MODULATING LINEAR FREQUENCY MODULATED PULSES TO SEND COMMUNICATIONS DATA IN A BISTATIC SAR SCENARIO

by

Jarren T. Worthen

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

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Todd K. Moon, Ph.D. Major Professor	Chad Knight, Ph.D. Committee Member
Johnathan Phillips, Ph.D. Committee Member	David F. Feldon, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2026

Copyright © Jarren T. Worthen 2026

All Rights Reserved

ABSTRACT

Modulating Linear Frequency Modulated Pulses to Send Communications Data in a Bistatic SAR Scenario

by

Jarren T. Worthen, Master of Science

Utah State University, 2026

Major Professor: Todd K. Moon, Ph.D.
Department: Electrical and Computer Engineering

In a bistatic SAR situation there are many scenarios where transmitting information between the transmitter and receiver would be beneficial. Taking advantage of the already-present radar pulses as information carriers would be efficient as the communications signal would benefit from the high power signal of the radar pulse. If the modulation of the communication signal degrades the linear frequency modulated pulse, then the SAR image would be degraded. The proposed research analyzes different modulation techniques that can be used to send data. The goal is minimize the bit error rate to signal-to-noise ratio while maximizing data rate, without degrading the quality of the SAR image.

(72 pages)

PUBLIC ABSTRACT

Modulating Linear Frequency Modulated Pulses to Send Communications Data in a Bistatic SAR Scenario

Jarren T. Worthen

Synthetic aperture radar (SAR) is the technology that enables the creation of images using radar waves, allowing images to be formed regardless of weather. In bistatic SAR a radar platform uses a radar pulse from a separate platform to form an image. It is important that these radar systems are able to communicate with each other. Rather than wasting energy and space flying with a separate communication system, the radar systems could use the already existing SAR system to send data between each other while still forming SAR images. The work of this thesis is to show how very simple changes to the industry standard radar pulse, would allow this communication to happen.

Go Aggies!

ACKNOWLEDGMENTS

I would like to thank Dr. Moon for being a mentor to me throughout my undergraduate and graduate studies. His passion for his subject matter is what helped me to fall in love with signal processing and its related fields. This has sent me on my current career trajectory and I am incredibly grateful for that.

I also want to thank Chad Knight for support of my career goals. He has given me both technical and life advice that has been of immense value to me. I am grateful to be able to work under his tutelage.

Finally, I am the most grateful of all to my family. My wife and eternal partner through life has supported me in every way that I have needed and it is a joy being able to climb the mountains of life with her. My three boys bring me so much joy and make everything doubly worth it.

Jarren Worthen

CONTENTS

LIST OF FIGURES

Figure Page

ACRONYMS

BiSAR	Bistatic SAR
BPSK	Binary Phase Shift Keying
FM	Frequency Modulation
IRW	Impulse Response Width
JRC	Joint Radar Communication
LFM	Linear Frequency Modulated
MATLAB	Matrix Laboratory
PRF	Pulse Repetition Frequency
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
RADAR	Radio Detection and Ranging
SAR	Synthetic Aperture Radar
SNR	Signal to Noise Ratio

CHAPTER 1

Introduction

Synthetic aperture radar (SAR) is a powerful radar imaging tool that is used in a large variety of applications. It has the advantage of typical radar, in that it is mostly unaffected by clouds or precipitation and it provides its own illumination [1]. In the general case a SAR image is formed by transmitting a series of phase-encoded pulses along a path. The received reflected pulses are used to form an image using signal processing.

Bistatic SAR (BiSAR) is a SAR system where the transmitter and receiver are spatially separated [1]. It can be useful for the transmitter to send communication data to the receiver while still receiving radar pulses. The most efficient solution to transmitting this communication data would be to utilize the high energy linear frequency modulated (LFM) pulses of the SAR system rather than sending a separate communication signal. This can be done by modulating the pulses with respect to the communication data.

The challenge of this solution is two fold. First, the modulation must not compromise the SAR systems ability to form a clear image. SAR uses a series of LFM pulses with certain mathematical properties to form its image. The proposed system may not alter the pulses in a way that significantly distorts the image. Second, the modulation of the pulses must be detectable by the receiver. The receiver super-imposes the returns from all ground targets in the scene. This distorts the modulation, making it more difficult to detect.

This paper reviews the literature relevant to this problem. It then analyzes three techniques that can be used to modulate the LFM chirp: up-down modulation, phase modulation, and time delay modulation. For each modulation technique we ensure that the modulation itself does not degrade the SAR image produced. We then calculate the theoretical probability of error when using the modulation technique as a communications waveform. Lastly each technique is simulated to prove the validity of the theoretical probability of error.

CHAPTER 2

Review of Literature

2.1 Joint Radar Communication Classifications

A single system that serves the dual purpose of processing radar and communications is called a joint radar communication (JRC) system. There are three classes of JRC systems: co-existence, cooperation and co-design [2]. Co-existence JRC systems have the radar and communications system see the other as interference and are simply using various methods to share the same antenna. In a cooperation design, the two system exchange information with each other to mitigate the interference relative to one another. In both cases the radar and communication waves match the industry standard for their use in a non-JRC system [3].

Co-designed systems attempt to redesign the radar and communications system from the ground up, keeping the functionality of both systems in mind [2]. An example of this is designing communication signals that are orthogonal to the radar LFM pulses so that there is little to no interference [4]. The proposed research is a co-designed system. The goal is to add no interference to the regular operation of the SAR system. As such, rather than designing communications and radar signals that are approximately orthogonal, we are modulating the radar pulses to encode the communications data.

2.2 Signal to Noise Ratio of Communication Signals Vs. SAR

One consideration that needs to be taken into account is how we can compare the quality of a SAR image to the quality of a communication signal. The simplest metric to measure quality would be signal to noise ratio (SNR). SNR is defined as the ratio of signal power to noise power [5]. In a communications setting where the signal is periodic, we define its power as such [6]

$$P_{comms} = \frac{1}{T_0}\int_{t_0}^{t_0+T_0} |x(t)|^2 dt, \tag{2.1}$$

where P is the power of the signal and T_0 is the period of the signal. The power of SAR used to calculate its SNR is [7]

$$P_{SAR} = P_T F_a T_r, \tag{2.2}$$

where P_T is the peak transmitted power, F_a is the radar pulse repetition frequency (PRF), and T_r is the duration of the transmitted pulses.

The period of a communications signal T_0 and the PRF F_a are both describing a similar characteristic. However, in a SAR scenario the PFR is affected by the range of the radar's target. To best model how effective a modulation technique is, we analyze the SNR per pulse rather than over time. This means that T_r and P_T will be used to calculate the SNR and F_a will be ignored.

2.3 LFM Pulse

LFM pulses are traditionally used as the transmitted radar signal that is then received and processed to form a SAR image [7]. As the name implies, these are signals whose frequency is modulated in a linear fashion. The most commonly used pulse has the complex form

$$s_{pul}(t) = \mathrm{rect}(\frac{t}{T_r}) \exp(j\pi K_r t^2), \tag{2.3}$$

where t is time, T_r is the pulse duration and K_r is the FM rate in Hertz per second. The rect function is defined as

$$\mathrm{rect}(x) = \begin{cases} 1 & \text{if } |x| \leq 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

In order to transmit this pulse we make the pulse real and add a frequency carrier of f_0. This realizable pulse is then

$$s_{real}(t) = \mathrm{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2). \tag{2.5}$$

Besides SNR, another important measurement to SAR image quality is resolution, which is correlated with impulse response width (IRW) [7]. A smaller IRW means two radar targets can be closer together and still be seen as two individual targets. When using an LFM pulse the IRW is approximately

$$\rho = \frac{1}{|K_r|T_r}, \tag{2.6}$$

or the reciprocal of the bandwidth. This means that a higher bandwidth in the LFM pulse leads to better resolution. In order to ensure that the communication modulation does not hurt the SAR's resolution, the modulation techniques must not alter the bandwidth of the pulses. We cannot change the magnitude of K_r without altering T_r to compensate. This however effects the power of the pulse which in turn changes the SNR. Therefore our modulation technique does not alter either $|K_r|$ or T_r.

2.4 Pulse Return from a Target

When an LFM pulse bounces off of a reflector present in the scene its return is approximately [7]

$$s_{return}(t) = \mathrm{rect}(\frac{t - R(\eta)}{T_r})\cos(2\pi f_0(t - R(\eta)) + \pi K_r(t - R(\eta))^2), \tag{2.7}$$

where $R(\eta)$ is the time it takes for a pulse to travel from the transmitter, to the target, and then to the receiver. We have removed some other terms like the properties of the reflector and the position of the reflector relative to the beam pattern. They are difficult to generalize and can be considered noise in regards to the communications system.

The real pulse is received and passed through a quadrature demodulation process. This removes the carrier $2\pi f_0 t$ from the phase of the signal. This leaves us with the phase of a complex exponential of [7]

$$-\frac{4\pi f_0 R(\eta)}{c} + \pi K_r \left(t - \frac{2R(\eta)}{c}\right)^2. \tag{2.8}$$

This complex exponential is then matched filtered to compress the pulse for further processing. It is after pulse compression that the waveforms are analyzed to determine what symbol was transmitted.

CHAPTER 3

Modulating the Sign of the FM Rate

One of the simpler and more intuitive ways to modulate the LFM pulse is by changing the sign of the FM rate. This chapter describes how this modulation is done, what limitations it has, and how effective it is at transmitting communications data. The modulation technique is also simulated in MATLAB to demonstrate its effectiveness.

3.1 Modulation Definition

This technique simply modulates the sign of K_r from the basic LFM chirp (2.3). When K_r is positive it is referred to as an up-chirp. Likewise, when K_r is negative it is referred to as a down-chirp. This does not change the bandwidth of the pulse, as only the magnitude of K_r affects the bandwidth.

As this only allows us one degree of freedom, each pulse only transmits a single bit of information. The modulated pulse is now

$$s_{modpul}(t) = \text{rect}(\frac{t}{T_r}) \exp(j\pi(-1)^b K_r t^2), \tag{3.1}$$

where b is the value of the bit being transferred.

3.2 SAR Image Resolution

Both up-chirps and down-chirps are valid and commonly used SAR pulses. In fact there are SAR systems that alternate between up-chirps and down-chirps in order to help with ambiguities found in certain applications [8]. Therefore the only obstacle to SAR image resolution is determining if an up-chirp or a down-chirp was transmitted. The simplest method to determine which pulse was transmitted is to have a matched filter for both pulses and see which one has the highest energy return.

The up LFM chirp is defined as (3.2) and its matched filter is (3.3).

$$s_{up}(t) = rect(\frac{t}{T_r})\exp(j\pi K_r t^2) \tag{3.2}$$

$$h_{up}(t) = rect(\frac{t}{T_r})\exp(-j\pi K_r t^2) \tag{3.3}$$

After some calculus [7], the output of the matched filter after processing the up chirp is

$$s_{out}(t) = s_{up}(t) * h_{up}(t) = (T_r - |t|)\,\mathrm{rect}(\frac{t}{2T})\,\mathrm{sinc}(K_r t(T_r - |t|)). \tag{3.4}$$

This is the output for both up-chirps and down-chirps. We then compare this to the output of an up-chirp going through the down-chirp's matched filter. The output is the convolution (3.5), which after much calculus gives us the closed form (3.6).

$$s_{out}(t) = \int_{-\inf}^{\inf} \mathrm{rect}(\frac{u}{T_r})\,\mathrm{rect}(\frac{t-u}{T_r})\exp(j\pi K_r u^2)\exp(j\pi K_r (t-u)^2)du \tag{3.5}$$

$$\frac{\exp(j7\pi K_r t^2/8)}{j\sqrt{j}\sqrt{8K_r}}(\mathrm{rect}\left(\frac{t}{2T_r}\right)(\mathrm{erf}\left(\frac{j\sqrt{j\pi K_r}(t+2T_r)}{\sqrt{8}}\right) - \mathrm{erf}\left(\frac{j\sqrt{j\pi K_r}(3t-2T_r)}{\sqrt{8}}\right))) \tag{3.6}$$

This equation can be split into three parts

$$\mathrm{rect}\left(\frac{t}{2T_r}\right)(\mathrm{erf}\left(\frac{j\sqrt{j\pi K_r}(t+2T_r)}{\sqrt{8}}\right) - \mathrm{erf}\left(\frac{j\sqrt{j\pi K_r}(3t-2T_r)}{\sqrt{8}}\right)), \tag{3.7}$$

$$\frac{\exp(j7\pi K_r t^2/8)}{j\sqrt{j}}, \tag{3.8}$$

$$\frac{1}{\sqrt{8K_r}}. \tag{3.9}$$

We observe that (3.7) is just acting as a windowing function that cannot have a magnitude greater than two. Equation (3.8) only effects the phase of the output of the matched filter. The important piece is (3.9). In a traditional SAR system, K_r is an incredibly large number, which drags the magnitude of the output of the matched filter to be significantly smaller than unit energy.

