TRACKING OF MULTIPLE SINUSOIDS USING COUPLED PHASE-LOCKED LOOPS

A THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Michael Stockmaster, B.S.E.E

* * * * *

The Ohio State University

1995

Master’s Examination Committee:
Steven B. Biblyk
Lee C. Potter

Approved by

[Signature]
Adviser
Department of Electrical Engineering
To my wife, Sheri, for her love and support, and for her understanding throughout these last 5 years.
ACKNOWLEDGEMENTS

I would like to thank Dr. Steven Bibly for his support and guidance throughout this past year, as well as Dr. Lee Potter for his helpful comments and time in serving on my committee. I also wish to thank graduate student Yuwen Zou for her help and knowledge in the area of VLSI and everyone at Ohio State for making my stay a pleasant experience.
VITA

October 6, 1971 ......................... Born - Sandusky, Ohio, U.S.A.

1994 ................................. B.S. Electrical Engineering
    Summa Cum Laude
    University of Toledo
    Toledo, Ohio, U.S.A.

1994-present .......................... Palace Knight Participant
    The Ohio State University
    Columbus, Ohio, U.S.A.

Fields of Study

Major Field: Electrical Engineering

Studies in: Signal Processing, Communications, VLSI
# Table of Contents

DEDICATION .................................................................................. ii

ACKNOWLEDGEMENTS ................................................................. iii

VITA .............................................................................................. iv

LIST OF FIGURES ........................................................................ vii

CHAPTER PAGE

I Introduction .............................................................................. 1
  1.1 Overview of Thesis ............................................................... 2
  1.2 Methods for Sinusoidal Tracking .......................................... 4
  1.3 History of CCPLL's ............................................................... 6
  1.4 A Brief Look at CCPLL Operation ....................................... 7
  1.5 Reasons for Using a CCPLL ................................................ 8

II Technical Background ............................................................. 12
  2.1 Brief Overview of PLL Usage .............................................. 12
  2.2 Description of the Phase-locked Loop .................................. 13
  2.3 Design Parameters ............................................................. 18
  2.4 Noise Theory and Optimality .............................................. 20
    2.4.1 Derivation of Maximum Likelihood Optimality ............... 20
    2.4.2 Optimizing the Loop Parameters ................................... 23
  2.5 Summary ............................................................................. 26

III Simulator Description ............................................................. 27
IV Phase-locked Loops: Angle Demodulation Simulation and Discussion  

4.1 Introduction to Angle Demodulation .............................................. 30  
4.1.1 Digital Communications Example: FSK Demodulator .................. 31  
4.1.2 Analog Communications Examples: AM and PM Demodulation ........ 35  
4.2 Simulations in the Presence of Noise ........................................... 40  
4.3 Simulation Conclusions ............................................................... 41  

V Simulation and Discussion for Coupled Phase-locked Loops ................. 43  

5.1 Optimality of CCPLL ................................................................. 43  
5.2 CCPLL Implementations ............................................................ 46  
5.2.1 CCPLL Without Amplitude Control ......................................... 46  
5.2.2 CCPLL With Amplitude Control ............................................. 47  
5.3 CCPLL Simulations: FM signals .................................................. 53  
5.3.1 Noiseless Simulation ............................................................. 53  
5.3.2 Simulation in Presence of AWGN ........................................... 58  
5.4 Proposed CCPLL Performance Measure ....................................... 60  
5.5 CCPLL and Phase Demodulation .................................................. 63  
5.6 Summary .................................................................................... 70  

VI Steps Toward VLSI Implementation ................................................. 71  

6.1 Proposed VLSI CCPLL- Implementation with Limiter .................... 76  
6.2 Proposed VLSI CCPLL- Implementation without Limiter ................ 79  
6.3 Summary .................................................................................... 83  

VII Conclusions ................................................................................. 84  

7.1 Summary of Work ........................................................................ 84  
7.2 Summary of Conclusions ............................................................. 85  
7.3 Areas of Future Research ............................................................ 87  

BIBLIOGRAPHY .................................................................................. 88
# List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functional diagram of a coupled PLL system</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Block diagram of a coupled PLL system</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Block diagram of basic PLL</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Block diagram of FSK demodulator</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>Modulating waveform for FSK and the demodulated output</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>Block diagram for demodulating 50Hz sinusoidal FM</td>
<td>35</td>
</tr>
<tr>
<td>7</td>
<td>Demodulated 50 Hz sinusoidal FM signal</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>Demodulated 50 Hz sinusoidal PM signal</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>Demodulated 50 Hz triangular PM signal</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>Demodulated 50 Hz triangular PM signal, demodulated as an FM square wave</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>Demodulated waveforms for a noisy input signal with no pre-filtering with SNR’s -13, -23, -33, and -43 dB respectively, moving CW from upper left.</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>Block diagram of CCPLL system with amplitude control</td>
<td>48</td>
</tr>
<tr>
<td>13</td>
<td>Amplitude control portion of circuit</td>
<td>48</td>
</tr>
<tr>
<td>Page</td>
<td>Section</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Block diagram of a coupled PLL system</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Demodulated FM outputs for a CCPLL system</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Block diagram of CCPLL with amplitude control, FM demodulator</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Demodulated FM outputs, amplitude control, SNR=-10dB</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Cross-channel interference ratio characteristics for identical PLL’s with FM input signals</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Partial diagram of CCPLL for PM derivation</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Simulation results of CCPLL, equal amplitude, 8 and 10kHz carriers</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Cross-channel interference ration for PM CCPLL</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CMOS current starved inverter</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Digital PLL to be implemented in VLSI technology</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Block diagram of CCPLL system in form to be realized in VLSI</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Block diagram of CCPLL system in form to be realized in VLSI, another implementation</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER I

Introduction

A sinusoidal signal in its most general form is given by $s(t) = A(t) \exp^{j\Phi(t)}$, where $A(t)$ is a (possibly) time varying amplitude and $\Phi(t)$ is a (possibly) time varying phase. When speaking of multiple sinusoids, it is meant as the superposition of sinusoidal signals. These signals may take the form of a radar signal, a digital communication signal (e.g. BPSK), an analog communications signal (e.g. FM), or one of other forms. In many cases, sinusoids may take the form of a real signal, a sine or cosine. The real sinusoid is the form considered throughout this thesis.

The tracking of multiple sinusoidal signals is a situation that arises frequently in engineering practice. The need to track multiple sinusoids is found in such areas as the tracking of radar signals. There may be more than one radar signal of interest occurring simultaneously in time, such as the radar return signals from two aircraft. In digital communications, many signals of various forms are transmitted simultaneously in time, one or more of which may be of interest to a particular receiver. It would be beneficial for the receiver to be able to track the desired signals while cancelling any interfering signals. If a desired signal is corrupted by co-channel or adjacent channel interference, the interference can be tracked as a means of suppressing the
interference, cancelling it out at the input where tracking of the desired signal is to take place. For example, in the field of analog communications, one may wish to demodulate an FM signal while cancelling out the interfering signal in the same frequency band generated from a neighboring city. The FM signal as well as the interferers can be tracked by the receiver so the interfering signals may be cancelled.

1.1 Overview of Thesis

A method of tracking multiple sinusoidal signals that allows for interference cancellation, as in the FM interference case described above, is the cross-coupled phase-locked loop (CCPLL). In its simplest form, the CCPLL is a system of two phase-locked loops that are connected in such a way to allow for tracking of multiple input signals while cancelling interfering signals at the input to each PLL. In this thesis, the CCPLL method for tracking multiple sinusoids will be analyzed and simulated in the two forms presented in the literature [1, 2]. The first form assumes knowledge of the input signal amplitudes while the second does not. The importance of the amplitudes in CCPLL functionality will become evident in Chapter V. Both forms of the CCPLL will be simulated and described. The CCPLL with no amplitude control will give a baseline for comparison with the CCPLL with amplitude control, which will be found as the more practical CCPLL implementation.

This thesis demonstrates that the CCPLL, in the phase estimation context, yields the approximate maximum likelihood estimates (MLEs) of the input signal phases when the phases are constant or slowly varying with time. It has previously been
shown that a single PLL is able to estimate a constant phase with MLE optimality [3]. Simulations are performed on single phase-locked loops (PLLs) and CCPLLs with input signals that are known to be readily tracked by PLLs. This is a means to determine the suitability of Matlab's Simulink Toolbox as a method for simulating PLLs. The suitability of Simulink is established, and a novel measure of CCPLL performance, the cross channel interference ratio, is introduced. This ratio is a measure of how well the CCPLL is able to cancel the interfering signals. Simulations are performed to determine this ratio for both phase and frequency modulated signals and comparisons made on the performance for each case. PLLs have been widely used for angle demodulation [4]. A phase demodulated output may be extracted from two different nodes in a PLL, with each node the desired output under certain conditions. When the modulation index is small and the PLL operates as a phase demodulator, the output may be extracted before the loop filter (the components of the PLL will be described in Chapter II). The conditions for which the CCPLL operates as a multiple signal phase demodulator with the above mentioned output extraction node are derived. Simulations are then performed which verify the derived conditions. Also in this thesis, two CCPLL systems are proposed which lend themselves to implementation in CMOS VLSI. Implementation in VLSI is desired as the CCPLL may be implemented cheaply and compactly while maintaining the same functionality as those CCPLLs previously suggested in [1, 2]. Modern VLSI technology shows promise that it can allow the CCPLL to be functional even when the individual components cannot be manufactured to close tolerances.
An outline of this thesis follows. Chapter I describes the sinusoidal tracking problem and current methods for solving the problem. A model for the CCPLL as a phase estimator is introduced and a description of CCPLL operation is given. Chapter II gives basic background information helpful in understanding the PLL, including the PLL components and their relation to design parameters. Optimality issues are addressed for operation in additive white Gaussian noise. Chapter III gives a description of Matlab's Simulink Toolbox, the simulation tool used throughout this thesis to simulate the CCPLL. Chapter IV involves the design of single PLL circuits in the angle demodulation context, verifies Simulink's ability to simulate PLLs, and demonstrate the performance of a PLL in a noisy environment. Chapter V describes both implementations (amplitude control and without) of the CCPLL, presents and describes the cross channel interference ratio, and provides simulation results of the CCPLL. Conditions are derived for which the CCPLL functions as a multiple signal phase demodulator. It is also demonstrated that approximate MLE optimality of the CCPLL is realized for a constant or slowly varying phase. Chapter VI proposes and describes two novel implementations of the CCPLL suitable for manufacture in VLSI and Chapter VII offers summaries and conclusions from the previous chapters.

1.2 Methods for Sinusoidal Tracking

Various methods outside of the CCPLL method have been or are being used to track multiple sinusoids. One such method is the use of a bank of band pass filters. However, the band pass filter method has problems when the center frequency of the signal(s)
being tracked is not constant or is unknown. Other methods have adapted this idea and use a bank of adaptive filters to take into account the possibility of a moving center frequency to allow for tracking of non-stationary signals. In [5], such a method is described and implemented in CMOS technology. Another method for multitone tracking is described in [6] and uses extended Kalman filters. Other methods of tracking multiple sinusoids have been developed and are fairly mature [7]. Many of these use a digital computer for implementation. These include FFT based methods and SVD based methods. However, one method for tracking multiple sinusoids has been relatively unexplored: using cross coupled phase-locked loops (CCPLLs).

One way of thinking of the CCPLL is the estimator model shown in Figure 1. Each function block shows the corresponding phase-locked loop (PLL) component for that particular function. A comparison of the CCPLL with the more traditional estimation methods will now be made. Many FFT or SVD based methods estimate parameters and generate an output based on an assumed signal model [7]. These methods sample the inputs and form estimates based on the samples. The CCPLL also makes a model based estimate of the signal parameters, but then compares the estimates with the parameters of the input signal. This comparison is implemented by way of a feedback loop. The information resulting from the comparison is then used to improve the estimation. The CCPLL, as previously discussed in the literature [1, 2, 8, 9] and in this thesis, implements a continuous time system which estimates and improves the signal parameters.
1.3 History of CCPLL’s

The CCPLL method was first proposed as a means of suppressing interference in FM demodulation [1]. It was demonstrated that repeating a cross-coupled PLL structure, two (and later three) separate FM signals could be tracked and demodulated. One of the two signals was considered the desired signal, while the other signal was considered interference. A receiver could be constructed using two interconnected phase-locked loops with approximate maximum a posteriori MAP optimality (when a Gaussian density was assumed on the phases) in the phase estimation. One PLL would track the stronger of the two received signals by the capture effect (an effect in which a PLL will track the strongest signal present at its input and ignore weaker signals) while the other weaker signal is left for the opposite PLL to track. Even if the interfering FM signal was co-channel with the desired signal, the interfering FM signal could be tracked and even be demodulated. This method, however, as-
sumes that the amplitudes of the input signals are known, an assumption which is usually undesirable and unrealistic. Later, a similar system was devised that generated amplitude estimates of the incoming signals to fully automate the receiver [2]. Amplitude and phase estimates were obtained simultaneously. In [9], the separation of closely spaced, equal amplitude FM signals has been demonstrated without the estimation of signal parameters. Under certain assumptions, it is possible to know in advance which PLL will track which signal. In [8], it has been demonstrated that the CCPLL can track multiple narrow band sources through the use of a passive array. This type of narrowband source tracking is used in such areas as radar, sonar, radio astronomy, and seismology. Other possible applications include recovering a weak or distant signal in a noisy environment, or even detecting independent modulations on a single carrier. CCPLLs could be used to track a frequency hopped signal in the presence of a jamming tone [5].

