

NOVEL LOW COST SYNCHRONISATION NETWORK FOR
SPREAD SPECTRUM SYSTEMS

A DISSERTATION
SUBMITTED TO THE FACULTY OF ELECTRICAL ENGINEERING
OF THE UNIVERSITY OF CAPE TOWN
IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Joannes Gerard van de Groenendaal
BEng (ELEC) Cum Laude
August 1995

The University of Cape Town
Library
1000 University Road, Rondebosch, Cape Town
7700

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

© Copyright 1995 by Joannes Gerard van de Groenendaal
BEng (ELEC) Cum Laude
All Rights Reserved

Declaration

I hereby declare that the work contained in this dissertation is my own, both in concept and execution. It is being submitted for the degree of Doctor of Philosophy in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination in any other university,

Signed by candidate

Signature Removed

Candidate

Signed by candidate

Signature Removed

Supervisor: Dr R.M. Braun

Abstract

Novel Low Cost Synchronisation Network For Spread Spectrum Systems

Joannes (Johan) Gerard van de Groenendaal

University of Cape Town

Rondebosch

7700 South Africa

Spread Spectrum systems are found in many flavours, used in many applications and have existed since the early days of radio communications. The properties of spread spectrum do however place restrictions on the design, and often make the implementation expensive and complex. When using spread spectrum to provide a basic communications infrastructure, many factors need to be considered. These include supplying the appropriate technology at the right cost. To achieve this a trade-off against performance is often required.

One of the more difficult aspects of Spread Spectrum design is the synchronisation of the spreading waveform. The primary characteristic of pseudonoise sequence synchronisation is the need for two levels of synchronisation namely acquisition (course synchronisation) and tracking (fine synchronisation). In these networks (the term network is used to describe a circuit or system throughout the thesis.) a decision is required to switch between the two synchronisation modes. The two layer structure of the typical pseudonoise sequence synchronisation network can increase the overall cost of spread spectrum systems. The objective of the research was therefore to find solutions to reduce the overall cost and complexity of the synchronisation network. The synchronisation structure should *perform acquisition and tracking in a single structure*, and thereby be *low cost*.

To achieve the primary objective of this dissertation a mixture of theory, simulations and practical implementation was used. The basis of the investigation was a time-variant spectral evaluation of pseudonoise sequences. It is shown that by multiplying a differentiated pseudonoise sequence with another pseudonoise sequence, useful information is obtained

that can form the basis of a synchronisation network.

Using the conclusions drawn from this investigation, a low cost synchronisation network was proposed. The network was simple in design and through simulation was shown to achieve the accurate synchronisation in one structure. The network was also implemented and was shown to work. However, the network was parameter sensitive and not robust. A second structure based on the same principles was proposed. This network was shown, through simulation and practical results, to be effective, low cost and achieved the primary objective.

The low-cost synchronisation network is highly non-linear and therefore the use of linear techniques to analyse the network is not possible. The non-linear equation for the synchronisation network was derived and phase-plane techniques were applied. The chaotic nature of the network was clearly illustrated. Using phase-plane plots the effect of various parameters on the synchronisation of spreading sequences was illustrated

One important result of this analysis was how the network evaluates the code-phase - frequency grid. This grid is a representation of the tracking space for spread spectrum synchronisation networks. The traditional methods used to perform course acquisition evaluate the grid uniformly and deterministically. The novel low cost synchronisation network evaluates this grid in a random fashion.

The low cost spread spectrum synchronisation network was compared to the methods discussed in the literature. Based on a number of assumptions, locking times were compared and it was shown that the novel synchronisation network's locking times were of the same order.

One of the short-comings of the basic network is that it is sensitive to data modulation. This is a common problem with networks not using energy detectors. The problem was solved using a mixture of correlation and digital circuitry. The detection circuitry altered the loop conditions and with that improved the noise performance of the low cost synchronisation network. The effect of noise and carrier tracking on the novel low cost synchronisation network was also investigated. The network was evaluated down to an input signal-to-noise ratio of -18 dB, where it still achieved lock. It is important to realize that due to the structure of the synchronisation network it is more suited to low-noise spread spectrum applications.

While implementing the network, a novel low cost square-wave differentiator was designed. This signed edge detector has a number of advantages over standard differentiators. It

is wide-band, with the only limiting factors being the speed of the logic gates and the bandwidth of the multiplier. It is also shown to perform better at lower signal-to-noise ratios. Other practical aspects that were investigated in the search for a low cost synchronisation network, was the synchronous oscillator (a form of an injection-locked oscillator with a constant output amplitude).

The novel low cost spread spectrum synchronisation network achieved its objective of offering a single solution to acquisition and tracking without the need for an intermediate decision stage. The network was shown to operate at an input signal-to-noise ratio of -18 dB (in half the simulation bandwidth) and was shown to deal effectively with data modulation. The network is also low cost and easy to implement. It is important to realize that due to the structure of the synchronisation network, it is more suited to higher signal-to-noise ratio applications.

Dedicated to:
My parents,
my sisters
and
Lian.

Preface

The field of spread spectrum is diverse and the number of applications of this technology is forever increasing. One application where spread spectrum may play an important role is in the supply of communications to under developed nations, where villages are wide spread and often in areas where traditional communication systems may not be practical. A number of organisations have investigated the benefit of supplying communication as a method of upliftment. It was from this that the concept of the Virtual Telephone network was born and this in turn inspired my interest in finding low-cost solutions to supply telecommunications. While investigating appropriate technologies for a low cost solution, spread spectrum appeared as a possible solution. On closer examination, it was found that little had been done in the field of synchronisation, especially in the search for a unified approach. This was the starting block of my research.

I would like to thank the following people for their support over the last few years. Dr R.M. Braun for his guidance and spending time listening to many of my ideas. Mark Davidson for his unfailing support, his friendship and spending many hours proofing my thesis. The members of the Communications Research group, past and present for their friendship. My parents for their understanding and support over the last three years. Special thanks to Lian Chen for her love, kindness and understanding.

I would like to acknowledge the financial support of the Foundation of Research and Development (FRD) and the ALTRON group of Companies.

Contents

Declaration	iii
Abstract	iv
Preface	viii
Table of Contents	ix
List of Tables	xviii
List of Figures	xix
List of Symbols	xxvii
1 Introduction	1
2 Present Methods of Synchronisation	5
2.1 What is Spread Spectrum?	6
2.1.1 Direct Sequence Spread Spectrum	7
2.1.2 Code Division Multiple Access (CDMA)	9
2.1.2.1 What is the basis of CDMA?	10
2.1.2.2 How does the receiver synchronise?	10

2.1.2.3	Why use CDMA?	10
2.1.3	Processing Gain and Jamming Margin	11
2.2	Spread Spectrum Synchronisation	11
2.2.1	Sources of Uncertainty	12
2.2.1.1	Code Phase Uncertainty	12
2.2.1.2	Frequency Uncertainty	13
2.2.2	Two Pronged Approach to Synchronisation	13
2.3	Code Acquisition	13
2.3.1	The Time-Optimal Synchroniser	15
2.3.2	Detection of a Signal in Additive White Gaussian Noise	17
2.3.2.1	Fixed Integration Time Energy Detection	18
2.3.2.2	Multiple-Dwell Detection	18
2.3.2.3	Sequential Detection	20
2.3.3	Serial Search Techniques	20
2.3.3.1	Non-Uniform Uncertainty Region	21
2.3.3.2	Continuous Linear Sweep	21
2.3.4	Matched Filter Synchronisation	22
2.3.5	Parallel Acquisition of PN Sequences	22
2.4	Code Tracking	23
2.4.1	Optimal Tracking	23
2.4.2	Baseband Full-Time Early-Late Tracking Loop	26
2.4.3	Full-Time Early-Late Noncoherent Tracking Loop	29
2.4.4	Tau-Dither Early-Late Non-coherent Tracking Loop	31
2.5	Estimation Tracking Techniques	32

2.5.1	Synchronisation by Estimating the Received Spreading Code	33
2.5.1.1	Acquisition Time	35
2.5.1.2	Noise	36
2.5.1.3	Doppler Effects	36
2.6	Summary	37
3	Theoretical Concepts: The Search for Low Cost Synchronisation Methods	38
3.1	Research Objectives	38
3.2	Time-Variant Spectral Analysis	39
3.2.1	Characteristics of Pseudo-noise Sequences	39
3.2.2	Positive Edge Detection	40
3.2.3	Effect of Despreading a Signed Edge Detected (Differentiated) Pseudo-Noise Sequence	42
3.3	Summary	47
4	Novel Synchronisation Technique-Method One	48
4.1	System Description	48
4.2	Simulation of System	49
4.2.1	Simulation Block Diagram	50
4.2.2	Simulation Parameters	50
4.2.3	Results of the Simulation	51
4.2.4	Change in parameters	55
4.2.5	Practical Implementation	55
4.3	Discussion of Method One	56

4.3.1	Short-comings of Method One	58
4.4	Comparison with Existing Methods	58
4.4.1	Structure	58
4.4.2	Locking time	60
4.5	Summary	61
5	Novel Synchronisation Technique-Method Two	62
5.1	Improving Stability	62
5.2	System Description	63
5.3	Tesla Simulation	64
5.3.1	Simulation Diagram	64
5.3.2	Simulation Parameters	64
5.3.3	Results	66
5.3.4	Practical Implementation	71
5.4	Discussion of Method Two	71
5.4.1	Comparison Between Method One and Method Two	71
5.5	Comparison with Existing Methods	72
5.5.1	Evaluation of Code Phase-Frequency Grid	72
5.5.2	Structure	72
5.5.3	Doppler	73
5.5.4	Locking time	73
5.6	Summary	74
6	Theoretical Analysis of Method Two (1)	75
6.1	Non-linear Analysis	76

6.2	Stage One: Phase-Locked Loops	78
6.2.1	Phase-Locked Loop Fundamentals	78
6.2.2	Phase-Plane Analysis of Second Order Phase-locked Loop	81
6.2.2.1	Phase-Plane Plot	83
6.2.2.2	Examples of Phase-Plane Plots	83
6.3	Stage 2: Clock Recovery PLL analysis	85
6.3.1	Mathematical Description of System	87
6.3.2	Phase-Plane analysis of Clock Recovery Implementation	91
6.3.3	Design Using Phase-Plane Plots	95
6.4	Summary	96
7	Theoretical Analysis of Method Two (2)	98
7.1	Mathematical Model	98
7.2	Phase-Plane Analysis	106
7.2.1	The effect of Various Code-Phase Offsets	108
7.2.2	Varying the Loop Filter Constant (a)	111
7.2.3	Varying the Loop Gain (AK)	114
7.2.3.1	Larger Code-Phase Offsets	116
7.2.3.2	Effect of Frequency Offset	117
7.2.3.3	Effect of Q on the System	119
7.3	Summary	121
8	Data Demodulation, Carrier Recovery and Noise	122
8.1	The Influence of Noise on Spread Spectrum Synchronisation	122
8.1.1	Factors Affecting Noise Performance	122

8.1.1.1	The System is coherent	123
8.1.1.2	Detection of Received Spreading Wave-form	123
8.1.1.3	Differentiation	123
8.1.2	Simulation under Noisy Conditions	124
8.1.2.1	Simulation Diagram	124
8.1.3	Simulation Parameters: Case One	125
8.1.3.1	Signal-to-Noise Ratio of -12 dB	125
8.1.3.2	Signal-to-Noise Ratio of -16.1 dB	127
8.1.3.3	Signal-to-Noise Ratio of -18 dB	129
8.1.4	Simulation Parameters: Case Two	130
8.1.4.1	Signal-to-Noise Ratio of -18 dB	130
8.1.5	Simulation Parameters: Case Three	132
8.1.5.1	Signal-to-Noise Ratio of -16.1 dB	133
8.1.5.2	Signal-to-Noise Ratio of -17.1 dB	134
8.1.5.3	Signal-to-Noise Ratio of -18 dB	134
8.1.5.4	Discussion	135
8.1.5.5	Comparison with RASE	137
8.2	Modulating the Spread Spectrum Signal	138
8.2.1	Loop Design	140
8.2.2	Block Diagrams	140
8.2.3	Simulation Parameters	142
8.2.3.1	Spread Spectrum Simulation Parameters	142
8.2.3.2	Simulation Parameters: Carrier Synchronisation Network	142
8.2.4	Noiseless Case	142

8.2.5	Effect of noise on Carrier Synchronisation	146
8.2.6	Synchronisation at an Average Signal-to-Noise ratio of -9 dB	146
8.2.7	Synchronisation at an Average Signal-to-Noise ratio of -15 dB	148
8.2.8	Discussion	148
8.3	The Influence of Data on Spread Spectrum Synchronisation	149
8.3.1	Spike Counting to Determine Data	150
8.3.1.1	Simulation Results	153
8.3.1.2	Discussion	154
8.4	Data Determination Under Noise	155
8.4.1	Noiseless Case	160
8.4.2	Signal-to-Noise Ratio of 0.3 dB	162
8.4.3	Signal-to-Noise Ratio of -13.6 dB	162
8.4.4	Signal-to-Noise Ratio of -16.1 dB	163
8.4.5	Discussion	164
8.5	Comparison with Existing Methods	164
8.6	Summary	165
9	Practical Implementation	166
9.1	Synchronous Oscillator	166
9.2	Wideband Low Noise Square-Wave Differentiator	168
9.2.1	Positive Edge Detection	168
9.2.2	Signed Edge Detection	169
9.2.3	Comparison: Practical	170
9.2.4	Comparison: Simulation	172

9.3	Practical Results: Novel Method One	174
9.4	Practical Results: Novel Method Two	176
9.4.1	Synchronisation without Data	176
9.4.2	With the Inclusion of Data	178
9.5	Summary	179
10	Conclusion	180
10.1	Recommendations	183
10.1.1	Longer Pseudonoise Sequences	183
10.1.2	Noise	183
10.1.3	Detection of PN sequence	183
10.1.4	Carrier Tracking	183
10.1.5	Applications and Improvements	184
A	Simulation of Spilker Delay-Lock Discriminator Error Signal	185
B	Derivation of Partial-Correlation Function	186
B.1	Derivation	186
B.2	Verification of Theoretical Formula	189
C	Baseband Full-Time Early-Late Tracking Loop	190
D	Derivation of Pseudo-noise Sequence Spectrum	192
E	Simulations of Despreading Edge Detected PN Sequences	195
E.1	No Code-Phase Offset	195
E.2	One Chip Code-Phase Offset	196

E.3	Fourier-Time Spectrum (All ones)	196
E.4	Fourier-Time Spectrum (varying ± 1)	197
F	Details of Tesla	199
	References	201

List of Tables

4.1	Simulation Parameters - Method 1	51
4.2	Comparison of Typical Building Blocks for Synchronisation Circuitry	59
5.1	Simulation Parameters - Method 2	65
5.2	Locking Time Results - Method 2	74
8.1	Simulation Parameters - Method 2 under Noise (1)	125
8.2	Simulation Parameters - Method 2 under Noise (2)	130
8.3	Simulation Parameters - Method 2 under Noise (3)	133
8.4	Locking times of RASE and the Novel Synchroniser	138
8.5	Simulation Parameters: Spread Spectrum Network	142
8.6	Simulation Parameters - Carrier Synchronisation Network	142
8.7	Simulation Parameters: Spread Spectrum Network	153
8.8	Logic Table for System Lock (1)	158
8.9	Logic Table for System Lock (2)	158
8.10	Logic Table for System Lock (1)	160
8.11	Simulation Parameters: Spread Spectrum Network	160

List of Figures

1.1	Route Map of the Dissertation.	4
2.1	Hierarchy of Synchronisation	5
2.2	DSSS Transmitter	7
2.3	DSSS Receiver	8
2.4	Auto-correlation of Code Sequence	13
2.5	Tracking Grid - code phase-frequency uncertainty region	15
2.6	Cell Synchroniser for a given phase and frequency offset	16
2.7	Fixed Integration Time Energy Detector	19
2.8	Multiple-Dwell Energy Detector	19
2.9	Sequential Detector	20
2.10	Synchroniser system for linear sweep of uncertainty region	21
2.11	Delay Lock Discriminator (see Spilker)	23
2.12	Error Signal	25
2.13	Base-Band Full-Time Early-Late Tracking Loop	26
2.14	DC-Component as a function of Δ and δ	28
2.15	Full-Time Early-Late Noncoherent Tracking Loop	29
2.16	DC-Component as a function of Δ and δ	30

2.17	Tau-Dither Early-Late Non-coherent Tracking Loop	31
2.18	RASE synchronisation with full-time early-late code tracking (See Ward)	34
2.19	Sequential Detector	34
3.1	One-Sided Fourier Spectrum of a Pseudo-Noise Sequence	40
3.2	A Positive Edge Detected PN Sequence	41
3.3	Block Diagram representation of the despreading process	43
3.4	Product of an edge detected and native pn sequence	44
3.5	Fourier-Time Spectrum (all ones)	45
3.6	Fourier-Time Spectrum (varying ± 1)	46
4.1	Novel Spread Spectrum Synchroniser	48
4.2	Tesla Simulation Model of Novel Method One	50
4.3	Spectrum out of the Bandpass Filter	51
4.4	Impulse Train Stimulates the Bandpass Filter	52
4.5	Comparison between Transmitted and Locally Generated PN Sequences	53
4.6	VCO Input Signal	54
4.7	Train of Impulses Stimulate Bandpass Filter	55
4.8	VCO Input Signal (simulation 1)	57
4.9	VCO Input Signal (simulation 2)	57
5.1	Novel Spread Spectrum Synchroniser	63
5.2	Simulation block diagram	65
5.3	The Product of the Signed Edge Detected Received PN Sequence and Locally generated PN sequence.	66
5.4	VCO Drive Signal Before Lock	67

5.5	Comparison between Transmitted and Receiver Generated Sequence.	67
5.6	The Product of the Signed Edge Detected Received PN Sequence and Locally generated PN sequence.	68
5.7	Comparison between Transmitted and Receiver Generated Sequence.	69
5.8	Three Dimensional Locking Surface	69
5.9	VCO Drive Signal	70
5.10	Bandpass Filter Input	71
6.1	Overall Receiver Model	76
6.2	Equilibria	78
6.3	Phase-Locked Loop	79
6.4	Phase-Locked Loop Equivalent Circuit	81
6.5	Synthesis Diagram	83
6.6	Phase-plane for $\frac{a}{AK} = 1$	84
6.7	Phase-plane for $\frac{a}{AK} = 1$	84
6.8	Phase-plane for $\frac{a}{AK} = 0.5$	85
6.9	Phase-plane for $\frac{a}{AK} = 0.25$	85
6.10	Clock Recovery Circuit	86
6.11	Time and Frequency response of output of BPF	86
6.12	Block Diagram Representation of the of Clock Recovery Circuit	90
6.13	Simulation Diagram for Clock Recovery Circuit	91
6.14	Phase-Plane Plot for Clock Recovery Implementation	92
6.15	Phase Error versus Bandpass Filter Q	93
6.16	Phase-Plane of Clock Recovery Implementation	94
6.17	Phase-Plane of Clock Recovery Implementation.	94

6.18	Phase-Plane of Clock Recovery Implementation	95
6.19	Phase-Plane of Clock Recovery Implementation	96
6.20	Phase-Plane of Clock Recovery Implementation	96
7.1	Novel Spread Spectrum Synchronisation Model (Stage 1)	99
7.2	Novel Spread Spectrum Synchronisation Model (Complete)	101
7.3	Limiting of a sinusoidal waveform	105
7.4	Equivalent Circuit of Synchronisation Network	106
7.5	Simulation Diagram of Synchronisation Network	107
7.6	Phase-Plane Plot of Novel Loop (Offset 5 chips)	109
7.7	Phase-Plane Plot of Novel Loop (Offset 7 chips)	110
7.8	Phase-Plane Plot of Novel Loop (Offset 8 chips)	111
7.9	Phase-Plane Plot of Novel Loop ($a = 0.3$)	112
7.10	Phase-Plane Plot of Novel Loop ($a = 0.6$)	112
7.11	Phase-Plane Plot of Novel Loop ($a = 1.3$)	113
7.12	Phase-Plane Plot of Novel Loop ($a = 2.6$)	113
7.13	Phase-Plane Plot of Novel Loop ($AK = 2.5$)	115
7.14	Phase-Plane Plot of Novel Loop ($AK = 1.8$)	115
7.15	Phase-Plane Plot of Novel Loop ($AK = 3.14$)	116
7.16	Phase-Plane Plot of Novel Loop ($a = 1.3, AK = 3.14$)	117
7.17	Phase-Plane Plot of Novel Loop ($a = 1.3, AK = 2.5$)	117
7.18	Phase-Plane Plot of Novel Loop: Frequency Offset	118
7.19	Phase-Plane Plot of Novel Loop: Frequency Offset.	119
7.20	Phase-Plane Plot of Novel Loop: $Q=50$	120

7.21	Phase-Plane Plot of Novel Loop: $Q=20$	120
7.22	Phase-Plane Plot of Novel Loop: $Q=10$	121
8.1	Simulation Model to Investigate the Effect of Noise	124
8.2	Transmitted Sequence, Received Sequence and Receiver Sequence	126
8.3	Correlator Output	126
8.4	Fourier Spectrum at the Output of the Bandpass Filter (Node 4)	127
8.5	Output of bandpass filter (Node 4)	127
8.6	Transmitted Sequence, Received Sequence and Receiver Sequence	128
8.7	Correlator Output	128
8.8	Fourier Spectrum at the Output of the Bandpass Filter (Node 4)	129
8.9	Output of bandpass filter (Node 4)	129
8.10	Transmitted Sequence and Locally Generated Sequence	131
8.11	Correlator Output	131
8.12	Transmitted Sequence and Receiver Sequence	132
8.13	Transmitted Sequence and Receiver Sequence	132
8.14	Correlator Output	133
8.15	Correlator Output	134
8.16	Correlator Output	134
8.17	Errors due to noise: -7 dB	135
8.18	Errors due to noise: -12 dB	136
8.19	Errors due to noise: -16.1 dB	136
8.20	Errors due to noise: -18 dB	136
8.21	Transmitter	140

8.22	Partial Receiver	141
8.23	Spectrum of Transmitted Signal	143
8.24	Phase-locked Loop Error Signal (<i>lock</i> \5)	143
8.25	Spectrum of Carrier and Locally Generated Reference	144
8.26	Spectrum of Demodulated Signal	144
8.27	Spectrum of Filtered Demodulated Signal	145
8.28	Input to Bandpass Filter	145
8.29	Comparison between Transmitted and Receiver PN Sequences	146
8.30	Spectrum of Transmitted Signal	146
8.31	Tx Carrier versus Locally Generated Reference	147
8.32	Spectrum of Demodulated Signal.	147
8.33	Spectrum of Transmitted Signal.	148
8.34	Transmitter Carrier and Locally generated reference.	148
8.35	Receiver with data detection	150
8.36	Circuit diagram of the Detection Circuitry.	151
8.37	Logic table for the detection circuitry.	152
8.38	Detected Data verses Transmitted Data.	153
8.39	Spikes Before Polarity Correction	154
8.40	Spikes After Polarity Correction	154
8.41	Transmitter	155
8.42	Frequency Divider	155
8.43	Partial Receiver	156
8.44	Lock Detect Circuitry	159
8.45	Detected Data verses Transmitted Data	161

8.46 Control Signal verses data signal	161
8.47 Rectified Spikes.	161
8.48 Detected Data verses Transmitted Data (0.3 dB)	162
8.49 Detected Data verses Transmitted Data (-13.6 dB)	162
8.50 Detected Data verses Transmitted Data (-16.1 dB)	163
8.51 Transmitted and Locally Generated PN Sequences	163
9.1 Circuit diagram of the synchronous oscillator.	167
9.2 Active Differentiator	168
9.3 Frequency Response of Differentiator	168
9.4 Digital Edge Detector	169
9.5 Signed Digital Differentiator	170
9.6 Output of Signed Edge Detector	170
9.7 1 MHz input square wave	171
9.8 Output of Analog differentiator	171
9.9 1 MHz input square wave	172
9.10 Simulation model	172
9.11 Input waveform	173
9.12 Filtered waveform	173
9.13 Ideal Differentiator	173
9.14 Signed Digital Differentiator	174
9.15 Comparison of Received and Locally Generated Sequences	175
9.16 Product of the Edge Detected and Received Sequences	175
9.17 15 Bit Maximal PN Sequence	176

9.18 32 Bit Maximal Length Sequence	177
9.19 63 Bit Maximal Length Sequence	177
9.20 Transmitted and Locally Generated PN Sequences	178
9.21 Transmitted and Locally Generated PN Sequences	178
10.1 Dissertation Breakdown	182

List of Symbols

ϕ	phase offset.
B_L	noise bandwidth.
BPF	Bandpass filter.
CDMA	Code Division Multiple Access.
$E\{x\}$	expectation of the parameter x .
η	damping.
f_b	data clocking frequency.
f_c	chipping frequency.
f_d	difference frequency.
f_o	carrier frequency.
$Frac\{x\}$	fractional part of x .
FDMA	Frequency Division Multiple Access.
$Int\{x\}$	integer part of x .
L	number of chips in one period of a maximal length sequence.
LPF	Low pass filter.
$n(t)$	additive gaussian noise in channel.
N	number of registers required for the sequence generator.
osc.	oscillator.
$p(t)$	spreading waveform.
P.D.	power divider.
p_{fa}	probability of false alarm.
p_{fd}	probability of false dismissal.
PN	pseudo noise.
$\Re\{x\}$	denotes the real part of parameter x .
Rx	Receiver.
$Sa\{x\}$	is equivalent to $\frac{\sin(x)}{x}$.

T_c	bit (chip) period of a maximal length sequence.
T_e	examination period.
Tesla	Transient Electronics System Level Analysis by TESOF ^T .
TDMA	Time Division Multiple Access.
Tx	transmitter.
\tilde{T}_d	estimate of propagation delay.
ω_o	carrier frequency.
ω_n	natural frequency.
$x(t) \otimes y(t)$	convolution between $x(t)$ and $y(t)$.

Chapter 1

Introduction

Spread Spectrum systems are found in many flavours, used in many applications and have existed since the early days of communications. The concept of spread spectrum was developed in the early twenties, but the technique of spreading the information signal over a wide bandwidth only became popular during the war years. The technology was hidden from the commercial, and indeed the academic world, until after the Second World War. In the early sixties a flurry of papers was published, but by the early seventies the excitement had faded. Systems that were developed were primarily for the military. It was not until the late seventies that the interest in spread spectrum communications re-emerged, especially in the field of code-division multiple access. With new high-density micro-electronic processes, code-division multiple access systems have become the reality of the late eighties to the early nineties.

In the last few years code-division multiple access has been proposed as a viable alternative to traditional TDMA and FDMA for wireless communication systems [1][2][3]. Uses of Spread Spectrum are, however not limited only to multiple access. Other applications of spread spectrum communications systems include [4]:

- Selective addressing.
- Low-density power spectra for signal hiding.
- Message screening.
- High-resolution ranging.
- Intentional interference rejection.

- Multi-path interference rejection.

In rural areas, especially in areas of Africa, India and Brazil, no or very little telecommunication facilities are available [5] [6][7][8]. Many of these rural areas are in mountainous regions which commonly suffer from multipath. There are a number of multiple access radio-based alternatives to meet the needs of these people, like time-division-multiple-access (TDMA), frequency-division-multiple-access (FDMA) and code-division-multiple access (CDMA). Of the techniques, spread spectrum offers more immunity than do the others [9]. Viterbi, in [3], states that neither FDMA or TDMA offer the same interference rejection that is achieved using CDMA and spread spectrum and that spread spectrum based solutions are the logical choice for personal, mobile and wireless digital access.

The benefits of spread spectrum do however place restrictions on the design, and therefore make the implementation often expensive and complex. When using spread spectrum to provide a basic communications infrastructure, many factors need to be considered. This includes supplying the appropriate technology at the right cost. To achieve this, a trade-off with performance is required. One of the fundamental aspects of using spread spectrum is that an extra layer of synchronisation is required.

One of the more difficult aspects of Spread Spectrum design is the synchronisation of the spreading waveform. The primary characteristic of pseudonoise sequence synchronisation is the need for two levels of synchronisation namely acquisition (course synchronisation) and tracking (fine synchronisation) [10]. In these networks (the term network is used to describe a circuit, system or structure through the thesis.) a decision is required to switch between the two synchronisation modes. The two layer structure of the typical pseudonoise sequence synchronisation network can increase the overall cost of spread spectrum systems. The objective of the research was therefore to find solutions to reduce the overall cost and complexity of the synchronisation network. The synchronisation structure should *perform acquisition and tracking in a single structure*, and thereby be *low cost*.

The dissertation **concentrates on the development of this novel synchronisation network**. The novelty of the network is that it uses the information generated from both the received and locally generated pseudorandom noise sequence to achieve lock. Synchronisation and tracking is achieved using **one network**, and the decision as to whether the system is in lock only occurs once the two sequences are aligned. No intermediate decision is required on the state of synchronisation. This feature of the network makes it cost effective and easy to implement. The drawback of having simplified the overall structure of

the synchronisation network is that the system requires a higher signal-to-noise ratio. This is a characteristic of system requiring the detection of the incoming pseudo-noise sequence. One of the interesting aspects of the network is that the code-phase frequency grid is evaluated randomly as opposed to the deterministic evaluation by standard initial acquisition networks. Figure 1.1 is a route map through the dissertation.

Chapter 2 introduces some of the basic features of spread spectrum systems to aid in the discussion and analysis of synchronisation and tracking networks. An in depth discussion of the more popular methods of synchronisation and tracking, as found in the literature, are presented. Where necessary some of the features of these networks have been highlighted.

In chapter 3 there is a discussion of the concept of time-variant spectral analysis. It also extends the analysis to the signed edge detection of pseudonoise sequences and how one can derive conditions to develop a system that transverses the frequency-code phase grid in a random fashion. Using the concepts presented in chapter 3, the first novel synchronisation network is discussed and analysed using simulations. This is discussed in chapter 4. The novel network is compared to the methods discussed in chapter 2. Some basic synchronisation time estimates are given. A number of drawbacks of the system are highlighted.

Given the drawbacks of the first implementation, a more robust network is presented in chapter 5. The network is illustrated using simulation. The system is compared to the synchronisation methods discussed in chapter 2. The system is also compared to the first implementation presented in chapter 3. The system is then analysed using phase-plane techniques in chapter 6.

In chapters 6 and 7 the concept of phase-plane analysis is presented. The network is analysed in three stages. The first stage is a phase-locked loop using phase-plane techniques. This is extended to a clock-recovery network and then to the overall novel synchronisation network in chapter 7.

A further description of the novel synchroniser to include data is presented in chapter 8. Circuitry is developed to overcome the effect of data on the novel synchronisation network. The novel synchronisation network is also analysed under noise. The issue of carrier recovery is also discussed.

Some of the practical issues of the implementation are discussed in Chapter 9. The synchronous oscillator is briefly discussed and a novel wideband signed edge detector (square wave differentiator) is presented. Both these networks can be an integral part of a low cost synchronisation network. In chapter 9 some practical results of the two synchronisation

networks are given.

Chapter 10 concludes the thesis. A discussion of issues related to using longer sequences is included. A brief discussion on how the novel network presented in this dissertation can be adapted to deal with longer sequences.

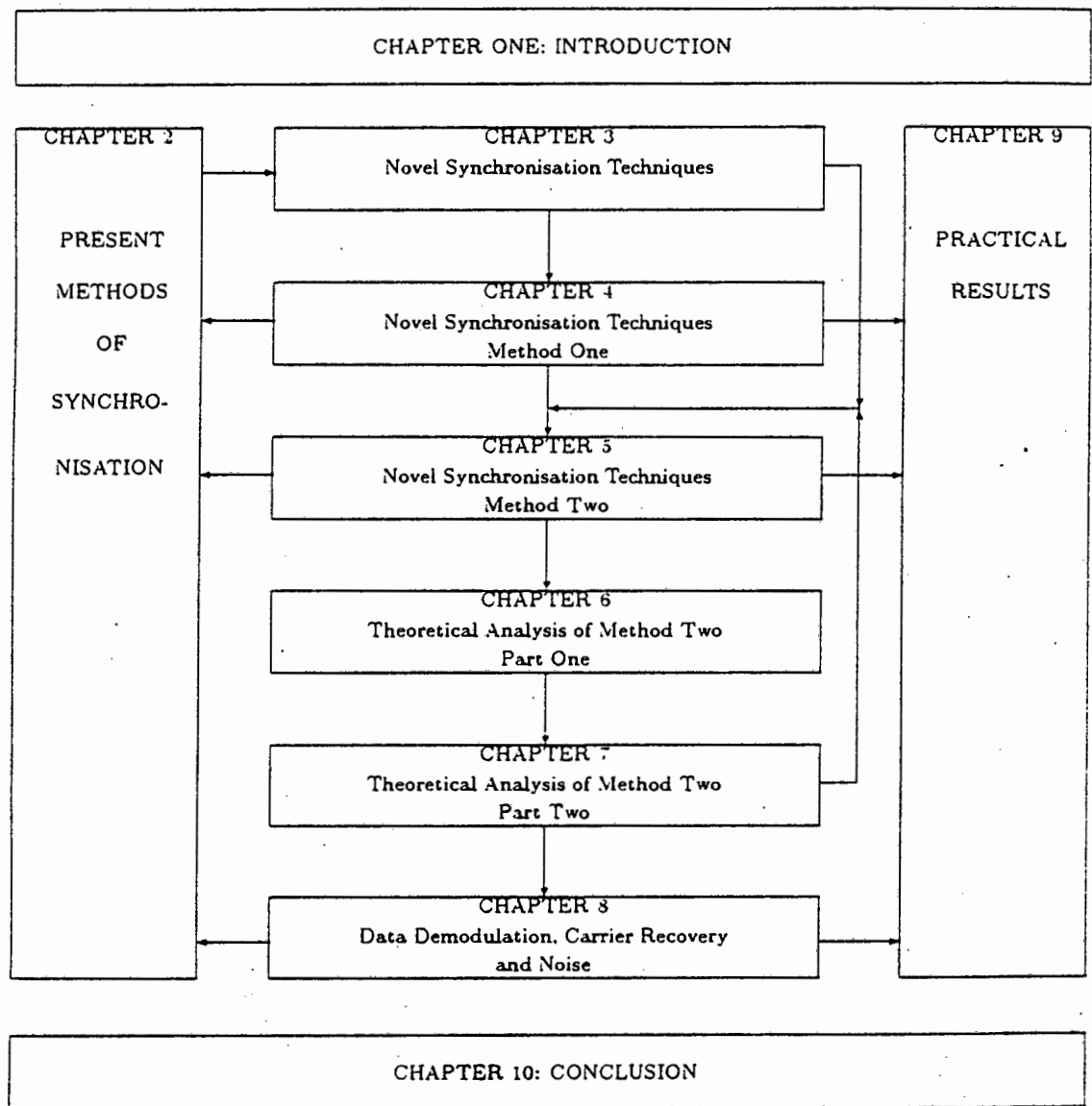
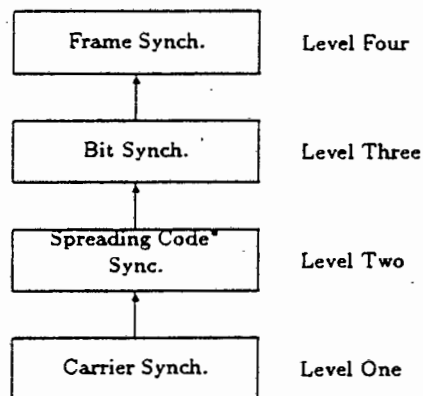


Figure 1.1: Route Map of the Dissertation.

Chapter 2

Present Methods of Synchronisation

Synchronisation is an important part of coherent digital communication systems. In these systems various levels of synchronisation are required depending on the specific implementation. The hierarchy of synchronisation is depicted in figure 2.1. Only levels one and two will be considered in this dissertation. Of the four levels the most complicated is level two; that is the synchronisation of the spreading sequence.



* Specific to Spread Spectrum Systems

Figure 2.1: Hierarchy of Synchronisation

Before discussing methods of spreading code synchronisation, a brief overview of spread spectrum systems will be given. (In the hierarchy given in figure 2.1, it does occur that

level one and two are swapped around. A typical example of this is in non-coherent spread spectrum tracking techniques which are discussed later in this chapter.)

2.1 What is Spread Spectrum?

The origins of Spread Spectrum are relatively obscure. Even today it is still unclear who can be considered to be the father of this useful communications technique. By all accounts, the first suggestion of spread spectrum was made in the early 1920's. From there it developed rapidly to become the heart of military communication systems [11] [12] [13]. Spread Spectrum technology was hidden from commercial uses until the mid-1950's. It is only in recent years that a surge in spread spectrum research has taken place, but, what is Spread Spectrum?

In a nutshell, spread spectrum takes a band limited signal and transmits it as a wide-band signal. This is an antithesis to modern day systems, which are designed to conserve bandwidth, while maximizing information throughput. Spread Spectrum, on the other hand, does not conserve bandwidth, but rather uses additional bandwidth over the original information signal bandwidth. A number of other modulation schemes, e.g. FM with deviation ratios greater than one also use wider bandwidths, but are not classified as Spread Spectrum systems. Therefore, to be classified as a Spread Spectrum system the following two conditions must hold true, namely:-

- The transmitted energy must occupy a bandwidth which is larger than the information bit rate and which is independent of the information bit rate.
- Demodulation must be accomplished, in part, by correlating the received signal with a replica of the signal used in the transmitter to spread the information signal.

Spread Spectrum systems can be classified into three primary groups, namely

1. Direct Sequence Spread Spectrum.

In Direct Sequence Spread Spectrum the data and pseudo-noise sequence are mixed and the resulting wide-band signal is used to modulate an RF carrier for transmission.

2. Frequency-hopped Spread Spectrum.

In Frequency-hopped Spread Spectrum, the carrier frequency remains at a given frequency for a duration called the dwell time and then hops to a new frequency in the spreading bandwidth. Frequency selection is based on a pseudo-random selection from a list of available channels.

3. Time-hopped Spread Spectrum.

In Time-hopped Spread Spectrum the carrier is keyed on and off by a PN generator. As in conventional CW transmissions, the bandwidth of a time-hopped signal is a function of the keying rate.

Each of these methods have their advantages and disadvantages [4]. In this dissertation, however, the synchronisation systems presented are for direct sequence spread spectrum (DSSS), and therefore any further reference to a spread spectrum system will imply direct sequence spread spectrum.

2.1.1 Direct Sequence Spread Spectrum

Direct Sequence Spread Spectrum (DSSS), or directly modulated carrier-modulated code sequence modulation, is the most common form of spread spectrum today. This is because of the relatively simple generation of such a signal. DSSS is used in many applications.

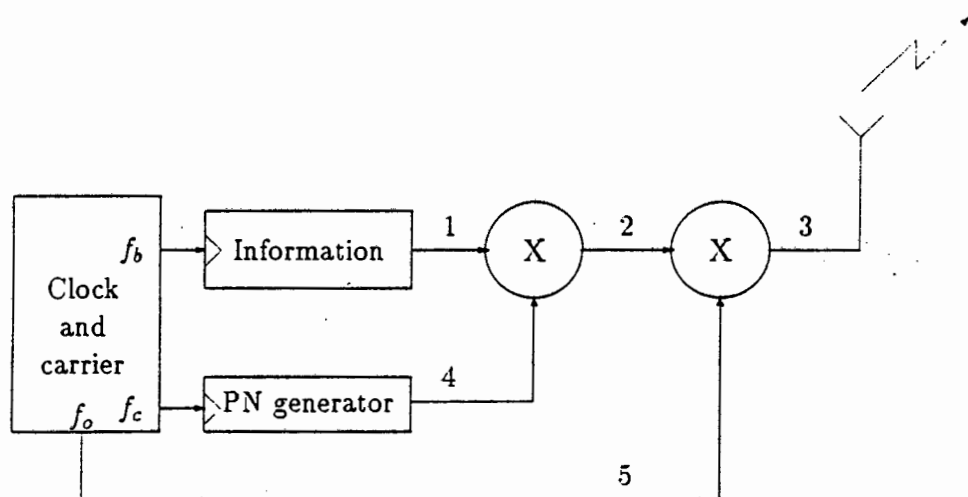


Figure 2.2: Idealized Direct Sequence Spread Spectrum Transmitter with filters omitted.

The transmitter, as shown in figure 2.2, is relatively simple. An information signal (1) with bit-rate f_b is mixed with a spreading sequence (4) with a chipping frequency f_c . This signal (2) is used to modulate a carrier (5) with frequency f_o . The overall signal (3) will have a bandwidth determined by the spreading sequence. In practical systems the bandwidth of the transmitted waveform is limited to $2f_c$. The carrier frequency should be chosen such that no spectral fold-over occurs, and the chipping frequency should be chosen such that no self-interference takes place. [9]

The receiver on the other hand is not so simple. Figure 2.3 shows a typical receiver structure. The received signal (6) is fed to a mixer, carrier recovery circuitry and the code acquisition and tracking circuitry. In the implementation given in figure 2.3, the carrier must be recovered before code acquisition can take place. The demodulated spread spectrum waveform (7) is despread by the locally generated PN sequence (9) and the resultant waveform (8) is fed to the data demodulator. In some configurations some modules are combined to improve performance and reduce the complexity of the circuitry. In practical systems the input to the receiver is filtered to a bandwidth of $2f_c$ and after the signal is demodulated to baseband, the signal is filtered in a bandwidth of f_c .

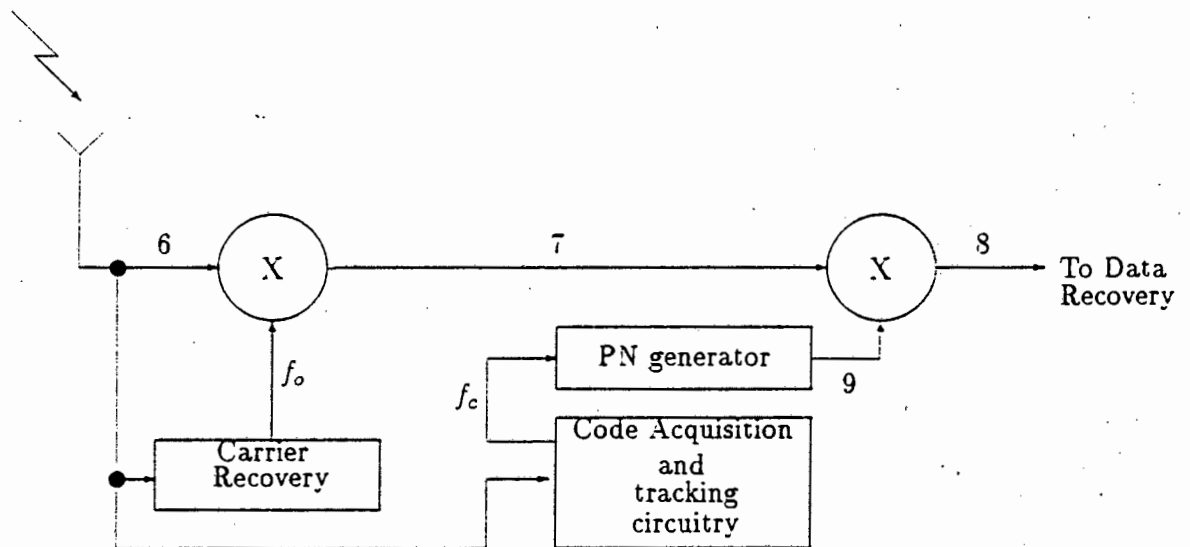


Figure 2.3: DSSS Receiver

The heart of the spread spectrum system is the pseudo-noise sequence which determines the various properties of the system, e.g. transmission bandwidth, message screening, processing gain (the ability of a spread spectrum to enhance the wanted signal, while suppressing the

effects of the unwanted signals) and so forth [13][10]. Various texts [14][15][16][17] (and others) have covered the properties and generation of pseudo-noise sequences in great detail. Some of the important properties of pseudo-noise sequences¹ are:

Property One: A maximal-length sequence (m-sequence) contains one more '1' than '0'. The number of '1's in the sequence are $\frac{1}{2}(L + 1)$.

Property Two: The modulo-2 sum of an m-sequence, and any phase-shift of the same sequence, is another phase of the same m-sequence.

Property Three: If a window of width r is slid along the sequence for L shifts, each r -tuple except the all zero r -tuple, will appear exactly once.

Property Four: The periodic auto-correlation function is two-valued when correlating over a complete period.

Property Five: The statistical distribution of ones and zeros is well defined and always the same. Relative positions of the runs vary from code sequence to code sequence, but the *number* of each run length does not.

These properties each have a purpose, especially in communications or ranging systems. The two properties which are of particular interest are the the first and the forth. Since the number of ones and zeros differ, a DC component will exist. This affects the degree of carrier suppression achieved. However the longer the sequence the greater the degree of suppression. In most cases it can be assumed that the DC value is zero.

The auto-correlation property is by far the most important property of the pseudo-noise sequence. Only when a sequence and a replica of itself are shifted by less than one chip will a positive correlation value exist. For all other shifts, a small negative correlation value will exist. Therefore the correlation value gives a clear indication of synchronisation and is used to synchronise spread spectrum systems.

2.1.2 Code Division Multiple Access (CDMA)

In the last few years much has been published in the area of CDMA. (see Appendix 2 of [10] for a detailed list of references.) Although CDMA is not considered in the thesis, its importance in multiple access environments warrants mention. What is the basis of CDMA?

¹The term 'pseudo-noise sequence' implies maximal length pseudo-noise sequences, unless otherwise stated.

2.1.2.1 What is the basis of CDMA?

The basis of CDMA is the spreading codes themselves. The spreading codes are used to separate one spread spectrum signal from another. This places specific requirements on the selection of the codes. A minimum requirement is that the set of codes employed in a given system should be near-orthogonal. Near-orthogonal simply means that all the codes used must have a low enough mutual cross-correlation that they do not significantly interfere with one another over the dynamic range of the signals presented to any receiver in a CDMA network. Since the number of codes in any given set of maximal length sequences is limited, extensive use is made of Gold codes, Kasami codes and others [15] [14][16].

2.1.2.2 How does the receiver synchronise?

In the CDMA environment a given synchronisation subsystem is faced with one or more spread spectrum signals. For the purpose of discussion, it is assumed that each synchronisation sub-system only knows one spreading code. It is also assumed that the known signal (i.e. spread with a replica of the code known by the receiver) has a higher signal-to-noise ratio than the interfering spread spectrum signal/s.

The negative effect on the receiver is related to the cross-correlation between the two or more spreading codes. The cross-correlation between two maximal length sequences results in a cross-correlation value of $|\frac{1}{N}|$, (assuming that one correlates over a complete period of the code), where N is the length of the sequences.) Therefore the degradation is directly proportional to the cross-correlation value. The level of interference increases for every user that is added to the system.

2.1.2.3 Why use CDMA?

Viterbi identifies four primary forms of interference that are found in personal, wireless and cellular communications, namely: multiple-user access, multiple cell-access, multipath and multiple media. He states that through the use of spread spectrum all four effects can be mitigated effectively, and in some cases (like multipath) can be used to improve communications performance. In spread spectrum the multiple paths can be isolated and through diversity combining (e.g. the Rake Receiver) can be used to an advantage.

Two of the key features of CDMA systems are:

- Overcomes the various forms of interference.
- Universal frequency reuse avoids the complicated issue of frequency-management planning when additional cells are introduced.

CDMA, although not in wide use at present, will become widely implemented in the future. Already it has been cited as the likely choice for personal as well as mobile communication services. [18] [19]

2.1.3 Processing Gain and Jamming Margin

All receivers, including broadcast AM, FM and TV receives have jamming margins. Jamming margin is the ability of a receiver to provide a usable receiver output when an undesired signal is present. Typically the input to a receiver consists of the desired signal, noise and interference. The output is demodulated (and/or despread) and at the output the desired component and the undesired noise and interference. The required signal-to-noise/interference ratio depends on the application. In voice systems, the typical value is 10 dB or more and in data systems signal-to-noise ratios of 15 – 18 dB is often specified [4]. (Error rates are inversely proportional to signal-to-noise ratio.) The signal-to-noise ratio (in dB) at the output of a receiver is therefore,

$$\left[\frac{S}{N} \right]_{out} = \left[\frac{S}{N} \right]_{in} + \text{processing gain} \quad (2.1)$$

In direct sequence spread spectrum systems the processing gain is defined as the transmitted bandwidth (BW_{RF}) over the data bandwidth (R). Therefore if the transmitted bandwidth is 100 kHz and the data bandwidth is 1 kHz, the maximum processing gain expected is 20 dB. The jamming margin of a direct sequence spread spectrum system is therefore:

$$\text{Jamming Margin} \leq \text{Processing Gain} - \left[\frac{S}{N} \right]_{out} \quad (2.2)$$

2.2 Spread Spectrum Synchronisation

Synchronisation of Direct Sequence Spread Spectrum is by far the most complicated process in the overall spread spectrum system. In analysis of spread spectrum systems, it is often

assumed that perfect synchronisation between transmitter and receiver exists. By 'perfect' synchronisation it is implied that the coded signal arriving at the receiver is accurately timed in both its code pattern position and the rate of chip generation with respect to the receiver generated local reference code. This assumption detracts from this complicated process. Dixon [10] states it rather elegantly,

"What an assumption! More time, effort and money has been spent on developing and improving synchronisation techniques than in any other area of spread spectrum systems. There is no reason to suspect that this will not continue to be true in the future."

2.2.1 Sources of Uncertainty

Two primary regions of uncertainty exist in Spread Spectrum systems. These uncertainties are namely code-phase and chipping frequency. The reason for this uncertainty is related to the typical period lengths of the spreading waveform, and the large bandwidths due to the spreading process. Therefore, the uncertainty in the estimated propagation delay \bar{T}_d translates into a number of symbols of code-phase uncertainty. Oscillator and Doppler effects result in frequency uncertainties and these are also required to be resolved. It is essential in all systems to achieve the correct code-phase and frequency in the minimum acceptable time. In many cases this is required in low signal-to-noise ratio situations, or in the presence of jamming.

2.2.1.1 Code Phase Uncertainty

To despread a spread spectrum signal, the code phase uncertainty between the local generated spreading sequence and the received spreading sequence must be resolved to be within one chip. A timing error of one chip is acceptable for rough synchronisation, but when in the tracking stage a smaller error is required. The indicator whether the code phase uncertainty has been resolved to be within one chip is the correlation function over one or more periods of the spreading sequence. The reason for this is clearly seen by examining the auto-correlation function of a maximal length pseudo-noise sequence, as shown in figure 2.4. When the sequences are T_c seconds apart, the correlation value is zero, and between $T_c < t < (L - 1)T_c$ the correlation is $-\frac{1}{L}$. The only time it can be stated that the two sequences are synchronised is for a positive correlation value (assuming no data is present).

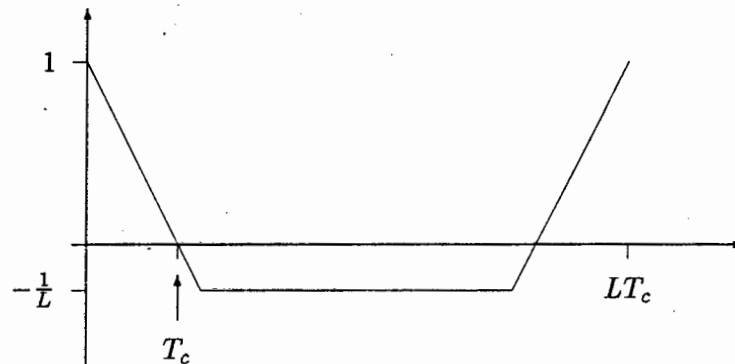


Figure 2.4: Auto-correlation of Code Sequence

2.2.1.2 Frequency Uncertainty

A number of factors contribute to frequency uncertainty. The largest contributing factor is that of Doppler shift. Doppler-related frequency errors cannot always be predicated and will affect both the carrier frequency and code-rate. In mobile environments, mobile transmitters and/or receivers see a relative code-phase change with every change in their relative positions. The amount of Doppler-frequency uncertainty applied to a received signal is a function of the relative velocity of the transmitter and receiver and of the transmitted frequency. [4] Other factors that affect frequency are oscillator instabilities, noise, natural component degradation and so forth. All these factors complicate synchronisation.

2.2.2 Two Pronged Approach to Synchronisation

Spread Spectrum synchronisation is a two pronged attack to resolve both code-phase and frequency uncertainty. The first prong is initial synchronisation (acquisition), while the second prong is code tracking. Of the two phases the first is the more difficult. In the next two sections the issue of code acquisition and code tracking will be addressed.

2.3 Code Acquisition

The issue of Code Acquisition, or initial synchronisation, has been dealt with in many texts [10][20][21][22] [23] (and others). The initial acquisition problem is complex, since both the

frequency of the code sequence and the phase position of the sequence is unknown. The initial synchronisation process can be divided into three groups based on the amount of prior knowledge. The groups are:

- Subsequent synchronisation
- Cold start synchronisation
- Prior knowledge synchronisation

Subsequent synchronisation

Subsequent synchronisation is based on timing knowledge gained from previous synchronisation. Although this can reduce the acquisition time, it does not change the basic hardware configuration. If anything, using such knowledge would possibly increase the cost, and this is not always justified. Since the emphasis in this dissertation is *low-cost*, subsequent synchronisation based on previous knowledge will not be discussed.

Prior knowledge synchronisation

Synchronisation based on prior knowledge is possible if the exact time of transmission is known to an accurate degree by both the transmitter and receiver stations. Through some algorithm the code-phase of both code generators could be set to coincide. The problem with such a method is that electrical path lengths may vary and the distance between transmitter and receiver will need to be known to some degree of accuracy, such that the code-phase offset could be calculated correctly.

Cold start synchronisation

Since both the previous groups require some form of prior knowledge, which is not always practical, cold start synchronisation is the reality that most designers are faced with. The methods of cold start synchronisation vary, with some methods more popular than others. Some of the more popular methods will be discussed briefly.

2.3.1 The Time-Optimal Synchroniser

A time-optimal synchroniser is one where all possible combinations of both frequency and code-phase are evaluated simultaneously. For the purpose of discussion it is assumed that both phase and frequency uncertainty are bounded to a range of ΔT seconds and $\Delta\Omega$ radians/second respectively. The synchronisation problem can therefore be represented in a grid structure as shown in figure 2.5.

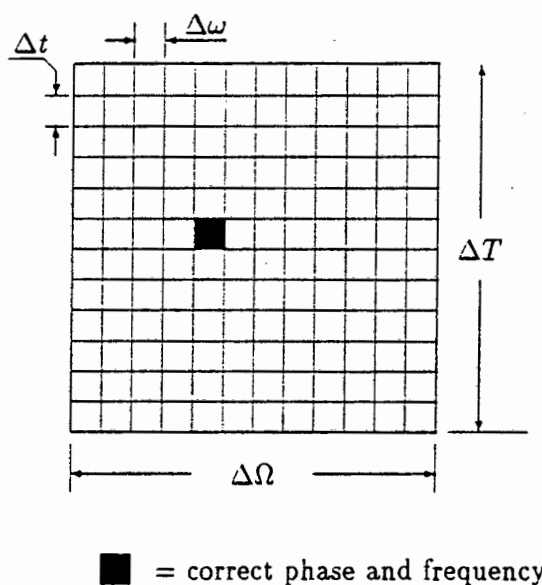


Figure 2.5: Tracking Grid - code phase-frequency uncertainty region

Each cell in the tracking grid represents a possible phase code offset and frequency offset. In the case of an $L = (2^N - 1)$ maximal length code, the number of rows would be greater than L . The number of divisions is dependent on the maximum allowable timing error for initial acquisition. Typical one would select a value of $2L$, thereby giving a timing error after acquisition of $\pm\frac{1}{2}$ a chip. For each of the cells in the grid a synchronisation circuit of the form shown in figure 2.6 would be required. From a hardware perspective this would be impractical and not cost effective.

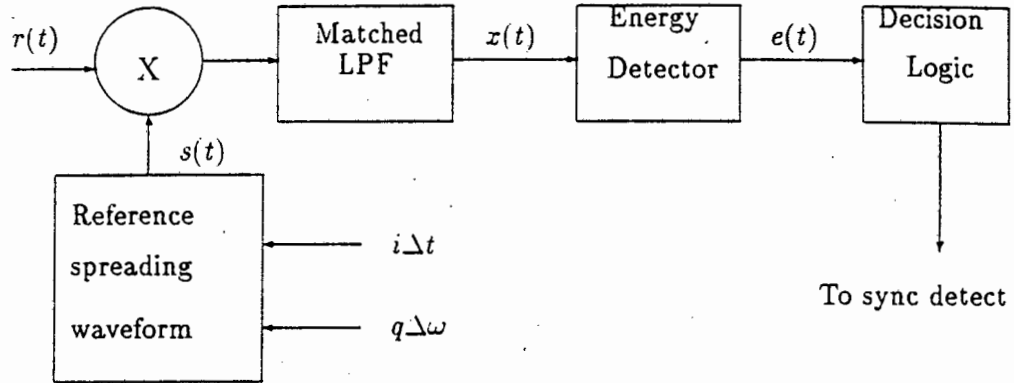


Figure 2.6: Cell Synchroniser for a given phase and frequency offset

The circuit in figure 2.6 works as follows. Suppose the received signal $r(t)$, is a carrier which is BPSK modulated by an m-sequence waveform $p(t)$. Therefore

$$r(t) = p(t - T_d) \Re \left\{ e^{j(\omega_0 t + \phi)} \right\} + n(t) \quad (2.3)$$

where T_d is the delay between the transmitter and receiver. The reference waveform is

$$s(t) = p(t - i\Delta t) \Re \left\{ e^{j q \Delta \omega t} \right\} \quad (2.4)$$

The two signals are multiplied together and lowpass filtered to remove higher order frequency terms, resulting in

$$x(t) = p(t - T_d) p(t - i\Delta t) \Re \left\{ e^{j((q\Delta\omega + \omega_0)t - \phi)} \right\} + n'(t) \quad (2.5)$$

The energy detector takes the expectation of the above equation. Assuming that the correct value of q was selected forcing the carrier term to 1, the output of the energy detector is given in 2.6. Equation 2.6 is the auto-correlation of $p(t)$. The energy detector is commonly implemented as an integrate and dump circuit. The output of the integrate and dump is squared to remove the effect of data. The integration time is selected based on a number of criteria, including the signal-to-noise ratio, and the length of the spreading sequence. The integration time should be much shorter than the data bit period.

$$e(t) = E \{p(t - T_d)p(t - i\Delta t)\} \quad (2.6)$$

If the correct i is selected, such that $i\Delta t \approx T_d$, $e(t)$ will be a maximum, implying that the correct cell was selected. In the case of the incorrect i , the correlation value will be small. The overall time-optimal synchroniser would compare the results from each cell synchroniser and select the maximum value.

The time-optimal synchroniser is considered impractical to implement in hardware because of the number of possible combinations of code-phase and frequency uncertainties. For every cell in the grid, a synchroniser is required. However, it does give insight into possible mechanisms that could achieve a similar result. Methods based on the time-optimal synchroniser are referred to as serial search techniques, since they evaluate the uncertainty region cell by cell. Before considering serial search techniques, an overview of energy detection in noise is given.

2.3.2 Detection of a Signal in Additive White Gaussian Noise

In the previous discussion, a mathematical model of an energy detector was used for illustration purposes. In spread spectrum systems, however, three main energy detection methods are used [23], namely:

- Fixed Integration Time Detection
- Multiple-Dwell Detection
- Sequential Detection

The acquisition time of the PN sequence is determined primarily by the type of detection used. Some of the system constants that determine the acquisition time are:

- Probability of detection (P_d)

This is the probability that the correct cell is detected when examined.

- Probability of false alarm (P_{fa})

This is the probability of declaring an incorrect cell to be the correct cell.

- Evaluation time for a cell (T_i)

This is the time specified to examine a cell.

- Reject time for an incorrect cell (T_{fa})

This is the time required to reject a cell when a false alarm occurs.

Each of the methods result in different relationships between P_d , P_{fa} , T_i and SNR. For each of the methods, this relationship also depends on factors like detection thresholds. Each of the methods will be discussed briefly. (For a detailed analytical analysis of the three methods, refer to Ziemer and Peterson [23].

2.3.2.1 Fixed Integration Time Energy Detection

This by far the simplest of the three methods. In figure 2.7 the input to the detector is taken from the output of the despreading mixer. If the reference wave-form and the received sequence are within one chip, the received wave-form will be despread and the signal plus noise will appear at the output of the bandpass filter. (Typically in spread spectrum systems acquisition and tracking take place at an intermediate frequency rather than at base-band.) The output of the filter is squared and then lowpass filtered to remove the double frequency terms. The output of the lowpass filter is integrated for T_i seconds and the output is compared to a predetermined threshold (V_T). The output of the comparator is high if the signal is present, or low if no signal was detected. The relationship between the parameters mentioned previously is therefore determined by T_i and V_T .

2.3.2.2 Multiple-Dwell Detection

The problem with the previous detector is that only two parameters can be varied to reduce synchronisation time. With signal-to-noise ratio and number of uncertainty cells fixed the selection of T_i and V_T completely determine P_d , P_{fa} and T_{fa} and therefore the average synchronisation time. Since there is only one correct cell within the uncertainty region, most of the cells evaluated by the energy detector are noise alone. This lead to the development of the multiple-dwell detector. This detector is designed to reduce the false alarm penalty, by rejecting cells rapidly. The first evaluation is very short and results in immediate rejection of many incorrect cells. When a false alarm occurs, the cell is evaluated

a second time over a longer integration time. During this second evaluation, most of the cells that caused a false alarm are rejected. The process can continue a third, fourth or as many as desired to achieve a particular performance level. Ziemer shows that the multiple-dwell detector results in a reduction in the overall synchronisation time. Figure 2.8 shows a block diagram of the multiple-dwell energy detector.

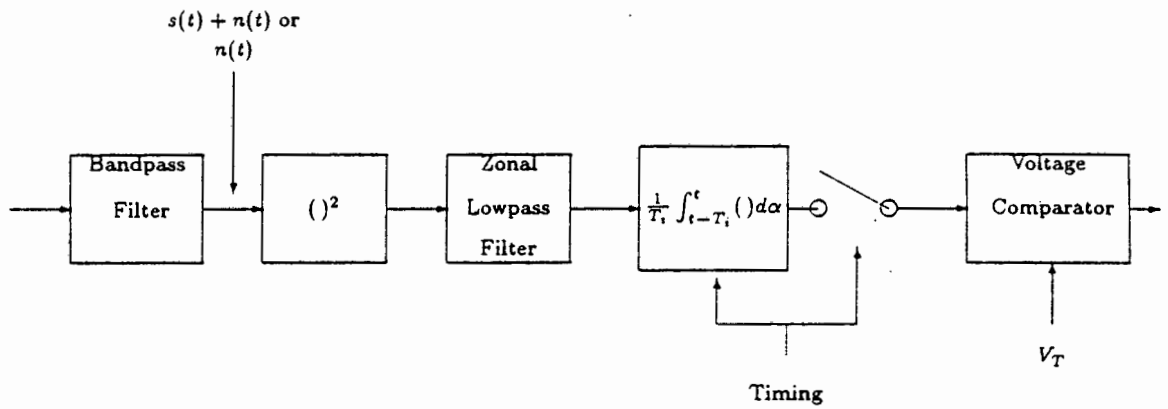


Figure 2.7: Fixed Integration Time Energy Detector

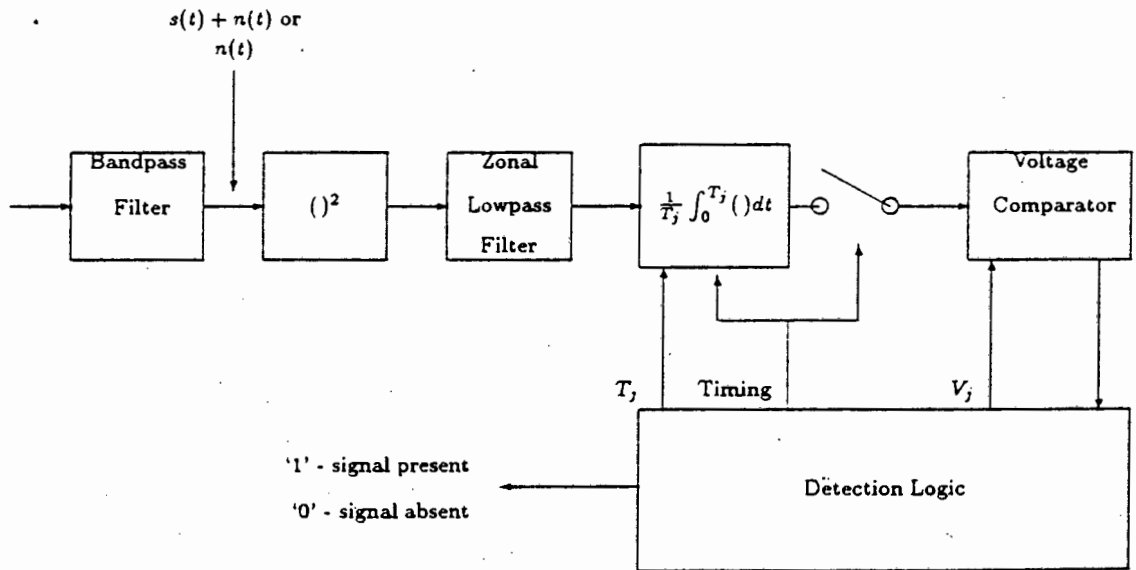


Figure 2.8: Multiple-Dwell Energy Detector

2.3.2.3 Sequential Detection

The difference between the sequential detector and the previous two detectors is that the sequential detector does not generate excessive false alarms. The output of the envelope detector is sampled many times during the evaluation of a single spreading waveform phase. The samples are feed one by one to a likelihood ratio calculator. For each sample, the likelihood ratio is calculated and compared to a lower (A) and upper (B) threshold value. The disadvantage of the sequential detector is the difficulty in implementing the likelihood ratio calculator, which is typically implemented using a micro-processor. Figure 2.9 gives a basic block diagram of the detector.

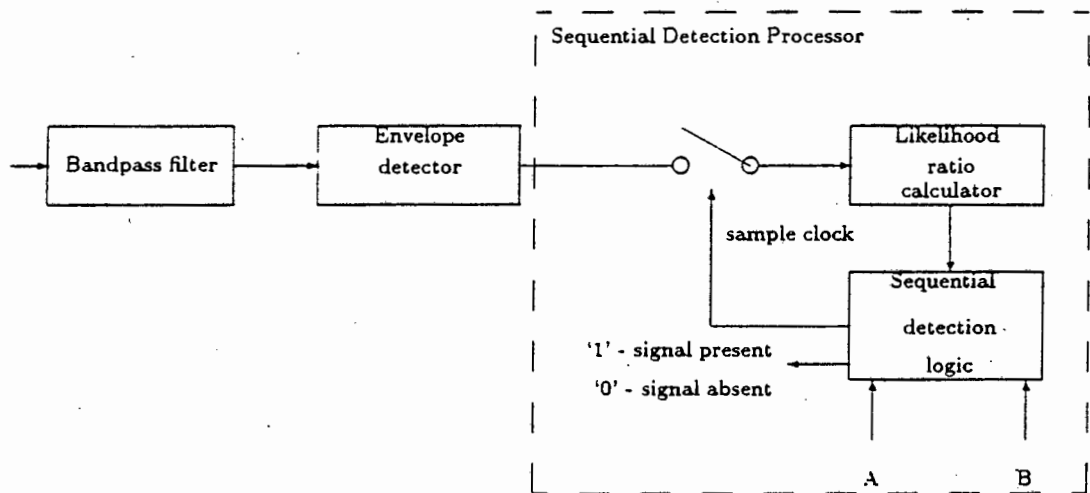


Figure 2.9: Sequential Detector

2.3.3 Serial Search Techniques

By far the most common synchronisation method is serial search techniques. These techniques evaluate each cell of the uniformly distributed uncertainty region sequentially until the correct cell is identified. Ziemer and Peterson [23] calculate the mean and variance of synchronisation time for generalized serial search methods based on a uniform distribution.

