQUARED DELAYED INPUT
sis=side.*side;
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sidi.*sqde)-(sqdi.*side);

figure(1)
subplot(2,1,1)
plot(t,si)
axis([0 0.01 -1 1])
title('In-phase Input i(t) Ideally Bandlimited to 2*Ni*fa Hz')
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid
subplot(2,1,2)
L2=2^16;
N=0:1/L2:(1-(1/L2));
si_dft=fft(si,L2);
semilogy(N,abs(si_dft));
axis([0 0.25 1e-1 1e4])
%plot(N,abs(si_dft));
%axis([0 2*Ni*fa/fs 0 350])
%axis([0 0 3 0 100])
title('65536-Point DFT of In-phase Input (it) Ideally Bandlimited to 2*Ni*fa Hz')

```

```

xlabel('Normalized Frequency')
ylabel('Magnitude')
grid

% ACCELERATION
acceleration_out=(dividend/divisor)/(-2*pi*k1);

% IDEAL OUTPUT
a=A*cos(2*pi*fa*((n-nd)/fs));

% ACCELERATION ERROR
error_acceleration=(acceleration_out)-a;
error_max=max(abs(error_acceleration(floor(L/2):1:L)))

figure(2)
subplot(2,1,1)
plot(t,error_acceleration);
axis([0 0 1 (-5*error_max) (5*error_max)])
grid;
title('Acceleration Error')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
subplot(2,1,2)
error_acceleration_dft=fft(error_acceleration(floor(L/4):1:L),L2);
semilogy(N,abs(error_acceleration_dft));
%plot(N,abs(error_acceleration_dft));
%axis([0 (2*N1*fa/fs) 0 350])
%axis([0 0.5 0 600])
title('65536-Point DFT of Acceleration Error')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid

```

A.4 SIM4

```

function [e]=sim4(M,NSB)
%
% SIM4(M,NSB) generates a matrix, e, that displays in its first column
% sideband numbers ranging from 1 to NSB; in its second column the
% positive frequency bandwidth corresponding to the sideband error in
% that row; and in the third column the maximum acceleration error for
% the QCDM demodulator acceleration readout. The errors are maximum
% values corresponding to the worst-case (maximum acceleration error)
% input parameters listed at the end of this description.
%
% When NSB is sufficiently large, the acceleration error converges to
% the maximum acceleration error found using SIM2(M) for differentiator
% order M. Prior to this point of convergence, the FM bandwidth limits
% the theoretical resolution of the demodulator. At the point of
% convergence and thereafter, the differentiator limits the theoretical
% resolution.
%
% The sampling frequency, fs, is 10-kHz. The input signal, i(t), used in
% the example plots is defined as in Fig. 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0
% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)))
%
% These parameters result in the maximum expected modulation index of the
% VIA oscillator outputs.

```

```

%
% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
fs=10e3; % fs=10 kHz

%***** INPUT DEFINITION *****
f=1e3;% Input frequency

% CONTINUOUS TIME AXIS
t=0:1/fs:10*(1/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.

% DISCRETE TIME AXIS
L=10*(1/f)*fs;
n=0:1:L;

% ***** FM INPUT *****
fa=100,% fa=100 Hz Frequency of acceleration,
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
A=20,% A=20 g <- Maximum acceleration, for max mf.
S1=1; % S1=1 unitless, voltage.
S2=1; % S2=1 unitless, voltage

% THESE ARE THE FUNCTIONS I NEED TO APPROXIMATE WITH BESSEL:
% si=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));
% sq=S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs)));

% GENERATE BANDLIMITED si
for Ni=1:NSB% Number of sidebands. Bandwidth is fa*Ni.

Bi=k1*A/fa;
si=S1*besselj(0,Bi)*ones(1,L+1);
for i=1:Ni
    si=si+(2*S1*besselj(2*i,Bi)*cos(2*i*2*pi*fa*(n/fs)));
end;

% GENERATE BANDLIMITED sq
Nq=Ni;% Number of sidebands. Bandwidth if fa*Nq.
Bq=k2*A/fa;
sq=zeros(1,L+1);
for i=0:Nq
    sq=sq+(2*S2*besselj(((2*i)+1),Bq)*sin(((2*i)+1)*2*pi*fa*(n/fs))),
end;

%***** FROM cdm3.m *****

% DIFFERENTIATE
hdiff=remez(M,[0 0 5],[0 1],'differentiator'),% Differentiator
a=1;
b=hdiff;

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq);

% IDEAL DIFFERENTIATED OUTPUT
nd=M/2;% Delay
sidi=(-A*k1*2*pi*(cos(2*pi*fa*((n-nd)/fs))).*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));
sqdi=(A*k2*2*pi*(cos(2*pi*fa*((n-nd)/fs))).*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));

% DELAYED (IDEAL) INPUT
% side=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
% sqde=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% GENERATED DELAYED BANDLIMITED INPUT side *****NOT FROM cdm.m *****
side=S1*besselj(0,Bi)*ones(1,L+1);
for i=1:Ni
    side=side+(2*S1*besselj(2*i,Bi)*cos(2*i*2*pi*fa*((n-nd)/fs)));
end;

```

```

% GENERATED DELAYED BANDLIMITED INPUT sqde *****NOT FROM cdm.m *****
sqde=zeros(1,L+1);
for i=0:Nq
    sqde=sqde+(2*S2*besselj(((2*i)+1),Bq)*sin(((2*i)+1)*2*pi*fa*((n-nd)/fs)));
end;

% SQUARED DELAYED INPUT
sis=side.*side,
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sidi.*sqde)-(sqdi.*side);

% ACCELERATION
acceleration_out=(dividend./divisor)/(-2*pi*k1);

% IDEAL OUTPUT
a=A*cos(2*pi*fa*((n-nd)/fs));