1.4 A Brief Look at CCPLL Operation

To briefly describe the idea behind the operation of a cross-coupled phase locked loop system, consider the case of two coupled phase-locked loops in the presence of two unity amplitude input signals of differing frequencies and phases. We would like the first PLL to track one of two signals while inhibiting the second from tracking the same signal, and vice-versa. The principle in inhibiting the other PLL to track the same signal is rather simple. In tracking a signal, it is well known that the VCO of a PLL generates an estimate of a signal that is 90 degrees out of phase with the
incoming signal once lock is achieved [4, 10]. If the VCO output is phase shifted another 90 degrees and is of unity amplitude, we have a signal that is an estimate of the desired signal, only negative in sign. If one of the phase locked loops is able to start tracking one of the signals, a cancellation can be forced at the opposite phase locked loop. Since it is undesirable to have the other PLL track the same signal as the first, we can add the phase shifted VCO output with the proper amplitude to the input of the second PLL, and cancel the first signal. Then the input of the second PLL is ideally the second signal alone. In a similar fashion, the first PLL can be inhibited from tracking the second signal after the second PLL starts tracking its signal. This idea can logically be extended to a bank of M coupled PLL’s. More details of CCPLL operation will be discussed in Chapter V.

1.5 Reasons for Using a CCPLL

A basic block diagram of a CCPLL system is shown in Figure 2. A CCPLL loop is able to simultaneously track, enhance, and demodulate multiple signals using a pair of phase-locked loops connected in such a way that each PLL tracks an individual signal and inhibits the other from tracking the same signal. The idea can be extended to more than two PLL’s connected together so that each PLL tracks one signal, and inhibits the others from tracking that same signal. One main advantage of the CCPLL is that it can separate sinusoidal components without explicit estimation of its parameters. The CCPLL takes advantage of the individual PLL’s ability to automatically track a signal’s frequency and phase. This claim is first made in
[1] with the assumption that the input signal amplitudes are known, and utilizes a novel approach to tracking multiple sinusoids. Other work has been done in the area of lifting the unrealistic restriction of knowing the amplitude of the signals [2, 11]. Instead of estimating all parameters (phase, frequency, and amplitude), it has been shown that a CCPLL can provide amplitude estimates while at the same time continuously tracking the desired signals. This allows the CCPLL to track signals with a time varying amplitude and make the proper cancellations within the CCPLL structure, a point which will be discussed later.
Another benefit of using CCPLL’s is the potential for fast convergence, which is particularly important when only a short data length or short time interval is available. In certain instances, the CCPLL is able to track the phases of the input signals over a short interval and without any bias [9]. CCPLLs can also track closely spaced sinusoids, again a situation where other methods break down due to short data lengths or real time constraints. CCPLLs do not require the computational expense of eigenstructure methods and continuous tracking is possible. Where other methods use two steps to estimate the sinusoids (one to estimate the frequency and one to estimated the phase and amplitude), it has been shown that a CCPLL can simultaneously estimate frequency, phase, and amplitude. With a CCPLL, two different information carrying signals at the same carrier frequency can be tracked and the information signals extracted [1, 2]. When sinusoids are closely spaced in frequency, eigenstructure or FFT based methods could fail to resolve both signals. When two sinusoids exist that are identical in frequency, the resolution of the two signals is impossible with FFT or eigenstructure based methods. CCPLLs can also separate out sinusoids at high frequencies, even in the RF range, where other methods fail because of the high sampling rates required [9]. The CCPLL is able to take advantage of the individual PLL’s narrow bandwidth, which is beneficial in rejection of noise. The theory of phase locked loops is relatively well developed [12], another fact which may be exploited in developing CCPLLs.

CCPLLs may take advantage of advances in VLSI technology to be built both cheaply and in a small area. At lower frequencies, phase-locked loops can be designed
digitally, making them ideal for implementation in VLSI. The analog PLL can also be adapted to be implemented in VLSI technology. In this thesis, two CCPLL systems will be proposed that lend themselves to VLSI implementation, one implemented with digital components, and the other mixed analog and digital components. Enhancements are proposed that utilize adaptive filters to bring together coupled PLL theory and adaptive filter theory. Not only does the CCPLL track and cancel multiple signals, but adaptive filters can be used to enhance performance further. This will be discussed in chapter VI.

So far, CCPLLs have been used to demodulate FM signals, cancel interfering signals, and track multiple targets. However, the potential for using CCPLLs in other applications exists. Any signal that can be modeled as sum of (complex) exponentials could take advantage of CCPLLs to separate the signal into its individual components, even if the components are closely spaced and narrow band. This could have a wide range of applications in communications and elsewhere as this model arises frequently. The CCPLL approach could be implemented cost effectively, especially if VLSI implementations of the PLL systems are used.
CHAPTER II

Technical Background

This chapter gives background information on PLL's helpful in understanding the operation of the CCPLL. It describes the operation of the PLL, identifies its components, and shows how choice of components affects the performance of the PLL. This chapter also touches on the effects of noise on PLL operation.

2.1 Brief Overview of PLL Usage

A PLL is a well known device whose utility in uncoupled form has been demonstrated over the years. It had its first use (and also some of its latest uses) in the area of receivers, most recently to recover signals which are deeply embedded in noise. The PLL is widely used as a frequency modulation (FM) or phase modulation (PM) demodulator. The PLL is also widely used in televisions as a method of synchronization, and in digital communication for the same purpose. It is used to achieve synchronization in point to point transmission, in the form of carrier, clock, and word synchronization [13]. Another widespread use of PLLs is in frequency synthesis.
2.2 Description of the Phase-locked Loop

In its simplest form, a PLL consists of three main blocks which are interconnected to form a feedback control system, as shown in Figure 3.

\[ y(t) = A \sin(\omega t + \theta_i) \]
\[ v(t) = B \cos(\omega t + \theta_o) \]

**Figure 3:** Block diagram of basic PLL

In this section, the operation of each block is considered. The first of these blocks is a phase detector, which has as inputs both the input signal and the output of a voltage controlled oscillator. The output of the phase detector is a signal which is a function of the phase difference between the two input signals, and depends on the type of phase detector used. The phase detector output provides a measure of the phase error between the input and the locally generated estimate. There is usually a gain associated with the phase detector and is generally given in units of radians per volt. An important aspect of the phase detector to be considered is the range over which its output is a linear function of the phase difference between the input
and the locally generated estimate. Different types of phase detectors have varying ranges of linearity such as $\pi$ or $2\pi$ radians. The analog multiplier phase detector has a linear characteristic when the phase difference between the input signal and locally generated estimate of the input signal is small, and is otherwise a sinusoidal function.

The phase detector that we will concentrate on in this thesis is the analog multiplier. This is done for a two reasons. First, the analog multiplier is easy to analyze and simulate. Second, it is in the multiplier type class of phase detectors. Although a true analog multiplier is rarely used in practice due to high cost and low frequency range of operation, modifications can be made which make it practical while maintaining the same functionality. A widely used implementation is the switching phase detector [13]. The VCO, the output of which is normally sinusoidal, is replaced by a VCO generating a unit amplitude square wave. When expanded into its Fourier series, the square wave has the largest component at the original sinusoidal VCO frequency, and progressively weaker components at odd multiples of this frequency. If this signal is one of the inputs to an ideal multiplier, it has the same functionality as multiplying by a sinusoidal VCO at the original frequency, as weaker higher order terms will be filtered out anyway. It is noticed, however, that multiplying by a unit amplitude square wave is equivalent to switching the polarity of the input signal. Therefore, the expensive multiplier can be replaced by a simple switch and still maintain the same functionality. The same functionality achieved by the switch, as in the case of the true analog multiplier, allows the simulation of the analog multiplier to be valid for practical implementations of the phase detector as well.
The second block of the PLL is the VCO, or voltage controlled oscillator. It is a device for which the output is a periodic waveform, generally a sinusoid, at a given frequency and phase. The frequency and phase are determined by the input control voltage to the VCO. The VCO has a gain associated with it given in units of Hz or radians per volt that determines how sensitive it is to the input voltage. It is seen from Figure 3 that the control voltage, $v(t)$, is a filtered version of the phase error signal, $e(t)$.

The filtering operation on the error signal is performed by the loop filter, the third block of the PLL. This filtering operation is generally low pass to remove double frequency components resulting from the phase detector multiplication. The loop filter is very important in determining the operation of the loop. It is mainly responsible for determining the bandwidth and operating characteristics of a PLL. The effect of the loop filter will become evident in chapter IV in the design examples given.

Next, the basic operation of the PLL will be described. Several authors have covered the fundamentals of phase-locked loops [4, 10], etc. The interested reader is referred to these texts for a more complete description of the PLL and its operation. To see clearly the operation of the phase-locked loop, a linear model is assumed. When the phase difference between the input signal and the VCO signal is small enough so we can approximate the phase detector's sinusoidal characteristic, $\sin[\theta_i(t) - \theta_{vco}(t)]$, as $\theta_i(t) - \theta_{vco}(t)$, the linear model is valid. A linear model is generally a reasonable assumption, as when the loop is tracking the signal properly, this condition will be met. Although the linear model is valid when the PLL is working as intended, it may
not be when the PLL is first introduced to a signal. There is generally a larger phase difference than that which would justify use of the linear model because there is a large difference between the input signal and the VCO signal. The VCO signal must be modified by some means so that it matches the input signal frequency. Hence, the PLL first needs to “acquire” the desired signal. Usually, some form of device is used to aid in the acquisition of the signal, as this is a slow process [4]. Acquisition aids can be used to help the PLL in locking on to a signal in situations where locking on to a signal is otherwise too slow or impossible. In this thesis, the effects of acquisition are ignored and it is assumed the difference in frequency between the VCO and the input signal is small enough that acquisition can be ignored. This is done so the simulations can concentrate on the cancellation effects of a CCPLL and be of reasonable length.

The sin[\theta_i(t) - \theta_{vco}(t)] term mentioned above will exist at the output of the phase detector, as well as a double frequency term, in the case where the phase detector is an analog multiplier. It is assumed that the double frequency terms will be removed by the loop filter, leaving only a phase difference component, e(t). It is also possible to remove the double frequency term with post-filtering, which is generally used when the PLL is used for some form of demodulation. The choice of a loop filter then determines how e(t) will be used in the loop. If the loop filter is such that the overall PLL bandwidth is low relative to the input signal modulation and an angle modulated signal is the input to the PLL, e(t) may be taken as a demodulated phase output. This is referred to as a carrier tracking loop as ideally none of the phase modulation is passed to the VCO. The output of the VCO is pure carrier and no
attempt is made to match it to the input signal's phase. If the angle modulation is FM and the PLL bandwidth is wide relative to the FM modulation, the filtered version of $e(t)$ is taken to be the demodulated FM output. This is referred to as a modulation tracking loop as the modulation is passed to the VCO and the VCO output tracks both the phase and frequency of the input signal.

The only assumption made about the frequency content of the input signal phase in any mode of operation is that the change in the input phase is much slower than the carrier frequency. If this is indeed the case, then filtering operations described earlier can remove unwanted high-frequency components while allowing the essential components to pass.

The basic principle of the overall feedback network is that the filtered phase difference signal is low frequency and causes the VCO to increase or decrease its frequency of oscillation based on the sign of the filtered phase difference. When the phase difference becomes larger, the input to the VCO becomes larger and it oscillates faster. When the phase difference becomes smaller, the input to the VCO becomes smaller and it oscillates slower. This causes the VCO to more closely track the input frequency so the phase difference becomes smaller. When the loop is exactly tracking the input (the phase difference is zero), the loop is said to be in lock.
2.3 Design Parameters

This section will describe how the components of the PLL affect its operation. Characteristics such as tracking ability and stability will be discussed, as well as the relation of the loop parameters to standard feedback control parameters.

Since a PLL is a feedback system, it can be analyzed with feedback control system theory. Various parameters such as damping constant, natural frequency, rise time, etc. can be used and their influence on PLL operation determined. Stability of the PLL is determined from these parameters. For the following discussion, it is assumed that the loop can be modeled as linear (the phase detector output is directly proportional to the phase difference). A baseband model for the PLL in terms of the various phases is used in the choice of parameters, since the ultimate goal is to track the phase.