Therefore to detect which pulse was transmitted, we can observe the outputs from both matched filters and select the correct one based off of a predetermined metric. It should be noted that in the event of an incorrect classification the incorrectly selected pulse degrades the quality of the SAR image.

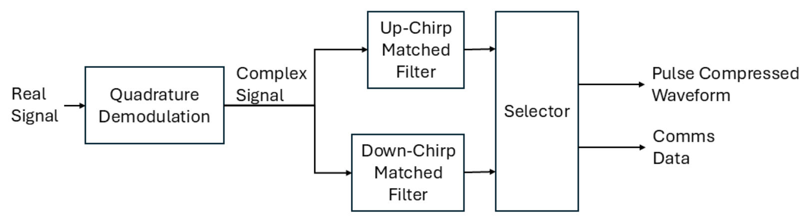

Fig. 3.1: Processing Diagram for Up/Down Modulation

Figure 3.1 is an example of the SAR processing flow that demodulates the incoming pulses. The quadrature demodulation is a needed step for all SAR processing as it gives us the complex signal from the real signal, it also removes the carrier frequency from the pulse. In future diagrams we assume this step is still occurring, even if not explicitly stated.

We can see that we need two individual matched filters for up and down-chirps, however, they can be run in parallel so it does not slow down processing. The outputs of both matched filters is then fed into our selector. The selector uses a metric to determine the most likely correctly pulse compressed output. There are many different possibilities, for example, looking for peak power or using a norm to look for the output with the most spikes.

The selector then passes through the chosen waveform to the rest of the SAR pipeline. It also returns a one or a zero to the comms system depending on which pulse was chosen.

3.2.1 SAR Simulation

To test out this pipeline we simulate a SAR system. Our simulation makes several key assumptions which may not accurately reflect a real SAR system depending on the application. This simulation is simply to show proof of concept.

We are trying to show in this section that given a high enough SNR, the pulses can be demodulated and that the demodulation process does not effect the SAR image quality. Therefore there is no Gaussian noise added to the system in this simulation. There is no background clutter, only point targets. All geometry is precisely known.

Fig. 3.2: Simulated SAR Image with no Comms Modulation

Figure 3.2 shows the output of this simple system after regular processing without our added JRC system. The image is low resolution, with each pixel representing ten meters. To process the phase history we used time-domain back-projection to give us a zoomed in subset of the image. As can be seen the point targets have been arrayed to form a smiley face.

Next figure 3.3 shows the SAR simulation where the sign of K_r has been modulated to transmit comms data. The peak energy metric was used to detect which modulation was used. We can see that the output is identical to the scenario where no demodulation was used. Therefore, we can conclude that comms system has little to no effect on the SAR image.

For comparison, figure 3.4 shows the output of the SAR system where every single pulse was misclassified and incorrectly pulse compressed. We can see that this causes noise throughout the image as well as smearing the image in the range axis. The targets

Fig. 3.3: Simulated SAR image using K-sign Modulation

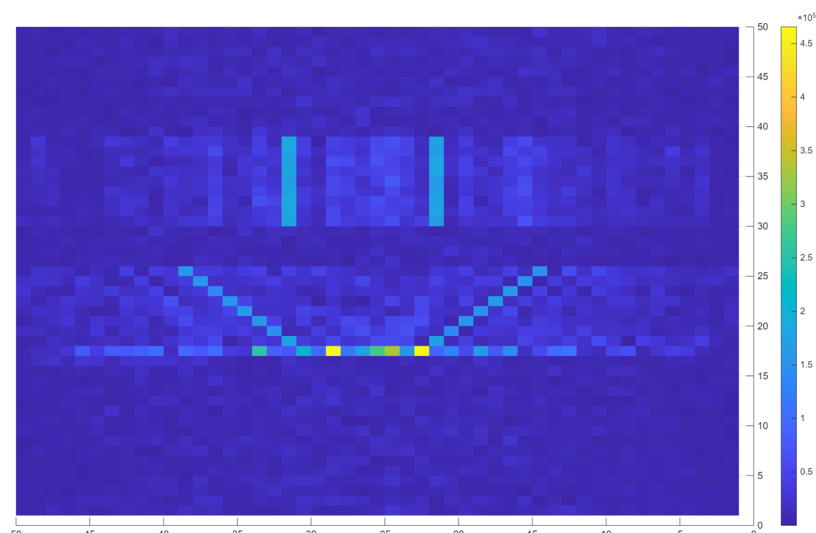

Fig. 3.4: Simulated SAR image using Incorrect K-sign Modulation

surprisingly can still be made out.

The smearing in range direction makes sense as the pulse compression is compromised and so we are unable to compress to a Sinc function. There is little to no smearing in the azimuth direction as the sign of K_r does not effect the Doppler of the image, and so azimuth compression is much less affected.

3.3 Probability of Error

We can now calculate the probability of a misclassified LFM chrip. This gives us the probability of bit error given an SNR and gives us a metric of how likely modulating results in a lower quality SAR image.

3.3.1 Theoretical Probability of Error

To compare the matched filter outputs, we assume we are comparing the outputs at peak power. This occurs at $t = 0$ for the convolutions done above. This gives us the comparison

$$T_r >> \frac{1}{\sqrt{8K_r}} |(2\,\mathrm{erf}\left(\frac{j\sqrt{j\pi K_r}(2T_r)}{\sqrt{8}}\right))| \approx \frac{1}{\sqrt{2K_r}}. \tag{3.10}$$

Because the output from the correct matched filter is significantly larger than a mismatch, we can approximate that the symbols are orthogonal to each other. Next, we calculate the distance between the waveforms. By distance we are treating the waveforms as vectors in an infinite dimensioned space. For ease of calculation we assume that both the up-chirp and down-chirp have unit energy. Using dot products we calculate the angle between the two vectors.

$$\cos(\theta) = \frac{s_{up} \cdot s_{down}}{s_{up} \cdot s_{up}} \tag{3.11}$$

Since we are approximating $s_{up} \cdot s_{down} = 0$, then $\theta = \pi/2$. We can now use the law of cosines to get the distance between the two waveforms with unit energy.

$$d_u = \sqrt{2((s_{up} \cdot s_{up})^2 - (s_{up} \cdot s_{up})(s_{up} \cdot s_{down}))} = \sqrt{2} \tag{3.12}$$

Now that we have the distance between the two vectors, we can determine the theoretical probability of bit error in a communications system. If we ignore that the signal is used in a radar scenario, we can safely model that the communications signal has additive white Gaussian noise. This noise moves the transmitted vector to somewhere else in the signal space. The matched filter detects whichever waveform the received signal is closest to.

To estimate the likelihood of an error in detection, we use a Q-function to find the probability of error, given a signal was transmitted. If $s(k)$ is the waveform that was transmitted, N_0 is the energy of the noise and E_s is the energy of the symbol then

$$P(E|s(k)) = Q(d/2\sigma), \tag{3.13}$$

$$\sigma = \frac{\sqrt{N_0}}{\sqrt{2}}, \tag{3.14}$$

$$d = d_u\sqrt{E_s}, \tag{3.15}$$

$$P(E|s(k)) = Q(\sqrt{\frac{E_s}{N_0}}). \tag{3.16}$$

Given that the probability of error is the same regardless of which symbol was transmitted we can use the theorem of total probability to get $P(E|s(k)) = P(E)$.

3.3.2 Simulation

To verify these results we simulated the communications system. As an example, we defined a radar symbol with bandwidth of 1 GHz, sampling frequency of 4 GHz, center frequency of 1.5 GHz, and a pulse duration of 1 microsecond.

A simulation of the waveforms was run using code found in Appendix A. In the simulation, the waveforms were given additive white noise of various strengths to simulate different SNRs. Figure 3.5 shows how the simulated probability of error compared with the

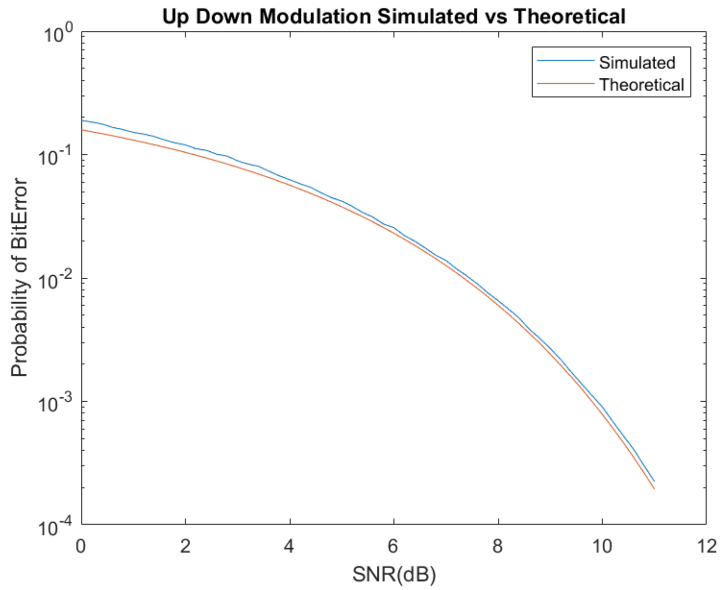

Fig. 3.5: Simulated vs Theoretical Bit Error for Modulated Sign of FM rate

theoretical bit error rate using the Q-function.

It can be seen that the simulated probability of bit error follows the theoretical bit error closely. The simulated error rate is slightly higher than the theoretical rate. This is to be expected due to the fact that the simulation is discrete and quantized in its calculations which adds some noise to the signal increasing its bit error rate.

Using this modulation method has the same bit throughput as binary phase shift keying (BPSK), an optimized signal constellation used in communications systems. If we compare the error rate of BPSK to modulating the sign of the FM rate, we see that BPSK is much more efficient. This is shown in figure 3.6. There is about a 3dB loss compared to BPSK. This is expected as BPSK is optimized for communications while modulating the sign of the FM rate is simply a way to transmit communications data on the back of existing radar waveforms.

3.4 Conclusion

We detailed how modulating the sign of the FM rate of existing SAR chips can be used to transmit communications data. The modulation technique won't have a negative

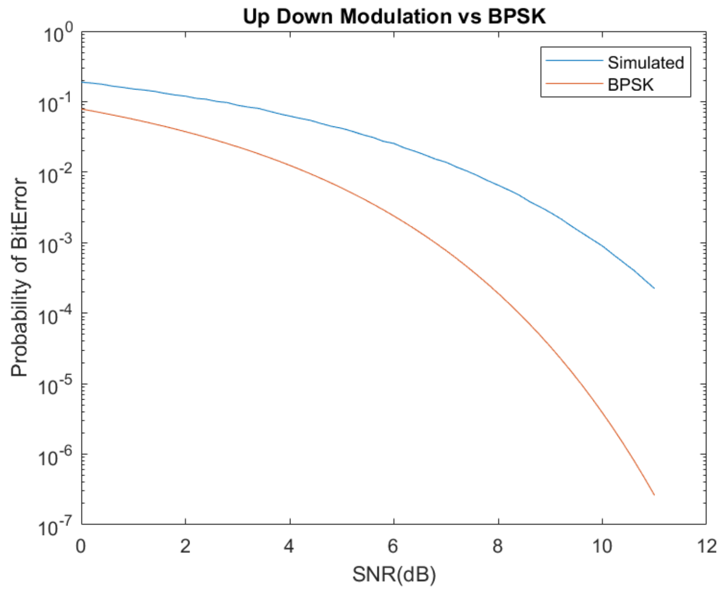

Fig. 3.6: Theoretical Bit Error for Modulated Sign of FM rate

effect on the SAR image as long as the received symbols are decoded correctly. When using modern chirp characteristics this modulation technique produces two waveforms that are approximately orthogonal to each other. This technique can be used to send communications data between two systems with success.

