Strategies for the Analysis and Design of a Low Noise, Frequency Agile Synthesiser

by

Ian Thompson

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

Department of Electronic & Electrical Engineering
University College London

May 2003

ABSTRACT

Frequency synthesisers have become the heart of many modern communications systems as they offer a repeatable means of producing a phase coherent signal source of almost arbitrary frequency and precision. In effect, they can be considered as very high-Q tracking filters capable of being digitally programmed to any integer or non-integer multiple of a reference frequency, with a limited amount of extraneous noise. Their relative simplicity and extraordinary benefit to communications systems has led to their success.

This thesis explores new strategies aimed at advancing our understanding of the noise and agility performance limits of modern phase locked loop frequency synthesisers and, in turn, allowing their better optimisation. A principal aim of this work is the development of a set of comprehensive models with which to analyse the noise performance of the common synthesiser elements and the system as a whole. A complete rigorous mathematical noise model of a sampled closed-loop phase locked loop synthesiser is developed and compared with measured results from a working prototype synthesiser. To complement this work, a detailed study of the mathematics of loop filter design is offered which does not compromise the loop bandwidth. Further work is presented to determine the fundamental noise limitations in such systems and to optimise the design rationale of synthesisers taking into account sampling effects, which are highly significant in high performance applications. Finally, a novel architecture offering the potential for low noise agile frequency synthesis, "the hybrid PLL/DDS synthesiser", is proposed and assessed both theoretically and experimentally.

.....to Melanie.

ACKNOWLEDGEMENTS

Throughout the course of this work there have been many difficult and challenging tasks, none more so than properly acknowledging the contributions of those people who helped in this work. Unfortunately it is not possible to acknowledge everyone, however their contribution certainly is not forgotten.

I would like to thank my parents, Catherine and Vernon Thompson for always being there, my son James for being so tolerant and my wife Melanie, for her immeasurable support and encouragement. Never once has she reminded me of my chores around the house, instead she has quietly assumed the extra workload and always been there when I needed her. Melanie, all I can say is thank you for your support, which has never wavered for a moment and has made this possible for me.

Being a natural worrier and believing myself to be a perfectionist, I must have appeared as a constant threat to the sanity of my supervisor, Dr. Paul Brennan, whom I wish to especially thank for his eternal patience. Through his skill and dedication, he has managed to guide me through my studies with apparent ease for which I shall always remain grateful. Thank you Paul.

I would like to acknowledge Nigel King and Charles Page who introduced me to the art of frequency synthesis especially fractional-n, turning the subject into a life-long fascination for me. Dr. Mahmoud Zadeh who initiated the Ph.D. studies through his skilful negotiation with Nokia. Lynne Lewis who diligently sourced all the publications I requested, no matter how obscure, which have proved invaluable to me throughout the course of these studies.

A special debt of gratitude is extended to Maggie Findlay for her help in keeping me sane during the worst moments.

To express my thoughts succinctly has proved to be one of the greatest challenges throughout my Ph.D. studies and leaves me permanently indebted to Prof. A. Seeds, (UCL), Dr. P. Brennan, (UCL), Dr. T. Busby, (Managing Director, Power Wave UK), Mr. M. Ward, (Analog Devices) and my father for the time they spent reviewing and commenting on extracts from this document. This significantly increased my confidence in submitting this document.

Finally I would like to remember my Grandfather, Stanley Wood-Higgs, who conspired with Melanie one memorable weekend in February 1988, to kick me back in to further education when I most needed it.

"... life is around the wrong way".

Stanley E. Wood-Higgs, OBE, FCIS, FAAI, FSAE, FBIM,

Freeman of the City of London.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PRINCIPAL SYMBOLS

SYMBOL	DEFINITION	UNIT
$\Delta\phi$	Phase Error.	rad
$\Delta\phi_{in}$	Input phase jitter.	rad
$\Delta\phi_{pk\text{-}pk}(t)$	Phase jitter caused by propagation delay in PFD.	rad
$\Delta\phi_{RMS}(t)$	RMS value of phase jitter caused by PFD propagation delay.	rad
$\Delta\phi(\omega_m)$	Phase fluctuations at a frequency offset.	rad
$\Delta\varphi$	PFD input phase difference.	rad
Δf_{peak}	Peak frequency deviation.	Hz
Δf_{RMS}	RMS frequency deviation.	Hz
ΔV_{RMS}	RMS noise at logic switching point.	V
Δt	Timing jitter.	s
Φ	Phase.	Deg
α, β, λ	Segments of binary word fed into sine look up table.	-
Δt	Timing jitter.	s
β	Modulation index.	-
θ_d	Peak phase deviation.	rad
δ	Dielectric loss.	
ε	Average |DNL| of the DAC.	-
ϕ	Generic phase value.	Deg
ϕ_{jitter}	Phase jitter.	rad
$\phi^2{}_{jitter}$	RMS phase jitter.	rad^2
σ	Standard deviation.	-
$\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7$	Generic loop filter time constants.	s
τ_{PD}	Phase detector propagation delay.	s
τ_{cp}	Charge pump on period.	s
ω	Angular frequency.	rad/s

ω_c	Cut-off frequency.	rad/s
ω_{in}	Angular input frequency.	rad/s
ω_{LO}	Angular Local Oscillator frequency.	rad/s
ω_m	Angular modulating frequency offset.	rad/s
ω_n	Loop natural frequency.	rad/s
ω_o	Angular output frequency.	rad/s
ω_{RF}	Angular Radio Frequency.	rad/s
ω_s	Angular Sampling frequency.	rad/s
A	-Width of digital word in DDS accumulator.	-
	-Digital divider down counter word.	-
	-Signal amplitude.	V
$A(t)$	Amplitude modulated component.	V
$Atten$	Attenuation.	dB
a	Generic phase term used in trigonometric identity.	rad
B	-Bandwidth.	Hz
	-Digital divider down counter word.	-
b	Generic phase term used in trigonometric identity.	rad
$C(s)$	Generic Laplace output signal.	-
C_1, C_2, C_3	Loop filter capacitor values.	F
D	Width of digital input word applied to DAC.	-
Div_{fl}	First Divider flicker noise breakpoint frequency.	Hz
$dv(t)/dt$	Slope of waveform.	V/s
$E(s)$	Error signal before sampling, (s-domain).	-
$E^*(s)$	Sampled error signal, (s-domain).	-
F	Noise factor.	-
$F(s)$	Loop filter Laplace transfer function.	-
f	Fourier frequency (sideband, offset, baseband).	Hz
f_{Clk}	Clock signal frequency supplied to the DDS and mixer.	Hz
f_c	-Corner frequency.	Hz
	-Cut-off frequency.	Hz
f_d	Frequency deviation.	Hz
f_{DDS}	DDS output frequency.	Hz
f_{LO}	Local Oscillator frequency.	Hz

f_i	Input frequency.	Hz
f_m	Modulating frequency.	Hz
$f_m^{\ 0}$,	Slope of spectral density: White phase.	
$f_m^{\ -1}$,	Flicker phase.	
$f_m^{\ -2}$,	Random walk phase (white FM).	
$f_m^{\ -3}$	Flicker FM.	
f_o	Output frequency.	Hz
f_{offset}	Offset frequency from carrier.	Hz
f_{RF}	Radio Frequency.	Hz
f_s	Sampling frequency.	Hz
$G(s)$	Open loop forward gain of PLL.	Rad/s
$GH(s)$	Loop gain of PLL.	Rad/s
h	Planck's constant, (6·626×10^{-34}).	J
$H(s)$	Feedback gain.	Rad/s
I_{Diode}	Diode current.	A
I_{Sat}	Diode Saturation current.	A
i	Current.	A
i_{cp}	Charge pump current into loop filter.	A
i_L	Leakage current profile.	A
$i_n(s)$	Laplace representation of current source into node n.	A
$i_{out}(s)$	Laplace representation of current out of loop filter.	A
$J_i(\beta)$	Bessel function of the first kind, order i and argument β.	-
K	Collective gain of all PLL frequency independent constants.	mA/V
K_{VCO}	VCO gain.	Hz/V
K_ϕ	Phase detector gain.	A/rad
k	Boltzmann's constant, (1·38×10^{-23}).	J/K
$\mathscr{L}(f_m)$	Single side band phase noise at offset frequency f_m.	dBc/Hz
$\mathscr{L}(\omega_m)$	Single sideband signal to carrier ratio at offset frequency ω_m.	dBc/Hz
L	Binary word length.	-
LBW	Loop Band Width.	Hz
l	LSB size in to DAC.	-
M	The binary equivalent of the DDS input word.	-
m	Multiplier of radio frequency.	-

N	Generic n-divider division value.	-
$N_{Ndiv}(s)$	N divider noise.	dBc/Hz
$N_{PFD}(s)$	Phase Frequency Detector noise.	dBc/Hz
$N_{Rdiv}(s)$	Reference divider noise.	dBc/Hz
$N_{Ref}(s)$	Reference source noise.	dBc/Hz
N_{v1}, N_{v2}, N_{v3}, N_{v4}	Divider noise voltages.	V
$N_{VCO}(s)$	VCO noise.	dBc/Hz
n	-Multiplier of local oscillator frequency.	-
	-Linear value of required loop filter attenuation.	-
P	Dual modulus pre-scaler division value.	-
P_{in}	Input power to active device.	W
P_{out}	Power out.	W
P_{sav}	Average power to active device.	W
$Plateau_{Div}$	Divider far out noise floor.	dBc/Hz
$Plateau_{Ref}$	Reference Oscillator far out noise floor.	dBc/Hz
$Plateau_{VCO}$	VCO far out noise floor.	dBc/Hz
PM	Phase Margin.	Deg
Q	-Quality factor.	-
	-Capacitor charge.	C
Q_{load}	Loaded Q of the tuned circuit.	-
Q_{unl}	Unloaded Q of the tuned circuit.	-
q	Charge on an electron, $(1{\cdot}6\times10^{-19})$.	C
R	Generic R-divider value.	-
R'	Resistor resistance value.	Ω
R_2, R_3	Loop filter resistor values.	Ω
Ref_{f1}	Reference oscillator flicker noise breakpoint frequency.	Hz
Ref_{f2}	Reference oscillator half resonator bandwidth frequency.	Hz
RMS	Root Mean-Square.	-
R_v	Equivalent resistance of the varactor diode.	Ω
S	Width of digital word into sine look up table.	-
$S^*_{Flicker}$	Sampled spectrum of flicker noise.	rad^2/Hz
$S_{Flicker}$	Spectrum of flicker noise.	rad^2/Hz
$S_{\Delta\phi}(f_m)$	Spectral density of phase fluctuations.	rad^2/Hz

$S\phi_{in}(f_m)$	Input double-sided spectral density of phase fluctuations.	rad^2/Hz
$S\phi_{out}(f_m)$	Output double-sided spectral density of phase fluctuations.	rad^2/Hz
s	Laplace operator.	-
T	Absolute temperature.	K
T_{Clk}	Clocking Period.	s
T_{jitter}	Time jitter.	s
T_{Ref}	Reference cycle period.	s
T_S	Sampling cycle period into phase frequency detector.	s
t	Time.	s
t_{jitter}	RMS DAC aperture jitter.	s
$\hat{V}$	Peak amplitude of voltage swing.	V
$\overline{V^2}$	Mean square voltage.	V^2
VCO_{f1}	VCO half resonator bandwidth frequency.	Hz
VCO_{f2}	VCO flicker noise breakpoint frequency.	Hz
V_{FS}	Full-scale DAC output voltage.	V
V_{in}	-Input voltage applied to loop filter.	V
	-Voltage across a diode.	V
V_{LO}	Local oscillator signal voltage.	V
$V_n(s)$	Laplace representation of voltage at node *n*.	V
V_{RF}	RF signal voltage (applied to a mixer input).	V
$V_{thermal}$	-Noise in DAC LSB.	V
	-Loop filter thermal noise applied to VCO input.	V
V_{tune}	voltage applied to VCO input by PLL from loop filter output.	V
$v_n(t)$	Generic voltage at node n at time *t*.	V
$v(s)$	Voltage signal, Laplace domain.	V
$v(t)$	Voltage Signal, time domain.	V
W_e	Reactive energy within the resonator.	W

LIST OF ABBREVIATIONS

SYMBOL	DEFINITION	UNIT
16QAM	16 Quadrature Amplitude Modulation.	-
3G	Third Generation, (WCDMA).	-
3GPP	3rd Generation Partnership Project.	-
AM	Amplitude Modulation.	-
API	Automatic Phase Interpolation.	-
BCD	Binary Coded Decimal.	-
CMOS	Complementary Metal Oxide Semiconductor.	-
CP	Charge Pump.	-
DAC	Digital to Analogue Converter.	-
DC	Direct Current.	-
DDFS	Direct Digital Frequency Synthesiser.	-
DDS	Direct Digital Synthesis.	-
DNL	Differential Non-Linearity.	-
DUT	Device Under Test.	-
ENOB	Effective Number Of Bits.	-
EVM	Error Vector Magnitude.	%
FET	Field Effect Transistor.	-
FOM	Figure Of Merit.	dBc/Hz^2
FSD	Full Scale Deviation.	-
GSHF	Generalised Sample-Hold Function.	-
GSPS	Giga Samples Per Second.	Hz
GSM	Global System for Mobile communications, (formally known as Groupe Speciale Mobile).	-
HSDPA	High Speed Data Packet Access.	-
IC	Integrated Circuit.	-
IEE	Institute of Electronic Engineers.	-
IF	Intermediate Frequency.	Hz

INL	Integral Non-Linearity.	-
InP	Indium Phosphate.	-
IQ	In-phase, Quadrature-phase.	-
LO	Local Oscillator.	-
LSB	Least Significant Bit.	-
MSB	Most Significant Bit.	-
NCO	Numerically Controlled Oscillator.	-
PCB	Printed Circuit Board.	-
PFD	Phase Frequency Detector.	-
PDF	Probability Density Function.	-
PLL	Phase Locked Loop.	-
PM	Phase Modulated.	-
PSD	Power Spectral Density, (single side-band phase noise).	dBc/Hz
PWM	Pulse Width Modulated.	-
ROM	Read Only Memory.	-
RF	Radio Frequency.	Hz
RMS	Root Mean Square.	-
SD	Standard Deviation.	-
SFDR	Spurious Free Dynamic Range.	-
SNR	Signal to Noise Ratio.	-
SSB	Single Side-Band.	-
TDMA	Time Division Multiple Access.	-
VCO	Voltage Controlled Oscillator.	-
WCDMA	Wide-band Code Division Multiple Access.	-
ZOH	Zero Order Hold.	-

1 INTRODUCTION

1.1 Frequency Synthesis in Mobile Communications

Over the last decade, the extraordinarily rapid growth in mobile communications for an ever-expanding variety of different applications has resulted in undue stress on the principal bottleneck of radio spectrum. This precious commodity has recently led to vast sums of money being exchanged for very small allocations of radio space. Universal to all systems and standards competing for this radio space are specifications defining the allowable amount of frequency drift and spurious emissions that each system can impart on its neighbours. For a limited number of applications simple stable oscillators will suffice, however, almost all higher frequency applications use a frequency synthesiser to meet the system specifications, irrespective of whether the system requires phase coherency or not, [1]. As a result, frequency synthesisers have become an intrinsic part of all modern coherent communications systems. Their value lies in their ability to synthesise another frequency from a given reference source and then continue to maintain phase coherence with that reference source.

Throughout this decade, the evolution of synthesiser technology within mobile communications has been a cyclic process dependent upon the demands of the standard commanding the highest attention at any given time. Within mobile communications, the base station has always borne the stress of the tightest operational specifications, which has continually pushed their synthesised local oscillator sources to their operational limits. For second generation GSM base station applications using base-band hopping, simple synthesisers were used to meet the basic GSM blocking and far out noise specifications. With the advent of third generation GSM systems requiring frequency agility in addition to phase noise performance, the universally adopted solution was a compromise between phase noise and a mediocre settling time using two identical synthesisers operating in parallel, (the "ping-pong" synthesiser). The preferred solution remains a single synthesiser, which has yet to be satisfactorily fulfilled. Currently the evolution of mobile communications into 3G WCDMA systems has shifted the emphasis to power amplifier and power control performance, whilst re-using the well-established second-generation GSM local oscillator sources whose refined performance now exceeds WCDMA requirements. This downturn in synthesiser performance expectations is likely to be short lived with commercial pressure striving for a universal system capable of supporting multiple carriers with different standards across multiple bands, (the "Multi-Radio" concept) and the proposed introduction of different modulation formats in 3GPP WCDMA systems[1]. To simultaneously satisfy these combined applications requires synthesisers capable of meeting a composite of the most challenging aspects from each specification, for each application.

This thesis focuses on understanding and modelling the noise and spurious products within a combination of analogue phase-locked loop, (PLL) and direct digital synthesis, (DDS) units and how to manage these quantities in the design of one example of a frequency agile, low noise synthesiser.

1.2 Scope Of The Thesis

The objective of this work is to produce a synthesiser capable of meeting the exacting demands of a multi-radio application. To achieve this, the philosophy has been to thoroughly understand and evaluate the performance of each element used within the synthesiser and exploit the most promising techniques to arrive at the solution offered here. This necessarily requires that each element is pushed to the edge of its performance envelope, within the confines of current technology. The purpose of this investigation is to predict these limits through simulation, then to design a suitable synthesiser to verify these models against measurements from the proposed system.

[1] Version 5 of the 3GPP standard is expected to introduce HSDPA capability [2], whilst Version 6 is expected to introduce 16QAM modulation, [3].

1.3 Principal Contributions Of This Work

As a direct result of this work the following advances have been achieved.

1) Two original techniques have been developed (and patents filed):

   i The DAC at the output of a direct digital synthesiser is replaced by a VCO, (controlled by the DDS accumulator), giving a greater level of spectral purity and requires less DC power. Although this idea was not implemented within this work, its inception was a consequence of thoroughly understanding the DDS architecture, section 3.5.

   ii The speed-up technique developed to overcome the problems of cycle slipping proved particularly effective. Because no references to this technique were found, Nokia has applied for a patent to protect this idea.

2) In chapter 4, an original analysis, (valid for sampled systems), has been developed which is used to account for each uncorrelated noise source within a sampled phase locked loop.

3) In chapter 5 an original mathematical analysis is presented which accounts for the contribution of the PFD noise to in-band synthesiser phase noise and how it is related to device parameters. This work has been published in IEE Electronics Letters, [4], and IEE Proceedings, appendix A5.

4) In chapter 6, a novel and simple method is presented for type II, fourth-order loop filter design[2].

5) In chapter 7 an original hybrid synthesiser technique is introduced which has been incorporated in the latest Analog Devices DDS chip, the AD9854, [6]. This part has now been released for sale, prompting a good deal of commercial interest. This hybrid DDS based application was displayed on the front cover of the September 2002 edition of the Wireless Systems Design magazine, appendix A6.

[2] During the course of this investigation over a dozen different publications were found addressing this problem each with their own interpretation of third and fourth order loop filter design.

1.4 Thesis Outline

The principal contributions of this work are spread across the three chapters that are concerned with phase noise modelling and the verification of these noise models within this demanding application.

Chapter 2 provides a brief history of frequency synthesisers and how they have sub-divided into fractional-n and DDS synthesisers.

Chapter 3 considers the basic frequency synthesiser in its many forms with particular emphasis on the building blocks used within the integrated PLL and DDS solution considered in chapter 7.

Chapter 4 provides an in-depth mathematical analysis of the different phase noise mechanisms found within a closed-loop PLL system. Attention is drawn to the mathematics developed in this chapter, which deal with the modelling of different phase noise sources in the sampled loop. The complete noise transfer formulae constructed from the summation of each uncorrelated noise source within the sampled phase locked loop, give results that match practical measurement results. However, several precursory laboratory measurements soon demonstrated the inadequacies of many of the noise models commonly offered, especially when pushed to their limits. These findings expanded the work, requiring the development of more rigorous mathematical models to accurately predict the digital phase frequency detector noise performance within a synthesiser, when pushed to its extremes, as in this study.

Chapter 5 demonstrates the significance of the PFD noise and its impact on in-band[3] phase noise and accordingly pays special attention to deriving an accurate noise model, correctly reflecting the PFD noise behaviour in all synthesiser applications. Contained within this chapter is an innovative mathematical insight into how the PFD device noises are manifest in the overall PLL phase noise profile, which became the subject of both an IEE Electronics Letter and an IEE Proceedings paper.

[3] The term “in-band” will be used throughout this document to describe the performance inside the loop bandwidth.

To complement the sampled loop phase noise analysis, chapter 6 considers the design of third and fourth-order loop filters, to ensure the required phase margin is achieved. The fourth-order loop filter design equations described in chapter 6, are believed to be new and creative.

Building on this foundation work, a low noise, frequency agile synthesiser solution is presented in chapter 7, with measurement results to verify the phase noise predictions. This synthesiser uses a DDS based hybrid architecture, which was considered to be sufficiently original that Analog Devices used the concept as the model for a new generation, 1GSPS DDS based synthesiser chip, the AD9858 that they were developing. This part has now been released and has since become the subject of a cover feature in the Wireless Systems Design magazine, appendix A6, which describes the integration of this chip in this synthesiser architecture.

Chapter 8 concludes with a summary of the work to date, and how it could proceed to investigate those areas that fell outside the remit of this study.

As a consequence of this study and the innovation required to overcome some practical implementation difficulties, Nokia submitted two patent applications. The first of these two patents successfully passed the prosecution stage in October 2002.

2 THE HISTORY OF FREQUENCY SYNTHESIS

2.1 The Origins Of Frequency Synthesis

Perhaps the earliest mention of a sampled, phase locked oscillating system, was by McCrea in 1929, [7]. In this system, the phase of a precision pendulum, used as a reference source, was electrically monitored and compared with the angular position of a synchronous motor, (linked to some large electric generators), to provide a pulse width modulated error signal. This error signal adjusted the spring loading on the governor and thereby corrected the frequency of the generator by changing its load. The techniques used in this system are still applied to modern, charge pump driven, phase locked loops.

One of the earliest descriptions of a free running electronic oscillator, (the key controlled element within a synthesiser), can be found in a patent filed in 1916 by Alexanderson, [8]. Whilst it was only much later in 1932 that the first published account of frequency locking an oscillator was given by De Bellescize, [9], in which an oscillator is described as locked to the incoming radio signal. In 1935, in his paper on frequency control, Travis [10], summarised the need for frequency synthesis as:

> "some way of supplementing the accuracy of manual tuning by more or less automatic means. Some method of bringing the signal carrier precisely to the centre of its intermediate-frequency band and anchoring it there in spite of small maladjustments of tuning or others that subsequently arise from thermal changes and the like".

In these early applications the emphasis was on frequency alignment using frequency discriminators whose operation was well understood, [11]. It was only later, in 1943, when the concepts of television were being developed that phase and frequency control were proposed by Wendt, [12], and Finden, [13], using systems which are familiar today, Figure 2.1. These advances were necessary to ensure the phase coherence essential for proper television display reproduction, [14].

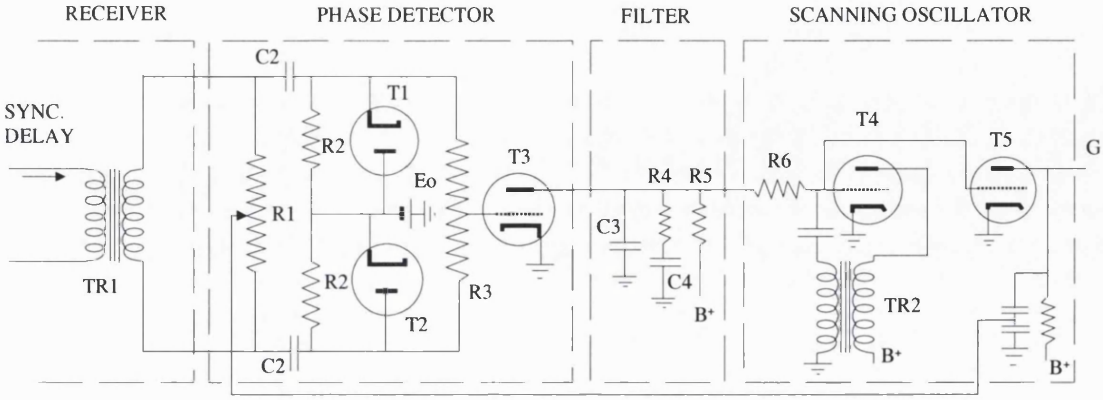


Figure 2.1 Wendt's proposal for a phase locked loop, [12, fig 5]

To synthesise one signal from another, non-linear circuit elements were commonly used, [15], however these were open-loop systems that suffer the disadvantages of noise degradation and distortion due to variations of the non-linear elements through ageing and thermal effects. A typical synthesiser would use a combination of multipliers, dividers and mixers to synthesise the required frequency, with each new frequency requiring a different configuration dependent upon the division and multiplication ratios available. Surprisingly, one of the earliest references to the use of digital dividers as the feedback element in a closed loop system is in a patent, "Frequency Multiplier and Frequency Waveform Generator" filed by Sepe *et al.* in 1968, [16][4]. Using digital dividers in the feedback path of a phase locked loop, (PLL), allowed frequency synthesisers to easily provide output frequencies, which were not derived by cascaded, multiply, divide and mix systems but instead were a simple multiplication of the PLL's sampling frequency.

It was around the 1960s that the concept of the digital phase accumulator became reality, splitting the evolution of frequency synthesisers into two paths; i.e. direct digital synthesis, (DDS), and fractional-n techniques, each using the digital accumulator as their core. Subtly, there is a distinct difference in the purpose of the digital accumulator

[4] A patent should not be granted if *prior art* can be established or the concept is deemed obvious to someone "skilled in the art". Patents have been adopted as the milestones in the history of frequency synthesis because their very existence suggests no previous knowledge of the technique was known, thus classifying them as new at that time.

between fractional-n and DDS applications. Within fractional-n, the carry-out (overflow) signal accounts for a complete cycle overflow when driving the integer n-dividers. Conversely for DDS, the digital word representing the current phase value held in the digital accumulator, is used as the basis of the DDS output sinusoid. With the evolution of the digital accumulator came the ability to accurately track phase within digital dividers, which moved analogue synthesiser evolution into fractional-n development.

2.2 Fractional-N Synthesisers

Fractional-n frequency synthesis is a frequency technique that seeks to increase the phase frequency detector, (PFD), sampling frequency beyond the minimum PLL resolution achievable using integer-n dividers. This increase in PFD sampling frequency offers the freedom of larger loop bandwidths than would otherwise be possible with integer-n loops requiring small raster values. Furthermore, the increase in PFD sampling frequency reduces the level of in-band phase noise, facilitating increased loop bandwidths to improve the synthesiser's overall noise performance. To achieve this objective, the integer-n dividers are digitally modulated so that their mean division value equals the desired integer plus fractional division value required to raise the PFD sampling frequency relative to the desired output frequency.

The evolution of fractional-n techniques is commercially very sensitive and is perhaps best traced through some key patents released over the last three decades, Figure 2.2. Although nowadays the trend has been towards using all digital, sigma delta noise shaping techniques, arguably the roots of fractional-n frequency synthesis stem from the Digiphase® principle described by Gillete, [17, 18]. Within this technique a second register tracks the required fractional offset and adds back the missing cycle by incrementing the n-division value for one reference period. Complications arise due to the addition of this additional cycle causing unwanted modulation sidebands to appear, therefore a compensation digital to analogue converter, (DAC), was required to minimise the DC ripple caused on the VCO tune line. This technique is limited because it requires accurate open-loop DAC current balancing to achieve substantial improvements of in-band phase noise performance, [19]. With the introduction of DDS in 1971, [20], the concept of using a digital accumulator to control phase was adopted, however using a single digital accumulator in a synthesiser still requires an analogue to digital converter to correct for the strong n-divider modulation products thereby generated. The use of a DAC to correct for the fractional-n modulation is often referred to as automatic phase interpolation, (API). In 1978[5] the landmark patent by King, [22], was filed whose significance lies in the cascade of two digital accumulators. This technique, sometimes referred to as "Kingphase", introduced two important features; 1) to provide an integrated phase correction signal into the DAC, thereby reducing the stress on DAC performance, and more importantly, 2) a digital delay of the second accumulator carry signal which is weighted and summed with the carry out of the first digital accumulator before being applied as the n-divider fractional modulation. Concept 2) was expanded by Wells in 1983, [23], to arrive at a cascade of digital accumulators whose respective carry-out signals were delayed, weighted and summed in accordance with Pascal's triangle before being applied as the fractional n-divider modulation, giving the first all digital noise shaping sequence and making the DAC redundant[6]. Jackson's patent, [24], released in 1986, developed the multiple accumulators into the first sigma delta architecture used within a PLL and included the mathematics to demonstrate the noise shaping properties. In 1990, two almost identical patents by Miller and Gaskell *et al.*, [25, 26], were simultaneously released detailing full sigma delta architectures spawning the current multitude of patents each offering their own improved version of sigma delta noise shaping algorithm. One of the latest patents offered by Brennan *et al.*, [27], offers an almost perfect high speed sigma delta system using sigma delta coefficients which are calculated off-line using floating point arithmetic developed by Walkington, [28], to a very high degree of accuracy.

[5] All references to patent dates are in terms of their original filing date.

[6] Worthy of mention at this point is that in 1977 Ritchie proposed a higher order sigma delta loop filter using a cascade of integrators, [21].

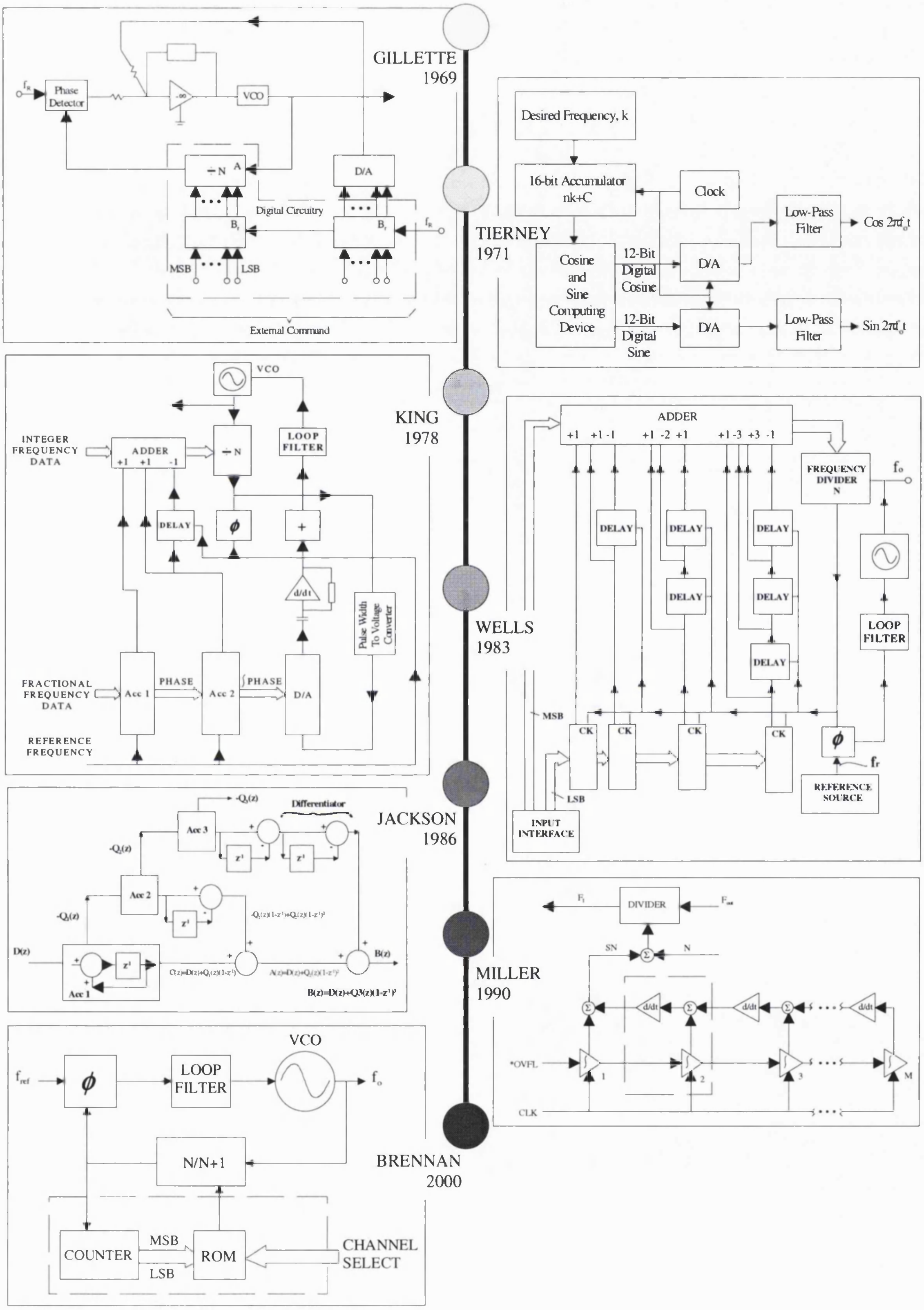


Figure 2.2 Summary of evolution of fractional-n through relevant published diagrams

Although sigma delta modulation techniques have been universally adopted as the answer to fractional-n frequency synthesis, they have their origins in work carried out in the 1940's and 1950's primarily for voice transmission in telephony applications, [29].

One of the earliest accounts of digital noise shaping can be traced back to the delta modulator described in 1952 by de Jager [30]. The better-known patent by Cutler *et al.* filed in 1954, [31], quickly superseded his work and describes the sigma delta modulator concept, Figure 2.3. The fundamental difference between sigma delta and a delta sigma, which has led to much confusion, is in the placement of the filtering either in the forward or in the feedback path of the system and the subsequent introduction of the two terms. Later in 1962 Inose *et al.*, [32], extended the concept by introducing the delta sigma modulator and presenting the first detailed mathematical analysis of this noise-shaping technique. Starting in 1974, Candy, [33], released a succession of articles that introduced the term "sigma delta" and its application in analogue to digital converters, which Ritchie extended in 1977 to give higher-order sigma delta filtering, by using a cascade of integrators, [21].

As described in the aforementioned text, sigma delta techniques were adopted in frequency synthesis as a repeatable means of increasing the PFD sampling frequency, whilst still confined to using integer-n dividers in the feedback path of a PLL. Digitally controlled sigma delta modulation of the integer-n dividers provides a reliable mean fractional division value, whilst minimising the spurious products thus produced.

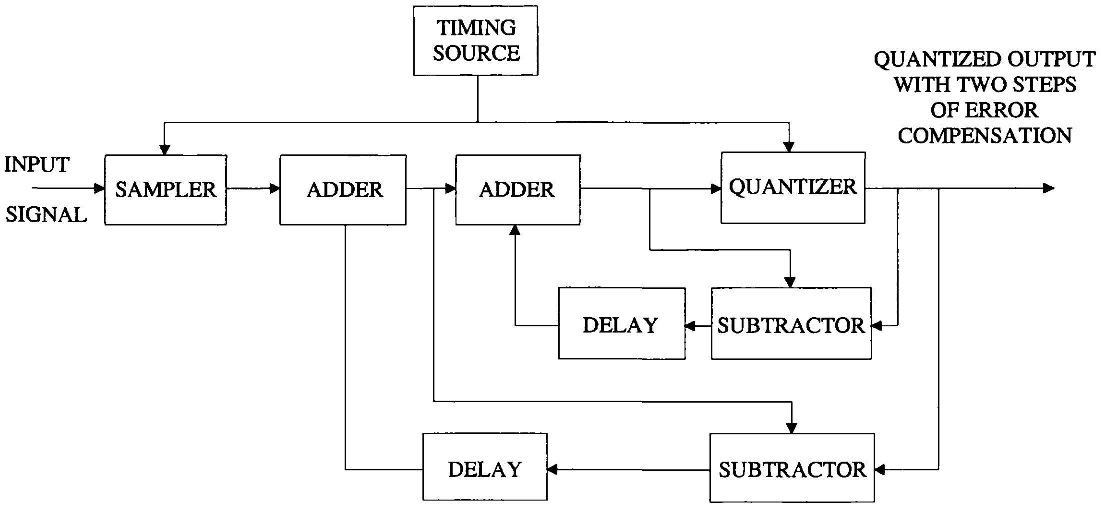


Figure 2.3 Cutler, [31], implementation of a sigma delta modulator

To date, not one commercially available single-chip fractional-n part has been released which demonstrates the expected phase noise improvement theoretically available whilst still offering the spurious performance of integer-n synthesisers. Unfortunately, neither fractional nor integer-n based synthesisers are likely to offer the in-band phase noise performance required to increase the loop bandwidths to the necessary offsets essential to meet the demanding single synthesiser GSM requirements, for lock time.

It is worth noting that there have been several very successful single-loop fractional-n, laboratory grade signal generators, available within the test equipment market, which also deploys analogue noise suppression techniques to achieve their superior phase noise performance [34, 35].

In contrast to fractional-n synthesiser evolution has been the parallel development of direct digital synthesis, which also relies on the digital accumulator to retain the synthesiser's phase at any given moment in time.

2.3 The Direct Digital Synthesiser

Direct digital synthesis, (DDS) is an all-digital frequency synthesis scheme that employs a single digital accumulator, which is clocked by the input clock signal to maintain the current sine wave phase value, Figure 2.4. The roll over rate of this digital accumulator is set by the digital input word, which is constantly being added to the result from the previous summation, processed by the digital accumulator. The output of the digital accumulator is non-linearly mapped onto the equivalent sine wave amplitude value before being converted to an analogue signal value by the output digital to analogue converter.

One of the earliest accounts of direct digital frequency synthesis, (DDS), can be traced to a patent filed by Cliff in 1966, for an all-digital synthesiser, [36]. This synthesiser was capable of synthesising a pulse sequence from a master clock source using a digital accumulator and an array of digital divider circuits. Later in 1970 Webb, [37], filed a patent for a "Digital Signal Generator Synthesizer" in which he proposed using a digital accumulator with a "Sum Register" and memory to convert $\omega_o T_{Clk}$ to $\cos(\omega_o T_{Clk})$ to feed a DAC giving a low frequency function generator. Co-incidentally, an influential paper was published in 1971 by Tierney *et al.*, [20], offering a thorough analysis of a complete DDS architecture using a digital accumulator, sine look up table and DAC, Figure 2.4. It is generally accepted that this landmark paper, almost undoubtedly written in isolation[7] of Webb's patent because of the time scales and processing of each of these two publications, defines the DDS architecture that has been universally adopted today.

Several notable modifications to the DDS core aimed at raising the maximum operational speed have addressed the compression algorithm used in the sine look up table. A technique first proposed by Hutchinson, [38], was further modified by Sunderland *et al.*, [39], to give what is perhaps one of the most popular and simple look up table compression schemes. An interesting alternative to the look up table, which has been adopted in DDS applications, is the Cordic algorithm first proposed in 1960 by Volder, [40], which generates the sine amplitude data in real-time. To overcome the infamous DDS spurious problem, Wheatley *et al.*, [41], are credited with using dithering as a means of reducing the level of some DDS spurious products.

Many alternative systems have since been offered to reduce spurious products, for example in 1991, Wilson, *et al.*, [42] offered an alternative DDS based synthesiser that employed two DACs at the DDS output suitably configured such that pseudo random noise added to the main DAC is cancelled by the second DAC. This technique helps reduce the level of the DDS spurious products and minimise the rise in the noise floor.

There are essentially three principal bottlenecks that limit the speed of a DDS unit; the digital accumulator, the non-linear phase to amplitude conversion and the maximum DAC operating speed that gives acceptable performance. DDS evolution over the last decade has been limited by the available technology and the methodology used in the design of these three elements, with the slowest element defining the maximum operating speed of the DDS unit. In 1991, Saul [43, 44], is recognised as being responsible for the design of the SP2002 device whose maximum operational speed of 2GHz driving two 8-bit output DACs up to 500MHz, remains an outstanding achievement. The SP2002 is still a viable product available today. Other DDS units, at that time, were limited to speeds of 20MHz.

[7] Patent time scales require that after 18 months of filing a patent a "Publication" is released irrespective of whether there are any outstanding questions concerning its legal value. Only after Granting the patent is the Prosecution process concluded and the patent becomes valid.

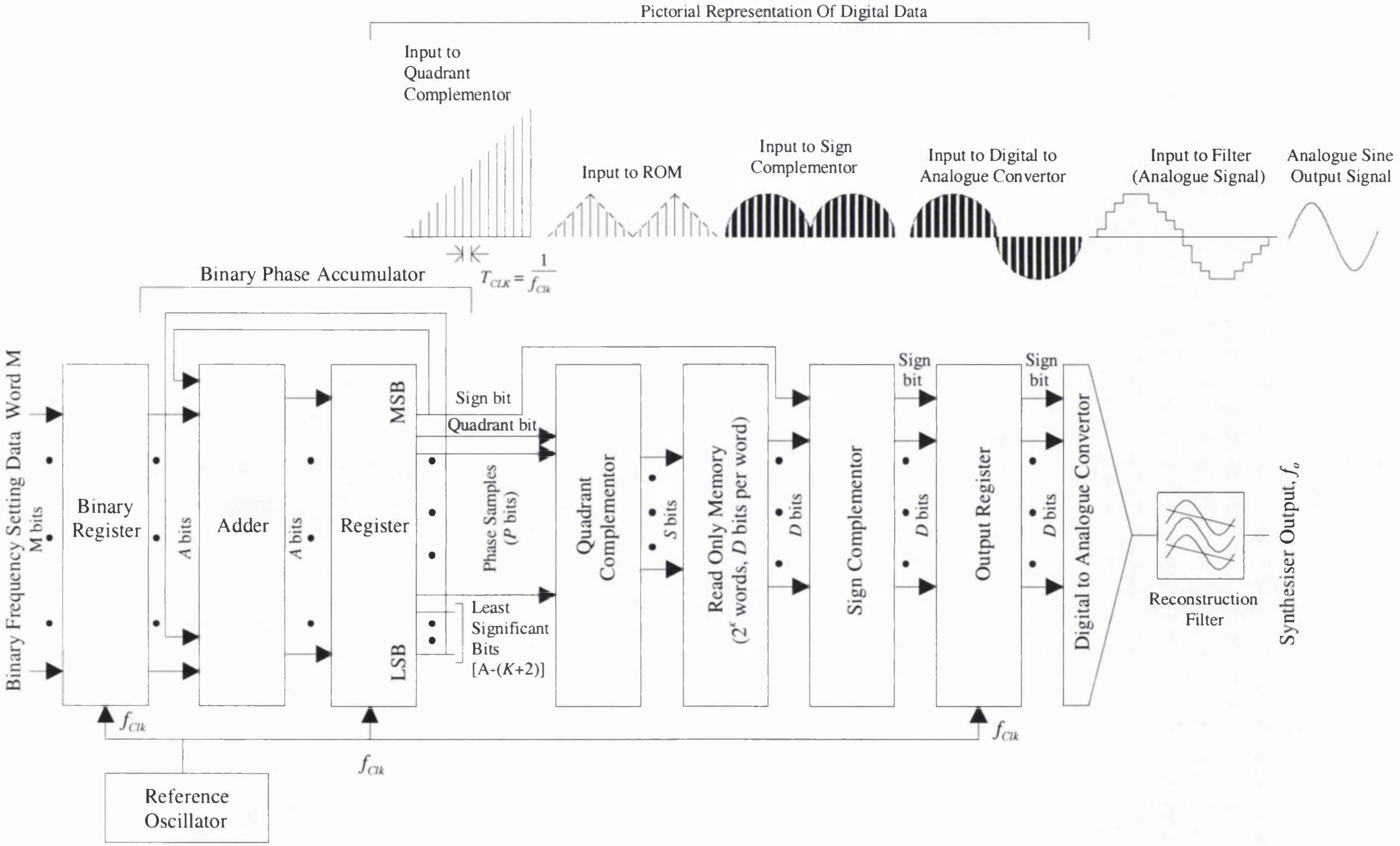

Pictorial Representation Of Digital Data
Input to Quadrant Complementor
Input to ROM
Input to Sign Complementor
Input to Digital to Analogue Convertor
Input to Filter (Analogue Signal)
Analogue Sine Output Signal
$T_{CLK} = \frac{1}{f_{Clk}}$
Binary Phase Accumulator
Binary Frequency Setting Data Word M
M bits
Binary Register
A bits
Adder
A bits
Register
MSB
LSB
Sign bit
Quadrant bit
Phase Samples (P bits)
Least Significant Bits [A-(K+2)]
Quadrant Complementor
S bits
Read Only Memory (2^K words, D bits per word)
D bits
Sign Complementor
Sign bit
D bits
Output Register
Sign bit
D bits
Digital to Analogue Convertor
Reconstruction Filter
Synthesiser Output, f_o
f_{Clk}
f_{Clk}
f_{Clk}
Reference Oscillator

Figure 2.4 Typical complete DDS + DAC architecture [45]

Over the last decade, the maximum operating speed of other commercial DDS units has crept up, with the latest unit, the AD9858, [6], offering 1GHz operation throughout, including an integrated 10-bit output DAC. Aside from these commercial DDS units other technologies such as Indium Phosphate, InP, have been applied using novel accumulator, look up table and DAC architectures, thus allowing the demonstration of DDS units up to a 9·2GHz operating speed, [46], with typically 40dB of SFDR. Clearly the evolution of DDS has accelerated in the last few years and now the phase noise, speed, bandwidth, and hence potential of DDS, make it a dominant factor in system planning.

The omnipresent DDS concept has become the topic of many interesting papers, patents and application notes which illustrate the flexibility and applicability of DDS in such diverse applications as radar, frequency scanning and modem signalling as well as in NCO cores used within mixed signal IC's. However, DDS fundamental limitations of spurious and operational speed have prevented it from replacing analogue frequency synthesisers, nevertheless, the ability of DDS to instantaneously change frequency digitally, with very high levels of precision and excellent noise performance, has ensured its place in synthesiser system planning. Hybrid synthesisers that use DDS are usually designed to exploit this frequency agility and resolution whilst simultaneously using the complete synthesiser architecture to suppress spurious products to a suitable level for the application.

3 DEFINING FREQUENCY SYNTHESISERS

3.1 Introduction

The purpose of this chapter is to consider different candidate architectures and their building blocks, which can be reconfigured in a variety of combinations to solve the conflicting specification requirements of phase noise, lock time and spurious performance. Each of the building blocks is presented to a sufficient level as a foundation for further development in subsequent chapters. The main reasons why neither fractional-n nor DDS can be used as the simple solution for meeting a low noise, frequency agile local oscillator specification are identified.

During the assessment stages of different synthesisers described in this chapter, the idea for a novel form of DDS synthesiser was proposed leading to a patent application by Nokia.

3.2 Overview of Frequency Synthesiser Configurations

Behind the commonly accepted phase locked loops, exist a whole range of different frequency synthesisers, which can broadly be sub-divided into brute force and non-brute force techniques, [47]. Their difference depends upon the techniques used to synthesise the signal, which range from cumbersome hardware intensive techniques to the very elegant hardware efficient direct form of frequency synthesiser. If greater operational performance is required, a variety of different techniques are often integrated to give a hybrid solution. Figure 3.1 categorises the known techniques for frequency synthesis. In general, most non-demanding synthesisers use only one of the simpler systems listed for reasons of convenience, cost and simplicity, whilst more demanding applications require a combination of techniques, each one designed to exploit a particular feature. The objective of this research work is to understand the performance of some of these synthesisers and develop a combination of the most promising, to meet the most demanding specifications, without resorting to an unduly complicated synthesiser.

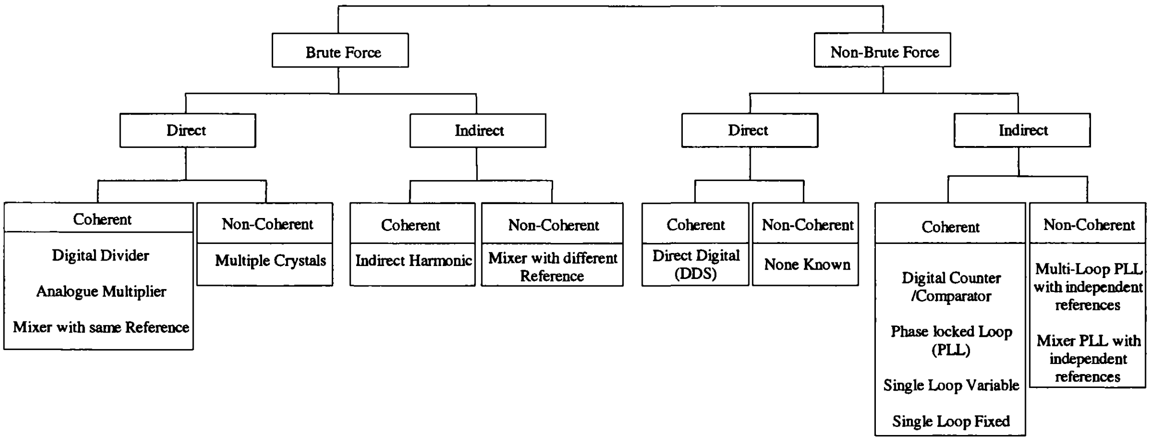


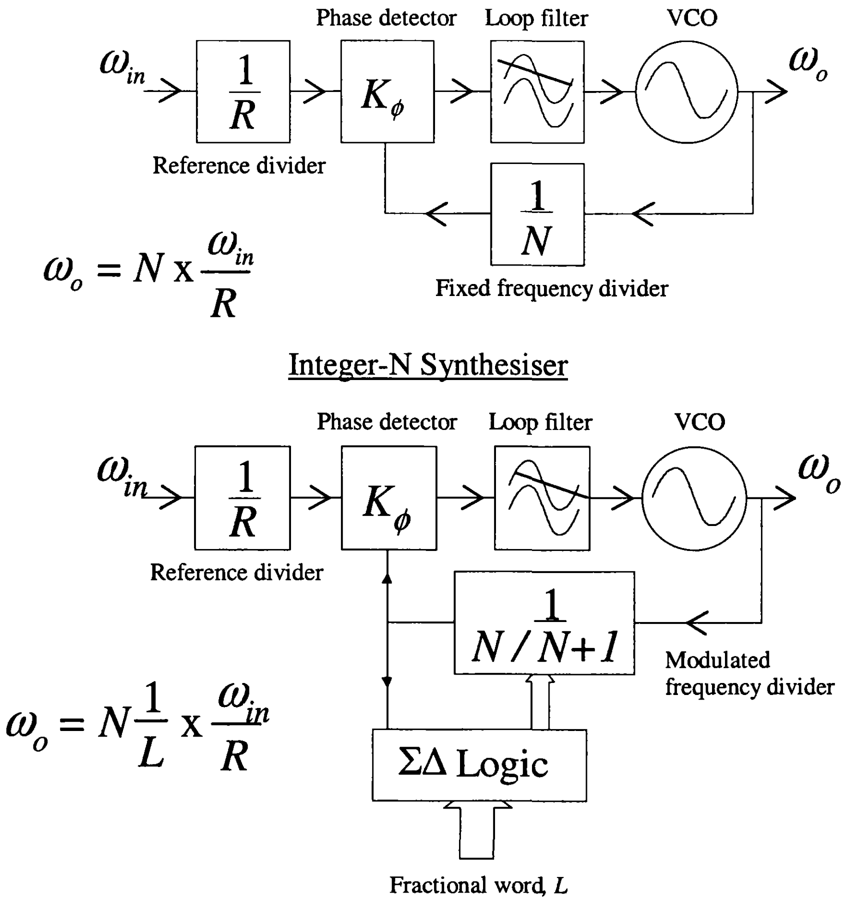
Figure 3.1 Classification of different synthesiser types, [47]

3.3 Mixed Signal Frequency Synthesisers

There exist two fundamental categories of single loop synthesiser; that of the integer-n and the fractional-n frequency synthesiser, Figure 3.2. Their respective evolution has been one of continued improvement of each element within the integer-n synthesiser structure, or advances in noise shaping techniques within fractional-n systems.

3.3.1 Integer-N Single Loop Synthesisers

Integer-n synthesisers are single loop synthesisers using a forward path comprised of phase detector followed by the loop filter, which controls the VCO. Phase lock is achieved by the n-divider feedback path that divides the VCO output signal to provide a comparison signal, at the same frequency as the reference signal for phase detection. Fractional-n synthesisers use the same architecture with the subtle difference that the feedback n-dividers are modulated to give a lower mean division value, Figure 3.2



Integer-N Synthesiser

Fractional-N Synthesiser

Figure 3.2 Integer-n and fractional-n single loop synthesisers compared

Because of their relative simplicity, integer-n synthesisers are the most prolific in modern commercial communications systems and are easily recognised as an integrated circuit solution containing the digital dividers and phase/frequency detector with a charge pump driven output. Over the last decade, the commercial viability of this type of synthesiser has resulted in considerable improvements in operational performance and an abundance of application notes.

3.3.2 Fractional-N Single Loop Frequency Synthesisers

Within a frequency synthesiser that uses a digital divider in the feedback path, the n-division value raises the level of in-band phase noise by raising the loop gain response to the reference input signal and phase detector noises. To overcome this problem the concept of reducing the feedback n-division value, and hence raising the sampling frequency found at the PFD input has led to the evolution of fractional-n synthesisers.

This has led most manufacturers to produce a fractional-n device, each with their own technique that allows integer dividers to be used in a pseudo fractional manner, whilst simultaneously maintaining the spurious performance expected of an integer-n synthesiser. Inevitably there is a penalty for modulating the integer n-dividers to give a mean fractional value, and this has resulted in a variety of patents being generated each aiming to reduce these spurious products, Figure 2.2.

The recent commercial interest in fractional-n frequency synthesisers has been based upon their perceived ability to optimise the performance of a frequency synthesiser by increasing the PFD sampling Frequency. With an increased PFD sampling frequency an expected decrease of in-band noise is anticipated allowing the loop bandwidth to increase. Hence the combination of improved in-band noise performance and increased loop bandwidth should theoretically make fractional-n frequency synthesisers suitable for meeting the specifications of a variety of otherwise impossible applications. However, it is worth considering the practical limitations of fractional-n frequency synthesis.

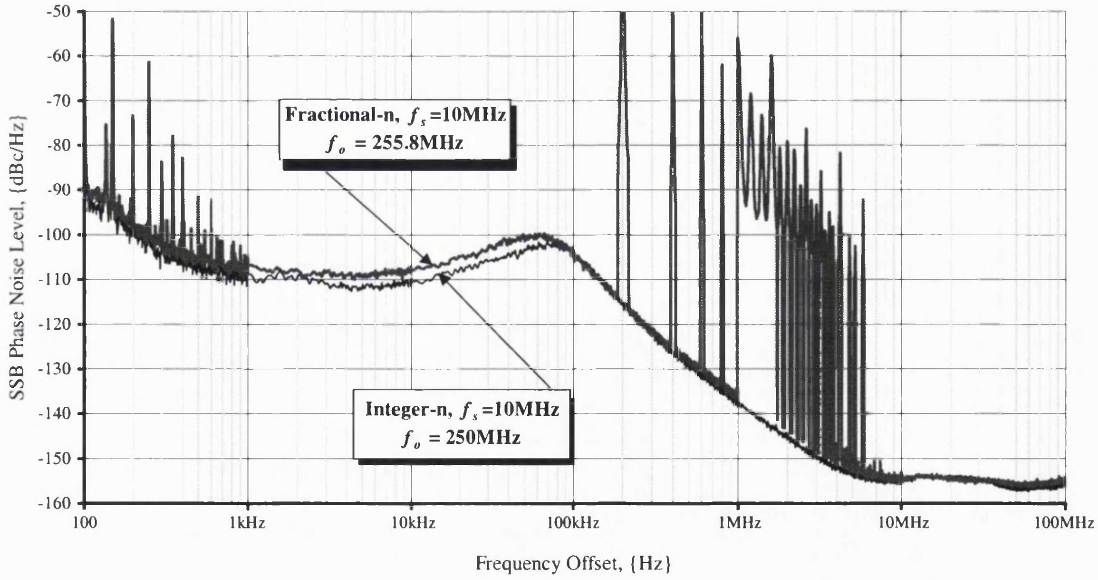


Figure 3.3 Comparison between fractional and integer-n SSB phase noise performance

Reviewing all the data sheets offered by manufacturers of commercial fractional-n synthesisers, it soon becomes clear the anticipated in-band phase noise expected with the higher PFD sampling frequencies, is not actually achieved. With the exception of Fan, [48], no explanation for this observed degradation has been offered.

Figure 3.3 is the measured SSB phase noise performance taken from a simple experiment that used the same synthesiser digital dividers and PFD driven firstly in integer-n mode and then by a simple first-order sigma delta fractional-n PLL[8]. From Figure 3.3, it is clear that the in-band phase noise performance is degraded by 3dB, whilst those parts of the SSB phase noise profile attributable to the VCO remain unchanged. This result suggests the in-band phase noise performance does not improve as much as expected, preventing the loop bandwidth from being increased to the desired extent. Furthermore, with only a modest increase in loop bandwidth and a reduction in the "*N*" feedback fraction used in the loop filter component calculations, the loop filter components quickly become very large in value, especially the capacitors, Figure 3.4.

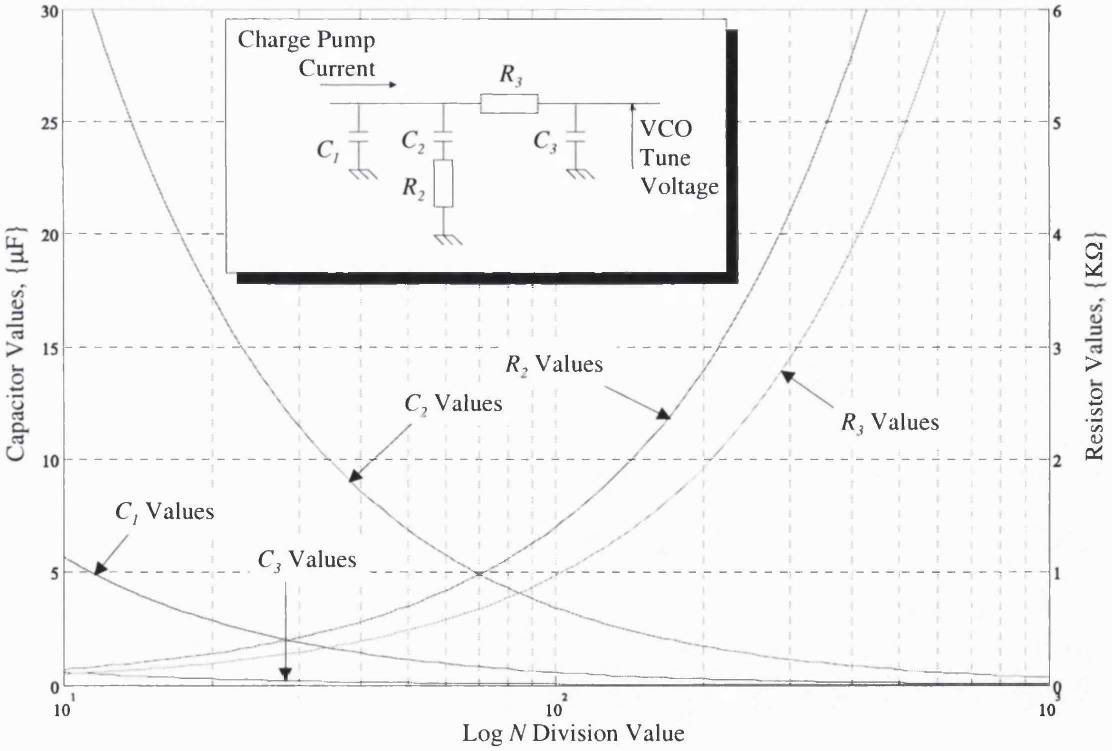


Figure 3.4 Loop filter values for increasing n-division values.

[8] For the purposes of this experiment no attempt was made to compensate for the large spurs, by modifying the loop filter in either measurement.

The problem with larger capacitor values is the dielectrics used have increased leakage, which as shown in section 3.4.2, will extend the lock time. In an attempt to reduce the size of these loop filter capacitor values the temptation is to reduce the loop gain. Assuming the n-divider value remains the same, then potentially only the charge pump gain, or the VCO gain can be reduced. Adjusting the VCO gain is not normally possible because of the tuning range versus available tuning voltage constraint, which leaves only the charge pump gain. Decreasing the charge pump gain, (or PLL error signal), is not normally the preferred solution in any closed loop system, particularly in low noise applications, because it requires the real resistive part of the loop filter to increase, Figure 3.5, to compensate for the change in loop gain. This increase in the resistive component of the loop filter raises the loop filter thermal noise contribution to the overall PLL noise. Therefore in a practical fractional-n synthesiser application a compromise between the charge pump gain and the maximum sampling frequency, which scales the loop filter component values, has to be made.

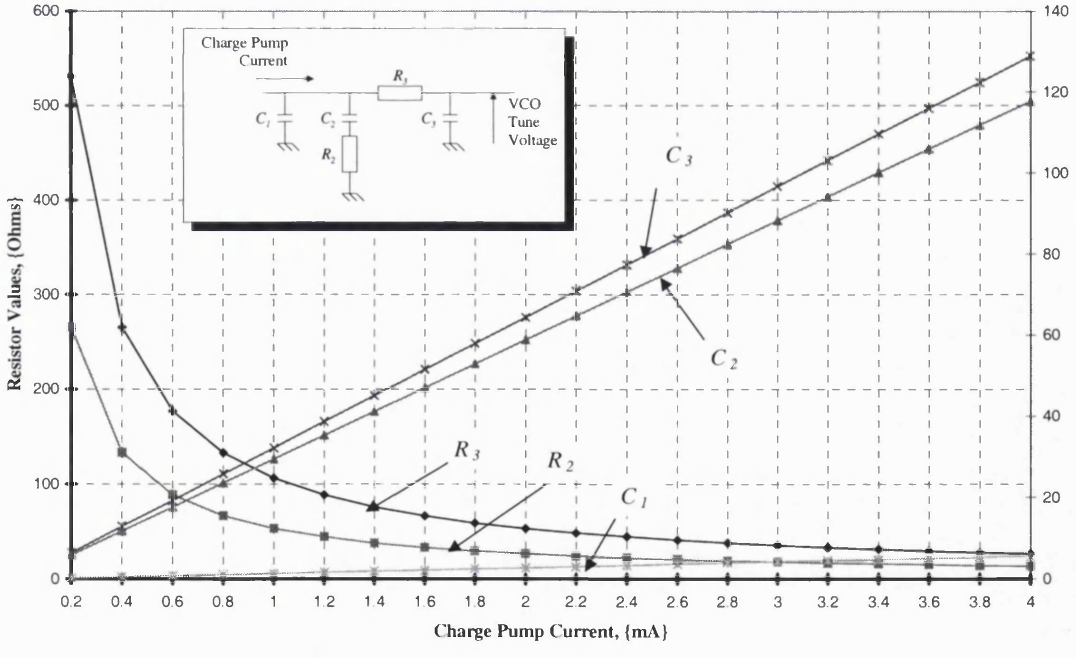


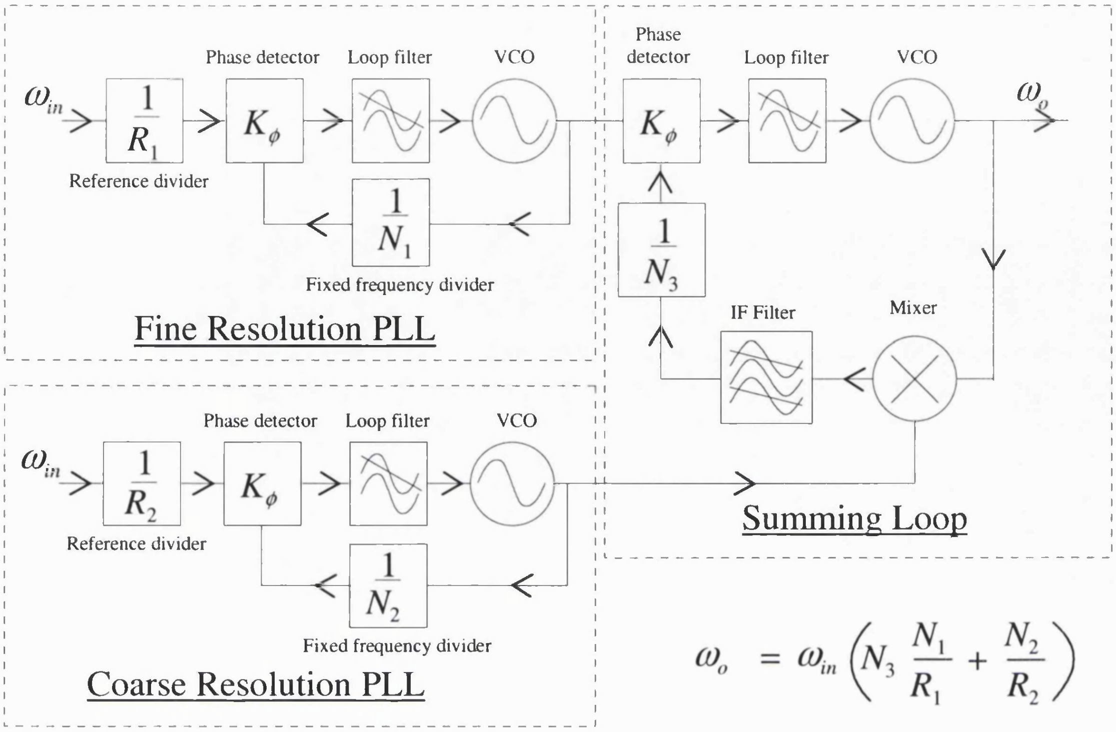
Figure 3.5 Typical variation of loop filter values for increasing charge pump gain

In conclusion, some of the potential advantages of a fractional-n synthesiser are lost with its disappointing in-band phase noise performance and the limitations of the large capacitors necessary to build the loop filter. For this reason, fractional-n frequency synthesiser techniques were dismissed in this study, as the single solution capable of meeting the demands required for a frequency agile, low noise synthesiser.