The order of the PLL is determined by adding one to the order of the loop filter. The VCO inherently has a zero at s=0 in its transfer function (in the baseband model), so any phase-locked loop is at least first order. This explains the addition of one. Some of the popular loop filters include a one pole low pass filter, a one pole low pass filter with phase-lead correction, and an integrator with phase-lead correction. Therefore, PLLs using these types of filters are considered second order PLLs. The loop filter can be omitted to realize a first order loop, but higher order loops are used for various reasons, including the removal of double frequency terms generated by the phase detector. Using a second order loop gives the designer more control over loop parameters than in a first order loop. This is also true among the class
of second order loops. A second order loop consisting of a low-pass filter with phase correction (effectively adding a zero to the original low pass filter transfer function) allows one to independently choose important loop parameters (e.g. natural frequency and damping factor), while a second order loop with a low pass filter as the loop filter does not.

System stability is also an issue. Filters are often chosen to provide more phase margin to ensure stability when loop parameters are changed by noise or other disturbances. While a first order loop is always stable, a second order filter is not. Choices of the loop filter can be made to allow for more phase margin to aid in stability in the case where the loop filter is not omitted. A second order loop with low pass filter and lag-lead correction allows more phase margin than a second order loop of a low pass filter only. Allowing more phase margin helps ensure the stability of the PLL as a result of transient inputs or other unpredictable occurrences.

One can also talk about the type of phase-locked loop. Type is the number of integrators (or poles at the origin) in the open loop transfer function. Type determines the accuracy with which a PLL can track a designated change in phase, if it can track it at all. For example, a type 1 loop can track a phase step with no error and a frequency step with finite error, just as found in control system theory. A type 2 loop can track a frequency step (phase ramp) with no error and a frequency ramp (phase quadratic) with finite error. A higher type loop can obviously track a more dynamic signal with less error than a lower type loop. Since type directly affects the loop order, one has to be careful in designing higher order loops as they are no
longer unconditionally stable as a first order loop would be. A second order loop is not necessarily a type 2 loop as the loop filter may be a low pass filter instead of an integrator.

2.4 Noise Theory and Optimality

The previous discussion centered on the nomenclature and the design of a PLL in a noise free environment. However, a more realistic situation is that when the phase-locked loop is operating in the presence of noise. We would like to find that the PLL is optimum in some sense in tracking a signal’s phase. The PLL is usually not used as a generally optimum device, but it can show optimality in certain cases. One such case in the estimation of a constant or slowly varying phase. The maximum likelihood estimate of such a phase will be derived.

2.4.1 Derivation of Maximum Likelihood Optimality

This derivation is adapted from [3]. Assume we have an input to the PLL of the form:

\[ y(t) = A \cos(\omega_0 t + \theta) + n(t) \quad 0 \leq t \leq T \quad (2.1) \]

where \( n(t) \) is additive white Gaussian noise with power spectral density \( \frac{N_0}{2} \) and \( T \) is the observation interval. The phase \( \theta \) is considered constant or nearly constant over each time interval, so it can be a slowly varying function of time. We can represent the \( A \cos(\omega_0 t + \theta) \) through a trigonometric expansion by:

\[ A \cos(\omega_0 t) \cos(\theta) - A \sin(\omega_0 t) \sin(\theta) \quad (2.2) \]
A set of orthonormal basis functions for this expansion is:

\[ \phi_1(t) = \sqrt{\frac{2}{T}} \cos(\omega_0 t), \quad 0 \leq t \leq T \] (2.3)

\[ \phi_2(t) = \sqrt{\frac{2}{T}} \sin(\omega_0 t), \quad 0 \leq t \leq T \] (2.4)

The estimate of the phase can be based on the vector \((Z_1, Z_2)\) where \(Z_i = \int_0^T y(t)\phi_i(t)dt\). We can find the joint density of \((Z_1, Z_2)\) rather easily since they are independent and Gaussian. The joint density of this vector given \(\theta\) is:

\[ f_{Z|\theta}(z_1, z_2|\theta) = C \exp \left[ (z_1 \cos(\theta) - z_2 \sin(\theta)) \right] \] (2.5)

where \(C\) contains every term that does not depend on \(\theta\). Then to set up the likelihood equation to find the maximum likelihood estimate of the phase, we take the natural log and then differentiate 2.5 to find:

\[-Z_1 \sin(\theta) - Z_2 \cos(\theta)|_{\theta=\hat{\theta}_{ML}} = 0 \] (2.6)

Then writing \(Z_1\) and \(Z_2\) in terms of their defining integrals and using trigonometric identities, we find that the maximum likelihood estimate of the signal’s phase is found from the equation:

\[ \int_0^T y(t) \sin(\omega_0 t + \hat{\theta}_{ML})dt = 0 \] (2.7)

This equation is implemented in a feedback system which consists of a phase-locked loop with the loop filter replaced by an integrator. The output of the integrator goes to zero when the loop is locked (for \(\theta\) constant) and hence the equation is solved by the loop. Notice that 2.7 can be implemented in a feedback structure which has the
same functionality of a phase-locked loop. Hence, the PLL is a device which can be interpreted as forcing a solution to the maximum likelihood equation [3, 14].

It is possible to put a lower bound on the variance of the phase estimate using the Cramer-Rao lower bound as the phase estimate in this case is unbiased [3]. This is beneficial as it tells us the best we can do in a given situation. After differentiating the likelihood equation twice we have:

\[
\frac{\delta^2 \log f_{Z|Y}}{\delta \theta^2} = \sqrt{2T} \frac{A}{N_0} (-Z_1 \sin \theta - Z_2 \cos \theta)
\]  

(2.8)

Substituting into the Cramer-Rao inequality yields:

\[
\text{var} \left[ \hat{\theta}_{ML}(Z) \right] \geq \frac{1}{\sqrt{2T}} \frac{N_0}{A} \left( E[Z_1] \cos \theta - E[Z_2] \sin \theta \right)^{-1}
\]

(2.9)

where \( E \) is the expectation operator. After finding the expectations and substituting them into 2.9, we have:

\[
\text{var} \left[ \hat{\theta}_{ML}(Z) \right] \geq \frac{N_0}{A^2 T} = \frac{N_0 B_L}{P_s}
\]

(2.10)

where \( B_L \) is the equivalent noise bandwidth (given by \( B_L = \int_0^\infty |H(j2\pi f)|^2 df \) ) of the estimator (PLL) and \( P_s = \frac{1}{2} A^2 \) is the average signal power. The bound improves as the signal to noise ratio increases and as the equivalent noise bandwidth is decreased.

Following the various texts [3, 4, 10], the variance of the phase estimate is found to be:

\[
\text{var} \left[ \hat{\theta}_{ML}(Z) \right] = \frac{N_0}{A^2 T} = \frac{N_0 B_L}{P_s}
\]

(2.11)

Hence we see that for the case of a constant or slowly varying phase that the Cramer-Rao lower bound is achieved. If the phase does not meet this restriction, the PLL is
still a useful device. The feedback action tries to force the error in the estimate to zero, regardless of the input phase. If the loop bandwidth is large enough, the loop will be able to track the phase so as to force the error to zero. The purpose of the loop is to make the phase error as small as possible. If minimum variance in the phase estimate is the sole criterion in arriving at a phase estimate, other methods would be more suited to achieving the desired optimality.

2.4.2 Optimizing the Loop Parameters

In the more general case of a time varying phase, it is possible to find the Wiener optimum filter to minimize the mean squared error of the phase estimate. This optimization can be done for a given input signal type (i.e., a phase step or frequency step) and signal to noise ratio. However, the solution may not lend itself to feasible implementation of a PLL. The loop filter and other parameters are chosen so the loop is achieving the desired minimum mean squared error estimate of the signal phase. The following noise analysis parallels that in [10], [4], and [14].

The parameters of the loop can be chosen to optimize the loop performance. The transfer function we need to optimize is that of ratio of the VCO phase to the input signal phase, which is seen from the linear model. The loop parameters directly affect this transfer function. We can think of the PLL as a filter with the input being the input phase corrupted by noise and the output being a cleaned up version of the phase. We choose to optimize the transfer function (with respect to minimum mean squared error \( E[(\phi_o(t) - \phi_i(t))^2] \)) and use the Wiener filter as the optimum filter, \( H(jw) \). This is the optimum causal filter found by solving the Wiener-Hopf equation.
The Wiener filter is given by

\[ H(j\omega) = 1 - \frac{\sqrt{\eta}}{\Psi(j\omega)} \]  \hspace{1cm} (2.12)

where \( \eta \) is the power spectral density of the noise. \( \Psi(j\omega) \) is found from factoring \( S(f) \) as follows:

\[ S(f) = \Psi(j\omega)\Psi(-j\omega) \]  \hspace{1cm} (2.13)

\( S(f) \) is the power spectral density of the input signal phase added to the equivalent noise phase modulation. The equivalent noise phase modulation is the difference of the in phase and quadrature noise components, found from the inner product of the noise with the basis functions as described in the MLE derivation. Since \( S(f) \) depends on a given input signal, the optimization can be done for a given input type (as in a phase step, frequency step, etc.). Furthermore, it is shown in [10] and [14] that the optimization depends on a particular value of the signal to noise ratio. Optimization is usually done by choosing the smallest signal to noise ratio expected to be encountered, since that is the worst case scenario.

To illustrate the above statements, consider the input signal involving a phase step, \( \Phi_i(s) = \frac{\theta}{\delta} \). The optimum filter is found to be:

\[ H_{opt}(s) = \frac{B_0}{j\omega + B_0} \]  \hspace{1cm} (2.14)

which is the transfer function for a first order PLL. The parameter \( B_0 \) is proportional to the signal to noise ratio as well as the square of the magnitude of the phase step. The choice of the loop gain \( K \) (as there is no loop filter) depends solely on the expected signal to noise ratio for a given input, so when the SNR is high, \( K \) is high to reduce
the phase step error. Its optimum value is:

\[ K = \lambda \theta \sqrt{\frac{A^2}{N_0}} \]  \hspace{1cm} (2.15)

where \( \lambda \) is a constant chosen to be a safety factor against the expected phase step.

Consider the case of the angular frequency step (phase ramp), \( \Phi_t(s) = \frac{\Delta \omega}{s^2} \). The optimum filter is then:

\[ H_{opt}(s) = \frac{B_1^2 + j\omega \sqrt{2}B_1}{B_1^2 - \omega^2 + j\omega \sqrt{2}B_1} \]  \hspace{1cm} (2.16)

When compared to the transfer function of the second order loop with integrator and phase correction as the loop filter, we see that they are identical. The optimum loop parameters are:

\[ \zeta = \frac{\sqrt{2}}{2} \]  \hspace{1cm} (2.17)

\[ \omega_n^2 = B_1^2 = \lambda \Delta \omega \sqrt{\frac{A^2}{N_0}} \]  \hspace{1cm} (2.18)

where \( \lambda \) is a constant giving a safety margin against the expected frequency step. Once again, the parameters are directly dependent on both the form of the input and the particular signal to noise ratio.
2.5 Summary

In this chapter the necessary background information to understand PLL's was covered. The PLL components were introduced, analyzed, and their relation to loop performance found. The defining equations for the linear PLL model were introduced. It was found that for a slowly varying phase, the PLL yields the MLE of the signal phase. Optimization with respect to mean squared error is possible by finding the optimum Wiener filter for a particular input signal and SNR.
CHAPTER III

Simulator Description

To aid in and verify the design of a phase locked loop, the cheapest and perhaps easiest method is to perform a computer simulation. This avoids the costs of actual implementation while allowing changes to be made rather quickly and easily. Although a simulation does not guarantee the functionality of a circuit, it gives a good guide as to how it will perform. Therefore, I will simulate the various levels of phase-locked loop design with a computer program that will allow me quick design turnaround and ease in changing parameters.

I have chosen to use Matlab's Simulink Toolbox to perform the desired simulations. Simulink is a versatile and powerful tool for creating and simulating dynamic systems. Since a phase-locked loop is a dynamic system, Simulink should be a good choice for the simulation. However, Simulink was not designed for the communications area, so problems may exist in finding all of the needed blocks. It is integrated with Matlab, so it can be a very powerful tool. It extends Matlab's functionality to dynamic systems while keeping Matlab's general purpose functionality [15]. There are two parts to a Simulink simulation. The first is designing a particular model. The second is the simulation of the model. Various algorithms exist for the simulation, so the
algorithm can be chosen to meet one’s particular needs. The numerical integration methods include Runge-Kutta orders three and five, Euler, and Gear, among others. It is easy to design a model, analyze it, and then change the model based on simulation results. Each model has an associated S-function, which is available in Matlab. The S-function that actually defines the system dynamics through its defining differential equations.