% ACCELERATION ERROR
error_acceleration=(acceleration_out)-a;
error_max(Ni,3)=max(abs(error_acceleration(floor(L/2):1:L)));
error_max(Ni,1)=Ni,
error_max(Ni,2)=Ni*fa*2,

end,

e=error_max

```

A.5 SIM5

```

function [hlpf,Nf]=sim5(Fp,Fs,Dp,Ds)
%
% SIM5(Fp,Fs,Dp,Ds) designs a linear-phase FIR filter using the remez
% algorithm. The filter is designed to meet the passband and stopband
% cutoff frequencies, Fp and Fs (normalized to the sampling frequency),
% and the passband and stopband maximum ripple, Dp and Ds. The function
% returns the filter coefficients in hlpf and the filter order in Nf.
%
% After designing the filter, the function filters an FM input to
% observe error introduced by the passband and stopband ripple, and
% distortion introduced by the bandlimiting of the FM input. The input
% and its spectrum, the filtered input and its spectrum, and the frequency
% response of the filter and its ripple characteristics are plotted.
%
% The sampling frequency, fs, is 10-kHz. The input signal, i(t), used in
% the example plots is defined as in Fig. 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0
% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)))
%
% These parameters result in the maximum expected modulation index of the
% V/A oscillator outputs.
%

```

```

% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
fs=10e3; % fs=10-kHz

%***** INPUT DEFINITION *****
f=1e3;% Input frequency

% CONTINUOUS TIME AXIS
t=0:1/fs.10*(100/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
    % frequency is fN=5e3. This will create 10 periods
    % of input frequency f.

% DISCRETE TIME AXIS
L=10*(100/f)*fs;
n=0:1:L;

% ***** FM INPUT *****
fa=100,% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
A=20;% A=20 g <-- Maximum acceleration, for max mf
S1=1;% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

sio=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));

% LOW PASS FILTER
[Nf,F,M,w]=remezord([Fp Fs], [1 0], [Dp Ds], fs); Nf
[hlpf,err]=remez(Nf,F,M,w), err
al=1;
bl=hlpf;

% FILTER FM INPUTS
si=filter(bl,al,sio);

% ERROR INTRODUCED BY FILTER
sio=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-(Nf/2))/fs)));
error_lpf=sio-si;
max(abs(error_lpf(floor(L/2):1:L)));

figure(1)
subplot(2,1,1)
plot(t,sio)
grid
title('Ideal Input i(t)')
xlabel('Time (Seconds)')
ylabel('Amplitude')
axis([0 0.01 -1 1])
subplot(2,1,2)
L2=2^16;
N=0:1/L2.(1-(1/L2)),
sio_dft=fft(sio,L2);
plot(N,abs(sio_dft))
title('65536-Point DFT of Filter Output')
grid
xlabel('Normalized Frequency')
ylabel('Magnitude')
axis([0 0.5 0 4000])

figure(2)
subplot(3,1,1)
hlpf_dft=fft(hlpf,L2);
plot(N,abs(hlpf_dft))
axis
title('Low Pass Filter hlpf')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid
subplot(3,1,2)
hideal_pass=(1./floor(L2*(Fp/fs)))=1;

```

```

pass_ripple=hideal_pass-abs(hlpf_dft(1:floor(L2*(Fp/fs))));
plot(N(1:floor(L2*(Fp/fs))),pass_ripple)
grid
title('Passband Ripple')
xlabel('Normalized Frequency')
ylabel('Amplitude')
subplot(3,1,3)
plot(N,abs(hlpf_dft))
grid
title('Stopband Ripple')
xlabel('Normalized Frequency')
ylabel('Magnitude')
axis([(Fs/fs) 0.5 (-Ds) (3*Dp)])

figure(3)
subplot(2,1,1)
plot(t,error_lpf);
grid
title('Error in i(t) Introduced by LPF')
xlabel('Time (Seconds)')
ylabel('Amplitude')
axis([0 1 (-3*Dp) (3*Dp)])
subplot(2,1,2)
error_lpf_dft=fft(error_lpf(floor(L/2):L),L/2);
plot(N,abs(error_lpf_dft))
title('65536-Point DFT of Error')
axis([0 0.5 0 (1.5*max(abs(error_lpf_dft)))]))
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid

```

A.6 SIM6

```

funcuon [e,Nf]=sim6(M,Fp,Fs,Dp,Ds)
%
% SIM6(M,Fp,Fs,Dp,Ds) designs an FIR differentiator of length M+1 as in
% SIM1, and a low pass filter with passband and stopband frequencies Fp
% and Fs, respectively, and passband and stopband ripple Dp and Ds, as
% in SIM5. The order of the designed FIR low pass filter is returned in
% Nf. The maximum acceleration error when hlpf is cascaded with the
% CDM block simulated in SIM4, is returned in e. The input to the system
% for this error is listed at the end of this function description.
%
% SIM6 generates a plot of the acceleration error and its spectrum.
%
% The sampling frequency, fs, is 10-kHz The input signal, i(t), used in
% the example plots is defined as in Fig 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0
% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos(k1*A/fa)*sin(2*pi*fa*t)
% q=S1*S2*sin(K1*A/fa)*sin(2*pi*fa*t)
%
% These parameters result in the maximum expected modulation index of the
% VIA oscillator outputs.
%
% (c) 1998 Lawrence K Chang

% SAMPLING FREQUENCY
fs=10e3; % fs=10-kHz

%***** INPUT DEFINITION *****

```

```

f=1e3;% Input frequency

% CONTINUOUS TIME AXIS
t=0.1/fs:10*(100/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
    % frequency is fN=5e3. This will create 10 periods
    % of input frequency f.

