

FREQUENCY STABILIZATION OF AN EXTERNAL
CAVITY DIODE LASER EMPLOYING
DIGITAL CONTROL

by

Cy Taylor Drollinger

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

of

Master of Science

In

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

June 2006

© COPYRIGHT

by

Cy Taylor Drollinger

2006

All Rights Reserved

APPROVAL

of a thesis submitted by

Cy Taylor Drollinger

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

Ross Snider

Approved for the Department of Electrical Engineering

James Petersen

Approved for the Division of Graduate Education

Joseph J. Fedock

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 (paper) in whole or in parts may be granted only by the copyright holder.

Cy Taylor Drollinger

May 2006

TABLE OF CONTENTS

1. INTRODUCTION	1
2. SYSTEM ANALYSIS	4
Model Verification of the S2 FSL	5
Frequency Tunable ECDL.....	5
Analog Feedback Control.....	9
Closed Loop S2 FSL.....	11
S2 Frequency Discriminator.....	14
Open Loop Gain of the ECDL.....	16
Optimum Control Design	17
Analog Feedback Control Presently Implemented	19
Digital Control Design	20
3. DIGITAL CONTROL DESIGNS	22
Design Approach	22
Time Domain Control Design	23
Digital Linear Quadratic Regulator.....	24
Luenberger Method.....	27
Infinite Impulse Response Low Pass Filter.....	29
4. SIMULATION OF THE DLQR.....	31
Time Based Simulations of the LSE Plus DLQR.....	31
Transient Response	32
Merit of the Transient Response.....	33
Stability Versus Computational Latency.....	36
Comparing Noise Floor Voltages	37
VHDL of the LSE plus DLQR	39
Infinite Impulse Response Low Pass Filter.....	42
5. RESULTS OF FPGA DIGITAL CONTROL IMPLEMENTATION	45
Digital Frequency Down Conversion.....	45
Digital Low Pass Infinite Impulse Response Filter	48
S2 FSL with Digital IIR LPF	49
6. CONCLUSIONS	54
REFERENCES CITED.....	55

TABLE OF CONTENTS - CONTINUED

APPENDICES	56
APPENDIX A: Matlab Code S2 FSL.....	57
APPENDIX B: VHDL Digital Mixing	60
APPENDIX C: VHDL IIR LPF	68

LIST OF FIGURES

Figure	Page
1. Three representations of ECDL line widths with spectral hole burning technique. (a) Experimentally taken data for an uncontrolled ECDL, line width 10 MHz (b) Experimentally taken data for a controlled ECDL, line width 10 kHz (c) Math representation of the controlled and uncontrolled spectral holes burned into the wide absorption of the spectral hole burning material.....	2
2. Block diagram of the photonics of a spatial spectral material frequency stabilized ECDL.....	4
3. Experimental setup, which measured the frequency response of the current driver plus the ECDL.....	6
4. Experimentally measured frequency response of the current driver plus the ECDL.....	7
5. A representation of the optical transmission through a 2GHz Fabry-Perot Cavity coupled into a photodetector, which yielded the cavity's free spectral range.....	8
6. Block diagram of the experimental setup measuring the frequency response of the complete electronic feedback control.....	10
7. The frequency response of the complete analog feedback presently used to frequency stabilize an ECDL at Spectrum Lab.....	11
8. This experimental setup measured the closed loop frequency response of the S2 FSL.....	12
9. The frequency response of the closed loop S2 FSL.....	13
10. Block diagram representing the closed loop S2 FSL with three subsystems	14
11. Calculated frequency response of the S2 sensor material.....	15
12. The closed loop model of the S2 FSL compensated with the digital controller designed with time domain control theory.....	29

LIST OF FIGURES - CONTINUED

Figure	Page
13. The magnitude frequency response of the LPF IIR filter calculated with the freqz command in Matlab with the coefficients designed by the FDATAOOL.....	30
14. The LP IIR filter block diagram implemented as a direct form II model.....	30
15. The Closed loop System as modeled in Simulink. The block diagram represents the S2 FSL controlled by the LSE plus DLQR. The error signal was simulated with a voltage signal added into the output of the S2 FSL. The suppression of the error signal was calculated as the power in db in reference to 1 watt. The calculation was 10 times the log base 10 of the square of the voltage output.....	32
16. The closed loop transient response of the S2 FSL simulation. (a) A voltage step function added to the output of the S2 FSL (b) The output voltage of the S2 FSL simulation, Y(t) (c) Error signal suppression in dB referenced to 1 watt, $\text{dB} = 10\text{Log}(Y(t)^2)$	33
17. Merit of the noise suppression M dB The merit, M equation (15), was the ratio of the variances between the disturbance added at the output of the S2 FSL simulation and the output response of the S2 FSL.....	35
18. Transient time required for 60 dB noise suppression of the closed system due to the computational latency k.....	36
19. The transient response of the closed loop output of the S2 FSL simulation controlled by LSE plus DLQR (a) Computational latency k is 1 or 200 ns (b) Computational latency k = 8 or 1.6 us.....	37
20. Xilinx's DSP48 slice the mathematical resource.....	39
21. Block diagram of the LSE plus DLQR showing the timing in relation to sample clock.....	40
22. The block diagram of a single data path through the LSE plus DLQR, equations (12) and (13) Three clock frequencies are necessary for the calculation.....	41
23. Network analyzer display of the linear frequency response of the XtremeDSP board, that converted an analog to digital signal and then converted back to analog.44	

LIST OF FIGURES - CONTINUED

Figure	Page
24. The Block diagram of the RF modulation and demodulation.....	46
25. a. Digital mixing of an IF 4 dBm at 14.9 MHz b. Digital mixing of an IF signal 4 dBm at 10 MHz.....	47
26. The frequency response of the LP IIR filter know as the digital compensation.....	48
27. The inner optical path represents the S2 FSL controlled by the digital IIR LPF. The outer optical path represents the diagnostics. The diagnostic path measured the optical resolution free running ECDL, digitally controlled S2 FSL, and analog controlled S2 FSL.....	50
28. The optical transmissions of a frequency chirped hole of the S2 FSL with analog, digital and free running techniques.....	52
29. The mathematically recovered line width of the hole showing transmission magnitude versus frequency.....	53

LIST OF ACRONYMS

(ADC)	Analog to Digital Converter
(AOM)	Acoustic Optic Modulator
(AOSP)	Analog Optical Signal Processors
(DAC)	Digital to Analog Converter
(DLQR)	Digital Linear Quadratic Regulator
(DSP)	Digital Signal Processing
(ECDL)	External Cavity Laser Diode
(EOM)	Electro Optical Modulator
(FPGA)	Field Programmable Gate Array
(IIR)	Infinite Impulse Response
(LPF)	Low Pass Filter
(LSB)	Least Significant Bit
(LSE)	Luenberger State Estimation
(LTI)	Linear Time Invariant
(NA)	Network Analyzer
(NCO)	Numerically Controlled Oscillator
(ODE)	Ordinary Differential Equations
(PDH)	Pound Drever Hall
(PSD)	Power Spectral Density
(RF)	Radio Frequency
(RRL)	Reciprocal Root Locus
(SA)	Spectrum Analyzer
(SE)	State Equations
(S2)	Spatial Spectral
(S2 FSL)	Spatial Spectral Frequency Stabilized Laser
(Tm:YAG)	Thulium Doped Yttrium Aluminum Garnet
(VHDL)	Very high speed integrated circuit Hardware Description Language

ABSTRACT

Optical signal processing can span across bandwidths of hundreds of GHz with resolutions in the kHz range. This can be done by using spectral hole burning materials. A limiting factor in the frequency resolution is the bandwidth of the laser driving the spectral hole burning material. Better control techniques of laser jitter will decrease laser line width and increase frequency resolution of optical processing.

This thesis explores the digital control technique of inherent noise in an external cavity diode laser and the use of reconfigurable platform to prototype digital implementation.

INTRODUCTION

Spectrum Lab at Montana State University is in the process of realizing a real time spectrum analyzer for communication. Finding used or unused channels of communication in a dense communication spectrum would be of great value to both the military and civilians. The battlefield is one such example where a dense communication spectrum exists and uninterrupted communication is paramount to saving lives of soldiers and civilians.

Spectral hole burning (SHB) technology lends itself to frequencies over a wide spectral range (hundreds of GHz) and high resolution (down to 10 kHz). Hole burning is the transmission of optical energy in a crystal, within a narrow band of frequencies, that can be tuned over a wide absorption range.

A SHB crystal, when excited by a laser at a specific frequency, will subsequently only allow light at this frequency to pass through and all other frequencies will be blocked. Thus, in the context of a spectrum analysis, the entire bandwidth of the RF spectrum can be presented to the SHB crystal and then read out by seeing how much light is transmitted at a particular frequency as the laser is tuned over the entire spectrum. Tuning the laser indicates the mapping of the particular frequency with time and is generally increased linearly over the spectrum.

Being able to process a very wide spectral range at high resolution is important to detecting narrow frequency regions of interest. A limiting factor is the spectral line width of the laser reading out the information.

If a laser is tuned over the spectrum of interest without feedback control techniques a time integrated frequency output has a bandwidth of 10 MHz (Figure 1a) and thus limiting the region of interest to 10 MHz of resolution (Figure 1c). However, a controlled laser can have a line width of 10 kHz (Figure 1b) and much finer resolution in a region of interest can be obtained (Figure 1c right side).

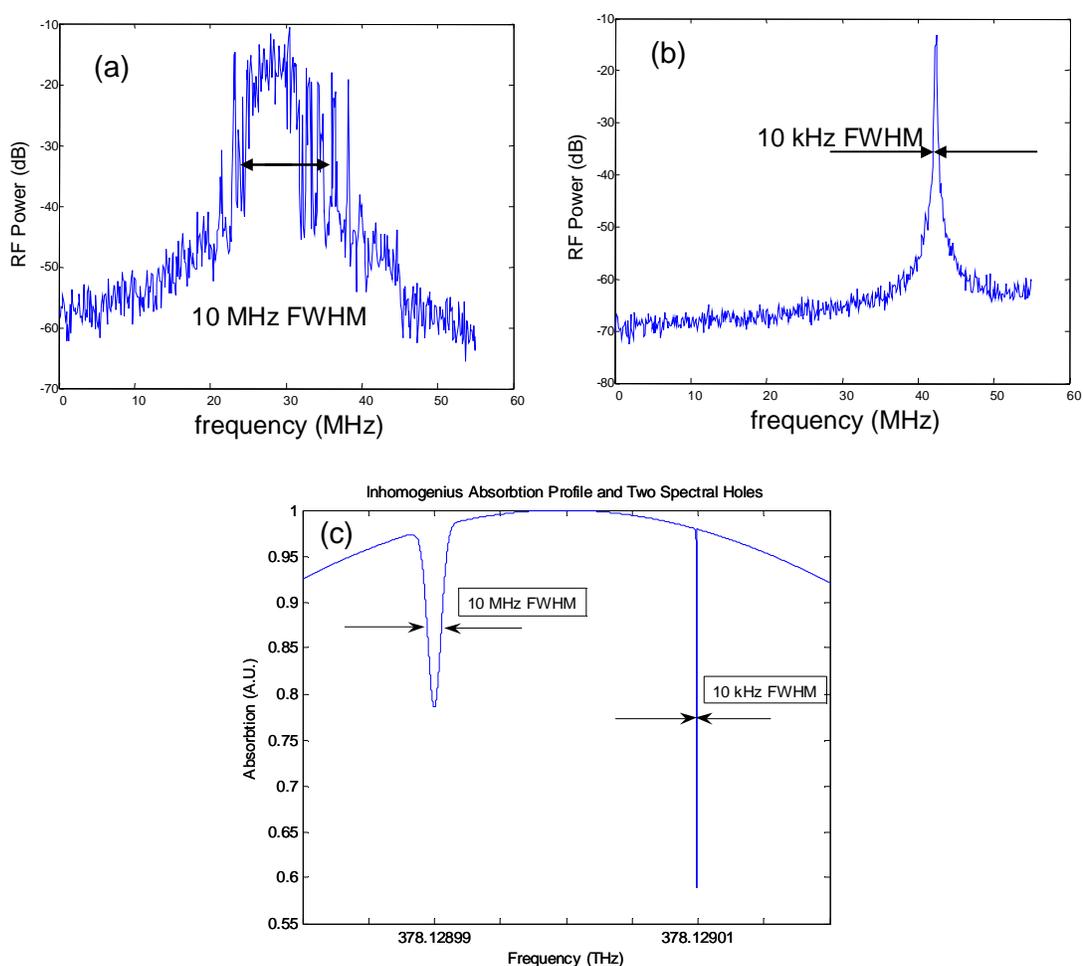


Figure 1. Three representations of ECDL line widths with spectral hole burning technique. (a) Data taken experimentally from an uncontrolled ECDL, line width 10 MHz (b) Data taken experimentally for a controlled ECDL, line width 10 kHz (c) Math representation of the controlled and uncontrolled spectral holes burned into the wide absorption of the spectral hole burning material.

Commercially available external cavity diode lasers (ECDL) suffer from inherent noise known as jitter. Jitter widens the range of frequencies from the free running ECDL to approximately 10 MHz. Fortunately, electronic feedback compensation reduces the frequencies, resulting in a reduced spectral line width as narrow as 10 kHz.

Current analog technology implementing the electronic compensation of the custom laser is a maximum of 1.5 MHz of jitter control bandwidth. Designing and replicating analog circuits is problematic due to discrete components values that can change with changing temperature.

Digital control offers the potential to reduce laser line widths below what analog techniques can do and can easily be replicated without the “tuning” needed in analog circuits. This thesis focuses on digital control of the laser and implementation in a reconfigurable platform.

SYSTEM ANALYSIS

The present system uses an external cavity diode laser (ECDL) with Pound Drever Hall (PDH) technique of frequency stabilization. The block diagram of the frequency stabilized ECDL is shown in Figure 2, and represents the components in three simplified subsystems (different shaded regions). The light shaded section represents the commercially available ECDL. The medium shaded section represents the electronics necessary for feedback compensation. The dark shaded section represents the frequency discriminator.

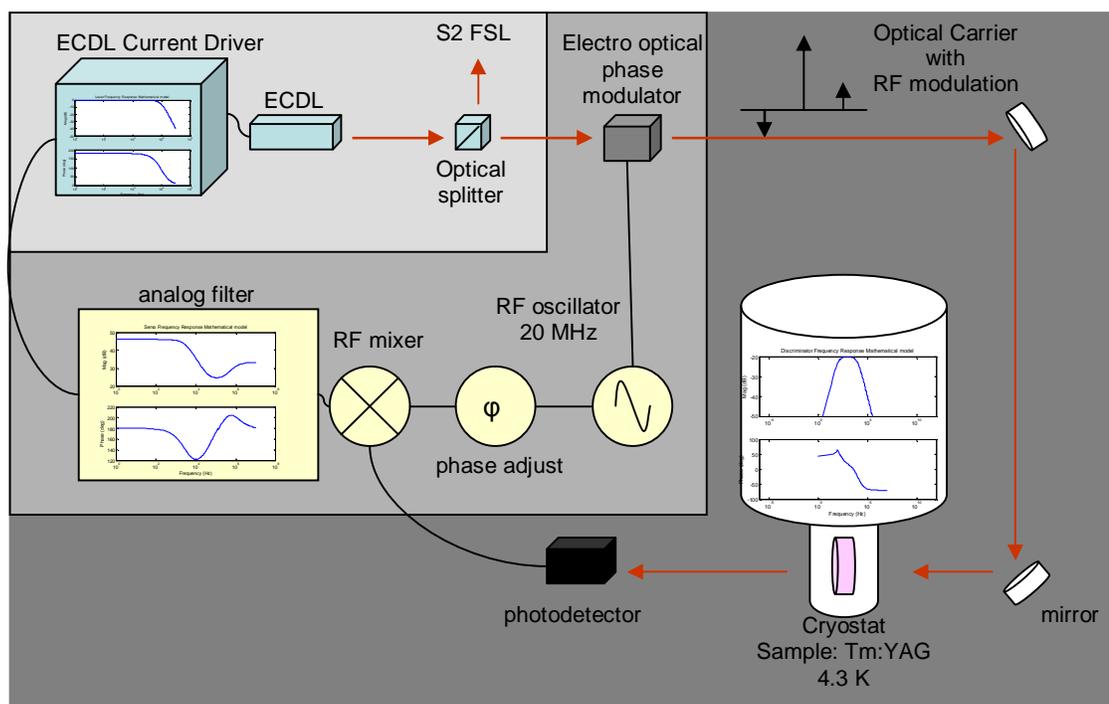


Figure 2. Block diagram of the photonics of a spatial spectral material frequency stabilized ECDL.

Model Verification of the S2 FSL

Linear time invariant approximations of the three regions in Figure 2 representing the photonics of the S2 FSL have been previously made [1]. Thomas Bottger's dissertation was the primary literature used to help aid the construction of the frequency model of the frequency stabilized ECDL prior to this work [2]. Verification of the prior frequency domain models was determined through experimental measurement. The systems verified were the ECDL, the analog feedback compensator, and the closed loop frequency Stabilized ECDL. The direct experimental analysis of the frequency discriminator was mathematically derived given the measurement of the other systems.

Frequency Tunable ECDL

The frequency tunable ECDL was commercially available from New Focus and was the vortex series model 6017. Three components are shown in the lightest section of Figure 2. Two components make up the ECDL subsystem: the electrical source and the laser diode. The third component is the optical splitter, which provided a frequency stabilized output of 378.129 THz. The current modulation of the ECDL was the control input of the frequency stabilized ECDL. The DC coupled current modulation of the ECDL allows a maximum of 600 MHz of tunable frequency. The following procedure verified the existing frequency response of the current modulation of the ECDL.

