



8-2017

Modeling and Design of a Low-Level RF Control System for the Accumulator Ring at Spallation Neutron Source

Michael G. Trout

University of Tennessee, Knoxville, mtrout1@vols.utk.edu

Follow this and additional works at: https://trace.tennessee.edu/utk_gradthes



Part of the [Controls and Control Theory Commons](#)

Recommended Citation

Trout, Michael G., "Modeling and Design of a Low-Level RF Control System for the Accumulator Ring at Spallation Neutron Source. " Master's Thesis, University of Tennessee, 2017.
https://trace.tennessee.edu/utk_gradthes/4907

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a thesis written by Michael G. Trout entitled "Modeling and Design of a Low-Level RF Control System for the Accumulator Ring at Spallation Neutron Source." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Donatello Materassi, Major Professor

We have read this thesis and recommend its acceptance:

Sarah M. Cousineau, Seddik M. Djouadi, Garrett S. Rose

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Modeling and Design of a Low-Level RF Control System for the Accumulator Ring at Spallation Neutron Source

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Michael G. Trout

August 2017

Acknowledgements

ORNL is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy. This material is based upon work supported by the U.S. Department of Energy, Office of Science under contract number DE-AC05-00OR22725.

Notice: This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains, and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

My sincerest thanks go out to my supervisor, colleague, and friend, Eric Breeding. The depth of his contributions to this thesis and my life in general cannot be overstated, and he will forever stand as one of the people that shaped me into the engineer and man that I am. Both Eric and my manager Doug Curry have been so quick to offer advice and strategy during this season, and if it weren't for their urgings, this document would not exist.

I would like to thank Oak Ridge National Laboratory and Spallation Neutron Source; it has been my deepest pleasure to be a part of your organization, and I greatly appreciate the opportunities ORNL and SNS have afforded me. Special thanks go to Alan Justice, Rob

Knudson, Steve Hicks, Stan Baker, and many others who enriched my experience at SNS - you often put a smile on my face.

I would like to thank the students and faculty of the Electrical Engineering and Computer Science Department at the University of Tennessee in Knoxville for their influence on my academic experience. I'm especially grateful for the influence people like Dr. Ben Blaylock, Dr. David Icove, Dr. Donatello Materassi, Dr. Nicole McFarlane, Dr. Aly Fathy, Dr. Ethan Farquahar, Dr. Chao Tian, Dr. Hairong Qi, and others have had on me. I was endlessly entertained and enlightened by your instruction, and many semesters were brightened by the thought that I had one of your classes that day.

I would also like to thank the members of my graduate committee: Dr. Sarah Cousineau, Dr. Seddik Djouati, and Dr. Garrett Rose. You each contributed to this process in a special way and I was honored to have you critique my work.

Special thanks to Uta Hartel from Technische Universitat Darmstadt for providing the MATLAB model used in "Modeling of the RF Control System of a Synchrotron Accelerating Unit". The early insights your model provided were invaluable.

Special thanks also go to John Sinclair, Chip Piller, Mark Crofford, and Franklin Frye for their technical support on this project, as well as to Karen White and Mark Champion for their approval and support.

Lastly, my heartfelt thanks go to my friends, my parents, and my God. Their encouragement, support, and enabling have been the fuel to this fire all along, and genuinely the only reason I continue to this day.

Abstract

Since its commissioning in 2006, Spallation Neutron Source (SNS) at Oak Ridge National Laboratory has greatly contributed to the field of neutron science, but some critical systems are reaching end-of-life. This obsolescence must be addressed for the accelerator to continue providing world-class research capabilities. One such system needing redesign is the low-level RF (LLRF) control system for the proton accumulator ring. While this system has performed acceptably for over a decade, it is sparsely documented and robust operational models are unavailable. To ensure the new design meets or exceeds current performance metrics, we analyzed the existing LLRF control system and designed a system-accurate controller model. This model included a state-space representation of the RF accelerator cavity dynamics. Both the controller and cavity models are combined to provide complete, functional simulation capabilities for the SNS accumulator ring LLRF control system. We then realized the modeled controller in an FPGA using VHDL cores which were subsequently used to successfully regulate the accumulator ring. The designed controller was functional at repetition rates up to 160 Hz while system specifications only require 60 Hz operation. The designed controller achieved 1 MW beam-on-target operation at 60 Hz repetition rate and a fundamental frequency of approximately 1 MHz.

Table of Contents

Chapter 1: Introduction	1
Chapter 2: Theory	7
2.1 Physical	7
2.2 System Timing	11
2.3 Control Actions of Signal of Interest	13
Chapter 3: Design	23
3.1 Cavity Modeling	23
3.2 Controller Model Design	27
3.3 Software Model Design	34
Chapter 4: Results	54
4.1 Introduction	54
4.2 Floating Point Model Results	55
4.3 Fixed Point Model Results	60
4.4 Floating Point and Fixed Point Comparison	68
4.5 Fixed Point Model with Delay Results	68
4.6 System Generator Model Results	72
4.7 Fixed Point with Delay and System Generator Comparison	72
Chapter 5: Physical System Testing	86
Chapter 6: Conclusion	89

Bibliography	91
Appendices	93
Appendix A	94
Appendix B	97
Appendix C	98
Appendix D	99
Vita	114

List of Tables

Table 3.1 List of PID_Main Inputs

List of Figures

Figure 1.1.1	Spallation Neutron Source Site Layout	2
Figure 1.1.2	Functional Block Diagram of Original Controller	4
Figure 2.1.1	Coulomb Force on a Charged Particle	8
Figure 2.1.2	Plot of What Field Early, Synchronous, and Late Particles Experience	8
Figure 2.1.3	Equivalent Circuit Model of an RF Cavity [9]	9
Figure 2.2.1	Breakdown of the SNS Beam Pulse Structure [8]	12
Figure 2.3.1	Accumulator Ring LLRF Controller Block Diagram	14
Figure 2.3.2	Traditional Feedback Controller Loop	15
Figure 2.3.3	IQ Modulation-Demodulation Linear Reciprocity	17
Figure 2.3.4	Discrete Time PID Controller Topology [9] [11]	18
Figure 2.3.5	Similarities and Differences between I and Q Paths	19
Figure 2.3.6	Block Diagram of Phase Rotator with Equations	21
Figure 3.1.1	Cavity Circuit Model [9]	24
Figure 3.2.1	IQ Demodulator Block Diagram	28
Figure 3.2.2	IQ Demodulator Signal Demonstration	29
Figure 3.2.3	PID Controller Diagram [9] [11]	30
Figure 3.2.4	Phase Rotator Diagram	32
Figure 3.2.5	IQ Modulator Diagram	33
Figure 3.3.1	Overview of Floating Point Model	37
Figure 3.3.2	Floating Point Model IQ Demodulator Subsystem	38
Figure 3.3.3	Floating Point Model Non-IQ Sampling Accumulator	39

Figure 3.3.4	Floating Point Model PID Controller Subsystem	40
Figure 3.3.5	Floating Point Model IQ Modulator	42
Figure 3.3.6	PID Tuning Script Flowchart	44
Figure 3.3.7	Example of PID Tuning Script Granularity Tuning	45
Figure 3.3.8	Fixed Point Notation Format Example	47
Figure 3.5.1	Overview of System Generator Model	49
Figure 3.5.2	SysGen Model VHDL Black Box Exploded View	51
Figure 4.2.1	Floating Point Model Cavity, IQ Error, and IQ Drive without Noise in Open-Loop	56
Figure 4.2.2	Floating Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop, $K_p = 2$ $K_i = 0.4$, $K_d = 0$	56
Figure 4.2.3	Floating Point Model Cavity, IQ Error, and IQ Drive with Noise in Open Loop	58
Figure 4.2.4	Floating Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop, $K_p = 2$ $K_i = 0.4$, $K_d = 0$	58
Figure 4.2.5	Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$	59
Figure 4.2.6	Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$	59
Figure 4.2.7	Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$	59
Figure 4.2.8	Floating Point Model at 952.2 KHz, Cavity and IQ Drive, Open-Loop	61

Figure 4.2.9	Floating Point Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop	61
Figure 4.2.10	Floating Point Model at 1163.8 KHz, Cavity and IQ Drive, Open-Loop	61
Figure 4.2.11	Floating Point Model at 952.2 KHz, Cavity and IQ Drive, Closed-Loop	62
Figure 4.2.12	Floating Point Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop	62
Figure 4.2.13	Floating Point Model at 1163.8 KHz, Cavity and IQ Drive, Closed-Loop	62
Figure 4.3.1	Fixed Point Model Cavity, IQ Error, and IQ Drive without Noise in Open Loop	63
Figure 4.3.2	Fixed Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop, $K_p = 2$ $K_i = 0.4$, $K_d = 0$	63
Figure 4.3.3	Fixed Point Model Cavity, IQ Error, and IQ Drive with Noise in Open Loop	64
Figure 4.3.4	Fixed Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop, $K_p = 2$ $K_i = 0.4$, $K_d = 0$	64
Figure 4.3.5	Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$	65
Figure 4.3.6	Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$	65
Figure 4.3.7	Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$	65
Figure 4.3.8	Fixed Point Model at 952.2 KHz, Cavity and IQ Drive, Open-Loop	66
Figure 4.3.9	Fixed Point Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop	66
Figure 4.3.10	Fixed Point Model at 1163.8 KHz, Cavity and IQ Drive, Open-Loop	66
Figure 4.3.11	Fixed Point Model at 952.2 KHz, Cavity and IQ Drive, Closed-Loop	67
Figure 4.3.12	Fixed Point Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop	67
Figure 4.3.13	Fixed Point Model at 1163.8 KHz, Cavity and IQ Drive, Closed-Loop	67

Figure 4.4.1	Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$	69
Figure 4.4.2	Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$	69
Figure 4.4.3	Percent Error of Fixed Point Model Results from Floating Point Model Results, 1	69
Figure 4.4.4	Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$	70
Figure 4.4.5	Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$	70
Figure 4.4.6	Percent Error of Fixed Point Model Results from Floating Point Model Results, 2	70
Figure 4.4.7	Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$	71
Figure 4.4.8	Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$	71
Figure 4.4.9	Percent Error of Fixed Point Model Results from Floating Point Model Results, 3	71
Figure 4.5.1	Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop	73
Figure 4.5.2	Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 1.5$, $K_i = 0.05$, $K_d = 0$	73

Figure 4.5.3	Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop	74
Figure 4.5.4	Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 1.5$, $K_i = 0.05$, $K_d = 0$	74
Figure 4.5.5	Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$	75
Figure 4.5.6	Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$	75
Figure 4.5.7	Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$	75
Figure 4.5.8	Fixed Point Delay Model at Low Frequency, Cavity and IQ Drive, Open Loop	76
Figure 4.5.9	Fixed Point Delay Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop	76
Figure 4.5.10	Fixed Point Delay Model at High Frequency, Cavity and IQ Drive, Open Loop	76
Figure 4.5.11	Fixed Point Delay Model at Low Frequency, Cavity and IQ Drive, Closed Loop	77
Figure 4.5.12	Fixed Point Delay Model at 1.058 MHz, Cavity and IQ Drive, Closed Loop	77
Figure 4.5.13	Fixed Point Delay Model at High Frequency, Cavity and IQ Drive, Closed-Loop	77

Figure 4.6.1	System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop	78
Figure 4.6.2	System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.125$, $K_d = 2.5$	78
Figure 4.6.3	System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop	79
Figure 4.6.4	System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.125$, $K_d = 2.5$	79
Figure 4.6.5	System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$	80
Figure 4.6.6	System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$	80
Figure 4.6.7	System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$	80
Figure 4.6.8	System Generator Model at Low Frequency, Cavity and IQ Drive, Open Loop	81
Figure 4.6.9	System Generator Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop	81
Figure 4.6.10	System Generator Model at High Frequency, Cavity and IQ Drive, Open Loop	81
Figure 4.6.11	System Generator Model at Low Frequency, Cavity and IQ Drive, Closed Loop	82

Figure 4.6.12	System Generator Model at 1.058 MHz, Cavity and IQ Drive, Closed Loop	82
Figure 4.6.13	System Generator Model at High Frequency, Cavity and IQ Drive, Closed-Loop	82
Figure 4.7.1	Fixed Point Delay Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.125$ $K_d = 0$	83
Figure 4.7.2	System Generator Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.125$ $K_d = 0$	83
Figure 4.7.3	Percent Error of Fixed Point Delay Model Results from System Generator Model Results, 1	83
Figure 4.7.4	Fixed Point Delay Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$	84
Figure 4.7.5	System Generator Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$	84
Figure 4.7.6	Percent Error of Fixed Point Model Results from System Generator Model Results, 2	84
Figure 4.7.7	Fixed Point Delay Model Cavity and IQ Error, Open-Loop	85
Figure 4.7.8	System Generator Cavity and IQ Error, Open-Loop	85
Figure 4.7.9	Percent Error of Fixed Point Delay Model Results from System Generator Model Results, 3	85
Figure 5.1.1	SNS Accumulator Ring LLRF Control System Dynamic Tuning Test Status Screen	87

Figure 5.1.2	SNS Accumulator Ring LLRF Control System 1 Megawatt 60Hz Test	
	Status Screen	88
Figure D.1	Floating Point Rise Time, $K_d = 0$	100
Figure D.2	Floating Point Settling Time, $K_d = 0$	100
Figure D.3	Floating Point Overshoot, $K_d = 0$	100
Figure D.4	Floating Point Rise Time, $K_d = 0.1$	101
Figure D.5	Floating Point Settling Time, $K_d = 0.1$	101
Figure D.6	Floating Point Overshoot, $K_d = 0.1$	101
Figure D.7	Floating Point Rise Time, $K_d = 0.25$	102
Figure D.8	Floating Point Settling Time, $K_d = 0.25$	102
Figure D.9	Floating Point Overshoot, $K_d = 0.25$	102
Figure D.10	Fixed Point Rise Time, $K_d = 0$	103
Figure D.11	Fixed Point Settling Time, $K_d = 0$	103
Figure D.12	Fixed Point Overshoot, $K_d = 0$	103
Figure D.13	Fixed Point Rise Time, $K_d = 0.1$	104
Figure D.14	Fixed Point Settling Time, $K_d = 0.1$	104
Figure D.15	Fixed Point Overshoot, $K_d = 0.1$	104
Figure D.16	Fixed Point Rise Time, $K_d = 0.25$	105
Figure D.17	Fixed Point Settling Time, $K_d = 0.25$	105
Figure D.18	Fixed Point Overshoot, $K_d = 0.25$	105
Figure D.19	Fixed Point Delay Rise Time, $K_d = 0$	106
Figure D.20	Fixed Point Delay Settling Time, $K_d = 0$	106

Figure D.21	Fixed Point Delay Overshoot, $K_d = 0$	106
Figure D.22	Fixed Point Delay Rise Time, $K_d = 1.5$	107
Figure D.23	Fixed Point Delay Settling Time, $K_d = 1.5$	107
Figure D.24	Fixed Point Delay Overshoot, $K_d = 1.5$	107
Figure D.25	Fixed Point Delay Rise Time, $K_d = 3$	108
Figure D.26	Fixed Point Delay Settling Time, $K_d = 3$	108
Figure D.27	Fixed Point Delay Overshoot, $K_d = 3$	108
Figure D.28	Fixed Point Delay Rise Time, $K_d = 4.5$	109
Figure D.29	Fixed Point Delay Settling Time, $K_d = 4.5$	109
Figure D.30	Fixed Point Delay Overshoot, $K_d = 4.5$	109
Figure D.31	System Generator Rise Time, $K_d = 0$	110
Figure D.32	System Generator Settling Time, $K_d = 0$	110
Figure D.33	System Generator Overshoot, $K_d = 0$	110
Figure D.34	System Generator Rise Time, $K_d = 1.5$	111
Figure D.35	System Generator Settling Time, $K_d = 1.5$	111
Figure D.36	System Generator Overshoot, $K_d = 1.5$	111
Figure D.37	System Generator Rise Time, $K_d = 3$	112
Figure D.38	System Generator Settling Time, $K_d = 3$	112
Figure D.39	System Generator Overshoot, $K_d = 3$	112
Figure D.40	System Generator Rise Time, $K_d = 4.5$	113
Figure D.41	System Generator Settling Time, $K_d = 4.5$	113
Figure D.42	System Generator Overshoot, $K_d = 4.5$	113

Abbreviations

ADC	Analog to Digital Converter
CT	Continuous Time
DAC	Digital to Analog Converter
DSP	Digital Signal Processor
DT	Discrete Time
FPGA	Field Programmable Gate Array
GeV	10^9 electron Volts
Hg	Mercury
H ⁻	Negative Hydrogen Ion
H ⁺	Positive Hydrogen Ion
I	In-Phase
K _P	Proportional Gain Component
K _I	Integral Gain Component
K _D	Derivative Gain Component
kV	10^3 Volts
LINAC	Linear Accelerator
LLRF	Low-Level Radio Frequency
μs	Microsecond
ms	Millisecond
ns	Nanosecond
ORNL	Oak Ridge National Laboratory

PID	Proportional Integral Derivative
Q	Quadrature
RF	Radio Frequency
RLC	Resistor-Inductor-Capacitor
SNS	Spallation Neutron Source
SysGen	System Generator for DSP™
TF	Transfer Function
VHDL	VHSIC Hardware Design Language
VHSIC	Very High Speed Integrated Circuit

Chapter 1

Introduction

The Spallation Neutron Source (SNS) at Oak Ridge National Lab (ORNL) in Oak Ridge Tennessee is the most powerful pulsed neutron source in the world [1]. Neutron production begins with generating ~ 1 millisecond (ms) “macro” pulses of negative hydrogen ions (H^-) which are then chopped into 1 microsecond (μs) “mini” pulses before being accelerated to nominally 1 GeV by the SNS linear accelerator (LINAC). Fully-accelerated H^- mini-pulses are then passed through a diamond foil to strip the electrons from the ions, thus transforming them into protons (H^+) [2]. These proton mini-pulses are then injected into an accumulator ring where the mini-pulses are gradually combined into a concentrated beam pulse. Once all the protons for a particular machine cycle have been accumulated, the proton beam pulse is extracted from the ring and transported to a liquid Hg target for neutron production. Proton beam pulses are generated at a machine cycle rate of 60 Hz which results in ~ 1 MW of power delivered to the target. As the energetic protons strike the Hg nuclei, neutrons are “spalled” off en masse and transported to various beamlines for neutron scattering experiments. Figure 1.1.1 illustrates the layout of these SNS neutron production systems with the Ion Source, Linear Accelerator, Accumulator Ring, and Target Building. The focus of this work, however, is primarily concerned with the third major component: The Accumulator Ring.

The SNS Accumulator Ring is comprised of proton collimators, “kicker” magnets for injecting protons into the ring, “kicker” magnets for extracting the accumulated proton pulse from the ring, various multipole magnets for steering and focusing the proton beam, and finally,

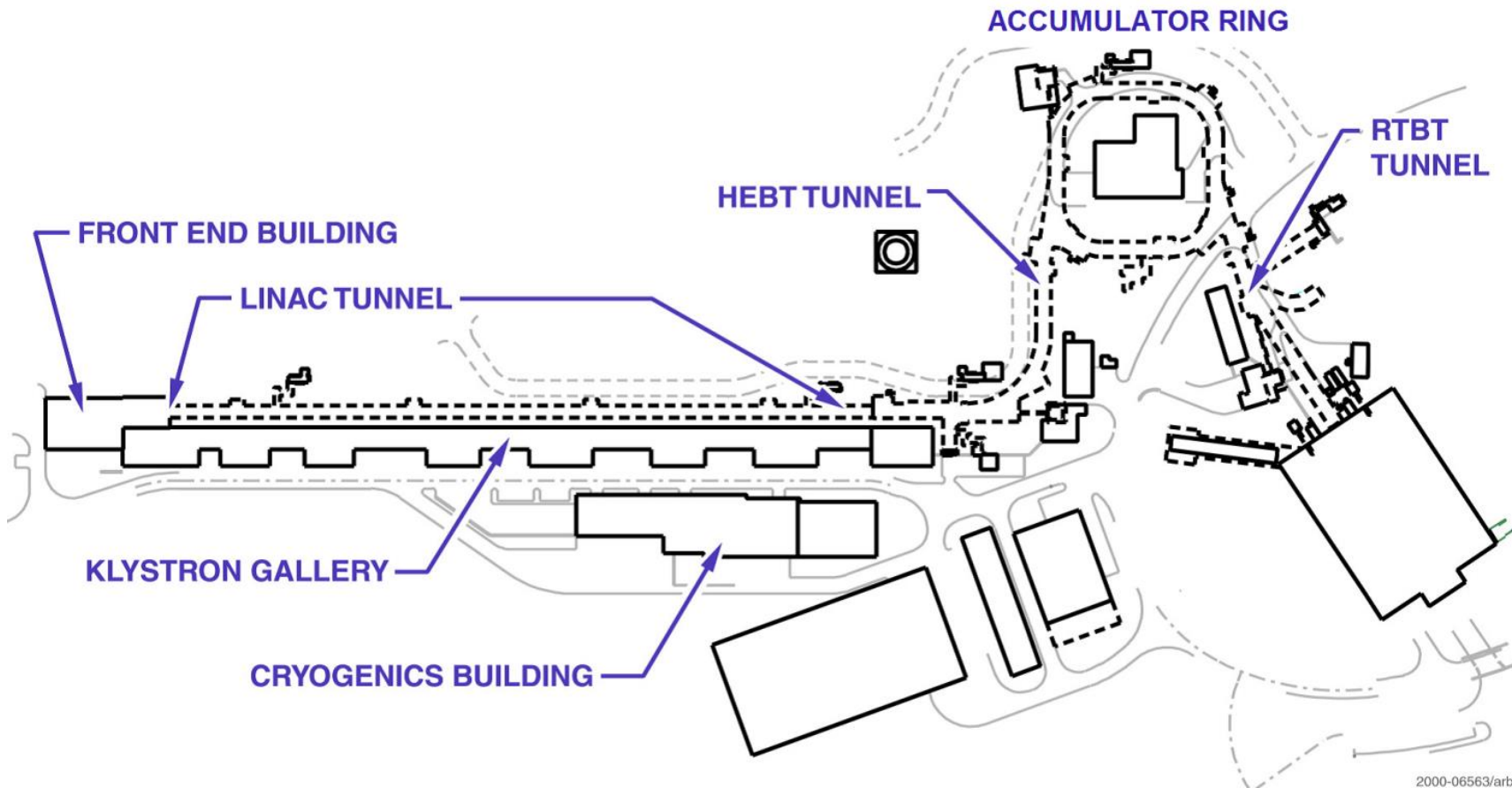


Figure 1.1.1 Spallation Neutron Source Site Layout

RF cavities for direct beam control and processing in the Accumulator Ring. The purpose of the RF cavities is twofold: maintain the “extraction gap” and reduce peak beam current [3]. A gap is defined simply as a continuous time period during which protons are absent. A gap of ~250 ns must be maintained so that the beam can be extracted without spraying highly-energetic protons across the beam pipes, magnets, and other critical hardware [2] [3]. Beam current throughout the Accumulator Ring pulse is kept as even as possible to reduce electrostatic repulsion forces between the like-charged protons. This is because higher beam current indicates a greater proton density, and since greater proton density implies smaller average separation distance, Coulomb’s Law forces between the protons see a quadratic increase. This powerful, internal separating force increases the possibility for beam instabilities and forces Accumulator Ring control systems to operate at higher powers, inducing premature wear and lower system reliability. Regardless, both beam current reduction and extraction gap regulation are accomplished by a Low-Level RF control system with the Accumulator Ring RF cavities serving as the actuators [2].

While the current LLRF subsystem has performed adequately for over 10 years, it is reaching the end of its functional life. Additionally, the Ring LLRF control system is quickly becoming obsolete. So in light of an aging system with unavailable replacement parts, it’s evident that the design of a new system is in order. And while a direct port of the existing system would be possible, a fresh study of the existing system followed by a considerably revamped version of the controller was pursued with the intent of a more robust, sustainable system.

The design process began with a study of each component of the LLRF control loop in question, beginning with the RF cavity – the plant of the system responsible for exerting control actions on incident beam. As one might expect, the RF cavity utilizes complex physical

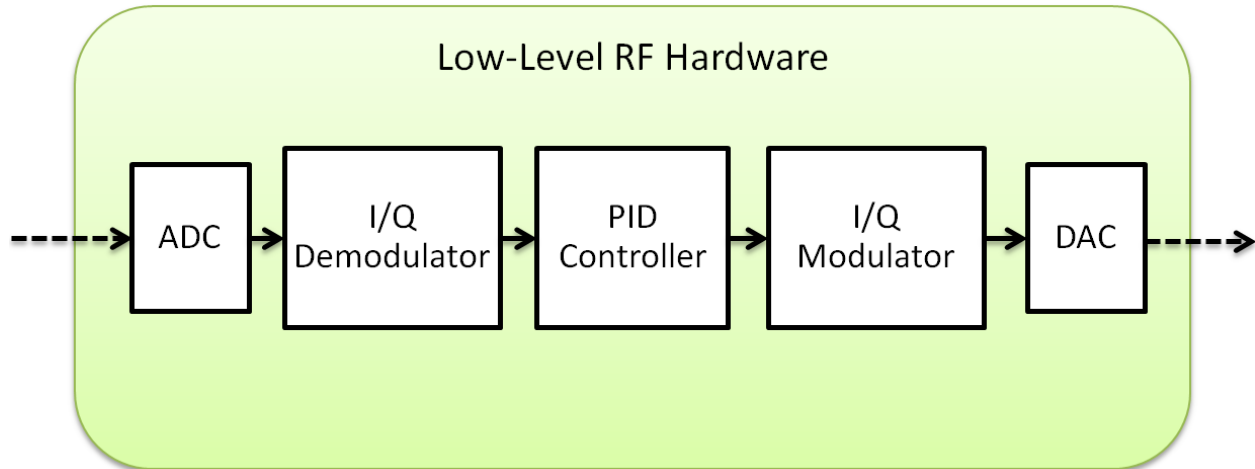


Figure 1.1.2 Functional Block Diagram of Original Controller

phenomena in its operation, and initial understanding of the physics defining these interactions was informed by existing SNS documentation [2] [3]. Deeper understanding and modeling was heavily derived from Harald Klingbeil’s work as well [4] [5]. From his analyses we were able to model the cavity as a parallel, current-driven, resistor-inductor-capacitor (RLC) network, and subsequently arrived at a state-space representation of the model.

Having established the model of the plant, development of a controller exhibiting desired topology and performance began. The developed controller model, seen in Figure 1.1.2, closely followed functions performed by the original controller, with in-phase (I) and quadrature (Q) demodulation of the incoming cavity state-signal feeding a proportional-integral-derivative (PID) controller that compared the split I and Q signals to individual reference values. An IQ actuating signal was then generated and modulated into an RF-regime drive signal controlling the cavity.

In the interest of being able to test possible changes more easily and to understand the various operational factors involved in the system, a software expression of the accumulator ring RF control loop model was produced. The software model is implemented in MATLAB

Simulink by MathWorks Inc., and makes special use of the Control Systems Toolbox. Because the new LLRF control system is FPGA-based, it was advantageous to modify the initial software model to accommodate the actual VHDL code. VHDL co-simulation was performed using Xilinx Incorporated's System Generator for DSP in conjunction with Simulink. This process greatly expedited development and verification prior to system integration testing on the Ring RF system.

The final step of testing the FPGA-based controller hardware provided crucial insight, verification, and validation for the previously completed work. Chapter 5 presents operational data in cases where the controller drove an in-situ cavity with both simulated and actual beam load. Discussion of this topic can be found in Chapter 5, though considerations thereof in this document are mostly relegated to high-level observations and correlations due to complexities of the physical control loop differing from the fundamental model whose design this thesis investigates. Such complexities include dynamic tuning of the RF cavities, beam loading as a result of incident beam, and operational heuristics.

This thesis is organized to guide the reader through a discussion of the process whereby a new LLRF control system for the SNS accumulator ring was designed and realized, while including sufficient background theory for an unfamiliar reader to understand the steps we took as well as demonstrate the validity of performed work through collected operational data. Chapter 2 will provide sufficient theory to explain the decisions and considerations behind much of the accumulator ring control loop design, including physics, operational sequences, and other design specifics. Once a base of theoretical knowledge is established, Chapter 3 will delve into the particulars of the work done in analyzing, modeling, simulating, and implementing the SNS

accumulator ring low-level RF control system, and Chapter 4 and Chapter 5 provide a more in-depth consideration of simulated and experimental results. Concluding remarks and future developments will be discussed in Chapter 6, after which appendices containing selected data may be found.

Chapter 2

Theory

Before discussing the details of my work on the system in question, it will be helpful to better understand some of Spallation Neutron Source's design and operation details concerning beam management. However, the SNS beam management systems represent vast amounts of technical information beyond the scope of this endeavor in many cases. So the following sections will eschew discussing unrelated subsystems for a thoughtful overview of topics whose particular considerations held direct influence over my work.

2.1 Physical

The principle of operation for an RF cavity is as follows: an electric field is created within the cavity coincident with the arrival of a charged particle having a velocity vector parallel to the field [4] [6]. The charged particle will experience a force whose direction is determined by charge and field polarity, thus controlling the particle to a desired state. This effect is illustrated in Figure 2.1.1 for both the field-between-two-plates case and the charges beam-pipe gap case.

It's important to remember, though, that no actual acceleration is performed by the Accumulator Ring RF cavities; as stated before, the ring LLRF control system is concerned with maintaining an adequate extraction gap while reducing peak beam current. These goals are accomplished by an acceleration scheme shown in Figure 2.2.1 where the ideal "synchronous" particle [7] passes through the cavity just as the gap voltage is crossing 0 volts, while

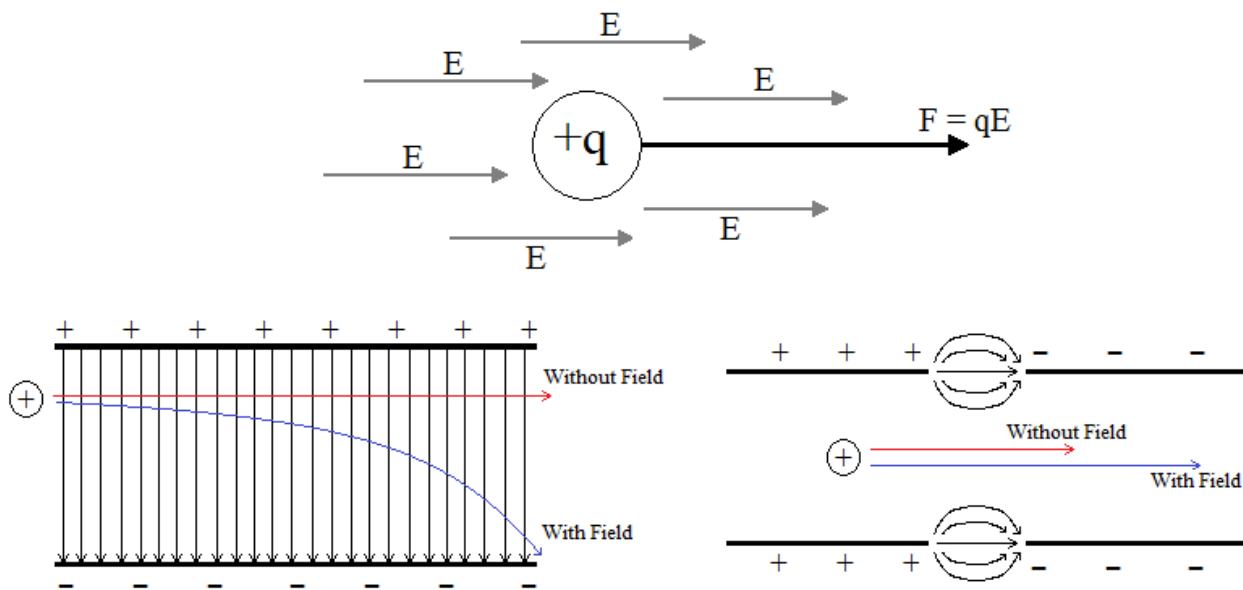


Figure 2.1.1 Coulomb Force on a Charged Particle

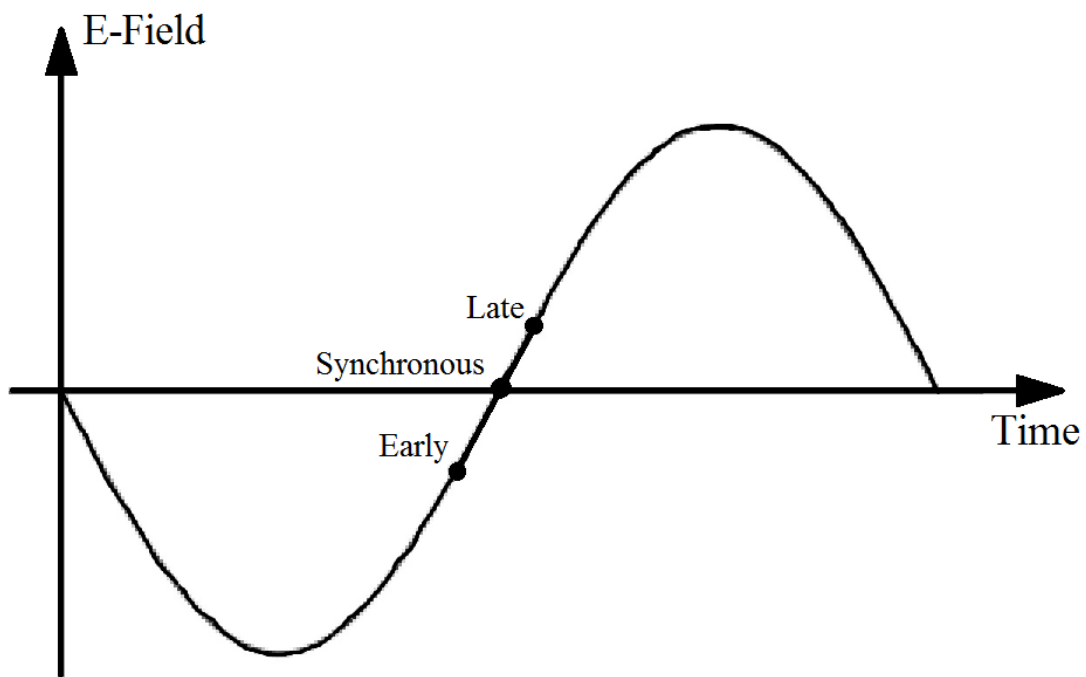


Figure 2.1.2 Plot of What Field Early, Synchronous, and Late Particles Experience

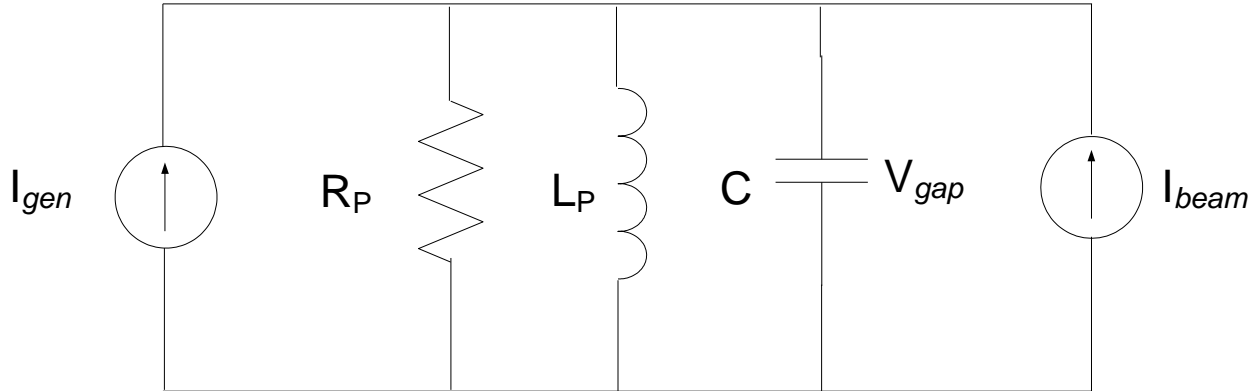


Figure 2.1.3 Equivalent Circuit Model of an RF Cavity [9]

a particle that's too early experiences negative acceleration and a particle that's late receives positive acceleration. This scenario results in a net bunching of the beam toward a central, synchronous phase value. So if protons are bunched toward one phase value, the phase value 180 degrees opposite necessarily experiences the opposite effect – a massive *reduction* in proton density. This opposite phase value where virtually all protons have been evacuated is precisely where the extraction gap appears.

This $F = qE$ phenomenon is leveraged by the Accumulator Ring ferrite-loaded RF cavities by inducing a voltage across a ceramic gap in the main cavity beam-pipe [4]. The cavities comprise four main features to accomplish this goal: the ceramic gap, the ferrite rings, ring-coupled windings, and the beam pipe itself. The beam pipe runs through the center of the flat faces of the ferrite rings, and those ferrite rings are driven by current-fed windings. Ultimately, the ferrite rings, capacitive gap, and windings all electromagnetically couple to induce the desired voltage across the ceramic gap, thus enabling force to be exerted on the beam.

We can see in Figure 2.1.3 that the three primary cavity components – the ferrite rings, capacitive gap, and windings – along with the driving current and beam current can all be

combined into an equivalent circuit mode [4] [5]. This equivalent circuit model can be used to form a differential equation describing time-domain system behavior, and s-domain techniques can be used to formulate a transfer function describing RF cavity frequency response as well. So if V_{gap} is understood as the gap voltage then

$$\frac{V_{gap}(t)}{R_p} + \frac{1}{L_p} \int V_{gap}(t) dt + C \frac{dV_{gap}(t)}{dt} = (I_{gen}(t) - I_{beam}(t)) \quad (2.1)$$

is the differential equation describing the time progression of the gap voltage with parameters R_p , L_p , and C as a function of the drive and beam currents. This can easily be converted into the s-domain expression

$$\frac{V_{gap}(s)}{R_p} + \frac{V_{gap}(s)}{sL_p} + sC \cdot V_{gap}(s) = (I_{gen} - I_{beam}) \quad (2.2)$$

which can be easily expressed in transfer function form as

$$H(s) = \frac{Y(s)}{X(s)} = \frac{I_{gen} - I_{beam}}{V_{gap}(s)} = \frac{1}{R_p} + \frac{1}{sL_p} + sC \quad (2.3).$$

However, these bunching effects won't occur at LLRF power levels. The cavities of interest in the accumulator ring are driven by a tetrode-amplified drive current produced by the LLRF proportional-derivative-integral (PID) based controller. This controller-driven current feeds windings that electromagnetically couple to the cavity as stated before and induce a desired voltage. Next, voltage 'pick-offs' present in the accumulator ring cavities then measure voltage across the cavity gap and feed that value back to the controller so feedback control may be performed and the cycle can continue.

In summary, the SNS physical systems of interest for this work are the ferrite loaded cavities in the accumulator ring and the accompanying RF controller. However, controller

operation is largely constrained by process timing requirements, the specifics of which can be found in the next section.

2.2 System Timing

Timing considerations for SNS operation are important to understand as these conventions heavily influence several design decisions covered in Chapter 3. Because the SNS accelerator is a pulsed machine, the systems responsible for performing critical SNS neutron production tasks have very cyclic operation. In addition to general machine cycle timing, there are also particular beam timing conventions in place for optimal acceleration and neutron production tasks.

Machine Cycle Timing

The fundamental machine cycle rate for SNS is 60 Hz, with “Time Critical” and “Non-Critical” tasks occurring therein [8]. Although the synchronization of these tasks could be described with seconds, timing of the tasks are actually a function of Ring frequency, and thus the use of seconds is less desirable. SNS machine cycles instead use ‘turns’ as the fundamental timing unit. A turn is the amount of time it takes for the beam to transit once around the accumulator ring which is nominally 1 ms.

This slightly variable timing for something as timing-sensitive as a particle accelerator is brought on as a function of two things: beam energy and Accumulator Ring circumference. As before, Accumulator Ring turn frequency is simply determined by how fast a particle is going around a ring of set circumference; higher energy implies higher velocity and thus, shorter transit time and higher turn-frequency. A derivation of this phenomenon with the particular values for ideal ring operation is available in Appendix A, though the end result is that with an ideal

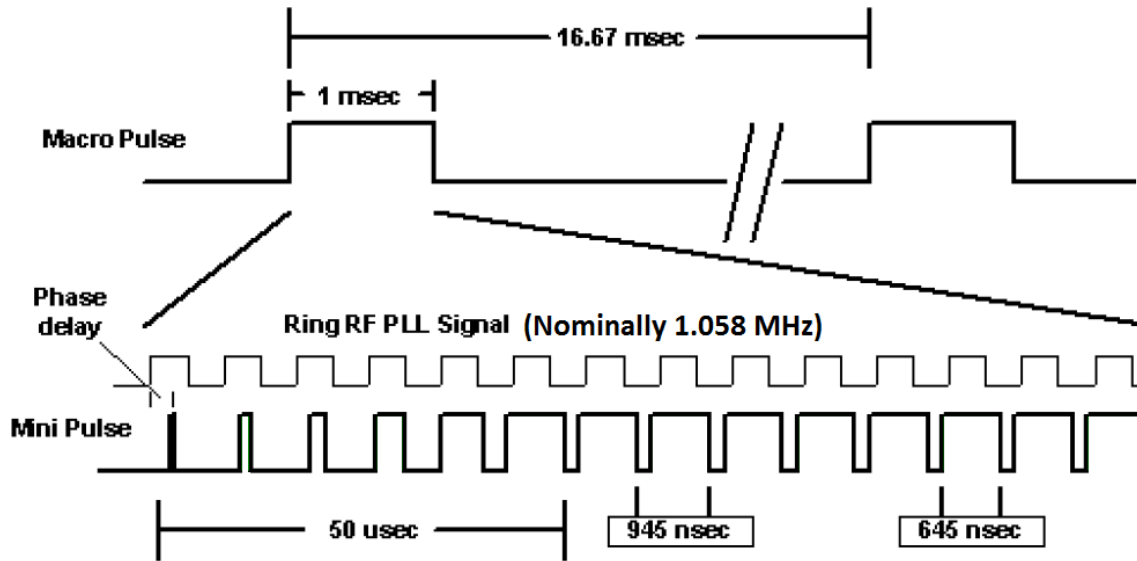


Figure 2.2.1 Breakdown of the SNS Beam Pulse Structure [8]

incident beam energy of 1 GeV the corresponding turn frequency turns out to be 1.058 MHz, which is a turn period of 945.1 ns [3].

Beam Timing

In order to accomplish desired beam operations, the SNS particle beam is produced and managed in a special form. Spallation Neutron Source is a pulsed beam accelerator, and each beam pulse contains three levels of structure: the macro pulse, the mini pulse, and the micro pulse. The macro pulse duration can range from 1 to 1060 turns [5], with more turns meaning longer acceleration time and higher beam intensity delivered to the target. Upon entering the accumulator ring each roughly millisecond macro pulse is bunched into a roughly microsecond mini pulse. Each mini pulse consists of several micro pulses occurring at a rate of 402.5 MHz, and the number of micro pulses determines the pulse-width for the corresponding mini pulse. A breakdown of these structures and their interrelationships can be seen in Figure 2.2.1.

Timing Summary

The sole purpose of SNS systems prior to the target is to produce and maintain beam pulses in accordance to specific timing characteristics that revolve around turn-defined events. These concepts and tasks strongly influenced the research, design, and analysis discussed in this work. Further inquiry should be directed to the document titled “SNS Timing Master Functional System Description” [8].

2.3 Control Actions on Signal of Interest

To fulfil timing requirements, the goal of the LLRF control system is to maintain RF amplitude within +/- 1 percent, and RF phase within +/- 1° of a given reference in the presence of beam current [4]. For the Accumulator Ring cavities in particular, a design basis of 10 kV was taken for gap voltage amplitude. The gap voltage frequency target is 1 MHz and 2 MHz for first and second harmonic operation respectively, though it's important to know that the exact reference values for operations can vary depending on operational settings. For instance, typical gap voltage amplitude for first harmonic cavities is about 8 kV and 6 kV for second harmonic cavities, and the frequencies are nominally 1.058 MHz and 2.116 MHz respectively [10].

