THE COOPER UNION FOR THE ADVANCEMENT OF SCIENCE AND ART

ALBERT NERKEN SCHOOL OF ENGINEERING

Adaptive Phased Locked Loop for Interference Mitigation

by

Kevin Nguyen

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering

09/10/2015

Professor Sam Keene, Advisor

THE COOPER UNION FOR THE ADVANCEMENT OF SCIENCE AND ART

ALBERT NERKEN SCHOOL OF ENGINEERING

This thesis was prepared under the direction of the Candidate's Thesis Advisor and has received approval. It was submitted to the Dean of the School of Engineering and the full Faculty, and was approved as partial fulfillment of the requirements for the degree of Master of Engineering.

Dean, School of Engineering - 09/10/2015

Professor Sam Keene - 09/10/2015

Candidate's Thesis Advisor

Acknowledgments

I would like to thank my advisor, Professor Same Keene, for putting up with me and guiding me through a proper thesis. In addition, I would like to thank Professor Fred Fontaine for giving me the inspiration for this topic. He said, ”Just stick the word adaptive in front of something and out comes a Master’s thesis.” I followed his advice and generated this topic.

Abstract

Due to an increase in wireless devices, the available spectrum in ISM bands is highly contested. Interference becomes a rising issue especially with a higher density of devices transmitting over WLAN and Bluetooth. Joint detection is one proposed method to handle high levels of interference, but current implementations assume that receivers will receive ideal constellations without any distortion from the RF front end. This work discusses an adaptive approach to the receiver front end phase locked loop that can be implemented to facilitate joint detection.

Contents

List of Figures

List of Tables

1 Introduction

Modern age wireless communications have rapidly increased in popularity in the past couple of decades. With increased wireless traffic, there are many more signals in the air than when wireless communication techniques were developed. Due to this, many practices were developed based on the assumption that the desired message signal could be isolated through filtering on the RF front end. With an abundance of signals now occupying the same bandwidth, these techniques need to be revised to accommodate the increase in wireless traffic.

A lot of research has been put into interference mitigation and many approaches have been from the digital processing side. One technique is TDMA which schedules users in different time slots of transmission to avoid interfering with each other. The more users there are, the longer each user has to wait before transmitting. This decreases the achievable data throughput in the channel. [4]

Another technique that is employed is CDMA which allow users to transmit in the same bandwidth by encoding the message with a pseudorandom chipping sequence. [4] As a result, instead of experiencing continuous wave interference, each user sees a decrease in signal to noise ratio (SNR) due to the pseudorandom code. The reduced SNR increases the bit error rate (BER) in the channel thus decreasing the overall throughput.

Each of the mentioned techniques involve separating users into different slots to transmit data. Doing so decreases the maximum potential data throughput, but allows users to communicate simultaneously. A restraint to these solutions is that all of the users must agree to the particular solution. In many cases, there will be users in the same band as each other that are not complying to the same transmission scheme. This mainly occurs in the ISM bands which are open for public use. In this case, there are many devices with different transmission schemes including, but not limited to, modulation type, encoding architecture, and frequency of transmission.

New techniques need to be developed to deal with unknown interferers. In these scenarios, the receiver does not know any properties of the interferer. The system needs to be able to adapt to to the situation depending on the interferer's properties.

Early research into this field uses joint interference detection to correctly demodulate the message signal. [5] However, these techniques require knowledge of the interfering modulation scheme. In addition to the lack of modulation knowledge, typical receivers will not receive organized constellations to demodulated due to the RF front ends inability to lock onto the signal.

The objective of this work is to develop a RF front end that will be able to stabilize constellations of unknown interferers and allow the receiver to intelligently demodulate the signal. Specifically, there are improvements that can be made to the phased locked loop of the RF front end that allow the receiver to adapt to the rapidly changing environmental conditions of the wireless communications environment.

2 Background

Interference is a common topic of research as an increasing number of devices are using wireless communications as the main mode of communications. Prior to discussing the proposed modifications, interference and and modern solutions to interference will be discussed in the following sections.

2.1 What is Interference?

Wireless communications takes advantage of radio frequency (RF) waves transport data from one point to another. Most wireless transmitters are considered omnidirectional, meaning that their transmitting antennas radiate RF energy in every direction equally. Most receivers are also considered omnidirectional. Similar to an omnidirectional transmitter, this means that the receive antenna will experience an excitation regardless of the angle of arrival of the transmitted signal. Ideally, there is only one transmitter generating a signal for the receivers to process. Due to the omnidirectional nature of both devices, the receiver can be located any where with respect to the transmitter. When multiple transmitters are present in an area, there is a possibility that a receiver will pick up signals from both transmitters as seen in Figure 1. [4]

In the pictured case, three transmitters are trying to communicate at the same time. The receiver pictured now captures the composite signal which results in a superposition of each of the three signals. This causes problems with decoding the intended signal. In the 802.11 standard, this is called a packet collision, [6] and there are protocols that are implemented to try and avoid this situation.

Wireless LAN is not the only protocol that should suffer from packet collisions. Any wireless standard with more than one user should encounter this problem. Products

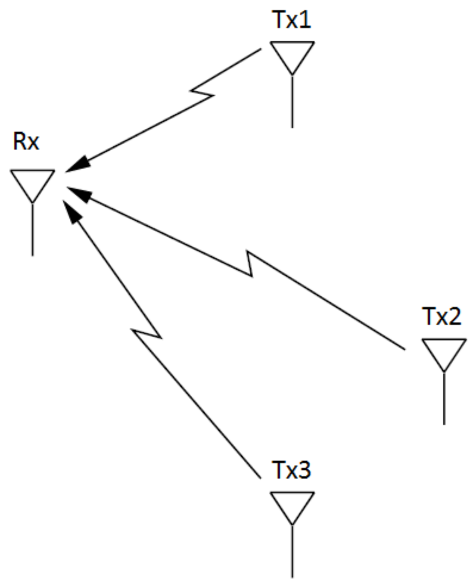


Figure 1: Example of Interference Condition

such as cell phones, radio, and satellites should also encounter collision problems with their data. However, these products condition their message signals prior to transmission in a particular manner prior to transmission to ensure that collisions can be avoided.

2.2 Modern Interference Mitigation Techniques

There are many ways that protocol were designed to combat interference. The popular techniques are Spatial Division Multiple Access (SDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Orthogonal Frequency Division Multiple Access (OFDMA). Each of these techniques involve dividing up a transmission property between users so that each user can transmit their information in their own transmission environment.

2.2.1 Spatial Division Multiple Access

Spatial Division Multiple Access utilizes directional antennas or antenna arrays to focus transmission energy into a particular area. This technique is effective when the intended receivers are sufficiently spaced apart. Suppose two transmitters wanted to communicate to two different receivers where the transmitters were at the same location and the receivers were spaced apart. If both transmitters were to broadcast their message omnidirectionally, then both messages would collide at both receivers and neither receiver would be able to decode the message. However if the transmitters focused their transmit power in the direction of the receivers, then less interfering power would arrive at the receivers and this provides better conditions for decoding. Power can be focused by using directional antennas such as satellite dishes, or through multiple element beamforming arrays. An example of this is in use with 802.11ac. [7] [8]

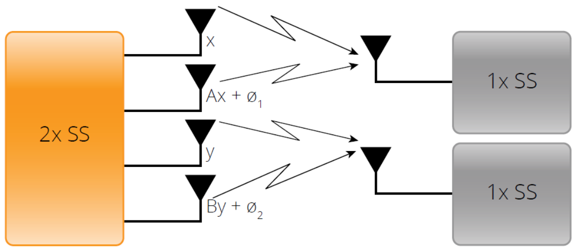


Figure 2: Example of Spatial Division Multiple Access

Figure 2 shows an example of a two receiver system with separate data streams. The transmitter in this example has 4 antenna elements that are used to send two independent signal streams (SS) to a respective receiver. The transmitter allocates two antennas to each data stream to beamform to the correct receiver. If done correctly, the receivers should only receive one message and not experience packet collision or interference. This technique can be used when there is sufficient separation between receivers and the transmitter has a method of forming a transmission beam. The transmission beam generally requires many antenna elements per user as each user needs a separate dedicated transmission beam. Another requirement is that the transmitters need the receivers to feedback data on the beamforming quality so that the transmitter can accurately set up the antenna element phase offsets.

2.2.2 Time Division Multiple Access

Time Division Multiple Access schedules each user to a different time segment to transmit. This creates a queue of users waiting to transmit over a certain channel. One standard that utilized this form of multiple access was the 2G cell phone standard, specifically GSM. [9] In the GSM standard, each user that wants to transmit in a given frequency channel must schedule their transmission time with an allocated base station. The base station assigns a particular time slot to each user that requested to transmit and each user transmits in succession. The time slots are defined within the specification so both the base station and the user know how long the time slots are and how many time slots there are in each transmission frame.

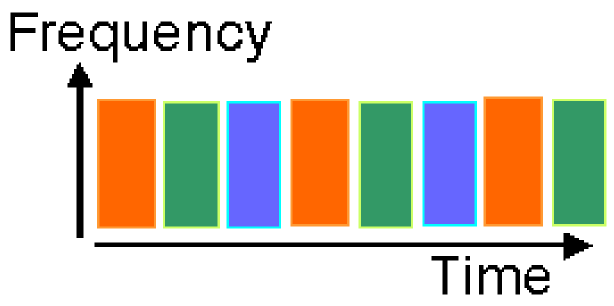


Figure 3: Illustration of Time Division Multiple Access [1]

Figure 3 illustrates how TDMA systems are organized. Each colored block represents a different user and each user is transmitting over the same frequency bandwidth. As previously described, each block is allocated a different time segment to transmit over. For example, the orange user is transmitting in time blocks 1, 4 and 7. Another characteristic of TDMA is that between each time block, there is a guard interval that allows minor timing errors between users during transmission. This is especially important when users in the system are moving as this influences the total travel time required for the signal to reach the base station. Although the guard interval allow for some timing errors to exist, the transmitter-receiver system must be tightly synchronized in time to successfully avoid interference.

2.2.3 Orthogonal Frequency Division Multiple Access

Orthogonal Frequency Division Multiple Access is an implementation to Orthogonal Frequency Division Multiplexing (OFDM) which allocates a set of orthogonal carriers to different users for transmission. The requirement for two sub-carriers to be orthogonal is to space the sub-carriers by: [4]

$$\Delta f_c = \frac{k}{T_n}$$

where Δ f$_c$ is the spacing between sub-carriers, k is a positive integer, and T$_n$ is the duration of a symbol. The minimum frequency spacing can then be defined as $\frac{1}{T_n}$. In a system that uses OFDM, the individual sub-carriers do not need to be filtered since during demodulation the other adjacent orthogonal carriers will be suppressed during the demodulation.

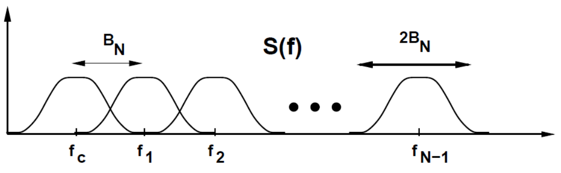


Figure 4: Illustration of Orthogonal Frequency Division Multiplexing

Although the adjacent sub-carriers are overlapping, the principle of orthogonality allows the system to still demodulate each channel independently. Modulation and demodulation of an OFDM signal can be done through the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). Each frequency is considered to be one of the

FFT sub-carrier frequencies. For each sub-carrier, a modulated constellation point is assigned to that frequency that determines the phase and amplitude of the particular sub-carrier. Once all of the coefficients are assigned, an IFFT is taken to get the time domain signal for transmission. On the receiver end, the receiver performs an FFT to return the frequency coefficients which contain the phase and amplitude information. From this information, the data signal can be retrieved. A block diagram of a transmitter and receiver can be seen in Figure 5

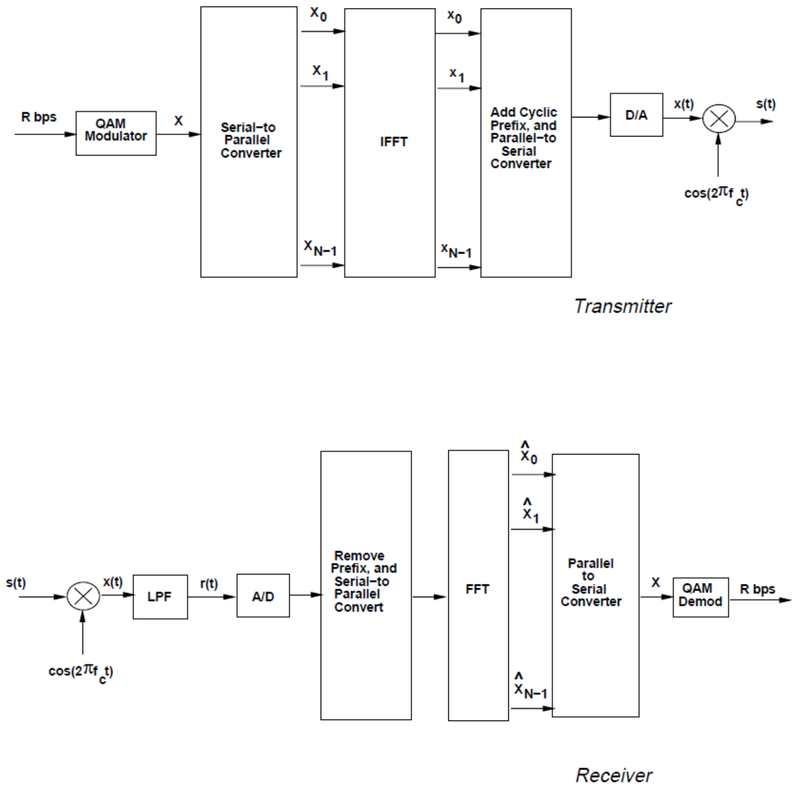

R bps
QAM Modulator
X
Serial−to Parallel Converter
X0
X1
XN−1
IFFT
x0
x1
xN−1
Add Cyclic Prefix, and Parallel−to Serial Converter
D/A
x(t)
s(t)
cos(2πfct)
Transmitter
s(t)
x(t)
cos(2πfct)
LPF
r(t)
A/D
Remove Prefix, and Serial−to Parallel Convert
FFT
X̂0
X̂1
X̂N−1
Parallel to Serial Converter
X
QAM Demod
R bps
Receiver

Figure 5: Block Diagram for OFDM Transmitter and Receiver

OFDMA takes this principle and allocates each tone to different users. The orthogonal sub-carrier tone spacing principle remains the same but when the receivers decode the message data, they only use the portion of the received data that was designated to them.

