AD-A259 315

①

AFIT/GE/ENG/92D-20

DTIC
S ELECTE JAN 6 1993 D
C

ANALYSIS AND SIMULATION OF MODIFIED TANLOCK AND DELAY LOCK LOOPS FOR GPS RECEIVER DESIGN

THESIS

James A. Hird, Captain, USAF

AFIT/GE/ENG/92D-20

93-00127

Approved for public release; distribution unlimited

93 1 04 001

AFIT/GE/ENG/92D-20

ANALYSIS AND SIMULATION OF MODIFIED TANLOCK AND DELAY LOCK LOOPS FOR GPS RECEIVER DESIGN

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

James A. Hird, B.S.E.E.
Captain, USAF

December 1992

DTIC QUALITY INSPECTED 5

Approved for public release; distribution unlimited

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

Preface

The purpose of this thesis is to investigate the performance of two types of tracking loops, the code-tracking Delay Lock Loop (DLL) and the carrier-tracking Modified Tanlock Loop (MTLL), used in Global Positioning System (GPS) receivers. The ability of the GPS receiver to operate in noisy and dynamic environments is dependent primarily on the ability of these two loops to overcome corruption of the input signal due to noise and modulation due to movement. Expanding the operating environment of these loops, therefore, can significantly improve GPS receiver operation.

I would like to extend my sincerest thanks to the many people who helped me complete this research. First of all, I wish to thank my study partners, Brian, Chaz, Dennis, Jim and Ron, who helped me, immensely, to understand the material in the many classes we shared. I want to thank my advisors, Major Mark Mehalic, Captain Joe Sachinni, and Mr Bill Watson, for their patience and insight. Finally, I want to thank my loving wife, Yvette, for the constant support and encouragement she provided during the 19 month separation we endured while I completed my studies.

Table of Contents

Page

List of Figures vi

List of Symbols vii

List of Abbreviations ix

Abstract x

I. Introduction 1

1.1 Background 1
1.2 Problem Statement 2
1.3 Assumptions 3
1.4 Scope 3
1.5 Approach 3
1.6 Thesis Organization 4

II. Literature Review 5

2.1 Chapter Overview 5
2.2 The Delay Lock Loop 5
2.3 The Tanlock Loop 6
2.4 Chapter Summary 8

III. Delay Lock Loop Analysis 9

3.1 Chapter Overview 9
3.2 Loop Preview 9
3.3 Equation Development 12
3.4 Linear Loop Analysis 18
3.5 Loop Bandwidth 20
3.6 Delay Error Components 21

3.6.1 Delay Error Due to Modulation 21
3.6.2 Delay Error Due to Noise 22

3.7 Worst-Case Signal Dynamics 23
3.8 Chapter Summary 24

IV. Delay Lock Loop Simulation 26

- 4.1 Chapter Overview 26
- 4.2 The Delay Lock Loop System 26
- 4.3 The PN SOURCE Block 29
  - 4.3.1 Overview 29
  - 4.3.2 The RAM Method 29
    - 4.3.2.1 The VCC Block 32
    - 4.3.2.2 The ADDRESS COUNTER Block 34
  - 4.3.3 The Real Time Method 35
- 4.4 The INTEGRATOR Block 37
- 4.5 The DELAY DETECTOR Block 37
- 4.6 The LOOP FILTER Block 38
- 4.7 The INPUT SIGNAL Block 39
  - 4.7.1 The PROGRAMMABLE PN GENERATOR Block 40
  - 4.7.2 The NOISE MAKER Block 42
- 4.8 DLL Simulation Methodology and Results 42
- 4.9 Chapter Summary 45

V. Modified Tanlock Loop Analysis 46

- 5.1 Chapter Overview 46
- 5.2 Loop Preview 46
- 5.3 Equation Development 48
- 5.4 Analysis Without Noise 54
- 5.5 Phase Error Components 56
  - 5.5.1 Phase Error Due to Modulation 56
  - 5.5.2 Phase Error Due to Noise 56
- 5.6 Worst-Case Signal Dynamics 58
- 5.7 Chapter Summary 59

VI. Modified Tanlock Loop Simulation 60

- 6.1 Chapter Overview 60
- 6.2 The Modified Tanlock Loop System 60
- 6.3 The INTEGRATOR Block 62
- 6.4 The LOOP FILTER Block 62
- 6.5 The MTLL INPUT SIGNAL Block 63

6.6 MTLL Simulation Methodology and Results 63
6.7 Chapter Summary 66

VII. Conclusion and Recommendations 67

7.1 Summary 67
7.2 Conclusions/Lessons Learned 68

7.2.1 The Delay Lock Loop 68
7.2.2 The Modified Tanlock Loop 69

7.3 Recommendations for Further Research 70

Appendix A: Loop Errors Due To Modulation 72

A.1 Modulation Errors in the Delay Lock Loop 72

A.1.1 DLL Case A: Frequency Step 73
A.1.2 DLL Case B: Frequency Ramp 74

A.2 Modulation Errors in the Modified Tanlock Loop 76

A.2.1 MTLL Case A: Frequency Step 76
A.2.2 MTLL Case B: Frequency Ramp 77

A.3 Summary 78

Appendix B: Worst Case Signal Dynamics 80

B.1 Overview 80
B.2 Frequency Step 80
B.3 Frequency Ramp 81
B.4 Summary 82

Bibliography 83

Vita 85

List of Figures

Figure Page

1. Delay Lock Loop 10
2. Delay Error Detector Characteristic Curves 16
3. Normalized Delay Error Curves for the DLL 25
4. DLL Test Configuration 27
5. PN SOURCE Block 30
6. VCC Block 33
7. ADDRESS COUNTER Block 35
8. REAL-TIME PN SOURCE Block 36
9. INTEGRATOR Block 37
10. DELAY DETECTOR Block 38
11. LOOP FILTER Block 38
12. INPUT SIGNAL Block 39
13. PROGRAMMABLE PN GENERATOR Block 40
14. PROGRAMMABLE IMPULSE TRAIN Block 41
15. NOISE MAKER Block 42
16. DLL Simulator Results 44
17. Modified Tanlock Loop 47
18. Phase Error Detector Characteristic Curves 53
19. Normalized Phase Error Curves for the MTLL 59
20. MTLL Test Configuration 61
21. MTLL INPUT SIGNAL Block 63
22. MTLL Simulation Results 65

List of Symbols

Symbol		Introduced on Page
B_L	Two-sided loop bandwidth in Hz	14
$D_g(\tau_e/\Delta)$	Autocorrelation difference function, $R_{g+}(\tau_e/\Delta) - R_{g-}(\tau_e/\Delta)$	16
$F(p)$	Filter transfer function using the Heaviside operator	18
G_1, G_2	Loop filter control parameters	20
g	Acceleration due to gravity, $1\ g = 9.81\ m/s^2$	3
$g(t)$	Spreading pseudo-noise (PN) sequence	12
$H(p)$	Loop transfer function using the Heaviside operator	19
K_c	Voltage controlled clock gain	18
K_g, K_m	Loop mixer gains	13
K_0	Voltage controlled oscillator gain	52
M	Number of chips in one period of $g(t)$	12
$m(t)$	Message signal	12
$N_c(t)$, $N_s(t)$	Additive white Gaussian noise components of input noise	12
N_0	Input noise power spectral density (One-sided)	12
$n(t)$	Additive white Gaussian noise on the received signal	12
$n_e(t)$	Phase error noise	51
P	Input signal power	12
P_v	Power of VCO output signal	49
p	Heaviside operator, $p \triangle d/dt$	18
$R_{g+}(\tau_e/\Delta)$	Autocorrelation function of PN sequence offset by $+\Delta/2$	14
$R_{g-}(\tau_e/\Delta)$	Autocorrelation function of PN sequence offset by $-\Delta/2$	16
$S_g(\tau_e/\Delta)$	Autocorrelation summation function, $R_{g+}(\tau_e/\Delta) + R_{g-}(\tau_e/\Delta)$	16
$S_n(\omega)$	Noise power spectral density	19
s	Laplace Transform variable $s = p = j\omega$	20
$s(t)$	Loop input signal	12
T	Chip duration in seconds	12
W_L	Two-sided loop bandwidth in rads/sec	20
W_n	Two-sided receiver front end noise bandwidth in rads/sec	12
$\bar{x}(t)$	The filtered version of $x(t)$	14
$x'(t)$	Version of $x(t)$ rotated 90 degrees	51
$\Delta\omega$	Frequency step change to input signal	21
$\Delta\omega_v$	Offset between received signal and VCO output frequency	49
$\Delta/2$	Delay lock loop offset (normally equal to $T/2$)	13
$\Theta(t)$	Phase of received signal	12
$\Theta_e(t)$	Loop phase error, $\Theta(t) - \Theta_v(t)$	50
$\Theta_{em}(t)$	Phase error due to modulation	56
$\Theta_{en}(t)$	Phase error due to noise	56
$\Theta_v(t)$	Phase of VCO output signal	49
θ_0	Arbitrary phase of transmitted signal	12
ζ	Loop damping factor	20
σ_{en}^2	Noise error variance	22
τ	Transmission delay of satellite signal	12

$\hat{\tau}$	Transmission delay estimate	13
$\hat{\tau}/\Delta$	Normalized transmission delay estimate	18
τ_e	Transmission delay estimate error, $(\tau - \hat{\tau})$	13
$\hat{\tau}_e/\Delta$	Normalized transmission delay estimate error	24
τ_{em}	Delay error due to modulation	21
τ_{en}	Delay error due to noise	21
ω	Actual frequency of carrier at reception	12
ω_n	Loop natural frequency	21
ω_0	Nominal carrier frequency	12
$\dot{\omega}$	Linear frequency ramp change to input signal	22

List of Abbreviations

Abbreviation		Introduced on Page
AGC	Automatic Gain Control	7
AWGN	Additive White Gaussian Noise	3
BPSK	Binary Phase-Shift Keying	48
DLL	Delay Lock Loop	2
DPLL	Digital Phase Lock Loop	8
DS	Direct Sequence	1
GPS	Global Positioning System	1
MCTL	Modified Code Tracking Loop	6
MTLL	Modified Tanlock Loop	2
PLL	Phase Lock Loop	6
PMF	Probability Mass Function	7
PN	Pseudo-Noise	3
RF	Radio Frequency	5
SNR	Signal-to-Noise Ratio	4
SPW™	Signal Processing Workstation®	3
SS	Spread Spectrum	1
TLL	Tanlock Loop	7
VCC	Voltage Controlled Clock	11
VCO	Voltage Controlled Oscillator	6
BPF	Band-pass Filter	11
PSD	Power Spectral Density	42

AFIT/GE/ENG/9[illegible] 20

Abstract

The purpose of this thesis is to investigate two types of tracking loops used in Global Positioning System (GPS) receiver design. The first loop, the Delay Lock Loop (DLL), is a code tracking loop used to synchronize a locally generated pseudo-noise (PN) sequence with the PN sequence in the GPS satellite broadcast. Synchronization of the PN sequences is essential for de-spreading the direct-sequence spread spectrum (DS/SS) broadcast and demodulating the transmitted data. The second loop, the Modified Tanlock Loop (MTLL), is a carrier tracking loop used to synchronize the phase of a voltage controlled oscillator (VCO) with the carrier of the GPS satellite broadcast. Carrier synchronization is essential for optimum data demodulation. This thesis derives equations predicting the theoretical performance of each loops' ability to track a GPS signal corrupted by noise and signal dynamics arising from transmitter and/or receiver motion. In addition, computer simulations of the DLL and MTLL were developed and the results are presented. The simulations display phenomena which were not present in the theoretical predictions.

ANALYSIS AND SIMULATION OF MODIFIED TANLOCK AND DELAY LOCK LOOPS FOR GPS RECEIVER DESIGN

I. INTRODUCTION

1.1 BACKGROUND

The Global Positioning System (GPS) is a navigational aid that uses signals from satellites to enable receivers to determine their position and velocity. GPS has introduced a new era in world-wide navigation accuracy. The ability to quickly and accurately fix position and velocity, anywhere in the world, with unprecedented precision, promotes a multitude of applications in geodesic survey, commercial transportation, consumer products, and military operations.

The GPS receiver determines its position by simultaneously measuring the transmission delays of signals of at least four satellites from a total constellation of 22 satellites. The distance between the receiver and the satellites can be calculated from these transmission delays. Navigation and position information are triangulated based on the known positions of the satellites.

The ability of the receiver to continuously determine the propagation delay of the satellite signal depends on a local oscillator in the receiver being phase-synchronous with the arriving satellite signal carrier. Because the satellite transmission is a direct-sequence spread spectrum (DS/SS) signal, the receiver must also be phase synchronous with the DS spreading code. These two requirements are normally met and maintained by two special circuits in the receiver: a carrier tracking loop and a code tracking loop. These loops must be able to maintain synchronization with the received signal despite corruption of the received signal by noise and dynamics. An engineering design trade-off exists in optimizing loop performance in noisy, dynamic environments. To reduce the effects of noise on the signal, the loop bandwidth should be made as small as possible. The narrower the bandwidth, the less noise entering the system. However, if the bandwidth is made too narrow, signal dynamics (such as Doppler caused by transmitter and/or receiver movement) will cause the received signal to move outside the loop bandwidth and be attenuated. The wider the loop bandwidth, the greater the signal dynamics the loop can track. The amount of noise and dynamics a GPS receiver will encounter depends on the type of application for which it will be used. GPS receivers on high performance jet fighter aircraft can expect to encounter a worst-case environment of high dynamics due to aircraft motion and high noise in the form of jamming. The wide variety of GPS applications and environments complicate optimizing GPS receiver design. Building prototypes to test receiver configurations is expensive and time consuming. Models and simulations can enhance design decisions without the expense and time required to build prototypes.

1.2 PROBLEM STATEMENT

Models and simulations of GPS receiver designs must be developed if continued advances in GPS hardware are to continue. This thesis investigates one type of code tracking loop, a delay lock loop (DLL), and one type of carrier tracking loop, a modified Tanlock loop (MTLL). In each case, theoretical loop performance will be calculated and simulations run to compare simulated and theoretical performance. Specific performance characteristics measured will include minimum allowable signal-to-noise ratio, maximum allowable frequency step, maximum allowable frequency ramp, and probability of losing lock.

1.3 ASSUMPTIONS

Signal acquisition is assumed to have been accomplished prior to tracking loop initiation. An undefined system processor is assumed to have initiated the proper pseudo-noise (PN) code within the PN generator and an undefined acquisition method is assumed to have determined the code position to within 1/2 chip. Received signal dynamics are assumed to result from a worst-case scenario of the GPS receiver being located aboard a jet aircraft traveling at 700 m/s (approx. Mach 2.1 at sea level) and executing a nine-g turn.

1.4 SCOPE

The scope of this thesis is limited to the analysis and simulation of the performance of a single type of MTLL and DLL in additive, white, Gaussian noise (AWGN). No measurement of the accuracy of the receiver position or velocity derivatives is attempted. No attempt is made to determine the tracking loops' immunity to narrowband interference. No consideration is given to the cost of a hardware implementation of the designs discussed.

1.5 APPROACH

This thesis will first derive equations governing the theoretical performance of the DLL and the MTLL. Graphs will be presented to illustrate the loops' predicted performance as key parameters are varied. Simulations will then be introduced, using the Signal Processing Workstation® (SPW™) developed by Comdisco® Systems, Inc of Foster City, California. The theoretical and simulated performance will then be compared. Possible reasons for differences between theoretical and simulated performance will be discussed. Finally, conclusions and recommendations for further research will be presented.

1.6 THESIS ORGANIZATION

Chapter II presents a literature review of previous efforts in DLL and MTLL design and analysis. Because of Air Force interest in high dynamic environments, designs concentrating on extending the linear region of operation (and hence extending the range of allowable dynamics and noise) are emphasized. Chapter III is a detailed derivation of equations governing the theoretical performance of the DLL. Particular parameters of interest are the probability of losing lock, the minimum required signal-to-noise ratio (SNR), and performance under worst-case signal dynamics. Chapter IV discusses the method and results of the SPW™ simulations of the DLL and compares the simulated results to the theoretical expectations derived in Chapter III. Similarly, Chapter V is a detailed derivation of equations governing the performance of the TLL followed by simulation results in Chapter VI. Chapter VII summarizes the findings of this thesis and discusses possible causes for discrepancies between theoretical expectations and simulated results. Finally, Chapter VII finishes with conclusions and recommendations for further research.

II. LITERATURE REVIEW

2.1 CHAPTER OVERVIEW

This chapter will present a brief summary of previous efforts in spread-spectrum tracking loop design. Section 2.2 will provide a historical background on the development of the code-tracking delay lock loop (DLL). Section 2.3 will present a historical background on the development of the carrier-tracking modified Tanlock loop (MTLL).

2.2 THE DELAY LOCK LOOP

The delay lock loop (DLL) is a device for tracking the delay difference between two correlated waveforms. The DLL was first investigated by Spilker and Magill in 1961 [SPI61]. In 1963, Spilker analyzed the performance of a DLL in tracking binary signals generated by n-stage feedback shift registers [SPI63]. The interest in these types of sequences stems from their ability to be easily regenerated with any desired delay and their two-level autocorrelation functions. That is, within the period of the sequence, only one autocorrelation peak occurs, thus allowing unambiguous delay determination up to a maximum delay equal to the period of the sequence. Given a long enough sequence period, unambiguous delay determination is possible.

In 1966, Gill investigated DLL performance for various radio frequency (RF) implementations [GIL66]. Gill discussed two forms of modulation and three forms of demodulation. The two forms of modulation were amplitude and bi-phase modulation. The three demodulation techniques considered were envelope correlation, phase-coherent correlation, and phase-lock demodulation followed by video correlation. Many subsequent papers analyzed the performance and proposed modifications to these implementations, particularly the easily implemented envelope correlator. A particular disadvantage to each of these implementations is their sensitivity to gain imbalance in the loop arms. Many implementations also introduce additional noise and excessive jitter due to a squaring function in the loop arms.

Simon performed extensive linear and nonlinear analysis of the performance of DLL's and demonstrated an optimum filter bandwidth could be determined to minimize loop tracking jitter. His 1977 paper presented numerical results for several types of filters and compared the DLL performance to its time-sharing counterpart, the tau-dither loop [SIM77].

In 1980, Yost and Boyd proposed a modified code tracking loop (MCTL) to address the issue of loop arm gain imbalance [YOS80] [YOS82]. The MCTL was shown to have the hardware simplicity of the tau-dither loop and a tracking performance superior to the traditional DLL. Squaring loss and steady-state jitter, however, were still problems. Digital implementations have also been proposed which eliminate the effect of arm imbalance and reduce steady-state jitter [GAU91].

2.3 THE TANLOCK LOOP

The phase lock loop (PLL), as a method of synchronizing an oscillator's signal to a reference signal, has found extensive applications in communications systems constrained by limited transmitter power. Costas and Squaring loops were developed to eliminate the effects of reversals due to data modulation on the error detector signal [GAR79]. In both loops, the error signal is proportional to the sine of twice the phase error between the arriving (reference) signal and the output of a local voltage controlled oscillator (VCO). Using the small angle approximation, the detector output is linear with respect to the phase error up to about 30° [GAR79]. As the phase error increases beyond 30°, loop performance degrades until loss of lock is assured when the phase error exceeds 90°.

Robinson discussed a modification in 1962 that replaced the mixer phase detector of the Costas loop with a detector that calculated a function similar to the tangent of the ratio of the loop arms [ROB62]. Called the Tanlock loop (TLL), this modification extends the linear range of the detector output, allowing the TLL to track signals with greater dynamics and lower signal-to-noise ratios (SNRs). The tangent detector maintains linear operation with phase errors up to approximately 157° [ROB62]. Balodis showed a 2 to 6 dB improvement in reception threshold is possible with the TLL over the conventional PLL [BAL64]. In addition, the TLL enables faster acquisition and a wider acquisition range.

