

WIRELESS SENSOR INTERROGATOR DESIGN FOR
PASSIVE, RESONANT FREQUENCY SENSORS USING
FREQUENCY MODULATION SPECTROSCOPY

by

Brian James Peterson

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2009

©COPYRIGHT

by

Brian James Peterson

2009

All Rights Reserved

APPROVAL

of a thesis submitted by

Brian James Peterson

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citation, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. Todd J. Kaiser

Approved for the Department of Electrical and Computer Engineering

Dr. Robert C. Maher

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Brian James Peterson

April 2009

ACKNOWLEDGEMENTS

I am extremely grateful to everyone who helped make this project a success. First I would like to thank my graduate chair, major professor, and academic advisor, Dr. Todd J. Kaiser, for his assistance and the opportunity to work on this project. In addition, I would like to thank the other members of my graduate committee, Dr. Jim Becker and Prof. Andy Olson. I would also like to thank my fellow graduate students for their discussions and constructive comments. Finally, I would like to acknowledge the financial support of the National Science Foundation (NSF) under grant #0428571, without which this research would not have been possible.

TABLE OF CONTENTS

1. INTRODUCTION TO MEMS SENSORS.....	1
Introduction	1
MEMS Resonant Frequency Sensors	4
Statement of the Problem	6
Thesis Overview	7
2. FREQUENCY MODULATION SPECTROSCOPY	8
Introduction to Frequency Modulation Spectroscopy	8
Frequency Modulation Spectroscopy Theory	9
3. WIRELESS SENSOR INTERROGATOR DESIGN.....	19
Introduction	19
Wireless Sensor Interrogator Design Detail.....	20
Transmitted Signal Generation.....	21
Signal Generator	22
DDS IC.....	22
Frequency Modulator	24
Microcontroller	28
Received signal processing	28
Automatic Gain Control.....	30
Square-law IC	31
Bandpass Filter	32
Amplifier	33
Signal Generator	34
Phase Shifter	34
Mixer	35
Lowpass Filter	36
Feedback Signal Generation.....	37
System control/closing the loop	43
Personal Computer.....	43
4. WIRELESS SENSOR INTERROGATOR DESIGN RESULTS	45
Introduction	45
MATLAB Simulink Simulation	46
Differences Between the Wireless Sensor Interrogator Design and the MATLAB Simulink Model	47
MATLAB Simulink Simulation Results.....	48
Simulation Results Without the Sensor Model and Without the Feedback Signals Employed.....	48
Simulation Results With the Sensor Model and Without the Feedback Signals Employed.....	49

TABLE OF CONTENTS - CONTINUED

Simulation Results Without the Sensor Model and With the Feedback Signals Employed.....	51
Simulation Results With the Sensor Model and With the Feedback Signals Employed.....	52
Wireless Interrogator Design Hardware Implementation	53
Hardware Implementation Results With a Hard-wired LC Resonant Sensor	54
Hardware Results Without a Hard-wired Sensor and Without the Feedback Signals Employed.....	54
Hardware Results With a Hard-wired Sensor and Without the Feedback Signals Employed.....	55
Hardware Results Without a Hard-wired Sensor and With the Feedback Signals Employed.....	57
Hardware Results With a Hard-wired Sensor and With the Feedback Signals Employed.....	58
Hardware Implementation Results With a Wireless LC Resonant Sensor.....	59
Hardware Results Without a Wireless Sensor and Without the Feedback Signals Employed.....	60
Hardware Results With a Wireless Sensor and Without the Feedback Signals Employed.....	61
Hardware Results Without a Wireless Sensor and With the Feedback Signals Employed.....	62
Hardware Results With a Wireless Sensor and With the Feedback Signals Employed.....	64
Additional Wireless Interrogator Results.....	65
Design Improvements	68
Antenna Modeling and Design.....	68
The Digital Divide	69
Conclusions	70
5. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH	71
Summary	71
Recommendations for Future Research.....	72
REFERENCES CITED	74
APPENDICES.....	77
APPENDIX A: Derivation of the Absorption and Dispersion Output Signals Using Frequency Modulation Spectroscopy.....	78

TABLE OF CONTENTS – CONTINUED

APPENDIX B: Microcontroller Source Files	89
APPENDIX C: MATLAB GUI Source File	101

LIST OF TABLES

Table		Page
2.1	The relative amplitudes of the frequency components in a frequency modulated signal with a frequency modulation index of 1.08.....	12

LIST OF FIGURES

Figure		Page
2.1	Typical experimental setup for frequency modulation spectroscopy.....	9
2.2	Graphical determination of the optimal frequency modulation index, β	11
2.3	Sequential frequency domain plots of FM signal resulting from a linear frequency ramp of the carrier	13
2.4	Transfer function of a passive resonant frequency sensor	14
2.5	Sequential frequency domain plots showing the effect of the passive resonant frequency sensor on the ramped FM signal.....	14
2.6	Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=10$	17
2.7	Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=20$	17
2.8	Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=50$	17
2.9	Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=100$	18
3.1	Typical experimental setup for frequency modulation spectroscopy.....	19
3.2	Block diagram of the wireless sensor interrogator design.....	20
3.3	Block diagram of transmitted signal generation	21
3.4	Picture of signal generator	22
3.5	PCB assembly with AD9851 DDS IC	23
3.6	Basic elements of a PLL	24
3.7	PLL with sinusoid added to the VCO control voltage	26

LIST OF FIGURES - CONTINUED

Figure		Page
3.8	Physical implementation of the loop filter.....	27
3.9	Block diagram of received signal processing	29
3.10	Picture of AD8367 evaluation board from Analog Devices, Inc.....	31
3.11	Functional block diagram of AD835 analog multiplier IC	32
3.12	Block diagram of bandpass filter (BPF)	32
3.13	Schematic of bandpass filter (BPF) circuit	33
3.14	Schematic of amplifier circuit.....	34
3.15	Schematic of phase shifter circuit	35
3.16	Pictures of the double balanced mixer.....	36
3.17	Schematic of lowpass filter (LPF) circuit.....	37
3.18	Output dispersion signal showing detection levels for generating feedback signals.....	38
3.19	State diagram of the finite state machine used to generate the LOCK feedback signal.....	39
3.20	Schematic of the finite state machine used to generate the LOCK feedback signal.....	40
3.21	Schematic of the INC or DEC feedback signal generation circuitry	42
3.22	MATLAB graphical user interface (GUI) used to control the wireless sensor interrogator	44
4.1	Block diagram of the wireless sensor interrogator design.....	46
4.2	MATLAB Simulink model of the wireless sensor interrogator	47

LIST OF FIGURES - CONTINUED

Figure		Page
4.3	Carrier frequency data of the transmitted FM signal without the sensor model and without the feedback signals employed	49
4.4	Output signal without the sensor model and without the feedback signals employed showing the absence of the dispersion signal	49
4.5	Carrier frequency data of the transmitted FM signal with the sensor model and without the feedback signals employed	50
4.6	Output signal with the sensor model and without the feedback signals employed showing the presence of the dispersion signal	50
4.7	Carrier frequency data of the transmitted FM signal without the sensor model and with the feedback signals employed	51
4.8	Output signal without the sensor model and with the feedback signals employed showing the absence of the dispersion signal	52
4.9	Carrier frequency data of the transmitted FM signal with the sensor model and with the feedback signals employed	53
4.10	Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the sensor model	53
4.11	Carrier frequency data of the transmitted FM signal without the hard-wired sensor and without the feedback signals employed	55
4.12	Output signal without the hard-wired sensor and without the feedback signals employed showing the absence of the dispersion signal	55
4.13	Carrier frequency data of the transmitted FM signal with the hard-wired sensor and without the feedback signals employed	56
4.14	Output signal with the hard-wired sensor and without the feedback signals employed showing the presence of the dispersion signal	56
4.15	Carrier frequency data of the transmitted FM signal without the hard-wired sensor and with the feedback signals employed	57

LIST OF FIGURES - CONTINUED

Figure		Page
4.16	Output signal without the hard-wired sensor and with the feedback signals employed showing the absence of the dispersion signal	58
4.17	Carrier frequency data of the transmitted FM signal with the hard-wired sensor and with the feedback signals employed	59
4.18	Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the hard-wired sensor	59
4.19	Carrier frequency data of the transmitted FM signal without the wireless sensor and without the feedback signals employed	60
4.20	Output signal without the wireless sensor and without the feedback signals employed showing the absence of the dispersion signal	61
4.21	Carrier frequency data of the transmitted FM signal with the wireless sensor and without the feedback signals employed	62
4.22	Output signal with the wireless sensor and without the feedback signals employed showing the presence of the dispersion signal	62
4.23	Carrier frequency data of the transmitted FM signal without the wireless sensor and with the feedback signals employed	63
4.24	Output signal without the wireless sensor and with the feedback signals employed showing the absence of the dispersion signal	63
4.25	Carrier frequency data of the transmitted FM signal with the wireless sensor and with the feedback signals employed	64
4.26	Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the wireless sensor	65
4.27	Carrier frequency data from a simple test of the scan functionality of the wireless sensor interrogator	66

LIST OF FIGURES - CONTINUED

Figure		Page
4.28	The output signal showing the partial dispersion signals as the wireless interrogator locks to each of two sensors	66
4.29	Carrier frequency data from a second test of the scan functionality of the wireless sensor interrogator	67

ABSTRACT

The lack of passive, wireless, chemical and biological sensor systems is a significant impediment to sensor system applications. While active sensors with a wireless communications link continue to decrease in power consumption, they still require a power source, such as a battery. This active power consumption limits the useful life of the sensor and its applications. A more attractive solution would be a passive, wireless, chemical and biological sensor integrated with a wireless interrogation platform to monitor the sensor. The focus of this thesis is the realization of a wireless sensor interrogator capable of monitoring multiple, passive, resonant-frequency sensors. It is demonstrated, using Frequency Modulation Spectroscopy techniques, that the resonant frequency of a passive sensor can be detected and tracked over time. Simulated results are presented that verify the functionality of the proposed wireless sensor interrogator. In addition, an experimental hardware setup and subsequent experimental results are presented that verify the simulation results. Considerations for the design of the wireless sensor interrogator and opportunities for future research are discussed.

CHAPTER ONE

INTRODUCTION TO MEMS SENSORS

Introduction

The dominance of CMOS technology in the microfabrication of integrated circuits can be attributed to its reliability, high density, low cost, and high performance. The silicon often used as the starting material for integrated circuits has also made the microfabrication of mechanical structures feasible. Microelectromechanical Systems (MEMS) incorporate both electrical and mechanical components into their design. MEMS devices enable both actuation and sensing via mechanical means. MEMS sensors have been developed that are capable of sensing temperature, pressure, acceleration, rotation, stress, chemicals, biological agents, and other physical parameters [1-5].

The ability to monitor chemical and biological materials has become even more important after the terrorist attacks on 9/11. By applying a chemical or biological receptor material to the surface of a MEMS structure, such as a cantilever beam or a thin membrane, changes in the resonant frequency of the structure can be detected. The change in the resonant frequency is due to either the change in mass or the change in stress on the structure resulting from the attached chemical or biological agents [6].

Active wireless sensors capable of monitoring chemical and biological agents require additional electronics to detect the resonant frequency and transmit this information over a wireless communications link. This transmitted information, or transmitted signal, must be detected by a reader. The reader must be able to detect the

signal transmitted from the sensor above the noise in the wireless communications link. Devices requiring additional electronics to be combined with the sensor require a power source and are usually battery operated [5]. These active wireless sensors use the power from the battery to transmit the resonant frequency information to the reader. The strong signals from active wireless sensors can be transmitted over long distances before becoming too weak to be detected by the reader. However, because batteries contain a finite amount of energy and have a limited shelf life, active wireless sensors may not work continuously for a long time. Therefore, for long term reliability, it is desirable to have a passive sensor and thus eliminate the need for a battery. A passive sensor, however, is limited to working with the power from signals transmitted to the passive sensor from the reader. Therefore, the transmitted signal from a passive wireless sensor is often weaker than the transmitted signal from an active wireless sensor. For that reason, the sensitivity of the sensor detection scheme must be increased in order to get any appreciable range or distance between the reader or wireless interrogator and the passive wireless sensor.

To monitor chemical and biological agents, a wireless sensing system capable of detecting and tracking the resonant frequency of a passive sensor is proposed. The passive sensor consists of an electrical resonance coupled to one or more mechanical resonances. The electrical resonance consists of an inductor coil in parallel with a capacitor. The mechanical resonances originate from thin membranes designed to resonant at the same frequency, or close to the same frequency, as the electrical resonance. A piezoelectric material couples the mechanical resonances to the electrical

resonance. The thin membranes of the mechanical resonators can be coated with a chemical or biological receptor material. Specific chemical or biological agents can adsorb to the material, and thus alter the physical properties of the resonator. The mechanical resonant frequency of the device changes with the addition of adsorbed chemical or biological substances. It is this change in resonant frequency that we wish to detect, which indicates the presence of adsorbed substances. The magnitude of resonant frequency change might, in the future, be correlated with the concentration of adsorbed substance.

It should be noted that it is not the goal of this research to develop the sensor itself. The sensor is being designed, fabricated and analyzed by a related but separate research project at the University of Alabama, Huntsville. The scope of this research is to detect and track the resonant frequency of the sensor. Subsequently, we will now turn our attention to the sensing technique employed.

Since passive wireless sensors produce inherently weak signals that must be tracked in order to detect the presence of the chemical or biological substance, the sensitivity of the detection technique is important. Sensitivity limits for the detection of these weak signals depend on the power of the transmitted signal, the energy absorption of the sensor, the sensitivity of the receiver, and the noise in the system. Most environmental noise sources (interference) can be mitigated by using modulation techniques. Frequency modulation spectroscopy (FMS) is a highly sensitive measurement technique that has been employed in optical measurement experiments for

many years [7-12]. FMS is explained in detail in chapter two. The following section develops the background of MEMS resonant frequency sensors.

MEMS Resonant Frequency Sensors

Many types of physical structures can be used for resonant frequency sensing. The key figures of merit for mechanical resonators are the resonant frequency (f_r) and the quality factor (Q). The quality factor is a dimensionless parameter that compares the frequency at which a system oscillates to the rate at which it dissipates energy. The lower the energy loss per cycle of oscillation, the greater the quality factor is. At or near the resonant frequency, the energy loss decreases and the mechanical vibration amplitude sharply increases. The sharpness of this resonant peak is proportional to the quality factor. In sensing applications, the MEMS structure is driven at this resonant peak, thereby increasing the energy efficiency and sensitivity of the sensor. Sensors with higher quality factors are more efficient with the power captured from the signals sent by the reader. In addition, a sensor with a higher quality factor uses less bandwidth in the frequency spectrum, so multiple sensors can have resonant frequencies that are close to one another without interfering with each other. However, if the quality factor is too high, the reader may not be able to detect the sensor if the frequency step size of the reader is larger than the sensors bandwidth.

The quality factor can be defined in several ways. From the energy point of view, it is the ratio of the total stored energy in a system over the energy lost over each cycle of oscillation. Mathematically, the quality factor is related to the half power bandwidth

(Δf). The ratio of the resonant frequency (f_r) and the bandwidth at half power gives the quality factor [13]:

$$Q = \frac{f_r}{\Delta f} \quad [1.1].$$

The value of the mechanical resonant frequency depends on the physical dimensions and the properties of the material that the mechanical resonator is made from. In the case of a mass m attached to a spring with a force constant, k , the resonant frequency is given by

$$f_r = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad [1.2].$$

The appropriate equations to use for k and m depend on the boundary and loading conditions of the mechanical resonator. The resonant frequency given in equation 1.2 represents the first mode of vibration or the fundamental mode of vibration. Both cantilever beams and thin membranes can have multiple frequencies at which they will resonate. These frequencies correspond to different modes of vibration of the physical structure. In most cases, the first harmonic or the natural frequency is the one of most interest [13].

The most common structure used as a mechanical resonator is a cantilever beam. For chemical and biological sensing, a chemical or biological receptor material is applied to the surface of the cantilever beam. The cantilever beam is driven into resonance, and this resonant frequency is monitored by an electrical circuit. The mass of the cantilever beam is increased by the adsorbed chemical or biological material, thereby decreasing the resonant frequency of the cantilever beam. Alternatively, a sometimes competing effect

results from the changes in surface stress of the mechanical structure, which can also change the resonant frequency. Depending on the type and amount of material adsorbed onto the mechanical structure, either the change in mass or the change in mechanical stress will dominate the change in the resonant frequency.

Thin membranes can also be used as mechanical resonators. Much like the surface of a drum, the thin membrane can be driven into resonance. By coating the surface of the thin membrane with a chemical or biological receptor material, similar detection can be achieved as with the cantilever beams. Many different physical geometries may be used for a thin membrane including circular, square, or rectangular. The same equations for the resonant frequency and the quality factor can be used. Again, the equations for the mass m and the spring constant k will depend on the physical properties of the material that the thin membrane is made out of, the physical dimensions of the device, the boundary conditions, and loading conditions.

Statement of the Problem

The purpose of this research is to design, build, and test a wireless sensor interrogator capable of detecting and tracking the resonant frequency of a passive, wireless, resonant frequency sensor. The wireless interrogator will lock to and track the resonant frequency of the sensor over time.

Thesis Overview

In order to achieve a high sensitivity of the sensor detection technique, and therefore obtain adequate range of operation, frequency modulation spectroscopy was chosen as the sensing technique in this design. A detailed description of frequency modulation spectroscopy is presented in Chapter two. The design of the wireless sensor interrogator and the hardware components selected for the experimental setup are presented in Chapter three. Chapter four discusses the results of both the simulation and experimental setup of the wireless sensor interrogator. Included in Chapter four is a brief discussion of suggested improvements for the wireless sensor interrogator design. Concluding remarks and suggestions for future work are commented upon in Chapter five.

CHAPTER TWO

FREQUENCY MODULATION SPECTROSCOPY

Introduction to Frequency Modulation Spectroscopy

The resonant frequency of a passive sensor can be measured using several different methods. One of the most basic methods used in experimental testing is based on direct power absorption of a single frequency signal. While this method works if the approximate resonant frequency is known a priori and if the sensor is in close proximity of the transmitting and receiving antennas, it is inefficient for longer distances and if the approximate resonant frequency is not already known. A sensing technique employed in optical systems called frequency modulation spectroscopy overcomes these limitations. Although this technique requires complex signal generation and control, the benefits in sensitivity and frequency resolution make using it worth while.

The wireless interrogator transmits a radio frequency signal that is inductively coupled with the electrical resonance of the sensor. The sensor will resonant, both electrically and mechanically, when the transmitted signal contains a frequency component at or near the resonant frequency of the sensor. The sensor affects both the magnitude and the phase of the frequency components at or near its resonant frequency. The effect on the magnitude of the frequency component is a frequency dependent attenuation, and is called absorption. The effect on the phase of the frequency component is a frequency dependent phase shift, and is called dispersion. Frequency modulation spectroscopy is the measurement technique used in this design because of its high

sensitivity in detecting weak absorptions and dispersions, like those associated with the resonant frequency of a passive sensor.

Frequency modulation spectroscopy was originally developed by Gary C. Bjorklund [7] using lasers; however, in this thesis it will be applied to Radio Frequencies (RF). The next section will examine frequency modulation spectroscopy, describing the theory of the technique in detail.

Frequency Modulation Spectroscopy Theory

Frequency modulation spectroscopy (FMS) is a sensitive measurement technique based on the changes in amplitude and phase of the frequency components of a frequency modulated (FM) signal. The wireless sensing system or wireless interrogator consists of the following functions: signal generation, wireless transmission to the sensor, receiving the signal modified by the sensor, received signal processing, and system control. A typical experimental setup for FMS is shown in Figure 2.1.

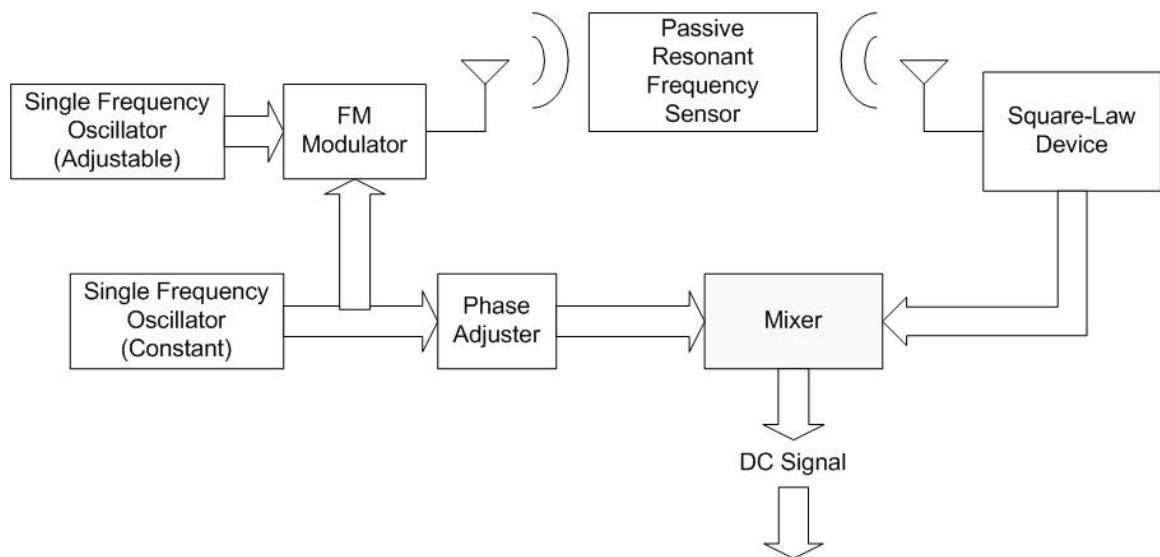


Figure 2.1: Typical experimental setup for frequency modulation spectroscopy.

The adjustable frequency oscillator in Figure 2.1 has the sinusoidal output

$$c(t) = A_c \cos(2\pi f_c t) \quad [2.1]$$

where f_c is adjustable. The constant frequency oscillator in Figure 2.1 has the sinusoidal output

$$m(t) = a \cos(2\pi f_m t) \quad [2.2]$$

where f_m is constant. Both of these signals are input to the FM modulator block in Figure 2.1 to form the transmitted frequency modulated (FM) signal

$$s(t) = A_c \cos(2\pi f_c t + 2\pi k_f \int_{-\infty}^t a \cos(2\pi f_m \tau) d\tau) \quad [2.3]$$

where k_f is the frequency deviation constant. Completing the integration

$$s(t) = A_c \cos(2\pi f_c t + \frac{k_f a}{f_m} \sin(2\pi f_m t)) \quad [2.4]$$

From communication systems theory [14], the constant in front of the *sin* term is easily recognized as the frequency modulation index β .

$$\beta = \frac{k_f a}{f_m} \quad [2.5]$$

Combining equations 2.4 and 2.5 we have

$$s(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t)) \quad [2.6]$$

Using Euler's relation, taking the Fourier series, and recognizing the Bessel function of the first kind, which is an involved process presented in appendix A, equation 2.6 can be written as

$$s(t) = \sum_{n=-\infty}^{\infty} A_c J_n(\beta) \exp[j(2\pi f_c + n 2\pi f_m)t] \quad [2.7]$$

where $J_n(\beta)$ is the Bessel function of the first kind of order n . Appendix A also shows a mathematically complex derivation of frequency modulation spectroscopy. The amplitude of the frequency components of the FM signal $s(t)$ vary with the frequency modulation index β . The transmitted FM signal $s(t)$ contains all frequencies of the form $f_c + n \cdot f_m$ for $n=0, \pm 1, \pm 2, \dots$. It will be shown that the product of the carrier and first sidebands play a significant role in the detection of the passive, resonant frequency sensor. Therefore, the frequency modulation index β is chosen to maximize the product of the carrier and first sideband, or the J_0 and J_1 terms. The product of the J_0 and J_1 terms versus the frequency modulation index β is shown in Figure 2.2.

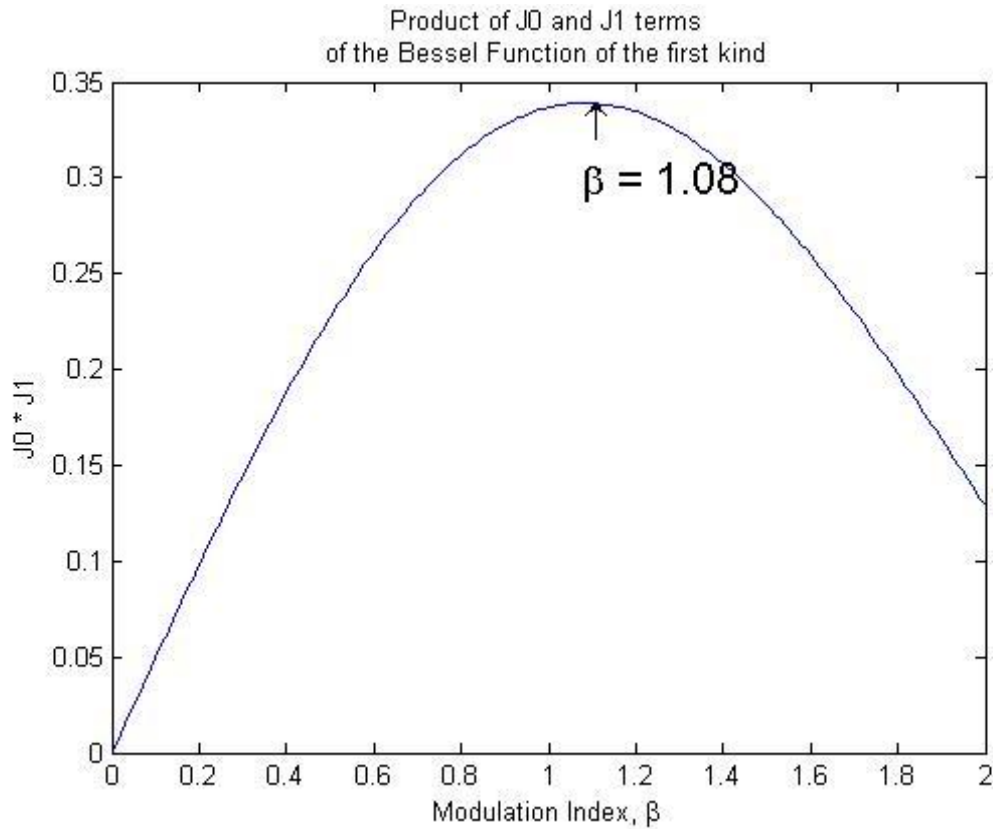


Figure 2.2: Graphical determination of the optimal frequency modulation index, β .