A block diagram (model) can be laid out rather simply. It uses a graphical user interface with drag and drop capability. One opens a file in a window and drags and drops various blocks onto the window and connects them as one sees fit. All commands and functions can be mouse driven. Many common blocks are available in the various block libraries included with Simulink. All of the function blocks are organized accordingly in groups so as to make them easy to find. However, if a desired block is not available, it is possible to redefine blocks to meet current needs. New blocks can also be defined.

One nice feature is the ability to import data or export data either to or from the Matlab workspace or a file for use in Simulink. It saves simulation time as the simulator does not have to generate the needed data each time. Any node of a system can be analyzed by connecting a scope block to it or by saving the results to file or the Matlab workspace. A scope block enables one to view the simulation in progress. One need not perform the entire simulation if it is not working or enhancements need to be made. Once a block diagram is laid out and a simulation is begun, one can change most parameters on the fly to see the effects on system performance.
The block diagram definition is the first level of simulation and is the level best used for design and modification purposes. The S-function is accessed through a block diagram. The block diagram method, however, is generally slower than the other two levels of simulation, especially when graphical output blocks are used. The next level, command line simulation, is more flexible as one can directly manipulate the simulation parameters in Matlab and the results are saved directly to the workspace or to file. At this level, the S-function is accessed through M-files, the Matlab language itself. The last and third level of simulation is by accessing the S-functions through C or Fortran subroutines. The S-functions can be nested inside each other, even at the different levels. For example, a block diagram could include blocks which represent S-functions created at the m-file or C programming level.
CHAPTER IV

Phase-locked Loops: Angle Demodulation
Simulation and Discussion

In this chapter, simulations will be performed to verify Simulink as a valid method of simulation for PLL's. The PLL will be simulated as an angle demodulator, as the investigation of coupled PLLs in later chapters will focus on simulations involving angle demodulation. An introduction to angle demodulation theory is presented to aid in the understanding of the simulations. Simulations in the presence of noise will be performed to determine how the PLL responds to a noisy input.

4.1 Introduction to Angle Demodulation

Since the second order loop is very common, its ability to demodulate phase and frequency modulated signals will be considered. For a sinusoidal phase modulation, we have the following input signal:

\[ y_i(t) = A \sin(\omega t + m_\theta \sin[\Omega t + \theta_\theta]) \]  \hspace{1cm} (4.1)

where \( m_\theta \) is the modulation index. For the linear approximation to hold, it must be small enough so that \( \sin \theta \approx \theta \). The error signal, \( \phi(t) \), is then:

\[ \phi(t) = m \sin(\Omega t + \theta) \]  \hspace{1cm} (4.2)
If \( H(s) \) is the s-domain transfer function for the VCO phase relative to the input phase, we have:

\[
H(s) = \frac{\Phi_0(s)}{\Phi_i(s)} \tag{4.3}
\]

\[
m = m_i |1 - H(s)| \tag{4.4}
\]

\[
\theta = \theta_i + \text{Arg}(1 - H(s)) \tag{4.5}
\]

There are two instances generally useful to demodulate phase or frequency modulated signals. The first is when \( \Omega \ll \omega_n \), the natural frequency. In this case, the input signal modulation is transferred to the VCO with little attenuation or phase shift. If the input signal is frequency modulated, the signal to the VCO in s-domain form is \( V_c(s) = \frac{s\theta_i(s)H(s)}{K_v} \), where \( K_v \) is the VCO sensitivity. Since frequency is the derivative of phase, if \( H(s) \) passes the modulation without distortion, then the signal at the input to the VCO is the desired demodulated FM signal. The second instance is when \( \Omega \gg \omega_n \). Here, the input signal modulation is not transferred to the VCO. If the phase modulation is not passed on to the VCO, and \( H(s) \) is such that it is a high pass function relative to the frequency of the modulating signal, then the error signal \( \Phi(s) \) is \( \approx \Phi_i(s) \), and we have demodulated the phase modulated signal.

4.1.1 Digital Communications Example: FSK Demodulator

Phase locked loops are able to demodulate a wide variety of signals. PLLs are commonly found in radio receivers to demodulate FM signals. Likewise, frequency shift keying (FSK) modulation is easily demodulated using a PLL. FSK is a digital modulation technique which can be viewed as phase modulation involving steps in the phase
of the signal. Phase demodulation will be demonstrated using a Matlab Simulink simulation. The input FSK signal was generated using Matlab code and consisted of a unit amplitude sinusoidal carrier with a carrier frequency of 10kHz. Its frequency is then stepped ± 500Hz about the carrier frequency.

\[ s(t) = \sin(\omega_0 t \pm 2\pi \Delta f t) \]  

where \( \omega_0 \) is \( 2\pi \times 10000 \frac{rad}{s} \) and \( \Delta f \) is 500 Hz. This is the form of a binary frequency shift keyed signal with the bits represented by the frequencies 10500 and 9500 Hz respectively. The signal can also be viewed as a phase modulated signal, with the modulation being a sequence of phase steps. The unity amplitude is representative of an automatic gain control (AGC) in place before the input to the phase locked loop. An AGC is usually necessary as loop performance is dependent on the amplitude of the input signal if an analog multiplier is used. Alternatively, a different type of phase detector may be used which allows the PLL to be function essentially independent of signal level, such as a digital PLL. In any case, the amplitude of the input signal may vary widely, thus necessitating the use of an AGC. The bit interval for the FSK modulated bit stream was chosen to be 10 milliseconds which satisfies the constraint that the bandwidth of the phase modulation \( B_{\theta(t)} \) must be much less than the carrier frequency, \( f_c \), for the phase locked loop to be able to track the desired signal \( (B_{\theta(t)} \ll f_c) \). The input signal was chosen to be an alternating sequence of bits, which is the most difficult sequence for the PLL to track, as a change is occurring at every bit interval. The desired demodulated signal then takes the form of a 50 Hz square wave. The output was taken after the phase detector in this case.
Figure 4: Block diagram of FSK demodulator

The phase locked loop was chosen to be the popular second order loop to allow for independent choice of loop parameters. The phase detector is an analog multiplier with unity gain. The loop filter was chosen to be a first order low pass filter. The voltage controlled oscillator (VCO) is the sinusoidal type. The demodulated output is then filtered with a 1st order low pass filter to remove any high frequency ripple not removed from the loop itself.

As is seen in Figure 5, the FSK signal is demodulated properly and a clean square wave results. Its frequency is 50 Hz, while the amplitude is not important as it can be amplified to achieve the desired signal level. Changing the phase-locked loop block parameters can yield various rise times, overshoots, and settling times as with any control system. Changing these parameters was indeed easily accomplished with Simulink. In designing the PLL, it was easy to modify circuit parameters to see their
effect on the circuit, even while the simulation was running. Cutoff frequencies, gains, and simulation step sizes, etc. could all be modified in the middle of a simulation. The order of the system could not be changed in the middle of the simulation, however. The ability to modify the block diagram in mid-simulation was especially helpful to tweak parameters to achieve better demodulation in the above PLL. This simulation has demonstrated that Simulink is a viable program for PLL simulation. Alternative modulation schemes will be looked at next to demonstrate the utility of the PLL as well as Simulink.

The above simulation results were achieved mainly by adjusting circuit parameters during the simulation until the signal was demodulated properly. The next simulation was explicitly designed for the signals of interest and all parameters chosen accordingly. Adjustment of parameters was done only to see effects on the circuit, not to be a major modification of the original design.
4.1.2 Analog Communications Examples: AM and PM Demodulation

The next signals of interest are phase and frequency modulated signals, as their demodulation has been a major application of the phase-locked loop. We will show that Simulink has no problem simulating the demodulation of these signals, at least in the case of a single PLL. The input signal consisted again of a unity amplitude sinusoidal carrier at 10 kHz, this time frequency modulated by a 50 Hz sinusoidal wave. Hence, the input when FM modulated is:

\[ y_i(t) = \sin \left( \omega_0 t + m_i \int_0^t m(t) dt \right) \]  \hspace{1cm} (4.7)

where

\[ m(t) = \cos(2\pi 50t) \]  \hspace{1cm} (4.8)

Figure 6: Block diagram for demodulating 50Hz sinusoidal FM
See Figure 6 for a block diagram of the FM demodulator. The phase-locked loop was designed to demodulate the 50 Hz sinusoidal wave. Again, a second order loop filter was used. Following the analysis in [10], the loop parameters were chosen to ensure stability and functionality, similar to a prototype second order system. A loop filter consisting of a low pass filter with phase correction was chosen. Its transfer function was chosen to be:

\[ F(s) = \frac{1 + s\tau_2}{1 + s\tau_1} \]  \hspace{1cm} (4.9)

where \( \tau_1 = 1 \) second and \( \tau_2 = 4.44 \times 10^{-4} \) seconds. The parameters were chosen so that \( \omega_n = 500 \times 2\pi \) rad/sec which is much greater than \( \Omega = 50 \times 2\pi \) for the modulation frequency, thus satisfying \( \omega_n \gg \Omega \). For this type of filter, \( \omega_n^2 = K/\tau_1 \) and \( 2\zeta\omega_n = (1 + K\tau_2)/\tau_1 \) [10]. Then \( K=49\text{E}6 \) rad/V for \( \zeta = 0.7 \). The demodulated signal is shown in Figure 7.

\[ \text{Figure 7: Demodulated 50 Hz sinusoidal FM signal} \]
We see that the PLL does indeed track the desired signal in Figure 7. The output is a 50 Hz sinusoidal wave. It takes about 1 cycle for the loop to lock on to the FM signal and track it in a steady state condition. Again, Simulink shows it can simulate a PLL as we would like. The main problems encountered during this simulation were not in the theory of PLL's, but in setting up the loop and the simulation. The step size of the simulation must be chosen small enough to not introduce errors into the simulation, yet large enough to make the simulation speed reasonable (of course this is processor dependent). When the time step of the input data does not correspond with the time step of the numerical integration, the data is interpolated to find the data values corresponding to the integration time steps. The other difficulty was that the units of the VCO block and the filter blocks as defined by Simulink are different (radians vs. Hz). One must be careful to enter the proper units in the proper blocks. Each defined block does not necessarily have the same units for similar components, so it is rather easy to make a mistake here with the simulation model definition.

Next, a phase modulated signal will be demodulated with a PLL similar to the above. The input signal is defined as:

\[ y_i(t) = \sin(\omega_0 t + m_i m(t)dt) \]  \hspace{1cm} (4.10)

where

\[ m(t) = \sin(2\pi 50t) \]  \hspace{1cm} (4.11)

but the output is taken at the output of the phase detector and the parameters are chosen accordingly so that \( \Omega \gg \omega_n \). The phase demodulated signal is shown in Figure 8.
Again, we see the expected results. After the initial transients, the PLL is able to properly track and demodulate the desired signal. The demodulated output is the 50Hz sinusoidal wave.

To demonstrate a phased-lock loop’s effectiveness on non-sinusoidal modulation, more simulations were performed. The input signal was taken to be a triangular phase modulated signal, which also could be interpreted as a square frequency modulated signal. The loop will be designed and outputs taken accordingly for each interpretation. The modulated input signal is defined as:

$$y_i(t) = \sin(\omega_0 t + m_i m(t) dt)$$  \hspace{1cm} (4.12)$$

where

$$m(t) = \text{triangle}(t)$$  \hspace{1cm} (4.13)$$
which is a 50Hz triangular wave with amplitude one. The modulation index is chosen to be $\pi/6$ so that the linear approximation for the phase-locked loop holds. Two loops will be designed with the same inputs and different parameters to show how different outputs can be had from the same input. The phase demodulated signal is shown in Figure 9 and the frequency demodulated signal in Figure 10.

![Graph of the demodulated 50 Hz triangular PM signal](image)

**Figure 9: Demodulated 50 Hz triangular PM signal**

This simulation demonstrates the interesting fact that for the same input signal, two different message outputs can be obtained simply by changing the phase-locked loop parameters.
Figure 10: Demodulated 50 Hz triangular PM signal, demodulated as an FM square wave

4.2 Simulations in the Presence of Noise

One advantage of the phase-locked loop over other sinusoidal tracking methods is the ability to accurately track the desired signal in the presence of noise. This is especially true when the phase detector is of the multiplier type (i.e. the analog multiplier). The ability to angle demodulate a signal in noise will now be demonstrated using the phase-locked loop used previously in demodulating a sinusoidally phase modulated wave. This time, various levels of additive white Gaussian noise were added to the input signal. The same loop that was previously simulated will be used.