CHAPTER 4

Modulating the Phase of the LFM Chirp

We have demonstrated how to modulate the LFM chirp to give us two almost orthogonal waveforms. A natural direction to go is to see if we can form the negative version of these waveforms. As shown in this chapter, this allows us to form a signal constellation that approximates quadrature phase shift keying (QPSK).

4.1 Sign Modulation

Sign modulation involves changing the sign of the entire transmitted chirp. As shown in this section, it gives us a communications technique that approximates BPSK. It can also be combined with previous chapters modulation to create an even larger constellation.

4.1.1 Modulation Definition and SAR Image Resolution

Changing the sign of the waveform gives us two waveforms, allowing each waveform to transmit a single bit. The corresponding waveforms where b is the bit would be

$$s_b(t) = (-1)^b * \text{rect}(\frac{t}{T_r}) \cos(2\pi f_0 t + \pi K_r t^2). \tag{4.1}$$

Changing the sign of the LFM pulse does not affect K_r or T_r, therefore its bandwidth is unaffected. If we are able to accurately determine which pulse was sent and use the appropriate waveform when match filtering, then this modulation technique does not affect the resolution of the SAR image. The processing of these LFM pulses for a SAR image is discussed later on in the chapter.

4.1.2 Probability of Error

It is intuitive that if we take the dot product of an LFM chirp with its negative counterpart, the result is its negative energy. As this is BPSK, calculating the probability of error is simple. The distance between the two signals is twice the L2 norm of each LFM waveform. Therefore, with this modulation technique

$$d = 2\sqrt{E_s}, \tag{4.2}$$

$$P(E|s(k)) = Q(\sqrt{\frac{2E_s}{N_0}}). \tag{4.3}$$

4.1.3 Simulation

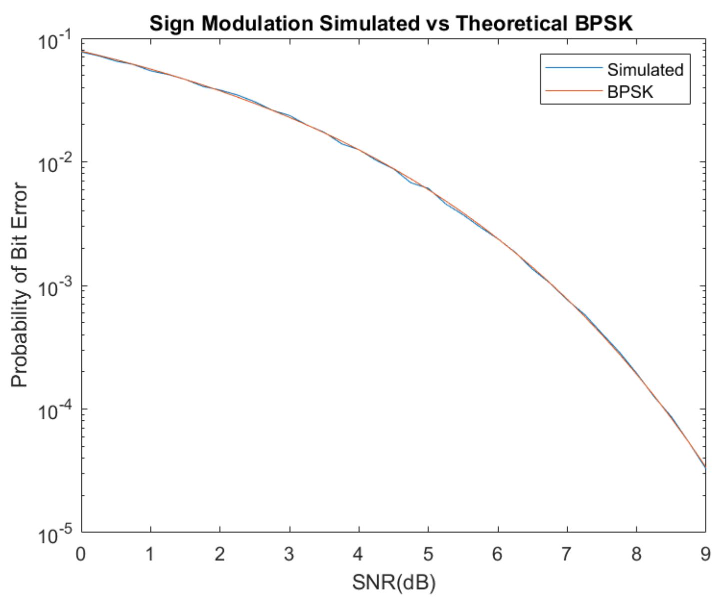

Fig. 4.1: Comparing traditional BPSK error rate with sign change modulation

To verify these results, we run the waveforms through the communications simulation. As can be seen in figure 4.1, our theoretical bit error rate matches the simulated result. This confirms that this modulation technique is in fact BPSK, as the theoretical bit error rate in equation 4.3 is the bit error rate for BPSK. It should be noted that this modulation technique would still have a relatively low data transfer rate as it can only transmit one bit per radar pulse which are relatively infrequent when compared to a typical communications system.

4.2 Phase Modulation

There is an important insight to be gained by realizing that, since the LFM chirp is a sinusoid, when we take its negative, we are just performing a phase shift on the pulse

$$s_{pul}(t) = -\operatorname{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2) = \operatorname{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2 + \pi). \tag{4.4}$$

This mirrors how modern communications systems function which modulate both phase and amplitude simultaneously to form their constellations. As noted in the literature review, we do not consider modulating the amplitude of the LFM chirp, however we still are able to modulate the phase.

4.2.1 Modulation Definition and SAR Image Resolution

This is the first modulation technique that we have considered that isn't a simple binary switch, but instead a continuous variable that we can change. Therefore, in theory, we could form a constellation containing an infinite number of symbols. In practice, this would be impractical as the probability of bit error would be one hundred percent. In the real world, an engineer designing the communications system would choose the best phase modulation for their needed bit transfer rate.

A typical phase modulation consists of a certain phase offset that divides into 2π evenly so that each symbol has an equal probability of bit error. For this section we explore 8-PSK to give us eight different phase offsets. This gives us an eight symbol constellation defined by the following equation where b is the byte being transmitted

$$s_b = \operatorname{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2 + 0.25 b\pi). \tag{4.5}$$

4.2.2 SAR image Processing

Adding a phase change to the LFM chirp does not alter the standard qualities that are needed for SAR. Therefore, we know that this signal constellation or any constellation that uses phase modulation maintains the quality of the SAR image if all the pulses are detected and processed correctly.

Because the phase offset is unchanged by time, it can be moved out of the matched filter integral. Therefore, we only need a single matched filter for all possible phase offsets. After matched filtering we can then look at the compressed pulse and classify which phase offset was transmitted. This can be a difficult task. Looking at equation 2.8 we can see that there is a phase offset based on the distance between the target and the antenna, which constantly changes.

If these changes are small or predictable from pulse to pulse then we can account for these offset changes. The distance between a target and the antenna is approximated using (4.6) where R_0 is the range of closest approach and V_r is the velocity of the platform.

$$R(\eta) = \sqrt{R_0^2 + V_r^2 \eta^2} \tag{4.6}$$

Substituting (4.6) into the linear phase from (2.8) we see that the phase changes according to

$$\theta(\eta) = -\frac{4\pi f_0 \sqrt{R_0^2 + V_r^2 \eta^2}}{c}. \tag{4.7}$$

We see from the equation that it's a quadratic where the rate of change is low when η is near 0. The parabolic nature becomes more apparent when η is larger. The solution to this problem depends on the physics of the problem. For example, in certain SAR senarios, the platforms would be moving slow enough compared to the PRF, that the average scene phase shift would be relatively small. That is, the phase between one pulse and the next would be similar.

The correct approach to take would depend heavily on the circumstances of the application. We could also use large phase shifts in modulation for better detection, however, this would result in a lower data throughput. There may be some applications where we could have an accurate knowledge of how the geometry of the scenario was changing the phase and account for that. Regardless it is a problem that can be solved and so this modulation technique is still worth investigating.

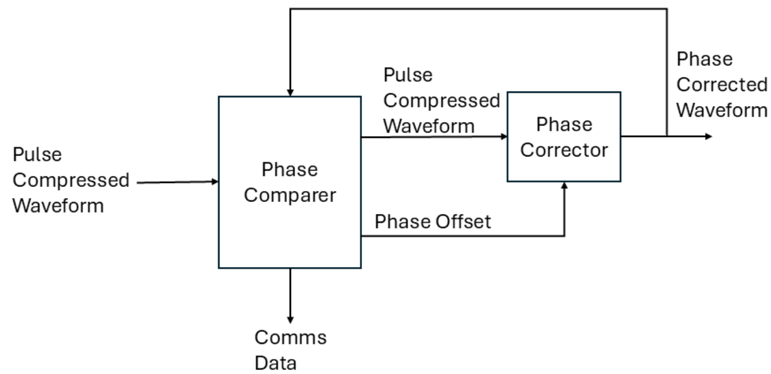

Fig. 4.2: Processing Diagram for Phase Modulation

Figure 4.2 is an example of how the phase modulation can be detected and corrected assuming the phase change between pulses is small. The process happens after pulse compression. When the compressed pulse is received, it is then compared to the last phase corrected pulse. The phase comparer then cycles between different possible phase offsets to see which one would result in the current waveform being the closest to the previous waveform.

After this selection, we output the waveform with predicted phase offset as well as the comms bit associated with it. The waveform is phase corrected and sent off for further SAR processing. A copy of the phase corrected waveform is also sent back to the phase comparer so that it can compare the next waveform with the previous waveform.

4.2.3 SAR Image Simulation

We can implement the proposed pulse correction algorithm by running the same simulation as the previous chapter. This time we add phase modulation, increasing the number of phase steps until the system begins to accrue errors. There are a few key insights that were gained.

First, the proposed system is unstable. Since we are only reading in the previously corrected waveform, if a waveform is misclassified, the system uses this as a new truth and continues to misclassify the waveforms. In a real system we would want to mimic modern communication systems which perform carrier phase synchronization using maximum likelihood estimates. The objective of this thesis is to demonstrate that the modulation technique is possible. Discussing the phase synchronizer is beyond the scope of the thesis.

A second insight is that we can handle smaller phase shift modulations if we either increase the PRF or decrease the velocity of the platform. This makes intuitive sense since doing so decreases how much the geometry changes between pulses. Therefore, the phase change between pulses would be smaller.

Fig. 4.3: Simulated SAR image using Phase Modulation

Simulating a platform two kilometers altitude, traveling at 40 m/s and with a PRF of one millisecond, we were able to create the SAR image figure 4.3. Using these parameters, we were only able to use a phase modulation between 0 and π. Anything higher resulted in misclassifications for this setup. As can be seen however, the demodulation process worked perfectly and the SAR image was unaffected.

Figure 4.4 shows the output of the simulation if we misclassify every phase shift. We can see that is has almost the opposite effect that mismatching the sign of K_r had. Now we have smearing in the azimuth direction and little smearing in range. This is because the pulses themselves were still pulse compressed properly, while the phase element of Doppler was inaccurate.

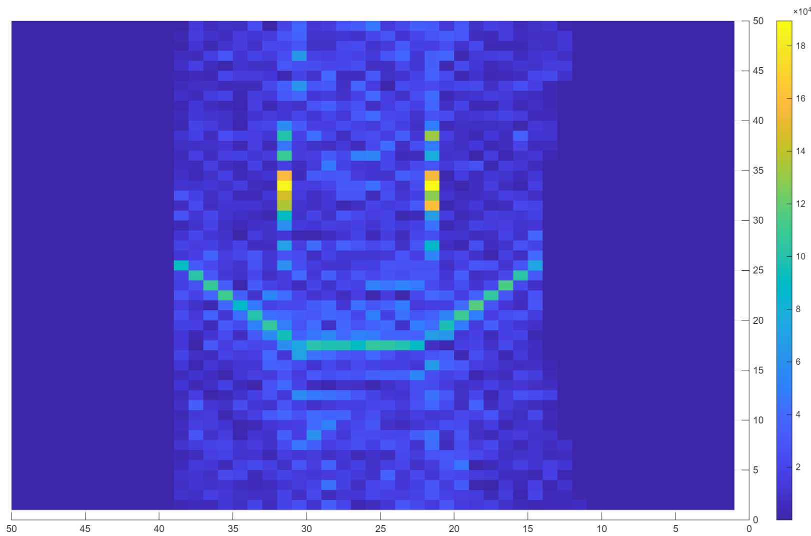

Fig. 4.4: Simulated SAR image using Incorrect Phase Modulation

4.2.4 Probability of Error

To accurately determine the theoretical probability of error we want to establish that there is a 2D plane that contains all the symbols in a phase modulated constellation. This is relatively easy to show. Let us take two generic symbols, s_{phaseA} and s_{phaseB} where

$$s_{phaseA} = \text{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2 + A), \tag{4.8}$$

$$s_{phaseB} = \text{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + \pi K_r t^2 + B). \tag{4.9}$$

Now we see if we can produce any other phase modulated symbol by taking a linear combination of s_{phaseA} and s_{phaseB}.

$$Cs_{phaseA} + Ds_{phaseB} = \sqrt{(C\cos(A) + D\cos(B))^2 + (C\sin(A) + D\sin(B))^2} \tag{4.10}$$

$$= \text{rect}\left(\frac{t}{T_r}\right)\cos(2\pi f_0 t + \pi K_r t^2 + \tan^{-1}\left(\frac{C\sin(A) + D\sin(B)}{C\cos(A) + D\cos(B)}\right)) \tag{4.11}$$

The first term is just a scaling factor that can be ignored. The inverse tangent is the part of interest. We can observe that with the correct selection of A and B we can get any different phase of LFM chirp. Therefore they are all on the same 2D plane.