Continuing the quest for larger linear operating ranges, Lee and Un introduced a digital modified TLL (MTLL) design in 1982 [LEE82]. The main feature of the MTLL is the arctangent phase error detector which replaces the TLL tangent function detector. The arctangent detector is exactly linear with respect to the phase error with a period of 2π. The locking range is thereby extended to ±180°. Additional attractive features include insensitivity to input signal power, thus eliminating the need for automatic gain control (AGC) circuitry, increased noise immunity, and the ability to analyze the loop with a relatively simple linear difference equation instead of linear approximations, as had been done with previous loop designs. Cho and Un showed that by increasing the sampling rate, the digital MTLL locking range can be extended even further [CHO87].

In 1988, Pomalaza-Raez measured the phase error probability mass function (PMF) and mean time to slip cycle versus input SNR performance of the MTLL in noisy environments. Through analysis and simulation, he demonstrated the digital MTLL performance for low to moderate input SNR is only slightly improved over the digital PLL (DPLL). The MTLL, however, has a larger linear characteristic then the DPLL, making it more attractive for applications that require an increased tracking range [POM88].

2.4 CHAPTER SUMMARY

The historical developments leading to the modern DLL and MTLL have been presented. The primary push in the development of these two subsystems has been to improve performance in low signal-to-noise ratio (SNR) environments without sacrificing dynamic signal tracking. With this self-contradictory goal in mind, the theoretical performance of the DLL and MTLL in a GPS application will now be derived.

III. DELAY LOCK LOOP ANALYSIS

3.1 CHAPTER OVERVIEW

This chapter will present a detailed derivation of the theoretical performance of the delay lock loop (DLL). Section 3.2 will provide a quick overview of the loop components and operations, introducing concepts, terminology, and notation. Section 3.3 will begin the detailed development of equations governing the loop's performance. An analysis of the loop limited to operating in its linear region is presented in Section 3.4. The resulting loop bandwidth will be derived in Section 3.5. In Section 3.6, the loop delay error will be derived and values resulting from the two dominant sources, modulation and noise, will be predicted. In Section 3.7, the worst-case signal dynamics (derived in Appendix B) will be used to predict loop performance for various values of SNR and the loop parameters, K_c and G_1. Section 3.8 will summarize the findings of this chapter.

3.2 LOOP PREVIEW

Each satellite in the GPS constellation repeatedly broadcasts a unique pseudo-noise (PN) sequence precisely aligned with a universal time standard. A delay lock loop (DLL) in the GPS receiver is responsible for synchronizing a locally generated replica of the PN sequence to the received GPS satellite broadcast. This synchronicity allows precise measurement of the delay between when a chip in the PN sequence (analogous to a bit in a data sequence) was transmitted and when it was received. The propagation delay can then be converted to a range measurement between the satellite and the receiver. The receiver maintains a record of the position of each satellite in the GPS constellation. Theoretically, simultaneous range measurements to three satellites would allow the receiver to determine its three-dimensional position and velocity through triangulation. In practice, four range measurements are made to eliminate inaccuracies in the receiver's clock. In addition, multiplying the precisely aligned locally generated sequence with the received signal removes the PN sequence from the received signal. This multiplication despreads the spectrum of the received signal and allows data riding on the signal to be recovered, even in noisy environments.

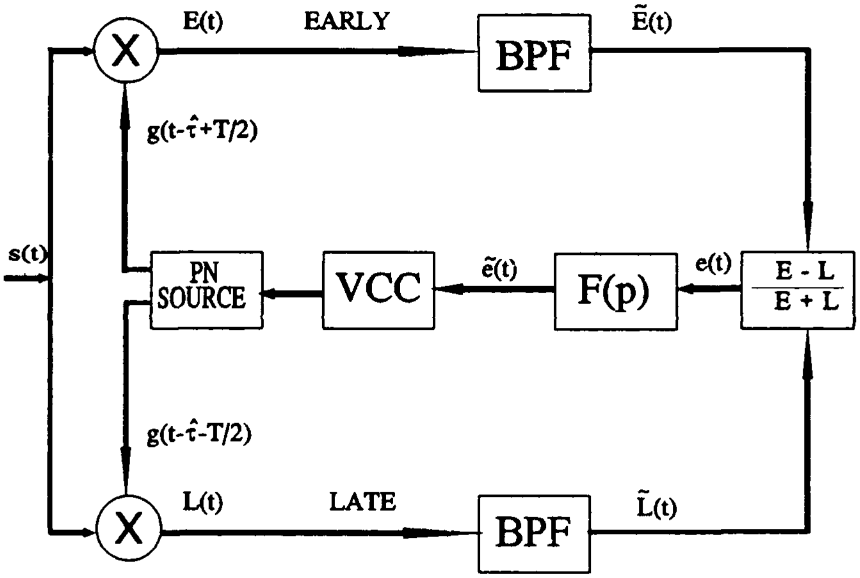


Figure 1 Delay Lock Loop

A block diagram of the DLL is shown in Figure 1. The received signal, s(t), is first mixed with early and late replicas of the original spreading PN sequence, g(t). The "Early" replica, $g(t-\hat{\tau}+\Delta/2)$, is advanced $\Delta/2 = T/2$ seconds, where T is the duration of a single PN bit (called a chip to prevent confusion with data bits). The "Late" replica is delayed by $\Delta/2$.

The $\hat{\tau}$ indicates the loop's approximation of the propagation delay, τ. The Early and Late signals are then band-pass filtered (BPF) and passed to the delay error detector. The output of the detector is the normalized difference between these Late and Early signals. If there is more energy in the Early signal than in the Late signal (meaning the received signal more closely correlated with the advanced replica), the output of the delay error detector will be a positive voltage. Conversely, if there is more energy in the Late signal, the delay error detector output will be negative. Thus, the detector output is proportional to the delay error, assuming the error is small. It will be shown that to ensure the detector remains in a linear operating region, the delay error must be limited to approximately half the duration of a single chip. The ratio used in the detector is a variation on the more common difference detector. Normalizing the error by dividing by the sum enhances the loops performance in the presence of noise [OUL84].

The error signal out of the detector is next passed through the loop filter, F(p), and onto the voltage controlled clock (VCC). The output clocking rate of the VCC is controlled by the input signal in a manner similar to a voltage controlled oscillator (VCO). A positive voltage will increase the clock rate, while a negative voltage will decrease the clock rate. Therefore, if the locally generated PN sequence leads the received sequence, the received sequence will more closely correlate with the delayed replica, a negative signal will be output by the delay error detector and the VCC will slow down the PN source. Similarly, if the local PN sequence lags the received signal, the received and advanced sequences will correlate, a positive voltage will be output, and the VCC will speed up the PN source. In this manner, the delay error between the two signals is brought to approximately zero.

3.3 EQUATION DEVELOPMENT

Let s(t) be the input to the DLL consisting of the desired signal and additive white Gaussian noise (AWGN), n(t). The input signal can then be expressed as:

$$s(t) = \sqrt{P}\, g(t-\tau)m(t-\tau)\cos[\omega_0 t + \Theta(t)] + n(t) \tag{1}$$

where: P is the power in the input signal

τ is the transmission or propagation delay of the signal

g(t-τ) is a repeating pseudo-noise (PN) sequence with values $\pm$ 1, M chips in length, each chip of T sec duration. The period of g(t-τ) is MT.

m(t-τ) is a message signal with values $\pm$ 1, edge synchronized with g(t-τ).

ω_0 is the nominal carrier frequency

$\Theta(t) \triangleq \Delta\omega t + \theta_0$ is the channel rotation resulting from dynamics such as Doppler

$\Delta\omega \triangleq [\omega - \omega_0]$ is the offset between the nominal carrier frequency, ω_0, and the actual frequency of s(t), ω.

n(t) is AWGN with the following bandpass representation:

$$n(t) = N_c(t)\cos[\omega_0 t + \Theta(t)] - N_s(t)\sin[\omega_0 t + \Theta(t)] \tag{2}$$

where $N_c(t)$ and $N_s(t)$ are independent, stationary, low-pass, white Gaussian noise processes with one-sided noise spectral density N_0 watt-sec/rad and one-sided bandwidth $W_n/2 << \omega_0$.

In the top (or Early) arm of the DLL, the multiplier mixes s(t) with a locally generated version of the spreading PN sequence, g(t). This locally generated sequence is expressed as $g(t-\hat{\tau}+\Delta/2)$ where $\hat{\tau}$ represents the loop's estimate of the actual propagation delay, τ. The $\Delta/2$ term is an intentionally added offset of half a chip duration or T/2. Multiplying Equation (1) by $g(t-\hat{\tau}+\Delta/2)$ yields the output of the first mixer:

$$
\begin{aligned}
E(t) = {} & \sqrt{P}\,K_m\, m(t-\tau)\overline{g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)}\cos[\omega_0 t + \Theta(t)] \\
& + \left\{\left[g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right) - \overline{g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)}\right]\right. \\
& \qquad \left.\bullet\ \sqrt{P}\,K_m\, m(t-\tau)\cos[\omega_0 t + \Theta(t)]\right\} \\
& + K_m\, g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)n(t)
\end{aligned}
\tag{3}
$$

with K_m denoting the mixer gain and the overbar representing statistical expectation or mean. In Equation (3), the product $g(t-\tau)g(t-\hat{\tau}+\Delta/2)$ has been broken into two components. The first component (the overbar term) is the mean and is time invariant. The second component (Line 2 of Equation (3)) is the remainder after the mean has been subtracted out. This component, called the self-noise term, is time varying. We define a new term $\tau_e \triangle \tau - \hat{\tau}$, the delay estimate error between the received PN sequence and the locally generated PN sequence.

Note the overbar (or mean) term in Equation (3) is the auto-correlation function of the PN sequence, offset by $+\Delta/2$, defined as [SIM77]:

$$R_{g^{+}}(\tau_e/\Delta) = \begin{cases} -\dfrac{1}{M}, & -M \le \dfrac{\tau_e}{\Delta} \le -\dfrac{3}{2} \\ 1 + \left(1+\dfrac{1}{M}\right)\left(\dfrac{\tau_e}{\Delta} + \dfrac{1}{2}\right), & -\dfrac{3}{2} \le \dfrac{\tau_e}{\Delta} \le -\dfrac{1}{2} \\ 1 - \left(1+\dfrac{1}{M}\right)\left(\dfrac{\tau_e}{\Delta} + \dfrac{1}{2}\right), & -\dfrac{1}{2} \le \dfrac{\tau_e}{\Delta} \le \dfrac{1}{2} \\ -\dfrac{1}{M}, & \dfrac{1}{2} \le \dfrac{\tau_e}{\Delta} \le M \end{cases} \tag{4}$$

where M is the number of chips in the PN sequence and MT is the period of the PN sequence. See Figure 2a.

Spilker demonstrated that if the single-sided loop bandwidth $B_L/2$ (in Hz) is much less than the PN sequence chip rate, 1/T (as it is in this case), then the self-noise term can be neglected [SPI63]. Neglecting the self-noise term and passing E(t) through the low-pass filter, yields:

$$\begin{aligned} \tilde{E}(t) &= \sqrt{P}K_m \tilde{m}(t-\tau)R_{g^{+}}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] \\ &\quad + K_m \tilde{g}\left(t-\hat{\tau}+\frac{\Delta}{2}\right)\tilde{n}(t) \end{aligned} \tag{5}$$

where the tilde represents a signal that has been filtered. Notice that at this point, the message signal power has been despread and the power in the bandlimited noise signal has been spread across the bandwidth of g(t) and subsequently reduced through filtering. The noise spreading effect is negligible for wideband AWGN but significant for narrow-band interference or jamming. This effect is responsible for significant research in spread spectrum communication systems.

A similar analysis shows the signal entering the phase detector from the Late arm is:

$$\begin{aligned}\tilde{L}(t) &= \sqrt{P}K_m\,\tilde{m}(t-\tau)R_{g^-}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] \\ &\quad + K_m\,\tilde{g}\left(t-\hat{\tau}-\frac{\Delta}{2}\right)\tilde{n}(t)\end{aligned} \tag{6}$$

The PN autocorrelation function, offset by $-\Delta/2$, is defined as [SIM77]:

$$R_{g^-}(\tau_e/\Delta) = \begin{cases} -\dfrac{1}{M}, & -M \le \dfrac{\tau_e}{\Delta} \le -\dfrac{1}{2} \\ 1 + \left(1+\dfrac{1}{M}\right)\left(\dfrac{\tau_e}{\Delta}-\dfrac{1}{2}\right), & -\dfrac{1}{2} \le \dfrac{\tau_e}{\Delta} \le \dfrac{1}{2} \\ 1 - \left(1+\dfrac{1}{M}\right)\left(\dfrac{\tau_e}{\Delta}-\dfrac{1}{2}\right), & \dfrac{1}{2} \le \dfrac{\tau_e}{\Delta} \le \dfrac{3}{2} \\ -\dfrac{1}{M}, & \dfrac{3}{2} \le \dfrac{\tau_e}{\Delta} \le M \end{cases} \tag{7}$$

Note $R_{g^-}(\tau_e) = R_{g^+}(\tau_e - T)$. Figure 2a graphs the two offset autocorrelation functions.

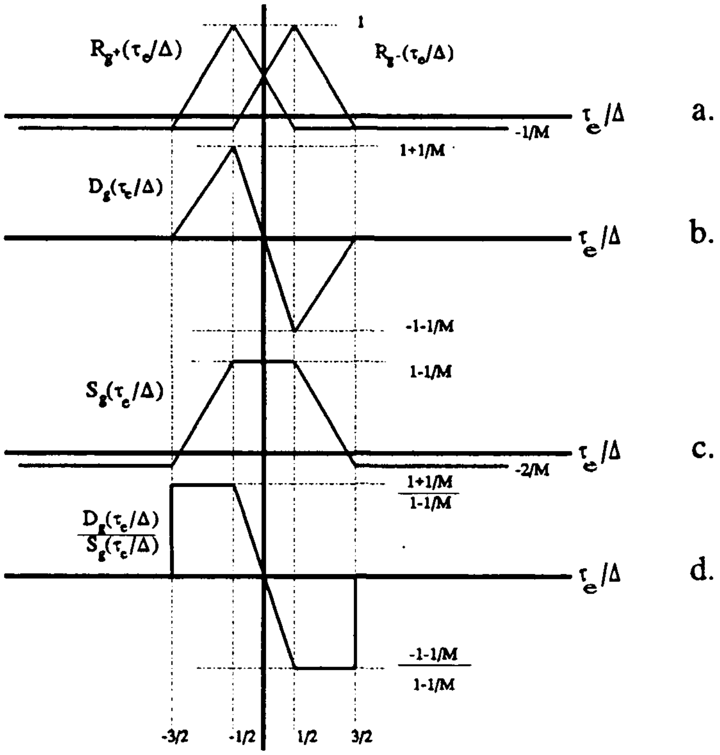


Figure 2 Delay Error Detector Characteristic Curves

The output of the delay error detector, e(t), can be written as:

$$e(t) = \frac{\tilde{E} - \tilde{L}}{\tilde{E} + \tilde{L}} = \frac{A(t)D_g(\tau_e/\Delta) + n_-(t)}{A(t)S_g(\tau_e/\Delta) + n_+(t)} \qquad (8)$$

where: $A(t) = \sqrt{P}K_m\tilde{m}(t-\tau)\cos[\omega_0 t + \theta(t)]$

$D_g(\tau_e/\Delta) = R_{g+}(\tau_e/\Delta) - R_{g-}(\tau_e/\Delta)$

$S_g(\tau_e/\Delta) = R_{g+}(\tau_e/\Delta) + R_{g-}(\tau_e/\Delta)$

$n_\pm(t) = K_m\tilde{n}(t)\,[\tilde{g}(t - \hat{\tau} + \Delta/2) \pm \tilde{g}(t - \hat{\tau} - \Delta/2)]$

Manipulating Equations (4) and (7) yields the autocorrelation difference function, $D_g(\tau_e/\Delta)$:

$$D_g(\tau_e/\Delta) = \begin{cases} 0, & -M \le \frac{\tau_e}{\Delta} \le -\frac{3}{2} \\ \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{3}{2}\right), & -\frac{3}{2} \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ \left(1+\frac{1}{M}\right)\frac{2\tau_e}{\Delta}, & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{3}{2}\right), & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{3}{2} \\ 0, & \frac{3}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \tag{9}$$

$D_g(\tau_e/\Delta)$ is graphed in Figure 2b. Similar manipulation yields the autocorrelation summation function, $S_g(\tau_e/\Delta)$:

$$S_g(\tau_e/\Delta) = \begin{cases} -\frac{2}{M}, & -M \le \frac{\tau_e}{\Delta} \le -\frac{3}{2} \\ 2 + \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{1}{2}\right), & -\frac{3}{2} \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ \left(1-\frac{1}{M}\right), & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ 2 - \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{1}{2}\right), & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{3}{2} \\ -\frac{2}{M}, & \frac{3}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \tag{10}$$

$S_g(\tau_e/\Delta)$ is graphed in Figure 2c. Note that in the absence of noise, the detector output, e(t), simplifies to the ratio $D_g(\tau_e/\Delta)/S_g(\tau_e/\Delta)$, which is defined as the discriminator characteristic. Increasing the linearity of this discriminator characteristic curve increases the dynamic range of the DLL. See Figure 2d.

The output of the voltage controlled clock (VCC) is $\hat{\tau}/\Delta$, the normalized propagation delay estimate [SIM77]. This observation, along with the fact the VCC acts as an integrator on its input, allows us to write the stochastic integro-differential equation describing the loop operation:

$$\begin{aligned} \frac{\hat{\tau}}{\Delta} &= \int K_c \bar{e}(t)\, dt \\ &= \frac{K_c F(p)\, e(t)}{p} \end{aligned} \tag{11}$$

where K_c is the voltage controlled clock gain and F(p) is the transfer function of the loop filter using the Heaviside operator, $p \triangleq d/dt$. Using this notation, $\bar{e}(t) \triangleq F(p)e(t)$ and integration is equivalent to multiplication by 1/p.

3.4 LINEAR LOOP ANALYSIS

Restricting $|\tau_e| \le \Delta/2$ causes the detector output to remain in the linear region (See Figure 2d). With this restriction, substituting Equations (8), (9), and (10), into Equation (11) yields:

$$\frac{\hat{\tau}}{\Delta} = \frac{K_c F(p)}{p} \left[\frac{A(t)\left(1 + \frac{1}{M}\right)\frac{2\tau_e}{\Delta} + n_-(t)}{A(t)\left(1 - \frac{1}{M}\right) + n_+(t)} \right] \tag{12}$$

For a long PN sequence, M>>1. Therefore, 1/M≈0. Recalling $\tau_e = \tau - \hat{\tau}$, letting 1/M = 0, and disregarding the noise term for the moment, Equation (12) can be rewritten as:

$$H(p) = \frac{\hat{\tau}}{\tau} = \frac{2K_c F(p)}{p + 2K_c F(p)} \tag{13}$$

where H(p) is defined as the linearized loop transfer function. Substituting H(p) defined in Equation (13) back into Equation (12) and continuing the assumption that 1/M = 0, Equation (12) becomes [GIL66]:

$$\frac{\hat{\tau}}{\Delta} = H(p)\left[\frac{A(t)\frac{\tau}{\Delta} + n_{-}(t)}{A(t) + n_{+}(t)}\right] \tag{14}$$

which is the linearized loop equation in the presence of noise. Because the input noise, n(t), was assumed white, the noise component of the output of the delay error detector, $n_{\pm}(t)$, (defined after Equation (8)), will also be white [COO86]. The power spectral density (PSD) of this component will then be:

$$\begin{aligned} S_n(\omega) &= 2\left(\frac{M+1}{M}\right)\frac{N_0}{2} \\ &= N_0 \qquad \textit{for large } M \end{aligned} \tag{15}$$

where 2[(M+1)/M] is the average power in the signal $[g(t-\hat{\tau}-\Delta/2) \pm g(t-\hat{\tau}+\Delta/2)]$ [LIN73].

