Techniques for Generating a Steered Timing Signal in a Low- Size, Weight, and Power Clock Ensemble

by

Henry Dixon

B.S., Auburn University, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Mechanical Engineering & Applied Sciences

2021

Committee Members:

Penina Axelrad, Chair

Robert B. MacCurdy

Shalom D. Ruben

Dixon, Henry (M.S., Mechanical Engineering)

Techniques for Generating a Steered Timing Signal in a Low- Size, Weight, and Power Clock Ensemble

Thesis directed by Prof. Penina Axelrad

Precise and stable timing plays a pivotal role in communication, navigation, and remote sensing systems. High stability, low- size, weight, and power (SWaP) clocks are of interest for use in advanced positioning, navigation, and timing (PNT) systems, as a means to reduce dependence on the Global Positioning System (GPS). To compensate for the lower performance typical of low-SWaP clocks, one can use a Kalman Filter to create an ensemble of of multiple clocks; estimating a composite clock, or implicit ensemble mean (IEM) with better stability than any individual ensemble member. A stable timing source is then produced by steering an oscillator to the IEM, based on the filter estimates. The Colorado Nanosat Atomic Clock Testbed (CONTACT) project is an ongoing research effort to construct a testbed for the development and analysis of methods for this type of low-SWaP timing system.

This thesis presents and evaluates several techniques for realizing the IEM of the testbed ensemble. The process comprises two primary steps: calculation of an optimal steering control value, and generation of a steered, low phase noise, timing signal. For the optimal steering control portion, an H_∞ control approach is proposed and numerical simulation results are compared to a published LQG control method. For the signal generation portion, a nominal direct digital synthesis (DDS) design is described and compared to an alternative baseband generation approach. The latter was developed for compatibility with the current hardware platform used by the CONTACT project. Both methods are implemented and demonstrated with low frequency signals, using a myRIO digital controller. Results show comparable output performance between the two signal generation approaches.

Dedication

This thesis is dedicated to my family, friends, and coworkers who provided continued support and encouragement during this process of returning to academia to learn new skills and become a better engineer.

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my thesis advisor Dr. Penina Axelrad. I could not have asked for a better mentor to study under and learn from over the past two years. From the day she welcomed me into her research group, she not only challenged me and taught me how to be a better scientist and problem solver, but treated me like family and made Boulder feel like home. She is one of the hardest workers I've ever had the privilege of knowing, and her guidance was invaluable. I will be forever grateful for the opportunity to work for her in the COMPASS Lab.

I would also like to thank and recognize my thesis committee members, Dr. Shalom Ruben and Dr. Robert MacCurdy. They each provided great suggestions and assistance throughout this process, as well as helped me look at problems from a different perspective.

I would like to acknowledge all of the members of the CONTACT project team and my colleagues in Dr. Axelrad's lab. This thesis would not have been possible without their help.

Lastly, I would like to acknowledge Dr. David Bevly from Auburn Universtiy. He had a profound impact on my life and career, both during and after my undergraduate coursework at Auburn. I sincerely appreciate all of the support and encouragement he has provided over the years.

Contents

Chapter

1 Introduction 1

1.1 Background 1

1.2 Motivation 6

1.3 Research Contributions 11

1.4 Organization 11

2 Signal Generation System Design: Generation Subsystem 13

2.1 Overview 13

2.2 Frequency Multiplication 15

2.3 Frequency Synthesis 18

2.4 Steered Output Smoothing 20

3 Signal Generation System Design: Steering Control 24

3.1 Overview 24

3.2 Steering Control Subsystem Structure & Clock Dynamics Model 25

3.3 LQG Steering Approach 28

3.4 H_∞ Steering Approach 31

3.5 Controller Simulation & Analysis 40

3.6 Steering Analysis Conclusions 47

4 USRP Front-End Analysis & Baseband Steering System Design 49

4.1 Overview 49

4.2 N310 USRP Front-End Analysis 49

4.3 Technical Differences Between Steering Methods 52

5 Steering System Design Implementation & Performance Analysis 55

5.1 Overview 55

5.2 Steering Experiment Parameters 55

5.2.1 Nominal Design Implementation 57

5.2.2 Baseband Design Implementation 61

5.3 Steering Experiment Results 65

5.3.1 Short-Term Performance Analysis 66

5.3.2 Frequency Stability Analysis 69

6 Summary & Future Work 75

6.1 Summary & Conclusions 75

6.2 Future Work Recommendations 76

Bibliography 77

Appendix

A Additional Steering Simulation Results 81

Tables

Table

1.1 Characteristic oscillator stability ratings 4

3.1 LQG steering controller parameters 30

3.2 Process and measurement noise parameters 30

3.3 H_∞ loop-shaping parameters and control intervals 33

3.4 Robust stability test results for the synthesized H_∞ controllers 37

Figures

Figure

1.1 Basic block diagram for the clock ensemble testbed 2

1.2 Example of typical OADEV slopes 4

1.3 Characteristic frequency stability for various types of oscillators 5

1.4 Clock ensemble block diagram 6

1.5 Ettus N310 USRP 7

1.6 Ettus N310 Front-End 9

2.1 Block diagram for the signal generation system design 13

2.2 Generation subsystem block diagram 14

2.3 Block diagram for the DSP frequency multiplication method 15

2.4 Spectral power in the squared generation reference input signal 16

2.5 Frequency multiplier signal flow example 17

2.6 Integer-N PLL frequency multiplier 17

2.7 PSD comparison of frequency multiplier output signals 18

2.8 DDS block diagram 19

2.9 DDS output example 20

2.10 Block diagram for the generation subsystem smoothing PLL 21

2.11 Smoothing PLL output example 22

3.1 Steering control subsystem block diagram 25

3.2 Block diagram for the LQG state feedback control system 31
3.3 Block diagram for the H_∞ shaped plant/controller configuration 32
3.4 Singular values of the shaped and unshaped plants for H_∞ controller 1 34
3.5 Singular values of the shaped and unshaped plants for H_∞ controller 2 34
3.6 Perturbed feedback system block diagram for the H_∞ robust stabilization problem . 35
3.7 Z-domain block diagram for the closed loop H_∞ steering systems 38
3.8 State feedback structure block diagram for the H_∞ central controllers 39
3.9 Open loop and closed loop singular values for H_∞ steering controller 1 39
3.10 Open loop and closed loop singular values for H_∞ steering controller 2 40
3.11 Nominal steering scenario OADEV analysis: all controllers 41
3.12 Nominal steering scenario OADEV analysis: selected controllers 42
3.13 Phase and frequency deviations from the nominal steering scenario 43
3.14 Process noise mismodeling steering scenario OADEV analysis: selected controllers . 44
3.15 Phase and frequency deviations from the process noise mismodeling scenario 45
3.16 Measurement noise mismodeling steering scenario OADEV analysis: selected controllers 46
3.17 Phase and frequency deviations from the measurement noise mismodeling scenario . 47

4.1 Functional block diagram for the testbed hardware/software interface 49
4.2 N310 daughterboard RX front-end chain 50
4.3 N310 daughterboard TX front-end chain 51
4.4 Block diagram for the baseband steering system design 53

5.1 Block diagram for the myRIO hardware and software configuration 56
5.2 Experimental block diagram for the nominal steering system implementation 58
5.3 LabVIEW DSP chain for the nominal steering system frequency multiplier 59
5.4 LabVIEW block diagram for the DDS algorithm in the nominal design implementation 60

5.5 LabVIEW block diagram for the phase-locked loop algorithm in the nominal design implementation 60
5.6 Experimental block diagram for the baseband steering system implementation 62
5.7 Block diagrams for the downconversion and upconversion chains in the baseband system implementation 62
5.8 LabVIEW block diagram for the DDC chain 63
5.9 LabVIEW block diagram for the DDS algorithm in the baseband steering system implementation 64
5.10 LabVIEW block diagram for the phase-locked loop and single-sideband up-converter in the baseband steering system implementation 65
5.11 Spectral analysis results from the nominal design implementation 66
5.12 Spectral analysis results from the baseband design implementation 67
5.13 Experimental beat signal evolution 68
5.14 Zoomed-in views of the experimental beat signals 69
5.15 Constellations for the experimental input and output signals 70
5.16 Phase error growth calculated from both steering experiments 71
5.17 Phase-time deviations from both steering experiments 72
5.18 OADEV computed for the input signals and steered output signals over two experimental trials 73
5.19 OADEV computed for the experimental measurements between the inputs and steered outputs 74

A.1 Nominal steering scenario OADEV analysis: all controllers 81
A.2 Process noise mismodeling steering scenario OADEV analysis: all controllers 82
A.3 Measurement noise mismodeling steering scenario OADEV analysis: all controllers . 82

Chapter 1

Introduction

1.1 Background

The United States Department of Defense is currently seeking to decrease vulnerabilities in global positioning, navigation, and timing (PNT) capabilities by leveraging the development and use of complementary PNT systems and alleviating dependence on the Global Positioning System (GPS) [13]. One of the most important services GPS provides is the dissemination of precise time, time intervals, and frequency, which is critical for the proper operation of communication, navigation, and remote sensing systems. Complementary PNT systems can mitigate the reliance on GPS by providing a redundancy to this service.

Ground-based atomic clocks are considered to be the standard for timekeeping, but typically have large size, weight, and power (SWaP) requirements. These requirements preclude deployment of such clocks on typical mobile ground, air, and space platforms. Low-SWaP clocks can be used to enable complementary PNT capabilities for these types of platforms. While low-SWaP clocks typically do not exhibit the same level of stability and performance as ground-based atomic clocks, creating a composite clock from an ensemble of low-SWaP clocks may provide a means to produce a stable and robust timescale.

The research discussed in this thesis stems from the Colorado Nanosat Atomic Clock Testbed (CONTACT) project. This is an ongoing graduate research project at the University of Colorado Boulder, College of Aerospace Engineering and Applied Sciences. The objective of the CONTACT project is to develop a testbed capable of measuring, ensembling, and characterizing multiple

low-SWaP atomic clocks; then realizing the composite clock computed from a weighted average of ensemble contributions. The realization of this "paper clock" estimated from combined clock contributions produces a timing signal with better frequency stability than any individual member clock [48] [57] [24]. A high level functional block diagram for the testbed system is provided in Figure 1.1.

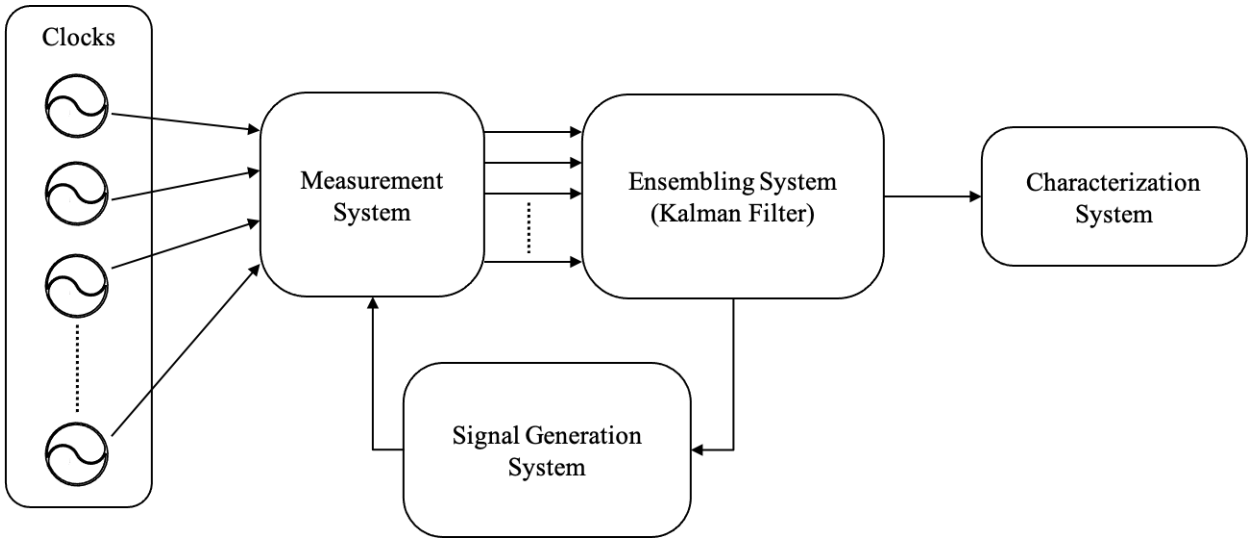

Figure 1.1: Basic block diagram for the clock ensemble testbed.

The testbed includes the ensemble members, a measurement system, ensembling system, signal generation system, and characterization system. The current ensemble members are made up of three or more *SA.45s* Chip Scale Atomic Clocks (CSACs), which are the low-SWaP member clocks. The measurement system computes phase measurements for each ensemble member, using methods described by [51] - comparing them against the clock selected as the ensemble reference. The phase measurements are input the ensembling system. A Kalman Filter performs the ensembling operations, following techniques described by [8] and [25], where the estimated phase and frequency deviation states are used to compute a composite clock with improved stability. The composite clock is known as the implicit ensemble mean (IEM) and is not experimentally observable. The role of the signal generation system is to realize the IEM, and the development and analysis of this

system is the focus of this thesis. The last system in the testbed is the characterization system, which uses the state estimates from the Kalman Filter to calculate Overlapping Allan Deviations (OADEV) and characterize individual clock stability.

Following [55], the model for the sinusoidal analog signal produced by an oscillator is given as

$$V(t) = [V_0 + \varepsilon(t)]\, sin\, [2\pi f_n t + \phi(t)] \quad (V) \tag{1.1}$$

where V_0 is the nominal amplitude, $\varepsilon(t)$ is the amplitude fluctuation, f_n is the nominal frequency, and $\phi(t)$ is the phase fluctuation. The instantaneous frequency of a clock is defined as

$$f(t) = f_n + \frac{1}{2\pi}\frac{d}{dt}\phi(t) \quad (Hz) \tag{1.2}$$

and the normalized phase-time deviation is determined by

$$x(t) = \frac{\phi(t)}{2\pi f_n} \quad (s) \tag{1.3}$$

Given these definitions, the normalized clock frequency deviation (or fractional frequency error) is

$$y(t) = \frac{f(t) - f_n}{f_n} = \frac{\Delta f}{f_n} = \frac{d}{dt}x(t) \quad (s/s) \tag{1.4}$$

The behavior of clocks can be described by a deterministic component and a stochastic component. The Overlapping Allan Deviation represents the stochastic component of clock behavior, and is the most commonly used tool in metrology for analyzing oscillator stability [47]. OADEVs are calculated from the square root of the Overlapping Allan Variance (OAVAR). The OAVAR ($\boldsymbol{\sigma}_y^2(\tau)$) of a clock is computed from a set of time series measurements by

$$\boldsymbol{\sigma}_y^2(\tau) = \frac{1}{2(N-2m)\tau^2} \sum_{i=1}^{N-2m} [\boldsymbol{x}_{i+2m} - 2\boldsymbol{x}_{i+m} + \boldsymbol{x}_i]^2 \tag{1.5}$$

where $N = M + 1$ time measurements, M is the number of phase measurements, $\tau = m\tau_0$ is the averaging interval, m is the averaging interval factor, and τ_0 is the fundamental measurement period.

Figure 1.2 illustrates typical characteristics displayed in an OADEV plot. There are four common noise types seen in OADEV clock analyses, represented as different slopes when plotted.

White or flicker phase noise is indicated by a slope of -1. White frequency noise appears as a slope of $-1/2$. Flicker frequency noise is represented by a slope of 0, and random walk frequency noise is appears as a slope of $+1/2$.

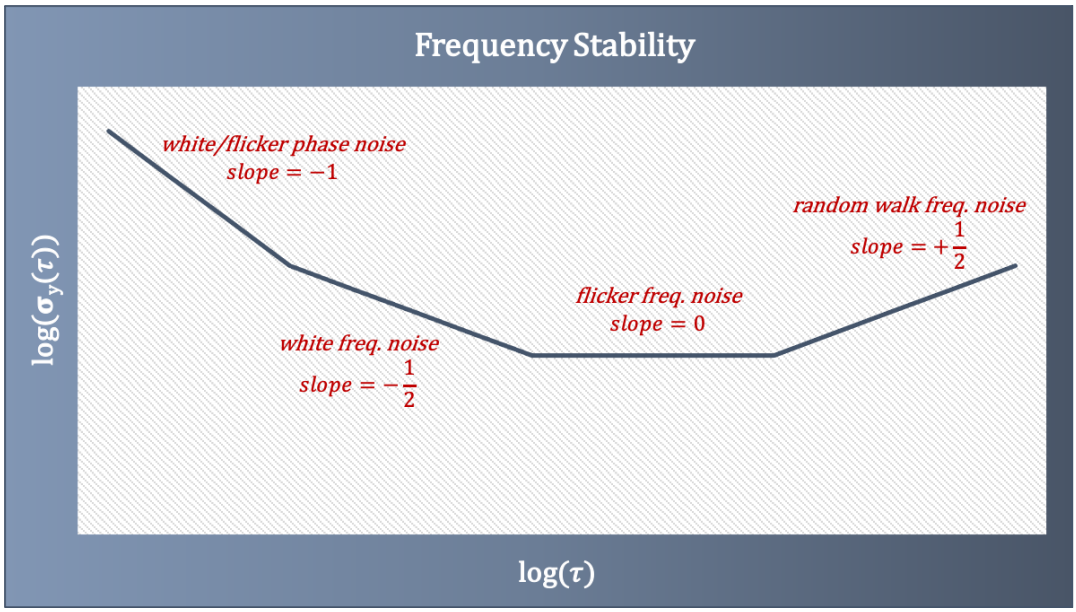

Figure 1.2: Example of different noise processes represented in an OADEV plot.

Reported Allan Deviation stability characteristics for several types of crystal oscillators, atomic frequency standards, and a CSAC are listed in Table 1.1.

Table 1.1: Characteristic oscillator stability ratings.

Oscillator Type	Stability at $\tau = 1\ s$	Stability at $\tau = 100\ s$	Stability at $\tau = 1000\ s$
MCXO [36]	$\leq 5 \times 10^{-11}$		
MV180 OCXO [41]	$< 2 \times 10^{-12}$		
HP-5065A Rb [44]	$\leq 5 \times 10^{-12}$	$< 5 \times 10^{-13}$	$< 5 \times 10^{-13}$
5071A Cs [35]	5×10^{-12}	8.5×10^{-13}	2.7×10^{-13}
MHM 2010 Maser [38]	1.5×10^{-13}	5×10^{-15}	2×10^{-15}
SA.45s CSAC [37]	3×10^{-10}	3×10^{-11}	1×10^{-11}

The rubidium, cesium, and active hydrogen maser are the atomic frequency standards. The two crystal oscillators in the table are a microprocessor controlled crystal oscillator (MCXO) and a double oven compensated crystal oscillator (OCXO). While crystal oscillators exhibit good short-term stability, this stability degrades over longer time intervals - where the atomic frequency standards exhibit far superior stability characteristics. The CSAC has stability characteristics slightly worse than the OCXO over short time intervals, but it performs better over time intervals $\tau > 100s$. An additional example OADEV plot is provided in Figure 1.3 to graphically show characteristic stability curves for some of these oscillators.

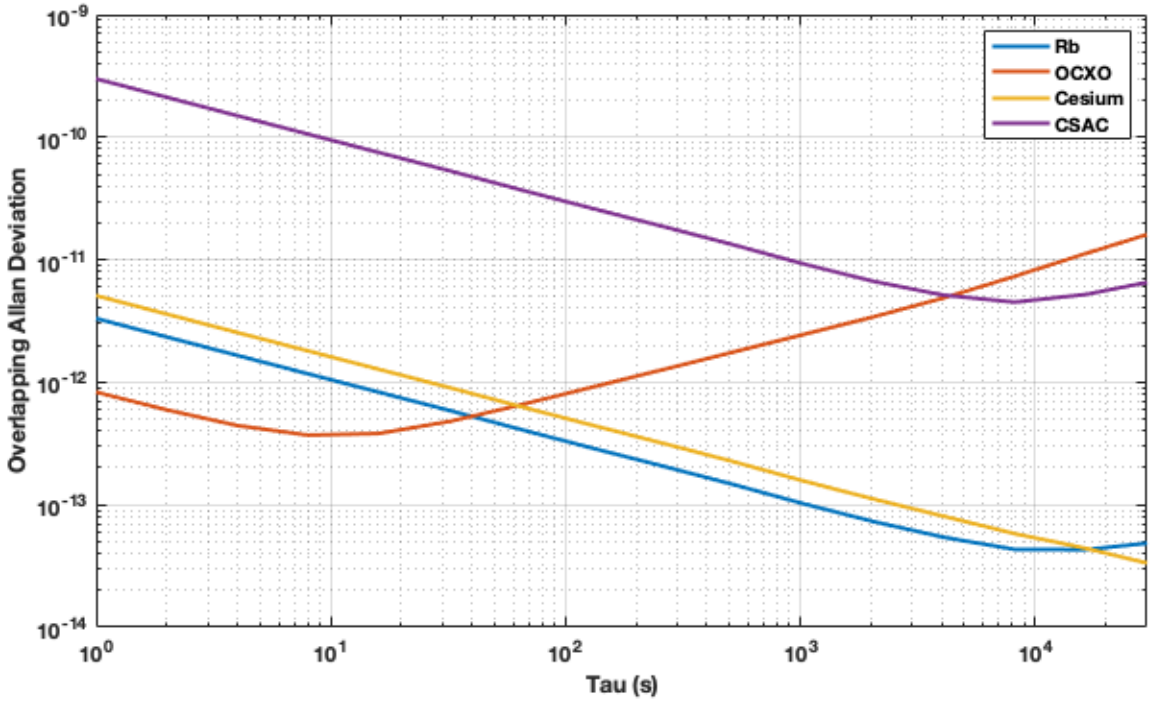

Figure 1.3: Characteristic frequency stability for various types of oscillators.

1.2 Motivation

My research focuses on the development of a signal generation system for the clock ensemble testbed, which meets overall system requirements and can interface with the testbed hardware platform. Beginning with system requirements, the signal generation system must be able to generate a steered output (IEM realization) based on a generation reference clock, and information received from the measurement and ensembling systems. A more detailed block diagram describing the functional operation of the testbed is displayed below in Figure 1.4.

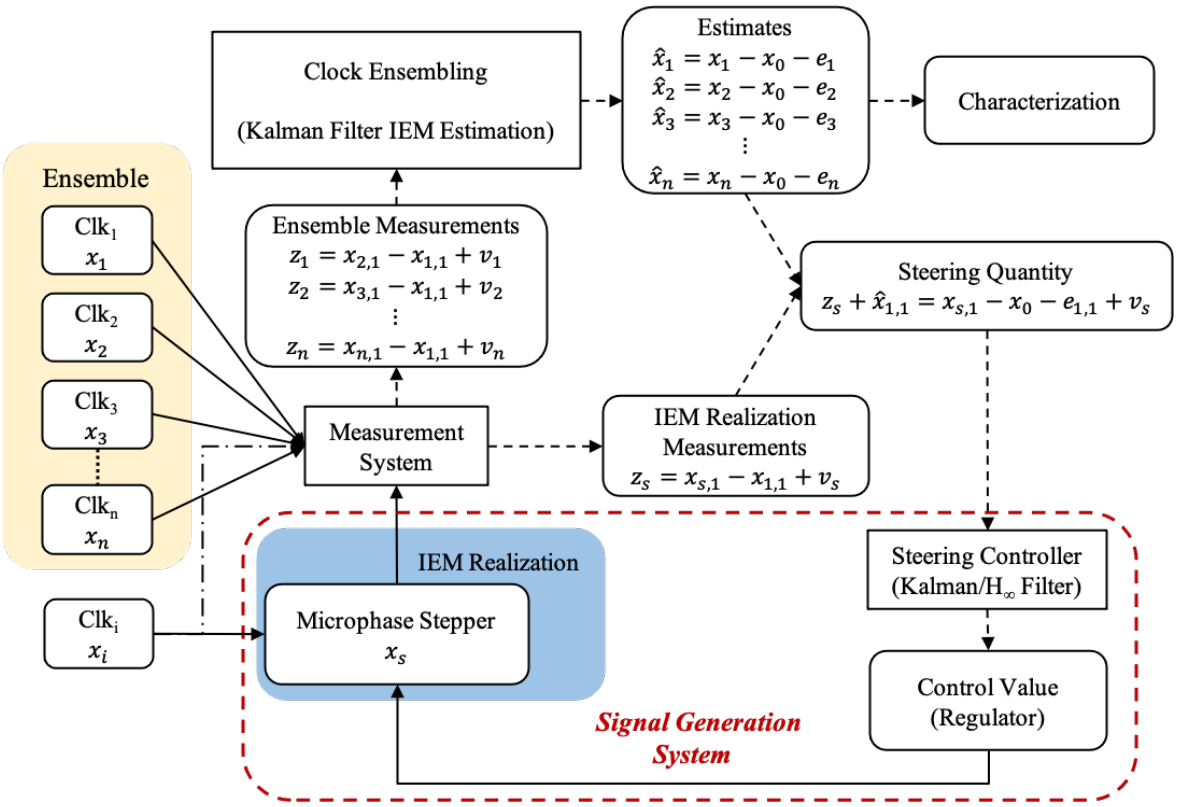

Figure 1.4: Detailed block diagram for the clock ensemble (adapted from [23]).

Solid lines indicate physical signals and dashed lines indicate numerical values. Phase measurements are denoted by z_j; v_j is the measurement noise; $\hat{\boldsymbol{x}}_j$ are the clock state estimates; $\boldsymbol{e}_j$ are the state estimate errors; and $\boldsymbol{x}_0$ is the IEM. The generation reference clock is Clk_i, and the dot-dash line represents that it can optionally be included in the ensemble. Measurements of the IEM realization

are only used by the steering controller, and the ensembling filter only uses measurements of the ensemble members.

Operations performed by the signal generation system are enclosed within the red dashed box in Figure 1.4. The estimator and regulator compute an optimal control value from the steering quantity calculated. This control value corresponds to the fractional frequency change of the microphase stepper (MPS) controlling the frequency of the steered output [29]. The IEM is realized by applying this control value to the microphase stepper. This divides the signal generation system up into two subsystems: the steering control subsystem, and the generation subsystem. The nominal design of the generation subsystem is described in Chapter 2, and the steering control subsystem design is discussed in Chapter 3.

Hardware interfacing requirements are discussed in Chapters 4 and 5 of this thesis, which focus on design considerations for compatibility with the current CONTACT hardware platform. The hardware device selected to host the clock ensemble is an Ettus N310 universal software radio platform (USRP). The N310 USRP (shown in Figure 1.5) was selected because it has a promising architecture for measuring up to four clocks simultaneously, using phase measurement techniques from [51].

Figure 1.5: Ettus N310 USRP [18]

This software defined radio (SDR) is a networked device. Signal processing occurs off-board on a host PC using a 1 gigabit network connection for data transfer and signal processing chains

written in GNU Radio using C++ or Python. In addition to four simultaneous inputs, the SDR is also capable of transmitting up to four signals, making it possible to take phase measurements and generate an output signal at the same time. However, there are a few aspects of the N310 architecture that must be investigated and analyzed for integration of the signal generation system.

The two primary architectural concerns for signal generation with the N310 USRP are the analog front-end signal processing chains on the daughterboards, and the complex ADC/DAC in the *AD9371* transceiver. The rated operational frequency range for the USRP is specified as 10 MHz to 6 GHz [18], but the rated frequency range for the transceiver is 300 MHz to 6 GHz [4]. The extended frequency range for the USRP is made possible by low band upconverters for the input signals, and low band downconverters for the output signals in the analog front-end processing chains. These are highlighted with red dashed boxes in Figure 1.6.