If we assume that phase offsets of 0 and 0.5π are orthogonal basis vectors then the theoretical probability of error is that of 8-PSK. This error function can be found by transforming the problem into polar coordinates and basing the probability of error based on the angle around the unit circle [6]. The result is

$$P(E) = 2Q\left(\sqrt{\frac{E_{avg}}{N_0}2\sin^2\left(\frac{\pi}{8}\right)}\right), \tag{4.12}$$

$$P_b = \frac{2}{3}Q\left(\sqrt{\frac{E_{avg}}{N_0}6\sin^2\left(\frac{\pi}{8}\right)}\right). \tag{4.13}$$

4.2.5 Simulation

For the simulation we use the same parameters of the simulation performed in chapter 3. First we check to see how closely the vectors approximate a true 8-PSK constellation by projecting them onto their shared 2D plain. As we can see from figure 4.5 the assumption that this is approximately a traditional 8-PSK constellation holds.

Running the simulation, we can see that the bit error rate matches the theoretical bit rate error as shown in figure 4.6. It performs marginally worse. This is expected, as the constellation is not a perfect 8-PSK constellation.

4.3 Chirp Direction Combined with Phase Modulation

We now analyze how practical it is to add both the phase offset modulation technique and changing the sign of the FM rate. This allows us to form even larger and efficient signal constellations. To demonstrate how this can increase efficiency we simulate a constellation with 8 symbols and compare it to the previous example with just phase offset modulation.

4.3.1 Modulation Definition

It is relatively simple for us to define the signal constellation. The constellation that should be generated should be symmetrical. Therefore there is an even number of symbols in the constellation, half with an up-chirp and half with a down-chirp. The equation for

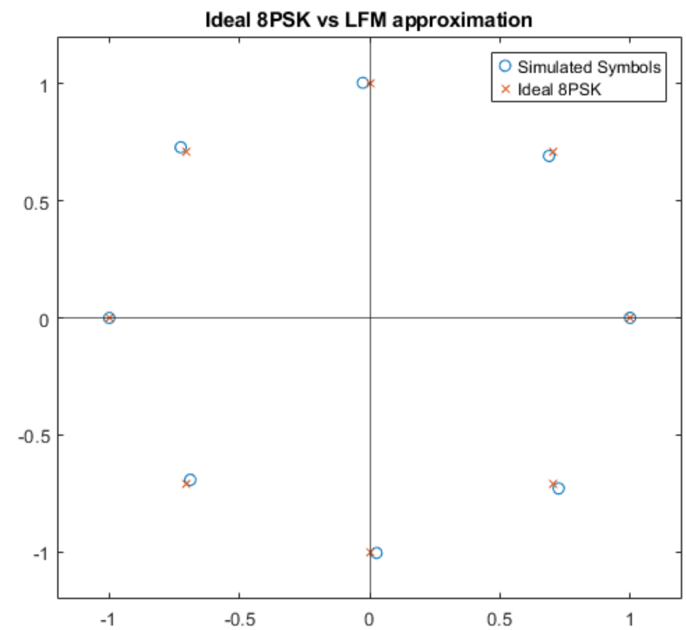

Fig. 4.5: Comparison of 8-PSK with simulation

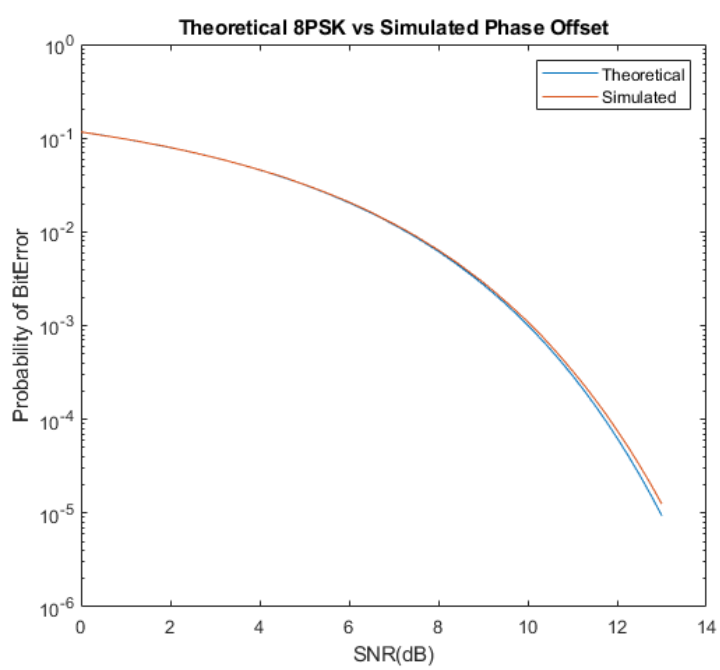

Fig. 4.6: Comparison of 8-PSK with simulation

this constellation where b is the value of the symbol and b_n is the number of symbols in the constellation is

$$s_b = \text{rect}(\frac{t}{T_r})\cos(2\pi f_0 t + (-1)^{\lfloor(2b/b_n)\rfloor}\pi K_r t^2 + (2\pi/b_n)b\pi). \tag{4.14}$$

We know that separately these modulation techniques do not interfere with the LFM chirp. Combining them does not change the bandwidth of the chirp and so it does not effect the properties of the LFM chirp.

4.3.2 SAR Image Processing

As the LFM chirp is unaffected, the key to maintaining the image quality is ensuring that the pulses are correctly processed. The first step is determining the direction of the chirp. Since adding a constant phase only effects the phase of the output of the matched filter, modulating the LFM chirp does not affect the double matched filter system that was discussed in Chapter 3. After we have determined the direction of the LFM chirp and have compressed the pulse using the right matched filter, we then run the processing to account for the phase change.

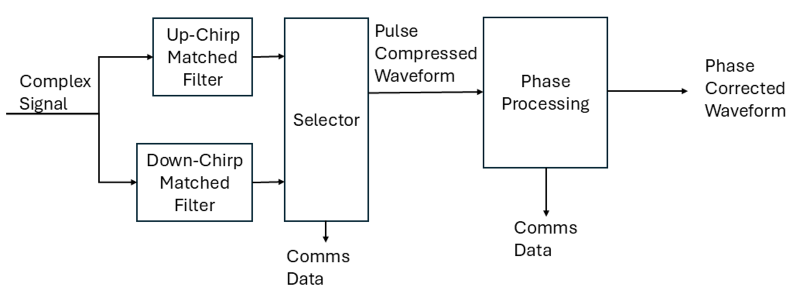

Fig. 4.7: Processing Diagram for Phase and K Sign Modulation

This is shown in figure 4.7. The phase processing box represents all parts shown in figure 4.2. The different comms data outputs represent different significant bits that would be combined after processing.

As a side note: It is beyond the scope of this thesis, but it would be practical to use error correction coding on the comms data. If the codes do detect an error, besides simply correcting the comms data, the SAR data could also be reprocessed with the correct demodulation as each bit represents some demodulation that needs to be performed.

4.3.3 Theoretical Probability of Error

We now compare the previously discussed 8-PSK constellation with this new eight piece constellation that uses both modulation techniques.

We have already analyzed the distance between symbols with different phase modulations and symbols with different signs of the FM rate. Regardless of the phase given to the LFM pulses, the scaling factor from the output from the incorrect matched filter is still low enough that we can consider the symbols to be orthogonal. This means that each symbol is orthogonal to six other symbols and is the negated version of the last symbol.

To simplify the estimate, we use the union bound to find an upper bound for the probability of error. The union bound is the sum of probability of failure between each point and all other points. It can be represented by (4.15) where M is the number of symbols. $d_{n,m}$ is the distance between pulse n and pulse m.

$$P(E) \leq \frac{1}{M} \sum_{m=0}^{M-1} \sum_{\substack{n=0 \\ n \neq m}}^{M-1} Q\left(\frac{d_{n,m}}{2\sigma}\right) \tag{4.15}$$

$$P_b \leq \frac{1}{M \log_2(M)} \sum_{m=0}^{M-1} \sum_{\substack{n=0 \\ n \neq m}}^{M-1} Q\left(\frac{d_{n,m}\sqrt{\log_2(M)}}{2\sigma}\right) \tag{4.16}$$

Using this union bound we can calculate the probability of bit error for this combination of modulation techniques. This results in the following probability of bit error

$$P_b = \frac{1}{3}\left(Q(\sqrt{\frac{6E_b}{N_0}}) + 6Q(\sqrt{\frac{3E_b}{N_0}})\right). \tag{4.17}$$

4.3.4 Simulation

Figure 4.8 shows the simulated error rate compared to the theoretical error rate. We can see that the simulation performs slightly better than the theoretical probability of error. This is to be expected as a union bound estimation slightly overestimates the rate of error and simply gives us the upper bound of the error.

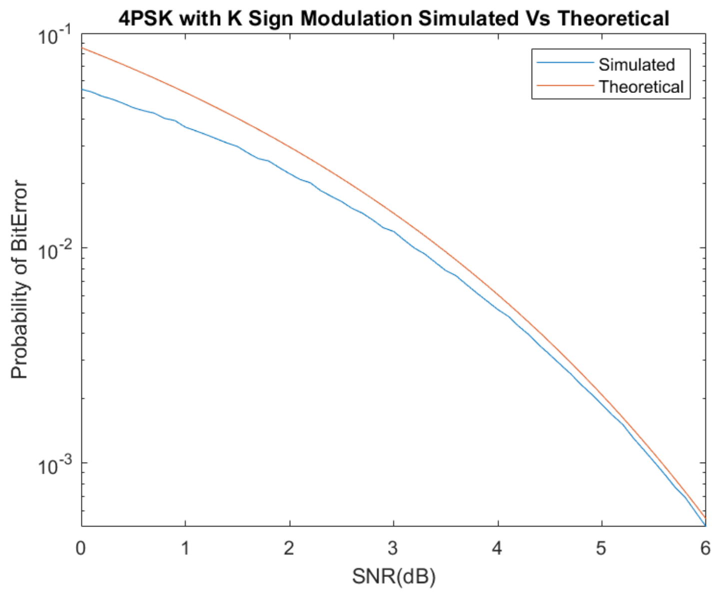

Fig. 4.8: Comparison of 4PSK with K-sign change theoretical vs simulation

Since this constellation has the same number of symbols as the 8-PSK system discussed previously in the chapter, they have the same bit transfer rate. Figure 4.9 compares their error rates. We can clearly see that including the K sign modulation significantly improves our rate of error by a little more than 4dB.

4.4 Conclusion

We have demonstrated that similar to a traditional communications system, we can modulate the phase of the signal to produce a constellation. We have also shown that we can combine the previous modulation of up/down-chirps with this phase modulation to create even larger constellations. Compared with an only phase modulated constellation of the same size, these combined constellations are more effective as they have more dimensions that are relatively orthogonal to each other.