% DISCRETE TIME AXIS
L=10*(100/f)*fs;
n=0:1:L;

% ***** FM INPUT *****
fa=100,% fa=100 Hz Frequency of acceleration.
k1=50,% k1=50 Hz/g
k2=50,% k2=50 Hz/g
A=20,% A=20 g <-- Maximum acceleration, for max mf.
S1=1;% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

sio=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));
sqo=S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs)));

% LOW PASS FILTER
[Nf,F,M1,w]=remezord([Fp Fs],[1 0],[Dp Ds],fs); Nf
[hlpf,err]=remez(Nf,F,M1,w), err
al=1;
bl=hlpf;

% FILTER FM INPUTS
si=filter(bl,al,sio);
sq=filter(bl,al,sqo);

% ***** Cross Differentiate Multiply *****
% DIFFERENTIATE
hdiff=remez(M,[0 0.5],[0 1],'differentiator');% Differentiator
a=1;
b=hdiff;

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq);

% IDEAL DIFFERENTIATED OUTPUT
nd=(M+Nf)/2;% *****THIS DELAY IS DIFFERENT FROM sim4.m.
sid_ideal=(-A*k1*2*pi)*(cos(2*pi*fa*((n-nd)/fs)))*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));
sqd_ideal=(A*k2*2*pi)*(cos(2*pi*fa*((n-nd)/fs)))*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs))));

% DELAYED (IDEAL) INPUT
side_ideal=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqde_ideal=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% DELAYED (FILTERED) INPUT
a_delay=1;
b_delay=zeros(1,(M/2)+1);
b_delay(1,(M/2)+1)=1;
side=filter(b_delay,a_delay,si);
sqde=filter(b_delay,a_delay,sq);

% SQUARED DELAYED INPUT
sis=side.*side;
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sid_ideal.*sqde_ideal)-(sqd_ideal.*side_ideal);

% ACCELERATION
acceleration_out=(dividend./divisor)/(-2*pi*k1);

```

```

% IDEAL OUTPUT
a=A*cos(2*pi*fa*((n-nd)/fs));

% ACCELERATION ERROR
error_acceleration=(acceleration_out)-a;
max_error_acceleration=max(abs(error_acceleration(floor(L/2):1:L)))

figure(1)
subplot(2,1,1)
plot(t,error_acceleration);
axis([0 1 (-100*Ds) (100*Ds)])
grid;
title('Acceleration Error')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')

subplot(2,1,2)
L2=2^16;
N=0:1/L2:(1-(1/L2)),
error_acceleration_dft=fft(error_acceleration(floor(L/2):1:L),L2);
semilogy(N,abs(error_acceleration_dft))
title('65536-Point DFT of Acceleration Error')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid

e=max_error_acceleration;

```

A.7 SIM7

```

function [hnbf,error_nbf_max]=sim7(Fp2,Fs2,Dp2,Ds2)
%
% SIM8(Fp2,Fs2,Dp2,Ds2) designs a narrowband filter. This file is very
% similar to SIM5, with the exception that the test input is the
% acceleration signal, and not an FM waveform
%
% (c) 1998 Lawrence K. Chang

%***** DIGITAL SYSTEM PARAMETERS *****
fs=10e3,% fs=10 kHz
t=0 1/fs:0.5;% DISCRETE-TIME AXIS WITH SAMPLING RATE fs
L=0.5*fs,
n=0 1:L,% SAMPLE NUMBER

%***** ACCELERATION INPUT *****
A=20,% Acceleration Magnitude
fa=100;% Acceleration frequency
a=A*cos(2*pi*fa*t);% COSINE ACCELERATION SIGNAL.

%***** LOW PASS FILTER *****
[Nf2,F2,M2,w2]=remezord([Fp2 Fs2], [1 0], [Dp2 Ds2], fs); Nf2
[hnbf,err2]=remez(Nf2,F2,M2,w2);
a12=1;
b12=hnbf;

%***** FILTER DEMODULATED SIGNAL *****
af=filter(b12,a12,a);

%***** ERROR INTRODUCED BY FILTER *****
ai=A*cos(2*pi*fa*(t-(Nf2/(2*fs))));
error_nbf=ai-af;
error_nbf_max=max(abs(error_nbf(floor(L/2):1:L)));

%***** PLOTS *****
figure(1)
plot(t,af,'-',t,ai,'--')

```

```

figure(2)
subplot(3,1,1)
L2=2^16;
N=0.1/L2*(1-(1/L2));
hnbfdft=fft(hnbf,L2);
plot(N,abs(hnbfdft))
title('Low Pass Filter hnbf')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid
subplot(3,1,2)
hideal_pass(1:floor(L2*(Fp2/fs)))=1;
pass_ripple=hideal_pass-abs(hnbfdft(1:floor(L2*(Fp2/fs))));
plot(N(1:floor(L2*(Fp2/fs))),pass_ripple)
grid
title('Passband Ripple')
xlabel('Normalized Frequency')
ylabel('Amplitude')
subplot(3,1,3)
plot(N,abs(hnbfdft))
grid
title('Stopband Ripple')
xlabel('Normalized Frequency')
ylabel('Magnitude')
axis([(Fs2/fs) 0 5 (-Ds2) (3*Ds2)])

figure(3)
subplot(2,1,1)
plot(t,error_nbf);
grid
title('Error in a(t) Introduced by LPF')
xlabel('Time (Seconds)')
ylabel('Amplitude')
axis([0 0.05 (-10*Dp2) (10*Dp2)])
subplot(2,1,2)
error_nbfdft=fft(error_nbf(floor(L2)/1:L),L2);
plot(N,abs(error_nbfdft))
title('65536-Point DFT of Error')
axis([0 0.5 0 (1.5*max(abs(error_nbfdft)))]))
xlabel('Time (Seconds)')
ylabel('Amplitude')
grid

```

A.8 SIM8