3.5 LOOP BANDWIDTH

The equivalent two-sided bandwidth of the loop, W_L, is found by [LIN72:136]:

$$W_L = \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega \tag{16}$$

Substituting $F(p) = (G_1+G_2/p)$ for a proportional plus integral loop filter into Equation (13), H(p) becomes:

$$H(p) = \frac{2K_cG_1p + 2K_cG_2}{p^2 + 2K_cG_1p + 2K_cG_2} \tag{17}$$

Using the Laplace Transform variable $s = j\omega = p$, Equation (17) can be rewritten in the standard transfer function form for a loop feedback system employing an active first-order loop filter, as described in [GAR79:11]:

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{18}$$

where:

$\zeta = G_1\{K_c/(2G_2)\}^{1/2}$ is the loop damping factor normally set to the critical damping value of .707

$\omega_n = (2G_2K_c)^{1/2}$ is the natural frequency of the loop in rad/sec.

Substituting Equation (18) into Equation (16) yields the loop bandwidth [COO86]:

$$\begin{aligned} W_L &= \int_{-\infty}^{\infty} \left| \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right|^2 ds \\ &= \omega_n \left(\zeta + \frac{1}{4\zeta} \right) \\ &= K_c G_1 + \frac{G_2}{2G_1} \end{aligned} \tag{19}$$

3.6 *DELAY ERROR COMPONENTS*

The loop delay error, τ_e, arises from two main causes and so can be divided into two components. The first component, τ_{em}, is the error due to modulation, resulting from dynamics such as Doppler. The second component, τ_{en}, is the error due to noise. The total delay error is then [MAR62]:

$$\tau_e = \tau_{em} + \tau_{en} \tag{20}$$

3.6.1 DELAY ERROR DUE TO MODULATION. The peak delay error due to modulation is derived in Appendix A and the main results are summarized here:

a. For a frequency step input of magnitude $\Delta\omega$ to the frequency ω:

Peak $\tau_{em} = 0.322\Delta\omega/(\omega K_c G_1)$ seconds at time $t = \pi/(4K_c G_1)$

b. For a frequency ramp increasing linearly with time at a rate $\dot{\omega}$ rad/sec^2 to a frequency ω:

Peak $\tau_{em} = 0.5\dot{\omega}/[\omega(K_c G_1)^2]$ seconds at time $t = 0$

3.6.2 DELAY ERROR DUE TO NOISE. Recalling Equation (15), the variance of the random delay error due to the noise, τ_{en}, is [GIL66]:

$$\begin{aligned}\left(\frac{\sigma_{en}}{\Delta}\right)^2 &= \frac{1}{P}\int_{-\infty}^{\infty} S_n(\omega)|H(\omega)|^2 d\omega \\ &= \frac{N_0 W_L}{P} \\ &= \frac{N_0}{P}\left(K_c G_1 + \frac{G_2}{2G_1}\right)\end{aligned} \tag{21}$$

In the absence of errors due to modulation, if $|\tau_{en}| \leq \Delta/2$, the loop will remain locked. The probability of losing lock is [COO86:351]:

$$\begin{aligned}P\left[\tau_{en} > \frac{\Delta}{2}\right] &= 2Q\left[\frac{\Delta}{2\sigma_{en}}\right] \\ &= 2Q\left[\frac{1}{2}\sqrt{\frac{P}{N_0 W_L}}\right]\end{aligned} \tag{22}$$

Note the quantity inside the square root operator is the signal-to-noise ratio (SNR) of the loop signal.

The loop will remain in lock as long as $|\tau_e| < \Delta/2$. Combining the errors due to modulation and noise results in:

$$
\begin{aligned}
\frac{T}{2} &\geq \tau_{em} + \sigma_{en} \\
&\geq \frac{0.322\,\Delta\omega}{\omega K_c G_1} + \frac{0.5\dot{\omega}}{\omega (K_c G_1)^2} + T\sqrt{\frac{N_0}{P}\left(K_c G_1 + \frac{G_2}{2G_1}\right)}
\end{aligned}
\tag{23}
$$

Equation (23) is a fundamental expression defining loop performance and depicts the classic trade-off in receiver design. As K_c, G_1, and G_2 increase, delay errors due to modulation decrease while the delay error due to noise increases. By identifying worst-case signal dynamics and noise, optimum loop parameters K_c, G_1 and G_2, can be determined.

3.7 WORST-CASE SIGNAL DYNAMICS

Expected worst-case values for the signal dynamics this receiver will encounter are derived in Appendix B and summarized here:

Worst-case frequency step: $\Delta\omega = 44{,}000$ rad/sec

Worst-case frequency ramp: $\dot{\omega} = 1760$ rad/sec^2

Substituting the above values, $T = 9.78 \times 10^{-7}$ (for a PN chipping rate of 1.023 MHz), and $\omega = 2\pi \bullet 1.5$ GHz (the nominal carrier frequency of GPS) into Equation (23) and ignoring the noise term for the moment yields:

$$4.89 \cdot 10^{-7} \geq \frac{1.5 \cdot 10^{-6}}{K_c G_1} + \frac{9.34 \cdot 10^{-8}}{(K_c G_1)^2} \tag{24}$$

Algebraic manipulation shows K_cG_1 must be greater than 3.2 to satisfy Equation (24), for worst-case signal dynamics and no noise.

Because of the variety of environments the GPS receiver can operate in, including jamming situations, no practical worst-case noise value can be assumed. Figure 3 graphs Equation (23), the delay error normalized with respect to the maximum allowable delay error T/2, versus K_cG_1 for worst-case dynamics and various values of N_0/P. The loop damping factor, ζ, was set to the common value of 0.707 so that (recalling Equation (18)) $G_1^2K_c$ could be substituted for G_2. The negative slope of the curves on the left side of Figure 3 shows the error due to signal dynamics decreases as K_cG_1 increases. The positive slope of the curves on the right side of the figure shows the error due to noise increases as K_cG_1 increases. Assuming an average N_0/P of 1.4×10^{-4} for GPS operation [SPI80:51], Figure 3 indicates a choice of $K_cG_1 \approx 40$ minimizes the loop delay error. In addition, because the maximum expected delay error is an order of magnitude below $\Delta/2$, the loop will remain in the linear operating region. Therefore, our earlier requirement of limiting the loop to operating in the linear region is not unduly restrictive or unreasonable.

3.8 CHAPTER SUMMARY

This chapter derived the equations governing the theoretical performance of the DLL. Limiting the loop to linear operation, the linearized loop transfer function, loop bandwidth, and maximum delay error were derived. Worst-case signal dynamics were then used to predict

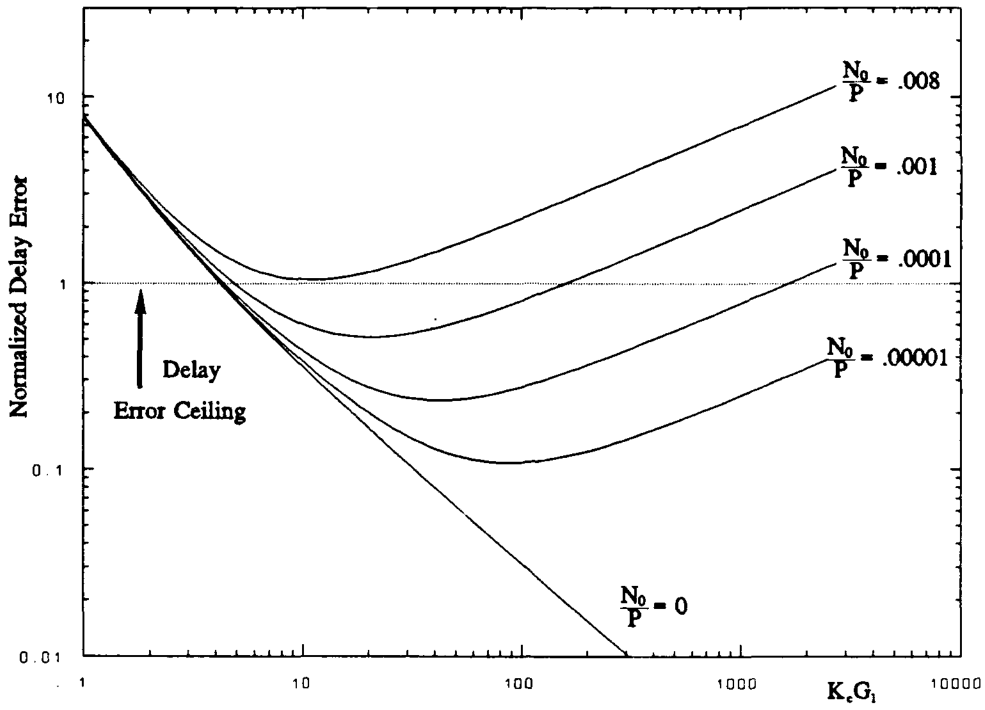


Figure 3 Normalized Delay Error Curves for the DLL

and graph the performance of the loop for various values of SNR and the loop parameters K_c and G_1. Finally, predicted loop delay errors proved the linear operation constraint is not unreasonable. Theoretical performance can now be compared to simulated performance.

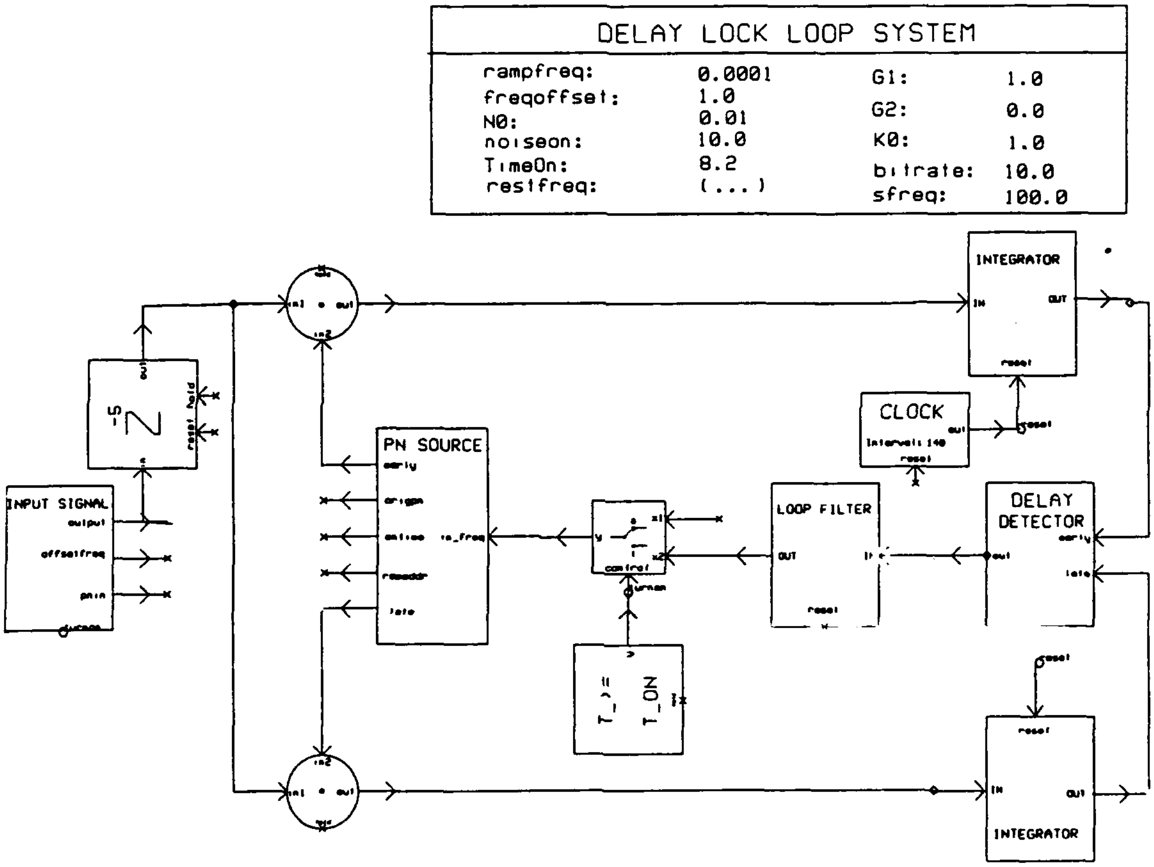
IV. DELAY LOCK LOOP SIMULATION

4.1 CHAPTER OVERVIEW

Although attempts to verify the theoretical performance of the Delay Lock Loop (DLL) through the use of simulations were ultimately unsuccessful, useful information was gathered in the process. Simulations were generated using the Signal Processing Workstation® (SPW™) simulation software developed by Comdisco® Systems of Foster City, California. SPW is a block oriented simulator that allows hierarchical designs to be built using an integrated graphical user interface. Simulations were constructed and executed on a SUN-4™ workstation. This chapter will first describe the DLL simulation configuration in general, followed by a detailed description of each customized subcomponent or block. The simulation procedure and results will then be presented. Finally, probable reasons for differences between theoretical expectation and simulation results will be discussed.

4.2 THE DELAY LOCK LOOP SYSTEM

The DLL is responsible for synchronizing a locally generated PN sequence with an identical PN sequence in the received signal. Figure 4 is an SPW generated depiction of the DLL test configuration. For simplicity, the numerous signal sinks (points where, during the simulation, signals are recorded for later analysis) have been removed. A signal, simulating the PN sequence in the GPS broadcast, is generated in the block labeled INPUT SIGNAL. Channel distortions in the form of additive, white, Gaussian noise (AWGN), Doppler shifts, and frequency ramps can be added to the applied signal via this block. See Section 4.7 for a detailed discussion of the INPUT SIGNAL block. The input signal is delayed by an amount equal to one-half chip duration as it enters the loop. By delaying the input, an undelayed replica produced by the PN source can be considered "Early". The inability of the simulator to dynamically modify this delay is one probable cause of simulation failure that will be discussed in Section 4.9. (NOTE: The phrase "inability of the simulator" in this context, can be read as a limitation of the SPW software and/or the author to produce the desired result.)


DELAY LOCK LOOP SYSTEM

rampfreq: 0.0001
freqoffset: 1.0
N0: 0.01
noiseon: 10.0
TimeOn: 8.2
restfreq: (...)
G1: 1.0
G2: 0.0
K0: 1.0
bitrate: 10.0
sfreq: 100.0

Figure 4 DLL Test Configuration

As Figure 4 shows, the input signal is first mixed with an "Early" and a "Late" replica of the PN sequence. The PN SOURCE, based on its input signal, attempts to produce PN sequences that are "On time" (match the received signal), "Early" (half a chip ahead of the received signal), and "Late" (half a chip behind the received signal). Section 4.3 provides further details on the PN SOURCE operation. After mixing, the signal is low-pass filtered (integrated) by the INTEGRATOR block. See Section 4.4 for a detailed description of the INTEGRATOR block.

After filtering, the Early and Late signals are passed through the DELAY DETECTOR which performs the operation (E-L)/(E+L). Section 4.5 discusses this operation. The detector output is next passed through the LOOP FILTER before being fed back into the PN SOURCE. LOOP FILTER operation is discussed in Section 4.6. Between the LOOP FILTER and the PN SOURCE is a timed switch. This switch prevents feedback into the PN SOURCE during the initial start-up time of the PN SOURCE.

Two PN SOURCE blocks were designed and tested for this thesis and will be discussed in detail in the next section. One PN SOURCE initially loads three random-access memory (RAM) banks with the PN sequence. Later, by varying the rate at which the RAM addresses are sequentially read, PN sequences of varying chip rates can be generated. Feedback during this RAM loading process would corrupt the sequence being written into RAM. After the RAM banks are loaded, the timer closes the switch and normal loop feedback operation begins. This same delay will be seen in the INPUT SIGNAL block discussed in Section 4.7.

4.3 THE PN SOURCE BLOCK

4.3.1 OVERVIEW. The PN SOURCE, the most complicated block in the simulation, is responsible for generating a PN sequence of the appropriate chip rate and replicas exactly one-half chip ahead and one-half chip behind. Two distinct types of PN sources were developed and tested. The first type uses RAM banks, first loading the PN sequence into memory and then varying the rate the RAM addresses are sequentially read, resulting in a PN sequence of varying chip rate. The second type of PN source attempts to generate the PN sequence of the appropriate rate on-the-fly or in real time. By holding (disabling) and enabling a PN generator at the appropriate rate, the desired PN sequence can be produced. Both of these approaches encountered difficulties based on limitations of SPW and are suspected of contributing to simulation failure.

4.3.2 THE RAM METHOD. Figure 5 depicts the components involved in producing a PN sequence by first writing to and then reading from RAM. A PN_SEQUENCE GENERATOR (upper-left corner of Figure 5) generates a random sequence of ones and zeroes at a predefined rate. The BINARY TO NUMERIC block converts each zero to a minus one to make the signal bi-polar. The TAPPED DELAY LINE converts the scalar sequence to a vector sequence. This is required because the VECTOR RAM will only accept vector inputs. The PN sequence is converted to a vector of length one, so the conversion is for format only.

The sequence is written into the VECTOR RAM at addresses selected by the ADDRESS COUNTER which is, in turn, controlled by the voltage controlled clock (VCC). The VCC block outputs a train of impulses at a frequency determined by its internal settings

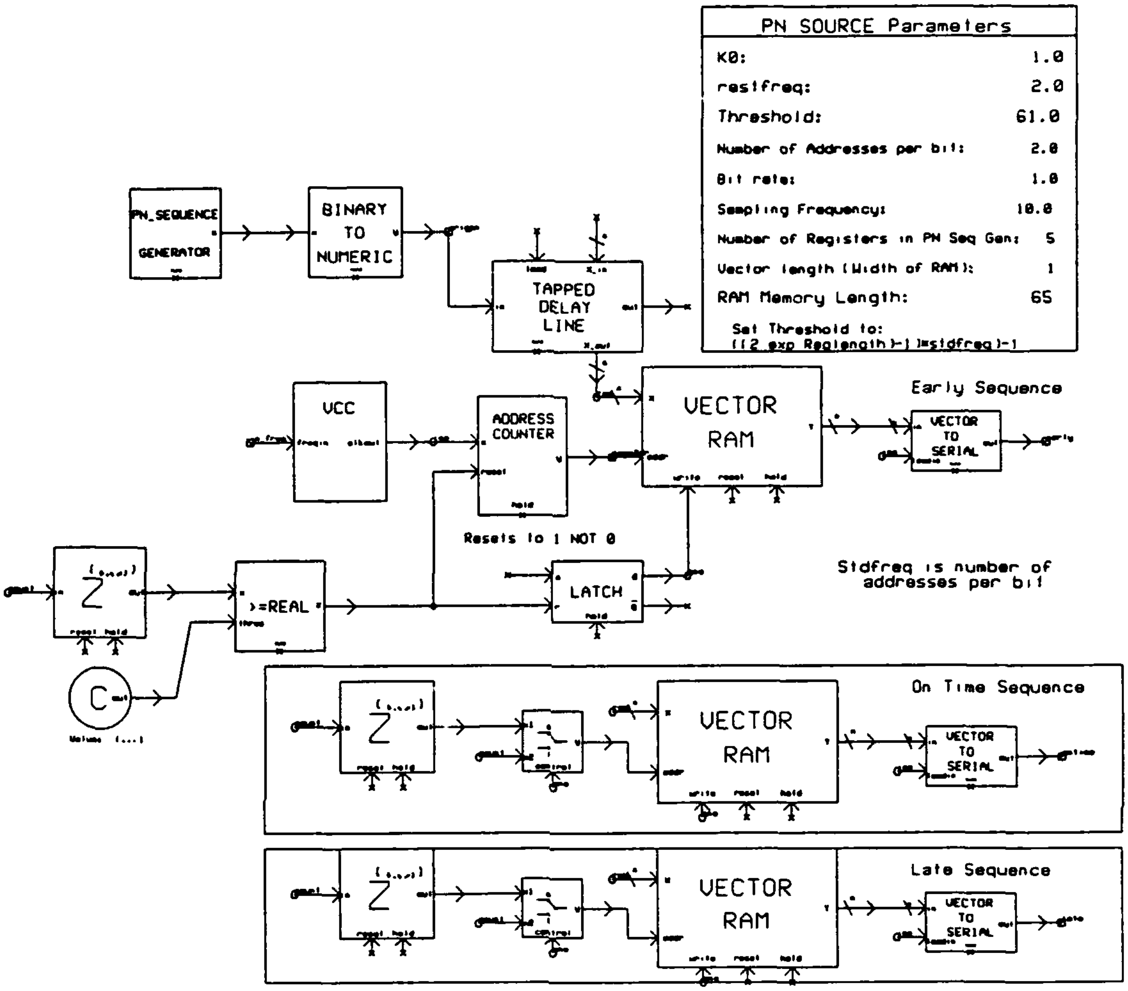


Figure 5 PN SOURCE Block

and the input signal. VCC operation is discussed in Section 4.3.2.1. Recall from Section 4.2 that the feedback input to the PN SOURCE (which inside the PN SOURCE is the VCC input) is disconnected during the initial RAM loading. Therefore, the VCC output frequency at this stage is determined solely by an internal setting, called its rest frequency. For the simulations performed, a typical setting was a rest frequency of 20 Hz. The PN sequence was generated at 10 chips (bits) per second. The impulses out of the VCC are counted by the ADDRESS COUNTER block. The impulse count is applied to the address input of the

VECTOR RAM. By setting the VCC rest frequency to twice the PN_SEQUENCE GENERATOR chip rate, the VECTOR RAM addresses are incremented twice as fast as the chip rate; therefore, each chip is written into two consecutive RAM addresses.