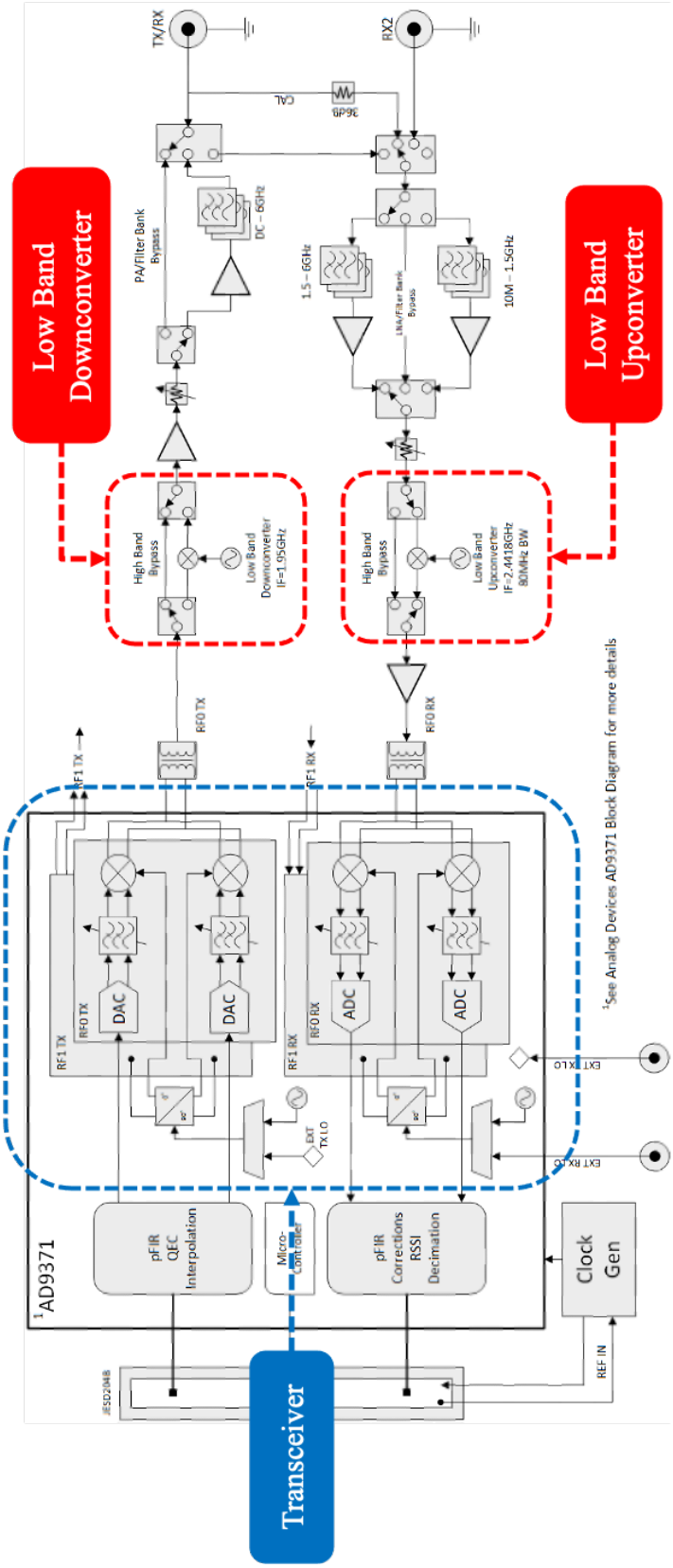

Figure 1.6: Front-end schematic for the N310 USRP (adapted from [18]).

The low band upconverter shifts up the frequency of input signals below 1.5 GHz to be closer to the center of the operational range of the transceiver (3.15 GHz) prior to downconversion and sampling by the complex ADC. Following the high frequency analog signal generation by the complex DAC and upconversion performed by the transceiver, the low band downconverter shifts down the frequency of the output signals being transmitted down. While the local oscillators (LOs) for the transceiver mixers can be precisely tuned with commands from GNU Radio and are synchronized with the N310 master clock, the LO's for the low band up/downconverters are not synchronized and cannot be tuned through GNU Radio. Tuning of the front-end LO's is not of concern, but the lack of synchronization is. This issue is addressed and analyzed in Chapter 4.

The secondary concern with the N310 USRP architecture is the operation of the transceiver. This type of transceiver is designed to downconvert the frequency of the input signal to an intermediate frequency (IF) or baseband. Conversely, the output signal from GNU Radio signal processing is mixed onto a carrier signal to upconvert the signal to the nominal frequency range. While downconversion to a lower frequency is advantageous for digital signal processing (DSP) by reducing stress on host PC resource usage, it enforces signal steering and generation operations to be performed at baseband or a near-baseband IF. Frequency steering has been demonstrated by [1] for high performance antenna beam forming, but the author has not found evidence of this approach being applied to the steering of timing signals. Typically, the synthesis or frequency correction of timing signals is performed at or above the nominal frequency of the timing signal - as shown in [12] and [44]. The nominal generation subsystem design, discussed in Chapter 2, incorporates this more typical steered signal generation approach. Chapter 4 describes necessary modifications to the nominal design for baseband steering, and Chapter 5 discusses the hardware implementation of both methods and compares experimental output signal performance.

1.3 Research Contributions

The research contributions discussed in this thesis are as follows:

(1) Development of a steered signal generation system design for a low-SWaP clock ensemble

   (a) Implementation of a nominal design in hardware - with low frequency signals

   (b) Analysis of experimental performance

(2) Analysis of a baseband steering concept, based on the software defined radio platform

   (a) Implementation of baseband steering system design in hardware - with low frequency signals

   (b) Analysis and comparison of experimental performance, with respect to the nominal steering approach

(3) Development and analysis of an H_∞ control approach for clock steering

1.4 Organization

The chapter by chapter organization of this thesis is as follows.

Chapter 2: The signal generation system is made up of two subsystems that work together. One subsystem produces the steered signal (generation subsystem), and the other computes the optimal frequency adjustment for the steering input (steering control subsystem). This chapter describes the design and operation of the generation subsystem.

Chapter 3: This chapter describes the synthesis and analysis of two H_∞ control systems that provide optimal steering inputs to the generation subsystem. The simulated stability of these controllers are compared with the performance of a recently published Linear Quadratic Gaussian (LQG) control approach for clock steering.

Chapter 4: This chapter discusses an anlysis of the N310 USRP front-end and presents modifications to the nominal steering system design for steering at baseband.

Chapter 5: Hardware implementation and testing are used to demonstrate design functionality of the real-time system. This chapter discusses the implementation of both steering methods with a digital controller and analysis of results from the hardware trials.

Chapter 6: The final chapter summarizes results and conclusions from the study conducted to investigate differences between the two signal steering approaches. Recommendations are also provided for future work contributing to the continued development of the CONTACT signal generation system.

Chapter 2

Signal Generation System Design: Generation Subsystem

2.1 Overview

The design for the CONTACT signal generation system was developed to meet the testbed requirements. The requirements for the signal generation system are to use measurement and estimate data to realize the IEM - by adjusting the frequency of a microphase stepper. A detailed block diagram of the signal generation system is provided below in Figure 2.1. Inputs to the signal generation system are the timing signal of the generation reference (Clk_i), the phase-time deviation estimate of the ensemble reference ($\hat{\boldsymbol{x}}_{1,1}$), and the feedback measurement of the steered output (z_s).

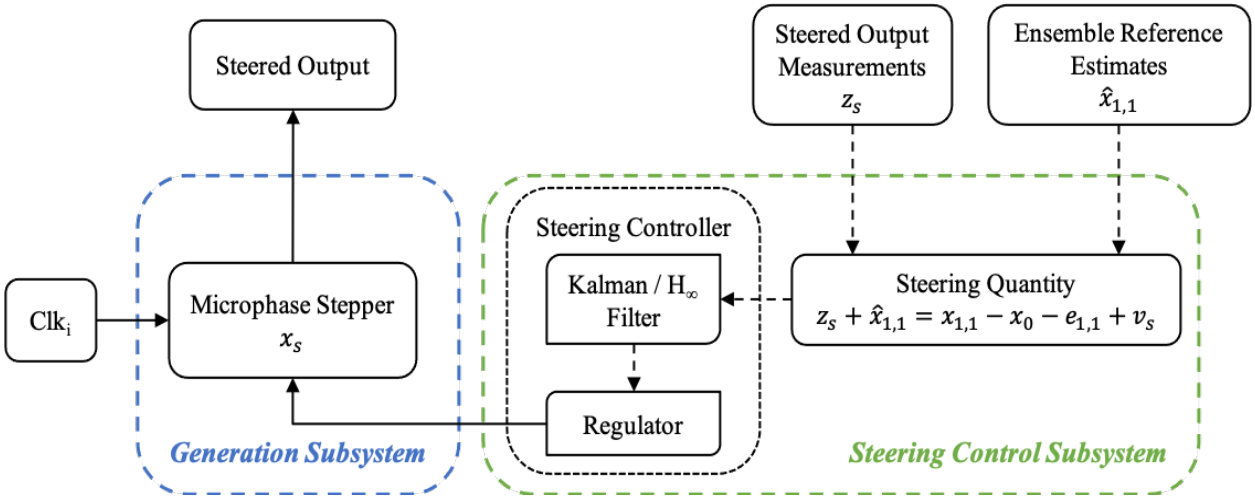

Figure 2.1: Block diagram for the signal generation system design. Solid lines indicate physical signals, and dashed lines indicate numerical values.

This system produces an analog timing signal that is steered to the IEM of the clock ensemble.

Operations to realize the IEM are performed by two subsystems: the steering control subsystem and the steered signal generation subsystem. This chapter focuses on the design of the generation subsystem. A detailed functional block diagram for the generation subsystem design is provided in Figure 2.2.

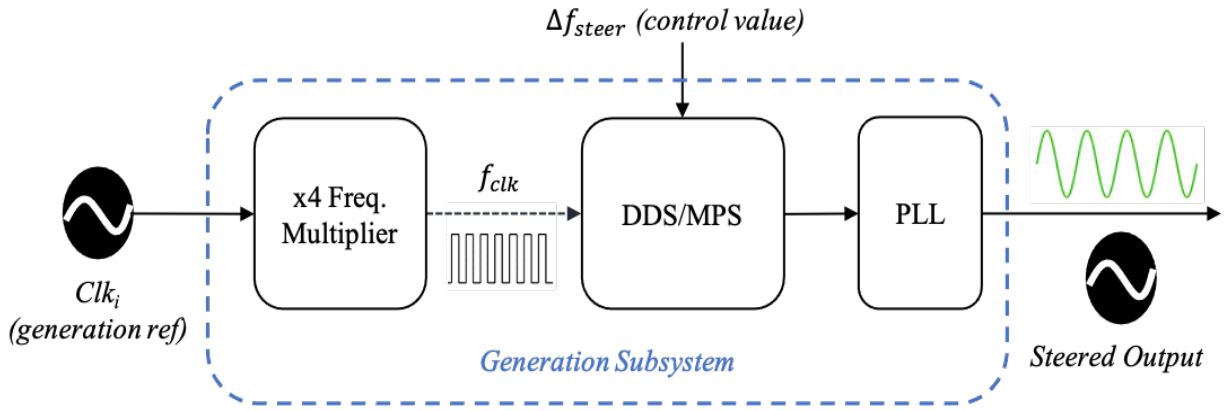

Figure 2.2: Functional block diagram for the generation subsystem design.

The subsystem design for producing a steered output signal follows a synthesis method applied in some atomic frequency standards [12] [44]. A stable frequency source is used to clock a frequency synthesis routine, and the timing signal output is produced by driving an analog smoothing operation with a digitally synthesized signal. The subsystem design incorporates three main components to perform these tasks. Shown in Figure 2.1, these components are: a frequency multiplier, MPS/direct digital frequency synthesizer (DDS), and a phase-locked loop (PLL). The stable reference frequency is supplied by the generation reference Clk_i, which serves as the basis for steering. Since the steering adjustments to the nominal frequency (f_n) of the generation reference will be very small, the frequency multiplier component multiplies the frequency of the reference signal up by a factor of four to meet and slightly exceed Nyquist requirements for digital synthesis. The frequency multiplied reference signal provides the digital frequency synthesis routine with a timing source. Steering corrections are applied to the microphase stepper/DDS component, which synthesizes a low amplitude resolution, frequency corrected, sine wave. This low amplitude resolution output is smoothed to produce a steered, low phase noise, timing signal. The smoothing

operation is performed by the last component in the subsystem - a PLL. The following sections in this chapter describe how each subsystem component operates to generate a steered output signal.

2.2 Frequency Multiplication

There are different techniques for frequency multiplication that can be employed, depending on the application requirements. Frequency multiplication is commonly performed by using Fractional/Integer-N PLL's [7]. However, by taking advantage of high-speed sampling and DSP capabilities of the CONTACT platform, the frequency multiplier component of the subsystem uses a simpler approach, with similar phase noise performance (see Figure 2.7). This technique consists of squaring and high pass filtering the steering reference input signal to pick off the second harmonic in the frequency spectrum, then using an "either direction" zero-crossing detector to produce a four-times frequency multiplied digital signal. A block diagram for this frequency multiplication method is provided in Figure 2.3.

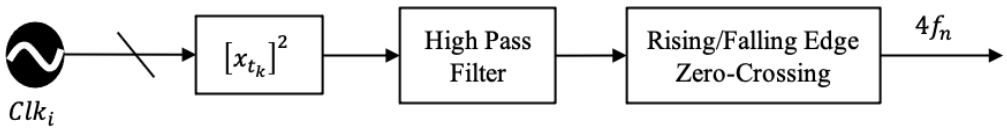

Figure 2.3: Block diagram for the DSP frequency multiplication method.

Given a sinusoidal generation reference signal, $s(t_k) = sin(2\pi f_n t_k)$, squaring this signal produces a signal with high power in the second harmonic of f_n. This can be shown by computing the discrete Fourier transform (DFT) of the squared reference signal and plotting spectral power. The DFT is given by [30]

$$F_n\{s(t_k)^2\} = \hat{s}_n^2 = \sum_{k=0}^{N-1} s_k^2 e^{\left[\frac{-2\pi ink}{N}\right]} \quad n = 0, 1, ..., N-1 \tag{2.1}$$

and power spectral density of the squared reference signal is displayed in Figure 2.4.

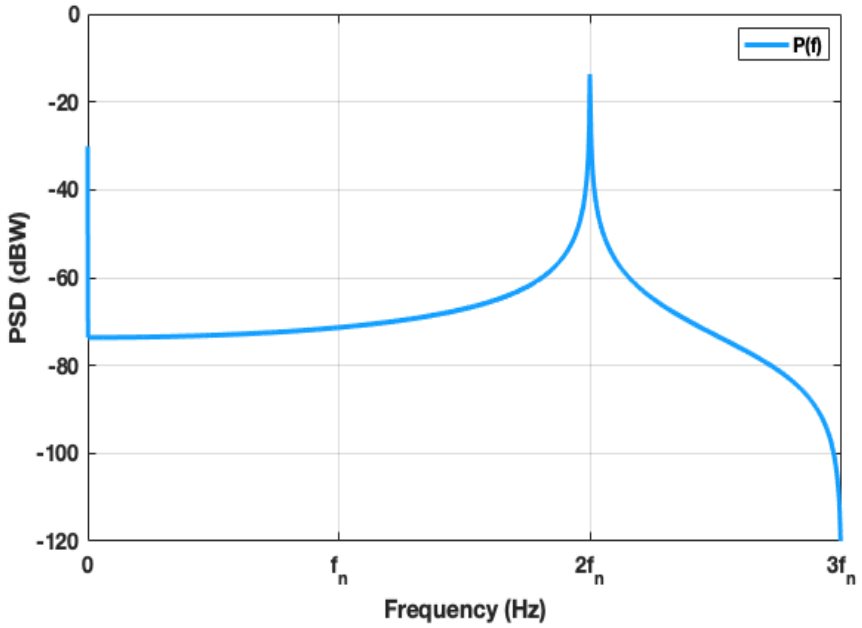

Figure 2.4: Spectral power in the squared generation reference input signal.

Spectral analysis shows that squaring the input produces a signal with power focused at twice the nominal frequency of the signal. High-pass filtering attenuates the DC noise component and renders a frequency doubled sine wave, centered at zero Volts. This frequency doubled signal is input to the rising/falling edge zero-crossing detector, which generates the final four-times frequency multiplied digital clocking signal for frequency synthesis. This is shown in Figure 2.5.

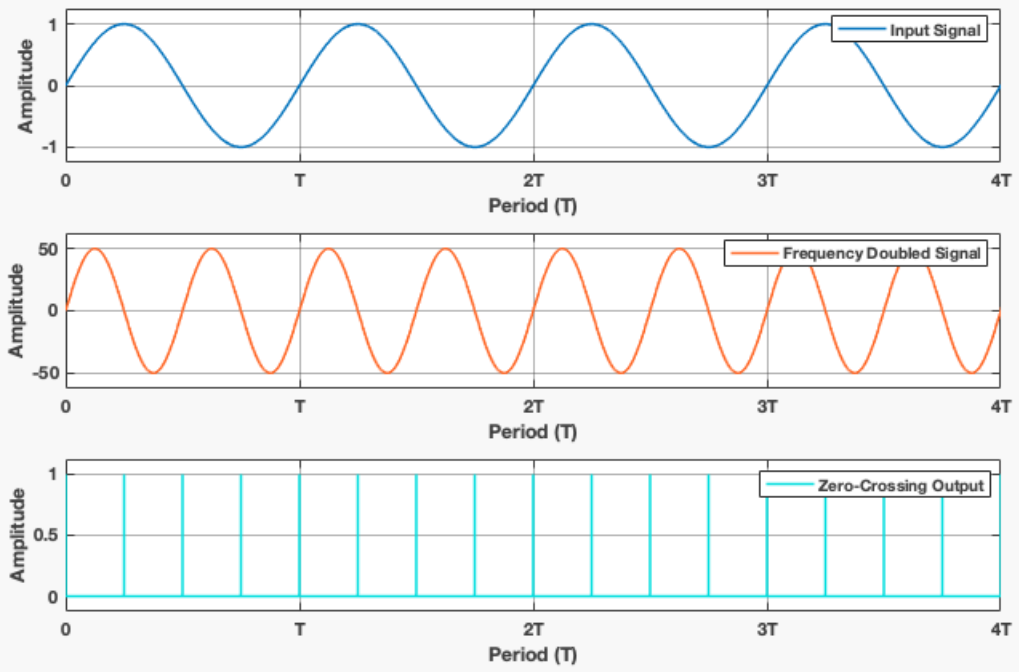

Figure 2.5: Example of signal flow in the frequency multiplier component. Top: input signal. Middle: frequency doubled signal (after high pass filtering). Bottom: x4 frequency multiplied clocking signal.

The frequency multiplier design is interchangeable with the Integer-N PLL method, and if desired, can replace the digital frequency multiplier without affecting the rest of the nominal generation subsystem design. If the Integer-N method is used, the analog timing signal from the generation reference is input to the feedback system shown in Figure 2.6, before being sampled by the testbed hardware system.

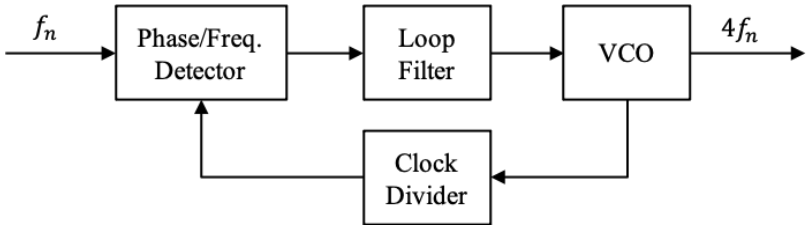

Figure 2.6: Integer-N PLL frequency multiplier.

The Integer-N PLL frequency multiplier is made up of a phase/frequency detector, loop filter, voltage controlled oscillator, and clock divider. These PLL components, except for the clock divider,

are explained in detail in Section 2.4 of this chapter. Refer to [3] and [7] for further information on both Fractional-N and Integer-N PLLs. To demonstrate the similarities in short-term phase noise between this approach and the Integer-N PLL approach, spectral analysis comparisons are provided below. Figure 2.7 displays a side-by-side comparison of power spectral density for simulated outputs from both frequency multiplier methods.

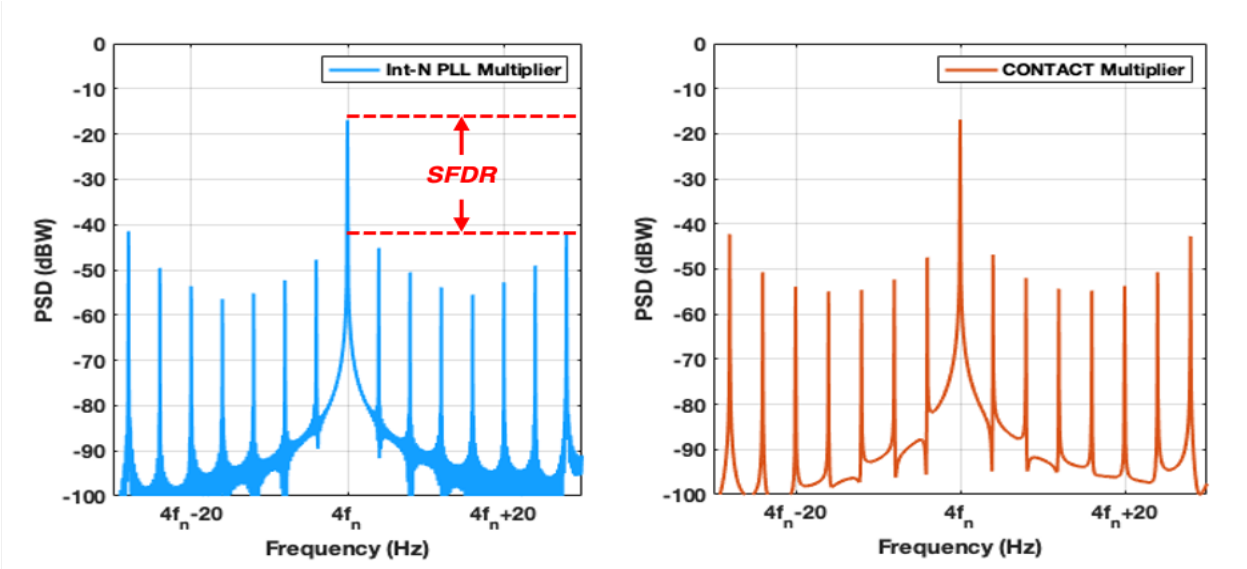

Figure 2.7: PSD comparison of frequency multiplier output signals. Left: Integer-N PLL method. Right: CONTACT frequency multiplication method.

Figure 2.7 shows the phase noise present in the four-times frequency multiplied clocking signals is approximately the same for both methods. The spurious-free dynamic range (SFDR) of the frequency multiplied output is also almost identical. The SFDR for the Integer-N approach is $\approx 25\,dBW$, and the SFDR for the proposed approach for the CONTACT system is $\approx 26\,dBW$.

2.3 Frequency Synthesis

The frequency synthesis routine is where steering is applied, and a low resolution version of the frequency corrected signal is digitally generated by the microphase stepper - implemented as a direct digital frequency synthesizer. A functional block diagram for the DDS is provided in Figure 2.8. This component operates by accumulating phase increments, at each clocking event from the

frequency multiplied reference, and the accumulated phase is converted to a sinusoidal amplitude [52]. Phase increment size is determined by the frequency tuning word (M), which is limited by the application dependent memory width of the N-bit phase accumulator (PA). The output of the PA is converted to a sinusoidal amplitude using a lookup table (LUT), where accumulated phase values are allowed to overflow and wrap around [33]. In between clock cycles, the last output signal value from the LUT is held. This is what gives the DDS output signal a stair-step like shape.

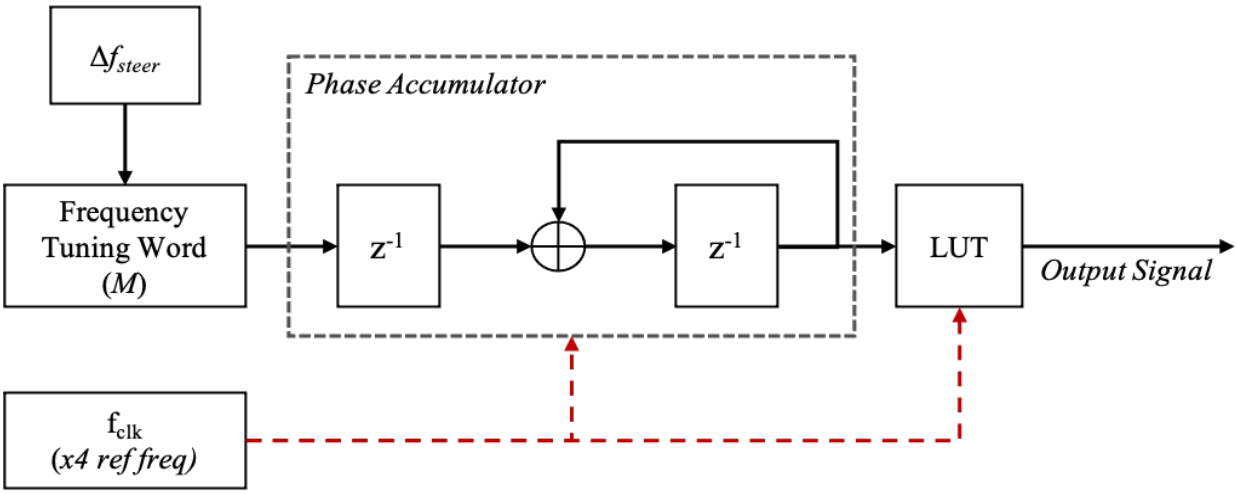

Figure 2.8: Block diagram for the direct digital frequency synthesis routine.

Steering commands change the phase increment size of M, such that the DDS synthesizes the corrected frequency. Control values for steering are received from the regulator in the steering control subsystem (shown in Figure 2.1). In the absence of steering adjustments, M is sized to produce a signal at the nominal frequency of the generation reference signal. The frequency tuning word is calculated as

$$M = \frac{(f_n + \Delta f_{steer})2^{N_{bits}}}{f_{clk}} = \frac{(f_n + \Delta f_{steer})2^{N_{bits}}}{4f_n} \tag{2.2}$$

and frequency resolution of the output signal from the DDS is defined by

$$f_{res} = \frac{4f_n}{2^{N_{bits}}}\ Hz \tag{2.3}$$

An example of the output signal produced by the DDS is shown in Figure 2.9. Since the LUT

employs an approximation technique for phase-amplitude conversion, there are inherent periodic errors in this process due to phase truncation [52]. The periodicity of the errors can result in undesired high-power, close-in, sideband noise (or "spurs"). If these spurs become problematic for frequency and phase tracking in the PLL component, phase dithering can be incorporated. Dithering breaks up the periodicity of the phase truncation errors - reducing power in the spurs and improving the SFDR of the output signal.

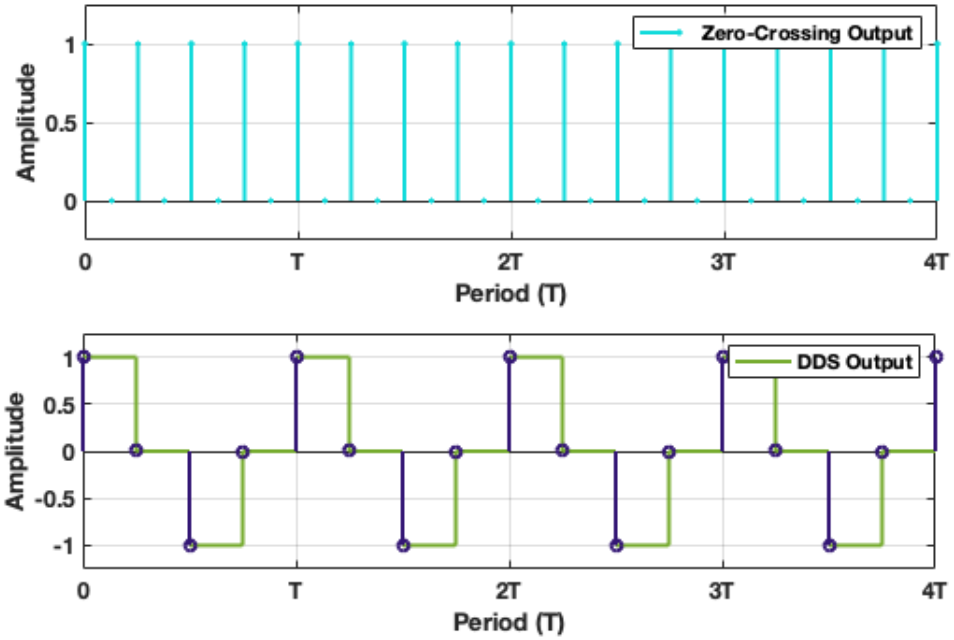

Figure 2.9: Example output signal from the digital frequency synthesis routine. Top: x4 frequency multiplied output from the rising/falling edge zero-crossing detector. Bottom: DDS output signal.