```

function [e1,e2]=sim8(M,Fp,Fs,Dp,Ds,Fp2,Fs2,Dp2,Ds2)
%
% SIM8(M,Fp,Fs,Dp,Ds,Fp2,Fs2,Dp2,Ds2) combines the code of SIM6 and SIM7.
% Parameter M is the differentiator order. Parameters Fp, Fs, Dp, and Ds
% are the passband and stopband cutoff frequencies, passband and stopband
% ripple of the low pass filter. Parameters Fp2, Fs2, Dp2, and Ds2 are the
% passband and stopband cutoff frequencies, passband and stopband ripple of
% the narrowband filter.
%
% SIM9 generates three figures. Figure 1 is the acceleration error before
% the narrowband filter. Figure 2 is the error after the narrowband filter.
% Figure 3 is the acceleration output, which should give an indication of
% the SNR.
%
% The next step is to add noise to the input, and to see how well the FM
% demodulator can shape additive white noise that degrades the resolution.
%
% The sampling frequency, fs, is 10-kHz. The input signal, i(t), used in
% the example plots is defined as in Fig. 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0

```

```

% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*t))
% q=S1*S2*sin((K1*A/fa)*sin(2*pi*fa*t))
%
% These parameters result in the maximum expected modulation index of the
% VIA oscillator outputs.
%
% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
fs=10e3; % fs=10-kHz

%***** INPUT DEFINITION *****
f=1e3,% Input frequency

% CONTINUOUS TIME AXIS
t=0:1/fs:10*(100/f);% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.

% DISCRETE TIME AXIS
L=10*(100/f)*fs;
n=0:L;

% ***** FM INPUT *****
fa=100;% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
A=20;% A=20 g <-- Maximum acceleration, for max mf.
S1=1;% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

sio=S1*cos((k1*A/fa)*sin(2*pi*fa*(n/fs)));
sqo=S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs)));

% LOW PASS FILTER
[Nf,F,M1,w]=remezord([Fp Fs], [1 0], [Dp Ds], fs), Nf
[hlpf,err]=remez(Nf,F,M1,w); err
a1=1,
b1=hlpf.

% FILTER FM INPUTS
si=filter(b1,a1,sio);
sq=filter(b1,a1,sqo);

% ***** Cross Differentiate Multiply *****
% DIFFERENTIATE
hdiff=remez(M,[0 0.5],[0 1],'differentiator');% Differentiator
a1=1,
b=hdiff,

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq);

% IDEAL DIFFERENTIATED OUTPUT
nd=(M+Nf)/2,% *****THIS DELAY IS DIFFERENT FROM sim4.m.
sid_ideal=(-A*k1*2*pi)*(cos(2*pi*fa*((n-nd)/fs)))*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqd_ideal=(A*k2*2*pi)*(cos(2*pi*fa*((n-nd)/fs)))*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% DELAYED (IDEAL) INPUT
side_ideal=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqde_ideal=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% DELAYED (FILTERED) INPUT
a_delay=1;
b_delay=zeros(1,(M/2)+1);
b_delay(1,(M/2)+1)=1;
side=filter(b_delay,a_delay,si);

```

```

sqde=filter(b_delay,a_delay,sq);

% SQUARED DELAYED INPUT
sis=side.*side,
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sid_ideal.*sqde_ideal)-(sqd_ideal.*side_ideal);

% ACCELERATION
acceleration_out=(dividend./divisor)/(-2*pi*k1);

% IDEAL OUTPUT
a=A*cos(2*pi*fa*((n-nd)/fs));

% ACCELERATION ERROR
error_acceleration1=(acceleration_out)-a,
max_error_acceleration1=max(abs(error_acceleration1(floor(L/2):1:L)))

% ***** NARROWBAND FILTER *****
[Nf2,F2,M2,w2]=remezord([Fp2 Fs2], [1 0], [Dp2 Ds2], fs); Nf2
[hnbf,err2]=remez(Nf2,F2,M2,w2);
al2=1;
bl2=hnbf,

%***** FILTER DEMODULATED SIGNAL *****
acceleration_out_f=filter(bl2,al2,acceleration_out);

% IDEAL OUTPUT AFTER NARROWBAND FILTERING
a2=A*cos(2*pi*fa*((n-nd-(Nf2/2))/fs));

% ACCELERATION ERROR AFTER NARROWBAND FILTERING
error_acceleration2=(acceleration_out_f)-a2,
max_error_acceleration2=max(abs(error_acceleration2(floor(L/2):1:L)))

figure(1)
subplot(2,1,1)
plot(t,error_acceleration1);
axis([0 1 (-100*D_s) (100*D_s)])
grid;
title('Acceleration Error')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')

subplot(2,1,2)
L2=2^16;
N=0.1/L2:(1-(1/L2));
error_acceleration1_dft=fft(error_acceleration1(floor(L/2):1:L),L2);
semilogy(N,abs(error_acceleration1_dft))
title('65536-Point DFT of Acceleration Error')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid

figure(2)
subplot(2,1,1)
plot(t,error_acceleration2);
axis([0 1 (-100*D_s) (100*D_s)])
grid;
title('Acceleration Error After Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')

subplot(2,1,2)
error_acceleration2_dft=fft(error_acceleration2(floor(L/2):1:L),L2);

```