3.3.3 Multiple Loop Synthesisers

Traditionally, for all demanding synthesisers, multiple loops are used [49, 50, 51], because they allow the flexibility of independently optimising each synthesiser's performance as necessary before they are combined to give the complete synthesised signal source. A typical multiple loop synthesiser will use one synthesiser for coarse frequency steps and another for fine frequency resolution before being summed to give their combined performance, Figure 3.6.

The fine frequency resolution loop offers the fine frequency steps at the expense of tuning range, whilst the coarse frequency loop provides coarse frequency steps limited to the maximum tuning range of the fine frequency synthesiser. With some specifications requiring fairly stringent lock time requirements the speed limitation is the narrow bandwidth required of the fine frequency synthesiser. To overcome this problem, one technique is to synthesise at a much higher frequency and then divide down the output signal. This gives larger frequency steps within this PLL and hence better lock times, before being divided down to give the required frequency resolution. An additional advantage of this technique is that both the phase noise and spurious components are also reduced by the frequency dividing action. Using this approach the designer is better able to simultaneously meet the usually incompatible preconditions of phase noise, lock time and frequency resolution required for more demanding applications. The major disadvantage of this synthesiser architecture is the number of signal sources required, each with their potential for introducing unwanted spurious products whose levels are limited by the electrical and mechanical isolation available within the system.



$$\omega_o = \omega_{in}\left(N_3 \frac{N_1}{R_1} + \frac{N_2}{R_2}\right)$$

Figure 3.6 Example of a multiple loop synthesiser

3.4 Elements of Mixed Signal Frequency Synthesisers

3.4.1 Phase Frequency Detectors

The phase frequency detector, (PFD), is undoubtedly the heart of any PLL synthesiser. This is implemented in many disguises either as a discrete combination of phase detector and frequency discriminator, or more commonly as a single phase/frequency element. The embodiment of a phase/frequency detector within the phase locked loop provides for feedback and phase comparison, frequency steering during the locking process and, as will be shown later, accounts for the level of in-band phase noise in modern digital phase locked loops. Irrespective of the PFD topology, every version must possess the dual features of frequency and phase discrimination ensuring unambiguous steering towards phase lock of the fundamental components of each of the reference and fed-back VCO signals.

With the VCO within the phase capture range, a seamless transition to phase capture and maintained phase lock is provided, preferably with high gain. High gain is preferred for best signal to noise of the resultant output error signal. An inadequate error signal to noise ratio, will lead to loop instability.

Within most modern commercial integrated synthesiser chips the PFD output is followed by a charge pump circuit. Charge pump circuits have become popular because they momentarily connect the loop filter to a current source, (defined by the PFD error signal period), and then remain tri-state for the remainder of the sampling period. This current impulse signal is integrated by the loop filter to provide the VCO control voltage necessary for maintaining phase lock.

$$V = \frac{1}{C}\int_{0}^{\tau} i\,dt \qquad \text{Eq 3.1}$$

Generally, the limitation of the tri-state phase/frequency detector, (PFD), is the dead-band that occurs at zero phase difference between the two PFD input signals. The normal operating point of a PLL is for zero phase difference between the two input signals at the two PFD inputs, which is where this tri-state PFD non-linearity exists. Consequently, this leads to a degree of uncertainty by the PFD, at the normal operating point of a PLL, Figure 3.7.

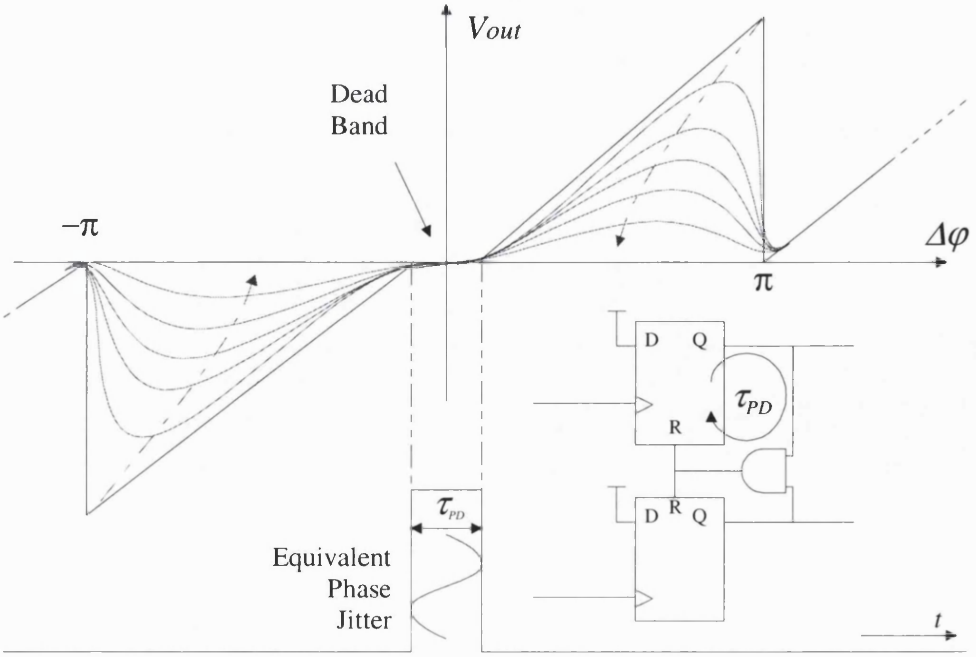


Figure 3.7 Typical PFD transfer characteristic showing dead-band region

The dead band is the point in the loop where the phase error falls to zero thereby reducing the loop bandwidth, consequently several alternative PFD logic configurations have been proposed, each attempting to provide an offset from this dead-zone region. One interesting approach described by Keese, [52], combines the charge pump with the digital PFD to give a closed loop system that derives the logic reset from the charge pump output, Figure 3.8.

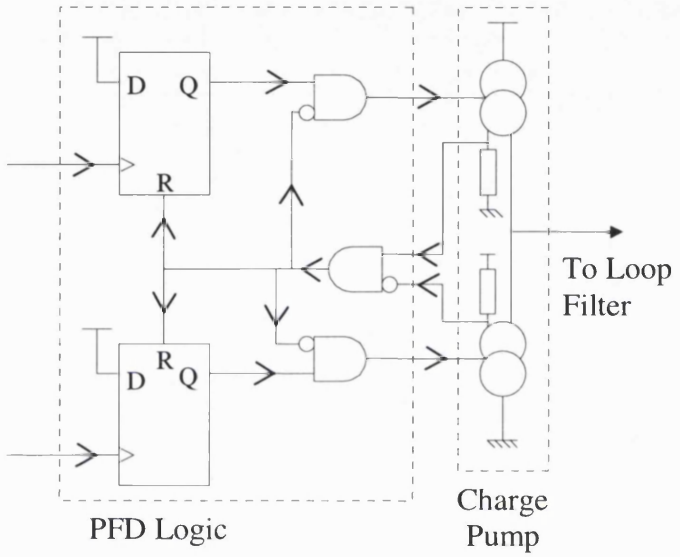


Figure 3.8 Combination of PFD and charge pump to offset the dead-band region

Although this configuration does not remove the dead-band, it does minimise the current imbalance injected into the loop filter caused by the combination of the dead-zone region and different switching times of the charge pump current source and sink circuits.

The dead-band effect, (sometimes known as the dead-beat), is an inherent problem within the tri-state phase frequency detector. It is caused by the finite propagation delay, τ_{PD} Figure 3.7, which exists in the combined feedback and D-type reset path, and gives rise to a minimum phase difference that cannot be resolved by this type of phase frequency detector. When the phase difference between the two input edges to the PFD approach zero degrees, the PFD transfer characteristic possesses zero gain, (reducing the loop gain to near-zero), causing the PFD output to oscillate about this zero phase difference value, Figure 3.7.

With this finite phase uncertainty close to phase lock, the phase locked loop will tend to randomly oscillate thereby introducing a level of phase jitter proportional to the sampling period, T_s.

$$\Delta\phi_{pk-pk}(t) \cong \frac{2\pi\tau_{PD}}{T_s} \qquad \text{Eq 3.2}$$

This is the peak-to-peak value, assuming this is a uniform probability density function then the equivalent RMS value becomes:

$$\Delta\phi_{RMS}(t) \cong \frac{\pi}{\sqrt{3}}\frac{\tau_{PD}}{T_s} \qquad \text{Eq 3.3}$$

The absolute value of this dead-band phase jitter seen at the PLL's VCO output, will depend upon the phase transfer gain from the digital part of the PFD to the PLL output, at the VCO output frequency f_o. The propagation delay of the PFD represents the maximum useful operating frequency of that PFD, Figure 3.9.

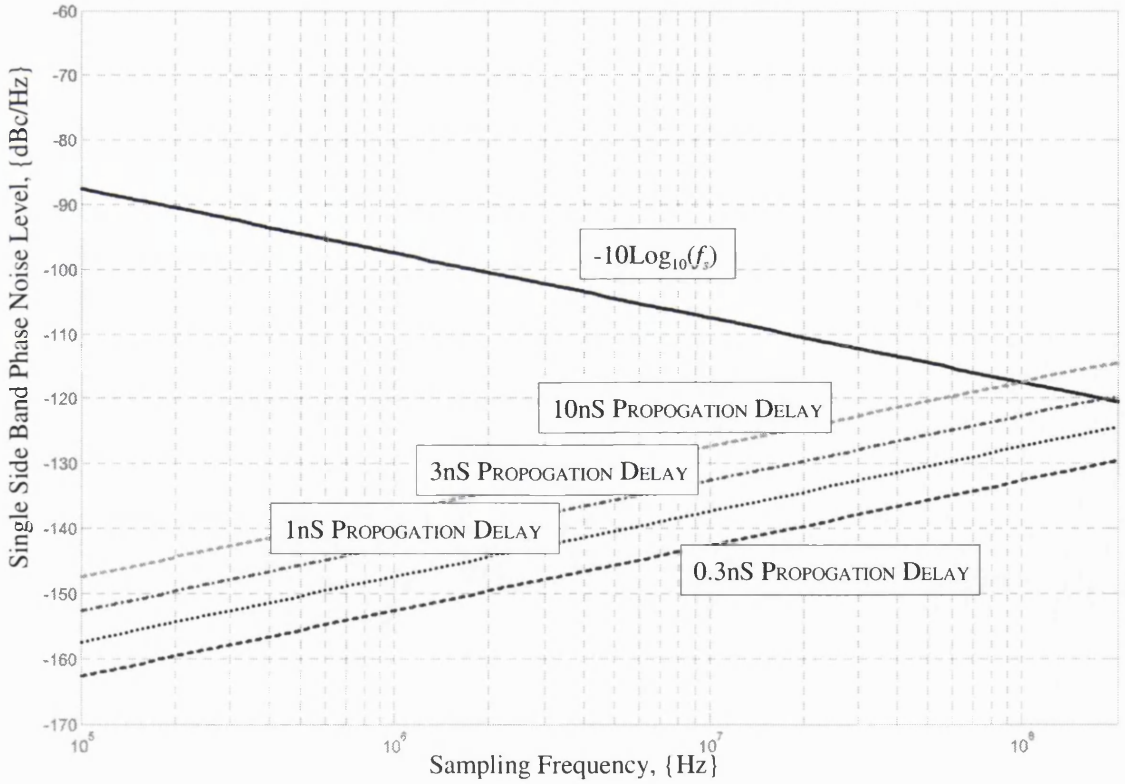
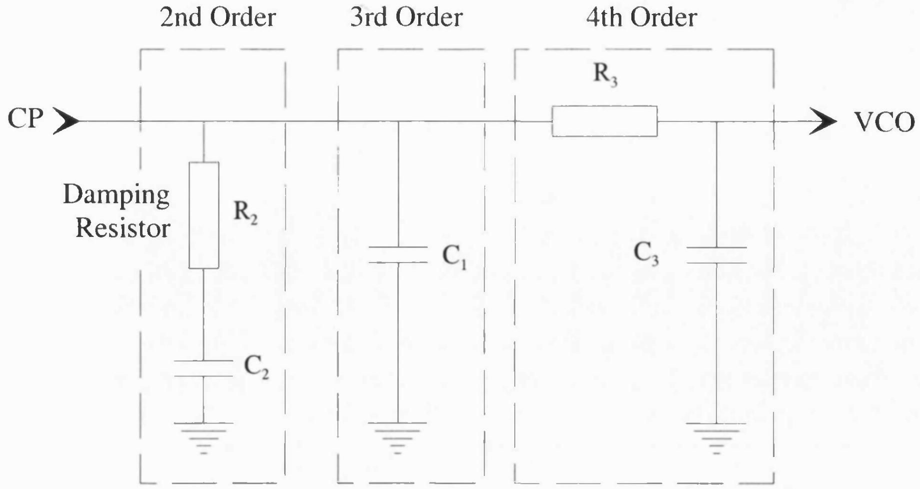


Figure 3.9 PFD propagation delay limits on PFD noise floor

Clearly, for sampling periods less than the PFD propagation period, the PFD is unable to resolve the PLL phase error and will "oscillate" across this propagation period resulting in catastrophic failure of the PFD operation.

3.4.2 The Loop Filter

The loop filter is the key element within the PLL, which is used to set the phase noise, spurious performance, lock time and tracking dynamics within the confines of all other elements found within the system. A number of typical loop filter topologies exist depending upon the type of phase/frequency detection system employed, however it is interesting to note that the popular passive "trans-resistance" filter, first used in 1943 by Wendt *et al.*, [12], in the synthesiser shown in Figure 2.1, still prevails in modern charge pump driven PLLs. Within this document, only this passive charge pump driven filter will be considered. The input signal to this filter is a pulse width modulated current sequence, with each pulse width comprising the correction pulse and the indeterminate dead band period of the digital PFD. The loop filter serves the dual purpose of filtering and integrating this pulse, (a hold function), to convert this current pulse into the steady state voltage used to control the VCO.



Order	Loop Gain Transfer Function (s-domain)
2^{nd}	$\frac{K}{s}\frac{1+sR_2C_2}{sC_2}$
3^{rd}	$\frac{K}{s}\frac{1+sR_2C_2}{s\left(C_1+C_2\right)\left(1+sR_2\frac{C_1C_2}{C_1+C_2}\right)}$
4^{th}	$\frac{K}{s}\frac{1+sR_2C_2}{s\left(C_1+C_2+C_3\right)\left(1+s\left(\frac{\left(R_2C_2\right)\left(C_1+C_3\right)+\left(R_3C_3\right)\left(C_1+C_2\right)}{C_1+C_2+C_3}\right)+s^2\frac{R_2R_3C_1C_2C_3}{C_1+C_2+C_3}\right)}$

Table 3.1 Charge pump driven passive loop filter configurations

Referring to Table 3.1, the ideal loop filter equations overlook the parasitic effects typically found in an actual filter which degrade both the loop gain suppression, alter the lock time performance and raise the sampling spurious levels of the complete phase locked loop, from the ideal. To fully account for these parasitic effects the filter model should include i), the combined finite charge pump output and VCO input resistances, ii), capacitor dielectric absorption parasitics and iii), the small voltage found at the VCO tune voltage input, Figure 3.10.

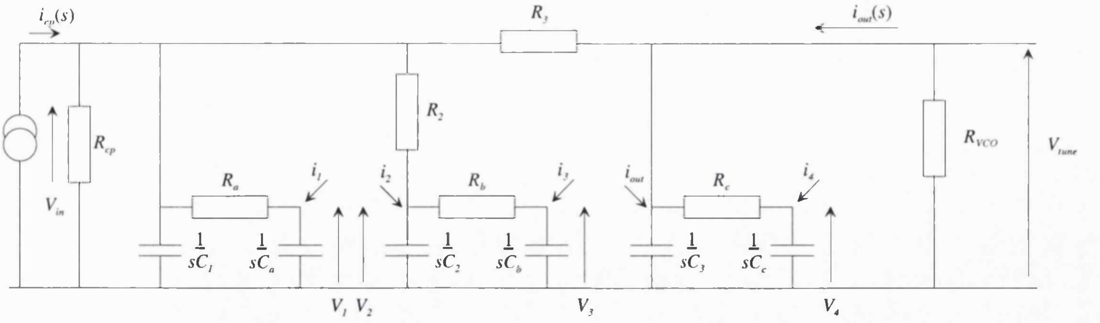


Figure 3.10 Loop filter model with all parasitics added

$$\begin{bmatrix} i_{cp} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_{out} \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{R_{cp}}+\frac{1}{R_a}+\frac{1}{R_2}+\frac{1}{R_3}+sC_1\right) & \left(\frac{-1}{R_a}\right) & \left(\frac{-1}{R_2}\right) & 0 & 0 & \left(\frac{-1}{R_3}\right) \\ \left(\frac{-1}{R_a}\right) & \left(\frac{1}{R_a}+sC_a\right) & 0 & 0 & 0 & 0 \\ \left(\frac{-1}{R_2}\right) & 0 & \left(\frac{1}{R_2}+\frac{1}{R_b}+sR_b\right) & \left(\frac{-1}{R_b}\right) & 0 & 0 \\ 0 & 0 & \left(\frac{-1}{R_b}\right) & \left(\frac{1}{R_b}+sC_b\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{1}{R_c}+sC_c\right) & \left(\frac{-1}{R_c}\right) \\ \left(\frac{-1}{R_3}\right) & 0 & 0 & 0 & \left(\frac{-1}{R_c}\right) & \left(\frac{1}{R_3}+\frac{1}{R_c}+\frac{1}{R_{vco}}+sC_b\right) \end{bmatrix} \begin{bmatrix} V_{in} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_{tune} \end{bmatrix} \qquad \text{Eq 3.4}$$

i) There are two values of charge pump output resistance; one during the “on” period and the other during the “off” period. However, the charge pump “on” period is usually very small compared with its “off” period therefore only the charge pump “off” resistance is of interest. This resistance should ideally be very large to minimise the amount of current leakage during the charge pump off period, otherwise this current leakage sets up a small saw-tooth type waveform that modulates the VCO thereby introducing unwanted phase error and sampling sidebands to be observed on the synthesiser output. When this finite charge pump off resistance is included within the model for the filter transfer function, a very low frequency pole results, limiting the filter gain found very close to the carrier, Figure 3.11.

ii) One of the first descriptions of dielectric absorption (hysteresis) properties was given in 1937, [53], and linked to capacitors by Dow in 1958, [54]. This effect is caused by the dielectric loss of the loop filter capacitors used, (i.e. the tan(δ)), which exhibits an effect sometimes called; "Dielectric Relaxation", "Dielectric Absorption" or "Capacitor Soakage", depending upon the author. The net effect is the exceedingly long time constants observed across the larger loop filter capacitors, which cannot be explained through the mathematics of the obvious loop time constants. Indeed their low values, (typically several milliseconds), suggest poles that should make the loop unstable, Figure 3.12 although the presence of this parasitic is not obvious on the Bode plot of the loop filter, Figure 3.11. When a capacitor is subject to the sudden stress caused by a change in charge across its plates, a capacitor requires a period of "soaking", [55], to return to internal charge equilibrium. Thus the d*C*/d*t* term cannot be neglected in the definition of current, $i=\frac{dQ}{dt}=C\frac{dV}{dt}+V\frac{dC}{dt}$ making it responsible for these unexpectedly long settling periods observed.

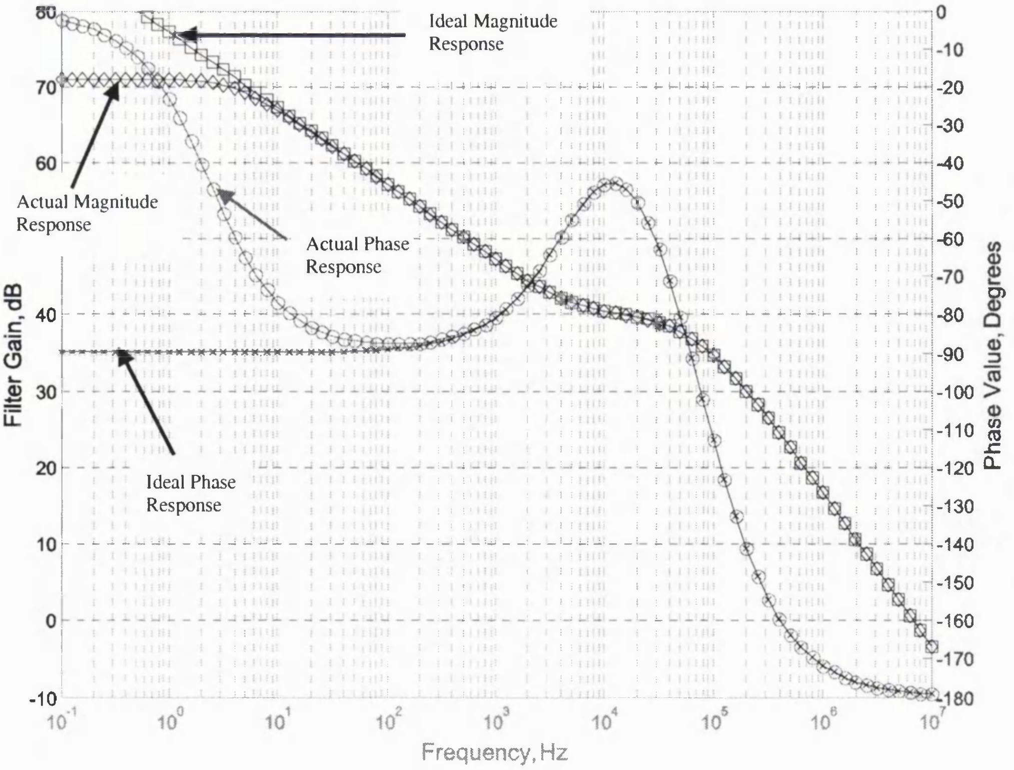


Figure 3.11 Bode plot showing effects of finite charge pump resistance on loop filter response

In a typical GSM TDMA application where the synthesisers are constantly being re-programmed to new frequencies at approximately 577μS intervals, the capacitors exhibit a memory effect lasting many previous frequency-hopping periods. Consequently, although the synthesiser has apparently settled in phase, there is a gradual frequency drift throughout each burst period. In a typical measurement[9], this result can be compared to the situation when a non-hopping synthesiser is used as the local oscillator source giving a steady frequency error of typically 8Hz. However, when the same synthesiser is frequency hopping, (to the same channel, from different frequency offsets), the frequency error can rise to 40Hz whilst the measured RMS phase error remains constant during both measurements, Figure 3.12. If the hopping pattern is varied, ensuring the synthesiser hops from different frequency offsets giving rise to different capacitor charges, an alternative frequency error pattern for each individual burst period will be recorded, because of the memory effect. This memory effect accounts for the relative change in frequency errors measured.

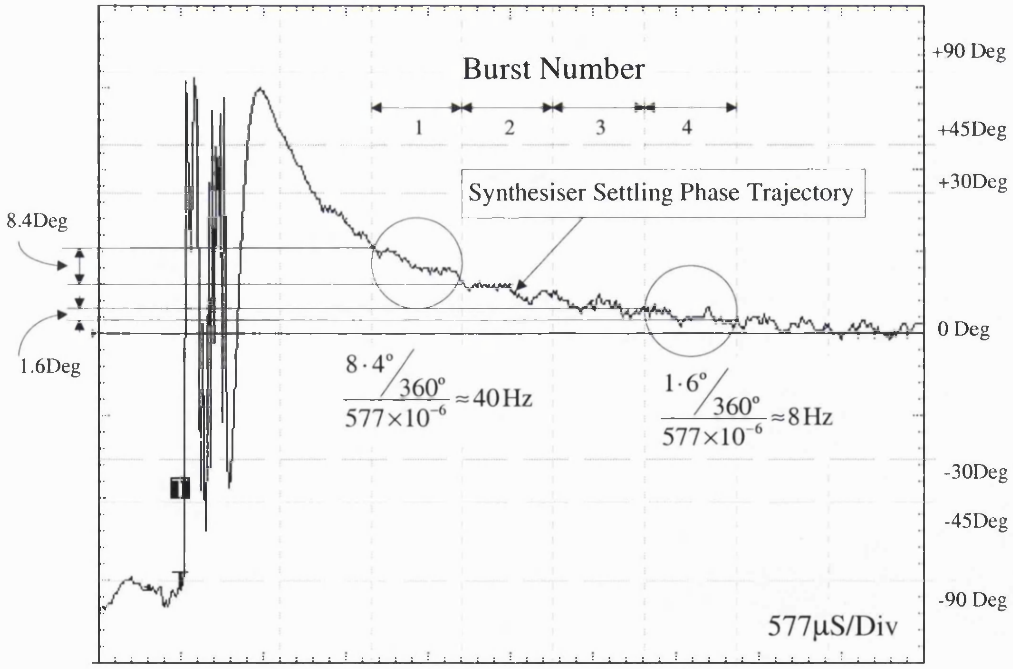


Figure 3.12 Measured phase error across multiple burst periods and their frequency error

[9] This example is taken from a typical GSM 900MHz application where one example of this problem was observed.

iii) VCO "backfire" comes from the partial rectification of the RF signal across the varactor diodes. This partial rectification injects a small DC offset into the loop filter that stresses the loop in its attempt to compensate for this error signal, artificially raising the sampling sideband levels. For optimum VCO phase noise performance the wanted signal level should be kept high in the resonator, to minimise the signal to noise ratio, however this can be a problem when large tuning ranges are required of the VCO. For a given varactor capacitance range, it is very often necessary to tightly couple the resonator to the varactor thereby increasing the risk of RF energy being rectified by the varactor diode.

3.4.3 The VCO

The VCO is the output element universally used in all analogue phase locked loops and is controlled by the voltage held on the loop filter circuit, to whatever degree of accuracy defined by the demands of the remainder of the synthesiser. The synthesiser could be a single loop integer or fractional-n system, or alternatively a multi-loop synthesiser comprised of a variety of synthesis techniques. In whatever form of synthesiser, the VCO output synthesises the required output frequency from the given reference signal. Because the analysis of VCO phase noise performance is well documented, only the salient formulae during the evolution of this VCO noise model are quoted in this text.

Although one of the most popular mathematical descriptions of the VCO phase noise profile is often attributed to Leeson, there are many references, which precede his work. Work published by Berstein in 1938, [56], that relates to the theory of free-running oscillators, provides an interesting account of how oscillator noise can be derived from an analysis of the modulation of noise. Since then, these concepts have been developed by many authors most notably Edson and Mullen, [57, 58]. In Edson's work he considers how noise results from the thermal properties of the components found in the VCO.

Given the work of Leeson and Scherer, [59, 60] provide a good insight into the mechanics of phase noise using arguments based around a feedback system which is appropriate to this work on closed loop systems, their work will be presented here.

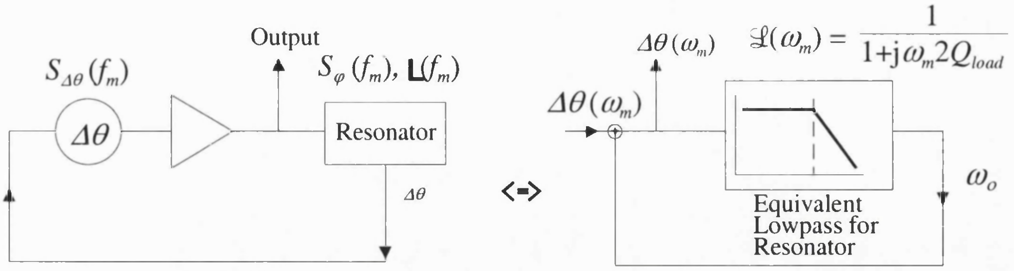


Figure 3.13 Scherer's description of Leeson's phase noise model, [60]

Leeson [59] is accredited with presenting the first practical model for phase noise in a free running oscillator. His model described by Scherer [60] in Figure 3.13, consists of an amplifier with a noise factor F and a resonator to limit the loop bandwidth, which he then translates into a phase driven feedback loop for analysis. The resonator bandwidth limits the closed loop gain inside the feedback system and hence the amplification of the active device baseband noise. Using transmission theory, he equates the double-sided bandwidth of the resonator to an equivalent low pass function. From this assumption, he arrives at the following equation.

$$\mathcal{L}(\omega_m) = \frac{1}{1 + j\dfrac{\omega_m 2Q_{load}}{\omega_o}} \qquad \text{Eq 3.5}$$

Where:

$\mathcal{L}(\omega_m)$ = the ratio of sideband power in a 1Hz bandwidth at ω_m,

ω_m = the frequency offset in radians,

ω_o = the carrier frequency in radians and

Q_{load} = the loaded Q of the tuned circuit.

In particular $\dfrac{\omega_o}{2Q_{load}}$ represents the half power bandwidth of the resonator. Scherer extended this work to add the flicker corner that accounts for the noise departure point from the flat noise floor and in so doing demonstrated the properties which need considering when designing a free running oscillator for minimum phase noise. His formula, Eq 3.6, isolates the different factors that make up the total energy within the resonator.

$$\mathcal{L}(\omega_m)=\underbrace{\frac{1}{8}\frac{FkT}{P_{sav}}\frac{\omega_o^2}{\omega_m^2}}_{\text{Phase Perturbation}}\left(\underbrace{\frac{P_{in}}{\omega_o W_e}}_{\text{Input Power over Reactive Power}}+\underbrace{\frac{1}{Q_{unl}}}_{\text{Resonator Q}}+\underbrace{\frac{P_{sig}}{\omega_o W_e}}_{\text{Signal Power over Reactive Power}}\right)^2\underbrace{\left(1+\frac{\omega_c}{\omega_m}\right)}_{\text{Flicker Effect}} \qquad \text{Eq 3.6}$$

Where:

$\mathcal{L}(\omega_m)$ = the ratio of sideband power in a 1Hz bandwidth at ω_m,

ω_m = the frequency offset in radians,

ω_o = the carrier frequency in radians,

ω_c = flicker frequency of the active device in radians,

F = noise factor,

k = Boltzmann's constant, ($1{\cdot}374\times10^{-23}$ J/K),

T = temperature in Kelvins,

P_{in} = input power to active device,

P_{sig} = oscillator output signal power,

P_{sav} = the average power at the input of the active device,

Q_{unl} = unloaded Q of resonator,

W_e = reactive energy within the resonator.

For optimal phase noise performance, the above equation requires high signal levels in a high Q resonator to reduce the loop bandwidth and hence the loop gain. By limiting this gain, the amplification of intrinsic baseband current and voltage noise of the active device up-converted to the carrier by the commutating action of the oscillating device, is minimised. Beyond the loop bandwidth and the flicker corner frequency, (whichever may be the larger frequency offset), the noise floor is limited by the signal to noise ratio of the signal within the resonator which should be maximised within a given oscillator design.

To account for the degradation of the resonator by the reactance control circuit, which pushes the breakpoint further from the flicker corners, Rohde [61] modified Eq 3.6 to give:

$$\mathcal{L}(f_m)\{dBc/Hz\} = 10\,\mathrm{Log}\left\{\left(1+\frac{f_o^2}{(2f_m Q_{load})^2}\right)\times\left(1+\frac{f_c}{f_m}\right)\times\frac{FkT}{2P_{sav}}\times\left[1-\left(\frac{Q_{load}}{Q_O}\right)\right]+\frac{2kTR_v K_{VCO}^2}{f_m^2}\right\} \qquad \text{Eq 3.7}$$

where

$\mathcal{L}(f_m)$ = the ratio of sideband power in a 1Hz bandwidth at f_m to the total power in decibels (spectral density),

f_m = the frequency offset in Hertz,

f_o = the output frequency in Hertz,

f_c = flicker frequency of the active device in Hertz,

Q_{load} = the loaded Q of the tuned circuit,

Q_o = the unloaded Q of the tuned circuit; $Q_o > Q_{load}$,

F = noise factor,

k = Boltzmann's constant, ($1{\cdot}374\times10^{-23}$ J/K),

T = temperature in Kelvins,

P_{sav} = the average power at the input of the active device,

R_v = the equivalent resistance of the varactor diode (typically 200Ω to 10kΩ), and

K_{VCO} = oscillator gain, (Hz/V).

Very often the large frequency ranges required of a VCO necessitate heavy coupling of the varactor diode to the resonator circuit, which in turn reduces the overall loaded Q of the resonator and hence the phase noise performance of the VCO.

For an oscillator with a low Q resonator the oscillator loop gain amplifies the flat device noise floor to initially give a 20dB per decade rise in noise, followed by a 30dB per decade slope within the device's natural flicker corner frequency. Conversely, an oscillator using a high Q resonator, possessing a half bandwidth value less than the device flicker corner frequency, initially gives a 10dB per decade departure from the noise floor followed by a 30dB per decade slope inside the resonator half bandwidth frequency offset, Figure 3.14.

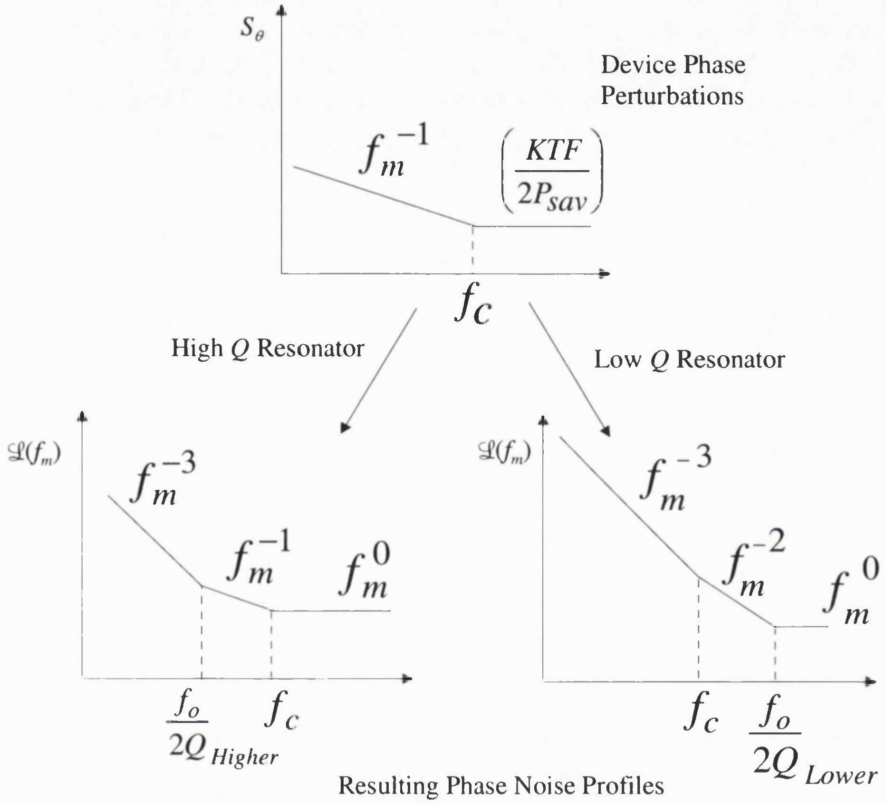


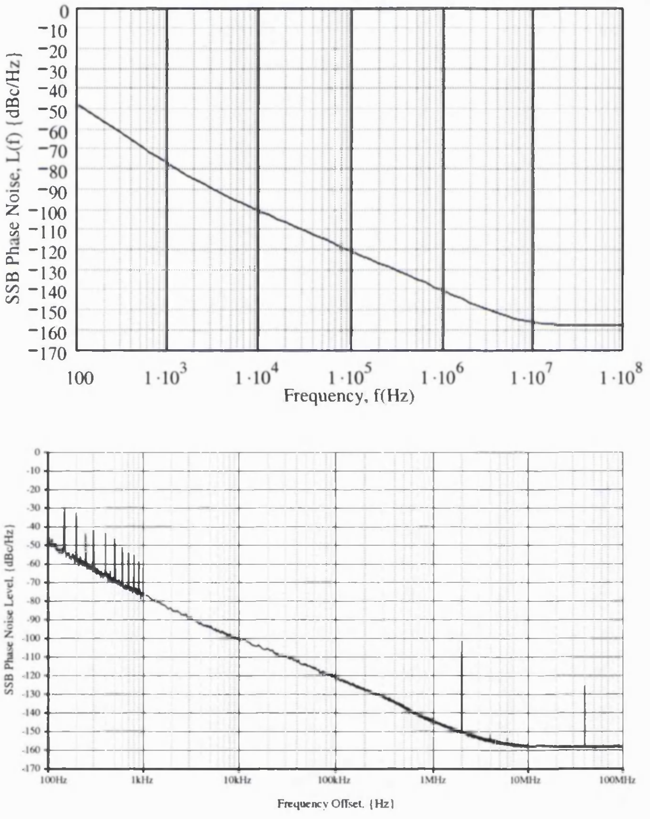
Figure 3.14 Effect of high and low-Q resonators on amplification of device noise, [60]

The value of the preceding outline on VCO theory is in modelling the free running VCO noise profile. This noise profile is included as one of the dominant noise sources found in a PLL when predicting the overall SSB phase noise profile of the complete synthesiser. For this work it will be assumed all high frequency VCOs possess a half-bandwidth frequency beyond the flicker frequency.

Taking the Rice, [62], assertion that the VCO phase noise is Gaussian distributed then the mean value of SSB phase noise can be represented by the magnitude of the offset noise formula Eq 3.7.

$$\mathcal{L}(f_m) = Plateau_{VCO} + 20\mathrm{Log}_{10}\left(1 + \frac{VCO_{f1}}{f_m}\right) + 10\mathrm{Log}_{10}\left(1 + \frac{VCO_{f2}}{f_m}\right) \qquad \text{Eq 3.8}$$

For a higher frequency VCO possessing a lower resonator Q, the breakpoint, VCO_{f1} defines the 20dB per decade slope departure from the device noise floor set by the resonator half-power bandwidth and VCO_{f2} is the device flicker corner frequency. For example, the measured free running phase noise profile of a typical 1·6GHz VCO with a gain of 33MHz/V, defines the $Plateau_{VCO}$ SSB noise value as –158dBc/Hz, the first breakpoint, VCO_{f1} as 1·7MHz and the flicker corner frequency breakpoint, VCO_{f2} as 1·3kHz. A comparison between the measured and modelled SSB phase noise profiles of this VCO are shown in Figure 3.15.



Modelled Noise Profile

Measured Noise Profile

Figure 3.15 Comparison between VCO phase noise model and measurements

Modelling crystal reference noise can be accomplished using the same mathematics, noting that generally the higher Q of the crystal results in the half bandwidth breakpoint falling at a much lower carrier offset frequency than the device flicker corner frequency. This gives rise to the different slope profile:

$$\mathcal{L}(f_m) = Plateau_{Ref} + 10\mathrm{Log}_{10}\left(1 + \frac{Ref_{f1}}{f_m}\right) + 20\mathrm{Log}_{10}\left(1 + \frac{Ref_{f2}}{f_m}\right) \qquad \text{Eq 3.9}$$

The breakpoint, *Ref*$_{f1}$ accounts for the device flicker corner frequency, giving the 10dB per decade slope departure from the flat device noise floor whilst *Ref*$_{f2}$ raises the slope profile to 30dB per decade, to account for the VCO closed loop gain.

VCO operation is usually limited to at best a single octave. Therefore, if the required frequency ranges are integer ratios of one another, then simple divider circuits can be used, however, when non-integer frequency ranges are required the only practical solution is to use switched tuning resonators to retain the overall noise performance of a single oscillator. This technique is the preferred means of straddling different mobile communications bands using a single PLL.

3.4.4 Digital Dividers

Digital dividers are typically found both inside and outside a phase-locked loop. Outside the loop they are used to divide down the incoming reference signal to give the required sampling frequency, whilst inside a loop they are typically used to down convert the VCO signal in the feedback path. The output of this down converted VCO signal is used to drive the other input of the PFD at the same frequency as the sampling signal.

Within a PLL, the use of a digital divider in the feedback path and separate digital phase frequency detector, combine to form a multi-rate system. At the divider input there exists a high-speed self-sampler at the VCO frequency, whilst at the divider output the much slower, but phase coherent output signal, is re-sampled at the phase detector input by the reference signal[10]. The self-sampled input signal at the divider input undergoes an averaging process by the internal "*A*" and "*B*" counters used in a typical dual-modulus circuit. This serves to smooth out the instantaneous phase changes, reducing their relative phase noise levels.

[10] Within the scope of this work it has been assumed a 1:1 ratio of signal frequencies exists between the two inputs of the phase frequency detector, and sub-harmonic sampling is not used.

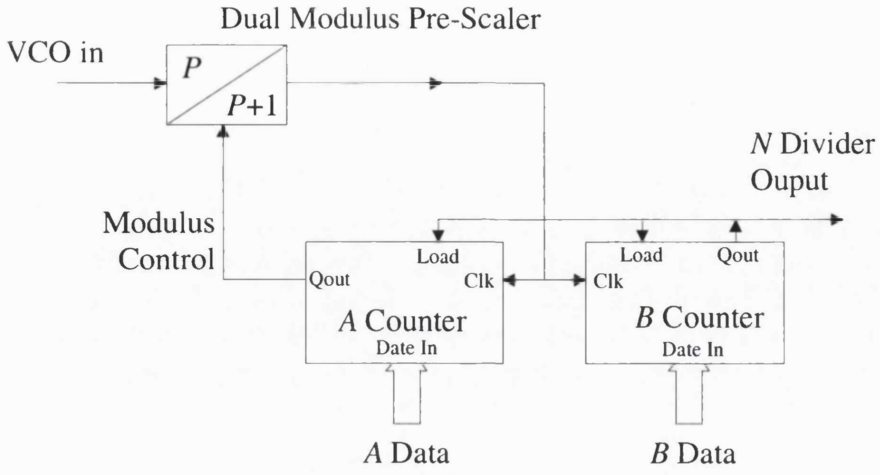


Figure 3.16 Typical dual modulus divider circuit

The digital dividers in the feedback of both integer and fractional-n phase locked loops usually consist of a high frequency dual modulus pre-scaler circuit, whose output is used to drive a pair of lower frequency digital down counters, Figure 3.16. The operation of this circuit starts with the dual modulus pre-scaler dividing the VCO input by P+1 to give an output clock signal, which drives both the A and B down counters. When the A counter reaches zero, its output toggles the dual modulus counter to cause it to divide by P for the remainder of the cycle. During the P+1 division period, the B counter will also have been down-counting and will continue to down-count in the P division period, which defines that $B \geq A$. When the B counter reaches zero its output is used to reload, (and hence reset), each of these two counters and provide a pulse representing the divided down VCO signal at the end of the division cycle.

Thus, for the first part of this division cycle the division value, (into the down counters), is $A(P+1)$, whilst for the second part of the division cycle the division value is $(B\text{-}A)P$, this gives the total N division value as:

$$\begin{aligned} N &= A(P+1)+(B-A)P \\ &= A+BP \end{aligned} \quad \text{Eq 3.10}$$

Modelling the digital divider noise profile follows the same format as that of the VCO and reference source above. However, to build this model no known techniques are available for isolating the noise floor or the noise breakpoints of the digital N and R dividers within a commercial integrated synthesiser chip. To overcome this problem a simulated noise profile was obtained from the manufacturer from which the noise plateau and noise break points were determined for the synthesiser chip used throughout these investigations. The n and r-dividers are assumed to possess similar noise profiles within the same integrated circuit.

Of particular significance is the r-divider that is outside the PLL, particularly when the value of R is high and the reference crystal noise is already low. Theoretically the r-divider will reduce the incoming crystal reference noise by "$20\text{Log}_{10}(R)$", which in some PLL applications will fall well below the r-divider's own noise floor causing the r-divider noise to dominate, Figure 3.17. Furthermore, as the r-division value increases some of the lower level spurs found on the crystal reference signal, fall below this r-divider noise floor and are lost.

For the n-division process the level of phase noise entering the n-dividers is much higher and is unlikely to be lost below the n-divider noise floor.

Taking Div_{fl} as the device flicker noise breakpoint, the r-divider noise model with an ultimate noise floor value given by $Plateau_{Div}$ can simply be modelled as:

$$\mathcal{L}(f_m) = Plateau_{\text{Div}} + 10\text{Log}_{10}\left(1+\frac{Div_{f1}}{f_m}\right) \qquad \text{Eq 3.11}$$

Which is summed using an RMS noise addition with the reference source phase noise model, Eq 3.9. The result is demonstrated in Figure 3.17 where it can be seen that the theoretical divided down reference noise falls below the r-divider noise. This is a particularly useful technique for evaluating synthesiser chips when using less than optimal reference sources that contain a combination of poor phase noise or spurious products.

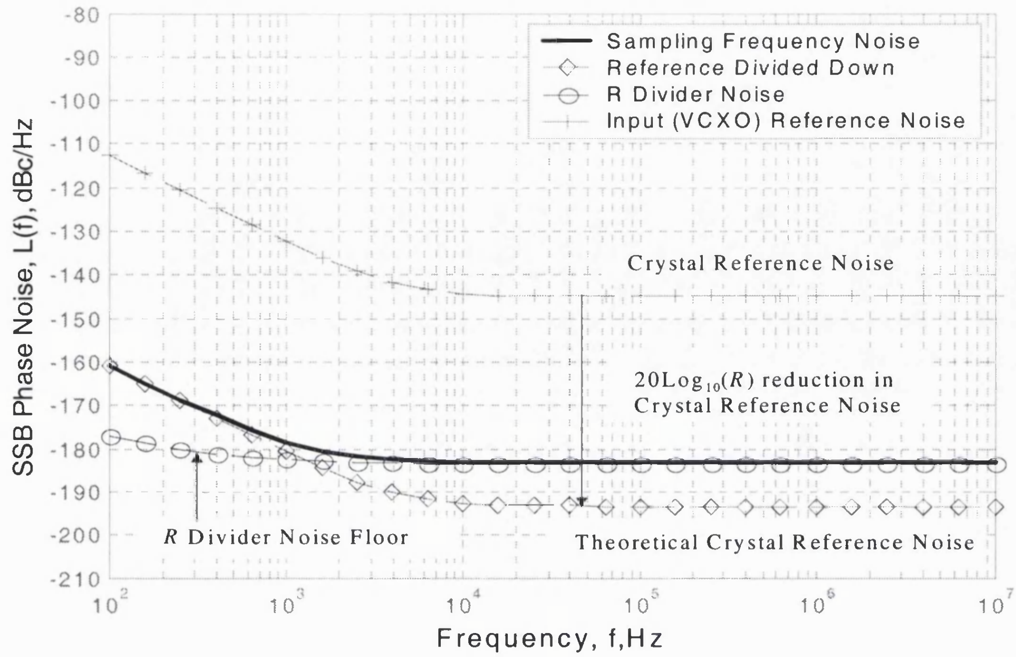


Figure 3.17 Effect of *R* division on crystal reference noise into PFD

The DDS unit is another form of digital divider and can be modelled using the same noise model, although the level of its output noise floor is determined by the DAC performance at the DDS output. Using residual phase noise measurements, (described in section 5.3), the phase noise profile for a DAC was measured to provide a DDS noise model, Figure 3.25.

3.4.5 Analogue Mixers

One technique often used within a PLL to reduce the noise multiplication factor resulting from the n-divider in the feedback path, is to mix the VCO output down to a lower frequency. This mixer output is then divided by *N* to give the correct multiplication of the PLL sampling frequency required for the complete design. Thus the sum of *N* times the sampling frequency at the synthesiser's phase frequency detector input plus the mixer offset frequency, gives the PLL output frequency.

$$f_o = N \times f_s + f_{LO} \qquad \text{Eq 3.12}$$

The consequence of this approach is the extra local oscillator source required to drive the mixer and the related intermodulation products that maybe generated.

From the analysis given in appendix A3, the location of mixer intermodulation products can be determined using the general $n \times m$ product term mathematics.

$$\text{Mixer Spurii} = \pm n \times f_{LO} \pm m \times f_{RF} \qquad \text{Eq 3.13}$$

A typical mixer intermodulation spurious analysis consists of independently sweeping the n and m terms through a range of negative through positive values and then choosing local oscillator and intermediate output frequencies for the given RF input frequency range. Usually a compromise is required, which limits the order of acceptable intermodulation products within the application. Most mixer manufacturers produce mixer spurious level charts for their mixers showing measured data of the low order products, because empirical calculation of these signal levels is not straightforward and very dependent upon the source and load impedance presented to the mixer ports. Most of these charts indicate that the odd order n and m products exhibit higher spurious levels as suggested by the Fourier mathematics of the square wave frequency spectrum, typically used to drive the mixer's internal diodes or FET devices.

Within any frequency synthesiser using a frequency mixer, the level of these mixer intermodulation products needs to be minimised to prevent them being amplified within the closed loop system and also being re-mixed by the sampling action found within the PFD and digital dividers of a PLL.

3.5 Elements of Direct Digital Synthesisers

The principles of DDS operation, (sometimes referred to as a Numerically Controlled Oscillator, NCO or Direct Digital Frequency Synthesis, DDFS), are well known; the digital accumulator whose displacement with time generates a phase-ramp at a rate programmed by the input frequency word value is mapped, according to the non-linear transformation of phase to sine, to generate the digital equivalent of a sine wave used to drive the DAC output, Figure 2.4.

3.5.1 The Digital Accumulator

With reference to Figure 3.18, the digital accumulator used as the heart of a DDS, utilises a simple digital adder followed by a gated latch, to synchronise each sum value, ready for feeding back to one input of the adder circuit, where it is again added to the frequency word. In effect this simple circuit is the discrete equivalent of an integrator, with each complete cycle causing an overflow and effectively resetting the integrator. According to Gilmore *et al.*, [63], an A-bit wide digital accumulator provides a periodic sawtooth waveform whose instantaneous value represents the calculated phase for each clock cycle, giving a phase resolution of $2\pi/2^A$ radians.

The input control word fed into the other input of the digital adder determines the rate of change of the phase and hence the required output frequency. During operation, the DDS accumulator is often programmed to give a frequency value, which is not exactly the required frequency because of the finite width of the digital accumulators used. When this happens a small but finite error signal results, whose level will depend upon the ratio of digital bus width to the accumulator clock frequency and will give rise to unwanted spurious signals at the DDS output.

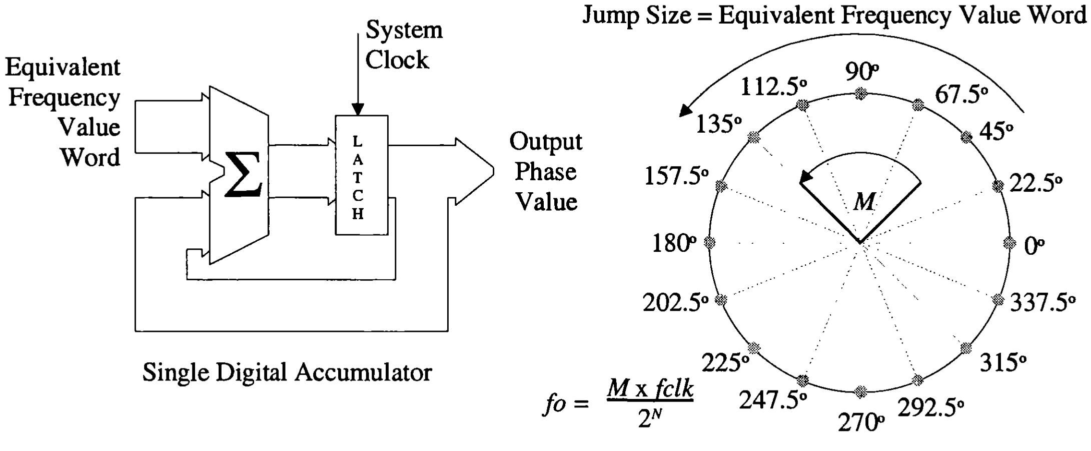

Equivalent Frequency Value Word
System Clock
Σ
LATCH
Output Phase Value
Single Digital Accumulator
Jump Size = Equivalent Frequency Value Word
90°
112.5°
67.5°
135°
45°
157.5°
22.5°
M
180°
0°
202.5°
337.5°
225°
315°
247.5°
270°
292.5°
$fo = \frac{M \times fclk}{2^N}$

Figure 3.18 Digital translation of frequency word to phase value in accumulator

DDS accumulators are essentially a DSP system and as such can easily be modelled using a variety of mathematical tools to emulate their exact operation giving very accurate predictions of MSB spurious frequencies and their amplitudes relative to the carrier. One example of the accuracy of such a model is given in, Figure 3.19, which compares the results from a simple model to those observed on a spectrum analyser using the same DDS set-up, as the model. The DDS used here was the Mitel SP2002, which provides access to the MSB output from the accumulator core.

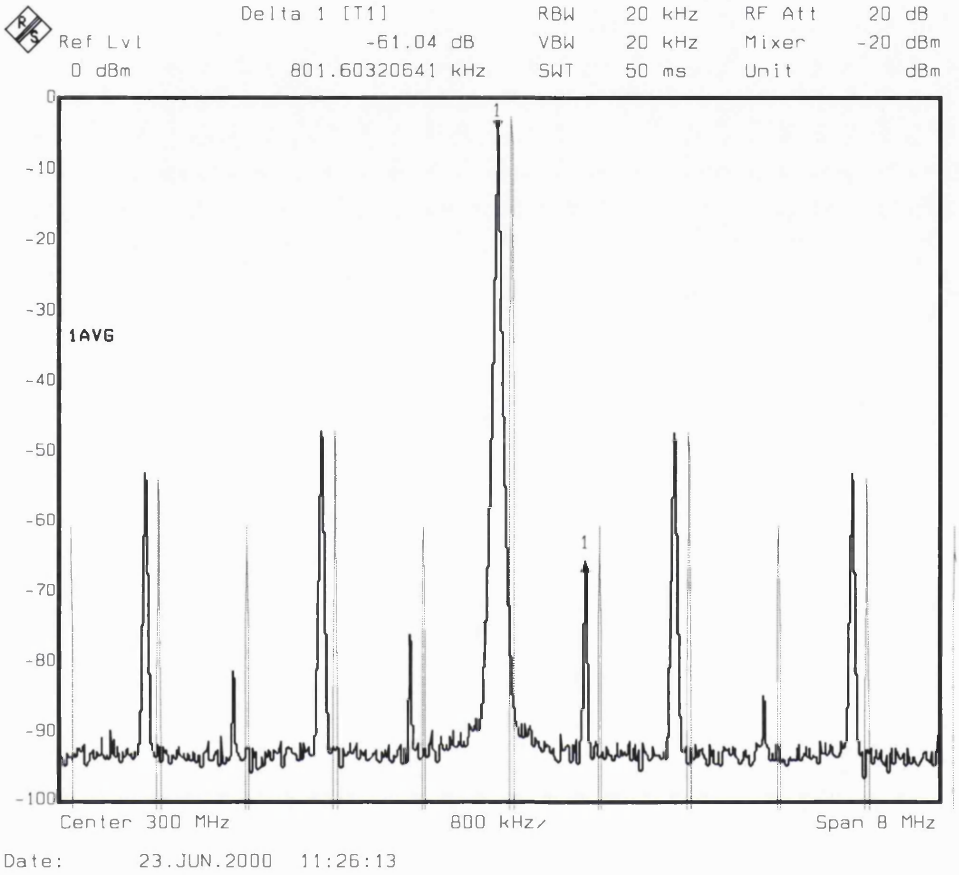


Figure 3.19 Plot showing correlation between simulated and measured DDS MSB accumulator output frequency spectrum

Similarly by emulating the digital process found within a DDS it is possible to model the signal truncation through the digital sine wave look up table before it is applied to the Digital to Analogue Converter at the DDS output.

3.5.2 The Sine Look Up Table

The sine look up table has always been one of the major bottlenecks in the core of a DDS, because of its potential to become very large, especially if the designer is attempting to reduce the truncation spurs caused by the digital signal passing through this element, by increasing the look up table resolution. In its simplest form, the size of the sine look up table equates to 2^L words[11], where L is the width of the digital word. Clearly with this potential to double in size for each single bit increase in resolution, this area of DDS evolution has been subject to a variety of alternative schemes each aimed at reducing the amount of memory required and also to increase the speed of operation.

The first obvious look up table compression scheme exploits the sine quarter wave symmetry to reduce the memory size by a factor of four, i.e. a 75% size reduction. To achieve this, the top two most significant data bits are used to invert and complement the sine converter's output data bits, thus enabling the same quarter wave look up table data to be mirrored into all four sine quadrants.

Phase Range	MSB	MSB-1	Function
$0 \leq \Phi < 90°$	0	0	Sin (Φ)
$90° \leq \Phi < 180°$	0	1	Sin (90°- Φ)
$180° \leq \Phi < 270°$	1	0	-Sin (Φ)
$270° \leq \Phi < 360°$	1	1	-Sin (90°- Φ)

To further compress the size of the sine look up table, almost all compression techniques start with the Taylor approximation for a sine wave:

$$\sin(a+b)=\sin(a)+b\cos(a)-\frac{b^2\sin(a)}{2}+\text{higher order terms} \qquad \text{Eq 3.14}$$

The simplest use of this equation is based on a straightforward digital interpretation of this expression, [64]. From this formula many derivatives have been offered, however one of the best known evolves from the Hutchinson approach, [38]. The Hutchinson method is based on splitting the incoming phase word to simultaneously drive both coarse and fine ROM look up tables, whose outputs are summed together to give the sine amplitude value. The Sunderland approach, [39], is a refinement to this technique that offers a further level of compression to the sine look up table. This method starts by using the identity:

[11] This assumes binary operation. The implementation of a BCD DDS is very much more complex.

$$\sin(\alpha+\beta+\gamma)=\sin(\alpha+\beta)\cos(\gamma)+\cos(\alpha)\cos(\beta)\sin(\gamma)-\sin(\alpha)\sin(\beta)\sin(\gamma) \qquad \text{Eq 3.15}$$

to give the approximation:

$$\sin(\alpha+\beta+\gamma)\cong\sin(\alpha+\beta)+\cos(\alpha)\sin(\gamma) \qquad \text{Eq 3.16}$$

With this result the incoming digital word is segmented into three parts; "α", "β" and "γ" so that sections "α" and "β" are fed into one smaller look up table, whilst segments "α" and "γ" are fed into another look up table. The outputs of both look up tables are summed to give the sine amplitude approximation, Figure 3.20. This further simplification has become very popular because of its simplicity and the level of compression it offers.

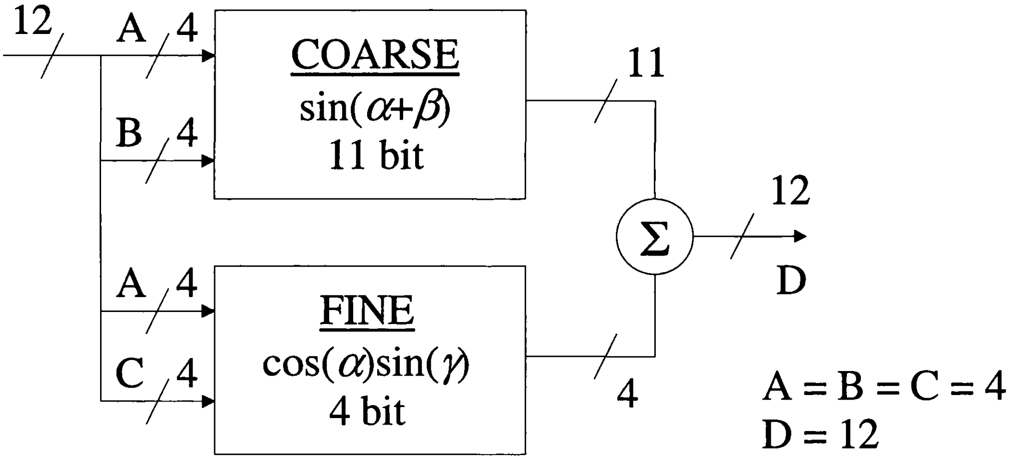


Figure 3.20 Sunderland sin look up table system, [39]

An alternative to the sine look up table, which has been adopted in a limited number of applications, is the Cordic algorithm proposed by Volder, as a means of solving navigational computation problems in 1960, [40]. This algorithm generates the sine amplitude values in real time, using a pair of shift registers followed by a couple of add and subtract circuits included to minimise the distortion products. The key advantage with this technique is the availability of quadrature components sometimes required by a DDS being used to drive IQ modulators.

3.5.3 The Digital to Analogue Converter

According to Goldberg, [65], the DAC performance is the least reliably modelled of all elements within the DDS structure. One example that supports this bold statement is shown in Figure 3.21, which shows the measured spurious performance of the AD9858, 10-bit DDS chip. With a 10-bit output DAC, the theoretical maximum spur levels should be 6×*D* dB below the carrier, where *D* is the DAC resolution, or 10 bits in the case of AD9858. This assumes the 10-bit DAC is capable of giving full 10-bit performance, which is generally not the case of DACs driven at higher clocking speeds, due to slew rate settling at the DAC output.