As stated in Chapter 1, these voltage amplitude and phase requirements are met through a PID-based feedback control system illustrated in Figure 2.3.1 that comprises an IQ demodulator, a PID controller, a phase rotator, and an IQ modulator. Following the feedback control system's signal processing and drive-signal modulation, the resulting waveform is amplified into a high-power drive current. That drive current is then passed through windings that couple to ferrite rings within the ferrite-loaded accumulator ring cavity, thus inducing the desired voltage in the

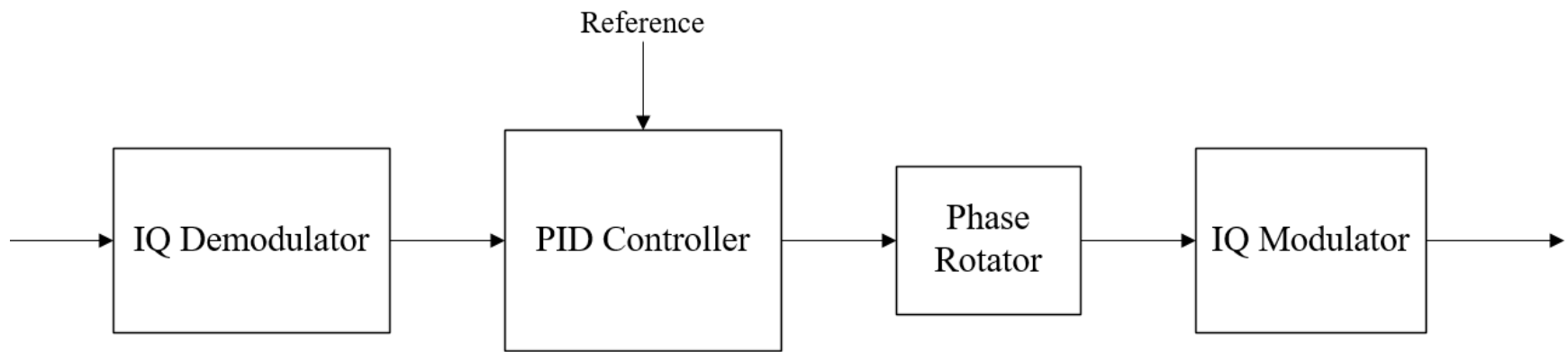


Figure 2.3.1 Accumulator Ring LLRF Controller Block Diagram

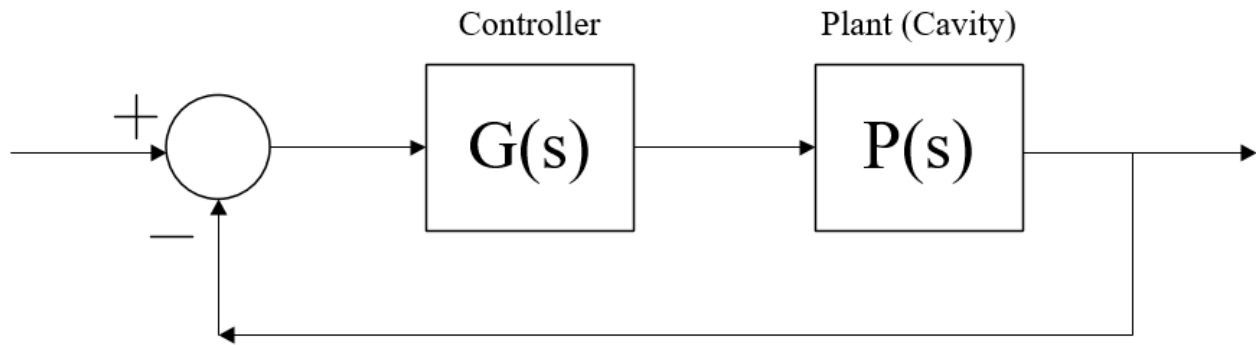


Figure 2.3.2 Traditional Feedback Controller Loop

cavity and controlling the system to a likewise desired state. This state is then observed by measuring cavity voltage using voltage pick-off devices and fed back to the controller input, completing the feedback control loop. Like most feedback control loops, the system can be represented with a block diagram of the salient components, shown in Figure 2.3.2.

The Figure 2.3.2 block diagram of the control system exhibits traditional structures, with the plant $P(s)$ - in our case the ferrite-loaded cavity – being driven by controller $G(s)$. It should be emphasized that the parameters to be controlled are the amplitude and phase of the sinusoid. The frequency of the sinusoid is not the object of the controller. The controller first demodulates the RF input into In-Phase and Quadrature (IQ) components before performing actual control functions on the signal. If IQ demodulation was not performed, the controller would constantly try to drive based on a 1 or 2 MHz reference signal when the true system reference is more concerned with amplitude and phase. Incorporating a demodulator results in a simpler controller which likely decreases hardware cost while increasing system performance.

Once the appropriate drive signal has been produced, the I and Q components are then re-modulated back to the appropriate frequency to be fed to the plant and the cycle continues the

same as before. Fortunately, system operation as described by the resulting transfer functions of the controller is the same with and without IQ modulation and demodulation; proof that IQ modulation and demodulation are linearly reciprocal actions may be found in Appendix B, but a graphical representation of the concept can be seen in Figure 2.3.3.

So IQ demodulation is performed on the incoming feedback signal, and once the I and Q components have undergone PID control action they can be re-modulated without issue, but how are those PID control actions defined in this system? The chosen SNS Accumulator Ring LLRF controller follows a structure presented by Thomas Schilcher in his work “Digital Signal Processing in RF Applications” and shown in Figure 2.3.4, and Appendix C shows that Schilcher’s structure is functionally identical to the existing system PID controller form. So traditional PID control actions are performed by the Accumulator Ring LLRF controller, and due to IQ demodulation just prior to the control block, the system has separate PID controllers within that control block for both I and Q signal paths; each control path has independent reference and gain values with no signal coupling between, and the main commonality between paths is the PID gains K_P , K_I , and K_D , though even then there isn’t a requirement for them to be shared between paths beyond ease of analysis. See Figure 2.3.5 for a representation of these aspects.

Continuing the analysis of Figure 2.3.2, after the PID controllers comes a less-familiar component: the phase rotator. The principle behind phase rotation is that in the event of a known, preferably constant phase error up to $\pm 180^\circ$ from a desired phase as a result of time delay within a control loop, a phase rotator can perform trigonometric operations on an IQ-format signal to correct such error. Unfortunately this delay-induced error can’t be infinitely corrected due to cyclic repetition of sinusoids and especially since all real systems are necessarily causal. Despite

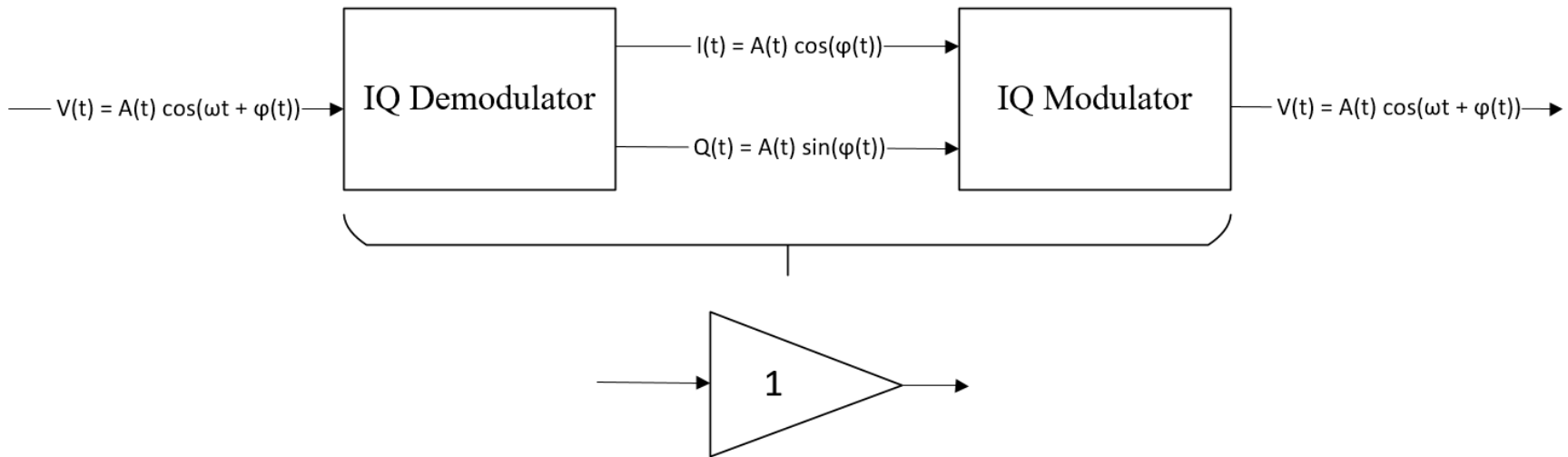


Figure 2.3.3 IQ Modulation-Demodulation Linear Reciprocity

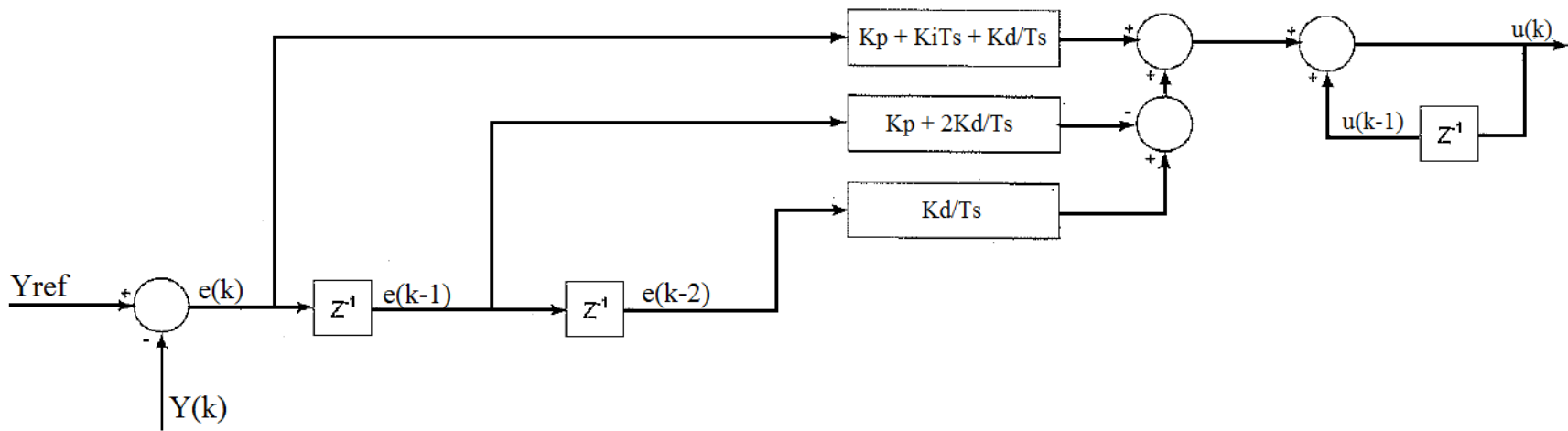


Figure 2.3.4 Discrete Time PID Controller Topology [9] [11]

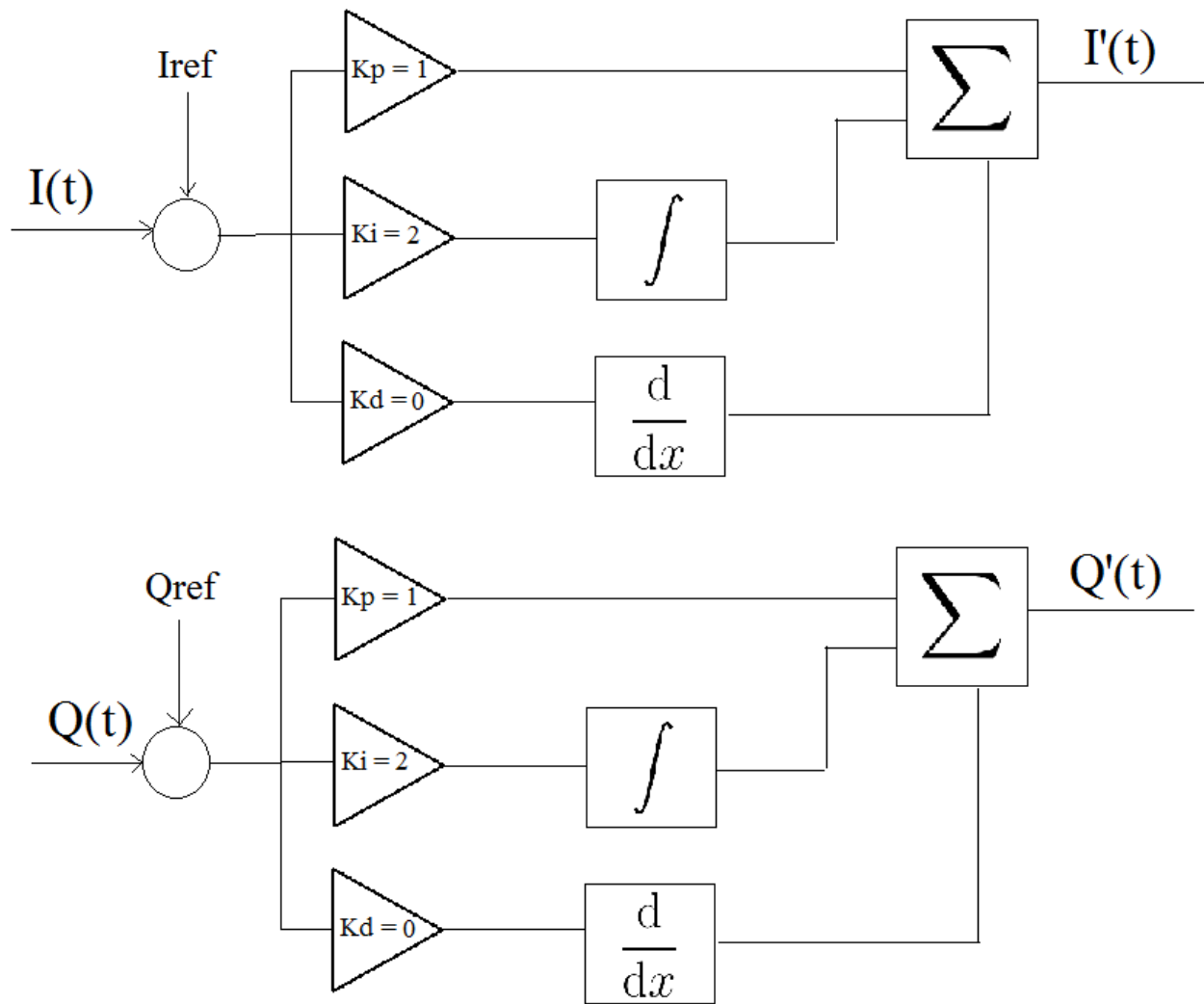
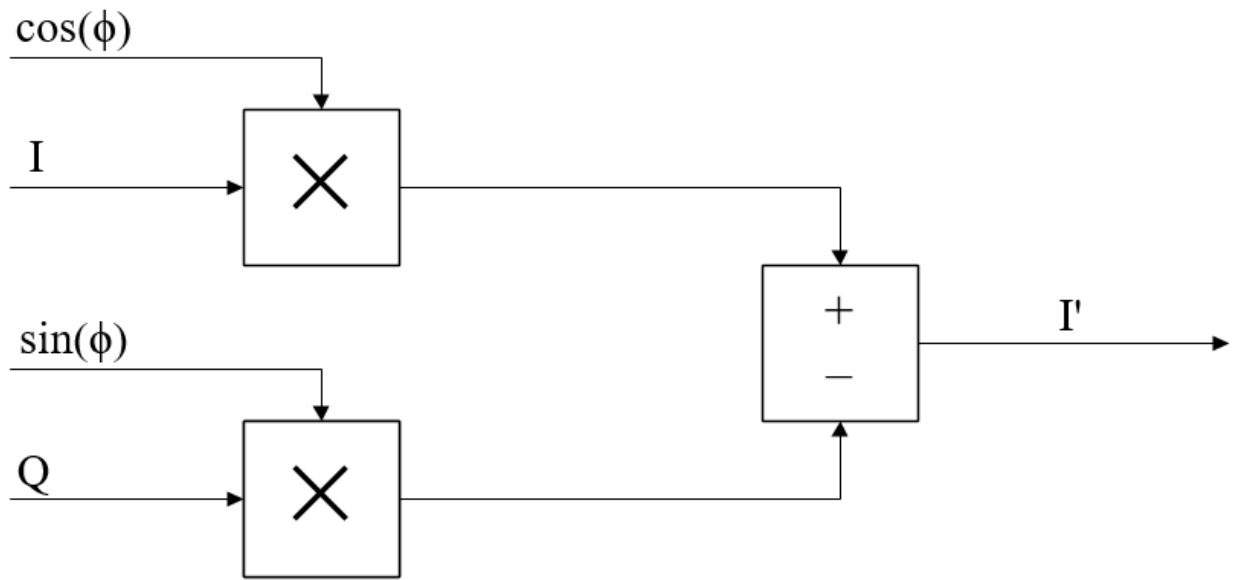


Figure 2.3.5 Similarities and Differences between I and Q Paths

practical and theoretical limitations, a phase rotator allows for some degree of correction within a frequency's period, and Figure 2.3.6 demonstrates the equations and layout of how phase rotation is implemented. Examples of implemented phase rotator operation are shown in Chapter 4, while design considerations are discussed in Chapter 3.

The final theoretical consideration of the system regards the method used for IQ demodulation. As mentioned before, the demodulator operates by multiplying an incoming signal of a given frequency by a reference sinusoid of typically similar frequency, and depending on the phase of the multiplying reference sinusoid the result is either the I component or Q component, though together they represent the amplitude and phase of the input signal. So from input to output, generation of the I and Q components usually happens instantly or at least on every clock in digital systems, but the SNS Accumulator Ring uses a technique known as “Non-IQ Sampling” [11]. Non-IQ Sampling is similar to IQ sampling, but instead of generating new I and Q values each clock, scaled I and Q values are accumulated over a cyclic period, with the output being this accumulated value which updates once per period. This is a result of a Least Mean Squares method for estimating the correct phase in the presence of noise and other errors [11]. SNS non-IQ sampling involves dividing 64 MHz I and Q samples by 64 then accumulating them over the course of a single 1 MHz turn before outputting the accumulated value and resetting the sum for the next turn. Each output value is held for the duration of a turn before being updated. This summing can perhaps be better expressed by [11]

$$I = \frac{2}{N} \cdot \sum_{i=0}^{N-1} y_i \cdot \sin(i \cdot \Delta\varphi) \quad (2.1)$$



$$I' = I \cdot \cos(\phi) - Q \cdot \sin(\phi)$$

$$Q' = I \cdot \sin(\phi) + Q \cdot \cos(\phi)$$

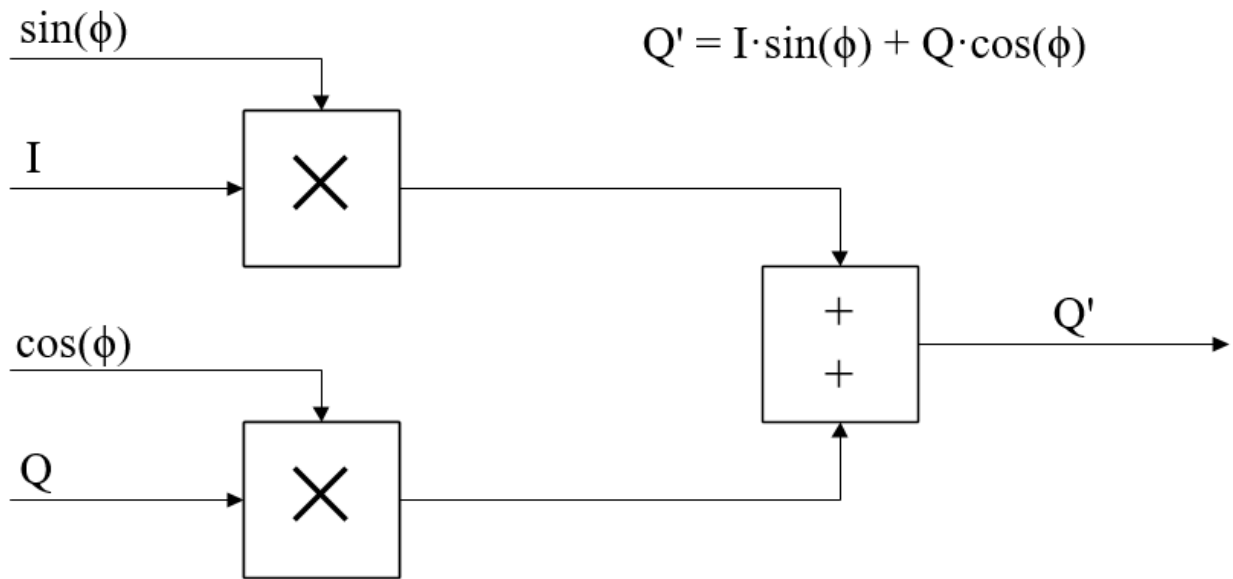


Figure 2.3.6 Block Diagram of Phase Rotator with Equations

$$Q = \frac{2}{N} \cdot \sum_{i=0}^{N-1} y_i \cdot \cos(i \cdot \Delta\varphi) \quad (2.2)$$

Where N is the number of samples taken in M sample periods, which are 64 and 1 respectively for the SNS Accumulator Ring, and

$$\Delta\varphi = \frac{2\pi M}{N} \quad (2.3).$$

So the SNS Accumulator Ring non-IQ sampling is defined for I and Q as

$$I = \frac{2}{N} \cdot \sum_{i=0}^{N-1} y_i \cdot \sin\left(i \cdot \frac{\pi}{32}\right) \quad (2.4)$$

$$Q = \frac{2}{N} \cdot \sum_{i=0}^{N-1} y_i \cdot \cos\left(i \cdot \frac{\pi}{32}\right) \quad (2.5).$$

Chapter 3

Design

3.1 Cavity Modeling

The first step in creating a model of the LLRF ring cavity control stations was to model the cavity itself. As discussed previously, the beam control device in question is a ferrite-loaded RF cavity whose fields are induced by driving the appropriate current through windings that electrically couple to the ferrite plates. Most of the information regarding the design and operation the cavity in question came from internal documents as well as “Theoretical Foundations of Synchrotron and Storage Ring RF” Authored by Harald Klingbeil. However, for understanding the physical phenomena for purposes of modeling cavity operation Klingbeil's work was invaluable.

As discussed in Chapter 2, the physical configuration of the cavity can be extended to its electrical equivalents: the ferrite rings are modeled as inductors, the ceramic gaps in the conductive beam guiding structures can be conceptualized as capacitors, and the loss incurred from driving the cavity through its windings correlates to resistors. All together, the drive current, ferrite rings, ceramic gaps, windings, and eventual beam current can be combined into a parallel Resistor-Inductor-Capacitor (RLC) circuit, visible in Figure 2.1.2. This conversion was the key in moving forward, and conveniently allowed for some possibly daunting math to be represented in a keenly familiar format.

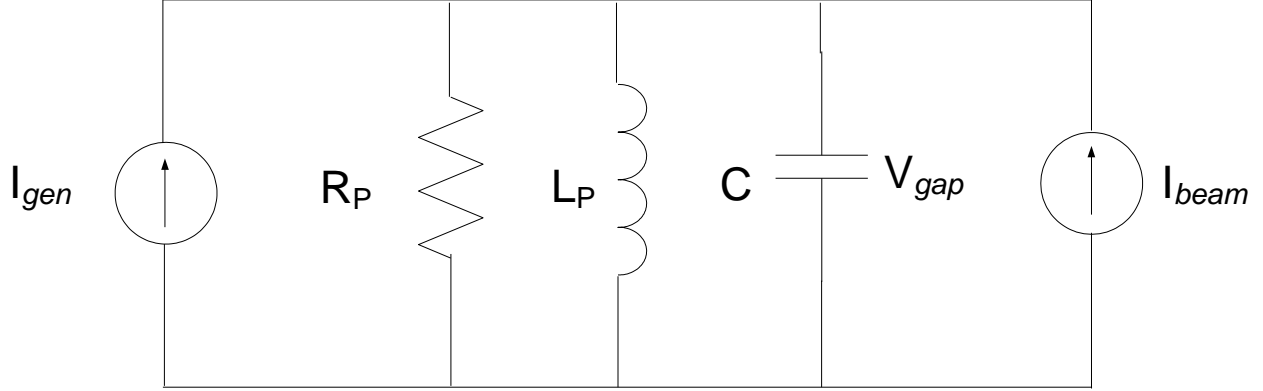


Figure 3.1.1 Cavity Circuit Model [9]

Once the RLC representation of the SNS cavities was arrived at, conversion into a more manageable format for modeling began. Transfer functions and explicit per-component descriptions were investigated, but state-space formalism was ultimately selected for its ease of application despite difficulty in initial formation of many state-matrices. So, returning to Figure 3.1.1 we know that a particular voltage or current can be understood through the differential equation

$$\frac{V_{gap}}{R_p} + \frac{1}{L_p} \int V_{gap} dt + C \frac{dV_{gap}}{dt} = (I_{gen} - I_{beam}) \quad (3.1)$$

which can be re-written as

$$\frac{dV_{gap}}{dt} = \frac{1}{C} (I_{gen} - I_{beam}) - \frac{1}{R_p C} V_{gap} - \frac{1}{L_p C} \int V_{gap} dt \quad (3.2)$$

and if we define the state vector x as

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \int V_{gap} \\ V_{gap} \end{pmatrix} \quad (3.3)$$

and input vector u as

$$u = (I_{gen} - I_{beam}) \quad (3.4)$$

then

$$\dot{x} = \begin{pmatrix} V_{gap} \\ \dot{V}_{gap} \end{pmatrix} \quad (3.5)$$

And when equation 3.2 is described in terms of equation 3.4 and 3.5, the system of equations

$$\begin{cases} \dot{x}_1 = \int \dot{V}_{gap} = x_2 \\ \dot{x}_2 = \dot{V}_{gap} = -\frac{1}{L_p C} \cdot x_1 - \frac{1}{R_p C} \cdot x_2 + \frac{1}{C} \cdot u \\ y = V_{gap} = x_2 \end{cases} \quad (3.6)$$

arises, which can then be converted to a matrix form

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -\frac{1}{L_p C} & -\frac{1}{R_p C} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{C} \end{pmatrix} u(t) \quad (3.7a)$$

$$y = (0 \quad 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (3.7b)$$

a format that conveniently matches the general state-space expression

$$\dot{x} = A \cdot x + B \cdot u \quad (3.8a)$$

$$y = C \cdot x + D \cdot u \quad (3.8b)$$

with

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{1}{L_p C} & -\frac{1}{R_p C} \end{pmatrix} \quad (3.9a) \quad B = \begin{pmatrix} 0 \\ \frac{1}{C} \end{pmatrix} \quad (3.9b)$$

$$C = (0 \quad 1) \quad (3.9c) \quad D = 0 \quad (3.9d).$$

The relationship between the continuous-time (CT) transfer function and state-space variables is given by [12]

$$W(s) = C(sI - A)^{-1} + D \quad (3.10)$$

The transfer function, $P(s)$, describing the plant of our system,

$$W(s) = P(s) = \frac{1}{C} \cdot \frac{s}{s^2 - \frac{1}{L_p C} s + \frac{1}{R_p C}} \quad (3.11)$$

Other Considerations

For the SNS first-harmonic cavity, the component parameters are taken as $R_p = 800 \Omega$, $L_p = 7.54 \mu\text{H}$, and $C = 3 \text{ nF}$. These values are primarily taken from SNS literature [2] [3]. For our purposes the chosen parameters mimic cavity operation well enough for the model to be valid. Also, the effects of incident beam on cavity operation have been largely neglected for purposes of modeling. The effects include a phenomena known as Lorentz detuning of the cavity [2] as well as equivalent circuit interactions with what is essentially a positive current since a proton beam is simply a flow of positive charge carriers. Cavity detuning can be represented by a parameterized inductor value in the equivalent circuit and the positive current can be accounted for by simply modifying the existing drive current. In the real system, normal operation of the controller hardware largely accounts for the perturbations, and an adaptive feed-forward system serves to further ameliorate remaining effects. A detailed treatment of these considerations is outside the scope of this work.

3.2 Controller Model Design

Once the ferrite-loaded cavity was understood and modeled, the next task was to design a model of a controller. The decision was taken by the project team to base the updated controller design off the legacy system's topology, after which improvements above baseline performance could

be pursued. It's also worth noting that even if there weren't substantial performance improvements in the updated system, the design process would greatly advance overall system quality as the redesign brought about a workable system model: something operationally non-existent beforehand. Thus, the updated Ring LLRF Controller comprises an IQ demodulator, parallel PID controllers, a phase rotator, and finally an IQ demodulator. The following subsections will discuss the specific expression of these components for the designed controller model.

IQ Demodulator

As described in Section 2.3, the controller is designed for the slowly varying characteristics of the plant; therefore, as seen in Figure 3.2.1, the input signal from the cavity to the controller is demodulated from a fully sinusoidal wave into its in-phase and quadrature components. Non-IQ sampling is applied to the cavity feedback signal to produce its I and Q components. This technique provides the additional benefit of decimating the sampling frequency such that IQ values are generated once per turn. Non-IQ sampling involves first dividing the magnitude of each input sample by a factor N – the number of ADC samples per turn – after which the signal path is split in two. One branch is multiplied by a cosine running at the input frequency to create the in-phase (I) term, and the other branch is multiplied by a sine running at the same input frequency, creating the quadrature (Q) term. Each of these branches then feed into separate accumulators where N samples of the scaled, demodulated I or Q signal are summed together and output. For the SNS application, N is set to 64. When the output is generated, the running sum is reset back to 0 for the next accumulation cycle. To reiterate, the IQ Demodulator output changes once per 64-sample turn, holding that value for the duration of the

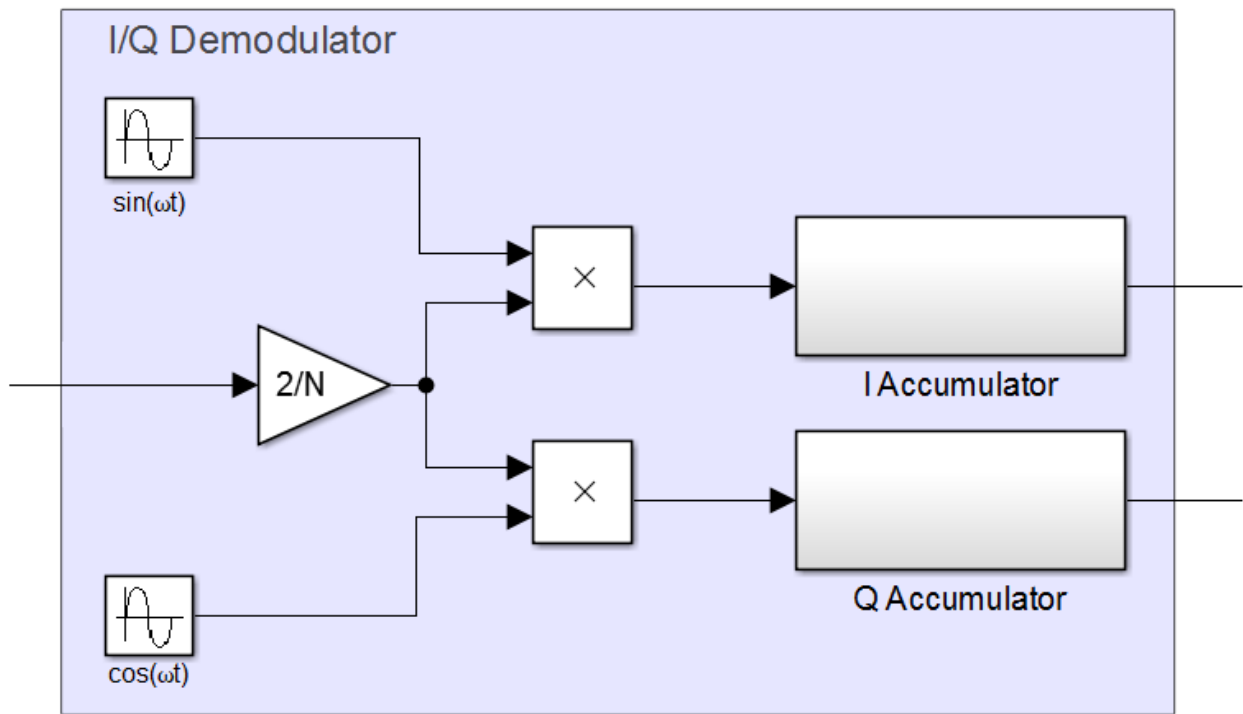


Figure 3.2.1 IQ Demodulator Block Diagram

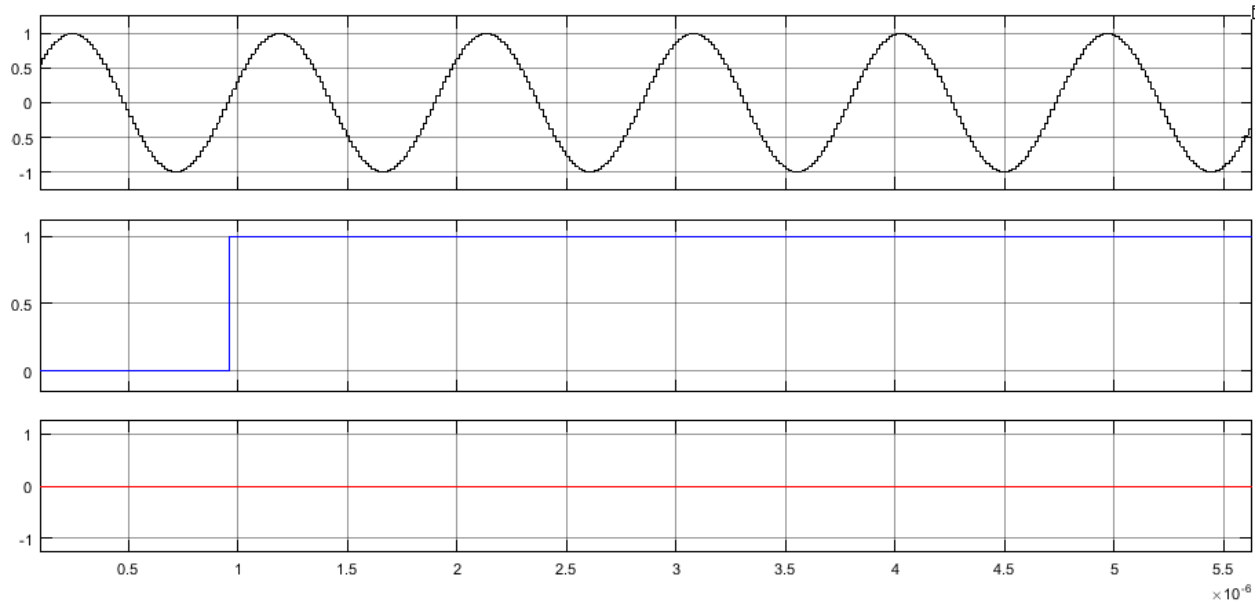


Figure 3.2.2 IQ Demodulator Signal Demonstration

turn, with the currently held value always being that of the previous accumulation cycle. Figure 3.2.1 shows the structure of the IQ Demodulator, and Figure 3.2.2 shows the IQ decomposition of a signal perfectly in-phase with the reference.

PID Controller

The block diagram for the designed system's PID controller follows T. Schilcher's form, and it was proven in Appendix C that traditional PID control can be accomplished using Schilcher's form. The choice to follow Schilcher's design was based on the fact that early in the design process his cavity model definitions and analysis were being considered for use in our model; however, his modalities became less attractive as time went on, and eventually the in-house state-space model of the cavity was settled on, though the already functional elements of his design remained intact. A diagram of the chosen controller may be seen in Figure 3.2.3 which has the following transfer function:

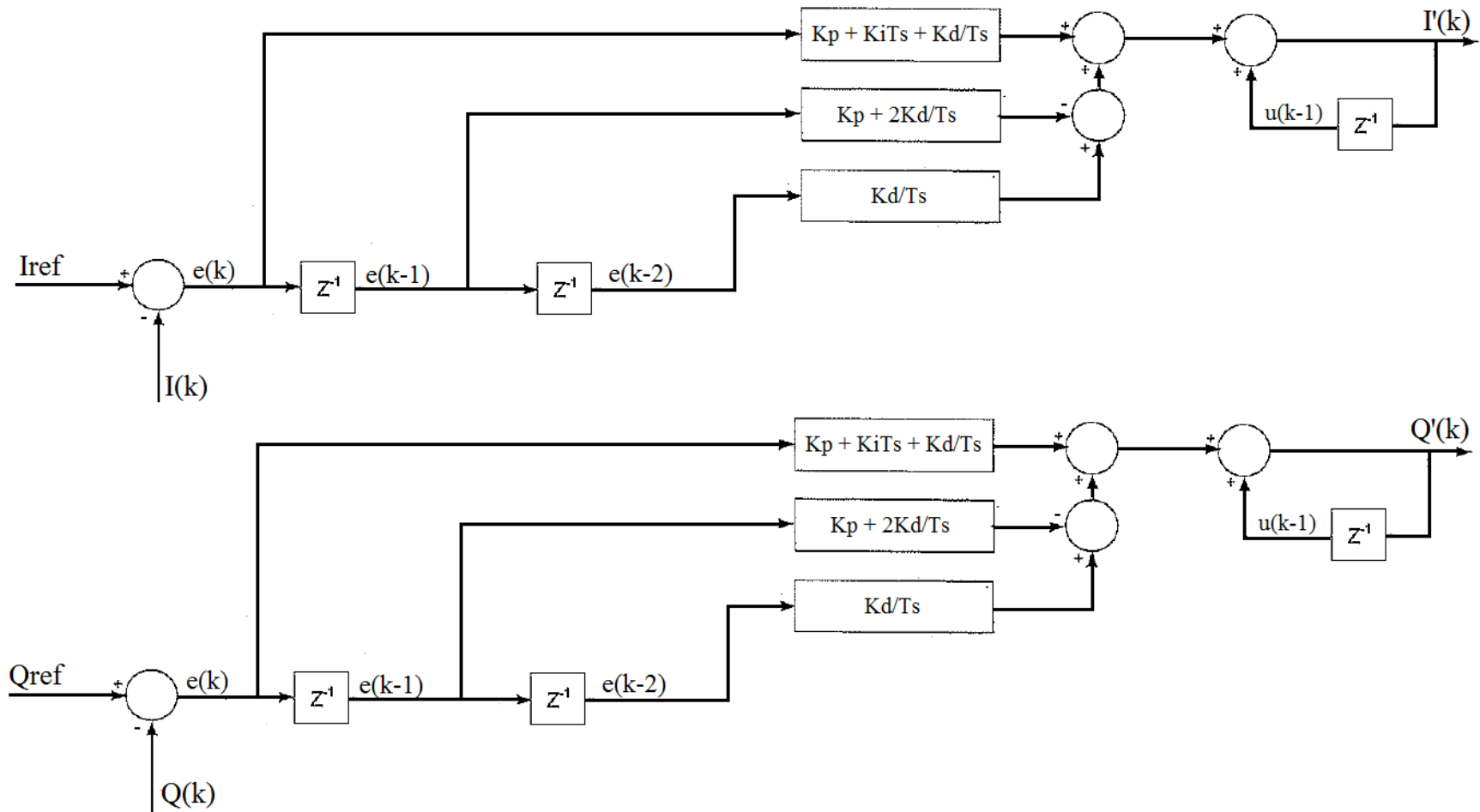


Figure 3.2.3 PID Controller Diagram [9] [11]

$$\frac{u(z)}{e(z)} = \frac{[K_p + K_i + K_d] - z^{-1}[K_p + 2K_d] + z^{-2}K_d}{1 - z^{-1}} \quad (3.12)$$

Phase Rotator

Figure 3.2.4 shows the particular design chosen for the model's Phase Rotator. It multiplies incident I and Q signal components with sine and cosine terms with a desired rotation angle before summing the appropriate signals back into corrected I and Q components while preserving IQ composite magnitude. Since the theoretical model has zero latency this block's transfer function will not be considered at this time, and influence from this block in latency-probable situations will be numerically compensated.

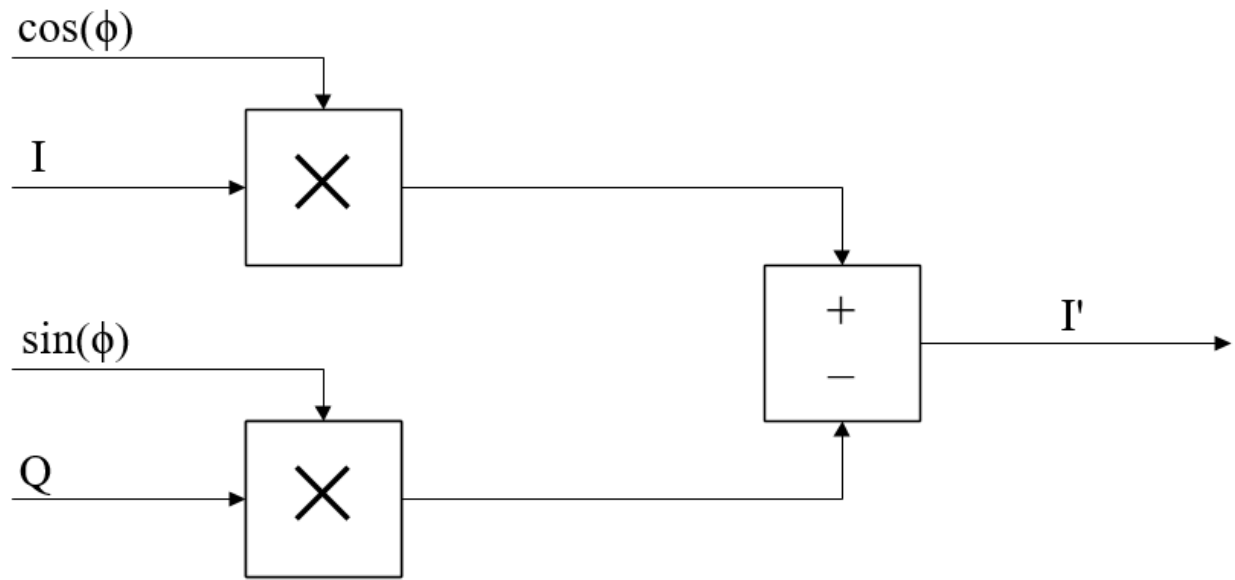
IQ Modulator

The design process for the IQ Modulator involved a simple translation of the governing equations into block form. As proven before, the transfer function of IQ modulation and demodulation linearly combined is 1; the IQ modulator transfer function is nevertheless provided in Figure 3.2.5 alongside the expression thereof in block diagram form for the model.