Figure 6 shows an example allocation scheme used in OFDMA. Each of the deltas represent a different sub carrier with the larger peaks called Pilot Subcarriers. These pilot

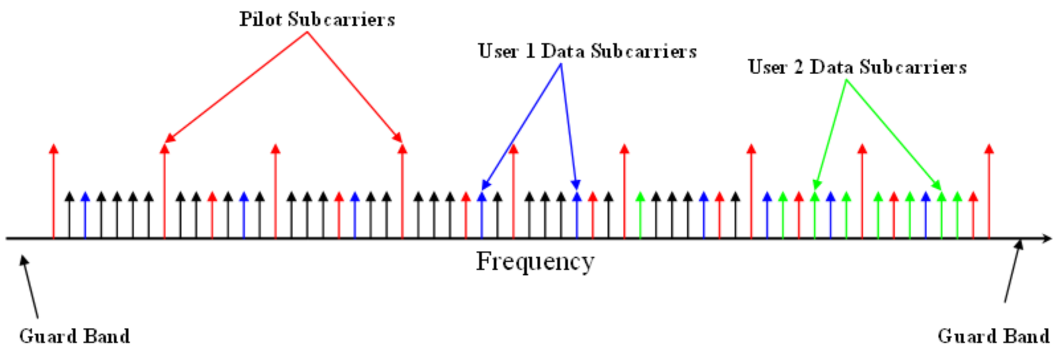


Figure 6: Subcarrier Allocation for OFDMA System

sub-carriers are used to maintain accurate frequency lock between the receiver and transmitter. Both parties know the frequency at which the pilot tones should be, and the rest of the sub-carriers can be found in relation to these known signals. Due to the necessity for accurate frequency lock, OFDMA is susceptible to frequency drifting. Frequency drift is a change in frequency of the carrier due to environmental effects. Temperature changes can cause the local oscillators of both the transmitter and receiver to to drift in frequency causing errors in both the generation and receiving of the signal. Motion can also cause a drift in frequency in the form of a Doppler shift.

OFDMA allows users to utilize the same bandwidth by allocating specific sub-carriers to each user. In order to successfully implement this scheme, the users and the transmitters must have the same sub-carrier spacing and must maintain frequency lock.

2.2.4 Code Division Multiple Access

The last of the common multiple access schemes involves encoding the message signal with a code from a set of pseudorandom codes to achieve direct sequence spread spectrum (DSSS). Each of these codes are uncorrelated with each other. In other words, the correlation between a given code to any other code in the pseudorandom set of codes if approximately 0. [10] The pseudorandom codes have much smaller bit times than the data signal bits. To apply a pseudo random code, the code and the data signal are XOR'ed together. Illustration of this can be see in Figure 7.

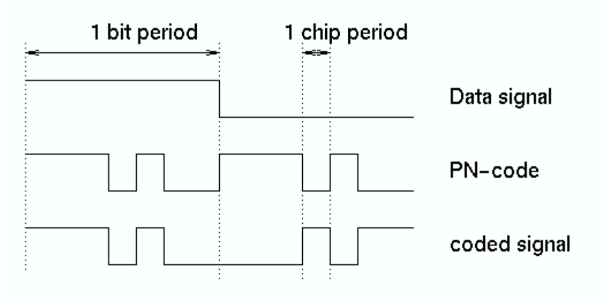


Figure 7: Illustration of Chipping in CDMA [2]

By XOR'ing the code with the message, the message bandwidth increases due to the increased rate of bit changes. The encoded data signal therefore occupies a larger bandwidth, but has a lower amplitude over the bandwidth. The total energy of the signal is the same, but it is spread over a larger bandwidth. What this does is make the signal appear to other users as noise. Receivers that have the correct code can correlate with the incoming signal and recover the encoded message. All users are transmitting simultaneously and since each code are uncorrelated, the correlator each receiver will only decode the desired message while suppressing the effects of the other transmitting signals. The ratio of code bit time to data bit time determines how many users can be simultaneously transmitting over a given bandwidth. More users that are trying to transmit means more codes need to be generated. A side effect of this is that the codes need to become longer in length which means that each data bit needs to be lengthened to fit the new longer codes. This effectively decreases the overall throughput for each user, but allows each user to communicate simultaneously.

As with the other multiple access schemes, all of the transmitters and receivers must be cooperating with each other to avoid interference. The users in each of these transmission schemes share their resources to allow all users to effectively use the given bandwidth resource. Systems can be designed for users to share space, time, frequency, power or a combination of any of the previous resources with each other to avoid interference. Systems that employ this have licensed bands where no user outside of the system may use their frequencies. This allows for a controllable transmission ecosystem for all of the users to cooperate within.

2.3 Bluetooth and WiFi

The ISM bands are a portion of the RF spectrum that is unlicensed. This means that anyone can transmit within this band as long as rules about power and bandwidth are followed. As a result, many users occupy these bands and the ISM bands have become crowded. One of the more prevalent bands is the 2.4 GHz band which contains both WiFi and Bluetooth traffic. Devices employing either of these two wireless protocols will not share transmission information with each other. It is up to the device to find the optimal operating conditions for ideal transmission.

2.3.1 Bluetooth

Bluetooth uses frequency hopping spread spectrum to avoid interference. [11] While CDMA, directly modifies the message signal with a pseudorandom code to achieve spectrum, For versions before Bluetooth 4.0, Bluetooth transceivers hop between a known list of 79 different 1 MHz channels to reduce the total average power at each channel. If interference is detected in a specific channel, the transceivers will hop to the next 1 MHz channel and try again. In a crowded area, there can be many Bluetooth devices present and a transceiver pair could end up hopping channels for an extended period of time.

Spread spectrum was effective in the case of CDMA because all of the users in the band cooperated by agreeing to a set of psudorandom codes. Bluetooth does not have this luxury. Devices are paired to a host with the host denoted as the master and the device as the slave. The master slave pair then agree on a channel hopping scheme and begin to transmit. Each master can have up to 6 slaves, but individual masters will not communicate with each other. There can be many different groups of Bluetooth devices attempting to communicate, and since not all devices are communicating with each other, this can lead to many interference problems.

2.3.2 WiFi

WiFi uses DSSS, similar to CDMA, or OFDM for all of its users in a network. There are 11 overlapping channels in the 802.11 standard, but only 3 different networks can exist at a time without overlapping. [6] When there are more than 3 networks, interference between networks becomes a problem. In 802.11 when wireless networks encounter interference, the response is to decrease the datarate and try again. The underlying assumption is that the packet error was caused by insufficient signal to noise ratio (SNR). This effectively increases the time it takes to transmit the same information. When all networks are reducing their data rates, all of the networks will spend more time transmitting This results in an increase in the likelihood for additional interference condition which is the opposite of the desired outcome.

The situation is similar to Bluetooth. There are many independent networks attempting to transmit in a limited bandwidth and no communication between individual networks. The problem becomes worse when both wireless protocol are operating in the same area.

2.3.3 Interference between Bluetooth and WiFi

Both Bluetooth and WiFi occupy the same range of frequencies. When one WiFi channel is active, it occupies 22 of the 79 Bluetooth channels. [12] This increases the odds of interference between the two devices significantly and when there are more than one network in the area, interference is immanent.

As mentioned in the previous section, the WiFi transmitter will decrease the data rate when WiFi encounters interference. [13] The increased WiFi transmission rate in conjunction with occupying 28% of the available Bluetooth channels means that in crowded areas interference will be prevalent.

To try and prevent this, research has gone into techniques to mitigate the effects of interference in both WiFi and Bluetooth.

2.3.4 Mitigating Interference in WiFi and Bluetooth

There are several approaches to avoiding interference for either WiFi and Bluetooth. One phenomena is the capture phenomena where given two simultaneous transmissions, there is a probability that one of the transmissions will be received. This phenomena only applies to two signals with a large difference in power so that one of the signals is more dominant than the other. A group at National University of Ireland explored the capture effect on overall throughput of an 802.11 system. [14] The probability of successful capture was dependant on the power ratio between the two interferers. The larger the power ratio, the better the probability of a successful capture. They considered any two transmissions with the same or near same power level to be a failed transmission.

Other groups apply multiple access techniques used in licensed bands to reduce the probability of interferers. One of these groups proposed for a "Smart Antenna" system to be used with WLAN. [15] This smart antenna system locates and beamforms toward a desired receiver. Any signals from other directions will be suppressed, therefore reducing the impact of interference and simultaneously enhancing the capture effect. TDMA would have to be used in conjunction with beamforming to support multiple users. New users to the network would also have to go through a period of overhead so that the router can identify the location of the new user.

Another group enhances Bluetooth's frequency hopping by analyzing the channel prior to transmission. [16] This technique requires an accurate estimation of the current active channels in both the Bluetooth and WLAN bands. After determining the active channels, this frequency algorithm restricts the specific pair of Bluetooth devices to the unoccupied bands resulting in shorter interference periods and less frequency hopping. The issue arises when there are so many users that no matter what channels the pair of devices selects, they will get a poor transmission channel.

2.4 Carrier Sense Multiple Access vs Joint Detection

Other than utilizing traditional multiple access techniques, researchers are working on methods that do not require complete cooperation from all users in a given channel. Two of these techniques are carrier sense multiple access and joint detection.

Carrier sense multiple access (CSMA) is a technique where the transmitter measures the total signal power in the channel prior to transmitting. If the power level is greater than a certain threshold, the transmitter considers the channel as occupied and will not transmit. This method is able to reduce the number of collisions and increase the capacity of a multi-user channel. [17]

One major problem with CSMA is the hidden node problem. CSMA assumes that the transmitter will detect similar signal powers as the intended receiver. There are situations where this is not the case and multiple users could still interfere with one another. Figure 8 depicts a hidden node transceiver topology that will result in interference. In this example, assume transmitter C has an existing communications link with receiver B. Transmitter A would like to transmit to receiver B, but first performs spectrum sensing to determine of there are any other users in the channel. The dotted circles around transmitter A and C show the transmission range of each transmitter. Transmitter C's transmitted signals do not reach Transmitter A at a significant power level for A to detect the existance of C. Since transmitter A does not sense any other transmitter, it will attempt to transmit data to B. As a result, there is a collision at transmitter B.

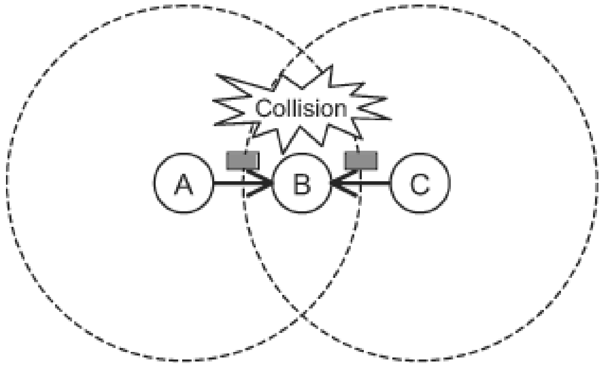


Figure 8: Illustration of the Hidden Node Problem [3]

What would happen if the system could demodulate the intended signal regardless of whether there is interference or not? The second method, joint demodulation attempts to accomplish this. Receivers that demodulate both the desired signal and the interfering signal are said to be performing joint demodulation. [5] Knowledge of the interferer is required to successfully accomplish this. Modulation scheme, frequency, and power need to be estimated for joint demodulation to work. These parameters are theoretically easy to estimate through digital processing if accurate constellations are used for the prediction. The problem is that the architecture of the RF front end was designed to recover a single carrier frequency and was not designed to operate with two different signals that are approximately the same frequency. When presented with a signal corrupted by an interferer, the phase lock loop attempts to recover the carrier signal and behaves unpredictably with the output signal. Details on the phase lock loop's behavior under these conditions is discussed in the later Section 4.2. The next section discusses a specific type of phase locked loop that performs carrier recovery, the Costas loop. Afterwards, the behavior of the Costas loop will be studied in the presence of a variety of interferers and a new architecture for carrier recovery will be proposed.