As one can see in Figure 11, the PLL is able to demodulate the signal properly in all cases. Most notable is its ability to demodulate the signal at a -43dB signal to noise ratio (signal power=0.5, noise power (variance)=1000). It is noticed that there
appears to be a small amplitude modulation in the lower SNR cases. This seems to be a simulation artifact, as lowering the order of the output band pass filter decreases the ripple and vice-versa, which is opposite what one would expect if the ripple were a result of the circuit itself. Still, these are difficult operating circumstances, so some degradation in performance should be expected anyway. The band pass filter serves to filter noise from the PLL output. Although the above results do not guarantee that the same resilience to noise will be observed in a coupled PLL system, it demonstrates the ability of the PLL to operate in very difficult situations.

4.3 Simulation Conclusions

In this chapter, it has been demonstrated that Simulink is a useful tool to simulate PLLs. The simulation of PLLs was easy with block diagrams created in Simulink. The anticipated results were obtained in each of the simulations, thus supporting the claim that Simulink can properly simulate a PLL. After it was established that Simulink was able to simulate PLLs, noise was added to a simulation at various levels. It was found that using an appropriate step size, Simulink was able to simulate a noisy PM input. The noise level was varied, and proper demodulation was achieved over a wide range of SNRs.
Figure 11: Demodulated waveforms for a noisy input signal with no pre-filtering with SNR’s -13, -23, -33, and -43 dB respectively, moving CW from upper left.
CHAPTER V

Simulation and Discussion for Coupled Phase-locked Loops

This chapter will investigate the principles behind the currently proposed cross coupled PLLs (CCPLLs) as found in [1, 2, 11] by way of simulation. It is demonstrated that a CCPLL, for the case of slowly varying input signal phases, realizes the approximate maximum likelihood solution for the estimation of the input signal phases. A method of measuring the effectiveness of CCPLLs is proposed. For the application of using a CCPLL as a phase demodulator, the criteria for proper operation are derived when the demodulated output is taken at the output of the phase detector. Phase demodulation in this manner does not require the use of an integrator as it would if the demodulated output were taken at the output of the loop filter.

5.1 Optimality of CCPLL

As in the case of a single phase-locked loop, it would be beneficial for a coupled PLL system to track the desired phases in some optimum manner. It has been shown that a single PLL with an input signal consisting of a sinusoidal signal in the presence of AWGN forces a solution to the likelihood equation and provides the maximum
likelihood estimate (MLE) of the desired input signal phase for a slowly varying phase [3]. Consider the case of the coupled pair of PLLs with known input amplitude described in [1] and [9], or the CCPLL with amplitude control [2]. The input signal is now the superposition of two sinusoids in the presence of AWGN:

\[ y(t) = \cos[\omega_1 t + \Phi_1(t)] + \cos[\omega_2 t + \Phi_2(t)] + n(t) \quad 0 \leq t \leq T \]  

(5.1)

where \( \Phi_i(t) \ i = 1, 2 \) is a slowly varying function over the above interval or a constant and \( n(t) \) is AWGN. The coupled PLLs operate as described in section 1.5. The CCPLL system tries to remove all interfering sinusoids from the input to each PLL. After one PLL has begun tracking the frequency and phase of its particular signal and cancelling it at the input of the other PLL, the second PLL is able to track the remaining signal and reinforces the tracking of the first PLL by cancelling out the second signal at the input to the first PLL. This sequence of locking occurs in the initial transient before steady state operation is reached. The ability of one PLL to lock onto one of the signals in a mixture is through the capture effect, in which the PLL will lock on to the strongest signal at its input.

As in the previous optimality discussion for a single PLL, the phase of each signal is assumed constant or slowly varying over the observation interval. If the unwanted signal could be exactly cancelled at the input to each PLL, then each PLL would be able to provide the maximum likelihood estimate of its respective signal phase, as each input would be a sinusoid in the presence of AWGN. This is, however, an ideal case. When the opposite signal is subtracted from a PLL’s input, the opposite signal phase is at best the maximum likelihood estimate for the signal phase, so it is
an approximate cancellation. Only when the estimate of the opposite signal’s phase is equal to the opposite signal’s actual phase will the coupled PLL system be truly optimum in the maximum likelihood sense. An exact cancellation could ideally be made if noise was not present. The input to PLL 1 in the pair is:

$$\overline{y_1(t)} = y_1(t) + ay_2(t) + n(t) \quad 0 < t < T$$ (5.2)

where $a$ is a small, non-zero constant which ideally is 0. In essence, a single PLL in the coupled PLL pair is forcing a solution to the following equation when the loop filter in each case is an integrator:

$$\int_0^T y_1(t) \sin(\omega_0 t + \hat{\theta}_1ML) dt + a \int_0^T y_2(t) \sin(\omega_0 t + \hat{\theta}_2) dt = 0$$ (5.3)

If $a$ were zero, we have the same equation as for a single PLL estimating a single phase. If the loops have achieved lock and are operating in steady state, then $a$ will be small. Hence we have an approximate maximum likelihood solution.

Consider the coupled PLL pair described in [2], where manual adjustment of the summer inputs is not used or knowledge of the input signal amplitudes is lacking, as is usually the case. This PLL system is again trying to cancel the unwanted signal at the input to each respective PLL. It does this by generating the optimum amplitude estimates for both of the signals. We now have an additional source of possible error when the loops are operating in lock as intended. In addition to a possible difference in signal phase, there is now a possible difference in amplitude as well. Again, if the loop operated ideally, we would be forcing the maximum likelihood solution for the signal phases. The coupled PLL pair would again be forcing the solution to equation 5.3.
5.2 CCPLL Implementations

This section will describe the two implementations of a CCPLL that have previously been proposed in [1, 2]. A description of operation and block diagram are given in each case.

5.2.1 CCPLL Without Amplitude Control

This section describes the operation of the CCPLL without the use of an amplitude estimation circuit. It is assumed that the amplitudes of the input carrier signals are known. Knowledge of the input signal amplitudes is necessary for cancellation purposes in the CCPLL. Although this is not the most practical implementation of the CCPLL, it is instructive to analyze and simulate this implementation as it is easy to understand and the simulations will provide a comparison for the simulations of the CCPLL with amplitude estimation.

Consider the case of the coupled PLL where the amplitudes of the input signals are considered known. This system has been described as a maximum a posteriori (MAP) estimator for suppression of interchannel interference in FM receivers [1]. A block diagram for such a system was shown in Figure 2 of Chapter I.

Two individual PLLs can be cross-coupled, with the VCO's and 90 degree phase shifts providing the respective signal estimates needed for the cancellation of the interfering signal at the opposite PLL. The output of the VCO of a PLL is ideally 90 degrees out of phase with the input signal when it is tracking properly. If the VCO signal is further phase shifted another 90 degrees, it is the opposite of the input signal
to the PLL. This is the signal estimate that is needed at the input to the opposite
PLL in each case to cancel the signal that is interference to that particular PLL .
Hence, a cross-coupled PLL system can be used to jointly estimate the interfering
signal so that they may be cancelled at the appropriate inputs.

5.2.2 CCPLL With Amplitude Control

The previous section described a CCPLL that is able to track multiple sinusoids.
Unfortunately, it is not very practical because of the assumption that the input signal
amplitudes are known. This section describes a CCPLL that removes the restriction
of known input amplitudes.

In [2] a CCPLL structure has been proposed in which estimates are generated
for the amplitudes of the signals in the input mixture. Rather than using assumed
knowledge of the input signal amplitudes to cancel interfering signals in a CCPLL,
the amplitude estimates are used for this cancellation. A block diagram of such a
system is shown in Figures 12 and 13. The entire block diagram is rather confusing
so it is broken up into two figures.

The derivation of this particular CCPLL system follows. The input mixture signal
is described as:

\[ v_1(t) = x_1(t) \sin[\omega_0 t + \psi_1(t)] + x_2(t) \sin[\omega_0 t + \psi_2(t)] \]  \hspace{1cm} (5.4)

where \( x_1(t) \) and \( x_2(t) \) are the (possibly) time varying amplitudes to be estimated. The
signals are written with a common carrier frequency, which is done by incorporating
a difference between the noted carrier frequency and the original into the phase term.
Figure 12: Block diagram of CCPLL system with amplitude control

Figure 13: Amplitude control portion of circuit
It is desired to form an estimate \( \hat{s}_2 \) of the amplitude of the interfering signal with amplitude \( x_2 \). This will be subtracted from the mixture at the input of the first PLL. In [2], the optimum weights are derived by minimizing \( e^2(t) \), the squared error (the error resulting from subtracting the estimated amplitude from the actual amplitude), using the steepest descent algorithm. This is for the noiseless case. The resulting equations are:

\[
\frac{de^2}{dY_{2R}} = \frac{-dY_{2R}}{\beta dt} = -[x_2 \cos(\psi_2 - \phi_2) - Y_{2R}]
\]

\[
\frac{de^2}{dY_{2I}} = \frac{-dY_{2I}}{\beta dt} = -[x_2 \sin(\psi_2 - \phi_2) - Y_{2I}]
\]

(5.5)

(5.6)

where the \( (Y_{ij}'s) \) are the amplitude component estimates (weights) as shown in Figure 13. This analysis is for one half of the amplitude estimation circuitry. The other half is identical to the first with the subscripts changed to reflect the appropriate inputs and estimates. Equations 5.5 and 5.6 set the gradient of the squared error with respect to the weights proportional to the negative time derivative of the the weights. \( \beta \), the proportionality constant, determines how fast the weights converge to their steady state values. It is related to the bandwidth of the low pass filters used in the amplitude control loop. Double frequency terms are neglected in the above equations and are assumed filtered out, a realistic assumption as the integrator and low pass filter in the loop serve to remove high frequency terms. If \( \beta \) is sufficiently large, the weight estimates will achieve a steady state values given by:

\[
Y_{2R}(t) = x_2(t) \cos(\psi_2 - \phi_2)
\]

(5.7)

\[
Y_{2I}(t) = x_2(t) \sin(\psi_2 - \phi_2)
\]

(5.8)
Therefore, we can find the amplitude estimate by:

$$x_2(t) = \sqrt{[Y_{2R}(t)]^2 + [Y_{2I}(t)]^2}$$  \hspace{1cm} (5.9)

The amplitude component estimates, the \((Y_{ij}')s\), multiplied by the in phase and quadrature VCO signals give the components of the amplitude estimate in the I and Q channels, and are then added together and subtracted from the input. The in phase and quadrature VCO signals are generated by phase shifting the VCO output by 90 degrees, as was done in the case with no amplitude control. We have:

$$v_2(t) = x_1(t) \sin(\omega_0 t + \psi_1) + x_2(t) \sin(\omega_0 t + \psi_2)$$  

$$-B_2 Y_{2R}(t) \sin(\omega_0 t + \phi_2) - B_2 Y_{2I}(t) \cos(\omega_0 t + \phi_2)$$

$$\approx x_1(t) \sin(\omega_0 t + \psi_1)$$  \hspace{1cm} (5.10)

So, the interferer is approximately cancelled. \(B_2\) is the amplitude of the VCO signal, which is taken to be one in future simulations. \(\phi_1\) and \(\phi_2\) are the estimates of the input phases (the phases of the VCO's). To see how all this is realized, we investigate further. We want to verify that equation 5.5 is solved by this loop. From the block diagram and ignoring the integration for the time being (as it is a linear operation and can switch the order of operations), we have that the time derivative of the weight \(Y_{2R}\) is equal to \(v_2\) as given in the equality of equation 5.11, and multiplied by the VCO signal, \(B_2 \sin(\omega_0 t + \phi_2)\). This is then filtered by the low pass filter response, \(g_{2R}(t)\). Ignoring the double frequency terms which are generated by this multiplication and using trigonometric identities, we can write:

$$\frac{dY_{2R}}{dt} = \frac{B_2}{2} \left[ x_1 \cos(\psi_1 - \phi_2) + x_2 \cos(\psi_2 - \phi_2) - B_2 Y_{2R} \right] \ast g_{2R}(t)$$  \hspace{1cm} (5.11)
If the low pass filter response is chosen so that the bandwidth is narrow compared to the term containing $x_1$ and wide compared to the term containing $x_2$, the result realized by the CCPLL, after normalization, is the following equation:

$$\frac{dY_{2R}}{d\tau} = \beta_2 [x_2 \cos(\psi_2 - \phi_2) - Y_{2R}]$$  \hspace{1cm} (5.12)$$

which is identical to the desired equation 5.5. $\beta_2$ is given by $B_2^2/2\omega_n$ and $\tau$ by $\omega_n t$. We see that this system yields the appropriate amplitudes to satisfy the criteria set forth by the steepest descent algorithm. The above term is then integrated in the amplitude control loop to yield the desired estimate for $Y_{2R}$. All of the previous analysis is paralleled for the opposite PLL and amplitude generation circuit, replacing the 2’s by 1’s and taking the appropriate VCO outputs. As a result, the amplitude of the remaining signal is estimated just as in the previous case. The transient operation of this system is such that one PLL begins to track one of the inputs (by the capture effect or other means), the appropriate amplitude is estimated, and a cancellation takes place at the opposite PLL. The opposite PLL begins tracking its signal and then cancels it at the opposite PLL to reinforce the tracking that has already begun there. Then, the steady state operation has both PLLs tracking their respective inputs and cancelling them at the opposite PLL.