The frequency response of the ECDL was obtained by applying a linearly increasing frequency signal to the optical system containing an ECDL, and a Fabry-Perot cavity. The response was then measured by the photodetector. The Fabry-Perot cavity and the photodetector provided the frequency discrimination of the output of the ECDL, seen in the shaded region of Figure 3. The network analyzer output, a -10 dBm linearly increasing frequency signal into a power splitter, modulated the current input of the ECDL current driver. The setup is shown here in Figure 3.

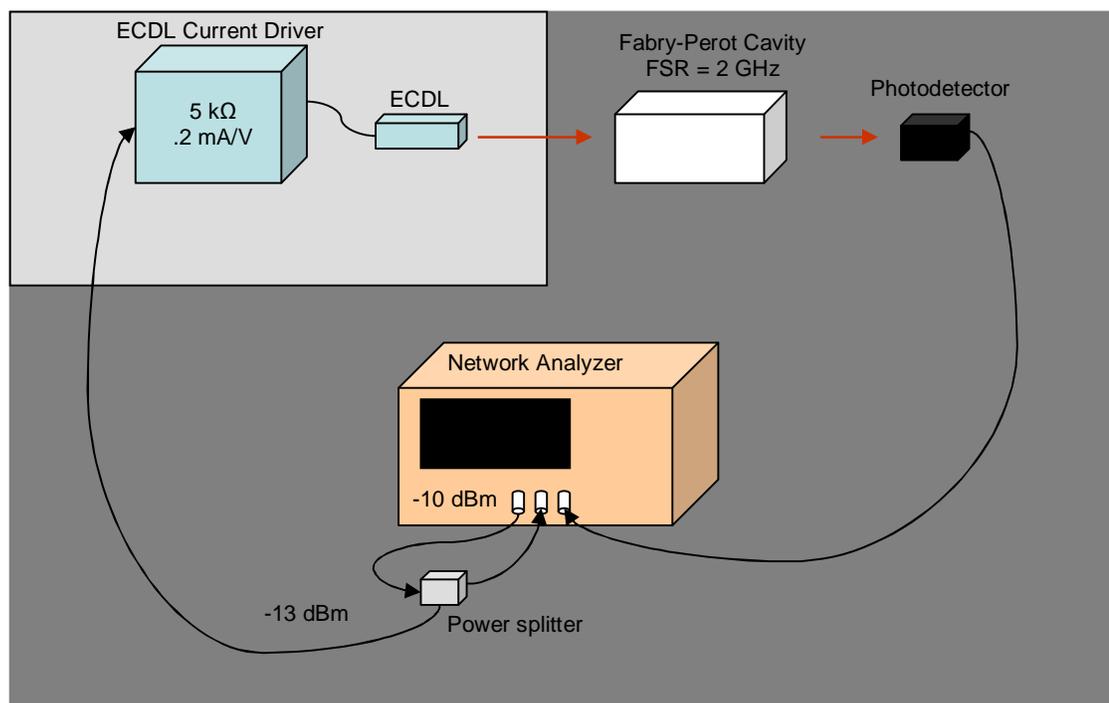


Figure 3. Experimental setup, which measured the frequency response of the current driver plus the ECDL

The resulting frequency response is shown as the Bode diagram in Figure 4. The response is the optical response of the ECDL through the Fabry-Perot cavity and measured with the photodetector. Transmission of the optical system

represented the frequency deviation due to the constant -13 dBm frequency chirp input on the current modulation of the frequency tunable ECDL. The frequency deviation due to current modulation input was seen to be the gain constant of the current modulation of the ECDL. A flat response of -5.8 dBm electrical transmission through the optical system persisted over 200 kHz.

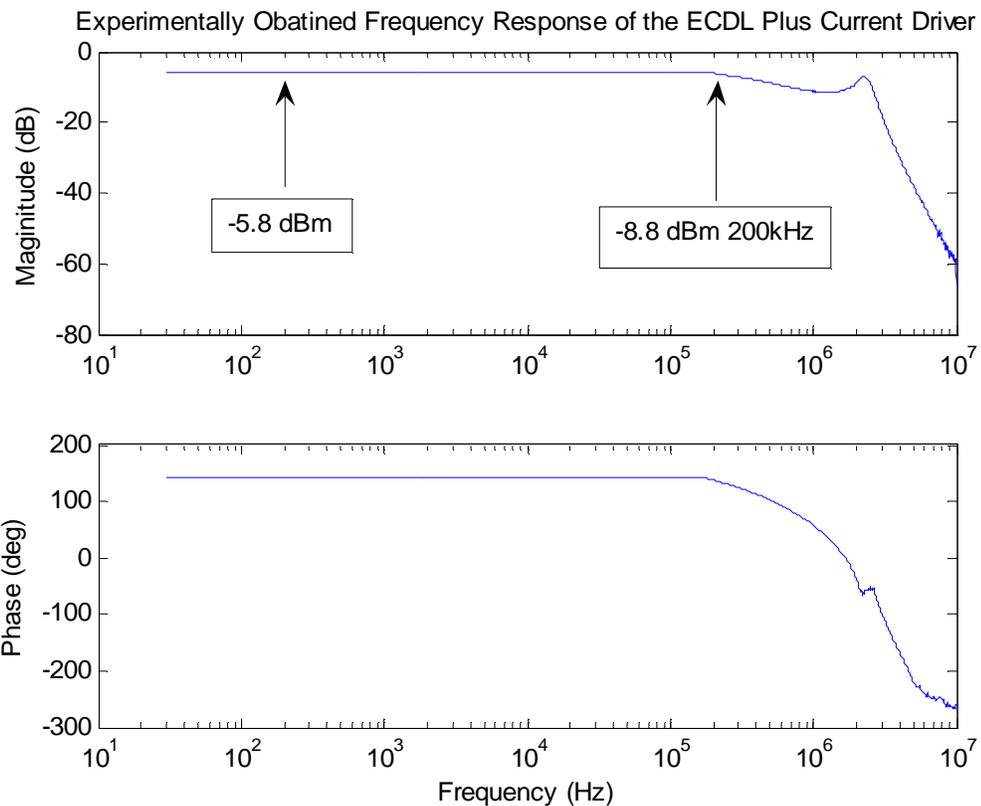


Figure 4. Experimentally measured frequency response of the current driver plus the ECDL

The modulation index of the ECDL was calculated knowing the low frequency magnitude response of the ECDL, -5.8 dBm, and the slope of the frequency discriminator. Frequency discrimination was the transmission through the Fabry-Perot cavity and commercially available photodetector. The slope of the frequency

discriminator was determined by recording the transmission of the free spectral range of the 2 GHz Fabry-Perot cavity and the New Focus model 1801 photodetector, a representation of the transmission shown in Figure 5.

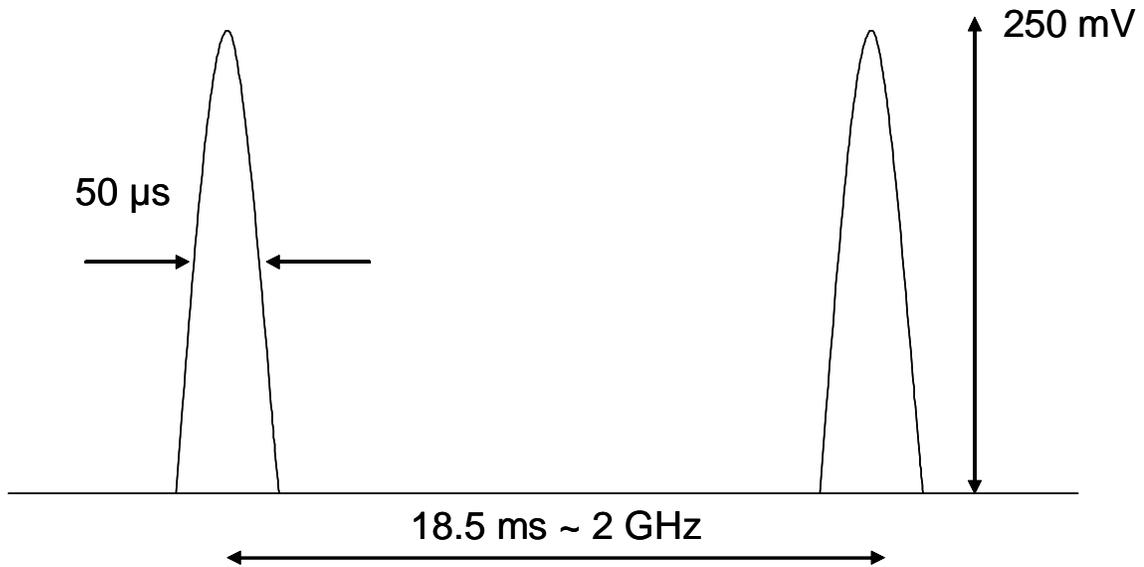


Figure 5. A representation of the optical transmission through a 2GHz Fabry-Perot Cavity coupled into a photodetector, which yielded the cavity's free spectral range.

The modulation index was calculated with the information represented graphically in the experimental setup obtaining the frequency response of the ECDL, the resulting bode diagram, and the free spectral range of the Fabry-Perot cavity. The modulation index of the ECDL was 54 GHz/A and was a close match to the existing modulation index, 50 GHz/A in the work completed by Khallaayoun et al. [1].

The experimentally determined frequency response of the ECDL was modeled simply as a two pole low pass filter. Two problems were noted when modeling the ECDL with a second order low pass filter: the magnitude response and the total phase.

The order of the experimentally obtained ECDL indicated a higher order than the two pole low pass filter. The indications were the magnitude peaking out past the 200 kHz cutoff frequency and the total phase loss; however, reducing the order of the ECDL simplifies calculations.

The ECDL and its current driver were verified from the above procedure and the transfer function from earlier work shown here in equation (1).

$$ECDL(jf) = \frac{K_c F_c^2}{(jf + F_c)^2} \quad (1)$$

Here, K_c is the modulation index, $5e10$ Hz/A and F_c is the cutoff frequency located at 200 kHz. The frequency response of equation (1), a second order system, was determined by the poles of the system.

Analog Feedback Control

The frequency response of the current analog compensator was obtained experimentally with the network analyzer. An additional RF mixer was added to the experimental setup in order to include the RF oscillator, phase adjuster, RF mixer, and the analog compensation filter describing the complete analog electronics for feedback control, shown as the block diagram in Figure 6. The addition of the RF mixer modulates the frequency chirp, which represents the jitter signal of the S2 FSL. This is modulated onto the 20 MHz RF oscillator. The frequency bandwidth of the chirp represents the frequency bandwidth of the jitter. The frequency chirp was then down

converted to baseband for the input of the analog compensation filter. Analysis generated the representation of the control input produced by the analog circuitry due to signals corresponding to jitter from the frequency discriminator.

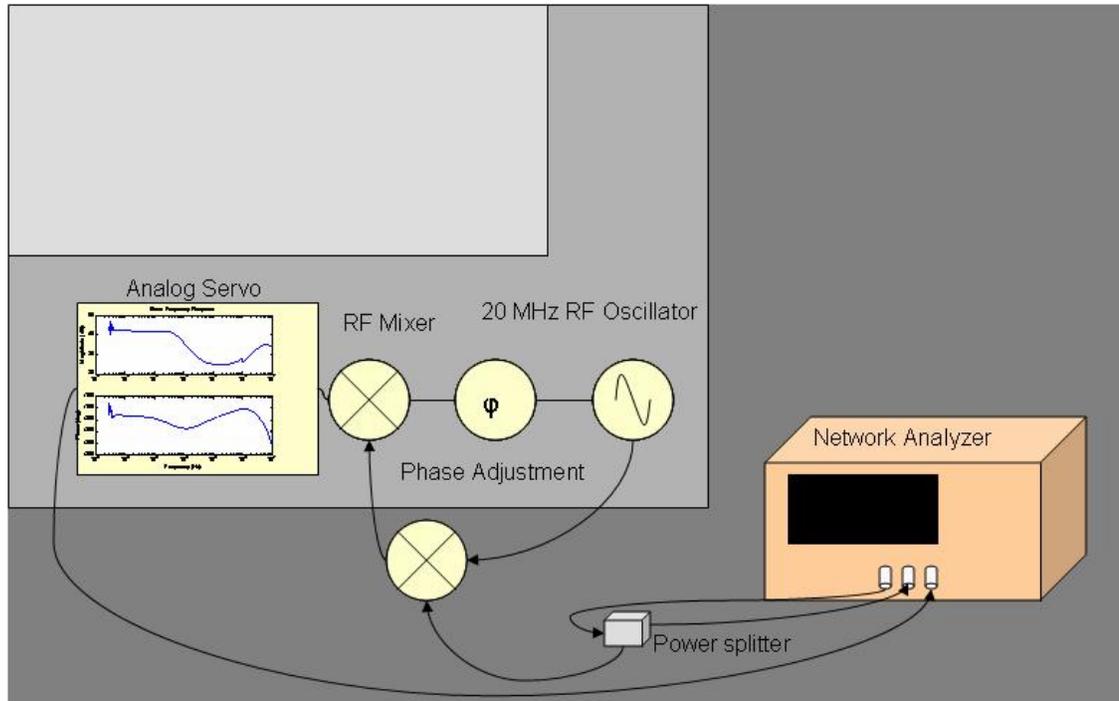


Figure 6. Block diagram of the experimental setup measuring the frequency response of the complete electronic feedback control.

The frequency response of the complete analog control was obtained with the above description and is shown in Figure 7. The analog control can compensate low frequency jitter signal with a maximum of 42 dB gain. The feedback gain falls off at 7 kHz but phase lead compensation extends the controllable frequencies of the jitter of the closed loop S2 FSL.

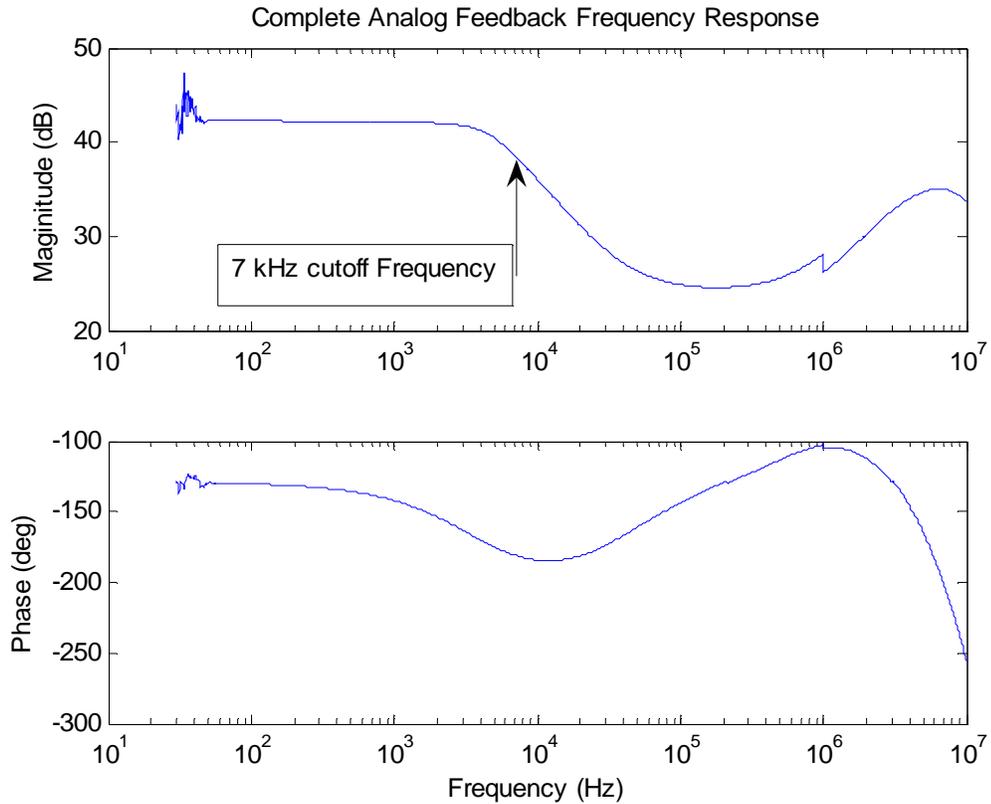


Figure 7. The frequency response of the complete analog feedback presently used to frequency stabilize an ECDL at Spectrum Lab.

Closed Loop S2 FSL

The closed loop frequency response of the S2 FSL was analyzed with the same process described above for the frequency tunable ECDL; however, the frequency discrimination of the Fabry-Perot cavity was exchanged for the frequency discrimination of the S2 sensor material. A block diagram representing the experimental setup is shown in Figure 8. Once the frequency response of the closed loop S2 was measured, the only unknown response was the S2 sensor material.