2.4 Steered Output Smoothing

Given the low amplitude resolution of the DDS output, the spectrum of the synthesized signal contains undesired power at harmonics of the output frequency that need to be suppressed by smoothing the frequency corrected signal. This smoothing operation is accomplished by the final component in the generation subsystem, a phase-locked loop driven by the DDS output signal. A functional block diagram for the smoothing PLL is displayed in Figure 2.10. A PLL is a closed-loop

feedback system that produces an output signal locked to the phase of the input signal. The closed loop system consists of three main sub-components, which are: a phase detector, loop filter, and voltage controlled oscillator (VCO).

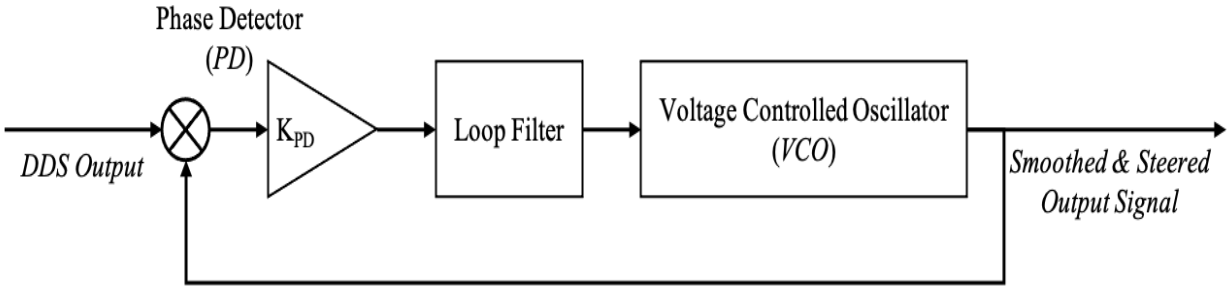

Figure 2.10: Block diagram for the generation subsystem smoothing PLL.

The phase detector in this design comprises a mixer and amplifier, which multiplies the DDS output signal with the fed back output signal from the VCO. The PD output value (v_{pd}) oscillates between $\pm K_{pd}$ and is given by

$$v_{pd} = K_{pd}(sin(\phi_{in})sin(\phi_{out})) = K_{pd}(cos((\omega_{in} - \omega_{out})t + \phi) - cos((\omega_{in} + \omega_{out})t + \phi)) \tag{2.4}$$

The loop filter is a low pass filter which removes the high frequency term from the PD output. If the PLL is closely tracking the input signal, the loop filter output is proportional to $cos((\omega_{in} - \omega_{out})t + \phi)$ [17]. This generates a control voltage that drives the VCO output to be 90^o phase-shifted from the input signal, when the angular frequencies are equal ($cos((0)t + 90) = 0$). Once phase-lock is achieved, the control voltage produced by the loop filter goes to a constant value [3]. The loop filter in this design is a first order active low pass filter, defined by the transfer function

$$LF(s) = \frac{K_{lf}}{s + \omega_c} \tag{2.5}$$

where ω_c is the filter cutoff frequency.

The VCO operates at a free-running frequency, and incorporates an integrator and gain term (K_{vco}). This control voltage drives the frequency of the VCO to increase or decrease such that the output signal locks to the phase and frequency of the input signal. The gain terms and loop filter

cutoff frequency all contribute to locking speed and capture range characteristics of the PLL. The frequency capture range is given by [11]

$$\omega_{cr} = \omega_{fr} \pm \frac{K_{pd}K_{lf}K_{vco}}{\sqrt{1+\left(\frac{\omega_{in}-\omega_{out}}{\omega_c}\right)^2}} \tag{2.6}$$

where ω_{fr} is the free-running angular frequency of the VCO. The open loop transfer function for the system is defined as

$$L(s) = \frac{K_{pd}K_{lf}\omega_c K_{vco}}{s^2+\omega_c s} \tag{2.7}$$

which drives the closed loop system transfer function to be

$$H(s) = \frac{L(s)}{1+L(s)} = \frac{K_{pd}K_{lf}\omega_c K_{vco}s^2 + K_{pd}K_{lf}K_{vco}\omega_c^2 s}{s^4+2\omega_c s^3+(K_{pd}K_{lf}K_{vco}\omega_c+\omega_c^2)s^2+K_{pd}K_{lf}K_{vco}\omega_c^2 s} \tag{2.8}$$

The output from the VCO is the final steered and smoothed output signal from the generation subsystem. An example of the smoothed output produced by driving the PLL with the digitally synthesized signal from the DDS is displayed in Figure 2.11.

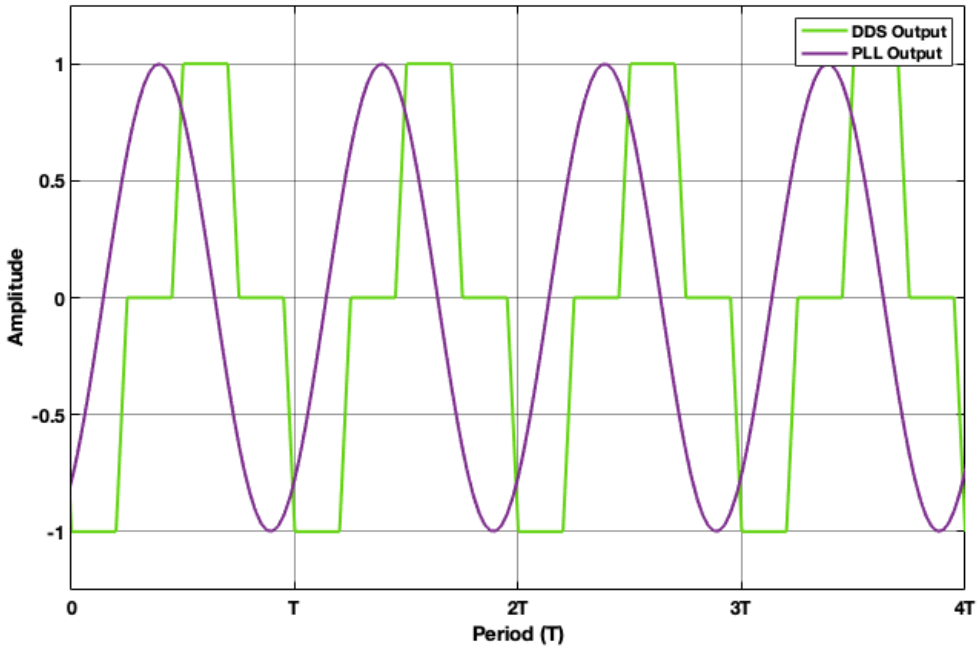

Figure 2.11: Example of a smoothed output from the PLL, when phase-lock has been achieved.

In Figure 2.11, we can see that the smoothed output from the PLL is phase-locked 90 degrees out of phase with the DDS output signal. This is a characteristic of the type of phase detector used in this PLL design, but is not common to all PLL's.

This concludes the discussion of the generation subsystem design. The following chapter describes the steering control subsystem.

Chapter 3

Signal Generation System Design: Steering Control

3.1 Overview

The other half of the signal generation system is the steering control subsystem. It is responsible for determining the steering adjustments to be applied to the microphase stepper (control values) in the generation subsystem, as described in the previous chapter. A general design approach for such a control system, is to first simplify the system by decoupling it from the ensemble estimator, then to analyze performance by steering one clock to another [24]. Furthermore, by leveraging numerical simulation, controllers can be rapidly tuned and tested to assess and improve performance of design candidates prior to hardware implementation. This chapter describes the development of the steering control subsystem - using this type of design and analysis approach.

Control techniques reported in literature were leveraged for the development and performance analysis of steering control subsystem. Linear Quadradic Gaussian (LQG) control has been demonstrated by [29], [39], [24], [57], [34], [27], and [48] as a suitable means for clock steering. I implemented the LQG control method as a nominal design for comparison and considered an H_∞ control approach. To compare steering performance between the use of LQG control and the H_∞ approach proposed in this chapter, the steering scenario from [48] is simulated for both control methods. OADEV analysis is used to assess the frequency stability of the steered clocks.

The steering scenario from [48] comprises a Morion MV180 OCXO that is to be steered to a Microsemi 5071A cesium atomic frequency standard. This is a highly relevant example because the cesium clock provides excellent long-term stability, while the OCXO provides superior short-term

stability that quickly degrades over averaging intervals $\tau \geq 10\,s$. Hence, the desired response of the controlled system preserves the short-term stability of the OCXO, but ties the long-term stability to the cesium reference.

The following section provides an overview of the steering control subsystem functions to produce a control value that will guide the generated output to the IEM. The parameters of the dynamical state space model used to simulate clock behavior are also described.

3.2 Steering Control Subsystem Structure & Clock Dynamics Model

Figure 3.1 displays a functional block diagram of the steering control subsystem. It is made up of a state estimator and regulator. The input to the subsystem is the steering quantity ($x_{r_{t_k}}$), which is the observed phase deviation of the steered output signal, summed with the estimated phase deviation of the ensemble reference. The output of this subsystem is the control value (u_{t_k}), which is the updated normalized frequency correction ($\Delta f_{steer_{t_k}} = {}^{\Delta f}/_{f_n}$) computed to drive the error between the steered output and IEM to zero.

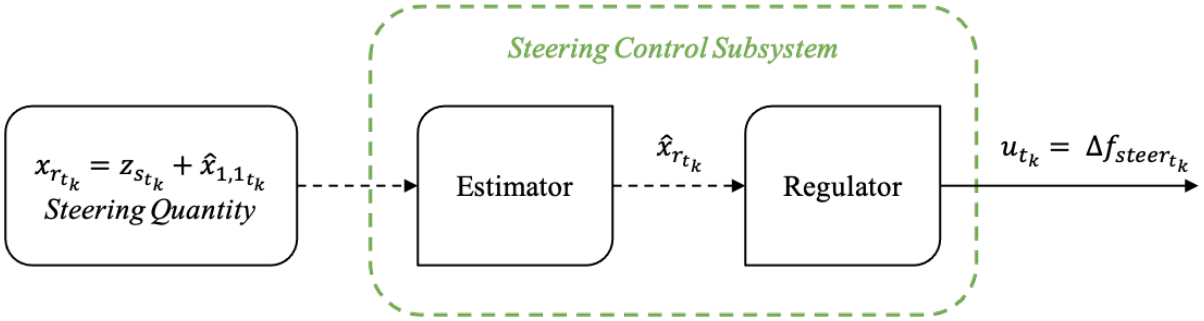

Figure 3.1: Functional block diagram for the steering control subsystem.

An estimator is needed for this control system to determine the phase error between the steered output ($\boldsymbol{x}_{s,1_{t_k}}$) and $x_{0_{t_k}}$ from $x_{r_{t_k}}$. Expanding the equation for $x_{r_{t_k}}$, we observe that the steering quantity is disturbed by the phase estimation error of the ensemble reference ($\boldsymbol{e}_{1,1_{t_k}}$) and measurement noise ($v_{s_{t_k}}$) [23]

$$\boldsymbol{x}_{r_{t_k}} = z_{s_{t_k}} + \hat{\boldsymbol{x}}_{1,1_{t_k}} = \boldsymbol{x}_{1,1_{t_k}} - x_{0_{t_k}} - \boldsymbol{e}_{1,1_{t_k}} + v_{s_{t_k}} \tag{3.1}$$

The filter uses knowledge of the uses knowledge of clock dynamics to estimate the difference between $\boldsymbol{x}_{s,1}$ and x_0. Then, the regulator multiplies the estimated steered clock error ($\hat{\boldsymbol{x}}_{r_{t_k}}$) with a static gain matrix to produce the control value for the microphase stepper. The dynamical model used to synthesize the optimal filter and regulator gains for the control system is the two-state clock model.

The two-state clock model is a dynamic state space model, developed by [20], and is widely used in the field of metrology to simulate natural behavior of clocks. While there is also a more accurate three-state model, the two-state model is preferred for steering because the addition of a drift state (third state) makes the system not fully controllable (demonstrated in [23]). Clock behavior is simulated through propagation of normalized phase-time deviation and normalized frequency deviation, from t_k to t_{k+1}. The state equation is defined as

$$\boldsymbol{x}_{t_{k+1}} = \boldsymbol{F}(\tau)\boldsymbol{x}_{t_k} + \boldsymbol{G}(\tau)u_{t_k} + \boldsymbol{w}_{t_k} \tag{3.2}$$

where the state transition matrix $\boldsymbol{F}(\tau)$ written as

$$\boldsymbol{F}(\tau) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

and the control input propagation matrix $\boldsymbol{G}(\tau)$ is

$$\boldsymbol{G}(\tau) = \begin{bmatrix} \tau \\ 1 \end{bmatrix} \tag{3.4}$$

Here, $\boldsymbol{x}$ is the state vector and τ represents the time step interval.

The normally distributed process noise $\boldsymbol{w}_{t_k}$ is defined as

$$\boldsymbol{w}_{t_k} \sim \mathcal{N}(0, \boldsymbol{Q}) \tag{3.5}$$

The process noise covariance matrix $\boldsymbol{Q}$ accounts for the characteristic stochastic component of clock behavior

$$\boldsymbol{Q} = \begin{bmatrix} q_1\tau + \frac{q_2\tau^3}{3} & \frac{q_2\tau^2}{2} \\ \frac{q_2\tau^2}{2} & q_2\tau \end{bmatrix} = \begin{bmatrix} \frac{1}{2}h_0\tau + 2h_{-1}\tau^2 + \frac{2}{3}\pi^2 h_{-2}\tau^3 & \pi^2 h_{-2}\tau^2 \\ \pi^2 h_{-2}\tau^2 & 2\pi^2 h_{-2}\tau \end{bmatrix} \tag{3.6}$$

where q_1 and q_2 are the diffusion coefficients, and the h's are the Allan Variance parameters [29].

We can verify that the system is controllable by constructing controllability matrix ($\boldsymbol{\Theta}$), and demonstrating that it has the same rank as number of states in the system or that the determinant ($det(\boldsymbol{\Theta})$) does not equal zero [53].

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{G}(\tau) & \boldsymbol{F}(\tau)\boldsymbol{G}(\tau) \end{bmatrix} = \begin{bmatrix} \tau & 2\tau \\ 1 & 1 \end{bmatrix}, \quad det\left(\boldsymbol{\Theta}\right) = -\tau \neq 0 \tag{3.7}$$

By inspection, we can see that $\boldsymbol{\Theta}$ has a rank of two, and the calculated determinant does not equal zero. This confirms that the system is fully controllable.

Lastly, since we measure phase difference, this yields the following equation for state observations y_{t_k} and observation sensitivity matrix $\boldsymbol{H}$

$$y_{t_k} = \boldsymbol{H}\boldsymbol{x}_{t_k} + v_{t_k} \tag{3.8}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{3.9}$$

where v_{t_k} represents the normally distributed additive measurement noise.

$$v_{t_k} \sim \mathcal{N}(0, \boldsymbol{R}) \tag{3.10}$$

3.3 LQG Steering Approach

LQG controllers are made up of a Kalman Filter and a Linear Quadratic Regulator (LQR) in a state feedback structure. This type of control system uses knowledge of system noise parameters and assumes that perturbations to the system plant are zero mean, Gaussian signals with known covariances [45]. The state estimate equation for this observer-based state feedback controller is [29]

$$\hat{\boldsymbol{x}}_{t_{k+1}} = \boldsymbol{F}\hat{\boldsymbol{x}}_{t_k} + \boldsymbol{G}u_{t_k} + \boldsymbol{K}_{t_{k+1}}\left[z_{t_{k+1}} - \boldsymbol{H}\left(\boldsymbol{F}\hat{\boldsymbol{x}}_{t_k} + \boldsymbol{G}u_{t_k}\right)\right] \tag{3.11}$$

where $\boldsymbol{K}_{t_k}$ is the Kalman gain. The optimal Kalman gain minimizes the mean-square estimation error [9]. Calculation of the updated gain and Kalman estimates, from each observation, follow the procedure from [9] and [53] (with the addition of an error covariance reduction step). The recursive filter follows the steps below:

Step 1: initialize values for the one step ahead predictions of the state estimates $\hat{\boldsymbol{x}}_0^-$ and error covariance $\boldsymbol{P}_0^-$.

Step 2: calculate the updated Kalman gain

$$\boldsymbol{K}_{t_k} = \boldsymbol{P}_{t_k}^-\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_{t_k}^-\boldsymbol{H}^T + \boldsymbol{R})^{-1}$$

Step 3: update state estimates from the most recent measurement ($\hat{\boldsymbol{x}}_{t_k}^+$ is the estimated state vector for the regulator)

$$\hat{\boldsymbol{x}}_{t_k}^+ = \hat{\boldsymbol{x}}_{t_k}^- + \boldsymbol{K}_{t_k}(z_{t_k} - \boldsymbol{H}\hat{\boldsymbol{x}}_{t_k}^-)$$

Step 4: compute updated error covariance

$$\boldsymbol{P}r_{t_k}^+ = (\boldsymbol{I} - \boldsymbol{K}_{t_k}\boldsymbol{H})\boldsymbol{P}r_{t_k}^-$$

Step 5: compute reduced error covariance

$$\boldsymbol{P}_{t_k}^+ = \frac{1}{2}(\boldsymbol{P}r_{t_k}^+ + \boldsymbol{P}r_{t_k}^{+T})$$

Step 6: calculate predictions for the next time step

$$\hat{\boldsymbol{x}}_{t_{k+1}}^- = \boldsymbol{F}(\Delta t)\hat{\boldsymbol{x}}_{t_k}^+ + \boldsymbol{G}(\Delta t)u_{t_k}$$

$$\boldsymbol{P}^{-}_{t_{k+1}} = \boldsymbol{F}(\Delta t)\boldsymbol{P}^{+}_{t_k}\boldsymbol{F}(\Delta t)^T + \boldsymbol{Q}$$

Step 7: return to Step 2 and repeat for each measurement.

The second part of the LQG controller structure is the Linear Quadratic Regulator. The LQR problem is solved by determining the optimal control value u_{t_k} which minimizes the following cost function [48] [29] [24]

$$J_R = E\left[\sum_k \left[\boldsymbol{x}^T_{t_k}\boldsymbol{W}_Q\boldsymbol{x}_{t_k} + \boldsymbol{u}^T_{t_k}\boldsymbol{W}_R\boldsymbol{u}_{t_k}\right]\right] \tag{3.12}$$

where $\boldsymbol{W}_Q = \boldsymbol{W}_Q^T \geq \boldsymbol{0}$ and $\boldsymbol{W}_R = \boldsymbol{W}_R^T \geq \boldsymbol{0}$ are the designer specified weighting parameters for tuning the controller. $\boldsymbol{W}_Q$ penalizes excursions from equilibrium and $\boldsymbol{W}_R$ penalizes actuator usage [29]. The optimal control which minimizes cost function J_R is [48] [29] [24] [57]

$$u_{t_k} = -\boldsymbol{K}_R\hat{\boldsymbol{x}}_{t_k} \tag{3.13}$$

where $\hat{\boldsymbol{x}}_{t_k}$ is the updated state estimate produced by the Kalman Filter. The optimal LQR gain ($\boldsymbol{K}_R$) is calculated as

$$\boldsymbol{K}_R = (\boldsymbol{G}^T\boldsymbol{X}\boldsymbol{G} + \boldsymbol{W}_R)^{-1}\boldsymbol{G}^T\boldsymbol{X}\boldsymbol{F} \tag{3.14}$$

where $\boldsymbol{X}$ is the unique, positive semi-definite solution to the following discrete-time algebraic Riccati equation [48] [29] [24]

$$\boldsymbol{X} = \boldsymbol{F}^T\boldsymbol{X}\boldsymbol{F} + \boldsymbol{W}_Q - \boldsymbol{F}^T\boldsymbol{X}\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{X}\boldsymbol{G} + \boldsymbol{W}_R)^{-1}\boldsymbol{G}^T\boldsymbol{X}\boldsymbol{F} \tag{3.15}$$

Applying the framework above, four LQG controllers were proposed by [48]. Table 3.1 provides the tuning parameters for each controller. $\boldsymbol{W}_Q$ was chosen to be the same for all of the LQG controllers, but each controller results from the variation of $\boldsymbol{W}_R$ and Δt. The control interval Δt defines how often steering is applied, which affects the state transition and control input matrices ($\boldsymbol{F}(\Delta t)$, $\boldsymbol{G}(\Delta t)$) for synthesis of the LQR and Kalman gains.

Table 3.1: LQG steering controller parameters.

Controller	Steady State Error ($\boldsymbol{W_Q}$)	Control Effort ($\boldsymbol{W_R}$)	Control Interval
LQG Controller 1	$\begin{bmatrix} 10^{-1} & 0 \\ 0 & 10^{-10} \end{bmatrix}$	1	$\Delta t = 5$ {s}
LQG Controller 2	$\begin{bmatrix} 10^{-1} & 0 \\ 0 & 10^{-10} \end{bmatrix}$	100	$\Delta t = 5$ {s}
LQG Controller 3	$\begin{bmatrix} 10^{-1} & 0 \\ 0 & 10^{-10} \end{bmatrix}$	1	$\Delta t = 20$ {s}
LQG Controller 4	$\begin{bmatrix} 10^{-1} & 0 \\ 0 & 10^{-10} \end{bmatrix}$	100	$\Delta t = 20$ {s}

The parameters for clock model process noise diffusion coefficients, and the measurement noise variance applied in my simulations are listed in Table 3.2. In addition, a block diagram for the LQG state feedback system is provided in Figure 3.2.

Table 3.2: Process and measurement noise parameters.

Parameter	Value
$q_{1,OCXO}$	$6.5x10^{-25}$
$q_{2,OCXO}$	$1.9x10^{-26}$
$q_{1,Cs}$	$3.6x10^{-23}$
$q_{2,Cs}$	0
R	1.0×10^{-24}

†Reported values are from [48].

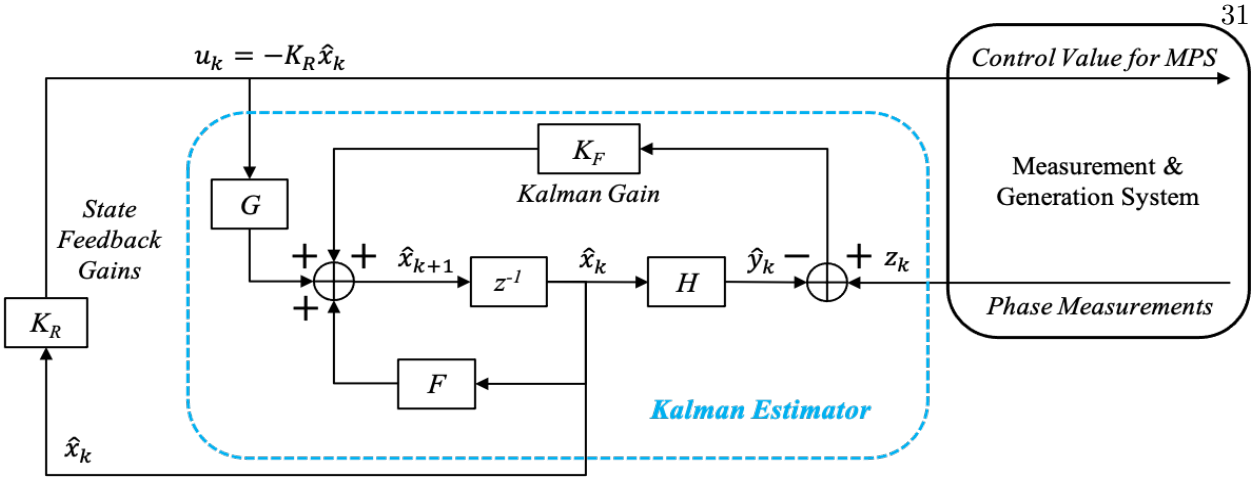

Figure 3.2: Block diagram for the LQG state feedback control system.

3.4 H_∞ Steering Approach

H_∞ control is a robust control strategy that can be advantageous for control of systems with a large number of states and/or uncertainties [54]. The control system structure also takes a state feedback form and is very similar to that of an LQG controller. Where H_∞ differs from LQG is in the optimal control strategy. LQG control seeks to minimize the mean-square reference tracking error covariance, but H_∞ imposes a limit on closed loop system energy gain from exogeneous inputs [45] and does not make the same assumptions about system disturbances. The peak closed loop system gain, over all frequencies, is given by the infinity norm of the system ($\|.\|_\infty$) [14], or equivalently, the maximum singular value of the system ($\bar{\boldsymbol{\sigma}}[.]$). This results in an H_∞ controller, $\boldsymbol{K}$, being one which limits the closed loop system gain to the minimum upper bound γ (H_∞ cost).

A common approach for general H_∞ control optimization is to use γ-iteration (described by [15]) to synthesize a controller which achieves the optimal value for γ (γ_{min}) - within a designer specified tolerance. This is because the solution to the general H_∞ optimization problem is not unique. However, I applied an alternative optimization technique for the H_∞ steering controllers, which does not require γ-iteration and has an exact solution. The approach employs the method developed by [22], using H_∞ loop-shaping and robust stabilization. This solution procedure is used synthesize an H_∞ controller ($\boldsymbol{K}$) that provides desired reference tracking and maximizes the

stability margin of the closed loop system, given a broad description of plant uncertainties.

In this section, two H_∞ controllers are synthesized to steer the OCXO. The design process begins by shaping the singular values of the plant and open loop system to achieve desired closed loop performance. This is conducted by applying designer specified pre- and post-compensating weighting functions ($\boldsymbol{W}_1$ and $\boldsymbol{W}_2$) to the system [22]. The shaped plant and controller system is shown in Figure 3.3; where $\boldsymbol{K} = \boldsymbol{W}_1\boldsymbol{K}_s\boldsymbol{W}_2$ is the feedback controller for the shaped plant $\boldsymbol{G}_s = \boldsymbol{W}_2\boldsymbol{G}\boldsymbol{W}_1$.

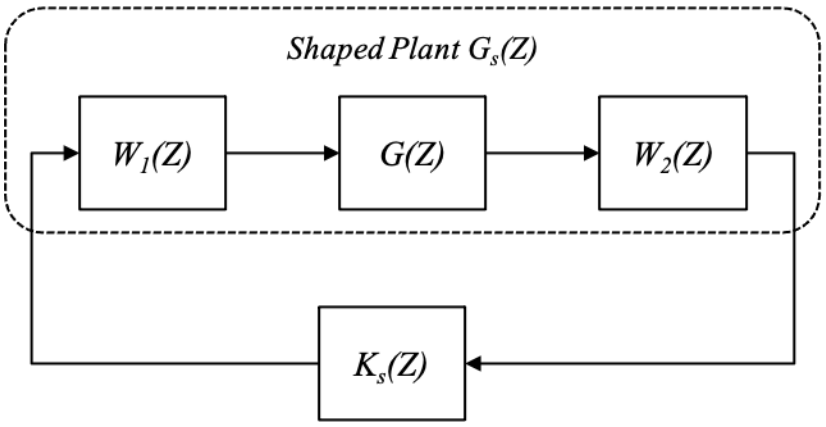

Figure 3.3: Block diagram for the shaped plant/controller. Here, $\boldsymbol{G}(Z)$ is the discrete-time transfer function for the plant and $\boldsymbol{K}_s(Z)$ is the controller synthesized to robustly stabilize the shaped plant $\boldsymbol{G}_s(Z)$ (adapted from [54]).