3 The Costas Loop

The Costas loop is an effective phased locked loop when used with constant amplitude modulation schemes. By putting feedback around two mixers, this architecture is able to directly down-convert and separate the in-phase and quadrature components of the received message signal. This section will describe the theory and basic performance of a Costas loop under various conditions.

3.1 Costas Loop Theory

The Costas loop, as with all other PLLs, can be broken down into three major components. [18] First is the error detector, which computes the phase error between the local reference signal and the carrier of the received signal. The output of the error detector is fed to the loop filter. The voltage output of the loop filter changes slowly depending on the output of the error detector. This voltage controls the last component, which is the voltage controlled oscillator. A voltage controlled oscillator outputs a sine wave with a frequency determined by the input voltage. Since phase and frequency are related by a derivative, the phase of a sine wave can be controlled by increasing the frequency for a phase lead and decreasing the frequency for a phase lag. The output of the VCO is fed to the error detector and used as the reference signal. A block diagram of the Costas loop can be seen in Figure 9.

3.1.1 Phase Error Detector

The phase error detector determines the phase difference between the reference oscillator in the receiver and the underlying carrier in the received message signal. First, the received signal is split using a power splitter. Each signal will be locked to the inphase

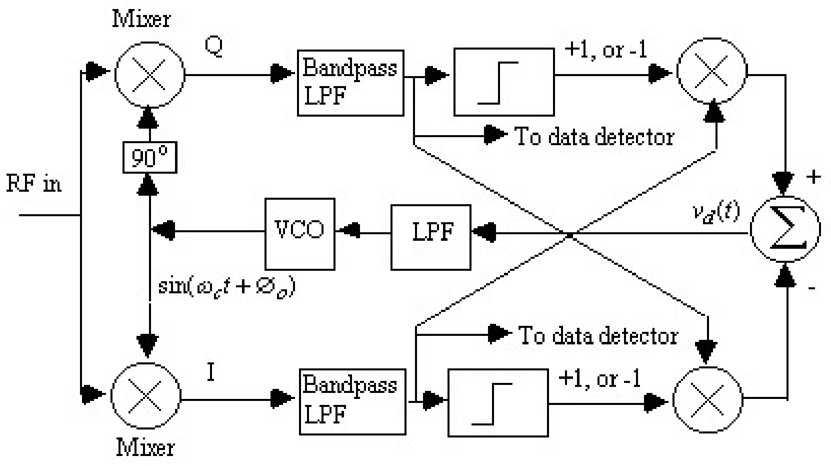


Figure 9: Block Diagram of a QPSK Costas Loop

and quadrature components of the message signal. This will be accomplished using two mixers driven by the same oscillator with a 90° phase offset on one of the branches. A stream of data is split up into two data streams and is encoded using QPSK. A QPSK message signal can be represented by:

$$m(t) = x_I(t)sin(2\pi f_c t + \phi) + x_Q(t)cos(2\pi f_c t + \phi)$$

Where ϕ represents a random phase offset from the local oscillator and x_I (t) and x_Q (t) are the message signals for the in-phase and quadrature branches respectively. In QPSK, m_I (t) and m_Q (t) can take on the values -1 or +1. After mixing the incoming signal with a sin wave (on the in-phase branch) or a cosine wave (on the quadrature branch), the resultant signals (m_I (t) and m_Q (t)) are as follows:

$$\begin{aligned}m_I(t) = m(t) * sin(2\pi f_c t) =& x_I(t)cos(4\pi f_c t + \phi) + x_I(t)cos(\phi)\\ &+ x_Q(t)sin(4\pi f_c t + \phi) - x_Q(t)sin(\phi)\\ m_Q(t) = m(t) * cos(2\pi f_c t) =& x_I(t)sin(4\pi f_c t + \phi) + x_I(t)sin(\phi)\\ &+ x_Q(t)cos(4\pi f_c t + \phi) + x_Q(t)cos(\phi)\end{aligned}$$

The low pass filter removes the higher frequency components of the signal and the resulting signals are

$$m'_I(t) = x_I(t)cos(\phi) - x_Q(t)sin(\phi)$$
$$m'_Q(t) = x_I(t)sin(\phi) + x_Q(t)cos(\phi)$$

The next component is a limiter which saturates the signal to either -1 or +1 depending on the voltage of the signal. Positive voltages will saturate to +1 and negative voltages will saturate to -1. Assuming that the phase error (ϕ) is small, the output of the limiters are dominated by the cosine term in each branch. Thus, the output of the limiters are the in-phase/quadrature data signals associated with the cosine term:

$$l_I(t) = x_I(t)$$
$$l_Q(t) = x_Q(t)$$

The limited signals are then multiplied by the filtered signals (m'$_I$ (t) and m'$_Q$ (t)). Then, the signals are subtracted to obtain the error signal.

$$m'_I(t)l_Q(t) = x_I(t)x_Q(t)cos(\phi) - x_Q^2(t)sin(\phi)$$

$$m'_Q(t)l_I(t) = x_I^2(t)sin(\phi) + x_I(t)x_Q(t)sin(\phi)$$

$$e(t) = m'_Q(t)l_I(t) - m'_I(t)l_Q(t) = x_I^2(t)sin(\phi) + x_Q^2(t)sin(\phi) = 2sin(\phi) \approx 2\phi$$

From this derivation, we can see that this phase error detector output is linear to the phase offset between the incoming modulated signal and the reference local oscillator. This error signal can be used to adjust the phase and frequency of the VCO and will be discussed in Section 3.1.3

3.1.2 Loop Filter

As derived in the previous section, the output of the phase error detector is linearly proportional to the phase offset between the reference and message signal. The derivation was performed under the assumption that the signal is noise free. In practical environments, noise is present in every part of the system and must be addressed. Noise present on the output of the phase error detector would effect the phase and frequency accuracy of the voltage controlled oscillator. A loop filter is used to minimize the impact of noise.

In addition to noise suppression, the loop filter impacts how quickly the voltage along the control line can change. Similar to control systems, quicker changes allow for a more rapid convergence time, but has significant steady state error. In the case of a PLL, this would result in a larger phase error when the loop is locked. Smaller allowed changes have longer convergence times but smaller steady state errors. Most loop filters are designed to be somewhere in between these two extremes.

3.1.3 Voltage Controlled Oscillator

The filtered error signal is fed into the voltage controlled oscillator. A VCO outputs a sine wave with frequency corresponding to a control signal. By varying the frequency output from the VCO, the PLL can sync the VCO with the received signal. Since frequency the frequency of a signal is proportional to the derivative of phase over time, varying the frequency varies the rate of phase change in a given time period. Thus, if the reference signal is lagging behind the received signal, the VCO can make up that phase by temporarily increasing the frequency to increase the rate of phase change and then decrease the frequency to the carrier frequency of the received signal. This technique allows for the Costas loop and phase loked loops in general to track received signals and allow for proper demodulation.

3.2 Fundamental Problems

The Costas Loop and many PLLs are designed based on the assumption that the signal of interest is a noisy modulated message signal. In today's environment, there are many RF signals in the air and the chances of receiving only the signal of interest is slim. There are several sources of interferences that will be considered and will range from other transmitters and interference generated by the channel itself. The following are the interference cases that will be discussed.

1. Another signal source interferes with the desired signal at a different carrier frequency.
2. Another signal source interferes with the desired signal at the same carrier frequency.
3. The desired carrier drifts in phase or frequency due to channel effects such as Doppler shifting.

These interferences are well studied and digital techniques have been developed to try to mitigate these effects. With additional signal sources introduced to the model, the receiver's PLL can no longer obtain a clean error signal. As derived in 3.1.1, the output of the phase error detector is supposed to remain constant given a phase offset value. In the first interference case, the mixer in the PED will correctly generate a DC voltage (after filtering) proportional to the phase offset. However, due to the additional signal present, the PED voltage will oscillate. To demonstrate this phenomena, both the message signal (m(t)) and the interference (I(t)) signal will be modeled as an unmodulated sine wave. The inphase component of the received signal can be calculated as:

$$\begin{aligned}
m(t) &= Asin(2\pi f_m t + \phi_1) \\
I(t) &= Bsin(2\pi f_I t + \phi_2) \\
r(t) &= m(t) + I(t) = Asin(2\pi f_m t + \phi_1) + Bsin(2\pi f_I t + \phi_2) \\
r_{I1}(t) &= r(t) * sin(2\pi f_m t) = Acos(4\pi f_m t + \phi_1) + Acos(\phi_1) \\
&\quad + Bcos(2\pi(f_I + f_m)t + \phi_2) + Bcos(2\pi(f_I - f_m)t + \phi_2)
\end{aligned}$$

Where ϕ_1 and ϕ_2 are a random phase offset for the message and the interferer respectively. After a low pass filter, the resulting signal is

$$r'_{I1}(t) = Acos(\phi_1) + Bcos(2\pi(f_I - f_m)t + \phi_2)$$

By viewing the result in the phasor domain, it can be shown that the error signal comprises of two parts. The first term is the expected constant error term that represents the angle between the carrier and the local oscillator. This is represented as a vector with magnitude A and angle ϕ_1. The second term is the term introduced by the interferer. In the phasor domain, this vector has magnitude B and a time varying angle which increases as a function of the difference in frequency betwen the carrier and the interferer. If the error signal were to be plotted in the phasor domain, the result would be a circle whose center is located at the amplitude and phase of the carrier and radius of the amplitude of the interferer.

Figure 10 shows a phasor domain plot where the red vector is the message carrier and the green dashed vector rotates around at frequency f_I-f_m. In the plotted example, the interferer has the same amplitude as the message carrier.

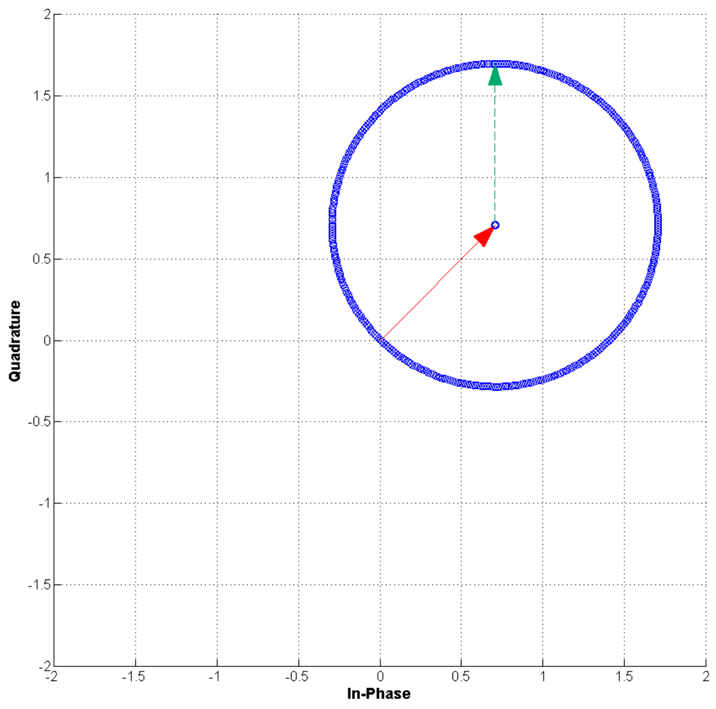


Figure 10: Phasor Domain Plot of Incoherent Continuous Wave Interference

The second case of interference is when the interferer is at the same frequency as the message carrier. The result from the previous interference calculation can be used by setting f$_I$=f$_m$. Doing so removes the time dependence from the error equation.

$$r'_{I2}(t) = Acos(\phi_1) + Bcos(2\pi(f_I - f_m)t + \phi_2)$$
$$r'_{I2}(t) = Acos(\phi_1) + Bcos(\phi_2)$$

The result is the sum of two constant terms that are defined by the incoming phase of each signal. This interference can also be visualized in the phasor domain as seen in Figure 11. The resultant phasor is the sum of the message and interferer phasor. As in the previous example, the message and interferer have the same amplitudes, but different phases.