The PN_SEQUENCE GENERATOR consists of five feedback shift registers (FSRs). The actual GPS satellite transmission is generated by ten FSRs to produce a PN sequence with a period of 1023 chips. For the purposes of this simulation, five registers were deemed sufficient. Fewer registers result in a shorter PN sequence period (epoch) permitting a smaller RAM. Shorter epochs ease the evaluation of results, such as those obtained during testing of the RAM source. The PN sequence resulting from five FSRs has a length of (2^5-1) or 31 chips before it repeats. Once the 31st chip has been read into the 62nd address location of the VECTOR RAM, the PN_SEQUENCE GENERATOR is no longer needed and the "write" control signal of the RAM is pulled low to prevent further writing to RAM. This operation is performed by the threshold detector (>=REAL) and LATCH blocks in the center of Figure 5.

The ADDRESS COUNTER output, "y", is fed into the threshold detector (>=REAL) block where it is compared to the scalar constant (C) block value. The C block value is set to 61.0. Once the ADDRESS COUNTER output equals 61.0, the >=REAL block outputs a high signal that resets the ADDRESS COUNTER and the LATCH block. The LATCH block operates as an RS flipflop. The LATCH output is initially set high to provide a "write-enable" signal to the VECTOR RAM. The high output of the threshold detector resets the LATCH, dropping the "write-enable" signal and stopping the write operation of the RAM. Because the simulator does not allow feedback without some type of delay (to prevent indeterminate states), a delay block has to be inserted prior to the threshold detector. Because of this delay, the threshold value is set to 61.0 instead of the expected value of 62.0. The output of the VECTOR RAM is converted from vector back to scalar format by the VECTOR TO SERIAL block and output as the "Early" signal of the PN SOURCE.

While the first RAM is being loaded, the other two VECTOR RAMs in the PN SOURCE are also being written to. In this manner, all three RAMs contain identical values at identical addresses. The LATCH output controls the "write" signals of these RAMs in the same manner as the "Early" RAM. The LATCH also directs the switches controlling the VECTOR RAM address inputs. The moment the "write-enable" signals of the RAMs are dropped, the switches on the lower two RAMs insert delays into the address signals. The middle VECTOR RAM address is delayed an amount equivalent to half a chip from the "Early" RAM. Thus the middle VECTOR RAM output is equivalent to an "On Time" signal. The lower RAM address signal is delayed an amount equal to an entire chip. Therefore, the lower VECTOR RAM output is equivalent to a "Late" signal.

The two DELAY blocks may be another cause of simulation failure. While they are easily set to meet initial conditions, as the input signal experiences Doppler shifts and frequency ramps, the delays cannot be dynamically varied to maintain half a chip offset.

4.3.2.1 THE VCC BLOCK. The voltage controlled clock (VCC) operates much as a voltage controlled oscillator (VCO). The frequency of the output signal is a function of internal parameters and the input signal. The VCC outputs an impulse train with a frequency equal to its rest frequency (an internal parameter) plus K_c times the input signal. K_c is an internal gain parameter. Figure 6 depicts the internal components of the VCC block.

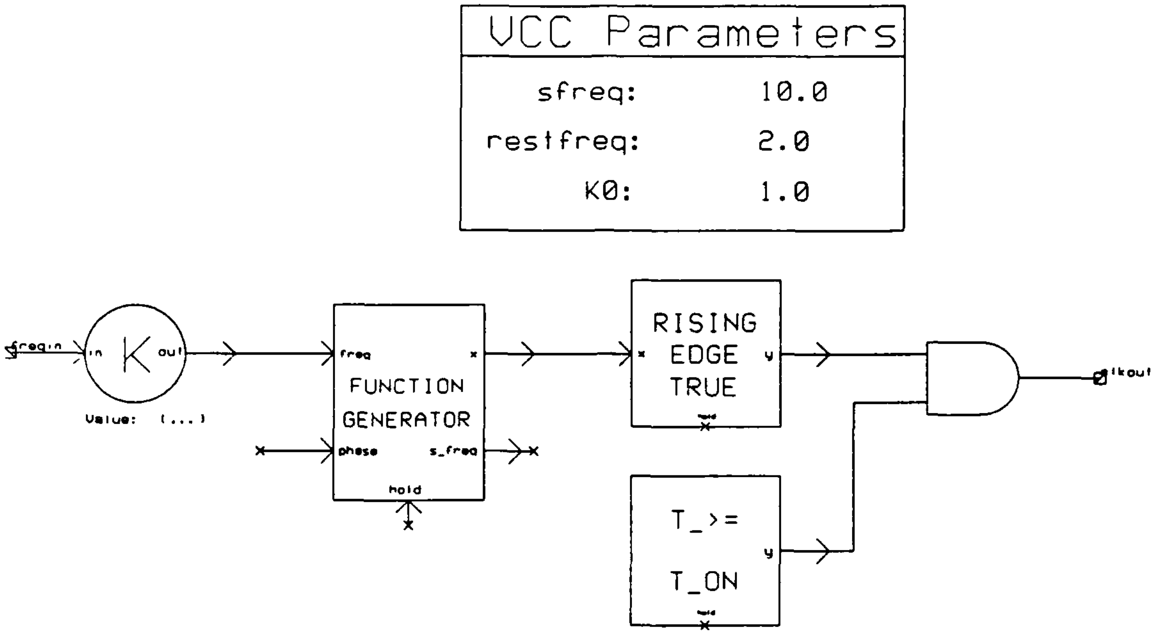


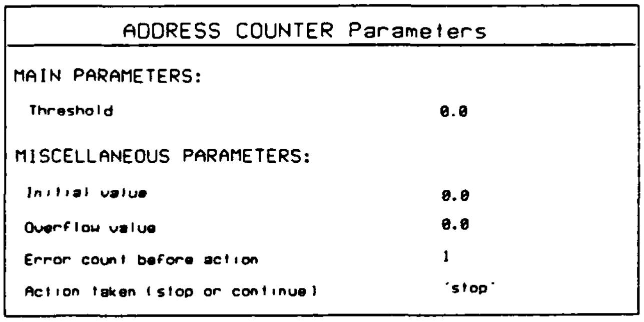
Figure 6 VCC Block

The input signal, "freqin", is multiplied by the constant K (K_c for our purposes) and is then passed to the FUNCTION GENERATOR. The FUNCTION GENERATOR is set to output a square wave at a frequency equal to the input signal plus an internally set rest frequency. The square wave output is then converted to an impulse train by the RISING EDGE TRUE block which outputs an impulse when the input signal crosses zero in a positive direction. The timed switch (T_>= T_ON) turns on once the simulator reaches an internally set point. The turn-on point is set just after zero so the initial rise of the square wave generated by the FUNCTION GENERATOR is not converted to an impulse. This prevents the ADDRESS COUNTER from incrementing to 1 at time zero. This delay in the first address increment allows the zero address of the VECTOR RAM to be written to.

A VCO was originally used instead of the K and FUNCTION GENERATOR blocks, but the oscillator signal was too unstable. The slope of the sine wave results in an ambiguity in the zero crossing, depending on the resolution provided by the sampling frequency. With a square wave output from the FUNCTION GENERATOR, the zero crossing is well defined.

4.3.2.2 THE ADDRESS COUNTER BLOCK. Figure 7 shows the ADDRESS COUNTER block components. The input signal, "x", is compared to zero by the >REAL block. If the signal is positive, a true signal is output. The output signal is next inverted, logically OR'ed with a "hold" control signal and then used to hold the SIMPLE COUNTER block. The SIMPLE COUNTER increments its output at each sample of the simulator's clock as long as its "hold" signal is low. Because the block's input is an impulse train, the threshold detector's output will be low until an impulse sends it high for one sample. Assuming the "hold" signal into the ADDRESS COUNTER block is low, the SIMPLE COUNTER "hold" will be high until the >REAL threshold detector detects an impulse. At that moment, the SIMPLE COUNTER's "hold" signal drops, thus allowing the COUNTER's output to be incremented. When the "reset" signal is strobed high, the SIMPLE COUNTER is reset to an internally set parameter,

For these simulations, due to delays in the loop, the counter resets to address ONE instead of ZERO. It then remains at ONE for two pulses instead of incrementing. Because address location ZERO and ONE have identical values, this skipping of address ZERO and repeating of address ONE has no effect on the simulation.



ADDRESS COUNTER Parameters

MAIN PARAMETERS:	
Threshold	0.0
MISCELLANEOUS PARAMETERS:	
Initial value	0.0
Overflow value	0.0
Error count before action	1
Action taken (stop or continue)	'stop'

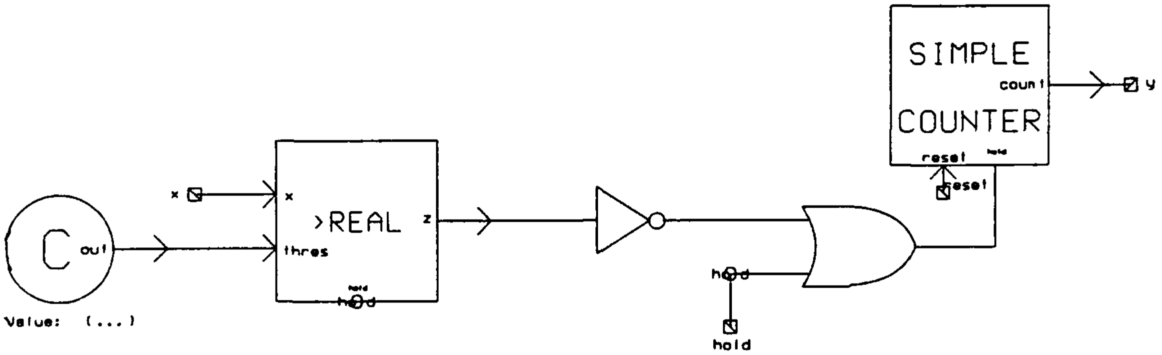
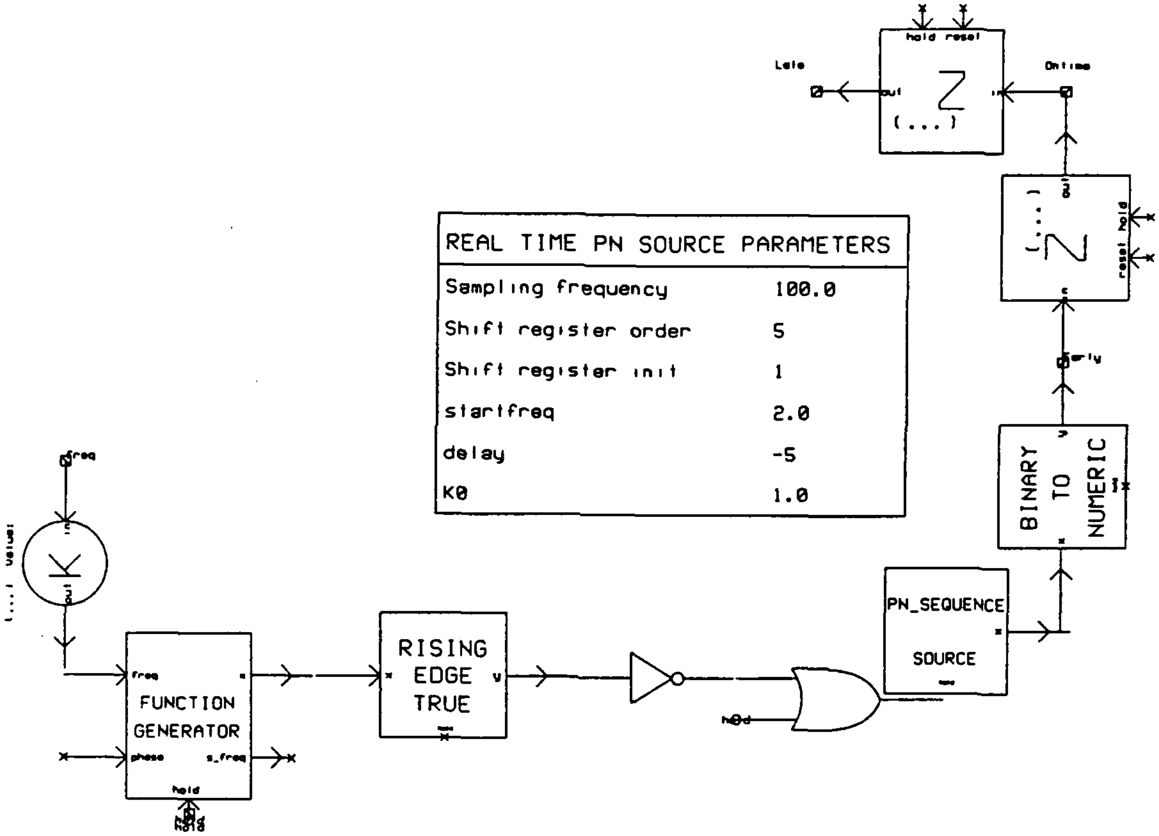


Figure 7 ADDRESS COUNTER Block

4.3.3 THE REAL TIME METHOD. Figure 8 depicts the second method of PN SOURCE attempted for this thesis. Instead of incrementing through RAM, as the previous method attempted, this PN SOURCE attempts to generate the PN sequence in real time. The block is essentially an inverted VCC controlling the output of a PN sequence generator.

The first three blocks create a VCC that outputs an impulse train with a frequency equal to K times the input voltage, "freq", plus an internal parameter, "startfreq". The impulse train is inverted and (assuming the block's "hold" signal is low) controls the PN_SEQUENCE SOURCE output. Each inverted impulse out of the VCC section drops the "hold" signal of the PN_SEQUENCE SOURCE, allowing it to output a chip. The output is held until the next impulse, at which time the next chip is generated. The sequence of chips is converted to a bi-polar signal by the BINARY TO NUMERIC block. Finally, the bi-polar PN sequence



REAL TIME PN SOURCE PARAMETERS	
Sampling frequency	100.0
Shift register order	5
Shift register init	1
startfreq	2.0
delay	-5
K0	1.0

Figure 8 REAL-TIME PN SOURCE Block

is passed through two bulk DELAY blocks. The first block sequence, "Early", is output before the first DELAY block. After a delay equivalent to half a chip, the next sequence, "On-time", is output. Finally, after a second DELAY equal to half a chip, the "Late" sequence is output.

Again, bulk DELAY blocks are used to create "Late" and "Early" sequences. Without the capability to vary these delays dynamically, the loop's performance is questionable. The theoretical impact of static delays was not analyzed.

4.4 The INTEGRATOR Block

The components of the INTEGRATOR block are shown in Figure 9. The block performs a simple digital integration (summation) by looping the output back and summing it with the next input. The switch in the feedback loop allows the integrator to be reset on command.

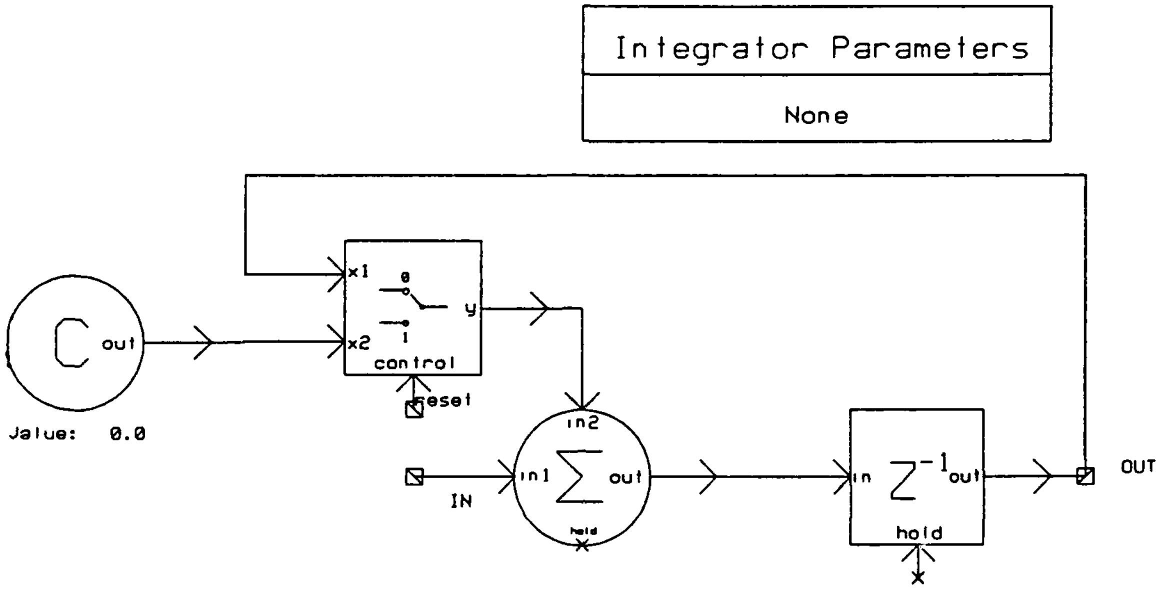


Figure 9 INTEGRATOR Block

4.5 THE DELAY DETECTOR BLOCK

The DELAY DETECTOR components are depicted in Figure 10. The "Late" input signal is negated by the K block and summed with the "Early" input signal. At the same time, the positive "Late" and "Early" signals are summed. The difference signal is then divided by the summation signal to implement the detector formula (E-L)/(E+L). Thus, the output signal is the normalized difference between the two inputs. It is possible for the inputs to sum to zero, therefore, the divider block may attempt to divide by zero. The block is set to output a zero and continue instead of terminating, should this event occur.

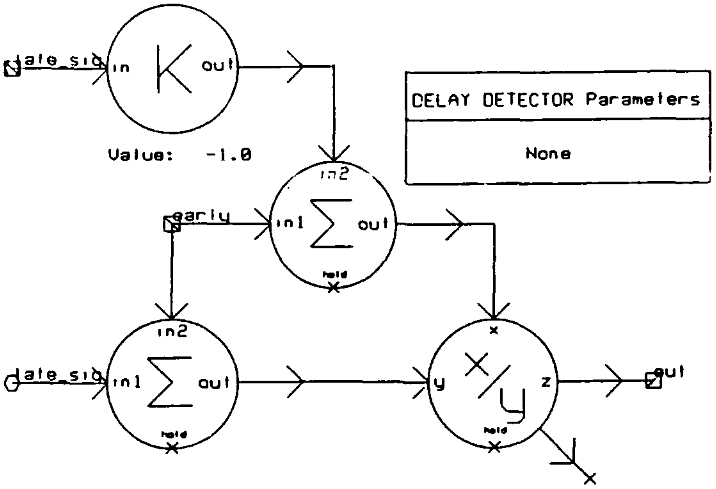


Figure 10 DELAY DETECTOR Block

4.6 *THE LOOP FILTER BLOCK*

Figure 11 shows the LOOP FILTER block components. The input signal is split and two operations performed. In the upper arm, the signal is multiplied by the value G_2 and then integrated by an imbedded INTEGRATOR block (described in Section 4.4). The lower arm is multiplied by the value G_1. These two arm signals are then summed and output. The transform function for this first-order filter is $F(z) = G_1 + G_2/(1 - z^{-1})$ or in Laplace notation $F(s) = G_1 + G_2/s$. If G_2 is set to zero, the loop filter becomes zero-order with a gain of G_1.