Since the controller synthesized from the solving the H_∞ robust stabilization problem does not have a large effect on the open loop singular values, controller performance is primarily impacted by the plant shaping process. The weighting of $\boldsymbol{W}_1$ affects tracking performance, disturbance rejection and the bandwidth of the closed loop system. In general, high gain at low frequencies corresponds to good reference tracking, and low gain at higher frequencies corresponds to good disturbance rejection. Weighting of $\boldsymbol{W}_2$ is often set to $\boldsymbol{W}_2 = \boldsymbol{I}$, and affects the importance of controlled outputs and feedback measurements [54].

Table 3.3 displays the weighting parameters for my single-input single-output (SISO) controller designs. Since the plant is a simple integrator and we do not want to degrade short-term performance by steering too quickly, simple weights that scale the system gain are selected. I chose

values for W_1, W_2, and the control intervals that result in different system bandwidths and reductions in overall system gain. By doing so, favorable and unfavorable characteristics in singular value shapes for steering performance are distinguishable in OADEV analyses.

Table 3.3: H_∞ loop-shaping parameters and control intervals.

Controller	$\boldsymbol{W_1}$	$\boldsymbol{W_2}$	Control Interval
H_∞ Controller 1	$1/700$	1	$\Delta t = 1$ {s}
H_∞ Controller 2	$1/95$	1	$\Delta t = 5$ {s}

The singular values of the shaped and unshaped plants are compared in Figures 3.4 and 3.5, where the discrete-time transfer function for the plant is computed by

$$\boldsymbol{G}(Z) = \boldsymbol{H}\left(z\boldsymbol{I} - \boldsymbol{F}(\Delta t)\right)^{-1}\boldsymbol{G}(\Delta t) \tag{3.16}$$

and the shaped discrete-time plant is

$$\boldsymbol{G}_s(Z) = \boldsymbol{W}_2(Z)\boldsymbol{G}(Z)\boldsymbol{W}_1(Z) \tag{3.17}$$

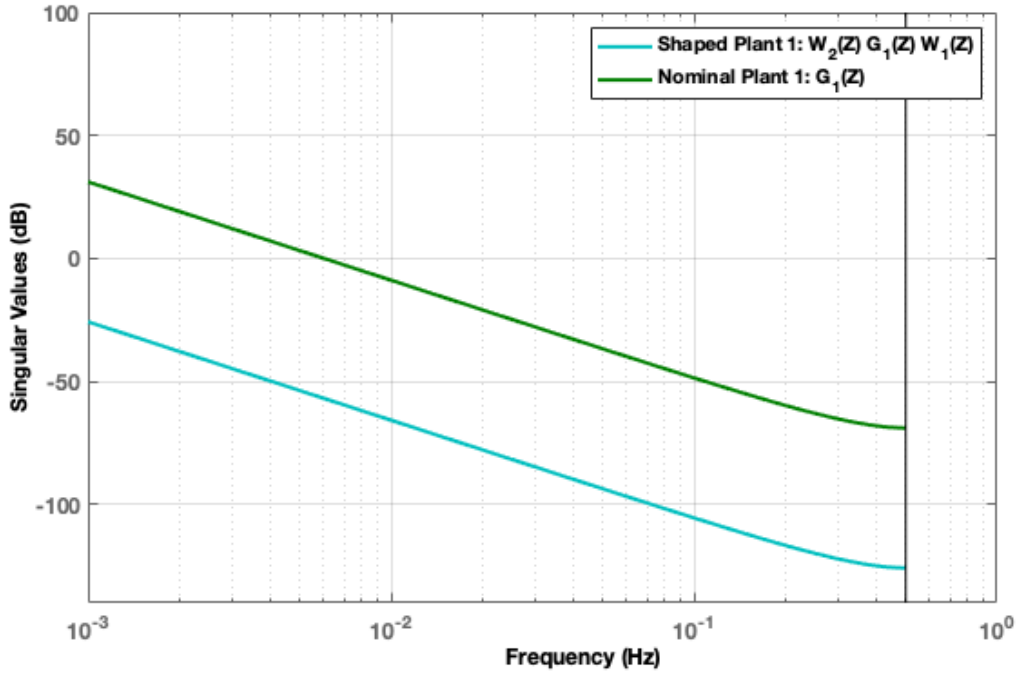

Figure 3.4: Singular values of the shaped and unshaped plants for H_∞ controller 1.

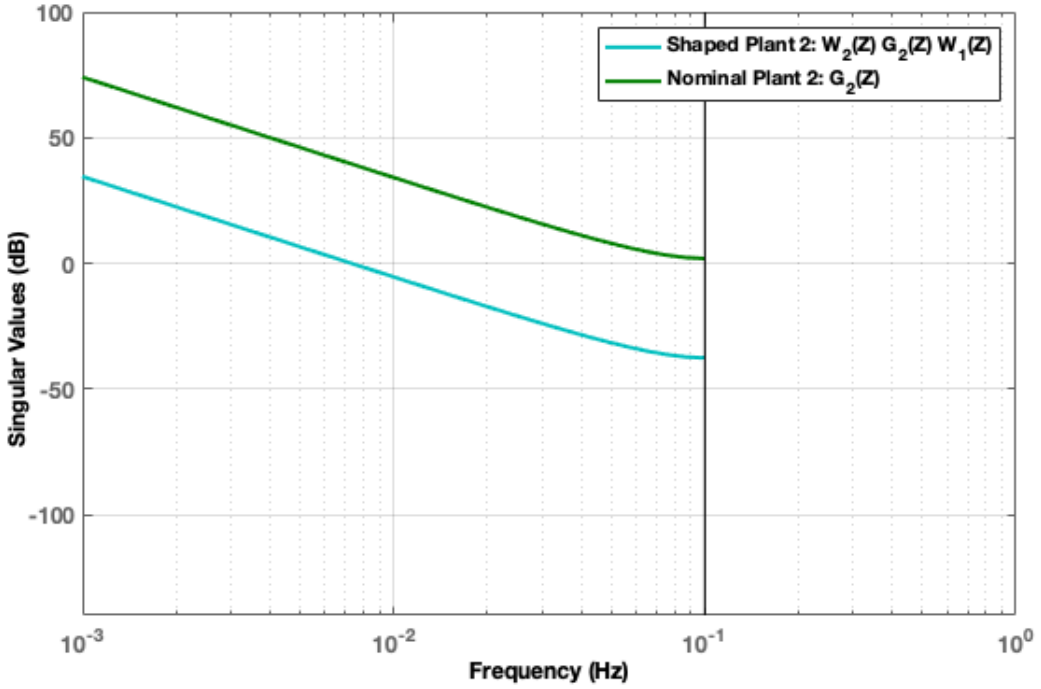

Figure 3.5: Singular values of the shaped and unshaped plants for H_∞ controller 2.

The singular values of both shaped and unshaped plants show that the larger control interval reduces system bandwidth by a factor of five. Based on magnitudes for the shaped plants, we should expect H_∞ controller 1 to provide better noise rejection and H_∞ controller 2 to provide better reference tracking.

The H_∞ optimization problem to synthesize a controller that ensures robust stability for the closed loop system, follows the solution derived by [22] using coprime uncertainty models. The premise for the solution is that the shaped plants have a left coprime factorization written as

$$\boldsymbol{G}_s = \boldsymbol{M}^{-1}\boldsymbol{N} \tag{3.18}$$

The solution objective is to select a controller which stabilizes a family of perturbed plants $\boldsymbol{G}_p$ with coprime uncertainty defined by [54] [22] [49]

$$\boldsymbol{G}_p = \{(\boldsymbol{M} + \boldsymbol{\Delta}_M)^{-1}(\boldsymbol{N} + \boldsymbol{\Delta}_N), \quad \|[\boldsymbol{\Delta}_N \;\; \boldsymbol{\Delta}_M]\|_\infty \leq \epsilon\} \tag{3.19}$$

where $[\boldsymbol{\Delta}_N \;\; \boldsymbol{\Delta}_M]$ are unknown transfer functions representing the maximum amount of uncertainty and ϵ is the stability margin. The block diagram for the perturbed feedback system with coprime uncertainty is displayed in Figure 3.6.

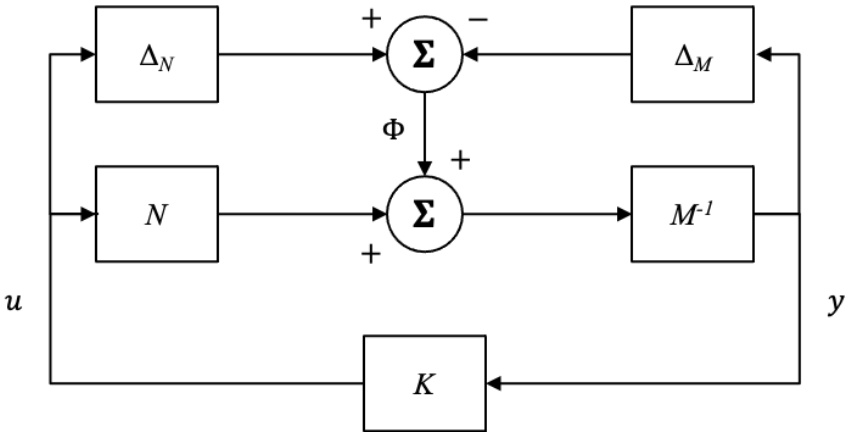

Figure 3.6: Perturbed feedback system block diagram for the H_∞ robust stabilization problem (adapted from [54] and [22]), where $\boldsymbol{\Phi}$ is the perturbed input.

Using this coprime uncertainty description is advantageous because it provides a very general

definition of model uncertainty and does not require knowledge of specific system uncertainty parameters. If the nominal feedback system is stable, then this perturbed feedback system is robustly stable if and only if

$$\gamma \triangleq \left\| \begin{bmatrix} \boldsymbol{K} \\ \boldsymbol{I} \end{bmatrix} (\boldsymbol{I} - \boldsymbol{G}_s \boldsymbol{K})^{-1} \boldsymbol{M}^{-1} \right\|_\infty = \gamma_{min} \leq \frac{1}{\epsilon} \tag{3.20}$$

where γ is the H_∞ norm from $\boldsymbol{\Phi}$ to $[u \quad y]^T$ [54] and γ_{min} is the optimal H_∞ cost. The lowest possible value that can be achieved for γ_{min} is defined as [22]

$$\gamma_{min} = \frac{1}{\epsilon_{max}} = \left\{ 1 - \|[\boldsymbol{N} \quad \boldsymbol{M}]\|_H^2 \right\}^{-1/2} = [1 + \rho(\boldsymbol{X}\boldsymbol{Z})]^{\frac{1}{2}} \tag{3.21}$$

where ρ represents the maximum eigenvalue, $\|.\|_H$ is the Hankel norm, and $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the unique positive definite solutions to the discrete-time matrix Riccati equations 3.22 and 3.23 [5].

$$\boldsymbol{E}^T \boldsymbol{X} \boldsymbol{E} = \bar{\boldsymbol{F}}_x^T \boldsymbol{X} \bar{\boldsymbol{F}}_x - \bar{\boldsymbol{F}}_x \boldsymbol{X} \boldsymbol{G}_x (\boldsymbol{G}_x^T \boldsymbol{X} \boldsymbol{G}_x + \boldsymbol{R})^{-1} \boldsymbol{G}_x^T \boldsymbol{X} \bar{\boldsymbol{F}}_x^T + \boldsymbol{H}_x^T \boldsymbol{Q}_x \boldsymbol{H}_x - \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T \tag{3.22}$$

$$\boldsymbol{E}^T \boldsymbol{Z} \boldsymbol{E} = \bar{\boldsymbol{F}}_z^T \boldsymbol{Z} \bar{\boldsymbol{F}}_z - \bar{\boldsymbol{F}}_z \boldsymbol{Z} \boldsymbol{H}_2 (\boldsymbol{H}_2^T \boldsymbol{Z} \boldsymbol{H}_2 + \boldsymbol{R})^{-1} \boldsymbol{H}_2^T \boldsymbol{Z} \bar{\boldsymbol{F}}_z^T + \boldsymbol{H}^T \boldsymbol{Q}_z \boldsymbol{H} - \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T \tag{3.23}$$

In the first equation 3.22, $(\boldsymbol{F}_x, \boldsymbol{G}_x, \boldsymbol{H}_x, \boldsymbol{M}_x)$ is the minimum state space realization of $\boldsymbol{G}_s$, $\bar{\boldsymbol{F}}_x = \boldsymbol{F}_x - \boldsymbol{G}_x \boldsymbol{S}^{-1} \boldsymbol{M}_x^T \boldsymbol{H}_x$, $\boldsymbol{Q}_x = \boldsymbol{H}_x^T \boldsymbol{R}^{-1} \boldsymbol{H}_x$, $\boldsymbol{S} = \boldsymbol{I} + \boldsymbol{M}_x^T \boldsymbol{M}_x$, and $\boldsymbol{R} = \boldsymbol{I} + \boldsymbol{M}_x \boldsymbol{M}_x^T$ [54]. For equation 3.23, $\bar{\boldsymbol{F}}_z = \bar{\boldsymbol{F}}_x^T$, $\boldsymbol{H}_2 = \boldsymbol{H}_x^T$, and $\boldsymbol{Q}_z = \boldsymbol{G}_x \boldsymbol{S}^{-1} \boldsymbol{G}_x^T$. The solution to equations 3.22 and 3.23 can be computed using the *dare()* function in MATLAB, where $\boldsymbol{E} = \boldsymbol{I}$ [33].

After solving the two discrete-time matrix Riccati equations, the H_∞ controller is produced by [22]

$$\boldsymbol{K} = \left[\begin{array}{c|c} \boldsymbol{F}_x + \boldsymbol{G}_x \boldsymbol{F}_\infty + \gamma^2 \boldsymbol{L}^{-1} \boldsymbol{Z} \boldsymbol{H}^T (\boldsymbol{H}_x + \boldsymbol{M} \boldsymbol{F}_\infty) & \gamma^2 \boldsymbol{L}^{-1} \boldsymbol{Z} \boldsymbol{H}_x^T \\ \hline \boldsymbol{G}_x^T \boldsymbol{X} & -\boldsymbol{M}_x^T \end{array} \right] \tag{3.24}$$

where

$$\boldsymbol{F}_\infty = -\boldsymbol{S}^{-1} (\boldsymbol{M}_x^T \boldsymbol{H}_x + \boldsymbol{G}_x \boldsymbol{X}) \tag{3.25}$$

$$\boldsymbol{L} = (1 - \gamma^2) \boldsymbol{I} + \boldsymbol{X}\boldsymbol{Z} \tag{3.26}$$

Since robust stability (RS), for coprime uncertainty, is defined as [54]

$$RS \; \forall \;\; \|\boldsymbol{\Delta}_N \;\; \boldsymbol{\Delta}_M\|_\infty \leq \epsilon \;\; \Leftrightarrow \;\; \|\boldsymbol{M}\|_\infty < \frac{1}{\epsilon} \tag{3.27}$$

and

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{K} \\ \boldsymbol{I} \end{bmatrix} (\boldsymbol{I} - \boldsymbol{G}_s\boldsymbol{K})^{-1}\boldsymbol{M}^{-1} \tag{3.28}$$

The controller solution guarantees that the closed loop system is robustly stable.

If γ is set to exactly equal γ_{min}, it can result in $\boldsymbol{L}$ (calculated in equation 3.26) being singular. To avoid this while solving for both H_∞ controllers, the values for γ are computed as

$$\gamma = 1.1(\gamma_{min}) \tag{3.29}$$

This means that the final γ values for both synthesized controllers are slightly sub-optimal, but both systems are still robustly stabilized. To confirm this assertion numerically, we can compare the γ and $1/\epsilon$ values for both controllers. The γ and $1/\epsilon$ values achieved for H_∞ steering controller 1 and 2 are shown in Table 3.4.

Table 3.4: Robust stability test results.

Controller	$1/\epsilon$	γ
H_∞ Controller 1	5.83	2.87
H_∞ Controller 2	5.94	2.90

The γ values for both controllers meet robust stability requirements - even though they are 110% of the optimal γ_{min} values. The Z-domain block diagram for these closed loop systems is provided in Figure 3.7.

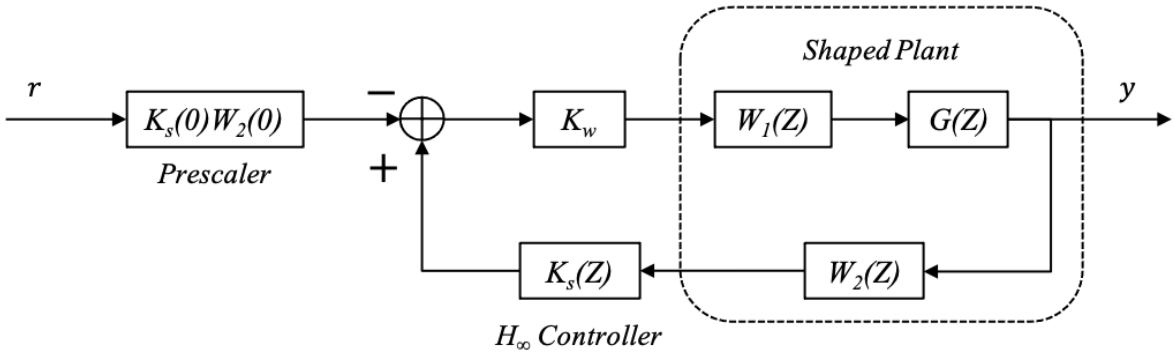

Figure 3.7: Z-domain block diagram for the closed loop steering system (adapted from [54]).

The closed loop system uses positive feedback and includes two additional blocks in the signal routing path. These additional blocks are a reference prescaler and a constant weighting matrix $\boldsymbol{K}_w$. The reference prescaler is computed from the DC gain of $\boldsymbol{K}(Z)\boldsymbol{W}_2(Z)$ and this term eliminates the steady state error of the closed loop system. The constant weighting matrices for controller designs are left as $\boldsymbol{K}_w = \boldsymbol{I}$.

This type of H_∞ design, called the "central controller", has a direct state feedback realization in the time domain. The H_∞ loop-shaping version of the central controller is similar to a Kalman Filter and has a simpler observer form than the general H_∞ version [49] [54].

$$\hat{\boldsymbol{x}}_{t_{k+1}} = \boldsymbol{F}_x \hat{\boldsymbol{x}}_{t_k} + \boldsymbol{Y}\left(\boldsymbol{H}_x \hat{\boldsymbol{x}}_{t_k} - z_{t_k}\left(\frac{1}{N}\right)\right) + \boldsymbol{G}_x u_{t_k} \tag{3.30}$$

$$u_{t_k} = \boldsymbol{K}_s \hat{\boldsymbol{x}}_{t_k} \tag{3.31}$$

where u_{t_k} is the optimal control and

$$\boldsymbol{Y} = -\boldsymbol{Z}\boldsymbol{H}_x^T \tag{3.32}$$

$$\boldsymbol{K}_s = -\boldsymbol{G}_x^T \left[\boldsymbol{I} - \gamma^{-2}\boldsymbol{X}\boldsymbol{Z}\right]^{-1} \boldsymbol{X} \tag{3.33}$$

The block diagram for this central controller in state feedback form is shown in Figure 3.8. Figures 3.9 and 3.10 display the singular values for H_∞ Controller 1 and 2, respectively.

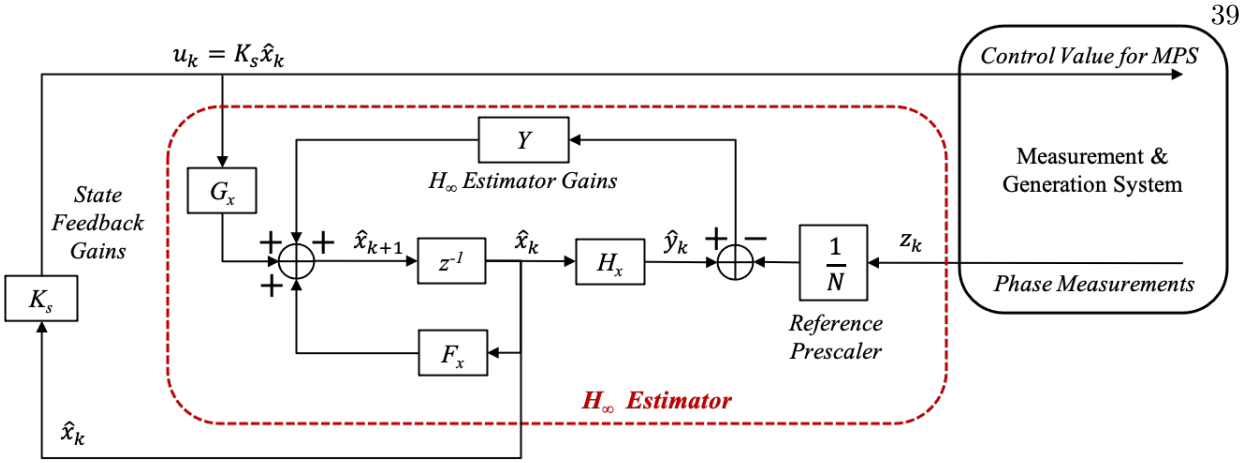

Figure 3.8: State feedback structure block diagram for the H_∞ central controllers.

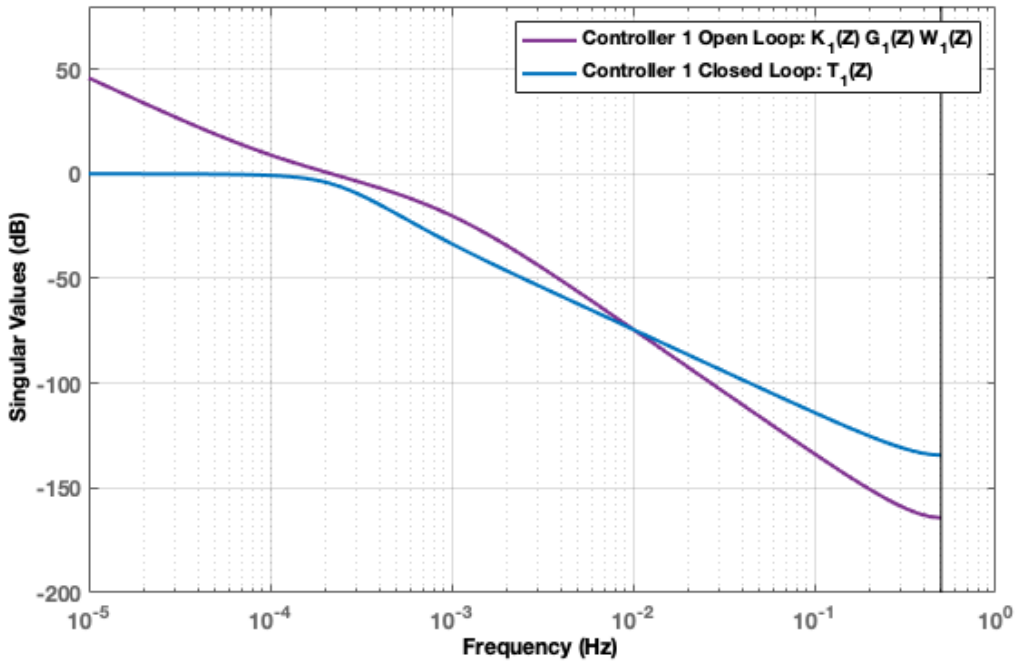

Figure 3.9: Open loop and closed loop singular values for H_∞ steering controller 1.

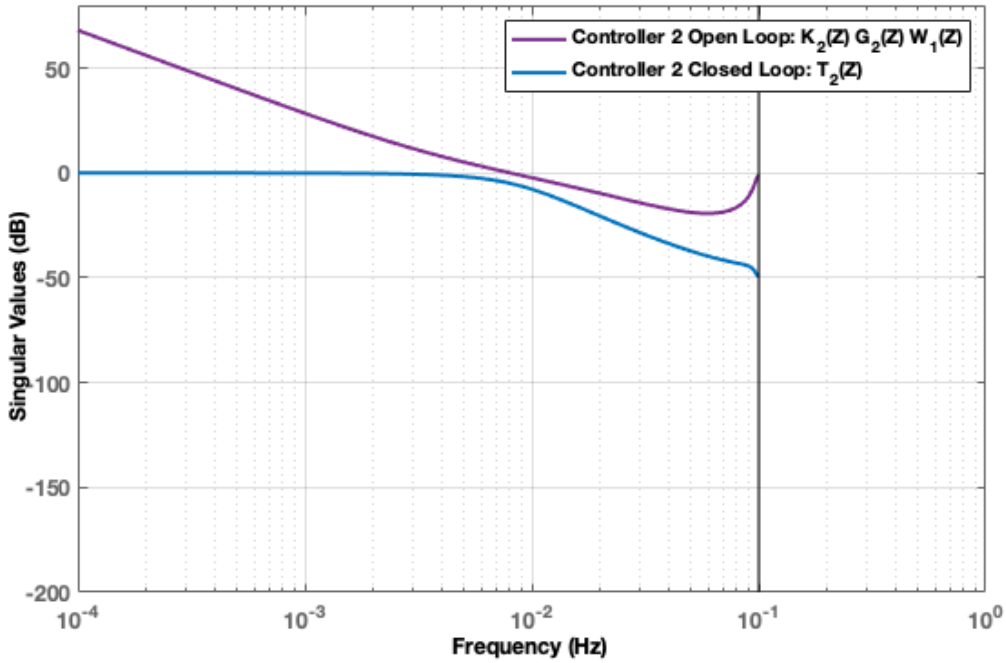

Figure 3.10: Open loop and closed loop singular values for H_∞ steering controller 2.

The maximum singular values of the open and closed loop systems for both controllers exhibit certain similarities, as well as some differences. Both closed loop systems have a DC gain of 0 dB, which is expected due to integral action in the plant and reference prescaling to eliminate steady state error [54]. Both controllers do not significantly affect the open loop singular values, but do improve robustness characteristics. Therefore, we should still expect slightly better noise rejection from controller 1 and superior reference tracking from controller 2.

3.5 Controller Simulation & Analysis

To compare relative stability performance between the two different steering approaches, the unsteered OCXO, cesium reference, and all six steering controllers are simulated for a period of 3.0×10^4 s in MATLAB. In addition to the nominal scenario, we also consider two permutations of the scenario; where process and measurement noise mismodeling are included, in order to observe and compare controller behavior in the presence of modeling errors.

The performance analysis focuses on comparing frequency stability of the best performing

LQG controller with the two H_∞ controllers. The LQG controller is selected from the OADEV results for the nominal scenario. The mismodeling scenario results are only presented for these three controllers. Frequency stability results from the mismodeling scenarios for all six controllers are provided in Appendix A. Figure 3.11 displays the frequency stability results from the nominal steering scenario.