However, it is not realistic to assume there are always just two signals of interest that we need to track. We need to consider the case of M input signals and M PLLs in a CCPLL structure. Having M PLLs is a more useful implementation of the CCPLL as it is able to track more signals or cancel more interference than than the case of two PLLs. For the CCPLL with no amplitude control, the cancellation action is achieved
simply by adding more PLLs and adjusting the summer gains appropriate to the
input signal levels in the mixture and subtracting a given PLL’s phase shifted VCO
output from each of the other PLLs. For the CCPLL with amplitude estimation,
the same amplitude control structure is now repeated M times instead of twice. The
same amplitude estimation circuit is used for each PLL, but the amplitude estimate is
subtracted from the remaining M-1 PLLs in the M CCPLL structure [11]. When the
PLLs are in tracking mode, the M-1 interfering signals are approximately cancelled
at the input to each PLL.

For M input signals, the input to each PLL is the sum of input signals minus
the estimates of the unwanted signals. The complexity also increases in this case.
For M PLLs in the CCPLL system, it requires 2M(M-1) integrators (and filters and
multipliers) as implemented in [11]. For M=2, this required 4 integrators, for M=3,
12 integrators, and for M=4, 24 integrators. Complexity of the system will definitely
be a consideration for \( M > 2 \).

As in the M=2 case, the filters in the amplitude control loops need to be low
pass and pass the \( \cos(\psi_i - \phi_i) \) terms while blocking the \( \cos(\psi_i - \phi_j) \) terms, \( i \neq j \). This will be the case if the PLLs can accurately track their respective phases. The
bandwidth of the amplitude control filters affect the settling times of the circuit.
A wider bandwidth allows for faster settling times, but could allow more error in
the estimate. In [11], simulations are performed to determine how the change in
amplitude control filter bandwidth affects the acquisition time of the CCPLL system.
It is found that wider bandwidths correspond to shorter acquisition times, a result
which seems intuitive.
5.3 CCPLL Simulations: FM signals

This section demonstrates Simulink's ability to simulate CCPLL's. Simulations are performed for cases where the CCPLL is already known to work, and the simulation results shown to agree with the expected output.

5.3.1 Noiseless Simulation

Consider first the CCPLL without amplitude control. The input signal is of the form:

\[ y(t) = s_1(t) + s_2(t) \]  \hspace{1cm} (5.13)

where \( s_1(t) \) and \( s_2(t) \) are frequency modulated signals of differing frequency, amplitude, and carrier frequency. In the following simulations, the modulating signals will be a 50Hz sinusoidal wave and a 25Hz sinusoidal wave. The modulation index is \( \pi/6 \) so the linear model of the PLL is valid. The carrier frequencies as well as the amplitudes will be varied. In the first simulation, the 25Hz sinusoid is modulating an 8kHz sinusoidal carrier and the 50Hz sinusoid a 10kHz carrier. The amplitudes are both unity. The block diagram for this system is shown in Figure 14.

Note that the PLLs each have an extra block not previously described. The memory block is a simulation trick used to break algebraic loops in the diagram by Simulink. An algebraic loop is a loop containing more than two of a certain type of block, including products, VCO's, and zero-pole blocks. When an algebraic loop is present, the simulator will have to stop at each step to solve the loop equation and greatly slows down the simulation. A memory block causes a one integration step delay at the output of the block. The integration step is very small for accurate
Figure 14: Block diagram of a coupled PLL system

simulations so this has little, if any, effect on the simulations. This has been verified for the case of a single PLL. For a single PLL, the simulation could be performed in a reasonable time without a memory block, but problems arose when simulating a CCPLL. A coupled PLL simulation that took 45 minutes with no memory blocks took less than a minute with the addition of memory blocks in the feedback path. A clock block has also been added. This is to output the simulation time steps to
the Matlab workspace so the outputs can be plotted correctly, as the integration time steps are not guaranteed to be uniform in a given simulation. There is also a second VCO block for each PLL. This is for simulation purposes only. Simulink, a control systems simulator, does not contain some of the blocks applicable to the communications area. This includes the 90 degree phase shift needed for the CCPLL system. I have introduced a 90 degree phase shift by unmasking the Simulink VCO block and changing an internal sine block to a cosine block to achieve a 90 degree shift at the output relative to the regular VCO when their input signals are identical.

The individual PLLs are identical in this block diagram. For comparison’s sake later in the chapter, the PLLs are identical so none of the PLLs has an advantage in tracking any of the signals. Their bandwidth is such that the both the 25 and 50Hz modulation is passed to the VCO undistorted. This allows for an accurate comparison of how well each PLL is cancelling at the opposite input. In practice, the PLLs could be designed so one is better able to track one of the signals. In this way, it may be possible to tell which PLL will output which demodulated signal.

Both PLLs in Figure 14 use a low pass filter with phase correction as the loop filter. The output filters are used to clean (post-filter) the output signals. There are two filters at the output of each PLL. One is the filter used to clean up the desired output and the other is used to check for a component of the opposite signal at the output of each PLL. Using both output filters, it can be verified that the desired signal is being correctly demodulated while at the same time cancelling the unwanted signals. The VCO’s each are preset to the respective carrier frequency to avoid the
acquisition phase of tracking the signals. This thesis focuses on cancellation aspects of the CCPLL, rather than acquisition. Acquisition will be considered in future research. In practice, acquisition devices may have to be used to achieve reasonable times to lock. The gains of the summer devices are set to unity as the signals are both unity amplitude. The four outputs of the CCPLL system are shown in Figure 15. The output of each PLL has two filters, one to function as the normal post-filtering operation and one to determine the signal level of the interfering signal.

The desired outputs are a 25 and a 50Hz sinusoid, as well as the interfering sinusoids to each of these to die out as the cancellation takes effect. This is indeed what happens in Figure 15. The interfering signals for each PLL do die out as expected. Hence, we see that Simulink is able to simulate a CCPLL accurately.

Simulations were also performed for the same input signals as in the above simulation, this time using a CCPLL with amplitude control. No noticeable difference was found from this simulation compared to the results found in Figure 15. So Simulink also simulates a CCPLL with amplitude control properly. A description of the amplitude control block diagram follows.

A block diagram of the amplitude control circuit is shown in Figure 16. Both loop filters were again chosen to be identical and so that they passed the modulation to the VCO undistorted. The bandwidth of the loop filters is ten times the highest modulation frequency. Another important bandwidth to choose now that an amplitude control circuit is used is that of the filters in the amplitude estimation loop. Recall that this bandwidth must filter out the terms at a frequency of the difference
between the modulating phases, in this case, 50-25=25Hz. A larger bandwidth also allows the amplitude estimates to achieve their steady states faster. In this circuit, the bandwidth of the low pass filters was chosen to meet both of the above criteria.

Note that each of the PLLs have the memory blocks and the modified VCO as in the Simulink diagram for the CCPLL without amplitude control. The four outputs are used again to determine the desired outputs and the interference. The amplitude estimation portion of the circuit was implemented without problems in Simulink.
Figure 16: Block diagram of CCPLL with amplitude control, FM demodulator

5.3.2 Simulation in Presence of AWGN

Simulations were also performed for the case when the input mixture was in the presence of AWGN. A number of signal to noise ratios were simulated. It was found in both types of CCPLLs that the performance of the simulations degraded sharply as the signal to noise ratio (SNR) was decreased much lower than -10dB. Figure 17
shows the results for the SNR=-10dB for the CCPLL with amplitude control. The results for the no amplitude control case are very similar.

Figure 17: Demodulated FM outputs, amplitude control, SNR=-10dB

It should be mentioned that the performance of the CCPLL was poor when the integration step size of the simulation was too large. This performance decrease is an effect of the simulation and not of the CCPLL. Simulations with noise required a
smaller step size with the Simulink simulator than for the identical circuit without noise. For example, the above results were achieved by halving the step size of the same simulation without noise. Without the reduction of stepsize, the outputs were indecipherable. This leaves the possibility that the performance of the CCPLL could be acceptable (desired demodulated outputs with interfering signals cancelled) at a lower SNR. I was unable to investigate this possibility as computer memory limitations made the further decrease of integration step size impractical. In [8], it was demonstrated that a CCPLL had a threshold effect at about -5dB SNR, so the above SNR performance threshold at least seems reasonable.

5.4 Proposed CCPLL Performance Measure

In this section a performance measure is proposed that gives a measure of how well a CCPLL is cancelling the interfering signals at each respective PLL.

Consider the block diagrams of Figures 14 and 16. Four outputs are measured: the desired signal at the output of $PLL_1$ ($S_{1,D}$), the desired signal at the output of $PLL_2$ ($S_{2,D}$), the interfering signal to $PLL_1$ which is of the form of the desired output of $PLL_2$ ($S_{2,I}$), and the interfering signal to $PLL_2$ which is of the form of the desired signal output of $PLL_1$ ($S_{1,I}$). All of the above signal levels are measured after the circuit has achieved steady state. The cross channel interference ratio (CCIR) is defined as $20 \log_{10} \frac{S_{1,I}}{S_{2,D}}$. The CCIR is a ratio of the power of the unwanted signal to the power of the desired signal for a given signal type (e.g. 25Hz sinusoidal modulation). The was useful for the simulations performed in this thesis, as the outputs that were
useful in determining whether a CCPLL was functioning correctly could also be used to give a quantitative measure of the interference cancellation. Another performance measure that could be used to quantify cancellation performance but was not used in this thesis is the squared error in the cancellation. This error could be found by subtracting the 90 degree phase shifted VCO output estimate from the actual input signal and squaring the difference. The error would be squared as both positive and negative errors should be weighted equally and the measure is better defined as a positive quantity.

The CCIR was calculated for a number of simulations and plotted against the difference in carrier frequency of the input signals to the CCPLL. Plotting against the difference in carrier frequencies is a logical choice as one would expect the performance of the CCPLL to become poorer in cancellation ability as the two input signals became more and more alike. CCIRs were calculated for CCPLLs with and without amplitude control for the circuits used in the above simulations (FM signals). The CCIR vs. carrier difference frequency plot is shown in Figure 18.

We see that the performance for each signal type and CCPLL type is nearly identical. We would ideally want the CCIR constant over the entire range of frequencies of interest, but that is not the case here. Note that the amplitude control circuits have a better CCIR for the range in which the carrier difference is small (100-200Hz), and that the CCIR ratio increases in general as the carrier difference decreases. It is concluded that the CCPLL has excellent cancellation performance over a wide range of carrier difference frequencies. The increase in CCIR as the carrier difference decreases
is reasonable as the operating conditions of the CCPLL are becoming more and more difficult. The improvement in performance at small carrier difference realized by the amplitude estimating CCPLLs can be explained by the fact that this type of CCPLL uses feedback in two parts of the circuit. When the PLL feedback is giving a poor estimate of the phase for whatever reason, we could expect the amplitude estimation circuit to compensate for the error in the PLL estimate. The amplitude estimation circuitry is trying to minimize the squared error in the cancellation and will do this
regardless of how the PLLs are performing. The main reason for the poor performance at small carrier difference frequencies is that the capture effect is not taking place. The capture effect is a well known effect in which a PLL, presented with more than one input, will lock onto and track the stronger signal. Both of the input signals in this case are of unity amplitude, so the loop is not able to lock onto to only one of the signal as desired. Individual simulations have been performed which demonstrate that for an carrier input amplitude ratio of 10:1, the CCPLL is able to use the capture effect to allow for some cancellation at even identical carrier frequencies.

It should also be noted that in [1], CCPLL's without amplitude control have actually been built and were able to demodulate and cancel two FM signals of identical carrier frequency, at differing amplitudes. This fact supports the above claim regarding the capture effect. It would also be beneficial to build the CCPLL simulated to get the CCIR curves to verify the simulator. The performance at small carrier difference frequency observed here is possibly a partial result of the simulation itself. As in the case of a simulation in the presence of noise, the stepsize may need to be decreased further to get a more accurate simulation, an avenue that may be pursued in the future.

5.5 CCPLL and Phase Demodulation

This section pertains to the use of the CCPLL as a phase demodulator for the case of a small modulation index. In certain instances, phase demodulation taken at the output of the phase detector can be useful. A CCPLL can track and demodulate signals that
can be modeled as a small modulation index PM wave. CCPLL performance identical to FM demodulation can be achieved under certain circumstances which are derived here.