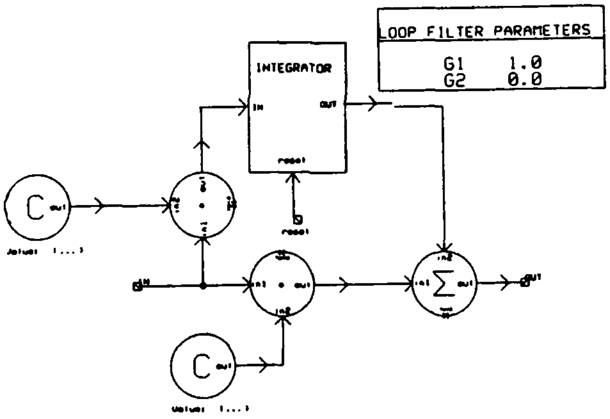
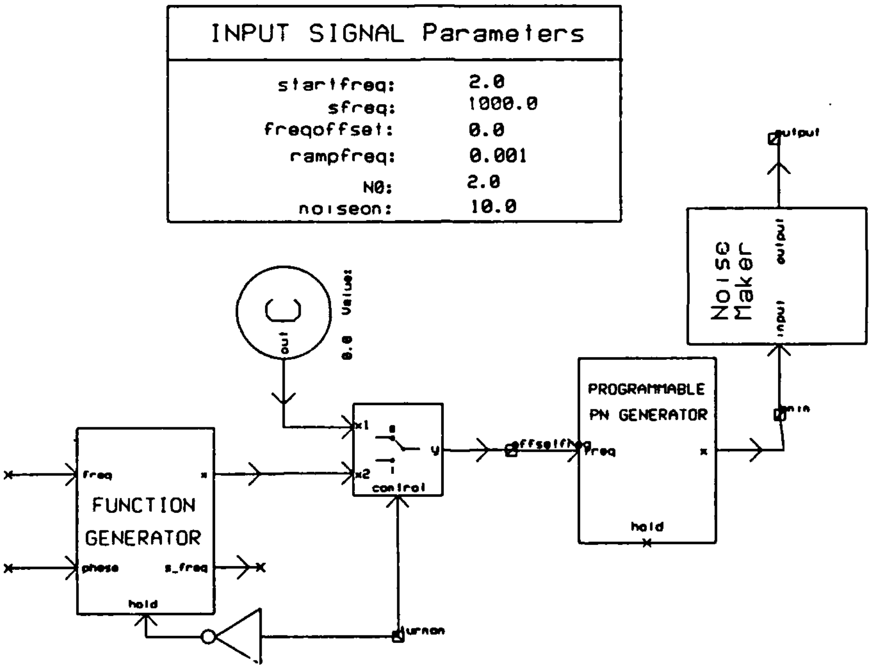


Figure 11 LOOP FILTER Block

4.7 THE INPUT SIGNAL BLOCK

The INPUT SIGNAL block generates the reference signal that the loop will attempt to synchronize with. This block also allows channel distortions such as AWGN, phase and frequency offsets, and frequency ramps to be introduced into the input signal. Figure 12 shows the components that make up this block. A PROGRAMMABLE PN GENERATOR block (described in Section 4.7.1) feeds a PN sequence into the NOISEMAKER block (described in Section 4.7.2) at a rate determined by internal parameters and its input signal. The input signal is initially connected to the C block with a setting of zero.



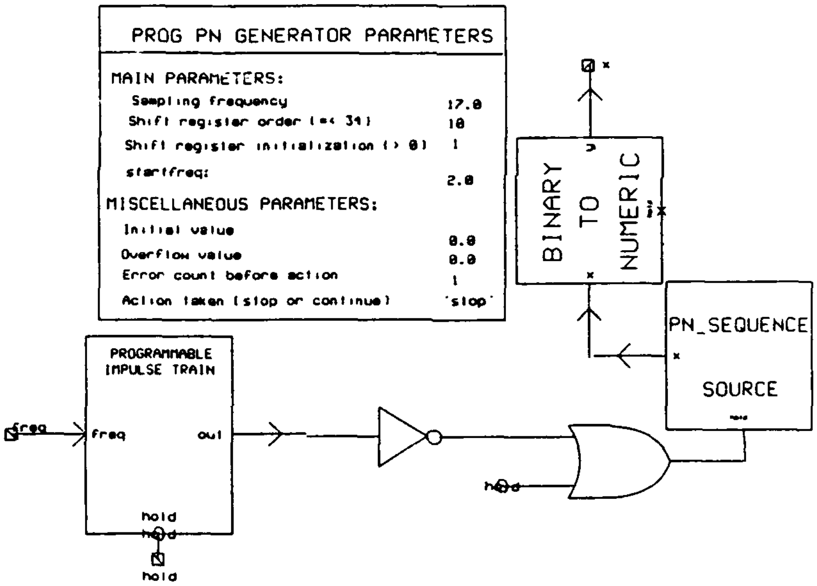
INPUT SIGNAL Parameters	
startfreq:	2.0
sfreq:	1000.0
freqoffset:	0.0
rampfreq:	0.001
N0:	2.0
noiseon:	10.0

Figure 12 INPUT SIGNAL Block

At a later time, determined by the external parameter "turnon", the FUNCTION GENERATOR is enabled and its output switched into the PROGRAMMABLE PN GENERATOR. This delayed switching to the FUNCTION GENERATOR allows the PN SOURCE in the loop to load its RAMs before signal dynamics begin to challenge the loop.

By setting the FUNCTION GENERATOR to output a low frequency, high amplitude triangular wave with a $\pi/2$ initial phase, the PN GENERATOR input becomes a slowly rising ramp that causes the output to be a PN sequence slowly increasing in frequency. The PN sequence is fed into the NOISE MAKER block which adds noise to the signal to produce a noisy PN sequence with a preset amount of noise power.

4.7.1 THE PROGRAMMABLE PN GENERATOR BLOCK. The PROGRAMMABLE PN GENERATOR block produces a PN sequence with a controllable period and chip rate. Figure 13 shows the components that make up this block. Note the similarity to the REAL TIME PN SOURCE of Section 4.3.3.



PROG PN GENERATOR PARAMETERS

MAIN PARAMETERS:	
Sampling frequency	17.0
Shift register order (=< 34)	10
Shift register initialization (> 0)	1
startfreq:	2.0
MISCELLANEOUS PARAMETERS:	
Initial value	0.0
Overflow value	0.0
Error count before action	1
Action taken (stop or continue)	'stop'

Figure 13 PROGRAMMABLE PN GENERATOR Block

The PROGRAMMABLE IMPULSE TRAIN outputs an impulse train with a frequency determined by its input and internally set parameters. The impulse train is inverted and used as a "hold" signal for the PN_SEQUENCE SOURCE. At each impulse, the "hold" signal drops and the PN_SEQUENCE SOURCE is allowed to output a chip. The stream of

ONES and ZEROES emitted by the PN_SEQUENCE SOURCE is converted to ONES and MINUS ONES by the BINARY TO NUMERIC block. Thus the output to the NOISE MAKER block is a bi-polar PN sequence.

Figure 14 depicts the internal components of the PROGRAMMABLE IMPULSE TRAIN. A FUNCTION GENERATOR produces a square wave with a frequency equal to the input signal plus an internally set rest frequency. This square wave is then passed through the RISING EDGE TRUE block which converts each rising edge of the square wave to an impulse. Thus the output of the PROGRAMMABLE IMPULSE TRAIN is a train of impulses with a frequency equal to the FUNCTION GENERATOR's rest frequency plus the input signal.

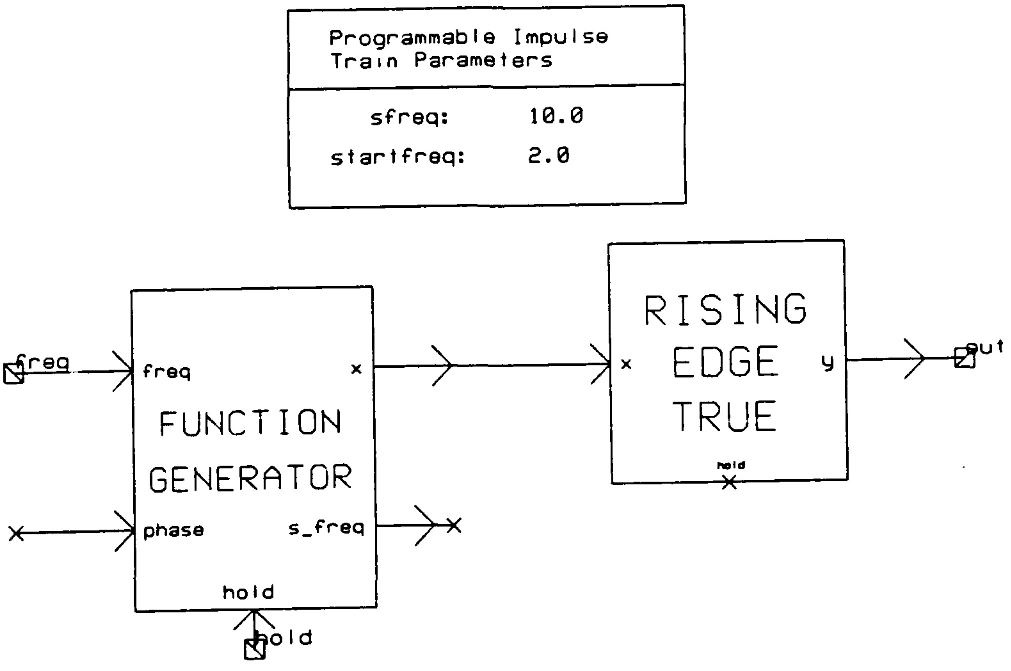
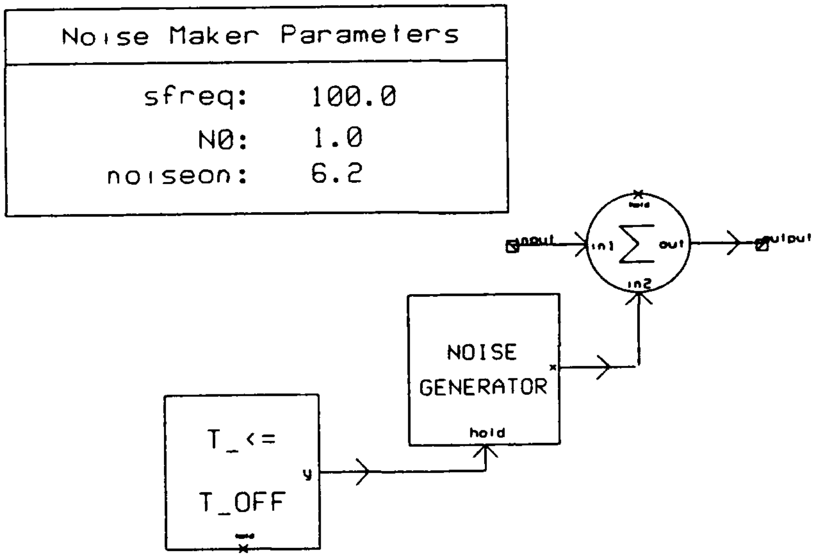


Figure 14 PROGRAMMABLE IMPULSE TRAIN Block

4.7.2 THE NOISE MAKER BLOCK. The NOISE MAKER block adds white, Gaussian noise with a definable variance (power spectral density) to its input signal to produce a desired N_0/P (input noise power spectral density over input signal power). Figure 15 shows the internal workings of the NOISE MAKER block. At a time specified by the internal parameter "noiseon", the timer (T_<=T_OFF) drops the hold signal of the NOISE GENERATOR block. The NOISE GENERATOR block begins generating WGN with a PSD set to the internal parameter, "N_0". The noise is then added to the input signal, "input", to produce an output signal with the appropriate N_0/P.



Noise Maker Parameters

sfreq: 100.0
N0: 1.0
noiseon: 6.2

Figure 15 NOISE MAKER Block

4.8 DLL SIMULATION METHODOLOGY AND RESULTS

The DLL, as discussed above, did not perform according to the theoretical predictions derived in Chapter III. Although the loop tracked an input signal experiencing the simplest signal dynamic, the phase step, it did not do so with the parameters K_c, G_1 and G_2 set to predicted values. More importantly, the DLL did not track any form of higher order signal dynamic such as the frequency step or ramp. In theory, using a first-order loop filter enables the loop to eliminate errors caused by phase and frequency steps and limit errors caused by frequency ramps (provided the frequency is not changing faster than a certain, predictable value) [SKL88:437]. The parameter G_2 determines the amount of "memory" the loop filter has. Setting G_2 equal to zero turns off the"memory" and the loop filter becomes zero-order. In theory, a loop with a zero-order filter can eliminate only those delay errors caused by a phase step.

The simulation of the DLL became unstable when G_2 was set to any value above zero. Therefore, G_2 was set equal to zero and signal dynamics such as frequency steps and ramps were turned off. Only noise was used to corrupt the input signal. Simulations were then run using various values of K_cG_1 and N_0/P. The loop was allowed to track the noise corrupted input signal for up to 20,000 iterations of the simulator. The loop delay error was found by counting the number of samples between zero crossings of the input PN sequence and the loop's estimate of the PN sequence. Dividing the number of samples of delay by the total number of samples in each chip at that moment in time provides a normalized delay error. For example, if the loop's predicted PN sequence has a zero crossing 25 samples after the input sequence's zero crossing and the sampling rate is such that each chip consists of 100 samples, then the loop normalized delay error is 0.25. The maximum delay was found by visually inspecting the two superimposed sequences.

Maximum delay errors for particular values of N_0/P and K_cG_1 are plotted in Figure 16. The diamonds and stars depict simulation points. The lines display theoretical performance of the loop for the respective values of N_0/P. As Figure 16 depicts, the simulated loop's performance is not predicted by the results of Chapter III. The general trend of increased delay error for increased K_cG_1 is apparent, as is the trend of increased delay error for increased noise power.

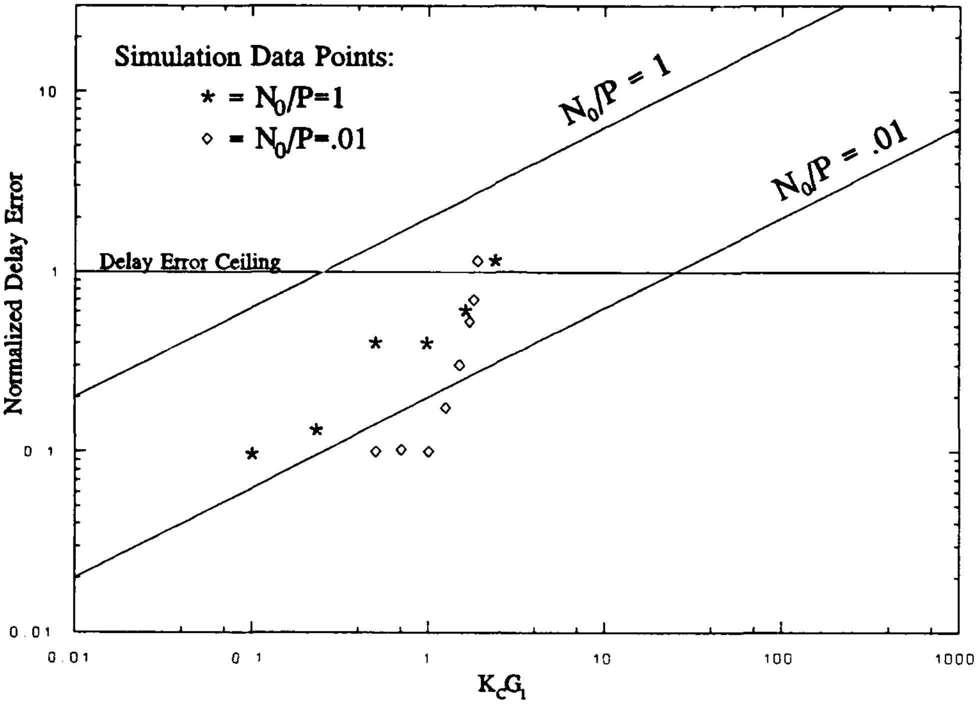


Figure 16 DLL Simulator Results

The sensitivity of the loop to the value of N_0/P is much greater than predicted by theory. The theoretical curves are parallel for various values of N_0/P. A line connecting the simulation points, on the other hand, has an increasing slope as N_0/P decreases. This increase in slope apparently results from a threshold effect on the value of K_cG_1. Increasing K_cG_1 to the value two results in the loop going unstable, irrespective of the value of N_0/P. For N_0/P less than 0.01 and K_cG_1 less than two, the delay error remains minor (one sample or less). As K_cG_1 approaches two, however, the delay error abruptly increases and the loop becomes unstable. For higher values of N_0/P, this increase in the delay error is more gradual.

4.9 CHAPTER SUMMARY

The procedure used for simulating the delay lock loop has been discussed. Each block and sub-block of the simulation has been described in detail. Two distinct types of PN sources have been described. The first PN source initially loads three RAMs and then varies the rate at which these RAMs are read to produce early, on-time, and late sequences. The second PN source generates the three sequences as needed. Both methods rely on bulk delay blocks that cannot be adjusted as the chip duration changes. Results have been presented to show the simulations do not adhere to predicted performance. Curves plotting simulation results follow general trends predicted by theory, but additional influences are apparent. A threshold effect around $K_cG_1=2$, not predicted by theory, limits the range of K_cG_1.

V. MODIFIED TANLOCK LOOP ANALYSIS

5.1 CHAPTER OVERVIEW

This chapter will present a detailed derivation of the equations governing the operation of the modified Tanlock loop (MTLL). Following this chapter overview, Section 5.2 will present a quick overview of the MTLL. Components, terminology, and notation will be introduced to prepare the reader for the detailed equation derivations of Section 5.3. Section 5.4 will continue the performance analysis under the simplifying assumption of no noise corrupting the input signal. Section 5.5 will then introduce noise into the system and derive the phase errors resulting from modulation and noise corrupting the input signal. Worst-case signal dynamics (derived in Appendix B) will be used in Section 5.6 to predict and graph the performance of the MTLL with various input SNRs and loop parameters. Section 5.7 will summarize the results of this chapter.

5.2 LOOP PREVIEW

The modified Tanlock loop (MTLL) is responsible for phase synchronizing the signal of a voltage controlled oscillator (VCO) in the GPS receiver with the carrier of the received GPS satellite broadcast. The VCO's phase synchronous output can then be mixed with the arriving signal, shifting it to baseband and allowing simplified data demodulation. A correctly designed loop will have a bandwidth sufficiently narrow to prevent as much noise as possible from entering the system, yet sufficiently wide to enable the system to track signals undergoing dynamics such as Doppler shifts. To improve performance in both noise and signal dynamics, phase error detectors with large linear operating regions are desired [LEE82].

A block diagram of the MTLL is shown in Figure 17. The received signal is first mixed with the output of the VCO to bring the signal to baseband. The VCO's output is split and one arm rotated 90 degrees to produce in-phase and quadrature components when mixed with the received signal. The signals are next mixed with a locally generated replica of the pseudo-noise (PN) sequence to despread the message signal.

