A New Behavioral Model for Spurious Tone Prediction in Fractional-N PLLs

by

Darren R. Frenette, B.Eng.

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Masters of Applied Science

in

Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

Carleton University

Department of Electronics

Ottawa, Ontario

Your file Votre référence
ISBN: 978-0-494-93476-0

Our file Notre référence
ISBN: 978-0-494-93476-0

Abstract

This thesis presents two behavioral models that are capable of predicting spur amplitudes and locations in Fractional-N (Frac-N) PLLs. Traditional PLL models consists of linearized components. While this is acceptable for phase noise analysis, spur prediction requires certain PLL non-linearities to be modeled.

The irregular nature of the charge pump pulse is one non-linearity. The proposed models simplify this pulse somewhat, but maintain its pulsed nature. In contrast, traditional behavioral models typically average the pulse across an entire reference cycle.

The irregular clocking of the Sigma Delta Modulator (SDM) is another non-linearity. With a Frac-N PLL the feedback signal is constantly trying to synchronize with the reference signal as the SDM changes divide values. This results in an inconsistent clock driving the SDM. The proposed model includes this varying clock.

One final potential non-linearity is wideband FM modulation from the VCO. This non-linearity is not modeled, but the ramifications of it are discussed. An understanding of this is needed in order to anticipate how spurs at the VCO input will appear at the VCO output. An equation is presented that predicts spur amplitude at the VCO output based on its amplitude at the VCO input.

One major benefit of the proposed models is that simulation times are reduced many times over a typical behavioral model. The proposed models are generally capable of predicting spurs amplitude with an accuracy of better than ± 3dB up to the first harmonic of the reference frequency.

The benefits of the proposed models in this thesis over other models are fast, accurate prediction of Frac-N spurs up to the first reference frequency harmonic as well as support for 1^{st}, 2^{nd}, and 3^{rd} order sigma-delta modulators. Other papers may only predict spurs at frequencies close to the carrier, support only 1^{st} order sigma-delta modulators, or predict only phase noise and not Frac-N spurs.

This thesis also discusses the true source of reference spurs. Additionally, it offers empirical equations to calculate the sequence length of a subset of 2^{nd} order sigma-delta modulators. It also offers an equation used to determine Frac-N spur spacing based on the sigma-delta modulator sequence length.

It was noted that one of the simplifications of the proposed models greatly reduces reference spurs. Based on this observation, a design is proposed that can be used to reduce reference spurs in a Fractional-N PLL. Additionally a technique is developed that can reduce spurs at any desired frequency outside of the first harmonic of the reference frequency.

Acknowledgements

I would first like to thank Dr. John Rogers. His expertise and enthusiasm for the subject material was inspiring. He guided my studies in such a way that I was always in a position to learn while working on interesting topics. His availability and willingness to help made for smooth transitions from periods of difficulty.

I feel very fortunate to have been able to work with Dr. Ardeshir Namdar during this thesis. Ardeshir is experienced in my area of study, and I have learned a great deal from our many discussions. George Khoury has also been very helpful. He has always been willing to answer any question I may have. I am honored to be involved with a company with such talented individuals as Ardeshir and George. Both Ardeshir and George acted as a champion for me, and for that I will always be grateful.

John, Ardeshir, and George have all helped to make my time in graduate studies an exciting and fulfilling endeavor.

I would also like to thank Skyworks for providing me with financial support as well as access to their facilities throughout my thesis.

I would like to dedicate this thesis to my beautiful son Keegan. My greatest wish for you is happiness, love, and fulfillment in this life.

Finally, I would like to thank my partner in life Farrah, for her constant love, support, and for her understanding of the demands of a graduate degree.

Table of Contents

List of Figures

List of Tables

Abbreviations

ACG	Automatic Gain Control
ADC	Analog to Digital Converter
CP	Charge Pump
DFT	Discrete Fourier Transform
EFM	Error Feedback Module
FWVACP	Fixed Width Variable Amplitude Charge Pump
GCD	Greatest Common Divisor
IF	Intermediate Frequency
PFD	Phase Frequency Detector
PLL	Phase-locked Loop
PTG	Pulse Train Generator
RF	Radio Frequency
SDM	Sigma Delta Modulator
SNR	Signal-to-noise ratio
VCO	Voltage Controlled Oscillator

1 Introduction

Communication has played a crucial role in the progress of humanity over the last 150 years. From the advent of the telegraph in 1844, to radio and television, to digital communications and computer networks, we have become increasingly reliant on technology as a means of communication [1].

At the heart of many forms of communication lies the radio; a device used to send information from a transmitter to a receiver. In order for a radio to communicate properly both transmitter and receiver must agree on the frequency at which the communication will take place. A critical component in this frequency selection process is the phase-locked loop (PLL). The primary role of a PLL (in terms of a radio) is to provide a clean, stable output frequency that is relatively free of noise. For this reason a PLL may be referred to as a frequency synthesizer. Implementing a PLL that is completely free of noise is, simply put, an impossible task. Noise in a PLL will undoubtedly show up in the form of phase noise or spurious tones (spurs). Phase noise refers to small variations of the intended output frequency. Spurs are large unwanted frequency components that manifest themselves at some offset of the intended output frequency [2].

In a radio both phase noise and spurs have the ability to corrupt the signal that is being either transmitted or received. For this reason it is important to understand the noise that is generated from a particular design. Computer

models are generally used to simulate PLL noise in an effort to reduce the time and cost of developing a real-world circuit. Transistor-level models use accurate descriptions of transistor characteristics to model a circuit's behavior. Behavioral models simply describe in code what a particular circuit should do. Behavioral models are generally faster than their transistor level counterpart. The proposed behavioral models presented in this thesis improve simulation times even further while still providing an accurate prediction of spurious performance. Phase noise is not considered in the proposed models.

A PLL is a non-linear system, however for phase noise analysis it is common to linearize non-linear PLL components [2]. Linearization is permitted in this case as phase noise is a small-signal phenomenon. This thesis analyzes the spurs that result from the sigma-delta modulator (SDM) used in Fractional-N (Frac-N) synthesis. The switching of integer divider values that result from an SDM is often a large signal phenomenon. For this reason linearization of the PLL cannot be utilized, which significantly complicates analysis and is the primary reason simple expressions to accurately predict spur levels are elusive.

This thesis studies the most prominent spurs both in-band and out-of-band until the first reference frequency harmonic. Many radio architectures employ a PLL to act as a frequency synthesizer which feeds a mixer. It is the mixing of spurious components that contribute to degraded signal quality.

Frequencies past the first reference frequency harmonic are generally not important as their amplitude will be greatly attenuated by a loop filter [2].

1.1 Thesis Goals

The primary goal of this thesis is to provide a model that efficiently and accurately predicts Frac-N spur location and amplitude. Two Simulink-based models are presented that are capable of this. The first model (called the ***standard proposed model*** henceforth) is the most efficient but fails to predict spurs around the carrier and first reference frequency harmonic accurately. The second model (called the ***hybrid proposed model*** henceforth) remedies these short comings at the expense of simulation time. Both of these models (collectively called the ***proposed models*** henceforth) are behavioral models. In order to determine the accuracy of these models, they are compared to a traditional behavioral model (called simply the ***behavioral model*** henceforth).

As with any model, a number of approximations are made. With these approximations comes a divergence from reality in terms of the models output. In order for a model to be useful, the effects of the models approximation must be understood. Based on this understanding a set of recommended operating parameters can be proposed. One of the goals of this thesis is to provide such recommendations.

The mechanism which contributes to the inaccurate regions of prediction in the ***standard proposed model*** can be taken advantage of to rid a Frac-N PLL of reference spurs. This is discussed in Sections 5.4.1 and 5.5.1.

Another goal is to provide a deeper understanding of the complex nature of a PLL. Traditionally it is thought that reference spurs come from leakage current from the Phase Frequency Detector and/or a mismatch in the charge pump current [3]. This thesis shows evidence that a major contributor to reference spurs is simply the shape of the charge pump pulse and will occur to a lesser degree even in an ideal system.

This thesis also offers some insight into the effect of the length of the sequence that is output from a sigma-delta modulator. An equation is presented that predicts the location of all Frac-N reference spurs (see Section 2.3.2.2). Equations are also presented that predict the sequence length of certain 2^{nd} order sigma-delta modulators (see Chapter 4).

One other contribution of this thesis deals with the voltage controlled oscillator (VCO) in a PLL. The VCO is in essence an FM modulator. An understanding of this is needed in order to anticipate how spurs at the VCO input will appear at the VCO output. This is discussed in Section 2.4.2.

1.2 Thesis Outline

Chapter 2 provides the technical background that forms a basis for this thesis. It discusses what a PLL is and how one operates. It then discusses each PLL component in detail. Finally it discusses noise and some non-linearities that exist within a PLL. Chapter 3 discusses the state-of-the-art in terms of PLL models used for phase noise and spur prediction. Chapter 4 presents empirical equations for determining the sequence length of 2^{nd} order sigma-delta modulators. Chapter 5 develops Simulink-based behavioral models used for testing. Chapter 6 discusses the test procedure and the test results for the proposed models. Chapter 7 draws conclusions from this thesis, sums up future work, and summarizes its contributions.

2 Technical Background

This chapter provides the necessary background information on which this thesis was written. Section 2.1 reviews general phase-locked loop (PLL) design. Section 2.2 defines and examines the difference between an Integer-N and a Frac-N PLL. Section 2.3 talks about the different sources of noise within a Frac-N PLL. Section 2.4 discusses some non-linearities inherent in a Frac-N PLL.

2.1 What is a PLL?

A PLL is a negative feedback control system designed to achieve an output at a particular frequency. A PLL can be used in applications such as clock recovery and frequency synthesis. Clock recovery is the process of extracting clock information from a given stream of data. Frequency synthesis is employed when particular clock frequencies need to be generated. The required frequencies are generally a multiple of some reliable reference frequency [2]. This thesis will focus on phase-locking with regards to frequency synthesis.

In radio communications a signal is often passed through a transmission medium (such as air) at a specific frequency. Figure 1 offers an example of how a frequency synthesizer works in the context of a receiver chain in superheterodyne radio design. In short, the purpose of the first frequency synthesizer is to down convert the radio frequency (RF) signal to an intermediate

frequency (IF). The second frequency synthesizer then converts this IF signal down to baseband.

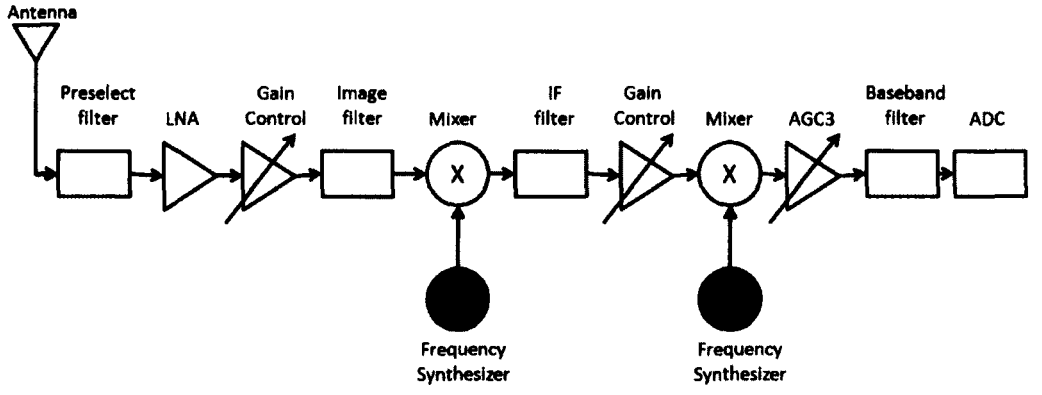


Figure 1 - Receiver for a Superheterodyne Radio Architecture.

To better understand the frequency synthesizer's role a brief overview of each block in the system is presented. A signal received by the antenna first gets passed through a preselect filter which is responsible for only passing the receive band (the band of frequencies the radio wishes to receive) to the radio. The preselect filter works in conjunction with the image filter to help isolate the receive band. Both of these filters must pass the entire receive band, since channel selection does not occur until the first mixer. The low-noise amplifier (LNA) is used to boost the incoming signal. The important feature of the LNA is to provide this gain while degrading the signal-to-noise ratio (SNR) as little as possible. Intuitively, we can see that noise early on in the chain is more damaging to the radio's SNR since noise will see gain from each of the blocks that follow it. The Automatic Gain Control (AGC) blocks help ensure the radio can deal with the full range of expected input power levels. For example, a low

powered incoming signal requires a greater 'boost' than a higher powered incoming signal. The job of the first mixer is to convert the RF channels down to lower frequencies and center the channel we are interested in at a specific IF. Here is where the frequency synthesizer becomes important. The IF at which the mixer moves the signal to is determined by the frequency the synthesizer outputs. At this point the entire receive band is now mixed down to the IF. The IF filter is responsible for isolating only the channel of interest from the receive band. The second mixer (with help from the second frequency synthesizer) then takes this signal and converts it down to baseband. After a final filtering from the baseband filter the analog signal is sampled by an analog-to-digital converter (ADC) and passed to the digital signal processor (DSP) backend [4].

2.1.1 Integer-N vs. Fractional-N PLLs

The output frequency of a PLL is a multiple of a predetermined reference frequency. If this multiple is an integer the PLL is considered an Integer-N PLL. If the multiple contains a fractional component the PLL is termed a Frac-N PLL (or Fractional-N PLL). Both types will now be discussed in more detail.

2.1.2 Integer-N PLL

An Integer-N PLL generates an output frequency at an integer multiple of its input frequency. A block diagram of an Integer-N PLL is presented in Figure 2.

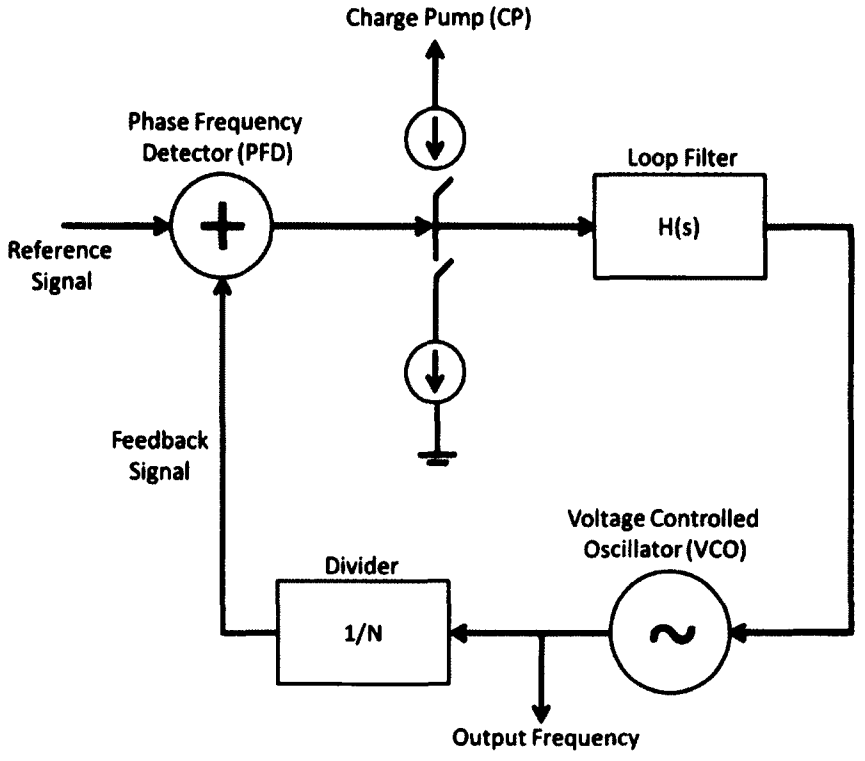


Figure 2 - Block Diagram of Integer-N PLL.

The input into the PLL is a ***Reference Signal***, generally provided from a crystal oscillator. A ***Phase Frequency Detector (PFD)*** compares the rising edges of the ***Reference Signal*** and ***Feedback Signal*** to determine if the feedback signal is leading or lagging with respect to the reference. Based on this comparison the PFD outputs a signal to the ***Charge Pump***. The charge pump then outputs a current which is related to the phase difference of the reference and feedback signals. This charge pump current is then seen by the ***Loop Filter*** which has two main responsibilities. Firstly it converts the charge pump current into a voltage used to drive the ***Voltage Controlled Oscillator (VCO)***. Secondly the filter controls the loop dynamics of the PLL (bandwidth, settling time, etc.). The VCO outputs a frequency that is related to the input voltage of the VCO. Note that the

output of the VCO is also the output of the entire PLL (hence VCO output and PLL output are used interchangeably throughout this thesis). The output of the PLL is then fed into a divider block which divides the frequency back down to the reference frequency. The feedback signal from the divider is then fed back into the PFD, completing the loop. This feedback mechanism allows the output frequency of the PLL to lock on to a frequency that is an integer multiple of the reference [2].

One major problem with an Integer-N PLL is that the output frequency is limited to be an integer multiple of the reference frequency. Therefore in order to have a small resolution in the output frequency a small reference frequency is required. Small reference frequencies bring with it some problems such as decreased bandwidth, longer settling times, and increased noise. A Frac-N PLL can be used to circumvent the shortcomings of an Integer-N PLL [5].

2.1.3 Fractional-N PLL (Frac-N PLL)

Figure 3 shows a Frac-N PLL block diagram. The only way in which the diagram differs from an Integer-N PLL is the additional feedback loop which includes the ***sigma-delta modulator (SDM)***. The SDM outputs a signal that tells the divider which integer value to divide by.

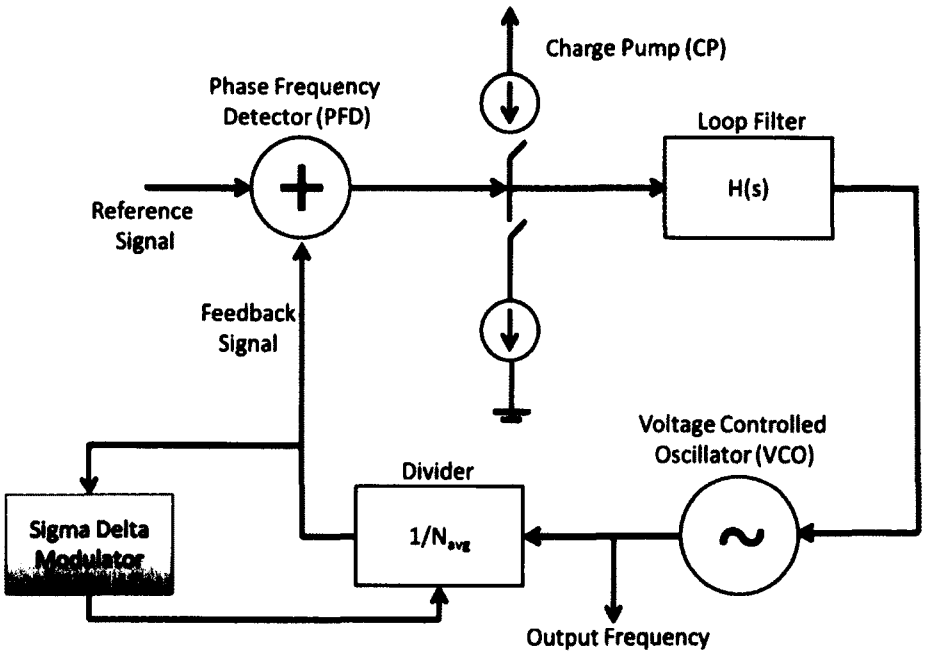


Figure 3 - Block Diagram of a Frac-N PLL.

A simple example will demonstrate the mechanics of a Frac-N PLL. Assume a PLL has a reference signal frequency of 40MHz. We wish our PLL output to have a frequency of 2.41GHz. This requires a divide ratio of 60 1/4. One way a PLL could achieve this divide ratio by dividing by 60 for three reference cycles, then 61 for one cycle. This pattern then repeats. Over each repetition we see that the average divide value, N_{avg}, is 60 1/4 as expected. It is the responsibility of the SDM to tell the divider to divide by 60 or 61. One might ask why can't the PLL divider simply divide by 60.25 and be done with this dithering between two integer divide ratios. The reason is the hardware implementation of the divider block allows only for integer division [2]. This means the output frequency of a Frac-N PLL must be an averaged result of many integer divide values (two in our example, but more divide values are possible).

One can imagine how any frequency could be synthesized as now the output resolution is dependent on the divide ratio and not the reference clock. This is a key concept in Frac-N PLLs. To continue our example, now assume the output frequency required is 2.405GHz. This can be achieved by changing the divide ratio from 60 1/4 to 60 1/8.

The divider value N_{avg} in Eq 2-1 is made up of an integer portion ***I*** and a fractional portion ***X/M***. ***N*** is defined as the instantaneous divide value.

$$N_{avg} = I\frac{X}{M} \qquad \text{Eq 2-1}$$

In switching between different integers the PLL is rarely *instantaneously* providing the proper output frequency. One important point to consider is that the dithering between integer values comes at a cost in terms of spurious tones (called spurs) in the frequency domain. This will be described in detail in Section 2.3.2.2.

One major benefit of a Frac-N PLL is the high resolution in output frequency. As discussed earlier, the output resolution of an integer-N PLL is limited to an integer multiple of the reference frequency. No such limit exists in a frac-N PLL.

2.2 PLL Components

Each block of a PLL will now be examined in detail. Refer to Figure 3 for the overall diagram.

2.2.1 Phase Frequency Detector and Charge Pump

The Phase Frequency Detector and Charge Pump work in concert to produce pulses of current which are proportional to the phase difference between the Reference Signal and Feedback Signal.

A block diagram of the PFD and charge pump (collectively called a PFD/CP) is presented in Figure 4 (Many PFD architectures exist. Due to its ubiquity, the tristate detector will be used exclusively in this thesis).

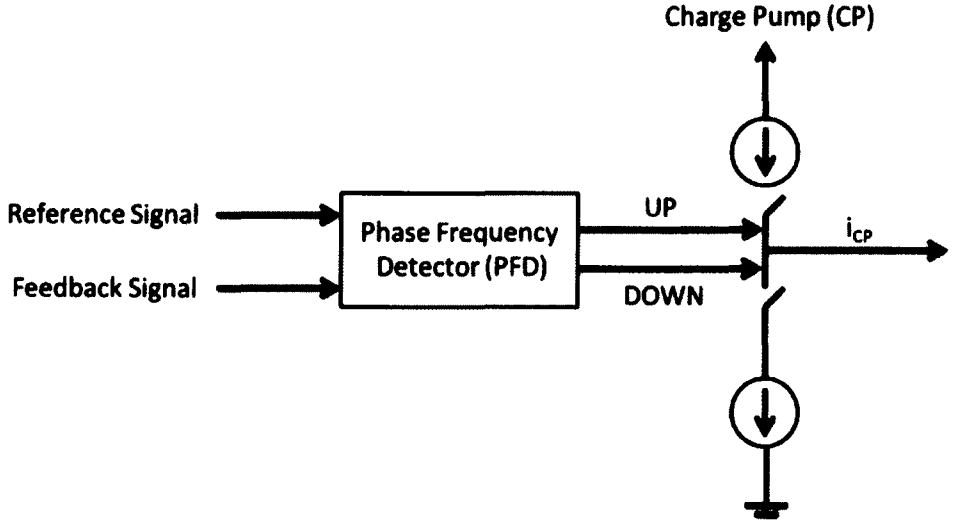


Figure 4 - Block Diagram of PFD/CP.

The pulses of current, i_{CP}, alter the tuning voltage of the VCO. In this manner, the phase of the feedback signal is either increased or decreased in order to more closely match the reference phase.

To better understand the operation of the PFD/CP we examine the timing diagram presented in Figure 5. At time A the feedback signal goes high. This indicates the feedback signal is leading the reference signal. From this we can intuit that the VCO output frequency needs to be decreased, effectively 'slowing down' the feedback signal. To accomplish this, the PFD issues a DOWN pulse until time B (at which the reference signal has shown its rising edge). At this point the PFD will output no signal. It will wait for the next rising edge of either the reference of feedback signals. This occurs at time C in this example. Since both rising edges occur simultaneously there is no UP or DOWN signal. We see at time D the reference now leads, so we get an UP pulse between times D and E to increase the VCO output frequency.

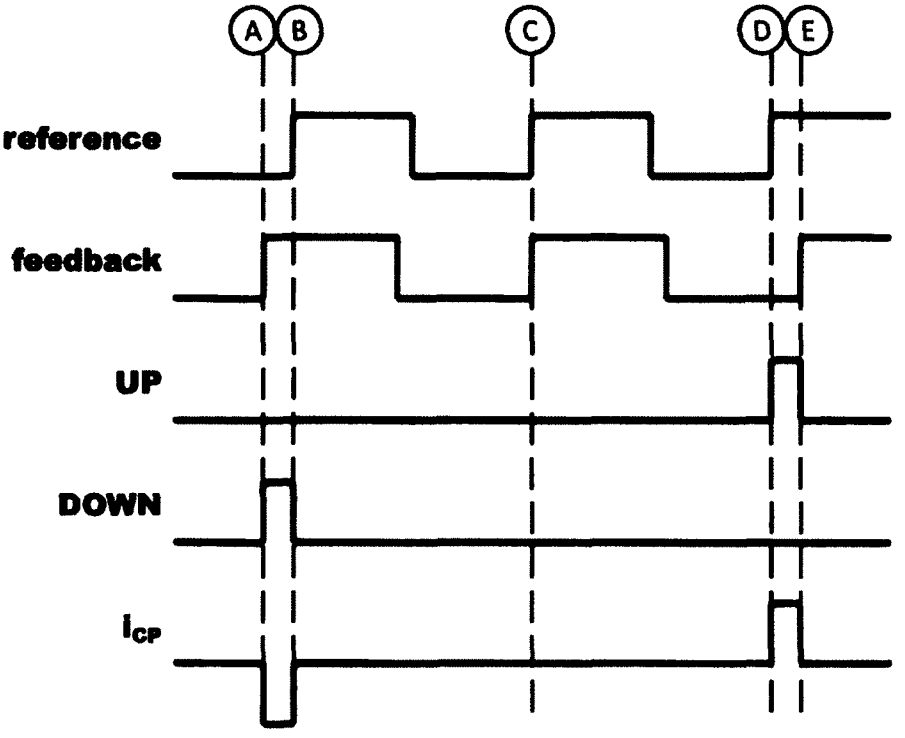


Figure 5 - Timing Diagram for PFD/CP.

From this example we can now understand the three states in the 'tristate' PFD as shown in Figure 6. Referring back to Figure 5 allows us to visualize the UP and DOWN signal and how they relate to the state of the PFD. Assume before time A we are in the Zero State. The rising feedback edge at time A pushes the PFD into the DOWN State. At time B the rising reference edge moves us back to the Zero State. The rest of Figure 5 can be worked through in this manner.

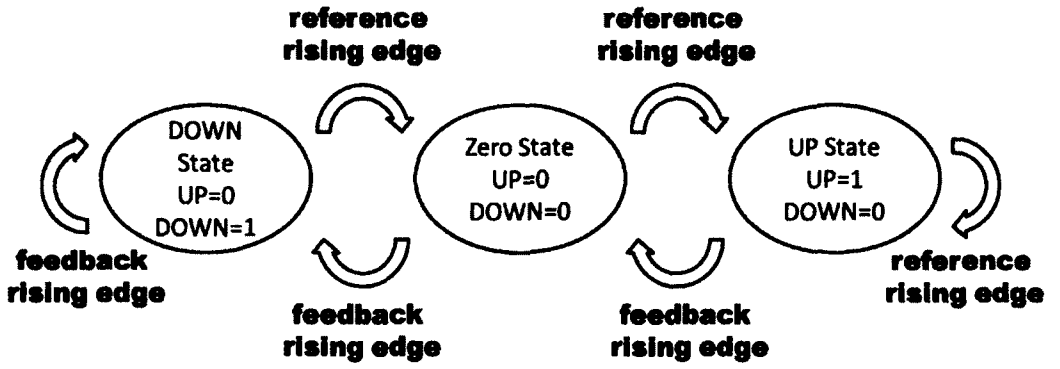


Figure 6 - State Diagram of PFD.

We have established that the UP and DOWN pulses seen in Figure 5 result from the current state of the PFD. The role of the charge pump is to take these signals and amalgamate them into a current pulse train called i_{CP}. We stated earlier that i_{CP} is related to the phase error of the reference and feedback signals. Using this notion we can begin to derive a formula for i_{CP}.

Both the reference and feedback signals have an instantaneous phase associated with them. We will call these θ_{ref} and θ_{fb} respectively. We can now express the phase difference between θ_{ref} and θ_{fb} as a percentage of a full revolution in phase, 2π.

$$\frac{\theta_{ref} - \theta_{fb}}{2\pi} \qquad \text{Eq 2-2}$$

We now define ***I*** as the current that the charge pump can output. Referring back to the PFD state diagram we note that the PFD outputs positive ***I*** current in the UP state, negative ***I*** current in the DOWN state, and no current in zero state. By multiplying Eq 2-2 with ***I*** we now have a current that is

proportional to the phase difference between θ_{ref} and θ_{fb}. Constant k_{phase} is defined as a coefficient by which the phase difference is multiplied by to give i_{CP}. From Eq 2-3 we obtain k_{phase} for a tristate phase detector [2].

$$i_{CP} = \frac{I}{2\pi}(\theta_{ref} - \theta_{fb}) \quad \text{Eq 2-3}$$

$$k_{phase} = \frac{I}{2\pi} \quad \text{Eq 2-4}$$

2.2.2 Voltage Controlled Oscillator (VCO)

A VCO is simply an oscillator whose output frequency is related to its input voltage. Figure 7 shows the characteristic of a linear VCO. We see as the tuning voltage, v_{Tune}, increases so does the frequency. A VCO has a range of voltages over which it can operate. This range is shown as v_{Min} to v_{Max}. v_{Nom} is the nominal voltage at which the VCO operates. k_{vco} is the slope of the line given in Hz/V. It may also be expressed in radians/Sec/V. With no input voltage to the VCO (that is v_{Tune} = 0), the output is the nominal frequency, f_{nom}. The control voltage v_{Tune} moves the output frequency away from the nominal frequency. The instantaneous frequency of the VCO is given in Eq 2-5.

$$f_{vco} = f_{nom} + k_{vco}v_{tune} \quad \text{Eq 2-5}$$

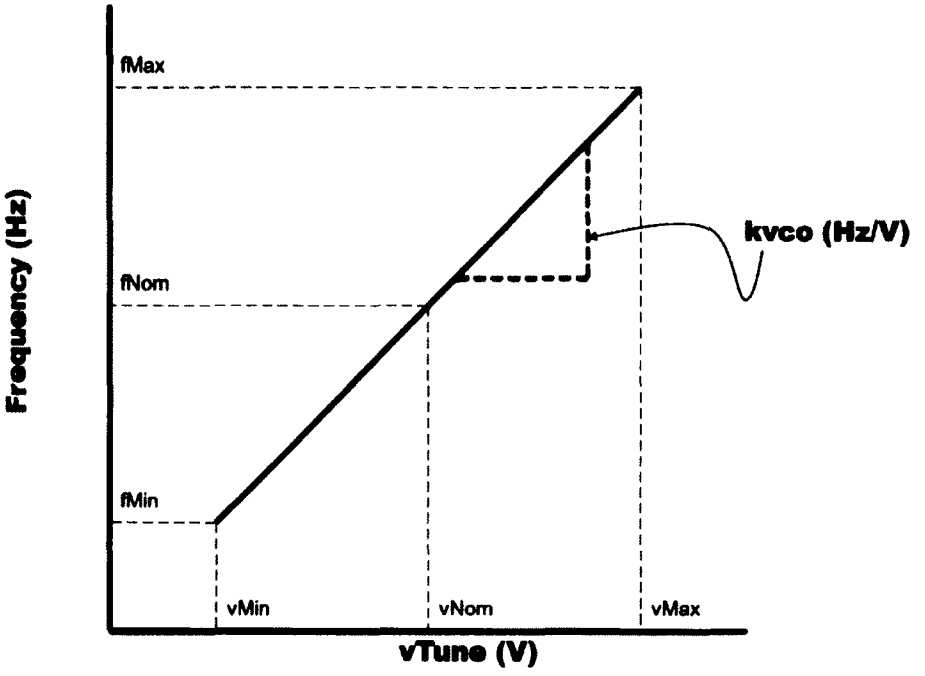


Figure 7 - VCO characteristic.

In the Laplace domain the VCO is generally expressed as k_{vco}/s [2].

2.2.3 Loop Filter

As discussed earlier the loop filter has two roles. Firstly it converts the charge pump current into a voltage used to drive the VCO. Secondly it controls the loop dynamics of the PLL (bandwidth, settling time, etc.). Figure 8 shows typical 2^{nd} and 3^{rd} order passive filters. This thesis uses 2^{nd} order filters exclusively.

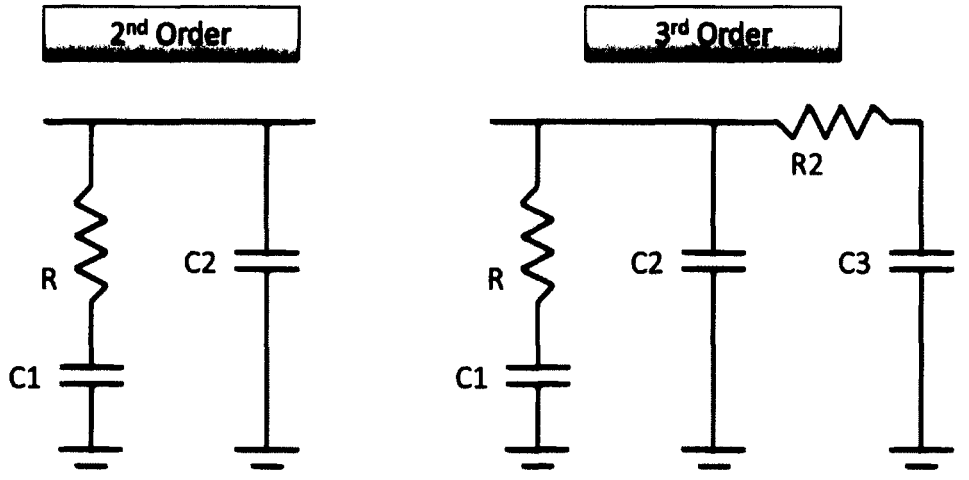


Figure 8 - Schematics of 2nd and 3rd order passive loop filters.

Equations for the loop transfer functions can be found in [6].

2.2.4 Sigma-delta Modulator

Section 2.1.3 explains how an SDM is incorporated into a Frac-N PLL. This section will focus on different SDM orders and what effect they have.

Recall the role of an SDM is to output a signal that represents the fractional portion of the divide value. This is expressed as X/M, where both X and M are integers. What does X/M really mean in terms of SDM operation? Figure 9 provides outputs of 1^{st}, 2^{nd}, and 3^{rd} order SDMs for an example divide fraction of 1/4. This continues the example described in Section 2.1.3 where the goal is to have an average divide ratio of 60 1/4. The 1^{st} order SDM output is straight forward; it is high for one clock cycle and low for the next three. This pattern then repeats. The length of this pattern, in terms of the number of clock cycles, is termed the ***SDM sequence length***. In this case the sequence length is 4. The

2nd and 3rd order SDMs have more complicated patterns, however they still average out to 1/4 over their sequence length. Regardless of the order, these examples would still result in an average divide ratio of 60 1/4. Higher order sigma-delta modulators will lessen spurs (Section 2.3.2) at low frequencies and move the energy to higher frequencies.

Note that the sequence length may or may not increase as the order of the SDM increases. Also note that for a given order, many SDM architectures exist. The output for each of these will vary. The output signals provided are simply one example.

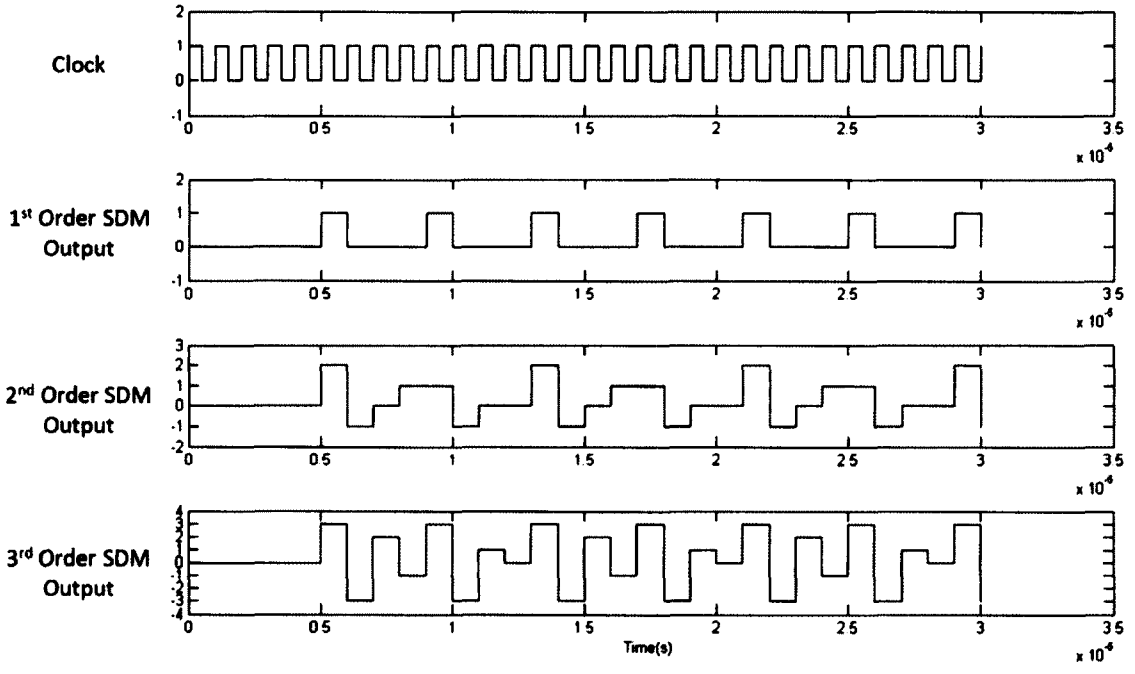


Figure 9 - Example SDM Output Signals for 1/4 divider fraction.

A major issue in Frac-N PLLs stems from the periodic nature of the output signal in Figure 9 which will generate Frac-N spurs in the frequency domain. Frac-N spurs are explained in greater detail in Section 2.3.2.2.

Many SDM architectures exist. The output of an SDM may be classified as single-bit or multi-bit. Single-bit architectures will output only a logic 0 or 1. This implies the divider can only divide between two values. The 1st Order SDM in Figure 9 is an example of a single-bit architecture. A multi-bit output can output many values. The ***maximum*** number of values is related to the number of bits by Eq 2-6 [1]. ***n*** is the order of the SDM.

$$maximum\ number\ of\ values = 2^n \qquad \text{Eq 2-6}$$

The 2nd and 3rd order SDMs in Figure 9 show examples of multi-bit outputs. The 3rd order output ranges from -3 to 3, or seven possible values. Three bits are needed to store these values, hence the term multi-bit output. 1st order SDMs will always have single-bit outputs. Higher order SDMs may have either single-bit or multi-bit outputs depending on the SDM architecture. This thesis uses a MASH SDM architecture that has a multi-bit output.

The internal workings of an SDM are not important for the purposes of this thesis. For a more detailed explanation of MASH structures and SDMs in general see [2].

For a detailed discussion of sigma-delta modulators on their use in a PLL see [12].

2.2.5 Divider

The divider simply takes the output of the VCO and divides the frequency down by whatever integer it is programmed with. As previously discussed this integer divide value will change (via the SDM) if we are dealing with a Frac-N PLL.

Figure 10 - Divider Operation.

2.3 Noise in PLLs

Noise in a PLL can be looked at from different perspectives depending on the purpose of the PLL. For example a PLL may be used to extract information from a noisy signal [5]. As this thesis is concerned with the frequency synthesizer variation of a PLL we are not concerned with 'input signal noise' so to speak. Instead we are concerned with noise that is generated within the PLL itself. We separate noise into two distinct categories; phase noise (or jitter) and spurs.