2.3.3.1 Non-Uniform Uncertainty Region

If the uncertainty region is not uniformly distributed, the search strategy should be modified to search the most likely cells first. For example, if the phase uncertainty region were Gaussian distributed, a triangular wave with an ever increasing magnitude at each peak and valley would be a reasonable search strategy [24]. Other search strategies have shown to be more effective, for example in [25]. Jovanovic proposed the Uniformly expanding alternate search strategy (UEA). The standard Z strategy is amplitude modulated with a low frequency sawtooth. In the same paper, Jovanovic also analyses the non-uniformly expanding alternate search strategy (NUEA). The NUEA is similar to the UEA, except that the sawtooths period and amplitude are not fixed.

Although discrete methods are more common, an alternative approach is continuous linear sweeps of the uncertainty region.

2.3.3.2 Continuous Linear Sweep

The basic synchronisation system for continuous linear sweeping of the uncertainty region is depicted in figure 2.10.

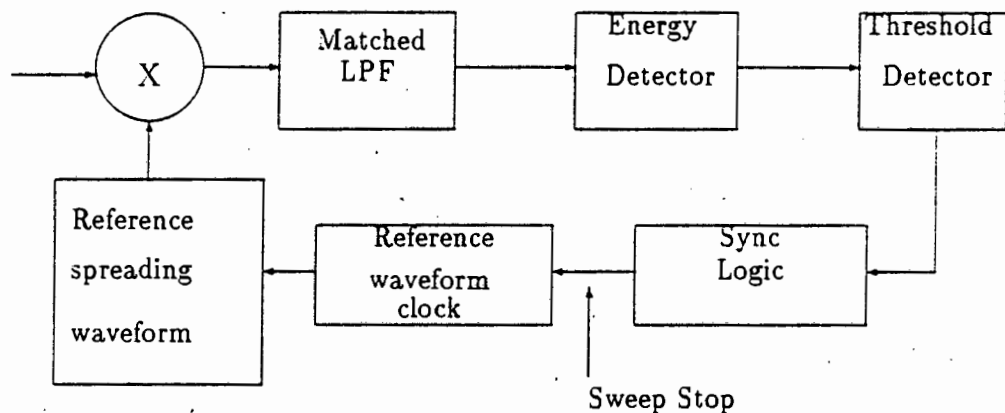


Figure 2.10: Synchroniser system for linear sweep of uncertainty region

The analysis of figure 2.10 follows the same approach as that of figure 2.6. The reference waveform's clock frequency is offset slightly so that the phase of the waveform slips linearly past the received waveform phase. When the received and reference phase have slipped

sufficiently close to each other, despreading will occur and the threshold detector value will be exceeded. The synchronisation logic will stop the sweep process and indicate that the system is in lock. Sage analysed this technique in [21].

2.3.4 Matched Filter Synchronisation

In the serial search techniques, the received pseudo noise sequence was correlated with a replica generated at the receiver. The output of this correlation process was then evaluated to determine if actual despreading had taken place. From filter theory it is well known that the output is simply a convolution between the filter impulse response and the filter input.

In the case of Spread Spectrum the filter impulse response is a time reversed and delayed replica of a segment of the spreading pseudonoise sequence. The matched filter continuously correlates the received segment with the local segment and generates a maximum output when a maximum value is achieved. At this point the local pseudo noise generator is activated with the specific starting phase and synchronisation will be accomplished.

Many methods can be used to implement the matched filter. Depending on the speed requirement surface acoustic wave (SAW) technology [26] is becoming popular. A two level SAW technique was presented at ISSSTA 1992 [27]. Other methods include CCD delay line technology [28] and DSP techniques for low speeds are required [29]. Modified matched filter techniques to cope with Doppler have been designed in recent years [30] using digital filters.

2.3.5 Parallel Acquisition of PN Sequences

The research for newer and faster methods is ongoing. Chwala investigated methods of acquisition using parallel techniques. In [31] he investigates four methods, namely optimal and maximum-likelihood estimation schemes, and hypothesis-testing and a locally optimal detection schemes. The essence of the schemes are that the grid structure in figure 2.6 is processed in parallel. He shows mathematically that the schemes are faster. He highlights the fact that the parallel structure implies an exponential increase in hardware.

2.4 Code Tracking

Once initial synchronisation has been achieved, i.e. the transmitted and locally generated receiver sequence code-phase are less than one chip out, code tracking loops are used to complete the synchronisation problem, that is of accurately locking and maintaining lock. Code tracking is accomplished using phase-locked loop techniques, except for the phase discriminator implementation. The primary difference between the discriminator for carrier tracking and code tracking is that the former is a simple multiplier, while the latter is a combination of multipliers, filters and envelope detectors.

Code tracking loops can be classified in several ways. First, loops can be either coherent or non-coherent. Coherent loops make use of the received carrier phase information, while non-coherent loops do not. All but one of the loops to be discussed make use of correlation operations between the received signal and two different phases of the locally generated code sequence.

Many texts [4][23][32] have covered code tracking, and only a brief discussion of the methods used will be presented.

2.4.1 Optimal Tracking

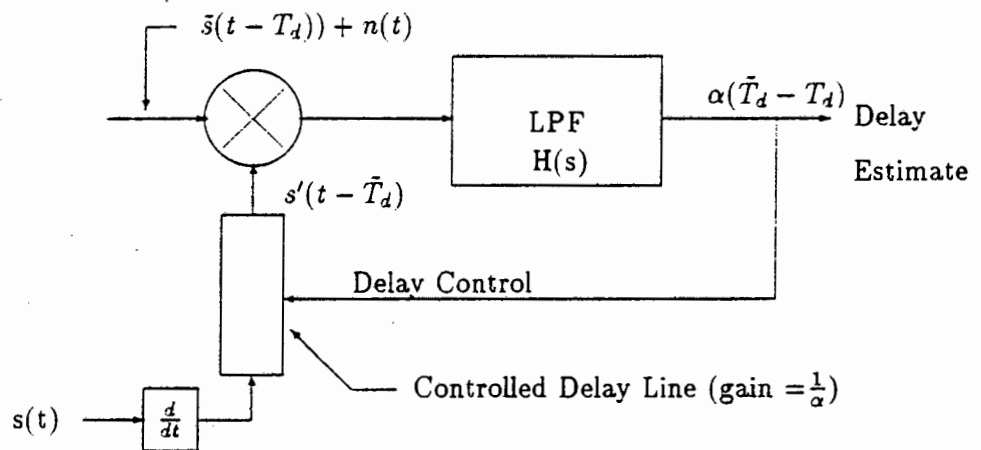


Figure 2.11: Delay Lock Discriminator (see Spilker)

The basis of code tracking stems from a paper by Spilker published in 1961 [33]. Spilker showed that the proper error signals to maintain lock could be derived by correlating the received signal with a locally generated time differentiated replica of the spreading sequence.

This original work serves as the basis of modern tracking loops. The tracking loop can be represented as in figure 2.11.

The analysis of Spilker's optimal discriminator provides insight into one of the novel methods analysed in this dissertation. The following analysis assumes that the input spreading sequence is unmodulated and that the channel is time-invariant. Given an input sequence $s(t - T_d)$ and a locally generated sequence $s(t)$, the dc component of the multiplier is given by the expectation (or time average) of the multiplied signals, i.e.

$$e(t) = \frac{1}{T} \int_0^T \bar{s}(t - T_d) s'(t - \bar{T}_d) dt \quad (2.7)$$

Equation 2.7 is nothing more than the cross-correlation function. The maximal length sequence $\bar{s}(t)$ is,

$$\bar{s}(t) = \sum_{n=0}^{\infty} a_{mod(n,L)} [u(t - nT_c) - u(t - (n+1)T_c)] \quad (2.8)$$

Given that $s(t)$ is of the same form as equation 2.8, the time derivative of $s(t)$ is,

$$\begin{aligned} s'(t) &= \sum_{n=0}^{\infty} b_{mod(n,L)} [u'(t - nT_c) - u'(t - (n+1)T_c)] \\ &= \sum_{n=0}^{\infty} b_{mod(n,L)} [\delta(t - nT_c) - \delta(t - (n+1)T_c)] \end{aligned} \quad (2.9)$$

Substituting equations 2.8 and 2.9 into equation 2.7, and evaluating the integral, the dc component is:

$$\begin{aligned} e(t, \Delta) &= \frac{1}{T} \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} a_{mod(n,L)} b_{mod(m,L)} [2u(\Delta + (m-n)T_c) - u(\Delta + (m-n-1)T_c) \\ &\quad - u(\Delta + (m-n+1)T_c)] \end{aligned} \quad (2.10)$$

where $\Delta = \bar{T}_d - T_d$.

Equation 2.10 can be best described by an example. The two cases that are of direct interest are for $\|\Delta\| < T_c$ and $\|\Delta\| \geq T_c$. Assume that the spreading sequence is of length L and

that averaging is over a period of $T = 3L$. Using equation 2.10 and solving for a sequence of length 7, the error signal is:

$$e(\Delta) \approx \begin{cases} \frac{1}{T_c} & 0 < t \leq T_c \\ \frac{-1}{T_c} & -T_c < t \leq 0 \\ 0 & \text{elsewhere} \end{cases} \quad (2.11)$$

The Mathcad simulation is presented in Appendix A. Figure 2.12 shows an idealized representation of equation 2.11. From the figure it can be observed that for a positive Δ the delay line input will be positive and will increase \bar{T}_d and therefore drive Δ to zero. Outside the range $\|\Delta\| \leq T_c$, the DC component is approximately zero and therefore the tracking loop will not function correctly.

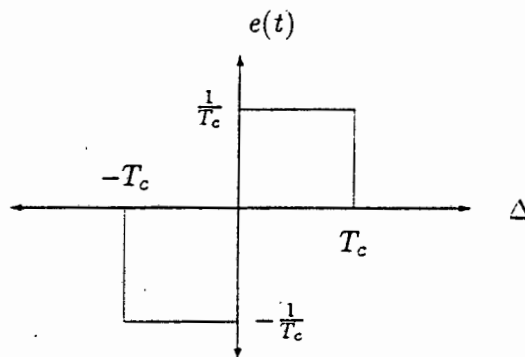


Figure 2.12: Error Signal

Spilker shows in his paper [33] that for a pure sinusoidal input, the delay-lock discriminator functions like a phase-locked loop. Due to the differentiation process and the fact that data modulation will also modulate the delay line control signal, Spilker's implementation is not suited to practical implementation.

A number of implementations based on the Spilker Discriminator have been presented in the literature and are the basis of modern spread spectrum systems. These tracking systems make use of an early phase and late phase of the locally generated receiver code to perform demodulation. These tracking devices are commonly known as Early-Late Tracking Loops.

2.4.2 Baseband Full-Time Early-Late Tracking Loop

Although the baseband early-late tracking loop is not suited to low signal-to-noise ratio applications ², it does serve as the basis for the analysis of other early-late tracking loops. This section will briefly discuss the workings of this tracking loop, based on the analysis presented in [23].

Figure 2.13 shows the basic block diagram of the Baseband Early-Late Tracking Loop.

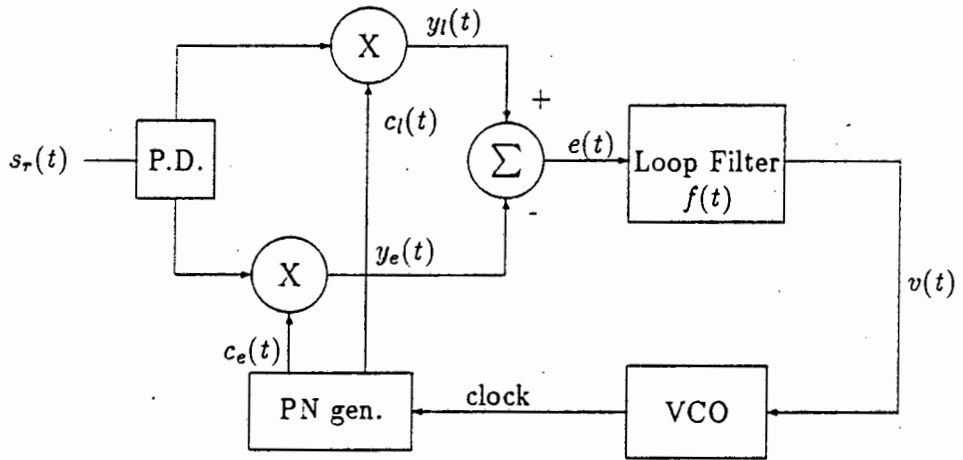


Figure 2.13: Base-Band Full-Time Early-Late Tracking Loop

The signals in figure 2.13 are defined as:

$$s_r(t) = \sqrt{P}c(t - T_d) + n(t) \quad (2.12)$$

$$c_e(t) = K_1c(t - \tilde{T}_d - \frac{\Delta T_c}{2}) \quad (2.13)$$

$$c_l(t) = K_1c(t - \tilde{T}_d + \frac{\Delta T_c}{2}) \quad (2.14)$$

$$y_e(t) = \sqrt{\frac{P}{2}}K_1c(t - \tilde{T}_d - \frac{\Delta T_c}{2})c(t - T_d) \quad (2.15)$$

$$y_l(t) = \sqrt{\frac{P}{2}}K_1c(t - \tilde{T}_d + \frac{\Delta T_c}{2})c(t - T_d) \quad (2.16)$$

The delay-lock discriminator's error signal is the difference between the early and the late arm. i.e. equations 2.15 and 2.16. The error signal is,

²assuming that carrier demodulation is required before code acquisition and tracking

$$e(t) = \sqrt{\frac{P}{2}} K_1 c(t - \tilde{T}_d - \frac{\Delta T_c}{2}) c(t - T_d) - \sqrt{\frac{P}{2}} K_1 c(t - \tilde{T}_d + \frac{\Delta T_c}{2}) c(t - T_d) \quad (2.17)$$

Using the same approach as previously, the time average of the difference signal is a representation of the phase offset. Taking the expectation over one period of the spreading sequence, the dc component $D_\Delta(T_d, \tilde{T}_d)$ is,

$$\begin{aligned} D_\Delta(T_d, \tilde{T}_d) &= \frac{1}{LT_c} \int_{-\frac{LT_c}{2}}^{\frac{LT_c}{2}} c(t - T_d) \left[c(t - \tilde{T}_d - \frac{\Delta T_c}{2}) - c(t - \tilde{T}_d + \frac{\Delta T_c}{2}) \right] \\ &\doteq D_\Delta(\delta) \end{aligned} \quad (2.18)$$

where:

$$\delta = \frac{T_d - \tilde{T}_d}{T_c}$$

Δ is the normalized time difference between early and late channels.

To illustrate what the dc output is for various values of Δ , equation B.13 of Appendix B is used. (repeated here for easy reference)

$$R_{cc}(x) = \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k} a_m \left(1 - \frac{x}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k+1} a_m \left(\frac{x}{T_c}\right) \quad (2.19)$$

Rewriting equation 2.18 in terms of equation 2.19, the dc component is,

$$\begin{aligned} D_\Delta(\delta_1, \delta_2, k_1, k_2) &= \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_1} a_m \left(1 - \frac{\delta_1}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_1+1} a_m \left(\frac{\delta_1}{T_c}\right) \\ &\quad - \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_2} a_m \left(1 - \frac{\delta_2}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_2+1} a_m \left(\frac{\delta_2}{T_c}\right) \end{aligned} \quad (2.20)$$

where:

$$\delta_1 = \text{Frac} \left\{ \left[\delta - \frac{\Delta}{2} \right] T_c \right\}$$

$$\delta_2 = \text{Frac} \left\{ \left[\delta + \frac{\Delta}{2} \right] T_c \right\}$$

$$k_1 = \text{Int} \left\{ \left[\delta - \frac{\Delta}{2} \right] T_c \right\}$$

$$k_2 = \text{Int} \left\{ \left[\delta + \frac{\Delta}{2} \right] T_c \right\}$$

The dc component for various Δ as a function of δ is shown in figure 2.14.

$$R_{cc}(\delta, \Delta) := R_{cc1}(\delta, \Delta) - R_{cc2}(\delta, \Delta)$$

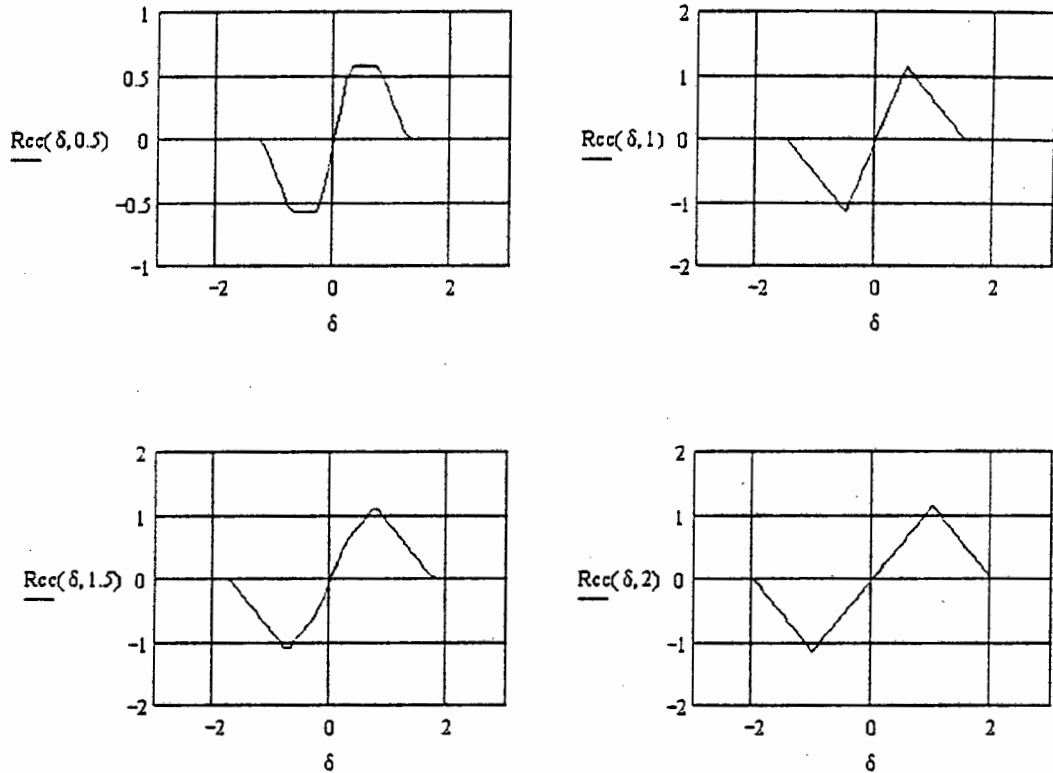


Figure 2.14: DC-Component as a function of Δ and δ

From figure 2.14 it can be observed that there is a range of δ near zero for which $D_{\Delta}(\delta)$ is linearly related to δ . This region is always selected as the normal tracking region of the tracking loop. The question is which of the tracking loops is the best one. The decision is based on locking time versus lock-in range. From the graphs it is clear that a $\Delta = 2$ can cope with the large phase-offsets, but the convergence to zero will be slower and the phase detector noise is enhanced. A $\Delta \leq 1$ will cope only with small phase-offsets, but will be faster if the phase-offset is within the linear range; otherwise it will be slower. The choice of Δ should therefore be selected based on the requirements of the system.

The filtered error signal will either speed up or slow down the voltage controlled oscillator. Thus for a negative offset, the VCO frequency is decreased, while for a positive offset the frequency is increased.

The circuit presented in figure 2.13 has a number of drawbacks related to data modulation and the need for coherent synchronisation before the despreading process. The generation of a coherent reference at low signal-to-noise ratios is difficult. The tracking loop is also sensitive to data modulation. The discriminator curve will be inverted according to the

received sign of the data bit. Thus for a negative bit the VCO will be driven in the wrong direction and cause the system to become unsynchronised. (In [34] a tracking loop is presented that does not require carrier demodulation before using a coherent early-late tracking loop.) The alternative is to employ non-coherent techniques. The non-coherent loops differs significantly from the base-band loop just presented.

2.4.3 Full-Time Early-Late Noncoherent Tracking Loop

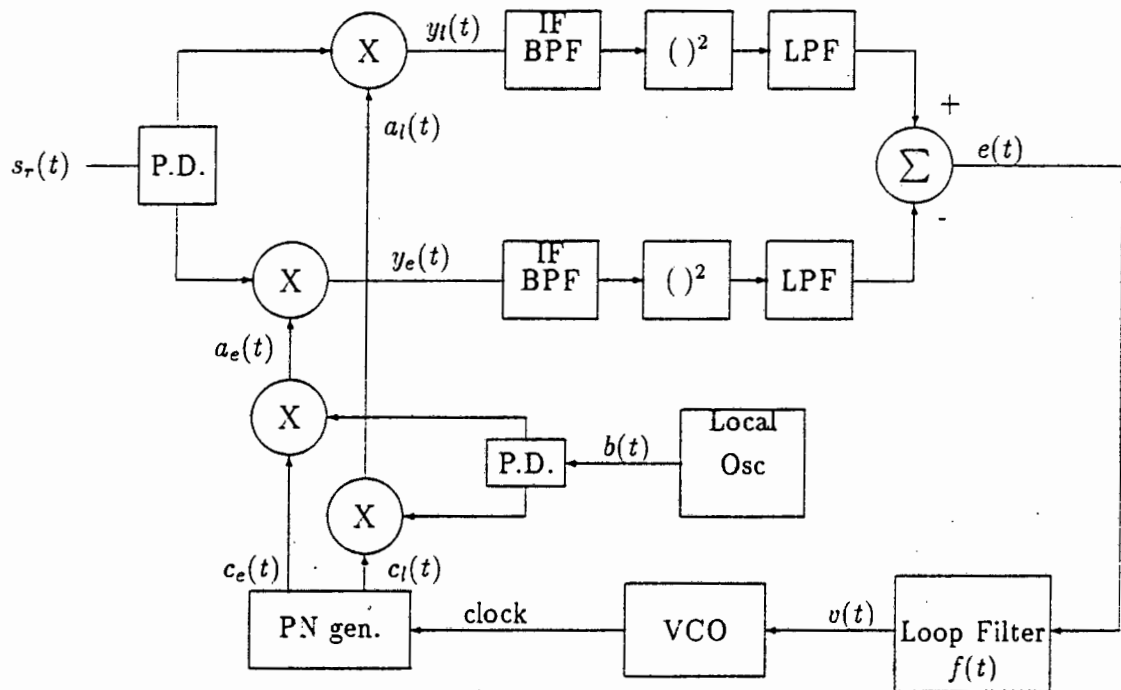


Figure 2.15: Full-Time Early-Late Noncoherent Tracking Loop

The fundamental difference between the previous implementation and this implementation is the use of energy detectors. Energy detectors are not sensitive to data modulation or carrier phase. This enables the tracking loop to ignore these attributes of the received signal.

The error signal dc component used to drive the VCO can be calculated in a similar way to the baseband tracking loop. The dc component is [23]:

$$D_{\Delta}(\delta_1, \delta_2, k_1, k_2) = \left[\frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_1} a_m \left(1 - \frac{\delta_1}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_1+1} a_m \left(\frac{\delta_1}{T_c}\right) \right]^2 - \left[\frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_2} a_m \left(1 - \frac{\delta_2}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+k_2+1} a_m \left(\frac{\delta_2}{T_c}\right) \right]^2 \quad (2.21)$$

where:

$$\delta_1 = \text{Frac} \left\{ \left[\delta - \frac{\Delta}{2} \right] T_c \right\}$$

$$\delta_2 = \text{Frac} \left\{ \left[\delta + \frac{\Delta}{2} \right] T_c \right\}$$

$$k_1 = \text{Int} \left\{ \left[\delta - \frac{\Delta}{2} \right] T_c \right\}$$

$$k_2 = \text{Int} \left\{ \left[\delta + \frac{\Delta}{2} \right] T_c \right\}$$

The dc component for various Δ as a function of δ is shown in figure 2.16.

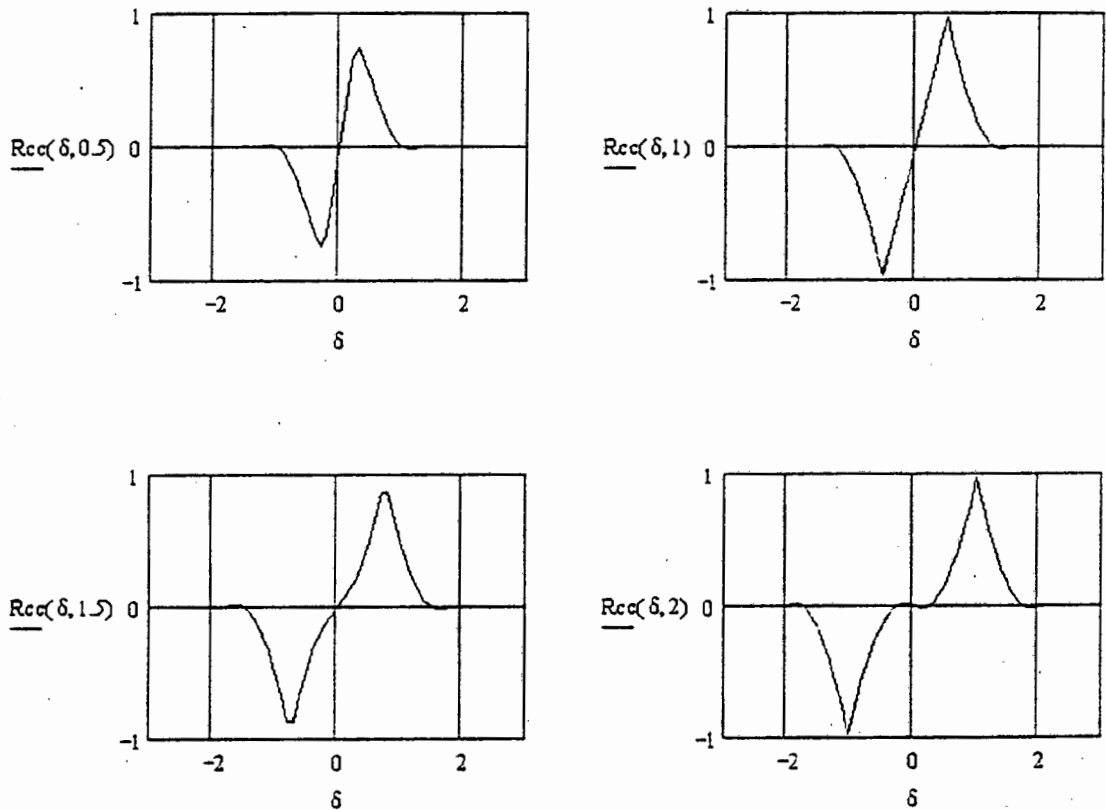


Figure 2.16: DC-Component as a function of Δ and δ

From the graphs, an interesting effect occurs for $\Delta = 2$. Around $\delta = 0$, the gradient tends to flatten out, which is an undesirable characteristic. It is therefore necessary to select a Δ

less than 2. A $\Delta = 1.5$ is a reasonable compromise. For a detailed analysis see Ziemer and Pieterse [23].

The full-time early-late non-coherent tracking loop, although widely used, has two problems. The first is that the components of the loop are expensive, and the second is that the early and late channels need to be amplitude balanced. When the channels are unbalanced, the discriminator characteristic is offset and does not produce zero volts when the phase-offset is zero. This led researchers to find new ways of overcoming the amplitude imbalance and reduce the cost of implementation. Hartmann [35] came up with a solution known as the tau-dither early-late non-coherent tracking loop.

2.4.4 Tau-Dither Early-Late Non-coherent Tracking Loop

The Tau-Dither Early-Late Non-coherent Tracking Loop solves both the problems of the full-time loops discussed in the previous section. This is achieved by time-sharing the correlation circuitry between the early and late channels. A block diagram of the tau-dither loop is shown in figure 2.17.

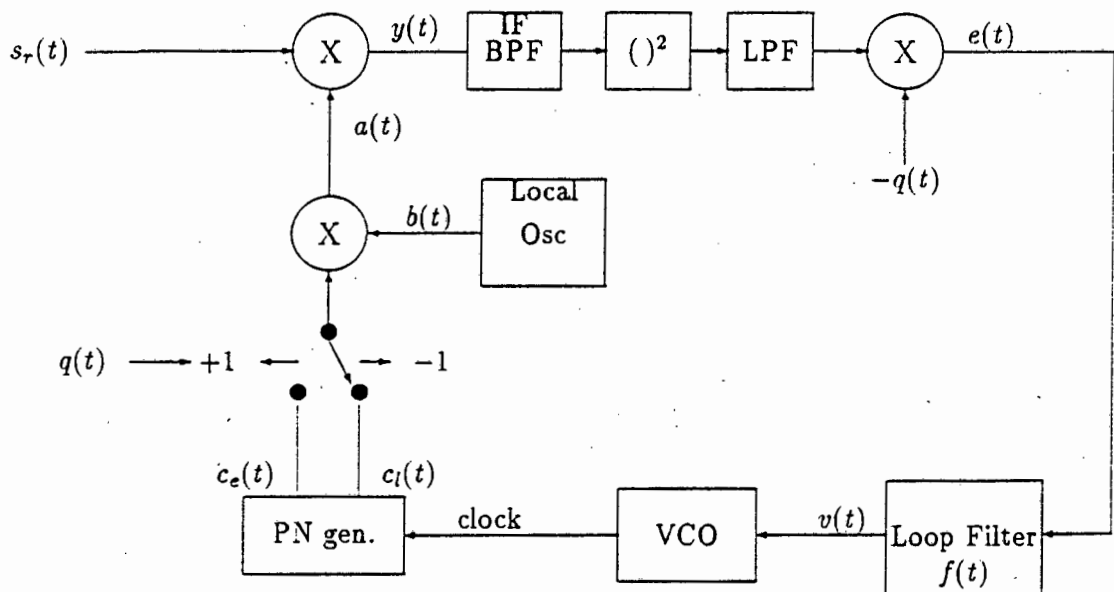


Figure 2.17: Tau-Dither Early-Late Non-coherent Tracking Loop

The loop operation is very similar to that of the full-time loop. $q(t)$ is a periodic square-wave, switching between ± 1 . When $q(t)$ is negative the loop acts as a late-correlator, and

when positive as an early correlator. To obtain the correct discriminator characteristic an inverted $q(t)$ is included at the output of the low-pass filter. This ensures that similar curves, as shown in figure 2.16, are obtained.

From the implementation it is clear that no channel imbalance will occur to cause the discriminator characteristic to be offset. However if $q(t)$ is offset the discriminator characteristic will be affected.

Noise performance of the tau-dither loop is inferior to that of the full-time loop. To overcome this problem, and still overcome channel imbalance, Hopkins [36] proposed the double-dither loop. The noise performance of this loop is the same as that of the full-time loop. However the price paid for this noise improvement is complexity.

The double-dither loop is a combination of the full-time and tau-dither loop. It consists of two correlator branches. The early and late channels are switched between the two correlator channels, i.e. during the negative cycle of $q(t)$ the lower branch is used to correlate the late channel and during the positive cycle of $q(t)$ the lower branch is used to correlate the early channel.

Other methods have also been proposed to overcome the carrier imbalance problem. However, many of these methods imply an increase in hardware complexity. One such method was proposed in 1991 by De Gaudenzi and Luise [34]. They proposed a coherent decision directed coherent delay-lock tracking loop. The essential changes were the inclusion of a Costas Loop for carrier and data recovery. This is in turn used to derive the base-band error signal. Moreover, a single passband correlator is used to perform the early-late correlation.

2.5 Estimation Tracking Techniques

The methods described up to now, have only considered code acquisition and tracking as separate issues. At higher signal-to-noise ratios it is possible to reduce the hardware complexity and estimate the received pseudonoise sequence and use this information to load the local code generator with the estimated tuple. One such method was proposed by Ward [22] and later extended by Ward and Yiu [37].

2.5.1 Synchronisation by Estimating the Received Spreading Code

RASE or Rapid Acquisition by Sequential Estimation is fundamentally different from the acquisition techniques discussed in this chapter. RASE is suitable for medium to high input signal-to-noise ratios, a common condition of many high-quality tracking systems [22]. RASE is based on a simple principle that in the absence of noise the optimal tracking technique is simply to load the first n received chips into the receiver sequence generator and let the generator start from that initial condition. The generator will then continue to generate a sequence which is nearly in phase (less than one chip) with the incoming sequence. A tracking loop can then maintain phase and cope with Doppler from that time on.

The RASE system makes its best estimate of the first n received bits, loads the receiver sequence generator with that estimate, and starts the operation of the sequence generator and tracking circuits. If the correct estimate was loaded, tracking will occur. At the same time, a cross-correlation is performed between the incoming sequence (after the low pass filter) and the locally generated sequence. The cross-correlation value gives an indication of whether the correct estimate was made and if tracking is taking place. In the case where the correlation value indicates that an incorrect estimate was made, a new estimate will be loaded and tracked. The process continues until the correct estimate is obtained.

The complete RASE system is shown in figure 2.18, combined with a delay-lock loop. The input signal is a NRZ binary sequence plus additive white Gaussian noise. The signal plus noise is low-pass filtered (using an ideal brickwall filter) in a bandwidth of $\frac{1}{T_c}$. The filtered signal is hard-limited and the output is an estimate of the input binary chip. An n -chip cycle-and-hold counter is reset at intervals T_e called the examination period. Each time the counter is reset, it counts for n periods of the local generator clock and then holds until reset. While the n -bit counter is counting, the local clock causes the limited signal to be sampled n times and those samples are shifted into the sequence generator through the loading switch. Figure 2.19 shows the chip estimation circuitry. If no Doppler shift is present, then the local clock period will be nearly the same as the incoming clock period. This implies that the sample process obtains n consecutive chips of the input. When the n chips are loaded, the tracking loop is closed.

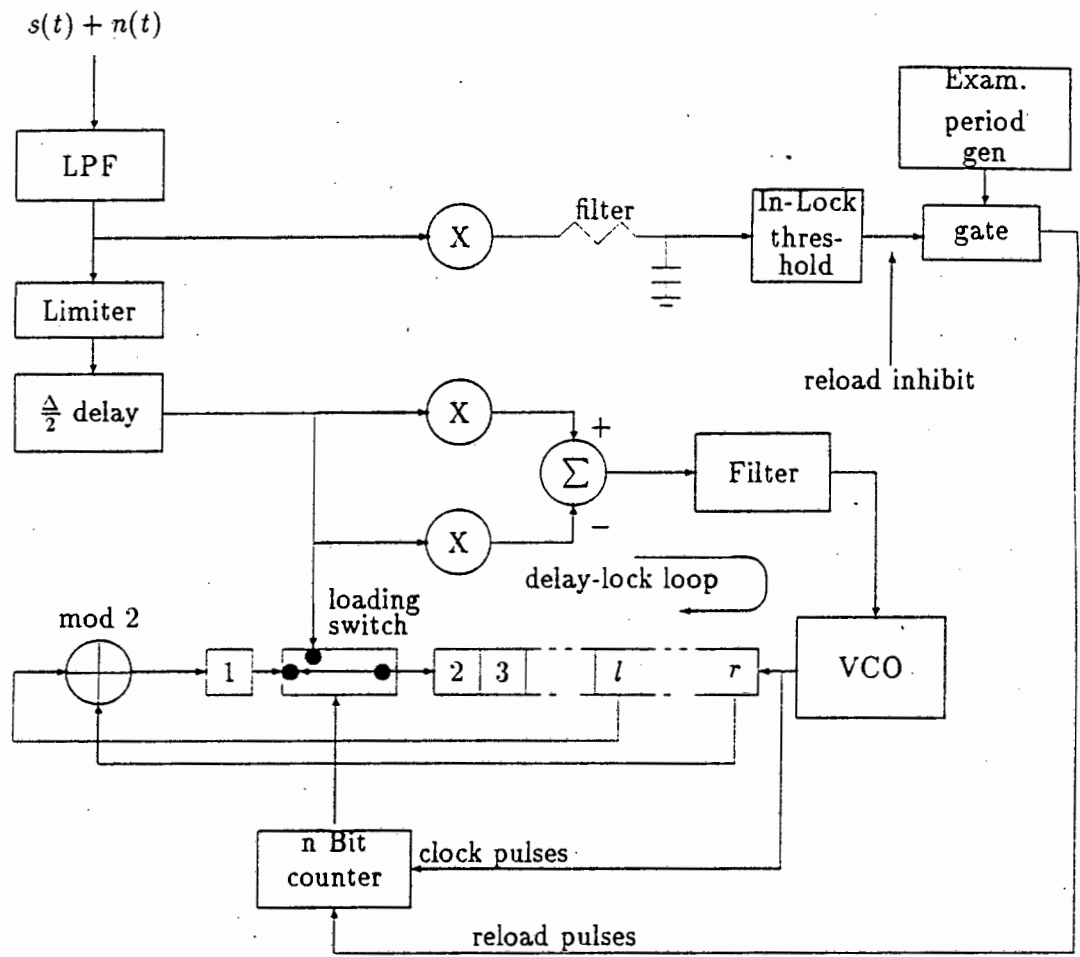


Figure 2.18: RASE synchronisation with full-time early-late code tracking (See Ward)

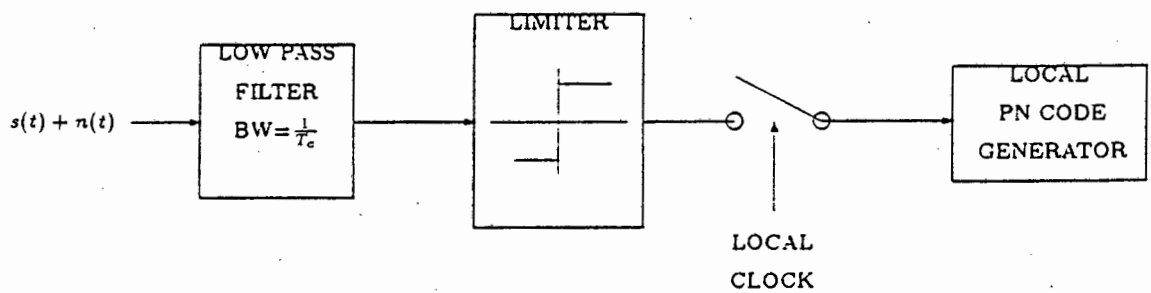


Figure 2.19: Sequential Detector

If the correct estimate was loaded, the delay-lock loop will correct for any phase offsets and reduce the timing error to much less than half a chip. If the incorrect estimate was loaded, the VCO frequency will not shift significantly because of the near zero correlation value. To determine if the system has achieved lock, the low pass filtered signal is correlated with the locally generated sequence. The signal is feed to an integrate-and-dump circuit. The integration period is typically over many chip periods (the lower the signal-to-noise ratio, the longer the integration time (T_e)). If the output of the integrate-and-dump circuit is below a pre-determined threshold, the system enters the acquisition mode again.

2.5.1.1 Acquisition Time

Ward, in [22], gives a detailed analysis of the acquisition time. The average locking time is, assuming no Doppler effects is:

$$T_a = \frac{T_e + nT_c}{p^n} \quad (2.22)$$

where:

T_a is the acquisition time

T_e is the examination period

T_c is the chip period

p is the probability of success

n is the number of registers in the sequence generator

At very high signal-to-noise, the probability of success approaches 1, and equation 2.22 becomes:

$$T_a = T_e + nT_c \quad (2.23)$$

At low signal-to-noise ratios p approaches 0.5, giving a mean synchronisation time of:

$$T_a = (T_e + nT_c)2^n \quad (2.24)$$

2.5.1.2 Noise

So far the effect of noise has been ignored on the average acquisition time. It does occur that the noise will occasionally cause a false in-lock indication (false alarm), and also occasionally will cause a true lock to go unnoticed (false dismissal.) Ward gives the average acquisition time as:

$$T_a = \frac{T_e + nT_c}{p^n(1 - p_{fa})(1 - p_{fd})} \quad (2.25)$$

where:

p_{fa} is the probability of false alarm

p_{fd} is the probability of false dismissal

2.5.1.3 Doppler Effects

One of the issues that has so far not been addressed for RASE is the effect of Doppler. Since the lock process relies on a local clock, which is only synchronised after tracking is complete, RASE cannot deal effectively with Doppler. If the incoming frequency is below the receiver's local clock, the same chip will be sampled more than once. If the frequency is above the local clock frequency, the chips are not sampled consecutively. Therefore if the frequency deviation from the local clock is large, the time to synchronise will be increased dramatically. Equation 2.23 gives the mean synchronisation (acquisition) time taking the effects of Doppler, oscillator instabilities and receiver input noise into account (see [22] for detailed derivation).

$$T_a = \frac{T_e + nT_c}{p^n(1 - p_{fa})^2(1 - \frac{nf_d}{f_e})} \quad (2.26)$$

where:

f_d is the difference frequency

f_e is the local reference frequency.

2.6 Summary

In this chapter existing code acquisition and tracking techniques were presented. It was emphasised that the acquisition and tracking was a two prong approach, where two synchronisation loops are required to achieve complete synchronisation. In the next chapter time-variant spectral analysis techniques will be discussed. The results derived will be used in subsequent chapters to design two novel synchronisation networks.

Chapter 3

Theoretical Concepts: The Search for Low Cost Synchronisation Methods

In Chapter 2 it was stated that spread spectrum synchronisation requires a two-pronged approach. By this it was implied that first, the code-phase and frequency had to be obtained within certain bounds, and then the code-phase tracked to maintain synchronisation. These methods are expensive and relatively complex in their implementation.

The purpose of this dissertation is to find low-cost methods of code-phase acquisition and code-phase tracking that are easy to implement, low in cost and use minimal hardware. The research method used was to investigate both theoretical and practical aspects at the same time. This led to some interesting designs.

In this chapter and those that follow, a step by step approach will be presented in the analysis and design of this novel base-band synchronisation loop. The tools required to analyse the pseudo-noise sequence synchronisation system will be developed and presented.

3.1 Research Objectives

The designer is faced with a number of trade-offs to obtain a reasonable solution that meets the given requirements. In engineering research, the researcher also needs to keep these trade-offs in perspective. One of the trade-offs is signal-to-noise ratio versus complexity.

The compromise is complexity with higher signal-to-noise performance.

3.2 Time-Variant Spectral Analysis

Signals obtain a wealth of hidden and useful frequency information that can be helpful in various applications. Unfortunately in many engineering courses and textbooks, time variant spectral analysis is seldom, if ever, mentioned or discussed, yet it is fundamental to many of the tools and equipment that are used in everyday engineering. Frequency spectra build up as time moves on, and as more information is obtained, so the picture becomes clearer, and a fairer representation of the infinite-time spectrum is obtained. In this build-up of the infinite-time spectrum (i.e. Fourier Transform), the changes in frequency could be useful when applied to certain applications.

The heart of spread spectrum systems are the pseudo-noise sequences, which forms the basis of all code synchronisation networks used in one way or another. The question arises whether it is possible to use the time-variant spectral information contained in the sequences to achieve the goal of this research: *to obtain a low-cost all-in-one synchronisation network*. The rest of the chapter focuses on the frequency information that can be obtained from spreading sequences and whether it can be applied to the development of a synchronisation network.

3.2.1 Characteristics of Pseudo-noise Sequences

Pseudo-noise signals appear to have little or no useful frequency information. The Fourier spectrum of pseudo-noise sequences are discrete, since the sequences are periodic in nature. The characteristic spectrum is shown in figure 3.1. (See Appendix D.3 for derivation.)

The following characteristics can be identified from figure 3.1:

- No frequency component exists at the chipping frequency, i.e. $\frac{1}{T_c}$.
- The number of spectral lines that occur between 0 and the first null is $L - 1$.
- The discrete components are spaced $\frac{1}{LT_c}$ apart.

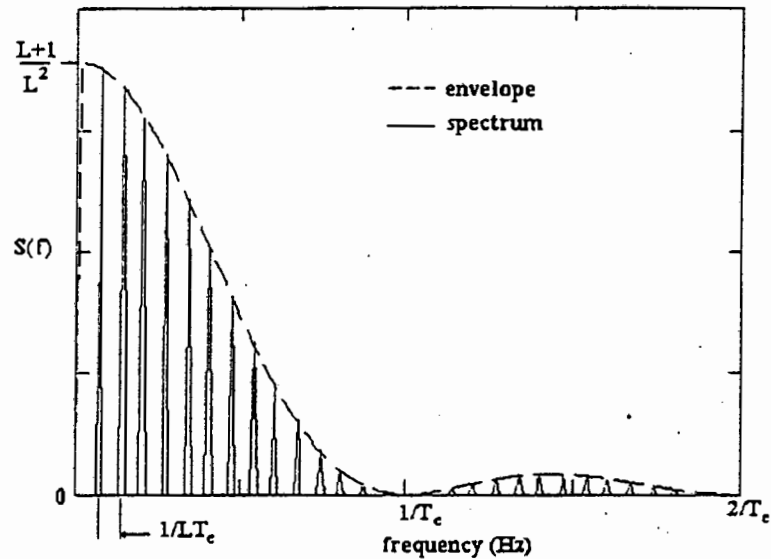


Figure 3.1: One-Sided Fourier Spectrum of a Pseudo-Noise Sequence

It can be seen from the infinite-time spectrum, that no useful information is present in the sequence when examined over a complete period. Not even the chipping frequency is present in the spectrum (typical characteristic of NRZ type waveforms). It is therefore necessary to probe deeper into what operations can be performed on the waveform to obtain more information.

3.2.2 Positive Edge Detection

A number of methods have been developed to obtain clock information from NRZ type waveforms. One well known operation is positive edge detection. On every edge a positive impulse is produced. Depending on the implementation, the frequency obtained would be at the original frequency or at a harmonic of the original frequency.

Since a pseudo-noise sequence can be classified as an NRZ waveform, the original clock frequency will be obtained after positive edge detection. Figure 3.2 shows the time waveform of a pseudo-noise sequence and the resultant waveform after positive edge detection.

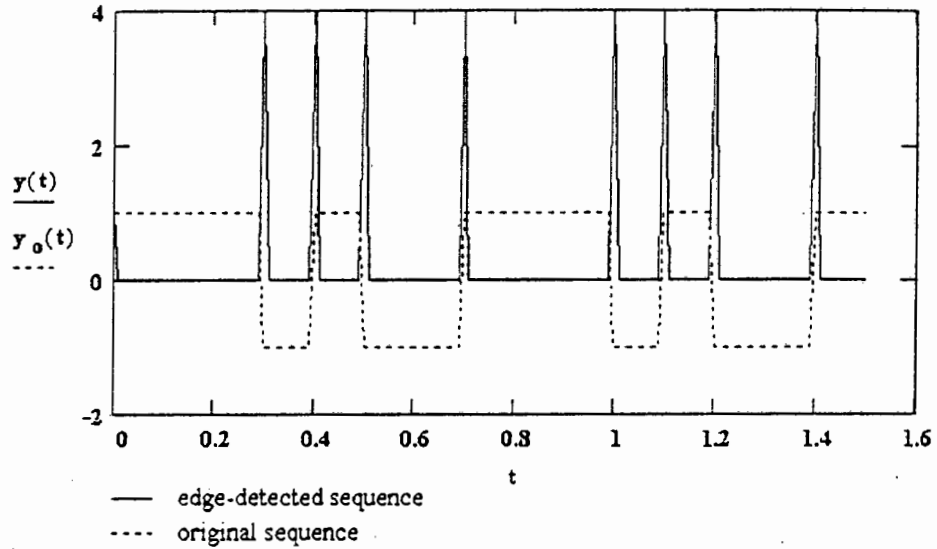


Figure 3.2: A Positive Edge Detected PN Sequence

To show that the edge detection does give a frequency component at the clock frequency (or double the frequency depending on definition), consider the following derivation. For simplicity of calculation it is assumed that an impulse occurs every T_c seconds where a_n is either '1' or '0'. The edge detected time wave-form can be defined as :

$$f(t) = \sum_{n=-\infty}^{\infty} a_n \delta(t - nT_c) \quad (3.1)$$

Since $f(t)$ is not periodic, it is necessary to determine the autocorrelation function and then obtain the Fourier transform of the resultant correlation. Using the result given in Stremler [38], the autocorrelation of $f(t)$ is

$$R_{ff}(\tau) = \sum_{n=-\infty}^{\infty} \delta(\tau - nT_c) \quad (3.2)$$

Since $R_{ff}(\tau)$ is a periodic function with period T_c , it can be expressed as a Fourier series by choosing $\omega_o = \frac{2\pi}{T_c}$. The Fourier series of equation 3.2 is:

$$R_{ff}(\tau) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_o\tau} \quad (3.3)$$

where

$$F_n = \frac{1}{T_c} \int_{-\frac{T_c}{2}}^{\frac{T_c}{2}} \delta(\tau) d\tau = \frac{1}{T_c} \quad (3.4)$$

The Power Spectrum, given in equation 3.5, is calculated by taking the Fourier transform of equation 3.3.

$$\begin{aligned} F(\omega) &= \mathcal{F}\{R_{ff}(\tau)\} \\ &= \frac{1}{T_c} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} e^{jn\omega_0\tau} e^{-j\omega\tau} d\tau \\ &= \frac{2\pi}{T_c} \sum_{n=-\infty}^{\infty} \delta(\omega - n\frac{2\pi}{T_c}) \end{aligned} \quad (3.5)$$

where:

$$\mathcal{F}\{e^{j\omega_0 t}\} = 2\pi\delta(\omega - \omega_0)$$

It is clearly seen from equation 3.5 that the power spectrum is a series of impulses spaced $\frac{1}{T_c}$ Hz apart, and that a frequency component does exist at the frequency of interest. Using a phase-locked loop type implementation it is possible to obtain and track the original clock frequency [39].

3.2.3 Effect of Despreading a Signed Edge Detected (Differentiated) Pseudo-Noise Sequence

Despreading a signed edge detected pseudo-noise sequence produces some very interesting results. Intuitively and from the information presented thus far it is known that two pseudo-noise sequences are uncorrelated if the code-offset is greater than one chip, while two sequences offset by less than one chip are correlated. What occurs when a pseudo-noise sequence is multiplied by a signed edge-detected sequence?

Mathematically, a pseudo-noise sequence can be expressed as:

$$p(t) = \sum_{i=-\infty}^{\infty} a_{\text{mod}(i,L)} [u(t - iT_c) - u(t - (i+1)T_c)] \quad (3.6)$$

where the co-efficients ($a_{mod(i,L)}$) of the PN sequence are ± 1 .

Applying signed edge detection ,i.e. differentiation, the sequence can be written as

$$q(t) = \sum_{j=-\infty}^{\infty} b_{mod(j,L)} [\delta(t - jT_c) - \delta(t - (j + 1)T_c)] \quad (3.7)$$

where the co-efficients ($b_{mod(j,L)}$) of the original PN sequence are ± 1 .

Multiplying equation 3.6 and 3.7 and simplifying

$$\begin{aligned} r(t) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \left[\begin{array}{l} u(t - iT_c)\delta(t - jT_c) \\ -u(t - (i + 1)T_c)\delta(t - jT_c) \\ -u(t - iT_c)\delta(t - (j + 1)T_c) \\ +u(t - (i + 1)T_c)\delta(t - (j + 1)T_c) \end{array} \right] \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \left[\begin{array}{l} u(jT_c - iT_c)\delta(t - jT_c) \\ -u(jT_c - (i + 1)T_c)\delta(t - jT_c) \\ -u((j + 1)T_c - iT_c)\delta(t - (j + 1)T_c) \\ +((j + 1)T_c - (i + 1)T_c)\delta(t - (j + 1)T_c) \end{array} \right] \quad (3.8) \end{aligned}$$

where a_i and b_j are the same pseudo-noise sequences, except that their starting phases may differ. Figure 3.3 is a diagrammatic representation of equation 3.8.

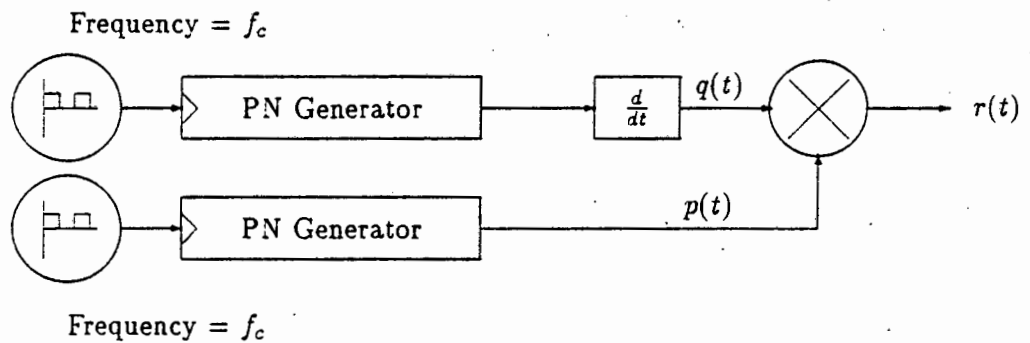


Figure 3.3: Block Diagram representation of the despreading process

To illustrate equation 3.8, consider the case of two 16 bit pseudo-noise sequence offset by one chip (see Appendix E for Mathcad Simulation). Figure 3.4 shows the result. From the figure interesting characteristics can be identified.

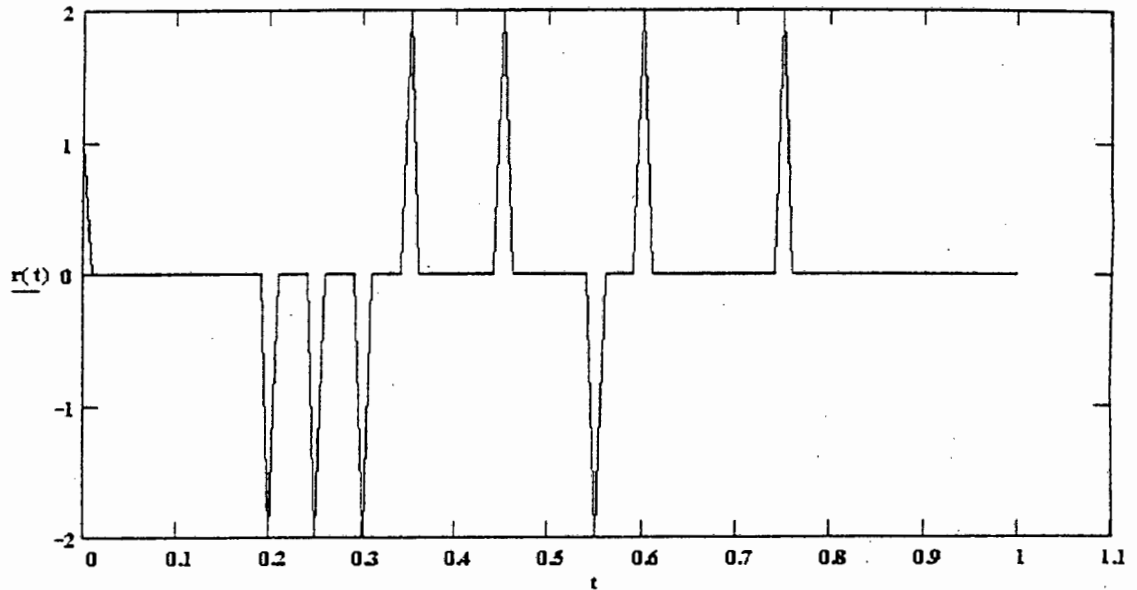


Figure 3.4: Product of an edge detected and native pn sequence

From spectral analysis it is well known that taking Fourier transform implies averaging the signal multiplied by a complex exponential over the time duration of the signal. For long sequences the time duration extends to infinity. In real-time applications, the frequency at any instant in time can be useful. A typical example of such an application is speech processing. There is no need for complex windowing when dealing with trains of impulses. Assuming that the $r(t)$ is aperiodic, the Fourier time-transform can be calculated directly. Therefore,

$$R(\omega, t) = \int_0^t r(\tau) e^{-j\omega\tau} d\tau \quad (3.9)$$

Note that the start-time of integration is a constant value. It has been assumed that the integrator has no losses and its memory is infinite. It is also assumed that prior to $t = 0$ the input to the integrator was zero. Substituting equation 3.8 into 3.9 and simplifying,

$$R(\omega, t) = \int_0^t \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{mod(m,L)} b_{mod(n,L)} \left[\begin{array}{l} u(nT_c - mT_c)\delta(\tau - nT_c) \\ -u(nT_c - (m+1)T_c)\delta(\tau - nT_c) \\ -u((n+1)T_c - mT_c)\delta(\tau - (n+1)T_c) \\ +((n+1)T_c - (m+1)T_c)\delta(\tau - (n+1)T_c) \end{array} \right] e^{-j\omega\tau} d\tau$$

$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{mod(m,L)} b_{mod(n,L)} \begin{bmatrix} u(nT_c - mT_c)e^{-j\omega nT_c} \\ -u(nT_c - (m+1)T_c)e^{-j\omega nT_c} \\ -u((n+1)T_c - mT_c)e^{-j\omega(n+1)T_c} \\ +((n+1)T_c - (m+1)T_c)e^{-j\omega(n+1)T_c} \end{bmatrix} \quad (3.10)$$

where $j = \sqrt{-1}$

Assuming a long pseudo-noise sequence with a chipping frequency of one Hertz and evaluating over a time window of the sequence using equation 3.10 some interesting results are obtained. Consider figure 3.5. As, the evaluation window is increased, so the frequency component at the chipping frequency becomes more pronounced.

Now consider the case where ones and zeros alternate in some random fashion (figure 3.6). $|F1(\omega, T_c)|$ shows the case of a positive impulse followed by a negative impulse. $|F2(\omega, 5T_c)|$ shows the case for a number of positive impulses following the previous case. It is clear that a frequency component appears at the chipping frequency. $|F3(\omega, 8T_c)|$ shows the effect when a number of negative impulses follow the series of positive impulses. The component at the chipping frequency disappears. $|F4(\omega, 11T_c)|$ and $|F5(\omega, 13T_c)|$ show similar results. (See Appendix E.4 for more details)

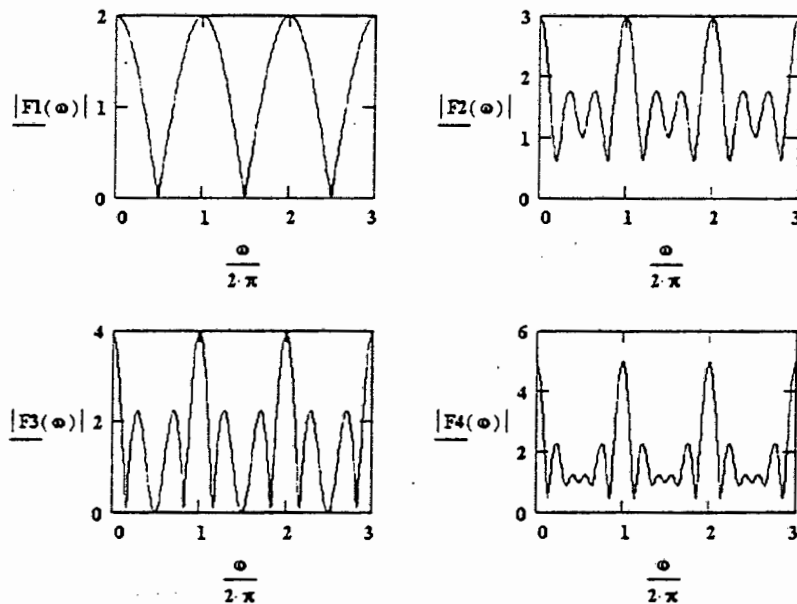


Figure 3.5: Fourier-Time Spectrum (all ones). The frequency component at 1 Hz and multiples thereof become more defined as the evaluation time window increases.

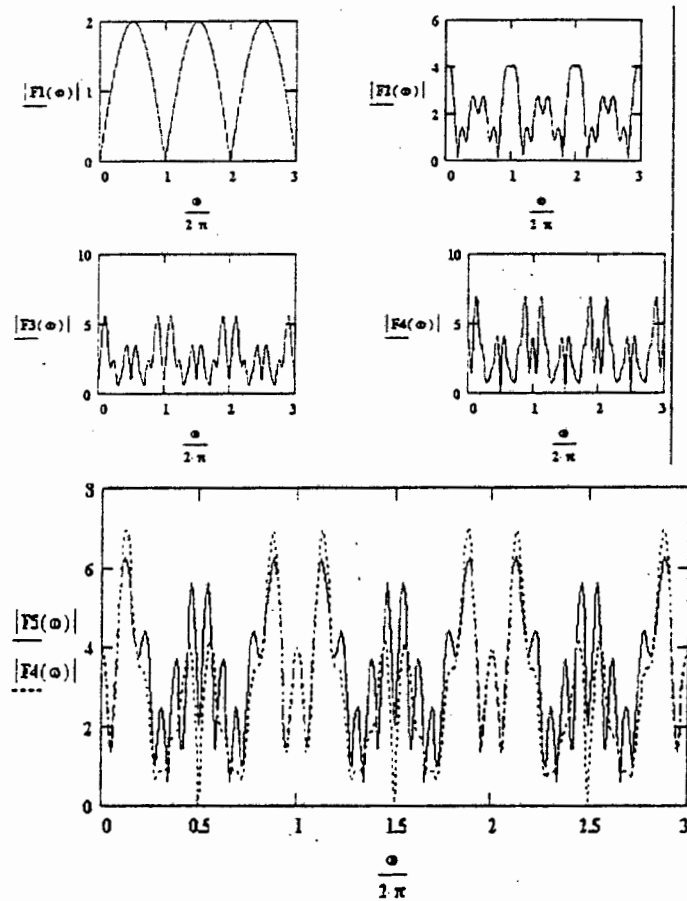


Figure 3.6: Fourier-Time Spectrum at various time intervals. By randomly varying the sign of the impulses, the frequency component's amplitude at 1 Hz and multiples thereof will not remain constant.

It was assumed that an impulse occurred at every chipping instant. In reality, a pseudo-noise sequence consists of a combination of varying pulse widths and therefore the distance between impulses is not constant. This does not effect the resultant spectrum dramatically, except for introducing other frequency components based on the distances between the impulses.

The question arises how the time-variant spectral analysis can be applied to the synchronisation of spread spectrum systems. From the analysis presented, the following can be defined:

- Spectral information is concentrated around the clock frequency, and is forever changing. This information can therefore be used to control the frequency of a voltage controlled oscillator.

- Only when a continuous time series of all positive or all negative impulses is obtained, can it be stated that the system has achieved lock.
- When correlating a PN sequence with an edge detected PN sequence, a three valued correlation function is obtained. (see chapter 2,section 2.4.1, figure 2.11)

From these three results it is possible to postulate a system that will perform both acquisition and tracking. The basic elements of such a system are a frequency tracking element, a local code generator, signed edge detector and filters. From the results two systems were developed, analysed and simulated. In the next chapter the first novel implementation is presented.

3.3 Summary

In this chapter the spectral characteristics of pseudo-noise sequences was shown. The analysis was extended to determine if it is possible to obtain a unified approach to the problem of synchronising spreading sequences. Three facts were stated and from this the basic building blocks of a unified synchronisation network were defined.

Chapter 4

Novel Synchronisation Technique-Method One

In Chapter 3 a detailed time-variant spectral analysis of the product of two PN sequences (one edge detected) was presented. From the results it was stated that two possible spread spectrum synchronisers could be postulated. In this chapter, the first system will be presented and compared with the present state of the art presented in the literature. It will be shown that this method is novel and low-cost.

4.1 System Description

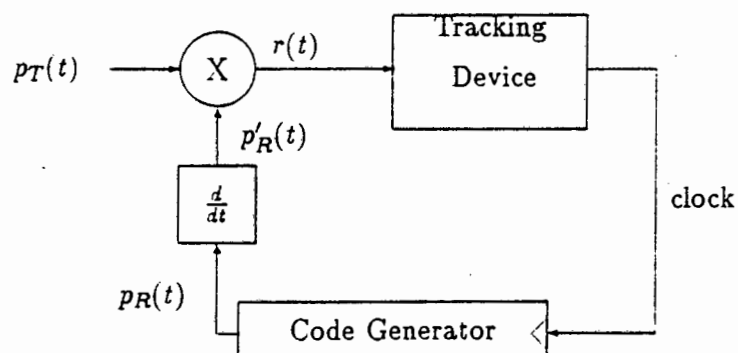


Figure 4.1: Novel Spread Spectrum Synchroniser

From the results of the time variant spectral analysis the following novel system shown in

figure 4.1 was developed. The output of the multiplier $r(t)$ is given by equation 3.8, except for one fundamental change. The frequency of the locally generated code is a function of the frequency of the clock generator, and is in turn a function of the driving signal of impulses. Equation 3.8 therefore becomes,

$$\begin{aligned}
 r(t) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \begin{bmatrix} u(t - iT_c)\delta(t - jT_v(t)) \\ -u(t - (i+1)T_c)\delta(t - jT_v(t)) \\ -u(t - iT_c)\delta(t - (j+1)T_v(t)) \\ +u(t - (i+1)T_c)\delta(t - (j+1)T_v(t)) \end{bmatrix} \\
 &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \begin{bmatrix} u(jT_v(t) - iT_c)\delta(t - jT_v(t)) \\ -u(jT_v(t) - (i+1)T_c)\delta(t - jT_v(t)) \\ -u((j+1)T_v(t) - iT_c)\delta(t - (j+1)T_v(t)) \\ +(j+1)T_v(t) - (i+1)T_c)\delta(t - (j+1)T_v(t)) \end{bmatrix}
 \end{aligned} \tag{4.1}$$

where $\frac{1}{T_v(t)}$ is the clock frequency at time t , a_n is the transmitted sequence co-efficients and b_n the receiver generated sequence co-efficients. With the introduction of time dependency in equation 3.8, equation 4.1 is non-linear. The term $T_v(t)$ is dependent on the frequency of the clock and therefore is dependent on the impulses entering the tracking device. The change in polarity of the impulses and the period between two impulses is therefore random. These attributes of the system make the analysis complex. As a first stage investigation, a practical system was constructed (see chapter 9) and simulations using Tesla¹ were run.

4.2 Simulation of System

To implement the simulation of the system, the diagram given in 4.1 had to be converted to a practical implementation. The clock generator was designed as a second order phase-locked loop. A bandpass filter was placed before the input of the phase-locked loop. The purpose of this was to create sinusoidal signals as input to the phase-locked loop.

¹©1990-1993 by TESOF, Inc. (see Appendix F for details.)

4.2.1 Simulation Block Diagram

The simulation block diagram is given in figure 4.2, with the circuit node numbers as indicated.