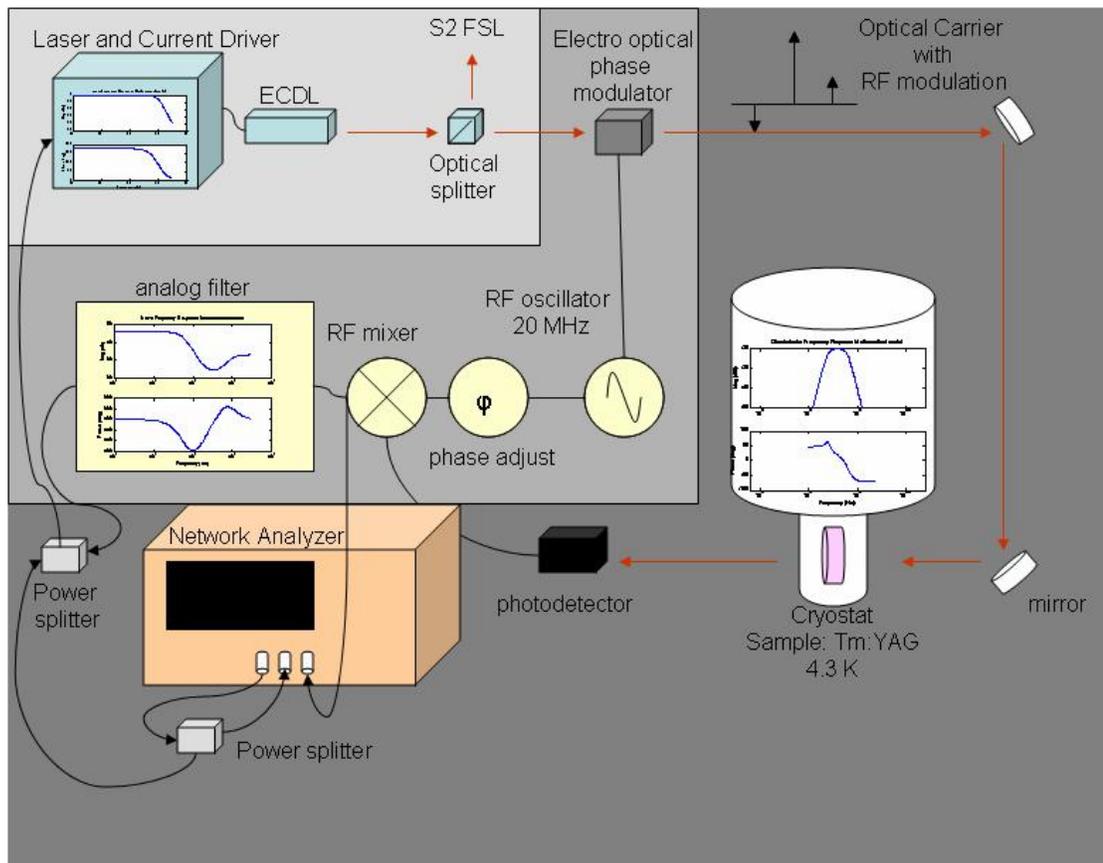


Figure 8. This experimental setup measured the closed loop frequency response of the S2 FSL

The light shaded region of Figure 8 represented the start of the frequency stabilization process containing the ECDL and the electrical current driver. The output of the ECDL was optical energy spread over 10 MHz of bandwidth centered at the optical carrier frequency, approximately 378 THz. The dark shaded region represented the discrimination of the instantaneous frequency deviation of the ECDL and was the error signal. The last process in the frequency stabilization of the ECDL, the middle shade of the three regions, was the feedback of the electronically compensated error signal into the electric-current modulation of the ECDL.

The network analyzer was placed in parallel with the S2 FSL, which measured the output response for a given input response over 10 MHz. The network analyzer amplitude modulated through the electrical power splitter onto the compensated error signal feedback, which corresponded to the input of the control loop. The output response of the S2 FSL was taken at the output of the RF mixer. The device under test was the complete S2 FSL and the analysis was the frequency response of the modulation input of the ECDL in comparison to the error taken from RF mixer. The frequency response is shown in Figure 9.

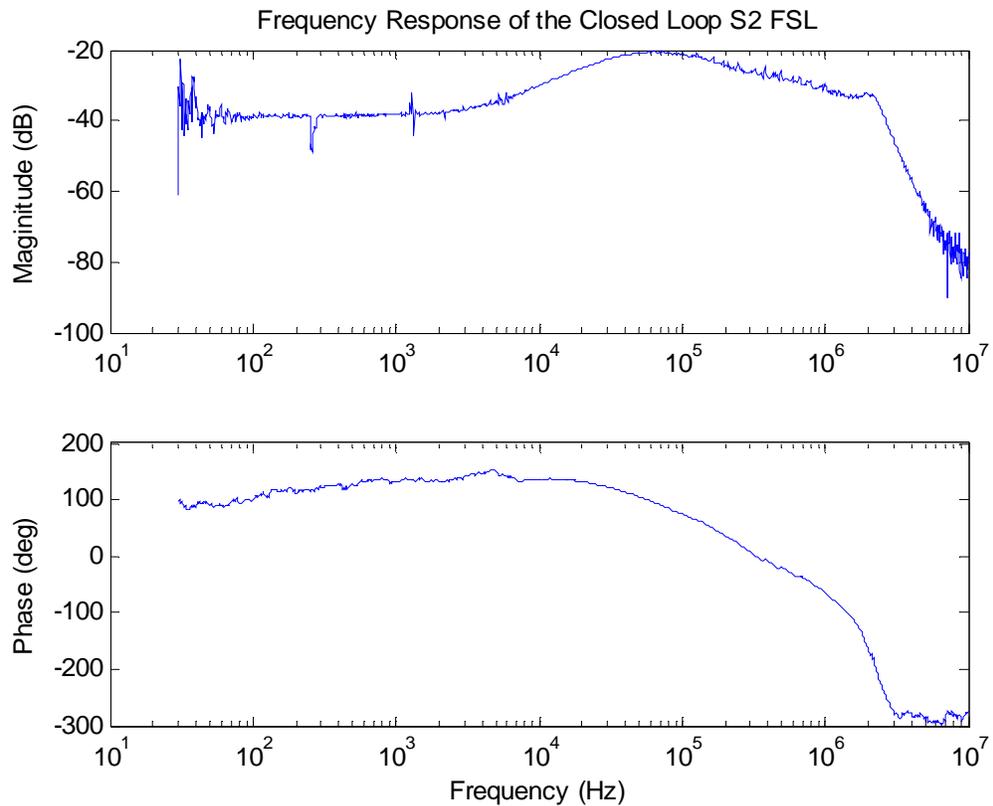


Figure 9. The frequency response of the closed loop S2 FSL

S2 Frequency Discriminator

Unlike the previous three frequency responses of the S2 FSL, an attempt was made to mathematically calculate the frequency response of the S2 frequency discriminator. Three of the four frequency responses were verified by experimental measurement. Four variables make up the closed loop S2 FSL. The block diagram is shown in Figure 10 and the corresponding frequency domain mathematical model in equation (2). The control input to the frequency tunable ECDL is $U(s)$. The resulting frequency output due to $U(s)$ is $Y(s)$. Here, the ratio of the output over the input represents the S2 FSL.

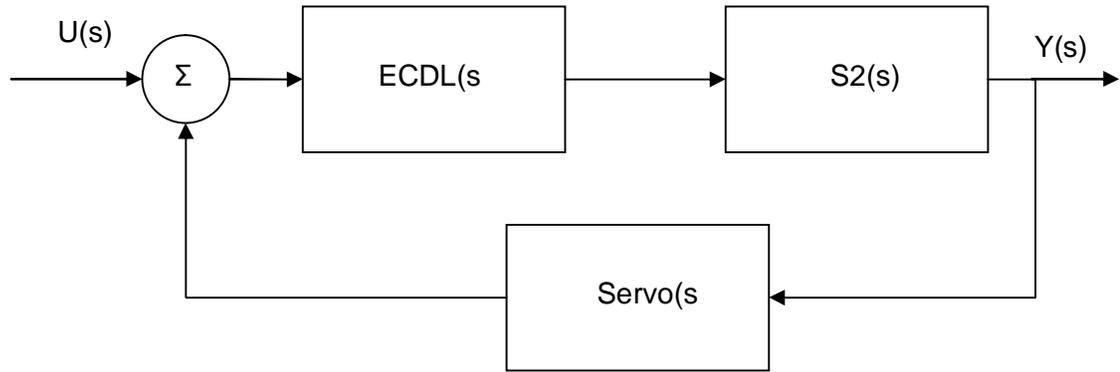


Figure 10. Block diagram representing the closed loop S2 FSL with three subsystems

$$\frac{Y(s)}{U(s)} = \frac{ECDL(s)S2(s)}{1 + ECDL(s)S2(s)Servo(s)} \quad (2)$$

The frequency discriminator is the S2 sensor material and is a difficult photonic setup to measure. Solving the frequency response of the S2 material was not difficult

having measured the ECDL, servo, and the closed loop S2 FSL. The previous frequency domain mathematical models validated this procedure. Solving for the S2 frequency discriminator, given the ECDL, servo, and the closed loop S2 FSL, resulted in an exact match of the modeled S2 frequency discriminator.

$$S2(s) = \frac{\frac{Y(s)}{U(s)}}{ECDL(s) - \frac{Y(s)}{U(s)} ECDL(s) Servo(s)} \quad (3)$$

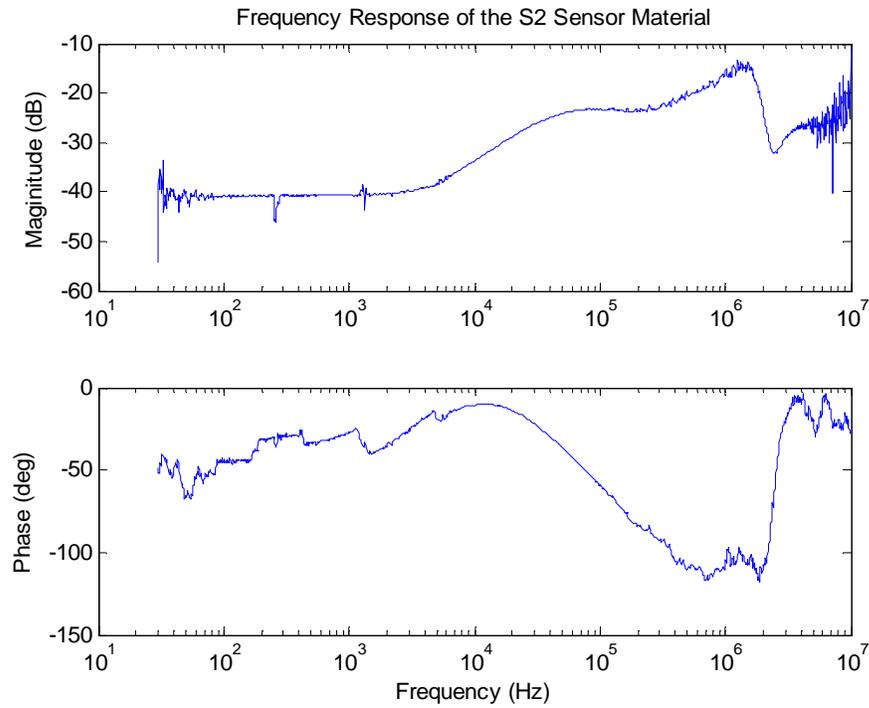


Figure 11. Calculated frequency response of the S2 sensor material

The frequency response calculated from equation (3) with the experimentally obtained frequency responses did not result in the bandpass filter response of the

experimentally measured S2 material, described in Bottger's dissertation. Three features that described the bandpass feature were a low corner frequency, a high corner frequency, and the mid band gain of the S2 frequency discriminator

$$D(jf) = \frac{DF_h jf}{(jf + F_h)(jf + F_l)} \quad (4)$$

The low frequency pole, F_l , was located at 100 Hz. The high frequency, F_h , pole was located at 73.5 kHz. The mid-band gain, D , 280e-6 A/Hz described the gain of the S2 frequency discriminator through a photodetector. These values were calculated from earlier work done at Spectrum Lab [1].

Open Loop Gain of the ECDL

The frequency model of the ECDL and the S2 frequency discriminator in series made up the frequency domain model. The open loop gain of the system represented the electrical signal due to the irradiation of the photodetector with the optical transmission of the ECDL through the S2 material shown in equation (5). This frequency response was the combination of the two poles of the ECDL and the two poles of the band pass response of the S2 material. The poles of the transfer function were the roots of the denominator and corresponded to the natural modes of the system. The four poles made this system a fourth order model.

$$ECDL_S2_open_loop(jf) = \frac{K_c D (F_c)^2 F_h jf}{(jf + F_c)^2 (jf + F_h)(jf + F_l)} \quad (5)$$

Control of the fourth order model required the design of a compensator placing the natural modes of the closed loop S2 FSL in the desired location for a desired response.

A single input single output model described the system in equation (5). Frequency designed control compensates the single output by changing the gain and phase corresponding to its frequency. In comparison, the time domain control identifies the single output as the individual effect of each natural mode and compensates them while constructing a single input.

Optimum Control Design

Several papers have described noise suppression using the PDH technique for frequency stabilizing an ECDL. Most notably were Mor and Arie [3]. They described the best performance of a frequency stabilized ECDL as a shot noise limited performance.

The sum of the noise sources around the PDH FSL were described with a closed loop equation [3,4]. The noise sources are ECDL, $S_{f,ECDL}$, frequency discriminator, $S_{f,disc}$, and the electronic implementation of the compensation filter, $S_{f,elec}$. The individual noise sources were summed into the closed loop system to obtain the total noise power spectral density of the system, shown in equation (6).

$$S_{f,out} = \frac{\sqrt{\left(S_{f,ECDL}\right)^2 + \left(S_{f,disc}K(f)G(f)\right)^2 + \left(S_{f,elec}K(f)\right)^2}}{\left|1 + G(f)K(f)D(f)\right|} \quad (6)$$

In the above equation, $K(f)$ is the magnitude frequency response of the ECDL and $G(f)$ is the magnitude frequency response of the feedback compensator. $D(f)$ is the frequency response of the frequency discriminator, which has been described above as

the S2 material and the photodetector. If $G(f)$ has no limit, then the noise suppression of the locked laser equation (6) reduces to equation (7). Equation (7) limits the total noise in the frequency stabilized ECDL with the PDH method to the parameters of the frequency discriminator. Minimum noise was calculated as the ratio of the noise inherent in the frequency discriminator, $S_{f,disc}$, and its ability to translate the change in frequency deviation of the ECDL, $D(f)$, into electrical signal.

$$S_{f,out_min} = \frac{S_{f,discc}}{D(f)} \quad (7)$$

The greatest contribution to $S_{f,out}$ is low frequency noise having a profile matching that of random walk and flicker noise. This is a common noise profile of most electronic components. The reduction of the low frequency noise is accomplished with closed loop control through a compensation filter.

Analog Feedback Control Presently Implemented

A simple compensator can be a low pass filter and is easily implemented with operational amplifiers. Operational amplifiers have tremendous gain at DC, however quickly fall off at higher frequency. Operational amplifiers designed with classic frequency control techniques extend the operational bandwidth, as well as the shape of the magnitude and phase of the frequency response. Gain bandwidth product, a metric for comparing different operational amplifiers, indicates the gain allowable for a compensator over a particular bandwidth.

The frequency response of the present analog compensator is shown in Figure 7. The compensator is more complex than the simple low pass filter. It had 42 dB of gain for noise compensation until the cutoff frequency at 7 kHz. Phase lead compensation extends the useful bandwidth of the compensator in the closed loop S2 FSL. Stability of compensator as a closed loop control for the S2 FSL can be determined by the forward gain. Forward gain is the combined frequency responses of the laser, the frequency discriminator, and the compensator. The bode plot of the forward gain of the S2 FSL can not have a magnitude greater than 0 dB for a phase loss of 180 degrees

relative to low frequency. Presently the compensator has pushed the noise suppression bandwidth of the S2 FSL to approximately 1.5 MHz [5], by utilizing the best operational amplifiers available.

Digital Control Design

Compensation designed with time domain control theory can be simple and elegant or complex and intractable. Time domain compensators are not limited to the control of linear time invariant (LTI) systems. A compensator designed in time can be optimized to a set of criteria beneficial to the system in need control. Time domain control theory accelerated alongside the growth of digital hardware calculation, the result of the PC boom [6]. Time domain designed compensation with digital hardware implementation expands the available compensation to the frequency stabilization of an ECDL.

The digital hardware chosen to implement the time domain designed compensation was the field programmable gate array (FPGA). In recent years, many industries have had to spend millions of dollar to implement very high-speed systems into a single silicon chip. As a result, generic microprocessors of varied architectures have evolved to higher speeds of operation with more available design resources. Two efficient processors commercially available are the dedicated digital signal processor and the FPGA. The dedicated digital signal processor implements a single hardware architecture, operating at relatively high clock speeds, which can be greater than

1 GHz. Digital signal processing (DSP) necessary for implementing digital algorithm, like time the time domain compensator, is done sequentially through the fixed hardware.

FPGAs, in contrast, have numerous hardware resources available for configuration. This leads to DSP which operates simultaneously, known as parallel processing. Every calculation done simultaneously reduces the necessary clock speed of the FPGA in comparison to the dedicated digital signal processor. The time domain algorithm designed for control of the S2 FSL benefits from parallel processing.

FPGAs are packaged into convenient digital hardware for rapid prototyping. A convenient feature is the hardware necessary to convert an analog signal to digital words and digital words back into an analog signal. Analog to digital conversions (ADC) can occur at high sampling frequencies of 2 GHz. Pentek is a company that incorporates an ADC by Atmel, converting analog signals to a 10 bit digital words, available for processing in the Virtex II Pro FPGA, in under 10 ns. The short time delay of the conversion makes high bandwidth compensators possible. Other features include several on/off board communication protocols, human interfaces, master clock oscillators, and daughter board expansion. These convenient resources available on the FPGA prototyping board have made successful the digital implementation of the necessary systems required of the active electronic feedback control for S2 FSL.

DIGITAL CONTROL DESIGNS

Compensation, designed through time domain control theory, controls the open loop gain of the ECDL through the S2 material for a desired response in mathematical simulation. A simple low pass filter implemented in an FPGA successfully frequency stabilized the S2 FSL.

Design Approach

Designing a digital control for the open loop gain of the free running ECDL and the S2 frequency discriminator is similar to the procedure for designing an analog feedback control system. Procedures are numbered as follows:

1. Establish the desired approach, either the classical frequency method or the time domain, known also as state space method. In this work both time and frequency methods were used.
2. Develop a mathematical model of the open loop system. Models are described in either frequency or the time domain, depending on the choice of step one; however, methods are available to transform one to the other.
3. Design a controller to meet the design specifications using the method selected in step 1.
4. Evaluate the following for the design in simulation arrived at in step 3 for the specifications.
 - a. Bandwidth of the closed loop system

- b. Transient response of the output due to input disturbance
 - c. Noise considerations of the digital system
5. Repeat steps three and four until a satisfactory design is achieved.
 6. Construct the prototype, using one of the numerous available programmable embedded microcomputers.
 7. Test by “Closing the loop” around the photonics of the S2 FSL using the embedded microcomputer programmed with the control algorithm.
 8. Repeat steps one through eight until a system is designed that satisfies the design specifications.