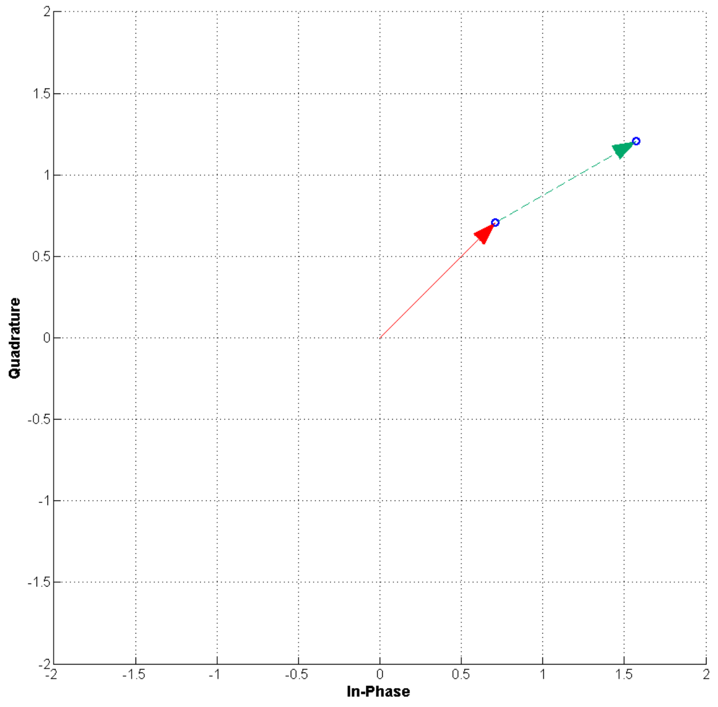


Figure 11: Phasor Domain Plot of Coherent Continuous Wave Interference

The third interference type is carrier drift due to channel effects. This is modeled as a frequency difference between the received carrier frequency and the expected carrier frequency. A frequency offset, f_O, will be introduced to the message signal.

$$\begin{aligned}
m(t) &= Asin(2\pi f_m t + \phi_1) \\
r(t) &= m(t)|_{f_m = f_m + f_O} = Asin(2\pi(f_m + f_O)t + \phi_1) \\
r_{I3}(t) &= r(t) * sin(2\pi f_m t) = Acos(2\pi(2f_m + f_O)t + \phi_1) + Acos(2\pi f_O t + \phi_1) \\
r_{I3}(t) &= Acos(2\pi f_O t + \phi_1)
\end{aligned}$$

The result is a time varying phase without a constant component. This is equivalent to a rotating vector anchored at the origin with magnitude A and frequency of rotation f_O. Figure 12 shows what this type of interference would look like in the phasor domain.

These three types of interferences will be applied to a receiver system that is phase locking using a Costas Loop and demodulating QPSK.

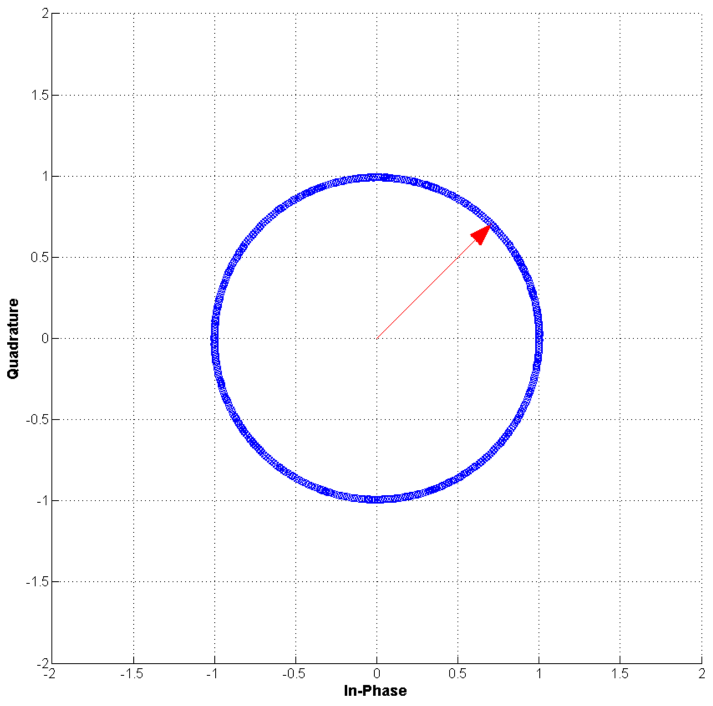


Figure 12: Phasor Domain Plot of Doppler Shift Interference

4 Simulation and Modeling

To study the performance of the Costas loop with different interference types, a simulation was built in MATLAB to mimic the analog system shown in Figure 9. Table 1 shows the list of parameter values used in the simulation.

Message Carrier Frequency (f_m)	3 MHz
Modulation Order	QPSK
Sampling Rate	10 MHz
Samples per Symbol	100
Number of Symbols	800

Table 1: Parameter Values Used in Simulation

Generally, simulating signal flow through an RF system to obtain accurate results using a scripting language is difficult. To ensure an accurate model of the system behavior, a fast sampling rate was chosen to accurately represent the received analog signal in a discrete processing environment. Each component in the system was modeled with an ideal mathematical representation of the components function. Effects on system performance using practical components will be discussed in a later section.

4.1 Simulation Architecture

The first section of the system is the RF signal processing components. The RF front end includes both a mixer and an image rejection filter. The mixer is represented by an ideal multiplier and the image rejection filter is implemented using an N-tap Finite Impulse Response (FIR) filter. These two components provide the received baseband signal to the error detector for phase error calculations. As calculated in Section 3.2, a low pass filter is needed to reject the higher frequency signal components after mixing. The cutoff frequency of this filter needs to be high enough in frequency to minimize distortion of the baseband signal but low enough in frequency to provide sufficient rejection at the higher frequency images at the output of the mixer. In simulation, the length of the FIR filter determines the cutoff frequency. The longer the filter, the lower in frequency the filter cutoff will be and vice versa.

The next section of the simulation involves phase error calculation. The block diagram shown in Figure 9 shows how the error is calculated. When simplified to a formula, the calculation can be represented by:

$$\Theta_{error}(t) = sign(Q(t)) * I(t) - sign(I(t)) * Q(t);$$

Where I(t) and Q(t) are the baseband Inphase and Quadrature signals from the output of the RF filter. The value of Θ_{error}(t) is zero when the magnitude of the Inphase and

Quadrature signals are equal. This error computation is specifically designed for QPSK modulation.

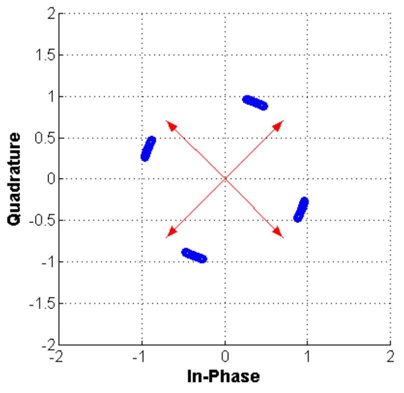
The error signal is then filtered and the filtered signal controls the VCO. Like the RF filter, the phase error filter is implemented using an FIR filter. The response of the filter controls how rapidly the error signal can change. Shorter filters, or filters with higher frequency cutoffs, allow the phase locked loop to react to phase and frequency offsets more rapidly. The frequency of the sine wave generated by the VCO is controlled by the filtered error signal. Generally, the VCO can be modeled to have a linear frequency response with respect to applied voltage. Different VCOs can have different frequency vs voltage responses. VCOs with larger response slopes mean that the small changes in the error signal will result in a larger frequency changes. This allows for faster corrections when there are any phase or frequency errors in the received signal. The VCO implemented in the simulation uses the error signal and linearly scales it by the VCO gain, K_{VCO}. The output sine wave from the VCO is then routed to the RF mixer to continue processing the next discrete time sample.

4.2 System Performance

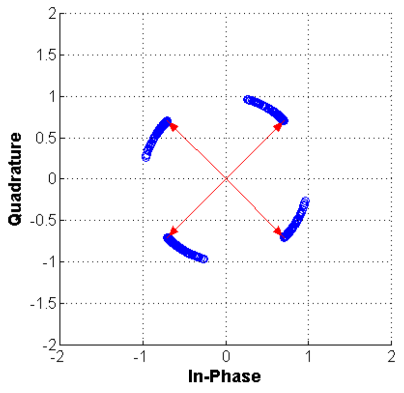
The performance of the Costas Loop needs to be established as a control prior to experimenting with the architecture. Three values of K_{VCO} were used to illustrate how the Costas Loop behaves with different loop parameters.

4.2.1 Phase Offset and Noise

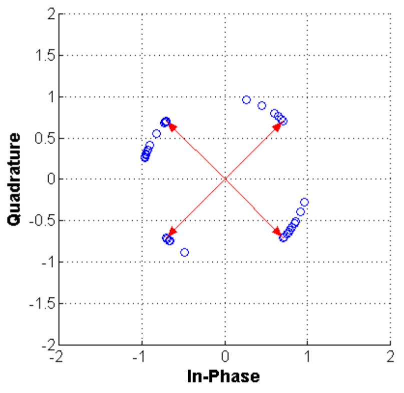
A basic communications channel introduces both noise and an unknown delay to the carrier. The role of the phase locked loop is to account for the unknown delay by phase locking with the carrier signal. Constellation diagrams and error signal plots will be used



(a) $K_{VCO} = 10$



(b) $K_{VCO} = 100$



(c) $K_{VCO} = 1000$

Figure 13: Constellation Diagrams of Costas Loop Behavior with Initial Phase Offset

to evaluate the system with different values for K_{VCO}.

Figure 13 shows the constellation diagrams of each system when an initial phase offset value of $\frac{\pi}{6}$ is used and no noise is present. 800 symbols were used to generate the plots.

Each blue circle plotted on the constellation represents a different symbol. The desired location for each of the 4 symbols is at the end of each of the 4 red vectors. The initial symbols start with a $\frac{\pi}{6}$ offset from the desired constellation points. As the value of K_{VCO} increases, the spacing between each symbol also increases as the constellation converges to the correct points. At the lowest K_{VCO} setting, the loop was not able to correct for the error in 800 symbols. To better analyze the convergence times, Figure 14 shows the error signal for each of the three values for K_{VCO}.

It is important to not that the when the message signal symbols change, small discontinuities will appear in the error signal by nature of the Costas loop. An example of this is shown in Figure 15 where the error signal is of the three K_{VCO} values are plotted. Both the plot shown in Figure 14 and Figure 15 show similar trends, but Figure 14 is clearer. Figure 14 was generated by forcing the message signal to have all the same symbols, which removes and symbol transitions. For clarity, error signal plots will be generated in a similar manner as mentioned above.

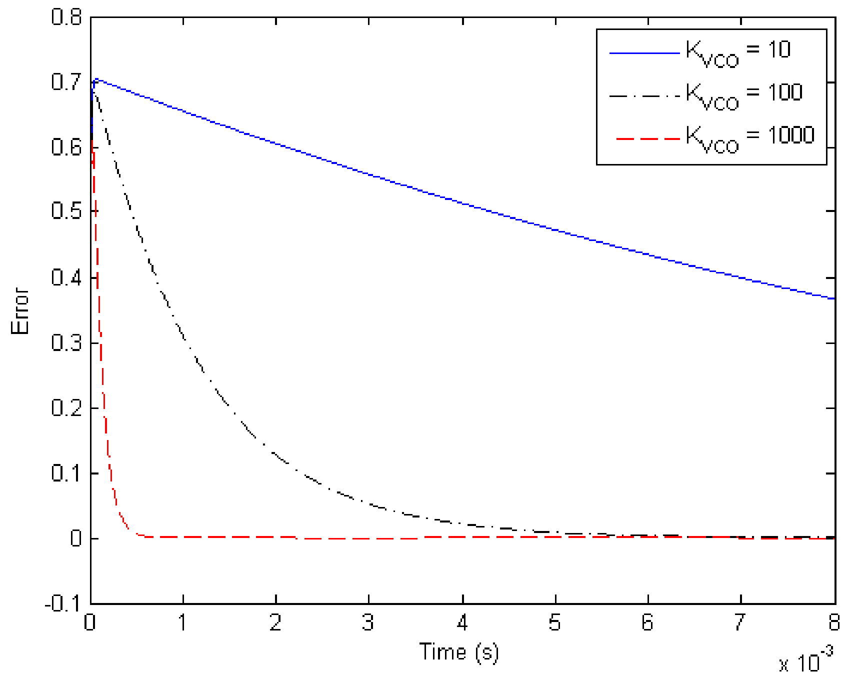


Figure 14: Error Signal of Costas Loop with Initial Phase Offset

It is clear from Figure 14 that larger values of K_{VCO} result in faster convergence times. If the channel only introduced an unknown delay to the signal causing a phase shift, then a larger value for K_{VCO} is desired. When noise is added, the constellation and error plots for $K_{VCO} = 100$ are shown in Figures 16a and 16b.