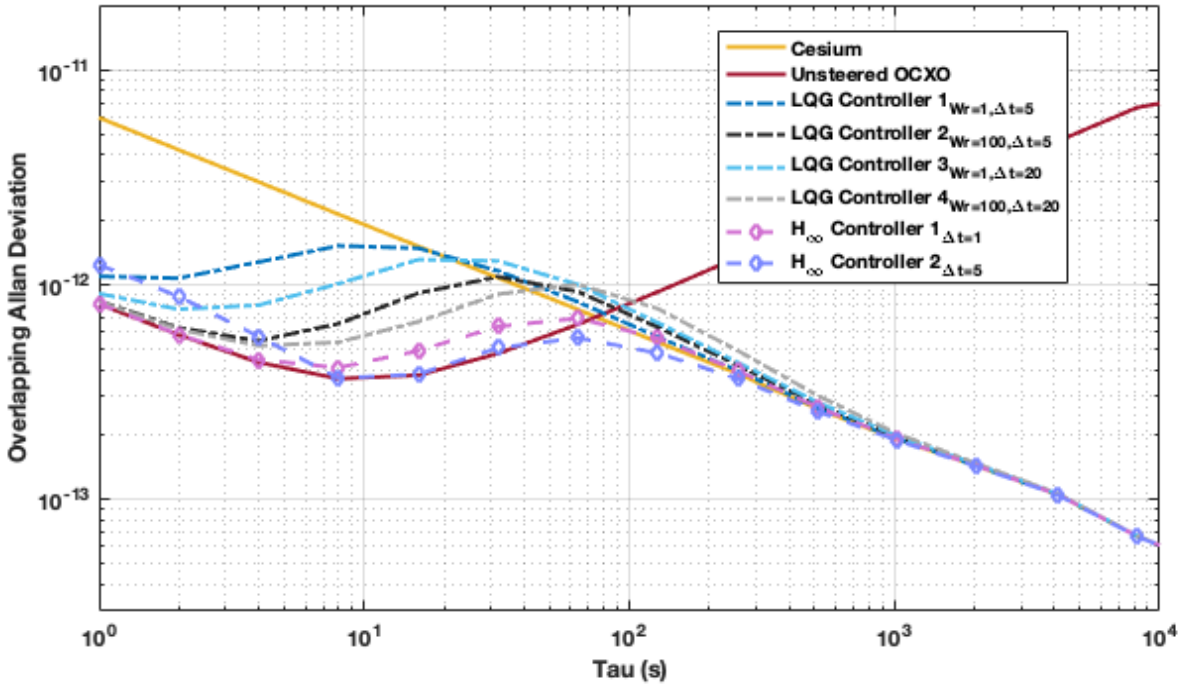

Figure 3.11: Frequency stability of all controllers from the nominal steering scenario.

The stability results computed for the four LQG controllers (dashed lines without symbols) exhibit a similar shape and steering characteristics that make sense, based on their $\boldsymbol{W}_R$ and Δt parameters. LQG controllers with the larger $\boldsymbol{W}_R$ value steer to the cesium reference more slowly than their counterparts using the same control interval size. This makes sense because the larger $\boldsymbol{W}_R$ value imposes a higher penalty on actuator authority - so the controller cannot drive the reference error to zero as quickly. The two LQG controllers exhibiting the best steering performance, for this nominal case, are controller 1 and controller 2. Since LQG controller 2 ($\boldsymbol{W}_R = 100$, $\Delta t = 5\ s$) does a slightly better job of preserving the short-term performance of the

OCXO, it is the controller selected for further comparison with the two H_∞ controllers.

The nominal results for these controllers are shown again in Figure 3.12. Frequency stability observed for the two H_∞ controllers exhibit steering performance superior to all four LQG controllers in this nominal scenario. Both controllers do an excellent job in preserving the short-term stability of the OCXO and steer to the cesium reference around the crossing point of the cesium and free running OCXO. Furthermore, H_∞ controller 2 is able to achieve better stability than both the free running OCXO and the cesium around the crossing point where they are close in stability. It is similar to characteristic OADEV performance seen in IEM predictions from [57], over averaging intervals where the stability of ensemble members are close together. This is where the ensembling filter (or in this case, the controller) can take advantage of the similarities in frequency stability of the clocks for this range of time intervals. In addition, time series results for the phase and frequency deviations are provided in Figure 3.13.

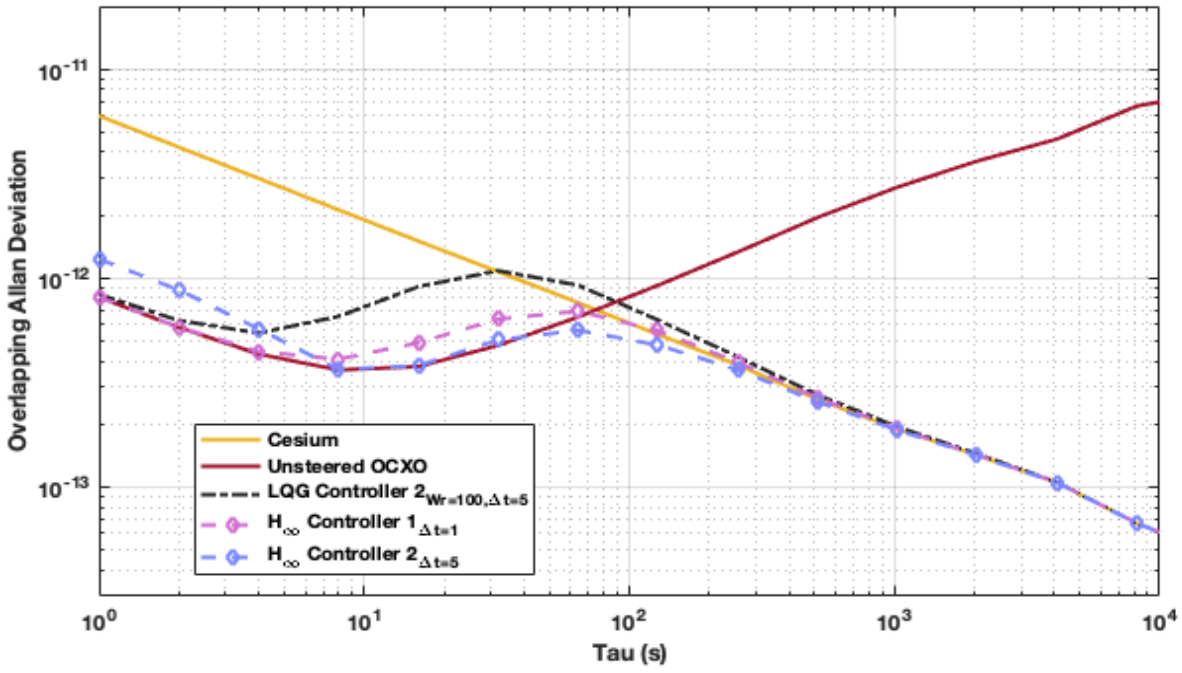

Figure 3.12: Frequency stability of LQG controller 2 and the H_∞ controllers for the nominal scenario.

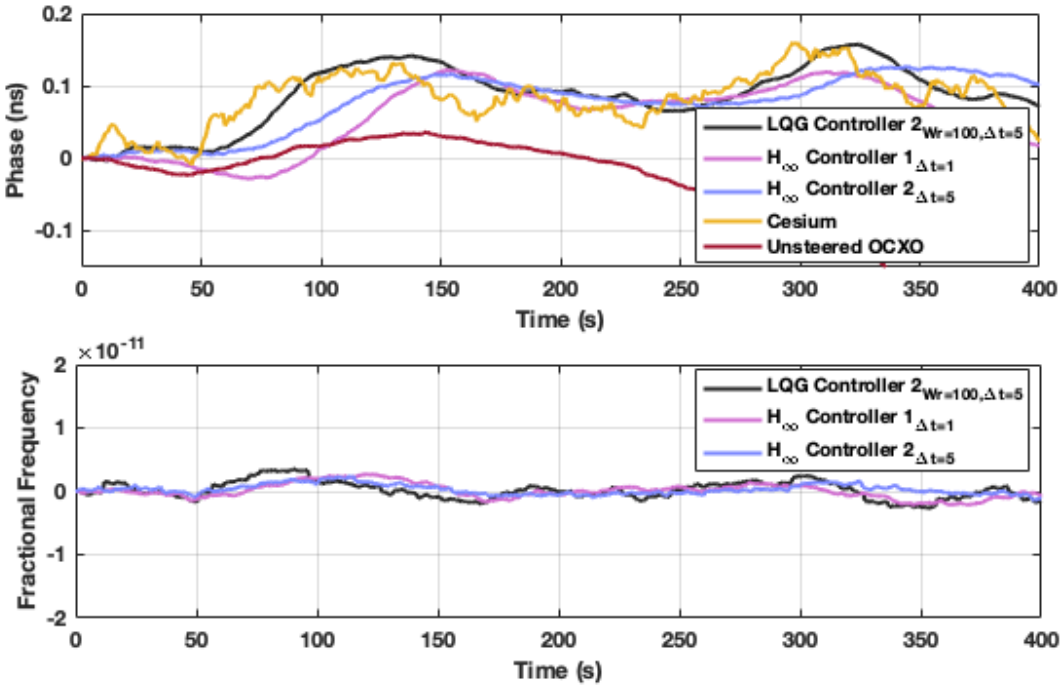

Figure 3.13: Phase and frequency deviations from the nominal steering scenario.

For comparison with results from the mismodeling scenarios, the y-axis scale for frequency deviations is kept constant. Fractional frequency errors from the selected controllers are similar in this scenario. However, the LQG controller output exhibits slightly larger frequency errors between $t = 50\ s$ and $t = 175\ s$.

The first additional steering scenario introduces mismodeling errors by increasing the assumed process noise variance in the clock simulation by a factor of 10 greater than the characteristic variance (used for the Kalman Filter parameters in the LQG controller). The OADEV results for this steering scenario are shown below in Figure 3.14. While the H_∞ optimization applied does not require a priori knowledge of system noise, the weighting functions for plant shaping were tuned based on the characteristic short-term stability of the OCXO. We should expect this mismodeling error to affect all of the controllers, but potentially have a more adverse effect on the LQG controller because the Kalman Filter is too optimistic about the process noise.

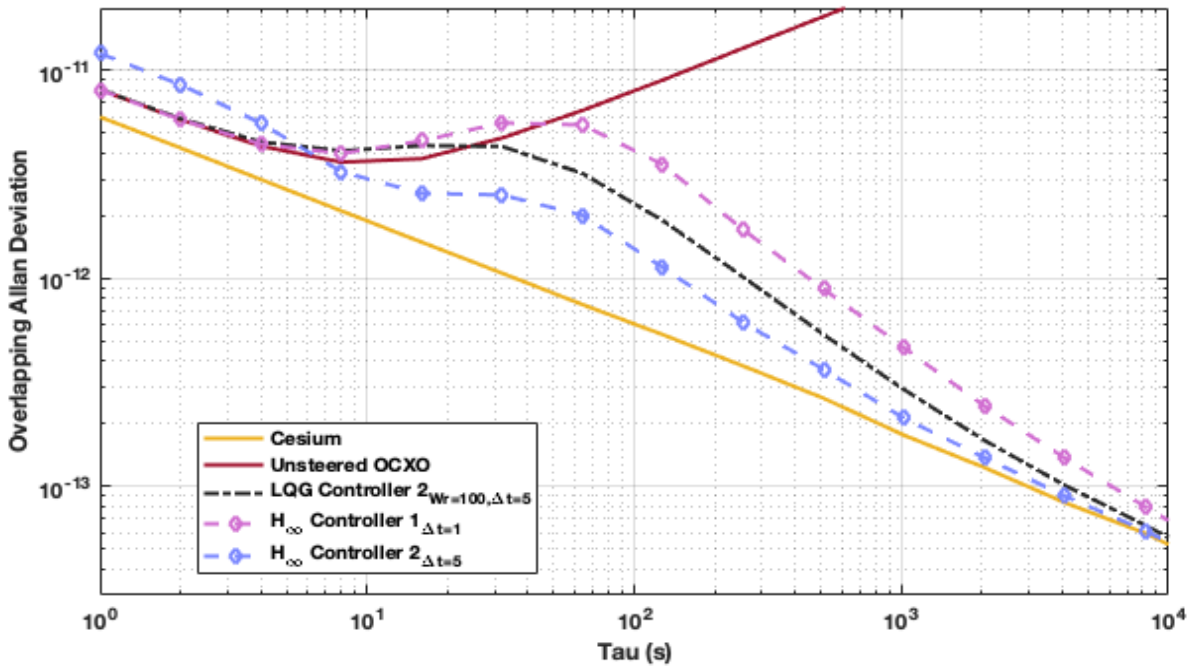

Figure 3.14: Frequency stability of LQG controller 2 and the H_∞ controllers, with process noise variance increased in the truth data by a factor of 10.

With mismodeling included in the process noise for the truth data, we observe that steering performance seen in the nominal scenario is degraded for all three controllers. H_∞ controller 2 does provide better frequency stability over the interval $5\,s \geq \tau \geq 8000\,s$, but the LQG controller out-performs H_∞ controller 1 over $\tau \geq 10\,s$. The overall performance of H_∞ controller 1 is degraded the most out of the three controllers. While this type of mismodeling error negatively affects the steering performance of every controller, they are still able to successfully steer to the reference clock. The time series for phase and frequency deviations are provided in Figure 3.15.

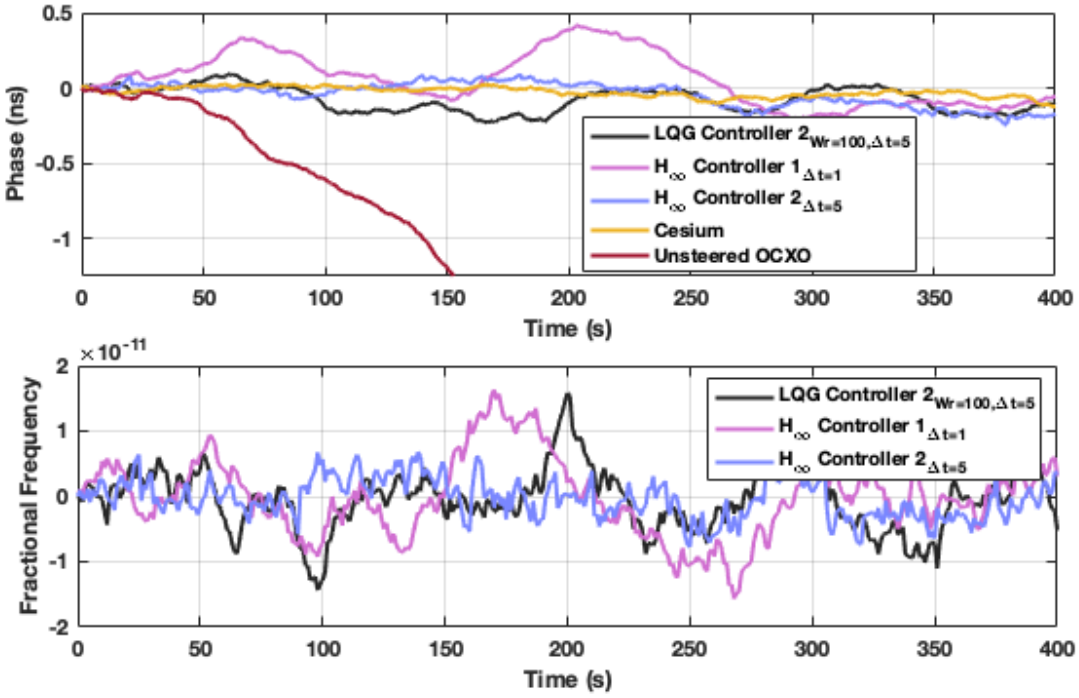

Figure 3.15: Phase and frequency deviations from the process noise mismodeling scenario.

Again, the time series of fractional frequency errors for LQG controller 2 and both H_∞ controllers are similar. The frequency deviations from this scenario are an order of magnitude larger than seen from the nominal scenario, which corresponds $\times 10$ increase in truth data process noise. The slightly better performance of H_∞ controller 2 is most easily observed in the phase deviations.

The second additional steering scenario incorporates increased measurement noise in the truth data, and results are presented in Figure 3.16. Here, the variance of the actual measurement noise is increased by a factor of 100; such that the controller design is based on an optimistic model for measurement errors. In this scenario, we expect to also see a dramatic effect in steering performance for the LQG controller, based on results observed from a similar case in [57]. We should also expect to see frequency stability similar to the nominal scenario for the H_∞ controller, based on the optimization performed for robust stabilization.

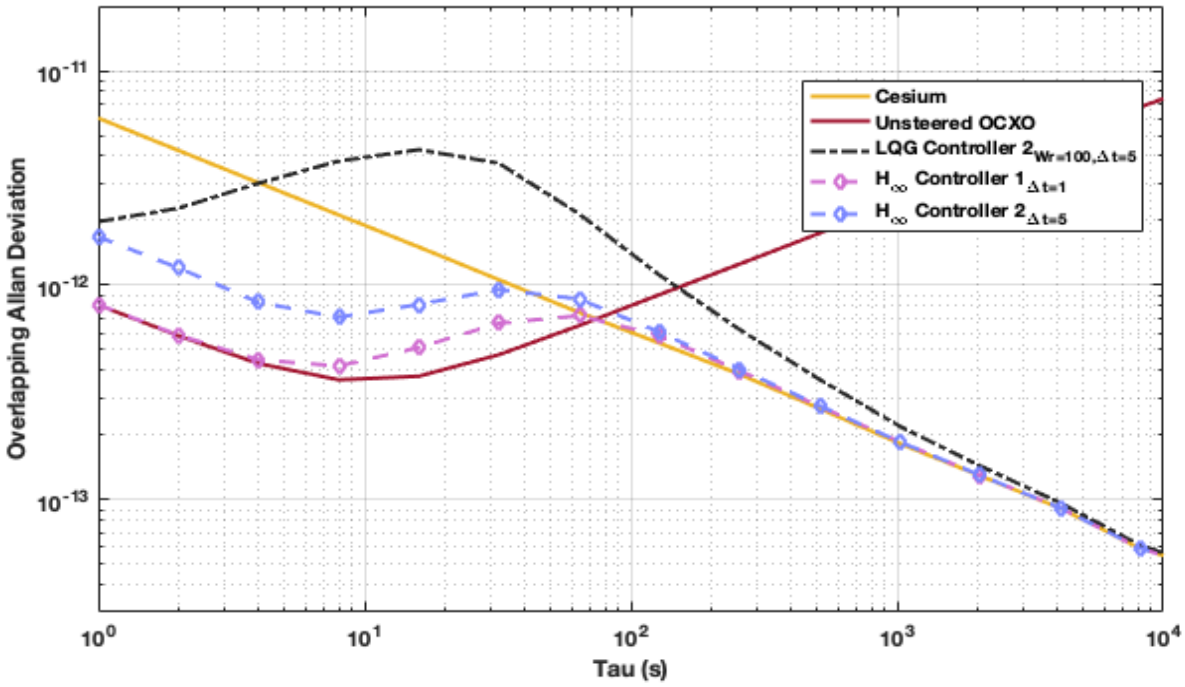

Figure 3.16: Frequency stability of LQG controller 2 and the H_∞ controllers, with measurement noise variance increased in the truth data by a factor of 100.

The simulation of measurement noise mismodeling shows a severe degradation in LQG steering performance over shorter time intervals. However, H_∞ controller 2 exhibits only a slight decrease in steering performance, and the performance of H_∞ controller 1 is completely unaffected. This makes sense intuitively since the H_∞ controllers do not rely on a priori knowledge of measurement noise and the open loop gain of controller 1 is much lower at high frequencies than controller 2. The LQG controller is able to steer to the reference around the same averaging interval in both cases of mismodeling. This effect is seen in the results for all four LQG controllers in Figure A.3 (Appendix A). By analyzing the time series of the short-term phase and frequency deviations we can gain further insight to the OADEV results for this mismodeling scenario. The phase and frequency deviation results are shown below in Figure 3.17.

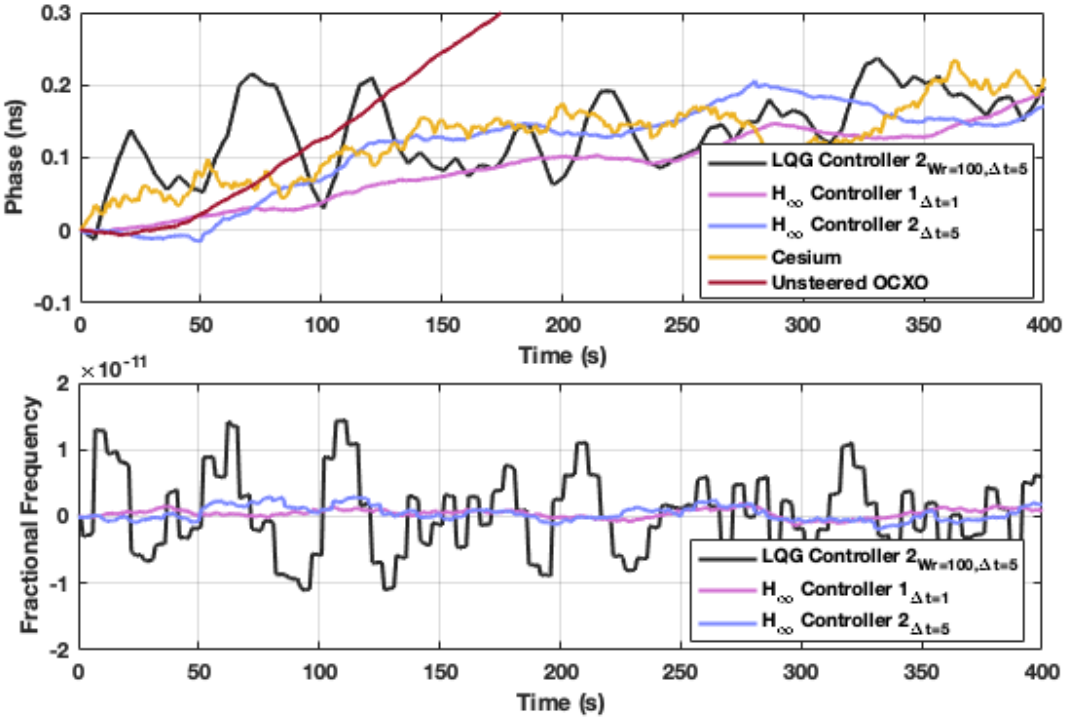

Figure 3.17: Phase and frequency deviations from the measurement noise mismodeling scenario.

In comparison to the time series results from the two previous scenarios, we see much larger spikes in frequency deviations from the LQG controller corrections - occuring at the five second control intervals. The controlled output from both H_∞ systems do not exhibit these characteristics and perform significantly better. The time series data provides a good illustration of the fundamental differences between the two control methods.

3.6 Steering Analysis Conclusions

The results from this analysis illustrate that H_∞ loop-shaping and optimization is a promising alternative approach for clock steering. Numerical simulations demonstrated that this design and control method can provide better frequency performance than both the steering reference and unsteered clock performance - in cases where the unsteered short-term stability is better than the steering reference. Both H_∞ controllers were able to significantly outperform all four LQG controllers in the nominal and measurement noise mismodeling scenarios. In addition, expected

performance from the designs for open loop singular value shapes were confirmed. However, even though H_∞ controller 1 did exhibit slightly better disturbance rejection from increased measurement noise, controller 2 displayed better overall performance robustness.

The results from these simulations are highly relevant, since a similar LQG control approach was applied to steering in clock ensembles by [57] and [24] in both simulation and hardware implementation. Furthermore, [48] and [24] demonstrated that simulated performance showed fairly good agreement with experimental performance. This provides support for the use of H_∞ loop-shaping and optimization techniques for IEM realization in the low-SWaP clock ensemble testbed.

Chapter 4

USRP Front-End Analysis & Baseband Steering System Design

4.1 Overview

The second half of this thesis begins with an analysis of the testbed hardware platform and design considerations for implementing the signal generation design with the N310 USRP. This chapter discusses the front-end structure of the N310 USRP and modifications to the nominal signal generation method, presented in Chapter 2, for baseband steering. Beginning with the front-end analysis, the block diagram in Figure 4.1 provides a high level description of the hardware/software interface.

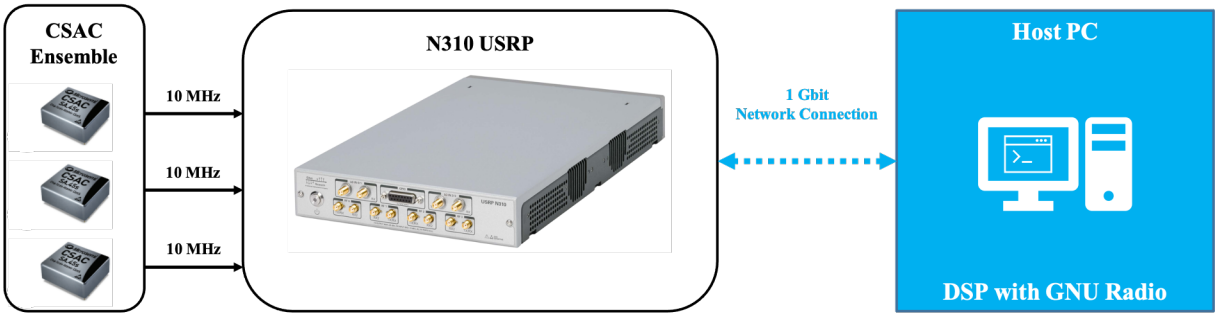

Figure 4.1: Functional block diagram for the testbed hardware/software interface [37] [18].

4.2 N310 USRP Front-End Analysis

The current testbed implementation approach seeks to use a software defined radio platform (Ettus N310 USRP) to host the CSAC ensemble. The 10 MHz CSAC signals are input to the

N310 USRP, where the signals are discretized for off-board digital signal processing (DSP) on a host PC via a 1 gigabit network connection. However, prior to sampling and discretization, the CSAC timing signals are routed through an analog signal processing chain on the front-end of the N310 daughterboards (recall Figure 1.6 from Chapter 1). The signal processing chain conditions the analog input signals to be at the center of the operational frequency range for the transceiver ADC. This operates on the 10 MHz CSAC output. Signals lower than 1.5 GHz are routed through a low-band upconverter. Likewise, on the TX side, the steered output signal is generated at a higher frequency and then routed through an analog low-band downconverter.

The additional up and downconverter chains for the RX and TX ports on the daughterboard front-ends pose potential concerns for steered signal generation with this hardware platform. If the front-end processing chains generate additional phase noise in the signals, then this will degrade steered output performance. To explore these concerns, the following analysis steps through each front-end operation. Beginning at the RX port and ending at the ADC output, the following equations describe the signal operations to the input signal $S_{in}(t)$ - highlighted in Figure 4.2.

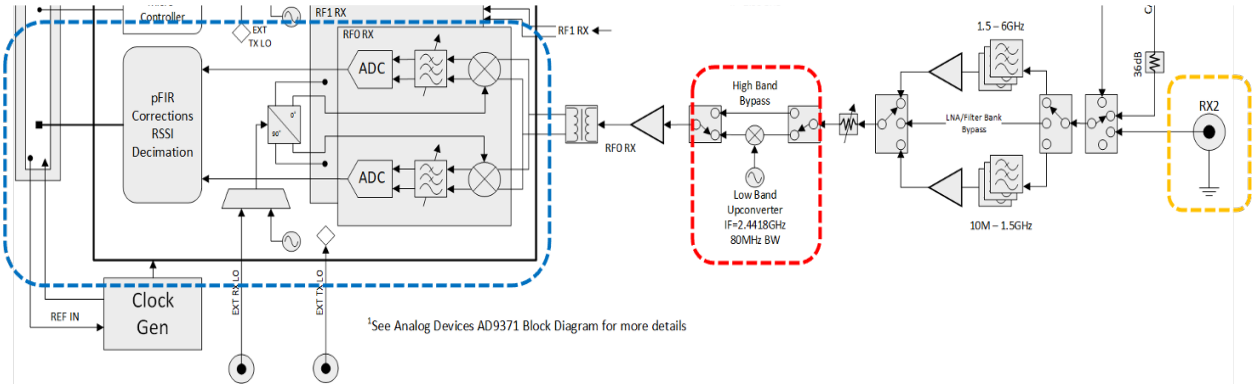

Figure 4.2: RX analog front-end chain [18].

Using the timing signal voltage model from equation (1.1), the input signal to the RX port (yellow box) is written as

$$S_{in}(t) = [V_o + \varepsilon(t)]cos[2\pi f_n t + \phi_{in}(t)] \tag{4.1}$$

where $\phi_{in}(t)$ is the phase of the input signal. To simplify equation descriptions, $\omega_{in} = 2\pi f_n t$ and

the amplitude term will be written as A. Moving right to left across the RX chain, the output from the low-band upconverter (red box) is written as

$$S_{LB,in}(t) = A\left[cos\left((\omega_{in} + \omega_{LB,in})t + \phi_{in}(t) + \phi_{LB,in}(t)\right)\right] \tag{4.2}$$

where $\phi_{LB,in}(t)$ and $\omega_{LB,in}$ are the phase and angular frequency of the local oscillator (LO) for the low-band upconverter. The last operation on the RX front-end chain is the downconversion and sampling stage performed by the complex ADC (blue box). The downconverted and sampled output from the complex ADC is

$$S_{DDC}(t_k) = A\left[cos\left((\omega_{in} + \omega_{LB,in} - \omega_{DDC})t_k + \phi_{in}(t_k) + \phi_{LB,in}(t_k) - \phi_{DDC}(t_k)\right)\right] \tag{4.3}$$

Here, $\phi_{DDC}(t_k)$ and ω_{DDC} are the phase and frequency of the downconversion mixers, and this discretized signal is passed to the host PC for signal processing in GNU Radio. Unlike the low-band mixers, the user has precise control over the tuning of the transceiver mixers with commands from the host PC.

Now we will refer to Figure 4.3 to front-end operations to the output signal from the N310.

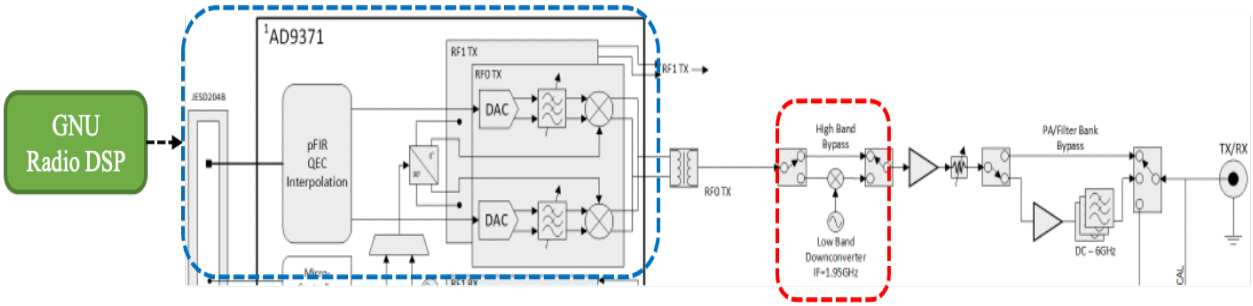

Figure 4.3: TX analog front-end chain [18].

The digitally processed input to the complex DAC is

$$S_{DSP}(t_k) = A\left[cos\left((\omega_{in} + \omega_{LB,in} - \omega_{DDC})t_k + \phi_{in}(t_k) + \phi_{LB,in}(t_k) - \phi_{DDC}(t_k)\right)\right] \tag{4.4}$$

When steering operations are performed, this results in an integrated frequency change in the form of an additional phase term. The analog voltage signal produced by the DAC and upconverter

(blue box) is written as

$$S_{DUC}(t) = A\left[cos\left((\omega_{in} + \omega_{LB,in})t + \phi_{in}(t) + \phi_{LB,in}(t)\right)\right] \tag{4.5}$$

If the LOs for the ADC and DAC up and downconverters in the transceiver are synchronized, the $\phi_{DDC}(t_k)$ and ω_{DDC} terms will cancel out. Then, the final output from the TX port, after passing through the low-band downconverter (red box) is

$$S_{out}(t) = A\left[cos\left(\omega_{in}t + \phi_{in}(t) + \phi_{LB,in}(t) - \phi_{LB,out}(t)\right)\right] \tag{4.6}$$

The angular frequency term $\omega_{LB,in}$ is wiped off by the downconverter, and if the LOs for the TX and RX low-band mixers are synchronized, the $\phi_{LB,in}(t)$ and $\phi_{LB,out}$ terms should also cancel out. Given that the LO's for the low-band mixers can be synchronized using an external 300 MHz reference clock, this analysis shows that there should be minimal phase noise introduced to the signal from the front-end operations - assuming the front-end amplifiers do not add AM noise.

Concerns surrounding peripheral operations to the USRP input and output signals have now been addressed and the discussion can move forward to the next topic covered. The second part of this chapter focuses on the input and output modifications to the nominal steering method to operate within the architecture between the analog front end.

4.3 Technical Differences Between Steering Methods

A functional block diagram of the modified baseband steering system design is provided in Figure 4.4. This illustrates the slight alterations in the designs, where the entire nominal design is re-positioned between down and upconverters, and operates as a fully digital system.

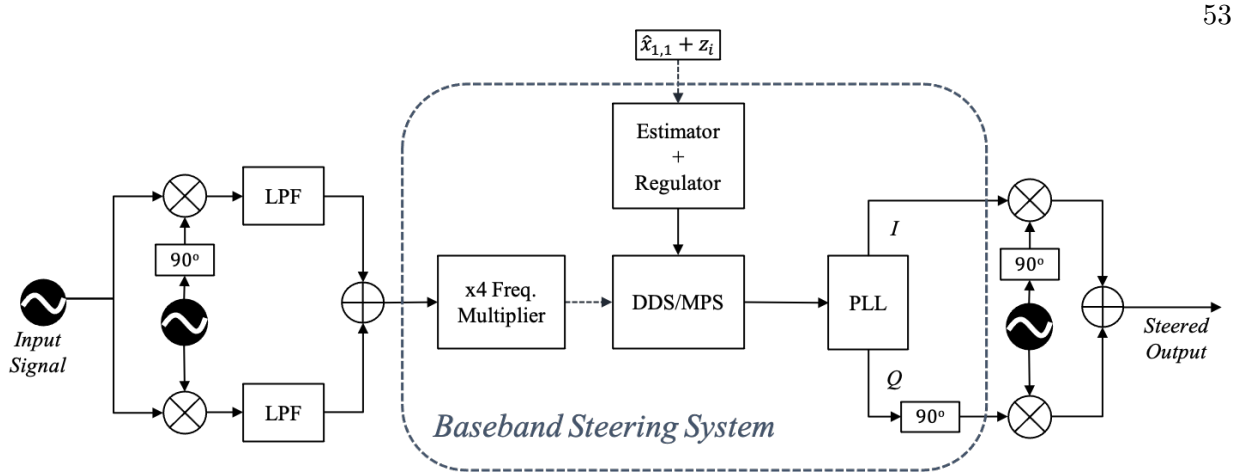

Figure 4.4: Block diagram for the baseband steering system design approach.

There are several important differences between the nominal steering system design and the modified baseband steering system. The first, and most obvious difference, is that the digitally synthesized steered signal is much lower in frequency. The second difference is that the steered signal smoothing operation is also performed digitally, and at the downconverted frequency. Lastly, to produce the final steered output, there is an additional single-sideband (SSB) upconversion operation needed to shift the steered signal back up to the nominal frequency. Each of these differences also vary in how they effect the steering process.

Steered signal synthesis at a down converted frequency has both potential advantages and disadvantages. The main advantage to working with a lower frequency is that, the minimum sampling rate needed to operate on the signal to prevent aliasing issues is significantly reduced. This alleviates stress on computing resources and memory usage for data post-processing. Conversely, a potential disadvantage is that applied steering corrections or offset commands are much larger in relative fractional frequency size - which is illustrated in the following equation

$$\frac{\Delta f_{steer}}{f_{baseband}} = \frac{\Delta f_{steer}}{(f_n - f_{DDC})} \tag{4.7}$$

where f_{DDC} is the frequency of the sine/cosine mixers in the downconversion chain.

The other two technical differences do not provide any potential advantages for baseband steering. Since the smoothing operation is performed digitally with a PLL, there is inherent quantization noise that will still be present after the signal has been smoothed. As shown in Figure 4.4,

the digital smoothing PLL produces a complex steered output, with the quadrature portion shifted 90^o to remove the lower sideband during upconversion. This results in the quantization noise in the baseband steered signal bleeding into the upconverted output, during the complex mixing operation. This could be a potential area of performance loss in the baseband steering method. As well, any amplitude mismatch due to noise can also degrade sideband wipe-off. The experiments and analysis discussed in the following chapter use low frequency input signals to investigate if these differences in steering mechanics degrade performance.

Chapter 5

Steering System Design Implementation & Performance Analysis

5.1 Overview

To investigate the potential steering performance differences between the direct and baseband steering methods at a fundamental level, this study applies the simplification of using low frequency input signals. This removes the additional technical challenges present when working with signals in the RF range and the other factors in the N310 USRP. Since analysis of both long-term and short-term performance is of interest, data must be collected over long periods of time. Using low frequency signals reduces minimum sampling rate demands and data storage for hardware experimentation and data analysis.

5.2 Steering Experiment Parameters

The following experiments demonstrate the ability to implement both steering system designs with hardware. Both system designs were implemented using a digital controller. Refer to Figure 5.1 for an overview of all the hardware components. Each steering system design was implemented with a myRIO digital controller. The steering systems were programmed to generate a steered output signal, based on an input signal from an external function generator, with a 1 Hz frequency offset. A *Koolertron* CJDS66 function generator provided a 53 Hz analog input signal to the digital controller, and an Arduino Uno microcontroller was used to sample and store the analog voltages from the input and steered output signals. The input signal frequency of 53 Hz was selected such that it would not be commensurate with the Arduino and myRIO sampling rates.

The nominal design implementation directly steers at the frequency of the input signals. The baseband implementation downconverts the input signal to 5 Hz prior to steering, and upconverts the steered output.

The hardware configuration for the myRIO is as follows, based on manufacturer specifications in [43], and a block diagram for the entire hardware system is provided in Figure 5.1. The 12 bit A/D converter samples the input signal at a rate of 500 kS/s and writes the sample voltage values to the Xilinx Z-7010 667 MHZ dual core FPGA. The FPGA reads in the samples and passes them to the ARM processor - when called by the deployed real-time (RT) algorithm. The RT algorithms run inside of a 1 kHz timed loop. In each loop iteration, a sample is retrieved from the FPGA, operated on, and passed back to the FPGA. The samples passed back to the FPGA are written to the 12 bit D/A converter, running at a rate of 345 kS/s, and output as an analog voltage.

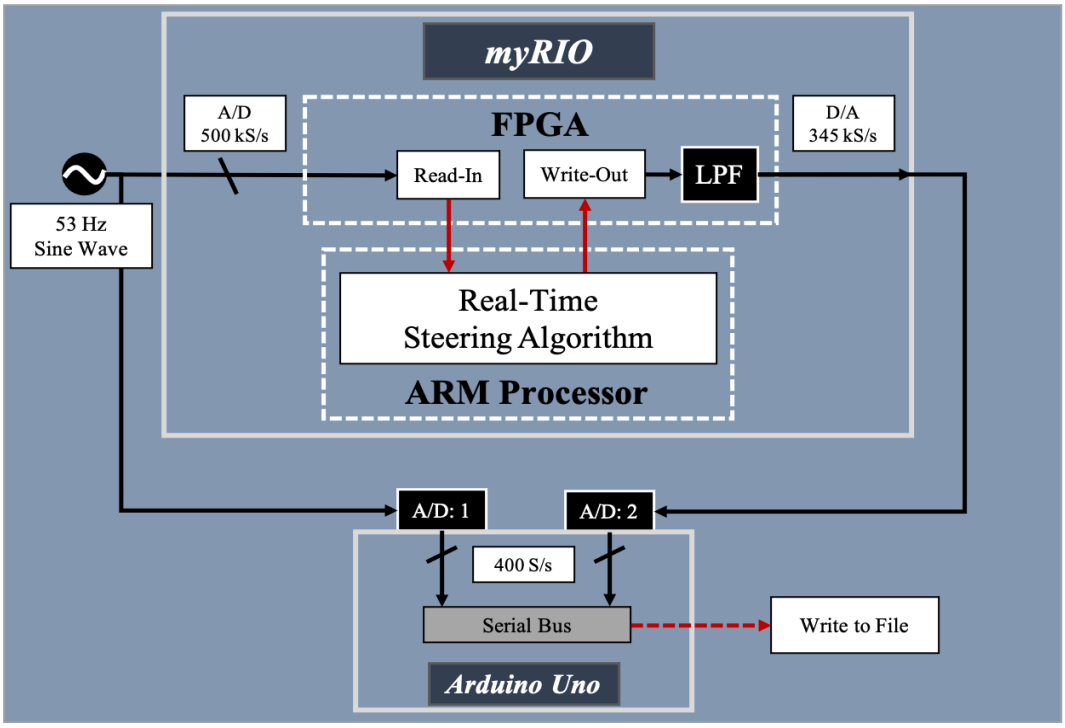

Figure 5.1: Block diagram for the myRIO hardware and software configuration.

The default FPGA image on the myRIO was customized using LabVIEW FPGA. This customization simply removed additional pre-configured operations to read from and write to the all

of the other DIO and AIO ports on the myRIO, and low pass filter the output sample stream before writing to the D/A converter. Using the customized FPGA code, the RT algorithms running on the ARM processor need to appropriately scale the sample values pulled from and passed back to the FPGA - based on the A/D and D/A converter resolution. Both converters have a maximum resolution of 12 bits, so the conversion factor between Volts and counts/Volt is

$$\frac{counts}{Volt} = \frac{2^{12}}{20} = 204.8 \tag{5.1}$$

$$Volts = \frac{1}{204.8} \tag{5.2}$$

This scaling factor is applied at the beginning and end of the RT algorithms running in timed loop structures on the Zync-7010 ARM processor.

5.2.1 Nominal Design Implementation

This steering system is a fully digital implementation of the nominal signal generation subsystem design - tuned for a 53 Hz input signal. The Real-Time target algorithm, developed in LabVIEW, is compiled and deployed from a host PC onto the myRIO. The CJDS66 function generator, myRIO, and Arduino Uno are connected via a bread board and the Arduino serial bus communicates with the host PC over a UART connection. A block diagram for the nominal steering system experiment setup is shown in Figure 5.2.

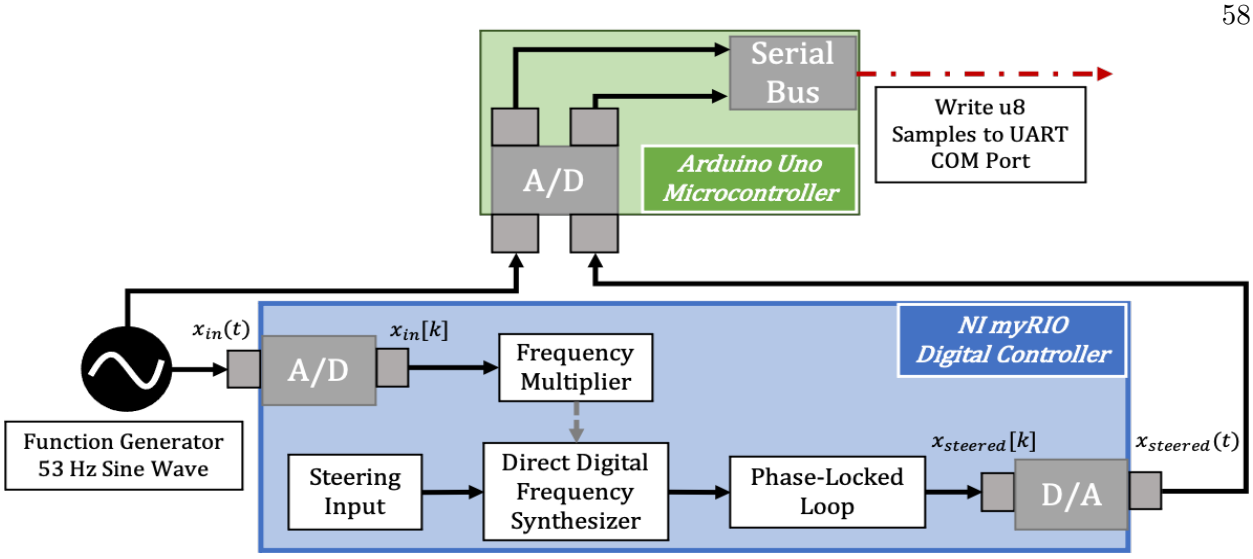

Figure 5.2: Experimental block diagram for the nominal steering system.

The 53 Hz sine wave from the function generator is input to both, the myRIO analog input port C/AI0 and the Arduino analog input port AO, and sampled by the A/D converters in both controllers. The myRIO performs digital steering operations on the input voltage samples and outputs an analog steered signal to the Arduino A2 analog input port. Due to the 0-5 volt analog voltage range of the Arduino A/D converter, the function generator and myRIO apply a +1.5 volt offset to their analog output signals. The Arduino Uno samples the analog voltages at a rate of 400 Hz, and writes the unsigned 8 bit sample values and sample time stamps to the serial bus at a rate of 115200 bits per second. The serial data are recorded by the host PC and post-processed in MATLAB.

The signal processing chain for the system is built within a timed loop structure, executing at a rate of 1 kHz. The timed loop uses a 1 MHz on-board timing source as its clock. Inside of the timed loop structure, the *Connector C/AI0* function pulls in the most recent sample value read by the FPGA. The retrieved voltage sample is appropriately scaled and input to the frequency multiplier chain displayed in Figure 5.3.

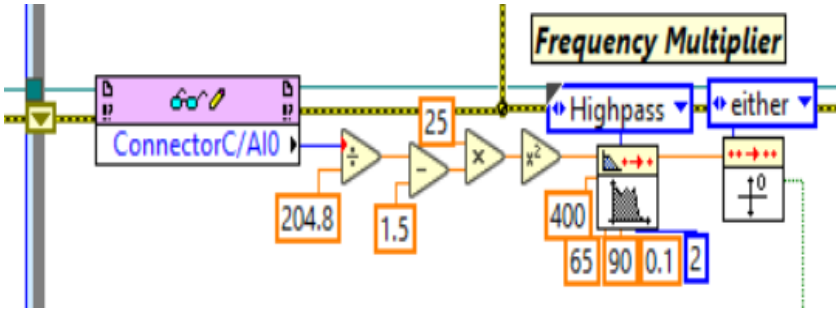

Figure 5.3: DSP chain for the nominal steering system frequency multiplier. Filter and gain values were tuned by spectral analysis of the frequency multiplied output signal.

This sequence in the LabVIEW code removes the +1.5 Volt offset and multiplies the frequency of the input signal by a factor of 4 as a Boolean type signal - increasing the frequency from 53 Hz to 212 Hz. This signal was then used as the clock source for the DDS chain.

The direct digital synthesis routine generates the frequency steered signal, and the LabVIEW code for this function is provided in Figure 5.4. The DDS components, described in Chapter 2, operate within a *True/False* structure so that the system can be clocked by the frequency multiplied input signal. On every *True* input, M updates the phase register, and the phase value is added to the phase accumulator. Also within the True/False structure, is the offset input function, which is used to update the value of M by increasing the frequency of the signal by 0.005 Hz once per second. This operation is performed by counting the number of *True* cycles, and every instance that $mod(count, f_{clk}) = 0$, a Δf of 0.005 Hz is added to the calculation of M.

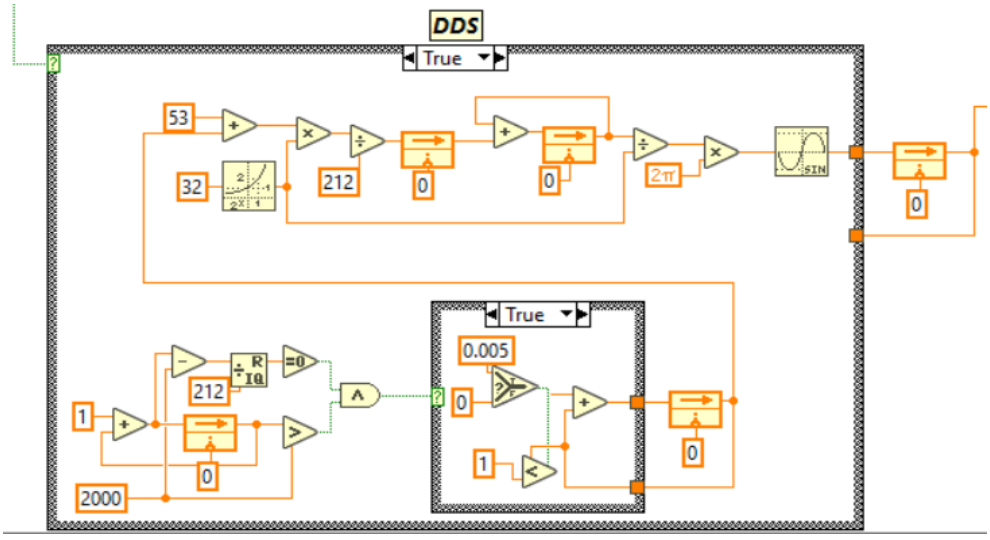

Figure 5.4: LabVIEW block diagram for the direct digital synthesis algorithm, given a *True* message input from the frequency multiplier.

Given the low resolution of the DDS output, the signal spectrum contains relatively high power in the harmonics of the nominal output frequency. These harmonics are removed and a smoothed version of the steered signal is produced by low pass filtering the DDS output and using the filtered signal to drive a PLL. In these digital implementations for both, the nominal steering system and baseband steering system, the VCO is represented by a numerically controlled oscillator (NCO). The LabVIEW code for the PLL is shown in Figure 5.5.

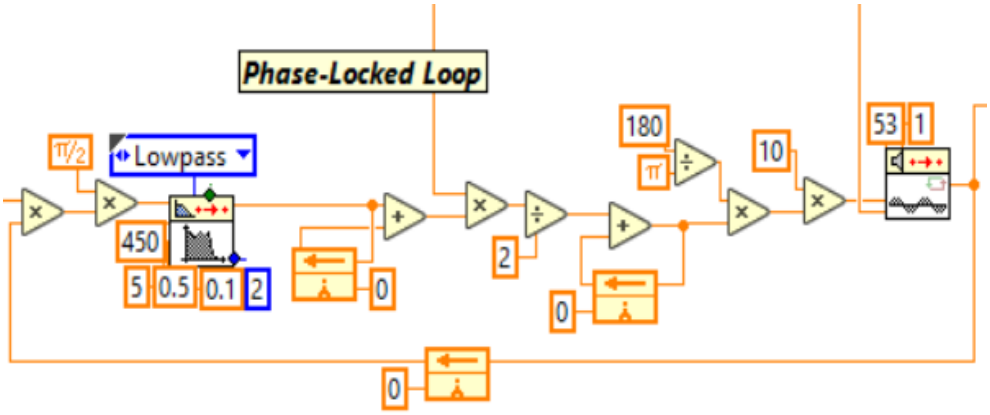

Figure 5.5: LabVIEW block diagram for the phase-locked loop algorithm.

In this PLL, the loop filter is a second order low pass filter, with a cutoff frequency of 0.5 Hz.

The filter output is numerically integrated and converted from radians to degrees. Numerical integration is performed using a central difference method, which is defined as:

$$\phi_{e,k} = \phi_{e,k-1} + \frac{(\omega_{e,k} + \omega_{e,k-1})\Delta t_k}{2} \tag{5.3}$$

Where ϕ_e is phase error, ω_e is frequency error, and Δt_k is the real-time time step. After gain is applied to the phase error value, it is input to the phase offset port of a sine wave generator function (VCO) - driven by the RT microsecond tick-counter and a free-running frequency of 53 Hz. The VCO output is the steered and smoothed signal and is fed back to the phase detector to complete the feedback loop. Before the digitally steered signal is passed back to the FPGA, the +1.5 voltage offset is reapplied and the value is scaled in terms of counts per volt. This concludes the description of the nominal RT steering algorithm.

5.2.2 Baseband Design Implementation

The baseband implementation is very similar to the nominal steering system. The main differences between the two systems are in additions of the digital downconversion chain (DDC) and digital upconversion chain (DUC) in the myRIO RT program. The experimental setup for the hardware devices is identical to the nominal steering system experiment. A block diagram for the baseband steering system experiment is provided in Figure 5.6.

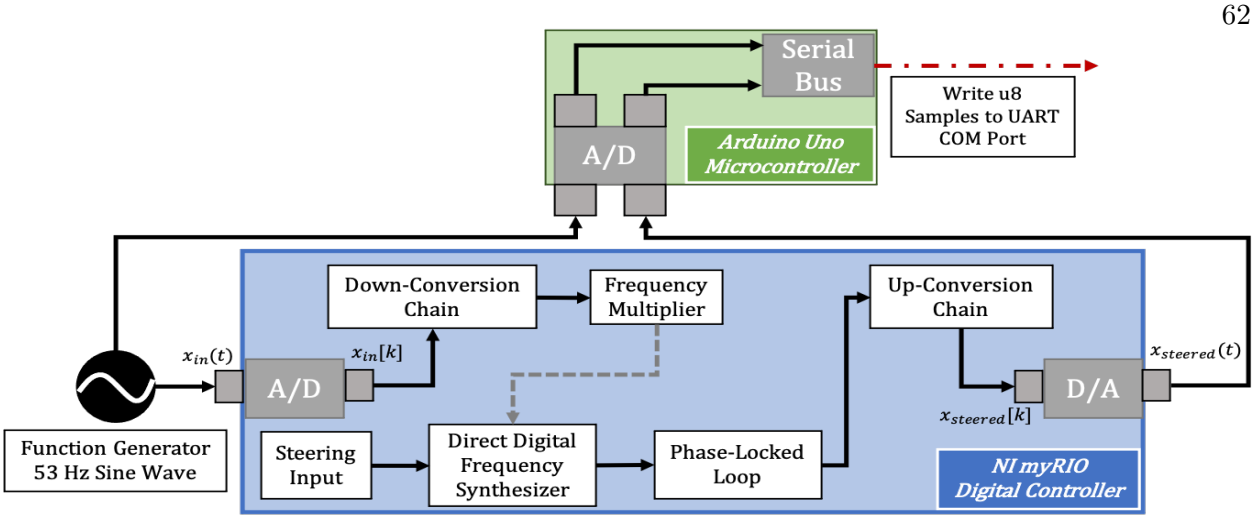

Figure 5.6: Experimental block diagram for the baseband steering system.

Block diagrams for the DDC and DUC chains are displayed in Figure 5.7. The DDC chain shifts down the frequency of the real discrete-time 53 Hz signal to a complex 5 Hz signal by means of a mixing and filtering operation. In the upconversion process, the quadrature component (imaginary part) of the complex steered signal is phase-shifted by 90^o such that the lower sideband is eliminated by amplitude cancellation when the real and imaginary parts of the upconverted signal are combined in the summing amplifier. This up-conversion technique provides a means to transmit a steered signal without unwanted lower-sideband noise, while working within the confines of a software radio architecture. This single-sideband complex transmission technique is described in detail in [19].

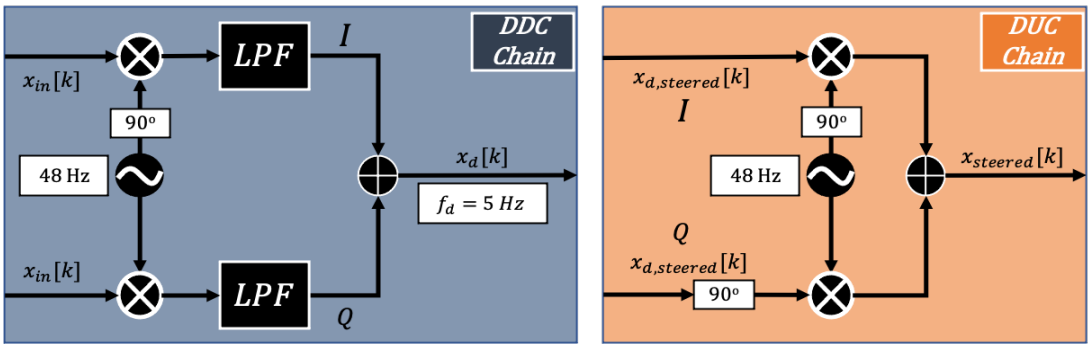

Figure 5.7: Left: myRIO downconversion chain. Right: myRIO upconversion chain.