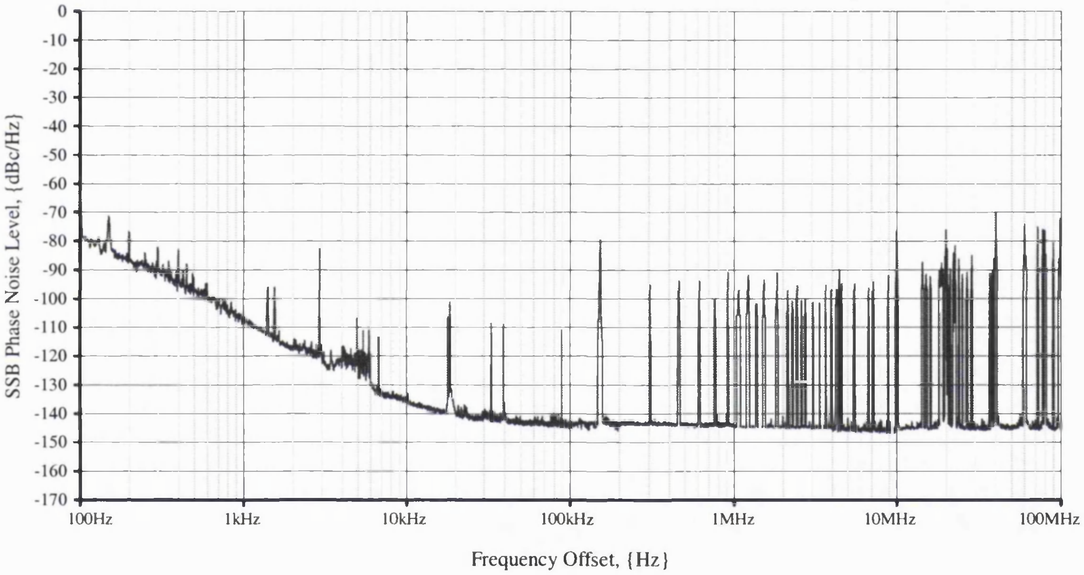


Figure 3.21 Measured phase noise and spurious performance of the AD9858 DDS

Figure 3.21, is a plot of the AD9858 driven by a clean 960MHz clock and asked to provide the frequency 439·996875MHz, chosen specifically to stress the DAC performance and generate a field of DAC related spurious products. Clearly there are many spurious products in this plot, but none of these products reach the –60dBc spurious level expected from an ideal 10-bit output DAC. No explanation is offered as to why this is the case, however Goldberg, [65], does believe this is an area requiring further investigation.

The spectrum of a waveform reconstructed by a DAC from digital data will have in addition to the expected frequency components, some DAC noise and distortion products. Distortion may be specified in terms of harmonic distortion, Spurious Free Dynamic Range (SFDR), intermodulation distortion, or all of these. Harmonic distortion is defined as the ratio of harmonics to fundamental when a theoretically pure Sine wave is reconstructed, and is the most common specification. SFDR is the ratio of the worst spur, (usually, but not necessarily always a harmonic of the fundamental) to the fundamental whilst Signal to Noise Ratio, (SNR), is the ratio of RMS signal to RMS noise power.

In practice many different phenomena are observed at the DAC output, some being application specific whilst others are manufacturer process dependent, Figure 3.22.

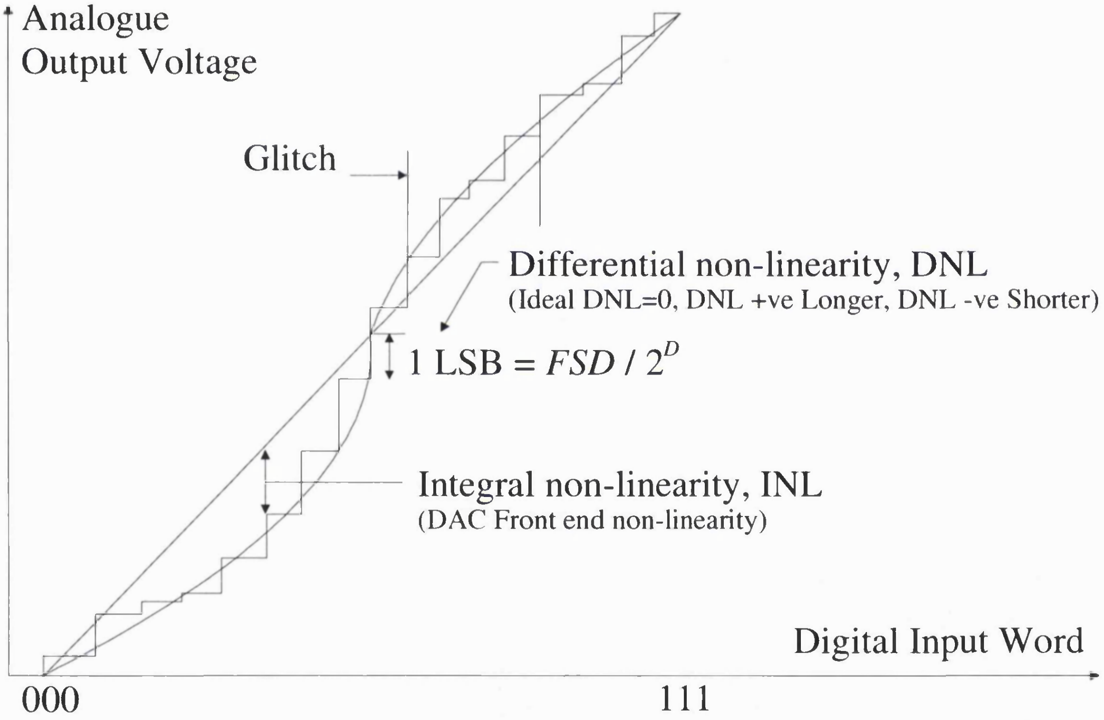

Analogue
Output Voltage
Glitch
Differential non-linearity, DNL
(Ideal DNL=0, DNL +ve Longer, DNL -ve Shorter)
1 LSB = $FSD / 2^D$
Integral non-linearity, INL
(DAC Front end non-linearity)
Digital Input Word
000
111

Figure 3.22 Diagram illustrating typical DAC non-linearity's in a full output sweep, [66]

Two common quantities often used to define the dynamic performance of a DAC are: Differential Non-Linearity, (DNL), which is defined as the variation in the spacing between adjacent analogue output values from the ideal single LSB value and Integral Non Linearity, (INL). INL is defined as the worst case variation in any of the analogue outputs with respect to the ideal straight line drawn through the end points. In reality, DNL induced SFDR errors will dominate a good DACs performance.

The theoretically ideal signal-to-noise ratio, (SNR), assuming the only noise source is quantization noise of a D-bit wide DAC with a full-scale voltage value, V_{FS} and LSB size, (l), has been shown to be:

$$\text{Signal}(RMS) = \frac{V_{FS}}{\sqrt{2}} = \frac{2^{D-1}l}{\sqrt{2}} \qquad \text{Noise}(RMS) = \frac{l}{\sqrt{12}}$$
$$\therefore \text{SNR} = \frac{\text{Signal}(RMS)}{\text{Noise}(RMS)} = \frac{l\,2^{D-1}/\sqrt{2}}{l/\sqrt{12}} = 2^{D-1}\sqrt{6} \qquad \text{Eq 3.17}$$
$$\text{SNR}(dB) = 20\text{Log}_{10}\left(2^{D-1}\sqrt{6}\right) = 6\cdot 02\times D + 1.76$$

Unfortunately, this equation rather optimistically assumes the expected SNR is directly related to the number of bits within the converter and ignores linearity, overshoot and other factors such as slew rate limiting. Slew rate limiting limits the maximum rise and fall times, effectively reducing the wanted signal levels especially at higher DAC operating frequencies. Thus a higher frequency DDS, will not necessarily show an improved performance, because slew rate limiting reduces the effective number of bits, (ENOB).

Eq 3.17 further ignores the differential non-linearity errors found in a DAC. Differential non-linearity errors give a measure of device mismatches causing deviations from code to code within the converter from the ideal step sizes and need accounting for, to better gauge the actual devices SNR performance. Similarly, the integral non-linearity of a given device will degrade the SNR and SFDR by raising the harmonic distortion components found at the DAC output, particularly the troublesome lower-order 2nd and 3rd harmonic components at larger signal levels. These code dependent glitches produce harmonics when the DAC is reconstructing a digitally generated sine wave as in a DDS. The mid-scale glitch occurs twice during a single cycle of a reconstructed sine wave, (at each mid scale crossing), and will therefore produce a second harmonic of the sine wave. It is the higher order harmonics of the sine wave which alias back into the Nyquist bandwidth (DC to $f_{Clk}/2$) and cannot be filtered, Figure 3.23.

It is worth noting that integral non-linearity errors can be derived numerically by integrating the differential non-linearity errors whilst differentiating the integral non-linearity errors, will allow the differential non-linearity errors to be determined.

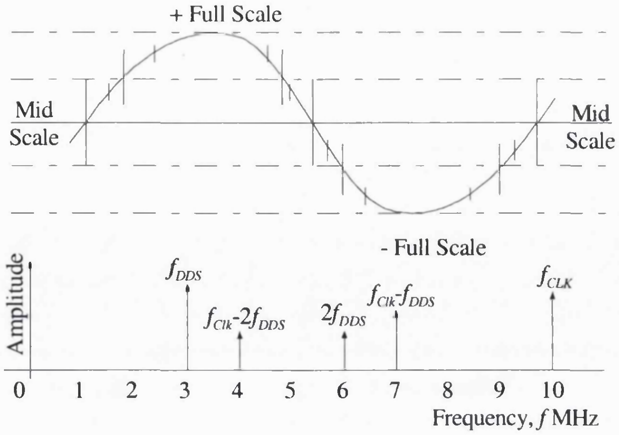


Figure 3.23 Effects of code dependent DAC glitches on spectral output, [66].

According to Moreland, [67], if the magnitude of the average DNL error is combined with the LSB thermal noise then a more representative prediction of a DAC's SNR can be derived that takes into account the additional noise caused by a non-ideal DNL as well as the thermal noise of the converter.

$$\text{SNR} = -20\text{Log}_{10}\sqrt{\left(\frac{2}{3}\sqrt{1+3\varepsilon^2}\cdot\frac{1}{2^D}\right)^2+\left(\frac{V_{thermal}}{2^D}\right)^2}\quad\{dB\} \qquad \text{Eq 3.18}$$

Where ε = average $|\text{DNL}|$ of the DAC and $V_{thermal}$ = noise in LSB.

Since a DAC behaves much as a mixer, the edge-to-edge variation found on the encode clock is convolved with the sampled spectrum. This reciprocal mixing of any noise on the clock signal shows up on the sampled spectrum as an elevated noise floor and hence reduced SNR. Including the effects of this encode clock edge jitter in terms of an RMS aperture jitter gives the more complete equation for predicting SNR as:

$$\text{SNR} = -20\text{Log}_{10}\sqrt{\left(\sqrt{2}\pi\cdot f_{in}\cdot t_{jitter}\right)^2+\left(\frac{2}{3}\sqrt{1+3\varepsilon^2}\cdot\frac{1}{2^D}\right)^2+\left(\frac{V_{thermal}}{2^D}\right)^2}\quad\{dB\} \qquad \text{Eq 3.19}$$

Where f_{in} is the input encode clock frequency, t_{jitter}, is the RMS aperture jitter, ε is the average |DNL| error, (in LSBs) and $V_{thermal}$ is the thermal noise in LSBs.

The operation of a DAC can be considered as a sample and hold of the digital input word, by the clock frequency, which in this case is the same clock driving the DDS accumulator, f_{CLK}. This allows the $n{\times}m$ intermodulation mathematics to be used for predicting the location of DAC created spurious products.

To summarise, the DAC output spectrum is a composite of the aliased image products, the DAC settling characteristics, the DDS truncation spurious products, DAC non-linearity, all shaped by the sinc function associated with the sample and hold at the DAC output, Figure 3.24.

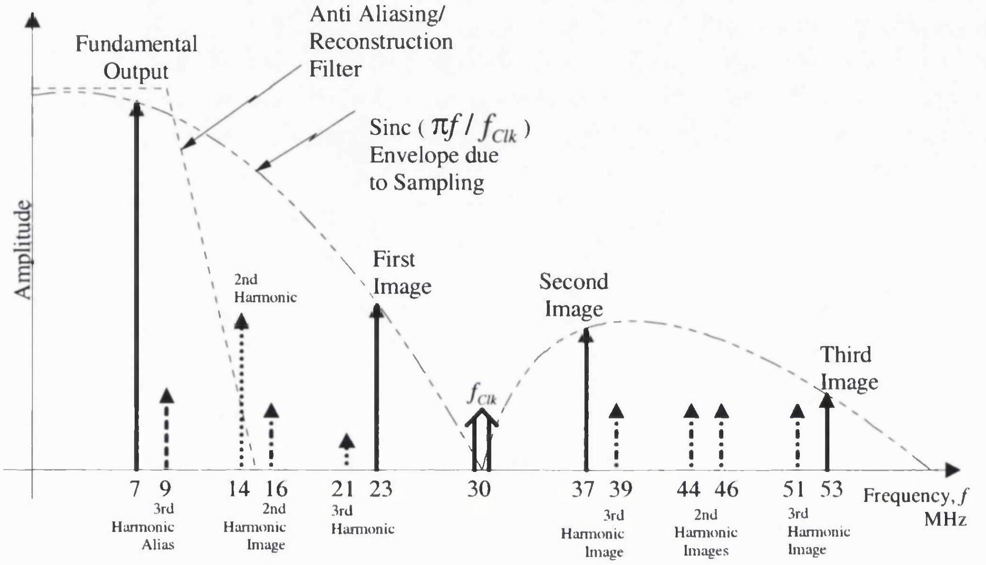


Figure 3.24 DDS output showing effects of sampling and DAC non linearity, [63]

The issues in DAC design for DDS are in both output glitch content and the overall linearity of the DAC. Clearly, glitches will add to the output spurious content, since they represent amplitude and phase terms. The DAC accuracy is more difficult to define, since the critical issue is the accuracy achieved in the available settling time. A DAC which is not fully settled will give a worse spurious performance than one which is; but the latter will be more difficult to manufacture and hence more expensive.

The DAC at the DDS output can be modelled as a mixer, with one port being the DAC analogue clock frequency whilst the other is the equivalent frequency of the digital word leaving the DDS core. Hence predictable intermodulation products result, (see appendix A3). The presence of these intermodulation products will further restrict the DDS useful range within the 40% DDS clock frequency set by a practical application of the Nyquist criteria. Clearly a higher frequency DDS coupled to a good quality, high resolution DAC would be invaluable, because of the increased useful fundamental frequency tuning range allowed by the increased separation between the DAC intermodulation products[12].

The phase noise performance of the DDS is ultimately limited by the edge-to-edge jitter performance of the input clock, f_{Clk}, although at the DDS output, the signal cycle to jitter performance will improve by the natural division process within the DDS digital accumulators down to the DAC noise floor. Using residual phase noise measurement techniques, [68], a DDS DAC noise floor close to –150dBc/Hz was measured, Figure 3.25. From the preceding text, there is little doubt the low level of this DAC noise floor in the presence of edge jitter, quantization noise, differential non-linearity errors, integral non-linearity errors and slew rate limitations will be a second level of noise that can largely be ignored when modelling a DDS element within a mixed signal synthesiser system.

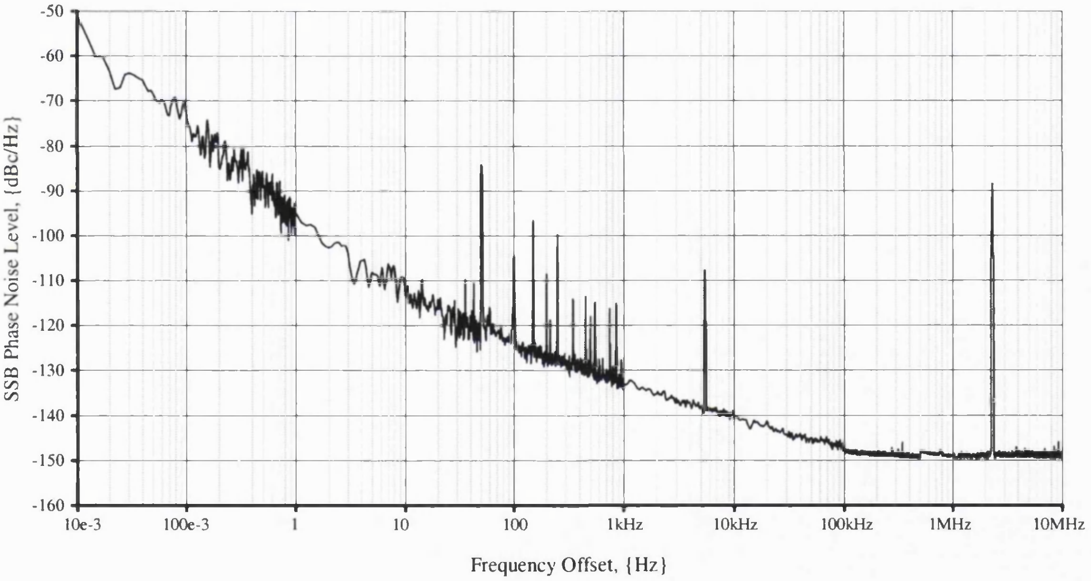


Figure 3.25 Measured DDS DAC noise floor

[12] Very often these intermodulation products give a "sea of spurs" which is often mistakenly identified as the noise floor of the DAC on a spectrum analyser.

3.6 Summary

Within this chapter, a condensed introduction of frequency synthesiser types has been offered. The different types of synthesiser were sub-divided into functional groups depending upon their method of operation, ease of use and phase coherence to the master reference signal. This chapter then proceeds to consider a few of the more pertinent functional blocks found in a synthesiser. In particular, the DDS and fractional-n frequency synthesisers have been discussed paying special attention to those performance limitations that limit their application in a demanding synthesiser application, when used in their basic form.

Although many of the characteristics presented here are secondary or even tertiary effects, they can be observed in some phase locked loops and therefore need considering when trying to stress one loop element to the detriment of all other elements within the loop.

The spurious performance of a DDS comes from a combination of factors within the DDS structure including the truncation of the accumulator output into the sine look up table, the sine look up table compression technique and the DAC linearity and glitch performance.

Within the synthesiser system described in chapter 7, a DDS is used to provide the necessary frequency resolution. Given the basic understanding of DDS generated spurious products offered here, the frequency planning of this synthesiser system is refined to manage the impact of these spurs.

4 MODELLING PLL PHASE NOISE

4.1 Introduction

This chapter focuses on modelling the phase noise of an analogue phase locked loop with the intention of building up a generic noise model, which can be used to analyse any phase locked loop. Although deceptively simple in its description and construction, the phase locked loop has proven surprisingly difficult to analyse precisely because of a PLL's inherent non-linearity. However, by making some approximations, the phase noise performance can be reliably modelled using a set of simple closed loop functions allowing the noise transfer function from any input node to the system output to be derived. Using this basic model, any multiple loop system can be modelled using a combination of these loop analyses.

When dealing with the summation of independent noise sources it is very difficult to conclusively prove that the mathematics match the results, because the noise sources combine to give a single output noise, burying the actual noise profiles attributable to each source. To gain confidence in the mathematics the trick is to try and remove, or at least minimise, all the other noise sources, using special PLL configurations to expose the noise of interest. This was the case for proving the mathematics of the loop filter thermal noise contribution. Based on the confidence thus derived from measurements, a complete PLL in a typical application is compared to the predicted noise.

Construction of this phase noise model begins with a set of noise profile models for each of the key noise determining elements within the phase locked loop, with the exception of the phase frequency detector, and then proceeds to combine them using a generic model for each noise injection point of a PLL. The phase frequency detector analysis is described in more detail in chapter 5, because of its particularly significant contribution to the in-band phase noise and the need to ensure it is properly modelled for all frequencies of operation.

The history and evolution of sampled data systems is extensively covered by Sklansky and Amit, [69, 70] and explained in texts by Ragazzini, Tuo, Kuo and Franklin, [71, 72, 73, 74]. However, the difference between the mathematical approach adopted in these texts and the analysis presented here, is that all PLL noise transfer functions must avoid any correlation between terms in the transfer function, for each applied noise source.

4.2 Mathematical Preliminaries

In order to successfully model PLL phase noise a number of assumptions were made:

1. The phase locked loop is both locked and stable and represents a linear, time invariant, causal system.
2. VCO noise modelling by Rice, [62] and PFD noise measurements in Figure 5.4 demonstrate each noise source possesses a Gaussian noise profile, with only the mean value of each noise across the base-band noise spectrum being used for these calculations.
3. All noise sources are stochastic[13] processes possessing a magnitude only.
4. Each noise source exhibits low levels of noise, and hence phase modulation. This equivalent phase modulation is less than 0·1 rads(RMS) and does not violate the small angle modulation criteria, appendix A1, assumed for SSB phase noise.
5. Thermal noise is the result of random charges in conductors. Since thermal noise is random, instantaneous values of the equivalent generator are not known, however for this analysis the assumption is that all the loop filter resistors are in thermal equilibrium, allowing the mean-squared value of the noise generator to be assumed.
6. The noise from each noise source will be un-correlated to all other noise sources allowing the laws of superposition to be applied, (appendix A2). From a simulation point of view, this implies that none of these terms interfere with one another.

[13] Stochastic is defined as having a random probability distribution or pattern that may be analysed statistically but may not be predicted precisely.

7. Convolution of the mathematics in the frequency domain is appropriate, because of the sampling function.

4.3 Phase Noise in a PLL

Modern integrated synthesiser chips rely on digital techniques to close the loop. Within the digital elements of these synthesiser chips, there exists a sampling process giving the system both sampled and continuous data. Deriving the linear loop transfer functions from each noise source to the PLL output is a straightforward translation from the noise injection source to the system output using transfer function analysis. For a sampled system the situation is quite different because the signal translation through the sampler is no longer a multiplication in the frequency domain but instead is an infinite summation, [75].

$$E^*(s)=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}E(s+jn\omega_s) \qquad \text{Eq 4.1}$$

A further complication is the injected noise signal is stochastic. To overcome the problem of signal injection into the closed sampling loop the techniques developed by Salzer, [75], will be employed. His analysis proceeded by breaking the loop at the sampler gate and treating the sampled signal as an independent signal, subject to the linear forward transfer function from the sampler output to the system output. Exercising this technique with due care to account for the stochastic nature of the noise-input signal, allows the noise transfer function to be derived. This is the foundation upon which the following noise analysis is constructed.

The following analysis aims to give the output quantity, Single Sideband, (SSB), phase noise, $\mathcal{L}(f_m)$, which is the ratio between the noise power in a sideband of bandwidth 1Hz, at a deviation $f_m = f_o\text{-}f_{offset}$ from the carrier frequency, compared with the total signal power of that carrier [76].

$$\mathcal{L}(f_m)=\frac{\text{power density in one phase noise modulated sideband, per Hz}}{\text{total signal power}}\ \{dBc/Hz\} \qquad \text{Eq 4.2}$$

Frequently, modern transceiver specifications express the total noise in terms of RMS phase jitter, or a derivative of this quantity. The RMS phase jitter can be determined from a given synthesiser's phase noise profile by integrating the SSB phase noise profile across a specified frequency offset, f_1 to f_2:

$$\phi_{jitter}^2 = 2\int_{f_1}^{f_2} \mathcal{L}(f_m)df_m \quad \{rad^2\} \qquad \text{Eq 4.3}$$

This allows the equivalent time jitter, T_{jitter}, to then be calculated at a given output frequency, f_o:

$$T_{jitter} = \frac{\phi_{jitter}}{2\pi f_o} \quad \{\sec\} \qquad \text{Eq 4.4}$$

It should be noted that this method of calculation actually measures the jitter standard deviation, $\sigma = \phi^2_{jitter}$, since the noise is assumed to be Gaussian distributed.

When analysing the phase noise transfer functions in sampled closed loop systems the relative position of the injected noise to the sampler determines the type of transfer function, which can be expected.

4.4 Sampled N-Divider and Reference Signal Noise at the PFD Input

The phase frequency detector has the responsibility of comparing the fed-back VCO signal to the reference-sampling signal. As such the PFD output noise is a composite noise built up from both these inputs because the PFD is unable to discriminate between either of these two independent noise profiles. All noise introduced at the phase frequency detector input is sampled by the edge triggered logic of the PFD, before being subject to the forward gain transfer characteristics of the loop.

Because both the sampled n-divider and the reference signal noise are compared at the input to the digital phase frequency detector, their noise transfer function to the PLL output is the same. Consequently only one sampling analysis is required to account for both phase noise transfer functions.

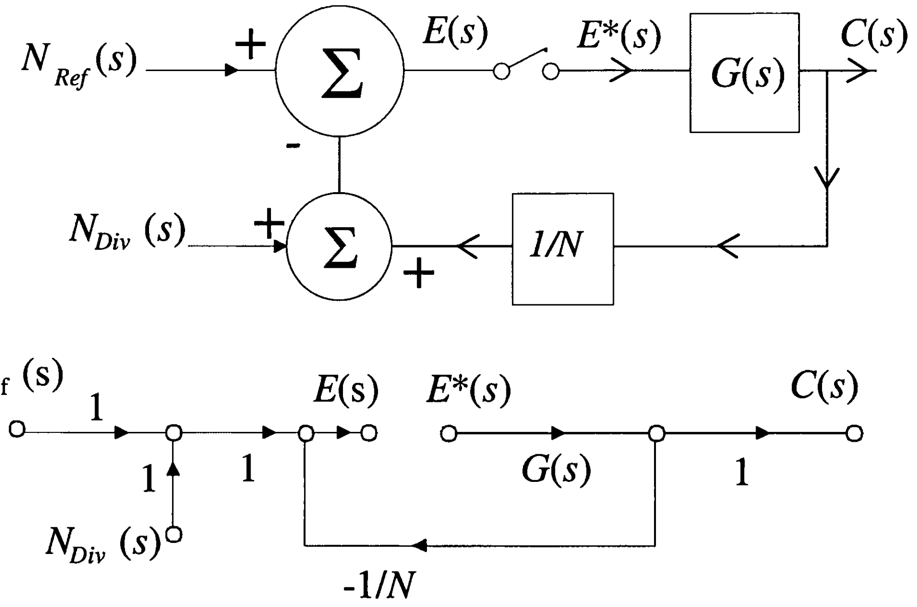


Figure 4.1 Reference and n-divider noise diagrams

With reference to Figure 4.1 and using the sampling notation of Kuo and Franklin [73, 74], the sampled signal is determined by taking the input signal to the sample gate, $E(s)$, and the sampler output $E^*(s)$ containing all the alias terms of the noise $N_{Ref}(s)$:

$$E(s) = N_{Ref}(s) - \frac{1}{N} \cdot G(s) \cdot E^*(s)$$

$$E^*(s) = \left[N_{Ref}(s) - \frac{1}{N} \cdot G(s) \cdot E^*(s) \right]^* \qquad \text{Eq 4.5}$$

$$\Rightarrow E^*(s) = N_{Ref}(s)^* - \left[\frac{1}{N} \cdot G(s) \cdot E^*(s) \right]^* = N_{Ref}(s)^* - E^*(s) \left[\frac{1}{N} \cdot G(s) \right]^*$$

Where the sampled loop term is

$$\left[\frac{1}{N} \cdot G(s) \right]^* = \frac{1}{N} \sum_{n=-\infty}^{\infty} G(s + jn\omega_s) \qquad \text{Eq 4.6}$$

Re-arranging Eq 4.5 to determine the sampled signal $E^*(s)$,

$$E^*(s) = \frac{N_{Ref}(s)^*}{1 + \frac{1}{N}[G(s)]^*} \qquad \text{Eq 4.7}$$

Then the system output response becomes,

$$C(s) = G(s) \cdot E^*(s) = \frac{G(s)}{1 + \frac{1}{N}[G(s)]^*} N_{Ref}(s)^* \qquad \text{Eq 4.8}$$

This expression is similar to the equivalent linear loop gain transfer function, with the exception that the loop gain is sampled.

$$C(s) = G(s) \cdot E(s) = \frac{G(s)}{1 + \frac{1}{N}[G(s)]} N_{Ref}(s) \qquad \text{Eq 4.9}$$

Because these noise sources appear before the sampler in the forward path of the loop, only one term was found in the argument of this expression. An example of a sampled signal applied to the PFD compared with the linear noise profile is given in Figure 4.2. This plot shows how sampling has raised the level of noise between each sampling offset frequency by aliasing the noise in these regions. This rise in noise is omitted from any linear analysis.

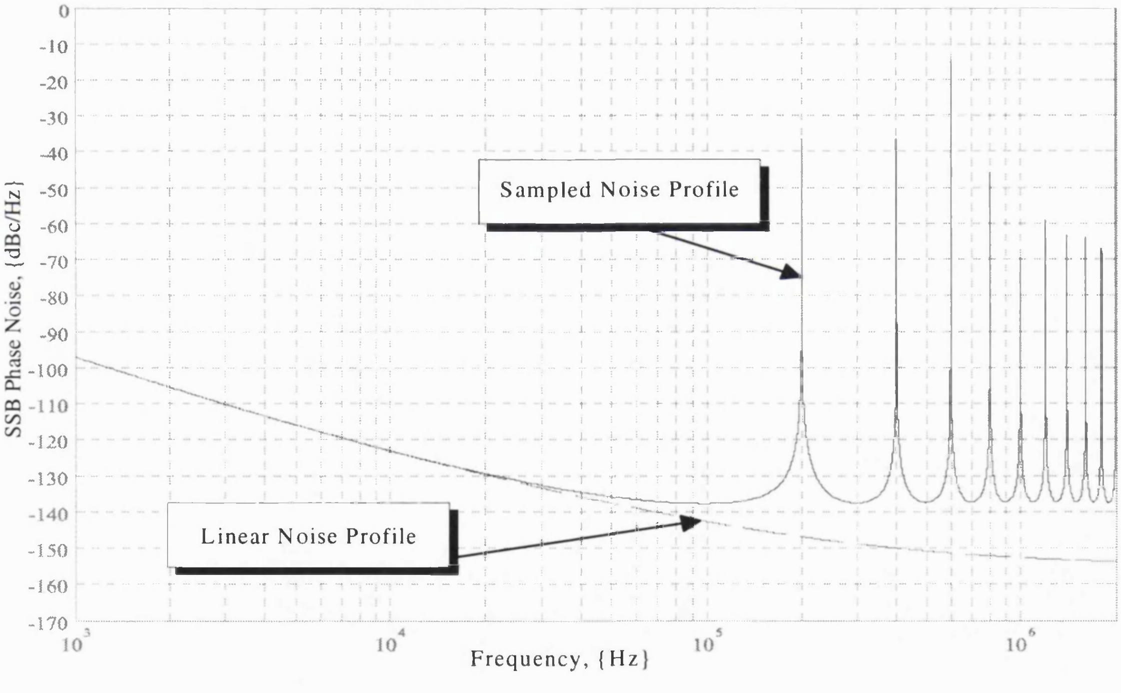


Figure 4.2 Sampled reference source phase noise profile

4.5 Thermal Noise in Loop Filter Components

The thermal noise contribution of the resistive components within the loop filter will modulate the VCO, distorting its noise profile. Thermal noise introduced within a receiver local oscillator source can often degrade the receiver's adjacent channel rejection performance through reciprocal mixing [77]. Reciprocal mixing is the parasitic process of mixing a strong adjacent channel signal with the local oscillator phase noise profile or spurious at an intermediate frequency offset from the carrier, to give an unwanted IF signal. This parasitic IF signal can de-sensitise the receiver or at least reduce the signal quality of the wanted signal. The standard approach to reducing the level of local oscillator spurious products is to use a higher order loop filter, which offers more attenuation at these spurious frequency offsets. However, due care is required because a higher order filter has the potential to raise the SSB phase noise level at these adjacent channel frequency offsets.

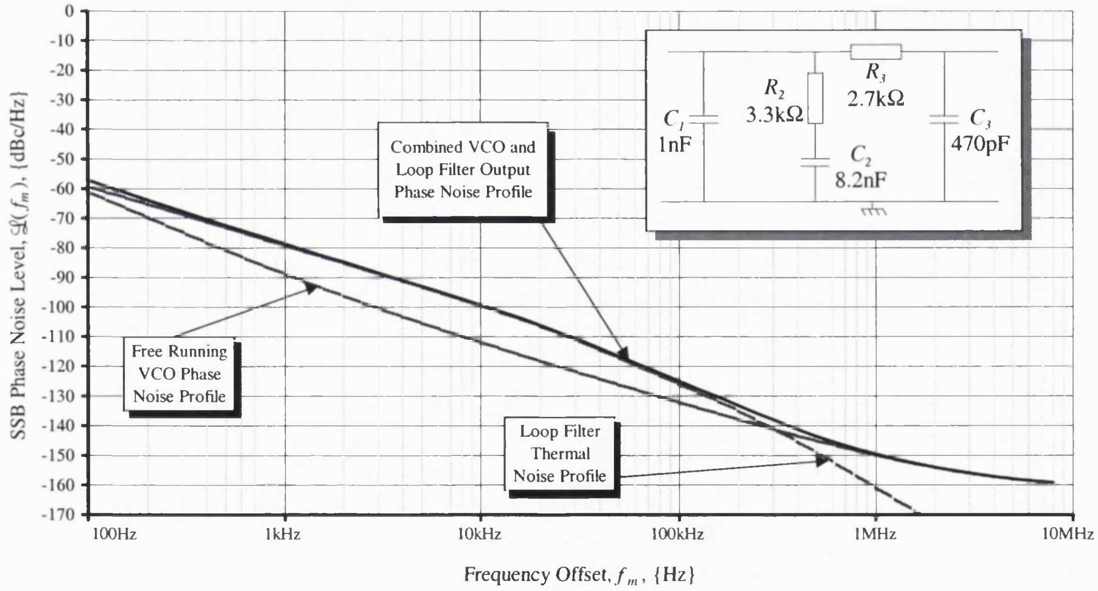


Figure 4.3 Typical loop filter thermal noise profiles

Figure 4.3 is a representative SSB phase noise plot of a fourth-order PLL filter indicating how loop filter thermal noise degrades the VCO noise profile in a PLL. The thermal noise raises the VCO noise level outside the loop bandwidth.

The model for thermal noise is white noise, effectively being flat for these frequencies.

When considering the reactive elements found in the loop filter the real part of the impedance or of the admittance is used in noise calculations. Since the real part of an impedance is in general a function of frequency, the equation for the voltage noise source (open-circuit mean-square voltage) may be written as:

$$\overline{V^2} = 4kT\int_{f_1}^{f_2} \mathrm{Re}(F(s))df \qquad \text{Eq 4.10}$$

Where *F*(*s*) is the loop filter transfer function. In terms of SSB phase noise being analysed here, the integration bandwidth, is 1Hz [76]. From Manassewitsch [78] the single sideband-to-carrier ratio in terms of peak phase deviation can be expressed as:

$$\mathcal{L}(f_m) \cong 20\mathrm{Log}_{10}\left(\frac{\theta_d}{2}\right) \left\{\frac{dBc}{Hz}\right\} \qquad \text{Eq 4.11}$$

And since the peak phase deviation is related to peak frequency deviation Δf_{peak} by:

$$\theta_d = \frac{\Delta f_{peak}}{f_m} \qquad \text{Eq 4.12}$$

Where f_m is the modulating frequency. Thus the SSB phase noise can be expressed in terms of Δf_{peak} as:

$$\mathcal{L}(f_m) \cong 20\mathrm{Log}_{10}\left(\frac{\Delta f_{peak}}{2f_m}\right) \text{ or } 20\mathrm{Log}_{10}\left(\frac{\Delta f_{rms}}{\sqrt{2}\,f_m}\right) \left\{\frac{dBc}{Hz}\right\} \qquad \text{Eq 4.13}$$

Given that the RMS frequency deviation seen at the VCO output is related to the open-circuit mean-square voltage (Eq 4.10) and the VCO gain, K_{VCO}, then the externally induced phase noise is

$$\mathcal{L}(f_m) \cong 20\mathrm{Log}_{10}\left(\frac{\overline{V^2}\cdot K_{VCO}}{\sqrt{2}\,f_m}\right) \left\{\frac{dBc}{Hz}\right\} \qquad \text{Eq 4.14}$$

To analyse the thermal contribution of the loop filter of a phase locked loop, the loop filter is taken in isolation of the remainder of the loop to give a thermal noise voltage, which drives the VCO, Figure 4.4.

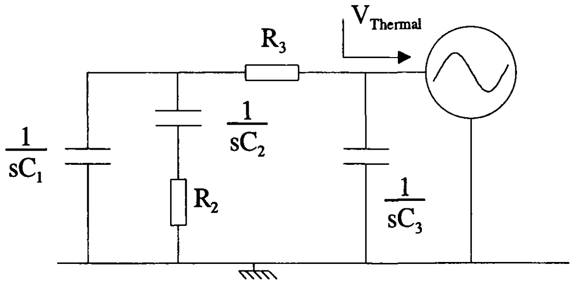


Figure 4.4 Loop filter thermal noise model

Giving the real part of the loop filter as:

$$\mathrm{Re}(Filter(s)) = \frac{R_3(C_1+C_2)^2 + R_2C_2^2 + \omega^2 R_3 (R_2C_1C_2)^2}{\begin{array}{l}(R_2R_3C_1C_2C_3)^2\omega^4 + \\ \left[((C_1+C_2)C_3R_3)^2 + \left(2R_2C_2^2C_3^2\right)R_3 + (R_2C_2(C_1+C_3))^2\right]\omega^2 + \\ (C_1+C_2+C_3)^2\end{array}} \qquad \text{Eq 4.15}$$

Buried within Eq 4.15 is the series resistor R_3, found only in the fourth-order filter configuration, whose presence will increase the level of loop filter thermal noise as an unwanted effect of increasing the loop's out-of-band attenuation. Thus for any optimised synthesiser design, the level of increased thermal noise contributed by R_3 needs balancing against the added attenuation to sampling spurs that this additional pole provides, for the same phase margin.

The implications of loop filter thermal noise are best seen when specifying the amount of additional attenuation required in the loop filter. For example, Figure 4.5 indicates how the loop filter values, especially series resistor R_3, vary with increasing loop filter attenuation[14] at the first sampling frequency offset.

[14] Note that the preparation of this graph assumes the separation between the loop bandwidth and sampling frequency ratio is large, so the additional attenuation does not impact the closed loop bandwidth.

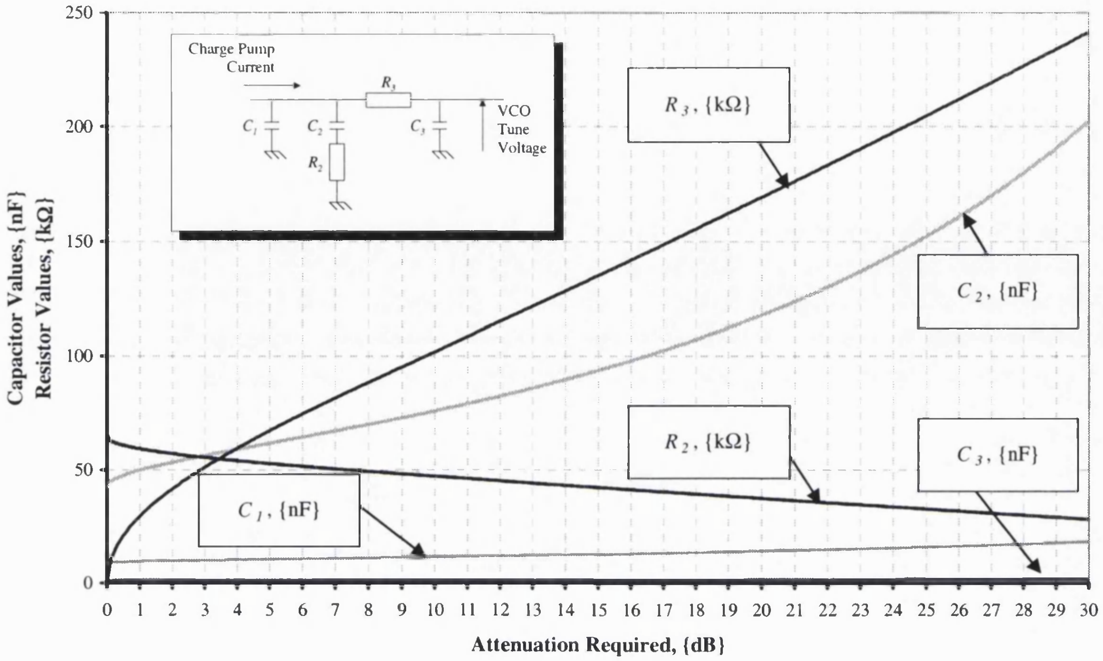


Figure 4.5 Typical variation of loop filter component values for increasing attenuation requirements

Texts focusing on modelling PLL phase noise indicate that the nulls found in the PLL SSB phase noise profile will fall to the VCO noise level and will very often neglect the thermal noise contribution of the loop filter components. Figure 4.6 is a SSB phase plot taken by the author from a test synthesiser specifically designed to view the null floors compared with the free running VCO phase noise profile used in this experiment. This experiment used a standard synthesiser chip with a modified feedback path that used a clean signal source to mix the VCO output signal down to the minimal n-divider ratios available within the test synthesiser chip. In addition, the loop bandwidth was raised to an unusually large value within the confines of loop stability. This measurement set-up reduces the effects of n-divider multiplication on other noise sources within the loop and accentuates the width, depth and number of nulls produced so their noise floors become visible.

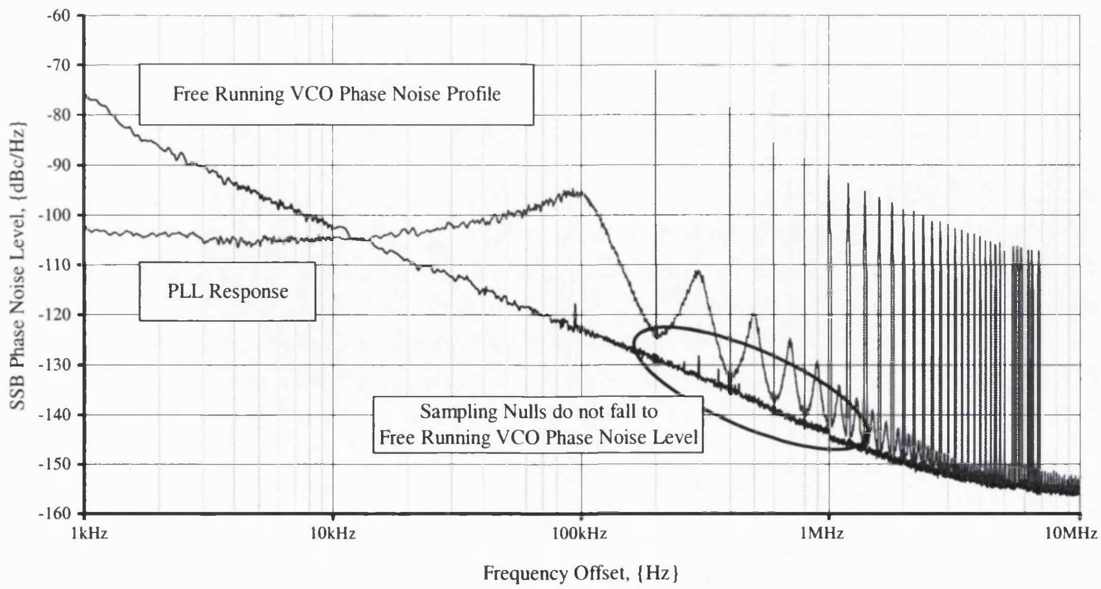


Figure 4.6 Overlay of two phase noise plots showing null depths

With reference to Figure 4.6, it is apparent that the sampling nulls do not fall to the expected free running VCO noise profile, but instead they initially fall several dBs short and only approach the VCO noise profile well beyond the loop bandwidth. Referring to Figure 4.3, it can be seen that the VCO noise profile is modified by the loop filter thermal noise modulating the VCO and it is this new level of VCO noise that defines the depth of these sampling nulls. No known reports have been found, which offer alternative explanations that clarify why the floor of the sampling nulls do not fall to the VCO noise profile.

Although thermal noise is the prime focus within this investigation, other noise introduced by inadequate power supply regulation to the VCO, will also raise the level of noise in the same manner.

4.6 Sampled VCO Noise

The objective of this section is to develop a set of noise transfer functions from each noise injection point into a phase locked loop, which accounts for the effects of the sampling action of the PFD. Very often a linear analysis will suffice, however as the loop bandwidth approaches the sampling frequency, the aliasing of complementary noise products significantly increases the total noise level and it is for this reason that the following analysis is presented here.

Modelling the VCO noise is a challenging task because the straightforward mathematics used for analysing the effect of an injected signal into a closed loop sampled system yields an expression, which contains several auto-correlated terms. These terms need sifting to ensure that the stochastic VCO noise, which possesses magnitude only, should not find itself mistakenly repeated.

At the end of this analysis it will be shown that the VCO noise transfer expression can be simplified without incurring significant inaccuracy, however the purpose of this explanation is to demonstrate the mathematics so they can be re-applied for other injected noise sources in the PLL forward path after the sampling gate. The resulting transfer formula can then easily be modified for all the post sampler noise sources injected in the PLL forward path, using the summing node shift theorems, [72].

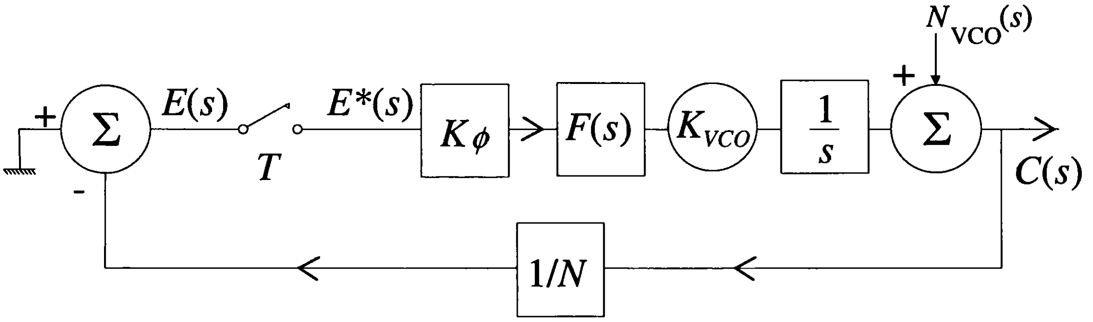


Figure 4.7 Model of VCO noise injected into closed loop PLL

Based on Figure 4.7, the linear closed loop response to the fundamental, (non-aliased), VCO noise component can be written by inspection:

$$C(s)|_{n=0} = \frac{N_{VCO}(s)}{1+\frac{1}{N}G(s)} \qquad \text{Eq 4.16}$$

Where $G(s) = K_\phi \cdot F(s) \cdot K_{vco}/s$, and is the forward path gain of the PLL. To account for each of the complementary, (alias), products, (i.e. $n \neq 0$ terms), Salzer's technique , [75], of expressing a signal as a separate entity at the sampler output is exercised. Using ordinary closed loop algebraic manipulations to translate the VCO noise to an equivalent sampled error signal, $E^*(s)$ at the sampler output proceeds as follows:

$$E(s) = -\frac{1}{N}N_{VCO}(s) - \frac{1}{N}G(s)E^*(s) \qquad \text{Eq 4.17}$$

The translation of this expression into its sampled equivalent form requires care should be taken to ensure there is no duplication of any sampled signal, which has already been accounted for.

$$\therefore E^*(s)|_{n\neq 0} = -\left[\frac{1}{N}N_{VCO}(s)\right]^* - \left[\frac{1}{N}G(s)E^*(s)\right]^*$$

$$E^*(s)|_{n\neq 0} = \frac{-\left[\frac{1}{N}N_{VCO}(s)\right]^*}{1+\left[\frac{1}{N}G(s)\right]^*} \qquad \text{Eq 4.18}$$

The translation from $[E(s)^*]^* \Rightarrow [E(s)]^*$ contained within Eq 4.18, is valid when a sampled signal is re-sampled at the same sampling rate, [74]. The terms N, K_ϕ and K_{vco} are assumed to be frequency independent and remain constant irrespective of sampling effects. At this point, the presence of each alias product is considered in isolation and treated as independent signals to avoid introducing correlated noise components into the mathematics. The non-aliased term, $n = 0$ is excluded because it has already been accounted for in Eq 4.16. As a result, the fundamental of each of these sampled terms is:

$$E^*(s)|_{n\neq 0} = \frac{\frac{1}{N}\sum_{n=-\infty}^{-1} N_{VCO}(s+jn\omega_s)}{1+\frac{1}{N}\sum_{n=-\infty}^{-1} G(s+jn\omega_s)} + \frac{\frac{1}{N}\sum_{n=1}^{\infty} N_{VCO}(s+jn\omega_s)}{1+\frac{1}{N}\sum_{n=1}^{\infty} G(s+jn\omega_s)} \qquad \text{Eq 4.19}$$

For mathematical convenience and knowing that a sampled signal when re-sampled at the same sampling rate, faithfully reproduces itself, [74], then this error signal, $E^*(s)$, can be moved in front of the sampler, Figure 4.8. The term $E'^*(s)$ is introduced to account for the sampler output which includes all these fundamental sampled components and their alias products:

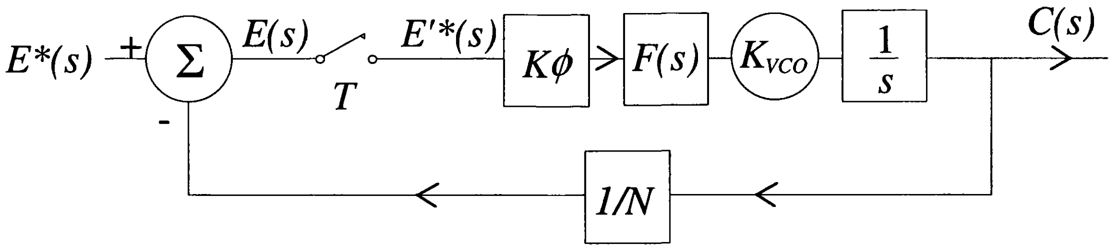


Figure 4.8 Model for aliased VCO noise

Proceeding with a closed loop analysis gives:

$$E(s) = -E'^{*}(s)G(s)\frac{1}{N} + E^{*}(s) \qquad \text{Eq 4.20}$$

With the sampling signal, $E^{*}(s)$, Eq 4.19, being re-sampled at the same rate, then:

$$\begin{aligned} E'^{*}(s) &= -\left[G(s)\frac{1}{N}\right]^{*} E'^{*}(s) + E^{*}(s) \\ \therefore E'^{*}(s) &= \frac{E^{*}(s)}{1+\left[G(s)\frac{1}{N}\right]^{*}} \end{aligned} \qquad \text{Eq 4.21}$$

Therefore the closed loop output is:

$$C(s) = G(s)E'^{*}(s) = \frac{G(s)E^{*}(s)}{1+\left[G(s)\frac{1}{N}\right]^{*}} \qquad \text{Eq 4.22}$$

Removing frequency independent terms and noting that this sampling term in the denominator needs to account for all the subsequent re-sampling around the loop gives:

$$C(s) = \frac{K\phi F(s)\frac{Kv}{s}}{1+\frac{1}{N}\sum_{n=-\infty}^{\infty} G(s + jn\omega_s)} E^{*}(s) \qquad \text{Eq 4.23}$$

Substituting Eq 4.19 into Eq 4.23 and including the non-aliased term accounted for in Eq 4.16, provides the complete VCO noise expression:

$$
\begin{aligned}
C(s) = {} & \frac{N_{VCO}(s)}{1+\frac{1}{N}G(s)} + \\
& \frac{G(s)}{1+\frac{1}{N}\sum_{n=-\infty}^{\infty}G(s+jn\omega_s)} \cdot \frac{\frac{1}{N}\sum_{n=-\infty}^{-1}N_{VCO}(s+jn\omega_s)}{1+\frac{1}{N}\sum_{n=-\infty}^{-1}G(s+jn\omega_s)} + \\
& \frac{G(s)}{1+\frac{1}{N}\sum_{n=-\infty}^{\infty}G(s+jn\omega_s)} \cdot \frac{\frac{1}{N}\sum_{n=1}^{\infty}N_{VCO}(s+jn\omega_s)}{1+\frac{1}{N}\sum_{n=1}^{\infty}G(s+jn\omega_s)}
\end{aligned}
\qquad \text{Eq 4.24}
$$

Eq 4.24 contains the complete VCO noise transfer including the effects of sampling. The first expression accounts for the linear transfer function of the fundamental VCO noise component to the system output as would be derived using a linear analysis. Each of the two further terms account for the alias products resulting from multiple passes around the loop through the sampling gate. Their transfer functions differ from that of the linear expression because they are now considered independent noise sources injected at the sampler and consequently possess the loop transfer function from the VCO to the sampler input concatenated with the transfer function from the sampler output to PLL output. Reviewing this equation, it must be appreciated that each sample of the VCO signal appears once, (hence there is no cross correlation), and that the second and third transfer functions account for the alias products only.

In summary, this analysis proceeded by first accounting for the primary linear loop analysis of the VCO noise, followed by a piecewise summation of each independent alias product of this signal at the error output of the sampling gate, using Salzer's sampled loop analysis method, [75].

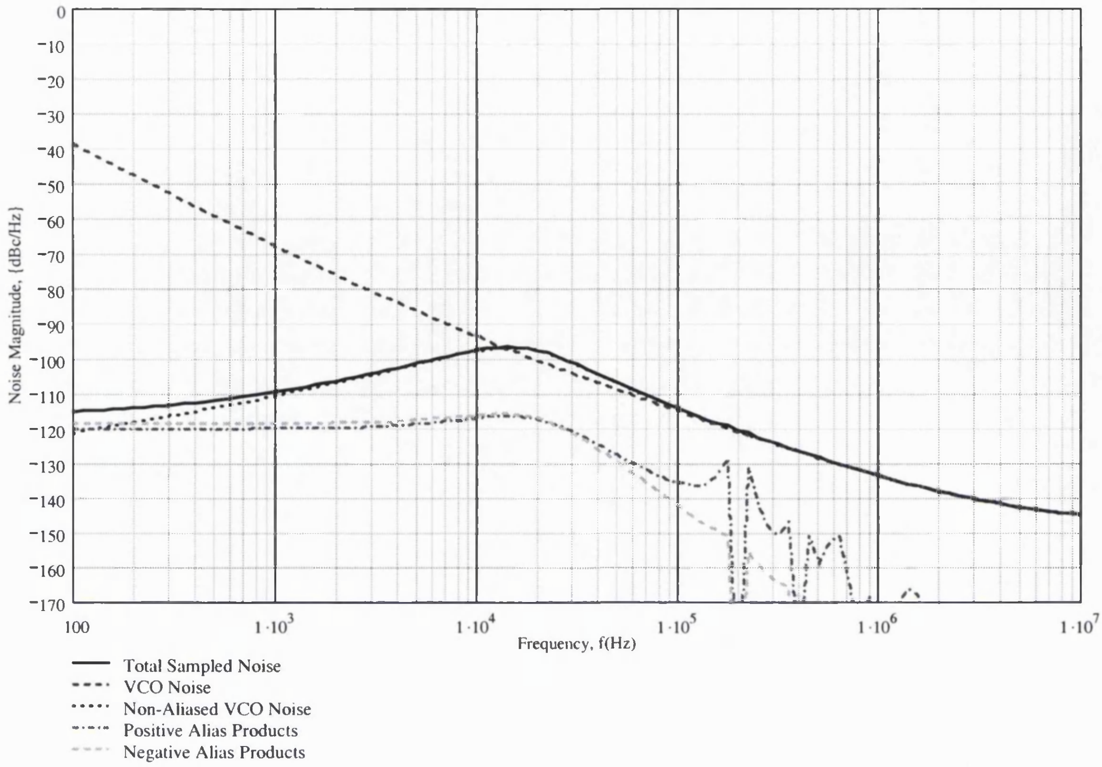


Figure 4.9 Graphical description of VCO noise terms described in Eq 4.24

The VCO correction for sampling contained in these two alias expressions is very small and only has significance around the sampling frequencies when $G^*(s)$ becomes large, Figure 4.9. Usually the maximum 10:1 ratio of sampling frequency to loop bandwidth guarantees the low pass filtering of both the loop filter and the VCO sufficiently attenuates the noise contribution of these second and higher order noise products so they can be ignored[15]. This "swamping" of the secondary component spectra is also noted by Underhill *et al.*, [79]. With this assumption, Eq 4.24 can be replaced by the simpler linear VCO noise transfer function originally given in Eq 4.16.

Figure 4.9 presents each of the aliased and non-aliased sampled VCO noise components given in Eq 4.24. Together with the laboratory observations given in Figure 4.6 and the analysis of loop filter thermal noise presented in section 4.5, this substantiates the claim that a linear analysis will suffice in the case of the VCO noise only, because the VCO

[15] Within this analysis VCO spurious are not considered, however for high level spurious and parasitic noise introduced at the VCO input, the loop would inadequately attenuate this excess noise and these additional alias terms will need including in the phase noise analysis to account for these products, if so desired.

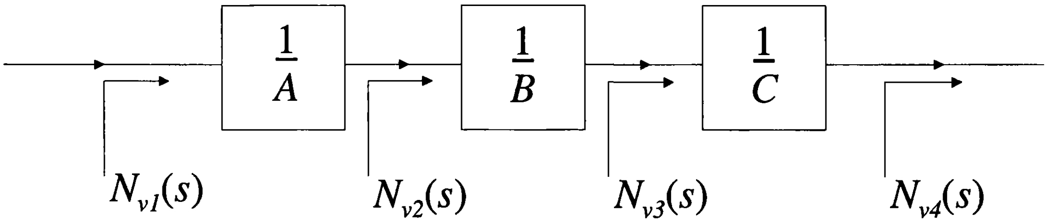
noise will be lost underneath the loop filter thermal noise. The presence of the aliased VCO noise is insignificant compared with the thermal noise expected at these frequency offsets, even when the loop bandwidth approaches the PFD sampling frequency.

4.7 N Dividers

The n-dividers add an interesting dimension to the analysis of phase locked loops because they themselves are samplers and depending upon whether the PLL is an integer-n or a fractional-n system, alter the sampled loop analysis. When considering an integer-n PLL, a complete mathematical analysis would result in a multi-rate analysis, whilst a fractional-n PLL analysis would become a non-uniform sampled system analysis, [80]. Multi-sampled and multi-rate loop analyses are beyond the scope of this thesis. Within a PLL only the phase is of concern, whilst the effective n-divider delay equivalent to one PFD sampling period allows the closed loop PLL to be broken for a sampling noise analysis, because of the de-correlation that this delay introduces.

For example, the feedback dividers sample the VCO input signal at their input and are sensitive to disturbances only at the logic level crossings of the input signals. Furthermore, the n-division process normally comprises a high frequency dual, (or multi), modulus digital divider followed by a set of lower frequency down counters. Any noise introduced in the logic interface between the high frequency dividers and the lower frequency down counters will not be subject to the full $20\text{Log}_{10}(N)$ reduction offered by the full n-divider. Instead only a limited noise reduction will be available depending upon the subsequent divider ratios, Figure 4.10. This interface between the dual modulus pre-scaler and the digital down counters is usually the bipolar to CMOS interface introduced by synthesiser manufacturers, in an effort to minimise the overall DC power consumption within their synthesiser chip.

The process of n-division noise improvement in a PLL is virtually the opposite of noise figure understood within radio receiver design, in which any attenuation found at the receiver input provides the base line noise figure that is further degraded by the following receiver amplifier, filter and mixer stages.



$$\text{Noise Power}_{out}(s) = \left|N_{v1}(s)\frac{1}{A}\frac{1}{B}\frac{1}{C}\right|^2 + \left|N_{v2}(s)\frac{1}{B}\frac{1}{C}\right|^2 + \left|N_{v3}(s)\frac{1}{C}\right|^2 + \left|N_{v4}(s)\right|^2$$

Figure 4.10 Diagram indicating individual noise source variation through an n-divider chain

4.8 Sampling Spurious Products

Sampling spurii are a mixture of the leakage around or through the loop filter components and the PFD finite propagation delay.

It is very difficult to accurately gauge the parasitic leakage found around the loop filter causing the charge held within the loop filter components to leak away between sample updates. These models have been discussed in section 3.4.2 and amount to a charge leakage caused by the finite charge pump output resistance as well as the VCO finite input impedance and any conductance across the printed circuit board. Their net effect is to introduce a small saw-tooth like waveform at the VCO input which frequency modulates the VCO and cannot be corrected for by the loop natural feedback action, because they fall outside the loop bandwidth.

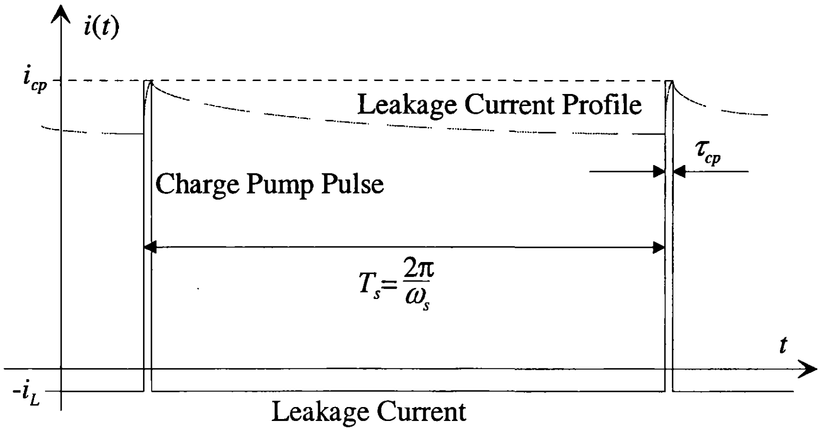


Figure 4.11 VCO Modulation Waveform Caused by Leakage Currents Around the Loop Filter

With reference to Figure 4.11, the resulting waveform across the loop filter comprises a short duration current pulse and a much longer exponential decay covering the remainder of the charge pump tri-state period between sample updates.

Since the Fourier frequency components of the short duration charge pump current pulse dominate then the longer exponential decay can be ignored. To determine the phase error this modulated signal introduces into the loop, consider the unmodulated pulse train

$$i(t) = i_{cp}\left[\frac{\tau_{cp}}{T_s} + \frac{2\tau_{cp}}{T_s}\sum_{n=1}^{\infty}\frac{\sin(n\omega_s\tau_{cp}/2)}{n\omega_s\tau_{cp}/2}\cos(n\omega_s t)\right] \qquad \text{Eq 4.25}$$

In Eq 4.25 ω_s is the sample update frequency, τ_{cp} is the charge pump pulse duration period, and T_s is the sampling update period. In appendix A4 it is shown that if Eq 4.25 was Pulse Width Modulated, (PWM), then the charge pump current signal passing through the loop filter, approximates to:

$$i_{PWM}(t) \cong i_{cp}\left[\frac{\tau_{cp}}{T_s} + \frac{\omega_s\tau_{cp}}{\pi}\cos(\omega_s t)\right] \qquad \text{Eq 4.26}$$

Taking the loop filter impulse response [$f(t)$], this gives a VCO modulating signal:

$$v_{PWM}(t) \cong i_{cp}\left[\frac{\tau_{cp}}{T_s} + \frac{\omega_s\tau_{cp}}{\pi}\cos(\omega_s t)\right]\cdot[f(t)] \qquad \text{Eq 4.27}$$

Within frequency synthesis it is desirable to minimise these sampling spurs to very low levels. To properly account[16] for this low-level sampling signal modulating the VCO, a full Fourier series expansion using $J_n(\beta)$ as the nth-order Bessel function of the first kind gives:

[16] As an approximation for small modulation indices and using FM modulation theory, this Voltage modulates the VCO with its inherent gain, K_{vco}, to give sidebands 6dB below the amplitude of the modulating signal: $v(t)=\cos(\omega_O t)+\frac{1}{2}\beta\cos((\omega_O+\omega_S)t)-\frac{1}{2}\beta\cos((\omega_O-\omega_S)t)$

$$\begin{aligned} v(t) = J_0(\beta)\cos(\omega_o t) - A_1 J_1(\beta)[\cos(\omega_o - \omega_s)t - \cos(\omega_o + \omega_s)t] \\ +A_2 J_2(\beta)[\cos(\omega_o - 2\omega_s)t - \cos(\omega_o + 2\omega_s)t] \\ -A_3 J_3(\beta)[\cos(\omega_o - 3\omega_s)t - \cos(\omega_o + 3\omega_s)t] \\ +..... \end{aligned} \qquad \text{Eq 4.28}$$

With the amplitude of each sampling harmonic *n*, seen at the VCO output being:

$$\text{Spur Harmonic}_n(s) = \frac{n\omega_s \tau_{cp}}{\pi} \cdot i_{cp} \cdot F(n \cdot s) \cdot K_{VCO} \qquad \text{Eq 4.29}$$

With no further loop correction of these sampling spurs necessary because the loop transfer function is high pass from the VCO output, this expression provides a good approximation for each sample spur harmonic observed at the PLL's VCO output.