From Figure 2.2, the optimal frequency modulation index β is 1.08. Equation 2.8 gives the relative amplitude in dB of each frequency component of the FM signal, J_n , to the carrier, J_0 . Table 2.1 shows the values of Equation 2.8 for all of the frequency components of the FM signal.

$$20 \cdot \log_{10} \left(\frac{J_n(1.08)}{J_0(1.08)} \right) \quad [2.8]$$

Frequency component, J_n	Amplitude relative to the carrier, J_0 , in dB
J_0	0 dB
J_1	-3.9 dB
J_2	-14.8 dB
$J_m, m \geq 3$	≤ -29.5 dB

Table 2.1: The relative amplitudes of the frequency components in a frequency modulated signal with a frequency modulation index of 1.08.

The FM signal consists of a strong carrier and two weak sidebands, therefore only the $n = -1, 0, 1$ terms corresponding to the lower sideband, the carrier, and the upper sideband need to be considered. All other frequency components are insignificant as shown by Table 2.1.

The FM signal is then wirelessly transmitted while linearly ramping the carrier frequency, f_c , of the adjustable frequency oscillator producing the frequency domain signal shown in the plots in Figure 2.3. These plots resulted from a MATLAB based simulation.

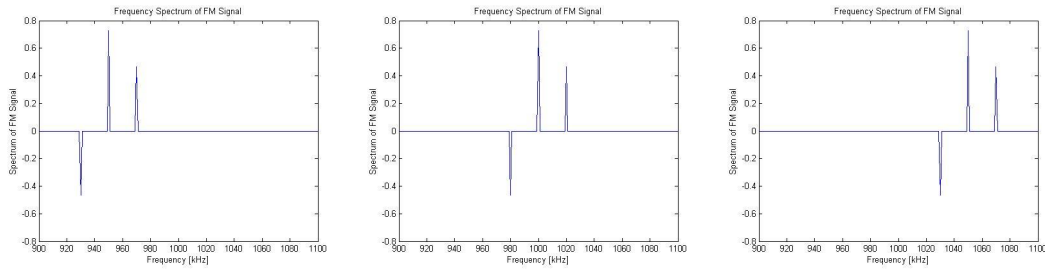


Figure 2.3: Sequential frequency domain plots of the FM signal resulting from a linear frequency ramp of the carrier.

As shown in Figure 2.3, the lower sideband is 180° out of phase with the carrier and upper sideband. The ramped FM signal inductively couples to the passive resonant frequency sensor. The sensor absorbs energy from the transmitted FM signal near the sensors resonant frequency. At all other frequencies, the sensor has no effect on the transmitted FM signal. Near the sensors resonant frequency, the sensor affects both the amplitude and phase of the transmitted FM signal as seen by the frequency response of a passive resonant frequency sensor shown in Figure 2.4. Therefore, the three frequency components of the transmitted FM signal experience both amplitude attenuation and phase shift due to the sensor. The results of passing the transmitted signal through the passive resonant frequency sensor are shown in the frequency domain plots in Figure 2.5.

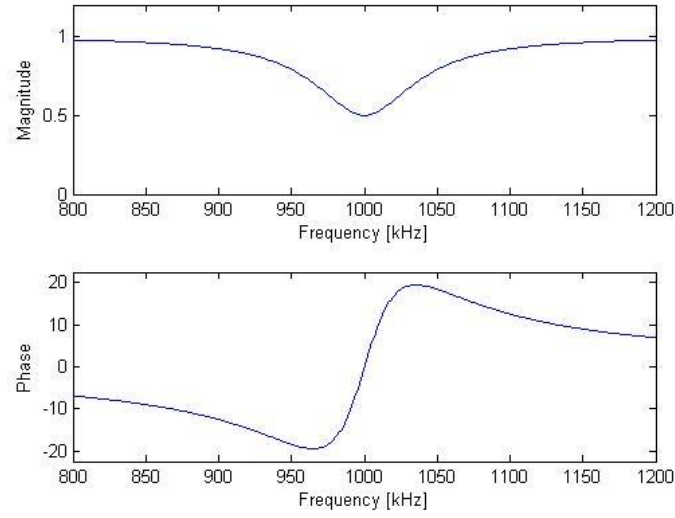


Figure 2.4: Transfer function of a passive resonant frequency sensor.
($f_r = 1000$ kHz, $Q = 10$)

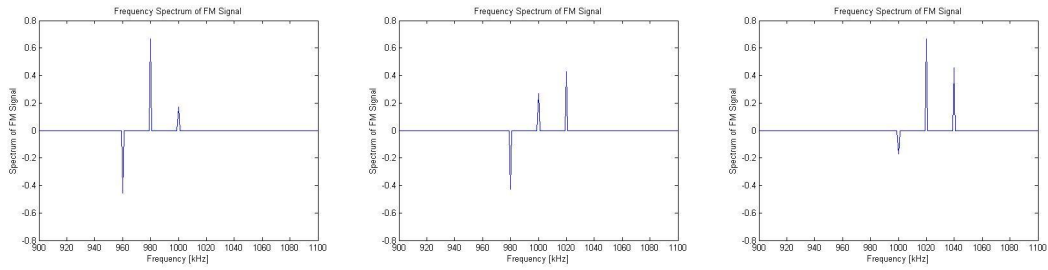


Figure 2.5: Sequential frequency domain plots showing the effect of the passive resonant frequency sensor on the ramped FM signal.

It is useful to define the sensor transfer function as a frequency component dependent amplitude attenuation and phase shift. As shown in Equation 2.9, T_n is the frequency component dependent transfer function of the sensor. δ_n is the attenuation of the n^{th} frequency component due to the sensor, and ϕ_n is the phase shift of the n^{th} frequency component due to the sensor.

$$T_n = \exp[-\delta_n - j\phi_n] \quad [2.9]$$

After passing through the passive resonant frequency sensor, the received signal $r(t)$ is

$$r(t) = A_c [T_{-1} J_{-1}(\beta) \exp(j(2\pi f_c - 2\pi f_m)t) + T_0 J_0(\beta) \exp(j2\pi f_c t) + T_1 J_1(\beta) \exp(j(2\pi f_c + 2\pi f_m)t)] \quad [2.10]$$

The received signal $r(t)$ is next passed through a square-law device. The output of the square-law device is the magnitude squared of the input, $|r(t)|^2$. If $|\delta_0 - \delta_1|$, $|\delta_0 - \delta_{-1}|$, $|\phi_0 - \phi_1|$, and $|\phi_0 - \phi_{-1}|$ are all $\ll 1$ and all of the terms of $|r(t)|^2$ containing $2f_c$ are dropped, then the output of the square-law device can be written as

$$\begin{aligned} |r(t)|^2 = A_c^2 \exp(-2\delta_0) [J_0^2(\beta) + (\delta_{-1} - \delta_1) J_0(\beta) J_1(\beta) \cos(2\pi f_m t) \\ + (\phi_1 + \phi_{-1} - 2\phi_0) J_0(\beta) J_1(\beta) \sin(2\pi f_m t)] \end{aligned} \quad [2.11]$$

Thus, the output of the square-law device will contain a heterodyne beat signal at the modulation frequency $2\pi f_m$, if $\delta_{-1} - \delta_1 \neq 0$ or if $\phi_1 + \phi_{-1} - 2\phi_0 \neq 0$. The heterodyne beat signal arises from the upper and lower sidebands beating against the carrier.

If the modulation frequency $2\pi f_m$ is chosen to be large compared to the spectral feature of interest, then the spectral feature can be probed by a single sideband. In this case, the amplitude attenuations and phase shifts experienced by the carrier and lower sideband remain almost constant. If we let $\delta_{-1} = \delta_0 = \bar{\delta}$ and $\phi_{-1} = \phi_0 = \bar{\phi}$, where $\bar{\delta}$ and $\bar{\phi}$ are constant background amplitude attenuation and phase shift respectively, then equation 2.10 can be written as

$$\begin{aligned} |r(t)|^2 = A_c^2 \exp(-2\delta_0) [J_0^2(\beta) + \Delta\delta J_0(\beta) J_1(\beta) \cos(2\pi f_m t) \\ + \Delta\phi J_0(\beta) J_1(\beta) \sin(2\pi f_m t)] \end{aligned} \quad [2.12]$$

where $\Delta\delta$ and $\Delta\phi$ are the changes from the constant background amplitude attenuation and phase shift. The $\cos(2\pi f_m t)$ and $\sin(2\pi f_m t)$ components of the heterodyne beat signal are, respectively, proportional to the absorption and dispersion from the spectral feature of the passive resonant frequency sensor. In the absence of any sensors, the output heterodyne beat signal is zero. This can be viewed as the perfect cancellation of the signal arising from the upper sideband beating against the carrier with the signal from the lower sideband beating against the carrier. The inherent high sensitivity of this sensing technique to amplitude or phase changes experienced by the probing sideband stems from the disturbance of this perfect cancellation.

To reiterate, the in-phase $\cos(2\pi f_m t)$ component of the heterodyne beat signal is proportional to the absorption induced by the spectral feature and the quadrature $\sin(2\pi f_m t)$ component of the heterodyne beat signal is proportional to the dispersion induced by the spectral feature. The heterodyne beat signal is then passed through a mixer to output a DC signal. The phase adjuster shown in Figure 2.1 is used to adjust the phase of the single frequency oscillator signal mixed with the heterodyne beat signal. Depending on the phase shift of the phase adjuster, the absorption output signal or the dispersion output signal can be obtained. The sensor transfer function, absorption output signal and the dispersion output signal are shown in Figures 2.6-2.9 for increasing values of the quality factor Q .

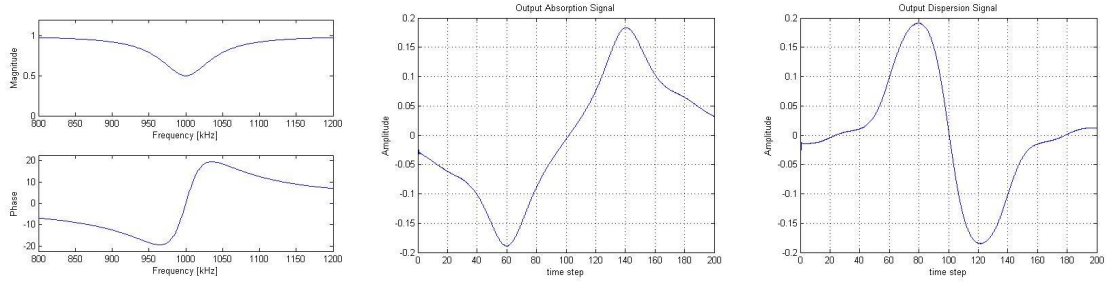


Figure 2.6: Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=10$.

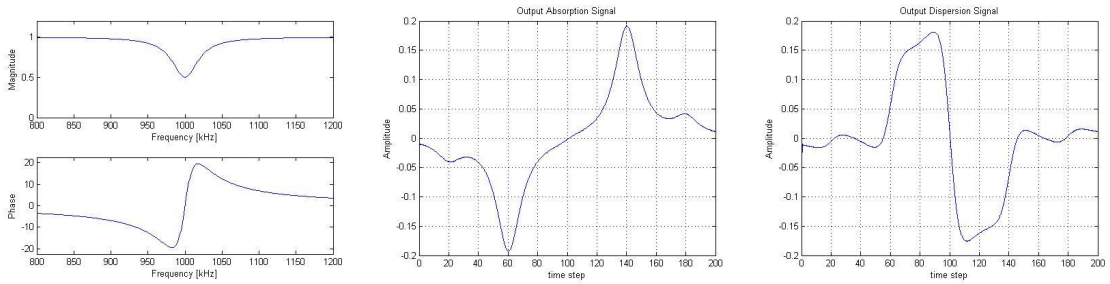


Figure 2.7: Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=20$.

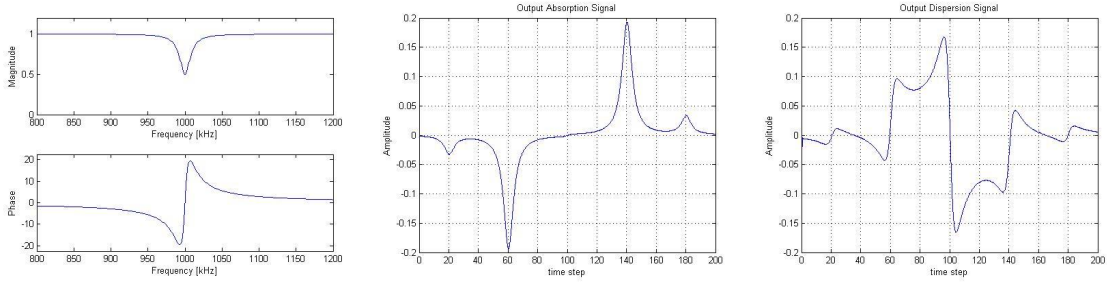


Figure 2.8: Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=50$.

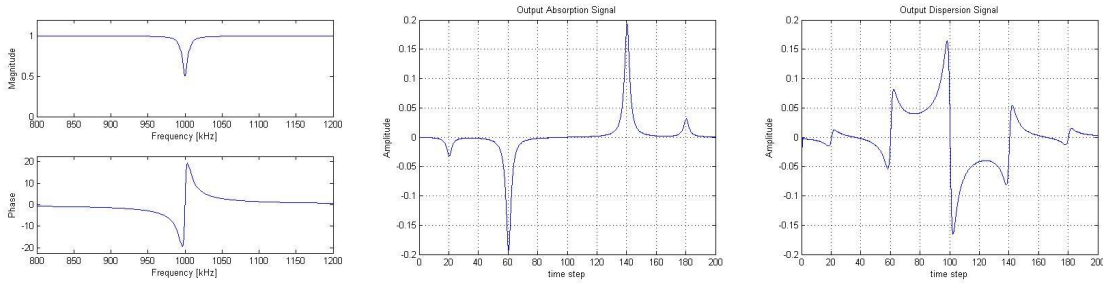


Figure 2.9: Sensor transfer function, output absorption signal, and output dispersion signal for a resonant frequency sensor with $f_r=1000$ kHz and $Q=100$.

The wireless interrogator is able to lock to the resonant frequency of a passive sensor by using the output DC signal to adjust the carrier frequency, f_c , of the adjustable frequency oscillator shown in Figure 2.1. The dispersion output signal (Figures 2.6-2.9) is the signal used to detect the resonant frequency since it always has a sharp, well defined zero crossing at the time corresponding to when the carrier frequency, f_c , equals the resonant frequency of the passive wireless sensor.

The effective portion of the dispersion output signal, which is useful for detecting the resonant frequency of the sensor, is limited to the linear region near the negative slope zero crossing seen near time step 100 in Figures 2.6-2.9. If operation is outside this linear region, the wireless interrogator will lose its lock on the resonant frequency of the passive sensor. Inside the linear region, the wireless interrogator is able to lock to the resonant frequency of the passive sensor and track changes in the resonant frequency of the passive sensor over time.

CHAPTER THREE

WIRELESS SENSOR INTERROGATOR DESIGN

Introduction

The purpose of this research is to develop a wireless sensing system capable of monitoring the resonant frequency of a passive sensor. The wireless sensor interrogator uses the highly sensitive sensing technique called frequency modulation spectroscopy explained in detail in Chapter 2. This chapter will expand on the basic experimental setup for frequency modulation spectroscopy depicted in Figure 2.1 which is repeated here for convenience in Figure 3.1. The wireless sensor interrogator design includes transmitted signal generation, received signal processing, feedback signal generation, and system control.

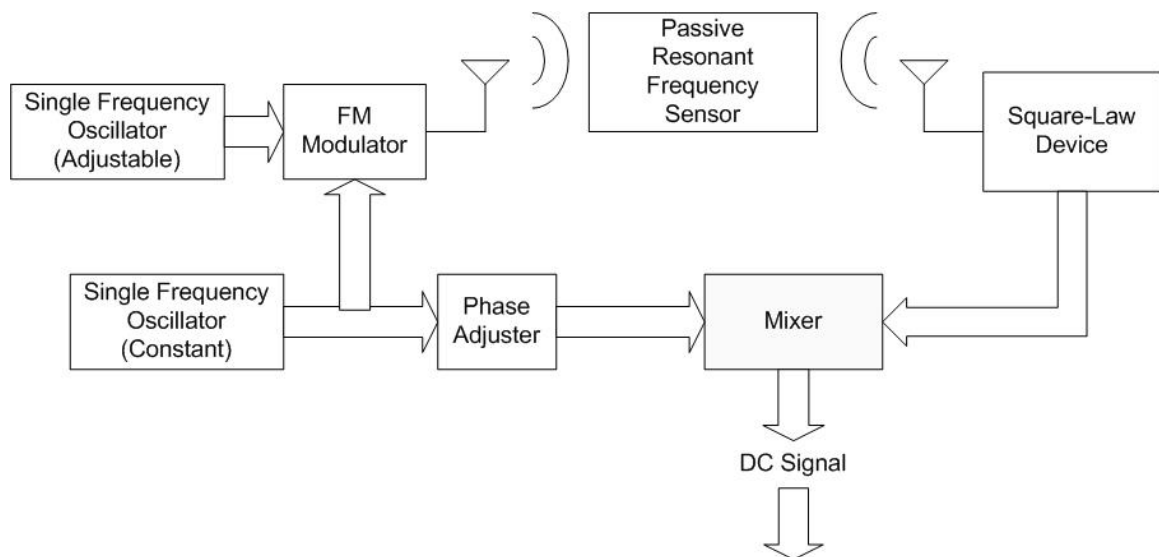


Figure 3.1: Typical experimental setup for frequency modulation spectroscopy.

Wireless Sensor Interrogator Design Detail

In Figure 3.1, the transmitted signal generation is depicted by two sinusoidal signal sources and a frequency modulation block. Although shown as just three boxes, generating the transmitted FM signal with a swept carrier frequency is a challenging design problem. The carrier frequency of the transmitted FM signal must be adjustable, have small frequency steps, and be highly accurate. Additional design challenges include careful filtering of the outputs of the square-law device and the mixer, generation of the feedback signals for tracking, and outputting the carrier frequency data to a personal computer (PC). The block diagram of the wireless sensor interrogator is shown in Figure 3.2.

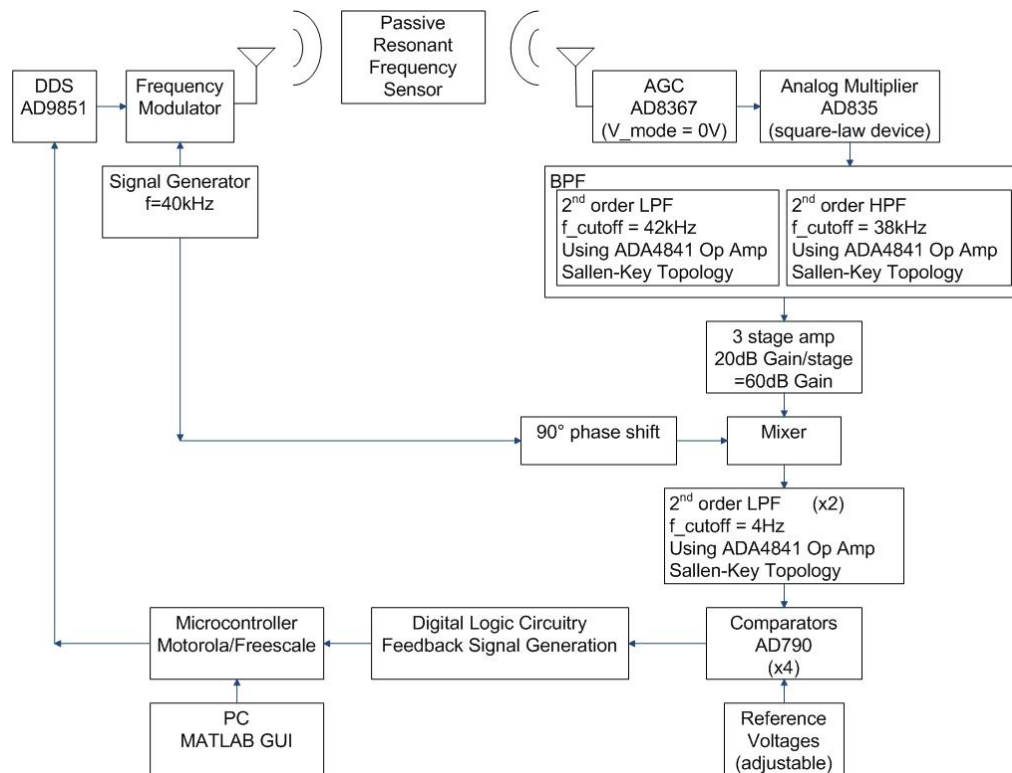


Figure 3.2: Block diagram of the wireless sensor interrogator design.

Transmitted Signal Generation

The transmitted signal is a frequency modulated (FM) signal with an adjustable carrier frequency. During normal operation, the carrier frequency is linearly swept through the frequency range of interest. When a resonant frequency sensor is encountered in the frequency range of interest, the carrier frequency of the transmitted FM signal is adjusted such that the output dispersion signal is at the negative slope zero crossing shown in the center of Figure 2.7.

The design of the transmitted signal generation circuitry consists of a signal generator to provide a programmable frequency sinusoid, a microcontroller to program the Direct Digital Synthesis (DDS) IC, and a modified PLL to perform the frequency modulation. Each of these 4 blocks are presented in detail in the following sections.

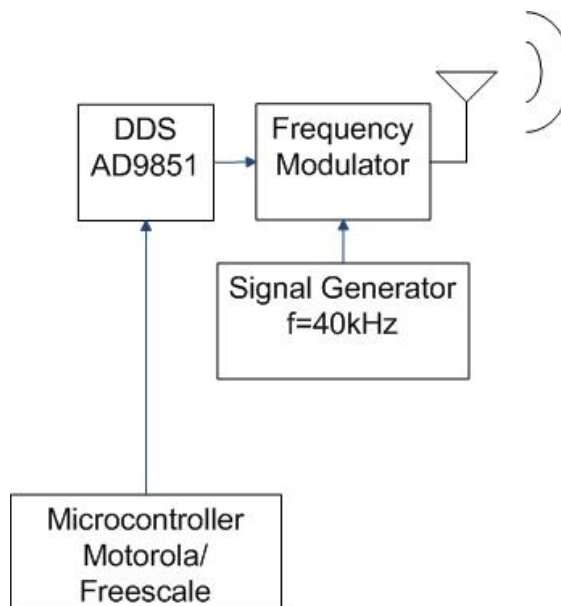


Figure 3.3: Block diagram of transmitted signal generation.

Signal Generator: An Agilent 33220A signal generator was used to generate the constant single frequency sinusoidal signal used to modulate the carrier signal. In future designs, the signal generator could easily be replaced by a crystal oscillator or a second DDS IC.



Figure 3.4: Picture of signal generator.

DDS IC: In order to generate a sinusoidal signal with a highly adjustable and accurate output frequency, a programmable signal synthesizer was chosen. With a high quality reference frequency, such as a crystal oscillator, the signal synthesizer is able to produce a sinusoidal output signal that is very accurate and repeatable.

Analog Devices, Inc. has combined frequency multipliers, phase accumulators, and high performance digital to analog converters to make programmable signal generation ICs called Direct Digital Synthesis (DDS). The AD9851 DDS IC from Analog Devices was selected for its high frequency accuracy and agile frequency control. A printed circuit board assembly utilizing this IC was obtained and is shown in Figure 3.5.

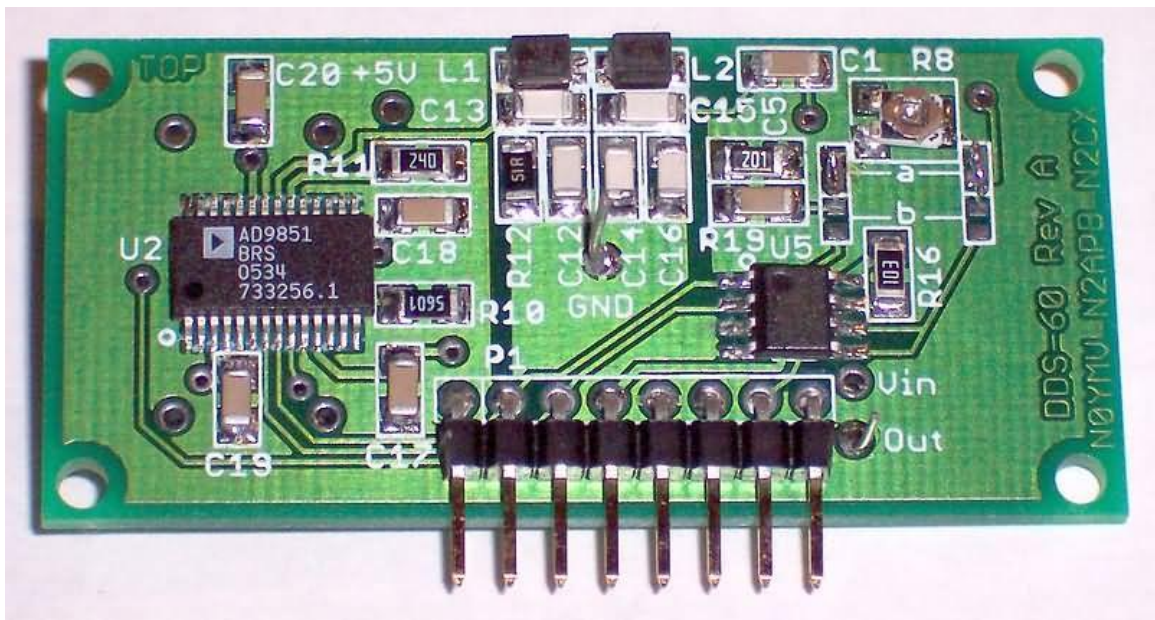


Figure 3.5: PCB assembly with AD9851 DDS IC.