2.3.1 Phase Noise and Jitter

The terms 'phase noise' and 'jitter' are often used interchangeably. Strictly speaking phase noise refers to a random fluctuation in the desired output frequency as seen in the frequency domain. Jitter refers to the same phenomenon as seen in the time domain, shown in Figure 11. It is intuitive to think of jitter as a 'wobbling' of the desired output signal (depicted as the thick black line).

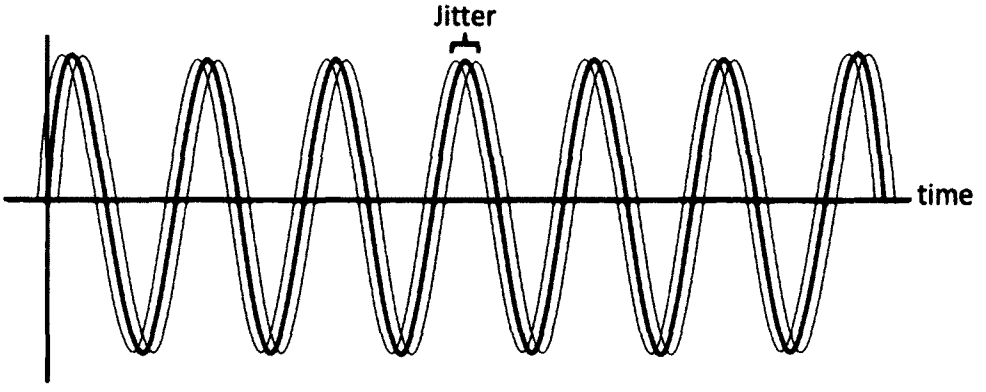


Figure 11 - Jitter in the time domain.

There are many sources of phase noise. In fact each component within the PLL may contribute to phase noise. The crystal used for the reference clock can contribute to close-in phase noise (that is phase noise that resides near the desired frequency). The phase detector contributes to phase noise in terms of flicker noise and thermal noise. The charge pump may contribute to phase noise when it is pumping current. This implies that having a charge pump off as much as possible is desirable. The loop filter contributes in the form of thermal noise

due to random electron motion of any resistors present. The VCO can contribute phase noise and has most impact outside the loop bandwidth [2].

From [2], a generally accepted linear model used for phase noise analysis is presented in Figure 12. For this analysis all the blocks are linearly modeled in the s-domain based on Section 2.2.

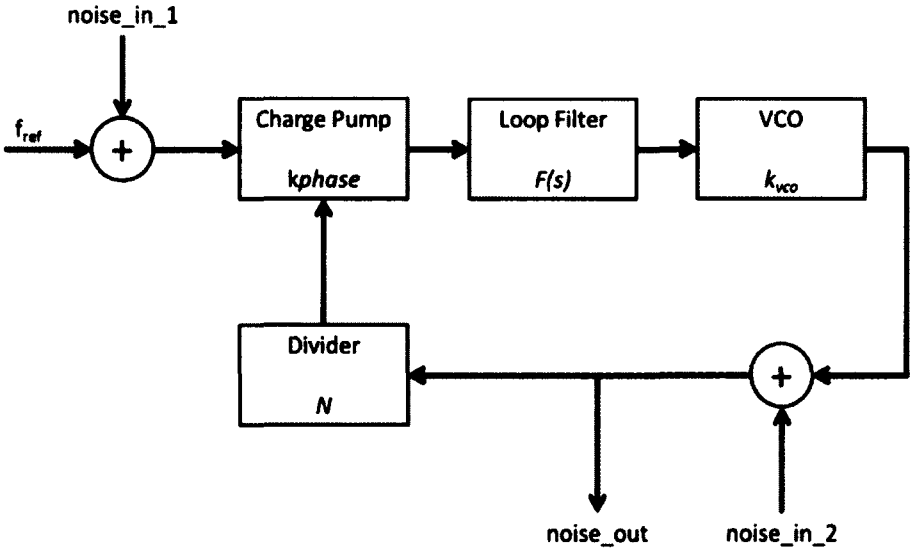


Figure 12 - Phase Noise Analysis of an Integer-N PLL.

From this figure we can derive equations for both input noise sources in terms of the output noise.

$$\frac{noise\_out}{noise\_in\_1} = \frac{F(s)k_{vco}k_{phase}}{s + \frac{F(s)k_{vco}k_{phase}}{N}} \qquad \text{Eq 2-7}$$

$$\frac{noise\_out}{noise\_in\_2} = \frac{s}{s + \frac{F(s)k_{vco}k_{phase}}{N}}$$ Eq 2-8

Note that Eq 2-7 is a low-pass filter that is input referred, while Eq 2-8 is a high-pass filter that is referred to the output. The VCO noise is referred to the output. The remaining blocks are referred to the input. [2] contains the phase noise equations for each block in the PLL. These equations can be put into Eq 2-7 and Eq 2-8 to come up with a plot similar to the one seen in Figure 13. There are a few things to note here. The loop filter noise is generally low and can be ignored. Additionally the charge pump dominates within the loop bandwidth (recall the low-pass filter characteristics of Eq 2-7).

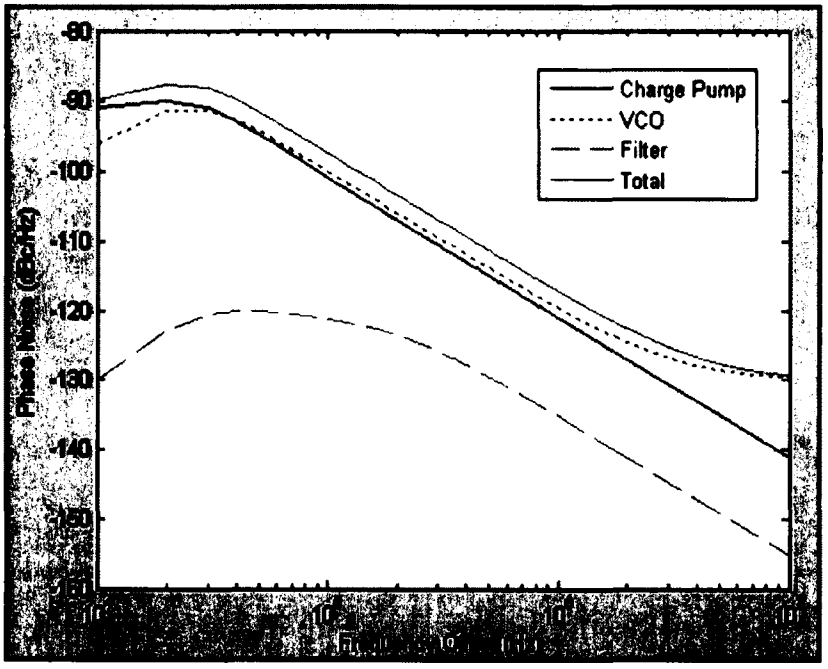


Figure 13 - Example Phase Noise plot of PLL Components.

In terms of phase noise analysis this model is appropriate. The shortcoming of this model is the inability to predict spurs that result from SDM switching. The reasons for this are discussed in Section 2.4.

As this thesis is concerned strictly with spurs we ignore all sources of phase noise.

2.3.2 Spurs

Phase noise occurs at all frequencies. In contrast, a spur is an unwanted tone in the frequency domain at a *particular frequency*. They can be large and appear as distinct spikes that rise above the phase noise. This is shown in Figure 14.

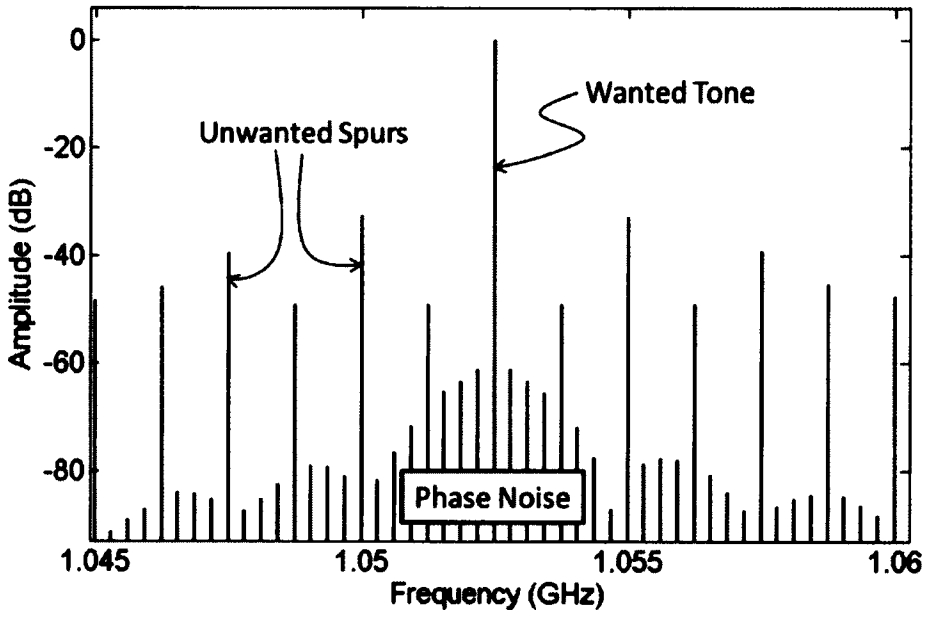


Figure 14 - This figure shows an example of how spurs show up around a desired tone.

Spurs result from two common sources; reference spurs and Frac-n spurs. Both will now be explained in detail.

2.3.2.1 Reference Spurs

Reference spurs appear in the frequency domain at multiples of the reference frequency. In Section 5.2.2, how reference spurs can be greatly reduced will be discussed. However, for the purposes of background material this section will describe how a phenomenon called 'charge pump mismatch' generates reference spurs. To get a better idea of how this occurs we must first understand the output pulses from the PFD. Ideally, the PFD is in one of three states. Using Figure 6 as a reference these states correspond to an UP pulse, a DOWN pulse, or no pulse (hence the name tri-state PFD). In reality there exists another state in which both the UP and DOWN signals are on for a brief time. Using an ideal charge pump this wouldn't be problematic as the currents I_{UP} and I_{DOWN} in Figure 15 would be equal.

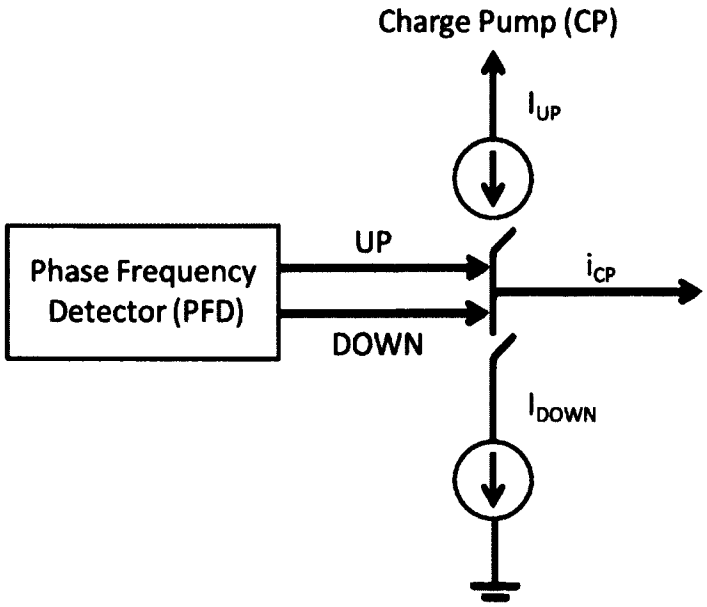


Figure 15 - PFD and CP.

If the two currents are *not* matched (which is likely) there will be an unwanted current introduced into the loop. This mismatched current must be compensated for on the next reference cycle. The fact that this phenomenon is periodic with respect to a reference cycle explains the spurs showing up at multiples of the reference frequency [2]. This thesis will show that even with a perfectly ideal system reference spurs will still be generated.

Reference spur magnitude relative to the carrier can be calculated by the Eq 2-9 [2] where δ is the reset path delay and ΔI is the current source mismatch amount.

$$|Spur| = 20log\left[\frac{\delta^2 \cdot \Delta I \cdot k_{vco}}{4\pi \cdot C_2}\left(1 + \frac{\Delta I}{I_{CP}}\right)\right] dBc \qquad \text{Eq 2-9}$$

2.3.2.2 Frac-N Spurs

It is understood that a signal which is periodic in the time domain will have frequency components related to its periodicity. Sections 2.1.3 and 2.2.4 discuss how a Frac-N PLL switches between different integer divide values. This switching is performed in a periodic fashion. It is this periodicity in which spurs are generated. The creation of Frac-N spurs may be best demonstrated graphically. Figure 16 shows an example time domain signal and how it may look in the frequency domain. The time domain signal represents the SDM output in a Frac-N PLL. The spurs at the output of the SDM then get processed by the PLL transfer function and eventually end up at the output of the PLL. This is problematic.

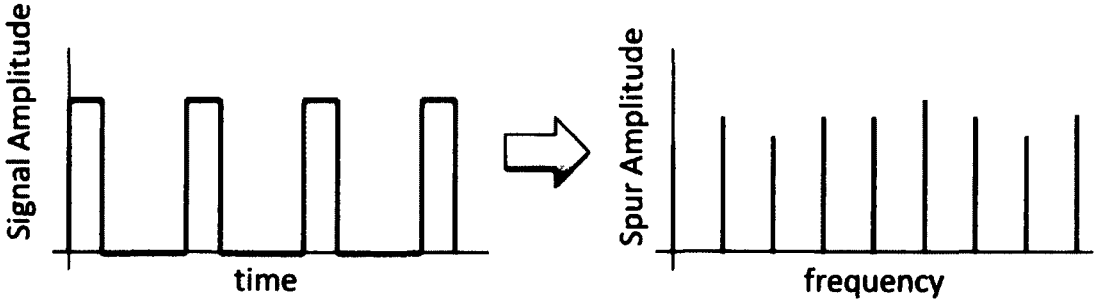


Figure 16 - Time domain to Frequency Domain Example.

Increasing the order of the SDM can be used to lessen the impact of the spurs. The SDM does this by 'randomizing' the output as shown in Figure 9. This randomization attenuates low-frequency spurs (which can show up in-band) by moving the energy to higher frequencies where they get attenuated by the loop

filter. This process is called ***noise-shaping*** and is shown in Figure 17. A higher order SDM will result in more noise shaping [2].

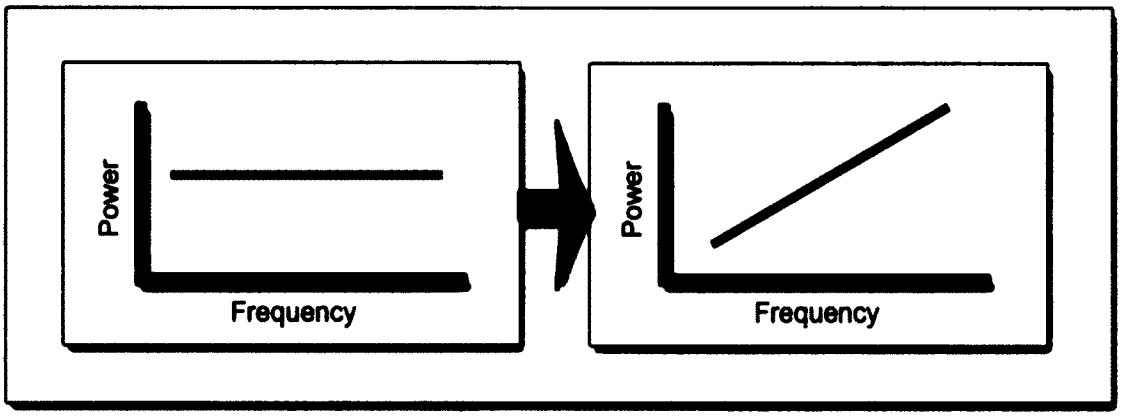


Figure 17 - Noise Shaping.

The spacing of the spurs is dependent on the reference frequency as well as the sequence length L_s of the SDM, given by Eq 2-10. This equation is a result of empirical evidence.

$$spur\ spacing = \frac{f_{ref}}{L_s} \qquad \text{Eq 2-10}$$

2.4 PLL Non-linearity's

This thesis proposes that a PLL consisting of only linearized components cannot properly predict spur levels. This section discusses why this is so.

2.4.1 *Charge Pump Output Current Non-linearity*

The output of the PFD/CP is traditionally linearly modeled as a current which depends on the instantaneous phase error and k_{phase}, shown in Eq 2-11. This current is updated once every reference cycle.

$$i_{CP} = k_{phase}(\varphi_{ref} - \varphi_{fb}) \qquad \text{Eq 2-11}$$

The problem with this linear model is the output current now has lost any information regarding the pulsed nature of the charge pump. Instead the output is presented as an average. This is shown in Figure 18 where each ***Tristate Charge Pump*** Pulse is averaged over a reference cycle in the ***Linear Charge Pump Model***. Intuition tells us that these two signals will produce noticeably different frequency content.

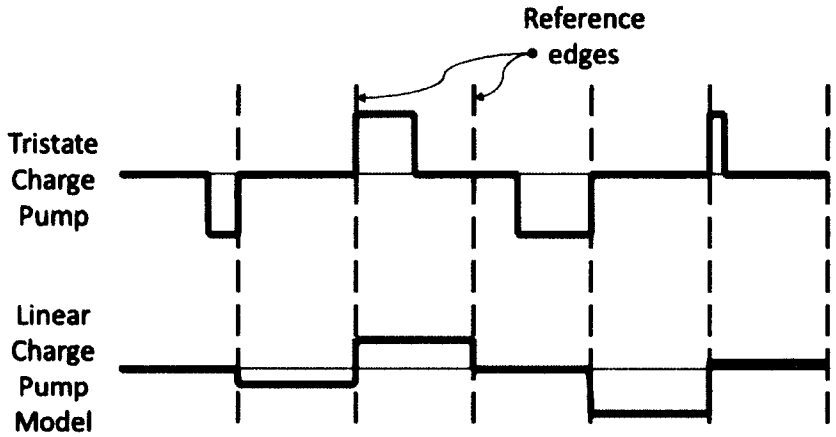


Figure 18 - Example of Charge Pump Pulse Averaging.

To understand the non-linearity that results from the charge pump pulse we examine the nature of the pulse itself, shown in Figure 19.

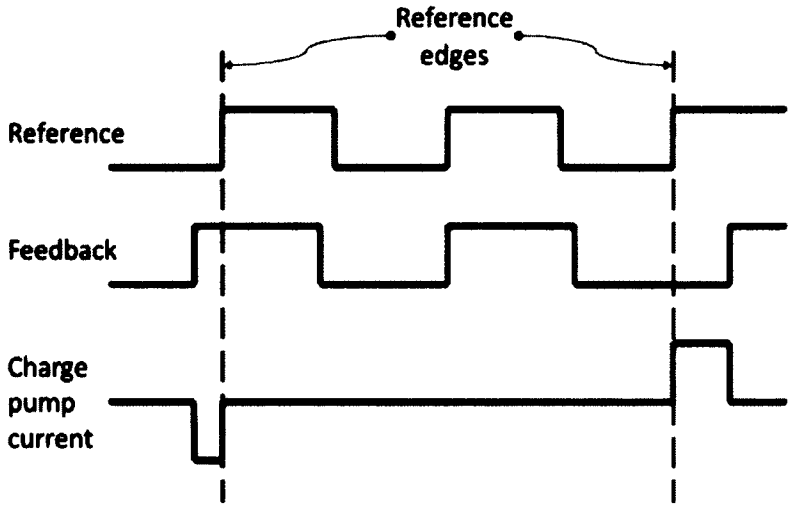


Figure 19 - Phase detector/charge pump timing diagram.

The charge pump current pulse is affected in two ways by the reference and feedback signals. Firstly the width of the charge pump pulse is dependent upon the spacing between the rising edges of the two signals. Secondly the pulse may occur either before or after the reference rising edge depending on whether the reference signal leads or lags the feedback signal. It is these two factors that cause the charge pump to behave non-linearly.

See Section 5.4.1 for an enhanced discussion on this subject.

2.4.2 VCO FM Modulation Non-linearity

The proposed and behavioral models in this thesis both utilize a linear VCO. By linear it is meant that the output frequency is linearly related to the input frequency as described in Section 2.2.2.

This section will show that even a linear VCO can exhibit non-linear behavior due to FM modulation.

Understanding how spurs move through the VCO is important for two reasons. Firstly, if a non-linearity exists within a VCO we must understand it in order to properly model it. Secondly, the proposed model determines spur location and amplitude at the input of the VCO. However, designers are interested in the spur information at the output of the VCO as these are the artifacts that will get mixed into the receive and transmit paths.

We now look at a VCO from the perspective of frequency modulation (FM). Since the output frequency changes linearly with the input voltage we are essentially frequency modulating the 'carrier' signal of the VCO. The frequency of the 'carrier' signal is related to the average VCO input voltage v_{Tune}.

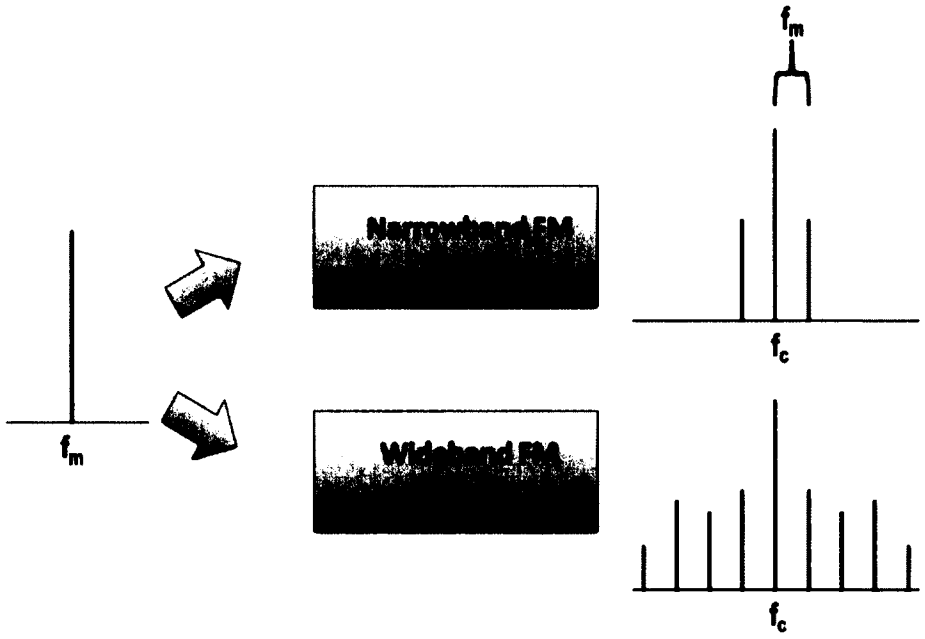


Figure 20 - Narrowband vs. Wideband FM.

FM is categorized into narrowband FM and wideband FM as shown in Figure 20. Assuming a single sinusoid as input, narrowband FM is generally confined to a single set of sidebands, whereas wideband FM has multiple sidebands. The modulation index, β, is used to categorize the type of modulation and is defined as [1]:

$$\beta = \frac{k_{vco} A_m}{f_m} \qquad \text{Eq 2-12}$$

A_m is the amplitude of the modulating signal. f_m is the frequency of the modulating signal.

If β is less than 0.1 a narrowband approximation may be used. If β is greater than 0.1 full wideband analysis must be used [1]. The value of β is a rule of thumb given in [1], and may change depending on the source literature.

Figure 21 shows an example VCO with a sinusoidal input which results in wideband FM. If we examine the frequency components (using a DFT) of the input and the output we quickly see the potential non-linear nature of a VCO in terms of spurs. If this non-linearity does exist, it is extremely difficult to mathematically analyze how spurs transfer through a VCO.

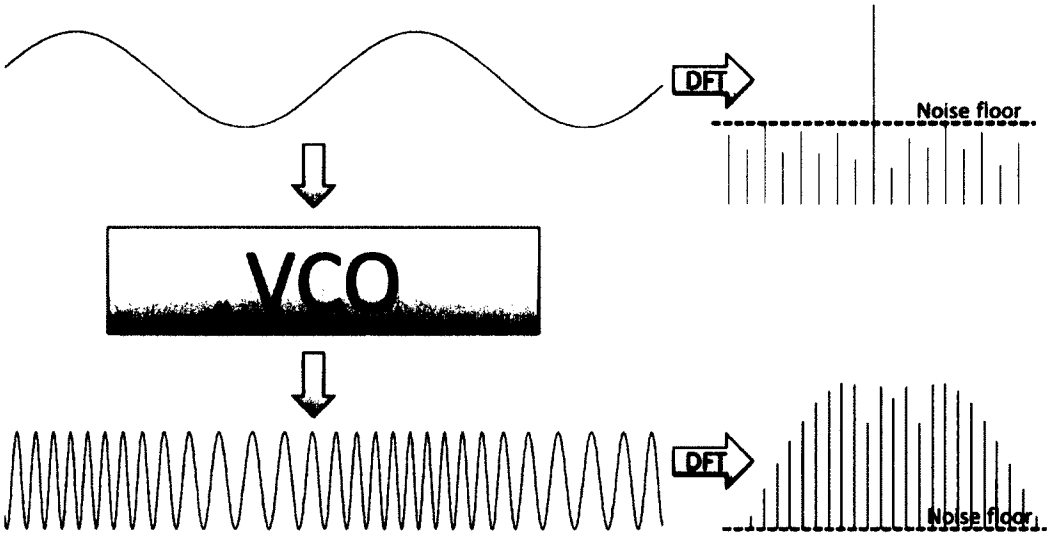


Figure 21 - Wideband Frequency Modulation using a VCO.

For both narrowband and wideband, classical FM theory assumes a single sinusoidal modulating frequency. In reality the VCO input in a PLL would never be sinusoidal. The signal will be much more complex, made up of infinite sinusoids in theory. In terms of FM it is important to realize that each of the

frequency components that comprise a generic VCO input signal will have a β associated with it. In order to use the narrowband approximation for a generic signal β_{tot}, the sum of all of the frequency components β's, must total less than 0.1 (see Section 6.3.3 for empirical evidence of this). If this is the case we can determine an equation to predict the amplitude of each spur. The analysis begins with the narrowband FM time-domain equation given in [1]. s(t) is the FM wave.

$$s(t) \approx A_c \cos(2\pi f_c t) - \beta A_c \sin(2\pi f_c t) \sin(2\pi f_m t) \quad \text{Eq 2-13}$$

Taking the Fourier transform yields the following.

$$\begin{aligned} S(f) \approx & \frac{A_c}{2}[\delta(f - f_c) + \delta(f + f_c)] \\ & - \frac{\beta A_c}{4}[\delta(f - f_c + f_m) + \delta(f + f_c - f_m)] \\ & + \frac{\beta A_c}{4}[\delta(f - f_c - f_m) + \delta(f + f_c + f_m)] \end{aligned} \quad \text{Eq 2-14}$$

From Eq 2-14 it can be shown that the n^{th} spur will have the following amplitude:

$$n^{th} spur\ amplitude = \frac{\beta_n A_c}{2} \quad \text{Eq 2-15}$$

where A_c is the amplitude of the carrier (basically the amplitude of the VCO output) and n is the index of the spur.

Knowing that each frequency component has a unique β, Eq 2-12 is examined in more detail. Firstly, all β values will have a constant k_{vco}. Secondly A_m tends to decrease at higher frequencies as a result of the loop filter. Finally f_m will increase since it is a harmonic of the fundamental frequency of v_{Tune}. With these three points in mind we see how β will generally decrease as the frequency of the spur in question increases. This means the narrowband approximation becomes more appropriate at higher harmonics. Conversely this implies that the most 'wideband' components in the frequency spectrum of v_{Tune} occur closest to DC.

We now test the math to see if we can predict spur levels at the output of the VCO given an input signal. Figure 22 shows a comparison between spurs generated via Eq 2-15 versus spurs generated from a VCO. The figure shows that the math is accurate within a few dB for all spurs except the spur at the first reference frequency harmonic. The math predicts a complete eradication of the reference spur. However the reference spur generated by the VCO is larger than predicted. It is still much smaller than the spurs that surround it, but it has clearly not been completely eradicated. This may indicate that wideband FM is occurring for this example as neighboring spurs mix onto the reference frequency harmonics.

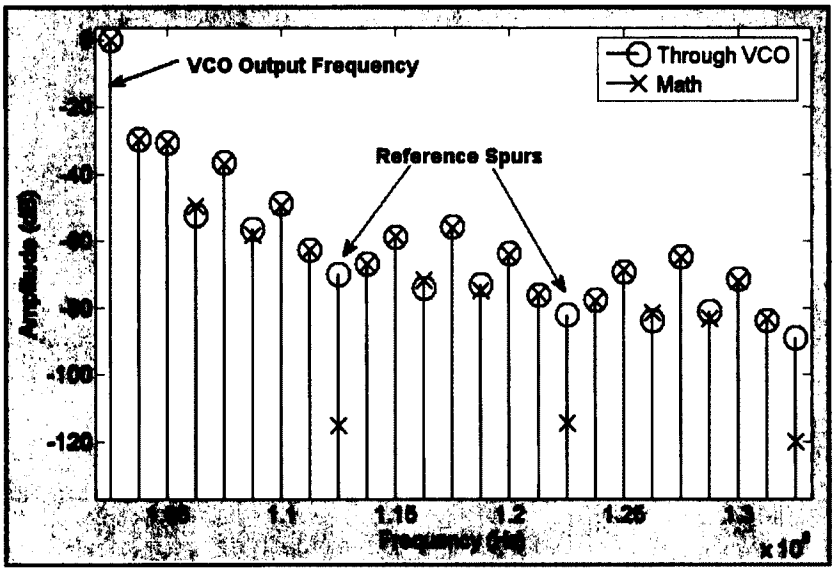


Figure 22 - This figure shows spurs at the VCO output. One set of spurs is generated using a behavioral VCO. The other set is generated using Eq 2-15.

If we cannot use the narrowband approximation, analysis becomes much more difficult. [7] presents an equation for FM modulation assuming an arbitrary input signal. The equation is extremely cumbersome for a signal comprised of a large number of sinusoids.

Section 6.3.3 discusses test results looking at the effect of β on spur prediction.

2.4.3 Sigma Delta Modulator Clock Non-linearity

In a typical PLL the sigma-delta modulator is clocked on the rising edge of the feedback signal. In a locked Integer-N PLL this signal would be locked to the reference, so the divider would have a consistent clock driving it. With a frac-N PLL the feedback signal is constantly trying to sync with the reference as the

divider changes integer values. This results in an inconsistent clock driving the modulator.

3 State-of-the-Art

This section will describe a number of papers that contribute to the state-of-the-art in terms of noise analysis in PLLs. Phase noise models are examined and explanations are given as to why they cannot be used for spur prediction. Spur prediction models are also compared to the proposed models in this thesis.

3.1 Closed-Loop Nonlinear Modeling of Wideband Sigma Delta Fractional-N Frequency Synthesizers [3]

[3] presents a model for in-band spur prediction. An equation for the VCO tuning voltage is presented. One limitation of [3] is that the output of the model is given at the input of the VCO. The proposed model in this thesis shows a method to find spur level at the output of the VCO.

One other limitation of [3] is that only a very small set of test results are published. Only a single spur for two different offset frequencies is presented. [3] discusses little in terms of out of band spurs.

This paper states that the cause of reference spurs is mainly a result of PFD current leakage and charge pump mismatch. This thesis demonstrates how reference spurs will be present even if these non-idealities do not exist.

3.2 An accurate and fast behavioral model for PLL Frequency Synthesizer phase noise/spurs prediction [8]

[8] proposes a model for phase noise and reference spur prediction for an Integer-N PLL. The paper claims to support Frac-N architectures as well, but offers no results. One shortcoming of this paper is that the simulation times are reported to be on the order of hours. The proposed model in this thesis is designed to operate on the order of minutes or even seconds. Additionally [8] only deals with reference spur prediction.

The purpose of including this paper is that it deals with co-simulation with transistor level circuits in order to increase accuracy. The implementation of the proposed model in this thesis is somewhat limited in terms of the ability to swap out blocks for their transistor level counterparts. See Section 5.4.2 for a discussion regarding this.

3.3 A modeling approach for Σ-Δ fractional-N frequency synthesizers allowing straightforward noise analysis [9]

[9] claims to predict phase noise within 3dB over a frequency offset between 25 kHz and 10 MHz. A simplified time domain model from [9] is shown in Figure 23. ***θref[k]*** is the phase of the reference signal with reference cycle index ***k***. ***n[k]*** is the deviation in the divider value.

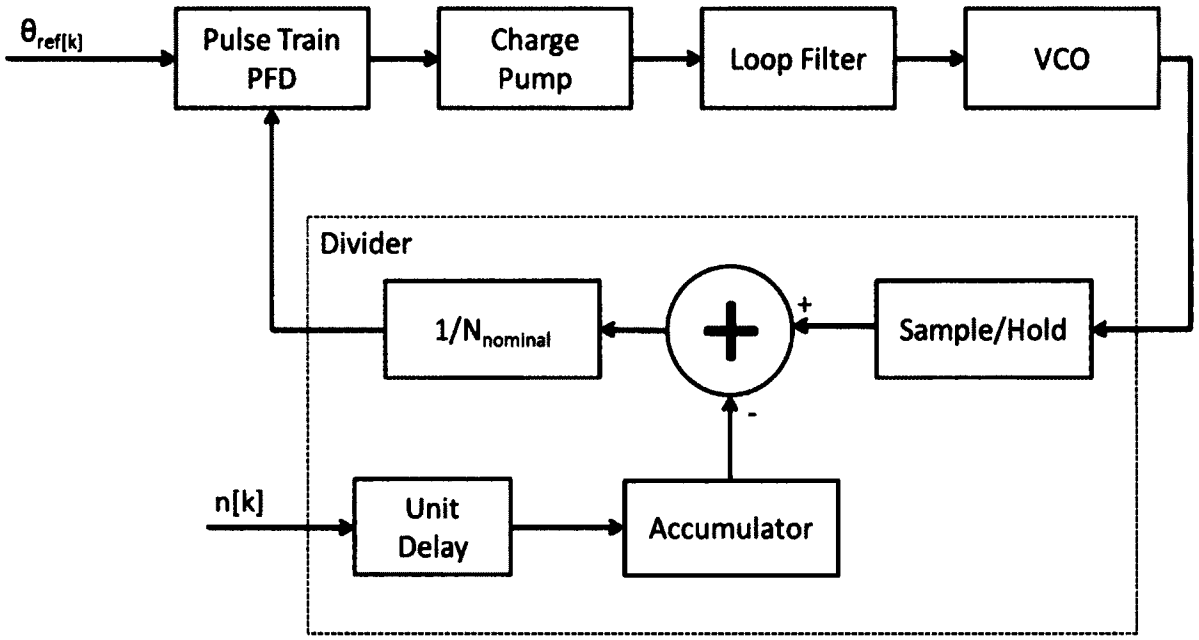


Figure 23 - Simplified Time Domain Model from [9].

The VCO in [9] includes an integration function. This means the output of the VCO is phase and not frequency. Section 5.2.4 explains why this does not allow for accurate Frac-N spur prediction.

[9] does not claim to predict Frac-N spurs, however it does introduce the concept of an impulse sequence that serves as a basis for the Pulse Train Generator that is used in this thesis (see Section 5.2.2). The ***Pulse Train PFD*** outputs an impulse that represents the phase error at ***k***. This impulse simplifies the charge pump output by eliminating its variable width and starting position as discussed in Section 2.4.1.

[9] also considers the varying nature of the divider clock based on a changing VCO output. This thesis must consider this variation as well.

3.4 *Analytical model and behavioral simulation approach for a ΣΔ fractional-N synthesizer employing a sample-hold element [10]*

[10] proposes the addition of a sample-and-hold circuit after the charge pump. Recall from Section 2.4.1 that a typical charge pump outputs a pulse that varies in width as well as position. The sample-and-hold operation eliminates these variances and instead outputs a pulse that is clocked on the falling edge of the reference clock, shown in Figure 24. The falling edge of the reference must be used since it is unknown whether the i_{cp} pulse will come before or after the rising edge of the reference clock. The important feature of Figure 24 is that the i_{CP_SH} signal is held over one reference cycle. A major benefit of this is the elimination (or at least reduction) of reference spurs since the pulses now begin and end at regular intervals. This phenomenon is explained in detail in Section 5.4.1.

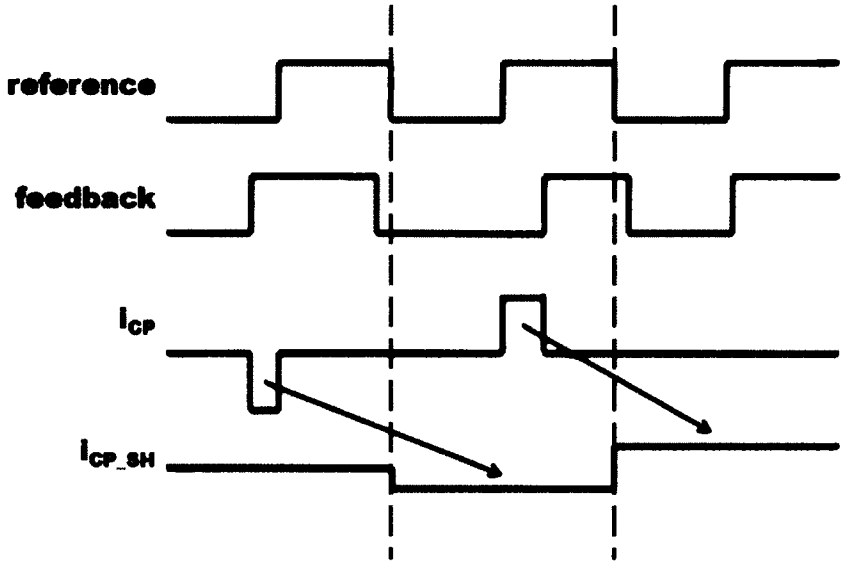


Figure 24 - Charge Pump output with Sample-and-Hold.

In terms of modeling this technique is also beneficial in that the pulse has now been simplified. The output pulse amplitude is now simply related to the phase error of the reference and feedback signal. For phase noise analysis this is acceptable, but in terms of modeling spurs a sample-and-hold operation will not work. This is shown is the test results, Section 6.3.8.

With this in mind [10] presents a mathematical analysis of the non-linearity produced from the i_{CP} pulse in Figure 24. From this analysis it presents a block diagram that could be inserted into an otherwise linear model. The paper does not implement this analysis into their results. Additionally, [10] does not comment on Frac-N spur prediction.

3.5 *Prediction of fractional-N spurs for UHF PLL frequency synthesizers [11]*

[11] is one of the few publications that presents results for spur prediction. Spur levels are predicted using Eq 3-1.

$$Spur_k = 20\log\left(\frac{\beta_k}{2}\right), for\ \beta_k \ll 1 \qquad \text{Eq 3-1}$$

β_k is the modulation index (as explained in Section 2.4.2) and is calculated using Eq 3-2.

$$\beta_k = \left|(I_{cp_k} + I_{comp_k})Z_{CL}\left(j2\pi k\frac{f_{ref}}{M}\right)\right|\frac{k_{vco}}{2\pi k\frac{f_{ref}}{M}} \qquad \text{Eq 3-2}$$

k is the index of the spur. I_{comp} is a compensation current that [11] develops with the purpose of reducing spurs. Z_{CL} is the closed loop transfer function of the PLL. The rest of the variables have been described previously in this thesis.

[11] claims any SDM architecture may be used. However, only results from a 1st order SDM are included. Frac-N spur accuracy may be off as much as an order of magnitude. [11] does offer a comparison between the model and an real-world synthesizer. The authors suggest that the order of magnitude error may be a result of the non-linear effects such as charge pump mismatch.