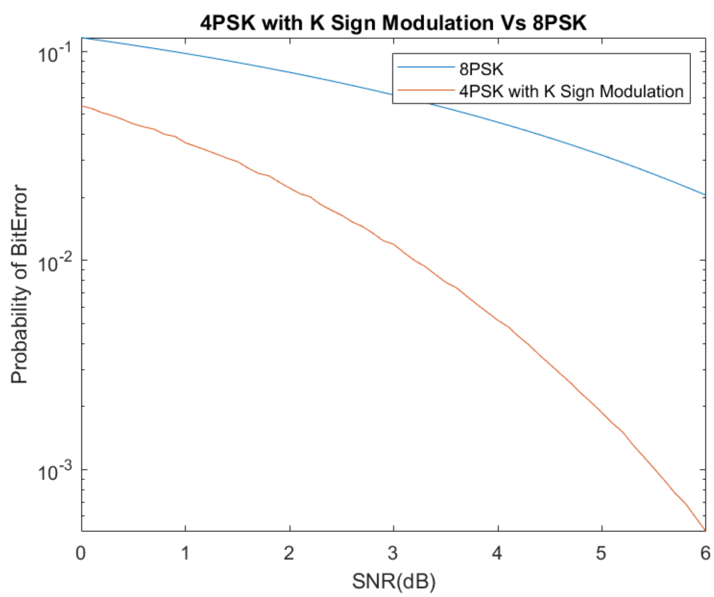

Fig. 4.9: Comparison of 8-PSK with 4-PSK with K-sign change

CHAPTER 5

Modulating the Time Delay

The next modulation technique that we analyze is modulating the time delay between pulses. In a traditional communications system the data is transmitted continuously until all the data has been transmitted. In the SAR scenario each of the symbols has a time delay before the next symbol is transmitted. This chapter investigates the effectiveness of modulating that time delay to transmit information.

5.1 Orthogonal Time Delays

We can divide this modulation technique into two classes. The first class is when the size of the time delay is large enough that the signals become orthogonal to each other. For this to happen the time delays have to be greater than or equal to T_r.

5.1.1 Modulation Definition

It is relatively simple to define this modulation technique as all we are doing is time delaying the signal. It should be noted that the time delay must be reflected in both the rect function and the LFM cosine otherwise the center frequency will change if we only time delay the rect function.

$$s_b = \text{rect}\left(\frac{t - bT_r}{T_r}\right)\cos(2\pi f_0(t - bT_r) + \pi K_r(t - bT_r)^2) \tag{5.1}$$

The LFM chirp remains unaltered so the resolution of the SAR image is unaffected. In practice the constellation needs to be designed such that it still allows for the required pulses per second as using this method slows down the rate at which we can transmit pulses.

5.1.2 SAR Image Processing

Processing the time delayed LFM pulses is relatively straight forward. We once again only need a singular type of matched filter. The pulse compressed chirp shares the same time-delay as the uncompressed pulse. If we are able to detect the time delay that is assigned to the LFM chirp, then we are able to compensate by forwarding in time the compressed pulse.

It should be noted that the time delay will also be affected by the distance between the transmitter, receiver, and reflector. This is the exact same difficulty that the phase modulation technique suffered from. Therefore it can be overcome using the techniques mentioned in chapter 4, depending on what is most appropriate for the application.

Depending on what hardware is being used there are multiple ways to process the demodulation pipeline. If we have the ability to parallel compute, then having a separate matched filter for each possible time delay could save processing time. The output of these matched filters would then be individually processed to determine which modulation was applied. This could either be done by summing the energy in a certain window, or comparing the output with the last pulse assuming the change between pulses was small.

Figure 5.1 shows this parallel computing configuration. The selector unit may keep the last waveform if it is comparing between waveforms. It should be noted that it would just need to compare magnitude. It also should be noted that such a comparison would need to be beyond simply comparing the various range bins. It would need to compare nearby ones as well to deal with range cell migration.

5.1.3 SAR Image Simulation

Once again we use the same SAR simulation to test how practical it is to use this time delay modulation. We use the same parameters as before. For modulation we have four time delayed waveforms. Figure 5.2 shows the output, demonstrating that it does in fact work as a modulation technique.

Figure 5.3 shows the output if all waveforms are misclassified. As we can see this is the most harmful of the modulation techniques if it is classified incorrectly. In theory, if all of the waveforms were misclassified with the same time delay, the image should be fine

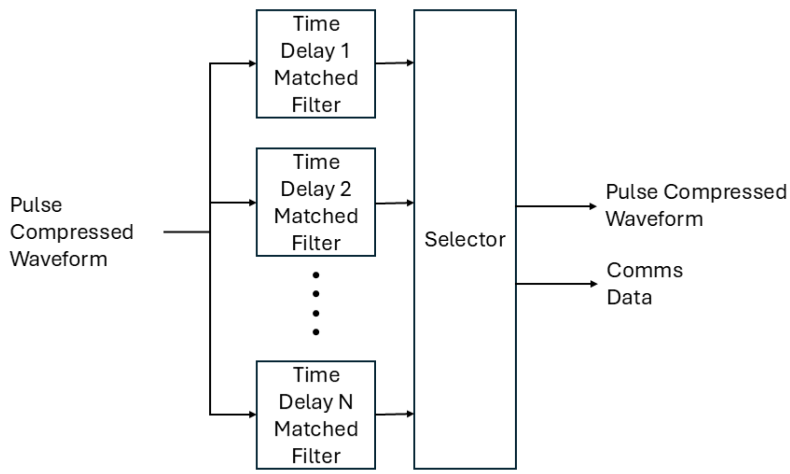

Fig. 5.1: Processing Diagram for Time Delay Modulation

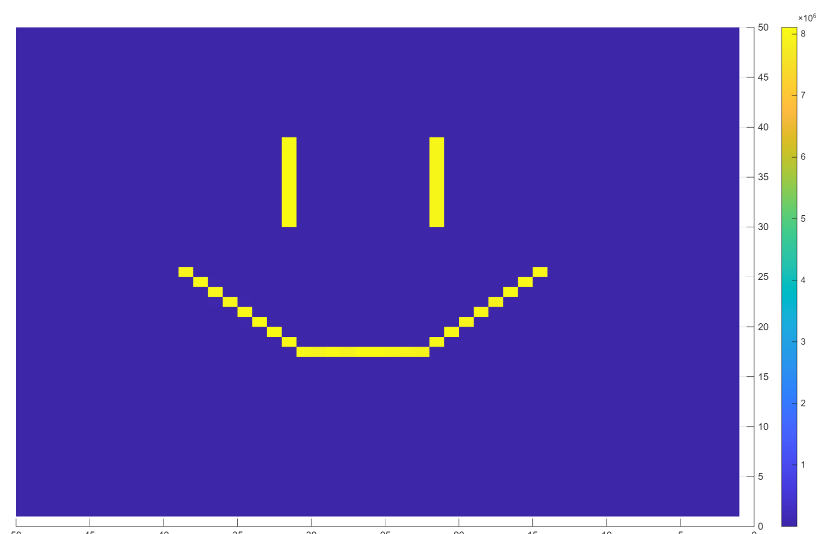

Fig. 5.2: Simulated SAR image using Time Delay Modulation

just shifted in range. However, with all the possible time delays together, all the images are adding destructively to give us what looks like mostly noise.

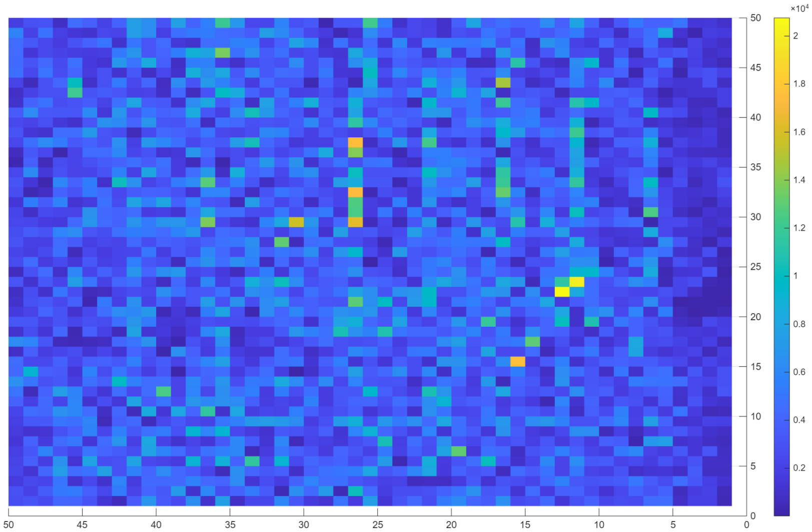

Fig. 5.3: Simulated SAR image using Incorrect Time Delay Modulation

5.1.4 Probability of Error

The estimated probability of error is easy to calculate. Because there is no overlap in time between the different signals, they are strictly orthogonal, no matter how many symbols are placed into the constellation. The probability of error is

$$P(E) = (M-1)Q\left(\sqrt{\frac{E_s}{N_0}}\right), \tag{5.2}$$

$$P_b = \frac{(M-1)}{\log_2(M)}Q\left(\sqrt{\frac{\log_2(M)E_s}{N_0}}\right). \tag{5.3}$$

5.1.5 Simulation

Figure 5.4 shows the accuracy of our estimate with a simulated 8 symbol constellation. Once again, the estimate is an upper bound of the rate of error as we are using the union bound.

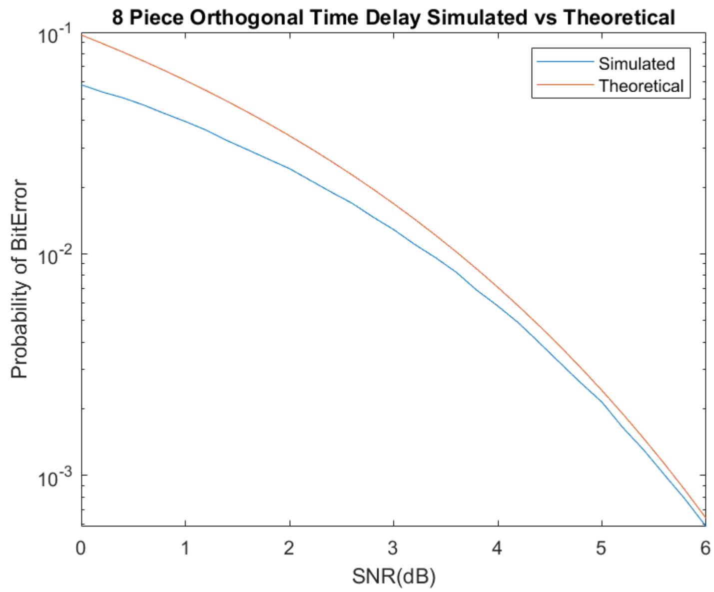

Fig. 5.4: Error rate of orthogonal time-delay constellation with 8 symbols

The more symbols that we add to this constellation the better the rate of error becomes. This is demonstrated in figure 5.5 where we plot the upper bound of rate of error of a two piece constellation to a 64 piece constellation. The trade off is that the total length of each symbol increases. The amount of orthogonal symbols you would be able to add depends on the pulse rate and the SAR scenario.

5.2 Non-Orthogonal Time Delays

The next modulation technique is to allow smaller time delays so that parts of each symbol in the constellation overlap with each other. This should increase the likelihood of bit error, but decrease the needed time window for the given constellation. This allows us to form larger constellations than the strictly orthogonal case with the same pulse rate and width.