Digital control of the AD9851 is necessary to program the output frequency of each frequency step. Once a passive resonant frequency sensor is detected, the output frequency must be highly adjustable to enable the wireless sensor interrogator to lock to and track the resonant frequency of the sensor. A Freescale (formerly Motorola) M68HC12 16 bit microcontroller is used to program the AD9851 DDS IC. The

frequency modulator uses both the constant and adjustable frequency sinusoidal signals to make the FM signal. The frequency modulator is discussed next.

Frequency Modulator: To implement the frequency modulator shown in Figure 3.2, a modified Phase Lock Loop (PLL) scheme has been designed. This section describes the basic components of the modified PLL scheme and the desired output from the PLL.

A PLL is simply a feedback system that compares the phase of the output signal from the PLL with the phase of a reference signal and works to minimize the differences in phase. There are three main components to a basic PLL: a phase comparator, a loop filter, and a voltage controlled oscillator (VCO). A basic PLL is shown in Figure 3.6.

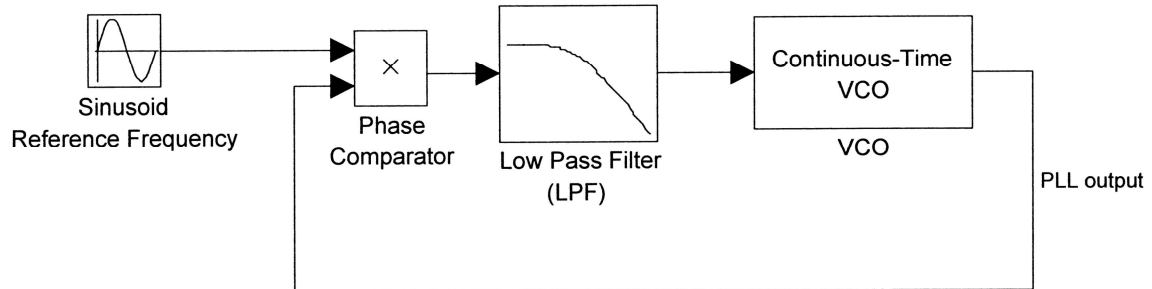


Figure 3.6: Basic elements of a PLL.

The phase comparator outputs an error signal proportional to the difference in phase between the output signal from the PLL and the reference input signal. The loop filter is a low pass filter (LPF) that works to stabilize the PLL feedback system. Careful consideration must be paid to the design of the LPF. The wider the bandwidth of the LPF, the faster the PLL will lock to the reference signal, but the PLL will also have more

frequency and phase jitter in the output signal. This inherent tradeoff must be balanced thoughtfully. The VCO translates the loop filters output voltage, which is a filtered version of the error signal, into an output frequency. The nominal frequency of oscillation of the VCO must be close to that of the reference signal, unless a frequency divider is employed.

For this design, the basic configuration of the PLL is modified to inject a low frequency sinusoid into the loop in order to implement frequency modulation. The basic equations governing the operation of the VCO show how frequency modulation is made possible. The VCO is basically a sinusoidal signal generator with an instantaneous phase given by

$$\phi = 2\pi K_{VCO} \int_{-\infty}^t v(\tau) d\tau \quad [3.1]$$

for a linear model of the PLL. The input $v(t)$ is called the VCO's control voltage and is taken from the output of the LPF, in the case of the basic PLL. In the modified PLL design, a low frequency sinusoid is added to the output of the LPF such that the VCO's control voltage becomes

$$v(t) = LPF_{out} + a \cos(2\pi f_m t) \quad [3.2]$$

The modified PLL design is shown in Figure 3.7.

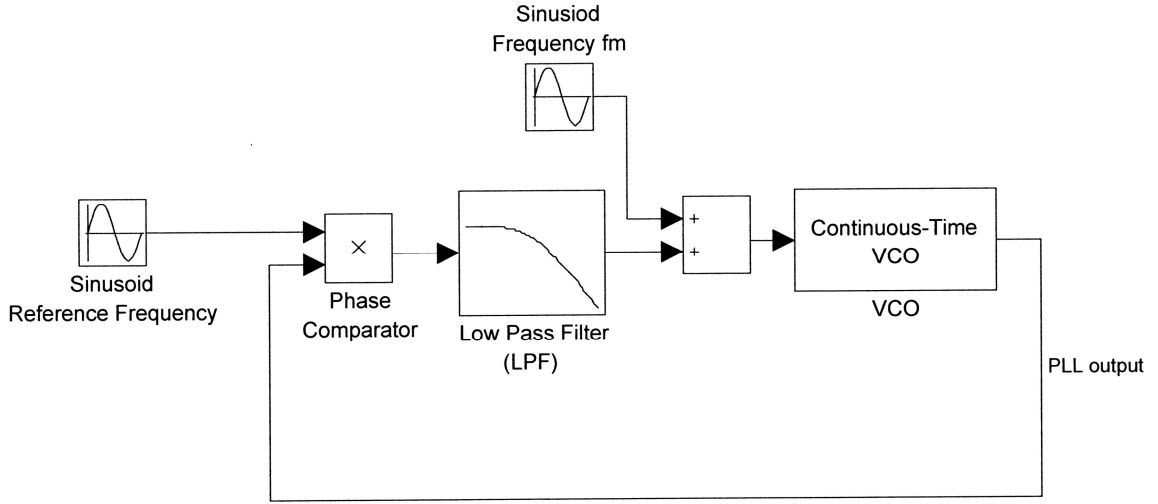


Figure 3.7: PLL with sinusoid added to the VCO control voltage.

The time varying output voltage from the VCO is

$$V_{out}(t) = V_m \cos(2\pi f_o t + 2\pi K_{vco} \int_{-\infty}^t v(\tau) d\tau) \quad [3.3]$$

Combining equations 3.2 and 3.3 and completing the integration, we get

$$V_{out}(t) = V_m \cos(2\pi f_o t + 2\pi K_{vco} LPF_{out} t + \frac{K_{vco} a}{f_m} \sin(2\pi f_m t)) \quad [3.4]$$

From communication systems theory [14], the constant in front of the sin term is recognized as the frequency modulation index β .

$$\beta = \frac{K_{vco} a}{f_m} \quad [3.5]$$

This configuration allows an arbitrary frequency modulation index to be chosen based on the proportional constant of the VCO, the amplitude of the added sinusoid, and the frequency of the added sinusoid. Control of the frequency modulation index is important for optimizing the performance of the wireless sensor interrogator. Note the

similarity between equation 3.5 and equation 2.5. Combining equations 3.4 and 3.5, we get

$$V_{out}(t) = V_m \cos(2\pi f_o t + 2\pi K_{vco} LPF_{out} t + \beta \sin(2\pi f_m t)) \quad [3.6]$$

Thus, the PLL is able to output a frequency modulated signal whose carrier frequency tracks the input reference frequency. This signal is the transmitted signal for the wireless sensor interrogator. The physical implementation of the loop filter design is shown in Figure 3.8.

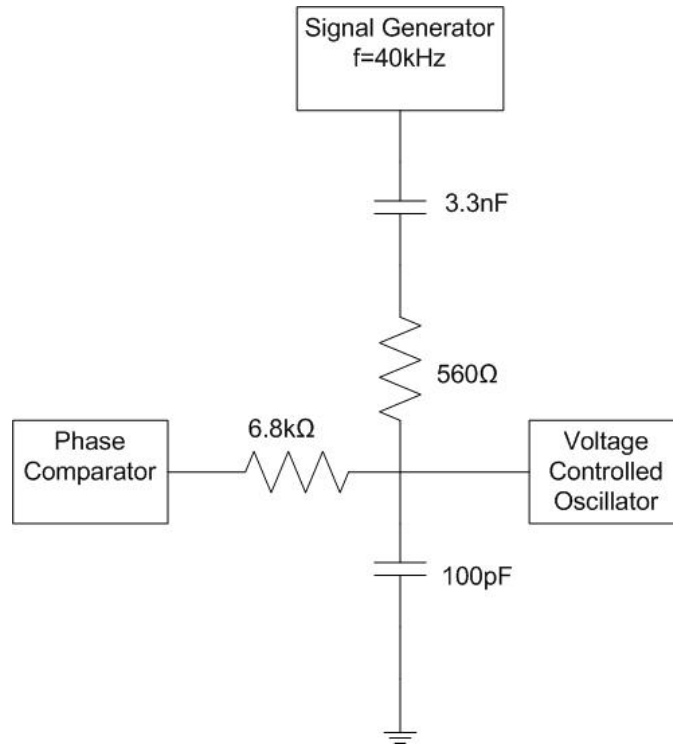


Figure 3.8: Physical implementation of the loop filter.

The MC14046B PLL IC provides the phase comparator and the VCO.

Microcontroller: A M68HC12 16 bit microcontroller from Freescale was selected to program the AD9851 DDS IC. The DDS IC outputs a sinusoid with a frequency determined by the value of an internal register. The microcontroller loads the internal register of the DDS IC through a serial peripheral interface (SPI). The SPI has three wires: serial clock, data, and load. Unlike standard SPI, there is no data transfer from the DDS IC to the microcontroller. Under normal operation, the microcontroller outputs a serial bit stream to the DDS IC and toggles the load pin on the DDS IC to update the output frequency at a specified time interval. This process is repeated, producing an output sinusoid whose frequency keeps incrementing up. The ramped frequency sinusoid is then provided to the frequency modulator block as the carrier signal.

In addition to being able to increment the carrier frequency, the microcontroller is also able to decrement the carrier frequency. The ability to precisely control and adjust the carrier frequency of the FM signal allows the wireless sensor interrogator to lock to and track the resonant frequency of the passive resonant frequency sensor. The carrier frequency is incremented or decremented according to the value of three feedback signals: lock, increment, and decrement. The three feedback signals are generated by monitoring the dispersion output signal seen in Figure 2.7. The generation of the feedback signals is discussed later. The microcontroller code is shown in appendix B and was written by Justin Hadella, a student at Montana State University.

Received Signal Processing

The transmitted FM signal with swept carrier frequency is affected by the passive resonant frequency sensor as described in the frequency modulation spectroscopy theory

section in Chapter 2. In order to extract the associated amplitude attenuations and phase shifts due to the resonant frequency sensor, careful signal processing of the received signal is required. The block diagram for the processing of the received signal is shown in Figure 3.9.

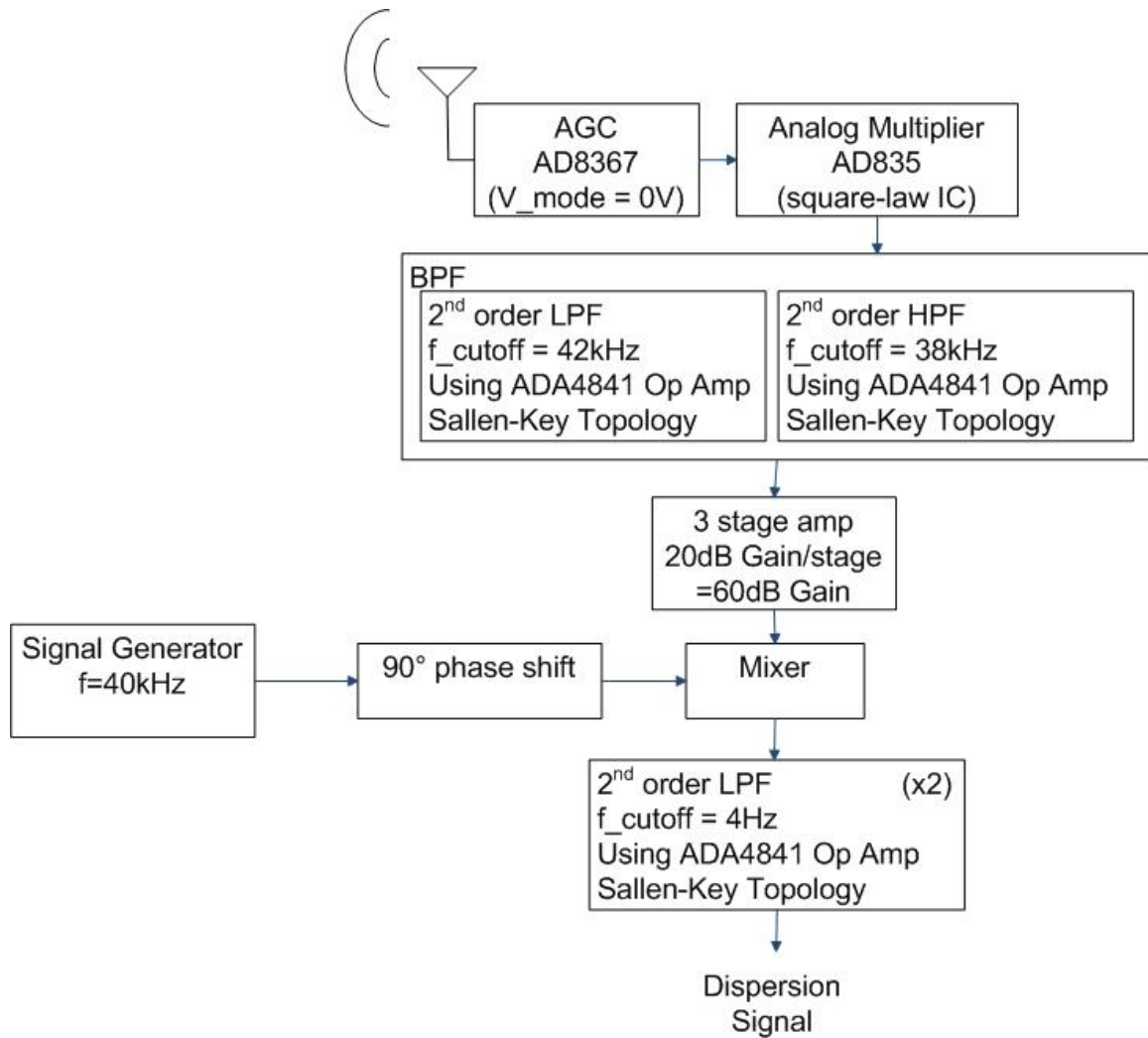


Figure 3.9: Block diagram of received signal processing.

Depending on the amount of phase shift applied to the sinusoidal signal from the signal generator, either the effects of the amplitude attenuations or the effects of the

phase shifts due to the sensor can be obtained. The output signal arising from the effects of the sensors amplitude attenuations is called the absorption signal, and the output signal arising from the effects of the sensors phase shifts is called the dispersion signal. The output absorption signal and the output dispersion signal are near DC in frequency. The output dispersion signal, shown in Figure 2.7, has a negative slope zero crossing at the time corresponding to when the carrier frequency of the transmitted FM signal equals the resonant frequency of the sensor. This zero crossing can be detected and tracked, thereby enabling the wireless interrogator to lock to and track the resonant frequency of the sensor. Thus, this achieves the purpose of the received signal processing which is to generate the output dispersion signal.

The following sections describe the blocks in the received signal processing chain including automatic gain control, square-law IC, bandpass filter, amplification, signal generator, phase shifter, mixer, and lowpass filter.

Automatic Gain Control: In any wireless communication system it is very difficult, if not impossible, to accurately predict the power level of the received signal. Therefore, it is almost always necessary to employ some amplification of received low power signals. Amplifiers with automatic gain control (AGC) adjust the gain until the peak amplitude of the signal reaches some predetermined value. These amplifiers are often called variable gain amplifiers. The AD8367 amplifier with AGC from Analog Devices was selected for use in this design because of the availability of an evaluation board from Analog Devices. The evaluation board is shown in Figure 3.10. The square-law IC is discussed next.

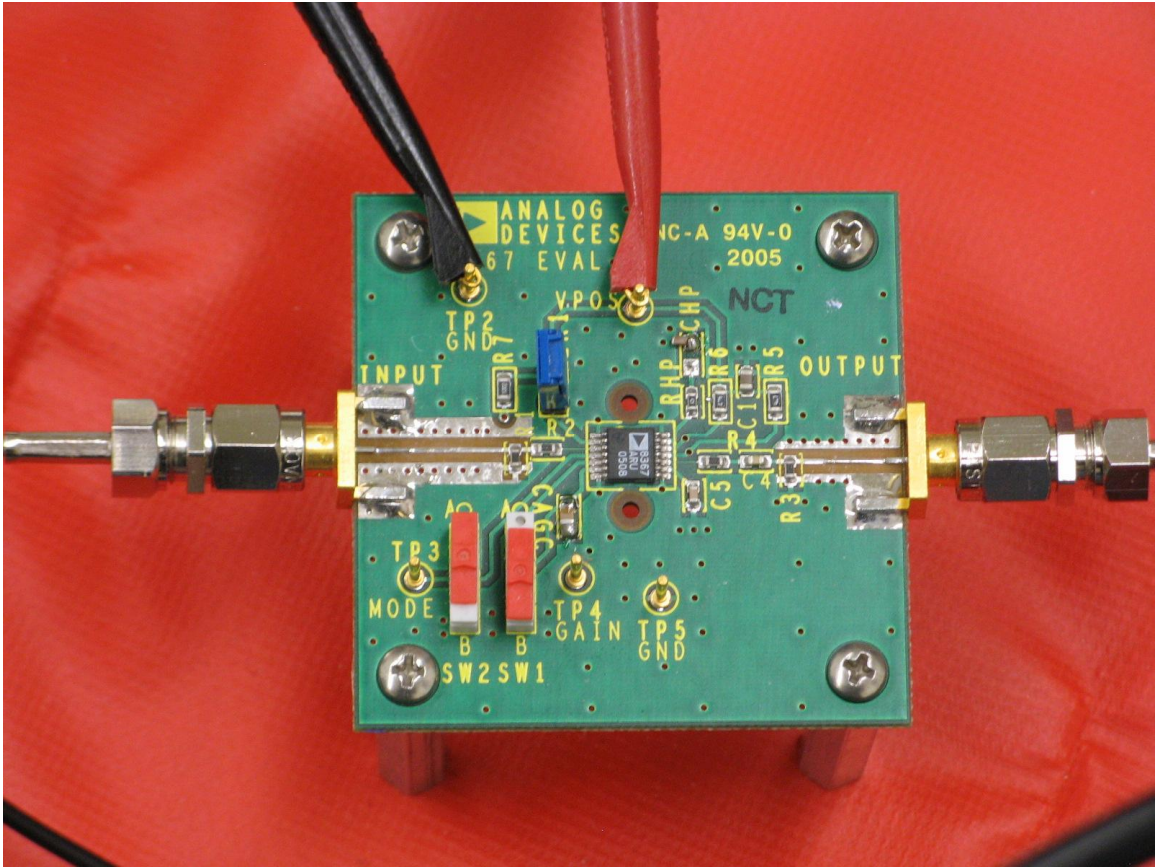


Figure 3.10: Picture of AD8367 evaluation board from Analog Devices, Inc..

Square-law IC: The AD835, an analog multiplier from Analog Devices, Inc. was selected to perform the square-law function. The functional block diagram of the AD835 is shown in Figure 3.11. For $X_1=Y_1=\text{input}$ and $X_2=Y_2=Z=0$, the square-law output, W , is the magnitude-squared of the input. This signal processing function creates signal frequency components clustered near twice the carrier frequency, near DC, and, if a resonant frequency sensor is present, at the modulation frequency f_m . Information about the amplitude attenuations and phase shifts experienced by the transmitted FM signal due to the resonant frequency sensor is contained in the signal component at the modulation frequency f_m . For an explanation of why the information is contained in the signal

component at frequency f_m , see the frequency modulation spectroscopy theory section in Chapter Two.

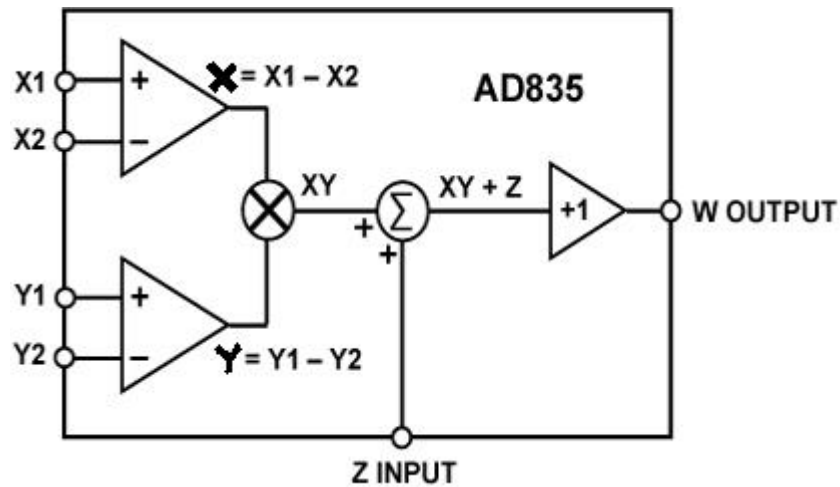


Figure 3.11: Functional block diagram of AD835 analog multiplier IC.

Bandpass Filter: The output of the square-law IC is sent to a bandpass filter (BPF) in order to pass frequencies near f_m and reject all other frequencies. The BPF is a second order filter, which is designed to have zero phase shift at the modulation frequency f_m . The BPF is an active filter designed and built using operational amplifiers in a Sallen-Key topology and consists of a 2nd order LPF cascaded with a 2nd order HPF. The block diagram of the BPF is shown in Figure 3.12 and the schematic of the BPF is shown in Figure 3.13. The output of the BPF is one of two required inputs to the mixer.

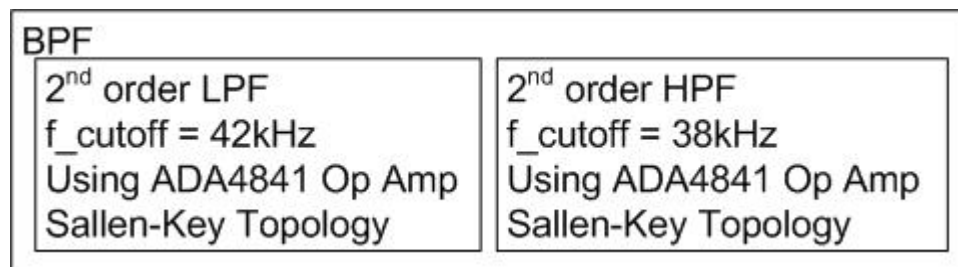


Figure 3.12: Block diagram of bandpass filter (BPF).

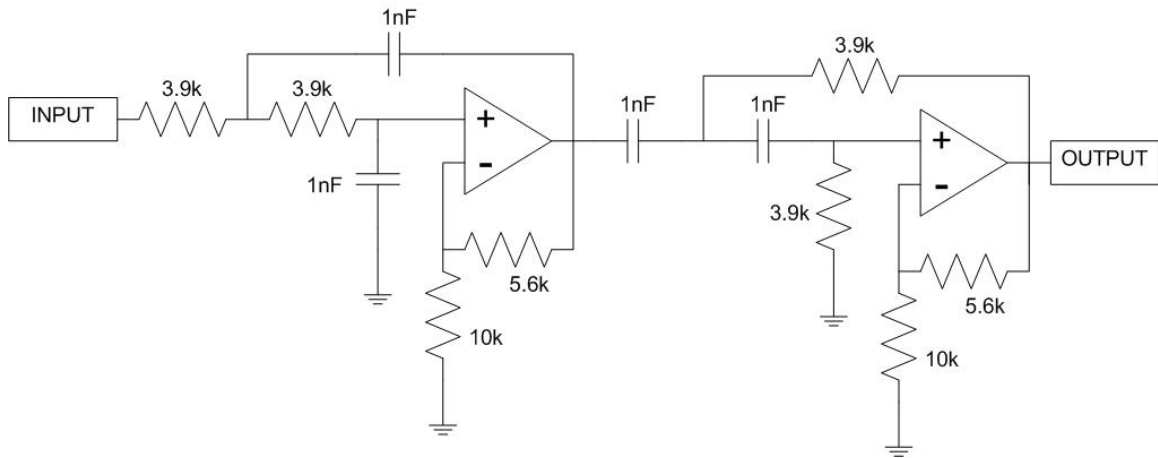


Figure 3.13: Schematic of bandpass filter (BPF) circuit.

Amplifier: The output of the BPF was found to be too small in magnitude to input directly into the mixer, therefore an amplifier was added to the design between the BPF and the mixer. The signal of interest is a voltage signal with a frequency near the modulation frequency, $f_m=40\text{kHz}$. The design of the amplifier is simple given the relatively low frequency of the signal. A basic non-inverting amplifier topology using an ADA4841 Op Amp was selected. The circuit schematic for the amplifier is shown in Figure 3.14. The amplifier consists of three identical stages cascaded together. Each stage provides 20dB of gain, therefore all three stages together provide a total of 60dB of gain.