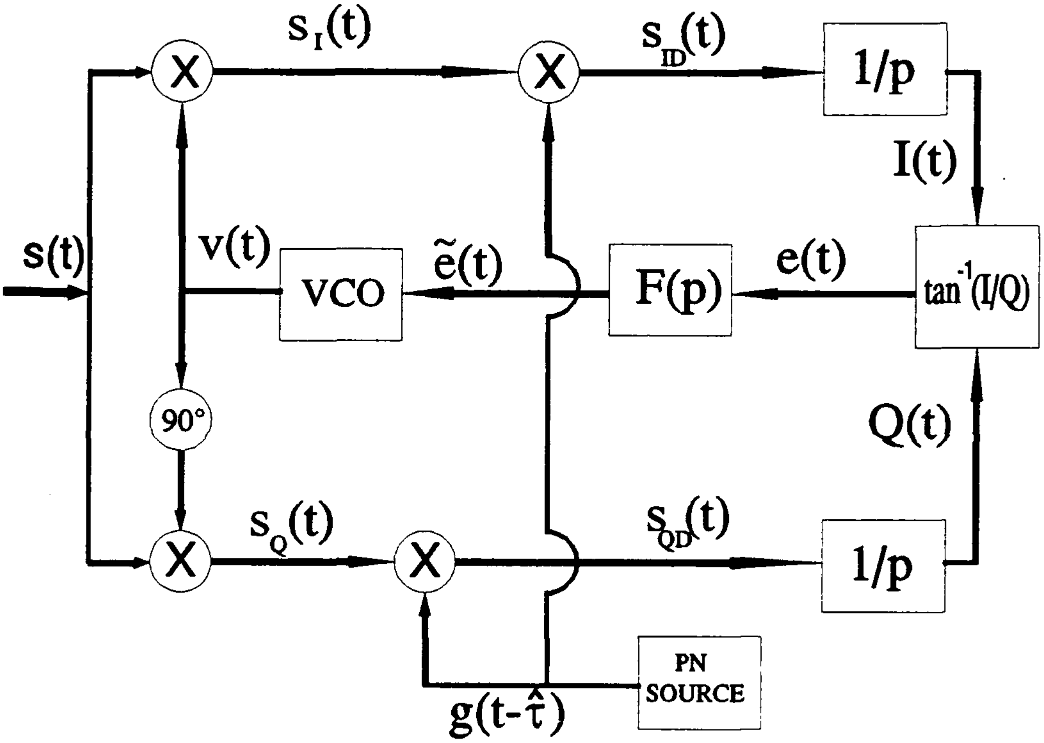


Figure 17 Modified Tanlock Loop

The despread signals are passed through low-pass filters to remove additive frequency components arising from the mixers. Next, the arctangent function is performed on the filtered signals to produce the phase error between the VCO output and the arriving signal. Finally, this phase error signal is passed through the loop filter and into the VCO to adjust the VCO output signal to bring any remaining phase error to zero.

5.3 EQUATION DEVELOPMENT

Let s(t) be the received signal entering the MTLL. The received signal consists of the message signal, m(t), and the spreading signal, g(t), BPSK modulated onto a carrier of frequency ω_0 plus additive white Gaussian noise (AWGN). The received signal can then be written as:

$$s(t) = \sqrt{2P}\, m(t)\, g(t) \sin[\omega_0 t + \Theta(t)] + n(t) \tag{25}$$

where:

P is the power of the received signal, s(t), in watts

m(t) is the unknown message signal BPSK modulated on the carrier, having values of ± 1

g(t) is the pseudo-random noise (PN) spreading signal, edge synchronized with m(t), having values of ± 1

$\Theta(t) \triangleq (\Delta\omega t + \theta_0)$ and is the result of channel rotation

$\Delta\omega \triangleq (\omega - \omega_0)$

ω is the actual frequency of s(t) in radians/sec

θ_0 is an arbitrary initial phase offset, in radians n(t) is AWGN, bandlimited by the IF frontend, having a bandpass representation of:

$$n(t) = \sqrt{2}\,\{N_c(t)\cos[\omega_0 t + \Theta(t)] - N_s(t)\sin[\omega_0 t + \Theta(t)]\} \tag{26}$$

where $N_c(t)$ and $N_s(t)$ are approximately statistically independent, stationary, low-pass WGN processes with single-sided noise spectral density N_0 watts-sec/rad and bandwidth $W_N/2 << \omega_0$. Note the propagation delay in the equations of Chapter III is not of interest in this discussion and has been incorporated into the value of t. Referring to Figure 17, s(t) is first mixed with the output of the voltage controlled oscillator (VCO), v(t), which is defined as:

$$v(t) = \sqrt{2P_v}\cos[\omega_0 t + \Theta_v(t)] \tag{27}$$

where:

P_v is the power of the VCO output, v(t), in watts

$\Theta_v(t) = (\Delta\omega_v t + \theta_v)$

$\Delta\omega_v = (\omega_v - \omega_0)$

ω_v is the actual frequency of the VCO output, in radians/sec

θ_v is an arbitrary phase offset, in radians

The output of the first mixer in the top or in-phase arm of the MTLL is then the in-phase baseband component of the received signal, $s_I(t) = s(t)v(t)$.

Multiplying Equations (25) and (27) yields:

$$s_I(t) = \sqrt{PP_v}\, K_m\, m(t) g(t) \{ \sin[2\omega_0 t + \Theta(t) + \Theta_v(t)] + \sin[\Theta(t) - \Theta_v(t)] \} \\ + \sqrt{2P_v}\, \cos[\omega_0 t + \Theta_v(t)]\, n(t) \tag{28}$$

where K_m is the mixer gain. Note $[\Theta(t) - \Theta_v(t)]$ is the phase error between the VCO output and the received signal. It will be defined as $\Theta_e(t)$. The second mixer in the in-phase arm of the MTLL in Figure 17 mixes the in-phase baseband signal, $s_I(t)$, with a locally generated replica of the spreading signal, g(t). This locally generated signal is denoted $g(t - \tau_e)$ where τ_d is the time offset between the locally generated PN sequence and the arriving PN sequence. Multiplying $g(t-\tau_e)$ by Equation (28) yields an equation having a product term $g(t)g(t-\tau_e)$. If $\tau_e=0$, this product is equal to one for all time. If $\tau_e \neq 0$, we can redefine this product to distinguish time varying and time invariant components to yield:

$$g(t)g(t-\tau_e) = R_g(\tau_e) + n_s(t, \tau_e) \tag{29}$$

where:

$R_g(\tau_e) = E[g(t)g(t - \tau_e)]$, the mean of the product $g(t)g(t-\tau_e)$ also known as the autocorrelation function of g(t) and

$n_s(t, \tau_e) = g(t)g(t-\tau_e) - R_g(\tau_e)$, the product term with the mean subtracted out, also called the self-noise.

Spilker demonstrated the self-noise term can be eliminated by the low pass filter following the despreading mixer, if the loop bandwidth, $B_L/2$ (in Hz), is much less than the PN sequence chip rate, 1/T [SPI63]. The resulting signal, I(t), entering the arctan function from the in-phase arm of the tanlock loop can be written:

$$\begin{aligned} \mathrm{I}(t) &= \sqrt{PP_v}\,K_m K_g \tilde{m}(t) R_g(\tau_e) \sin[\Theta_e(t)] \\ &\quad + \sqrt{2P_v}\,K_m K_g \tilde{g}(t-\tau_e)\,\tilde{n}_e(t) \end{aligned} \tag{30}$$

where K_g is the gain of the second mixer, a tilde represents a signal that has been filtered, and $n_e(t)$, the phase error noise term is defined as:

$$\tilde{n}_e(t) = \tilde{N}_c(t)\cos[\Theta_e(t)] - \tilde{N}_s(t)\sin[\Theta_e(t)] \tag{31}$$

A similar analysis results in the following equation for the signal, Q(t), entering the arctangent phase error detector from the quadrature arm:

$$\begin{aligned} Q(t) &= \sqrt{PP_v}\,K_m K_g \tilde{m}(t) R_g(\tau_e) \cos[\Theta_e(t)] \\ &\quad + \sqrt{2P_v}\,K_m K_g \tilde{g}(t-\tau_e)\,\tilde{n}'_e(t) \end{aligned} \tag{32}$$

where $n'_e(t)$ is a 90° phase shifted version of $n_e(t)$.

The output of the phase error detector, $e(t) = \tan^{-1}[I(t)/Q(t)]$, then becomes:

$$e(t) = \tan^{-1}\left[\frac{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\sin[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}_e(t)}{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\cos[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}'_e(t)}\right] \tag{33}$$

Notice this function is different than the original TLL introduced by Robinson in 1962 [ROB62]. It is, instead, the modification proposed by Lee and Un in 1982 [LEE82]. Robinson's phase error detector produced the $\tan[I(t)/Q(t)]$ which extended the linear region of the detector output over the traditional PLL or Costas loop. Lee's and Un's modification produces the $\tan^{-1}[I(t)/Q(t)]$, which is exactly linear over the region $-\pi \leq \Theta_e \leq \pi$. See Figure 18. Notice also the mixer gains, K_m and K_g, and the VCO output power, P_v, have been eliminated from the equations. Arm imbalance, which plagued early analog versions of the Costas loop, is not a factor here.

The output of the VCO, $v(t)$, is related to the input, $\tilde{e}(t)$, by the following equation:

$$v(t) = \exp\{j[\omega_0 t + K_0 \tilde{e}(t)]\} \tag{34}$$

where ω_0 is the quiescent frequency of the VCO (set to the nominal value of the carrier frequency) and K_0 is the VCO gain constant [SKL88:435]. The phase of the VCO output, $\Theta_v(t)$, is the integral of the phase component in Equation (34). Because the VCO acts as an integrator, the input to the VCO is the derivative of the desired phase component.

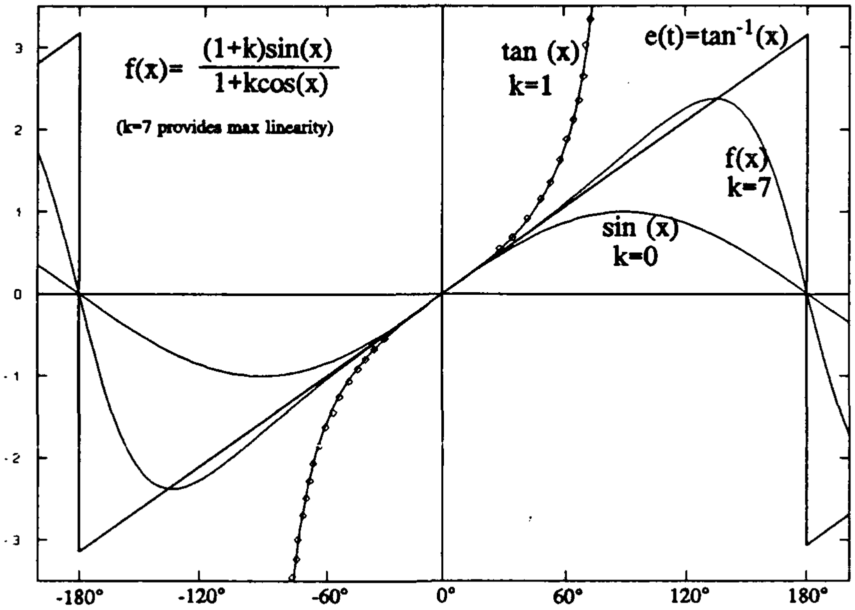


Figure 18 Phase Error Detector Characteristic Curves

This observation leads to the stochastic integro-differential equation describing the loop operation:

$$\begin{aligned} \Theta_v(t) &= \int K_0 \tilde{e}(t)\, dt \\ &= \frac{K_0 F(p)\, e(t)}{p} \end{aligned} \tag{35}$$

where:

p is the Heaviside operator, p = d/dt and 1/p indicates integration

F(p) is the loop filter transfer function

$\tilde{e}(t) = e(t)F(p)$

Letting the loop filter be a proportional plus integral filter with transfer function, $F(p)=(G_1p+G_2)/p$, we can substitute Equation (33) into Equation (35) to obtain:

$$\Theta_v(t) = K_0\left(\frac{G_1p + G_2}{p^2}\right)\tan^{-1}\left[\frac{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\sin[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}_e(t)}{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\cos[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}'_e(t)}\right] \tag{36}$$

5.4 ANALYSIS WITHOUT NOISE

Equation (33), in the absence of noise, simplifies to $\Theta_e(t)$, the phase error between the VCO output and the received signal, s(t). This illustrates two advantages of the MTLL over the traditional phase lock or Costas loops. First, the characteristic curve of the arctan phase detector is linear with a period of 2π as is depicted in Figure 18. This linearity extends the lock-in range of the MTLL over the phase lock or Costas loops. Extended linearity increases the dynamic range of the detector. Second, the signal entering the loop filter is not a function of the input signal power. Therefore, automatic gain control (AGC) circuitry is not required with this phase error detector.

Recalling $\Theta_e(t) = \Theta(t) - \Theta_v(t)$ and ignoring the noise component for the time being, algebraic manipulation allows Equation (36) to be rewritten as:

$$H(p) = \frac{\Theta_v(t)}{\Theta(t)} = \frac{K_0G_1p + K_0G_2}{p^2 + K_0G_1p + K_0G_2} \tag{37}$$

where H(p) is defined as the closed loop transfer function.

The error signal transfer function is then:

$$\frac{\Theta_e(t)}{\Theta(t)} = \frac{\Theta(t) - \Theta_v(t)}{\Theta(t)}$$

$$= 1 - H(p) \tag{38}$$

$$= \frac{p^2}{p^2 + K_0 G_1 p + K_0 G_2}$$

Using the Laplace Transform variable s=jω=p, Equation (37) can be rewritten in the standard transfer function form for a loop feedback system with an active first-order filter as described in [GAR79]:

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{39}$$

where:

$\zeta = (G_1/2)(K_0/G_2)^{1/2}$ is the loop damping factor, normally set to the critical damping value of 0.707 and

$\omega_n = (K_0G_2)^{1/2}$, the natural frequency of the loop in rads/sec.

The loop bandwidth, W_L, is then [COO86]:

$$W_L = \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega$$

$$= \omega_n\left(\zeta + \frac{1}{4\zeta}\right) \tag{40}$$

$$= \frac{1}{2}\left(K_0 G_1 + \frac{G_2}{G_1}\right) \frac{rads}{\sec}$$

5.5 *PHASE ERROR COMPONENTS*

The error signal, $\Theta_e(t)$, has two components. $\Theta_{em}(t)$ is the error due to signal modulation resulting from dynamics such as Doppler. $\Theta_{en}(t)$ is the error due to noise. The total phase error is then:

$$\Theta_e(t) = \Theta_{em}(t) + \Theta_{en}(t) \tag{41}$$

5.5.1 PHASE ERROR DUE TO MODULATION. The peak phase error due to modulation, Θ_{em}, is derived in Appendix A and is summarized here for the following cases of interest:

a. For a frequency step input of magnitude $\Delta\omega$:

Peak $\Theta_{em} = .645\Delta\omega/(K_0G_1)$ radians at time, $t = \pi/(2K_0G_1)$

b. For a frequency ramp input of magnitude $\dot{\omega}$ rads/sec^2:

Peak $\Theta_{em} = 2\dot{\omega}/(K_0G_1)^2$, radians at time, $t = 0$

5.5.2 PHASE ERROR DUE TO NOISE. Recalling the input noise power spectral density is $S_n(\omega) = N_0/2$ and Equation (40), the variance of the phase error due to noise, $\Theta_{en}(t)$, is found to be:

$$\begin{aligned} \sigma^2_{en} &= \frac{1}{P}\int_{-\infty}^{\infty} S_n(\omega)\,|H(\omega)|^2\,d\omega \\ &= \frac{N_0W_L}{2P} \\ &= \frac{N_0}{4P}\left(K_0G_1 + \frac{G_2}{G_1}\right) \end{aligned} \tag{42}$$

Note the SNR in the loop is the reciprocal of the phase error variance.

The loop will lose lock when the phase error exceeds π. Assuming Gaussian noise, the probability of the loop losing lock due to noise is then [COO86:351]:

$$\begin{aligned} P[\Theta_{en} > \pi] &= 2Q\left[\frac{\pi}{\sigma_{en}}\right] \\ &= 2Q\left[\pi\sqrt{\frac{2P}{N_0 W_L}}\right] \qquad (43) \\ &= 2Q\left[\pi\sqrt{\frac{4PG_1}{N_0(K_0G_1^2 + G_2)}}\right] \end{aligned}$$

Combining the errors due to modulation and noise results in the following restriction for the loop to maintain lock:

$$\begin{aligned} \pi &\geq \Theta_{em} + \sigma_{en} \\ &\geq \frac{.645\Delta\omega}{K_0G_1} + \frac{2\dot{\omega}}{(K_0G_1)^2} + \sqrt{\frac{N_0}{4P}\left(K_0G_1 + \frac{G_2}{G_1}\right)} \qquad (44) \end{aligned}$$

Again we have the classic tradeoff in receiver design. As the product of K_0G_1 increases, the phase errors due to modulation decrease, while the phase error due to noise increases. By determining worst-case signal dynamics and expected noise levels, optimum loop parameters K_0, G_1, and G_2 can be calculated.

5.6 *WORST-CASE SIGNAL DYNAMICS*

Worst-case signal dynamics expected for this receiver are derived in Appendix B and are summarized here:

Worst-case frequency step: $\Delta\omega$ = 44,000 rad/sec

Worst-case frequency ramp: $\dot{\omega}$ = 1760 rad/sec^2

Substituting these values into Equation (44) and ignoring the noise term, for the moment, yields:

$$\pi \geq \frac{28380}{K_0 G_1} + \frac{3520}{(K_0 G_1)^2} \tag{45}$$

To satisfy Equation (45), $K_0G_1 > 9035$ for worst-case signal dynamics and no noise.

Because of the variety of GPS receiver operating environments, no practical worst-case noise value can be assumed for this system. Instead, Figure 19 graphs the phase error normalized with respect to the maximum allowable phase error π, versus K_0G_1 for worst-case signal dynamics and various values of N_0/P. Figure 19 graphs Equation (44). The loop damping factor introduced in Equation (39) has been set to 0.707. Therefore, G_2 can be replaced with $(G_1^2K_0)/2$. The assumed worst-case signal dynamics have been reduced by a factor of 100 to reduce the error due to signal dynamics. This reduction is required to bring the total phase error into the lock-in region. As Figure 19 depicts, an increasing K_0G_1 reduces the phase error due to modulation but also increases the error due to noise.