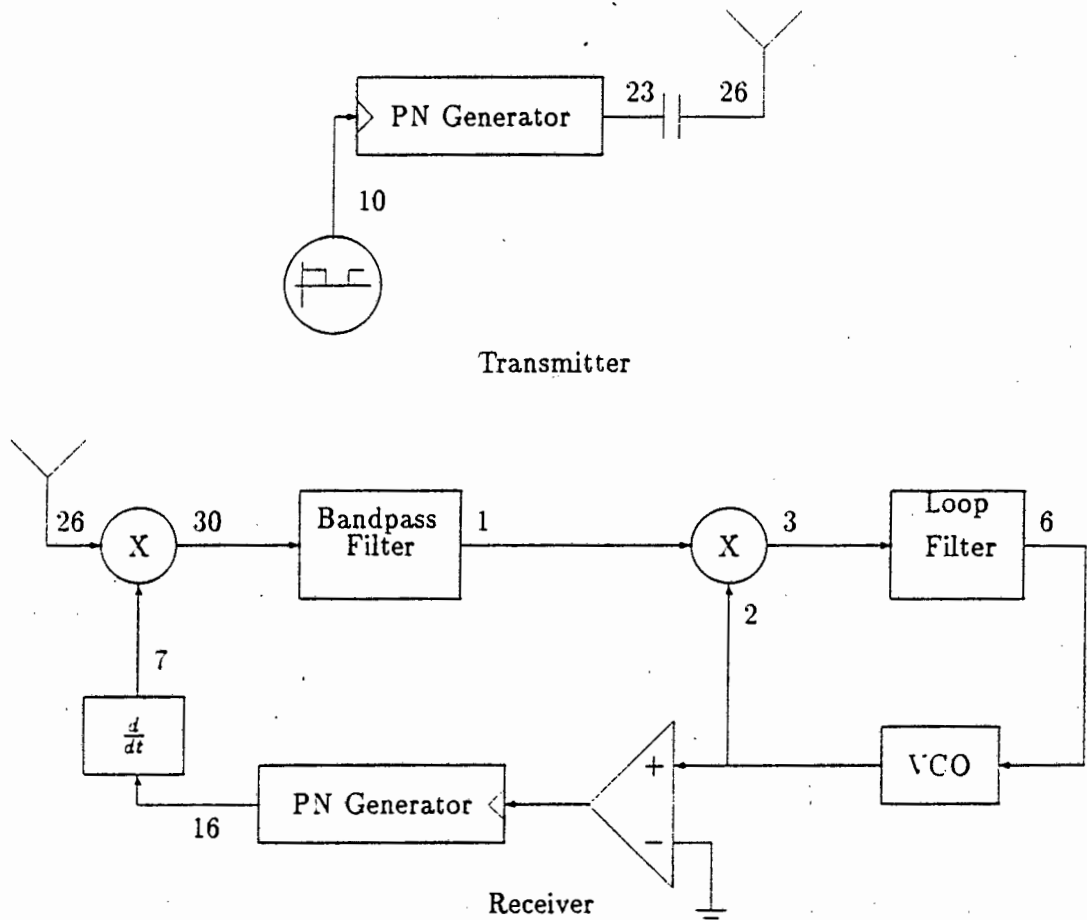


Figure 4.2: Tesla Simulation Model of Novel Method One

4.2.2 Simulation Parameters

For the purpose of this simulation the parameters are given in table 4.1.

Parameter	Value
Sequence Length	15
Code Offset	2
Tx Clock Frequency	90 kHz
BPF Bandwidth	20 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	40 kHz
Loop Filter Bandwidth	20 kHz

Table 4.1: Simulation Parameters - Method 1

4.2.3 Results of the Simulation

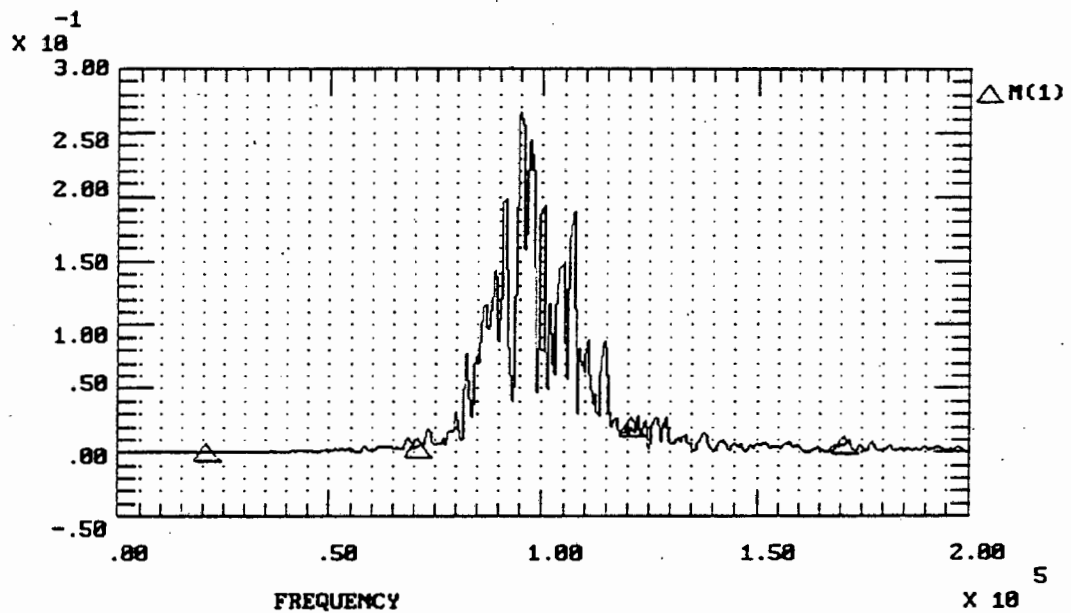


Figure 4.3: Spectrum out of Bandpass Filter. The frequency of the sinusoid generated by the bandpass filter varied around the centre frequency. This would in turn cause the phase-locked loop to change frequency, which will increase or decrease the chipping rate of the locally generated PN sequence.

Interesting results were obtained from the simulation. Figure 4.3 shows the output of the bandpass filter (node 1). The frequency of the sinusoid generated by the bandpass filter varied around the centre frequency. This would in turn cause the phase-locked loop to

change frequency, which will increase or decrease the chipping rate of the locally generated PN sequence. The frequency of the transmitter code generator being most prominent. With time, the other frequency components will become less apparent.

Figure 4.4 shows the output of the first multiplier (node 30). As was previously mentioned, a series of all positive or all negative spikes is an indication of synchronisation. For this case a negative impulse lock was achieved within 0.8 milliseconds. However, the loop suffers from jitter and this is indicated by the presence of positive spikes after lock. The loop does however recover lock.

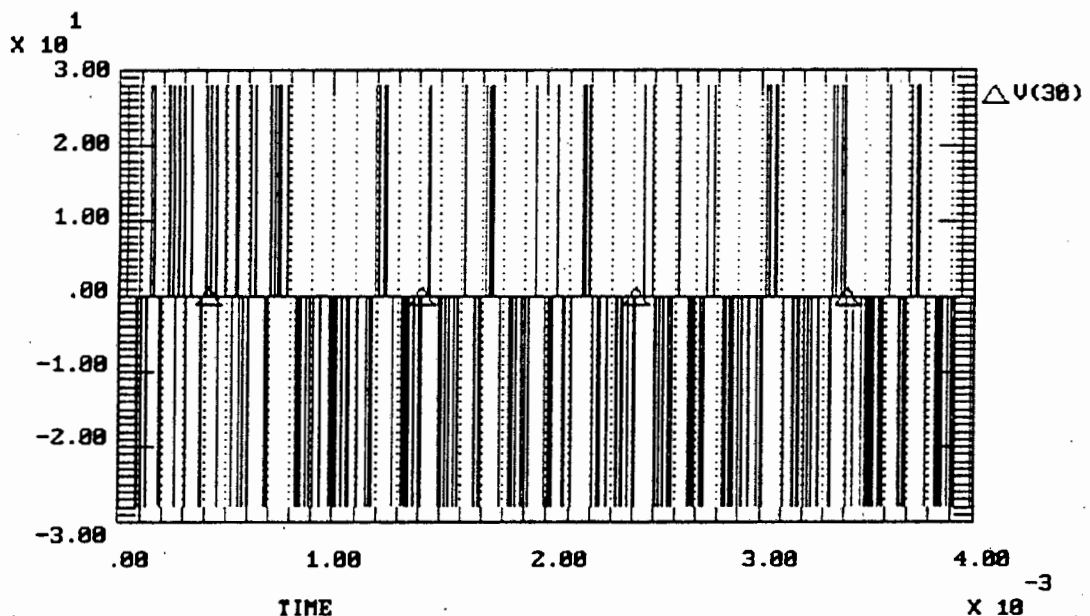


Figure 4.4: Impulses Stimulate the Bandpass Filter. Lock can be determined by the impulses going all positive or all negative. This condition is met at 0.8 milliseconds. The positive going impulses are due to the instability of the loop.

To minimize the loss of lock, the loop filter of the phase-locked loop would need to be narrowed after achieving lock. The primary reason for not doing this before lock is achieved, is a trade-off between locking time and stability. The instability ensures that the frequency will change more rapidly, and only stabilize once lock is achieved. Figure 4.5 compares the transmit and locally generated sequence for the first few milliseconds and the last few milliseconds. It is clear from the plot that the sequences are synchronised.

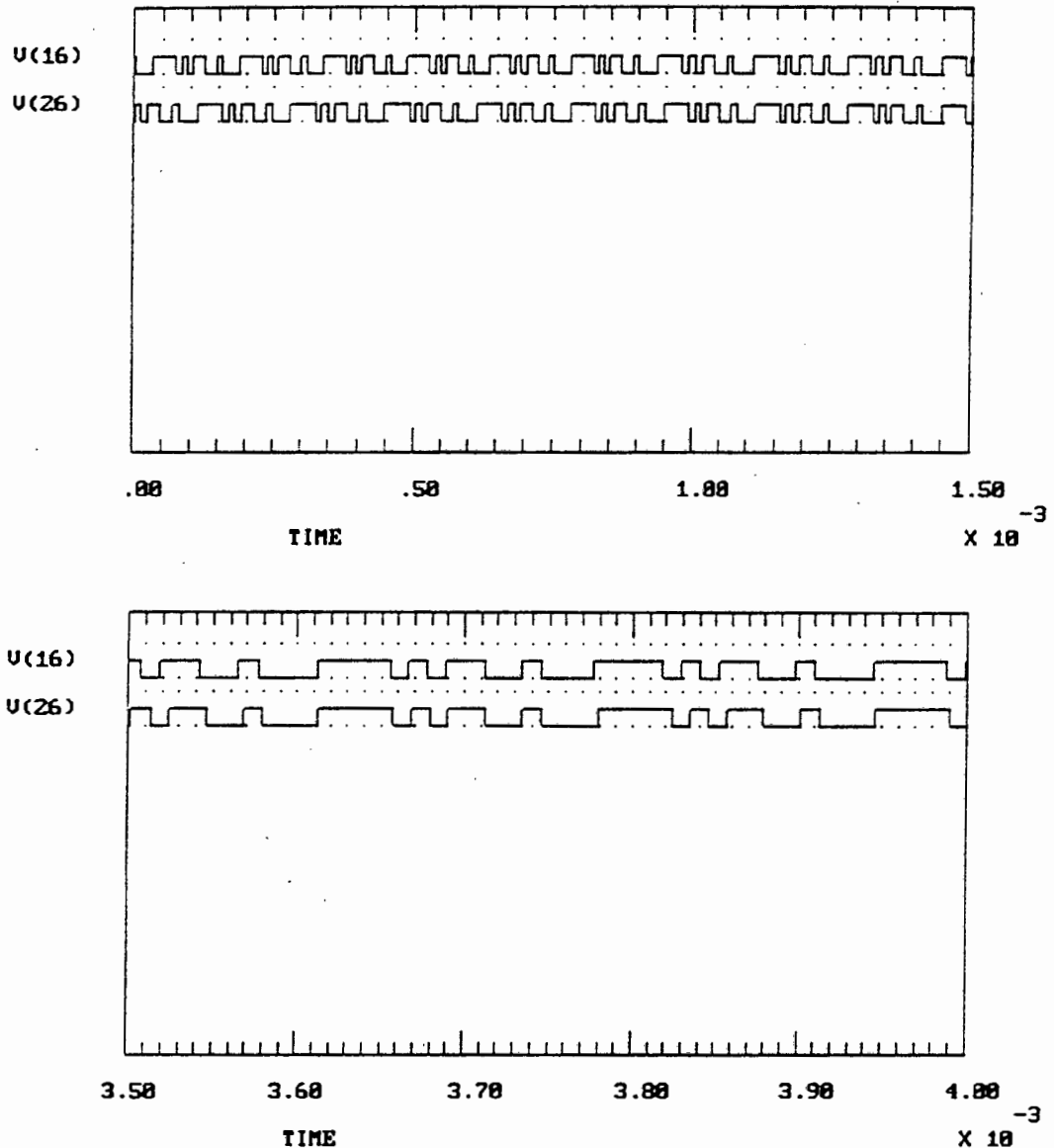


Figure 4.5: Comparison between Transmitted and Locally Generated PN Sequences. The figures illustrate the convergence of the locally generated PN sequence to the received PN sequence.

The VCO input signal gives a clear indication of how the VCO frequency will change with time. For the first few milli-seconds the error signal varies dramatically. After lock, the signal settles around a negative DC value, except when the loop loses lock and has to resynchronise. This is shown in figure 4.6

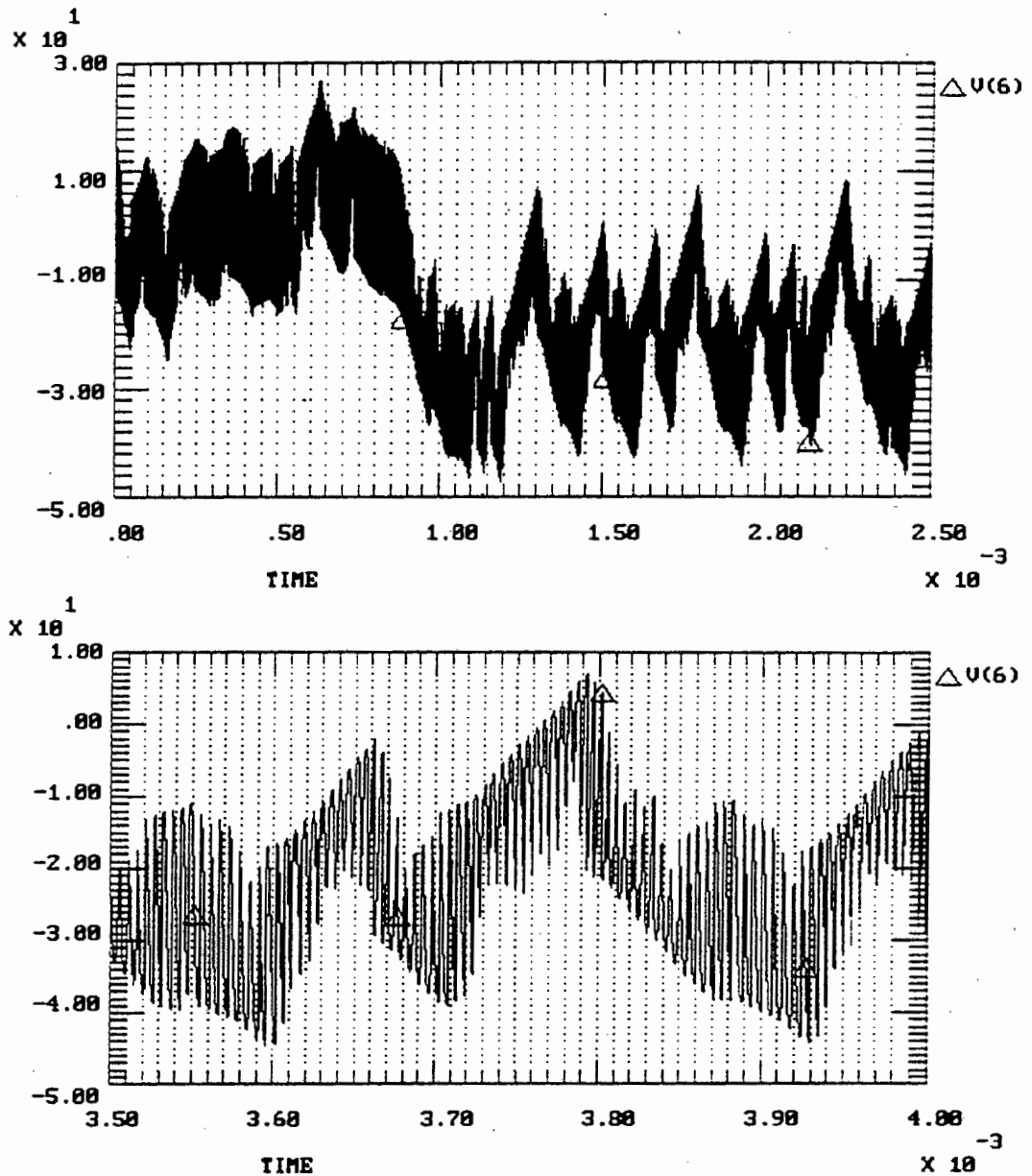


Figure 4.6: VCO Input Signal. From the VCO input signal, the convergence towards lock is clearly illustrated. The jitter in the system is identified by the moving away from the steady state value. The upper plot shows a 2.5 millisecond time segment. The loss of lock is clearly illustrated. The lower plot shows a 0.5 millisecond time segment, showing more detail of how the VCO drive signal changes when the system loses and regains lock.

4.2.4 Change in parameters

Changing simulation parameters, like filter bandwidths and sampling time had a dramatic effect on the overall performance. By increasing the sampling rate slightly, the loop performance was improved. Figure 4.7 shows the output at node 30 (output of first multiplier). From this figure it is clear the system is more stable and that the number of slippages is reduced dramatically.

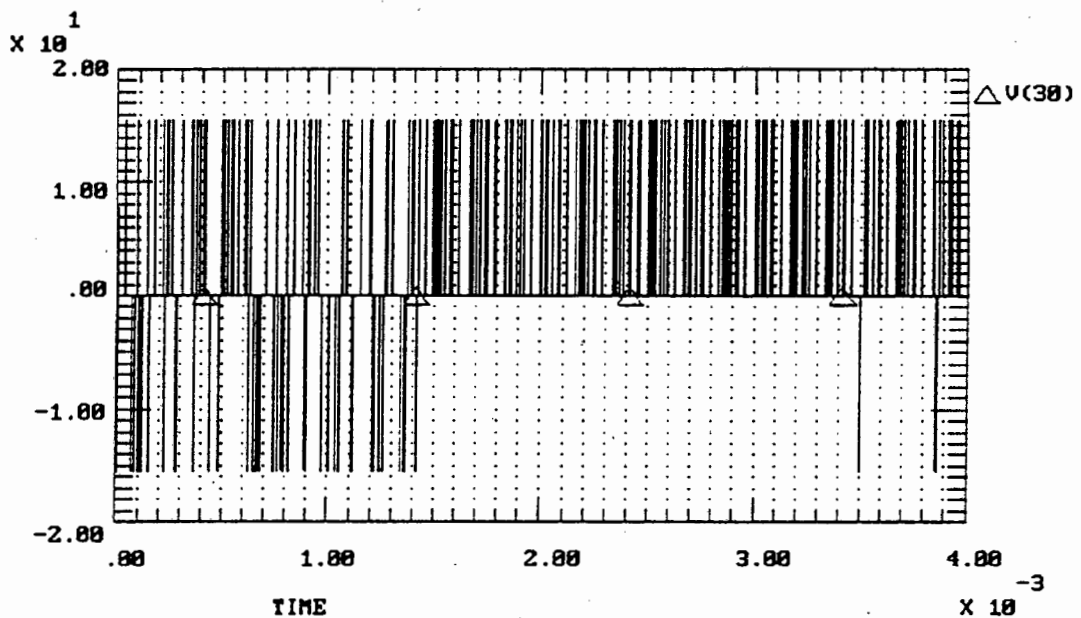


Figure 4.7: Train of Impulses Stimulate Bandpass Filter. Lock was achieved in 1.4 milliseconds. The jitter present in the previous case has been reduced.

However in other cases when the system parameters were changed the system degraded dramatically and did not achieve lock. This was attributed to the instability created by the differentiation block being included in the feedback loop.

4.2.5 Practical Implementation

To evaluate the anomalies in the simulation results, a practical system was built. In the practical system, the anomalies were not present, but a false-lock condition occurred occasionally. It was found that adjusting the transmitter frequency slightly caused the system to overcome the meta-stable condition, and the system achieved lock. The reason for the meta-stable condition is that, for certain combinations of code-phase and frequency, short

lengths of impulses of either ones or zeros are generated. On examination of the drive signal into the bandpass filter, the number of positive and negative spikes were approximately equal and therefore, the frequency of the synchronous oscillator remains around a fixed point.

In the cases where the system did achieve correct lock, the system ran for 24 hours without losing lock. Chapter 9 gives basic results of the practical implementation of method one.

4.3 Discussion of Method One

The analysis of the novel method presented in this chapter uses a time-variant spectral analysis of the resultant impulses after the first multiplier to create the evaluation curve of the grid structure as shown in chapter 2, figure 2.5. Figure 4.8 shows a filtered version of figure 4.6, using a moving average filter, of the VCO input signal. The curve shown in figure 4.6 is dependent on the phase-offset between the transmitted and receiver generated sequence and the frequency difference between the transmitted sequence clock and the centre frequency of the voltage controlled oscillator.

The y-axis (voltage) is an indication of the change in frequency, which is proportional to the input error voltage. An increase in voltage implies an increase in frequency and a decrease implies a decrease in frequency from the centre frequency. In the cases presented thus far, the transmitted sequence clock was below the VCO centre frequency. From the figure it can be seen that the frequency of the VCO was decreased and increased as the phase-relationship between the two sequences changed, implying a change in the VCO frequency. Once the correct phase-frequency cell was determined the VCO settled to a negative DC value. A negative value will always result if the transmitted sequence frequency is below the the VCO centre frequency.

When the sequences became unsynchronised, around 1.3 milliseconds, the VCO drive voltage increased. but returned quickly to the DC value as lock was re-acquired.

Figure 4.9 shows the filtered version of the VCO drive signal for the second, more stable, simulation. For the first millisecond the same discussion as for figure 4.8 holds true. After 1.4 milliseconds the system is synchronised and the control voltage settles around a DC value. Since the jitter in the system is low, sudden loss of synchronisation does not occur as in the previous case.

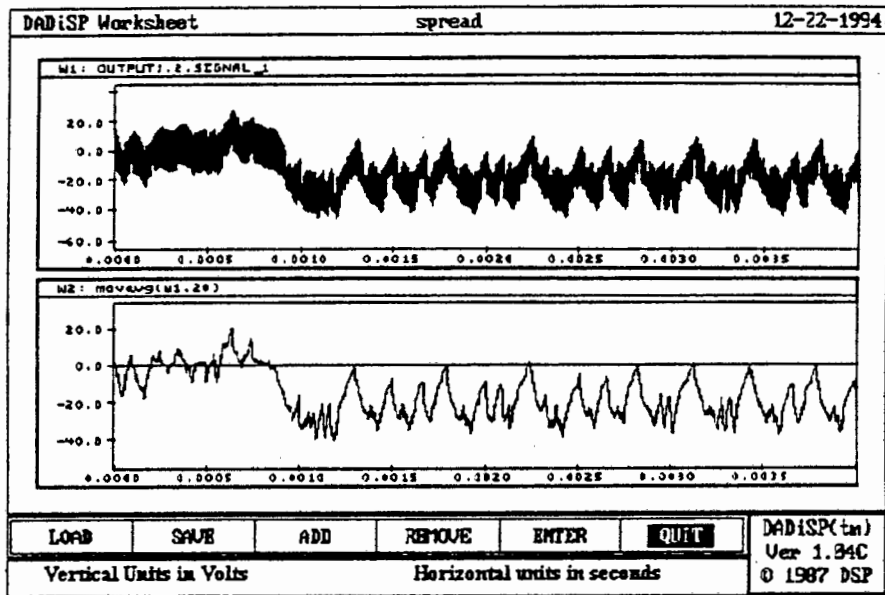


Figure 4.8: VCO Input Signal (simulation 1). The VCO signal has been filtered using a moving average filter. The jitter is identified by the deviation from the steady state value.

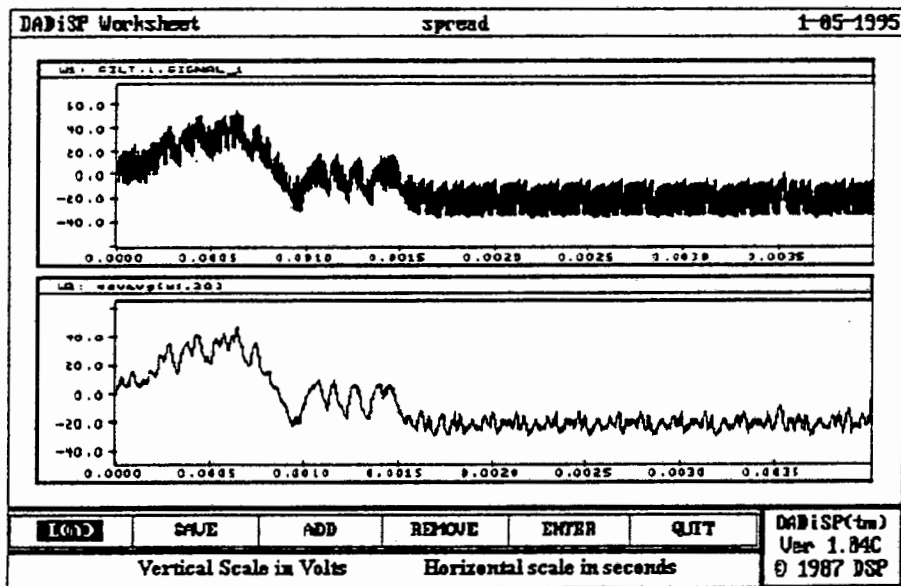


Figure 4.9: VCO Input Signal (simulation 2). The VCO signal has been filtered using a moving average filter. The network is stable with no jitter present.

4.3.1 Short-comings of Method One

From both the simulation and practical results it is clear that the implementation has a number of short comings namely,

- Parametric Sensitivity.
- Suffers from false lock.
- Inherently unstable.

The question arises as to why these problems occurred. The input to the band-pass filter (node 30, figure 4.2 is directly related to the frequency of the VCO. The received signal simply changes the sign of the differentiated signal to prevent the system from achieving lock unless the two sequence are aligned. Since the relationship between impulses is determined by the phase-locked loop VCO frequency, the phase-locked loop 'chases' itself. This in turn implies that, unless the system is designed for a critical point, lock will never be achieved. Although this was possible, the loop suffered from jitter and lost lock every few milliseconds.

4.4 Comparison with Existing Methods

The comparison of different methods of synchronisation, when the methods are fundamentally different, makes comparison complex, often requiring a number of assumptions to be made. In the case of the synchronisation methods presented here. this certainly holds true.

4.4.1 Structure

Present spread spectrum systems require the two pronged approach to synchronisation, i.e acquisition and tracking, which implies that at least two synchronisation networks are required. For comparison it is assumed that the system is implemented at base-band and no data is present (i.e. the systems are data sensitive.) In table 4.2 a standard synchronisation system, consisting of a full-time early-late tracking loop (figure 2.13), a discrete acquisition loop (figure 2.6) and a fixed integration time energy detector (figure 2.7), is compared to RASE and the novel synchroniser. In the case of RASE, the elements have been divided into those that are for acquisition and those that are for tracking. In the case of the novel synchroniser the elements have been listed under tracking, since it performs both tasks.

Element/s	Standard	RASE	NOVEL
ACQUISITION:			
Filters	1	2	
Decision Logic Block	1		
Reference Waveform Generator	1		
Multipliers	1	1	
Examination Period Generator		1	
Gate		1	
n-Bit Counter		1	
Threshold Detector		1	
Limiter		1	
Delay Element		1	
Energy Detector:			
Filters	2		
Squarer	1		
Integrate and Dump	1		
Adjustable Timing Block	1		
Comparator	1		
TRACKING/COMBINATION:			
Multipliers	2	2	2
Power Dividers	1	1	
Filters	1	1	2
Summer	1	1	
Oscillators	1	1	1
PN Generator	1		1
Differentiator			1
Discrete PN Generator		1	
Comparator			1
Limiter			1
TOTAL:	17	16	9

Table 4.2: Comparison of Typical Building Blocks for Synchronisation Circuitry

From the number of elements required for each of the implementations it is clear that

the novel synchronisation network has a lower component count. It is therefore more cost effective and given the fact that there are few elements that need 'fine tuning' it is relatively easier to implement than a standard spread spectrum acquisition and tracking system

4.4.2 Locking time

Locking time is an important issue in any design of a communication system. In spread spectrum system, locking times are often long and often require more complex hardware to improve on these locking times. The analysis of locking time is dependent on the implementation and therefore any real comparison is near impossible without a number of assumptions.

Ziemer [23] and Dixon [10] presents some basic results on the calculation of the mean synchronisation time of various acquisition networks. In the analysis however, it is assumed that the spreading sequence clock frequency has already been determined. Using the assumptions given in Ziemer for the mean acquisition time for a serial search network at high signal-to-noise ratios, the mean synchronisation time is:

$$\bar{T}_t = LT_e + T_c \quad (4.2)$$

where T_e is the time taken to determine if a cell is correct or not.

For short sequences, T_e in equation 4.2 is usually one period of the sequence, i.e. LT_c . The simulation results presented were based on 15 bit maximal length sequence, with a chipping rate of 90 kHz. To make the comparison, the same parameters are assumed. Therefore $\bar{T}_t = L^2T_c + T_c = 2.5$ milliseconds. This value excludes the time taken to synchronise the chipping frequency and the false alarm penalty time.

For a RASE system the mean synchronisation time is

$$\bar{T}_t = \log_2(L + 1)T_c + T_e \quad (4.3)$$

T_e for RASE is the same as for the serial search technique. Therefore for a 15 bit code, the mean synchronisation time is $\bar{T}_t = \log_2(15 + 1)T_c + LT_c = 0.2$ milliseconds.

From figure 4.4 the synchronisation time is 1.0 milliseconds and for figure 4.7 1.4 milliseconds. It should be noted that these values include the time taken to acquire the chipping

frequency. From these results it is clear that RASE performs best. The novel method presented in this chapter performs better than the serial search technique.

4.5 Summary

In this chapter a novel synchronisation network was presented. This network is simpler, more cost effective and is easy to implement. It was shown that the given the problems, the system did work both in simulation and in practice. A number of short-comings were identified. *In the next chapter a more stable and robust network is presented.* The overall network is similar to the novel synchronisation network presented in this chapter.

Chapter 5

Novel Synchronisation Technique-Method Two

In chapter 4 a novel synchronisation circuit that does acquisition and tracking using a simple solution was presented and discussed with the aid of simulation and practical results. From these results a number of short-comings of the implementation were identified, especially the dependency on circuit parameters and the unstable nature of the network. The problems identified were attributed to the inclusion of the differentiation block in the feed-back loop.

In this chapter, a new synchronisation network is presented that is robust, insensitive to simulation parameters and yet does not do away with the simplicity of the previous circuit. The uniqueness of the network in relation to present synchronisation spread spectrum methods will be highlighted and discussed in detail.

5.1 Improving Stability

To improve the stability of the synchronisation network presented in the previous chapter it is necessary to remove the differentiation block from the feedback loop, and to place the differentiator block in the receive path. The immediate implication of this change is that frequency stability within the feedback structure is improved. What this means is that the time relationship from one impulse to the next is deterministic and is a function of the spreading sequence and transmitter generated chipping frequency only.

The change in sign of an impulse is however dependent on the VCO clock frequency and

locally generated sequence. It is important to highlight the fact that the locally generated pseudo-noise sequence is still dependent on the frequency of the clock generator. The sign change of the impulse is therefore dependent on:

- The sign of the current impulse.
- The sign of the current locally generated chip , which is in turn related to the frequency of the VCO.

5.2 System Description

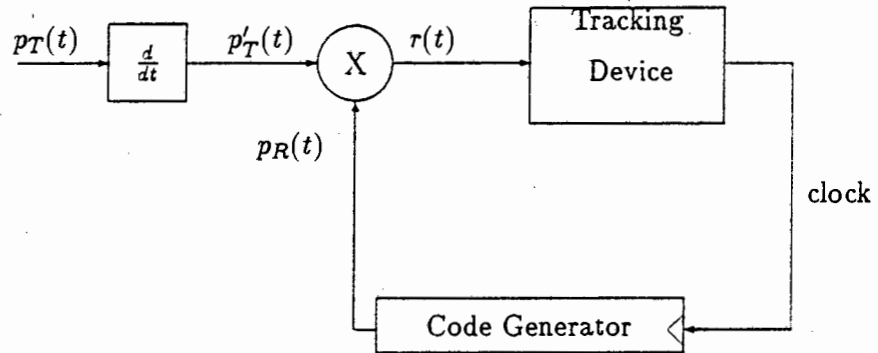


Figure 5.1: Novel Spread Spectrum Synchroniser

Figure 5.1 shows the new spread spectrum synchronisation network [40]. The output of the multiplier $r(t)$ is given by equation 3.8, except for one fundamental change. The frequency of the locally generated code is a function of the frequency of the clock generator, which is in turn a function of the driving signal of impulses. Equation 3.8 therefore becomes,

$$\begin{aligned}
 r(t) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} b_{mod(i,L)} a_{mod(j,L)} \begin{bmatrix} u(t - iT_v(t))\delta(t - jT_c) \\ -u(t - (i+1)T_v(t))\delta(t - jT_c) \\ -u(t - iT_v(t))\delta(t - (i+j)T_c) \\ +u(t - (i+1)T_v(t))\delta(t - (j+1)T_c) \end{bmatrix} \\
 &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} b_{mod(i,L)} a_{mod(j,L)} \begin{bmatrix} u(jT_c - iT_v(jT_c))\delta(t - jT_c) \\ -u(jT_c - (i+1)T_v(jT_c))\delta(t - jT_c) \\ -u((j+1)T_v(t) - iT_v((j+1)T_c))\delta(t - (j+1)T_c) \\ +((i+1)T_c - (i+1)T_v((j+1)T_c))\delta(t - (j+1)T_c) \end{bmatrix}
 \end{aligned} \tag{5.1}$$

where $\frac{1}{T_v(t)}$ is the clock frequency at time t , a_n is the transmitted sequence co-efficients and b_n the receiver generated sequence co-efficients. With the introduction of time dependency in equation 3.8, equation 5.1 is non-linear. This makes the analysis complex. To analyse the system three approaches were taken namely,

- Tesla Simulation.
- A mathematical model was created and analysed using non-linear theory.
- Practical Implementation.

5.3 Tesla Simulation

To show the benefit the changes made to the previous implementation, a number of extensive Tesla simulations were executed. Tesla gives the ability to determine the sensitivity of the implementation to parameter variation, and also to simulate various other conditions.

5.3.1 Simulation Diagram

As for the network presented in chapter 4. the block diagram in figure 5.1 needs to be converted into a practical implementation. The clock generator was designed as a second order phase-locked loop. A bandpass filter was placed before the input of the phase-locked loop. The purpose of this was to create exponentially decaying sinusoidal signals, to compare with the locally generated sinusoidal signal. The differentiator was implemented using a Tesla differentiation block. The simulation block diagram is given in figure 5.2, with the circuit node numbers as indicated.

5.3.2 Simulation Parameters

The novel synchronisation network was simulated for a number of sequence lengths and offsets. The basic parameters are given in table 5.1.

Parameter	Value
Sequence Length	15
Code Offset	various
Tx Clock Frequency	various
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz

Table 5.1: Simulation Parameters - Method 2

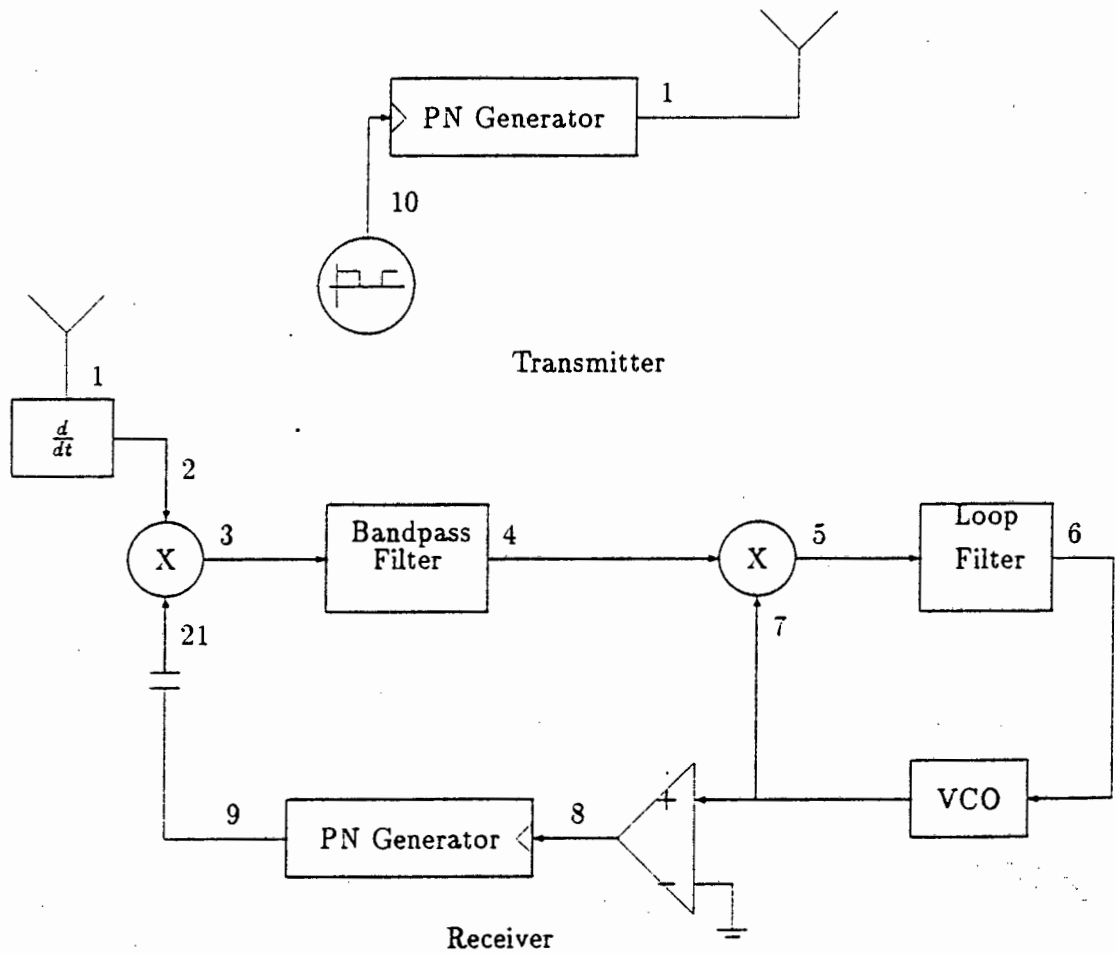


Figure 5.2: Simulation block diagram

5.3.3 Results

For each of the simulations a huge amount of information was obtained on the performance and characteristics of the novel synchronisation network. As has previously been mentioned, the output at node 3 gives a clear indication of whether the system has achieved and maintained synchronisation or not. From figure 5.3 it is clear that the system has achieved lock after 6 milliseconds. It can be seen that once lock is achieved, it is maintained.

Figure 5.4 shows the first few milliseconds of the VCO drive signal before lock. The VCO frequency is increased and decreased as indicated by a positive or negative slope respectively. The change in the drive signal voltage is not as erratic as the previous implementation. This more regular pattern can be attributed to the differentiator block being removed from the feedback loop. Therefore the time-relationship between impulses is deterministic. The change in direction of the impulses is however dependent on the frequency of the VCO and the locally generated sequence output chip.

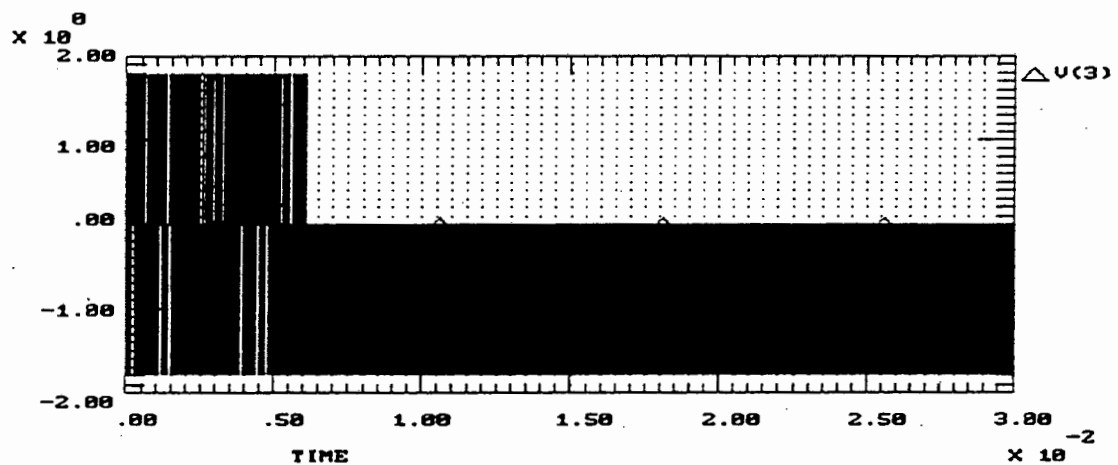


Figure 5.3: The Product of the Signed Edge Detected Received PN Sequence and Locally generated PN sequence. Lock was achieved in 6 milliseconds. The system is clearly stable.

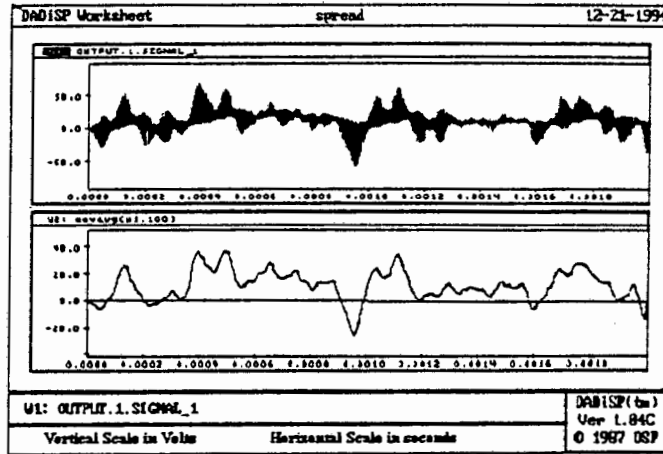


Figure 5.4: VCO Drive Signal Before Lock. The change in the VCO's output frequency is directly proportional to the change in voltage of the error signal.

Figure 5.5 shows the relationship between the transmitted sequence (node 1) and the locally generated receiver sequence (node 9). The plot shows the phase-relationship just before lock (6 milliseconds). After 6 milliseconds the two sequence are aligned.

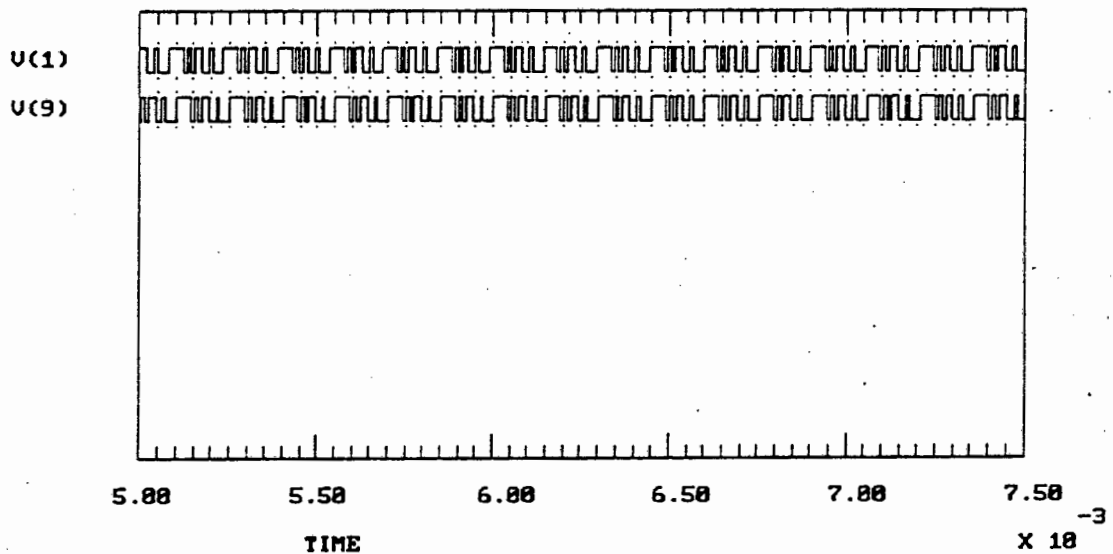


Figure 5.5: Comparison between Transmitted and Receiver Generated Sequence. The plot illustrates the convergence of the transmitted and receiver generated PN sequences.

Under certain circumstances positive impulse lock is achieved, i.e. the spikes at the output of the first multiplier (node 3) are positive rather than negative. The reason for this is the alignment of the edges when the system achieves lock. For the case of positive impulse lock

the sequences lock with a one chip offset. This would imply that under normal correlation conditions, the system would appear uncorrelated. Figure 5.6 shows the output of the first multiplier (node 3). Figure 5.7 show the transmitted and locally generated receiver sequences. It is clear that after the system locks, a one chip offset is present. (In the initial implementation this condition is detected by the use of comparators and digital counters. ([41] and chapter 8)). The offset can be dealt with by the inclusion of a delay element when the sequences are despread.

Taking a closer look at figures 5.5 and 5.7, an interesting effect is noted. The direction from which the locally generated sequence approaches the transmitted sequence, will determine if the lock is positive or negative. For positive impulse lock the phase of the received sequence is behind that of the locally generated sequence, while for negative impulse lock the opposite is true.

The locking time is dependent on a number of factors, namely the frequency difference between the transmitter clock and the receiver's VCO centre frequency, the phase difference between the transmitter and receiver sequences and whether the locally generated sequence leads or lags the received sequence. A number of simulations were run taking these factors into account. Figure 5.8 is a three dimensional plot of phase offset versus frequency difference versus locking time.

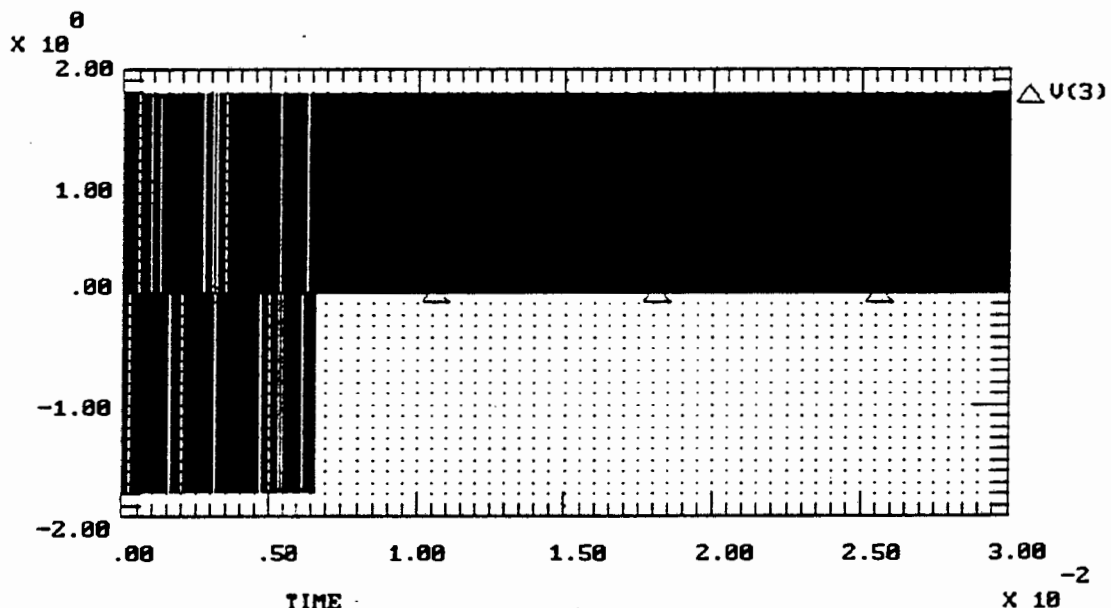


Figure 5.6: The Product of the Signed Edge Detected Received PN Sequence and Locally generated PN sequence. Positive impulse lock was achieved in 6.1 milliseconds. The system is clearly stable.

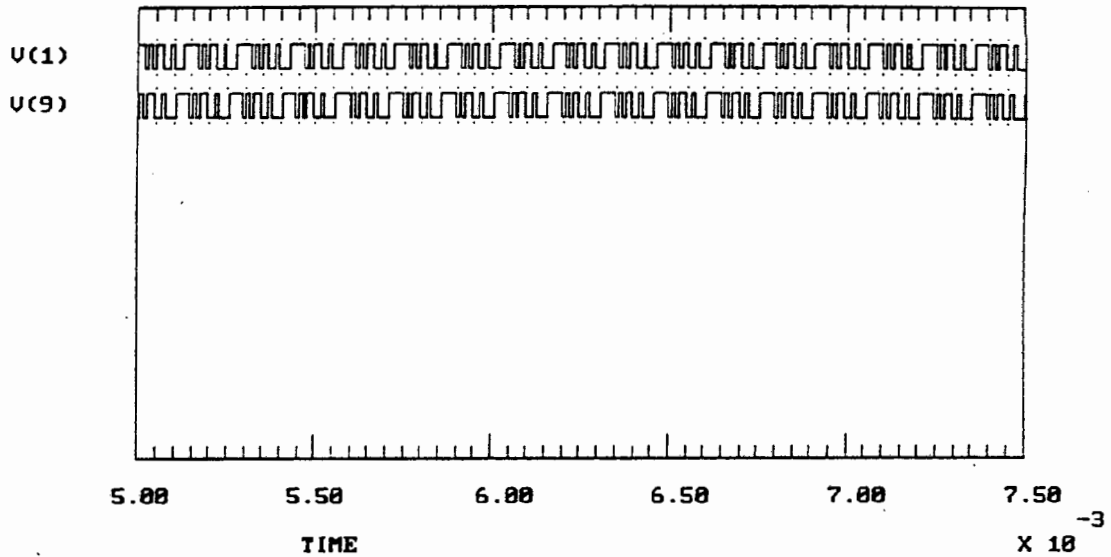


Figure 5.7: Comparison between Transmitted and Receiver Generated Sequence. The plot illustrates the convergence of the transmitted and receiver generated PN sequences.

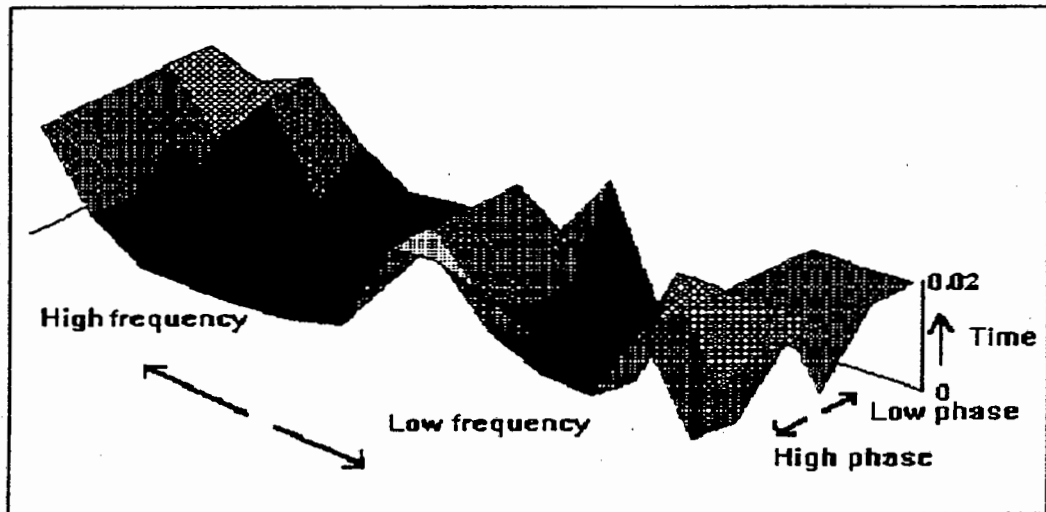


Figure 5.8: Three Dimensional Locking Surface. The locking times form a valley around the centre frequency. The time are also mirrored around the centre frequency and code-phase offsets.

The locking surface of figure 5.8 reveals some interesting trends. The locking times form a valley around the centre frequency. The frequency difference was extended until lock was

not achieved within a few milliseconds. It was found that the spikes were generally negative going for a transmitter frequency below the VCO centre frequency and positive going for frequencies above the VCO centre frequency. The locking times are also mirrored around the centre frequency and code-phase offsets.

Very near,(and at) the centre frequency the system does not lock within the time constraint. The primary reason for this is that it is more difficult to shift the phase-locked loop centre frequency when the input frequency is very near the centre frequency. Figure 5.9 shows the input to the VCO. The transmitter and VCO have the same centre frequency, but with a phase difference of 5 degrees. The sequences are offset by seven chips. The fact that the signal oscillates around zero is correct for a system that is in lock, however since there is an offset, the impulses into the system are not all positive, or all negative. By examining the input to the bandpass filter, figure 5.10, this can be clearly seen. Note the periodic nature of the impulses. This implies that the system is in a metastable condition.

It was found that by including a larger phase difference between the VCO and transmitter clocks, the system did achieve lock. This would be expected. A phase of 5 degrees translates to a small voltage offset. As the phase offset is increased so the likelihood of shifting the VCO frequency increases.

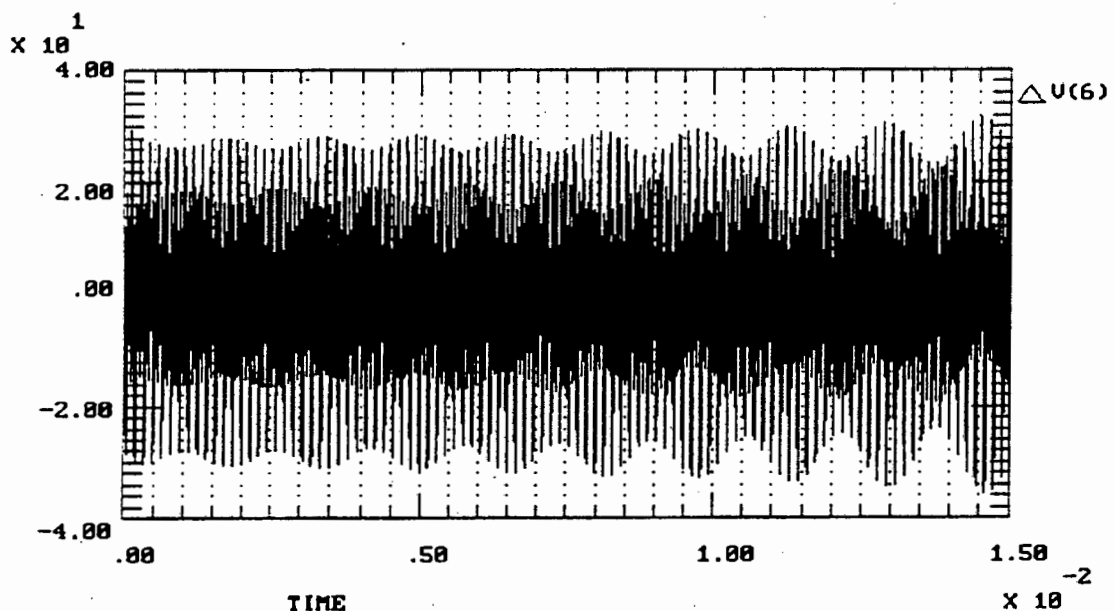


Figure 5.9: VCO Drive Signal. The frequency is not shifted when the centre frequency of the VCO and the Transmitter have the same starting frequency and phase.

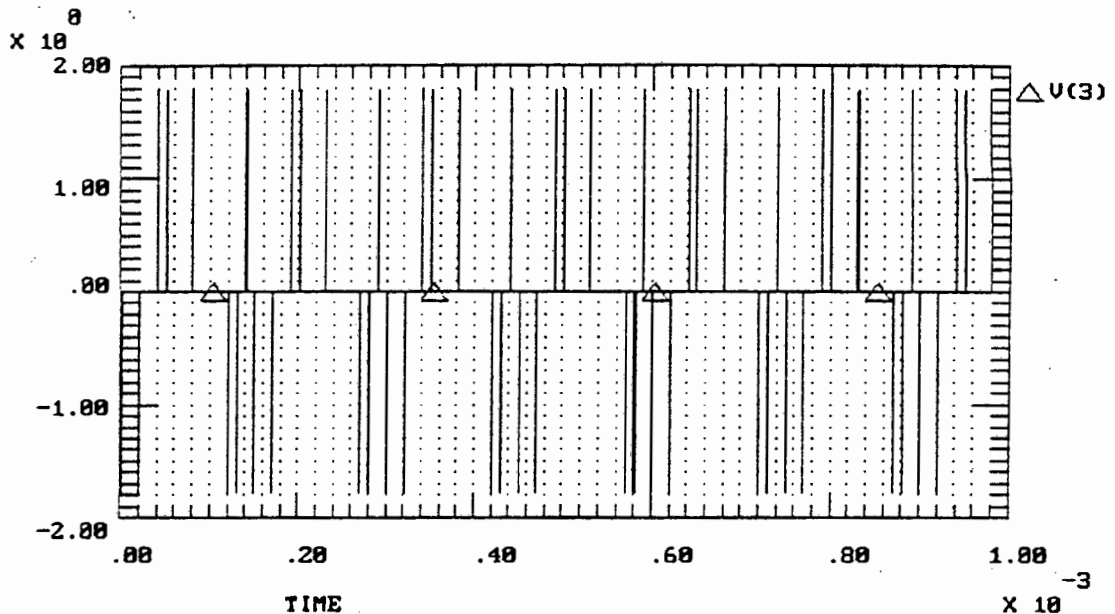


Figure 5.10: Bandpass Filter Input. The impulse train appears to be periodic.

5.3.4 Practical Implementation

To determine the validity of the simulation results, a practical system was built. The system was built using a second-order phase-locked loop as opposed to the synchronous oscillator used for method one (see chapter 9). The bandpass filter was implemented as a second order Butterworth filter. The practical system did not suffer from false-lock as was experienced with the first implementation. The practical system was evaluated with a code sequence up to a length of 127. The system ran for 24 hours without losing lock. Chapter 9 gives an overview of the practical implementation of method two.

5.4 Discussion of Method Two

5.4.1 Comparison Between Method One and Method Two

Fundamentally one would expect method one and two to perform equally well, but from the results presented this is not the case. For method one a number short-comings were given, relating to the overall stability of the network and the sensitivity to simulation parameters. In the case of method two, these short-comings were not present. The simulation parameters of the system were changed as was done for method one. The system was insensitive to

the changes and the system performed as would be expected. The synchronisation network presented in this chapter is therefore stable and robust.

5.5 Comparison with Existing Methods

5.5.1 Evaluation of Code Phase-Frequency Grid

As in the case of method one, the novel method presented in this chapter uses a time-variant spectral analysis of the resultant impulses after the first multiplier to create the evaluation curve of the grid structure as shown in chapter 2, figure 2.5. The curve shown in figure 5.4 is dependent on the phase-offset between the transmitter and receiver generated sequence and the frequency difference between the transmitted sequence clock and the centre frequency of the voltage controlled oscillator. As was the case for method one, the VCO signal will settle around a negative value when the VCO centre frequency is above the transmitted sequence's clock frequency. In the opposite case, the VCO drive voltage will settle around a positive value.

The method by which the code phase-frequency grid structure in chapter 2, figure 2.5 is evaluated is random, whereas the method of the systems presented in chapter 2 are orderly. The primary benefit of this is that lock should be achieved in some cases more quickly, while in other cases more slowly.

5.5.2 Structure

Present spread spectrum systems require the two pronged approach to synchronisation. i.e acquisition and tracking, which implies that at least two synchronisation networks are required. The acquisition methods presented in chapter 2 required the definition of the search waveform for the grid structure. Although the two pronged approach is a good practical solution, the cost of such an implementation is expensive, requires more hardware and is therefore more complex. The novel network presented in this chapter uses only one synchronisation network to perform both acquisition and tracking. The component count is low and the system is cost effective. (See section 4.4.1 for a complete comparison.)

5.5.3 Doppler

Standard spread spectrum acquisition systems and RASE do not track the frequency of the incoming sequence and therefore any Doppler effects will increase the acquisition time of the system. The tracking of spreading sequence frequency only occurs when the synchronisation system is in the tracking mode. The novel synchronisation network however does cater for Doppler. For the system to achieve lock both the spreading sequence frequency and the code-phase must be correct. Since the time-relationship between the impulses from the differentiator are determined by the incoming spreading wave-form clock frequency, the novel network will always attempt to track that frequency and therefore is not affected by Doppler substantially. In practical experiments the frequency was shifted to prove that the system coped with Doppler effects.

5.5.4 Locking time

From the surface give in figure 5.8 the locking times of the system vary depending on the starting conditions of the original system. The results presented for both serial search and RASE did not take into account the false alarm penalty and the time to acquire the transmitted sequence clock frequency. Given a 15 bit PN sequence and a chipping frequency of 99 kHz, the following locking times can be calculated.

The mean synchronisation time for serial search techniques using equation 4.2 is:

$$\bar{T}_t = L^2 T_c + T_c = 2.3 \text{ milliseconds.}$$

For a RASE system the mean synchronisation time is:

$$\bar{T}_t = \log_2(15 + 1)T_c + LT_c = 0.2 \text{ milliseconds.}$$

From figure 5.6 the synchronisation time is 6 milliseconds and for figure 5.3 6 milliseconds. Before comparing to RASE and other serial search techniques, consider table 5.2. Table 5.2 gives locking times, in milliseconds, for various phase offsets (horizontal) and frequency offsets (vertical). From the results it is clear that for certain conditions the performance is 'worse' than both RASE and serial search techniques, but under other conditions the novel method performs the same or better.

It must be stressed again that the comparison is subjective because of the exclusion of the chipping frequency recovery and false alarm penalties in the serial search and RASE

techniques locking time formulae.

	-7	-5	-3	5	7
$f_c + 6$	16.30	9.90	2.80	7.40	7.50
$f_c + 5$	20.00	20.00	3.80	20.00	1.56
$f_c + 4$	11.60	5.10	3.80	2.15	0.66
$f_c + 3$	6.15	5.70	5.25	2.45	0.84
$f_c + 2$	7.00	8.50	6.50	3.60	1.15
$f_c + 1$	7.10	12.10	9.10	6.20	3.40
$f_c - 1$	8.40	20.00	20.00	20.00	20.00
$f_c - 2$	20.00	4.30	5.70	10.50	20.00
$f_c - 3$	0.36	2.25	4.40	7.00	10.6
$f_c - 4$	0.26	1.52	2.80	5.2	6.36
$f_c - 5$	6.80	1.20	20.00	4.50	5.91

Table 5.2: Locking Time Results - Method 2. The locking times (ms) are dependent on the code-phase offsets (columns) and frequency (kHz) offsets (rows).

5.6 Summary

In this chapter a novel synchronisation network was presented. This network is simpler and more cost effective than the methods discussed in chapter 2. It was shown that the system worked both in simulation and in practice, and that method two is stable and robust. *In the next chapter, the theoretical aspects of this novel synchronisation method (method two) will be presented in detail.*

Chapter 6

Theoretical Analysis of Method Two (1)

In the preceding chapters simulation and practical implementations have been used to study the novel synchronisation networks presented in this dissertation. Unfortunately block simulation packages, like Tesla, and practical implementations do not give a complete understanding of the behaviour of the network or allow one the flexibility of mathematical models.

In this chapter the novel synchronisation network in figure 6.1 which was presented in the previous chapter will be analysed using non-linear techniques [40]. To aid the analysis, phase-locked loop and phase-plane basics will be discussed. The analysis of the network will be in three stages, namely the non-linear analysis of :

- Phase-locked loop.
- Clock recovery network.
- Overall novel synchronisation network

In the analysis of synchronisation networks, phase-plane techniques are a popular choice. Although phase-plane techniques are classified as graphical techniques, a fair amount of analysis is required to obtain an equation, or set of equations that are in the form required to apply phase-plane methods.

Phase-Plane techniques make use of the method of isoclines. The method of isoclines however is only suited to first order nonlinear equations of the form,

$$\frac{dy}{dt} = f(x, y) \quad (6.1)$$

The method of isoclines can however be extended to second order systems of the form,

$$\ddot{x} + F(x, \dot{x}) = 0 \quad (6.2)$$

By substituting $y = \dot{x}$, equation 6.3 is obtained.

$$y \frac{dy}{dx} = \frac{dx}{dt} \frac{dy}{dx} = \frac{dy}{dt} = \ddot{x} \quad (6.3)$$

which reduces to

$$\frac{dy}{dx} = -\frac{F(x, y)}{y} \quad (6.4)$$

To plot the phase-plane graph, it is necessary to calculate the trajectories at any given point in the plane defined by x and \dot{x} , where x is the abscissa and \dot{x} the ordinate. For large \dot{x} the trajectories are nearly sinusoidal. For $x = 0$ the ordinate is the -1 isocline. For other values, the isoclines for other slopes are not zero, and are generally not straight lines. Finding isoclines is a tedious process. (Refer to [43, ch1.10, pg 61] for details on calculating isoclines). However with numerical techniques and computers it is possible to avoid the method of isoclines completely.

From phase-plane plots it is possible to determine if a system is stable or unstable. Four equilibrium points exist, namely a saddle (figure 6.2a), a node (figure 6.2b), a focus (figure 6.2c) and a centre (figure 6.2d). Of the four a node and a focus are stable, while a saddle is unstable, and a centre conditionally stable.

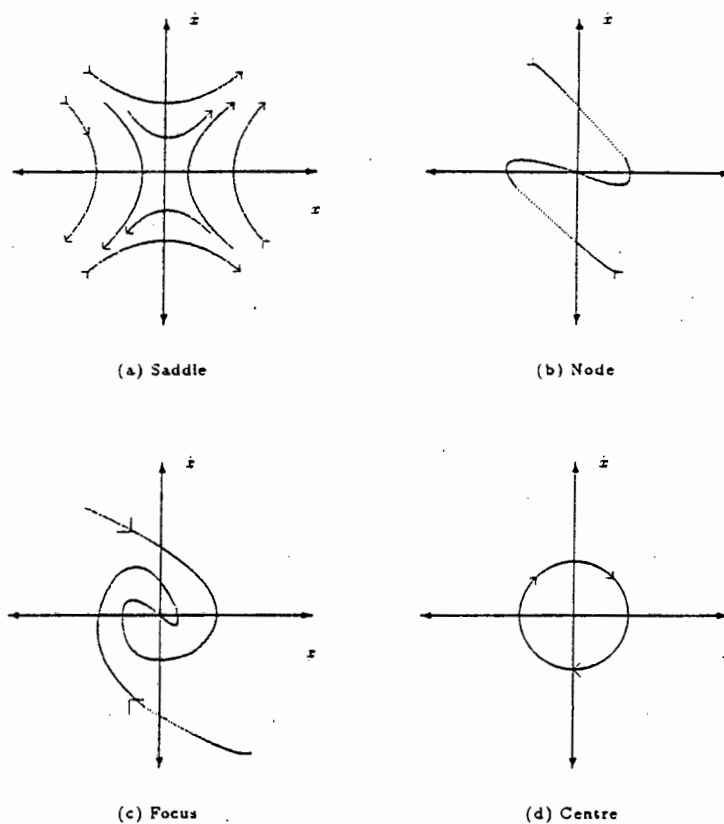


Figure 6.2: Equilibria

6.2 Stage One: Phase-Locked Loops

The analysis of phase-locked loops is a well researched area, with many papers presented on the various attributes of phase-locked loops under a variety of conditions [44][45]. The method of analysis varies from one paper to another, with the linearised approach being a popular choice. Viterbi did much research on the analysis of phase-locked loops and made use of phase-plane and Fokker-Plank techniques to describe the behaviour of various order loops.