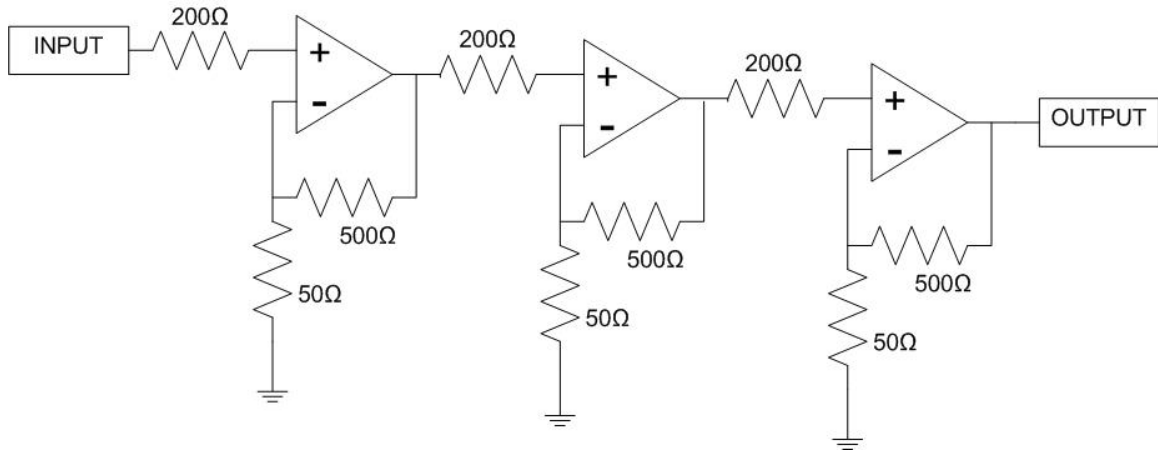


Figure 3.14: Schematic of amplifier circuit.

Signal Generator: The same signal used to modulate the carrier signal in the transmitted signal generation is also used in the received signal processing, as shown in Figure 3.2. The modulating signal, shown in equation 3.1, is output by the signal generator.

$$m(t) = a \cos(2\pi f_m t) \quad [3.1]$$

The amplitude a is selected by the designer to meet the modulation index requirements. The modulation frequency f_m is selected to be 40kHz. The signal generator is shown in Figure 3.4.

Phase Shifter: The modulation signal, Equation 3.1, is sent to a phase shifter. The phase shifter is an allpass filter designed to introduce a constant 90° phase shift at the modulation frequency, f_m . The output of the phase shifter is a phase shifted version of the modulating signal, and is given by

$$m_{shifted}(t) = a \cos(2\pi f_m t - \phi) \quad [3.2]$$

where $\phi = \frac{\pi}{2}$ radians. This phase shifted version of the modulating signal is the second input to the mixer. The schematic of the phase shifter circuit is shown in Figure 3.15.

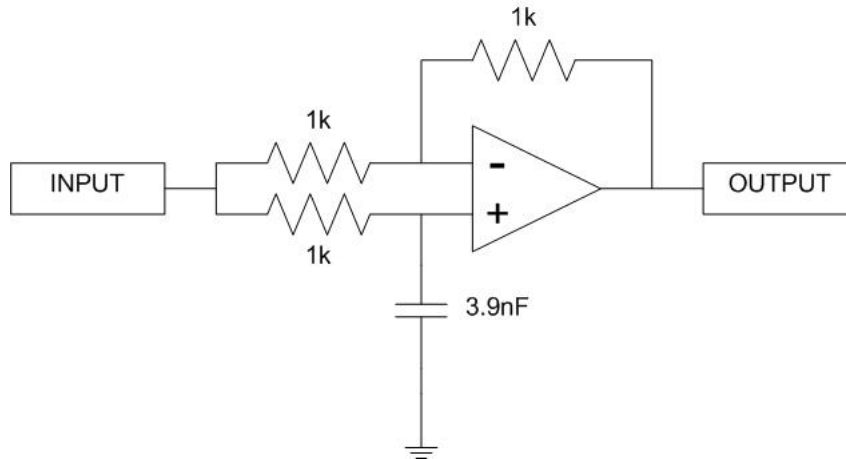


Figure 3.15: Schematic of phase shifter circuit.

Mixer: A mixer multiplies two input signals together. The bandpass filtered output of the square-law IC is the first signal provided to the mixer and the phase shifted modulating signal in Equation 3.2 is the second signal provided to the mixer. The output of the mixer contains frequency components near twice the modulation signal frequency and frequency components near DC. A good mixer should exhibit the ability to function properly with small magnitude input signals, have a high degree of isolation between ports, and low loss in power between the input signals and output signal. A passive, double-balanced mixer from Mini-Circuits was chosen for its low noise and good isolation. The mixer is shown in Figure 3.16.

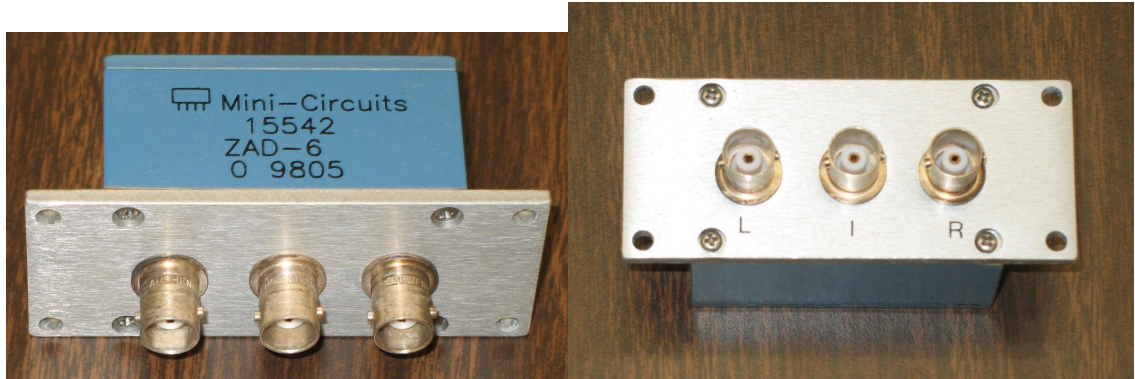


Figure 3.16: Pictures of the double-balanced mixer.

Lowpass Filter: The output of the mixer is sent to a lowpass filter (LPF) in order to pass frequencies near DC and reject all other frequencies, especially those strong frequency components near twice the modulation frequency. The chosen LPF is a second order active filter designed and built using operational amplifiers in a Sallen-Key circuit topology. The schematic of the LPF is shown in Figure 3.17. In order to achieve the necessary attenuation, two identical LPF's are cascaded together. The cutoff frequency for each filter is 4Hz. The output of the LPF corresponds to the DC output signal in Figure 3.1, which is the desired dispersion signal. The output dispersion signal is shown in Figure 3.18.

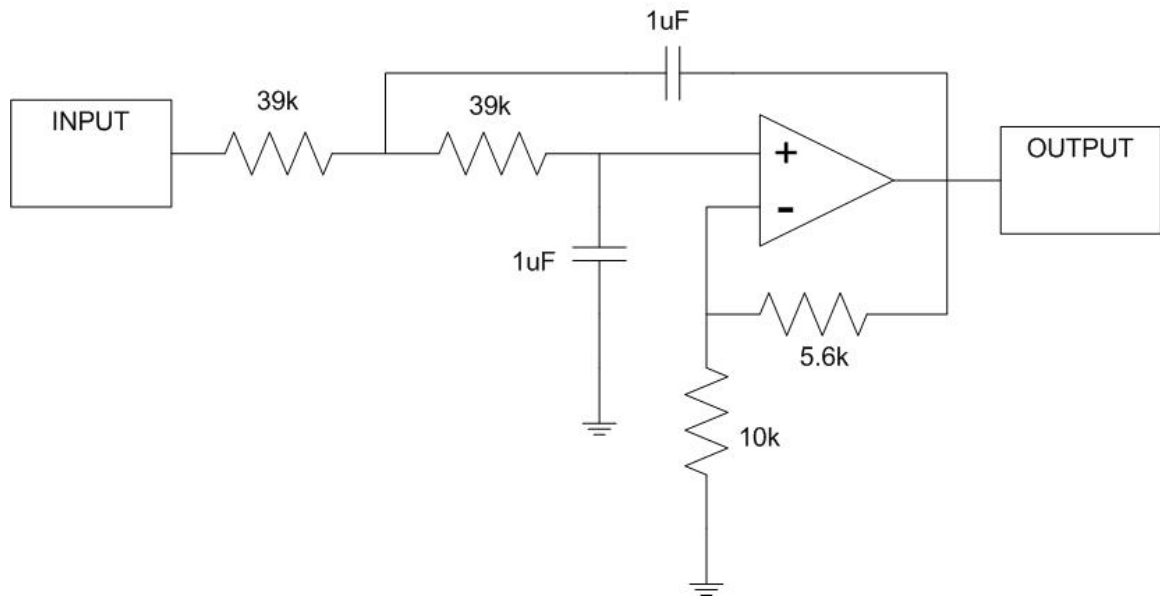


Figure 3.17: Schematic of lowpass filter (LPF) circuit.

Feedback Signal Generation

In order for the wireless sensor interrogator to lock to and track the resonant frequency of a passive sensor, some sort of feedback scheme must be used to control the adjustment of the transmitted carrier frequency. The output of the received signal processing is the output dispersion signal shown in Figure 3.18. In the feedback signal generation, the output dispersion signal is used to generate three different feedback signals: LOCK, increment (INC), and decrement (DEC). The detection levels for generating these three feedback signals are shown in Figure 3.18.

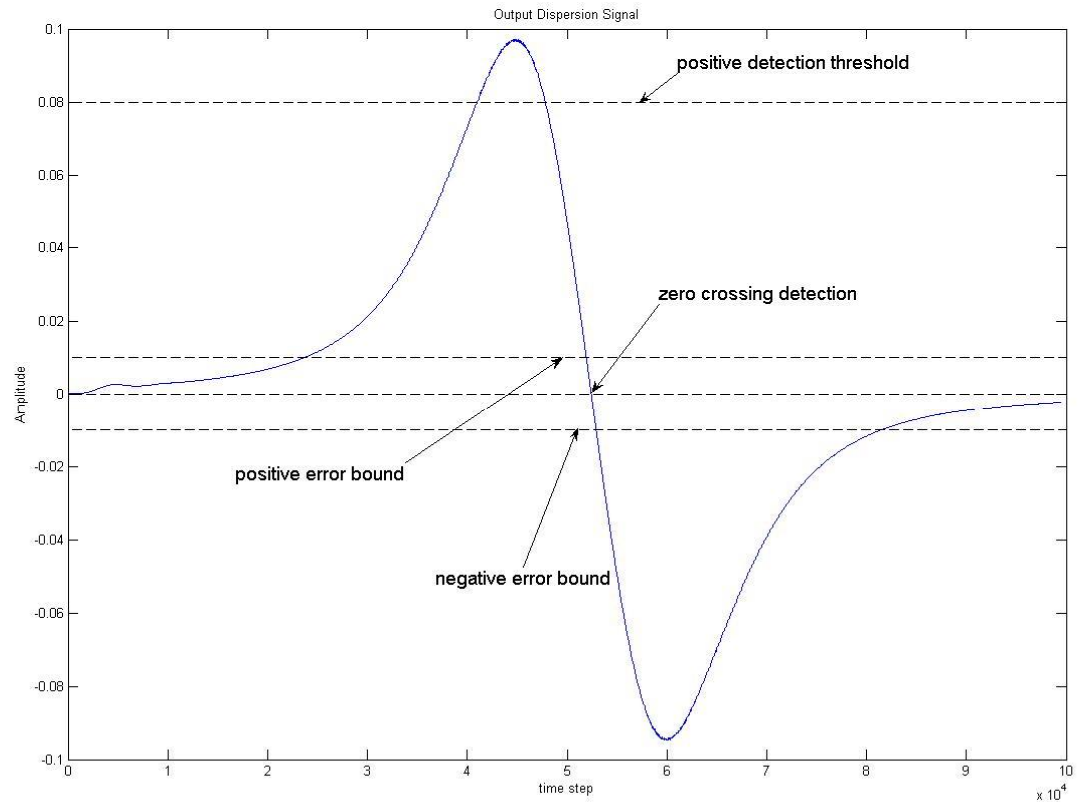


Figure 3.18: Output dispersion signal showing detection levels for generating feedback signals.

The LOCK feedback signal is used to stop the frequency sweep of the carrier frequency of the transmitted FM signal. The objective here is to stop the frequency sweep when the carrier frequency is exactly equal to the resonant frequency of the sensor. Although this is not possible, it is possible to get close. If the frequency sweep is stopped somewhere on the linear region of the negative slope zero-crossing of the output dispersion signal, then the INC and DEC feedback signals can be used to adjust the carrier frequency of the transmitted FM signal to be equal to the resonant frequency of the sensor.

The LOCK feedback signal is generated using two comparators and some digital logic components. The first comparator is used to detect when the output dispersion signal exceeds some predetermined threshold, such as three times the standard deviation of the noise, called the positive detection threshold. The first comparator indicates the start of the output dispersion signal, since the output is zero when there is no sensor in the frequency band of interest. The second comparator then detects when the carrier frequency of the transmitted FM signal equals the resonant frequency of the sensor. The LOCK signal generation is sequential, that is the first comparator must trigger before the second comparator.

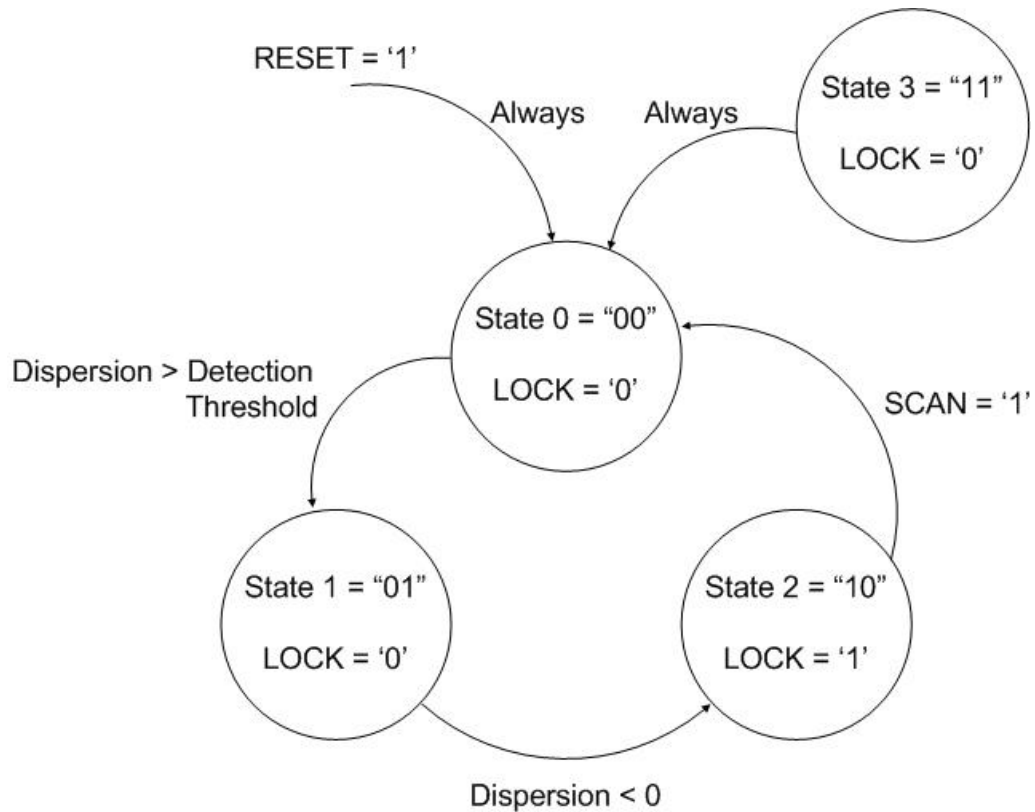


Figure 3.19: State diagram of the finite state machine used to generate the LOCK feedback signal.

The sequential behavior is implemented by a finite state machine (FSM). The state diagram is shown in Figure 3.19. The initial state, state 1, is entered whenever RESET is asserted by the microcontroller. The FSM advances to state 2 when the output dispersion signal exceeds the positive detection threshold. In state 2, when the output dispersion signal goes negative the FSM advances to state 3. The LOCK feedback signal is asserted in state 3. The FSM will stay in state 3 until either of the two control signals from the microcontroller, SCAN or RESET, are asserted. The schematic of the FSM is shown in Figure 3.20. There is no output logic since Q1 and LOCK are the same. Therefore Q1 is output to the microcontroller as the LOCK feedback signal.

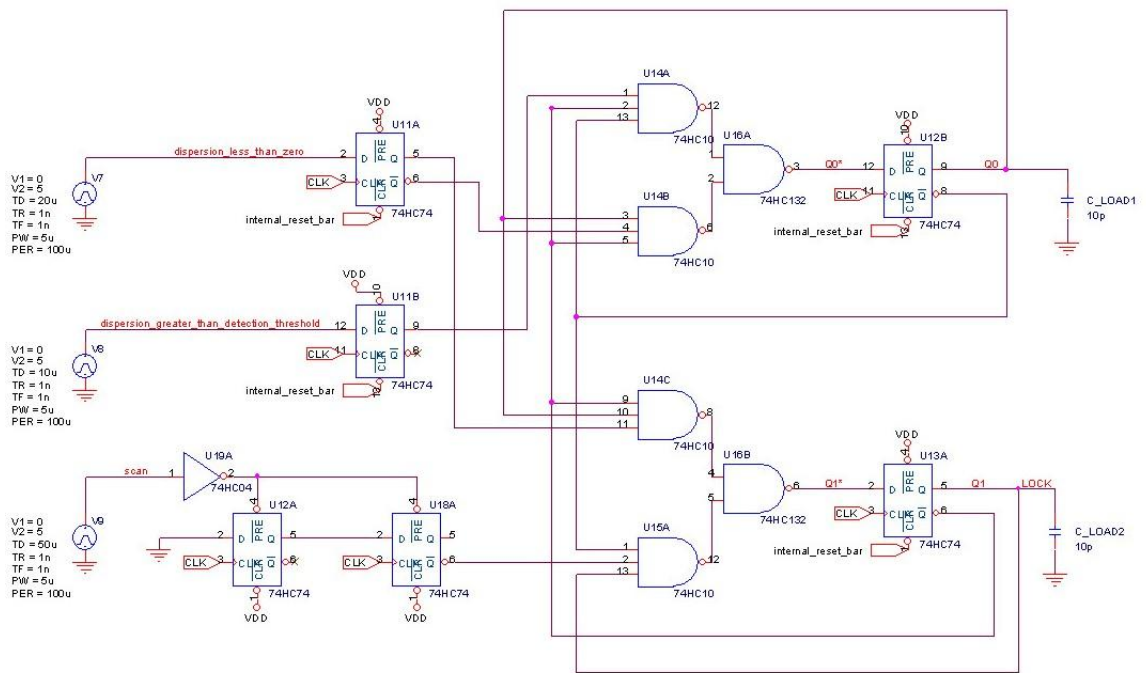


Figure 3.20: Schematic of the finite state machine used to generate the LOCK feedback signal.

The remaining two feedback signals, INC and DEC, are generated in a similar manner to each other. The INC feedback signal is generated by comparing the output

dispersion signal with some predetermined positive error threshold. When the output dispersion signal is larger than the positive error threshold, then the carrier frequency of the transmitted FM signal is less than the resonant frequency of the sensor, and the carrier frequency should be increased or incremented. The schematic of the circuitry for generating the INC feedback signal is shown in Figure 3.21. Likewise, the DEC feedback signal is generated by comparing the output dispersion signal with some predetermined negative error threshold. When the output dispersion signal is less than the negative error threshold, then the carrier frequency of the transmitted FM signal is greater than the resonant frequency of the sensor, and the carrier frequency should be decreased or decremented. The schematic of the circuitry for generating the DEC feedback signal is shown in Figure 3.21. As seen in Figure 3.21, there is a 4 bit binary ripple counter. The following paragraph explains the purpose of the ripple counter.

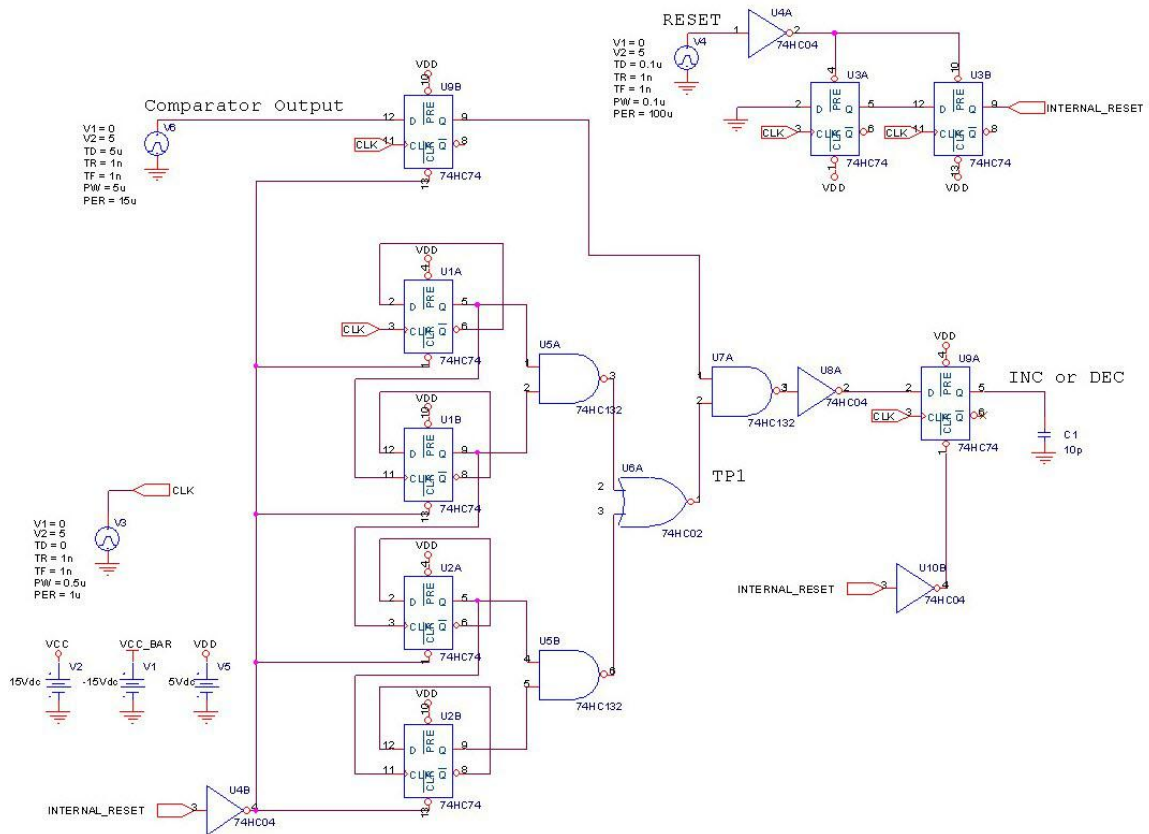


Figure 3.21: Schematic of the INC or DEC feedback signal generation circuitry.

There is a finite settling time of the output dispersion signal following the adjustment of the carrier frequency of the transmitted FM signal. This settling time arises from the transient response of the system. If the INC and DEC feedback signals are generated before the transient response of the wireless interrogator has died out, then errors in the INC or DEC feedback signals may result.

The signal at TP1 is a pulse with a pulse width of 1 clock period and a frequency that is 1/16 of the clock frequency. The signal at TP1 is AND-ed with the comparator output before being passed onto the microcontroller as the INC feedback signal.

Therefore the INC feedback signal will be asserted at most once every 16 clock cycles.

This gives the system time for the transient response to dissipate.

Similarly, the signal at TP1 is AND-ed with the other comparator output to generate the DEC feedback signal. The signal at TP1 can be used in the generation of both the INC and DEC feedback signals. Therefore, only one 4 bit ripple counter is required for both the INC and DEC feedback signal generation.

System Control/Closing the Loop

The wireless interrogator is controlled exclusively through a graphical user interface (GUI) in MATLAB. The user can start a frequency sweep, stop a frequency sweep, continue scanning to the next sensor after locking to a sensor, and plot the results in MATLAB.

Personal Computer: In addition to the SPI connection to the DDS IC, there is a serial port connection from the microcontroller to a personal computer (PC) to enable user control of the wireless sensor interrogator and to output the resonant frequency data to the PC for recording and post-processing. The RS-232 standard for transferring data is used to transfer commands and data between the microcontroller and the PC.

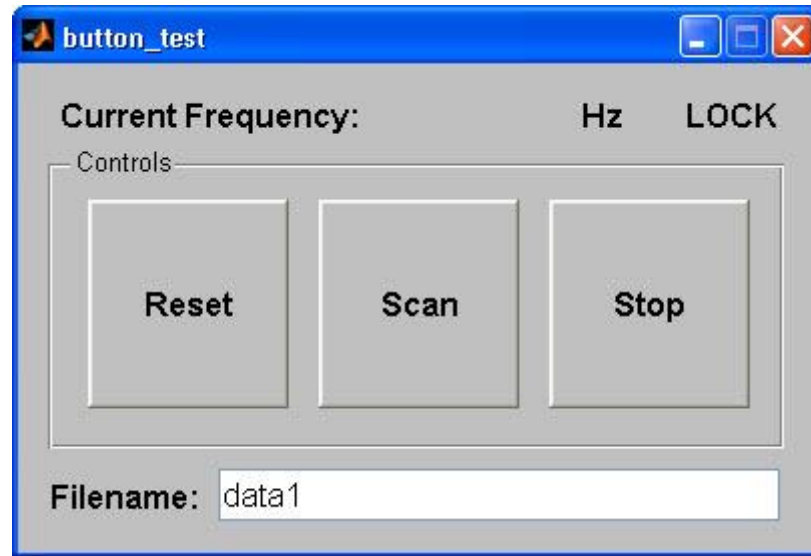


Figure 3.22: MATLAB graphical user interface used to control the wireless sensor interrogator.