3.6 Paper Comparison

Table 1 offers a comparison between the papers previously discussed and this thesis. The benefits of the proposed models in this thesis over other models are fast, accurate prediction of Frac-N spurs up to the first reference frequency harmonic as well as support for 1st, 2nd, and 3rd order sigma-delta modulators. Other papers may only predict spurs at frequencies close to the carrier, support only 1st order sigma-delta modulators, or predict only phase noise and not Frac-N spurs.

Reference	Phase Noise Analysis	Reference Spur Prediction	Frac-N Spur Prediction (VCO Input)	Frac-N Spur Prediction (VCO Output)	Comments
[1]	Y	N	Y	N	- Only discusses in-band - Few results shown
[8]	Y	Y	N	N	- No Frac-N prediction
[9]	Y	N	N	N	- Only for phase noise
[10]	Y	N	N	N	- Only for phase noise
[11]	N	N	Y	Y	- Only uses 1ST order SDM - Results only show low frequency spurs
Thesis	N	N	Y	Y	- Frac-N prediction - Predicts up to reference frequency - Supports 3RD order SDM or higher

Table 1 - Paper Comparison

4 Sequence Length Equations for 2nd Order SDM

The sequence length of a sigma-delta modulator is important as it determines the spacing of the Frac-N spurs that occur at the output of the PLL. This chapter provides equations to determine the sequence length of a 2nd order Error Feedback Modulator (EFM). An EFM is an SDM architecture shown in [13]. The equations provided are based on empirical data. The increased complexity of 3rd order EFMs makes obtaining equations a difficult task. Therefore the discussion is limited to 2nd order EFMs.

Figure 25 presents a 2nd order Error Feedback Modulator (EFM). [13] predicts only minimum and maximum sequence lengths for 2nd order EFMs. This portion of the thesis is concerned with predicting the exact sequence length.

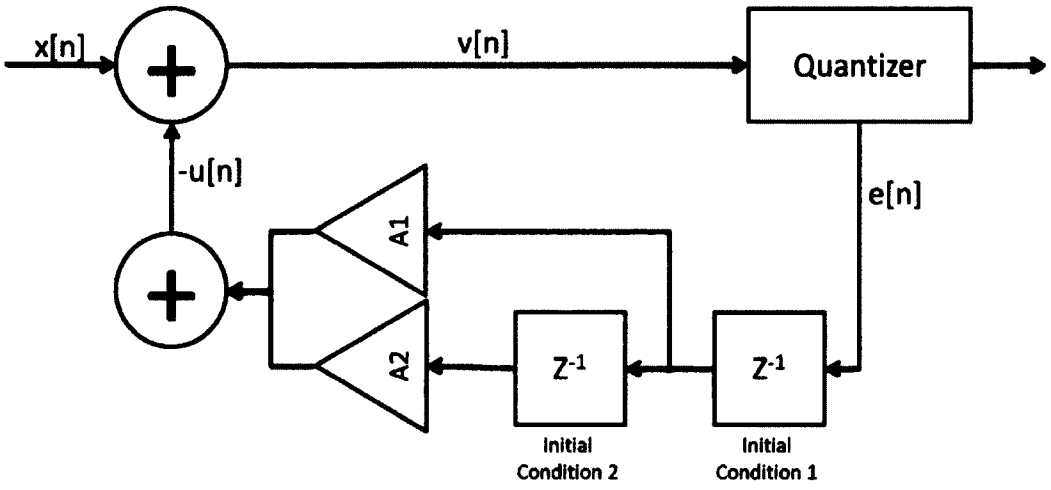


Figure 25 - 2nd Order Error Feedback Modulator Block Diagram.

A Cadence-based tool was developed to run a suite of simulations to determine sequence lengths. The simulation swept all valid combinations of X, M, and initial conditions up to M=16. Subsets of tests were also ran up to M=64 to verify the equations. The auto-correlation function was used on the simulations output to determine the sequence lengths. Based on the results, a number of equations were obtained to predict the sequence length.

Figure 26 shows a checklist for the equations that have been determined so far. The only equation that has not yet been discovered is when M is even and the initial condition sums to an even number.

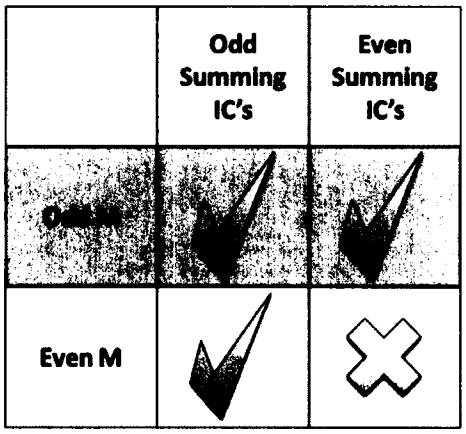


Figure 26 - Checklist for 2nd Order EFM SDM equations.

4.1 Odd M, Any Initial Conditions

For all odd M in a 2nd order EFM, a single equation can be used to find the sequence length (Ls_2, where the subscript 2 denoted the order of the EFM).

$$Ls_2 = \frac{M}{GCD(X, M, \Delta IC)}, where\ \Delta IC = abs(IC1 - IC2) \quad \text{Eq 4-1}$$

The sequence length is dependent on three variables; X, M and ΔIC. Recall X/M defines the fractional portion of the divider value. The GCD function finds the greatest common divisor. In order to increase the sequence length to its maximum possible value, designers can set ΔIC=1. This is most easily achieved by setting one of the initial conditions to 1 and the other to 0. Having ΔIC=1 reduces Eq 4-1 to Eq 4-2. This is also the case when X and M are co-prime.

$$Ls_2 = M \quad \text{Eq 4-2}$$

4.2 Even M, Odd Summing Initial Conditions

The equations for even M with odd summing initial conditions are presented here.

If X is odd:

$$Ls_2 = \frac{2M}{GCD(X, M, \Delta IC)} \quad \text{Eq 4-3}$$

If x=4, 8, 12, 16...

$$Ls_2 = \frac{M}{GCD(X, M, \Delta IC)} \qquad \text{Eq 4-4}$$

If x=2, 6, 10, 14...

$$Ls_2 = \frac{M}{GCD\left[Y, \frac{M}{Y}\right] GCD(X, M, \Delta IC)}, where\ Y = GCD(X, M) \qquad \text{Eq 4-5}$$

5 PLL Models

This chapter describes the models used in this thesis in detail. Section 5.1 describes the traditional behavioral model (called the ***behavioral model*** henceforth) which is used to verify the proposed models results. Section 5.2 describes the ***standard proposed model***. Section 5.3 describes the ***hybrid proposed model***. The ***standard proposed model*** and the ***hybrid proposed model*** are collectively called the ***proposed models***. Section 5.4 discusses the ramifications of the ***proposed models***. Section 5.5 proposes a charge pump architecture that can reduce reference spurs.

All models were implemented in Simulink due to the author's familiarity to the software. Any tools capable of behavioral simulation could be used.

5.1 Behavioral Model

Figure 27 shows a block diagram of the ***behavioral model***. The operations of each block are explained in Section 2.2. The purpose of this section is to show the implementation details of each block.

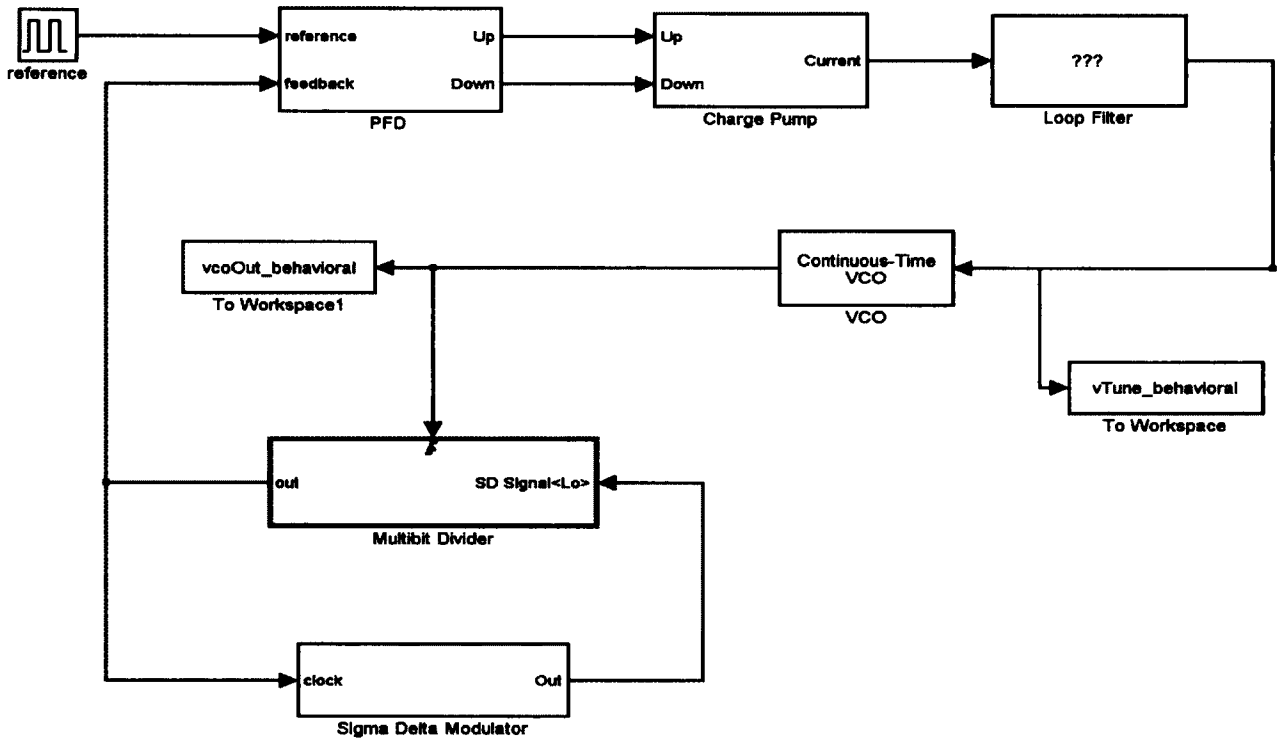


Figure 27 - Simulink Block diagram of behavioral PLL.

5.1.1 Reference cycle

The ***reference*** block is simply a 50% duty cycle pulse generator. Any type of signal can be used for this block so long as its rising edge mimics a reference clock edge properly.

5.1.2 Phase Frequency Detector (PFD)

Figure 28 shows a block diagram of the PFD. This is a traditional implementation of a tristate PFD [2].

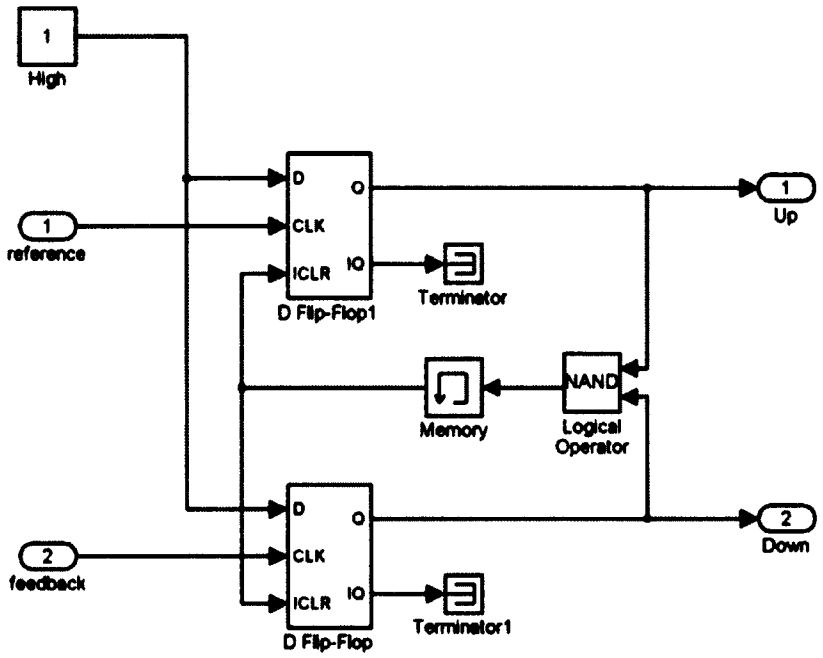


Figure 28 - Simulink Block level diagram of PFD.

5.1.3 Charge Pump

The charge pump is a standard design [1]. There are two currents called ***PositiveIcp*** and ***NegativeIcp***. These represent the current amplitude of the positive and negative currents. ***Up*** and ***Down*** pulses control ***Switch*** and ***Switch1*** to output a final current pulse ***Current***. Charge pump mismatch could be modeled here by setting ***PositiveIcp*** and ***NegativeIcp*** to unequal values.

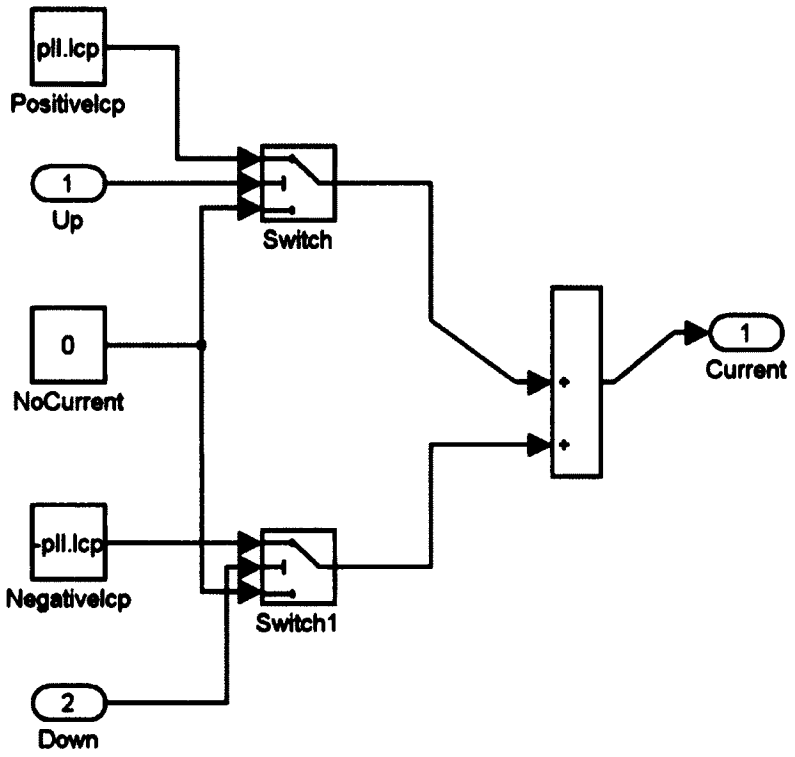


Figure 29 - Simulink Block Level Diagram for Charge Pump.

5.1.4 Loop Filter and VCO

Both the loop filter and VCO use standard MATLAB blocks. The loop filter is implemented as an ***LTI (Linear Time Invariant) System*** Block. This block can be assigned any transfer function. The VCO is implemented as a ***Continuous Time VCO***.

5.1.5 Multi-bit Divider

The multi-bit divider shown in Figure 30 uses an embedded MATLAB function. The code listing for the embedded function is shown in Figure 31. The divider has two inputs. ***I*** is a constant. ***SD signal*** is a signal that has an integer

value that may change each clock cycle. The clock is provided via the trigger block. At each clock edge an internal counter is incremented by one. The divider will roll over and reset when the internal count is greater than or equal to the constant *I* plus the value of ***SD signal***.

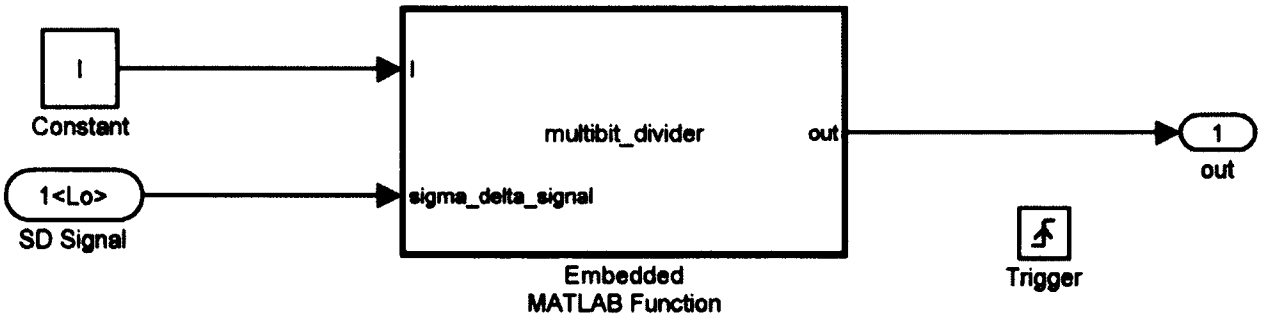


Figure 30 - Simulink Block Diagram of Multi-bit Divider.

```
function out = multibit_divider(I, sigma_delta_signal)

persistent counter;

if isempty(counter)
 counter=1;
end

if (counter >= (I+sigma_delta_signal))
   counter=1;
   out=1;
else
  counter=counter+1;
  out=0;
end
```

Figure 31 - Code Listing for Multi-bit Divider.

5.1.6 Sigma Delta Modulator

The top level of the sigma-delta modulator block (shown in Figure 32) selects which order of SDM is used based on the input signal ***sigma_delta_order***. An enable block is used to ensure only the correct block is active. This SDM block supports 1st, 2nd, and 3rd order MASH modulators. Figure 33, Figure 34, and

Figure 35 show the block level details of 1st, 2nd, and 3rd order sigma-delta modulators respectively. These modulators utilize ***Accumulator*** blocks (Figure 36) and ***Differentiator*** blocks (Figure 37).

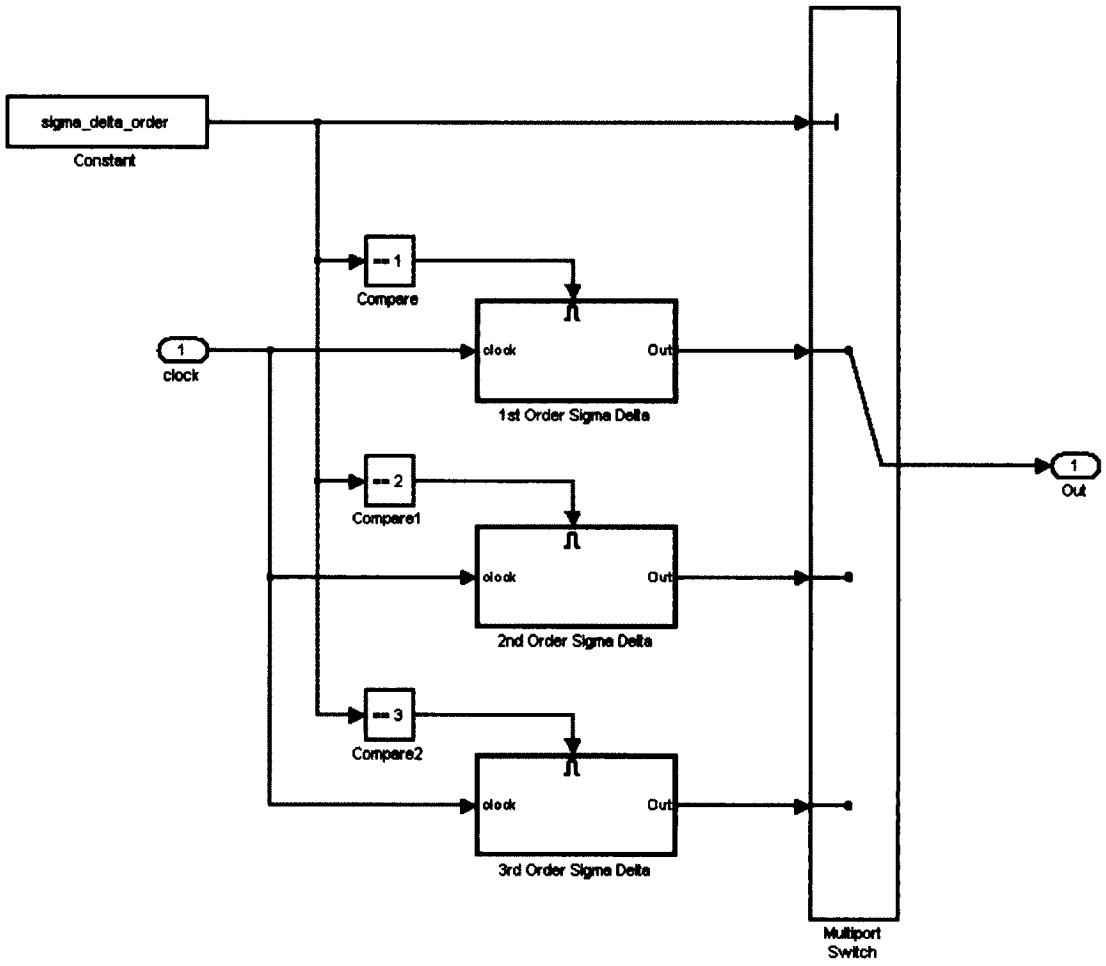


Figure 32 - Simulink Block Diagram of sigma-delta modulator.

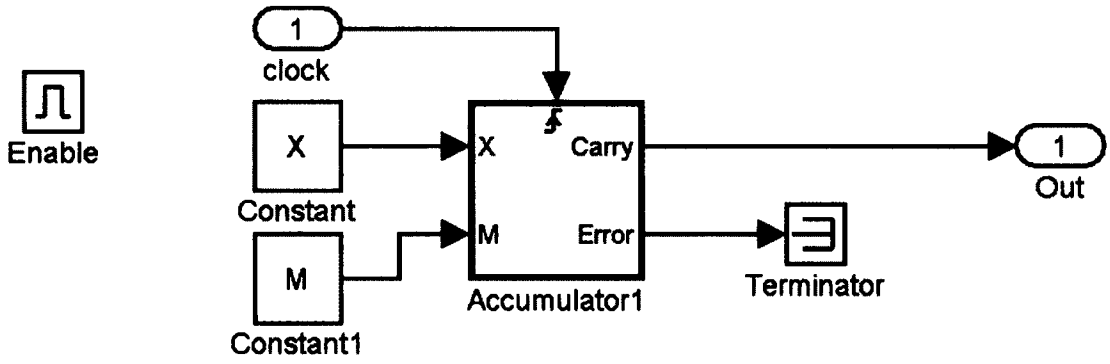


Figure 33 - Simulink Block Diagram of a 1st Order SDM.

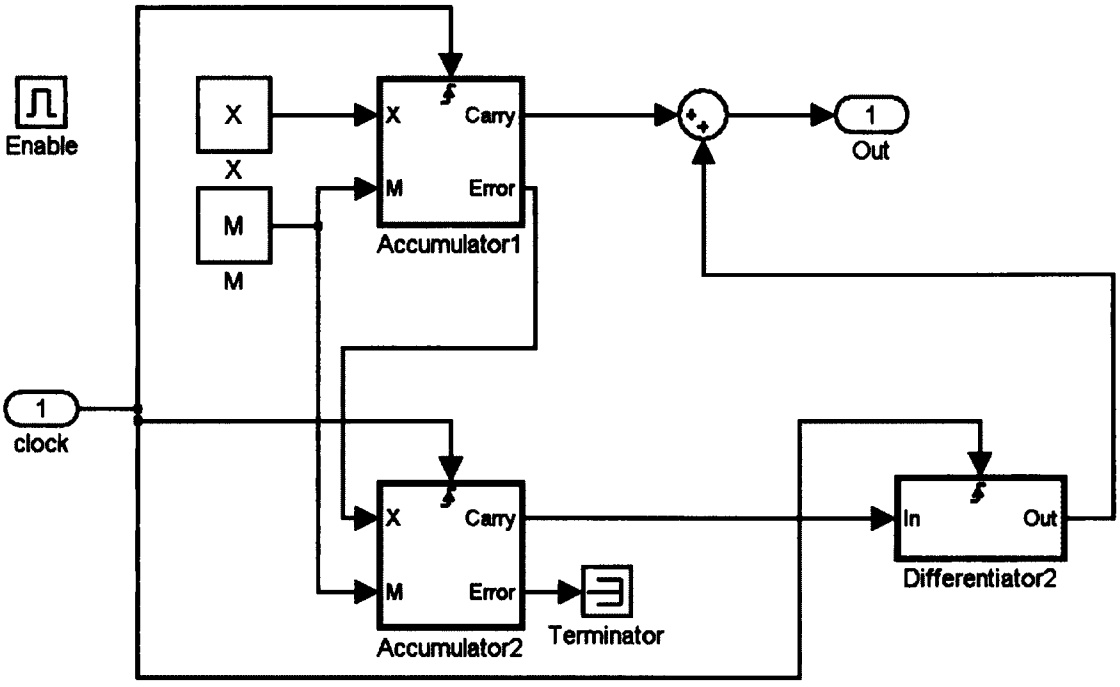


Figure 34 - Simulink Block Diagram of a 2nd Order MASH SDM.

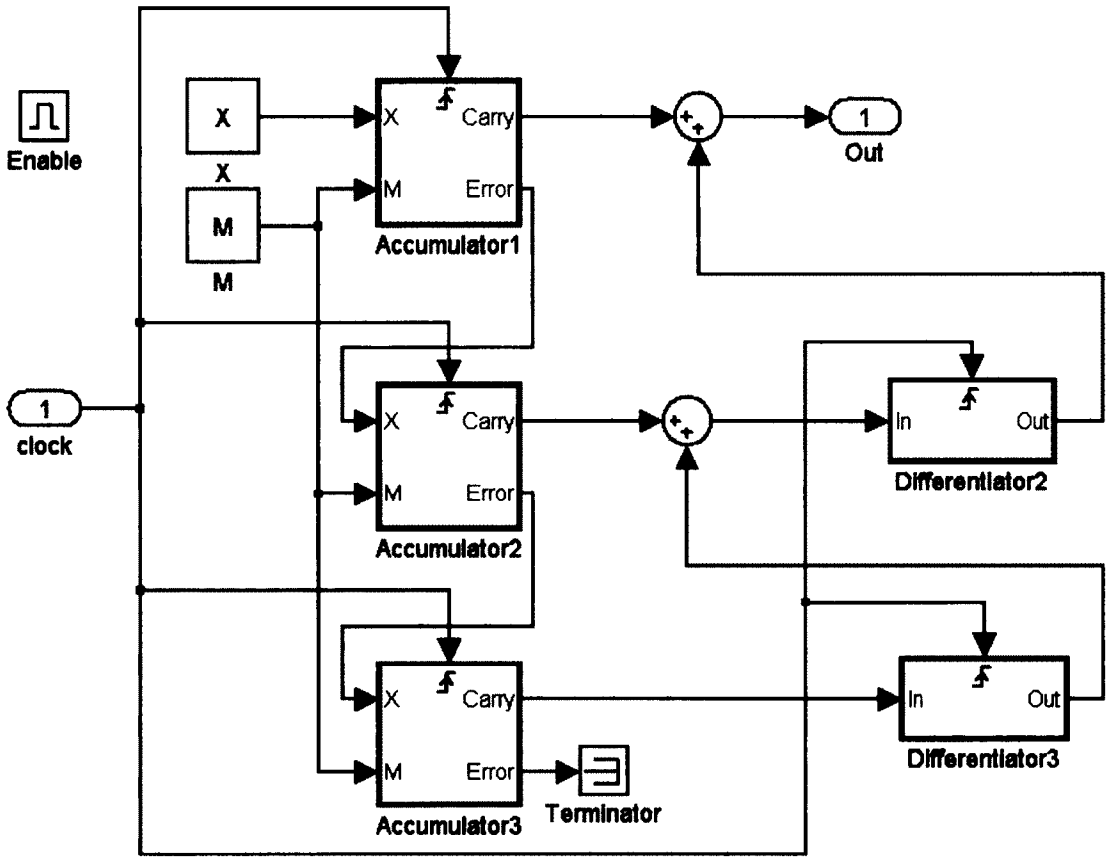


Figure 35 - Simulink Block Diagram of a 3rd Order MASH SDM.

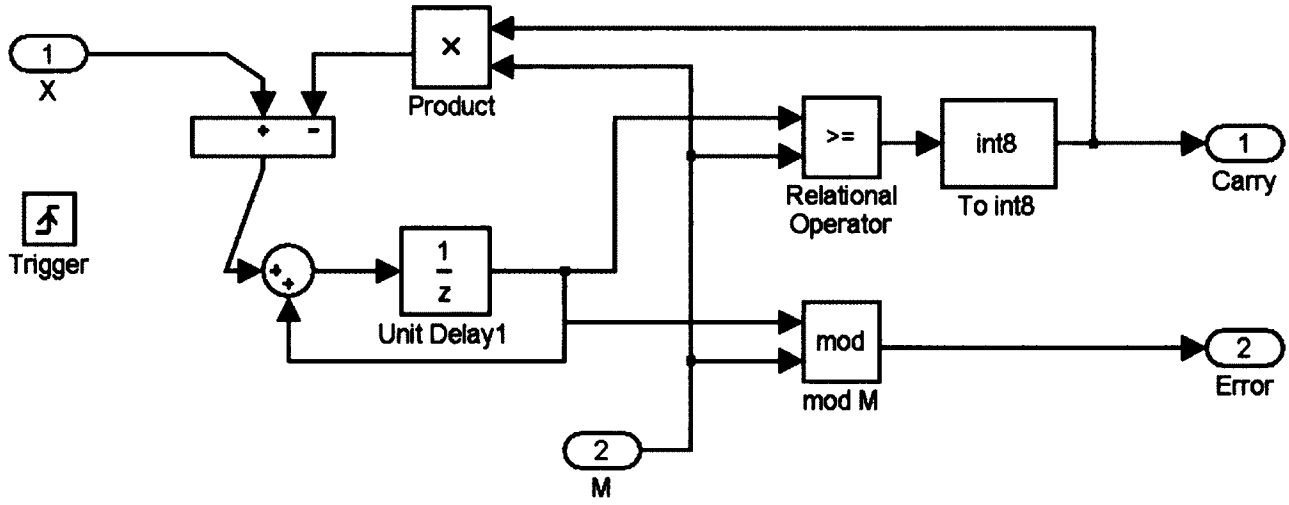


Figure 36 - Simulink Block Diagram for Accumulator.

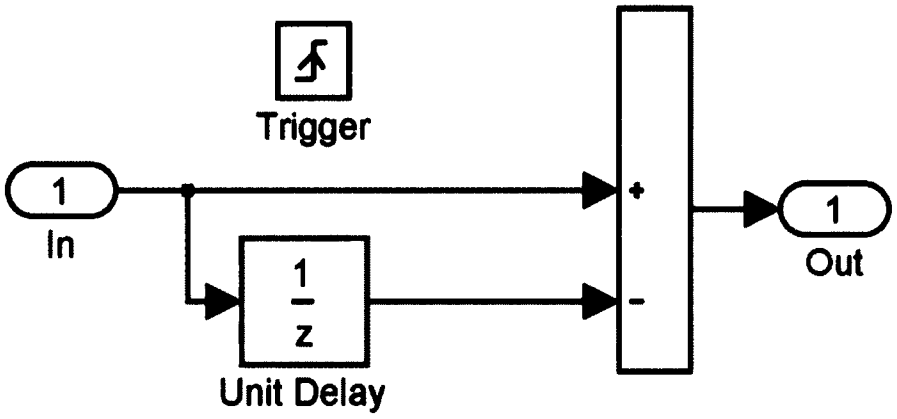


Figure 37 - Simulink Block Diagram for Differentiator.

5.2 Standard Proposed Model

Figure 38 shows the ***standard proposed model***. To reiterate, the purpose of this model is to provide accurate spur prediction at reduced simulation times when compared to the ***behavioral model***. Quick and accurate spur prediction will allow circuit designers to efficiently iterate through the initial design process.

There is a number of differences between this model and a ***behavioral model***. We will now examine each of these differences in detail starting from the reference signal.

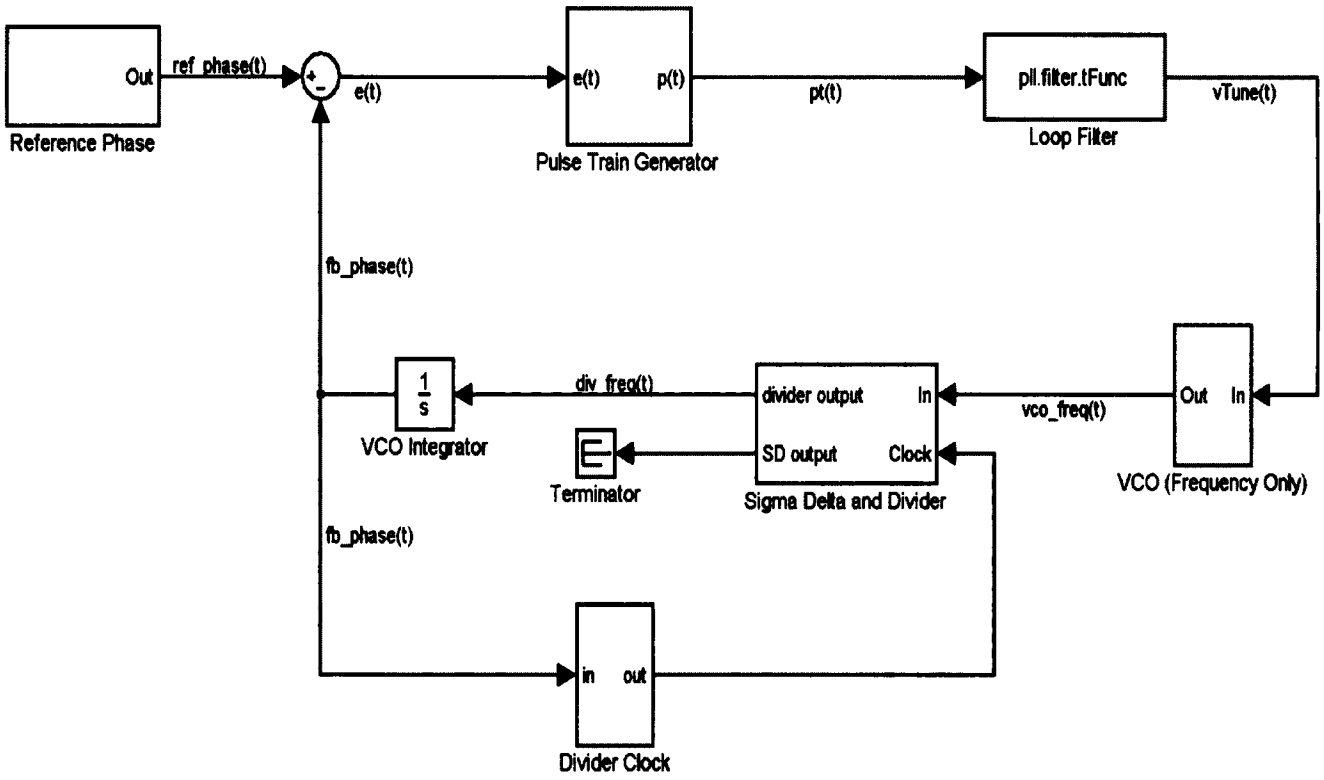


Figure 38 - Complete MATLAB Model.

5.2.1 Reference Phase

The ***Reference Phase*** is simply the constant reference frequency followed by an integrator. The reference phase can be thought of as a ramp in the time domain as phase accumulates. Figure 39 compares the reference signal of the ***behavioral model*** and ***proposed models***. The ***behavioral model*** is a constant sinusoid, whereas the ***proposed models*** contain only the instantaneous phase information of the sinusoid. This is a standard method of implementing a PLL model.

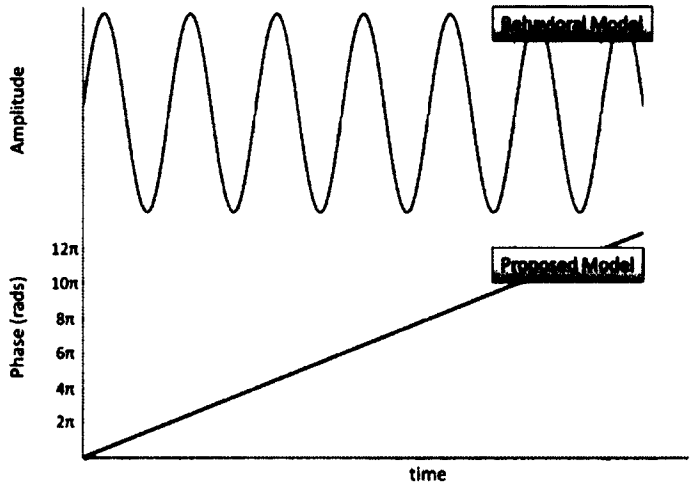


Figure 39 - Reference signal for Behavioral and Proposed Models.

5.2.2 Pulse Train Generator

The ***Pulse Train Generator (PTG)*** is a block that replaces the behavioral PFD/CP. It outputs a train of constant width pulses with the variable pulse amplitude reflecting the instantaneous phase error between the rising edges of the reference and feedback signals. Figure 40 shows the difference between the charge pump output in the ***behavioral model*** and the output of the PTG. The first difference to note is that the PTG output always begins on a reference clock rising edge. Secondly the pulse widths from the PTG are all the same. The phase error information is embedded in the amplitude of the signal, not the width of the pulse as with the ***behavioral model***. The ramifications of this are discussed in Section 5.5.

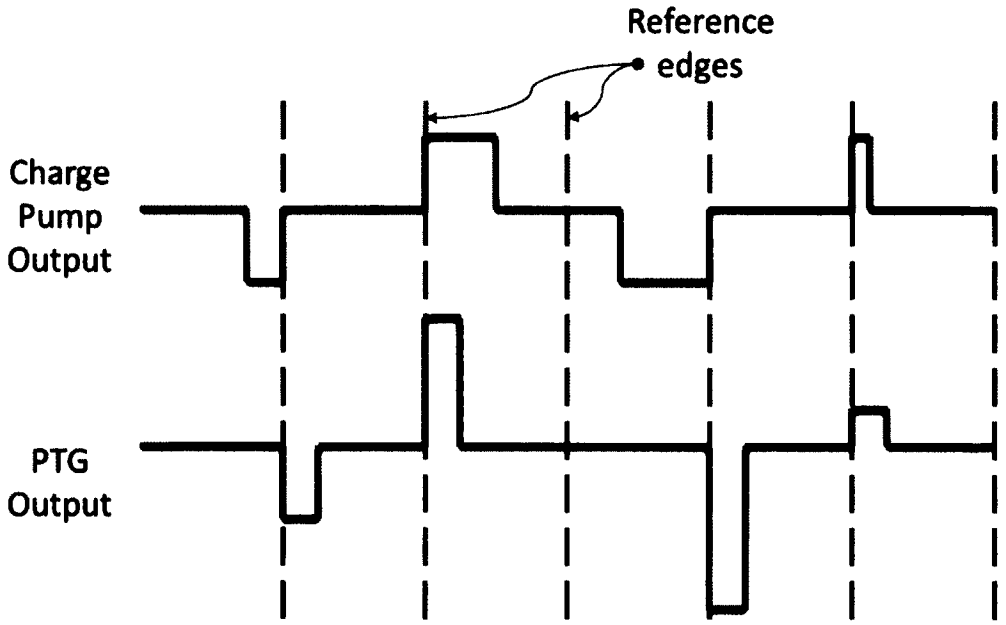


Figure 40 - Behavioral Charge Pump vs. Propose PTG output signals.

The PTG Output pulse can be expressed mathematically. To do so the signal is viewed as a set of time-shifted unit step functions with varying amplitudes as shown in Figure 41. Each pulse has a unique amplitude called a_n where n is the index of the pulse. This ranges from a_1 to a_p where p is the number of pulses in a period. T_{ref} is the period of the reference frequency and is equal to the length of time between pulse rising edges. T_{pw} is the time of a pulse width.