6.2.1 Phase-Locked Loop Fundamentals

The basic phase-locked loop consists of three main elements, namely a loop filter (or time-invariant linear filter), a voltage controlled oscillator (VCO) and a multiplier. These can be

schematically represented as shown in figure 6.3. Assuming that the received power is A^2 and therefore the peak voltage is $\sqrt{2}A$, the received signal can be written as:

$$\sqrt{2}A \sin(\theta_1(t)) \tag{6.5}$$

and the VCO output signal as:

$$\sqrt{2}K_v \cos(\theta_2(t)) \tag{6.6}$$

where K_v is its root-mean square amplitude. If the error signal to the VCO, $e(t)$, were to be removed, the VCO would oscillate at a fixed frequency, which shall be referred to as ω_{vco} . When the control signal is applied the VCO frequency becomes $\omega_{vco} + K_{vco}e(t)$, where K_{vco} is a proportionality constant with dimensions of radians per second per volt. Therefore the time-derivative of the phase-plane angle of the VCO-output phase is

$$\frac{d\theta_2(t)}{dt} = \omega_{vco} + K_{vco}e(t) \tag{6.7}$$

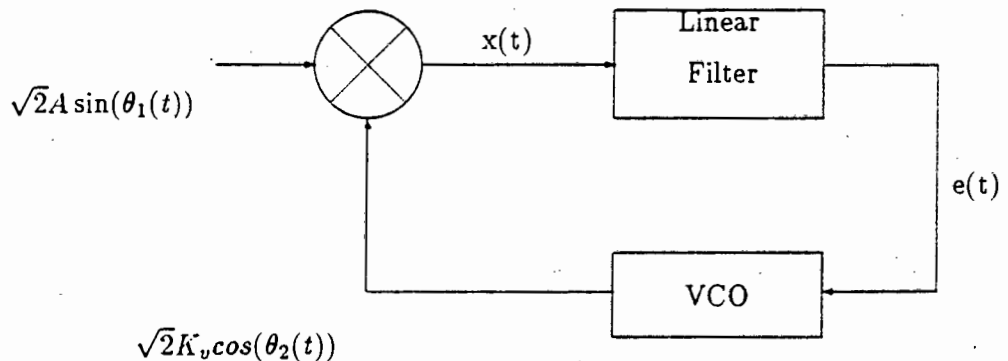


Figure 6.3: Phase-Locked Loop

Equations 6.5 and 6.6 are multiplied and simplified using standard trigometric identities to give

$$x(t) = AK_v [\sin(\theta_1(t) - \theta_2(t)) + \sin(\theta_1(t) + \theta_2(t))] \tag{6.8}$$

The sum-frequency term is eliminated by the VCO-filter combination and therefore may be discarded. Signal $x(t)$ is filtered by $F(\omega)$ to produce the error signal

$$\begin{aligned} e(t) &= e_o(t) + \int_0^t x(t-u)f(u)du \\ &= e_o(t) + \int_0^t x(u)f(t-u)du \quad t \geq 0 \end{aligned} \quad (6.9)$$

where it is assumed that the input is applied at time zero. The value $e_o(t)$ is the zero-input response which depends only on the initial conditions existing in the filter at time zero. For a stable filter, this value approaches zero, as time approaches infinity and therefore can be set to zero.

The impulse response, $f(t)$ of the filter is chosen such that the correct order of loop is obtained. For now no specific order for the filter will be defined. From figure 6.3, and assuming that all high-order terms are removed by the loop, an integro-differential equation for the phase-locked loop can be written. Using equations 6.7 through 6.9, the new VCO frequency, as given by equation 6.10 is obtained.

$$\frac{d\theta_2(t)}{dt} = \omega_{vco} + K_{vco} \int_0^t AK_v f(t-u) \sin(\phi(u)) du \quad (6.10)$$

where

$$\phi(t) = \theta_1(t) - \theta_2(t) \quad (6.11)$$

is the phase error. The loop gain is defined as

$$K = K_v K_{vco} \quad (6.12)$$

Therefore for a given input phase $\theta_1(t)$, the solution $\phi(t)$ to this integro-differential equation describes the operation of the phase-locked loop exactly. Using equation 6.10, the loop may be represented as shown in figure 6.4.

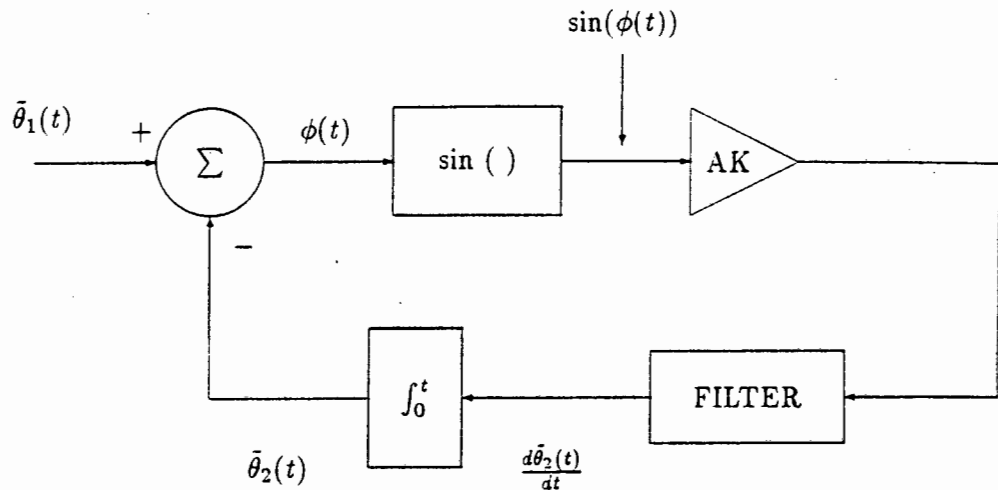


Figure 6.4: Phase-Locked Loop Equivalent Circuit

Since the VCO centre frequency is fixed, the VCO centre frequency may be included as part of the input phase. The phases $\bar{\theta}_1(t)$ and $\bar{\theta}_2(t)$ in figure 6.4 are

$$\bar{\theta}_1(t) \equiv \theta_1(t) - \omega_{vco}t - \theta_{vco} \quad (6.13)$$

$$\bar{\theta}_2(t) \equiv \theta_2(t) - \omega_{vco}t - \theta_{vco} \quad (6.14)$$

6.2.2 Phase-Plane Analysis of Second Order Phase-locked Loop

The second order phase-locked loop is by far the most widely used. The predominate reason for this is that as the number of poles (i.e. the order) of the loop is increased, the more unstable the loop dynamics become. This is clearly identified in the root loci of a PLL transfer function. A second order loop is stable. Therefore the choice of filter should have one pole. The ideal filter is a lead-lag filter. The lead-lag filter used in the analysis is of the form

$$F(s) = 1 + \frac{a}{s} \quad (6.15)$$

For such a filter, the steady state error, i.e. $\lim_{t \rightarrow \infty} \phi(t)$ is zero.

Using phase-plane techniques, both the transient and steady-state behaviour of a phase-locked loop can be obtained. Before determining the form of the phase-plane plot, the differential equation 6.10 needs to be rewritten. The lead-lag filter's input-output time-relationship may be written as

$$y(t) = z(t) + a \int_0^t z(\eta) d\eta \quad (6.16)$$

or

$$\frac{dy(t)}{dt} = \frac{dz(t)}{dt} + a z(t) \quad (6.17)$$

From figure 6.4 and equation 6.17, the following substitutions, $y(t) = \frac{d\bar{\theta}_2}{dt}$ and $z(t) = AK \sin(\phi(t))$, can be made. The resultant equation is

$$\frac{d}{dt} \left\{ \frac{d\bar{\theta}_2}{dt} \right\} = \frac{d}{dt} \{ AK \sin(\phi(t)) \} + a AK \sin(\phi(t)) \quad (6.18)$$

This can be rewritten as

$$\begin{aligned} \frac{d^2 \dot{\phi}(t)}{dt^2} + AK \cos(\phi(t)) \frac{d\dot{\phi}(t)}{dt} + a AK \sin(\phi(t)) \\ = \frac{d^2 \bar{\theta}_1(t)}{dt^2} \end{aligned} \quad (6.19)$$

Equation 6.19 cannot be used in its present form. Define $\dot{\phi} = \frac{d\phi}{dt}$, and rewrite equation 6.19 using this definition.

$$\ddot{\phi} + AK \dot{\phi} \cos(\phi) + a AK \sin(\phi) = \frac{d^2 \bar{\theta}_1(t)}{dt^2} \quad (6.20)$$

Dividing by $\dot{\phi}$ and recognizing that $\frac{\ddot{\phi}}{\dot{\phi}} = \frac{d\dot{\phi}}{d\phi}$, equation 6.20 becomes

$$\frac{d\dot{\phi}}{d\phi} = -AK \cos(\phi) - a AK \frac{\sin(\phi)}{\dot{\phi}} + \frac{d^2 \bar{\theta}_1(t)}{dt^2} \quad (6.21)$$

Equation 6.21 defines the dynamics of a second order phase-locked loop completely. From this equation it is possible to determine the phase-plane plot. For our analysis it is assumed

that $\bar{\theta}_1(t)$ is of the form $(\omega_{in} - \omega_{vco})t - \theta_{vco}$. Substituting $\bar{\theta}_1(t)$ in equation 6.21, it is clear that the term $\frac{d^2\bar{\theta}_1(t)}{dt^2}$ becomes zero.

6.2.2.1 Phase-Plane Plot

The phase-plane plot for phase-locked loops is a plot of frequency error versus phase error, i.e. $\frac{d\phi(t)}{dt}$ and $\phi(t)$ respectively. To develop the software to calculate the phase-plane plot for the second order phase-locked loop a synthesis diagram was determined. Using equation 6.16 and figure 6.4 the synthesis diagram, figure 6.5, was obtained.

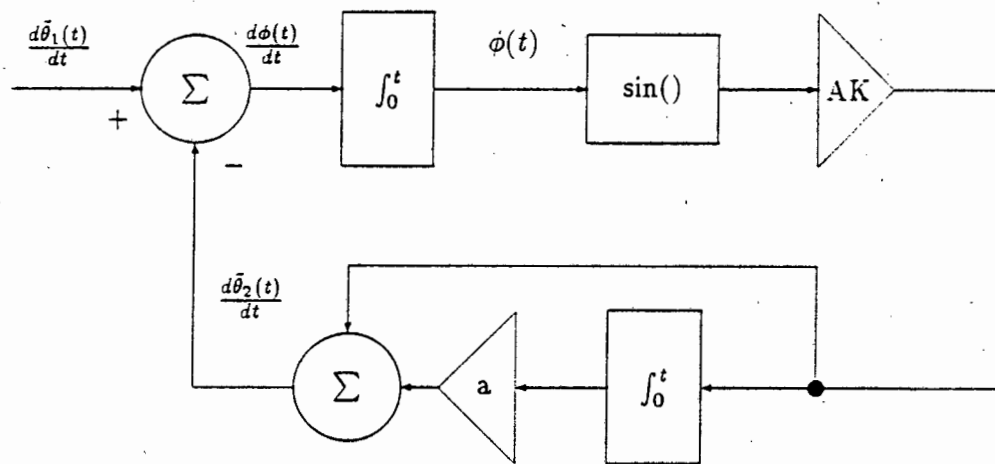


Figure 6.5: Synthesis Diagram

6.2.2.2 Examples of Phase-Plane Plots

For the case of the loop just described, a series of plots with different initial conditions for various filter constants and loop gains were plotted. Before comparing the various plots, some interesting characteristics of these plots need to be mentioned. Examining the sinusoidal trajectories of figures 6.7 and 6.9, it is clear that the sinusoid does not return to the same point as the initial condition, implying that the trajectories are decaying. As the initial value of ϕ is decreased, so the decay increases more rapidly. If one were to plot the curves over a large range of ϕ , all the trajectories will eventually be pulled in. This occurs when the system stops skipping cycles. At this point the system is said to be in frequency lock. At odd multiples of π a saddle point exists. This is clear from the graphs, since at

these odd multiples the curves move away from the abscissa.

Comparing figure 6.6 and 6.8, both under-damped systems, it is clear that the value of a determines that damping of the system, while AK determines the pull-in range. For both these cases, no cycle slipping occurred. In the case of figures 6.7 and 6.9, the pull in range of the system depicted in figure 6.9 is narrower for the same AK , but different a . The values for a and AK chosen for the loop depicted in figure 6.9 are optimal in the sense that the system is critically damped.

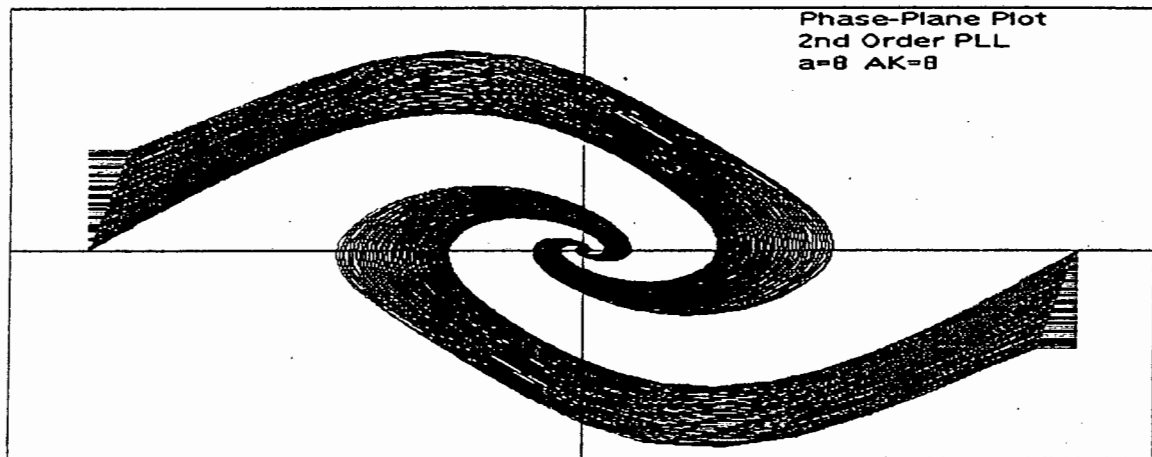


Figure 6.6: Phase-plane for $\frac{a}{AK} = 1$. The loop is under-damped, but all the trajectories are pulled-in without cycle slippages.

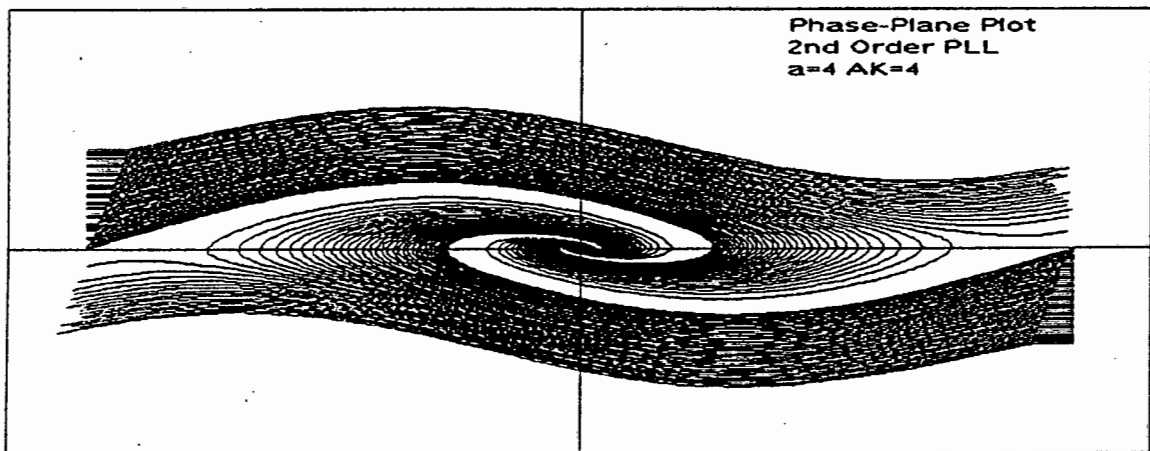


Figure 6.7: Phase-plane for $\frac{a}{AK} = 1$. The loop is under-damped, and the pull-in range narrow than in the previous figure.

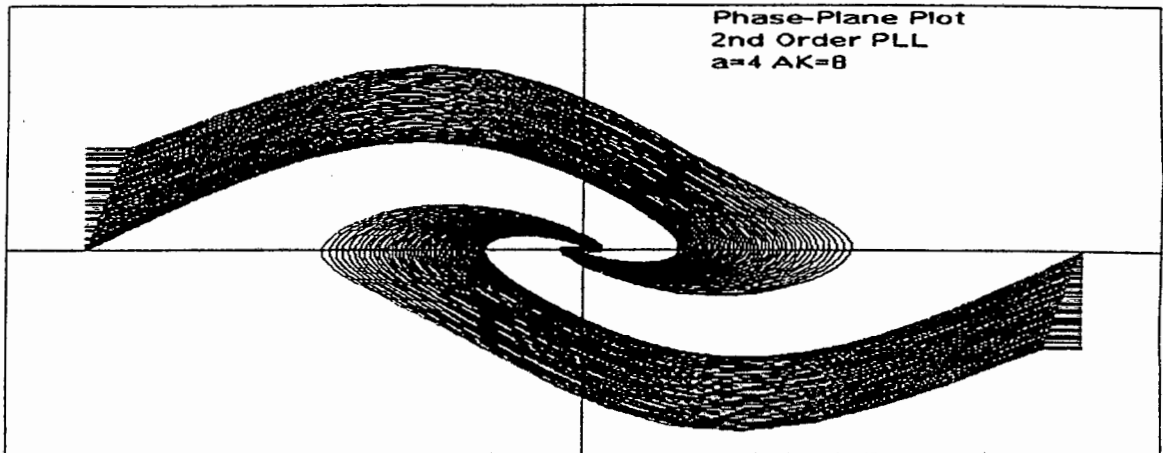


Figure 6.8: Phase-plane for $\frac{a}{AK} = 0.5$. The loop is under-damped. AK determines the pull-in range of the phase-locked loop.

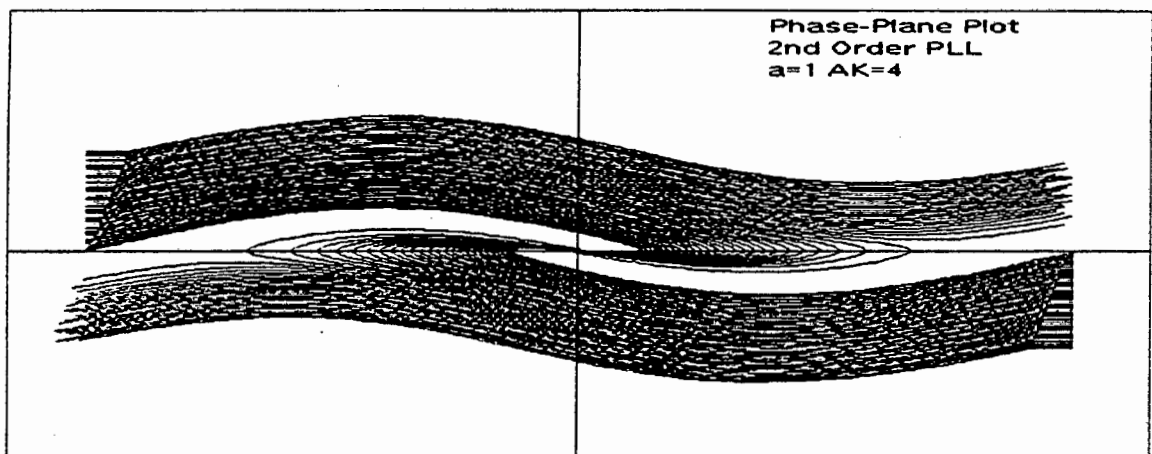


Figure 6.9: Phase-plane for $\frac{a}{AK} = 0.25$. The loop is critically damped, having a narrow pull-in range and longer locking time.

6.3 Stage 2: Clock Recovery PLL analysis

Phase-locked loops are used to recover the clock frequency of a transmitted data sequence to ensure the correct sampling of the incoming sequence [46]. A number of implementations exist. The one that concerns us directly is that of an analog implementation, as shown figure 6.10. The primary reason for this is that the results and methods have direct application to the analysis of the Code-phase Synchronisation network for direct sequence spread spectrum presented in this dissertation.

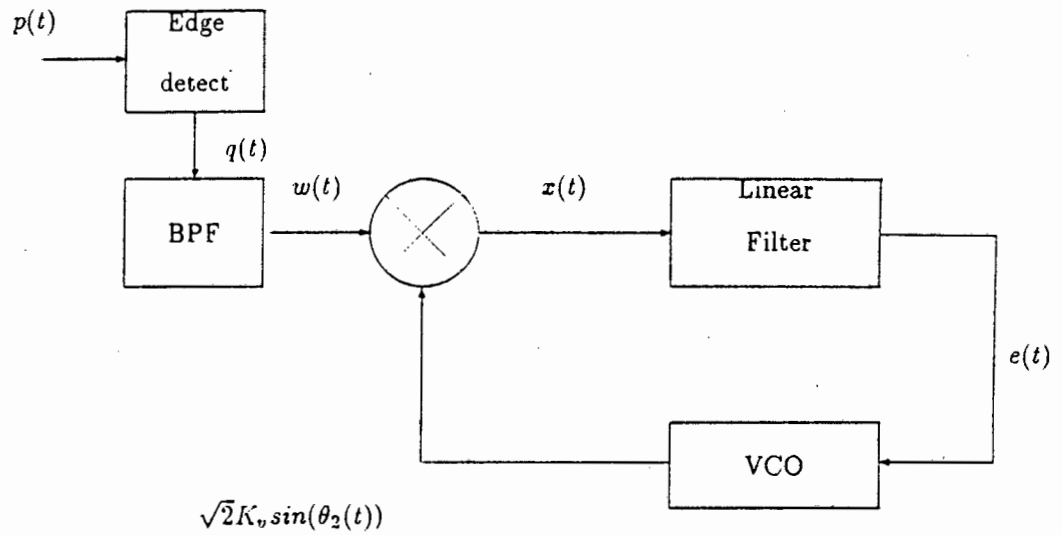


Figure 6.10: Clock Recovery Circuit

If one were to stimulate a bandpass filter, with a reasonable Q , periodically, and within the bandwidth of the filter, the filter would start resonating at the frequency of the input impulse train. This is clearly shown in 6.11, where the filter has a centre frequency of 50 rad/s and the frequency of the impulse train is at 40 rad/s.

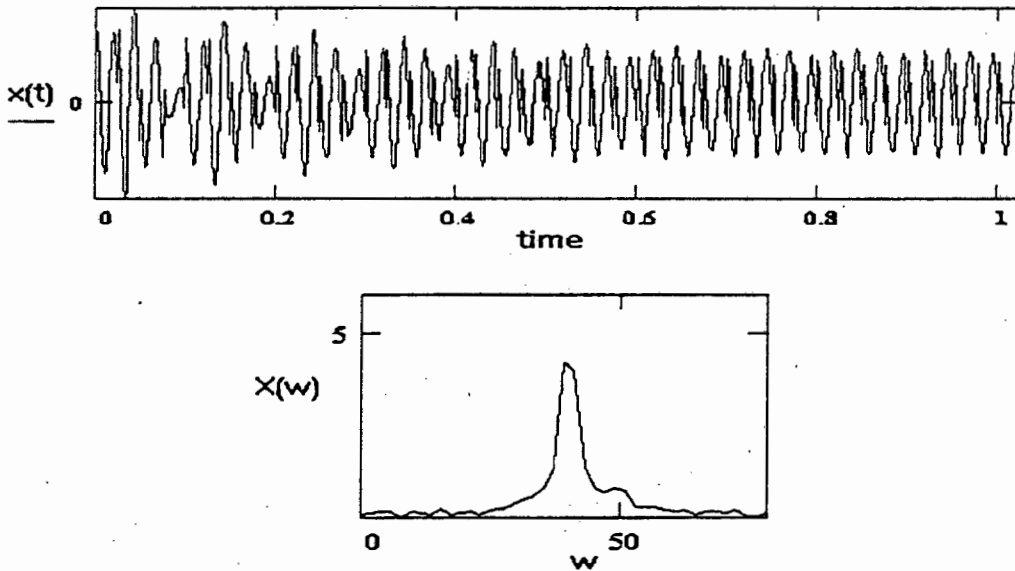


Figure 6.11: Time and Frequency response of output of BPF

Although from figure 6.11 it is possible to assume that the input becomes sinusoidal, the dynamics of how the loop reacts within the first number of cycles would be lost. To overcome this problem using figure 6.10 and with phase-plane techniques it becomes possible to examine the loop dynamics and transient behaviour for this implementation.

6.3.1 Mathematical Description of System

To determine the overall equation that defines the dynamics of the system, it is necessary to determine the impulse response of the bandpass filter. Although it is possible to choose one of the many types, for the purpose of this analysis a Butterworth filter was selected.

The transfer function of the bandpass filter is:

$$H(s) = \frac{s}{s^2 + s\frac{\omega_{bpf}}{Q} + \omega_{bpf}^2} \quad (6.22)$$

where ω_{bpf} is the centre frequency of the bandpass filter. The inverse Laplace transform of equation 6.22 is,

$$h(t) = e^{-\frac{\omega_{bpf}}{2Q}t} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} t) - \frac{e^{-\frac{\omega_{bpf}}{2Q}t}}{\sqrt{4Q^2 - 1}} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} t) \quad (6.23)$$

The input to the bandpass filter, is an edge detected sequence, given by equation 6.24 where $a_{mod(i,L)} = \pm 1$.

$$q(t) = \sqrt{2}A \sum_{n=-\infty}^{\infty} a_n \delta(t - nT_c) \quad (6.24)$$

The output of the bandpass filter is the convolution of the impulse train, equation 6.24, and equation 6.23. Therefore,

$$\begin{aligned} w(t) &= q(t) \otimes h(t) \\ &= \sqrt{2}A \sum_{n=-\infty}^{\infty} a_n e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \\ &\quad - \sqrt{2}A \sum_{n=-\infty}^{\infty} a_n \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)}}{\sqrt{4Q^2 - 1}} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \end{aligned} \quad (6.25)$$

The input to the phase-locked loop is therefore the sum of decaying sinusoids that are time shifted. This signal is fed into a multiplier and is multiplied by the output of the VCO (equation 6.26).

$$v(t) = \sqrt{2}K' \sin(\theta'(t)) \quad (6.26)$$

where

$$\frac{d\theta'(t)}{dt} = \omega_{vco} + K_{vco}e(t) \quad (6.27)$$

Multiplying equations 6.25 and 6.26, and using the following identities:

$$\begin{aligned} \sin(A)\cos(B) &= \frac{1}{2} [\sin(A+B) + \sin(A-B)] \text{ and} \\ \sin(A)\sin(B) &= \frac{1}{2} [\cos(A-B) - \cos(A+B)] \end{aligned}$$

the output of the multiplier is given in equation 6.28 with the assumption that the higher order frequency terms are filtered out by filtering action of the loop filter and VCO combination.

$$\begin{aligned} x(t) &= 2AK' \sum_{n=-\infty}^{\infty} a_n e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \sin(\theta'(t)) \\ &- 2AK' \sum_{n=-\infty}^{\infty} a_n \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)}}{\sqrt{4Q^2 - 1}} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \sin(\theta'(t)) \\ &= AK' \sum_{n=-\infty}^{\infty} a_n e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)} \sin(\theta'(t) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \\ &- AK' \sum_{n=-\infty}^{\infty} a_n \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-nT_c)}}{\sqrt{4Q^2 - 1}} \cos(\theta'(t) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - nT_c)) \end{aligned} \quad (6.28)$$

Equation 6.28 is fed into a filter $f(t)$ to generate the error signal that drives the VCO. The output of the filter is therefore the convolution between $x(t)$ and $f(t)$. It has been assumed that at time $t = 0^-$, $f(t) = 0$.

$$e(t) = f(t) \otimes x(t)$$

$$\begin{aligned}
&= \int_0^t x(u)f(t-u)du \\
&= \int_0^t AK' \sum_{n=0}^{\infty} a_n e^{-\frac{\omega_{bpf}}{2Q}(u-nT_c)} \sin(\theta'(u) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}}(u - nT_c)) f(t-u) du \\
&- \int_0^t AK' \sum_{n=-\infty}^{\infty} a_n \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-nT_c)}}{\sqrt{4Q^2 - 1}} \cos(\theta'(u) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}}(u - nT_c)) f(t-u) du
\end{aligned} \tag{6.29}$$

Combining equation 6.27 and 6.29, the new VCO frequency is obtained, namely:

$$\begin{aligned}
\frac{d\theta'(t)}{dt} &= \omega_{vco} + K_{vco} \int_0^t AK' \sum_{n=0}^{\infty} a_n e^{-\frac{\omega_{bpf}}{2Q}(u-nT_c)} \sin(\theta'(u) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}}(u - nT_c)) f(t-u) du \\
&- K_{vco} \int_0^t AK' \sum_{n=-\infty}^{\infty} a_n \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-nT_c)}}{\sqrt{4Q^2 - 1}} \cos(\theta'(u) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}}(u - nT_c)) f(t-u) du
\end{aligned} \tag{6.3}$$

Define:

$$K = K' K_{vco} \tag{6.31}$$

$$\phi(t) = \theta'(t) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} t \tag{6.32}$$

Differentiating equation 6.32, equation 6.33 is obtained.

$$\frac{d\phi(t)}{dt} = \frac{d\theta'(t)}{dt} - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} \tag{6.33}$$

Substituting equations 6.31 and 6.33 into equation 6.30, the integro-differential equation, equation 6.34 that defines the loop dynamics of the clock-recovery implementation is obtained.

$$\begin{aligned}
\frac{d\phi(t)}{dt} &= \omega_{vco} - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} \\
&+ AK \sum_{n=0}^{\infty} a_n \int_0^t \left[f(t-u) e^{-\frac{\omega_{bpf}}{2Q}(u-nT_c)} \cdot \sin(\theta_2(u) - \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}}(u - nT_c)) \right] du
\end{aligned}$$

must deal with an excessive number of decaying sinusoids, and maintain their phase relationship to time zero. The reason for this is that for every impulse, the filter's impulse response is obtained. The phase-relationship between the decaying sinusoids is determined by the frequency of the impulse train. As the sinusoids add and subtract, a new sinusoid will emerge which is at the frequency of the impulse train. The Q of the filter will determine the time that it takes to reach steady state. The simplest way to obtain the transient and steady state nature of the loop is through phase-plane analysis techniques.

6.3.2 Phase-Plane analysis of Clock Recovery Implementation

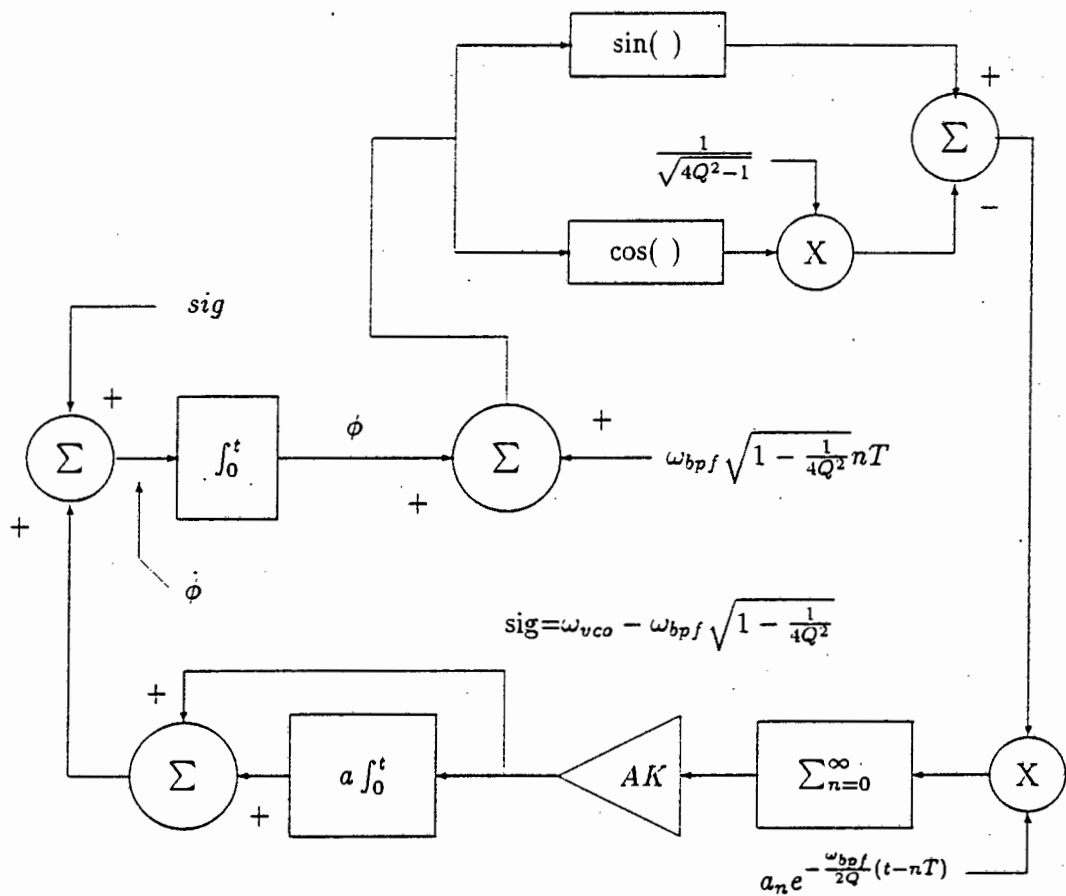


Figure 6.13: Simulation Diagram for Clock Recovery Circuit

The phase-plane analysis follows a similar approach to that of the second order phase-locked loop analysed in section 6.2.2.

The filter $f(t)$ was selected to be of the form as given by equation 6.15. Manipulating

equations 6.34 and 6.16, the simulation diagram 6.13 was obtained.

Examining figure 6.13 it is apparent that the loop will attempt to drive the phase error to zero. However, because of the implementation, it occurs that a signal may be generated offset from the bandpass centre frequency. This will cause a low frequency beat frequency to be generated equal to the frequency offset. In all the simulations this frequency has been compensated for by the inclusion of the offset frequency as part of the term *sig*.

The simulation diagram of figure 6.13 was programmed in 'C', and the simulation was normalised to 1 hertz for simplicity and speed. In the simulation a number of parameters may be varied to determine the effect of each parameter, as was done in the previous section on a standard phase-locked loop implementation. The ordinate of the figures that follow represents the frequency difference (ϕ), while the abscissa represents the phase difference (ϕ).

The first analysis using the numerical software packaged written is shown in figure 6.14. The parameters of the system were selected such that a frequency offset existed between the input impulse train and the centre frequency of the bandpass filter and the VCO. The impulse train was inverted to compensate for the 180 degree phase-shift introduced by the bandpass filter.

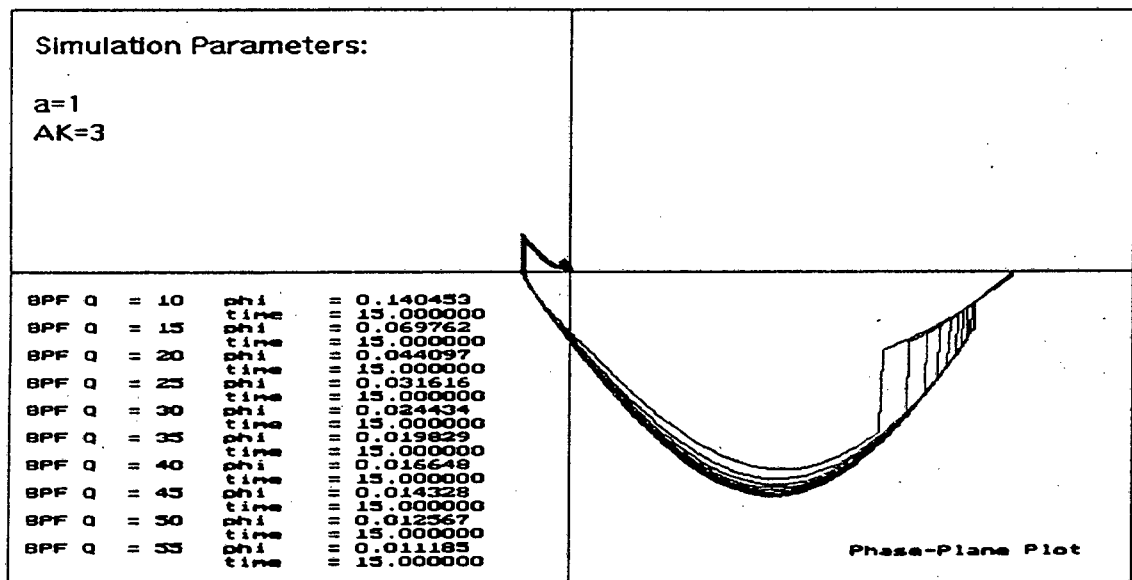


Figure 6.14: Phase-Plane Plot for Clock Recovery Implementation. The Q of the filter was varied to illustrate its effect on the phase error after a period of time. The phase-error is not linearly related to the change in Q.

The first parameter that was varied was the Q of the bandpass filter. The filter constant a and the gain AK of the phase-locked loop were selected as 1 and 3 respectively. Q was varied in steps of 5 from 10 through 55. The results are shown in figure 6.14. The steady-state phase error after 15 seconds is clearly not linear with a change in Q . Figure 6.15 shows the relationship is exponential.

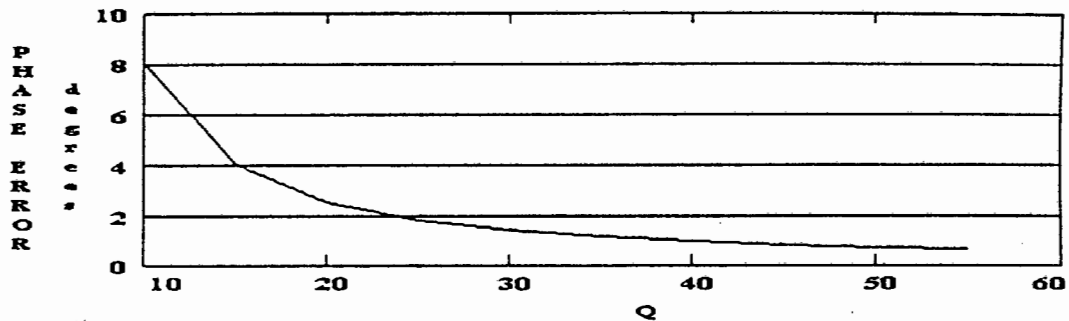


Figure 6.15: Phase Error versus Bandpass Filter Q . The phase error is not linearly related to the change in Q .

One characteristic of the figure should be highlighted. The graphs shows sudden jumps in frequency error. These are caused by the hitting of the bandpass filter by the impulse train. The other important aspect with regard to the plot is that as the Q increases, the sudden change in frequency is less.

Figure 6.16 shows the case when the impulse train frequency is outside the pull-in range. The trajectories are sinusoidal in nature except for the jumps which are caused by the impulses. The parameters of the phase-lock loop were selected as follows:

$$a = 0.5$$

$$AK = 2$$

$$Q = 20$$

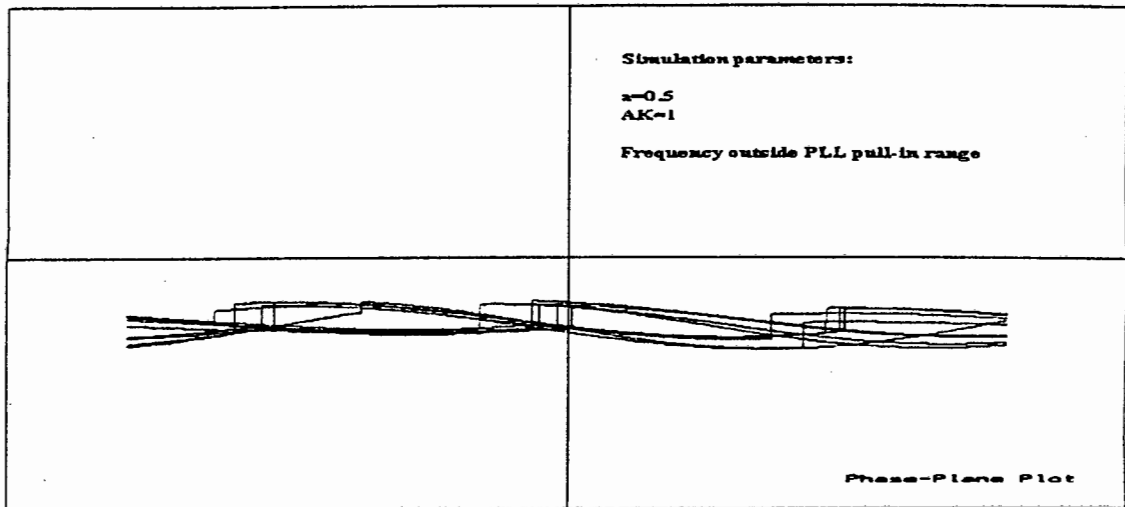


Figure 6.16: Phase-Plane of Clock Recovery Implementation. The frequency is outside the locking range of the system.

Figure 6.17 shows the case where the impulse train's frequency is within the pull-in range of the phase-locked loop. The impulse train frequency was set equal to that of the centre frequency of the bandpass filter and VCO. The loop parameters have been set to $a = 1$ and $AK = 2$. The Q was set to 20.

For Figure 6.18, the same configuration as figure 6.17 was used, except that after the loop had locked, the spikes were inverted. The loop was forced to re-acquire lock. From the figure, it can be seen that the loop went through a 180 degree phase-shift.

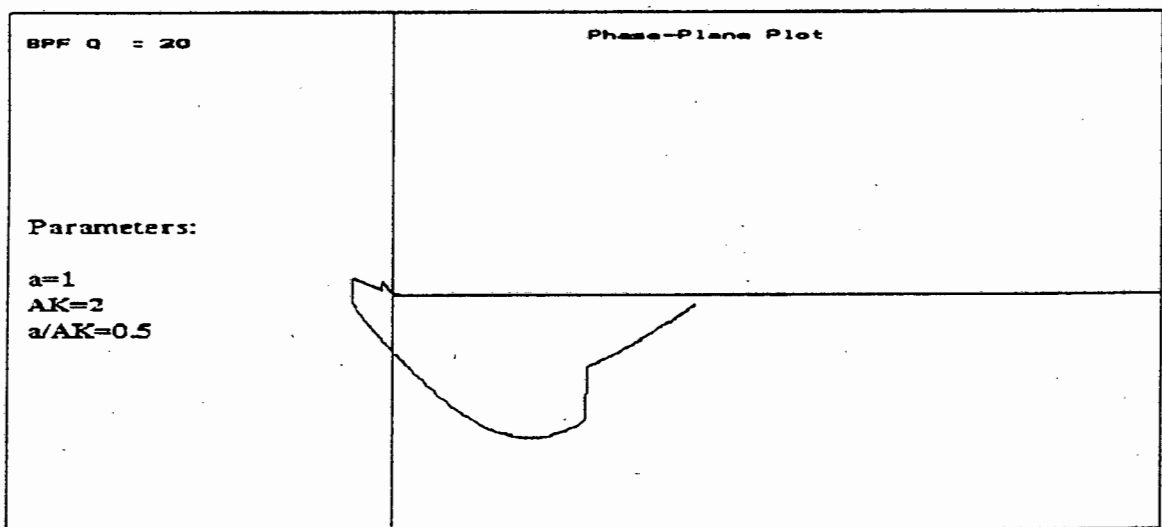


Figure 6.17: Phase-Plane of Clock Recovery Implementation. Shows the trajectory with negative impulses driving the clock-recovery loop.

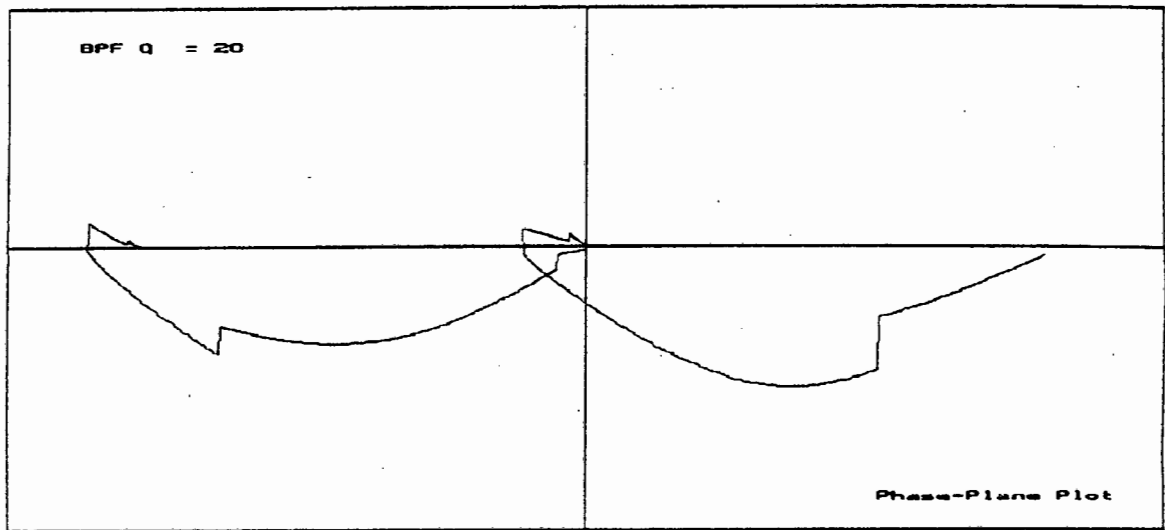


Figure 6.18: Phase-Plane of Clock Recovery Implementation. Once the system was in lock, the impulses were inverted. The phase goes through a 180 degree phase-shift.

6.3.3 Design Using Phase-Plane Plots

By using numerical techniques on a personal computer, it becomes possible to evaluate locking times, damping and investigate what instabilities exist in a system, without making unnecessary assumptions. The number of parameters that can vary within the design are many, but by generating a set of phase-plane curves, a good overview of the behaviour of the system can be obtained. To illustrate the concept consider figures 6.19 and 6.20.

Figure 6.19 shows the case for a fixed a and the varying of the loop gain AK . As the gain is increased, so the phase and frequency converge quicker, but the overall steady state phase error after 15 seconds is bigger. On the otherhand figure 6.20 shows the case for a fixed AK and varying a . Clearly as a is increased so the damping factor is decreased. Similarly, the locking time is improved with increasing a . This is consistent with what is observed in practice. As the loop gain is increased, or the filter bandwidth is widened, so the damping of the system decreases and the average locking time improves. This was clearly shown in the phase-plane analysis of phase-locked loops.

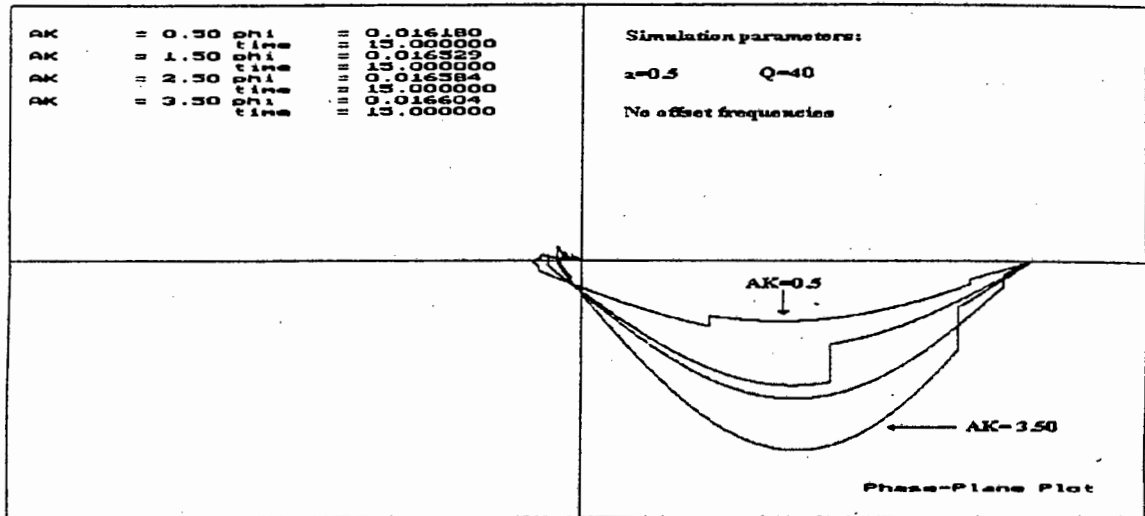


Figure 6.19: Phase-Plane of Clock Recovery Implementation. The phase-error gets smaller as AK is decreased.

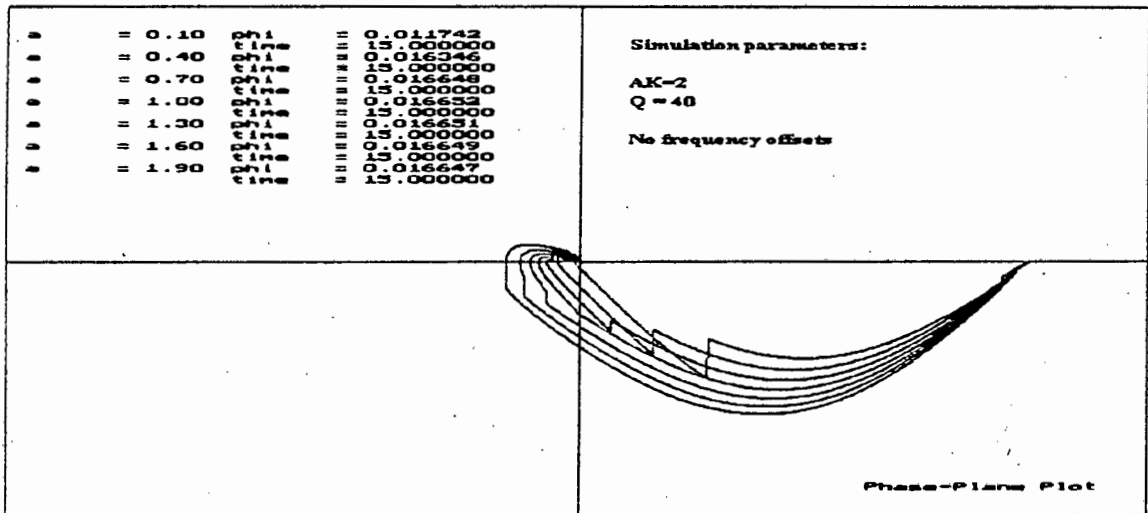


Figure 6.20: Phase-Plane of Clock Recovery Implementation. For small a, i.e. a narrow loop filter, the phase-error will be small. The trajectory path gets longer with an increase in a, but the acquisition time is shorter.

6.4 Summary

In this chapter it has been attempted to explain how phase-plane plots may be used to analyse second order phase-locked loops. It was also shown how phase-plane techniques can be used to analyse more complex structures, where the result is not as intuitive as the

simple case. Although phase-plane techniques are seldom used to design tracking loops, they have a definite place in theoretical engineering. The ability to show the transient and steady-state response of second-order phase-locked loops and to extract vital information like locking time makes it a very useful engineering tool. In the following chapter the techniques developed in this chapter will be used to analyse the novel synchronisation structure.

Chapter 7

Theoretical Analysis of Method Two (2)

In part one of Theoretical Analysis of Method Two the concept of phase-plane analysis was explained. Phase-plane techniques were applied to a second order phase-locked loop to illustrate the basic concepts of the technique. The analysis was extended to a clock recovery implementation and it was shown how phase-plane techniques can be used as a guide to determine the parameters of a practical implementation.

In this chapter, part two of Theoretical Analysis of Method Two, the concepts are extended to the analysis of the novel spread spectrum synchronisation network. An equation describing the dynamics of the novel synchronisation network is derived and analysed using phase-plane techniques.

7.1 Mathematical Model

To determine the overall equation, the synchronisation network has been broken down into two stages. Figure 7.1 shows stage one. The input to the bandpass filter is determined by the product of the differentiated input sequence and the locally generated pseudonoise sequence, whose chipping frequency is dependent on the output of the VCO of the phase-locked loop. This in turn implies that the input to the bandpass filter is dependent on the output of the VCO. As was described in preceding chapters, this multiplication results in a non-deterministic signal.

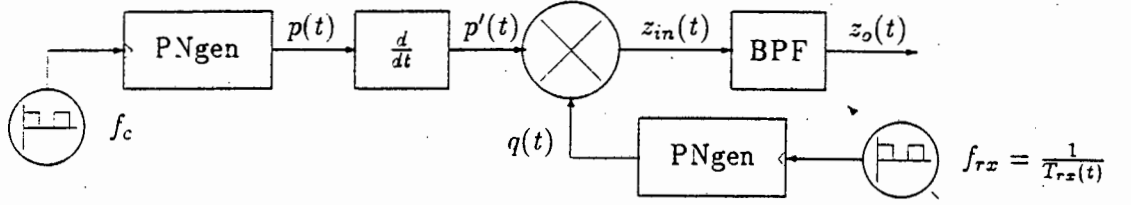


Figure 7.1: Novel Spread Spectrum Synchronisation Model (Stage 1)

To determine the output of the the bandpass filter, the following definitions are required. Equation 7.1 is the received pseudonoise sequence.

$$p(t) = \sqrt{2}A \sum_{i=-\infty}^{\infty} a_{mod(i,L)} [u(t - iT_c) - u(t - (i+1)T_c)] \quad (7.1)$$

where $a_{mod}(i, L)$ implies that $a_{i+L} \equiv a_i$ for all i .

$p(t)$ is fed into a signed edge detector (or differentiator) which results in equation 7.2.

$$p'(t) = \sqrt{2}A \sum_{i=-\infty}^{\infty} a_{mod(i,L)} [\delta(t - iT_c) - \delta(t - (i+1)T_c)] \quad (7.2)$$

The locally generated pseudo-noise sequence is defined in equation 7.3. The widths of the chips vary depending on the input frequency which is time-variant.

$$q(t) = \sum_{j=-\infty}^{\infty} b_{mod(j,L)} [u(t - jT_{rx}(t)) - u(t - (j+1)T_{rx}(t))] \quad (7.3)$$

The input to the bandpass filter is the product of equations 7.2 and 7.3, as given in equation 7.4.

$$\begin{aligned} z_{in}(t) &= p'(t)q(t) \\ &= A \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \cdot \left\{ \begin{array}{l} [u(t - jT_{rx}(t)) - u(t - (j+1)T_{rx}(t))] \delta(t - iT_c) \\ - [u(t - jT_{rx}(t)) - u(t - (j+1)T_{rx}(t))] \delta(t - (i+1)T_c) \end{array} \right\} \end{aligned} \quad (7.4)$$

Using the sampling property of the dirac delta the input to the bandpass filter becomes:

$$z_{in}(t) = A \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \left\{ \begin{array}{l} u(iT_c - jT_{rx}(iT_c))\delta(t - iT_c) \\ -u(iT_c - (j+1)T_{rx}(iT_c))\delta(t - iT_c) \\ -u((i+1)T_c - jT_{rx}((i+1)T_c))\delta(t - (i+1)T_c) \\ +u((i+1)T_c - (j+1)T_{rx}((i+1)T_c))\delta(t - (i+1)T_c) \end{array} \right\} \quad (7.5)$$

Define:

$$\begin{aligned} U_1(i, j, T_{rx}(iT_c), T_c) &= u(iT_c - jT_{rx}(iT_c)) - u(iT_c - (j+1)T_{rx}(iT_c)) \\ U_2(i, j, T_{rx}((i+1)T_c), T_c) &= u((i+1)T_c - jT_{rx}((i+1)T_c)) - u((i+1)T_c - (j+1)T_{rx}((i+1)T_c)) \end{aligned} \quad (7.6)$$

The signal defined by equation 7.5 is fed into a bandpass filter. As was the case for the clock recovery loop presented in chapter 6, the transfer function of the Butterworth bandpass filter is:

$$H(s) = \frac{s}{s^2 + s\frac{\omega_{bpf}}{Q} + \omega_{bpf}^2} \quad (7.7)$$

where ω_{bpf} is the centre frequency of the bandpass filter. The inverse Laplace transform of equation 7.7 is,

$$h(t) = e^{-\frac{\omega_{bpf}}{2Q}t} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} t) - \frac{e^{-\frac{\omega_{bpf}}{2Q}t}}{\sqrt{4Q^2 - 1}} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} t) \quad (7.8)$$

The output of the bandpass filter is the convolution of equation 7.8 and equation 7.5. Therefore, using the definitions defined in equation 7.6, the output of the filter is,

$$z_o(t) = \sqrt{2}A \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \left\{ \begin{array}{l} U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - iT_c)) \\ -U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)}}{\sqrt{4Q^2 - 1}} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - iT_c)) \end{array} \right\}$$

$$- \left\{ \begin{array}{l} U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)} \cos(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - (i+1)T_c)) \\ -U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)} \sin(\omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} (t - (i+1)T_c)) \end{array} \right\} \quad (7.9)$$

From equation 7.9. it is clear that the output of the bandpass is the sum of decaying sinusoids, randomly inverted according to the sign of the product of a_i and b_j . The output of the bandpass filter is fed into the phase-locked loop. This is illustrated in figure 7.2.

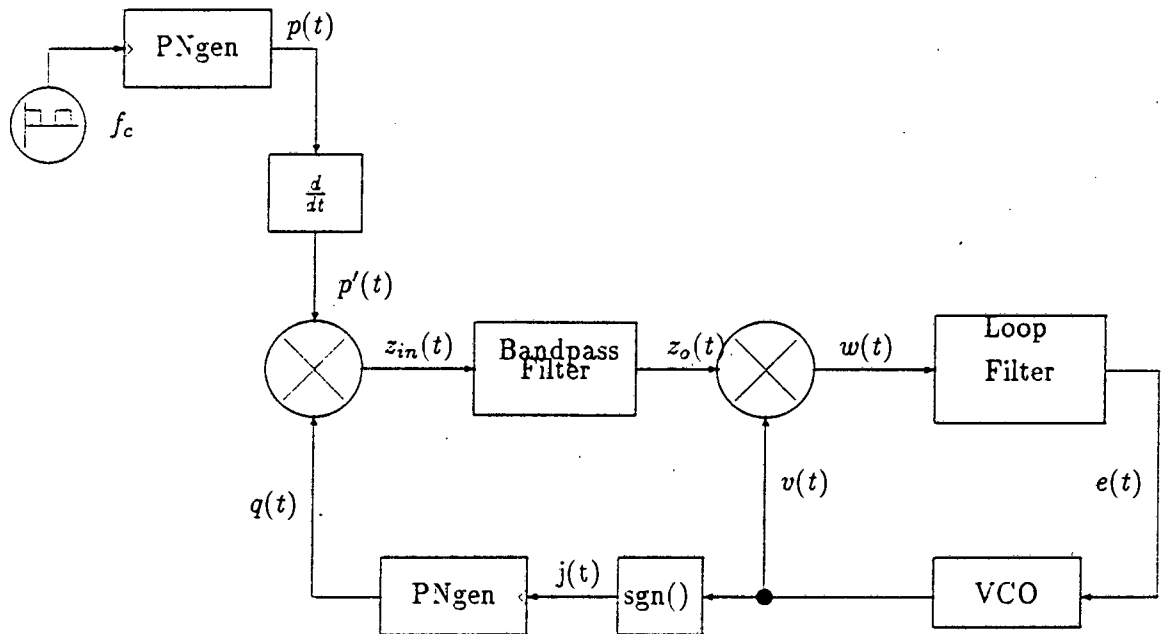


Figure 7.2: Novel Spread Spectrum Synchronisation Model (Complete)

The signal at the output of the VCO is given by

$$v(t) = \sqrt{2}K' \sin(\theta'(t)) \quad (7.10)$$

where

$$\frac{d\theta'(t)}{dt} = \omega_{vco} + K_{vco} e(t) \quad (7.11)$$

Multiplying equations 7.9 and 7.10, and using the following identities:

$$\begin{aligned}\sin(A)\cos(B) &= \frac{1}{2} [\sin(A+B) + \sin(A-B)] \text{ and} \\ \sin(A)\sin(B) &= \frac{1}{2} [\cos(A-B) - \cos(A+B)]\end{aligned}$$

the output of the multiplier is given in equation 7.13 with the assumption that the higher order frequency terms are filtered out by the VCO-filter action and therefore can be ignored.

Define:

$$\Omega_{bpf} = \omega_{bpf} \sqrt{1 - \frac{1}{4Q^2}} \quad (7.12)$$

$$\begin{aligned}w(t) &= \sqrt{2}A \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \\ &\quad \left\{ \left\{ \begin{aligned} &U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)} \cos(\Omega_{bpf}(t-iT_c)) \\ &-U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)}}{\sqrt{4Q^2-1}} \sin(\Omega_{bpf}(t-iT_c)) \end{aligned} \right\} \sqrt{2}K' \sin(\theta'(t)) \right. \\ &\quad \left. - \left\{ \begin{aligned} &U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)} \cos(\Omega_{bpf}(t-(i+1)T_c)) \\ &-U_2(i, j, T_{rx}((i+1)T_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)}}{\sqrt{4Q^2-1}} \sin(\Omega_{bpf}(t-(i+1)T_c)) \end{aligned} \right\} \sqrt{2}K' \sin(\theta'(t)) \right\} \\ &= AK' \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)} \\ &\quad \left\{ \left\{ \begin{aligned} &U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)} \sin(\theta'(t) - \Omega_{bpf}(t-iT_c)) \\ &-U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-iT_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(t) - \Omega_{bpf}(t-iT_c)) \end{aligned} \right\} \right. \\ &\quad \left. - \left\{ \begin{aligned} &U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)} \sin(\theta'(t) - \Omega_{bpf}(t-(i+1)T_c)) \\ &-U_2(i, j, T_{rx}((i+1)T_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(t-(i+1)T_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(t) - \Omega_{bpf}(t-(i+1)T_c)) \end{aligned} \right\} \right\} \end{aligned} \quad (7.13)$$

Equation 6.28 is fed into a loop filter $f(t)$ to generate the error signal that drives the VCO. The output of the filter is therefore the convolution of $x(t)$ with $f(t)$. It has been assumed that at time $t = 0^-$, $f(t) = 0$.

$$\begin{aligned}e(t) &= f(t) \otimes x(t) \\ &= \int_0^t x(u) f(t-u) du \\ &= \int_0^t AK' \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(j,L)}\end{aligned}$$

$$\begin{aligned}
& \left\{ \left\{ \begin{aligned} & U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)} \sin(\theta'(u) - \Omega_{bpf}(u - iT_c)) \\ & -U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(u) - \Omega_{bpf}(u - iT_c)) \end{aligned} \right\} \right. \\
& \left. - \left\{ \left\{ \begin{aligned} & U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)} \sin(\theta'(u) - \Omega_{bpf}(u - (i+1)T_c)) \\ & -U_2(i, j, T_{rx}((i+1)T_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(u) - \Omega_{bpf}(u - (i+1)T_c)) \end{aligned} \right\} \right\} f(t-u) du \right. \\
& \left. \right\} \quad (7.14)
\end{aligned}$$

Combining equation 7.11 and 7.14, the new VCO frequency is obtained, namely:

$$\begin{aligned}
\frac{d\theta'(t)}{dt} &= \omega_{vco} + K_{vco} \int_0^t \int_0^t AK' \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(i,L)} f(t-u) \\
& \left\{ \left\{ \begin{aligned} & U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)} \sin(\theta'(u) - \Omega_{bpf}(u - iT_c)) \\ & -U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(u) - \Omega_{bpf}(u - iT_c)) \end{aligned} \right\} \right. \\
& \left. - \left\{ \left\{ \begin{aligned} & U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)} \sin(\theta'(u) - \Omega_{bpf}(u - (i+1)T_c)) \\ & -U_2(i, j, T_{rx}((i+1)T_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(u) - \Omega_{bpf}(u - (i+1)T_c)) \end{aligned} \right\} \right\} dudt \right. \\
& \left. \right\} \quad (7.15)
\end{aligned}$$

Define:

$$K = K' K_{vco} \quad (7.16)$$

$$\phi(t) = \theta'(t) - \Omega_{bpf} t \quad (7.17)$$

Differentiating equation 7.17, equation 7.18 becomes

$$\frac{d\phi(t)}{dt} = \frac{d\theta'(t)}{dt} - \Omega_{bpf} \quad (7.18)$$

Substituting equation 7.16 and 7.18 into equation 7.15, equation 7.19 is obtained. It defines the loop dynamics of the spread spectrum synchronisation network.

$$\begin{aligned}
 \frac{d\theta'(t)}{dt} = & \omega_{vco} - \Omega_{bpf} + AK \int_0^t \int_0^t \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a_{mod(i,L)} b_{mod(i,L)} f(t-u) \\
 & \left\{ \left\{ \begin{aligned} & U_1(i, j, T_{rx}(iT_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)} \sin(\theta'(u) - \Omega_{bpf}(u-iT_c)) \\ & -U_1(i, j, T_{rx}(iT_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-iT_c)}}{\sqrt{4Q^2-1}} \cos(\theta'(u) - \Omega_{bpf}(u-iT_c)) \end{aligned} \right\} \right. \\
 & \left. - \left\{ \begin{aligned} & [U_2(i, j, T_{rx}((i+1)T_c), T_c) e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)} \\ & \sin(\theta'(u) - \Omega_{bpf}(u-(i+1)T_c)) \\ & -[U_2(i, j, T_{rx}((i+1)T_c), T_c) \frac{e^{-\frac{\omega_{bpf}}{2Q}(u-(i+1)T_c)}}{\sqrt{4Q^2-1}} \\ & \cos(\theta'(u) - \Omega_{bpf}(u-(i+1)T_c))] \end{aligned} \right\} \right\} dudt
 \end{aligned} \tag{7.19}$$

To determine the equivalent diagram of equation 7.19, the terms $T_{rx}(iT_c)$ and $T_{rx}((i+1)T_c)$ of equation 7.5 are required. These terms are a function of the output of the VCO, which is defined as,

$$v(t) = \sqrt{2}K' \sin(\omega_{vco}t + K_{vco} \int_0^t e(\tau) d\tau) \tag{7.20}$$

The VCO signal, $v(t)$, is fed into a limiter of the form,

$$sgn(x) = \begin{cases} 5 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{7.21}$$

which generates a hard-limited digital clock signal. The clock signal can be written as

$$\begin{aligned}
 j(t) &= sgn \left(\sqrt{2}K' \sin(\omega_{vco}t + K_{vco} \int_0^t e(\tau) d\tau) \right) \\
 &= 5 \sum_{l=0}^{\infty} \left\{ \begin{aligned} & u(t - \frac{2\pi l}{\omega_{vco}} + \frac{\text{mod}(K_{vco} \int_0^t e(\tau) d\tau, 2\pi)}{\omega_{vco}}) \\ & u(t - \frac{2\pi(2l+1)}{2\omega_{vco}} + \frac{\text{mod}(K_{vco} \int_0^t e(\tau) d\tau, 2\pi)}{\omega_{vco}}) \end{aligned} \right\}
 \end{aligned} \tag{7.22}$$

To show that equation 7.22 is a hardlimited version of equation 7.20, consider the following example:

Let $e(t) = 0.5t$ and $\omega_{vco} = 2\pi$. Using Mathcad to calculate the waveforms, figure 7.3

is obtained. From the graph it is clear that equation 7.22 is indeed a limited version of equation 7.20.

Having determined the hard-limited digital waveform to drive the local pseudonoise sequence generator, it is possible to define $T_{rx}(t)$. The sequence output only changes on the positive edge of the TTL waveform. From equation 7.22 and figure 7.3, the positive edges are determined by the first term of equation 7.22. Therefore,

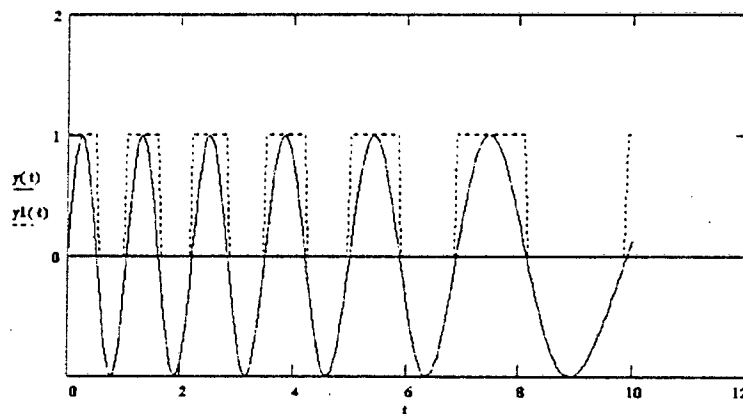


Figure 7.3: Limiting of a sinusoidal waveform.

$$T_{rx}(t, l) = \frac{2\pi l}{\omega_{vco}} - \frac{\text{mod}(K_{vco} \int_0^t e(\tau) d\tau, 2\pi)}{\omega_{vco}} \quad (7.23)$$

where l is the l th positive going edge since time 0.

Having determined all the equations that define the loop dynamics of the spread spectrum synchronisation network, it is possible to determine the equivalent diagram. From equation 7.19 and equation 7.23 the equivalent diagram, shown in figure 7.4, can be drawn.

