New Approaches in Noise-Shaping Fractional-N Synthesis

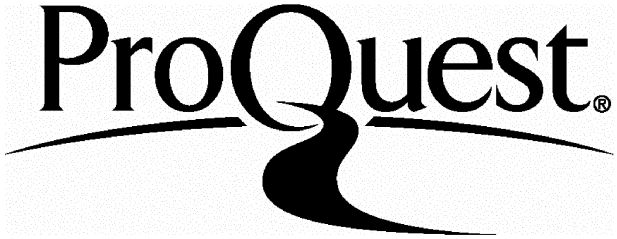
by

Rob Walkington

A thesis submitted to the University of London for the Degree of Doctor of Philosophy in Electronic Engineering

Department of Electronic and Electrical Engineering
University College London

December, 2001



Abstract

Modern mobile communication systems require high stablility high resolution frequency sources able to change frequency at high speed under digital control. The high speed channel changing requirement can result from security requirements and the need for frequency diversity to reduce the apparent effects of multipath fading.

For stability the signal source is invariably implemented as a phase locked loop (PLL) or indirect frequency synthesiser. This allows an unstable voltage controlled oscillator to be phase locked to a highly stable crystal oscillator by means of a control loop.

The requirements of high resolution and high speed channel change are conflicting and frequently lead to multiplexing two synthesisers to satisfy demanding system specifications. This results in a high component count and demanding isolation requirements between the synthesisers, leading to costly screening. An alternative technique to increase the performance of a single loop is a Fractional-N synthesiser switched by a $\Sigma\Delta$ modulator.

The main limitations of $\Sigma\Delta$ Fractional-N loops stem from the need to implement the $\Sigma\Delta$ modulator in digital hardware. Digital multipliers and adders limit the reference frequency of the modulator which limits the acheivable noise spreading performance of the synthesiser system. The need to operate at high reference frequencies also leads to a requirement to truncate the $\Sigma\Delta$ loop filter coefficients to allow implementation as bit shifts which further degrades the noise response.

This thesis concerns the study of an alternative system where a modulator sequence is generated offline for each channel and stored in memory, making very high reference frequency operation possible with highly optimised noise shaping. The increased exploitation of the $\Sigma\Delta$ noise shaping would allow a single chip solution in many applications currently using dual synthesisers.

.....To Michelle and my Parents.

Acknowledgements

I would like to express my sincerest gratitude to Dr. P.V. Brennan for his expert supervision throughout the course of this work. It was he who first interested me in synthesiser design. His skill in helping me deal with numerous problems during the course of this work extended way beyond the call of duty. In addition his insightful and interesting discussions on engineering and life have done a great deal to enlighten me over the years and will never be forgotten. Without his input and direction this work would never have been possible.

In addition I would also like to give my sincerest thanks to Mr. I.V. Thompson who has acted as my Industrial Supervisor during this work. His profound knowledge of microwave engineering and synthesisers in particular have allowed me to learn an invaluable amount about the theoretical and hardware aspects of engineering across the board. Obtaining the results in this thesis would not have been possible without his involvement.

Finally I am exceedingly grateful to Nokia (Camberley) in the U.K. and the Engineering and Physical Sciences Research Council for providing the financial support which made this work possible.

Contents

List of Figures

Symbols

b_n	Peak amplitude of Nth Fourier component	V
e_{RMS}	Mean quantisation noise power	W
f_b	Baseband bandwidth	Hz
f_{clk}	Clock frequency	Hz
f_s	Sampling frequency	Hz
F_{out}	Output frequency	Hz
F_{ref}	Reference frequency	Hz
H_{NTF}	Noise transfer function	V/V
H_{STF}	Signal transfer function	V/V
K_p	Phase detector gain	V/Rad
K_v	VCO gain	Rad/s/V
n_b	Mean baseband noise power	W
N	Divider ratio	
R_D	Dynamic range	dB
t_A	Memory access time	s
t_H	Hold time	s
t_{MX}	Multiplexer combinational logic delay	s
t_S	Setup time	s
t_{TR}	Wire transport delay	s
T_{HP}	Highpass transfer function	V/V
T_{LP}	Lowpass transfer function	V/V
V_{fm}	Frequency modulating voltage	V
V_{pm}	Phase modulating voltage	V
W_A	Width of address bus	bits
W_D	Width of data bus	bits
X_m	Complex Fourier component	V

Δ	Peak quantisation error	V
$\Delta V_{eff}[n]$	Effective divider modulus step	V
$\Delta\phi$	Quantiser phase uncertainty	Rads
$\Delta\phi[n]$	Instantaneous phase step at phase detector	Rads
λ	Real quantiser gain at output fundamental frequency	V/V
ϕ_e	Phase error	Rads
ϕ_i	Input phase	Rads
ϕ_o	Output phase	Rads
ϕ_{pm}	Phase margin	Rads
ω_c	Filter cutoff frequency	Rad/s
ω_i	Input natural frequency	Rad/s
ω_n	Loop natural frequency	Rad/s
ω_o	Output natural frequency	Rad/s

Chapter 1

Introduction

1.1 Historical Background

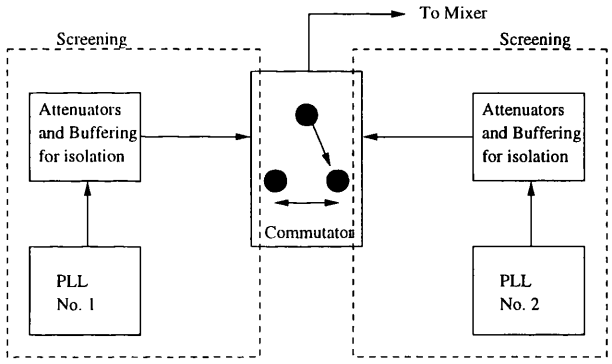


Figure 1.1: Multiplexed synthesisers for frequency hopping.

Phase locked loops (PLL's) have been in widespread use in commercial equipment since the seventies and are firmly entrenched as one of the standard building blocks of electronic design. Applications are diverse, the most well known perhaps being their use as local oscillators in communications equipment. Another widespread application is in television signal carrier recovery and they are also used in large digital systems for clock synchronisation at the board level and generation of higher speed internal clocks within ASICs and CPUs, where GHz I/O would be problematic.

In the 50s, before the advent of PLL's, the provision of a stable and accurate local oscillator was a great problem in the design of communications equipment, which was generally of the single superhet, or single intermediate frequency variety. In general, free-running LC oscillators were in use tuned by mechanical variable capacitors and inductors had to be switched to cover a wide tuning range. The main problem was one of thermal drift.

Heating and cooling of capacitors and inductors caused changes in the resonance of the tuned circuit, further exacerbated by the heat from thermionic valves and the long wiring to the necessary band change switch. This problem could be countered to a degree by the use of positive and negative temperature coefficient capacitors, but setting up such a design for low drift was frought with problems.

The advent of the transistor around the sixties did much to alleviate the situation of oscillator drift by reducing heat generation. Stability was achieved within a few minutes of switch on rather than hours for valved equipment, however the frequency was still prone to change by the ambient temperature of the surroundings. In addition a high resolution readout by mechanical means was a costly affair, requiring elaborate precision-gearboxes and dials between the manual dial and the variable capacitor. The problem of tuning a narrow band R.F. transformer in perfect unison with the local oscillator added to the manual adjustment problems.

An improvement to the problems of resolution, stability and image rejection was to use a double superhet architecture, where two stages of mixing were used. A common arrangement would be to have a first stage mixing with a crystal oscillator, with a number of switched crystals for different frequency bands to mix down to a constant lower frequency band. A high first I.F. eased the RF filtering requirements allowing a wide bandpass filter whilst a low second I.F. provided good selectivity. The output signal from first stage of conversion was a further mixing with the free running oscillator, which would now only have to cover a narrow band at low frequency which eased the problems of both calibration and stability. Such receivers were in general expensive to manufacture and still required hand calibration.

When the advent of integrated circuits the PLL became a perfect candidate for mass production as it cured in one fell swoop the problems of stability, resolution and frequency readout and could be automatically manufactured with repeatable results and no manual adjustment. Development of highly selective narrow band crystal and ceramic filters which also had high center frequencies led to a return to a single superhet architecture in many communications applications. Cheap mass production of high performance equipment opened up a whole new realm of mobile applications. As demand for spectrum for mobile services increased and technology developed the push to higher frequencies led to more and more applications in the GHz range, the most widespread today being mobile telephony.

At GHz frequencies there is considerable multipath fading of signals in an urban environment due to the wavelength of these signals in relation to buildings. In addition the requirement to maintain communications in fast moving vehicles added to the problems of the communications link leading to the use of frequency hopping to provide frequency diversity. With the most demanding applications it was not possible to simultaneously meet the resolution and hopping requirements with a conventional single synthesiser. Present equipment generally overcomes this problem by using two synthesisers, one of which is re-tuning whilst the other is in use (figure 1.1.). Although the individual synthesiser requirements are considerably lower this technique is costly and technically demanding due to the problems of isolating the active synthesiser from the re-tuning one and extensive mechanical screening and buffering is generally required.

This work concerns the study of a novel type of Fractional-N synthesiser, known as a stored-sequence Fractional-N synthesiser, which has particular advantages in the application of frequency hopping mobile communications systems. The technique combines the increased tuning agility and resolution advantage of a traditional Fractional-N synthesiser with the noise shaping abilities of a Sigma-Delta ($\Sigma\Delta$)Fractional-N synthesiser and the high reference frequency capabilities enabled by offline sequence generation. This enables a very high spectral purity to be achieved simultaneously with fast channel switching.

1.2 An example application, The DCS 1800 standard

An example application, forming the original impetus for this work, which typifies the synthesiser requirements for a modern mobile communications system is the DCS mobile phone standard. Designs for this specification at present are invariably met using a dual synthesiser approach.

The table below summarises a few of the DCS requirements:

Tuning Range	1805-1880 MHz
Switching time	10μs
Timeslot Length	577μs
Channel spacing	200 kHz

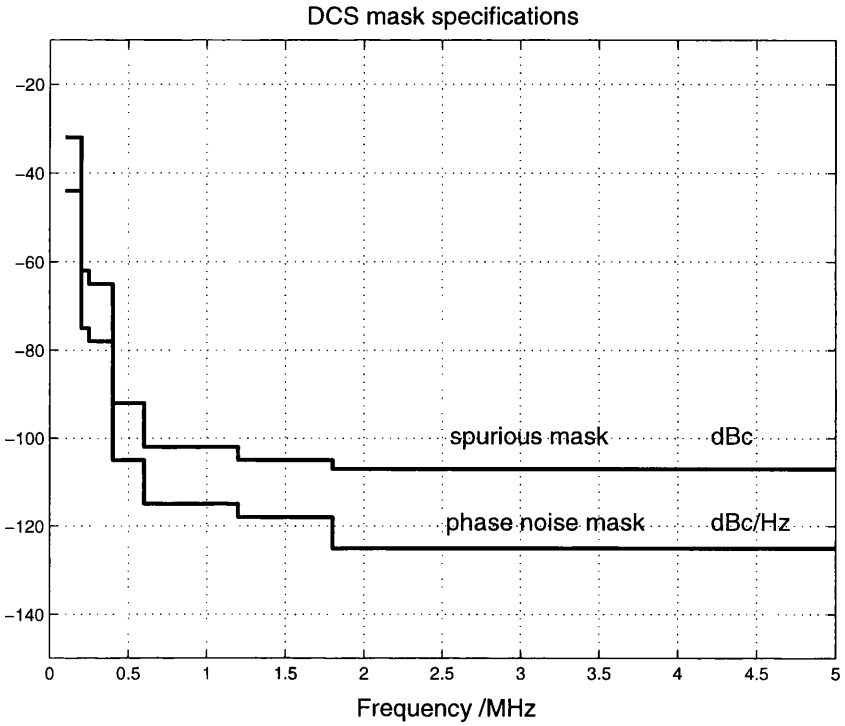


Figure 1.2: DCS specification masks.

In addition, figure 1.2 shows the specification masks for the phase noise and spurs at the output of a DCS transceiver. This represents a budget for spurious and phase noise for the complete transceiver system which means that the actual synthesiser specification will be somewhat more demanding, depending on the amount of this budget used up by the other system blocks, most notably the power amplifier where the issue of intermodulation distortion will be critical. The actual synthesiser tuning range will be at some offset from the receiver input or transmitter output frequency, dictated by the overall architecture.

1.3 Limitations of a single loop synthesiser

Figure 1.3 shows a block diagram and signal flow graph for a basic indirect synthesiser system. The design of an indirect synthesiser is little different from any other feedback control system with the same criteria used to determine stability and inverse Laplace transforms used to convert from the frequency to the time domain response. The loop filter is denoted by G(s) and determines the characteristics of the loop. Other branches of the signal flow graph are tabulated below:

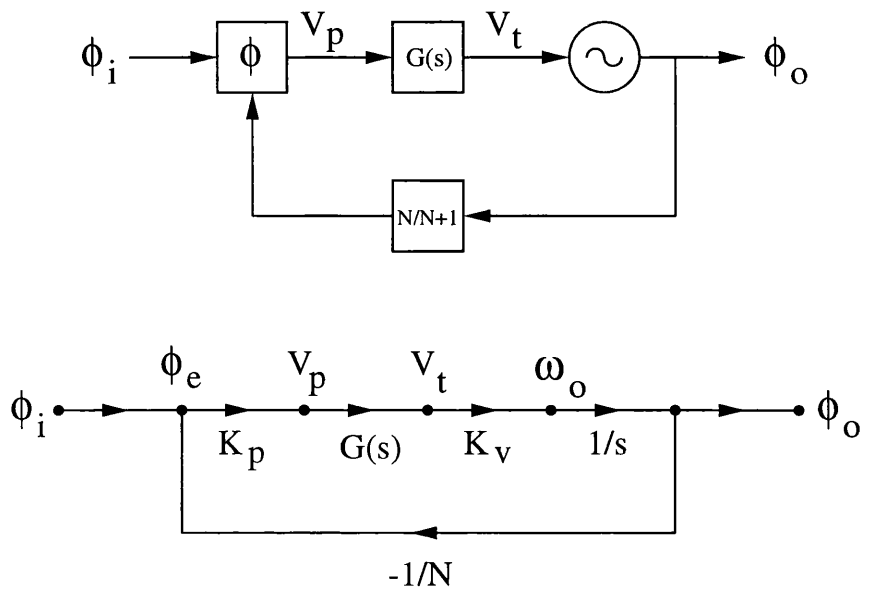


Figure 1.3: Block diagram and signal flow graph for a generic indirect synthesiser.

K_p	=	Phase detector gain in volts/radian.
K_v	=	VCO gain in radians/second/volt.
N	=	The ratio of the divider in the feedback path.
$1/s$	=	1/s factor to convert from VCO frequency to phase.

The phase detector is worthy of note due to it's limited linear range of 4π radians, which causes an effect known as cycle slipping to occur if the range is exceeded. Even if the phase detector range is exceeded the effect of cycle slipping can often be neglected as it is usually a small proportion of the total acquisition time.

To demonstrate the performance limitations of current designs, we consider an example design using a third order type 2 loop filter [1], which is invariably used in commercial PLL designs. This loop filter has a high frequency roll off of 40dB/decade. The reference frequency is determined by the channel spacing, which in the case of the DCS system is 200KHz. A primary concern in the design of the loop filter is that the reference spurs due to the AC component of the phase detector output are adequately suppressed to prevent adjacent channel interference. Referring to the spurious mask of figure 1.2 a spur at this offset from the carrier must be below -30dBc.

1.4 Fractional-N Phase Locked Loops

In order to raise the loop reference frequency whilst retaining a small step size, a non-integer divider is required in the loop feedback path. In fractional-N loops the problem of obtaining a fractional division ratio in the feedback path of the loop is addressed by toggling the ratio of a dual modulus divider in a cyclic fashion [1]. The actual division ratio is found from the average of the two fixed ratios, e.g. 8.75 could be obtained by dividing by nine three times and eight once. This makes it possible to obtain output frequency steps of less than the loop reference frequency, allowing a larger reference frequency which in theory allows a larger loop bandwidth to be used. The major problem with this approach is that spurs appear at the loop output, offset at the repetition rate of the dual modulus divider switching. Open loop compensation techniques have been used to cancel the spurs introduced by divider switching, but a spur level of -60dBc is typical of the best performance that can be obtained. A very low loop bandwidth could be used to ensure the loop only responds to the average division ratio, but this nullifies the agility advantages of the higher reference frequency.

1.5 $\Sigma\Delta$ Modulators

The $\Sigma\Delta$ modulator [2, 3]) is best known for its use in oversampling analog to digital converters. Figure 1.4. shows a third order $\Sigma\Delta$ modulator architecture. For a DC input, the output bitstream has the same average value as the input over its limit cycle. Although the quantiser, in this case, has only a one-bit binary output of either zero or one, the action of the feedback loop ensures that the average value of the output must equal the DC input.

The modulator, having at least one integrator in the forward path, exhibits a low-pass response to the input signal and a high-pass response to the quantisation noise. This results in an output bitstream with reduced low frequency and increased high frequency quantisation noise. This feature is most useful in fractional-N loops where the modulator can be used to drive a dual modulus divider for a particular average division ratio. The high pass filtering of the quantisation noise achieved by the modulator allows the remaining higher frequency quantisation noise to be more readily removed by the loop filter. Because of the real time generation of the $\Sigma\Delta$ bitstream, this approach to implementing a fractional-N synthesiser has the following limitations:

1. *Limited operating speed*, due to technology restrictions on digital logic generating real-

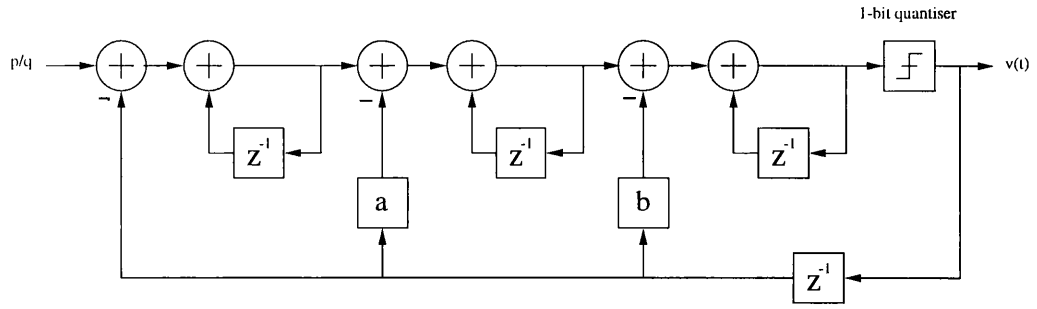


Figure 1.4: Third order ΣΔ modulator architecture

time ΣΔ bitstreams. Recent designs have reported operation up to 50MHz (refs. [4] to [5])

2. *Restriction to integer weights.* In order to achieve fast operation, the ΣΔ loop uses integer weights which results in short limit cycles and gives limited design flexibility. Similarly, the modulator resolution is limited by hardware constraints.

3. *Non optimal weights for each channel.* The optimum weights required to minimise in-band tones at the output vary from channel to channel, a feature that would be difficult to incorporate in this architecture.

4. *Stability is not guaranteed.* It is difficult to predict the stability of the hardware ΣΔ modulator.

1.6 A new approach for fractional-N synthesisers

Figure 1.5. shows a new approach to the implementation of a sigma delta controlled fractional-N synthesiser [6, 7, 4]. Instead of fabricating a sigma delta modulator in hard-wired logic a memory can be used to store pre-generated ΣΔ modulator output bitstreams for each of the required division ratios corresponding to each of the output channels. These can then be clocked out of the memory to drive the dual modulus prescaler division ratio.This avoids many of the limitations inherent in a fixed point logic implementation, but at the expense of having to store a pre-simulated bitstream for each channel.

One of the benefits of this approach is in the increased speed that the bitstream can be clocked out into the divider control input, which is mainly determined by the memory access time. By avoiding the inherent trade off between precision and speed found in the hardware adders and multipliers necessary in a conventional digital modulator it is possible to achieve both very high speed of operation and very high effective bit precision of the internal modulator circuitry. High precision is necessary to ensure long ΣΔ mod-

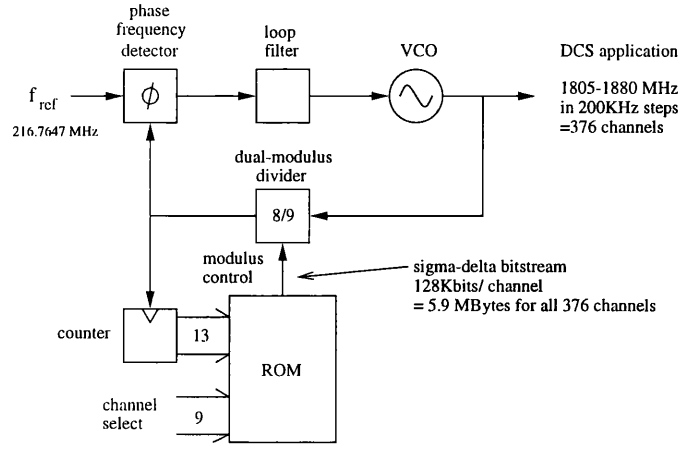


Figure 1.5: ΣΔ Fractional-N synthesiser based on memory stored bitstreams

ulator limit cycles which will repeat at a very low rate and thus have the maximum spur spreading capability due to the mostly noise-like characteristics of the output waveform. Higher speed operation pushes the quantisation noise even further away from the carrier.

The ΣΔ modulator is simulated offline with a different fixed input between 0 and 1 corresponding to the division ratio for each channel from N to N+1. The resulting output of the simulation is downloaded into an eprom on the hardware allowing the final synthesiser easier to obtain the same results in terms of resolution and ΣΔ noise shaping performance as if operating with a modulator with 32 bits or more of precision, but with an effective clock speed of up to ten times that of a conventional hardware implementation.

As DCS1800 signals transmit or receive GMSK on a particular frequency for only a 577μs time slot this makes it feasible to store a complete recording of the output sequence of the modulator to play back over this period of time, and in principle obtain an identical final result to an actual modulator if the synthesiser is only to be used in this burst mode of operation. For a 200 MHz reference this results in a storage requirement of below 128kbits/channel and for 400 200kHz channels to cover the DCS 80MHz range the requirement is 51.2Mbits. Using a lower reference frequency results in a smaller storage requirement but also increases the bin size in the frequency domain, which will result in a smaller number of larger amplitude spurs at the modulator output. If the spurs can be filtered by the loop filter to an adequate extent to meet system specifications then there is no requirement for more speed and memory, but there is the inevitable trade off here with the synthesiser switching speed.

1.7 Structure and Contents of Thesis

Chapter 2 provides a fundamental theory of frequency synthesiser operation presenting the direct and indirect methods. For the indirect method detailed analysis is given of the major loop components and the derivation of the control equations describing the closed loop system. The closed loop performance is analysed in the frequency domain and the time domain leading to a discussion of the design trade-offs inherent in the design of indirect synthesisers. This leads to a discussion of the reasons for using Fractional-N methods to increase the performance of an indirect synthesiser and the limitations of conventional Fractional-N implementations.

Chapter 3 presents an introduction to Sigma Delta modulators and describes their fundamental advantages when used as the modulus controller for a Fractional-N loop. It is shown that despite the non-linear nature of the quantiser in the $\Sigma\Delta$ loop it is possible to derive useful analytical results relating the linear part of the $\Sigma\Delta$ system to the quantisation noise shaping. Finally loop filter design for modulators with arbitrary noise transfer functions is also discussed.

Chapter 4 presents a detailed description of the design and analysis of the new synthesiser using a stored noise shaping sequence approach. Coverage is given to the linear analysis of the synthesis loop, modelling of the sigma delta sequence spectra and combining these results using a narrow band phase modulation approximation of the switching noise injected by the dual modulus divider switching.

Chapter 5 presents results from a hardware implementation of the new architecture and compares these with the results from simulations for a number of frequencies across the synthesiser output range. By a performance comparison of simulations of various published architectures it is shown how the application of higher order noise shaping sequences combined with the new architecture allows much higher spectral purity and/or frequency hopping speed than alternative techniques.

Chapter 6 presents new results relating to the optimisation of sigma delta modulators specifically in relation to fractional-N synthesis applications. Recent published results allow a modified root locus technique to be applied to the nonlinear sigma delta loop in order to ascertain a design starting point for the loop filter for stable running under conditions of zero input. From this initial loop filter the large signal stability of the modulator

is determined by simulation for a range of DC inputs necessary for Fractional-N applications. It is shown that this allows a frequency-specific optimisation of the loop filter to be performed.

Following the synthesis of a range of fractional sequences across the band of operation, the second stage of optimisation consists of finding the optimum truncation length in order to remove the effects of the finite length of the stored sequences. It is shown that this can be achieved using an auto-correlation of the generated sequences to measure the repetition length of all of the required fractional sequences. It is shown that by selecting a length which is the lowest common multiple of all sequence lengths across the band, the effects of the finite sequence length can be avoided.

Chapter 7 summarises the work and draws conclusions from the results gained. This is followed by a discussion of the system applications of the new architecture and some suggestions for future work.

Chapter 2

PLL Systems Design

This chapter covers the design of generic phase locked loop systems, and uses techniques common to the design of all linear systems. Although some of the effects and system blocks present in a PLL system operate in a highly non-linear manner it will be shown that, for the extraction of certain key performance criteria such as loop filter suppression and lock time, assumptions can be made which in most cases allow highly accurate results to be obtained.

A study of the overall system performance will then be presented using the different classes of loop filter. This will develop a clear picture of how the global system performance relates to the specification of this key subsystem.

The final part of this chapter introduces the concept of a fractional-N loop and shows how this technique can be used to increase the performance of the loop. In addition coverage is given to subsystems which are specific to this type of PLL synthesiser.

G(s)	Loop Filter response $\frac{V_t}{V_p}$
K_p	Phase detector gain in volts/radian.
K_v	VCO gain in radians/second/volt.
N	The ratio of the divider in the feedback path.

Table 2.1: PLL system components

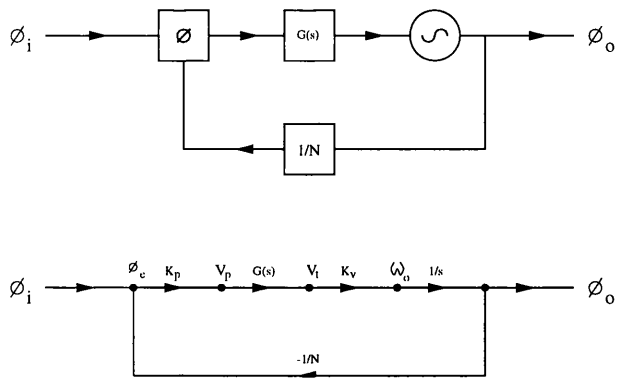


Figure 2.1: Block diagram and signal flow graph of a phase locked loop system

k	=	Number of forward paths
T_k	=	the *k*th forward path gain
Δ	=	1-$\sum$ loop gains + $\sum$ non-touching loop gains taken two at a time - $\sum$ non-touching loop gains taken three at a time + . . .
Δ_k	=	Δ – Σ loop gains touching the *k*th forward path.

Table 2.2: Components of Masons Rule.

2.1 Phase locked loop systems

Figure 2.1 shows the major components of a phase locked loop system. The system components are defined in table 2.1. It can be seen that the VCO is represented by a gain factor, K_v and also a 1/s term. This term serves to translate variations in VCO output frequency into phase and means that all loops are at least type 1 (*1 integrator*) regardless of the type of the loop filter. K_p is the gain of a phase/frequency detector which is the average differential output voltage divided by the phase difference between its inputs.

The transfer function of this signal flow graph is found using Masons rule, the general form of which is given by equation 2.1. The individual components of Masons rule are detailed in table 2.1. As there is only a single loop in the system the loop response can be written by inspection giving equation 2.2.

$$G(s) = \frac{C(s)}{R(s)} = \frac{\Sigma_k T_k \Delta_k}{\Delta} \tag{2.1}$$

$$\frac{\phi_o(s)}{\phi_i(s)} = \frac{\frac{K_p K_v G(s)}{s}}{1 + \frac{K_p K_v G(s)}{Ns}} \tag{2.2}$$

The major points of concern in the design of such a loop are the loop filter suppression, which affects the suppression of noise and in particular reference frequency feed-through from the phase detector. In general increasing the loop filter suppression will require a lower loop filter cutoff which will narrow the loop bandwidth and reduce the acquisition speed of the loop resulting in a major design tradeoff. In addition in any feedback loop the problem of stability must be addressed. Stability of a linear system may be easily achieved by ensuring that all closed loop poles lie on the left hand side of the s plane, ensuring that all system poles have convergent responses.

2.2 Simple Loop Filter Acquisition and Tracking

With reference to equation 2.2, the simplest possible form of PLL system arises when the loop filter transfer function G(s) is set equal to one. This results in a first order type one system due to the VCO integrating action, with a response given by 2.3. This is a low pass response with a high frequency roll-off of 20dB/decade and a corner frequency of $s = K_p K_v / N$. The time constant of this first order system is given by equation 2.5, allowing the response to be rewritten as equation 2.4.

$$\frac{\phi_o(s)}{\phi_i(s)} = \frac{\frac{K_p K_v}{s}}{1 + \frac{K_p K_v}{Ns}} \equiv \frac{K_p K_v}{s + \frac{K_p K_v}{N}} \tag{2.3}$$

Or

$$\frac{\phi_o(s)}{\phi_i(s)} = \frac{N}{\tau} \frac{1}{s + \frac{1}{\tau}} \tag{2.4}$$

Where

$$\tau = \frac{N}{K_p K_v} \tag{2.5}$$

2.2.1 First Order type One Phase Step Response

Each time the divider value N of the loop is changed this is equivalent to injecting a step change in phase into the loop. The inverse Laplace transform of a unit step is 1/s, allowing the Laplace transform of the step response to be written (equation 2.6). A partial fraction expansion of equation 2.7 yields the result of 2.8.

$$\phi_i(s) = \Delta\phi\frac{1}{s} \tag{2.6}$$

$$\phi_o(s) = \Delta\phi\frac{N}{\tau}[\frac{1}{s}\frac{1}{s+\frac{1}{\tau}}] \tag{2.7}$$

$$\phi_o(s) = \Delta\phi\frac{N}{\tau}[\frac{\tau}{s} - \frac{\tau}{s+\frac{1}{\tau}}] \tag{2.8}$$

Inverse Laplace transforming 2.8 yields the time domain step response 2.9. And substituting the result for the time constant back in yields the step response in terms of the loop filter values 2.10. The time constant yields the amount of time taken for the loop to reach 63% of its final value of phase. Figure 3.23 shows the normalised time domain step response confirming the relation between time constant and acquisition time.

$$\phi_o(t) = \Delta\phi\frac{N}{\tau}[\tau - \tau e^{-(\frac{1}{\tau})t}] \tag{2.9}$$

$$\phi_o(t) = \Delta\phi N[1 - e^{-(\frac{1}{\tau})t}] \tag{2.10}$$

$$\phi_e(t) = \Delta\phi(t) - \frac{\phi_o(t)}{N} \equiv \Delta\phi(t) - \Delta\phi[1 - e^{-(\frac{1}{\tau}t)}] \tag{2.11}$$

The resulting phase error at the phase detector output is found by subtracting the output phase referred to the phase detector node from the applied input phase. It is clear from equation 2.11 that as t tends to infinity the steady state phase error due to an input phase step is zero.

2.2.2 First Order type One Frequency Step Response

Of greater interest in PLL applications is the steady state phase error due to a frequency step or phase ramp. Equation 2.83 gives the input phase for a frequency step and its

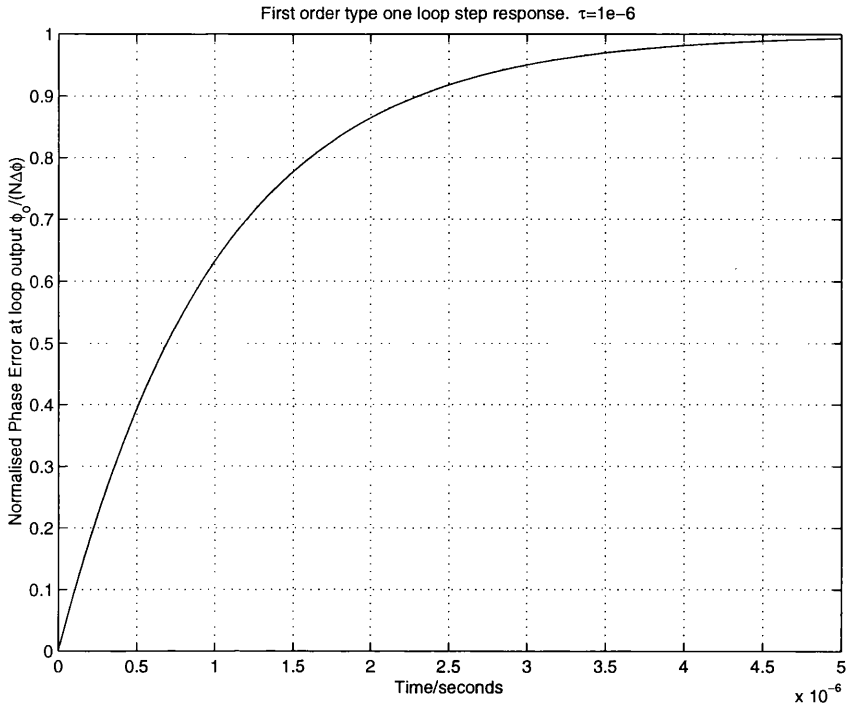


Figure 2.2: First order type One loop phase step response.

Laplace transform. The resulting system response is found from the product of the input phase and system response (2.13). Using the partial fraction identity of 2.14 allows the system response to be re-written as equation 2.15. Inverse Laplace transforming equation 2.15 allows the time domain output phase to be written (2.16).

$$\phi_i(t) = \Delta\omega t \Longleftrightarrow \phi_i(s) = \frac{\Delta\omega}{s^2} \tag{2.12}$$

$$\phi_o(s) = \frac{N\Delta\omega}{\tau}[\frac{1}{s^2}\frac{1}{s+\frac{1}{\tau}}] \tag{2.13}$$

$$\frac{1}{s^2}\frac{1}{s+K} = \frac{\frac{1}{K}}{s^2} + \frac{\frac{1}{K^2}}{s+K} - \frac{\frac{1}{K^2}}{s} \tag{2.14}$$

$$\phi_o(s) = \frac{N\Delta\omega}{\tau}[\frac{\tau}{s^2} + \frac{\tau^2}{s+\frac{1}{\tau}} - \frac{\tau^2}{s}] \tag{2.15}$$

$$\phi_o(t) = \frac{N\Delta\omega}{\tau}[\tau t + \tau^2 e^{-(\frac{1}{\tau})t} - \tau^2] \equiv N\Delta\omega(t + \tau(e^{-(\frac{1}{\tau})t} - 1)) \tag{2.16}$$

Equation 2.17 calculates the phase error at the phase detector output as before. The steady state phase error is calculated from allowing time in 2.16 to tend to infinity. This results in the steady state phase error result of expression 2.18.

$$\phi_e(t) = \phi_i(t) - \phi_o(t)/N \tag{2.17}$$

$$\phi_e(t) = \Delta\omega t - \Delta\omega(t-\tau) = \Delta\omega\tau = \frac{\Delta\omega N}{K_p K_v} \tag{2.18}$$

Clearly the first order type one loop responds to an input frequency step with a finite phase error. which is clearly an unacceptable result for the majority of PLL applications. Further analysis for steady state response for a frequency ramp (parabolic change in input phase) shows that the resulting steady state error is infinite, which in explains the practical problems with such loops in acquisition and tracking. Figure 2.3 shows the resulting phase error transient at the phase detector output with $\Delta\omega$= 2 radians/second and $\tau = 1e-6$s.

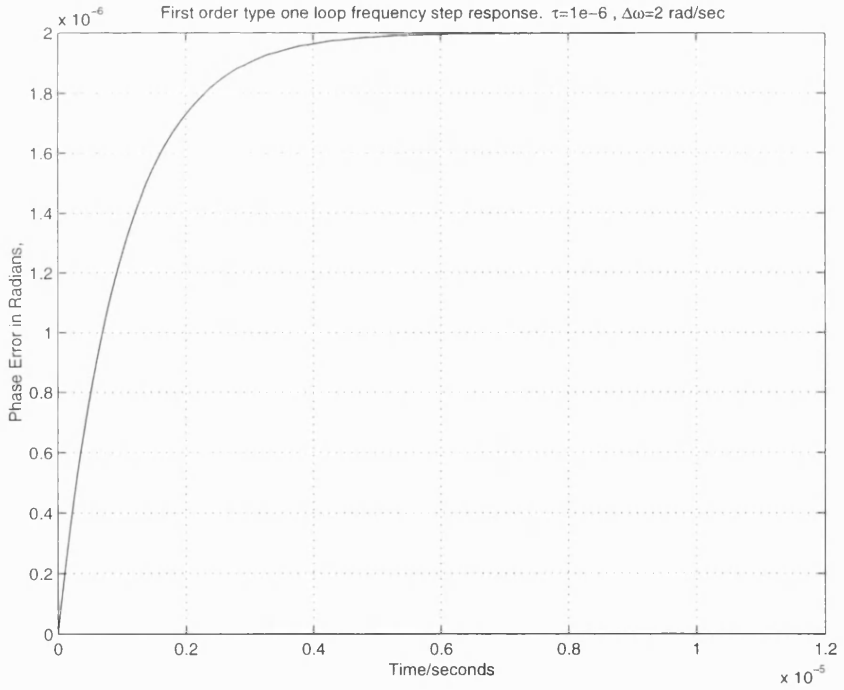


Figure 2.3: First Order Type 1 Loop Frequency Step Response

Whilst the order of the passive loop filter can be increased, increasing the rate of roll off, this will do nothing to reduce the steady state error due to an input phase ramp and the same acquisition problems will be encountered. For these reasons passive loop filters are generally the last resort in certain high speed applications where active loop filters prove

too difficult or unstable to implement.

To improve dynamic performance at least two integrators are required in the PLL control loop. With two integrators it becomes possible to track a frequency step with zero phase error or a frequency ramp with a finite error. This means an additional integrator must be provided in the loop filter in addition to the integration inherent to the VCO. One way of doing this is to use active loop filter incorporating an op-amp. This has the potential to combine the functions of integrator and loop filter. A higher order filter will also improve spectral performance due to offering increased suppression of reference sidebands.

2.3 Second Order type 2 loop Filters

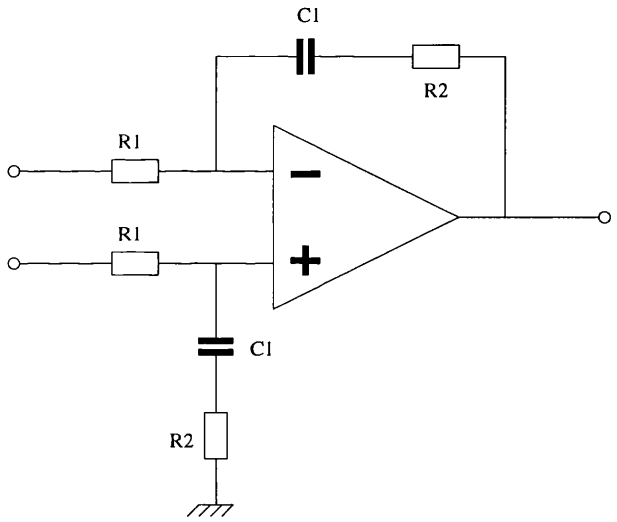


Figure 2.4: Second Order type 2 Loop Filter

Figure 2.4 shows the circuit diagram of a second order type 2 loop filter. The response of this circuit can be written directly from inspection giving the result of equation 2.19. The open loop filter gain clearly shows the open loop transfer function to contain a zero frequency pole due to the integrator function and a single zero due to the high pass function.

$$\frac{V_o}{V_i} = \frac{R2 + \frac{1}{sC}}{R1} = \frac{1}{CR1}\frac{1}{s}(scR2 + 1) \tag{2.19}$$

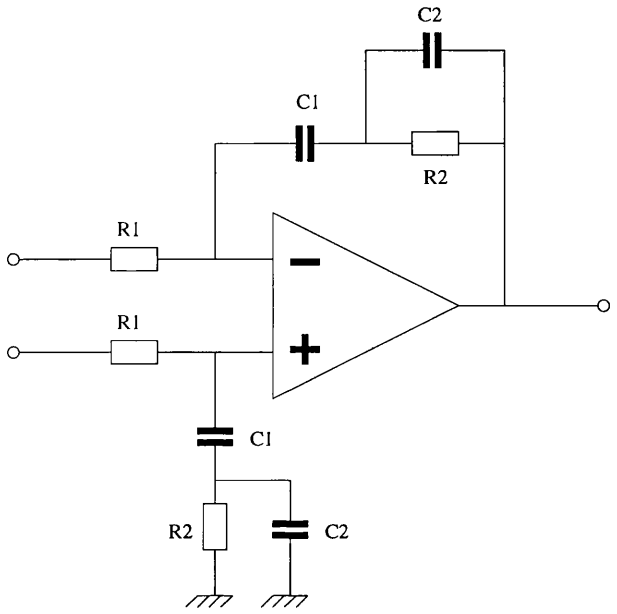


Figure 2.5: Active filter for the third order type 2 loop

2.4 Third Order type 2 Loop Filters

Figure 2.5 shows the circuit of a third order type II loop active filter. This loop filter is a common choice as it offers a good tradeoff between loop agility, spur suppression and ease of analysis [1]. The transfer function of this filter is:

$$G_F(s) = \frac{\frac{1}{sC_1} + \left(\frac{\frac{R_2}{sC_2}}{R_2 + \frac{1}{sC_2}}\right)}{R_1} \equiv \frac{\frac{1}{sC_1} + \left(\frac{R_2}{sC_2R_2+1}\right)}{R_1} \tag{2.20}$$

$$G_F(s) = \frac{1}{sC_1}\left(1 + \frac{sR_2C_1}{sC_2R_2 + 1}\right) \equiv \frac{1}{sC_1R_1}\left(\frac{sC_2R_2 + 1 + sC_1R_2}{sC_2R_2 + 1}\right) \tag{2.21}$$

$$G_F(s)\frac{1}{sC_1R_1}\left(\frac{1 + s(C_1 + C_2)R_2}{1 + sC_2R_2}\right) \tag{2.22}$$

which can also be expressed as:

$$G_F(s) = \frac{1}{s\tau_1}\left[\frac{1+s\tau_3}{1+s\tau_2}\right] \tag{2.23}$$

Where:

$$\begin{aligned} \tau_1 &= R_1C_1 \\ \tau_2 &= R_2C_2 \\ \tau_3 &= R_2(C_1+C_2) \end{aligned}$$

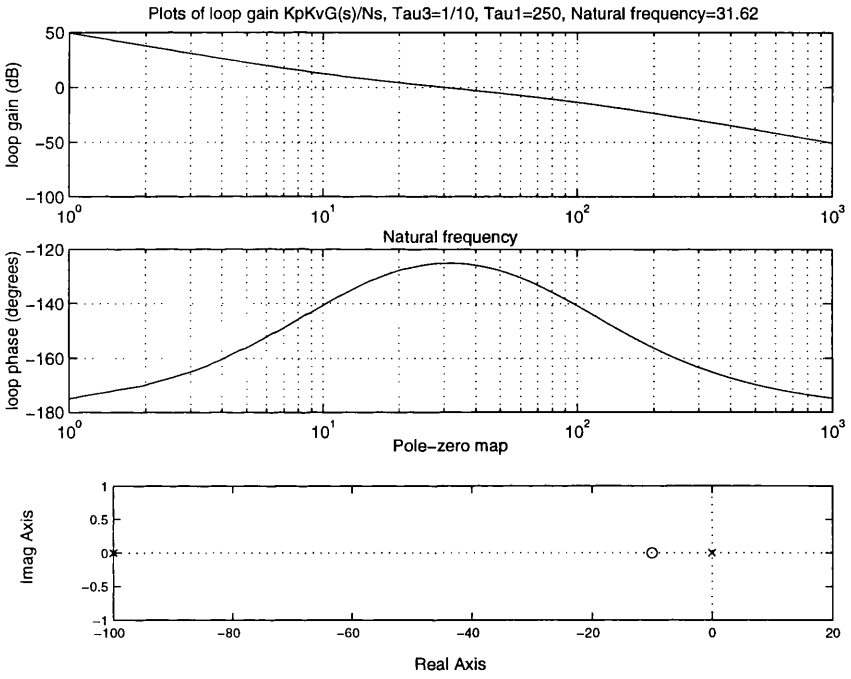


Figure 2.6: Open Loop Gain of Third Order Type 2 Loop Filter.

Figure 2.6 shows open loop bode plots and a pole zero plot for the third order type 2 loop filter. The design equations for the loop filter are derived by maximising the open loop phase at the frequency of zero open loop gain. This approach results in the most stable system for a given gain-bandwidth product. The Nyquist stability criterion states that for a feedback system there must be no frequency at which the loop gain is zero and the phase 180^o. The extent to which loop phase is below 180^o at the zero gain frequency is termed the phase margin and gives a measure of the stability of the system.

The phase of the loop filter is calculated by summing numerator phase angles and subtracting denominator phase angles:

$$\angle G_F(s) = -\frac{pi}{2} + atan(s\tau_3) - atan(s\tau_2) \tag{2.24}$$

The integrator is expressed as a $\pi/2$ phase shift whilst the lag and lead phase angles can be combined by re-arranging the formula for the tangent of two angles:

$$tan(a-b) = \frac{tan(a) - tan(b)}{1 + tan(a)tan(b)} \tag{2.25}$$

Putting tan(a)=A and tan(b)=B,

$$tan(a-b) = \frac{A-B}{1+AB} \tag{2.26}$$

$$a - b = atan\left(\frac{A-B}{1+AB}\right) \tag{2.27}$$

Substituting for a and b,

$$atan(A) - atan(B) = atan\left(\frac{A-B}{1+AB}\right) \tag{2.28}$$

Giving a formula for the arctangent of the two phases. Applying this to 2.24 yields the following:

$$\angle G_F(s) = -\frac{pi}{2} + atan\left(\frac{\omega\left(\tau_3 - \tau_2\right)}{1+\omega^2\tau_2\tau_3}\right) \tag{2.29}$$

To find the frequency at which maximum phase shift occurs in the filter the differential of the filter phase (2.29) is equated to zero:

Using:

$$\frac{d\left(atan\left(\angle G_F(s)\right)\right)}{d\omega} = \frac{1}{1+\theta^2} \tag{2.30}$$

$$\frac{d\left(atan\left(\angle G_F(s)\right)\right)}{d\omega} = \frac{1}{1+\left(\frac{\omega(\tau_3-\tau_2)}{1+\omega^2\tau_2\tau_3}\right)^2} \tag{2.31}$$

Expanding gives:

$$\frac{d\left(atan\left(\angle G_F(s)\right)\right)}{d\omega} = \frac{1 + 2\omega^2\tau_2\tau_3 + \omega^4\tau_2^2\tau_3^2}{1 + 2\omega^2\tau_2\tau_3 + \omega^4\tau_2^2\tau_3^2 + \omega^2\tau_2^2 + 2\omega^2\tau_2\tau_3 + \omega^2\tau_3^2} \tag{2.32}$$

Equating this derivative of the filter phase response to zero will find the frequency at which minimum phase shift occurs through the filter. The right hand side of 2.32 must be equal to zero if:

$$\omega^4\tau_2^2\tau_3^2 + 2\omega^2\tau_2\tau_3 + 1 = 0 \tag{2.33}$$

Which is a quadratic in ω^2. This quadratic is solved to find the loop natural frequency when:

$$\omega_n^2 = \frac{1}{\tau_2\tau_3} \tag{2.34}$$

The phase margin of the loop is given by:

$$\phi_{pm} = \angle G_F(s) - \frac{\pi}{2} + \pi \equiv \angle G_L(s) + \pi \tag{2.35}$$

Where $G_L(s)$ is the entire loop gain $K_pK_vG_L(s)/Ns$. $\angle G_L(s)$ is equal to the filter phase shift $\angle G(s)$ with the additional $\pi/2$ constant phase shift due to the VCO integrating action. Substituting 2.34 into the equation for the loop filter phase(2.29):

$$\phi_{pm} = -\frac{\pi}{2} + atan\left(\frac{\sqrt{\frac{1}{\tau_2\tau_3}}\left(\tau_3 - \tau_2\right)}{1 + \left(\frac{1}{\tau_2\tau_3}\right)\tau_2\tau_3}\right) - \frac{\pi}{2} + \pi \tag{2.36}$$

$$\phi_{pm} = atan\left(\frac{\tau_3 - \tau_2}{2\sqrt{\tau_2\tau_3}}\right) \tag{2.37}$$

Substituting $\tau_2 = 1/\omega_n^2\tau_3$ from 2.34 into 2.37:

$$\phi_{PM} = atan\left(\frac{\tau_3 - \left(\frac{1}{\omega_n^2\tau_3}\right)}{2\sqrt{\frac{\tau_3}{\omega_n^2\tau_3}}}\right) \tag{2.38}$$

Which gives a quadratic in τ_3:

$$\tau_3 - \frac{2tan\phi_{pm}}{\omega_n}\tau_3 - \frac{1}{\omega_n^2} = 0 \tag{2.39}$$

Where the roots are found in the usual way using:

$$roots = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2.40}$$

In this case $a = 1$, $b = \frac{-2tan\phi_pm}{\omega_n}$ and $c = \frac{-1}{\omega_n^2}$. Equation 2.39 equals zero when:

$$\tau_3 = \frac{tan\phi_{pm}}{\omega_n} - \frac{1}{2}\sqrt{\frac{4tan^2\phi_{pm}}{\omega_n^2} + \frac{4}{\omega_n^2}} \tag{2.41}$$

$$\tau_3 = \frac{tan\phi_{pm}}{\omega_n} - \frac{2}{2}\sqrt{\frac{tan^2\phi_{pm} + 1}{\omega_n^2}} \tag{2.42}$$

$$\tau_3 = \frac{tan\phi_{pm}}{\omega_n} - \frac{1}{\omega_n}\sqrt{sec^2\phi_{pm}} \tag{2.43}$$

$$\tau_3 = \frac{tan\phi_{pm} - sec\phi_{pm}}{\omega_n} \tag{2.44}$$

As $\tau_2 = 1/(\omega_n^2\tau_3)$ the result for τ_2 is:

$$\tau_2 = \frac{1}{\omega_n(tan\phi_{pm} + sec\phi_{pm})} \tag{2.45}$$

Allowing both τ_2 and τ_3 to be found for any desired phase margin, ϕ_{pm}. In order to determine τ_1 the magnitude of the loop gain, $|G_L(s)|$, is equated to unity at the loop natural frequency, ω_n and τ_2 and τ_3 eliminated using 2.44 and 2.45. Substituting 2.23 into $G_L(s)$:

$$|G_L(s)| = \left|\frac{K_pK_v}{\omega_nN}\frac{1}{\omega_n\tau_1}\sqrt{\frac{1 + \omega_n^2\tau_3^2}{1 + \omega_n^2\tau_2^2}}\right| \tag{2.46}$$

$$|G_L(s)| = \left|\frac{K_pK_v}{N\omega_n^2\tau_1}\sqrt{\frac{1 + \omega_n^2\left(\frac{tan\phi_{pm}+sec\phi_{pm}}{\omega_n}\right)^2}{1 + \omega_n^2\left(\frac{1}{\omega_n(tan\phi_{pm}+sec\phi_{pm})}\right)^2}}\right| \tag{2.47}$$

Letting $A = tan\phi_{pm} + sec\phi_{pm}$,

$$|G_L(s)| = \left| \frac{K_p K_v}{N\omega_n^2 \tau_1} \sqrt{\frac{1+A^2}{1+\left(\frac{1}{A^2}\right)}} \right| \tag{2.48}$$

$$|G_L(s)| = \left| \frac{K_p K_v}{N\omega_n^2 \tau_1} \sqrt{A^2} \right| \tag{2.49}$$

$$|G_L(s)| = \left| \frac{K_p K_v}{N\omega_n^2 \tau_1} (tan\phi_{pm} + sec\phi_{pm}) \right| \tag{2.50}$$

Letting this equal unity,

$$\tau_1 = \frac{K_p K_v}{N\omega_n^2} (tan\phi_{pm} + sec\phi_{pm}) \tag{2.51}$$

Allowing the loop design equations to be written for τ_1 and τ_2:

$$R_1 C_1 = \frac{K_p K_v}{N\omega_n^2} (tan\phi_{pm} + sec\phi_{pm}) \tag{2.52}$$

$$R_2 C_2 = \frac{1}{\omega_n (tan\phi_{pm} + sec\phi_{pm})} \tag{2.53}$$

Finding $R_2 C_1$ requires some further trigonometry:

$$R_2 C_1 = \tau_3 - \tau_2 = \frac{1}{\omega_n} \left[tan\phi_{pm} + sec\phi_{pm} - \frac{1}{tan\phi_{pm} + sec\phi_{pm}} \right] \tag{2.54}$$

Which is reduced to:

$$R_2 C_1 = \frac{1}{\omega_n} \left[tan\phi_{pm} + \frac{1}{cos\phi_{pm}} - \frac{1}{tan\phi_{pm} + \frac{1}{cos\phi_{pm}}} \right] \tag{2.55}$$

$$R_2 C_1 = \frac{1}{\omega_n} \left[\frac{cos\phi_{pm} tan\phi_{pm} + 1}{cos\phi_{pm}} - \frac{1}{tan\phi_{pm} + \frac{1}{cos\phi_{pm}}} \right] \tag{2.56}$$

$$R_2 C_1 = \frac{1}{\omega_n} \left[\frac{sin\phi_{pm} + 1}{cos\phi_{pm}} - \frac{cos\phi_{pm}}{cos\phi_{pm} tan\phi_{pm} + 1} \right] \tag{2.57}$$

$$R_2 C_1 = \frac{1}{\omega_n} \left[\frac{sin\phi_{pm} + 1}{cos\phi_{pm}} - \frac{cos\phi_{pm}}{sin\phi_{pm} + 1} \right] \tag{2.58}$$

$$R_2 C_1 = \frac{1}{\omega_n} \left[\frac{(sin\phi_{pm} + 1)^2 - cos^2\phi_{pm}}{cos\phi_{pm}(sin\phi_{pm} + 1)} \right] \tag{2.59}$$

$$R_2 C_1 = \frac{1}{\omega_n} \left[\frac{1 + 2sin\phi_{pm} + sin^2\phi_{pm} - cos^2\phi_{pm}}{cos\phi_{pm}(sin\phi_{pm} + 1)} \right] \tag{2.60}$$

$$R_2C_1 = \frac{1}{\omega_n}\left[\frac{2(sin\phi_{pm} + sin^2\phi_{pm})}{cos\phi_{pm}(sin\phi_{pm}+1)}\right] \tag{2.61}$$

$$R_2C_1 = \frac{1}{\omega_n}\left[\frac{2(sin\phi_{pm}(1 + sin\phi_{pm}))}{cos\phi_{pm}(sin\phi_{pm}+1)}\right] \tag{2.62}$$

$$R_2C_1 = \frac{2tan\phi_{pm}}{\omega_n} \tag{2.63}$$

In summary, the optimal response of the third order type 2 loop filter can be achieved using the following design equations:

$$R_1C_1 = \frac{K_pK_v}{N\omega_n^2}(tan\phi_{pm} + sec\phi_{pm}) \tag{2.64}$$

$$R_2C_2 = \frac{1}{\omega_n(tan\phi_{pm} + sec\phi_{pm})} \tag{2.65}$$

$$R_2C_1 = \frac{2tan\phi_{pm}}{\omega_n} \tag{2.66}$$

2.4.1 Closed loop response

The closed loop response is found by substituting the filter response (2.23) into the closed loop transfer function (2.2):

$$\frac{\phi_o(s)}{N\phi_i(s)} = \frac{\frac{K_pK_v}{s^2\tau_1}\left[\frac{1+s\tau_3}{1+s\tau_2}\right]}{1 + \frac{K_pK_v}{Ns^2\tau_1}\left[\frac{1+s\tau_3}{1+s\tau_2}\right]} \tag{2.67}$$

Which simplifies to obtain the phase modulation characteristic:

$$\frac{\phi_o(s)}{N\phi_i(s)} \equiv \frac{K_p\phi_o(s)}{NV_{pm}(s)} = \frac{\omega_n^2\,(\tan\phi + \sec\phi)\,s + \omega_n^3}{s^3 + \omega_n\,(tan\phi + \sec\phi)\,s^2 + \omega_n^2\,(\tan\phi + \sec\phi)\,s + \omega_n^3} \tag{2.68}$$

Where ω_n is the loop natural frequency, ϕ is the phase margin and N is the divide ratio in the feedback path. This third order low pass response has a roll-off of 40dB/decade at low frequencies. The FM response to the voltage variations at the VCO input can be found by referring this node to the input of the system and taking the output of the system in terms of natural frequency, equivalent to ϕ_o/s:

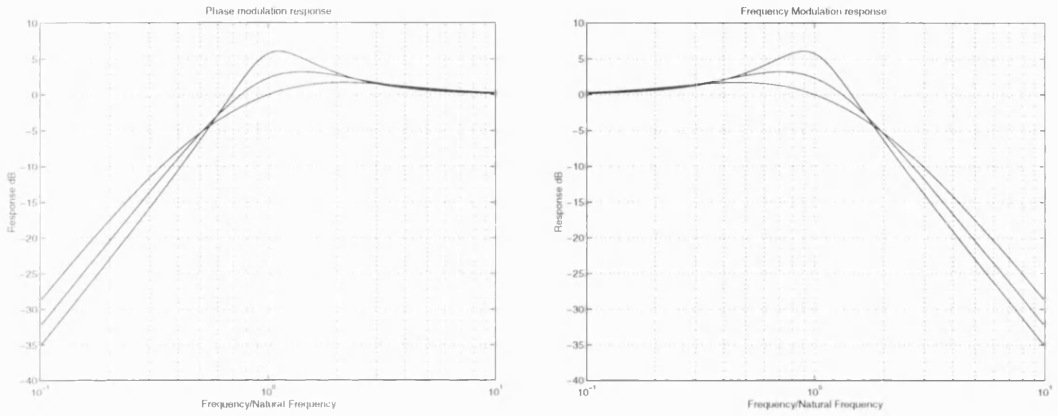


Figure 2.7: Modulation response of loop for phase margins of 30, 45 and 60 degrees.

$$\frac{\omega_o(s)}{V_{fm}(s)} = \frac{K_v}{1 + \frac{K_p K_v G(s)}{N(s)}} \tag{2.69}$$

$$\frac{\omega_o(s)}{N\omega_i} V_{fm}(s) = Kv \left[\frac{s^3 + \omega_n(tan\phi + sec\phi)s^2}{s^3 + \omega_n(tan\phi + sec\phi)s^2 + \omega_n^2(tan\phi + sec\phi)s + \omega_n^3} \right] \tag{2.70}$$

The frequency modulation response (2.70) is a mirror of the phase response, with a high pass characteristic which rolls off at 40dB/decade at low frequencies. Figure 2.7 shows the normalised phase and frequency modulation response of the third order type 2 loop. As phase margin is decreased from 60^o to 30^o the peaks in the modulation responses grow.

2.4.2 Transient Performance

When the divider ratio in the feedback path changes this introduces a step change in frequency into the loop at the divider input node. The closed loop transient response to this step is obtained by an inverse Laplace Transform of the closed loop phase modulation

response (2.68).