In the DDC chain, $x_{in,k}$ is the sampled input signal and $x_{d,k}$ is the downconverted signal. In the DUC chain, $x_{d,steered,k}$ is the complex downconverted steered signal and $x_{steered,k}$ is the upconverted steered signal.

Like the nominal steering algorithm, the baseband system algorithm operates within a timed loop structure. However, unlike the nominal system, the sampled signal is first input to the DDC chain to downconvert the frequency of the signal before being passed to the frequency multiplier. The DDC mixes the input signal with a 48 Hz cosine wave and a 48 Hz sine wave to produce two signals with high and low frequency terms in each:

$$I\ Mixer:\ sin(2\pi(53)t_k)cos(2\pi(48)t_k) = \frac{1}{2}\left[sin(2\pi(5)t_k) + sin(2\pi(101)t_k)\right]$$

$$Q\ Mixer:\ sin(2\pi(53)t_k)sin(2\pi(48)t_k) = \frac{1}{2}\left[cos(2\pi(5)t_k) - cos(2\pi(101)t_k)\right]$$

High frequency terms are removed from both signals by low pass filtering - forming the in-phase (I) and quadrature (Q) components of the complex 5 Hz signal ($x_{d,k}$). The oscillators for the sine and cosine mixers in the DDC chain are driven by the RT microsecond timer in the myRIO. The LabVIEW code for the digital downconversion chain is shown in Figure 5.8.

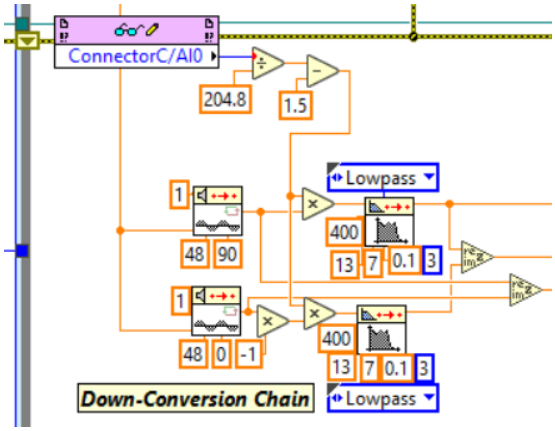

Figure 5.8: LabVIEW block diagram for the DDC chain.

The frequency of the 5 Hz down-converted signal is multiplied up to 20 Hz. The frequency multiplier operates in the same manner as the nominal design implementation, and the x4 frequency

multiplied signal is a Boolean type that clocks the baseband steering system DDS. The baseband DDS also employs a *True/False* structure that allows phase to accumulate on *True* messages from the frequency multiplier output. The LabVIEW code for the DDS, given a *True* input, is provided in Figure 5.9.

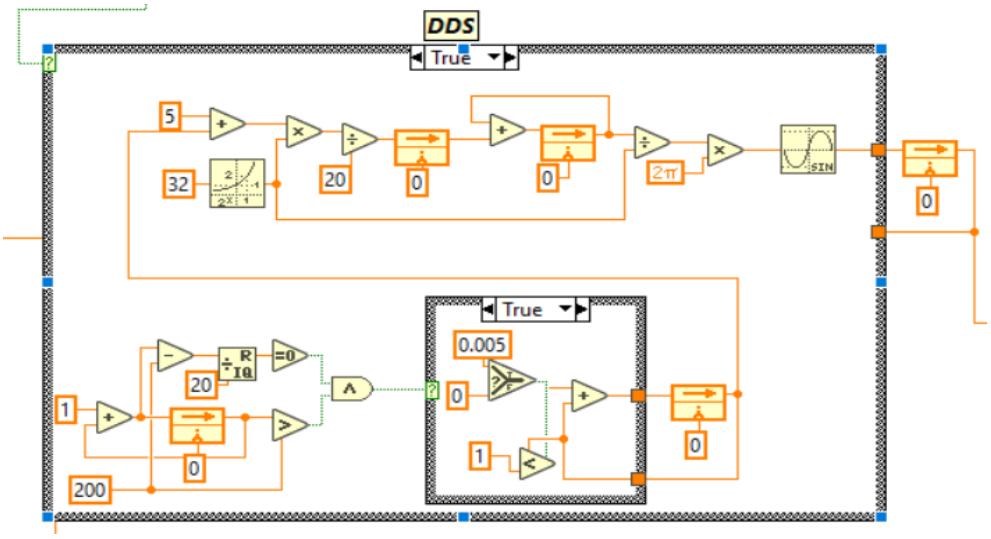

Figure 5.9: LabVIEW block diagram for the direct digital synthesis algorithm in the baseband steering system, given a *True* message input from the frequency multiplier.

Another difference between the nominal and baseband design implementations is in the smoothing PLL VCO. The free-running frequency of the VCO in this implementation is 5 Hz. Since the phase of the VCO output is offset by 90^o, when phase-lock is achieved, this becomes the 90^o phase shifted Q component for the upconversion chain. To generate the I component of the baseband steered signal, the phase error signal is input to a second oscillator with a constant phase offset of 90^o. The I and Q components of the steered and smoothed signal are input to the upconversion chain, but only the Q component is fed back to complete the control loop. The baseband smoothing PLL and single-sideband upconverter LabVIEW algorithm is shown in Figure 5.10.

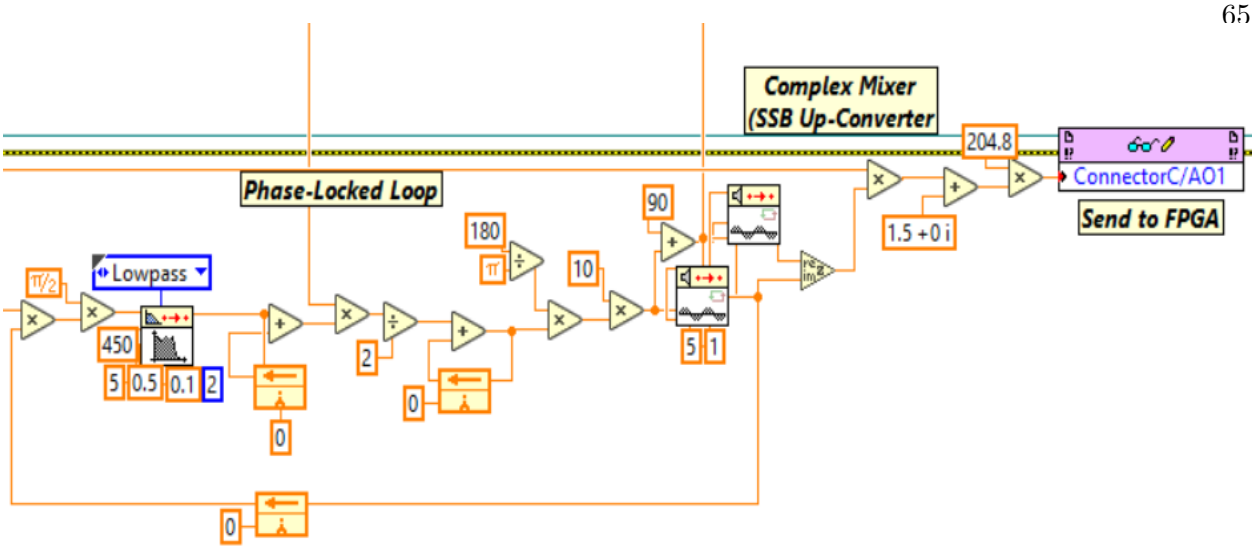

Figure 5.10: LabVIEW block diagram for the phase-locked loop and single-sideband up-converter in the baseband steering system algorithm. Filter and gain values were tuned by spectral analysis of the PLL output signal.

At the end of the single-sideband upconverter in Figure 5.10, the real voltage value of the complex upconverted signal is offset by +1.5 Volts. After the voltage offset and resolution scaling are applied, the sample value is handed back to the FPGA.

5.3 Steering Experiment Results

Performance analyses of both frequency accuracy and stability were conducted to compare the steering methods. Frequency accuracy was assessed to analyze offset performance and sideband noise using spectral analysis, as well as inspecting the quality of the signal formed by beating the input signals against the steered outputs. Frequency stability was analyzed by measuring the phase-time deviations of the inputs and outputs, and observing OADEV results to compare short-term and long-term performance between the two steering methods. Multiple data sets were collected for each steering system to verify repeatability of experimental results, and data were collected for over 3000 seconds in each experimental trial.

5.3.1 Short-Term Performance Analysis

Spectral analysis by Fourier Transforms is one of the most commonly used tools for assessing short-term performance. This measurement technique was used to quantitatively compare the output performance of both steering systems - by inspection of the amount of power present across the frequency spectrum of interest. Spectral power in the input signals and steered output signals was computed over 150,000 sample points - rendering a resolution bandwidth of 0.0027 Hz. Spectral results from the nominal steering experiment are shown in Figure 5.11.

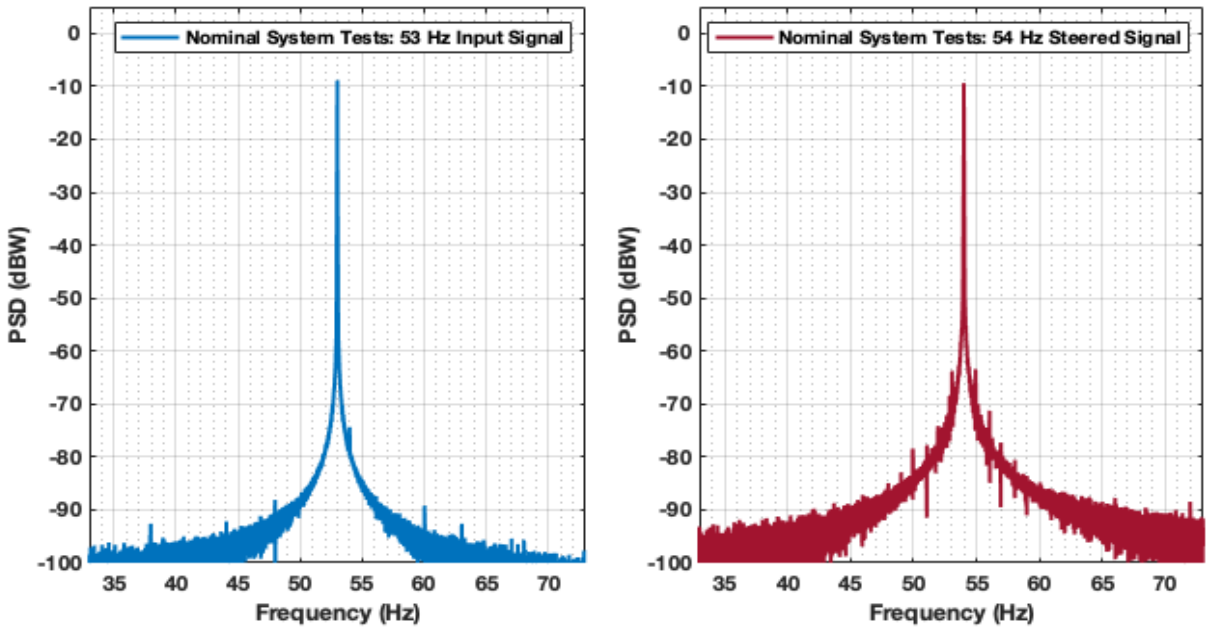

Figure 5.11: Spectral analysis results show clean spectra for the input and steered output signals, with power focused in the desired frequency bands.

The spectral analysis results in the plots above show highly satisfactory frequency spectra power for both signals. Sideband noise is below 60 dBW and power is focused at exactly 53 Hz and 54 Hz for the input and output signals, respectively. This is a positive indication that the high-rate steering system exhibits desirable short-term performance and the frequency of the input signal is accurately steered. The spectral analysis results for the baseband steering experiments are provided in Figure 5.12.

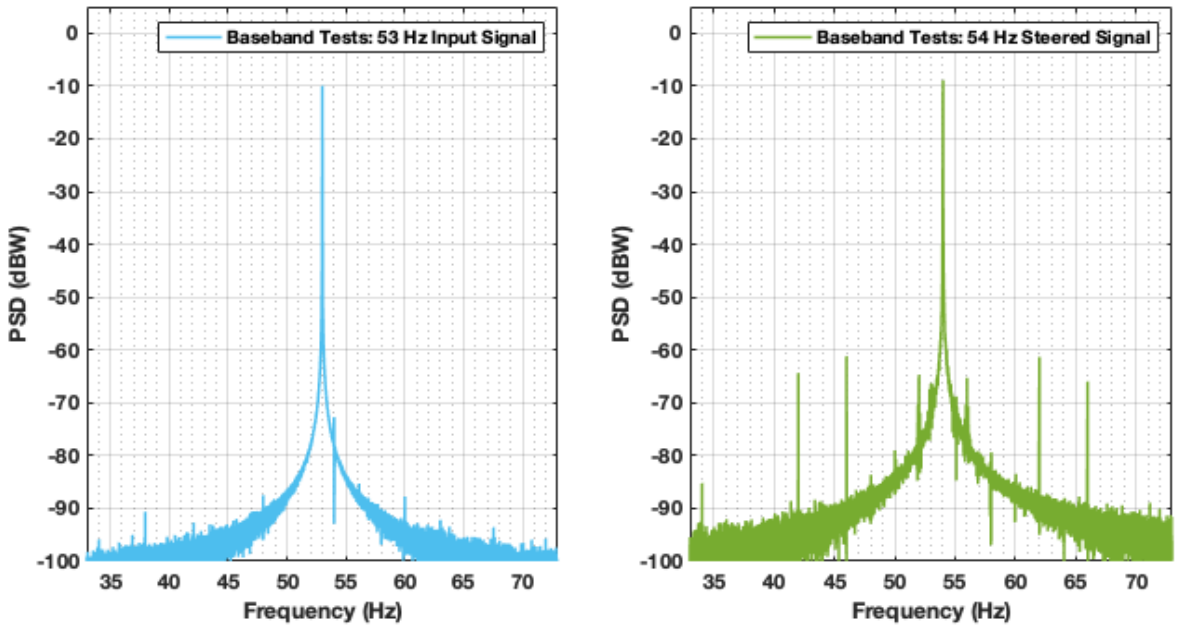

Figure 5.12: Spectral analysis again shows clean spectra for the input and steered output signals from the baseband steering experiments, with power focused in the desired frequency bands.

Spectral analysis of the baseband steering experiment data also shows satisfactory results. Again, power in both signals is focused at 53 Hz and 54 Hz and sideband noise is below 60 dBW - as observed in the high-rate steering results. These results are an indicate that short-term performance and applied offset accuracy between both steering methods is comparable. Inspection of the beat signals formed in both steering method experiments will be used to verify the superb short-term performance results observed from spectral analyses.

Beat signals were formed from each data set by mixing the steered output signals with the input signals, and low pass filtering the mixed signals to eliminate the high frequency terms. Mixing operations were performed during post-processing in MATLAB, using element by element multiplication of the sampled signal arrays - producing $sin(2\pi(1Hz)t_k) + sin(2\pi(107Hz)t_k)$. High frequency terms were removed by using a second order Butterworth low pass filter with a cutoff frequency of 10 Hz. The filtered output signal should evolve into a clean 1 Hz sine wave after the 1 Hz offset has been applied. Time series results for the beat signals, over the interval $t = 0\,s$ to $t = 210\,s$, are shown in Figure 5.13.

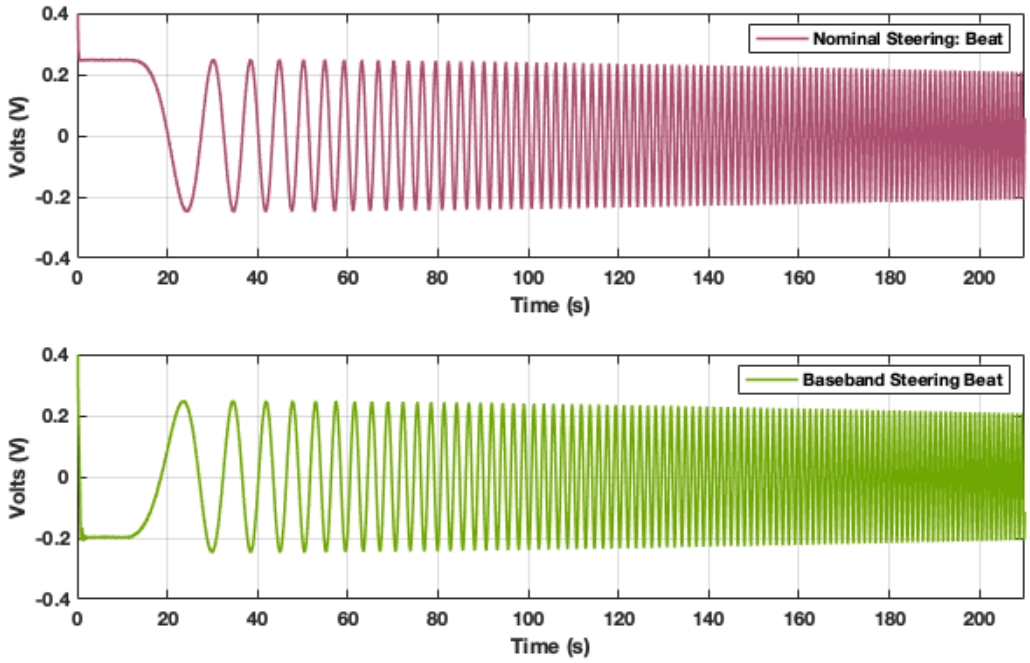

Figure 5.13: Both beat signals formed evolve into a clean 1 Hz sine wave.

The results from beat signal analyses over the period where the frequency offsets were applied show a desired smooth transition into a 1 Hz sine wave. Zoomed-in views of the signals, after the 1 Hz offset has been applied, are provided in Figure 5.14, and show clean sine waves that appear to have a periodicity of one second at this scale. These plots of the beat signals indicate that performance of both steering methods is comparable and confirm the spectral analysis results.

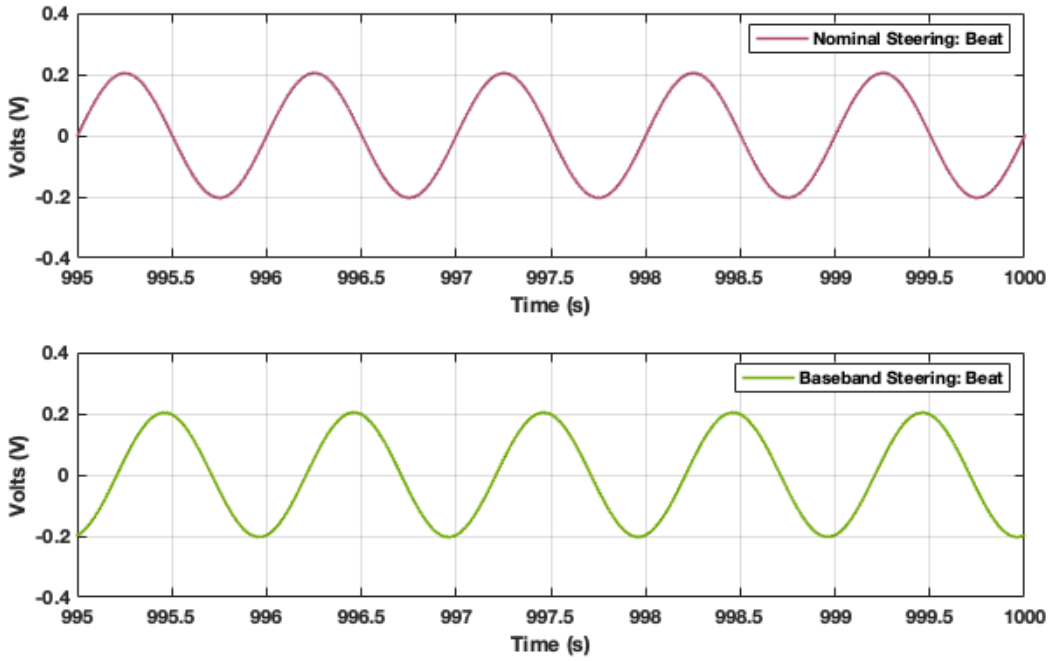

Figure 5.14: Both beat signals formed confirm frequency offset accuracy.

5.3.2 Frequency Stability Analysis

The frequency stability of the two steering methods will now be assessed and compared. Overlapping Allan Deviations were used as the performance metric and are computed from fractional frequency error over various averaging intervals - after the 1 Hz offsets were applied. In order to compute the OADEVs and examine frequency stability, the real sampled signals were converted into analytic form and phase measurements were computed by following the approach demonstrated by [51]. Transformation into complex form was computed by taking the Hilbert Transform of the signal arrays in MATLAB. I & Q component constellations from the transformed steered signal data is shown in Figure 5.15.

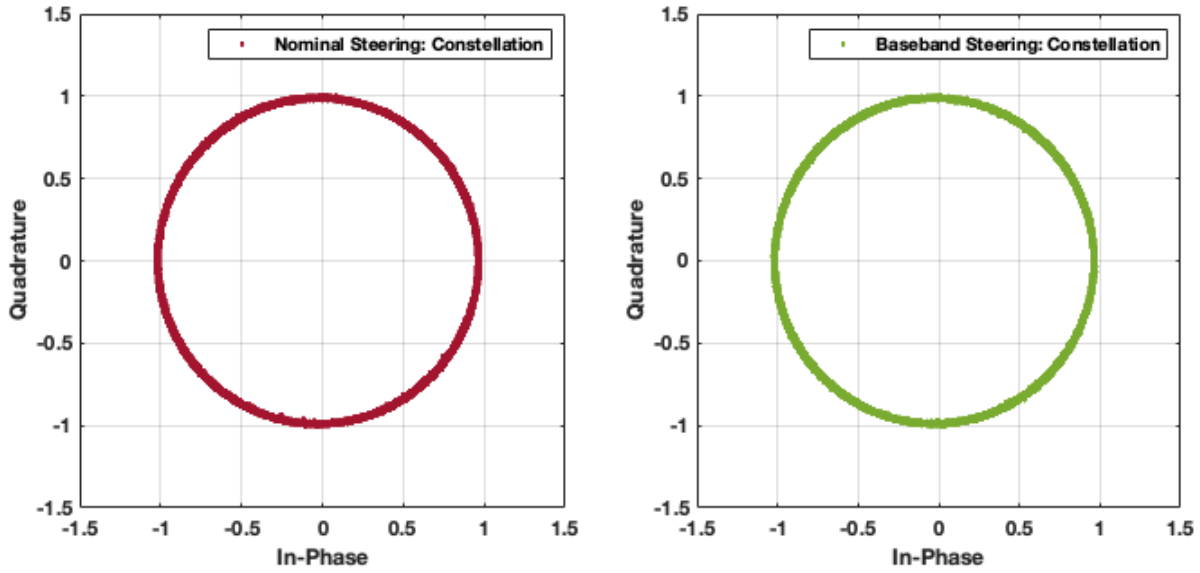

Figure 5.15: Both complex steered signal constellations lie on a circle and confirm adequate sample spacing.

The Hilbert Transform expects even time spacing between samples for accurate conversion to complex form, and can produce a small amount of outliers at the beginning and end of converted data sets. Outliers in present the first and last ten sample points were removed, and the constellation plots above confirm satisfactory conversion. Following complex signal conversion, unwrapped phase growth in the input and output signals was computed by

$$\phi_{t_k} = unwrap\left(tan^{-1}\left(\frac{imag(z_{t_k})}{real(z_{t_k})}\right)\right) \ \{rad\} \tag{5.4}$$

Where z_{t_k} is the complex form of each sampled signal array, and phase error growth ($\Delta\phi_{t_k}$) was computed by

$$\Delta\phi_{t_k} = \phi_{t_k} - \phi_{Ideal,t_k} \ \{rad\} \tag{5.5}$$

where ϕ_{Ideal,t_k} is ideal phase growth. For computing phase error of the input signals, $\phi_{Ideal,t_k} = 2\pi(53Hz)t_k$ and $\phi_{Ideal,t_k} = 2\pi(54Hz)t_k$ for steered output phase error calculations. The phase error growths computed for the input and steered output signals from both experiments are provided in Figure 5.16.

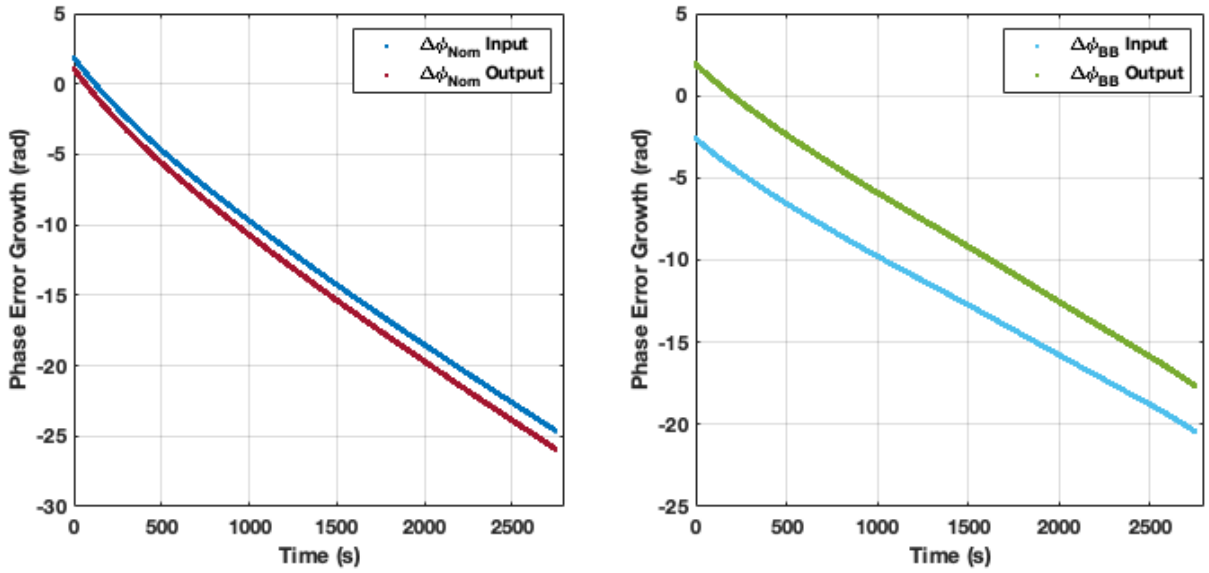

Figure 5.16: Phase error growth calculated from both steering experiments.

Phase error growths of each signal were converted into time errors by

$$\Delta T_{t_k} = \frac{\Delta\phi_{t_k}}{2\pi f_n} \{s\} \tag{5.6}$$

Here, ΔT_{t_k} is the calculated normalized phase-time deviation. Since the signals were sampled by the Arduino Uno microcontroller, the effect of the Arduino sample clock also needed to be removed. The normalized phase-time deviations of the input and steered output signals, after detrending the effect of the Arduino, are shown in Figure 5.17. This effect was detrended from the computed phase-time deviations by calculating a linear least-squares fit for the input signals and removing the estimated slopes from the input signals and output signals.