The literature has addressed the CCPLL for use as an FM demodulator. When a PLL is designed to demodulate FM, the loop is generally set up to pass the modulation to the VCO (modulation tracking loop). The VCO then tracks both the frequency and phase of the input signal, so its output is a model of the input, and 90 degrees out of phase with the input. However, when using the PLL as a demodulator for a small modulation index PM signal, the modulation is not passed to the VCO (carrier tracking loop) as the output is taken at the output of the phase detector. The VCO then just tracks the frequency of the input signal, not its phase. The output could also be taken at the input to the VCO if the modulation was allowed to pass, but that signal would have to be integrated for to arrive at the desired PM signal. Taking the demodulated output at the output of the phase detector avoids the use of the integrator. This is beneficial in instances where including an extra integrator would be detrimental. One such case is a VLSI implementation of a CCPLL in which an integrator requires a number of transistors and an area consuming on chip capacitor. If small chip area was an important issue, the use of an integrator may be avoided in certain PM applications. Note that a small modulation index is assumed for the modulation so that the linear approximation for an analog multiplier phase detector is valid.
The conditions for proper operation as a phase demodulator when the demodulated output is taken at the output of the phase detector are now derived. Consider Figure 19, which is a block diagram of the cancellation and phase detector portions for one of the PLLs in the CCPLL.

![Block Diagram](image)

Figure 19: Partial diagram of CCPLL for PM derivation

The input mixture composed of the superposition of two sinusoids is $y_{\text{sig}}$. The estimate of the interfering signal at the input to this PLL is $\hat{y}_2$ (the phase shifted output from the opposite VCO) while $y_{\text{VCO1}}$ is the VCO output for this PLL. The output of the phase detector and also the demodulated output is $y_{\text{in}}$. The relevant equations follow.

\[ y_{\text{sig}} = \sin(\omega_1 t + \phi_1) + \sin(\omega_2 t + \phi_2) \]  \hspace{1cm} (5.14)

\[ \hat{y}_2 = \sin(\omega_2 t) \]  \hspace{1cm} (5.15)

\[ y_{\text{VCO1}} = \cos(\omega_1 t) \]  \hspace{1cm} (5.16)
\[ y_{in} = (y_{sig} - \hat{y}_2) y_{vco1} \]
\[ = 0.5(\sin \phi_1 + \sin([\omega_1 - \omega_2]t - \phi_2) - \sin([\omega_1 - \omega_2]t)) + \text{double freq. terms} \quad (5.18) \]

Then expanding using trigonometric identities and collection of like terms we have:

\[ y_{sig} = 0.5 (\sin \phi_1 + \sin([\omega_1 - \omega_2]t\{\cos \phi_2 - 1\} - \cos[\omega_1 - \omega_2]t \sin \phi_2) \quad (5.19) \]

Since the modulation index is considered small, the amplitude of the phases is small as well, and we can approximate \(\sin(\theta)\) by \(\theta\) and \(\cos(\theta)\) by 1. This lead to the following signal at the input to the first PLL after an attempt was made to cancel the second signal:

\[ y_{in} = 0.5\phi_1 - 0.5\phi_2 \cos(\omega_1 - \omega_2)t \quad (5.20) \]

We see we have the desired term, \(\phi_1\), as well as an unwanted term, \(\phi_2 \cos(\omega_1 - \omega_2)t\).

Since the transfer function relating the output of the phase detector to the input phase is a high pass function, we must remove the unwanted term by post-filtering the output. The bandwidth of the low pass filter must be such that it passes the desired term and removes the undesired term. How the filter bandwidths are chosen will depend on the application since knowledge of \(\omega_1\) and \(\omega_2\) is necessary to determine the appropriate cutoff frequency. If \(|\omega_1 - \omega_2|\) is greater than the sum of the bandwidths of \(\phi_1\) and \(\phi_2\), a low pass filter exists that can remove the undesired term while passing the desired term.

Simulations were performed to verify that the CCPLL is able to function as a phase demodulator in the above context. In the first example, we have two inputs to
a CCPLL with amplitude control. One of the modulating signals is a 50Hz triangular wave while the other is a 25Hz sine wave. The phase demodulated outputs are shown in Figure 20.

![Graphs showing triangular PM output, 50Hz, sinusoidal interference to triangle wave, sinusoidal PM output, 25Hz, and triangular interference to sine wave.](image)

Figure 20: Simulation results of CCPLL, equal amplitude, 8 and 10kHz carriers

We see that the CCPLL is able to demodulate the desired signals properly and cancel the interferer to each PLL in the CCPLL, just as in the case of FM signals. The outputs are taken with low pass filters which serve to remove the unwanted term.
It is also possible to find the cross channel interference ratio for PM signals. The CCIR is defined as before. Simulations were performed across the same range of carrier difference frequencies as in the FM case for the same two sinusoidal input signals. The main difference in the simulations is in the point where the output is taken. For the PM CCPLL, the output is taken after the phase detector and for the FM CCPLL the output is taken after the loop filter. The loop filter in the PM CCPLL case is designed so the modulation is not passed to the VCO. The CCIR is shown in Figure 21.

The CCIR is similar to that found for the FM case. The trend of increasing CCIR as the carrier difference is decreased is again observed. We do see a difference in the level of cancellation between the two modulating signal types. This is likely the result of the choice of loop filter bandwidth. The PLL that was tracking the 25Hz sinusoid was most likely passing some of the modulation to the VCO due to the choice of bandwidth of the loop filter and allowing a better cancellation of its signal at the opposite input. This effect was not evident until the simulations were completed.

Note that as the carrier difference goes to zero in this case, we expect to see both output signals at each PLL output. The CCPLL cannot cancel the interferer when the carrier difference drops below the threshold given above. When the carrier difference becomes smaller than the sum of the individual phase bandwidths, we can no longer filter out the unwanted term. As the carrier difference is decreased further, it is possible to alias the unwanted phase into the desired phase, even though the phases may be different, and even cancel the desired phase. When the carrier difference
Figure 21: Cross-channel interference ration for PM CCPLL

is zero, the output becomes the superposition of the the individual phases, which explains why there is no cancellation in this case. Still, where applications exist that allow the carrier differences to remain large enough, excellent cancellation of the PM interferers is possible. Another possibility is that one of the PLLs in the CCPLL can be biased towards the desired signal which is phase modulated. It would also be possible that one PLL could be designed as a PM demodulator, and the rest of the PLLs in the CCPLL system could then be modulation tracking as opposed to carrier
tracking. This would allow cancellation of the interfering signals all the way down to zero carrier difference frequency.

5.6 Summary

This chapter has demonstrated that Simulink is a viable method for simulating CCPLLs. Results were obtained which agreed with predicted results and those obtained in the literature for the case of FM signals. A novel performance measure was introduced, the cross channel interference ratio, that allows one to evaluate CCPLL performance vs. the difference in carrier frequency of the input signals. The CCPLL was also demonstrated to function as a phase demodulator for the case of a small modulation indexed phase modulated inputs. The conditions for proper operation were derived and the CCIR found for PM demodulation.
CHAPTER VI

Steps Toward VLSI Implementation

One of the fastest areas of growth today in electrical engineering is in the VLSI field. Advances in CMOS technology have enabled devices to be built at a size and cost unheard of just a few years ago. Fortunately, CMOS technology can also be applied to phase-locked loops, and hence coupled phase-locked loops. This gives the coupled PLL another benefit in addition to those already stated. It can be implemented cheaply and compactly on a VLSI chip.

Investigation has already been undertaken in this area for implementation of a single PLL which may be applied at a later time to a CCPLL system. Work done for master’s thesis by Yuwen Zou, as well as that undertaken myself, attempt to design and simulate PLL components in VLSI technology. Design of components was accomplished through the use of the MAGIC layout editor and extractions performed to SPICE for simulation. Design was undertaken for a $2\mu m$ process.

My contribution to this project was the design and simulation of the VCO, which was of the square wave type. The idea behind the implementation of such a VCO in CMOS is to cascade an odd number of inverters together, and connect the output of the last inverter to the input of the first. A feedback loop is formed and an oscillation
takes place. The number and size of inverters determines the frequency of oscillation. The VCO designed consisted of seven identical stages, cascaded together, and was adapted from a design found in [16]. It is considered a current starved oscillator. The origin of its name will become obvious shortly. Each stage consists of a pull down section of 2 NMOS transistors in series and a pull up section consisting of 2 PMOS transistors in series. One such stage is shown in Figure 22.

![Diagram of CMOS current starved inverter](image)

**Figure 22: CMOS current starved inverter**

Each stage has the functionality of an inverter, with the additional transistor in each half added in series to control the current flow, hence the name current starved. When one of the “regular” transistors is on, the voltage applied at the gate of the
controlling transistor determines how much current will be allowed to flow through the series transistors. Each of the PMOS transistors have a width to length ratio of 20:1, with the gate width 40 $\mu m$ and the gate length 2 $\mu m$. The NMOS transistors are half of this width, to maintain equal rise and fall times in the inverter. Larger than minimum sizes were chosen to reduce the effects of geometry biases and reduce the effect of routing capacitance in an actual implementation, both of which reduce the speed of operation, as described in [16]. Seven of these transistors were cascaded together, with the output of the inverter section connected to the input of the next stage. Each of the PMOS control transistors have the same voltage applied to the gate. Similarly, the NMOS control transistors all have the same control voltage.

The control voltage was set up by an input stage. This consisted of a PMOS and an NMOS transistor in series. The NMOS transistor has the input control voltage applied to its gate, which was then connected to each of the gates of the NMOS transistors in the pull down sections of the repeated inverter structure. The PMOS transistor has its drain and gate tied together which and connected to the drain of the NMOS transistor. Its source was tied to VDD, the power source. The gate was then connected to each of the PMOS transistors in the repeated inverter structure. In this way, a decrease in the NMOS control voltage will cause an increase in the PMOS control voltage and vice-versa. The transistors in this input stage can be scaled to vary the sensitivity of the control voltage.

After the oscillator exists a final output stage. It has the function of being a buffer and ensuring a clean signal is present for connection to other inputs. It is simply an inverter whose output is the output of the VCO.
This VCO was designed to be part of a PLL which is implemented entirely in CMOS. The remaining components will now be described. The phase detector is designed to detect the difference between the reference clock (usually found by limiting the input signal) and the VCO signal. It can be implemented with something as simple as an XOR gate. However, it is more useful to implement it with a device which can detect whether the VCO signal is lagging or leading the reference signal. Such an implementation is described in [16]. The phase detector outputs pulses (whose widths are dependent on the phase difference between the VCO and reference) which are then fed to a charge pump. The charge pump is a current source whose operation is to charge or discharge a capacitor. The charging or discharging states are determined by the existence and width of the pulses, and whether they are charge up or charge down in origin, determined by whether the VCO is lagging or leading the reference. The capacitor voltage is then filtered by the loop filter and is passed to the VCO as the control voltage. This control voltage is slowly changing due to the filtering action of the loop filter, and is fed directly to the control transistors, described earlier, in the VCO. We see that this arrangement of loop components forms a closed loop and is one form of a PLL. The block diagram for this PLL is shown in Figure 23.

This implementation of a PLL cannot directly be applied to CCPLL's, but could be modified to yield a feasible VLSI solution to the CCPLL. One of the key points of operation of the CCPLL is that they cancel out the interfering signals at the input to the opposite PLL. As described previously, the cancellation is done through a subtraction operation. The previous cases dealt with a mixture of sinusoids which
were not limited. When an estimate of the desired input signal was formed from a PLL, it was a sinusoidal signal which could be subtracted from the mixture at the input to that PLL. This is a linear operation. However, the process of limiting a signal is a non-linear operation. If the input mixture is limited, especially in the case when the individual signals are of comparable amplitude, it is impossible to subtract away the undesired component using the square wave VCO output from the limited signal. The process of limiting the input signal is also detrimental in noisy situations. Although the output power of a limiter is constant, under a low SNR the output is almost entirely noise power. The process of limiting in the presence of noise causes jitter in the zero crossings of its output, which is especially crucial in phase detector which operates on zero crossings.

Two CCPLL structures are proposed that can be implemented in VLSI. One will use components found in digital circuits and use the PLL described above, and one will be a combination of digital and analog components. The former is easily
implemented in VLSI, and uses a limiter in such a way to allow for the cancellation of interfering signals despite the non-linear limiting operation. The latter avoids the use of a limiter, and hence is better suited to noisy operating conditions. Both implementations described here use the CCPLL without amplitude control. This is for simplicity's sake in understanding the ideas presented. Practical implementations would include an amplitude estimation circuit as well, whose components can be implemented in VLSI. The first implementation will now be described.