```

semilogy(N,abs(error_acceleration2_dft))
title('65536-Point DFT of Acceleration Error')
xlabel('Normalized Frequency')
ylabel('Magnitude')
grid

figure(3)
subplot(2,1,1)
plot(t,acceleration_out_f)
title('Acceleration Ouput After Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
axis([0 0.1 -25 25])
grid;

subplot(2,1,2)
acceleration_out_f_dft=fft(acceleration_out_f(floor(L/2):L),L2);
semilogy(N,abs(acceleration_out_f_dft))
title('65536-Point DFT of Acceleration Error')
xlabel('Normalized Frequency')
ylabel('Magnitude')
axis([0 0.04 10e0 10e5])
grid

e1=max_error_acceleration1;
e2=max_error_acceleration2;

```

A.9 SIM9

```

function [error_yacc_max,error_ytrap_max,error_ysimp_max]=sim9(fs,A,fa,input)
%
% SIM9(fs,A,fa,input) examines the performance of three different
% numerical integration methods: the Riemann Sum, the Trapezoidal
% rule, and Simpson's Rule. The discrete-time signal processing
% implementations of each of these methods are accumulation, bilinear
% transformation, and decimation. The decimation is accomplished
% using an analysis filter bank.
%
% SIM7 numerically integrates either a constant input, specified by
% input='con', or a cosine input specified by input='cos'. The
% amplitude and frequency of the inputs must be specified by A and fa.
% The sampling frequency is specified by fs, and time axis ranges from
% zero to 0.5 seconds.
%
% The function returns the maximum amplitude error resulting from each
% of the methods.
%
% (c) 1998 Lawrence K. Chang

%***** DIGITAL SYSTEM PARAMETERS *****
R=2; % DECIMATION FACTOR.
t=0:1/fs:0.5;% DISCRETE-TIME AXIS WITH SAMPLING RATE fs.
t2=0.R/fs:0.5;% DISCRETE-TIME AXIS WITH SAMPLING RATE fs/R.

%***** TEST INPUT AND IDEAL OUTPUT DEFINITION *****
if input=='con'
    a=A+(0*t);% CONSTANT ACCELERATION SIGNAL.
    yideal=A*t;% IDEAL OUTPUT.
    yideal2=A*t2;
else
    a=A*cos(2*pi*fa*t);% COSINE ACCELERATION SIGNAL.
    yideal=(A/(2*pi*fa))*sin(2*pi*fa*t);% IDEAL OUTPUT.
    yideal2=(A/(2*pi*fa))*sin(2*pi*fa*t2);
end

```

```

%***** INTEGRATED OUTPUTS *****
% ACCUMULATOR IMPLEMENTATION OF A RIEMANN SUM USING A FIRST ORDER
% IIR FILTER.
yacc=(1/fs)*filter(1,[1 -1],a);
error_yacc=yideal-yacc;
error_yacc_max=max(abs(error_yacc));
%figure(1)
%subplot(2,1,1)
%plot(t,yacc')
%title('yacc')
%subplot(2,1,2)
%plot(t,error_yacc)
%title('error_yacc')

% BILINEAR TRANSFORM IMPLEMENTATION OF THE TRAPAZOIDAL RULE USING
% A FIRST ORDER IIR FILTER.
ytrap=(1/(2*fs))*filter([1 1],[1 -1],a);
error_ytrap=yideal-ytrap;
error_ytrap_max=max(abs(error_ytrap));
%figure(2)
%subplot(2,1,1)
%plot(t,ytrap)
%title('ytrap')
%subplot(2,1,2)
%plot(t,error_ytrap)
%title('error_ytrap')

% POLYPHASE IMPLEMENTATION OF SIMPSON'S RULE USING A
% DECIMATION.
ae=a(1:2:length(a));% DOWNSAMPLE BY 2 TO GET EVEN SAMPLES.
ad=filter([0 1],[1 1],a);% UNIT SAMPLE DELAY.
ao=ad(1:2:length(a));% DOWNSAMPLE BY 2 TO GET ODD SAMPLES.
aef=filter([1 1],[1 -1],ae);% TRAPAZOIDAL INTEGRATION.
aof=4*filter([1],[1 -1],ao);% RIEMANN SUM.
ysimp=(1/(fs*3))*(aef+aof);% SIMPSON'S RULE.
error_ysimp=yideal2-ysimp;
error_ysimp_max=max(abs(error_ysimp));
%figure(3)
%subplot(2,1,1)
%plot(t2,ysimp)
%title('ysimp')
%subplot(2,1,2)
%plot(t2,error_ysimp)
%title('error_ysimp')

%***** PLOT OF ERRORS *****
%figure(4)
%plot(t,abs(error_yacc),'- ',t,abs(error_ytrap),'-- ',t2,abs(error_ysimp),'- ');
%title('Digital integration error using (-) Accumulation, (-- ) Bilinear Transformation, (-) Decimation')
%xlabel('Time (seconds)')
%ylabel('Velocity (m/s)')
%grid

```

A.10 QCDM

```

function [parameter_names,parameter_values]=qcdm(M,Fp,Fs,Dp,Ds,Fp2,Fs2,Dp2,Ds2,Q,input,A,fa,fs,t)
%
% SYSTEM SIMULATION OF THE QCDM DEMODULATION SCHEME.
%
% M = Differentiator filter order
% Fp = Low pass filter passband cutoff frequency
% Fs = Low pass filter stopband cutoff frequency
% Dp = Low pass filter passband ripple amplitude
% Ds = Low pass filter stopband ripple amplitude
% Fp2= Narrowband filter passband cutoff frequency
% Fs2= Narrowband filter stopband cutoff frequency
% Dp2= Narrowband filter passband ripple amplitude
% Ds2= Narrowband filter stopband ripple amplitude
% Q = Uniform random noise between -Q and Q, with variance Q^2/12

```