Time Domain Control Design

Transformation of the linear time invariant (LTI) frequency domain model of the S2 open loop ECDL system, equation (5), was performed to utilize the time domain control theory. The transformation was a set of first order ordinary differential equations (ODE) in time. The order of the transfer function determines the number of ODE necessary to represent the state equations (SE), which, when transformed, represented equation (5). The practice of transforming frequency domain models into the time domain is so common that the technique was automated by mathematical modeling software. Matlab software in conjunction with the controls toolbox utilized this automation to transform the frequency domain representation of the ECDL S2 open loop into the time domain SE, shown as equations (8). The text commands used to transform the transfer function into the SE within Matlab is shown in Appendix A.

Many controls engineering text books detail this transformation [6] and will not be described here.

$$\begin{aligned}\frac{d\bar{x}(t)}{dt} &= A\bar{x}(t) + B\bar{u}(t) \\ \bar{y}(t) &= C\bar{x}(t)\end{aligned}\tag{8}$$

Digital Linear Quadratic Regulator

Time domain designed compensators can theoretically regulate controllable systems in arbitrarily fast times. The natural modes of equation (5) corresponded to the number of ODEs in equations (8). Controllability of the model was established as linear independence between all four ODEs. The rank of the controllability matrix $[B \ AB \ A^2B \ A^3B]$ quickly determined that the number of linearly independent equations was three. The indication was that one of the four states of vector $\bar{x}(t)$ in equations (8) were not controllable through the design feedback of vector, $\bar{u}(t)$.

Though a time domain designed compensator could not regulate the system in arbitrarily fast times, stability to the desired response was achievable. The inverse Laplace transform of the partial fraction expansion of equation (5) produced a four-term function of decaying exponentials in time that represented the natural modes of the system. The state unaffected by a specific input control decayed in a reasonable time for the design of a compensator achieving a desirable closed loop response. Here

the desired response was determined to have 60 dB of suppression on the voltage disturbances added to the input, with 1 MHz of control bandwidth of the S2 FSL.

The available design parameter for closed loop control was the control vector $u(t)$. The design of the control vector was an automated process by Matlab with the control system toolbox. The steps to completing the time domain controller, the control vector $u(t)$, were described in the subsequent list. The description of the design is a description of the machinery utilized to design a digital linear quadratic regulator (DLQR).

1. The time domain model equation (8) was transformed from the fourth order LTI frequency model of the ECDL plus S2 open loop gain.
2. The SE was established as an uncontrollable model; however, it was stable with closed loop compensation for a desired response. The rank of the controllability matrix was three. The inverse Laplace transform of partial fraction expansion of equation (5) was performed to see all natural modes of the system decayed in time.
3. The time domain design of the compensator applied through $u(t)$ in equation (8) did not abandon useful frequency domain design tools. The root locus was the graphical technique that matched a particular transient response of the ECDL S2 open loop system to a gain applied closed loop. The reciprocal root locus (RRL) was formed with the open loop gain of the S2 FSL, equation (5), multiplied by its reciprocal, shown here in equation (9), [7].

$$ECDL_S2_open_loop(jf)xECDL_S2_open_loop(-jf) \quad (9)$$

4. The desired transient response of the S2 FSL was established as 60 dB of voltage suppression at the output of the closed loop S2 FSL to a bandwidth of 1 MHz. The effort required due to the established transient response was the gain vector G . The control law, equation (10), was the vector multiplication of the gain vector and the state vector.

$$u[k] = -Gx[k] \quad (10)$$

Here, k represented the sample domain, which was an integer multiple of the time duration corresponding to the inverse of the sampling frequency of the digital hardware. The sampling frequency for simulation was 5 MHz.

5. The values of the gain vector were calculated by minimizing a cost function that was constrained by equation (8) and the criteria of effort, Q , placed on the S2 FSL.

$$J = \sum x'Qx + u'Ru \quad (11)$$

6. The effort of the control law was determined in step 3 from the desired transient response of the noise suppression in the closed loop S2 FSL. The effort matrix Q was formed with the output vector C , equations (8) and the

feedback gain selected from the RRL in step 3, $C'qC$. This procedure was accomplished with a command file written for Matlab, Appendix A.

7. The calculation of the feedback gains, G known as the DLQR, was iterative. Steps 3 through 6 were repeated until satisfactory results could be found for digital programming. Satisfactory results were 64 bit fixed point calculations sampled at 5 MHz.

Luenberger Method

The control law represented in equation (10) can be simple to implement with digital hardware. Control law was the vector multiplication of the constant DLQR vector, G , with the analog to digital conversions of the states of the system, $x(k)$. This implies that each state of the system was observable as output; in fact the system was a single input single output system, equation (5).

The system was checked for observability. The test for observability was checked through the rank of the matrix formed by the state transition matrix A and the output vector C from equation (8). Full rank of the observability matrix was determined.

A linear algebra system designed the estimated states of the ECDL S2 open loop time domain model. The estimated states were fed back through the DLQR for frequency stability of the S2 FSL in simulation. A full Luenberger state estimator (LSE) was calculated for the observation of the single state output of the ECDL S2 open loop gain. The control law, equation (10), substitutes each state $x[k]$ for an estimation, $\hat{x}[k]$. Here the revised control law was shown in equation (12).

$$u[k] = -G \hat{x}[k] \quad (12)$$

The theory describing the LSE can be seen in many modern control text books [6]. Equation (13), used from theory to obtain a vector of state estimators, $\hat{x}[k+1]$, in conjunction with equation (12), completed the time domain designed feedback compensation.

$$\hat{x}[k+1] = \hat{A} \hat{x}[k] + \hat{B} u[k] + H y[k] \quad (13)$$

The carrots over the system parameters and the vectors in equation (13) indicates the estimation of the modeled system and not the actual modeled system as in equation (8). The variables \hat{A} , \hat{B} , and $\hat{x}[k]$ were estimation values of equations (8) and directly correspond to like variables. Here, H was the observation vector, and it split the single output value into four observation variables. The vector \hat{B} was a representation of the input of the system and therefore multiplied the control law. The matrix \hat{A} , and the vector $\hat{x}[k]$ represented the estimated evolution of the states of the modeled system. These three terms in equation (13) calculated the predictions of the states and therefore the estimated states were available for feedback through the DLQR. A block diagram shown in Figure 12 represented the simulation of the closed loop compensated S2 FSL.

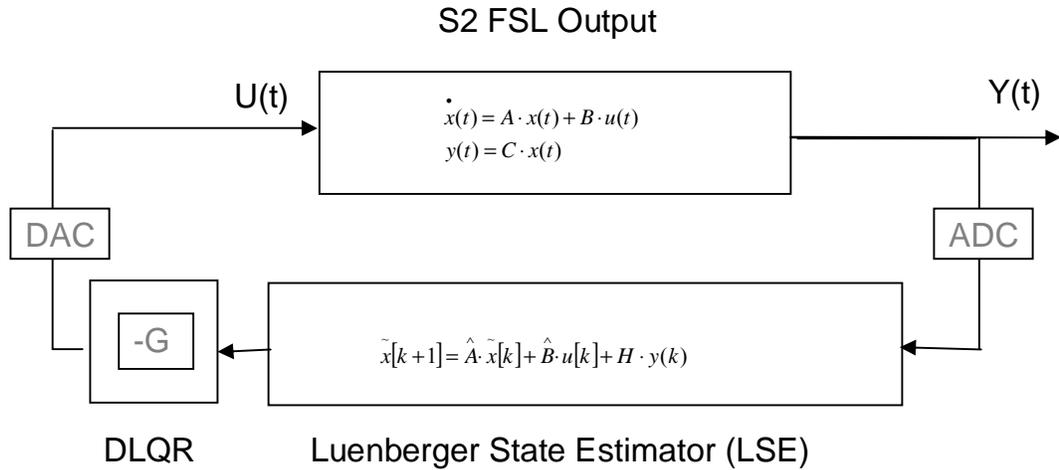


Figure 12. The closed loop model of the S2 FSL compensated with the digital controller designed with time domain control theory.

Infinite Impulse Response Low Pass Filter

The filter design and analysis tool available in Matlab was used to quickly obtain a low pass (LP) infinite impulse response (IIR) filter. Double precision floating point implemented the design. Tools quickly transformed the calculation precision of the LP IIR filter into the fixed point precision required for very high speed integrated circuit hardware description language (VHDL). The input and output to the LP IIR filter was necessarily limited to 14 bits, due to the on board digital conversion and 36 bit internal fixed point calculations, were implemented within the FPGA.

Evaluation of the frequency response of the 0 dB low frequency gain and 20 dB attenuation at 1MHz is seen in Figure 13.

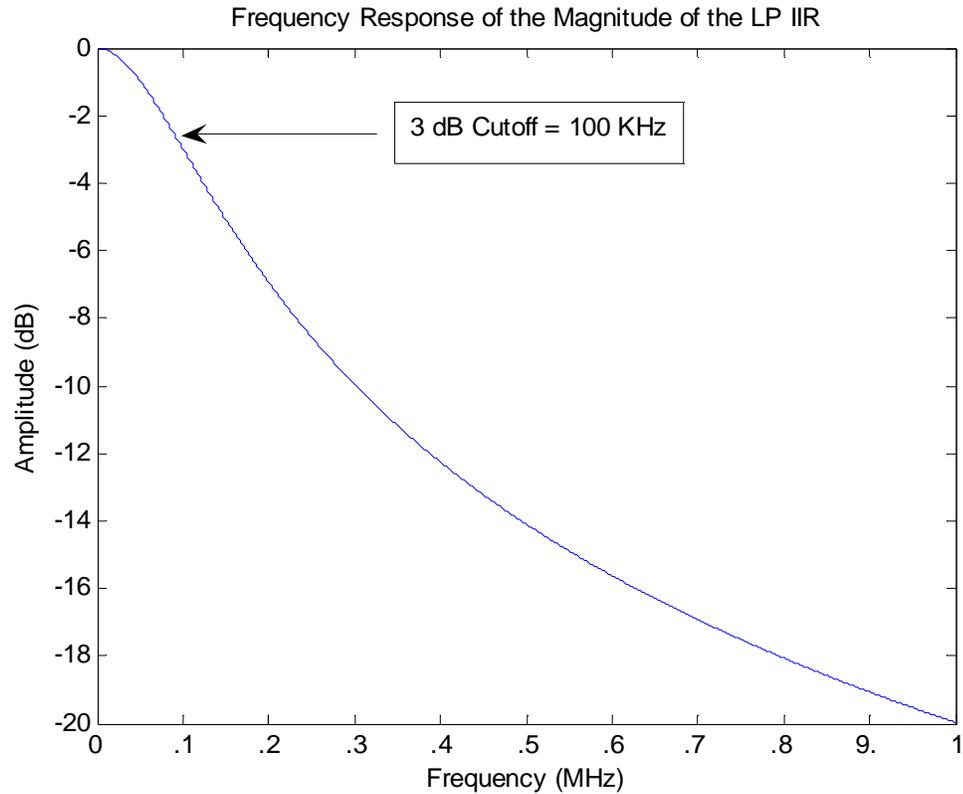


Figure 13. The magnitude frequency response of the LPF IIR filter calculated with the freqz command in Matlab with the coefficients designed by the FDATool.

The direct form II block diagram of the LP IIR filter is shown in Figure 14.

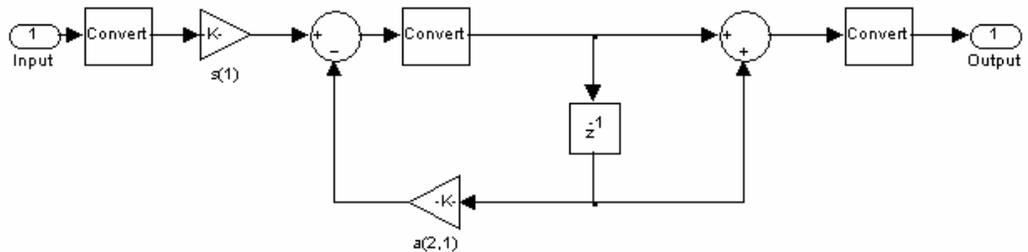


Figure 14. The LP IIR filter block diagram implemented as a direct form II model

SIMULATION OF THE DLQR

The simulations were developed to test the validity of the designs of both digital controllers. The closed loop system was tested as the time based model of equation (8) which simulated the ECDL S2 open loop compensated by the DLQR equation (13).

Time Based Simulations of the LSE Plus DLQR

The implementation of the LSE plus DLQR compensator in the closed loop system with digital signal processing (DSP) was assessed for feasibility. The design with time domain control theory and the evaluation of the model was finished with the aid of Simulink, Matlab's graphical simulation tool. The criteria that were used to evaluate the LSE plus DLQR controller included transient response, stabilization due to time delay, and bandwidth of the closed loop system.

The SE equations (8) of the S2 open loop ECDL were graphically represented in Simulink and are seen in Figure 15. The recursive calculations for the time evolution of the S2 FSL were represented with the high precision of double floating point arithmetic. Here, the distinction was made between the open loop gain of the ECDL through the S2 material and the S2 FSL. Closed loop compensation of the open loop gain of the ECDL through the S2 material was the S2 FSL. The LSE plus DLQR were calculated using equation (10), and (13). The calculations required of the equations were performed with 64 bit fixed point arithmetic based on a 5MHz sample clock. The relatively slow sampling frequency, inversely proportional to the sample variable k ,

was selected before hardware was acquired. The relatively longer duration of time, k , required wider fixed point word, 64 bits. The fixed point word length determined the range of numbers that were represented within the range of calculations.

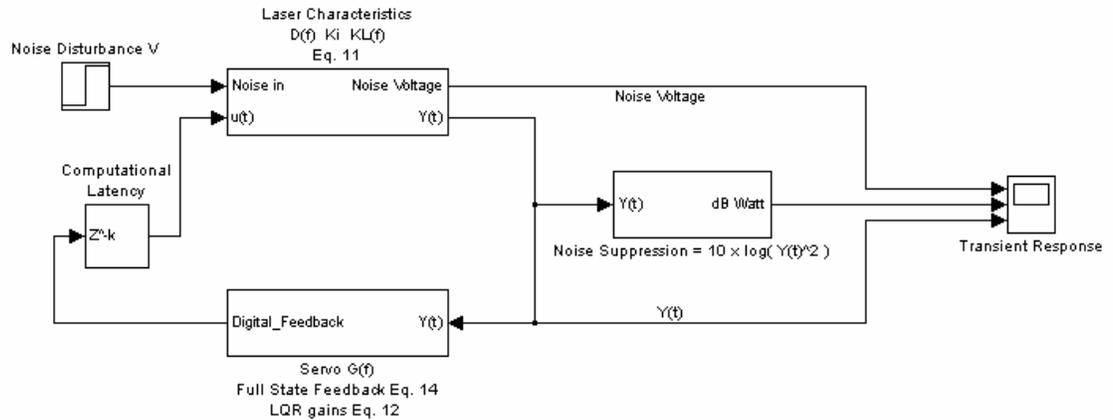


Figure 15. The Closed loop System as modeled in Simulink. The block diagram represents the S2 FSL controlled by the LSE plus DLQR. The error signal was simulated with a voltage signal added into the output of the S2 FSL. The suppression of the error signal was calculated as the power in db in reference to 1 watt.

Transient Response

First investigation of the digital controller was the transient response which consisted of the closed loop system disturbed at the output. The output was monitored for voltage suppression of a step voltage added directly at the output of the S2 FSL. The output of the S2 FSL model was the error signal measured in voltage, which represented the instantaneous frequency deviation of the ECDL output away from the optical carrier. The model determined the suppression of the added voltage at the output of the S2 FSL, due to the LSE plus the DLQR compensation controlled closed loop. The suppression was calculated as the voltage power into a one ohm load in dB

watts. The results were graphed and the continuous 60 dB of voltage suppression after 9 us is seen after the arrow in Figure 16c. The transient analysis indicated that error signals lasting longer than 9 us in duration would be suppressed by 60 dB in power. The step function, the output voltage, and the error signal suppression, were plotted and shown in Figure 16.

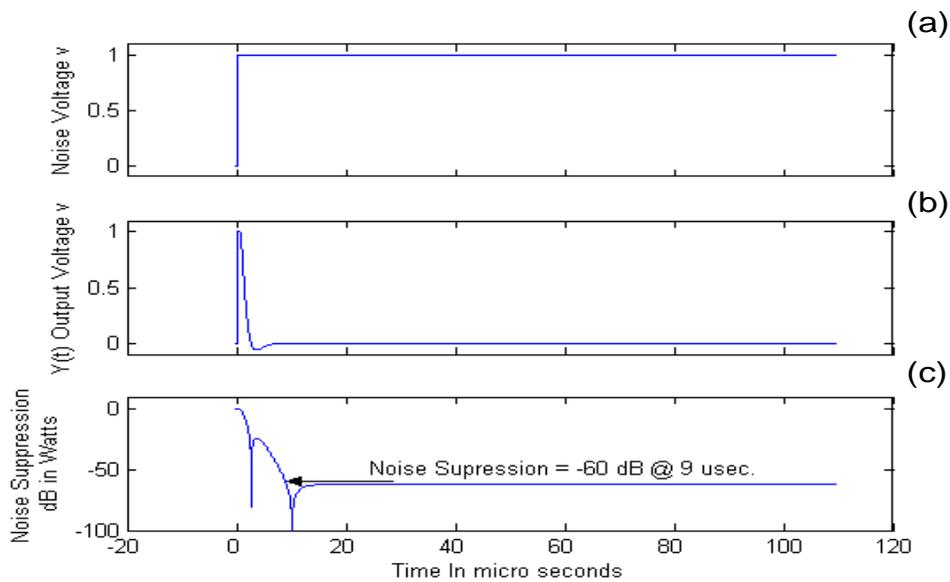


Figure 16. The closed loop transient response of the S2 FSL simulation. (a) A voltage step function added to the output of the S2 FSL (b) The output voltage of the S2 FSL simulation, $Y(t)$ (c) Error signal suppression in dB referenced to 1 watt, $\text{dB} = 10\text{Log}(Y(t)^2)$.

Merit of the Transient Response

The merit of the transient response of the S2 FSL was established by comparing the reduction of the variance at the output of the S2 FSL to the variance of the added disturbance [8]. Literature has described the full width half max line-width reduction [8][9]. Different from the transient response, the disturbance added at the output of the S2 FSL was a band limited white noise source. The merit was established as the ratio of the variances of the disturbance and the corresponding output response of the S2

FSL. The disturbance and the output response were voltages. The disturbance represented the noise at the output of the S2 FSL. The corresponding output response represented the error signal of the S2 FSL. The variances were defined as the sum of the squares of the difference between the mean and the actual amplitudes of both the disturbance and the output response [9].

$$\sigma^2(x) = E[(X - \mu)^2] \quad (14)$$

The expected values of both the disturbance and the output response were assumed zero. This assumption reduced to the sum of the squares of the actual values of the disturbance and the output response.

The effective suppression, M , of the system was seen as the ratio of the variances of the disturbance and the output response. The effective merit was calculated as the power in dB in reference to one watt of the ratio of variances, equation (15).

$$M_{dB}(white_noise_bandwidth) = 10 \cdot \text{Log} \left(\frac{\sigma_o^2}{\sigma_d^2} \right) \quad (15)$$

The effective suppression was calculated as a function of the disturbance bandwidth. The bandwidth of the disturbance added at the output of the S2 FSL simulation was increased up to 100 MHz and the results are displayed in Figure 17.

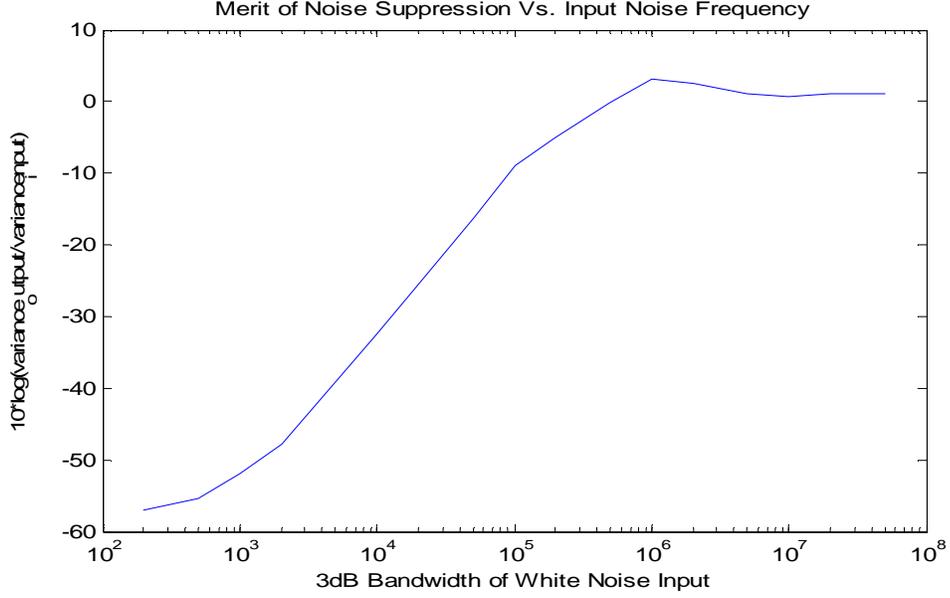


Figure 17. Merit of the noise suppression M dB The merit, M equation (15), was the ratio of the variances between the disturbance added at the output of the S2 FSL simulation and the output response of the S2 FSL.

The relationship of the transient response, the effective merit, and the line width of the laser was established and is shown in equation (16) [9].

$$\Delta_{FWHM} = 2(2\ln(2))^{.5} DV_{rms} \quad (16)$$

The slope of the tuning curve of the oscillator, D , was compared to the slope of the S2 material as a frequency discriminator. The data V_{rms} was the variance of the output response voltage, σ_0^2 . It was shown that the Gaussian line width reduction would be proportional to the square root of the effective suppression, M [9].

$$FWHM_{locked} = FWHM_{unlocked} \sqrt{|M|} \quad (17)$$

Stability Versus Computational Latency

Computational latency, k , was defined as the time required to calculate the estimated states with the LSE and the feedback gain with the DLQR. The digital control algorithm required a number of clock cycles to complete the calculations. The computational latency of the LSE plus DLQR control system was investigated to determine the stability of the S2 FSL. The closed loop stability of the S2 FSL, due to the increased computational latency, is shown in Figure 18. The minimum computational latency was the first point, $k = 1$, with a latency of 200 ns, which corresponded to one clock cycle of the sample frequency 5 MHz. The last point, $k = 8$, was the last point of stability with a transient time of 1.6 μ s to reach 60 dB of Suppression. The system was unstable at $k=9$.

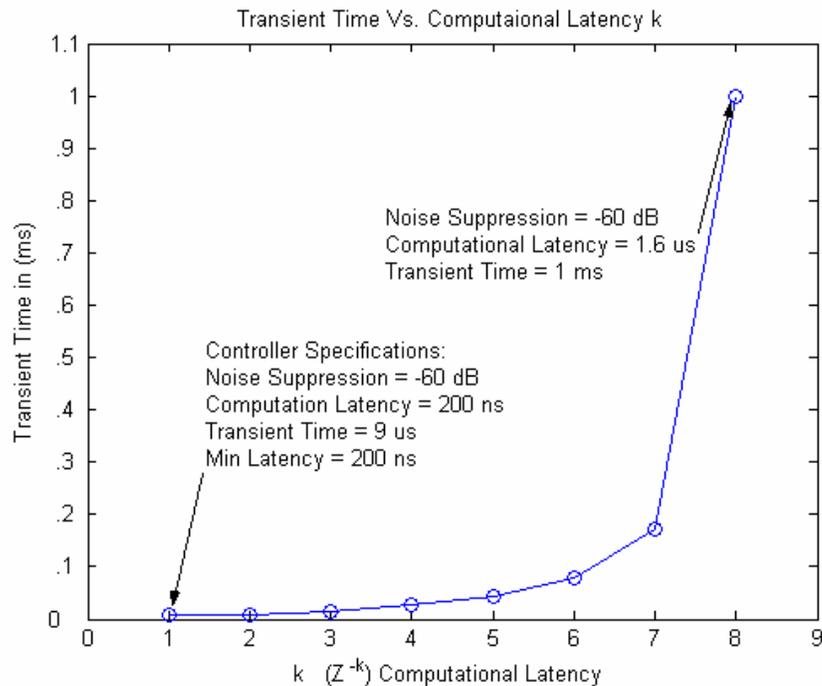


Figure 18. Transient time required for 60 dB noise suppression of the closed system due to the computational latency k .

The comparison of the transient response of the closed loop system due to the minimum and maximum computational latency, k , is shown in Figure 19.

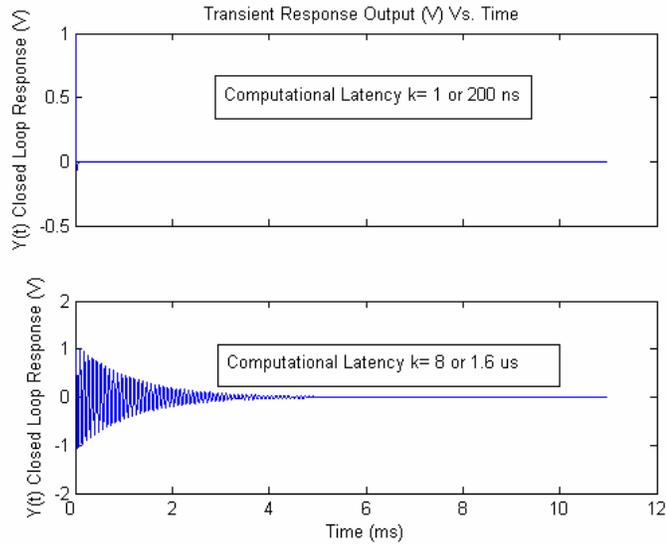


Figure 19. The transient response of the closed loop output of the S2 FSL simulation controlled by LSE plus DLQR (a) Computational latency k is 1 or 200 ns (b) Computational latency $k = 8$ or 1.6 μ s

The comparison shows the increase of the oscillatory behavior of the transient response of the S2 FSL with the increase of the computational latency. The computational latency, $k = 9$, does not oscillate, but increases to the saturation of the output of the S2 FSL, and is thus unstable.

Comparing Noise Floor Voltages

The signal to noise ratio (SNR) of the ECDL, through the frequency discriminator, was simply analyzed as the output of the photodetector. The analysis was designed to establish a metric for comparing noise floor voltages out of the frequency discriminator and the least significant bit of the ADC of the digital hardware.

The photodetector, manufactured by New Focus, used in the S2 frequency discriminator was the AC coupled model 1801. Data sheets indicated noise was constant at $30 \text{ pW}/(\text{Hz})^{1/2}$ with a conversion factor of $2.4 \times 10^4 \text{ V}/\text{W}$ for frequencies above 10 MHz. RF of the S2 FSL was 20 MHz. Bandwidth of the error signal was estimated at 1MHz. Calculation of the noise floor voltage of the photodetector is seen in equation (18).

$$\text{Noise_Voltage}(1\text{MHz_bandwidth}) = 30 \frac{\text{pW}}{\sqrt{\text{Hz}}} 2.4 \times 10^4 \frac{\text{V}}{\text{W}} \sqrt{1\text{MHz}} = 720 \mu\text{V} \quad (18)$$

The prototype board acquired for the implementation for digital control was the XtremeDSP Development Kit by Nallatech. The XtremeDSP digital conversion channel converts analog signal into 14 bit digital word at a maximum speed of 105 MHz. The ADC is manufactured by Analog Devices and is the AD6645. The digital to analog converter (DAC) is the AD9772A which converts a 14 bit digital word into analog signal at a maximum sample rate of 160 MHz.

The full scale voltage is 1 Vpp. The 14 bit conversion of the channel converts 16,384 unique codes of the magnitude of the input analog signal. An analog signal, with a 2 volt differential, has a LSB resolution of 122 uV. The resolution of the LSB is well below the noise floor of the photo detector's 720 uV. The first 3 LSB of the digital channel converted the noise floor out of the photo-detector. The remaining 11 bits converted the error signal. This indicated that 68 dB of SNR for the digital hardware is available for conversion of the frequency discriminated signal.

VHDL of the LSE plus DLQR

The FPGA on the XtremeDSP development kit by Nallatech is the VirtexIV SX35 by Xilinx. VHDL was the software used to allocate the hardware resources, implementing terms functionally equivalent to the LSE plus DLQR equations (12) and (13). Both equations are shown below.

$$u[k] = -G \hat{x}[k]$$

$$\hat{x}[k+1] = \hat{A} \hat{x}[k] + \hat{B} u[k] + Hy[k]$$

The DSP resource available on the VirtexIV, to implement fixed point multiplication, addition, and subtraction required of the LSE plus DLQR, was the DSP48 slice, shown in Figure 20.

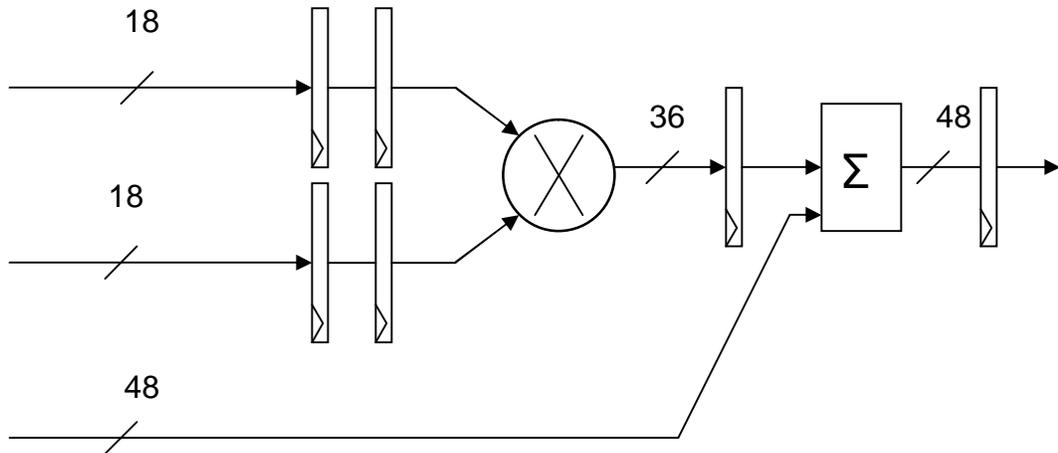


Figure 20. Xilinx's digital signal processing resource is the DSP48 slice

The DSP48 is a single fixed point multiplication and accumulator. Input precision to the DSP48 slice product is 18 bits and the resulting output precision of product is 36 bits. Accumulation output can be accurate to 48 bits. Four registers are available in the hardware path and are removable from a design for flexibility. According to the datasheet, there are 192 available DSP48 slices in the SX35 with a speed grade of -11 and can operate at 400 MHz.

The block diagram representation of the LSE plus DLQR is shown in Figure 21 to aid the description of the timing constraints. The estimated states $\hat{x}[k]$ are a single sample clock delay, block Z^{-1} , away from the estimation prediction, $\hat{x}[k+1]$. The terms of the estimated prediction are the observation, the state transition, and the input. The three terms need to be calculated and available at the summation during every clock transition of the sample frequency clock.

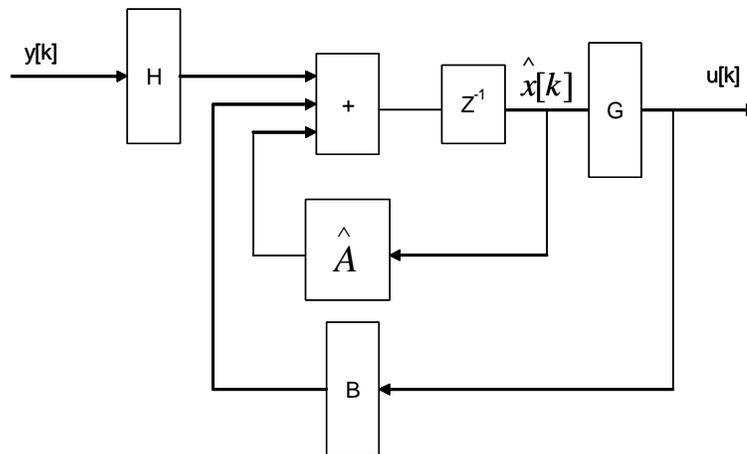


Figure 21. Block diagram of the LSE plus DLQR showing the timing in relation to sample clock

Three different clocks were required to calculate the LSE plus the DLQR. A single data path through the LSE plus DLQR is shown graphically in Figure 22.

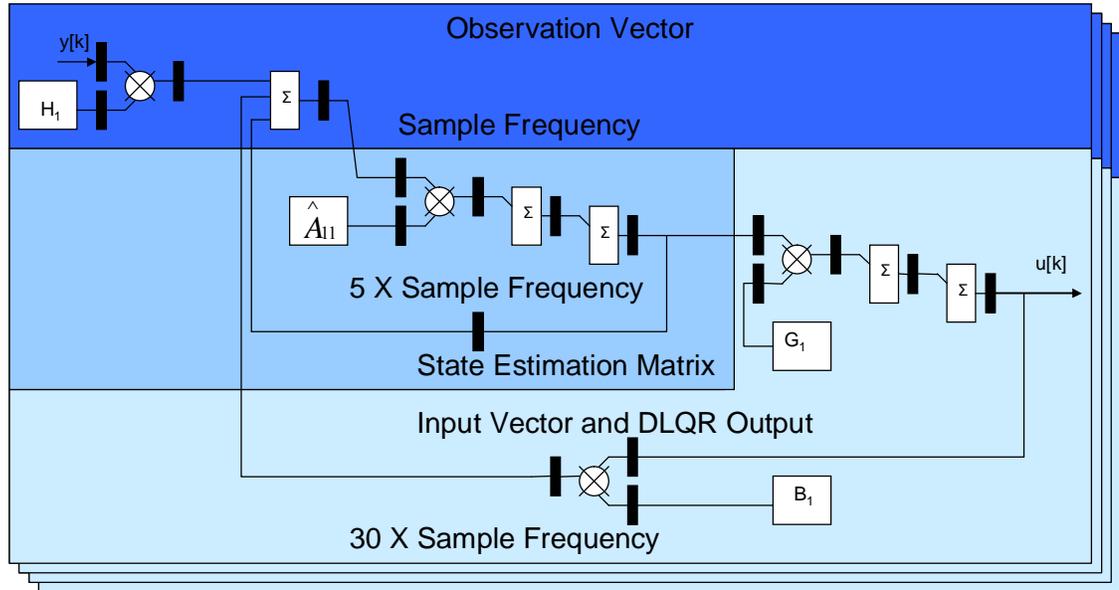


Figure 22. The block diagram of a single data path through the LSE plus DLQR, equations (12) and (13) Three clock frequencies are necessary for the calculation.

The darkest shade in the top third of Figure 22 is the observation term clocking at the data sampling frequency. The middle shade is the state transition matrix clocking at five times the sample frequency. The lightest shade is the input term clocking at thirty times the sample frequency. Pipelining was necessary to shorten the physical length between operations, and thus increased the clock speed. Pipelining is seen as the black rectangles in the data paths representing data registers. The estimated state transition term has five registers within the feedback path requiring clock speed five times faster than the sample frequency. The input term was fed back from the DLQR and in turn fed the estimated state transition term that had six registers. The input term

summing into the LSE requires a clock running thirty times faster than the sample frequency.

The DSP48 slice utilized two registers for every multiplication and could be clocked at a maximum of 240 MHz. The sampling frequency was set by the maximum speed obtained by the DSP48 slice and the number of clocks necessary to finish the calculation. The thirty clock cycles necessary for the calculation of the input term reduced the sample frequency to 8 MHz. This compares well to the successful time based simulation in Simulink where 5 MHz was the sample frequency. A discrepancy between the two architectures (the VHDL description and the Simulink) was the width of the fixed point length that implemented the calculations. Simulink required 64 bit precision and the VHDL had 36 bit. Functional equivalence of the LSE plus DLQR was not completed in the VHDL simulation and therefore a comparison of the magnitude and timing was not completed. Two DSP48 slices could implement the 64 bit fixed point precision required of Simulink simulations.

Infinite Impulse Response Low Pass Filter

The design of the second controller was completed for implementation in the FPGA and established the S2 FSL with digital control. The design criteria was to implement 20 dB gain with a cutoff frequency at 100 kHz.

The phase of the frequency response of the LP IIR in the digital implementation was the time delay required for DSP. Pure time delay through the digital hardware of the controller limits the compensated bandwidth of the closed loop S2 FSL [3].

The inherent time delay of the XtremeDSP board was measured to indicate the minimum delay through the system. The inherent delay indicated only the time it took to convert an analog signal through the ADC and to output the equivalent analog through the DAC. Therefore, any digital controller implemented in the XtremeDSP would incur more time delay.

A VHDL program, supplied with the XtremeDSP board, configured the FPGA and implemented a 14 bit digital converting channel sampling at 105 MHz. The digital hardware converted an analog signal increasing in frequency, sourced by a network analyzer, and returned an analog equivalent for analysis. A linear response of the phase due to the fixed time delay necessary for conversion, was seen on the network analyzer. The phase lag at 1 MHz measured 84 degrees. Comparing 90 degrees of a 1 MHz sinusoid to the time required to complete a cycle, $1\mu\text{s}$, determined 250 ns per 90 degrees of phase lag. The minimum time delay calculated through the digital hardware of the Nallatech prototype board was 233 ns. The sampling frequency of 105 MHz corresponded to 9.5 ns per clock and therefore the sample delay was 25 clock cycles. The magnitude and phase of the frequency response of a digital conversion is shown in Figure 23.

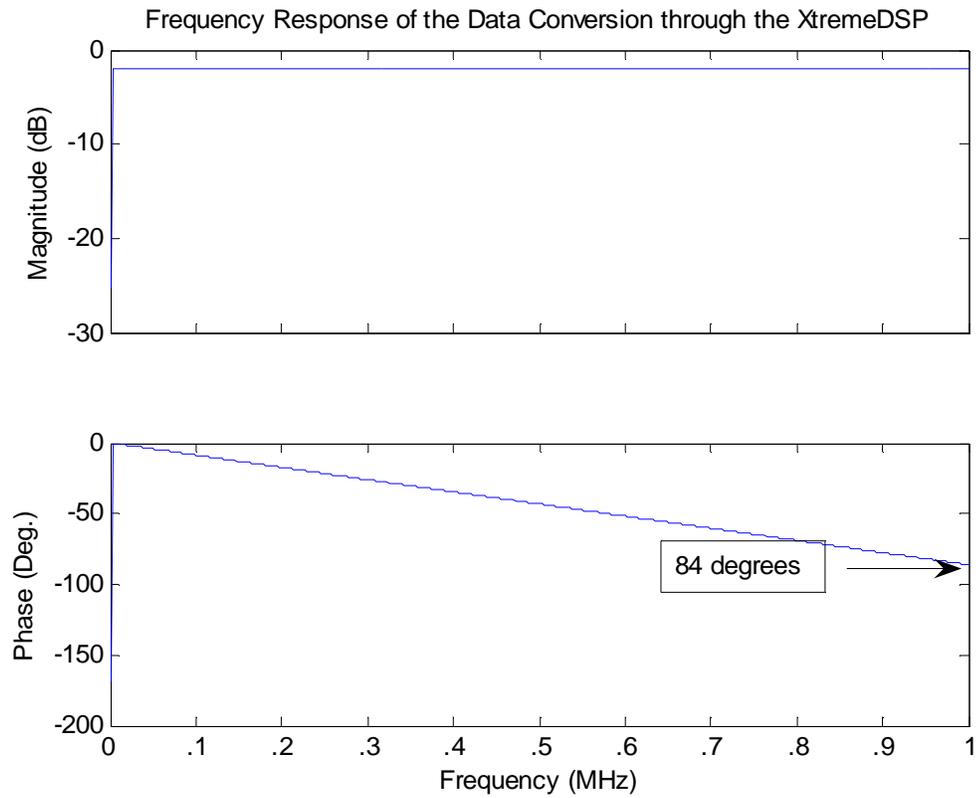


Figure 23. Network analyzer display of the linear frequency response of the XtremeDSP board, that converted an analog to digital signal and then converted back to analog.

RESULTS OF FPGA DIGITAL CONTROL IMPLEMENTATION

Four functions for a complete analog electronic feedback control were mentioned in the chapter System Analysis in the section Analog Feedback Control: the RF generation, the phase adjustment, the RF demodulation, and the analog compensation filter. The digital hardware implemented two systems to implement these four subsystems for complete digital control of the S2 FSL. First was the implementation of the first three subsystems: RF modulation, phase adjustment and RF demodulation . Second was the implementation of a digital compensation, LP IIR filter, that compensated the error signal for feedback into the current modulation. The modulation index was 50 GHz/A and was the input to the frequency tunable ECDL. These two systems, necessary for a complete digital electronic feedback controller, were tested separately. The error signal for the second test was produced with the present analog RF modulation and demodulation with phase adjustment circuitry.

Digital Frequency Down Conversion

RF generation was completed with a numerically controlled oscillator (NCO). The NCO was clocked at the maximum sampling frequency of the ADC, 105 MHz. The NCO generated a 15 MHz oscillator calculated from seven phase values per cycle stored in a lookup table. The approximate phase values were 52 degrees apart and were the input to a sine wave generator, a predefined core by Xilinx's integrated software environment (ISE) 7.1. The outputs of the sine wave generator were 14 bit magnitude values of the RF oscillator. Phase adjustment of the NCO was simply the

addition of a constant value to the seven phase values in the lookup table. The VHDL, describing the digital down conversion, is in Appendix B. The block diagram of the completed digital down converter is seen in Figure 24

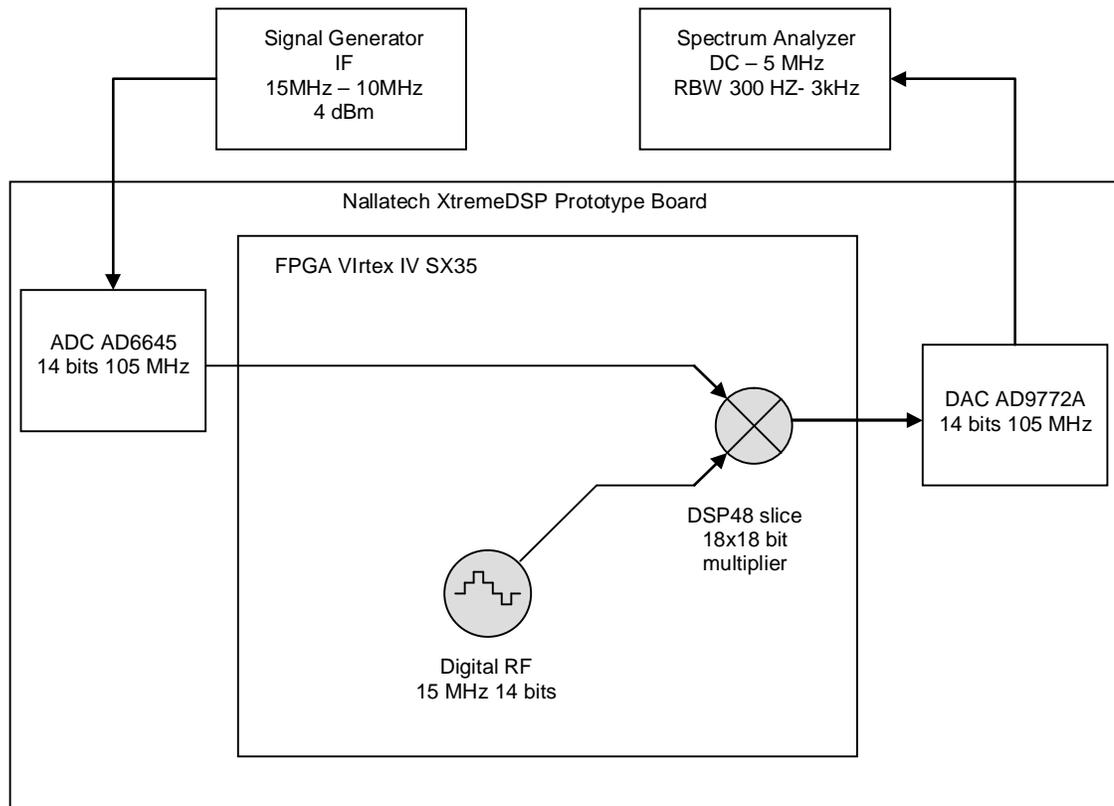


Figure 24. The Block diagram of the RF modulation and demodulation

Digital down conversion was the process of demodulation of the error signal from the RF oscillator frequency. Demodulation was the mixing of the RF signal with an intermediate frequency (IF) signal. Mixing in the sample domain is represented here by the multiplication of the 14 bit RF digital word with the 14 bit IF digital word.

Results of the multiplication of the sampled signals were the frequency difference between the IF and RF.

The difference between the frequency output of the digital RF oscillator, 15 MHz, and the present analog RF oscillator within the S2 FSL, 20 MHz, was unfortunate; however, it does not represent any design limitations.

The IF was a signal produced through a standard signal generator. The magnitude of the IF was .354 mVpp and the output frequency was adjusted from 10 to 15 MHz. Results of the digital down converter for two specific IF frequencies, 14.9 MHz and 10 MHz, are shown in Figure 25 (a) and (b).

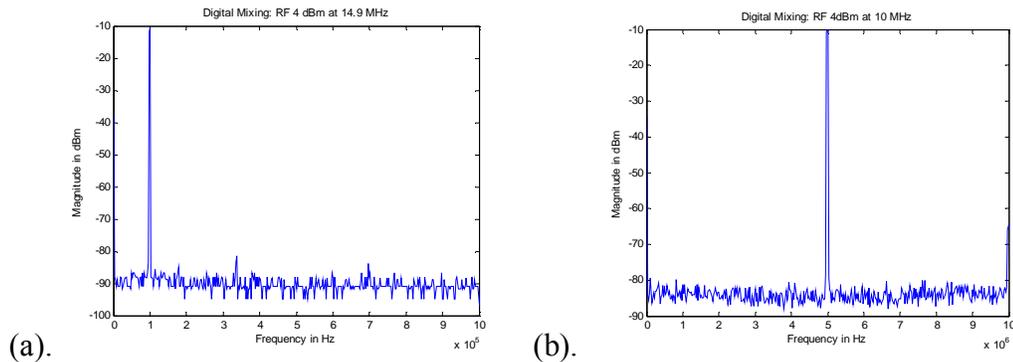


Figure 25. (a) Digital mixing of an IF 4 dBm at 14.9 MHz measured with a resolution bandwidth of 300 Hz (b) Digital mixing of an IF signal 4 dBm at 10 MHz measured with a resolution bandwidth of 3 kHz.

Displayed on the spectrum analyzer was a 100 kHz signal, which was the difference frequency between a 15 MHz RF and a 14.9 MHz IF. The amplitude of the output signal was -10 dBm with the noise floor, measured with a resolution bandwidth of 300 Hz, which was -90 dBm. A second display on the spectrum analyzer was a 5 MHz signal the difference frequency between a 15 MHz RF and a 10 MHz IF. The

amplitude of the output signal was -10 dBm with the noise floor, measured with a resolution bandwidth of 3 kHz, which was -84 dBm.

The digital down converter had a conversion loss of 14 dBm and a local oscillator leakage of -58 dBm. The present analog down converter had a conversion loss of 16.5 dBm and a local oscillator leakage of -71 dBm.

Digital Low Pass Infinite Impulse Response Filter

The successful simulation of the LP IIR filter led to the implementation of the VHDL into the digital hardware in the FPGA. The VHDL of the LP IIR compensator was pasted into the same example program used to test the time delay through the XtremeDSP, Appendix C. The digital controller was tested on the NA to obtain the frequency response, shown in Figure 26.

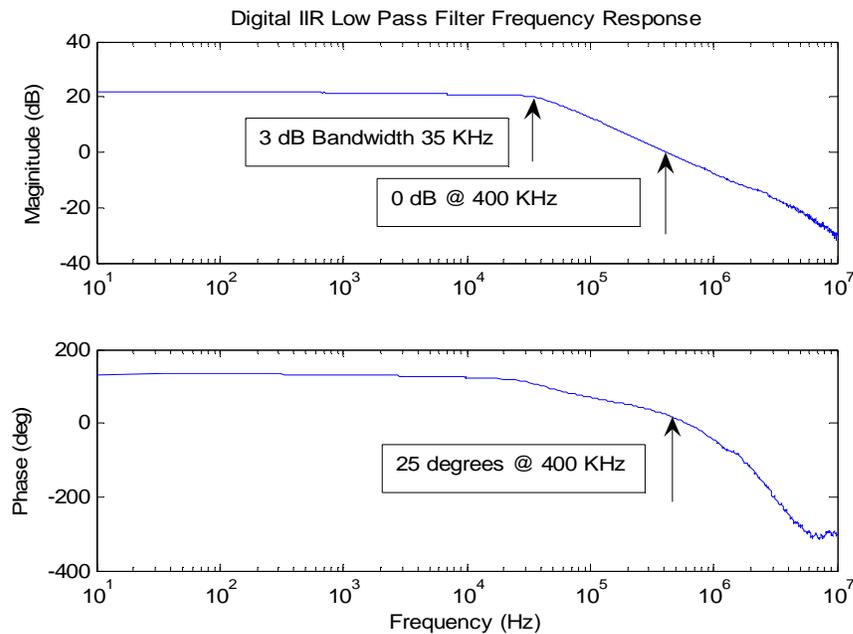


Figure 26. The frequency response of the LP IIR filter know as the digital compensation

S2 FSL with Digital IIR LPF

The most directly tested digital hardware in the S2 FSL was the closed loop compensation with the LP IIR filter. The LP IIR filter replaced the analog compensation filter in the active electronic feedback control. The error signal produced through RF modulation and demodulation was the analog circuitry of the present active feedback control. The ADC, AD6645, converted the error signal out of the analog circuitry. The DSP of the LP IIR filter compensated the error signal, which was fourteen bits sampled at 52.5 MHz. The on board DAC, AD9772A, converted the digital compensation into an analog signal fed back into the current modulation input of the frequency tunable ECDL.

An additionally constructed optical path provided the diagnostics to measure the spectral hole burned with the S2 FSL. The optical splitter routed the second optical transmission out of the ECDL back through the S2 material and burned a second spectral hole. The diagnostic path is the outer arrows seen in Figure 27.

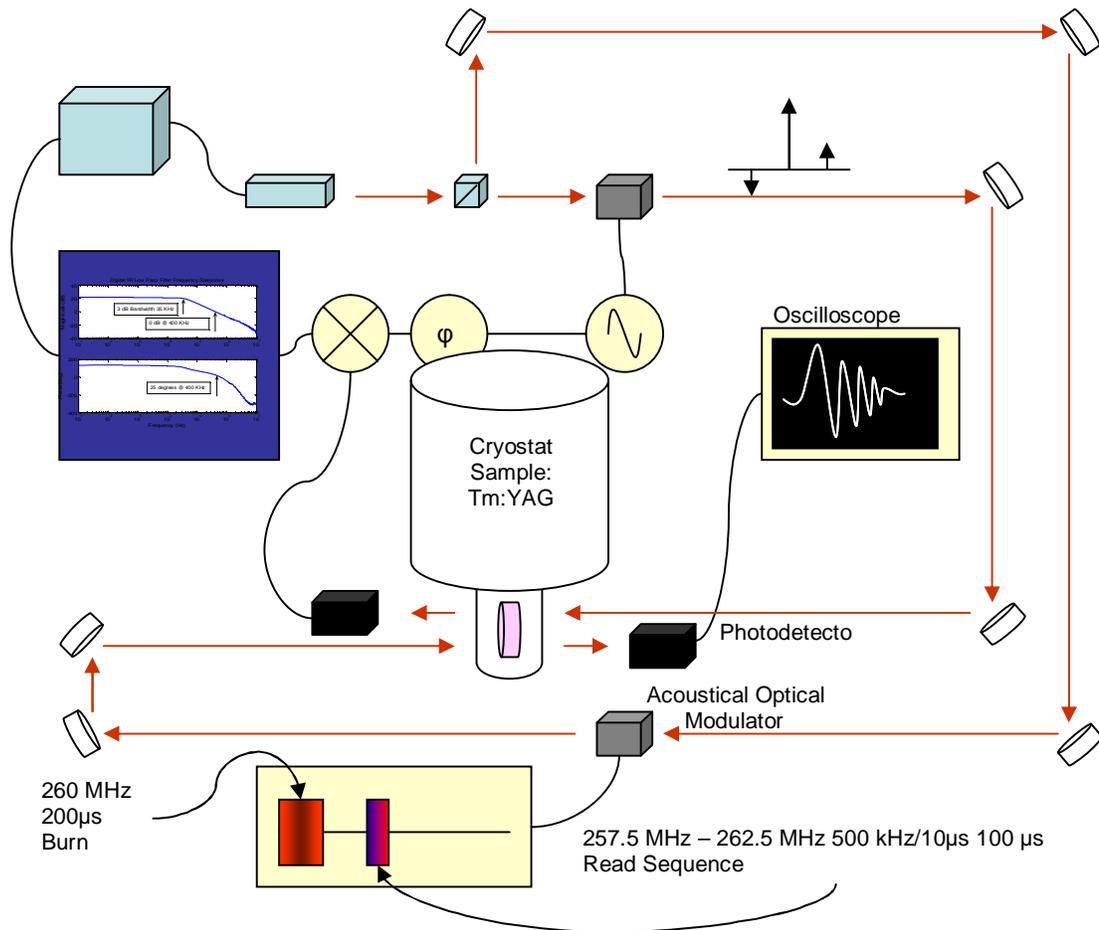


Figure 27. The inner optical path represents the S2 FSL controlled by the digital IIR LPF. The outer optical path represents the diagnostics. The diagnostic path measured the optical resolution free running ECDL, digitally controlled S2 FSL, and analog controlled S2 FSL.