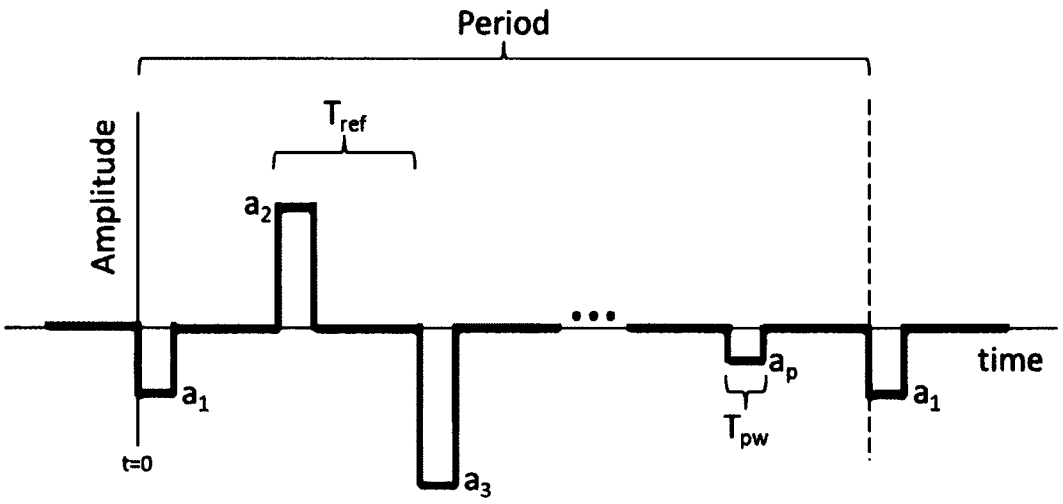


Figure 41 - PTG Output Pulse.

A generic PTG output pulse is defined in the time domain as shown in Eq 5-1.

$$\begin{aligned} x(t) = {} & a_1[u(t) - u(t - T_{pw})] \\ & + a_2\left[u(t - T_{ref}) - u\left(t - (T_{ref} + T_{pw})\right)\right] \\ & + a_3\left[u(t - 2T_{ref}) - u\left(t - (2T_{ref} + T_{pw})\right)\right] \\ & + \cdots + a_p[u(t - (p-1)T_{ref}) - u(t - ((p-1)T_{ref} + T_{pw}))] \end{aligned}$$

Eq 5-1

Expressed in the Laplace domain this yields Eq 5-2.

$$X(s) = \frac{a_1}{s}(1 - e^{-sT_{pw}}) + \frac{a_2}{s}\left(e^{-sT_{ref}} - e^{-s(T_{ref}+T_{pw})}\right) + \frac{a_3}{s}\left(e^{-s2T_{ref}} - e^{-s(2T_{ref}+T_{pw})}\right) + \cdots + \frac{a_p}{s}\left(e^{-s(p-1)T_{ref}} - e^{-s((p-1)T_{ref}+T_{pw})}\right)$$

Eq 5-2

Figure 42 shows a block diagram of the Pulse Train Generator. The ***Pulse Generator*** block takes an external parameter that sets its pulse width. Its frequency is the reference frequency. The ***Sample and Hold*** block stores the instantaneous phase error between the reference and feedback clocks. The ***Gain*** block generates the pulse amplitude of the PTG. The ***Switch*** block controls the pulse width.

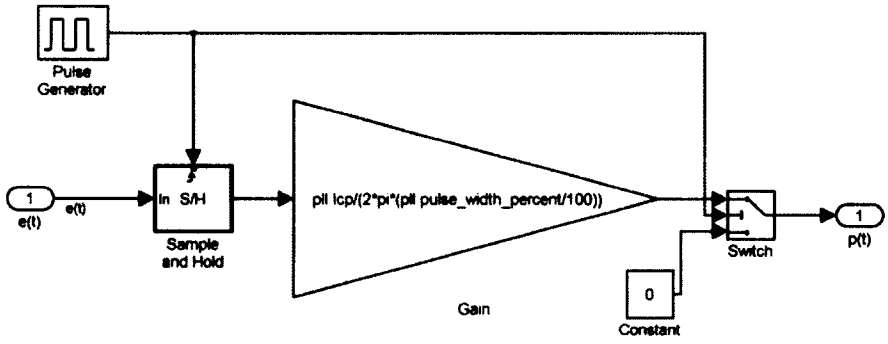


Figure 42 - Block Diagram of Pulse Train Generator

5.2.3 Loop Filter

The loop filter is implemented as an LTI Block as it is in the ***behavioral model***. This block can be assigned any transfer function.

The output of the loop filter is an extremely important location in the proposed models as this is the only location in which the signal matches up with the time domain signal seen in the ***behavioral model***. Therefore we obtain a DFT of the loop filter output of all models and use this to verify the accuracy of the ***proposed models***.

5.2.4 VCO and Divider

For reasons that will be explained shortly, we analyze the VCO and the divider simultaneously. Traditionally the equation to model a VCO is given by k_{vco}/s. Following that would be the divider. This is shown in Figure 43(a). From this we may imagine a Simulink implementation as shown in Figure 43(b) which consists of a k_{vco} gain block, an integrator, and finally the divider.

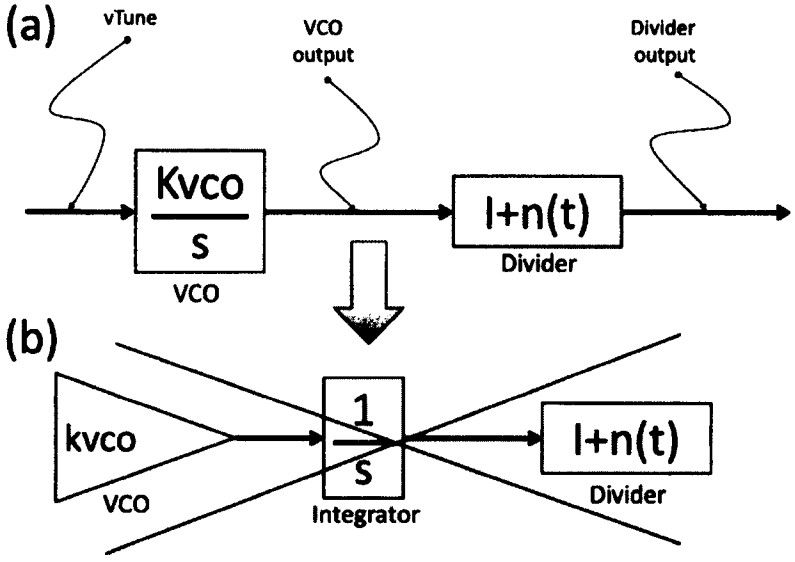


Figure 43 - Improper Modeling of the VCO and Divider.

The problem with this is the that signal being input to the divider is actually a *phase*, whereas before the integrator the signal was a *frequency* (recall the integral of frequency is phase). In the ***proposed models*** we cannot instantaneously divide phase if we wish to determine spur levels. Figure 44 demonstrates why this is so.

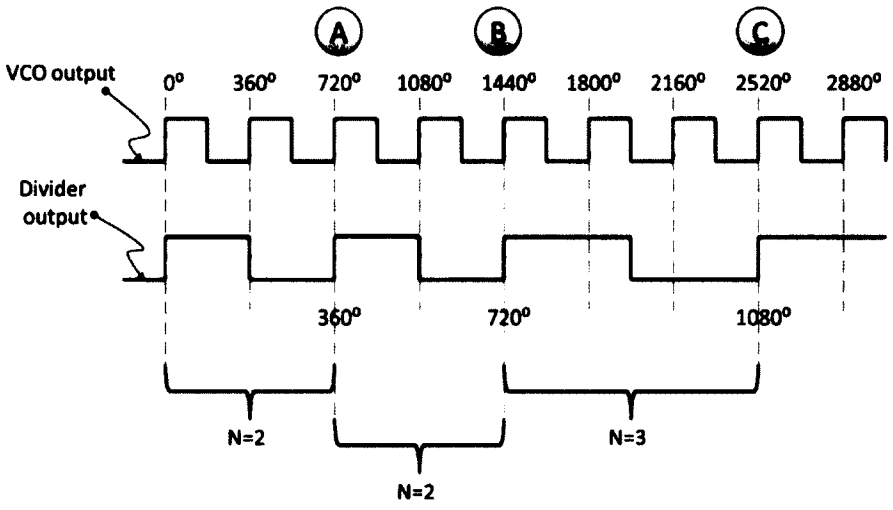


Figure 44 - Phase of original and divided down signals.

Points A, B, and C represent the rising edges of the divider output. If dividing the phase by N were permitted we should be able to divide the VCO output phase by N and get the divider output phase. For points A and B this holds true. The problem comes from when we change N from 2 to 3. At point C it becomes clear that the total divider output phase is no longer a divided-down total VCO output phase. However, Figure 44 does demonstrate that the *frequency* of the VCO output is always N times greater than the divider output

frequency. This implies that we should perform the division first *then* integrate the result as shown in Figure 45.

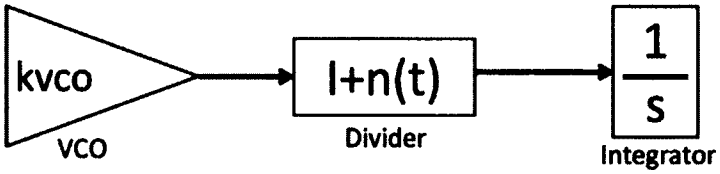


Figure 45 - Proper Modeling of the VCO and Divider.

Figure 46 shows the MATLAB implementation of the frequency portion of the VCO. The output of Figure 46 is a number that represents the frequency at the output of the VCO. This is in contrast to the ***behavioral model*** where the output of the VCO has its frequency information embedded within a sinusoid. This demonstrates a major advantage of the proposed model in terms of efficiency. The output of the VCO in the behavioral model requires a large number of samples to gather all spur information from the sinusoid. The required sampling rate increases as the output frequency of the PLL increases. Referring back to Figure 38 the output of Figure 46 feeds into the SDM/Divider block (explained shortly) and the VCO integrator.

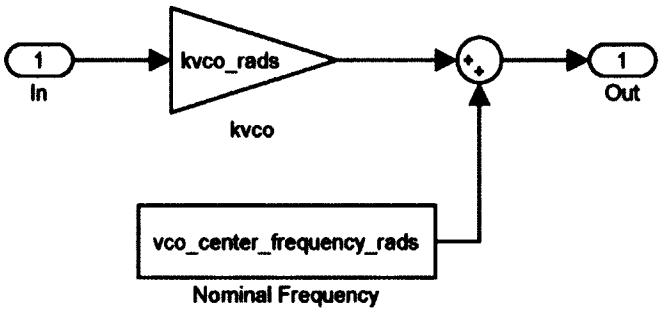


Figure 46 - MATLAB implementation of the frequency portion of the VCO.

Figure 47 shows the ***Sigma Delta and Divider*** block from Figure 38. ***In*** represents the instantaneous frequency that exists at the VCO output. This frequency is divided by the instantaneous divide value as defined by ***I*** plus the SDM output. The ***Rate Transition*** block is Simulink specific and can be ignored. The ***Sigma Delta Modulator*** block is the same block used in Section 5.1.6.

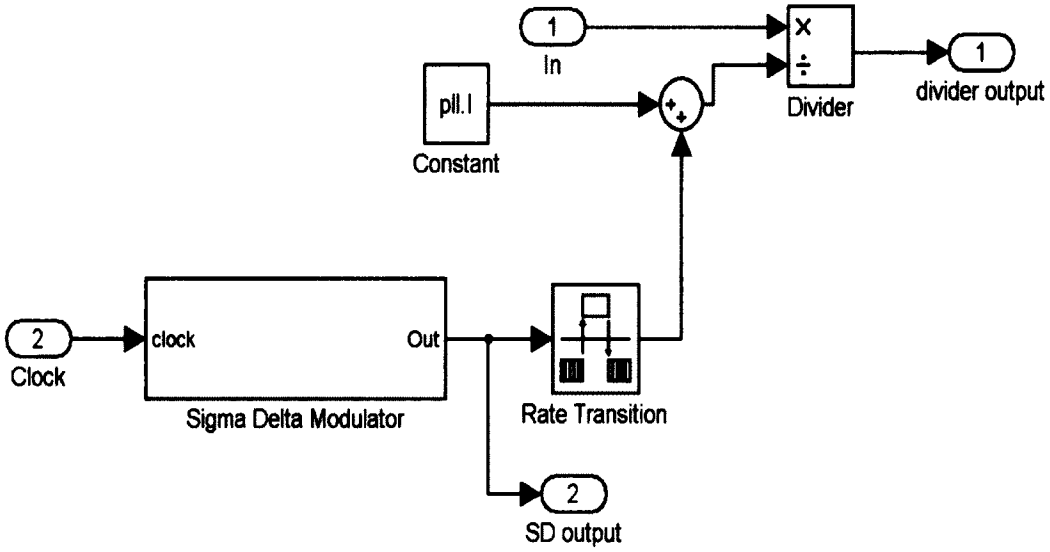


Figure 47 - Block Diagram of the SDM and Divider.

5.2.5 *Divider Clock*

The divider clock in Figure 48 is responsible for converting the feedback signal into a clock edge that the SDM can respond to. The ***Mod 2Pi*** block is used to determine when the input phase ***in*** wraps around 2π. This wraparound corresponds to a rising edge of the feedback signal. The signal after the modulus operation ramps up to 2π, then jumps down to zero. This 'drop' looks like a falling edge so the ***Gain*** block inverts the signal to present the drop as a rising edge. At this point the signal is bounded from -2π to zero, so π is added via the ***PI*** block to center the signal on zero. The signal can now be used as a clock.

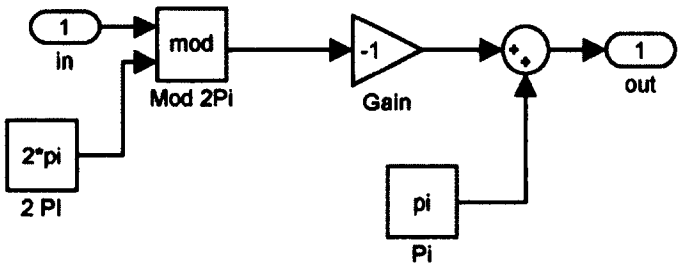


Figure 48 - Block Diagram of the SDM Clock.

5.3 *Hybrid Proposed Model*

One weakness of the ***standard proposed model*** is the inability to predict spurs close to the carrier. While the reason for this is not fully understood, test results (see Section 6.3.9) show that the prediction error stems from the way in which the ***standard proposed models*** Pulse Train Generator simplifies the ***behavioral models*** charge pump pulse train. To demonstrate this, a ***hybrid***

proposed model is developed. The model shown in Figure 49 replaces the Pulse Train Generator in the ***standard proposed model*** with the ***behavioral models*** PFD/CP.

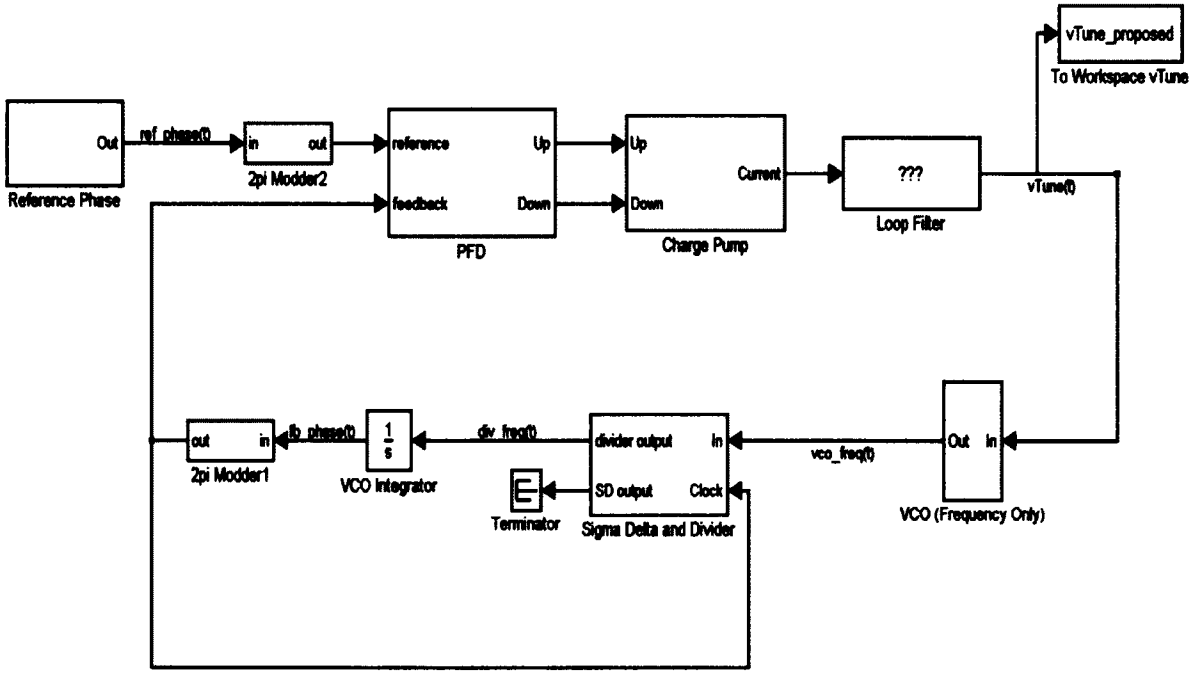


Figure 49 - Hybrid model block diagram.

The difference remaining between the ***hybrid proposed model*** and the ***behavioral model*** is now simply how the VCO and divider are handled. This model also requires the two **2pi Modder** blocks to extract clock edges from a phase.

5.4 Proposed Models Discussion

As in any model there are fundamental differences between it and what it is intending to model. This section is interested in discussing these differences.

5.4.1 Effect of Pulse Train Generator

Section 5.2.2 shows the difference between a pulse at the output of a ***behavioral models*** charge pump and the output of the ***standard proposed models*** pulse train generator. It is the differences between these signals that inhibit the ***standard proposed model*** from predicting reference spurs properly. In a behavioral PLL reference spurs can be made worse due to a number of causes such as charge pump mismatch (explained in Section 2.3.2.1). The reason that they exist in the first place is due to the varying position and pulse width of the charge pump current.

In order to show this we look at how the pulse train from the ***standard proposed model*** eliminates the reference spurs. We look at Figure 50(a) to define a generic signal which represents the output of the PTG.

We must place a few restrictions on the signal. The width of each pulse must be equal. The spacing between each pulse must also be equal. The amplitude of each pulse is set to be arbitrary to solve the general case. Finally the signal is periodic. A period here represents one periods output of an SDM.

We intend to use discrete Fourier analysis to examine the frequency spectrum of this signal. What the analysis will show is a total cancellation of the frequency components at integer multiples of the reference frequency. In the generic signal the 'reference spurs' are the frequencies related to the initial edges of each pulse. These edges represent the reference clock in a PLL.

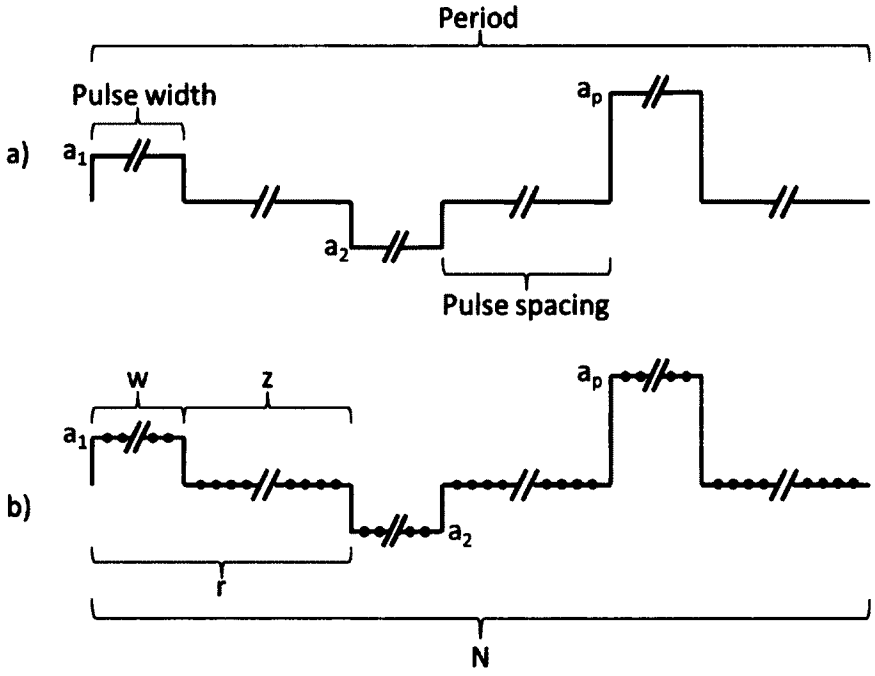


Figure 50 - This generic pulse train that represents the output of the proposed models pulse train generator. a) shows a continuous signal. b) shows the signal after sampling.

This signal is continuous in the time domain. We will sample this signal in order to use discrete-time Fourier analysis. Figure 50(b) shows the signal after it has been sampled. Table 2 defines the variables used in the analysis.

Variable	Variable Definition
x(n)	The time domain signal
p	Total number of pulses per period *N*
n	Index of the signal x(n)
w	Number of samples per pulse
z	Number of zeros between pulses
r	Reference period: *w + z*
N	Period of signal: *p(w + z)*

Table 2 - Variable Definition.

Now we define the discrete signal x(n) as:

$$x(n) = \{a_1\ a_{1\ldots}\ a_1\ 0\ 0\ \ldots\ 0\ 0\ a_2\ a_{2\ldots}\ a_2\ 0\ 0\ \ldots\ 0\ 0\ a_p\ a_{p\ldots}\ a_p\ 0\ 0\ \ldots\ 0\ 0\}$$

Using the discrete Fourier equation we can come up with the spur amplitudes of the reference frequencies. To begin we turn to the analysis equation (Eq 5-3) for discrete-time periodic signals [1], where k is the index of the frequency component. All other variables are defined in Table 2.

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-(j2\pi kn)/N} \qquad \text{Eq 5-3}$$

This can be expressed as follows, if we substitute in x(n).

$$\begin{aligned} c_k = \frac{1}{N}\Big[& a_1\left(1 + e^{-\frac{j2\pi k}{N}} + e^{-\frac{j2\pi k2}{N}} + \cdots + e^{-\frac{j2\pi k(w-1)}{N}}\right) \\ & + a_2\left(e^{-\frac{j2\pi k(r)}{N}} + e^{-\frac{j2\pi k(r+1)}{N}} + \cdots + e^{-\frac{j2\pi k(r+w-1)}{N}}\right) \\ & + a_3\left(e^{-\frac{j2\pi k(2r)}{N}} + e^{-\frac{j2\pi k(2r+1)}{N}} + \cdots + e^{-\frac{j2\pi k(2r+w-1)}{N}}\right) \end{aligned} \qquad \text{Eq 5-4}$$

$$+\cdots+a_p\left(e^{-\frac{j2\pi k((p-1)r)}{N}}+e^{-\frac{j2\pi k((p-1)r+1)}{N}}+\cdots+e^{-\frac{j2\pi k((p-1)r+w-1)}{N}}\right)\Big]$$

From Eq 2-10, the number of spurs between each reference frequency harmonic is equal to the sequence length, or p in this analysis. This means the first reference spur will occur at c_p. We now solve for c_p. This equation uses N=pr from Table 2 where appropriate.

$$\begin{aligned}c_p=\frac{1}{N}\Big[&a_1\left(1+e^{-\frac{j2\pi}{r}}+e^{-\frac{j2\pi 2}{r}}+\cdots+e^{-\frac{j2\pi(w-1)}{r}}\right)\\&+a_2\left(e^{-\frac{j2\pi(r)}{r}}+e^{-\frac{j2\pi(r+1)}{r}}+\cdots+e^{-\frac{j2\pi(r+w-1)}{r}}\right)\\&+a_3\left(e^{-\frac{j2\pi(2r)}{r}}+e^{-\frac{j2\pi(2r+1)}{r}}+\cdots+e^{-\frac{j2\pi(2r+w-1)}{r}}\right)\\&+\cdots+a_p\left(e^{-\frac{j2\pi((p-1)r)}{r}}+e^{-\frac{j2\pi((p-1)r+1)}{r}}+\cdots+e^{-\frac{j2\pi((p-1)r+w-1)}{r}}\right)\Big]\end{aligned}$$ Eq 5-5

We can express everything inside the brackets as X.

$$\begin{aligned}c_p&=\frac{1}{N}[a_1X+a_2X+a_3X+\cdots+a_pX]\\&=\frac{X}{N}[a_1+a_2+a_3+\cdots+a_p]\end{aligned}$$ Eq 5-6

How can we do this? In Eq 5-5 we look at the first terms in each of the brackets.

$$1 = e^{-\frac{j2\pi(r)}{r}} = e^{-\frac{j2\pi(2r)}{r}} = e^{-\frac{j2\pi((p-1)r)}{r}} \quad \text{Eq 5-7}$$

Now we look at the second terms...

$$e^{-\frac{j2\pi}{r}} = e^{-\frac{j2\pi(r+1)}{r}} = e^{-\frac{j2\pi(2r+1)}{r}} = e^{-\frac{j2\pi((p-1)r+1)}{r}} \quad \text{Eq 5-8}$$

This process can be done for all of the terms in the brackets.

From Eq 5-6 we can see that as long as the pulse-train amplitudes a_1 through a_p all sum to zero there is no reference spur. Note that X is irrelevant in this equation. The same strategy can be used to determine the spur amplitude for integer multiples of the p^{th} spur; that is, for any reference frequency harmonic.

In order to show the derived equation is valid MATLAB is used to generate an arbitrary signal that fits the description in Figure 50. We then use the fft function in MATLAB and compare those results to spurs calculated using Eq 5-4. The results of one such test are shown in Figure 51. Eq 5-4 exactly predicts the expected frequency amplitudes.

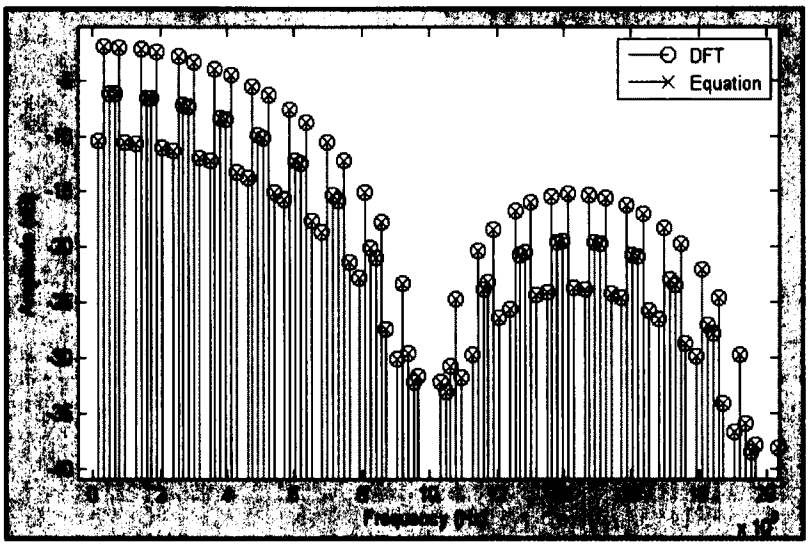


Figure 51 - The frequency components of a generic signal. The amplitudes are computed in two ways; using an fft() function from MATLAB and Eq 5-4.

Figure 51 was generated using a pulse train with seven distinct pulses. This results in a cancelation of every seventh frequency components. Recall that, in this example, every seventh frequency component is analogous to a frequency at which a reference spur would occupy in a PLL.

What has been shown here is why the ***standard proposed model*** fails to predict the reference spurs. Looked at from a more positive perspective, the ***standard proposed model*** demonstrates a technique of completely removing reference spurs. A design is presented in Section 5.5.1 to exploit this.

5.4.2 Using Transistor-level Blocks in Co-simulation

Co-simulation refers to a mix of behavioral and transistor level blocks in a model. Behavioral and transistor level blocks are identical in terms of their input

and output signals. This implies one can be swapped for another. This technique of replacing a behavioral block for a transistor level block may be used to improve accuracy at the expense of simulation time [8].

We refer to Figure 38 to identify what needs to be done in order to replace particular blocks of the ***standard proposed model*** with a behavioral/transistor level implementation. For convenience a simplified ***standard proposed model*** block diagram has been reproduced here in Figure 52.

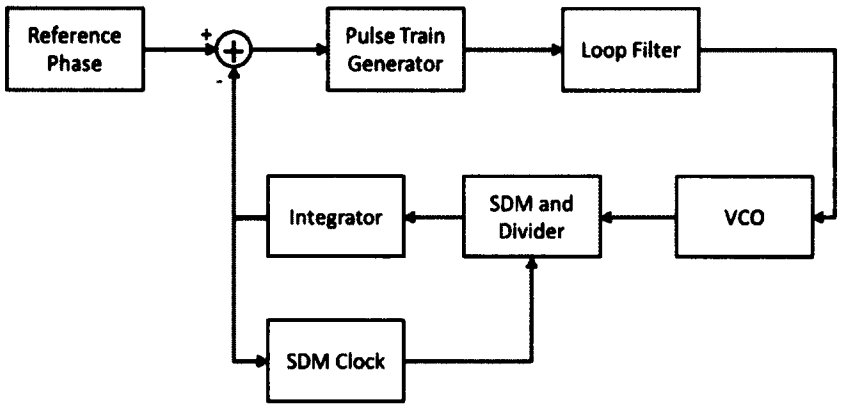


Figure 52 – A Simplified Block Diagram of the Proposed Model.

First we discuss replacing the ***Pulse Train Generator (PTG)***. Recall that the PTG (see Section 5.2.2) replaces the PFD/CP (see Sections 5.1.2 and 5.1.3) in the ***behavioral model***. The input to the PTG is an instantaneous error in phase. In order to replace the PTG with a behavioral/transistor level PFD/CP we must replace the phase error with actual clock signals that represent the reference clock and the feedback signal. These two locations correspond to the ***Reference Phase*** and the ***Integrator*** outputs in Figure 52. In the ***standard proposed model***

these two locations are simply a number that represents the instantaneous phase. We must extract a clock signal from these phases in order to plug into a behavioral/transistor level PFD/CP. We can do this simply by outputting a rising clock edge as each of the phase signals roll over 2π. This technique is employed in the ***Divider Clock*** block discussed in Section 5.2.5. The output of both the PTG and behavioral PFD/CP is a time-domain charge pump pulse, so the ***standard proposed model*** and ***behavioral model*** are compatible at this point. The procedure just explained was used in order to produce the ***hybrid proposed model***.

We now look at replacing the VCO. The output of a traditional VCO is a time domain sinusoid of a frequency that varies with the VCO input. The ***standard proposed models*** VCO output is simply a number that represents the instantaneous frequency. Swapping in a traditional VCO means the ***standard proposed models*** divider must also be changed to respond to clock edges. In the ***standard proposed models*** this division is a mathematical operation; that is the instantaneous frequency is divided down by whatever divide value is currently given by the SDM. This means swapping in a transistor-level VCO would also imply swapping in a divider from the ***behavioral model***. This swap would also change the nature of the signal feeding back into the summer in Figure 52. In short, swapping out the VCO is not a practical endeavor.

The SDM operates the same in the ***behavioral model*** and ***proposed models***. A swap here could be performed seamlessly.

5.5 Fixed Width Variable Amplitude Charge Pump (FWVACP)

An interesting result from this work stems from the ***standard proposed models*** inaccuracy in spur prediction around the reference spurs. In fact the ***standard proposed model*** doesn't actually have any reference spurs. Section 5.4.1 implies that if we could have a charge pump that outputs a pulse train with a fixed width and a variable amplitude reference spurs could be eliminated. Section 5.5.1 presents one potential design.

5.5.1 FWVACP Design

Figure 53 below shows the implementation details for the FWVACP. A corresponding timing diagram is provided in Figure 54.

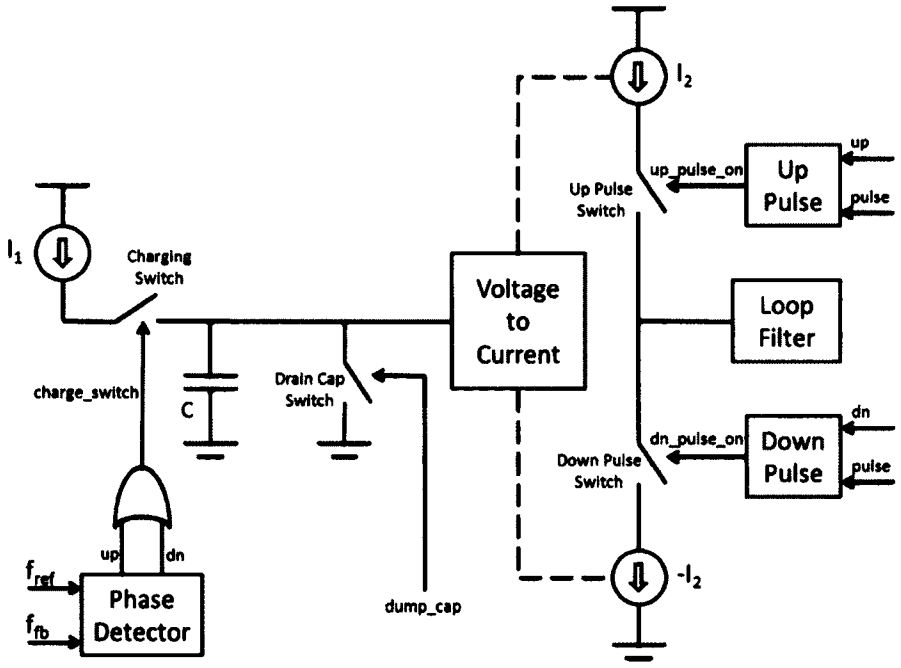


Figure 53 - Block Diagram for a FWVACP.

The two circuit inputs are f_{ref} and f_{fb}. f_{ref} is a reference clock. f_{fb} represents the feedback signal in a PLL. The 'Phase Detector' block compares f_{ref} with f_{fb} and sends out an 'up' or 'down' pulse that relates to the phase error for a given cycle of f_{ref}. To this point, the behavior is that of a traditional tristate phase detector based PLL.

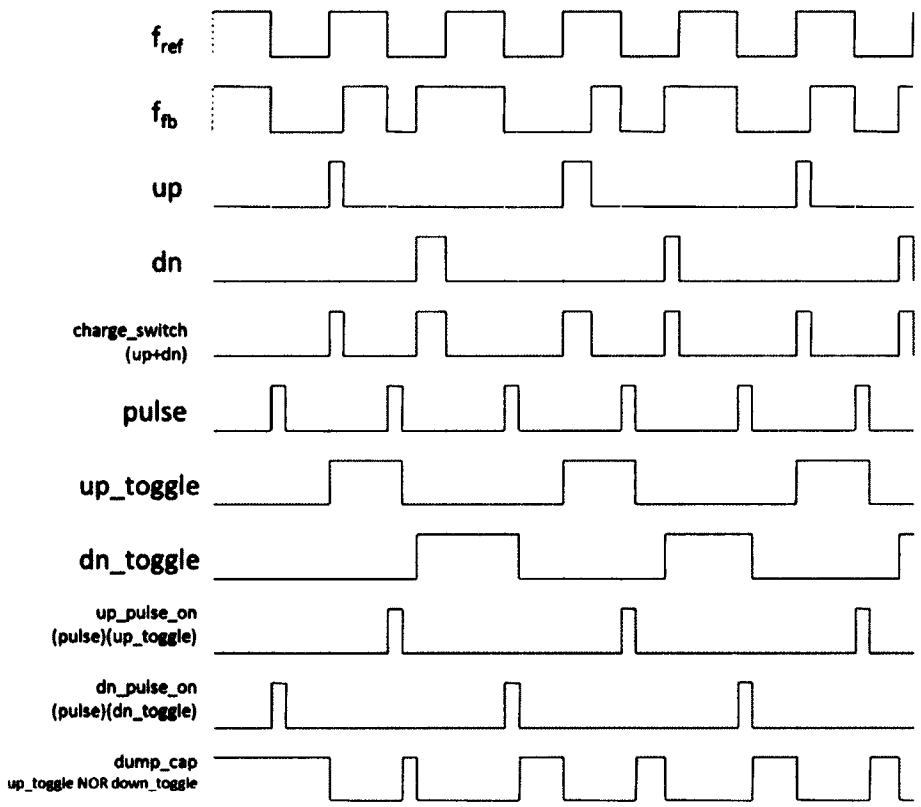


Figure 54 - Timing Diagram for a FWVACP.

The ***up*** or ***down*** pulses OR'ed together generate the ***charge_switch*** signal which engages the ***Charging Switch***. With the ***Charging Switch*** engaged current I_1 charges capacitor ***C***. The charge on the capacitor does not know if it resulted from an ***up*** or ***down*** pulse. This information is recorded in the signals from the ***Up Pulse*** and ***Down Pulse*** blocks, described later.

The charge on the capacitor ***C*** represents the accumulated phase error for a single reference clock cycle as defined by clock signal f_{ref}. The ***Voltage to Current*** block converts the voltage over ***C*** into a current I_2. With the current I_2 now set we now must engage a set of switches to pass the current on to the loop

filter. The ***Up Pulse*** and ***Down Pulse*** blocks are responsible for engaging ***Up Pulse Switch*** and ***Down Pulse Switch*** respectively.

Recall that a key idea with this design is that the signal that comes from the FWVACP must be of fixed width. The width of the pulse from these two blocks is constant with the width being controlled by the ***pulse*** signal. The pulse signal must begin on the falling edge of f_{ref} in order to capture phase error information from both ***up*** and ***down*** pulses.

The ***Up Pulse*** block works as follows. Signals ***up*** and ***pulse*** are fed into the block. A signal internal to ***Up Pulse*** called ***up_toggle*** goes high when an ***up*** rising edge is emitted from the ***Phase Detector*** and goes low at the falling edge of the ***pulse*** signal. ***up_toggle*** is then AND'ed with the ***pulse*** signal to generate ***up_pulse_on***. The purpose of ***up_pulse_on*** is to engage the ***Up Pulse Switch***. The ***Down Pulse*** block works in a similar way but instead responds to the ***down*** signal and controls the ***Down Pulse Switch***. With either the ***Up Pulse Switch*** or ***Down Pulse Switch*** switch is engaged a current is passed on to the loop filter.

The ***dump_cap*** signal is responsible for draining the charge on ***C*** at an appropriate time by engaging the ***Drain Cap Switch***.

In order for the FWVACP to operate as intended timing is important. Switches ***Up Pulse Switch*** or ***Down Pulse Switch*** can only be engaged when the capacitor C has fully stored the charge for a given reference cycle. Said another

way if either ***Up Pulse Switch*** or ***Down Pulse Switch*** are engaged the ***Charging Switch*** must NOT be engaged. Also the capacitor ***C*** must be fully drained before the Charging Switch is engaged. Finally if a pulse is being output to the loop filter the capacitor must not be draining.

This circuit must produce a proper amount of current for a given phase error. The gain of the ***Voltage to Current*** block must be tuned to the width of the ***pulse*** signal in order to give the correct current.

5.5.2 Arbitrary Spur Eradication

Varying the width of the FWVACP pulse is a technique that can be used to eradicate specific frequencies. A signal that has the form of the signal as described in section 5.4.1 will have nulls in the frequency domain at frequencies that are related to its pulse width.

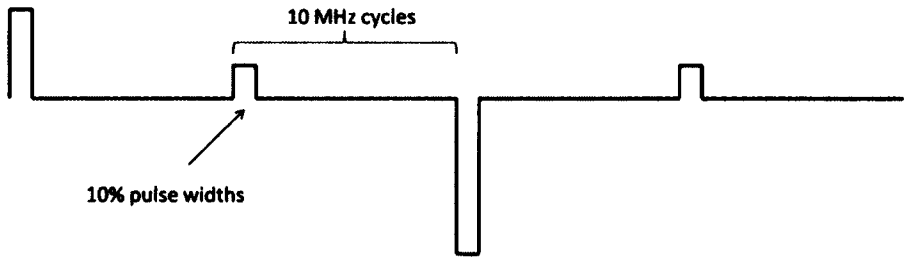


Figure 55 - Sample Signal.