5.2.1 Modulation Definition

The definition of this modulation technique is almost identical to the orthogonal case, however, this time the time delay factor is not required to be multiples of T_r. The definition

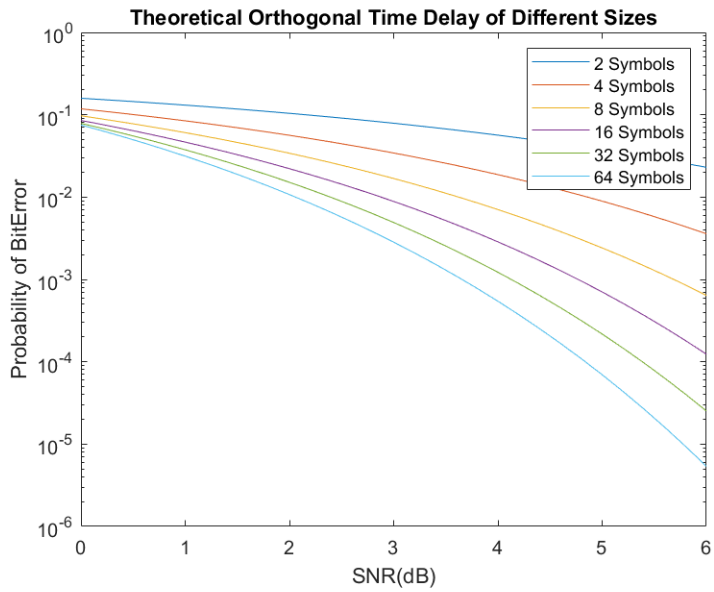

Fig. 5.5: Error rates of orthogonal constellations of different sizes

is shown in (5.4) where the delay is $D < T_r$. Depending on the size of D and how many symbols are in the constellation, some symbols may still be strictly orthogonal to each other.

$$s_b = rect\left(\frac{t - bD}{T_r}\right)\cos(2\pi f_0(t - bD) + \pi K_r(t - bD)^2) \tag{5.4}$$

5.2.2 Theoretical Probability of Error

Our intuition might be that the symbols that are overlapping would have a significant decrease in performance compared to the strictly orthogonal time delays. However, our symbols are LFM chirps which have been specifically designed to give incredibly accurate range compression.

As we already showed, the output of a correctly assigned matched filter for an LFM pulse is a sinc function with a window applied (3.4). Even though the pulse length is T_r, after we pulse compress the LFM chirp, the main lobe of the sinc function is $\frac{1}{T_r K_r}$. Therefore, we can model the non-orthogonal time delayed pulses to be close to orthogonal as long as we use the post-compressed LFM chirp to determine the time delay and the time delays are at least larger than $\frac{1}{T_r K_r}$.

5.2.3 Simulation

To demonstrate how close to orthogonal the symbols are, figure 5.6 shows the theoretical probability of bit error assuming that the wave forms are orthogonal compared to the actual simulation. The simulation uses 8 waveforms each time delayed by 0.2 of the waveform length. We can see that once again the theoretical probability of error closely matches and gives the upper bound for the simulated probability of error.

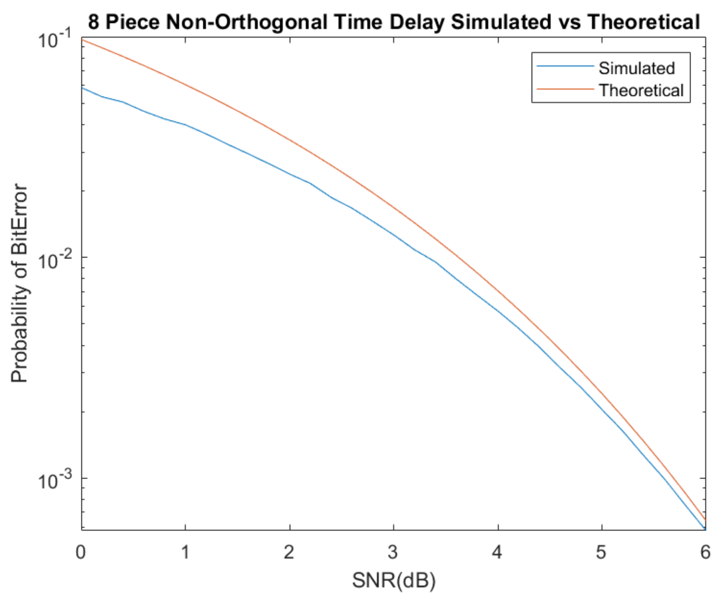

Fig. 5.6: Simulation of Non-Orthogonal Time Delay Modulation Technique

5.3 Combined Techniques

We can now show the practicality of combining all three currently discussed modulation techniques. Doing so allows us to squeeze a better comms bit rate through the system without significantly impacting the SNR of the system.

5.3.1 SAR Image Processing

So far each technique maintains the integrity of the LFM chirp. We can also see from the LFM chirp equation that adding a time delay will not effect the pulses phase or how it responds to the up/down matched filter. Therefore as long as each part is processed separately we are able to demodulate the pulse and retrieve the comms data encoded in it.

Figure 5.7 shows a possible system capable of demodulating all three techniques. The first step is to pulse compress the chirp to give us better peaks to detect modulation from. Since we are modulating the sign of K_r we need two separate matched filters.

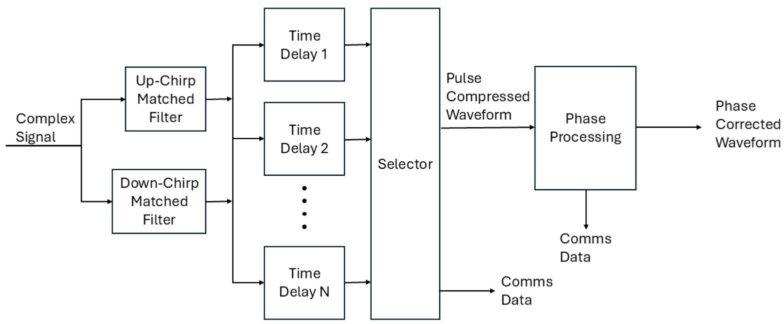

Fig. 5.7: Processing Diagram for Phase, K Sign, and Time Delay Modulation

The outputs of both filters then need to be sent through all possible time delays. This allows us to check the correct time window for energy or any other metric being used to select the best waveform. From this the correct waveform is passed on, as well as some of the communication bits. Finally this is sent to the same phase processing discussed in the previous chapter. We now have all the comms data as well as the final demodulated waveform.

Figure 5.8 confirms that the SAR image will remain unaffected.

5.3.2 Simulation

For our simulation we use the same waveforms that we used in section 4.3.4. We quadruple the amount of symbols by having four different time delays, each spaced 0.2 of a wavelength from the next. This gives us 32 symbols with each symbol orthogonal to all other symbols except the symbol that has a π phase shift from it. Under those assumptions we get the theoretical probability of error

Fig. 5.8: Simulated SAR image using all discussed modulation types

$$P_b = \frac{1}{5}\left(Q(\sqrt{\frac{6E_s}{N_0}}) + 30Q(\sqrt{\frac{3E_s}{N_0}})\right). \tag{5.5}$$

Figure 5.9 shows the theoretical compared with the simulated probability of error. We can clearly see that there is a mismatch between the theoretical error and the simulated error. This shows that as we add more symbols that are approximately orthogonal we get farther and farther away from the assumption we made that they are truly orthogonal. Also the simulation is discrete rather than continuous to simulate the real world. Some of the discrete waveforms when convolved show that the energy of a matched filter between different waveforms can go as high as one fifth of the energy of a perfect match.

When we correct our assumptions by assuming that the waveforms are almost orthogonal we get a better upper bound, as seen in figure 5.10. We have now shown that at high waveform counts we cannot use the assumption that the waveforms are orthogonal.

5.4 Conclusion

From our analysis we can see that using time delays is an effective way to modulate LFM chirps for communication signals. Due to pulse compression, the time delayed waveforms can be approximately orthogonal to each other. However, when we use multiple discrete

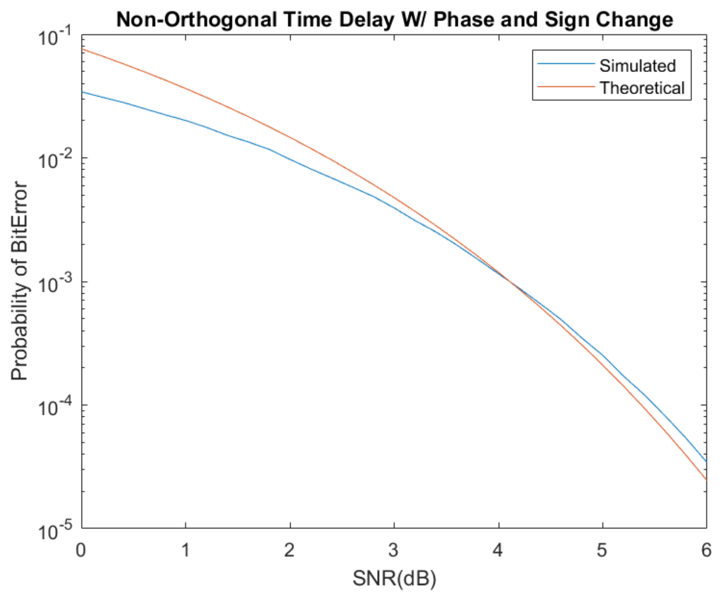

Fig. 5.9: Simulation of 32 piece signal constellation using all modulation techniques

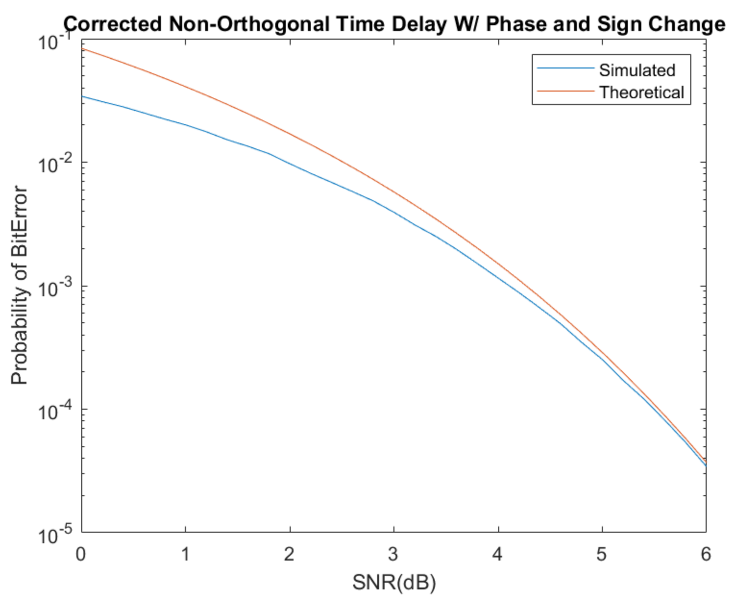

Fig. 5.10: Simulation of 32 piece signal constellation using all modulation techniques with corrected theoretical error rate

waveforms, the orthogonal assumption cannot be used for accurate predictions. The main drawback to this method is that is does lengthen the symbol length to be longer than an LFM chirp, which lowers our bit rate as it decreases our PRF.

CHAPTER 6

Conclusion

6.1 Final Analysis

Now that we have seen how the different modulation techniques perform, we can see what constellations are the most efficient for a given bit per symbol rate. For a bit rate of one bit per chirp, the best signal constellation would be formed by an LFM chirp and an LFM chirp with a phase offset of π. Regardless of what the original phase of the LFM chirp is or the sign of K_r this is the signal constellation with the lowest probability of bit error.

If a higher bit rate is needed per pulse, then the designing engineer has four types of modulation that they can add to the signal constellations that are approximately orthogonal. The options are, in order of most orthogonal to least:

- Time shift the pulses so that there is no overlap between them.
- Add a phase shift of $\pm\pi/2$.
- Change the sign of K_r
- Time shift the pulses so that there is some overlap between them.

The last two modulation techniques might need to have their order switched depending on what the original LFM chirp looks like, how large of the time shift, and what sampling frequency is used. The time shift would only be used if it is practical to increase the length of the LFM pulse.

If after adding all possible approximately orthogonal modulation techniques an even higher bit rate is needed, then smaller phase shifts can be added. This will increase the probability of bit error as this modulation technique is not orthogonal. Once again the exact increase of the bit error will be determined by the characteristics of the LFM chirp.

6.2 Concluding Summary

Throughout this thesis we have attempted to demonstrate a simple way to modulate LFM chirps. The goal was for the chirps to allow communication between a radar transmitter and receiver. It was important that the modulation did not effect the bandwidth of the LFM chirp. By tweaking the LFM pulses in ways that did not change the bandwidth we were able to show that communication signals could be sent.