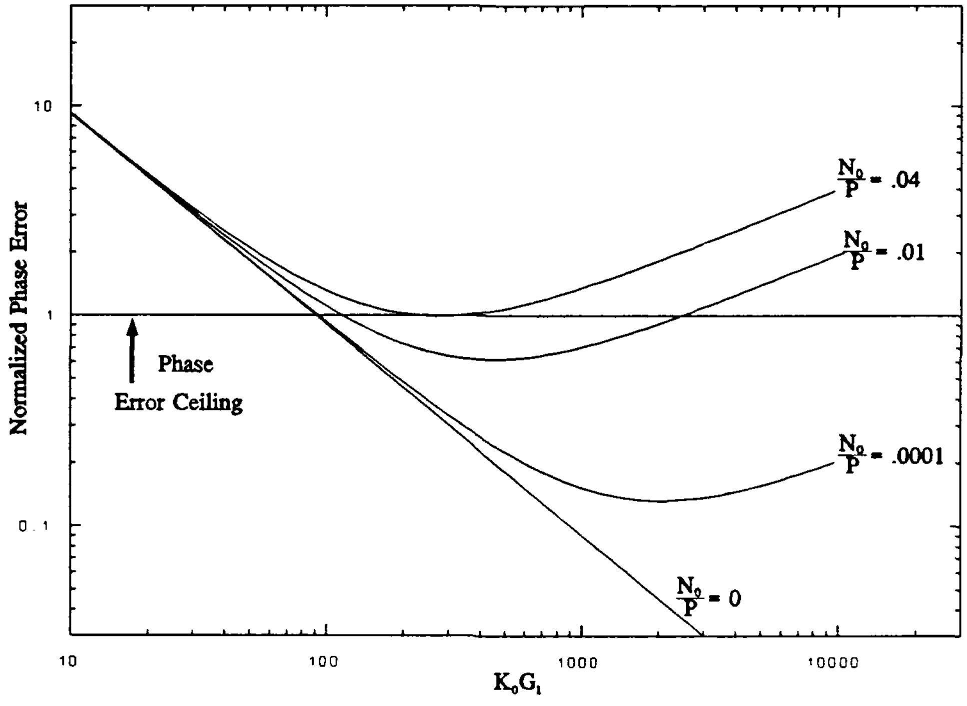


Figure 19 Normalized Phase Error Curves for the MTLL

5.7 CHAPTER SUMMARY

Equations governing the theoretical performance of the modified Tanlock loop have been derived. The extended linear region provided by the arctangent function over the tangent function has been demonstrated, as has the immunity of the MTLL to variations in input signal power and arm imbalance. The extended linear region of the MTLL allowed calculation of the loop bandwidth, probability of losing lock, and phase error without using artificial linear restrictions as was required with previous loop designs. Using expected worst-case signal dynamics, the peak phase error has been predicted and graphed for various values of N_0/P and loop parameters K_0 and G_1. Simulation results can now be compared to these theoretical predictions.

VI. MODIFIED TANLOCK LOOP SIMULATION

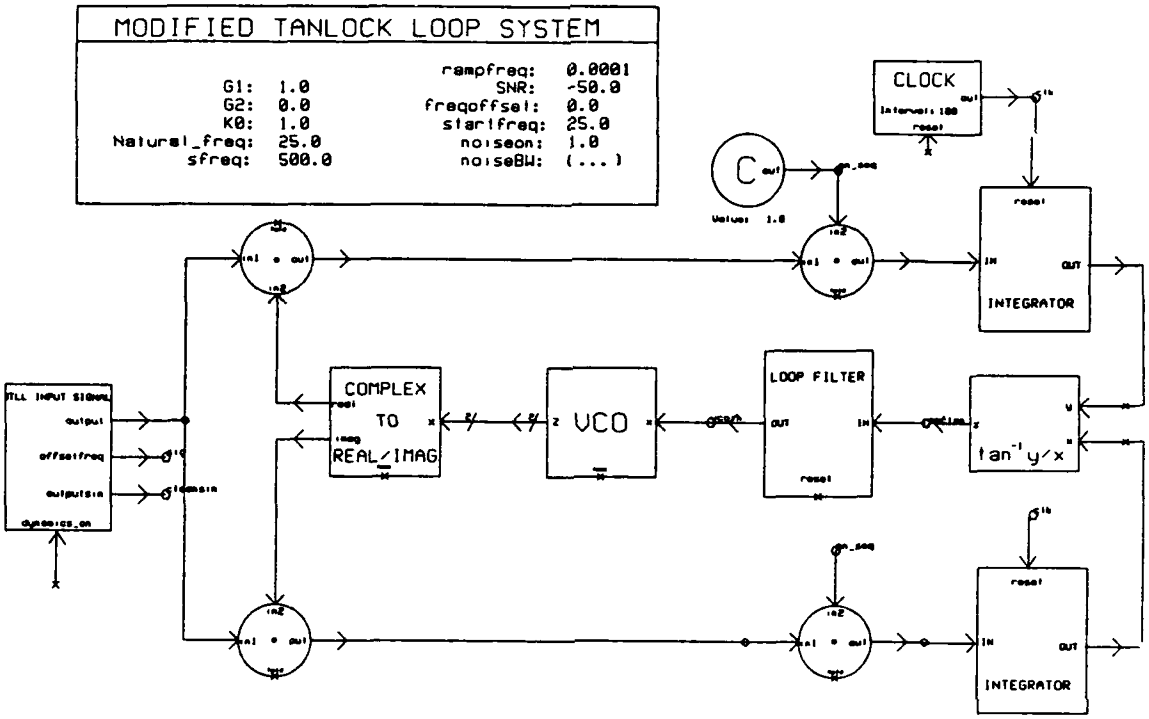
6.1 CHAPTER OVERVIEW

Simulations developed for this thesis failed to verify the theoretical performance of the Modified Tanlock Loop (MTLL) predicted in Chapter V. This chapter will provide an overview of the MTLL simulations performed on a SUN-4 workstation using the software package Signal Processing Workstation (SPW) developed by Comdisco Systems, Inc of Foster City, California. This chapter will first describe the MTLL simulator configuration in general, followed by a detailed discussion of each customized subcomponent or block. The simulation procedure and results will then be introduced. Finally, probable reasons for differences between theoretical and simulated performance will be discussed.

6.2 THE MODIFIED TANLOCK LOOP SYSTEM

The purpose of the MTLL is to synchronize the phase of a local oscillator to the phase of a received signal. Once synchronization is achieved, the oscillator output can be mixed with the received signal to eliminate any modulation and allow easy recovery of data superimposed on the signal. In a GPS application, the MTLL is responsible for tracking and removing the carrier of the GPS satellite broadcast.

Figure 20 is an SPW generated depiction of the MTLL. Signal sinks (points where signals are recorded during the simulation for later analysis) have been removed for clarity. An input signal is generated by the MTLL INPUT SIGNAL block. The block is capable of adding distortions to the input signal such as noise, Doppler shifts, and frequency ramps. Section 6.5 provides a detailed discussion of the MTLL INPUT SIGNAL block.


MODIFIED TANLOCK LOOP SYSTEM

G1: 1.0
G2: 0.0
K0: 1.0
Natural_freq: 25.0
sfreq: 500.0
rampfreq: 0.0001
SNR: -50.0
freqoffset: 0.0
startfreq: 25.0
noiseon: 1.0
noiseBW: (...)

Figure 20 MTLL Test Configuration

The input signal is first split into an upper and lower arm. The upper arm signal is mixed with the output of the VCO block and then enters another mixer. In an actual GPS carrier tracking loop, the second mixer would multiply the arm signal with a locally generated PN sequence to remove the spreading PN sequence in the arriving signal. This is the "On-time" PN sequence discussed in Chapters III and IV. Because of the difficulties encountered in making the loop work, the second input to this mixer was never upgraded beyond a constant value of one.

The signal next enters the INTEGRATOR block which performs a low pass filtering function. Section 6.3 describes the operation of the INTEGRATOR block. Meanwhile, the lower arm signal has been mixed with the second VCO output which is 90° out of phase with the first VCO output. For this reason, the upper arm is normally referred to as the in-phase or I arm and the lower arm referred to as the Quadrature or Q arm. After being mixed with a value of one, the signal is integrated by the INTEGRATOR block.

The I and Q signals then enter the $\mathrm{TAN}^{-1}(X/Y)$ block which performs the four quadrant arctangent function on the inputs. This block detects the phase error between the arriving signal and the VCO output. The arctangent output is then passed through the loop filter described in Section 6.4. Finally, the filtered signal is used to drive the VCO block which produces a complex output. The complex signal is split by the COMPLEX TO REAL/IMAG block which is equivalent to producing a sine and cosine output. The outputs are then fed back into the first mixers in the loop arms.

6.3 THE INTEGRATOR BLOCK

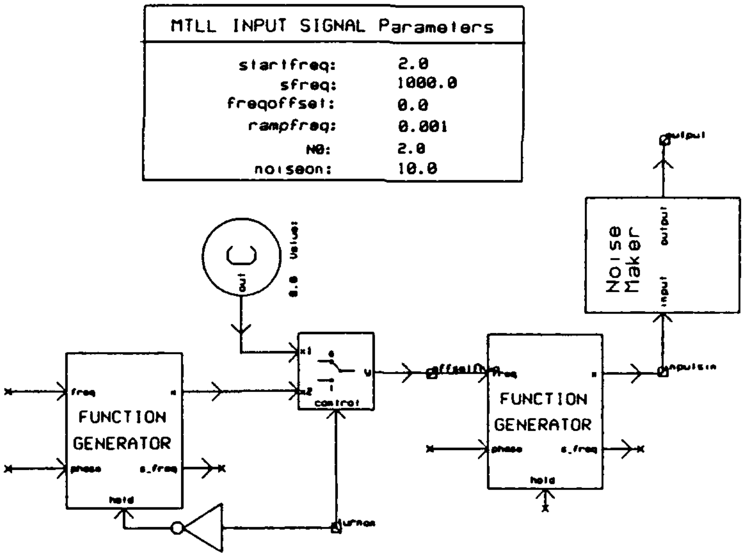
The INTEGRATOR block is the same one used in the DLL. See Figure 9 for a depiction of the internal components and Section 4.4 for a detailed description of the block's operation.

6.4 THE LOOP FILTER BLOCK

The LOOP FILTER block is the same one used in the DLL. See Figure 11 for a depiction of the internal components and Section 4.6 for a detailed description of the block's operation.

6.5 *THE MTLL INPUT SIGNAL BLOCK*

The MTLL INPUT SIGNAL block is similar to the one used in the DLL. It generates a sinusoid to simulate the carrier of the GPS satellite broadcast. It also adds noise and signal dynamics such as Doppler shifts and frequency ramps to the input signal. Figure 21 depicts the internal components of the MTLL INPUT SIGNAL block. The PROGRAMMABLE PN GENERATOR found in the INPUT SIGNAL block of the DLL has been replaced with a FUNCTION GENERATOR. In this way, a sinusoid instead of a PN sequence is generated for the loop. The NOISE MAKER block is identical to the NOISE MAKER block in the DLL. See Figure 15 for a depiction of the internal components and Section 4.7.2 for an explanation of its operation.



MTLL INPUT SIGNAL Parameters

startfreq: 2.0
sfreq: 1000.0
freqoffset: 0.0
rampfreq: 0.001
N0: 2.0
noiseon: 10.0

Figure 21 MTLL INPUT SIGNAL Block

6.6 *MTLL SIMULATION METHODOLOGY AND RESULTS*

The SPW simulations failed to verify the theoretical results predicted in Chapter V. While the loop was capable of tracking a wide range of input signal dynamics and noise, the performance did not coincide with theoretical expectation. Regions of loop stability and instability for the simulations did not occur where theory predicted. One point of disagreement between theory and simulation was the loop filter parameter G_2. Setting $G_2 > 0.01$ resulted in the loop becoming unstable and oscillating uncontrollably, although theory predicted much greater values of G_2 were possible. A similar problem plagued the DLL of Chapters III and IV.

Because of the apparent difficulties with G_2, higher order signal dynamics were turned off and G_2 was set to zero. Simulations were then performed using only AWGN to corrupt the input signal. The simulation was allowed to run for 20,000 iterations during which time the loop attempted to track an input sinusoid corrupted by a preset amount of noise. At the end of the simulation, the VCO's output was superimposed on the input sinusoid and the number of samples between zero crossings of the VCO output and the sinusoid input were counted. Dividing the magnitude of this difference by the number of samples in half a sinusoidal cycle at that moment produces the normalized phase error. For example, if the VCO output has a zero crossing 20 samples before (or after) the input sinusoid and the sinusoid consists of 100 samples per cycle, the normalized phase error is 0.4. The maximum normalized phase error for a particular noise power and K_0G_1 setting was found by visually inspecting the entire simulation record and measuring the error at the greatest separation between the two signals. The low variation in the phase error for the duration of the simulation made this approach feasible.

Figure 22 compares the theoretical and simulated results of the MTLL with no signal dynamics, $G_2=0$, and various values of input N_0/P (input noise PSD over input signal power) and K_0G_1. Diamonds, stars, and plusses indicate simulation data points and lines indicate

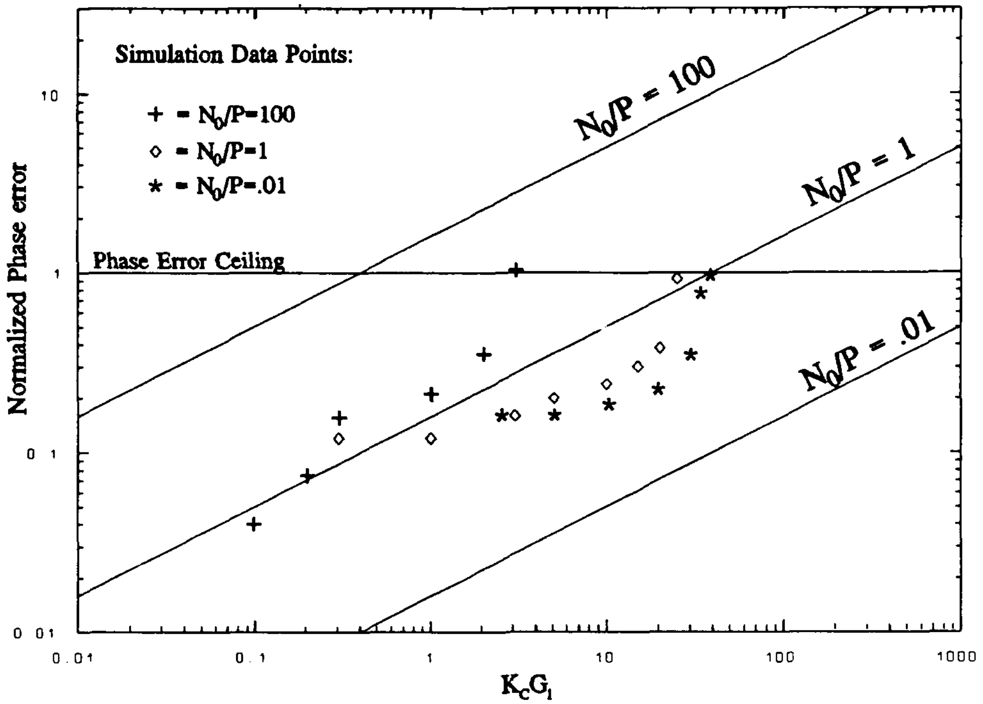


Figure 22 MTLL Simulation Results

theoretical performance. As Figure 22 shows, simulations did not confirm theoretical predictions made in Chapter V. As in the case of the DLL, three general trends are readily apparent:

1) Phase error increases as the product K_0G_1 increases.
2) Phase error increases as noise power increases.
3) A threshold effect comes into play as K_0G_1 increases. In this case the threshold effect occurs when K_0G_1 nears 40.

As N_0/P was reduced below 0.01, the normalized phase error continued to exceed 1 as K_0G_1 approached 40. No apparent reason for this threshold effect was found.

6.7 CHAPTER SUMMARY

The procedure used to simulate the MTLL has been discussed. Each customized block and sub-block used in the simulation has been described in detail. Results have been presented to show the simulations do not adhere to predicted performance. In particular the loop parameter G_2, which controls the amount of "memory" in the loop filter, made the loop unstable for values above 0.01. Without the influence of a large G_2, higher order signal dynamics could not be investigated and so were turned off. Simulations were performed with only AWGN corrupting the input signal. Plots of simulation results show the simulation following general trends predicted by theory, but additional influences are apparent. A threshold effect around $K_0G_1=40$, not predicted by theory, limits the range of K_0G_1.

VII. CONCLUSION AND RECOMMENDATIONS

7.1 SUMMARY

This thesis analyzed two critical circuits for GPS receiver design. A code-tracking loop, the delay lock loop (DLL), and a carrier-tracking loop, the modified Tanlock loop (MTLL), were investigated. The theoretical performance of the DLL tracking an input signal corrupted by modulation due to movement and noise was analyzed.

The characteristic curve of the loop delay detector (discriminator characteristic) was shown to be exactly linear when the delay error was restricted to remaining less than half a chip duration. Restricting the loop operation to this linear region, the linearized loop transfer function and equation were derived. The linearized loop equation expresses the normalized loop delay estimation in terms of the loop transfer function and the output of the delay detector. The loop transfer function was re-written in the standard form of a second order loop feedback system with an active first order loop filter. This standard form led to an expression for the loop noise PSD and bandwidth as well as introducing the concept of the loop natural frequency and damping factor. With an expression for the loop transfer function and the noise PSD, the maximum expected delay error due to noise, and the probability of losing lock due to noise could be predicted.

Expressions were derived for worst-case signal dynamics due to velocity and acceleration. In turn, maximum expected delay errors due to these signal dynamics were calculated. Graphing the delay error due to signal dynamics for various values of noise and loop parameters K_c and G_1, defined a region of stability where the delay error was less than half a chip in duration. The size of the stable region indicated restricting the loop to the linear operating region was neither unduly restrictive nor unreasonable. General trends in loop performance were evident in the graph. As input noise increased, the delay error was predicted to increase. Increasing the loop gain, in the absence of signal dynamics, was also predicted to increase the delay error.

A similar procedure was followed in the analysis of the MTLL, with one notable exception. The arctangent phase error detector of the MTLL was shown to be exactly linear with a period of 2π. Therefore, no linear approximation of the loop operation was required.

Once theoretical performance was derived, simulations were presented to compare the theoretical predictions to simulated performance. Simulations were written using SPW, a signal processing software package, developed by Comdisco, Inc. The simulations were executed on a Sun-4 workstation. Graphs, depicting the maximum estimation errors produced by the loops, were presented. Although the general trends predicted by the theoretical expressions were evident, the simulated loops failed to perform as expected. Additional (unpredicted) influences appeared to be driving the loop errors as K_cG_1 increased. No quantitative measurement of the discrepancies was attempted due to the magnitude of the disagreement.

7.2 CONCLUSIONS/LESSONS LEARNED

7.2.1 THE DELAY LOCK LOOP. The primary difficulty in simulating the DLL is in accurately generating PN sequences with the proper amount of delay. SPW is unable to simulate a DLL with either of the PN source methods described in Chapter IV. Both the RAM method and the real time method require DELAY blocks to produce the "On-time" and "Late" PN sequences, delayed by half a chip and a whole chip, respectively. As the chip rate varies, these delays must vary. SPW is unable to vary this delay dynamically. The delay is a parameter set at the beginning of the simulation and is forced to remain constant for the duration of the simulation. The theoretical derivation of Chapter III was based on half a chip offset between sequences, however, this is not required [GIL66]. Other delays have been investigated. As long as the "Early" and "Late" signals are equally separated from the "On-time" signal, the result should be similar. The limits under which this generalization is valid are unknown to this author. The theoretical impact of a varying delay was not investigated.

An unexplained problem arose with the DLL that may or may not be related to the PN source difficulties. There is an apparent threshold effect limiting the maximum value of the product K_cG_1. While no such limit is apparent in the theoretical derivation of Chapter III, an obvious threshold effect occurs in the simulations performed. See Figure 16. The rate at which the simulated signals were sampled may have a bearing on this threshold. If the sampling rate is increased, greater resolution of the signal is obtained and finer adjustments to the PN sequence spacing is possible. However, the practical amount of oversampling of the received signal is limited due to the high chip rate of the two PN sequences used in GPS (1.023 MHz for one sequence and 10.23 MHz for the other).

7.2.2 THE MODIFIED TANLOCK LOOP. A possible cause of the MTLL not performing as predicted may be the method in which performance was measured. The method used to measure the normalized phase error was identical to the one used to measure the normalized delay error of the DLL in Chapter IV. Its validity is unquestionable for measuring the delay between PN sequences in the DLL but a problem arises for the MTLL. The difference between zero crossings for sinusoids is a measure of the phase difference only if the instantaneous slopes of the two signals are identical at the point of measurement. The phase difference between two sinusoids undergoing erratic changes (often the case for the VCO output as it attempts to track an extremely noisy sinusoid) are almost impossible to measure. Because of the consistency of the measurement, however, it does provide a rough estimate of the phase error and should be reliable to depict trends in the data.