The user interface with the PC consists of a MATLAB graphical user interface (GUI) that sends commands to the microcontroller. The basic display of the GUI is shown in Figure 3.22. The microcontroller accepts commands and sends the transmitted FM signal carrier frequency data to the PC. MATLAB puts the FM signal carrier frequency data into a MATLAB data file for post processing and/or graphing. Data processing and graphing are easily achieved using MATLAB capabilities. The MATLAB code for the GUI is shown in appendix C and was written by Justin Hadella, a student at Montana State University.

CHAPTER FOUR

WIRELESS SENSOR INTERROGATOR DESIGN RESULTS

Introduction

The wireless interrogator design presented in Chapter three utilizing a resonant frequency sensing technique taken from optical systems called frequency modulation spectroscopy was simulated in MATLAB's Simulink and built in hardware. The block diagram of the wireless sensor interrogator design is repeated in Figure 4.1. This chapter first presents the simulated results and then the hardware implementation results of the wireless sensor interrogator designed for this thesis work. A discussion focused on the effectiveness of the overall design and improvements that could be made to the design concludes the chapter.

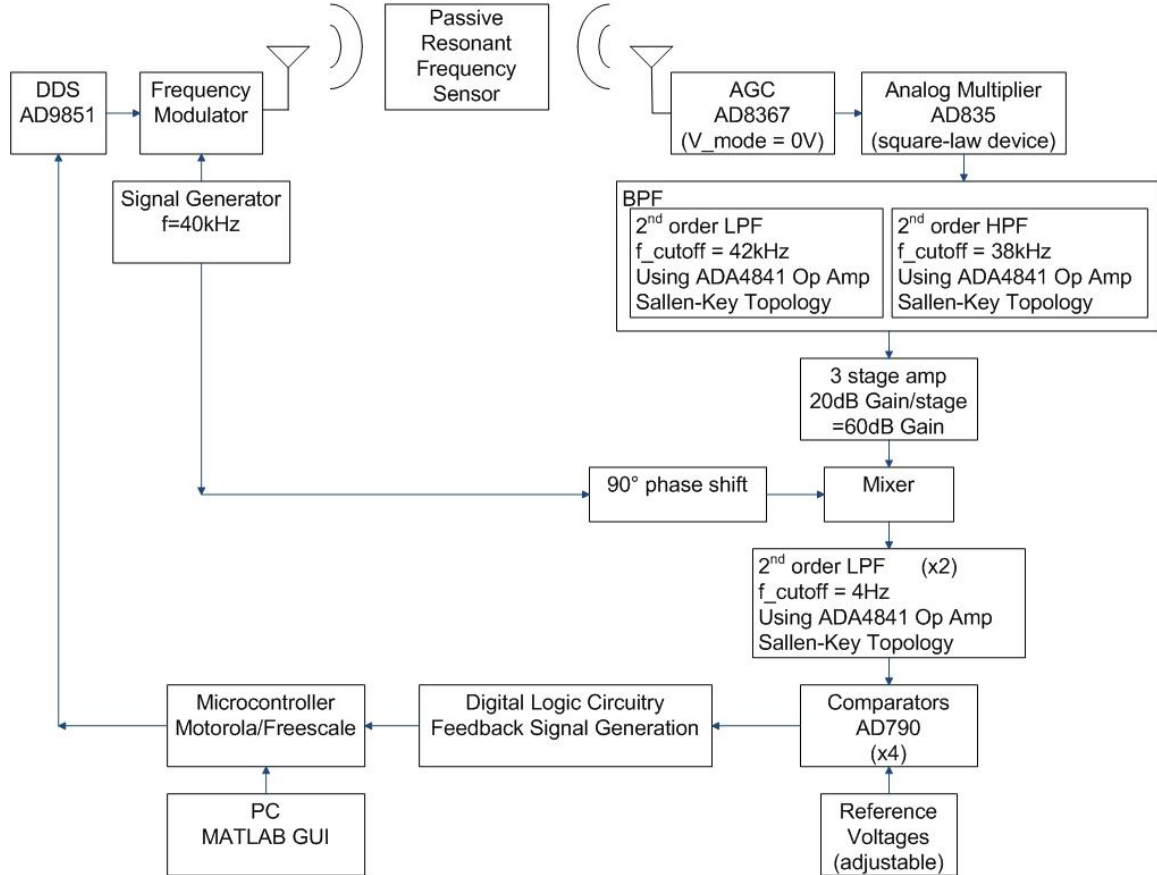


Figure 4.1: Block diagram of the wireless sensor interrogator design.

MATLAB Simulink Simulation

In Figure 4.2 the MATLAB Simulink model of the wireless sensor interrogator is shown. Differences between the Simulink model and the design as shown in block diagram form in Figure 4.1 are discussed next.

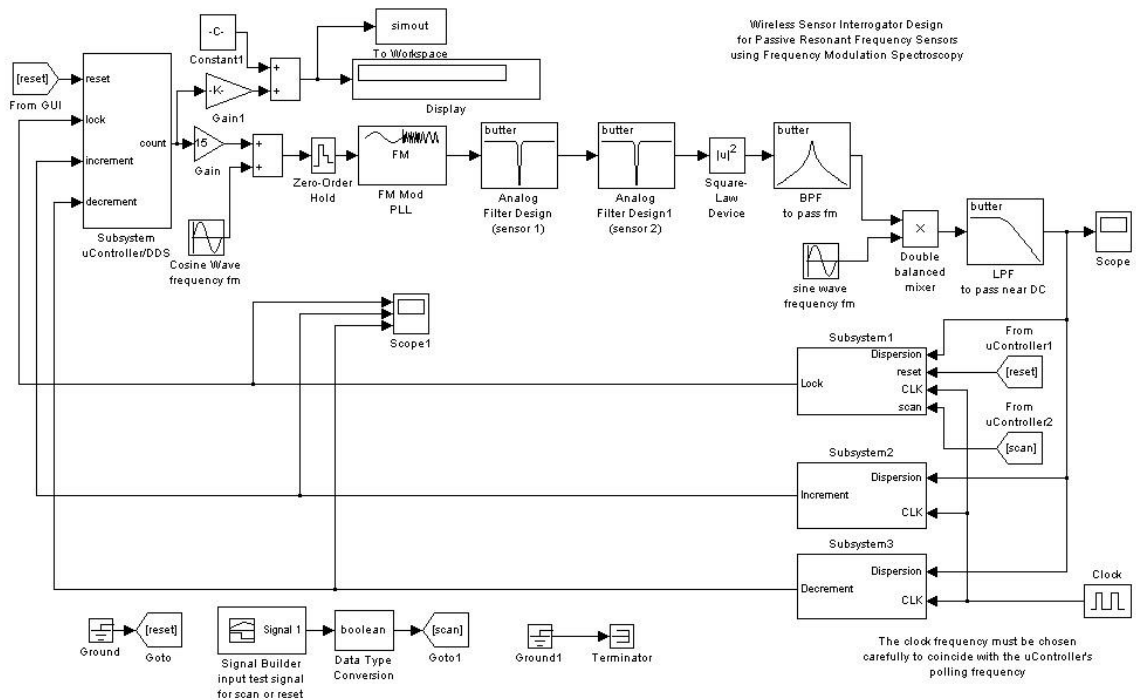


Figure 4.2: MATLAB Simulink model of the wireless sensor interrogator.

Differences Between the Wireless Sensor Interrogator Design and the MATLAB Simulink Model

Due to the length of simulation time required for running the wireless sensor interrogator design in Simulink the modified PLL frequency modulator is simply modeled with a built-in block from the Simulink communications blockset. The block modulates using frequency modulation, but does not model any of the PLLs characteristics such as bandwidth or nonlinearity. This greatly simplifies the computational burden of the simulation without compromising the simulation results when operating within the frequency range of interest.

MATLAB Simulink Simulation Results

The results of running frequency sweep simulations with and without the feedback signals are reported in this section. The feedback design approach is discussed in Chapter three. The carrier frequency of the transmitted FM signal was swept from 800 kHz to 1200 kHz for these simulations. This frequency range is the chosen frequency range of interest for both the simulated Simulink model and the hardware implementation. The resonant frequency sensor model has a resonant frequency of 1000 kHz and a quality factor of 10.

Simulation Results Without the Sensor Model and Without the Feedback Signals Employed:

Without the feedback signals employed, the wireless interrogator model will not lock to the resonant frequency of the sensor model. However, the frequency sweep rate and frequency range can be seen, as well as the absence of the dispersion signal. The simulation results are presented in Figures 4.3 and 4.4 and show the carrier frequency sweep and the absence of the dispersion signal.

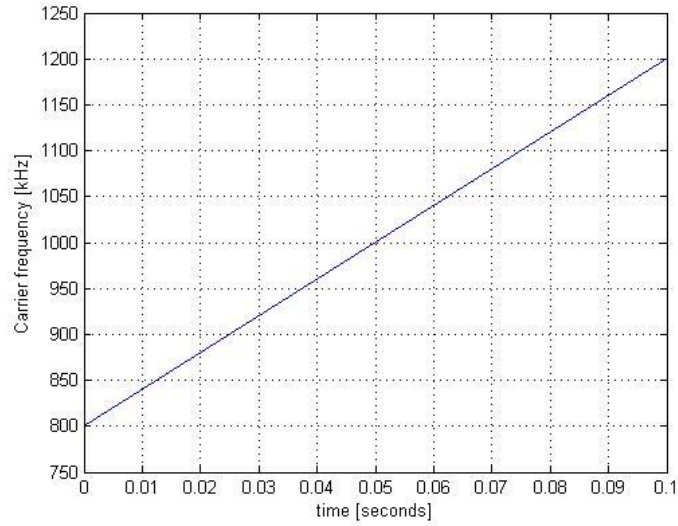


Figure 4.3: Carrier frequency data of the transmitted FM signal without the sensor model and without the feedback signals employed.

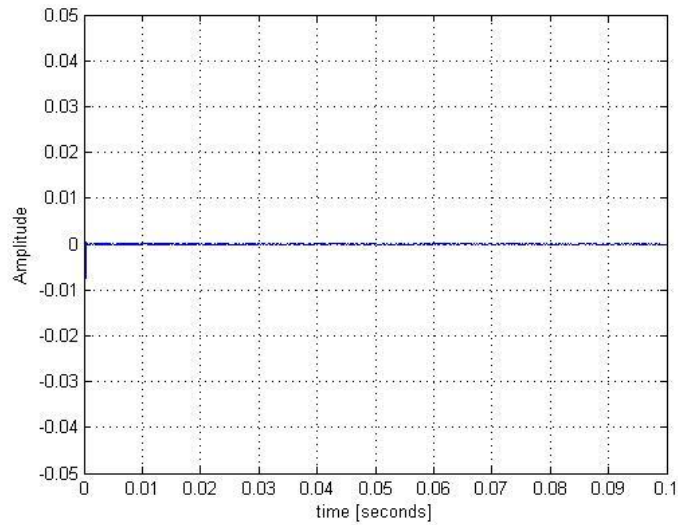


Figure 4.4: Output signal without the sensor model and without the feedback signals employed showing the absence of the dispersion signal.

Simulation Results With the Sensor Model

and Without the Feedback Signals Employed: Without the feedback signals

employed, the wireless interrogator model will not lock to the resonant frequency of the

sensor model. However, the frequency sweep rate and frequency range can be seen, as well as the output dispersion signal. The simulation results are presented in Figures 4.5 and 4.6 and show the carrier frequency sweep and the presence of the dispersion signal.

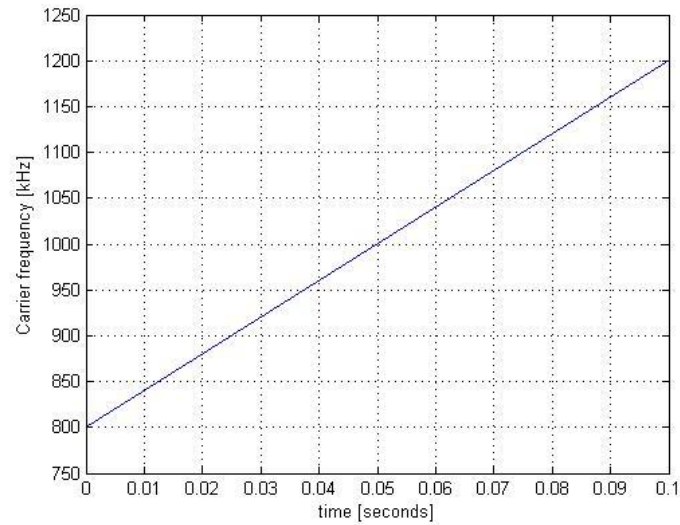


Figure 4.5: Carrier frequency data of the transmitted FM signal with the sensor model and without the feedback signals employed.

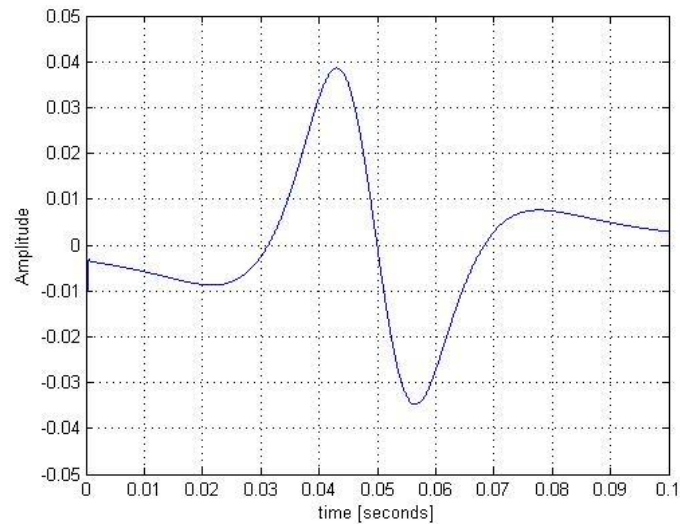


Figure 4.6: Output signal with the sensor model and without the feedback signals employed showing the presence of the dispersion signal.

Simulation Results Without the Sensor Model

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator model will not lock to the resonant frequency of the sensor model if the sensor model is not included. The simulation results are presented in Figures 4.7 and 4.8 and show the carrier frequency sweep and the absence of the dispersion signal.

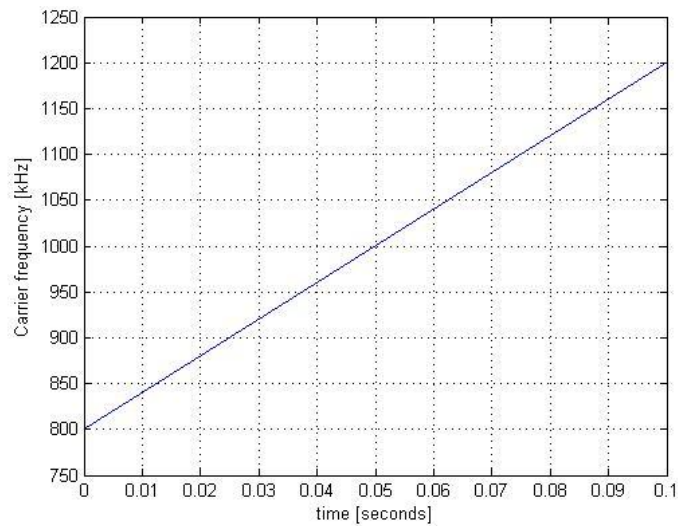


Figure 4.7: Carrier frequency data of the transmitted FM signal without the sensor model and with the feedback signals employed.

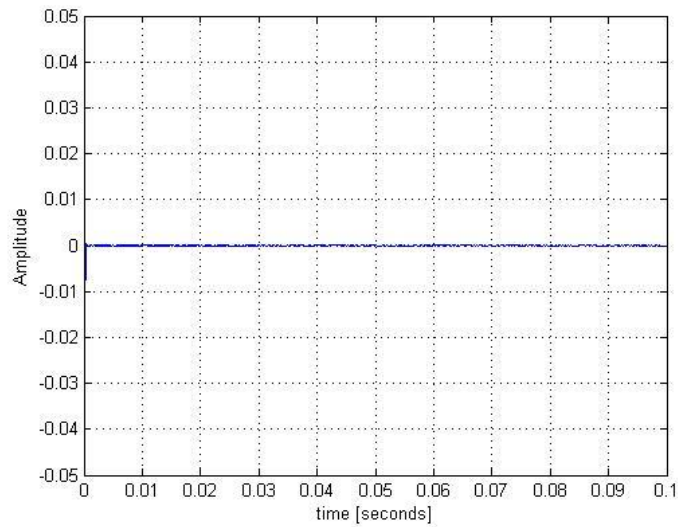


Figure 4.8: Output signal without the sensor model and with the feedback signals employed showing the absence of the dispersion signal.

Simulation Results With the Sensor Model

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator model will lock to the resonant frequency of the sensor model.

The simulation results are presented in Figures 4.9 and 4.10. Figure 4.9 shows the carrier frequency sweeping up to and locking to the resonant frequency of the sensor model.

Figure 4.10 shows the first half of the dispersion signal and then the continuing output when the interrogator model is locked to the resonant frequency of the sensor model.

When the interrogator model is locked, the carrier frequency of the transmitted FM signal equals the resonant frequency of the sensor model.

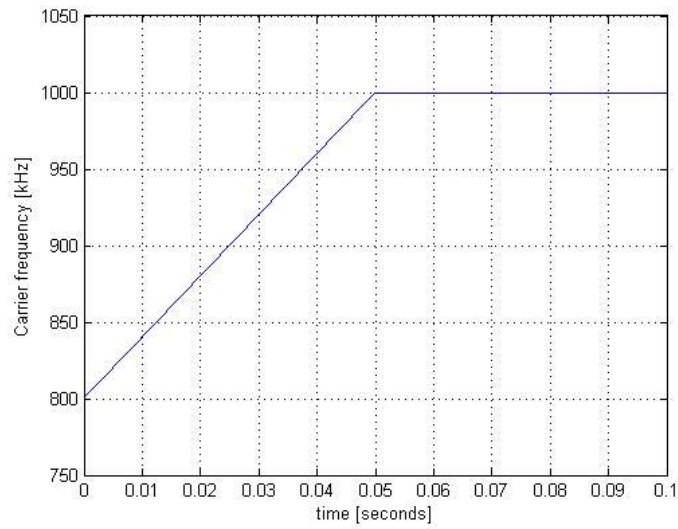


Figure 4.9: Carrier frequency data of the transmitted FM signal with the sensor model and with the feedback signals employed.

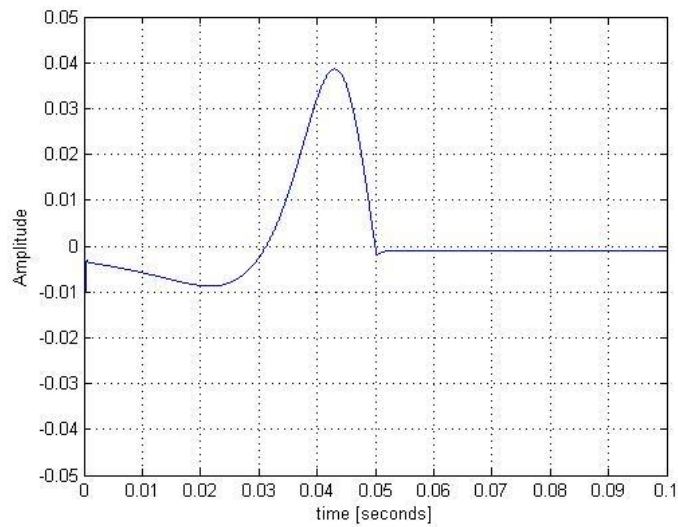


Figure 4.10: Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the sensor model.

Wireless Interrogator Design Hardware Implementation

The wireless interrogator design presented in Chapter three was implemented in hardware. The results of the hardware implementation are presented next.

Hardware Implementation Results with a Hard-wired LC Resonant Sensor

The results of running frequency sweep experiments on the hardware implementation of the wireless interrogator design with a hard-wired LC resonant sensor are reported in this section. The carrier frequency of the transmitted FM signal was swept from 800 kHz to 1200 kHz for these experiments. This is the same frequency range used in the Simulink simulations. In lieu of a MEMS sensor, LC resonant circuits are used to test the wireless interrogator. Henceforth, the LC resonant circuits used to test the wireless interrogator will be called sensors. The hard-wired resonant frequency sensor has a resonant frequency of 1005 kHz and a quality factor of 10. As with the Simulink simulations, the hardware experiments are performed with and without the hard-wired sensor and with and without the feedback signals employed.

Hardware Results Without the Hard-wired LC Sensor

and Without the Feedback Signals Employed: Without the feedback signals employed, the wireless interrogator will not lock to the resonant frequency of the hard-wired sensor. However, the frequency sweep rate and frequency range can be seen, as well as the absence of the dispersion signal. The hardware results are presented in Figures 4.11 and 4.12 and show the carrier frequency sweep and the absence of the dispersion signal.

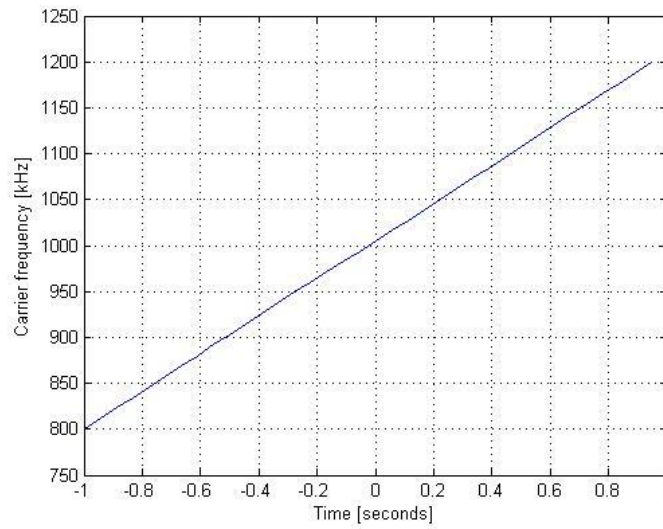


Figure 4.11: Carrier frequency data of the transmitted FM signal without the hard-wired sensor and without the feedback signals employed.

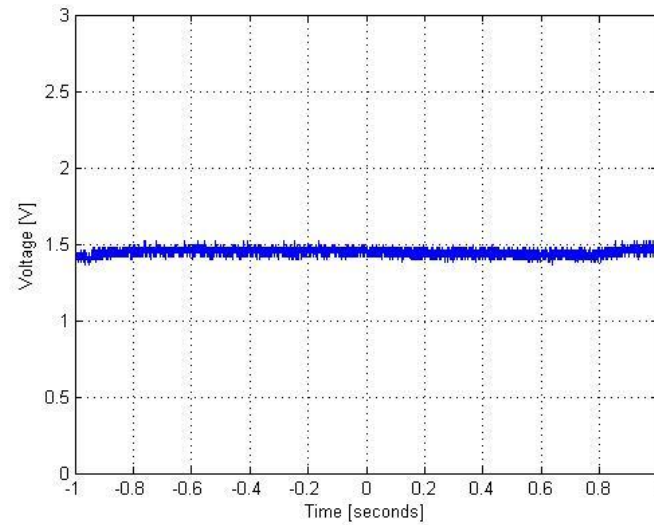


Figure 4.12: Output signal without the hard-wired sensor and without the feedback signals employed showing the absence of the dispersion signal.

**Hardware Results With the Hard-wired Sensor
and Without the Feedback Signals Employed:** Without the feedback signals

employed, the wireless interrogator will not lock to the resonant frequency of the hard-

wired sensor. However, the frequency sweep rate and frequency range can be seen, as well as the output dispersion signal. The hardware results are presented in Figures 4.13 and 4.14 and show the carrier frequency sweep and the presence of the dispersion signal.

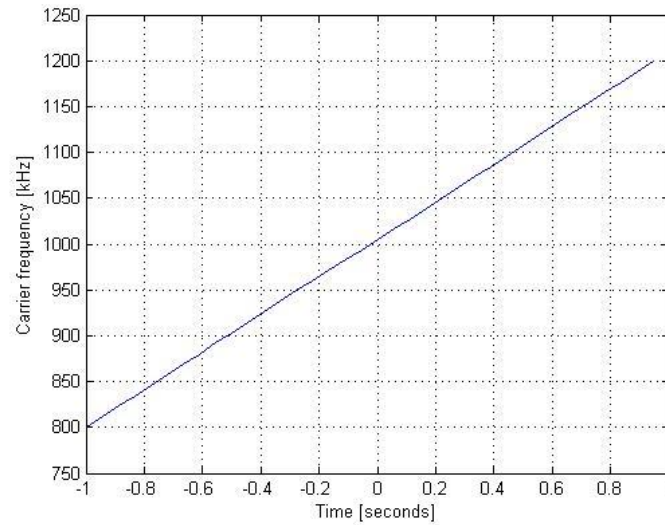


Figure 4.13: Carrier frequency data of the transmitted FM signal with the hard-wired sensor and without the feedback signals employed.