Figure 55 shows the time domain of an example FWVACP pulse. The reference is at 10 MHz and the pulse widths are 10%. Figure 56 shows the frequency domain of the signal. First we see nulls every 10 MHz as predicted by

Eq 5-6. We also notice the sync function shape has a null at 100 MHz. Having a 10% pulse width gives the zero crossing at 10 times the reference. Having a pulse width of 100% would put the sync pulse null on top of the first reference pulse null (which has been done before in [10]). This implies that the first reference frequency harmonic is the furthest the sync pulse null can be pulled in.

Using this technique we can target any frequency greater than the first reference frequency harmonic by tuning the pulse width of the FWVACP.

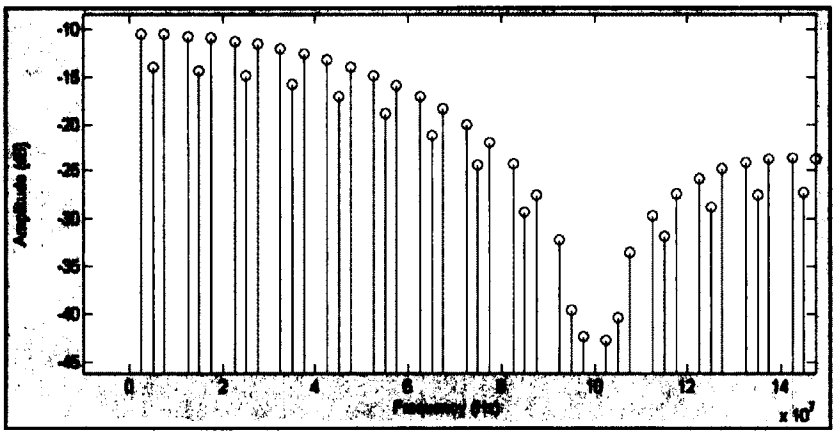


Figure 56 - Frequency Spectrum of an Example FWVACP Pulse.

6 Results

This chapter discusses the test results for the ***proposed models***. Section 6.1 discusses the test procedure. Section 6.2 outlines which tests will be run. The test results are discussed in Section 6.3. The full set of test results are presented in Appendix A.

6.1 Test Procedure

In order to verify the accuracy of the ***proposed models*** we must compare their results with the results of the ***behavioral models***. This is done by running a large number of test cases and drawing empirical conclusion from the results.

Some terminology must first be explained. A single ***test case*** covers a single PLL design. It includes the ***behavioral models*** results and one of the two ***proposed models*** results. A test case is defined by a ***test definition***. The test definition holds all the parameters of the PLL. All tests are run using Matlab scripts and Simulink models, collectively called the ***test harness***. The test harness was developed in order to handle an arbitrary number of test cases. Figure 57 shows the structure in which the test cases are stored. Individual test cases are grouped into ***test sets***. These test sets combined make up all of the test cases run. The purpose of test sets is to group similar test cases together. For example, one test set may sweep the order of the SDM, and another would be interested in changing the bandwidth of the PLL.

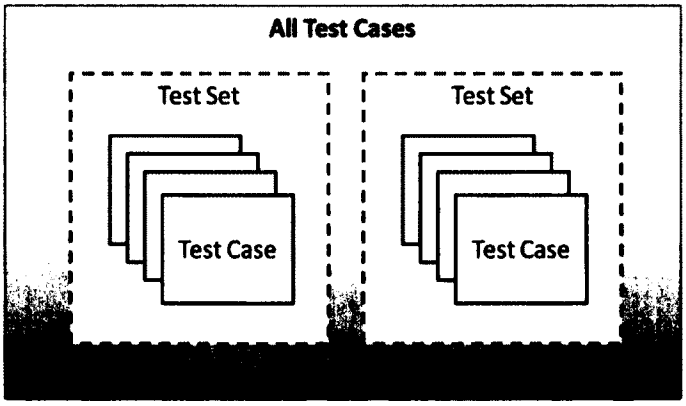


Figure 57 - Test Case Structure.

The high level work flow of the test harness is shown in Figure 58. First, all the test cases are defined and grouped into test sets. Each of the test cases are processed individually and then aggregated. A separate script is then run on the results to present the data in the form of graphs.

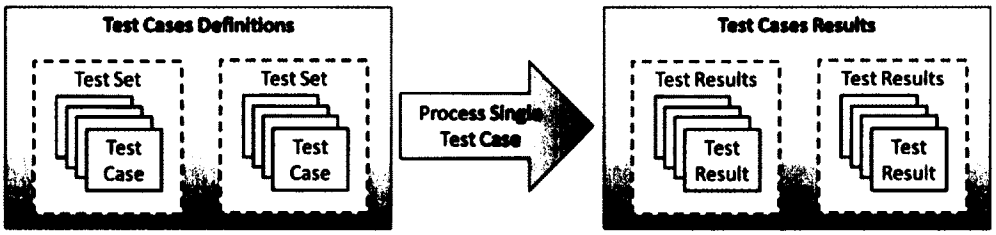


Figure 58 - High Level Test Harness Workflow.

Figure 59 shows the work flow each test case undergoes. The ***Setup Test Parameters*** block is used to set up a number of MATLAB structures that are read in by the Simulink models. Some examples of these parameters would be a PLL parameter such as f_{ref}, or a simulation parameter such as simulation time. The ***Run Each Test*** block runs both the ***behavioral model*** and one of the ***proposed models*** tests for each test case. Results from these simulations are gathered in

the ***Gather Test Results*** block. The ***Process Results*** block is used to prepare the results to be interpreted in a coherent manner. For example the results from the models are in the time domain, so an important processing step is to perform a DFT on the time domain data.

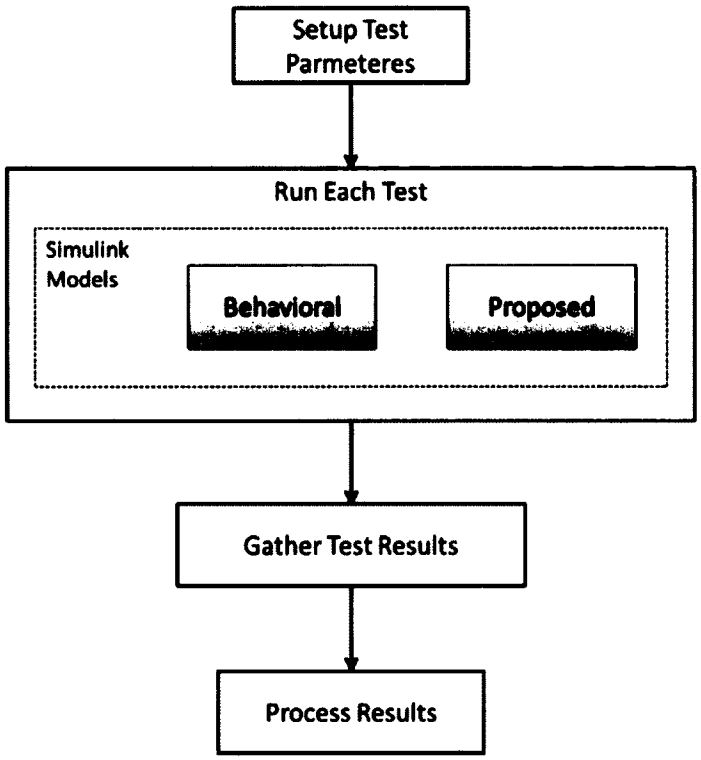


Figure 59 - Work flow for Test Procedure.

6.2 Test Setup

Each test case must compare the results of the ***behavioral model*** to the results of the ***proposed models*** at both the input and output of the VCO. Each model is capable of determining spurs at the input of the VCO so we can compare these spurs directly. However, as earlier discussed, the ***proposed***

models cannot show spurs at the output of the VCO. We must take the input of the VCO and run it through Eq 2-15 to determine what it would be at the output of the VCO. We can then take these results and directly compare them with the VCO output of the ***behavioral model***.

Each test case involves sweeping one or more parameters of a PLL. In order to perform proper comparisons we use the PLL in Table 3 as a basis. Unless otherwise noted the PLL under test will have the parameters in Table 3.

PARAMETER	QUANTITY	VALUE
f_{ref}	REFERENCE FREQUENCY	10 MHz
f_{BW}	BANDWIDTH	50 kHz
I_{CP}	CHARGE PUMP CURRENT	50 uA
f_{nom}	NOMINAL OUTPUT FREQUENCY	200 MHz
N_{nom}	NOMINAL DIVIDE RATIO	20
f_{center}	VCO CENTER FREQUENCY	200 MHz
k_{vco}	VCO CHARACTERISTIC	250 MHz/V
I	INTEGER DIVIDE RATIO	20
X/M	FRACTIONAL DIVIDE RATIO	1/20
SD Clocking	HOW SIGMA-DELTA IS CLOCKED	FEEDBACK
pw	CHARGE PUMP PULSE WIDTH	1%

Table 3 - Common PLL Parameters for test cases

We now look at a number of test cases which were designed in order to verify the model over a realistic range of design parameters. Table 4 lists the test cases that were run. It specifies which parameter was swept and what the parameters value was. Discussions of the results are found in Section 6.3. All of these test cases use the ***standard proposed model***. The ***hybrid proposed model*** results are discussed in Section 6.3.9.

Test Group	Test #	Swept Parameter	Parameter Value
A	1	Sigma-delta Order	1
	2	Sigma-delta Order	2
	3	Sigma-delta Order	3
B	4	X/M	1/40
	5	X/M	13/40
	6	X/M	37/40
C	7	X/M	3/5
	8	X/M	7/10
	9	X/M	13/20
	10	X/M	21/50
D	11	Bandwidth	10 kHz
	12	Bandwidth	50 kHz
	13	Bandwidth	500 kHz
E	14	F_{REF}	5 MHz
	15	F_{REF}	20 MHz
	16	F_{REF}	40 MHz
F	17	K_{VCO}	10 MHz
	18	K_{VCO}	50 MHz
	19	K_{VCO}	200 MHz
G	20	I_{CP}	10 uA
	21	I_{CP}	50 uA
	22	I_{CP}	500 uA
H	23	I	11
	24	I	21
	25	I	51
I	26	Pulse Width	1
	27	Pulse Width	25
	28	Pulse Width	50
	29	Pulse Width	99

Table 4 - Test Cases for Standard Proposed Model.

6.3 Test Results Discussion

Table 5 outlines the test results. The ***Results Affected*** column indicates whether sweeping a particular parameter had any effect on spur prediction. Simulation time is also presented. Section 6.3.1 discusses the simulation time improvement of the ***standard proposed model.*** Section 6.3.2 discusses deficiencies in prediction accuracy. Section 6.3.3 discusses the effect of Wideband FM on prediction accuracy. Sections 6.3.4 through 6.3.8 discuss each

of the test group's results. Section 6.3.9 discusses the ***hybrid proposed model*** results.

Test Group	Test #	Swept Parameter	Results Affected	Behavioral Model Simulation Time (s)	Proposed Model Simulation Time (s)	Approximate Simulation Time Improvement
A	*1*	SDM Order	Y	771	23	33x
	2	SDM Order	Y	818	26	31x
	3	SDM Order	Y	839	27	31x
B	4	X/M	N	843	28	30x
	5	X/M	N	849	28	30x
	6	X/M	N	851	28	30x
C	*7*	X/M	N	845	28	30x
	8	X/M	N	842	28	30x
	9	X/M	N	837	28	30x
	10	X/M	N	832	28	30x
D	*11*	Bandwidth	N	4245	136	31x
	12	Bandwidth	N	846	28	30x
	13	Bandwidth	Y	85	3	28x
E	*14*	fref	N	430	21	20x
	15	fref	N	1738	46	38x
	16	fref	N	3408	76	45x
F	*17*	kvco	N	834	27	31x
	18	kvco	N	838	27	31x
	19	kvco	N	843	27	31x
G	*20*	ICP	N	834	27	31x
	21	ICP	N	836	28	30x
	22	ICP	N	844	27	31x
H	*23*	I	Y	826	28	29x
	24	I	Y	839	27	31x
	25	I	Y	868	28	31x
I	*26*	Pulse Width	N	837	27	31x
	27	Pulse Width	Y	837	26	32x
	28	Pulse Width	Y	845	26	32x
	29	Pulse Width	Y	828	28	30x

Table 5 - Test Result Summary.

6.3.1 Simulation Time

In general simulation time is drastically decreased when using the ***standard proposed model***. Table 5 shows an average of 30x speed improvement. This figure may be misleading. The 30x speed improvement stems from the fact that most of the test cases presented have output frequencies that are similar to

each other. Test Group E shows how the simulation time improvement is related to f_{ref}, or more specifically to the output frequency of the PLL. Increasing f_{ref} while maintaining the same divide value greatly increases the PLL's output frequency. Since the output frequency is sinusoidal many more samples must be taken to capture the increased frequency information. The simulation time of the proposed model does not suffer this same issue.

6.3.2 Standard Proposed Model Deficiencies

In general the test results expose three situations in which the ***standard proposed model*** fails to accurately predict spur amplitudes. Firstly, high frequencies are not predicted properly. For this reason this thesis is only concerned with spurs up to the first reference spur. Secondly, spurs around the carrier are not predicted properly. In general spurs that fall within 10% of the carrier are predicted improperly. Thirdly, spurs around the reference spur are underestimated. In general spurs that fall within 10% of the reference frequencies are predicted improperly. Note that these results are from systems using a 3rd order PLL. As the order of the PLL decreases, the prediction accuracy increases.

These discrepancies in accuracy stem from replacing the behavioral phase detector and charge pump with the proposed pulse train generator.

Section 6.3.9 shows how the ***hybrid proposed model*** overcomes these deficiencies and discusses possible reasons why it is so.

6.3.3 Effect of Wideband FM on Spur Prediction Accuracy

This section discusses the effect of wideband FM modulation on spur prediction. Figure 60 shows the PLL output spur prediction error for three simulations with varying β_{tot} values. β_{tot} is the sum of the β values (the modulation index) for each frequency at the VCO input. Section 2.4.2 discusses how larger β values lead to wideband FM modulation. In order to accurately predict spurs this thesis assumes only narrowband FM modulation through the VCO. What Figure 60 demonstrates is the larger β_{tot} becomes, the larger the spur prediction error becomes.

Figure 60 also shows how β for each spur decreases the further the spur is from the carrier. This implies that spurs far from the carrier can be ignored when considering what type of FM modulation will occur.

The 500 kHz bandwidth test case in test group D is an example of wideband FM modulation through the VCO.

In general the proposed model will lose accuracy at the PLL output as β_{tot} exceeds 0.1.

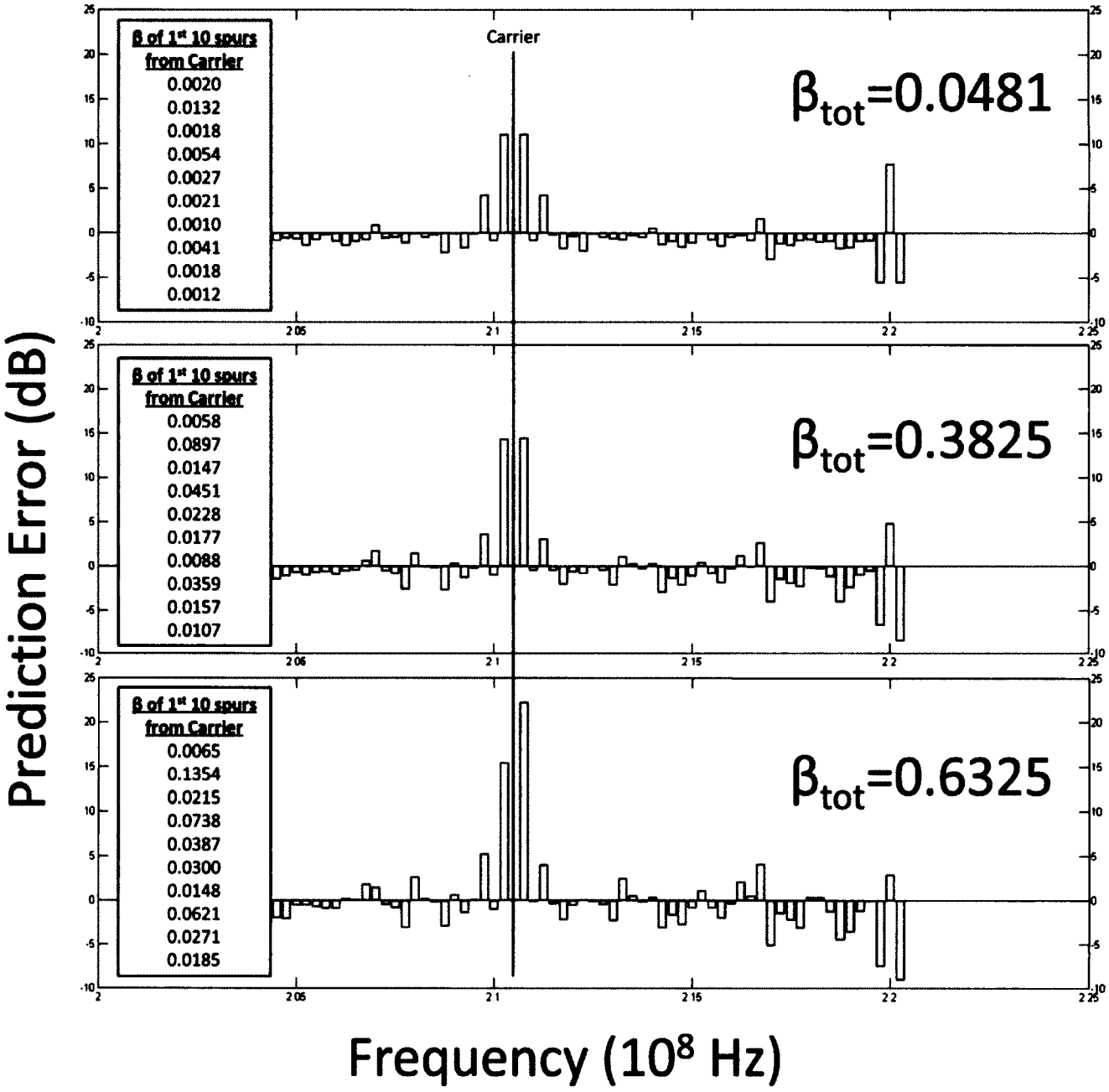


Figure 60 - Effect of Wideband FM modulation at PLL output.

6.3.4 *Test Group A – Sweep sigma-delta Modulator Order*

Test Group A is concerned with the order of the sigma-delta modulator. We see that the ***standard proposed models*** accuracy decreases as the order of the sigma-delta modulator increases. Subsequent test cases in this paper use a 3rd order SDM. For this reason we look at the test results as a whole and come up with an empirical observation that spurs are predicted accurately from 10% to 90% of the first reference frequency harmonic.

6.3.5 *Test Group B and C – Sweep X/M*

Test Group B and C both deal with changing the fractional divide ratio as defined by X/M. The results show that these parameters have no effect on the accuracy of the ***standard proposed model***. However, changing X or M can change the sequence length of the sigma-delta modulator output. From Eq 2-10 we see that as the sequence length increases the spurs spacing decreases. What this means is that there will be more spurs residing in the inaccurate regions as the sequence length increases.

6.3.6 *Test Group D – Sweep Bandwidth*

Test Group D sweeps the bandwidth of the PLL. Results show that bandwidth has little impact on the spur prediction at the input of the VCO. However, spur prediction at the output of the VCO may be radically off due to

wideband FM modulation. This is the case with the 500 kHz bandwidth test case. See Section 6.3.3 for more details regarding FM modulation.

6.3.7 Test Groups E through H – Sweep f_{ref}, k_{vco}, I_{CP}, and I

Test Group E looks at the reference frequency f_{ref}. Test Group F sweeps k_{vco}. Test Group G sweeps I_{cp}. None of these parameters have any effect on spur prediction accuracy. Test Group H deals with I, the integer portion of the divide ratio. The accuracy around the carrier seems to increases as I increases. This has not been looked into in enough detail to draw any conclusions.

6.3.8 Test Groups I – Sweep pulse width

This test group looks at different pulse train generator pulse widths. The pulse widths tested are 1%, 25%, 50%, and 99%. The trend is that the prediction accuracy decreases as the pulse width increases. Section 5.4.1 explains that the pulse train generator outputs pulses that show up as a sync function in the frequency domain. As the width of the pulses increase the first null of its sync function is pulled to lower frequencies.

Figure 61 shows the results of the 99% pulse width test. With such a high pulse width the first null of the sync function is located around the first reference frequency harmonic at 10 MHz. This null contributes to spur predictions that are lower than reality.

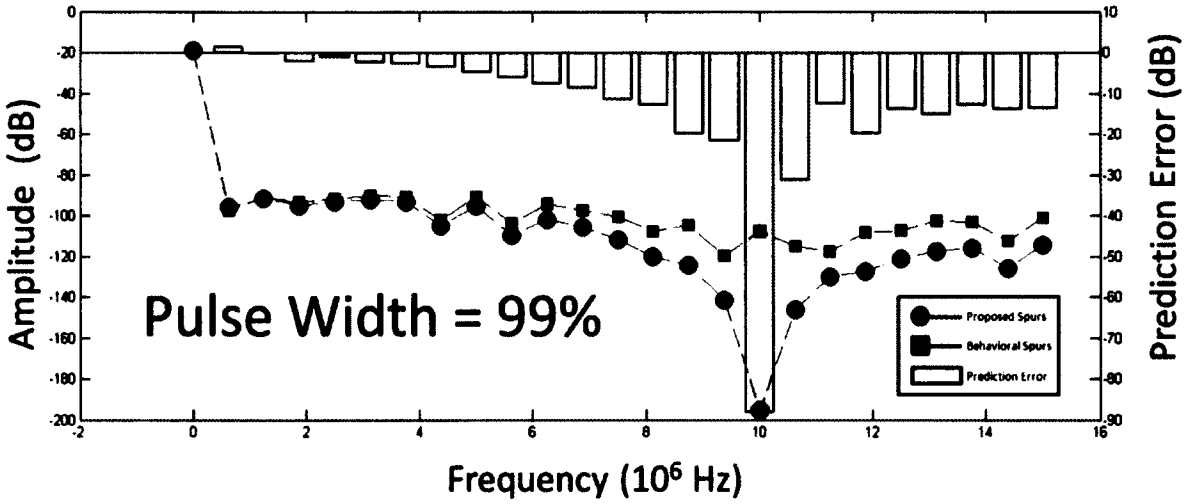


Figure 61 - Pulse width of 99%, results at VCO Input.

In contrast, the 1% pulse width test shown in Figure 62 shows accurate prediction up to near the first reference frequency harmonic. Note that the low first reference frequency harmonic prediction here is due to the fact that the pulse train pulses start on a reference cycle, and nothing to do with the null in the sync pulse. With a 1% pulse width the first null from the sync function is at a much higher frequency.

Lower pulse width percentages, such as 1%, provide more accurate spur prediction. This is because their width more closely resembles the width of the charge pump pulses that exist in the behavioral charge pump.

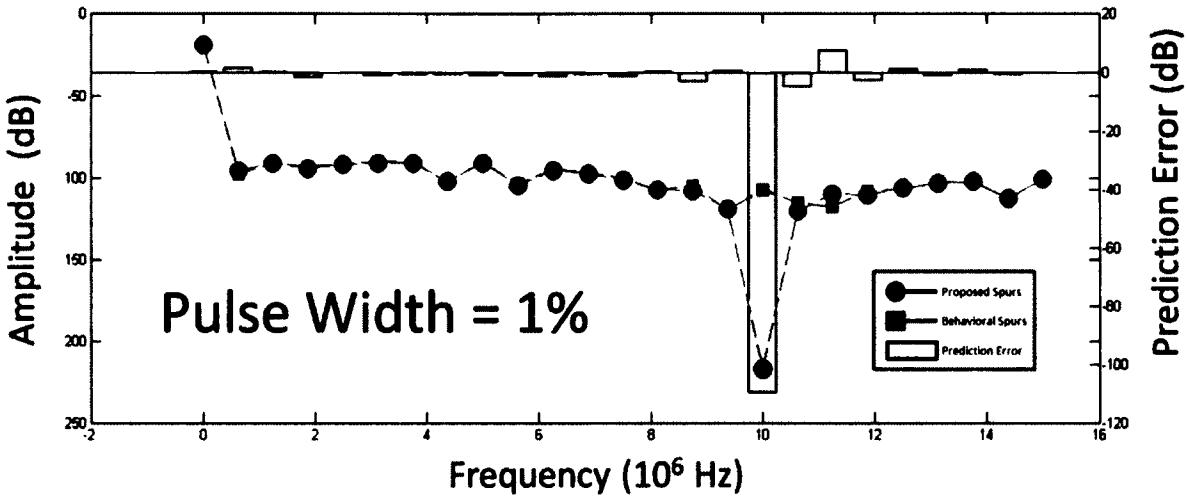


Figure 62 - Pulse width of 1%, results at VCO Input.

In summary, for most accurate spur prediction, it is best to choose a pulse width that most closely resembles the behavioral models charge pump pulse.

6.3.9 Proposed Model vs. Hybrid Model Comparison

Figure 63 shows a comparison at the VCO input between the proposed model and the hybrid model. Low frequency spurs are predicted much more accurately. Additionally the reference spurs are now properly predicted. This offers a clear example of how reference spurs are a direct result of the charge pump current pulses in a behavioral PLL.

The fact that now both reference spurs and carrier spurs are predicted accurately implies that the two may be related. This has not been proven.

While the ***hybrid proposed model*** provides a more accurate spur prediction it does so at the expense of simulation time. The ***hybrid proposed model*** was roughly 17x as fast as the ***behavioral model***. This is compared to a 30x gain with the ***standard proposed model***. The ***hybrid proposed model*** has not been tested extensively, but has been included to show a possible solution to the carrier spur and reference spur prediction issues.

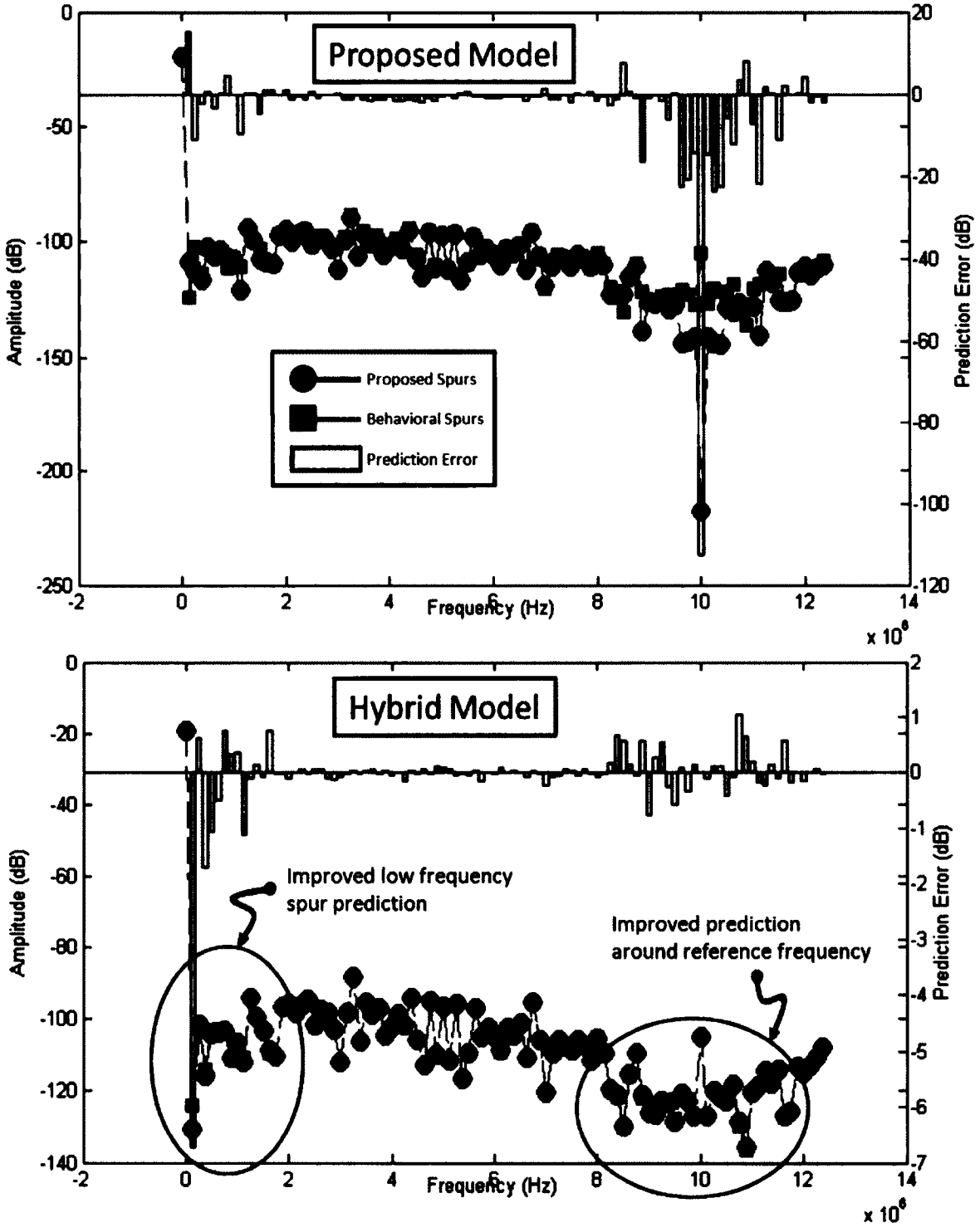


Figure 63 - Comparison between hybrid and proposed models.

7 Conclusion

The two ***proposed models*** presented in this thesis have been shown to predict spur amplitudes and locations accurately over a frequency range of up to the first reference frequency harmonic. The models enjoy a drastic speed increase over their traditional ***behavioral model*** counterpart due to the manner in which the PLL output is expressed. This speed increase becomes more apparent at higher PLL output frequencies.

Accurate spur prediction is largely a result of the method for implementing the divider function of the PLL. The ***proposed models*** divides frequency and not phase, in contrast to tradition ***behavioral models***. The trade-off between the ***standard proposed model*** and the ***hybrid proposed model*** is accuracy for simulation speed. The ***hybrid proposed model*** is slower but predicts close-in spurs and spurs around the first reference frequency harmonic more accurately. Both ***proposed models*** maintain the violent changes that occur at the VCO input. These changes are required to predict spur amplitude properly and are overlooked in traditional models.

The ***Fixed Width Variable Amplitude Charge Pump*** was developed in response to the ***standard proposed model*** and allows for a drastic reduction in reference spurs. With this design it is not only possible to reduce reference spurs but also to target spurs at specific frequencies at or above the first reference frequency harmonic.

The analysis of the pulse train generator output demonstrates that reference spurs do exist even in an ideal system. For this reason their existence is not solely attributed to non-idealities such as charge pump mismatch. It is simply the shape of the ideal charge pump output that is a major contributor to reference spurs.

An empirical equation was presented that accurately predicts the location of Frac-N spurs for 1^{st}, 2^{nd}, and 3^{rd} order sigma-delta modulators. Additionally, empirical equations that predict the sequence length of certain 2^{nd} order sigma-delta modulators were given.

Finally, this thesis discusses FM modulation that occurs in a VCO. An equation is given that predicts the amplitude of a spur at the output of a VCO given its amplitude at the VCO input. This equation assumes narrowband FM modulation, and a guideline is given to as to how to assess this for a particular PLL.

7.1 Future Work

A robust PLL model should not be limited to just spur prediction. It should handle phase noise as well. A number of papers mentioned in this thesis have varying phase noise model implementations. One item of future work is to integrate phase noise analysis.

All of the components used in the ***proposed models*** are ideal. Another item for future work is to integrate non-idealities into different PLL components. A linear VCO is impractical, so this would be one of the first non-idealities to examine as the equation Eq 2-15 relies on a fully linear VCO. Once real-world non-idealities are introduced the model would be ready to compare to a fabricated chip.

This thesis offers Simulink based models. In future, a fully mathematical equation that represents the models would be ideal. In theory a purely mathematical representation would operate even faster than the ***proposed models***. Another useful transfer function to generate would show how a spur at the SDM output translates to a spur in the PLL output.

One final work item is to implement the FWVACP.

7.2 Contributions

The contributions of this thesis are listed below:

- Two models have been presented that can be used to quickly and accurately predict Frac-N spur levels up to the first reference frequency harmonic.

- A design is offered to both reduce reference spurs and target specific frequencies in a PLL. A provisional patent has been filed based on this.
- Evidence that a major contributor to reference spurs is the nature of the charge pump pulse.
- An empirical equation that predicts Frac-N spur spacing.
- Empirical equations that predicts the sequence length of certain 2^{nd} order sigma-delta modulators.
- A description of FM modulation due to the VCO including spur amplitude equations assuming Narrowband FM modulation.

9 Appendix A – Proposed Models Test Results

Figure 64 shows an example of how the results are presented. The x-axis is always ***Frequency*** in Hertz. The y-axis contains two important pieces of information. The left y-axis shows the ***Amplitude*** in dB of the spurs from the behavioral and proposed models. The right y-axis shows the ***Prediction Error*** in dB. This is simply the difference between the two models predictions. A positive prediction error indicates the proposed model overestimates the spur at a given frequency. A negative prediction error indicates the proposed model underestimates the spur at a given frequency.

Figure 64 begins at DC implying that this test is run at the input of the VCO. This DC component is not a spur. It is related to the average voltage at the VCO input and can be ignored. The rest of the frequency components are the spurs we are interested in.

Note the large error at 10 MHz. This frequency corresponds to the first reference frequency harmonic for this example. As discussed earlier the proposed model greatly underestimates reference spurs.

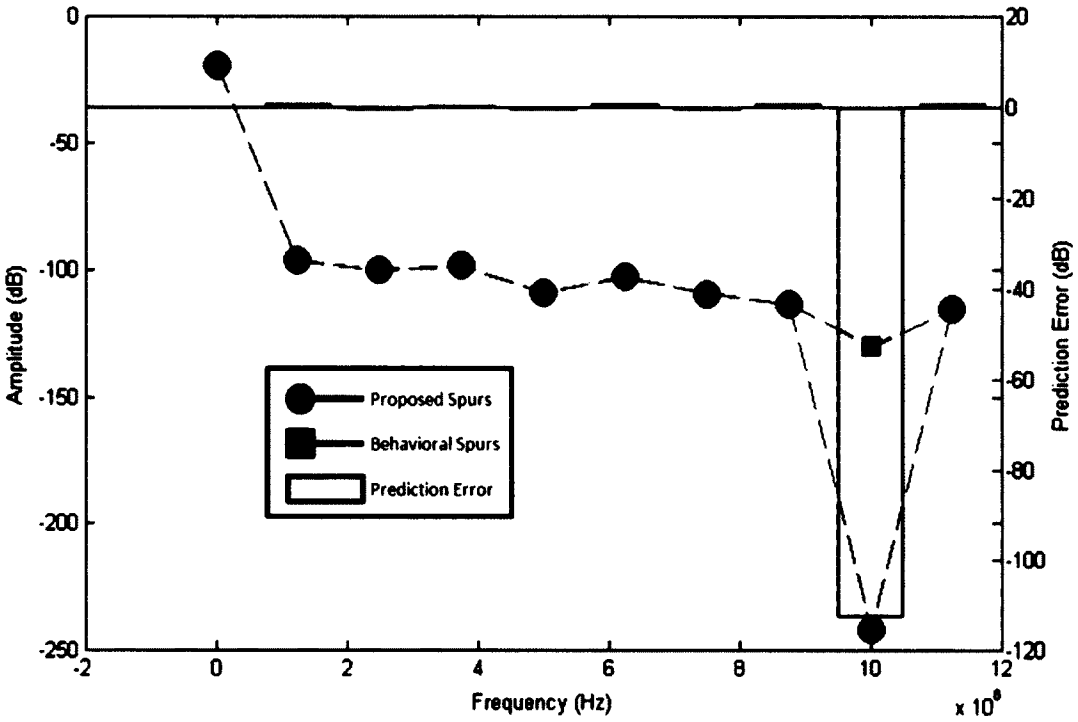


Figure 64 - Example Results from Before the VCO.

Figure 65 shows an example of results from the output of the VCO. The dominant frequency component is not a spur, but the output frequency that is expected from the PLL. All other frequency components centered around the output frequency are spurs.