Combined Controller Transfer Function

So due to the fact that the modulator and demodulator have a net transfer function of 1, and for the delayless case the phase rotator has a transfer function of 1, the resulting transfer function of the entire SNS Accumulator Ring RF Cavity Station controller $G(s)$ is simply that of the chosen PID controller transfer function:

$$\frac{u(z)}{e(z)} = G(s) = \frac{[K_p + K_i + K_d] - z^{-1}[K_p + 2K_d] + z^{-2}K_d}{1 - z^{-1}} \quad (3.13)$$



$$I' = I \cdot \cos(\phi) - Q \cdot \sin(\phi)$$

$$Q' = I \cdot \sin(\phi) + Q \cdot \cos(\phi)$$

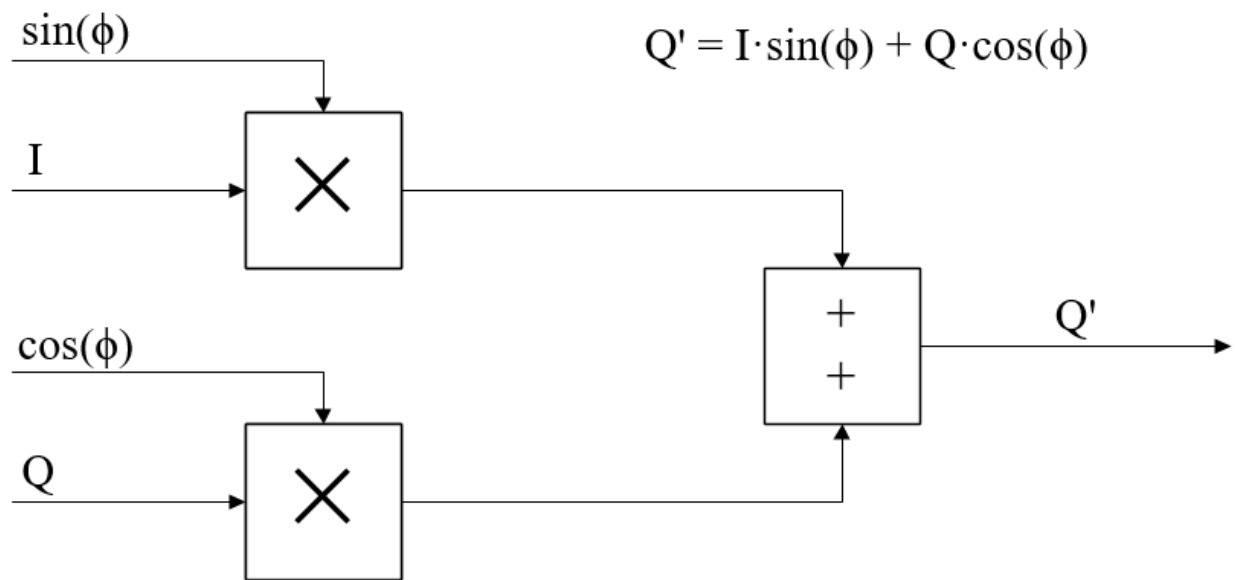


Figure 3.2.4 Phase Rotator Diagram

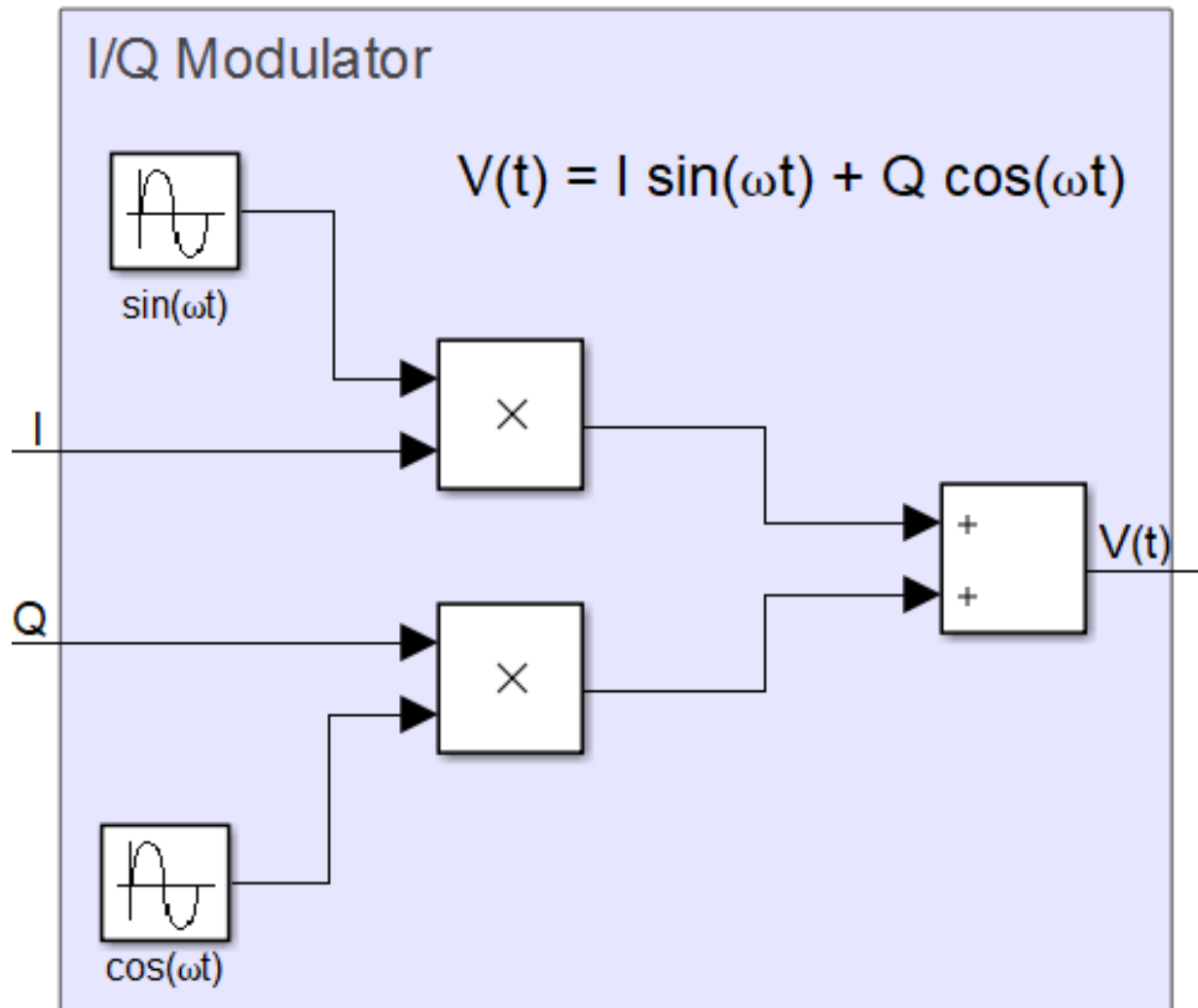


Figure 3.2.5 IQ Modulator Diagram

And when expressed as a complete control loop with controller $G(s)$ and plant $P(s)$ the open-loop (no feedback) transfer function can be expressed as H_{OL} while the closed-loop (with feedback) transfer function avail itself as H_{CL} :

$$H_{OL} = G(s) \cdot P(s) \quad (3.14) \qquad H_{CL} = \frac{G(s) \cdot P(s)}{1 + G(s) \cdot P(s)} \quad (3.15).$$

3.3 Software Model Design

Another key element of the design process was verifying the theoretical model by expressing it in the MathWorks MATLAB Simulink software environment. Having a software version of the theoretical model was critical for achieving analysis and verification goals for the project within a reasonable amount of time as relying on pure theory made experimentation involving multiple configurations difficult. Indeed, testing parameter configurations in an equation-only model requires a complete rework of existing equations to analyze each change while a software model enables rapid, simultaneous iteration, even allowing for automation of multi-version testing involving multiple parameters. Additionally, for the purposes here a software model promised the ability to inform construction, co-verification, and final implementation of the subsequent hardware model.

These considerable advantages lead to a quick adoption of the Simulink environment as the modeling method of choice, and this section describes the design process and results of this pursuit. Model simulation data and comparisons between the various system models may be found in Chapter 4. Additionally, even more simulation data may be found in Appendix D, though with a more metrics-oriented emphasis.

Floating Point Model

Three software model versions are maintained: a floating point math model, a fixed point math model, and a Xilinx System Generator VHDL-integrated model. The floating point math model serves a primarily theoretical purpose, with most of the conversion from system equations to Simulink modalities occurring here. The fixed point model saw limited use as a place where the effects of fraction length on system behavior were investigated before finally deciding on a fixed point integer and fraction length. Finally, the Xilinx System Generator VHDL-integrated model used Xilinx SysGen to run VHDL black-box code within a Simulink model prior to that code being programmed onto physical FPGA hardware.

Starting with the earliest model, initial conversion from control system block and transfer function representation of the desired topology was implemented in a Simulink model using floating point math despite the fact that the ultimate system would use fixed point math. This design choice was made so that early model expression could focus on general theory and model verification without the process being obfuscated by unrelated arithmetic precision concerns. Nevertheless, the following subsection discusses other such decisions regarding how initial expression of the theoretical representation of the SNS Ring LLRF control system was done in Simulink, with an emphasis on the differences introduced to accommodate the Simulink environment.

Generally, the floating point model follows the theoretical model quite closely. Indeed, the only high-level difference between the floating point model as seen in Figure 3.3.1 and the theoretical model in Figure 2.3.2 is a gain block in the feedback path. But a deeper examination of each model component reveals slight changes on all fronts.

IQ Demodulator

The software model version of the IQ Demodulator block can be seen in Figure 3.3.2. The only notable difference introduced in reproducing the demodulator in Simulink is found in the accumulators where various counters, registers and triggers had to be utilized to produce the somewhat unusual periodic summing function. Figure 3.3.3 illustrates the non-IQ sampling accumulator structure in more detail; essentially what happens is each discrete time instance the current sample is summed with either the previous sample or zero if the 0-63 counter just reset. Each counter reset also updates the output value with the current, usually maximum accumulated value. Finally, a “Data Ready” 1-sample pulse is output on counter reset as well, though that signal is only used in the hardware implementations discussed later in this chapter. An example of nominal IQ Demodulator operation with perfect phase-matching is provided Figure 3.2.2 located in Section 3.2.

PID Controller

The software model version of the PID Controller block can be seen in Figure 3.3.4. The structure of this block differs significantly from traditional PID controller structure, but is much closer to T. Schilcher’s PID structure. The block is largely a translation of Schilcher’s PID equations notably occurring with paths for both I and Q signals and with latch-type blocks at key points in the signal path to rectify timing initialization errors in the model. These “latches” prevent the block from providing meaningful output until a certain number of simulation time steps have been completed; this was to address a timing mismatch that arose from the PID controller producing an incorrect drive signal before the demodulator had provided its first valid output after startup.

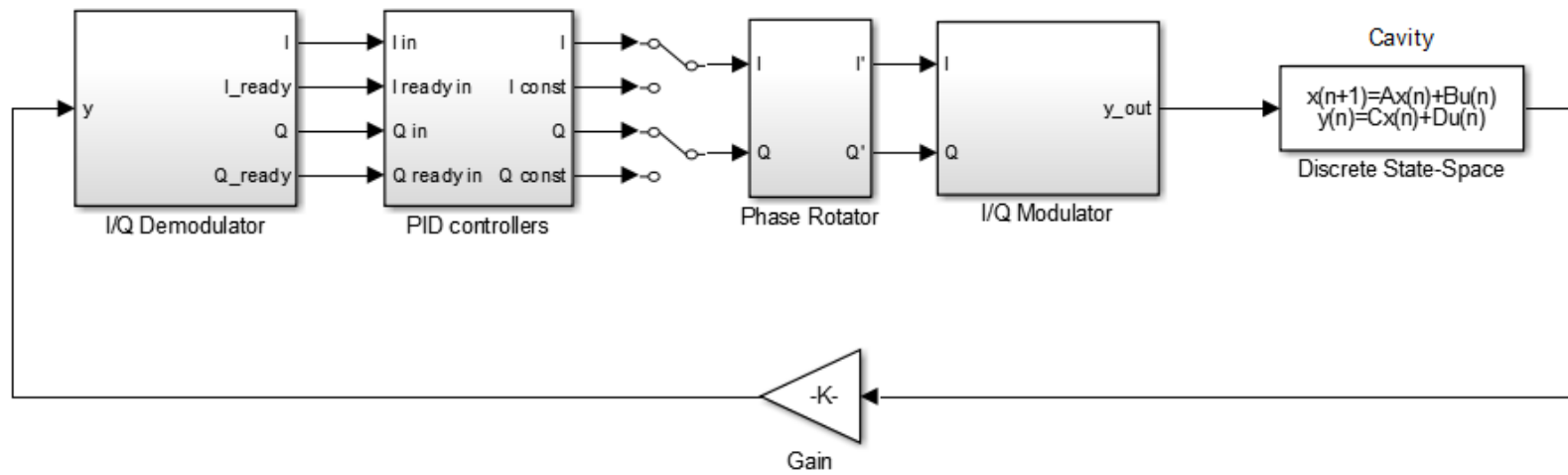


Figure 3.3.1 Overview of Floating Point Model

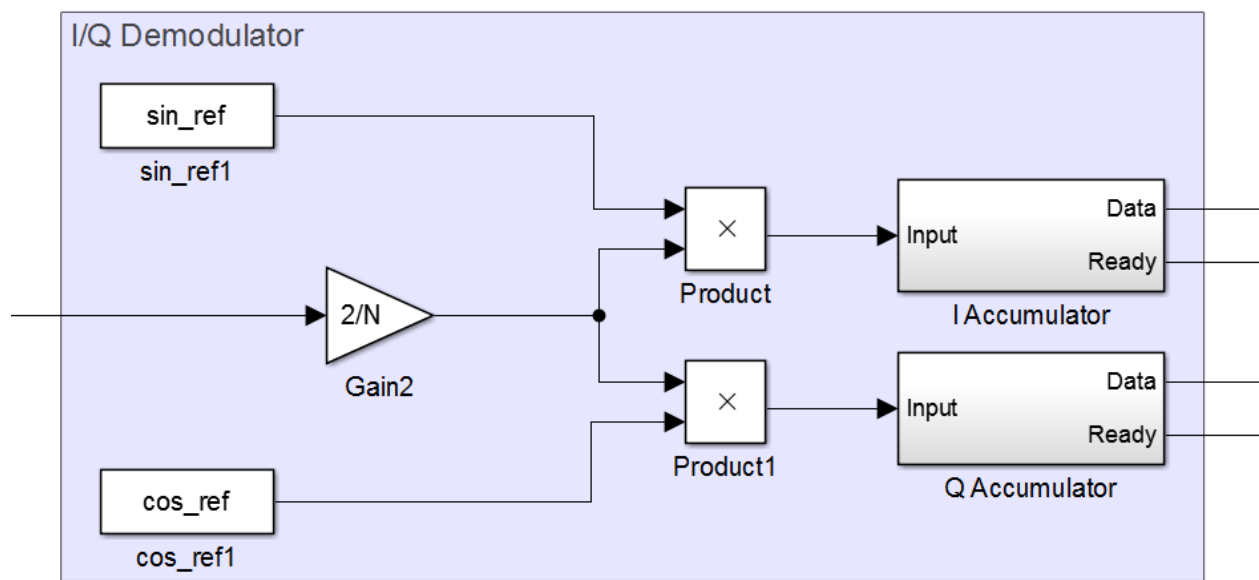


Figure 3.3.2 Floating Point Model IQ Demodulator Subsystem

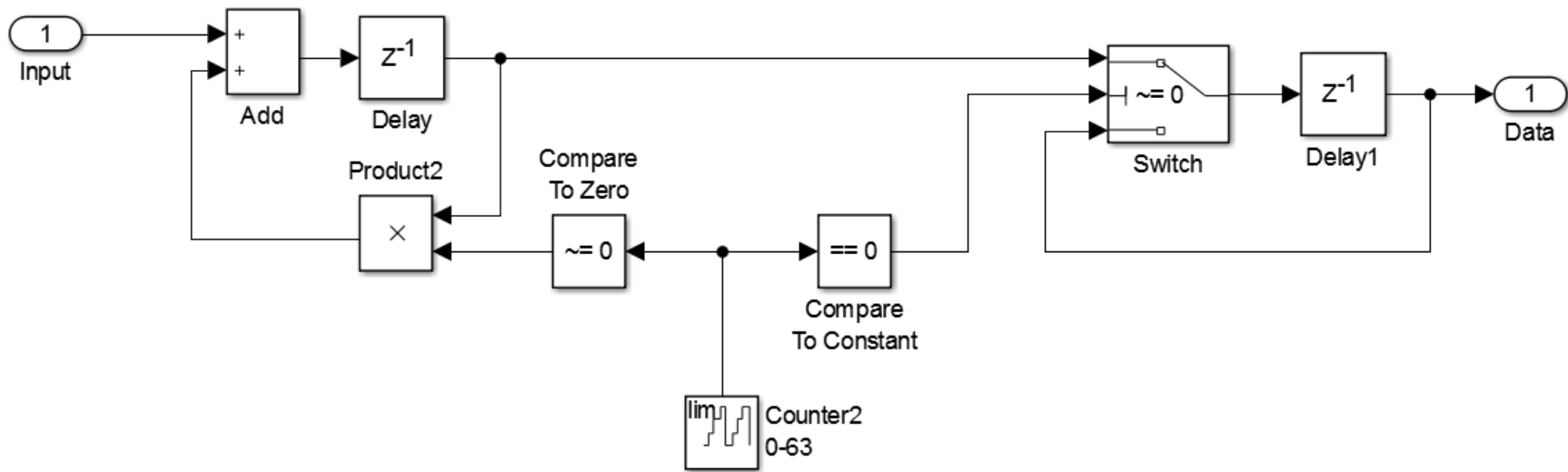


Figure 3.3.3 Floating Point Model Non-IQ Sampling Accumulator

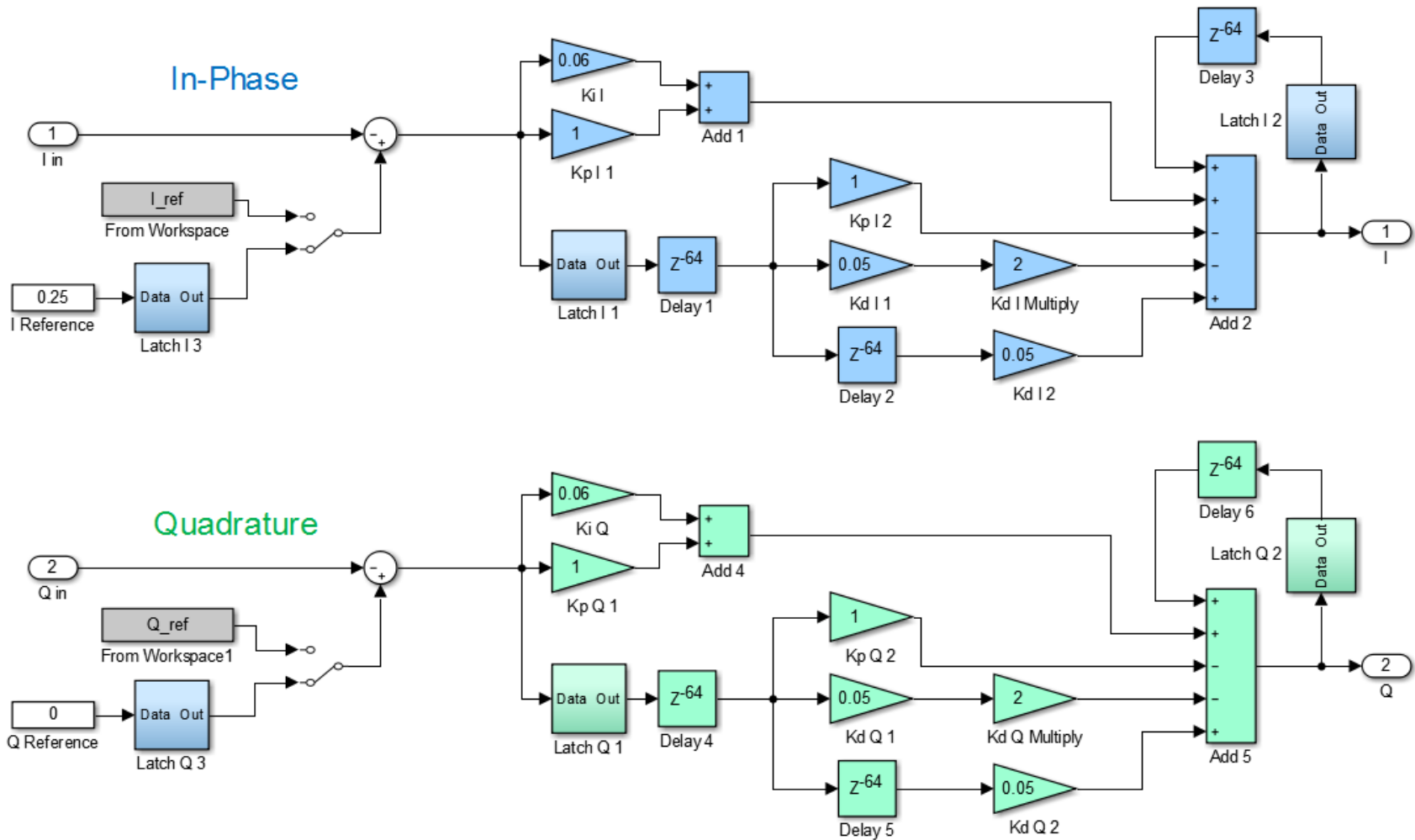


Figure 3.3.4 Floating Point Model PID Controller Subsystem

Phase Rotator

The floating point software model of the SNS ring control system phase rotator is functionally and structurally identical to the theoretical expression seen previously in Figure 3.2.4.

IQ Modulator

The floating point model IQ modulator is also fairly straightforward where each sample of I and Q are multiplied by the appropriate sinusoid, the products added together, and the sum output to the DAC - in this case, the cavity block. The implemented IQ Demodulator structure is shown in Figure 3.3.5.

Ferrite-Loaded RF Cavity

Once the drive signal is modulated into the RF regime it's fed to the cavity block which is represented, quite simply, by a Simulink "Discrete State-Space" block with the appropriate state matrices. However, the initial state matrices derived from the cavity differential equations had to first be discretized using MATLAB's c2d (continuous-to-discrete) function for linear systems. Once the discretized state matrices were obtained the appropriate variables were referenced in the discrete state-space block. Comparison between continuous and discrete state-space expressions of the system can be found in Chapter 4.

PID Tuning Script

Once the floating point controller model was constructed and performance verification began, it was soon evident that finding PID gain values that optimized for minimal rise-time, overshoot, and ringing would be quite the task due to complicated trends between gain parameters as they varied. So in lieu of manually setting, simulating, analyzing, and recording hundreds of test-cases the PID Tuning Script was written.

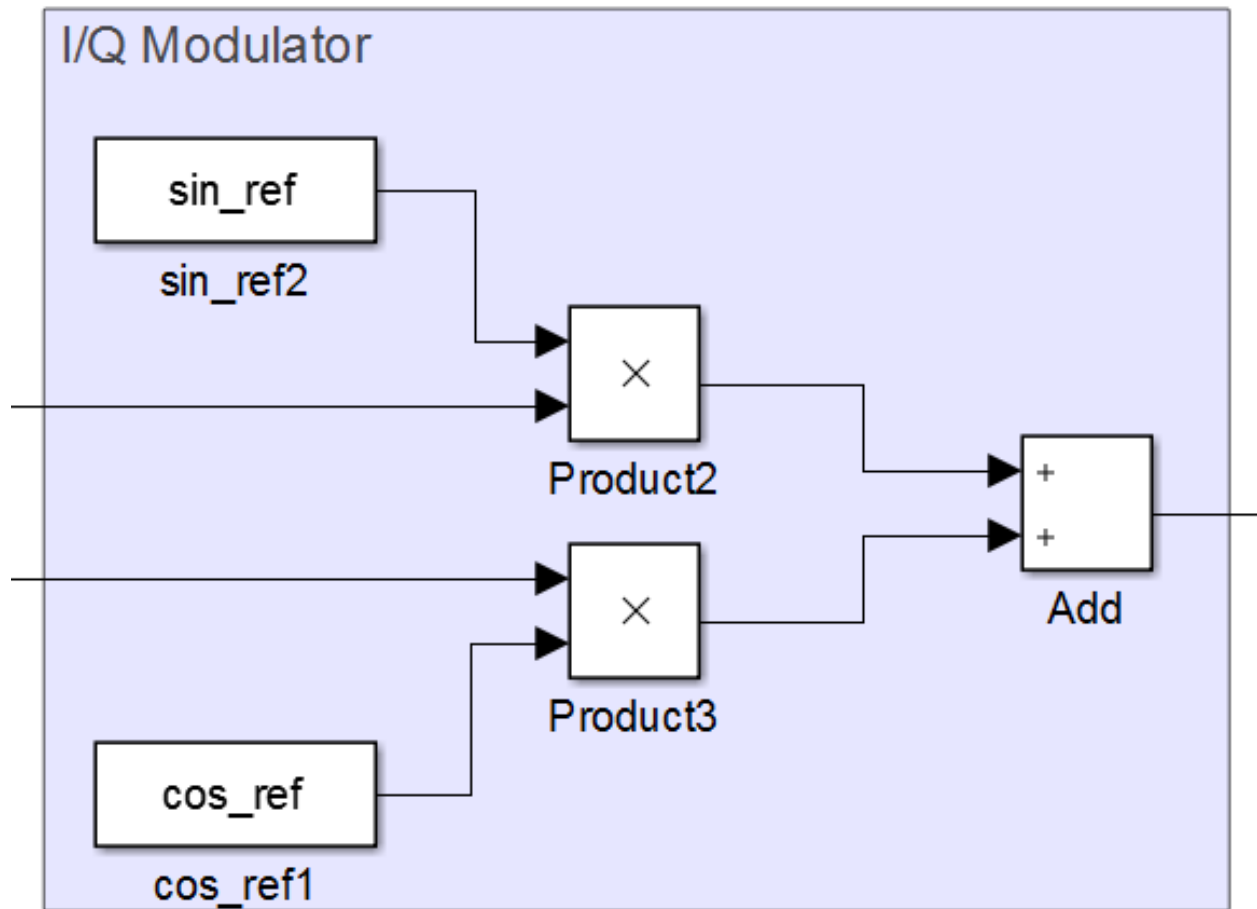


Figure 3.3.5 Floating Point Model IQ Modulator

The PID Tuning Script is a MATLAB script that iteratively approaches optimal gain parameters for the SNS Accumulator Ring LLRF floating point control system model. This is accomplished by first establishing a range of proportional, integral, and derivative gain values to test over, and the number of each gain type settings to test within that specified range. Once the parameters are initialized, the script begins analyzing model performance across a series of gain values before selecting the particular run that best fits the current definition of optimal and outputting the optimal parameters to console.

This process is described broadly by Figure 3.3.6 which describes how derivative gain is the highest-level parameter, followed by integral gain and finally proportional gain. However, once a particular pass through a range of gains has been completed and an optimal gain is selected the script doesn't just increment the parent loop and continue with another run of the child loop. Instead, the script forms another region around that newly-found pseudo-optimal point to increase the granularity of the test without vastly increasing simulation time. A visual explanation of this variable-granularity abstraction is presented in Figure 3.3.7.

Fixed Point Model

Once the floating point model was complete, the process to create an intermediate model began. The fixed point model was necessary to investigate and verify that model operation was still valid with a particular fixed point data format, with the intention of using that data format in the eventual hardware expression of the Ring LLRF cavity controller. An overview of fixed point format concepts will be helpful before discussing the particulars of the process further.

A fixed point format binary number has two sections: The Integer and the Fraction. The integer is the bits to the left of the binary point – representing the whole-number portion of the

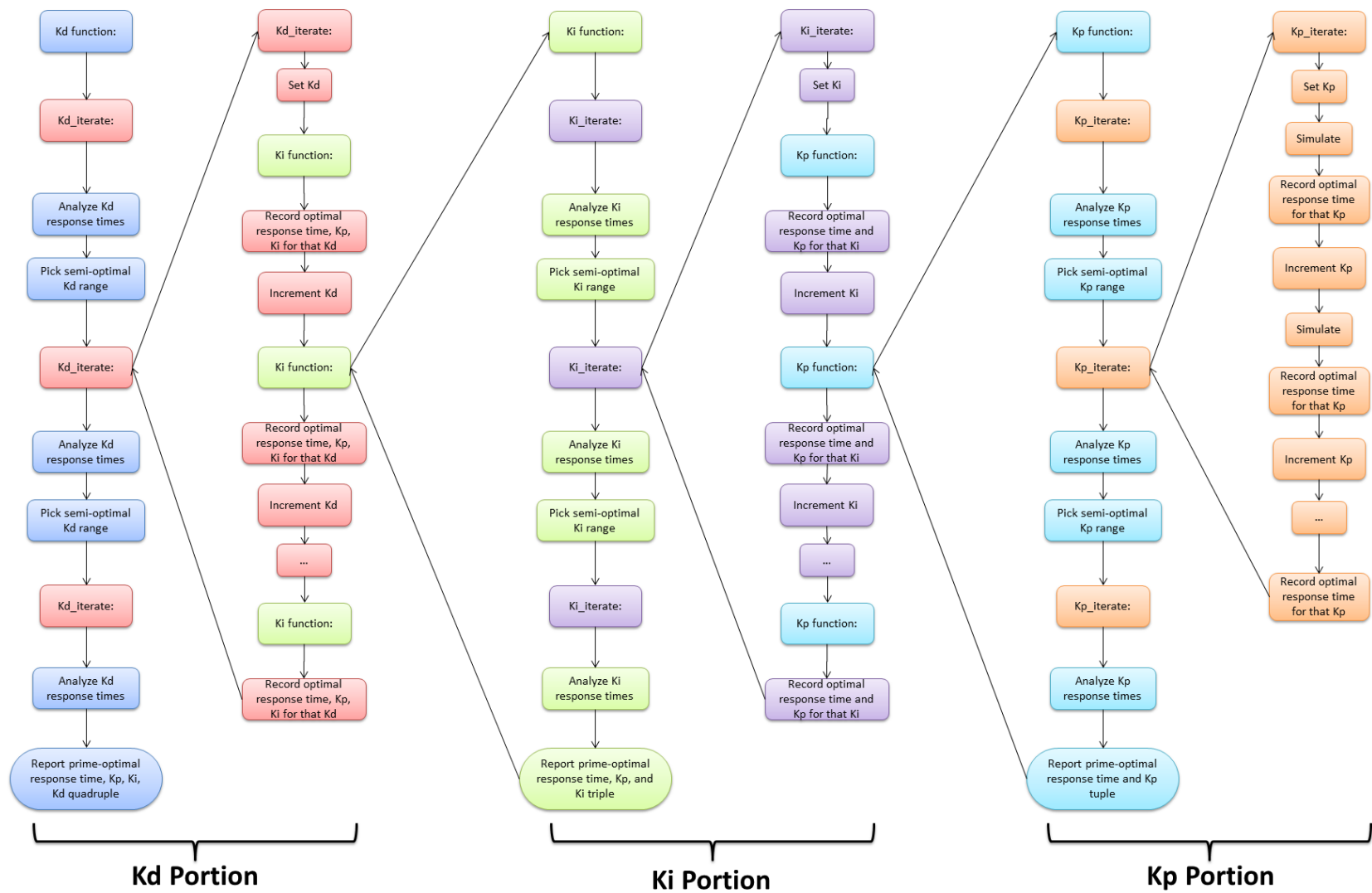
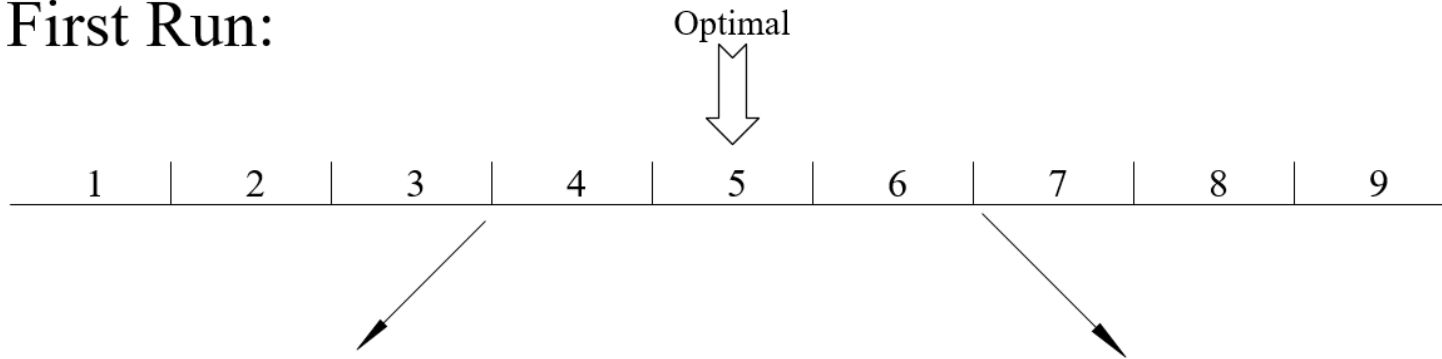
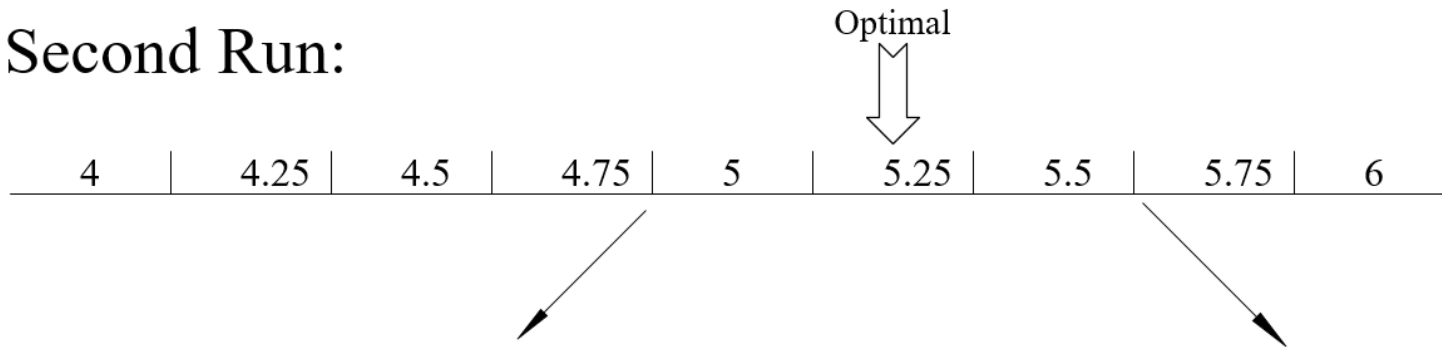


Figure 3.3.6 PID Tuning Script Flowchart

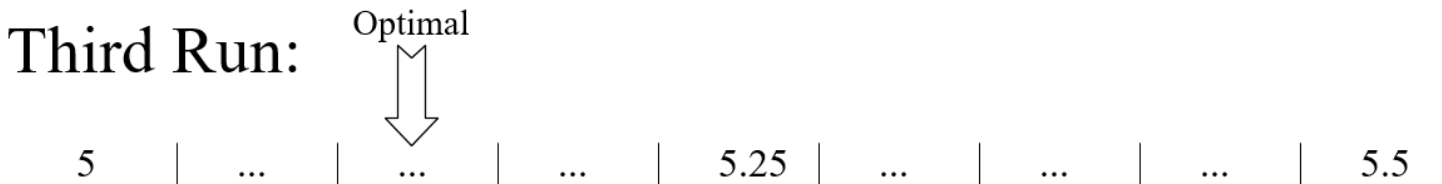
First Run:



Second Run:



Third Run:



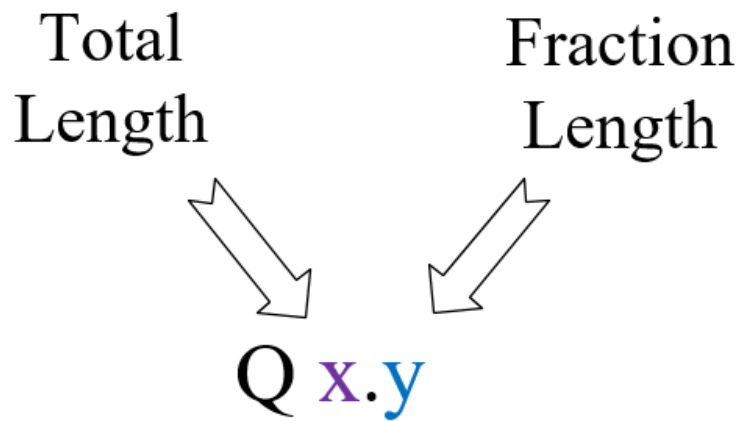
•
•
•

Figure 3.3.7 Example of PID Tuning Script Granularity Tunin

value – while the fraction is the bits to the right of the binary point. Shown in Figure 3.3.8, the notation for a particular fixed point format follows a $Q_{x.y}$ structure, where x is the number of total bits and y is the number of fraction bits. As with any finitely-represented number format, there are maximum value and precision limits; Integer-length determines maximum value and fraction-length determines maximum precision for fixed point format numbers.

The first step in selecting an appropriate fixed point data format was to establish feasible limits for signal range and hardware bit-length. It was known from the outset that the ADC and DAC responsible for hardware signal-path input and output ranged from -1 V to +1 V, that the RF cavities operated up to 10 kV, that the selected FPGAs had 18 bit by 18 bit multipliers, and that the aforementioned DAC and ADC data format was 14-bit sign-and-magnitude binary. Armed with this knowledge, a design basis for the model was established: total length would be 18 bits wherever multiplications were used in the model, with the input being sign-extended from 14 bits to 18 bits and the output being truncated from 18 bits back to 14 bits for compatibility with the DAC.

However, the process didn't quite follow such a direct path. An astute observer might notice that trying to span from millivolts within the controller to kilovolts for the cavity with one data format might result in some slight incompatibilities. To accommodate this, a $Q_{32.16}$ format was chosen in initial conversion to verify that fixed-point arithmetic would function at all. Once this was confirmed, an integer-heavy format for the cavity side of the model remained, while bits were gradually trimmed from the controller signals. So the process of trimming and verifying continued until it was concluded that a 1-bit integer length was sufficient, while a fraction length of 13 bits offered sufficient precision to represent controller operation without undue error. In the



Q 16.6 ↔ 0000 0000 00.00 0000

Q 11.1 ↔ 0000 0000 00.0

Q 11.6 ↔ 000 00.00 0000

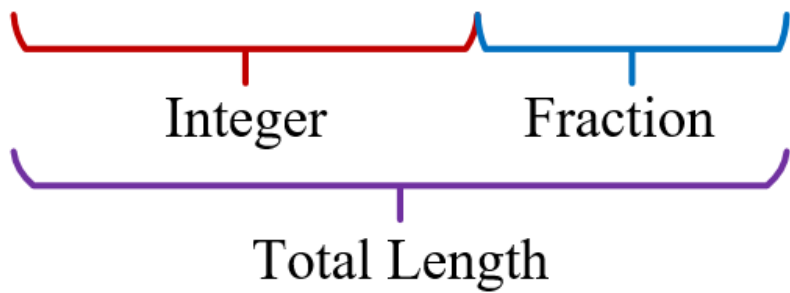


Figure 3.3.8 Fixed Point Format Notation Example

interest of computational efficiency, this Q14.13 format was retained for the remainder of development, seeing its only revision in actual hardware where the integer was “sign-extended” to 18 total bits resulting in a Q18.13 format.

So despite conventions that may seem odd in hindsight, the choice of fixed point data format was sufficient for modeling system operation and successfully supported tests on the actual system. Results and discussion thereof are found in Chapter 4.

System Generator Model

The System Generator Model serves as a bridge between software and hardware expressions of the designed controller. It was critical for the software controller models to be verified in a hardware simulation, and though this process could have been completed through painstaking traditional HDL simulator techniques, hardware model verification was instead enabled by System Generator functions. Xilinx System Generator for DSP (SysGen) is a MATLAB library that allows certain elements in a Simulink model to interface with the Xilinx Vivado VHDL and Verilog simulator, so a VHDL controller expression could be effortlessly integrated into a traditional Simulink model.

However, unlike previous designs, the VHDL black-box controller block doesn’t present itself as several distinct subsystems that could be actively swapped into the floating or fixed-point models; instead, the entire control path from ADC to DAC is encapsulated in one top-level controller block. Thus, a more significant conversion procedure occurs where all non-cavity blocks are removed and replaced with a single black-box block. This was done so the simulated VHDL design would be as identical as possible to the code configured onto the FPGA in the implemented hardware. The layout of the System Generator Model can be seen in Figure 3.3.9.

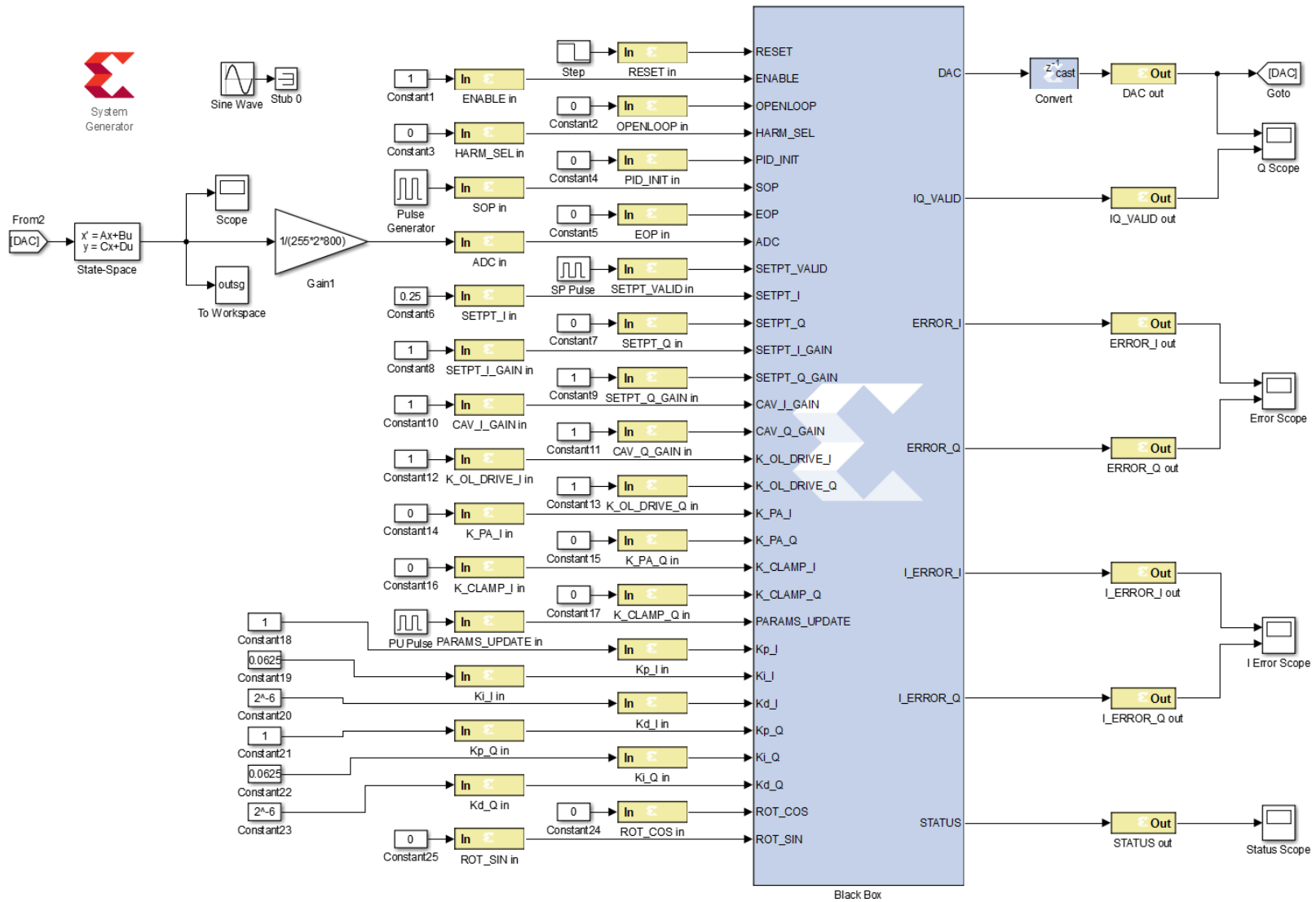


Figure 3.3.9 Overview of System Generator Model

Figure 3.3.9 provides an overview that may inform an astute reader of several key design elements that influence the inner structure of the SysGen Model. First, it should be quite obvious that the previous software-only models have far fewer inputs than the SysGen Model. These additional inputs reveal that the VHDL implementation of the Ring Cavity LLRF Controller possesses expanded control of existing controller subsystems alongside additional subsystems altogether. To understand these new features and improvements it's easiest to break the top-level black box down into its constituents and discuss those individually.

As seen in the exploded view of the VHDL controller block in Figure 3.3.10, the first component in the signal path is once again the IQ Demodulator. Yet this time around it has 3 extra inputs of interest – SOP, EOP, and HARM_SEL – and 2 extra outputs of interest: ERR_FLAG and DATA_VALID. Beginning with the new inputs, SOP tells the demodulator when Start of Pulse has occurred so the address-counter for sine and cosine lookup tables can begin incrementing, and the output can be activated. EOP tells the demodulator when End of Pulse has occurred so the output can be disabled and the block reset to its initial state. HARM_SEL selects the desired operating harmonic by having the sine and cosine address counter increment through every entry in the lookup tables for first harmonic operation, or by having the address counter increment skip every other address, effectively doubling the frequency for second harmonic IQ demodulation. Next, the first new output ERR_FLAG is a one-bit signal that latches high if the harmonic select bit changes state while the output is active. Finally, DATA_READY is a bit that goes high once per turn when the demodulator outputs a new, ready-to-be-processed data value; DATA_READY is used in subsequent internal triggering.

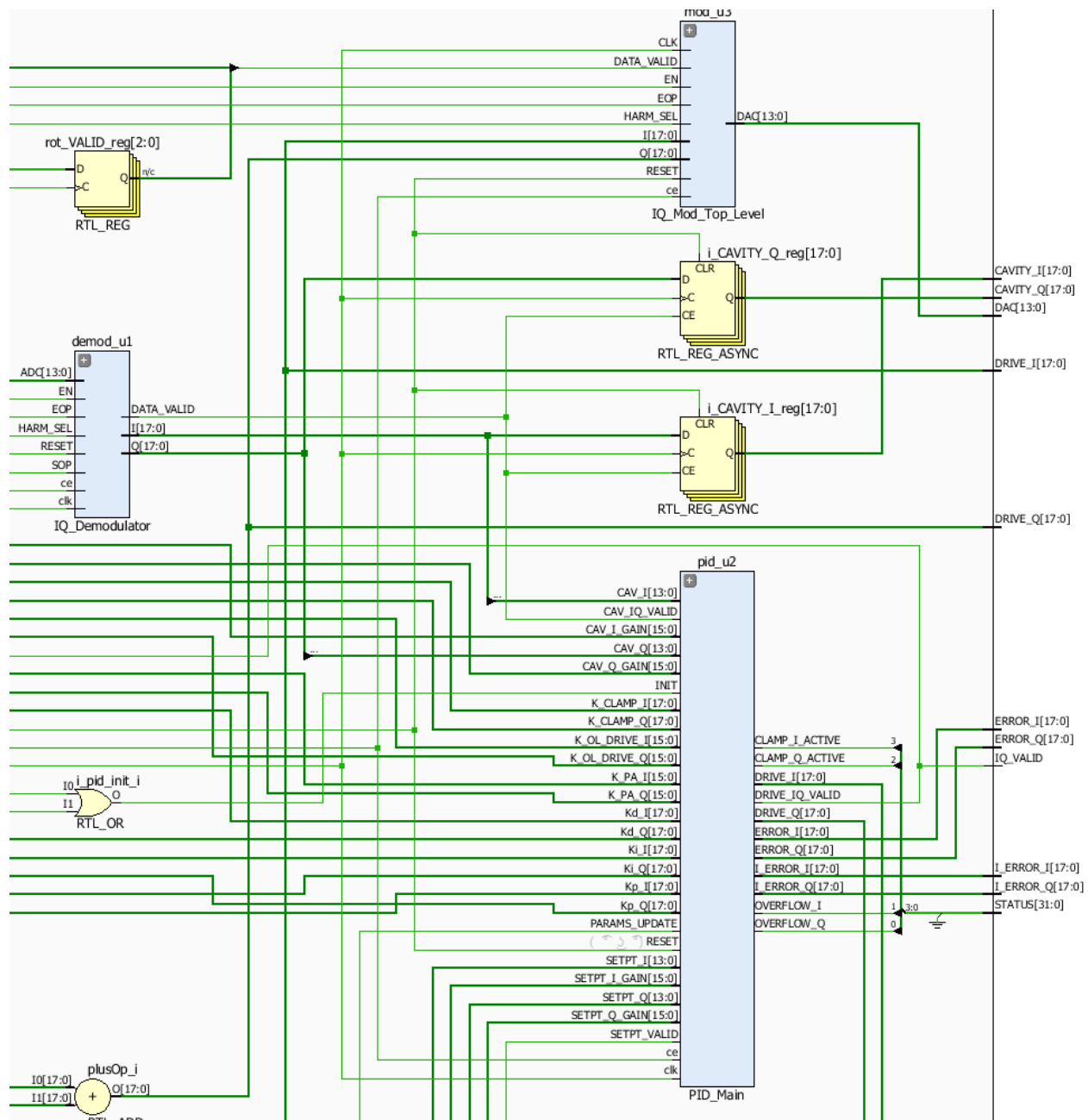


Figure 3.3.10 SysGen Model VHDL Black Box Exploded View

The next component in the VHDL controller signal path is the PID controller. This block is heavily based on the original PID control hardware design and has a myriad of new signals to continue support for all the old features. Table 3.1 lists, disambiguates the abbreviations, and summarizes the function of each of the signals. The SysGen Model's phase rotator is technically incorporated into the PID VHDL code module as well, and the VHDL code for this block was developed by Breeding [3].

Finally, the IQ Modulator block sees the addition of HARM_SEL and DATA_VALID inputs, with HARM-SEL selecting between 1st or 2nd harmonic IQ modulation and DATA_VALID acting as a trigger flag to maintain phase synchronization with the other blocks.

Table 3.1 List of PID_Main Inputs

Signal	Length	Expanded Name	Note
CAV_I	14	Cavity I Input	
CAV_IQ_VALID	1	Cavity IQ Valid bit	Triggered with new value
CAV_I_GAIN	16	Cavity I Gain	Legacy Feature
CAV_Q	14	Cavity Q Input	
CAV_Q_GAIN	16	Cavity Q Gain	Legacy Feature
INIT	1	Initialization bit	Initial internal process trigger
K_CLAMP_I	18	I Clamp Gain	Legacy Feature
K_CLAMP_Q	18	Q Clamp Gain	Legacy Feature
K_OL_DRIVE_I	16	I Open-Loop Drive Gain	I path gain if open loop is active
K_OL_DRIVE_Q	16	Q Open-Loop Drive Gain	Q path gain if open loop is active
K_PA_I	16	I Power Amplifier Gain	Legacy Feature
K_PA_Q	16	Q Power Amplifier Gain	Legacy Feature
Kd_I	18	I Derivative Gain	
Kd_Q	18	Q Derivative Gain	
Ki_I	18	I Integral Gain	
Ki_Q	18	Q Integral Gain	
Kp_I	18	I Proportional Gain	
Kp_Q	18	Q Proportional Gain	
PARAMS_UPDATE	1	Parameter Update bit	Must be triggered to latch new gains
RESET	1	Reset bit	
SETPT_I	14	I Setpoint	
SETPT_I_GAIN	16	I Setpoint Gain	Legacy Feature
SETPT_Q	14	Q Setpoint	
SETPT_Q_GAIN	16	Q Setpoint Gain	Legacy Feature
SETPT_VALID	1	Setpoint Valid bit	Must be triggered to latch new setpoint
ce	1	Clock Enable Bit	Enable when high, disable when low
clk	1	Clock	

Chapter 4

Results

4.1 Introduction

In this chapter we present, compare, and discuss selected results of the three models set forth in Chapter 3. We also present selected test cases of actual system operation on the SNS Accumulator Ring itself. The metric of primary interest is the capability of a given controller to drive the cavity gap voltage to the set point for given parameters. Other metrics such as though rise time, settling time, and overshoot are also used for objective comparison between different models and configurations. The following case studies are presented: Floating Point Model, Fixed Point Model, Fixed Point Model with delay, and System Generator Model. These case studies include a comparison between the Floating and Fixed Point Models, and a comparison of the System Generator Model with the Fixed Point Model using equivalent delay. The case studies will examine a model's ability to: regulate with and without noise, exhibit expected PID behavior in various PID tuning configurations, and mimic correct behavior across various on- and off-resonance configurations. Finally, some point-by-point error comparisons will be performed.

Things to Note

The purpose of the models is primarily to provide qualitative insights into system behavior rather than quantitative verification. That is, successful system behavior is less about specific conformity to predicted values and more about general operational trends to obtain quantitative

results, the complexity of the actual system would require a model whose level of detail is beyond that of this thesis. Regardless, operational requirements are still achieved across all systems, and the following model results are still a valid measure for quality of the designed systems.

It is critically important to note that comparisons between System Generator Model results with software model results requires the inclusion of a delay in the software models to match the pipeline-delay present in the System Generator Model. This pipeline-delay is due to the natural latency present in the VHDL controller expression since the hardware implementation is pipelined. As a result, both delayed and non-delayed Fixed Point Model simulation results will be provided for comparison purposes with the System Generator and Floating Point Models, respectively.

In all plots the blue line is I, the green line is Q, and the orange line is cavity gap voltage.

4.2 Floating Point Model Results

Injecting Noise to Test Controller Regulation

The first test for the Floating Point Model examines how well it regulates in general open-loop and closed-loop control cases. Figure 4.2.1 shows the Floating Point Model's RF cavity output, IQ error, and IQ drive signals for open-loop general operation without noise. Since PID gains are appropriately set, it's easy to see that regulation is achieved, with rapid progression to steady-state operation. Next, Figure 4.2.2 shows the Floating Point Model's RF cavity output, IQ error, and IQ drive signals for closed-loop general operation without noise. PID gain values do not affect cavity output for open-loop as a constant, pre-specified drive signal is output from the

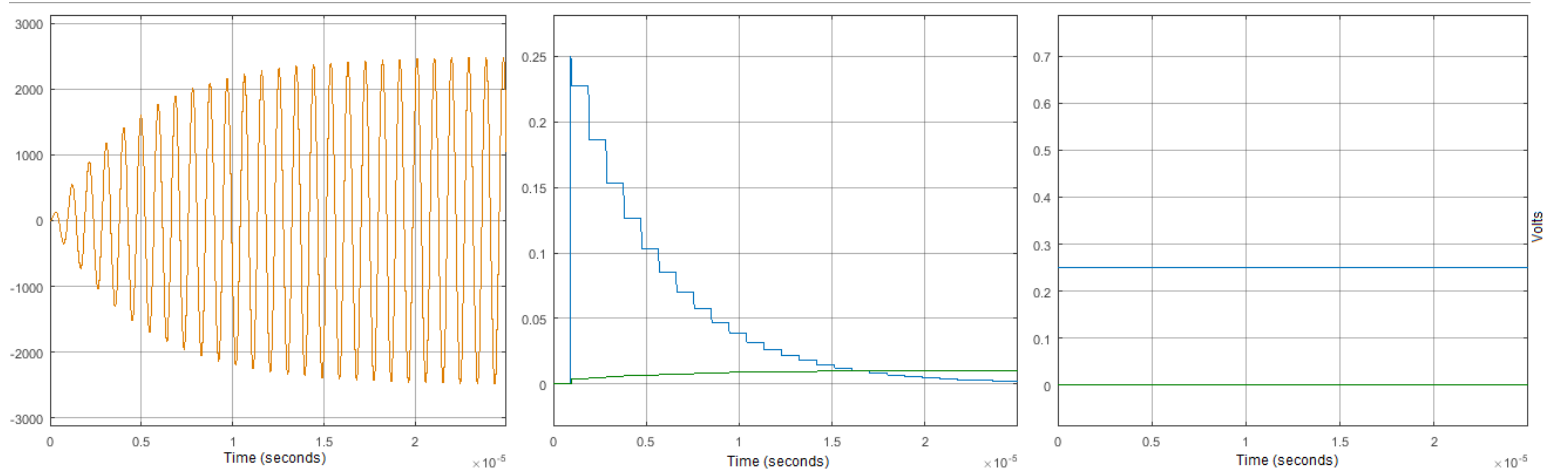


Figure 4.2.1 Floating Point Model Cavity, IQ Error, and IQ Drive without Noise in Open-Loop

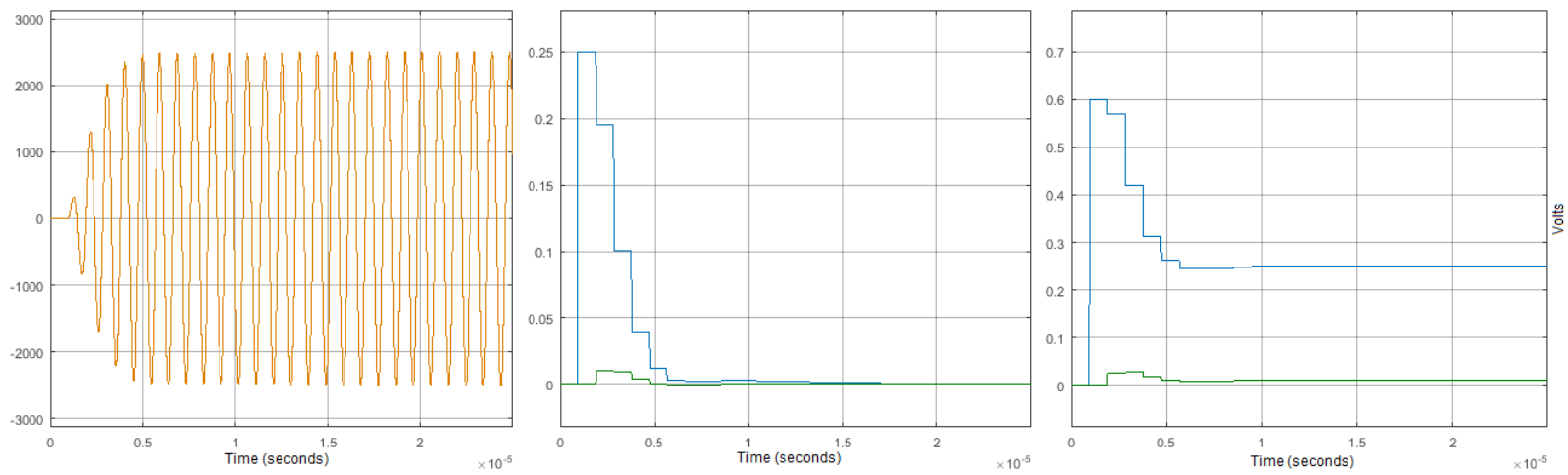


Figure 4.2.2 Floating Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop. $K_p = 2$, $K_i = 0.4$, $K_d = 0$

controller regardless of feedback values. Nevertheless, the open-loop controller configuration still regulates the RF cavity output, though more slowly than in closed-loop.

Next, Figure 4.2.3 and Figure 4.3.4 exhibit the same tests as Figure 4.2.1 and Figure 4.2.2, respectively, though this time with added white Gaussian noise. Operation with noise is practically identical to operation without noise except for slight variance in steady state values.

Changing Gain Values to Test PID Characteristics

Not only does general regulation for the Floating Point Model need to be verified, but specific behaviors of the chosen controller topology need to be investigated as well. Figure 4.2.5, Figure 4.2.6, and Figure 4.2.7 work in concert to verify that the Floating Point Model successfully demonstrates behavior typical of a PID controller in the modeled scenario. Figure 4.2.5 shows that when the Floating Point Model PID gains are configured such that only proportional gain is used that regulation does occur, but with a steady-state error. Following a similar process, Figure 4.2.6 reveals that the Floating Point Model configured with only integral gain regulates with heavy oscillation due to integrator wind-up. Finally, it can be seen in Figure 4.2.7 that combining proportional and integral gains strikes a balance between the other two cases. It should be noted here that though derivative gain cases could be included, for a second order system like the Ring RF cavity derivative gain showed miniscule performance benefit, and despite being excluded in the non-delay cases, accurate PID behavior in the Floating Point Model is still sufficiently verified.

Off-Resonance Operation to Test General Model Accuracy

The last of the standalone tests involves testing Floating Point Model operation at an ill-configured resonant frequency. Off-resonance operation in the open-loop case can be seen to

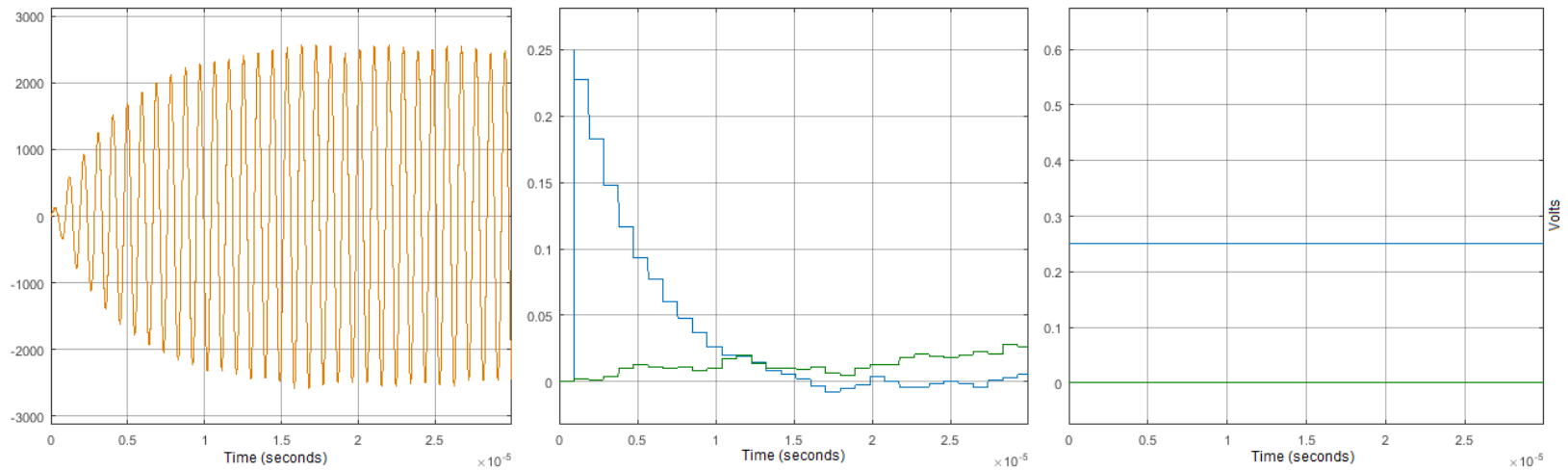


Figure 4.2.3 Floating Point Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

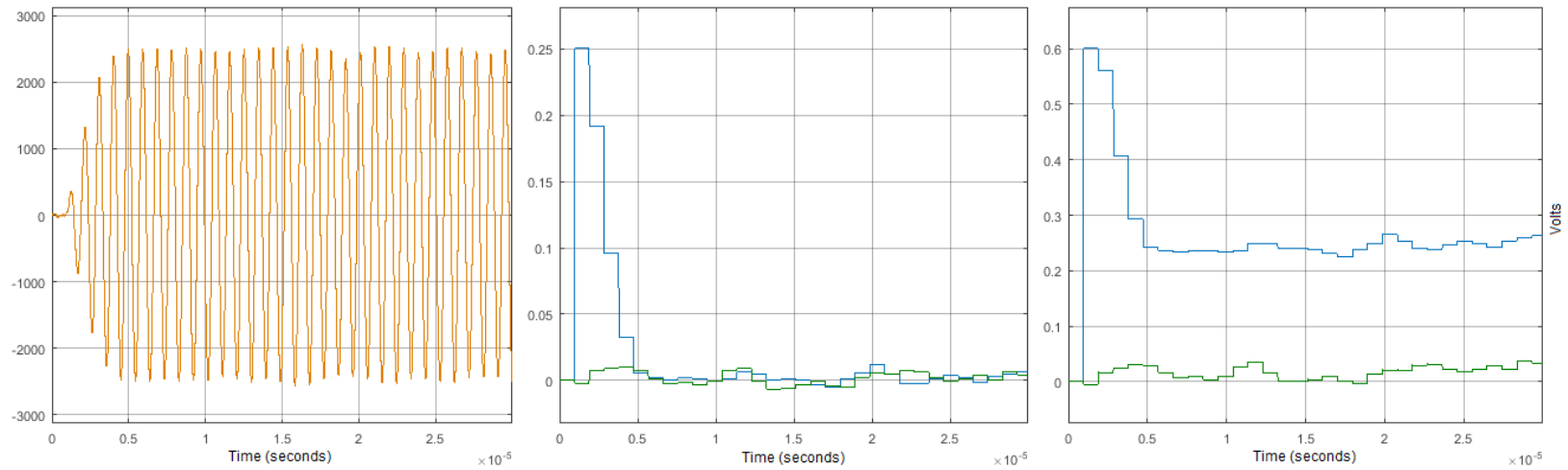


Figure 4.2.4 Floating Point Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.4$, $K_d = 0$

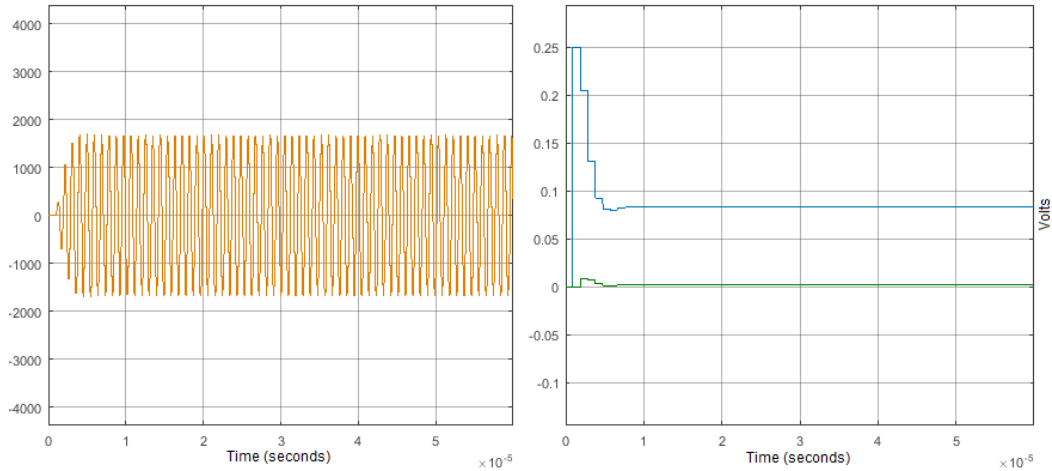


Figure 4.2.5 Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$

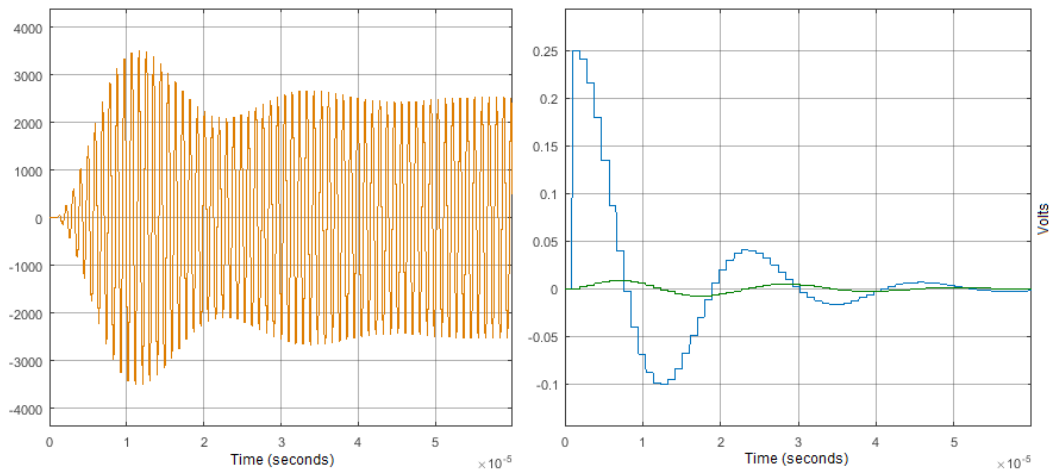


Figure 4.2.6 Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$

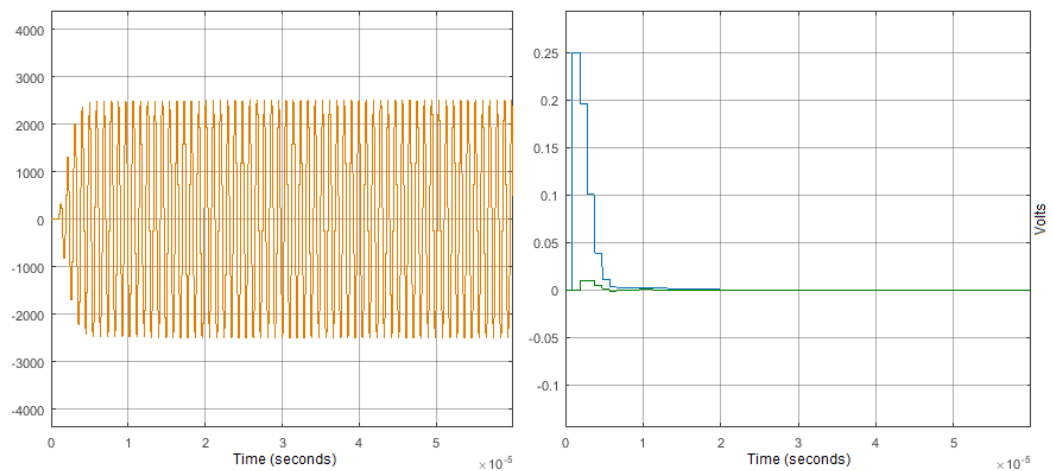


Figure 4.2.7 Floating Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$

cause attenuation and ringing in the cavity output signal for both high detune and low detune cases, seen in Figure 4.2.10 and Figure 4.2.8 respectively. Figure 4.2.9 illustrates typical on-resonance, open-loop Ring LLRF controller model operation. Figure 4.2.11, Figure 4.2.12, and Figure 4.2.13 present the closed-loop case for system frequency detuning ($K_p = 2$, $K_i = 0.7$, $K_d = 0$), though instead of attenuation and ringing, the controller successfully regulates on- and off-resonance RF cavities to a zero-error steady-state solution. Note that the IQ drive signals in the closed-loop, off-resonance cases are much more aggressive than in the on-resonance drive signal case, signifying that the controller is successfully reacting to the higher-level perturbations.

4.3 Fixed Point Model Results

Verification for the Fixed Point Model focuses more on comparison with the Floating Point Model than comparison with expected system dynamics. The same tests used in the Floating Point Model were performed using the Fixed Point Model to facilitate comparison with existing results, with Figure 4.3.1 through Figure 4.3.4 demonstrating Fixed Point Model behavior nearly identical to the Floating Point Model case regarding regulation with and without injected noise. Next, Figure 4.3.5 through Figure 4.3.7 show that the Fixed Point Model closely tracks the PID behavior exhibited in the Floating Point Model results. Finally, Figure 4.3.8 through Figure 4.3.13 deal with Fixed Point Model performance under detuning conditions, once again showing striking similarity to the Floating Point Model results. These comparisons validate that the Fixed Point Model accurately models the dynamics of the system in question, and closely tracks the same performance as the initial Floating Point Model the Fixed Point Model is based on. Section 4.4 will take these observations a step further with selected error analysis.

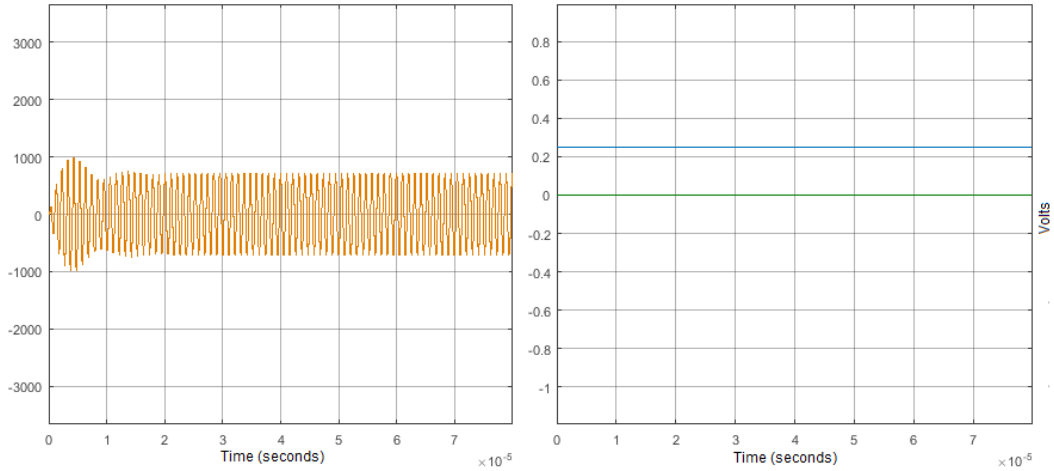


Figure 4.2.8 Floating Point Model at 952.2 KHz, Cavity and IQ Drive, Open-Loop

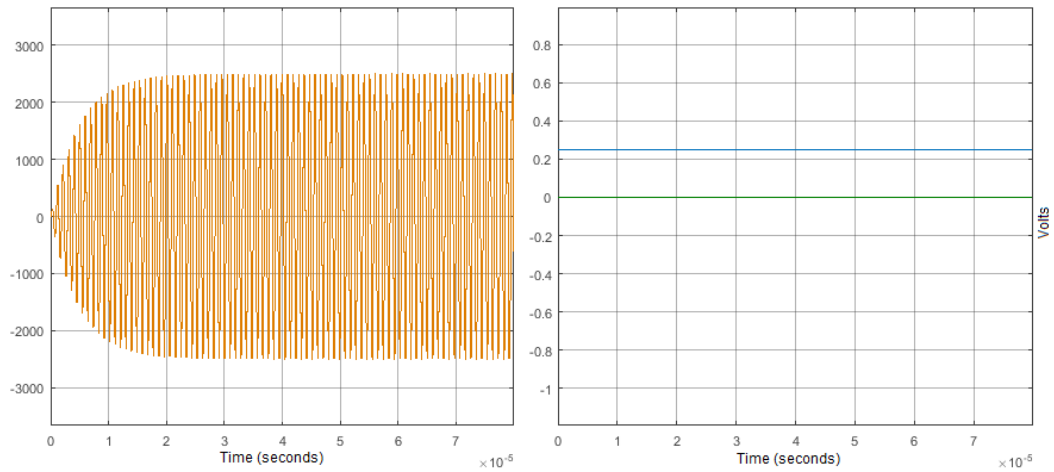


Figure 4.2.9 Floating Point Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop

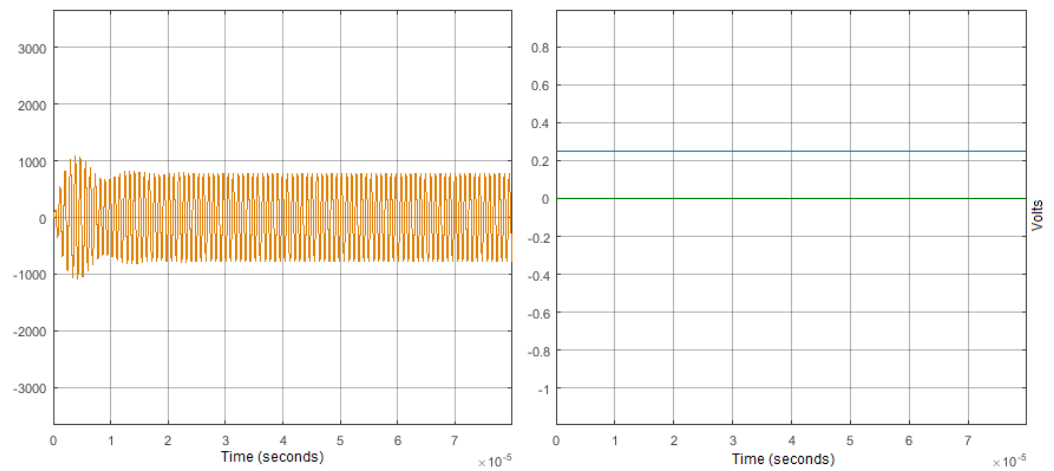


Figure 4.2.10 Floating Point Model at 1163.8 KHz, Cavity and IQ Drive, Open-Loop

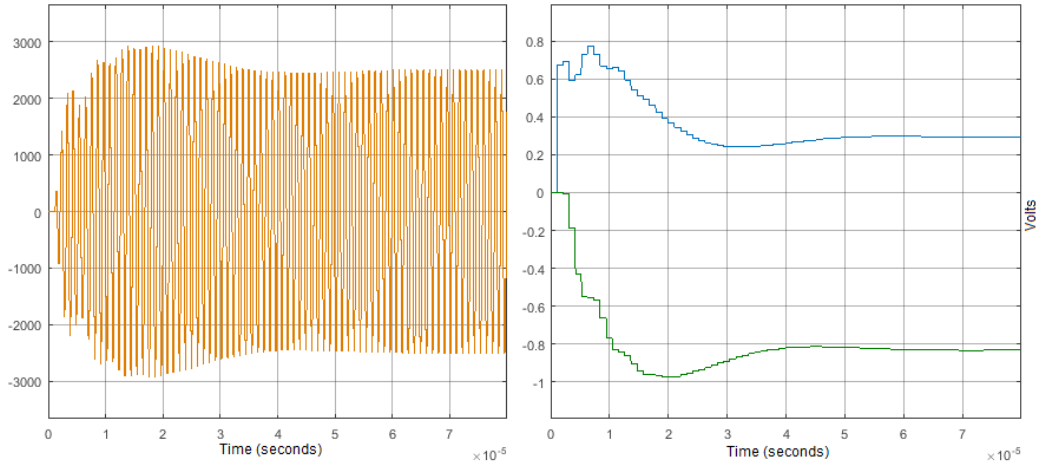


Figure 4.2.11 Floating Point Model at 952.2 KHz, Cavity and IQ Drive, Closed-Loop

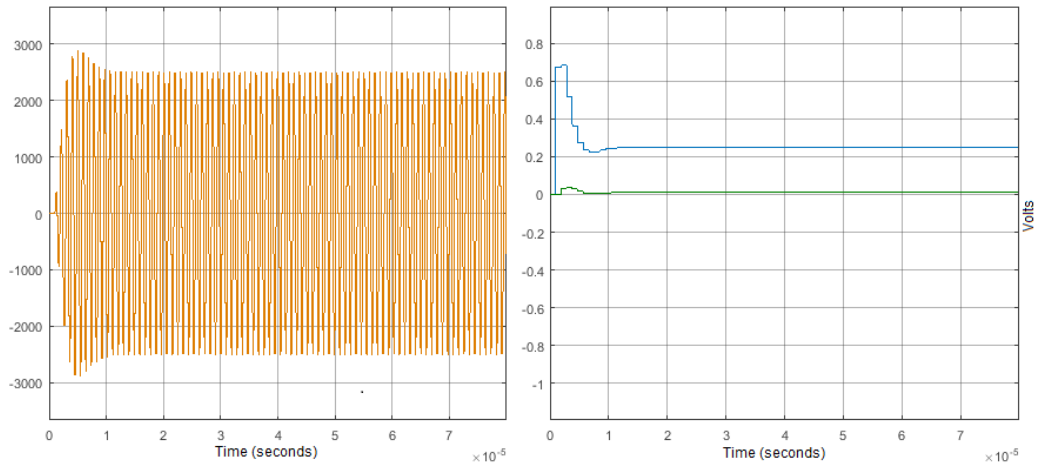


Figure 4.2.12 Floating Point Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop

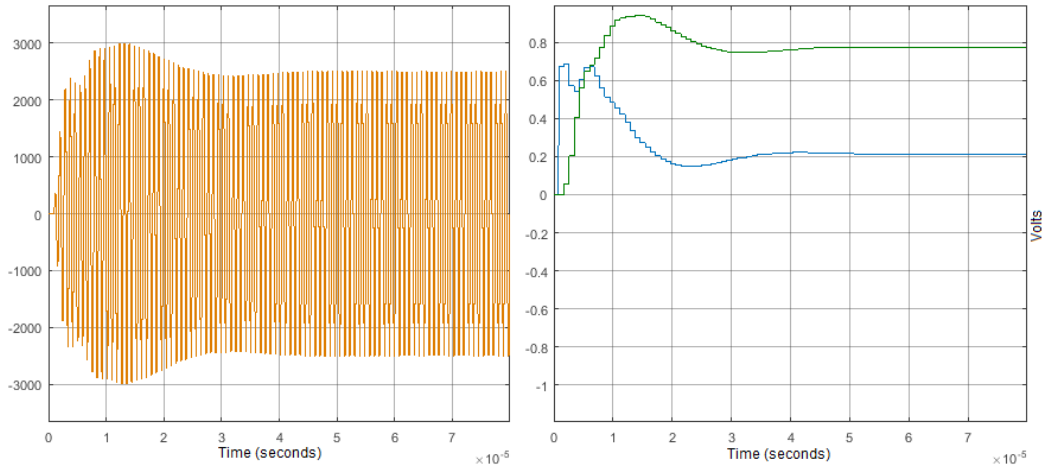


Figure 4.2.13 Floating Point Model at 1163.8 KHz, Cavity and IQ Drive, Closed-Loop

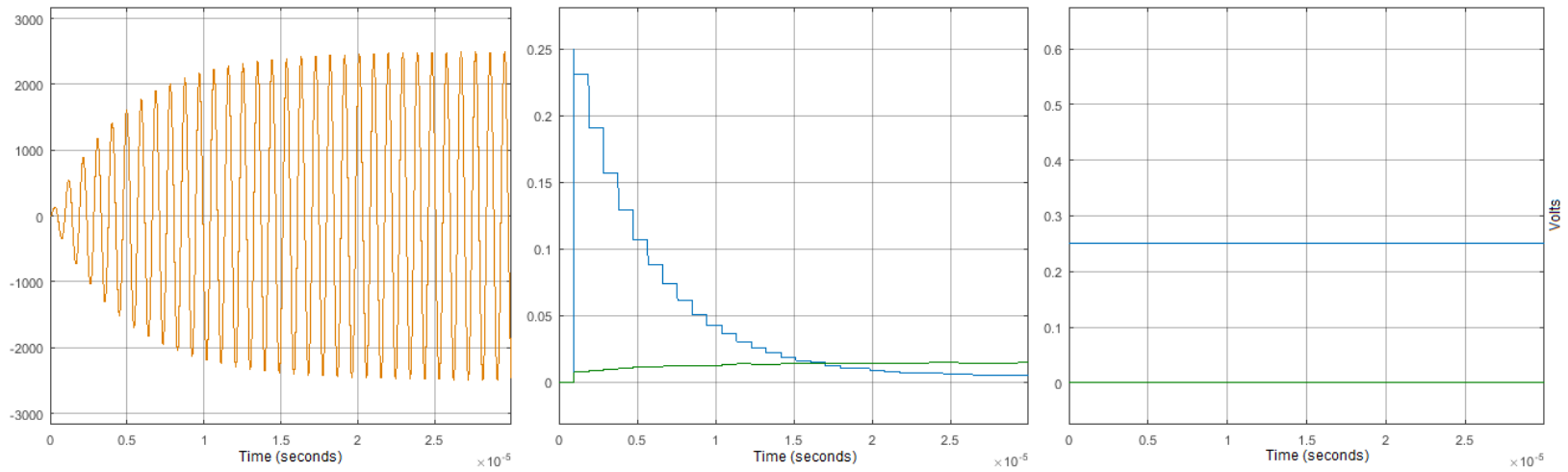


Figure 4.3.1 Fixed Point Model Cavity, IQ Error, and IQ Drive without Noise in Open-Loop

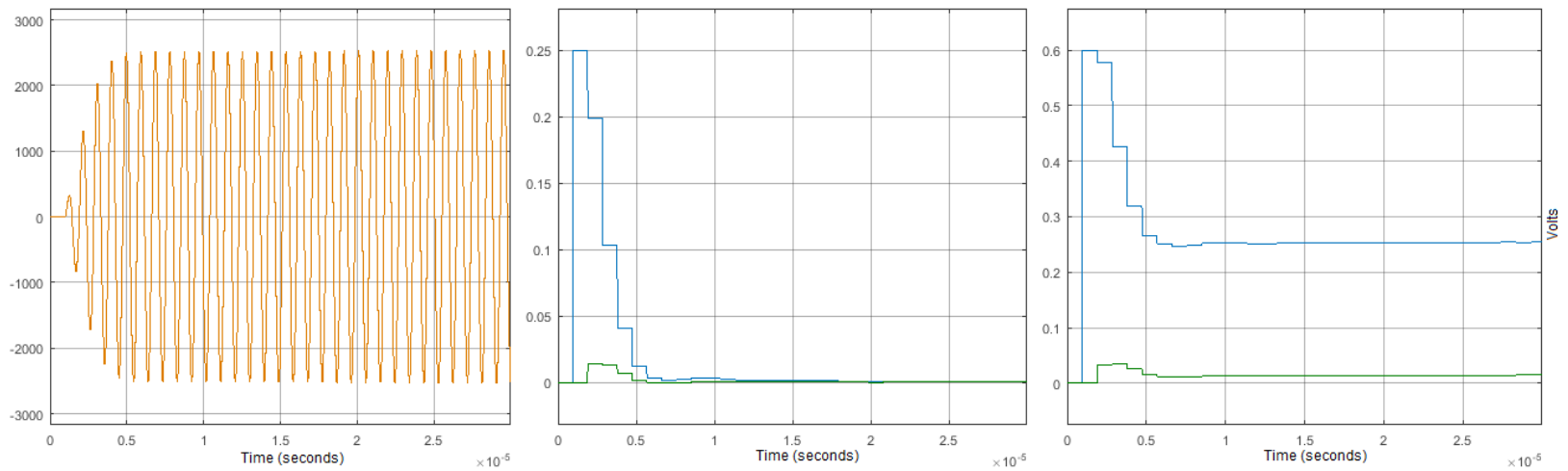


Figure 4.3.2 Fixed Point Model Cavity, IQ Error, and IQ Drive without Noise in Closed-Loop. $K_p = 2$, $K_i = 0.4$, $K_d = 0$

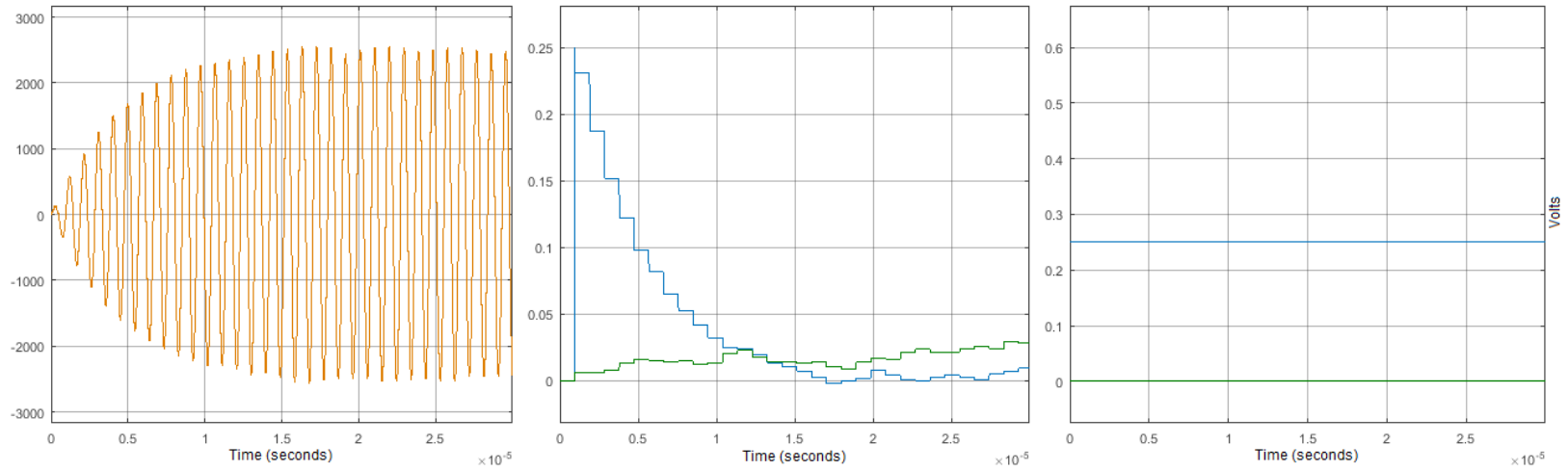


Figure 4.3.3 Fixed Point Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

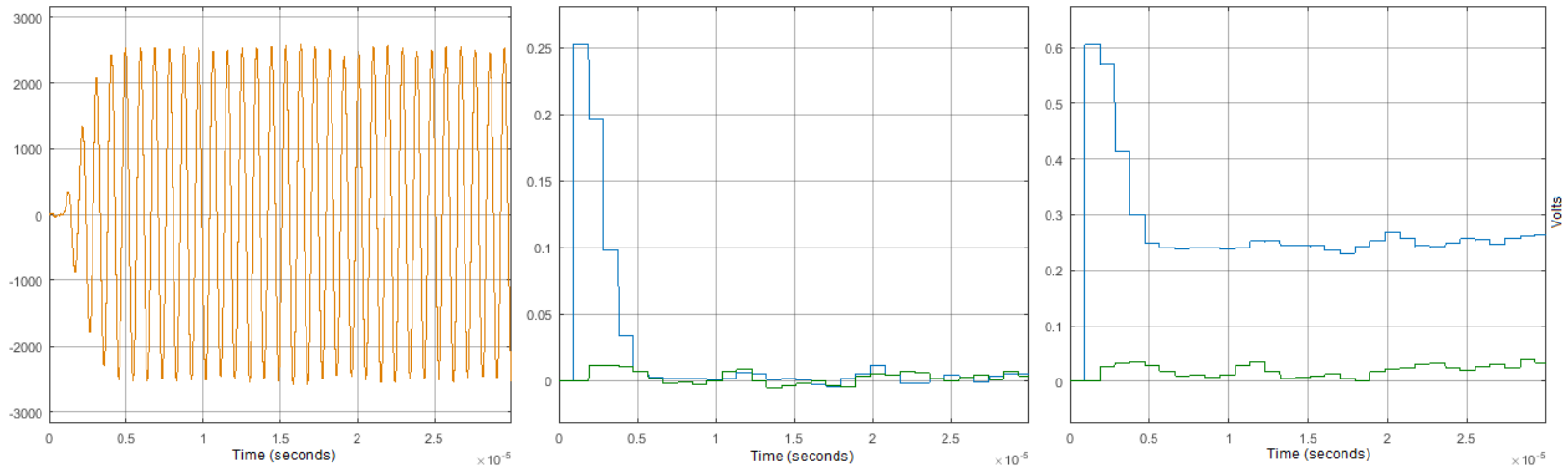


Figure 4.3.4 Fixed Point Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.4$, $K_d = 0$

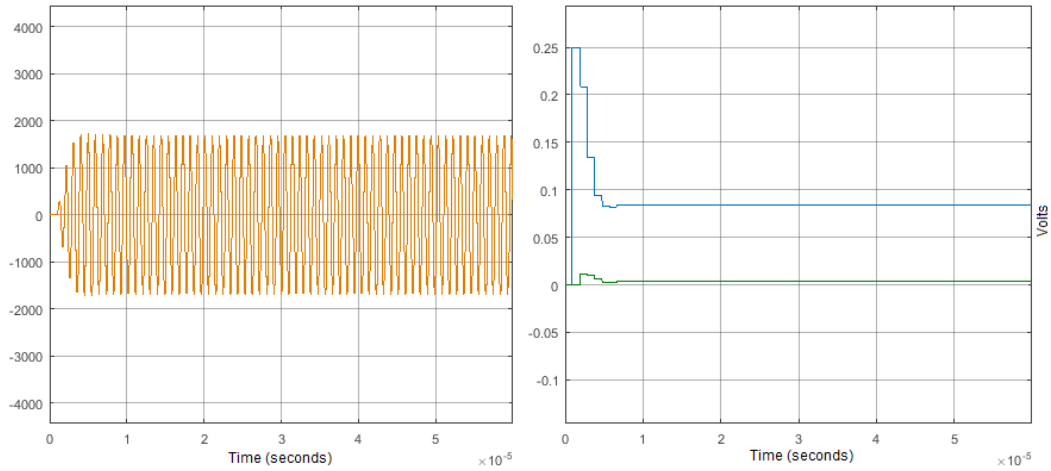


Figure 4.3.5 Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$

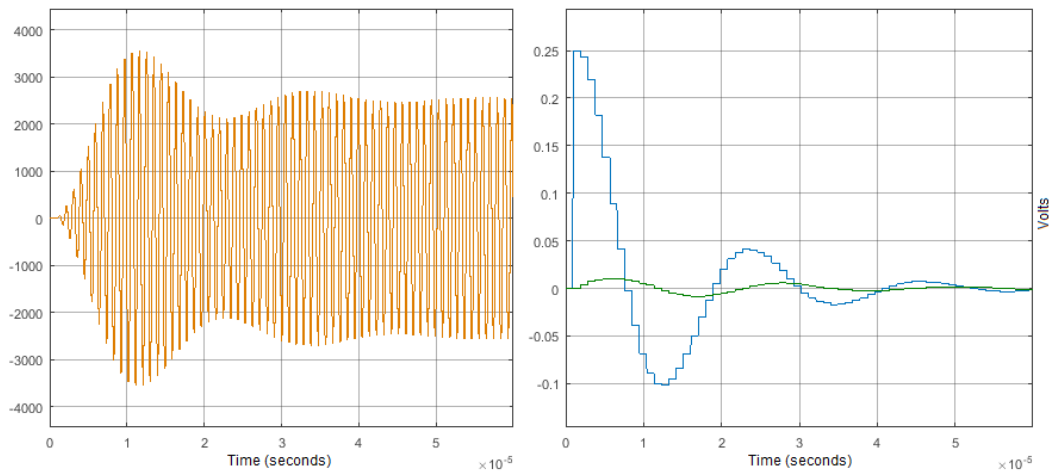


Figure 4.3.6 Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$

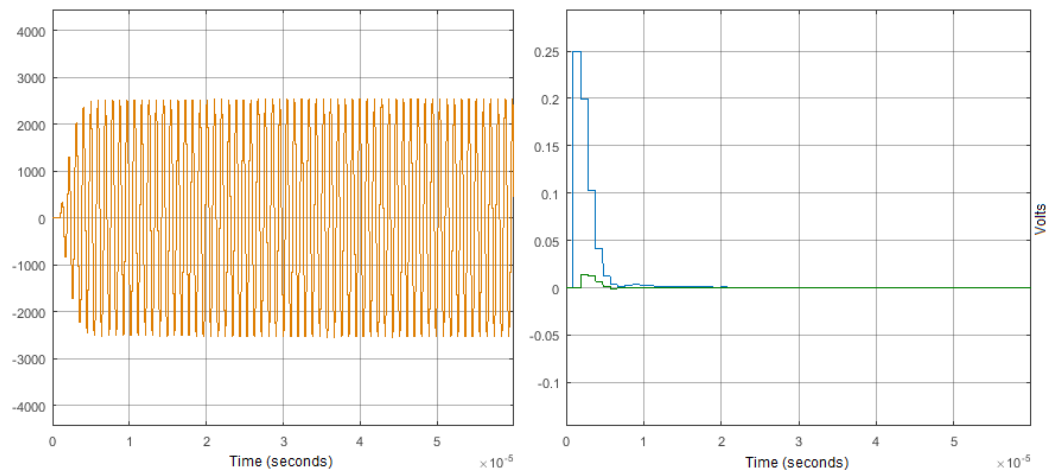


Figure 4.3.7 Fixed Point Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$

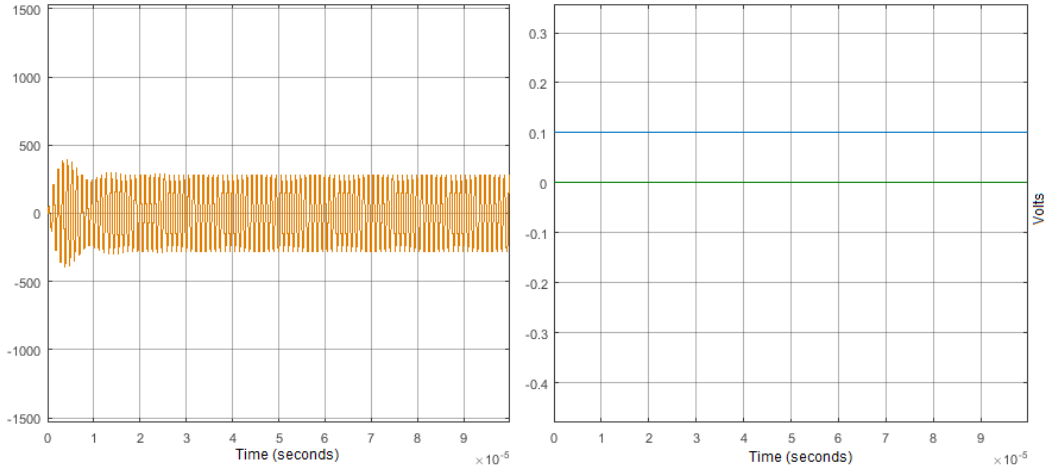


Figure 4.3.8 Fixed Point Model at 952.2 KHz, Cavity and IQ Drive, Open-Loop

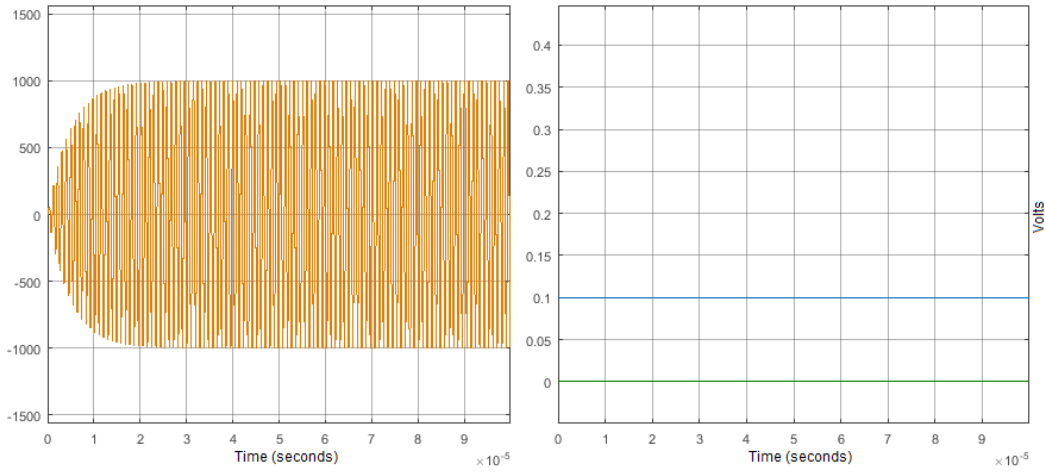


Figure 4.3.9 Fixed Point Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop

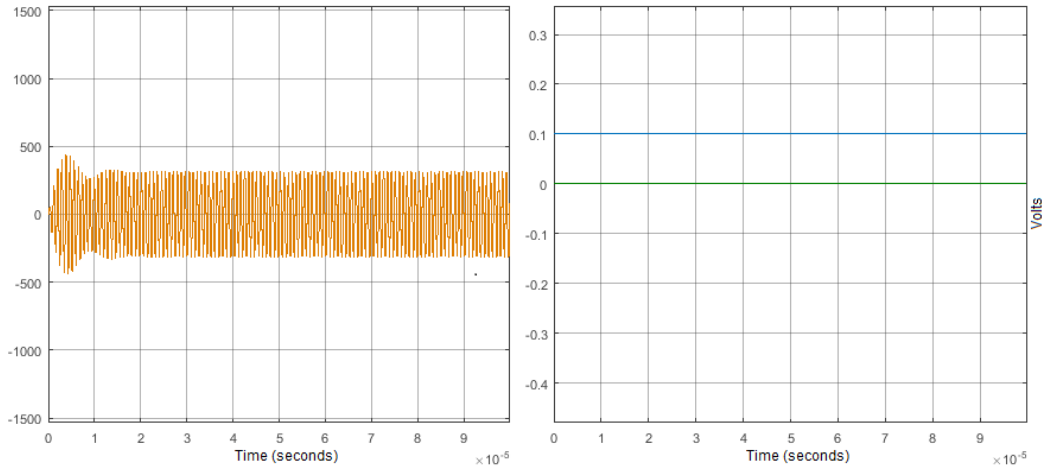


Figure 4.3.10 Fixed Point Model at 1163.8 KHz, Cavity and IQ Drive, Open-Loop

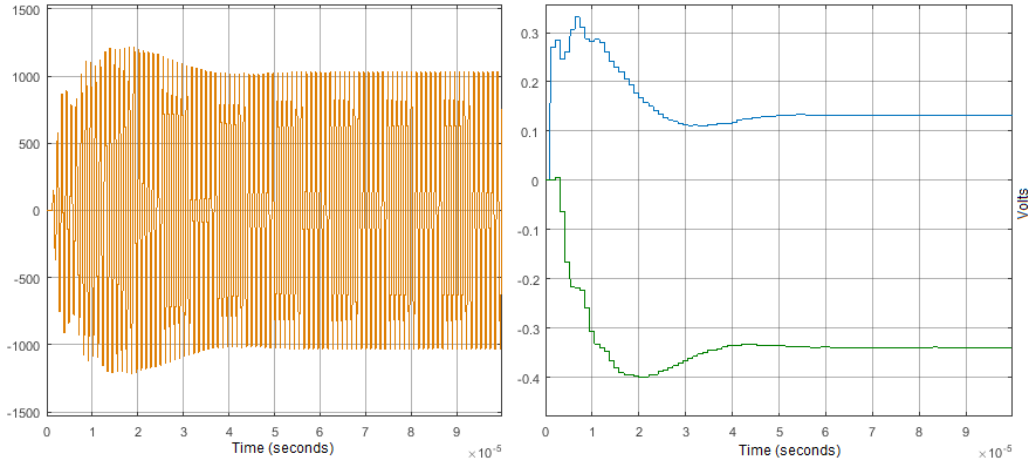


Figure 4.3.11 Fixed Point Model at 952.2 KHz, Cavity and IQ Drive, Closed-Loop

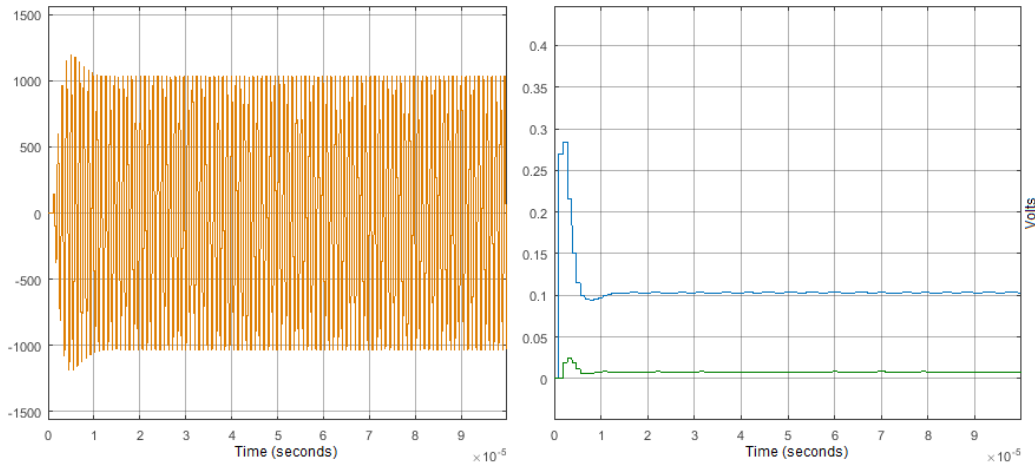


Figure 4.3.12 Fixed Point Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop

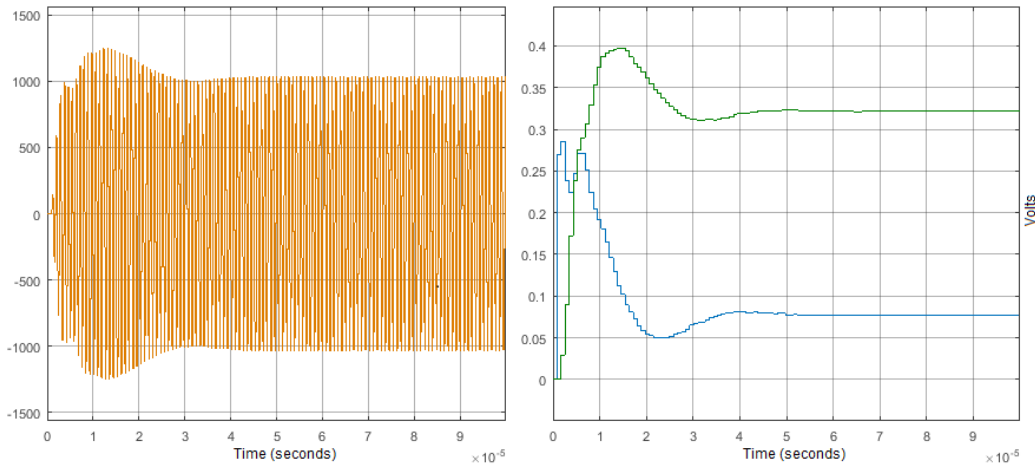


Figure 4.3.13 Fixed Point Model at 1163.8 KHz, Cavity and IQ Drive, Closed-Loop

4.4 Floating Point and Fixed Point Comparison

While examination of the preceding plots provides a high degree of confidence in the Fixed Point Model, a more robust analysis can be achieved through point-by-point error comparison of various Floating Point Model and Fixed Point Model results. Figure 4.4.1 and Figure 4.4.2 show the Floating Point Model and Fixed Point Model simulation results, respectively for a typical closed-loop PI controller configuration. The results shown in Figure 4.4.1 and Figure 4.4.2 were compared and used to generate the error plot in Figure 4.4.3, which examines by what percent the Fixed Point Model results differ from the Floating Point Model results. Figures 4.4.4 through 4.4.6 follow suit for a proportional-only controller. Finally, Figures 4.4.7 through 4.4.9 show the integral-only controller case. All in all the error between Floating Point Model and Fixed Point Model results oscillates between $\pm 2\%$ in all ranges, though the percent error prior to steady state in all cases is lower due to percent error in this case being calculated based on maximum values.

4.5 Fixed Point Model with Delay Results

In order to accurately compare between the software-based and the hardware-based models, a multi-sample delay must be introduced to the software model's path to match the latency present in hardware controller expressions. At the time of this writing, the hardware implementation of the controller introduces a 12 clock-cycle delay between input and output. This effect automatically presents itself in the System Generator Model, but the Fixed Point Model requires the addition of a special 12-clock delay block in the control loop. Aside from this notable change, the test cases presented in this section are identical to the test cases seen for both the

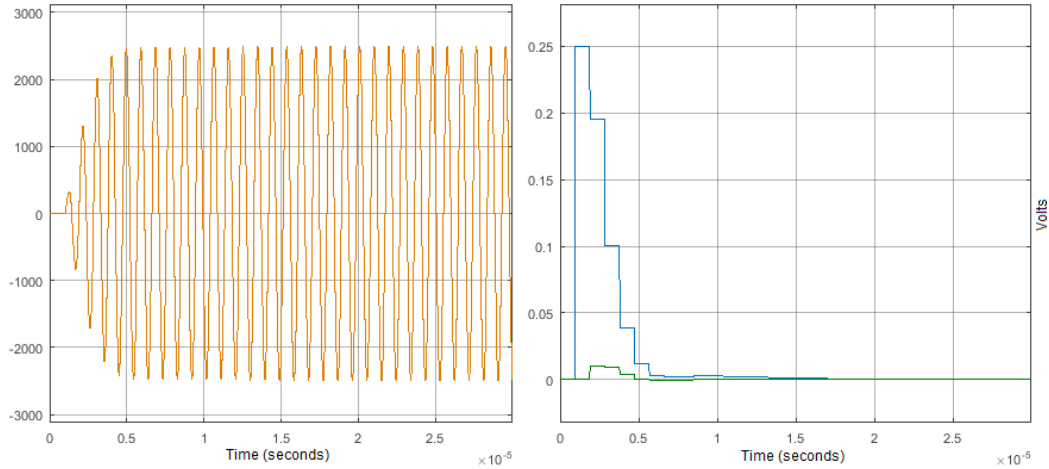


Figure 4.4.1 Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$

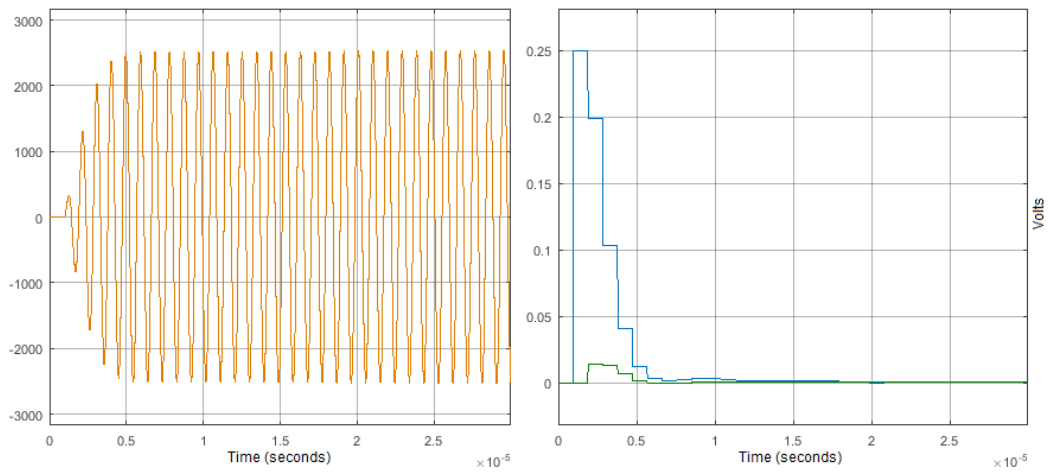


Figure 4.4.2 Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$

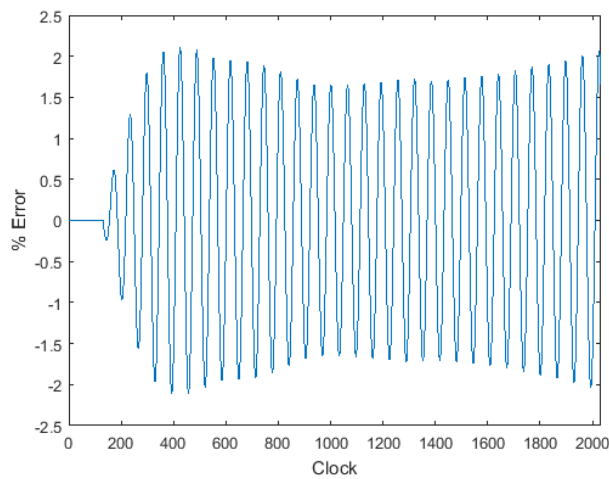


Figure 4.4.3 Percent Error of Fixed Point Model Results from Floating Point Model Results, 1

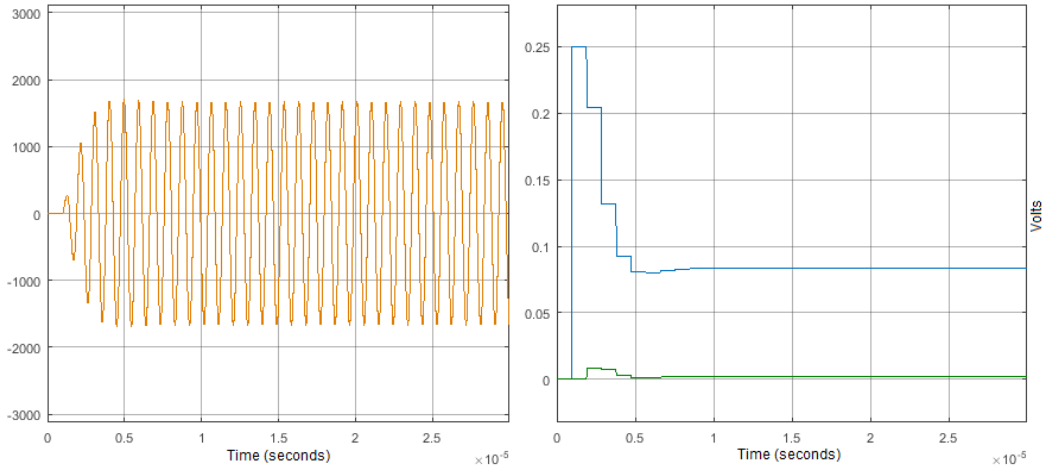


Figure 4.4.4 Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$

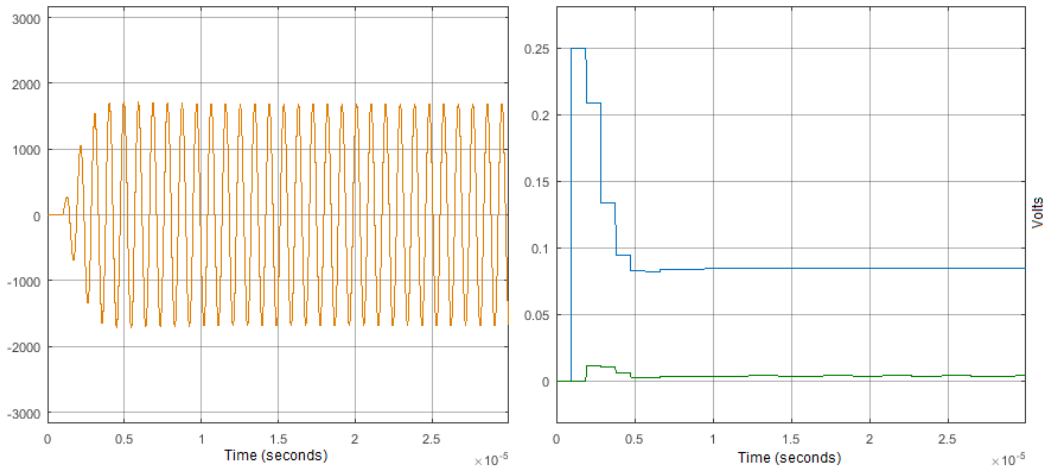


Figure 4.4.5 Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$

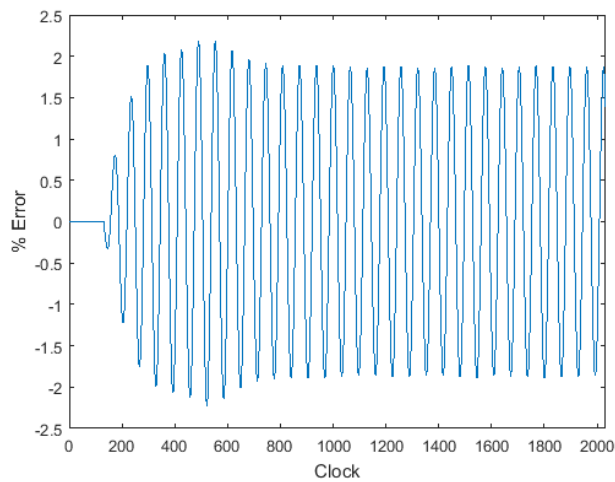


Figure 4.4.6 Percent Error of Fixed Point Model Results from Floating Point Model Results, 2

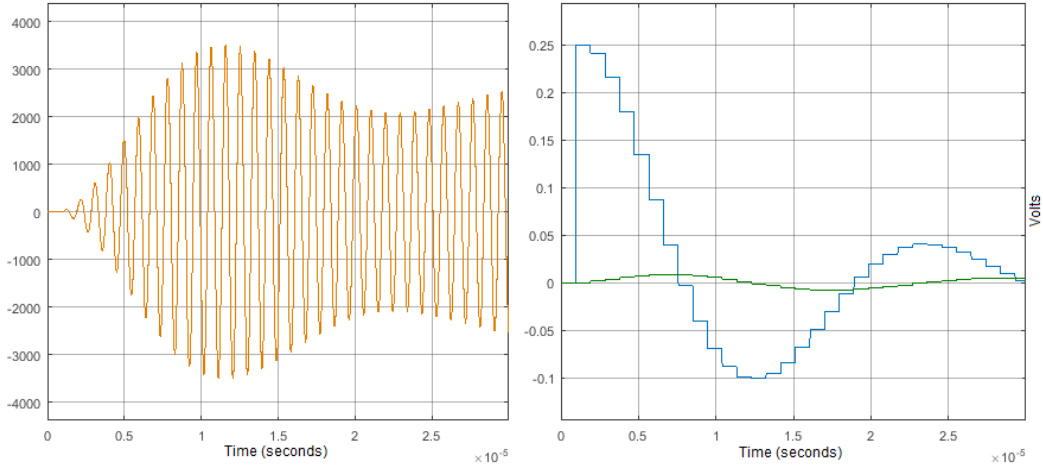


Figure 4.4.7 Floating Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$

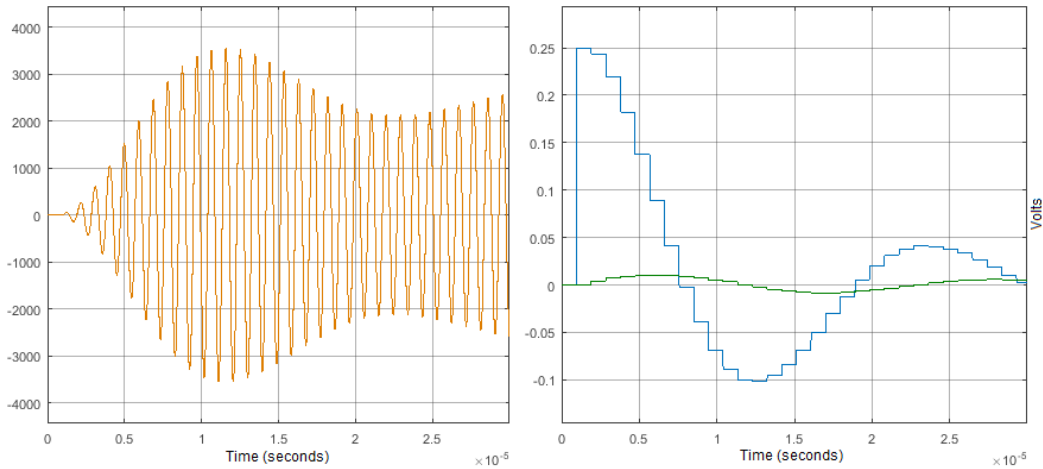


Figure 4.4.8 Fixed Point Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$ $K_d = 0$

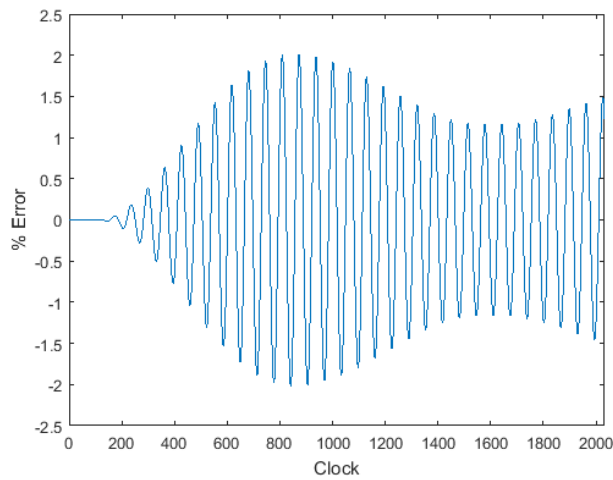


Figure 4.4.9 Percent Error of Fixed Point Model Results from Floating Point Model Results, 3

Floating Point Model and the Fixed Point Model up to this point. Figures 4.5.1 through 4.5.4 demonstrate the modelled controller successfully regulating with and without injected cavity noise. Figures 4.5.5 through 4.5.7 demonstrate the performance of the Fixed Point Model with delay. Finally, Figures 4.5.8 through 4.5.13 demonstrates how well the Fixed Point Model with added delay handles off-resonance configurations.

4.6 System Generator Model Results

The final set of results for this chapter is from the System Generator Model. The System Generator Model underwent the same tests as the previous 3 cases, with general regulation tests with and without noise seen in Figures 4.6.1 through 4.6.4, PID verification tests seen in Figures 4.6.5 through 4.6.7, and resonance-based tests seen in Figures 4.6.8 through 4.6.13.

4.7 Fixed Point with Delay and System Generator Comparison

Once again, side-by-side comparisons combined with point-by-point error analysis provide a robust method for verification modelled controller function. The following plots demonstrate the results of these investigations and while the error between the System Generator Model results and the delayed Fixed Point Model peaks at about $\pm 15\%$ error compared to $\pm 2\%$ constant error in Section 4.5, there is still general agreement across the cases and the error gradually settles to less than $\pm 5\%$. There are three cases where System Generator Model results and delayed Fixed Point Model results are compared: proportional-integral (PI) Figures 4.7.1 through 4.7.3, proportional only in Figures 4.7.4 through 4.7.6, and open-loop in Figure 4.7.7 through 4.7.9.

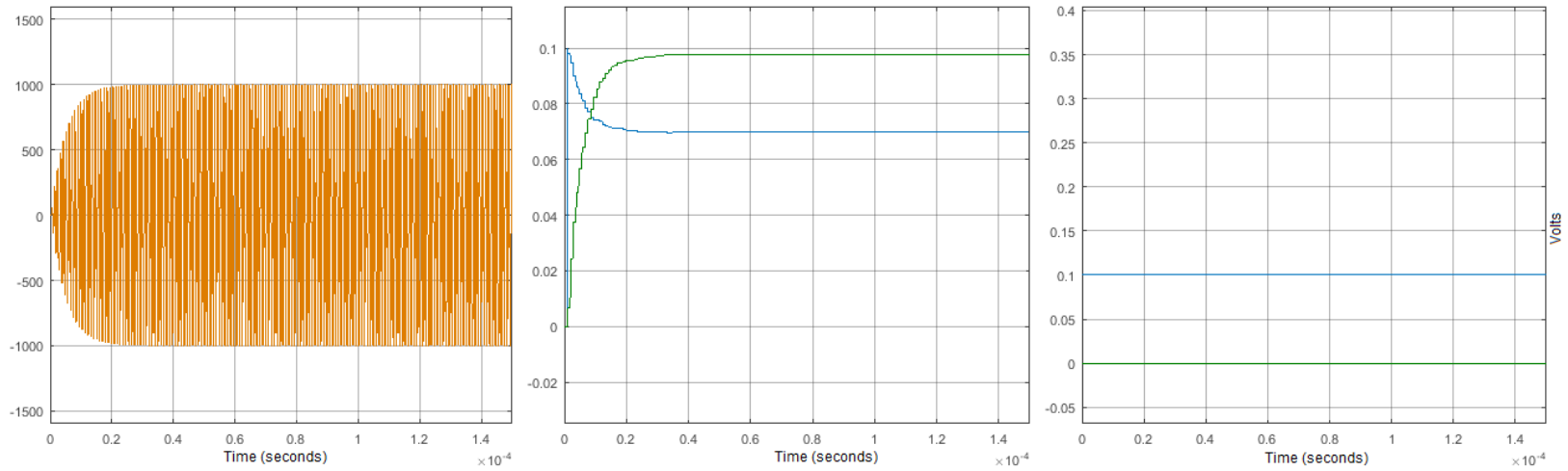


Figure 4.5.1 Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

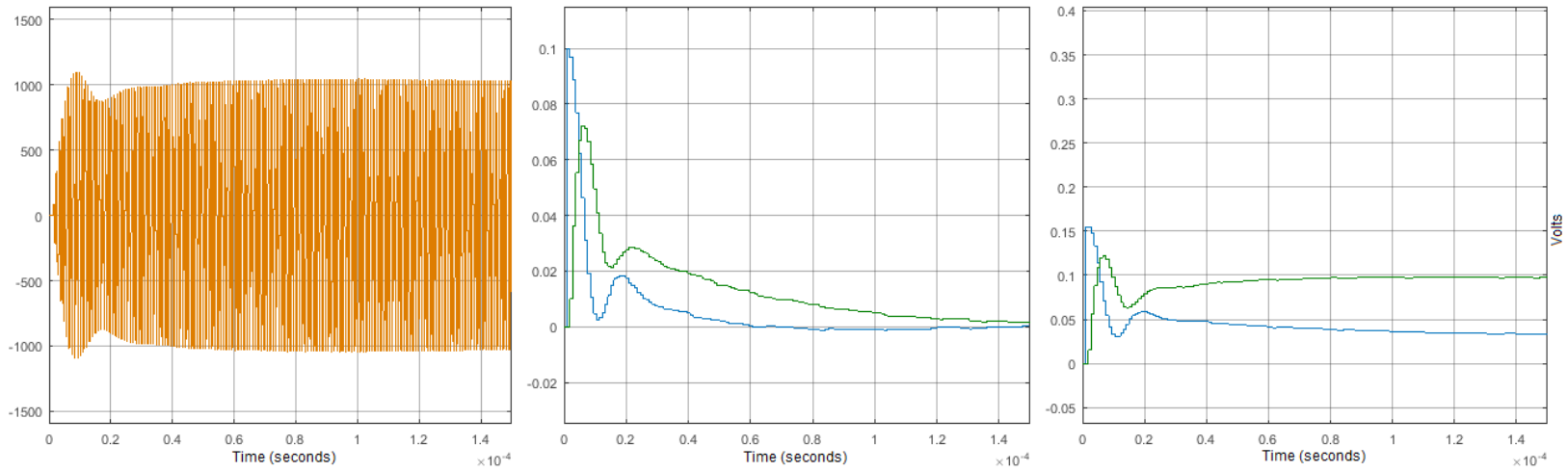


Figure 4.5.2 Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 1.5$, $K_i = 0.05$, $K_d = 0$

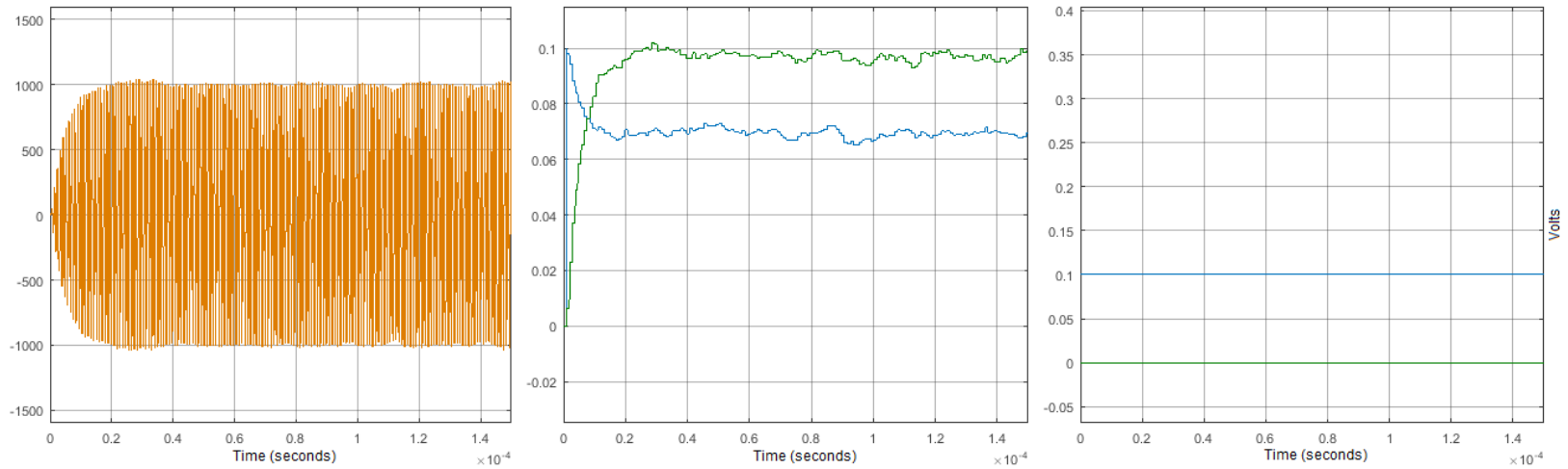


Figure 4.5.3 Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

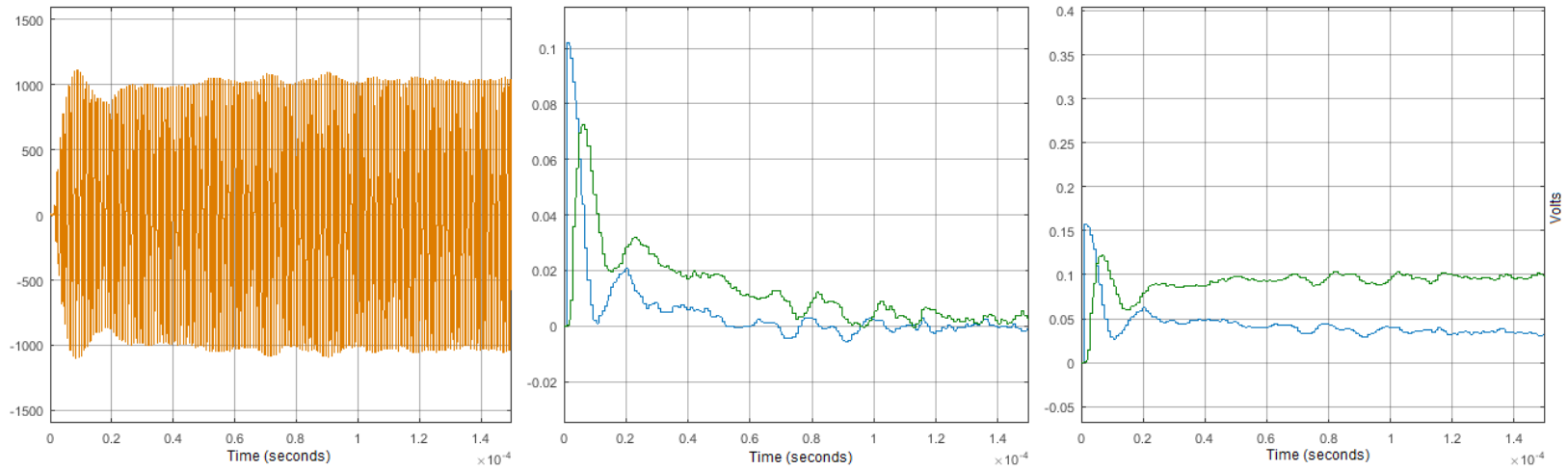


Figure 4.5.4 Fixed Point Delay Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 1.5$, $K_i = 0.05$, $K_d = 0$

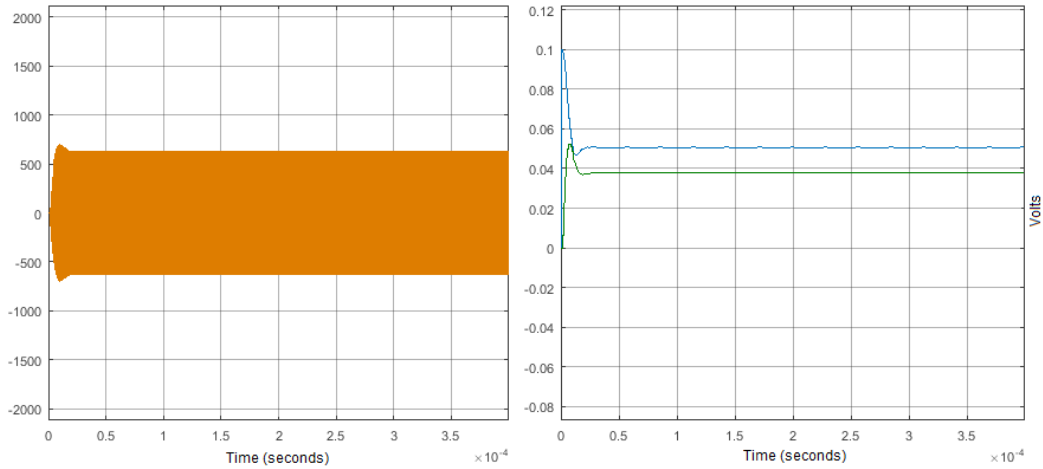


Figure 4.5.5 Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$

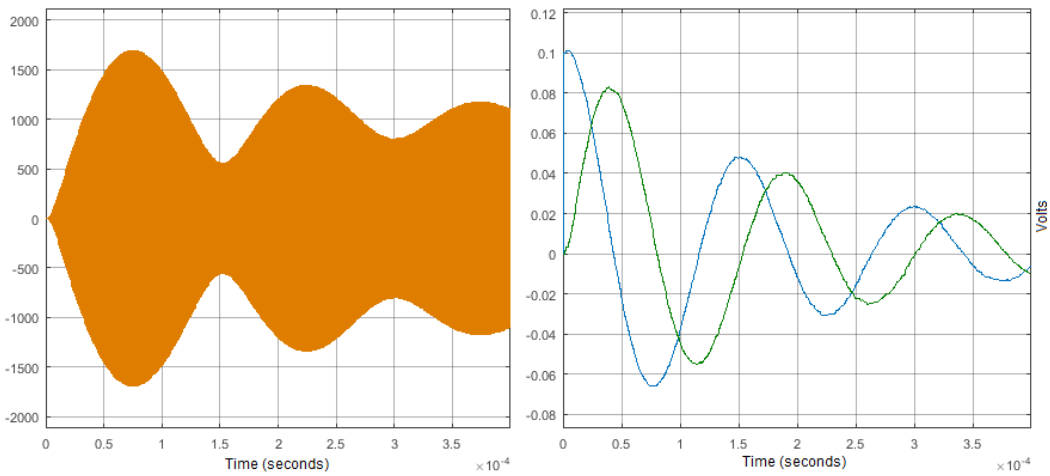


Figure 4.5.6 Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$

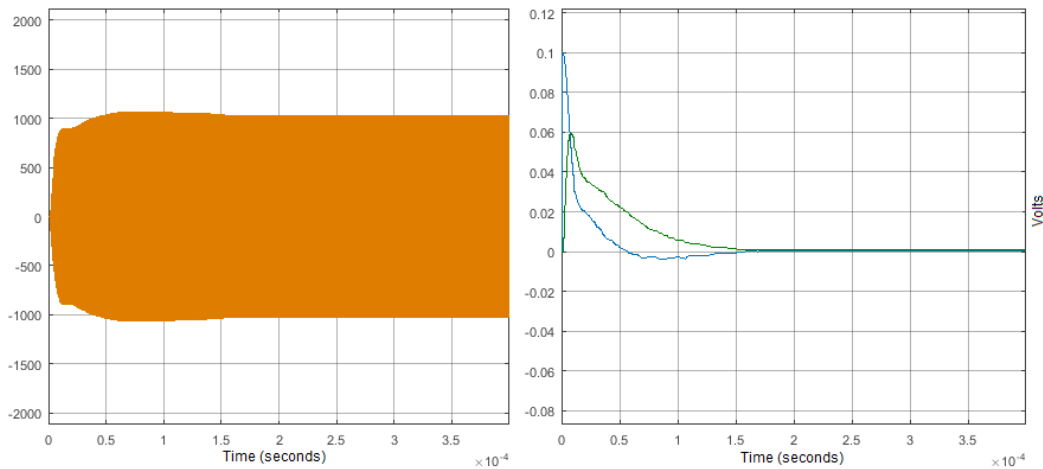


Figure 4.5.7 Fixed Point Delay Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$

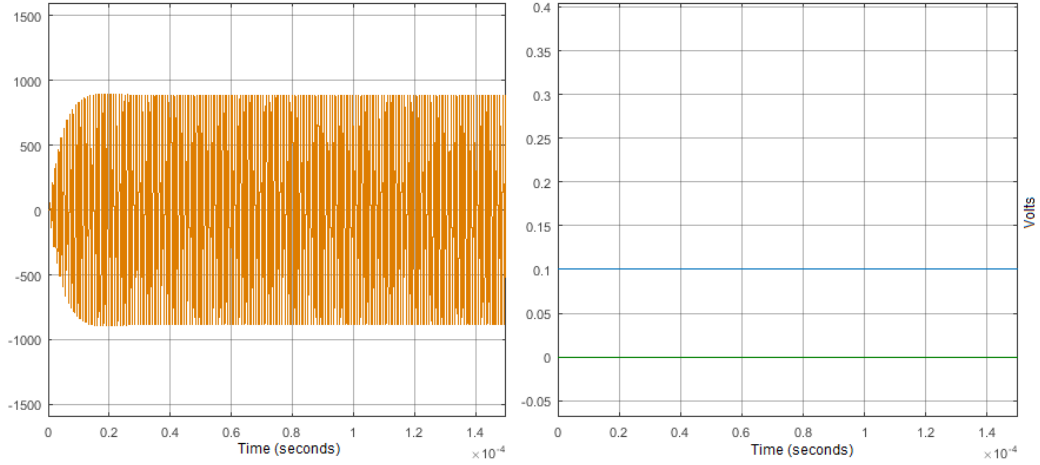


Figure 4.5.8 Fixed Point Delay Model at Low Frequency, Cavity and IQ Drive, Open-Loop

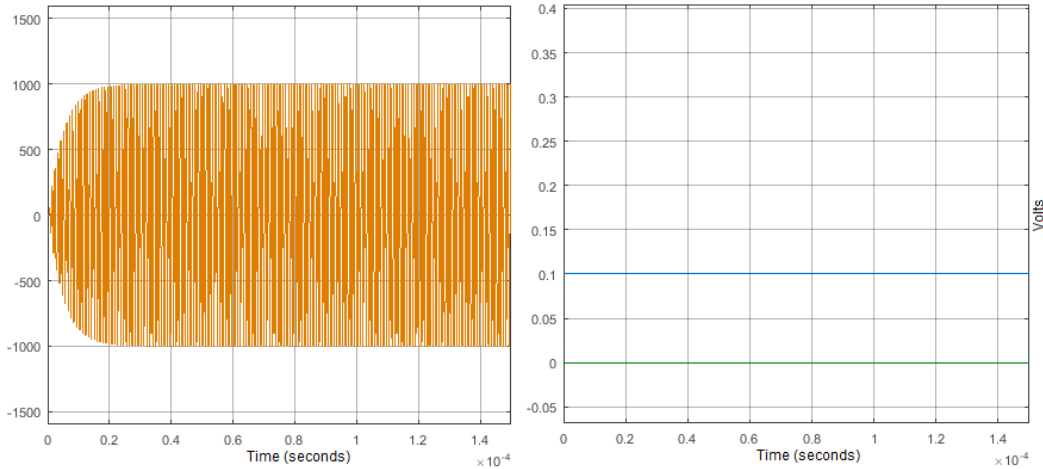


Figure 4.5.9 Fixed Point Delay Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop

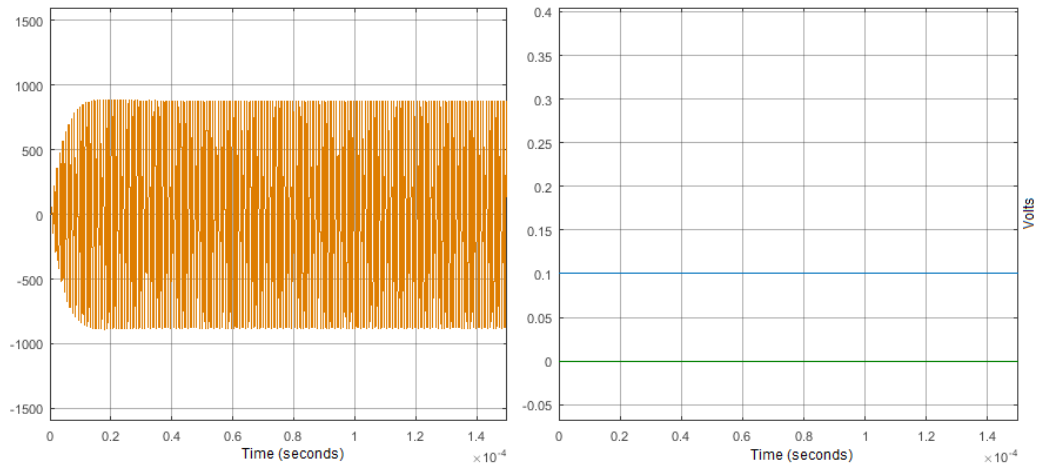


Figure 4.5.10 Fixed Point Delay Model at High Frequency, Cavity and IQ Drive, Open-Loop

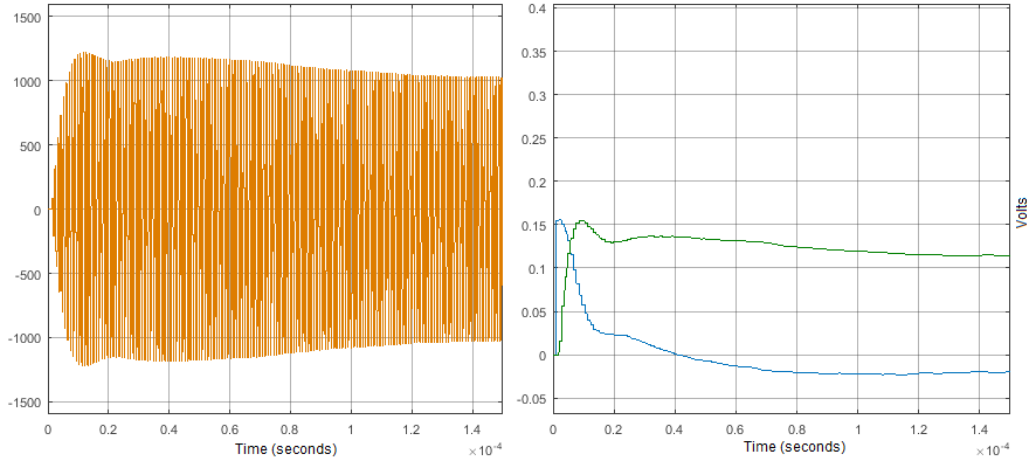


Figure 4.5.11 Fixed Point Delay Model at Low Frequency, Cavity and IQ Drive, Closed-Loop

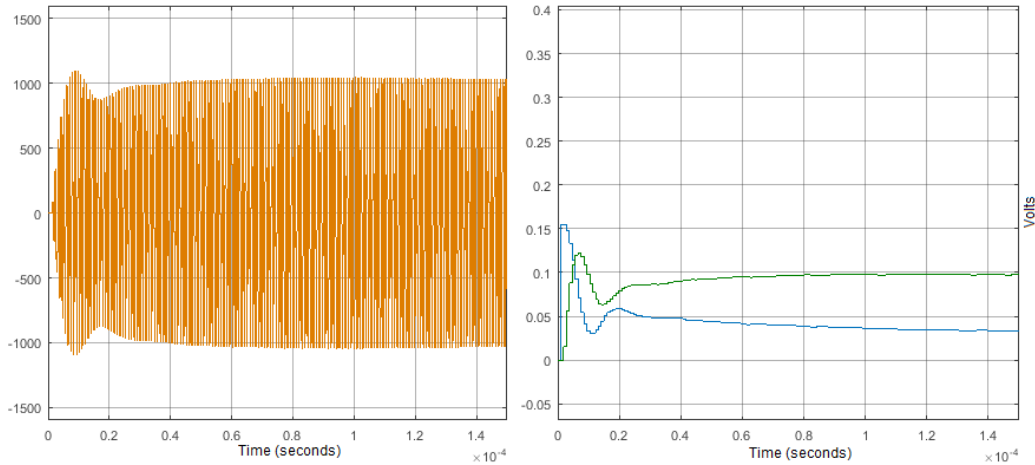


Figure 4.5.12 Fixed Point Delay Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop

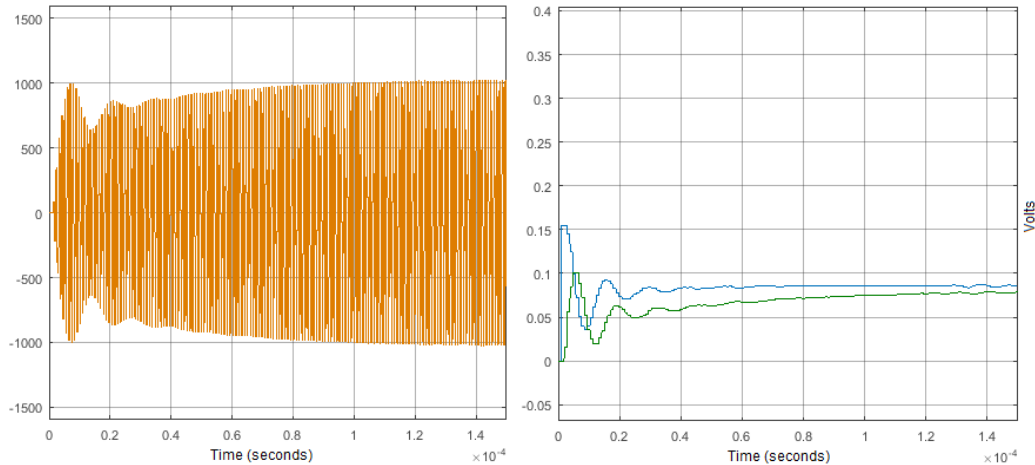


Figure 4.5.13 Fixed Point Delay Model at High Frequency, Cavity and IQ Drive, Closed-Loop

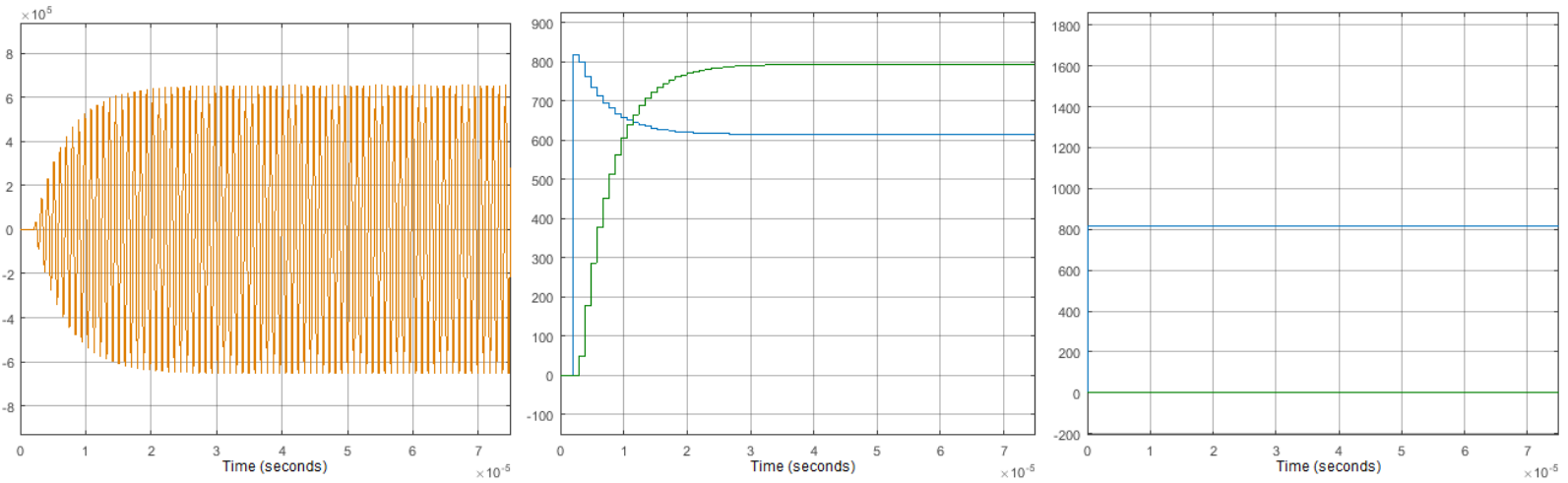


Figure 4.6.1 System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

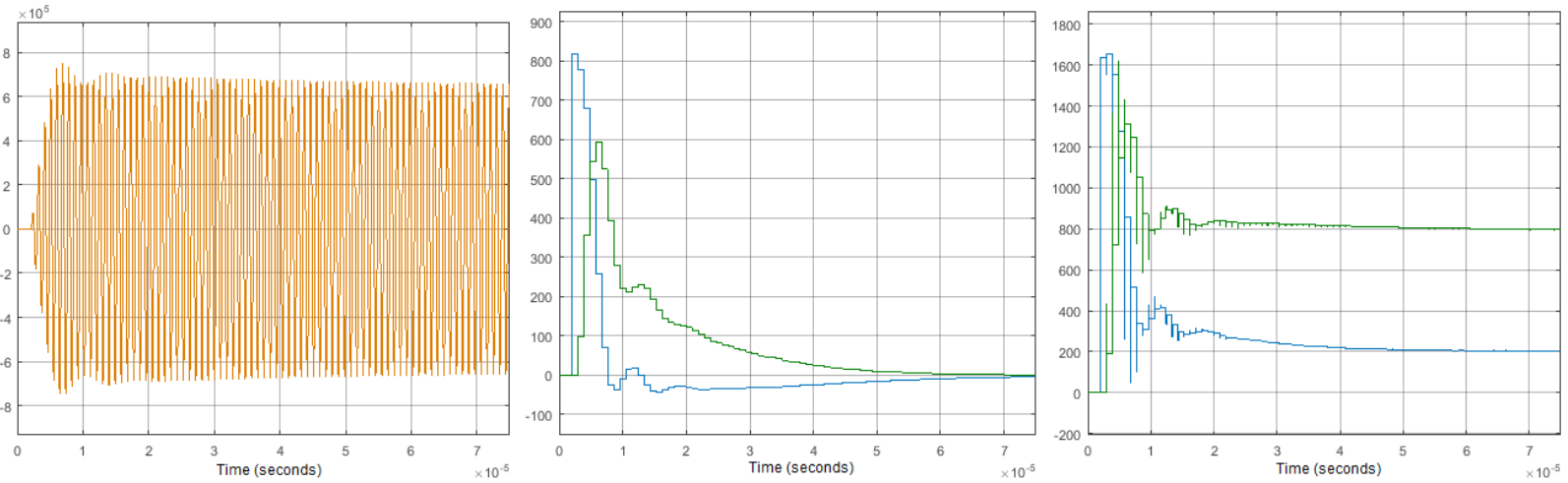


Figure 4.6.2 System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.125$, $K_d = 2.5$

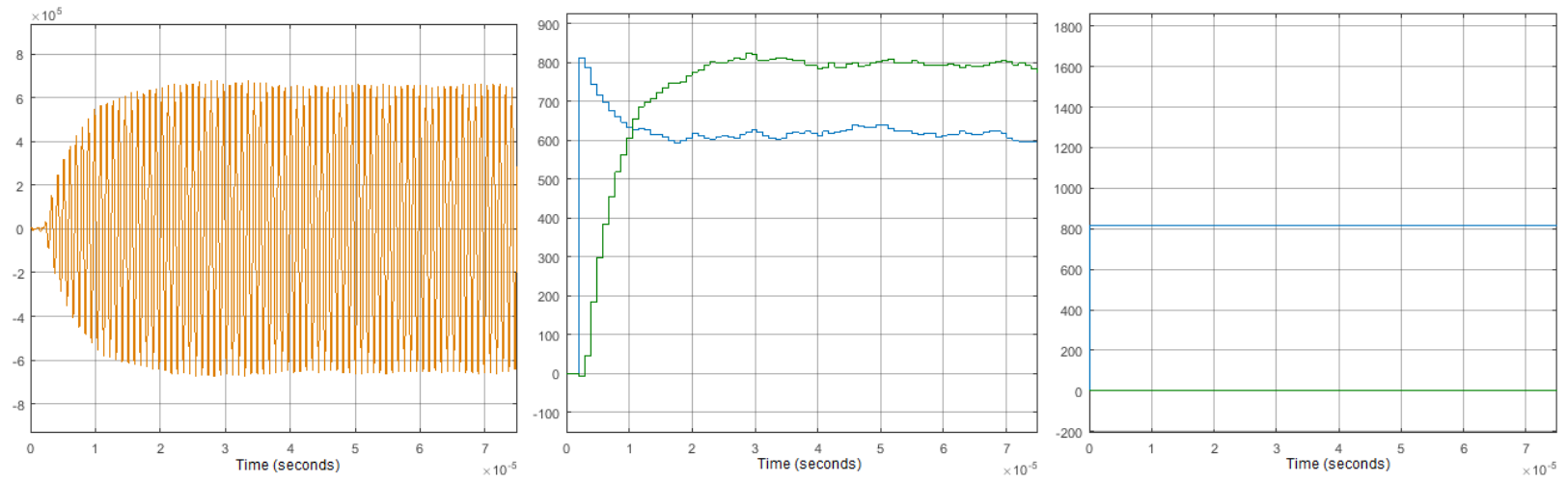


Figure 4.6.3 System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Open-Loop

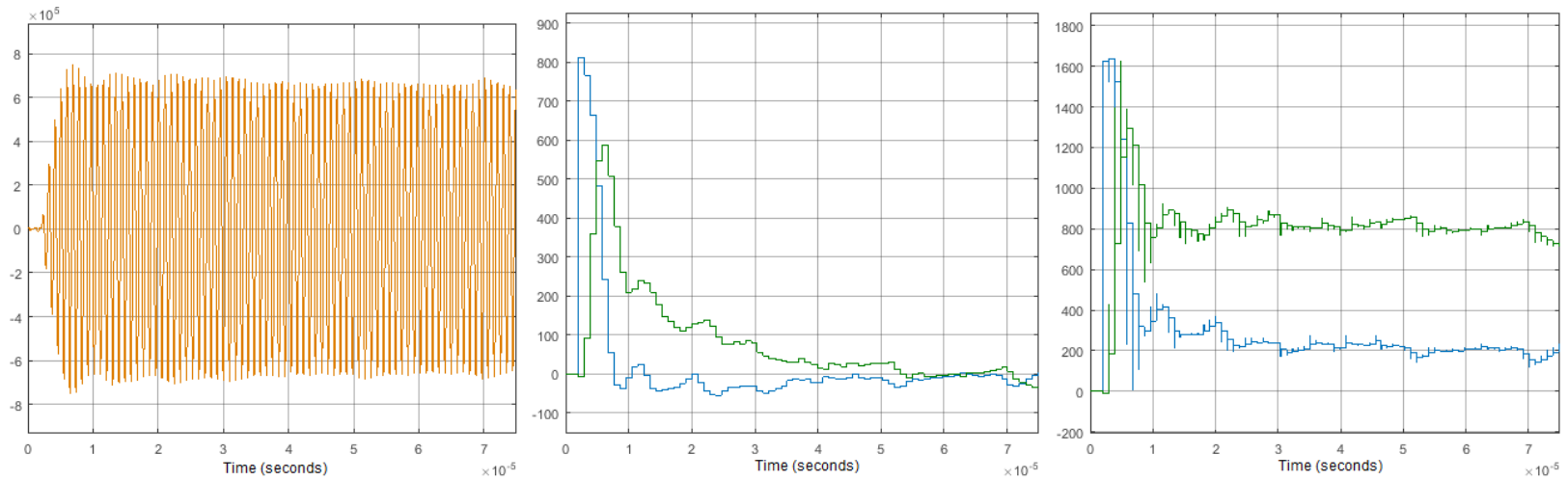


Figure 4.6.4 System Generator Model Cavity, IQ Error, and IQ Drive with Noise in Closed-Loop. $K_p = 2$, $K_i = 0.125$, $K_d = 2.5$

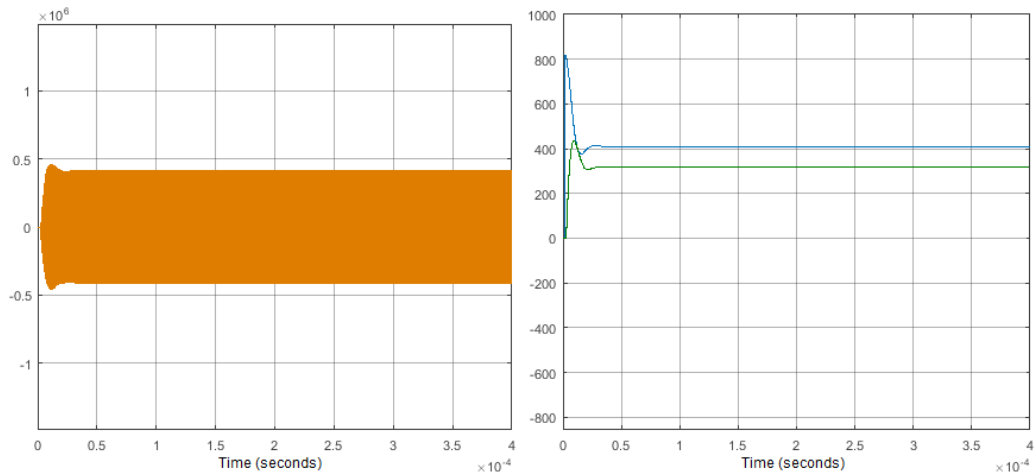


Figure 4.6.5 System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$

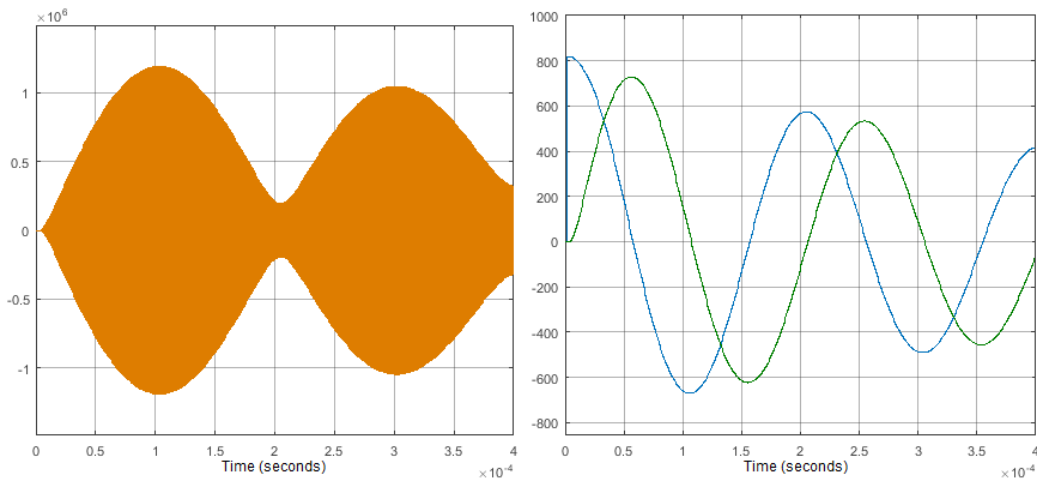


Figure 4.6.6 System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 0$ $K_i = 0.4$

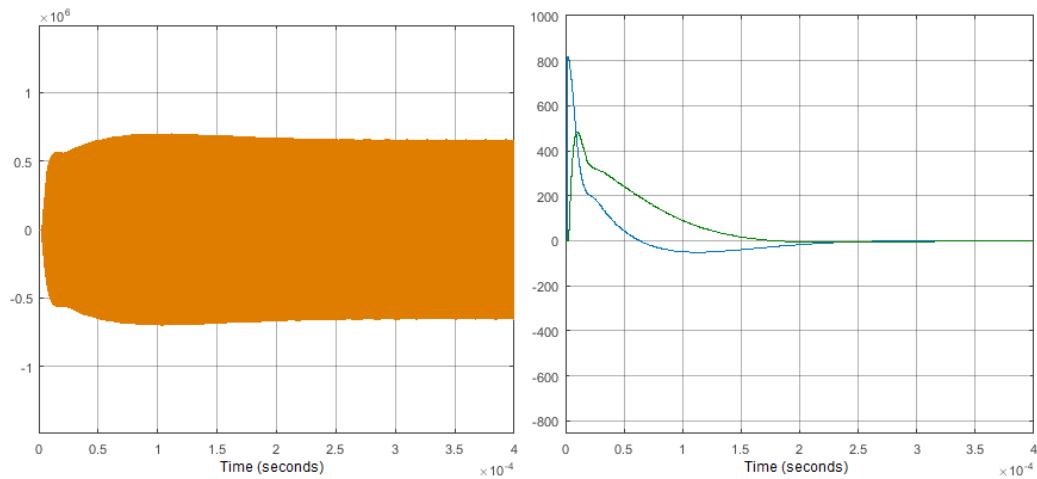


Figure 4.6.7 System Generator Model, Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.4$

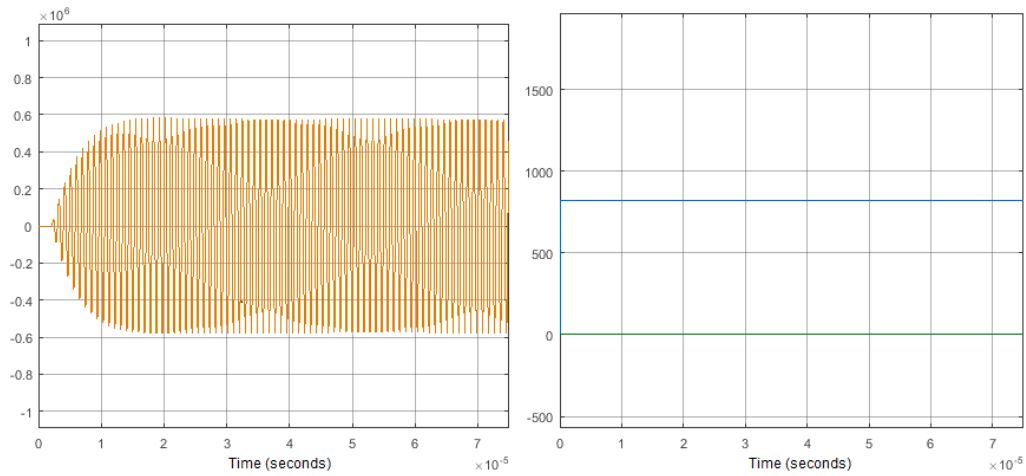


Figure 4.6.8 System Generator Model at Low Frequency, Cavity and IQ Drive, Open-Loop

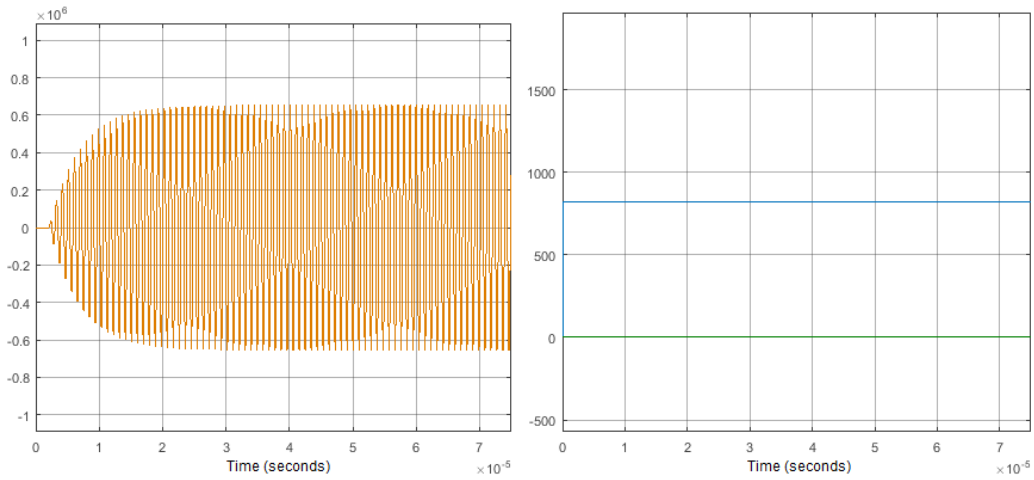


Figure 4.6.9 System Generator Model at 1.058 MHz, Cavity and IQ Drive, Open-Loop

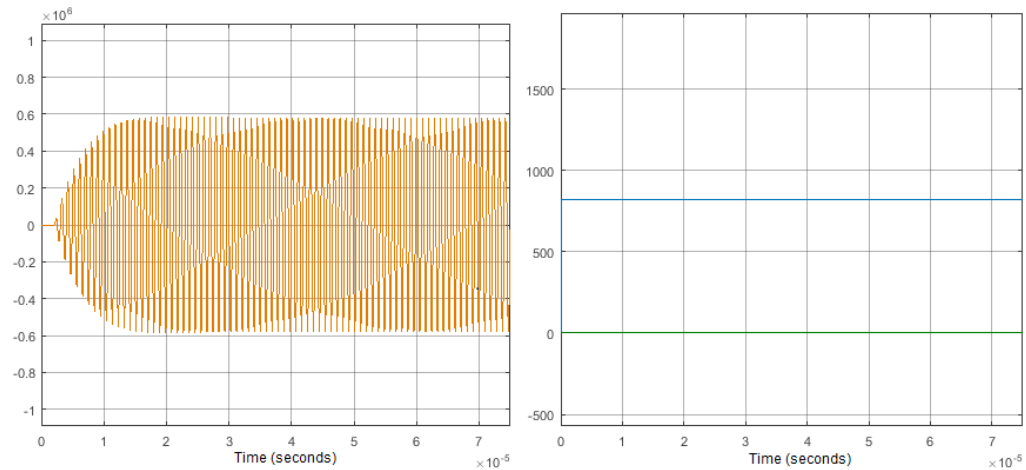


Figure 4.6.10 System Generator Model at High Frequency, Cavity and IQ Drive, Open-Loop

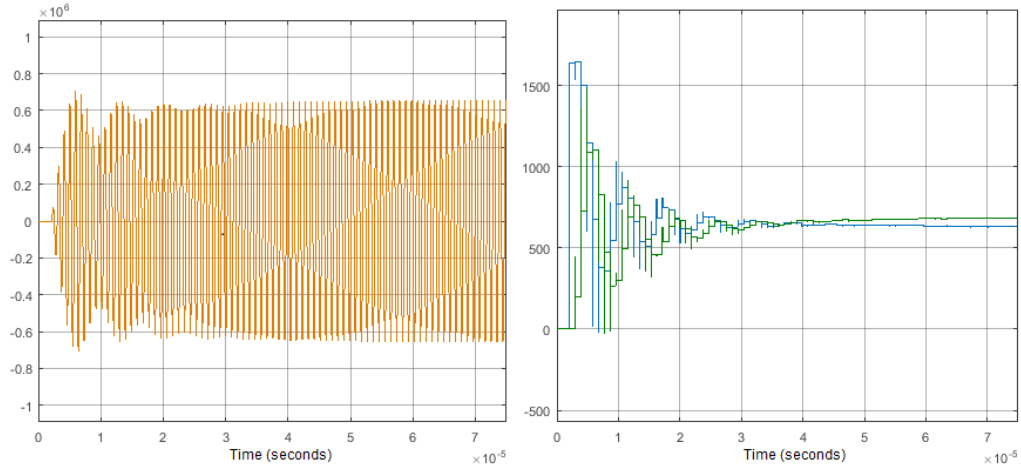


Figure 4.6.11 System Generator Model at Low Frequency, Cavity and IQ Drive, Closed-Loop

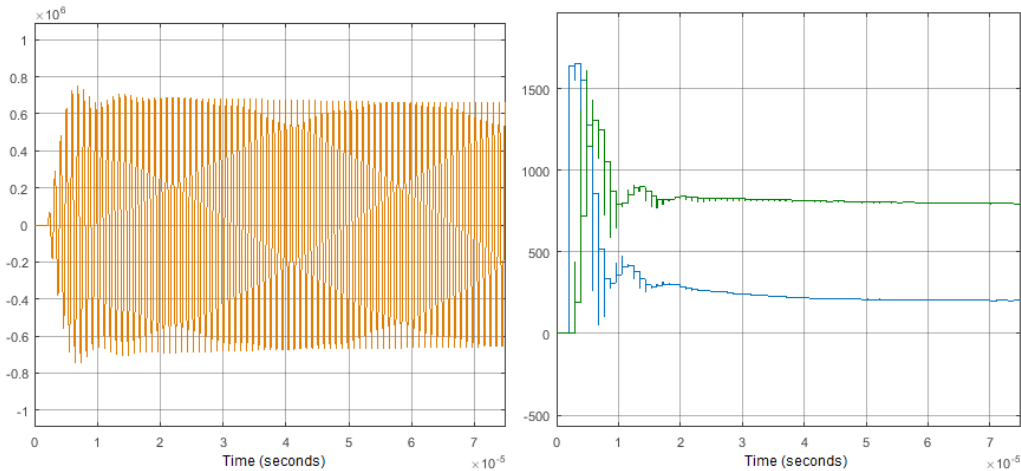


Figure 4.6.12 System Generator Model at 1.058 MHz, Cavity and IQ Drive, Closed-Loop

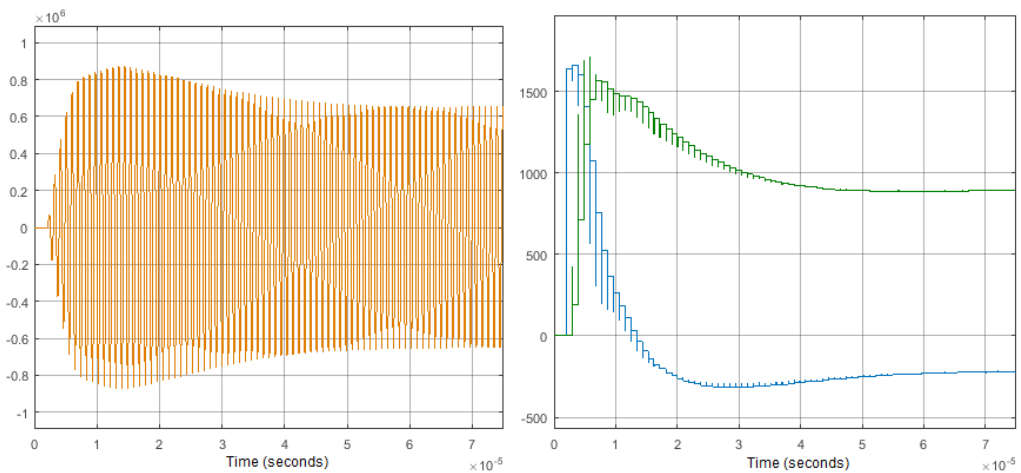


Figure 4.6.13 System Generator Model at High Frequency, Cavity and IQ Drive, Closed-Loop

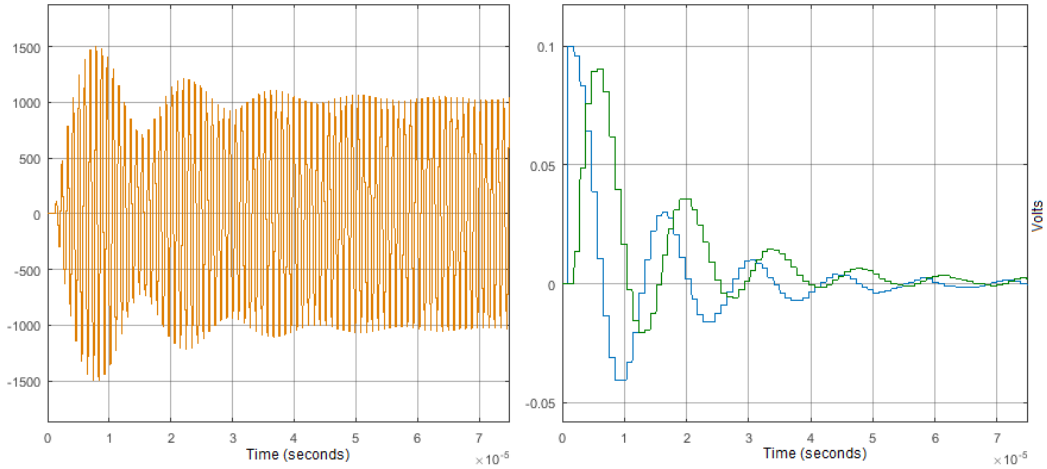


Figure 4.7.1 Fixed Point Delay Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.125$ $K_d = 0$

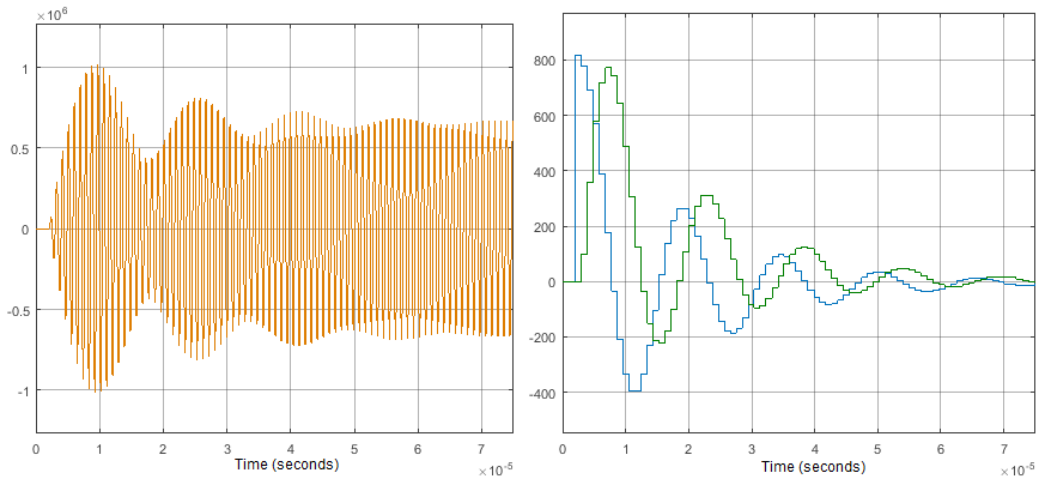


Figure 4.7.2 System Generator Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0.125$ $K_d = 0$

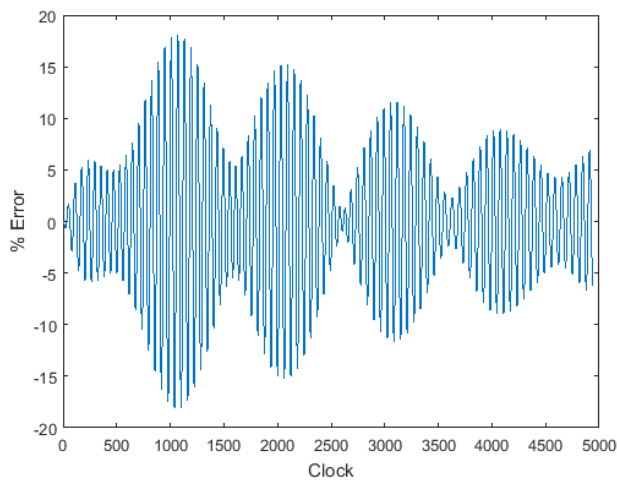


Figure 4.7.3 Percent Error of Fixed Point Delay Model Results from System Generator Model Results, 1

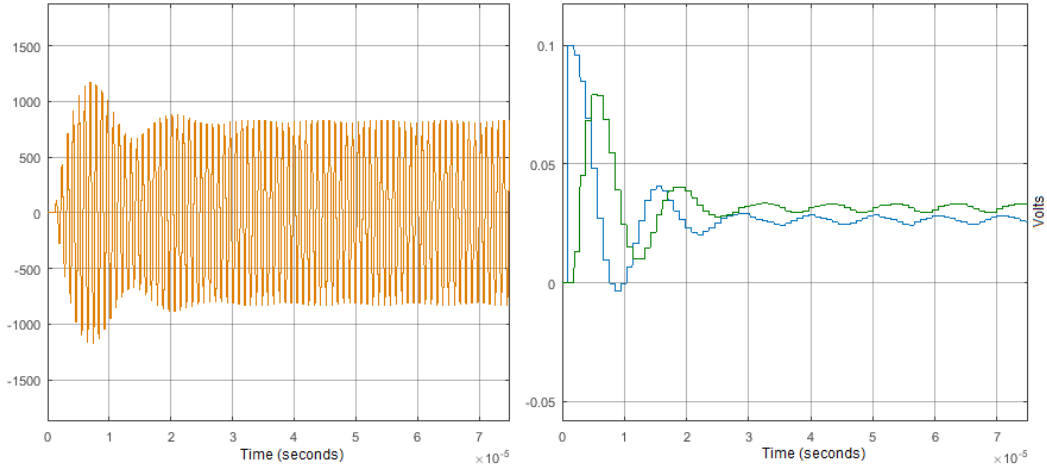


Figure 4.7.4 Fixed Point Delay Model Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$

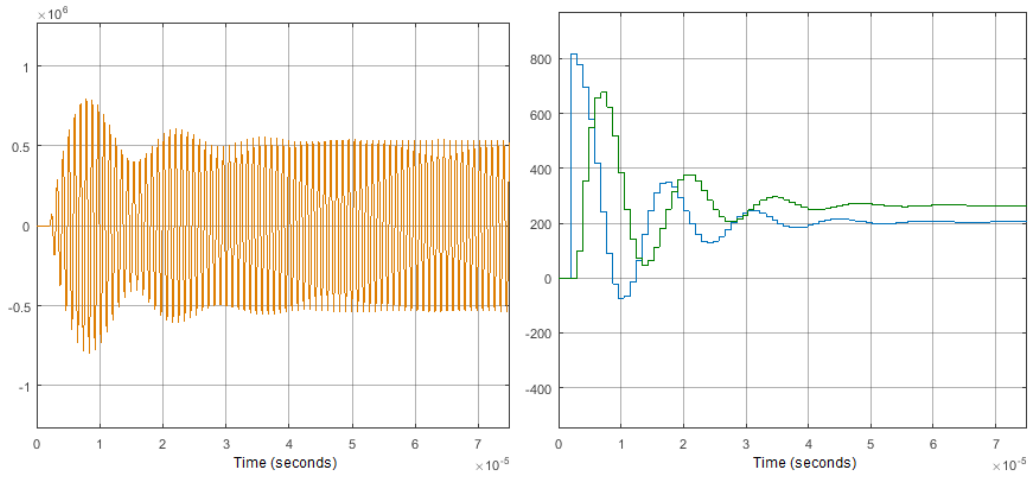


Figure 4.7.5 System Generator Cavity and IQ Error, Closed-Loop, $K_p = 2$ $K_i = 0$ $K_d = 0$

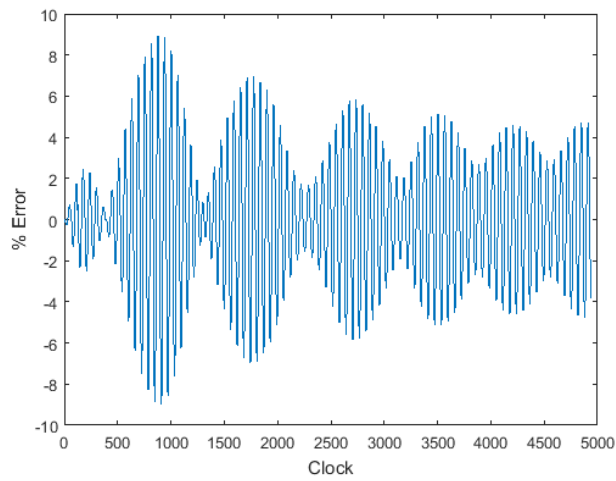


Figure 4.7.6 Percent Error of Fixed Point Model Results from System Generator Model Results, 2

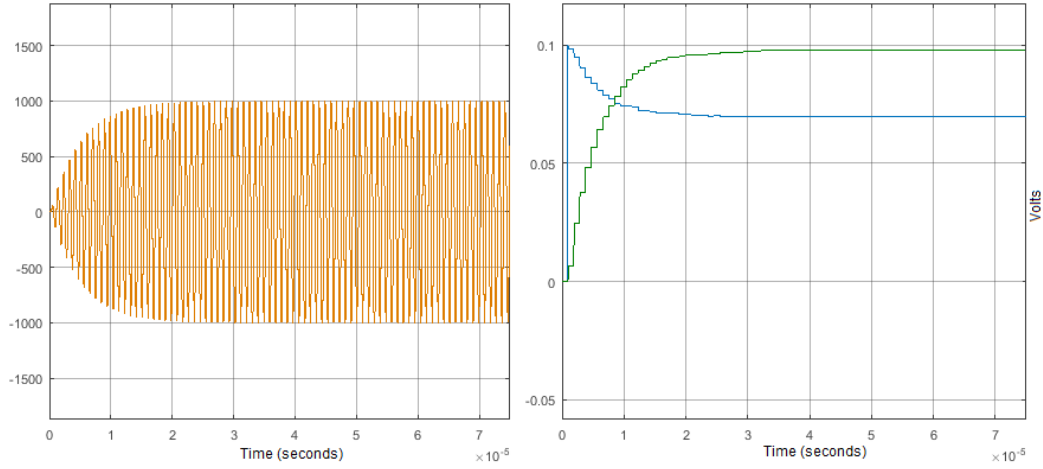


Figure 4.7.7 Fixed Point Delay Model Cavity and IQ Error, Open-Loop

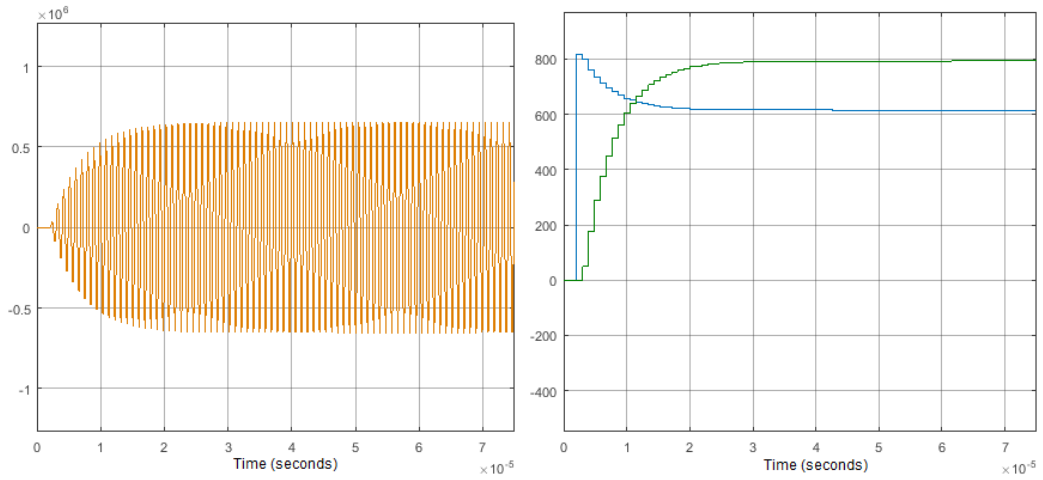


Figure 4.7.8 System Generator Cavity and IQ Error, Open-Loop

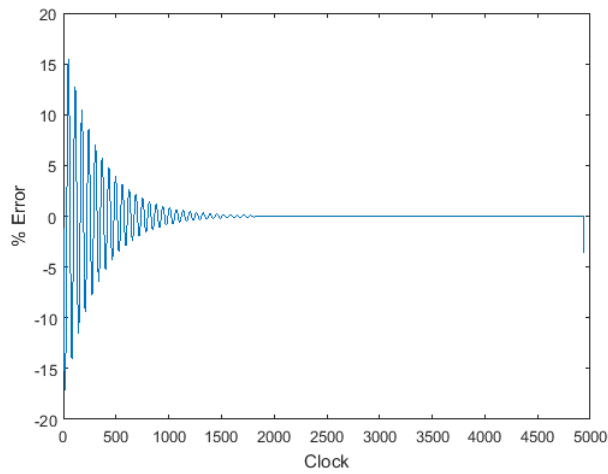


Figure 4.7.9 Percent Error of Fixed Point Delay Model Results from System Generator Model Results, 3

Chapter 5

Physical System Testing

After extensive simulation of the controller models, the VHDL controller code was integrated into a Xilinx Kintex-7 FPGA design for the new LLRF controller hardware. Details of this broader control hardware solution are beyond the scope of this thesis, but the VHDL code previously verified in the System Generator Model served as the controller core in this system, interfacing with the new LLRF controller hardware's DAC and ADC.

Figure 5.1.1 shows the status screen from the LLRF control system user interface for a preliminary RF cavity test. This first test dealt only with how the cavity and control system would react to a simulated load achieved by varying the tuning of the cavity. Note in the Cavity I and Cavity Q plots how the cavity gap voltage quickly reaches the set point with minimal ringing. This controller unit was then used to successfully operate as the SNS Accumulator Ring LLRF control system on May 28th, 2016 when 1 MW beam-load at a repetition rate of 60 Hz was successfully regulated by our controller. A status screen displaying this regulation is shown in Figure 5.1.2. In both Figure 5.1.1 and Figure 5.1.2 it's important to note that the pulse in question is only ~1.5 ms, so the waveform portions beyond roughly turn 1500 are not relevant despite being displayed in some of the plots.

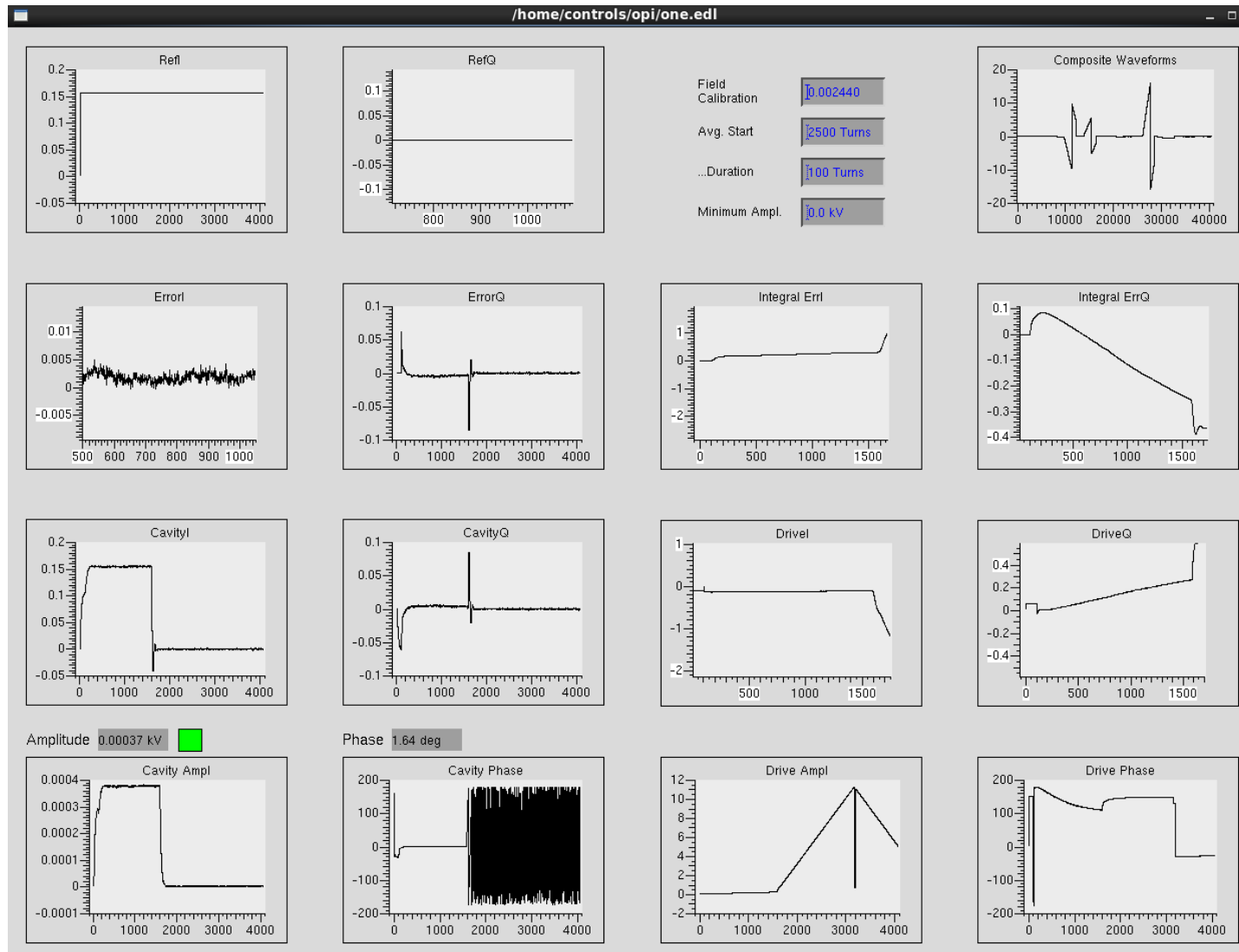


Figure 5.1.1 SNS Accumulator Ring LLRF Control System Dynamic Tuning Test Status Screen

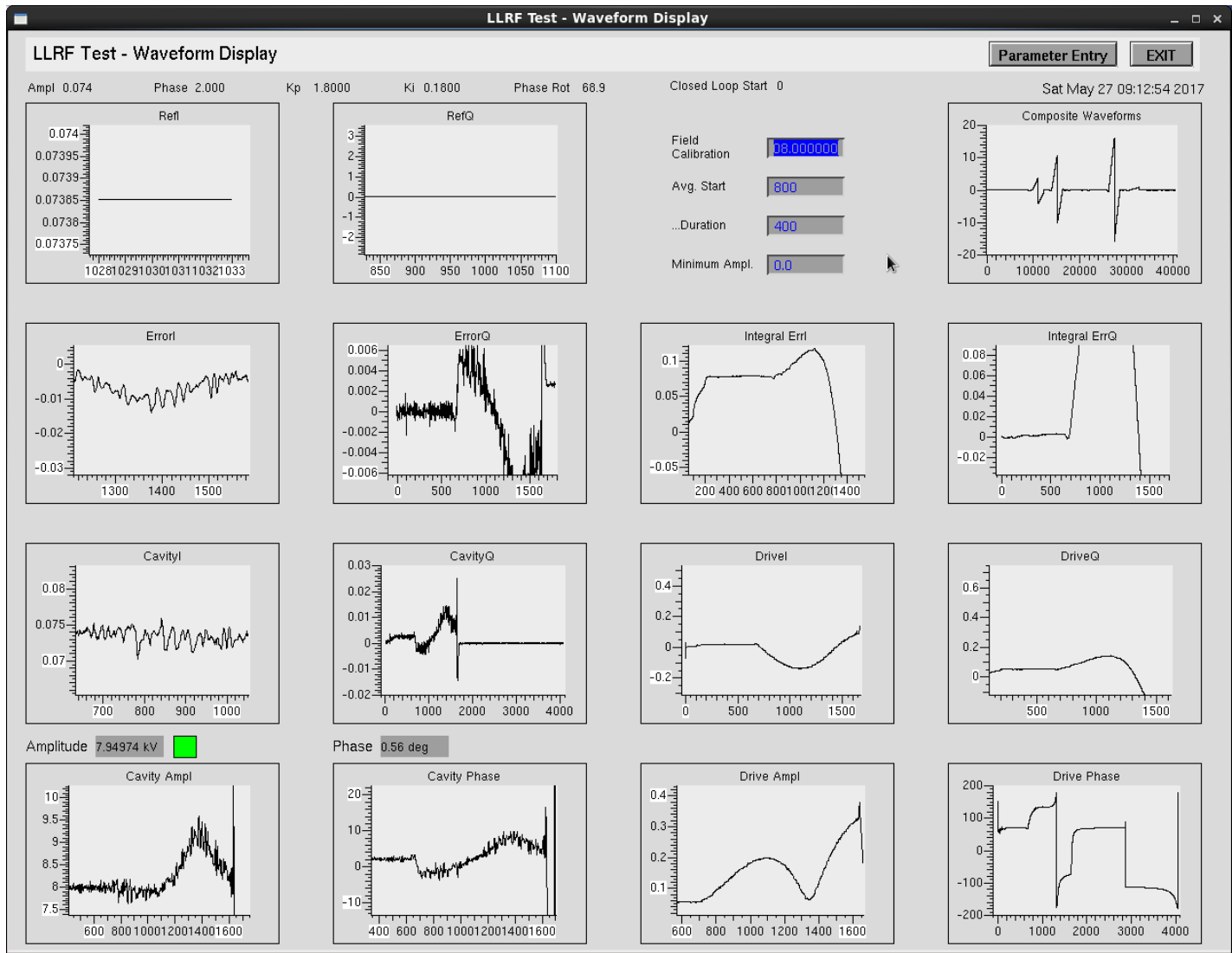


Figure 5.1.2 SNS Accumulator Ring LLRF Control System 1 Megawatt 60Hz Test Status Screen

Chapter 6

Conclusion

This thesis demonstrates that software modeling can successfully inform the broader design of particle accelerator RF control systems, and can aid the commissioning of such systems as well. These findings are of interest to not only Spallation Neutron Source and ORNL generally, but also the entire accelerator community in a very practical sense; the modeling described in this thesis was critical in expediting the development and deployment of the new SNS Ring LLRF system. Indeed, the project went from concept to 1 MW beam testing in less than one year. Further, the PID tuning script leveraged the software models to provide a sense of optimal system configuration without tedious, expensive, manual hardware tests. So not only did model-enabled design enable robust controller development, but it also afforded considerable time-savings over initial projections.

The case studies in Chapter 4 verify that the controller models are indeed process-accurate and the general data in Appendix D demonstrate the performance of the controller models. However, the Chapter 5 hardware controller tests performed on the Accumulator Ring itself stand as the true capstone for this endeavor; the ultimate conclusion to this thesis is that a successful Low-Level RF Controller was achieved in no small part due to the process this document describes.

Long-term benefits of system modeling on maintenance and further experimentation have yet to be proven, but our understanding of system dynamics can only improve with time as

cavity behavior models are refined and improved. Even now, a parameterized, actively tunable cavity model is being investigated for purposes of correcting for beam loading and dynamic cavity tuning. So while the controller aspects of the model are very true to form – in the case of the System Generator model identically so – the next step in improving the system models is perfecting cavity model operation beyond the basics. These concepts are already being considered for generalization to SNS Linear Accelerator RF cavities.

To the academic, this thesis demonstrates what may have been already known: control theory and physics can be expressed through software models to inform controller design. But to the engineer, this thesis offers very real assurance of the benefits that can be found through similar inquiries: expedited development, more secure understanding of system limitations, verification of potential designs prior to capital investment, and an attractive supplement or even alternative to traditional HDL simulator debugging.

Bibliography

- [1] <http://neutrons.ornl.gov/>
- [2] “The Spallation Neutron Source Accelerator System Design” S. Henderson, et al.,
Nuclear Instruments & Methods in Physics Research A (2014)
- [3] “RF System for the SNS Accumulator Ring” M. Blaskiewicz, et al., 2001
- [4] “Theoretical Foundations of Synchrotron and Storage Ring RF Systems” H. Klingbeil, U.
Laier, D. Lens, 2015
- [5] “Modeling of the RF Control System of a Synchrotron Accelerating Unit” Uta Hartel,
Diploma Thesis, Technische Universitat Darmstadt (Technical University of Darmstadt),
2011
- [6] “RF Linear Accelerators”, Thomas Wangler, 1998
- [7] “Fermilab Concepts Rookie Book”, Accelerator Division – Operations Department
http://operations.fnal.gov/rookie_books/concepts.pdf 2013
- [8] “SNS Timing Master Function System Description” T. Justice, 2011
- [9] “SNS LLRF Controls Engineering Notebook: Volume 1” J. Breeding, 2017
- [10] “Chip Piller Engineering Notebook” M. Piller, 2017
- [11] “RF Applications in Digital Signal Processing” T. Schilcher, 2007
- [12] “Control System Design: An Introduction to State-Space Methods” B. Friedland, 1986

Appendices

Appendix A

Derivation of Accumulator Ring Turn Frequency

The frequency and period of the Spallation Neutron Source Accumulator Ring is determined by the circumference of the Ring as well as the energy of a particular particle incident to the Ring. Generally speaking, the ideal Accumulator Ring particle is a 1 GeV proton whose total energy E can be expressed as

$$E = k + mc^2 \quad (eq A. 1)$$

where k is the kinetic energy of the proton, m is the mass of the proton, and c is the speed of light in a vacuum . Substituting the appropriate parameter values shows that

$$E = (1 \text{ GeV}) + \left(0.938 \frac{\text{GeV}}{c^2}\right) c^2 = 1.938 \text{ GeV} \quad (eq A. 2).$$

Next, we can use the mass-momentum-energy relationship for a particle with momentum p to find the following:

$$\sqrt{E^2} = \sqrt{(mc^2)^2 + (pc)^2} = E = 1.938\text{GeV} \quad (eq A. 3)$$

$$(pc)^2 = E^2 - (mc^2)^2 \quad (eq A. 4)$$

$$(pc)^2 = (1.938)^2 - (0.938)^2 = 2.876 \text{ GeV}^2 \quad (eq A. 5)$$

$$pc = 1.696 \text{ GeV} \quad (eq A. 6)$$

$$p = \frac{1.696 \text{ GeV}}{c} = \gamma m v \quad (\text{eq A. 7}).$$

Since we know that the particle in question is approaching the speed of light, it is appropriate to restate velocity v as βc , which finally yields equation A.8

$$\gamma m \beta c = \frac{1.696 \text{ GeV}}{c} \quad (\text{eq A. 8})$$

Returning to another definition of the energy of a mass traveling near the speed of light, we know that

$$E = \gamma m c^2 \quad (\text{eq A. 9})$$

which through rearranging and substituting becomes

$$\gamma = \frac{E}{m c^2} = \frac{1.938 \text{ GeV}}{\left(0.938 \frac{\text{GeV}}{c^2}\right) c^2} = 2.066 \quad (\text{eq A. 10}).$$

So now that we know the value for γ , we can express equation A.8 as

$$(2.066) \left(0.938 \frac{\text{GeV}}{c^2}\right) \beta c = \frac{1.696 \text{ GeV}}{c} \quad (\text{eq A. 11})$$

and solving for β we arrive at equation A.12:

$$\beta = \frac{1.696}{(2.066)(0.938)} = 0.875 \quad (\text{eq A. 12}).$$

With this new found knowledge concerning β , we see that

$$v = \beta c = (0.875) \left(3 \times 10^8 \frac{m}{s}\right) = 2.625 \times 10^8 \frac{m}{s} \quad (\text{eq A. 13}).$$

This particle velocity v can be applied to the Accumulator Ring length l of 248 meters to return Ring period τ and Ring frequency f

$$\tau = \frac{l}{v} = \frac{248 \text{ m}}{2.625 \times 10^8 \frac{\text{m}}{\text{s}}} = 945 \text{ ns} \quad (\text{eq A. 14})$$

$$f = \frac{1}{\tau} = \frac{1}{945 \times 10^{-9} \text{ s}} = 1.058467 \text{ MHz} \quad (\text{eq A. 15})$$

Appendix B

Proof of In-Phase/Quadrature Modulation Linearity

Starting with the input signal

$$V(t) = A(t) \cos(\omega_c t + \varphi(t)) \quad (\text{eq B.1})$$

where $A(t)$ is the signal envelope, ω_c is the center frequency, and $\varphi(t)$ is the time-varying phase

$$I(t) = A(t) \cos(\varphi(t)) \quad (\text{eq B.2}) \quad Q(t) = A(t) \sin(\varphi(t)) \quad (\text{eq B.3})$$

define the in-phase and quadrature signal components, relating to input signal $V(t)$ as follows:

$$V(t) = I(t) \cos(\omega_c t) - Q(t) \sin(\omega_c t) \quad (\text{eq B.4})$$

$$V(t) = A(t) \cos(\varphi(t)) \cos(\omega_c t) - A(t) \sin(\varphi(t)) \sin(\omega_c t) \quad (\text{eq B.5})$$

$$V(t) = \frac{1}{2} A(t) [\cos(\varphi(t) - \omega_c t) + \cos(\varphi(t) + \omega_c t)] \quad (\text{eq B.6})$$

$$- \frac{1}{2} A(t) [\cos(\varphi(t) - \omega_c t) - \cos(\varphi(t) + \omega_c t)]$$

$$V(t) = A(t) \cos(\omega_c t + \varphi(t)) \quad (\text{eq B.7})$$

Appendix C

Schlicher-Traditional Discrete-Time PID Controller Equivalence

Starting with equation 3.12, we know that

$$\frac{u(z)}{e(z)} = \frac{[K_p + K_i + K_d] - z^{-1}[K_p + 2K_d] + z^{-2}K_d}{1 - z^{-1}} \quad (\text{eq C.1})$$

for Schlicher's discrete-time PID controller [11]. The following equations show the progression from equation C.1 to the more traditional form seen in equation C.2:

$$\frac{u(z)}{e(z)} = K_p + (1 - z^{-1})K_D + \frac{1}{(1 - z^{-1})}K_I \quad (\text{eq C.2}).$$

$$\frac{u(z)}{e(z)} = \frac{[K_p + K_i + K_d] - z^{-1}[K_p + 2K_d] + z^{-2}K_d}{1 - z^{-1}} \quad (\text{eq C.1})$$

$$u(z)(1 - z^{-1}) = e(z)([K_p + K_i + K_d] - z^{-1}[K_p + 2K_d] + z^{-2}K_d) \quad (\text{eq C.3})$$

$$u(z)(1 - z^{-1}) = e(z)(K_p + K_i + K_d - z^{-1}K_p - 2z^{-1}K_d + z^{-2}K_d) \quad (\text{eq C.4})$$

$$u(z) = e(z) \frac{(K_p(1 - z^{-1}) + K_i + K_D(1 - z^{-1})^2)}{1 - z^{-1}} \quad (\text{eq C.5})$$

$$u(z) = e(z) \left[K_p + \frac{K_i}{1 - z^{-1}} + K_D(1 - z^{-1}) \right] \quad (\text{eq C.6})$$

$$\frac{u(z)}{e(z)} = K_p + \frac{1}{(1 - z^{-1})}K_i + (1 - z^{-1})K_D \quad (\text{eq C.7}).$$

Appendix D

Selected Rise-Time, Settling-Time, and Overshoot Data

Preface

This appendix contains selected model simulation data concerning rise time, settling time, and overshoot across a range of proportional, integral, and derivative gain values. Each figure includes both a 3-dimensional perspective representation of the plotted data as well as a top-down view. Take special note of the scales for each plot before drawing comparisons; due to the volume of data being processed, the plot axis scales were automatically generated based on the current data rather than hand-selected for each comparison case.

Rise Time is calculated as the number of discrete-time simulation samples from time-zero until the first sample that reaches or exceeds 99% of the final, steady-state value. Settling time is the number of discrete-time simulation samples until the signal settles to within $\pm 1\%$ of the steady-state value. Overshoot is calculated as the percent that the maximum value for a given signal exceeds the steady-state value by.

Please note that though an effort has been made to cull unstable simulation results from the included data, the sheer number of performed simulations prevented individual verification in every case for invalid results. Because of this, the results herein are best used to understand trends and broad-scale system operation rather than to examine precise numerical phenomena. Also, if a plot in this appendix has sparse data, as a rule-of-thumb, simply assume that the ‘missing’ results were unstable or out of range.