A typical sampling reference signal applied at the PFD input, often possesses a combination of its own residual 1/*f* noise and the more dominant r-divider noise floor preceding the PFD input, (Figure 3.17). Therefore, with such low noise profiles so close to the carrier, the effects of modulating these sampling spurs by an already low-noise sampling signal, will be masked by other noise signals present within the synthesiser and can safely be ignored. When a typical synthesiser output is viewed on a spectrum analyser these sampling spurs may appear to possess a phase noise profile, however as described by Underhill *et al.*, [79], this apparent phase noise skirt is a product of the spectrum analyser's own Gaussian IF filter shape.

4.9 Summary Of Loop Response To Each Noise Input

Using the mathematics given in section 4.6 a set of transfer functions, Eq 4.30 to Eq 4.34, can be developed from each node of the closed loop PLL to the output, whilst taking care to eliminate all cross correlated components found in each transfer function.

Figure 4.12 is a pictorial overview of typical noise shaping for each noise injection point within a PLL, to the PLL output, for both the linear and sampled components in Eq 4.30 to Eq 4.34. However, the very nature of the following equations, Eq 4.30 to Eq 4.34 requires noise components to be buried within them, thus Figure 4.12 uses a typical set of PLL parameters for this illustration. To avoid distorting these noise gain responses, the typical noise profiles found at each of these injection points have been omitted.

$$\phi_{PFD}(s)=\left|\frac{N_{PFD}(s)\cdot F(s)\cdot\frac{K_{VCO}}{s}}{1+\frac{1}{N}G(s)}\right|+\left|\frac{\left[\sum_{n=-\infty}^{n=-1}N_{PFD}(s-jn\omega_s)\cdot F(s-jn\omega_s)\cdot\frac{K_{VCO}}{(s-jn\omega_s)}\right]\cdot\frac{1}{N}\cdot G(s)}{1+\frac{1}{N}\cdot\sum_{n=-\infty}^{n=-1}G(s-jn\omega_s)}\right|+\left|\frac{\left[\sum_{n=1}^{n=\infty}N_{PFD}(s-jn\omega_s)\cdot F(s-jn\omega_s)\cdot\frac{K_{VCO}}{(s-jn\omega_s)}\right]\cdot\frac{1}{N}\cdot G(s)}{1+\frac{1}{N}\cdot\sum_{n=1}^{n=\infty}G(s-jn\omega_s)}\right| \qquad \text{Eq 4.30}$$

PFD Output to PLL Output

$$\phi_{Filt}(s)=\left|\frac{N_{Filt}(s)\cdot\frac{K_{VCO}}{s}}{1+\frac{1}{N}G(s)}\right|+\left|\frac{\left[\sum_{n=-\infty}^{n=-1}N_{Filt}(s-jn\omega_s)\cdot\frac{K_{VCO}}{(s-jn\omega_s)}\right]\cdot\frac{1}{N}\cdot G(s)}{1+\frac{1}{N}\cdot\sum_{n=-\infty}^{n=-1}G(s-jn\omega_s)}\right|+\left|\frac{\left[\sum_{n=1}^{n=\infty}N_{Filt}(s-jn\omega_s)\cdot\frac{K_{VCO}}{(s-jn\omega_s)}\right]\cdot\frac{1}{N}\cdot G(s)}{1+\frac{1}{N}\cdot\sum_{n=1}^{n=\infty}G(s-jn\omega_s)}\right| \qquad \text{Eq 4.31}$$

Filter Output to PLL Output

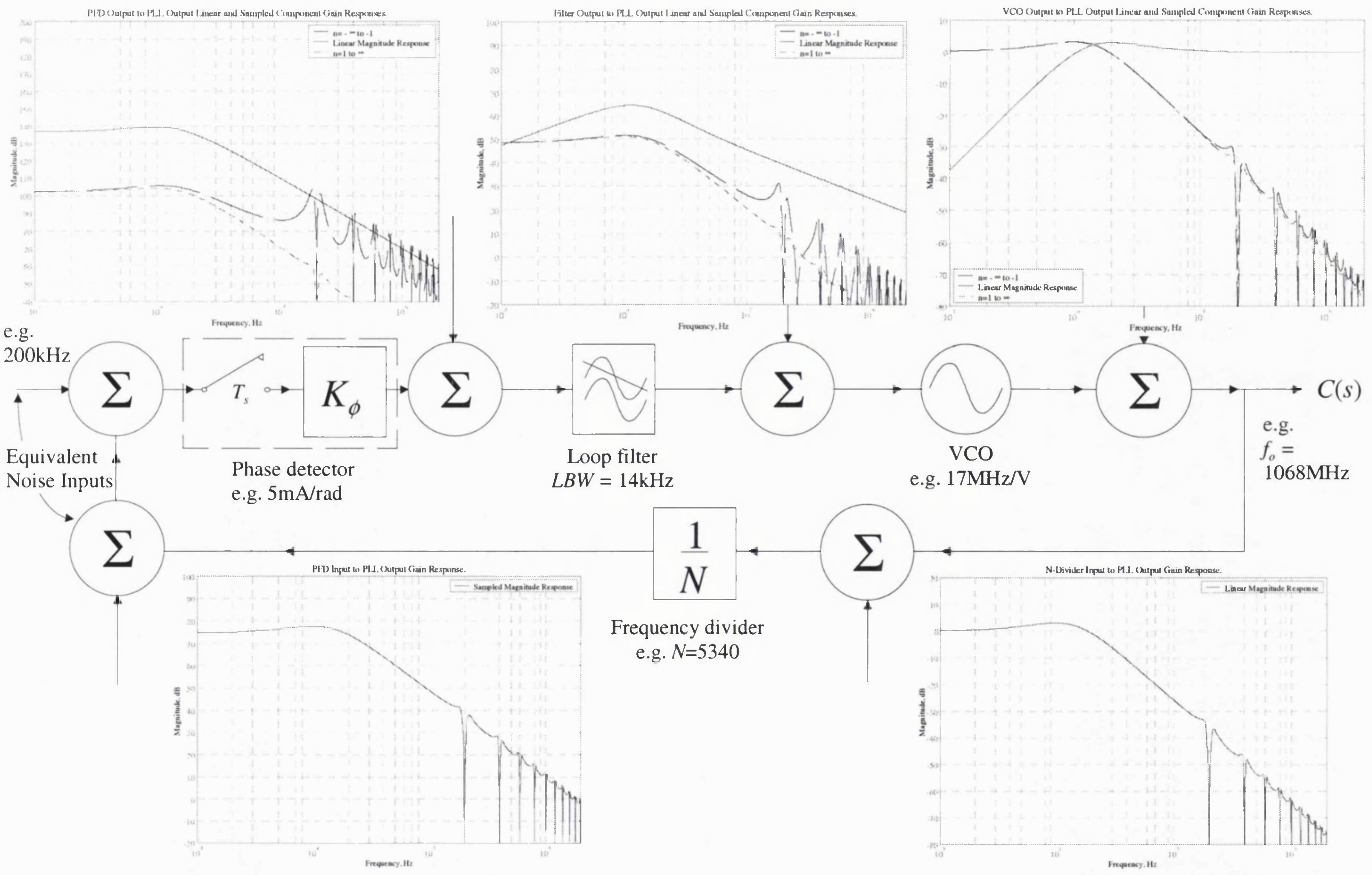

PFD Output to PLL Output Linear and Sampled Component Gain Responses.
n= - ∞ to -1
Linear Magnitude Response
n=1 to ∞
Magnitude, dB
Frequency, Hz
Filter Output to PLL Output Linear and Sampled Component Gain Responses.
n= - ∞ to -1
Linear Magnitude Response
n=1 to ∞
Magnitude, dB
Frequency, Hz
VCO Output to PLL Output Linear and Sampled Component Gain Responses.
n= - ∞ to -1
Linear Magnitude Response
n=1 to ∞
Magnitude, dB
Frequency, Hz
e.g.
200kHz
Σ
T_s
K_ϕ
Σ
Σ
Σ
$C(s)$
e.g.
f_o =
1068MHz
Equivalent
Noise Inputs
Phase detector
e.g. 5mA/rad
Loop filter
LBW = 14kHz
VCO
e.g. 17MHz/V
Σ
$\frac{1}{N}$
Σ
Frequency divider
e.g. N=5340
PFD Input to PLL Output Gain Response.
Sampled Magnitude Response
Magnitude, dB
Frequency, Hz
N-Divider Input to PLL Output Gain Response.
Linear Magnitude Response
Magnitude, dB
Frequency, Hz

Figure 4.12 Loop noise modulation characteristics from each node to PLL output

A precursory inspection of Eq 4.30 and Eq 4.31, indicates that these two formula are not simple transfer functions, because each contains the product of the noise and a frequency dependent loop gain parameter in the sampling summation.

$$\phi_{VCO}(s) = \left| \frac{N_{VCO}(s)}{1+\frac{1}{N}G(s)} \right| + \left| \frac{\left[\sum_{n=-\infty}^{n=-1} N_{VCO}(s - jn\omega_s)\right] \cdot \frac{1}{N} \cdot G(s)}{1+\frac{1}{N}\cdot \sum_{n=-\infty}^{n=-1} G(s-jn\omega_s)} \right| + \left| \frac{\left[\sum_{n=1}^{n=\infty} N_{VCO}(s - jn\omega_s)\right] \cdot \frac{1}{N} \cdot G(s)}{1+\frac{1}{N}\cdot \sum_{n=1}^{n=\infty} G(s-jn\omega_s)} \right| \qquad \text{Eq 4.32}$$

VCO Output to PLL Output

$$\phi_{NDiv}(s) = \left| \frac{\left[\sum_{n=-\infty}^{n=\infty} N_{NDiv}(s - jn\omega_s)\right] \cdot \frac{1}{N} G(s)}{1+\frac{1}{N}\cdot \sum_{n=-\infty}^{n=-1} G(s-jn\omega_s)} \right| \qquad \text{Eq 4.33}$$

N-Divider Input to PLL Output

$$\phi_{Ref}(s) = \left| \frac{\left[\sum_{n=-\infty}^{n=\infty} N_{Ref}(s - jn\omega_s)\right] \cdot G(s)}{1+\frac{1}{N}\cdot \sum_{n=-\infty}^{n=-1} G(s-jn\omega_s)} \right| \qquad \text{Eq 4.34}$$

PFD Input to PLL Output

Where *G*(*s*) is the forward loop gain of the PLL.

4.10 Experimental Results

Figure 4.13 and Figure 4.14 are two plots comparing the theoretical composition of a SSB phase noise profile using the techniques described in this section and the actual measured SSB phase noise profile of a test synthesiser. The theoretical predicted noise profile compares well with the measured, particularly in the areas around the sampling nulls, which demonstrates that the depth of these nulls is limited by the thermal noise of the loop filter. The extra noise seen close-in to the carrier at 100Hz in Figure 4.14 comes from the addition of reference signal generator, (an HP8665B), noise, used in the phase noise measurement system. The spurs seen close to the carrier on the measured plot, Figure 4.14 are parasitic 50Hz mains spurs and their harmonics, whilst the PLL sampling spurs at 200kHz offset and beyond compare favourably with the theoretical prediction shown in Figure 4.13. The aliasing of the loop bandwidth noise profile is clearly demonstrated by both the level and the shape of the noise observed around the base of the sampling spurs and vindicates the sampling mathematics used in the phase noise prediction.

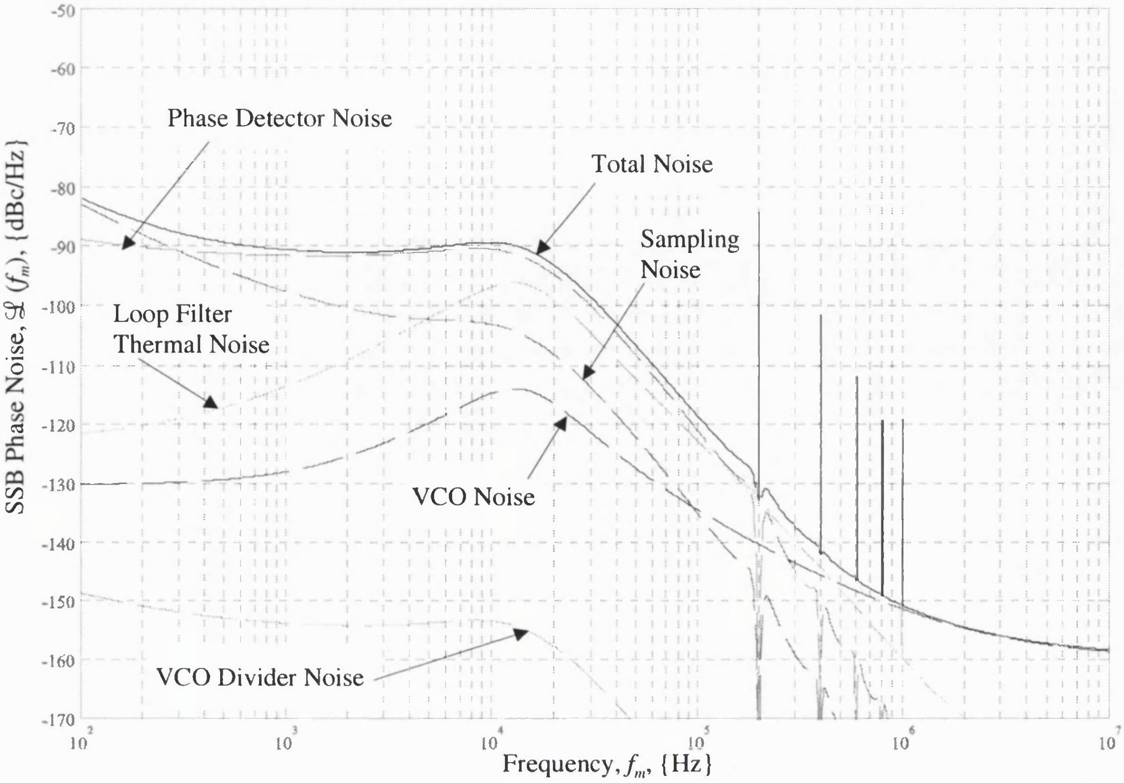


Figure 4.13 Theoretical composition of measured SSB phase noise

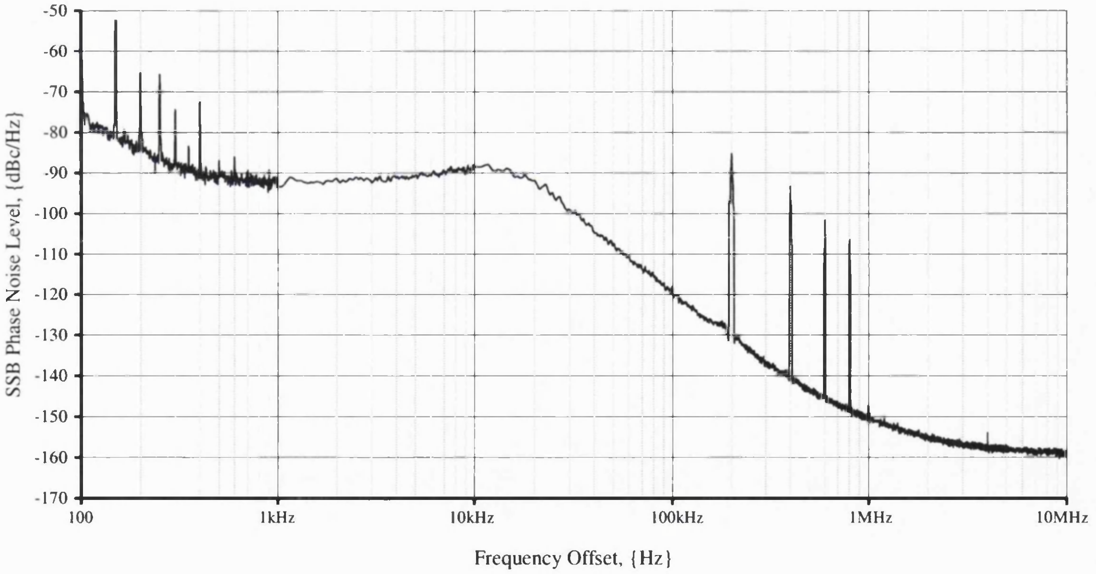


Figure 4.14 Measured SSB phase noise profile of PLL modelled in Figure 4.13

4.11 Summary

Understanding the effect of the loop filter thermal noise contribution to the overall SSB phase noise profile of a PLL, transpired to be one of the most rewarding of the analyses within this chapter. This is because, during the laboratory investigation aimed at verifying the mathematics developed here, it became clear that loop filter thermal noise accounts for the discrepancies often seen between simulation and measurement.

Modelling a sampled data system such as a PLL requires considerable care to eliminate from the mathematics any correlated components of noise. From the preceding analysis it is clear that these components only arise in the solution to transfer functions of signals injected after the PFD sampler in the forward path of the loop. Thus the format of the noise transfer formula from each noise injection point into the PLL, to the PLL output, will vary. This results from the mechanics of the mathematics used to account for uncorrelated noise being injected into the loop, whose instantaneous values vary from moment to moment. For this reason the simple sampled loop mathematics used to analyse a closed loop system containing one or more samplers, cannot be rigorously applied to a PLL application. As shown, the key is to remove all correlated components in the transfer function, by working across the sampling gate for each sample of noise and then repeating the analysis assuming this noise has independently been re-applied, to the loop at the sample gate. Using this technique, each source of noise will become de-correlated at the sampling gate and remains de-correlated from the original injected noise source, throughout the analysis.

5 PFD NOISE LIMITATIONS

5.1 Introduction to the PFD Problem

This chapter concentrates on the PFD element only. Contained within this chapter is an innovative mathematical insight into how the PFD device noise is manifest in the overall PLL phase noise profile. The application of these analyses is demonstrated and verified by laboratory measurements. These measurement results highlight the importance of the PFD in the overall synthesiser noise profile and its relation to the PLL configuration, whilst the associated analysis provides a simple set of formulae, which can be applied equally in both integer and fractional-n synthesiser simulations. The contents of this chapter were the subject of an IEE Electronics Letter and an IEE Proceedings Paper, [appendix A5].

Since the circuit for the tri-state phase frequency detector was first published [81], followed by the charge pump addition [82], the combination of these two elements has become ubiquitous in digital synthesiser applications. Several mathematical operators describing the noise contribution of the phase/frequency detector, (PFD), within a synthesiser have been offered, [83, 84] along with laboratory observations and theory, confirming that PFD jitter noise is dominant. These mathematical operators offer a 1Hz normalisation technique giving a figure of merit for a given synthesiser chip, allowing the performance of different devices to be compared. The figure of merit, (FOM), is a very similar approach to the intercept point concept adopted for extrapolating an amplifier's intermodulation products for a given operation scenario. The FOM can similarly be applied to a given synthesiser configuration to determine the PFD noise contribution contained within the PFD output phase noise profile.

The objective of this chapter is to consider the thermal noise calculations used to arrive at the FOM for a given logic family and illustrate how this value is translated to the expected in-band phase noise contribution of the PFD in the synthesiser output. The translation of the FOM to in-band phase noise includes the effects of the sampling frequency and subsequent multiplication of the synthesiser. To complete the PFD noise profile, the normally much lower flicker noise bandwidth is also accounted for.

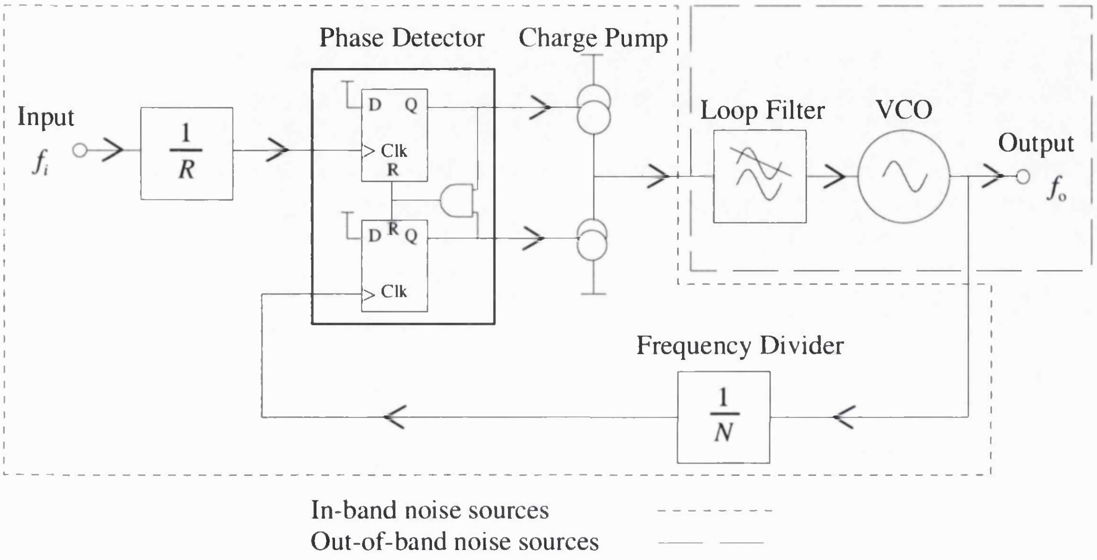


Figure 5.1 Phase/frequency detector under consideration, within synthesiser

5.2 PFD Noise Model

5.2.1 PFD Noise Processes

The phase noise profile observed at the output of a digital phase locked loop is a composite of all noise sources within the loop, each individually modified by the phase transfer characteristic from their point of injection to the loop output. With reference to Figure 5.1, any noise injected before the loop filter will predominantly contribute to the in-band phase noise, conversely any noise source injected in the loop filter and beyond will predominantly contribute to the out-of-band component of the phase noise profile. Of all the noise sources within a PLL there are three prime sources which dominate the overall phase noise profile, as might be measured on a phase noise test set. These sources are the multiplied reference signal noise, the digital PFD noise and the free running VCO phase noise. Additional noise introduced by a less-than-noise-optimal loop bandwidth can easily be predicted by the roll-off characteristics of the filter used within the PLL, however, the base-line level of this noise is again dependent upon the digital phase/frequency detector noise. The following analysis concerns the timing jitter introduced at the phase/frequency detector input, which largely dominates the in-band component of the phase noise profile, and how this noise manifests itself within the output phase noise profile of a synthesiser.

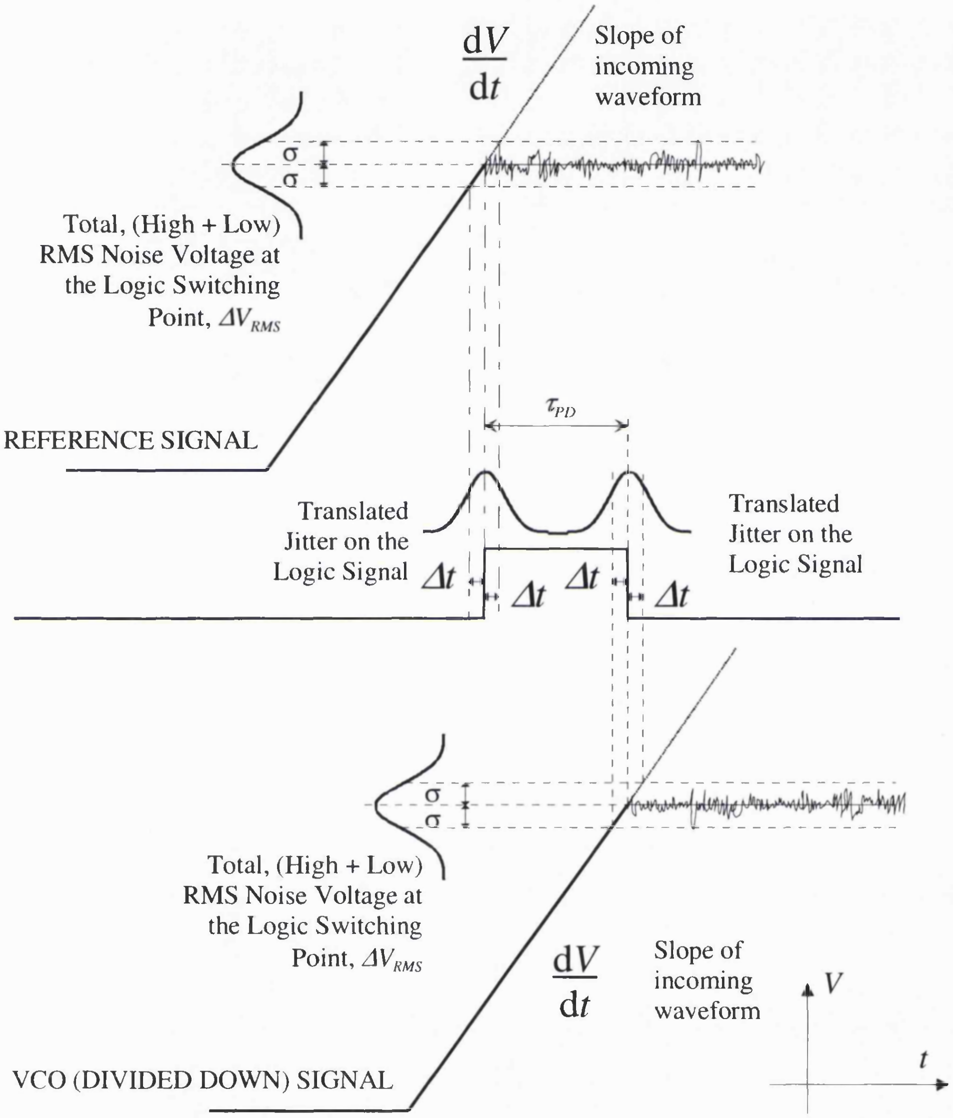

$\frac{dV}{dt}$
Slope of incoming waveform
σ
σ
Total, (High + Low) RMS Noise Voltage at the Logic Switching Point, ΔV_{RMS}
REFERENCE SIGNAL
τ_{PD}
Translated Jitter on the Logic Signal
Translated Jitter on the Logic Signal
Δt
Δt
Δt
Δt
σ
σ
Total, (High + Low) RMS Noise Voltage at the Logic Switching Point, ΔV_{RMS}
$\frac{dV}{dt}$
Slope of incoming waveform
V
t
VCO (DIVIDED DOWN) SIGNAL

Figure 5.2 Logic transfer characteristic at the D-type input of the PFD [85]

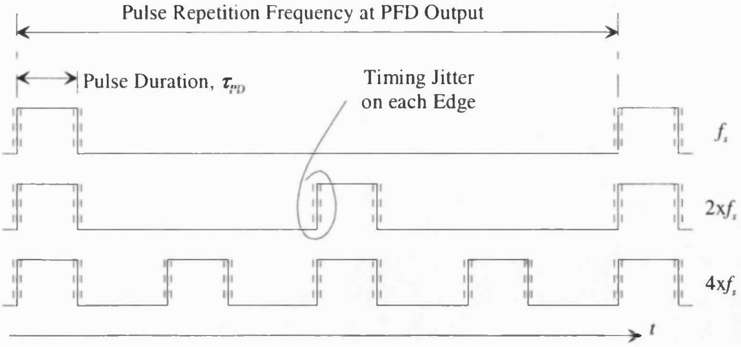

Pulse Repetition Frequency at PFD Output
Pulse Duration, τ_{PD}
Timing Jitter on each Edge
f_s
$2xf_s$
$4xf_s$
t

Figure 5.3 Effects of pulse duration and timing jitter for increasing PFD operating frequencies

5.2.2 PFD Timing Jitter Analysis

Additive noise, predominantly thermal, within the PFD gives rise to timing jitter in the edges of the output pulses, Figure 5.2. This jitter, which is uncorrelated and present on both rising and falling edges, may be considered equivalent to a certain input timing jitter of Δt seconds RMS on the reference input of an ideal, noise free PFD. Depending upon the PFD design, one or other of the edges of each incoming waveform independently triggers each input device giving the phase comparison with the time difference between these edges being translated to the required current pulse duration at the charge pump output, necessary to maintain phase lock. By superposition, each occurrence of this PFD timing jitter is similarly translated to an equivalent phase jitter, which modulates the width of the pulse operating the charge pump, Figure 5.3, injecting this noise into the remainder of the synthesiser. For a phase detector with an operating frequency of f_s the equivalent phase jitter is:

$$\Delta\phi_{in} = 2\pi f_s \Delta t \qquad \text{[rad(rms)]} \qquad \text{Eq 5.1}$$

In practice, Δt is very small - of the order of pico-seconds - whilst the thermal noise possesses a bandwidth much greater than the PFD sampling frequency. The PFD, being an edge triggered sampling device with an output pulse train of very low duty cycle in a locked loop, is a good approximation to an impulse sampler possessing an equivalent noise bandwidth of half the sampling frequency and virtually uniform spectral density of translated components over this frequency range, [86]. Hence, the equivalent input double-sided spectral density of phase fluctuations is:

$$S\phi_{in}(f_m) = \frac{(\Delta\phi_{in})^2}{f_s/2} = 8\pi^2 f_s \Delta t^2 \quad \left[\frac{\text{rad}^2}{\text{Hz}}\right] \qquad \text{Eq 5.2}$$

This indicates a 10dB/decade increase in PFD noise output with the phase detector operating frequency, f_s. For a typical phase locked synthesiser with only a divider feedback path, the resulting output phase noise spectral density is subject to a gain equivalent to the divider ratio gain $N = f_o / f_s$, where f_o is the output frequency of the synthesiser. The value of N can be either integer or fractional. Therefore, it follows that,

$$\begin{aligned} S\phi_{out}(f_m) &= 8\pi^2 f_s \Delta t^2 N^2 \\ &\equiv \frac{8\pi^2 \Delta t^2 f_o^2}{f_s} \quad \left[\frac{\text{rad}^2}{\text{Hz}}\right] \end{aligned} \qquad \text{Eq 5.3}$$

This now indicates an overall 10dB/decade decrease in output phase noise with the phase detector operating frequency, f_s. Assuming the overall output phase noise is smaller than 0·1 radians, then from [76], the output SSB phase noise power, $\mathcal{L}(f_m)$, is equal to one half of the double-sideband power spectral density of phase fluctuations;

$$\mathcal{L}(f_m) = \frac{1}{2} S\phi_{out}(f_m) = 10\text{Log}_{10}\left(\frac{4\pi^2 \Delta t^2 f_o^2}{f_s}\right) \quad [\text{dBc/Hz}] \qquad \text{Eq 5.4}$$

This can more conveniently be expressed in terms of dB as:

$$\mathcal{L}(f_m) = \text{FOM}\{\text{dBc/Hz}^2\} + 20\text{Log}_{10} f_o - 10\text{Log}_{10} f_s \quad [\text{dBc/Hz}] \qquad \text{Eq 5.5}$$

Returning to the divider substitution in Eq 5.3, gives the more familiar version of this formula dependent on a $20\text{Log}_{10}(N)$ term:

$$\mathcal{L}(f_m) = \text{FOM}\{\text{dBc/Hz}^2\} + 20\text{Log}_{10} N + 10\text{Log}_{10} f_s \quad [\text{dBc/Hz}] \qquad \text{Eq 5.6}$$

Where FOM is the figure of merit of the particular phase/frequency detector used and is constant. The implication of Eq 5.5 is not widely appreciated but is highly significant as it indicates that for a given synthesiser output frequency, f_o, the phase noise decreases by 10dB/decade with the PFD operating frequency (and also increases by 20dB/decade with output frequency). It is therefore, clearly advantageous to run phase/frequency detectors at the highest possible operating frequency available within the constraints of the given application, to reduce the in-band phase noise plateau. Either expression Eq 5.5, or its equivalent Eq 5.6, serves two purposes; they allow calculation of in-band phase noise attributed to the PFD operation for a given synthesiser frequency configuration and allow a FOM to be derived from experimental measurement, offering a simple means of comparing different synthesiser chips. Typical FOM values range from –213dBc/Hz^2 for some older devices to –222dBc/Hz^2 for the latest device.

Throughout this analysis the assumption is that the PFD jitter noise is Gaussian amplitude distributed with a noise bandwidth in excess of the bandwidth of the PFD input circuitry. To support the assumption of a Gaussian amplitude distribution, the in-band phase noise distribution of 14789 identical synthesiser chips from different wafer lots, was plotted, Figure 5.4. The logic noise bandwidth of the PFD used for these measurements was determined by manufacturer's simulations to be in excess of 2GHz, and well beyond the maximum specified operating frequency of the PFD under test.

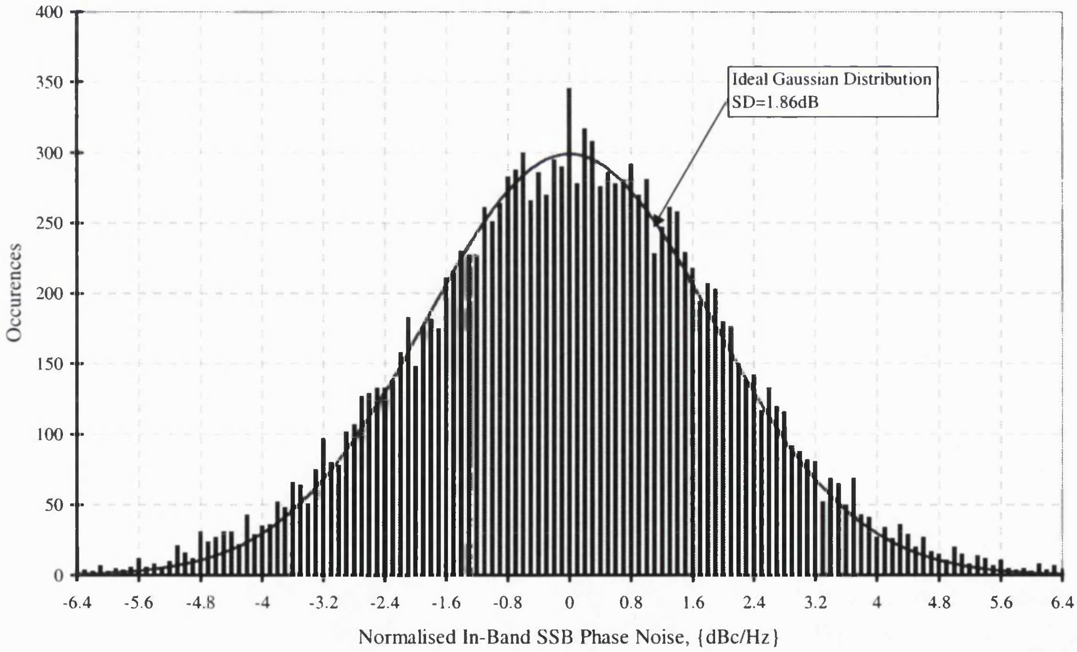


Figure 5.4 Normalised distribution of measured in-band phase noise

5.2.3 Additive Thermal Noise

An estimate of the Figure Of Merit as a function of the design parameters of a given PFD may be obtained as follows. It will be assumed that the input is a rectangular pulse train of peak amplitude $\hat{V}$, (the logic level swing), with a degree of rounding of the pulse edges as modelled by a first order low-pass function of 3dB cut-off frequency ω_c. In addition, the PFD input will be band-limited according to a similar first order low-pass function. Thus, the pulse edges arriving at the PFD may be described by

$$\begin{aligned} v(s) &= \frac{\hat{V}}{s}\frac{\omega_c^2}{(s+\omega_c)^2} \\ &\equiv \hat{V}\left(\frac{1}{s}-\frac{1}{s+\omega_c}-\frac{\omega_c}{(s+\omega_c)^2}\right) \end{aligned} \qquad \text{Eq 5.7}$$

and so

$$v(t)=\hat{V}\left\lfloor 1-e^{-\omega_c t}-\omega_c t\,e^{-\omega_c t}\right\rfloor \qquad \text{Eq 5.8}$$

where the slope of these edges is given by

$$\frac{dv(t)}{dt}=\hat{V}\omega_c^2 t\,e^{-\omega_c t} \qquad \text{Eq 5.9}$$

Now assuming the logic threshold is mid-way between logic "0" and logic "1" levels then, from Eq 5.8, the PFD will be triggered at time $\omega_c t' = 1{\cdot}68$ giving a slope at this threshold of

$$\left.\frac{dv(t)}{dt}\right|_{t'}=2f_c\hat{V} \qquad \text{Eq 5.10}$$

The equivalent rectangular noise bandwidth of a first-order low-pass filter is [87]

$$B=\frac{\pi}{2}f_c \qquad \text{Eq 5.11}$$

and so the noise admitted to the PFD has an amplitude of

$$V_n=\sqrt{kTBFR'}=\sqrt{\frac{\pi kTFf_cR'}{2}} \quad [\mathrm{V(rms)}] \qquad \text{Eq 5.12}$$

where F is the noise factor and R' is the impedance of the PFD input. This additive noise is converted into timing jitter according to the slope of the input pulse train, Figure 5.2.

$$\text{Timing Jitter} = {V_n} \Big/ \left.\frac{dv(t)}{dt}\right|_{t'} \quad [\text{s(rms)}] \qquad \text{Eq 5.13}$$

where it is clear that a high input slew rate, especially across the logic trip point, is preferable to reduce the translation of thermal noise at the PFD input to timing jitter. Since this jitter is independently generated at each PFD input and is uncorrelated, it is present on both edges of the output pulses giving a total RMS value of √2 times this result,

$$\begin{aligned} \Delta t &= {\sqrt{2}\,V_n} \Big/ \left.\frac{dv(t)}{dt}\right|_{t'} \\ &= \frac{1}{2\hat{V}}\sqrt{\frac{\pi kTFR'}{f_c}} \quad [\text{s(rms)}] \end{aligned} \qquad \text{Eq 5.14}$$

By substitution in equation Eq 5.4 the FOM can be defined as,

$$\begin{aligned} FOM &= 4\pi^2\Delta t^2 \\ &= \frac{\pi^3 kTFR'}{\hat{V}^2 f_c} \\ &= 10\,\text{Log}_{10}\left[\frac{\pi^3 kTFR'}{\hat{V}^2 f_c}\right] \quad \left[\text{dBc}/\text{Hz}^2\right] \end{aligned} \qquad \text{Eq 5.15}$$

Using known parameters for a given logic used to construct the PFD, it is possible, using this expression, to approximate the FOM for that device.

5.2.4 Flicker Noise Analysis

To complement the preceding analysis the flicker noise contribution is considered here. For the normally much lower flicker noise bandwidths compared with the sampling frequency, the flicker noise suffers minimum degradation from aliasing, consequently its profile is faithfully reproduced. The level of this flicker noise remains constant and independent of the synthesiser loop gain and sampling frequency, because it is usually band limited to frequencies below the PFD sampling frequencies used. With reference to a known level of flicker noise for a given device, the analysis is straightforward and can be accounted for using the notation in Kuo and Franklin, [73, 74] by:

$$S^{*}_{Flicker}(j\omega) = \sum_{n=-\infty}^{\infty} S_{Flicker}(j\omega + jn\omega_s) \qquad \text{Eq 5.16}$$

Where $S_{Flicker}$ is the power spectral density of the flicker noise profile. Therefore, the flicker noise will repeat at each multiple of the sampling frequency.

5.3 Experimental Results

For the first result shown in this chapter, the residual phase-noise measurement technique, [68], was used. This technique requires two identical device-under-test units to be fed from the same clean reference source which, being correlated, is subsequently removed by the measurement process making it possible to view the close-in noise of these test devices. If three identical device-under-test units are available, it is possible under software control, to measure each permutation of pair of test devices and then determine the individual noise for each test device. Because this measurement technique is especially useful for revealing the flicker noise of the devices-under-test, particular care should be taken to avoid adding the flicker noise of any external buffer amplifiers.

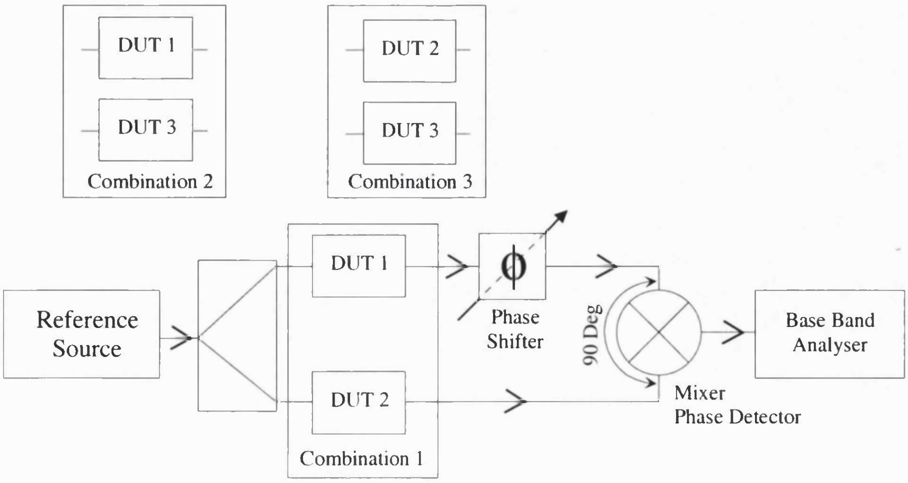


Figure 5.5 Residual phase noise set-up used for three-device measurements

Using the residual phase-noise measurement technique and three identical synthesiser device-under-test units, each combination of these pairs was measured against one another, Figure 5.5. For each measurement, the sampling frequency was successively doubled providing three sets of measurements at each sampling frequency, whilst maintaining a constant output frequency, (1638·4MHz). The test units were set for a nominal loop natural frequency of 300kHz at a 6·4MHz sampling frequency. Using processing available within the phase-noise measurement software, the absolute noise profile of one test synthesiser was then determined for each sampling frequency. With the reference signal removed, the device flicker-noise profile and the in-band phase noise plateau were exposed, Figure 5.6. From any one of the in-band noise plateaux thus measured, the FOM can be calculated.

Contained within Figure 5.6 is PFD flicker noise and the effects of aliasing of wideband jitter noise, giving rise to the FOM values discussed earlier. The FOM of merit for the PFD used can be calculated from Eq 5.5 using the measured value for in-band phase noise, (across the offset frequency of 100Hz-10kHz), in Figure 5.6:

$$\begin{aligned} FOM &= -92\cdot 64\text{dBc/Hz} + 10\text{Log}_{10}(800\text{kHz}) - 20\text{Log}_{10}(1638\cdot 4\text{MHz}) \\ &= -217\cdot 9\,\text{dBc/Hz}^2 \end{aligned} \qquad \text{Eq 5.17}$$

With the overlay of each successive doubling in sampling frequency, a clear 3dB improvement of measured in-band phase noise is apparent. This is predicted by equations Eq 5.5 and Eq 5.6, where each successive doubling of sampling frequency increases the amount of uncorrelated jitter noise by a $10\text{Log}_{10}(f_s)$ factor, whilst concurrently halving the value of N for the $20\text{Log}_{10}(N)$ term, arriving at an overall 3dB noise reduction, for a fixed output frequency. A pictorial demonstration for each noise level measured in Figure 5.6, using equation Eq 5.6, is presented diagrammatically in Figure 5.7.

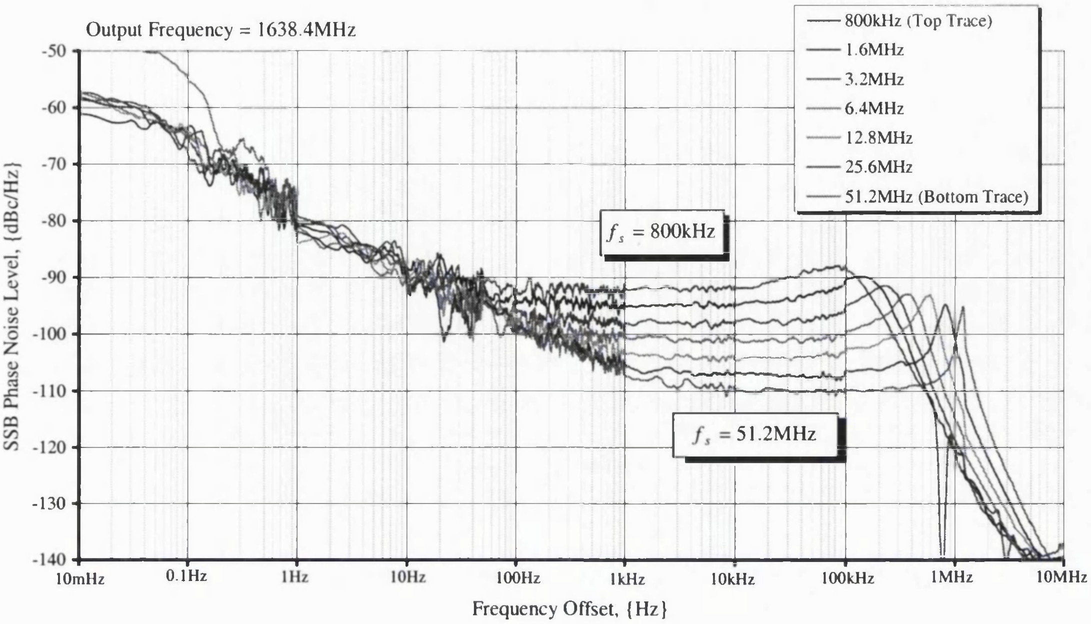


Figure 5.6 Measured phase noise profiles in a 3rd order, type II PLL synthesiser for PFD operating frequencies of 0·8, 1·6, 3·2, 6·4, 12·8, 25·6 & 51·2MHz

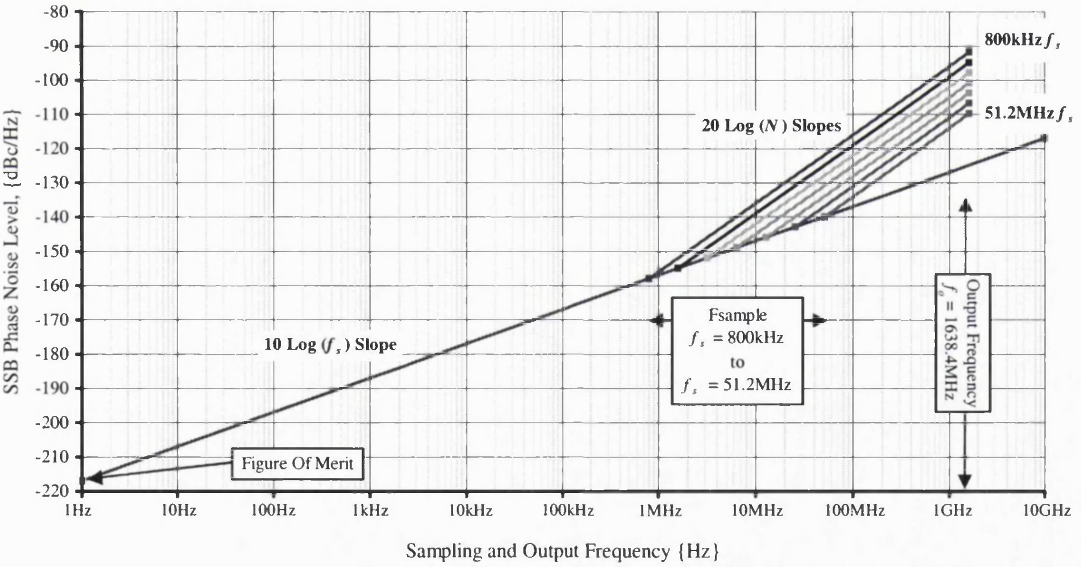


Figure 5.7 White noise plateaux for different sampling frequencies and f_o = 1638·4MHz

Eq 5.6 suggests that the in-band phase noise profile will change with output frequency in a $20\mathrm{Log}_{10}(f_o)$ manner for a constant sampling frequency. The measurements shown in Figure 5.8, were taken using a suitably buffered PFD output to drive a signal generator emulating a VCO. In this way, the measurement could be repeated over octave frequency ranges. As expected, a $20\text{Log}_{10}(f_o)$ slope is observed.

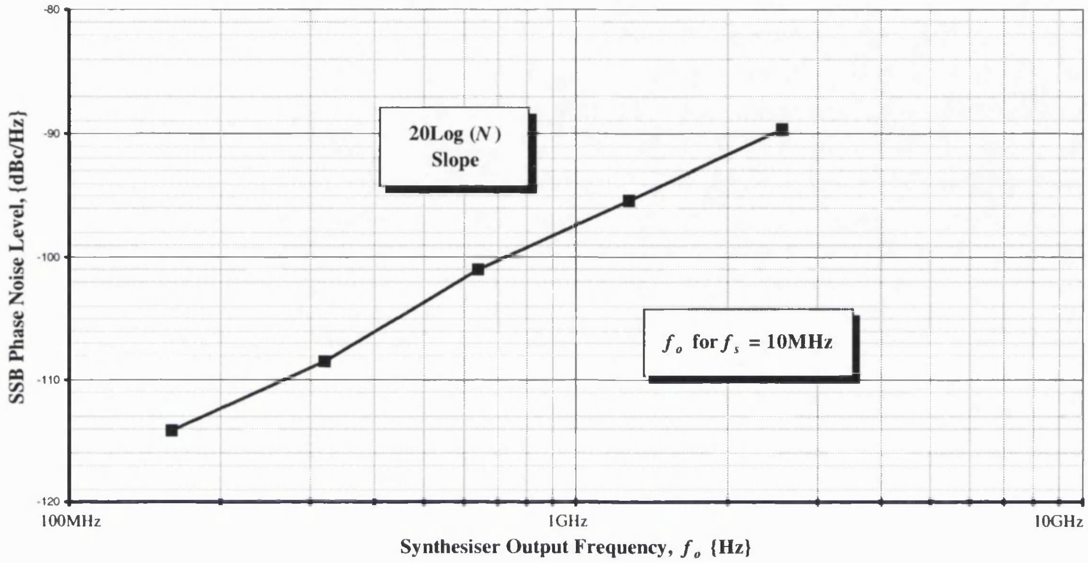


Figure 5.8 Measured in-band phase noise variations with output frequency for constant sampling frequencies

5.4 Discussion

This model highlights the limitations of some operational aspects of synthesisers and indicates how they might be optimised. From the analysis offered in section 5.2.3, with Eq 5.7 representing one type of input waveform characteristic, it can be seen that there is an optimum PFD logic switching point to minimise the translation of PFD input noise to PFD output timing jitter. Comparing this waveform with other theoretical waveform shapes considered during the course of this study, Figure 5.9, the intuitive link between bandwidth and optimal switching point can be demonstrated, illustrating the need to optimise the bandwidths of both the n and r-divider outputs to PFD input circuits. In general, a wider bandwidth should offer better performance.

For any given waveform a further degree of freedom open to the synthesiser chip designer, is the choice of logic switching point to ensure the PFD is switched at the peak change of slope of that incoming waveform. Therefore, for any given digital synthesiser chip design, both the bandwidth and the switching point of the PFD input interface circuits need to be optimised for minimal thermal noise translation. For example, from Figure 5.9, the optimum threshold for the PFD model considered in this chapter is 28% of the logic level swing, suggesting that the more typical 50% switching threshold is somewhat sub-optimal. The three slopes shown in Figure 5.9 are derived from three different example input slopes, which might be applied to the PFD input.

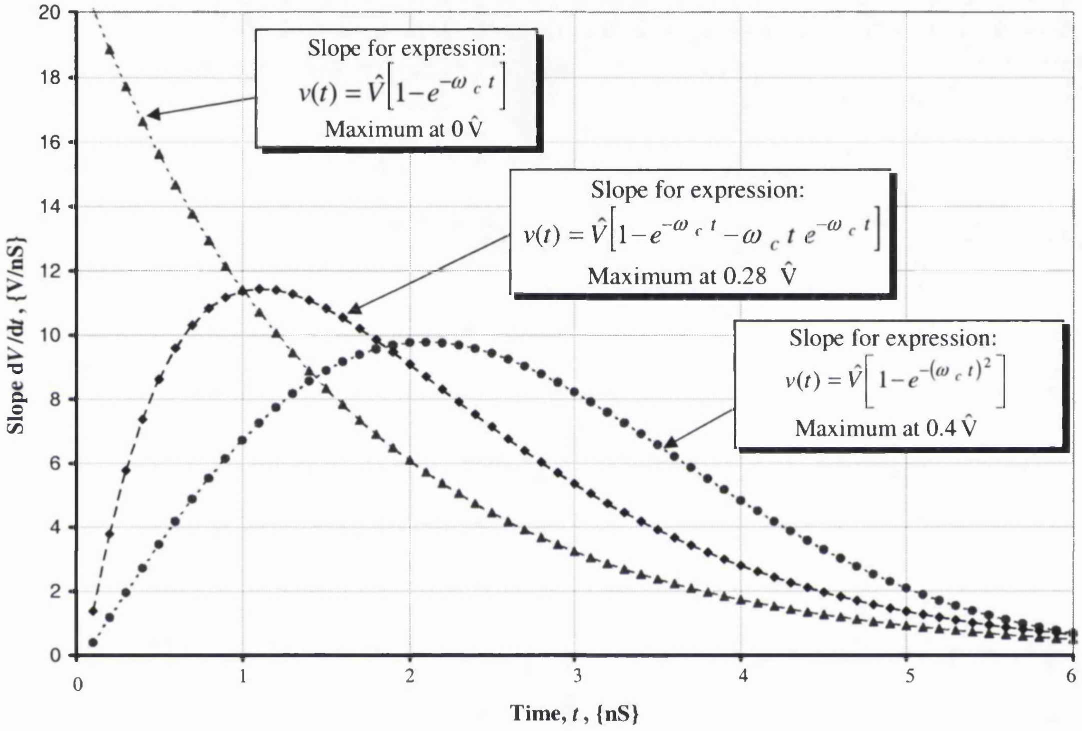


Figure 5.9 Optimal switching points for different PFD input slope types

One particularly interesting observation from Figure 5.7, is that with a PFD operating at the PLL output frequency of 1638·4MHz in this example, the best achievable white noise plateau value is ≈-124dBc/Hz. Consequently, no sigma delta based fractional synthesiser, using this PFD, can ever achieve a plateau noise floor better than this value, irrespective of the sigma delta close-in noise suppression, because of the jitter performance of this PFD. Sigma delta noise shaping applies only to noise contributed by the fractional-n division ratio switching process and has no bearing on the more fundamental limitation considered here, PFD noise.

5.5 Summary

This chapter has presented a theoretical basis for a technique used to predict the in-band phase noise plateau level for a given synthesiser configuration based on a figure of merit, (FOM), approach. The FOM equates to a normalisation of the PFD plateau noise, which can easily be measured and used to compare different synthesiser chips in the first stages of any selection process. An analysis has been presented which enables the FOM to be predicted for a given PFD from device parameters. Using this method, it has been shown how the in-band phase noise of the PFD can then be predicted for a given synthesiser frequency configuration. It has further been shown that the typical 50% PFD switching threshold is somewhat sub-optimal and that a lower threshold should provide better noise performance. By inference from these analyses, several options are open to the synthesiser designer to improve the in-band phase noise performance of a given synthesiser, whilst in contrast, the PFD is suggested as the fundamental limitation of both integer and fractional-n synthesisers, requiring attention before the full potential of fractional-n synthesisers can be realised.

Experimental results have been presented for an integer synthesiser showing excellent agreement with theory and, in particular, clearly demonstrating the anticipated 10 dB/decade improvement of synthesiser phase noise with increasing PFD operating frequency. The theory is equally applicable to fractional-n synthesiser applications.

6 LOOP FILTER DESIGN

6.1 Introduction

The loop filter within any closed loop system is the single element under the designer's control enabling both the system static noise and dynamic performance to be set, within the confines of all other elements within the system.

Typically, for a phase locked synthesiser, a type II third-order loop[17] is chosen because it offers zero steady state error for a ramped phase input and good filtering of PFD sampling components. Generally for single channel IF radio receivers[18], the synthesiser raster is usually set to the channel spacing or an integer sub-multiple of the channel spacing. In both instances, the common charge pump driven integer synthesiser introduces sampling spurs in the middle of the adjacent channels. This can be a problem within a receiver system, because reciprocal mixing within the first mixer driven by this local oscillator source can desensitise the receiver in the presence of strong adjacent signals. Therefore a fourth-order loop is often preferred, allowing the designer the freedom to introduce an extra pole for further attenuation of any sampling spurs, before they modulate the VCO.

With this extra pole the complexity of the mathematics associated with the accurate calculation of the component values has led to many approximations in their determination. To further compound this problem, the impact of sampling with respect to different loop bandwidth to sampling ratios is often overlooked and its consequent impact on the closed loop phase margin.

[17] Within a phase locked loop the loop filter provides the majority of the poles whilst the "loop" gain adds an additional pole taken from the VCO phase transfer characteristic, K_{VCO}/s.

[18] As compared with wide IF radios designed for multiple carrier transmission and reception.

This chapter will first consider the design of PLL loop filters using linear techniques that in general are satisfactory for most PLL systems where the loop bandwidth is small compared with the sampling frequency of the PFD, [82]. However in some applications small loop bandwidth to sampling frequency ratios are necessary to meet the lock time requirements. To account for the effects of sampling, particularly when the loop bandwidth approaches the loop sampling frequency, a simple approximation is included to pre-adjust the phase margin before the loop filter components are calculated. Throughout this work the underlying requirement of peak performance is maintained without compromising the phase margin or loop bandwidth, so as to guarantee optimal lock times.

As an introduction to the problem, the polar plot given in Figure 6.1 shows a typical type II, fourth-order loop gain response when the loop filter values are calculated using the mathematics provided in articles [88, 89, 90]. The difficulty is that the extra pole added into the filter to provide additional attenuation, provides this attenuation at the expense of the loop bandwidth, thereby increasing the lock time and reducing the phase margin.

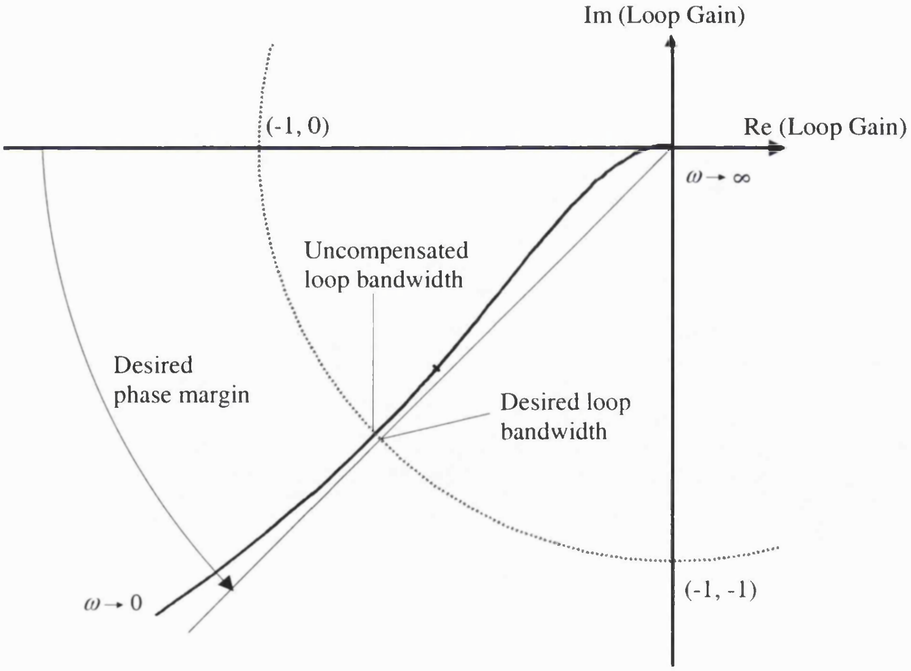


Figure 6.1 Polar plot of uncompensated type II, fourth-order loop design

6.2 The Mathematics

Two important characteristics exploited to assist the mathematics of loop filter design are; 1) the loop gain becomes equal to 1 (or 0dB) at the loop natural frequency and 2) at this frequency, there exists a phase turning point indicating the loop phase margin.

For each of the following loop filter analyses the transfer functions will be described in terms of generic time constants and it is the determination of these time constants that will be sought. This approach avoids limiting the analysis to any particular passive or active loop filter topology that may be chosen for a given application.

6.2.1 Type II, Third-Order Loop Filter Design

To guarantee a zero steady state phase error between the two inputs of the phase detector the minimum system requirements are a type II system. This is the minimum type of system that produces a phase coherent output relative to a ramp input. For this analysis, an appreciation of the design techniques used for the benchmark second-order filters[19] provides a good basis upon which to develop the third-order loop filter required here.

A third-order loop filter possesses the generalised transfer function:

$$\text{Filter}_{3rd}(s) = \frac{1}{s\tau_0} \cdot \frac{(1+s\tau_1)}{(1+s\tau_2)} \qquad \text{Eq 6.1}$$

Where $1/s\tau_0$ is the low frequency filter gain and τ_1 and τ_2 are the time constants. From Eq 6.1 the open loop gain is:

$$GH(s) = \frac{K_\phi K_v}{Ns} \cdot \frac{1}{s\tau_0} \cdot \frac{(1+s\tau_1)}{(1+s\tau_2)} \qquad \text{Eq 6.2}$$

Typically the first step in the analysis is to define the turning point of the phase trajectory of this expression such that it occurs at the loop natural frequency, ω_n. Substituting $j\omega$ for s, the phase of Eq 6.1 at instantaneous frequency, ω, is:

[19] Throughout this analysis the definitions of system *type* refers to the number of poles of the loop gain transfer function; *G(s)H(s)* located at the origin, whilst the *order* of the system refers to the highest degree of the Characteristic Equation polynomial expression.

$$\angle GH(s)\big|_{s=j\omega} = -\pi + \arctan\left(\frac{1+j\omega\tau_1}{1+j\omega\tau_2}\right)$$
$$\therefore \angle GH(\omega) = -\pi + \arctan\left(\frac{\omega(\tau_1-\tau_2)}{1+\omega^2\tau_1\tau_2}\right) \qquad \text{Eq 6.3}$$

The argument of *arctan* contains the frequency dependent phase value, which when differentiated with respect to frequency, *ω.* Equating this differential to zero and choosing the positive frequency solution, allows the phase turning frequency, now designated the loop natural frequency ω_n, to be expressed in terms of the zero and pole breakpoints, τ_1 and τ_2:

$$\omega_n = \frac{1}{\sqrt{\tau_1\tau_2}} \qquad \text{Eq 6.4}$$

The phase trajectory with respect to frequency is symmetrical about the phase turning point, ω_n, suggesting ω_n should be the geometric mean between τ_1 and τ_2. Substituting Eq 6.4 into the phase value of Eq 6.3, noting that the phase variation equals the required phase margin, *PM*, at $\omega = \omega_n$, the loop natural frequency, gives the quadratic:

$$\tau_1^2 - \frac{2\tau_1}{\omega_n}\tan(PM) - \frac{1}{\omega_n^2} = 0 \qquad \text{Eq 6.5}$$

The positive root of Eq 6.5 provides an expression for calculating time constant τ_1 in terms of the loop design parameters of phase margin, *PM*, and loop natural frequency, ω_n:

$$\tau_1 = \frac{\tan(PM)+\sec(PM)}{\omega_n} \qquad \text{Eq 6.6}$$

By substituting Eq 6.6 into Eq 6.4 similarly allows time constant τ_2 to be calculated using the design parameters of phase margin and loop natural frequency:

$$\tau_2 = \frac{1}{\omega_\text{n}\left[\tan(\text{PM})+\sec(\text{PM})\right]} \qquad \text{Eq 6.7}$$

To determine the gain term, τ_0, of the loop filter, the magnitude of the loop gain at the loop natural frequency is equated to unity.

$$|Loop\,Gain(\omega)| = \frac{K_\phi K_v}{N\,\omega_n^2\,\tau_0}\sqrt{\frac{\left(1+\omega_n^2\tau_1^2\right)}{\left(1+\omega_n^2\tau_2^2\right)}} \equiv \frac{K_\phi K_v}{N\,\omega_n^2\,\tau_0}\left(\tan(PM)+\sec(PM)\right)$$

$$\therefore \tau_0 = \frac{K_\phi K_v}{N\,\omega_n^2}\left(\tan(PM)+\sec(PM)\right)$$

Eq 6.8

With values for these two time-constants and the loop filter gain, then using the standard loop filter component designations C_1, C_2 and R_2, Figure 6.2, these components can be calculated for a second-order passive charge pump driven loop filter topology.

$$C_1 = \frac{\tau_2}{\tau_1}\tau_0$$

$$C_2 = \tau_0 - C_1$$

$$R_2 = \frac{\tau_1}{C_2}$$

Eq 6.9

This technique for calculating the filter component values in a type II, third-order loop remains universally accepted and is well documented, [87]. Paradoxically, the design of the component values for the filter used in a type II, fourth-order loop has been the subject of many different papers and articles each reporting the authors' interpretation of how the extra time constant could be accounted for.