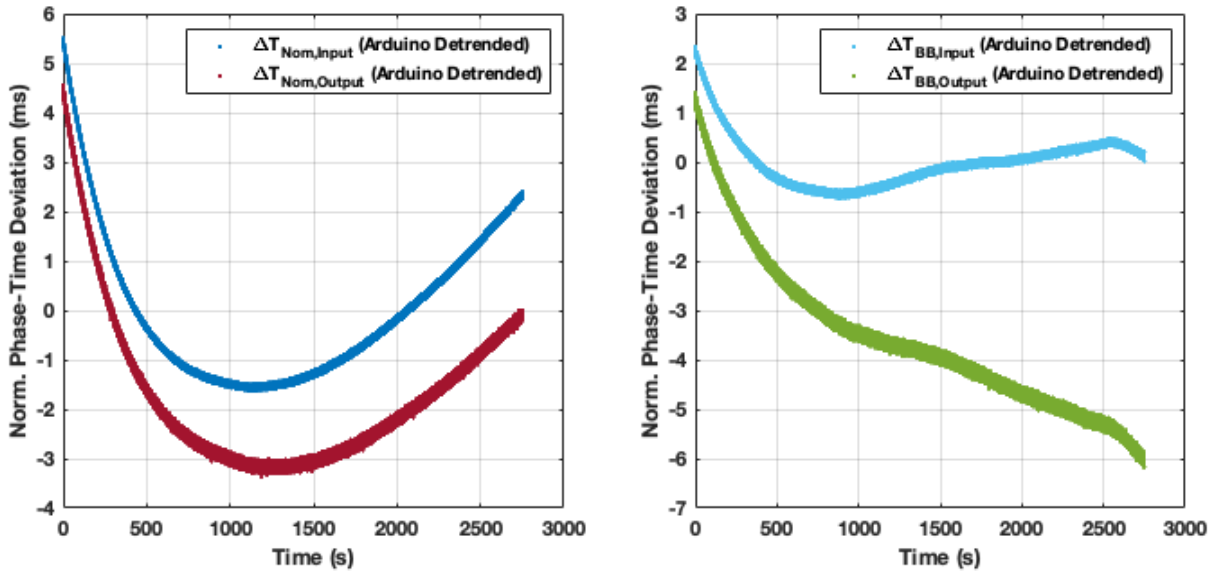

Figure 5.17: Phase-time deviations from both steering experiments. Signals were offset by ± 1 ms for plotting purposes, such that the data points did not lay on top of each other over early time intervals.

The normalized phase-time deviations show that there are multiple types of long-term noise present in the input signals. The types of noise observable in the phase of the input signals are white frequency, flicker phase, and random walk phase noise. These noise types will show up as different slopes in the Overlapping Allan Deviations. The phase-time deviations for the steered output signals show that both steered output phases grow at a slower rate than ideal 54 Hz signals. The scaling of the y-axis units reflect the order of magnitude for a 54 Hz signal. However, with respect to a 10 MHz signal, the phase-time deviations are reduced by a factor of $\approx 5 \times 10^{-6} s$ and the phase errors are on the nanosecond level.

Using the phase-time deviations computed for the input signals and steered output signals, OADEVs were computed. The OADEV values were calculated for multiple experimental trials over averaging intervals of $\tau = 1s$ to $\tau = 1000s$. The frequency stability results calculated are shown in Figure 5.18.

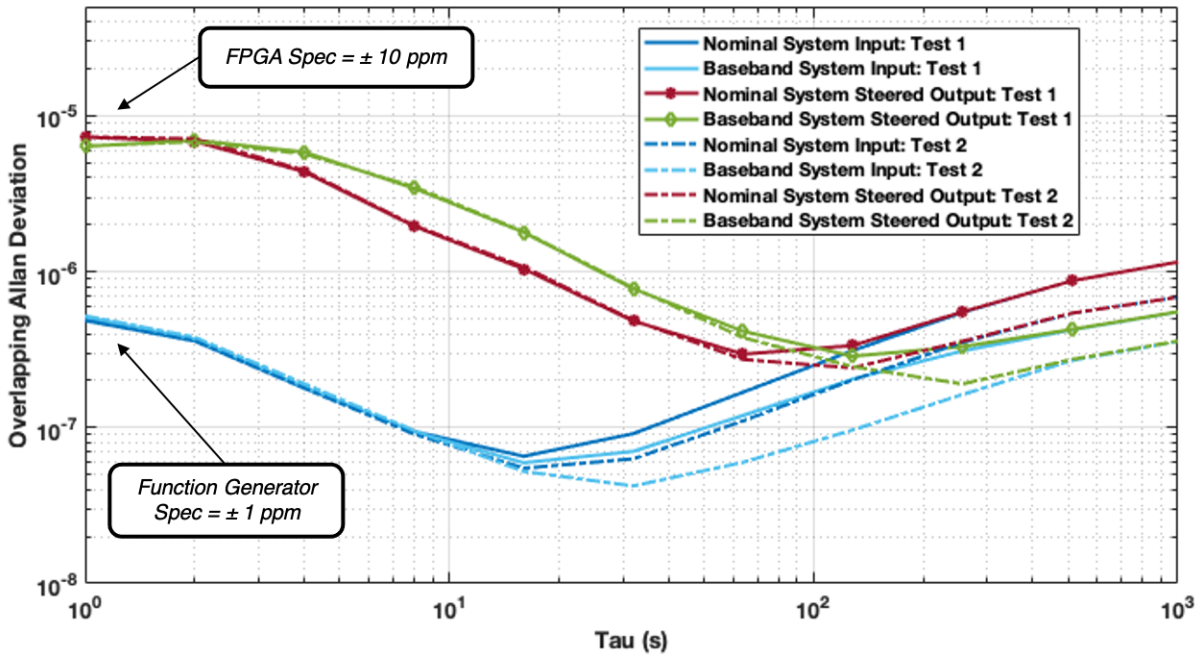

Figure 5.18: OADEV computed for the input signals and steered output signals over two experimental trials.

We see that the input signals show good agreement with the rated frequency stability of the function generator clock ($\pm 1\,ppm$), and the output signals also show good agreement with the specified clock stability values for the Xilinx Zync-7010 FPGA in the myRIO ($\pm 10\,ppm$). This indicates that when steering with a digital platform without an external reference, short-term frequency stability is limited to the stability of the hardware system clock. Secondly, the results show that there is no significant degradation in long-term frequency stability performance between the nominal steering method and the baseband steering method. Even though steered output signals show worse short-term performance than the input signals over the intervals $\tau < 200\,s$, the steered outputs begin to follow the input signals at longer-time intervals.

Another means for assessing steering performance using OADEV analysis is to analyze the stability of the measurement between the input and steered output. When the stability of this measurement exhibits a slope of -1 over long averaging intervals, this indicates that steering has been successfully accomplished and the clocks are syntonized [48] [24]. The frequency stability of these measurements for both tests are shown below in Figure 5.19.

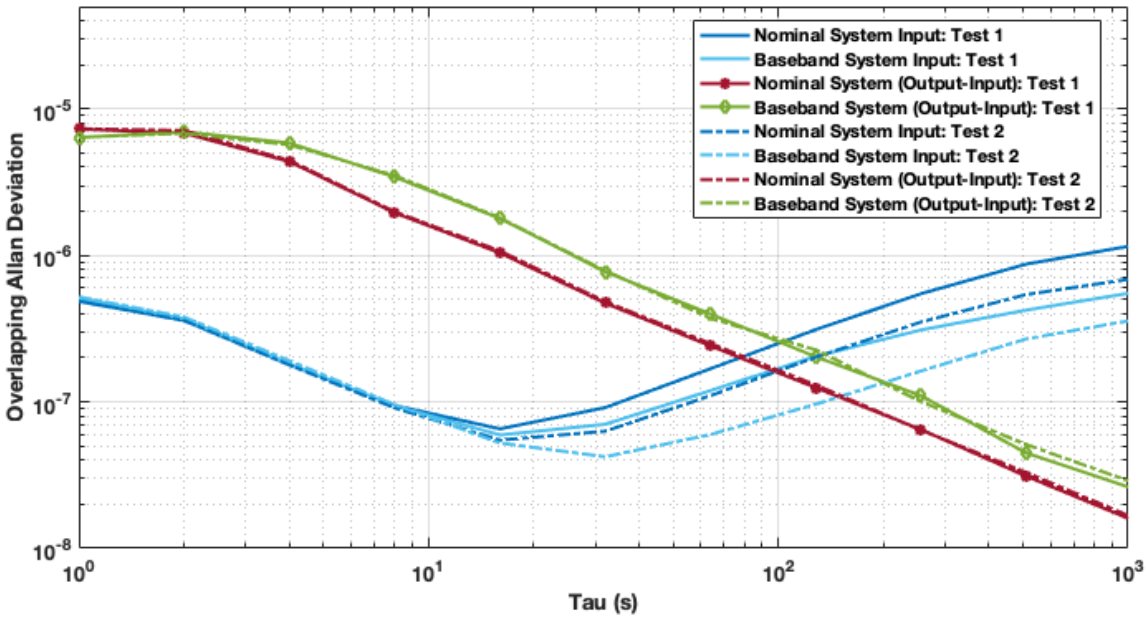

Figure 5.19: OADEV computed for the experimental measurements between the inputs and steered outputs.

Observable in Figure 5.19, a slope of -1 is visible over averaging intervals $\tau \geq 3s$. This confirms that steering was accomplished by both steering systems and again shows comparable performance between the two methods. This finding is positive news for the use of the baseband steering approach. The results from this analysis indicate that neither the larger relative fractional frequency adjustment size nor difference in frequency resolution significantly affected steering performance in these experiments.

Chapter 6

Summary & Future Work

6.1 Summary & Conclusions

This thesis provided an overview of the CONTACT research project and the role of the signal generation system in Chapter 1, as well as described a common method for analyzing clock performance. In Chapter 2, the nominal design of the signal generation system for the low-SWaP clock ensemble was summarized, and the design generation subsystem was described in detail. Chapter 3 provided an overview of the steering control subsystem, presented an H_∞ control approach for clock steering, and compared steering performance with a previously established LQG steering approach. In Chapter 4, the front-end of the hardware platform for the testbed was analyzed and modifications to the nominal steering system design for baseband steering were presented. Lastly, in Chapter 5, the hardware implementations of both steering approaches were described and experimental performance was analyzed.

There are four main conclusions drawn from the research efforts discussed in this thesis. First, performance analysis from the hardware experiments for the nominal and baseband design implementations demonstrated that the baseband steering method does not significantly degrade steering performance. Second, the experimental results showed that when steering is implemented with a digital platform, short-term frequency stability is limited to the performance of the on board hardware clock. Third, if the LO's for the N310 front-end low-band upconverter and downconverter mixers are synchronized by an external reference, the front-end chains should add minimal phase noise to the signal. The final conclusion drawn from this research is that using loop shaping and

H_∞ optimization for steering control can provide desirable frequency stability, and in certain cases, provide better stability than the free running behavior and the steering reference.

6.2 Future Work Recommendations

There are two primary recommendations for future work on the continued development of the signal generation system for the low-SWaP clock ensemble testbed. These recommendations are as follows.

The first is to replicate the hardware implementation experiments with RF signals using an Ettus N210, or preferably with an Ettus E210. The N210 and E210 do not use the same daughterboards or transceiver and allows the input signals to be directly sampled by the ADC and output by the DAC. The E210 is the preferable choice because it is an embedded device and signal processing algorithms can be directly deployed onto the SDR.

The second recommendation is to try to implement the H_∞ steering control design to steer a CSAC to a superior atomic frequency standard. The SA.45s CSAC has an on board microphase stepper that can be tuned by an analog or digital control signal. However, the use of digital steering commands should be avoided or done with caution. Digital steering commands are stored in the on board memory of a CSAC, so there are only a finite number of digital steering opportunities available. Since the CONTACT team has access to a Rubidium atomic frequency standard, this would be an excellent method to tune and validate simulated steering performance.

Bibliography

[1] H. Aliakbarian, V. Volski, E. van der Westhuizen, R. Wolhuter, and G. A. E. Vandenbosch. Analogue versus digital for baseband beam steerable array used for LEO satellite applications. In Proceedings of the Fourth European Conference on Antennas and Propagation, pages 1–4, 2010.

[2] D. W. Allan. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control, 34(6):647–654, 1987.

[3] Analog Devices. MT-086: Fundamentals of Phase Locked Loops (PLLs). 2009.

[4] Analog Devices. AD9371 Data Sheet: Integrated, Dual RF Transceiver with Observation Path, 2017. Rev. B.

[5] W. F. Arnold and A. J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. Proceedings of the IEEE, 72(12):1746–1754, 1984.

[6] D. Banerjee. PLL Performance, Simulation, and Design. Dog Ear Publishing, 4th edition, 2006.

[7] C. Barrett. Fractional/integer-n pll basics. Technical Report SWRA029, Texas Instruments, Dallas, TX, August 1999.

[8] K. R. Brown. The theory of the gps composite clock. In Proceedings of the 4th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GPS, pages 223–242, 1991.

[9] R. Brown and P. Hwang. Introduction to Random Signals and Applied Kalman Filtering. John Wiley & Sons, 3rd edition, 1997.

[10] P. H. Dana and B. M. Penrod. The role of gps in precise time and frequency dissemination. GPS World, 1990.

[11] A. De Lima Fernandes. Demystifying the pll. https://www.cypress.com/file/60101, April 2013.

[12] I. Diegel. Hydrogen maser frequency standard. In Proceedings of Seventh IVS Technical Operations Workshop, Westford, MA, 2013.

[13] DoD Chief Information Officer and Principal Staff Assistant for PNT Policy. Strategy for the department of defense positioning, navigation, and timing (PNT) enterprise [unclassified version]. Technical report, 2019.

[14] J. Doyle, B. Francis, and A. Tannenbaum. Feedback Control Theory. Macmillan Publishing Co., 1990.

[15] J. C. Doyle. Synthesis of robust controllers and filters. In The 22nd IEEE Conference on Decision and Control, pages 109–114, 1983.

[16] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. IEEE Transactions on Automatic Control, 34(8):831–847, 1989.

[17] W. Egan. Phase-Lock Basics. Wiley-Interscience, New York, NY, 1998.

[18] Ettus Research. N310 USRP Daughterboard Diagram. 2020.

[19] D. Frizelle and F. Kearny. Complex RF Mixers, Zero-IF Architecture, and Advanced Algorithms: The Black Magic in Next-Generation SDR Transceivers. Analog Devices, February 2017.

[20] L. Galleani, L. Sacerdote, P. Tavella, and C. Zucca. A mathematical model for the atomic clock error. Metrologia, 40(3):S257–S264, June 2003.

[21] K. Gentile and R. Cushing. A Technical Tutorial on Digital Signal Synthesis. Analog Devices, 1999.

[22] K. Glover and D. McFarlane. Robust stabilization of normalized coprime factor plant descriptions with h_∞ bounded uncertainty. IEEE Transactions on Automatic Control, 34(8):821–830, 1989.

[23] M. Godel and J. Furthner. Robust ensemble time onboard a satellite. In Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting, ION PTTI, pages 23–46, 2017.

[24] M. Godel, T.D. Schmidt, and J. Furthner. Comparison between simulation and hardware realization for different clock steering techniques. Metrologia, 56(035001):1–11, 2019.

[25] C. A. Greenhall. A kalman filter clock ensemble algorithm that admits measurement noise. Metrologia, 43(4):311–321, 2006.

[26] M. J. Grimble and D. Fragopoulos. Solution of discrete H_∞ optimal control problems using a state-space approach. In 29th IEEE Conference on Decision and Control, pages 1775–1780 vol.3, 1990.

[27] Y. Haibo and G. Wei. Frequency steering and the control of utc(k). In 2012 IEEE International Frequency Control Symposium Proceedings, pages 1–4, 2012.

[28] B. Han and X. Lin. Adapt the steady-state kalman gain using the normalized autocorrelation of innovations. IEEE Signal Processing Letters, 12(11):780–783, 2005.

[29] P. Koppang and R. Leland. Linear quadradic stochastic control of atomic hydrogen masers. IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control, 46(3):517–522, 1999.

[30] E. Kreyszig. Advanced Engineering Mathematics. John Wiley Sons, Inc., Hoboken, NJ, 10th edition, 2011.

[31] J. Levine. Steering a time scale. In Proceedings of the 40th Annual Precise Time and Time Interval (PTTI) Meeting, ION PTTI, pages 205–218, 2008.

[32] L. L. Lewis. An introduction to frequency standards. In Proceedings of the IEEE, volume 79, pages 927–935, July 1991.

[33] MathWorks. MATLAB Documentation. 2020.

[34] D. Matsakis. The effects of proportional steering strategies on the behavior of controlled clocks. Metrologia, 56(025007):1–11, 2019.

[35] Microsemi. 5071A Primary Frequency Standard Datasheet. 2019.

[36] Microsemi. MX-503 High Precision Microprocessor Controlled Crystal Oscillator Datasheet. 2018.

[37] Microsemi. SA.45s CSAC Datasheet. 2019.

[38] Microsemi. Microsemi MHM 2010 Active Hydrogen Maser Datasheet, 2016. Rev. D.

[39] K. Mishagin, V. Lysenko, and S. Yur'evich Medvedev. A practical approach to optimal control problem for atomic clocks. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 67(5):1080–1087, 2020.

[40] P. Misra and P. Enge. Global Positioning System: Signals, Measurements, and Performance. Ganjan-Jamuna Press, Lincoln, MA, 2nd edition, 2012.

[41] Morion, Inc. Double Oven Low Profile Ultra Precision OCXO MV180, 2015. Revision 2.

[42] National Instruments. LabVIEW Documentation. 2020.

[43] National Instruments. NI myRIO-1900 User Guide and Specifications. 2016.

[44] T. Poulter Jr. A practical time-shared computer system. Hewlett-Packard Journal, 7, July 1968.

[45] P. Rawicz. H_∞/H_2/Kalman Filtering of Linear Dynamical Systems via Variational Techniques with Applications to Target Tracking. PhD thesis, Drexel University, Philadelphia, PA, 2000.

[46] W. J. Riley. Techniques for frequency stability analysis. In Proceedings of the IEEE International Frequency Control Symposium, Tampa, FL, 2003. Symmetricom.

[47] W.J. Riley. Handbook of Frequency Stability Analysis. National Institute of Standards and Technology, Time and Frequency Division, 2008.

[48] T. D. Schmidt, C. Trainotti, and J. Furthner. Comparing clock steering technique performances in simulations and measurements. In Proceedings of the 50th Annual Precise Time and Time Interval Systems and Applications Meeting, ION PTTI, pages 290–298, 2019.

[49] J. A. Sefton, R. Ober, and K. Glover. Robust stabilization in the presence of coprime factor perturbations. In 29th IEEE Conference on Decision and Control, volume 3, pages 1197–1198, 1990.

[50] K. L. Senior and M. J. Coleman. The next generation GPS time. Journal of The Institute of Navigation, 0(0), 2017.

[51] J. Sherman and R. Jordens. Oscillator metrology with software defined radio. Review of Scientific Instruments, 87(054711):1–11, 2016.

[52] S. Shrimali. Direct digital frequency synthesizer. M.S. Thesis, Texas Tech University, Lubbock, TX, 2007.

[53] D. Simon. Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches. John Wiley & Sons, 2006.

[54] S. Skogestad and I. Postlethwaite. Multivariable Feedback Control Analysis and Design. John Wiley & Sons, Inc., Hoboken, NJ, 2nd edition, 2001.

[55] P. Tavella. Statistical and mathematical tools for atomic clocks. Metrologia, 45:s183–s192, 2008.

[56] C. Trainotti, T. D. Schmidt, and J. Furthner. Comparison of clock models in view of clock composition, clock steering, and measurement fitting. In Proceedings of the 50th Annual Precise Time and Time Interval Systems and Applications Meeting, ION PTTI, pages 265–283, 2019.

[57] C. Trainotti, T. D. Schmidt, and J. Furthner. Simulating the realization of a mixed clock ensemble. In 2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC), pages 1–11, 2019.

[58] S. Veres and H. Hjalmarsson. Tuning for robustness and performance using iterative feedback tuning. In Proceedings of the 41st IEEE Conference on Decision and Control, pages 4682–4687, 2002.

[59] Y. Wu, H. Gong, X. Zhu, and G. Ou. A dpll method applied to clock steering. IEEE Transactions on Instrumentation and Measurement, 65(6):1331–1342, 2016.

[60] C. Zucca and P. Tavella. The clock model and its relationship with the allan and related variances. IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control, 56:289–296, 2005.

Appendix A

Additional Steering Simulation Results

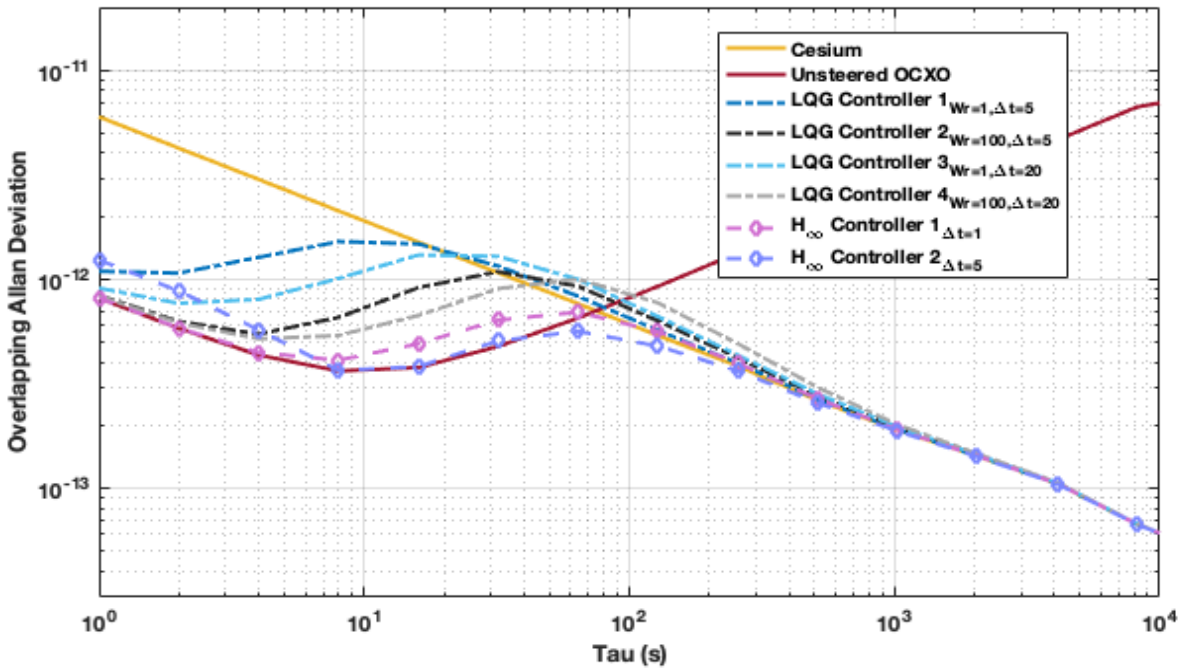

Figure A.1: Frequency stability results for all controllers from the nominal steering scenario.

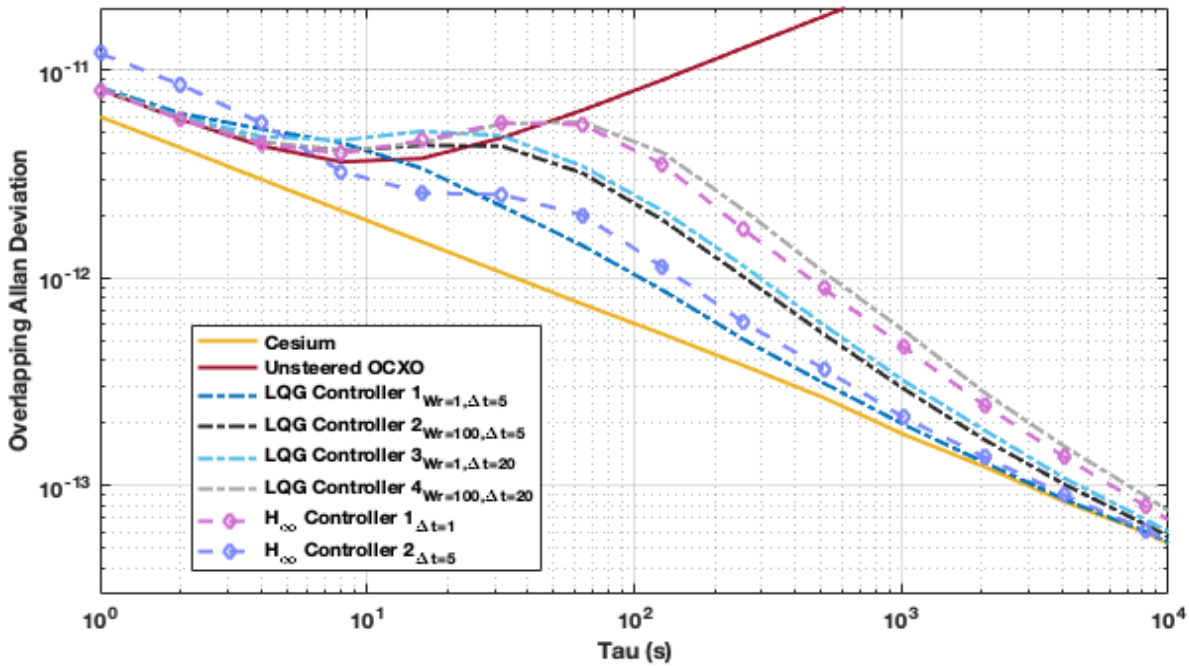

Figure A.2: Frequency stability results for all controllers, with process noise variance increased in the truth data by a factor of 10.

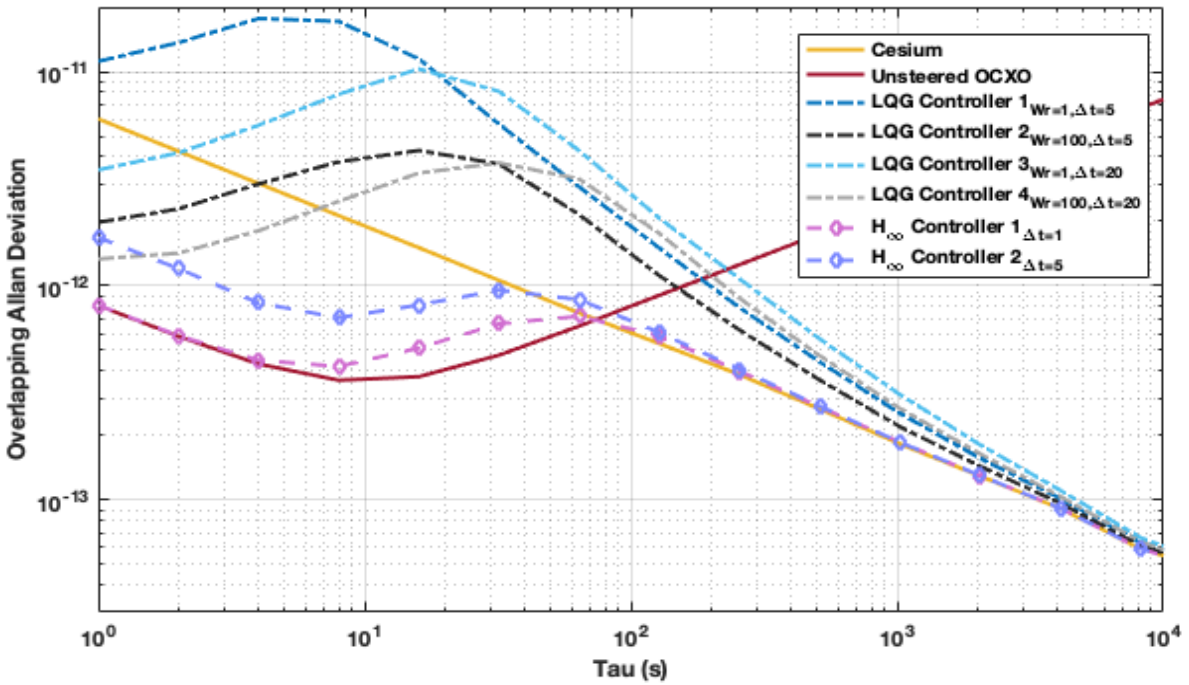

Figure A.3: Frequency stability results for all controllers, with measurement noise variance increased in the truth data by a factor of 100.