The introduction of noise blurs the response of the PLL and makes it difficult to describe and study trends that are occurring in the system. If Figures 16a is compared to Figure 13b, it is difficult to recognise that there was and initial phase offset. The error signal

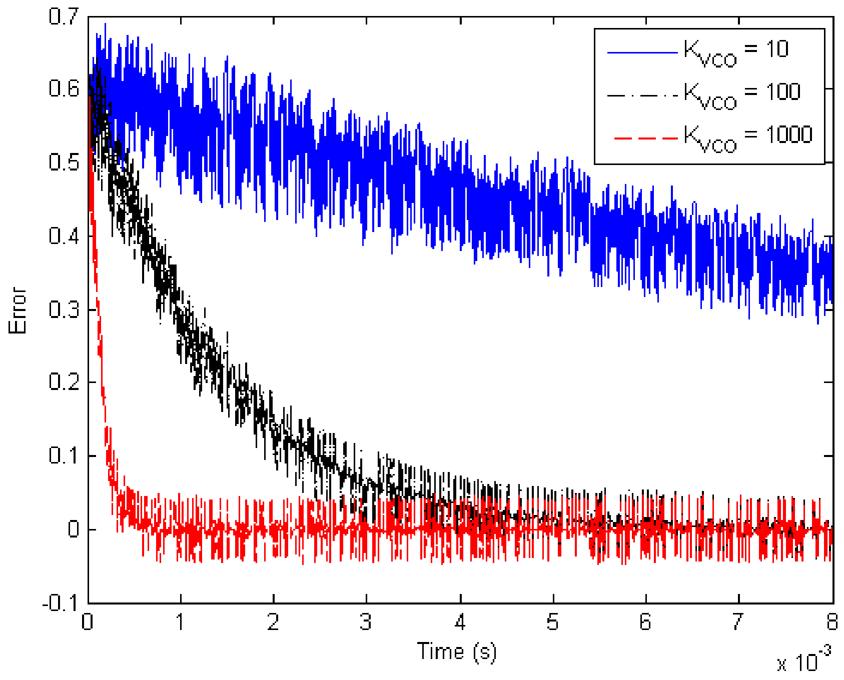
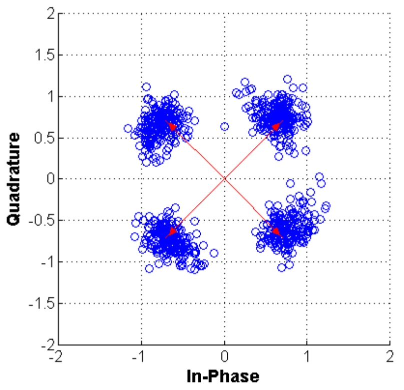


Figure 15: Error Signal of Costas Loop with Initial Phase Offset and Symbol Transitions

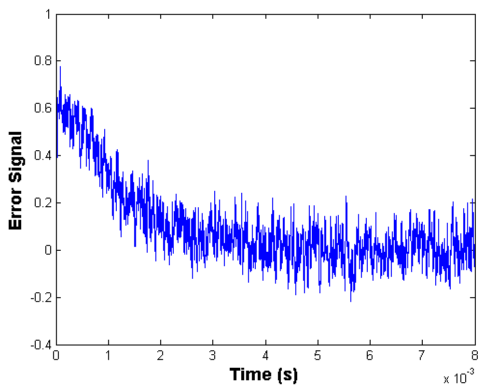
plots are also smoother and easier to analyze when tuning loop parameters. Therefore, noise will not be added for clarity when studying the other types of channel distortions and interferences.

4.2.2 Incoherent Interference

After verifying the operation of the phase locked loop simulation, interferers can be introduced to the system. The first interferer is an incoherent, unmodulated sine wave. The interfering signal's amplitude and frequency will heavily affect system performance. Larger amplitude interferers will affect the phase locked loop more than a lower amplitude interferer. To see how the phase locked loop is impacted, a large amplitude interferer will be used. The frequency difference between the interferer and the carrier frequency also have an impact on the PLL. If the frequency difference is large, then the RF filter will suppress the effect by filtering out the interfering signal. To maximize the stress on the



(a) Constellation of Phase Offset Performance with Noise



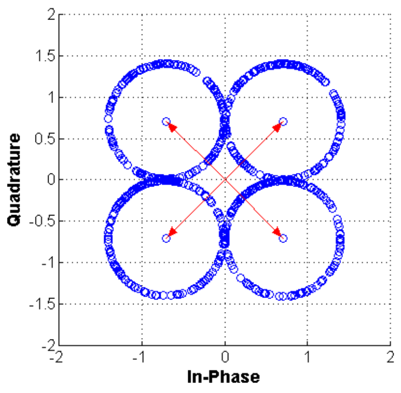
(b) Error Signal of Phase Offset Performance with Noise

Figure 16: Performance of PLL with Initial Phase Offset and Noise

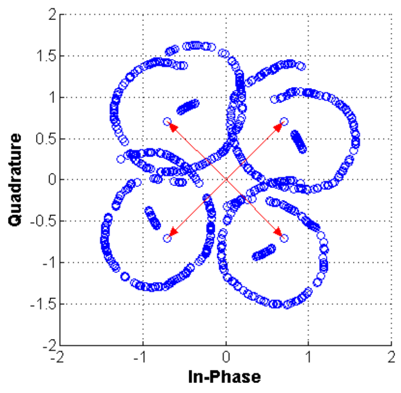
phase locked loop, the frequency difference will be kept small with respect to the carrier frequency. Similar to the tests performed in Section 4.2.1, the composite signal will be applied to the Costas loop with three different values for VCO gain.

In Section 3.2, this type of interference was studied mathematically to predict how the constellation would behave. It was predicted that an incoherent interferer would generate a circle around the desired constellation points with radius and revolution speed determined by the interferer. Comparing Figure 10 to the figures presented in Figure 17, it can be concluded that the behavior of the Costas loop when $K_{VCO} = 10$ closely resembles the predicted performance. At VCO gain of 1000, the constellation does not have any pattern and the behavior is difficult to predict. With an intermediate VCO gain of 100, the constellations are closer to the mathematically determined behavior, with some distortion.

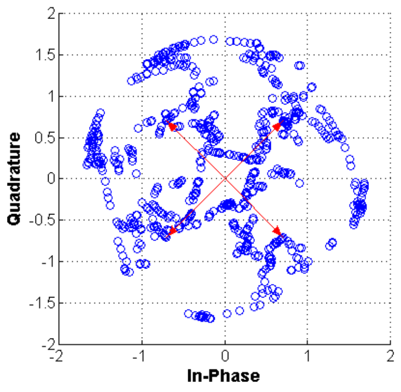
The most predictable constellation is obtained when the VCO gain is set to a minimum. When the interference is active, the phase locked loop attempts to correct for the interference by rapidly adjusting the error signal. The PLL is designed with the expectation that it is operating on one carrier frequency. When PLL attempts to downconvert a multi-tone signal, the result appears to be a random signal instead of a baseband signal.



(a) K$_{VCO}$ = 10



(b) K$_{VCO}$ = 100



(c) K$_{VCO}$ = 1000

Figure 17: Constellation Diagrams of Costas Loop Behavior with Incoherent Interference

The frequency correction factor for each value of K$_{VCO}$ is shown in Figure 18

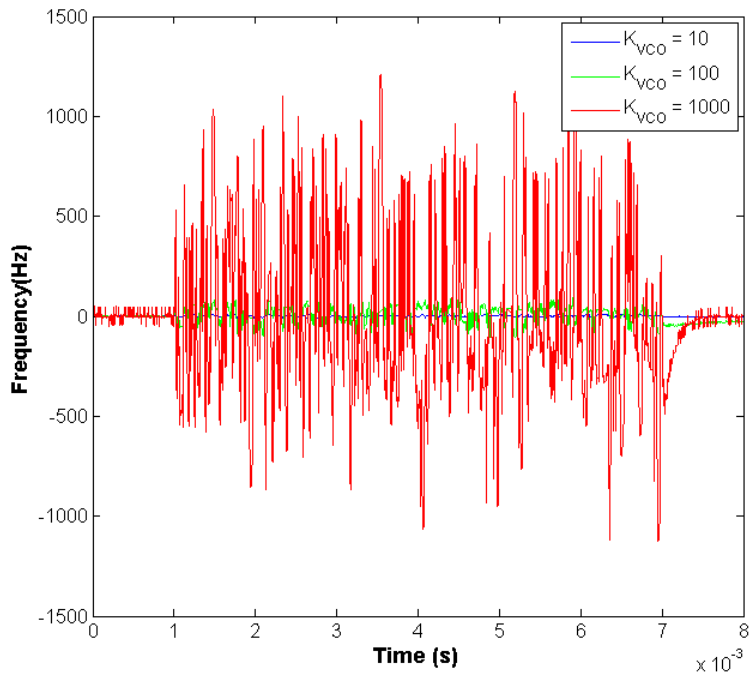
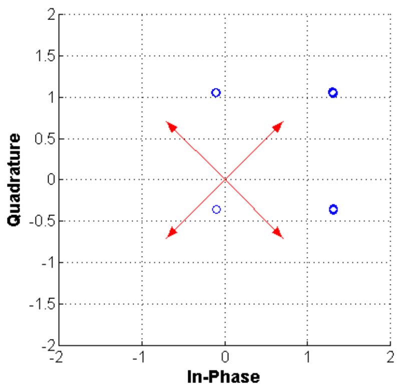


Figure 18: Frequency Correction vs Time with Incoherent Interferer

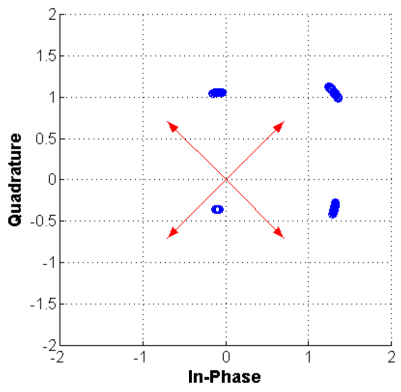
While the interferer is present, the underlying message signal can be assumed to be stationary. This means that to properly demodulate the signal it is best to keep the frequency offset at the same state as it was prior to the interferer being introduced. Thus either decreasing the VCO gain or decreasing the bandwidth of the error filter will result in a more stable constellation.

4.2.3 Coherent Interference

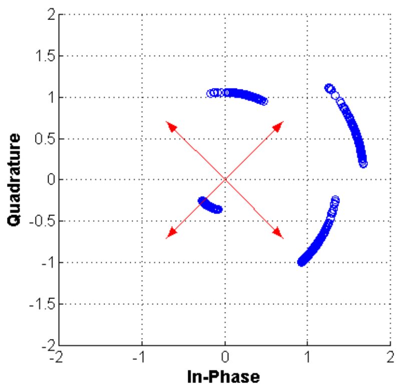
The second type of interferer that will be simulated is a coherent interferer. Mathematically, a coherent interferer is simpler than an incoherent interferer since the result is a constant vector offset. However, if the phase lock loop reacts to a coherent interferer in a similar manner to the incoherent interferer, then the simple mathematical expression is lost.



(a) K$_{VCO}$ = 10

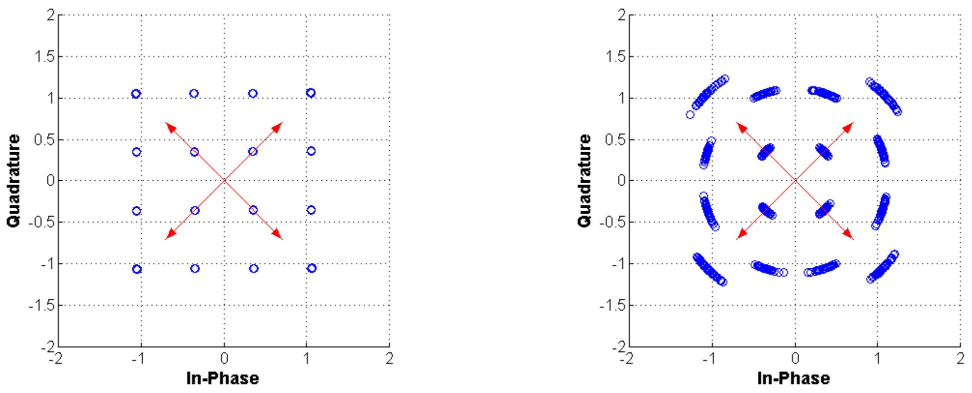


(b) K$_{VCO}$ = 100



(c) K$_{VCO}$ = 1000

Figure 19: Constellation Diagrams of Costas Loop Behavior with Coherent Interference



(a) $K_{VCO} = 10$ (b) $K_{VCO} = 1000$

Figure 20: Constellation Diagrams of Costas Loop Behavior with Modulated Coherent Interference

As expected, the lower values of K_{VCO} produced closer results to the mathematical predictions. It is easier and more straight forward to correct for a constant vector offset than it is to correct for a constantly changing error error vector. Even if the interfering tone was modulated, the resulting constellation can still be decoded. Assuming that the interfering signal has a QPSK modulation scheme and an amplitude of 0.5 the amplitude of the carrier signal, the constellation diagrams are shown in Figure 20

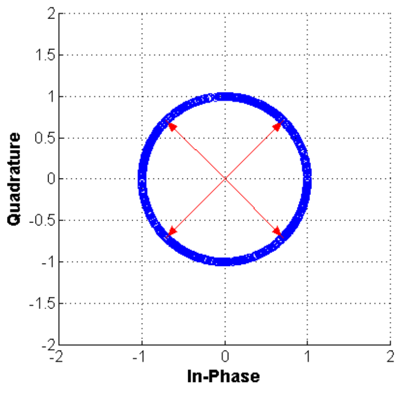
With a modulated interferer, the constellation diagrams are further altered from the ideal 4 constellation points outlined at the end of each red arrow. The result seen in Figure 20a is expected since the modulation of the interfering signal directly influences the error vector. The interferer's amplitude and phase are mapped onto each of the carriers constellation points resulting in multiple copies of the interferer's constellation centered at each of the message's constellation points. At higher VCO gains, the constellation is more difficult to identify.