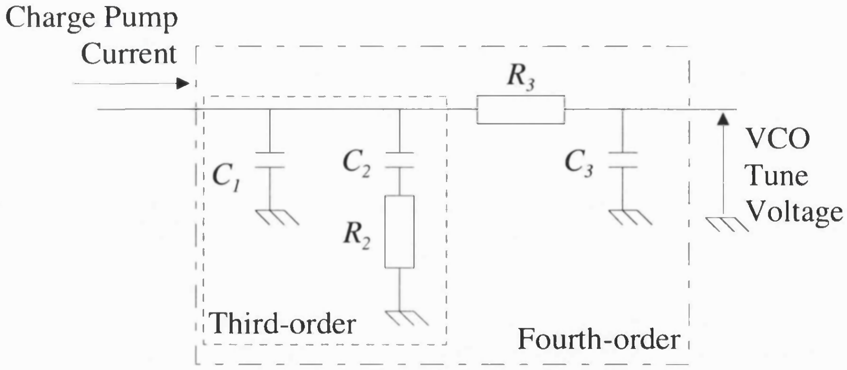


Figure 6.2 Standard passive loop filter component designations

6.2.2 Type II, Fourth-Order Loop Filter Design

The design of higher order filters is well documented, [88], however interestingly the underlying problem with all these designs is that they fail to account for the increasing reduction in loop natural frequency resulting from the increased attenuation required of the extra filtering that has been added. In summary, they achieve superior levels of attenuation by decreasing the loop natural frequency, which is a point touched on by Rohde *et al.*, [91]. In keeping with the objective of this work, this section will focus on the limitations of adding extra filtering before it starts to compromise the loop natural frequency and phase margin, with the consequent reduction in lock time. The intention is to determine the maximum amount of attenuation that can be introduced before the loop natural frequency is disturbed, for a given loop natural frequency to sampling frequency ratio.

Increasingly often a fourth-order loop filter design is used to improve the first reference spur attenuation by adding another "RC" pole, Figure 6.2, although the specified attenuation frequency could equally be required at another frequency offset, to reduce other known spurious products within the synthesiser architecture. For the convenience of this analysis, the required attenuation will be assumed to be at the first sampling frequency spur offset, ω_s, from the carrier.

Analysing this fourth-order loop filter in terms of time constants only, this additional pole is designated the time constant, τ_6. The required value for τ_6, given an additional attenuation requirement at the sampling frequency, ω_s, is:

$$\tau_6 = \frac{1}{\omega_s}\sqrt{10^{\frac{Atten}{20}} - 1} \qquad \text{Eq 6.10}$$

Where *Atten* is the additional attenuation required at the first sampling frequency offset from the carrier. With this additional pole the generic loop filter transfer function becomes:

$$\text{Filter}_{4th}(s) = \frac{1}{s\tau_3} \cdot \frac{(1+s\tau_4)}{(1+s\tau_5)(1+s\tau_6)} \qquad \text{Eq 6.11}$$

i.e. a third-order filter, giving a fourth-order closed loop response.

Therefore for a given required attenuation, *Atten*, this additional time constant can be simply calculated and included in the expressions based on Eq 6.10. Unfortunately, this extra time constant introduces phase and amplitude components of their own that need to be compensated for in the calculation of the fourth-order loop filter time constants τ_4 and τ_5, Figure 6.3.

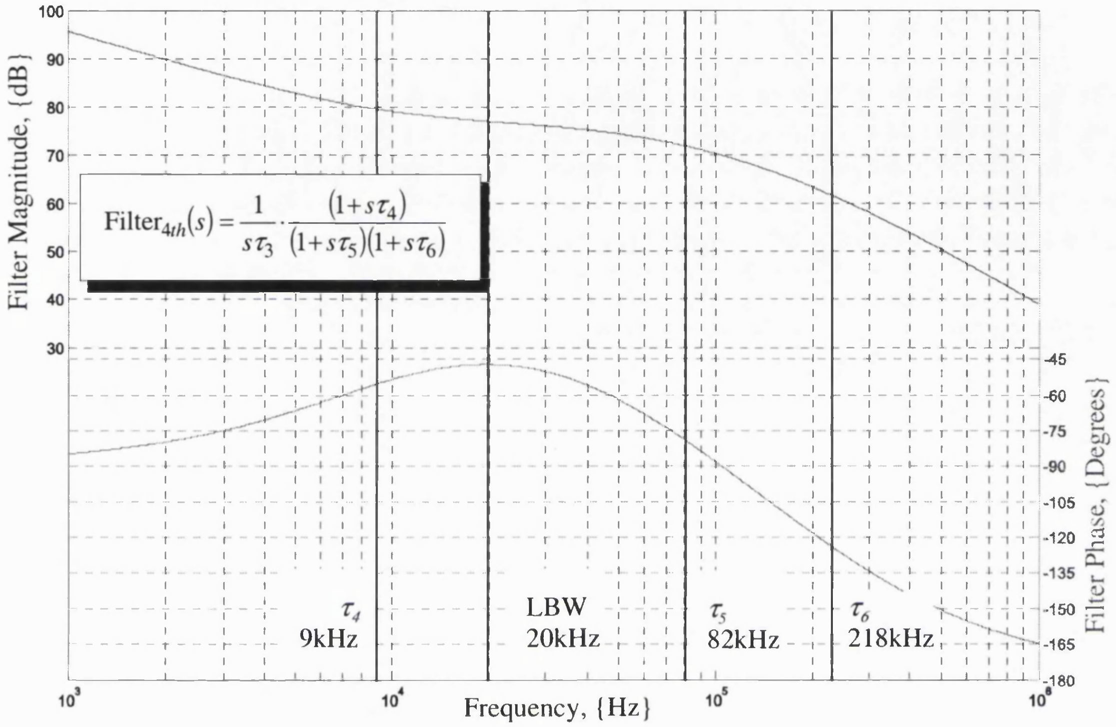


Figure 6.3 Typical type II, fourth-order loop filter Bode plot showing breakpoints.

The key to developing the desired mathematics is to start with the requirement that the phase and gain margin remain the same at the loop natural frequency for both the third and fourth-order loop filters. Using this fact then both the gain and phase of each of these two filters must be equal at the loop natural frequency. This assertion leads to the following sets of equations.

$$\frac{1+j\omega_n\tau_4}{j\omega_n\tau_3(1+j\omega_n\tau_5)(1+j\omega_n\tau_6)} = \frac{1+j\omega_n\tau_1}{j\omega_n\tau_0(1+j\omega_n\tau_2)} \qquad \text{Eq 6.12}$$

Fourth-order loop filter Third-order loop filter

$$\frac{1+\omega_n^2(\tau_4\tau_5+\tau_4\tau_6-\tau_5\tau_6)}{\omega_n\left[-\tau_4+\tau_5+\tau_6+\omega_s^2(\tau_4\tau_5\tau_6)\right]} = \frac{1+\omega_n^2\tau_1\tau_2}{\omega_n(-\tau_1+\tau_2)} \qquad \text{Eq 6.13}$$

Fourth-order phase Third-order phase

$$\frac{1+\omega_n^2\tau_4^2}{\omega_n^2\tau_3^2\left(1+\omega_n^2\tau_5^2\right)\left(1+\omega_n^2\tau_6^2\right)}=\frac{1+\omega_n^2\tau_1^2}{\omega_n^2\tau_0^2\left(1+\omega_n^2\tau_2^2\right)} \qquad \text{Eq 6.14}$$

Fourth-order gain Third-order gain

Equating coefficients from Eq 6.13 gives from this phase equation:

$$-\tau_4+\tau_5+\tau_6+\omega_s^2\left(\tau_4\tau_5\tau_6\right)=-\tau_1+\tau_2 \qquad \text{Eq 6.15}$$

$$\tau_4\tau_5+\tau_4\tau_6-\tau_5\tau_6=\tau_1\tau_2 \qquad \text{Eq 6.16}$$

Similarly equating coefficients from Eq 6.14 gives for the gain equation:

$$\tau_4=\tau_1 \qquad \text{Eq 6.17}$$

$$\tau_3^2\left(1+\omega_n^2\tau_5^2\right)\left(1+\omega_n^2\tau_6^2\right)=\tau_0^2\left(1+\omega_n^2\tau_2^2\right) \qquad \text{Eq 6.18}$$

Eq 6.17 is particularly useful because it establishes that τ_4 equals τ_1 permitting its calculation to be a straightforward extension of Eq 6.6 derived earlier.

$$\tau_4=\tau_1=\frac{\tan(PM)+\sec(PM)}{\omega_n} \qquad \text{Eq 6.19}$$

To derive τ_5, there are a number of routes that could be taken, however, if Eq 6.19 is substituted back into Eq 6.16, using the simplification

$$n=\sqrt{10^{\frac{Atten}{20}}-1} \qquad \text{Eq 6.20}$$

Making

$$\tau_6=\frac{n}{\omega_s} \qquad \text{Eq 6.21}$$

Then the desired expression for τ_5 is:

$$\tau_5=\frac{\omega_s\cos(PM)-n\omega_n(1+\sin(PM))}{\omega_n\left[\omega_s(1+\sin(PM))+n\omega_n\cos(PM)\right]} \qquad \text{Eq 6.22}$$

This crucial equation, which calculates τ_5, differs from Eq 6.7 used to calculate τ_2, by accounting for the consequence of adding the attenuation. The significance of Eq 6.22 is that it contains the attenuation term, n, as well as both the sampling and loop natural frequencies, ω_s and ω_n within its argument and hence accounts for the interaction of these two parameters in the calculation of this time constant.

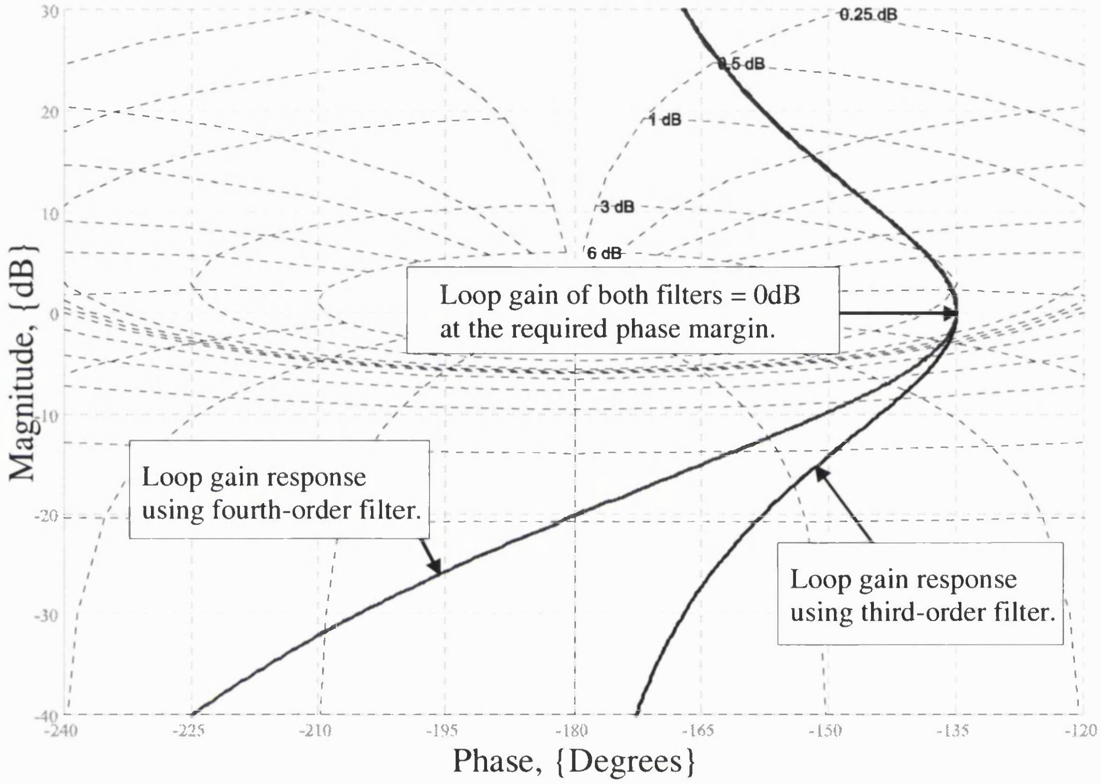


Figure 6.4 Nichols plot presenting loop gain trajectories using above design formulae

To calculate the loop filter gain term, τ_3, the cleanest method is to re-arrange Eq 6.18.

$$\tau_3 = \tau_0 \sqrt{\frac{\left(1+\omega_n^2\tau_2^2\right)}{\left(1+\omega_n^2\tau_5^2\right)\left(1+\omega_n^2\tau_6^2\right)}} \qquad \text{Eq 6.23}$$

A comparison of the third and fourth order loop gains using these new loop filter time constants gives the desired point of co-incidence at 0dB of both loop gains, for the required phase margin, Figure 6.4.

Deriving the physical loop filter component values depends upon the topology of loop filter used. Translating these time constants into loop filter values for the passive, charge pump driven loop filter given in Figure 6.2, involves equating like terms in the loop filter transfer equation with the generic time constant equivalent version given in Eq 6.11. The generic loop filter equation becomes:

$$\text{Filter}_{4th}(s) = \frac{(1+s\tau_4)}{s\left(\tau_3(\tau_5\tau_6)s^2 + \tau_3(\tau_5+\tau_6)s + \tau_3\right)} \qquad \text{Eq 6.24}$$

Which is compared with the passive loop filter transfer equation taken from Figure 6.2:

$$\text{Filter}_{4th}(s) = \frac{1+sR_2C_2}{s\left[(C_1C_2C_3R_2R_3)s^2 + (R_2C_2(C_1+C_2)+R_3C_3(C_1+C_2))s + (C_1+C_2+C_3)\right]} \qquad \text{Eq 6.25}$$

To give the four equations:

$$\begin{aligned} \tau_4 &= R_2C_2 \\ \tau_3\tau_5\tau_6 &= C_1C_2C_3R_2R_3 \\ \tau_3(\tau_5+\tau_6) &= R_2C_2(C_1+C_2)+R_3C_3(C_1+C_2) \\ \tau_3 &= C_1+C_2+C_3 \end{aligned} \qquad \text{Eq 6.26}$$

With only four identities and five required quantities, the actual determination of these loop filters has become a popular subject in the journals over the years, because there exists a range of valid component values, with no definitive answer. In order to determine five component values it is clear there are insufficient relationships in Eq 6.26, therefore if we assume capacitor C_1 is the variable, we arrive at the following four equations.

$$C_2 = \frac{\tau_3C_1(-\tau_4+\tau_5)(-\tau_4+\tau_6)}{C_1\tau_4^2 - \tau_3\tau_5\tau_6} \qquad \text{Eq 6.27}$$

$$C_3 = \frac{-(\tau_4C_1-\tau_5\tau_3)(C_1\tau_4-\tau_3\tau_6)}{C_1\tau_4^2 - \tau_3\tau_5\tau_6} \qquad \text{Eq 6.28}$$

$$R_2 = \frac{\tau_4(C_1\tau_4^2 - \tau_3\tau_5\tau_6)}{\tau_3C_1\left[(\tau_4^2+\tau_5\tau_6)-\tau_4(\tau_5+\tau_6)\right]} \qquad \text{Eq 6.29}$$

$$R_3 = \frac{-\tau_3\tau_5\tau_6(C_1\tau_4^2 - \tau_3\tau_5\tau_6)}{\tau_4C_1\left[(\tau_4C_1-\tau_3\tau_5)(\tau_4C_1-\tau_3\tau_6)\right]} \qquad \text{Eq 6.30}$$

Figure 6.5 is a plot of each of these equations. With reference to Figure 6.5, graphs a) and c), the limits on the range of C_2 and R_2, which give positive component values, is given by:

$$C_1 > \frac{\tau_3\tau_5\tau_6}{\tau_4^2} \qquad \text{Eq 6.31}$$

Further, with reference to Figure 6.5 graphs b) and d), the limits on the range of C_3 and R_3, which result in positive component values, is given by:

$$\frac{\tau_3\tau_6}{\tau_4} < C_1 < \frac{\tau_3\tau_5}{\tau_4} \qquad \text{Eq 6.32}$$

Noting that C_3 and R_3 values below $C_1 = \frac{\tau_3\tau_5\tau_6}{\tau_4^2}$ are disqualified because both R_2 and C_2 will be negative in this range. Consequently, the only valid range for C_1 that simultaneously gives positive component for all C_2, R_2, C_3 and R_3 is $\frac{\tau_3\tau_6}{\tau_4} < C_1 < \frac{\tau_3\tau_5}{\tau_4}$.

The value of R_3 shown in Figure 6.5, graph d), varies significantly across this range of C_1 values and in turn should ideally be minimised to restrict the quantity of loop filter thermal noise it will introduce. With this goal of reducing loop filter thermal noise, the value of C_1 at this minimum value of R_3 will be chosen as the starting point for calculating all the other loop filter component values. To find this value of C_1, the minimum value of R_3 is found by differentiating Eq 6.30 and equating the resultant expression to zero to find the value of C_1 that falls in the range $\frac{\tau_3\tau_6}{\tau_4} < C_1 < \frac{\tau_3\tau_5}{\tau_4}$. The exact answer to this calculation is very cumbersome offering no easy simplifications. However, simulations show that an approximation which holds for different ratios of loop natural frequency to sampling frequency ratios and at different phase margins, is:

$$C_1 \cong \frac{3}{5}\frac{\tau_3\tau_5}{\tau_4} \qquad \text{Eq 6.33}$$

With this value of C_1, Eq 6.27, Eq 6.28, Eq 6.29 and Eq 6.30 can be solved to give C_2, C_3, R_2 and R_3 respectively.

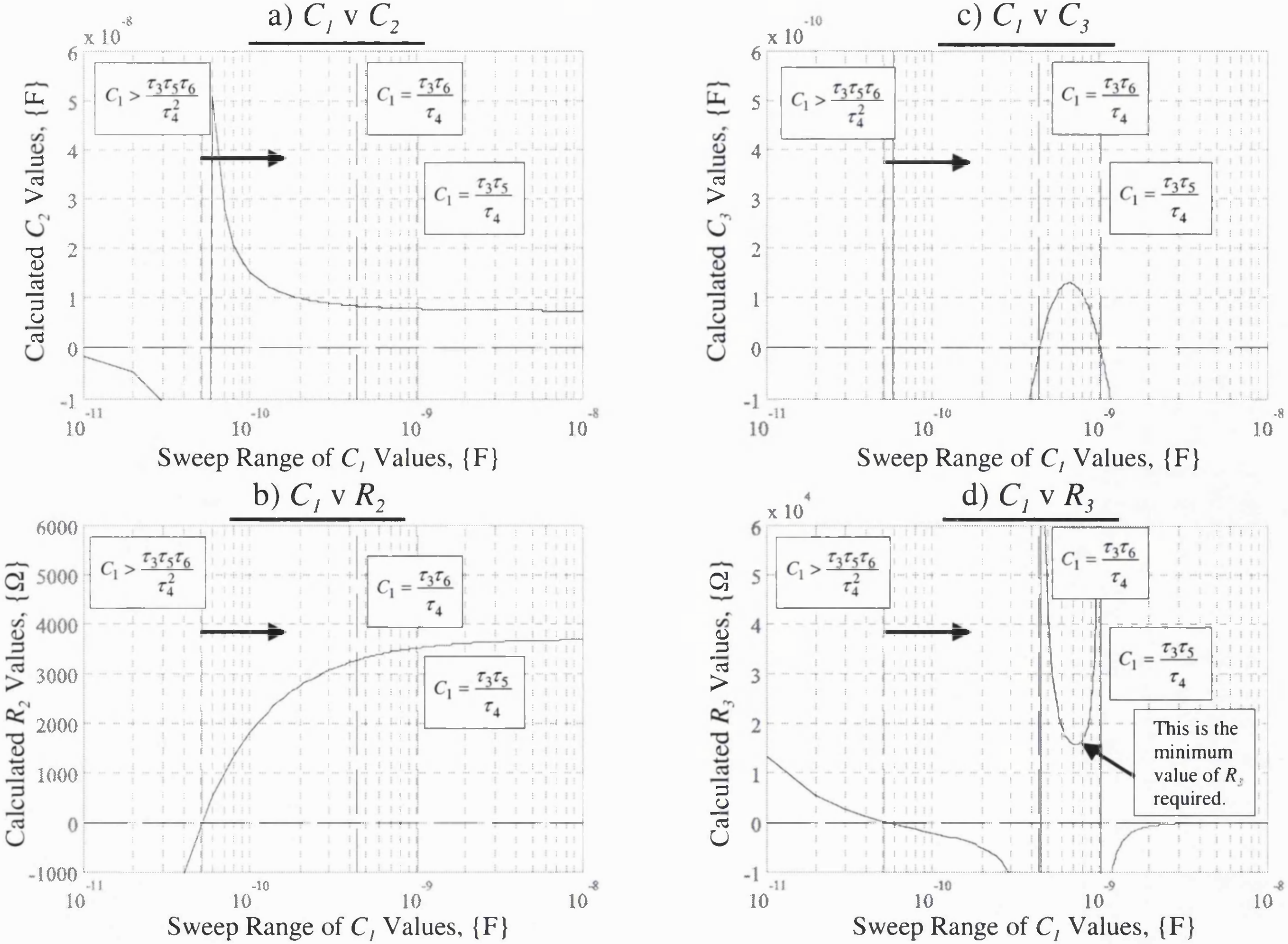


Figure 6.5 Variation of C_2, C_3, R_2 and R_3 passive loop filter component values

6.3 Maximum Allowable Attenuation for a Fourth-Order Loop Filter

After some simple investigations of the above results it quickly becomes clear that there exist both a maximum for the quantity *Atten* as well as a turning point. Intuition suggests these values will depend upon the ratio of loop natural frequency, ω_n, to sampling frequency, ω_s, for a given phase margin, *PM,* and will occur where the two pole time constants, τ_5 and τ_6, coincide.

The key to determining this maximum value of attenuation comes from recognising that, of all the above time constant calculations, only that of τ_5, Eq 6.22, contains all the key dependant terms of n, the attenuation term, ω_s, the sampling frequency and ω_n the loop natural frequency. Because Eq 6.18 is the only expression containing all the loop filter terms it was chosen as the starting point into which Eq 6.22 and Eq 6.21 were substituted.

$$\tau_3^2\left(1+\frac{\left(\omega_s\cos(PM)-n\omega_n(1+\sin(PM))\right)^2}{\left(\omega_s(1+\sin(PM))+n\omega_n\cos(PM)\right)^2}\right)\left(1+\frac{\omega_n^2n^2}{\omega_s^2}\right)=\tau_0^2\left(1+\omega_n^2\tau_2^2\right) \qquad \text{Eq 6.34}$$

Differentiating this expression with respect to n, and equating to zero to find the turning point gives, Eq 6.35:

$$\tau_3^2\left(\begin{array}{l}-2\dfrac{\left(\omega_s\cos(PM)-n\omega_n(1+\sin(PM))\right)\omega_n(1+\sin(PM))}{\left(\omega_s(1+\sin(PM))+n\omega_n\cos(PM)\right)^2}\\[2ex] -2\dfrac{\left(\omega_s\cos(PM)-n\omega_n(1+\sin(PM))\right)^2\omega_n\cos(PM)}{\left(\omega_s(1+\sin(PM))+n\omega_n\cos(PM)\right)^3}\end{array}\right)\left(1+\frac{\omega_n^2n^2}{\omega_s^2}\right)$$
$$+2\frac{\tau_3^2\left(1+\dfrac{\left(\omega_s\cos(PM)-n\omega_n(1+\sin(PM))\right)^2}{\left(\omega_s(1+\sin(PM))+n\omega_n\cos(PM)\right)^2}\right)\omega_n^2n}{\omega_s^2}=0 \qquad \text{Eq 6.35}$$

From this solution there exist four values of n of which only one is real and positive:

$$n=\frac{\omega_s}{\omega_n}\frac{\cos(PM)-\sqrt{2-2\sin(PM)}}{\sin(PM)-1} \qquad \text{Eq 6.36}$$

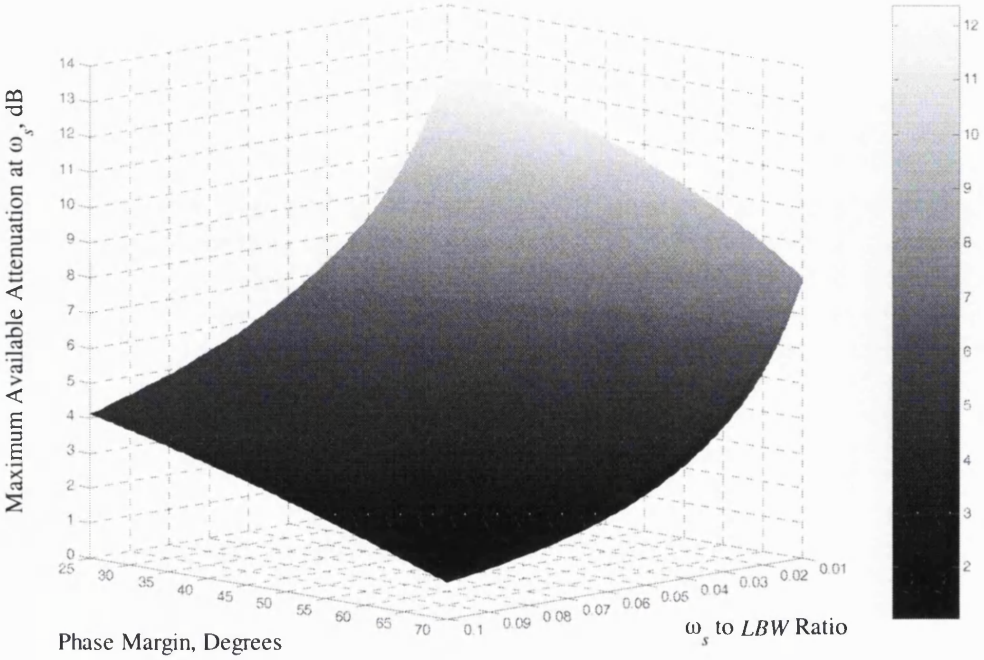


Figure 6.6 Plot showing variation of maximum attenuation achievable for a given sampling frequency to loop natural frequency ratio at different phase margins

Reviewing Figure 6.6 taken from Eq 6.36, shows that as the sampling frequency becomes smaller relative to the loop natural frequency, greater attenuation can be achieved and confirms that smaller phase margins increase the loop filter roll-off characteristic, allowing more attenuation. Generally, lower phase margins are avoided to limit loop peaking and potential loop instability under adverse closed loop operating conditions.

6.4 Accounting for the Effects of Sampling in Passive Loop Filter Design

The objective here is to compensate for the effects of sampling within the phase locked loop, so that a suitable adjustment can be made to the required phase margin, before using the linear equations derived in Sections, 6.2.1 and 6.2.2.

From the work undertaken by Crawford, [92], he suggests that a suitable phase correction will depend upon the ratio of loop natural frequency to phase margin. This approximation uses the sampling term, e^{sT_s}, where s is used to express the loop natural frequency and T_s the sampling rate.

Plotting this simple relationship gives a phase change characteristic for different loop natural frequency to sampling frequency ratios shown in Figure 6.7.

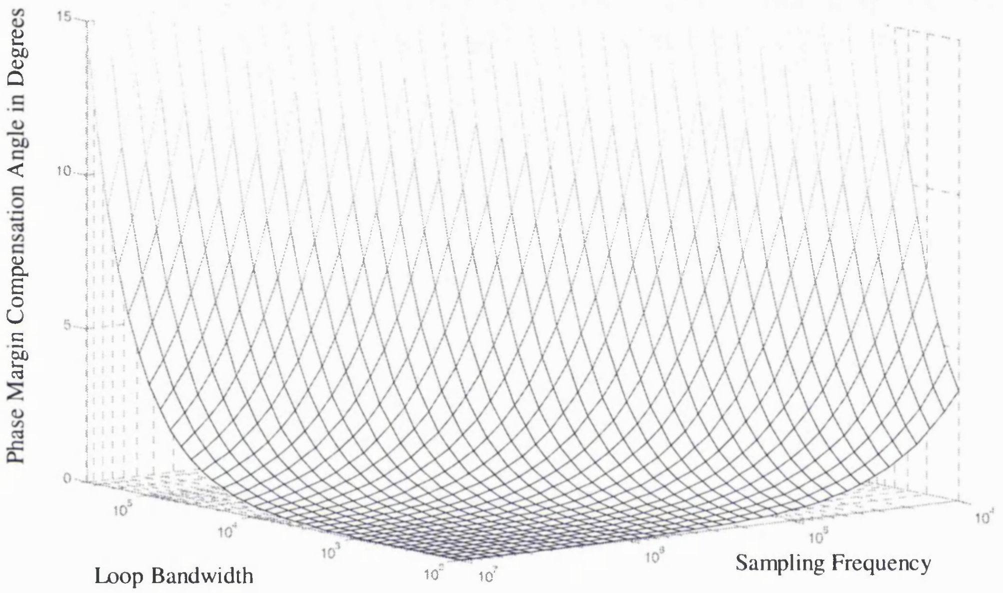


Figure 6.7 Phase margin variation for differing loop natural frequency to sampling frequencies.

Figure 6.7 indicates that for a low loop natural frequency relative to the sampling frequency, only minimal phase margin compensation is required, and indeed can be ignored, as stated by Gardner, [93]. However as the sampling frequency approaches the loop natural frequency the amount of phase margin correction starts to increase. It is this correction factor that is required in designs where the emphasis is on lock time at the expense of a small increase in noise.

By pre-adjusting the design phase margin and loop gain value with these sampling compensation values, before proceeding to complete the loop filter design using the linear loop filter calculations given in Section 6.2.1, and 6.2.2, the effects of sampling can be compensated for. These corrections help to restore the phase margin to the required loop natural frequency, to within the confines of practical loop filter and other loop component gain tolerances.

6.5 An Evaluation of the Loop Filter in the Sampled Loop

The purpose of this section is to mathematically evaluate the fourth-order loop filter design, in the sampled loop, using the sampling phase compensation term. Using sampled loop mathematics, the sampled loop gain expressions for both the third and fourth-order loops are derived. Taking the sampled fourth-order loop expression, the new fourth-order loop filter design with the sampling compensation is compared to an example loop design that requires a 45° phase margin.

When considering the convolution sum for determining the sampled loop gain, generally a limited range of "*n*" summation values will give an acceptable answer, Eq 6.37.

$$GH^*(s) = K \sum_{n=-\infty}^{\infty} GH(s - j \cdot n \cdot \omega_s) \qquad \text{Eq 6.37}$$

Instead of using the convolution summation, the impulse invariant approach will be adopted here, given the assumptions that the PLL natural frequency is well below the sampling rate and the system is stable, [94]. The resulting expression is a closed form of the sampling expression, which is simpler than using convolution.

The system function for the continuous-time type II, third-order loop[20] is:

$$GH_{3rd}(s) = \frac{1 + s\tau_1}{s^2\tau_0(1 + s\tau_2)} \cdot K \qquad \text{Eq 6.38}$$

where the constant *K* represents the non-frequency dependent constants of the phase detector, VCO gain and n-divider ratio in the overall loop gain. This expression is then converted back to give the impulse response using the inverse Laplace transform[21].

[20] This loop gain contains two integrators and highest denominator exponent of three, making a type two, third-order loop.

[21] The assumption here is the system is causal and stable to satisfy the requirements for a Laplace transform.

$$gh_{3rd}(t) = \frac{K}{\tau_0}\left(\tau_1 - \tau_2 + t + (\tau_2 - \tau_1)\exp\left(\frac{-t}{\tau_2}\right)\right)u(t) \qquad \text{Eq 6.39}$$

Where *u*(*t*) is the unit step function. Sampling $gh_{3rd}(t)$ with a sampling period T_s, gives the impulse response of the discrete-time system:

$$gh_{3rd}(kT_s) = \frac{K}{\tau_0}\left(\tau_1 - \tau_2 + kT_s + (\tau_2 - \tau_1)\exp\left(\frac{-kT_s}{\tau_2}\right)\right)u(n) \qquad \text{Eq 6.40}$$

Taking the z-transform of Eq 6.40, the system function $GH_{3rd}(z)$ of the discrete-time loop gain is found after some simplification to be:

$$GH_{3rd}(z) = \frac{K}{\tau_0}\left[z\frac{\left[(\tau_1 - \tau_2 + T_s) - (\tau_1 - \tau_2)\exp\left(\frac{-T_s}{\tau_2}\right)\right]z + \left[(\tau_1 - \tau_2 - T_s)\exp\left(\frac{-T_s}{\tau_2}\right) - (\tau_1 - \tau_2)\right]}{(z-1)^2\left(z - \exp\left(\frac{-T_s}{\tau_2}\right)\right)}\right] \qquad \text{Eq 6.41}$$

Finally the impulse invariant response of the loop gain can be determined using the substitution z = e^{sT_s}, giving:

$$GH^*_{3rd}(s) = \frac{T_s \cdot K}{\tau_0}\left[\exp(sT_s)\frac{\left[(\tau_1 - \tau_2 + T_s) - (\tau_1 - \tau_2)\exp\left(\frac{-T_s}{\tau_2}\right)\right]\exp(sT_s) + \left[(\tau_1 - \tau_2 - T_s)\exp\left(\frac{-T_s}{\tau_2}\right) - (\tau_1 - \tau_2)\right]}{(\exp(sT_s) - 1)^2\left(\exp(sT_s) - \exp\left(\frac{-T_s}{\tau_2}\right)\right)}\right] \qquad \text{Eq 6.42}$$

This is the desired impulse invariant expression for the third-order loop gain. Contained within this formula is the unit delay associated with the sampling process, e^{sT_s}, which will vary for a given loop natural frequency to sampling frequency ratio, and also the system time constants τ_1 and τ_2, which are themselves dependent upon the phase margin of the loop filter. These are the effects that the hold mathematics of section 6.4 are trying to compensate for.

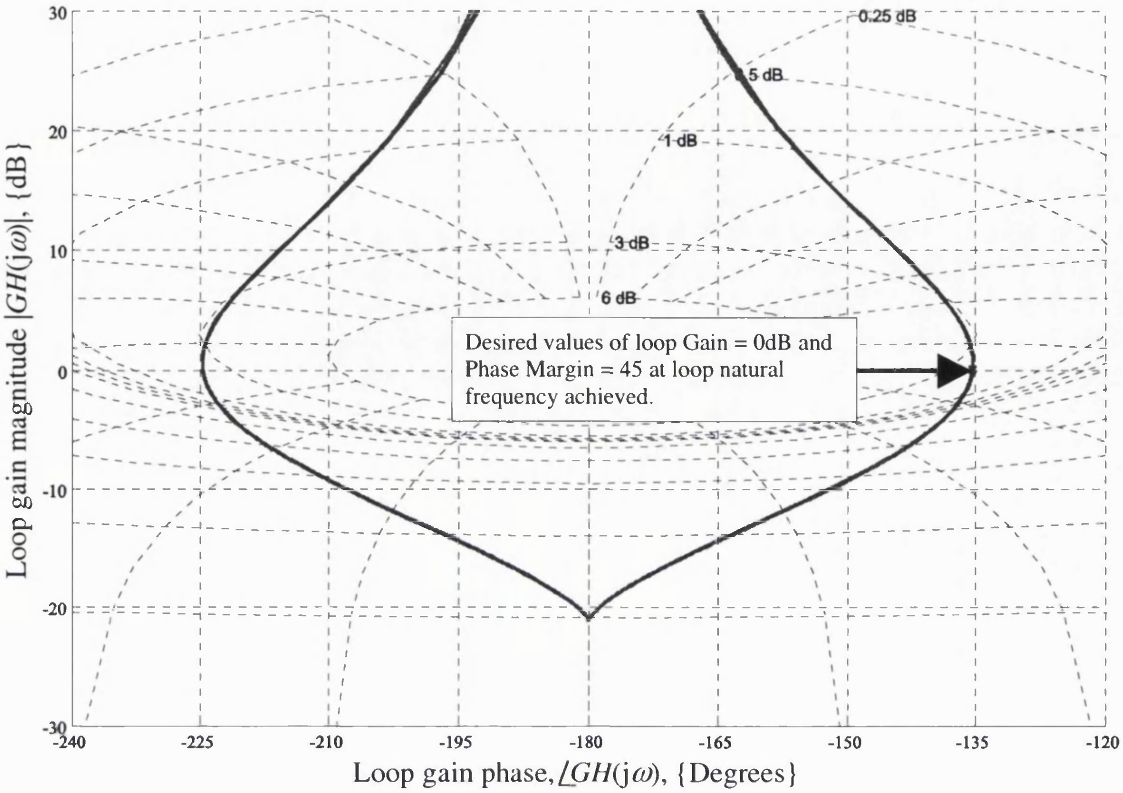
This same mathematical sequence can be applied to the type II, fourth-order loop linear transfer response:

$$GH_{4th}(s) = \frac{1 + s\tau_4}{s^2\tau_3(1 + s\tau_2)(1 + s\tau_3)} \cdot K \qquad \text{Eq 6.43}$$

to arrive at the rather lengthy full expression for the sampled loop gain response, Eq 6.15:

$$
\begin{aligned}
GH_{4th}(s) &= \frac{T_s\,K\,e^{sT}\left(Ae^{2sT_s}+Be^{sT_s}+C\right)}{\left(\tau_5-\tau_6\right)\left(e^{sT_s}-e^{-\frac{T_s}{\tau_5}}\right)\left(e^{sT}-e^{-\frac{T_s}{\tau_6}}\right)\left(e^{sT_s}-1\right)^2} \\
\text{where } A &= \left(\tau_4\tau_6-\tau_6^2\right)e^{-\frac{T_s}{\tau_6}}+\left(\tau_5^2-\tau_4\tau_5\right)e^{-\frac{T_s}{\tau_6}} \\
&\quad +T_s\left(\tau_5-\tau_6\right)+\tau_4\tau_5-\tau_4\tau_6-\tau_5^2+\tau_6^2 \\
B &= \left(T_s\left(-\tau_5+\tau_6\right)-\tau_4\tau_5-\tau_4\tau_6+\tau_5^2+\tau_6^2\right)e^{-\frac{T_s}{\tau_6}} \\
&\quad +\left(T_s\left(-\tau_5+\tau_6\right)+\tau_4\tau_5+\tau_4\tau_6-\tau_5^2-\tau_6^2\right)e^{-\frac{T_s}{\tau_5}} \\
&\quad +\left(\tau_4\tau_5-\tau_4\tau_6-\tau_5^2+\tau_6^2\right)e^{-\frac{T_s\left(\tau_5^2+\tau_6^2\right)}{\tau_5\tau_6}} \\
&\quad -\tau_4\tau_5+\tau_4\tau_6+\tau_5^2-\tau_6^2 \\
C &= \left(\tau_4\tau_5-\tau_6^2\right)e^{-\frac{T_s}{\tau_6}}+\left(-\tau_4\tau_6+\tau_6^2\right)e^{-\frac{T_s}{\tau_5}} \\
&\quad +\left(-\tau_4\tau_5+\tau_4\tau_6+\tau_5^2-\tau_6^2\right)e^{-\frac{T_s\left(\tau_5^2+\tau_6^2\right)}{\tau_5\tau_6}}
\end{aligned}
\qquad \text{Eq 6.44}
$$

Using the same set of loop gain parameters as in Figure 6.1, a sampled system that employs the phase margin correction factors is plotted on a Nichols plot in Figure 6.8 from Eq 6.44. This plot uses the impulse invariant mathematics of Eq 6.42 to represent the sampled loop.



Desired values of loop Gain = 0dB and Phase Margin = 45 at loop natural frequency achieved.

Figure 6.8 Nichols plot of type II, fourth-order sampled loop trajectory

With reference to Figure 6.8 the following characteristics of the sampled PLL are observed, [72]:

1) The sampled nature of the loop gain, $GH(s)^*$, takes on the same value at congruent points, with each successive sample. Thus, the value of the loop gain, $\sum_{n=-\infty}^{\infty} GH(j\omega + jn\omega_s)$ for the frequency range $-\infty \leq \omega \leq \infty$ is a periodic function with period equal to $j\omega_s$ and is the same as those for the frequency range $0 \leq \omega \leq \omega_s$.

2) The value of $\sum_{n=-\infty}^{\infty} GH(j\omega + jn\omega_s)$ at $\omega = \omega_s/2$ is real and gives the gain margin.

At $\omega = \omega_s/2$,

$$\begin{aligned}\sum_{n=-\infty}^{\infty} GH(j\omega + jn\omega_s) &= GH\left(\frac{j\omega_s}{2}\right) + GH\left(\frac{j3\omega_s}{2}\right) + \ldots \\ &\quad + GH\left(\frac{-j\omega_s}{2}\right) + GH\left(\frac{-j3\omega_s}{2}\right) + \ldots. \\ &= 2\left[\Re\left(GH\left(\frac{j\omega_s}{2}\right)\right) + \Re\left(GH\left(\frac{j3\omega_s}{2}\right)\right) + \ldots.\right] \\ &= \text{a real quantity}\end{aligned} \qquad \text{Eq 6.45}$$

3) The gain margin for a sampled PLL analysis differs noticeably to that of a linear PLL.

6.6 Summary

This chapter started with an accepted analysis for a third-order loop filter design and proceeded to find a manageable set of design equations for the design of a fourth-order loop filter. The useful result found in Eq 6.22, included the expected variables of sampling frequency, loop natural frequency, phase margin and attenuation that would be required to calculate this time constant. When comparing the results given by Eq 6.22 with that of Eq 6.4 used in the design of a third-order loop filter, it is clear that the locations of the zero and pole are no longer symmetrical about the loop natural frequency, to account for the extra attenuation. The translation of these time constants into a variety of different active or passive loop filter topologies is relatively straightforward, with the exception of the passive charge pump driven filter. To explore this translation of time constants to passive loop filter components, a precursory examination of each component value relative to C_1, highlighted a single minima in the value of R_3, thus giving the obvious clue to reducing the loop filter thermal noise in the determination of the loop filter values, Figure 6.5.

Realising that maximum attenuation can be achieved in the design of a fourth-order loop filter when its two poles coincide, this chapter proceeded to determine this maximum attenuation value in Eq 6.36, which gives the interesting plot, Figure 6.6.

Following the linear loop filter design is an approximation technique aimed at reducing the phase margin degradation of the loop natural frequency, in the presence of sampling.

In order to verify the mathematics, the impulse invariant approach was chosen in preference to convolution, because it gives a closed expression for the loop gain, which can readily be applied to any ratio of frequencies within a sampled system, without having to consider the number of summations required in the convolution integral. Using both the type II, third and fourth order sampling loop gain expressions, Figure 6.8 was derived which indicates the usefulness of these sampling approximations when applied to loop filter design.

7 HYBRID PLL/DDS BASED SYNTHESISER

7.1 Introduction

This chapter starts with a generic specification and proceeds by outlining the reasoning behind the design of the hybrid PLL/DDS based synthesiser to meet this specification, based on the foundation work of the previous chapters. The emphasis within this section is on measurements made on this complete synthesiser architecture which are compared against predictions made using the theory developed in the previous chapters, as a vehicle to validate this investigation.

It was during this practical work that discussions began with Analog Devices encouraging them to develop a business case for the development of their next generation DDS chip, the AD9858, [6]. Analog Devices became very interested in the architecture used in this chapter and have incorporated all the elements necessary to build a single chip version of this hybrid PLL/DDS based synthesiser in their AD9858 DDS chip, appendix A6.

As a further consequence of this practical work one of the two patents was developed and is now proceeding through prosecution in the European patent office.

It is to premature to arrive at the conclusion that this type of synthesiser is sufficient to provide the universal base station compliant synthesiser, however, as the following results show, this synthesiser offers exceptional phase noise, spurious and lock time performance.

7.2 System Specification

The generic specification given in Table 7.1 embodies most of the demanding GSM requirements and will be used as the basic design criteria for the following synthesiser design.

	Parameter	Specification	Units	Comments
1	Frequency Range	1540 to 1615	MHz	Dependant on system intermediate frequency value.
2	Frequency Resolution	200	kHz	GSM Channel Spacing
3	EVM	<1	%	Measures the integrated RMS phase error.
4	Phase Noise			
	a) 600kHz	<-125	dBc/Hz	
	b) >20MHz	<-155	dBc/Hz	For wide band noise emissions.
5	Lock Time to < 6° Phase error	<10	µS	To fit in the guard band period between GSM bursts.
6	Spurious			
	a) <200kHz	<-60	dBc	Taken from EVM measurements.
	b) ≥200kHz<1MHz	<-80	dBc	
	c) ≥1MHz	<-100	dBc	Adjacent channel interference requirement.

Table 7.1 Generic synthesiser specification for the hybrid Synthesiser

7.3 System Design Criteria

7.3.1 Preferred Solution

Accepted at the very beginning of this design was the underlying need to reduce the in-band phase noise level below the specified –125dBc/Hz at 600kHz offset, in order to raise the loop bandwidth to around 250kHz. Earlier experimentation had suggested a minimum 250kHz loop bandwidth was required to meet the sub 10μS lock time specifications. Given this design requirement as the basis for this synthesiser and knowing that the phase frequency detector limits the in-band phase noise, a thorough understanding of the PFD was undertaken, chapter 5. The conclusion from this work was that a low n-division value in the feedback path would be required to minimise the PFD noise contribution, Figure 5.7.

From the outset of this development, fractional-n frequency synthesisers were dismissed because of their disappointing performance. One hypothesis, proposed here, considers how the two samplers and the sigma delta modulator interact within the closed loop to alias noise into the PLL pass band. This process can be explained by considering a sigma delta PLL as equivalent to a closed loop, multi-rate sampling system, [80], with two non-integer related samplers[22] and a pseudo-random modulator. This conjecture begins by asserting that the collection of varying offset sigma delta and constant offset reference spurs are generated as predicted, however they then pass around the loop for further modulation and re-sampling. As a consequence of multiple passes around the loop and the sigma delta modulation process, a layer of wideband spurious products result, which result in a "sea" of in-band spurs that artificially raises the PFD in-band phase noise plateau. Further, because any phase noise measurement system is unlikely to be phase coherent with the sigma delta modulator, it is impossible to isolate these smaller spurs. Therefore, the conclusion is this aliasing degrades the measured in-band phase-noise plateau of a sigma delta fractional-n synthesiser, when compared against the theoretically expected in-band phase-noise plateau, Figure 3.3. Unfortunately, a proof of this interesting hypothesis is beyond the scope of this work.

[22] Note the mean fractional ratio between the two samplers remains constant because the system is phase locked.

With this conclusion, DDS techniques were considered despite their much larger but predictable spurs. Generally, these higher spur levels have relegated DDS to second place in frequency synthesis. However the predictability of these spur offsets was regarded as a distinct advantage because they can easily be managed by sensible design and so DDS was judged the preferred synthesiser technique. Furthermore, DDS is an all-digital solution offering very fast lock times to very high levels of accuracy with the additional feature of accurate phase modulation, should this be necessary in any applications.

The SP2002 DDS plus DAC integrated chip, [44], was chosen as the DDS test unit for this system because of its high operational frequency and integrated DAC. From the SP2002 integral 8-bit DAC specification, the potential for spurs as high as -48dBc at the output dictated that a digital divider at the DAC output, was essential in this design to suppress these spur levels. To further improve the attenuation of these DDS spurii an analogue PLL architecture was considered necessary to act as a tracking filter. This analogue PLL would use the DDS output as its reference input, but must possess the essential property of unity gain from this reference input to the VCO output. In practice, these worst case DDS spurs only occur at the low order fractional values of the master clock frequency, e.g. ½, ⅓, ¼ etc., [appendix A3]. Although it could be argued the DDS alone could synthesise the required output frequency, using digital dividers at the DAC output gives three distinct advantages:

1. The spurious levels are suppressed by the division action, however, their frequency offset are not altered:

Using the accepted expression for a carrier phase modulated by a sinusoid:

$$v_1(t) = A(t)\cos(\omega_c t + \beta \sin \omega_m t) \qquad \text{Eq 7.1}$$

Where β is the modulation index. Then the phase component is:

$$\phi(y_1(t)) = \omega_c t + \beta \sin \omega_m t \qquad \text{Eq 7.2}$$

From which the frequency is:

$$f_1(t) = \frac{1}{2\pi}\frac{d\phi}{dt} = \frac{1}{2\pi}(\omega_c + \beta\omega_m \cos \omega_m t) \qquad \text{Eq 7.3}$$

When this signal frequency is divided by the r-divider value, R, where $f_2 = f_1/R$, then:

$$f_2(t)=\frac{1}{R}f_1(t)=\frac{1}{2\pi}\left(\frac{1}{R}\omega_c+\frac{1}{R}\beta\omega_m\cos\omega_m t\right)$$
$$=\frac{1}{2\pi}\left(\frac{1}{R}\omega_c+\frac{1}{R}\beta\omega_m\cos\omega_m t\right) \qquad \text{Eq 7.4}$$

Integrating this expression to give the phase of this new output frequency:

$$\phi(y_2(t))=\frac{1}{R}\omega_c t+\frac{1}{R}\beta\sin\omega_m t \qquad \text{Eq 7.5}$$

The phase modulated carrier at the divider output is:

$$v_2(t)=A(t)\cos\left(\frac{1}{R}\omega_c t+\frac{1}{R}\beta\sin\omega_m t\right) \qquad \text{Eq 7.6}$$

Comparing Eq 7.6, above, with Eq 7.1 earlier, the conclusion is the value of the modulation index, *β*, has now been adjusted by the division value *R*, but not the sideband frequency, *ω_m* offset from the carrier *ω_c*. The amplitude of the sideband frequency, *ω_m* is reduced by the division value *R*.

Using Bessel functions the spectrum of the original signal, Eq 7.1 is given by:

$$v_1(t)=A(t)\cos(\omega_c t+\beta\sin\omega_m t)$$
$$=A(t)\sum_{i=-\infty}^{\infty}J_i(\beta)\cos(\omega_c t-j\omega_m)t \qquad \text{Eq 7.7}$$

Whereas the equivalent spectrum of the divided down signal, Eq 7.6 is:

$$v_2(t)=A(t)\sum_{i=-\infty}^{\infty}J_i\left(\frac{1}{R}\beta\right)\cos(\omega_c t-j\omega_m)t \qquad \text{Eq 7.8}$$

From which the change in spur ratio power ratio can be calculated as:

$$\text{Spur Power Ratio}=\left(\frac{J_i\left(\frac{1}{R}\beta\right)}{J_i(\beta)}\right)^2 \Rightarrow 20\mathrm{Log}_{10}\left(\frac{1}{R}\right) \qquad [dB] \qquad \text{Eq 7.9}$$

Thus the division of the input signal by a factor of *R*, reduces the power in the PM spur at frequency offset, ω_m, from the divider output carrier frequency ω_c/R.

2. Introducing an r-divider at the DAC output of the DDS reduces the frequency resolution required of the DDS. This in turn increases the minimum offset of the DDS spurious, pushing them beyond the loop bandwidth of the analogue PLL, providing the DDS clock source, f_{Clk}, is set to an integer value of this frequency resolution.

3. The digital input to the r-divider is a natural limiting action, which will suppress any DAC induced, AM components.

To achieve a sub 10µS lock time, the minimum loop bandwidth was determined by measurement to be around 250kHz, which is sufficiently close to the 600kHz phase noise specification that the in-band phase noise would need to be circa –130dBc/Hz. This value was determined by simulation and also accounts for the noise peak at the loop natural frequency. By taking the 75MHz tuning range requirement, the maximum PFD input frequency was determined using Eq 5.5, to be 100MHz to limit the in-band noise to less than –130dBc/Hz. In order to minimise the DDS reference spurs the PLL had to have no multiplicative gain, which dictated a PLL with no dividers in the feedback path, Eq 5.6. This limits the options to an analogue mix down, in the feedback path of the PLL, which, according to Eq 5.6, has the added benefit of minimising PFD noise floor amplification, essential to this design. From this reasoning, the system in Figure 7.1 was proposed.

From Figure 7.1, with a low side injection mixer, the frequency planning gives an output frequency dependant upon the DDS binary input word, M, the width of the DDS accumulator, A, the r-divider value, R and the fixed frequency clock source, f_{Clk}:

$$\begin{aligned} f_o &= f_{Clk} - \frac{1}{R} f_{Clk} \frac{M}{2^A} \\ &= f_{Clk}\left(1 - \frac{M}{R \times 2^A}\right) \end{aligned} \qquad \text{Eq 7.10}$$

The problem of DDS spurious is usually limited by using a reconstruction filter at the DAC output, however for this application the PLL essentially becomes a high Q tracking filter. The Q factor of this tracking filter is set by the loop filter bandwidth within the PLL. Thus the PLL further attenuates the DDS spurious products that always fall outside the PLL loop bandwidth.

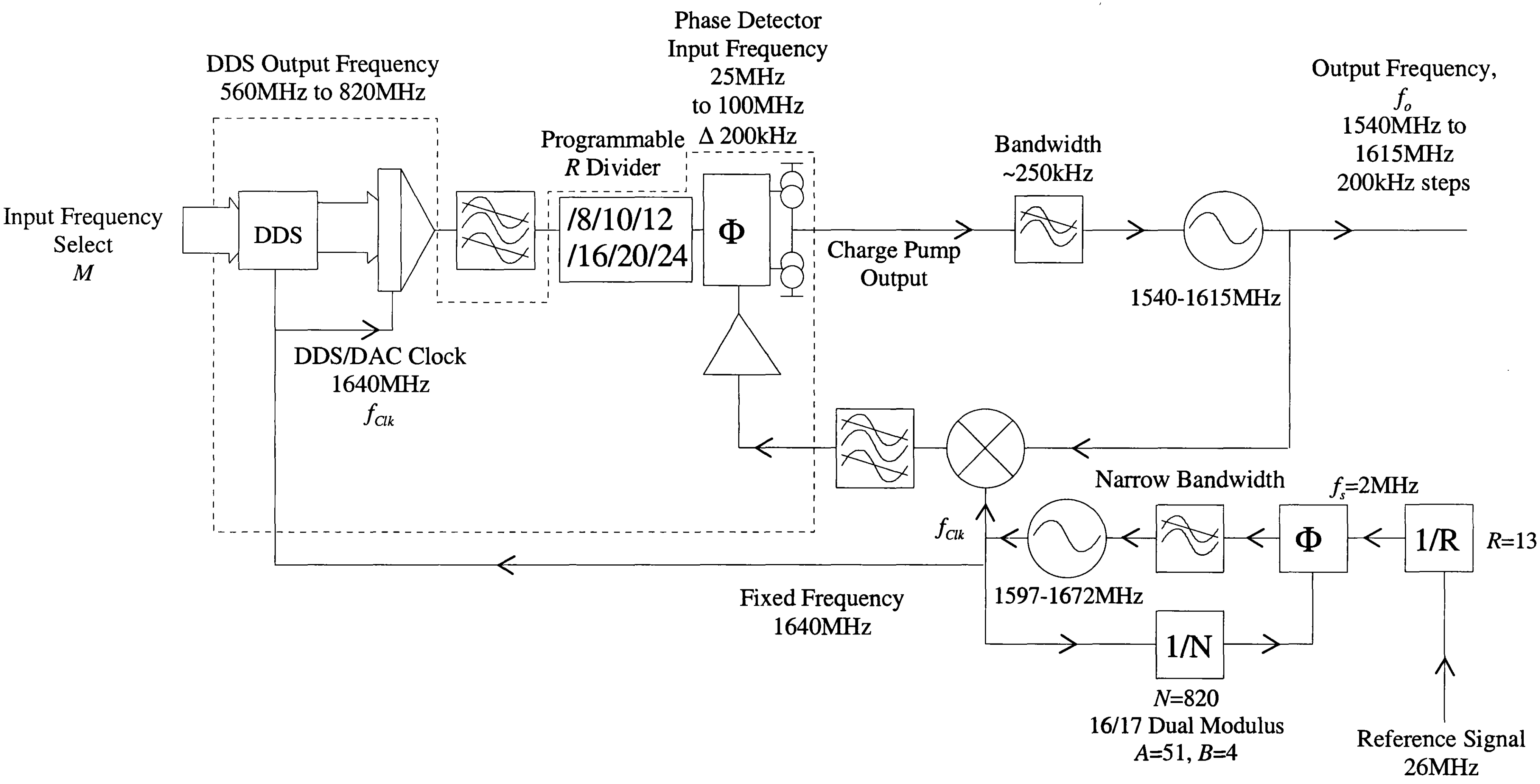


Figure 7.1 Block diagram of proposed synthesiser

To cover the 25MHz to 100MHz reference output range the problem was to find a sensible frequency plan, which maximised the DDS and DAC related spurious offsets, whilst providing a suitable clock source to the available DDS. After some searching the following frequency plan was established using a 1640MHz master clock source that covered the full 25MHz to 100MHz PFD input frequency range. The minimum DDS output frequency was chosen as 560MHz whilst the maximum DDS output frequency was limited to 800MHz, based on the spurious analysis in Figure 7.2. These limits gave the divided down output frequency range shown, with the choice of division ranges based on keeping the DDS spur offsets, as far from the output signal as possible, Table 7.2.

Division Value (Attn)	Step Size	Minimum Spur	f_{DDS} Minimum 560MHz	f_{DDS} Maximum 800MHz	Minimum to maximum input and output frequencies chosen.
÷24 (27.6dB)	4·8MHz	1·6MHz	23·3MHz ↔ 33·3MHz		600MHz ↔ 667·2MHz 25MHz ↔ 27·8MHz
÷20 (26dB)	4MHz	4MHz	28MHz ↔ 40MHz		560MHz ↔ 800MHz 28MHz ↔ 40MHz
÷16 (24dB)	3·2MHz	1·6MHz	35MHz ↔ 50MHz		643·2MHz ↔ 800MHz 40·2MHz ↔ 50MHz
÷12 (21.6dB)	2·4MHz	800kHz	46·6MHz ↔ 66·6MHz		602·4MHz ↔ 669·6MHz 50·2MHz ↔ 55·8MHz
÷10 (20dB)	2MHz	2MHz	56MHz ↔ 80MHz		560MHz ↔ 798MHz 56MHz ↔ 79·8MHz
÷8 (18dB)	1·6MHz	1·6MHz	70MHz ↔ 100MHz		640MHz ↔ 800MHz 80MHz ↔ 100MHz

Table 7.2 DDS output frequency range and r-divider settings

1640MHz was chosen as the master clock frequency because this frequency is an exact integer value of 800kHz and 1MHz guaranteeing the DDS will not produce spurious products at lower frequency offsets from its output carrier, [95]. Given this configuration all spurious products remain outside the PLL bandwidth and are attenuated by both the digital r-divider and the loop filter. In this scheme the r-divider is asked to change its value to limit the maximum output tuning range required of the DDS, allowing the DDS to operate in the band between ½ and ⅓ of its clock frequency, i.e. 820 and 546·6MHz, Figure 7.2. These two fractional values of the clock frequency produce high level spurs, which cannot be adequately attenuated by the PLL loop filter and cannot be filtered by any passive reconstruction filter. The r-divider DAC spurious attenuation values are also given in Table 7.2.

From the frequency plan of Figure 7.1 and using the mixer intermodulation analysis described in section appendix A3, Figure 7.2 is a prediction of the spurious products expected at different DDS output frequencies. For this analysis, the DDS input clock was fixed at a frequency of 1640MHz and then swept across the DDS output frequency band. In practice, the spurious products sweep rapidly through the loop bandwidth of the PLL. This will limit the frequency offset of troublesome spurious products to 800kHz either side of the diagonal line representing the desired DDS output signal frequency.

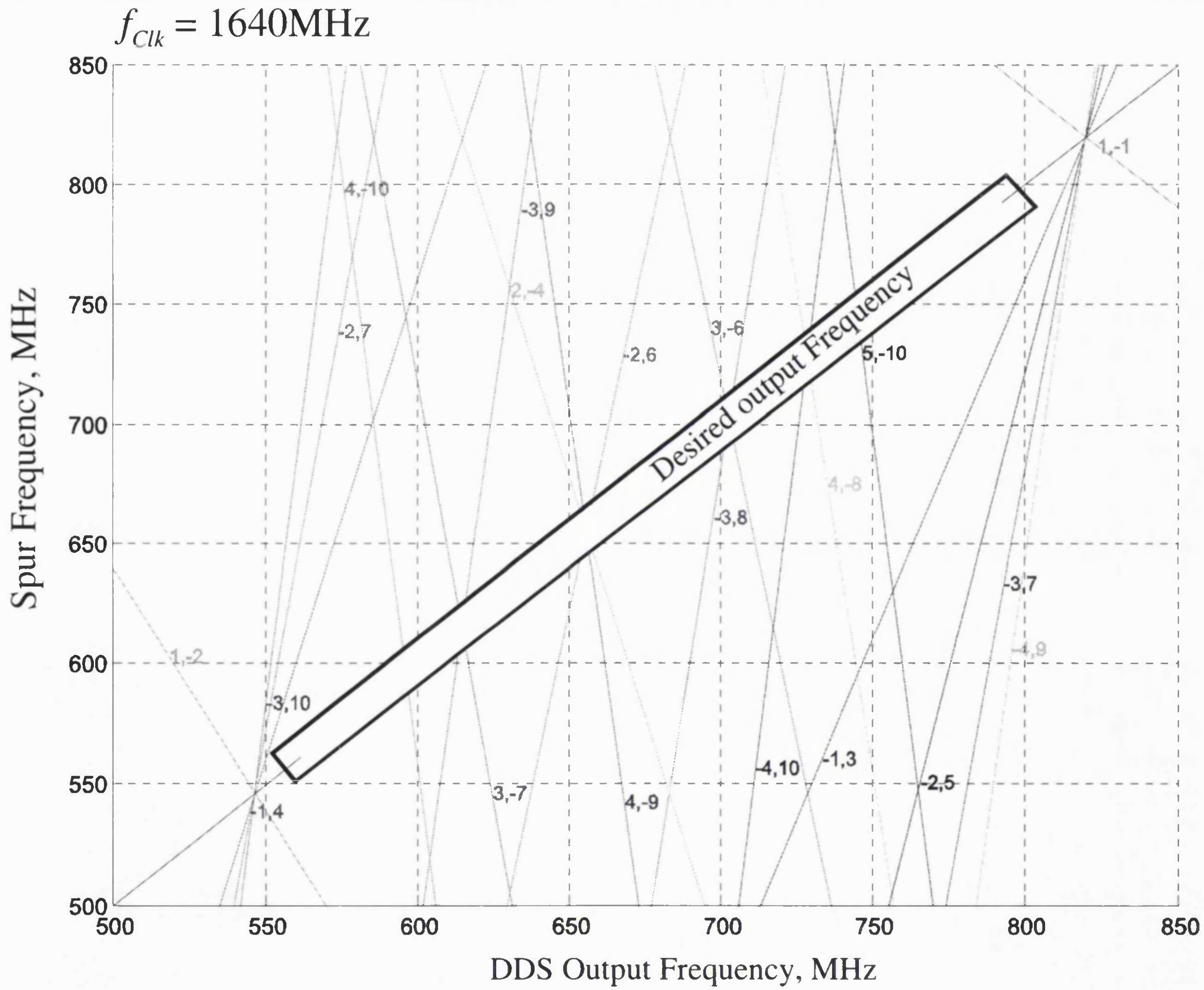


Figure 7.2 Predicted intermodulation, (spurious) product output from DDS

7.3.2 Tested Solution

Although the SP2002 integrated DDS and DAC is specified to clock up to 1600MHz the output of any one of the four 8-bit DACs is a multiplexed version of the accumulator value, [43]. Consequently, the maximum available output frequency from any one DAC is limited to one quarter of the clock input, f_{Clk}. This limitation prevents the device being used with the preceding frequency plan, instead the following frequency plan was adopted and used for the measurements and subsequent performance analysis. In practice the phase noise and lock time analysis remain unaffected by this halving of the ideal frequency plan, however, the spurious performance was reduced because of the change in divider ratios following the DAC output.

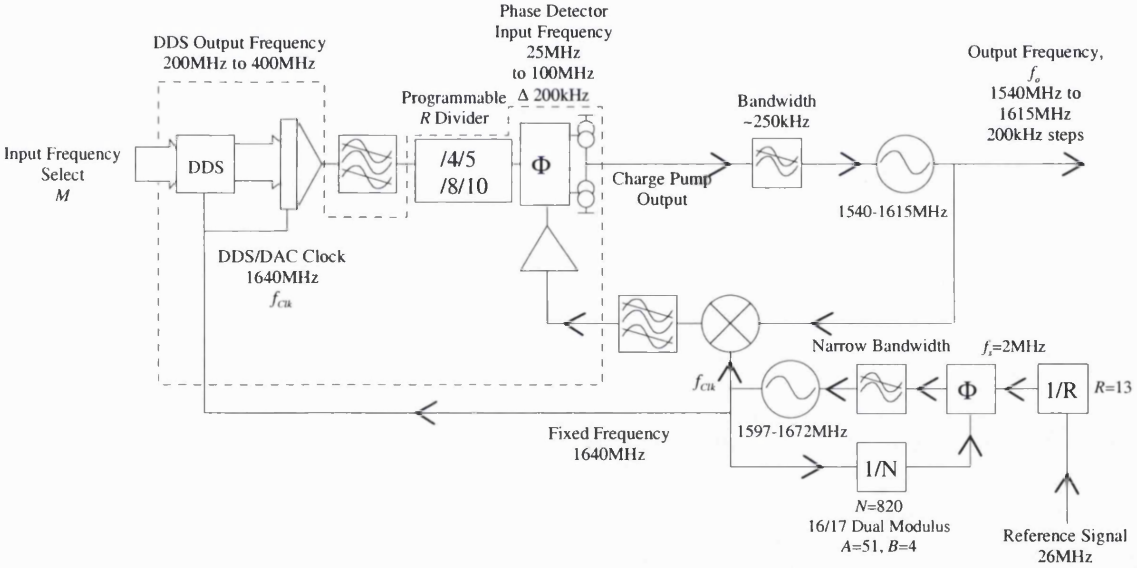


Figure 7.3 DDS output frequency range and *R*-divider settings tested

The spurious analysis prediction for this set-up passes through the restricted DDS output range 200 to 400MHz, whilst the input clock frequency f_{Clk}, remains at 1640MHz, Figure 7.4.

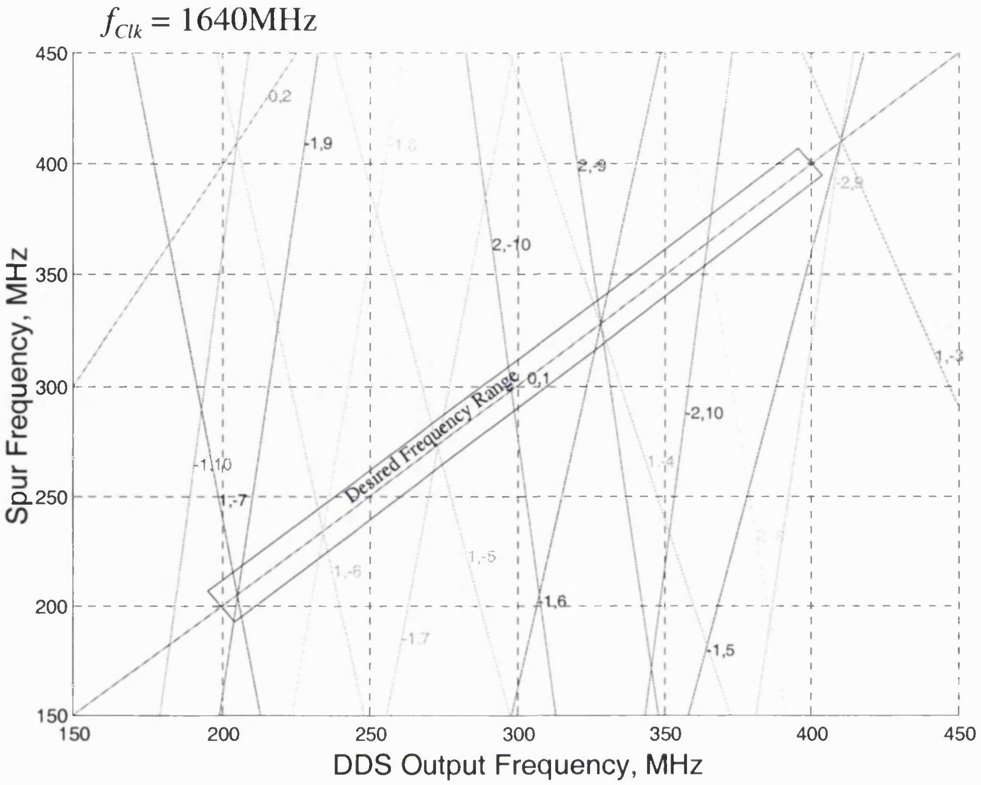
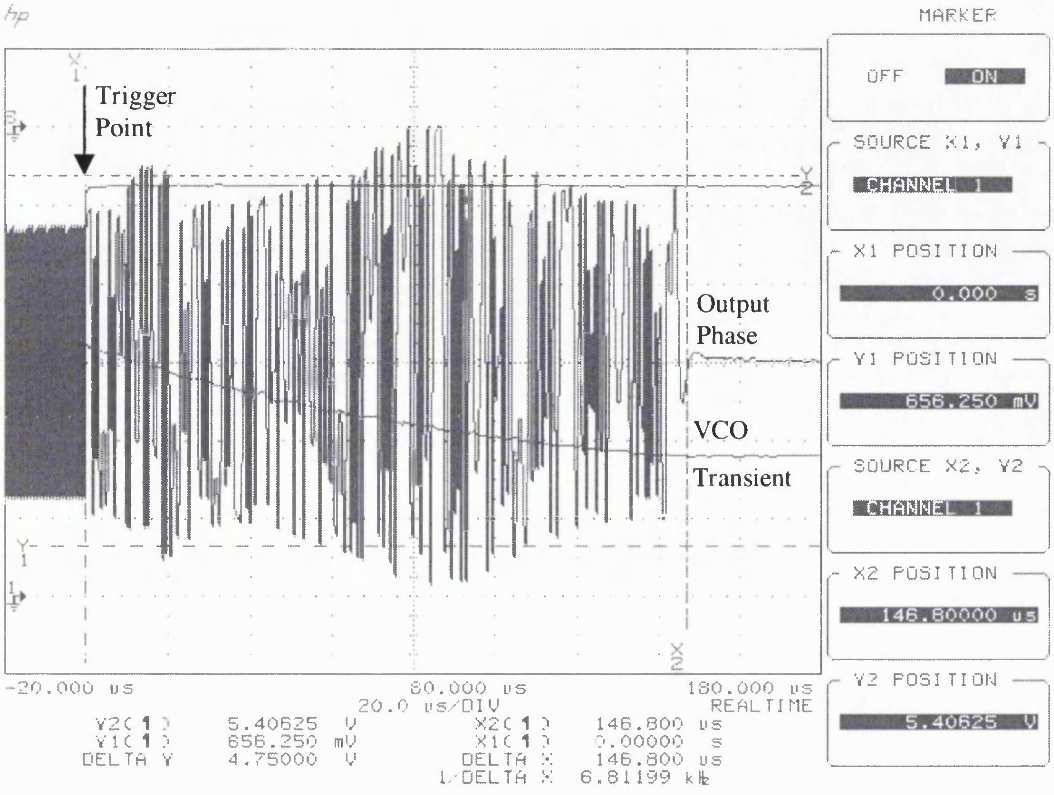


Figure 7.4 Spurious prediction for tested DDS set up in synthesiser

7.3.3 Lock Time Considerations

Generally, the analogue PLL lock time is dependent upon either the loop bandwidth or the sampling frequency at the PFD input. In practice, many factors need considering to accurately gauge the actual lock time of a synthesiser set-up. Initially for this work a set of laboratory measurements comparing loop bandwidth against lock time using an integer-n synthesiser with a 10MHz sampling frequency, indicated a 250kHz-loop bandwidth would be adequate. Subsequently once this hybrid DDS based synthesiser was built, the problem of cycle slipping became apparent.

Contained within Figure 7.5 is the measured lock time at marker "X2" of 146·8μS to within 6° of the target settling phase. The phase transient appears to spend an inordinately long time passing through a frequency transition period, which appears to follow the long exponential decay of the VCO transient suggesting a ~40μS (25kHz) time constant. Clearly, if this pole existed inside the loop bandwidth of the loop, the PLL would be unstable. The explanation taken from [87, 92], is cycle slipping within the digital phase frequency detector because of the direct translation of VCO output phase to PFD input phase.



```
MARKER          OFF  ON
SOURCE X1, Y1   CHANNEL 1
X1 POSITION     0.000 s
Y1 POSITION     656.250 mV
SOURCE X2, Y2   CHANNEL 1
X2 POSITION     146.80000 us
Y2 POSITION     5.40625 V
```

```
-20.000 us                    80.000 us                  180.000 us
                            20.0 us/DIV                    REALTIME
      Y2( 1 )     5.40625  V         X2( 1 )    146.800 us
      Y1( 1 )     656.250 mV         X1( 1 )    0.00000  s
     DELTA Y      4.75000  V        DELTA X     146.800 us
                                  1/DELTA X    6.81199 kHz
```

Figure 7.5 Measured lock time on DDS synthesiser

To overcome this problem the novel architecture in Figure 7.6 was introduced across the accepted fourth-order loop filter configuration. With reference to Figure 7.6, the phase locked loop is used to maintain phase lock, however the DC voltage level is not referenced to zero volts, (ground), instead this reference voltage level is offset by the DAC output voltage.

Particularly significant within Figure 7.6 is that any DAC noise or spurious products are subject to the same band-pass loop response as loop filter thermal noise, Figure 4.12. Capacitor C_3 was left connected to the circuit ground to further limit this high-pass response to any DAC noise. The time constant thus formed, gives a band pass response to the DAC noise and spurious products whilst the PLL loop filter transfer function remains unaffected.

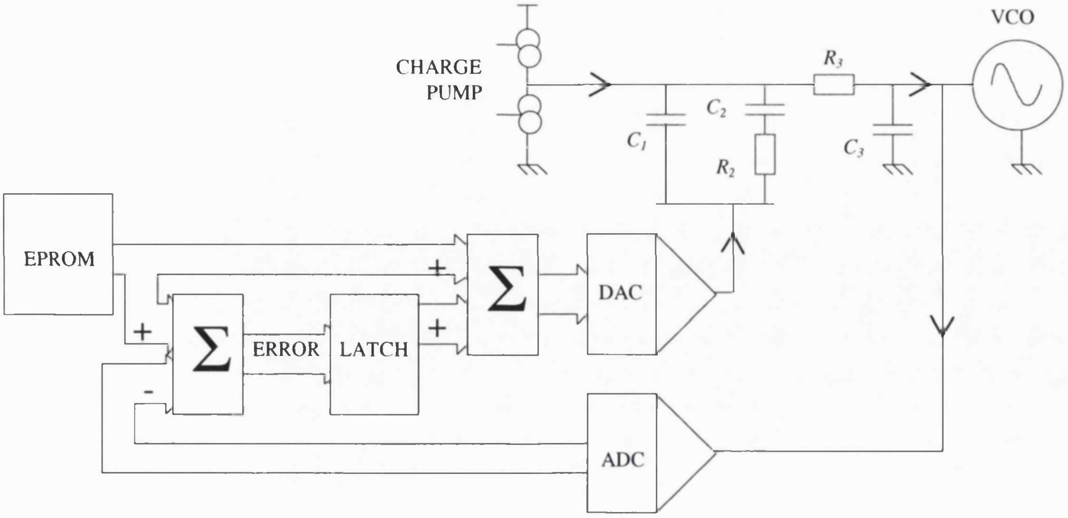


Figure 7.6 DAC compensation for loop filter charge

The concepts used to overcome the large changes introduced in this system, shown in Figure 7.6, are believed to be new and are the subject of a patent application.

The operation of this system requires the voltage pedestal to be changed at the moment the phase locked loop is required to change frequency, thereafter the new voltage level is held constant. This constant voltage offset replaces the charge held in the loop filter capacitors that is normally used to provide a mean DC level offset, relative to 0V.

Changing the voltage pedestal instead of changing the charge in the loop filter capacitors are effectively identical operations and was proven in the laboratory to give no degradation to the phase noise performance of the phase locked loop. When phase locked, a PLL's performance becomes independent of individual charges stored on each loop filter capacitor, but instead is dependant on any incremental changes in charge necessary to overcome the vagaries of all the noise components within the PLL. Without any additional filtering in the loop filter, any noise introduced by the DAC, will undergo high pass filtering, which does not degrade the in-band phase noise. The noise analysis in chapter 4 confirms this DAC noise will be high pass. Additional filtering such as a simple "RC" ladder network, (R_3 and C_3), in cascade with the accepted third-order loop filter configuration, limits this high pass response, (Section 6.2.2).

Figure 7.6, includes a simple circuit used to map the VCO gain characteristics to minimise the error voltages with variations in VCO gain, caused by the effects of temperature and ageing. This simple system assumes the absolute VCO gain profile remains constant and is independent of temperature and ageing.

With this system, the digital PFD need only maintain phase lock and could easily be replaced by an analogue mixer, which has even lower noise. This option was dismissed because of the limited gain of a mixer phase detector and the already limited PLL in-band gain, potentially being unable to adequately suppress the VCO noise.

7.4 Measurement Results

The following measurement results were taken from a prototype of the system described above. For reference, the fourth-order filters used in each PLL were calculated using the following design parameters.

Parameter	Symbol	1640MHz PLL	Hybrid PLL	Units
VCO Gain	K_{VCO}	35	35	MHz/V
Charge Pump Gain	K_{ϕ}	2·5	2	mA/Rad
Loop Bandwidth	*LBW*	3	250	kHz
Phase Margin	-	45	45	°
Additional Attenuation[23]	-	20	20	dB
Sampling Frequency	f_s	2	25 to 100	MHz
Output Frequency	f_o	1640	25 to 100[24]	MHz
N Divider Value	*N*	1640 ÷ 2 =820	1[25]	-

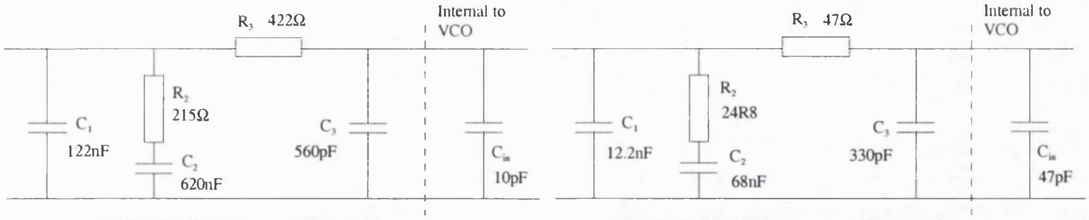
Table 7.3 Loop filter design parameters

Using these parameters, the loop filter component values were calculated using the formulae developed in chapter 6, Figure 7.7.

[23] This attenuation is for calculating the values of R_3 and C_3.

[24] For this loop the IF frequency is used in these calculations.

[25] The DDS Hybrid loop is a tracking PLL with a 1:1 ratio between input and output frequency changes.



1640MHz PLL Loop Filter Values

DDS Hybrid PLL Loop Filter Values

Figure 7.7 Calculated loop filter values for both synthesisers

Figure 7.8 shows the simulated phase noise profile for the 1640MHz f_{Clk} source, showing the principal noise sources of reference noise, PFD noise and VCO noise. Figure 7.9 is the measured phase noise plot for this 1640MHz f_{Clk} source for comparison, which includes the free running VCO used in the design.

Figure 7.10 is a composite phase noise plot using the mathematics presented in section 4, including the DDS output phase noise, (which is a divided down version of the 1640MHz f_{Clk} source), the VCO and PFD noise within this synthesiser. This too can be compared with the measured hybrid synthesiser output phase noise profile, Figure 7.11, suggesting a good correlation between theory and practice. Both of these comparisons between simulation and measured results suggest the mathematics presented in section 4 are correct.

Initially when the synthesiser lock time was measured, very long lock times were observed, Figure 7.5. To fully explain and analyse the loop response during this locking process, the effects of dielectric absorption in the X7R loop filter capacitors and the rapid change in phase seen at the PFD feedback input, would need to be considered. Both of these phenomena result in cycle slipping within the PFD as one of the internal D-type flip-flops is locked on, forcing the charge pump to be held on. During this period, the charge pump current is forced to charge or discharge into the loop filter capacitors and will be slew rate limited. With the DAC compensation circuit, the improvement in lock time is dramatic, Figure 7.12.

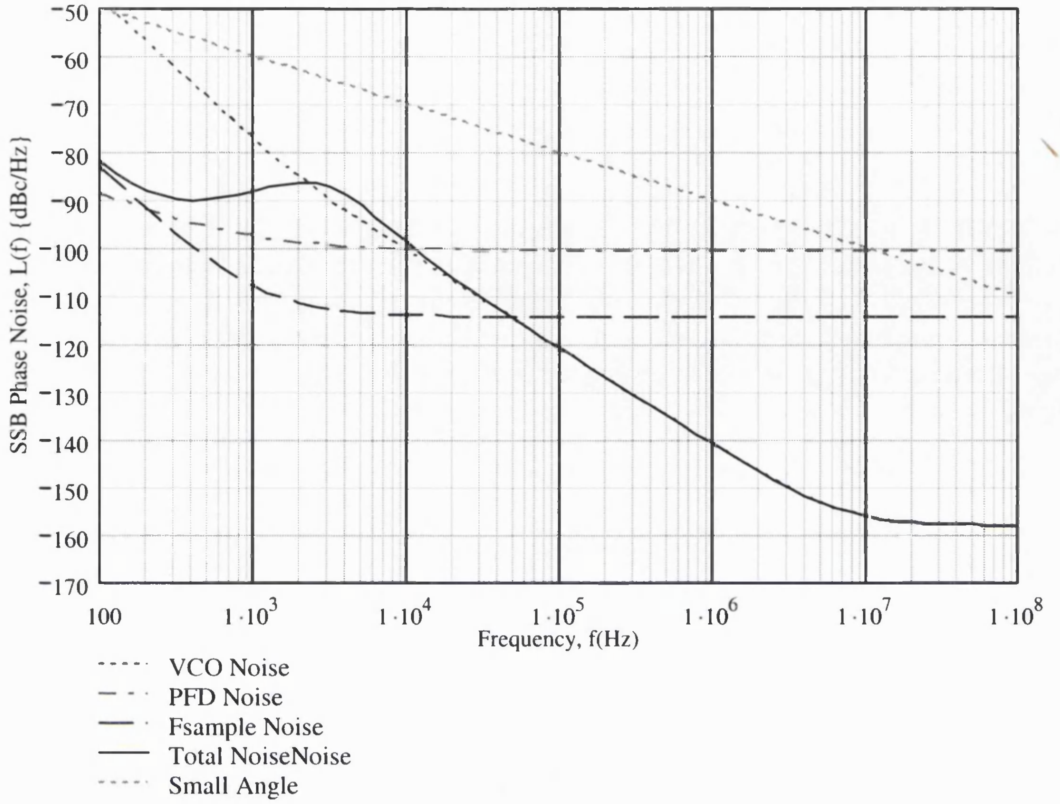


Figure 7.8 Predicted 1640MHz PLL phase noise profile

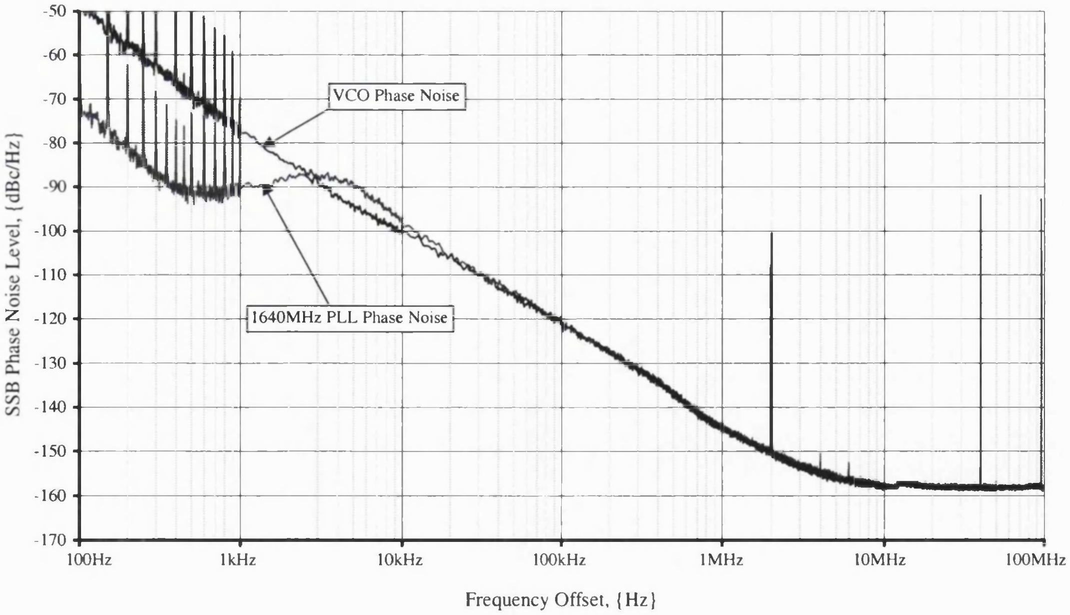


Figure 7.9 Measured 1640MHz PLL and its free running VCO at 1640MHz

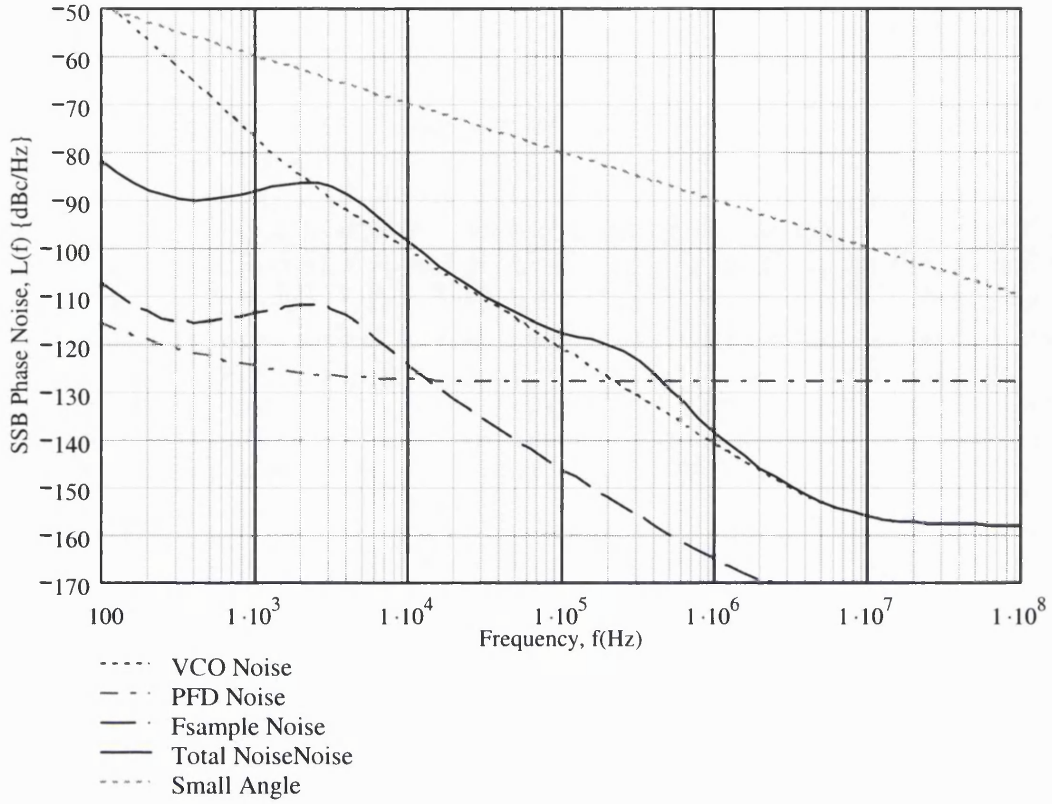


Figure 7.10 Predicted hybrid DDS PLL phase noise profile

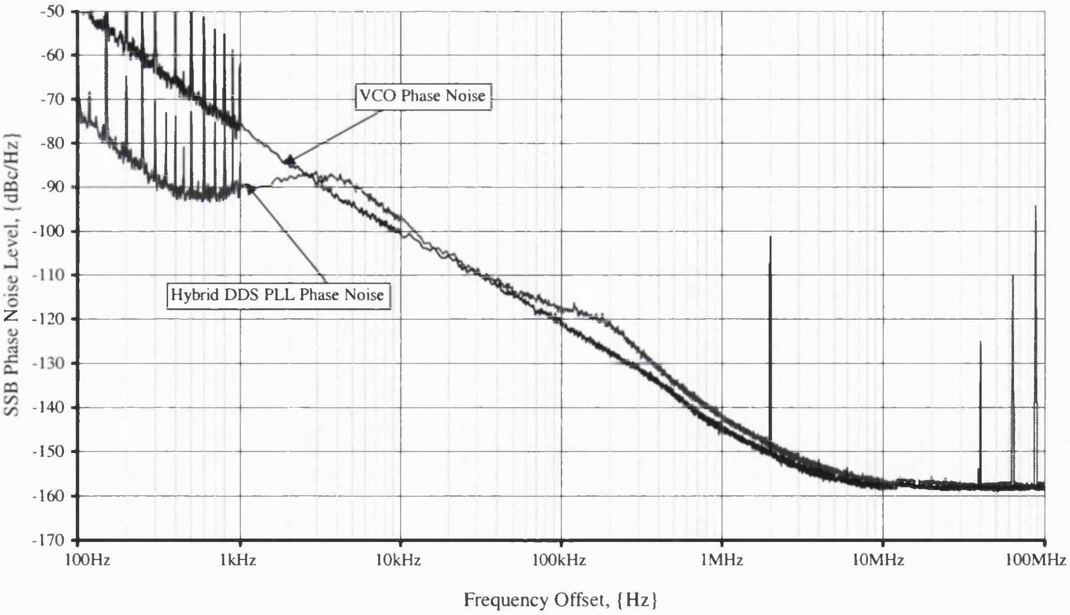
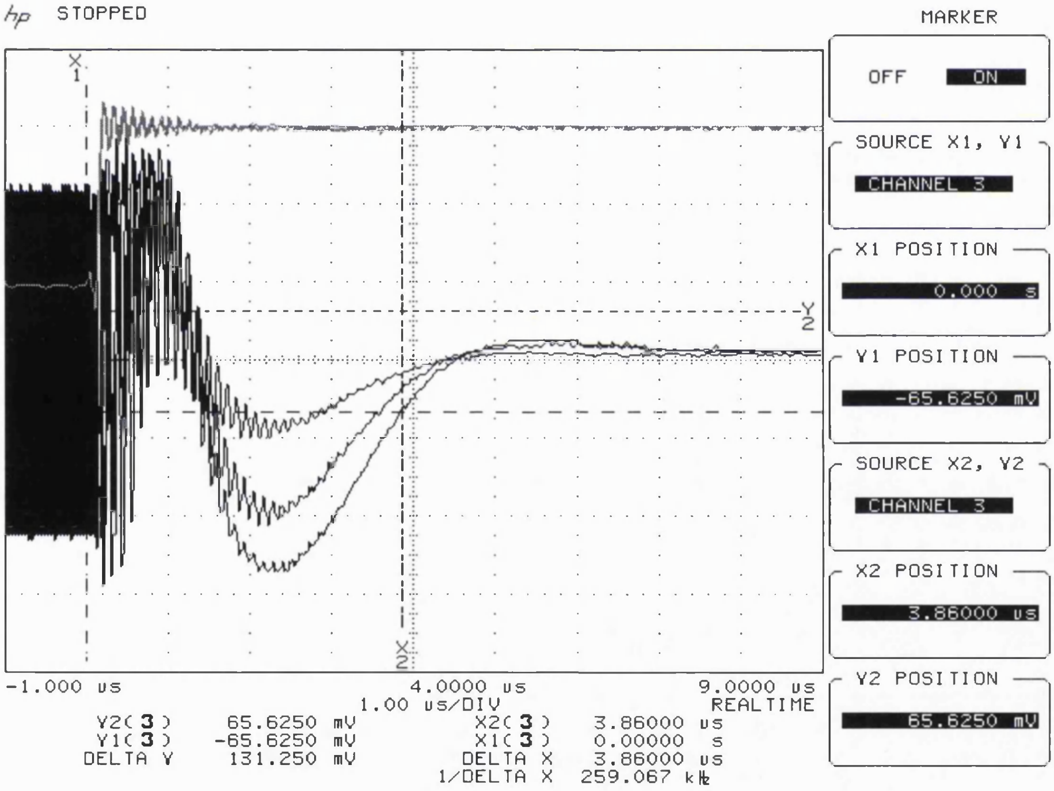


Figure 7.11 Measured hybrid DDS PLL and its free running VCO at 1552·4MHz

In practice, the DAC attempts to remove any change in charge within the loop filter capacitors, leaving the phase locked loop the task of dynamically adjusting the small amounts of charge necessary to maintain phase lock. In principal, this should be straightforward; however there will be small variations in charge within the loop filter capacitors and this will need adjusting during the locking process. These variations will result from inaccuracies between the DAC output and the actual voltage applied to the VCO at the starting frequency and similarly between the DAC output voltage and target V_{tune} value into the VCO. Figure 7.12, shows an overlay of how these voltage differences affect the lock time. During the course of these laboratory measurements, the lock time never exceeded 4·3μS and cycle slipping was not observed indicating that the inherent settling of the synthesiser was restricted by its natural loop frequency, [87].



```
MARKER
OFF  ON
SOURCE X1, Y1
CHANNEL 3
X1 POSITION
0.000 s
Y1 POSITION
-65.6250 mV
SOURCE X2, Y2
CHANNEL 3
X2 POSITION
3.86000 us
Y2 POSITION
65.6250 mV

                         1.00 us/DIV                    REALTIME
   Y2( 3 )    65.6250 mV        X2( 3 )   3.86000 us
   Y1( 3 )   -65.6250 mV        X1( 3 )   0.00000  s
  DELTA Y     131.250 mV       DELTA X    3.86000 us
                             1/DELTA X   259.067 kHz
```

Figure 7.12 Measured lock time with DAC showing distribution due to voltage variations

These measurements were particularly difficult to make, because the DDS settling phase could not be guaranteed. The absolute value held in the accumulator at the instant it was asked to change frequency will vary for each frequency hop, resulting in different settling phases seen at the system output. No attempt was made to reset the DDS accumulator and in any event would have been futile if not applied synchronously with the system reference clock.

For these measurements, it was not possible to momentarily disable the PFD during the DAC settling period, causing the PLL to try to track the ringing inherent on the DAC output. Contained within the first ~1µS of Figure 7.12, there exists a frequency disturbance of the VCO, which is a function of this PLL compensating for the DAC settling transient. Unfortunately, this unnecessarily extends the lock time, but would not be worthwhile compensating for because when the PFD is re-enabled this asynchronous process will also disturb the PLL.

Several phase noise measurements were made using this prototype system specifically looking for predicted spurious products. The results were encouraging, as they were able to predict the spurious hotspots with accuracy. Despite all the precautions, inevitably some unexpected spurious products found their way into the closed loop PLL, Figure 7.11, because of the wide tuning ranges required. Further investigation of some of the mystery spurs uncovered evidence to suggest these were not directly caused by the operation of the system. Instead, some of these spurs were found on the master reference source, introduced through cross talk across the test PCB and from LO to RF leakage back through the analogue mixer used.

7.5 Summary

This chapter describes the design philosophy behind the test synthesiser developed here and draws on an understanding of the component and system limitations discussed in chapters 3, 4 and 5.

The measured phase noise profiles correlated well with their predictions, however unexpected problems with the lock times, Figure 7.5, required some additional design effort to overcome. The concept used to overcome the large changes in loop filter capacitor charges introduced in this system, shown in Figure 7.6, are believed to be new and are the subject of a patent application.

The DAC based solution appeared to work very reliably and vindicated the choice of a 250kHz loop bandwidth to give fast settling.

For this prototype, the SP2002 combined DDS and DAC chip was used which had the limitation of providing a signal up to only a quarter of its clock frequency. In theory this limited the spurious performance of the system, however, measurements suggested this was not the case. The real problem was poor isolation from both the mixer and across the open prototype system PCB, allowing unwanted digital interference to enter the phase locked loop.

As a footnote to this work, Analog Devices have adopted the hybrid DDS based synthesiser concept presented in this chapter and developed their next generation DDS chip, AD9858 [6], for this application. This hybrid DDS based application was displayed on the front cover of the September 2002 edition of the Wireless Systems Design magazine, appendix A6.

The hybrid synthesiser concept demonstrated here exploits the key attributes of two very different forms of frequency synthesis; the analogue PLL and the all digital DDS. DDS was chosen for its frequency agility that is not limited by analogue components, whilst an analogue PLL was added to act as a high Q tracking filter to limit the inherent problem of DDS spurii. To optimise the phase noise and tracking capability of the analogue PLL, the DDS clock source was employed to remove the feedback dividers and hence minimise the PFD noise inherent within the analogue PLL. Thus the two synthesisers became an integrated unit mutually dependent upon one another, for their operation.

Although an all-analogue solution could have achieved the same objective using a judicious combination of loops, the overall architecture would have been considerably more complex. Further, the quantity of circuitry and isolation between individual sections would have resulted in a design requiring considerably more PCB area than would be available within the base station.

8 CONCLUSIONS AND FURTHER WORK

8.1 Conclusions

The underlying objective of this work has been to consider different strategies for the design and analysis of phase locked loop frequency synthesisers. As far as possible a general approach has been adopted so that the principles can be applied equally to other synthesiser configurations though a specific application has been considered by way of example. The techniques developed in this work are finally applied to the design of a high performance synthesiser configuration that meets the exacting demands of a high performance system, without undue complication.

The thesis starts by presenting a brief historical overview of the evolution of different synthesisers because, as is demonstrated, many of the principles used today were conceived in the infancy of frequency synthesis. An important aspect of synthesiser evolution is the digital accumulator and how its deployment has split into two parallel, but distinct paths; i.e. the analogue fractional-n synthesiser and the all-digital DDS system. Both forms of synthesis remain firm favourites in modern synthesiser designs.

To set the scene and present each of the individual building blocks that are available to the synthesiser architect, chapter 3 reviews the workings of many of the common building blocks, including those found in DDS and fractional-n schemes. The purpose of this discussion is to highlight their advantages and disadvantages to fully appreciate their applicability in a given synthesiser system.

Chapter 4 is devoted to modelling the noise performance of each component found in a typical analogue phase locked loop and how their presence contributes to the overall single side band phase noise profile, observed at the synthesiser output. Particular emphasis is given to the sampling action of the phase-frequency detector and how it alters the noise profile, especially as this component cannot benefit from the luxury of an anti-alias filter at its input. Of particular significance is the loop filter thermal noise modelling and the impact this has on the VCO noise profile seen beyond the loop bandwidth. Generally this noise is believed to be solely attributed to the VCO, however it has been shown in this work that the loop filter thermal noise makes a significant contribution to VCO phase noise beyond the loop bandwidth. This degrades the overall level of noise seen at this frequency offset, which is often a problem in radio design because it can reduce the selectivity of the receiver. Experimental validation of this has been performed, with convincing results. This chapter concludes by comparing the measured phase noise profile of a typical PLL with that expected theoretically. Good agreement is achieved, especially around the sampling spurs.

The phase-frequency detector is singled out for separate treatment because of its dominant contribution to in-band phase noise. Very often the level of noise observed inside the loop bandwidth, is mistakenly attributed to the input crystal reference performance, being described by a $20\text{Log}_{10}(N)$ relationship. In some very narrow loop bandwidth, or very poor quality reference signal applications this assumption will be correct, however, this is generally not true for most modern designs. Chapter 5 analyses the phase-frequency detector contribution to the in-band phase noise in which a $10\text{Log}_{10}(f_s)$ relationship is established and experimentally validated. A figure of merit, (FOM), concept is then developed which allows the performance of different phase detectors to be measured under a variety of operating conditions and compared in a normalised fashion. With this technique, the FOM value for a particular PFD can be used to estimate the phase-frequency detector noise contribution in an arbitrary synthesiser.

Since this source of noise is often a limiting factor in both integer and fractional designs then this technique has great value. The effects of phase-frequency detector input waveform shapes and logic switching points have also been considered in this chapter, in order to minimise the PFD logic noise. These are important observations very relevant to a silicon designer seeking to optimise the design a synthesiser integrated circuit.

Frequency agility is a key parameter that is often compromised for the sake of improved spur reduction when deploying commercial analogue synthesiser chips. Very often extra filtering is added to reduce sampling spurs, especially in applications that require the loop bandwidth to approach one-tenth of the sampling frequency. Chapter 6 reviews the loop natural frequency and phase margin degradation when using the popular fourth-order loop filter design methods, and develops an alternative scheme that does not compromise either of these parameters. Analysis has proved that the zero position remains constant in both a third and fourth-order loop filter transfer function, whilst the pole position varies according to the amount of attenuation required and the ratio of loop natural frequency to sampling frequency. This result is confirmed by comparing third-order and fourth-order loop gain responses and noting the co-incidence of the loop bandwidth point at the desired 0dB crossing point for the required phase margin goal in both cases. For increased levels of required spurious attenuation there exists a point at which the added attenuation pole in the loop filter transfer function coincides with the existing pole, to give the maximum available attenuation before impinging on the loop bandwidth. The existence of this point is demonstrated in the mathematics of this chapter. To retain the theme of portability, the mathematics of this loop filter design are conducted in terms of time constants making the result applicable to any form of active or passive loop filter topology. Perhaps the most difficult of translations from time constants to component values is a passive charge pump driven filter configuration, where all component values interact and do not benefit from buffering as might be found in an active filter design. In this work, the translation of time constants to a set of passive loop filter component values is developed around the premise of minimal loop filter thermal noise and hence minimum resistor values in order to achieve the best possible noise performance.

To appropriately conclude this work, a novel DDS-based hybrid synthesiser is developed in chapter 7 and built to provide measurement results that are compared with theoretical expectations. This design draws on the synthesiser building block descriptions in chapter 3, to identify the most promising building blocks with the phase noise analysis of chapters 4 and 5, in order to determine an optimal configuration with reduced noise and DDS spurious product levels. The result is an integrated architecture, now adopted by Analog Devices, using analogue PLL techniques combined with DDS in such a way that they complement and support the operation of each other and achieve hitherto unobtainable performance from a single loop synthesiser. To further optimise the design, the loop filter equations in chapter 6 are used to maximise frequency agility. The measured results presented at the end of this chapter compare very favourably with those predicted using a full summation of all the noise sources found within this DDS based hybrid loop architecture, thus validating the noise analysis.

8.2 Further Work

During the course of these investigations several interesting areas of further work were identified. Notably fractional-n was not deployed in this work in favour of a DDS system, especially as fractional-n chip development is commercially a "hot topic" at this time. The explanation as demonstrated in this work, is that the level of measured in-band phase noise rises compared to that of an equivalent integer-n synthesiser, as a result of the phase frequency detector noise. Although this rise in noise level has been acknowledged in the data sheets for these new fractional-n synthesiser chips, the mechanism that accounts for this disparity remains obscure.

Arguably one brief explanation has been offered here based around the concept of an asynchronously multi-sampled loop, but this needs verifying and would prove a very useful and interesting area of work, because it deviates from the analysis of sigma delta used in linear DAC and ADC converters. Perhaps a better understanding of the mechanisms involved would lead to an alternative sigma delta configuration specific to closed loop applications?

Several authors have touched on the development of higher order loop filters over the years; although some have acknowledged the consequent reduction of loop natural frequency they appear to have failed to address this problem. This surprising fact brings into question the value of higher order loop filter designs as a technique for increasing the filter attenuation since if the initial target loop bandwidth had been reduced, then this would have automatically increased the filter attenuation. This paradox is worthy of further scrutiny, perhaps leading to alternative filter topologies.

The phase-frequency detector analysis of chapter 5, highlights the deficiencies of the ubiquitous tri-state phase frequency architecture, when it is generally acknowledged that an analogue mixer offers a notable improvement in operational performance at phase lock. With higher levels of mixed signal integration possible, a potentially very useful area of work that breaks this deadlock between increased PFD sampling frequency and improved phase noise performance would be the integration of an analogue, (Gilbert cell), type mixer with a digital frequency discriminator. Such a component would re-introduce the virtues of an analogue mixer as a phase detector whilst overcoming its limitation as a pure phase detector, by judiciously using digital technology. One simple answer might be to take advantage of the alternative fast locking part of the PFD architecture proposed by Sharpe in his original article, by virtue of its memory function essential for frequency discrimination and combine this with a Gilbert cell mixer solution. The challenge is to ensure a seamless transition between functional parts of the complete architecture, perhaps using an active loop filter topology as the vehicle for this hand-over.

REFERENCES[26]

CHAPTER 1, INTRODUCTION

1 STREMLER, F. G.: 'Introduction To Communications Systems', (Addison-Wesley Publishing Company, Inc., New York, 3rd Edition, Oct 1992).

2 TSG-RAN Working Group 1 meeting No. 21 TSGR1-01-0495 May 21-25, Busan, Korea Source: Mannesmann Mobilfunk, Vodafone Group plc, SIEMENS AG Title: Requirements for the evaluation of techniques for High Speed Downlink Packet Access HSDPA Agenda point: 10.

3 Slide 1 November 2000 Link Level Simulation Results for HSDPA TSG-RAN Working Group 1 Meeting #17 TSGR1# 17(00) 1326 Stockholm, Sweden, Nov. 21-24, 2000 Agenda Item: AH24, HSDPA Source: Wiscom Technologies Slide 2 November 2000 Simulation Parameters.

4 BRENNAN, P.V., THOMPSON, I.: 'Phase/Frequency Detector Noise Contribution In PLL Frequency Synthesiser', Electronics Letters, Vol. 37, No. 15, pp. 939-940, 19th July 2001.

5 THOMPSON, I., BRENNAN, P.V.: 'Phase Noise Contribution Of The Phase/Frequency Detector In A Digital PLL Frequency Synthesiser', *IEE Proc. Circuits, Devices & Systems*, Vol. 150, No. 1, pp. 1-5, February 2003.

6 ANALOG DEVICES: 1GSPS Direct Digital Synthesizer, Preliminary Data Sheet, Rev. PrA, 1/22/2002.

CHAPTER 2, THE HISTORY OF FREQUENCY SYNTHESIS

7 McCREA, H.A.: 'Automatic Control Of Frequency And Load', *General Electric Review*, Vol. 32, pp. 309-313, June, 1929.

[26] All patents referenced have been given both their filing and publication dates. The filing date suggests the true point in time at which the invention was conceived as compared with the time it took for the invention to be approved. Further, all known U.K., European or US version of the patent are referenced in their order of filing for the same reason.

8 ALEXANDERSON, E.F.W.: 'Wireless Signalling System', US 1 465 961, Filed: April 19, 1916, Patented: Aug. 28, 1923.

9 De BELLESCIZE, H.: 'La Reception Synchrone', *Onde Electr.*, Vol. 11, pp. 230-240, June 1932.

10 TRAVIS, C.F.: 'Automatic Frequency Control', *Proc. I.R.E.*, Vol. 23, pp. 1125-1141, October 1935.

11 ROUND, H.J.: 'Radio Signalling System', Filed in England, March 18, 1921 and U.S. 1,642,173, Filed: Sept. 13th 1927.

12 WENDT, K.R., FREDENDALL, G.L.: 'Automatic Frequency And Phase Control Of Synchronization In Television Receivers', *Proc. I.R.E.*, Vol. 31, pp.7-15, January 1943.

13 FINDEN, H.J.: 'The Frequency Synthesizer', *Journal I.E.E.E. 90*, Pt. 3, pp. 165-180, 1943.

14 RICHMAN, D.: 'Color-Carrier Reference Phase Synchronization Accuracy in NTSC Color Television', *Proc. I.R.E.*, Vol.42, pp. 106-133, January, 1954.

15 MARTIN, Jr., T.L.: 'Electronic Circuits', (Prentice-Hall, Inc., New York), pp. 405, 465, 1955.

16 SEPE, R.B., JOHNSTON, R.I.: 'Frequency Multiplier And Frequency Waveform Generator', U.S. Patent No. 3,551,826, December 29, 1970, Filed: May 16, 1968.

17 GILLETTE, G.C.: 'The Digiphase Synthesizer', *Frequency Technology*, pp. 25-29, August 1969.

18 GILLETTE, G.C.: 'Frequency Synthesizer System', U.S. Patent No. U.S. 3,582,810, Filed: May 5, 1969, Patented: June 1, 1971.

19 GORSKI-POPIEL, J.: 'Frequency Synthesis: Techniques And Applications', (IEEE Press, 1975).

20 TIERNEY, J., RADER C.M., GOLD B.: 'A Digital Frequency Synthesizer', *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-19, No. 1, pp. 48-57, March 1971.

21 RITCHIE, G.R.: 'Higher Order Interpolation Analog To Digital Converters', Ph.D. Dissertation, University of Pennsylvania, 1977.

22 KING, N.J.R.: ' Frequency Synthesizers', GB 2 026 268 A, Filed: 22nd July 1978, Patented: 30th Jan. 1980, US 4,204,174, Filed: Nov. 9 1978, Patented: May 20, 1980.

23 WELLS, J.N.: ' Frequency Synthesisers', GB 2 140 232 A, Filed: 17th May 1983, Patented: 21st Nov 1984, EP 0 125 790, Filed: 11th April 1984, Patented: 21st June 1989, US 4,609,881, Filed: May 3, 1984, Patented: Sept. 2, 1986.

24 JACKSON, T.: 'Fractional-N Frequency', GB 2 172 760 A, Filed: 21st Feb. 1986, Patented: 24th Sept. 1986, EP 211 921, Filed: 21st February 1986, Patented: 28th August 1986, US 4,758,802, Filed: Feb. 21, 1986, Patented: Jul. 19, 1988.

25 MILLER, B.M.: 'Multiple-Modulator Fractional-N Divider', US 5,038,117, Filed: Sept. 7, 1990, Patented: Aug. 6, 1991.

26 GASKELL, P.S., BREAKENRIDGE, E., KING, N.J.R., BALL, M.J.: 'Fractional-N Frequency Synthesizer Using Delta-Sigma Modulator In Divider Control', EP 0 429 217 A3, Filed: 7th Nov. 1990, Patented: 29th May 1991, US 5,079,521, Filed: Nov. 21, 1990, Patented: Jan. 7, 1992.

27 BRENNAN, P.V., THOMPSON, I., BORJAK, A.: 'Frequency Synthesiser', WO 01/61858 A1, Filed: 14th February 2000.

28 WALKINGTON, R.: 'New Approaches In Noise-Shaping Fractional-N Synthesis', Ph.D. Thesis, University College London, December 2001.

29 GLOVER, M., HERMES, B., KUP, B., SLOWGROVE D.: 'A Fully Integrated Bitstream Analogue-To-Digital Converter For Digital Audio', International Conference on Analogue to Digital and Digital to Analogue Conversion, (Conf. Publ. No. 343), IEE, Swansea, pp. 165-169, 17-19 Sept. 1991.

30 De JAGER, F.: 'Delta Modulation – A Method Of PCM Transmission Using The One Unit Code', *Philips Research Report*, Vol. 7, pp.442-466, 1952.

31 CUTLER, C.C., GILLETTE. N.J.: 'Transmission Systems Employing Quantization', US 2,927,962, Filed April 26, 1954, Patented March 8, 1960.

32 INOSE, H., YASUDA, Y., MURIKAMI, J.: 'A Telemetering System By Code Modulation-$\Delta\Sigma$ Modulation', *IRE Trans. Space Electron. Telem.*, pp. 204-209, 8th Sept. 1962.

33 CANDY, J.C.: 'A Use Of Limit Cycle Oscillations To Obtain Robust Analog-to-Digital Converters', *IEEE Trans. Comm.*, pp. 298-305, 22nd March 1974.

34 TELEWSKI, F., CRAFT, K., DRUCKER, E., MARTINS, J.: 'Delay Lines Give RF Generator Spectral Purity, Programmability' *Electronics*, pp. 133 – 142, August 28, 1980.

35 McNAMEE, M.D., PLATT, D.L.: 'A Modular Family Of High-Performance Signal Generators', *Hewlett-Packard Journal*, Vol. 40, No. 10, pp. 14 – 20, October 1989.

36 CLIFF, R.A.: 'Digitally Controlled Frequency Synthesizer', Filed Aug. 26, 1966, Patented Aug. 26, 1969.

37 WEBB, J.A.: 'Digital Signal Generator Synthesizer', US 3,654,450, Filed Apr. 3, 1970, Patented Apr. 4, 1972.

38 HUTCHINSON Jr., B.H.: 'Frequency Synthesis And Applications', IEEE Press, 1975.

39 SUNDERLAND, D., STRAUCH, R., WHARFIELD, S., PETERSON, H., COLE, C.: 'CMOS/SOS Frequency Synthesizer LSI Circuit For Spread Spectrum Communications', *IEEE Trans. Solid State Circuits*, SC-19, No. 4, 1984.

40 VOLDER, J.E.: 'The CORDIC Trigonometric Computing Technique', *IRE Trans. On Electronic Computers*, pp. 330-334, Sept. 1960.

41 WHEATLEY, C.E., PHILIPS, D.E.: 'Spurious Suppression In DDS', *Proceedings of the 35th Annual Frequency Control Symposium*, May 1981.

42 WILSON, M.P., TOZER, T.C.: 'Spurious Reduction Techniques For Direct Digital Synthesis', IEE Colloquium on "Direct Digital Frequency Synthesis", Digest No., 1991/172, Tuesday, 19th Nov, 1991.

43 SAUL, P.H., BARBER, W., TAYLOR, D.G., WARD, T.: 'Single Chip 500MHz Function Generator', *IEE Proceedings-G*, Vol. 138, No. 2, pp. 239-243, April 1991.

44 MITEL SEMICONDUCTORS: SP2002A/B Data Sheet, DS2303 Issue 4·3, October 1999.

45 HOSKING, R.H.: 'A Unique Synthesizer And Its Advantages', *IEEE Intercon Technical Papers*, Paper 34/1, 1973.

46 GUTIERREZ-AITKEN, A., MATSUI, J., KANESHIRO, E., OYAMO, B., SAWADAI, D., OKI, A., STREIT, D.: 'Ultra High Speed Direct Digital Synthesizer Using InP DHBT Technology', IEEE GaAs Digest, pp. 265-268, 2001.

CHAPTER 3, DEFINING FREQUENCY SYNTHESISERS

47 DRENTEA, C.: 'Designing Frequency Synthesizers', RF Technology Expo-88, pp. 37-58.

48 FAN, Y.: 'Model, Analyze, And Simulate ΣΔ Fractional-N Frequency Synthesizers, Part 1 & 2', Microwaves & RF, pp. 183-194, December 2000, pp. 150-154, January 2001.

49 SCHERER, D., CHAN, B.S., IVES, F.H., CRILLY Jr., W.J., MATHIESEN, D.W.: 'Low-Noise RF Signal Generator Design', *Hewlett-Packard Journal*, Vol. 32, No. 2, pp. 12 – 22, February 1981.

50 BURNS, R.E.: 'A High-Performance Signal Generator For RF Communications Testing', *Hewlett-Packard Journal*, Vol. 36, No. 12, pp. 4 – 6, December 1985.

51 OWEN, D.: 'A New Approach To Low Noise Signal Generators', *Electronic Engineering*, pp. 55 – 60, March 1991.

52 KEESE, O. K.: 'Dual PLL IC Achieves Fastest Lock Time With Minimal Reference Spurs', *RF Design*, Vol. 18, No. 8, pp. 30, 32, 34, 37-38, August 1995.

53 MURPHY, E.J., MORGAN, S.O.: 'The Dielectric Properties Of Insulating Materials', *Bell Systems Tech. Journal*, Vol. 16, pp. 493-512, October 1937.

54 DOW, P.C.: 'An Analysis Of Certain Errors In Electronic Differential Analyzers II – Capacitor Dielectric Absorption', *IRE Trans. On Electronic Computers*, Vol. EC-7, pp. 17-22, March 1958.

55 PEASE, B.: 'Understanding Capacitor Soakage To Optimize Analog Systems', *EDN*, October 13th, 1982.

56 BERSTEIN, I.L.: 'On Fluctuations In The Neighbourhood Of Periodic Motion Of An Auto-Oscillating System', *Doklad. Akad. Nauk.*, Vol. 20, pp. 11, January 1938.

57 EDSON, W.A.: 'Noise In Oscillators', *Proc. IRE*, Vol. 48, pp. 1454-1466, August 1960.

58 MULLEN, J.A.: 'Background Noise In Non-Linear Oscillators', *Proc. IRE*, Vol. 48, pp. 1467-1473, August 1960.

59 LEESON, D.B.: 'A Simple Model Of Feedback Oscillator Noise Spectrum', *Proceedings of the IEEE*, Vol. 42, pp. 329-330, February 1966.

60 SCHERER, D.: 'Design Principals And Test Methods For Low Phase Noise RF And Microwave Sources', RF & Microwave Measurement Symposium and Exhibition, April 1979.

61 ROHDE, U.L., WHITAKER, J., BUCHER, T.N.N.: 'Communications Receivers' (McGraw Hill, New York, pp. 380, Second Edition, January 1997).

62 RICE, S.O.: 'Mathematical Analysis Of Random Noise', in N. Wax and N. Wax, eds. *Selected papers on Noise and Stochastic Processes*, Dover, New York, pp. 133-294, 1954.

63 GILMORE, R., KORNFELD. R.: 'Hybrid PLL/DDS Frequency Synthesizers', Qualcomm, Inc., Application Note AN2334-4, March 1992.

64 WEAVER, L., KERR, R.J.: 'High Resolution Phase To Sine Amplitude Conversion', U.S. 4905177, Qualcomm, Filed January 19, 1988, Patented February 27, 1990.

65 GOLDBERG, BAR-GIORA.: 'Digital Techniques In Frequency Synthesis', McGraw-Hill, New York, 1996.

66 ANALOG DEVICES: 'Mixed Signal Processing Design Seminar', ISBN-0-916550-08-7, 1991.

67 MORELAND, C.: 'Data Converters In Advanced Communications Systems', Oxford Continuing Education Electronics & Communications, 2000.

68 ROHDE, U.L.: 'Microwave And Wireless Synthesizers Theory And Design', (John Wiley, New York, 1997).

CHAPTER 4, MODELLING PLL PHASE NOISE

69 SKLANSKY, J.: 'Network Compensation Of Error-Sampled Feedback Control Systems', Technical Report T-7/B, Ph.D. thesis, Columbia University Engineering Centre, New York, April 1955.

70 AMIT, N.: 'Optimal Control Of Multirate Digital Control Systems', Ph.D. thesis, Stanford University, July 1980.

71 RAGAZZINI, J.R., FRANKLIN, G.F.: 'Sampled Data Control Systems' (McGraw-Hill, New York, 1958).

72 TOU, J.T.: 'Digital And Sampled-Data Control Systems', (McGraw-Hill, New York, 1959).

73 KUO, B.C.: 'Digital Control Systems' (Oxford University Press, New York 2nd Edition, 1992).

74 FRANKLIN, G. F., POWELL, D., WORKMAN, M.L.: 'Digital Control Of Dynamic Systems' (Addison Wesley Longman, Inc., Menlo Park, CA, 3rd Edition, 1997).

75 SALZER, J. M.: 'Signal Flow Reductions In Sampled-Data Systems', *Wescon Convention Record*, Institute of Radio Engineers, pt. 4, pp. 166-170, 1957.

76 IEEE Std 1139-1999, IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology-Random Instabilities, IEEE Standards Co-ordinating Committee 27 on Time and Frequency, ISBN 0-7381-1753-6, (Print), 26th June 1999.

77 CHADWICK, P.E.: 'Dynamic Range Intermodulation And Phase Noise', *Radio Communication*, pp. 223-228, March 1984.

78 MANASSEWITSCH, V.: 'Frequency Synthesizers Theory And Design', (Wiley Inter-Science, New York, 3rd Edition, 1987).

79 UNDERHILL, M.J., SCOTT, R.I.H.: 'The Effect Of The Sampling Action Of Phase Comparators On Frequency Synthesizer Performance', *Proceedings of the 33rd Annual Symposium on Frequency Control*, pp. 449-457, 1979.

80 WHITBECK, R.F., DIDALEUSKY D.G.J.: 'Multi-rate Digital Control Systems With Simulation Applications', Vol. I, II, III, U.S. Air Force Technical Report, AFWAL-TR-80-3031, Systems Technology, Inc., Hawthorne, California, 1980.

CHAPTER 5, PFD NOISE LIMITATIONS

81 SHARPE, C.A.: 'A 3-State Phase Detector Can Improve Your Next PLL Design', *EDN*, pp. 55-59, Sept. 1976.

82 GARDNER, F.M.: 'Charge-Pump Phase-Lock Loops', *IEEE Trans. Communications*, Vol. COM-28, pp. 1849-1858, Nov. 1980.

83 UTSI, V.: 'Evaluation Of Phase Noise In Emitter Coupled Logic', Fifth International Conference on Radio Receivers and Associated Systems, pp. 227-228, July 1990.

84 WILSON, M.P., TOZER, T. C.: 'Synthesisers For Low Data-Rate Satellite Receivers', *Proceedings IEE Conference on 'Frequency Control & Synthesis'*, Leicester, pp. 73-78, May 1989.

85 KROUPA, V.F.: 'Frequency Synthesis: Theory, Design And Applications', (Griffin, London, 1973).

86 WHITE, D.R.: 'The Noise Bandwidth Of Sampled Data Systems', *IEEE Trans. On Instrumentation and Measurement*, Vol. 38, No. 6, pp. 1036-1043, Dec. 1989.

87 BRENNAN, P.V.: 'Phase-Locked Loops, Principals And Practice', (McGraw-Hill, 1996).

CHAPTER 6, LOOP FILTER DESIGN

88 BANERJEE, D.: 'PLL Performance, Simulation And Design', National Semiconductor, Second Edition, http://www.nsc.com/, 2001.

89 KRUG, F., WILWERT, J.: 'Calculation And Measurement Of Lock Time In A Phase-Locked Loop Frequency Synthesizer', Microwave Journal, pp. 22-38, April 2002.

90 STEPINSKI, R.: 'Design High-Order PLLs', Microwaves & RF, pp. 69-86, July 2001.

91 ROHDE, U.L., NEWKIRK, D.P.: 'RF/Microwave Circuit Design For Wireless Applications', (Wiley Inter-Science, New York, 2000).

92 CRAWFORD, J.A.: 'Frequency Synthesizer Design Handbook', (Artech House, Inc., Norwood, MA, 1994).

93 GARDNER, F.M.: 'Phaselock Techniques', (Wiley, New York, 2nd Edition, 1980).

94 HIGGINS, R.J.: 'Digital Signal Processing In VLSI', (Prentice-Hall, Inc., New Jersey, 1990).

CHAPTER 7, HYBRID PLL/DDS BASED SYNTHESISER

95 BENN, H.P.: 'Spurious Frequency Generation In Direct Digital Synthesizers', *IEE Colloquium on 'Direct Digital Frequency Synthesis'*, Ref. No. 1991/172, 19th Nov 1991.

APPENDICES

A1 DEFINITION OF SSB PHASE NOISE

This section provides a mathematical derivation of the accepted expression:

$$\mathcal{L}(f)=\frac{1}{2}S_{\phi}(f) \qquad \text{Eq A1.1}$$

Taking the general expression for a practical signal source containing both a phase noise component, ϕ and an amplitude modulation component, $A(t)$ superimposed on a carrier ω_c of amplitude A:

$$V(t)=(A+AM(t))\cdot\cos[\omega_c t+\phi(t)] \qquad \text{Eq A1.2}$$

The amplitude modulation component can be ignored because within a PLL the self-limiting action of successive stages reduces this component to a very low level. Therefore, rewriting this equation in terms of narrow band phase modulation:

$$V(t)=A(t)\cdot\cos\left[\omega_c t+\frac{f_d}{f_m}\sin(\omega_m t)\right] \qquad \text{Eq A1.3}$$

where $\phi(t)=\frac{f_d}{f_m}\sin(\omega_m t)$

where f_d is the frequency deviation and f_m is the modulating frequency.

Using trigonometric identities the cosine term can be expanded:

$$V(t)=A(t)\cdot\left\{\cos(\omega_c t)\times\cos\left[\frac{f_d}{f_m}\sin(\omega_m t)\right]-\sin(\omega_c t)\times\sin\left[\frac{f_d}{f_m}\sin(\omega_m t)\right]\right\} \qquad \text{Eq A1.4}$$

For narrow band phase modulation, (ϕ < 0.1 Radian), the approximations $\sin(x)\approx x$ and $\cos(x)\approx 1$ can be made, leading to the simplifications:

$$\cos\left[\frac{f_d}{f_m}\sin(\omega_m t)\right]\approx 1 \qquad \text{Eq A1.5}$$

and

$$\sin\left[\frac{f_d}{f_m}\sin(\omega_m t)\right]\approx\frac{f_d}{f_m}\sin(\omega_m t) \qquad \text{Eq A1.6}$$

Accounting for these approximations in the equation gives:

$$V(t)=A(t)\cdot\left\{\cos(\omega_c t)-\sin(\omega_c t)\times\frac{f_d}{f_m}\sin(\omega_m t)\right\} \qquad \text{Eq A1.7}$$

using the standard trigonometric for sin(*x*) × sin(*y*) results in

$$V(t)=A(t)\cdot\left\{\cos(\omega_c t)-\frac{f_d}{2f_m}\sin(\omega_c t+\omega_m t)+\frac{f_d}{2f_m}\sin(\omega_c t-\omega_m t)\right\} \qquad \text{Eq A1.8}$$

which demonstrates how the phase energy is distributed equally on either side of the carrier at power levels equivalent to half the modulation index, $\beta=\frac{f_d}{f_m}$. Therefore the sidebands in the RF spectrum, relative to the carrier, are 6 dB below the amplitude of the modulating signal whilst the sidebands in the RF spectrum, relative to the carrier, are 3 dB below the power of the modulation.

Given a signal with $\mathcal{L}(f_m)$ single sideband phase noise profile, the equivalent phase jitter, ϕ_{jitter}, in radians can be calculated between the two frequency offsets f_1 and f_2 in this SSB phase noise profile:

$$\phi_{jitter}=\sqrt{2\int_{f_1}^{f_2}\mathcal{L}(f_m)df_m} \qquad \left\{rad^2\right\} \qquad \text{Eq A1.9}$$

From which the equivalent timing jitter, T_{jitter}, can be calculated for the output frequency, f_o:

$$T_{jitter}=\frac{\phi_{jitter}}{2\pi f_o} \qquad \{\sec\} \qquad \text{Eq A1.10}$$

A2 PROPAGATION OF ERRORS

If some quantity *z* is calculated as a function of two other quantities *x* and *y*, then for some nominal or reference values x_0 and y_0,

$$z_0 = f(x_0, y_0) \qquad \text{Eq A2.1}$$

and for small deviations from this normal value,

$$z_0 + dz = f(x_0 + dx, y_0 + dy) \qquad \text{Eq A2.2}$$

Is a first-order approximation, good for *dx* and *dy* much smaller than x_0 and y_0, respectively,

$$z_0 + dz = f(x_0, y_0) + \frac{\delta f}{\delta x}\, dx\Big|_{\substack{x=x_0\\ y=y_0}} + \frac{\delta f}{\delta x}\, dy\Big|_{\substack{x=x_0\\ y=y_0}} \qquad \text{Eq A2.3}$$

$$dz = \frac{\delta f}{\delta x}\, dx\Big|_{\substack{x=x_0\\ y=y_0}} + \frac{\delta f}{\delta x}\, dy\Big|_{\substack{x=x_0\\ y=y_0}} \qquad \text{Eq A2.4}$$

If *dx* and *dy* are taken to represent errors in *x* and *y*, respectively, then *dz* is the error in *z* that results from these errors.

If *x* and *y* are random variables, as for example measurements having random errors, then *z* is also a random variable, and to a first order approximation,

$$\mu_z = f(\mu_x, \mu_y) \qquad \text{Eq A2.5}$$

$$\sigma_z^2 = \left[\frac{\delta f}{\delta x}\right]^2 \sigma_x^2 + \left[\frac{\delta f}{\delta y}\right]^2 \sigma_y^2 + 2\rho_{xy} \frac{\delta f}{\delta x}\frac{\delta f}{\delta y} \sigma_x \sigma_y \qquad \text{Eq A2.6}$$

for σ_x and σ_y sufficiently small that $f(x,y)$ is nearly linear in the vicinity of μ_x and μ_y. The square brackets signify that the derivatives within the brackets are to be evaluated at μ_x and μ_y. If *x* and *y* are uncorrelated, (usually true if they are independent), then the correlation coefficient ρ_{xy} is zero and the second derivative reduces to

$$\sigma_z^2 = \left[\frac{\delta f}{\delta x}\right]^2 \sigma_x^2 + \left[\frac{\delta f}{\delta y}\right]^2 \sigma_y^2 \qquad \text{Eq A2.7}$$

The extension to functions of more than two variables is straightforward.

A3 MIXER INTERMODULATION ANALYSIS

An ideal mixer is a device driven by two inputs; the local oscillator and RF signal inputs and produces a single intermediate frequency output, by effectively multiplying these two signals together. Using trigonometry the ideal output consists of the sum and difference terms:

$$\cos\omega_{RF}t \times \cos\omega_{LO}t = \frac{1}{2}\cos(\omega_{RF} - \omega_{LO})t + \frac{1}{2}\cos(\omega_{RF} + \omega_{LO})t \qquad \text{Eq A3.1}$$

Depending upon the application either one of these two outputs are selected, whilst the other is attenuated by IF filtering. In practice one input, usually the local oscillator input, is driven by a high signal level to obtain the best conversion gain of the RF signal to IF signal as well as minimising the mixer noise contribution. This high drive signal level alternately switches on each pair of gates in a bridge network, which serves to sample the RF signal first in one direction then in the other.

Mathematically this switching function can be modelled using the signum function:

$$S(t) = Sgn(\cos\omega_{LO}t) \qquad \text{Eq A3.2}$$

whose Fourier expansion is:

$$\begin{aligned} S(t) &= \frac{2A}{\pi}\left\{\cos\omega_{LO}t - \frac{1}{3}\cos 3\omega_{LO}t + \frac{1}{5}\cos 5\omega_{LO}t - .....\right\} \\ &= \sum_{n=1,3,5,.....}^{\infty} \frac{2A}{n\pi}\cos n \cos\omega_{LO}t \end{aligned} \qquad \text{Eq A3.3}$$

This switching function is then multiplied with the incoming RF signal, usually a sinusoid, to give the ideal mixer output:

$$\text{Mixer Output} = \frac{1}{2}\frac{2A}{n\pi}\left(\sum_{n=1,3,5,...}^{\infty} \cos(\omega_{RF} - n\omega_{LO}t) + \sum_{n=1,3,5,...}^{\infty}\cos(\omega_{RF} - n\omega_{LO}t)\right) \qquad \text{Eq A3.4}$$

From this expansion, it is clear the output contains a number of products whose amplitude is weighted by the Fourier decay of the signum function centred on the sum and difference products. However, the large signal drive of the internal active device, usually diodes, drives these devices into their non-linear region of operation and distorts the output waveform, causing harmonic components to be generated. To account for this non-linearity the transfer characteristic of the diodes used within the mixer can be modelled using the exponential function:

$$I_{Out} = I_{Sat}\left(e^{rV_{In}} - 1\right) \qquad \text{Eq A3.5}$$

Where, I_{sat} = diode saturation current, $r = q/kt$ = (26mV)$^{-1}$ and V_{In} is the voltage across the diode.

Expanding this transfer function gives the ascending power series:

$$I_{Out} = I_{Sat}\left[rV_{In} + \frac{1}{2!}(rV_{In})^2 + \frac{1}{3!}(rV_{In})^3 + \ldots + \frac{1}{n!}(rV_{In})^n\right] \qquad \text{Eq A3.6}$$

Given the input voltage across these diodes comes from the combined local oscillator and RF inputs:

$$V_{In} = V_{RF}\sin\omega_{RF}t + V_{LO}\sin\omega_{LO}t \qquad \text{Eq A3.7}$$

then the output current from these diodes driving the output transformers becomes:

$$\begin{aligned}
I_{Out} = I_{Sat}[&r(V_{RF}\sin\omega_{RF}t + V_{LO}\sin\omega_{LO}t)\\
&+\frac{1}{2!}(r(V_{RF}\sin\omega_{RF}t + V_{LO}\sin\omega_{LO}t))^2\\
&+\frac{1}{3!}(r(V_{RF}\sin\omega_{RF}t + V_{LO}\sin\omega_{LO}t))^3 \qquad \text{Eq A3.8}\\
&+\ldots\\
&+\frac{1}{n!}(r(V_{RF}\sin\omega_{RF}t + V_{LO}\sin\omega_{LO}t))^n + \ldots]
\end{aligned}$$

Using trigonometric identities, the above equation can be partitioned into DC, sum, difference and harmonic terms:

$$\begin{aligned} I_{Out} = A_{00} &+ A_{10}\sin\omega_{RF}t + A_{01}\sin\omega_{LO}t + A_{11}\sin(\omega_{RF} \pm \omega_{LO})t \\ &+ A_{20}\sin 2\omega_{RF}t + A_{12}\sin(\omega_{RF} \pm 2\omega_{LO})t \\ &+ A_{21}\sin(2\omega_{RF} \pm \omega_{LO})t + A_{22}\sin(2\omega_{RF} \pm 2\omega_{LO})t \\ &+ \ldots + A_{mn}\sin(m\omega_{RF} \pm n\omega_{LO})t + \ldots \end{aligned} \qquad \text{Eq A3.9}$$

Hence, the mixer output contains many different terms, which require filtering to prevent their existence from reducing the spurious free dynamic range of the remainder of the system.

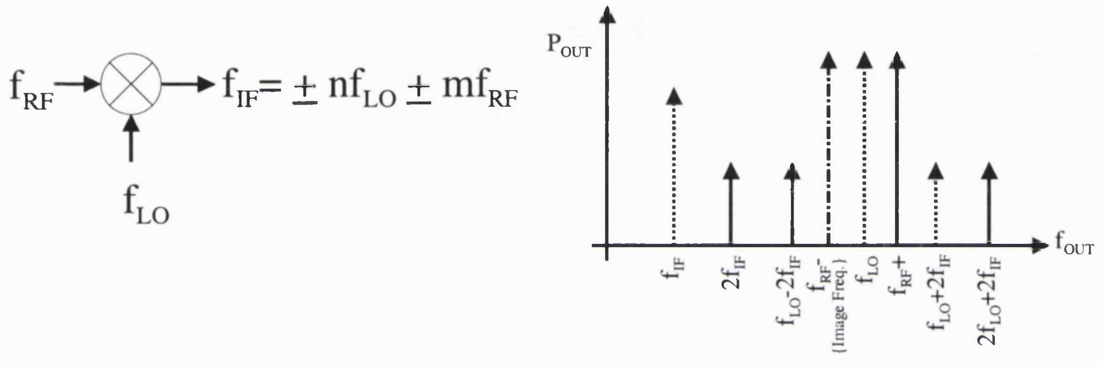


Figure A3.1 Non-linear mixer generated spurious products

From this analysis, the location of spurious products can be determined using the general $n \times m$ product term in Eq A3.9

$$\text{Mixer Spurious} = \pm n \times f_{LO} \pm m \times f_{RF} \qquad \text{Eq A3.10}$$

A4 ANALYSIS OF MODULATED CHARGE PUMP CURRENT, PULSE WIDTH MODULATION

Consider the alternative expression for Eq 4.25, describing the unmodulated charge pump current pulse:

$$i(t)=i_{cp}\left[\frac{\tau}{T}+\sum_{n=1}^{\infty}\frac{2}{n\pi}\sin(n\omega_s\tau/2)\cos(n\omega_s t)\right] \qquad \text{Eq A4.1}$$

When this pulse width is modulated by the error correction signal; $\tau(1+\beta\, sin(\omega_m t))$, where the modulation factor, $\beta=\Delta\omega/2\omega_m$ the modulated pulse train is given by

$$i(t)=i_{cp}\left[\frac{\tau}{T}(1+\beta\sin(\omega_m t))+\sum_{n=1}^{\infty}\frac{2}{n\pi}\sin(n\omega_s\tau/2(1+\beta\sin(\omega_m t)))\cos(n\omega_s t)\right] \qquad \text{Eq A4.2}$$

Expanding this expression and taking the Bessel functions, yields:

$$\begin{aligned}
i(\mathrm{t})=i_{cp}\Bigg[&\frac{\tau}{T_s}+\frac{\beta\tau}{T_s}\sin(\omega_m t)+\frac{2}{\pi}\sin\left(\omega_s\frac{\tau}{2}\right)J_0\left\{\beta\omega_s\frac{\tau}{2}\right\}\cos(\omega_s t)\\
&+\frac{2}{\pi}\cos\left(\omega_s\frac{\tau}{2}\right)J_1\left\{\beta\omega_s\frac{\tau}{2}\right\}[\sin((\omega_s+\omega_m)t)-\sin((\omega_s-\omega_m)t)] \qquad \text{Eq A4.3}\\
&+\frac{2}{\pi}\sin\left(\omega_s\frac{\tau}{2}\right)J_2\left\{\beta\omega_s\frac{\tau}{2}\right\}[\cos((\omega_s+2\omega_m)t)+\cos((\omega_s-2\omega_m)t)]\\
&+\ldots\ldots
\end{aligned}$$

At this point this modulated current pulse is filtered and converted into a voltage value to be applied to the tune voltage input of the VCO. However, the value of the modulation index, β, represents the modulation value taken from the phase noise profile of the sampling signal, which should be very low, Figure 4.2. This value of β is taken as part of the argument for the Bessel values, which makes all the higher order Bessel values very small:

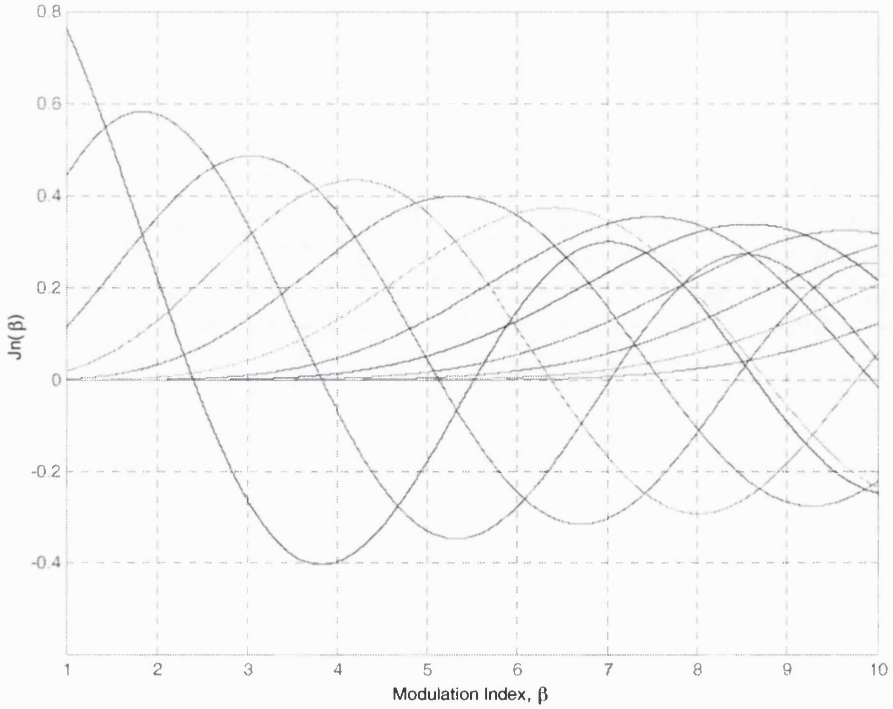


Figure A4.1 Plot of Bessel Function of the First Kind, $J_n(\beta)$

Leaving only the terms:

$$i(t)=i_{cp}\left[\frac{\tau}{T_s}+\frac{\beta\tau}{T_s}\sin(\omega_m t)+\frac{2}{\pi}\sin\left(\omega_s\frac{\tau}{2}\right)J_0\left\{\beta\omega_s\frac{\tau}{2}\right\}\cos(\omega_s t)\right] \qquad \text{Eq A4.4}$$

Of these terms the sine term, $\frac{\beta\tau}{T_s}\sin(\omega_m t)$, becomes very small as τ becomes small, whilst the second sine term approximates its argument and the Bessel value $J_0\left\{\beta\omega_s\frac{\tau}{2}\right\} \cong 1$, again as the τ value in the sine argument becomes very small. This leaves

1) the $\frac{\tau}{T_s}$ term, which represents the small but finite voltage offset introduced by this leakage current. It is this small DC term, which is subject to the loop filter trans-impedance before being added to the voltage already present on the loop filter to control the VCO output.

$$\omega_{out}(t)=A\cos\left(\left[\left(i_{cp}\left[\frac{\tau}{T_s}+\frac{\omega_s\tau}{\pi}\cos(\omega_s t)\right]\cdot|f(t)|+V_{tune}\right)2\pi K_{vco}\right]t+\omega t\right) \qquad \text{Eq A4.5}$$

2) the $\frac{\omega_S\tau}{\pi}\cos(\omega_S t)$ term representing the sampling frequency that modulates the VCO.

For a desirable sampling signal whose Single Side Band phase noise profile is very good as shown in Figure A4.2, the equivalent modulation index orders greater than unity can be ignored:

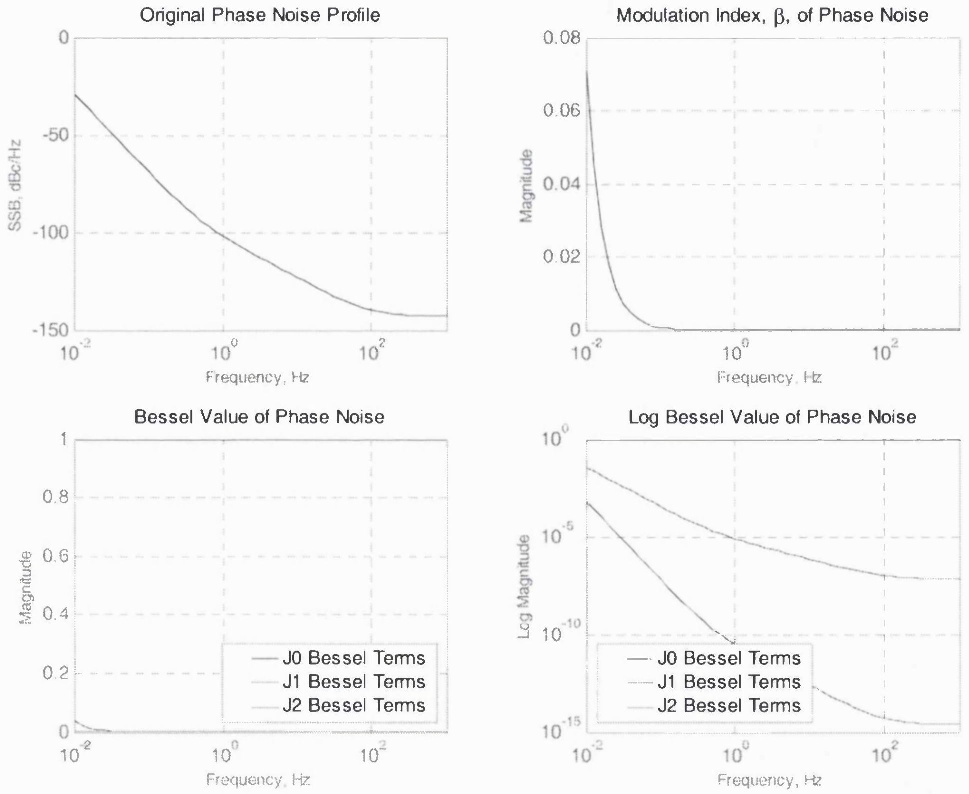


Figure A4.2 Translation of Typical Sampling Signal Phase Noise Profile into Low Level Modulation Components that can be Ignored.

This analysis further illustrates how the very low phase noise profile found on the sampling signal can be ignored when compiling a composite phase noise profile including sampling spurs.

A5 CANDIDATES PUBLICATIONS

Phase/frequency detector phase noise contribution in PLL frequency synthesiser

P.V. Brennan and I. Thompson

A straightforward mathematical analysis of the noise contribution of a phase/frequency detector in a PLL frequency synthesiser is presented. A figure of merit is derived which allows comparison of different phase/frequency detectors and an accurate estimate of their performance in given configurations. Experimental results show excellent agreement with theory.

Introduction: Since the circuit for the tri-state phase/frequency detector (PFD) was first published [1], followed by the charge pump addition [2], the combination of these two elements has become ubiquitous in digital frequency synthesiser applications. Several mathematical methods describing the noise contribution of the PFD within a synthesiser have been offered [3, 4], along with laboratory observations and theory confirming that PFD jitter noise is dominant. These semi-empirical methods involve a normalisation technique and an associated figure of merit (FOM) for a given synthesiser chip which allows the performance of different devices to be compared. The approach is analogous to the intercept point concept used to predict intermodulation product levels in nonlinear devices. The FOM can similarly be applied to a given synthesiser configuration to predict the PFD noise contribution contained within its output phase noise profile.

This Letter describes an analytic basis for such an approach, along with convincing experimental verification.

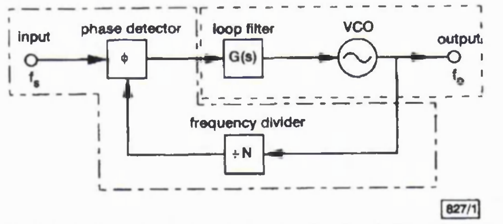


Fig. 1 *Noise sources contributing to in-band and out-of-band noise in PLL frequency synthesiser*

–·–·– in-band noise sources
– – – – out-of-band noise sources

PFD noise model: The various sources of phase noise in a synthesiser can be categorised as either in-band or out-of-band, as shown in Fig. 1. The loop filter and VCO are subject to high-pass filtering by loop action and are thus significant sources of out-of-band phase noise, whereas the frequency divider, phase detector and input signal are subject to low-pass filtering by loop action and are thus significant sources of in-band phase noise [5]. The PFD has been found, experimentally, to be the dominant source of such in-band phase noise.

Additive noise, predominantly thermal, within the PFD gives rise to timing jitter on both the rising and falling edges of the output pulses. This may be considered equivalent to a certain input timing jitter of Δt seconds RMS on the reference input of a hypothetical, noise-free PFD and, in turn, may be related to an equivalent phase jitter at the PFD input, which for a phase detector operating frequency of f_s, is given by

$$\Delta\phi_{in} = 2\pi f_s \Delta t \quad \text{rad RMS} \qquad (1)$$

In practice, Δt is very small (of the order of picoseconds). The PFD, being a sampling device with an output pulse train of low duty cycle in a locked loop, is a good approximation to an impulse sampler thus having an equivalent noise bandwidth of half the sampling frequency and virtually uniform spectral density of translated components over this frequency range [6]. Hence the equivalent input phase noise power spectral density is

$$S\phi_{in}(f) = \frac{(\Delta\phi_{in})^2}{f_s/2} = 8\pi^2 f_s \Delta t^2 \quad \text{rad}^2/\text{Hz} \qquad (2)$$

This indicates a 10 dB/decade increase in PFD phase noise with the phase detector operating frequency, f_s. The resulting output phase noise power spectral density is subject to a gain within the loop bandwidth equivalent to the divider ratio, $N = f_o/f_s$, where f_o is the output frequency of the synthesiser, giving the following expression:

$$S\phi_{out}(f) = 8\pi^2 f_s \Delta t^2 N^2 \equiv \frac{8\pi^2 \Delta t^2 f_o^2}{f_s} \quad \text{rad}^2/\text{Hz} \qquad (3)$$

This now shows a 10 dB/decade *decrease* in output phase noise with the phase detector operating frequency, f_s. Assuming that the overall output phase noise is smaller than 0.1 rad, then a narrowband phase modulation approximation [5] may be employed to arrive at the output SSB phase noise.

$$\text{SSB noise power density relative to carrier} \\ = S\phi_{out}(f)/4 = \frac{2\pi^2 \Delta t^2 f_o^2}{f_s} \quad /\text{Hz} \qquad (4)$$

and this may be expressed more conveniently in dB terms:

$$\text{SSB noise density (dBc/Hz)} \\ = FOM\ (\text{dBc/Hz}^2) + 20\log_{10} f_o - 10\log_{10} f_s \qquad (5)$$

where *FOM* is the figure of merit of the phase/frequency detector, which is constant for a given device. This result indicates that the phase noise increases at a rate of 20 dB/decade with the output frequency and decreases at a rate of 10 dB/decade with the PFD operating frequency. This latter –10 dB/decade dependency on f_s is highly significant but not widely appreciated. It explains why it is clearly advantageous, in synthesiser design, to operate phase/frequency detectors at the highest possible frequency in order to reduce the in-band phase noise plateau. The expression in eqn. 5 serves two purposes: it allows calculation of the PFD-borne in-band phase noise for a given phase detector under any combination of reference and output frequencies; it also allows a normalised figure of merit to be associated with any phase detector in order to compare the noise performance between devices. A typical figure of merit is –220 dBc/Hz^2 though the authors are aware of variations over the range –222 dBc/Hz^2 to –204 dBc/Hz^2.

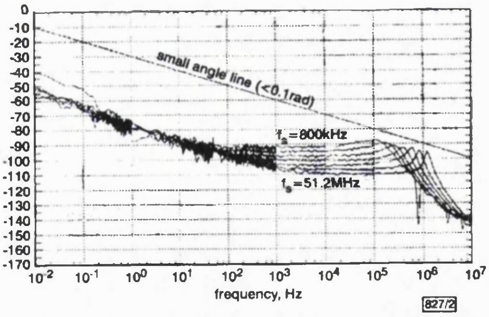


Fig. 2 *Measured phase noise profiles for phase/frequency detector operating frequencies of 0.8, 1.6, 3.2, 6.4, 12.8, 25.6 and 51.2 MHz*

Experimental results: A set of measured phase noise profiles is shown in Fig. 2 for a synthesiser operating at a constant output frequency of 1.64 GHz but with PFD operating frequencies varying in octave intervals from 800 kHz to 51.2 MHz. There is a substantially flat phase noise plateau between 1 and 100 kHz offset from the carrier. This region represents in-band phase noise arising from the PFD and lies between the low frequency flicker noise and the loop noise bandwidth. For every octave increase in the

PFD operating frequency it is evident that there is a 3 dB improvement in phase noise. At f_s = 800 kHz the phase noise (over 1 to 100 kHz offset) averages –91 dBc/Hz^2 whereas at f_s = 51.2 MHz, an increase of six octaves, the phase noise averages –109 dBc/Hz^2 – an improvement of 18 dB – exactly in accordance with eqn. 5.

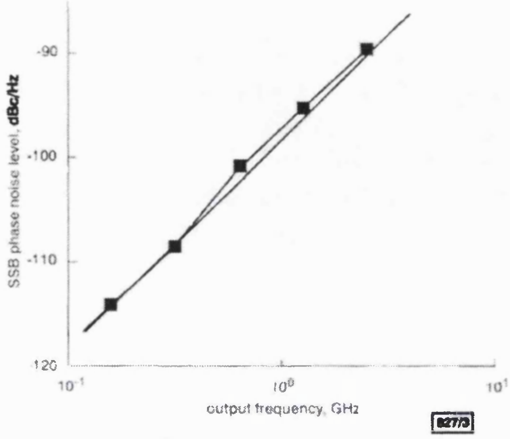


Fig. 3 *Measured in-band phase noise variation with output frequency*

Fig. 3 is an example of the measured in-band phase noise as a function of output frequency, f_o, the results clearly showing the expected 20 dB/decade increase with f_o. Further results have been obtained, for two separate synthesisers operating at a constant output frequency of 1.64 GHz, showing the measured in-band phase noise as a function of PFD operating frequency. These results, shown in Fig. 4, indicate a very clear 10 dB/decade decrease in phase noise with increasing PFD operating frequency, as predicted. The 10 dB/decade trend is observed to within 1 dB over a four octave frequency range, beyond which a noise floor is reached due to other noise sources in the synthesiser and/or test equipment. By applying eqn. 5, the figures of merit of the PFDs in these two synthesisers are found to be –212 dBc/Hz^2 and –220 dBc/Hz^2, respectively.

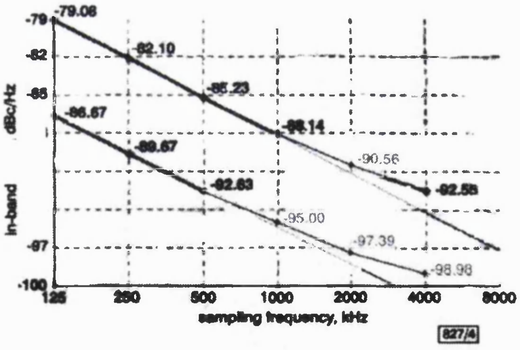


Fig. 4 *Measured in-band phase noise against PFD operating frequency for two different synthesiser designs*

Conclusions: An analytic technique has been presented which predicts the in-band phase noise contribution of a phase/frequency detector in a range of synthesiser configurations and also enables the figure of merit of a particular phase/frequency detector to be established. This figure of merit allows direct comparison of the noise performance of different phase/frequency detector devices, which is crucial in determining the ultimate in-band phase noise limit of a given synthesiser design. Experimental results have been presented showing excellent agreement with theory and, in particular, clearly demonstrating the anticipated 10 dB/decade improvement of synthesiser phase noise with increasing PFD operating frequency.

© IEE 2001 29 May 2001
Electronics Letters Online No: 20010665
DOI: 10.1049/el:20010665

P.V. Brennan (*Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom*)

I. Thompson (*Nokia Telecommunications Ltd, Stanhope Road, Yorktown Industrial Estate, Camberley, Surrey GU15 3BW, United Kingdom*)

References

1 SHARPE, C.A.: 'A 3-state phase detector can improve your next PLL design', *EDN*, 1976, **21**, (17), pp. 55–59

2 GARDNER, F.M.: 'Charge-pump phase-lock loops', *IEEE Trans. Commun.*, 1980, **COM-28**, pp. 1849–1858

3 UTSI, V.: 'Evaluation of phase noise in emitter coupled logic'. 5th Int. Conf. Radio Receivers and Associated Systems, July 1990, pp. 227–228

4 WILSON, M.P., and TOZER, T.C.: 'Synthesisers for low data-rate satellite receivers'. Proc. IEE Conf. Frequency Control and Synthesis, Leicester, United Kingdom, May 1989, pp. 73–78

5 BRENNAN, P.V.: 'Phase-locked loops, principles and practice' (McGraw-Hill, 1996)

6 STREMLER, F.G.: 'Introduction to communication systems' (Addison-Wesley, 1990), 3rd edn.

Phase noise contribution of the phase/frequency detector in a digital PLL frequency synthesiser

I. Thompson and P.V. Brennan

Abstract: A theoretical basis for the figure of merit method used to quantify the phase noise plateau of a PLL frequency synthesiser is described. Analyses are developed both to calculate the in-band phase noise of a given synthesiser architecture and to predict the figure of merit from the phase/frequency detector parameters. A range of experimental results is provided to validate the theory.

1 Introduction

Since the circuit for the tristate phase frequency detector was first published [1], followed by the charge pump addition [2], the combination of these two elements has become ubiquitous in digital synthesiser applications. Several mathematical operators describing the noise contribution of the phase/frequency detector, (PFD), within a synthesiser have been offered [3, 4] along with laboratory observations and theory, confirming that PFD jitter noise is dominant. These mathematical operators offer a 1 Hz normalisation technique, giving a figure of merit for a given synthesiser chip, allowing the performance of different devices to be compared. The figure of merit (FOM), is a very similar approach to the intercept point concept adopted for extrapolating an amplifier's intermodulation products for a given operation scenario. The FOM can similarly be applied to a given synthesiser configuration to determine the PFD noise contribution contained within its output phase noise profile.

The objective of this paper is to consider the thermal noise calculations used to arrive at the FOM for a given logic family and to illustrate how this value is translated to the expected in-band phase noise contribution of the PFD in the synthesiser output. In particular, analysis is presented to determine how the PFD noise is related to device parameters. The translation of the FOM to in-band phase noise includes the effects of the sampling frequency and subsequent multiplication of the synthesiser.

2 PFD noise model

2.1 PFD noise processes

The phase noise profile observed at the output of a digital phase locked loop is a composite of all noise sources within the loop, each individually modified by the phase transfer characteristic from its point of injection to the loop output. With reference to Fig. 1, any noise injected before the loop filter will predominantly contribute to the in-band phase noise; conversely any noise source injected beyond the loop filter will predominantly contribute to the out-of-band part of the phase noise profile. Of all the noise sources within a PLL there are three prime sources which dominate the overall phase noise profile, as might be measured on a phase noise test set. These sources are the multiplied reference signal noise, the digital PFD noise and the free running VCO phase noise. Additional noise introduced by a less-than-noise-optimal loop bandwidth can easily be predicted by the roll-off characteristics of the filter used within the PLL; however, the baseline level of this noise is again dependent upon the digital phase/frequency detector noise. The following analysis concerns the timing jitter introduced at the phase/frequency detector input, which largely dominates the in-band component of the phase noise profile, and how this noise manifests itself within the output phase noise profile of a synthesiser.

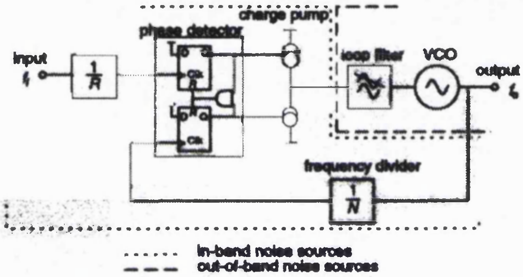


Fig. 1 *Phase/frequency detector under consideration, within synthesiser*

2.2 PFD timing jitter analysis

Additive noise, predominantly thermal, within the PFD gives rise to timing jitter in the edges of the output pulses, Fig. 2, [5]. This jitter, which is uncorrelated and present on both rising and falling edges, may be considered equivalent to a certain input timing jitter of Δt seconds RMS on the reference input of an ideal, noise free PFD. Depending upon the PFD design, one or other of the edges of each incoming waveform independently triggers each input device giving the phase comparison, with the time difference between these edges being translated to the required current pulse duration at the charge pump output, necessary to

IEE Proceedings online no. 20030221

doi:10.1049/ip-cds:20030221

Paper first received 12th June 2001 and in revised form 23rd May 2002

I. Thompson is with Nokia Telecommunications Ltd, Stanhope Road, Yorktown Industrial Estate, Camberley, Surrey GU15 3BW, UK

P.V. Brennan is with the Department of Electronic & Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

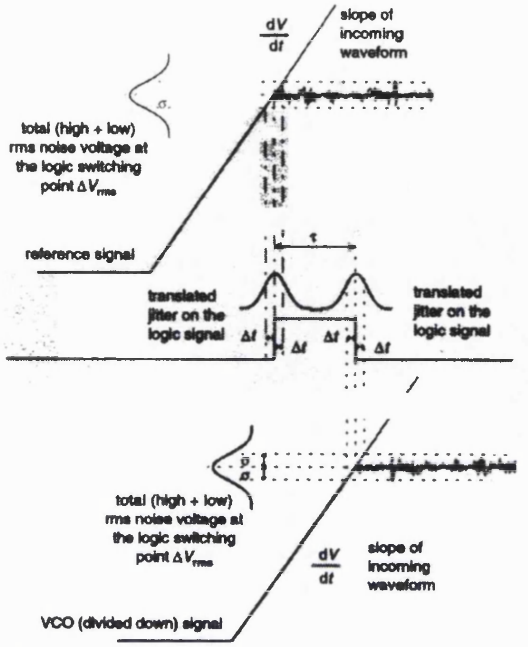

dV/dt
slope of incoming waveform
σ
total (high + low) rms noise voltage at the logic switching point ΔV_{rms}
reference signal
τ
translated jitter on the logic signal
Δt
Δt
Δt
Δt
translated jitter on the logic signal
σ
σ
total (high + low) rms noise voltage at the logic switching point ΔV_{rms}
dV/dt
slope of incoming waveform
VCO (divided down) signal

Fig. 2 *Logic transfer characteristic at D-type input of the PFD*

maintain phase lock. By superposition, each occurrence of this PFD timing jitter is similarly translated to an equivalent phase jitter, which modulates the width of the pulse operating the charge pump, injecting this noise into the remainder of the synthesiser. For a phase detector with an operating frequency of f_s the equivalent phase jitter is:

$$\Delta\phi_{in} = 2\pi f_s \Delta t \quad [\text{rad(rms)}] \qquad (1)$$

In practice, Δt is very small, of the order of picoseconds, while the thermal noise possesses a bandwidth much greater than the PFD sampling frequencies. The PFD, being an edge-triggered sampling device with an output pulse train of very low duty cycle pulses in a locked loop, is a good approximation to an impulse sampler possessing an equivalent noise bandwidth of half the sampling frequency and virtually uniform spectral density of translated components over this frequency range [6]. Hence, the equivalent input double-sided spectral density of phase fluctuations is:

$$S\phi_{in}(f) = \frac{(\Delta\phi_{in})^2}{f_s/2} = 8\pi^2 f_s \Delta t^2 \quad \left[\frac{\text{rad}^2}{\text{Hz}}\right] \qquad (2)$$

This indicates a 10 dB/decade increase in PFD phase noise contribution with the phase detector operating frequency f_s. For a typical phase locked synthesiser with only a divider feedback path, the resulting output double-sided spectral density of phase fluctuations is subject to a gain equivalent to the divider ratio gain $N = f_o/f_s$, where f_o is the output frequency of the synthesiser. The value of N can be either integer or fractional. Therefore, it follows that,

$$S\phi_{out}(f) = 8\pi^2 f_s \Delta t^2 N^2$$
$$\equiv \frac{8\pi^2 \Delta t^2 f_o^2}{f_s} \quad \left[\frac{\text{rad}^2}{\text{Hz}}\right] \qquad (3)$$

This now indicates an overall 10 dB/decade decrease in output phase noise with the phase detector operating frequency f_s. Assuming the overall output phase noise is smaller than 0.1 rad, from [7] the output SSB phase noise power $\mathcal{L}(f)$ is equal to one-half of the double-sideband power spectral density of phase fluctuations:

$$\mathcal{L}(f) = \frac{1}{2} S\phi_{out}(f) = 10\log_{10}\left(\frac{4\pi^2 \Delta t^2 f_o^2}{f_s}\right) \quad [\text{dBc/Hz}] \qquad (4)$$

This can more conveniently be expressed in terms of dB as:

$$\mathcal{L}(f) = \text{FOM}\{\text{dBc/Hz}^2\} + 20\log_{10} f_o - 10\log_{10} f_s \quad [\text{dBc/Hz}] \qquad (5)$$

Returning to the divider substitution in (3) gives arguably the more familiar version of this formula as:

$$\mathcal{L}(f) = \text{FOM}\{\text{dBc/Hz}^2\} + 20\log_{10} N + 10\log_{10} f_s \quad [\text{dBc/Hz}] \qquad (6)$$

Where FOM is the figure of merit of the particular phase/frequency detector used and is constant. The implication of (5) is not widely appreciated, but is highly significant as it indicates that, for a given synthesiser output frequency f_o, the phase noise decreases by 10 dB/decade with the PFD operating frequency (and also increases by 20 dB/decade with output frequency). It is therefore, advantageous to operate phase/frequency detectors at the highest possible operating frequency available within the constraints of the given application, to reduce the in-band phase noise plateau. Either expressions (5), or its equivalent (6), serve two purposes; they allow calculation of in-band phase noise attributed to the PFD operation for a given synthesiser frequency configuration and allow a FOM to be derived offering a simple means of comparing different synthesiser chips. Typical FOM values range from −213 dBc/Hz^2 for some 5-year-old devices to −222 dBc/Hz^2 for the latest device.

2.3 Additive thermal noise

An estimate of the figure of merit as a function of the design parameters of a given PFD is of significant value and may be obtained as follows. It will be assumed that the input is a rectangular pulse train of peak amplitude $\hat{V}$ (the logic level swing), with a degree of rounding of the pulse edges as modelled by a first-order low-pass function of 3 dB cut-off frequency f_c. In addition, the PFD input will be band-limited according to a similar first-order low-pass function. Thus, the pulse edges arriving at the PFD may be described by

$$v(s) = \frac{\hat{V}}{s}\frac{\omega_c^2}{(s+\omega_c)^2}$$
$$\equiv \hat{V}\left(\frac{1}{s} - \frac{1}{s+\omega_c} - \frac{\omega_c}{(s+\omega_c)^2}\right) \qquad (7)$$

and so

$$v(t) = \hat{V}[1 - e^{-\omega_c t} - \omega_c t e^{-\omega_c t}] \qquad (8)$$

where the slope of these edges is given by

$$\frac{dv(t)}{dt} = \hat{V}\omega_c^2 t e^{-\omega_c t} \qquad (9)$$

Now assuming the logic threshold is midway between logic '0' and logic '1' levels, from (8), the PFD will be triggered at time $\omega_c t' = 1.68$ giving a slope at this threshold of

$$\left.\frac{dv(t)}{dt}\right|_{t'} = 2f_c\hat{V} \qquad (10)$$

The equivalent rectangular noise bandwidth of a first-order low-pass filter is [8]:

$$B = \frac{\pi}{2} f_c \qquad (11)$$

and so the noise admitted to the PFD has an amplitude of

$$V_n = \sqrt{kTBFR} = \sqrt{\frac{\pi kTFf_cR}{2}} \quad [\mathrm{V(rms)}] \qquad (12)$$

where F is the noise factor and R is the impedance of the PFD input. This additive noise is converted into timing jitter according to the slope of the input pulse train.

$$\text{Timing jitter} = V_n \Big/ \left.\frac{dv(t)}{dt}\right|_{t'} \quad [\mathrm{s(rms)}] \qquad (13)$$

where it is clear that a high input slew rate, especially across the logic trip point, is preferable to reduce the translation of thermal noise at the PFD input to timing jitter. Since this jitter is independently generated at each PFD input and is uncorrelated, it is present on both edges of the output pulses giving a total RMS value of 2 times this result,

$$\begin{aligned} \Delta t &= \sqrt{2} V_n \Big/ \left.\frac{dv(t)}{dt}\right|_{t'} \\ &= \frac{1}{2\hat{V}} \sqrt{\frac{\pi kTFR}{f_c}} \quad [\mathrm{s(rms)}] \end{aligned} \qquad (14)$$

By substitution in (4), the FOM can be defined as,

$$\begin{aligned} FOM &= 4\pi^2 \Delta t^2 \\ &\frac{\pi^3 kTFR}{\hat{V}^2 f_c} \\ &= 10\log_{10}\left[\frac{\pi^3 kTFR}{\hat{V}^2 f_c}\right] \quad [\mathrm{dBc/Hz^2}] \end{aligned} \qquad (15)$$

Using this expression with the following approximate parameters for an ECL phase/frequency detector: $T = 300\,\mathrm{K}$, $F = 6$, $R = 50\,\mathrm{k\Omega}$ (DC input termination), $\hat{V} = 0.7\,\mathrm{V}$ and $f_c = 400\,\mathrm{MHz}$, gives an FOM of $-217\,\mathrm{dBc/Hz^2}$ which is in good agreement with experimentally observed values.

3 Experimental results

The experimental results presented here are based on measurements of a number of complete PLL synthesisers. Using the residual phase-noise measurement technique [9], and three identical synthesiser device-under-test units, each combination of these pairs was measured against one another. For each measurement, the sampling frequency was successively doubled, providing three sets of measurements at each sampling frequency while maintaining a constant output frequency (1638.4 MHz). Using processing available within the phase noise measurement software, the absolute noise profile of each synthesiser was determined. Using this technique, the clean correlated reference signal in each measurement for each pair of synthesisers being measured is removed, exposing the device flicker-noise profile and the in-band phase noise plateau (Fig. 3). From any one of the in-band noise plateaus thus measured, the FOM can be calculated.

Fig. 3 shows a plot of one synthesiser test unit from all three used after processing by the phase-noise software, for all sampling frequency settings. Contained within this figure is flicker noise and the effects of aliasing of wideband jitter noise, giving rise to the FOM values discussed earlier. The FOM of merit for the PFD used can be calculated from (5) for the 800 kHz sampling frequency in-band phase noise, measured in Fig. 3:

$$\begin{aligned} FOM &= -92.64\,\mathrm{dBc/Hz} + 10\log_{10}(800\,\mathrm{kHz}) \\ &\quad - 20\log_{10}(1638.4\,\mathrm{MHz}) \\ &= -217.9\,\mathrm{dBc/Hz^2} \end{aligned} \qquad (16)$$

With the overlay of each successive doubling in sampling frequency, a 3 dB improvement of measured in-band phase noise results. This is predicted by (5) and (6), where each successive doubling of sampling frequency increases the amount of uncorrelated jitter noise by a $10\log_{10}(f_s)$ factor, while concurrently halving the value of N for the $20\log_{10}(N)$ term, arriving at an overall 3 dB noise reduction, for a fixed output frequency. An explanation for each noise level measured in Fig. 3, using (6) is presented diagrammatically in Fig. 4.

Equation (6) suggests that the in-band phase noise profile will change with output frequency in a $20\log_{10}(f_o)$ manner for a constant sampling frequency. The measured results shown in Fig. 5 were taken using a suitably buffered PFD output to drive a signal generator emulating a VCO. In this way, the measurement could be repeated over octave frequency ranges. As expected, a $20\log_{10}(f_o)$ slope is observed.

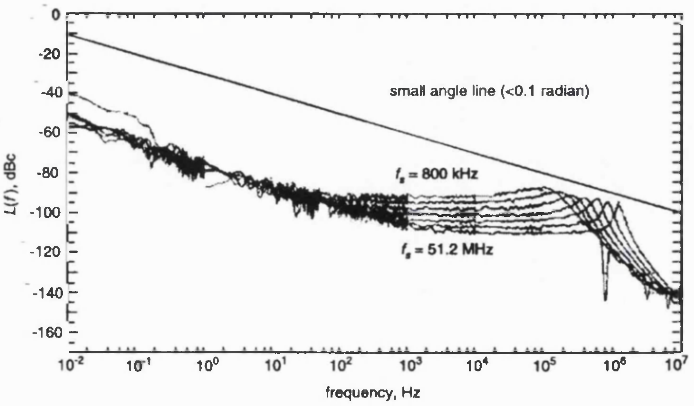


Fig. 3 *Measured phase noise profiles in third-order, type II PLL synthesiser for PFD operating frequencies of 0.8, 1.6, 3.2, 6.4, 12.8, 25.6 and 51.2 MHz*

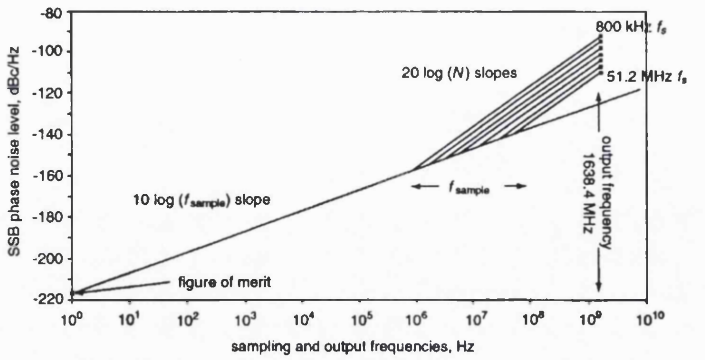


Fig. 4 *White noise plateau for different sampling frequencies and $f_o = 1638.4$ MHz*

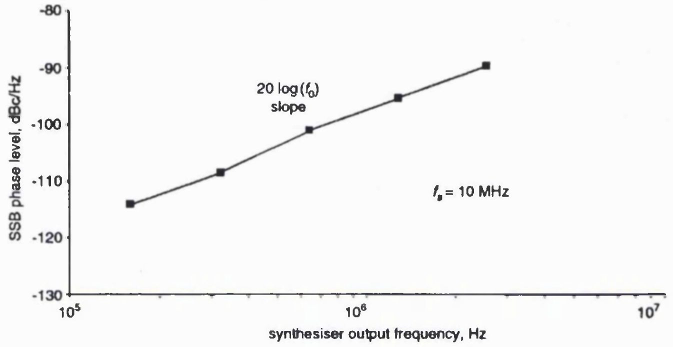


Fig. 5 *Effects of different output frequencies on in-band phase noise*

4 Discussion

This model highlights the limitations of some operational aspects of synthesisers and indicates how they might be optimised. From the analysis offered in Section 2.3, with (7) representing one type of input waveform characteristic, it can be seen that there is an optimum PFD logic switching point to minimise the translation of PFD input noise to PFD output timing jitter. Comparing this waveform with other theoretical waveform shapes considered during the course of this study (Fig. 6), the intuitive link between bandwidth and optimal switching point can be demonstrated, illustrating the need to optimise the bandwidths of both the N and R divider outputs to PFD input circuits. In general, a wider bandwidth should offer better performance.

For any given waveform, a further degree of freedom open to the synthesiser chip designer is the choice of logic switching point to ensure the PFD is switched at the peak change of slope for that incoming waveform. Therefore, for any given digital synthesiser chip design, both the bandwidth and the switching point of the PFD input interface circuits need to be optimised for minimal thermal noise translation. For example, from Fig. 6, the optimum threshold for the PFD model considered in this paper is 28% of the logic level swing, suggesting that the more typical 50% switching threshold is somewhat suboptimal.

To minimise thermal noise voltage, the PFD logic circuit should be designed for minimum input impedance and noise figure, as suggested by (15).

One observation from Fig. 4, is that, with a PFD operating at the PLL output frequency of 1638·4 MHz in this example, the best achievable white noise plateau value is ≈ −124 dBc/Hz. Consequently, no sigma–delta based fractional synthesiser, using this PFD, can ever achieve a plateau noise floor better than this value, irrespective of its close-in noise suppression because of the jitter performance of this PFD. Sigma–delta noise shaping applies only to noise contributed by the fractional-N division ratio switching process and has no bearing on the more fundamental limitation considered here, PFD noise.

5 Conclusions

This paper has presented a theoretical basis for a technique used to predict the in-band phase noise plateau level for a given synthesiser configuration based on a figure of merit (FOM) approach. The FOM equates to a normalisation of the PFD plateau noise, which can easily be measured and used to compare different synthesiser chips in the first stages of any selection process. An analysis has been presented that enables the FOM to be predicted for a given PFD from device parameters. Using this method, it has been shown how the in-band phase noise of the PFD can then be predicted for a given synthesiser frequency configuration. It has further been shown that the typical 50% PFD switching threshold is somewhat suboptimal and that a lower threshold should provide better noise performance. By inference from these analyses, several options are open to the synthesiser designer to improve the in-band phase noise performance of a given synthesiser, while in contrast, the

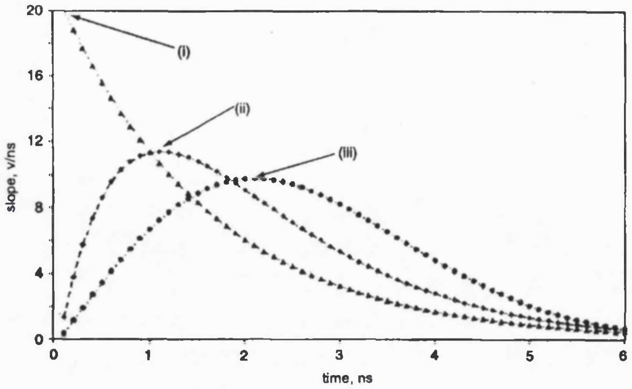


Fig. 6 *Optimal switching points for different PFD input slope types*
(i) Slope for expression $V(t) = V_{hi}\ (1-\exp(-\omega t))$; maximum at $0^{*}V_{hi}$
(ii) Slope for expression $V(t) = V_{hi}\ (1-\exp(-\omega t)-(\omega t)\exp(-\omega t))$; maximum at $0.28^{*}V_{hi}$
(iii) Slope for expression $V(t) = V_{hi}\ (1-\exp(-(\omega t)^2$; maximum at $0.4^{*}V_{hi}$

PFD is suggested as the fundamental limitation of both integer and fractional-N synthesisers, requiring attention before the full potential of fractional-N synthesisers can be realised.

Experimental results have been presented for an integer synthesiser showing excellent agreement with theory and, in particular, clearly demonstrating the anticipated 10 dB/decade improvement of synthesiser phase noise with increasing PFD operating frequency. The theory is equally applicable to fractional-N synthesiser applications.

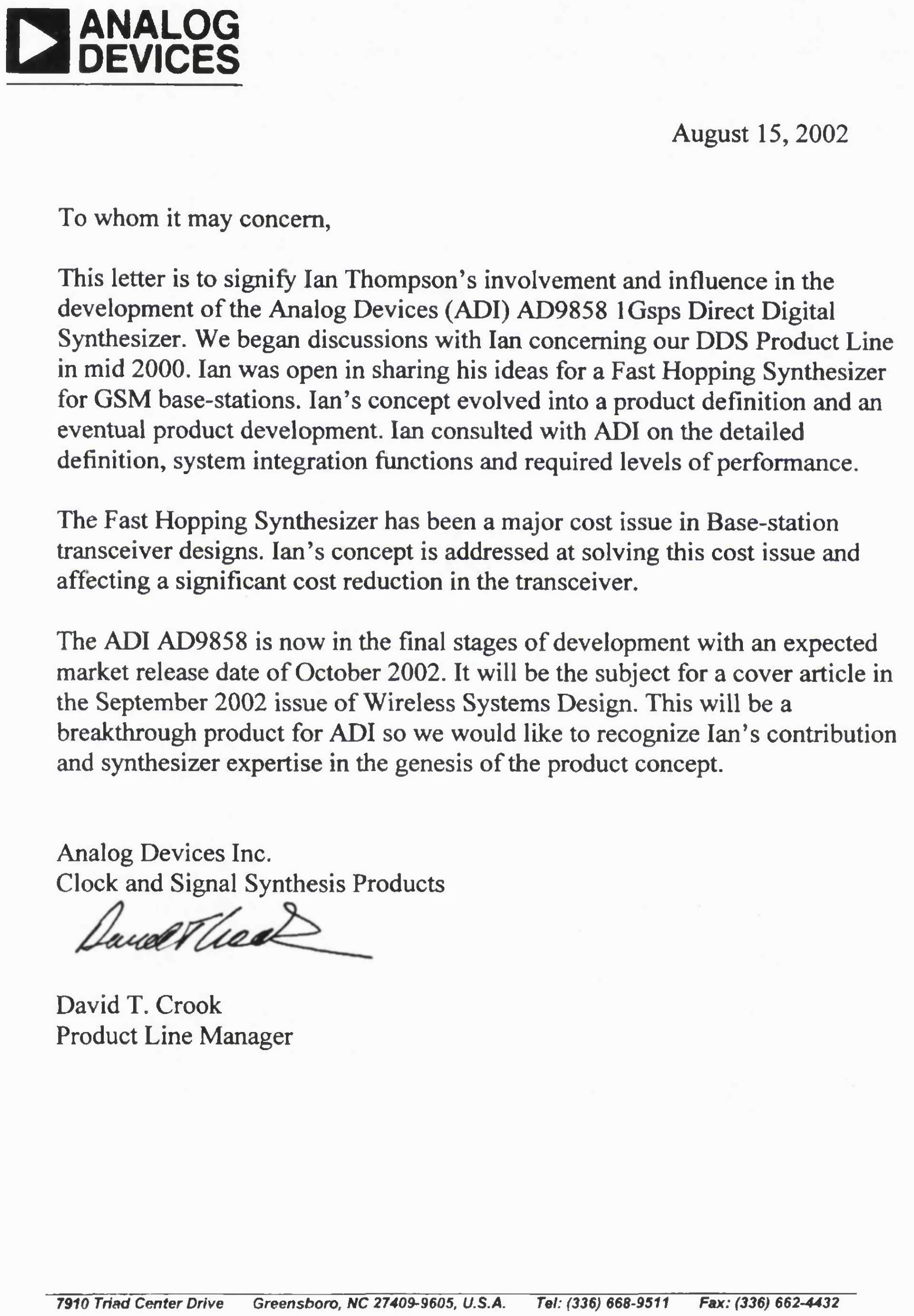
6 Acknowledgments

The authors thank A. Harney for his patience in answering all the applications questions, enabling a better in-depth understanding of a real device's operating characteristics and Dr T. Busby for the many interesting discussions during this work.

7 References

1 SHARPE, C.A.: 'A 3-state phase detector can improve your next PLL design' *EDN*, 1976, pp. 55–59
2 GARDNER, F.M.: 'Charge-pump phase-lock loops', *IEEE Trans. Commun.*, 1980, **28**, pp. 1849–1858
3 UTSI, V.: 'Evaluation of phase noise in emitter coupled logic'. Proceedings of Fifth International Conference on Radio receivers and associated systems, 1990, pp. 227–228
4 WILSON, M.P., and TOZER, T.C.: 'Synthesisers for low date-rate satellite receivers'. Proceedings of IEE conference on Frequency control and synthesis, Leicester, 1989, pp. 73–78
5 KROUPA, V.F.: 'Frequency synthesis: theory, design and applications' (Griffin, London, 1973)
6 WHITE, D.R.: 'The noise bandwidth of sampled data systems', *IEEE Trans. Instrum. Meas.*, 1989, **38**, (6), pp. 1036–1043
7 IEEE Std 1139-1999, The Institute of Electrical and Electronic Engineers, Inc., 345 East 47th Street, New York
8 BRENNAN, P.V.: 'Phase-locked loops, principles and practice' (McGraw-Hill, 1996)
9 ROHDE, U.L.: 'Microwave and wireless synthesizers theory and design' (John Wiley, New York, 1997)

A6 OTHER MAGAZINE PUBLICATIONS



ANALOG DEVICES

August 15, 2002

To whom it may concern,

This letter is to signify Ian Thompson's involvement and influence in the development of the Analog Devices (ADI) AD9858 1Gsps Direct Digital Synthesizer. We began discussions with Ian concerning our DDS Product Line in mid 2000. Ian was open in sharing his ideas for a Fast Hopping Synthesizer for GSM base-stations. Ian's concept evolved into a product definition and an eventual product development. Ian consulted with ADI on the detailed definition, system integration functions and required levels of performance.

The Fast Hopping Synthesizer has been a major cost issue in Base-station transceiver designs. Ian's concept is addressed at solving this cost issue and affecting a significant cost reduction in the transceiver.

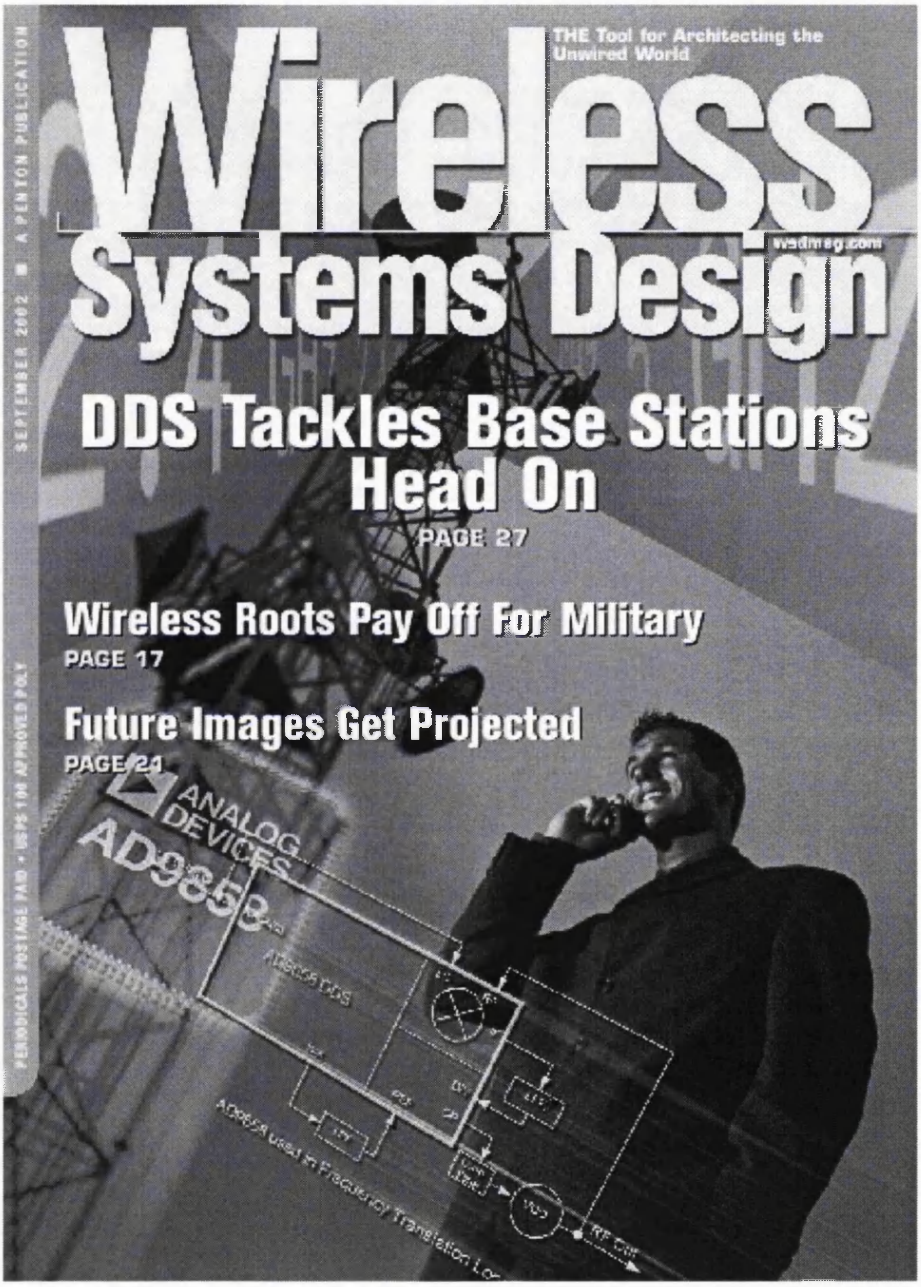
The ADI AD9858 is now in the final stages of development with an expected market release date of October 2002. It will be the subject for a cover article in the September 2002 issue of Wireless Systems Design. This will be a breakthrough product for ADI so we would like to recognize Ian's contribution and synthesizer expertise in the genesis of the product concept.

Analog Devices Inc.
Clock and Signal Synthesis Products

David T. Crook
Product Line Manager

7910 Triad Center Drive   Greensboro, NC 27409-9605, U.S.A.   Tel: (336) 668-9511   Fax: (336) 662-4432

A6 Other Magazine Publications


THE Tool for Architecting the Unwired World
Wireless Systems Design
wsdmag.com
SEPTEMBER 2002 ■ A PENTON PUBLICATION
DDS Tackles Base Stations Head On
PAGE 27
Wireless Roots Pay Off For Military
PAGE 17
Future Images Get Projected
PAGE 21
ANALOG DEVICES
AD9858
AD9858 DDS
AD9858 used in Frequency Translation Loop
RF Out
PERIODICALS POSTAGE PAID • USPS 100 APPROVED POLY

Reprinted with permission.