```

% input = 'con' specifies constant acceleration input
%   = 'cos' specifies cosine acceleration input
% A = Amplitude of acceleration input (g)
% fa = Frequency of acceleration input (Hz)
% fs = Sampling frequency (Hz)
% t = Duration of simulation (seconds)
%
% The sampling frequency, fs, is 10-kHz. The input signal, i(t), used in
% the example plots is defined as in Fig. 3.7 of this thesis, with the
% following parameter values:
%
% a1(t)=A*cos(2*pi*fa*t)
% a2(t)=0
% k1=50-Hz/g
% A=20-g
% fa=100-Hz
% S1=S2=1
% i=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*t))
% q=S1*S2*sin((K1*A/fa)*sin(2*pi*fa*t))
%
% These parameters result in the maximum expected modulation index of the
% VIA oscillator outputs.
%
% (c) 1998 Lawrence K. Chang

% SAMPLING FREQUENCY
%fs=10e3; % fs=10 kHz

%***** INPUT DEFINITION *****
% DISCRETE TIME AXIS
%L=1e4;% Number of samples in simulation.
L=floor(t*fs);
n=0:1:L;

% CONTINUOUS TIME AXIS
t=0:1/fs:L/fs;% Nyquist rate is fN=fs/2=5e3, so maximum input
% frequency is fN=5e3. This will create 10 periods
% of input frequency f.
t2=0:2/fs:L/fs;% Time axis downsampled by two.

% ***** UNIFORMLY DISTRIBUTED WHITE NOISE GENERATOR *****
noise=unifrnd(-Q,Q,1,length(t));

% ***** FM INPUT *****
%fa=100;% fa=100 Hz Frequency of acceleration.
k1=50;% k1=50 Hz/g
k2=50;% k2=50 Hz/g
%A=20;% A=20 g <-- Maximum acceleration, for max mf.
S1=1,% S1=1 unitless, voltage.
S2=1;% S2=1 unitless, voltage.

if input=='con'
    sio=S1*S2*cos(k1*2*pi*A*(n/fs))+noise;
    sqo=S2*S2*sin(k2*2*pi*A*(n/fs))+noise;
else
    sio=S1*S2*cos((k1*A/fa)*sin(2*pi*fa*(n/fs))+noise);
    sqo=S2*S2*sin((k2*A/fa)*sin(2*pi*fa*(n/fs))+noise);
end

SNR_in=10*log10(var(sio)/var(noise))

% LOW PASS FILTER
[Nf,F,M,w]=remezord([Fp Fs], [1 0], [Dp Ds], fs); Nf
[hlpf,err]=remez(Nf,F,M1,w); err
al=1;
bl=hlpf;

% FILTER FM INPUTS
si=filter(bl,al,sio);
sq=filter(bl,al,sqo);

```

```

% ***** Cross Differentiate Multiply *****
% DIFFERENTIATE
hdiff=remez(M,[0 0.5],[0 1],'differentiator');% Differentiator
a=1;
b=hdiff;

sid=(pi*fs/2)*filter(b,a,si);
sqd=(pi*fs/2)*filter(b,a,sq);

% IDEAL DIFFERENTIATED OUTPUT
nd=(M+Nf)/2;% *****THIS DELAY IS DIFFERENT FROM sim4.m.
sid_ideal=(-A*k1*2*pi*cos(2*pi*fa*((n-nd)/fs)))*(S1*sin((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqd_ideal=(A*k2*2*pi*cos(2*pi*fa*((n-nd)/fs)))*(S2*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% DELAYED (IDEAL) INPUT
side_ideal=S1*cos((k1*A/fa)*sin(2*pi*fa*((n-nd)/fs)));
sqde_ideal=S2*sin((k2*A/fa)*sin(2*pi*fa*((n-nd)/fs)));

% DELAYED (FILTERED) INPUT
a_delay=1;
b_delay=zeros(1,(M/2)+1);
b_delay(1,(M/2)+1)=1;
side=filter(b_delay,a_delay,si);
sqde=filter(b_delay,a_delay,sq);

% SQUARED DELAYED INPUT
sis=side.*side;
sqs=sqde.*sqde;

% AMPLITUDE NORMALIZED TERM FOR AM REJECTION
divisor=(sis+sqs);

% CROSS MULTIPLIED TERMS
dividend=(sid.*sqde)-(sqd.*side);

% IDEAL CROSS MULTIPLIED TERMS
dividendi=(sid_ideal.*sqde_ideal)-(sqd_ideal.*side_ideal);

% ACCELERATION
acceleration_out=(dividend./divisor)/(-2*pi*k1);

% IDEAL OUTPUT
if input=='con'
    a=A+(0*t);
    var_a=A^2
else
    a=A*cos(2*pi*fa*((n-nd)/fs));
    var_a=var(a)
end

% ACCELERATION ERROR
error_acceleration1=(acceleration_out)-a;
max_error_acceleration1=max(abs(error_acceleration1(floor(L/2):1:L)))
var_error1=var(error_acceleration1(floor(L/2):length(error_acceleration1)))

% ***** NARROWBAND FILTER *****
[Nf2,F2,M2,w2]=remezord([Fp2 Fs2], [1 0], [Dp2 Ds2], fs); Nf2
[hbfb,err2]=remez(Nf2,F2,M2,w2);
a12=1;
b12=hbfb;

%***** FILTER DEMODULATED SIGNAL *****
acceleration_out_f=filter(b12,a12,acceleration_out);

% IDEAL OUTPUT AFTER NARROWBAND FILTERING
if input=='con'
    a2=A+(0*t);
else
    a2=A*cos(2*pi*fa*((n-nd)-(Nf2/2))/fs));
end

```

```

% ACCELERATION ERROR AFTER NARROWBAND FILTERING
error_acceleration2=(acceleration_out_f)-a2;
max_error_acceleration2=max(abs(error_acceleration2(floor(L/2):1:L)))
var_error2=var(error_acceleration2(floor(L/2):length(error_acceleration2)))

% ACCELERATION SIGNAL TO NOISE RATIO
SNR_out=10*log10(var_a/var_error2)

startup_transient=isnan(acceleration_out_f);% DETECT NaN VALUES OF VECTOR.
startup_transient=find(startup_transient);% FIGURE HOW MANY.
total_filter_delay=floor(nd+(Nf/2));
%if total_filter_delay<length(startup_transient)
% startup_transient_length=length(startup_transient)
%else
% startup_transient_length=total_filter_delay;
%end;
startup_transient_length=2*total_filter_delay;
acceleration_out_f(1:startup_transient_length)=0;% SET NaN VALUES = 0.