When attempting to cancel the interferers, it is important to have a stable constellation. J. Lee, D. Toumpakaris, and W. Yu describe how joint interference detection can be used to successfully to improve BER. [5] In their paper, they develop stochastic models based on both the carrier signal and interfering signal characteristics. Information about the modulation scheme, amplitude, and phase offset are needed to ensure that joint interference demodulation works. This is because this joint demodulation technique uses estimators based on all the possible constellation point locations. Since the magnitude and phase modulation of the interferer directly affects the received constellation, estimators are generated based on all the possible symbol combinations between the desired message and the interferer. With a standard phase locked loop, the information about the interferer signal could not be obtained easily. This is because standard PLLs use allow the error signal to have a large impact on the output frequency of the VCO. To obtain stable constellations in the presence of an interferer, the output error signal from the phase error detector needs remain as constant as possible. This can be done by either suppressing the VCO gain or decreasing the bandwidth of the error signal filter.

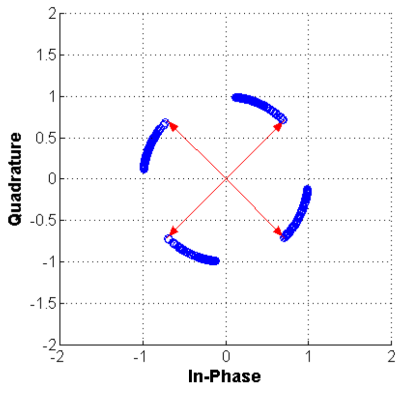
Based on all the results presented, the best solution to reduce the impact of interferers is to reduce the influence of the error signal in the PLL feedback loop. However, there are downsides to designing a PLL with low VCO gains or narrow loop bandwidths. One of the downsides was discussed earlier when demonstrating the system behavior to phase offsets. The other downside will be apparent when the system encounters a Doppler shift.

4.2.4 Carrier Frequency Drift

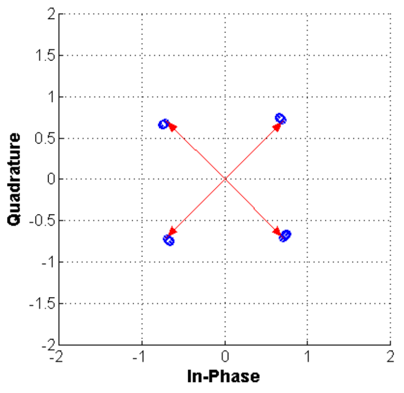
Carrier frequency drift is typically a result of either a moving transmitter or a moving receiver. The motion of these devices causes the apparent frequency of the radio wave to change depending on the velocity of the transceivers relative to each other. Transceivers that move towards each other will cause the carrier frequency to shift up or increase in frequency. Movement away from each other causes the carrier frequency to shift down or decrease in frequency. The magnitude of the frequency shift depends on the relative velocity of the two transceivers. The total frequency shift (f_D)can be calculated as



(a) $K_{VCO} = 10$



(b) $K_{VCO} = 100$



(c) $K_{VCO} = 1000$

Figure 21: Constellation Diagrams of Costas Loop Behavior with +75 Hz Frequency Drift

follows: [4]

$$f_D = \frac{v}{\lambda}$$

Where v is the relative speed between the source and the receiver and λ is the wavelength of the carrier frequency. To exaggerate the effects of frequency drift, a +75 Hz offset will be used with a carrier frequency of 3 MHz. The constellation diagrams are shown in Figure 21.

The main objective of a PLL is to be able to lock onto the phase and frequency of the received signal. In this particular scenario, the larger K_{VCO} performed better to the drifting carrier frequency. Since there is a limit to the maximum output voltage, lowering the VCO frequency gain decreases the range of frequencies that the VCO can output. If the VCO cannot output the correct reference frequency for the PLL to lock onto, then the system will remain unlocked. In the simulation, the error signal is bounded between ± 1. Thus, for a K_{VCO} of 10, the VCO could not represent a +75 Hz frequency offset as the frequency control range is limited to ± 10 Hz This phenomena is apparent in the error signal plots in Figure 22

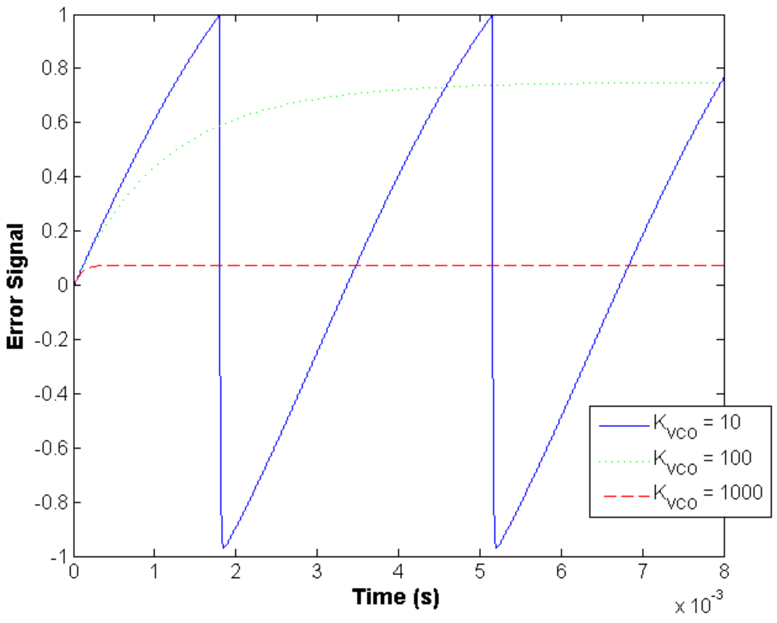


Figure 22: Error Signal vs Time with +75 Hz Frequency Offset

As mentioned, the lowest VCO gain cannot maintain lock, so the error signal continues to ramp as the phase error is never settled. The highest value of K_{VCO} settled very quickly as small changes in the error signal resulted in large frequency corrections. K_{VCO} of 100 was able to achieve steady state, but slower than K_{VCO} of 1000. The frequency gain of the VCO and the output voltage range of the phase error detector determine the magnitude of frequency drift the PLL can handle. Due to the shorter wavelengths, higher frequencies will result in larger frequency drifts at a given velocity. Higher frequency drifts correspond to faster rotating constellations which makes demodulation difficult.

In addition to frequency drift, PLLs also need to be able to account for the channelization of communication standards. Many communication standards are channelized, meaning that data signal bandwidths can be centered at different frequencies. The most prominent solution to this is to structure the receiver as a two stage down converter. The first stage converts the signal to a known IF frequency. The IF frequency is then processed by a demodulator such as the Costas loop. In this scenario the demodulator only needs to operate at a single known frequency and thus the IF VCO does not need to have a large frequency gain.

On the other hand, if the system is structured as a single stage downconversion, then the VCO needs to be able to tune to all of the channel frequencies. A larger frequency gain VCO can accommodate all of the different channel frequencies.

With every design parameter, there are always trade-offs. For each type of interference there are ideal conditions that best address the situation. The next section discusses optimal parameters that can be used to mitigate the effects of interfering signals and allow for digital joint demodulation of the received signals.

4.3 Recommended Architecture

The previous section demonstrated how a Costas loop performs given different interfering signals and different loop parameters. The main parameter that was modified was the VCO frequency gain which controls the frequency range of the VCO. Other parameters that can be tuned are the loop filter bandwidth and the error signal magnitude. All of these parameters can be tuned to make the PLL more or less responsive to carrier signal errors. Table 2 describes how to adjust these parameters to get the same effect as adjusting the VCO frequency gain.

To Increase PLL Sensitivity:	To Decrease PLL Sensitivity:
Increase VCO gain	Decrease VCO Gain
Increase Loop Bandwidth	Decrease Loop Bandwidth
Increase Error Signal Magnitude	Decrease Error Signal Magnitude

Table 2: Parameter Value Effects on System Performance

The optimal system takes advantage of both system characteristics. A high sensitivity PLL is needed when the initial phase and frequency are unknown. Once those two characteristics are established, the error signal remains constant. At this time, the sensitivity of the PLL can be reduced to prepare for any interfering signals that may appear. When the PLL sensitivity is reduced, the system is only vulnerable to frequency and phase drifts. In both cases, the convergence time is long as shown in Figure 14. In addition, the loop may not converge if the frequency offset is large enough as shown in Figure 22. One solution is to use external telemetry to determine if the the receiver is moving. This movement information can be used to increase the PLL sensitivity to account for the inevitable frequency and phase offset. This method will only account for receiver movement and will not work for transmitter motion. If the transmitter is moving, the received constellation will slowly drift following a circle centered at the origin. When the receiver identifies that the constellation is rotating, the sensitivity can be increased. After the errors due to motion are resolved, the sensitivity can be reduced again to combat interference.

Of the presented types of interference, a phase offset and Doppler shift benefit from higher values of K_{VCO}. It is also important to note that both types of interferences can be distinguished by rotational translations about the origin of the desired constellation. This means that under these conditions, the magnitude of the rotating constellation remains constant while the phase drifts.

The other interference types, coherent and incoherent interference, benefit from lower values of K_{VCO}. at low values of K_{VCO}, the constellations of these types of interferers cause constellation offsets proportional to the phase and amplitude of the interfering signal. Joint detection can be performed on these signals if information on the interferer can be obtained while the PLL is set to a low value of K_{VCO}.

A simulation was constructed to illustrate the benefits of an adaptive PLL taking into account the different kinds of interferers. Low values of K_{VCO} will be used when other signal interferers are present and larger values of K_{VCO} will be used when phase offsets or Doppler shifts are occurring. The difference between the two types of interference will be whether or not the constellation is rotating about the origin.

4.4 Adaptive Simulation Algorithm and Theory

As mentioned in the previous section, the critical part of the adaptive algorithm are the nature of the different interferers. The assumption in this algorithm is that there are no other signals present execpt the desired signal during the initialization of the loop. On start up, the phase lock loop defaults to a high value of K_{VCO} since the assumption is that the desired signal has a phase or frequency offset. The optimal setting to compensate for this is a high K_{VCO} value.

Error vector magnitude(EVM) is calculated by taking the difference vector between the received constellation point and the desired constellation point. Since the receiver cannot know what the intended received symbol is, the calculation is performed by taking the difference between the received vector and each of the desired constellation points. The minimum magnitude error vector is then recorded as the EVM for that symbol. Once the overall phase or frequency offset is accounted for by the PLL, the EVM will decrease closer to 0. Once several symbol EVMs are below a threshold, the PLL will then begin to decrease the value of K_{VCO} until it reaches the minimum possible value or until the EVM surpasses the threshold again. If the PLL does not decrease to the minimum value of K_{VCO}, this means that there is a Doppler frequency shift and any lower value for K_{VCO}

will prevent the loop from locking onto the drift.

Once the PLL is at the lowest allowed value for K_{VCO}, any foreign signals that interfere with the desire signal can be studied as mentioned in Section 4.2.3. The low value of K_{VCO} is ideal for this application. During this time, information about the interferer can be obtained and dealt with accordingly.

If additional phase or frequency offsets are introduced, the PLL will increase K_{VCO} to accomodate. When this occurs at low values of K_{VCO}, the constellation slowly rotates as shown in Section 4.2. Therefore, the criteria for increasing K_{VCO} is if the magnitude of the constellation point remains the same and the phase of the point is drifting over time.

With these two criteria in place to increase and decrease the value of K_{VCO}, the PLL can now adapt to changing conditions in the environment. There are several parameters that can be modified to adjust for the signal conditions. The thresholds for EVM and phase offsets can be adjusted to make the algorithm more or less reactive to the environment. If it is a noisy environment, it is best to keep the algorithm to be less reactive to the randomness of noise. In noisy environments, EVM calculations can also be averaged together to ensure accurate estimates are used. The number of samples averaged can will also influence how reactive the algorithm is to changes.

4.5 Adaptive Simulation Results

A composite signal was generated containing different types of interferers that occur at different time intervals. The results of the adaptive algorithm will be compared to a traditional PLL with K_{VCO} set to 10 and 1000.

The simulation and signal parameters are the same as listed in Table 1. 800 symbols will be transmitted with an initial phase offset of $\frac{\pi}{6}$. During symbols 200-300, a coherent, unmodulated interferer will be added to the message signal. During symbols 400-600, a

Doppler shift of -100 Hz will be added to the carrier frequency. To better understand the results graphs, it is important to note that 100 symbols will be transmitted every 1 ms.