The first modulation technique was switching between up-chirps and down-chirps by changing the sign of K_r. We showed that an up-chirp and a down-chirp were practically orthogonal to each other. We also demonstrated that this modulation technique can be used for communications. Using this technique by itself only allows for one bit to be transferred per pulse.

The second technique was to add phase changes to the LFM pulse. We demonstrated that we could form similar constellations to a traditional communications system with this technique, with the caveat that we could not change the amplitude of the symbols. We then showed that this technique could be paired with up-chirps and down-chirps to double the bit rate of the system.

The last technique that we analyzed was to add a time delay to the pulses. We could cause the waveforms to be orthogonal to each other by making the time delay be equal to the chirp length. We could also use a shorter time delay. This decreases the size of the waveforms, but as we showed, causes the waveforms to no longer be perfectly orthogonal to each other. Just like the phase change, this modulation technique can be added to all the other modulation techniques to make even larger signal constellations.

There are many further directions that this research could be taken. These techniques work because they maintain the bandwidth of the LFM chirp, but they are not the only techniques that do so. It would be useful to see if changing the phase of the LFM chirp throughout the pulse to encode communication bits would be possible. It's possible that even though the bandwidth of the chirp remains intact with this method, that it would still degrade the quality of any SAR images that were formed from the chirps.

We have shown that, using very simple modulation techniques on an LFM chirp, we can create waveforms that be used in a JRC system. Each technique produced waveforms that maintain the general structure of an LFM chirp. As long as the symbols are demodulated correctly, the modulation would not affect any post processing when forming an image.

REFERENCES

[1] Wang and Deng, *Bistatic SAR System and Signal Processing Technology*. Springer Singapore, 2018.

[2] A. R. Chiriyath, B. Paul, and D. W. Bliss, "Radar-communications convergence: Coexistence, cooperation, and co-design," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 1, pp. 1–12, 2017.

[3] Z. Feng, Z. Fang, Z. Wei, X. Chen, Z. Quan, and D. Ji, "Joint radar and communication: A survey," *China Communications*, vol. 17, no. 1, pp. 1–27, 2020.

[4] M. Roberton and E. Brown, "Integrated radar and communications based on chirped spread-spectrum techniques," in *IEEE MTT-S International Microwave Symposium Digest, 2003*, vol. 1, 2003, pp. 611–614 vol.1.

[5] J. Price and T. Goble, "Signals and noise," in *Telecommunications Engineer's Reference Book*, F. Mazda, Ed. Butterworth-Heinemann, 1993, pp. 10–1–10–15. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750611626500162

[6] M. Rice, *Digital Communications: A discrete-time approach*. Upper Saddle River, NJ: Pearson/Prentice Hall, 2009.

[7] I. G. Cumming and F. H. Wong, *Digital Processing of Synthetic Aperture Radar Data*. Artech House Publishers, 2005.

[8] N. Sakar, P. Prats-Iraola, and M. Rodriguez-Cassola, "Intrapulse effect compensation method for SAR systems with up- and down-chirp modulation," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

APPENDICES

APPENDIX A

Example Appendix with Computer Code

A.1 Bit Error Calculation

```
function ErrorRate = CalculateBitError(s, SNRs)
% s: array of the symbols in the constilation
% SNRs: array of the signal to noise ratios to test
% ErrorRate: array of the rate of error at the SNRs

% How many bits of information can be transerfed in a symbol
bitcount = log2(size(s,1));
% Singal Energy per bit
Eb = norm(s(1,:),2)^2/bitcount;

% How many errors to find before calculating error rate
ErrorThreshhold = 10000;

BitErrors = zeros(size(SNRs));
Symbols = zeros(size(SNRs));
j = 0;
for SNRdB = SNRs
    % Log the current SNR being calculated
    SNRdB
    j = j +1;
    % Find the standard deviation of the noise
    N0 = Eb/(10^(SNRdB/10));
    sigma = sqrt(N0/2);
    SymbolErrorCount = 0;
    BitErrorCount = 0;
```

```
    SymbolsCount = 0;
    % Run the simulation until the threshold is reached
    while(SymbolErrorCount < ErrorThreshhold)
        % Choose a symbol and add noise to it
        SymbolsCount = SymbolsCount + 1;
        symbol = randi([0,size(s,1)-1])+1;
        signal = s(symbol,:);
        signal = signal + randn(size(signal))*sigma;
        symbolHat = -1;
        bestFit = 0;
        for i = 1:size(s,1)
            % Find the best match for the recieved symbol
            matchedfilter = s(i,:);
            filterOut = sum(matchedfilter .* signal);
            if (filterOut > bestFit)
                bestFit = filterOut;
                symbolHat = i;
            end
        end
        % if its not a correct match flag an error
        if(symbolHat ~= symbol)
            SymbolErrorCount = SymbolErrorCount + 1;
        end
    end
BitErrors(j) = BitErrorCount;
Symbols(j) = SymbolsCount;
end
ErrorRate = ErrorThreshhold./(Symbols*bitcount);
end
```

A.2 SAR Simulation

```
    % Preloop Setup
image = zeros((Xmax- Xmin)/pixelScale,(Ymax- Ymin)/pixelScale);
i = 1;
firstPulse = 1;
phaseBit = 0;
phaseMod = 0;
phaseChangeSize = 1;
timeDelay = 0.25*PulseWidth;
timeModCount = 4;

% For each slowtime
for slowT = 0:Prf:TotalTime
    planePos = [0, Vel* slowT, Alt];
    kSignBit = randi(2); % 1 is Down 2 is up
    completeReturn = zeros(size(fastT));
    Km = K *(-1)^kSignBit;
    % For each target get fastime returns
    for Targ = 1:size(Targets, 1)

        range = RangeD(planePos, Targets(Targ,:));
        windowStart = floor((2* range/c - PulseWidth/2)/SamplePeriod)- startSamp;
        windowStop = ceil((2* range/c + PulseWidth/2)/SamplePeriod) - startSamp;
        window = [zeros(1,windowStart), ones(1,windowStop-windowStart), zeros(1,
            size(fastT,2)-windowStop)];
        pulseReturn = exp(1j*((-4*pi*f0*range/c) + pi*Km*(fastT*SamplePeriod - 2*
            range/c).^2 + phaseMod));
        pulseReturn = pulseReturn .* window;
        completeReturn = completeReturn + pulseReturn;
    end
    % Deal with K-Sign Modulation
    compressedUp = conv(completeReturn, matchedFilterUp,"same");
    compressedDown = conv(completeReturn, matchedFilterDown, "same");
```

```
if (max(abs(compressedUp)) < max(abs(compressedDown)))
compressed = compressedUp;
    if (kSignBit == 1)
        "Mismatch Failure"
    end
else
compressed = compressedDown;
    if (kSignBit == 2)
        "Mismatch Failure"
    end
end
% Deal with Phase Modulation
if (firstPulse == 0)
    phaseChange = zeros(phaseChangeSize, 1);
    phaseCorrected = zeros(phaseChangeSize, size(compressed,2));
    for phaseModPos = 1:phaseChangeSize
        phaseCorrected(phaseModPos, :) = compressed*exp(-2*pi*1j/phaseModPos);
        phaseChange(phaseModPos) = sum(abs(wrapToPi(angle(phaseCorrected(
            phaseModPos, :))-angle(previousCompressed))));
    end
    bestMatch = find(phaseChange == min(phaseChange));

    if (bestMatch ~= phaseBit)
        "Phase Mismatch Failure"
    end
    compressed = phaseCorrected(bestMatch,:);
end
firstPulse = 0;
phaseBit = randi(phaseChangeSize);
phaseMod = 2*pi/phaseBit;
previousCompressed = compressed;
```

```
    % Back projection Algorithm
    i = i + 1;
    xcnt = 0;
    for X = Xmin:pixelScale:(Xmax-pixelScale)
        ycnt = 0;
        xcnt = xcnt + 1;
        for Y = Ymin:pixelScale:(Ymax-pixelScale)
            ycnt = ycnt + 1;
            rangetoPix = RangeD(planePos, [X,Y,0]);
            sample = 2* rangetoPix/c/SamplePeriod;
            value = exp(1j*2*pi*f0*2* rangetoPix/c)*interp1(fastT, compressed,
                sample);
            image(xcnt, ycnt) = image(xcnt, ycnt) + value;
        end
    end

end
```

A.3 Chapter 3 Graphs

```
% Define LFM Chirp Properties
SamplePeriod = 1/(1.2e9);
BandWidth = 1e9;
CenterFreq = 10e9;
PulseWidth = 1e-6;
K = BandWidth/PulseWidth;
t = 0:SamplePeriod:PulseWidth;

%% Setup Up/Down Chirp Constilation
startSNRdB = 0.0;
stopSNRdB = 11.0;
stepSNRdB = 0.2;
```

```
SNRs = startSNRdB:stepSNRdB:stopSNRdB;
s0 = cos(2*pi*CenterFreq*t + pi*K*t.^2);
s1 = cos(2*pi*CenterFreq*t - pi*K*t.^2);
s = [s1; s0];
% Find the error rate
bitErrorRate = CalculateBitError(s, SNRs);

%% Calculate Theoretical Error
TheoError = zeros(size(SNRs));
BPSKError = zeros(size(SNRs));
j = 0;
s0s0 = 1;
for SNRdB = SNRs

    j = j +1;
    N0 = s0s0/(10^(SNRdB/10));
    TheoError(j) = UpDownErrorFun(s0s0, N0);
    BPSKError(j) = BPSKErrorFun(s0s0, N0);
end

%% Plotting
clf;
figure(1)
semilogy(SNRs, bitErrorRate)
title("Up Down Modulation Probability of Bit Error")
ylabel("Probability of BitError")
xlabel("SNR(dB)")

figure(2)
semilogy(SNRs, bitErrorRate)
hold on
semilogy(SNRs,TheoError )
```

```
title("Up Down Modulation Simulated vs Theoretical")
ylabel("Probability of BitError")
xlabel("SNR(dB)")
legend("Simulated", "Theoretical")

figure(3)
semilogy(SNRs, bitErrorRate)
hold on
semilogy(SNRs, BPSKError)
title("Up Down Modulation vs BPSK")
ylabel("Probability of BitError")
xlabel("SNR(dB)")
legend("Simulated", "BPSK")



function E = BPSKErrorFun (Eb, N0)
E = qfunc(sqrt(2*Eb/N0));
end

function E = UpDownErrorFun(Eb, N0)
E = qfunc(sqrt(Eb/N0));
end

function out = qfunc(x)
out=erfc(x/sqrt(2))/2;
end
```

A.4 Chapter 4 Graphs

A.1 Constellation Graph

```
SamplePeriod = 3.3e-9;
```

```
BandWidth = 1e9;
CenterFreq = 15e9;
PulseWidth = 1e-6;
K = BandWidth/PulseWidth;
t = 0:SamplePeriod:PulseWidth;

s0 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0);
s1 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.25*pi);
s2 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.5*pi);
s3 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.75*pi);
s4 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1*pi);
s5 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.25*pi);
s6 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.5*pi);
s7 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.75*pi);
s = [s0; s1; s2; s3; s4; s5; s6; s7];

X = zeros(8,1);
Y = zeros(8,1);
for i = 1:8
   A = sum(s0.*s(i,:));
   B = sqrt(sum(s(i,:).*s(i,:))^2 - A^2);
   A = A/sum(s0.*s0);
   B = B/sum(s0.*s0);
   X(i) = A;
   if i > 5
      B = -B;
   end
   Y(i) = B;
end

clf
figure(1);
```

```
plot(X,Y,'o')
hold on
s = 1/sqrt(2);
X1 = [1; s; 0; -s; -1; -s; 0; s];
Y1 = [0; s; 1; s; 0; -s; -1; -s];
plot(X1,Y1,'x')
xlim([-1.2 1.2]);
ylim([-1.2 1.2]);

xline(0)
yline(0)
legend("Simulated Symbols","Ideal 8PSK")
title("Ideal 8PSK vs LFM approximation")
```