%***** HIGH RESOLUTION INTEGRATION *****
% DOWNSAMPLED IDEAL VELOCITY
if input=='con'
    velocity_ideal=A*t;
    velocity_ideal=velocity_ideal(1:2:length(velocity_ideal));
else
    velocity_ideal=(A/(2*pi*fa))*sin(2*pi*fa*((n-nd-(Nf/2))/fs));
    velocity_ideal=velocity_ideal(1:2:length(velocity_ideal));
end

% POLYPHASE IMPLEMENTATION OF SIMPSON'S RULE USING DECIMATION.
ae=acceleration_out_f(1:2:length(acceleration_out_f));% EVEN SAMPLES
ad=filter([0 1],[1],acceleration_out_f);% UNIT SAMPLE DELAY.
ao=ad(1:2:length(acceleration_out_f));% ODD SAMPLES.
aef=filter([1 1],[1 -1],ae);% TRAPEZOIDAL INTEGR.
aof=4*filter([1],[1 -1],ao);% RIEMANN SUM.

% INITIAL CONDITION FROM IDEAL VELOCITY.
vic=velocity_ideal(floor(startup_transient_length/2)+1)*(ones(1,length(aef))),

velocity=((1/(fs*3))*(aef+aof))+vic;% SIMPSON'S RULE.

% VELOCITY ERROR
error_velocity=velocity_ideal-velocity;
error_velocity_max=max(abs(error_velocity(floor(L/4):length(velocity))))
var_error3=var(error_velocity);

% VELOCITY SIGNAL TO NOISE RATIO
SNR_vel=10*log10(var(velocity)/var_error3)

%***** SIMULATION OUTPUT PARAMETERS *****

parameter_values=[SNR_in;Nf;var_a;max_error_acceleration1;var_error1,Nf2;max_error_acceleration2;var_error2;SNR_out;error_velocity_max;
var_error3,SNR_vel];
parameter_names=['Input SNR    ';
'LPF Order    ';
'Acc Signal Power';
'Acc Error Var1  ';
'Max Acc Error1  ';
'NBF Order    ';
'Max Acc Error2  ';
'Acc Error Var2  ';
'Acc Output SNR  ';
'Max Vel Error  ';
'Vel Error Var  ';
'Vel Output SNR  '];
output=[parameter_names, parameter_values];

%***** PLOTS *****
L2=2^16,
N=0.1/L2.(1-(1/L2));
p=1e3;% NUMBER OF POINTS IN PLOTS

```

```

figure(8)
subplot(2,1,1)
%plot(t2(floor(L/2)-p:floor(L/2)),velocity((L/2)-p:floor(L/2)),'-',t2(floor(L/2)-p:floor(L/2)),velocity_ideal(floor(L/2)-p:floor(L/2)),'--')
plot(t2,velocity,'-',t2,velocity_ideal,'--')
title('(-) Velocity Output, Ideal Velocity Output (-)')
xlabel('Time (seconds)')
ylabel('Velocity (m/s)')
grid
%axis([0.1 0.2 -0.04 0 05])
subplot(2,1,2)
%plot(t2(floor(L/2)-p:floor(L/2)), error_velocity(floor(L/2)-p floor(L/2))
plot(t2,error_velocity)
title('Velocity Error')
xlabel('Time (seconds)')
ylabel('Velocity (m/s)')
grid
axis([0 10 (-2*error_velocity_max) (2*error_velocity_max)])

figure(5)
subplot(2,1,1)
plot(t(L-p:L),acceleration_out(L-p:L))
title('Acceleration Output Before Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleraiton (g)')
grid
%axis([0 0.1 -30 30])
subplot(2,1,2)
%acceleration_out_dft=fft(acceleration_out(floor(L/2):length(acceleration_out)),L2);
%plot(N,20*log10(abs(acceleration_out_dft)))
%xlabel('Normalized Frequency')
%ylabel('Magnitude (dB)')
%grid
plot(t(L-p:L),error_acceleration1(L-p:L));
%axis([0 0.1 (-2*max_error_acceleration1) (2*max_error_acceleration1)])
grid;
title('Acceleration Error Before Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')

figure(3)
subplot(2,1,1)
plot(t(L-p:L),acceleration_out_f(L-p:L))
title('Acceleration Ouput After Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')
%axis([0 0.3 -25 25])
grid;
subplot(2,1,2)
%acceleration_out_f_dft=fft(acceleration_out_f(floor(L/2):1:L),L2);
%plot(N,20*log10(abs(acceleration_out_f_dft)))
%title('65536-Point DFT of Acceleration Output After Narrowband Filtering')
%xlabel('Normalized Frequency')
%ylabel('Magnitude (dB)')
%axis([0 0.1 10e0 10e5])
%axis([0 0.3 0 100])
%grid
plot(t(L-p:L),error_acceleration2(L-p:L));
%axis([0 0.3 (-2*max_error_acceleration2) (2*max_error_acceleration2)])
grid;
title('Acceleration Error After Narrowband Filtering')
xlabel('Time (Seconds)')
ylabel('Acceleration (g)')

%figure(4)
%seo_dft=fft(sio,L2),
%subplot(2,1,1)
%plot(N,20*log10(abs(sio_dft)))
%grid
%xlabel('Normalized Frequency')
%ylabel('Magnitude (dB)')
%title('Inphase FM Input Corrupted with Noise')

```