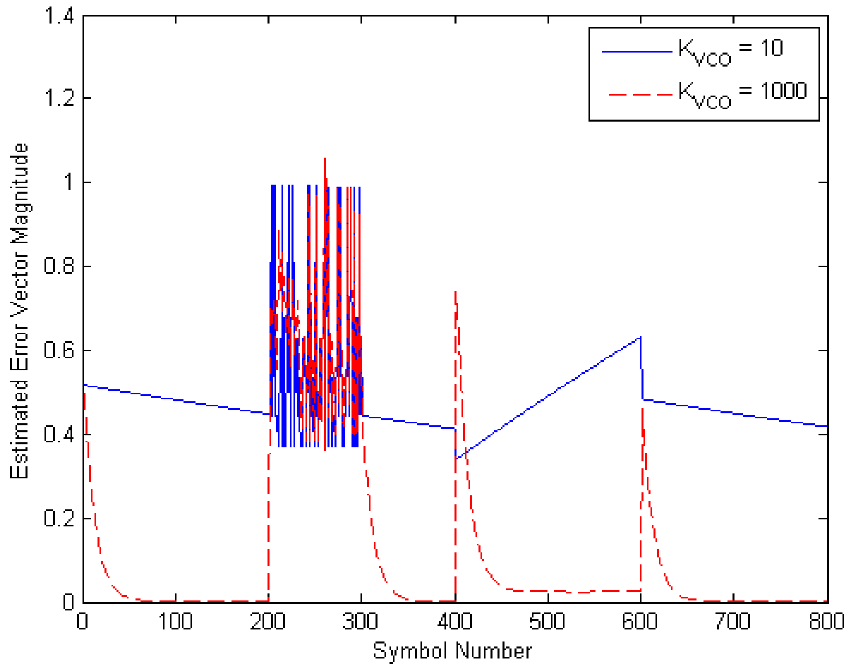


Figure 23: Estimated EVM for Static Phase Lock Loops

Figure 23 shows the estimated EVM over time where K_{VCO} is set to a constant 10 or 1000. The higher value K_{VCO} performs well regardless of the poor performance when the interfering signal is present between symbols 200-300. As expected, the phase lock loop does not behave well during this period of time.

The lower value of K_{VCO} does not overcome the first obstacle, which is the initial phase offset, before the interferer is active. Although low values of K_{VCO} are better for joint demodulation of an interfering signal, if the loop does not lock to the original carrier, it is difficult to estimate any interfering properties. Throughout the entire signal, the phase locked loop with a K_{VCO} of 10 has trouble adapting to the changing signal conditions.

Using the previously adaptive rules, the results of the adaptive PLL can be compared to the static control experiments. Figure 24 adds the EVM performance of the adaptive system to the plot.

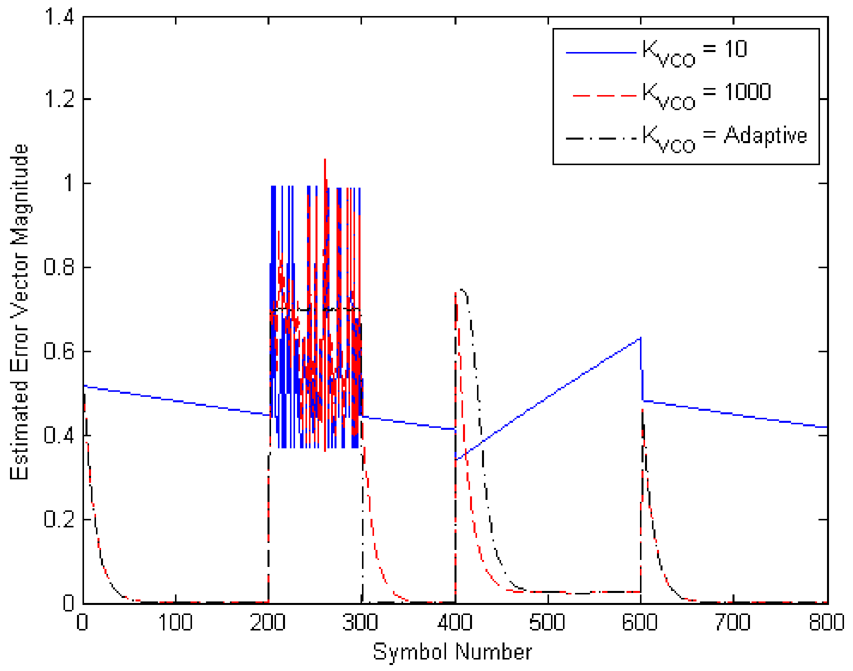


Figure 24: Estimated EVM for Adaptive Phase Lock Loop

The adaptive PLL performs significantly better than the static PLL. The initial phase offset is overcome with a large value of K_{VCO} and is able to converge prior to the introduction of the interferer. While the interferer is present, the EVM stays a constant 0.7 which corresponds to the amplitude of the interferer. When the Doppler shift is applied after the interferer, the adaptive algorithm is able to adapt. Although it adapted slower than the high static K_{VCO} experiment, it was still able to overcome a Doppler shift after performing well during interference which is not the case with a static K_{VCO} value of 10. After the Doppler shift ends, both the static K_{VCO} value of 1000 and the adaptive PLL reconverge at similar rates to the desired signal. The plot of K_{VCO} values over time can be seen in Figure 25.

After the PLL overcomes the initial phase offset, K_{VCO} slowly decays before the interferer begins. From 2 ms to 3 ms, K_{VCO} does not change even with an interferer present as desired. When the Doppler shift begins at 4 ms, the PLL recognizes this and begins to

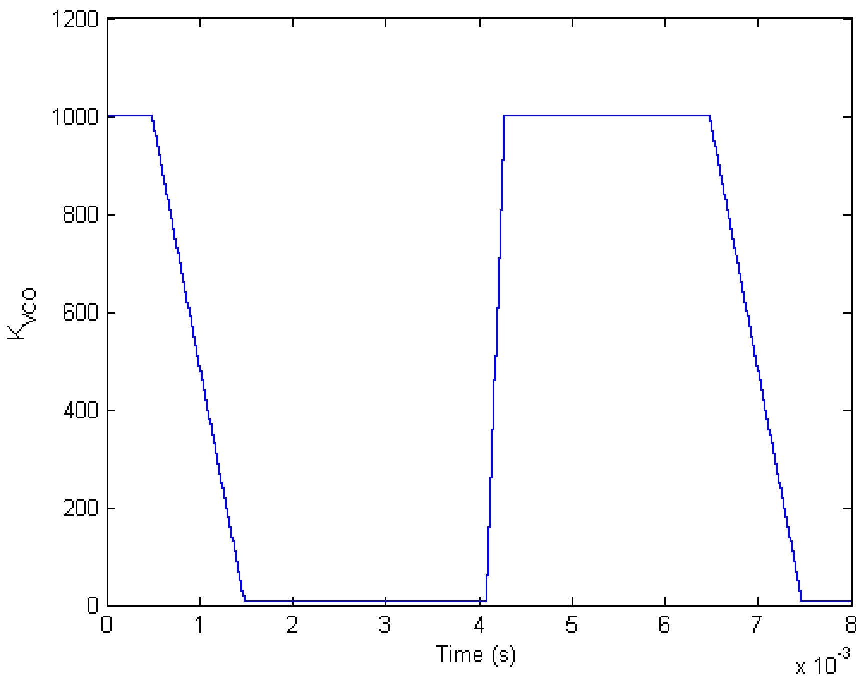


Figure 25: K$_{VCO}$ Value Over Time

increase K$_{VCO}$ until the Doppler shift is overcome. Once the Doppler shift ends at 6 ms, the PLL returns to a low K$_{VCO}$ value to mitigate any future interference.

The behavior of the adaptive PLL can be adjusted depending on signal conditions. The thresholds for when the algorithm decides to increase or decrease K$_{VCO}$ can be modified to make the algorithm more or less reactive. Some other properties that can be modified is how quickly or slowly the algorithm steps K$_{VCO}$ and the maximum and minimum values for K$_{VCO}$.

Noise will affect how EVM is estimated and will affect how the algorithm decides if there is a phase or frequency offset. One way to reduce the impact of noise is to average multiple samples together to get a better estimation of the received symbol. The thresholds for the adaptation algorithm can also be modified to have smaller thresholds so when noise is present there is more room for symbol estimation error.

5 Conclusion and Future Work

Phase locked loops have a significant role in demodulating signals where carrier recovery is required. Carrier recovery techniques have been developed for single carrier singles. When interferers are introduced to the system, traditional Costas loops have difficulty locking onto the carrier. As a result, the output constellations are skewed and many times uninterpretable. A method to suppress the impact of high power interference is to reduce the sensitivity of the Costas loop. Doing so allows the receiver to observe the status of the channel and react appropriately. Joint demodulation of the message and interferer is possible since information on the interfering modulation scheme can be deduced from the constellation diagrams. Enabling joint detection allows transceivers to continue communicating even if there are interfering signals present.

Further improvements to the system include developing an integrated front end that utilizes and adaptive phase locked loop. The implementation should involve digital decision based feedback to the oscillator so that the receiver can react to a variety of different environmental conditions. Adapting the RF front end of an OFDM receiver can also provide some benefits. OFDM is vulnerable to frequency drift and having an adaptive phase locked loop to track frequency drift while maintaining accurate sync to the subcarriers. Allowing RF hardware to adapt in conjunction with digital processing will allow for better performance in presence of interference.

References

[1] J.-P. Linnartz. (1993) Time division multiple access (tdma). [Online]. Available: http://www.wirelesscommunication.nl/reference/chaptr04/multi/tdma.htm

[2] ——. (1993) Direct sequence cdma. [Online]. Available: http://www.wirelesscommunication.nl/reference/chaptr05/cdma/dscdma.htm

[3] J. H. Reed, *An Introduction to Ultra Wideband Communication systems.* New Jersey: Prentice Hall PTR, 2005.

[4] A. Goldsmith, *Wireless Communications.* New York: Cambridge Univ. Press, 2005.

[5] J. Lee, D. Toumpakaris, and W. Yu, "Interference mitigation via joint detection," vol. 29, no. 6, pp. 1172–1184, 2011.

[6] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 2012.

[7] A. Networks. (2014) 802.11ac in-depth. [Online]. Available: http://www.arubanetworks.com/pdf/technology/whitepapers/WP_80211acInDepth.pdf

[8] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, IEEE Std. 802.11ac, 2013.

[9] *Physical layer on the radio path; General description*, 3GPP Std. TS 45.001, 2014.

[10] V. P. Ipatov, *Spread Spectrum and CDMA.* New Jersey: John Wiley and Sons, 2005.

[11] *Master Table of Contents and Compliance Requirements*, Bluetooth Std. 4.2, 2014.

[12] Hewlett-Packard. (2002) Wi-fi™ and bluetooth™ - interference issues. [Online]. Available: http://www.hp.com/rnd/library/pdf/WiFi_Bluetooth_coexistance.pdf

[13] I. Howitt, "Bluetooth performance in the presence of 802.11b wlan."

[14] P. Patras, H. Qi, and D. Malone. (2012) Exploiting the capture effect to improve wlan throughput. [Online]. Available: http://homepages.inf.ed.ac.uk/ppatras/pub/wowmom12.pdf

[15] C. . C. F. Jeng, S.S. ; Tsung, "Wlan smart antenna with bluetooth interference reduction."

[16] S.-H. L. . Y.-H. Lee, "Adaptive frequency hopping for bluetooth robust to wlan interference."

[17] C.-K. C. M. C. . S. C. Liew, "Capacity of large-scale csma wireless networks."

[18] F. M. Gardner, *Phaselock Techniques.* New Jersey: John Wiley and Sons, 2005.