A.2 Simulation Graphs

```
% Set up LFM Properties
SamplePeriod = 1/(1.2e9);
BandWidth = 1e9;
CenterFreq = 10e9;
PulseWidth = 1e-6;
K = BandWidth/PulseWidth;
t = 0:SamplePeriod:PulseWidth;

startSNRdB = 0.0;
stopSNRdB = 6.0;
stepSNRdB = .1;
SNRs = startSNRdB:stepSNRdB:stopSNRdB;
%% PSK4UpDown
s0 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0);
s1 = cos(2*pi*CenterFreq*t - pi*K*t.^2 + 0);
s3 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.5*pi);
```

```
s2 = cos(2*pi*CenterFreq*t - pi*K*t.^2 + 0.5*pi);
s6 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1*pi);
s7 = cos(2*pi*CenterFreq*t - pi*K*t.^2 + 1*pi);
s5 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.5*pi);
s4 = cos(2*pi*CenterFreq*t - pi*K*t.^2 + 1.5*pi);

s = [s0; s1; s2; s3; s4; s5; s6; s7];
bitErrorRate4Psk = CalculateBitError(s, SNRs);



%% 8PSK
s0 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0);
s1 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.25*pi);
s3 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.5*pi);
s2 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 0.75*pi);
s6 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1*pi);
s7 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.25*pi);
s5 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.5*pi);
s4 = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 1.75*pi);

s = [s0; s1; s2; s3; s4; s5; s6; s7];
bitErrorRate8Psk = CalculateBitError(s, SNRs);

%% Theoretical
PSK4UpDownError = zeros(size(SNRs));
PSK8Error = zeros(size(SNRs));
j = 0;
s0s0 = 1;
for SNRdB = SNRs
    j = j +1;
    N0 = s0s0/(10^(SNRdB/10));
    PSK4UpDownError(j) = PSK4UpDownBitErrorFun(s0s0, N0);
```

```
    PSK8Error(j) = PSK8BitErrorFun(s0s0, N0);
end

%% Plotting
clf;
figure(1)
semilogy(SNRs, bitErrorRate8Psk)
hold on
semilogy(SNRs, PSK8Error)
title("Sign Modulation Vs Theoretical BPSK")
ylabel("Probability of BitError")
xlabel("SNR(dB)")
legend("Simulated", "8BPSK")

figure(2)
semilogy(SNRs, bitErrorRate4Psk)
hold on
semilogy(SNRs, PSK4UpDownError)
title("4PSK with K Sign Modulation Simulated Vs Theoretical")
ylabel("Probability of BitError")
xlabel("SNR(dB)")
legend("Simulated", "Theoretical")

figure(3)
semilogy(SNRs, PSK8Error)
hold on
semilogy(SNRs, bitErrorRate4Psk)
title("4PSK with K Sign Modulation Vs 8PSK")
ylabel("Probability of BitError")
xlabel("SNR(dB)")
legend("8PSK", "4PSK with K Sign Modulation")
```

```
function E = PSK4UpDownBitErrorFun(Ebit, N0)
E = 1/3*(qfunc(sqrt(6*Ebit/N0))+6*qfunc(sqrt(3*Ebit/N0)));
end

function E = PSK8BitErrorFun(Ebit, N0)
E = 2/3*qfunc(sqrt(Ebit/N0 * 6 * sin(pi/8)^2));
end

function out = qfunc(x)
out=erfc(x/sqrt(2))/2;
end
```

A.5 Chapter 5 Graphs

A.1 Orthogonal Time Delay Graphs

```
%% Orthogonal time delay
startSNRdB = 0.0;
stopSNRdB = 6.0;
stepSNRdB = 0.2;
SNRs = startSNRdB:stepSNRdB:stopSNRdB;

% To speed up computation time we can just use orthogonal binary signals
% This is equivilent to whatever orthogonal signals we could use
%% Simulated
On = [1,1];
Off = [0,0];
s0 = [On,Off,Off,Off,Off,Off,Off,Off];
s1 = [Off,On,Off,Off,Off,Off,Off,Off];
s2 = [Off,Off,On,Off,Off,Off,Off,Off];
s3 = [Off,Off,Off,On,Off,Off,Off,Off];
s4 = [Off,Off,Off,Off,On,Off,Off,Off];
```

```
s5 = [Off,Off,Off,Off,Off,On,Off,Off];
s6 = [Off,Off,Off,Off,Off,Off,On,Off];
s7 = [Off,Off,Off,Off,Off,Off,Off,On];
s = [s0;s1;s2;s3;s4;s5;s6;s7];


bitErrorRate = CalculateBitError(s, SNRs);
%% Theoretical
Error2 = zeros(size(SNRs));
Error4 = zeros(size(SNRs));
Error8 = zeros(size(SNRs));
Error16 = zeros(size(SNRs));
Error32 = zeros(size(SNRs));
Error64 = zeros(size(SNRs));

j = 0;
s0s0 = 1;
for SNRdB = SNRs
   j = j +1;
   N0 = s0s0/(10^(SNRdB/10));
   Error2(j) = OrthogonalTimeDelay(s0s0, N0, 2);
   Error4(j) = OrthogonalTimeDelay(s0s0, N0, 4);
   Error8(j) = OrthogonalTimeDelay(s0s0, N0, 8);
   Error16(j) = OrthogonalTimeDelay(s0s0, N0, 16);
   Error32(j) = OrthogonalTimeDelay(s0s0, N0, 32);
   Error64(j) = OrthogonalTimeDelay(s0s0, N0, 64);
end

%% Plotting
figure(1)
semilogy(SNRs, bitErrorRate)
hold on
```

```
semilogy(SNRs, Error8)
legend("Simulated", "Theoretical")
title("8 Piece Orthogonal Time Delay Simulated vs Theoretical")
ylabel("Probability of BitError")
xlabel("SNR(dB)")

figure(2)
semilogy(SNRs, Error2)
hold on
semilogy(SNRs, Error4)
semilogy(SNRs, Error8)
semilogy(SNRs, Error16)
semilogy(SNRs, Error32)
semilogy(SNRs, Error64)
legend("2 Symbols", "4 Symbols", "8 Symbols", "16 Symbols", "32 Symbols","64
    Symbols")
title("Theoretical Orthogonal Time Delay of Different Sizes")
ylabel("Probability of BitError")
xlabel("SNR(dB)")

function E = OrthogonalTimeDelay(Eb, N0, M)
E = (M-1)/log2(M)*qfunc(sqrt(log2(M)*Eb/N0));
end
```

A.2 Non-Orthogonal Time Delay Simulation Graphs

```
    %% Non Orthogonal time delay
startSNRdB = 0.0;
stopSNRdB = 6.0;
stepSNRdB = 0.2;
SNRs = startSNRdB:stepSNRdB:stopSNRdB;
```

```
SamplePeriod = 1/(1.2e9);
BandWidth = 1e9;
CenterFreq = 10e9;
PulseWidth = 1e-6;
K = BandWidth/PulseWidth;
t = 0:SamplePeriod:PulseWidth;
%% Simulated All Modulation
Norm = cos(2*pi*CenterFreq*t + pi*K*t.^2);
Waveforms = zeros(8, size(Norm,2));
Waveforms(1, :) = Norm;
Waveforms(2, :) = cos(2*pi*CenterFreq*t + pi*K*t.^2 + pi/2);
Waveforms(3, :) = cos(2*pi*CenterFreq*t + pi*K*t.^2 + pi);
Waveforms(4, :) = cos(2*pi*CenterFreq*t + pi*K*t.^2 + 3*pi/2);
Waveforms(5, :) = cos(2*pi*CenterFreq*t - pi*K*t.^2);
Waveforms(6, :) = cos(2*pi*CenterFreq*t - pi*K*t.^2 + pi/2);
Waveforms(7, :) = cos(2*pi*CenterFreq*t - pi*K*t.^2 + pi);
Waveforms(8, :) = cos(2*pi*CenterFreq*t - pi*K*t.^2 + 3*pi/2);
Off = zeros(1,ceil(size(Norm,2)/5));

s = zeros(32, size(Norm,2) + size(Off,2)*3);

for i = 1:8
    s(1 + (i-1)*4, :) = [Waveforms(i, :), Off, Off, Off];
    s(2 + (i-1)*4, :) = [Off, Waveforms(i, :), Off, Off];
    s(3 + (i-1)*4, :) = [Off, Off, Waveforms(i, :), Off];
    s(4 + (i-1)*4, :) = [Off, Off, Off, Waveforms(i, :)];
end

bitErrorRateAllMod = CalculateBitError(s, SNRs);



%% Simulated Only Time Delay Modulation
```

```
On = cos(2*pi*CenterFreq*t + pi*K*t.^2);
Off = zeros(1,ceil(size(On,2)/5));
s0 = [On,Off,Off,Off,Off,Off,Off,Off];
s1 = [Off,On,Off,Off,Off,Off,Off,Off];
s2 = [Off,Off,On,Off,Off,Off,Off,Off];
s3 = [Off,Off,Off,On,Off,Off,Off,Off];
s4 = [Off,Off,Off,Off,On,Off,Off,Off];
s5 = [Off,Off,Off,Off,Off,On,Off,Off];
s6 = [Off,Off,Off,Off,Off,Off,On,Off];
s7 = [Off,Off,Off,Off,Off,Off,Off,On];
s = [s0;s1;s2;s3;s4;s5;s6;s7];

bitErrorRateOnlyTimeDelay = CalculateBitError(s, SNRs);
%% Theoretical
TheoError = zeros(size(SNRs));
Error8 = zeros(size(SNRs));
TheoErrorCorrected = zeros(size(SNRs));
j = 0;
s0s0 = 1;
for SNRdB = SNRs
    j = j +1;
    N0 = s0s0/(10^(SNRdB/10));
    TheoError(j) = TimeDelayAllMod32(s0s0, N0);
    TheoErrorCorrected(j) = TimeDelayAllMod32Corrected(s0s0, N0);
    Error8(j) = NonOrthogonalTimeDelay(s0s0, N0, 8);
end

clf;
figure(1)
semilogy(SNRs, bitErrorRateAllMod)
hold on
semilogy(SNRs, TheoError)
```

```
legend("Simulated", "Theoretical")
title("Non-Orthogonal Time Delay W/ Phase and Sign Change")
ylabel("Probability of BitError")
xlabel("SNR(dB)")

figure(2)
semilogy(SNRs, bitErrorRateAllMod)
hold on
semilogy(SNRs, TheoErrorCorrected)
legend("Simulated", "Theoretical")
title("Corrected Non-Orthogonal Time Delay W/ Phase and Sign Change")
ylabel("Probability of BitError")
xlabel("SNR(dB)")

figure(3)
semilogy(SNRs, bitErrorRateOnlyTimeDelay)
hold on
semilogy(SNRs, Error8)
legend("Simulated", "Theoretical")
title("8 Piece Non-Orthogonal Time Delay Simulated vs Theoretical")
ylabel("Probability of BitError")
xlabel("SNR(dB)")



function E = TimeDelayAllMod32(Ebit, N0)
E = 1/5*(qfunc(sqrt(10*Ebit/N0))+30*qfunc(sqrt(5*Ebit/N0)));
end

function E = TimeDelayAllMod32Corrected(Ebit, N0)
E = 1/5*(qfunc(sqrt(10*Ebit/N0))+24*qfunc(sqrt((1.4/sqrt(2))*5*Ebit/N0)) + ...
    6*qfunc(sqrt((1.26/sqrt(2))*5*Ebit/N0)));
end
```

```
function E = NonOrthogonalTimeDelay(Eb, N0, M)
E = (M-1)/log2(M)*qfunc(sqrt(log2(M)*Eb/N0));
end
```

CURRICULUM VITAE

Jarren T. Worthen

Published Conference Papers

- Adaptive Regression Trees for Nonlinear Adaptive Filtering, Jarren T. Worthen, Todd K. Moon and Jacob H. Gunther, in *Asilomar Conference on Signals, Systems, and Computers*, 2022.

Education

- BS in Electrical Engineering, Honors Curriculum, Summa Cum Laude, Utah State University, Logan, Utah.