Examining figure 7.4 and equation 7.19 it is clear that no simple analytical solution exists. The dynamic loop equation is essentially the summation of non-linear, time variant decaying sinusoidal terms that are randomly inverted depending on the output of the VCO. It is therefore apparent that the simplest method of analysing the overall system is using non-linear numerical techniques, like phase-plane analysis.

7.2 Phase-Plane Analysis

Although figure 7.4 is a fair representation of the overall network it is not in a form that can be implemented. The primary reason for this is the dependency on future values. This is due to the definition used for the pseudonoise sequences. Although it is possible to redefine the system such that the system is causal, it is also possible to modify the diagram such that the system is in a simpler form and therefore more efficient to implement. The complexity in figure 7.4 came from the need to determine the sign of the impulse entering the bandpass filter. It therefore required a mathematical definition of the pseudonoise sequences. The simulation diagram, shown in figure 7.5 is therefore a hybrid between the mathematical model (figure 7.4) and a practical implementation.

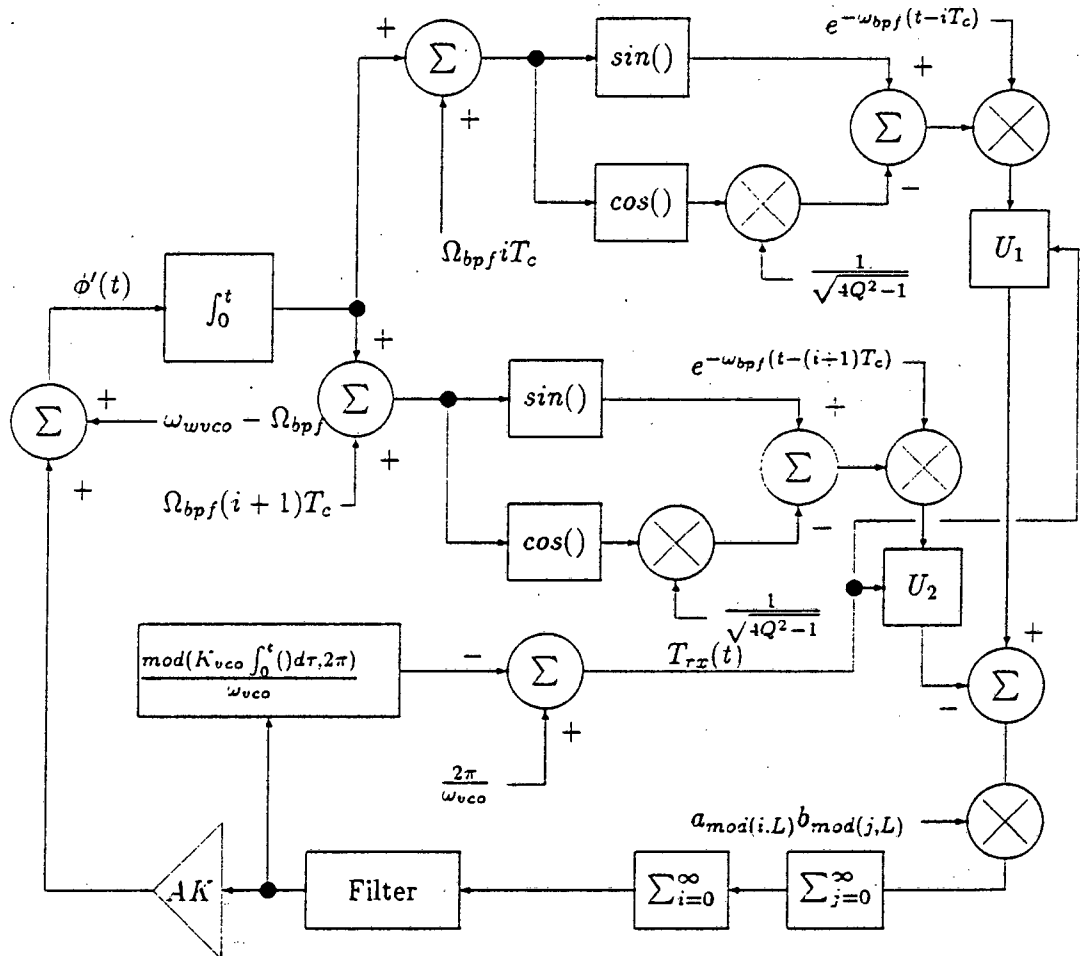


Figure 7.4: Equivalent Circuit of Synchronisation Network

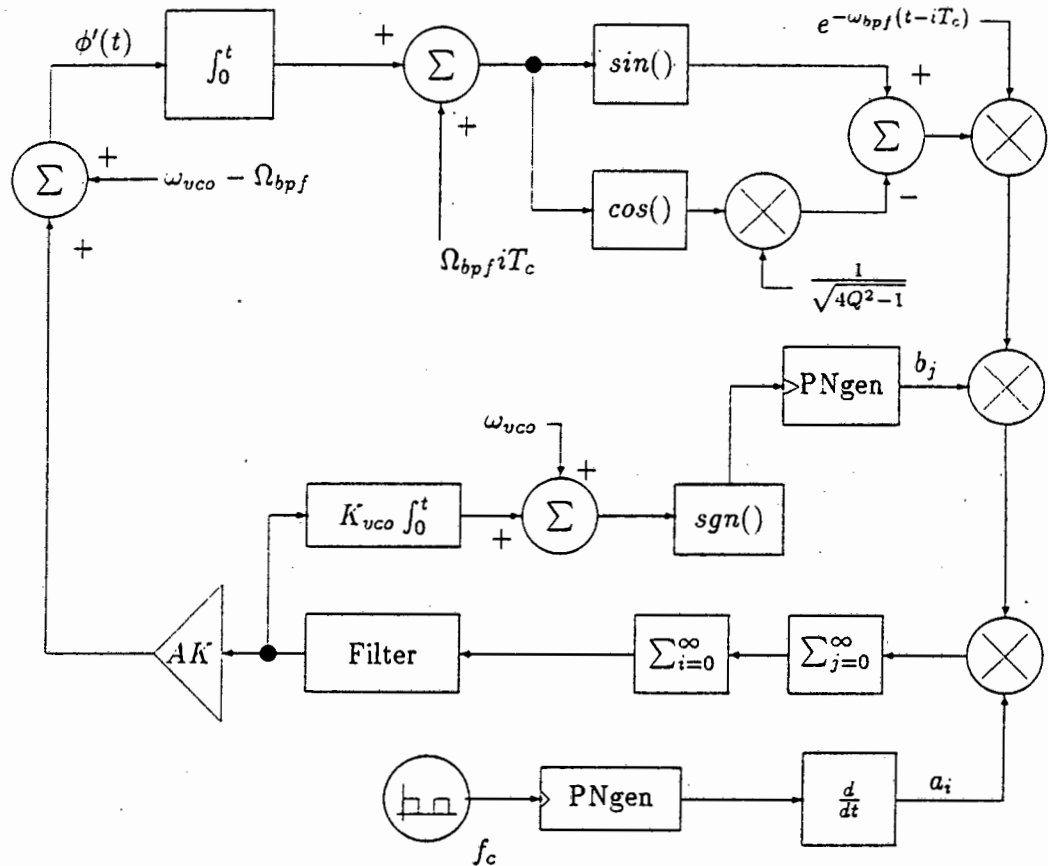


Figure 7.5: Simulation Diagram of Synchronisation Network

The phase-plane analysis follows a similar approach to that of the second order phase-locked loop analysed in section 6.2.2 and the phase-plane analysis of the clock recovery loop analysed in section 6.3.2. The filter $f(t)$ was selected to be of the form as given by equation 6.15.

The simulation diagram of figure 7.5 was programmed in 'C', and the simulation was normalised to 1 hertz for simplicity and scalability. In the simulation a number of parameters may be varied to determine the effect of each parameter, as was done in the previous section on a standard phase-locked loop implementation. The ordinate of the figures that follow represents the frequency difference (ϕ), while the abscissa represents the phase difference (ϕ). Lock occurs when the trajectory settles at either 0 or π radians. This corresponds to whether positive impulse or negative impulse lock is achieved.

Throughout the dissertation it has been highlighted that the frequency-phase grid of figure

2.5 in chapter 2 is analysed according to the information present in the product of the signed edge detected received PN sequence and the locally generated PN sequence. It was stated that this driving signal is non-deterministic and random in nature. In the next few sections, various conditions are simulated for a 15 bit maximal length sequence and various design issues are discussed. (Note that if: (x milliseconds at 100 kHz) is seen in the text, it is the locking time at a chipping frequency of 100 kHz and therefore can be compared to the locking times of the Tesla simulations.)

For the simulation results to be presented the following parameters were kept constant (unless otherwise indicated.):

- The bandpass filter centre frequency is set to that of the received PN sequence.
- The received PN sequence chipping frequency is $1.001 Hz$.
- The bandpass filter Q is 30.
- The amplitude (A) was assumed to be 1.
- A starting phase offset of 90 degrees.

7.2.1 The effect of Various Code-Phase Offsets

The code-phase offset and frequency offset between the received PN sequence and the locally generated PN sequence have a definite effect on the locking performance of the loop. This characteristic was illustrated in the Tesla Simulations in chapter 5. In this section the effect of code-phase offsets are investigated.

Simulation parameters:

$$a = 1.3$$

$$AK = 3.6$$

Figure 7.6 shows the case for a 5 chip offset between the received and locally generated PN sequences. The graph consists of four plots, namely the locally generated clock, the locally generated PN sequence, the received PN sequence and the phase-plane plot. The time reference is taken from the transmitted PN sequence.

From the graph the chaotic or random nature of the synchronisation technique is clearly illustrated. The trajectory path is determined by the sign of the impulses as they stimulate the bandpass filter. Lock is determined by the offset between the sequences and the state of

the phase-plane plot. If the trajectory has converged to either 0 or $\pm\pi$ and the sequences are either fully aligned or with a one chip offset (i.e. positive or negative lock) then the system is said to be in lock.

For the five chip offset lock was achieved after 67 seconds. Examining the locally generated clock (or the locally generated PN sequence), the slowing down of the locally generated PN sequence is clearly seen. If the simulation was shifted in frequency to 100 kHz, the locking time would be 0.67 milliseconds. This result is within the locking order obtained in the Tesla simulations.

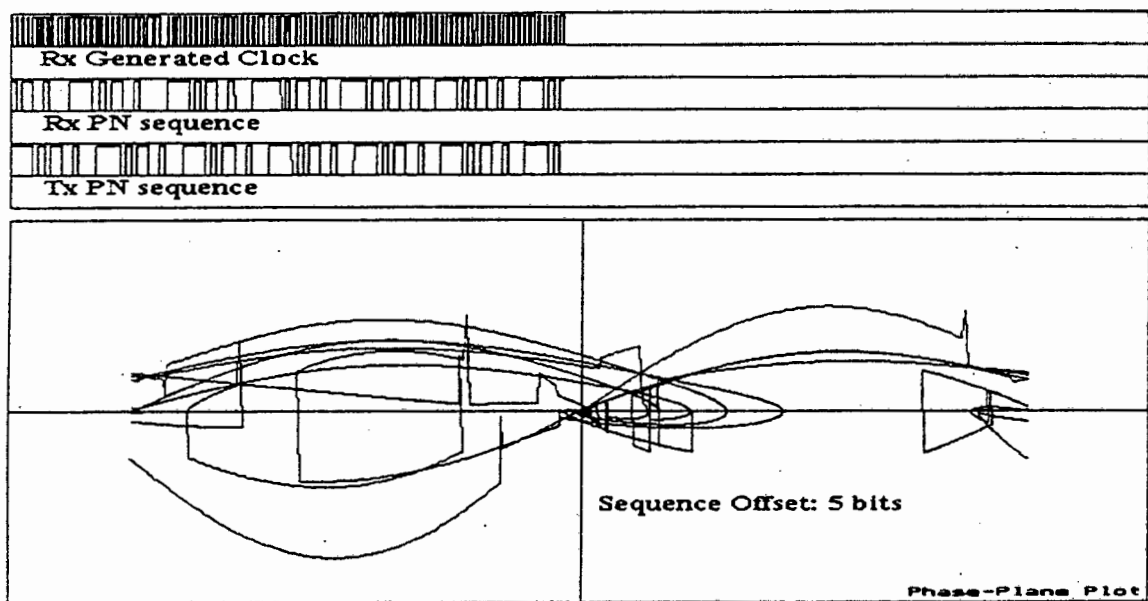


Figure 7.6: Phase-Plane Plot of Novel Loop (Offset 5 chips). Lock was achieved after 67 seconds. The chaotic nature of the system is clearly illustrated.

Using the same parameters as used for figure 7.6, except for a code phase offset of 7 chips, figure 7.7 was obtained. The locking time of the system was 124 seconds (1.24 milliseconds at 100 kHz), nearly double that of the previous case. Again the slowing down and speeding up of the locally generated sequence is prevalent.

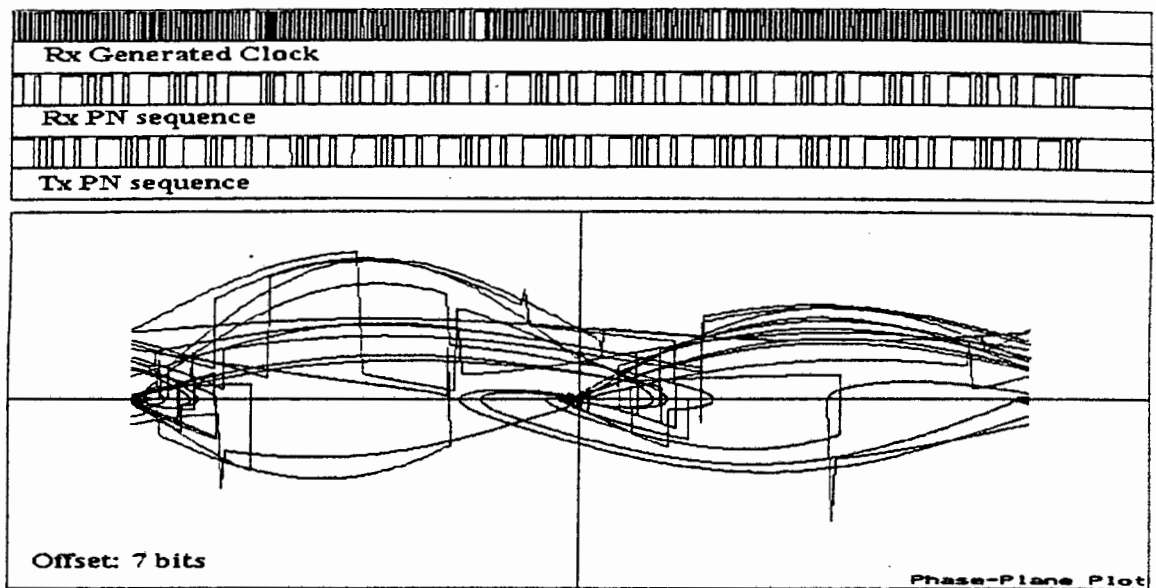


Figure 7.7: Phase-Plane Plot of Novel Loop (Offset 7 chips). Lock was achieved within 124 seconds. Comparing to the previous figure the unique nature of each plot is clearly identified.

Figure 7.8 shows the case for an 8 chip offset. The locking time is 220 seconds (2.2 milliseconds at 100 kHz). The convergence towards the locking point was slow, with the locally generated sequence clock frequency initially increasing rapidly in frequency, before slowing the locally generated PN sequence down sufficiently to achieve lock.

From the results presented thus far, the locking times are of the same order as those determined in the Tesla simulations. The convergence time is reasonable, but by tuning the various parameters that determine the characteristics of the system the locking time of the system can be improved (or made worse.). The question arises as to what values should be selected to improve the overall locking time of the system.

The model as such consists of a number of fundamental parameters that affect the operation of the loop. These parameters include the Q of the bandpass filter, the loop filter constant (a), the loop gain (AK), and the frequency offset between the transmitter chipping frequency and the phase-locked loop centre frequency.

In chapter 6 the effect of varying these parameters were shown under a number of conditions. In the next few sections, the effect of varying these parameters will be shown.

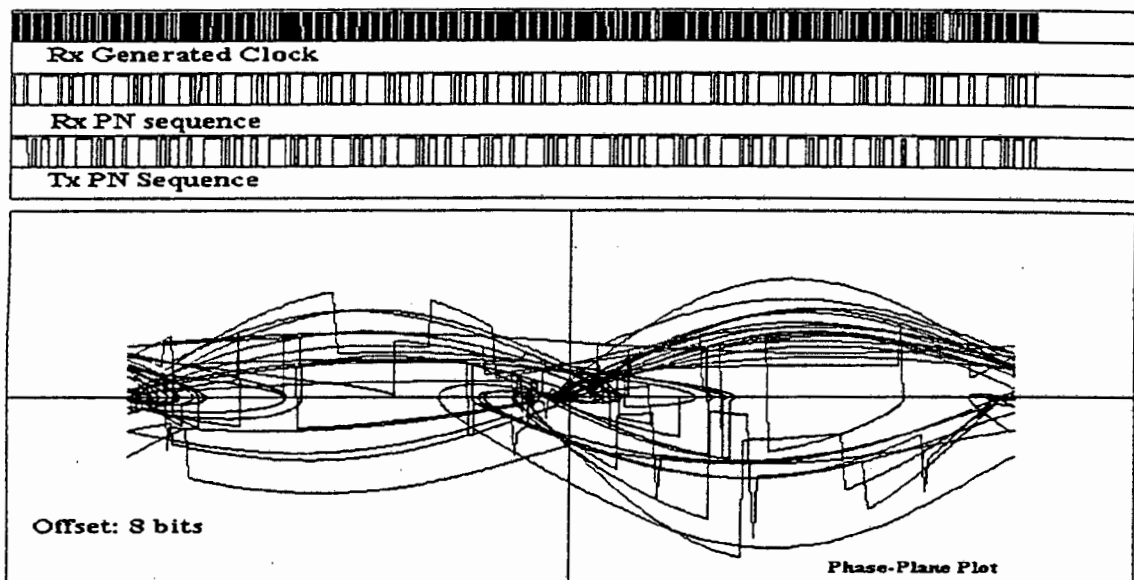


Figure 7.8: Phase-Plane Plot of Novel Loop (Offset 8 chips). Lock was achieved in 120 seconds.

7.2.2 Varying the Loop Filter Constant (a)

The loop filter constant (a) affects both the stability of the tracking loop and the locking time. As a is increased, so the bandwidth of the loop filter is increased. This implies that the overall noise bandwidth of the system is increased and so too is the locking range, while the locking time should be reduced.

Simulation values:

$$AK = 3.7$$

Offset of 2 chips.

Figure 7.9 shows the case for an a of 0.3. Clearly from figure 7.9 locking time is fairly short and little shifting of the VCO took place. Lock was achieved within 27 seconds (0.27 milliseconds at 100 kHz) and convergence after the first PN period was relatively quick. A dramatic slowing down took place after 21 seconds.

The a value was then increased to 0.6. The results are shown in figure 7.10. The system locked in approximately 25 seconds (0.25 milliseconds at 100 kHz). Similar dynamics as the previous case were obtain. One attribute of the phase-plane plots that should be pointed out, it that each phase-plane plot is unique depending on the parameter values selected. No

one plot is the same as another.

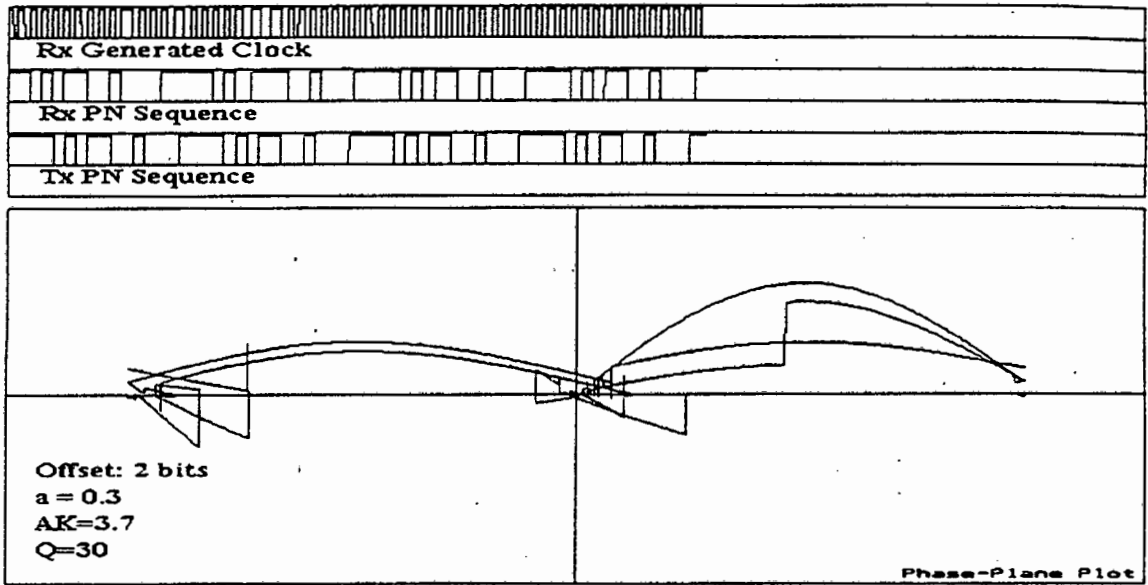


Figure 7.9: Phase-Plane Plot of Novel Loop ($a = 0.3$). Lock was achieved within 27 seconds. Convergence of the sequences was quick.

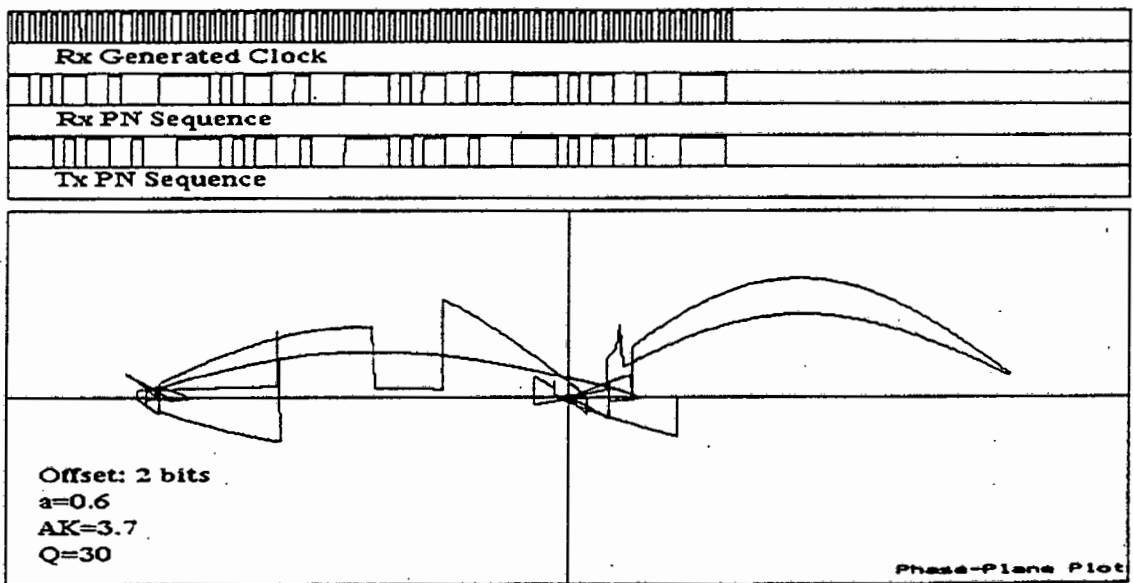


Figure 7.10: Phase-Plane Plot of Novel Loop ($a = 0.6$). The system locked within 25 seconds. A simple change in one parameter of the system, changes the characteristics of the phase-plane plot.

It is well known that convergence can be speeded up by selecting a and AK such that

the loop is not overly-damped. For figure 7.11, a was selected as 1.3 and AK was kept at 3.7. Examining the phase-plane plot it is clearly seen that the loop is critically damped. Although it was expected that the system would achieve lock quicker, lock was only achieved after 56 seconds (0.56 milliseconds at 100 kHz).

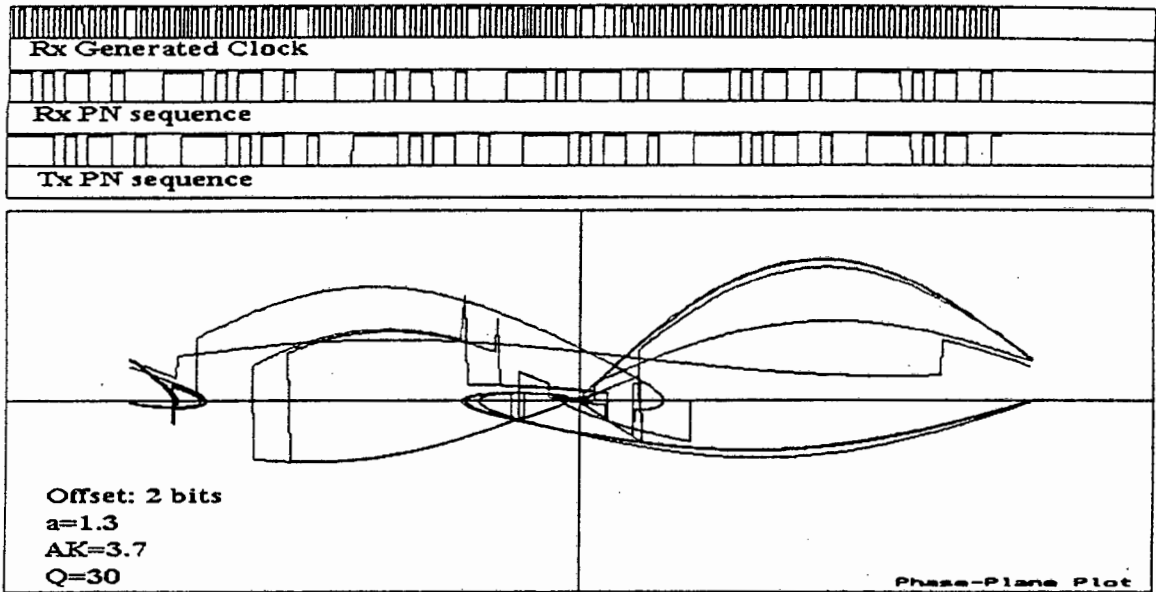


Figure 7.11: Phase-Plane Plot of Novel Loop ($a = 1.3$). Lock was achieved in 56 seconds.

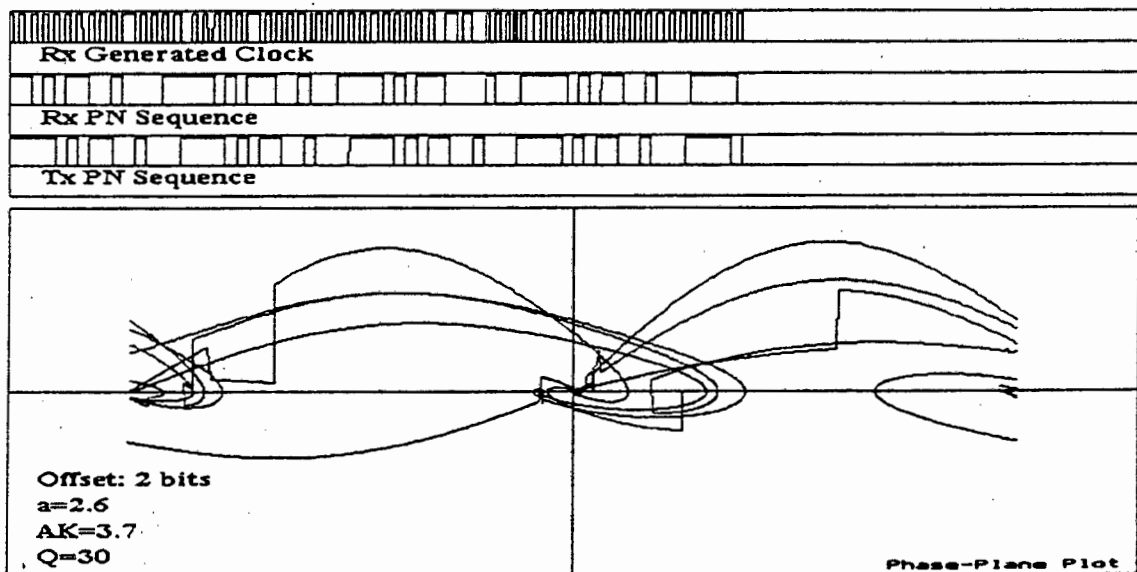


Figure 7.12: Phase-Plane Plot of Novel Loop ($a = 2.6$). Initial lock was achieved in 15 seconds, but lock was not maintained. This implies that the loop filter was too wide to maintain lock.

Figure 7.12 shows the case of an under-damped system. The value of a was selected as 2.6. Lock was achieved relatively quickly (around 15 seconds). Although it appears that the system remained in lock, the jitter in the system was high. It kept reacquiring lock every few seconds.

From the results it can be concluded that the value of a should not be selected too small or too big as both imply that locking time will not be as quick as one would like.

AK also plays an important role in the locking time. The VCO gain constant is proportional to the loop gain AK . In practical systems the allowable VCO gain is determined by physical constraints, a constraint which is not assumed in the mathematical model. In the next section the effect of the gain is considered.

7.2.3 Varying the Loop Gain (AK)

The gain constant (AK) also affects the stability of the tracking loop and the locking time. The loop gain is determined from the amplitudes of the incoming signal and the VCO output signal amplitude, the gain of the multipliers, and the VCO gain. In many designs, it is the VCO gain constant that determines the overall gain. The units of the VCO gain (K_{vco}) are radians per second per volt ($\frac{rad}{volt}$). Therefore, if the error signal is one volt, the instantaneous frequency will be shifted by $K_{vco} \frac{rad}{s}$. Therefore one would assume that by increasing the gain the system would achieve lock more quickly.

Simulation parameters:

$$a = 1.3$$

Offset of 2 chips.

Figure 7.13 shows the case for an AK of 2.5 and an a of 1.3. Examining the figure it would appear that lock should have been achieved within 15 seconds, but the loop characteristics are such that lock was only achieved in around 50 seconds (0.5 milliseconds at 100 kHz).

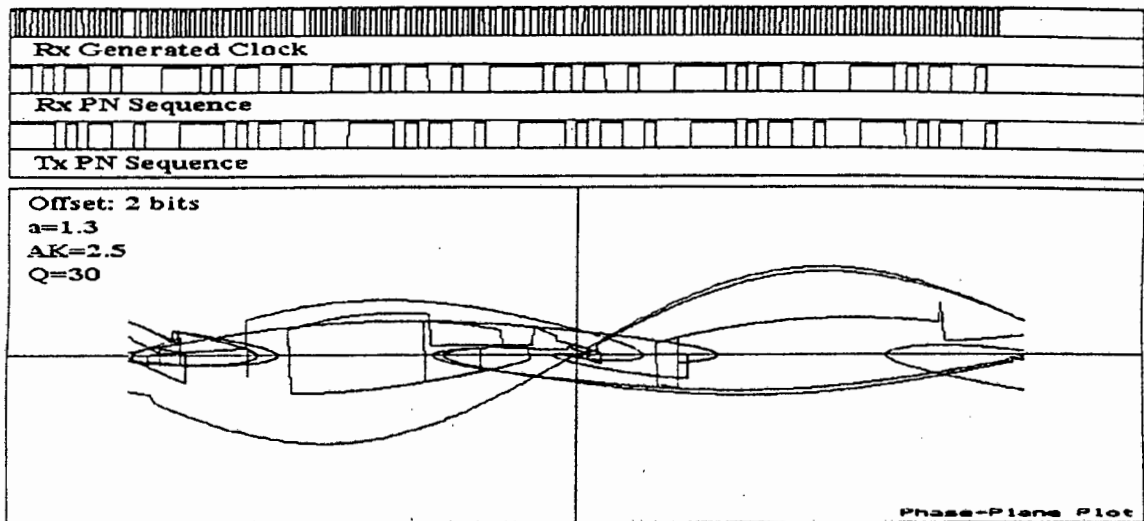


Figure 7.13: Phase-Plane Plot of Novel Loop ($AK = 2.5$). Lock was achieved in 50 seconds.

Figure 7.14 shows the case for an AK of 1.8. Lock was achieved within 11 seconds (0.11 milliseconds at 100 kHz) and maintained for a short period of time. The system was under-damped and the system could not maintain lock, but jittered. An important factor becomes clear when comparing figures 7.13 and 7.14. One would have expected that the bigger AK , the quicker the system should achieve lock. However, the relationship between a and AK is critical. By reducing AK and approaching a , the loop becomes underdamped and achieves lock quicker. However, the downside is that the loop is more sensitive to sudden changes and therefore the jitter is higher.

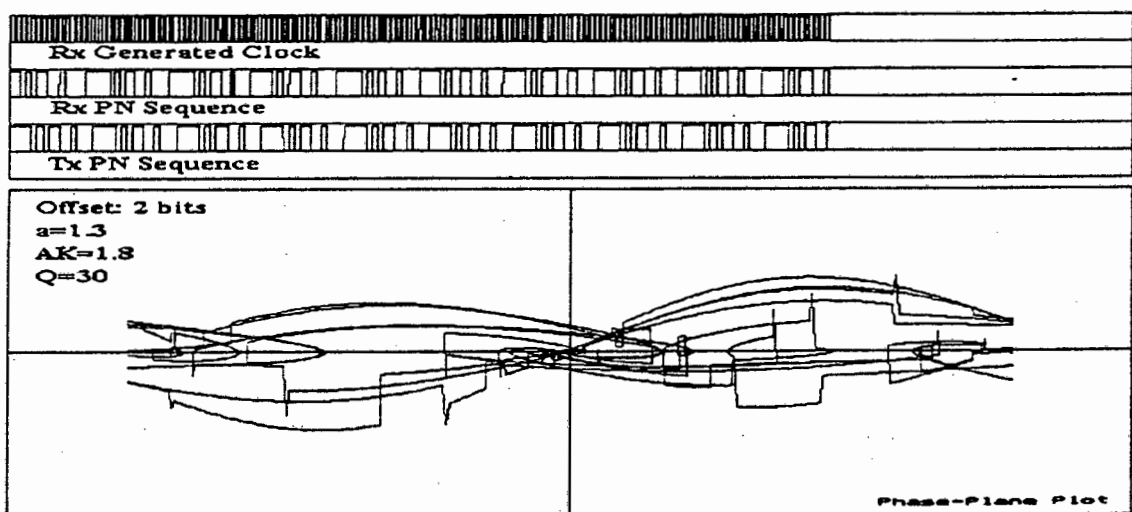


Figure 7.14: Phase-Plane Plot of Novel Loop ($AK = 1.8$). Lock was achieved relatively quickly, but lock was not maintained. The system was under-damped.

The value of AK was then increased to 3.14. The system is over-damped. Lock was achieved within 45 seconds (0.45 milliseconds at 100 kHz).

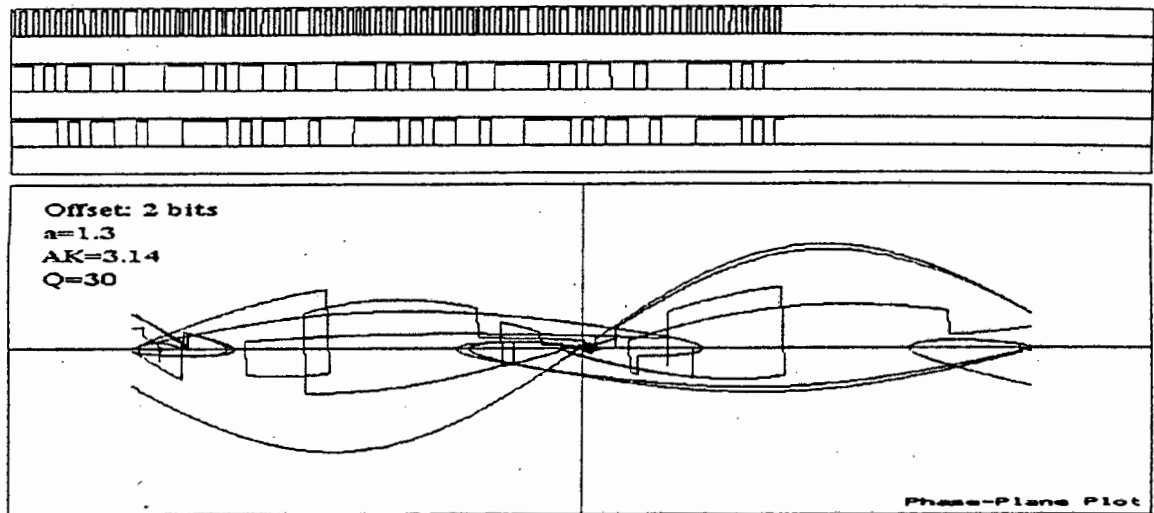


Figure 7.15: Phase-Plane Plot of Novel Loop ($AK = 3.14$). Lock was achieved within 45 seconds. The system was over-damped. The unique nature of each plot is still apparent.

From the three simulations, it is clear that benefit is gained from making K_{vco} larger, but at the same time it does reduce the overall stability of the loop. One fact that does become clear, is that a and AK are interdependent and that the relationship between the two determines the damping of the system.

7.2.3.1 Larger Code-Phase Offsets

Upto now, the values of a and AK had been selected to see how the system performs for an offset of 2 chips. In this section, two sets of values are applied to see the effect on locking time for an offset of 7 chips.

Figure 7.16 shows the case for an $a = 1.3$ and an $AK = 3.14$. The system achieved lock around 105 seconds (1.05 milliseconds at 100 kHz), a 20 second improvement on the locking time for an $a = 1.3$ and an $AK = 4$. Figure 7.17 shows the case for an $a = 1.3$ and an $AK = 2.5$. The system achieved lock in around 130 seconds (1.3 milliseconds at 100 kHz). Examining the phase-plane plots the system behaves more orderly in figure 7.17 than it does in 7.16.

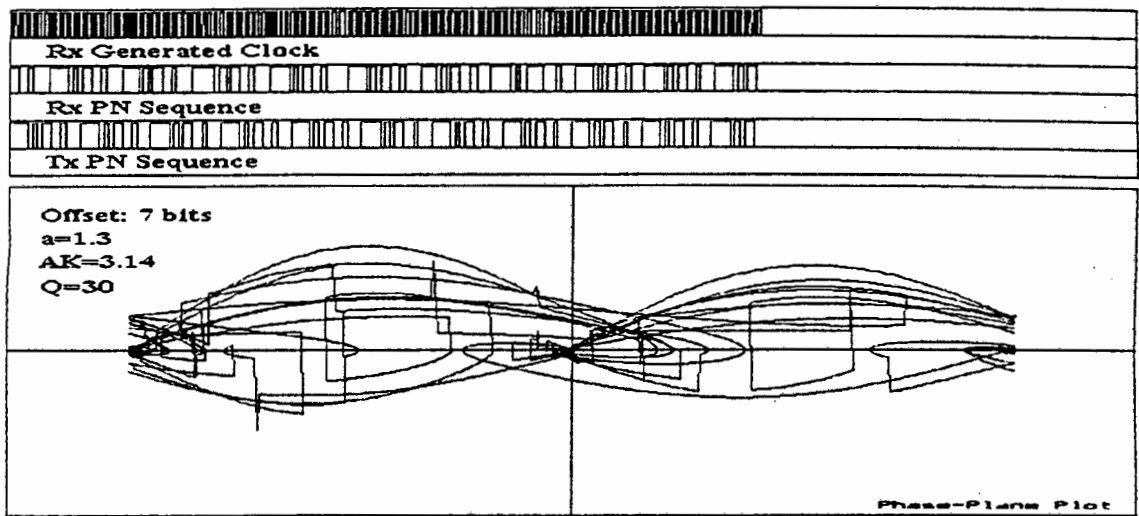


Figure 7.16: Phase-Plane Plot of Novel Loop ($a = 1.3, AK = 3.14$): Lock was achieved within 105 seconds.

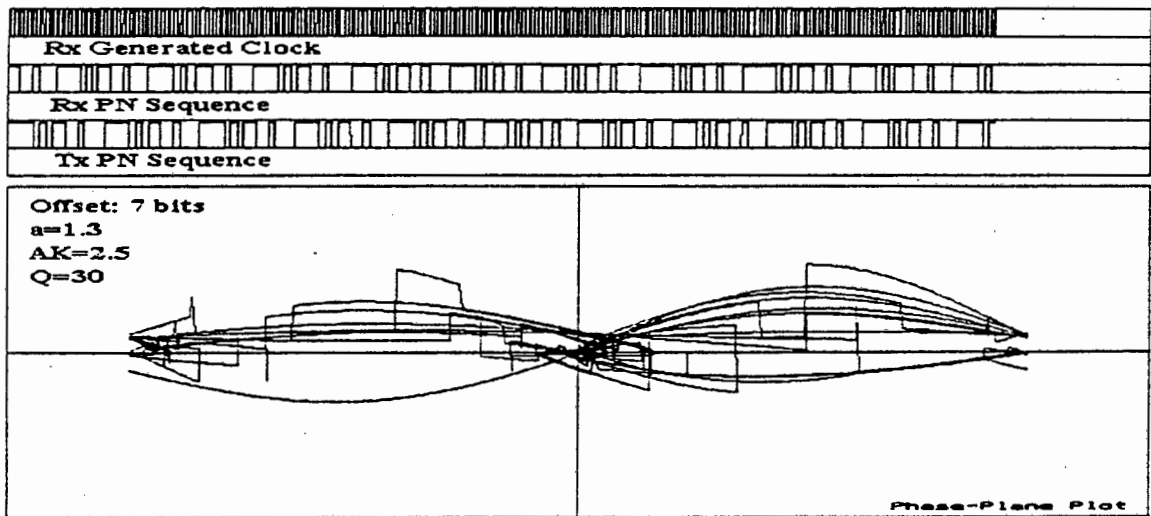


Figure 7.17: Phase-Plane Plot of Novel Loop ($a = 1.3, AK = 2.5$). The system achieved lock in 130 seconds.

7.2.3.2 Effect of Frequency Offset

The overall bandwidth of the phase-locked loop determines what the maximum range of the loop is. Although in this theoretical implementation it is possible to deal with any frequency offset, the reality is that the time required to achieve lock could translate into many cycles of the PN sequence.

Frequency offsets also have an effect on the convergence of the two sequences, although this convergence is also dependent on the code-phase offset between the sequences. Consider the case of a 15 bit sequence. If the receiver sequence lagged the incoming sequence by 2 chips, it would be make sense to have the VCO centre frequency higher than the chipping frequency of the transmitted sequence. On the other hand if the offset was 12 chips, the VCO centre frequency should be lower. In practice, however, the offset between the two sequences is unknown and therefore the only trade-off is to either make the VCO centre frequency higher or lower than the chipping frequency of the transmitted sequence.

To show the effect of changing the offset between the transmitted sequence chipping frequency and the VCO centre frequency, two examples are considered. Figure 7.18 shows the case for an offset of 0.01 Hz from the VCO centre frequency. From the figure, the system achieved lock within 15 seconds and maintains lock for only a short period. The fact the system lost lock relates to the selection of the loop filter and gain constants and the fact that the selected offset is on the fringe of the locking range. This can be solved by increasing the locking range of the PLL and widening the bandpass filter at the input.

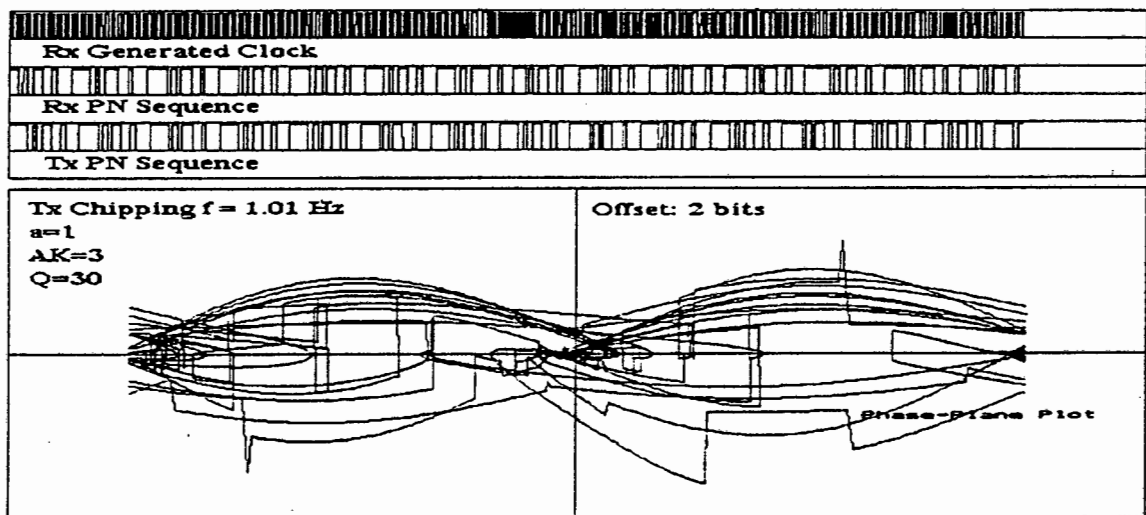


Figure 7.18: Phase-Plane Plot of Novel Loop: Frequency Offset. The frequency of the transmitted PN sequence was set above the centre frequency of the VCO. Initial lock was achieved in 15 seconds, but was not maintained.

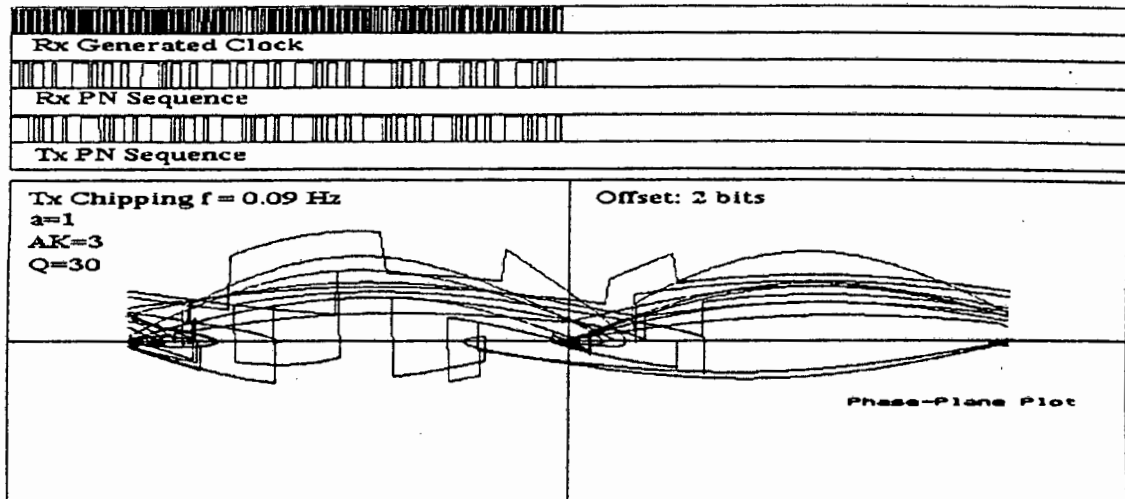


Figure 7.19: Phase-Plane Plot of Novel Loop: Frequency Offset. The frequency of the transmitted PN sequence was set below the centre frequency of the VCO. Initial lock was achieved in 15 seconds, but was not maintained.

In figure 7.19, the transmitted PN sequence chipping frequency is 0.01 Hz below the VCO centre frequency. Again the system achieves lock in just over 15 seconds, but does not maintain lock either. This result is not unexpected, since the system is symmetrical around the VCO centre frequency.

7.2.3.3 Effect of Q on the System

The Q of the bandpass filter plays a major role in the stability of the system. In figure 7.4 the system consists of two non-linear sinusoidal elements as opposed to the one found in the usual representation of the phase-locked loop. The cosine element can be considered the rogue in the overall design. Clearly as Q increases, so the influence of the cosine term on the system becomes less. Ideally one would consider making Q very large, but this reduces the flexibility in the system in changing frequency easily.

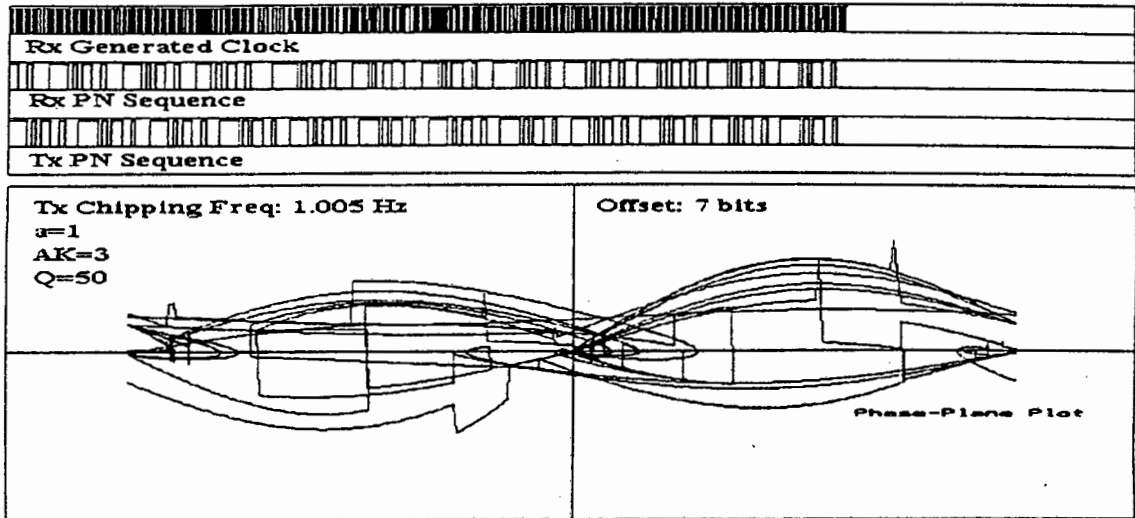


Figure 7.20: Phase-Plane Plot of Novel Loop: $Q=50$. Lock was achieved within 105 seconds. The effect of the cosine term in the loop equation is reduced with an increase in Q .

Figures 7.20 through 7.22 show the effect of Q on the overall locking performance and stability of the system for an offset of 7 chips and frequency offset of 0.005 Hz. For a $Q = 50$, the system achieved lock in around 105 seconds (1.05 milliseconds at 100 kHz). In the case of a $Q = 20$, the system achieved lock within 120 seconds. The last case is for a $Q = 10$. The system achieves lock more rapidly, but does not maintain lock. The system drifts due to the large contribution from the cosine term and therefore degrades the overall performance of the system.

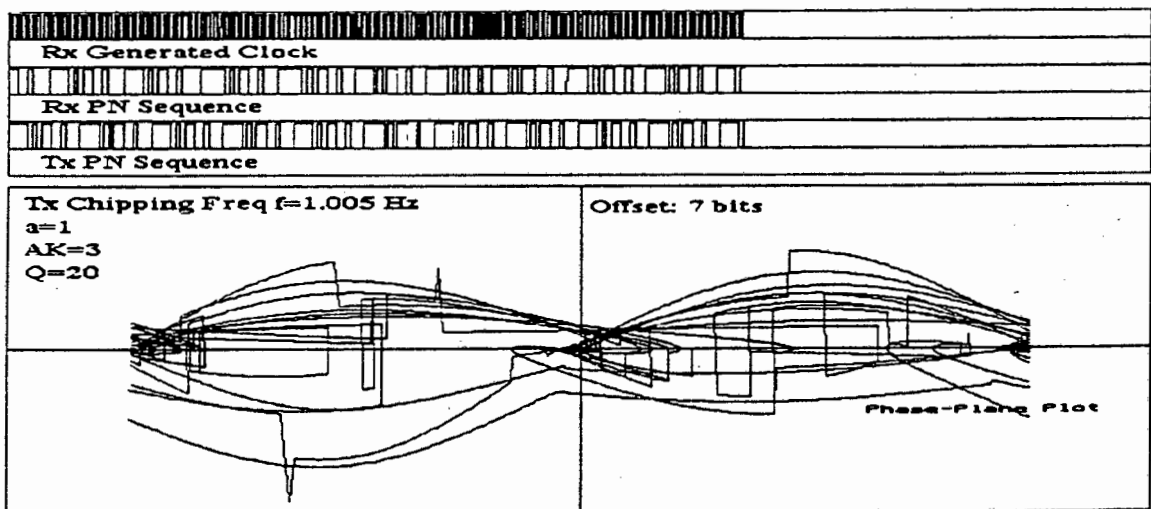


Figure 7.21: Phase-Plane Plot of Novel Loop: $Q=20$. Lock was achieved within 120 seconds. The cosine term in the loop equation has more effect at a lower Q .

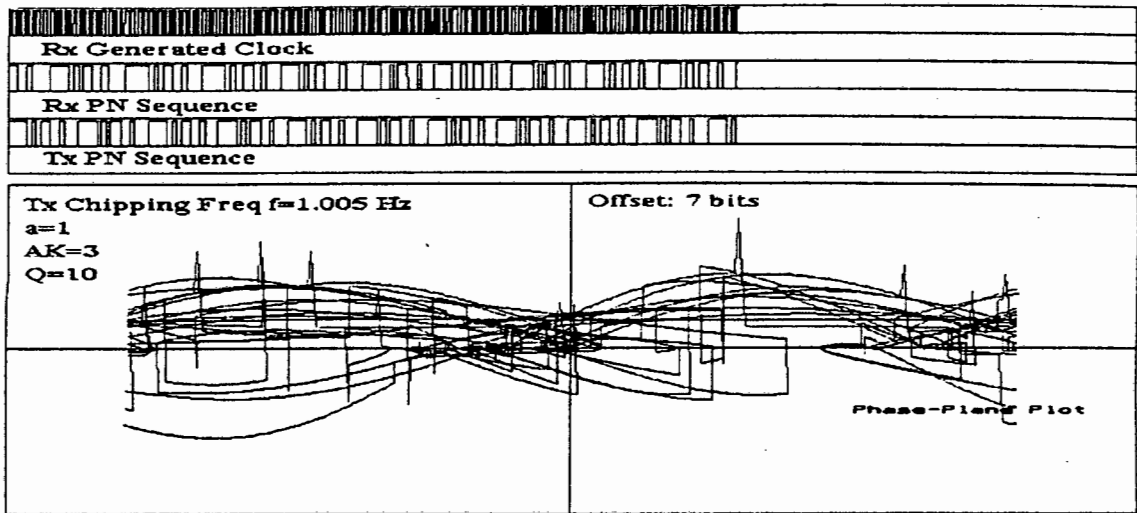


Figure 7.22: Phase-Plane Plot of Novel Loop: $Q=10$. Lock was achieved rapidly, but was not maintained. The cosine term in the loop equation reduces the overall stability of the system.

7.3 Summary

In this chapter the non-linear integro-differential equation for the spread spectrum synchronisation network was derived. It was highlighted that the complexity of the equation precludes any direct analysis of the equation. Phase-Plane techniques were applied and a number of case studies were presented to show the effect of the various system parameters on the locking performance of the system. Although, the phase-plane plots are difficult to analyse due to the complex trajectories of the system, they clearly show the chaotic nature of the synchronisation network and stress the fact that the phase-plane grid is solved using a random rather than deterministic walk. The locking time results were extrapolated to compare to the simulation results obtained from Tesla. The locking times were of the same order and therefore validate the various simulation models used.

Chapter 8

Data Demodulation, Carrier Recovery and Noise

In the previous chapters on the design and analysis of the Novel Spread Spectrum Synchronisation network, the issues of data recovery, carrier recovery and noise have been left out of the discussion. In this chapter, the low cost, novel network is extended to incorporate data demodulation and circuitry to prevent the loss of lock due to data inversion. The implementations are simulated with Tesla, and the results compared with the methods presented in chapter 2.

8.1 The Influence of Noise on Spread Spectrum Synchronisation

System design without noise is a different ball game from system design taking noise into account. Having designed a system that works both in theory and practice without the addition of noise, the next logical step is to consider the effect of noise on the system.

8.1.1 Factors Affecting Noise Performance

A number of criteria affect the noise performance of a synchronisation network under the influence of noise. Issues like the amount of filtering before the demodulation circuitry,

whether the spread spectrum tracking circuitry is coherent or non-coherent, the use of differentiators and so forth all need to be considered. Having stated this, the following factors in the design will have a definite effect on the noise performance of the synchronisation network:

8.1.1.1 The System is coherent

If the spread spectrum is modulated onto a carrier, the carrier needs to be recovered before spread spectrum demodulation takes place. This implies that the initial signal to noise requirements are determined by the carrier recovery network.

8.1.1.2 Detection of Received Spreading Wave-form

To achieve synchronisation the edges of the transmitted spreading sequence are required. Noise distorts these edges and makes them undetectable without the use of filtering and hard-limiting. To obtain edges the combination of a low-pass filter and comparator are required. To reduce the noise sufficiently, a wideband brick-wall filter would be ideal. However a brick-wall filter is not practical, and therefore a higher-order low-pass filter would be required.

It is possible that incorrect detection will occur and this will affect the synchronisation process. To determine the point of failure it is possible to treat the spreading sequence as a data wave-form and determine a bit error count. It must be pointed out however that the sequence is periodic in nature and therefore can not be consider to be a true statistical measure. To show the point of failure the signal-to-noise ratio will be decreased until the system fails to synchronise

8.1.1.3 Differentiation

The use of differentiation or the more flexible novel wide-band signed edge detector places a noise limit on the type of applications in which the novel synchronisation network can be applied. A certain signal-to-noise ratio is required to obtain the correct edges. Extra edges are obtained as the threshold of the comparator may be exceeded for a short period of time due to the added noise. However, statistically the noise should average out to zero, i.e. the number of incorrect positive going and negative going edges should cancel out. Therefore it

should be possible to operate the network at a lower signal to noise ratio than what would be intuitively expected.

8.1.2 Simulation under Noisy Conditions

Tesla is equipped with a noise generator based on the technique described by Rabiner and Gold [47]. The Gaussian noise sample is determined from the sum of twelve pseudo-random noise variables with a uniform distribution. This yields a maximum peak value of six times the selected RMS noise voltage.

8.1.2.1 Simulation Diagram

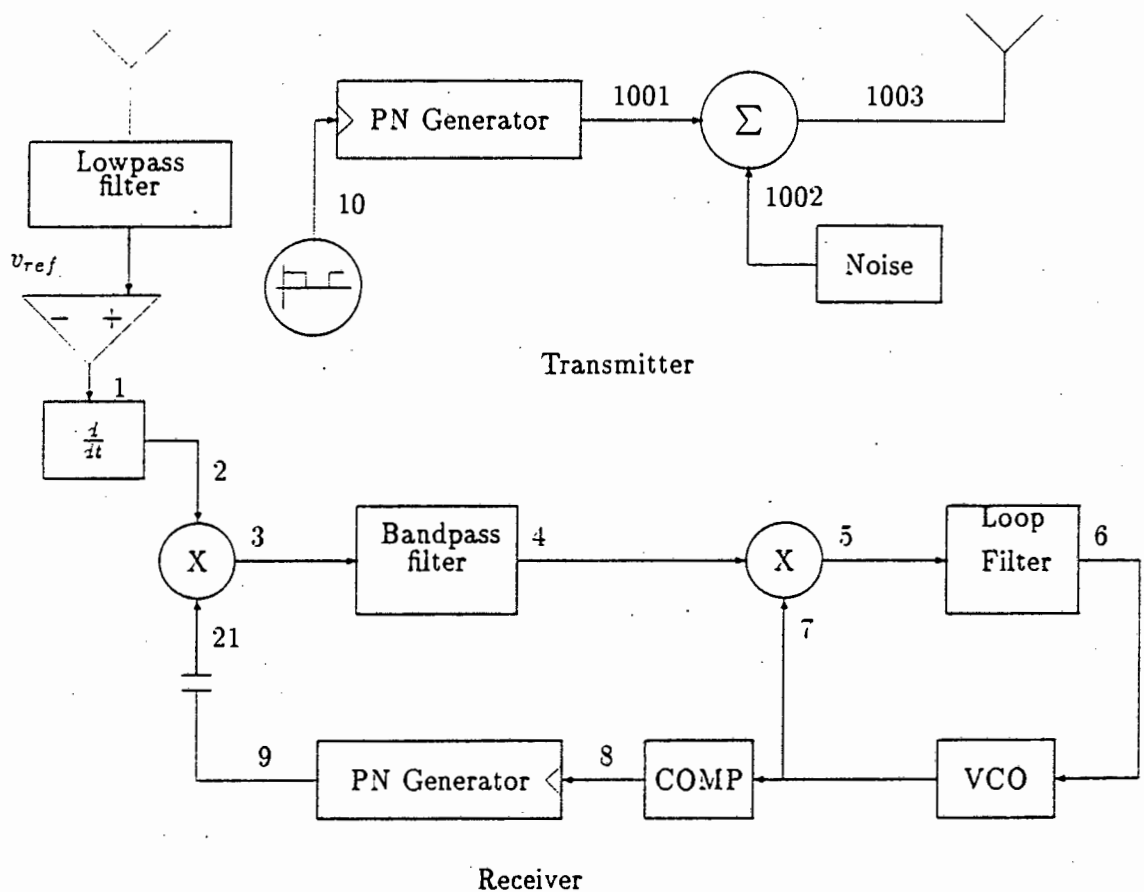


Figure 8.1: Simulation Model to Investigate the Effect of Noise

The simulation diagram is shown in figure 8.1. The transmitted sequence (1001) is summed

with the noise (1002) and the result fed into a low pass filter. The output of the filter (1004) is compared to a fixed voltage reference, the result of which is differentiated and fed into the synchronisation network. For the purpose of simulation the transmitted sequence (1001) and the locally generated sequence (21) are correlated to determine the point of lock. It should be noted that the sequences must be aligned for a lock condition to be declared. All signal-to-noise ratios are given in a bandwidth of half the simulation bandwidth (i.e. $B_{noise} = \frac{1}{2T_s} = 2.5$ MHz).

8.1.3 Simulation Parameters: Case One

Parameter	Value
Sequence Length	15
Code Offset	7 chips
Tx Clock Frequency	101 kHz
Signal Power	6.25 Watts
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	99 kHz
Receiver LPF Type	Single Pole RC

Table 8.1: Simulation Parameters - Method 2 under Noise (1)

8.1.3.1 Signal-to-Noise Ratio of -12 dB

The signal-to-noise ratio was set to -12 dB and the system was simulated using the parameters defined above. Figure 8.2 shows a 20 millisecond time segment of the transmitted sequence at node 1001, the filtered and limited sequence at node 1 and the locally generated PN sequence at node 9. The system achieved lock in 18 milliseconds and maintained lock for the simulation period. This is shown in figure 8.3. Taking a closer look at node 1, it is clear that errors are made in the estimation of the received chip. This however did not affect the system once locked had been achieved. Comparing it to figure 5.7 in chapter 5, the system took four times longer to synchronise.

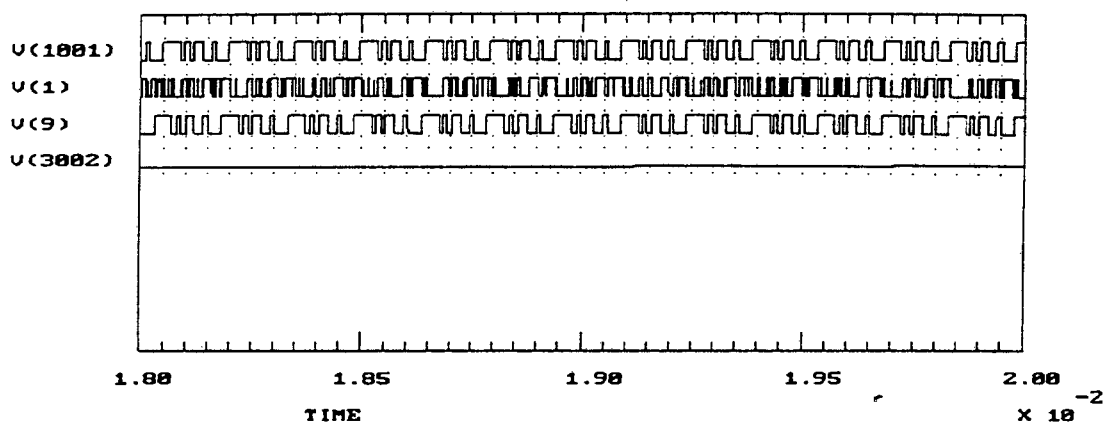


Figure 8.2: Transmitted Sequence, Received Sequence and Receiver Sequence. V(1) shows the estimated sequence at the output of the comparator. A number of errors are present.

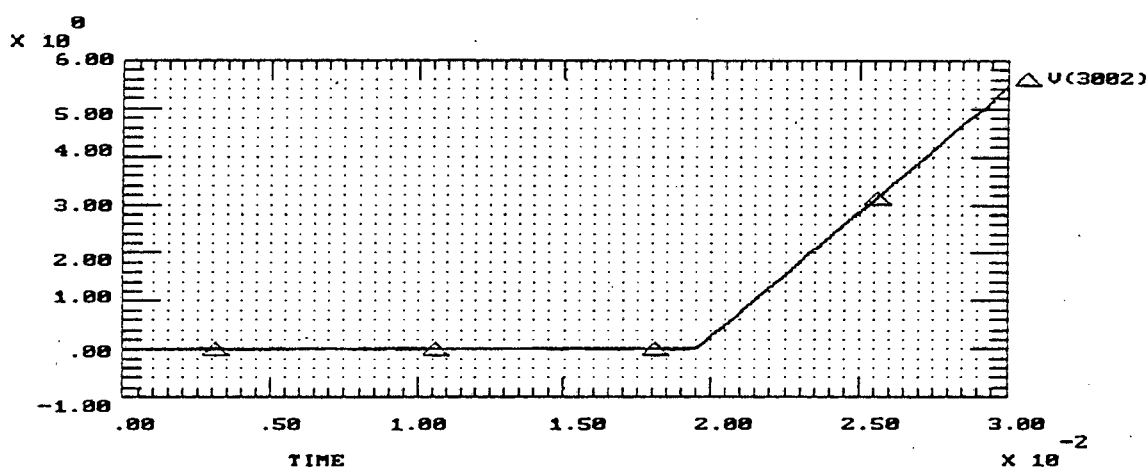


Figure 8.3: Correlator Output. The system achieved lock in 18 milliseconds. The point of lock is determined from the point the curve increases linearly.

Figure 8.4 shows the spectrum at the output of the bandpass filter (node 4). The frequency spike at 101 kHz is an indication of the system having achieved lock. For this frequency component to have occurred, the input to the bandpass filter had to be a series of positive or negative spikes. The frequency information around the spike is due to the shifting of the VCO frequency and the presence of noise. Figure 8.5 shows the time waveform at the output of the bandpass filter. Comparing to figure 5.5, the effect of noise is clearly seen in the envelope of the wave-form.

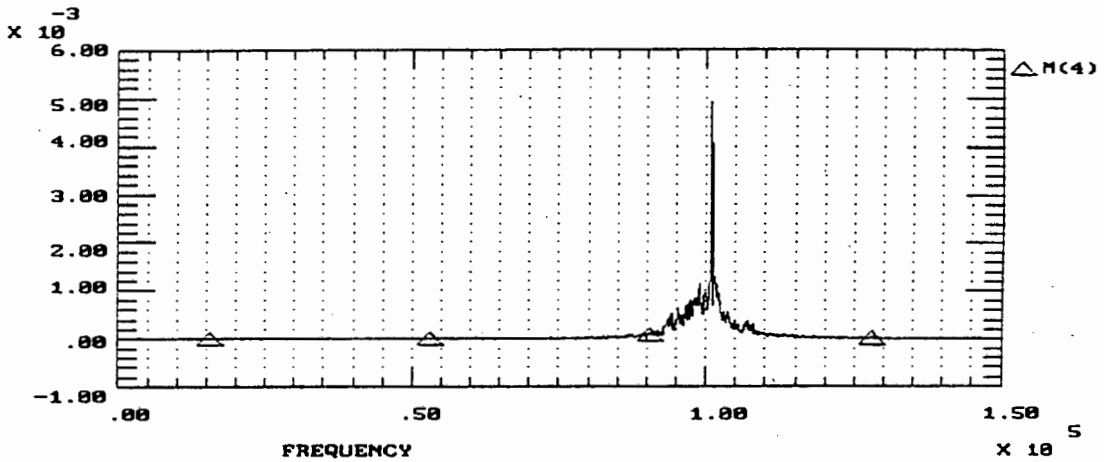


Figure 8.4: Fourier Spectrum at the Output of the Bandpass Filter (Node 4). The dominant frequency component occurs only when the system is in lock. The frequency information around the dominant frequency component is due to noise and the impulse train.

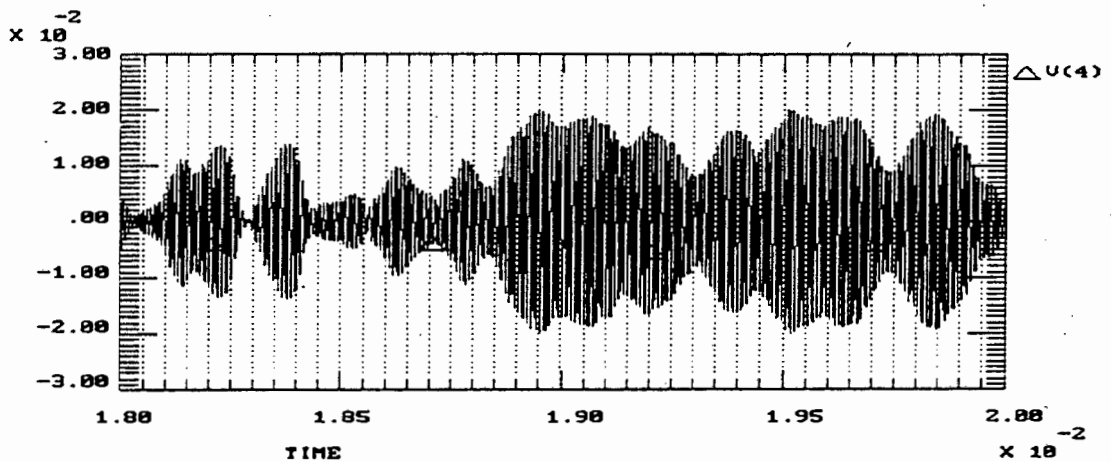


Figure 8.5: Output of bandpass filter (Node 4). The effect of noise is clearly seen on the envelope of the waveform.

8.1.3.2 Signal-to-Noise Ratio of -16.1 dB

The signal-to-noise ratio was set to -16.1 dB and the system was simulated using the parameters defined in table 8.1. Figure 8.6 shows a 20 millisecond time segment of the transmitted sequence at node 1001, the filtered and limited sequence at node 1 and the locally generated PN sequence at node 9. The system achieved lock in 25 milliseconds and maintained lock. This is shown in figure 8.7. Taking a closer look at node 1, it is clear that errors are made in the estimation of the received chip. This however did not affect the

system once lock had been achieved: Locking time increased with an increase in noise, not an unexpected result.

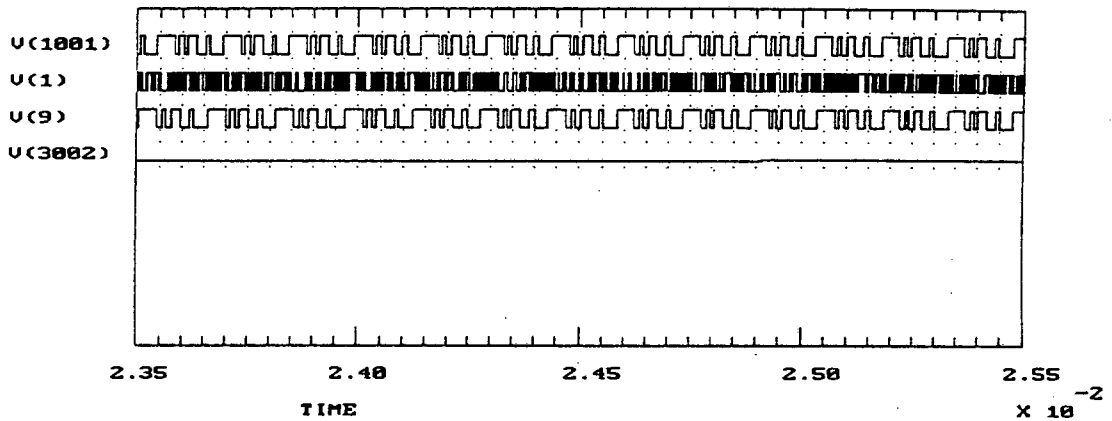


Figure 8.6: Transmitted Sequence, Received Sequence and Receiver Sequence. V(1) shows the output of the comparator. The waveform looks nothing like the transmitted waveform.

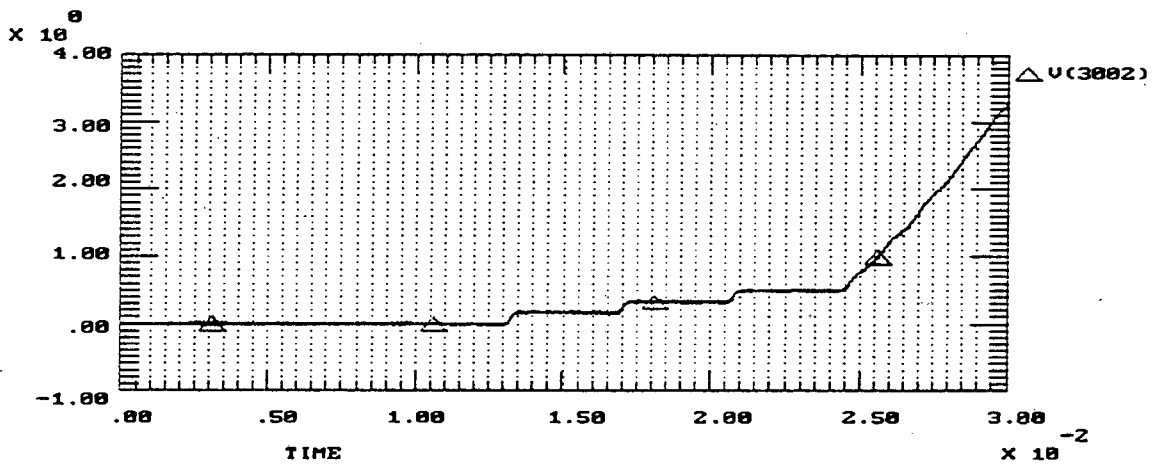


Figure 8.7: Correlator Output. The system achieved lock in 25 milliseconds.

Figure 8.8 shows the spectrum at the output of the bandpass filter (node 4). The frequency spike at 101 kHz is an indication of the system having achieved lock. For this spike to have occurred, the input to the bandpass filter had to be a series of positive or negative spikes. The frequency information around the spike is due to the shifting of the VCO frequency and noise. Figure 8.9 shows the time waveform at the output of the bandpass filter. Comparing to figure 8.5, the dramatic effect of noise is clearly seen in the envelope of the wave-form.

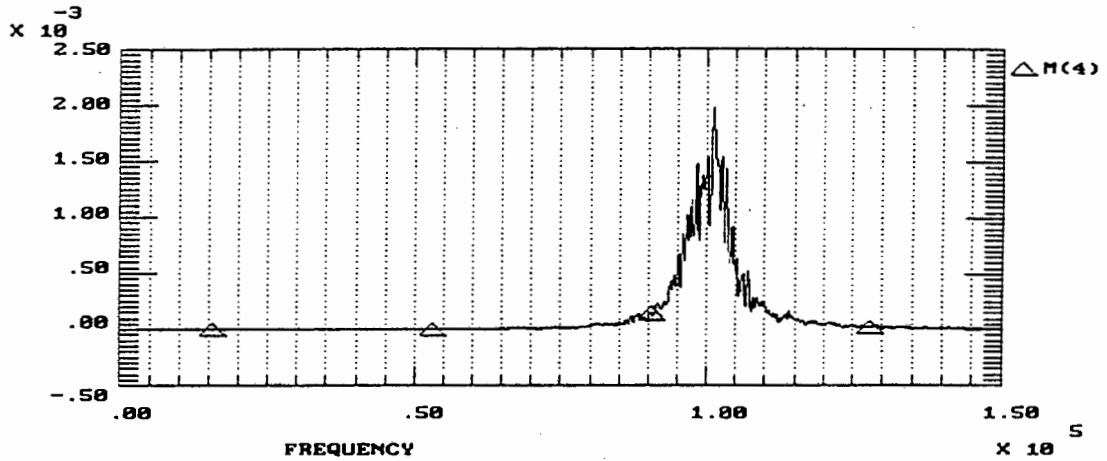


Figure 8.8: Fourier Spectrum at the Output of the Bandpass Filter (Node 4). The dominant frequency component occurs only when the system is in lock. The frequency information around the dominant frequency component is due to noise and the change in frequency caused by the impulse train before lock.

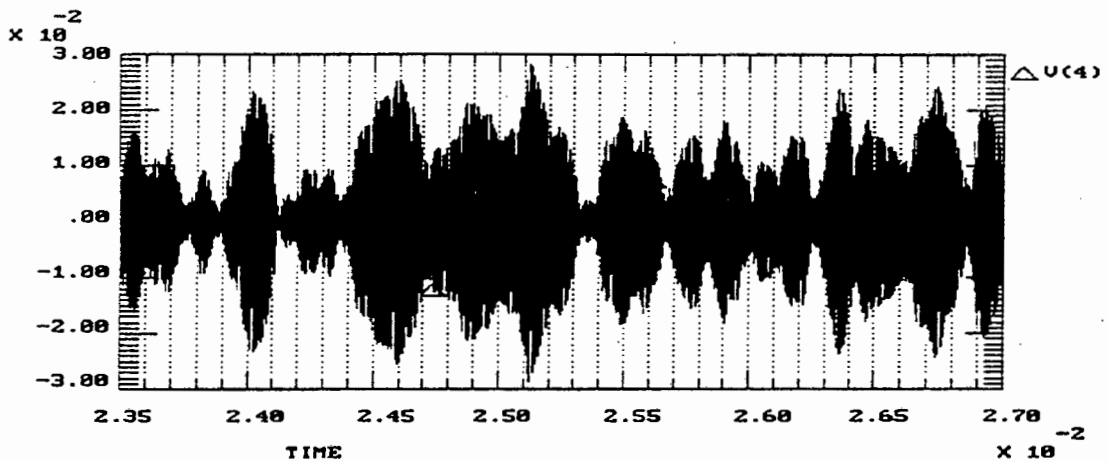


Figure 8.9: Output of bandpass filter (Node 4). The effect of noise is clearly seen on the envelope of the waveform.

8.1.3.3 Signal-to-Noise Ratio of -18 dB

The signal-to-noise ratio was decreased to -18 dB. The system was unable to achieve lock within the simulation time window of 40 milliseconds. It is conceivable the system could have achieved lock, but it is felt that a 40 millisecond locking time is reasonable.

8.1.4 Simulation Parameters: Case Two

It was stated that the amount of filtering is critical in the design of the synchronisation network in noisy conditions. In the simulations to follow the lowpass filter was replaced with a 4th order Butterworth. The simulation parameters for case two are given in table 8.2.

Parameter	Value
Sequence Length	15
Code Offset	7 chips
Tx Clock Frequency	101 kHz
Signal Power	6.25 Watts
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	99 kHz
Receiver LPF Type	4th Order Butterworth

Table 8.2: Simulation Parameters - Method 2 under Noise (2)

8.1.4.1 Signal-to-Noise Ratio of -18 dB

The signal-to-noise ratio was set to -18 dB and the system was simulated using the parameters defined in table 8.2. Figure 8.10 shows a 20 millisecond time segment of the transmitted sequence at node 1001 and the locally generated PN sequence at node 21. The system locked in 24 milliseconds and maintained synchronisation. This is shown in figure 8.11.

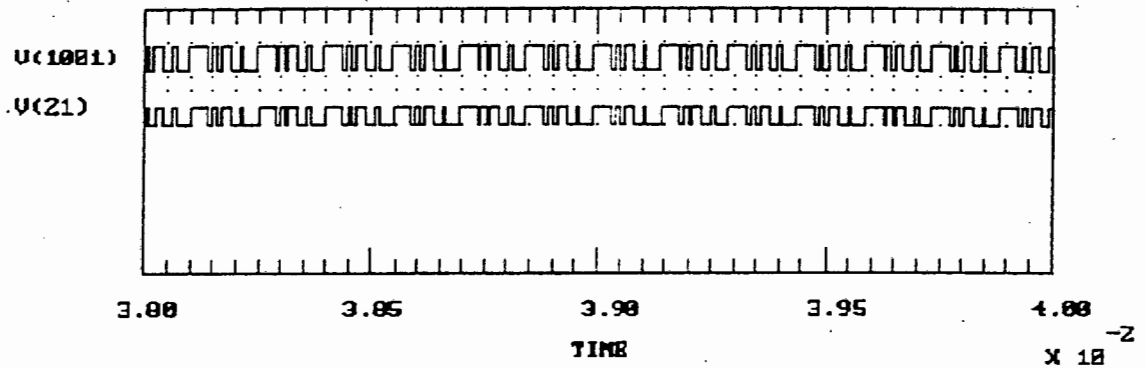


Figure 8.10: Transmitted Sequence and Locally Generated Sequence.

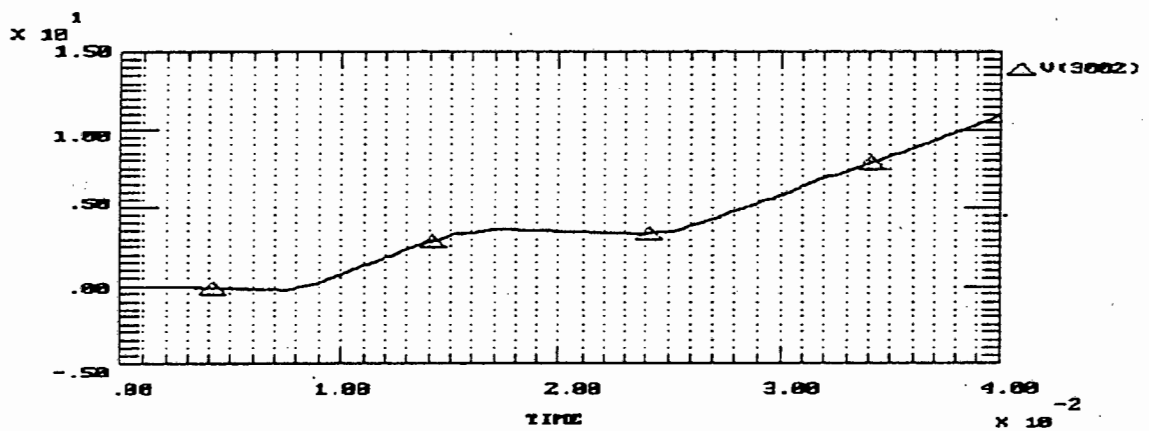


Figure 8.11: Correlator Output. The system initially achieved lock in 9 milliseconds, but lock was not maintained. The system re-acquired lock at 24 milliseconds and maintained lock.

An interesting characteristic of the system is that it achieved lock in 9 milliseconds, before losing lock at 15 milliseconds. Figure 8.12 shows the case before lock occurred. The shifting of the sequence is clearly seen. Figure 8.13 shows the case where the system locked the first time.

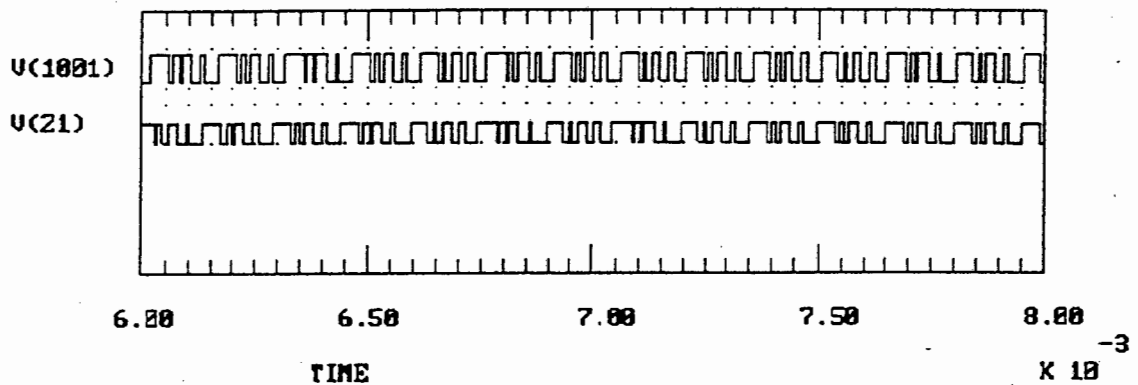


Figure 8.12: Transmitted Sequence and Receiver Sequence Before Lock. The pulling in of the PN sequences is clearly illustrated.

As expected by making the roll off tighter, the amount of out of band noise was reduced enough to allow the system to achieve lock at -18 dB. The question arises, "if by doubling the order to 8 can the signal-to-noise ratio be reduced further?"

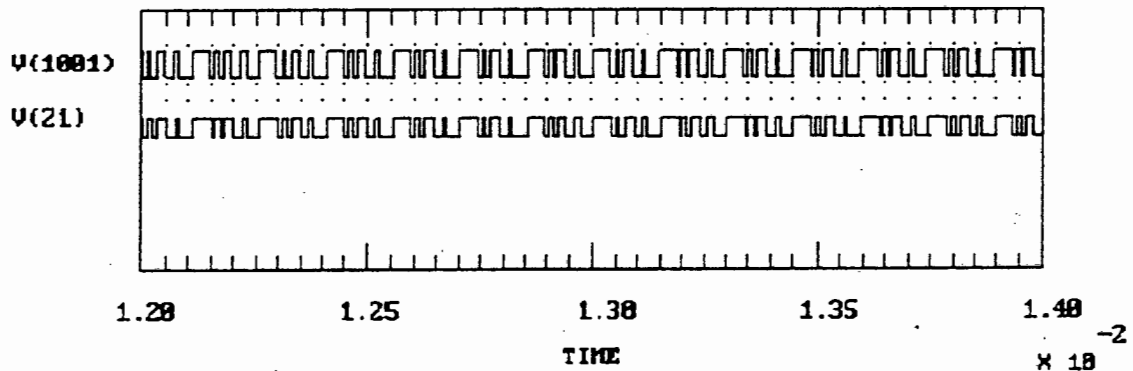


Figure 8.13: Transmitted Sequence and Receiver Sequence After Lock.

8.1.5 Simulation Parameters: Case Three

It is intuitively expected that an increase in the order of a filter will allow the system to synchronise at a lower signal-to-noise ratio. In the simulations to follow the lowpass filter was replaced with a 4th order Butterworth. The simulation parameters for case two are given in table 8.3.

Parameter	Value
Sequence Length	15
Code Offset	7 chips
Tx Clock Frequency	101 kHz
Signal Power	6.25 Watts
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	99 kHz
Receiver LPF Type	8th Order Butterworth

Table 8.3: Simulation Parameters - Method 2 under Noise (3)

8.1.5.1 Signal-to-Noise Ratio of -16.1 dB

The signal-to-noise ratio was set to -16.1 dB and the system was simulated using the parameters defined in table 8.3. Figure 8.14 shows the output of the correlator. The system achieved lock in 4 milliseconds. This is a dramatic improvement over the 20 milliseconds required using the single pole RC filter.

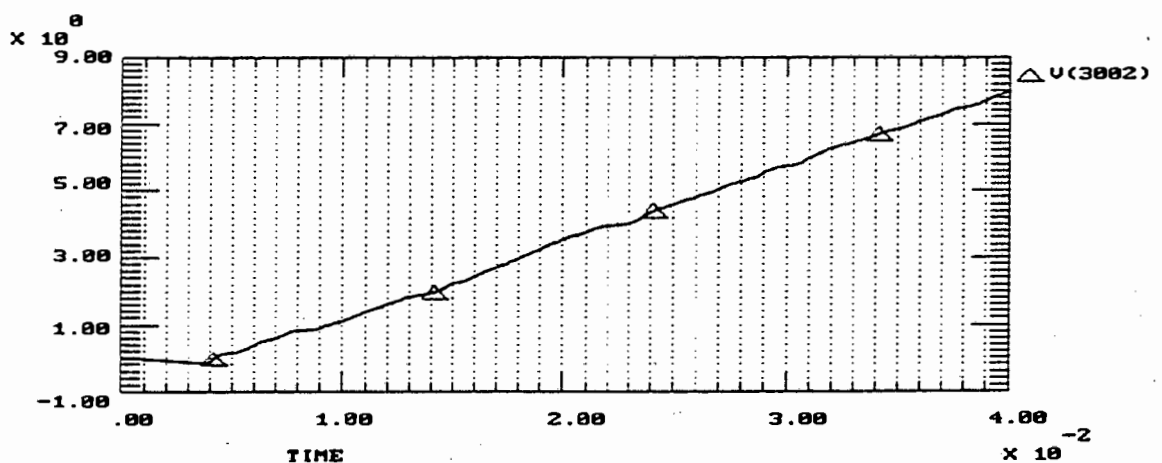


Figure 8.14: Correlator Output. Lock was achieved 4 milliseconds at a signal-to-noise ratio of -16.1 dB.

8.1.5.2 Signal-to-Noise Ratio of -17.1 dB

The signal-to-noise ratio was set to -17.1 dB and the system was simulated using the parameters defined in table 8.3. Figure 8.15 shows the output of the correlator. The system achieved lock in 24 milliseconds.

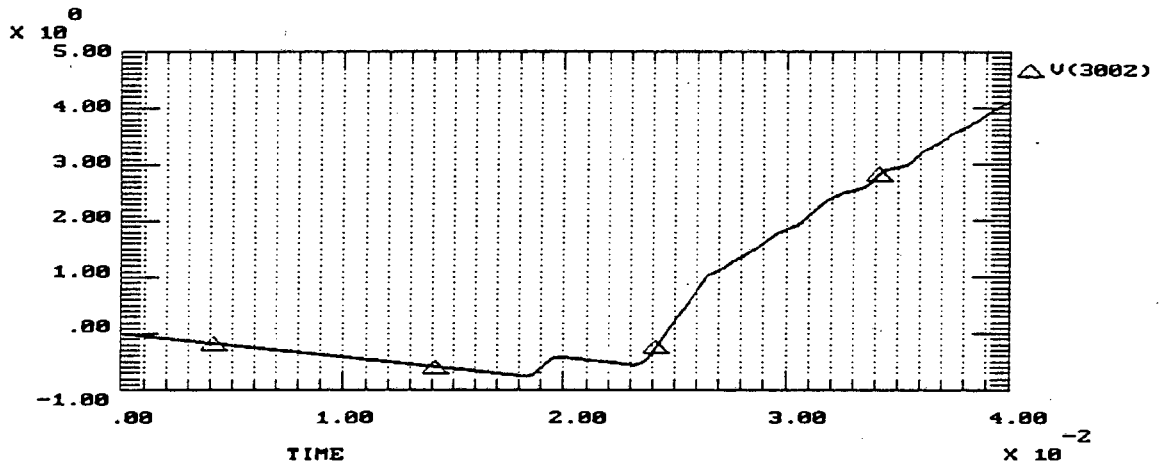


Figure 8.15: Correlator Output. Lock was achieved 24 milliseconds at a signal-to-noise ratio of -17.1 dB.

8.1.5.3 Signal-to-Noise Ratio of -18 dB

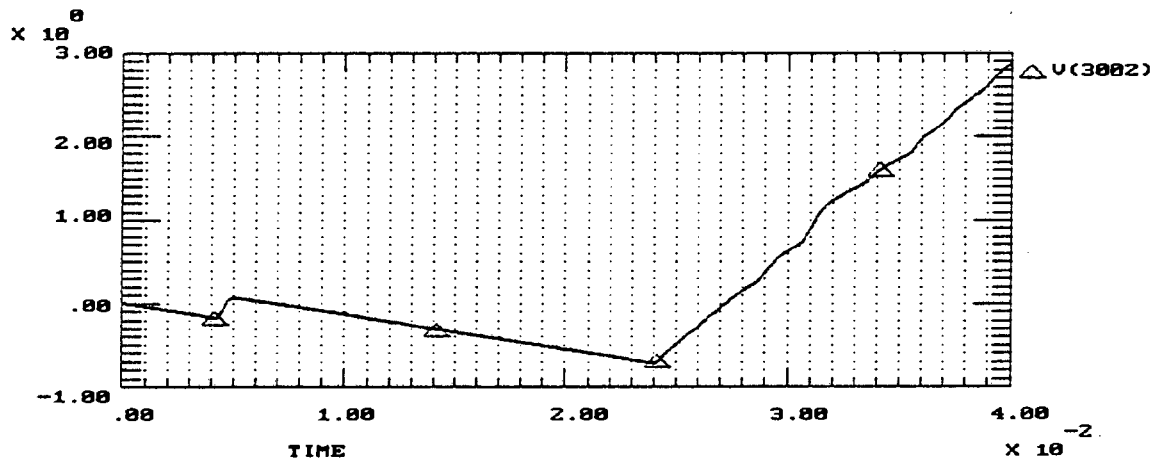


Figure 8.16: Correlator Output. Lock was achieved 24 milliseconds at a signal-to-noise ratio of -18 dB.

The signal-to-noise ratio was set to -18 dB and the system was simulated using the parameters defined above. Figure 8.16 shows the output of the correlator. The system achieved

lock in 24 milliseconds. This is a marginal improvement over the locking time when a 4th order filter was used.

8.1.5.4 Discussion

From the case studies, it was found that by increasing beyond -19 dB, the system did not achieve lock. The order of the filter, as expected, determined the signal-to-noise at which the system failed to achieve lock within the required 40 milliseconds. It was also shown that doubling the order of the filter from 4 to 8 had little effect on locking time. However by using a higher order filter over an RC filter, the locking time was greatly improved and the system achieved lock at lower signal-to-noise ratios. The weakness of the system is clearly due to the fact that the detection of the incoming sequence is required. By using a filter matched to a PN bit, it should be possible to operate at a lower signal-to-noise ratio.

The noise does contribute to the locking process by introducing extra spikes. These extra spikes aid in the shifting of the VCO frequency and therefore the evaluation of the frequency-code phase grid. However, the effect introduced by the extra spikes should average out, thereby allowing the system to lock with correct frequency and code phase. To illustrate the fact that the extra spikes introduced into the system do not cause the system to lose lock, consider figures 8.17 through 8.20. (Note: The inversion is due to the comparator.) The simulation parameters are from case one.

In figure 8.17 the number of errors are few. As the noise is increased the number of false chips introduced increases. In figure 8.18 the sequences are clearly identifiable.

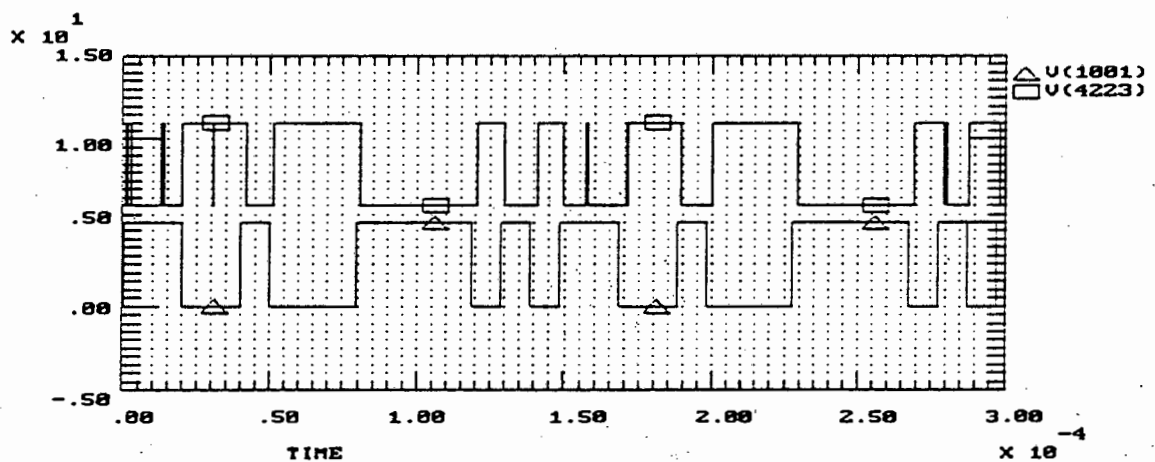


Figure 8.17: Errors due to noise: -7 dB

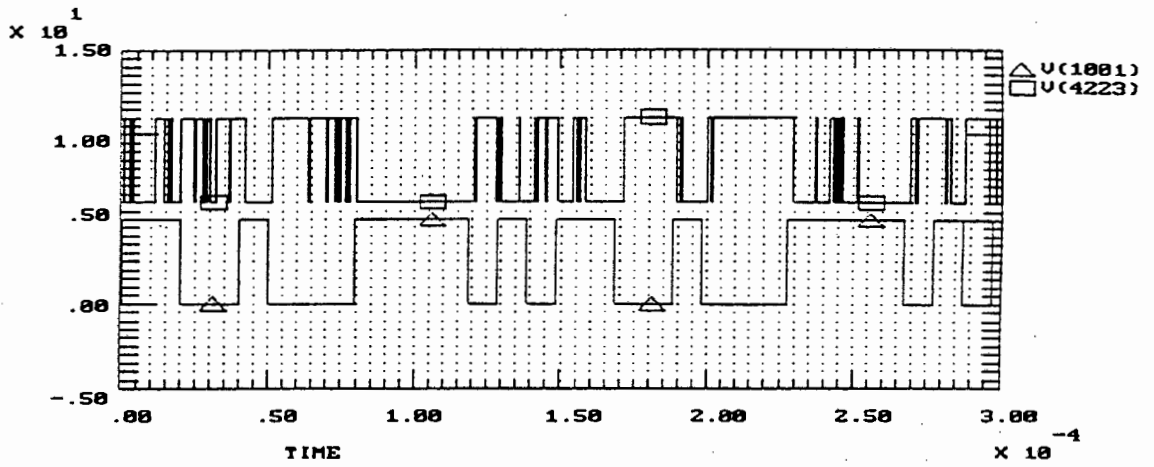


Figure 8.18: Errors due to noise: -12 dB

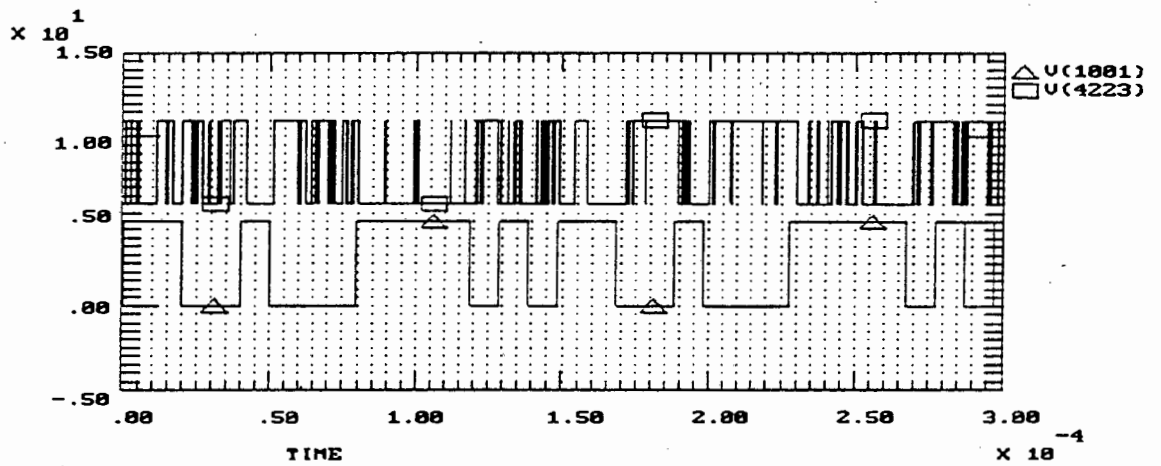


Figure 8.19: Errors due to noise: -16.1 dB

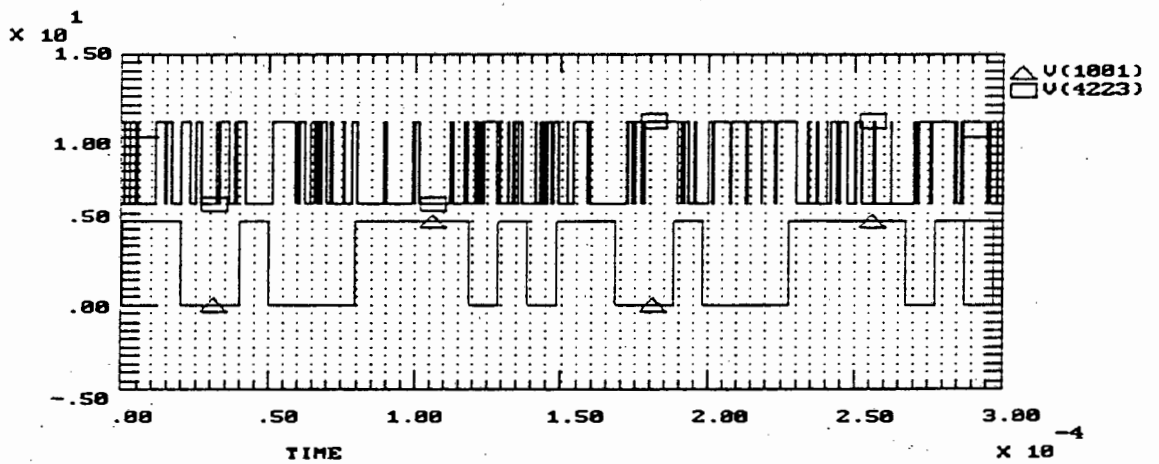


Figure 8.20: Errors due to noise: -18 dB

In figure 8.19 errors occur regularly. Although the sequences are aligned, the detected sequence differs from the received sequence. In figure 8.20 the system did not achieve lock in the required 40 milliseconds. The number of errors in the detected sequence was too large to enable the system to achieve lock.

8.1.5.5 Comparison with RASE

Since both the novel synchronisation system and RASE both required the detection of incoming chips and are both implemented to operate at base-band, it is realistic to draw some comparisons in how they perform under the influence of noise (with no Doppler present). In Chapter 2, the locking time for RASE was given as (assuming that noise does not give false alarms, etc):

$$T_a = (T_e + nT_c)2^n \quad (8.1)$$

If the probability of false alarm and false dismissal is taken into account, the average locking time is:

$$T_a = \frac{T_e + nT_c}{p^n(1 - p_{fa})(1 - p_{fd})} \quad (8.2)$$

where:

p_{fa} is the probability of false alarm

p_{fd} is the probability of false dismissal

To determine the values of p_{fa} and p_{fd} is difficult, since these parameters are dependent on the examination period generator, the threshold detector and other design factors. The comparisons that follow will assume that these values are small (0.2). It should be noted however that the locking times calculated for RASE will be optimistic, while the values given for the novel synchroniser are actual simulation results. It should also be noted that the examination period T_e was selected to be equal to one PN period, whereas in typical systems it could be multiple periods. Table 8.4 gives the locking times. If Doppler were to be include the locking times of the Novel Synchroniser would not change dramatically, while those for RASE will.

Case#	SNR_i dB	SNR_o dB	p	T_{novel} ms	T_{rase} ms
1	-12.0	+1.0	0.80	18	0.800
1	-16.1	-3.1	0.75	25	0.940
2	-18.1	-5.0	0.60	24	2.300
3	-16.1	-3.1	0.75	4	0.940
3	-17.1	-4.1	0.70	24	1.300
3	-18.0	-5.0	0.60	24	2.300

Table 8.4: Locking times of RASE and the Novel Synchroniser

where SNR_i is the signal-to-noise ratio in half the simulation bandwidth and SNR_o is the signal-to-noise ratio in a bandwidth of $\frac{1}{T_c}$ and p is the probability of correctness (see [22]).

8.2 Modulating the Spread Spectrum Signal

In the preceding chapters, the analysis and design has concentrated on the base-band model. Coherent demodulation techniques required by the specific implementations of the novel synchronisation network, imply that synchronisation of the PN sequences must occur at base-band. The fact that the synchronisation of the carrier is required before the synchronisation of the spreading wave-form implies that the system signal-to-noise ratio is determined by the component which fails first at a given signal-to-noise ratio. For example if the carrier tracking fails at -15 dB and the spread synchronisation network at -16 dB, the lowest signal-to-noise ratio at which the system can function is set by the carrier tracking network.

To explain why this is true, it is necessary to re-examine the implementation of spread spectrum systems. Dixon [10] and Gardner [48] state that the minimum signal-to-noise ratio a phase-locked loop can operate at is 0 dB in the noise bandwidth B_L . In non-coherent spread spectrum systems, carrier tracking is not performed, since the discriminator employed is not sensitive to data modulation or carrier phase (see figure 2.15). The fact that since no carrier tracking is included, but only a local reference, the noise performance of the tracking loop is affected.

Gaudenzi and Luise [34] improved the noise performance of the delay-lock loop by performing carrier tracking after the two sequences were in near-alignment (initially the system operates in a non-coherent mode). This implies that the signal-to-noise ratio at the input to the carrier tracking loop includes the processing gain, and therefore carrier tracking takes place at a much higher signal-to-noise ratio than other coherent techniques. To illustrate this fact, consider the following example:

Assume that the input signal-to-noise ratio to the demodulator of the spread spectrum system (with a 20 dB processing gain) after initial filtering and amplification is -20 dB in a bandwidth of 100 kHz. The carrier frequency is 1 MHz and the carrier tracking loop has a noise bandwidth of 20 kHz. Therefore the signal-to-noise ratio in the carrier tracking loop's noise bandwidth is -13 dB. According to Dixon it is not possible to track the carrier at this signal-to-noise ratio. However, after despreading, the signal-to-noise ratio is 0 dB. If the despread signal is now fed to the carrier tracking loop, the signal-to-noise ratio in the loop noise bandwidth would be 7 dB.

The analysis and design of carrier recovery networks is well documented. Gardner [48] discusses and analyses various implementations of phase-locked loops. In chapter 6 the analysis of phase-locked loops was briefly discussed. When designing phase-locked loops to work in noisy conditions, a third parameter enters the equation and that is noise bandwidth. (The other parameters were damping and loop bandwidth.)

Assuming a filter of the form,

$$F(s) = \frac{s\tau_2 + 1}{s\tau_1} \quad (8.3)$$

the primary design formula for a second order phase-locked loop are:

$$\omega_n = \frac{K}{\tau_1} \quad (8.4)$$

$$\eta = \frac{\tau_2}{\omega_n} \quad (8.5)$$

$$B_L = \frac{1}{2}\omega_n\left(\eta + \frac{1}{4\eta}\right) \quad (8.6)$$

8.2.1 Loop Design

For the design, a damping (η) of 0.5 was selected. This implies that the loop is marginally underdamped. The relationship between noise bandwidth and the natural frequency, using equation 8.6, is

$$B_L = \frac{\omega_n}{2}$$

τ_1 determines the loop bandwidth of the system. The choice of τ_1 depends on the environment that the system is required to operate in, the required pull in-range and noise-bandwidth. In the simulation the primary factor that is of concern is noise and therefore noise bandwidth. For the simulation, the noise bandwidth was selected as 20 kHz. Therefore ω_n is 40 krad/s. Using equation 8.5, and solving for τ_2 gives a value of $2.5 \cdot 10^{-5}$. τ_1 was selected as $5.3 \cdot 10^{-5}$, giving a K of 13 kHz/Volt.

8.2.2 Block Diagrams

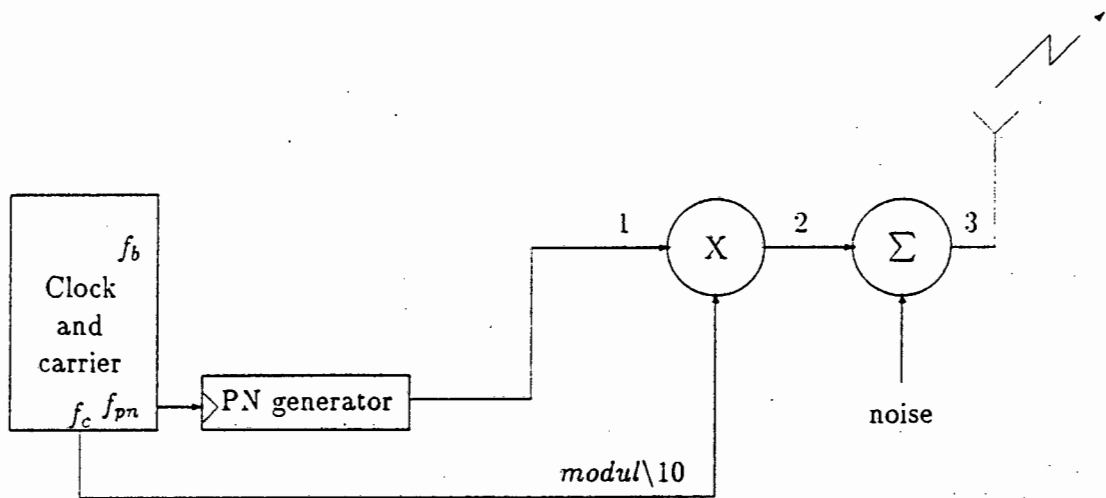


Figure 8.21: Transmitter

The transmitter, figure 8.21, consists of a signal source that generates the carrier and chipping frequencies. The carrier ($modul\10$)¹ is spread by the maximal length pseudo-noise sequence (1). The resultant signal (2) is added to noise and transmitted.

¹The notation of $xxx\yy$ is used to distinguish nodes specified as subcircuits in the Tesla simulation.

Figure 8.22 shows the partial receiver model. The received signal (assuming it has passed through a receiver front-end) is squared to remove the spread spectrum signal. The result is that a signal at twice the carrier frequency is obtained ($lock\backslash3$). The signal is hard-limited, giving a TTL compatible signal ($of\backslash7$). The clock signal is lowpass filtered to remove higher order terms and shifted by ninety degrees. The resultant signal (5) is used to demodulate the received signal (3). The signal is fed to the baseband spread spectrum synchronisation network as given in figure 8.1.

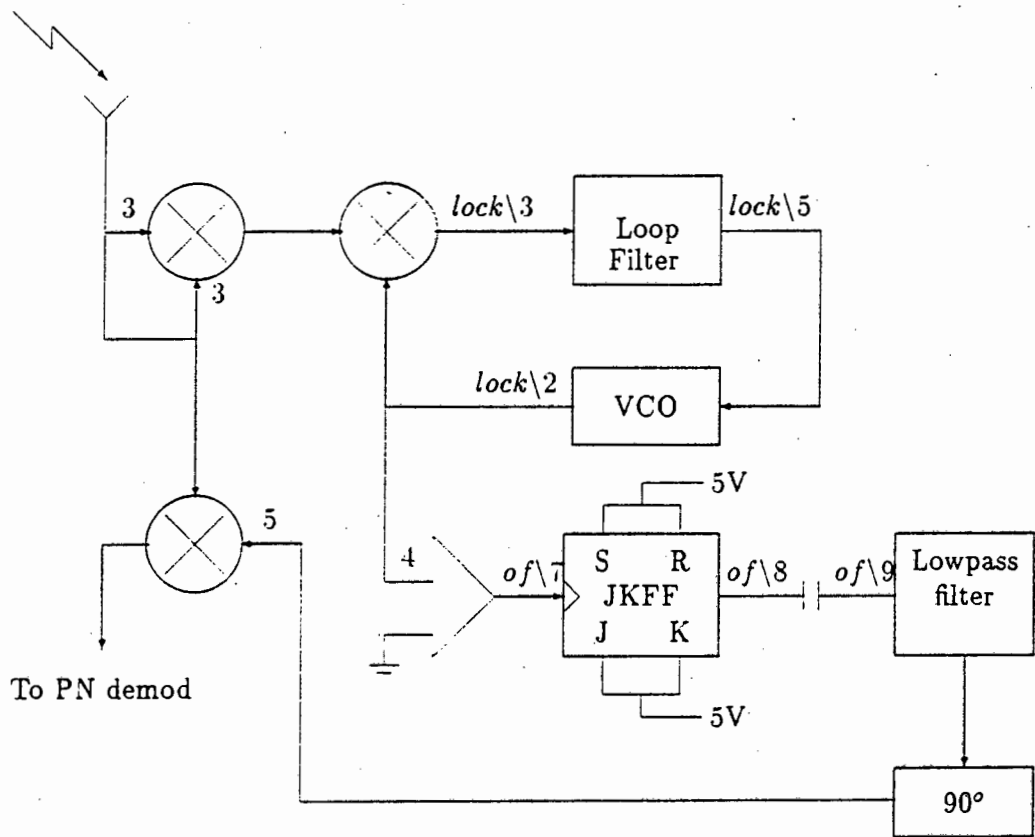


Figure 8.22: Partial Receiver

8.2.3 Simulation Parameters

8.2.3.1 Spread Spectrum Simulation Parameters

Parameter	Value
Sequence Length	15
Code Offset	7 chips
Tx Clock Frequency	101 kHz
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	101 kHz
Receiver LPF Type	4th Order Butterworth

Table 8.5: Simulation Parameters: Spread Spectrum Network

8.2.3.2 Simulation Parameters: Carrier Synchronisation Network

Parameter	Value
Carrier Frequency	1.08 MHz
Average Power	3.125 Watts
VCO Centre Frequency	1 MHz
VCO Deviation	400 kHz
Loop Filter Bandwidth	6 kHz
Loop Filter Type	1st Order Active Lowpass

Table 8.6: Simulation Parameters - Carrier Synchronisation Network

8.2.4 Noiseless Case

Figure 8.23 shows the frequency spectrum of the transmitted waveform. The characteristic spread spectrum frequency spectrum is prevalent. The carrier is -14 dB down from the largest frequency component. In practical systems only the main lobe is transmitted and

the side-lobes are removed. The primary reason is that more than 90 percent of the total power is carried in the main-lobe.

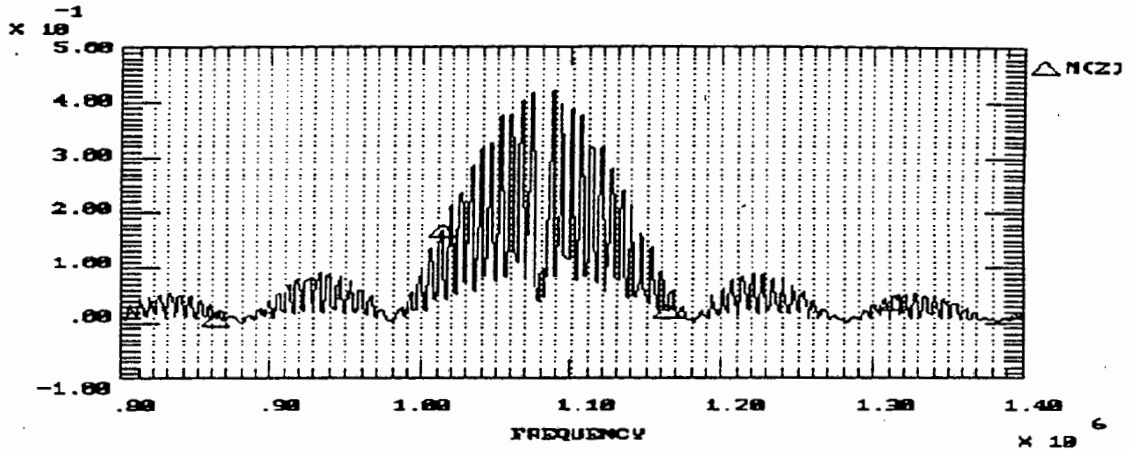


Figure 8.23: Spectrum of Transmitted Signal. The characteristic spectrum of a PN sequence is clearly illustrated. In practical designs only the primary lobe is transmitted, with the side-lobes filtered out.

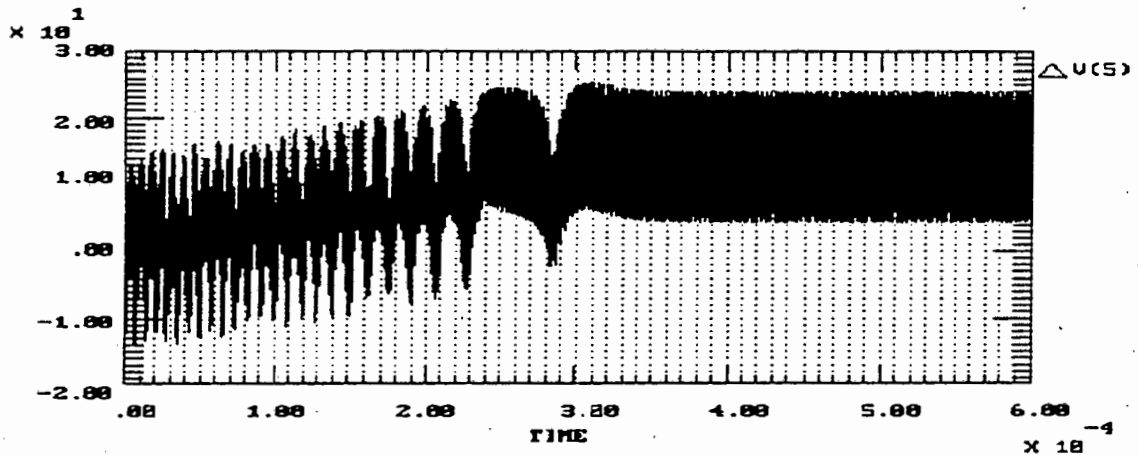


Figure 8.24: Phase-locked Loop Error Signal (*lock\5*). Carrier lock is achieved in 0.3 milliseconds. This is indicated by the error signal settling around a DC value.

Figure 8.24 shows the VCO drive signal. Carrier lock was achieved in 0.3 milliseconds and was maintained through the duration of the simulation. Figure 8.25 shows the spectrum of the transmitted carrier and the locally generated reference signal. The frequencies are both 1.08 MHz.

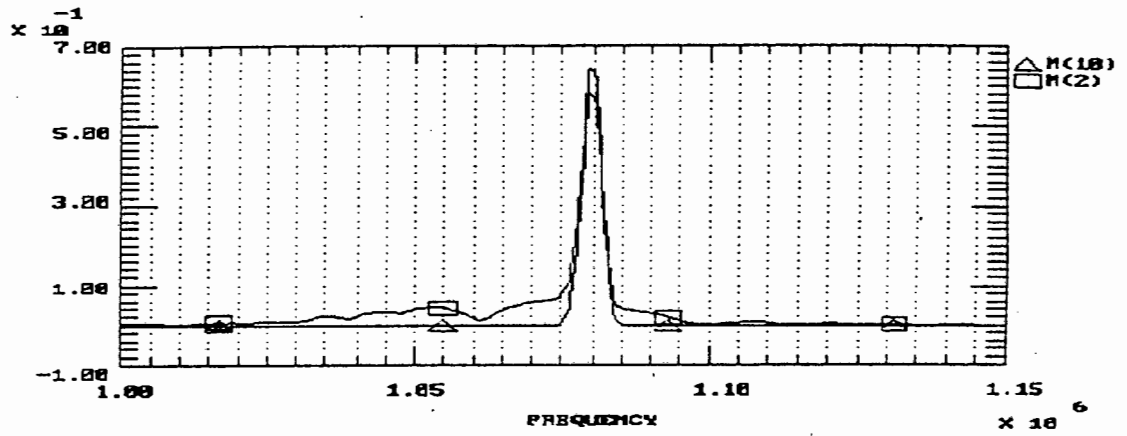


Figure 8.25: Spectrum of Carrier and Locally Generated Reference

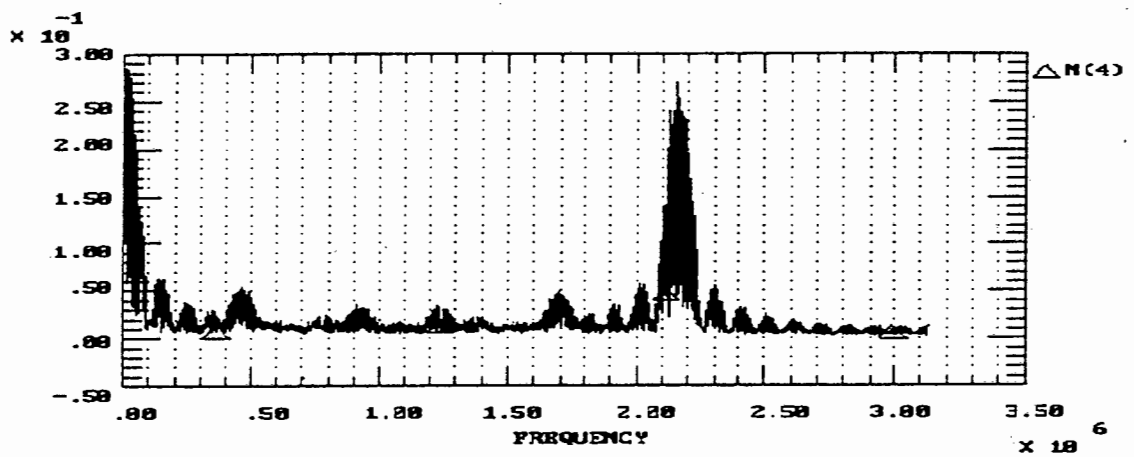


Figure 8.26: Spectrum of Demodulated Signal.

Figure 8.26 shows the demodulated signal. The PN spectrum is shifted to base-band and 2.16 MHz. The signal is filtered using a 4th order Butterworth lowpass filter. The 15 frequency components, are spaced 7.2 kHz apart.

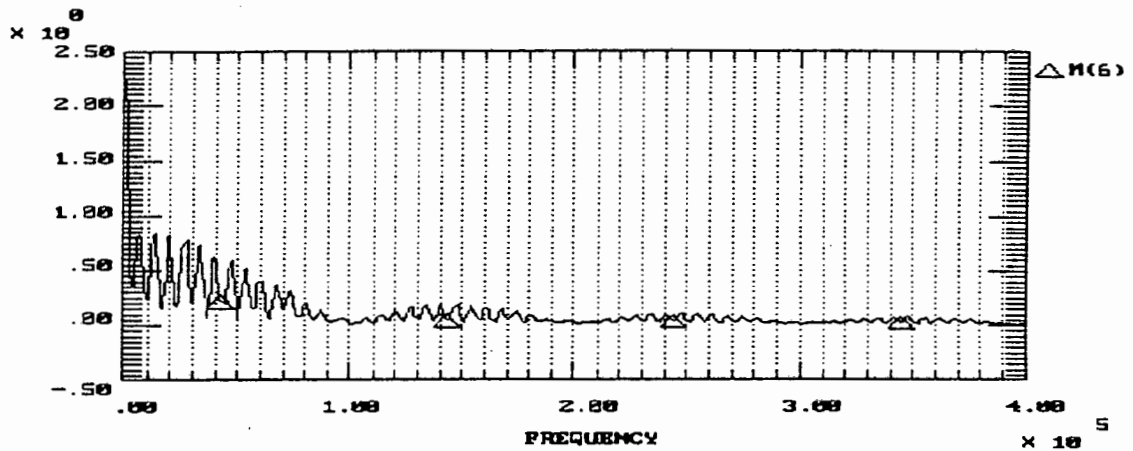


Figure 8.27: Spectrum of Filtered Demodulated Signal

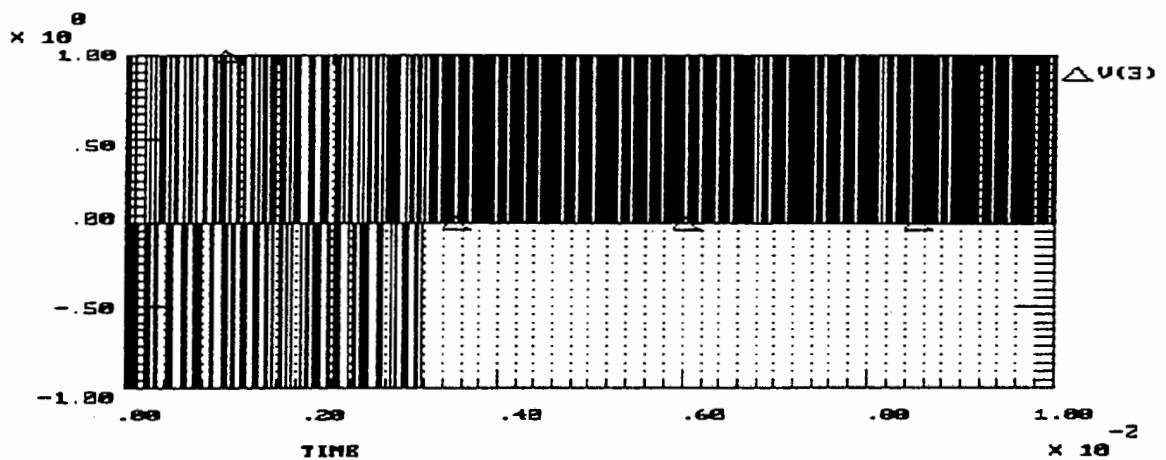


Figure 8.28: Input to Bandpass Filter. Lock is determined by a train of all positive or all negative. Lock was achieved in 3 milliseconds.

Figure 8.28 shows the input to the bandpass filter. In the design of the novel spread spectrum system it was stated that once the impulses are all positive or all negative the system is in lock. From figure 8.28 the system achieves lock in 3 milliseconds. This is still within the same order as the RASE and other serial search techniques discussed in chapter 9. Figure 8.29 shows the transmitted and locally generated PN sequences. The sequences locked with a one chip offset.

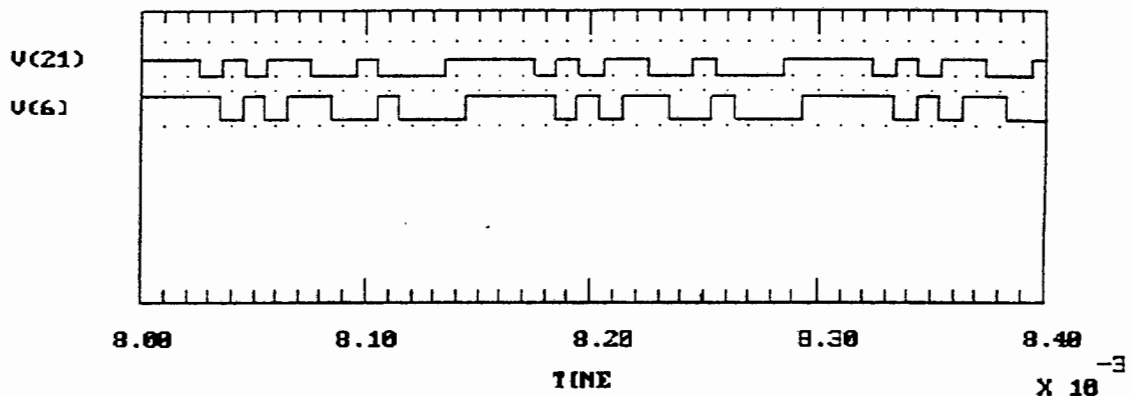


Figure 8.29: Comparison between Transmitted and Receiver PN Sequences. Since positive lock occurred, the sequences are offset by one chip.

8.2.5 Effect of noise on Carrier Synchronisation

8.2.6 Synchronisation at an Average Signal-to-Noise ratio of -9 dB

Using the same parameters as the previous case without noise, the system was evaluated under a noise power of 25 Watts. Therefore the average transmitted signal-to-noise ratio is -9 dB. The system was simulated for 100 milliseconds. Figure 8.30 shows the transmitted spectrum. The level of noise is clearly seen, with the side lobes embedded within the noise. (The signal-to-noise ratio in the noise bandwidth is greater than 0 dB.)

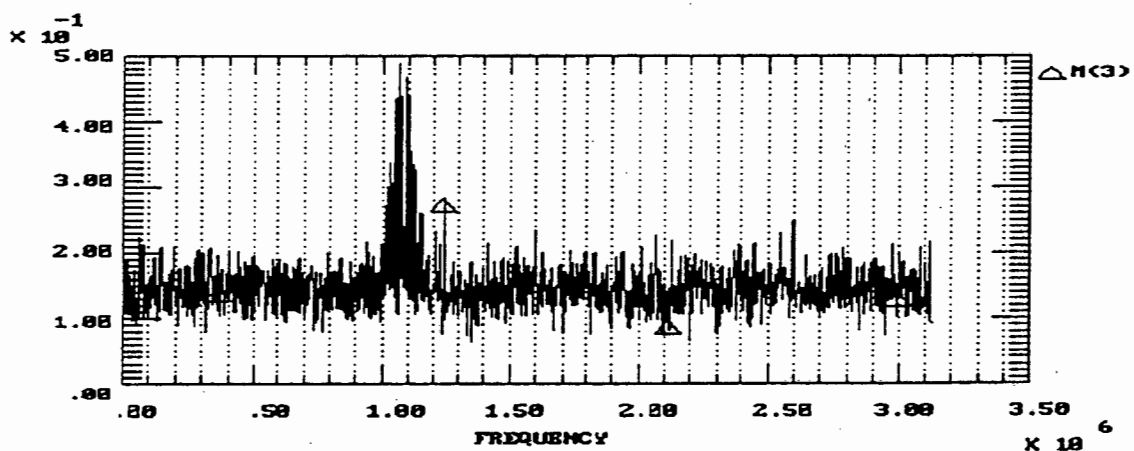


Figure 8.30: Spectrum of Transmitted Signal. The characteristic PN sequence spectrum is distorted by noise.

Using the VCO drive signal to determine if lock was achieved was not possible due to the excessive noise. To determine whether the system was in lock, the locally generated reference was compared to the transmitted carrier waveform. The carrier was synchronised in around 8.345 milliseconds. Figure 8.31 shows the transmitter carrier and the locally generated reference signal from just before phase acquisition.

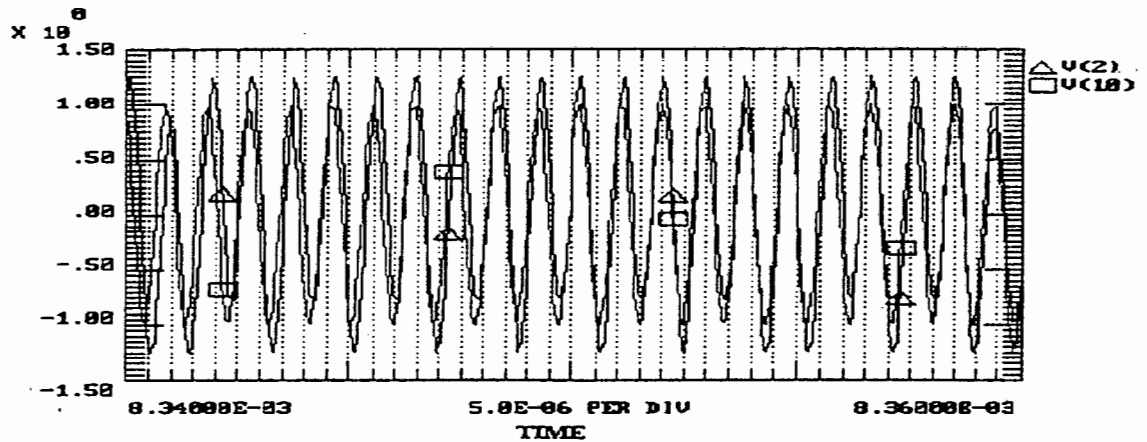


Figure 8.31: Tx Carrier versus Locally Generated Reference

Figure 8.32 shows the demodulated frequency spectrum. The spectrum of the PN sequence is distorted by the noise. The spread spectrum did not synchronise within the simulation time window. The synchronisation of the spread spectrum under noise is dealt with in section 8.1.

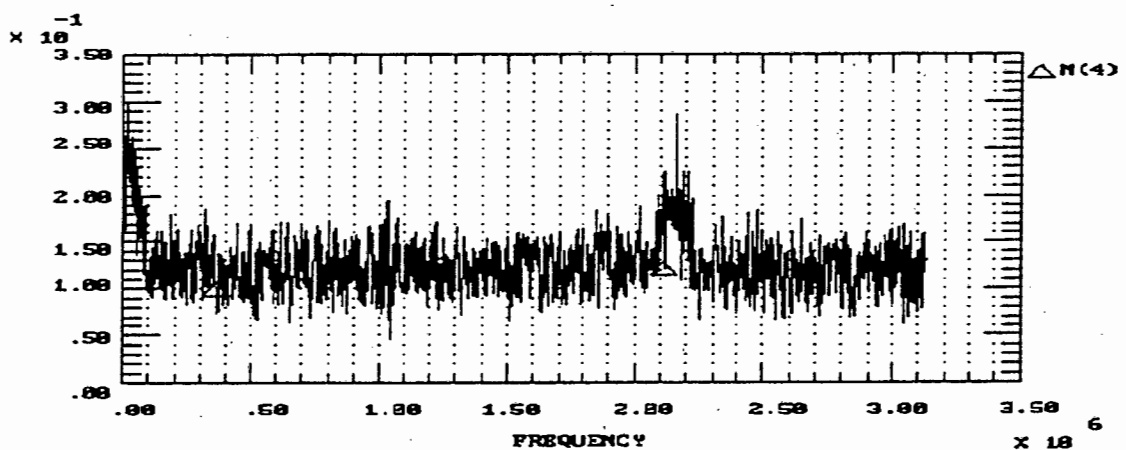


Figure 8.32: Spectrum of Demodulated Signal.

8.2.7 Synchronisation at an Average Signal-to-Noise ratio of -15 dB

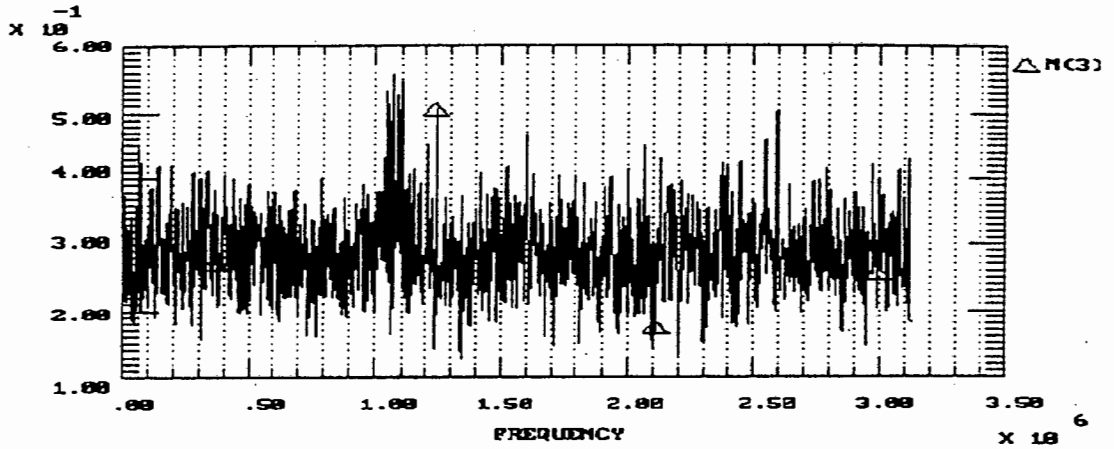


Figure 8.33: Spectrum of Transmitted Signal. The transmitted signal is embedded in noise.