The diagnostic path burned a second spectral hole into the S2 material into another spatial location in order to measure the frequency stability of the system. The spectral hole represented a narrow absorption of output frequencies from the ECDL. Spectral width of the laser was represented by the optical transmission through the spectral hole while the frequency output of the ECDL was tuned in time. This technique of frequency tuning the ECDL over the frequencies of the spectral hole maps the frequency transmission in time; therefore, an oscilloscope can indirectly view the

spectral width of the laser. Three configurations measured were the ECDL without electronic feedback control, ECDL with digital compensation of the LP IIR filter, and ECDL with the present analog feedback control.

The spectral hole was burned into the S2 material by modulating a gated sine onto the optical carrier frequency with an acoustical optical modulator (AOM). The gated sine was a 260 MHz frequency and burned for a 200 μ s duration.

Reading the spectral hole consisted of a linearly increasing frequency sinusoid known as a frequency chirped signal. The starting frequency of the chirp was 257.5 MHz and the ending frequency was 262.5 MHz. Chirp rate of the read sequence was programmed for 500 kHz/10 μ s, which lasted 100 μ s. The spectral hole measured in time at this relatively fast chirp rate contains a secondary interaction between the S2 material and the quadratic phase of the chirp signal. The secondary interaction presents as a ringing of the spectral hole. Chang et al. [10] have uniquely identified this effect, and the method of recovery used post processing, to identify the spectral hole in the S2 material at 260 MHz.

Three spectral holes were measured with the experimental setup described. First, the complete analog active control frequency stabilized the S2 ECDL. The second experiment was the replacement of the analog filter with the digital LP IIR filter frequency stabilizing the S2 ECDL. The third experiment was the free running ECDL. All three spectral features were measured with an oscilloscope and plotted together for comparison in Figure 28. The signal with the highest peak and uniform oscillations, due to the secondary interactions, was the present analog active control feedback of

the S2 FSL. The signal with the second highest peak and less uniform oscillation was the digitally compensated S2 FSL. The signal with the lowest peak and little uniform oscillation was the free running ECDL.

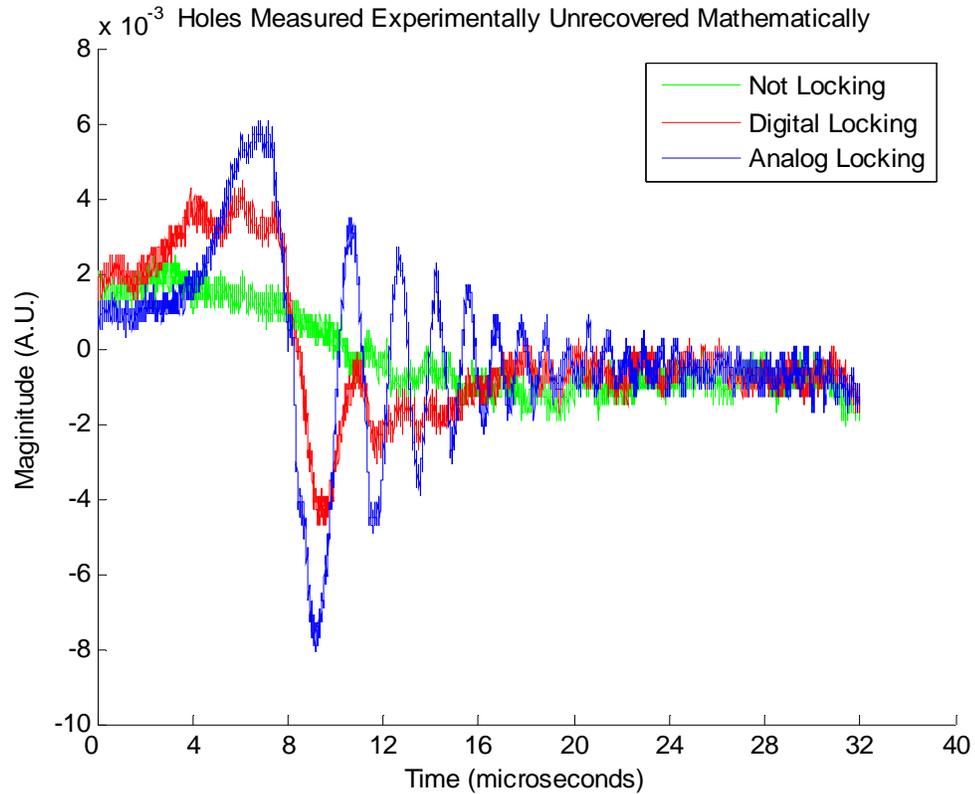


Figure 28. The optical transmissions of a frequency chirped hole of the S2 FSL with analog, digital, and free running techniques

The signals were captured from the oscilloscope and post processed. Each signal was recovered by post processing, described by Chang et al. [10], for the measurement of spectral hole width burned by the ECDL, known as full width half maximum (FWHM). These recovered holes are shown in Figure 29. The results measured the

analog S2 FSL FWHM at 55.8 KHz. The digital IIR LPF FWHM was 99.2 KHz. The free running ECDL FWHM was 590 KHz.

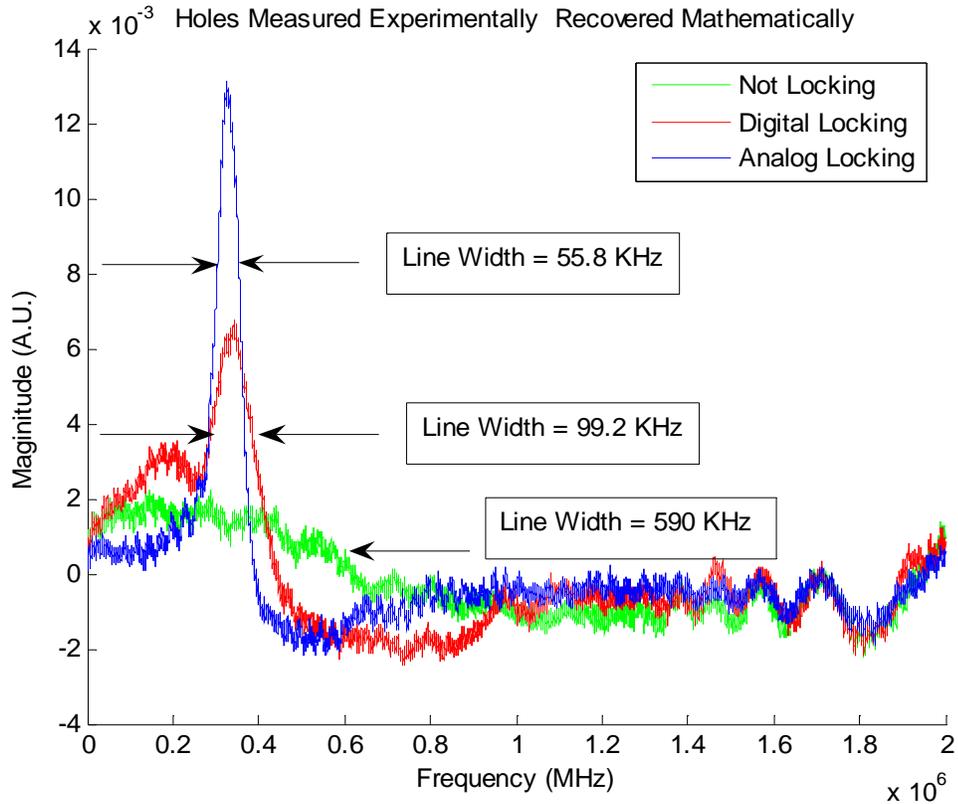


Figure 29 The mathematically recovered line width of the hole showing transmission magnitude versus frequency.

The simple LP IIR compensation implemented in digital hardware compared qualitatively this single experiment, which underperformed the present analog active feedback control by a factor of two. The digital compensation outperformed the free running ECDL by a factor of six.

CONCLUSIONS

Simulation of the S2 FSL with the time designed compensator showed 60 dB of suppression at low frequency of the inherent noise of the S2 FSL (Figure 16). The overall bandwidth of the time designed compensator in simulation was approximately 1 MHz in Figure 17. The low frequency gain and control bandwidth of the compensation greatly reduces the effect of noise inherent on the free running ECDL. The desired closed loop response, having been achieved in simulation, is a good indication for successful implementation in digital hardware.

Successful implementation of the two digital systems, which represented the complete active feedback control for S2 FSL, provides a platform for fast evaluation of future control algorithms. Solutions being implemented in the available platform would decrease both the time and expense between control algorithm implemented for performance increases.

Further development of the time based model, describing additional states out of the S2 FSL, would eliminate the need for calculation of a full state observation of the system, which was a heavy calculation burden. Time domain designed compensators for an LTI system can be as simple as multiplying constants by the sampled states of the S2 FSL. This simple digital compensation would decrease the time required of the compensator, which would increase the bandwidth, and thus narrow the laser line width of a controlled ECDL.

REFERENCES CITED

-
- 1 Kallaayoun, Ahmed. Work Completed at Spectrum Lab MSU and personnel Conversations 2004-2005
 - 2 Bottger, Thomas “Laser Frequency Stabilization to Spectral Hole Burning Frequency References in Erbium-Doped Crystals: Material and Device Optimization”, Dissertation of Doctor of Philosophy Montana State University, Bozeman Montana, April 2002
 - 3 O. Mor, A. Arie “Performance Analysis of Drever-Hall Laser frequency Stabilization Using a Proportional + Integral Servo,” IEEE Journal of Quantum Electronics, vol. 33 pp. 532-539, 1997.
 - 4 Roos, Peter Aaron “The Diode-Pumped Continuous-Wave Raman Laser: Classical, Quantum, And Thermo Optical Fundamentals”, Dissertation of Doctor of Philosophy Montana State University, Bozeman Montana, April 2002
 - 5 Discussion with Pete Roos 2/18/06, 3/15/06
 - 6 Friedland, Bernard. Control System Design-An Introduction to State-Space Methods. McGraw Hill USA 1986
 - 7 R.W. Gunderson “ Lecture EE 522 Montana State University ”, April 14, 2004
 - 8 Olsen, Andy. Personnel Conversation 22 Feb. 2005
 - 9 R.S. Elliot, R. Roy, S.J. Smith “Extracavity Laser Band-Shape and Bandwidth Modification” Physical Review A, vol. 26 no. 1, July 1982
 - 10 T. Chang, M. Tian, R. K. Mohan, C. Renner, K. Merkel, Wm. R. Babbitt, “Recovery of spectral features readout with frequency- chirped laser field” Optics Letters, Vol. 30, No 10, pp. 1129-1131, May 2005.

APPENDICES

APPENDIX A

Matlab Code S2 FSL

```

%% Author: Cy Drollinger
%% Date: 9/19/04
%% Purpose: Control of TF of Plant-Laser Utilizing LQR and full State Estimation
%% Information: Taken from Ahmed Khallaayoun's work

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%                      CONSTANTS OF THE SYSTEM                      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KiKl      = 5e10;          %Hz/A
D         = 280e-6;       %A/Hz providing adc optimization

Fc        = 200e3*2*pi;   %Rad
Fl        = 100*2*pi;     %Rad
Fh        = 7.35e4*2*pi;  %Rad

Ts        = 2e-7;        %sec, Sampling time

s         = tf('s');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%                      TRANFER FUNCTION open loop                  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Laser_tf = KiKl*Fc^2/((s+Fc)^2          % Two poles at 200 KHz

discriminator_tf = D*Fh*s/((s+Fh)*(s+Fl)) % Two poles - high at 73.5 KHz
% The second - low at 100 Hz
open_loop_laser_tf = Laser_tf*discriminator_tf % The overall TF of the laser plant

DIGITAL_LASER_TF = c2d(open_loop_laser_tf,Ts) %The analog to digital 5MHz sampling

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%                      STATE SPACE OF SYSTEM                      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% small letters represents analog system                          %%
%% Capital letters represents digital                              %%
[a,b,c,d] = ssdata(open_loop_laser_tf); %State Space translation anaolog

[A,B,C,D,E,Ts] = dssdata(DIGITAL_LASER_TF) %State Space digital 5MHz sampling

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%                      SYMMETRIC ROOT LOCUS                       %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Symmetric Root Locus utilized to determine                    %%
%% the gain to establish the weighting matrix Q                  %%
%% performance index J = integration(x'*Q*x + u'R*u)dt          %%
open_loop_laser_inv_tf = Kc/(((s)+Fh)*((s)+Fl))*Gain*Fh*(-s)/((-s/Fc)+1)^2;

DIGITAL_LASER_INV_TF = c2d(open_loop_laser_inv_tf,Ts)

```

```

srl = (open_loop_laser_tf*open_loop_laser_inv_tf);

RL = (DIGITAL_LASER_TF*DIGITAL_LASER_INV_TF);

figure

rlocus(srl)

q = 4e-5 %q selected from the sym root locus

Q = q*c'*c %wiegthed matrix for LQR

r =1

[G,s,e] = lqr(a,b,Q,r)

observer_poles = [-1.9e7 -1.8e7 -1.7e7 -1.6e7];

h = place(a',c',observer_poles)'

ahat = a - h*c

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% OBSERVER POLES %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Z_Poles = exp(e*Ts)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CONTROL GAINS %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
K = place(A,B,Z_Poles)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FULL STATE OBSERVER %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Observer_Z_poles = [.81 .82 .83 .84]

H = place(A',C',Observer_Z_poles)'

AHAT = A - H*C

```

APPENDIX B

VHDL Digital Mixing

```

-----
-- Title       : adc_to_dac_hookup
-- Project     : adc_to_dac_hookup
-----
-- File        : adc_to_dac_hookup.vhd
-- Author      : <derekstark@DELL2000AXP>
-- Company     : Nallatech Ltd
-- Created     : 2003-11-12
-- Last update : 2003-11-12
-- Platform    : DIME-II
-- Standard    : VHDL'87
-----
-- Description: Simple adc to dac hookup with a count being driven
-- out the LEDs. The design simply captures data from the ADCs
-- (registering --- it) and then outputs this on the DACs with a
-- slight conversion to change from 2s complement to offset binary.
-----
-- Copyright (c) 2003
-----
-- Revisions  :
-- Date       Version          Author          Description
-- 10/19/05   digital mixing      Cy Drollinger  15 MHz digital
--                                     mixing with ADC
-- 2003-11-12 1.0            derekstark     Created
-----

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_SIGNED.all;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
library UNISIM;
use UNISIM.VComponents.all;

entity mixer is
  port (
    oscillator -- main clock input from
    CLK1_FB    : in  std_logic;    -- main reset input from mb
    RESET1     : in  std_logic;    -- configuration done signal
    CONFIG_DONE : out std_logic;   -- dac 14 bit data outputs
    DAC1_D     : out std_logic_vector(13 downto 0);
    DAC2_D     : out std_logic_vector(13 downto 0);
    ADC1_D     : in  std_logic_vector(13 downto 0);
    ADC2_D     : in  std_logic_vector(13 downto 0);
    DAC1_RESET : out std_logic;
    DAC2_RESET : out std_logic;   -- dac reset signals
    DAC1_MOD0  : out std_logic;
    DAC1_MOD1  : out std_logic;   -- dac setup

```

```

DAC2_MOD0    : out std_logic;
DAC2_MOD1    : out std_logic;    -- dac clock divider setup
DAC1_DIV0    : out std_logic;
DAC1_DIV1    : out std_logic;
DAC2_DIV0    : out std_logic;
DAC2_DIV1    : out std_logic;    -- led flash signals
LED1_Red     : out std_logic;
LED2_Red     : out std_logic;
LED1_Green   : out std_logic;
LED2_Green   : out std_logic;
);
end mixer;

```

architecture Behavioral of mixer is

```

-- clock components
component BUFG
  port (
    I : in  std_logic;
    O : out std_logic
  );
end component;
component IBUFG
  port (
    I : in  std_logic;
    O : out std_logic
  );
end component;
component DCM
  generic (
    DLL_FREQUENCY_MODE      : string := "LOW";
    DUTY_CYCLE_CORRECTION   : string := "TRUE";
    STARTUP_WAIT             : string := "FALSE"
  );
  port (
    CLKIN      : in  std_logic;
    CLKFB      : in  std_logic;
    DSSEN      : in  std_logic;
    PSINCDEC   : in  std_logic;
    PSEN       : in  std_logic;
    PSCLK      : in  std_logic;
    RST        : in  std_logic;
    CLK0       : out std_logic;
    CLK90      : out std_logic;
    CLK180     : out std_logic;
    CLK270     : out std_logic;
    CLK2X      : out std_logic;
    CLK2X180   : out std_logic;
    CLKDV      : out std_logic;
    CLKFX      : out std_logic;
    CLKFX180   : out std_logic;
    LOCKED     : out std_logic;
    PSDONE     : out std_logic;
    STATUS     : out std_logic_vector(7 downto 0)
  );