Appendix A MATLAB Simulation Static Costas Loop Code

```
%Kevin Nguyen
%Static Costas Loop Simulation
% ------------------------------------------------------------------------
clear
close all
clc
% --------------------------Signal Generation------------------------------

%PSK Modulation Order
mod_order = 4;
%Number of symbols in Transmission
nsymb = 800;
%Generate Symbols
load('symb'); %Predefined set of random symbols
%symb =  randi(mod_order,[1 nsymb])-1; %Generate a random string of symbol
%symb = zeros(1,nsymb); %Generate all zeros

%Modulate symbols to get complex coefficients
mod_samp = pskmod(symb,mod_order,pi/4);

%Initialize interferer modulation scheme QPSK or all Zeros
int_symb =  randi(mod_order,[1 nsymb])-1;% zeros(1,nsymb); %
int_mod_samp = pskmod(int_symb,mod_order,pi/4);
```

```
%Number of samples per symbol
symb_samp = 100;
%Total samples = samples per symbol * number of symbols
nsamp = nsymb*symb_samp;
%Generate sample enumeration
samp = 0:nsamp-1;
%Set frequency parameters
fs = 1e7;                     % Sampling Frequency
fc = 3e6;                    % Carrier Frequency
fi = fc+50;                     % Interferer Frequency
fd = 0;                      %Doppler Shift

%compute Real and imaginary part of symbols for debugging
realpart = real(reshape(repmat(mod_samp,symb_samp,1),1,[]));
imagpart = imag(reshape(repmat(mod_samp,symb_samp,1),1,[]));
%Determine the phase of each symbol
sig_phase = angle(reshape(repmat(mod_samp,symb_samp,1),1,[]));
%Determine the phase of each interfering symbol
int_sig_phase = angle(reshape(repmat(int_mod_samp,symb_samp,1),1,[]));
%Generate noise
noise = randn([1 nsamp])/2;
%Sample to Start interference
start = 1;
%Sample to stop interference
stop = 80000-1;
%Generate interfering signal
```

```
interferer = [zeros(1,start),0.7*cos(2*pi*fi*[start:stop]/fs), ...
    zeros(1,nsamp-stop-1)];
%interferer = 0.6*cos(2*pi*fi*[start-1:stop-1]/fs+int_sig_phase);
%Modulate carrier
c = cos(2*pi*(fc+fd)*samp/fs + sig_phase+pi/6);
%Apply all interfering effects
st = c+0*noise+0*interferer;

% --------------------------Signal Demodulation------------------------
%Length of received signal
N = length(st);
%discrete time vector
t = 0:1:N-1;
%Correction frequency
f_corr = zeros(1,N);
%I branch signal
s1 = zeros(1,N);
%Q branch signal
s2 = zeros(1,N);
%output from VCO
lo_1 = zeros(1,N);
%output from VCO delayed by 90 deg
lo_2 = zeros(1,N);
%Down converted signal after low pass filter
y1 = zeros(1,N);
y2 = zeros(1,N);
%Error signal after filtering
```

```
error = zeros(1,N);
%Error signal before filtering
error_temp = zeros(1,N);
%received demodulated symbols
rec_symb = zeros(1,nsymb);
%current phase of VCO used to maintain continuity
theta_temp = zeros(1,N);
%VCO frequency gain
kvco = 1000;



symb_count = 1;
for ii = 1:N
    %Phase error estimation
    if ii>1
       error_temp(ii) = sign(y2(ii-1))*y1(ii-1)-sign(y1(ii-1))*y2(ii-1);
       %filter raw error signal
       error(ii) = 0.99*error(ii-1)+ 0.01*error_temp(ii);
       %correction frequency calculated based on filtered error
       f_corr(ii) = kvco*error(ii);
       %compute new phase of VCO depending on frequency correction
       theta_temp(ii) = theta_temp(ii-1)+2*pi*(fc+f_corr(ii))/fs;
    end
    %Generate new LO signals
    lo_1(ii) = cos(theta_temp(ii));
    lo_2(ii) = sin(theta_temp(ii));
    %Determine new output signals from mixers
    s1(ii) = 2*st(ii) * lo_1(ii);
    s2(ii) = 2*st(ii) * lo_2(ii);

% -----------------------INTEGRATOR (LPF)----------------------------------
    %Length of filter or integrator
    integrate_len = symb_samp/4;
    %Apply mean filter
    if ii<=integrate_len


        y1(ii) = mean(s1(1:ii));
        y2(ii) = mean(s2(1:ii));
    else
        y1(ii) = mean(s1(ii-integrate_len+1:ii));
        y2(ii) = mean(s2(ii-integrate_len+1:ii));
    end

    %At the center of every symbol, record the received symbol
    if mod(ii,symb_samp)==symb_samp*1/2
        rec_symb(symb_count) = y1(ii)-j*y2(ii);
        symb_count = symb_count+1;
    end



end
%-------------------------Figure Generation--------------------------------
```

```
%Compare real part of symbols for debugging
figure;
plot(t,y1,t,realpart);title('Output signal');
xlabel('Time');ylabel('Amplitude');

%Compare Imaginary part of symbols
figure;
plot(t,y2,t,imagpart);title('Output signal');
xlabel('Time');ylabel('Amplitude');

%Demodulate received symbols and calculate BER
out_symb = pskdemod(rec_symb,mod_order,pi/4);
BER = sum(out_symb==symb)/nsymb;

%EVM calculation
EVM = abs(rec_symb-mod_samp)./abs(mod_samp);

amp_error = abs(rec_symb) - abs(mod_samp);
phase_error = angle(rec_symb)-angle(mod_samp);

%Scatter plot received symbols to form constellation diagrams
figure
scatter(real(rec_symb),imag(rec_symb))
hold on
xlim([-2 2]);
ylim([-2 2]);
axis square
```

```
grid on
%Draw arrows to target constellation points
arrow3([0,0],[sqrt(2)/2,sqrt(2)/2],'r','',1.5);
arrow3([0,0],[-sqrt(2)/2,sqrt(2)/2],'r','',1.5);
arrow3([0,0],[sqrt(2)/2,-sqrt(2)/2],'r','',1.5);
arrow3([0,0],[-sqrt(2)/2,-sqrt(2)/2],'r','',1.5);
%arrow3([sqrt(2)/2,sqrt(2)/2],[sqrt(2)/2+cos(pi/6),sqrt(2)/2+sin(pi/6)],'j-
%-','',1.5);
xlabel('In-Phase','FontSize',14,'FontWeight','bold')
ylabel('Quadrature','FontSize',14,'FontWeight','bold')
set(gca,'FontSize',12)
```

Appendix B MATLAB Simulation Adaptive Costas Loop Code

```
%Kevin Nguyen
%Adaptive Costas Loop Simulation
% ------------------------------------------------------------------------
clear
close all
clc
% --------------------------Signal Generation------------------------------

%PSK Modulation Order
mod_order = 4;
%Number of symbols in Transmission
nsymb = 800;
%Generate Symbols
load('symb'); %Predefined set of random symbols
%symb =  randi(mod_order,[1 nsymb])-1; %Generate a random string of symbol
%symb = zeros(1,nsymb); %Generate all zeros

%Modulate symbols to get complex coefficients
mod_samp = pskmod(symb,mod_order,pi/4);

%Initialize interferer modulation scheme QPSK or all Zeros
int_symb =  randi(mod_order,[1 nsymb])-1;% zeros(1,nsymb); %
int_mod_samp = pskmod(int_symb,mod_order,pi/4);
```

```
%Number of samples per symbol

symb_samp = 100;

%Total samples = samples per symbol * number of symbols

nsamp = nsymb*symb_samp;

%Generate sample enumeration

samp = 0:nsamp-1;

%Set frequency parameters

fs = 1e7;                     % Sampling Frequency

fc = 3e6;                    % Carrier Frequency

fi = fc;                     % Interferer Frequency

fd = -30;                      %Doppler Shift

doppl_start = 40000;

doppl_stop = 60000;

f = zeros(1,nsamp);

f(1:doppl_start-1) = fc;

f(doppl_start:doppl_stop-1) = fc+fd;

f(doppl_stop:end) = fc;


%compute Real and imaginary part of symbols for debugging

realpart = real(reshape(repmat(mod_samp,symb_samp,1),1,[]));

imagpart = imag(reshape(repmat(mod_samp,symb_samp,1),1,[]));

%Determine the phase of each symbol

sig_phase = angle(reshape(repmat(mod_samp,symb_samp,1),1,[]));

%Determine the phase of each interfering symbol

int_sig_phase = angle(reshape(repmat(int_mod_samp,symb_samp,1),1,[]));

%Generate noise
```

```
noise = randn([1 nsamp])/5;

%Sample to Start interference

start = 20000;

%Sample to stop interference

stop = 30000;

%Generate interfering signal

interferer = [zeros(1,start),0.7*cos(2*pi*fi*[start:stop]/fs)...

    ,zeros(1,nsamp-stop-1)];

%interferer = 0.6*cos(2*pi*fi*[start-1:stop-1]/fs+int_sig_phase);

%Modulate carrier

c = cos(2*pi*f.*samp/fs + sig_phase+pi/6);

%Apply all interfering effects

st = c+0*noise+interferer;


% --------------------------Signal Demodulation------------------------

%Length of received signal

N = length(st);

%discrete time vector

t = 0:1:N-1;

%Correction frequency

f_corr = zeros(1,N);

%I branch signal

s1 = zeros(1,N);

%Q branch signal

s2 = zeros(1,N);

%output from VCO

lo_1 = zeros(1,N);
```

```
%output from VCO delayed by 90 deg
lo_2 = zeros(1,N);
%Down converted signal after low pass filter
y1 = zeros(1,N);
y2 = zeros(1,N);
%Error signal after filtering
error = zeros(1,N);
%Error signal before filtering
error_temp = zeros(1,N);
%received demodulated symbols
rec_symb = zeros(1,nsymb);
min_evm_mag = zeros(1,nsymb);
min_evm_pha = zeros(1,nsymb);
%current phase of VCO used to maintain continuity
theta_temp = zeros(1,N);
%VCO frequency gain
init_kvco = 1000;
kvco = repmat(init_kvco,1,N);
%Number of symbols to wait before processing
evm_length = 10;

symb_count = 1;
for ii = 1:N
    %Phase error estimation
    if ii>1
       error_temp(ii) = sign(y2(ii-1))*y1(ii-1)-sign(y1(ii-1))*y2(ii-1);
       %filter raw error signal
```

```
      error(ii) = 0.99*error(ii-1)+ 0.01*error_temp(ii);
      %correction frequency calculated based on filtered error
      f_corr(ii) = error(ii)*kvco(ii);
      %compute new phase of VCO depending on frequency correction
      theta_temp(ii) = theta_temp(ii-1)+2*pi*(fc+f_corr(ii))/fs;
   end
   %Generate new LO signals
   lo_1(ii) = cos(theta_temp(ii));
   lo_2(ii) = sin(theta_temp(ii));
   %Determine new output signals from mixers
   s1(ii) = 2*st(ii) * lo_1(ii);
   s2(ii) = 2*st(ii) * lo_2(ii);

% -----------------------INTEGRATOR (LPF)----------------------------------
   %Length of filter or integrator
   integrate_len = symb_samp/4;
   %Apply mean filter
   if ii<=integrate_len

       y1(ii) = mean(s1(1:ii));
       y2(ii) = mean(s2(1:ii));
   else
       y1(ii) = mean(s1(ii-integrate_len+1:ii));
       y2(ii) = mean(s2(ii-integrate_len+1:ii));
   end

   %At the center of every symbol, record the received symbol
```

```
if mod(ii,symb_samp)==symb_samp*1/2
    rec_symb(symb_count) = y1(ii)-j*y2(ii);
    %Estimate EVM Magnitude
    min_evm_mag(symb_count) = min(rec_symb(symb_count)- ...
        [exp(j*pi/4) exp(j*3*pi/4) exp(-j*3*pi/4) exp(-j*pi/4)]);
    %Estimate Phase Error
    phase_error_temp = wrapTo2Pi(angle(rec_symb(symb_count))- ...
        [pi/4 3*pi/4 5*pi/4 7*pi/4]);
    [phase_error_value,phase_error_loc] = min(abs(phase_error_temp));
    min_evm_pha(symb_count) = phase_error_temp(phase_error_loc);
    %Increment Total Number of Received Symbols
    symb_count = symb_count+1;

    %Adaptive kvco Algorithm
   if symb_count > evm_length
       %Extract Phase information of the previous 'evm_length' symbols
       phase_vec_wind = ...
           min_evm_pha(symb_count-evm_length:symb_count-1);
       %Extract symbol information of the previous 'evm_length' symbols
       symb_wind = rec_symb(symb_count-evm_length:symb_count-1);
       %If the magnitude of EVM is below a threshold, decrease kvco
       if max(abs(min_evm_mag(symb_count-evm_length:symb_count-1)))...
                < 0.02
            kvco(ii+1:end) = max(10,kvco(ii)-10);
       %If Magnitude of symbols is constant and phase is drifting,
       %Increase kvco
        elseif (max(abs(symb_wind))-min(abs(symb_wind)) < 0.01) & ...
```

```
                    (min(abs(phase_vec_wind(2:end)-...
                    phase_vec_wind(1:end-1))) > 0.001)
                kvco(ii+1:end) = min(1000, kvco(ii)+50);
            end
        end
    end
end
%------------------------Figure Generation--------------------------------

%Compare real part of symbols for debugging
figure;
plot(t,y1,t,realpart);title('Output signal');
xlabel('Time');ylabel('Amplitude');

%Compare Imaginary part of symbols
figure;
plot(t,y2,t,imagpart);title('Output signal');
xlabel('Time');ylabel('Amplitude');

%Demodulate received symbols and calculate BER
out_symb = pskdemod(rec_symb,mod_order,pi/4);
BER = sum(out_symb==symb)/nsymb;

%EVM calculation
EVM = abs(rec_symb-mod_samp)./abs(mod_samp);

amp_error = abs(rec_symb) - abs(mod_samp);
```

```
phase_error = angle(rec_symb)-angle(mod_samp);


%Scatter plot received symbols to form constellation diagrams
figure
scatter(real(rec_symb),imag(rec_symb))
hold on
xlim([-2 2]);
ylim([-2 2]);
axis square
grid on
%Draw arrows to target constellation points
arrow3([0,0],[sqrt(2)/2,sqrt(2)/2],'r','',1.5);
arrow3([0,0],[-sqrt(2)/2,sqrt(2)/2],'r','',1.5);
arrow3([0,0],[sqrt(2)/2,-sqrt(2)/2],'r','',1.5);
arrow3([0,0],[-sqrt(2)/2,-sqrt(2)/2],'r','',1.5);
%arrow3([sqrt(2)/2,sqrt(2)/2],[sqrt(2)/2+cos(pi/6),sqrt(2)/2+sin(pi/6)],'j-
%-','',1.5);
xlabel('In-Phase','FontSize',14,'FontWeight','bold')
ylabel('Quadrature','FontSize',14,'FontWeight','bold')
set(gca,'FontSize',12)
```