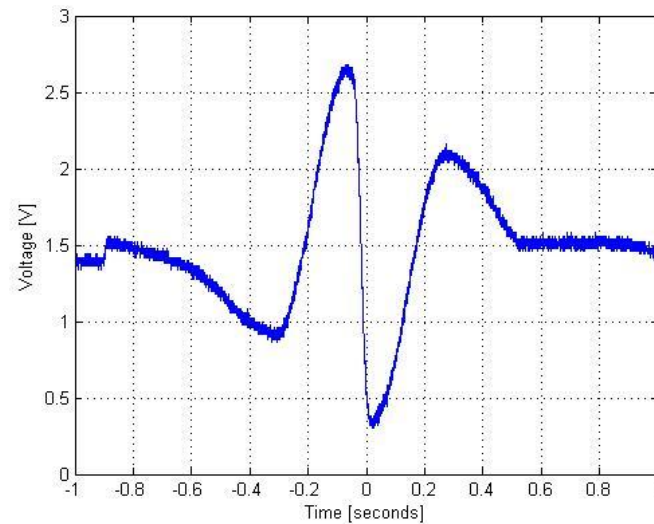


Figure 4.14: Output signal with the hard-wired sensor and without the feedback signals employed showing the presence of the dispersion signal.

Hardware Results Without the Hard-wired Sensor

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator will not lock to the resonant frequency of the hard-wired sensor if the hard-wired sensor is not included. The hardware results are presented in Figures 4.15 and 4.16 and show the carrier frequency sweep and the absence of the dispersion signal.

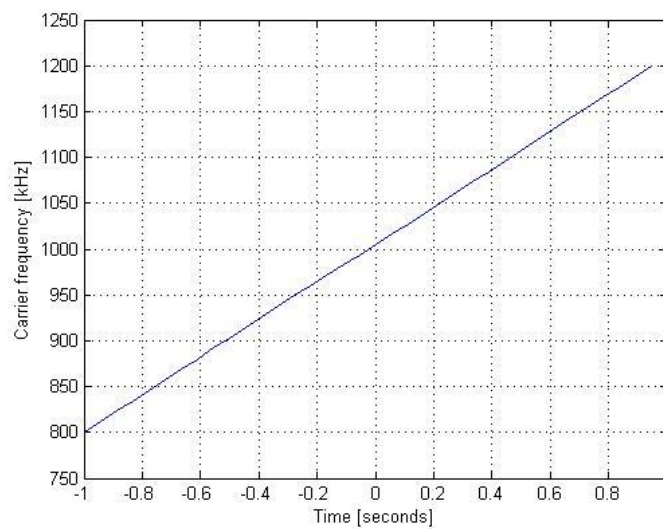


Figure 4.15: Carrier frequency data of the transmitted FM signal without the hard-wired sensor and with the feedback signals employed.

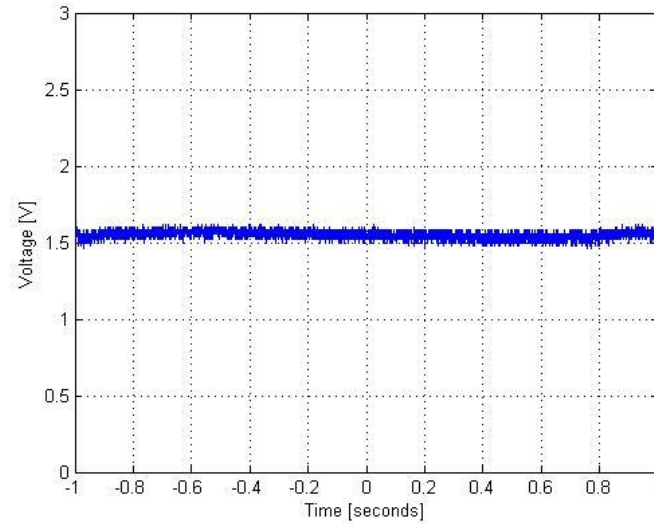


Figure 4.16: Output signal without the hard-wired sensor and with the feedback signals employed showing the absence of the dispersion signal.

Hardware Results With the Hard-wired Sensor

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator will lock to the resonant frequency of the hard-wired sensor.

The hardware results are presented in Figures 4.17 and 4.18. Figure 4.17 shows the carrier frequency sweeping up to and locking to the resonant frequency of the hard-wired sensor. Figure 4.18 shows the first half of the dispersion signal and then the continuing output when the interrogator is locked to the resonant frequency of the hard-wired sensor. When the interrogator is locked, the carrier frequency of the transmitted FM signal equals the resonant frequency of the hard-wired sensor.

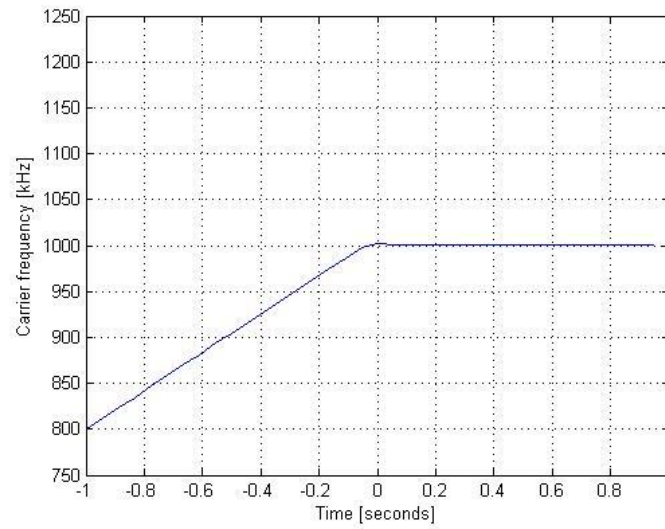


Figure 4.17: Carrier frequency data of the transmitted FM signal with the hard-wired sensor and with the feedback signals employed.

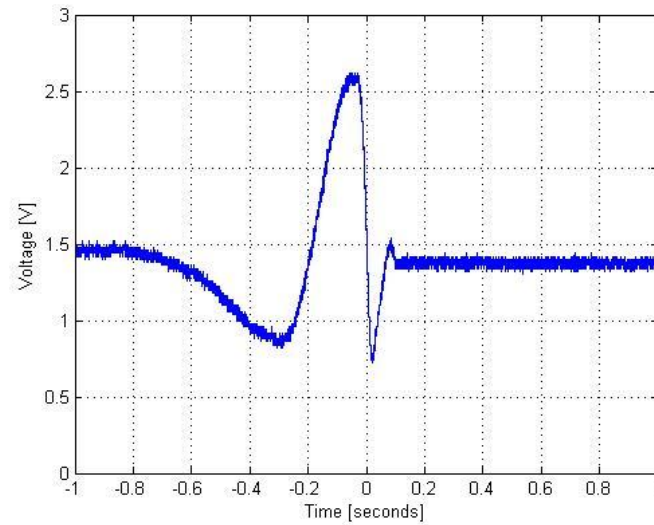


Figure 4.18: Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the hard-wired sensor.

Hardware Implementation Results with a wireless LC resonant sensor

The results of running frequency sweep experiments on the hardware implementation of the wireless interrogator design with a wireless LC resonant sensor are

reported in this section. The carrier frequency of the transmitted FM signal was swept from 800 kHz to 1200 kHz for these experiments. This is the same frequency range used in the Simulink simulations. The wireless resonant frequency sensor has a resonant frequency of 1010 kHz and a quality factor of 10. As with the Simulink simulations, the hardware experiments are performed with and without the wireless sensor and with and without the feedback signals employed.

**Hardware Results Without the Wireless LC Sensor
and Without the Feedback Signals Employed:** Without the wireless sensor

included, the wireless interrogator will not lock to the resonant frequency of a wireless sensor. However, the frequency sweep rate and frequency range can be seen, as well as the absence of the dispersion signal. The hardware results are presented in Figures 4.19 and 4.20 and show the carrier frequency sweep and the absence of the dispersion signal.

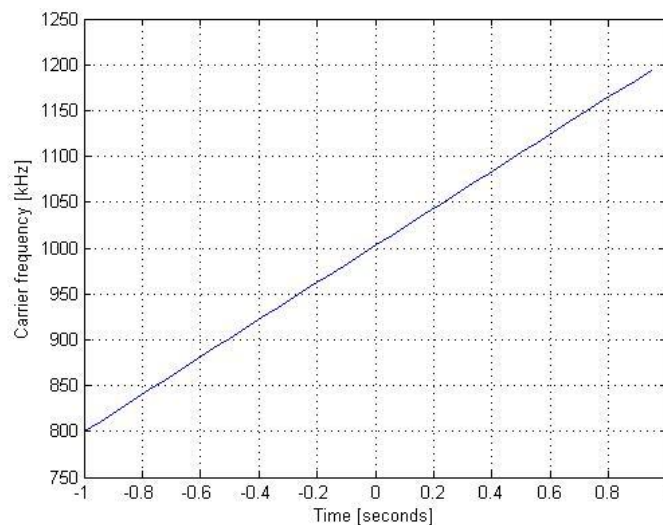


Figure 4.19: Carrier frequency data of the transmitted FM signal without the wireless sensor and without the feedback signals employed.

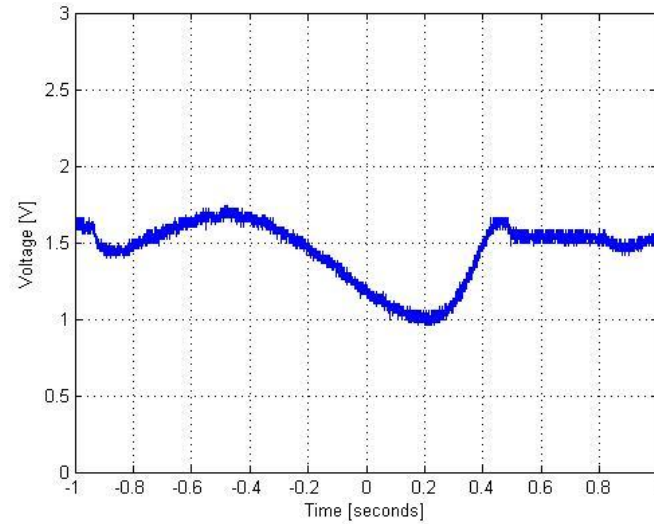


Figure 4.20: Output signal without the wireless sensor and without the feedback signals employed showing the absence of the dispersion signal.

**Hardware Results With the Wireless Sensor
and Without the Feedback Signals Employed:** Without the feedback signals

employed, the wireless interrogator will not lock to the resonant frequency of the wireless sensor. However, the frequency sweep rate and frequency range can be seen, as well as the output dispersion signal. The hardware results are presented in Figures 4.21 and 4.22 and show the carrier frequency sweep and the presence of the dispersion signal.

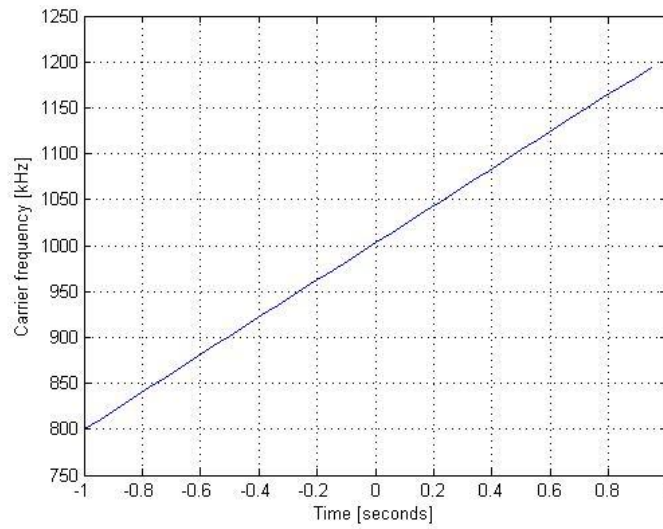


Figure 4.21: Carrier frequency data of the transmitted FM signal with the wireless sensor and without the feedback signals employed.

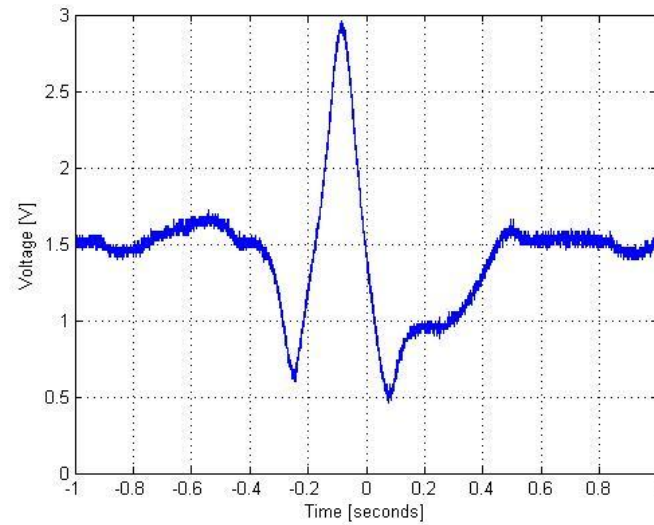


Figure 4.22: Output signal with the wireless sensor and without the feedback signals employed showing the presence of the dispersion signal.

Hardware Results Without the Wireless Sensor

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator will not lock to the resonant frequency of the wireless sensor if

the wireless sensor is not included. The hardware results are presented in Figures 4.23 and 4.24 and show the carrier frequency sweep and the absence of the dispersion signal.

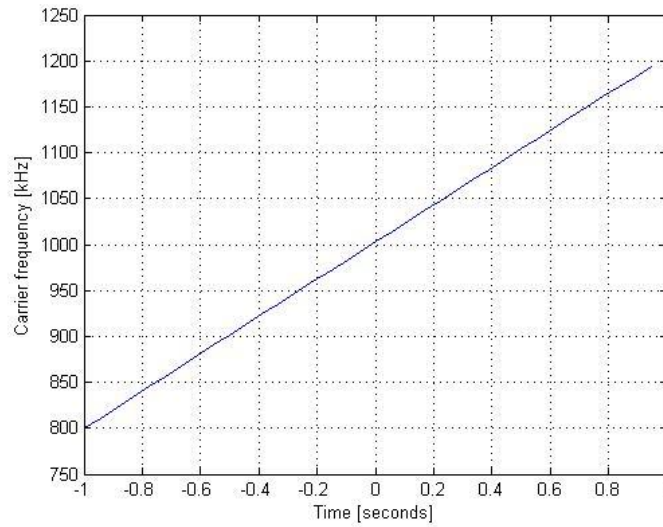


Figure 4.23: Carrier frequency data of the transmitted FM signal without the wireless sensor and with the feedback signals employed.

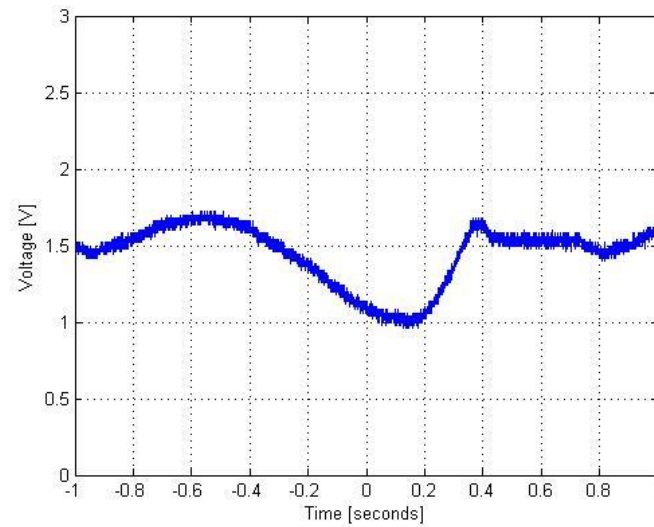


Figure 4.24: Output signal without the wireless sensor and with the feedback signals employed showing the absence of the dispersion signal.

Hardware Results With the Wireless Sensor

and With the Feedback Signals Employed: With the feedback signals employed,

the wireless interrogator will lock to the resonant frequency of the wireless sensor. The hardware results are presented in Figures 4.25 and 4.26. Figure 4.25 shows the carrier frequency sweeping up to and locking to the resonant frequency of the hard-wired sensor. Figure 4.26 shows the first half of the dispersion signal and then the continuing output when the interrogator is locked to the resonant frequency of the wireless sensor. When the interrogator is locked, the carrier frequency of the transmitted FM signal equals the resonant frequency of the hard-wired sensor.

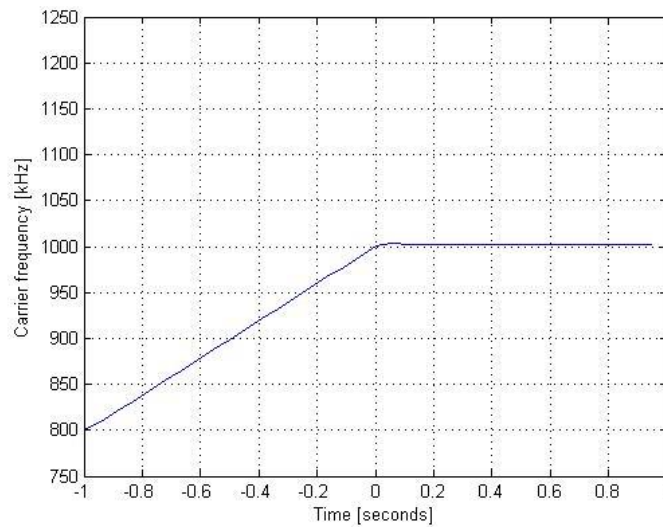


Figure 4.25: Carrier frequency data of the transmitted FM signal with the wireless sensor and with the feedback signals employed.

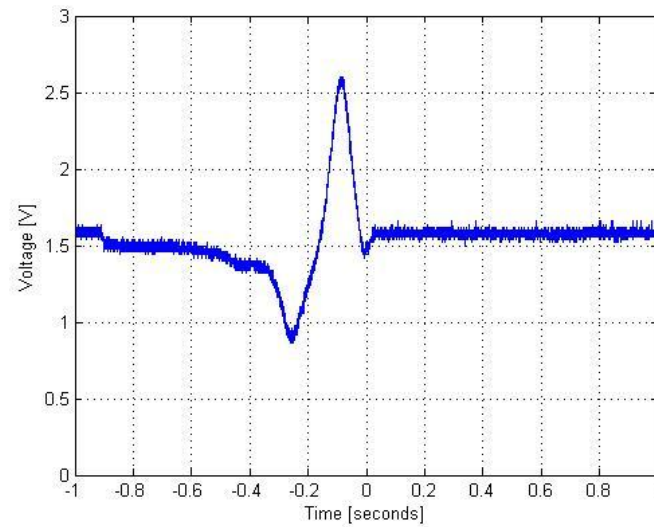


Figure 4.26: Output signal showing the presence of the first half of the dispersion signal and the wireless interrogator locking to the resonant frequency of the wireless sensor.

Additional Wireless Interrogator Results

As presented in earlier chapters, the wireless sensor interrogator operates by sweeping the carrier frequency to find and lock on to the resonant frequency of the wireless sensor. However, if there is another wireless sensor nearby with a different resonant frequency, it would be useful for the interrogator to have an option to cause continued scanning to search for another resonant frequency sensor. As described in Chapter three, the interrogator design includes a scan button in the MATLAB GUI that enables the user to continue scanning for more sensors once the system has already locked to a sensor. This design feature worked successfully in simulation to continue scanning and to lock to a second resonant frequency sensor model. The hardware results from testing the scan functionality using two wireless resonant frequency sensors are shown in Figures 4.27-29. As shown in these figures, this design feature worked successfully in hardware. Figure 4.27 shows the interrogator sweeping and locking to the

920 kHz resonant frequency of the first sensor and then at approximately 1.8 seconds, the scan button is pushed. The interrogator continues sweeping and locks to the 1010 kHz resonant frequency of the second sensor. Figure 4.28 shows the partial dispersion signals for each of the two sensors as the interrogator locks to the sensors.

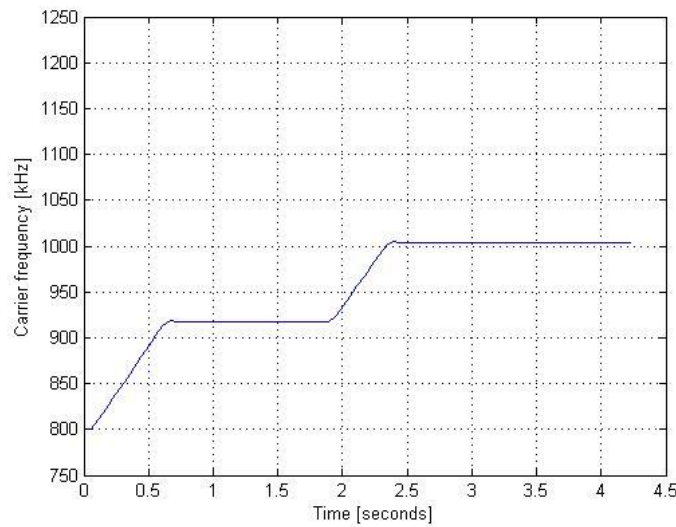


Figure 4.27: Carrier frequency data from a simple test of the scan functionality of the wireless interrogator.

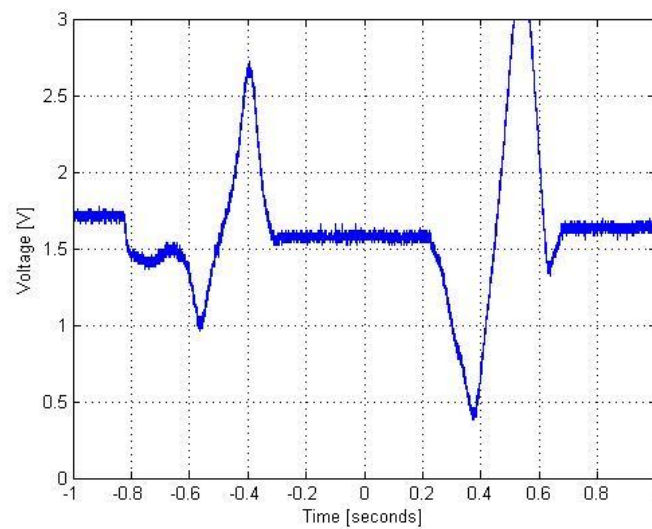


Figure 4.28: The output signal showing the partial dispersion signals as the wireless interrogator locks to each of two sensors.

Figure 4.29 shows a second hardware test of the scan functionality of the interrogator. Two wireless sensors are repeatedly locked to and then the scan button is pressed in the MATLAB GUI. As seen in Figure 4.29, the frequency range of the wireless sensor interrogator is 800 kHz to 1200 kHz by design. When the wireless sensor interrogator is searching for a sensor and comes to the upper limit of this frequency range, 1200 kHz, then the carrier frequency is automatically reset to the starting frequency of 800 kHz. The wireless sensor interrogator is capable of sensing one or more resonant frequency sensors in this frequency range.

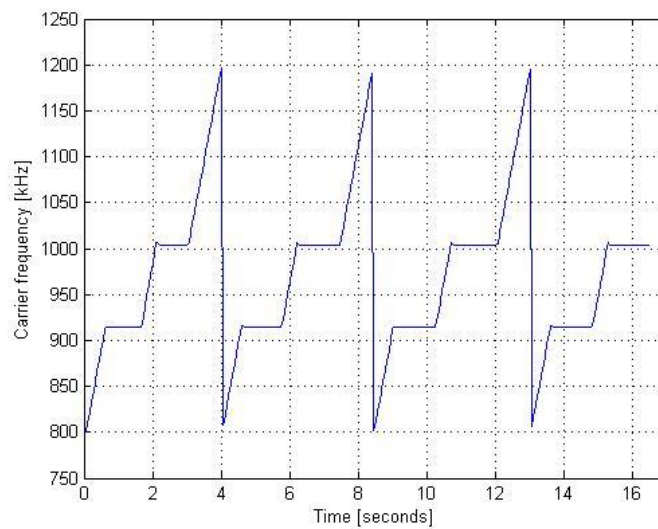


Figure 4.29: Carrier frequency data from a second test of the scan functionality of the wireless sensor interrogator.

The maximum antenna separation distance while still being able to lock to the resonant frequency of a passive sensor was found to be 29 inches. This distance could be increased with an increase in the transmitted signal power.

Design Improvements

Although the Simulink simulation results and the hardware implementation results with a hard-wired sensor appear effective for accomplishing frequency modulation spectroscopy with passive resonant frequency sensors, the added complexity of incorporating the antenna resonances for modeling wireless operation presents significant additional design effort and challenges.

Antenna Modeling and Design

In order for an antenna to transmit a signal, the antenna must resonant at or near the frequency of the signal being sent. This antenna resonance is not currently modeled in the MATLAB Simulink model of the wireless interrogator design. Since the sensing technique relies on the phase shifts of the frequency components of the transmitted signal, any affect of the antenna resonances on the transmitted or received signals should be carefully considered.

The antennas used in this design are off the shelf AM antennas that have been modified to broaden their resonances. The wireless interrogator could be designed to operate in a different frequency range of interest, in which case different antennas might need to be used. Although both the transmit and receive antennas could be identical, different antenna designs could be used for the transmit and receive antennas. In

addition, the antennas could be designed from scratch or selected from commercially available options.

The Digital Divide

Digital signal processing offers precision and repeatability that is much more difficult to attain with analog signal processing. In addition, the digital portion of the design, whether implemented in a microcontroller or an FPGA, can be modified and reprogrammed usually without changing the hardware configuration. By reprogramming the microcontroller or FPGA, new design ideas can be tested out in hardware with minimal additional cost.

Currently, this wireless interrogator design changes to digital after the comparator outputs. This is by no means an optimal solution. What other portions of the design could be implemented digitally? Perhaps after the dispersion signal, which is a very low frequency signal, a high quality ADC could be employed. In this case, the voltage comparisons with the dispersion signal could be implemented digitally, offering improved precision and ease of adjustment of the reference voltage levels. Another option might be to employ an ADC prior to the dispersion signal. For example, depending on the processing speed of the digital portion of the system and the sampling rate of the ADC, the signal output from the AGC block could be input to an ADC and the entire received signal processing could be implemented in an FPGA.