A7 MATLAB LISTING

```
%The following routine calculates the sampled phase noise profile for a simple PLL.
%  This routine is the master program, which calls up the following functions in order to work properly.
%
%     Sampled_Time_Constant_Calc.m
%     Time_Const_2_Component_Calc.m
%     SpurLevelCalc.m
%     SampThermalNoise.m
%     Time_Constant_Calculator.m
%     SampledGain.m
%     Linear_T23O.m
%     s_to_zT23O.m
%     SampVCONoise.m
%     SampledGainRC.m
%     RMSPhase.m
%     ResFMValue.m
%     ngrid.m                    %Note this file is available in the Matlab Control toolbox.
%
%  Ian Thompson,     28th March 2003

%-----------------------------------------------oOo------------------------------------------------------
%  Clear the Matlab platform for a new run of this routine.
clear all
close all
clc
format                                                  %Sets the numerically displayed format to the default value.
scrsz = get(0,'ScreenSize');                            %Determine the screen size for figure scaling.

%-----------------------------------------------oOo------------------------------------------------------
%  Define some constants.
pS=10^(-12); nS=10^(-9); uS=10^(-6); mS=10^(-3); Hz=1; kHz=10^3; MHz=10^6; GHz=10^9;
pF=10^(-12); nF=10^(-9); uF=10^(-6); Ohm=1; kOhm=10^3; MOhm=10^6; mA=10^-3; uA=10e-6; dB=1; dBcHz=1;

%-----------------------------------------------oOo------------------------------------------------------
%  Establish the loop operating parameters.
Fs=200*kHz; Fout=1068*MHz; Fref=52*MHz;
Atten=6*dB; Kvco=17.54*MHz*2*pi; PM=45; LBW=14*kHz; R=Fref/Fs; Ts=1/Fs; N=Fout/Fs; Kphi=8*mA/(2*pi);
        %1068MHz test synthesiser.

%-----------------------------------------------oOo------------------------------------------------------
%  Defining the known noise sources.....
%Reference Oscillator Noise.
Ref_3dB= 12*kHz; Ref_9dB= 3*kHz; RefNoise_Plateau=-145;                     %52MHz VCXO
%VCO Noise.
VCO_3dB=1.7*MHz; VCO_9dB=1.3*kHz; VCONoise_Plateau=-158*dBcHz;             %Sorep EWSO1808 1068MHz VCO.
%Divider Noise, (this assumed to be the same for both the R and N dividers.
NDiv_3dB=300*Hz; NDiv_6dB=10*Hz; NDiv_Plateau=-232*dBcHz;
RDiv_3dB=5*kHz; RDiv_6dB=10*Hz; RDiv_Plateau=-232*dBcHz;
%Phase Detector Noise.
PD_3dB=100*Hz; PD_6dB=5; PD_Plateau=-220*dBcHz;                            %ADF4113 measured, Vd=3.3V, Vp=5V.
%Phase Detector Dead-Zone Width.
tau=18*nS;

%-----------------------------------------------oOo------------------------------------------------------
%  Loop Filter Calculations.
j=sqrt(-1);                                                     %Clearly define "j" value.
s=j*2*pi*LBW;                                                   %Establish the Laplace natural frequency.
%This first part corrects for the order hold of the filter.
if Atten==0                                                     %Type II, third Order filter.
   GsohLBW=((1-exp(-s*Ts))/s)^3*(s^2+(3/(2*Ts))*s+1/Ts^2);      %determine gain and phase of SOH sampling.
   DeltaPhase=180/pi*unwrap(angle(GsohLBW))
   DeltaGain=-20*log10(abs(1/Ts*GsohLBW))
Else                                                                    %Type II, fourth Order Filter.
   GsohLBW=((1-exp(-s*Ts))/s)^4*(s^3+(11/(6*Ts))*s^2+(2/(Ts^2))*s+(1/Ts^3));     %Determine the gain and phase of Third
OH sampling.
   DeltaGain=-20*log10(abs(1/Ts*GsohLBW))
   DeltaPhase=180/pi*unwrap(angle(GsohLBW))
end
%Calculating the loop filter time constants first.
[t1, t2, t3, C, Attn]=Sampled_Time_Constant_Calc(Kphi*(2*pi), LBW, PM, Atten, Kvco/(2*pi), N, Fs, DeltaGain, DeltaPhase, 3);
%Now to calculate the loop filter values.
[C1, C2, C3, R2, R3]=Time_Const_2_Component_Calc(t1, t2, t3, C)
%Determine the breakpoints thus calculated.
w1=1/t1; f1=w1/(2*pi); w2=1/t2; f2=w2/(2*pi);
if R3~=0                                                                %Look for the Order of the filter.
   w3=1/t3; f3=w3/(2*pi);
else
   t3=0; w3=0; f3=0;
end
fnn=sqrt(f1*f2);                                                        %The new mid point between t1 and t2.

%---------------------------------------------oOo-----------------------------------------------------
%  Defining the log sweep range and resolution.
StartFreq=100*Hz;
StopFreq=10*MHz;
%Two sets of frequency points are established depending upon the resolution required of the output graph.
R_LowRes=floor(50*(log10(StopFreq)-log10(StartFreq)));          %Set the number of points per decade and automatically
calculate
R_HighRes=floor(2000*(log10(StopFreq)-log10(StartFreq)));       %for non-overlapping lines in the Nichols plot.
%Performing the log sweep.....
NumPoints=1+R_LowRes*log10(StopFreq/StartFreq);                 %Calculate the number of points between the start and stop
frequencies.
f_LowRes=logspace(log10(StartFreq), log10(StopFreq), NumPoints);
NumPoints=1+R_HighRes*log10(StopFreq/StartFreq);                %Calculate the number of points between the start and stop
frequencies.
f_HighRes=logspace(log10(StartFreq), log10(StopFreq), NumPoints);
f=f_LowRes;                                                     %Start with low resolution frequency sweep.

%---------------------------------------------oOo-----------------------------------------------------
%  Start the loop calculations.
SpurLevels=SpurLevelCalc(Fs, tau, Kvco, Kphi, N, C, t1, t2, t3, f, 5, Fs/8, -173);        %Calculate the spur levels.
s=j*2*pi.*f;                                            %Frequency point matrix as complex values.
Filt=1./(s*C).*(1+s*t1)./((1+s*t2).*(1+s*t3));          %Loop filter response.
GH=1/N.*Kphi.*Kvco./s.*Filt;                            %Loop gain response.
Attn=((1/N*Kphi*Filt*Kvco./s)./(1+GH));
Attn_dB=20*log10(abs(Attn));
Attn_Deg=180/pi*unwrap(angle(Attn));
Spur_Levels=SpurLevels(find(SpurLevels>-153))           %Print the spurs found on the control screen.

%---------------------------------------------oOo-----------------------------------------------------
%  NOISE CALCULATIONS.
%Sampling Noise Calculations.
R_Div_Noise=RDiv_Plateau+10*log10(1+RDiv_3dB./f)+10*log10(1+RDiv_6dB./f)+20*log10(R); %This gives the absolute
divider noise profile.
N_Div_Noise=NDiv_Plateau+10*log10(1+NDiv_3dB./f)+10*log10(1+NDiv_6dB./f);            %This gives the absolute
divider noise profile.
Ref_Div =RefNoise_Plateau+10*log10(1+Ref_3dB./f)+20*log10(1+Ref_9dB./f);             %Build the theoretical
sampling noise profile. This is derived from the incoming reference signal seen at the r-divider output.
Samp_Freq_Noise=10*log10(10.^(Ref_Div/10)+10.^(R_Div_Noise/10));                     %Hence the actual sampling
noise profile.

%---------------------------------------------oOo-----------------------------------------------------
%  Figure plotting-some defaults.
set(0,'DefaultFigureColor','w',...
    'DefaultAxesColor','w',...
    'DefaultAxesXColor',[0.5 0.5 0.5],...
    'DefaultAxesYColor',[0.5 0.5 0.5],...
    'DefaultAxesZColor',[0.5 0.5 0.5],...
    'DefaultTextColor','k',...
    'DefaultLineColor',[0.5 0.5 0.5]);
```

```
%---------------------------------------------oOo------------------------------------------------------
% figure(1);
figure('Name', 'Sampling Phase Noise');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
semilogx(f,Samp_Freq_Noise, 'k', f, Ref_Div,'r--',f, R_Div_Noise, 'b--',f,Ref_Div+20*log10(R),'m--'); hold off;
h = findobj('type', 'axes');
set(h(1),'XScale','log','Xlim',[StartFreq StopFreq],...
  'Ylim',[-210 -80],'YTick',[-210:10:-80],'Yscale','linear');
set(h(1), 'XColor',[0.5 0.5 0.5],'YColor',[0.5 0.5 0.5]); grid on;
title(['Reference VCXO and Sampling Signal SSB Phase Noise'],'FontSize',14);
xlabel(['Frequency, f, {Hz}'],'Color',[0,0,0],'FontSize',12);
ylabel(['SSB Phase Noise, L(f), {dBc/Hz}'],'Color',[0,0,0],'FontSize',12);
h = legend('Sampling Frequency Noise', 'Reference Divided Down', 'R Divider Noise', 'Input (VCXO) Reference Noise',1);

%---------------------------------------------oOo------------------------------------------------------
% VCO Noise Calculations + Loop Filter Thermal Noise.
VCONoise=VCONoise_Plateau+20*log10(1+VCO_3dB./f)+ 10*log10(1+VCO_9dB./f);          %Build free running VCO
noise profile.
%The linear and sampled thermal noise are compared as both are required.....
Thermal_Noise_Samp=SampThermalNoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, 290, Ts, -8, 8, f);
Thermal_Noise_Lin=SampThermalNoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, 290, Ts, 0, 0, f);

%---------------------------------------------oOo------------------------------------------------------
% figure(2);
figure('Name','VCO Phase Noise','Position',[100 100 scrsz(3)/2 scrsz(4)/2]);
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
semilogx(f,VCONoise,'b',f,Thermal_Noise_Samp,'r--', f, Thermal_Noise_Lin, 'm'); grid on; hold on;
h = findobj('type','axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
  'YLim',[-170 0],'YTick',[-170:10:0],'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Free Running VCO SSB Phase Noise and Loop Filter Resistor Thermal Noise'],'FontSize',12);
xlabel(['Frequency, f, {Hz}'],'Color',[0,0,0]);
ylabel(['SSB Phase Noise, L(f), {dBc/Hz}'],'Color',[0,0,0]);
h = legend('Free Running VCO Noise','Sampled Thermal Noise at PLL Output','Linear Thermal Noise at PLL Output',1);

%---------------------------------------------oOo------------------------------------------------------
% Phase Detector Noise Calculations.
PD_Noise_Norm=PD_Plateau+10*log10(1+PD_3dB./f)+10*log10(1+PD_6dB./f);         %Build PFD noise profile.
PD_Noise=PD_Noise_Norm+10*log10(Fs)+20*log10(N);                              %De-normalise the PFD noise profile.

%---------------------------------------------oOo------------------------------------------------------
% figure(3);
figure('Name', 'Phase Detector Noise');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
semilogx(f,PD_Noise, 'r'); grid on; hold on;
h = findobj('type', 'axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
  'YLim',[-140 -60],'YTick',[-140:10:-60],'XColor', [0.5 0.5 0.5], 'FontName', 'Times New Roman');
title(['Phase Detector Denormalised SSB Phase Noise'],'FontSize',12);
xlabel(['Frequency, f, {Hz}'],'Color',[0,0,0]);
ylabel(['SSB Phase Noise, L(f), {dBc/Hz}'],'Color',[0,0,0]);

%---------------------------------------------oOo------------------------------------------------------
% Loop gain linear and sampled responses.
GHLinResponse=Linear_T23O(R2, R3, C1, C2, C3, Kphi, Kvco, 1, f);                %N=1, no n-dividers.
GHSampResponse=s_to_zT23O(t1, t2, t3, Kphi, Kvco, 1, C, Ts, LBW, f);            %N=1, no n-dividers.
%Closed loop linear and sampled responses.
CLsampledFs=GHLinResponse./(1+1/N.*GHSampResponse);                            %This is the loop response to any noise at
the PFD reference or N divider inputs.
CLsampledPD=1/N.*GHLinResponse./(1+1/N.*GHSampResponse);                       %This is the loop response to noise out of
the PFD.
VCO_Loop_Gain=1./(1+1/N.*GHLinResponse);        %This should be the VCO response, (although this is an approximation).

%This saves the linear response but with the sampling phase offset.....
GHLinOffset_dB=20*log10(abs(GHLinResponse(:)))-20*log10(N);                    %Normalised value.
GHLinOffset_Deg=180/pi*unwrap((angle(GHLinResponse(:))));

%The following lines calculates the linear loop gain for a comparison. (The assumption is a very high sampling frequency).
[t1_lin, t2_lin, t3_lin, C_lin, Attn_lin]=Sampled_Time_Constant_Calc(Kphi*(2*pi), LBW, PM, Atten, Kvco/(2*pi), N, Fs*10^10,
0, 0, 3);
```

```
GH_Linear=(Kphi*Kvco./(N.*s)).*(1+s.*t1_lin)./(s.*C_lin.*(1+s.*t2_lin).*(1+s.*t3_lin));
GHLin_dB=20*log10(abs(GH_Linear(:)));
GHLin_Deg=180/pi*unwrap((angle(GH_Linear(:))));

%The following saves the sampled response.....
CLSamp_dB=20*log10(abs(CLsampledFs(:)))-20*log10(N);
CLSamp_Deg=180/pi*angle(CLsampledFs(:));

%---------------------------------------------oOo----------------------------------------------------
% The next few lines are specific to the Nichols plot and sampling.
f=f_HighRes;                                          %High resolution frequency points required.
GHSampResponse=s_to_zT23O(t1, t2, t3, Kphi, Kvco, N, C, Ts, LBW, f);
CE=zeros(length(f),2);
CE(:,1)=20*log10(abs(GHSampResponse'));
CE(:,2)=180/pi*unwrap(angle(conj(GHSampResponse')));
GHValue=s_to_zT23O(t1, t2, t3, Kphi, Kvco, N, C, Ts, LBW, LBW);

%---------------------------------------------oOo----------------------------------------------------
% Now to calculate some marker values for the following displays.
Marker(1)=20*log10(abs(GHValue));
Marker(2)=180/pi*unwrap(angle(GHValue));
%The following lines find the first minimum phase value.
CrossIndex=1; LastAngle=-180;
while CE(CrossIndex,2)>LastAngle
   LastAngle=CE(CrossIndex,2);
   CrossIndex=CrossIndex+1;
end
Marker(3)=CE(CrossIndex,1);
Marker(4)=CE(CrossIndex,2);
Marker(5)=f(CrossIndex);                              %Saves the marker frequency.

%The following lines find the first 0dB crossing point.
CrossIndex=1;
CrossIndexMax=length(CE);
while and(CE(CrossIndex,1)>0,CrossIndex<CrossIndexMax)
   CrossIndex=CrossIndex+1;
end
Marker(6)=CE(CrossIndex,1);
Marker(7)=CE(CrossIndex,2);
Marker(8)=f(CrossIndex);                              %Saves the marker frequency.

f=f_LowRes;                                           %Low resolution frequency points required.

%---------------------------------------------oOo----------------------------------------------------
% Plotting this filter response on the two axes.....
%figure(4);
figure('Name','Closed Loop Bode Plots');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
subplot(2,1,1); semilogx(f,CLSamp_dB,'b',f,Attn_dB,'r'); grid on; hold on;
h=plot([f1 f1],[-60 10],...
   [LBW LBW],[-60 10],...
   [f2 f2],[-60 10],...
   [fnn fnn],[-60 10]);                               %This places vertical marker lines at the breakpoints.
h = findobj('type','axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
   'YLim',[-60 10],'YTick',[-60:10:0,3,6,10],'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Closed Loop (Sampled and Linear) Gain Responses'],'FontSize',12);
xlabel(['Frequency, {Hz}'],'Color',[0,0,0]);
ylabel(['Closed Loop Gain, {dB}'],'Color',[0,0,0]);
h = legend('Sampled Magnitude','Linear Magnitude',3);

subplot(2,1,2); semilogx(f,CLSamp_Deg,'b',f,Attn_Deg,'r'); grid on; hold on;
h=plot([f1 f1],[-180 180],...
   [LBW LBW],[-180 180],...
   [f2 f2],[-180 180],...
   [fnn fnn],[-180 180]);                             %This places vertical marker lines at the breakpoints.
h = findobj('type','axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],'YTick', [-180:30:180],'YLim',[-180 180],...
   'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Closed Loop (Sampled and Linear) Phase Responses'],'FontSize',12);
```

```
xlabel(['Frequency, {Hz}'],'Color',[0,0,0]);
ylabel(['Closed Loop Phase, {Deg}'],'Color',[0,0,0]);
text(f1,-180,['  ',num2str(round(f1/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(LBW,-180,['  ',num2str(round(LBW/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(f2,-180,['  ',num2str(round(f2/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(fnn,-180,['  ',num2str(round(fnn/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
h = legend('Sampled Phase','Linear Phase',3);

%---------------------------------------------oOo---------------------------------------------------
% Final Noise Summation.
VCO_Loop_Noise=SampVCONoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, VCO_3dB, VCO_3dB,...
   VCO_9dB, VCONoise_Plateau, Ts, -8, 8, f);
Thermal_Loop_Noise=Thermal_Noise_Samp;                     %This output value is calculated earlier.
Samp_Freq_Loop_Noise=20*log10(abs(CLsampledFs))+Samp_Freq_Noise;
VCODiv_Loop_Noise=20*log10(abs(CLsampledFs))+N_Div_Noise;
PD_Loop_Noise=20*log10(abs(CLsampledPD))+PD_Noise;
Total_Noise=10*log10(10.^(VCO_Loop_Noise/10)+10.^(Thermal_Loop_Noise/10)+10.^(Samp_Freq_Loop_Noise/10)+...
   10.^(VCODiv_Loop_Noise/10)+10.^(PD_Loop_Noise/10));
Total_Noise=10*log10(10.^(Total_Noise/10)+10.^(SpurLevels/10));

%Some additional calculations useful for system work....
PhaseValue=RMSPhase(50, 8*kHz, Total_Noise, f);            %Degrees.
EVMValue=(2*sin((PhaseValue*pi/180)/2))*100;               %Per cent.
FreqValue=ResFMValue(300, 3*kHz, Total_Noise, f);          %Hz.

%---------------------------------------------oOo---------------------------------------------------
% figure (5);
figure('Name','SSB Phase Noise');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
semilogx(f,VCO_Loop_Noise,'b--',f,Thermal_Loop_Noise,'c--',f,Samp_Freq_Loop_Noise,'r--',f,PD_Loop_Noise,'m--',...
   f,VCODiv_Loop_Noise,'g--',f,Total_Noise,'k'); grid on; hold on;
h = findobj('type','axes');
set(h(1),'Xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
   'YLim',[-170 -50],'YTick',[-170:10:-50],'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Total SSB Phase Noise Of Synthesiser'],'FontSize',14);
xlabel(['Frequency, f, {Hz}'],'Color',[0,0,0],'FontSize',12);
ylabel(['SSB Phase Noise, L(f), {dBc/Hz}'],'Color',[0,0,0],'FontSize',12);
h = legend('VCO Noise','Loop Filter Thermal Noise','Sampling Noise','Phase Detector Noise','VCO Divider Noise','Total Noise',...
   ['RMS Phase Error ',num2str(PhaseValue,'%6.2f'),' Deg'],...
   ['EVM Value ',num2str(EVMValue,'%6.2f'),' %'],...
   ['Residual FM ',num2str(FreqValue,'%6.2f'),' Hz'],1);

%---------------------------------------------oOo---------------------------------------------------
% figure (6);
figure('Name','SSB Phase Noise 2');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
semilogx(f,VCONoise,'b--',f,Thermal_Noise_Samp,'c--',f,Samp_Freq_Noise+20*log10(N),'r--',f,PD_Noise,'m--',...
   f,N_Div_Noise+20*log10(N),'g--',f,Total_Noise,'k'); grid on; hold on;
h = findobj('type','axes');
set(h(1),'Xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
  'YLim',[-170 0],'YTick',[-170:10:0],'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['SSB Noise Profiles of all Noise Sources'],'FontSize',12);
xlabel(['Frequency, f, {Hz}'],'Color',[0,0,0]);
ylabel(['SSB Phase Noise, L(f), {dBc/Hz}'],'Color',[0,0,0]);
h = legend('VCO Noise','Loop Filter Thermal Noise','Sampling Noise','Phase Detector Noise','VCO Divider Noise','Total Noise',...
   ['RMS Phase Error ',num2str(PhaseValue,'%6.2f'),' Deg'],...
   ['EVM Value ',num2str(EVMValue,'%6.2f'),' %'],...
   ['Residual FM ',num2str(FreqValue,'%6.2f'),' Hz'],1);

%---------------------------------------------oOo---------------------------------------------------
% Now to look at the loop stability by analysing the characteristic equation.
%figure (7);
figure('Name','Nichols 1');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
ngrid('new');                                                                   %Plots the Nichols grid.
plot(CE(:,2),CE(:,1),'b',GHLin_Deg,GHLin_dB,'r',...
   Marker(2),Marker(1),'bx', Marker(4),Marker(3),'bx');grid on; hold on;
grid on;
h=get(gcf,'Children');                %Determines and returns the figure 'Children'.
```

```
set(h(1),'Xlim',[-270 -90],'XTick',[-270:15:-90],'Ylim',[-40 40],'YColor', [0.6 0.6 0.6],'XColor', [0.6 0.6 0.6],'FontName','Times New
Roman');hold off;
title(['Nichols Plot of Sampled Loop Response, GH.'],'FontSize',12);
ylabel(['Magnitude, {dB}'],'Color',[0,0,0]);
xlabel(['Phase, {Deg}'],'Color',[0,0,0]);
text(Marker(2),Marker(1),[' LBW ',num2str(round(10*LBW/10^3)/10,'%6.3g'),' kHz '],...
   'HorizontalAlignment','Right','FontSize',8,'Color','k');
text(Marker(2),Marker(1),[' ',num2str(Marker(1),'%6.3g'),' dB',' ',num2str(Marker(2),'%6.6g'),' Deg'],...
   'HorizontalAlignment','Left','FontSize',8,'Color','b');
if not(round(10*LBW/10^floor(log10(Fs)))/10==round(10*Marker(5)/10^floor(log10(Fs)))/10);%Check to see if these are the same
value and prevent overwrite.
   text(Marker(4),Marker(3),[' Min Phase ',num2str(round(100*Marker(5)/10^3)/100,'%6.3g'),' kHz '],...
      'HorizontalAlignment','Right','FontSize',8,'Color','k');
   text(Marker(4),Marker(3),[' ',num2str(Marker(3),'%6.3g'),' dB',' ',num2str(Marker(4),'%6.6g'),' Deg'],...
      'HorizontalAlignment','Left','FontSize',8,'Color','b');
end
h = legend('Sampled Loop Response','Ideal Linear Loop Response',...
   [' LBW ',num2str(round(100*LBW/10^3)/100,'%6.2f'),' kHz ',num2str(Marker(1),'%6.3g'),' dB ',num2str(Marker(2),'%6.3f'),'
Deg'],...
   [' Min Phase ',num2str(round(100*Marker(5)/10^3)/100,'%6.2f'),' kHz ',num2str(Marker(3),'%6.3g'),' dB
',num2str(Marker(4),'%6.3f'),' Deg'],2);

%---------------------------------------------oOo------------------------------------------------------
% Looking closer into the Nichols chart.
%figure(8);
figure('Name','Nichols 2');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
ngrid('new');                                                        %Plots the Nichols grid.
plot(CE(:,2),CE(:,1),'b',GHLin_Deg,GHLin_dB,'r',...
   Marker(2),Marker(1),'bx', Marker(4),Marker(3),'bx',Marker(7),Marker(6),'bx');grid on; hold on;
grid on;
h=get(gcf,'Children');            %Determines and returns the figure 'Children'.
set(h(1),'Xlim',[-185 -120],'XTick',[-185:5:-120],'Ylim',[-10 10],'YTick',[-10:1:10],'YColor', [0.5 0.5 0.5],...
   'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');hold off;
title(['Nichols Plot of Sampled Loop Response, GH.'],'FontSize',12);
ylabel(['Magnitude, {dB}'],'Color',[0,0,0]);
xlabel(['Phase, {Deg}'],'Color',[0,0,0]);
text(Marker(2),Marker(1),[' LBW ',num2str(round(10*LBW/10^3)/10,'%6.3g'),' kHz
'],'HorizontalAlignment','Right','FontSize',8,'Color','k');
text(Marker(2),Marker(1),[' ',num2str(Marker(1),'%6.3g'),' dB',' ',num2str(Marker(2),'%6.6g'),' Deg'],...
   'HorizontalAlignment','Left','FontSize',8,'Color','b');
if not(round(10*LBW/10^floor(log10(Fs)))/10==round(10*Marker(5)/10^floor(log10(Fs)))/10);    %Check to see if these are the
same value and prevent overwrite.
   text(Marker(4),Marker(3),[' Min Phase ',num2str(round(100*Marker(5)/10^3)/100,'%6.3g'),' kHz '],...
      'HorizontalAlignment','Right','FontSize',8,'Color','k');
   text(Marker(4),Marker(3),[' ',num2str(Marker(3),'%6.3g'),' dB',' ',num2str(Marker(4),'%6.6g'),' Deg'],...
      'HorizontalAlignment','Left','FontSize',8,'Color','b');
end
if not(round(10*LBW/10^floor(log10(Fs)))/10==round(10*Marker(8)/10^floor(log10(Fs)))/10);    %Check to see if these are the
same value and prevent overwrite.
   text(Marker(7),Marker(6),[' 0dB ',num2str(round(100*Marker(8)/10^3)/100,'%6.3g'),' kHz '],...
      'HorizontalAlignment','Right','FontSize',8,'Color','k');
   text(Marker(7),Marker(6),[' ',num2str(Marker(6),'%6.3g'),' dB',' ',num2str(Marker(7),'%6.6g'),' Deg'],...
      'HorizontalAlignment','Left','FontSize',8,'Color','b');
end
h = legend('Sampled Loop Response','Ideal Linear Loop Response',...
   [' LBW ',num2str(round(100*LBW/10^3)/100,'%6.2f'),' kHz ',num2str(Marker(1),'%6.3f'),' dB ',num2str(Marker(2),'%6.3f'),'
Deg'],...
   [' Min Phase ',num2str(round(100*Marker(5)/10^3)/100,'%6.2f'),' kHz ',num2str(Marker(3),'%6.3f'),' dB
',num2str(Marker(4),'%6.3f'),' Deg'],...
   [' 0dB ',num2str(round(100*Marker(8)/10^3)/100,'%6.3g'),' kHz ',num2str(Marker(6),'%6.3f'),' dB ',num2str(Marker(7),'%6.3f'),'
Deg'],2);

%---------------------------------------------oOo------------------------------------------------------
% Reviewing the loop filter response only.
%figure(9);
figure('Name','Loop Filter Response');
set(gcf,'PaperPositionMode','auto','PaperOrientation','landscape','PaperType','A4');
subplot(2,1,1); semilogx(f,GHLinOffset_dB,'b',f,GHLin_dB,'r'); grid on; hold on;
h = findobj('type','axes');
```

```
h=plot([f1 f1],[-60 50],'b',...
   [LBW LBW],[-60 50],'c',...
   [f2 f2],[-60 50],'r',...
   [fnn fnn],[-60 50],'m',...
   [f3 f3],[-60 50],'g');                                          %This places vertical marker lines at the breakpoints.
h = findobj('type','axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],...
   'YLim',[-60 50],'YTick',[-60:10:50],'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Loop Filter Gain Values, (shows the attenuation achieved).'],'FontSize',12);
xlabel(['Frequency, {Hz}'],'Color',[0,0,0]);
ylabel(['Filter Gain, {dB}'],'Color',[0,0,0]);
h = legend('Sampled Loop Response','Ideal Linear Loop Response',1);

subplot(2,1,2); semilogx(f,GHLinOffset_Deg,'b',f,GHLin_Deg,'r'); grid on; hold on;
h=plot([f1 f1],[-180 -90],'b',...
   [LBW LBW],[-180 -90],'c',...
   [f2 f2],[-180 -90],'r',...
   [fnn fnn],[-180 -90],'m',...
   [f3 f3],[-180 -90],'g');                                        %This places vertical marker lines at the breakpoints.
h = findobj('type','axes');
set(h(1),'xscale','log','Xlim',[StartFreq StopFreq],'YColor', [0.5 0.5 0.5],'YTick', [-180:15:-90],'YLim',[-180 -90],...
    'XColor', [0.5 0.5 0.5],'FontName','Times New Roman');
title(['Loop Filter Phase Values, (shows the effect of sampling phase).'],'FontSize',12);
xlabel(['Frequency, {Hz}'],'Color',[0,0,0]);
ylabel(['Filter Phase, {Deg}'],'Color',[0,0,0]);
text(f1,-180,['  ',num2str(round(f1/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(LBW,-180,['  ',num2str(round(LBW/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(f2,-170,['  ',num2str(round(f2/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(fnn,-170,['  ',num2str(round(fnn/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
text(f3,-180,['  ',num2str(round(f3/10^3),'%5.1d'),'kHz'],'VerticalAlignment','Bottom','FontSize',8);
h = legend('Sampled Loop Response','Ideal Linear Loop Response',1);

oooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooooo

function [t1, t2, t3, C, Attn]=Sampled_Time_Constant_Calc(Kphi, LBW, pm, Atten, Kvco, N, Fs, GainAdjust, PhaseAdjust,
Back_off)
% -------------------------oOo---------------------------
% This function returns the time constant values, t1, t2, t3, C in that order. The inputs are
%                                                   Kphi in mA;
%                                                   LBW in Hz,
%                                                   pm in Degrees,
%                                                   Atten in dB,
%                                                   Kvco in Hz/V,
%                                                   N has no units,
%                                                   Fs in Hz,
%                                                   GainAdjust in dB,
%                                                   PhaseAdjust in degrees,
%                                                   Back_off in dB.
% If the attenuation is set to zero then t3 and Attn become zero, as a third-order filter is calculated but with the back-off
% value so sensible loop filter component values can be used. Setting Back_off = 0 allows the theoretical max to viewed.
% To call this function use:
%
%   [t1, t2, t3, C, Attn]=Sampled_Time_Constant_Calc(0.0045, 20000, 45, 6, 15000000, 9000, 200000, 1.3456, 3.6083, 3);
% or [tt1, tt2, tt3, Cc, Attnt]=Sampled_Time_Constant_Calc(Kphi, LBW, PM*180/pi, Atten, Kvco, N, Fs, GainAdjust,
PhaseAdjust, 3);
%
% Then values for C1, C2, etc. are individually available as required. The quantity Attn gives a value set to the attenuation
% used in the following calculations. If the requested attenuation value, Atten, exceeds the theoretical maximum then this value
% of Atten will be truncated to this maximum value before the remainder of the calculations proceed.
%
% Note this is the same routine used earlier except it now returns Attn as well, and consequently will cause all unmodified
% Matlab routines to fail at the execution of this function.
%
% A 3dB back-off seems a reasonable value to avoid generating very large values of R3 that create large amounts of thermal noise!
%
%  All the mathematics used in this routine are tested in stand alone file:  TimeConstant.m
%
%  This function differs from Filter_Calculator as it uses a modified form of Paul's loop filter calculation methods.
% -------------------------oOo---------------------------
```

```
G=10^(GainAdjust/20);                   %This is the linear value of the adjusted gain value.
PM=(pm+PhaseAdjust)*pi/180;             %This is the modified phase margin converted to radians.
wn=2*pi*LBW;                            %Changes the LBW to the radian value of natural loop frequency.
ws=2*pi*Fs;                             %Changes the sampling frequency to its radian equivalent value.
k=G*Kphi*Kvco/N;                        %Calculate the loop gain including the gain adjustment value.

%First calculate the third-order filter values.
t1=(tan(PM)+sec(PM))/wn;
t2=1/(wn*(tan(PM)+sec(PM)));
c=(k/wn^2)*(tan(PM)+sec(PM));

if not(Atten==0)                        %Check to see if a fourth-order loop is requested.
  %First check to see if the requested attenuation value is realistic.  The maximum value of attenuation occurs at n=
  n=1/2*(2*cos(PM)-2*sqrt(cos(PM)^2-2*sin(PM)+sin(PM)^2+1))*ws/((sin(PM)-1)*wn);
  Atten_Max=20*log10(1+n^2);            %Atten_Max=10*log(1+n^2)/log(10);
  if Atten_Max<Atten
    Atten=Atten_Max-Back_off;           %The 0.5dB restricts the component values!
  end
  %Next calculate the values for the fourth-order filter values.
  tau3=1/ws*sqrt(10^(Atten/20)-1);      %tau3=1/ws*sqrt(10^(Atten/10)-1);
  tau1=t1;
  n=sqrt(10^(Atten/20)-1);
  tau2=(ws*cos(PM)-wn*n*(1+sin(PM)))/(wn*(ws*(1+sin(PM))+n*wn*cos(PM)));
  C=c*sqrt((1+wn^2*t2^2)/((1+wn^2*tau2^2)*(1+wn^2*tau3^2)));
  t1=tau1;
  t2=tau2;
  t3=tau3;
  C=C;
  Attn=Atten;
else
  t1=t1;
  t2=t2;
  t3=0;
  C=c;
  Attn=0;
End

ooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooooo

function [C1, C2, C3, R2, R3]=Time_Const_2_Component_Calc(tau1, tau2, tau3, C)
% ------------------------oOo--------------------------
% This function returns the loop filter values, C1, C2, C3, R2, R3 in that order.  The inputs are
%                                       tau1, tau2, tau3 time constants in rad/s;
%                                       C in Farads.
%
% To use this function a typical call might be....
%
%               [C1, C2, C3, R2, R3]=Time_Const_2_Component_Calc(tau1, tau2, tau3, C);
%
% All the mathematics used in this routine are tested in stand alone file:   TimeConstantGraphs.m
%
% ------------------------oOo--------------------------

if tau3==0         %i.e. a third-order filter.
  C1=(tau2/tau1)*C;
  C2=C-C1;
  C3=0;
  R2=tau1/C2;
  R3=0;
else               %i.e. a fourth-order filter.
  %The following line calculates the minimum point for R3 in terms of C1 and is the exact formula.
  C1=(1/6*(-27*tau2^3*tau1*tau3^2-9*tau2^3*tau3*tau1^2+36*tau2^2*tau3^2*tau1^2-...
    27*tau3^3*tau1*tau2^2-9*tau2*tau3^3*tau1^2+27*tau3^3*tau2^3+tau2^3*tau1^3+...
    3*tau2^2*tau1^3*tau3+3*tau2*tau1^3*tau3^2+tau3^3*tau1^3+...
    6*tau3*tau2*sqrt(3*tau3^3*tau1^3+3*tau2^3*tau1^3+9*tau2^2*tau1^3*tau3+9*tau2*tau1^3*tau3^2+...
    30*tau2^4*tau3*tau1+9*tau2^3*tau1*tau3^2-27*tau2^4*tau3^2-3*tau2^4*tau1^2-39*tau2^3*tau3*tau1^2+...
    9*tau2^2*tau3^2*tau1^2+9*tau3^3*tau1*tau2^2+27*tau3^3*tau2^3-39*tau2*tau3^3*tau1^2+...
    30*tau2*tau3^4*tau1-27*tau2^2*tau3^4-3*tau3^4*tau1^2)*tau1)^(1/3)/tau1+1/6*(-6*tau2^2*tau3*tau1-...
    6*tau2*tau3^2*tau1+9*tau2^2*tau3^2+tau2^2*tau1^2+2*tau2*tau1^2*tau3+tau3^2*tau1^2)/...
    (tau1*(-27*tau2^3*tau1*tau3^2-9*tau2^3*tau3*tau1^2+36*tau2^2*tau3^2*tau1^2-27*tau3^3*tau1*tau2^2-...
    9*tau2*tau3^3*tau1^2+27*tau3^3*tau2^3+tau2^3*tau1^3+3*tau2^2*tau1^3*tau3+3*tau2*tau1^3*tau3^2+...
    tau3^3*tau1^3+6*tau3*tau2*sqrt(3*tau3^3*tau1^3+3*tau2^3*tau1^3+9*tau2^2*tau1^3*tau3+...
    9*tau2*tau1^3*tau3^2+30*tau2^4*tau3*tau1+9*tau2^3*tau1*tau3^2-27*tau2^4*tau3^2-3*tau2^4*tau1^2-...
    39*tau2^3*tau3*tau1^2+9*tau2^2*tau3^2*tau1^2+9*tau3^3*tau1*tau2^2+27*tau3^3*tau2^3-39*tau2*tau3^3*tau1^2+...
    30*tau2*tau3^4*tau1-27*tau2^2*tau3^4-
3*tau3^4*tau1^2)*tau1)^(1/3))+1/6*(3*tau3*tau2+tau2*tau1+tau3*tau1)/tau1)*C/tau1;
  %This value of C1 is then used to calculate the remainder of loop filter values.
  C2=C1*C*(-tau1+tau2)*(-tau1+tau3)/(C1*tau1^2-C*tau2*tau3);
  C3=-(C1*tau1-C*tau2)*(C1*tau1-C*tau3)/(C1*tau1^2-C*tau2*tau3);
  R2=tau1*(C1*tau1^2-C*tau2*tau3)/(C1*C*(-tau1+tau2)*(-tau1+tau3));
  R3=-C*tau2*tau3*(C1*tau1^2-C*tau2*tau3)/(C1*tau1*(C1*tau1-C*tau2)*(C1*tau1-C*tau3));
end

ooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooo

function SpurLevels=SpurLevelCalc(fs, tau, Kvco, Kphi, N, C, t1, t2, t3, f, n, res, NoiseFloor);
%---------------------------------------------------------------------------------------------------------
% This function calculates the sampling spurs for the given frequency points supplied.
% These calculations are based on the "log Sampling Spurs" files written in both MathCad and Matlab.
%
% The inputs are.....
%       fs is the sampling frequency in Hertz,
%       tau is the equivalent charge pump on period in seconds,
%       Kvco is the VCO gain in radians/V,
%       Kphi is the phase detector gain in mA/radian,
%       N is the division value,
%       C is the loop filter gain,
%       t1, t2 and t3 are the loop filter time constants in radians,
%       f is the frequency vector for the calculations,
%       n is the maximum spur harmonic required,
%       res is the frequency resolution used for the frequency search. In practice this value could be set to fs/2.
%       NoiseFloor is the absolute noise floor if no spur is found.
%
% The output is the vector "SpurLevels", of log amplitudes, specified in terms of dBc.
%
% To use this function:
%                     SpurLevels=SpurLevelCalc(fs, tau, Kvco, Kphi, N, C, t1, t2, t3, f, n, fs/8, -153);
%
% Note because of the very narrow sampling spurs created which may not fall on the exact frequencies supplied in f some
% interpolation has been added to make sure the spurs are not missed due to poor frequency resolution. Also the translation
% from double sided to single sided, (6dB difference), has been accounted for.
%
% Ian Thompson   14th June 2002
%---------------------------------------------------------------------------------------------------------

j=sqrt(-1);
T=1/fs;                         %Calculate sampling period.
ws=2*pi*fs;

SpurLevels(1:length(f))=NoiseFloor;    %Set all values to the noise floor. (Assumes excellent signal noise, i.e. no aliasing).

%First the spur vector values needs to be identified.
for h=1:n
  k=find(h*fs-res < f & f < h*fs+res);   %This will return a zero value if not found.
  if length(k)>1                         %else the noise floor value will be returned.
    c=1;                                 %This and the next few lines finds the closest frequency to the current sampling
frequency.
    while and(abs(f(k(c))-h*fs)>abs(abs(f(k(c+1))-h*fs)),c<length(k)-1)
      c=c+1;
    end
    s=j*2*pi*f(k(c));                    %Calculate the s-domain frequency value.
    Filt=1/(s*C)*(1+s*t1)/((1+s*t2)*(1+s*t3));     %Loop filter gain.
    GH=1/N*Kphi*Kvco/s*Filt;                       %Loop gain.
    Attn=abs((1/N*Kphi*Filt*Kvco/s)/(1+GH));       %Loop gain from spur injection point.
    Spur=ws*tau/pi;
    SpurLevels(k(c))=20*log10(Spur*Attn);
  end
end

ooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooo
```

```
function ThermNoise=SampThermalNoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, Tmp, Ts, nMin, nMax, f);
%-------------------------------------------------------------------------------------------------------------------
%This function calculates the thermal noise for a specific frequency value or set of frequency values in Hz, for the loop
%filter.  The returned value is the sampled value of thermal noise power seen at the VCO output.
%
%        Kvco is in radians/V.
%
%  The inputs are.....
%        Kphi phase detector constant, mA/rad,
%        C1, C2, C3, R2 and R3 are the loop filter component values,
%        Kvco is the VCO gain, in Hz/V,
%        N is the loop division value,
%        Tmp is the noise temparature
%        Ts is the sampling period in Hz,
%        nMin, nMax are the summation limits,
%        f is the frequency vector for the calculations.
%
%  The output is the vector "ThermNoise" as a vector of noise values in dBc/Hz.
%
%  One way of using this function is:
%
%        Thermal_Noise=SampThermalNoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, 290, Ts, -(StopFreq/Fs+1), 0, f);
%
%     Note: in practice nMax=0 is okay for single side-band phase noise power additions such as these.
%          Using (StopFreq/Fs+1) ensures one more sample point greater than required is calculated.
%
%  Ian Thompson    14th July 2002
%-------------------------------------------------------------------------------------------------------------------

j=sqrt(-1);
S=j*2*pi.*f;                                        %s-Domain frequency values.
ws=2*pi*1/Ts;                                       %Radian value of sampling period.

K=1.38*10^(-23);                                    %Boltzmann's Constant defined.
BW=1;                                               %Bandwidth = 1Hz normalised.
NF=10*log10(K*Tmp*BW/(10^(-3)));                    %Thermal noise floor dBm/Hz.

%Find the loop filter time constants.....
[C, t1, t2a, t2b, t3a, t3b]=Time_Constant_Calculator(C1, C2, R2, R3, C3);
t2=t2a; t3=t3a;

%Calculate the gain from sampler output to PLL output.....
SampleValues=SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, nMin, nMax, f);      %Sampled response test, n=-inf to inf.
Num=SampledGain(Ts, 1, Kphi, C, t1, t2, t3, Kvco, 0, 0, f);                     %Numerator values.
E2Cs=Num./(1+SampleValues);                                                     %This runs from the E(s) sampler input to
C(s) PLL output.

%Setting up some time constants.....
A=C1*C2*R2*R3;
B=R2*C2+R3*C2+R3*C1;
D=C1*C2*C3*R2*R3;
E=C2*C3*(R2+R3)+C1*(C3*R3+C2*R2);
F=C1+C2+C3;
S2=S.*S;
FiltLin=(A.*S2+B.*S+1)./(S.*(D.*S2+E.*S+F));

%Calculate the linear noise component, if required.....
if and(nMin<=0, nMax>=0)                                %Check to find if n=0 value is required.
   FiltReal=real(FiltLin);
   NoiseVolts=sqrt(4*K*Tmp*BW.*FiltReal)./sqrt(2);      %Linear value of real part of loop filter transfer function.
   LinValuesDen=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, 0, 0, f));      %Linear response test.
   LinValuesNum=abs(SampledGain(Ts, 1, 1, 1, 0, 0, 0, Kvco, 0, 0, f));
   P=LinValuesNum./LinValuesDen;
   LinNoise=P'.*NoiseVolts;
end
clear FiltReal NoiseVolts LinValuesNum LinValuesDen S2

if nMin<=-1                                             %For all terms less than -1 called for.
   SampNoise=0; GsSamp=0;                               %Reset the sum values.
   for n=nMin:-1;                %These few lines apply only to the frequency dependent terms.
      s=S-j*n*ws;                %Calculate the magnitude of the complex sampled frequency vector for this value of n.
      s2=s.*s;
      Filt=(A.*s2+B.*s+1)./(s.*(D.*s2+E.*s+F)+eps);          %Filter calculation.
      FiltReal=real(Filt);                                   %Find the real part of the loop filter transfer function.
      NoiseVolts=sqrt(4*K*Tmp*BW.*FiltReal)./sqrt(2);        %Calculate the equivalent noise Voltage.
      KvcoSamp=Kvco./s;                        %Sampled VCO term.
      SampNoise=NoiseVolts.*KvcoSamp+SampNoise;              %Complete the running summation of these terms.
   end
   Den1=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, nMin, -1, f)); %Sampled response test, n=-inf to -1.
   Num1=1/N;
   Samp_T_Function1=abs(E2Cs).*Num1./Den1;                   %n=-inf to -1.
   SNegNoise=Samp_T_Function1'.*SampNoise;                   %Calculate the sampled noise for n=-inf to -1.
else
   SNegNoise=zeros(1,length(f));
end
clear n s Filt FiltReal NoiseVolts KvcoSamp SampNoise Samp_T_Function1 Num1 Den1

if nMax>=1                                                   %For all terms greater than 1 called for.
   SampNoise=0; GsSamp=0;                                    %Reset the sum values.
   for n=1:nMax;                 %These few lines apply only to the frequency dependent terms.
      s=S-j*n*ws;                %Calculate the magnitude of the complex sampled frequency vector for this value of n.
      Filt=(A.*s2+B.*s+1)./(s.*(D.*s2+E.*s+F)+eps);          %Filter calculation.
      FiltReal=real(Filt);                                   %Find the real part of the loop filter transfer function.
      NoiseVolts=sqrt(4*K*Tmp*BW.*FiltReal)./sqrt(2);        %Calculate the equivalent noise Voltage.
      KvcoSamp=Kvco./(s+eps);                                %Sampled VCO term.
      SampNoise=NoiseVolts.*KvcoSamp+SampNoise;              %Complete the running summation of these terms.
   end
   Den2=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, 1, nMax, f));   %Sampled response test, n=1 to inf.
   Num2=1/N;
   Samp_T_Function2=abs(E2Cs).*Num2./Den2;                   %n=1 to inf.
   SPosNoise=Samp_T_Function2'.*SampNoise;                   %Calculate the sampled noise for n=1 to inf.
else
   SPosNoise=zeros(1,length(f));
end
clear n s Filt FiltReal NoiseVolts KvcoSamp SampNoise Samp_T_Function2 Num2 Den2

%Adding all the noise terms together.....
D=abs(LinNoise)+abs(SNegNoise)+abs(SPosNoise);   %Linear summation of all calculated noise Voltage components.
NP=20*log10(D);                                  %Convert Noise Voltages to noise power in terms of dBc/Hz.
ThermNoise=10*log10(10.^(NP./10)+10^(NF/10));

oooooooooooooooooooooooooooooooooooooooooooooo0ooooooooooooooooooooooooooooooooooooooooooooooooooooooo

function [C, T1, T2a, T2b, T3a, T3b]=Time_Constant_Calculator(C1, C2, R2, R3, C3)
% -------------------------o0o---------------------------
% This function requires the values for components C1, C2, R2, R3 and C3, in that order. If second Order loop filter values are
% input, (i.e. R3=0), then time constant T2a=T2b=T2 and T3a=T3b=0 are returned.
% To call this function use:
%
%                   [C, T1, T2a, T2b, T3a, T3b]=Time_Constant_Calculator(C1, C2, R2, R3, C3);
%
% These calculations were derived in Maple file: LoopFilter 2.mws
% -------------------------o0o---------------------------

C=C1+C2+C3;
T1=R2*C2;
if R3~=0                                                     %i.e. R3 has a value.
   T2a=(-1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2+sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-2*...
      R2*C2*C1^2*R3*C3-2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+...
      R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+R3^2*C3^2*C2^2))*C1/(C1+C2+C3)-1/2*(R2*C2*C1+R2*C2*C3+...
      R3*C3*C1+R3*C3*C2+sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-2*R2*C2*C1^2*R3*C3-2*R2*C2^2*C1*R3*C3+...
      R2^2*C2^2*C3^2-
2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+R3^2*C3^2*C2^2))*...
      C2/(C1+C2+C3)-1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2+sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-
2*R2*C2*...
      C1^2*R3*C3-2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-
2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+...

2*R3^2*C3^2*C1*C2+R3^2*C3^2*C2^2))*C3/(C1+C2+C3)+R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2)/(C1+C2+C3);
```

```
  T2b=(-1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2-sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-...
    2*R2*C2*C1^2*R3*C3-2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+...
    R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+R3^2*C3^2*C2^2))*C1/(C1+C2+C3)-
1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+...
    R3*C3*C2-sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-2*R2*C2*C1^2*R3*C3-
2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-...

2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+R3^2*C3^2*C2^2))*C2/(C1+C2+C3)
-...
    1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2-sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-
2*R2*C2*C1^2*R3*C3-...
    2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-
2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+...
    R3^2*C3^2*C2^2))*C3/(C1+C2+C3)+R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2)/(C1+C2+C3);
  T3a=1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2+sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-
2*R2*C2*C1^2*R3*C3-...
    2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-
2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+...
    R3^2*C3^2*C2^2))/(C1+C2+C3);
  T3b=1/2*(R2*C2*C1+R2*C2*C3+R3*C3*C1+R3*C3*C2-sqrt(R2^2*C2^2*C1^2+2*R2^2*C2^2*C1*C3-
2*R2*C2*C1^2*R3*C3-...
    2*R2*C2^2*C1*R3*C3+R2^2*C2^2*C3^2-
2*R2*C2*C3^2*R3*C1+2*R2*C2^2*C3^2*R3+R3^2*C3^2*C1^2+2*R3^2*C3^2*C1*C2+...
    R3^2*C3^2*C2^2))/(C1+C2+C3);
else
  T2a=R2*C1*C2/(C1+C2);
  T2b=T2a;
  T3a=0;
  T3b=0;
End

oooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooo

function Response=SampledGain(T, N, Kphi, C, t1, t2, t3, Kvco, nMin, nMax, f);
%---------------------------------------------------------------------------------------------------------------
% This function calculates the sampled gain summation response with time constant inputs..
% The companion is "SampledNoise.m".
%
% The inputs are.....
%       T is the sampling period in Hz,
%       N is the division value,
%       Kphi is the phase detector gain in mA/radian,
%       C is the loop filter gain,
%       t1, t2 and t3 are the loop filter time constants in radians,
%       Kvco is the VCO gain in radians/V,
%       nMin, nMax are the summation limits,
%       f is the frequency vector for the calculations.
%
% The output is the vector "Response" is a vector of complex frequency values.
%
% One way of using this function is:
%                         Response=SampledGain(T, N, Kphi, C, t1, t2, t3, Kvco, -10, -1, f);
%
% Note this function detects whether the filter is required or not, i.e. when all time constants are set to zero. All other
%    values can be set to unity to exclude their impact from the overall result returned.
%
% Ian Thompson   28th June 2002
%---------------------------------------------------------------------------------------------------------------

j=sqrt(-1);
ws=2*pi*1/T;                                          %Radian value of sampling frequency.

gh=zeros(1,length(f));
for n=nMin:nMax;                              %These few lines apply only to the frequency dependent terms.
  s=j*2*pi.*f+j*n*ws;                         %Calculate the complex sampled frequency vector for this value of n.
  if and(t1==0, t2==0)                        %Catches when the loop filter is not required.
    temp=1./s;                                %VCO integration term only.
  else
    if Kvco==1                                %Check to see if the VCO gain is required.
      temp=(1+s.*t1)./(s.*(1+s.*t2).*(1+s.*t3));    %Loop filter term.
      else
        temp=(1+s.*t1)./(s.*s.*(1+s.*t2).*(1+s.*t3)+eps);          %Loop filter and VCO integration terms.
      end
    end
    gh=temp+gh;
end
K=1/N*Kphi*1/C*Kvco;                               %Determine the frequency independent constant value.
Response=K.*gh(:);                                 %Return the result.

ooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooo

function Response=Linear_T23O(R2, R3, C1, C2, C3, Kphi, Kvco, N, f);
%------------------------------------------------------------------------------------------------------------------------
%To use this function:
%  Response=Linear_T23O(R2, R3, C1, C2, C3, Kphi, Kvco, N, f);
%
%  GHLin_dB=20*log10(abs(Response'));
%  GHLin_Deg=-180/pi*unwrap(angle(conj(Response')));
%
%------------------------------------------------------------------------------------------------------------------------
Pi=pi;
j=sqrt(-1);
s=j*2*Pi.*f;
s2=s.*s;
s3=s.*s2;
K=Kphi*Kvco/N;

A=R2*C2;
B=1+s.*A;

C=R2*R3*C1*C2*C3;
D=R3*C3*C2+R2*C2*C3+R2*C1*C2+R3*C3*C1;
E=C1+C2+C3;
F=C.*s3+D.*s2+E.*s;
G=F.*s;           %The VCO integrator.

Response=K.*B./G;

ooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooo

function Response=s_to_zT23O(t1, t2, t3, Kphi, Kvco, N, C, T, LBW, f);
%------------------------------------------------------------------------------------------------------------------------
%This function gives the impulse sampled response of the loop function given the loop filter component values and gains:
%        t1 in rad/sec;
%        t2 in rad/sec;
%        t3 in rad/secs;
%        Kphi in A/rad;
%        Kvco in Hz/rad;   Note the 2*pi terms cancel between Kphi and Kvco.
%        N is the N divider value and is unit-less;
%        C is the total loop filter capacitance in F;
%        T is the sampling period in Seconds;
%        LBW is the loop bandwidth in Hz, but is not used in this function,
%        f is the frequency matrix in Hz.
%
%To use this function:
%  Response=s_to_zT23O(t1, t2, t3, Kphi, Kvco, N, C, Ts, LBW, f);
%
%  GHSamp_dB=20*log10(abs(Response'));
%  GHSamp_Deg=-180/pi*unwrap(angle(conj(Response')));
%
%The calculations for this come from      s_to_z T120.mws and s_to_z T130.mws  (Maple files), although I have
%                                  added the "T" term in the numerator.
%------------------------------------------------------------------------------------------------------------------------
Pi=pi;
j=sqrt(-1);
s=j*2*Pi.*f;
K=Kphi*Kvco/(N*C);
S=exp(s.*T);

%Some common constants.....
```

```
T2=exp(-T/t2);
M=(S-1).*(S-1);
N=S-T2;

if t3==0                                        %i.e. a third Order system.  See s_to_z T120.mws Maple file.
   A=(t2-t1)*T2+T-t2+t1;
   B=(t1-T-t2)*T2+t2-t1;
   Response=T*K.*S.*(A.*S+B)./(M.*N);
Else                                            %i.e. a fourth Order system.  See s_to_z T13O.mws Maple file.
   T3=exp(-T/t3);
   P=S-T3;
   T4=exp(-T*(t3+t2)/(t2*t3));
   A=(-t3^2+t3*t1)*T3+(t2^2-t2*t1)*T2-T*t3+t2*t1-t2^2+T*t2-t3*t1+t3^2;
   B=(-t2*t1+t2^2+T*t3-T*t2-t3*t1+t3^2)*T3+(-T*t2+t3*t1-t3^2+T*t3-t2^2+t2*t1)*T2+(-t2^2+t2*t1-t3*t1+t3^2)*T4-t3^2-
t2*t1+t3*t1+t2^2;
   C=(-t2^2+t2*t1)*T3+(-t3*t1+t3^2)*T2+(-t3^2+t3*t1+T*t2+t2^2-t2*t1-T*t3)*T4;
   Response=K.*T.*S.*(A.*S.*S+B.*S+C)./((t2-t3).*M.*N.*P);
end

oooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooo

function VCONoise=SampVCONoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, VCO_3dB, VCO_6dB, VCO_9dB, VCONoise_Plateau,
Ts, nMin, nMax, f);
%---------------------------------------------------------------------------------------------------------
%This function calculates the thermal noise for a specific frequency value or set of frequency values in Hz, for the loop
%filter.  The returned value is the sampled value of thermal noise power seen at the VCO output.
%
%        Kvco is in radians/V.
%
%  The inputs are.....
%        Kphi phase detector constant, mA/rad,
%        C1, C2, C3, R2 and R3 are the loop filter component values,
%        Kvco is the VCO gain, in Hz/V,
%        N is the loop division value,
%        VCO_3dB, VCO_6dB, VCO_9dB, VCONoise_Plateau, are the VCO noise breakpoints, Hz,
%        Ts is the sampling period in Hz,
%        nMin, nMax are the summation limits,
%        f is the frequency vector for the calculations.
%
%  The output is the vector "VCONoise" as a vector of noise values in dBc/Hz.
%
%  One way of using this function is:
%
%  VCONoise=SampVCONoise(Kphi, C1, C2, C3, R2, R3, Kvco, N, VCO_3dB, VCO_6dB, VCO_9dB, VCONoise_Plateau, Ts, -
(StopFreq/Fs+1), 0, f);
%
%     Note: in practice nMax=0 is okay for single side-band phase noise power additions such as these.
%           Using (StopFreq/Fs+1) ensures one more sample point greater than required is calculated.
%
%  Ian Thompson    15th August 2002
%---------------------------------------------------------------------------------------------------------

j=sqrt(-1);
S=j*2*pi.*f;                                    %s-Domain frequency values.
ws=2*pi*1/Ts;                                   %Radian value of sampling period.

K=1.38*10^(-23);                                %Boltzmann's Constant defined.
BW=1;                                           %Bandwidth = 1Hz normalised.

%Find the loop filter time constants.....
[C, t1, t2a, t2b, t3a, t3b]=Time_Constant_Calculator(C1, C2, R2, R3, C3);
t2=t2a; t3=t3a;

%Calculate the gain from sampler output to PLL output.....
SampleValues=SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, nMin, nMax, f);        %Sampled response test, n=-inf to inf.
Num=SampledGain(Ts, 1, Kphi, C, t1, t2, t3, Kvco, 0, 0, f);                       %Numerator values.
E2Cs=Num./(1+SampleValues);                     %This runs from the E(s) sampler input to C(s) PLL output.

%Some loop constant values.....
t1=R2*C2;
```

```
C=C1+C2+C3;
m=((R2*C2*(C1+C3))+(R3*C3*(C1+C2)))/C;
p=C1*C2*C3*R2*R3/C;
%FiltLin=(1+S.*t1)./(S.*C.*(1+S.*m+S.*S.*p));        %Linear filter response.
FiltLin=SampledGainRC(Ts, 1, 1, C1, C2, C3, R2, R3, 1, 0, 0, f)';

%Calculate the linear noise component, if required.....
if and(nMin<=0, nMax>=0)                              %Check to find if n=0 value is required.
   NoisePower=abs(10^(VCONoise_Plateau/10).*(1+VCO_3dB./f).*(1+VCO_6dB./f).*(1+VCO_9dB./f));
   LinValuesDen=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, 0, 0, f));          %Linear response test.
   P=1./LinValuesDen;
   P=P.*P;
   LinNoise=P'.*NoisePower;
end
clear FiltReal NoiseVolts LinValuesNum LinValuesDen

if nMin<=-1                                           %For all terms less than -1 called for.
   SampNoise=0; GsSamp=0;                             %Reset the sum values.
   for n=nMin:-1;                                     %These few lines apply only to the frequency dependent terms.
      Freq=f-n/Ts;               %Calculate the magnitude of the complex sampled frequency vector for this value of n.
      NoisePower=abs(10^(VCONoise_Plateau/10).*(1+VCO_3dB./Freq).*(1+VCO_6dB./Freq).*(1+VCO_9dB./Freq));
      SampNoise=NoisePower+SampNoise;                 %Complete the running summation of these terms.
   end
   Den1=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, nMin, -1, f));   %Sampled response test, n=-inf to -1.
   Num1=1/N;
   Samp_T_Function1=abs(E2Cs).*Num1./Den1;                        %n=-inf to -1.
   Samp_T_Function1=Samp_T_Function1.*Samp_T_Function1;           %Converts to power gain.
   SNegNoise=Samp_T_Function1'.*SampNoise;                        %Calculate the sampled noise for n=-inf to -1.
else
   SNegNoise=zeros(1,length(f));
end
clear n s Filt FiltReal NoiseVolts KvcoSamp SampNoise Samp_T_Function1 Num1 Den1

if nMax>=1                                            %For all terms greater than 1 called for.
   SampNoise=0; GsSamp=0;                             %Reset the sum values.
   for n=1:nMax;                                      %These few lines apply only to the frequency dependent terms.
      freq=f-n/Ts;               %Calculate the magnitude of the complex sampled frequency vector for this value of n.
      NoisePower=abs(10^(VCONoise_Plateau/10).*(1+VCO_3dB./Freq).*(1+VCO_6dB./Freq).*(1+VCO_9dB./Freq));
      SampNoise=NoisePower+SampNoise;                 %Complete the running summation of these terms.
   end
   Den2=abs(1+SampledGain(Ts, N, Kphi, C, t1, t2, t3, Kvco, 1, nMax, f));   %Sampled response test, n=1 to inf.
   Num2=1/N;
   Samp_T_Function2=abs(E2Cs).*Num2./Den2;                        %n=1 to inf.
   Samp_T_Function2=Samp_T_Function2.*Samp_T_Function2;           %Convert to a power gain.
   SPosNoise=Samp_T_Function2'.*SampNoise;                        %Calculate the sampled noise for n=1 to inf.
else
   SPosNoise=zeros(1,length(f));
end
clear n s Filt FiltReal NoiseVolts KvcoSamp SampNoise Samp_T_Function2 Num2 Den2

%Adding all the noise terms together.....
D=abs(LinNoise)+abs(SNegNoise)+abs(SPosNoise);   %Linear summation of all calculated noise Voltage components.
VCONoise=10*log10(D);                            %Convert Noise Voltages to noise power in terms of dBc/Hz.

ooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

function Response=SampledGainRC(T, N, Kphi, C1, C2, C3, R2, R3, Kvco, nMin, nMax, f);
%----------------------------------------------------------------------------------------------------------------------
%  This function calculates the sampled gain summation response with component values as inputs.
%  The companion is "SampledNoise.m".
%
%  The inputs are.....
%        T is the sampling period in Hz,
%        N is the division value,
%        Kphi is the phase detector gain in mA/radian,
%        C1, C2, C3, R2 and R3 are the loop filter component values,
%        Kvco is the VCO gain in radians/V,
%        nMin, nMax are the summation limits,
%        f is the frequency vector for the calculations.
%
```

```
% The output is the vector "Response" is a vector of complex frequency values.
%
% One way of using this function is:
%                              Response=SampledGain(T, N, Kphi, C1, C2, C3, R2, R3, Kvco, -10, -1, f);
%
% Note this function detects whether the filter is required or not, i.e. when all time constants are set to zero. All other
%     values can be set to unity to exclude their impact from the overall result returned.
%
% Ian Thompson   28th June 2002
%-----------------------------------------------------------------------------------------------------------------------

j=sqrt(-1);
S=j*2*pi.*f;                                  %s-Domain frequency values.
ws=2*pi*1/T;                                  %Radian value of sampling frequency.

t1=R2*C2;
C=C1+C2+C3;
m=((R2*C2*(C1+C3))+(R3*C3*(C1+C2)))/C;
p=C1*C2*C3*R2*R3/C;

gh=zeros(1,length(f));
for n=nMin:nMax;                              %These few lines apply only to the frequency dependent terms.
   s=j*2*pi.*f+j*n*ws;                        %Calculate the complex sampled frequency vector for this value of n.
   if and(C1==0, R2==0)                       %Catches when the loop filter is not required.
      temp=1./s;                              %VCO integration term only.
   else
      if Kvco==1                              %Check to see if the VCO gain is required.
         temp=(1+S.*t1)./(S.*(1+S.*m+S.*S.*p));    %Loop filter term term.
      else
         temp=(1+S.*t1)./(S.*S.*(1+S.*m+S.*S.*p)); %Loop filter and VCO integration terms.
      end
   end
   gh=temp+gh;
end
K=1/N*Kphi*1/C*Kvco;                          %Determine the frequency independent constant value.
Response=K.*gh(:);                            %Return the result.
```

oooooooooooooooooooooooooooooooooooooooooooooooooo0oooooooooooooooooooooooooooooooooooooooooooooooooooo

```
function PhaseValue=RMSPhase(Lmin, Lmax, Noise, f);
%-----------------------------------------------------------------------------------------------------------------------
% This function calculates the RMS phase error.
%
% The inputs are.....
%      Lmin is the lower integral limit in Hz,
%      Lmax is the upper integral limit in Hz,
%      Noise is the vector of noise to be integrated, in dBc/Hz
%      f is the frequency vector for the calculations.
%
% The output is the result "PhaseValue".
%
% One way of using this function is:
%
%         PhaseValue=RMSPhase(1/577e-6, 135*kHz, Total_Noise, f)      %Degrees.
%
%      Note it is fairly easy to subsequently calculate the equivalent EVM value.....
%
%         EVMValue=(2*sin((PhaseValue*pi/180)/2))*100                 %Per cent.
%
% Ian Thompson   24th Sept 2002
%-----------------------------------------------------------------------------------------------------------------------

Rad2Deg=2*180/pi;

IntVal=find(Lmin < f & f < Lmax);             %Find all values within the defined integration limits.

NoiseLin=10.^(Noise(IntVal)./10);             %Linearise all the noise powers within the integral limits.

%Now for the integration. This will be piecewise linear, (y=mx+c), between each frequency point.....
Int_Sum=0;
```