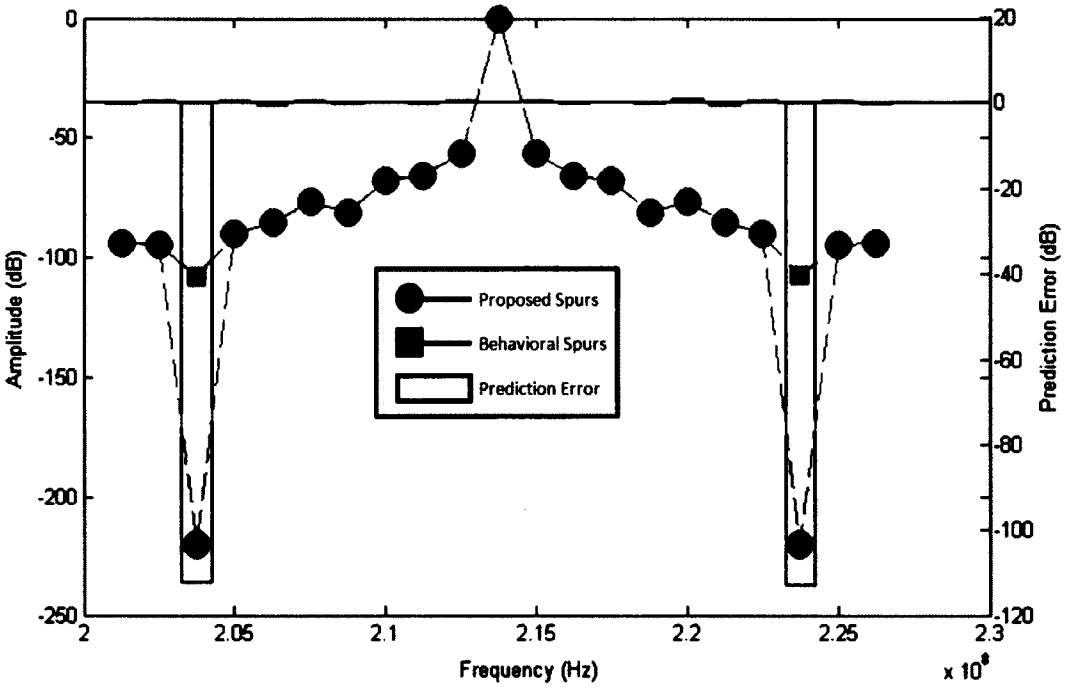
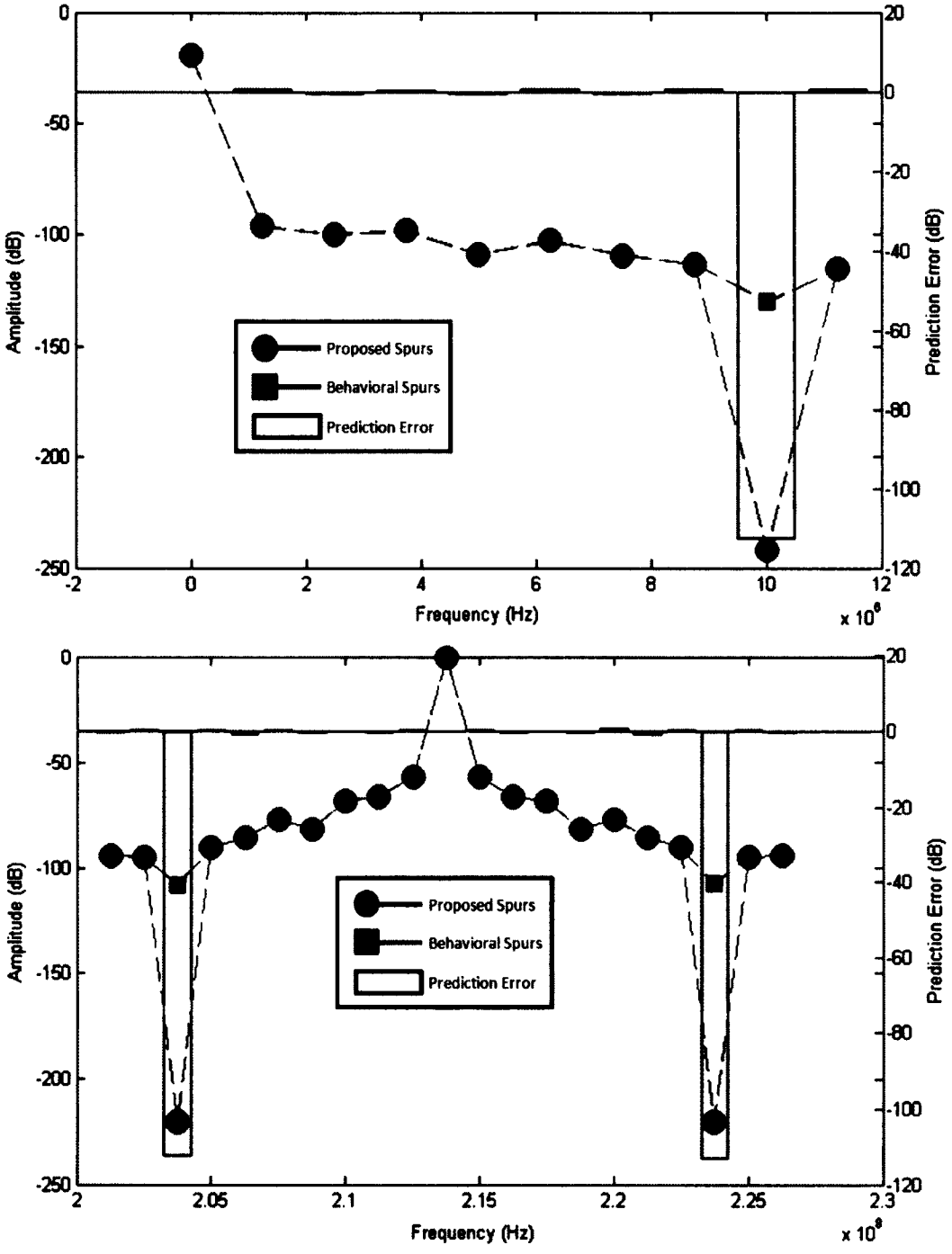


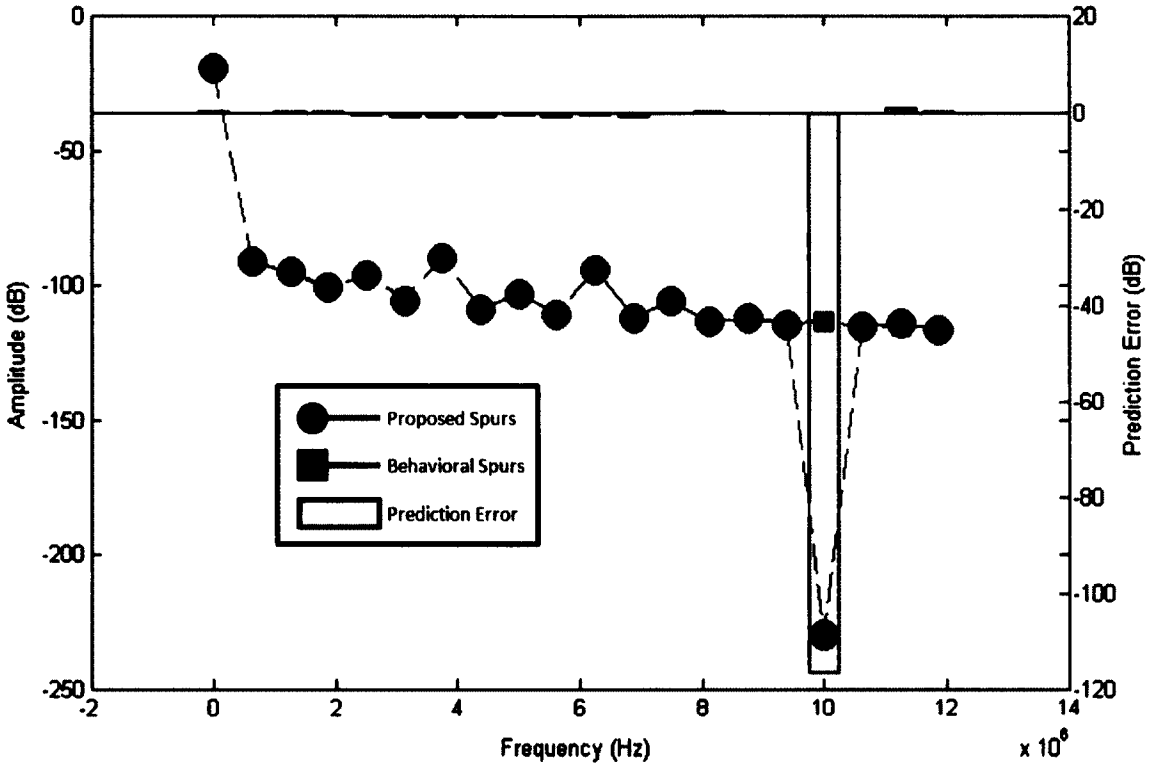
Figure 65 - Example Results from After the VCO.

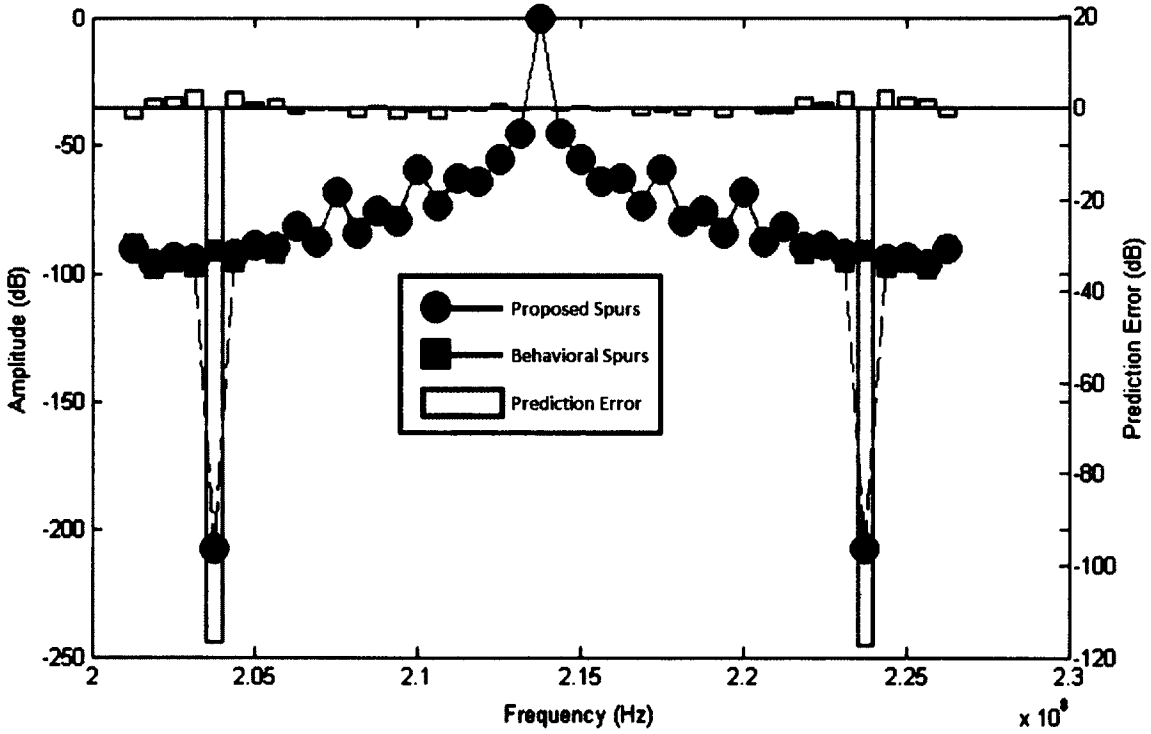
The test results begin on the following page.

9.1.1 *Test Group A – Sweep SDM Order – 1st Order*

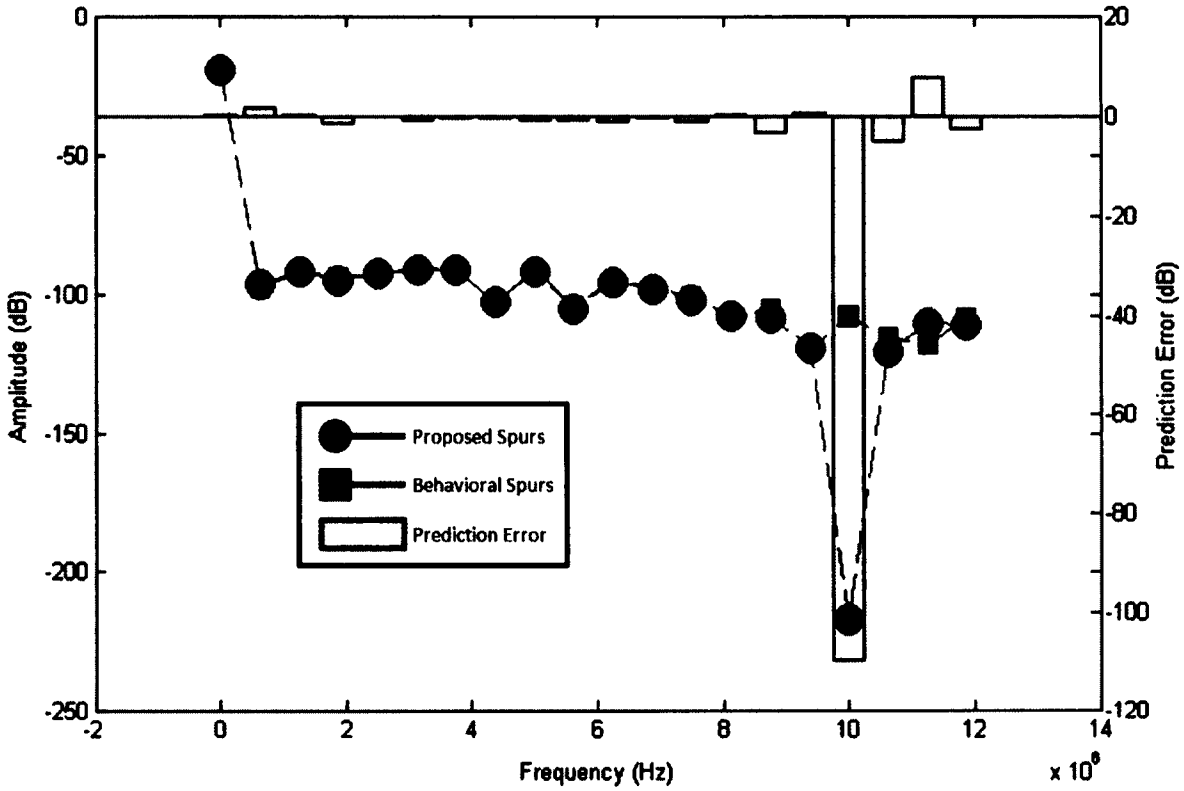


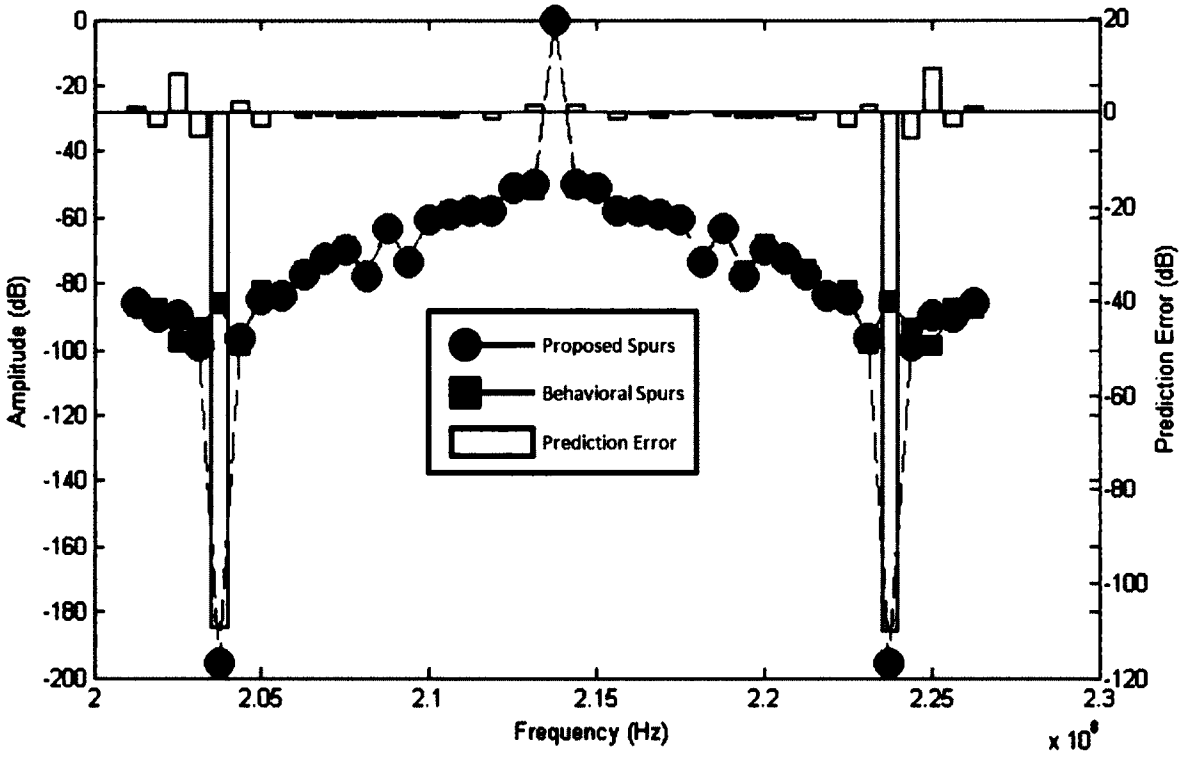
9.1.2 *Test Group A – Sweep SDM Order – 2nd Order*



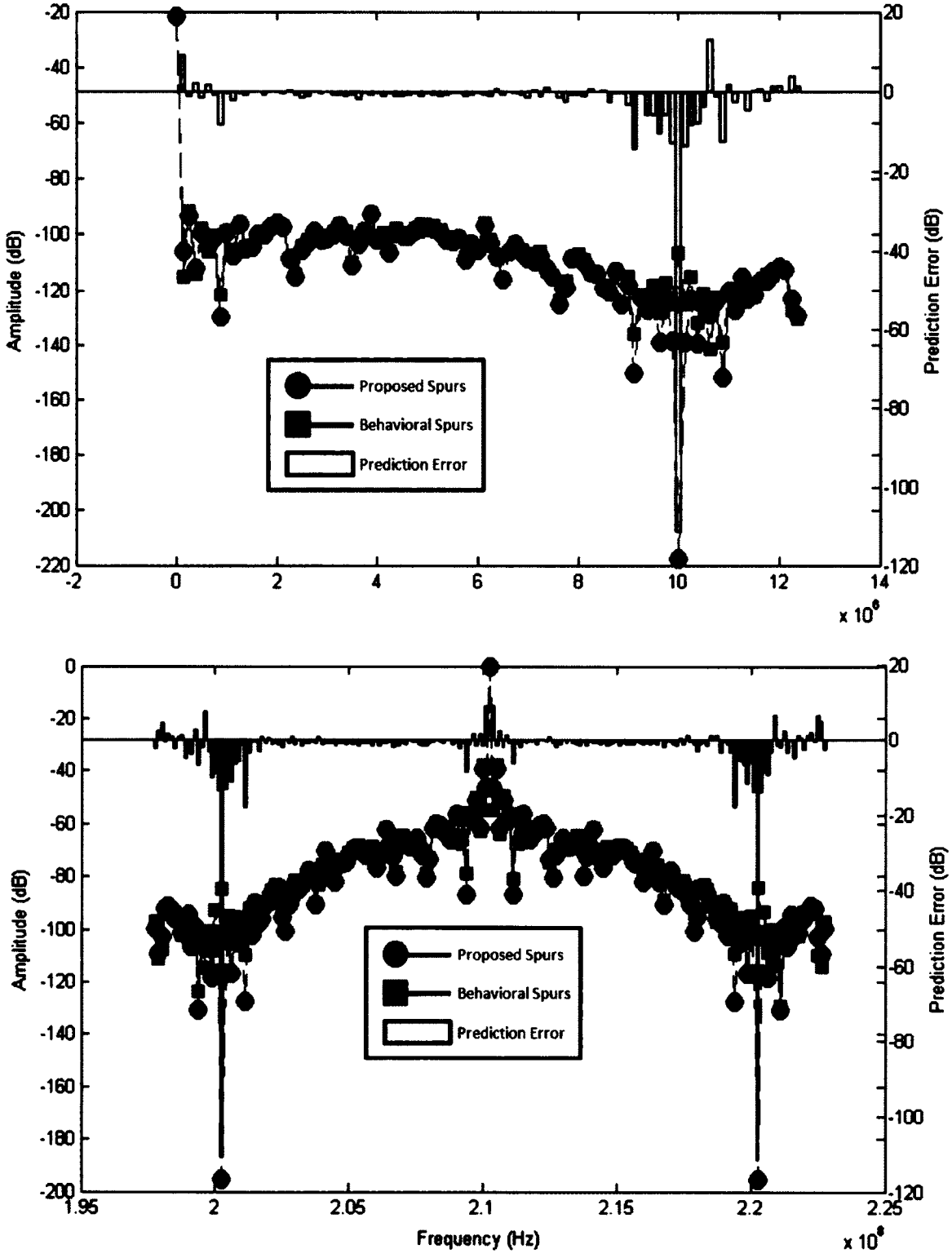


9.1.3 *Test Group A – Sweep SDM Order – 3rd Order*

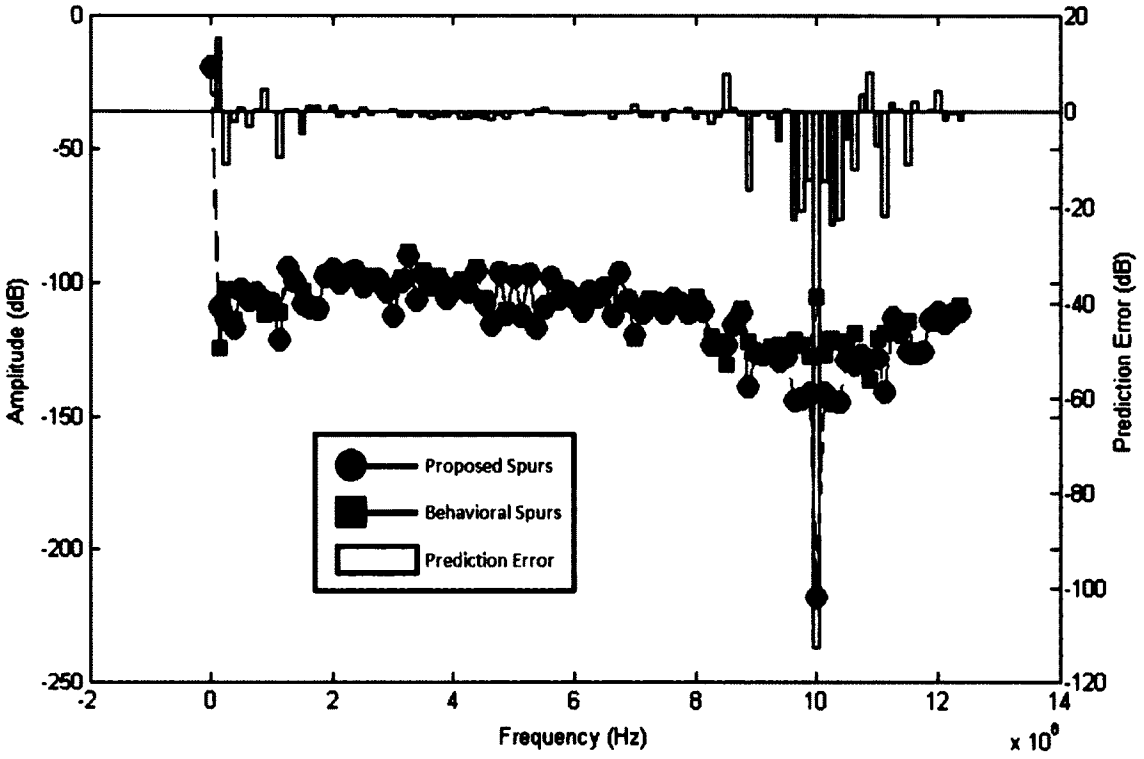


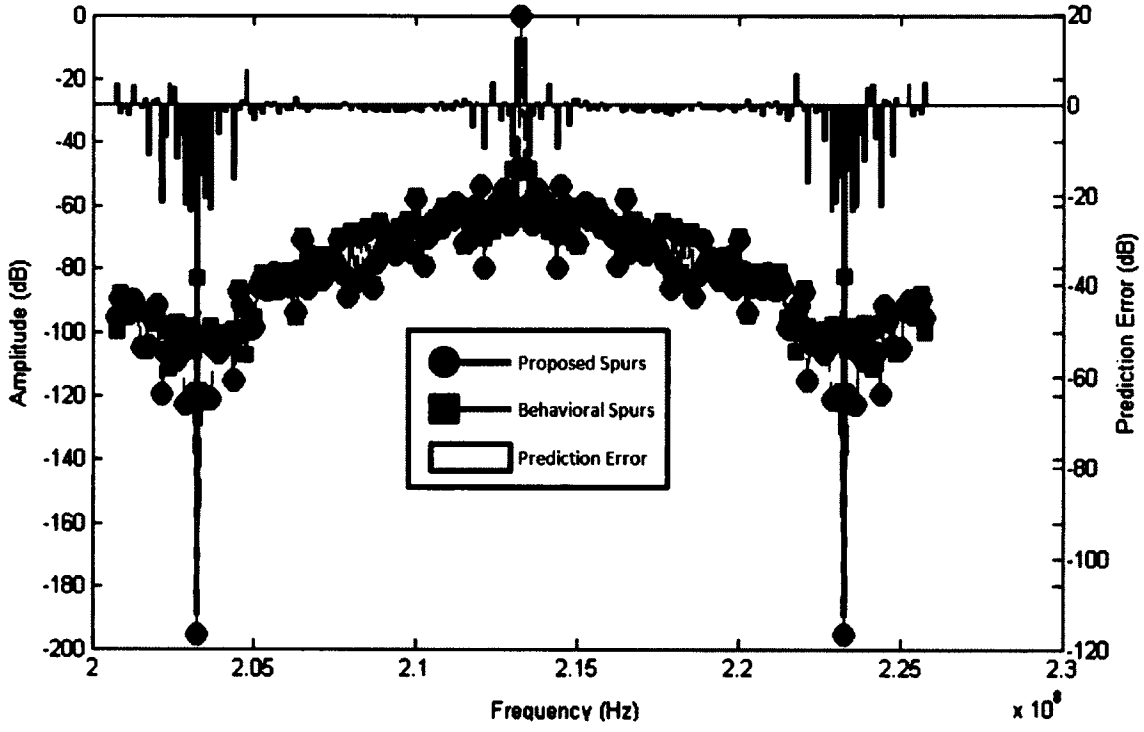


9.1.4 *Test Group B – Sweep X/M – 1/40*

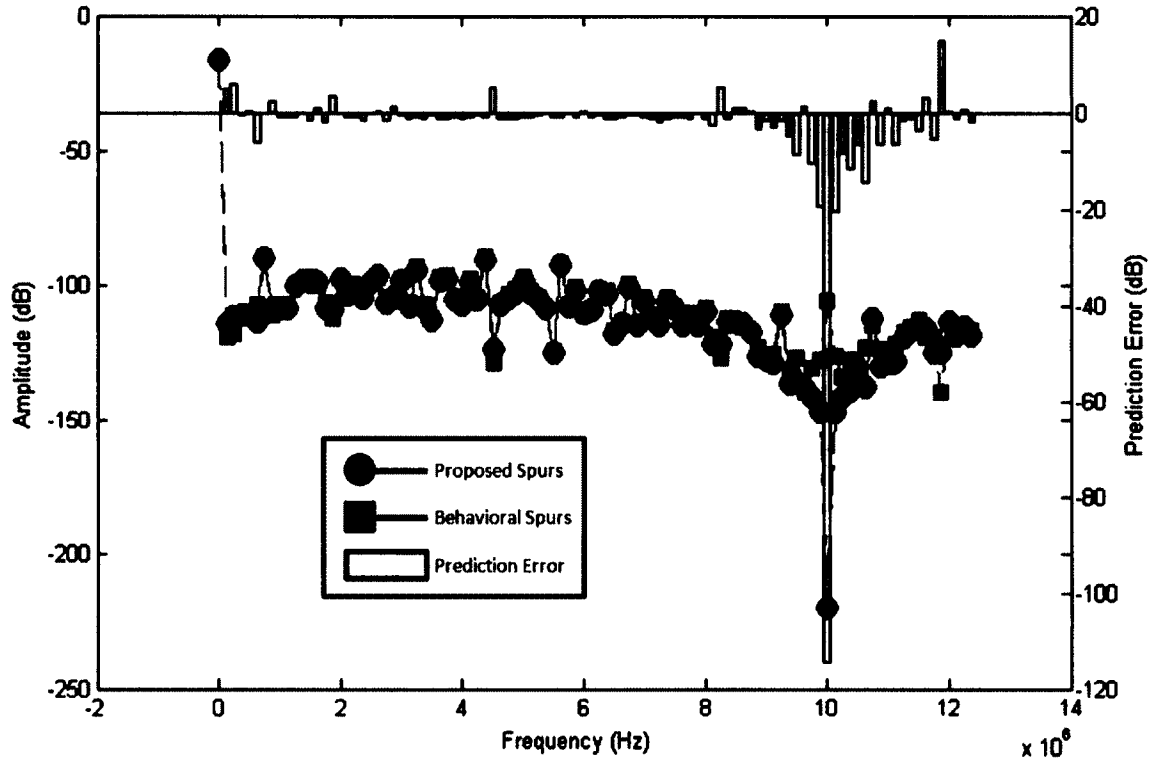


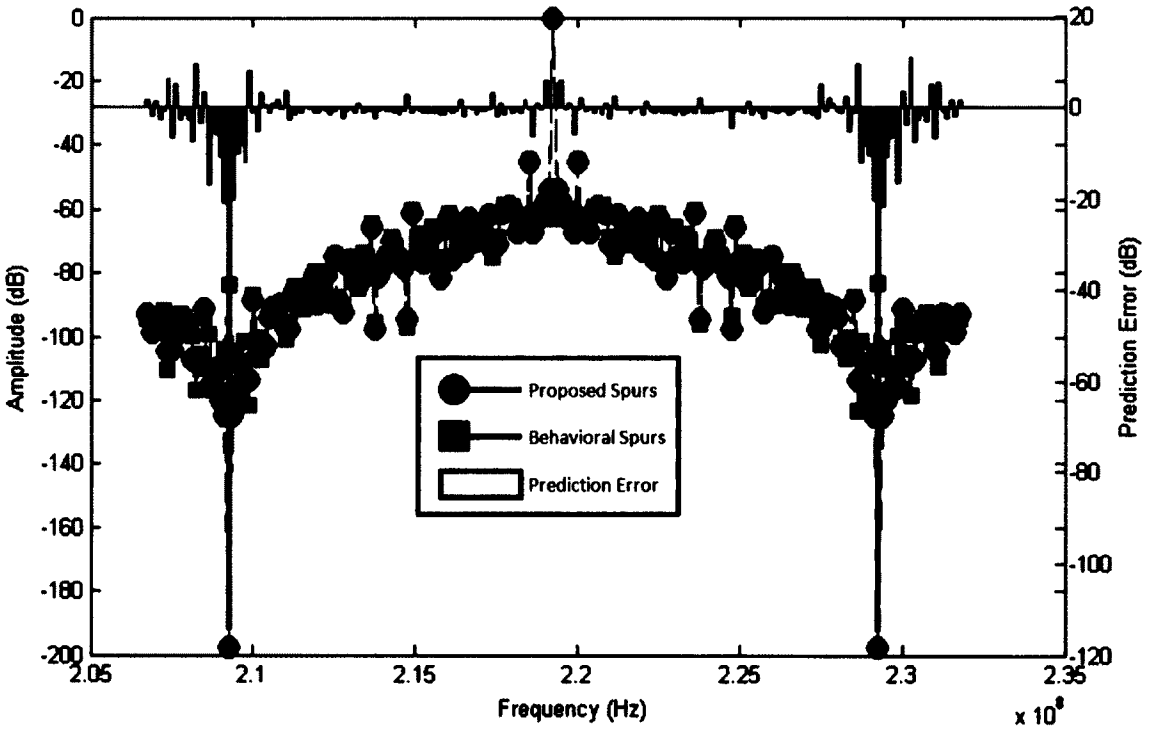
9.1.5 Test Group B – Sweep X/M – 13/40



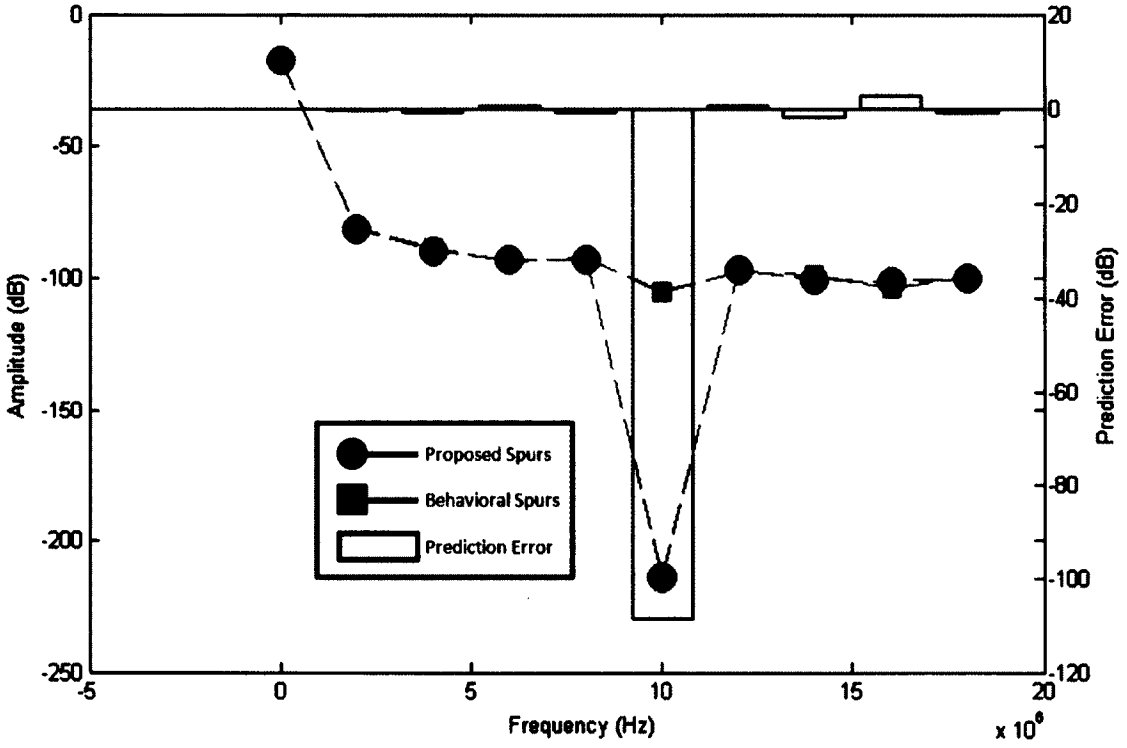


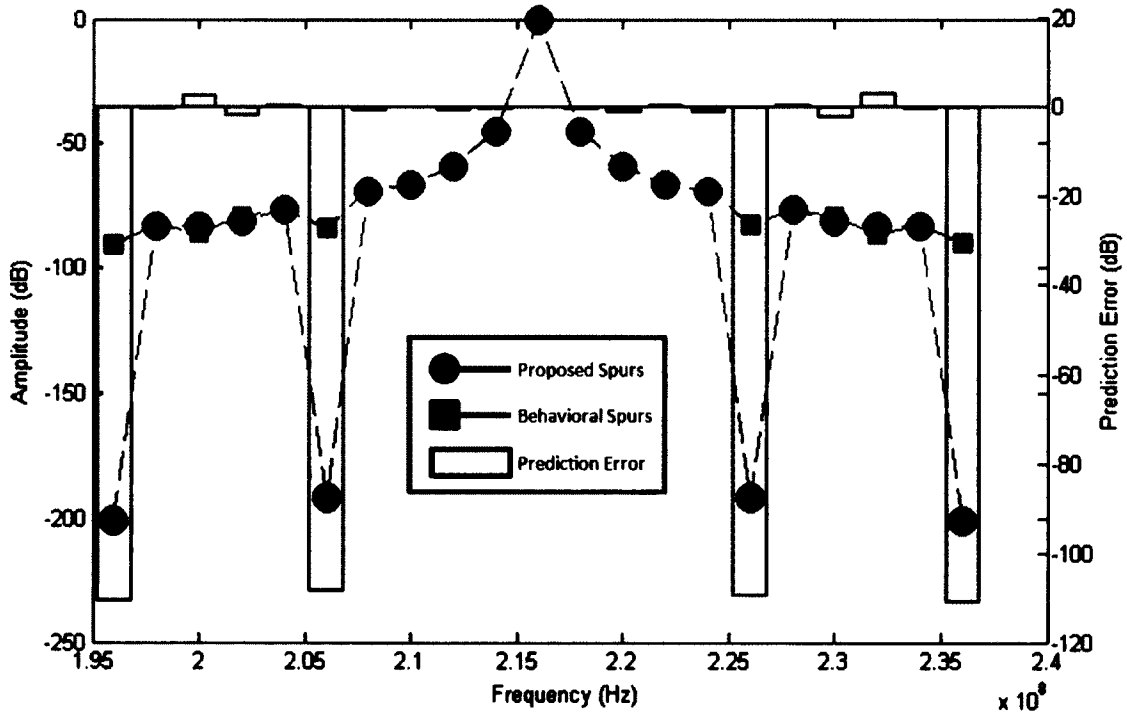
9.1.6 *Test Group B – Sweep X/M – 13/40*



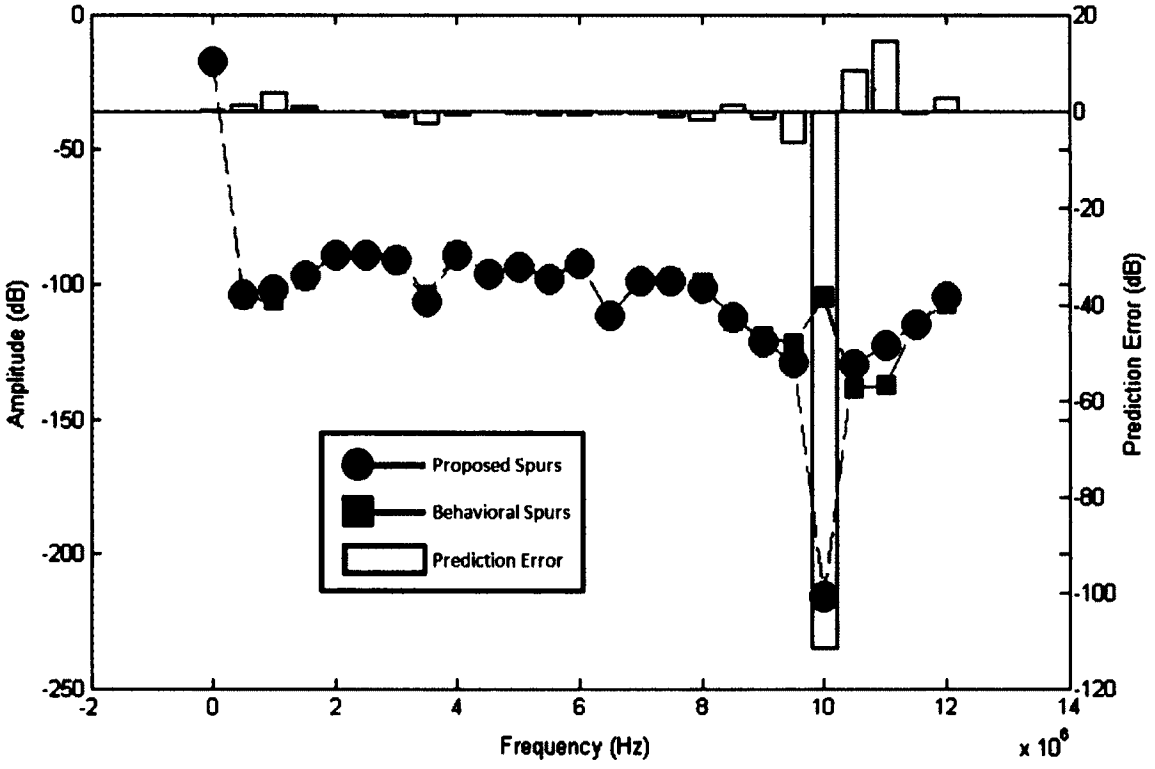


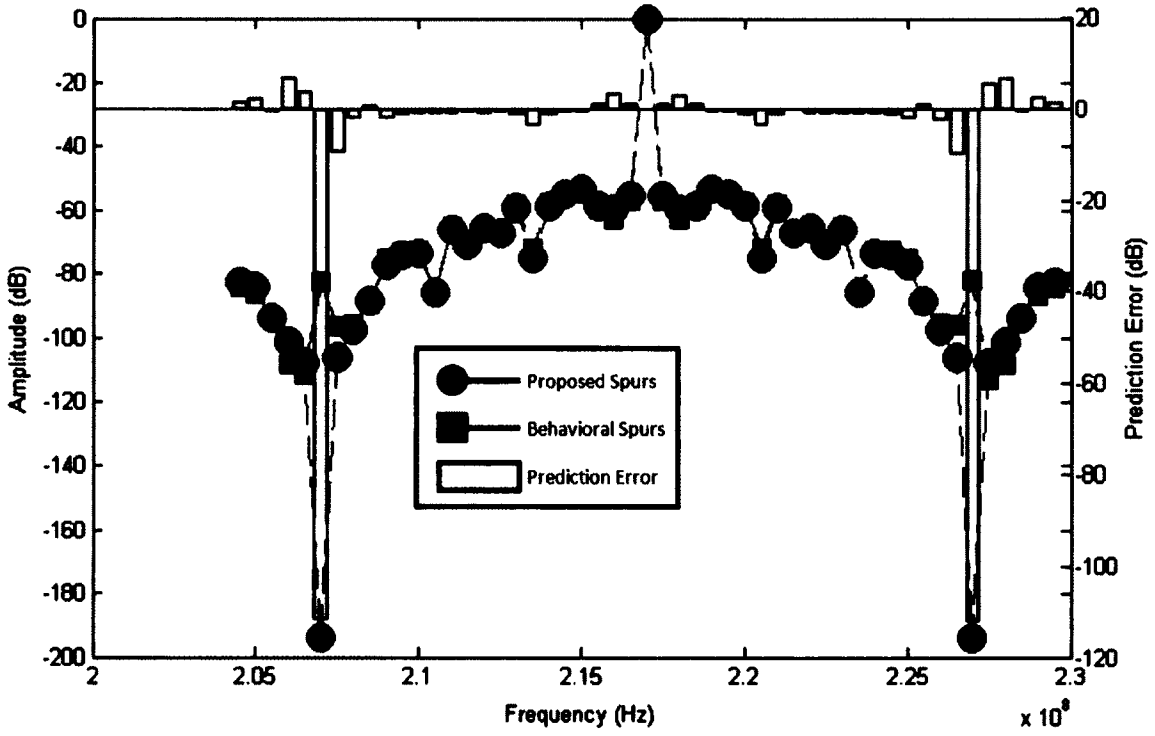
9.1.7 Test Group C – Sweep X/M – 3/5



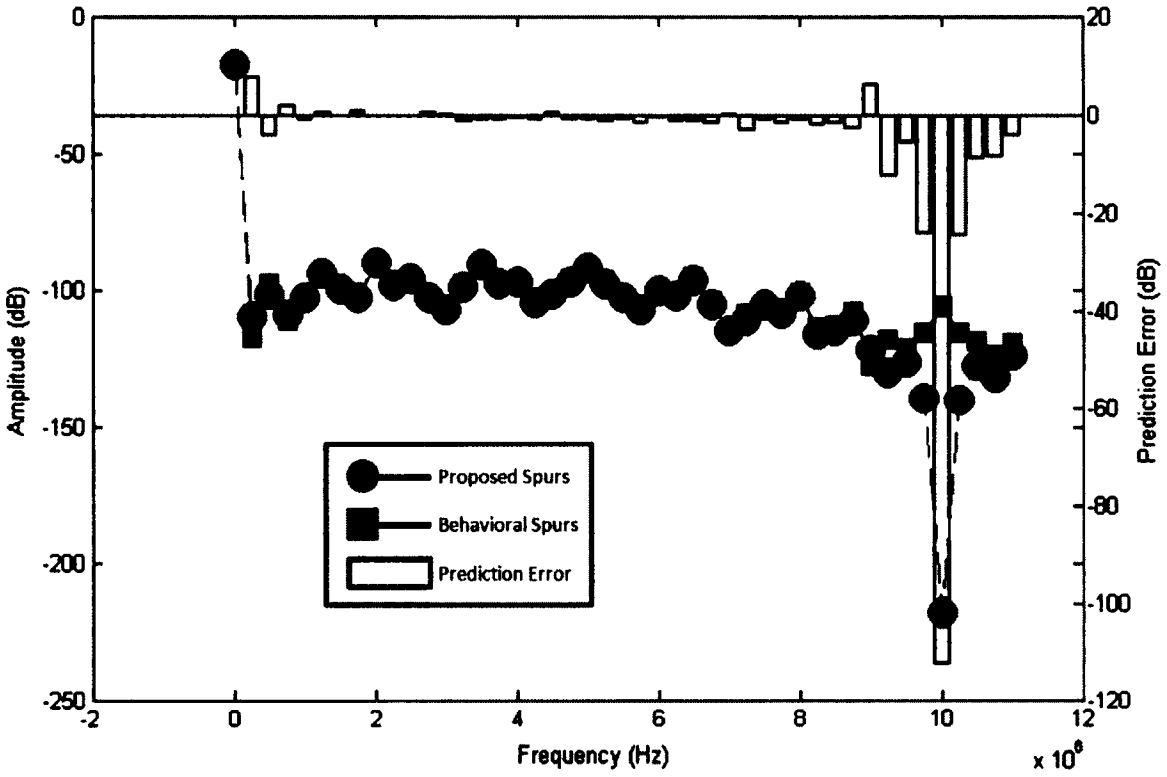


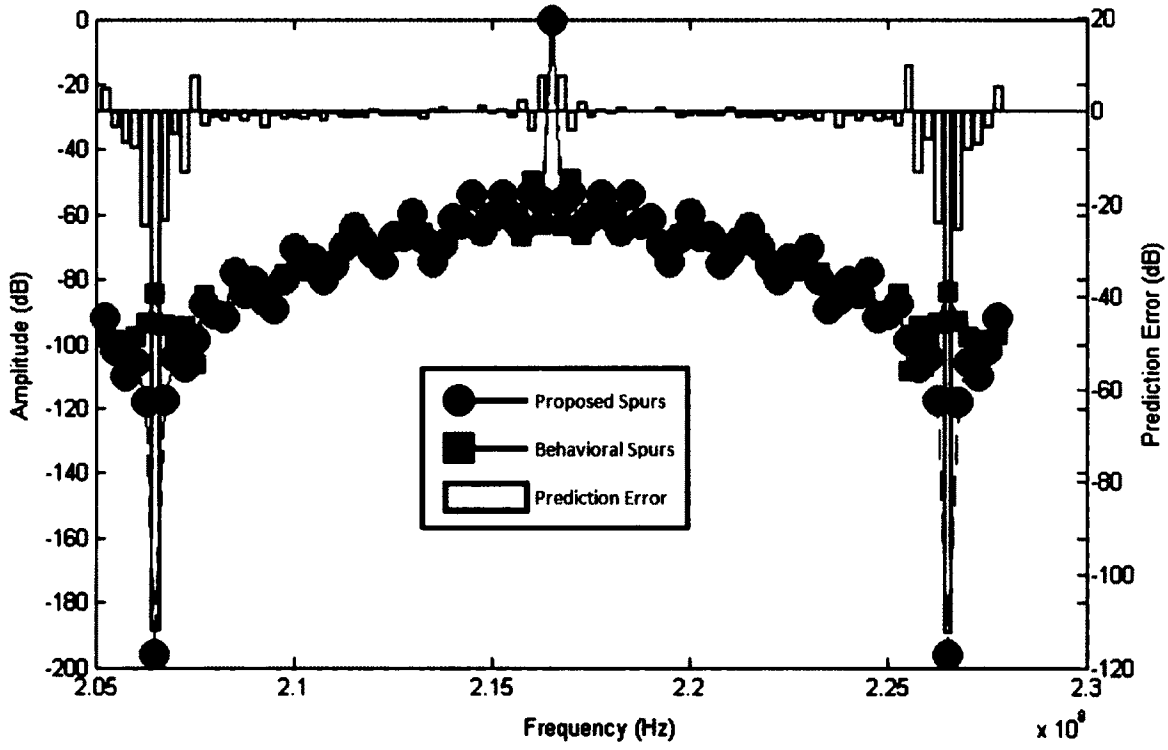
9.1.8 Test Group C – Sweep X/M – 7/10



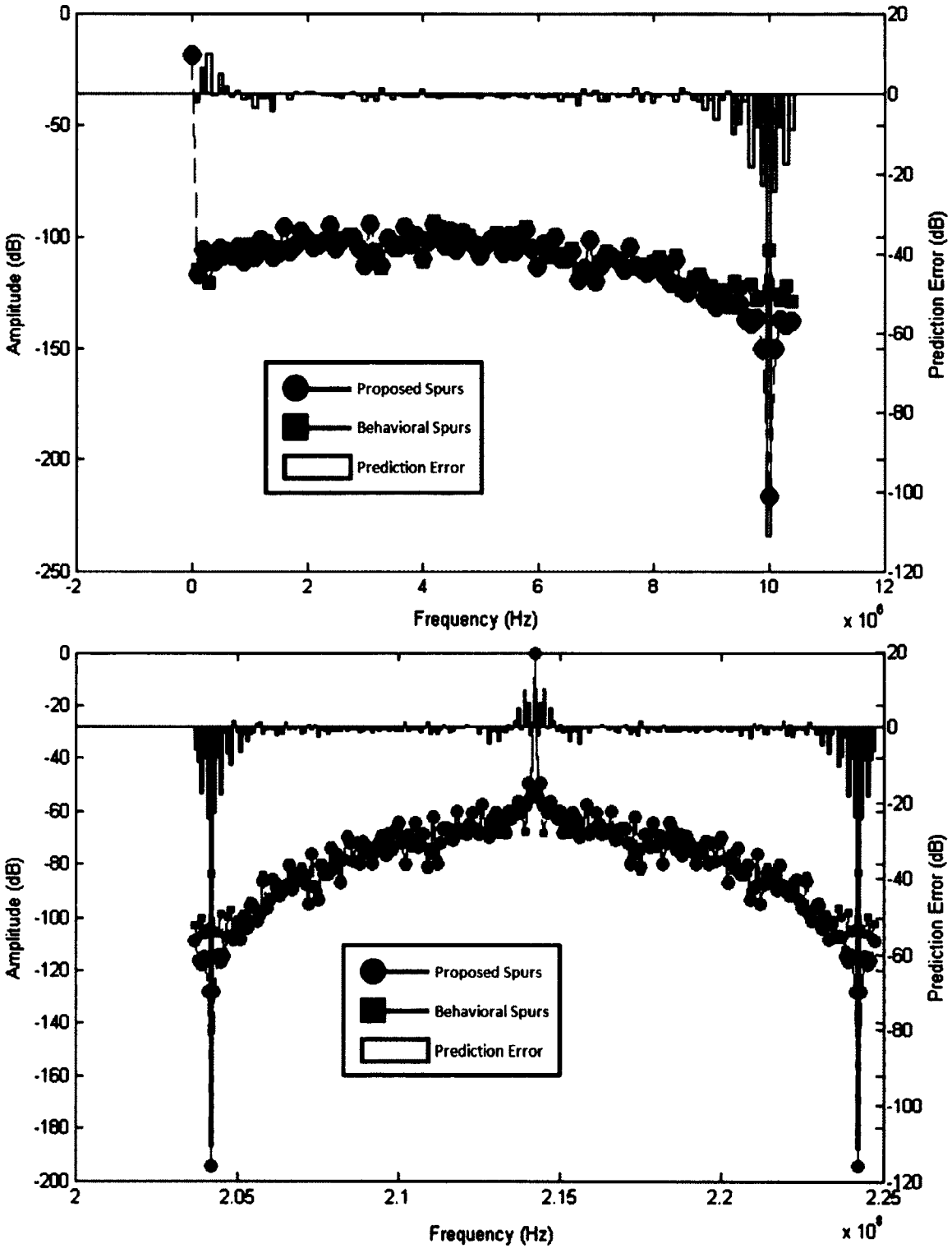


9.1.9 Test Group C – Sweep X/M – 13/20

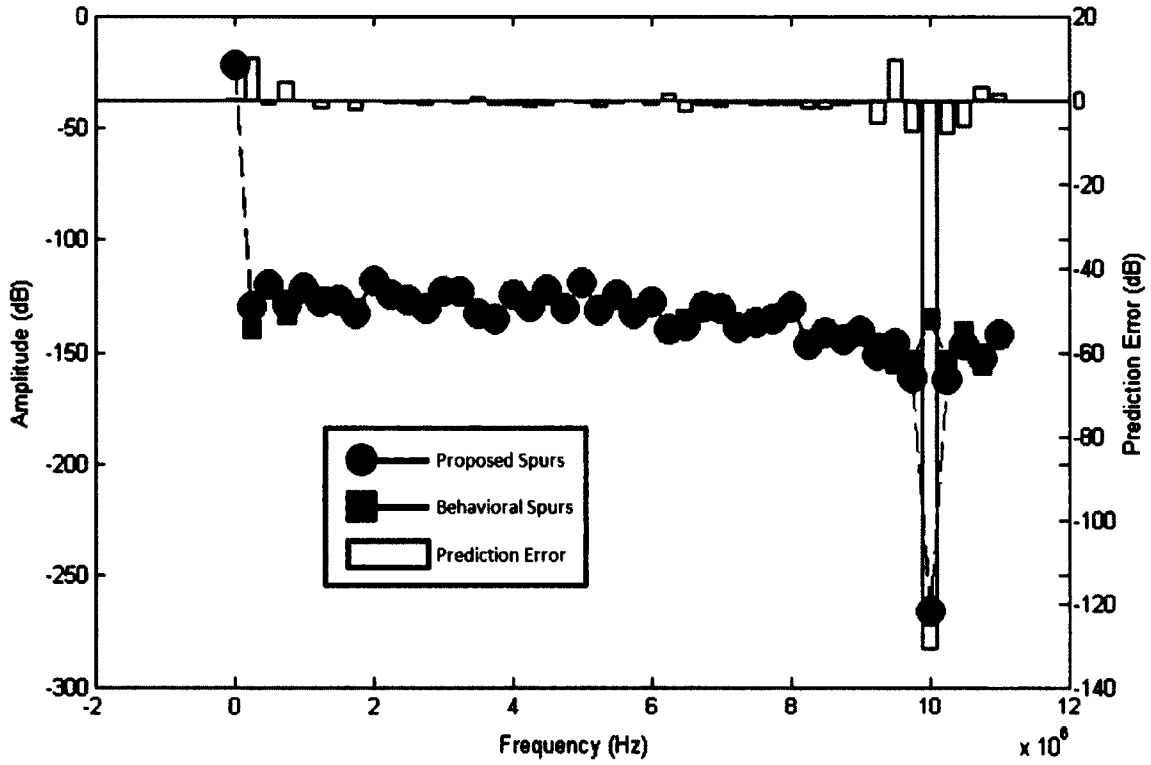


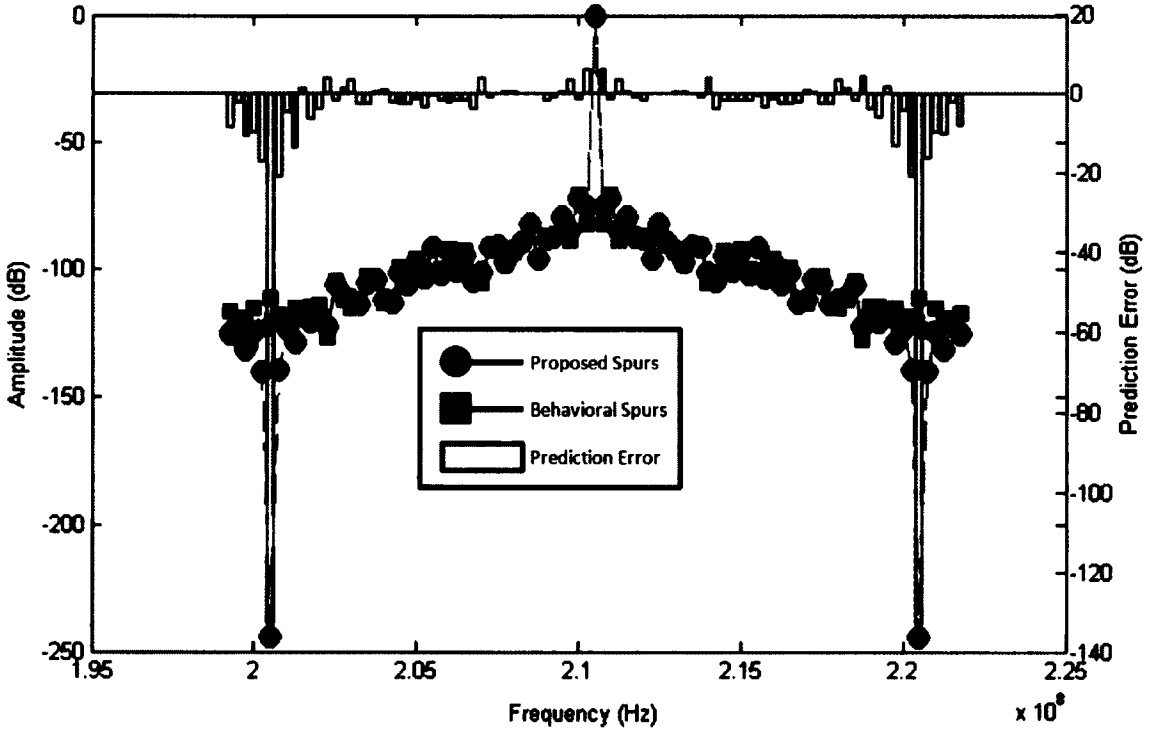


9.1.10 Test Group C – Sweep X/M – 21/50

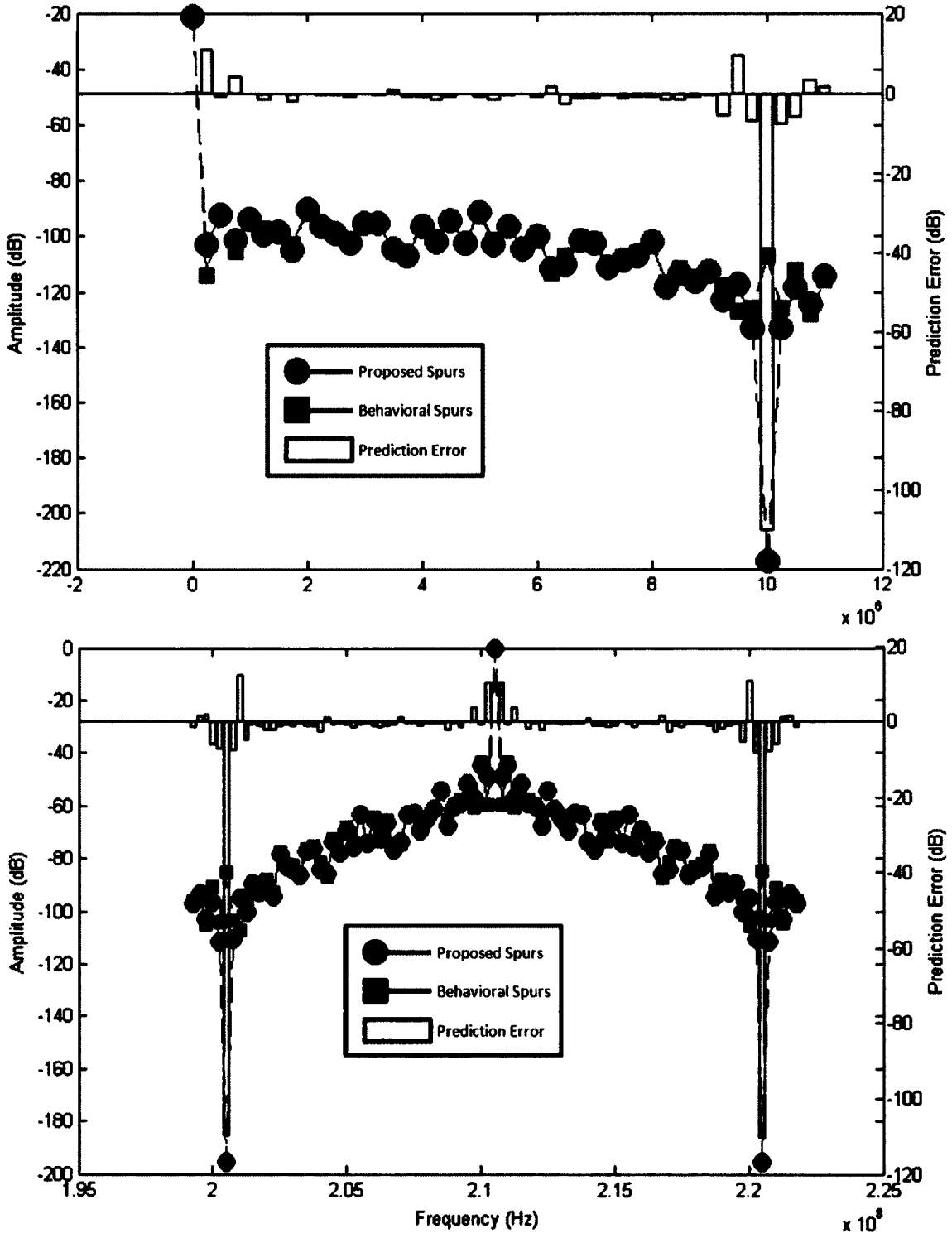


9.1.11 Test Group D – Sweep Bandwidth – 10 kHz

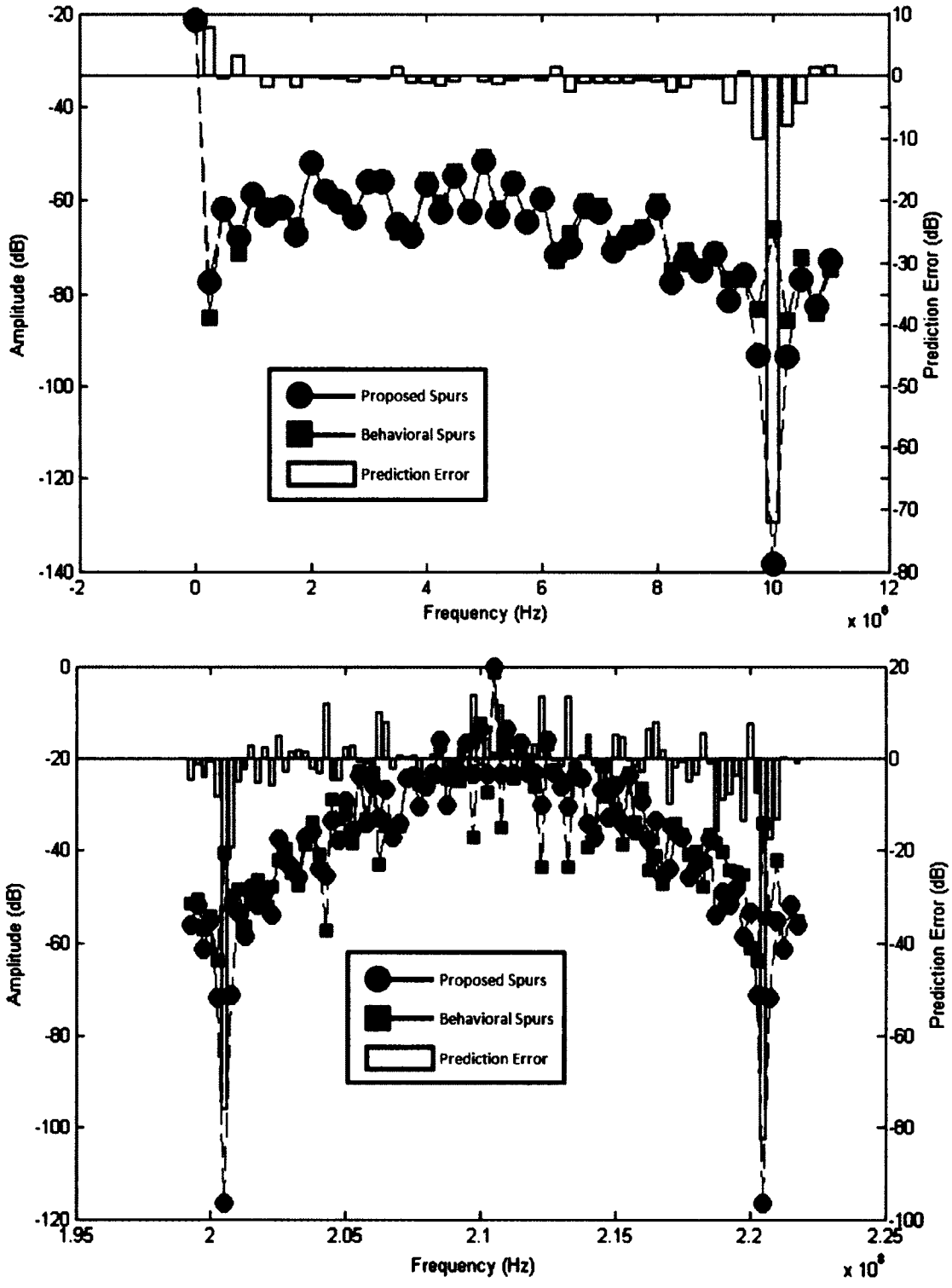




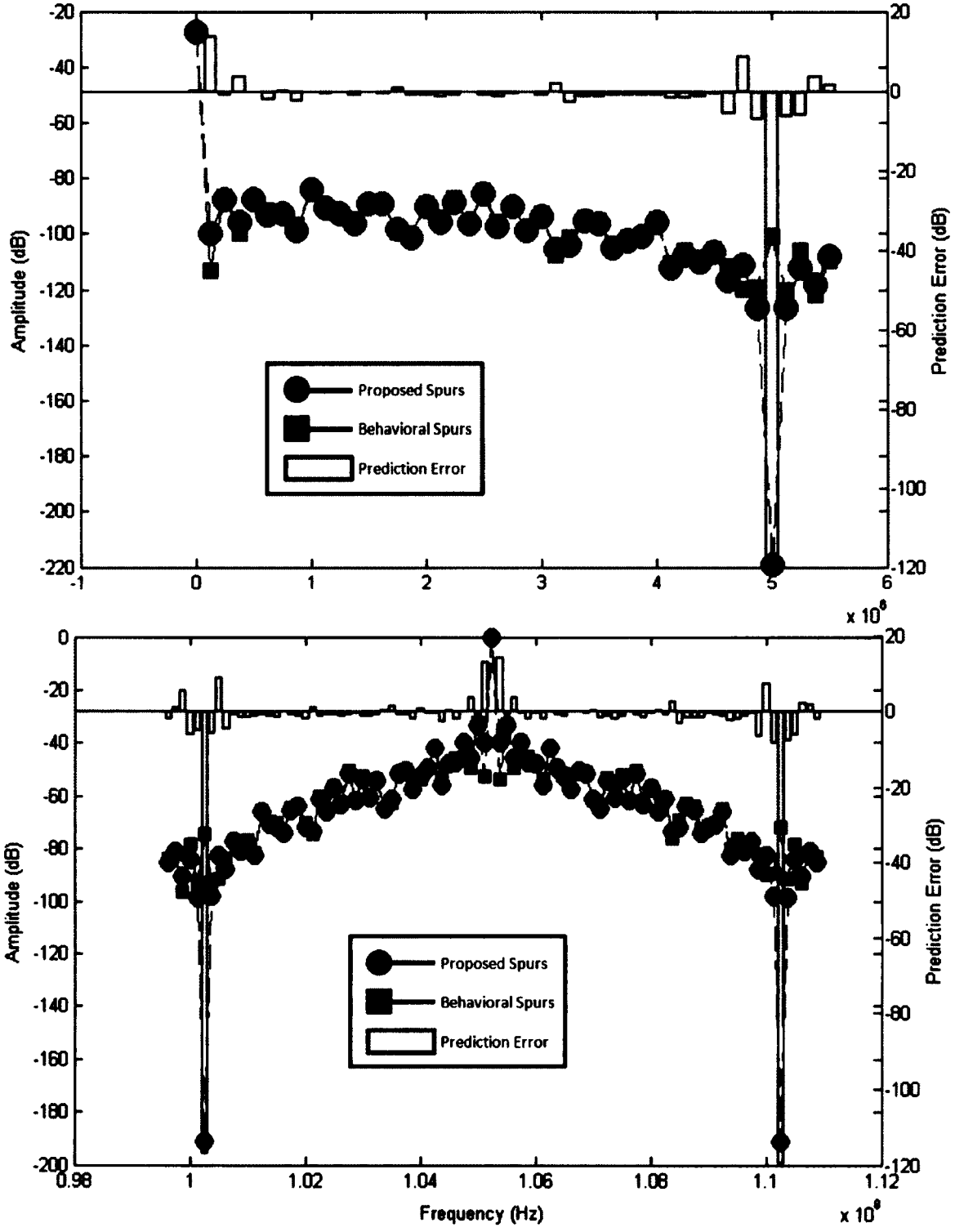
9.1.12 Test Group D – Sweep Bandwidth – 50 kHz



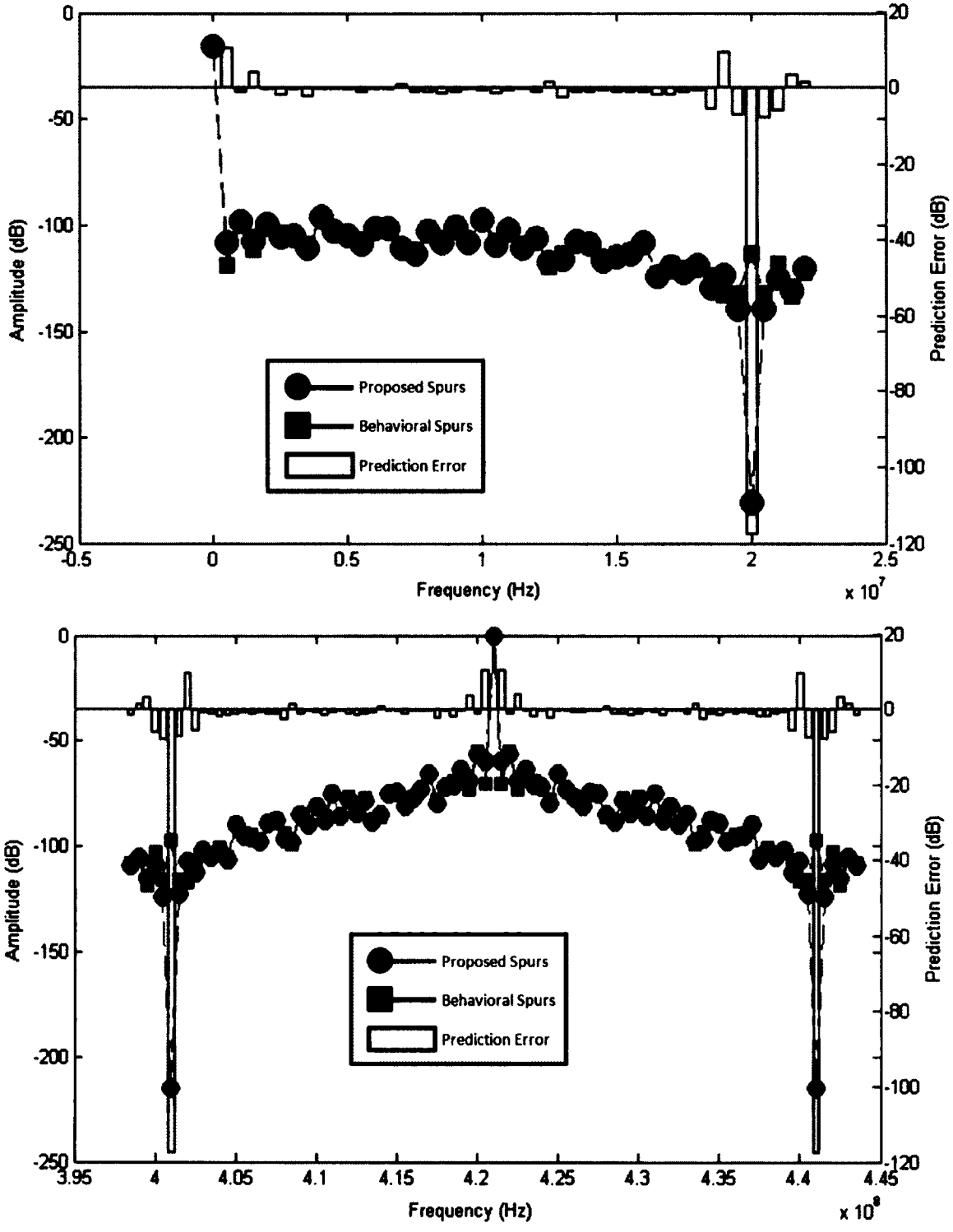
9.1.13 Test Group D – Sweep Bandwidth – 500 kHz



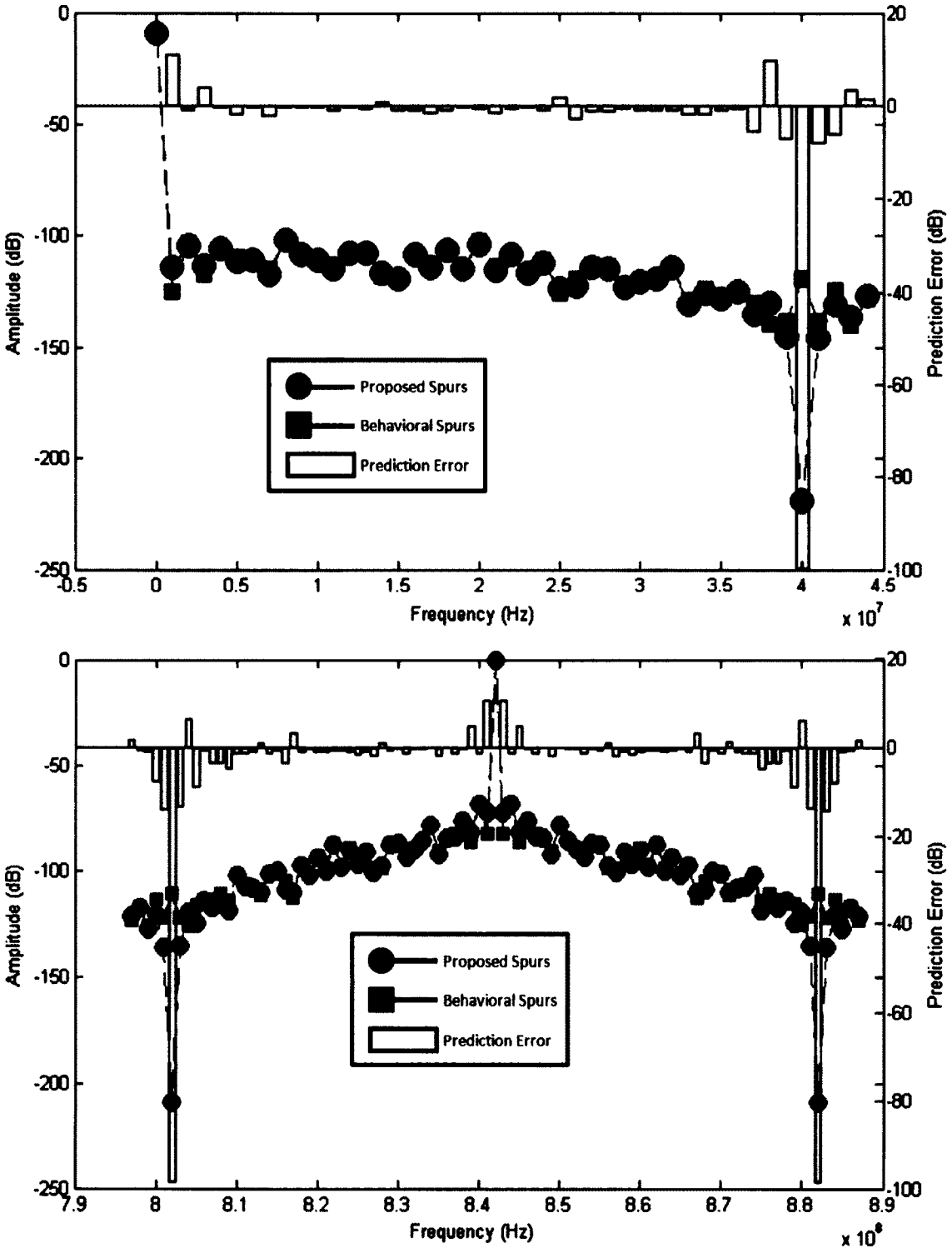
9.1.14 Test Group E – Sweep f_{ref} – 5 MHz



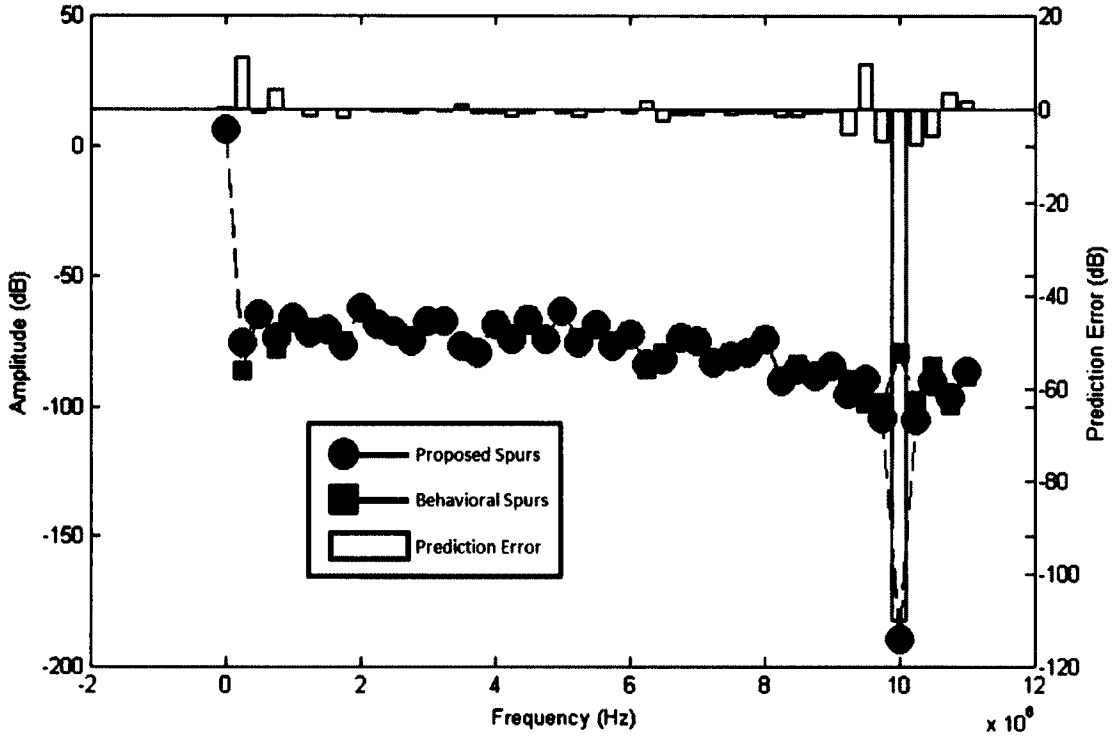
9.1.15 Test Group E – Sweep f_{ref} – 20 MHz

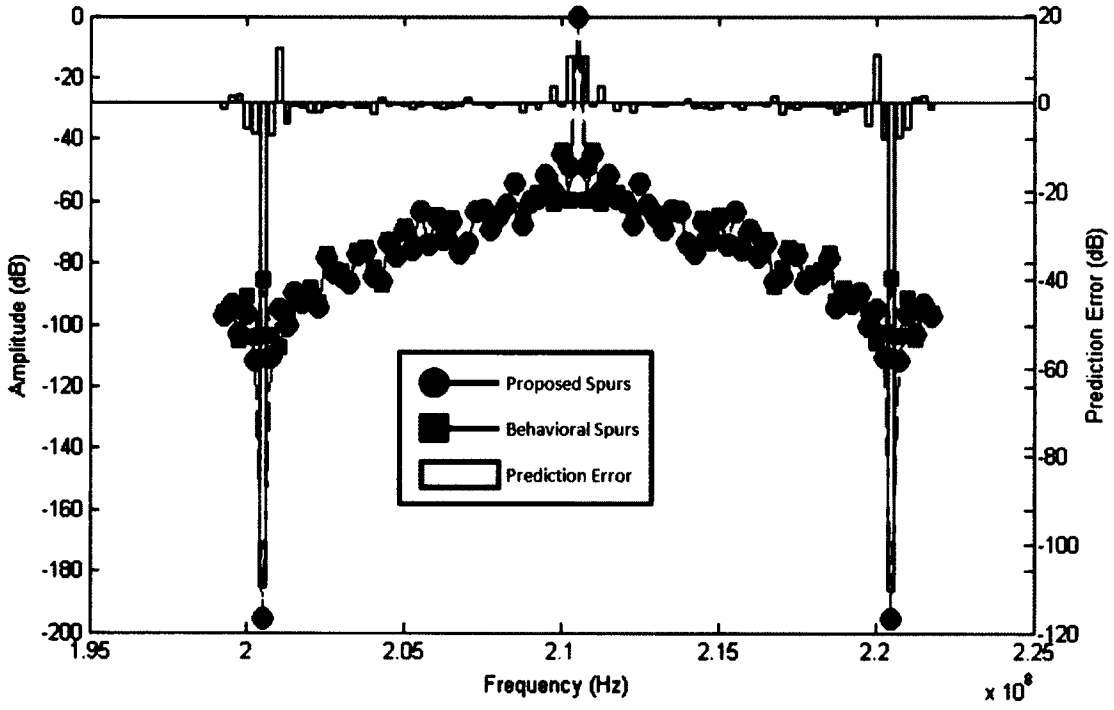


9.1.16 Test Group E – Sweep f_{ref} – 40 MHz

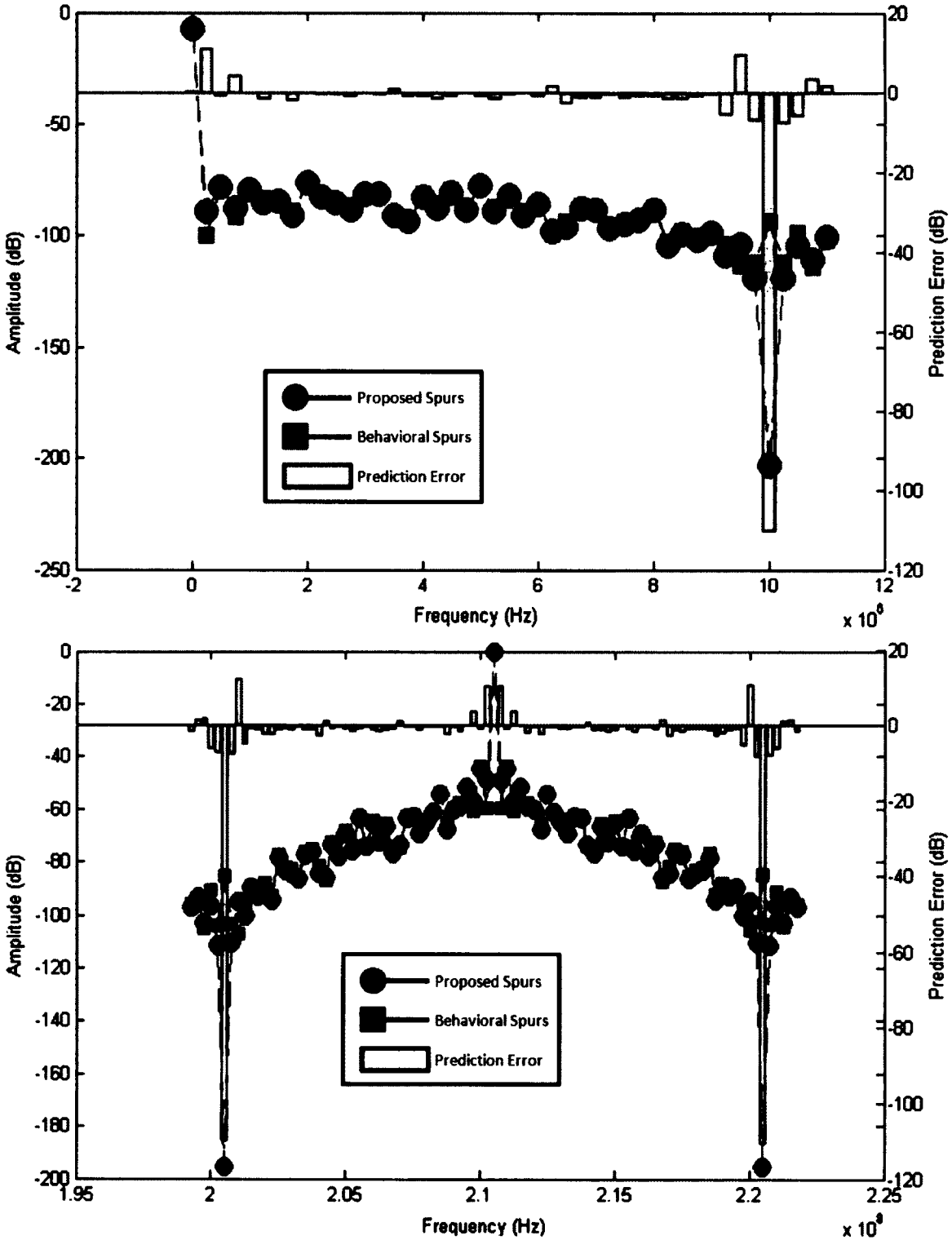


9.1.17 Test Group F – Sweep k_{vco} – 10 MHz

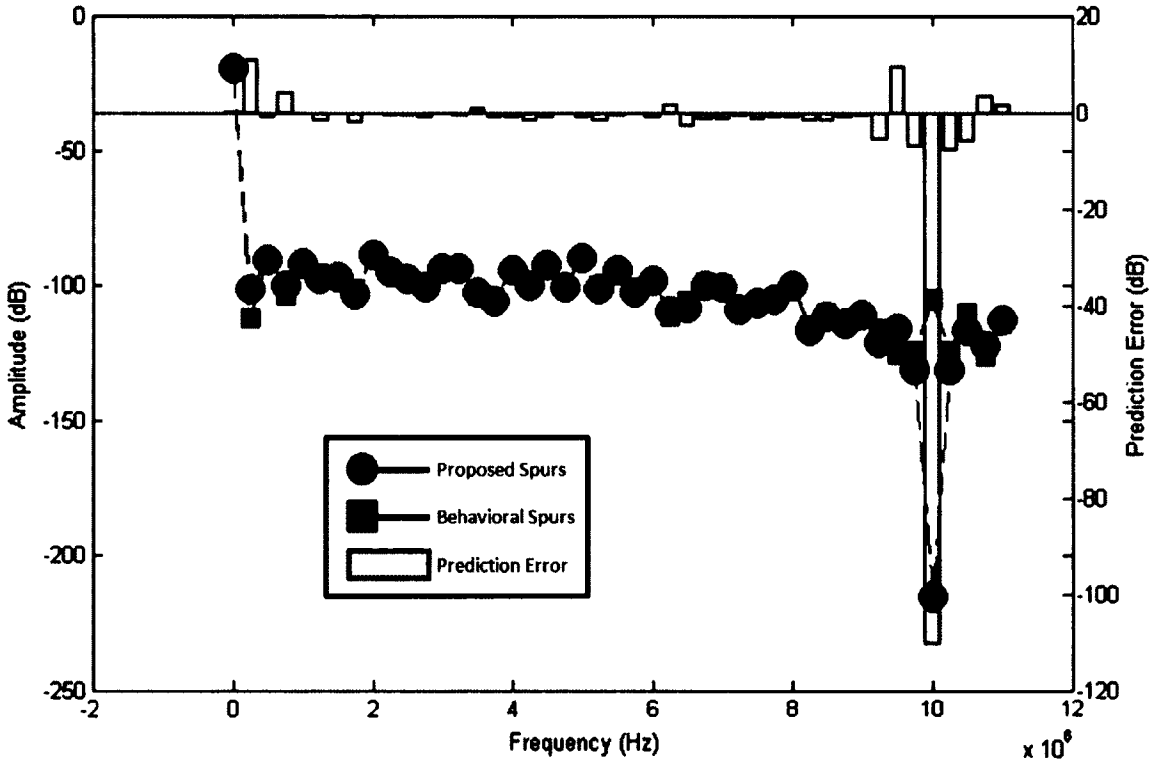


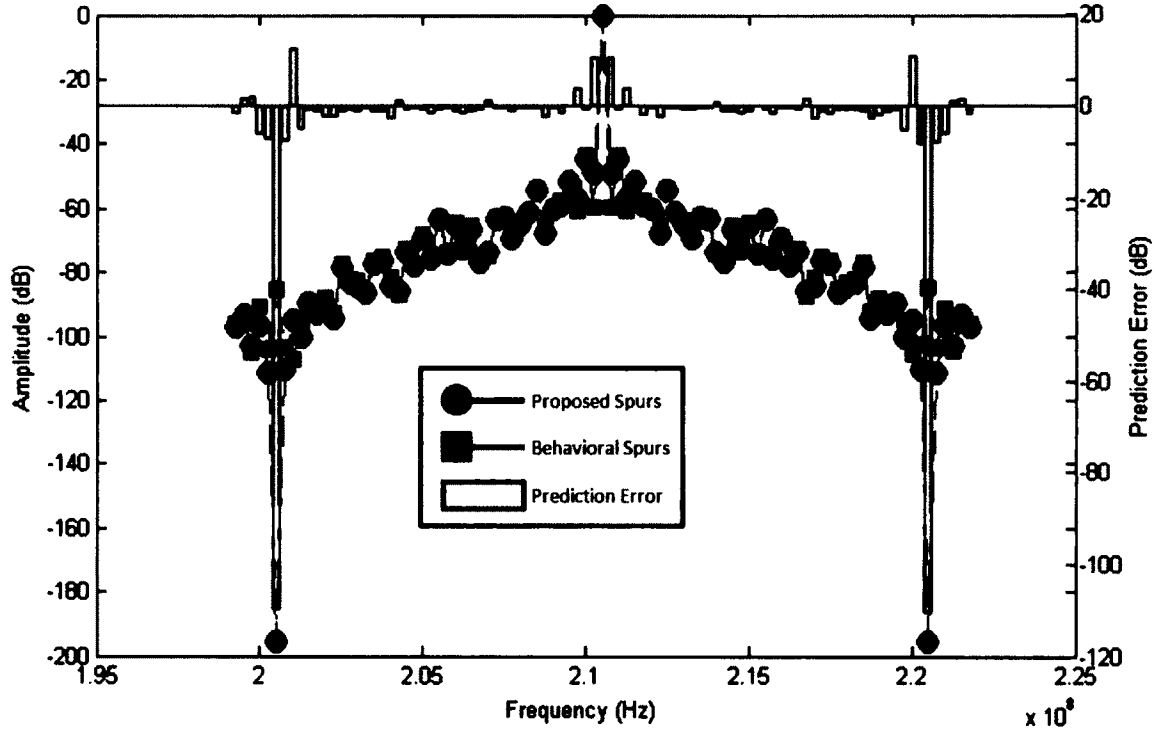


9.1.18 Test Group F – Sweep k_{vco} – 50 MHz

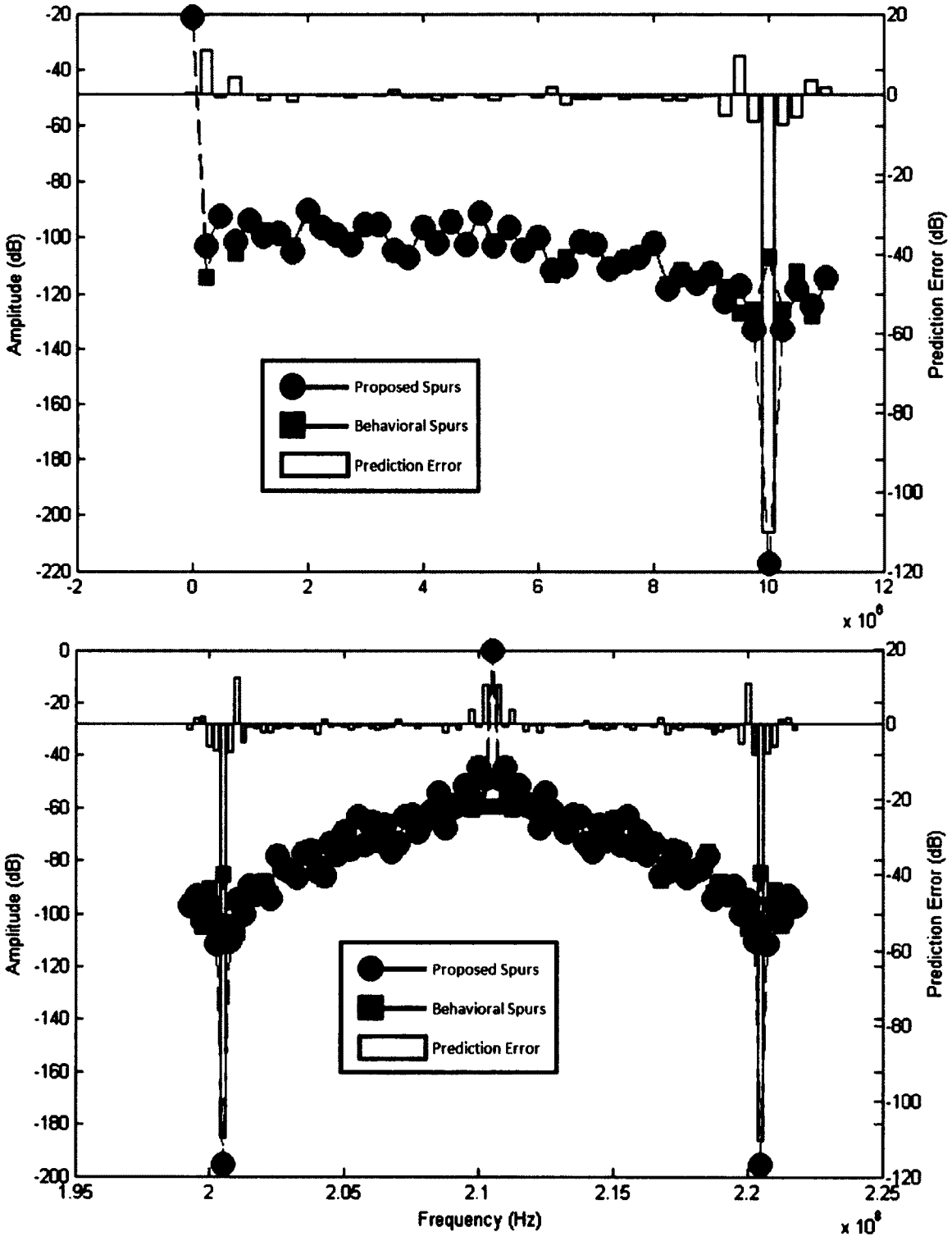


9.1.19 Test Group F – Sweep k_{VCO} – 200 MHz

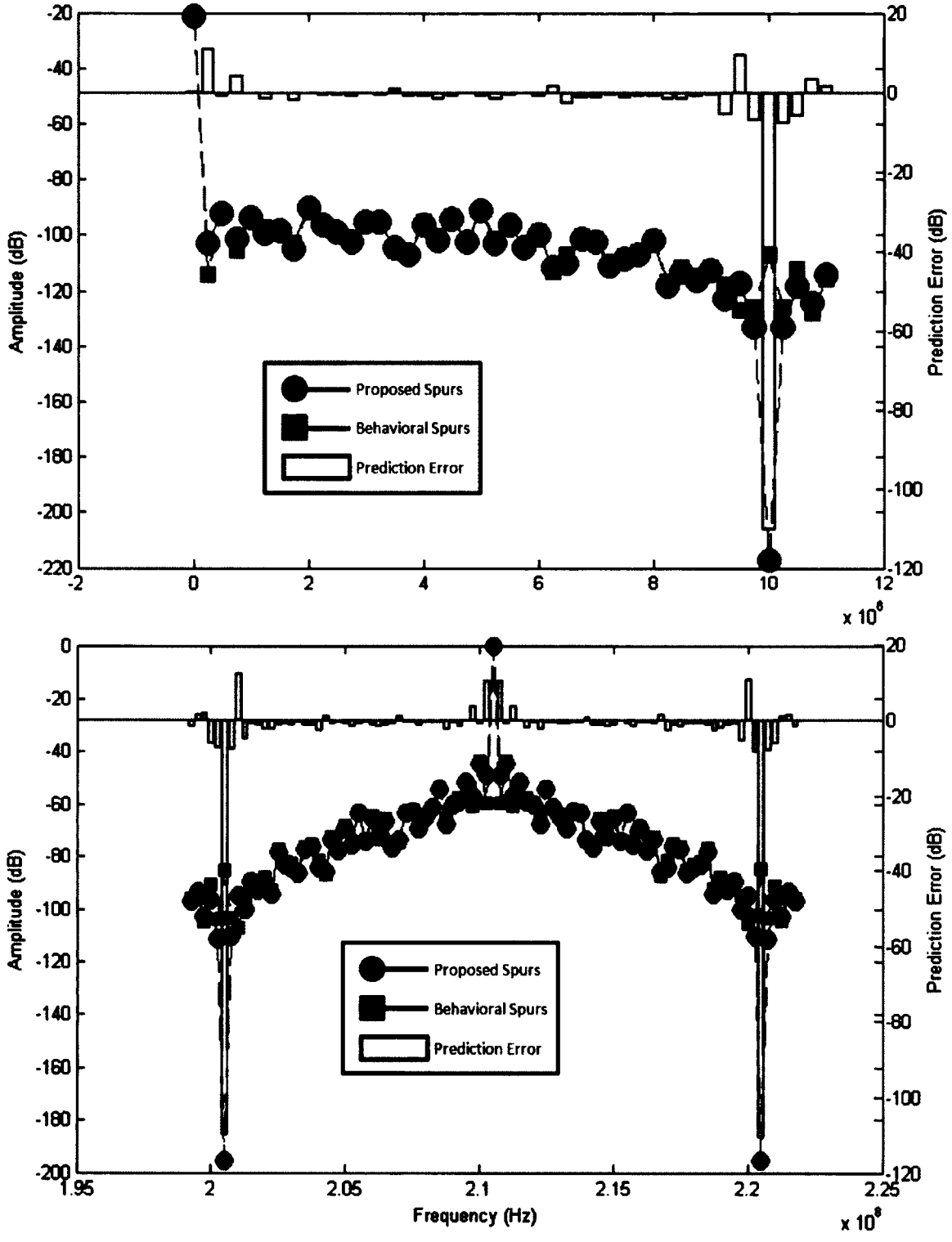




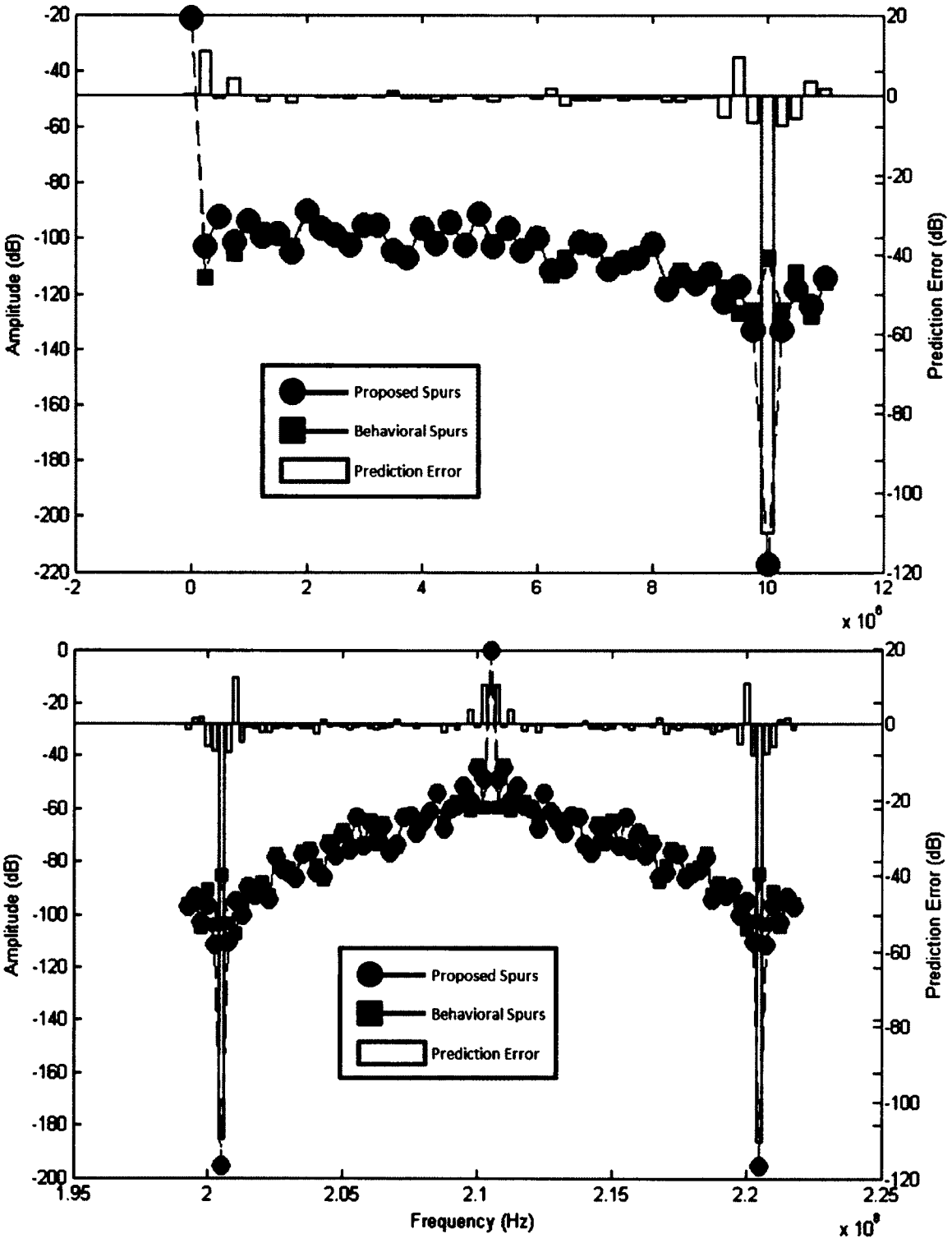
9.1.20 Test Group G – Sweep I_{CP} – 10 uA



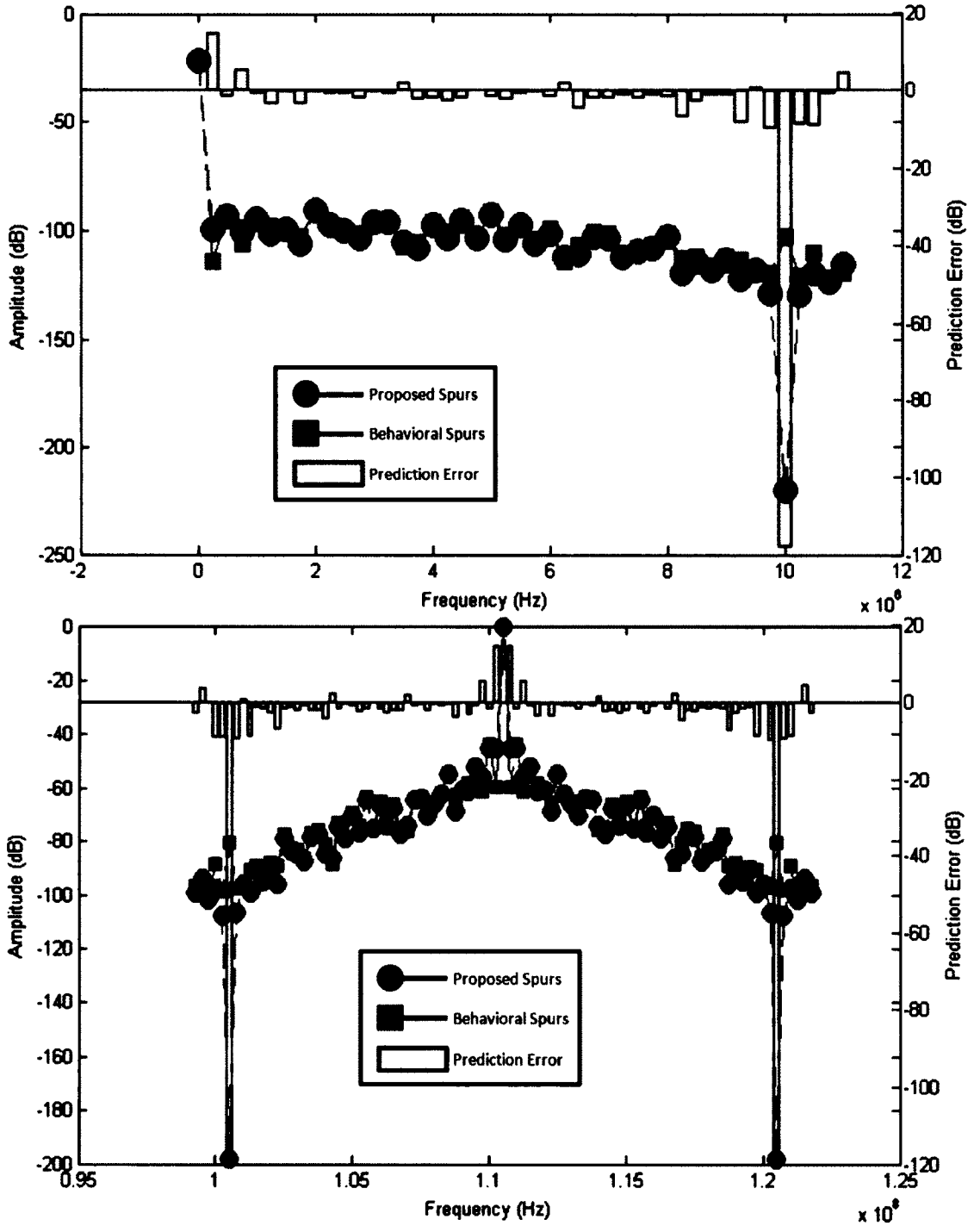
9.1.21 Test Group G – Sweep I_{CP} – 50 uA



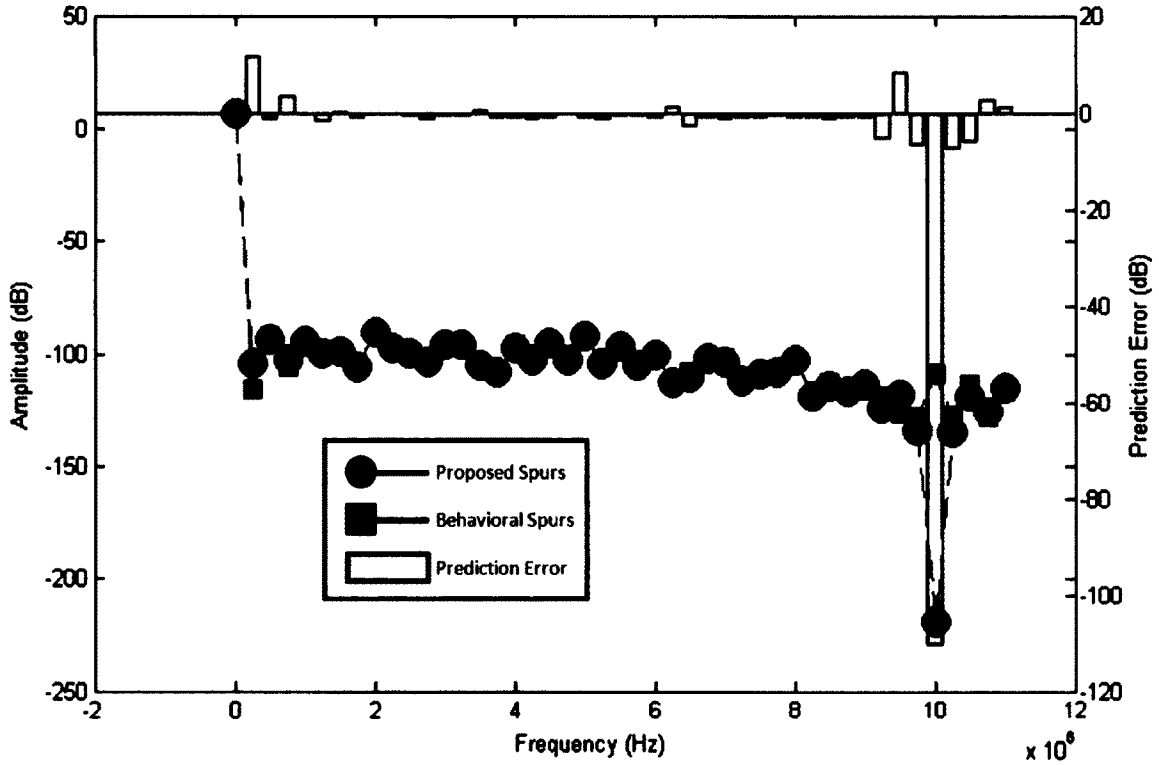
9.1.22 Test Group G – Sweep I_{CP} – 500 uA

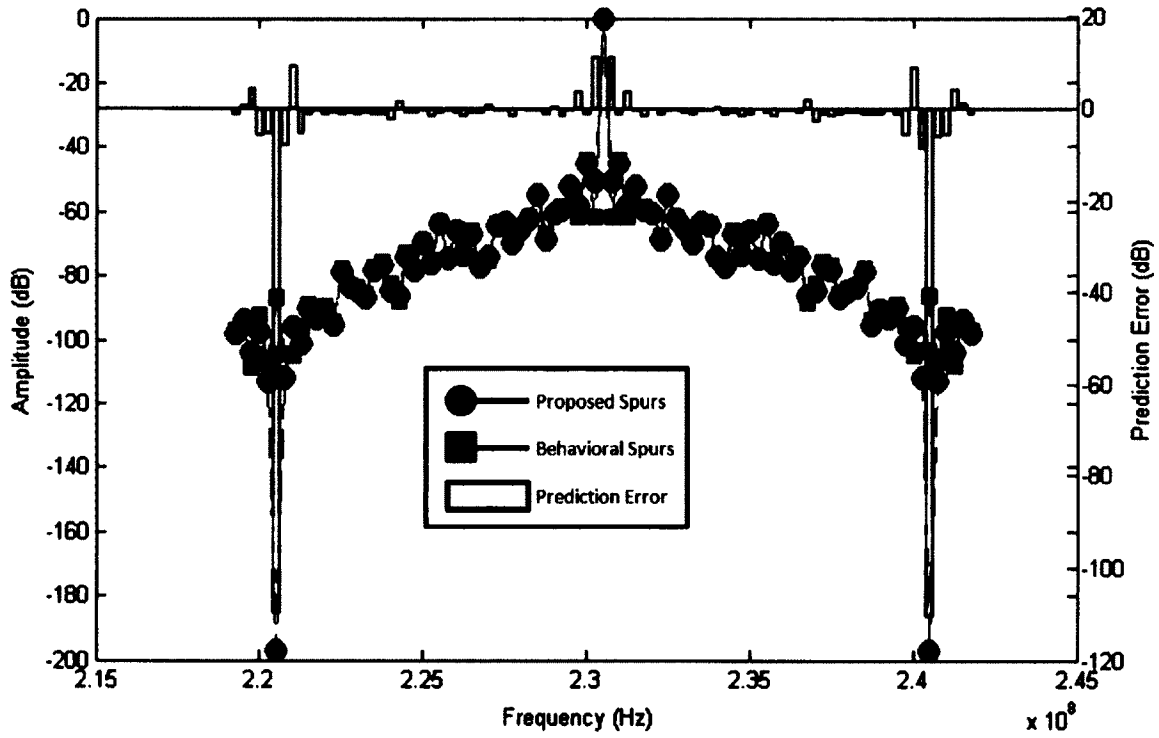


9.1.23 Test Group H – Sweep I – 11

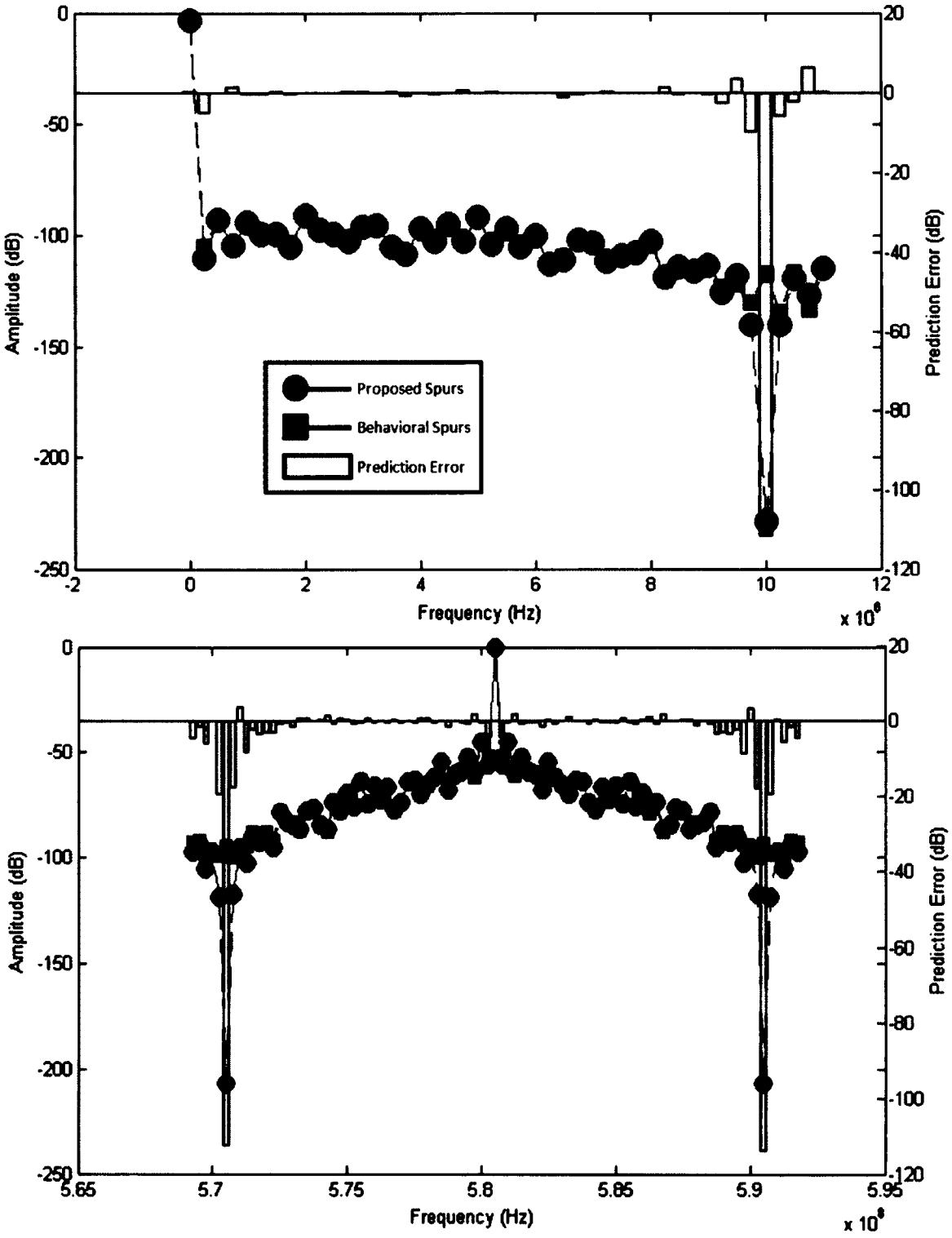


9.1.24 Test Group H – Sweep I – 23

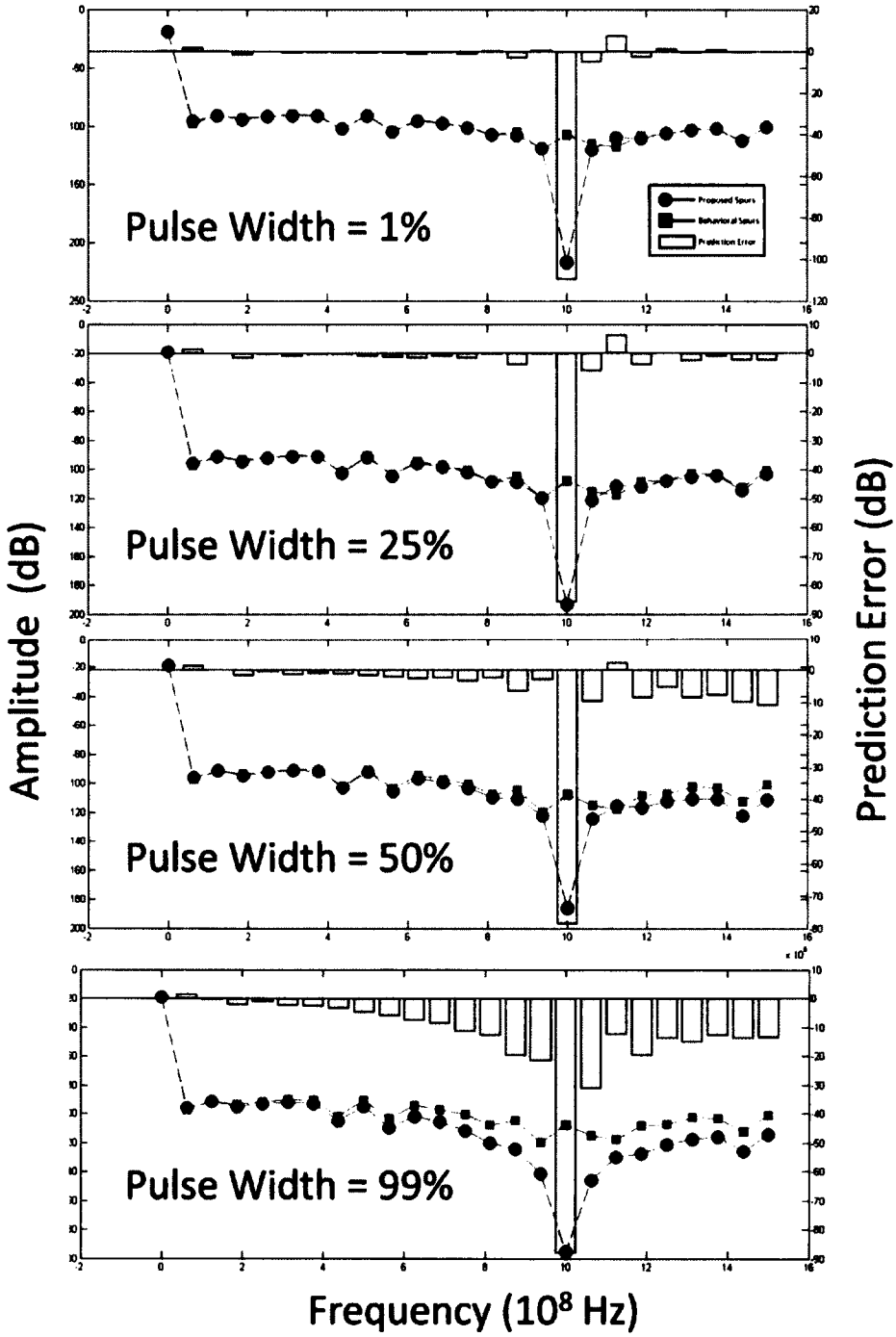




9.1.25 Test Group H – Sweep I – 58



9.1.26 Test Group I – Sweep Pulse Width – 1%, 25%, 50%, 99%



References

[1] S. Haykin, M. Moher, Introduction to Analog and Digital Communications, 2nd edition. New Jersey: Wiley, 2007.

[2] J. Rogers, C. Plett, F. Dai, Integrated Circuit Design for High-speed Frequency Synthesis. Norwood, MA: ARTECH HOUSE, INC, 2006.

[3] Hedayati, H.; Bakkaloglu, B.; Khalil, W.; , "Closed-Loop Nonlinear Modeling of Wideband Sigma Delta Fractional-N Frequency Synthesizers," Microwave Theory and Techniques, IEEE Transactions on , vol.54, no.10, pp.3654-3663, Oct. 2006.

[4] J. Rogers, C. Plett, Radio Frequency Integrated Circuit Design, 2nd edition. Norwood, MA: ARTECH HOUSE, INC, 2010.

[5] R. Best, Phase-locked Loops, 6th edition. New York, NY: McGraw-Hill, 2007.

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

[7] Le Brun, M.; , "A Derivation of the Spectrum of FM with a Complex Modulating Wave," Computer Music Journal , vol.1, no.4, pp.51-52, Nov. 1977.

[8] Xiaozhou Yan; Xiaofei Kuang; Nanjian Wu; , "An accurate and fast behavioral model for PLL Frequency Synthesizer phase noise/spurs prediction," Custom Integrated Circuits Conference, 2009. CICC '09. IEEE , vol., no., pp.223-226, 13-16 Sept. 2009.

[9] Perrott, M.H.; Trott, M.D.; Sodini, C.G.; , "A modeling approach for Σ-Δ fractional-N frequency synthesizers allowing straightforward noise analysis," Solid-State Circuits, IEEE Journal of , vol.37, no.8, pp. 1028- 1038, Aug 2002.

[10] Cassia, M.; Shah, P.; Bruun, E.; , "Analytical model and behavioral simulation approach for a ΣΔ fractional-N synthesizer employing a sample-hold element," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* , vol.50, no.11, pp. 850- 859, Nov. 2003.

[11] Butterfield, D.; Sun, B.; , "Prediction of fractional-N spurs for UHF PLL frequency synthesizers ," Technologies for Wireless Applications, 1999. Digest. 1999 IEEE MTT-S Symposium on , vol., no., pp.29-34, 21-24 Feb 1999.

[12] Riley, T.A.D.; Copeland, M.A.; Kwasniewski, T.A.; , "Delta-sigma modulation in fractional-N frequency synthesis ," *Solid-State Circuits, IEEE Journal of* , vol.28, no.5, pp.553-559, May 1993.

[13] Borkowski, M.J.; Riley, T.A.D.; Hakkinen, J.; Kostamovaara, J.; , "A practical Δ-Σ modulator design method based on periodical behavior analysis," Circuits and Systems II: Express Briefs, IEEE Transactions on , vol.52, no.10, pp. 626- 630, Oct. 2005.