Conclusions

With a hard-wired sensor the wireless interrogator operates as designed. With a wireless sensor the hardware implementation is functional, but the antenna resonances affect the quality of the output dispersion signal. For this design to be more robust, more effort would need to be focused on antenna design and modeling. Suggestions for future work and project conclusions are discussed in Chapter five.

CHAPTER FIVE

SUMMARY AND RECOMMENDATIONS
FOR FUTURE RESEARCHSummary

The motivation for this research work has been to develop a sensitive sensing technique for the detection of passive, resonant frequency sensors. The optical measurement technique called frequency modulation spectroscopy (FMS) has been reviewed in detail and then applied to a non-optical electronics design. This wireless interrogator utilizing FMS has been designed, modeled in MATLAB's Simulink, implemented in hardware, performance tested, and presented in this thesis. The functionality of the hardware implementation was verified for wirelessly detecting passive LC resonators.

A design of a passive wireless sensor including both an electrical resonance coupled with a mechanical resonance is not the focus of this thesis, and therefore was not presented. However, the benefits of such a micro-electrical-mechanical-system (MEMS) sensor are discussed and it is noted that separate but related research work on such a sensor is being carried out at the University of Alabama in Huntsville.

The results from the hardware implementation suggest that, with a MEMS sensor, the wireless interrogator would be a useful and viable sensor system. The sensor system could be applied to the detection of chemical or biological agents at airport security checkpoints, border crossings, or anywhere people or shipments of goods can pass

between two antennas. In addition to homeland security, the sensor system could also be applied to scientific measurement applications.

Recommendations for Future Research

There are several areas in this wireless interrogator design that could be improved. First, the implementation of a DSP version of the signal processing in this wireless interrogator design would result in better repeatability. The decision of where to go digital, as discussed in Chapter four, should be carefully examined. The design currently goes digital following the comparator outputs. One possibility would be to input the dispersion signal to an ADC and perform all of the voltage comparisons digitally. If an FPGA was included in the design, the comparators could be implemented in the FPGA making the reference voltage levels easily and precisely adjustable. In addition, the finite state machine used to generate the LOCK feedback signal could be implemented in the same FPGA, thereby reducing the physical size of the hardware.

The greatest flaw in the MATLAB Simulink modeling of the wireless interrogator design is the absence of any modeling of the antenna resonances. Therefore modeling of the antennas could be included in the Simulink model, and any resulting changes to the hardware implementation could be made. Additionally, the antennas used in the wireless interrogator hardware implementation are by no means optimal. An investigation into the best antenna design could be undertaken, possibly including custom as well as separate antenna designs for both the transmit antenna and the receive antenna.

Finally, incorporating a MEMS sensor with the wireless interrogator model and hardware implementation would complete the sensor system. For this wireless sensor system to be manufactured commercially, a target market would need to be identified and an appropriate chemical or biological receptor material for the underside of the MEMS sensor (on the piezoelectric membrane) would have to be developed.

REFERENCES CITED

- [1] J. Won, S. H. Choa, and Y. L. Zhao, "An integrated sensor for pressure, temperature, and relative humidity based on MEMS technology," *Journal of Mechanical Science and Technology*, Vol. 20, No. 4, pp. 505-512, April 2006.
- [2] M. Sasaki, M. Tabata, T. Haga, and K. Hane, "Piezoresistive rotation angle sensor integrated in micromirror," *Japanese Journal of Applied Physics, Part 1 – Regular Papers Brief communications & Review Papers*, Vol. 45, No. 4B, pp. 3789-3793, April 2006.
- [3] B. S. Davis, T. Denison, and J. Kuang, "A monolithic high-g SOI-MEMS accelerometer for measuring projectile launch and flight accelerations," *Shock and Vibration*, Vol. 13, No. 2, pp. 127-135, 2006.
- [4] M. C. Hsieh, Y. K. Fang, M.S. Ju, G. S. Chen, J. J. Ho, C. H. Yang, P. M. Wu, G. S. Wu, and T. Y. F. Chen, "A contact-type piezoresistive micro-shear stress sensor for above-knee prosthesis application," *Journal of Microelectromechanical Systems*, Vol. 10, No. 1, pp. 121-127, March 2001.
- [5] C. L. Britton, Jr., R. J. War;mack, S. F. Smith, P. I. Oden, R. L. Jones, T. Thundat, G. M. Brown, W. L. Bryan, J. C. Depriest, M. N. Ericson, M. S. Emery, M. R. Moore, G. W. Turner, A. L. Wintenberg, jT. D. Threatt, Z. Hu, L. G. Clonts, and J. M Rochelle, "Battery-powered, Wireless MEMS Sensors for High-Sensitivity Chemical and Biological Sensing," Presented at the Symposium on Advanced Research in VLSI, March 1999.
- [6] D. Kong, T. Mei, Y. Tao, L. Ni, T. Zhang, W. Lu, Z. Zhang, and R. Wang, "A MEMS Sensor Array for Explosive Particle Detection," *Proceedings of the 2004 International Conference on Information Acquisition*, pp. 278-281, June 2004.
- [7] G. C. Bjorklund, "Frequency-modulation spectroscopy: a new method for measuring weak absorptions and dispersions," *Optical Society of America, Optics Letters*, Vol. 5, No. 1, January 1980.
- [8] G. C. Bjorklund and M. D. Levenson, "Frequency Modulation (FM) Spectroscopy Theory of Lineshapes and Signal-to-Noise Analysis," *Applied Physics B – Photophysics and Laser Chemistry*, Vol. 32, pp. 145-152, 1983.
- [9] E. A. Whittaker and G. C. Bjorklund, "Absorption measurements using frequency modulation heterodyne spectroscopy," *Proceedings of SPIE*, Vol. 426, pp. 81-86, 1983.
- [10] M. Gehrtz and G. C. Bjorklund, "Quantum-limited laser frequency-modulation spectroscopy," *Journal of the Optical Society of America B*, Vol. 2, No. 9, September 1985.

- [11] J. A. Silver, "Frequency-modulation spectroscopy for trace species detection: theory and comparison among experimental methods," *Applied Optics*, Vol. 31, No. 6, pp. 707-717, February 1992.
- [12] J. M. Supplee, E. A. Whittaker, and W. Lenth, "Theoretical description of frequency modulation and wavelength modulation spectroscopy," *Applied Optics*, Vol. 33, No. 27, pp. 6294-6302.
- [13] C. Liu, *Foundations of MEMS*, Prentice Hall., 2006.
- [14] J. G. Proakis and M. Salehi, *Fundamentals of Communication Systems*, Prentice Hall, 2005.

APPENDICES

APPENDIX A

DERIVATION OF THE ABSORPTION AND DISPERSION OUTPUT SIGNALS
USING FREQUENCY MODULATION SPECTROSCOPY

Note: Analytic signal representations are used in this derivation. Thus, signals are represented using complex exponentials with non-negative frequencies.

Frequency Modulation Spectroscopy (FMS) begins with a sinusoidal carrier signal:

$$c(t) = A_c \cos(2\pi f_c t) \quad [A.1]$$

The carrier signal is frequency modulated by a constant frequency sinusoidal “message” signal [14]:

$$m(t) = a \cos(2\pi f_m t) \quad [A.2]$$

The frequency modulated (FM) signal is given by:

$$s(t) = A_c \cos(2\pi f_c t + \phi(t)) \quad [A.3]$$

where

$$\begin{aligned} \phi(t) &= 2\pi k_f \int_{-\infty}^t m(\tau) d\tau \\ &= 2\pi k_f \int_{-\infty}^t a \cos(2\pi f_m \tau) d\tau \\ &= \frac{2\pi k_f a}{2\pi f_m} \sin(2\pi f_m t) \\ &= \frac{k_f a}{f_m} \sin(2\pi f_m t) \end{aligned} \quad [A.4]$$

The parameter k_f is the frequency deviation constant [14]. Therefore the FM signal $s(t)$ becomes:

$$s(t) = A_c \cos\left(2\pi f_c t + \frac{k_f a}{f_m} \sin(2\pi f_m t)\right) \quad [A.5]$$

The frequency modulation index β can be defined as:

$$\beta = \frac{k_f a}{f_m} \quad [\text{A.6}]$$

The FM signal $s(t)$ now becomes:

$$s(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t)) \quad [\text{A.7}]$$

In signal processing, it is sometimes convenient to represent signals using complex exponentials. Sinusoidal signals can be thought of as having both a positive frequency component and a negative frequency component by using complex exponentials. Although this appears at first glance to complicate the situation, the use of Euler's equation and complex exponentials can actually ease the burden of mathematical computation.

Euler's equation states:

$$e^{jx} = \cos x + j \sin x \quad [\text{A.8}]$$

Therefore

$$A \cos x = \text{Re}(Ae^{jx}) \quad [\text{A.9}]$$

Using Euler's relation, the transmitted or sent signal $s(t)$ can be written as

$$\begin{aligned} s(t) &= A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t)) \\ &= \text{Re}(A_c e^{j(2\pi f_c t + \beta \sin(2\pi f_m t))}) \\ &= \text{Re}(A_c e^{j2\pi f_c t} e^{j\beta \sin(2\pi f_m t)}) \end{aligned}$$

Since $\sin(2\pi f_m t)$ is periodic with period $T_m=1/f_m$, the same is true for the complex exponential signal

$$e^{j\beta \sin(2\pi f_m t)}$$

Therefore, it can be expanded in a Fourier-series representation. The Fourier-series coefficients are obtained from the integral

$$\begin{aligned} C_n &= \frac{1}{T_m} \int_0^{T_m} e^{j\beta \sin(2\pi f_m t)} e^{-jn 2\pi f_m t} dt \\ &= f_m \int_0^{\frac{1}{f_m}} e^{j\beta \sin(2\pi f_m t)} e^{-jn 2\pi f_m t} dt \\ &= \frac{1}{2\pi} \int_0^{2\pi} e^{j\beta (\sin(u) - nu)} du \end{aligned}$$

$$\text{Where } u = 2\pi f_m t$$

This latter expression is a well known integral called the Bessel function of the first kind of order n and is denoted by $J_n(\beta)$:

$$C_n = J_n(\beta)$$

Therefore, we have the Fourier-series for the complex exponential as:

$$\begin{aligned} e^{j\beta \sin(2\pi f_m t)} &= \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_m t} \\ &= \sum_{n=-\infty}^{\infty} J_n(\beta) e^{j2\pi n f_m t} \end{aligned}$$

Therefore, $s(t)$ becomes:

$$s(t) = \text{Re}(A_c e^{j2\pi f_c t} e^{j\beta \sin(2\pi f_m t)})$$

$$\begin{aligned}
&= \text{Re}(A_c \sum_{n=-\infty}^{\infty} J_n(\beta) e^{j2\pi n f_m t} e^{j2\pi f_c t}) \\
&= \text{Re}(\sum_{n=-\infty}^{\infty} A_c J_n(\beta) e^{j(2\pi f_c t + n 2\pi f_m t)})
\end{aligned}$$

Using the analytic form of the complex exponential signals and using the radian frequencies:

$$\begin{aligned}
\omega_c &= 2\pi f_c \\
\omega_m &= 2\pi f_m
\end{aligned}$$

The FM signal $s(t)$ becomes:

$$s(t) = \sum_{n=-\infty}^{\infty} A_c J_n(\beta) e^{j(\omega_c + n \omega_m)t}$$

The preceding relation shows that the frequency modulated signal contains all frequencies of the form $f_c + n f_m$ for $n=0, \pm 1, \pm 2, \dots$. Therefore, the theoretical bandwidth of the modulated signal is infinite. However, the amplitude of the sinusoidal components of frequencies $f_c \pm n f_m$ for large n is very small. Depending on the frequency modulation index β , only a limited number of the frequency components will be significant. Frequency Modulation Spectroscopy (FMS) is based on changes in the amplitude and phase of the frequency components of $s(t)$, specifically the carrier and first sidebands. Therefore, the frequency modulation index β is chosen to maximize the product of the carrier and first sidebands. The magnitude of the carrier is given by:

$$A_c J_0(\beta)$$

The magnitudes of the first upper sideband and of the first lower sideband are equal:

$$|A_c J_{-1}(\beta)| = |A_c J_1(\beta)|$$

A_c is constant, therefore it is sufficient to maximize $J_0(\beta)*J_1(\beta)$. The optimal frequency modulation index β is found to be 1.08. Thus $s(t)$ consists of a strong carrier and two weak sidebands, therefore only the $n=-1,0,1$ terms corresponding to the lower sideband, carrier, and upper sideband are considered. All other frequency components are assumed to be insignificant. Therefore the transmitted FM signal $s(t)$ is:

$$s(t) = A_c [J_{-1}(1.08)e^{j(w_c - w_m)t} + J_0(1.08)e^{jw_c t} + J_1(1.08)e^{j(w_c + w_m)t}]$$

The FM signal $s(t)$ is wirelessly transmitted while linearly ramping the carrier frequency w_c through the frequency band of interest. The transmitted FM signal $s(t)$ is inductively coupled to the passive resonant frequency sensor. The frequency components of the transmitted signal experience both an amplitude attenuation and phase shift due to the passive resonant frequency sensor. It is useful to express the sensor transfer function T as an amplitude attenuation and phase shift that is frequency component dependent. Therefore, the sensor transfer function is given by:

$$T_n = e^{-\delta_n - j\phi_n} = e^{-\delta_n} e^{-j\phi_n}$$

where $e^{-\delta_n}$ represents an amplitude attenuation of the n^{th} frequency component and $e^{-j\phi_n}$ represents a phase shift of the n^{th} frequency component. After passing through the passive resonant frequency sensor, the received signal $r(t)$ is given by:

$$\begin{aligned} r(t) &= A_c T_{-1} J_{-1}(1.08)e^{j(w_c - w_m)t} + A_c T_0 J_0(1.08)e^{jw_c t} + A_c T_1 J_1(1.08)e^{j(w_c + w_m)t} \\ &= A_c J_{-1}(1.08)e^{-\delta_{-1} - j\phi_{-1}} e^{j(w_c - w_m)t} + A_c J_0(1.08)e^{-\delta_0 - j\phi_0} e^{jw_c t} + A_c J_1(1.08)e^{-\delta_1 - j\phi_1} e^{j(w_c + w_m)t} \\ &= A_c J_{-1}(1.08)e^{-\delta_{-1}} e^{j((w_c - w_m)t - \phi_{-1})} + A_c J_0(1.08)e^{-\delta_0} e^{j(w_c t - \phi_0)} + A_c J_1(1.08)e^{-\delta_1} e^{j((w_c + w_m)t - \phi_1)} \end{aligned}$$

The received signal $r(t)$ is then passed through a square-law device. The output of the square-law device is the magnitude squared of the input.

$$\begin{aligned}
|r(t)|^2 &= A_c^2 (J_{-1}(1.08))^2 e^{-2\delta_{-1}} e^{-j2\phi_{-1}} + A_c^2 (J_0(1.08))^2 e^{-2\delta_0} e^{-j2\phi_0} + A_c^2 (J_1(1.08))^2 e^{-2\delta_1} e^{-j2\phi_1} \\
&\quad + A_c^2 J_{-1}(1.08) J_0(1.08) e^{-\delta_{-1}} e^{-j\phi_{-1}} e^{-\delta_0} e^{j\phi_0} e^{j\omega_m t} + A_c^2 J_0(1.08) J_1(1.08) e^{-\delta_0} e^{-j\phi_0} e^{-\delta_1} e^{j\phi_1} e^{j\omega_m t} \\
&\quad + 2\omega_c \text{ terms}
\end{aligned}$$

Simplifying assumptions:

1. Terms containing $(J_{-1}(1.08))^2$ or $(J_1(1.08))^2$ are insignificant compared to the other terms and can be neglected
2. $2\omega_c$ terms will be filtered out, and can be neglected
3. The phase of the DC terms can be neglected

$$\begin{aligned}
|r(t)|^2 &= A_c^2 (J_0(1.08))^2 e^{-2\delta_0} + A_c^2 J_{-1}(1.08) J_0(1.08) e^{-\delta_{-1}} e^{-j\phi_{-1}} e^{-\delta_0} e^{j\phi_0} e^{j\omega_m t} \\
&\quad + A_c^2 J_0(1.08) J_1(1.08) e^{-\delta_0} e^{-j\phi_0} e^{-\delta_1} e^{j\phi_1} e^{j\omega_m t} \\
&= A_c^2 e^{-2\delta_0} \left\{ (J_0(1.08))^2 + [J_{-1}(1.08) J_0(1.08) e^{\delta_0 - \delta_{-1}} e^{-j\phi_{-1}} e^{j\phi_0} \right. \\
&\quad \left. + J_0(1.08) J_1(1.08) e^{\delta_0 - \delta_1} e^{-j\phi_0} e^{j\phi_1}] e^{j\omega_m t} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ (J_0(1.08))^2 + [-J_1(1.08) J_0(1.08) e^{\delta_0 - \delta_{-1}} e^{j(\phi_0 - \phi_{-1})} \right. \\
&\quad \left. + J_0(1.08) J_1(1.08) e^{\delta_0 - \delta_1} e^{j(\phi_1 - \phi_0)}] e^{j\omega_m t} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ (J_0(1.08))^2 + J_0(1.08) J_1(1.08) [e^{\delta_0 - \delta_1} e^{j(\phi_1 - \phi_0)} - e^{\delta_0 - \delta_{-1}} e^{j(\phi_0 - \phi_{-1})}] e^{j\omega_m t} \right\}
\end{aligned}$$

82

where $J_{-1}(\beta) = -J_1(\beta)$ (see equation 4.2.7 in [14])

$$|r(t)|^2 = A_c^2 e^{-2\delta_0} \left\{ (J_0(1.08))^2 + J_0(1.08) J_1(1.08) [e^{\delta_0 - \delta_1} e^{j(\phi_1 - \phi_0)} - e^{\delta_0 - \delta_{-1}} e^{j(\phi_0 - \phi_{-1})}] e^{j\omega_m t} \right\}$$

The goal of the next few steps in this derivation is to use Euler's equation to represent the magnitude squared of the received signal in terms of sines and cosines. Transforming to a non-analytic signal representation:

$$\begin{aligned}
|r(t)|^2 &= A_c^2 e^{-2\delta_0} \left\{ \begin{aligned} &(J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})}]0.5e^{jw_mt} \\ &+ J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]0.5e^{-jw_mt} \end{aligned} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ \begin{aligned} &(J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})} + e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})}]0.25e^{jw_mt} \\ &+ J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})} + e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]0.25e^{-jw_mt} \end{aligned} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ \begin{aligned} &(J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} + e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})} \\ &\quad + e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} - e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})} + e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]0.25e^{jw_mt} \\ &+ J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} + e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_1}e^{j(\phi_0-\phi_{-1})} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})} \\ &\quad - e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} + e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} + e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]0.25e^{-jw_mt} \end{aligned} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ \begin{aligned} &(J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} + e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_{-1}}e^{j(\phi_0-\phi_{-1})} - e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]* \\ &\quad [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \\ &+ J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}e^{j(\phi_1-\phi_0)} - e^{\delta_0-\delta_1}e^{-j(\phi_1-\phi_0)} - e^{\delta_0-\delta_1}e^{j(\phi_0-\phi_{-1})} + e^{\delta_0-\delta_{-1}}e^{-j(\phi_0-\phi_{-1})}]* \\ &\quad [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \end{aligned} \right\} \\
&= A_c^2 e^{-2\delta_0} \left\{ \begin{aligned} &(J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}(e^{j(\phi_1-\phi_0)} + e^{-j(\phi_1-\phi_0)}) - e^{\delta_0-\delta_{-1}}(e^{j(\phi_0-\phi_{-1})} + e^{-j(\phi_0-\phi_{-1})})]* \\ &\quad [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \\ &+ J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1}(e^{j(\phi_1-\phi_0)} - e^{-j(\phi_1-\phi_0)}) - e^{\delta_0-\delta_1}(e^{j(\phi_0-\phi_{-1})} - e^{-j(\phi_0-\phi_{-1})})]* \\ &\quad [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \end{aligned} \right\}
\end{aligned}$$

$$= A_c^2 e^{-2\delta_0} \left\{ \begin{array}{c} (J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} 2\cos(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} 2\cos(\phi_0-\phi_{-1})]^* \\ [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \\ + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} 2\sin(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} 2\sin(\phi_0-\phi_{-1})]^* \\ [0.25e^{jw_mt} + 0.25e^{-jw_mt}] \end{array} \right\}$$

$$= A_c^2 e^{-2\delta_0} \left\{ \begin{array}{c} (J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} \cos(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \cos(\phi_0-\phi_{-1})]^* \\ [0.5e^{jw_mt} + 0.5e^{-jw_mt}] \\ + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} \sin(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \sin(\phi_0-\phi_{-1})]^* \\ [0.5e^{jw_mt} + 0.5e^{-jw_mt}] \end{array} \right\}$$

$$= A_c^2 e^{-2\delta_0} \left\{ \begin{array}{c} (J_0(1.08))^2 + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} \cos(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \cos(\phi_0-\phi_{-1})][\cos w_mt] \\ + J_0(1.08)J_1(1.08)[e^{\delta_0-\delta_1} \sin(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \sin(\phi_0-\phi_{-1})][\sin w_mt] \end{array} \right\}$$

$$= A_c^2 e^{-2\delta_0} \left\{ \begin{array}{c} (J_0(1.08))^2 + [e^{\delta_0-\delta_1} \cos(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \cos(\phi_0-\phi_{-1})]J_0(1.08)J_1(1.08)\cos w_mt \\ + [e^{\delta_0-\delta_1} \sin(\phi_1-\phi_0) - e^{\delta_0-\delta_{-1}} \sin(\phi_0-\phi_{-1})]J_0(1.08)J_1(1.08)\sin w_mt \end{array} \right\}$$

This equation corresponds well to equation 3 from Bjorklund, “Frequency-modulation spectroscopy: a new method for measuring weak absorptions and dispersions” Optics Letters, January 1980. Please note that the * used in the above equations denotes multiplication.

If $|\delta_0 - \delta_1|, |\delta_0 - \delta_{-1}|, |\phi_0 - \phi_1|, |\phi_0 - \phi_{-1}| \ll 1$ then the equation above simplifies to

$$|r(t)|^2 = A_c^2 e^{-2\delta_0} \{ (J_0(1.08))^2 + (\delta_{-1} - \delta_1) J_0(1.08) J_1(1.08) \cos(w_m t) + (\phi_1 + \phi_{-1} - 2\phi_0) J_0(1.08) J_1(1.08) \sin(w_m t) \}$$

Approximations used:

For $\alpha \ll 1$

$$e^\alpha = 1 + \alpha$$

$$\cos(\alpha) = 1$$

$$\sin(\alpha) = \alpha$$

The assumptions $|\delta_0 - \delta_1|, |\delta_0 - \delta_{-1}|, |\phi_0 - \phi_1|, |\phi_0 - \phi_{-1}| \ll 1$ are all valid for weak absorptions and dispersions, the type encountered in wireless sensing of passive resonant frequency sensors.

Thus, the output of the square-law device will contain a heterodyne beat signal at the modulation frequency w_m if $\delta_{-1} - \delta_1 \neq 0$ or if $\phi_1 + \phi_{-1} - 2\phi_0 \neq 0$. The heterodyne beat signal arises from the upper and lower sidebands beating against the carrier.

57

If the modulation frequency w_m is chosen to be large compared to the spectral feature of interest, then the spectral feature can be probed by a single sideband. In this case, the amplitude attenuations and phase shifts experienced by the carrier and lower sideband remain almost constant. If we let $\delta_{-1} = \delta_0 = \bar{\delta}$ and $\phi_{-1} = \phi_0 = \bar{\phi}$, where $\bar{\delta}$ and $\bar{\phi}$ are constant background amplitude attenuation and phase shift respectively, then the equation above can be written as

$$|r(t)|^2 = A_c^2 e^{-2\delta_0} [J_0^2(1.08) + \Delta\delta J_0(1.08) J_1(1.08) \cos(w_m t) + \Delta\phi J_0(1.08) J_1(1.08) \sin(w_m t)]$$

where $\Delta\delta$ and $\Delta\phi$ are the changes from the constant background amplitude attenuation and phase shift. The $\cos(w_m t)$ and $\sin(w_m t)$ components of the heterodyne beat signal are, respectively, proportional to the absorption and dispersion from the

spectral feature of the passive resonant frequency sensor. The absorption signal is due to $\Delta\delta$, the change from the constant background amplitude attenuation. The dispersion signal is due to $\Delta\phi$, the change from the constant background phase shift.

The absorption output signal is formed by frequency mixing $\cos(2\pi f_m t)$ with the 'cos' portion of $|r(t)|^2$ and low pass filtering the result. The dispersion output signal is formed by frequency mixing $\sin(2\pi f_m t)$ with the 'sin' portion of $|r(t)|^2$ and low pass filtering the result. The output absorption and dispersion time waveforms are shown below.