```

```

    );
  end component;
-- end of clock components

  COMPONENT digital_lo
  PORT(
    clk : IN std_logic;
    digital_15MHz : OUT std_logic_vector(13 downto 0)
  );
  END COMPONENT;

-- internal clock and reset signals
signal CLKIN_OSC, CLKFB_OSC, CLK_OSC, RESET, RSTl : std_logic;

-- temporary registers
signal ADC1, ADC2, digital_15MHz: std_logic_vector(13 downto 0);
signal multiplied_data : std_logic_vector(27 downto 0);

-- common ground
signal GND : std_logic;

begin

GND <= '0';
RESET <= not RESETl;

-----clock deskew section-----

-- IBUFG Instantiation for CLK_IN
U0_IBUFG : IBUFG
  port map (
    I => CLK1_FB,
    O => CLKIN_OSC
  );
-- BUFG Instantiation for CLKFB
U0_BUFGB : BUFG
  port map (
    I => CLKFB_OSC,
    O => CLK_OSC
  );
-- DCM Instantiation for internal deskew of CLK0
U0_DCM : DCM
  port map (
    CLKIN    => CLKIN_OSC,
    CLKFB    => CLK_OSC,
    DSSEN    => GND,
    PSINCDEC => GND,
    PSEN     => GND,
    PSCLK    => GND,
    RST      => RESET,
    CLK0     => CLKFB_OSC,
    LOCKED   => RSTl
  );
-----end of clock deskew-----

```

```

Inst_digital_lo: digital_lo PORT MAP(
    clk           => CLK_OSC,
    digital_15MHz => digital_15MHz
);

-- module configured
CONFIG_DONE <= '0';

-- set low pass filter response and no zero stuffing for both DACs
DAC1_MOD0 <= '0';
DAC1_MOD1 <= '0';
DAC2_MOD0 <= '0';
DAC2_MOD1 <= '0';

-- disable resets for DACs
DAC1_RESET <= '0';
DAC2_RESET <= '0';

-- optimum settings for sampling rate
DAC1_DIV0 <= '1';
DAC1_DIV1 <= '0';
DAC2_DIV0 <= '1';
DAC2_DIV1 <= '0';

-- digital output of adc to digital input of DAC
DataRegisters : process (CLK_OSC, RSTl)
begin
    if RSTl = '0' then
        ADC1  <= "0000000000000000";
        ADC2  <= "0000000000000000";
        DAC1_D <= "0000000000000000";
        DAC2_D <= "0000000000000000";
    elsif CLK_OSC = '1' and CLK_OSC'event then
        ADC1  <= ADC1_D;
        ADC2  <= ADC2_D;
    end if;
    DAC1_D <= not(not multiplied_data(25)&multiplied_data(24 downto 12));
    DAC2_D <= not (not digital_15MHz(13) & digital_15MHz(12 downto 0));
end process;

-- digital mixing of adc and digital LO
Mixing_Register: process (CLK_OSC, RSTl)
begin
    if RSTl = '0' then
        multiplied_data <= "00000000000000000000000000000000";
    elsif CLK_OSC = '1' and CLK_OSC'event then
        multiplied_data <= digital_15MHz * ADC2;
        -- digital mixing at 15 MHz
    end if;
end process;

```

```
-----led flasher section-----  
  
-- led flash counter  
process (CLK_OSC, RST1)  
    variable COUNT : std_logic_vector(26 downto 0);  
begin  
    if RST1 = '0' then  
        COUNT      := (others => '0');          -- led assignments  
        LED1_Red   <= '0';  
        LED2_Red   <= '0';  
        LED1_Green <= '0';  
        LED2_Green <= '0';  
    elsif CLK_OSC = '1' and CLK_OSC'event then  
        COUNT      := COUNT + 1;              -- led assignments  
        LED1_Red   <= COUNT(26);  
        LED2_Red   <= COUNT(25);  
        LED1_Green <= COUNT(25);  
        LED2_Green <= COUNT(26);  
    end if;  
end process;  
-----end of led flasher-----  
  
end Behavioral;
```

```

-----
-- Company: Spectrum Lab at Montana State University
-- Engineer: Cy Drollinger
--
-- Create Date:      13:32:10 10/19/05
-- Design Name:     mixer_on_nallatech
-- Module Name:     digital_lo - Behavioral
-- Project Name:
-- Target Device:   Xtreme DSP by Nallatech Virtex4 SX35 speed -10
-- Tool versions:
-- Description:     NCO 10 bit angle input
--                  14 bit amplitude output
-- Dependencies:   ADC hookup revision digital mixing
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

```

```

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
---- library UNISIM;
---- use UNISIM.VComponents.all;

```

```

entity digital_lo is
    Port ( clk           : in std_logic;
          digital_15MHz : out std_logic_vector(13 downto 0));
end digital_lo;

```

```

architecture Behavioral of digital_lo is

```

```

    Constant zero           :std_logic_vector(9 downto 0):= "0000000000";
    Constant resolution     :std_logic_vector(9 downto 0):= "0010010010";
    -- 51 deg. is 146 in a 10 bit NCO
    Constant theta_full     :std_logic_vector(9 downto 0):= "1111111111";
    -- 1024 is 360 deg. in a 10 bit NCO
    Constant last           :std_logic_vector(9 downto 0):= "1011011011";
    --7 clock cycles at 146 is 1022
    Signal theta            :std_logic_vector(9 downto 0):= "0000000000";

```

```

    component sine          -- Core

```

```

Generated 10 bit NCO
    port (
        THETA: IN std_logic_VECTOR(9 downto 0);
        CLK: IN std_logic;
        SINE: OUT std_logic_VECTOR(13 downto 0));
    -- 14 bit output - 14 bit ADC

```

```

end component;

```

```
begin

LO : sine
    port map (
        THETA => theta,
        CLK => clk,
        SINE => digital_15MHz);

theta_accum: process(clk)    -- Process Generating control of theta
begin                        -- five consecutive theta values
                            -- corresponding to 51 deg. each
    if rising_edge(clk) then
        if theta < last then
            theta <= theta + resolution;
        else
            theta <= zero;
        end if;
    end if;
end process theta_accum;
end behavioral;
```

APPENDIX C

VHDL IIR LPF

```

-----
-- Title       : adc_to_dac_hookup
-- Project     : adc_to_dac_hookup
-----
-- File        : adc_to_dac_hookup.vhd
-- Author      : <derekstark@DELL2000AXP>
-- Company     : Nallatech Ltd
-- Created     : 2003-11-12
-- Last update : 2003-11-12
-- Platform    : DIME-II
-- Standard    : VHDL'87
-----
-- Description: Simple adc to dac hookup with a count being driven
-- out the LEDs. The design simply captures data from the ADCs
-- (registering it) and then outputs this on the DACs with a slight
-- conversion to change from 2s complement to offset binary.
-----
-- Copyright (c) 2003
-----
-- Revisions  :
-- Date        Version  Author          Description
-- 1/21/06     IIR LPF  Cy Drollinger  Implements a Direct form II
--              IIR FILTER 20 dB Gain
--              low pass filter to 35 kHz
--              Sample frequency 52.5 MHz
-- 2003-11-12  1.0      derekstark    Created
-----

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
library UNISIM;
use UNISIM.VComponents.all;

```

```

entity Toplevel is
  port (
    CLK1_FB      : in  std_logic;
    RESET1      : in  std_logic;
    CONFIG_DONE  : out std_logic;
    DAC1_D       : out std_logic_vector(13 downto 0);
    DAC2_D       : out std_logic_vector(13 downto 0);
    ADC1_D       : in  std_logic_vector(13 downto 0);
    ADC2_D       : in  std_logic_vector(13 downto 0);
    DAC1_RESET   : out std_logic;
  );
end entity Toplevel;

```

```

DAC2_RESET : out std_logic;
            -- dac setup
DAC1_MOD0  : out std_logic;
DAC1_MOD1  : out std_logic;
DAC2_MOD0  : out std_logic;
DAC2_MOD1  : out std_logic;
            -- dac clock divider setup
DAC1_DIV0  : out std_logic;
DAC1_DIV1  : out std_logic;
DAC2_DIV0  : out std_logic;
DAC2_DIV1  : out std_logic;
            -- led flash signals
LED1_Red   : out std_logic;
LED2_Red   : out std_logic;
LED1_Green : out std_logic;
LED2_Green : out std_logic
);
end Toplevel;

architecture Behavioral of Toplevel is

            -- clock components
component BUFG
port (
    I : in  std_logic;
    O : out std_logic
);
end component;
component IBUFG
port (
    I : in  std_logic;
    O : out std_logic
);
end component;

COMPONENT iir_lpf            -- digital IIR LPF direct form II
PORT(
    clk           : IN std_logic;
    clk_4X        : IN std_logic;
    adc1_d        : IN std_logic_vector(13 downto 0);
    dac1_d        : OUT std_logic_vector(13 downto 0)
);
END COMPONENT;

component DCM
generic (
    DLL_FREQUENCY_MODE      : string := "LOW";
    DUTY_CYCLE_CORRECTION  : string := "TRUE";
    STARTUP_WAIT            : string := "FALSE";
    CLKFX_MULTIPLY          : integer := 4;
    CLKDV_DIVIDE            : real := 2.0
);
port (

```

```

        CLKIN      : in  std_logic;
        CLKFB      : in  std_logic;
        DSSEN      : in  std_logic;
        PSINCDEC   : in  std_logic;
        PSEN       : in  std_logic;
        PSCLK      : in  std_logic;
        RST        : in  std_logic;
        CLK0       : out std_logic;
        CLK90      : out std_logic;
        CLK180     : out std_logic;
        CLK270     : out std_logic;
        CLK2X      : out std_logic;
        CLK2X180   : out std_logic;
        CLKDV      : out std_logic;
        CLKFX      : out std_logic;
        CLKFX180   : out std_logic;
        LOCKED     : out std_logic;
        PSDONE     : out std_logic;
        STATUS     : out std_logic_vector(7 downto 0)
    );
    end component;
-- end of clock components

-- internal clock and reset signals
signal CLKIN_OSC, CLKFB_OSC, CLK_OSC, CLK_DIV_2, CLK_2X, RESET,
        RST1: std_logic;

-- temporary registers
signal ADC1, ADC2 : std_logic_vector(13 downto 0);

-- common ground
signal GND : std_logic;

begin

GND <= '0';

RESET <= not RESET1;

-----clock deskew section-----

-- IBUFG Instantiation for CLK_IN
U0_IBUFG : IBUFG
    port map (
        I      => CLK1_FB,
        O      => CLKIN_OSC
    );
-- BUFG Instantiation for CLKFB
U0_BUFG : BUFG
    port map (
        I      => CLKFB_OSC,
        O      => CLK_OSC
    );

```

```

-- DCM Instantiation for internal deskew of CLK0
U0_DCM : DCM
  port map (
    CLKIN    => CLKIN_OSC,
    CLKFB    => CLK_OSC,
    CLK2X    => CLK_2X,
    CLKDV    => CLK_DIV_2,
    DSSEN    => GND,
    PSINCDEC => GND,
    PSEN     => GND,
    PSCLK    => GND,
    RST      => RESET,
    CLK0     => CLKFB_OSC,
    LOCKED   => RST1
  );
-----end of clock deskew-----

Inst_iir_lpf: iir_lpf PORT MAP(
  clk      => CLK_DIV_2,
  clk_4X   => CLK_2X,
  adc1_d   => ADC1,
  dac1_d   => DAC1_d
);

-- module configured
CONFIG_DONE <= '0';

-- set low pass filter response and no zero stuffing for both DACs
DAC1_MOD0 <= '0';
DAC1_MOD1 <= '0';
DAC2_MOD0 <= '0';
DAC2_MOD1 <= '0';

-- disable resets for DACs
DAC1_RESET <= '0';
DAC2_RESET <= '0';

-- optimum settings for sampling rate
DAC1_DIV0 <= '0';
DAC1_DIV1 <= '0';
DAC2_DIV0 <= '0';
DAC2_DIV1 <= '0';

-- digital output of adc to digital input of DAC
DataRegisters : process (CLK_DIV_2, RST1)
begin
  if RST1 = '0' then
    ADC1 <= "0000000000000000";
    ADC2 <= "0000000000000000";
--   DAC1_D <= "0000000000000000";
--   DAC2_D <= "0000000000000000";
  elsif CLK_DIV_2 = '1' and CLK_DIV_2'event then
    ADC1 <= ADC1_D;
    ADC2 <= ADC2_D;
  end if;
end process;

```

```

--      DAC1_D <= not (not ADC1(13) & ADC1(12 downto 0));
      DAC2_D <= not (not ADC2(13) & ADC2(12 downto 0));
    end if;
end process;

-----led flasher section-----
-- led flash counter
process (CLK_DIV_2, RST1)
  variable COUNT : std_logic_vector(26 downto 0);
begin
  if RST1 = '0' then
    COUNT      := (others => '0');      -- led assignments
    LED1_Red   <= '0';
    LED2_Red   <= '0';
    LED1_Green <= '0';
    LED2_Green <= '0';
  elsif CLK_DIV_2 = '1' and CLK_DIV_2'event then
    COUNT      := COUNT + 1;          -- led assignments
    LED1_Red   <= COUNT(26);
    LED2_Red   <= COUNT(25);
    LED1_Green <= COUNT(25);
    LED2_Green <= COUNT(26);
  end if;
end process;

-----end of led flasher-----

end Behavioral;

```

```

-----
-- Company:          Spectrum Lab at Montana State University
-- Engineer:         Cy Drollinger
-- Create Date:      12:38:42 01/21/06
-- Design Name:      IIR_LPF
-- Module Name:      iir_lpf - Behavioral
-- Project Name:     digital control for S2 FSL
-- Target Device:    Virtex IV SX35 speed -10
-- Tool versions:    ISE 7.1
-- Description:      Low Pass Infinite Impulse Response direct form II
--                   52.5 MHz samp freq 20 dB gain 35 kHz bandwidth
-- Dependencies:     ADC to DAC hookup
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
--use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;

entity iir_lpf is
    Port ( clk      : in std_logic;    --sampling frequency 52.5 MHz
          clk_4X   : in std_logic;    --DSP processing speed
          adc1_d   : in std_logic_vector(13 downto 0);
          --input to LP II filter from error signal
          dac1_d   : out std_logic_vector(13 downto 0));
          --compensated feedback to the S2 FSL
end iir_lpf;

architecture Behavioral of iir_lpf is

    constant scale:std_logic_vector(17 downto 0):="000000000011111000";
        --scale corresponds to 20 dB gain
        --input: -35 dBm ".1 - 1mv" error signals
    constant a_21 :std_logic_vector(17 downto 0):="100000000111110000";
    constant low  :std_logic_vector(0 downto 0) :="0";
    constant high : std_logic :='1';

    signal adc      : std_logic_vector (17 downto 0):="0000000000000000000";
    signal delayed_by_one: std_logic_vector (35 downto 0):=X"000000000";
    signal delayed_one:std_logic_vector(17downto0):="0000000000000000000";

    signal scaled          : std_logic_vector (35 downto 0);
    signal scale_n_delay   : std_logic_vector (35 downto 0);
    signal forward_delay_0 : std_logic_vector (35 downto 0);
    signal forward_delay   : std_logic_vector (35 downto 0);

    signal iir_lpf          : std_logic_vector (35 downto 0);

```

```

COMPONENT multiply
PORT(
    A_IN      : IN std_logic_vector(17 downto 0);
    B_IN      : IN std_logic_vector(17 downto 0);
    CLK_IN    : IN std_logic;
    P_OUT     : OUT std_logic_vector(35 downto 0)
);
END COMPONENT;

COMPONENT addition
PORT(
    AB_IN     : IN std_logic_vector(35 downto 0);
    CLK_IN    : IN std_logic;
    C_IN      : IN std_logic_vector(35 downto 0);
    P_OUT     : OUT std_logic_vector(35 downto 0)
);
END COMPONENT;

COMPONENT subtraction
PORT(
    AB_IN     : IN std_logic_vector(35 downto 0);
    CLK_IN    : IN std_logic;
    C_IN      : IN std_logic_vector(35 downto 0);
    P_OUT     : OUT std_logic_vector(35 downto 0)
);
END COMPONENT;

begin
----- LP IIR Direct Form II First Oder filter -----

scaled_multiply: multiply PORT MAP(
    A_IN      => adc,
    B_IN      => scale,
    CLK_IN    => clk,
    P_OUT     => scaled
);

delayed_by_one_multiply: multiply PORT MAP(
    A_IN      => a_21,
    B_IN      => delayed_one,
    CLK_IN    => clk_4X,
    P_OUT     => delayed_by_one
);

filtered_addition: subtraction PORT MAP(
    AB_IN     => delayed_by_one,
    CLK_IN    => clk,
    C_IN      => scaled,
    P_OUT     => scale_n_delay
);

Inst_addition: addition PORT MAP(

```

```
        AB_IN      => forward_delay,
        CLK_IN     => clk,
        C_IN       => scale_n_delay,
        P_OUT      => iir_lpf
    );

    process(clk)
    begin
        if clk = '0' and clk'event then
            forward_delay_0 <= scale_n_delay;
            forward_delay    <= forward_delay_0;
        end if;
    end process;

    process(clk_4X)
    begin
        if clk_4X = '1' and clk_4X'event then
            delayed_one <= scale_n_delay(34 downto 17);
            dac1_d <= iir_lpf(34 downto 21);
            adc <= adc1_d(13) & adc1_d(13) & adc1_d(13) & adc1_d(13) & adc1_d;
        end if;
    end process;

end Behavioral;
```