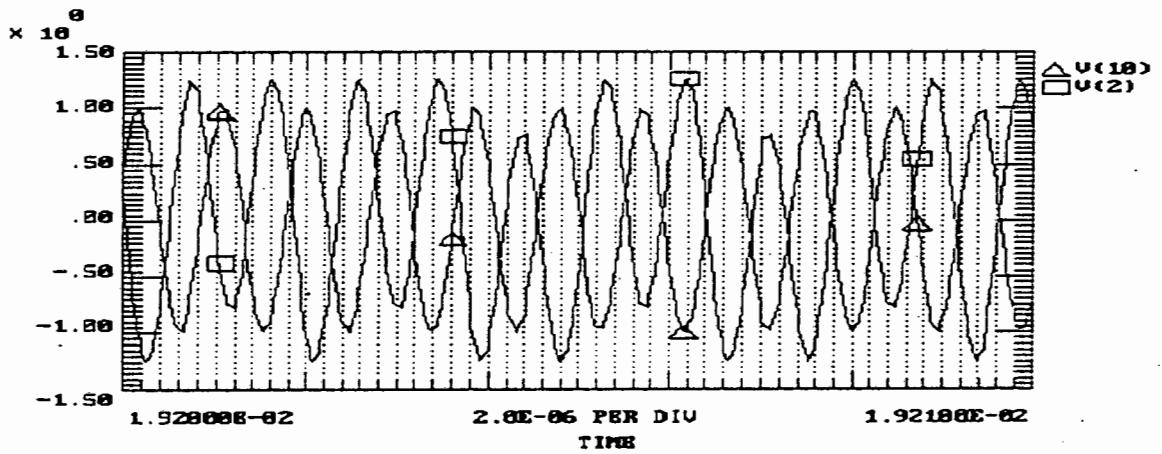


Figure 8.34: Transmitter Carrier and Locally generated reference. The signals locked 180 degrees out of phase.

8.2.8 Discussion

The noise power was increased further, but the system failed to synchronise within the given synchronisation time window. From the results presented it is clear that the carrier synchronisation can be a limiting factor on the overall synchronisation network. (In most spread spectrum implementations carrier tracking is either not required (non-coherent techniques) or is done after despreading [34]. By doing carrier tracking after despreading, the processing gain can be used to improve the recovery of the carrier.) The trade-off is the required synchronisation time of the system versus the signal-to-noise ratio at which

the system is to function. It should be stated, however, that the choice is dependent on the application. If the system is to be used for low-cost rural communication [49][50] the number of factors that enter the equation increases, especially when using code division multiple access, where long sequences and low signal-to-noise ratios are required. In the design, however, there are no overall set of rules that govern these choices and it is therefore up to the designer to ensure that all the factors are taken into account.

8.3 The Influence of Data on Spread Spectrum Synchronisation

The spreading of data in spread spectrum systems has a dramatic effect on the synchronisation of the spreading waveform. In chapter 2 it was shown that the optimal tracking network and baseband full-time early-late tracking loop failed under the presence of data. The novel implementation suffers from the same drawbacks when data is included.

The loop, while providing for simple pseudo-noise sequence synchronisation, cannot lock on a data modulated signal. A sign change in the input will result in the spikes reversing polarity, thus falsely causing the loop to adjust its code-phase. However, if the state of the receiver is monitored the loop can be adapted to a data driven environment. The loop can be designed to detect for code-phase synchronisation and then to rectify the polarity of the spikes when a sign change occurs due to the data. This will result in the loss of code-phase information as all the spikes will be unipolar, but as the sequences are in lock this information is redundant. Two sequences in code-phase lock will remain in lock if driven at the same frequency. As the spikes still contain frequency information the receiver will track the transmitter accurately.

In any data systems it is not unusual to allow a fixed period of time for the system to synchronise before modulating the wave-form. For the novel synchronisation network, the following design criteria were specified:

- The transmitted data is preceded by a series of 1's to allow for the acquisition of the spreading waveform.
- A reasonable Signal-to-Noise ratio is required.

8.3.1 Spike Counting to Determine Data

Using the design criteria above it is possible to overcome the inversion of the impulses by the data after lock. Once the system is in lock the output becomes a series of all positive or all negative impulses. Lock can be detected by counting at least half of the impulses. Figure 8.35 shows the receiver and figure 8.36 the detect circuitry.

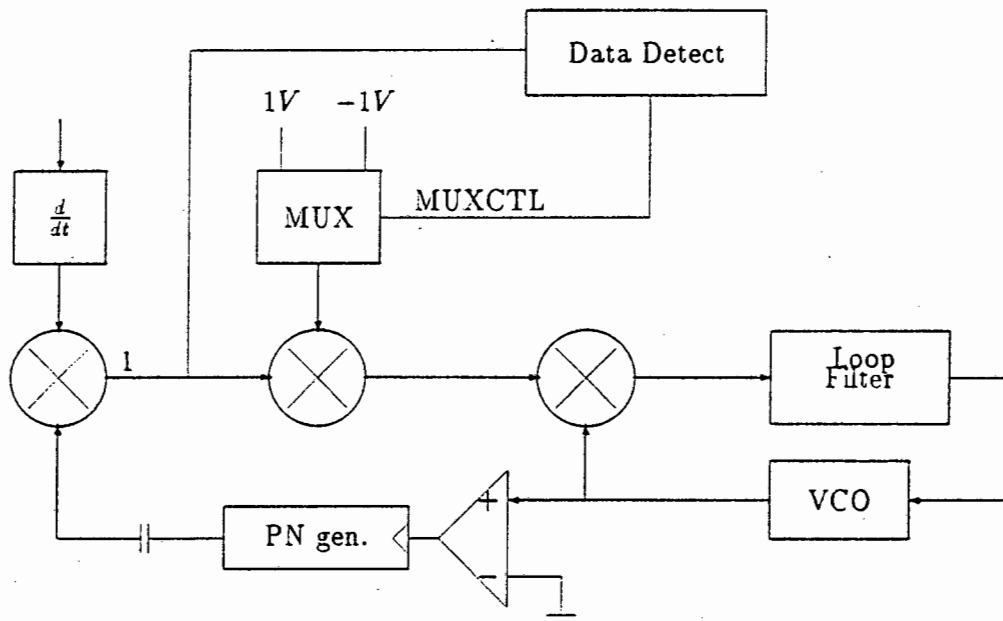


Figure 8.35: Receiver with data detection

Circuitry was thus designed to detect firstly code lock and secondly for the polarity of the locking spikes. (See Figure 8.36). The detection circuit consists of two arms, one to deal with the negative going spike and the other with the positive going spike. Each arm consists of a comparator, to determine the sign of the spike, a binary counter (CNTR), to detect for a lock condition, and a memory element (MEM), to save the polarity of the locking condition.

The input spikes are compared to set positive (POS) and negative (NEG) levels to determine locking direction. The narrow pulses generated by the comparators trigger monostables (M-S) with pulse widths set to half that of a chipping period so as to allow the digital circuitry time to settle and time to change the state of the tracking loop. These pulses are then counted using binary counters (CNTR). In simulation the loop was said to be in lock when the most significant bit (M) of the counter was high. The counters are

inter-locked as a negative going pulse will reset (RST) the positive counter and visa-versa as a spike of opposite polarity is an indication that lock still needs to be achieved. The counters are gated with their own outputs so that they do not continue counting once lock has been detected and wrap around back to zero.

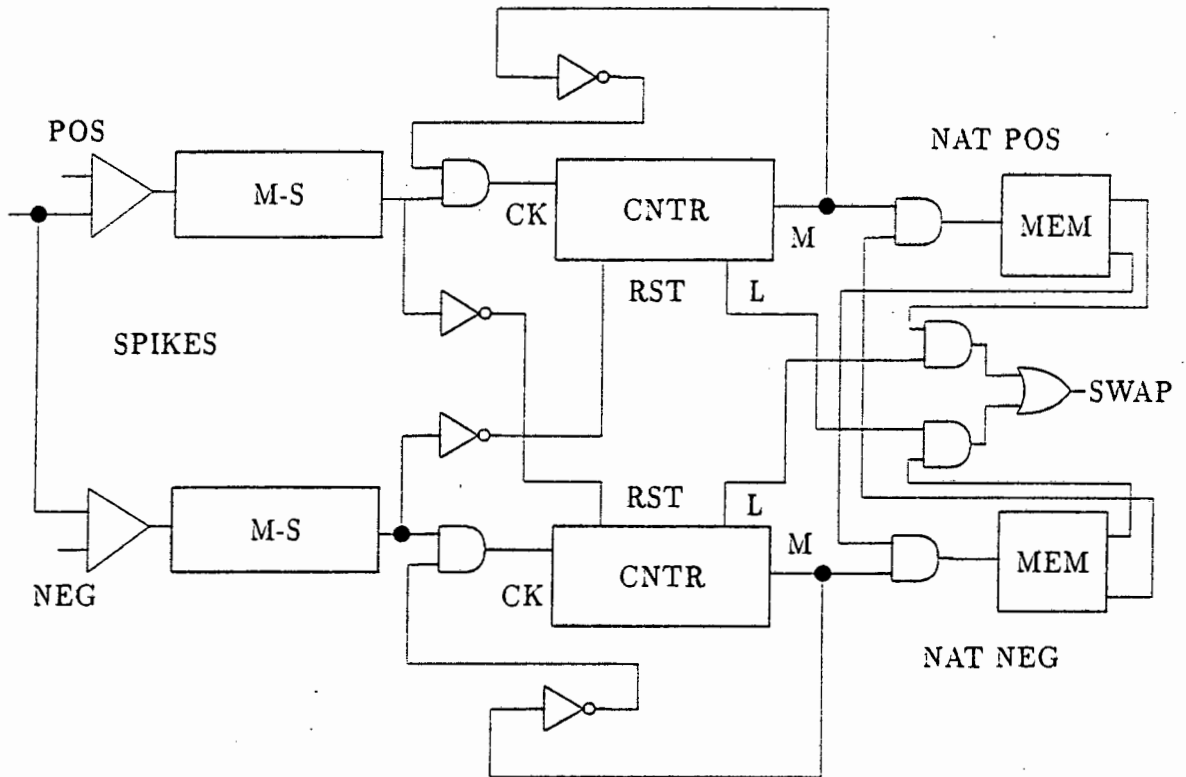


Figure 8.36: Circuit diagram of the Detection Circuitry.

Once the tracking loop has locked, as evidenced by either the negative or positive counter's most significant bit going high, then that polarity is declared the natural locking direction. This information is then stored in a memory element (NAT POS or NAT NEG). One then AND gates the natural locking direction with the least significant bit (L) of the opposite direction counter to detect when the loop should generate the SWAP signal. (See Figure 8.37). The SWAP signal is the control flag deciding whether one must rectify the spikes or not. (See Figure 8.35). This keeps the input spikes to the clock circuitry unipolar and the loop remains in lock.

	NEGS.POSS	$\overline{NEGS.POSS}$	$\overline{\overline{NEGS.POSS}}$	$NEGS.\overline{POSS}$
$NATN.NATP$	X	X	X	X
$\overline{NATN.NATP}$	X	1	X	0
$\overline{NATN.\overline{NATP}}$	X	X	X	X
$NATN.\overline{NATP}$	X	0	X	1

$$SWAP = (NATP \text{ AND } NEGS) \text{ OR } (NATN \text{ AND } POSS)$$

NEGS = NEGATIVE SPIKE
 NATN = NATURAL NEGATIVE

POSS = POSITIVE SPIKE
 NATP = NATURAL POSITIVE

Figure 8.37: Logic table for the detection circuitry.

The data was recovered by multiplying the input with the receiver's replica of the pseudo-noise sequence followed by a correlation receiver. However, this assumes that the data clock recovered. For the purpose of simulation the transmitter clock was carried across to the receiver. The integrate and dump circuit is triggered at the edge of a bit period. On closer examination of the detect circuitry, the MUX output signal is a representation of the data. It is therefore possible to use this as the data signal and avoid the use of integrate and dump circuitry. Although this is not optimal it reduces the complexity of the receiver and removes an extra layer of synchronisation from the receiver.

8.3.1.1 Simulation Results

Parameter	Value
Sequence Length	15
Code Offset	4 chips
Tx Clock Frequency	101 kHz
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	101 kHz
Receiver LPF Type	1st Order RC

Table 8.7: Simulation Parameters: Spread Spectrum Network

Using the simulation parameters given in figure 8.7 the following simulation results were obtained. Figure 8.38 shows the original data versus the output data. From the graph it is clear that the original data, although inverted, was successfully determined.

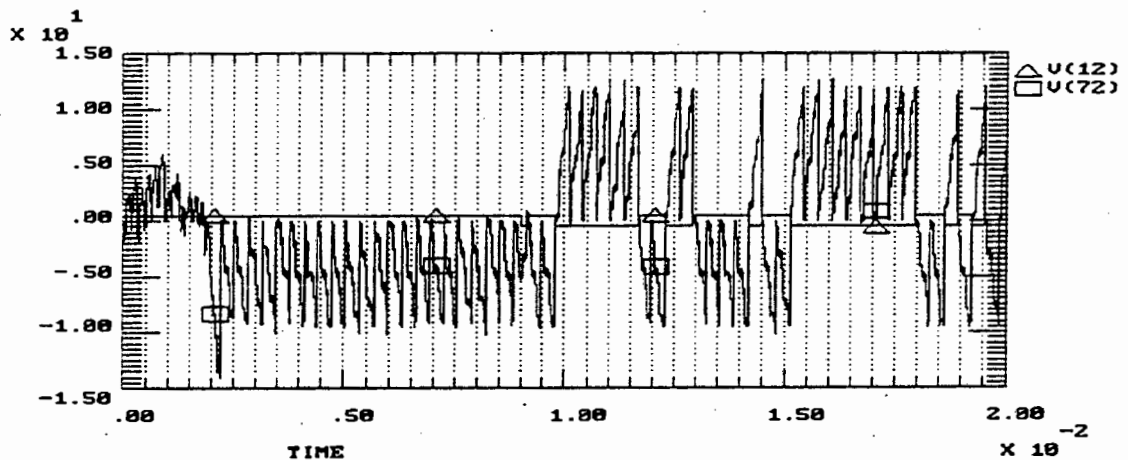


Figure 8.38: Detected Data versus Transmitted Data. The system achieved lock in 20 milliseconds.

Figure 8.39 shows the spikes before polarity correction. The envelope of the signal is a clear representation of the transmitted data. Figure 8.40 shows the spikes after correction. The single impulses are due to the fact that the change in sign must be detected first. These impulses with incorrect polarity will have little effect on the synchronism of the system.

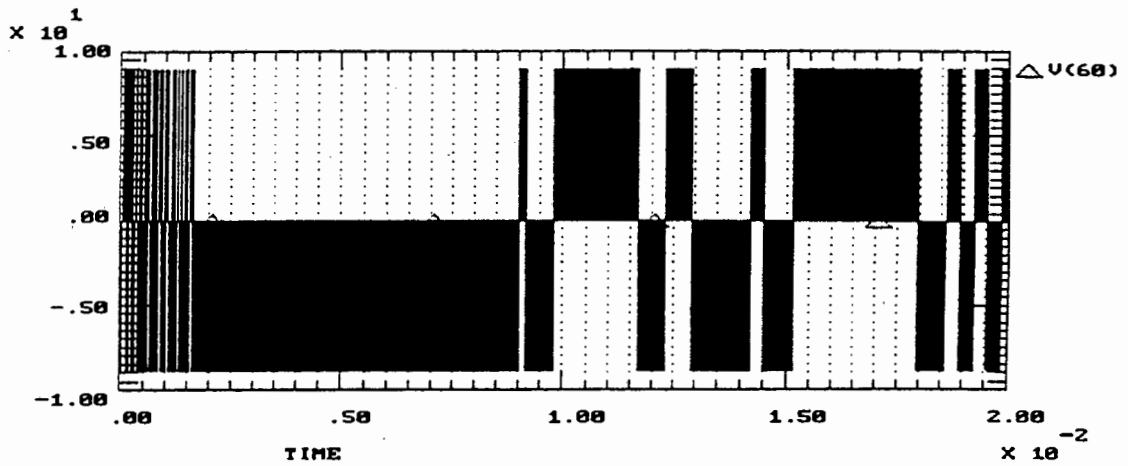


Figure 8.39: Spikes Before Polarity Correction. If the detect circuitry was not present, the synchronisation network would go through 180 degree phase-shifts which will affect the synchronisation network.

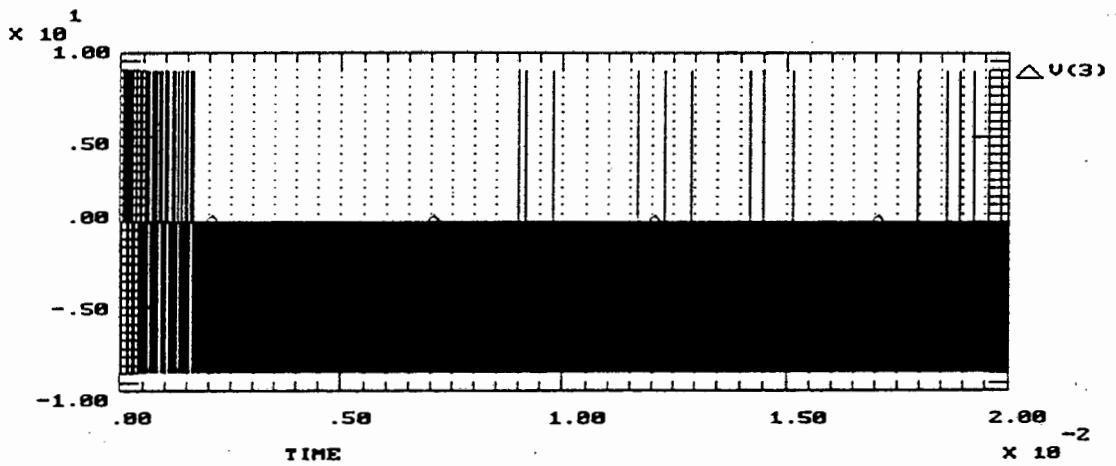


Figure 8.40: Spikes After Polarity Correction. Only the switching spikes are now present. When an impulse occurs, the sign of the impulses is inverted.

8.3.1.2 Discussion

It is clear from the results presented that for the system to work successfully the signal-to-noise ratio of the system should be such that the number of incorrect impulses that enter the system are kept to a minimum. In the detection circuitry the change of state occurred based on one impulse. With the design however it is possible to count a few spikes before changing the polarity of the impulses entering the system. In this way the system can operate at a slightly lower signal-to-noise ratio. In the next section, the issue of data detection in noise is dealt with. The primary difference between the detect circuitry is that

a small time frame of the PN sequence is correlated, and based on the value the polarity is changed.

8.4 Data Determination Under Noise

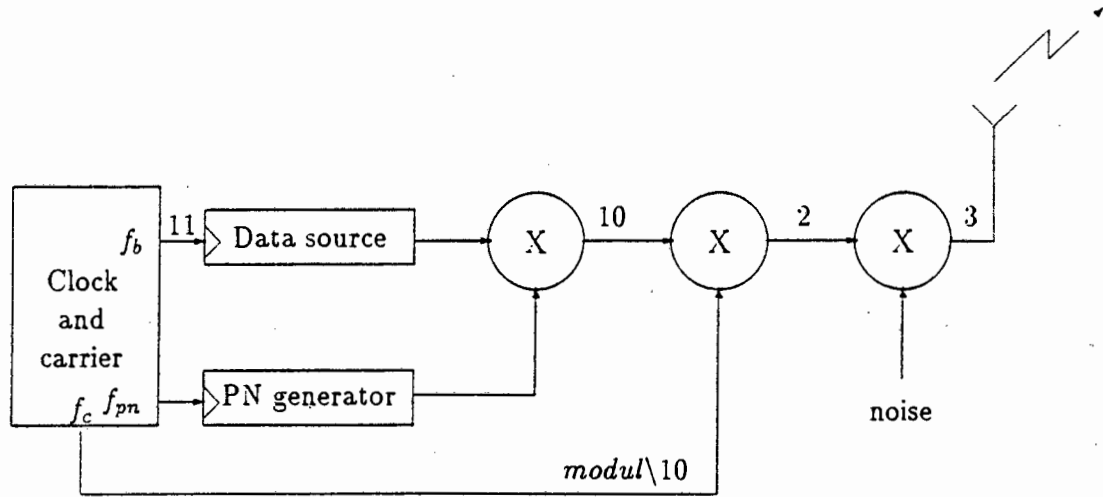


Figure 8.41: Transmitter

Figure 8.41 shows the basic design of a direct sequence spread spectrum transmitter. The data source includes the necessary logic to insert a number of ones before the actual data to allow for the receiver to achieve complete synchronisation. The data and the spreading sequence are multiplied together to form the baseband spread spectrum signal (10). The resultant signal is modulated to f_c (2).

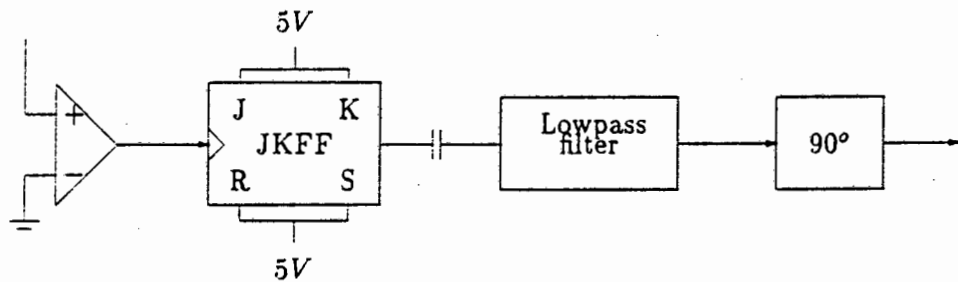


Figure 8.42: Frequency Divider

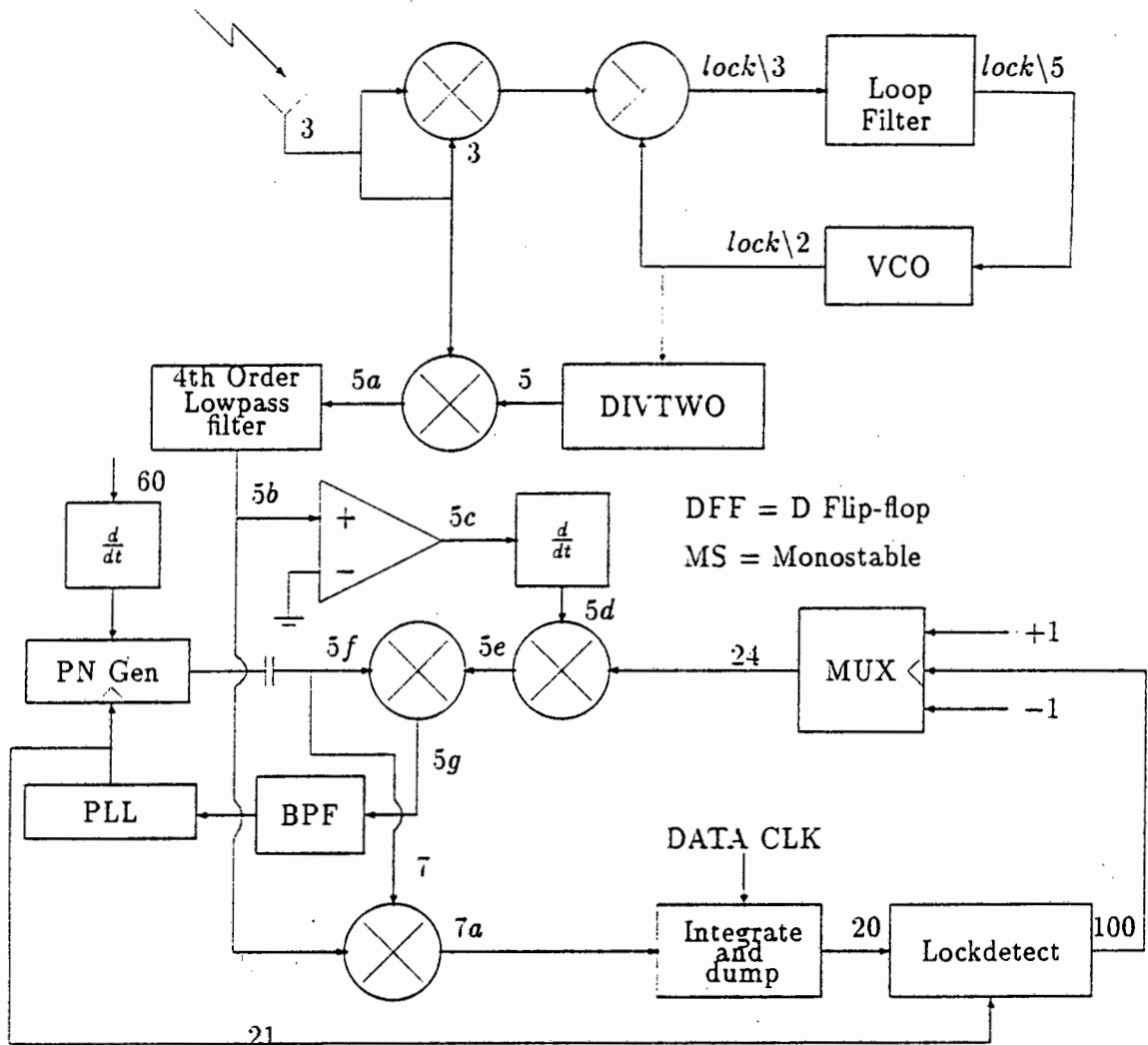


Figure 8.43: Partial Receiver

At the receiver, figure 8.43 the received signal is squared to remove the spreading and data signals, and is fed into a phase-locked loop running at twice the carrier frequency. The output of the VCO is at twice the original frequency. To obtain the original frequency the VCO output signal is fed into a frequency divider as given in figure 8.42. The sinusoid is hardlimited and fed into a JK flip-flop. Since the phase-locked loop and the original signal are out of phase, the signal is shifted by 90 degrees. The resultant signal is used to demodulate the received signal (3). The demodulated signal (5a) is lowpass filtered to remove the information signal at twice the carrier frequency and reduce the noise into the

base-band demodulation network. The signal (5b) is hard-limited to generate square pulses (5c) which are differentiated (or signed edge detected). The resultant impulses (5d) are multiplied with the locally generated PN sequence and a signal to remove the inversions caused by the data. The impulses (5g) stimulate a bandpass filter which in turn drives a phase-locked loop (see 8.1 for implementation).

The filtered baseband signal (5b) is correlated with the locally generated PN sequence to extract the original spread spectrum signal. In BPSK the clock is recovered from the data signal. Due to the limitations in the number of nodes Tesla can cope with, the transmitter carrier was used to obtain the transmitted data. Therefore the total synchronisation times given still require the data clock recovery time to be included. The output of the correlation receiver (20) is fed into the lock detect circuitry.

The inputs to the lock detect circuitry include the data (20) and the PN sequence clock (21). The circuit has two sections. The upper branch determines the direction of the present data, while the lower section determines whether the system is in positive or negative lock.

The output (100) drives the MUX that determines the direction of the impulses. The data signal (20) is fed to $C1$, $C2$ and the Integrate and Dump Circuit. The output of $C1$ and $C2$ is dependent on the input from the data correlation receiver. The comparison value of ± 8 volts is dependent on the level of noise and voltages of the PN sequences, received signal and so forth. Only one of the comparator branches will be positive. Assuming that the output of $C1$ goes from high to low, then a pulse is generated by the monostable. The state is permanently gated in the D-type flip-flop.

Table 8.8 shows the logic table for the input circuitry to the D flip-flops $DFF1$ and $DFF2$. Since A and B cannot be in the same state, only the relevant combinations are shown. Only when C goes high will D or E change state. The change of state is dependent on the D flip-flops. On start-up the flip-flops are set to the zero state, i.e. $DFFQ = 0$ and therefore $DFF\bar{Q} = 1$. Table 8.9 gives the relevant states for $DFF1$ and $DFF2$. From the table it is clear that once the state of the flip-flop is set, it will remain in that state irrespective of the input to the flip-flop. From the tables it is clear that $DFF1Q$ and $DFF2Q$ cannot be high at the same time.

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
0	1	0	0	0
1	0	0	0	0
0	1	1	0	1
1	0	1	1	0

Table 8.8: Logic Table for System Lock (1)

<i>D/E</i>	<i>DDFFQ_{past}</i>	<i>DDFFQ̄_{past}</i>	CLK	<i>DDFFQ_{pres}</i>	<i>DDFFQ̄_{pres}</i>
0	0	1	1	0	1
1	0	1	0	1	0
0	1	0	1	1	0
1	1	0	0	1	0
0	1	0	1	1	0

Table 8.9: Logic Table for System Lock (2)

The locally generated PN sequence clock is fed into monostable (*MS3*) and flip-flops *DDFF2* and *DDFF3*. When the clock goes low, a positive pulse is generated. The pulse is inverted and is used to reset the integrate and dump circuit (when drive signal is low). The data signal is therefore only correlated over one chipping period to determine a change in sign. The output of the integrate and dump circuit is fed in *C3* and *C4* and compared to ± 1 volts respectively. When the clock goes low the states of the comparators are stored. The states of the flip-flops determine whether the input spikes are inverted or not. Only one of the flip flops can be high at any one time. Therefore from the results in tables 8.8 and 8.9, the logic table 8.10 for the swap signal can be determined. Note that *C* in figure 8.44 is used to switch in a delay if the PN sequences are offset by one chip (positive impulse lock - chapter 5).

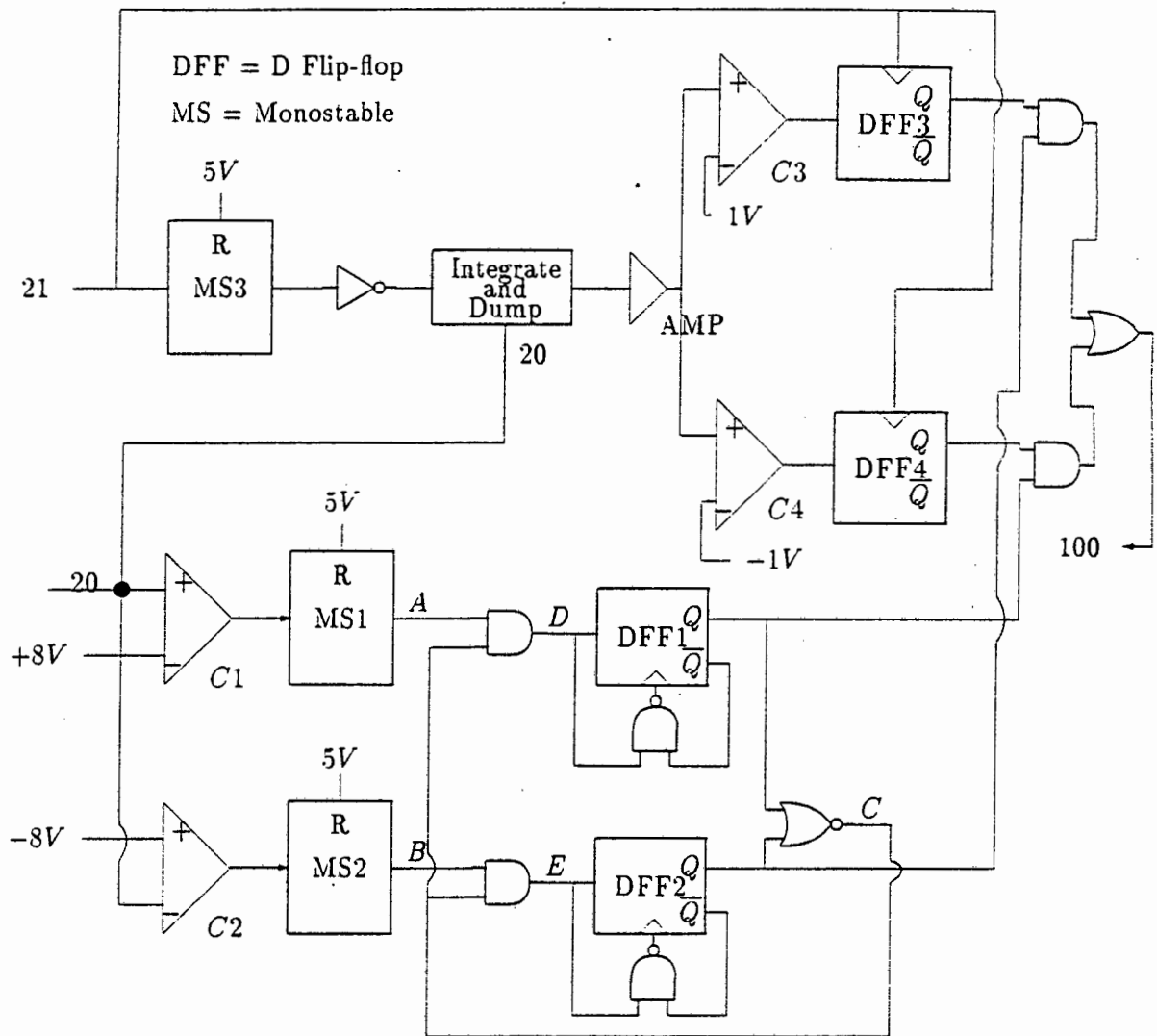


Figure 8.44: Lock Detect Circuitry

<i>DFF3Q</i>	<i>DFF4Q</i>	<i>DFF1Q</i>	<i>DFF2Q</i>	<i>SWAP</i>	ACTION
Pos	Neg	PosLoc	NegLoc		
0	1	0	1	0	
1	0	0	1	1	invert
0	1	1	0	1	invert
1	0	1	0	0	

Table 8.10: Logic Table for System Lock (1)

8.4.1 Noiseless Case

Parameter	Value
Sequence Length	15
Code Offset	7 chips
Tx Clock Frequency	101 kHz
BPF Bandwidth	7 kHz
VCO Centre Frequency	100 kHz
VCO Deviation	6 kHz
Loop Filter Bandwidth	1 kHz
Receiver LPF Filter Bandwidth	101 kHz
Receiver LPF Type	4th Order Butterworth

Table 8.11: Simulation Parameters: Spread Spectrum Network

The system was simulated without noise to show how the circuitry operates. Figure 8.45 shows the transmitted data and the output of the correlator. The system achieved lock in 4.7 milliseconds. Figure 8.45 shows the MUX control signal versus the transmitted data. When the data changes sign, the detect circuitry determined that a sign change had taken place and the input spikes were inverted as shown in figure 8.46. The opposite occurs when the data goes from low to high. In figure 8.47 the switching is clearly identified by the short periods of positive impulses. Since the number of impulses in the opposite direction are only a small proportion of the total number of negative going spikes they will not affect the synchronisation of the circuit.

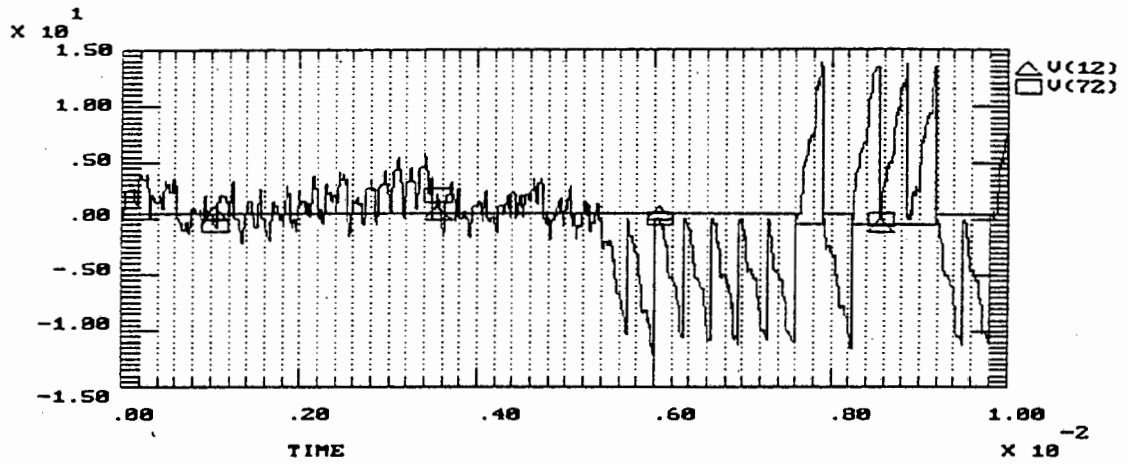


Figure 8.45: Detected Data versus Transmitted Data. Lock was achieved in 54 milliseconds.

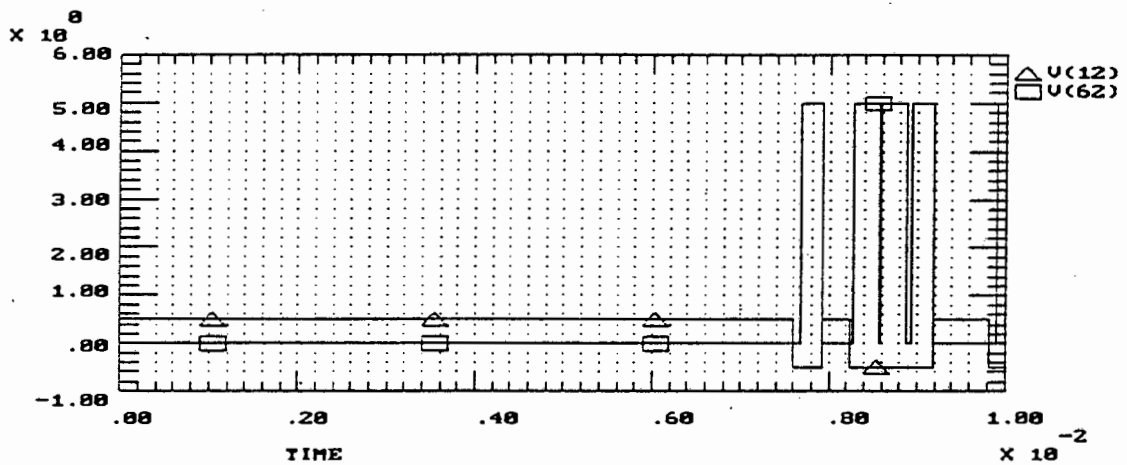


Figure 8.46: Control Signal (V(12)) versus data signal (V(62)).

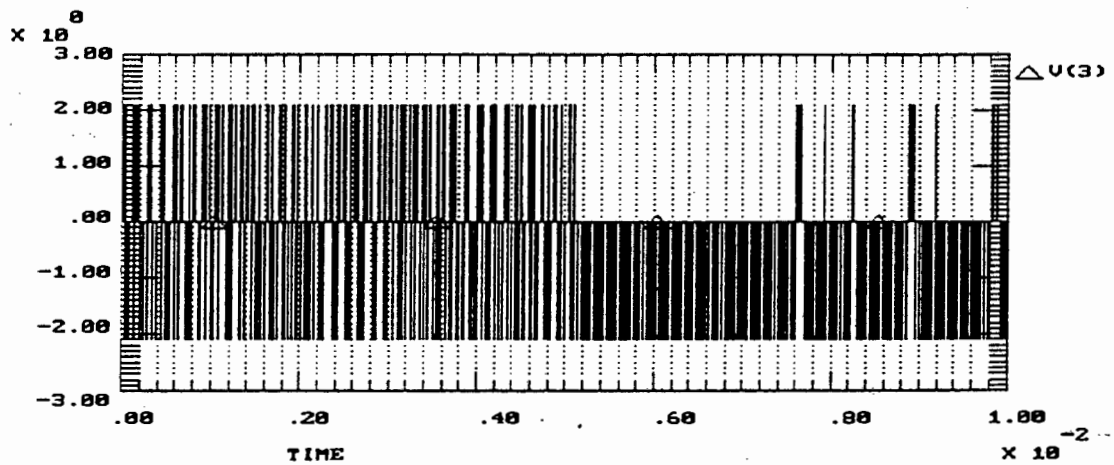


Figure 8.47: Rectified Spikes. The positive going spikes after lock are caused by the delay in determining whether the data has changed sign.

8.4.2 Signal-to-Noise Ratio of 0.3 dB

Noise was added to the system giving a signal-to-noise ratio of 0.3 dB. The system achieved synchronisation within 6 milliseconds, slightly slower than the ideal case. The data was decoded correctly as shown in figure 8.48. The data inversion can be caused by any number of things. Since the initial data is predefined, the inverted data can be corrected with little effort.

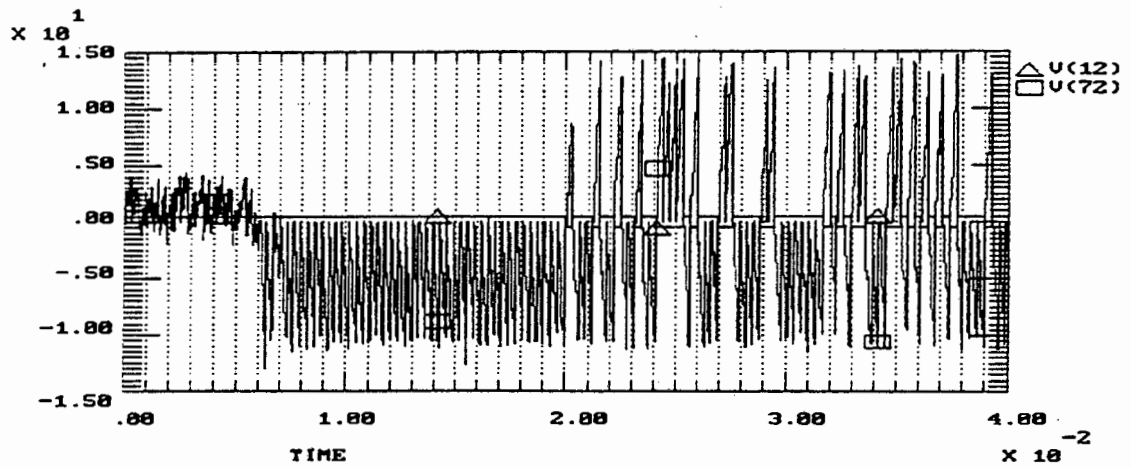


Figure 8.48: Detected Data versus Transmitted Data. Lock is achieved within 6 milliseconds.

8.4.3 Signal-to-Noise Ratio of -13.6 dB

The signal-to-noise ratio was then decreased to -13.6 dB. The resultant data is shown in figure 8.49. The system achieved lock in just over 8 milliseconds. The received data was decoded correctly.

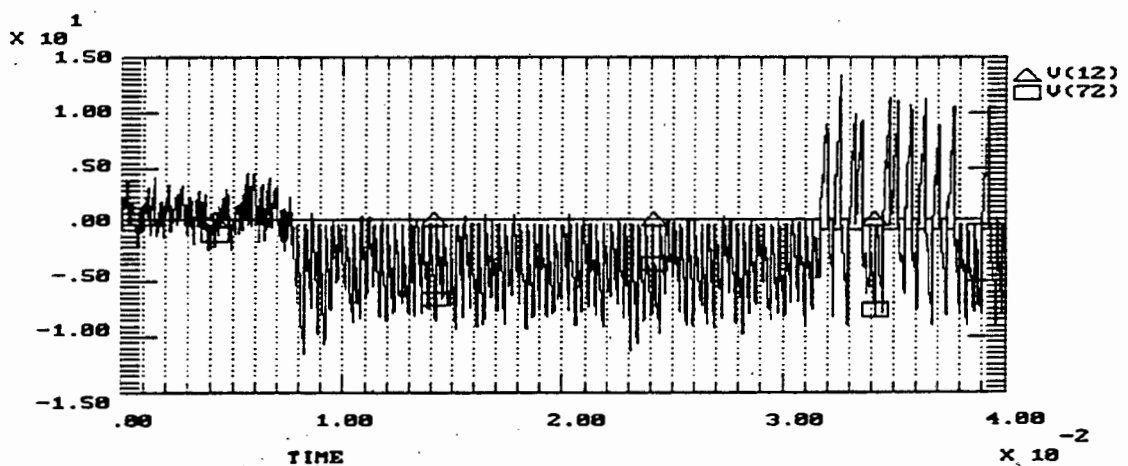


Figure 8.49: Detected Data versus Transmitted Data. Lock is achieved within 8 milliseconds.

8.4.4 Signal-to-Noise Ratio of -16.1 dB

The signal-to-noise ratio was decreased by a further 2 dB. The system achieved lock in 10 milliseconds. A number of data errors occurred around 16 milliseconds. Data was decoded correctly between 21 and 28 milliseconds before a number of errors were made. Data was decoded correctly around 32 milliseconds. Figure 8.51 show that the system did not lose lock when the data was incorrectly decoded. This implies that the system behaves well even when mistakes are made in the decoding of data.

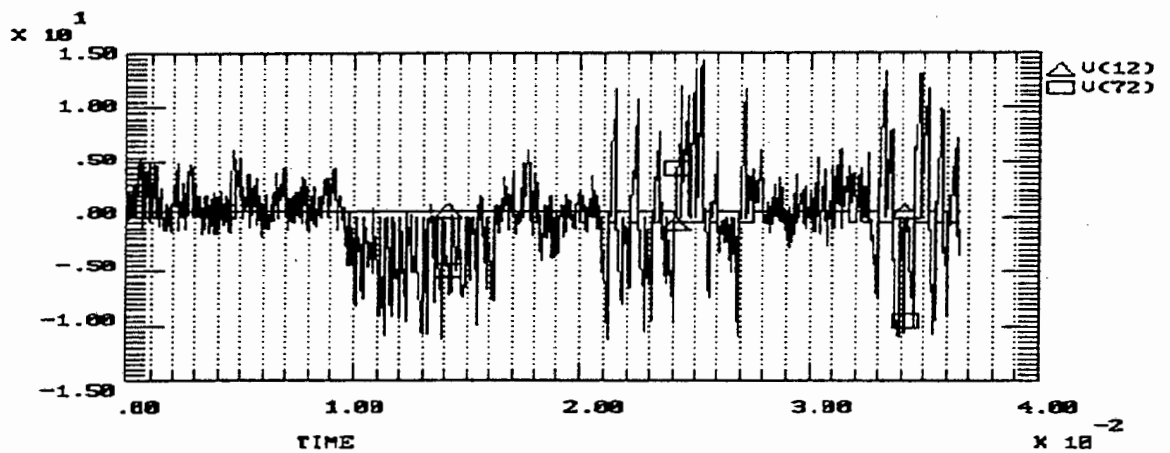


Figure 8.50: Detected Data verses Transmitted Data. Lock is achieved within 10 milliseconds.

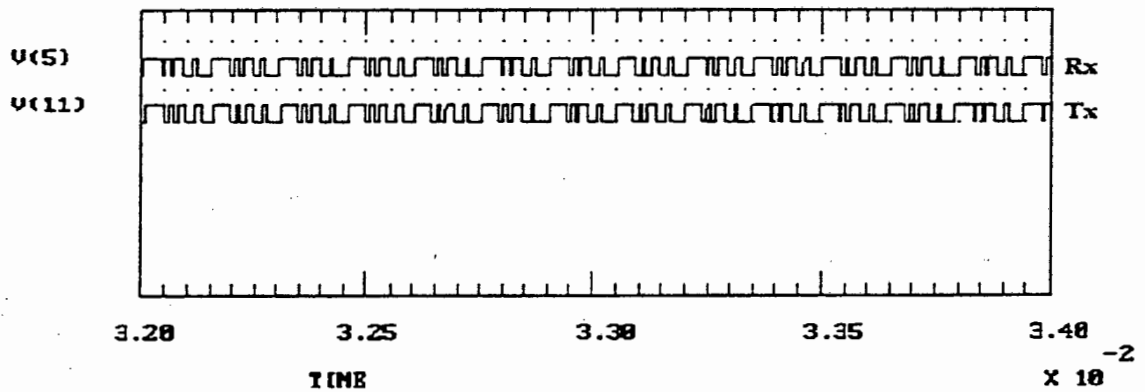


Figure 8.51: Transmitted and Locally Generated PN Sequences. Even though no data was decoded correctly between 2.9 and 3.2 milliseconds, the PN sequences remained in sync.

8.4.5 Discussion

The use of correlation to determine the lock direction, i.e. positive or negative lock, and to determine whether the input to the PN sequence synchronisation network has been inverted or not is more optimal than the original spike counting. Using correlation also allows the system to operate at lower signal-to-noise ratios when data is included. The system was evaluated at various signal-to-noise ratios. The system still synchronised at -16.1 dB (noise bandwidth half the simulation bandwidth.)

8.5 Comparison with Existing Methods

In chapter 2 it was shown that the Baseband Full-Time Early-Late Tracking Loop was sensitive to data, and therefore had little practical application unless the PN sequence occurred enough times within a bit to allow for resynchronisation in each bit interval. The novel method presented in this dissertation is also sensitive to noise. Again it would have been possible to have a high chipping rate and allow the system to re-acquire lock in each interval. The advantage of the novel synchronisation network is that it does not require the dual synchronisation structure. This approach is however not practical and it is for this reason that the non-coherent techniques are popular.

For the novel synchronisation network, extra circuitry was designed to allow the system to work effectively under the influence of data. The first design counted spikes to determine the direction of lock and the current data bit. The design, although not suited to lower signal-to-noise ratio applications served as the basis for designing a more effective method of dealing with data.

The circuitry that detects the data and lock condition was redesigned using correlation methods to detect the lock condition and the data. This design was a remarkable improvement over the spike counting design and the system was evaluated successfully at a signal-to-noise ratio of -16 dB. It must be highlighted that at very low signal-to-noise ratios the non-coherent early-late techniques are more effective. However it was the objective of this dissertation to find a cost-effective, easy to implement system, that achieved synchronisation and tracking without the addition of extra synchronisation networks at the cost of having to operate at higher signal-to-noise ratios.

8.6 Summary

In this chapter the more practical issues of communication systems were discussed. Issues like carrier tracking, noise and data were discussed and simulated. It was shown that the system operated successfully at a signal-to-noise ratio of -18 dB (in half the simulation bandwidth). The novel synchronisation network design was extended to deal with the influence of data on the synchronism of the network. Two designs were presented, one which operates at a high signal-to-noise ratios and one that operates at a lower signal-to-noise ratio.

Chapter 9

Practical Implementation

In the previous chapters simulation and theoretical techniques were used to analyse the novel synchronisation networks presented in this dissertation. However simulation and practical implementations do not always agree. To validate the simulation results and to verify that the simulations were legitimate, practical systems were built. The practical systems were also built to show that the implementation is indeed simple and low-cost. In the practical systems a novel wideband, low noise squarewave differentiator was designed and the phase-locked loop was implemented with a synchronous oscillator [51][52]. Both of these devices will be discussed as they have been extensively used in the practical systems that have been constructed. The practical work was done in part by Mark Davidson¹ under the direct supervision of the author. (See [41] for details on the practical implementations and results.)

9.1 Synchronous Oscillator

Injection locked oscillators are a method of achieving synchronisation. They combine a fast frequency locking time with an independent noise and tracking bandwidth. They do however introduce a measure of phase shift between their input and output. Uzunoglu and White [51] presented a new synchronisation network that performed the same function as an injection locked oscillator, but had superior characteristics. It was claimed that the synchronous oscillator was in fact not an injection locked oscillator at all. In [52] it is shown that although the synchronous oscillator is superior to the inject locked oscillator, it shows

¹Mark Davidson is a Masters student working in the same research group.

a phenomenon known as the 'current hump'. The current hump is an inherent phenomenon displayed by all injection locked oscillators. [53].

The synchronous oscillator has some interesting characteristics which makes it well suited to the synchronisation system presented in this dissertation. The network has the following characteristics:

- Flat tracking range.
- Storage time - circuit will maintain lock when no input is present for a period of time before losing lock.
- Tracking range narrows as input level decreases - improved noise performance over phase-locked loop.
- Fast locking time - the output frequency of the oscillator is a function of the instantaneous input frequency (no integration).

Given these attributes, the synchronous oscillator formed the basis of the tracking loop [54]. A number of tests were performed on the spread spectrum synchronisation network.

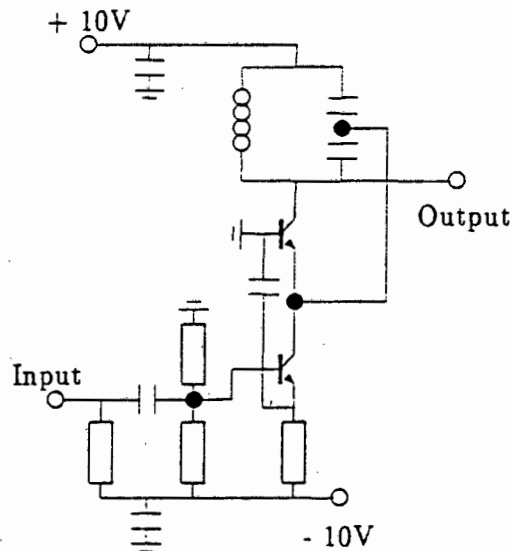


Figure 9.1: Circuit diagram of the synchronous oscillator.

9.2 Wideband Low Noise Square-Wave Differentiator

Signed edge detection (or square-wave differentiation) forms the basis of the novel synchronization network. Practically the standard differentiator implementation has both frequency and bandwidth limitations which are determined by the slew-rate of the operational amplifier. The differentiator also generates a fair amount of self-noise, which is compensated by adding a pole to the network. However this does not improve the bandwidth, but reduces it. Figure 9.2 shows an active differentiator and figure 9.3 its frequency response.

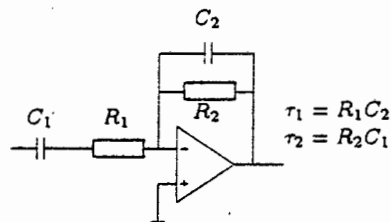


Figure 9.2: Active Differentiator

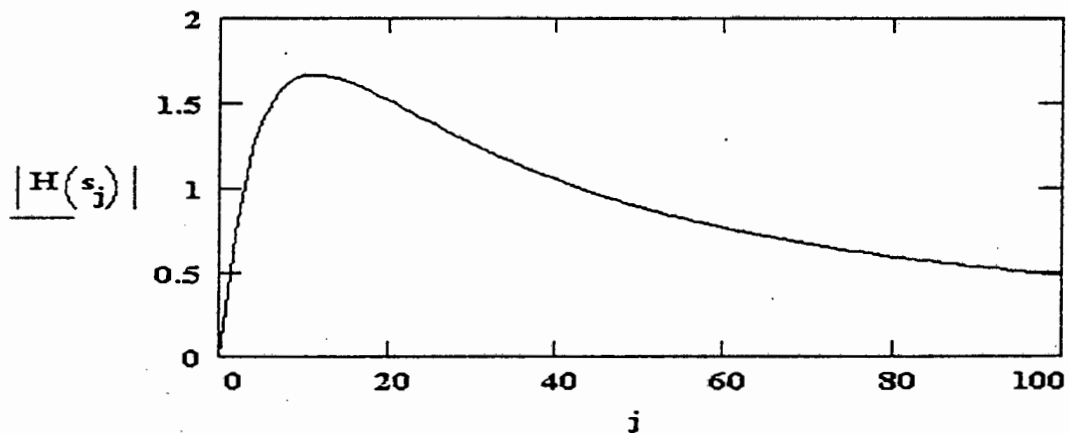


Figure 9.3: Frequency Response of Differentiator

9.2.1 Positive Edge Detection

There is an alternative method of generating a positive going spike for every edge of a square wave. (See Figure 9.4). This circuit is often used to recover the clock of data. Assuming that the input unipolar square wave swings between 0 and 5 volts, the delay introduced

by the exclusive OR gates tied to ground causes a glitch at the output. A spike is thus generated for every edge.

As the circuit is based on level comparison the edge detector is a low noise implementation as the input is only sampled when a marked change in state occurs. Furthermore the gate switches fast and therefore does not have the slew rate problems of opamps. The most important advantage of this circuit is that the delay value can be set according to the application. Thus the effect of the harmonics can be reduced by making the delay longer. This is due to the relationship between time and frequency. A train of square pulses in the time domain translates to a train of impulses in the frequency domain that are weighted by a $\frac{\sin(x)}{x}$ function. As the pulses are widened in the time domain so the $\frac{\sin(x)}{x}$ function narrows in the frequency domain and therefore performs a 'filtering' operation by reducing the higher harmonics.

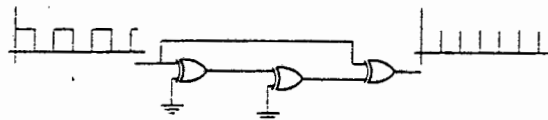


Figure 9.4: Digital Edge Detector

9.2.2 Signed Edge Detection

Edge detection does not truly differentiate the input square wave as the polarity of the impulses is lost. This led to the implementation of a 'signed edge detector'. It is possible to modify the edge detector (See Figure 9.4) by adding a few analogue components (See Figure 9.5) to generate signed pulses for every edge. The circuit retains the advantages of the edge detector while multiplying the output of the edge detector by the input square wave results in the correct polarity spikes. The circuit thus implements a low noise wideband square wave differentiator. The bandwidth of the output is limited only by the speed of the digital circuitry and the bandwidth of the multiplier.

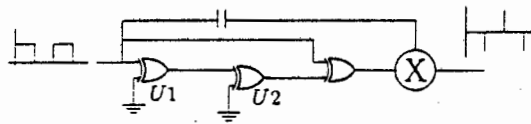


Figure 9.5: Signed Digital Differentiator

9.2.3 Comparison: Practical

The various implementations were built and tested in a noiseless environment. Under these conditions the signed edge detector was better than the differentiator. Although noise performance is equivalent, the signed edge detector spikes are cleaner and more defined.

Signed Edge Detector

The measured results of the signed edge detector are shown in figure 9.6. The input to the circuit is shown in figure 9.7. The output spikes display little overshoot or ringing. The output is clearly wideband due to the width of the spikes.

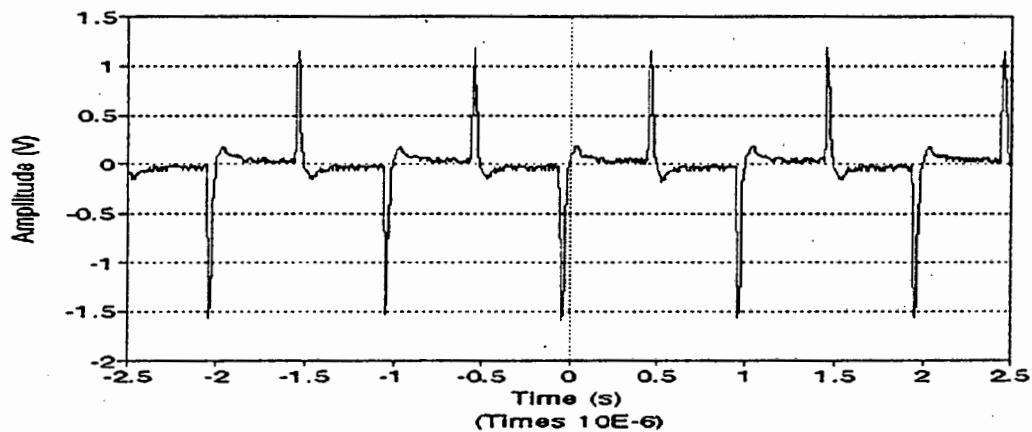


Figure 9.6: Output of Signed Edge Detector

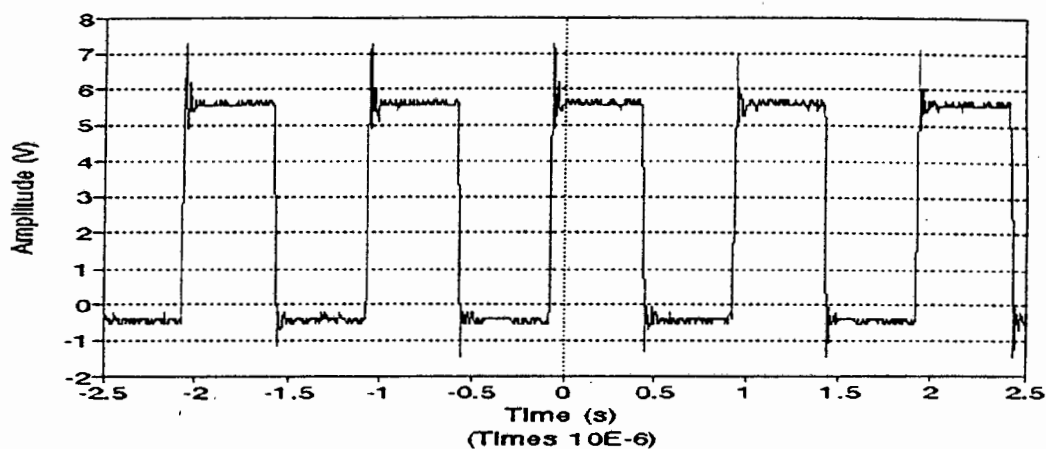


Figure 9.7: 1 MHz input square wave

Analog Differentiator

The measured results of the analog differentiator are shown in figure 9.8. The input to the circuit is shown in figure 9.9. Due to the integration component the spikes are not as sharp as those of the signed edge detector and approach zero volts exponentially.

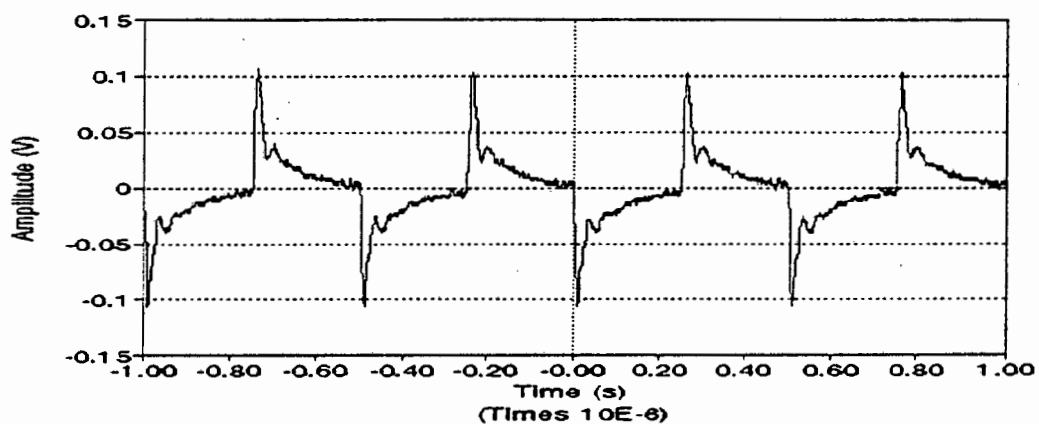


Figure 9.8: Output of Analog differentiator

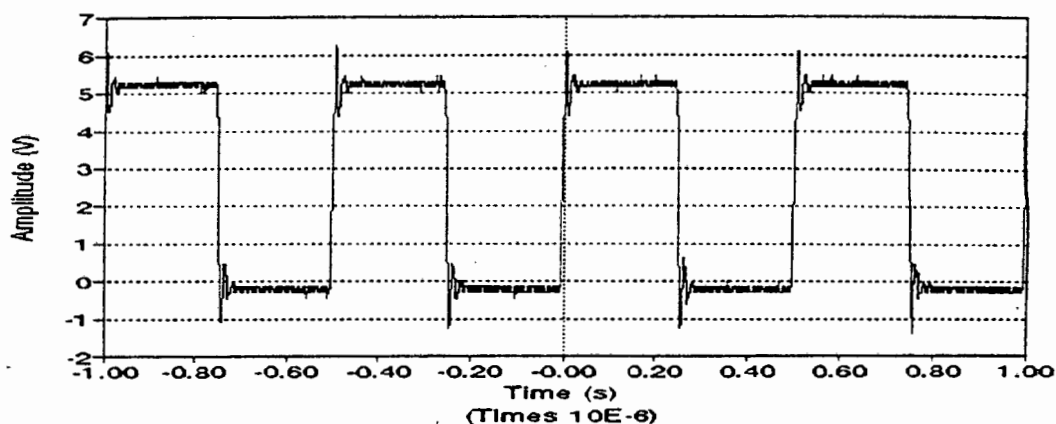


Figure 9.9: 1 MHz input square wave

9.2.4 Comparison: Simulation

The circuits shown in figures 9.2 and 9.5 were then simulated under noisy conditions. This was done using the simulation package TESLA² and the model shown in figure 9.10. The RC low-pass filter removes as much noise as possible. The input waveform to the filter is shown in figure 9.11 and the filtered output, which is fed to the differentiators is shown in figure 9.12.

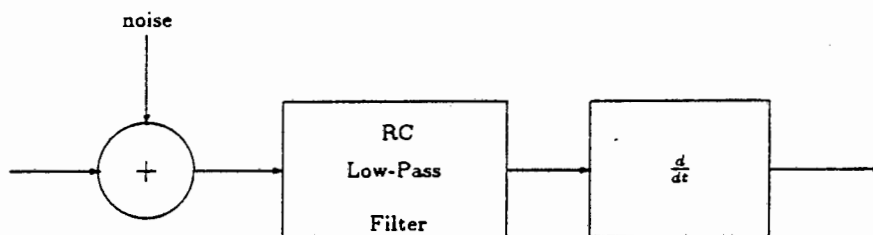


Figure 9.10: Simulation model

The results for the various implementations are shown in figures 9.13 and 9.14

Examining these figures it is clear that the signed edge detector implementation is far more

² © 1990-1993, by TESOFTE, Inc.

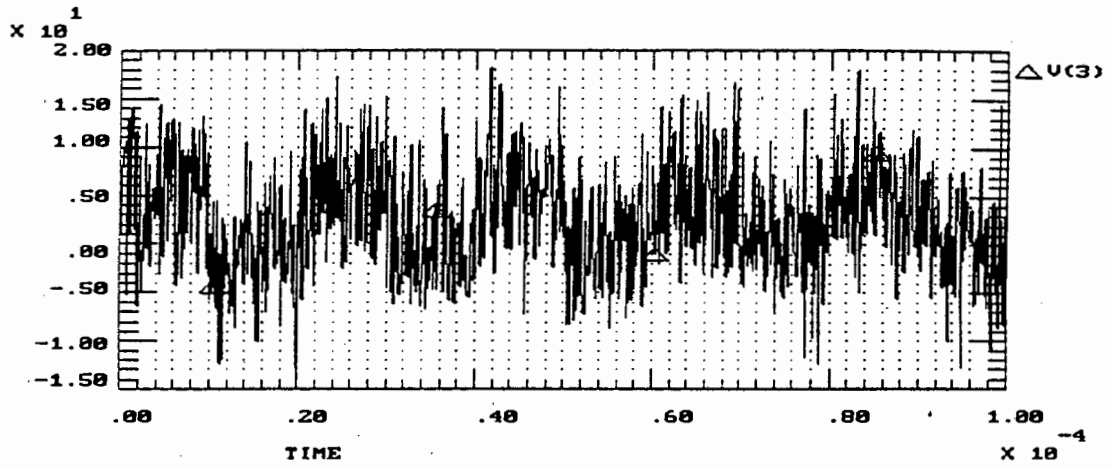


Figure 9.11: Input waveform

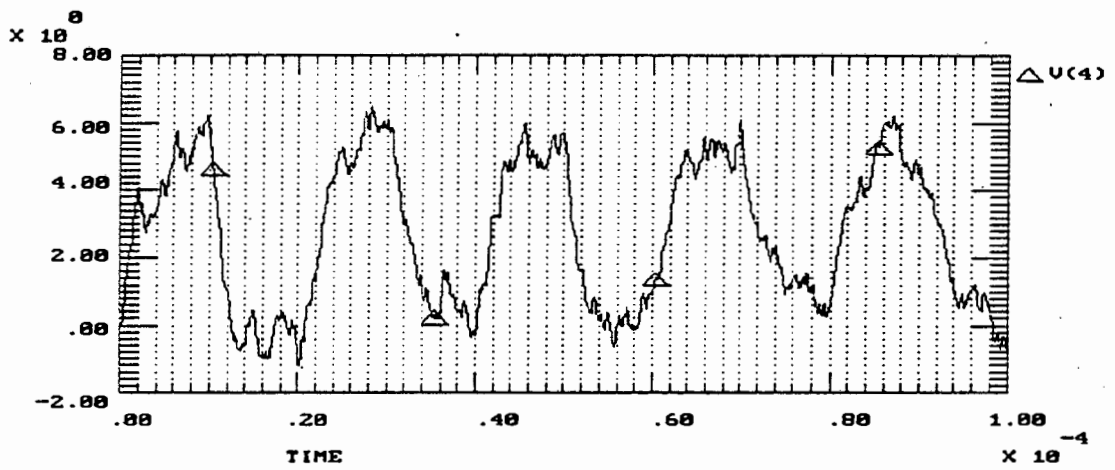


Figure 9.12: Filtered waveform

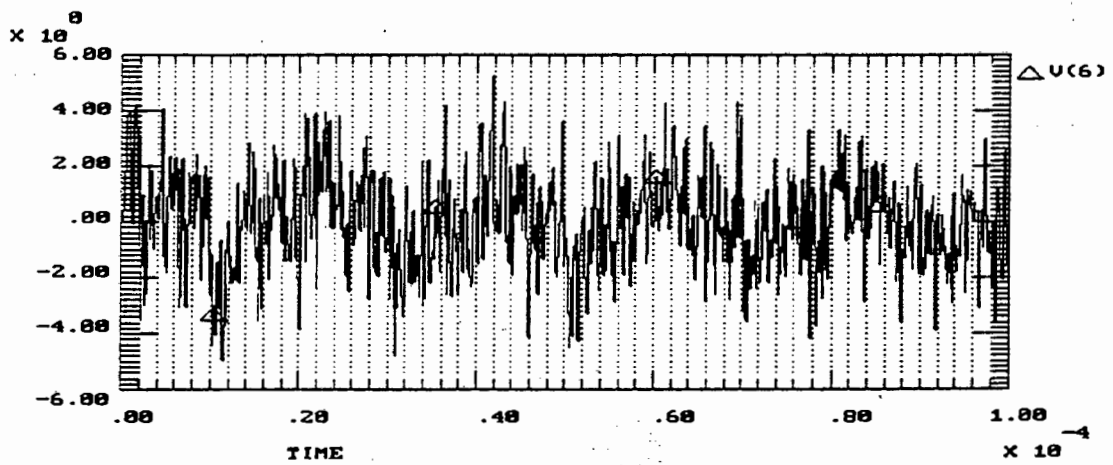


Figure 9.13: Ideal Differentiator

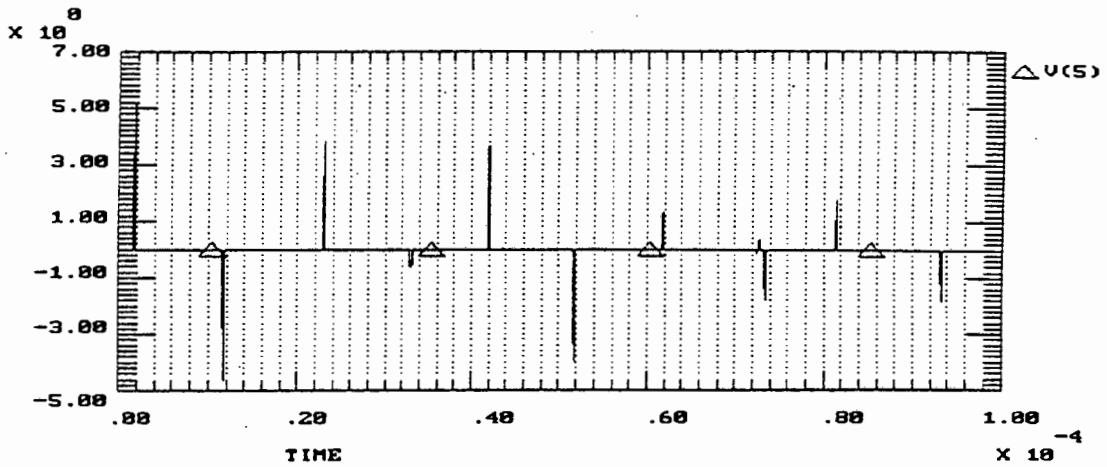


Figure 9.14: Signed Digital Differentiator

successful than the other implementations in differentiating a square wave in a noisy environment. It is possible by placing a one bit quantizer before the input to the differentiator block to improve the differentiated square-waves in both cases.