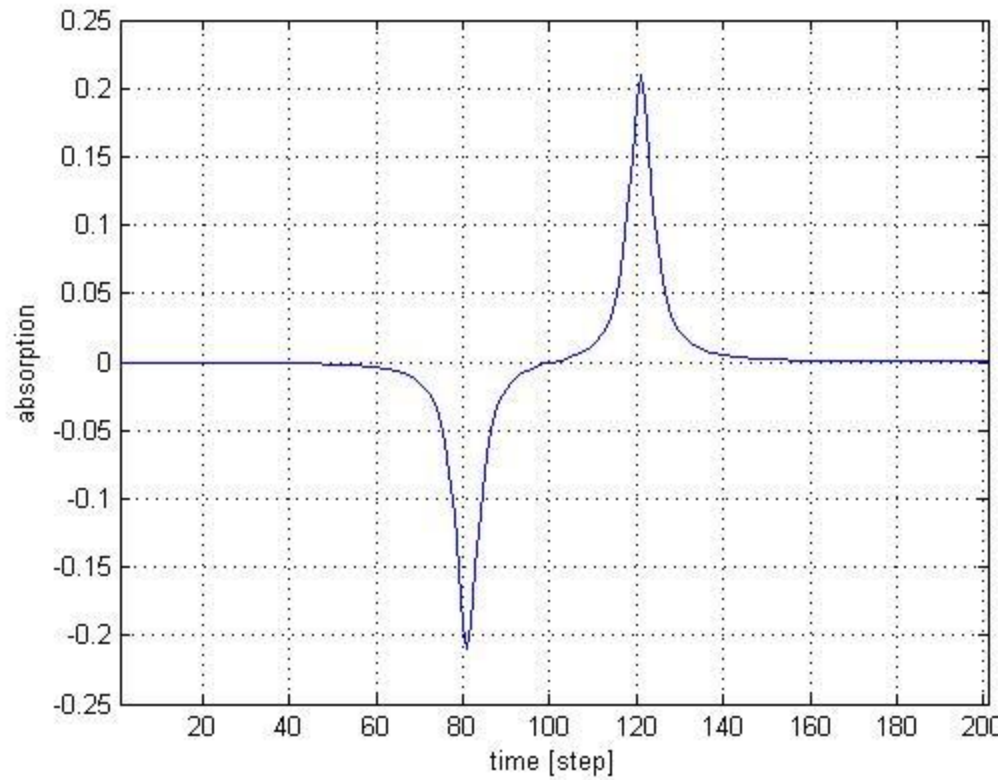


Figure A.1: Plot of output absorption signal.

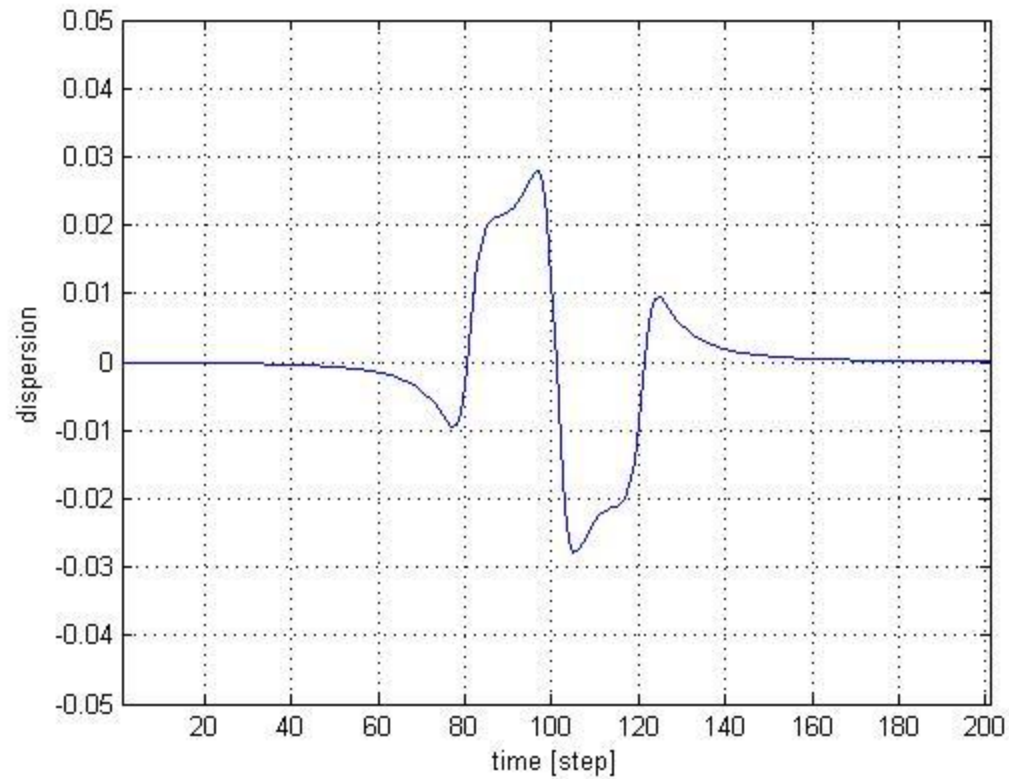


Figure A.2: Plot of output dispersion signal.

The carrier frequency, f_c , of the transmitted FM signal $s(t)$ equals the resonant frequency of the passive sensor near time step 100. The output absorption signal shown in Figure A.1 does not have a sharp zero crossing near time step 100. The output

dispersion signal shown in Figure A.2 has a sharp, negative slope zero-crossing when the carrier frequency is equal to the resonant frequency of the sensor. This sharp zero-crossing is used to measure the resonant frequency of the sensor, and thus the physical parameter being sensed (temperature, pressure, etc.).

APPENDIX B

MICROCONTROLLER SOURCE FILES

```

// *****
// ad9851.c
// ~~~~~
// Note: This collection of code uses the ad9851 chip in conjunction
// with the DDS-60 daughtercard
//
// Written by: Justin Hadella ( August 25, 2007 )
//
// The DDS-60 daughtercard has a ad9851 chip hard-wired for serial
// programming. The board includes built-in voltage regulation, as
// well as output amplification and filtering
//
// DDS-60 Pinout
//
//          ( TOP VIEW )
//          -----
//          |         |
//          | LOAD  --|-1 | AD  | |   Pin 1 - ad9851 FQ_UD
//          | CLOCK --|-2 |9851| |   Pin 2 - SPI SCLK
//          | DATA --|-3 |    | |   Pin 3 - SPI MOSI ( D7 on ad9851 )
//          | +5 V  --|-4 |-----|
//          | GND   --|-5 |         |   Pin 5 - GND
//          | RF OUT --|-6 |         |   Pin 6 - filtered output
//          | (NC)  --|-7 |         |
//          | 8-12 V --|-8 |         |   Pin 8 - DC voltage +8 to +12 V
//          |         | (POT) |
//          |         |-----|
//          |         |
//
// Note: the signal called "CLOCK" on the DDS-60 board is called
//       'SCLK' in the code. There is another signal called LOCK
//       which looks too similar. Careless typing could cause an
//       error.
//
// *****

#include <mc9s12c32.h>
#include "ad9851.h"
#include "hcs12_common.h"

// Initializes the ad9851 chip on the DDS-60 daughtercard
void ad9851_init(void)
{
    // initialize SPI data direction registers
    DATA_DDR = OUTPUT;
    LOAD_DDR = OUTPUT;
    SCLK_DDR = OUTPUT;

    // initialize the pulse out signals
    LOCKOUT_DDR = OUTPUT;
    SCANOUT_DDR = OUTPUT;
    LOCK_OUT = SCAN_OUT = 0;

    // Stage 1
    // -----
    // ad9851 chip is in parallel mode by default. The dds-60 card is hard-wired
    // for serial use, but it needs to be loaded in using CLOCK and LOAD

    // probe CLOCK to ensure ad9851 in serial mode
    SCLK = 1;
    asm(nop);    // BUGFIX: keep line to ensure the probe takes place

```

```

SCLK = 0;

// probe LOAD to load hard-coded bits into register
LOAD = 1;
asm(nop);    // BUGFIX: keep line to ensure the probe takes place
LOAD = 0;

// Stage 2
// -----
// ad9851 now in serial mode, init SPI and clear the ad9851 register

// enable SPI
init_SPI();

// write 32 zeroes to ad9851
SPI_write(0x00);
SPI_write(0x00);
SPI_write(0x00);
SPI_write(0x00);

// load in control word ( 0x01 enables 6x PLL, serial mode )
SPI_write(0x01);

// probe LODE to latch the 40-bit value just written
LOAD = 1;
asm(nop);    // BUGFIX: keep line to ensure the probe takes place
LOAD = 0;

// Define the three external control signals as inputs
LOCK_DDR = INPUT;
INC_DDR  = INPUT;
DEC_DDR  = INPUT;
}

// This function produces a tone at the frequency requested. On the DDS-60
// daughterboard, there is a 30 MHz oscillator. The ad9851 chip is set at
// the 6x PLL mode, so it has a clock rate of 180 MHz. The frequency code
// word is a fractional percentage of 180 MHz.
void ad9851_setfreq(uint32 freq)
{
    // points to the character to write
    byte *b;

    // Note: the frequency is stored in a 32-bit number. SPI_write can only
    //       transmit one byte at a time. Therefore, pointer manipulation is
    //       used to extract the individual bytes. This will cause a compiler
    //       warning, but it's OK

    // point to the MSB of freq
    b = &freq;

    // Note: the HCS12 stores number in a big-endian format

    SPI_write(*(b+3));    // send LSB
    SPI_write(*(b+2));
    SPI_write(*(b+1));
    SPI_write(*b);        // send MSB
    SPI_write(0x01);      // send command code ( 6x PLL, serial mode )

    // probe LOAD to indicate done writing

```

```
    LOAD = 1;  
    asm(nop);    // BUGFIX: keep line to ensure the probe takes place  
    LOAD = 0;  
}
```

```

// *****
// hcs12_common.c
// ~~~~~
// Note: This collection of code uses the hcs12 microcontroller in
//       conjunction with the CSM-12C32 module
//
// Written by: Justin Hadella ( August 15, 2007 )
//
// The code included are common and useful functions used in many
// types of programs
//
// Currently, the functions are related to the SCI and SPI modules
//
// *****

#include "hcs12_common.h"

// SCI related functions
// ~~~~~
// Note: In order to use the SCI module, init_SCI() must be called first

// Initializes the SCI module
void init_SCI(void) {

    // SCI control register 1
    // 1-start, 8-data, 1-stop (all defaults ok)

    // SCI control register 2
    // enable TX, RX, RX full interrupt
    SCICR2 = SCICR2_RE_MASK | SCICR2_TE_MASK | SCICR2_RIE_MASK;

    // SCI baud rate registers
    SCIBDH = 0;
    SCIBDL = 13; // baud rate = 38400

    // Other common baud rates:
    // baudrate   code
    // 38400      13
    // 19200      26
    // 9600       52

    // formula:
    //          Bus Clock
    // SCI baudrate = -----
    //          16 * SCIBR
}

// TX character using SCI
char putchar(char b) {

    // make sure TX register is empty
    while(SCISR1_TDRE == 0) continue;

    SCIDRL = b;

    // return value allows for checking things like \0 when
    // printing strings
    return b;
}

```

```

// RX character using SCI
char getchar(void) {
    byte temp;

    // while data is not ready
    while(SCISR1_RDRF == 0) continue;

    // clear SCI flags
    temp = SCISR1;
    temp = SCIDRL;

    // get the character
    return SCIDRL;
}

// Simple version of printf
// prints a null terminated string using SCI
void_putstr(char *s) {
    while(putchar(*s++));
}

// (same as above) + a carriage return and line feed
void_putstr_n(char *s) {
    while(putchar(*s++));
    putchar(CR);
    putchar(LF);
}

// Resets cursor to home position on a new line
void_newline(void) {
    putchar(CR);
    putchar(LF);
}

// TX a hexadecimal byte using SCI
//
// input: hex value 0x00-0xFF
// output: displays the discrete digits of the byte
//
// for example: 0x14 would display "14" to terminal
void_putbyte(byte b) {
    putchar(hex2ascii((b & 0xF0)>>4)); // display upper digit
    putchar(hex2ascii(b & 0x0F));      // display lower digit
}

// Converts a single digit hex value 0-F to the ASCII equivalent
//
// input: a digit 0-F
// output: ascii for that digit
char_hex2ascii(byte b) {
    if ( b < 9 )
        return (b | 0x30);
    else
        return ((b - 9) | 0x40);
}

// Displays 'n' characters in memory
// Note: useful for printing the contents of an array to a terminal
//       wo/ using a debugger
void_putarr(char *ptr, char n) {

```

```

    char i;

    for(i = 0; i < n; i++)
        putbyte(*ptr++);
}

// Displays 'n' bytes in memory
// Note: the values within are unaltered binary values.
void putraw(char *ptr, char n) {
    char i;

    for(i = 0; i < n; i++)
        putchar(*ptr++);
}

// SPI related functions
// ~~~~~
// Note: In order to use the SPI module, init_SPI() must be called first

// Initializes the SPI module
void init_SPI(void) {

    // enable SPI module, set in master mode, enable SS_L, LSB writing mode
    SPICR1 = SPICR1_SPE_MASK | SPICR1_MSTR_MASK | SPICR1_SS0E_MASK |
    SPICR1_LSBFE_MASK;
    SPICR2 = SPICR2_MODFEN_MASK;

    // set SPI sclk to 4 MHz
    SPIBR = 0;

    // Other SPI baud rate values:
    // baud rate    SPIBR value
    // 4.000 MHz    0
    // 2.000 MHz    1
    // 1.000 MHz    2
    // 500.0 kHz    3
}

// TX single byte using SPI
void SPI_write(volatile byte data) {
    volatile byte temp;

    // write data to SPI data register
    SPIDR = data;

    // wait until transfer is complete
    while(SPISR_SPIF == 0) continue;

    // clear the SPI flags
    temp = SPISR;
    temp = SPIDR;
}

void blink_LED1(void) {
}

```

```

// *****
// AD9851 Mode Test Program
// ~~~~~
// written by: Justin Hadella (September 11, 2007)
//
// This program uses the CSM-12C32 module, the DDS-60 daughtercard,
// and the Freescale Student SLK board. The following figure shows
// how the circuit is wired. The #'s represent the pins.
//
// CSM-12C32                DDS-60
// -----
// | 9 -|-----LOAD-----| - 1 8 -|---- +9 V
// | 21 -|-----SCLK-----| - 2 5 -|---- GND
// | 17 -|-----DATA-----| - 3 6 -|---- RF OUT
// |
// | 22 -|-----LOCK      }
// | 26 -|-----INCREMENT } } The external control signals
// | 28 -|-----DECREMENT }
// |
// | 18 -|-----LOCK_OUT  }
// | 20 -|-----SCAN_OUT  } Pulse signals ____| |____
// |
// | SCI =|=====RS-232--> PC
// | .
// -----
//
// The CSM-12C32 module is a MCU daughterboard used in conjunction
// with the Freescale SLK board. The CSM module hosts a 16-bit MCU,
// the hcs12. The DDS-60 daughterboard is a small board which hosts
// an Analog Devices ad9851 dds chip. The board includes power
// regulation and signal filtering. The RF OUT is adjustable via an
// integrated potentiometer.
//
// This code project and circuit works in
// conjunction with some external circuitry. There are corresponding
// MATLAB programs called "dds60.m" and "dds60_out.m" which serve as
// the visual interface to the user.
//
// *****

#include <hidef.h>
#include <mc9s12c32.h>
#pragma LINK_INFO DERIVATIVE "mc9s12c32"

#include "hcs12_common.h"
#include "ad9851.h"

#define PETERSON

// These defines represent button status in the MATLAB interface
#define RESET 7

```

```

#define SCAN      5
#define STOP      9
#define REQUEST  11

// Note: to calculate an arbitrary frequency code word, use the following:
//
//      freq code word = frequency * sysclk / 2^32
//
const double scale = 23.86092942; // == 2^32 / sysclk == 2^32 / 180 MHz
//
// for example, for a desired frequency of 1.5 MHz...
//
// freq = 1.5e6;
// freq_code = (uint32) freq * scale;

// These constants define the lower and upper bounds for sweeping
const uint32 FREQ_500K = (uint32) 500000 * scale;
const uint32 FREQ_1500K = (uint32) 1500000 * scale;

// for debugging...
const uint32 TEST1      = (uint32) 300000 * scale;
const uint32 TEST2      = (uint32) 750000 * scale;

// The global variables used in main program

//uint32 freq;                // frequency in Hz
uint32 freq_code;           // frequency code word for AD9851 (based on 180 MHz
clock)
char  status;               // represents state of MATLAB button presses
char  sweep_enable;         // enables sweep to take place - controlled by LOCK
status
byte  *f_ptr;               // pointer to freq_code - used in MATLAB
communications
uint32 start_freq;          // the lower frequency bound in sweep mode
uint32 end_freq;            // the upper frequency bound in sweep mode

void main(void) {

    // enable SW1 on CSM32 Module
    _asm("andcc #0xBF");
    INTCR = IRQ_ENABLE;

    // Initialize variables
    status = sweep_enable = 0;
    f_ptr = &freq_code;

    // Initialize the communication protocols
    init_SCI();

    // Initialize DDS-60 daughterboard
    ad9851_init();

    // Set output frequency to 0 Hz
    ad9851_setfreq((uint32) 0);

    // setup COP
    //COPCTL = COP_SETUP;

    // Reset COP before it times out
    //ARMCOP = 0x55;

```

```

//ARMCOP = 0xAA;

start:

DisableInterrupts;

// Initialize start frequency to 500 kHz
start_freq = freq_code = FREQ_500K;

// Initialize end frequency to 1.5 MHz
end_freq = FREQ_1500K;

// enable interrupts for external SCI messaging
EnableInterrupts;

while(1) {
    // wait until user presses 'RESET' button in MATLAB interface
    if ( status == RESET ) {
        status = 0;
        sweep_enable = 1;

        // Reset COP before it times out
        //ARMCOP = 0x55;
        //ARMCOP = 0xAA;
        break;
    }
}

while(1) {

    // check status of MATLAB buttons

    // If STOP is pressed at any point, stop the experiment
    if ( status == STOP )
        break;

    // If RESET is pressed, set lower frequency bound to 500 kHz and enable
    // frequency sweeping
    if ( status == RESET ) {
        status = 0;
        sweep_enable = 1;

        DisableInterrupts;
        freq_code = start_freq;
        EnableInterrupts;
    }
    // If SCAN is pressed, enable frequency sweeping. Using the logic above,
user // must press RESET before SCAN can even be acknowledged
    if ( status == SCAN ) {
        status = 0;
        sweep_enable = 1;
    }

    if ( sweep_enable ) {
        DisableInterrupts;

        // Send ad9851 the current frequency value to output
        ad9851_setfreq(freq_code);
    }
}

```

```

    // Output data to MATLAB
    //putraw(f_ptr,4);

    // Increment frequency ( 1 Hz )
    freq_code += 0x18;

    // Check bounds, if frequency > 1.5 MHz, reset to 500 KHz
    if ( freq_code > end_freq ) freq_code = start_freq;

    EnableInterrupts;
}
else {
    // In LOCK condition, check INCREMENT/DECREMENT signals

    if ( INCREMENT ) {
        DisableInterrupts;

        // Increment frequency ( 1 Hz )
        freq_code += 0x18;

        // Send ad9851 the current frequency value to output
        ad9851_setfreq(freq_code);

        EnableInterrupts;
        // Output data to MATLAB
        //putraw(f_ptr,4);
    }

    if ( DECREMENT ) {
        DisableInterrupts;

        // Decrement frequency ( 1 Hz )
        freq_code -= 0x18;

        // Send ad9851 the current frequency value to output
        ad9851_setfreq(freq_code);

        EnableInterrupts;
        // Output data to MATLAB
        //putraw(f_ptr,4);
    }
}

// Sweep enable and LOCK are opposites
if ( LOCK )
    sweep_enable = 0;
else
    sweep_enable = 1;

// Reset COP before it times out
//ARMCOP = 0x55;
//ARMCOP = 0xAA;
}

// This loop is reached only if STOP is pushed
//while(1) {}
goto start;

while(1){}
}

```

```

interrupt 20 void SCI_isr(void) {
    status = getchar();

    // if RESET button, send a pulse out on LOCK_OUT line
    if ( status == RESET ) {
        LOCK_OUT = 1;
        asm(nop);
        LOCK_OUT = 0;
    }
    // if SCAN button, send a pulse out on SCAN_OUT line
    if ( status == SCAN ) {
        SCAN_OUT = 1;
        asm(nop);
        SCAN_OUT = 0;
    }

    // REQUEST status means MATLAB is expecting data
    // send the current frequency + the value of LOCK
    if ( status == REQUEST ) {
        putraw(f_ptr,4);
        putchar(LOCK);
        status = 0;
    }
}

interrupt 5 void reset_isr(void) {
    ad9851_setfreq(TEST1);
    //main();
}

interrupt 2 void cop_isr(void) {
    ad9851_setfreq(TEST2);
    //main();
}

```

APPENDIX C

MATLAB GUI SOURCE FILE

% Much of the code was generated by the MATLAB GUIDE tool. Many of the
% comments included by this tool have been eliminated.

```
function varargout = dds60(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @dds60_OpeningFcn, ...
                  'gui_OutputFcn',    @dds60_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before dds60 is made visible.
function dds60_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for dds60
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% define the global variables used

global f;           % output file stream
global s;           % serial link to the hcs12
global t1;          % update timer for GUI controls
global t2;          % data request timer
global running;     % flag indicates 'RESET' button has been pushed
enabling           % serial and file I/O

global filename;    % name of output file
global count;       % total number of data points taken
global lock;        % condition of LOCK signal
global freq;        % the current frequency

% Initialize variables

running = 0;        % 0 == not running
count = 0;          % no data taken initially
lock = 0;

% Set up serial object
```

```

%s = serial('COM1','BaudRate',38400);
%s.BytesAvailableFcnCount = 4;
%s.BytesAvailableFcnMode = 'byte';
%s.BytesAvailableFcn = @writeFreq;

% Timer responsible for updating the GUI
%t1 = timer;
%set(t1, 'BusyMode','drop','ExecutionMode','fixedDelay','Period',1.0);
%set(t1, 'TimerFcn',{@updateGUI,handles});

% Timer responsible for sampling the data
%t2 = timer;
%set(t2, 'BusyMode','drop','ExecutionMode','fixedDelay','Period',0.5);
%set(t2, 'TimerFcn',@requestFreq);

% Set LOCK to invisible, and disable SCAN button initially
set(handles.locker,'Visible','off');
set(handles.scanButton,'Enable','off');

% --- Outputs from this function are returned to the command line.
function varargout = dds60_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in scanButton.
function scanButton_Callback(hObject, eventdata, handles)

global s;           % this event uses running flag & serial link
global running;

if running == 1
    fwrite(s,5,'uchar');    % send SCAN code to hcs12
end
return;

% --- Executes on button press in resetButton.
function resetButton_Callback(hObject, eventdata, handles)

global f;           % this event uses running flag, file stream
global s;           % & the serial link
global t1;
global t2;
global running;

if running == 0
    % Set up serial object
    s = serial('COM1','BaudRate',38400);
    s.BytesAvailableFcnCount = 4;
    s.BytesAvailableFcnMode = 'byte';
    s.BytesAvailableFcn = @writeFreq;
    fopen(s);

    f = fopen(get(handles.resultFile,'String'),'w');

```

```

    % Timer responsible for updating the GUI
    t1 = timer;
    set(t1,
'BusyMode', 'drop', 'ExecutionMode', 'fixedDelay', 'Period', 1.0);
    set(t1, 'TimerFcn', {@updateGUI, handles});

    % Timer responsible for sampling the data
    t2 = timer;
    set(t2,
'BusyMode', 'drop', 'ExecutionMode', 'fixedDelay', 'Period', 0.5);
    set(t2, 'TimerFcn', @requestFreq);

    start(t1);
    start(t2);
    running = 1;
end
fwrite(s, 7, 'uchar');    % send RESET code to hcs12

return;

% --- Executes on button press in stopButton.
function stopButton_Callback(hObject, eventdata, handles)

global f;                % this event uses the running flag, the file
stream,
global s;                % the serial link, and the data count
global t1;
global t2;
global running;
global count;

if running == 1
    fwrite(s, 9, 'uchar');    % send SCAN code to hcs12

    stop(t1);                % stop and delete the timers
    delete(t1);
    stop(t2);
    delete(t2);

    fclose(f);                % close file stream

    fclose(s);                % close serial link
    delete(s);
    clear s;

    running = 0;              % stop running
    count      % display count to terminal
    count = 0;                % reset count to 0 for future tests
end
return;

function resultFile_Callback(hObject, eventdata, handles)
return;

```

```

% --- Executes during object creation, after setting all properties.
function resultFile_CreateFcn(hObject, eventdata, handles)

if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'))
;
end

% --- Executes on a timer event
function requestFreq(hObject, eventdata, handles)
global s;

fwrite(s,11, 'uchar');    % send REQUEST code to hcs12

% Note: HCS12 will response with data - handled by writeFreq callback
return;

% --- Executes on a timer event
function updateGUI(hObject, eventdata, handles)
global lock;
global freq;

% calculate frequency in Hz and display to GUI
temp = freq / 23.86092942;
set(handles.CurrFreq, 'String', num2str(temp, '%g'));

% probe LOCK signal and take appropriate action
if lock == 1
    set(handles.locker, 'Visible', 'on');
    set(handles.scanButton, 'Enable', 'on');
else
    set(handles.locker, 'Visible', 'off');
    set(handles.scanButton, 'Enable', 'off');
end;

return;

% --- Executes when serial data received from HCS12
function writeFreq(hObject, eventdata, handles)
global f;
global s;
global count;
global lock;
global freq;

% get the raw binary data from the hcs12
temp = fread(s, 4, 'uchar');

% record state of LOCK signal
lock = fread(s, 1, 'uchar');

```

```
% record time
%time = now;

% construct a 32-bit value from individual bytes
freq = temp(4) + temp(3) * 2^8 + temp(2) * 2^16 + temp(1) * 2^24;

% write the current frequency/time variable to data file
fwrite(f,freq,'uint32');
%fwrite(f,time,'double');

% increment data count
count = count + 1;

return;
```