```
%subplot(2,1,2)
%acceleration_out_dft=fft(acceleration_out(floor(L/2):length(acceleration_out)),L/2);
%plot(N,20*log10(abs(acceleration_out_dft)))
%grid
%title('Acceleration Output Before Narrowband Filtering')
%xlabel('Normalized Frequency')
%ylabel('Magnitude (dB)')
```


References

- [1] M.E. Ash, M.F. Luniewicz, P.C.L. Leung, and M.A. Bates, "Coherent High Speed Frequency Measurement and Signal Processing" *International Conference on Signal Processing Applications and Technology*. Cambridge, MA, Nov. 1992.
- [2] F.J. Leibly III., "Measurement and Digitization of Phase Angle Modulated Sensor Data." *Master of Science Thesis, MIT*. Cambridge, MA. Sept. 1994.
- [3] P. Young. Electronic Communication Techniques. MacMillan Publishing Company. New York, 1994.
- [4] L.W. Couch II. Digital and Analog Communication Systems. Macmillan Publishing Company. New York, 1990.
- [5] A.V. Oppenheim and R.W. Schaffer. Discrete-Time Signal Processing. Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [6] J. Garodnick, J. Greco, and D. Schilling, "Response of an All Digital Phase-Locked Loop," *IEEE Transactions on Communication Technology*, vol. COM-22, pp. 751-764. June 1974.
- [7] M. Hagiwara and M. Nakagawa, "Digital Signal Processing Type Stereo FM Receiver," *IEEE Transactions on Consumer Electronics*, vol. CE-32, pp. 37-43, Feb. 1986.
- [8] N. Boutin, "An Arctangent type Wideband PM/FM Demodulator with Improved Performance," *IEEE Transactions on Consumer Electronics*, vol. CE-38, pp. 5-9, Feb. 1992.
- [9] R. Beards and M. Copeland, "An Oversampling Delta-Sigma Frequency Discriminator," *IEEE Transactions on Circuits and Systems II*, vol. 41, pp. 26-32, Jan. 1994.
- [10] M. Hovin, A. Olsen, T.S. Lande, and C. Toumazou, "Delta-Sigma using Frequency Modulated Intermediate Values," *IEEE Journal of Solid State Circuits*, vol. 32, No. 1, Jan. 1997.
- [11] M. Hovin, T. Saether, D.T. Wisland, T.S. Lande, "A Narrow-band Delta-Sigma Frequency-to-Digital Converter," *Proceedings of 1997 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 77-80.
- [12] B.S. Song and I.S. Lee, "A Digital FM Demodulator for FM, TV, and Wireless," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal*

Processing, vol. 42, No. 12, Dec. 1995.

- [13] H.M. Kwon and K.B. Lee, "A Novel Digital FM Receiver for Mobile and Personal Communications," *IEEE Transactions on Communications*, vol. 44, No. 11, Nov. 1996.
- [14] L.R. Rabiner and B. Gold. Theory and Application of Digital Signal Processing. Prentice-Hall, Inc. Englewood Cliffs, New Jersey: 1975.
- [15] M.W. Hauser. "Principles of Oversampling A/D Conversion," *J. Audio Eng. Soc.*, vol. 39, No. 1/2, Jan./Feb. 1991.
- [16] J.C. Candy, "Decimation for Sigma Delta Modulation," *IEEE Transactions on Communications*, vol. COM-34, No. 1, Jan. 1986.
- [17] Transmission Systems for Communications. Bell Telephone Laboratories, 4th Ed., 1970.
- [18] R.W. Hornbeck. Numerical Methods. Prentice-Hall, Inc. Englewood Cliffs, New Jersey: 1975.
- [19] P.P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 78, No. 1, Jan. 1990.
- [20] The American Heritage. Houghton Mifflin Company, Boston, MA, 1983.
- [21] N. Barbour, J. Elwell, and R. Setterland, "Inertial Instruments: Where to now?" AIAA-92-4414-CP.
- [22] J.R. Vig. Quartz Crystal Resonators and Oscillators for Frequency Control and Timing Applications. U.S. Army Electronics Technology and Devices, Jan. 1990.
- [23] E.A. Lee and D.G. Messerschmitt. Digital Communications. 2nd ed. Klumer Academic Publishers: 1994.
- [24] A.L. Coban and P.E. Allen, "A New Fourth-Order Single-Loop Delta-Sigma Modulator for Audio Applications," *1996 IEEE International Symposium on Circuits and Systems*. vol. 1, p. 461-4.
- [25] M.K. Kinyua and K. S. Chao, "High-Resolution Multi-Bit Sigma-Delta Modulator Architecture," *Proceedings of the Midwest Symposium on Circuits and Systems Dedicated to the Memory of Professor Mac Van Valkenburg*, vol. 1, p. 304-7.
- [26] N.Tan and S. Eriksson, "Fourth-Order Two-Stage Delta-Sigma Modulator Suing Both 1-bit and Multibit Quantisers," *Electronic Letters*, vol. 29, no. 11, p. 937-8.

- [27] G. Fischer and A.J. Davis, "Wideband Cascade Delta-Sigma Modulator with Digital Correction for Finite Amplifier Gain Effects," *Electronic Letters*, vol. 34, no. 6, p. 511-12.
- [28] S. Brigati, F. Francesconi, V. Liberali, F. Maloberti, P. Malcovati, and M. Poletti, "Design Considerations on Very High Resolution Sigma-Delta Modulators," *Proceedings of the Midwest Symposium on Circuits and Systems Dedicated to the Memory of Professor Mac Van Valkenburg*, vol. 1, p. 304-7.
- [29] E.B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. ASSP-29, No. 2, April 1981.
- [30] K. Kaiser, *memo*, Draper Laboratory, July 1996.