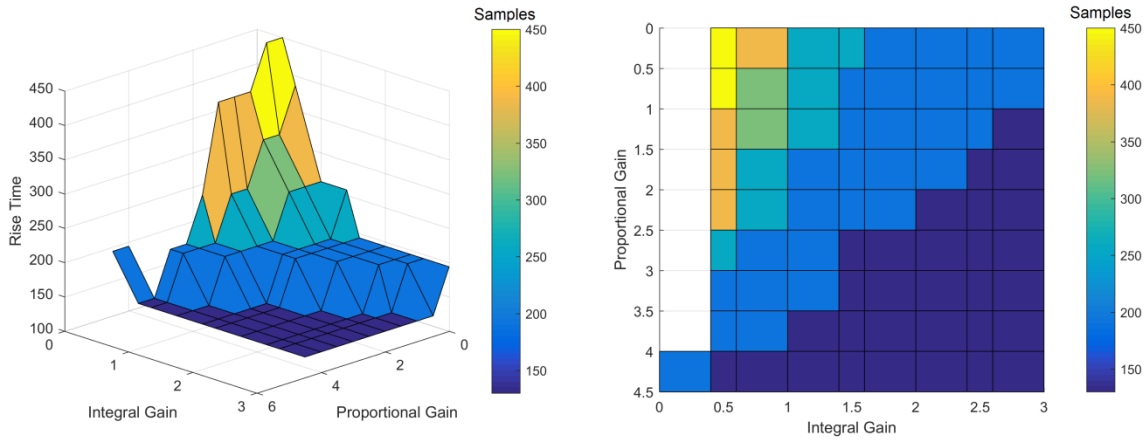


Figure D.1 Floating Point Rise Time, $K_d = 0$

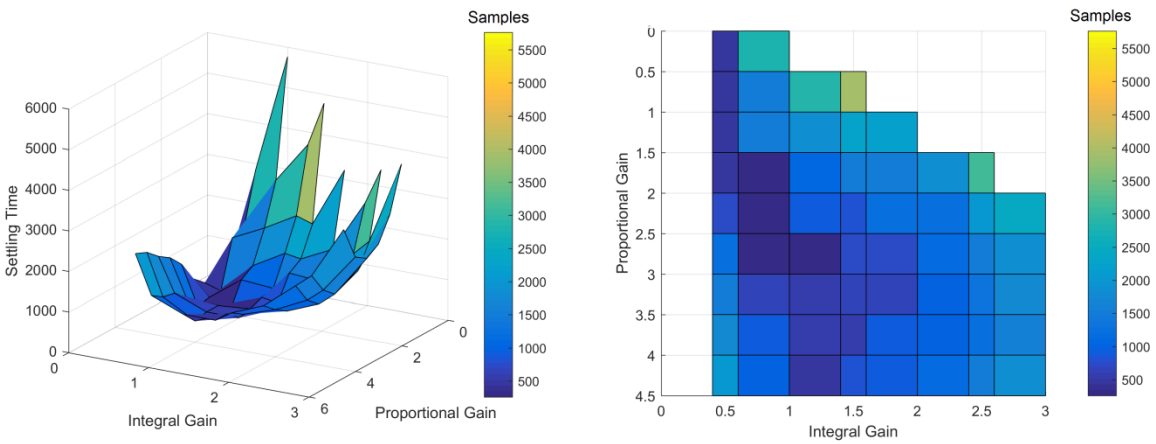


Figure D.2 Floating Point Settling Time, $K_d = 0$

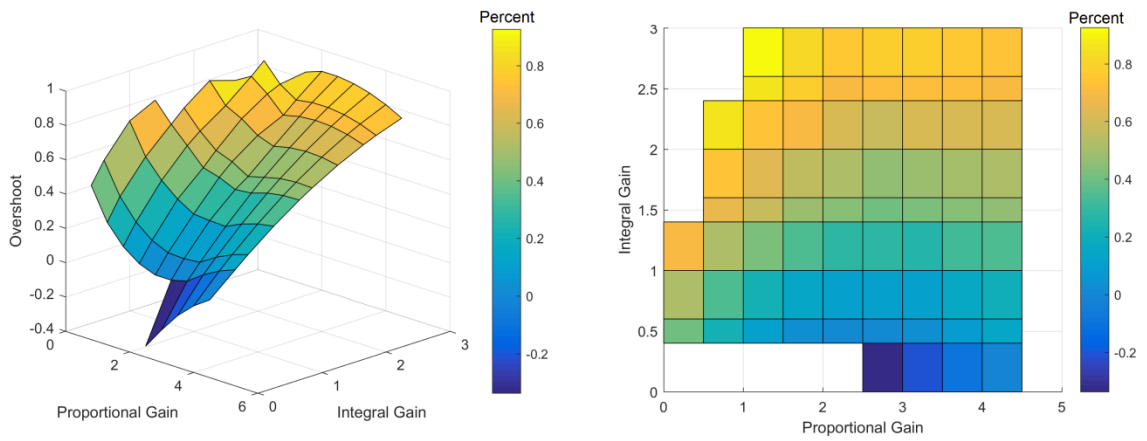


Figure D.3 Floating Point Overshoot, $K_d = 0$

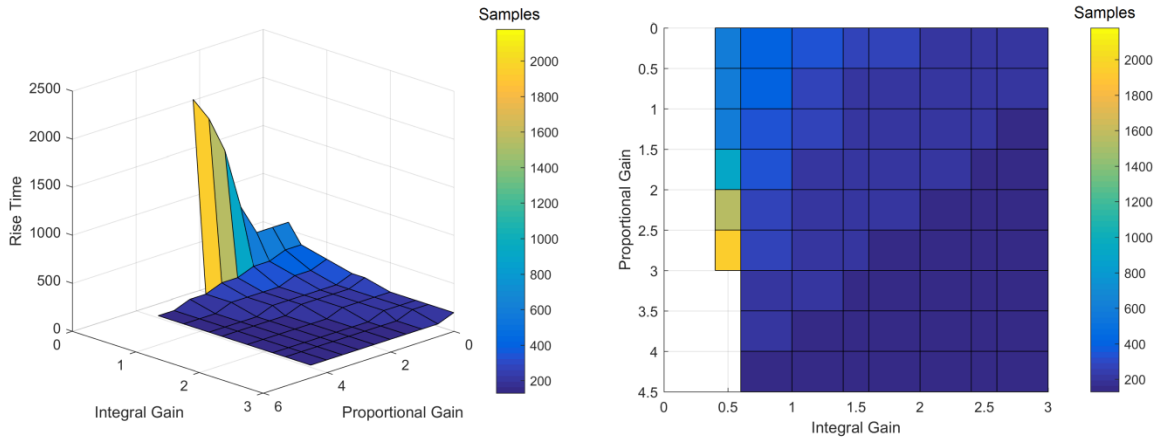


Figure D.4 Floating Point Rise Time, $K_d = 0.1$

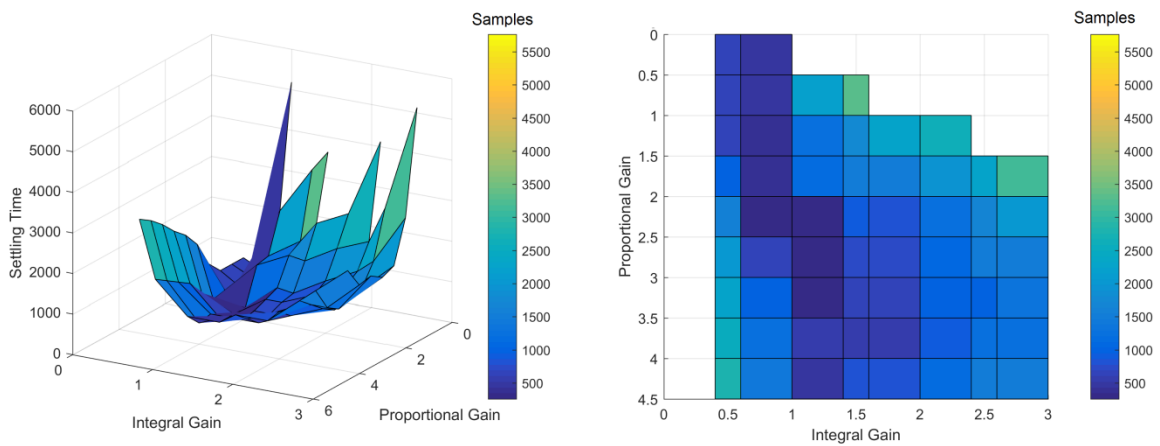


Figure D.5 Floating Point Settling Time, $K_d = 0.1$

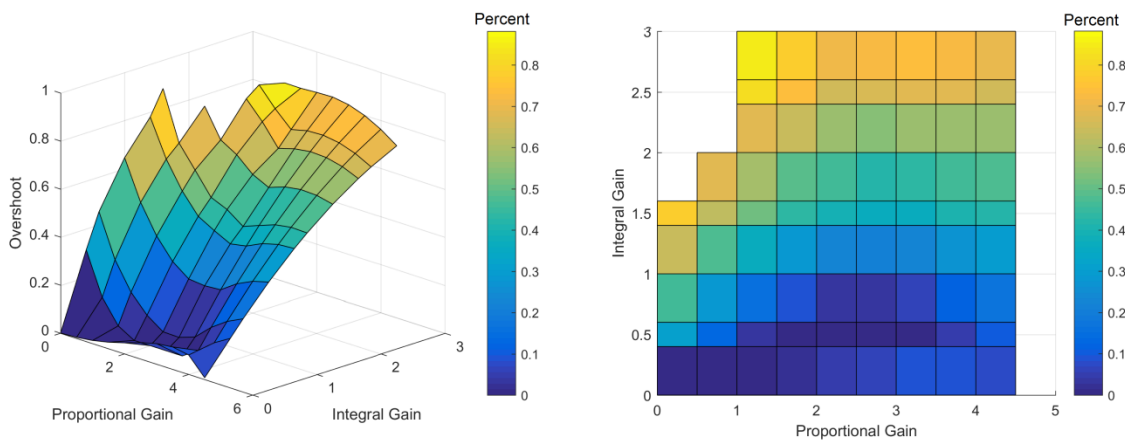


Figure D.6 Floating Point Overshoot, $K_d = 0.1$

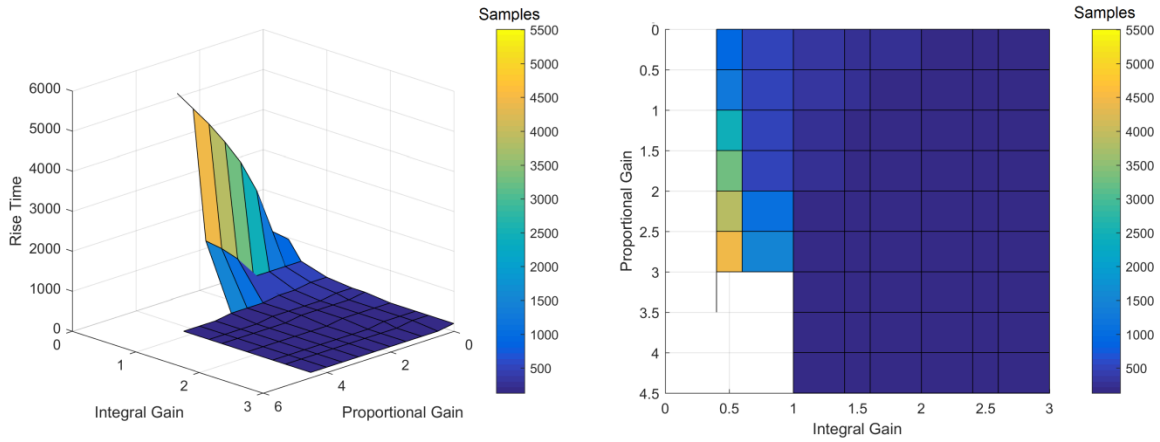


Figure D.7 Floating Point Rise Time, $K_d = 0.25$

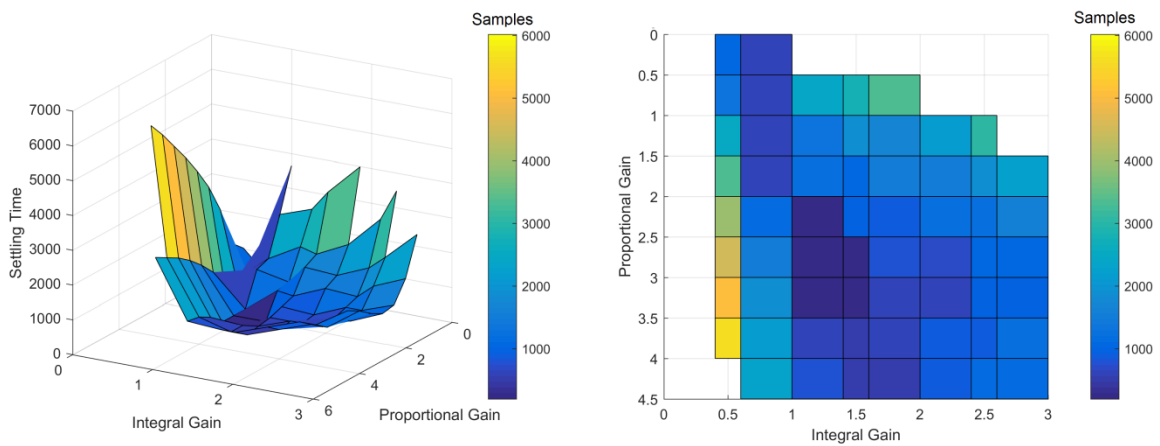


Figure D.8 Floating Point Settling Time, $K_d = 0.25$

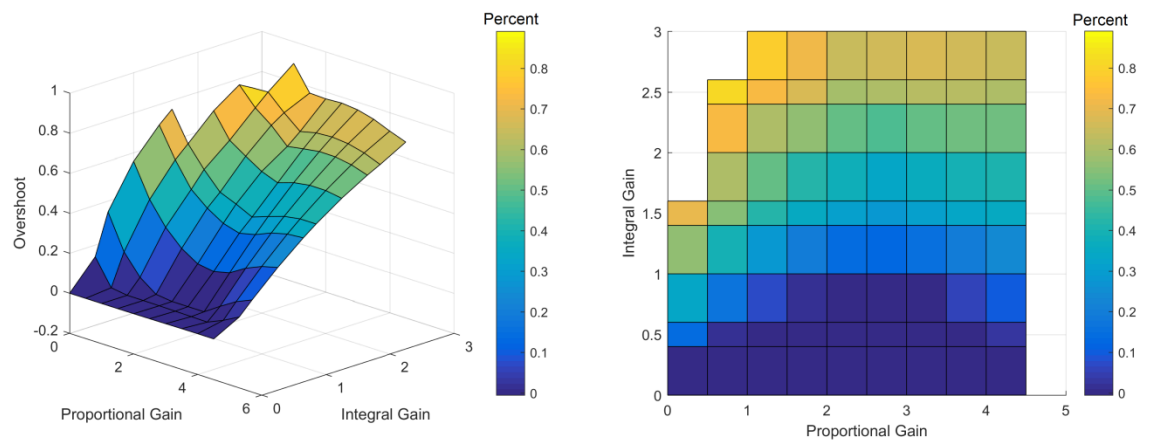


Figure D.9 Floating Point Overshoot, $K_d = 0.25$

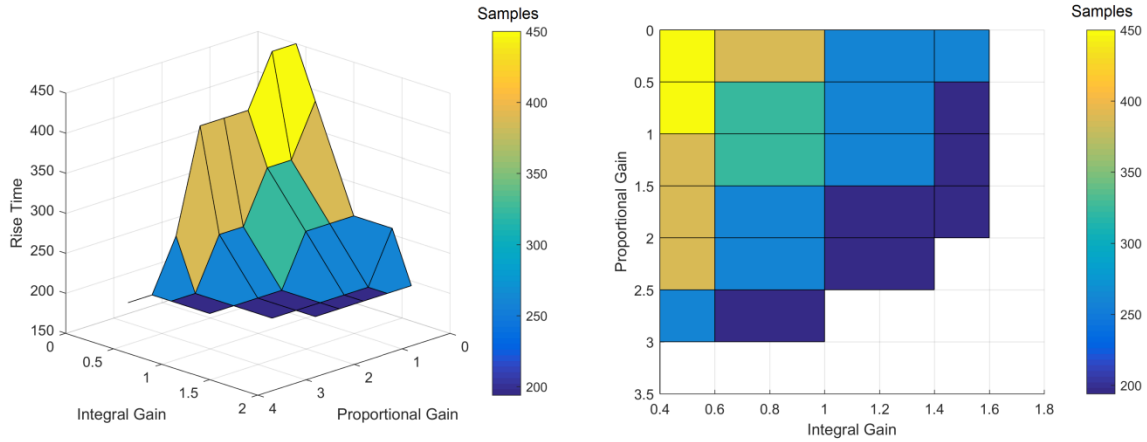


Figure D.10 Fixed Point Rise Time, $K_d = 0$

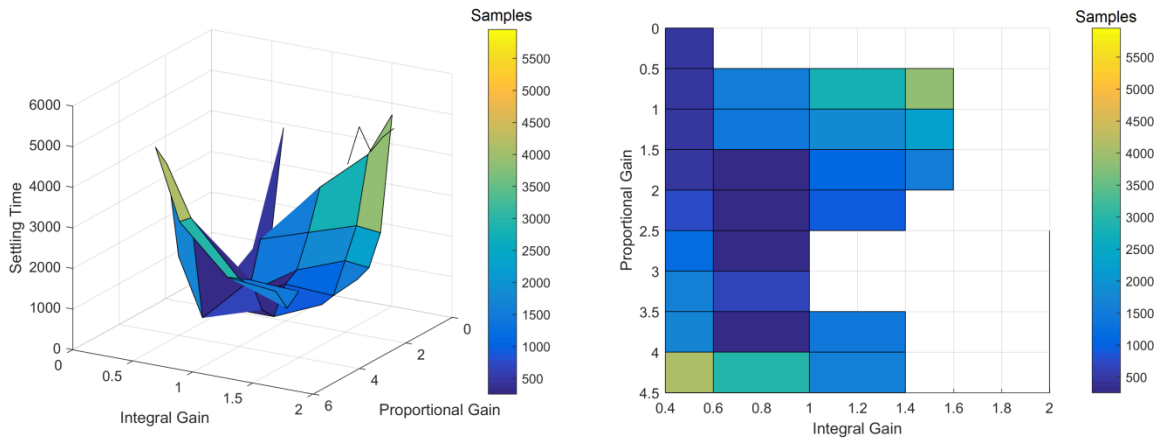


Figure D.11 Fixed Point Settling Time, $K_d = 0$

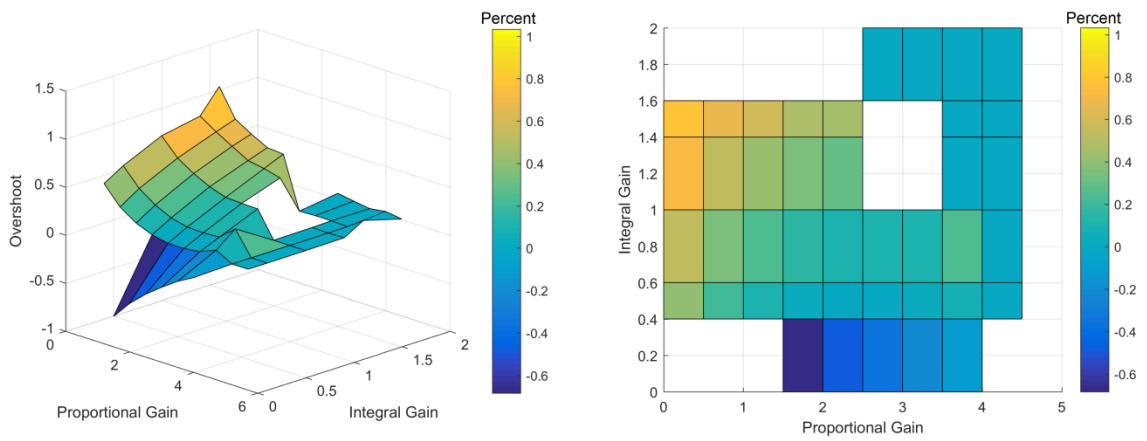


Figure D.12 Fixed Point Overshoot, $K_d = 0$

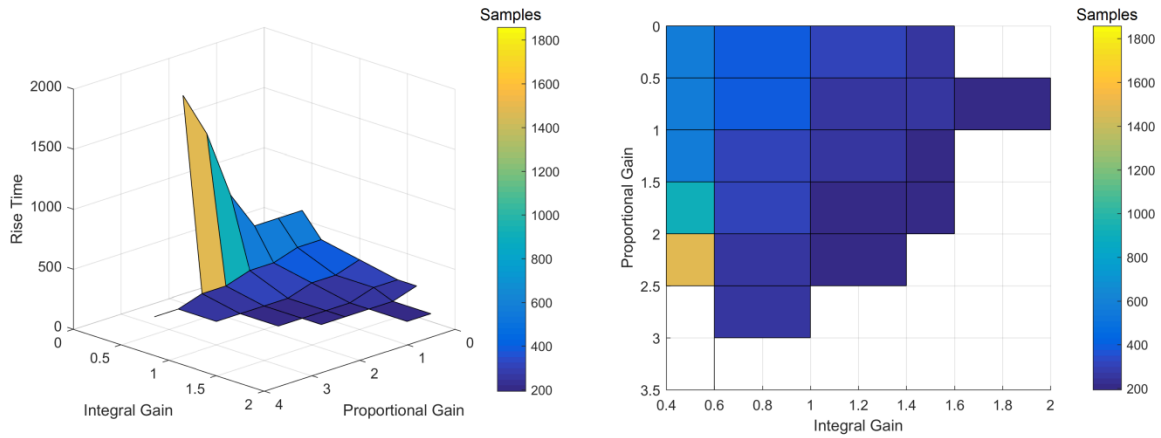


Figure D.13 Fixed Point Rise Time, $K_d = 0.1$

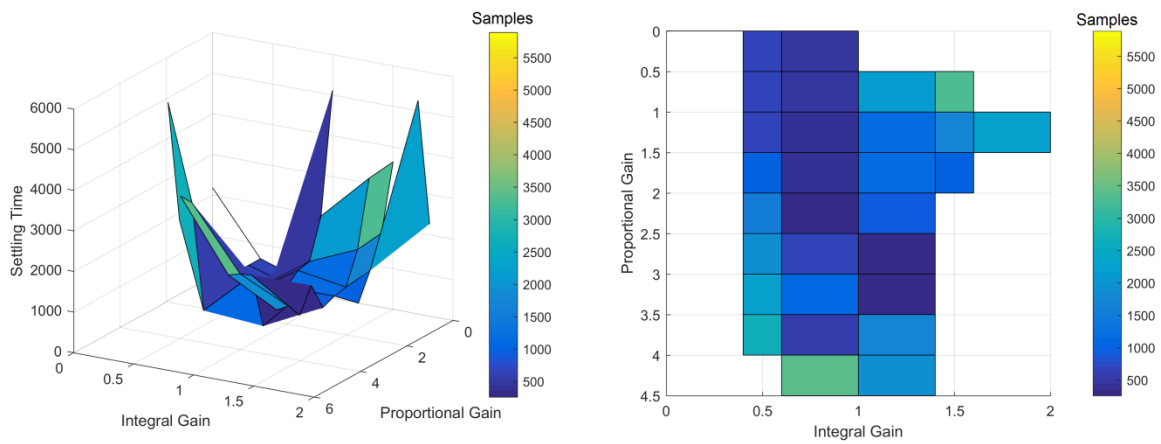


Figure D.14 Fixed Point Settling Time, $K_d = 0.1$

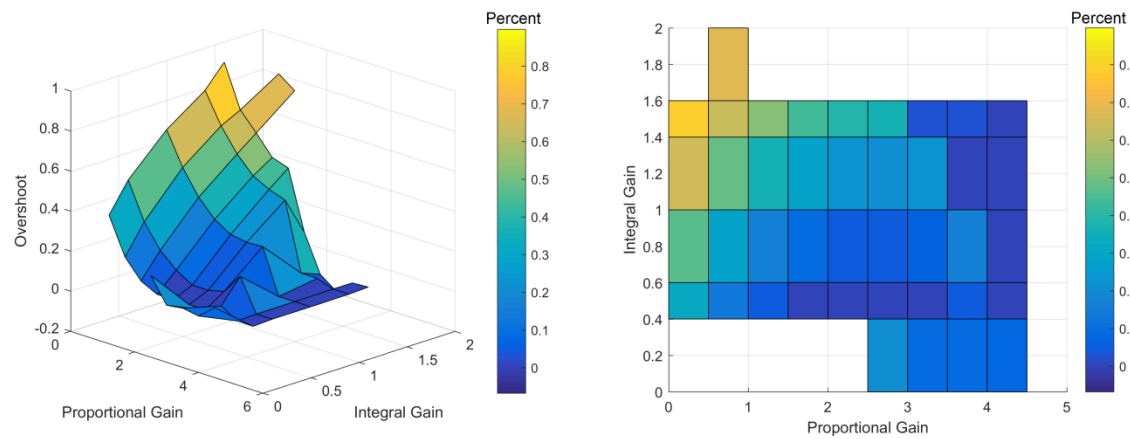


Figure D.15 Fixed Point Overshoot, $K_d = 0.1$

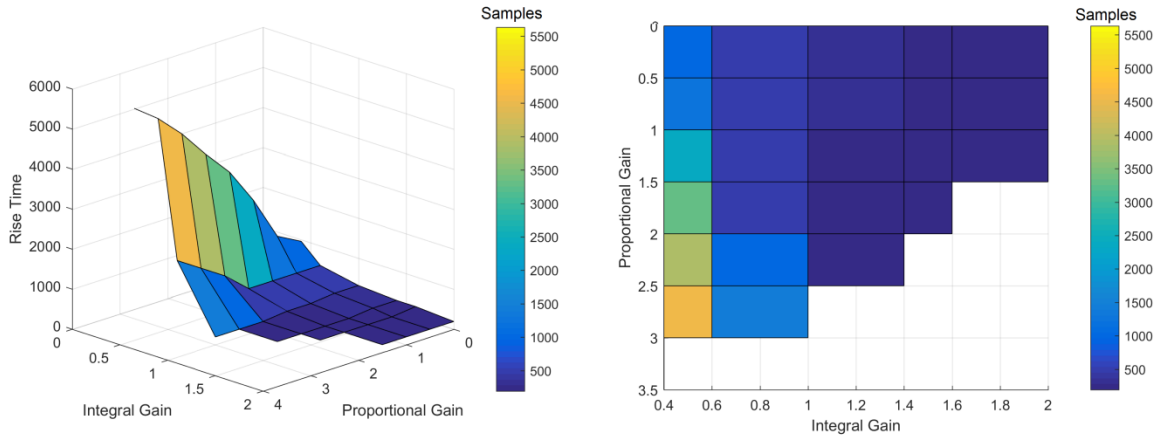


Figure D.16 Fixed Point Rise Time, $K_d = 0.25$

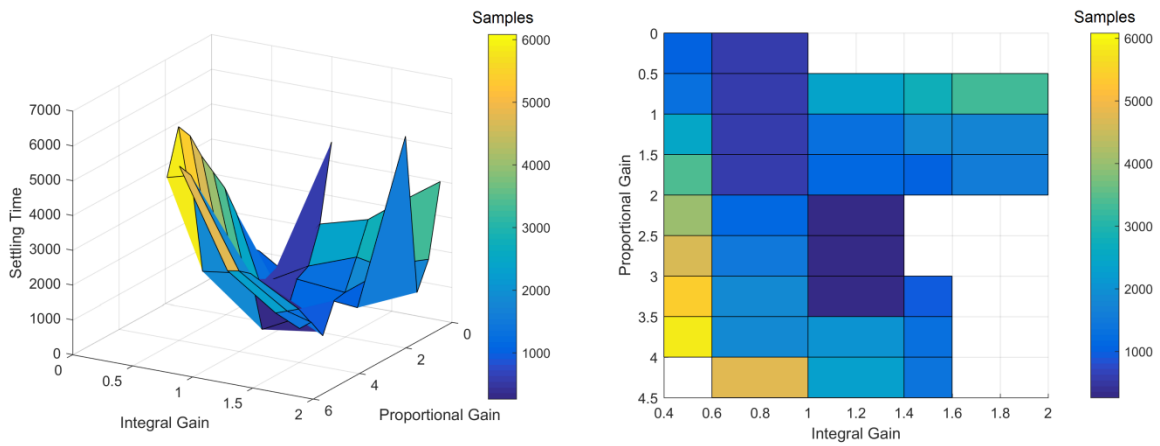


Figure D.17 Fixed Point Settling Time, $K_d = 0.25$

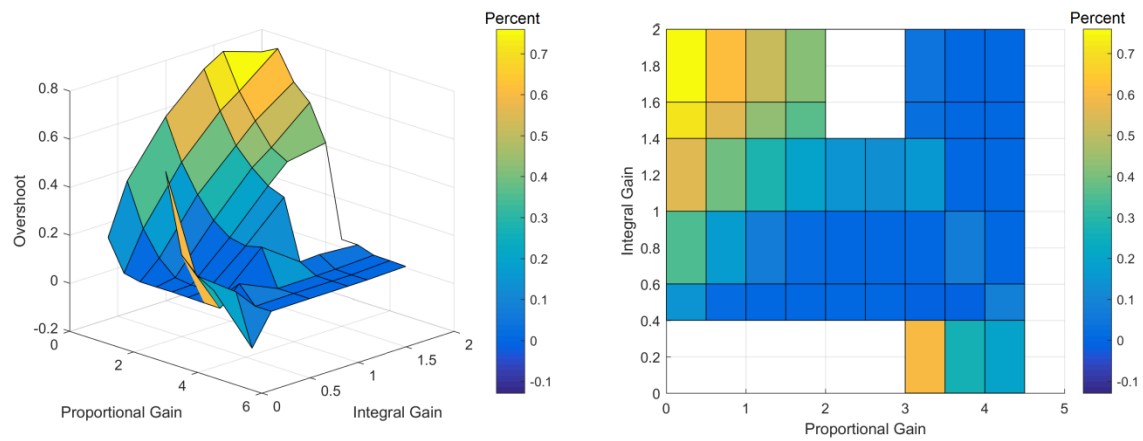


Figure D.18 Fixed Point Overshoot, $K_d = 0.25$

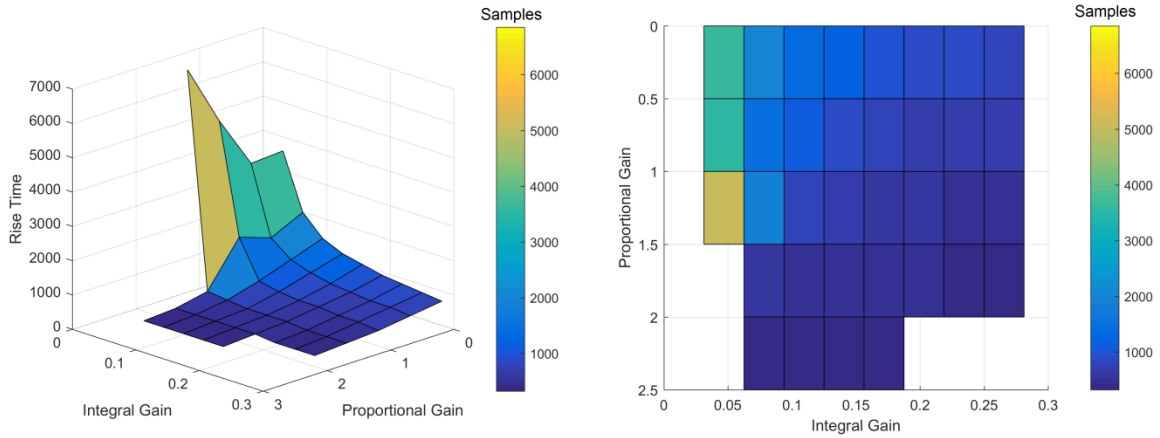


Figure D.19 Fixed Point Delay Rise Time, $K_d = 0$

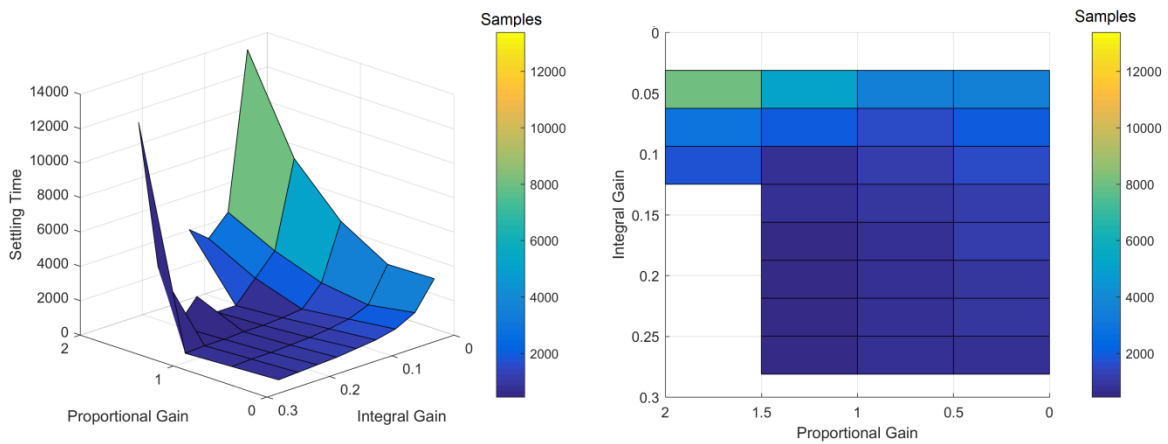


Figure D.20 Fixed Point Delay Settling Time, $K_d = 0$

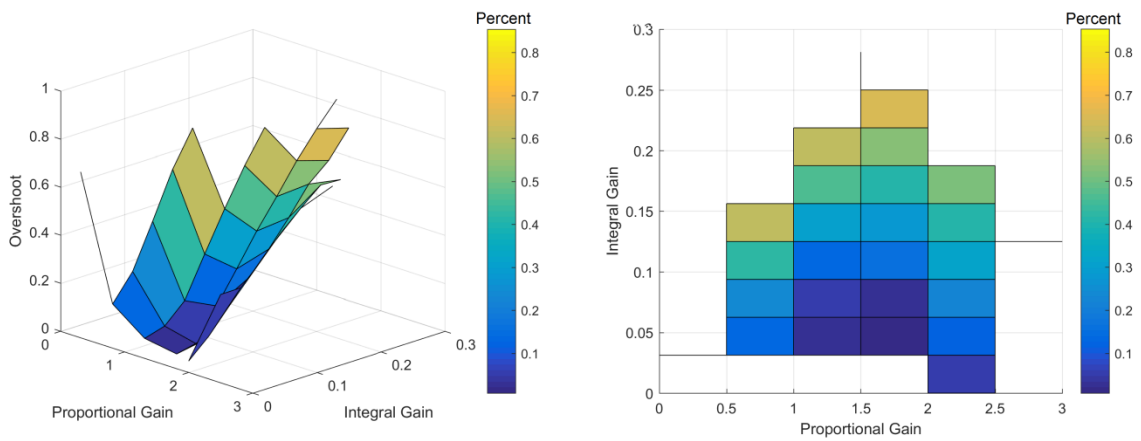


Figure D.21 Fixed Point Delay Overshoot, $K_d = 0$

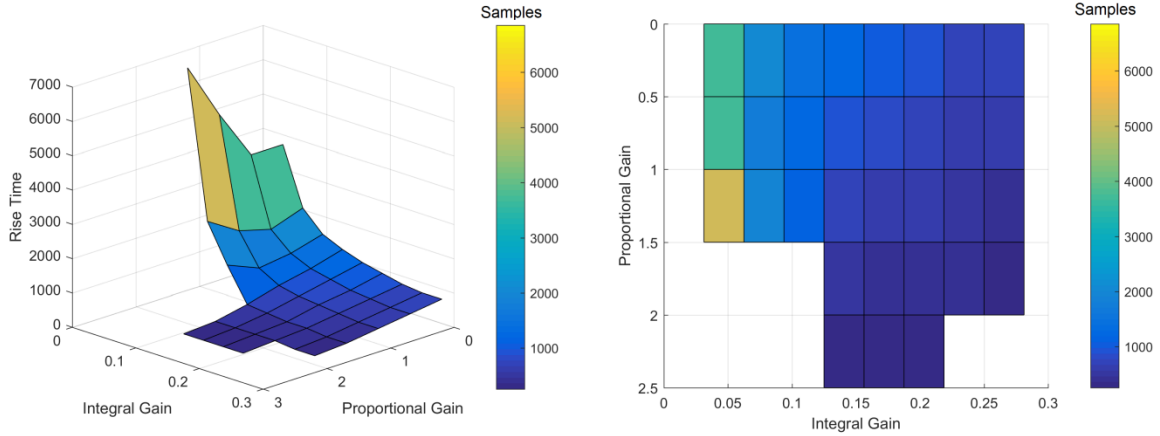


Figure D.22 Fixed Point Delay Rise Time, $K_d = 1.5$

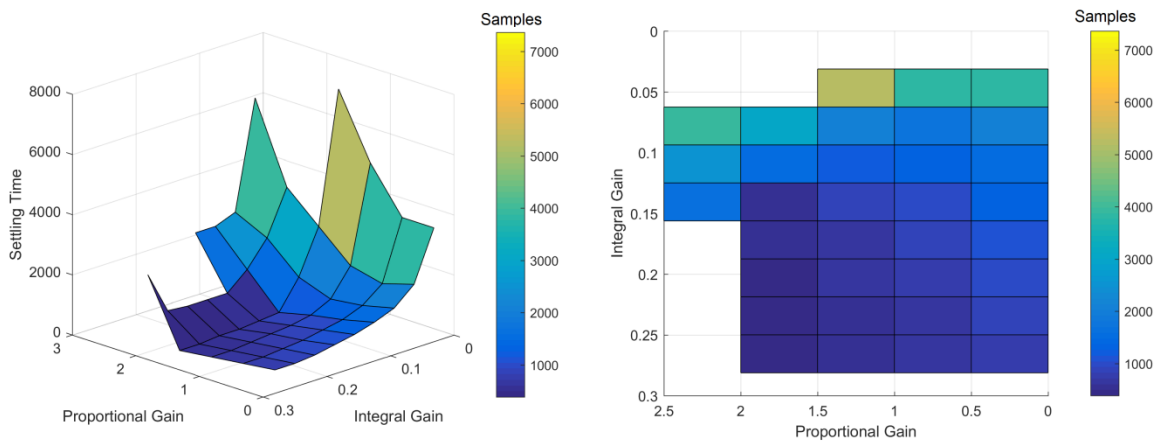


Figure D.23 Fixed Point Delay Settling Time, $K_d = 1.5$

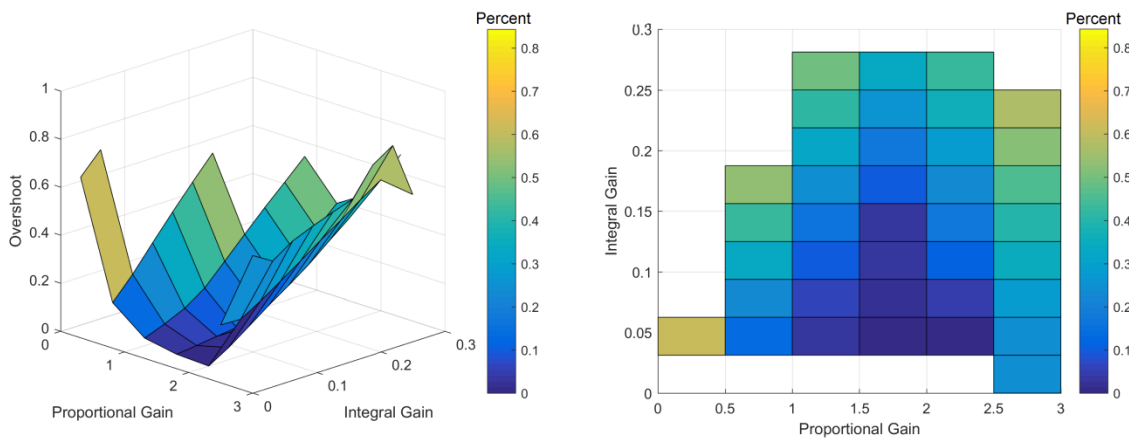


Figure D.24 Fixed Point Delay Overshoot, $K_d = 1.5$

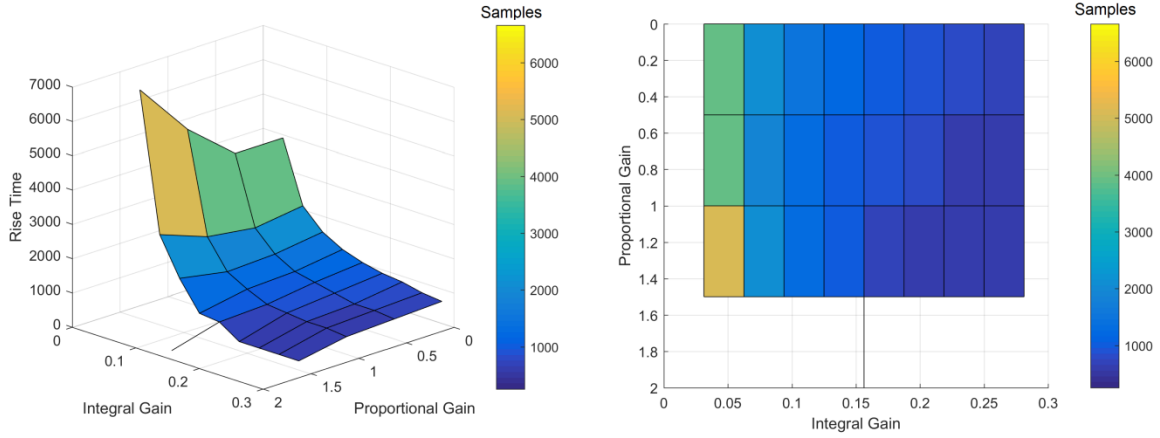


Figure D.25 Fixed Point Delay Rise Time, $K_d = 3$

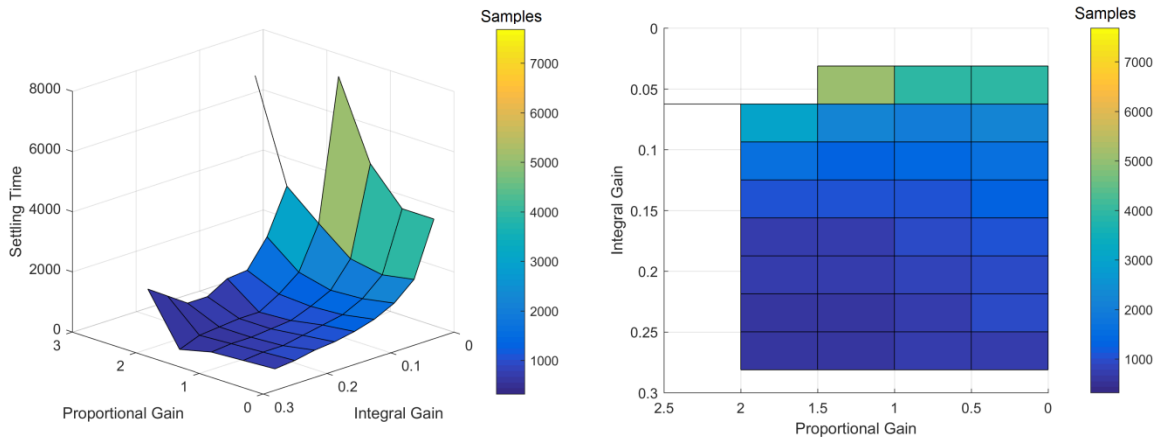


Figure D.26 Fixed Point Delay Settling Time, $K_d = 3$

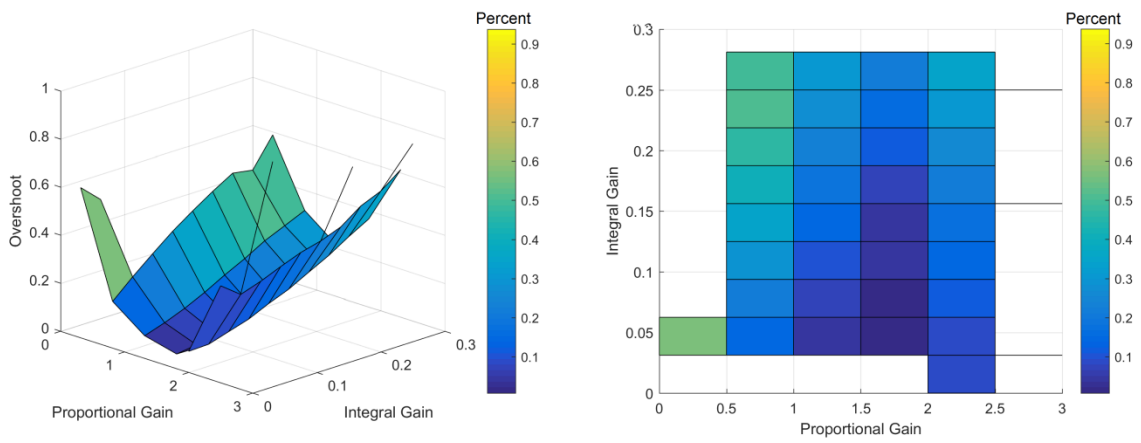


Figure D.27 Fixed Point Delay Overshoot, $K_d = 3$

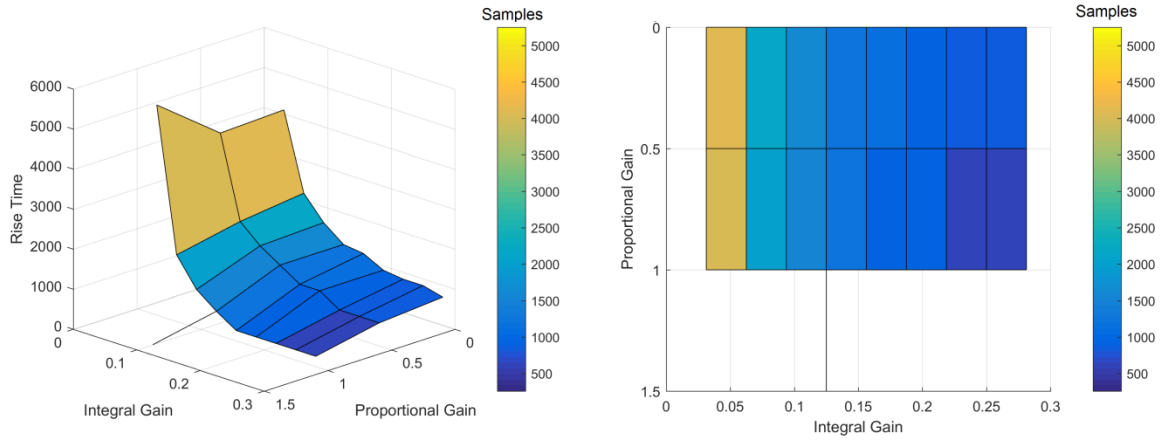


Figure D.28 Fixed Point Delay Rise Time, $K_d = 4.5$

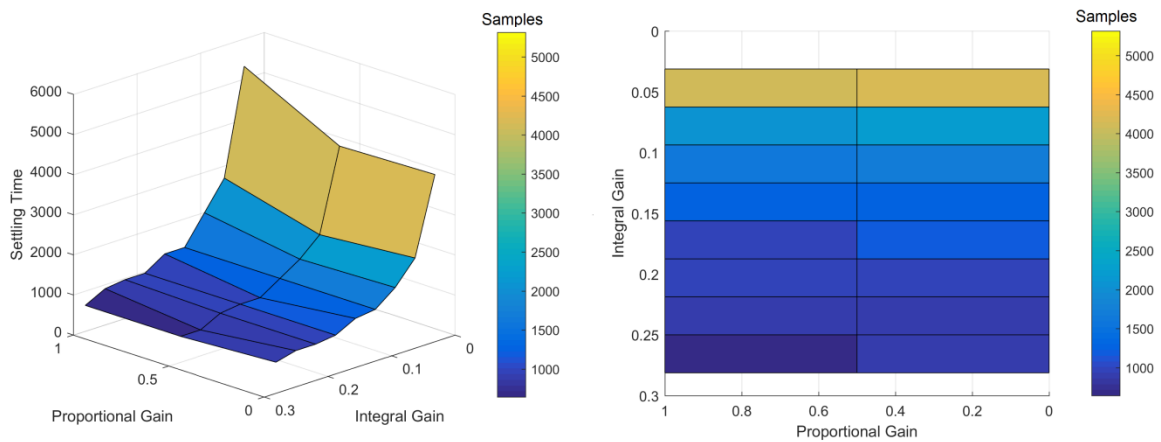


Figure D.29 Fixed Point Delay Settling Time, $K_d = 4.5$

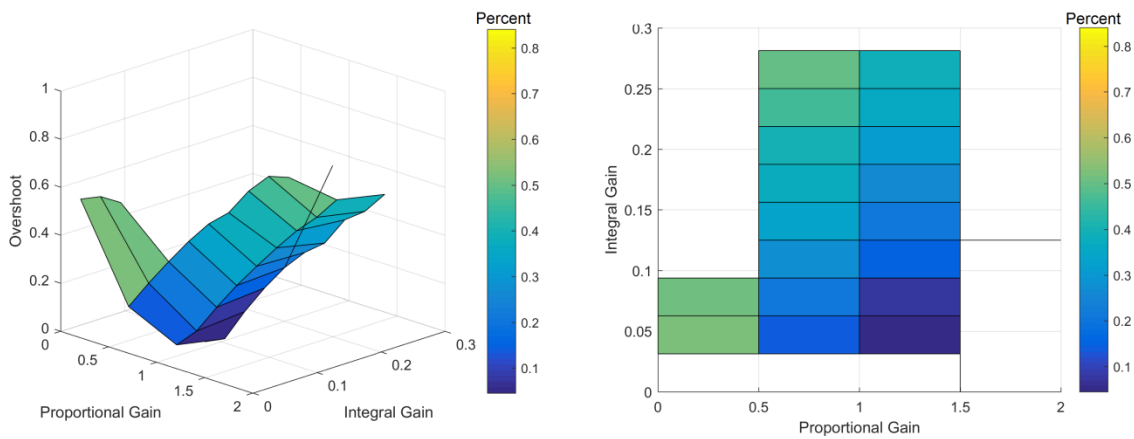


Figure D.30 Fixed Point Delay Overshoot, $K_d = 4.5$

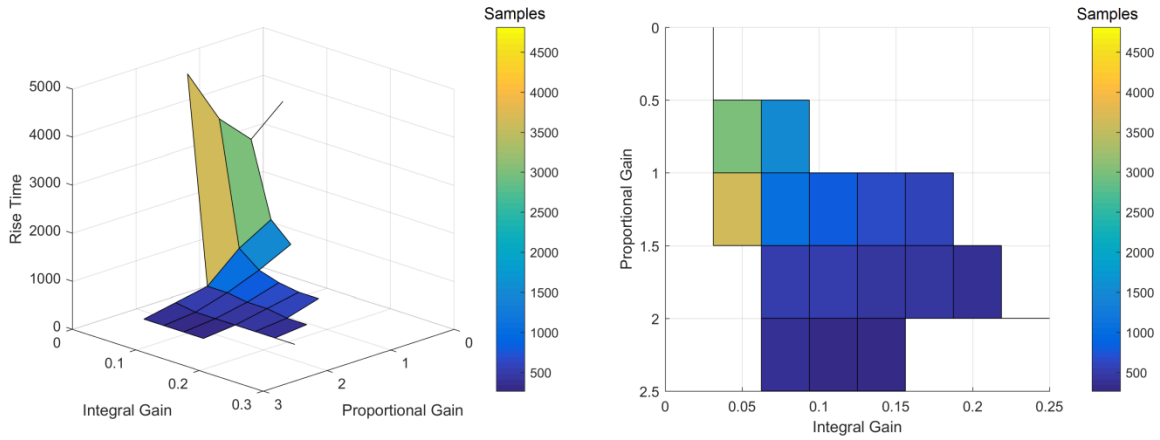


Figure D.31 System Generator Rise Time, $K_d = 0$

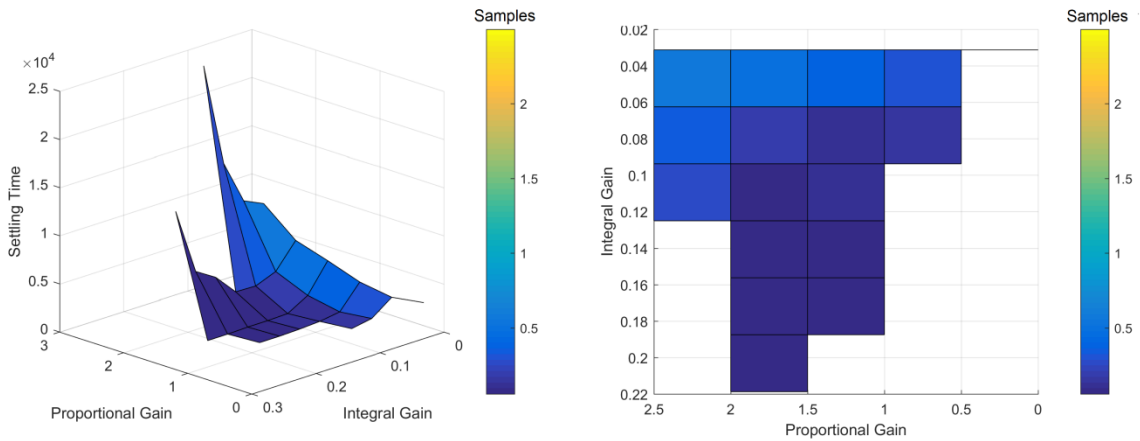


Figure D.32 System Generator Settling Time, $K_d = 0$

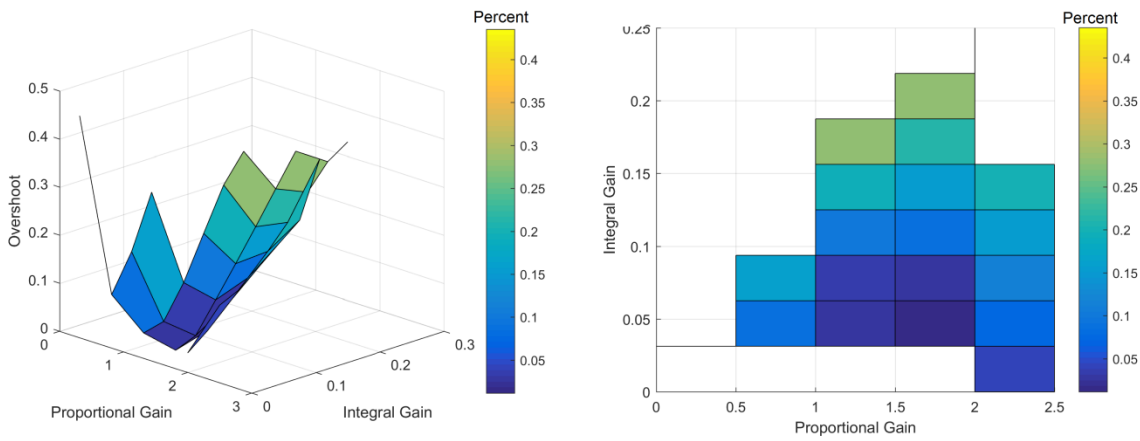


Figure D.33 System Generator Overshoot, $K_d = 0$

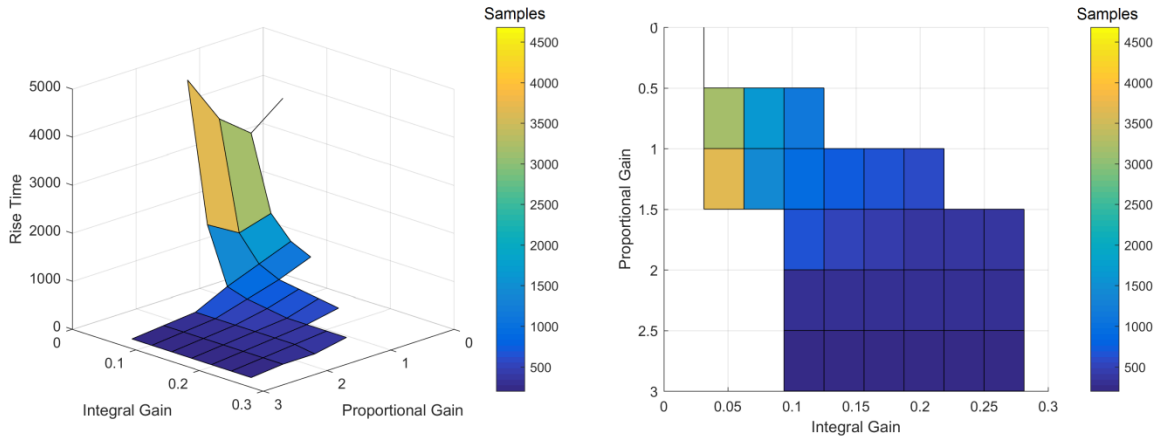


Figure D.34 System Generator Rise Time, $K_d = 1.5$

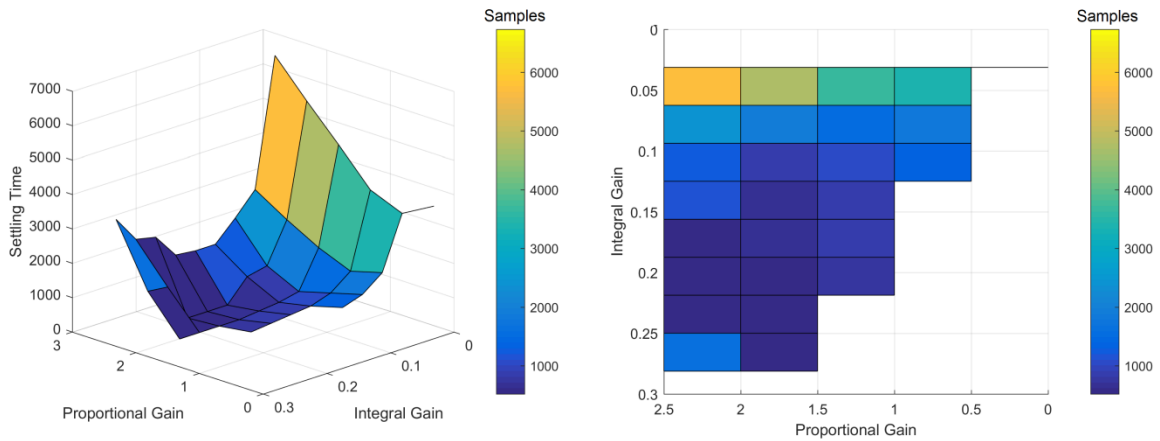


Figure D.35 System Generator Settling Time, $K_d = 1.5$

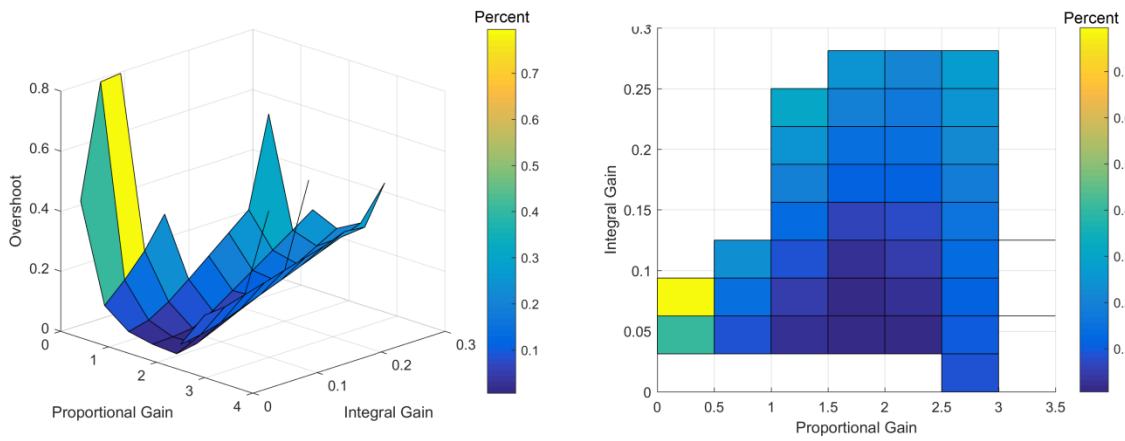


Figure D.36 System Generator Overshoot, $K_d = 1.5$

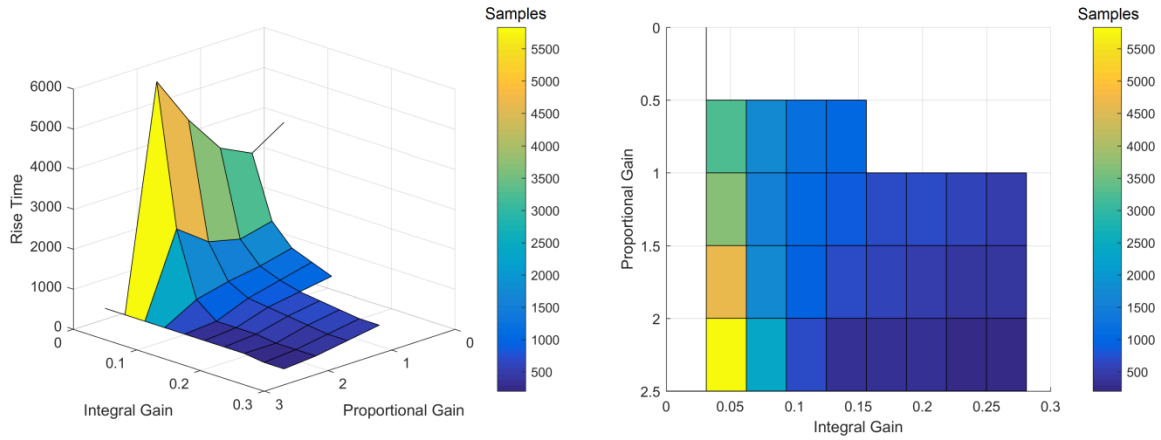


Figure D.37 System Generator Rise Time, $K_d = 3$

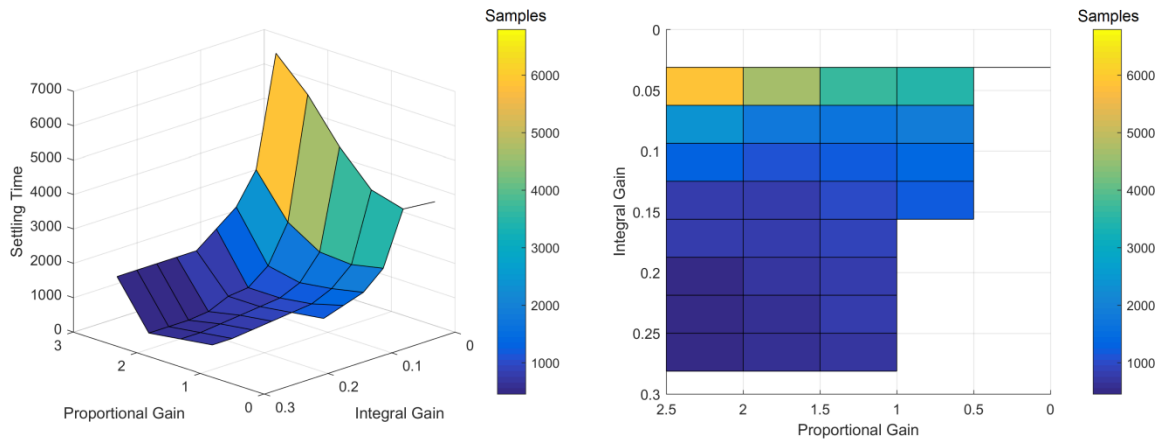


Figure D.38 System Generator Settling Time, $K_d = 3$

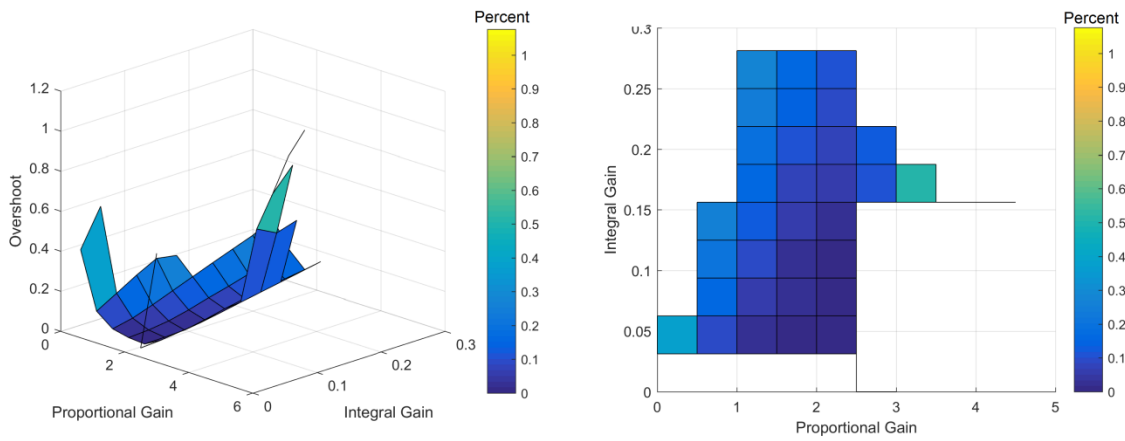


Figure D.39 System Generator Overshoot, $K_d = 3$

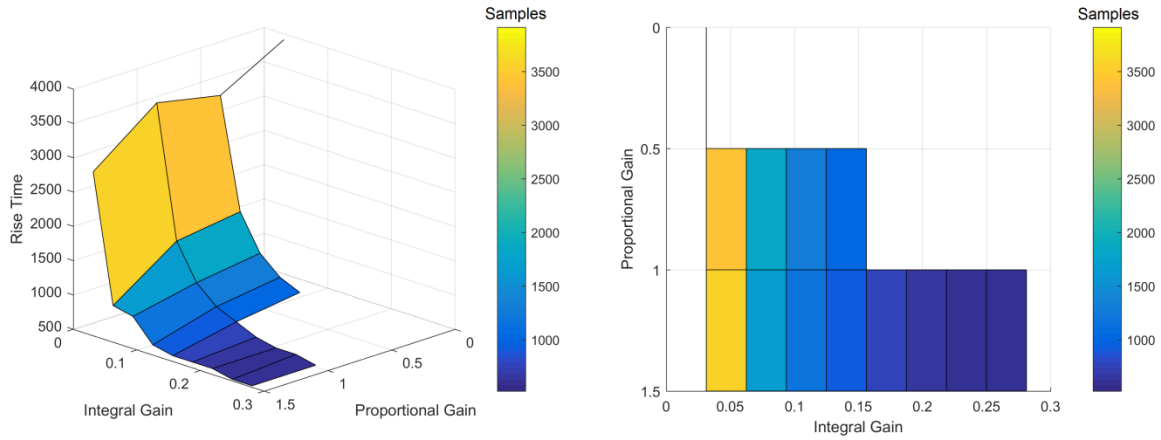


Figure D.40 System Generator Rise Time, $K_d = 4.5$

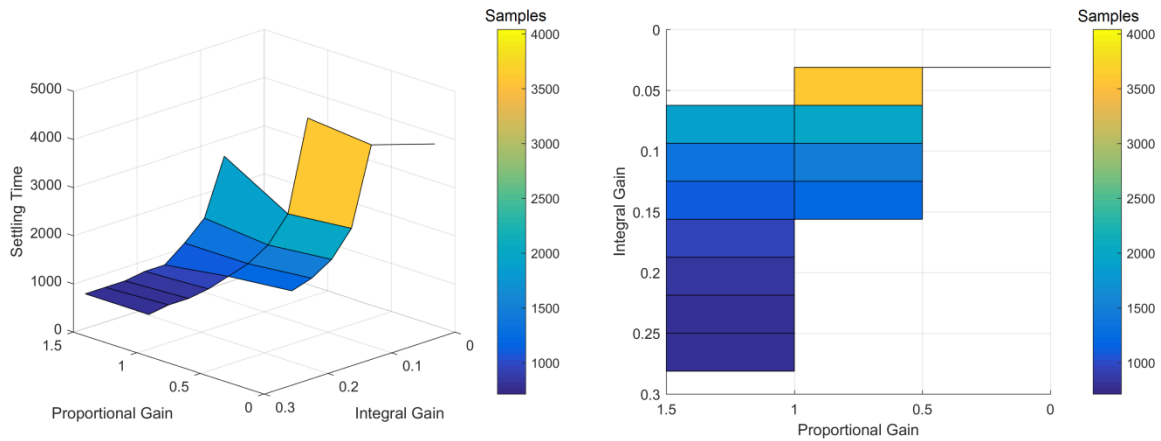


Figure D.41 System Generator Settling Time, $K_d = 4.5$

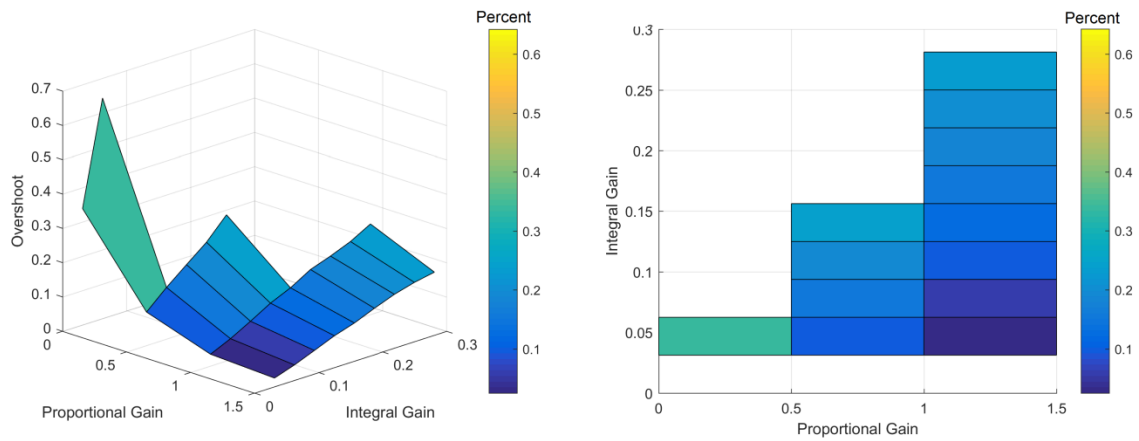


Figure D.42 System Generator Overshoot, $K_d = 4.5$

Vita

Michael Gabriel Trout was born in Knoxville, TN to parents Thomas W. Trout III and Debra D. Trout in March of 1994. Michael and his two older siblings were homeschooled by their mother Debra, and after graduating high school, Michael went on to receive his Bachelor of Science degree in Electrical Engineering from the University of Tennessee in Knoxville in May 2016. Immediately following undergraduate studies, Michael began work at Spallation Neutron Source at Oak Ridge National Laboratory on the Hardware Team of the Control Systems Group. Michael began graduate school at the University of Tennessee in Knoxville in August of 2016 as a part of the five-year BS/MS program and plans to graduate with a Master of Science in Electrical Engineering degree in August of 2017. His focus areas include Controls, Hardware Design, and Radio-Frequency Systems.