6.1 Proposed VLSI CCPLL- Implementation with Limiter

A proposed modification to the PLL structure described above is shown in Figure 24. The modification involves subtracting the VCO signal from the mixture before the non-linear limiting operation is performed. This change allows the CCPLL to be implemented with the digital components as above with modifications. An adaptive filtering block has been proposed as an addition to the CCPLL. This filter will take the form of a tunable band pass filter at the input of each individual PLL in the CCPLL system. The adaptive filter will track the desired input signal with a control signal derived from the PLL. The same signal that controls the VCO frequency of oscillation would be a logical choice as it capable of tuning the center frequency about its original value just as it does with the VCO frequency. This will serve to enhance performance by reinforcing the CCPLL action. Incomplete cancellation of undesired signals can be filtered with the band pass filter. It also serves as a way to filter out noise that may be present. If the frequency of the signal being tracked changes, the
PLL and the tunable filter both follow the change. It is necessary that the filter bandwidth is wide enough to allow for the non-stationarity of the input signal, so the desired signal is not filtered out accidentally at the input to the PLL. It also would give added noise suppression, as noise outside the filter’s bandwidth cannot reach the PLL.

![Block diagram of CCPLL system in form to be realized in VLSI](image)

Figure 24: Block diagram of CCPLL system in form to be realized in VLSI

As was noted in an earlier discussion on phase detector implementation, the frequency characteristics of a square wave VCO signal seen from a Fourier series expansion contain a strong sinusoidal component at the square wave frequency and then successively weaker components at odd harmonics. If the square wave output is subtracted from the input mixture before it is limited, the sinusoid at the VCO
frequency can be canceled if it is of the appropriate amplitude. For this type of phase
detector, the VCO signal and the input signal are in phase when the loop is in lock.
Therefore, an inversion of the VCO signal must take place to cancel the interfering
signals. One problem that must be addressed by adding an inverter is the time delay
it introduces and how to counteract that in the circuit. This is especially true for
high speed implementations.

If the 3rd and higher order harmonics can be filtered out or if their effect on
the circuit is minimal or non-existent, subtracting the square wave from the mixture
will yield the same performance as in the original analog CCPLL in cancelling the
interfering signals. The higher order harmonics can easily be filtered out by the
tunable band pass filter. If simplicity of circuit design was a major issue, the tunable
filter could be left out if the amplitudes of the higher order harmonics were small
enough such that the capture effect would ensure the desired signal was tracked and
demodulated, and not any of the harmonics. This scenario would still leave such an
implementation of the CCPLL suitable for fabrication in CMOS, but without the
enhanced functionality of the adaptive filtering.

The remaining blocks in Figure 24 will now be explained. The limiter is used to
transform the continuous sinusoidal wave into the discrete form which can be ma-
nipulated with digital circuitry. The PLL’s themselves can be of the form described
earlier, for which designs are already in progress. They consist of a phase detector
which outputs pulses that are realized based on whether the phase difference between
VCO and reference is lagging or leading, a charge pump to charge/discharge a ca-
pactor, a loop filter to filter the capacitor voltage, and the square wave VCO. The summer operation is the same as in the analog case. It would be beneficial for the summer to have adjustable gains at the inputs, as this may be necessary for proper operation, and are shown in the block diagram. Future implementations should be able to incorporate an amplitude estimation circuit similar to that used in the analog CCPLL with amplitude control simulated earlier.

This proposed circuitry will be able to take advantage of a digital component implementation which is well suited for VLSI technology. As before, for the system to work, one of the PLL's must be able to lock onto a signal to start the coupling and cancelling effects. This will be the case when the input mixture consists of a strong signal and a weaker signal where the capture effect will allow proper operation. Since the functionality is similar to that of its analog counterpart (with improvements), it is reasonable to expect that it will be able to lock onto one of the signals first in the scenario where the amplitudes are comparable and the frequencies are far enough apart. Recall that the analog CCPLL was able to work properly for each case except that of identical amplitude, identical carrier frequency. If the signals were “far enough” from each other in amplitude while at the same carrier frequency and vice-versa, proper operation was observed.

6.2 Proposed VLSI CCPLL- Implementation without Limiter

An alternative implementation of the CCPLL that could be realized in VLSI that mixes both analog and digital components will now be described. It will be some-
what more difficult to implement than the digital component loop described above in VLSI, because a phase shifter is now needed, but contains advantages in that the non-linear limiting operation is not needed. This CCPLL is better suited to operation in noisy environments due to the removal of the limiter. It also has the advantage over the original analog CCPLL in that it does away with the analog multiplier while still keeping the same functionality. Adjustable gains are needed for the summer inputs as amplitude control has not been added, and is a must for future practical implementations of the CCPLL. A block diagram for this proposed VLSI implementation is shown in Figure 25.

![Block diagram of CCPLL system in form to be realized in VLSI, another implementation](image-url)

Figure 25: Block diagram of CCPLL system in form to be realized in VLSI, another implementation
This mixed analog/digital component CCPLL uses a VCO of the square wave type. As mentioned before, its output contains a strong sinusoidal component at the first harmonic of the VCO frequency. When the PLL in question is in lock, this VCO will lag the desired input signal by 90 degrees, not be in phase as in the previous case. Hence, a 90 degree phase shift is needed to achieve the desired cancellation as in the analog CCPLL case. The main difference is in the phase detector, which now uses a simple switch. In the earlier discussion on phase detectors, it was shown that changing the polarity of the input each half cycle (in terms of VCO frequency) yields the same results as multiplying the input signal with a unit amplitude square wave from the VCO. Multiplying the input by a square wave would have the exact same functionality as multiplying it with a sinusoid of the same frequency, as it has a strong sinusoidal first harmonic. If the higher order harmonics resulting from this multiplication are filtered out by the loop filter or insignificant enough for the capture effect to take place, the functionality is the same as using an analog multiplier. Using this type of phase detector gives the same functionality as multiplying by a pure sinusoid, without the use of an analog multiplier. The VCO signal is used to time the switch that alternates polarity of the input signal. References [4] and [10] contain descriptions of such a phase detector.

If the 90 degree phase shift can be realized, then this CCPLL system will have the same functionality as the analog CCPLL. One way in which this phase shift can be realized in VLSI is to use a band pass filter. In the frequency domain, a 90 degree phase shift has a positive 20 dB/decade slope. By offsetting the center frequency of a
band pass filter from our desired frequency of operation, we can realize the phase shift by operating in the frequency range where the band pass filter has +20 dB/decade slope. Positive or negative shifts could be realized with this method depending on which side of the desired frequency the filter center frequency is offset. It is obvious that a 90 degree phase shift cannot be realized over all frequencies using this method (or any method for that matter). A band pass filter can be designed along with the CCPLL for a specific application so the phase shift will be realized in the appropriate frequency range. Alternatively, a tunable band pass filter can be implemented. This will not only allow for changing of the center frequency for flexibility’s sake, but also to allow for tolerances in manufacturing the VLSI chip, which are never ideal. The reduction of amplitude of the VCO signal if using a band pass filter must be addressed, as the amplitude is very important to achieve proper cancellation at the opposite PLL.

One consideration for implementing either of the above proposed circuits in CMOS is the tolerance allowed for manufacturing errors as no chip fabrication is exact. Most of the blocks could be designed with less than ideal parts and still maintain functionality. These blocks include the filters, VCO, and phase detectors. Blocks needed for the cancellation of the interfering signals are more critical. However, two factors are working in our favor with this type of design. First, the adaptive filters will help make up for an inexact cancellation by filtering out these undesired signals. Second, the fact that we are using a feedback loop is beneficial as the CCPLL can automatically adjust for small errors as it is trying to force the error signal to zero.
anyway. PLL's have been known to track out errors originating from the VCO itself, just as they would with outside noise. Hence, even these blocks may have a wider tolerance than expected. In any case, more attention should be spent on the proper design of the summer and phase shifter, as their performance is perhaps more crucial to good PLL performance than the other components.

### 6.3 Summary

This chapter introduces work under way to implement a PLL in CMOS VLSI. A description of the circuit and its principles is given. To realize a practical, cost effective CCPLL, one must be able to take advantage of VLSI technology. Hence, two novel CCPLLs are introduced which maintain the functionality of the discrete CCPLLs described in this thesis. Enhancements are proposed to the discrete CCPLL structure which lead to better noise performance and allow for inexactness in the manufacturing process.
CHAPTER VII

Conclusions

7.1 Summary of Work

This thesis has focused on phase-locked loops and the steps needed to couple them so they may track multiple sinusoidal signals. First, some of the background theory necessary to understand PLLs was covered. The components of the PLL and their relation to loop performance were discussed. Equations governing the PLL performance for the linear model were given and discussed. PLLs were designed based on the linear model. Simulations were performed for the single PLL case to verify that Simulink was an acceptable method to simulate PLLs. Then, the existing methods of coupling PLLs were discussed. Simulations were performed on existing CCPLL designs to verify that Simulink was able to simulate a CCPLL. A measure to evaluate CCPLL performance, the cross channel interference ratio, was introduced. CCPLLs were introduced that can track and demodulate PM signals, and the conditions for proper operation derived. New CCPLL circuits were proposed with enhancements to improve tracking and lend the design to implementation in CMOS VLSI using designs already in progress.
7.2 Summary of Conclusions

It was determined by way of comparing simulation results with the expected results that Simulink is able to simulate both PLLs and CCPLLs effectively. We found that the step size of the simulation can have a profound effect on the results if chosen too large, especially in a noisy simulation. Various simulations were done and support the previous work in the literature [1, 2] in showing that a CCPLL is able to track multiple FM signals. It has also been demonstrated by simulation that cancellation of interfering signals can be achieved by comparing the frequency demodulated outputs in both the CCPLL with amplitude control and that without. Excellent cancellation was achieved over a wide range of differences in carrier frequencies.

In the literature, no cancellation performance measure has been identified. In this thesis, the cross channel interference ratio (CCIR) is proposed which forms a ratio of the power of a particular modulating signal type as interference in the CCPLL to the power of that same signal at the desired demodulated output. We conclude that the CCIR is one valid measure of interference cancellation performance, as it gives a direct comparison of the level of the phase as interference in the CCPLL to the level of the phase where it is desired in the CCPLL. Simulations were performed to generate data for the CCIR. It was established that the cancellation performance is excellent when the frequency difference of the input carriers is greater than 100-200Hz for FM signals. The performance as indicated by simulation was poorer than expected for carrier difference less than 100-200Hz. However, the signal powers are identical so that the capture effect cannot be used to start the coupling action. A simulation was
performed which demonstrated the CCPLL's ability to cancel interference when one of the signal amplitudes was reduced by a factor of ten.

Conditions were derived for the use of a CCPLL as a small modulation index phase demodulator when the demodulated output is taken after the phase detector. In certain applications, the use of this type of PLL may be beneficial and fully functional. If the use of an integrator in phase demodulation is undesirable and the difference in carrier frequencies is large relative to the sum of the individual phase bandwidths, then the CCPLL is able to track, demodulate, and cancel interference as desired. The CCIR is demonstrated for PM signals and results comparable to the tracking of multiple FM signals achieved.

CCPLLs need to be implemented cheaply and compactly, especially when the circuitry becomes more complex in the case of $M > 2$, where $M$ is the number of PLLs in the CCPLL system. One cost effective way of implementing a CCPLL is through VLSI implementation. Circuits were proposed to make this a possibility with existing technology. One circuit was designed to take advantage of a VLSI PLL already in the design stages, while the other was designed for better noise performance. It is recommended that the CCPLL structures as described in [1, 2] be modified to include a tunable band pass filter before each loop that follows the tracking of the VCO. Improved noise performance is possible. The band pass filter also allows for the possibility of using inexact parts. If incomplete cancellations occur or other unwanted frequency components appear because some of the PLL components stray from their ideal functionality, adaptive filters would serve as a way of compensating for the errors introduced in this fashion.
7.3 Areas of Future Research

The CCPLL shows promise as a means of tracking multiple sinusoids. However, more work needs to be done to yield a practical CCPLL system. The noise performance of the CCPLL with amplitude control must be established. The ability to track more than two sinusoids must be established more clearly by way of simulation and through the testing of physical hardware. Stability of the CCPLL with amplitude control must be addressed. The complexity of the amplitude control CCPLL as proposed in [11] must be reduced for \( M > 2 \) to make for a realistic implementation. Implementation may be possible by forming only one amplitude estimate for each signal and using that for cancellation at each of the remaining \( M-1 \) inputs rather than generating \( M-1 \) estimates of the same signal.

Acquisition must be a major consideration in future research. Results were presented in [11] that demonstrate acquisition time vs. filter bandwidth in the amplitude control loop. Still, more investigation into the aided acquisition problem will be necessary for practical implementation, especially when the CCPLL will be used in high frequency applications (as in digital communications). We see that there exist a number of interesting research problems in the area of CCPLLs that can stimulate research for years to come.
Bibliography