Taking phase margin as 53.2^o reduces the value of $(\tan\phi + \sec\phi)$ in 2.68 to 3. This assumption gives realistic results for a range of other realistic phase margins for transient performance and allows simplification of the closed loop phase response 2.71. Splitting into partial fractions gives the result of equation 2.72 and taking the inverse Laplace transform gives equation 2.73 which is the loop response in the time domain. This gives the phase error transient of equation 2.74.

$$\frac{\phi_o(s)}{N\phi_i(s)} = \frac{3\omega_n^2 s + \omega_n^3}{s^3 + 3\omega_n s^2 + 3\omega_n^2 s + \omega_n^3} \tag{2.71}$$

$$\frac{\phi_o(s)}{N} = \Delta\phi\left[\frac{1}{s} - \frac{1}{s+\omega_n} - \frac{\omega_n}{(s+\omega_n)^2} + \frac{2\omega_n^2}{(s+\omega_n)^3}\right] \tag{2.72}$$

$$\frac{\phi_o(t)}{N} = \Delta\phi\left[1 + \left[(\omega_n t)^2 - \omega_n t - 1\right]e^{-\omega_n t}\right] \tag{2.73}$$

$$\phi_e(t) = \Delta\phi - \frac{\phi_o(t)}{N} = \Delta\phi\left[1 + \omega_n t - (\omega_n t)^2\right]e^{-\omega_n t} \tag{2.74}$$

The transient response in terms of frequency rather than phase at the loop output is found from integrating the phase result (2.74) and multiplying by a factor of $\Delta\omega/\Delta\phi$ giving equation 2.83. The $\Delta\omega/\Delta\phi$ factor converts the input step to the system to a frequency step of $\Delta\omega$ rather than a phase step of $\Delta\phi$. Using integration by parts on equation 2.74:

$$\int u\frac{dv}{dt}dt = uv - \int v\frac{du}{dt}dt \tag{2.75}$$

Making the following assignments, leaving out the $\Delta\phi$ term for clarity:

$$u = \left[1 + \omega_n t - \omega^2 t^2\right]e^{-\omega_n t} \longrightarrow \frac{du}{dt} = \omega_n - 2\omega^2 t \tag{2.76}$$

$$\frac{du}{dt} = \omega_n - 2\omega_n^2 t \longrightarrow v = \frac{e^{-\omega_n t}}{-\omega_n} \tag{2.77}$$

Allows an intermediate result to be written for the integral of the phase step response:

$$\int u\frac{dv}{dt}dt = \left(\omega_n t^2 - \frac{1}{\omega_n} - t\right)e^{-\omega_n t} - \int e^{-\omega_n t}(2\omega_n t - 1)dt \tag{2.78}$$

Performing integration by parts on the right hand section of this equation yields:

$$\int u\frac{dv}{dt}dt = \left(\omega_n t^2 - \frac{1}{\omega_n} - t\right)e^{-\omega_n t} - \left(\frac{1}{\omega_n} - 2t - \frac{2}{\omega_n}\right)e^{-\omega_n t} \tag{2.79}$$

And collecting the exponential terms results in:

$$\int u\frac{dv}{dt}dt = \left(\omega_n t^2 - \frac{1}{\omega_n t} - t - \frac{1}{\omega_n} + 2t + \frac{2}{\omega_n}\right) e^{-\omega_n t} \tag{2.80}$$

Cancelling terms and putting the $\Delta\phi$ constant back results in:

$$\int \phi_e(t)dt = \Delta\phi \left(\omega_n t^2 + t\right) e^{-\omega_n t} \tag{2.81}$$

This result is the integral of the phase response to an input phase step of $\Delta\phi$. As the response to frequency is the integral of the response to phase it is only necessary to replace the $\Delta\phi$ term with $\Delta\omega$ to obtain the phase response to a unit step of input frequency:

$$\phi_e(t)dt = \Delta\omega \left(\omega_n t^2 + t\right) e^{-\omega_n t} \tag{2.82}$$

To obtain the equivalent phase and frequency step responses referred to the VCO output node the equations are scaled by the loop divider value. A worst case result for dynamic performance can be obtained by using the larger of the two N-divider values involved in the frequency step.

In order to obtain the transient response in terms of frequency instead of phase, the result for phase response is differentiated, the frequency being the time derivative of phase. Differentiating 2.82 gives the transient frequency response to a frequency step:

$$\frac{d\phi_e}{dt} \equiv \omega_e(t) = \frac{d\Delta\omega[t + \omega_n t^2]e^{-\omega_n t}}{dt} = \Delta\omega[1 + \omega_n t - (\omega_n t)^2]e^{-\omega_n t} \tag{2.83}$$

Figure 2.8 shows the normalised loop transient response in terms of frequency and phase at the phase detector. Used in conjunction with figure 2.7 generated from equation 2.68, the loop filter can be designed for an optimal trade between spur suppression and transient performance. It can be seen that an increased loop natural frequency results in faster channel switching but lower suppression of reference spurs and other signals in the loop which will lead to reduced spectral purity at the loop output. An additional overhead on acquisition time may occur in a real loop due to cycle slipping, which occurs when the phase difference at the phase detector input exceeds the limited $\pm 2\pi$ wrap-around range.

2.5 Fractional-N synthesisers

When performance specifications cannot be met for both spectral purity and switching speed there is little that can be done with a conventional phase locked loop. Raising

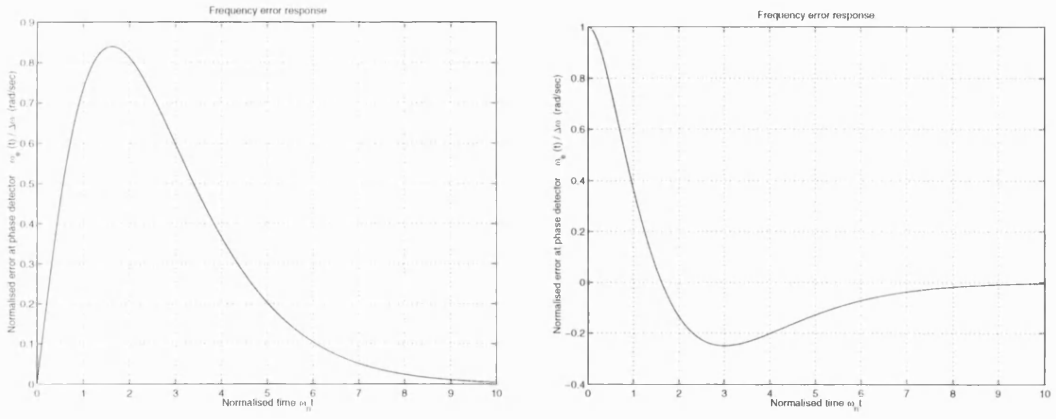


Figure 2.8: Third order type 2 loop response to a frequency step.

loop filter bandwidth to improve switching speed will only result in higher reference spur levels. Raising the reference frequency would move any reference spurs further from the carrier but this reduces the resolution of the synthesiser, leaving no solution if the required channel spacing is not equal to the loop reference.

Fractional-N loops are an attempt to overcome the problem of reduced synthesiser resolution with increasing reference frequency. By switching the divider between two ratios in a cyclic fashion an average non-integer division ratio can be achieved as the loop does not respond instantaneously to the phase changes. This allows the reference frequency to be increased hence also the loop bandwidth. This allows either faster channel changing or higher spectral purity, or a combination of both, to be achieved.

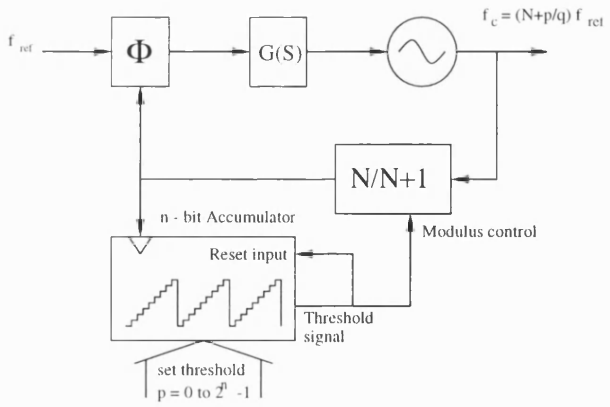


Figure 2.9: Basic single accumulator Fractional-N synthesiser

Figure 2.9 shows a block diagram of a basic fractional-N synthesiser. The primary addition to this system over a normal PLL is the digital accumulator clocked from the output of the feedback divider. The divider is arranged to overflow at some threshold, p, some-

where in its count range between 0 and $2^n - 1$. This threshold is programmed by a parallel digital input word of width n. When the accumulator overflows, an overflow signal is output which does two things. The first is to reset the accumulator to zero and the second is to switch the loop divider modulus from one to the other of its two integer values. The output frequency of this fractional-N loop is given by:

$$F_{out} = (N + (p/q))F_{ref} \tag{2.84}$$

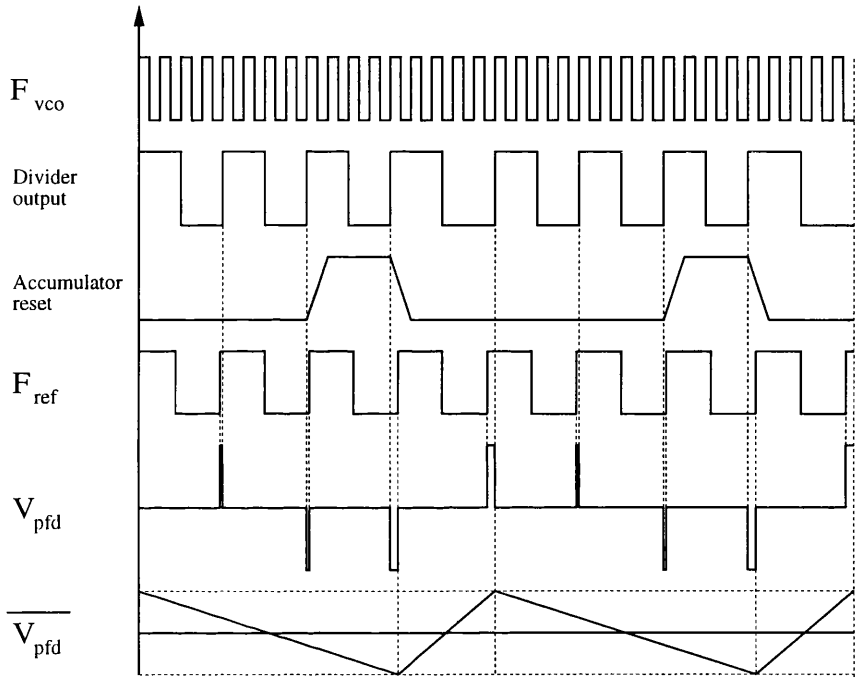


Figure 2.10: Fractional-N operation with 4/5 divider

Figure 2.10 shows a timing diagram of the operation of a fractional N synthesiser. In this example the two divider ratios are 4 and 5. Using three divide by 4 cycles and one divide by 5 one an average ratio of 4.25 is obtained. As is evident from the plot of the averaged phase detector output voltage, $\overline{V_{pfd}}$, the sawtooth phase error introduced into the loop is the main disadvantage of the fractional-N technique.

The peak value of the fractional-N phase sawtooth will be just below π/N radians as $\pm\pi$ of phase is available at the VCO end of the dual modulus divider. The frequency of this phase sawtooth will be F_{ref}/p where p is the reset threshold set for the phase accumulator. Combining this information with the closed loop transfer function allows an equation to be written for the worst case height of the fractional spurs at loop output:

From equation B.16, the height of the Nth Fourier component of a unit peak voltage sawtooth is given by:

$$b_n = \frac{2}{n\pi}(-1)^{n+1} \tag{2.85}$$

Thus the Nth harmonic of the fractional-N sawtooth has a peak phase deviation at the phase detector of:

$$\Delta\phi(n) = \phi_p k(t)\frac{2}{n\pi} \tag{2.86}$$

Where $\Delta\phi(n)$ is the phase deviation of the Nth component of the sawtooth and $\phi_{pk}(t)$ is the peak phase of the time domain sawtooth at the phase detector node. The peak phase deviation of the Nth harmonic of the sawtooth at the loop output is given by:

$$\Delta\phi(n)_{pk} = \frac{\pi}{N}\frac{2}{n\pi}\frac{\phi_o(nf_{ref}/q)}{N\phi_i(nf_{ref}/q)} \tag{2.87}$$

2.5.1 Fractional-N Resolution

On a standard integer-N synthesiser the achievable resolution at the loop output is equal to the reference frequency, as the loop output frequency is given by:

$$F_{out} = NF_{ref} \tag{2.88}$$

Without some way of fractional division there is no way of increasing the resolution of the loop without a smaller reference. For a single-accumulator Fractional-N loop, where the divider ratio is switched each time a digital accumulator overflows, the output frequency is given by:

$$F_{out} = \left[N + \frac{p}{q}\right] \tag{2.89}$$

Where the divider can switch between N and N+1, q is the size of the accumulator clocked by the output of the loop divider and p is the value the accumulator increases by on each rising edge of the divider output. The accumulator will on average overflow p/q of the

divider input cycles. A more generalised equation for other switching schemes might be:

$$F_{out} = (AN + 1 + BN)F_{ref} \tag{2.90}$$

Where A and B are the number of times the dual modulus divider divides by N and N+1 during a complete division cycle for a particular output frequency. Fractional loop resolution is in principle infinite as any average value of division between N and N+1 can be achieved, given a sufficient cycle length. Therefore if a $\Sigma\Delta$ modulator is used to switch the divider this will determine the loop output resolution.

The number of quantiser bits in the sigma delta only determines the quantisation noise added during the A/D conversion as the negative feedback will always cause the output to be a pulse width modulated version of the input. Internal data path resolution determines the discernable resolution at the input, and hence the resolution of the modulator. In a frequency hopping system, provided the length of the time slot is large compared with the length of the modulator conversion cycle, then there will be no restriction on the achievable output resolution other than the modulator data path.

2.6 Sources of Spurii

There are various sources of spurii in the system which affect the requirements of the main loop filter and any additional RF filtering at the VCO frequency. Points of particular note for fractional-N loops are phase detector aliasing and spurs due to the periodic phase steps caused by divider switching.

1. *Reference Spurs*

With all phase locked loops there will be the problem of suppressing the AC component of the phase detector output. The pulsed nature of the output gives rise to phase modulation of the loop at the reference rate, causing sidebands to appear at the loop output offset at the reference frequency. The level of these reference sidebands will be dependent on the closed loop phase modulation response, shown by figure 2.7. In general the loop reference should be at least hundred times higher than the loop natural frequency. This guarantees a reference spur suppression of at least 55 dBc, although many applications

require far more than this.

2 *Fractional-N spurs*

In a Fractional-N loop the periodic switching of the divider gives rise to a 2π phase error at the divider input, which will phase modulate the loop giving rise to fractional sidebands at an offset from the carrier corresponding to the repetition frequency. For a single accumulator fractional-N there is an obvious trade-off between resolution and fractional spur level, as the closer the synthesised frequency is to an integer multiple of the carrier, the more reference cycles will be required to overflow the accumulator and switch the divider.

3. *Sigma Delta Spurs*

With a $\Sigma\Delta$ Fractional-N loop, the fractional spurs are dependent on the spurs at the $\Sigma\Delta$ modulator output, as the modulator directly modulates the loop by means of the divider switching. The highly non-linear nature of the modulator results in components at the $\Sigma\Delta$ modulator output which are the intermodulation products of the reference frequency and synthesised fractional divider output. Figure 2.11 shows a modelled output spectrum of a 3rd order $\Sigma\Delta$ modulator controlled fractional loop, with significant spurious products labeled due to the modulator. The spectrum of the $\Sigma\Delta$ modulator output is referred to the loop output by the appropriate closed loop modulation response (equation 2.68). The NBPM can be approximated as a linear modulation system by a simple scaling of the amplitude of the modulator spectrum.

5. *Phase Detector Aliasing*

The phase detector of a synthesiser can alias components at its input into the loop bandwidth due to the its inherent sampling action. A component entering the phase detector at $F_{ref} - \Delta F$ will appear at the phase detector output aliased down to ΔF due to undersampling. This does not result in a problem which directly affects system design but any unwanted signal finding its way into the phase detector, for example via the power supply lines, will be subject to aliasing in this way. If after undersampling the aliased component falls within the loop bandwidth the loop filter can do nothing to prevent the loop being phase modulated.

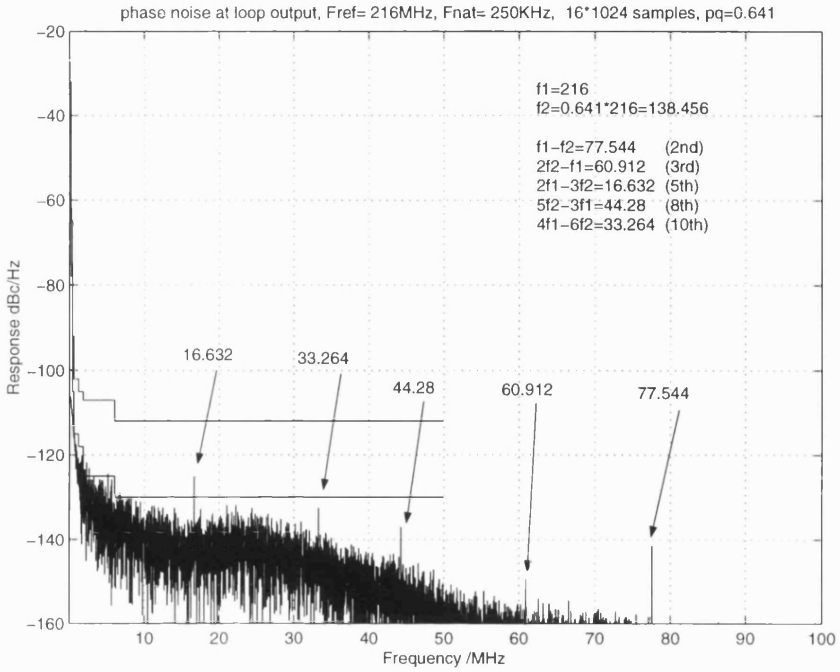


Figure 2.11: Spurs due to $\Sigma\Delta$ modulator referred to the loop output.

2.7 Aided Acquisition

One PLL technique for improving the compromise between tuning agility and spectral purity is known as aided acquisition. This technique reduces tuning time by applying a near-correct voltage for the required output frequency directly to the VCO tuning input after the loop filter. The loop then acquires lock naturally over the small remaining offset, saving the time which would be taken for cycle slips and natural acquisition. The aided acquisition voltage is applied directly at the VCO input, allowing a very fast change in frequency.

The disadvantage of this technique is the further complexity involved, although the amount of extra hardware is small, involving a memory and a DAC in its most simple form. A point of great concern is keeping noise out of the sensitive unfiltered VCO input. As this input will need to have a minimum of filtering to perform fast VCO tuning. Any noise appearing at the VCO tuning input will directly phase modulate the VCO. Where a DAC is used to apply the tuning voltage, any digital clock should be turned off after the aided acquisition is complete, leaving the stored value from the memory lookup table on the DACs input register.

The graphs of Figure 5.13 show acquisition for an example synthesiser with 250 kHz

Initial Frequency Offset	Frequency Offset at 10 μs	Phase Offset after 10 μs at VCO.
60 MHz	2 kHz	1e-2 Radians
20 MHz	1.5kHz	3.2e-3 Radians
10 MHz	500 Hz	1.5e-3 Radians
5 MHz	173 Hz	8e-4 Radians
2 MHz	70 Hz	3.15e-4 Radians
1 MHz	35 Hz	1.6e-4 Radians
500 kHz	17 Hz	7.9e-5 Radians
100 kHz	3.47 Hz	1.6e-5 Radians

Table 2.3: Acquisition tolerance against initial frequency offset.

natural frequency. The graph shows that a 15 MHz hop to within 500 Hz and 0.1 degrees takes around 10 μs assuming there is no cycle slipping. Assuming that a 15 MHz frequency step at the loop output does not result in exceeding the phase detector range, then course pre-steering to within this range can in theory be used to achieve acquisition in a similar timescale across the synthesiser output range, by restricting the worst case initial frequency offset before the exponential pull-in under the natural synthesiser loop characteristic.

To illustrate how the lock time varies according to the size of initial frequency offset, modelled acquisition tolerance for the same synthesiser after 10 μs is shown in table 2.7 for various initial frequency offsets Figure 2.12 plots these results.

Although for a fixed time the pre-steering provides a large reduction in phase and frequency offset, when comparing the time to acquire lock to within 2 kHz the gain does not appear to be as large due to the exponential nature of the acquisition. The 60 MHz offset takes 10 μs to achieve this result and the 10 MHz offset takes 8.7 μs, a reduction of only 10 per-cent in the tuning time despite such a large reduction in the start offset. From this it appears that dividing the initial offset by a factor of 6 has resulted in only a 10 per-cent reduction in tuning time, decreasing this initial offset by a further factor of 6 to 1.66 MHz results in achieving the 2 kHz in 7.33 μs.

Plotting the lock time to within 2 kHz for a number of initial offsets for the 250 kHz loop gives an indication of the relationship between lock time and the accuracy of any pre-steering (Figure 2.13). From this it is clear that for an appreciable speedup the accuracy

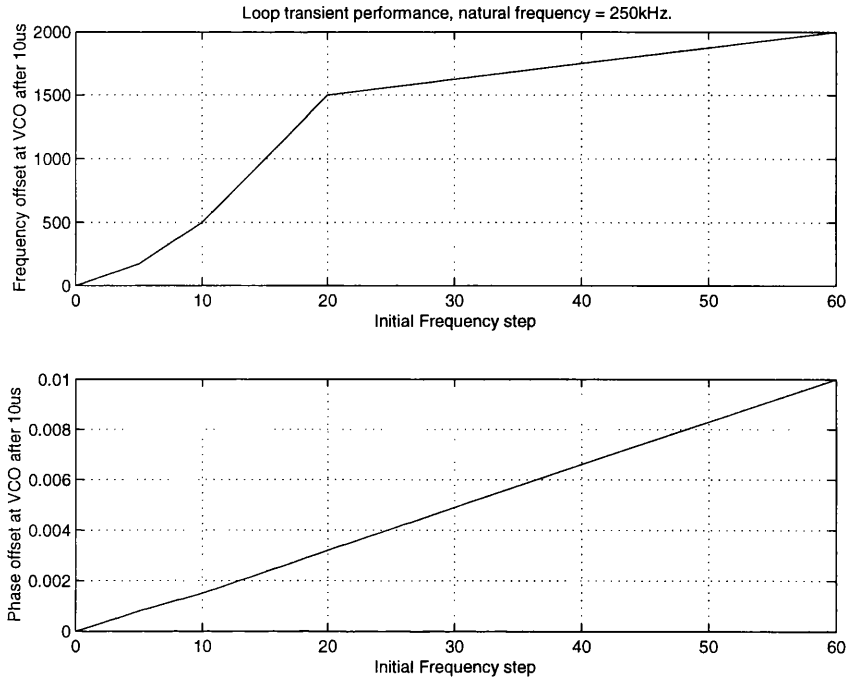


Figure 2.12: Acquisition after 10μs for various initial frequency offsets.

would have to be very high, as an initial offset in the 100 kHz range is required to reduce the tuning time to 2.5 μs with this particular loop frequency. This kind of accuracy would require an adaptive system where the tuning voltage for each required frequency is loaded into memory at system startup and refreshed throughout use, to counteract component drift. The table below shows these plotted results for the loop dynamics.

Initial offset (Hz)	Time to within 2 kHz, (μs)
100e6	10.4
60e6	10.03
10e6	8.7
6e6	8.316
1e6	6.929
100e3	2.55
10e3	2.54

Table 2.4: Table of initial offset against time to reach fixed frequency offset.

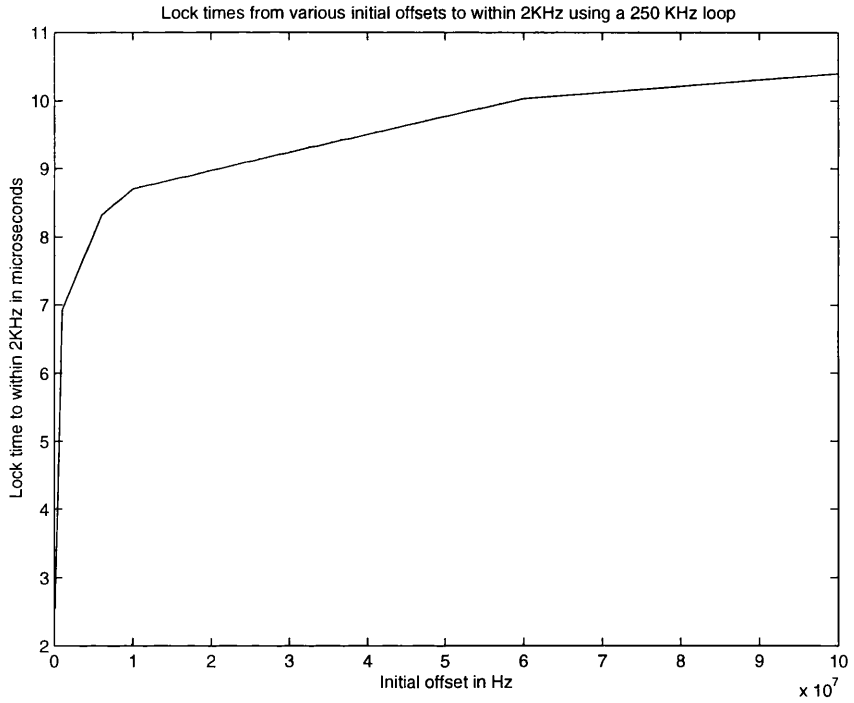


Figure 2.13: $\Sigma\Delta$ Time taken for a 250 kHz loop to acquire to within 2 kHz for various initial offset frequencies.

Chapter 3

Properties of $\Sigma\Delta$ Modulators.

3.1 Sampled Systems

This section gives a general coverage of the properties of sampled systems and goes on to develop the arguments for the advantages of high oversampling ratios. This is developed further to show the further compelling advantages of oversampled sigma delta techniques in signal processing applications.

3.1.1 Quantisation Noise

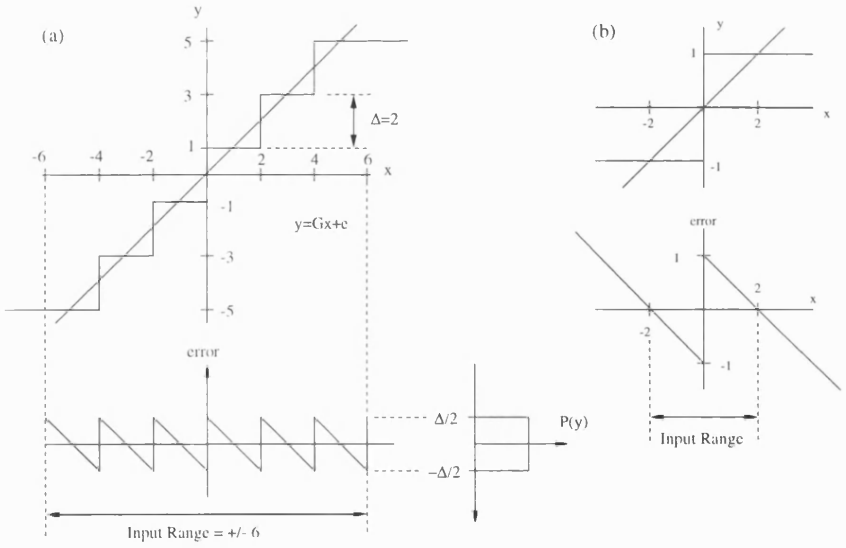


Figure 3.1: (a) Uniform multilevel quantisation characteristic represented by gain G and error *e*. (b) Gain G is arbitrary with single bit quantisation.

Figure 3.1 (a) shows an example quantisation characteristic which rounds off a continuous signal x to odd integers which lie in the range ± 5. In this example the level spacing

Δ is 2, and the quantised signal y can be represented by the straight line through the characteristic, Gx and an error factor e, i.e,

$$y = Gx + e \tag{3.1}$$

If the input signal stays within the range ± 6 then the quantiser will not saturate, and the error e will have a bound of $\pm\Delta/2$. In the case of the single bit quantiser characteristic of figure 3.1 (b) the peak error is also $\pm\Delta/2$ although the gain factor G is arbitrary.

The quantisation error is completely defined by the input signal. If the input signal changes randomly between each sample by an amount comparable to Δ then the error between samples will not be strongly correlated. Hence the error will have equal probability of lying anywhere in the range $\pm\Delta$. Making this assumption, of a constant noise probability density function *P(y)*, allows the quantisation error to be represented as a random noise which is largely independent of the input signal. This can then be used to specify the quantiser dynamic range in terms of the number of bits.

Many real-world signals are sufficiently random in nature that the random quantisation noise assumption yields useful results. However, if the input signal is *stationary* or has a constant periodicity the quantisation noise will be strongly correlated with the input, leading to a more pronounced visibility of the quantisation distortion. The use of more levels of quantisation or higher sample rates will often reduce the apparent effects of this distortion.

Equation 3.2 gives the mean squared noise voltage or noise variance assuming random quantiser noise.

$$e_{rms}^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} e^2 de = \frac{\Delta^2}{12} \tag{3.2}$$

3.1.2 Oversampling

Figure 3.2 (a) shows the one-sided spectral distribution of quantisation noise of a band-limited signal sampled at the Nyquist rate. When a quantised signal is sampled at Nyquist rate, all of the quantisation noise power folds into the band $0 \leq f < f_s/2$. The mean square quantisation noise power e_{rms}^2 can be written as the integral of the mean square noise power per Hz $PSD(e_{rms}^2)$ integrated over the whole frequency band:

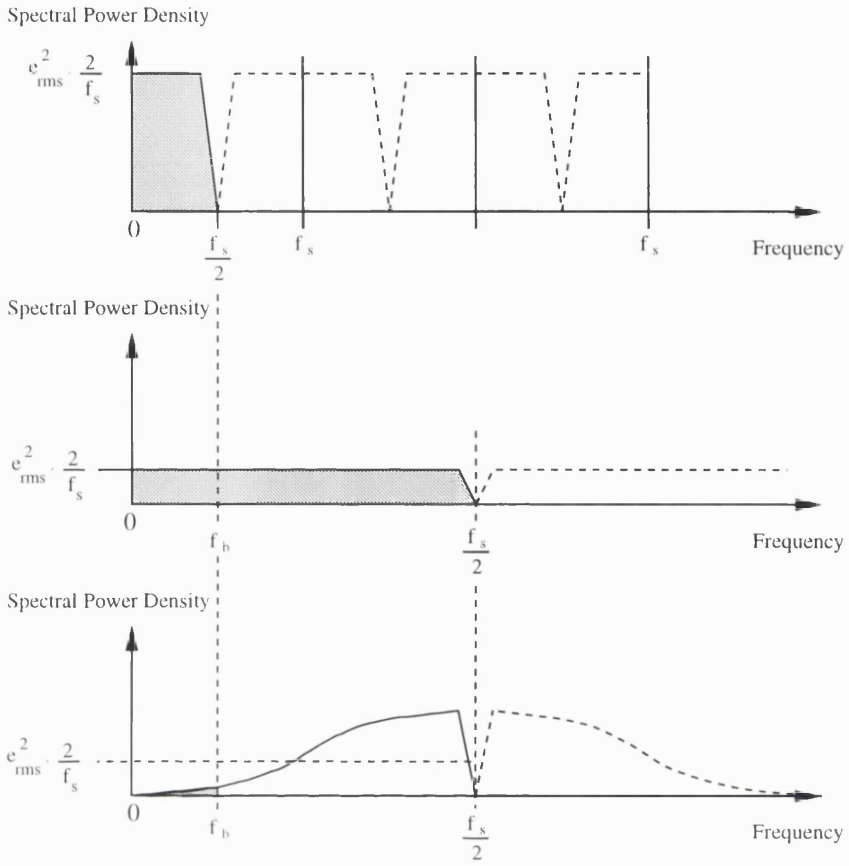


Figure 3.2: Power Spectral Density of Quantisation Noise in Band Limited Data. (a) Sampled at Nyquist rate. (b) Four Times Oversampling. (c) Oversampling with Noise Shaping.

$$e_{rms}^2 = \int_0^{fs_2} PSD(e_{rms}^2)df \tag{3.3}$$

As the one-sided noise spectral density assumes all power is in the range $0 \leq f < \inf$, the quantisation noise spectral density $PSD(e_{rms})$ is double that of the double sided representation. 3.4 gives the power spectral density of uniformly distributed quantisation noise in terms of the quantiser step size and sampling frequency [8]. From this we can see that increased reference frequency or reduced step size both reduce the noise level. Doubling the reference frequency will reduce the noise power density by 3dB, whilst halving the step size results in 6dB reduction. Therefore doubling the reference frequency will increase the resolution by half a bit.

$$PSD(e_{rms}^2) = e_{rms}^2 \cdot \frac{2}{f_s} = \frac{\Delta^2}{6f_s} \tag{3.4}$$

Figure 3.2 (b) illustrates the effect of multiplying the sampling frequency by four. Although the noise bandwidth is four times larger the noise density is reduced to a quarter of the Nyquist value. In A/D applications this is a distinct advantage as only the baseband

up to f_b is retained after conversion. This allows digital processing to be performed at a greatly reduced data rate and reduces the noise in the required baseband.

Sampling at a rate higher than Nyquist is *Oversampling* where the oversampling ratio (OSR) is defined as

$$OSR = \frac{f_s}{2f_b} \tag{3.5}$$

where f_s and f_b are the sampling rate and baseband bandwidth. In order to de-correlate the quantisation noise with the input signal a dither or random noise signal is often mixed with the signal to be converted. This will add some additional noise to the signal but may increase the Spurious Free Dynamic range (SFDR) of the conversion by whitening the quantisation noise spectrum. Assuming the dither signal is sufficiently large to decorrelate the quantisation error the baseband noise power n_b^2 of an oversampled system is given by

$$n_b^2 = \int_0^{f_b} e^2(f)df = \frac{e_{rms}^2}{OSR} \tag{3.6}$$

Figure 3.3 shows a comparison of an oversampled sinusoid power spectral density using a real one bit quantiser and an equivalent noise source. Although the mean noise levels are roughly the same this serves to illustrate the input dependent nature of the distortion caused by the quantiser. Although oversampling has been used, particularly severe distortion has been caused by the use of a single bit quantiser in this example. The rest of this section will show how by the use of oversampling and noise shaping the dynamic range may be increased without an increase in reference frequency.

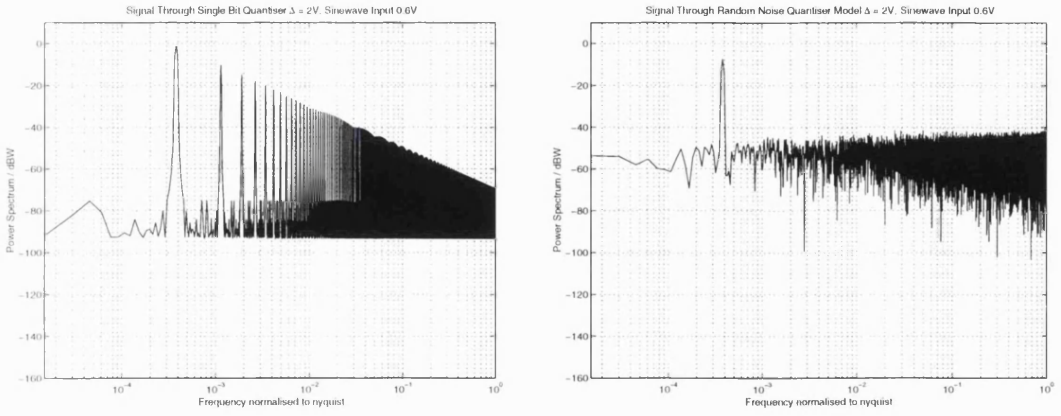


Figure 3.3: Comparison of Sinusoid Power Spectral Density through a one bit quantiser and with equivalent noise added.

3.1.3 Dynamic Range

Signal to noise ratio (SNR) is defined as 10 times the logarithm of the power ratio or 20 times the logarithm of the amplitude ratio of the largest and smallest representable signals in the system.

For an analogue system the dynamic range is defined as the ratio of the largest signal that can be represented without clipping to the smallest detectable signal. The largest representable signal is often determined by the power supply rails whilst the smallest signal is determined by how much noise or interference is present.

For a digital system the largest representable signal is usually determined by the power supply of the A/D or D/A, i.e, if an eight bit 2's complement A/D operates from a 5 V power supply the largest analogue input signal will be 2.5 V. A 2.5V signal will be represented as 11111111 and a -2.5V signal as 00000000. Any input signal above 2.5 V will still be represented by 11111111 and clipping will be occurring, so the largest possible AC input signal will be a 2.5 V sine wave, with a 2.5 V DC offset. The smallest representable signal has the same power as the quantisation error.

Using an N bit DAC with a power supply of V volts and assuming sinewave signals, the largest RMS input voltage is given by:

$$V_{MAX(rms)} = \frac{V}{2}\frac{1}{\sqrt{2}} = \frac{2^{(N-1)}\Delta}{2\sqrt{2}} \tag{3.7}$$

And the smallest rms signal voltage is given by:

$$V_{MIN(rms)} = \frac{\Delta}{\sqrt{12}} \tag{3.8}$$

And the dynamic range is 20 times the log of the ratio of these two rms voltages:

$$R_D = \frac{\frac{2^{(N-1)}\Delta}{2\sqrt{2}}}{\frac{\Delta}{\sqrt{12}}} = \frac{2^{(N-1)}\Delta}{\Delta}\cdot\frac{\sqrt{12}}{2\sqrt{2}} = 2^{N-1}\cdot\sqrt{\frac{3}{2}} \tag{3.9}$$

$$(SNR)_{dB} = 20[(N-1)log_{10}(2) + \frac{1}{2}log_{10}(\frac{3}{2})] = 6(N-1) + 1.76 \tag{3.10}$$

This provides an expression for signal to noise in terms of the number of quantisation levels. With a large number of bits adding another bit will increase signal to noise ratio by approximately 6dB.

3.1.4 Noise Shaping

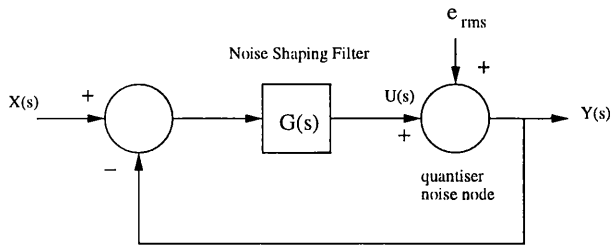


Figure 3.4: Shaping Quantisation Noise by means of a feedback loop and loop filter.

By embedding the quantiser in a feedback loop with suitable filtering it is possible to shape the quantisation noise away from the location of the baseband signal. Clearly with an oversampled system there is much scope to move noise into the unused part of the spectrum between the top of the baseband f_b and half the sampling frequency $f_s/2$.

Figure 3.4 shows the means of shaping quantisation noise using a feedback loop and filtering. The quantiser has is shown as additive noise to allow the shaping of the noise to be analysed, although as discussed previously this noise will seldom be truly white. By using Masons rule of equation 2.1 the noise shaping loop can be defined with a signal transfer function (STF), the transfer function between the signal input port and the output port, and a noise transfer function (NTF) which is the transfer function between the noise input port and the output port.

The STF of the system is written:

$$H_{STF}(s) = \frac{Y(s)}{X(s)} = \frac{H(s)}{1 + H(s)} \tag{3.11}$$

And the NTF:

$$H_{NTF}(s) = \frac{Y(s)}{N(s)} = \frac{1(s)}{1 + H(s)} \tag{3.12}$$

If a highpass function is desired for $H_{NTF}(s)$

$$H_{NTF}(s) = \frac{s}{\omega_c + s} \tag{3.13}$$

Then the required loop filter $H(s)$ can be found from

$$H(s) = \frac{1}{H_{NTF}} - 1 = \frac{\omega_c + s}{s} - 1 = \frac{\omega_c}{s} \tag{3.14}$$

And the resulting STF is

$$H_{STF}(s) = \frac{\frac{\omega_c}{s}}{1 + \frac{\omega_c}{s}} = \frac{\omega_c}{s + \omega_c} \tag{3.15}$$

This results in a highpass transfer function between the noise input port and the output and a lowpass transfer function between the signal input and the output. Both STF and NTF will have a 20dB/decade rolloff with the STF having little effect on the input signal if the cutoff ω_c is far enough above baseband.

The input signal effectively passes through the modulator unaltered as the action of the feedback loop attempts to minimise the difference between Y(s) and X(s). As a result of the limited number of quantiser values the quantiser is switched at a high rate in order to minimise this difference, the maximum rate being dependent on the rate of sampling of the quantiser input. Hence the output consists of the original input signal with the superimposed high frequency quantiser noise. As the input is also being oversampled it is possible to filter out the unwanted high frequency noise as in the case of figure 3.2 (b). Figure 3.2 (c) illustrates how the noise has been high pass filtered by the sigma delta (ΣΔ) loop which allows more of the noise to be removed by subsequent low pass filtering. Figure 3.5 shows the resulting noise and signal transfer functions for this first order modulator.

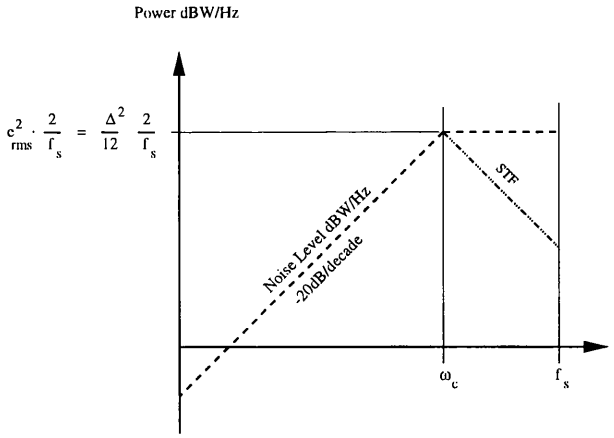


Figure 3.5: Signal and Noise transfer functions for First Order Sigma Delta Modulator

3.2 The First Order Sigma Delta Modulator

The proposed Fractional-N loop is to have its dual modulus divider controlled by ΣΔ modulator output waveforms, with average DC levels appropriate to the required division value. Referring to figure 3.7, the simplest form of digital sigma delta modulator consists

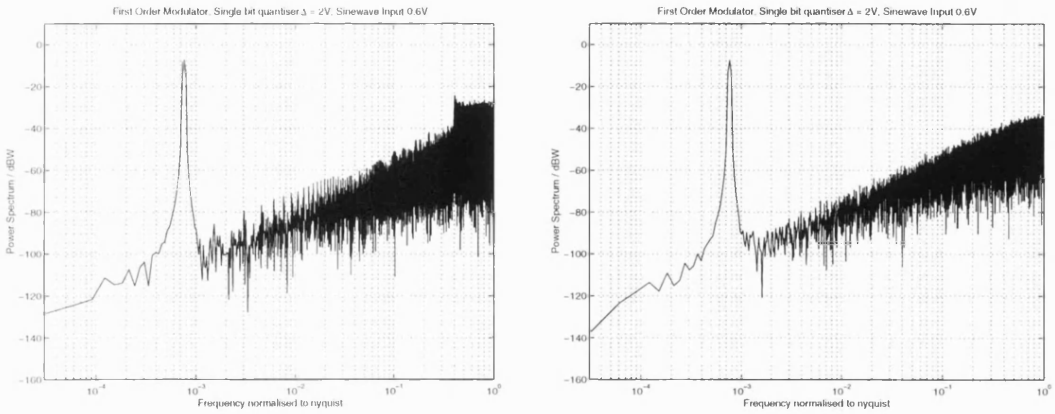


Figure 3.6: First Order Modulator Simulation with single bit quantiser and additive noise model. Input sinewave amplitude = 0.6V, Quantiser Δ = 2V.

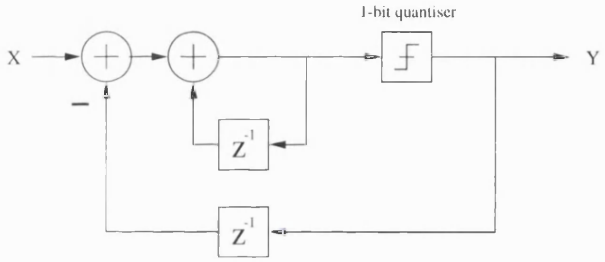


Figure 3.7: A first order Sigma Delta Modulator

of a feedback loop with an integrator, or accumulator, and a single bit quantiser in the forward path. A delay of one sample is included in the feedback path to avoid contention [2].

The input signal, X, which has range of 0 to 1, enters the modulator through the summing junction and passes through the integrator to the quantiser, which for a one bit sigma delta is a simple comparator. The integrator output is compared to some reference level, in this example 0.5. If the accumulator value is larger than 0.5 then the feedback into the input summing node is 1, which will tend to reduce the accumulator value. If the accumulator output is smaller than 0.5, the feedback signal to the input summing node is 0, which will allow the accumulator value to increase.

The action of the negative feedback ensures that the average modulator output over time equals the input value. In other words, the modulator output is a pulse-width modulated representation of the instantaneous input value. The time domain operation of the simple modulator for DC inputs of 0.25 and 0.53 is shown in figure 3.8.

Figure 3.6 shows the simulated output spectrum of the first order sigma delta modulator

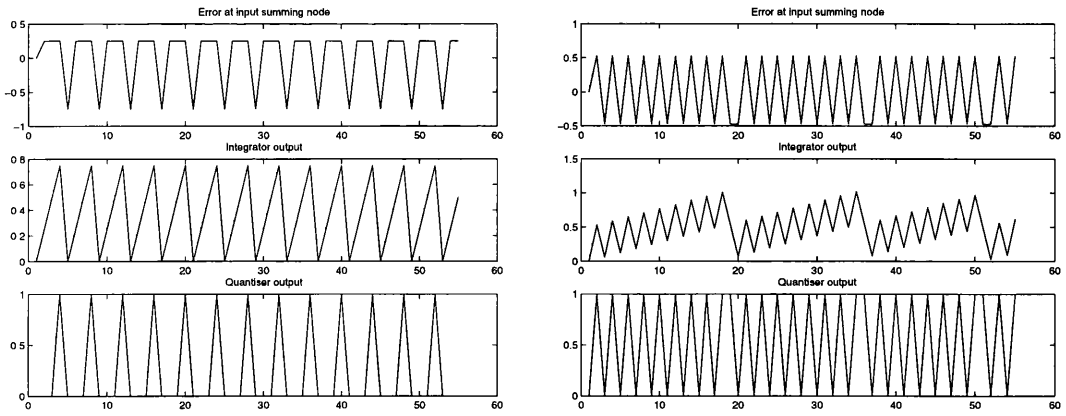


Figure 3.8: First order modulator operation for inputs of 0.25 and 0.53

for 32768 samples with a DC input. The first plot models the quantiser using the approach of figure 3.9. The -20dB/decade integrator suppression on the sigma delta quantiser noise to be clearly seen. The simulation with a real one bit quantiser clearly shows the pronounced periodicity in the output bitstream, caused by the non-linear quantiser element.

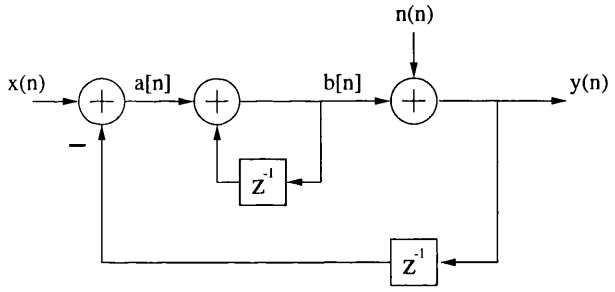


Figure 3.9: Modelling Sigma Delta noise suppression with an additive noise model.

By analysis of the additive noise model of figure 3.9 a discrete-term equation can be written relating the modulator output, Y[n], to the signals at the input node X[n] and the quantiser noise node Y[n]. Z transforms are used to convert from the discrete time domain model to the z-domain frequency response model:

$$a[n] = x[n] - y[n-1] \longrightarrow A(z) = X(z) - z^{-1}Y(z) \tag{3.16}$$

$$b[n] = a[n] + b[n-1] \longrightarrow B(z) = A(z) + z^{-1}B(z) \tag{3.17}$$

$$B(z)(1 - z^{-1}) = A(z) \longrightarrow B(z) = \frac{z}{z-1}A(z) \tag{3.18}$$

$$y[n] = b[n] + n[n] \longrightarrow Y(z) = B(z) + N(z) = \frac{z}{z-1}A(z) + N(z) \tag{3.19}$$

$$Y(z) = \frac{z}{z-1}\left[X(z) - z^{-1}Y(z)\right] + N(z) \tag{3.20}$$

$$Y(z)\left[1 + \frac{1}{z-1}\right] = \frac{z}{z-1}X(z) + N(z) \tag{3.21}$$

$$Y(z) = \frac{1}{1+\frac{1}{z-1}}\left(\frac{z}{z-1}\right)X(z) + \frac{N(z)}{1+\frac{1}{z-1}} \tag{3.22}$$

$$Y(z) = \left(1 - z^{-1}\right)N(z) + X(z) \tag{3.23}$$

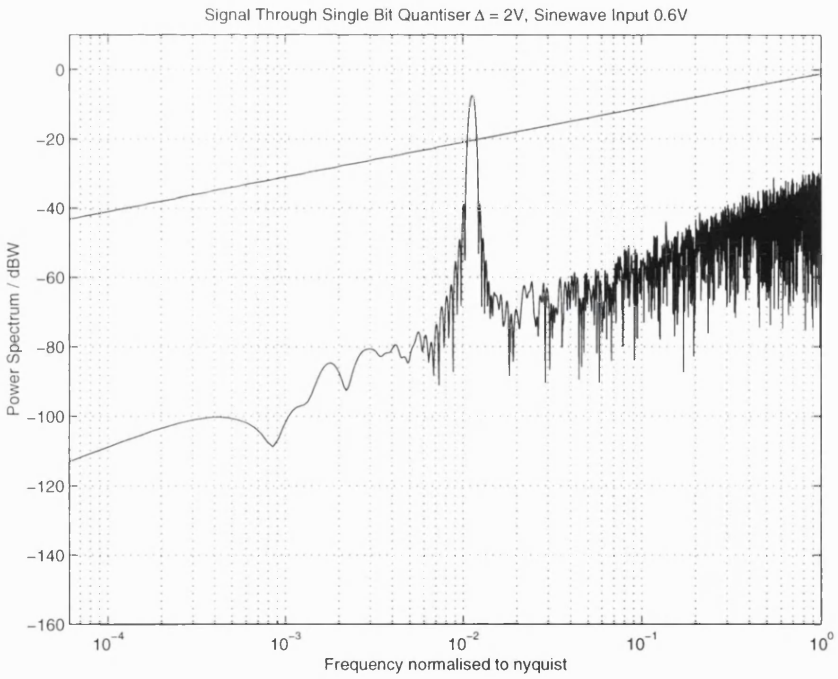


Figure 3.10: First order modulator spectrum compared with Noise Transfer Function.

Equation 3.23 shows that the input signal is sampled and passed through the loop unaltered and the noise at the output is given by $(1 - z^{-1})\,N(z)$. Figure 3.10 shows a simulation of the first order sigma delta modulator using an equivalent random noise source. Input to the modulator is a sinewave of 0.6V amplitude. The noise transfer function of

3.23 is plotted on the same axes confirming the theory. By cascading the integrators before the quantiser much greater high pass filtering of quantisation noise can in principle achieved. For three integrators in series the relation of output signal to the inputs would be:

$$Y(z) = \left(1 - z^{-1}\right)^3 N(z) + X(z) \tag{3.24}$$

In practice this configuration would result in instability due to the quantiser. A practical structure for a third order modulator is that of figure 3.11 where the a and b feedback coefficients will be somewhat more than one to increase the negative feedback and ensure stable operation.

The shape of the noise suppression response is determined from the relative proportions of output fed back to the inputs of the three quantisers, whilst the stability of the system can be controlled by the loop gain. With a totally linear system it is a simple matter to plot the poles of the closed loop system and determine stability. A stable feedback system will have all of its closed loop poles on or within the unit circle of the z plane. Due to the presence of the integrator in the sigma delta loop, traditional linear methods break down and the maximum loop gain for stability is often determined using simulation.

Considering the quantiser as an additive noise source, Masons rule can be applied to the resulting linear system to determine the third order sigma delta noise suppression, or noise transfer function (NTF):

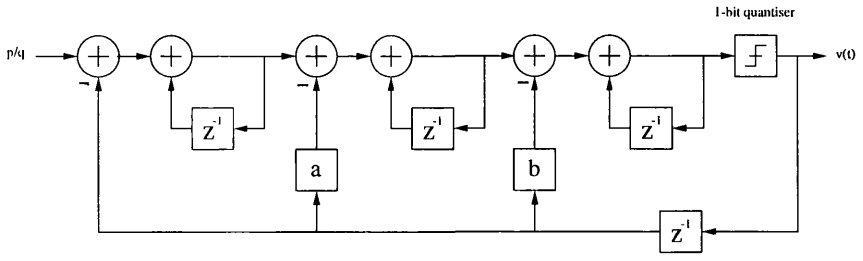


Figure 3.11: Third order sigma delta modulator structure.

$$\frac{Y[z]}{N[z]} = \frac{(z-1)^3}{z^3 - (2+a+b)z^2 + (3+a+2b)z - 1 - b} \tag{3.25}$$

A point worthy of note is that equation 3.25 only allows the *shape* of the NTF to be found. The absolute level of the quantisation noise must be determined by simulation or some other means as the transfer function of the quantiser has been assumed to be one.

Setting a and b equal to one would give a high pass characteristic of 60dB/decade although this is not possible for the stability problems discussed. A real advantage of this modulator is gained from the increased randomisation of the output bit sequence due to the presence of more integrators. The presence of extra integrators further whitens the sequence at the output of the quantiser, leading to increased dynamic range in the conversion of the input signal. Figure 3.12 shows a simulated output spectrum of a third order modulator with a single bit quantiser, which illustrates the reduced discrete tones in the output compared with the lower order result of figure 3.6.

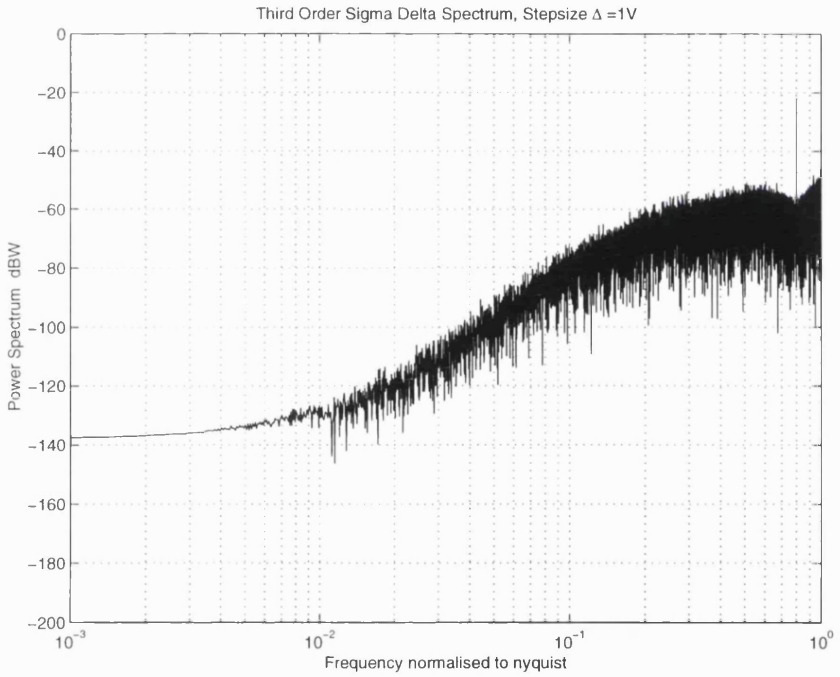


Figure 3.12: Simulated output spectrum of third order sigma delta with single bit quantiser.

3.3 Sigma Delta Output components

All components in the first order sigma delta modulator (figure 3.7) output spectrum can be related to the reference and synthesised frequencies. A model of intermodulation between the two frequencies followed by aliasing allows the frequency of all output components to be predicted. Insight into the formation of the output components can be gained

from studying output spectra with modulator inputs of 0.5, 0.25 or 0.125 and 0.51 shown in figure 3.13.

The figures show the phase modulation spectrum due to applying a first order modulator to switching a dual modulus divider. For an input of 0.5 the output spectrum consists of a single component at half the reference frequency, which is intuitively obvious.

For an input of 0.25 output components occur at a quarter, a half and three quarters of the reference frequency and for an input of 0.125 the trend continues. If the reference frequency is some round product of the synthesised frequency then spacing of the output components is always the reference frequency multiplied by the modulator input.

The spacing and height of the components can be explained as intermodulation between the modulator reference and the synthesised frequency which is some proportion of it. Aliasing of the intermod products occurs due to the sampling action of the digital system although in these first three examples only a few products appear due to the overlap of the aliased higher order products with the non-aliased ones.

The final spectral plot of figure 3.13 has a modulator input of 0.51. This reveals the aliasing of the intermod products more clearly because the aliased products do not overlap the non-aliased ones. The first order modulator output spectrum can be modelled by feeding the reference and synthesised frequencies into a suitable non-linear system followed by a sampler. For the case of the 0.51 input, the first order intermod product is aliased from 0.51 to 0.49 times the reference frequency (98 MHz), and the second order intermod is aliased from 1.02 to 0.02 times the reference frequency (2 MHz). Due to the aliasing the resulting symmetry about the 0.5 value of input results in exactly the same spectrum for an input of 0.49, as there is no reconstruction filter for the fractional divider until the closed loop phase response is applied. Extraction of all of all the intercept points would allow an analytic model for the modulator to be devised.

Figure 3.15 shows the phase modulation spectrum due to the 2nd order modulator structure in figure 3.14 operating under the same conditions used to obtain the first order results shown in figure 3.13. As the order of the modulator increases it becomes increasingly difficult to focus on the individual output components.

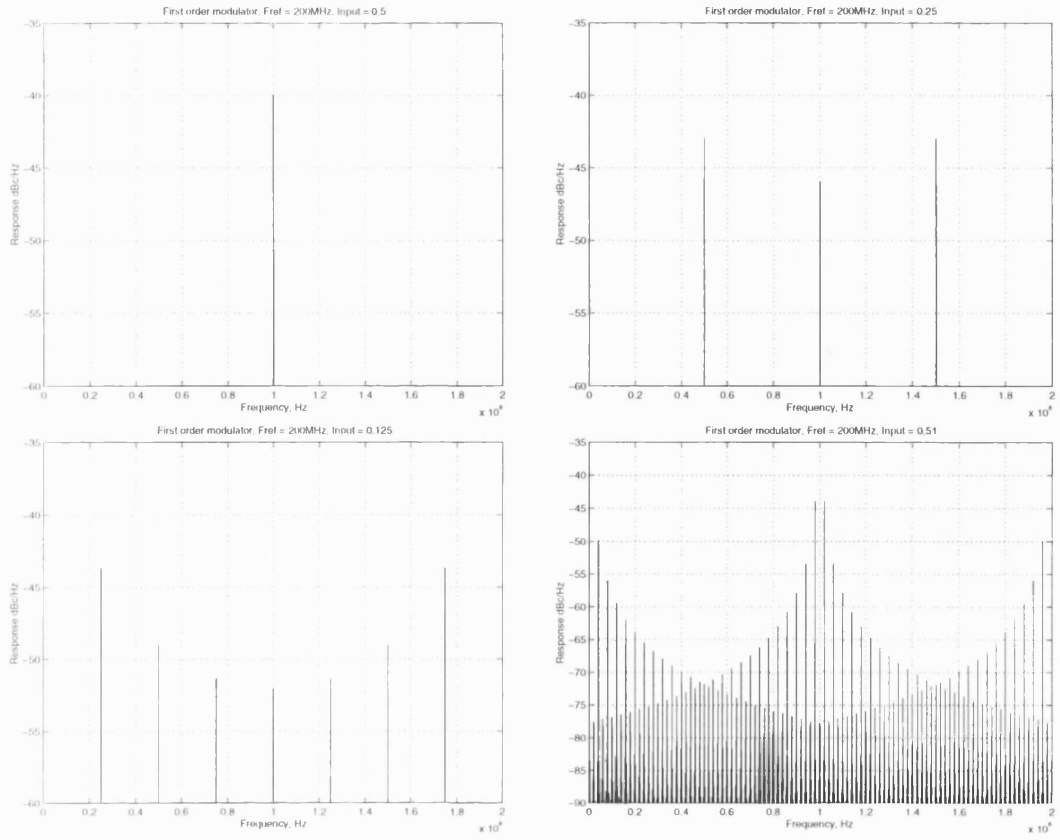


Figure 3.13: First order modulator output component formation.

3.4 Mash Architecture ΣΔ Modulators

As discussed, the third order topology of figure3.11 can only achieve true third order noise suppression if the feedback coefficients a and b are equal to one, but it is not possible to operate the modulator with this feedback as the quantiser causes instability leading to latch up. An alternative modulator topology using multistage noise shaping (MASH) can avoid the problems of latch up and achieve arbitrarily high quantiser noise suppression [5, 9]. Figure 3.16 shows a third order implementation in the MASH topology, with quantisers modelled as additive noise. The diagram shows that this system consists of a cascade of first order modulators with the input to the next stage in the chain being the quantisation error of the previous. The outputs from the three stages D1[Z], D2[Z] and D3[Z] are fed into a combining network which cancels out the quantisation error from all but the last stage.

The Output from the first stage is identical to that of the first order modulator (3.23):

$$D1[z] = \left(1 - z^{-1}\right) N1[z] + V_{in}[z] \tag{3.26}$$

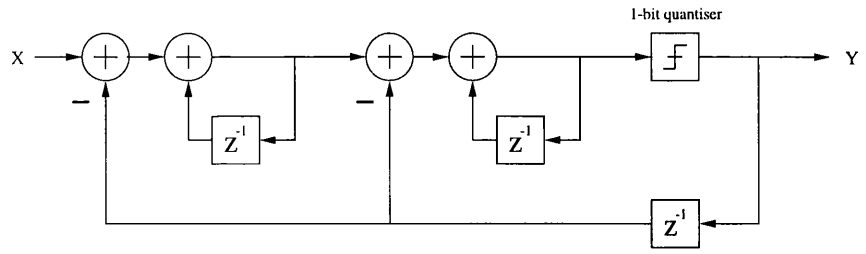


Figure 3.14: Second order sigma delta modulator structure used to obtain figure 3.15.

Where N1[Z] is the noise added by the first stage quantiser.
The output from the second stage modulator is given by:

$$D2[z] = \left(1 - z^{-1}\right) N2[z] - z^{-1} N1[z] \tag{3.27}$$

And that of the third:

$$D3[z] = \left(1 - z^{-1}\right) N3[z] - z^{-1} N2[z] \tag{3.28}$$

The proportion of first stage noise is a positive factor in 3.26 and a negative factor in 3.27 allowing a cancelling factor to be found to remove this term when the outputs of these two stages are combined. This is found by dividing one N1[Z] term by the other:

$$\frac{1 - z^{-1}}{z^{-1}} = z - 1 \tag{3.29}$$

Multiplying 3.27 by 3.29 allows a new output to be produced from stage 2:

$$(z-1)D2[z] = (z-1)z^{-1}N1[z] + (z-1)(1 - z^{-1})N2[z] \tag{3.30}$$

Which when added to the output from stage 1 (3.26) cancels the first stage noise:

$$D1[z] + (z-1)D2[z] = V_{in}[z] + (z-1)(1 - z^{-1})N2[z] \tag{3.31}$$

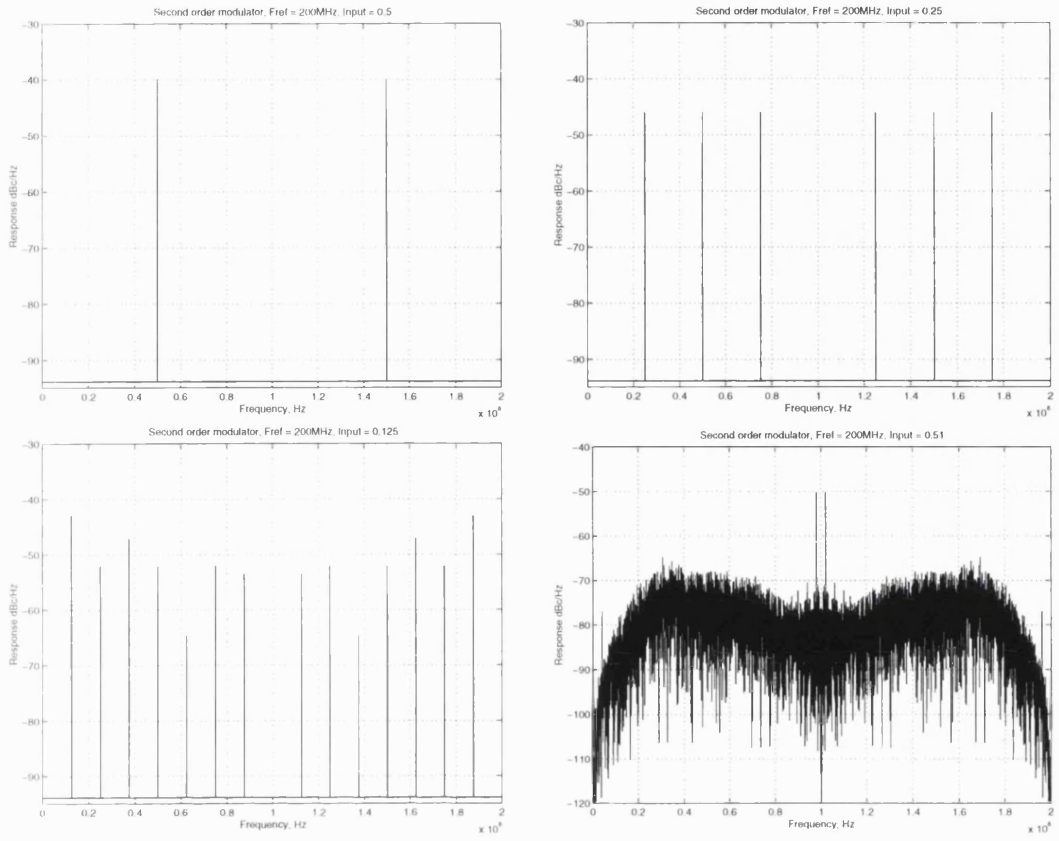


Figure 3.15: 2nd order modulator output components.

Noise from stage 2 can be cancelled in a similar way by comparing 3.31 with the output from stage 3 to obtain the cancelling factor:

$$(z-1)(1-z^{-1})z^{-1} = (1-z^{-1})^2 \tag{3.32}$$

Combining all three outputs derives the overall modulator response:

$$D[z] = V_{in}[z] + (1-z^{-1})^3 N3[z] \tag{3.33}$$

Giving the required third order suppression. The diagram of figure 3.16 includes additional unit delay elements in the combining network to overcome practical difficulties which arise in modelling or hardware implementations.

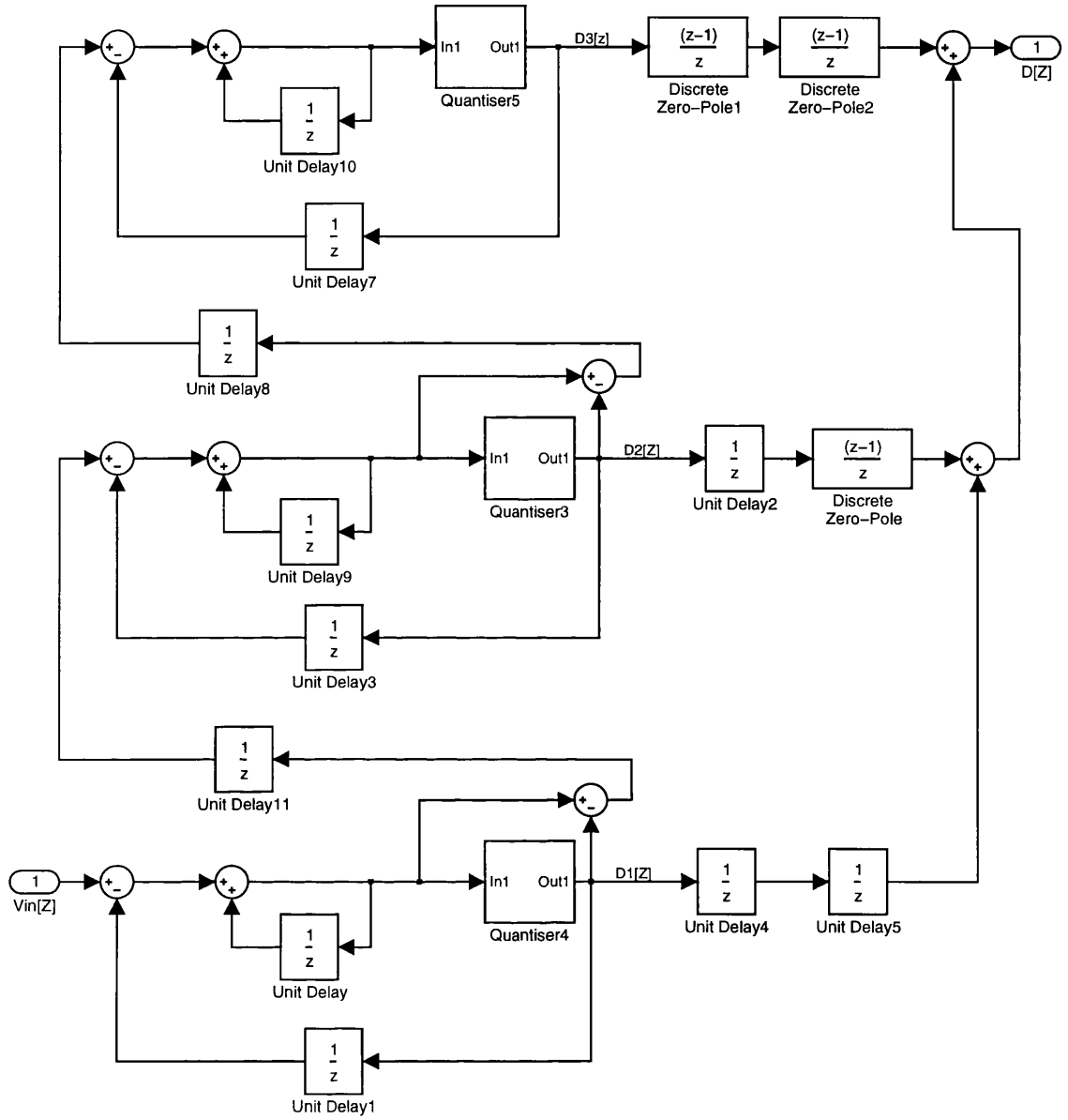


Figure 3.16: A third order Mash structure sigma delta modulator.

From inspecting figure 3.16, a general result for the noise cancelling and delay functions of the Nth stage can be written for an arbitrary order modulator:

$$z^{-(n-1)}(1 - z^{-1}) \tag{3.34}$$

One difficulty with this type of modulator is the multi-bit output. An n-modulator MASH will have an output range of 2^n times the full scale range of the quantiser used in the individual stages. For frequency synthesiser applications this means a multi-modulus divider

is required which can be achieved by a cascade of 2/3 dual modulus dividers. Another approach is to combine the multi bit output in a further sigma delta modulator which itself has a single bit quantiser [5] known as a concentrator. The disadvantage of the latter approach, although it allows dual modulus dividers to be used, is that the concentrator divides down the input range to avoid saturation, which for a third order MASH means only an eighth of the synthesiser output range can be used between N and N+1.

An Nth order MASH composed of n 1 bit modulators has a peak to peak range of 2^n bits at the output:

$$O_{pp} = 2^n * F_{pp} \tag{3.35}$$

Where O_{pp} is the peak to peak output range of the MASH modulator and F_{pp} is the peak to peak output range of each first order modulator in the MASH chain.

Figure 3.17 shows the simulated power spectrum of the third order MASH modulator, confirming the higher order noise shaping, and figure 3.18 show the time domain output. The quantisers used have a single bit output of ± 0.5 with a threshold of zero. The time domain plot confirms the larger amplitude swings due to the three bit output.

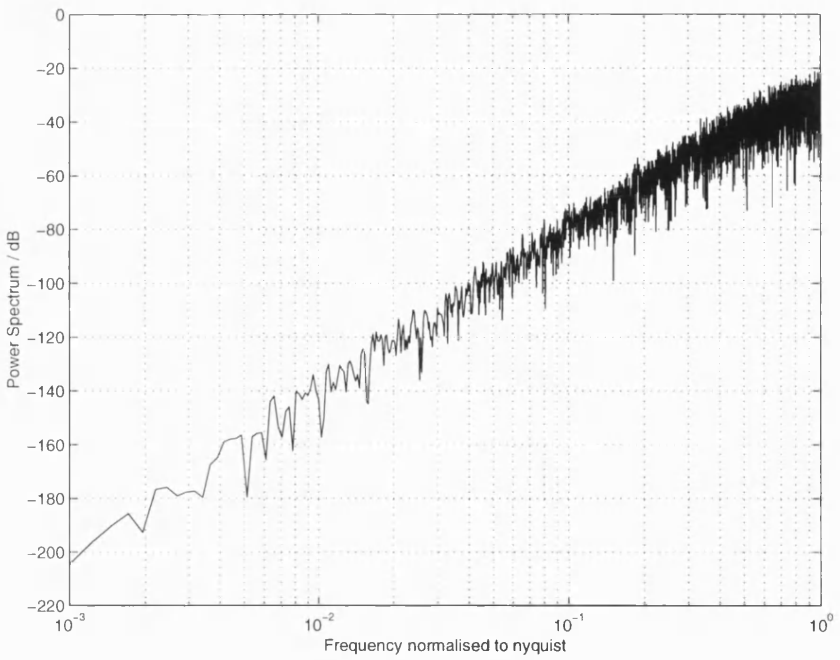


Figure 3.17: Power spectrum of the third order MASH modulator.

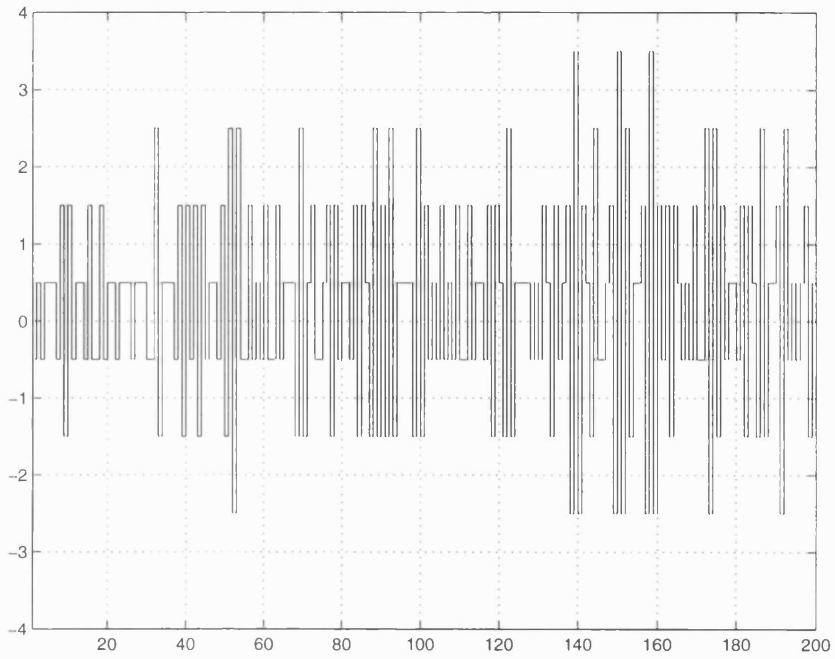


Figure 3.18: Time domain output of the third order MASH modulator.

3.5 Generalised ΣΔ Modulators

All sigma delta modulators can be partitioned into a linear and non-linear part. The non-linear part consists of the quantiser embedded in the sigma delta loop, which may have an arbitrary number of bits. Whilst the linear loop filter gives rise to the signal and noise transfer functions of the modulator. Although the shape of the noise and signal transfer functions can be designed by linear methods these cannot be used to determine the stability of the modulator, which will not be unconditionally stable for filter orders above 2. Stability analysis of nonlinear systems is still an unsolved problem although several approximate or empirical techniques can be successfully applied in practice.

Whilst a linear control loop can be designed by placing the closed loop system poles in stable regions of the s or z plane, the quantiser presence prevents these methods being used directly. To overcome this difficulty recent design methods can use an empirical approach based on simulated stability [10], or the approximate describing function technique which involves linearising the quantiser [11]. Whilst simulation alone does not prove stability and the latter method gives an approximate result these two methods can allow a good starting point to be found for loop filter designs in practice.

3.5.1 Prototype Analog Filter Transfer Functions

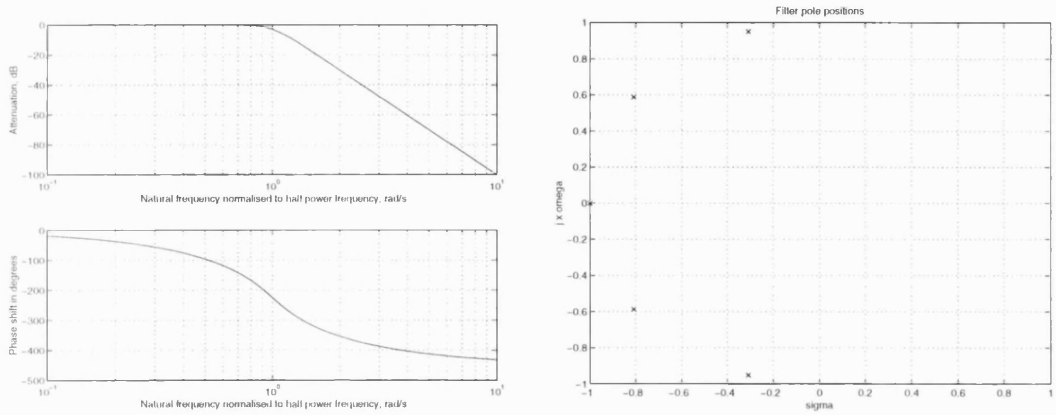


Figure 3.19: Response and poles of a normalised 5th order Butterworth filter.

This section covers the design procedure for the Butterworth analog filter which may be used as a prototype from which to synthesise a modulator loop filter for a desired noise transfer function. Because the sigma delta modulator is a sampled system there will be a two stages of synthesis to realise a desired loop filter from a desired analog NTF. Firstly, a digital filter must be created from the analog prototype as this will accurately represent the modulator response after the mapping of the poles from the s plane to the z plane.

Mapping from s to z plane is generally best performed using a bilinear transform as this ensures a stable analog filter maps to a stable digital filter [12]. Another stage of synthesis will then be required to map this representative digital filter NTF to a digital modulator loop filter topology.

The Butterworth approximation to an ideal low pass filter is optimised for a flat response in the passband rather than steep roll-off. It's transfer function is an all pole type and all of the poles lie on the unit circle [13]. The attenuation of this class of filters is given by equation 3.36 where ω_x is the given frequency, ω_c is the half power or -3dB frequency and n is the order of filter.

$$A_{dB} = 10log[1 + (\frac{\omega_x}{\omega_c})^{2n}] \tag{3.36}$$

A more general set of these filters is described by equation 3.37 where Ω is tabulated below for the different classes of filter.

Lowpass	$\frac{\omega_x}{\omega_c}$
Highpass	$\frac{\omega_c}{\omega_x}$
Bandpass	$\frac{BW_x}{BW_{3dB}}$
Bandreject	$\frac{BW_{3dB}}{BW_x}$

$$A_{dB} = 10log[1 + (\Omega)^{2n}] \tag{3.37}$$

The pole positions of the normalised low pass Butterworth all lie on the unit circle in the s plane and are given by equation 3.38.

$$P_b(K) = -\sin[\frac{(2K-1)\pi}{2n}] + j\cos[\frac{(2K-1)\pi}{2n}] \qquad K = 1, 2, \ldots n \tag{3.38}$$

Equation 3.39 shows the factored form of the transfer function of the 5th order lowpass filter, which can be directly written from inspection of the pole values by realising that the denominator roots are the filter poles.

$$T_{LP}(s) = \frac{1}{(s+1)(s+0.8090 \pm 0.5878j)(s+0.3090 \pm 0.9511j)} \tag{3.39}$$

Transformation of the prototype lowpass to a highpass response can be achieved in the usual was by substituting $1/s$ for s in the lowpass prototype, resulting in a highpass filter with cutoff at 1 rad/s. Equation 3.40 shows the 5th order filter transfer function re-written with the denominator in polynomial form. Whilst equation 3.41is the form of the resulting highpass normalised prototype, still with the half power frequency at 1 radian per second.

$$T_{LP}(s)\frac{1}{s^5 + 3.2361s^4 + 5.2361s^3 + 5.2361s^2 + 3.2361s + 1} \tag{3.40}$$

$$T_{HP}(s) = \frac{s^5}{s^5 + 3.2361s^4 + 5.2361s^3 + 5.2361s^2 + 3.2361s + 1} \tag{3.41}$$

Other forms of analog filter transfer function can be also used to form the prototype noise transfer function, although in data conversion applications the phase response of the filter must be considered due to it's effect on the converted signal, which is irrelevant in Fractional-N applications. The phase response of the target NTF will have a bearing on the phase response of the eventual modulator loop filter, which must be taken into account for stability calculations. Although Chebyshev types of filter generally have a faster rate of roll-off this will only be of true benefit if the loop filter in the main phase locked loop has a similarly inverse low pass response.

3.5.2 Digital prototype filter structure

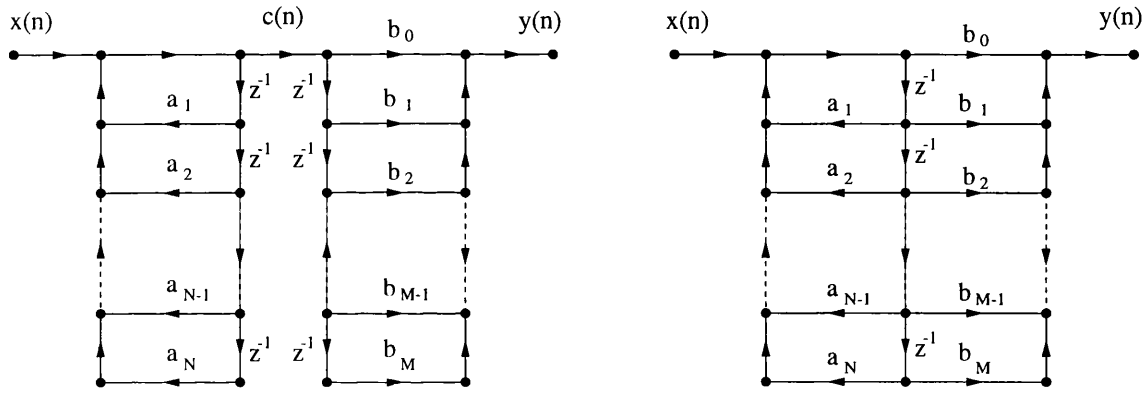


Figure 3.20: The direct form 2 digital filter structure.

The right hand side of figure 3.20 shows the common Direct Form 2 (DF2) digital filter structure [12] which allows simulation of a digital filter with the same response shape as the modulator noise transfer function.

The left side of figure 3.20 shows clearly how the DF2 filter can be split into two separate networks, one of which determines the poles of the system and one the zeros.

The system poles are determined by the first network in the signal path, the transfer function of which can be directly written using Masons rule [14]

$$\frac{C(z)}{X(z)} = \frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{3.42}$$

$$\frac{C(z)}{X(z)} = \frac{1}{1 - a_N z^{-N} - a_{N-1} z^{-N+1} \cdots - a_2^{-2} - a_1^{-1}} \tag{3.43}$$

Factoring the polynomial in the denominator allows the poles p(z) of the system to be found:

$$\frac{C(z)}{X(z)} = \frac{1}{p(z)} = \frac{1}{(z+p(1))(z+p(2))(z+p(3))\cdots(z+p(N))} \tag{3.44}$$

The second system in the signal path determines the system zeros and has the transfer function:

$$\frac{Y(z)}{C(z)} = \sum_{k=0}^{M} b_k z^{-k} \tag{3.45}$$

$$\frac{Y(z)}{C(z)} = q(z) = (z+q(1))(z+q(2))(z+q(3))\cdots(z+q(M)) \tag{3.46}$$

Where q(z) denotes the system zeros.
The complete DF2 system transfer function is given by:

$$\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{3.47}$$

$$\frac{Y(z)}{X(z)} = \frac{q(z)}{p(z)} = \frac{(z+q(1))(z+q(2))(z+q(3))\cdots(z+q(M))}{(z+p(1))(z+p(2))(z+p(3))\cdots(z+p(N))} \tag{3.48}$$

With an arbitrary number of poles or zeros.

3.5.3 Chao Sigma Delta Loop filter structure

Loop filters used in sigma delta modulators must incorporate integration to allow for correct modulator operation. A design method proposed by Chao[10] et al. is similar in structure to the DF2 digital filter, with the exception that the unit delays between filter coefficients have been replaced by delay integrators. Figure 3.21 shows the signal flow graph for this type of loop filter. This network shares some similarity in topology with the DF2 structure, except that between each successive fedback and fedforward node is

an integration in addition to the delay.

This similarity in topology allows the Chao loop filter transfer function to be written by inspecting the DF2 transfer function of equation 3.47 by replacing the delay z^{-1} by an appropriate factor.

The difference equation of a single integrator can be written:

$$y[n] = x[n] + y[n-1] \tag{3.49}$$

Where $y[n]$ denotes the integrator output and $x[n]$ the input. Applying a z transform yields:

$$Y(z) = X(z) + z^{-1}Y(z) \tag{3.50}$$

Which can be written as the discrete transfer function:

$$\frac{Y(z)}{X(z)} = \frac{1}{1-z^{-1}} = \frac{z}{z-1} \tag{3.51}$$

Including an additional delay adds an additional factor of z^{-1}, which means the following transformation must be performed on the DF2 transfer function to yield the loop filter transfer function:

$$z^{-1} \Longrightarrow \frac{1}{z-1} \equiv z \Longrightarrow (z-1) \tag{3.52}$$

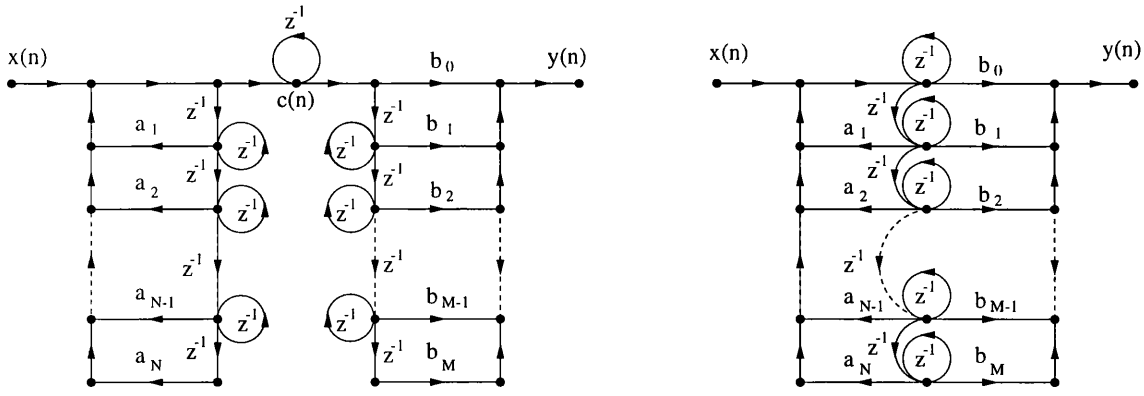


Figure 3.21: Generalised loop filter structure for sigma delta modulators.

Thus re-writing 3.47 under this transformation yields 3.53, the transfer function of the Chao sigma delta loop filter.

$$\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k (z-1)^{-k}}{1 - \sum_{k=1}^{N} a_k (z-1)^{-k}} \tag{3.53}$$

Designs using this form of loop filter have determined modulator stability using the empirical approach of Lee et al [10], stating that the out of band NTF gain of the closed loop system should be less than two for stability under conditions of zero input:

$$|NTF(z)| < 2 \qquad for: \qquad z = e^{j\theta} \tag{3.54}$$

This formula, based on simulated results, is widely used in the design of high order sigma delta modulators [15]. Intuitively it can be understood that at some point, when the out of band noise transfer function has too high a gain, then the out of band noise will be amplified such that the modulator will go unstable.

3.5.4 Stikvoort Sigma Delta Loop Filter Structure

This type of loop filter, designed by Stikvoort [16], does not rely on the initial design of a prototype analogue filter transfer function and is described by the following transfer function:

$$G(z) = \frac{(z-a)^n}{(z-b)^n} - 1 \tag{3.55}$$

The filter can be of an arbitrary order and places all poles and zeros of the loop filter at the same positions. For a low pass loop filter suitable for fractional N sequence generation all poles will be at DC (b=1) whilst the zeros are made as high as possible in order to achieve the best possible shaping of quantisation noise away from DC. If the zeros are made too high in frequency instability of the modulator will result.

In order to ascertain stability the approximate describing function by Engelen [17] can be used, based on results for the phase uncertainty of a sampled quantiser [18]. Whilst the use of the describing function technique [11] is approximate as it does not account for harmonics at the quantiser output it can yield very useful results for closed loop systems with low pass filtering in the loop, as this significantly increases the accuracy of the approximation.

The Stikvoort filter will be discussed in more detail in chapter 6, where it is used along with the describing function technique in order to yield a starting point for the design of modulator sequences for a stored sequence Fractional-N synthesiser. The use of a sigma delta modulator for such Fractional-N applications invalidates the zero-input assumption of the describing function technique referred to above therefore simulation is used at a second stage of the design process to confirm the stability of the modulator with the inevitable large DC inputs. Large DC inputs are required to generate Fractional-N sequences for ratios far from the N+1/2 value when using a N/N+1 dual modulus synthesiser divider.

3.6 Summary

This section has introduced sigma delta modulators and covered number of topologies which are currently in use in the areas of both Fractional Frequency Synthesis and analog to digital conversion. Because of the large swing in the DC level required at the sigma delta output for synthesis applications nearly all frequency synthesisers have tended to make use of unconditionally stable modulator topologies. These include the first and second order modulators as well as the cascaded MASH modulator architecture. The low order modulators have the disadvantage of poor noise shaping, whilst the MASH structure tends to be hardware intensive or require very specialised multi-modulus dividers.

Whilst the higher order modulator designs commonly applied in audio applications offer very good noise shaping performance, they are difficult to fabricate operating at high reference frequencies due to the limited speed of the digital data path elements. In particular as the resolution of the adders and multipliers is increased the delay due to the combinational logic increases. One way to implement fast multiplication is by barrel shifting elements although this can only be used for multiplication of powers of two, leading to a great restriction on the shape of the noise transfer function. A proposed way of overcoming these limitations is offline generation of the noise shaping sequences.

The following section will discuss the development of a synthesiser system designed to make use of sequences stored from high order sigma delta modulators of the type more commonly used in audio frequency data conversion. The objective of offline generation of the shaping sequences is to realise the advantages of these high order modulators at the higher clock speeds required for use in frequency synthesis applications. A further advantage is that simple divider hardware can be used.

Chapter 4

Prototype Design and Analysis

This chapter details the design procedure used for a prototype synthesiser system using stored sigma delta sequences. The main objective of building the prototype synthesiser was to provide hardware verification of the stored sequence synthesiser system and proof of the theory used in its design. Detailed coverage is given on to the circuitry hardware level and digital timing issues of the design. Following this, discussion is given to the method of simulating the synthesiser output spectrum, in particular the unusual aspects such as modelling of the dual modulus divider effects.

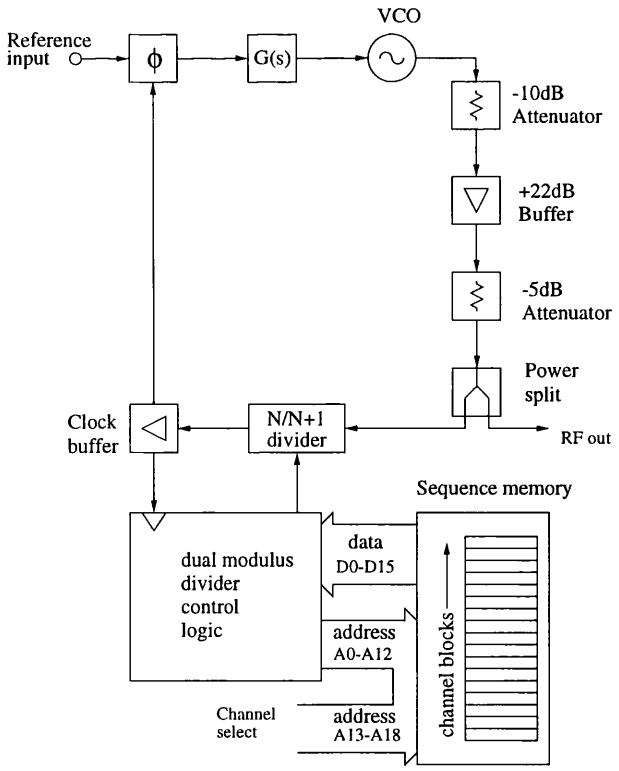


Figure 4.1: System block diagram of prototype synthesiser.

4.1 Prototype System Specification

The design process can be broken up into three main areas, that of the phase locked loop, the digital hardware used to store and read out sequences into the prescaler and the sigma delta modulator used to generate the sequences.

Each subsystem has various implications on the design of the other leading to the requirement for some iteration in the design procedure. As the intention of this work was to provide very high lock speeds with a single PLL the starting point taken was to choose the loop filter natural frequency by satisfying a lock time requirement. This fixed the closed loop response of the PLL whilst allowing variation of the reference frequency as long as the feedback divider is adjusted to compensate.

Following this step a suitable modulator can be selected of a certain reference frequency and order which will satisfy phase noise and spurious requirements with the chosen loop response. After the modulator has been finalised with suitable simulations then the digital and analog hardware can be designed subject to the timing constraints.

Figure 4.1 shows a block diagram of the prototype synthesiser whilst figures E.1 and E.2 show the loop and digital sections of the synthesiser.

The original intention of the synthesiser project was to design a synthesiser suitable for the DCS1800 mobile phone system, but the lack of a suitable prescaler part with a high enough reference frequency reduced the output and reference frequencies of the final design. This reduced performance somewhat whilst still allowing good results to be obtained scalable to systems with different parameters.

Loop Frequency	250kHz
Reference Frequency	156MHz
Output Range	1248 - 1404 MHz
Acquisition time	$10\mu s$ to within 1kHz and $2e10^{-1}$ Radians
No. Channels	80/Megabyte

The number of possible channels covered by the synthesiser depends on the amount of memory in use, whilst the resolution can be made arbitrarily high by lengthening the channel sequence. The length of channel sequence used for the prototype was 128k bits, corresponding to a sequence of length $840\mu s$ with a 156 MHz reference frequency. The

minimum timeslot length is dependent on the modulator spurious free dynamic range (SFDR) with the use of spectral density simulations, a longer sequence offering an improved result.

4.2 Loop Design for Stability

A third order type 2 loop design was selected as the starting point of the loop design process, due to its good reference suppression. The design equations used for this loop filter are 2.64 to 2.66 in chapter 2. The approach taken was to complete the basic filter design and then add further filtering to improve high frequency suppression performance without affecting loop dynamic performance or stability.

The component values for the filter were calculated simultaneously with plotting the open loop response using the Matlab program in the appendix 4.6. This program plots the open loop response from chapter 2:

$$\frac{\phi_o}{\phi_i} = \frac{K_p K_v G(s)}{Ns}. \tag{4.1}$$

The open loop response plot, figure 4.2, confirms the optimisation of the loop filter response from a stability point of view, the maximum phase margin of 60^o occurring at the zero gain frequency. The system phase margin was chosen as 60^o to allow some variation in the hardware loop parameters whilst maintaining stability.

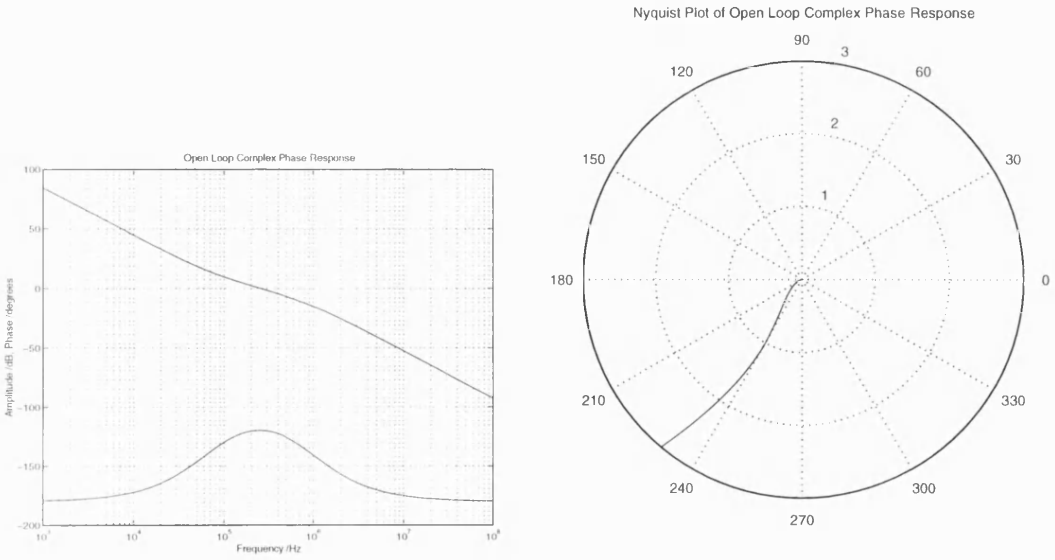


Figure 4.2: Magnitude and Phase of Open Loop Complex Phase Response.

The loop components have to be selected for optimum phase margin at a certain frequency as both the VCO gain and N divider value vary with the synthesiser output frequency. The starting point for the divider value was chosen as 8.5 corresponding to the centre of the band, whilst the nominal value of VCO gain was taken to be 36 MHz/V. With the 60^o phase margin and 250 kHz loop frequency this results in the actual component values of figure 4.8.

4.2.1 Enhanced Loop Filtering

The splitting of R1 and introduction of a capacitor from this node to ground adds an additional pole to the filter input [1], the response of which is described by:

$$H(s) = \frac{\omega_o}{s+\omega_o} \text{ , where } \omega_o = \frac{4}{R_1 C_4}.$$

The additional corner frequency is selected to give an additional phase shift of no more than 10^o at the loop frequency. The cutoff frequency of the extra section is calculated from the allowable phase shift at the loop filter frequency, which will be the degradation in phase margin and hence loop stability:

$$\phi = \arctan \omega\omega_o, \tag{4.2}$$

$$\omega_o = \frac{\omega_n}{\arctan(\frac{10\pi}{180})}. \tag{4.3}$$

For the 250kHz loop frequency this results in ω_o = 1.4 MHz. And a value for C4 of 187pF. 4.3 confirms the effect of the extra pole on the open loop filter response, and the degradation of the phase margin to 50^o.

Apart from additional filtering at the op-amp input an even higher frequency first order section was placed between the op-amp and VCO. The objective of this was to provide even greater loop suppression at frequencies much higher than the loop filter design frequency. Figure 4.4 shows the configuration of the extra filtering at the output of the op-amp. The values of C3 and R3 are calculated as before with attention to the phase shift at the critical zero loop gain frequency. The response of this single pole is given by

$$H2(s) = \frac{1}{sRC+1} \tag{4.4}$$

Where the phase shift is given by

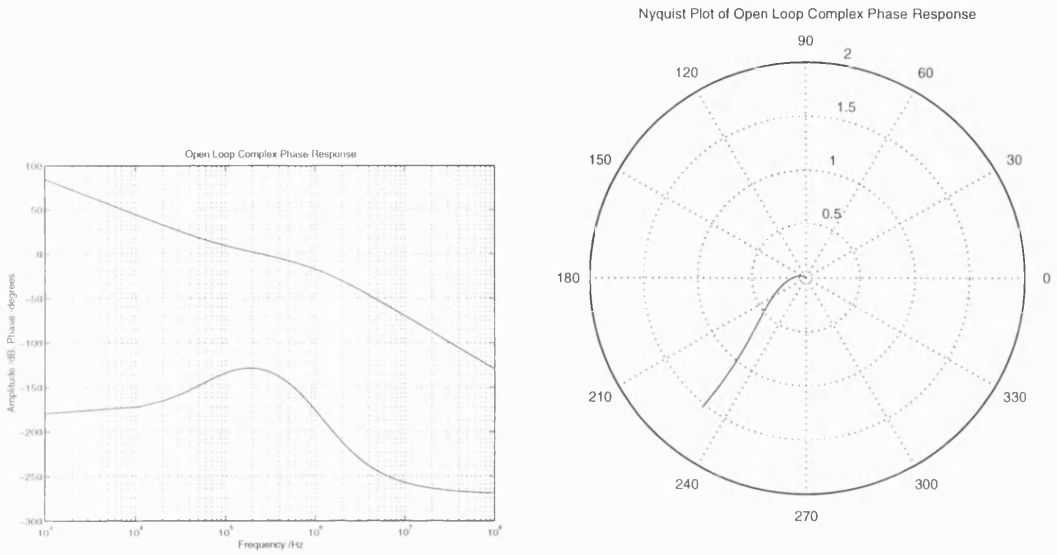


Figure 4.3: Open Loop Gain with additional Pole at Op Amp Input

$$\phi 2(s) = -\arctan(\omega RC) \tag{4.5}$$

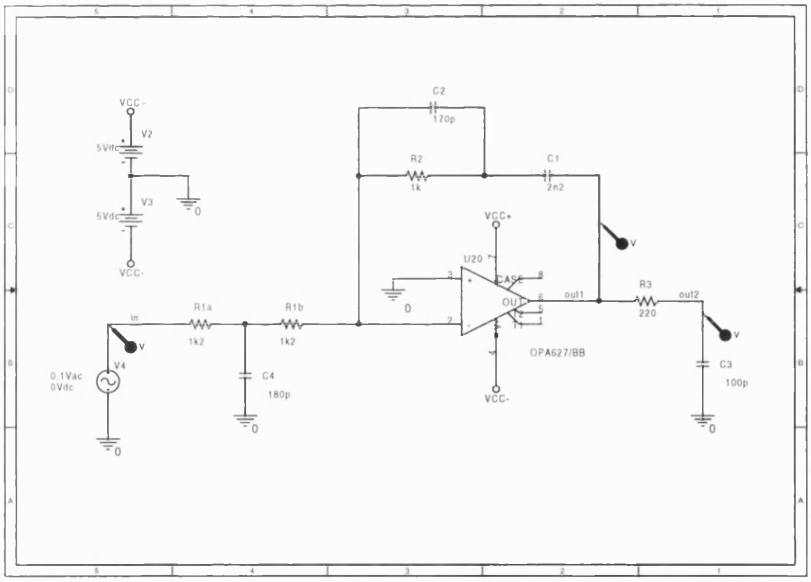


Figure 4.4: Schematic for Enhanced Loop filter.

Allowing a phase shift of 2^o at 250kHz, and choosing C3 as 100pF results in an actual value for R3 of 220Ω. Figure 4.4 shows the schematic of the final filter circuit whilst figure 4.5 shows a Spice simulation highlighting the effect of this additional pole. Although the extra phase shift at 250kHz is very small, the filter suppression is greatly improved

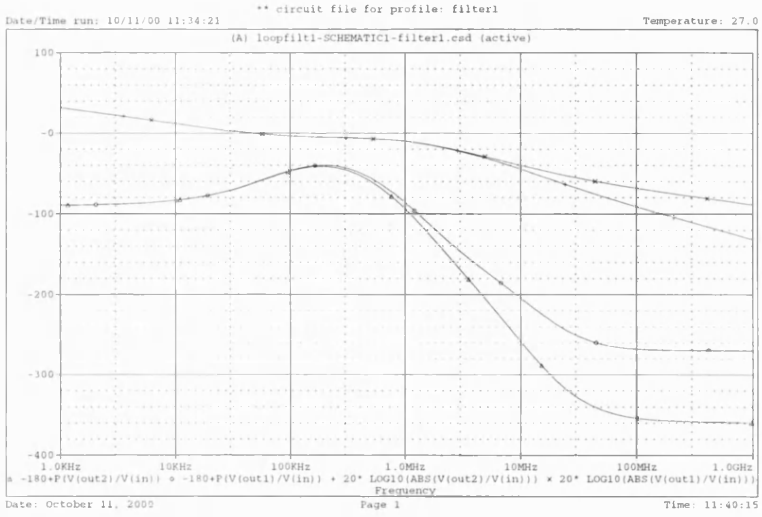


Figure 4.5: Simulation of enhanced loop filter showing the effect of additional R3 C3 pole.

above 100MHz which aids reference spur suppression. Figure D.4 shows the closed loop response, confirming a 40dB/decade roll-off.

Transient response of the synthesiser was modelled using equations 2.74 and 2.83 detailed in chapter 2. These give the transient response in terms of both phase and frequency for a step of frequency at the synthesiser input. Simulation of the transient response was achieved using the code of D.3 and figure 4.7 shows the results for frequency and phase for a simulated 10 MHz step in frequency. At the loop output a 10 MHz frequency step is possible to within 1 kHz and 2e-3 radians within 10 μs.

4.3 Circuit Description

Figure 2.5 shows the basic loop filter configuration used to implement the third order type 2 loop. As discussed, additional filtering was added to the basic configuration with poles placed far enough from the main filter response poles to avoid any effect on stability.

A Burr Brown type OPA627 op-amp was used for the loop filter. The plot of figure 4.9 highlights the need for the additional filtering due to the effect of a non-ideal op amp on the loop filter response. The ideal lines show a pole-zero simulation of the open loop response whilst the lines labeled spice1 indicate the same open loop response incorporating a Spice simulation of the op amp with the manufacturers model. Instead of a -40dB/decade roll off at high frequencies the simulation indicates a levelling off of

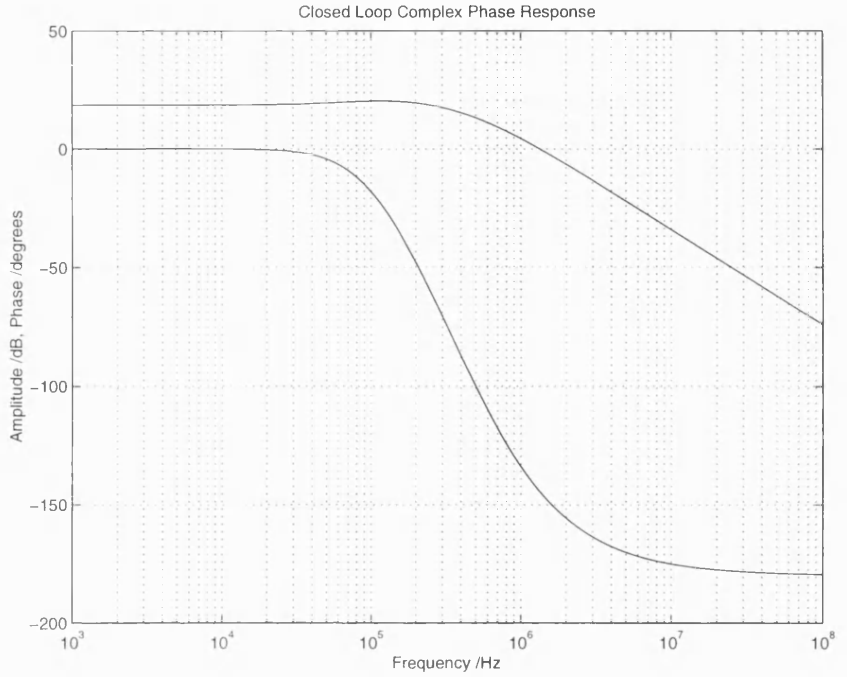


Figure 4.6: Magnitude and Phase of Closed Loop Complex Phase Response.

the suppression due to the effect of increasing frequency on the op amp response. The 'Spice2' lines on figure 4.9 show the Spice simulation of the final loop filter with all extra poles added, confirming the improved high frequency suppression.

The phase detector used was a Motorola type MCH12140. This device was chosen as it has an 800 MHz toggle frequency, the highest of any off the shelf unit available. For the very high speeds involved mixers were considered, but abandoned in favour of the improved capture range of the sequential detector, mixer phase detectors also require extra acquisition circuitry in order to sweep the VCO across its range to allow natural capture to take over. The nominal ECL levels are translated to PECL levels with the positive supply voltage, giving V_{ol}=3.2V and V_{oh}=4V. This leads to a phase detector gain of $0.8/(2\pi)$ or 0.127 Volts per radian.

An HP8642B signal generator provided a flexible reference source for the prototype, AC coupled into the phase detector at 0dBm. A resistive potential divider biases the coupled input signal to the centre of the PECL voltage range at 3.6 volts, an identical arrangement being used to couple the divider output signal to the phase detector.

A standard z-comm VCO, type V605ME01 SN-00190495 was used. Referring to figure E.1, attenuator and buffer amplifiers are used to provide isolation of the VCO output.

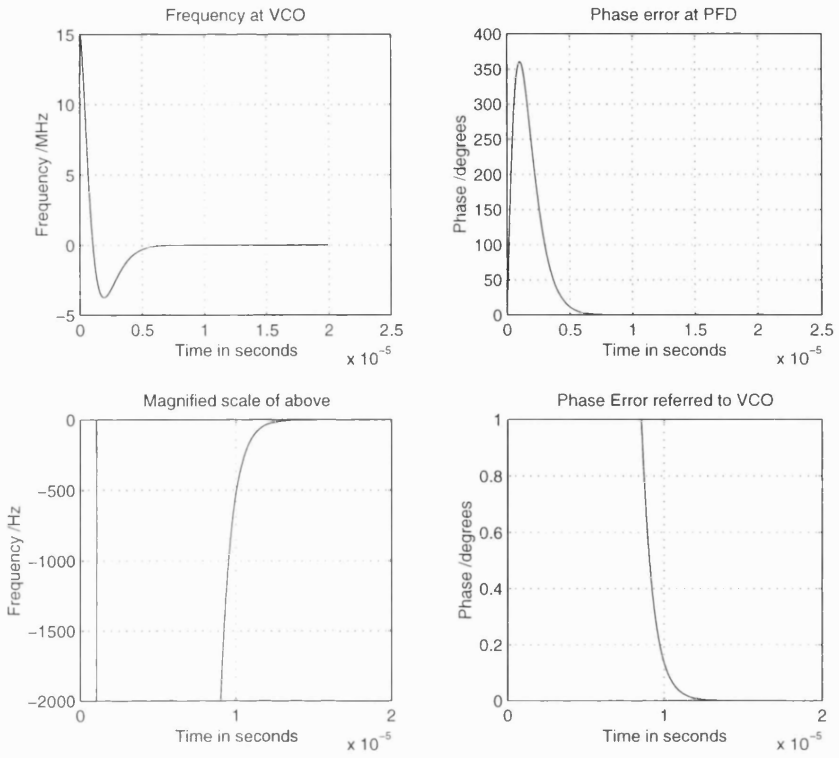


Figure 4.7: Synthesiser Transient Simulation for a 15 MHz Frequency Step. f_n = 250 kHz, Phase Margin = 53.2^o.

This prevents frequency pulling and any resulting increase in the synthesiser phase noise. The type MGA-865 MMIC buffer amplifier has 21 dB of gain from 1.5 to 2.5 GHz and a reverse gain of -40 dB. Figure E.1 shows the decoupling and power supply arrangements used for this device. R_{30} is used to flatten the Q of the inductor and prevent parasitics whilst L_2 (28nH) provides some degree of input matching. Aside from additional isolation, the MMIC is also required in order to boost the power level allowing the split signal to drive the prescaler input and provide a reasonable power output for the synthesiser.

A Motorola MC12026A ECL dual modulus prescaler was used due to its low division ratio of 8 or 9, allowing a high reference frequency implementation to be tested to see the benefits on the $\Sigma\Delta$ shaping. Many higher input frequency parts were available but these had division ratios in the 32/33 range, which would have lost the SFDR performance advantages of high reference frequency $\Sigma\Delta$ operation. Use of a lower reference frequency would also have led to greater phase detector noise at the loop output.

The MC12026A has a quoted toggle frequency of 1.1 GHz but was found to operate reliably up to 1.4 GHz without missed counts. Input and output levels were the usual PECL

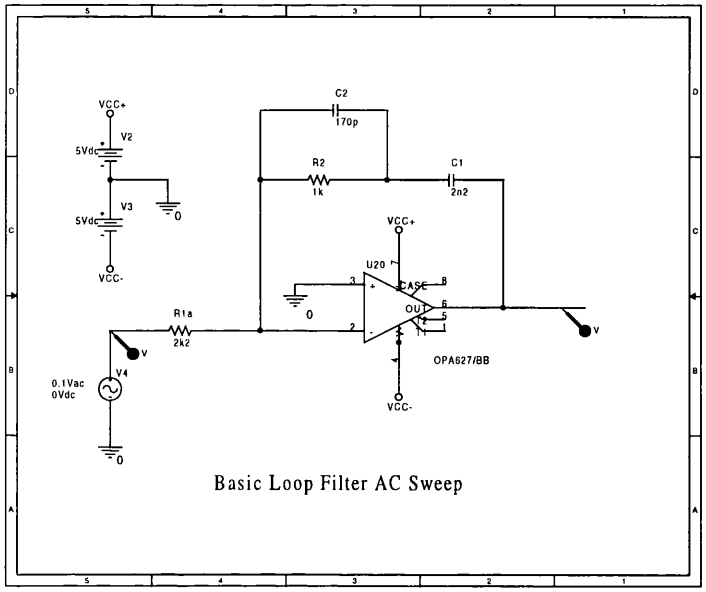


Figure 4.8: Spice Simulation Setup for Basic Loop Filter Configuration

ones with a swing of 800 mV centred on 3.6 V and a similar AC coupling arrangement was used to that for the phase detector. The input sensitivity at 1.1 GHz was quoted as 100-1000 mV_{pp} and the typical modulus setup time quoted as 6 ns. This meant that the toggle frequency would be at the limits of the device performance in this application.

4.4 ΣΔ Loop Design

A third order modulator was initially chosen in order to test the stored sequence synthesiser concept. The topology used was that of figure 4.11, consisting of three integrators placed in a feedback loop with feedback of the delayed output at the integrator inputs. It is necessary that there is unity gain magnitude between the output and the input node for the conversion gain of the modulator to be unity, therefore the coefficients A and B were adjusted in optimising the modulator noise spectrum and ensuring stability.

The reason for choosing this type of modulator was due to the increased noise shaping performance over the second order type and the possibility of using a two dimensional swept optimisation procedure of the a and b feedback coefficients. Although swept optimisation of such a modulator is somewhat computationally intensive the reasonably small number of variables for this modulator made this a viable proposition, allowing a large number of simulations to be performed and the optimal result chosen in terms of the maximum simulated synthesiser SFDR consistent with a stable modulator output.

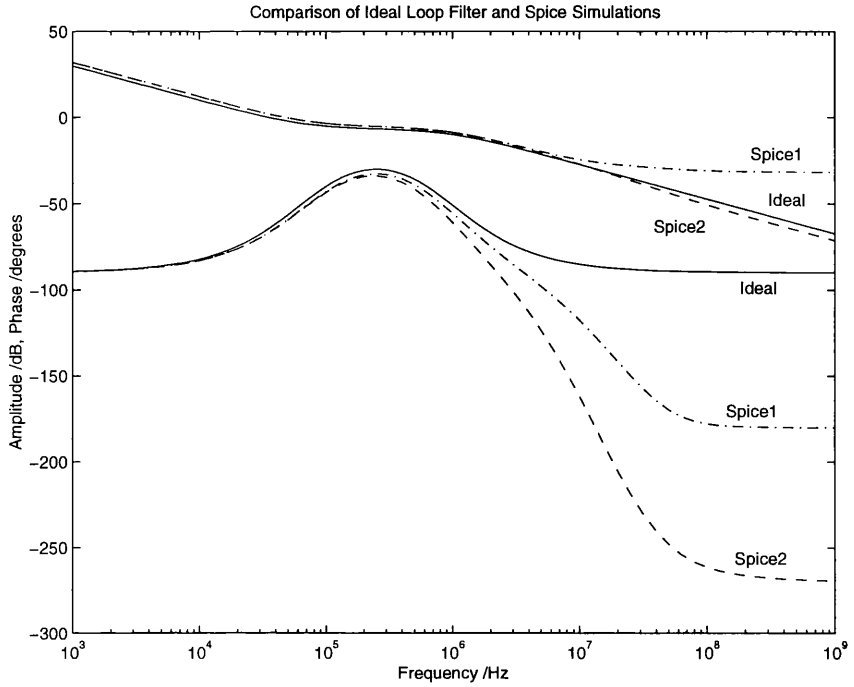


Figure 4.9: Comparison of Ideal Loop Filter and Spice Simulations

The starting values of the modulator coefficients a and b were initially chosen empirically as 8 and 9 for stable running of the modulator. It was found that small changes in the values of these coefficients of the order of 1% could result in large change in the worst case spur levels at the modulator output without appreciably affecting the overall shape of the noise transfer function, consequently these coefficients were swept from 8 to 9 and 9 to 10 respectively in steps of 0.01 and the highest SFDR stable result of the 2D sweep selected for the final sequence.

4.4.1 Modulator Topology and Simulation

The modulator is simulated over 131072 samples to represent a sequence length comparable to a DCS1800 time slot. Shorter sequences can be used but result in decreased noise shaping performance near to DC due to the spectral leakage which occurs. Figure 4.12 shows a simulation result for this amount of samples, with the single bit quantiser step size normalised to 1 volt. This result is typical of the shape obtained across the band after optimisation. Although it was found that the effect on the close-in noise shaping due to the sequence truncation was dependent on the modulator input it was found possible to obtain good results across the band by re-optimisation of the coefficients. It was also found important to allow a startup sequence of at least 1000 samples in order to remove

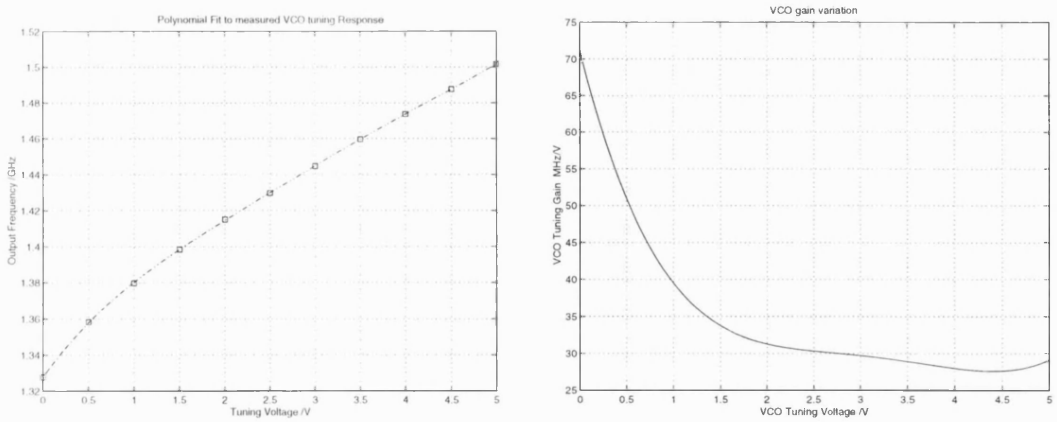


Figure 4.10: Curve Fit to Measured VCO tuning Response and Calculated VCO Tuning Gain.

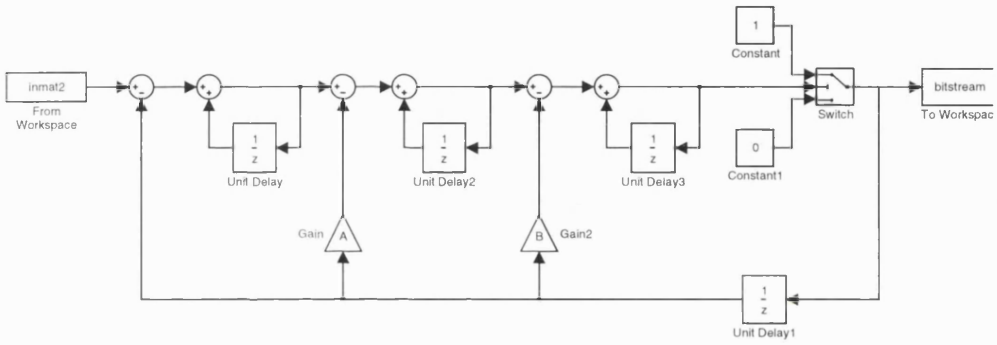


Figure 4.11: Third order modulator topology used for prototype sequence generation.

the unwanted effects on the output spectrum.

Figure 4.12 was produced by the simulink file third.mdl the schematic of which is shown in figure 4.11. This result was called by the matlab file modsim.m included in appendix section D.7. The effect of the truncation of the sequence is evidenced in the spectrum of figure 4.12 by a gradual levelling off of the noise spectrum at low frequencies. Shortening the sequence further results in extension of the whitened portion of the noise to higher frequencies and a rise in the RMS power of the whitened section of the noise. Figure 4.13 shows the same modulator simulation using a Hanning Window in order to smooth out the edges of the simulated sequence in order to more closely predict the output spectrum of the same modulator implemented as free running digital logic. The sequence truncation is not problematic if the sequence is long enough to reduce the low frequency sequence noise to just below the system mask specifications.

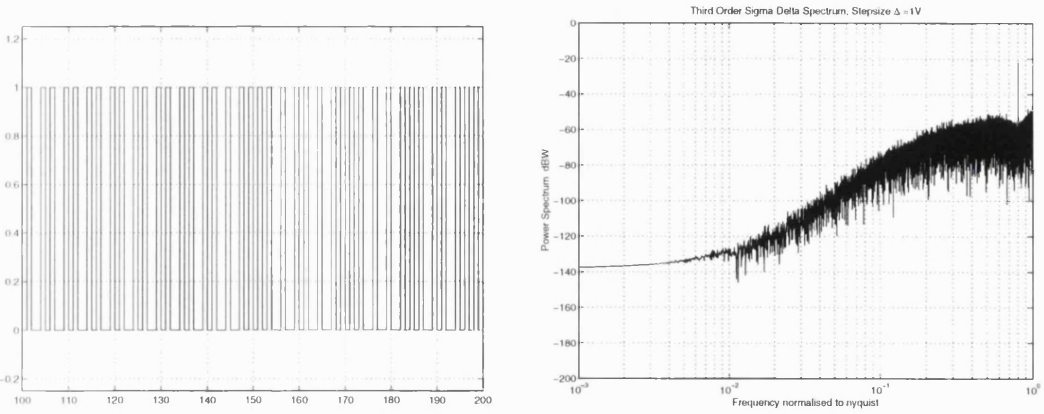


Figure 4.12: 3rd Order Modulator Spectrum.

4.5 Digital Hardware Design

The function of the digital section of the hardware is to drive the dual modulus divider according to a variety of digital sequences stored in memory for each required synthesiser channel. The input to the logic is a wide word of data latched from successive memory locations at a maximum rate dependent on the memory access time, which will be considerably below the reference frequency of the synthesiser. The output of the logic is a single bit sequence delivered at a rate corresponding to the reference frequency, allowing the dual modulus divider to be switched every reference cycle.

The logic is required to perform a variety of addressing, latching and multiplexing operations in the correct sequence to deliver the contents of the memory. The sequencing of all the operations is controlled by decoding the count value of a master clock and the length of the count of this master clock corresponds to the width of the word latched from the memory. This section will give a detailed description of the functioning of this logic and how the various subsystems work together.

4.5.1 Digital Circuitry Description

Figure E.2 shows the board level design of the digital section of the circuit. This corresponds to the lower part of the system block diagram of figure D.1 showing the arrangement of memory and complex programmable logic device (CPLD) in the system. This section will describe the operation, structure and timing of the internal logic of the CPLD device.

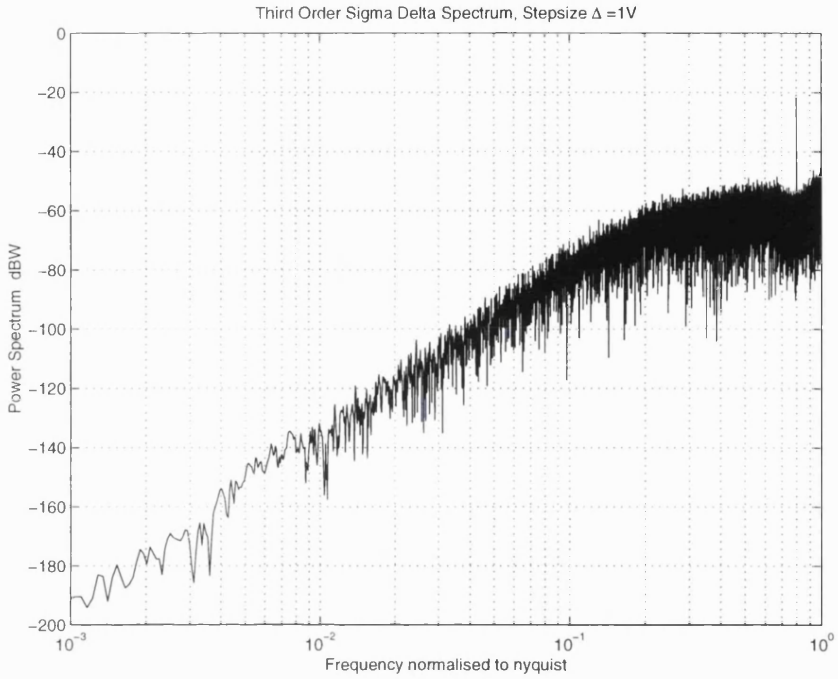


Figure 4.13: Hanning Windowed 3rd order Sigma Delta Bitstream.

Referring to figure E.2, the main components are an Altera (CPLD) type EPF10K30 and an 8 Megabit type 28F800B AMD Flash ROM to provide non-volatile storage of the bitstreams for each required synthesiser channel. Other components include an Altera serial Flash ROM type EPC2 to provide non-volatile backup of the data for the CPLD configuration SRAM. The byteblaster socket is chained through the CPLD and FLASH to provide a serial interface for download of the configuration data. The master clock signal for this circuit is derived from the output of the dual modulus divider and fed to global clock pin No. 125 on the Altera device. The single bit output from the CPLD is able to directly drive the TTL level modulus control pin of the dual modulus divider.

4.5.2 Behavioral Description and Modelling

Figure 4.14 shows the internal logic of the PLD circuit and figure 4.19 shows a behavioural simulation at 100 MHz before mapping to hardware. This logic performs serial readout of the bitstream data contained in blocks of memory 8192 words long. The particular block of data in memory for a given channel is selected by the upper address bits on the memory, allowing up to 62 channels to be stored in this particular memory device. Continuous rather than frequency hopping usage of the synthesiser uses a loop of the same block of stored sequence in the memory. The rest of this section describes the

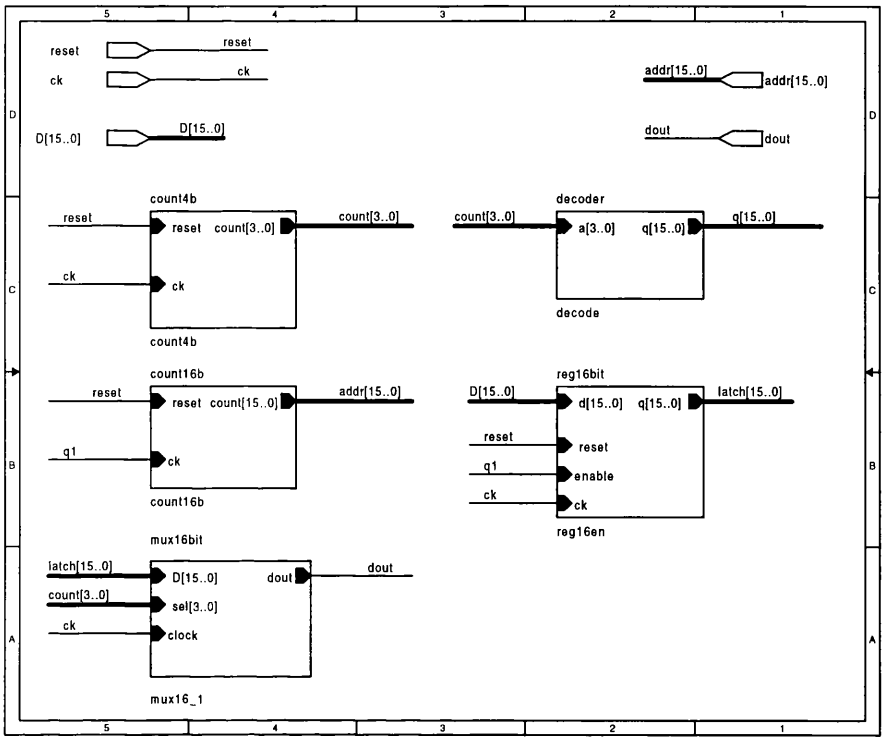


Figure 4.14: Top Level Logic Diagram of Prototype Synthesiser.

operation of the logic from a register transfer level.

Referring to the logic diagram, entity count4b is clocked from pin ck which is clocked externally by the synthesiser reference. This subsystem is a four bit counter with output on the bus count[3..0]. Entity decode is an address decoder driven by this four bit sequencing bus which provides 16 decoded outputs (q[15..0]) for the four bit clock. Only one of these decoded outputs is used, q1 is used.

This decoded output q1 controls the latching of the 16 bit register entity reg16en and also serves to increment the memory address counter count16b at the same instant. In this way the maximum time is available for memory access and the total speed of the circuit is maximised. The enable input of the entity reg16en is used to control latching rather than the clock input itself in order to avoid the problems of clock skew as much as possible.

The output of address counter entity count16b drives the 16 external pin resources addr[15..0] which are externally connected to the Flash memory address bus. As this memory is enabled all the time by a hardwiring of all the enable pins, the new data will be available in the minimum possible access time. The new memory data appears on the incoming PLD

t_S	Setup time for a DFF
t_H	Hold time for a DFF
t_A	Memory access time
t_{MX}	Mux combinational logic delay
t_{TR}	Worst case wire transport delay
w_D	Width of memory data bus
w_A	Width of memory address bus
w_I	Width of incrementing address
w_C	Width of channel bus
f_{CLK}	Maximum clock frequency
s_B	Size of channel block in bits
s_M	Size of memory in bits
n_C	Number of channels

Table 4.1: Parameters for the logic design.

pins D[15..0] for latching by reg16en.

The new memory word is latched on a single clock cycle and appears on the internal bus names latch[15..0] for the next cycle of the MUX entity mux16_1. This mux is also controlled from the four bit internal bus count[3..0] and the new bit of data appears at the mux output after two clock cycles of latency due to the MUX internal pipelining. The single bit synchronous output sequence is fed to the external pin dout to drive the dual modulus divider state.

4.5.3 Hardware timing considerations

Timing requirements of the logic design depend on the synthesiser reference frequency and the speed and data width of the sequence memory. The available speed depends on both the memory speed and width and the performance of the target technology. Table 4.5.3 describes the various parameters relating to the design process.

Maximum memory word latching frequency is dependent upon the memory access time, wiring delays and the setup and hold times of the D register to hold the latched word and can be described by the following equation:

$$f_{LAT} = \frac{1}{t_A + t_S + t_H + t_{TR}} \tag{4.6}$$

This puts one limit on the maximum achievable clock frequency of the circuit, as the maximum data rate is the latching frequency f_{LAT} multiplied by the memory word width w_D:

$$f_{CLK} \leq \frac{w_D}{t_A + t_S + t_H + t_{TR}} \tag{4.7}$$

In practice in both of these definitions the setup and hold times will have to be exceeded by a margin to the nearest clock cycle period.

As well as the memory data rate, the maximum clock frequency is also constrained by the logic delays in the fastest clocked portion of the circuit, in this case the multiplexer which switches at the reference frequency rate. Two kinds of delay affect the maximum speed at which the MUX can be reliably operated, the combinational logic delay between the successive D register stages and the propagation delay of the D registers themselves. The combinational logic delay is the selection logic of the MUX and will be dependent on the delay through the AND gate primitives forming the selection elements and the width of the MUX, which determines the number of series primitive delays. These constraints are summarised in the following equation:

$$f_{CLK} \leq \frac{1}{t_{MX} + t_S + t_H + t_{TR}}. \tag{4.8}$$

Where t_{MX} is the combinational delay due to the MUX logic and the setup and hold times must be met to the nearest clock cycle of reference.

For a frequency hopping system it may be desirable to use a stored memory sequence at least as long as the transmission time slot. The length of sequence relates to the memory storage requirements and the width of the memory address bus w_A used to increment within the channel block, or the width of the increment address w_I. The size of the channel block s_B relates to the width of the increment bus w_I and the data bus w_D:

$$s_B = \mathrm{w}_I \mathrm{w}_D. \tag{4.9}$$

The total size of memory also relates simply to other parts of the topology:

$$s_M = s_B n_C = \mathrm{w}_D(\mathrm{w}_I + \mathrm{w}_C). \tag{4.10}$$

By widening the data at the PLD input to 16 bits the data rate falls to a sixteenth of the rate of the reference frequency, 12.5 MHz for a 200 MHz frequency. This allows non-volatile memory to be used which has a longer access time. An alternative arrangement would involve a slow EPROM and fast SRAM with a cache between the two.

4.5.4 Timing Simulation

Figure 4.16 shows the behavioural logic description mapped to the internal hardware resources of the Altera logic device, whilst 4.15 shows the simulation incorporating the real path and propagation delays in the 10K30 device. The maximum clock frequency possible from the simulations was 220MHz.

The generic VHDL logic diagram of figure 4.14 was synthesised to an Altera Programmable logic device (Altera EPF 10K30 ETC144-1 series) using the vendors own tools. This resulted in a design of the same functionality as the register transfer level VHDL code but implemented in the architecture-specific macros available in the CPLD device.

The vendor CPLD software performed the place and route in the device, selecting where each logic entity would reside in the device hardware resources. After place and route is performed the extra timing delays due to the wiring between logic blocks and device pins can be back-annotated to the timing simulation to provide greater accuracy.

Referring to figure 4.15, below the clock signal the D[15..0] bus is the 16 bit input from the external memory. Stimulus signals are applied by simulator test bench which has been set up to simulate a memory access time of 50 ns between accessing two words of AAAA and F0AF. The AAAA input results in a 0101 0101 0101 0101 serial word at the MUX

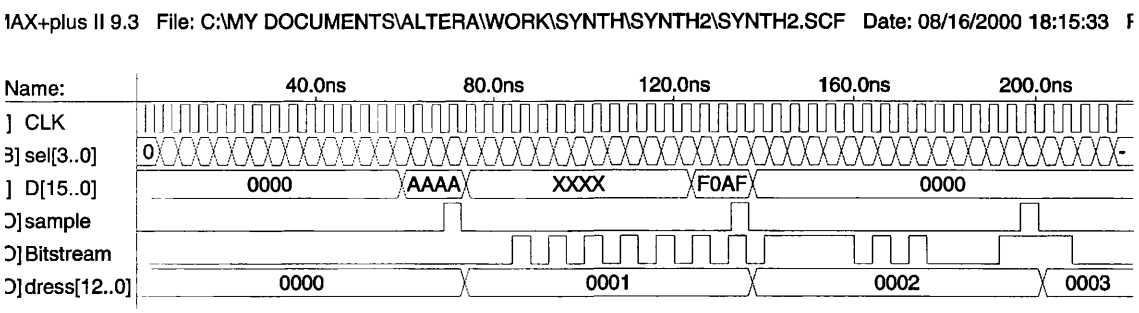


Figure 4.15: Back-annotated simulation of logic implemented in CPLD

output and the F0AF input results in the sequence 1111 0101 0000 1111. One sequence follows the other immediately at a bit rate of the reference frequency. In order to allow the maximum possible memory access time the memory address bus is changed on the same clock edge which latches the input word.

4.6 System Level Simulation

The purpose of the simulations was to predict the $\Sigma\Delta$ noise shaping referred to the output of the loop and provide a model with which to characterise the constructed hardware. An envelope simulation of the loop output spectrum was performed to allow an assessment of the impact of the noise shaper on both the RMS phase noise and spurious performance.

4.6.1 Overall Simulation Methodology

The simulation can be naturally divided into two sections, that of the modulator and that of the loop. The modulator simulation is a transient simulation which is truncated to the required sequence length followed by an FFT into the frequency domain. A different transient simulation is required for each channel with a modulator DC input corresponding to the required fractional division ratio.

The modulator sequence is converted to represent the instantaneous frequency steps introduced into the loop by the dual modulus divider switching. These frequency steps are then integrated to convert them to equivalent phase steps to apply at the phase detector node of the PLL. The rest of the simulation is performed in the frequency domain.

The spectrum is obtained from a direct FFT on the sequence of phase steps with no form

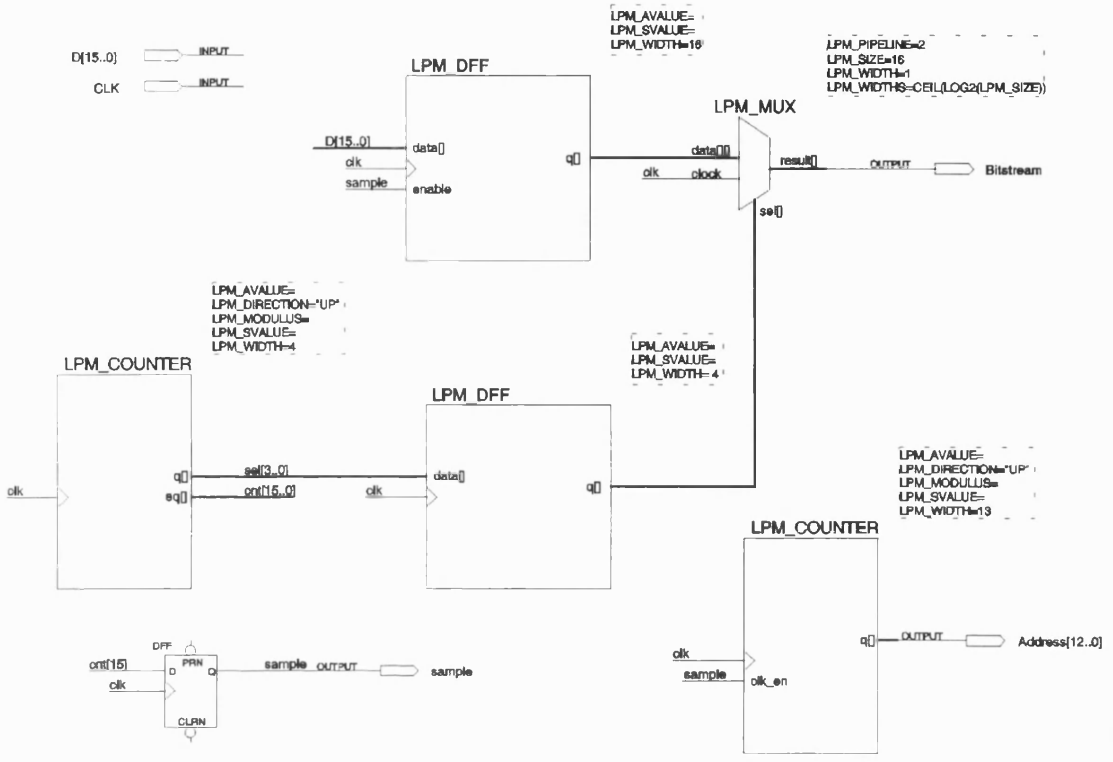


Figure 4.16: Synthesiser Logic mapped into to CPLD device.

of windowing, fully taking into account the effect of spectral leakage due to the sequence. truncation. Applying the closed loop response to the spectrum of phase steps allows comparison of the effect of different loop filtering schemes with the simulated modulator sequence. The system level simulation is performed with the matlab file system.m in appendix section D.8.

n	1	2	3	4	5	6	7
$V_{mod}[n]$	0	1	0	0	1	1	0
$\sum(V_{mod}[n])$	0	1	1	1	2	3	3
$\overline{V_{mod}[n]}$	0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{3}{6}$	$\frac{3}{7}$
$\Delta V_{eff}[n] = V_{mod}[n] - V_{av}$	$-\frac{3}{7}$	$\frac{4}{7}$	$-\frac{3}{7}$	$-\frac{3}{7}$	$\frac{4}{7}$	$\frac{4}{7}$	$-\frac{3}{7}$
$\Delta\phi[n] = 2\pi\sum_1^n(\Delta V_{eff}[n])$	$2\pi\frac{-3}{7}$	$2\pi\frac{1}{7}$	$2\pi\frac{-2}{7}$	$2\pi\frac{-5}{7}$	$2\pi\frac{-1}{7}$	$2\pi\frac{3}{7}$	$2\pi\frac{0}{7}$

Table 4.2: Relation of Instantaneous sequence value to instantaneous phase steps.

4.6.2 Modelling Divider Switching

The output sequence from the modulator is a stream of bits which must be converted into a signal representative of the effect of the divider switching. Each time the divider value changes a step change in frequency is injected into the loop at the phase detector input node. This change may be expressed equivalently as either frequency or phase, the frequency being the time derivative of the phase.

Due to the flywheel action of the loop a steady state value of phase will be achieved at the phase detector input somewhere between 0 and 2π, dependent on the duty cycle of fractional sequence. Therefore each further step change in phase is superimposed on the average phase value, which can be found from the cumulative sum of all previous phase steps. The modulator output sequence from file D.8 takes the values of either 1 or 0 as a 0.5 threshold is chosen for the quantiser. This merely has the effect of shifting the input and output range upwards by 0.5. Identical results are obtained with a 0 threshold quantiser and similarly shifted input.

Figure 4.18 shows a simulation of a fractional N synthesiser driven by a sequence from the third order sigma delta modulator. The start time of the simulation corresponds to switching to a new sequence of average value 0.3270, from an average value of 0. The top trace is a direct plot of V_{mod}, the sequence driving the modulus control of the divider. The second trace is a cumulative average of the modulator sequence over time, or the instantaneous averaged divide value:

$$\overline{V_{mod}}[n] = \frac{\sum_1^n V_{mod}[n]}{n} \tag{4.11}$$

This provides an illustration of the behaviour upon channel change. After some time period the averaged division value of the loop will settle to a fixed value corresponding to the required division value:

$$V_{av} = \lim_{n \to \text{inf}} \overline{V_{mod}}[n] \tag{4.12}$$

Once this steady state condition is reached the important information regarding the phase disturbances to the loop is the change in net change in the instantaneous phase in each reference cycle, which can be found from integrating the effective voltage step due to the difference between the instantaneous $V_{mod}[n]$ and the average V_{mod}, or V_{av}. The effective

voltage step is given by:

$$\Delta V_{eff}[n] = V_{mod}[n] - V_{av} \tag{4.13}$$

and the instantaneous phase step at the loop input is given by:

$$\Delta\phi[n] = 2\pi \int_1^n (V_{mod}[n] - V_{av}) = 2\pi \sum_1^n (\Delta V_{eff}[n]) \tag{4.14}$$

Therefore taking the cumulative sum of the effective instantaneous voltage step and multiplying by 2π results in the phase steps at the loop input. In this case voltage is used to refer to the range from 0 and 1 due to the time averaging of the divider modulus value. The third and fourth traces of figure 4.18 show the effective voltage step $\Delta V_{eff}[n]$ and the corresponding instantaneous phase step $\Delta\phi[n]$. Table 4.6.2 summarises the operations for a seven bit sequence. As a final step in order to represent the case of a settled loop it is necessary to remove the DC component, as this will only be present when the loop is in the process of re-tuning.

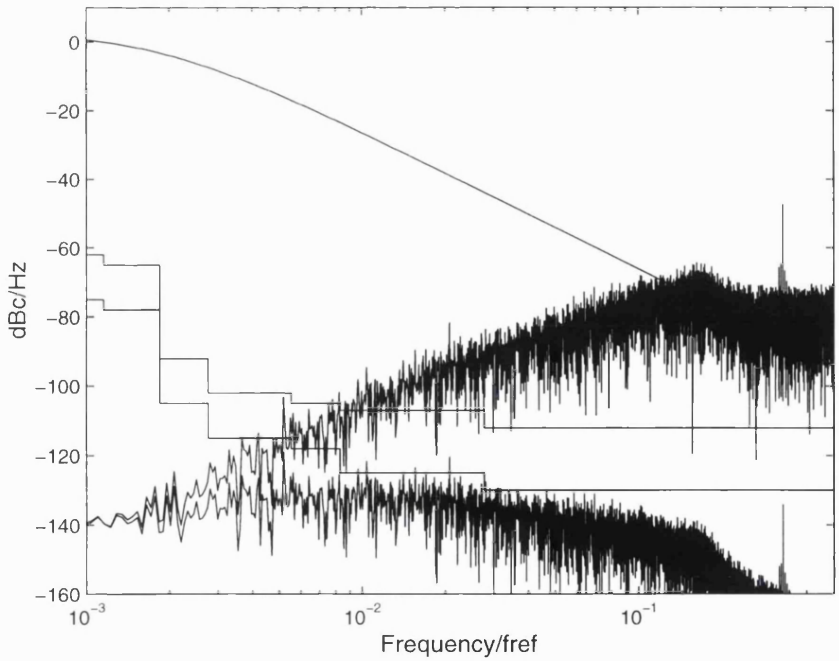


Figure 4.17: Simulation of Synthesiser Output Spectrum

4.6.3 Referring spectrum of phase sequence to the loop output

Once the sequence of phase steps at the PLL input is obtained, a discrete Fourier transform of the zero-meaned phase sequence is performed:

$$\delta\phi_{zm}[n] = \delta\phi[n] - \overline{\delta\phi[n]} \tag{4.15}$$

$$\Delta\Phi(k) = \sum_{0}^{N-1} \delta\phi_{zm}[n] exp\left(\frac{-jnk2\pi}{N}\right) \tag{4.16}$$

This DFT takes N consecutive samples of the sequence and calculates N equally spaced values of $\Delta\Phi$, the spectrum of the phase steps in the frequency domain, from 0 to the sampling rate ω_s. The integer k gives the DFT value $\Delta\Phi(k)$ corresponding to output frequency component $\omega(k)$ and is often referred to as the bin number. In practice a Fast Fourier Transform algorithm is employed in computer simulation due to the speed advantage in calculation. This has the practical effect of requiring a sequence length of a power of two. The nearest available power of two exceeding the required sequence length is used.

In the program of section D.8, a normalisation of the output of the FFT is performed by multiplying by a factor of 2/N. This normalises a peak amplitude of 1 on the time domain side of the FFT to an amplitude of 1 on the frequency domain side. The factor of 2 is to account for a single sided analytic representation of the bandpass signal. The analytic signal is a complex valued signal with a one-sided spectral density whose real part is the same as the original real-valued signal. By use of the Hilbert Transform [19] a real valued double sided signal can be represented in terms of a complex signal with a one sided spectral density.

Considering the time domain transform

$$f_a(t) = f(t) + jH(f(t)), \tag{4.17}$$

where $H(f(t))$ is the Hilbert transform of the real valued signal and $f_a(t)$ is the single sided complex analytic representation. The Fourier transform of this relation is given by

$$F_a(\omega) = F(\omega) + jH(F(\omega)). \tag{4.18}$$

In order for the analytic representation $F_a(t)$ to be single sided it is necessary for the complex part of the right hand side of this equation to cancel the negative frequency part of $F(\omega)$, therefore the following must be true for negative frequencies

$$jH(F(\omega)) = -F(\omega), \qquad \omega < 0, \tag{4.19}$$

and re-writing this relation obtains

$$H(F(\omega)) = jF(\omega), \qquad \omega < 0. \tag{4.20}$$

And as the complex signal $jH(F(\omega))$ must have odd symmetry then the positive frequency relation can be written

$$H(F(\omega)) = -jF(\omega), \qquad \omega > 0. \tag{4.21}$$

This results in the following expression for the Hilbert Transform in the frequency domain

$$H(F(\omega)) = -jF(\omega)sgn(\omega). \tag{4.22}$$

Allowing the spectral density to be written for the analytic representation

$$F_a(\omega) = \begin{cases} 2F(\omega) & : \quad \omega > 0 \\ 0 & : \quad \omega < 0 \end{cases} \tag{4.23}$$

4.6.4 Power Spectral Density and NBPM approximation

Referring to the simulation file D.8, once the normalised single sided spectrum of the phase noise voltage at the loop input is obtained, this is divided by two to account for a narrow band phase modulation of the carrier. The magnitude of this complex signal is taken and then squared to give a normalised phase noise power referred to the loop input. Following this the signal is converted to log form and the scaling is changed to dBc/Hz by division by the FFT resolution. The resolution of the FFT is dependent on the number of bins and the reference frequency chosen for the simulated synthesiser.

The spectrum at the loop output is finally obtained by application of the closed loop phase modulation characteristic 2.68. Figure 4.17 shows the output of the simulation. The top trace is a plot of the closed loop phase modulation characteristic, below this are plotted the spectrum of phase noise due to the divider switching and the same spectrum under the

influence of the top trace response.

Appendix section A gives a detailed coverage of phase modulation effects. The phase modulation index will be of a very small value close to the carrier due to the noise shaping of the modulator. Further out from the carrier the phase noise injected by the modulator switching is much larger, but due to the suppression of the closed loop response the resulting modulation index at these frequencies is also very small, allowing an accurate approximation.

At this stage the reference frequency or loop filter frequency of the simulation can be adjusted in order to satisfy specification masks for SFDR and phase noise. Figure 4.17 shows representative masks from the ETSI specifications for the DCS1800 mobile phone system standard. By optimising the modulator it was found possible to satisfy these specifications across the band using a 216 MHz reference frequency and a 250kHz natural frequency loop filter. Although the prototype reference frequency was constrained to 156MHz by the divider, to compare results the simulation reference frequency was later re-scaled to the same figure to provide a valid model for characterisation of the hardware.

4.7 Summary

This section has discussed the design, simulation and analytical techniques which were used in the design of a prototype Fractional-N frequency synthesiser using stored sigma delta sequences. The next section will present measured results from the fabricated synthesiser and compare them to the results predicted by the analysis. In this section the issue of sigma delta stability has not been fully addressed and empirical simulation has been used in order to optimise the spectra for a number of frequencies in the synthesiser range. The complex issue of modulator optimisation is addressed in depth in the optimisation chapter, which gives specific attention to the difficulty of generating sequences with effective noise shaping with the large signal modulator inputs necessary in Fractional-N applications. The use of empirically generated sequences is valid in the verification of the hardware platform as it allows the predicted and measured spectra to be compared for a particular shaping sequence, allowing an independent investigation of the performance of the Fractional-N loop itself.

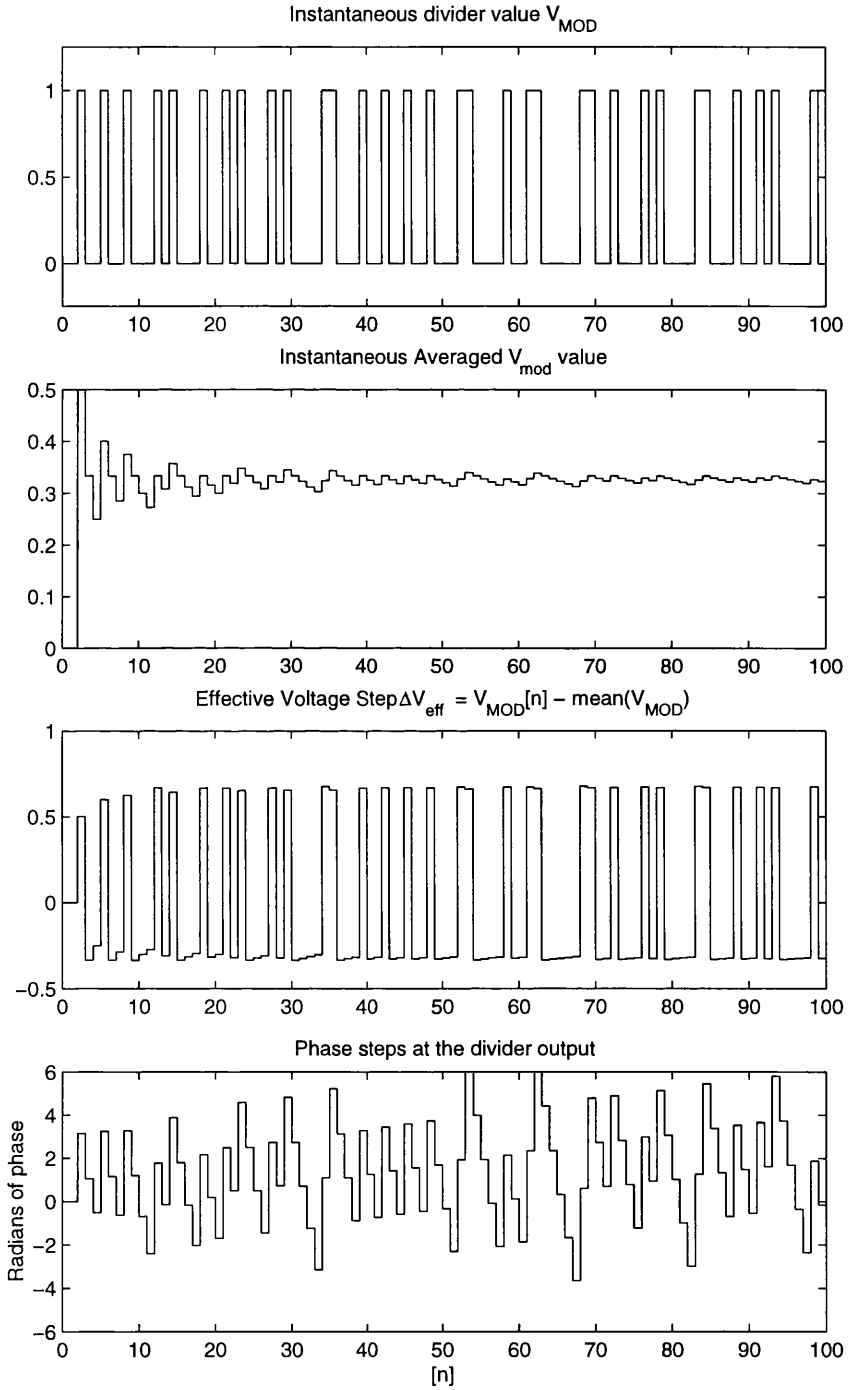


Figure 4.18: Instantaneous Phase at Phase detector Due to Divider Switching

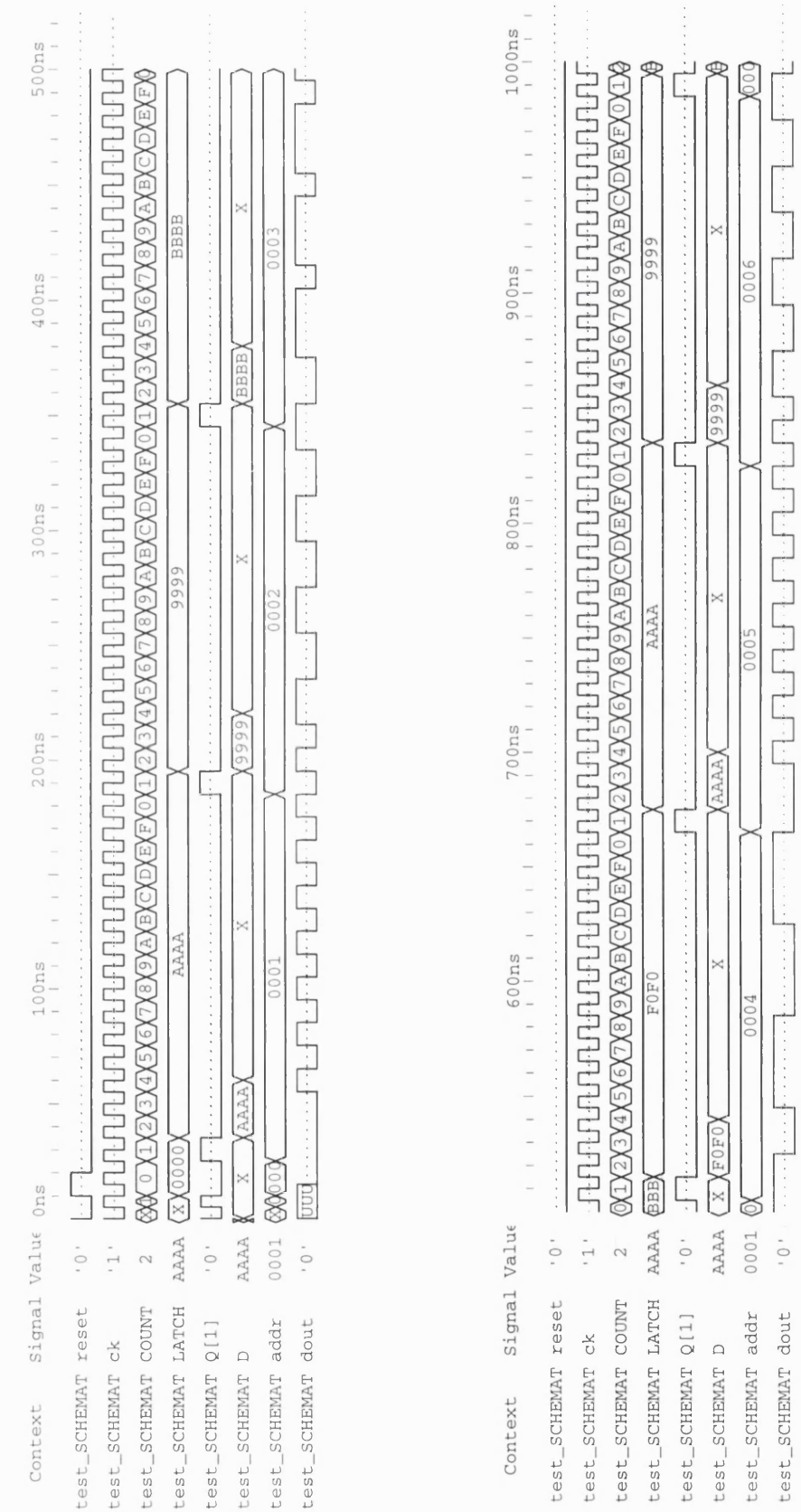


Figure 4.19: Example Timing Diagram for PLD memory access, 100 MHz clock.

Chapter 5

Prototype Synthesiser Results

This section presents measured against simulated results for the constructed hardware prototype. A hardware prototype was deemed valuable in providing proof of concept whilst revealing any potentially unforseen issues in the approach. In this chapter, results are presented for acquisition measurements, phase noise measurements and various plots of power spectra both at the input to the dual modulus divider and the loop output.

For the spectral plots the same eight frequencies are tested across the band and presented at the two locations in the circuit. This was to provide insight both into the performance variation across the band and also the nature of additional nonlinear effects contributed to the output spectrum by the non-ideal components in the synthesis loop. The intention of this was to provide a clear picture of which effects were attributable to the shaping sequence.

As the maximum clocking frequency of the 8/9 prescaler was 1.4 GHz, this restricted the loop reference frequency to 156 MHz even though the digital section of the circuit would operate at 250 MHz. This resulted in the output frequencies of table 5 being documented during the tests. Due to the divider clock frequency restriction it was also later found that the phase margin of the loop was lower than designed due to VCO gain variations. The phase margin figures in table 5 were obtained from a later Spice simulation which included the extracted VCO tuning characteristic. The extracted results were used to re-simulate the results of the actual hardware to provide an improved match.

The test frequencies resulted from the original choice of reference frequency of 216.7647 MHz, which corresponds to the centre of the DCS1800 band (1805-1880 MHz) divided by the central fractional division value of 8.5. The original impetus behind this approach

Fractional Division Ratio	Output Frequency (GHz)	Predicted Phase Margin
8.672999872	1.35298798	40^o
8.649876603	1.34938075	39^o
8.626817756	1.34578357	37^o
8.603758782	1.34218637	36^o
8.580699872	1.33858918	35^o
8.557640923	1.33499198	34^o
8.534646218	1.33140481	33^o
8.511523045	1.32779759	31^o

Table 5.1: Test Frequencies for Prototype Synthesiser

was to allow a halving of the memory requirements by inverting a sequence on one side of N+1/2 to cover the other side of the band in a symmetrical manner.

Signal integrity of the digital logic was verified by triggering a storage scope with one of the memory address pins and comparing the on-screen sequence at the divider control input with the corresponding location in the sequence that had been downloaded to memory. A sequence section up to 32 bits was compared and verified in this way.

5.1 Modulator Spectra

The spectrum of the shaping sequence was measured directly at the output pin of the programmable logic device driving the prescaler divide ratio input. The purpose of this was to provide an insight into the source of spurious components at the synthesiser output. Components present at the loop output which could not be explained from the spectrum of the baseband noise shaping sequence were then able to be attributed to other nonlinear effects in the loop rather than to the noise sequence. Figure 5.1 shows a spectrum analyser measurement from DC up to 200 MHz, evidencing the noise shaping of the sequence at the divider modulus control input.

5.1.1 Test Setup

Appendix D.9 shows the Matlab simulation file used to simulate the spectrum at the programmable logic device output from the pre-generated shaping sequence. The same se-

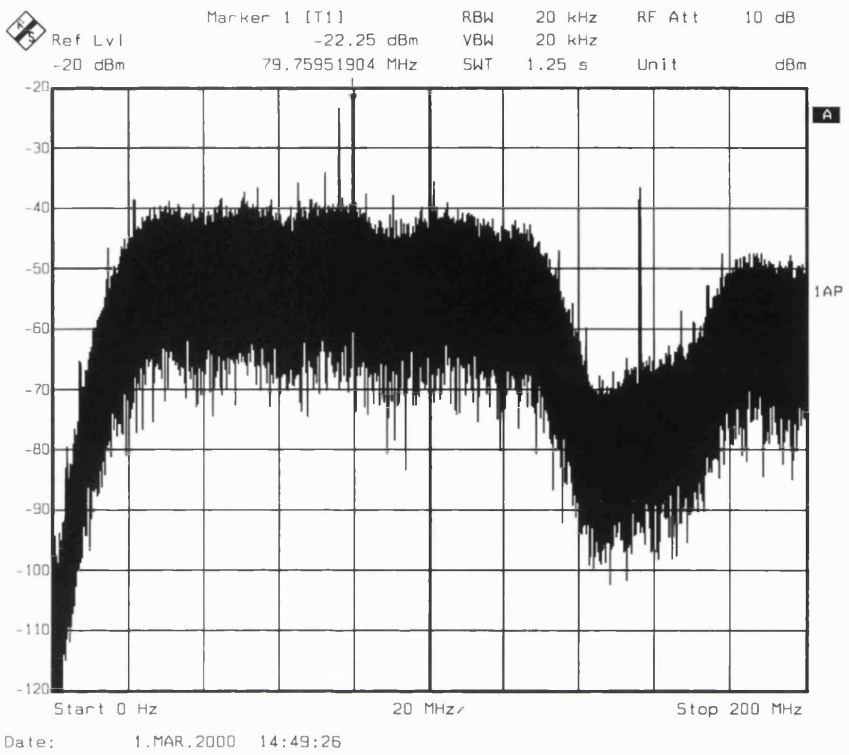


Figure 5.1: Baseband Spectrum at dual modulus divider input

quences used for the simulation were used in the memory of the hardware.

Referring to the text of appendix D.9, the program loads the sequence from disk and scales this to match the measured CMOS voltage level. Following this an FFT is taken and normalised to represent an analytic single sided spectrum of the signal. After squaring to give power in dBW, an additional 30dB is subtracted to give the power spectrum in dBm to match the analyser scale.

The spectra from the hardware were measured directly at the dual modulus divider control input with a synthesiser reference frequency of 156MHz. A Rohde and Schwarz FSEB spectrum analyser was used to measure the signal from the 5V CMOS level output of the programmable logic device using a Textronics P6139A 1MΩ probe.

5.1.2 Comparison of Results

Figures 5.2 through to 5.9 show the simulated results on the left and the corresponding analyser result on the right. The most significant immediate difference is the flattening of the noise shaping at the lower frequencies of the measured results and then a gradual rise

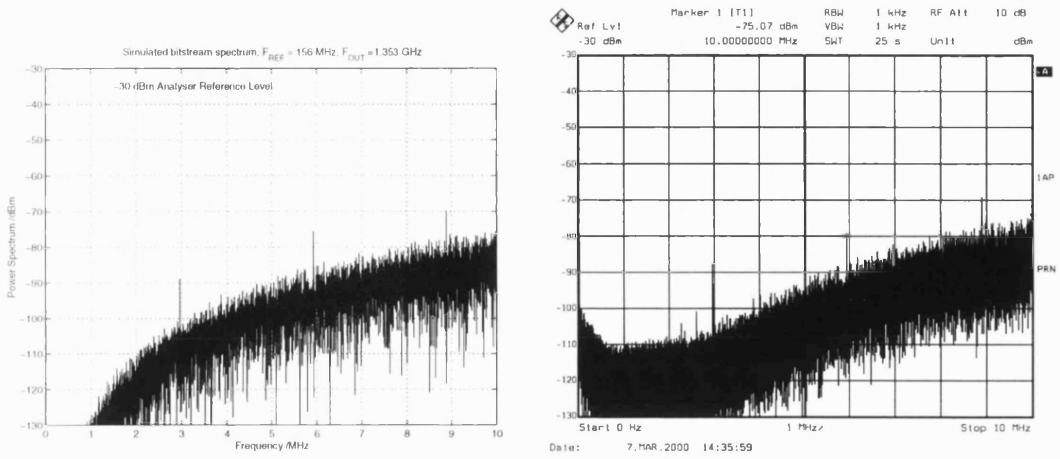


Figure 5.2: Sequence results for 1.3530 GHz

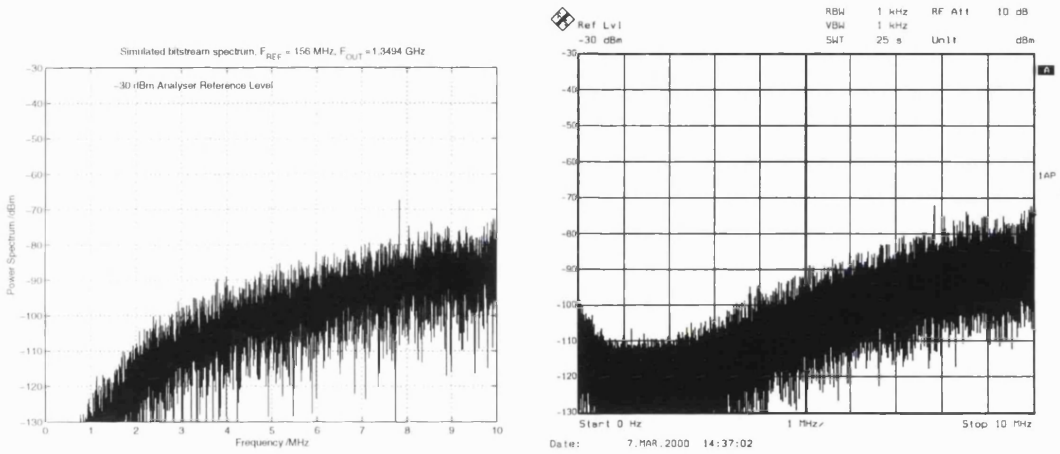


Figure 5.3: Sequence results for 1.3494 GHz

of the response very close to DC. This was due to the noise floor of the analyser itself and flicker noise respectively.

The results for the simulations compare very closely with the results measured from the hardware from the modulus control signal. The general level of components in the two sets of spectra is very close down to a frequency of around 3 MHz where the analyser noise becomes a restriction.

All eight of the comparisons display evidence of major spurs which are within 4 dB comparing the measured and simulated results, a particularly good example of this being figures 5.2 and 5.8 and 5.9. The extra component appearing in the measured result of figure 5.5 at a frequency near 3 MHz must derive from a source other than the intrinsic bitstream sequence.

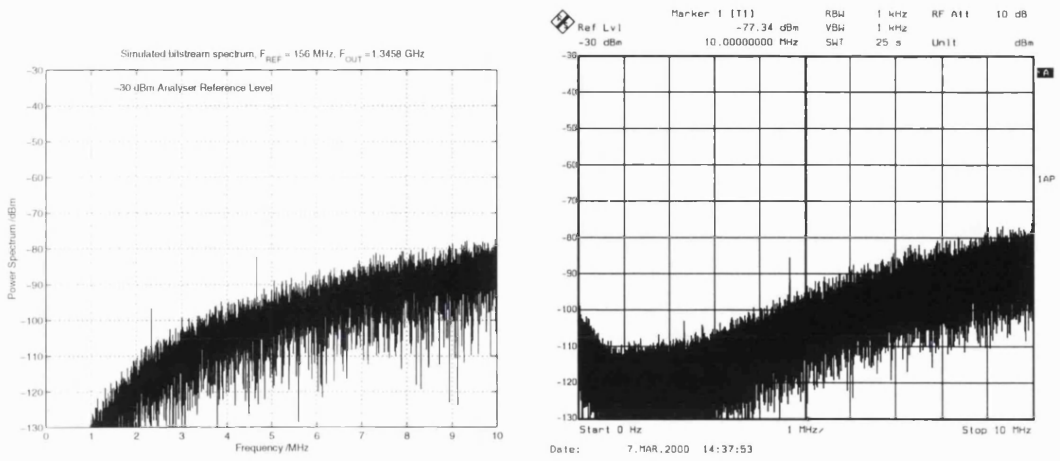


Figure 5.4: Sequence results for 1.3458 GHz

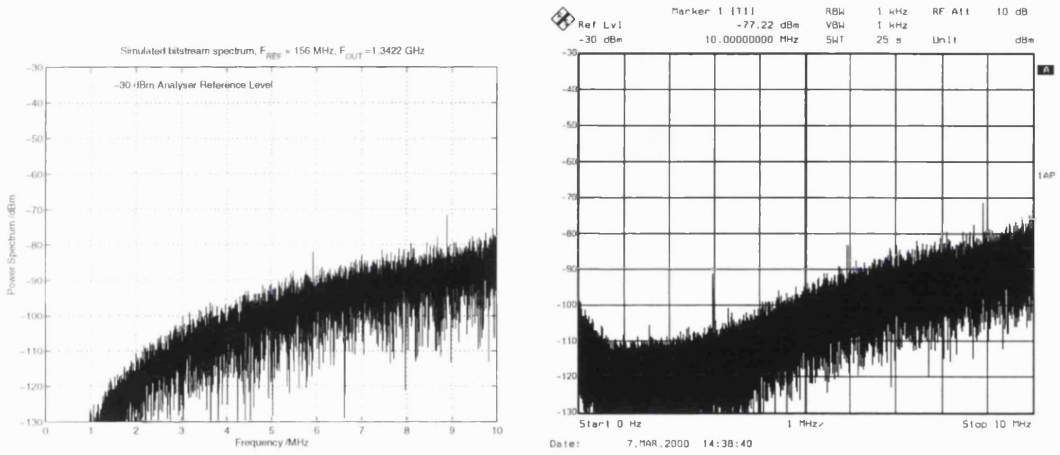


Figure 5.5: Sequence results for 1.3422 GHz

5.2 Phase Noise

Figure 5.10 shows a measurement of phase noise spectral density captured from the phase noise measurement feature of a Hewlett Packard 8560E spectrum analyser. Original simulations showed that the phase noise plateau of the synthesiser should be in the region of -110 dBc/Hz but this is not entirely evident on the measured result for a number of reasons.

One reason for the difference in measured result is the contribution of VCO noise to the noise at the loop output. From figure 5.10 this appears to be dominant out to 10 kHz from the carrier, as the oscillator noise is shown by the linear sloping part of the noise measurement close to the carrier.

The departure from the initially predicted results is the lump in the phase noise at an

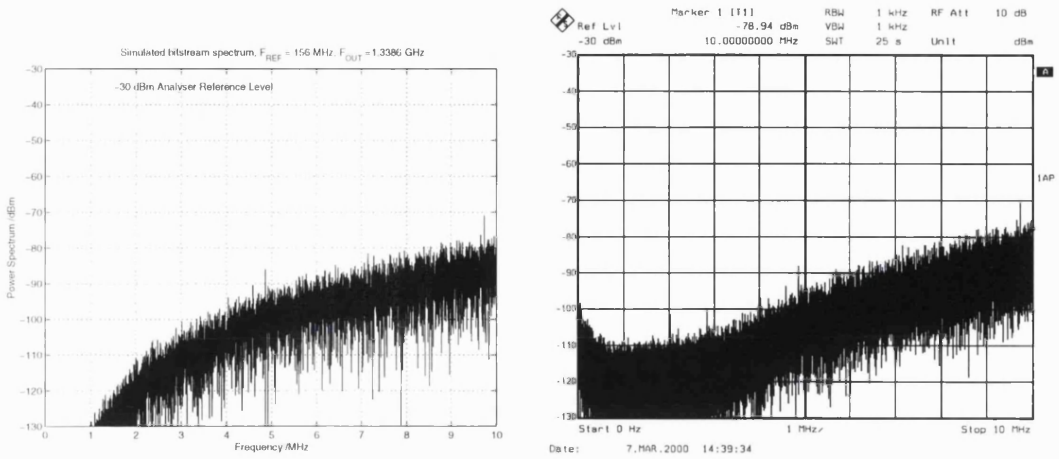


Figure 5.6: Sequence results for 1.3385 GHz

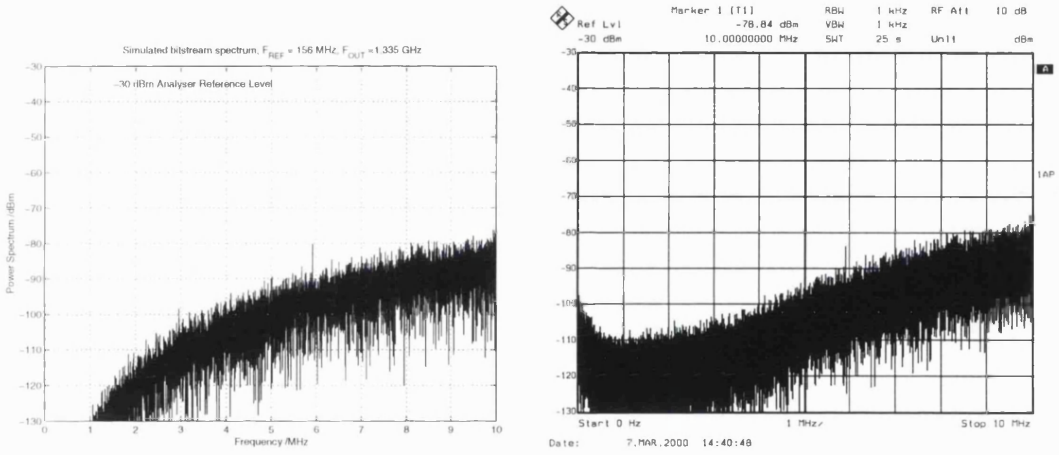


Figure 5.7: Sequence results for 1.3350 GHz

offset of around 250-500 kHz offset from the carrier. It is thought that this is due to a combination of non ideal effects in the loop filter and the result of a reduced phase margin due to the VCO being used away from its original intended frequency, which resulted in a reduction in the phase margin of the loop and consequent peaking of the closed loop response.

Figure 5.11 shows two simulations of the output spectrum of the synthesiser produced from the simulation file in appendix D.10. The left hand plot shows a result taking a phase margin of 5^o, whilst the right hand result shows a result with the original design phase margin of 50^o.

Clearly at least some of the peaking in the noise response is due to the reduction in phase margin caused by non-ideal loop filter design. Other effects which may have caused this

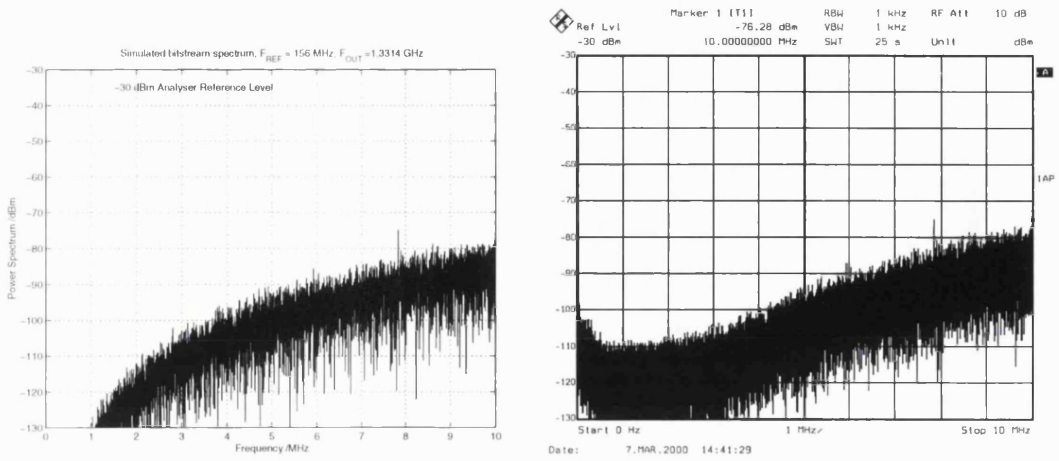


Figure 5.8: Sequence results for 1.3314 GHz

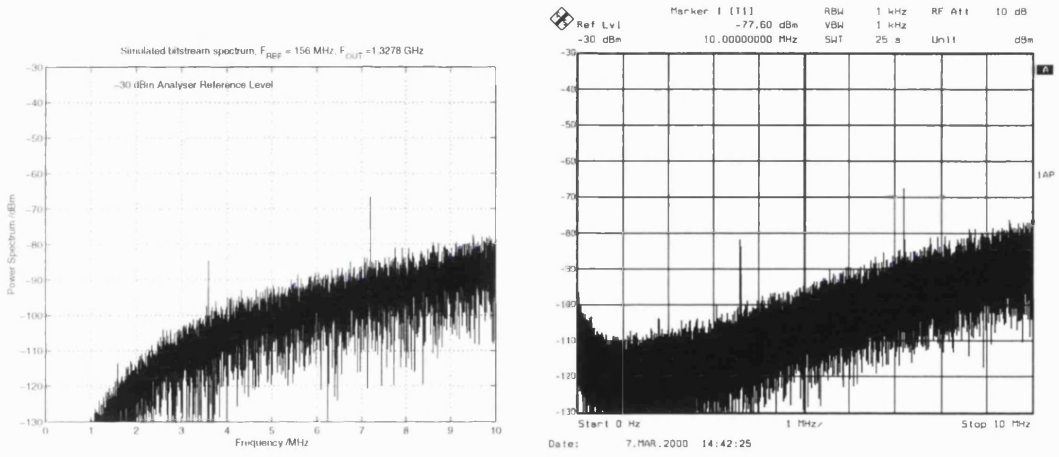


Figure 5.9: Sequence results for 1.3278 GHz

include noise contributions from the divider and further peaking of the closed loop response caused by the limited loop filter bandwidth.

Figure 5.12 shows the actual variation in phase margin of the prototype synthesiser. This data was calculated from combining a Spice simulation of the loop filter components with the variable VCO tuning gain derived from later measurements. The large variation in phase margin is due to the narrow operating band of the VCO. Appendix D.6 shows the code used for combining and plotting the data. Due to a restriction on the maximum frequency of the prescaler component it was not possible to operate the synthesiser at the optimal point for the loop filter design at the time of the measurements. In hindsight this could have been avoided by re-measurement of the VCO tuning gain over a much wider range at the time to reveal the large gain deviation outside the specified range.

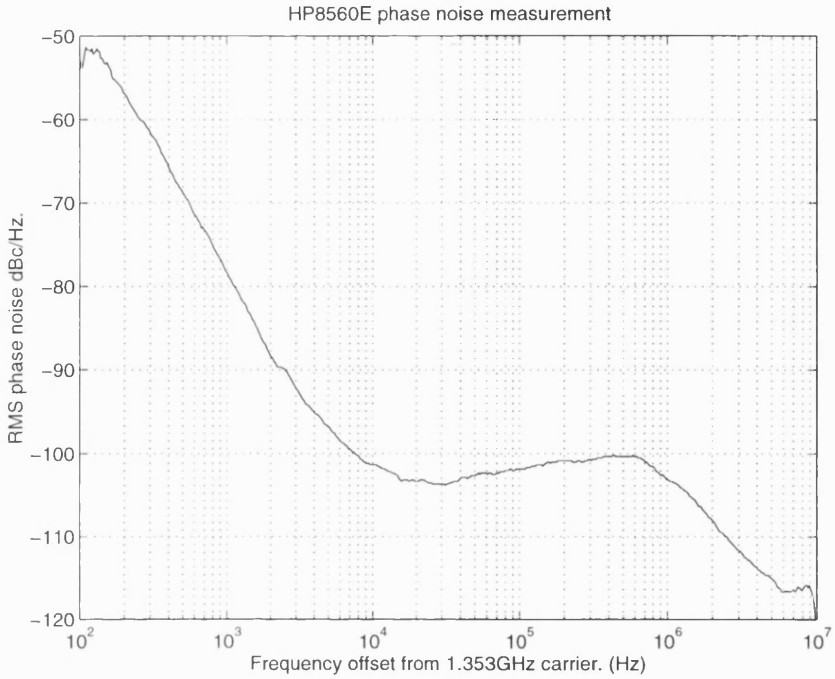


Figure 5.10: Measured phase noise at 1.353 GHz

5.3 Frequency Hopping

Prediction of the transient performance of the loop was performed from a transient simulation using the Laplace transform of the closed loop response. These were derived in chapter 2 in section 2.4.2. Equation 5.1 gives the frequency error transient at the phase detector, whilst equation 5.2 gives the phase error transient, also referred to the phase detector.

$$\omega_e(t) = \Delta\omega \left[1 + \omega_n t - (\omega_n t)^2\right] e^{-\omega_n t} \tag{5.1}$$

$$\phi_e(t) = \frac{\Delta\omega}{\omega_n} \left[\omega_n t + (\omega_n t)^2\right] e^{-\omega_n t} \tag{5.2}$$

Both of these equations can be modified to give the corresponding transients at the VCO output merely by multiplying by the feedback divider ratio N.

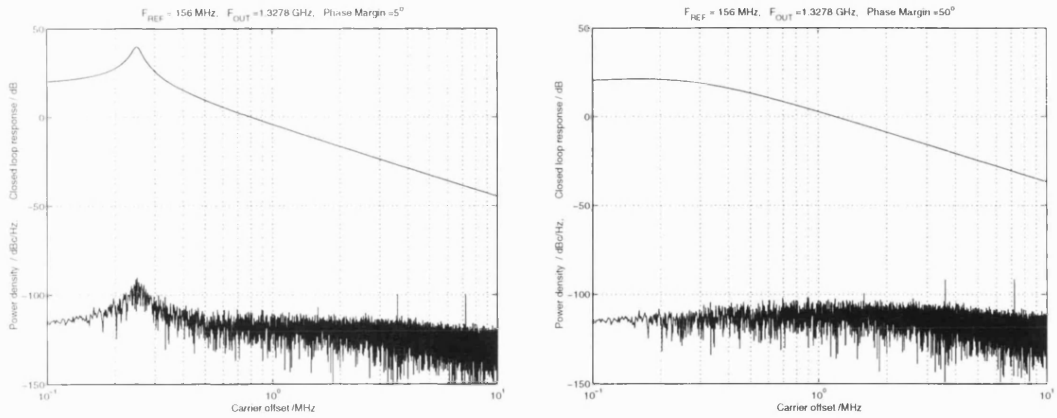


Figure 5.11: Simulated power spectrum for 5 degree and 50 degree phase margin

5.3.1 Simulation Method

Figure 5.13 shows the results of the transient simulation of a 15 MHz frequency step for a 250 kHz loop natural frequency with an 8/9 divider. The simulation was performed using the simulation file Ftrans.m in appendix D.3. A worst case result is obtained for the phase transient by taking a fixed value for N of the higher of the two divide ratios. Taking N=9 the phase error referred to the PFD after 10μs is less than 0.1 ^{o}, whilst with N=8 the 10μs phase error is 0.2 ^{o}. The top left plot of figure 5.13 shows that a 15 MHz frequency hop is the largest possible without incurring cycle slips, as the phase frequency detector will have a maximum range of $\pm 2\pi$.

5.3.2 Experimental Setup

To allow precise measurement of the frequency hopping of the prototype loop the mixer method was used. Figure 5.14 shows the experimental setup used. The output of the synthesiser is mixed with the output of a signal generator which is tuned to precisely the same frequency as one of the frequencies in the synthesiser hop to be measured.

The output of the mixer forms a beat note between the two input frequencies which is fed to the y input of the oscilloscope. The output component of interest is the f_1-f_2 product near to DC, which when f_1=f_2 will result in a flat scope trace at zero volts assuming there is no DC offset in the mixer output. To allow precise matching of the two frequencies it is necessary that both frequency generator and synthesiser have the same reference source to avoid relative drift of phase and triggering of the scope is performed with the synthesiser channel change signal. Figure 5.15 shows a 15 MHz hop being achieved in 10μs to within

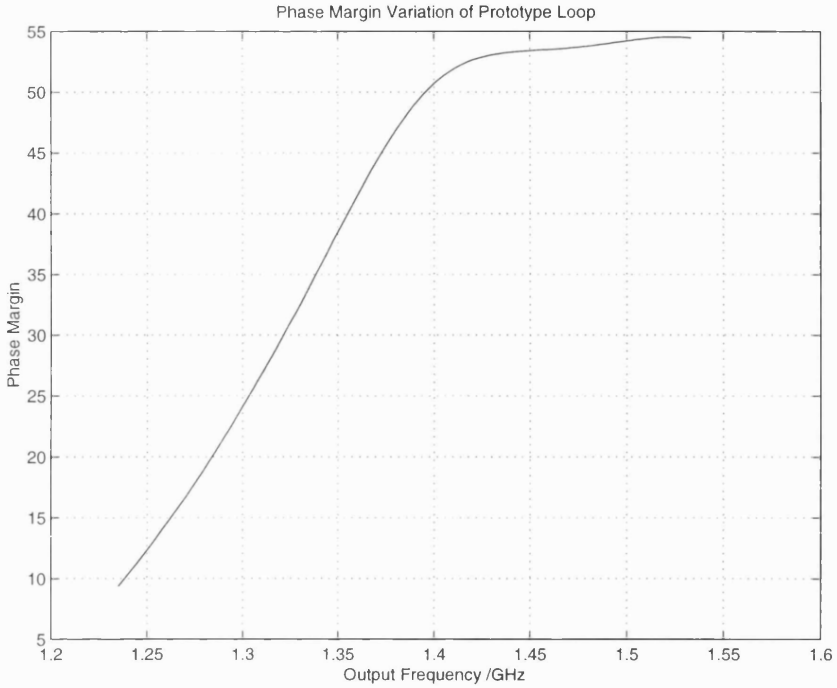


Figure 5.12: Phase Margin Variation of Prototype Loop

a few degrees, which compares favourably with the transient simulation of figure 5.13.

5.4 Output Spectra

The output spectrum of the synthesiser was measured directly at the VCO output node for the same frequencies corresponding to those of the baseband measurements. All measurements were performed on an FSEB spectrum analyser set to a 1 kHz resolution bandwidth. Simulation results are displayed on the right hand sides of figures 5.16 through to 5.23 with the corresponding measured results on the right hand side.

5.4.1 Simulation Setup

Appendix D.2 shows the Matlab simulation file used to simulate the spectrum at the loop output from the pre-generated shaping sequence. This simulation is similar to the initial system simulation of appendix D.1, with the difference that the phase margin of the loop filter was varied in accordance with extracted measurements from the prototype board.

Figure 5.12 shows the plot of the phase margin across the band for the actual hardware, taking into account Spice simulation for the operational amplifier and a curve fit to the

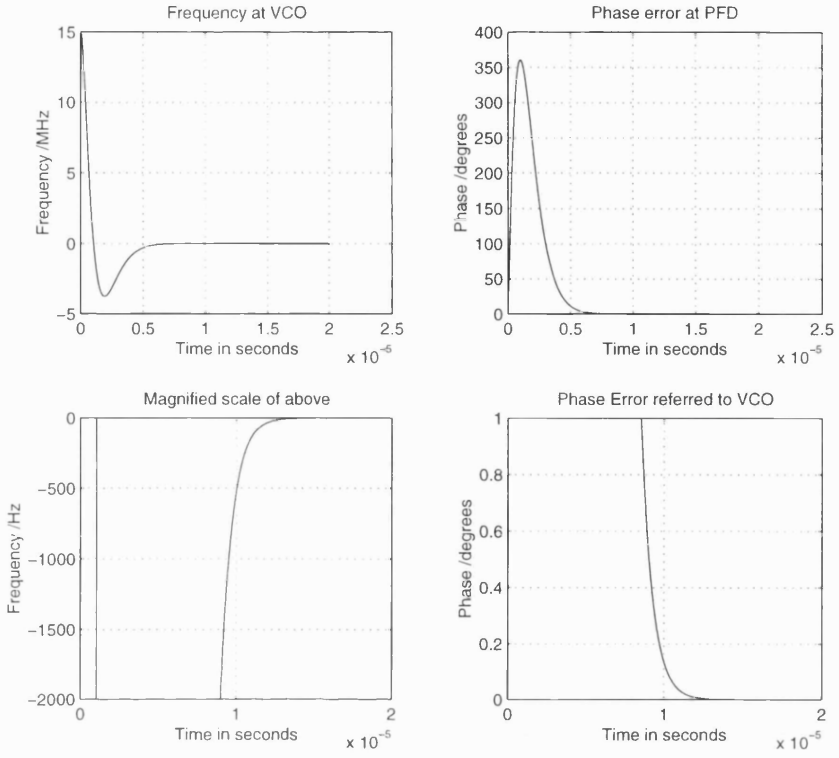


Figure 5.13: Modelling of a 15MHz frequency hop.

derived VCO tuning gain outside its intended operating range. Appendix D.5 shows the Matlab file used to perform a least means squared curve fit to the measured VCO tuning data and calculate the corresponding gain, whilst appendix D.6 shows the calculation of the resulting phase margin of the prototype loop. The phase results from this were entered into the file of appendix D.2 in order to take into account the resultant peaking of the closed loop response of the loop with the lower VCO gain.

The main steps in the simulation file are to perform an FFT of the memory sequence from the stored file on disk, then normalise and scale the FFT to provide a result in dBc. Following this the result is squared to show power and converted to dB. A -6dB scaling is used to account for the narrow band phase modulation and the closed loop response is then applied to refer the spectrum to the VCO output node. Appropriate scaling of the axes are applied to match the analyser.

5.4.2 Results Discussion

The general level of spectral components in the measured results matches closely with the simulations. Although the measured results display a higher variance the mean level at an offset of 5 MHz compares very favourably. The peaking closer to the carrier can be

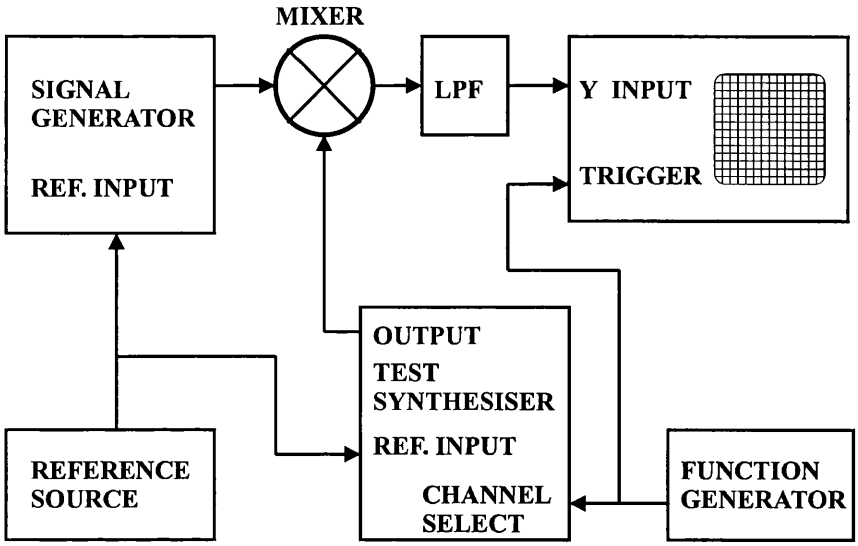


Figure 5.14: Experimental setup used for accurate lock time measurement

attributed to a combination of VCO noise and peaking of the loop filter response. Peaking of the loop filter response could be accounted for in part by the VCO gain although the physical layout of the circuitry is also capable of affecting this significantly.

The peaking evident in the spectrum becomes much more pronounced at the lower test output frequencies of the loop, which ties in with the phase margin plot of figure 5.12 and supports the notion this is one of the causes of the peaking. Another contribution to this effect is the divider gain variation which also has an effect on the closed loop performance of the loop. Both of these effects could be minimised by using a non-linear compensating amplifier in series with the loop filter.

Regarding discrete spurii, there are several instances of departure from the predictions. This reflects the numerous possible sources of non-linear effects in the analog section of the synthesiser. Plot 5.23 contains similar discrete spurii to the simulation although there is evidence of some skew symmetry of the measured result and the measured result is around 8 dB higher. This component at 3.5 MHz offset from the carrier can certainly be attributed to both measured and predicted divider sequences in the corresponding figure 5.9. The 3MHz offset component in output spectrum 5.16 also corresponds well with the divider measurement of figure 5.2.

The result of figure 5.3 also corresponds well with the shaping sequence plot of figure 5.17, as at this portion of the synthesiser tuning range the VCO tuning gain is fairly high and the peaking of the closed loop response is not too significant.

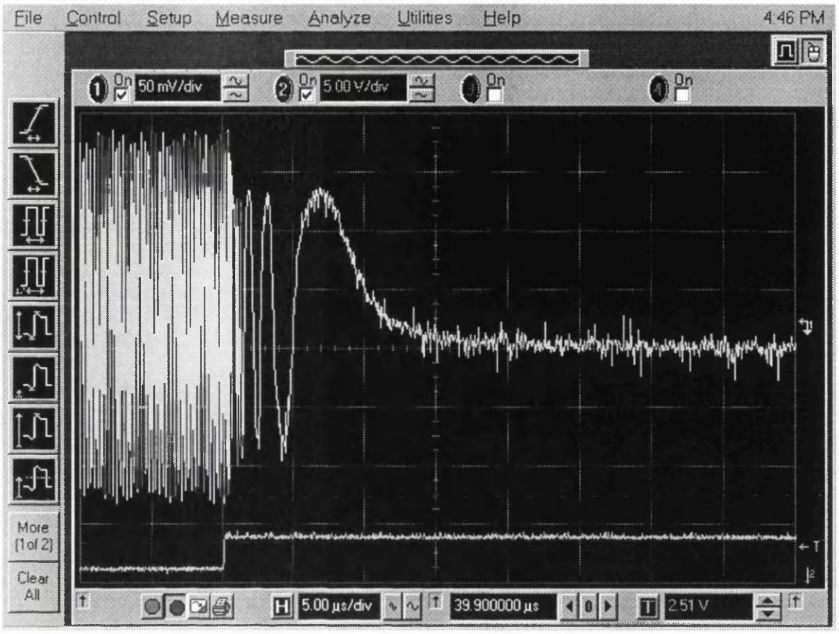


Figure 5.15: 15MHz Frequency hop in 15μs measured with the Mixer method

The result of figure 5.21 shows a number of components in the measured output which are not present on either the simulated output or the measured sequence spectra of figure 5.7. The power in these components was heavily dependent on the drive levels at both the reference frequency input and the divider input which suggests that these components are a result of intermodulation distortion rather than a limitation due to the shaping sequence. This is bourne out by the fact that most of the results show comparatively little evidence of this kind of effect.

Figure 5.24 shows a fullspan plot of the spectrum at the synthesiser output. A large amount of leakage occurs of signals related to the baseband sequence spectrum and various signals related to digital noise from the logic section. These results support the notion that a bandpass filter inserted immediately after the VCO output would aid greatly in spurious reduction. In particular this would reduce the amount of intermodulation products at the output of the buffer amplifier which follows the VCO. Another potential source of distortion is cycle skipping of the divider which was operating at the edge of its specified toggle rate.

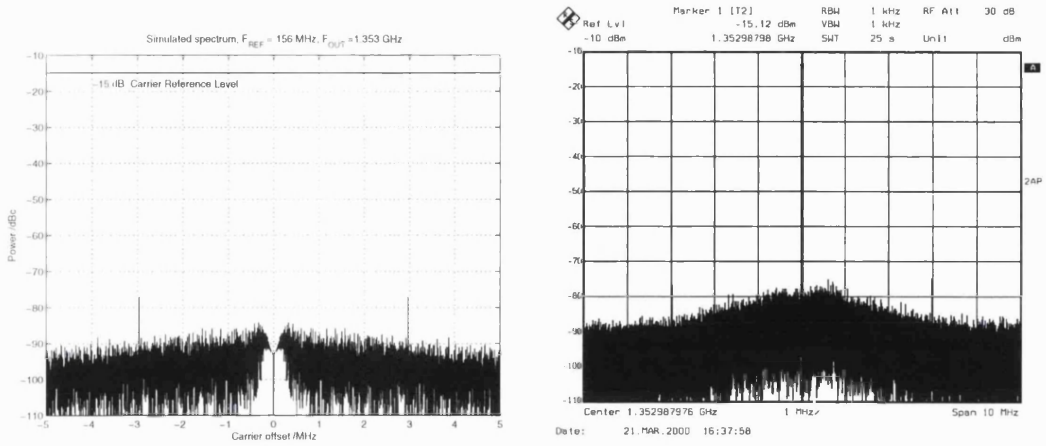


Figure 5.16: Results for 1.3530 GHz

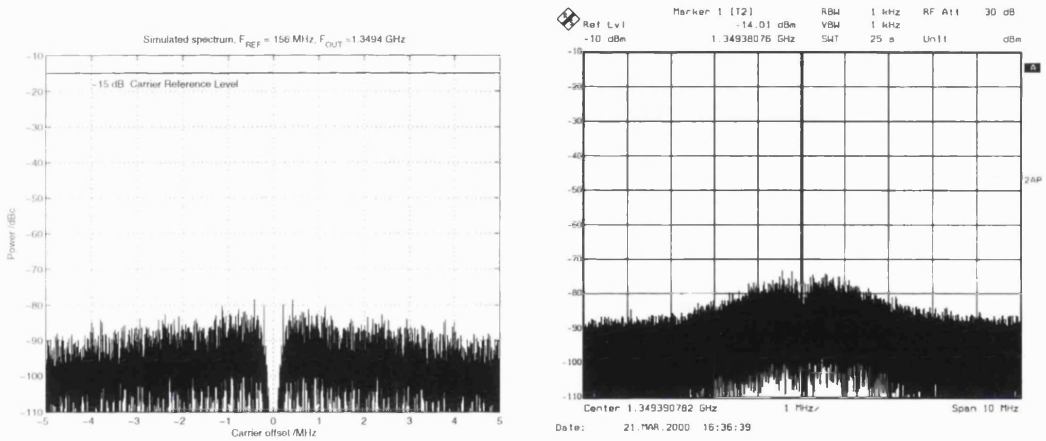


Figure 5.17: Results for 1.3494 GHz

5.5 Summary

SFDR is better than 60 dB across the band which is a very high level of spectral purity considering the extremely wide loop filter bandwidth and fast hopping speeds. Further spurious reduction could be obtained easily with a narrower loop filter for less speed intensive applications which require a high resolution synthesiser. Both spurious and noise performance compared well with predictions when imperfections in the prototype system were taken into account.

Higher than expected spur levels were attributed to non-linearity in the loop components due to their amplitude being strongly linked with that of the reference frequency and divider input drive levels. This assumption was strongly supported by the majority of the results where major discrete spurii were similar to those simulated. Therefore this situation may be considerably improved by optimisation of drive levels or additional filtering

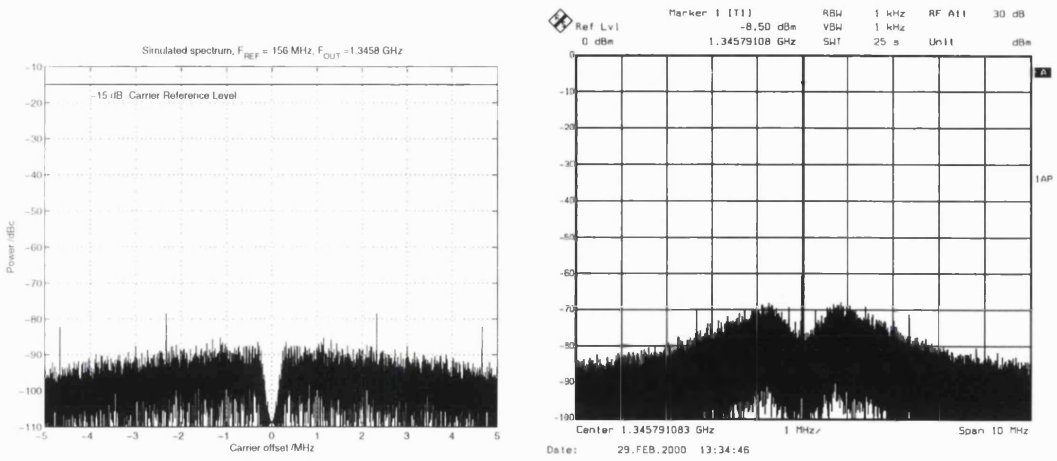


Figure 5.18: Results for 1.3458 GHz

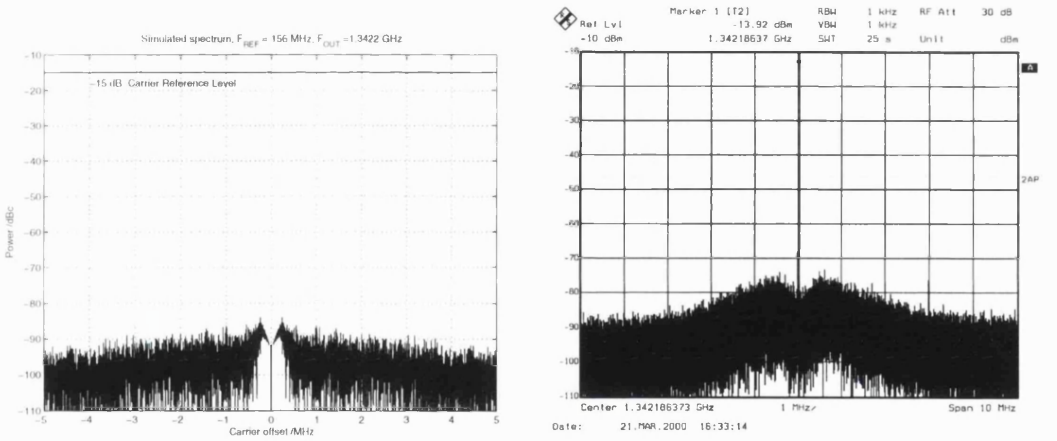


Figure 5.19: Results for 1.3422 GHz

in the loop following the VCO.

Considerably improved performance could be obtained by further optimisation of the hardware platform. Prototype performance did not appear to be limited by the shaping sequence which indicates room for a performance improvement. Example improvements would be the use of compensators to optimise loop filter response over a wider range as well as drive level optimisation with respect to spurious generation. In addition, the shaping sequence used in the prototype used only a crude optimisation procedure.

As performance conforms reasonably well with the predictions it is expected that spectral performance could be considerably improved upon by using a non-linear stability analysis to design the modulator producing the shaping sequence. Use of such techniques allows an analytical optimisation of the shaping sequence by operating the modulator closer to its

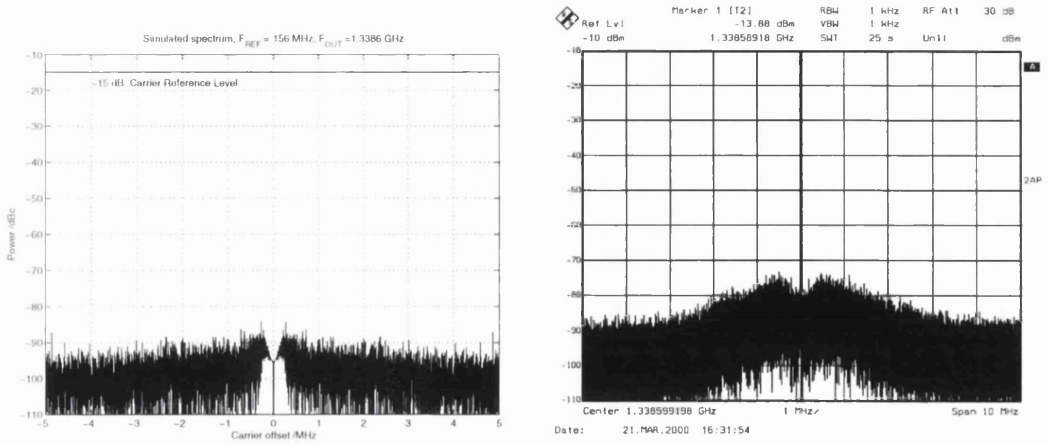


Figure 5.20: Results for 1.3385 GHz

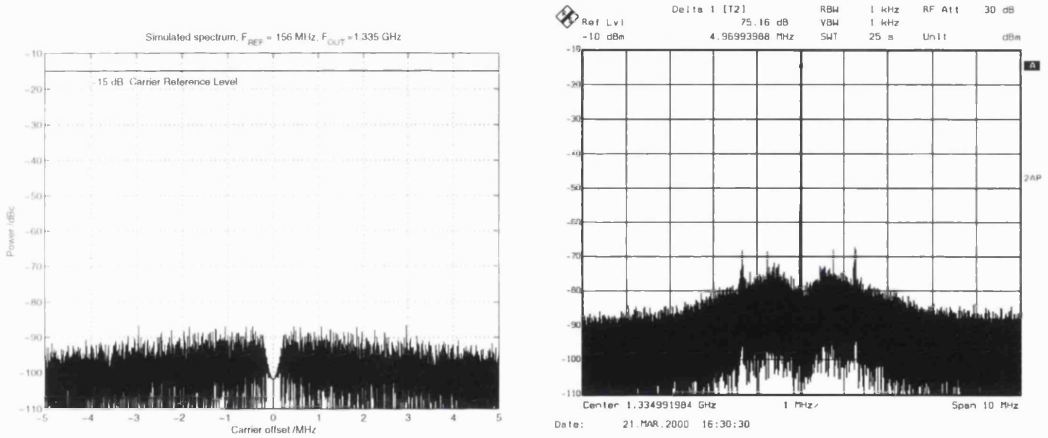


Figure 5.21: Results for 1.3350 GHz

stability margin. This would also allow the use of higher order modulators, which would further improve the spectral performance of the fractional division process.

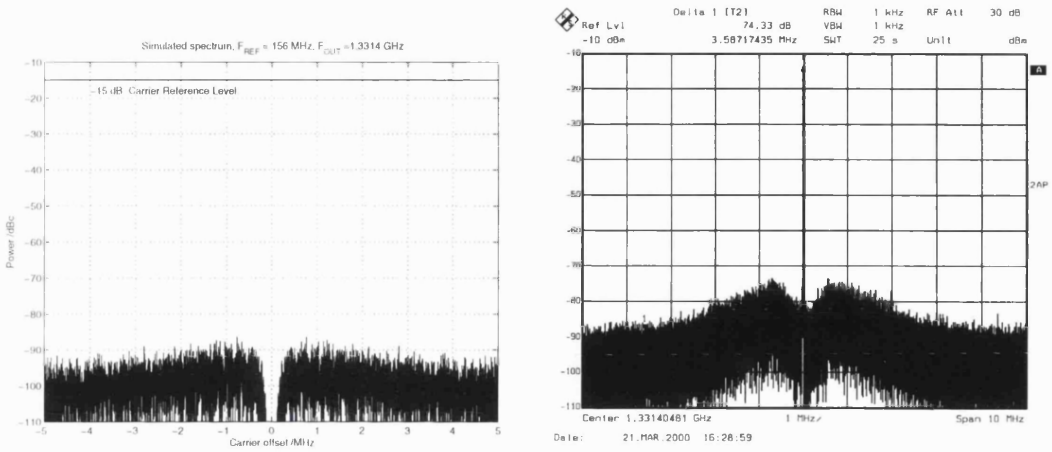


Figure 5.22: Results for 1.3314 GHz

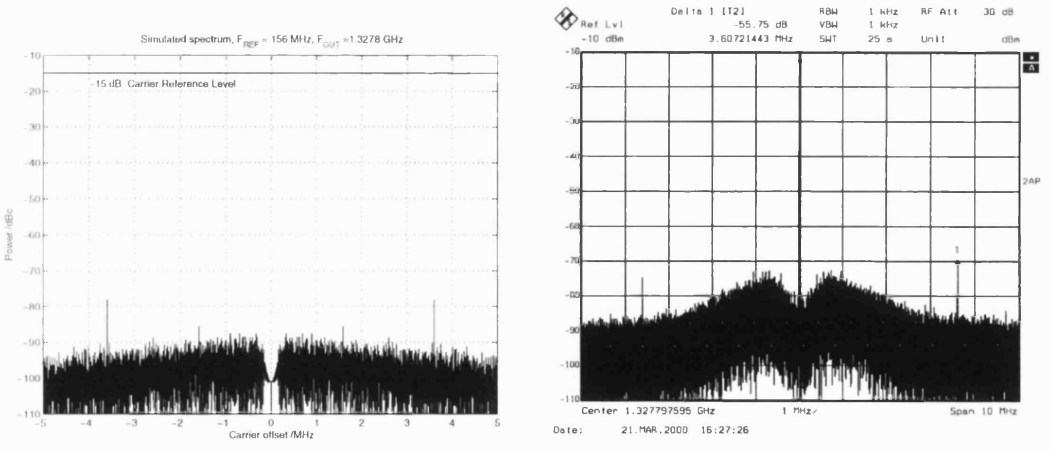


Figure 5.23: Results for 1.3278 GHz

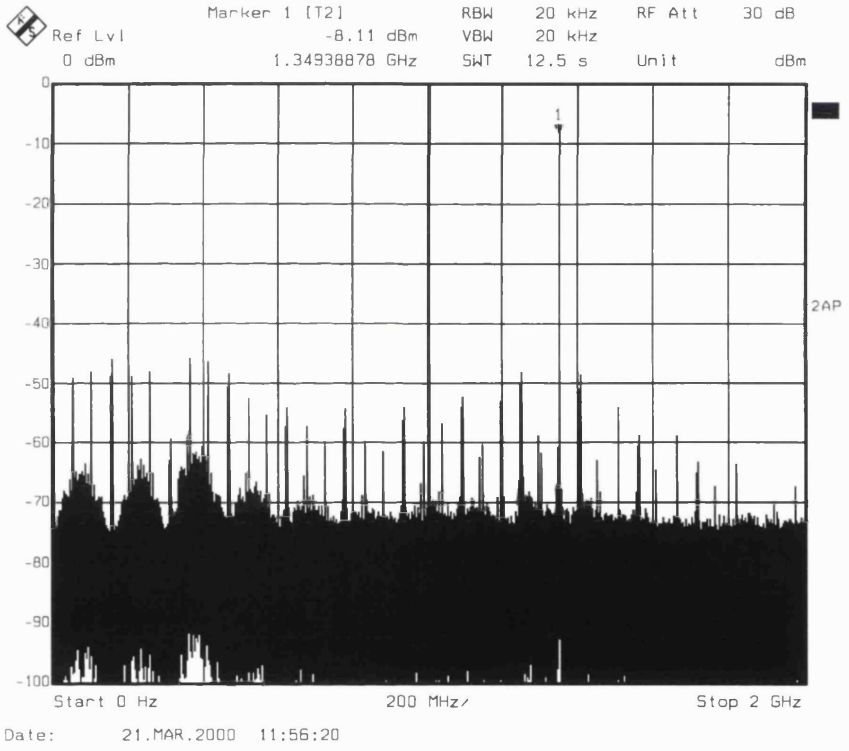


Figure 5.24: Fullspan Spectrum of Synthesiser output

Chapter 6

Optimisation

This chapter shows the means by which sequences may be designed and optimised for use in a stored sequence fractional-N synthesiser. Obtaining such sequences is complicated in a number of ways due to the nonlinear nature of the sigma delta modulator and the inherent truncation of these sequences when used in a stored sequence architecture. The nonlinearity results in difficulties in applying a stability analysis which is a necessary prerequisite to determine the optimal noise shaping of the system, whilst the truncation can introduce severe distortion into the spectrum which has the capability to create many new low frequency components. The distortion effects can be particularly severe as there is no opportunity to add windowing to the single bit sequence downloaded into the memory. The avoidance of the spectral distortion is of particular relevance in the use of such a synthesiser in continuous rather than frequency hopping mode, where it is desired to feed the sequence to the divider control input in a continuous loop. This chapter illustrates how an adaptive iterative method may be used to verify large signal input stability of the modulator across a range of DC inputs, once the initial design parameters of the filter have been determined using a small signal nonlinear stability analysis. Following this it is shown how by measuring the repetition length of all the modulator output sequences the optimal sequence length can be calculated to avoid spectral distortion.

The nonlinear nature of the quantiser element of the sigma delta modulator precludes the use of standard stability analysis techniques as the principles of superposition do not apply. Assuming there is an unbiased (i.e. at a DC level corresponding to halfway between the two quantiser outputs and corresponding to a fractional division ratio in synthesiser applications of N+1/2) sinewave input to the quantiser, the output of the quantiser will be a squarewave at the same frequency. This means that a true description of the quantiser element would have to take into account the harmonics at the output, which clearly

invalidates conventional analysis. The approach used in this chapter is the approximate describing function method [11] which approximates the gain of a nonlinear element by its gain at the fundamental frequency of its output. This initial small signal analysis using the describing function developed by Engelen [17] is used in combination with root locus techniques to provide a starting point for the design of the loop filter.

Following the small signal stability analysis an adaptive iterative technique is used to determine the large signal stability of the modulator by simulation. Large signal stability is particularly important in Fractional-N applications if the full N to N+1 range of the dual modulus divider output range is to be used for synthesis. Once stability margins are found over the DC input range this data is extracted and used to simulate all of the sequences for each channel of the synthesiser. It is shown that Hanning windowing applied to an arbitrary length modulator sequence may be used to reveal the noise shaping of the modulator.

The final part of the chapter discusses the impact of truncating the sequence and shows how the optimal length may be detected and used to avoid the negative effects. Rectangular windowing is used to demonstrate the poor low frequency noise shaping of arbitrary length modulator sequences. The concluding part of the chapter shows how autocorrelation of the sequences with a subset of themselves can be used to detect repetition length and how selecting the sequence length for the synthesiser as the lowest common multiple of all the sequences avoids the spectral distortion effects of truncating the sequence.

6.1 Nonlinear Stability Analysis

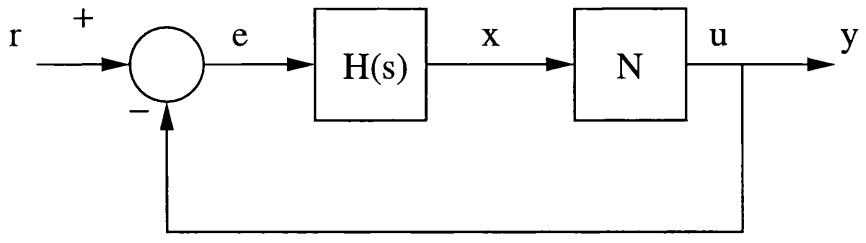


Figure 6.1: A Basic Nonlinear System.

The approach of describing function methods is to represent a non-linearity with the best possible linear approximation for a particular type of input. This representation is then adjusted depending on the size of the input. Figure 6.1 shows a feedback system consisting of a linear loop filter H and a nonlinear element N. The output of the nonlinear

element N will consist of the input signal from the loop filter H(s) plus any additional distortion produced by the nonlinear element. The distortion produced by the element N will be dependent on both the input to the system and the loop filter transfer function H(s).

In general describing function techniques rely on the fact that the linear system H(s) in figure 6.1 will be lowpass in nature, a reasonable assumption for many real world control applications. The assumption then is that only the fundamental component of the quantiser output needs to be accounted for, as the other higher frequency components will be much smaller, will be suppressed by the loop filter when they pass around the loop and will have much less of an effect on the system stability [20]. In some cases, particularly those where the second harmonic of the nonlinearity output is of significant amplitude, it may be necessary to take into account the additional effect on the stability or allow a greater margin of error.

As well as affecting the amplitude there may also be a delay associated with the nonlinearity which may or may not be frequency dependent, and which will result in a complex describing function.

Although the accuracy of describing functions for the purpose of stability analysis depends on the surrounding linear part of the system, they provide a reliable starting point for the design process and insight into the stability of limit cycles in the system.

In order to analyse the stability of the system a good starting point is to determine if the system is stable under conditions of zero input. For this purpose the Sinusoidal Input Describing Function (SIDF) is often used [20]. This approximates the nonlinear element transfer function for sinusoidal input signals in order to allow classical root locus techniques to be used for the stability analysis.

6.1.1 Describing function for a Relay

A common nonlinear element used in control systems is a relay. Figure 6.2 shows the describing function for the case of a non ideal relay for the nonlinear element N in the system of figure 6.1. And figure 6.3 shows the time domain switching operation. As the relay switches at thresholds V_{TH} and $-V_{TH}$ the hysteresis will give rise to a frequency independent phase shift in addition to the amplitude response. Clearly the gain of the relay will be inversely proportional to the input amplitude with fixed output levels.

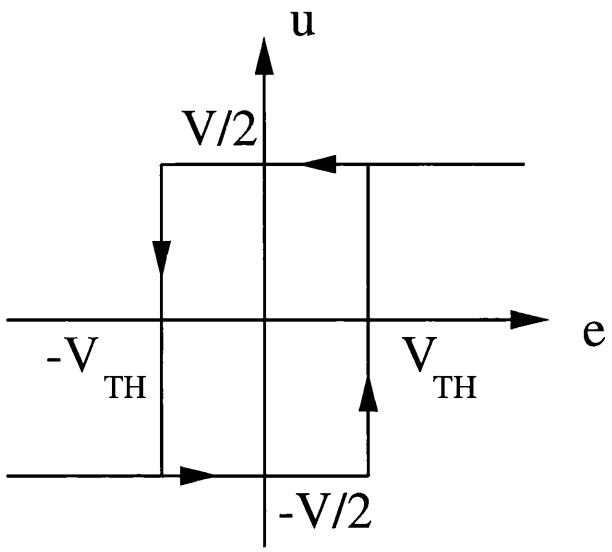


Figure 6.2: Nonlinearity due to a relay.

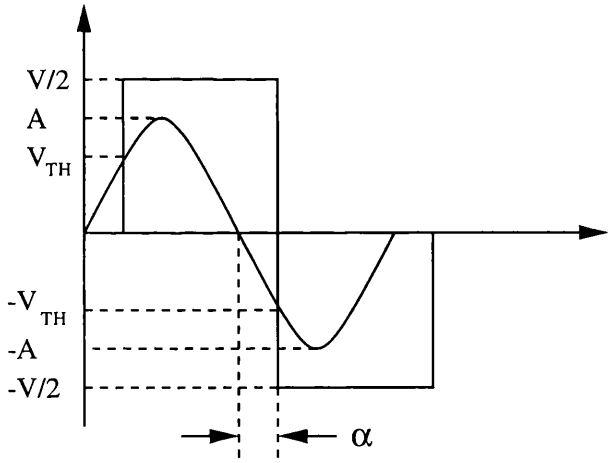


Figure 6.3: Phase shift due to a relay nonlinearity.

Referring to figure 6.1, the describing function (DF) of the nonlinearity is the ratio:

$$N = \frac{u}{x} \tag{6.1}$$

And for a sinusoidal input to the relay $x = A\sin(\omega t)$, the output from the nonlinearity can be described by a Fourier series with the following fundamental coefficients for u:

$$a_1 = \frac{\omega}{\pi}\int_0^{2\pi/\omega} u \sin \omega t \, dt \tag{6.2}$$

$$b_1 = \frac{\omega}{\pi} \int_0^{2\pi/\omega} u \cos \omega t \, dt \tag{6.3}$$

If the DF can be expressed as an odd function of time then the b_1 coefficient is zero, leading to the following expression for the amplitude response of the DF [11]:

$$|N| = \frac{a_1}{A} \tag{6.4}$$

This describing function has an equivalent gain which depends on the input amplitude but not on the input frequency. Solving 6.2 allows an expression to be written for the describing function amplitude in terms of the signal and DF levels:

$$a_1 = \frac{2}{\pi} \int_0^{\pi} D \sin t \, dt = \frac{4D}{\pi} \tag{6.5}$$

$$|N| = \frac{4D}{\pi A} \tag{6.6}$$

The phase shift in the DF, α can be determined from inspection of figure 6.3 and is given by equation 6.7. Hence equation 6.8 gives the full SIDF for the relay with backlash.

$$\alpha = sin^{-}1(\frac{V_{TH}}{A}) \tag{6.7}$$

$$N = \frac{4D}{\pi A} e^{-\alpha}. \tag{6.8}$$

Using the describing function of 6.8, although an approximate result, allows much insight to be gained into the operation of the nonlinear system. A particular advantage is that an analysis may be performed to find the possible oscillation frequencies of the closed loop system. These oscillations are known as limit cycles and are inherent to the operation of all nonlinear systems. Once the possible limit cycles have been determined, further analysis can then show their stability. This allows a good starting point for the system design as the autonomous system stability can be verified under zero input conditions before large signal design considerations.

The accuracy of the result will depend on the characteristics of the linear loop filter G(s), as this will determine the amount of filtering of the higher order components of the nonlinearity. In the case where there are no even harmonics in the DF output this will also considerably improve the accuracy of the describing function.

6.1.2 Limit Cycles

For a standard linear system, root locus techniques can be used to select an appropriate system parameter which will fix each system pole at the desired position on the root locus. To ensure stability this means placing all poles on the left hand side of the s plane or placing all poles inside the unit circle of the z plane in the case of a digital system.

In the case of a nonlinear system a root locus analysis will commonly use the gain of the nonlinear component as the variable parameter. The variable gain of nonlinear system blocks acts to move the poles along the root locus during the course of normal system operation. The periodic oscillations which occur in the system due to this pole movement are known as limit cycles and are an inherent part of the operation of nonlinear systems.

The use of describing functions in the stability analysis of nonlinear systems allows the prediction of limit cycles within the system, whilst a root locus plot reveals if such an oscillation can be maintained in the loop under zero input conditions.

Referring to the system diagram of figure 6.1, the closed loop system gain is given as usual by:

$$\frac{y(s)}{r(s)} = \frac{N(s)H(s)}{1 + N(s)H(s)} \tag{6.9}$$

Where N(s) denotes the nonlinear part of the system. The poles, determining the system stability are given by the characteristic equation:

$$1 + N(s)H(s) = 0 \tag{6.10}$$

The conditions for a limit cycle in a nonlinear system are the same as those for a sustained oscillation in any closed loop negative feedback system. To design an oscillator the gain

round the loop must be equal to -1 and the phase shift 180^o. This is the *Barkhausen criterion.*

In the case of a limit cycle there must be unity gain around the loop and one or more of the poles of the closed loop system will lie on the $s = j\omega$ axis. As $s = \sigma + j\omega$ if a constant amplitude limit cycle exists then the decay constant will be equal to zero. Solving the characteristic equation for constant amplitudes results in the condition for a limit cycle to exist:

$$N(j\omega)H(j\omega) = -1 \tag{6.11}$$

In the case of a digital system a constant amplitude limit cycle can only exist on the unit circle of the z plane, i.e. where $z = e^{j\theta}$, where θ is frequency normalised to the reference. This results in the following condition for limit cycles in digital systems:

$$N(e^{j\theta})G(e^{j\theta}) = -1 \tag{6.12}$$

This assumes the steady state case where z is of unit magnitude. Both of these conditions can be partitioned into separate modulus and phase requirements:
An analogue system:

$$|N(j\omega)|.|G(j\omega)| = 1 \tag{6.13}$$

$$\angle N(j\omega) + \angle G(j\omega) = \pi \tag{6.14}$$

A digital system:

$$|N(e^{j\theta})|.|G(e^{j\theta})| = 1 \tag{6.15}$$

$$\angle N(e^{j\theta}) + \angle G(e^{j\theta}) = \pi \tag{6.16}$$

6.1.3 Limit Cycle Stability

Figure 6.4 shows a hypothetical root locus plot for a continuous time nonlinear conditionally stable system with a single-valued saturating nonlinearity. An ideal relay with no backlash falls into this category in that the describing function $N(j\omega)$ is real and decreases with increasing input amplitude A.

The points where the root locus crosses the $s = j\omega$ axis denote possible limit cycles in the system, although some of these limit cycles may not be sustainable in practice.

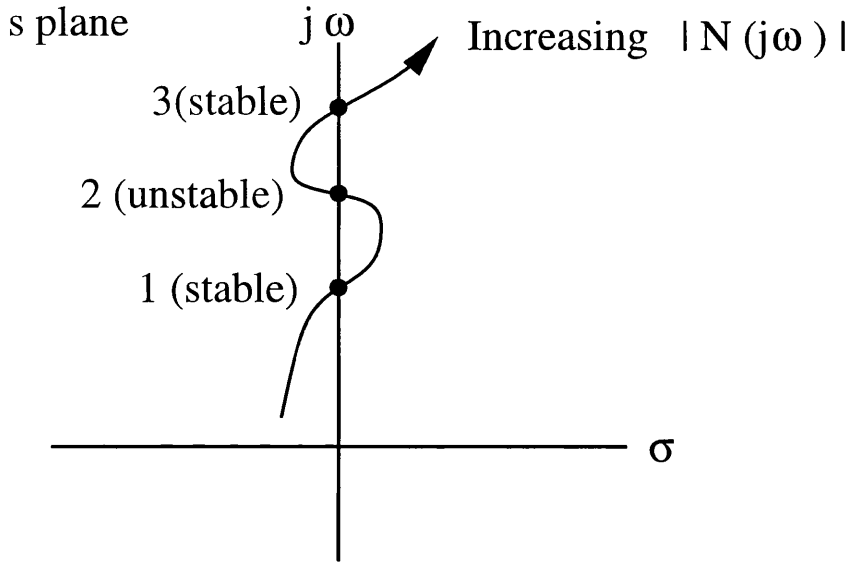


Figure 6.4: Root Locus plot for a non linear system.

The arrow on the root locus indicates the direction of increasing gain of the nonlinear describing function $|N(j\omega)|$. Referring to figure 6.1, assuming the system is turned on with zero initial condition (x and r equal to zero), at the instant of turn on the gain of the nonlinearity will be effectively infinite as the signal x at the input to the nonlinearity will have zero amplitude. The signal at e will grow due to the difference between u and r, and consequently the signal at x will grow leading to a reduction in $|N(j\omega)|$. Referring to the root locus, this reduction in the DF gain will occur until the gain rests at point 3. This corresponds to a stable limit cycle at the frequency of intersection with the $j\omega$ axis.

If a large enough gain reduction occurs in the nonlinear element that state 2 is reached from state 1, the system will converge to state 1 as state 2 is a divergent limit cycle. With the system at state 2 any slight increase in the input to the nonlinear element N will lead to a reduction in the gain of N, which will in turn will move the system poles further into the right half of the s plane along the root locus. Movement of the system poles further to the right will act to increase the input to the quantiser and the process continues until stabilisation at point 1.

Any decrease in N gain once at point 1 will have the stabilising effect of moving the poles further into the left half of the s plane, therefore once state 1 is reached there is no return to another state. Practical factors which cause switching between limit cycle states are disturbances at the system input and any changes in system parameters. Removal of the disturbance can never return the system to the original state.

6.2 Quantiser Describing Function

This section describes an approximation for a describing function for a single bit quantiser. A detailed analysis of the approximations derivation can be found in [18]. The notable feature of the describing function is the frequency dependent phase uncertainty, which arises due to the sampled nature of the system and has a profound effect on the closed loop stability. This results in many possible root locuses depending on the effective phase shift and frequency through the single bit quantiser. The phase shift is an entirely uncertain parameter as it results from the ambiguity between consecutive samples. Therefore the worst case, i.e. maximum, possible phase shift through the quantiser is used in order to allow a worst case root locus to be plotted.

6.2.1 Amplitude Response

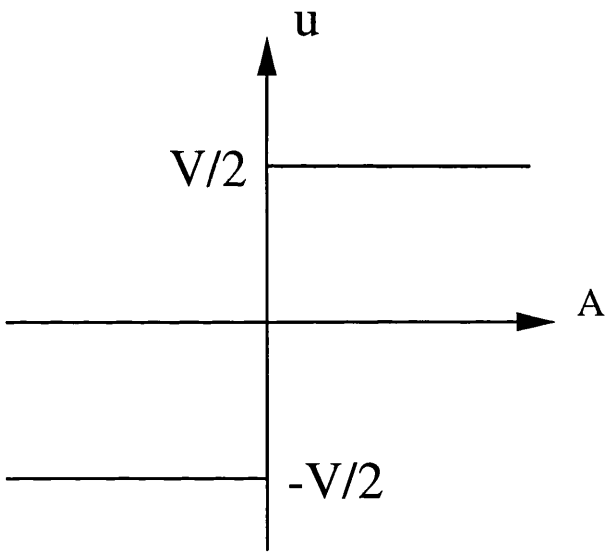


Figure 6.5: Single Bit Quantiser Nonlinearity amplitude response.

If the nonlinearity of system 6.1 is a single bit quantiser, then figure 6.5 shows the amplitude response. The output of the quantiser is the Fourier series of a pulse train at the input frequency multiplied by some delay. For a single sinewave input to the quantiser, the pulse train at the quantiser output can be described by a complex exponential Fourier series:

$$u(t) = \sum_{m=-\infty}^{\infty} X_m e^{m\omega_0 t}. \tag{6.17}$$

Where u(t) is represented as the sum of components of frequency $m\omega$ and X_m is a complex quantity giving the gain and phase of each Fourier component. A pulse train Fourier series is given by equation 6.18, which results from the parameters depicted in figure 6.6. The resulting periodic spectrum is shown by figure 6.7.

$$X_m = \frac{V\tau}{T_0} \frac{sin(\frac{m\omega_0\tau}{2})}{\frac{m\omega_0}{2}} \tag{6.18}$$

Thus the amplitude of a particular component of the pulse train is given by

$$X_m = (-1)^{m+1} \frac{V}{m\pi}. \tag{6.19}$$

where V is the peak to peak amplitude of the pulse train. The height of the X_0 component depends on the duty cycle of the squarewave and the DC offset and for a DC offset of half the squarewave amplitude this term is given by

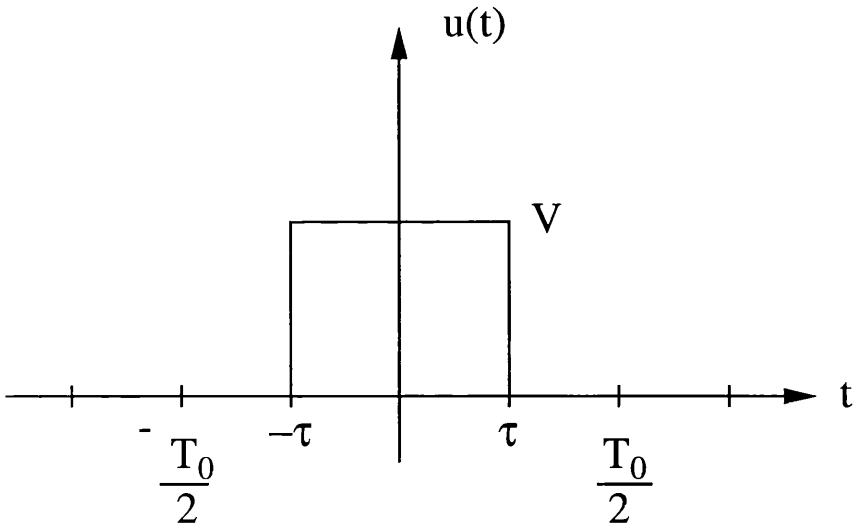


Figure 6.6: Squarewave output definition.

$$X_0 = V\tau/T_0. \tag{6.20}$$

Figure 6.7 gives an indication of the approximations involved when using a describing function for an ideal quantiser. In the case of there being zero DC bias in the nonlinearity, the X_o component in the output will be zero amplitude and there will be no even harmonics in the output. The most significant component other than the fundamental will be

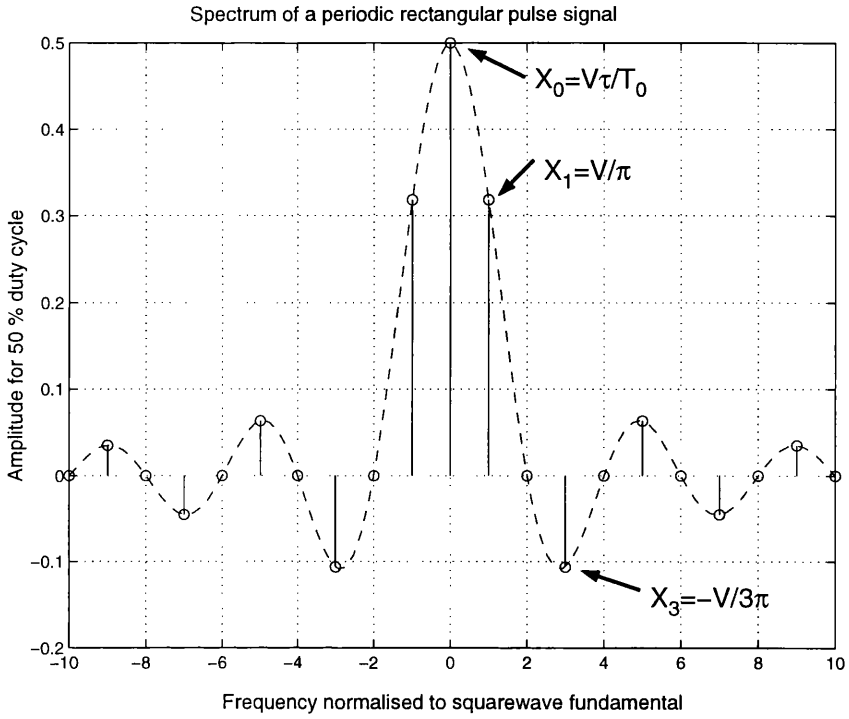


Figure 6.7: Spectrum of Periodic rectangular pulse.

the third harmonic, which is approximately one third of the amplitude of the fundamental.

The third harmonic is likely to be considerably attenuated by G(s) by the time it returns around the loop to the quantiser input. In particular, if G(s) contains a single ideal integrator there will be a 20 dB /decade attenuation of both the X_3 and X_1 components around the loop, which will reduce the amplitude of the X_3 component relative to the X_1 component by 9.54 dB. This means that the X_3 component will be only $1/9$th of the amplitude of the X_1 component or -19dB. Higher order low pass loop filters will yield correspondingly more accurate results from the describing function as the higher order nonlinearity outputs can be attenuated more relative to the fundamental.

For the purposes of the root locus plot the quantiser is simply assigned a global gain λ which is the gain of the quantiser for sinewave input at the fundamental frequency. For an input sinewave amplitude A and an output squarewave amplitude B, λ is simply B/A. This linearised gain is exactly analogous to the K parameter varied during a linear system root locus analysis.

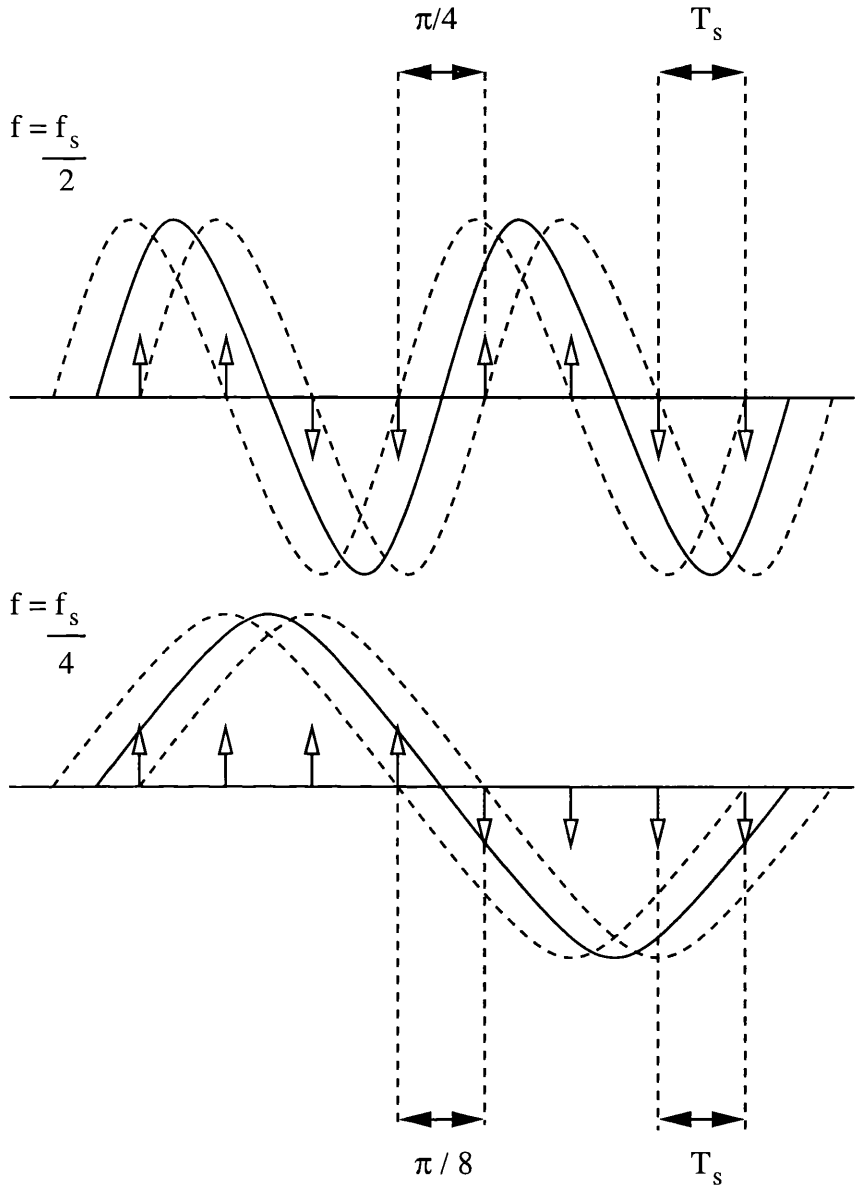


Figure 6.8: Bounds of the Phase Uncertainty due to a single bit quantiser.

6.2.2 Phase Response of a single bit quantiser.

Figure 6.8 Illustrates a worst case approximation for the bounds of the phase uncertainty when a sinusoid passes through a single bit quantiser [17, 18]. The single bit quantiser can only sample the sinusoid at certain times which results in a frequency dependent ambiguity. The size of the bounds of the ambiguity is proportional to the sampling period.

As shown in figure 6.8, the size of the bounds of this phase uncertainty depend on the relationship between the sinusoid frequency and the quantiser sampling frequency. The input signal is represented by the solid line which can be shifted in time without affecting the output of the quantiser, with the dotted lines showing the extremes of the input phase

possible for the same output.

Plotting the maximum bounds of this phase uncertainty results in the graph of figure 6.9. It should be noted that in this case the x axis of the graph refers to normalised frequency in a digital system. As the frequency axis in the z plane is arranged around the unit circle this means that π corresponds to the Nyquist frequency. The maximum value of phase uncertainty occurs at a frequency of $f = \pi/2$ where normalised frequency θ is given in terms of π and $\theta = \pi$ is the Nyquist frequency on the Z plane. The phase uncertainty through the quantiser can be written as

$$\Delta\phi = \alpha(|\theta + \frac{\pi}{2}| - |\theta - \frac{\pi}{2}|) \tag{6.21}$$

where

$$-\frac{1}{2} \leq \alpha =\leq \frac{1}{2} \tag{6.22}$$

Where the extremes of the possible phase offset occur when $|\alpha| = 1/2$. In using this describing function for root locus evaluation of stability the main point of interest is the root locus of the outer most closed loop poles and their relationship to the z plane unit circle.

As α may have a range of unknown values this converts the root locus into a continuous area on the z plane. For this reason worst case scenario value of phase shift is required to reduce the area to a single locus. This will occur when α is equal to -1/2, as this will reduce the phase margin of the system by the maximum amount.

The full describing function of the quantiser, Q(z), is written combining the gain of the quantiser λ with the value of phase uncertainty

$$Q(z) = \lambda e^{j\Delta\phi(\theta)} \tag{6.23}$$

In use, the worst case frequency dependent value of $\Delta\phi(\theta)$ can be obtained by choosing $\alpha = -1/2$ and the root locus is then plotted for all values of λ. A limit cycle analysis can then be performed for the worst case stability scenario.

6.3 Loop Filters

The loop filter determines the noise shaping characteristics of the sigma delta modulator and for phase locked loop applications should be designed to allow maximum noise

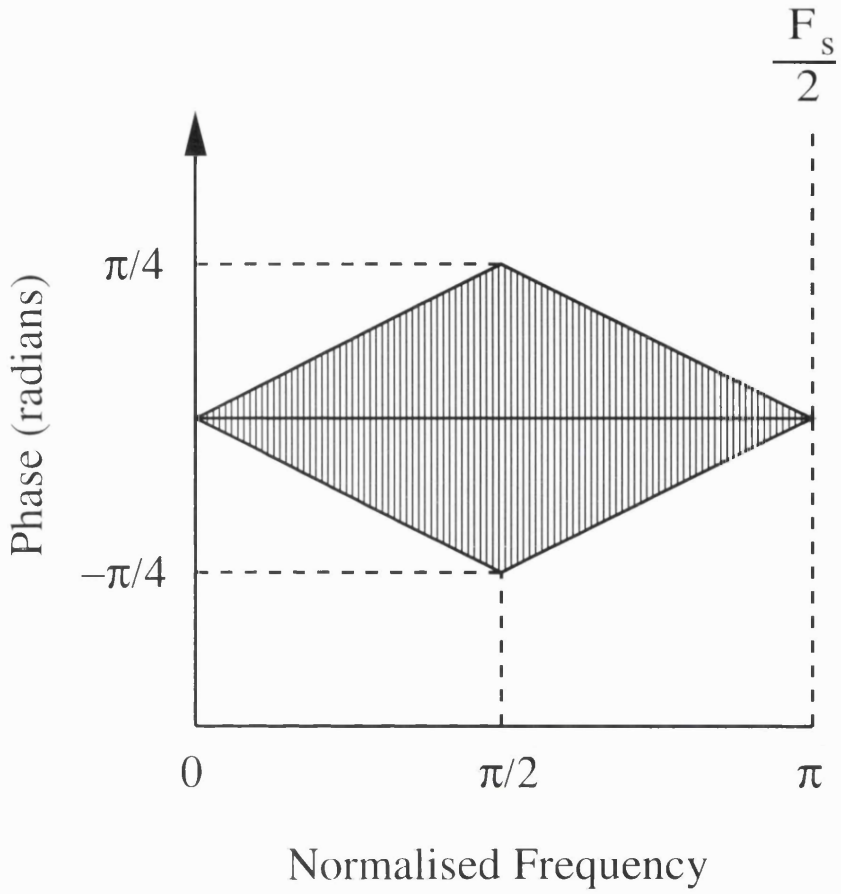


Figure 6.9: Bounds of the Phase uncertainty through a Single Bit Quantiser.

reduction at low frequencies consistent with stability. One type of filter recently used in sigma delta modulator applications is that of a type described by Stikvoort [16]. This loop filter is described by the following equation:

$$G(z) = \frac{(z-a)^n}{(z-b)^n} - 1 \tag{6.24}$$

Where n is the order of the loop filter, a determines the zero frequencies and b the pole frequencies. All zeros and poles of this filter are at the same frequency which leads to a single root locus when the closed loop root locus is plotted of a system incorporating this filter.

The response of this filter is not well optimised for data conversion applications as all of the poles of G(s) are at zero frequency. It has been shown that an improved shape of response can be obtained in terms of improved SFDR over a wider band by spreading the

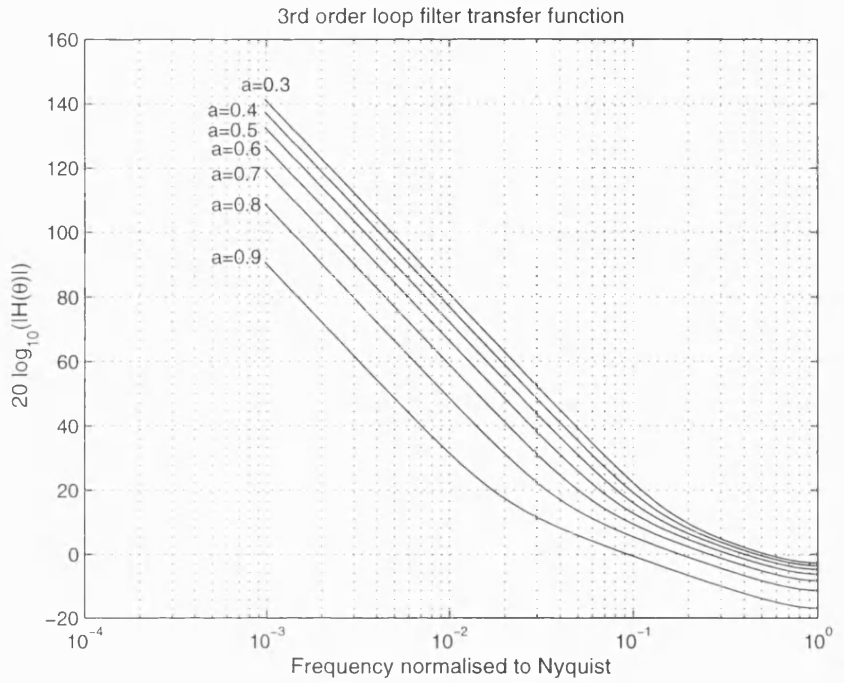


Figure 6.10: 3rd order loop filter transfer function with varying zero positions.

loop filter poles over the signal band [10]. In indirect frequency synthesis applications the spectrum of the modulator output is to be subject to further filtering by the closed loop response of the synthesis loop, therefore the importance of a very wide signal band is lessened making this form of filter response an attractive candidate.

In the case that the reference frequency of the sigma delta modulator is very high relative to the loop bandwidth, as becomes possible using offline generation of the modulator sequence, this results in most of the noise suppression requirement of the modulator being over a very small fraction of the modulator spectrum close to zero frequency. Therefore this type of loop filter should be expected to be better suited to PLL applications than A/D conversion.

For a practical loop filter for a lowpass sigma delta, the b coefficient is set to 1 in order to place the poles at DC and a is made as high as possible consistent with the stability of the closed loop system. The larger the size of a, the closer the filter zeros will occur to the Nyquist frequency, with a=1 being consistent with zeros at the Nyquist frequency. The higher the a parameter can be made the higher the quantiser noise suppression will be at low frequencies as the roll off of the loop filter will start at a higher frequency. This is shown in figure 6.10 which shows the loop filter magnitude response for a number of different values of the a parameter. Figure 6.11 shows the magnitude response of a second

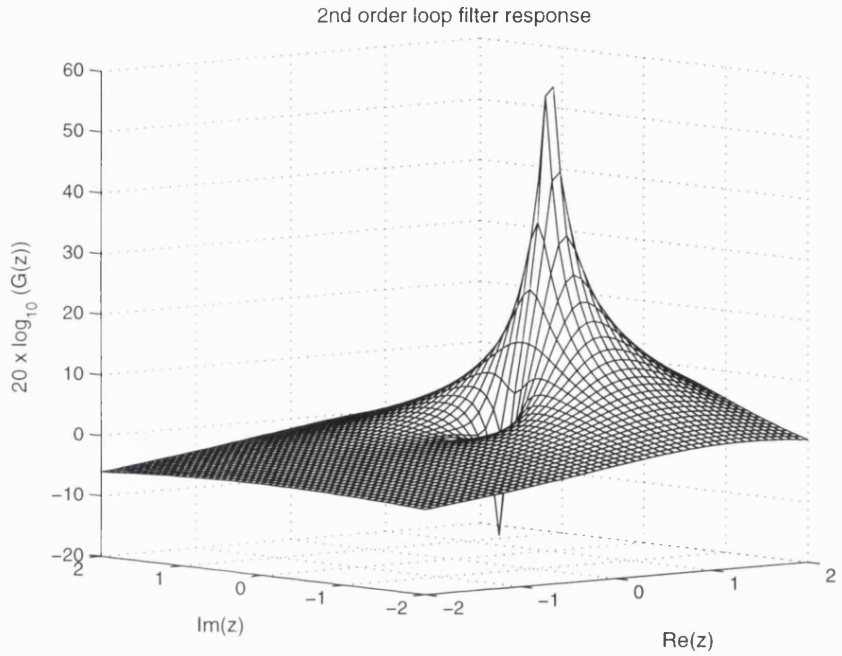


Figure 6.11: Z domain plot of 2nd order loop filter response.

order version of this loop filter in the z plane, illustrating the poles at z=1.

For a third order filter the difference equation required for implementation can be obtained from rewriting equation 6.24:

$$G(z) = \frac{(3b - 3a)z^2 + (3a^2 - 3b^2)z + (b^3 - a^3)}{z^3 - 3bz^2 + 3b^2z - b^3} \tag{6.25}$$

Multiplying this by z^{-n} then gives the difference equation for the filter:

$$G(z) = \frac{(3b - 3a)z^{-1} + (3a^2 - 3b^2)z^{-2} + (b^3 - a^3)z^{-3}}{1 - 3bz^{-1} + 3b^2z^{-2} - b^3z^{-3}} \tag{6.26}$$

In a standard direct form 2 digital filter structure, the numerator coefficients of equation 6.26 implement feedforward terms whilst the denominator coefficients implement feedback terms, leading to the structure of figure 6.13. A point worthy of note is that there is no b0 term as this would imply a path through the filter with zero delay. A non-zero b0 coefficient would create a contention in the closed loop of a modulator as there must always be at least one delay around the loop.

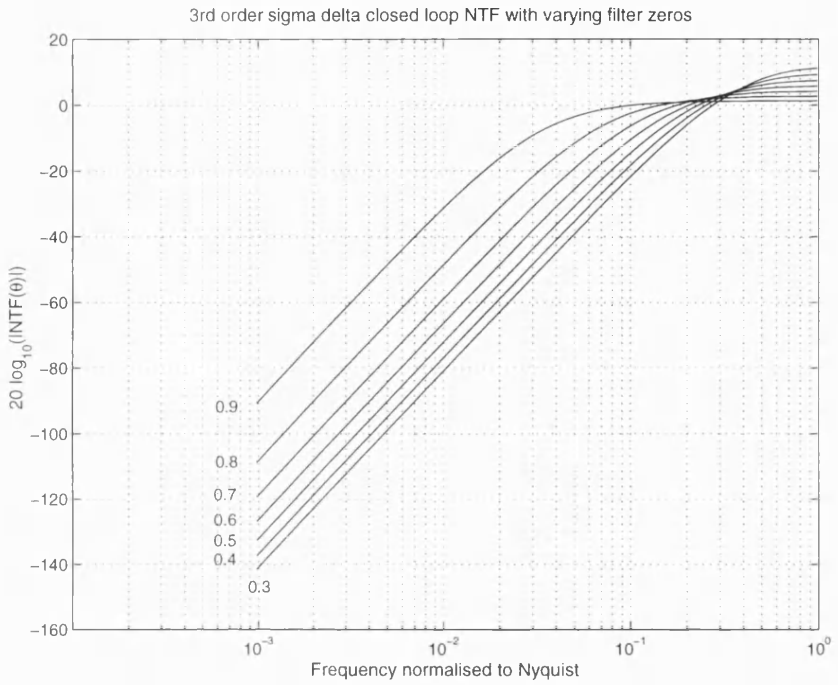


Figure 6.12: 3rd order sigma delta closed loop noise transfer function with varying filter zeros.

Figure 6.12 shows the variation in the closed loop noise transfer function (NTF) with the a parameter. This was derived by viewing the quantiser noise as an additive component which is uncorrelated with the loop dynamics. Clearly the larger the a value the higher the noise suppression of the filter and the better the final SFDR in synthesiser applications. It is also evident that the high frequency NTF gain becomes larger as a by product of the increased low frequency noise suppression, which suggests a trade off between stability margin and low frequency noise suppression.

6.4 Small Signal Modulator Stability

An estimate of small signal stability can be obtained easily by plotting the root locus of figure 6.1 under conditions of zero input signal. This is the first step in designing a stable system as autonomous stability of the loop is a prerequisite for stability under conditions of non zero input. The accuracy of the stability estimate will depend on the accuracy of the describing function approximation for the nonlinear quantiser element. This in turn will depend on the amount of suppression of the harmonics of the quantiser output by the linear filter G(z). This section shows small signal stability analysis of modulators using the filter described by Stikvoort [16] and the describing function technique mentioned previously. This approximate analysis provides a starting point for the large signal optimisation for the Fractional-N application.

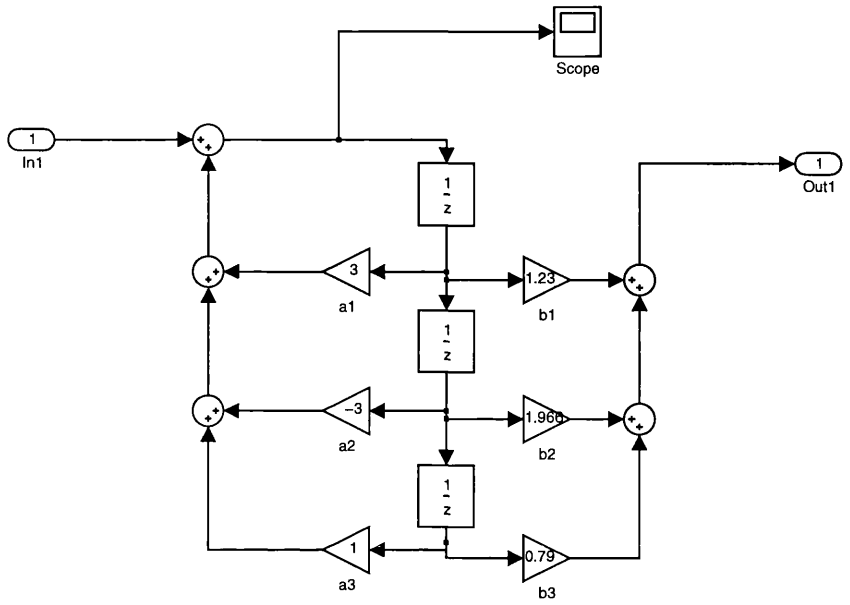


Figure 6.13: Third order loop filter subsystem in direct form 2 structure.

6.4.1 Stability Analysis of a Second Order modulator

For a second order modulator filter of the type described in equation 6.24, the difference equation for the filter response is given by the following equation:

$$G(z) = \frac{(2-2a)z^{-1} + (a^2-1)z^{-2}}{1 - 2z^{-1} + z^{-2}} \tag{6.27}$$

Placing the zeros of this filter at as high a frequency as possible, i.e. at the Nyquist frequency, can be accomplished by using a = 0, giving:

$$G(z) = \frac{2z^{-1} - 1z^{-2}}{1 - 2z^{-1} + z^{-2}} \tag{6.28}$$

This will result in the best possible noise transfer function. Combining this filter response with the quantiser describing function (equation 6.23) results in the following linearised description of the closed loop gain of figure 6.1:

$$CL(z) = \frac{G(z)Q(z)}{1 + G(z)Q(z)} = \frac{\left[\frac{2z^{-1}-1z^{-2}}{1-2z^{-1}+z^{-2}}\right]\lambda e^{j\Delta\phi(\theta)}}{1 + \left[\frac{2z^{-1}-1z^{-2}}{1-2z^{-1}+z^{-2}}\right]\lambda e^{j\Delta\phi(\theta)}} \tag{6.29}$$

The poles of equation 6.29 are given from the roots of the characteristic polynomial of the closed loop transfer function, which result from equating the denominator to zero:

$$1 + \left[\frac{2z^{-1} - 1z^{-2}}{1 - 2z^{-1} + z^{-2}}\right] \lambda e^{j\Delta\phi(\theta)} = 0 \tag{6.30}$$

In order to plot the root locus of the system it is desired to solve this equation for the closed loop poles at every possible value of the linearised quantiser gain λ. The values of the roots will be some complex value of z which depend on the real value λ and the frequency dependent value of phase uncertainty $\Delta\phi$. From the equation specifying the phase uncertainty bounds

$$\Delta\phi = \alpha(|\theta + \frac{\pi}{2}| - |\theta - \frac{\pi}{2}|) \tag{6.31}$$

where

$$-\frac{1}{2} \leq \alpha =\leq \frac{1}{2} \tag{6.32}$$

the worst case in terms of stability occurs with $\alpha = -1/2$. This worst case phase uncertainty can be described by

$$\Delta\phi = \frac{1}{2}(|\theta + \frac{\pi}{2}| - |\theta - \frac{\pi}{2}|) \tag{6.33}$$

Where θ is the frequency normalised to Nyquist. This worst case scenario corresponds to the lower bound of the uncertainty area shown in figure 6.9.

There is no analytical means of solving equation 6.30 because both the value of z to be solved for and the quantiser describing function Q(z) are complex quantities. This makes a numerical system of solution necessary as a complex solution of λ results from the characteristic equation solution.

Without the complex quantiser gain it would only be necessary to select a quantiser real valued gain and then scan the z plane surface in a systematic manner to solve the charac-

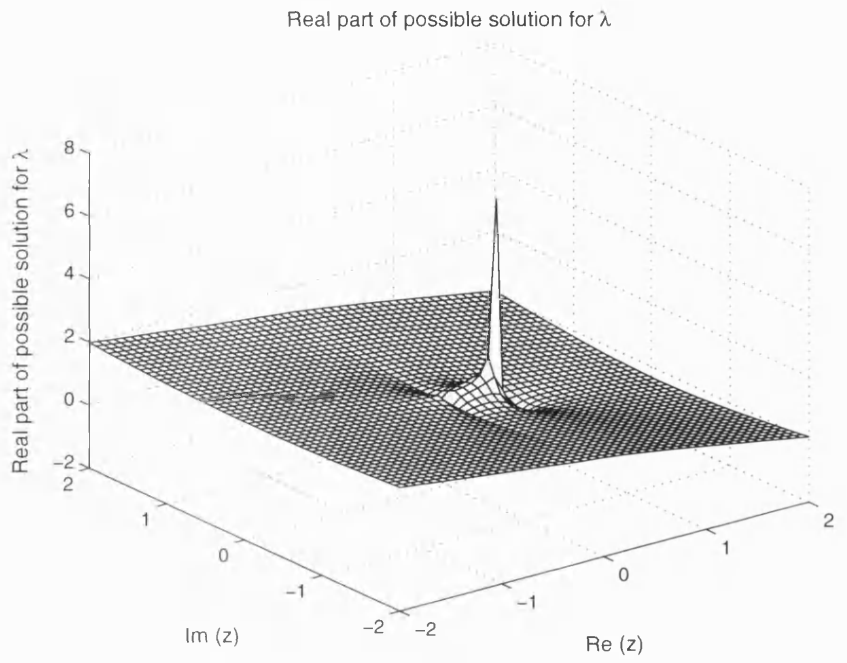


Figure 6.14: Real part of possible complex solution for λ from characteristic equation.

teristic equation for the complex root of the closed loop pole for that particular gain value. As the quantiser gain in this case is a complex quantity due to the frequency dependent phase uncertainty this approach still leaves a complex value of λ for each complex value on the z plane. The output of the scanning the z plane to solve the characteristic equation for λ is two surfaces plotted over the z plane which represent the possible real and imaginary solutions of λ as z is varied. Figure 6.14 shows the real part of the complex output whilst figure 6.15 shows the imaginary part. This result was obtained with the second order loop filter of equation 6.28 and the $\alpha = -1/2$ worst case of phase uncertainty.

In order to extract the root locus from the complex output of the scan over the complex plane, the case of interest is when λ is purely a real quantity representing the quantiser gain. The case of the imaginary part of λ being zero corresponds to a point on the root locus because all of the phase shift in the quantiser describing function is defined to be due to the phase uncertainty rather than the λ parameter. This means that the root locus may be found by finding those points on the z plane where the surface of figure 6.15 crosses zero.

Figure 6.16 shows the root locus obtained for the second order modulator after solving for a real value of λ. This plot illustrates the very large effect on the root locus of the quantiser phase uncertainty. The outermost plot corresponds to the worst case, maximum phase lag

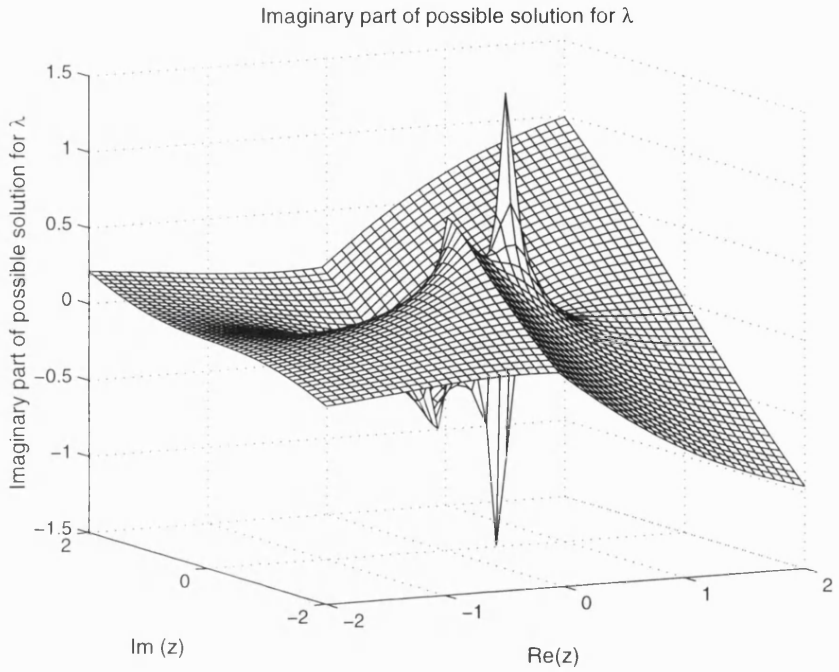


Figure 6.15: Imaginary part of possible complex solution for λ from characteristic equation.

uncertainty through the quantiser whilst the innermost plot represents the minimum phase shift. Intermediate plots show the effect of various values of the phase uncertainty parameter α for values between $\pm 1/2$. These plots were obtained using the simulation file of appendix D.11.

Referring to the outermost root locus of figure 6.16, it can be deduced that the second order modulator will be unconditionally stable because there is only one crossing of the z plane unit circle. Starting at the point where the locus intersects the unit circle, any increase in the signal level at the quantiser input will act to reduce the quantiser gain λ, which will result in the closed loop poles moving further inside the unit circle. This will be compensated for by the closed loop system poles being moved further inside the unit circle, in turn reducing the signal magnitude at the quantiser input until the quantiser gain returns to the original value. Alternatively, imagining a system at the point of turn on with zero initial conditions, λ will be infinite but will fall towards the equilibrium value as the signal inevitably grows at the loop filter output. As discussed in section D.11, the equilibrium limit cycle will exist at the point of intersection of the root locus and unit circle. In this case there may be any number of limit cycles due to the quantiser phase ambiguity, but the function of the root locus is to examine the most unstable possible condition of the modulator.

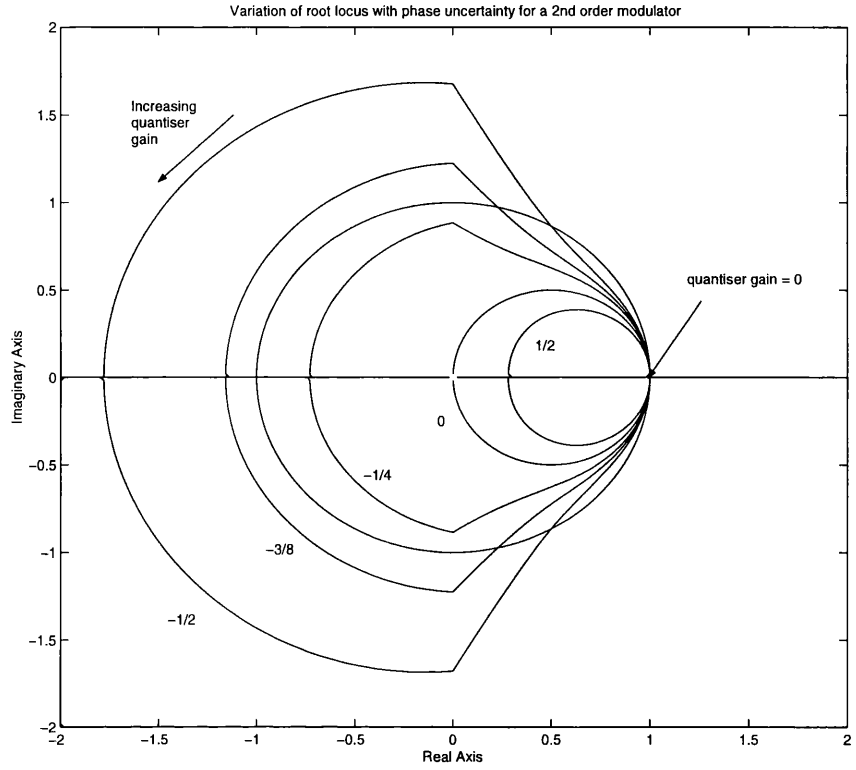


Figure 6.16: Effect on 2nd order modulator root locus of varying phase uncertainty.

6.4.2 Third Order Modulator Root Locus

Figure 6.17 shows a number of worst case stability root locus plots for a third order modulator with the loop filter of equation 6.26. The increased number of poles and zeros of this filter result in some differences in the shape of the root locus compared with figure 6.16. This results in the possibility of the modulator becoming stuck in a unoperational state if quantiser gain λ becomes too low. In addition, correct operation may be impossible if the loop filter G(s) is designed with its zeros at too high a frequency, as there may be no modulator states capable of sustaining a stable limit cycle.

The outermost root locus displayed on figure 6.17 is for the case of a third order filter with the a parameter set to 0.3. Clearly there is no value of quantiser gain λ at which this root locus crosses the unit circle apart from $\lambda = 0$. This means that there is no sustainable limit cycle. Taking the case of startup with zero initial output for the loop filter, the filter output will grow due to the non-zero quantiser output and the λ value will fall correspondingly. As the closed loop poles can only remain outside the unit circle on this locus, the gain of the closed loop system will remain in a divergent state and the filter output will continue growing until λ reaches a value of zero at the 0,-1 point on the z plane. At this point no operation of the modulator can take place and there will be no way out of changing the

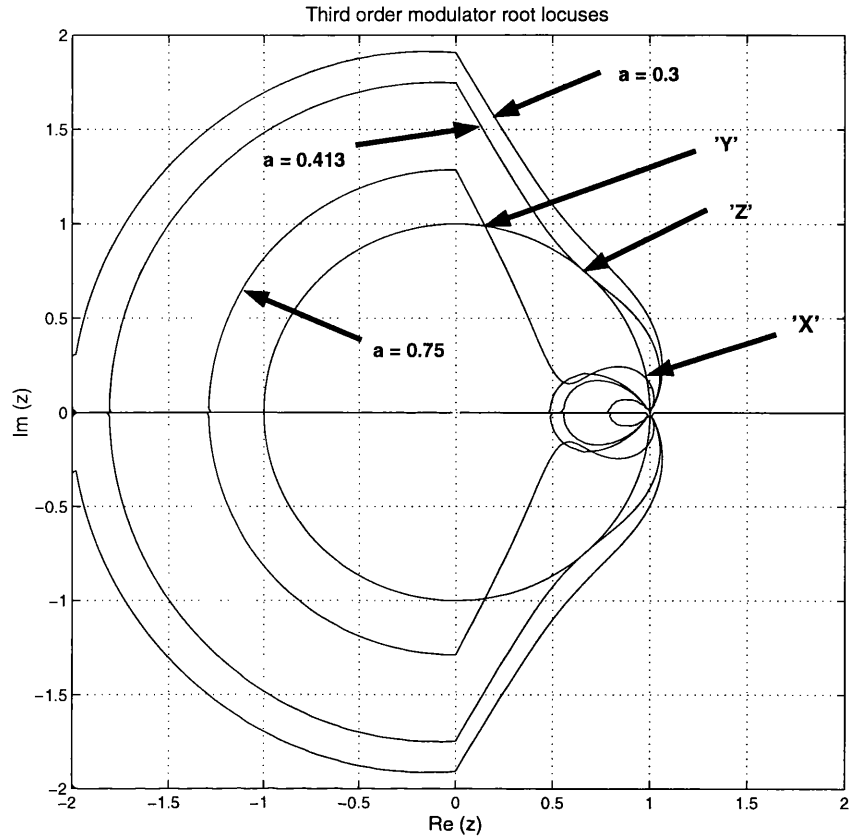


Figure 6.17: Third order modulator root locus with differing loop filter zero frequencies.

modulator state.

The innermost plot of figure 6.17 illustrates an alternative 3rd order loop filter with a much smaller a value of 0.3. This results in a conditionally stable modulator which will continue to run in a stable manner provided the value of λ does not go below a certain value, which is marked as 'X'. At switch on the λ value of this modulator will fall until a stable limit cycle is established at point 'Y' where the conditions for a limit cycle are satisfied. Once at this point any small deviation in lambda due to any disturbance will alter the system pole position in a way that will tend to re-establish oscillation at point 'Y'. However, at a certain minimum value of λ the root locus crosses outside the unit circle again. If λ falls below this critical value then it will continue falling until it reaches zero as the system poles will move further outside the unit circle. The system gets latched into an unusable state as before. As long as λ is kept above the critical value normal modulator operation will continue.

The root locus for a = 0.413 represents a theoretical optimum case. At switch on of the system a stable limit cycle will be established at point 'Z' which is the point at which the

root locus is an exact tangent to the unit circle. However, if for any reason λ falls below the value at point 'Z', then the modulator falls into a latch-up state as described before at the $\lambda = 0$ value of root locus. The difference in λ between the point at which the root locus enters the unit circle and the point at which it leaves the unit circle is analogous to the gain margin which is discussed in the stability of conventional linear systems. The larger the ratio of these two quantiser gains the larger the stability margin of the system and the less likely the system is to lock up under any external influence. The next section discusses the problems involved with the use of practical non linear systems with non-zero input signals.

6.5 Large Signal Modulator Stability

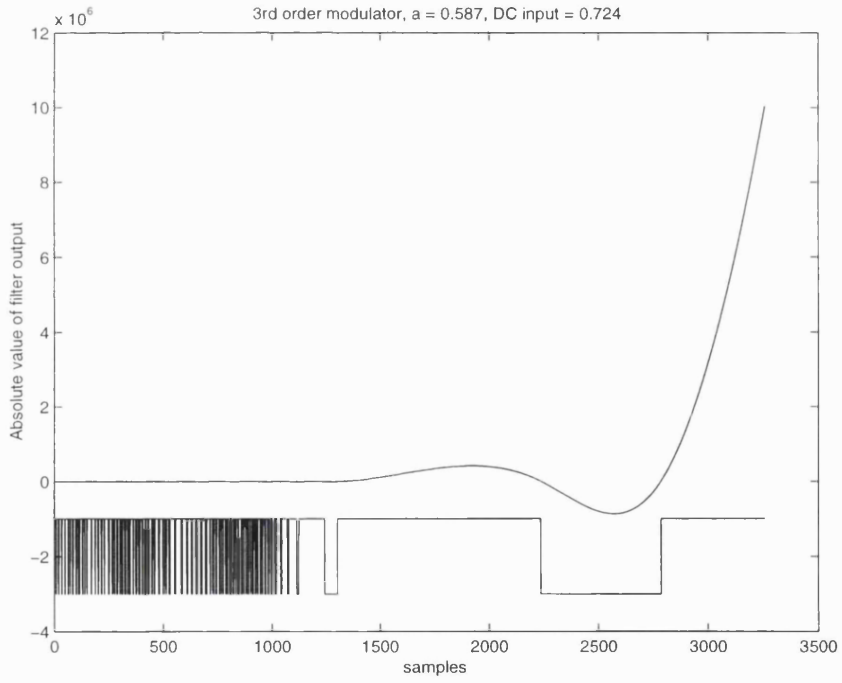


Figure 6.18: 3rd order modulator instability due to large DC input.

Although the small signal stability of a modulator can be estimated with a fair degree of reliability using the root locus method, the stability of the system is dependent not only on the poles of the loop filter, but also on the magnitude and frequency of the input signal. Consequently verifying that the closed loop system is stable under zero input conditions, whilst providing insight into system stability and a starting point for the design process, does not guarantee the large signal stability of the modulator.

6.5.1 Stability Checks by Simulation

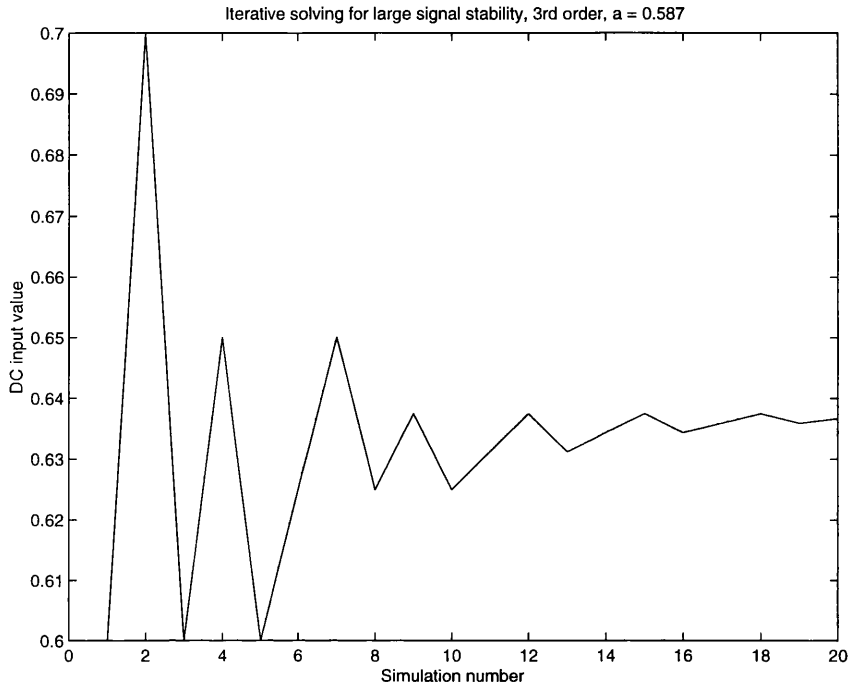


Figure 6.19: Iteratively finding the maximum stable DC modulator input.

The condition for stability is that all poles of the closed loop system must remain within the unit circle for the *minimum* value of quantiser gain. As applying an input signal to the modulator affects the size of the quantiser gain, then the modulator poles must remain within the unit circle for the largest required input signal.

A simple way of determining large signal modulator stability is by simulation, although as limit cycles may occur over extremely long periods this means that very long simulations are required in order to be confident of nonlinear system stability. Even with exhaustively long simulations, there is no actual proof of the long term stability making the design of such systems problematic.

In the case of the use of a modulator which will produce sequences off-line for use in a stored sequence synthesiser, stability of the modulator for a longer period than the required sequence length is not an issue. The offline generation of such sequences and subsequent storage avoids the requirement for recovery devices which are used to restart hardware modulators in the event of latch-up due to instability. For this reason simulation has been used as a simple method to determine the large signal stability of the third order modulator for sequence generation.

6.5.2 Instability Detection

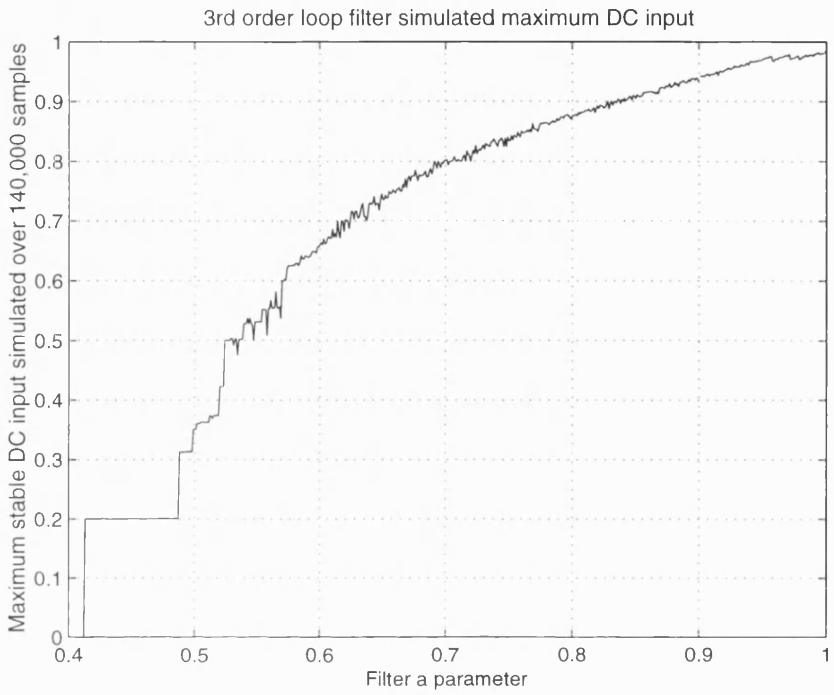


Figure 6.20: Simulated Maximum stable DC input with varying filter zeros.

Figure 6.18 illustrates the time domain operation of a modulator becoming unstable due to a large DC input signal. The lower trace of the graph indicates the single bit quantiser output whilst the y axis shows the absolute value of the signal at the filter output, shown by the analogue trace. In a stable modulator the signal at the filter output remains within very small bounds which are comparable to the signal levels at the quantiser output. When the signal at the filter output grows to a very much larger value then the stability of the modulator is assumed to be determined by a large signal limit cycle and the modulator can be considered unstable. During simulations the modulator was considered to be unstable when the signal at the filter output grew to over 1,000 times the quantiser output value. Simulations have shown that stably running system tends to have all states in a range comparable to the quantiser range, although the exact choice of threshold is arbitrary. For the purposes of sequence generation, a stable modulator was considered to result once the modulator had run without instability for 140,000 samples.

In addition to a stability test, a systematic search algorithm was used in order to allow unattended simulations where each successive simulation grew closer to finding a result

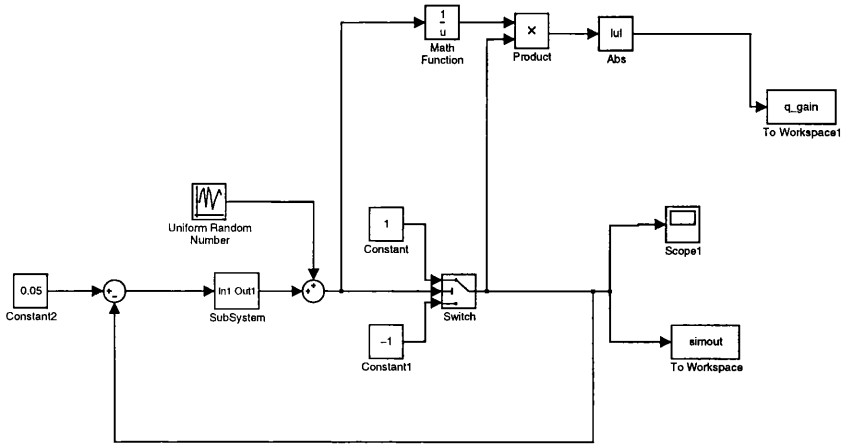


Figure 6.21: Simulation setup for measuring quantiser gain

for the maximum DC input value. The sequence of values created by 20 iterations of this search process is illustrated in figure 6.19. Whilst this process served to determine a maximum DC input value for a single modulator, it was also desired to find the most optimal modulator filter for each DC input, therefore a two dimensional sweep was performed which involved both the DC input and the a parameter of the filter response in equation 6.26.

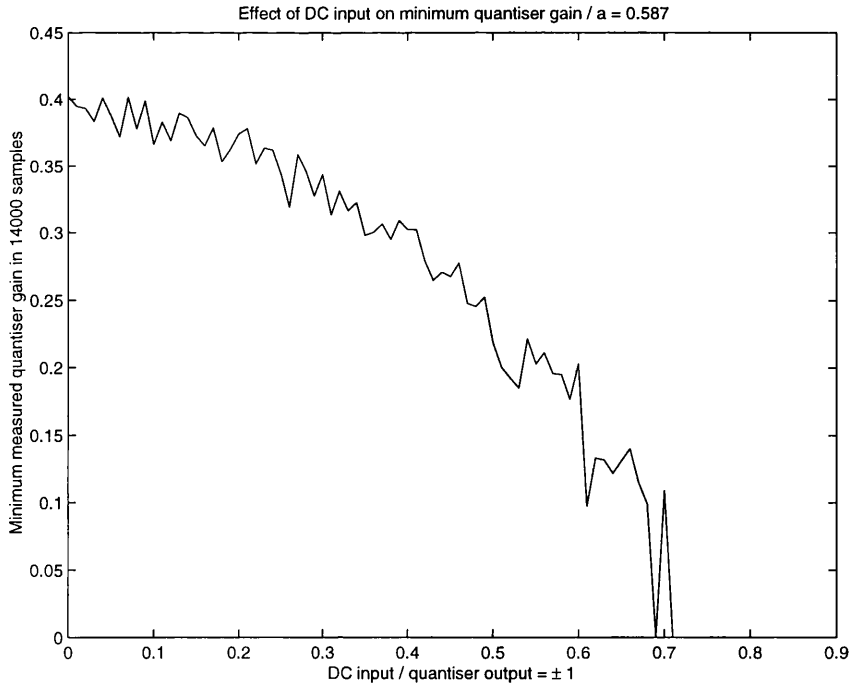


Figure 6.22: Relationship between minimum measured quantiser gain and DC input over 14000 samples.

Figure 6.20 shows the sweep results illustrating the variation in maximum modulator DC input against the loop filter a parameter, which determines the frequency of the loop filter zeros. There is a clear trade off between the achievable noise suppression performance of the modulator and the amount of possible DC input without instability. For Fractional-N synthesiser applications with a dual modulus divider this result means that it will be more difficult to obtain good noise shaping performance further away from the N+1/2 value of output frequency. This is an intuitive result as the more time the frequency divider has to spend at one particular value the less opportunity there is to modulate the switching noise to a higher frequency. In a stored sequence synthesiser application this is of little importance as the lack of the requirement for hardware data path elements allows much higher reference frequencies, meaning that the same output frequency swing may be obtained with a much smaller swing of the average divider range between N and N+1.

Figures 6.22 and 6.23 serve to illustrate the effect on the minimum possible quantiser gain λ of large signal DC inputs to the modulator. These results were obtained using a simulation run of 14,000 samples and the third order modulator shown in figure 6.21 with the filter topology shown in figure 6.13. The gain of the quantiser was measured for every sample and the minimum value over the simulation recorded. The comparison between the two figures shows how, by appropriate choice of loop filter zeros, the minimum gain under a large signal input can be increased. Figure 6.22 uses a value for a in the loop filter transfer function of equation 6.26 of 0.587, whilst using a value of 0.8 as in figure 6.23 clearly results in less reduction of the quantiser gain with a rising DC signal input and therefore a larger chance of stability for high input values.

6.6 Optimised noise spectra

This section illustrates how the modulator loop filter can be optimised according to the modulator DC input level. It is shown that, because of the dependence of the large signal stability on the level of DC modulator input, it is possible to obtain better noise shaping results at the centre of the synthesiser band near the mean divider ratio of N+1/2. It is also shown that by the use of a controlled amount of random noise introduced at the quantiser input the SFDR of the various synthesiser spectra can be increased across the frequency band. Because of the large signal modulator inputs it is not possible to rely on the small signal approximate design method involving the discussed modified root locus technique although this is successfully used as a starting point for the large signal optimisation pro-

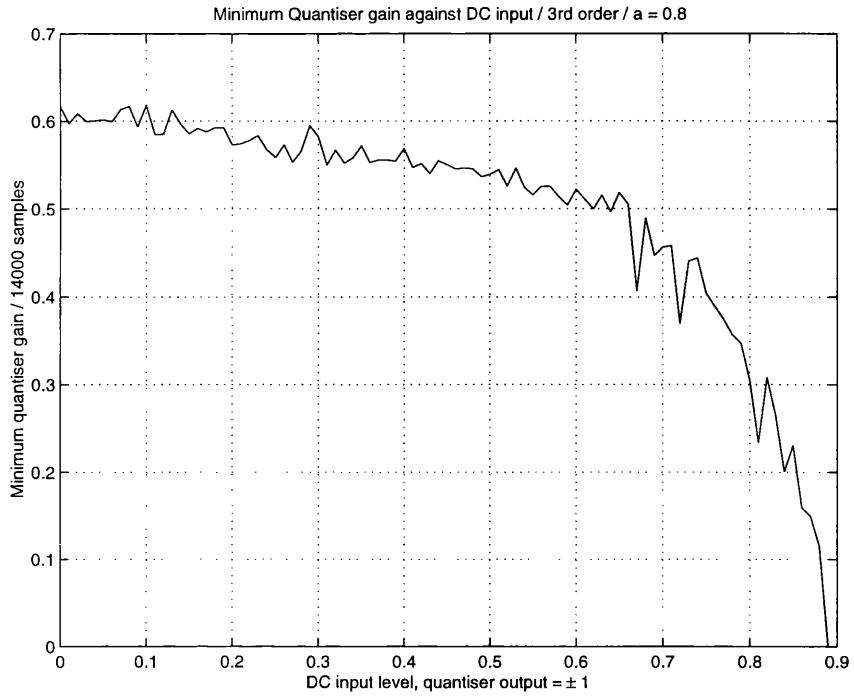


Figure 6.23: Minimum measured quantiser gain with increased system phase margin over 14000 samples

cess.

6.6.1 Truncated Sequence Spectra

By using the measured results for maximum stable DC input from figure 6.20, a modulator can be optimised to provide the maximum amount of noise shaping, consistent with stability, with the changing DC input to the modulator across the synthesiser band. Figures 6.25 and 6.26 show two examples of the optimisation that can be achieved by this process. In figure 6.25, the modulator output spectrum is shown for a DC input of 0.9 whilst varying the position of the filter zeros. The left hand plot of figure 6.25 shows the results for a=0.98 in equation 6.26, whilst the right hand plot shows the results for a=0.95. Clearly the right hand plot shows higher noise suppression near to DC. Figure 6.26 compared with figure 6.25 shows that an even higher gain in noise suppression can be achieved by optimising the zero values with a smaller DC input, as setting a=0.425 will not make the modulator unstable as there is less asymmetry introduced into the loop to disrupt the limit cycle.

The general result for maximum DC input for different loop filters can be used automat-

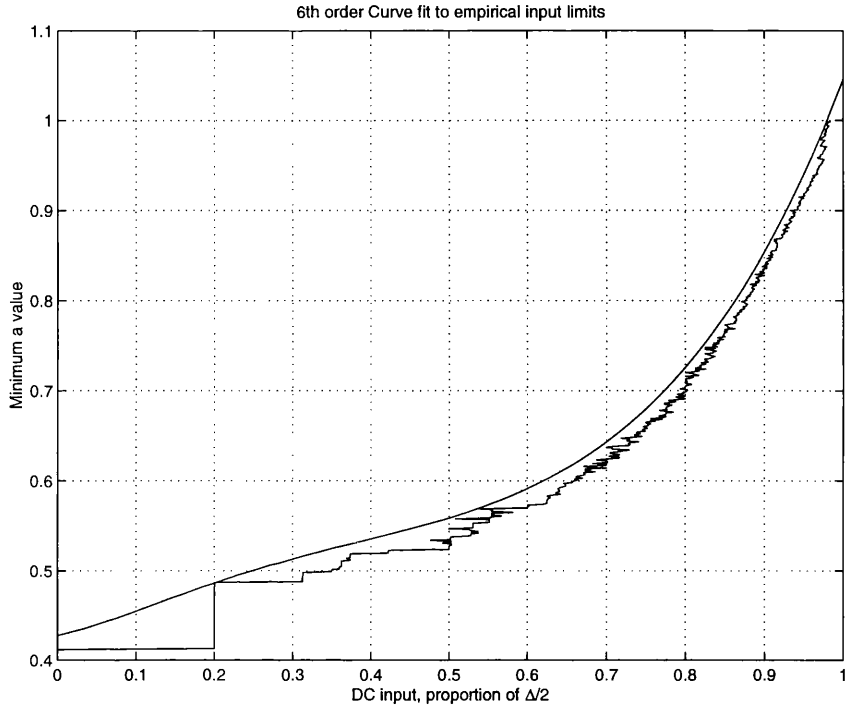


Figure 6.24: Curve fitting to empirical data on maximum DC input.

ically for optimisation purposes by performing a curve fit as shown in figure 6.24. A constant value is also added to the least squares fit in order to provide a stability margin. The extra margin used in the simulations was chosen empirically to be 0.11 after simulations across a range of DC inputs.

Figure 6.27 shows an ensemble of modulator spectra automatically generated by the above optimisation process. The spectra correspond to 50 DC modulator inputs in the range 0.1 to 0.9 normalised to $\Delta/2$ where Δ is the single bit quantiser output step size. Stable modulator running is achieved over the entire range of DC inputs as expected, although the noise shaping is degraded for some of the spectra. This effect can be attributed to the truncation of the sequence with a rectangular window. Hanning windowing of the modulator sequence illustrates that the modulator is operating correctly, illustrating that stable modulator running is not the only criterion required for good noise shaping if the sequence is to be truncated and looped.

One way to reduce the low frequency components is to use a very high reference frequency, which is easily achieved with a stored sequence type architecture. Figure 6.27 indicates a level of -70 dBc near to DC, but the level of these components can be made lower by using longer noise shaping sequences as discussed in section 6.6.3. Although

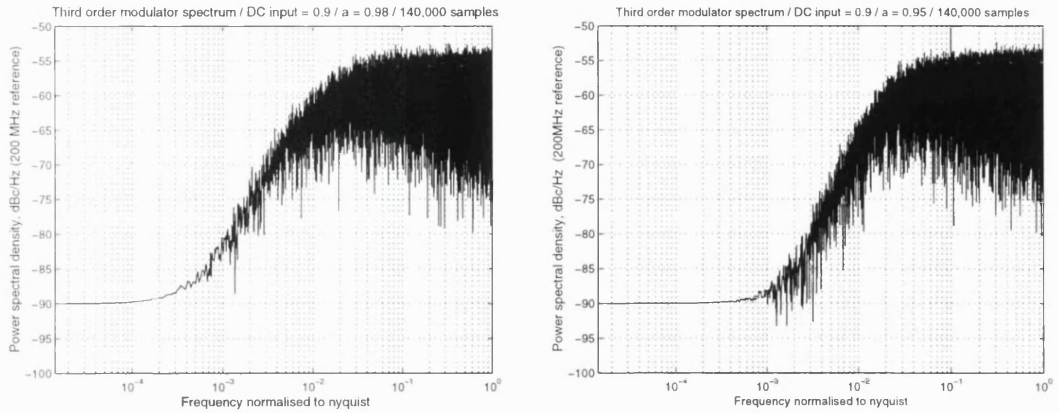


Figure 6.25: Modulator output spectrum with varying loop filter zeros position for DC input of 0.9.

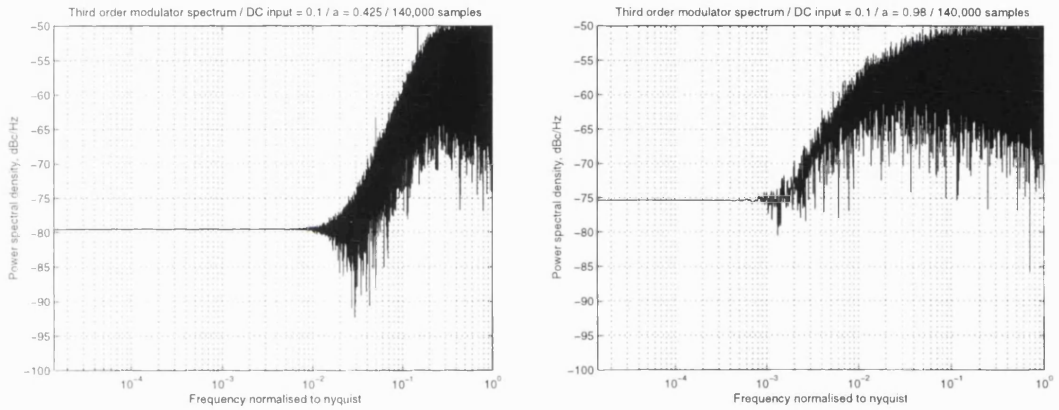


Figure 6.26: Modulator output spectrum with varying loop filter zeros position for DC input of 0.1.

this approach works the shaping at low frequencies is not good and the dynamic range advantages of the modulator shaping are to a great extent lost. Section 6.6.3 discusses the problems of the sequence truncation in more detail whilst 6.7 discusses a method of overcoming this limitation.

6.6.2 Dithering Effects

A number of unwanted artifacts can appear in the output spectrum of sigma delta modulators which can act to reduce the achievable spurious free dynamic range (SFDR) of the conversion. Repetitive tones which appear at the modulator output under zero input conditions are known as *Idle Patterns*. Idle patterns occur as a result of the sampled nature of the modulator and the nonlinear transfer function of the quantiser, which result in periodic oscillations in the modulator loop known as limit cycles. Idle patterns are of particular in-

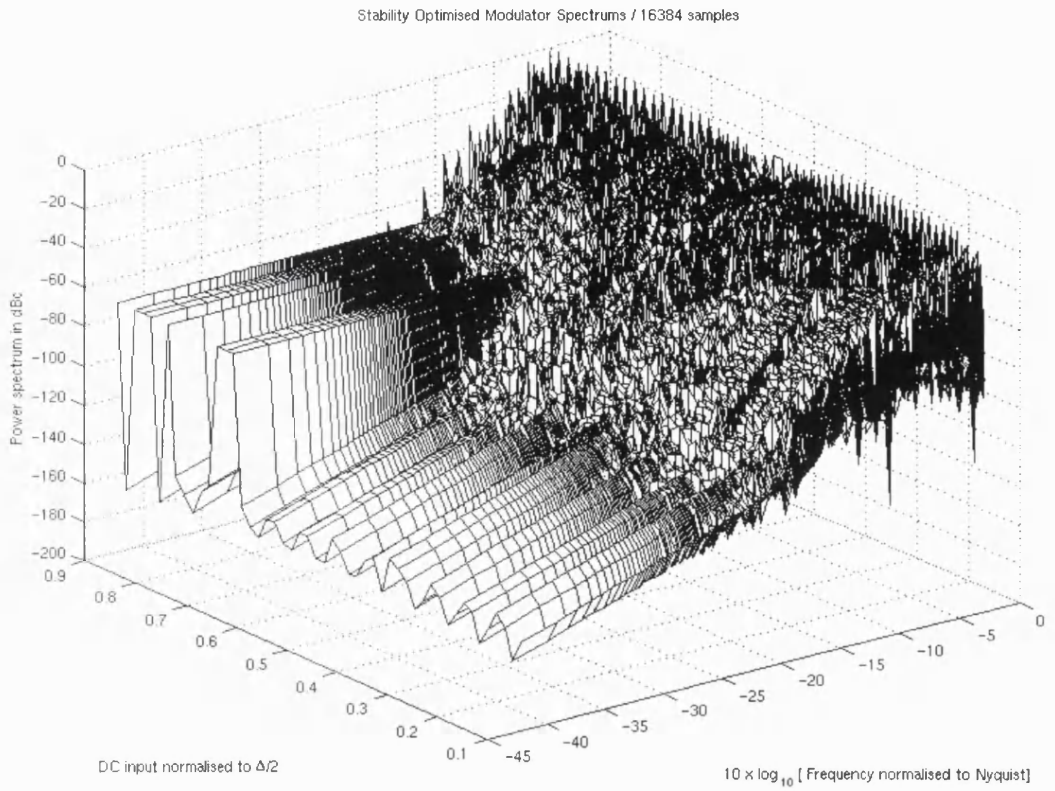


Figure 6.27: Automatically Optimised Modulator Spectra.

terest in Fractional-N applications as these are concerned with modulator operation under DC input conditions, where the spurious due to Idle patterns is of particular concern.

If required, it is possible to reduce the spurious levels at the modulator output by reducing the correlation between the noise added by the with single tone AC or DC inputs due to the action of the feedback loop. This section illustrates the potential advantages of dithering using Hanning windowed modulator spectra to temporarily remove the effects of the finite sequence length on the noise shaping.

The tonal components in a sigma delta modulator output spectrum can be reduced or completely eliminated by applying an appropriate dither signal to the modulator loop [21], at the expense of a reduction in SNR. In order to disturb a limit cycle a random signal can be injected at the quantiser input, which has the effect of randomising the quantisation noise. If the average value of the noise is zero, and if the modulator input signal is varying at a slow rate compared with the noise, then there will be a faithful reproduction of the input signal after the quantiser, although with some added noise. Spurious tones in the output may also be reduced by adding random noise at the signal input, but as this is not then

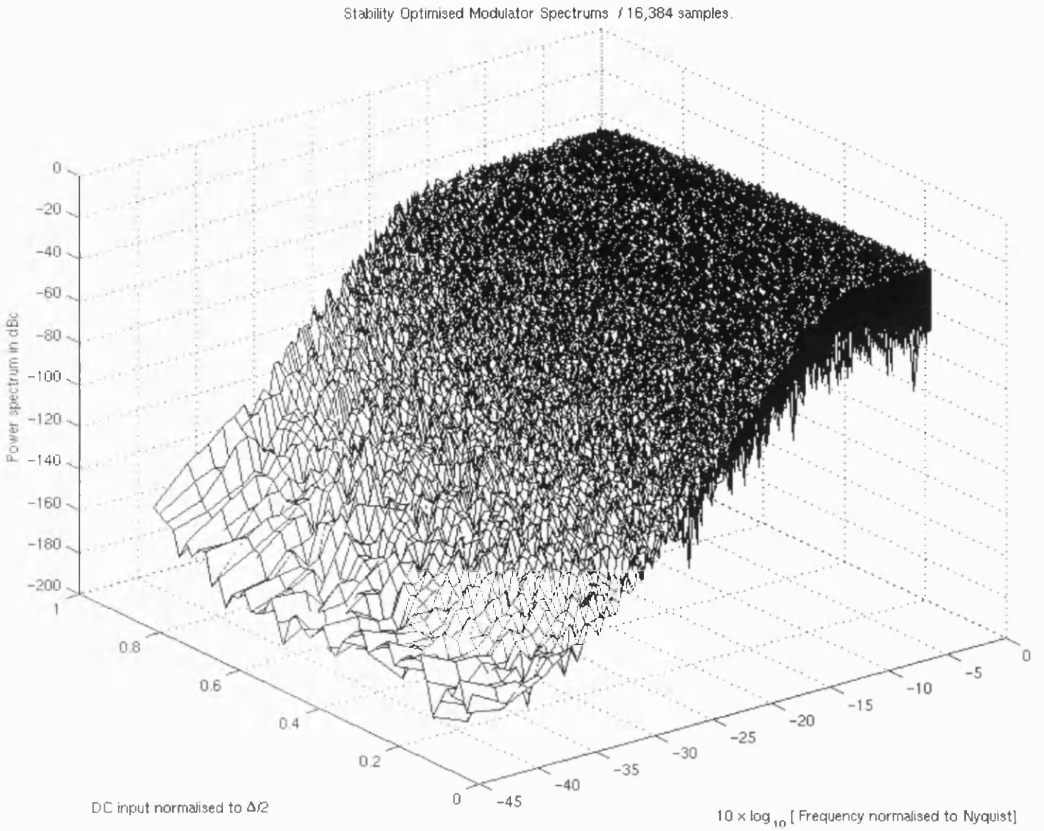


Figure 6.28: Windowed Dithered Modulator Spectra view No.1.

subject to the noise shaping of the modulator loop the effect on SNR will be increased. Dither added at the quantiser input will be subject to the same shaping as the quantiser noise.

The exact amount of noise to add as dither depends which point of the noise-spurious trade-off best suits the application, but a commonly used value is that the peak value of dither added should be no more than half of the quantisation step size [22]. All dithered results generated so far use a uniformly distributed dither signal with a peak value of half the quantiser step size and zero mean value, injected before the quantiser.

Figures 6.28 and 6.29 show an ensemble of modulator spectra with DC inputs which have been generated with a modulator using a uniformly distributed dither signal of amplitude 0.9Δ, where Δ is the step size of the single bit quantiser. Comparing figure 6.29 with figure 6.31 shows that by the use of dithering a reduction in spur levels of around 20dB has been achieved, at the expense of an increase in the distributed noise. Correspondingly the level of the components near to DC has increased from a level of -170 to -150 dBc. If a

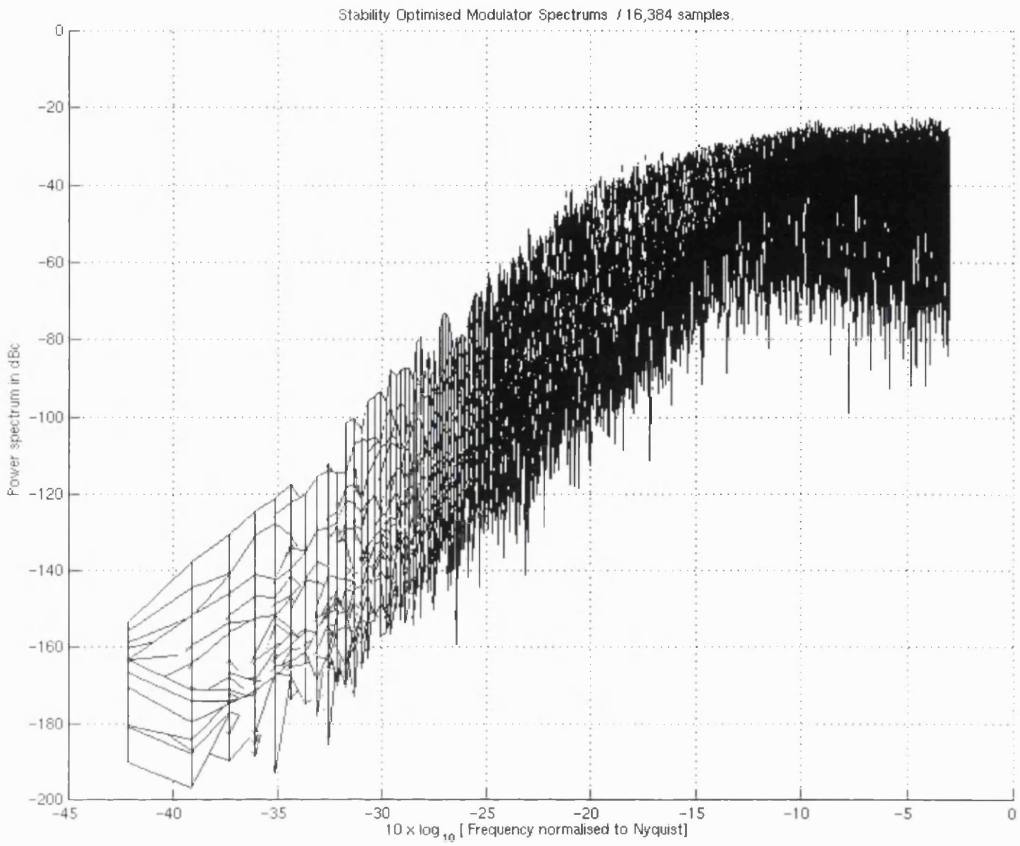


Figure 6.29: Windowed Dithered Modulator Spectra view No.2.

modulator spectrum can be synthesised which avoids the whitening of the noise spectrum close to DC due to truncation effects, then it is likely that the use of dither can offer a very significant performance improvement.

6.6.3 Truncation Effects

Figures 6.30 and 6.31 show the effects of applying a Hanning windowing function to the modulator bitstreams used to generate figure 6.27. This shows the true shaping of the modulator output by giving greater emphasis to the central portion of the sequence and removing the spectral leakage due to the rectangular windowing. It is immediately evident that the distortion in figure 6.27 occurs as a result of the limited sequence length.

One way to reduce the effects of the noise whitening at low frequencies is to use increased sequence lengths. Figure 6.32 shows the effects of lengthening the sequence used from the modulator for DC inputs from 0.7 to 0.9. Comparing this with figure 6.32 shows that

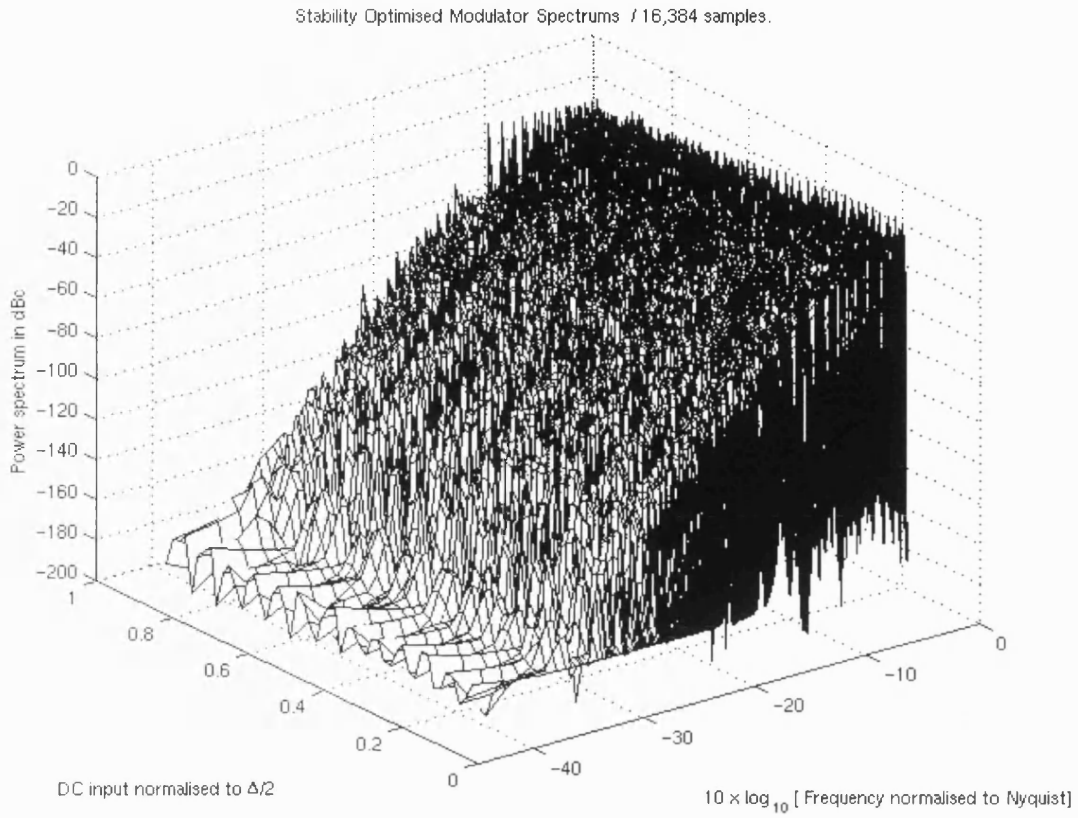


Figure 6.30: Windowed Modulator Spectra view No.1.

a large reduction in noise can be achieved at the expense of greater memory requirements.

If there is to be a minimum of spectral leakage and degradation of the noise shaping then the range of frequencies to be synthesised must consider the length of the modulator sequence and the ability to synthesise such a frequency using the single bit sequence of limited length. The length of sequence required will be longer if the resolution required of the modulator is larger. It follows that if the length of the sequence is to be a power of two then the achievable resolution will be the modulator output resolution divided by some power of two. This is a requirement if the truncated sequence is to be looped in a continuous manner without large discontinuities. A larger discontinuity in the looped sequence is likely to result if the input signal to the modulator does not fit in the achievable resolution which can be achieved with the limited length sequence.

Comparing figure 6.33 to figure 6.27 demonstrates that restriction of the modulator input resolution to the reciprocal of some power of two allows sequences to be generated which have improved low frequency characteristics over a wider DC input range with square

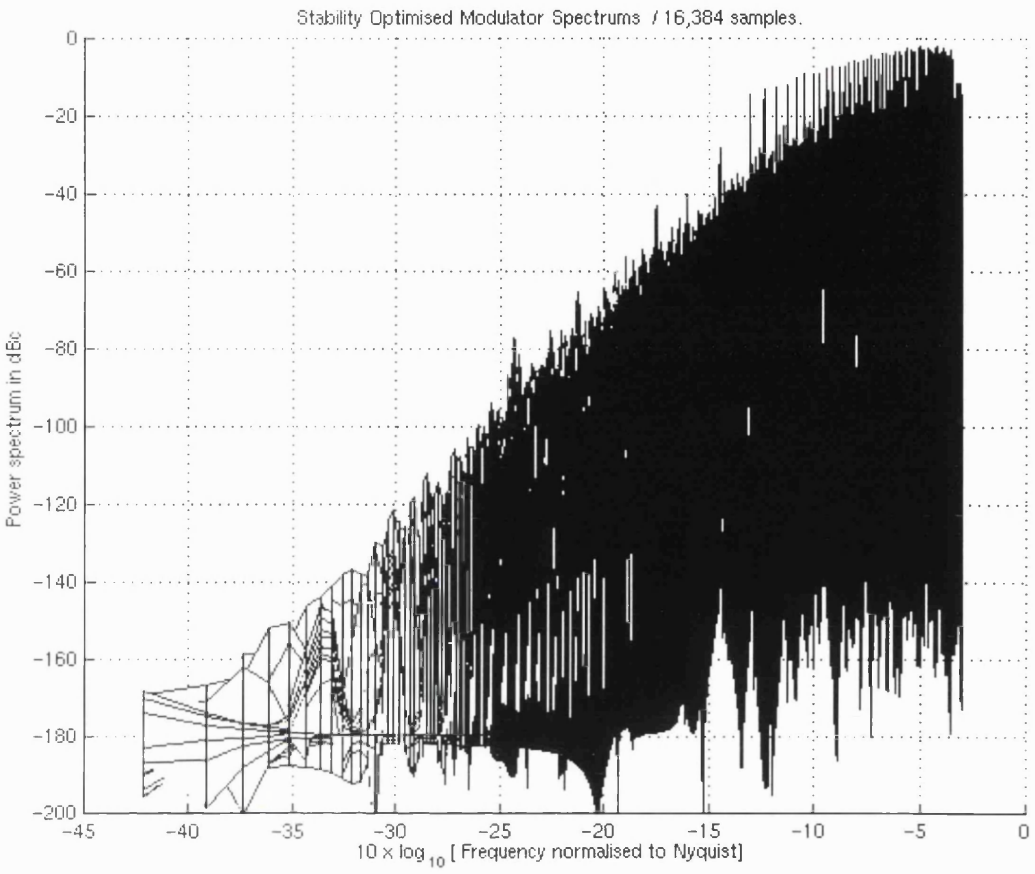


Figure 6.31: Windowed Modulator Spectra view No.2.

windowing sequences. These spectra were generated with modulator DC inputs in steps of 1/64th of the quantiser step size.

6.7 Optimal Sequence Length

In order to ascertain the correct length of sequence in order to minimise spectral distortion when using a truncated shaping sequence and square windowing, the shaping sequences used to create figure 6.33 were cross-correlated with a subset of themselves and the length of the repeating section measured. The length of the subset was unimportant as the only criterion was the detection of the repetition length. Figure 6.34 shows an example output of this process where the distance between the prominent peaks was recorded. The results for a fractionality of 1/32 of the quantiser step are tabulated in table of figure 6.7, where the DC input is specified as a proportion of $\Delta/2$ where Δ is the single bit quantiser step size. These modulator inputs therefore correspond to covering a divider range of N+1/2 to N+1/2+29/32, which corresponds to 90% of the range of the dual modulus divider. The

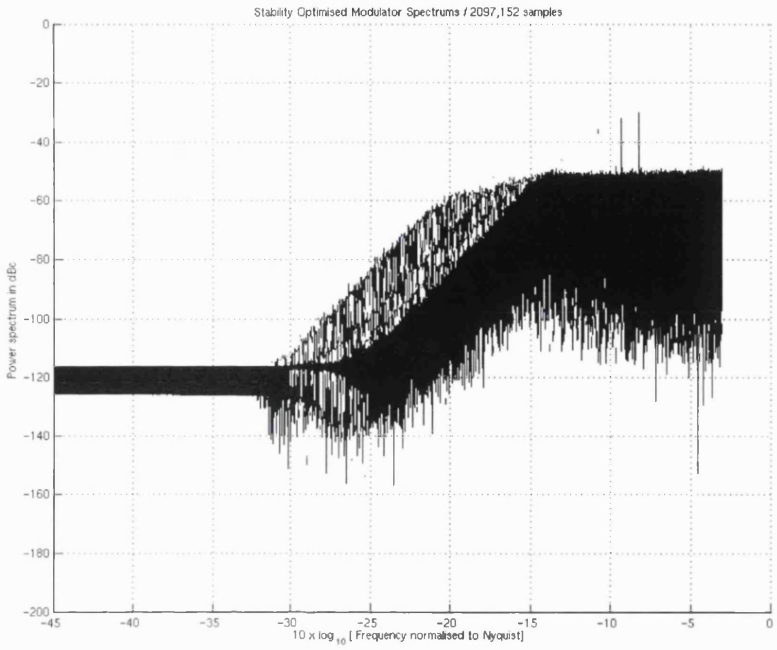


Figure 6.32: Spectra due to lengthened sequences for DC inputs from 0.7 to 0.9

other half of the range is entirely symmetrical.

Referring to figure 6.33 and the table of repetition lengths, it can be seen that one of the frequencies where there is non-optimal noise shaping, corresponding to the input of 15/32 of the full scale input, also has a non power of two repetition length of 384. This means there will be some sequence overlap if a 16.384 point FFT is performed as used to generate figure 6.33. The lowest common multiple of 384 and 16384 is 49,152. If all the sequence lengths are changed to this length then the spectra of 6.35 and 6.36 result instead, with no evidence of spectral distortion of the component corresponding to $(15/32)(\Delta/2)$. The side view of these spectra, figure 6.36 now compares very favourably with the windowed spectra shown in figure 6.31. A significant point is that some spectral distortion remains at two remaining frequencies., which is due to their sequence lengths not fitting an integer number of times into 49.152. Referring to table 6.7, the sequence length for the input of 25/32 corresponds to one of these two remaining distorted sequences, with a sequence length of 896. The fact that 896 does not divide into 49,152 is as expected, therefore there is more scope for optimising these sequences. This demonstrates the importance of selecting a sequence length which is an integer multiple of the modulator sequence repetition length if the stored sequence is to be looped for non-frequency hopping applications. The sequence length chosen should be the lowest common multiple of all of the modulator output sequences to be used.

DC input	Repetition Length	DC input	Repetition Length
$\frac{0}{32}$	8	$\frac{16}{32}$	32
$\frac{1}{32}$	256	$\frac{17}{32}$	128
$\frac{2}{32}$	64	$\frac{18}{32}$	192
$\frac{3}{32}$	128	$\frac{19}{32}$	128
$\frac{4}{32}$	32	$\frac{20}{32}$	32
$\frac{5}{32}$	128	$\frac{21}{32}$	128
$\frac{6}{32}$	64	$\frac{22}{32}$	64
$\frac{7}{32}$	128	$\frac{23}{32}$	512
$\frac{8}{32}$	32	$\frac{24}{32}$	128
$\frac{9}{32}$	256	$\frac{25}{32}$	896
$\frac{10}{32}$	64	$\frac{26}{32}$	256
$\frac{11}{32}$	128	$\frac{27}{32}$	512
$\frac{12}{32}$	96	$\frac{28}{32}$	512
$\frac{13}{32}$	128	$\frac{29}{32}$	128
$\frac{14}{32}$	64		
$\frac{15}{32}$	384		

Table 6.1: Measured sequence repetition lengths for 1/32 fractionality.

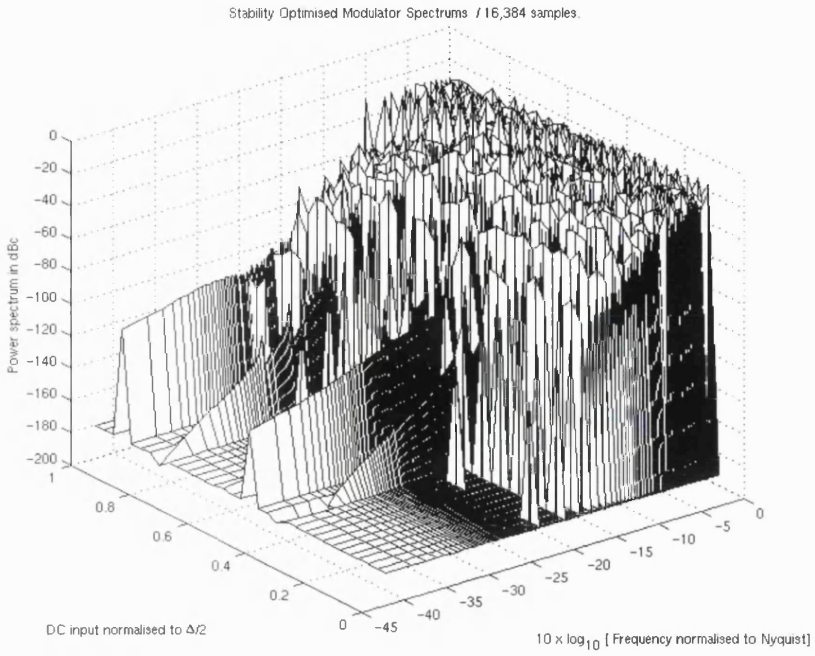


Figure 6.33: Spectra at 1/64 resolution.

6.8 Summary

It is clear that the main restriction on the quality of noise shaping that is achieved is due to the truncation of the sequence length. It is also clear that most of these problems can be overcome, by electing to use a subset of the full divider range, by increasing the reference frequency or by using longer noise shaping sequences to minimise the discontinuity problems inherent in a looped usage of the modulator sequence.

It should be noted that in frequency hopping applications where the synthesiser will be used in a burst of operation on one of many channels in sequence, then the system performance should be no different to the results achievable with a conventional hardware modulator as long as the shaping sequence is at least as long as the required channel burst. In this case the windowed results are applicable since the modulator sequence can be made longer than the required time slot, and any spectral leakage effects will be those which exist anyway elsewhere in the system due to the ramping up and down of signal power to change channel.

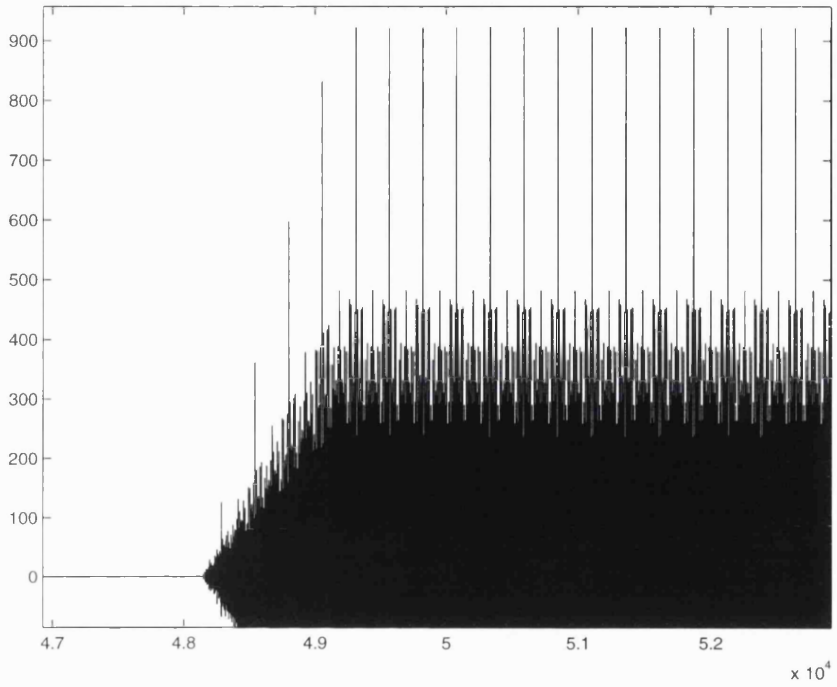


Figure 6.34: Cross Correlation of shaping sequence with a subset of itself.

The modulator performance was optimised for the maximum amount of noise shaping for each DC input signal consistent with stable modulator running. Due to the dependence of the quantiser describing function on both the instantaneous phase uncertainty and the level of offset due to the DC input, large signal stability was determined by simulation using an adaptive iterative process to methodically find the maximum input. This allowed extraction of a curve showing the absolute maximum input against the filter parameter controlling the loop filter zeros. A stability margin was then added to this to improve the consistency of results across a range of DC inputs.

The use of dither was investigated with shaping sequences with a Hanning window applied and found to allow a large reduction in spurious height at the cost of marginally increased low frequency noise. However, in the case of truncated shaping sequences for looped usage dither was considered to unduly degrade the low frequency characteristics of the modulator, causing whitening of the spectrum near to DC. For this reason dithering could be recommended for applications with continuously running hardware modulators but not for truncated sequence applications.

To conclude, it was found that use of resolutions of $1/(2^n)$ relative to the quantiser step could benefit stored sequence applications where the sequence is to be cycled through

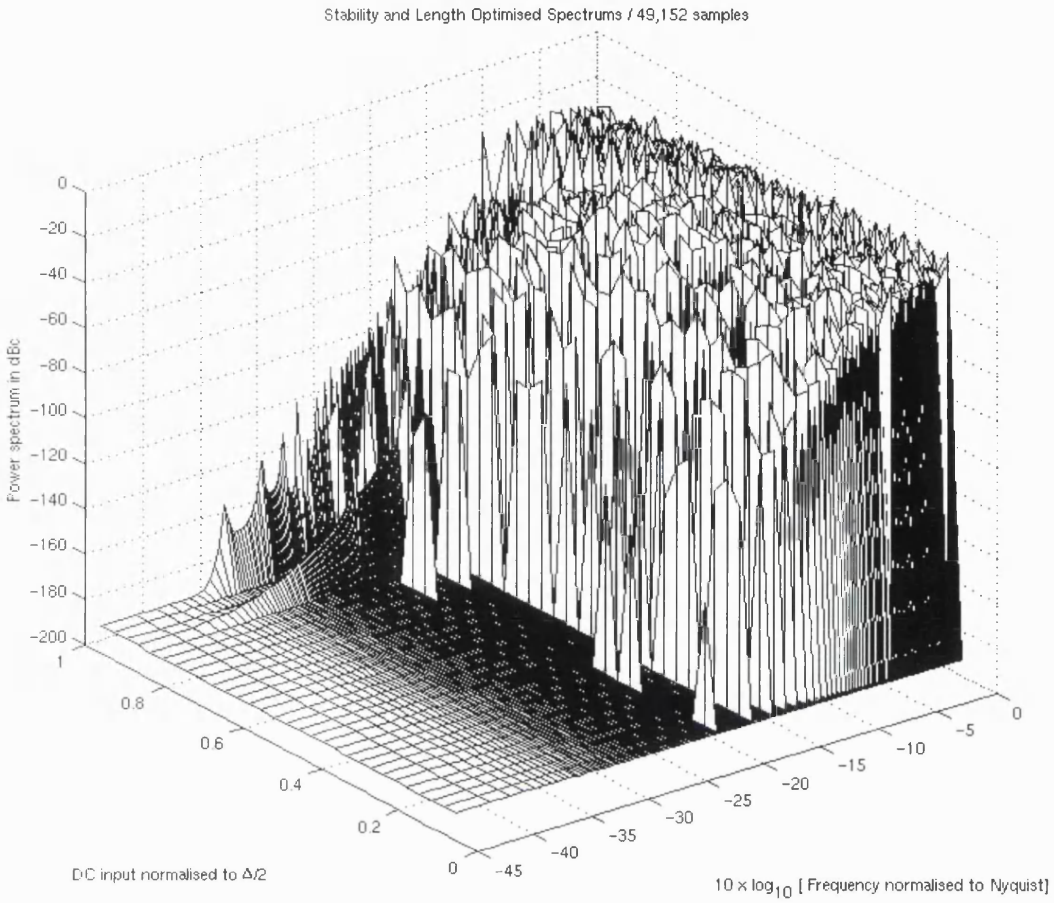


Figure 6.35: Spectra for Stability and Length Optimised Sequences.

in a continuous manner. In all cases good noise shaping performance was found easy to achieve with DC inputs up to $\pm\Delta/2$ from the zero bias value, where Δ is the single bit quantiser step size. Choice of the correct length of truncated sequence in looped applications was of paramount importance and easily measured in practice.

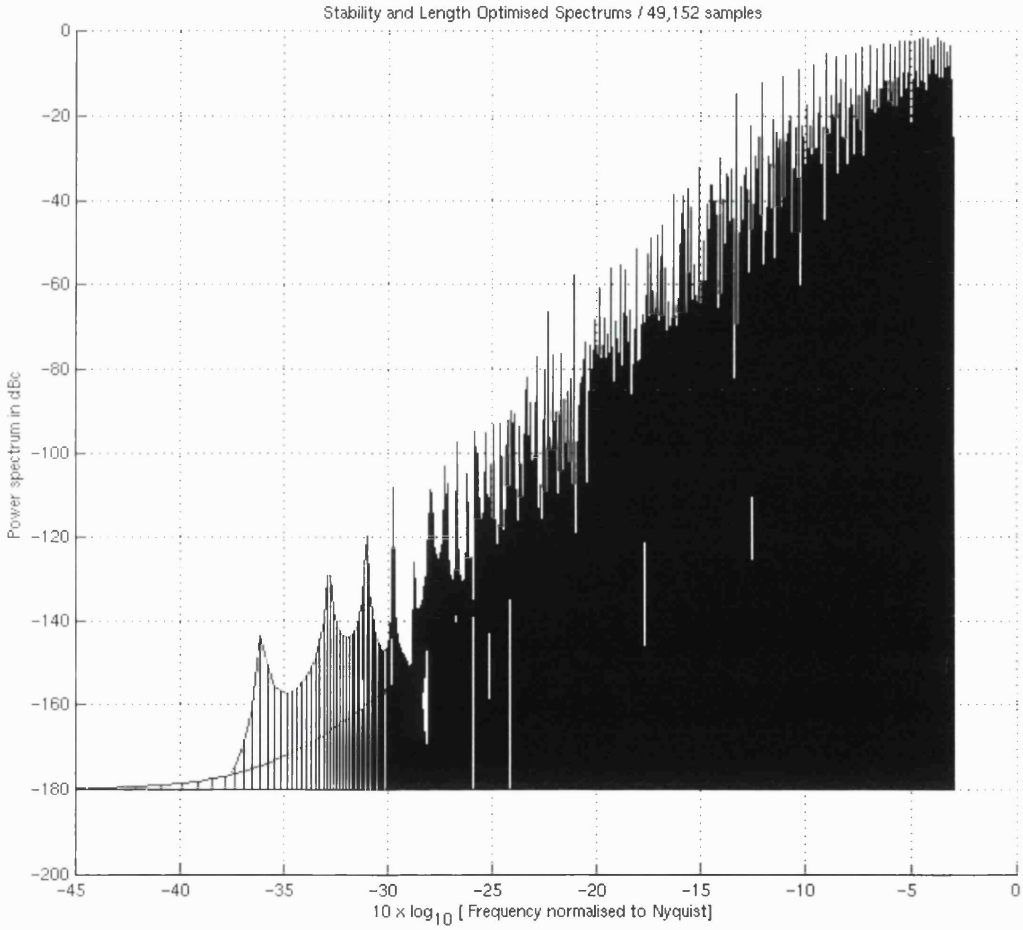


Figure 6.36: Spectra for Stability and Length Optimised Sequences, side view.

Chapter 7

Summary and Conclusions

7.1 Summary

This work has been concerned with the study of a new type of synthesiser system using stored sigma delta sequences. By off-line generation of such sequences and download of a limited length sequence into memory the large hardware overhead of a hardware modulator can be avoided. In addition to this benefit a number of advantages are gained, one of which is a considerable increase in operating speed for a given technology due to removing the requirement for any multipliers or adders in the synthesiser logic. The increase in sigma delta reference frequency speed is thought to be around a factor of 10 by from estimates based on recent results with real time sigma delta synthesisers [23]. This increase in reference frequency speed has been shown to have major benefits in terms of the achievable performance as it allows more effective noise shaping of phase quantisation noise away from the carrier. This allows a higher dynamic range at the loop output or increased loop agility for the same dynamic range. Which of these two is the result depends on the choice of the loop filter. The main achievements of this work are proof of the theoretical concept with the use of a hardware prototype and the creation of a systematic optimisation process which can be used to create the division sequences stored in the memory of the synthesiser.

The first stage of this work concerned a study of various types of synthesiser technology, of the indirect synthesis, or phase locked loop type. The highest performance PLLs were found to combine the Fractional-N technique, where the feedback divider of a synthesis loop is toggled to create a fractional division ratio, and the sigma delta technology, where the noise due to a quantisation process is shaping by placing a feedback loop around the quantiser with appropriate filtering. In a sigma delta fractional-N loop this noise shaping

process is applied to the phase noise inherent in the phase quantisation process inherent in Fractional-N loops. All of the current commercial and research realisations of sigma delta synthesisers were found to be using the multi stage noise shaping or MASH sigma delta modulator architecture [4]. The main reason for this was assumed to be the difficulty of stabilising high order single loop sigma deltas with the large amplitude DC input signals required for fractional-N purposes. An additional disadvantage of these implementations is the multiple-bit output, which leads to the unusual requirement for a multi-modulus prescaler.

The realisation that performance of such a Fractional-N loop is heavily dependent on the speed and implementation of the noise shaping modulator leads naturally into the concept of using such sequences stored into a memory for use in a frequency hopping application, which was the original impetus for this work. Further investigation of the DSP architectures used in the implementation of such digital modulator systems used in PLLs leads to the source of the speed limitation being the digital adders and accumulators. Furthermore, with such digital datapath elements there is invariably a tradeoff between the resolution and the achievable speed. Storing such sequences in memory whilst requiring an alternative hardware overhead, totally avoids the need for adders or multipliers and allows very high speeds to be achieved for the modulator reference frequency. Benefits are also achieved in terms of a direct reduction in the synthesiser reference sidebands as these will be at a larger offset from the carrier and hence further suppressed by the loop filter.

Due to the nonlinear nature of sigma delta modulators and the asymmetry which resulted from the required DC input signals, initial simulations of the spectra of such a synthesiser used a crude sweeping optimisation technique for the sigma delta modulator to achieve both stability and good low frequency noise shaping over a range of DC inputs across the synthesiser band. Once a number of these sequences were optimised, construction of a suitable hardware testbed was commenced in order to compare experimental results with those from the simulations. The results from this hardware were tested using the same sequences as were used for the simulations and the performance measured at various locations in the circuit to show areas of potential hardware limitation. Results measured at the dual modulus divider input corresponded very closely with simulations whilst those at the loop output performed as expected when the restrictions imposed by the loop hardware were taken into account. In particular, the prototype performance was constrained by the performance of the critical divider component both in terms of toggling rate and reference frequency which would not be an issue in a dedicated implementation. Some

additional spurious at the loop output was attributed to nonlinearities in the phase locked loop which could not be accounted for in the simulations, such as interstage coupling and nonlinearity in the divider. Dynamic performance conformed closely with predictions.

7.2 Sequence Optimisation

Optimisation of the stored sigma delta sequence was found to be critical to the performance of the synthesiser and complicated in a number of aspects. One of the main difficulties is in ensuring stability of the closed loop modulator. Approximate techniques for general nonlinear stability analysis generally involve dividing the system into the linear and nonlinear parts and assigning the nonlinear part a describing function, which is the closest possible linear approximation to the nonlinear element for a certain type of signal. A common describing function type used to determine the autonomous system stability is the sinusoidal input describing function (SIDF). In this case use was made of a describing function described by Engelen [18] and a loop filter described by Stikvoort [16] to provide a starting point for the modulator loop filter design by designing for zero-input stability.

Simulations have shown that the describing function gain for the quantiser is dependent not only on the instantaneous phase of the quantiser, an uncertain but bounded parameter, but also on the magnitude of any bias introduced into the sigma delta loop. This means a different describing function would exist for each modulator DC input. This led to the use of simulation to determine the stable boundary of DC input for different classes of loop filter. At iterative adaptive technique was used to determine the maximum stable input over a simulation period comparable with the required sequence length. One point worthy of note is that simulation in no way proves the stability of the modulator, which is still an unsolved problem. The longer the modulator will run without showing evidence of unstable limit cycles, the greater the confidence of stability can be. There is no way of knowing about the existence of unstable limit cycles which may have a time period greater than the simulation length. Stability in a timescale longer than the sequence length is of no concern in this application.

Whilst stability is a prerequisite to obtain a long sequence with the desired properties the optimum achievable noise shaping is also output frequency dependent in fractional-N applications where the sigma delta is not unconditionally stable, i.e. containing a loop filter of order greater than two. The studies of approximate zero input stability showed that a

larger stability margin must be provided to cope with the influence of larger modulator input signals and the required stability margin is dependent on both the frequency and DC level of the input. From plots of the worst case root locus of zero input stability it can be easily shown that for the third order sigma delta loop filter examined there is a trade off between how large the stability margin is and how high in frequency the poles of the closed loop modulator response can be placed. Placing these poles at a higher frequency will in theory give greater shaping of the quantiser noise away from DC but is more likely to result in instability. Because of this it becomes more difficult to provide good noise shaping as the modulator DC input is increased away from the zero-bias value. This level is the level at the input corresponding to the DC level halfway between the quantiser extremes. This translates in Fractional-N PLL applications to the N+1/2 value of division with a dual modulus divider and means that frequencies get more difficult to synthesise with good noise shaping towards the divider extremes.

Although the original impetus of this work was for frequency hopping applications the use of a stored-sequence synthesiser for continuous mode applications has also been addressed. In this case the logic surrounding the memory is arranged to loop the channel sequence in a continuous manner rather than switching to another channel block. In continuous mode use, of at least as much importance as the loop filter optimisation was found to be the length of the generated sequence output selected for the looping.

If the modulator sequence is to be truncated then this introduce large discontinuities at the ends of the sequence, the effect of which will be less the longer the sequence is. To minimise this effect it was decided to look for evidence of repetition in the output sequences of the modulator, in an attempt to determine the optimal length to generate a seamless loop. This was achieved by the autocorrelation of the sequences with a subset of themselves, which revealed the inherent length of their periodicity very clearly. By recording these repetition lengths over the whole range of desired modulator inputs it was possible to find the lowest common multiple of all the required sequence lengths. Figures 6.35 and 6.36 demonstrate that by methodically choosing the sequence length in this manner it is possible to get results as good as those with a hardware modulator implementation using truncated sequences.

7.3 Hardware Platforms

In order to fully benefit from the technique of storing the $\Sigma\Delta$ bitstreams the synthesiser needs as high a reference frequency as possible. The prototype performance was limited by the switching and input frequency limits of the dual modulus divider. An ideal technology for fabricating the complete system would be a 0.8μm BiCMOS HBT process [24] which combines the density benefits of CMOS technology with high frequency Heterojunction bipolar transistors which have a cutoff frequency of 30GHz. A high speed HBT core can be implemented for the divider and phase detector, whilst lower speed digital CMOS circuitry can be used for memory interfacing and FLASH memory implementation.

Alternative technologies such as GaAs would offer comparable speed but at a higher cost, whilst SiGe BiCMOS comes at a cost around 50% higher than conventional BiCMOS. CMOS is still the cheapest implementation option, with new dynamic techniques allowing GHz speed prescalers to be implemented using fine-line CMOS processes [25, 26]. With CMOS the most pressing implementation issues become the packaging and bondwire parasitics and the design of the ESD protection in the IO pads, as both of these can severely limit the speed when driven from an external input such as a VCO. This in combination with the lower output drive capability of CMOS makes board level I/O impossible to achieve at the core operating frequency. Consequently CMOS technology appears to be best suited to applications where an entire PLL is to be integrated along with a VCO. Two approaches to this are to use a multi-chip module with a separate MMIC for the VCO, which offers the best VCO Q performance, or to integrate the VCO using spiral inductors in the CMOS technology, which although very cheap, suffers from the low Q of the integrated inductor. New microwave CMOS technologies are gradually becoming available which improve inductor Q by removing the insulator under the inductor metal.

A highly suitable ECL circuit structure for high speed prescaler flip flops has shown to be the current mode logic dynamic design [27, 28, 24, 29] which allows extremely high speed operation. Approaches to analysing this circuit are an extension of the small signal models for the emitter coupled pair. Whilst common use is made of simulation to optimise propagation delay a number of techniques have been proposed to optimise the bias current with respect to operating frequency [30, 31, 32, 33].

7.4 Suggestions for Future Work

Improvements relating to this technique can be gained mainly in the area of sequence optimisation. This in turn can be broken down into the two areas of stability analysis and sequence length optimisation. An analytical method of determining the minimum common multiple of the set of desired modulator sequences would be of great utility in finding out the optimal sequence length for a desired resolution.

Exact stability analysis for nonlinear systems is an unsolved problem. Due to the lack of superposition in the nonlinear element the stability analysis techniques used for linear systems do not apply. Whilst stability for a limited length sequence can be achieved through simulations this is no proof of absolute stability as there is no way yet to predict unstable limit cycles which occur over a longer cycle time. Such simulations are also time consuming, therefore an analytical result for conditions other than the zero-input one would be of considerable aid to the design process.

The describing function technique, whilst being an approximate process, is of considerable practical value in systems such as sigma delta modulators where the linear part of the system is low pass in nature. The greater the suppression of high frequency components in the loop, the more accurate will be the approximation due to its reliance on the quantiser output being dominated by its fundamental output.

Whilst reference has been made to the zero-input sinusoidal describing function (SIDF) in plotting the root locus of the modulator to ascertain small signal stability of the modulator, modulators with large DC inputs invalidate this describing function model. DC input to the modulator has to be factored into the stability analysis as it is equivalent to introducing a DC offset directly at the nonlinearity output. This DC offset must be used to modify the quantiser nonlinearity and create a dual input describing function, or DIDF [11], which will model the resulting asymmetric nonlinearity as two separate describing functions, one being the sinusoidal describing function (SIDF) and the other describing the bias component of the signals round the loop.

Considering a generic sigma delta modulator loop, with a linear loop filter G(z) and nonlinear quantiser Q(z), if the signal at the input to the quantiser element contains a bias component of B and a sinusoidal component A then the quantiser input now has a separate describing function for the bias signal and AC signals. Q_a is the SIDF relating the

fundamental sinusoid output to a sinusoid at the quantiser input, whilst Q_b is the describing function relating the mean output to the mean input of the quantiser. For a sinusoid to propagate unattenuated around the loop, i.e. for a limit cycle to exist, requires:

$$Q_a(A, B, z)G(z) = -1 \tag{7.1}$$

Whilst for the DC condition around the loop which needs to be satisfied is:

$$Q_b(A, B, z)G(1) = -1 \tag{7.2}$$

Both of these conditions are dependent on the DC and AC components and need to be satisfied simultaneously to find the resulting quantiser AC and DC gain when a stable limit cycle exists. The frequency dependence of both of the quantiser describing functions as a result of the frequency dependent phase uncertainty. The impact of this is that the DC bias will change the effective amount of asymmetry in the describing function. This will alter the gain of the AC describing function Q_a. One way this will happen is that whilst a symmetric linearity contains only odd output harmonics, as the describing function becomes more asymmetric more power is present in the quantiser output at the even harmonic frequencies, resulting in less accuracy in the describing function, due primarily to the presence of second harmonic output.

7.5 Conclusions

The stored sequence noise shaping synthesiser is an attractive technique which offers a number of advantages in the implementation of low cost, high performance communications systems. By the use of optimised sigma delta shaping sequences, a synthesiser can be constructed achieving an extremely high reference frequency which has inherent advantages in maximising spurious free dynamic range and loop dynamic performance. This may be achieved without recourse to expensive technologies due to avoiding the requirement for real time addition or multiplication.

An significant advantage of the stored sequence approach is that the problem of long term stability of a nonlinear sigma delta loop is completely avoided in the final hardware. In addition, the off-line generation of the sequences allows an arbitrarily high order of noise

shaping to be achieved in the final product without incurring any cost penalty in the final product. The large signal stability which is critical in Fractional-N applications using high order sigma delta loops of this type, can be optimised across the synthesiser tuning range which is not possible with conventional real time implementations without significant penalty in terms of hardware count or speed.

The theoretical and experimental work performed during the course of this work suggest that the stored sequence synthesiser has much to offer in the field of mobile communications. Future use of the approach would seem to be dependent on streamlined methods of sequence generation as an appropriate sequence length removes any artifact of the sequence truncation. An highly integrated and economically viable solution should be easily possible in recent technologies allowing an ideal indirect digital synthesis solution for system on a chip applications.

References

[1] Paul V. Brennan. *Phase-locked loops, principles and practice.* Macmillan, 1996.

[2] David Jarman. A brief introduction to sigma delta conversion. *Harris Semiconductor application note*, AN9504, May 1995.

[3] Bhagwati P. Agrawal. Design methodology for $\sigma\delta$modulator. *IEEE Transactions in Communication.*, COM-31(31):360–370, March 1983.

[4] Terrence P. Kenny and Thomas A. D. Riley. Design and realisation of a digital $\sigma\delta$ modulator for fractional-n frequency synthesis. *IEEE Trans. Vehicular Technology*, 48(2):510–521, 1999 1999.

[5] Norman M. Filiol. An agile ism band frequency synthesiser with built-in gmsk data modulation. *IEEE J. Solid State Circuits*, 33(7):998–1008, July 1998.

[6] Tom A. D. Riley. $\sigma\delta$ modulation in fractional-n synthesis. *IEEE J. Solid State Circuits*, vol 28(No. 5):553–559, May 1993.

[7] Ulrich L. Rohde. Fractional-n methods tune base-station synthesiser. *Microwaves and RF*, page 1998, April 1998.

[8] W.R.Bennett. "spectra of quantized signals". *Bell Systems Technical Journal*, 27:446–472, July 1948.

[9] Kuniharu Uchimura, Toshio Hayashi, Tadakatsu Kimura, and Atsushi Iwata. Oversampling a-to-d and d-to-a converters with multistage noise shaping modulators. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(12):1899–1905, December 1988.

[10] Kirk C.H.Chao, Shuujaat Nadeem, Wai L. Lee, and Charles G. Sodini. A higher order topology for interpolative modulators for oversampling a/d converters. *IEEE Transactions on Circuits and Systems*, 37(3):309–318, March 1990.

[11] Arthur Gelb and Wallace E Vander Velde. *Multiple Input Describing Functions and Nonlinear System Design.* McGraw-Hill, 1968.

[12] Alan V. OppenHeim and Ronald W. Schafer. *Digital Signal Processing.* Prentice-Hall, 1975.

[13] Arthur B. Williams and Fred J. Taylor. *Electronic Filter Design Handbook.* McGraw-Hill, 2nd edition, 1988.

[14] Norman S. Nise. *Control Systems Engineering.* Benjamin Cummings, 1995.

[15] Steven R. Norsworthy, Richard Schreier, and Gabor C. Temes. *Data Converters; Theory, Design and Simulation.* IEEE press, 1997.

[16] Eduard F. Stikvoort. Some remarks on the stability and performance of the noise shaper or sigma-delta modulator. *IEEE Transactions on Communications,* 36(10):1157–1162, October 1988.

[17] J.A.E.P van Engelen and R.J. van de Plassche. New stability criteria for the design of low-pass sigma delta modulators. In *Proceedings of the International Symposium on Low Power Electronics and Design,* pages 114–118, 1997.

[18] J.A.E.P. van Engelen and B.E. Sarroukh. Phase uncertainty of a sampled quantiser. *Proc. of the Workshop on Circuits, Systems and Signal Processing (CSSP) Meirlo (NL)P), M,* pages 171–176, 1997.

[19] Ferrel G. Stremler. *Introduction to Communication Systems.* Addison Wesley, third edition, 1990.

[20] Derek P. Atherton. *Stability of Nonlinear Systems.* Research Studies Press, 1981.

[21] Chang Han Bae, Joon Hyoung Ryu, and Kwang Won Lee. Suppression of harmonic spikes in switching converter output using dithered sigma-delta modulation. *Industry Applications Conference, Thirty Sixth IAS Annual Meeting,* 4:2167–2174, 2001.

[22] Jurgen van Engelen and Rudy van de Plassche. *Bandpass Sigma Delta Modulators.* Kluwer Academic Publishers, 1st edition, 1999.

[23] Detlev Theil, Christian Durdodt, and Andre Hanke et al. A fully integrated cmos frequency synthesiser for bluetooth. *Radio Frequency Integrated Circuits Symposium,* pages 103–106, 2001.

[24] Katsuyoshi Washio, Reiki Hayami, Eiji Ohue, Katsuya Oda, Masamichi Tanabe, Hiromi Shimamoto, and Masao Kondo. 67 ghz static frequency divider using 0.2um self-aligned sige hbts. *IEEE Radio Frequency Integrated Circuits Symposium*, pages 31–34, 2000.

[25] R. Rogenmoser, N.Felber, Q. Huang, and W. Fichtner. 1.16 ghz dual modulus 1.2 micron cmos prescaler. *IEEE Custom Integrated Circuits Conference*, pages 27.6.1–27.6.4, 1993.

[26] R. Rogenmoser, Q. Huang, and F. Piazza. 1.57 ghz asynchronous and 1.4 ghz dual modulus 1.2 micron cmos prescalers. *IEEE Custom Integrated Circuits Conference*, pages 16.3.1–16.3.4, 1994.

[27] Tim Seneff, Lynelle McKay, Kurt Sakamoto, and Neil Tracht. A sub-1ma 1.5ghz silicon bipolar dual modulus prescaler. *IEEE Journal of Solid State Circuits*, 29(10):1206–1211, October 1994.

[28] Moriaki Mizuno, Hirokazu Suzuki, Masami Ogawa, Kouji Sato, and Hiromichi Ichikawa. A 3-mw 1.0ghz silicon-ecl dual modulus prescaler ic. *IEEE Journal of Solid State Circuits*, 27(12):1794–1798, December 1992.

[29] Katsuyoshi Washio, Wiji Ohue, Katsuya Oda, Reiko Hayami, Masamichi Tanabe, Hiromi Shimamoto, Takashi Harada, and Masao Kondo. An 82ghz dynamic frequency divider in 5.5ps ecl sige hbts. *IEEE International Solid State Circuits Conference*, ISSCC 2000/session 12/ frequency synthesisers and dividers:210–211, February 2000.

[30] M. Alioto and G. Palumbo. Modelling and optimised deisng of current mode mux/xor and flip flop. *IEEE Transactions on Circuits and Systems-2: Analog and Digital Signal Processing*, 47(5):452–461, September 2000.

[31] Massimo Alioto and Gaetano Palumbo. Cml and ecl: Optimised design and construction. *IEEE Transactions on Circuits and Systems-1 Fundamental Theory and Applications.*, 46(11):1330–1341, November 1999.

[32] M. Alioto and G. Palumbo. Highly accurate and simple models for cml and ecl gates. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 18(9):1369–1375, September 1999.

[33] Khaled. M. Sharaf and Mohammed I. Elmastry. Analysis and optimisation of series-gated cml and cml high speed bipolar circuits. *IEEE journal on Solid state circuits*, 31(2):202–211, February 1996.

[34] John. E. Pearson. *Basic Communication Theory*. Prentic Hall, 1992.

[35] M.L.Meade and C.R. Dillon. *Signals & Systems*. Chapman & Hall, 1991.

Appendix A

Phase Modulation

Phase modulation of a carrier is acheived by altering the phase constant, ϕ_c, of the carrier in proportion to the modulating signal amplitude. Starting with an equation for the carrier voltage:

$$v_c = E_c \cos(\omega_c t + \phi_c) \tag{A.1}$$

The modulation changes ϕ_c by an amount $\delta\phi$ where

$$\delta\phi = K_p v_m \tag{A.2}$$

giving the phase modulated signal

$$v_{pm} = E_c \cos(\omega_c t + K_p v_m + \phi_c) \tag{A.3}$$

Taking the modulating signal, v_m as a sinusoid of amplitude E_m and natural frequency ω_m the modulated signal is given by

$$v_{pm} = E_c \cos(\omega_c t + K_p E_m \cos \omega_m t) \tag{A.4}$$

where the initial phase of the carrier ϕ_c has been omitted for clarity. This is commonly written using the term β_p which is the phase modulation index. This is a dimensionless quantity which gives a measure of the properties of the modulated signal.

$$v_{pm} = E_c \cos(\omega_c t + \beta_p \cos \omega_m t) \tag{A.5}$$

The peak phase deviation is given by the phase modulation index

$$\delta\phi = \beta_p = K_p E_m \tag{A.6}$$

The amount of phase change possible is constrained by the phase detector range in PLL applications. The maximum possible phase change using a phase frequency type sequential detector will be $\pm 2\pi$ radians.

Expanding A.5 results in the general equation for phase modulation:

$$v_{pm} = E_c[\cos\omega_c t \cos(\beta_p \cos\omega_m t) - \sin\omega_c t \sin(\beta_p \cos\omega_m t)] \tag{A.7}$$

A full expansion of the generalised PM signal requires the use of Bessel functions of which in this case can be written:

$$\cos(\beta_p \cos\omega_m t) = J_0(\beta_p) - 2J_2(\beta_p)\cos 2\omega_m t + 2J_4(\beta_p) - ... \tag{A.8}$$

$$\sin(\beta_p \cos\omega_m t) = 2J1(\beta_p)\cos\omega_m t - 2J_3(\beta_p \cos 3\omega_m t + ... \tag{A.9}$$

Allowing the full expression for WBFM to be written:

$$v_{WBFM} = E_c \sum_{n=-\infty}^{n=\infty} J_n(\beta)\cos(\omega_c + n\omega_m)t \tag{A.10}$$

The bessel functions can be evaluated numerically in terms of the parameters n and β to evaluate the relative size of all the spectral components in the modulated signal. Figure A.1 shows a plot of the Bessel function of the first kind for various orders and varying phase modulation constant (β). This allows the amplitude of the various output components of the modulated signal to be determined.

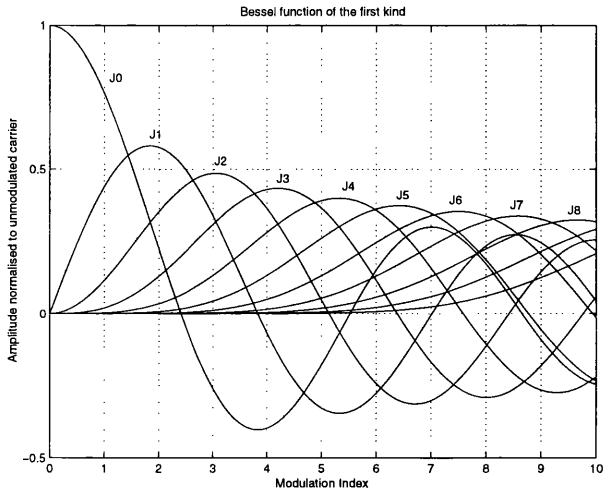


Figure A.1: Plot of Bessel function of first kind for varying orders and β

A.0.1 Narrow Band Phase Modulation

For small values of phase modulation constant (usually taken to mean $\beta < 0.2$) the following approximations of the general phase modulation expression (A.7) can be assumed:

$$\cos(\beta \sin \omega_m t) \approx 1 \tag{A.11}$$

$$\sin(\beta \sin \omega_m t) \approx \beta \sin \omega_m t \tag{A.12}$$

Allowing the expression for A.7 to be considerably simplified:

$$v_{NBPM} = E_c \cos \omega_c t - \beta_p E_c \sin \omega_c t \cos \omega_m t \tag{A.13}$$

$$v_{NBPM} = E_c \cos \omega_c t - \frac{1}{2} E_c \left[\sin(\omega_c - \omega_m)t + \sin(\omega_c + \omega_m)t\right] \tag{A.14}$$

This results in the simplified stationary phasor representation of figure A.2. Further indication of the accuracy of the approximation can be formed by plotting the bessel function for $\beta = 0.2$, the maximum value of phase modulation constant commonly taken to mean the approximation is valid (Figure A.3). The bessel function order above 2 has a level of 0.005 that of the unmodulated carrier level. Adoption of the NBPM approximation allows it to be treated as a linear modulation, where the baseband modulation appears on the carrier scaled by the factor $\beta_p E_c/2$. In practice this reduces the modulation process to a simple shift of carrier frequency and subtraction of 6dB from the power spectral density of the modulating signal. More detailed treatment of NBPM may be found in [19, 34].

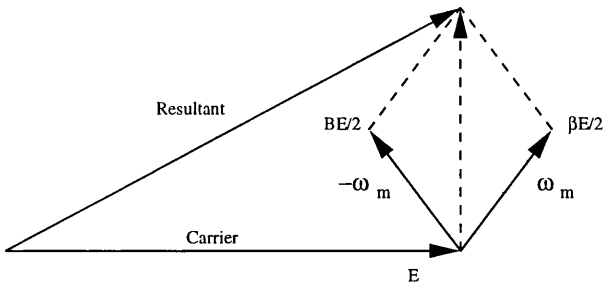


Figure A.2: Stationary phasors for NBPM

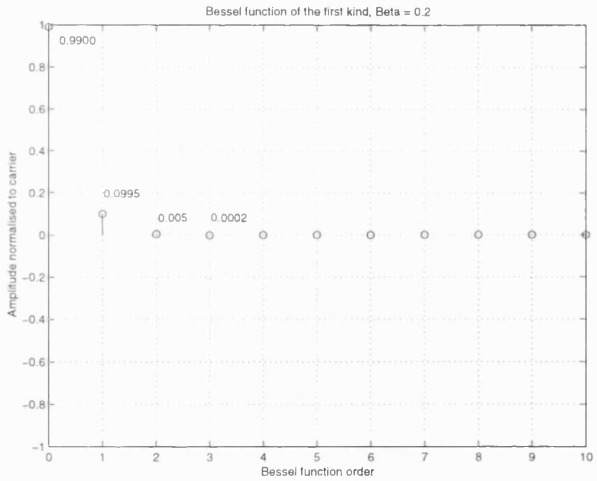


Figure A.3: Sine phase modulated carrier components with $\beta = 0.2$

Appendix B

Fourier Analysis of a Sawtooth

Sawtooth waveforms arise in many situations in the design of Fractional-N phase locked loops of any type. The following fourier analysis reveals the spectral content of the zero-mean sawtooth waveform shown in figure B.1.

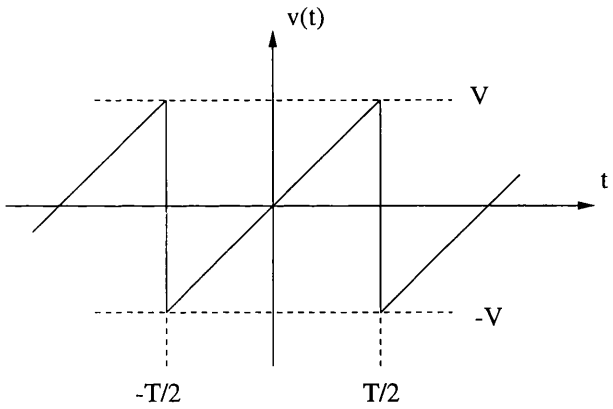


Figure B.1: Zero mean sawtooth waveform.

The general form of the fourier series describes a periodic waveform in terms of a summation of sines, cosines and a DC term over some exact multiple of its period:

$$v(t) = a_o + \sum_{n=1}^{\infty} a_o \cos(n\omega_0 t) + \sum_{n=1}^{\infty} \sin(n\omega_0 t) \tag{B.1}$$

Where a_n and b_n are the fourier coefficients of each constituent component. These coefficients are obtained from the following integrals:

$$a_0 = \frac{1}{T}\int_0^T v(t)dt \tag{B.2}$$

$$a_n = \frac{2}{T}\int_0^T v(t)\cos(n\omega_0 t)dt \tag{B.3}$$

$$b_n = \frac{2}{T}\int_0^T v(t)\sin(n\omega_0 t)dt \tag{B.4}$$

Wher T is the period of the waveform to be analysed and ω_0 the natural frequency corresponding to the same period. From inspection of figure B.1 it is clear that the component a_0 is zero as this is the mean DC level of the signal. It is also clear that all components of a_n will also be zero as the signal has an odd symmetry meaning equation B.3 will integrate to zero.

The amplitude of the cosine components is given by equation B.4. The equation for $v(t)$ over one period can be simply written if the period chosen for the analysis is from -T/2 to T/2:

$$v(t) = \frac{2}{T}t \tag{B.5}$$

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2}\frac{2}{T}t\sin(n\omega_0 t)dt \tag{B.6}$$

$$\int u\frac{dv}{dt}dt = uv - \int v\frac{du}{dt}dt \tag{B.7}$$

$$u = t \longrightarrow \frac{du}{dt} = 1 \tag{B.8}$$

$$\frac{dv}{dt} = sin(n\omega_0 t) \longrightarrow v = \frac{-\cos(n\omega_0 t)}{n\omega_0} \tag{B.9}$$

$$b_n = \frac{4}{T^2}\left[\frac{-t\cos(n\omega_0 t)}{n\omega_0}\right]_{-T/2}^{T/2} + \int_{-T/2}^{T/2}\frac{\cos(n\omega_0 t)}{n\omega_0}dt \tag{B.10}$$

$$b_n = \frac{4}{n\omega_0 T^2}\left[-t\cos(n\omega_0 t) + \frac{sin(n\omega_0 t)}{n\omega_0 t}\right]_{-T/2}^{T/2} \tag{B.11}$$

$$b_n = \frac{4}{n\frac{2\pi}{T}T^2}\left[-t\cos(n\frac{2\pi}{T}t) + \frac{sin(n\frac{2\pi}{T}t)}{n\frac{2\pi}{T}t}\right]_{-T/2}^{T/2} \tag{B.12}$$

$$b_n = \frac{1}{n\pi}\left[-\cos(n\pi) + \frac{sin(n\pi)}{2n\pi} - \cos(-n\pi) - \frac{\sin(-n\pi)}{n\pi}\right]_{-T/2}^{T/2} \tag{B.13}$$

$$b_n = \frac{1}{n\pi}\left[\frac{2\sin(n\pi)}{n\pi} - 2\cos(n\pi)\right] \tag{B.14}$$

$$b_n = \frac{-2\cos(n\pi)}{n\pi} \tag{B.15}$$

$$b_n = \frac{2}{n\pi}(-1)^{n+1} \tag{B.16}$$

$$b_1 = \frac{2}{\pi}, b_2 = -\frac{1}{\pi}, b_3 = \frac{2}{3\pi}, b_4 = \frac{-1}{2\pi} \tag{B.17}$$

Figure B.2 shows the first four components of the resulting single sided spectrum of the sawtooth. The amplitudes are scaled by V for the example sawtooth in figure B.1. Further coverage is provided in [19, 34, 35].

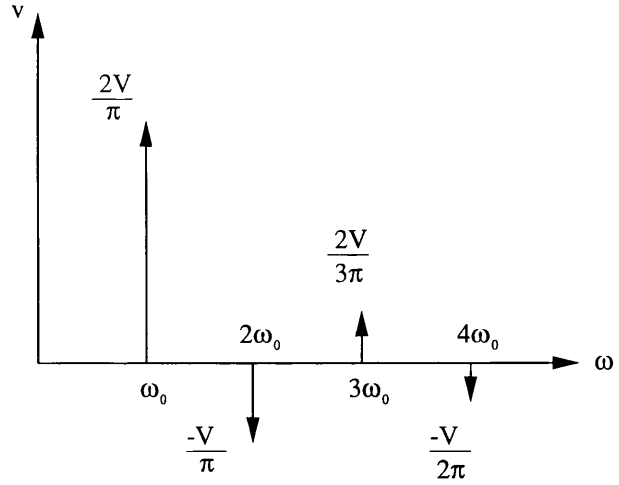


Figure B.2: First four components in single sided sawtooth spectrum.

Appendix C

Intermodulation Distortion

All active components in the loop carrying analogue signals will suffer from some degree of non-linearity which will cause intermodulation distortion to occur where more than one signal passes through a device. This causes a large number of output components which are harmonically related to those at the input, but at different frequencies. Intermodulation product generation for a small signal non-linear system is described by the Taylor series:

$$V_o(t) = a_1 V_i(t) + a_2 V_i^2(t) + a_3 V_i^3(t) + a_4 V_i^4(t) + ... \quad \text{(C.1)}$$

Where a_1 is the linear component of the system, a_2 is the square law component and so on. $V_i(t)$ is the sum of all signals at the input of the system. The amplitudes of the intermodulation components are related to the amplitude of the tones at the input of the system and the intercept point, the hypothetical output level where the level of the intermodulation products equals the level of the tones for a particular order of intermodulation. For two tones at the system input the level of an arbitrary intermodulation product is given by:

$$IM_{mn} = mE_1 + nE_2 - (m + n - 1)IP_(m + n) \quad \text{(C.2)}$$

Written in dB form. This gives the following results relating the level of intermod products to the level of input tones:

2nd order:

$$IM_{11} = E_1 + E_2 - IP_2 \quad \text{(C.3)}$$

$$IM_2 = 2E - IP_2 \quad \text{(C.4)}$$

3rd order:

$$IM_{12} = E_1 + 2E_2 - 2IP_3 \tag{C.5}$$

$$IM_3 = 3E - 2IP_3 \tag{C.6}$$

4th order:

$$IM_{13} = E_1 + 3E_2 - 3IP_4 \tag{C.7}$$

$$IM_{22} = 2E_1 + 2E_2 - 3IP_4 \tag{C.8}$$

$$IM_4 = 4E - 3IP_4 \tag{C.9}$$

5th order:

$$IM_{14} = E_1 + 4E_4 - 5IP_5 \tag{C.10}$$

$$IM_{23} = 2E_1 + 3E_2 - 5IP_5 \tag{C.11}$$

$$IM_5 = 5E_1 - 5IP_5 \tag{C.12}$$

In practice the intercept point level of output cannot be reached due to saturation. In real systems third order intercept commonly occurs with an output level at least 10dB higher than the 1dB linear gain compression signal level.

Appendix D

Matlab Routines

This appendix shows the code of a number of matlab simulations which were used in all aspects of the design, analysis and optimisation of the stored sequence synthesiser and preliminary investigations.

D.1 system.m

Matlab routine t4.m, which simulates a third order sigma delta modulator running with a fixed value input (pq) between 0 and 1, for number of sample periods nn. aa and bb are the two adjustable feedback weights which offer control over the corner frequencies of the noise shaping characteristic.

In this simulation the value of Fref is placed exactly centrally between the two end of band frequencies fmin and fmax. The for loop runs the modulator and stores the single bit output into an array Vc(n).

After storing the modulator output the bits have the DC offset removed and they are converted from amplitude to phase by taking the cumulative sum and multiplying by $2pi$, then the mean phase value is subtracted from all the points in the array to avoid a peak at DC when the FFT is taken.

After taking the FFT the complex magnitude is calculated and the array converted to power spectral density relative to the carrier. The array is converted from dBc to dBc/Hz by dividing by the FFT resolution (fref/l), giving the array mag.

Following dBc/Hz scaling a response is calculated for a third order type two loop filter and a new array generated (mag2) which gives the response after filtering by the third order loop action. The remaining parts of the routine find the highest spur in the array mag2 and measure the noise power in dBc/Hz close to the carrier. The filtered sigma delta bitstream is plotted with a log x axis and masks are overlaid for comparison with the DCS specifications for spurs and phase noise.

```
% simulate sigma delta modulator and plot contribution to PLL sidebands

clear
close all

% define p/q, fref and no. of samples
nn=1024*16
fmin=1805e6;
fmax=1880e6;
fref = (fmin+fmax)/17
fout=1879e6; %desired fout in MHz
desiredpq=fout/fref-8
actualpq=round(desiredpq*nn)/nn %nearest available p/q value
pq=actualpq;
%pq=desiredpq;
k=1500

% define prescaler modulation waveform

Vc=zeros(1,nn+k);
% initiate an array

b=0;
d=0;
f=0;
x=0;

% third order sigma-delta feedback coefficients
aa=8.765;
bb=11.002;

for n=2:nn+k;

    b=b+pq-x;
    d=d+b-aa*x;
f=f+d-bb*x;
    if f > 0.5
        Vc(n)=1;
x=1;
    else
        Vc(n)=0;
x=0;
    end
end

Vcy=Vc;
save data Vcy;
Vcx(1:nn)=Vc(1+k:nn+k);
Vc=Vcx;

% integrate frequency steps to get phase steps (with zero mean value)
l=length(Vc);
meanVc= mean(Vc)
deltaphi=2*pi*cumsum(Vc-meanVc);
meandeltaphi=mean(deltaphi);
deltaphi=deltaphi-meandeltaphi;

excesslbits=nn*(meanVc-actualpq)

% take spectrum of phase modulation due to the prescaler modulation waveform
```

```
fn=(2/l)*fft(deltaphi);

% -6dB narrow-band PM conversion - giving output spectrum in absence of any loop filter
fn=fn/2;
mag=10*log10((fn.*conj(fn)+10e-11));

% convert to dBc/Hz by dividing by FFT resolution (fref/l)
mag=mag-10*log10(fref/l);
f=[0:1/l:1-1/l];

% third order loop transfer function
phi=pi/3;
% normalised loop natural frequency
fn=250e3;
fn=fn/fref;
T=tan(phi)+sec(phi);
ff=f/fn;
response=(l+T*T*ff.*ff)./((l-T*ff.*ff).*(l-T*ff.*ff)+(T*ff-ff.*ff.*ff).*(T*ff-ff.*ff.*ff));
response=response.^0.5;
response=20*log10(response);

% apply third order loop response
mag2=mag+response;

[s,t]=max(mag2);

maxspurdbc = s + 10*log10(fref/l)
atfreq = t*fref/nn

% to return the close-in phase noise power in dBc/Hz
closenoise=mag2(100)

semilogx(f, response, f, mag2,f,mag)

% DCS phase noise mask
xmask=[1e5/fref 2e5/fref 2e5/fref 2.5e5/fref 2.5e5/fref 4e5/fref 4e5/fref 6e5/fref
6e5/fref 12e5/fref 12e5/fref 18e5/fref 18e5/fref 6e6/fref 6e6/fref 0.5];

ymask=[-44 -44 -75 -75 -78 -78 -105 -105 -115 -115 -118 -118 -125 -125 -130 -130];
line(xmask, ymask)

% DCS spurs mask
xmask2=[1e5/fref 2e5/fref 2e5/fref 2.5e5/fref 2.5e5/fref 4e5/fref 4e5/fref 6e5/fref
6e5/fref 12e5/fref 12e5/fref 18e5/fref 18e5/fref 6e6/fref 6e6/fref 0.5];

ymask2=[-32 -32 -62 -62 -65 -65 -92 -92 -102 -102 -105 -105 -107 -107 -112 -112];
line(xmask2, ymask2)

axis([0.001,0.5,-160,10])

xlabel('Frequency/fref')
ylabel('Response / dBc')
```

D.2 PLLPSD.m

```
% bitstream plotting routine to match measured results.
% loop output.
clear
close all

fname='111'; % enter the filename here xxx.mat
c_ref=-15; % shift carrier reference level to match measured.
PM=30;

load(fname);
nn=length(Vc);
fref = 156*1e6; % synth reference frequency.
fout=(9-mean(Vc))*fref % derive fout from mean(Vc)and average of bits.
```

```
desiredpq=mean(Vc);
actualpq=round(desiredpq*nn)/nn; %nearest available p/q value
pq=actualpq;
N=9-mean(Vc) % synth divide ratio

freq=fout/1e9; % to provide frequency in GHz.
str=num2str(freq);

% integrate frequency steps to get phase steps (with zero mean value)
l=length(Vc);
meanVc= mean(Vc) % Vc corresponds to the frequency steps
deltaphi=2*pi*cumsum(Vc-meanVc); % the integration step

meandeltaphi=mean(deltaphi);
deltaphi=deltaphi-meandeltaphi; % remove DC component.
excesslbits=nn*(meanVc-actualpq)

% take spectrum of phase modulation
spec=fft(deltaphi)*2/nn;
pwr=abs(spec).^2;  % convert to normalised power
pwr=10*log10(pwr); % convert to dBc

% -6dB narrow-band PM conversion
pwr=pwr-6;

% comment out for dBc, or use to convert to dBc/Hz by dividing by FFT resolution (fref/l)
%pwr=pwr-10*log10(fref/l);

% frequency vector used for filter response.
f=linspace(-fref/2,fref/2,nn);
%f=linspace(0,fref,nn); % frequency input for loop filter calculation
sigma=zeros(1,nn);
s=sigma+j*2*pi*f;
phi=(PM/360)*2*pi;
% set the phase margin for the loop filter
fn=250e3; % loop natural frequency (Hz)
wn=fn*2*pi; % loop natural frequency (rad/sec)
T=tan(phi)+sec(phi);
response=N.*(wn^2.*T.*s+wn^3)./(s.^3+wn*T.*s.^2+wn^2*T.*s+wn.^3); % third order loop transfer function
res_mag=abs(response);
res_m_dB=20*log10(res_mag);
semilogx(f,res_m_dB);
figure

% move PSD DC to zero
shift_pwr(1:nn/2)=pwr(1+nn/2:nn);
shift_pwr(1+nn/2:nn)=pwr(1:nn/2);

% apply filter response to SD spectrum.
pwr_filt=shift_pwr+res_m_dB;

% shift reference level to -15.12dB to match analyser plot
pwr_filt=pwr_filt+c_ref;

% frequency vector for plot axis
faxis=linspace(-fref/1e6/2,fref/1e6/2,nn);

plot(faxis,pwr_filt,'k');
xlabel('Carrier offset /MHz')
ylabel('Power /dBc')
axis([-5,5,-110,-10]);
grid on;

newstr=strcat('Simulated spectrum, F_{REF} = 156 MHz, F_{OUT} = ',str,' GHz');
title(newstr);
xmask=[-5 5];
ymask=[c_ref c_ref];
line(xmask,ymask);
text(-4,-18,'-15 dB  Carrier Reference Level');
```

D.3 Ftrans

```
% Program to plot frequency/phase error transient of a third order type 2 loop
% note: phase margin taken as 53.2 degrees

clear
close all

deltaF=15e6 % The frequency step
N=9; % N divider value.
deltaFpfd=deltaF/N % frequency step at the Phase detector (downwards hop worst case)
o_nat=2*pi*250e3 % loop natural frequency

K1=(2*pi*deltaFpfd/o_nat); % integrate the Frequency step to get the phase step

Ferror(1)=0;

for n=2:2000
time(n)=n/1e8;

% multiply by N to refer to output
Ferror(n)=N*deltaFpfd*(1+(o_nat*time(n))-(o_nat*time(n)).^2 )*exp(-o_nat*time(n));

% multiply by N to refer to output
Perror(n)=K1*((o_nat*time(n))+(o_nat*time(n)).^2 )*exp(-o_nat*time(n));
end

Perror_deg=Perror*360/2/pi;

% phase error at VCO output in rads.
Perror_vco=Perror_deg*N;

% provide Ferror in MHz.
Ferror_mhz=Ferror/1e6;

subplot(2,2,1), plot(time,Ferror_mhz);
%axis([0 2e-5 -2e7 8e7]);
grid on;
ylabel('Frequency /MHz');
xlabel('Time in seconds');
title('Frequency at VCO');

subplot(2,2,3), plot(time,Ferror);
axis([0 2e-5 -2000 0]);
grid on;
title('Magnified scale of above');
ylabel('Frequency /Hz');
xlabel('Time in seconds');

subplot(2,2,2), plot(time,Perror_deg);
%axis([0 2e-5 0 300]);
grid on;
title('Phase error at PFD');
xlabel('Time in seconds');
```

```
ylabel('Phase /degrees');

subplot(2,2,4), plot(time,Perror_vco);
axis([0 2e-5 0 1]);
grid on;
title('Phase Error referred to VCO');
xlabel('Time in seconds');
ylabel('Phase /degrees');
```

D.4 Cloop.m

```
% Third order type 2 loop filter closed and open loop complex phase response
clear
close all

f=linspace(1000,1000e5,100000); % frequency range
omega=2*pi.*f; % natural frequency range
sigma=zeros(1,length(f));
s=sigma+j.*omega;

f_nat=250e3;
omega_nat=f_nat*2*pi; % loop natural frequency
phi=pi*60/180; % loop phase margin in radians

% ECL phase detector levels
Vo_h=4;
Vo_l=3.2;
swing=(Vo_h-Vo_l);


% phase detector gain volts per radian
Kp=swing/(2*pi);

% VCO gain in Radians/volt.
Kv=36e6*2*pi;

% loop divider
N=8.5;

% loop filter time constants
T1=(Kp*Kv/(N*(omega_nat)^2))*(tan(phi)+sec(phi));

T2=1/(omega_nat * (tan(phi)+sec(phi)) );

T3=(tan(phi)+sec(phi))/omega_nat;

% calculate loop filter gain and phase
G=(1./(s.*T1)).*(1+s.*T3).*(1./(1+s.*T2));

f_path=(G./s).*(Kp*Kv);

loop_gain=(G./s).*(Kp*Kv/N);

phase_resp= (f_path)./(1+loop_gain);

semilogx(f,20*log10(abs(phase_resp)));
grid
hold
semilogx(f,angle(phase_resp)*180/pi);
title('Closed Loop Complex Phase Response');
xlabel('Frequency /Hz');
ylabel('Amplitude /dB, Phase /degrees');

figure(2)
semilogx(f,20*log10(abs(loop_gain)));
grid
hold
semilogx(f,angle(loop_gain)*180/pi);
```

```
title('Open Loop Complex Phase Response');
xlabel('Frequency /Hz');
ylabel('Amplitude /dB, Phase /degrees');


figure(3)
theta=angle(loop_gain);
rho=abs(loop_gain);
polar(theta(100:10000),rho(100:10000));
title('Nyquist Plot of Open Loop Complex Phase Response')


R2C1  = 2 * tan(phi)/omega_nat; % T3 (ns)
R1C1  = Kp * Kv / N / omega_nat^2 * (tan(phi) + 1/cos(phi)); % T1 (ns)
R2C2   = 1/(omega_nat *(tan(phi) +1/cos(phi))); % T2 (ns)

R2 = 1e3; % resistor in the op amps feedback path
C1 = R2C1 / R2; % capacitor in feedback path in series with R2 and C2
R1 = R1C1 / C1; % resistor at the op amp input.
C2 = R2C2 / R2; % capacitor in parallel with R2

disp('');
disp(['C1 = ' num2str(C1*1e12) ' pF']);
disp(['R1 = ' num2str(R1) ' ohms']);
disp(['C2 = ' num2str(C2*1e12) ' pF']);
disp(['R2 = ' num2str(R2) ' ohms']);
```

D.5 vco.m

```
% vco.m  VCO gain curve fitter / Z-COMM type V605ME01  SN-0019  0495
clear
close all

Vt=[
   0
   0.5
   1
   1.5
   2
   2.5
   3
   3.5
   4
   4.5
   5
]

Fout=[
   1.3277
   1.3583
   1.3797
   1.3983
   1.4150
   1.4297
   1.4447
   1.4597
   1.4737
   1.4877
   1.5017
]


plot(Vt,Fout,'sr');

poly_coeffs=polyfit(Vt,Fout,5);

Vt_fine=linspace(0,5,10000);
```

```
Fout_fine=polyval(poly_coeffs,Vt_fine);

hold on

plot(Vt_fine,Fout_fine,'-.');
title('Polynomial Fit to measured VCO tuning Response');
xlabel('Tuning Voltage /V');
ylabel('Output Frequency /GHz');
grid

figure

% VCO gain = d(Hz)/d(V);

len=length(Vt_fine);

for n=2:len
   VCO_gain(n)=(Fout_fine(n)-Fout_fine(n-1))/(Vt_fine(n)-Vt_fine(n-1));
end

% if len is large roughly
VCO_gain(1)=VCO_gain(2);

plot(Vt_fine,VCO_gain.*1000);
title('VCO gain variation')
xlabel('VCO Tuning Voltage /V')
ylabel('VCO Tuning Gain  MHz/V')
grid
```

D.6 PM.m

```
% plot phase margin across band
% plot the simulated Closed Loop Gain across the band
% uses imported spice data for loop filter and variable VCO gain.

clear
close all

fit_len=10000; % length of curve fit to VCO data

% Read in spice output data for filter with enhanced filtering

[freq,gain_dB]=textread('gain_enhanced','%f %f');
[freq2,phase_deg]=textread('phase_enhanced','%f %f');

% set up a complex frequency vector
omega=2*pi.*freq; % natural frequency range from spice freq axis
sigma=zeros(1,length(freq))';
s=sigma+j.*omega;



% Interpolate the VCO tuning data and calculate gain across band

% VCO tuning voltage
Vt=[-1  -0.5  0  0.5  1  1.5  2  2.5  3  3.5  4  4.5  5];
% output frequency in GHz
Fout=[1.2405  1.30617  1.36160  1.38993  1.41313  1.4334  1.4526  1.47  1.4878  1.5044  1.5204  1.53687  1.5527];

% Fit polynomial to VCO data
%plot(Vt,Fout,'sr');
poly_coeffs=polyfit(Vt,Fout,5);
hold;


% step from 0 to 5 volts in 10000 steps.
Vt_fine=linspace(-1,5,fit_len);
```

```
Fout_fine=polyval(poly_coeffs,Vt_fine);

%plot(Vt_fine,Fout_fine);
%title('VCO tuning characteristic');

% VCO gain = d(Hz)/d(V);
len=length(Vt_fine);
for n=2:len
   VCO_gain(n)=(Fout_fine(n)-Fout_fine(n-1))/(Vt_fine(n)-Vt_fine(n-1));
end
% if len is large
VCO_gain(1)=VCO_gain(2);
VCO_gain=1000.*VCO_gain; % convert from GHz to MHz/V


plot(Fout_fine,VCO_gain);
title('VCO Tuning Gain');
xlabel('Output Frequency /MHz');
ylabel('Gain MHz/V');
grid

% phase detector gain in volts per radian, assumed constant.
Vo_h=4;
Vo_l=3.2;
swing=(Vo_h-Vo_l);
Kp=swing/(2*pi);

% convert loop filter response to complex form
% this data already has a log x axis
r=10.^(gain_dB./20); % dB to magnitude
phi=pi.*phase_deg./180; % degrees to radians


G=complex(r.*cos(phi),r.*sin(phi));
% following plots the spice response
%figure(3);
%plot(10*log10(abs(G)));
%figure(4);
%plot(pi.*unwrap(angle(G))./180);



Fref=156e6; % designed reference frequency

index=100;

for index=1:100

Kvco=2*pi*1e6*VCO_gain(index*100);
Ndiv=Fout_fine(index*100)*1e9/Fref;
Frequency=Ndiv*Fref;

f_path=(G./s).*(Kp*Kvco);
loop_gain=(G./s).*(Kp*Kvco/Ndiv);
phase_resp= (f_path)./(1+loop_gain);

dB_loop_gain=20*log10(abs(loop_gain));

% find the phase margin by minimising the size of the gain
gain_squared=abs(dB_loop_gain.^2);
[val,pos]=min(gain_squared);

phase_unwrapped=unwrap(angle(loop_gain));
phase_marg(index)=180+unwrap(phase_unwrapped(pos)*180/pi);

end

%figure
%semilogx(freq,dB_loop_gain,'r');
%grid
%hold;
%semilogx(freq,unwrap(angle(loop_gain))*180/pi,'g');

%axis([1e4 1e7 -200 100]);
```

```
%figure
%plot(phase_marg)

figure
plot(decimate(Fout_fine,100),phase_marg);
xlabel('Output Frequency /GHz');
ylabel('Phase Margin');
title('Phase Margin Variation of Prototype Loop');
grid
```

D.7 modsim.m

```
% modsim.m routine to call simulink sigma deltas and plot output spectrum.

clear
close all

desired_input=0.4;
fref=4; % sampling frequency
f_nyq=fref/2;
nn=1024*64; % samples used for simulation
k=1000; % truncation factor for startup
simpoints=nn+k;

actual_input=round(desired_input*simpoints)/simpoints % nearest available input value

invec=ones(1,nn+k);
invec=invec'.*actual_input; % vector input to modulator

timevec=(linspace(1,nn+k,nn+k))'; % simulink requires time next to the value.
inmat2(:,1)=timevec; % fill up the simulink matrix
inmat2(:,2)=invec;

sim('third',simpoints); % call simulink model, second param= no. samples.
Vc(1:nn)=bitstream(1+k:nn+k);

figure(1);
stairs(Vc); % plot time domain modulator output
axis([100 200 -0.25 1.25]);
grid

DC=mean(Vc);
Vc=Vc-DC; % remove DC component

window=hann(nn);
window=window';
%Vc=Vc.*window; % hanning window option
%Vc=Vc.*2; % voltage level correction for hann window

l=length(Vc);
pwr=abs(fft(Vc)).^2/l;
pwr=pwr*2; % convert to single sided spectrum

out_len=length(pwr); % measure output length
pwr=pwr/l; % convert to power W
pwr=10*log10(pwr+10e-2000); % log scale dBW


f=[0:1/out_len:1-1/out_len]; % vector to form the frequency axis
f=f.*fref; % multiply by fref.

figure(2);
semilogx(f,pwr); % plot power spectrum
grid on;
xlabel('Frequency normalised to nyquist')
ylabel('Power Spectrum  dBW')
```

```
title('Third Order Sigma Delta Spectrum, Stepsize \Delta =1V')
axis([10e-4 1 -160 0])
```

D.8 system.m

```
% simulate sigma delta modulator and plot contribution to PLL sidebands

clear
close all

% define p/q, fref and no. of samples
nn=1024*16
fmin=1805e6;
fmax=1880e6;
fref = (fmin+fmax)/17
fout=1879e6; %desired fout in MHz
desiredpq=fout/fref-8
actualpq=round(desiredpq*nn)/nn %nearest available p/q value
pq=actualpq;
%pq=desiredpq;
k=1500

% define prescaler modulation waveform

Vc=zeros(1,nn+k);
% initiate an array

b=0;
d=0;
f=0;
x=0;

% third order sigma-delta feedback coefficients
aa=8.765;
bb=11.002;

for n=2:nn+k;

    b=b+pq-x;
    d=d+b-aa*x;
f=f+d-bb*x;
    if f > 0.5
        Vc(n)=1;
x=1;
    else
        Vc(n)=0;
x=0;
    end
end

Vcy=Vc;
save data Vcy;
Vcx(1:nn)=Vc(1+k:nn+k);
Vc=Vcx;

% integrate frequency steps to get phase steps (with zero mean value)
l=length(Vc);
meanVc= mean(Vc)
deltaphi=2*pi*cumsum(Vc-meanVc);

meandeltaphi=mean(deltaphi);
deltaphi=deltaphi-meandeltaphi;

excess1bits=nn*(meanVc-actualpq)

% take spectrum of phase modulation due to the prescaler modulation waveform
fn=(2/l)*fft(deltaphi);
```

```
% -6dB narrow-band PM conversion - giving output spectrum in absence of any loop filter
fn=fn/2;
mag=10*log10((fn.*conj(fn)+10e-11));

% convert to dBc/Hz by dividing by FFT resolution (fref/1)
mag=mag-10*log10(fref/1);
f=[0:1/1:1-1/1];

% third order loop transfer function
phi=pi/3;
% normalised loop natural frequency
fn=250e3;
fn=fn/fref;
T=tan(phi)+sec(phi);
ff=f/fn;
response=(1+T*T*ff.*ff)./((1-T*ff.*ff).*(1-T*ff.*ff)+(T*ff-ff.*ff.*ff).*(T*ff-ff.*ff.*ff));
response=response.^0.5;
response=20*log10(response);

% apply third order loop response
mag2=mag+response;

[s,t]=max(mag2);

maxspurdbc = s + 10*log10(fref/1)
atfreq = t*fref/nn

% to return the close-in phase noise power in dBc/Hz
closenoise=mag2(100)

semilogx(f, response, f, mag2,f,mag)

% DCS phase noise mask
xmask=[1e5/fref 2e5/fref 2e5/fref 2.5e5/fref 2.5e5/fref 4e5/fref 4e5/fref 6e5/fref 6e5/fref 12e5/fref 12e5/fref 18e5/fref 18e5/fref
ymask=[-44 -44 -75 -75 -78 -78 -105 -105 -115 -115 -118 -118 -125 -125 -130 -130];
line(xmask, ymask)

% DCS spurs mask
xmask2=[1e5/fref 2e5/fref 2e5/fref 2.5e5/fref 2.5e5/fref 4e5/fref 4e5/fref 6e5/fref 6e5/fref 12e5/fref 12e5/fref 18e5/fref 18e5/fre
ymask2=[-32 -32 -62 -62 -65 -65 -92 -92 -102 -102 -105 -105 -107 -107 -112 -112];
line(xmask2, ymask2)

axis([0.001,0.5,-160,10])

xlabel('Frequency/fref')
ylabel('Response / dBc')
```

D.9 spec2.m

```
% bitstream plotting routine to match measured results at CPLD output
clear
close all

fname='lll'; % enter the filename here xxx.mat

c_ref=-30; % spectrum analyser reference level.

load(fname);
nn=length(Vc);
fref = 156*1e6; % synth reference frequency.
fout=(9-mean(Vc))*fref % derive fout from mean(Vc)and average of bits.
desiredpq=mean(Vc);
actualpq=round(desiredpq*nn)/nn; % nearest available p/q value
pq=actualpq;
freq=fout/1e9; % to provide frequency in GHz.
str=num2str(freq); % convert to string to include in plot title
```

```
l=length(Vc);
meanVc=mean(Vc);
Vc=Vc.*5; % scale the bit sequence to CMOS voltage levels 0/5 V

spec=fft(Vc)*2/nn; % take spectrum of voltage steps
pwr=(abs(spec)).^2; % convert to power
pwr=10*log10(pwr);

pwr=pwr-30; % convert to dBm
faxis=linspace(0,fref/1e6,nn); % frequency vector for plot axis

plot(faxis,pwr,'k');
xlabel('Frequency /MHz');
ylabel('Power Spectrum /dBV');
axis([0,10,-130,-30]);
grid on;

newstr=strcat('Simulated bitstream spectrum, F_{REF} = 156 MHz, F_{OUT} = ',str,' GHz');
title(newstr);

text(1.5,-35,'-30 dBm Analyser Reference Level');
```

D.10 pnoise.m

```
% simulating phase noise at loop output

clear
close all

fname='111'; % enter the filename here xxx.mat
c_ref=-15; % shift carrier reference level to match measured.
PM=50;
str_deg=num2str(PM); % to display phase margin in output plot

load(fname);
nn=length(Vc);
fref = 156*1e6; % synth reference frequency.
fout=(9-mean(Vc))*fref % derive fout from mean(Vc)and average of bits.
desiredpq=mean(Vc);
actualpq=round(desiredpq*nn)/nn; %nearest available p/q value
pq=actualpq;
N=9-mean(Vc) % synth divide ratio

freq=fout/1e9; % to provide frequency in GHz.
str=num2str(freq);

% integrate frequency steps to get phase steps (with zero mean value)
l=length(Vc);
meanVc= mean(Vc) % Vc corresponds to the frequency steps
deltaphi=2*pi*cumsum(Vc-meanVc); % the integration step

meandeltaphi=mean(deltaphi);
deltaphi=deltaphi-meandeltaphi; % remove DC component.
excessibits=nn*(meanVc-actualpq)

% take spectrum of phase modulation
spec=fft(deltaphi)*2/nn;
pwr=abs(spec).^2;  % convert to normalised power
pwr=10*log10(pwr); % convert to dBc

% -6dB narrow-band PM conversion
pwr=pwr-6;

% comment out for dBc, or use to convert to dBc/Hz by dividing by FFT resolution (fref/l)
pwr=pwr-10*log10(fref/l);
```

```
% frequency vector used for filter response.
f=linspace(0,fref,nn);
%f=linspace(0,fref,nn); % frequency input for loop filter calculation
sigma=zeros(1,nn);
s=sigma+j*2*pi*f;
phi=(PM/360)*2*pi;
% set the phase margin for the loop filter
fn=250e3; % loop natural frequency (Hz)
wn=fn*2*pi; % loop natural frequency (rad/sec)
T=tan(phi)+sec(phi);
response=N.*(wn^2.*T.*s+wn^3)./(s.^3+wn*T.*s.^2+wn^2*T.*s+wn.^3); % third order loop transfer function
res_mag=abs(response);
res_m_dB=20*log10(res_mag);

pwr_filt=pwr+res_m_dB; % apply filter response to SD spectrum.
faxis=linspace(0,fref/1e6,nn); % frequency vector for plot axis

semilogx(faxis,pwr_filt,'k',faxis,res_m_dB);
xlabel('Carrier offset /MHz')
ylabel('Power density  / dBc/Hz,       Closed loop response / dB')
axis([10e-2,10,-150,50]);
grid on;

newstr=strcat('F_{REF} = 156 MHz,   F_{OUT} = ',str,' GHz,   Phase Margin =',str_deg,'^o');
title(newstr);
```

D.11 stab.m

```
% sigma delta root locus plotting routine
% finding the root locus by scanning for sign change
% of imaginary part of quantiser gain

clear
close all

N=6; % filter parameters
a=0.81;
b=1;

alpha=-1/2; % worst case stability scenario
range=2; % plus and minus range all axes
res_count=1; % array counter for storing root locus

re_max=range; % maximum real axis
re_min=-range; % minimum real axis
im_max=range; % maximum imag axis
im_min=-range; % minimum imag axis

steps=50; % steps across one side of the cube
step_re=(re_max-re_min)/steps; %real step size
step_im=(im_max-im_min)/steps; %imag step size

angle=zeros(1,steps);
trans=zeros(1,steps);
sweep=zeros(steps,steps);
%min_inner=zeros(1,steps);
store=zeros(steps,2);

crossings=zeros(steps,steps);
real_l=zeros(steps,steps); % zeros matrix for real part of lambda
imag_l=zeros(steps,steps); % zeros matrix for imag part of lambda
lambda=complex(real_l,imag_l); % store the result for lambda over the grid

% scan left to right moving up the imag axis.
for n2=1:steps %sweep the imag value in outer loop
imag_val=(n2-(steps/2))*step_im; % convert grid step to absolute value

for n1=1:steps  %sweep the real value
    real_val(n1)=(n1-(steps/2))*step_re; % convert grid step to absolute value
```

```
r=(real_val(n1)^2+imag_val^2)^(1/2); % get radius

       % section to generate theta
       if (n1==steps/2)&(n2>steps/2);
          theta=pi/2; % theta = 90 degrees
       elseif (n1==steps/2)&(n2<steps/2);
          theta=-pi/2; % theta = -90 degrees
       elseif (n2==steps/2)&(n1<steps/2);
          theta=pi; % theta = 180 degrees
       elseif (n2==steps/2)&(n1>steps/2);
          theta=0; % theta = 0 degrees

       elseif (n1>steps/2)&(n2>steps/2);
          theta=atan(abs(imag_val/real_val(n1))); % top right quadrant
       elseif (n1<steps/2)&(n2>steps/2);
          theta=pi/2+atan(abs(real_val(n1)/imag_val));  % top left quadrant

       elseif (n1<steps/2)&(n2<steps/2);
          theta=pi+atan(abs(imag_val/real_val(n1))); % bottom left quad
       else (n1>steps/2)&(n2<steps/2);
          theta=atan(imag_val/real_val(n1)); % bottom right quad
       end


      % assemble the complex variable
      z=r*exp(j*theta);
      % calculate delta phi, the phase uncertainty.
      delta_phi=(abs(theta+(pi/2))-abs(theta-(pi/2))-theta);

      % calculate the loop filter transfer function G(s)

G=((1-a*z^-1)^N/(1-b*z^-1)^N)-1;
% Nth order loop filter response

      % evaluate lambda
      lambda(n1,n2)=-1/(exp(j*alpha*delta_phi)*G);

      if (n1>1)&(n2>1)  % avoid exceeding matrix dimensions
       x1_val=imag(lambda(n1-1,n2));
         x2_val=imag(lambda(n1,n2));
         y1_val=imag(lambda(n1,n2-1));
          % tests for zero crossing
if (x2_val>0&x1_val<0)|(x1_val>0&x2_val<0)|(x2_val>0&y1_val<0)|(y1_val>0&x2_val<0);
               crossings(n2,n1)=32;
        end
      end
   end

   sweep(n2,:)=trans; % sweep[] holds real value sweeps at different imag values
end

% turn this the right way around to plot it.
lambda=lambda';

% lambda needs extra column and row to correct grid alignment
new_mat=zeros(steps+1,steps+1);
new_mat(2:steps+1,2:steps+1)=lambda;

% corrected axes for displaying output
re_axis=linspace(re_min,re_max,steps);
im_axis=linspace(im_min,im_max,steps);

% corrected axes for displaying output
re_axis_shift=linspace(re_min,re_max,steps+1);
im_axis_shift=linspace(im_min,im_max,steps+1);

%figure
%surf(re_axis_shift,im_axis_shift,abs(new_mat));
%xlabel('Real Axis')
%ylabel('Imaginary Axis')
%zlabel('Absolute value of lambda');

%figure
```

```
%surf(re_axis_shift,im_axis_shift,imag(new_mat));
%xlabel('Real Axis')
%ylabel('Imaginary Axis')
%zlabel('Imaginary part of lambda')

figure(1)
colormap([0 0 0])
contour(re_axis_shift,im_axis_shift,imag(new_mat),[0 0]);
grid
xlabel('Real Axis')
ylabel('Imaginary Axis')
zlabel('Imaginary part of lambda')
hold on
freq=linspace(0,2*pi,1000);x_cir=sin(freq);y_cir=cos(freq);
plot(x_cir,y_cir);
hold on

figure(2)
colormap([0 0 0])
mesh(re_axis_shift,im_axis_shift,imag(new_mat));
xlabel('Real Axis')
ylabel('Imaginary Axis')
zlabel('Imaginary part of lambda')
```

Appendix E

Prototype Circuit Diagrams

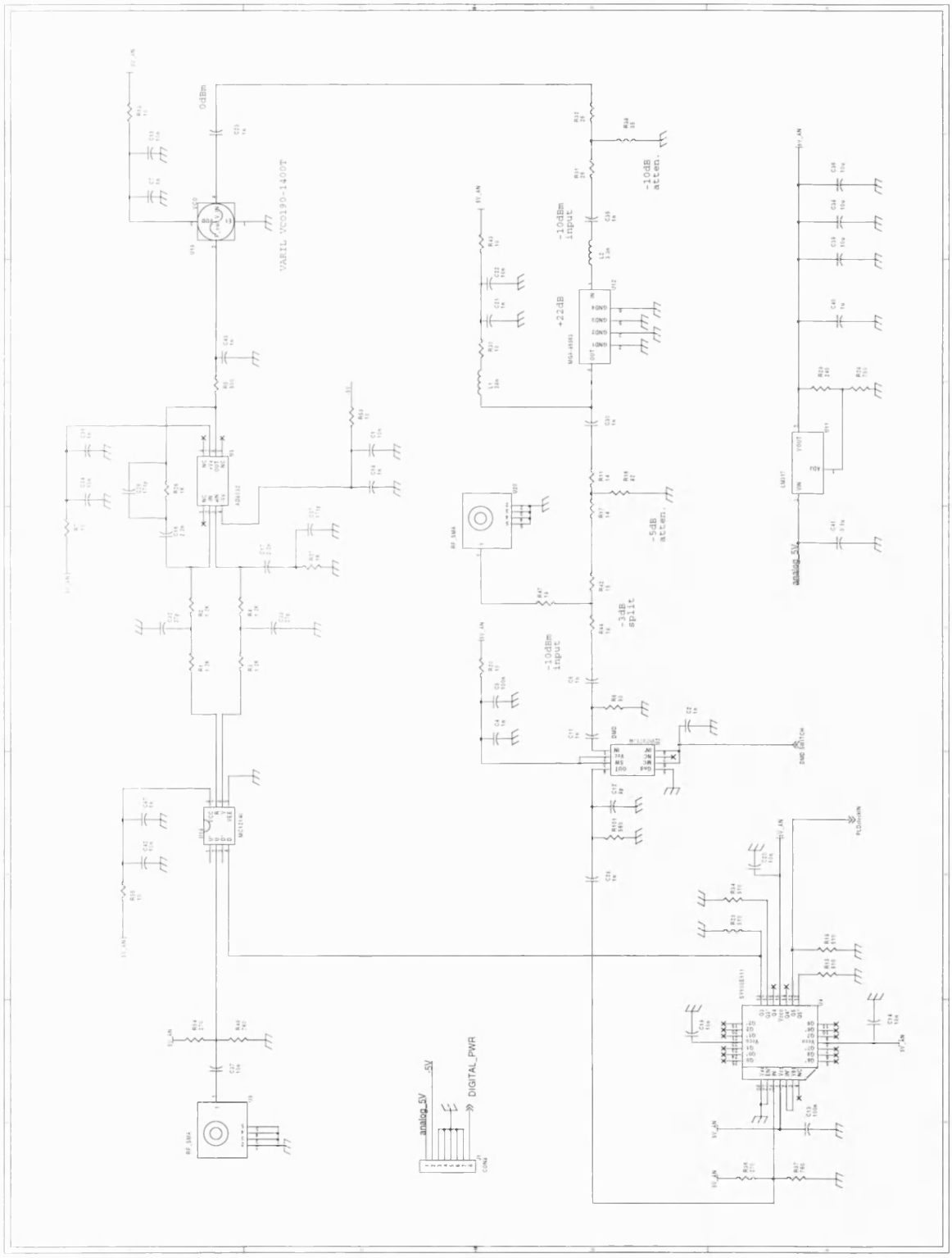


Figure E.1: Circuit diagram of loop section of prototype synthesiser.

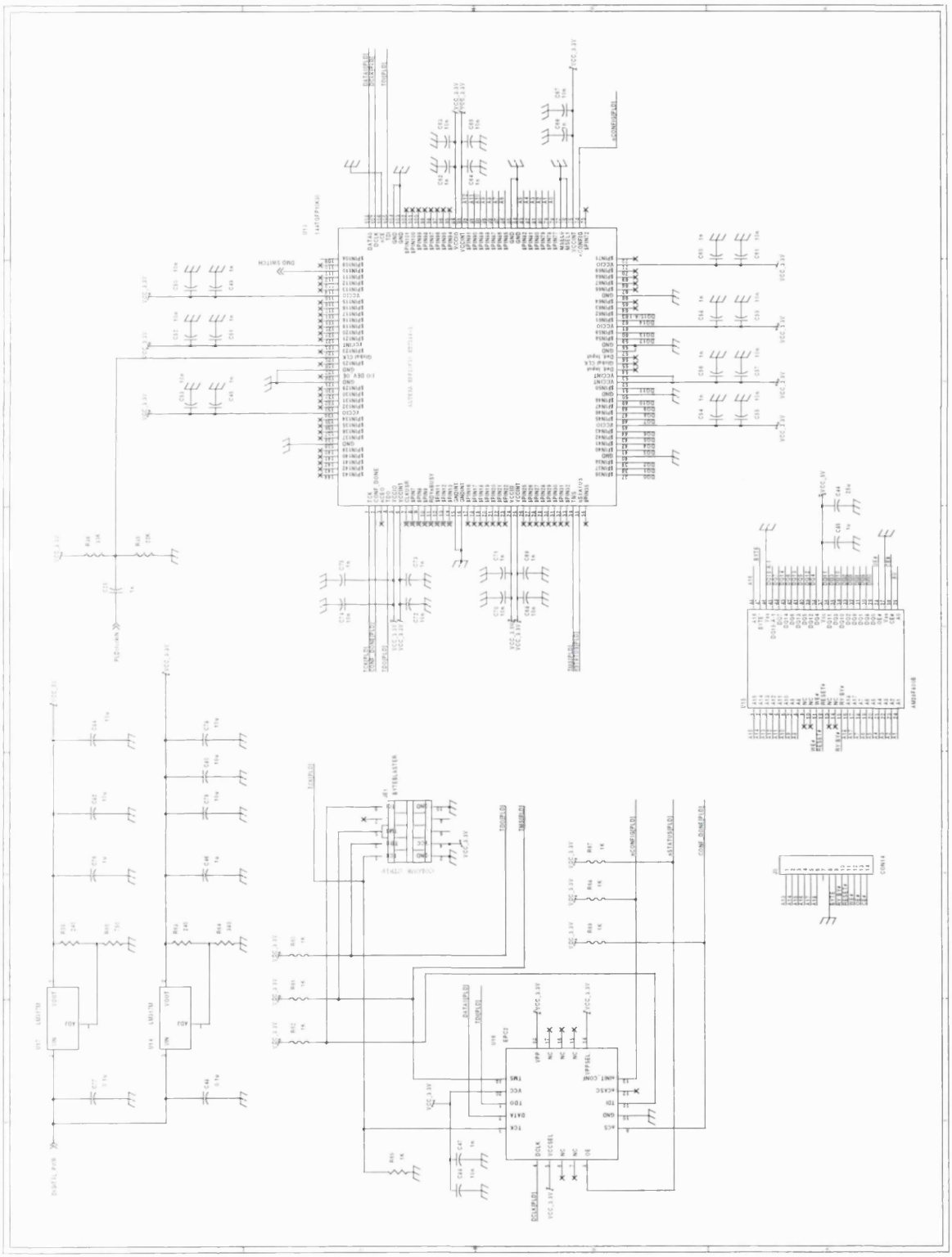


Figure E.2: Circuit diagram of digital section of prototype synthesiser.

Appendix F

Photo of Prototype

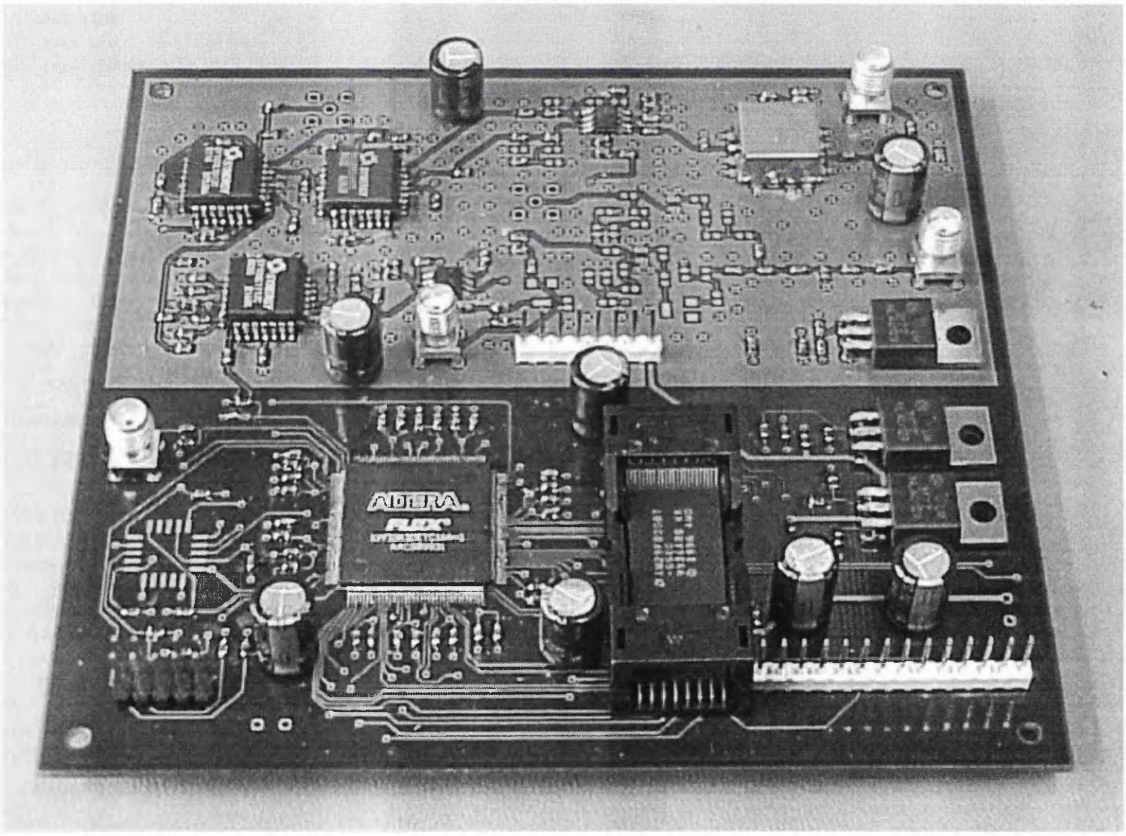


Figure F.1: Prototype Circuit Board used for Stored Sequence Synthesiser

Appendix G

VHDL Code for Logic

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
use ieee.numeric_std.all;


--------------The Top Level Design Entity ------------------

ENTITY SCHEMATIC1 IS PORT (
addr : OUT std_logic_vector(15 DOWNTO 0);
D : IN std_logic_vector(15 DOWNTO 0);
ck : IN std_logic;
reset : IN std_logic;
dout : OUT std_logic
); END SCHEMATIC1;


ARCHITECTURE STRUCTURE OF SCHEMATIC1 IS

-- COMPONENTS

COMPONENT reg16en
PORT (
reset : IN std_logic;
ck : IN std_logic;
enable : IN std_logic;
d : IN std_logic_vector(15 DOWNTO 0);
q : OUT std_logic_vector(15 DOWNTO 0)
); END COMPONENT;

COMPONENT mux16_1
PORT (
D : IN std_logic_vector(15 DOWNTO 0);
sel : IN std_logic_vector(3 DOWNTO 0);
clock : IN std_logic;
dout : OUT std_logic
); END COMPONENT;

COMPONENT count16b
PORT (
reset : IN std_logic;
ck : IN std_logic;
count : OUT std_logic_vector(15 DOWNTO 0)
); END COMPONENT;

COMPONENT decode
PORT (
a : IN std_logic_vector(3 DOWNTO 0);
q : OUT std_logic_vector(15 DOWNTO 0)
); END COMPONENT;
```

```
COMPONENT count4b
PORT (
reset : IN std_logic;
ck : IN std_logic;
count : OUT std_logic_vector(3 DOWNTO 0)
); END COMPONENT;

-- SIGNALS

SIGNAL COUNT : std_logic_vector(3 DOWNTO 0);
SIGNAL LATCH : std_logic_vector(15 DOWNTO 0);
SIGNAL Q : std_logic_vector(15 DOWNTO 0);

-- INSTANCE ATTRIBUTES


-- GATE INSTANCES

BEGIN
reg16bit : reg16en PORT MAP(
reset => RESET,
ck => CK,
enable => Q(1),
d(15) => D(15),
d(14) => D(14),
d(13) => D(13),
d(12) => D(12),
d(11) => D(11),
d(10) => D(10),
d(9) => D(9),
d(8) => D(8),
d(7) => D(7),
d(6) => D(6),
d(5) => D(5),
d(4) => D(4),
d(3) => D(3),
d(2) => D(2),
d(1) => D(1),
d(0) => D(0),
q(15) => LATCH(15),
q(14) => LATCH(14),
q(13) => LATCH(13),
q(12) => LATCH(12),
q(11) => LATCH(11),
q(10) => LATCH(10),
q(9) => LATCH(9),
q(8) => LATCH(8),
q(7) => LATCH(7),
q(6) => LATCH(6),
q(5) => LATCH(5),
q(4) => LATCH(4),
q(3) => LATCH(3),
q(2) => LATCH(2),
q(1) => LATCH(1),
q(0) => LATCH(0)
);
mux16bit : mux16_1 PORT MAP(
D(15) => LATCH(15),
D(14) => LATCH(14),
D(13) => LATCH(13),
D(12) => LATCH(12),
D(11) => LATCH(11),
D(10) => LATCH(10),
D(9) => LATCH(9),
D(8) => LATCH(8),
D(7) => LATCH(7),
D(6) => LATCH(6),
D(5) => LATCH(5),
D(4) => LATCH(4),
D(3) => LATCH(3),
D(2) => LATCH(2),
D(1) => LATCH(1),
D(0) => LATCH(0),
sel(3) => COUNT(3),
```

```
sel(2) => COUNT(2),
sel(1) => COUNT(1),
sel(0) => COUNT(0),
clock => CK,
dout => DOUT
);
count16b : count16b PORT MAP(
reset => RESET,
ck => Q(1),
count(15) => ADDR(15),
count(14) => ADDR(14),
count(13) => ADDR(13),
count(12) => ADDR(12),
count(11) => ADDR(11),
count(10) => ADDR(10),
count(9) => ADDR(9),
count(8) => ADDR(8),
count(7) => ADDR(7),
count(6) => ADDR(6),
count(5) => ADDR(5),
count(4) => ADDR(4),
count(3) => ADDR(3),
count(2) => ADDR(2),
count(1) => ADDR(1),
count(0) => ADDR(0)
);
decoder : decode PORT MAP(
a(3) => COUNT(3),
a(2) => COUNT(2),
a(1) => COUNT(1),
a(0) => COUNT(0),
q(15) => Q(15),
q(14) => Q(14),
q(13) => Q(13),
q(12) => Q(12),
q(11) => Q(11),
q(10) => Q(10),
q(9) => Q(9),
q(8) => Q(8),
q(7) => Q(7),
q(6) => Q(6),
q(5) => Q(5),
q(4) => Q(4),
q(3) => Q(3),
q(2) => Q(2),
q(1) => Q(1),
q(0) => Q(0)
);
count4b : count4b PORT MAP(
reset => RESET,
ck => CK,
count(3) => COUNT(3),
count(2) => COUNT(2),
count(1) => COUNT(1),
count(0) => COUNT(0)
);
END STRUCTURE;

------------lower level entities----------------------


entity count16b is
  port(reset, ck : in std_logic;
       count     : out std_logic_vector(15 downto 0));
end;

architecture behavior of count16b is
   signal count_n: unsigned(15 downto 0);
begin
  process (reset, ck, count_n)
  begin
    if reset='1' then
      count_n <= x"0000";
    elsif (ck='1' and ck'event) then
```

```
      count_n <= count_n + "1";
    end if;
  end process;
  count <= std_logic_vector(count_n);
end;

-------------------------------------------------------

entity decode is
  port (
    a : in std_logic_vector(3 downto 0);
    q : out std_logic_vector(15 downto 0));
end decode;

architecture behavior of decode is
   signal n_a : unsigned(3 downto 0);
   signal n_q : unsigned(15 downto 0);
begin
   n_a <= unsigned(a);
   process (n_a)
   begin
--     for i in natural range 15 downto 0 loop
     for i in integer range 15 downto 0 loop
       if (n_a=i) then
         n_q(i) <= '1';
       else
         n_q(i) <= '0';
       end if;
     end loop;
   end process;
   q <= std_logic_vector(n_q);
end behavior;
-------------------------------------------------------------

entity count4b is
  port(reset, ck : in std_logic;
       count     : out std_logic_vector(3 downto 0));
end;

architecture behavior of count4b is
   signal count_n: unsigned(3 downto 0);
begin
  process (reset, ck, count_n)
  begin
    if reset='1' then
      count_n <= x"0";
    elsif (ck='1' and ck'event) then
      count_n <= count_n + "1";
    end if;
  end process;
  count <= std_logic_vector(count_n);
end;

-------------------------------------------------------

entity mux16_1 is
  port(D    : in std_logic_vector(15 downto 0);
    sel     : in std_logic_vector(3 downto 0);
    clock   : in std_logic;
    dout    : out std_logic);
end;

-- two process description style with local variable mux_val
architecture behaviour of mux16_1 is
  signal mux_val: std_logic;
begin
  process(D,sel)
  begin
    if    (sel="0000") then
      mux_val <= D(0);
    elsif (sel="0001") then
      mux_val <= D(1);
    elsif (sel="0010") then
      mux_val <= D(2);
```

```
    elsif (sel="0011") then
      mux_val <= D(3);
    elsif (sel="0100") then
      mux_val <= D(4);
    elsif (sel="0101") then
      mux_val <= D(5);
    elsif (sel="0110") then
      mux_val <= D(6);
    elsif (sel="0111") then
      mux_val <= D(7);
    elsif (sel="1000") then
      mux_val <= D(8);
    elsif (sel="1001") then
      mux_val <= D(9);
    elsif (sel="1010") then
      mux_val <= D(10);
    elsif (sel="1011") then
      mux_val <= D(11);
    elsif (sel="1100") then
      mux_val <= D(12);
    elsif (sel="1101") then
      mux_val <= D(13);
    elsif (sel="1110") then
      mux_val <= D(14);
    else
      mux_val <= D(15);
    end if;
  end process;

  process
  begin
    wait until clock'event and clock='1';
  dout <= mux_val;
  end process;
end behaviour;
```

Appendix H

Paper Submission

An Agile stored ΣΔ sequence Fractional-N synthesiser

Rob Walkington and Dr. Paul Brennan

Department of Electronic and Electrical Engineering, University College London, Torrington Place, London, WC1E 7JE, United Kingdom

Abstract – **The performance of ΣΔ Fractional-N frequency synthesisers has a direct relation to reference frequency, the level of quantisation noise dropping by 6dB for each doubling of reference frequency. The upper limit on this reference frequency is often caused by the modulator, due to the limited speed achievable in fixed point hardware. Fixed point modulator feedback coefficients with limited precision also reduce modulator cycle length leading to unavoidable periodicity in the modulator output stream. To avoid these problems, a synthesiser has been designed which is particularly suited to burst mode systems such as DCS1800. The prototype described here stores pre-generated ΣΔ sequences in fast memory for each required channel, allowing a 'virtual' ΣΔ modulator operating at 240MHz to be implemented with a low cost FPGA and Flash memory.**

I. Introduction

Frequency hopping is commonly used to combat multipath fading in mobile radio. Requirements for agility and dynamic range are often severe and beyond the performance of an integer phase locked loop (PLL). DCS1800 is one such application, which commonly switches between two phase locked loops to achieve the required speed and spectral purity. This is a costly approach which does not lend itself to integration due to the difficulty of isolating the loops.

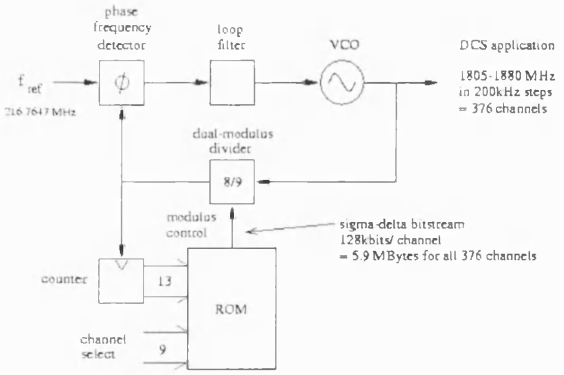


Fig 1. A Stored sequence fractional-N synthesiser

A recent approach which has proved popular in the quest for higher performance is a fractional-N synthesiser controlled by a sigma delta modulator [1]. This technique has the distinct advantage of high pass filtering and randomising the divider switching noise so it can be easily removed by the loop filter.

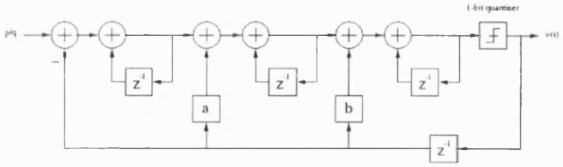


Fig 2. The 3rd order sigma delta modulator

A significant restriction on the performance of current implementations of such a synthesiser is often the modulator speed. Bottlenecks in operation speed are caused by hardware adders and multipliers and become worse as the data width increases, leading to a trade off between speed and precision. The use of fixed point feedback coefficients has a particularly damaging effect on modulator performance as it results in a reduction in the randomising of the modulator output, leading to shorter limit cycles and more in band tones. It would also be advantageous to be able to adjust modulator coefficients to suit the synthesised frequency, as the coefficients could then be optimised for each section of the band.

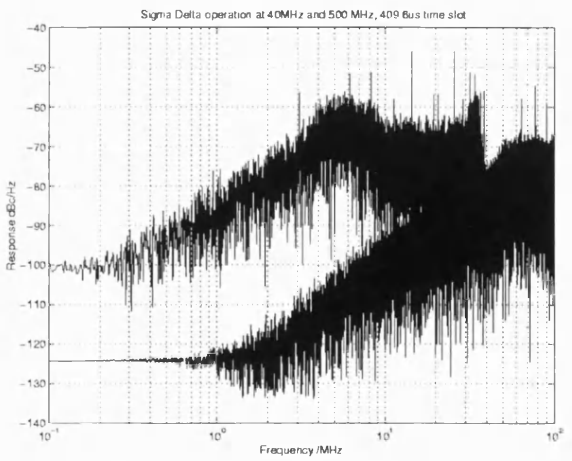


Fig 3. The effect of reference frequency on modulator NBPM referred to the loop output.

By exploiting the nature of frequency hopping systems, fabrication of a hardware sigma delta modulator can be avoided completely and instead an optimised sigma delta sequence for each required channel stored in fast memory [3]. Making the stored sequence as long as the time slot length ensures no undesirable truncation effects. This also allows reference frequencies of several hundred MHz which provides a significant increase in the noise spreading performance of the modulator. This paper describes simulation and implementation of such a synthesiser.

II. Sigma Delta Fractional-N Synthesisers

Fig 1. Shows an outline of a stored sequence synthesiser. A dual modulus divider placed in the feedback path is switched in a continuous manner to give an average division ratio between N and N+1. Fractional division allows a larger reference frequency for a given loop output resolution resulting in reference spurs further from the carrier which are less demanding to filter out, the ultimate aim being to widen the loop bandwidth and achieve much faster channel changing.

Figure 2 shows a block diagram of a third order sigma delta modulator. The modulator output is a pulse width modulated representation of its instantaneous input. Using a DC input to the modulator and one bit quantisation the feedback loop ensures that the average value at the modulator output equals that at the input, but a high pass response will be applied to the switching noise. By modelling the quantiser as a separate additive source the suppression of quantisation noise can be written using masons rule:

$$\frac{Y(z)}{N(z)} = \frac{(z-1)^3}{z^3-(2+a+b)z^2+(3+a+2b)z-1-b} \quad (1)$$

Where N(z) is the noise added due to the quantiser, Y(z) is the resulting noise at the output and a and b are negative feedback coefficients. Setting a and b equal to minus 1 gives a high pass characteristic of 60dB/ decade although in a real modulator instability results, leading to a requirement for more negative feedback and less noise suppression. Stability analysis is generally complicated due to the quantiser presence [4] although simulation can be used to obtain the best noise suppression consistent with stable operation.

Figure 3 shows the simulated dual modulus divider phase noise referred to the loop output for 40MHz and 500MHz sampling frequencies. It is no surprise to note that the level of the phase noise follows standard sampling theory, with a relation $20\log(F_s)$. Consequently each doubling of the sampling frequency will lead to a 6dB drop in the sigma delta noise. By storing the divider control sequence for each required channel in memory the necessity for real time adders or multipliers is avoided and it becomes feasible to fabricate sigma delta fractional-N loops with reference frequencies up to the GHz range, with suitable high speed logic between the memory and prescaler.

II. Simulation of Stored Sequence Synthesiser

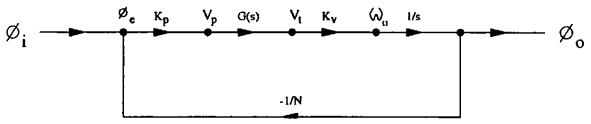


Fig 4. A generic synthesiser system

Figure. 4 shows a signal flow graph for a generic synthesiser system, where ϕ_o is the phase at the VCO output and ϕ_i is the phase at the reference input. K_p is the phase detector gain in Vrad^{-1}, G(s) is the loop filter transfer function and K_v is the VCO gain in rad/s/V. The additional 1/s term is due to the integrating action of the VCO and translates the output frequency ω_o to phase. With a third order type two loop filter, the closed loop response to phase variations at the divider input is given by [5]:

$$\frac{\phi_o(s)}{\phi_d(s)} = \left(\frac{\omega_n^2\psi s+\omega_n^3}{s^3+\omega_n\psi s^2+\omega_n^2\psi s+\omega_n^3}\right) = P(s) \quad (2)$$

$$(\psi = \tan\phi_{PM} + \sec\phi_{PM})$$

Where ϕ_{PM} is the loop phase margin. The N divider value is not required to calculate the loop phase noise response as the divider phase, ϕ_d, is referred to the divider output. Each time the divider is switched a step change in phase of plus or minus 2π is introduced at the divider output when V_{mod} changes between its two values. This will be added or subtracted from the cumulative sum of phase due to the divider switching. The phase at the start of the sigma delta sequence is set to zero to simulate an already locked loop. Therefore the loop phase response over the time slot can calculated. (4) removes the DC component of the modulator voltage, takes the cumulative sum of the voltage steps and converts them to phase steps, where n_s is the number of samples required at the chosen reference to cover the required time slot. V_{mod} is either 1 or 0:

$$\Delta\phi[n] = 2\pi\sum_{1}^{n_s}\left(V_{mod}[n]-\overline{V_{mod}}\right) \quad (3)$$

Taking the spectrum of the phase modulation, a $2/n_s$ factor equalises the peak amplitudes on both sides of the FFT. A further zero-meaning of the phase steps is

required to represent the case of a locked loop:

$$\phi_d(s) = \frac{2}{n_s} \mathrm{FFT}\left[\Delta\phi[n] - \overline{\Delta\phi[n]}\right] \quad (4)$$

Phase noise spectral density at the synthesiser output in dBc/Hz is found by applying the closed loop response, P(s), to the dual modulus divider phase modulation spectrum, $\phi_d(s)$, and applying dBc/Hz and NBPM scaling:

$$\phi_o(s) = \frac{2}{n_s} \phi_d(s) P(s) \frac{1}{2} \frac{F_{ref}}{n_s} \quad (5)$$

The results of such a spectral simulation are shown in figure 5, modelled for a fractionality of 0.65 and a third order sigma delta with feedback coefficients a and b of -9.345 and -8.459 respectively, obtained by optimisation of the maximum spur level in the output spectrum. On the same plot for comparison are ETSI DCS1800 masks for RMS phase noise in dBc/Hz (lower mask) and spurious in dBc. MASH modulator structures, whilst found to give much reduced noise very close to the carrier were found to not fit the masks as easily at critical points further out as the third order. It can be seen that close to the carrier the specification is more relaxed.

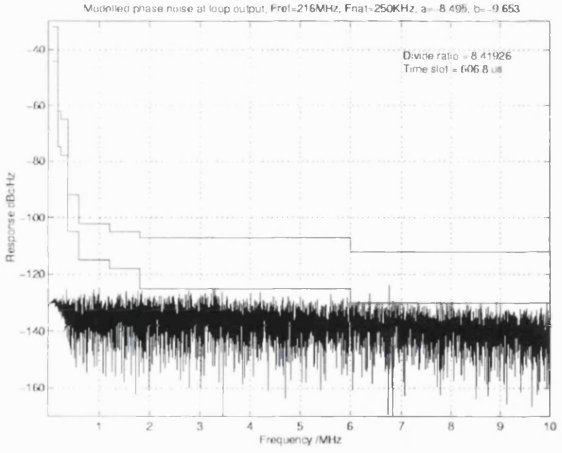


Fig 5. Simulated synthesiser phase noise spectral density, Fref = 200 MHz, 250Khz Natural frequency.

Fig 6. Shows the modelled transient performance of the third order type 2 loop with 250kHz natural frequency making a 15MHz frequency hop. Acquisition is to within 1kHz and $2x10^{-3}$ radians referred to the VCO within 10µs.

III. PROTOTYPE RESULTS

A discrete prototype has been fabricated on copper clad board using an ECL dual modulus divider and PFD, and a CMOS FPGA with an 8Mbit flash EPROM. Output frequency range was 1248 to 1404MHz and a 250KHz loop natural frequency was used.

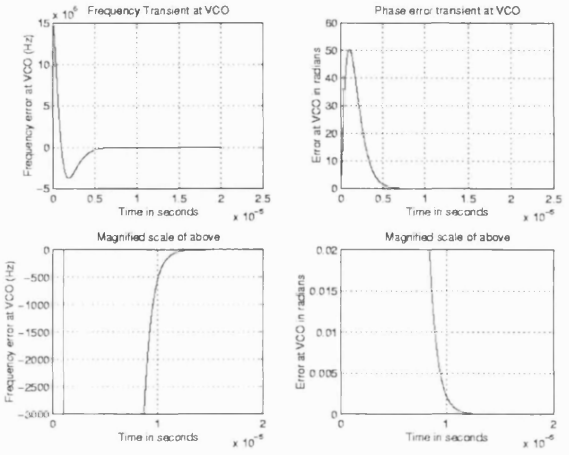


Fig 6. Simulated loop dynamics for a 15 MHz frequency hop.

The FPGA was capable of producing 250MHz data streams but reference frequency in this case was constrained to 156MHz by the 1.45GHz upper frequency limit of the ECL divider. Implementation in a silicon germanium HBT process would enable output frequencies well over 10GHz and correspondingly higher reference frequencies.

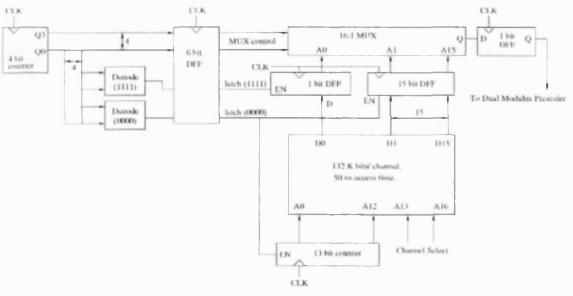


Fig 7. logic arrangement between memory and DMD.

Fig. 7 shows a the logic arrangement for serialising the memory data and Fig 8. Shows a representative output spectrum for a fractional division ratio of 8.58. SFDR was better than 60dBc/Hz across the band. Higher than predicted spur levels were attributed to non-linearities in the loop components and were particularly dependent on the drive levels of the various stages this was confirmed when the addition of a bandpass filter between the VCO output and the divider input was found to significantly reduce unwanted products at the output. Fig 10. Shows a measurement of RMS phase

noise captured from the phase noise measurement facility of an HP8560E spectrum analyser. The lump in the response at 1MHz was found to be due to a low speed op amp used in the active loop filter. Modulator noise cannot be seen due to the noise from the VCO, op amp and phase detector.

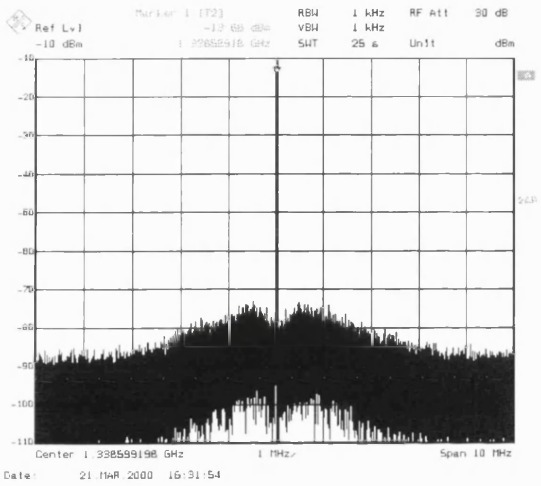


Fig 8. Prototype output spectrum with N=8.58.

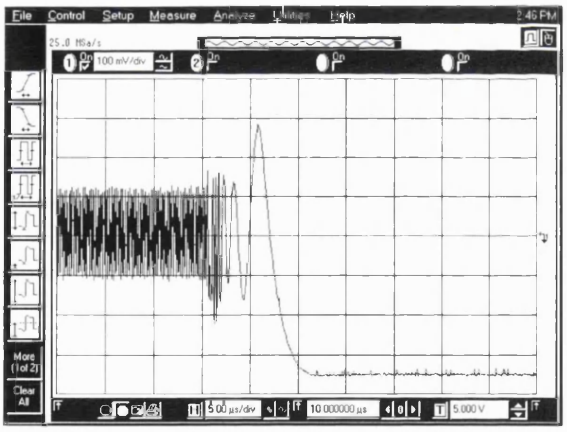


Fig 9. Measurement of 15MHz frequency hop to within 10° by mixing with synchronised signal generator.

Fig. 9 shows a 15MHz frequency hop in 10 µs, measured by mixing the loop output with a synchronised signal generator of identical frequency and phase. The mixer output is fed to the oscilloscope which is triggered from the channel change signal.

IV. Conclusion

It has been shown that by implementing the control of a fractional-N sigma delta synthesiser in memory it is possible to achieve significantly higher reference frequency operation for a given technology. This has

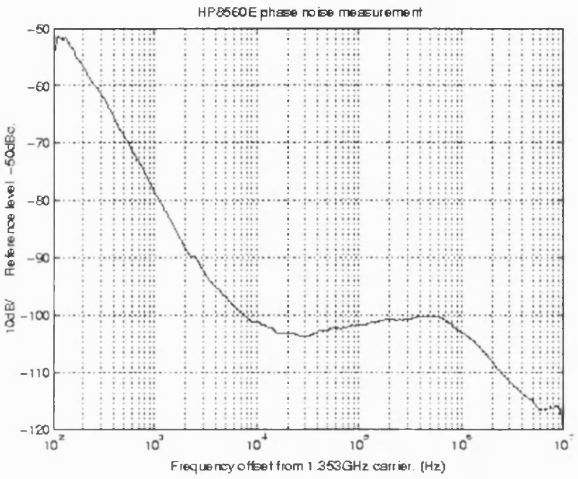


Fig 10. HP8560E RMS phase noise measurement of prototype.

the added advantages of allowing selective optimisation of modulator feedback across a band and reduced periodicity due to double precision simulation resolution. Implementing such a synthesiser in a silicon germanium process would allow significant cost reductions in many applications and would allow reference frequencies of several hundred MHz, allowing very high performance noise spreading. In addition the technique shows good performance in continuous mode use.

Acknowledgement

The authors wish to acknowledge the assistance and support of the Nokia Telecommunications Ltd. and the Electronics and physics research council.

References

[1] T.A.D. Riley, M. A. Copeland and T. Kwasnjeski, "Sigma-Delta modulation in fractional-N synthesis", *IEEE J. Solid-State Circuits,* vol 28 pp 553-559 May 1993.

[2] P.V.Brennan, R.Walkington, A.Borjak, & I. Thompson, "Very high-speed sigma-delta fractional-N synthesiser" IEE Electronics Letters, vol 36, issue 4, pp 298-300 February 2000.

[3] R. T. Baird, T. S. Fez, "Stability Analysis of High -Order Delta-Sigma Modulation for ADC's.*IEEE transactions on Circuits and Systems-2* vol 41 pp 59-62, January 1994.

[4] P. V. Brennan, *Phase-locked loops, principles and practice*, Macmillan, 1996.