9.3 Practical Results: Novel Method One

For the practical system variable length pseudo-noise sequence generators were built [54]. The edge detected locally generated pseudo-noise sequence and the received sequence were multiplied using a Mini-circuits SBL-1 double balance mixer. Verification of lock was determined by the number of negative going spikes and by comparing the two sequences. Figure 9.15 shows the two sequences, while figure 9.16 shows the product of the edge detected and received sequences. From figure 9.15 it can be seen that the sequences are offset by one bit. This can be overcome by including a one-chip delay.

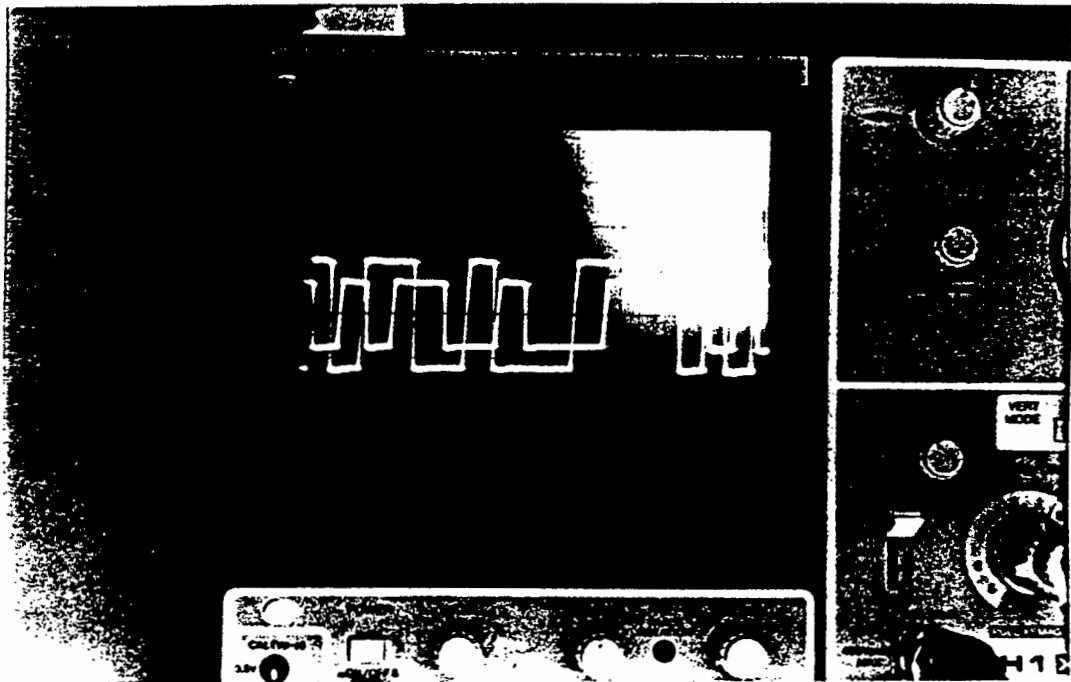


Figure 9.15: Comparison of Received and Locally Generated Sequences

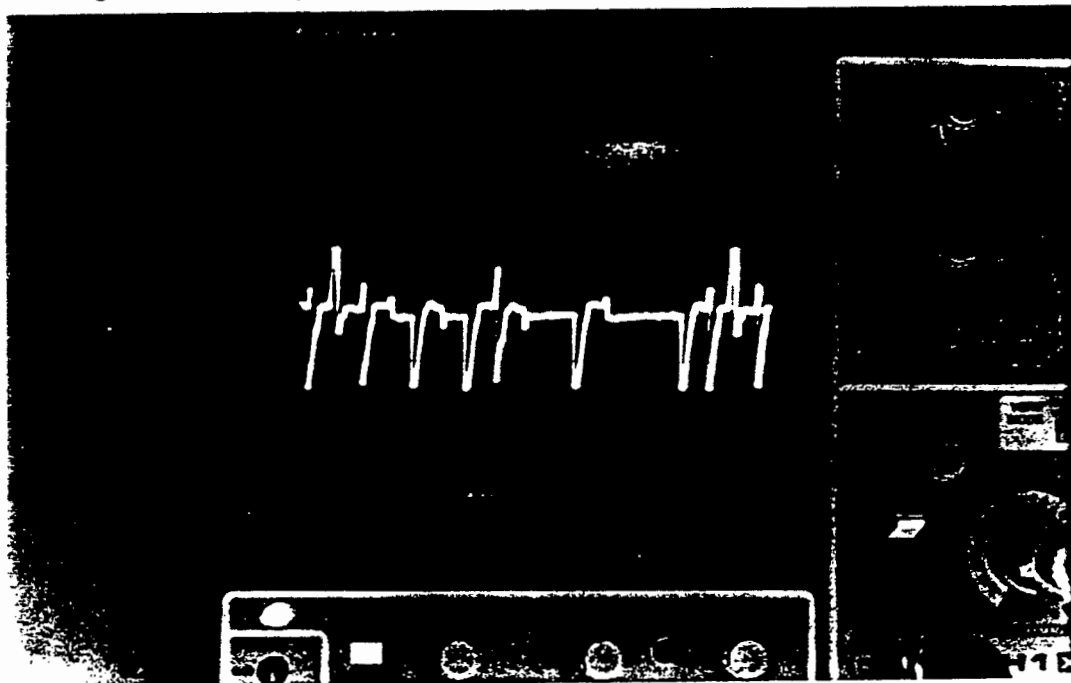


Figure 9.16: Product of the Edge Detected and Received Sequences

The problems experienced with the system losing lock or jittering were not experienced, although the system did often land up in a metastable condition. The other difference between the simulation and the practical implementation was that in the simulation it was not possible to use a Synchronous Oscillator, where there is no need to design for noise and loop bandwidths as in a phased-locked loop. The system did, however, often appear

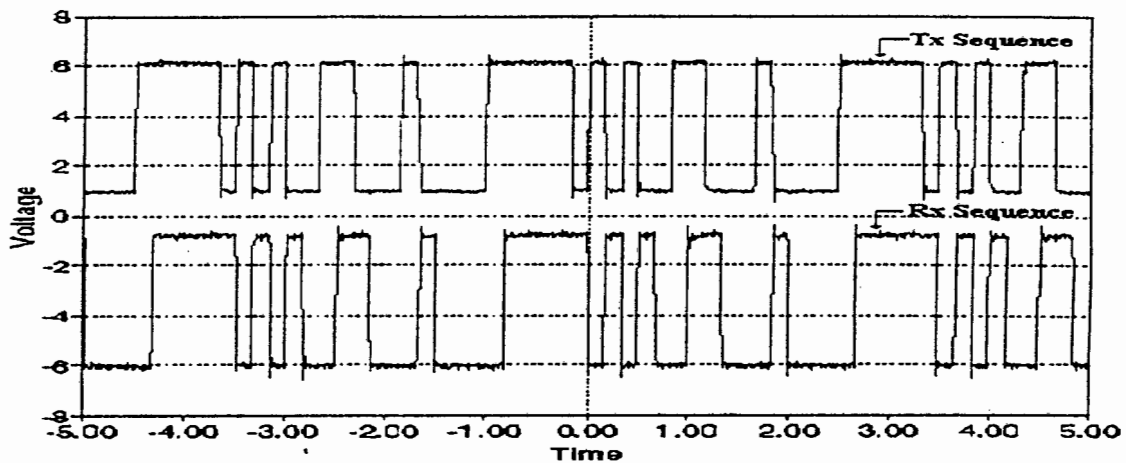


Figure 9.17: 15 Bit Maximal PN Sequence

to be in-lock, but the sequences were not aligned. This occurred most often very near the centre frequency and at the ends of the tracking range. The system was also subjected to the varying of the transmitted frequency. The system coped well with the sudden changes in frequency and did not lose lock.

9.4 Practical Results: Novel Method Two

Novel method two was built using a phase-locked loop as opposed to the use of the synchronous oscillator [41]. The choice of the phase-locked loop over the synchronous oscillator was based on the fact the phase-locked loop was used in the simulations. It is still felt that the Synchronous Oscillator is a viable choice in the design of the network. The design and practical circuits are detailed in [41]. Only the transmitted and locally generated PN sequences are compared. The system was built at 6 MHz and the system was implemented using the block diagrams given in the previous chapters.

9.4.1 Synchronisation without Data

Figures 9.17 through 9.19 show the results of the practical system. In all three cases it is clear that system is in-lock. The practical systems were run for more than 24 hours. In that time the system maintained lock. Other interesting factors that emerged from the design was the insensitivity to doppler shift. The transmit frequency was shifted in various steps and the system maintained lock. In figure 9.17 the system locked with a one bit offset. In figures 9.18 and 9.19 the systems had no lock offset.

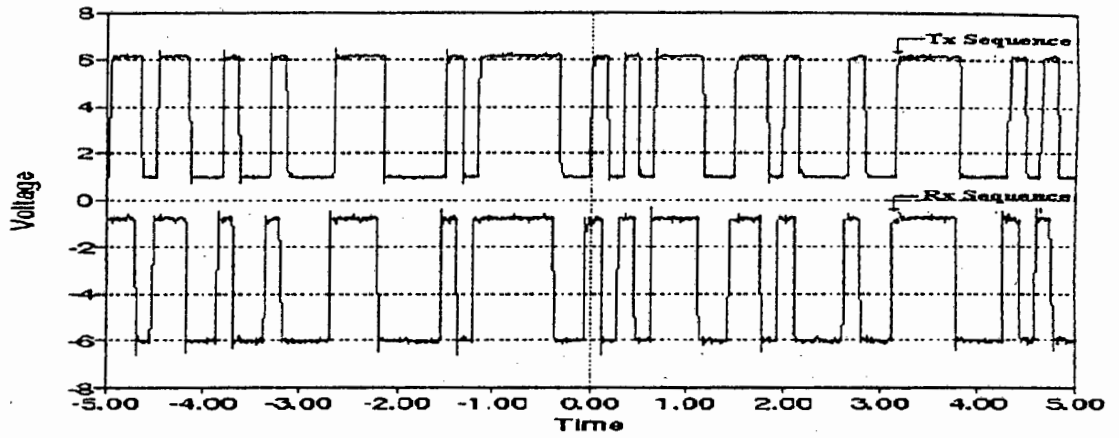


Figure 9.18: 32 Bit Maximal Length Sequence

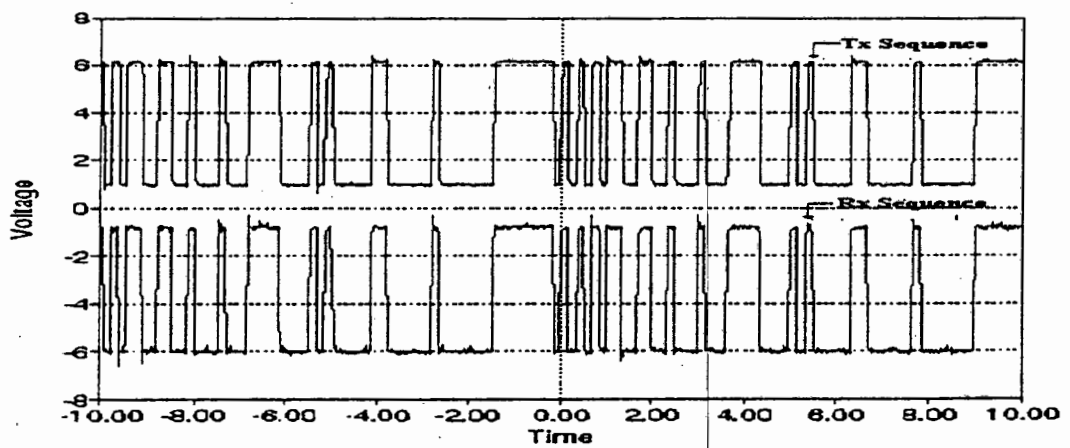


Figure 9.19: 63 Bit Maximal Length Sequence

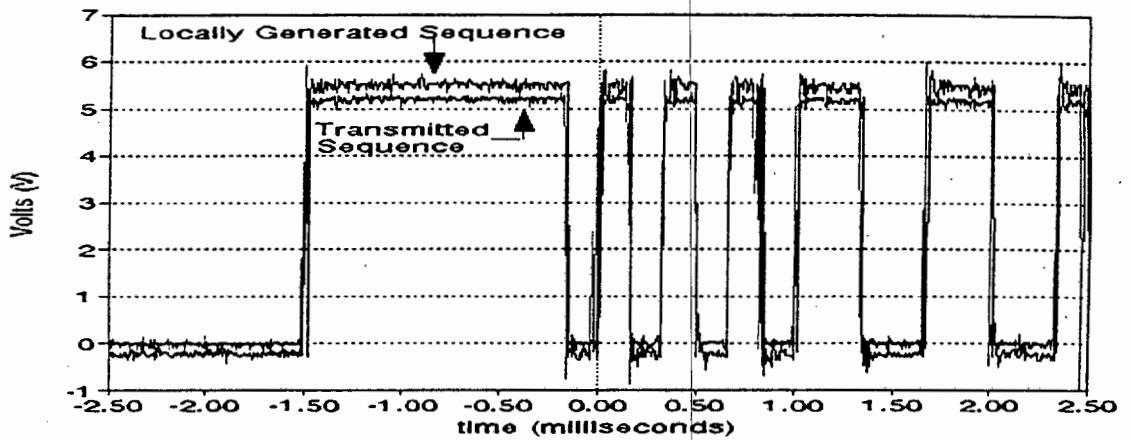


Figure 9.20: Transmitted and Locally Generated PN Sequences

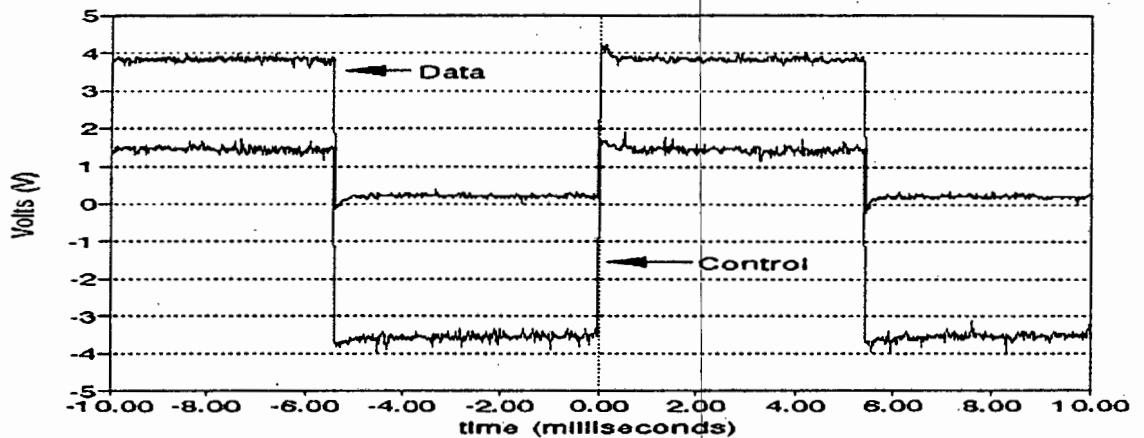


Figure 9.21: Transmitted and Locally Generated PN Sequences

9.4.2 With the Inclusion of Data

The practical circuit was extended to include data. The hardware, although a crude implementation, illustrated that data did not effect the synchronisation of the system after lock was achieved. The system was evaluated using a 127 maximal length sequence. Figure 9.20 shows the PN sequences in lock. The system achieved lock without the one bit offset.

Figure 9.21 shows the data signal and the control signal that switches the sign of the spikes entering the synchronisation network. As would be expected the two signals are fully aligned.

9.5 Summary

In this chapter some basic results of the practical implementations were presented. Also discussed were some of the areas investigated as part of the practical implementation. A novel wide-band signed edge detector was analysed, designed and tested. This novel design was used as part of the practical circuits in place of the differentiator.

Chapter 10

Conclusion

In this dissertation a novel synchronisation network was postulated, analysed, evaluated and designed that offers a unified approach to synchronisation of spread spectrum systems. The novel designs do not require an acquisition and tracking network as do the methods described in chapter 2. The primary characteristics of the network are:

- The network is coherent, that is carrier demodulation is required before code tracking.
- The network is a unified approach to pseudonoise sequence synchronisation and tracking.
- The system operates successfully at a signal-to-noise ratio of -18 dB.
- Synchronisation time is within the same order as the methods presented in chapter 2.
- The system is low-cost and easy to implement.

Given these characteristics, the following constraints are placed on the system:

- The system requires a synchronisation time window to achieve lock.
- The signal-to-noise ratio requirement is higher than that of early-late tracking networks. The primary reason for this the use of differentiation in the network.

To achieve the objective of obtaining a network that is low-cost, and achieves acquisition and tracking in one implementation without the need for an intermediate decision stage, the system was developed using three methods of evaluation, namely:

- Simulation.
- Non-linear Analysis based on phase-plane plots.
- Practical Implementation.

The method of research is clearly represented by figure 1.1, repeated here for convenience. The purpose of chapter 2 was to lay the foundation of what has been done in spread spectrum acquisition and tracking theory. The chapter served as a basis for comparison of the novel synchronisation network.

In chapter 3 the power of time variant spectral analysis was discussed. Using simple examples, a number of concepts were presented which served as the basis for the postulation of two networks that perform synchronisation without the need of the dual acquisition and tracking structure. Using the basic results, the first novel network was presented in chapter 4. It was shown that the postulated synchronisation does achieve the objective.

Due to the differentiator being placed in the feedback loop, the system was unstable and sensitive to parameter changes. A practical system (chapter 9) was built using a synchronous oscillator and the differentiator was implemented using a novel wide-band square-wave differentiator. The system functioned well, although often achieved false lock.

A second system was postulated in chapter 5. The system did not suffer from the problems of the first implementation. The system was analysed and compared to the methods described in chapter 2. The system was implemented using a phase-locked loop and the novel wide-band square wave differentiator. The system performed well.

Chapters 6 and 7 analysed the network using non-linear analysis. The basics of phase-plane plots were discussed. The analysis was applied to phase-locked loops and then extended to include a clock recovery network. The use of phase-plane plots to show the effect of varying parameters was shown. In chapter 7 the analysis was extended to the complete novel synchronisation network. An extensive analysis using phase-plane plots was presented. It was shown that each is unique and highlighted the fact that the analysis of the frequency-code phase grid is random.

Issues of noise, carrier tracking and data were addressed in chapter 8. It was shown that the network, with the addition of a few components, can operate at signal-to-noise ratios of -18 dB. Since the network can only synchronise at baseband, the signal-to-noise ratio of the carrier tracking places another constraint on the network. As the signal-to-noise

ratio is reduced, so the acquisition time increases. In all simulations a locking time of 40 milliseconds was placed on the overall synchronisation of the system. The system including data and noise was simulated. The design using correlation as opposed to impulse counting was presented and shown to be effective. The system with data was evaluated practically. Some of the measured results are shown in chapter 9.

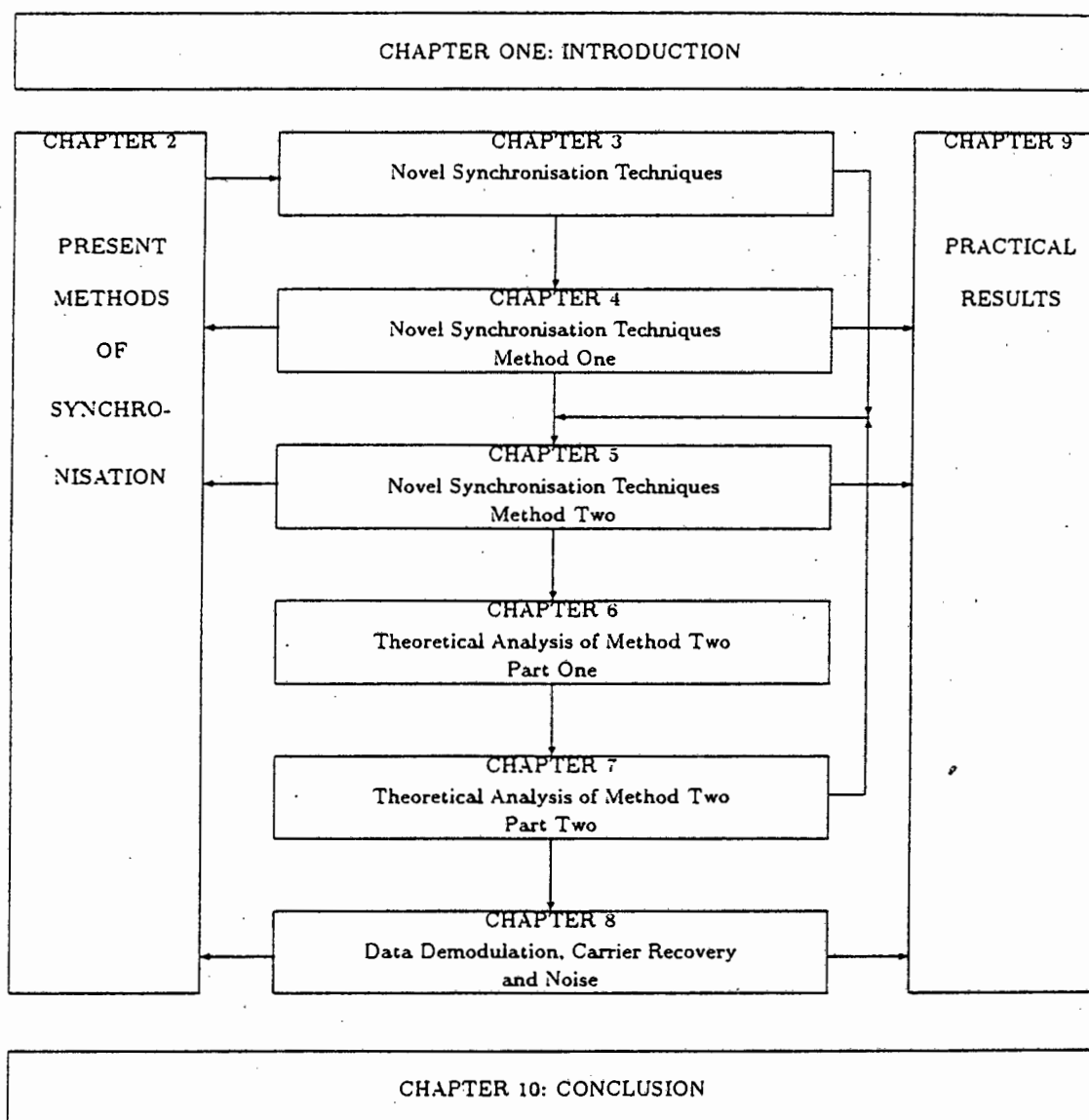


Figure 10.1: Dissertation Breakdown

10.1 Recommendations

The following sections discussed suggested areas of research to improve the performance of the network.

10.1.1 Longer Pseudonoise Sequences

The synchronisation of very long sequences always poses a problem in spread spectrum systems. Systems like GPS use a shorter sequence to allow for rapid synchronisation of the longer sequence. Depending on the application more intelligence is required in the system. The novel network presented in this dissertation suffers from the same problems. However the network is well suited to the use of RASE techniques, but it is not suggested that the shift register be load with the detected states. Using the detected chipping bits and doing a best match by comparing to a lookup table, and then only loading the shift register, (more than one shift register if using Gold sequences) with a valid initialization string it is possible to deal effectively with long pseudo-noise sequences.

10.1.2 Noise

Although the system failed below -18 dB with further design and more powerful simulation tools and equipment, it should be possible to improve the system to synchronisation at lower signal-to-noise ratios. The effect of non-Gaussian conditions also needs further investigation.

10.1.3 Detection of PN sequence

The detection of PN chips as part of the method of synchronisation is one of the drawbacks of the system under the influence of noise. The detection method needs further investigation, including the possibility of using a matched filter at the input.

10.1.4 Carrier Tracking

The fact that carrier tracking was required before code tracking places restrictions on the practical implementation of the novel synchronisation network. Initial design shows that it could be possible to implement a non-coherent solution and this is worth investigating further.

10.1.5 Applications and Improvements

Spread spectrum is not only used for multiple access applications, but has many other uses. The novel synchronisation network developed in this thesis is more suited for higher signal-to-noise ratio application. This factor makes it an unlikely choice for very complex spread spectrum systems (like Mobile Networks). It is believed that it is possible with additions and improvements to implement the synchronisation system at lower signal-to-noise ratios.

Appendix A

Simulation of Spilker Delay-Lock Discriminator Error Signal

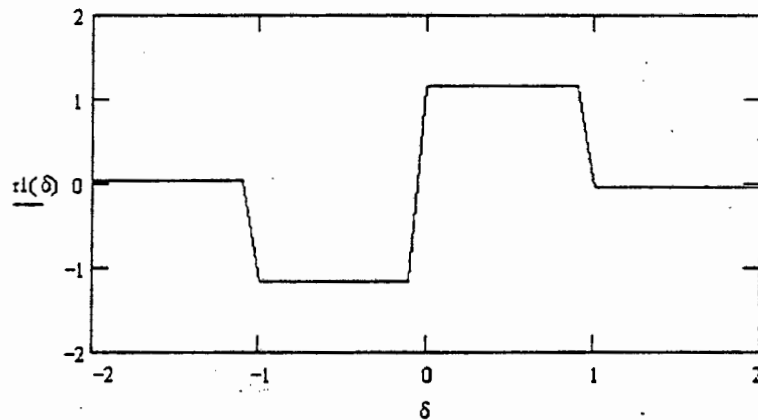
Simulation of the error signal used to drive the delay line $q := 0..6$
 Specify the range of calculations and function definitions: $T_c := 1$

$n := 1..20$ $\delta := -T_c - 1, -T_c - 1 + 0.1..T_c + 1$
 $m := 1..20$ $L := 7$ $u(z) := \Phi(z)$

$a_q :=$

1
1
1
-1
1
-1
-1

$$r1(\delta) := \frac{1}{20} \left[\sum_n \sum_m a_{\text{mod}(n,L)} a_{\text{mod}(m,L)} \left[\begin{array}{l} u[\delta + (m-n) \cdot T_c] - u[\delta + (m-n-1) \cdot T_c] \dots \\ + u[\delta + (m+1-n) \cdot T_c] \dots \\ + u[\delta + (m-n) \cdot T_c] \end{array} \right] \right]$$



Appendix B

Derivation of Partial-Correlation Function

B.1 Derivation

The correlation of a maximal length sequence is two valued when correlated over the complete period of the sequence. When dealing with delays in the channel and not correlating over the complete period, the correlation is not two-valued and depends on the delay between the two sequences and the correlation window. For proper analysis of what the optimal maximal length sequence is and the chipping factor, the partial correlation should take these factors into account. The following analysis is from [55].

Define the chipping wave-form $c(t)$ as,

$$c(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT_c) \quad (\text{B.1})$$

and the partial correlation $R_{cc}(t)$ as

$$R_{cc}(t) = \frac{1}{T_b} \int_t^{t+T_b} c(\lambda)c(t + \lambda)d\lambda \quad (\text{B.2})$$

Substituting equation (B.1) in equation (B.2),

$$R_{cc}(t) = \frac{1}{T_b} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_n a_m \int_t^{t+T_b} p(\lambda - mT_c) p(\lambda - nT_c + \tau_d) d\lambda \quad (\text{B.3})$$

Only the correlation over a bit period is of interest and therefore define the starting time of integration t as $t = iT_b$, and the delay τ_d over the interval $0 \leq \tau_d \leq T_b$. Therefore equation (B.3) becomes

$$R_{cc}(iT_b) = \frac{1}{T_b} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_n a_m \int_{iT_b}^{(i+1)T_b} p(\lambda - mT_c) p(\lambda - nT_c + \tau_d) d\lambda \quad (\text{B.4})$$

Let $\gamma = \lambda - iT_b$. Equation (B.4) may be rewritten as:

$$R_{cc}(iT_b) = \frac{1}{T_b} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_n a_m \int_0^{T_b} p(\gamma + iT_b - mT_c) p(\gamma - nT_c + iT_b + \tau_d) d\gamma \quad (\text{B.5})$$

For simplicity, redefine τ_d as

$$\tau_d = kT_c + T_d \quad (\text{B.6})$$

where $0 \leq T_d \leq T_c$. Substituting equation (B.6) into ((B.5),

$$R_{cc}(iT_b) = \frac{1}{T_b} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_n a_m \int_0^{T_b} p(\gamma + iT_b - mT_c) p(\gamma - (n - k)T_c + iT_b + T_d) d\gamma \quad (\text{B.7})$$

The integral of equation (B.7) is non-zero only for $n = m + k$ or $n = m + k + 1$ and therefore may write the correlation function as,

$$\begin{aligned} R_{cc}(iT_b) &= \frac{1}{T_b} \sum_{m=-\infty}^{\infty} a_{m+k} a_m \int_0^{T_b} p(\gamma + iT_b - mT_c) p(\gamma - mT_c + iT_b + T_d) d\gamma \\ &+ \frac{1}{T_b} \sum_{m=-\infty}^{\infty} a_{m+k+1} a_m \int_0^{T_b} p(\gamma + iT_b - mT_c) p(\gamma - (m+1)T_c + iT_b + T_d) d\gamma \quad (\text{B.8}) \end{aligned}$$

Let $\alpha = -iT_b + mT_c$, equation (B.8) is rewritten as

$$\begin{aligned}
 R_{cc}(iT_b) &= \frac{1}{T_b} \sum_{m=-\infty}^{\infty} a_{m+k} a_m \int_0^{T_b} p(\gamma - \alpha) p(\gamma - \alpha + T_d) d\gamma \\
 &+ \frac{1}{T_b} \sum_{m=-\infty}^{\infty} a_{m+k+1} a_m \int_0^{T_b} p(\gamma - \alpha) p(\gamma - \alpha - T_c + T_d) d\gamma
 \end{aligned} \quad (\text{B.9})$$

The integrand of equation (B.9) is non-zero within when the limits of integration when $0 \leq \alpha \leq T_b - T_c$. This implies that the limits of the summation may be reduced to $iL \leq m \leq (i+1)L - 1$, where $L = \frac{T_b}{T_c}$. Therefore,

$$\begin{aligned}
 R_{cc}(iT_b) &= \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m \int_0^{T_b} p(\gamma - \alpha) p(\gamma - \alpha + T_d) d\gamma \\
 &+ \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m \int_0^{T_b} p(\gamma - \alpha) p(\gamma - \alpha - T_c + T_d) d\gamma
 \end{aligned} \quad (\text{B.10})$$

For any fixed value of m , the integrand of the first integral is non-zero only for

$$mT_c - iT_b \leq \gamma \leq (m+1)T_c - iT_b - T_d$$

and the second the integrand of the second integral is non-zero only for

$$(m+1)T_c - iT_b - T_d \leq \gamma \leq (m+1)T_c - iT_b$$

Rewriting equation (B.10) in terms of these new integration limits,

$$\begin{aligned}
 R_{cc}(iT_b) &= \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m \int_{mT_c - iT_b}^{(m+1)T_c - iT_b - T_d} p(\gamma - \alpha) p(\gamma - \alpha + T_d) d\gamma \\
 &+ \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m \int_{(m+1)T_c - iT_b - T_d}^{(m+1)T_c - iT_b} p(\gamma - \alpha) p(\gamma - \alpha - T_c + T_d) d\gamma
 \end{aligned} \quad (\text{B.11})$$

Let $z = \gamma - \alpha$. Therefore,

$$\begin{aligned}
R_{cc}(iT_b) &= \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m \int_0^{T_c-T_d} p(z)p(z+T_d)dz \\
&+ \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m \int_{T_c-T_d}^{T_c} p(z)p(z+T_d)dz \\
&= \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m (T_c - T_d) + \frac{1}{T_b} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m (T_d) \quad (\text{B.12})
\end{aligned}$$

Multiplying through by T_c and noting that $L = \frac{T_b}{T_c}$, equation (B.12) becomes,

$$R_{cc}(iT_b) = \frac{1}{L} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m \left(1 - \frac{T_d}{T_c}\right) + \frac{1}{L} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m \left(\frac{T_d}{T_c}\right) \quad (\text{B.13})$$

Equation (B.13) therefore defines the partial correlation as a function of the chipping factor (L), the chipping period (T_c), the data bit period (T_b), the delay ($T_d = \tau_d - kT_c$) and the present bit of interest (i).

B.2 Verification of Theoretical Formula

Let $T_b = N$, where N is the sequence length, $L=N$, and the delay is $\tau_d = 0$. Solving equation (B.13), $R_{cc}(iT_b) = 1$ for all i . This is the normalized peak value of an auto-correlation for zero delay.

Appendix C

Baseband Full-Time Early-Late Tracking Loop

Simulation of the error signal used to drive the delay line
Specify the range of calculations and function definitions:

$$\delta := -3, -2.9 \dots 3$$

$$\text{sign}(x) := \text{if}(x \geq 0, 1, -1)$$

$$m := 7 \dots 13 \quad L := 7$$

$$\text{frac}(x) := x - \text{floor}(x)$$

$$\text{int}(x) := \text{floor}(x)$$

$$T_c := 1$$

$$\Delta := 0.8$$

$$q := 0 \dots 6$$

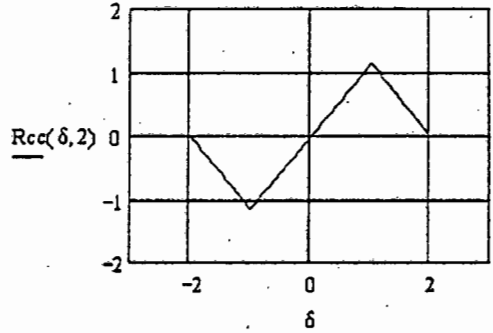
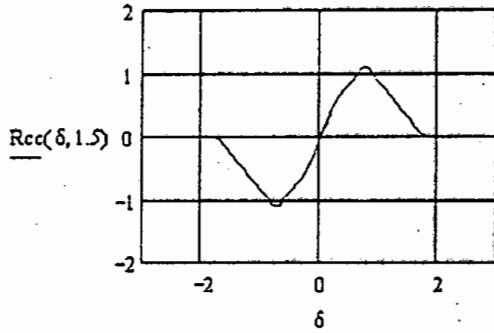
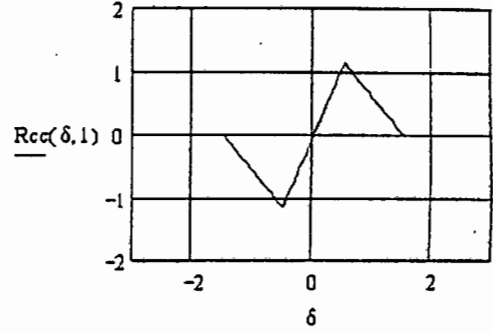
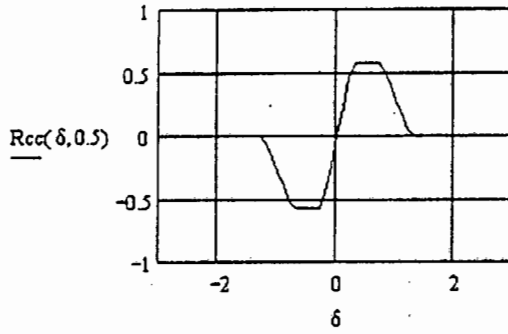
$$a_{q+7} := a_q$$

$$a_q :=$$

1
1
1
-1
1
-1
-1

$$\begin{aligned} \text{Rcc1}(\delta, \Delta) &:= \frac{1}{L} \left[\sum_m a_{\text{mod}(m,L)} a_{\text{mod}\left(m + \text{int}\left(\delta - \frac{\Delta}{2}\right), L\right)} \left[1 - \frac{\text{frac}\left(\left(\delta - \frac{\Delta}{2}\right)\right)}{T_c} \right] \dots \right. \\ &\quad \left. + \frac{1}{L} \sum_m a_{\text{mod}\left[\left[m + \text{int}\left(\left(\delta - \frac{\Delta}{2}\right)\right] + 1, L\right]} a_{\text{mod}(m,L)} \left[\frac{\text{frac}\left(\left(\delta - \frac{\Delta}{2}\right)\right)}{T_c} \right] \right] \\ \text{Rcc2}(\delta, \Delta) &:= \frac{1}{L} \left[\sum_m a_{\text{mod}(m,L)} a_{\text{mod}\left(m + \text{int}\left(\delta + \frac{\Delta}{2}\right), L\right)} \left[1 - \frac{\text{frac}\left(\delta + \frac{\Delta}{2}\right)}{T_c} \right] \dots \right. \\ &\quad \left. + \frac{1}{L} \sum_m a_{\text{mod}\left[\left(m + \text{int}\left(\delta + \frac{\Delta}{2}\right)\right) + 1, L\right]} a_{\text{mod}(m,L)} \left[\frac{\text{frac}\left(\delta + \frac{\Delta}{2}\right)}{T_c} \right] \right] \end{aligned}$$

$$R_{cc}(\delta, \Delta) := R_{cc1}(\delta, \Delta) - R_{cc2}(\delta, \Delta)$$



Appendix D

Derivation of Pseudo-noise Sequence Spectrum

The spectrum of Pseudonoise sequence can be obtained using a number of methods. The first is determining the fourier series and then taking the fourier transform of the resultant series. This is a long and complicated process. The second method is determining the correlation function and then taking the fourier transform.

It was shown in equation B.13 that the correlation function is

$$R_{cc}(iT_b, T_d) = \frac{1}{L} \sum_{m=iL}^{(i+1)L-1} a_{m+k} a_m \left(1 - \frac{T_d}{T_c}\right) + \frac{1}{L} \sum_{m=iL}^{(i+1)L-1} a_{m+k+1} a_m \left(\frac{T_d}{T_c}\right)$$

where: $T_d = \tau_d - kT_c$ and $0 \leq T_d \leq T_c$.

For the determination of the spectrum, the correlation function should be calculated over the period of the pseudo-noise sequence and evaluated over the intervals $0 \leq \tau_d \leq T_c$, $T_c < \tau_d < (L-1)T_c$ and $(L-1)T_c \leq \tau_d < LT_c$.

For $0 \leq \tau_d \leq T_c$ to be valid, $T_d = \tau_d$, (ie $k = 0$). Equation B.13 can be written as

$$R_{cc}(\tau_d) = \frac{1}{L} \sum_{m=0}^{L-1} a_m a_m \left(1 - \frac{\tau_d}{T_c}\right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+1} a_m \left(\frac{\tau_d}{T_c}\right) \quad (D.1)$$

Solving for the summations equation D.1 becomes:

$$\begin{aligned}
R_{cc}(\tau) &= 1 - \frac{\tau_d}{T_c} - \frac{1}{L} \left(\frac{\tau_d}{T_c} \right) \\
&= 1 - \frac{\tau_d}{T_c} \left[1 + \frac{1}{L} \right]
\end{aligned} \tag{D.2}$$

To solve for the interval $T_c < \tau_d < (L-1)T_c$, the range of T_d is required. Substituting for τ_d , the range of T_d becomes. $(1-k)T_c < T_d < (L-1)T_c - kT_c$. For the range to be valid, $k = 1$. Therefore equation B.13 can be rewritten as:

$$R_{cc}(T_d) = \frac{1}{L} \sum_{m=0}^{L-1} a_{m+1} a_m \left(1 - \frac{T_d}{T_c} \right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+2} a_m \left(\frac{T_d}{T_c} \right) \tag{D.3}$$

Solving the summations and simplifying, equation D.3 becomes,

$$R_{cc}(\tau_d) = -\frac{1}{L} \tag{D.4}$$

where $T_c < \tau_d < (L-1)T_c$.

Finally for the interval $(L-1)T_c \leq \tau_d < LT_c$, the same procedure used for the interval $T_c < \tau_d < (L-1)T_c$ is followed. The interval for T_d is therefore $(L-1-k)T_c \leq T_d < LT_c - kT_c$. For the interval to be valid, $k = L-1$. Therefore equation B.13 becomes:

$$R_{cc}(T_d) = \frac{1}{L} \sum_{m=0}^{L-1} a_{m+L-1} a_m \left(1 - \frac{T_d}{T_c} \right) + \frac{1}{L} \sum_{m=0}^{L-1} a_{m+L} a_m \left(\frac{T_d}{T_c} \right) \tag{D.5}$$

Solving for the summations and simplifying, equation D.5 becomes,

$$R_{cc}(T_d) = -\frac{1}{L} \left(1 - \frac{T_d}{T_c} \right) + \frac{T_d}{T_c} \tag{D.6}$$

Since the auto-correlation function is periodic, the fourier series is required. However using a well known result [38], the fourier transform of a fourier series under certain conditions can be written in the form

$$S(f) = \sum_{n=-\infty}^{\infty} S_n \delta \left(f - \frac{n}{LT_c} \right)$$

where $S_n = \frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} f(t) e^{-jn2\pi f_0 t} dt$

From this equation, the fourier spectrum of the pseudo-noise sequence can be obtained. Solving for S_n , the following is obtained:

$$S_n = \begin{cases} \frac{1}{L^2} & \text{for } n = 0 \\ \frac{L+1}{L^2} \text{Sa}\left\{\pi \frac{n}{L}\right\} & \text{for all other } n \end{cases} \quad (\text{D.7})$$

where $f_0 = \frac{1}{LT_c}$.

The overall spectrum is therefore defined as:

$$S(f) = \frac{1}{L^2} + \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \frac{L+1}{L^2} \text{Sa}\left\{\pi \frac{n}{L}\right\} \delta\left(f - \frac{n}{LT_c}\right) \quad (\text{D.8})$$

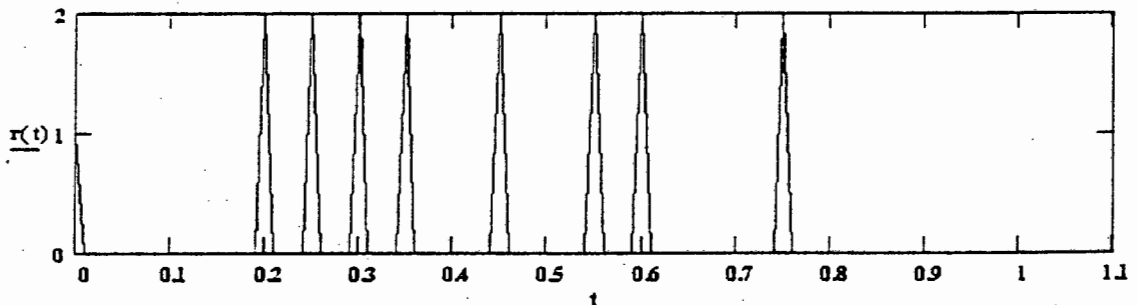
Appendix E

Simulations of Despreading Edge Detected PN Sequences

E.1 No Code-Phase Offset

$i := 0..15$ $j := 0..15$ $u(z) := \Phi(z)$ $\text{round}(x) := \text{if}(x - \text{floor}(x) < .5, \text{floor}(x), \text{ceil}(x))$
 $\delta(x) := \Phi(x) - \Phi(x - 0.01)$ $L := 15$ $T_c := 0.05$ $T_v(x) := 0.05$ $t := 0, 0.01..1$
 $q := 0..3$ $a_q := 1$
 $q := 4..15$ $a_q := \text{if}(a_{q-1} = a_{q-4}, 0, 1)$ $a_q := \text{if}(a_q = 0, -1, 1)$ $q := 0..15$ $b_q := a_{\text{mod}(q, L)}$

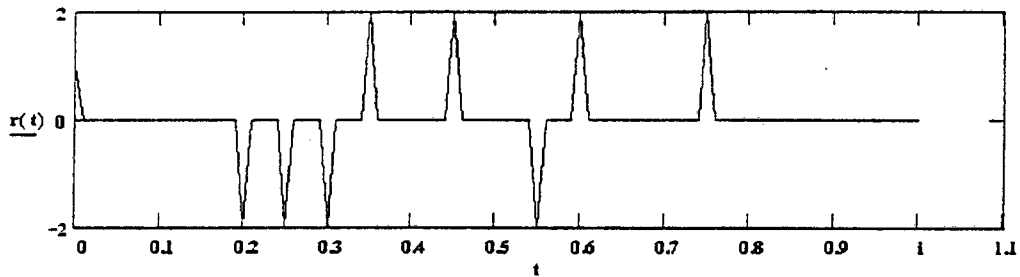
$$\begin{aligned}
 r(t) := & \sum_i \sum_j \left[a_{\text{mod}(i, L)} \cdot b_{\text{mod}(j, L)} \left[u(i \cdot T_c - j \cdot T_v(i \cdot T_c)) \dots \right. \right. \\
 & \left. \left. + (-1)^j \cdot u[i \cdot T_c - (j+1) \cdot T_v(i \cdot T_c)] \right] \cdot \delta(t - i \cdot T_c) \right] \dots \\
 & + (-1)^i \cdot \sum_i \sum_j \left[a_{\text{mod}(i, L)} \cdot b_{\text{mod}(j, L)} \left[u[(i+1) \cdot T_c - j \cdot T_v[(i+1) \cdot T_c]] \dots \right. \right. \\
 & \left. \left. + (-1)^j \cdot u[(i+1) \cdot T_c - (j+1) \cdot T_v[(i+1) \cdot T_c]] \right] \right] \cdot \delta[t - (i+1) \cdot T_c]
 \end{aligned}$$



E.2 One Chip Code-Phase Offset

$i := 0..15 \quad j := 0..15 \quad u(x) := \Phi(x) \quad \text{round}(x) := \text{if}(x - \text{floor}(x) < .5, \text{floor}(x), \text{ceil}(x))$
 $\delta(x) := \Phi(x) - \Phi(x - 0.01) \quad L := 15 \quad T_c := 0.05 \quad T_v(x) := 0.05 \quad t := 0, 0.01..1$
 $q := 0..3 \quad a_q := 1$
 $q := 4..15 \quad a_q := \text{if}(a_{q-1} = a_{q-4}, 0, 1) \quad a_q := \text{if}(a_q = 0, -1, 1) \quad q := 0..15 \quad b_q := a_{\text{mod}(q+1, L)}$

$$r(t) := \sum_i \sum_j \left[a_{\text{mod}(i, L)} \cdot b_{\text{mod}(j, L)} \cdot \left[u(i \cdot T_c - j \cdot T_v(i \cdot T_c)) \dots \right. \right. \\
 \left. \left. + (-1) \cdot u[(i+1) \cdot T_c - (j+1) \cdot T_v(i \cdot T_c)] \right] \cdot \delta(t - i \cdot T_c) \right] \dots \\
 + (-1) \cdot \sum_i \sum_j \left[a_{\text{mod}(i, L)} \cdot b_{\text{mod}(j, L)} \cdot \left[u[(i+1) \cdot T_c - j \cdot T_v[(i+1) \cdot T_c]] \dots \right. \right. \\
 \left. \left. + (-1) \cdot u[(i+1) \cdot T_c - (j+1) \cdot T_v[(i+1) \cdot T_c]] \right] \right] \cdot \delta[t - (i+1) \cdot T_c]$$



E.3 Fourier-Time Spectrum (All ones)

Evaluation of Spreading sequences

$$f(t) = d(t) + d(t - T_c) + d(t - 2T_c)$$

$$j := \sqrt{-1} \quad T_c := 1 \quad \omega := 0, 0.01 \cdot (2 \cdot \pi) .. 3 \cdot (2 \cdot \pi)$$

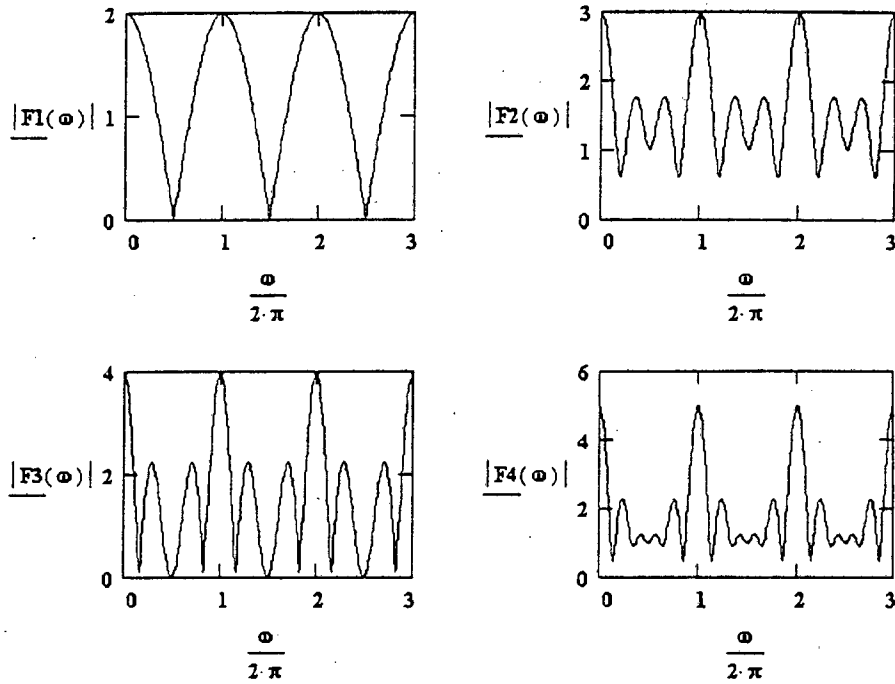
Case Study 1: Impulses all positive

$$F1(\omega) := 1 + e^{-j \cdot \omega \cdot T_c}$$

$$F2(\omega) := 1 + e^{j \cdot \omega \cdot T_c} + e^{-j \cdot \omega \cdot 2 \cdot T_c}$$

$$F3(\omega) := 1 + e^{j \cdot \omega \cdot T_c} + e^{-j \cdot \omega \cdot 2 \cdot T_c} + e^{-j \cdot \omega \cdot 3 \cdot T_c}$$

$$F4(\omega) := 1 + e^{j \cdot \omega \cdot T_c} + e^{-j \cdot \omega \cdot 2 \cdot T_c} + e^{-j \cdot \omega \cdot 3 \cdot T_c} + e^{-j \cdot \omega \cdot 4 \cdot T_c}$$



E.4 Fourier-Time Spectrum (varying ± 1)

One followed by a minus one

$$F1(\omega) := 1 - e^{j\omega \cdot T_c}$$

followed by a couple of plus 1's

$$F2(\omega) := 1 - e^{j\omega \cdot T_c} + e^{-j\omega \cdot 2 \cdot T_c} + e^{-j\omega \cdot 3 \cdot T_c} + e^{-j\omega \cdot 4 \cdot T_c} + e^{-j\omega \cdot 5 \cdot T_c}$$

followed by a couple of negative impulses

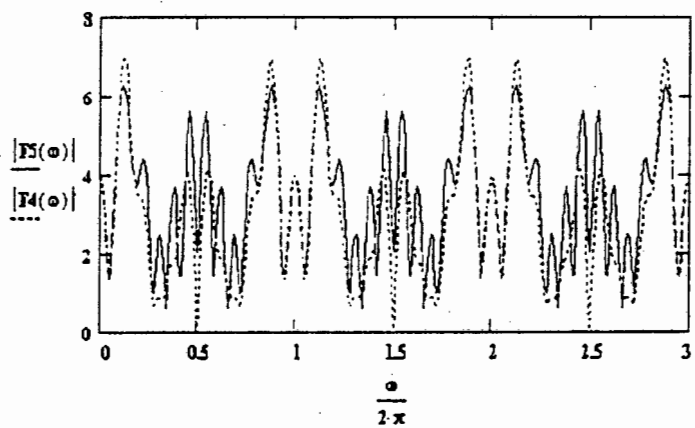
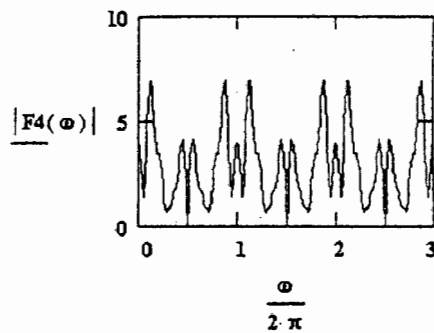
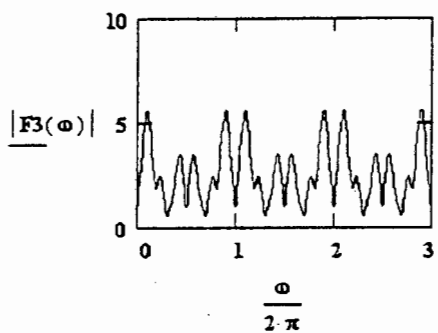
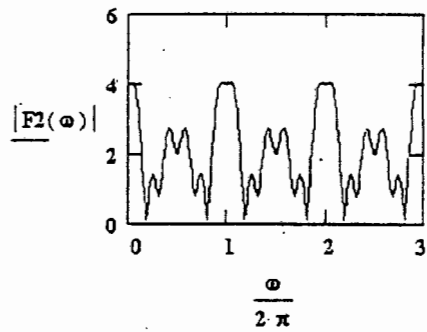
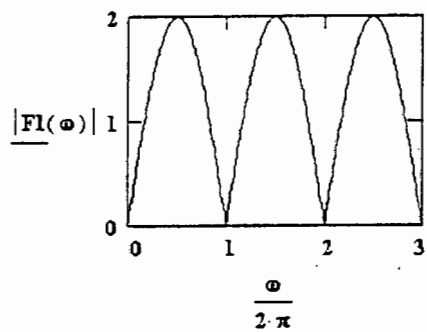
$$F3(\omega) := F2(\omega) - e^{-j\omega \cdot 6 \cdot T_c} - e^{-j\omega \cdot 7 \cdot T_c} - e^{-j\omega \cdot 8 \cdot T_c}$$

followed by a few positives again

$$F4(\omega) := F3(\omega) + e^{-j\omega \cdot 9 \cdot T_c} + e^{-j\omega \cdot 10 \cdot T_c} + e^{-j\omega \cdot 11 \cdot T_c}$$

followed by a neg and pos spike:

$$F5(\omega) := \left(F4(\omega) - e^{-j\omega \cdot 12 \cdot T_c} \right) + e^{-j\omega \cdot 13 \cdot T_c}$$



Appendix F

Details of Tesla

The following is an excerpt from the Tesla manual.

'We often draw block diagrams to represent electronic systems at the conceptual level. TESLA is a simulator which is a library of models which correspond to the blocks used to conceptualize electronic systems. The library contains models such as filters, VCOs, and mixers. It is particularly suited to communication systems. TESLA animates all of the blocks in a circuit in parallel, advancing time in time-steps. You can enter a block diagram using a schematic editor like OrCad. An optional library is available from TESOFTE to enable OrCad/SDT to generate TESLA circuit files. You can also create circuit files directly with a text editor.

The circuit file is basically a list of the model elements used, along with their node connections. The free format of the circuit file is similar to SPICE circuit files.

When you run Tesla, you are in an interactive environment from which you can edit files, run simulations, display results, and do spectrum analysis. TESLA calculates the spectra present at each node using an FFT.

Since TESLA does a direct time domain simulation of the behaviour in each block in parallel, there are no restrictions on the topology or characteristics of circuits represented. Nested, interlocking and leap-frog feed-back loops are handled with ease. Nonlinear and digital blocks coexist with linear functions such as filters. Instrument models let you make the same measurements you make in the lab.

Since the blocks are represented at the conceptual level, accuracy is inherent. TESLA shows

you what your concept is doing based on a high level behaviour notions which are built into the models. Of course, it is still necessary to ensure that an adequately small time-step is used to avoid parasitic delay effects. This is where Tesla stands out above nodal simulators such as SPICE. Due to the computational requirements of transistor level nodal analysis, it usually isn't practical to use circuit simulators to produce large numbers of time-steps required in communications simulation.'

Tesla is available from TeSoft, Inc. 205 Crossing Creek Ct., Roswell, GA 30076.

References

- [1] A.J. Viterbi. Spread Spectrum Communications -Myths and Realities. *IEEE Communications Magazine*, pages 11 - 18, May 1979.
- [2] A.J. Viterbi. When Not to Spread Spectrum - a Sequel. *IEEE Communications Magazine*, 23(4):12 - 17, April 1985.
- [3] A.J. Viterbi. Wireless Digital Communications: A View Based on Three Lessons Learned. *IEEE Communications Magazine*, pages 33-36, September 1991.
- [4] R.C. Dixon. *Spread Spectrum Systems*. John Wiley and Sons, 1976.
- [5] V. Shetty. Self-reliance is Crucial for Rural Comms Development. *Communications International*, pages 48-52, September 1992.
- [6] V. Shetty. The Famished Road to a Full Network. *Communications International*, pages 52-54, April 1994.
- [7] I. Scales. Virtual Telephone: A Solution to Line Scarcity. *Communications International*, pages 48-49, October 1992.
- [8] G. Zorpette. Technology in India. *IEEE Spectrum*, pages 25-53, March 1994.
- [9] J.G. van de Groenendaal and R.M. Braun. Rural Communications in Multi-path Channels: Spread Spectrum Applications. *IEEE SA Symposium on Communications and Signal Processing (COMSIG) 93*, September 1993.
- [10] R.C. Dixon. *Spread Spectrum Systems with Commercial Applications (Third Edition)*. John Wiley and Sons, 1994.
- [11] R.A. Scholtz. Origins of Spread Spectrum Communications. *IEEE Transactions on Communications*, 30(5):822-854, May 1982.

- [12] R. Price. Further Notes and Anecdotes on Spread Spectrum Origins. *IEEE Transactions on Communications*, 31:85-97, January 1983.
- [13] R.L. Pickholtz, D.L. Schilling, and L.B. Milstein. Theory of Spread Spectrum - A Tutorial. *IEEE Transactions on Communications*, 30(5):355-384, May 1982.
- [14] J.H. Lindholm. An Analysis of the Pseudo-Randomness Properties of Subsequences of Long m-Sequences. *IEEE Transactions on Information Theory*, 14(4):569 - 576, July 1968.
- [15] M.B. Pursley and D.V. Sarwate. Crosscorrelation Properties of Pseudorandom and Related Sequences. *Proceedings of the IEEE*, 68(5):593 - 619, May 1980.
- [16] J-S. No and P.V. Kumar. A new Family of Binary Pseudorandom Sequences Having Optimal Periodic Correlation Properties and Large Linear Span. *IEEE Transactions on Information Theory*, 35(2):371-379, March 1989.
- [17] R. Gold. Optimal Binary Sequences for Spread Spectrum Multiplexing. *IEEE Transactions on Information Theory*, IT-13:619-621, October 1967.
- [18] A.J. Viterbi. Implications of Mobile Cellular CDMA. *IEEE Communications Magazine*, pages 38-41, December 1992.
- [19] S. Chia. The Universal Mobile Telecommunications System. *IEEE Communications Magazine*, pages 54-62, December 1992.
- [20] J.K. Holmes. Acquisition Time Performance of PN Spread Spectrum Systems. *IEEE Transactions on Communications*, 25(8):778-784, August 1977.
- [21] G.F. Sage. Serial Synchronization of Pseudonoise Systems. *IEEE Transactions on Communication Technology*, December 1964.
- [22] R.B. Ward. Acquisition of Pseudonoise Signals by Sequential Estimation. *IEEE Transactions on Communication Technology*, pages 475-484, December 1965.
- [23] R.E. Ziemer and R.L. Peterson. *Digital Communications and Spread Spectrum Systems*. Macmillan Publishing Company, January 1985.
- [24] W.R. Braun. Performance Analysis for the Expanding Search PN Acquisition Algorithm. *IEEE Transactions on Communications*, 30(3):424-429, March 1982.

- [25] V.M. Jovanovic. Analysis of Strategies for Serial Search Spread Spectrum Code Acquisition - Direct Approach. *IEEE Transactions on Communications*, 36(11):1208-1220, November 1988.
- [26] P.W. Baier, K. Dostert, and M. Pandit. A Novel Spread Spectrum Receiver Synchronization Scheme Using SAW-Tapped Delay Line. *IEEE Transactions on Communications*, 30(5):1037-1047, May 1982.
- [27] M Shi. Performance of a Composite Code Sequence Acquisition Scheme in DSSS System Based on Two Cascade SAW Convolvers. *ISSSTA'92*, pages 15-7, November 1992.
- [28] Y.T. Su. Rapid Code Acquisition Algorithms Employing PN Matched filters. *IEEE Transactions on Communications*, 36(6):724-733, June 1988.
- [29] G.J.R. Povey and P.M. Grant. Simplified Matched Filter Receiver Designs for Spread Spectrum Communications Applications. *IEE Electronics and Communications Journal*, pages 59 - 64, April 1993.
- [30] F. Hongji and Y. Zhang. A New Method for Fast Acquisition of Pseudo-Random Code. *ISSSTA'92*, pages 15-6, November 1992.
- [31] D.V. Sarwate and K.K. Chwala. Parallel Acquisition of PN Sequences in DS/SS Systems. *IEEE Transactions on Communications*, 42(5):2155 - 2164, May 1994.
- [32] J.K. Holmes. *Coherent Spread Spectrum Systems*. John Wiley and Sons, January 1992.
- [33] J.J. Spilker and D.T. Magill. The Delay-Lock Discriminator - An Optimum Tracking Device. *Proceedings of the IRE*, 49:1403-1416, September 1961.
- [34] R. De Gaudenzi and M. Luise. Decision-Directed Coherent Delay-Lock Tracking Loop for DS-Spread-Spectrum Signals. *IEEE Transactions on Communications*, 39(5):758-765, May 1991.
- [35] H.P. Hartmann. Analysis of Dithering Loop for PN Code Tracking. *IEEE Transactions on Aerosp. Electron. Systems*, pages 2-9, January 1974.
- [36] P.M. Hopkins. Double Dither Loop for Pseudonoise Code Tracking. *IEEE Transactions on Aerosp. and Electron. Systems*, 13(6):644-650, November 1977.
- [37] R.B. Ward and K.P. Yiu. Acquisition of Pseudonoise Signals by Recursion-Aided Sequential Estimation. *IEEE Transactions on Communications*, 25(8):784-794, August 1977.

- [38] F.G. Stremler. *Introduction to Communication Systems (Third Edition)*. Addison-Wesley Publishing Company, 1990.
- [39] R.M. Braun. Novel Loop Structure Enables a Bit Synchronizer to Operate over Multi-Decade Data Rates. *IEEE SA Symposium on Communications and Signal Processing (COMSIG)*, pages 217-220, September 1992.
- [40] J.G. van de Groenendaal and M. Davidson. Novel Spread Spectrum Synchronisation Network. *Paper in Preparation*, January 1995.
- [41] M. Davidson and J.G. van de Groenendaal. A Novel Maximal-Length Sequence Synchronisation Network. *Master Thesis, University of Cape Town*, September 1995.
- [42] I.D. Huntley and R.M. Johnson. *Linear and Nonlinear Differential Equations*. Ellis Hopwood Ltd, January 1983.
- [43] P.V. O'Neil. *Advanced Engineering Mathematics*. Wadworth Inc, January 1986.
- [44] M.K. Simon and W.C. Lindsey. *Phase-Locked Loops & Their Applications*. IEEE PRESS, January 1978.
- [45] W.C. Lindsey and C.M. Chie. *Phase-Locked Loops*. IEEE PRESS, January 1986.
- [46] J.G. van de Groenendaal and R.M. Braun. Phase-Plane Analysis of Phase-Locked Loops used for Clock Recovery. *IEEE Symposium on Communications and Signal Processing (COMSIG)*, pages 99-104, October 1994.
- [47] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing*. Prentice-Hall, 1975.
- [48] F.M. Gardner. *Phaselock Techniques, 2nd edition*. John Wiley and Sons, 1979.
- [49] J.G. van de Groenendaal. Virtual Telephone Project. Technical Report 1993. *University of Cape Town Graduate School Report: Elec. Eng.*, March 1993.
- [50] J.G. van de Groenendaal. Virtual Telephone Network (VTN). In *GSM Symposium - Cape Technikon*, May 1994.
- [51] V. Uzunoglu and M.H. White. The Synchronous Oscillator: A Synchronisation and Tracking Network. *IEEE Journal on Solid-State Circuits*, 20(6):1214-1226, December 85.

- [52] J.G. van de Groenendaal, A.M. Davidson, and R.M. Braun. The Injection Locked Oscillator as a Synchronisation Network. *Second Review by IEE Journal on Electronics and Communications*, December 1994.
- [53] R.M. Braun. Injection-Lock Bandwidth Extensions by Control Modulation and Current Hump Slope Detection. *Electronics Letters*, 21(13):564-565, June 85.
- [54] A.M. Davidson. *Synchronous Oscillator*. University of Cape Town Undergraduate Thesis, November 1993. Supervisor: J.G. van de Groenendaal.
- [55] J.G. van de Groenendaal. Error Calculations for Spread Spectrum in Multi-Path Channels. *Internal Report*, June 94.