In an effort to better understand the simulated MTLL operation, attempts were made to map out the region of stability for various levels of noise and signal dynamics. However, too many variables prevented meaningful results. Variables that impact loop performance include: input sinusoid frequency, VCO rest frequency, K_0, G_1, G_2, N_0, the sampling rate of the simulation, the size of the input signal's phase step and frequency step, the rate of the input signal's frequency ramp, and the rate at which the INTEGRATOR blocks are reset. In general, the loop was able to track input signals at extremely low SNRs (below -30dB), large frequency steps (limited only by the sampling frequency's ability to accurately present the signal to the system), and large frequency ramps (400 Hz/sec). In all cases, K_0, G_1, and G_2 had to be much smaller than predicted.

7.3 RECOMMENDATIONS FOR FURTHER RESEARCH

A determination must be made as to whether or not SPW is adequate for modeling GPS receiver performance. Unless a method is found to simulate the PN source, the DLL as described in this thesis cannot be studied. Further study should include determining the cause of the apparent threshold on the loop gain in both loop simulations. Additional attention should be placed on insuring accurate measurements of the simulator's performance are being made. The method of measuring phase errors as described in this thesis has limited accuracy and may have, in fact, obscured actual loop performance.

GPS will continue to be extremely important for accurate determination of position and velocity for years to come. With U.S. government plans to phase out most other navigational programs within the next two decades, GPS will become the mainstay of not only military, but civilian and commercial navigation and positioning systems as well. Increased reliance on GPS requires maximum performance from user segment receivers. Only continued research will improve the accuracy and performance of GPS receivers enough to meet the demands of military and civilian applications as a sole navigation source.

APPENDIX A: LOOP ERRORS DUE TO MODULATION

Errors in the delay lock loop (DLL) and modified Tanlock Loop (MTLL) approximations of the received signal occur for two primary reasons. The first cause of errors is noise in the received signal. The second cause, and the subject of this appendix, is dynamics of the transmitter and/or receiver causing modulation of the received signal. The maximum expected error for two common signal dynamics, a frequency step and a frequency ramp, will be calculated for both the DLL and MTLL.

A.1 MODULATION ERRORS IN THE DELAY LOCK LOOP

Algebraic manipulation of Equations (13) and (17) in Chapter III results in:

$$\begin{aligned} \frac{\tau_d}{\tau} &= \frac{\tau - \hat{\tau}}{\tau} \\ &= 1 - \frac{\hat{\tau}}{\tau} \\ &= 1 - H(p) \\ &= \frac{p^2}{p^2 + 2K_cG_1p + 2K_cG_2} \end{aligned} \tag{46}$$

so that $\tau_d = \tau[1 - H(p)]$.

A.1.1 DLL CASE A: FREQUENCY STEP. If the received signal experiences an instantaneous change in frequency (a frequency step) $\Delta\omega = \omega - \omega_0$, such that $s(t) = \Delta\omega$, then the input phase can be found by integration, that is, $\theta(t) = \int s(t)dt$. Such a frequency step occurs when the receiver initially attempts to lock onto a carrier experiencing an unknown Doppler shift. The Laplace Transform of s(t) is then $S(s) = \Delta\omega/s$. The Laplace Transform of $\theta(t)$ is $\Theta(s) = S(s)/s = \Delta\omega/s^2$. Dividing by the signal frequency yields the signal delay. Therefore, the Laplace Transform of τ is $\tau(s) = \Delta\omega/(\omega s^2)$. Setting ζ (defined in Equation(18)) equal to .707 results in $G_2 = K_c G_1^2$. Substituting $\tau(s)$ into Equation (46) and replacing p with the Laplace variable s yields:

$$\begin{aligned}\tau_d(s) &= \tau(s)[1 - H(s)] \\ &= \frac{\Delta\omega}{\omega}\,\frac{1}{s^2 + 2K_cG_1s + 2K_c^2G_1^2}\end{aligned} \qquad (47)$$

The Final Value Theorem, i.e. the limit of $s\tau_d(s)$ as $s \to 0$, shows the steady state delay error is zero. The Inverse Laplace Transform of Equation (47) can be found by letting $a = b = K_cG_1$ so that Equation (47) takes the form:

$$\tau_d(s) = \frac{\Delta\omega}{\omega}\,\frac{1}{(s+a)^2 + b^2} \qquad (48)$$

The Inverse Laplace Transform of Equation (48) is then:

$$\begin{aligned}\tau_d(t) &= \frac{\Delta\omega}{\omega}\frac{1}{b}e^{-at}\sin(bt) \\ &= \frac{\Delta\omega}{\omega K_c G_1}\exp\left[-K_c G_1 t\right]\sin(K_c G_1 t)\end{aligned} \tag{49}$$

which takes on a maximum value of:

$$MAX(\tau_d) = \frac{0.322\Delta\omega}{\omega K_c G_1} \qquad at\ time \quad t = \frac{\pi}{4K_c G_1} \tag{50}$$

A.1.2 DLL CASE B: FREQUENCY RAMP. If the input signal experiences a frequency ramp, $\dot{\omega}$, such that the signal's frequency increases linearly with time, then $s(t) = \dot{\omega}t$. Such a ramp could result from acceleration of the GPS receiver platform. The Laplace Transform of $s(t)$ is then $S(s) = \dot{\omega}/s^2$. Because the phase is the integral of the frequency, $\Theta(s) = \dot{\omega}/s^3$. The instantaneous delay is found by dividing the phase by the frequency so that $\tau(s) = \dot{\omega}/(\omega s^3)$. As before, the Laplace Transform of the delay error can be expressed as:

$$\tau_d(s) = \frac{\dot{\omega}}{\omega s}\,\frac{1}{s^2 + 2K_c G_1 s + 2K_c^2 G_1^2} \tag{51}$$

Notice the Final Value Theorem for this case shows the steady state error is non-zero. The steady state error is:

$$\begin{aligned}\tau_{ss} &= \lim_{s\to 0}\,[s\tau_d(s)] \\ &= \frac{\dot{\omega}}{\omega}\,\frac{1}{2K_c^2G_1^2}\end{aligned} \tag{52}$$

Because Equation (51) is similar in form to Equation (47) divided by s, the Inverse Laplace Transform of Equation (51) is similar to the integral of the Inverse Laplace Transform of Equation (47). That is:

$$\begin{aligned}\tau_d(t) &= \int_{-\infty}^{\infty} \frac{\dot{\omega}}{\omega}\frac{1}{b}e^{-at}\sin(bt)\,dt \\ &= \frac{\dot{\omega}\,e^{-K_cG_1t}}{2\omega(K_cG_1)^2}\left[\cos(K_cG_1t) + \sin(K_cG_1t)\right]\end{aligned} \tag{53}$$

which takes on a maximum value of:

$$MAX(\tau_d) = \frac{0.5\,\dot{\omega}}{\omega(K_cG_1)^2} \qquad at \quad t = 0 \tag{54}$$

As Equation (54) indicates, the worst error delay will occur at the onset of the frequency ramp. If the loop remains in lock initially, it will stay in lock. It will not, however, be able to force the steady state delay error to zero. A higher order loop filter would be required to bring the steady state error to zero.

A.2 MODULATION ERRORS IN THE MODIFIED TANLOCK LOOP

Phase errors for the MTLL will be derived in a manner similar to those of the DLL.

A.2.1 MTLL CASE A: FREQUENCY STEP. Let ζ in Equation (39) of Chapter V equal .707. Then, $G_2 = K_0 G_1^2/2$. Equation (28) in Chapter V can be rewritten in the same form as Equation (47), above:

$$\begin{aligned} \Theta_e(s) &= \Theta(s)[1 - H(s)] \\ &= \frac{\Delta\omega}{s^2 + K_0 G_1 s + \dfrac{(K_0 G_1)^2}{2}} \end{aligned} \tag{55}$$

As in the DLL case, the Final Value Theorem proves the steady state phase error for the frequency step case is zero.

The Inverse Laplace Transform of Equation (55) is found by letting a = b = $(K_0G_1)/2$.

$$\begin{aligned}\Theta_e(t) &= \frac{\Delta\omega}{b}e^{-at}\sin(bt)\\ &= \frac{2\Delta\omega}{K_0G_1}\exp\left[-\frac{K_0G_1}{2}t\right]\sin\left(\frac{K_0G_1}{2}t\right)\end{aligned} \tag{56}$$

which takes on a maximum value of:

$$MAX(\Theta_e) = \frac{0.645\,\Delta\omega}{K_0G_1} \qquad \textit{at time} \quad t = \frac{\pi}{2K_0G_1} \tag{57}$$

A.2.2 MTLL CASE B: FREQUENCY RAMP. Similar to the DLL, in the case of a frequency ramp for the MTLL, the phase error in Laplace notation is:

$$\Theta_e(s) = \frac{\dot{\omega}}{s}\,\frac{1}{s^2 + K_0G_1s + \dfrac{(K_0G_1)^2}{2}} \tag{58}$$

Performing integration as in Equation (53) above, we get the phase error function for a frequency ramp:

$$\begin{aligned}\Theta_e(t) &= \int_{-\infty}^{\infty} \frac{\dot{\omega}}{b} e^{-at} \sin(bt)\, dt \\ &= \frac{2\,\dot{\omega}}{(K_0G_1)^2} \exp\left[\frac{-K_0G_1}{2}t\right]\left[\cos\left(\frac{K_0G_1}{2}t\right) + \sin\left(\frac{K_0G_1}{2}t\right)\right]\end{aligned} \qquad \textbf{(59)}$$

The maximum error is:

$$MAX(\Theta_e) = \frac{2\,\dot{\omega}}{(K_0G_1)^2} \qquad at \quad t = 0 \qquad \textbf{(60)}$$

Again we see that if the loop error remains below threshold at the onset of the frequency ramp, the loop will remain locked but the phase error cannot be reduced to zero.

A.3 SUMMARY

This appendix calculated delay and phase errors due to modulation for the DLL and MTLL, respectively. Modulation due to frequency steps and frequency ramps were considered. Table 1 summarizes the results of this appendix.

Table 1. Loop Errors Due to Modulation

Loop	Maximum Error Due to Frequency Step	Maximum Error Due to Frequency Ramp
DLL	$\frac{0.322\,\Delta\omega}{\omega K_c G_1}$ *seconds*	$\frac{0.5\,\dot{\omega}}{\omega (K_c G_1)^2}$ *seconds*
MTLL	$\frac{0.645\,\Delta\omega}{K_0 G_1}$ *radians*	$\frac{2\,\dot{\omega}}{(K_0 G_1)^2}$ *radians*

APPENDIX B: WORST CASE SIGNAL DYNAMICS

B.1 OVERVIEW

Two signal dynamics will be addressed: frequency step and frequency ramp. A frequency step arises from relative motion between the transmitter and receiver and takes the form of a Doppler shift. A frequency ramp results from acceleration of the receiver and/or transmitter.

B.2 FREQUENCY STEP

For the Global Positioning System (GPS), Doppler shifts of the carrier frequency will arise from two primary sources: satellite movement and receiver movement. Worst case Doppler due to satellite movement will occur just as the satellite rises over the horizon and again as it sets. Allowing for a minimum of five degrees above the horizon for reception, a GPS satellite at an altitude of 20,183 km in a 12-hour orbit will have a velocity component in the direction of a stationary receiver of approximately 700 m/s [SPI80].

The equation to calculate the Doppler shift is:

$$\Delta f(t) = \frac{f_c}{c} v(t) \tag{61}$$

where:

$\Delta f(t)$ is the Doppler shift of the carrier frequency in Hz

f_c is the carrier frequency in Hz

c is the speed of light

$v(t)$ is the relative velocity between the transmitter and receiver

Letting $f_c = 1.5$ GHz and $c = 3 \times 10^8$ m/s, the Doppler shift due to the satellite's 700 m/s velocity is 3500 Hz. The Doppler shift will be 3500 Hz as the satellite rises, decrease to zero as the satellite moves directly overhead, and fall to -3500 Hz as the satellite sets. Thus, the Doppler range due to satellite movement is 7000 Hz.

The worst case Doppler shift due to receiver movement will depend on the type of platform containing the receiver. A jet aircraft will experience the worst case Doppler shift when it is traveling directly toward or away from the satellite. Assuming a velocity of 700 m/s (Mach 2.1 at sea level) the worst case Doppler shift will be the same as that due to satellite movement or 3500 Hz.

Therefore, one worst-case Doppler scenario is an aircraft traveling directly toward a satellite that is rising over the horizon, resulting in a Doppler shift of 7000 Hz or $\Delta\omega =$ 44,000 rad/sec.

B.3 FREQUENCY RAMP

The acceleration due to satellite movement can be calculated from the Doppler shift and the satellite's orbit period. The Doppler will vary from 3500 Hz to -3500 Hz in the five hours it is visible to the receiver. This equates to a frequency ramp of .4 Hz/sec.

The acceleration due to receiver movement will, as in the case of Doppler shift, be based on a high performance jet aircraft. A simplified formula to equate acceleration and velocity is:

$$A = \frac{v^2}{r} \tag{62}$$

where: A is acceleration in m/s^2
v is velocity in m/s
r is the turning radius in meters

Thus, an aircraft traveling directly away from a satellite at 700 m/s can reverse direction with an acceleration of nine g's (nine times the acceleration of gravity) with a turning radius of 5550 meters. At 700 m/s, the jet can complete the 180° turn in 25 seconds. This equates to a rate of change of 280 Hz/sec.

Acceleration due to satellite motion is negligible, therefore a worst case acceleration scenario is an aircraft making a maximum acceleration turn either directly toward or directly away from a satellite. The maximum frequency rate of change is 280 Hz/sec or $\dot{\omega} = 1760$ rad/sec^2.

B.4 SUMMARY

This appendix calculated the expected worst case signal dynamics for a GPS receiver onboard a high performance jet aircraft. The jet was assumed to be performing a nine g turn at 700 m/s toward a satellite just rising over the horizon. Table 2 summarizes the findings of this appendix.

Table 2. Worst-Case Signal Dynamics

Signal Dynamic	Effect Due to Aircraft Motion	Effect Due to Satellite Motion	Total Effect
Frequency Step	22,000 rads/sec	22,000 rads/sec	44,000 rads/sec
Frequency Ramp	1760 rads/sec^2	2.5 rads/sec^2	1760 rads/sec^2

Bibliography

[BAL64] Balodis, M. "Laboratory Comparison of TANLOCK and Phaselock Receivers, *Conference Record, National Telemetry Conference, Paper 5-4*: Piscataway, NJ: IRE, 1964.

[CHO87] Cho, W. D. and C. K. Un. "On Improving the Performance of a Digital Tanlock Loop," *Proceedings of the IEEE, 75*: 520-522 (April 1987).

[COO86] Cooper, George R. and Clare D. McGillem. *Modern Communications and Spread Spectrum.* New York: McGraw-Hill Book Company, 1986.

[GAR79] Gardner, Floyd M. *Phaselock Techniques.* New York: John Wiley & Sons, 1979.

[GAU91] Gaudenzi, Riccardo De and Marco Luise. "Decision-Directed Coherent Delay-Lock Tracking Loop for DS-Spread Spectrum Signals," *IEEE Transactions on Communications, 39:* 758-765 (May 1991).

[GIL66] Gill, Walter J. "A Comparison of Binary Delay-Lock Tracking-Loop Implementations," *IEEE Transactions on Aerospace and Electronic Systems, AES-2*: 415-424 (July 1966).

[LEE82] Lee, Jae Chon and Chong Kwan Un. "Performance Analysis of Digital Tanlock Loop," *IEEE Transactions on Communications, 30*: *2398-2411* (October 1982).

[LIN72] Lindsey, William C. *Synchronization Systems in Communication and Control.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1972.

[MAR62] Martin, Benn D. "The Pioneer IV Lunar Probe: A Minimum-Power FM/PM System Design," *Jet Propulsion Laboratory Technical Report No. 32-215.* 15 March 1962.

[OUL84] Ould, Peter C. and Robert J. VanWechel. "All-Digital GPS Receiver Mechanization," *Papers published in Navigation.* 25-35. Washington D.C.: The Institute of Navigation, 1984.

[POM88] Pomalaza-Raez, Carlos A. "Noise Analysis of a Digital Tanlock Loop," *IEEE Transactions on Aerospace and Electronic Systems, 24*: 713-718 (November 1988).

[ROB62] Robinson, L. M. "TANLOCK: A Phase-Lock Loop of Extended Tracking Capability," *Proceedings of IRE Convention on Military Electronics*: 396-427. Piscataway, NJ: IRE, 1962.

[SIM77] Simon, Marvin K., "Noncoherent Pseudonoise Code Tracking Performance of Spread Spectrum Receivers," *IEEE Transactions on Communications, 3:* 327-345 (March 1977).

[SKL88] Sklar, Bernard. *Digital Communications.* Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[SPI61] Spilker, J. J., Jr. and D. T. Magill. "The Delay-Lock Discriminator-An Optimum Tracking Device," *Proceedings of the IRE, 49*: 1403-1416 (September 1961).

[SPI63] Spilker, J. J., Jr. "Delay-Lock Tracking of Binary Signals," *IEEE Transactions on Space Electronics and Telemetry, Set-9* : 1-8 (March 1963).

[SPI80] Spilker, J. J., Jr. "GPS Signal Structure and Performance Characteristics," *Papers Published in Navigation, Vol 1.* Washington D.C.: Institute of Navigation, 1980.

[YOS80] Yost, Richard A. and Robert W. Boyd. "A Modified PN Code Tracking Loop: Its Performance and Implementation Sensitivities," *Proceedings National Tele-Communications Conference*: 61.5-1 - 61.5-5. Houston, Tx: IEEE, December 1980.

[YOS82] Yost, Richard A. and Robert W. Boyd. "A Modified PN Code Tracking Loop: Its Performance Analysis and Comparative Analysis," *IEEE Transactions on Communications COM-30*: 1027-1036 (May 1982).

Vita

Captain James A. Hird was born in Farley, Iowa on November 8, 1962. He graduated from Western Dubuque High School in 1981 and entered the United States Air Force Academy. He graduated from the Air Force Academy in 1985 with a Bachelor of Science Degree in Electrical Engineering and a commission as a Second Lieutenant in the U.S. Air Force. He spent six years assigned to Germany, first at the European Communications Division in Kaiserslautern, where he worked on the design, procurement, and installation of a secure wing-level command and control computer network for air bases throughout Europe. In 1987, he transferred to the 1836th Engineering Installation Group in Wiesbaden, where he engineered the installation and testing of numerous command, control, and intelligence communications-computer systems throughout Europe and the Middle East. In 1991, he was assigned to the Air Force Institute of Technology to earn his Master of Science Degree in Electrical Engineering.

Permanent Address: 107 6th Ave N.W.
Farley, IA 52046

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1992

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
ANALYSIS AND SIMULATION OF MODIFIED
TANLOCK AND DELAY LOCK LOOPS
FOR GPS RECEIVER DESIGN

5. FUNDING NUMBERS

6. AUTHOR(S)
James A. Hird

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/92D-20

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Lt Col Wiernle
6585th Test Group
Guidance Test Division
Holloman AFB, NM 88330-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this thesis is to investigate two types of tracking loops used in Global Positioning System (GPS) receiver design. The first loop, the Delay Lock Loop (DLL), is a code tracking loop used to synchronize a locally generated pseudo-noise (PN) sequence with the PN sequence in the GPS satellite broadcast. Synchronization of the PN sequences is essential for de-spreading the direct-sequence spread spectrum (DS/SS) broadcast and demodulating the transmitted data. The second loop, the Modified Tanlock Loop (MTLL), is a carrier tracking loop used to synchronize the phase of a voltage controlled oscillator (VCO) with the carrier of the GPS satellite broadcast. Carrier synchronization is essential for optimum data demodulation. This thesis derives equations predicting the theoretical performance of each loops' ability to track a GPS signal corrupted by noise and signal dynamics arising from transmitter and/or receiver motion. In addition, computer simulations of the DLL and MTLL were developed and the results are presented. The simulations display phenomena which were not present in the theoretical predictions.

14. SUBJECT TERMS
GPS, Spread Spectrum, Delay Lock Loop, Tanlock Loop, Tracking, Synchronization

15. NUMBER OF PAGES
97

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL