

UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

SOMETHING TO DO WITH RADAR WHO CARES

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By
RUSSELL KENNEY
Norman, Oklahoma
2020

SOMETHING TO DO WITH RADAR WHO CARES

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Jay McDaniel, Chair

Dr. Hjalti Sigmarsson

Dr. Mark Yeary

© Copyright by RUSSELL KENNEY 2020

All Rights Reserved.

Acknowledgments

I would like to thank all the FB meme pages that made my degree doable when studying made me want to end my life.

Table of Contents

Acknowledgment

List of Figures vi

Abstract xii

1 Introduction 1

1.1 Thesis Outline 2

2 Background 3

2.1 Fundamentals of Pulse-Doppler Radar 3

2.1.1 Pulse Compression and Doppler Processing 6

2.1.2 Range Ambiguities Due to Digital Incoherence 11

2.2 Digital Radar Architecture 13

2.2.1 Digital Waveform Synthesis 13

2.2.2 Digitization of Received Signal 16

3 Initial System Design 19

3.1 Theory of Operation 19

3.2 The RF Front-End 20

3.3 The Digital Back-End 22

3.3.1 Evaluation Software 23

4	Digital System Configuration	27
4.1	DDS Configuration	27
4.1.1	LFM Pulse Generation	28
5	Pulse-to-Pulse Incoherence	31
6	Start-Up Incoherence and Loopback Calibration	32
7	Sensing Results	33
8	Conclusion and Future Work	34

List of Figures

2.1	A simple pulse train	4
2.2	A simple LFM pulse	6
2.3	Matched filter output	8
2.4	Matched filter output with noise	8
2.5	A simple Range Map	9
2.6	A simple Range-Doppler Map	10
2.7	A Range Map with incoherent pulses	12
2.8	A Range-Doppler Map with incoherent pulses	12
2.9	The ideal DAC output before and after analog filtering	15
3.1	The original radar block diagram	21
3.2	The original radar hardware	21
3.3	The original digital back-end block diagram	24
3.4	The HSDC Pro GUI	25

Abstract

This is the abstract.

Chapter 1

Introduction

Radar has found application in a number of avenues in modern life, including meteorology, national defense, terrain mapping, and air traffic control, to name a few. In all of these applications, the radar system is often designed to be a pulse-Doppler system, which typically increases the complexity of the digital electronics required to operate the system when compared to continuous-wave (CW) and frequency-modulated continuous-wave (FMCW) radar systems [1]. This added complexity often leads to added size, weight, power, and cost (SWaP-C) of the system. For example, some radar systems use a pre-made digital chassis, such as an NI-DAQ, to synthesize the transmitted waveform and to digitize and process the received waveform [2]. While such a system conveniently abstracts away the design of the digital transceiver, they are also typically large, heavy, and prohibitively expensive. Although the increased SWaP-C is tolerable for one-use ground-based radar platforms, these systems are not easily reproducible for bulk-production and cannot be placed on lightweight aerial platforms such as small UAVs or quad/octocopters.

One solution to this problem is to use small and cheap commercial-off-the-shelf (COTS) digital parts to construct the digital back-end of the radar. The use of COTS parts allows for a low-cost and easily replicated digital system that can be integrated

into a small form-factor PCB that satisfies the low SWaP-C requirement. The primary challenge of using COTS parts to build a digital radar is that typically the digital parts are not designed to specifically meet the strict timing requirements of a pulse-Doppler radar system. If care is not taken in the design and integration of the digital parts, particularly with regard to clock synchronization between the devices, the radar system will not operate as intended due to digital timing errors, which lead to unacceptable inaccuracies in the range measurements gathered by the radar. To further complicate the design, larger analog bandwidths and shorter pulse widths are desirable to give a more precise range resolution and a smaller blind range [3], which requires an increase in the sampling frequency of the analog-to-digital converter (ADC) and the reconstruction frequency of the digital-to-analog converter (DAC). As these frequencies increase, it becomes more difficult to synchronize the digital clocks operating on each device.

This thesis is concerned with the digital design of a high sampling wideband pulse-Doppler radar system for use in a synthetic aperture radar (SAR) system. The system is desired to be small and light enough that it can be placed on a light octo-copter, and must be robust enough to obtain a range resolution of 1 m with a look angle of 30° .

1.1 Thesis Outline

Chapter 2

Background

To fully grasp the scope of the work demonstrated in this thesis, the basics of digital pulse-Doppler radar systems must be understood. Additionally, the general operation and original design of the radar system used in this project should be understood to provide clarity to the system modifications presented in this thesis. As such, this section seeks to provide a foundational understanding of pulse-Doppler radar operation, an overview of digital radar architectures, and an explanation of both the RF and digital characteristics of the radar system used in this thesis.

2.1 Fundamentals of Pulse-Doppler Radar

A pulse-Doppler radar system operates by transmitting a train of short pulses with a low duty cycle and then capturing the pulses as they return to the radar after reflecting off objects (or *scatterers*). The pulses have a pulse width of T_p and are transmitted at a precise frequency called the *pulse repetition frequency*, or PRF. The period of the PRF, or the time between successive pulses, is known as the *pulse repetition interval*, or PRI. Typically the pulse width is much shorter than the PRI. A short pulse train consisting of five pulses with a PRI of $400\ \mu\text{s}$ and a pulse width of $50\ \mu\text{s}$ is shown in Figure 2.1.

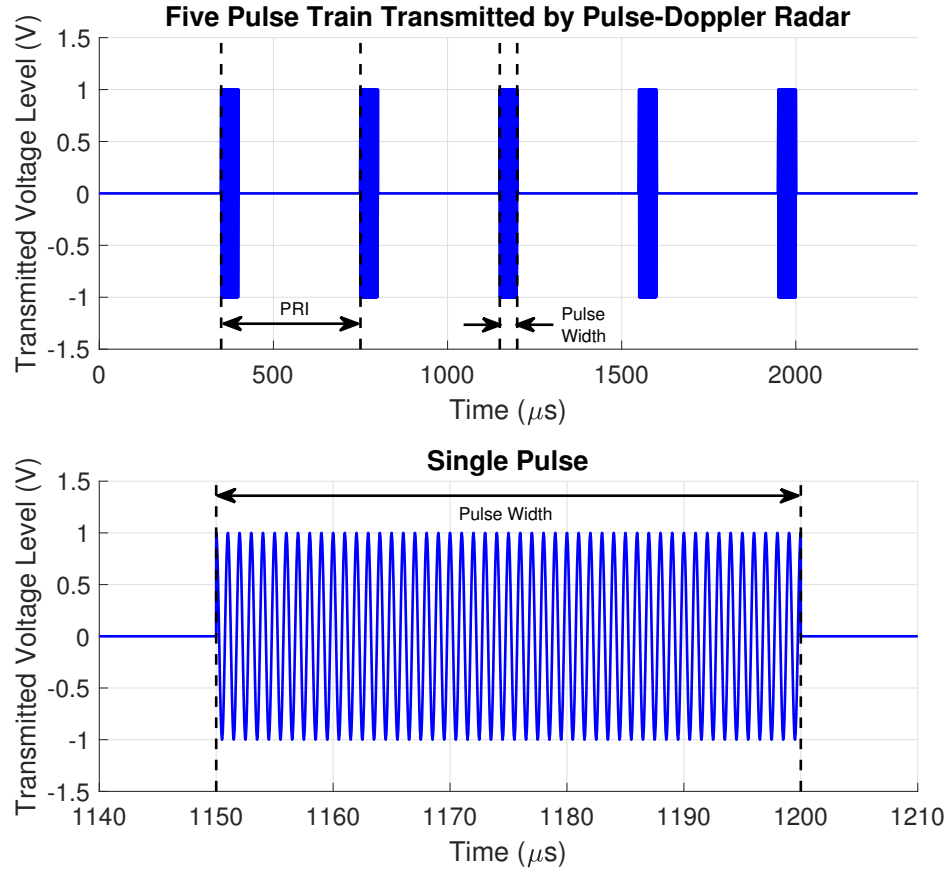


Figure 2.1: A simple pulse train

The radar is able to measure both the distance between the scatterers and the radar (R) and relative radial velocity of the scatterers with respect to the radar (v_d). The range R can be extracted from the time delay between the time that a pulse is transmitted and the time the pulse is received, denoted by τ_p . The relative radial velocity of the target can be extracted by determining the Doppler shift of the received waveform when compared to the original waveform, denoted by F_D . The range is calculated from τ_p by

$$R = \frac{c\tau_p}{2} \quad (2.1)$$

where c is the speed of light. This equation accounts for the two-way propagation

time of the radar pulse from the radar to the scatterer and then back.

To determine the radial velocity of a moving scatterer, the frequency of the Doppler shift caused by the relative motion of the scatterer can be extracted over the course of multiple pulses called a *coherent processing interval* (CPI). The Doppler shift frequency F_D is calculated by

$$F_D = \frac{2v_s}{c - v_s} F_c \approx \frac{2v_s}{c} F_c = \frac{2v_s}{\lambda_c} \quad (2.2)$$

where v_s is the radial velocity of the scatterer and F_c and λ_c are the frequency and wavelength of the carrier [3].

Given two scatterers within the observable scene of the radar, the range to each target cannot be resolved with infinite precision. Instead, the radar can only resolve targets into discrete “range bins,” where the size of each range bin is denoted as the *range resolution* ΔR of the radar. The range resolution of the radar is given by

$$\Delta R = \frac{c}{2\beta} \quad (2.3)$$

where β is the instantaneous bandwidth of the transmitted pulse [3]. For the most simple pulse waveform, each pulse is a single frequency tone. For this case, β is approximately equal to $\frac{1}{T_p}$. It is typical for a radar system to implement some sort of modulation to the pulse to increase the bandwidth without decreasing the pulse width. A frequently implemented pulse waveform is the *linear frequency modulated* (LFM) pulse. The LFM waveform, also commonly called a “chirp,” increases linearly in frequency from f_1 to f_2 over the course of the pulse width, as shown in Figure 2.2. At complex baseband, f_1 is $-\frac{\beta}{2}$ and f_2 is $\frac{\beta}{2}$. The complex

envelope of such a waveform is given by

$$x(t) = a(t) \exp \left[j2\pi \left(-\frac{\beta}{2} \right) t + j\pi \frac{\beta}{T_p} t^2 \right] \quad 0 \leq t \leq T_p \quad (2.4)$$

where $a(t)$ is some amplitude scale factor introduced by the RF front-end and the bandwidth of the waveform is given by

$$\beta = f_2 - f_1 \quad (2.5)$$

which is independent of the pulse width. Thus, a radar can obtain an arbitrarily fine resolution by increasing the band of frequencies covered by each LFM pulse.

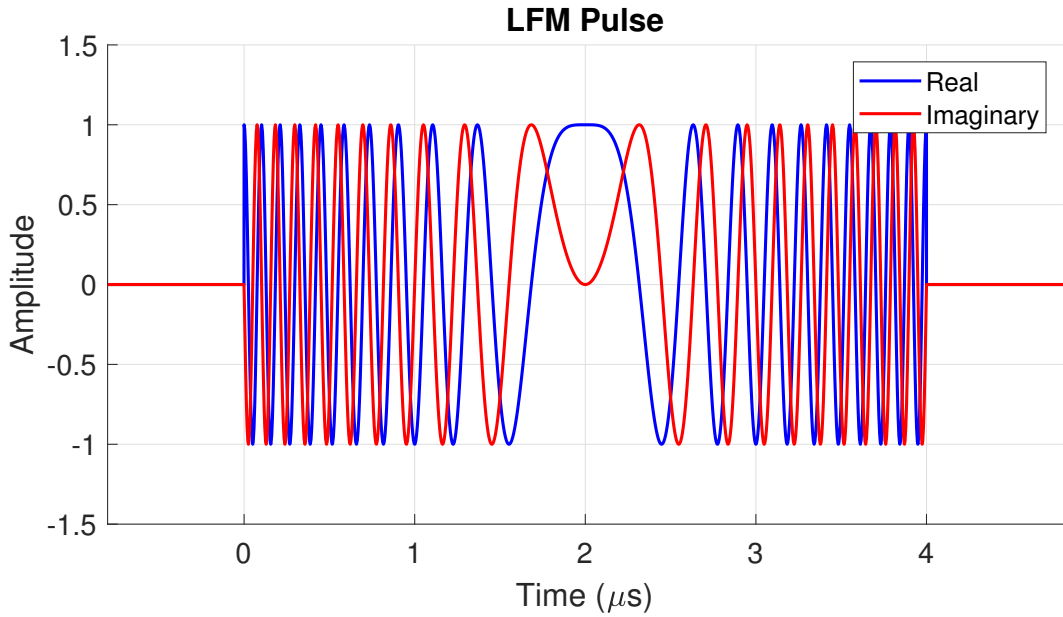


Figure 2.2: A simple LFM pulse

2.1.1 Pulse Compression and Doppler Processing

When a train of pulses is transmitted, the received signal may be completely obscured by noise. For targets a considerable distance from the radar, the received

reflection power is potentially much lower than the thermal noise power introduced by the system itself, making the discovery of target reflections within the received signal difficult. The balance of received signal power compared to the noise power is referred to as the *signal-to-noise ratio* (SNR). In order to reliably extract reflection signals in the noise received signal, it is desirable to implement a digital filter as a processing step that maximizes the SNR of the received signal.

The filter that maximize the SNR is known as the *matched filter*. It is called the matched filter because the impulse response is matched to the signal being transmitted by the radar. Say that a radar transmits an LFM pulse whose complex envelope is given by (2.4). The matched filter $h(t)$ used to maximize the SNR of the received signal will be the time-reversed, complex conjugate of the signal. That is,

$$\begin{aligned} h(t) &= x^*(T_p - t) \\ &= a^*(T_p - t) \exp \left[j2\pi \left(-\frac{\beta}{2} \right) (T_p - t) + j\pi \frac{\beta}{T_p} (T_p - t)^2 \right] \quad 0 \leq t \leq T_p \end{aligned} \quad (2.6)$$

The convolution of $x(t)$ with $h(t)$ is functionally equivalent to a mathematical cross-correlation.

The matched filter works by compressing the energy of each pulse into a smaller span of time so that a “peak” will appear, indicating the presence of the desired waveform at a particular time. For this reason, matched filtering is sometimes referred to as *pulse compression*. The peak appears at time $\tau_p + T_p$, which is the sum of the propagation time of the pulse and the delay of the causal matched filter. This peak has a 3-dB width approximately equal to $\frac{1}{\beta}$ in units of time, which can be converted to range using (2.1). The resulting expression is equivalent to the range resolution of the radar given in (2.3). The matched filter output of an LFM pulse

with a bandwidth of 20 MHz is shown in Figure 2.3.

As stated above, the matched filter allows a signal to be extracted from a very noisy data capture. Figure 2.4 shows the matched filter output of a received LFM signal corrupted with additive white Gaussian noise (AWGN) when the noise power of the un-processed signal is higher than the signal power.

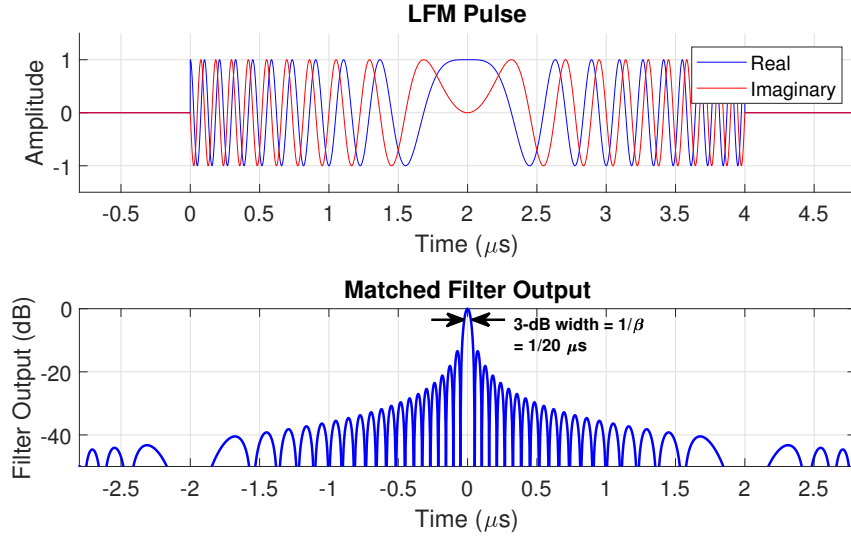


Figure 2.3: Matched filter output

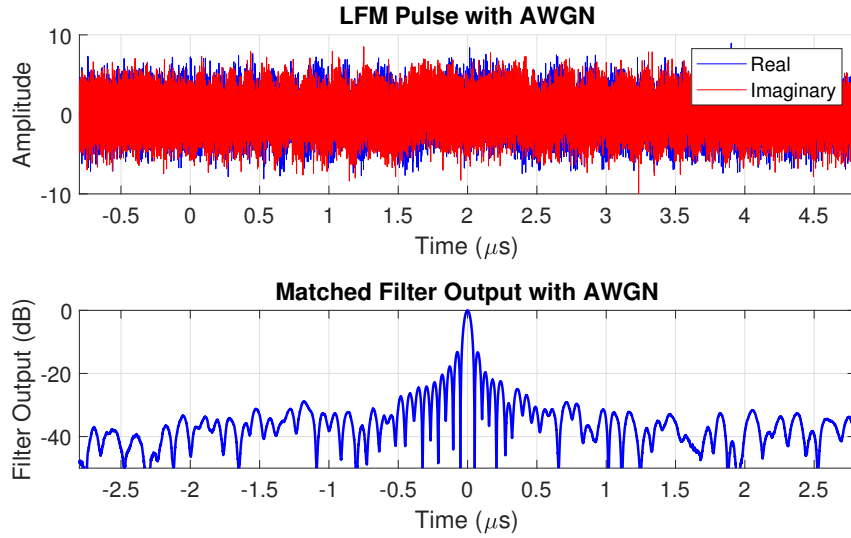


Figure 2.4: Matched filter output with noise

When a pulse train similar to the one shown in Figure 2.1 is transmitted, the resulting data can be organized into two dimensions: one dimension in terms of ADC samples for each pulse (called “fast-time”) and the other dimension in terms of pulses (called “slow-time”). The pulse compression operation, performed in the fast-time dimension, will generate peaks at the target range at each pulse. If the target is not moving or is moving slowly, or if the CPI is very short, then the pulse compression peaks will occur at the same range for each pulse. An image of the pulse-compressed reflection intensity as a function of range and pulse number yields a “range map” of the radar scene. An example of a range map of a target moving toward the radar at 5 m/s at a range of 100 m is shown in Figure 2.5.

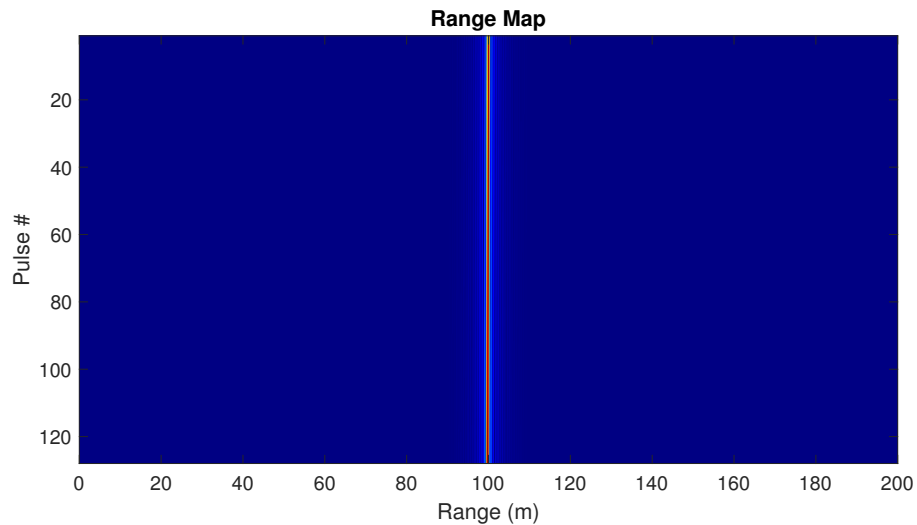


Figure 2.5: A simple Range Map

On top of sensing target range, pulse-Doppler radar systems are also useful for sensing target velocity. The primary vehicle for measuring velocity is the Doppler shift frequency of the received waveform that is caused by the relative radial velocity between the radar and the target. Generally, the Doppler shift frequency is on the order of kHz. Because pulses are on the order of microseconds long, the

Doppler shift will typically not cause a substantial phase rotation of the received waveform over the course of a single pulse, meaning that it is not possible to extract the Doppler shift from a single pulse. To observe the Doppler shift over a sufficiently long time frame, a pulse train is transmitted to effectively “sample” the Doppler frequency at the PRF. Therefore, each pulse can be thought of as a digital sample of the Doppler shift. It follows then, that the Doppler shift can be extracted from the received pulse train by taking the Fourier Transform in the slow-time dimension. Taking the fast Fourier Transform (FFT) in the slow-time dimension at every range value yields a peak at located at the range and Doppler shift frequency of scatterers in the scene. Solving (2.2) for v_s shows that the Doppler shift frequency can be directly correlated to velocity. The 2D intensity image of return energy as a function of range and velocity (or Doppler shift) is called a “range-Doppler map.” The range-Doppler map generated by performing Doppler processing on the range map shown in Figure 2.5 is shown in Figure 2.6. Notice the single bright dot located at 100 m in range and 5 m/s, indicating the presence of a scatterer.

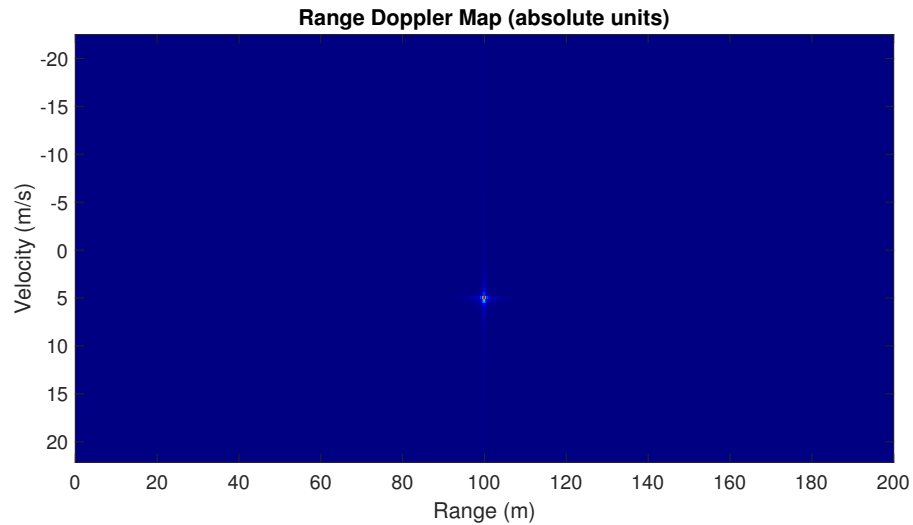


Figure 2.6: A simple Range-Doppler Map

2.1.2 Range Ambiguities Due to Digital Incoherence

Due to practical delays introduced by the system itself, the radar cannot measure the waveform propagation time τ_p directly. Instead, the actual delay measured by the radar τ_m is the sum of three components: the actual propagation time of the waveform from the radar to the target and back (τ_p), the delay of the waveform as it is conditioned by the RF transmitter and receiver (τ_{RF}), and the delay introduced by the digital back-end as the waveform is synthesized and digitally captured (τ_{dig}). Ideally, both τ_{RF} and τ_{dig} are known constants and can be easily accounted for in processing. While this is typically true of τ_{RF} , τ_{dig} may be random in nature, introducing potentially large errors in the computation of R . This is especially problematic if τ_{dig} varies from pulse to pulse.

To demonstrate the negative impact that a random τ_{dig} can have on the range estimation given by a radar system, suppose that τ_{dig} is a random variable distributed uniformly in the interval $(-\tau_r, \tau_r)$. Assuming that τ_{RF} is known and removed from the computation, the transformation of τ_m to R by (2.1) gives

$$R_m = \frac{c\tau_p}{2} + \frac{c\tau_{\text{dig}}}{2} = R + R_{\text{dig}} \quad (2.7)$$

where R_m is the measured range, R is the actual range, and R_{dig} is the range error introduced by varying τ_{dig} . If τ_{dig} varies randomly from pulse to pulse, there is no way to compensate for it, and the range error of R_m will vary within the interval $(-\frac{c\tau_r}{2}, \frac{c\tau_r}{2})$. If τ_r is even as large as 50 ns, the measured range for a stationary scatterer can vary by as much as 15 m, which is an unacceptable ambiguity for any radar with moderate range resolution. When the incoherent pulses are compressed to form a range map, as in Figure 2.7, the problem becomes evident, as the true range to the target clearly cannot be seen. After Doppler processing, the incoher-

ence results in essentially a large amplitude noise signal, as shown in Figure 2.8.

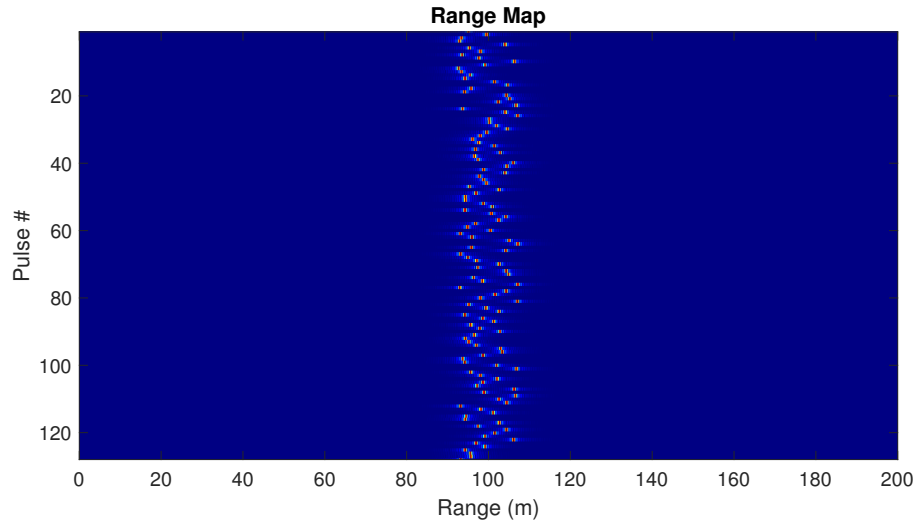


Figure 2.7: A Range Map with incoherent pulses

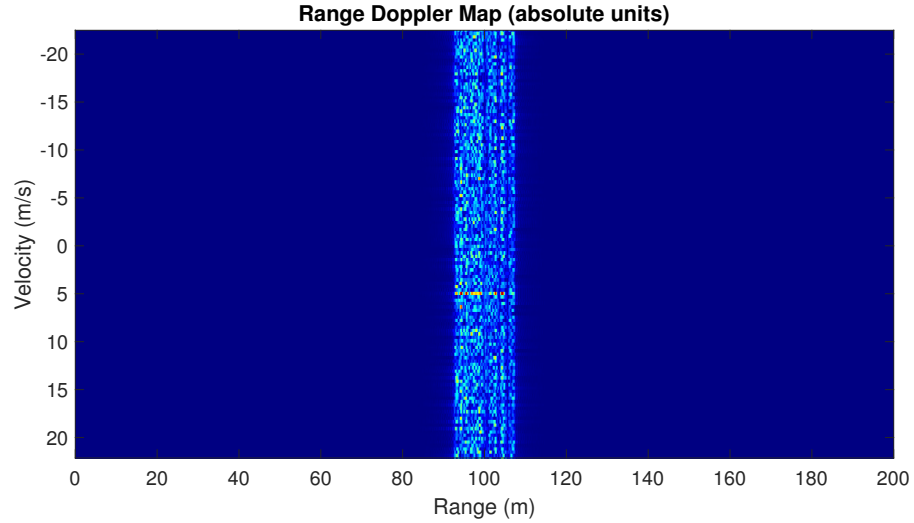


Figure 2.8: A Range-Doppler Map with incoherent pulses

2.2 Digital Radar Architecture

In a typical digital radar system, the radar can be broken up into two main sections: the RF front-end, and the digital back-end. The digital back-end synthesizes a digital waveform, often as a complex baseband waveform. This digital waveform is then reconstructed as an analog waveform and fed to the RF front-end. The RF front-end conditions the signal for transmission, often through analog filtering, amplification, and mixing with a high-frequency local oscillator (LO) to convert the waveform to the carrier frequency. The waveform is then coupled into free space by an antenna. When the waveform returns, an antenna captures the propagating wave and feeds it to the receive section of the RF front-end. The RF front-end conditions the signal for digital storage by amplification, removal of out-of-band interferers through analog filtering, and mixing back down to an intermediate frequency (IF) using the same LO as the transmitter to maintain coherence. The digital back-end digitizes the conditioned received waveform and stores it, and the digitized data is then processed to detect targets, extract range and velocity information, form images, or other processing steps. Depending on the application, this processing can be done in real-time or after the data has been collected in a post-processing step. The focus of this section is on the components and operation of a typical digital back-end.

2.2.1 Digital Waveform Synthesis

As alluded to above, the digital back-end of a radar has two primary roles: synthesizing the transmitted waveform and digitizing and processing the received waveform. To synthesize the waveform, the digital back-end must both digitally define the waveform to be transmitted and then convert the waveform from a digital

sequence to an analog signal.

Before a waveform can be properly converted from digital to analog, it must be accurately defined as a digital signal. An increasingly common method for doing this is called *direct digital synthesis*, which is performed by a device called a *direct digital synthesizer* (DDS). A DDS enables the generation of complex waveforms without an external processor defining each point in the digital waveform sequence [4]. Instead, the DDS can synthesize waveforms with only preliminary programming through SPI and a sufficiently high frequency clock signal.

The DDS synthesizes waveforms using a numerically controlled oscillator (NCO). The NCO consists of two components: a phase accumulator, and a phase-to-amplitude converter. The phase accumulator is a register of N bits whose value corresponds to the current phase of the NCO such that each of the 2^N possible values are evenly spaced in the range $[0, 2\pi)$. The phase accumulator is incremented by some amount θ_{inc} every clock cycle of the DDS, where θ_{inc} is also defined by an N bit register mapping it to the range $[0, 2\pi)$. Given a DDS frequency of f_{DDS} , the output frequency of the DDS is

$$f_{\text{out}} = \frac{\theta_{\text{inc}}}{2\pi} f_{\text{DDS}} \quad (2.8)$$

Note that in order to satisfy the Nyquist Theorem, θ_{inc} should be less than π ; otherwise, the DDS will synthesize an aliased frequency.

At every iteration of the DDS clock, the value of the phase accumulator is input to the phase-to-amplitude converter. The phase-to-amplitude converter determines the amplitude value of a sine or cosine wave for the given phase value, often using a lookup table of amplitude values.

The DDS conveniently allows for quick changes in output frequency simply by modifying the value of θ_{inc} . Some DDS products implement a digital ramp

feature, which allows θ_{inc} to be swept upward or downward quickly, enabling the generation of the LFM waveform. Additionally, in some DDS products, the phase of the phase accumulator and the amplitude of the output can be modified, allowing for modulation of the amplitude and the phase of the output waveform in addition to the frequency.

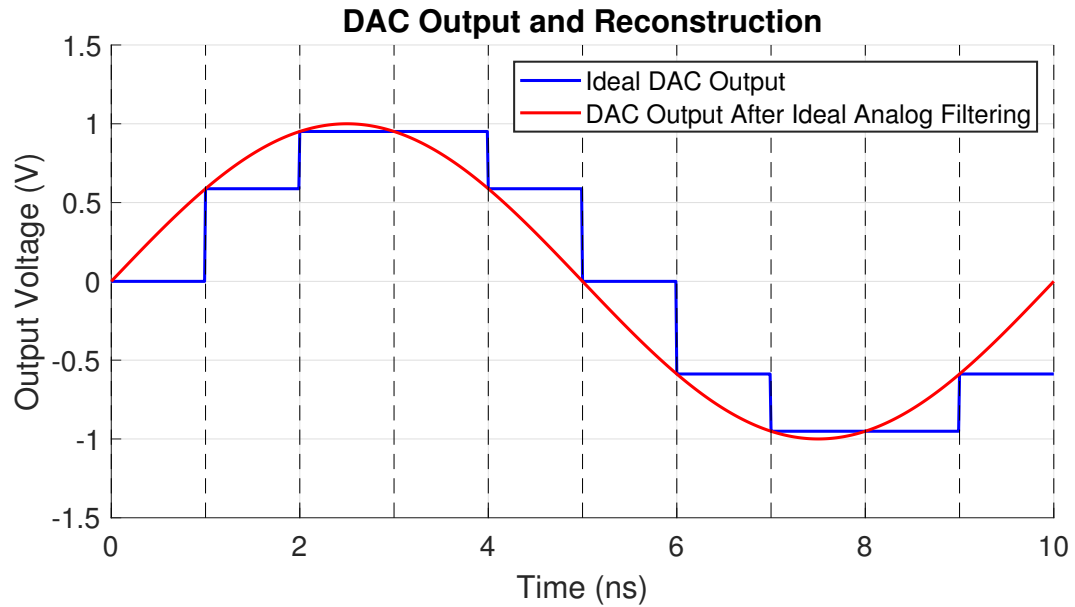


Figure 2.9: The ideal DAC output before and after analog filtering

The conversion from digital to analog requires the use of a digital-to-analog converter (DAC). Most DDS products include a DAC integrated within the same IC containing the DDS core described above. As such, the DAC will often use the DDS clock to time the output of each sample. Each sample of the waveform output by the DAC does not transition smoothly into the next. Instead, each sample is held constant at its value for the entire sample period before transitioning to the next sample. The resulting output waveform has a stairstep appearance to it, resulting from the unfiltered harmonics of the digital signal. An analog filter at the output of the DAC removes the harmonics and smooths the waveform into the desired

sinusoid, as shown in Figure 2.9.

2.2.2 Digitization of Received Signal

In addition to synthesizing the transmit waveform, the radar digital back-end must also convert the received signal back into a digital signal to be stored and processed. Nominally, this step requires only an analog-to-digital converter (ADC) with a sampling frequency at least two times greater than the bandwidth β of the radar signal. However, if the sampling frequency and bit resolution of the ADC are high enough, the ADC will generate a massive amount of data every second that must be reliably stored in memory. For instance, given a signal bandwidth β of 100 MHz sampled by a 10-bit ADC at a sampling frequency of 200 MHz, the data channel communicating each sample of the ADC into memory must be capable of handling 2 Gbps at a minimum. To accommodate larger signal bandwidths and better receiver dynamic range, the number of bits per sample and the sampling frequency can become very large, which makes the high-speed capture of the ADC data difficult.

In order to handle these high data rates, radar systems often connect the ADC to a field-programmable gate array (FPGA) with some high speed data transfer protocol. The FPGA is then configured by the user to process and store the data, or transfer it to a computer for processing. Typically, the ADC will begin sampling as soon as it is turned on, so the FPGA must be programmed to decide when to store the samples generated by the ADC in memory. This may be accomplished by applying some internal or external trigger command to the FPGA to signal that data should be captured from the ADC. The trigger command issued to the FPGA often must be issued synchronously with the clock of the high-speed communication

protocol. Therefore, the configuration of the high-speed communication protocol between the FPGA and the ADC has a direct impact on the digital coherence of the radar system, potentially leading to increased variance in τ_{dig} and a degraded range resolution as discussed in Section 2.1.2.

The JESD204B Protocol

One popular communication protocol between the FPGA and ADC is the JEDEC JESD204B protocol [5]. JESD204B utilizes differentially-fed serial lanes operating at speeds up to 12.5 Gbps to communicate data between devices. The serial lanes are much easier to route on a PCB than a parallel bus, and JESD204B encodes data using an 8b/10b scheme, allowing clock recovery from the data and eliminating the need for a separate data clock line.

To allow for data communication without a clock line, the interface uses a low-speed signal called the SYSREF to set up and align an internal clock in each device called the Local Multiframe Clock (LMFC). The LMFC times the transmission and receipt of large groups of samples called multiframe. Each multiframe consists of K frames of samples, and each frame consists of F octets (bytes) worth of sample data per lane. Each sample of data contains N bits per sample and the data is communicated over L different serial lanes. The number of data converters (i.e. ADCs and DACs) taking advantage of the serial link is M , and the number of samples transmitted by each converter in a single frame is S . The bit rate on each lane f_L is generally related to the sampling frequency of the data converter. In many cases, the parameters given above are set for a particular data communication application, with only K being defined by the user.

The frequency of the LMFC f_{LMFC} is then given in [6] by

$$f_{\text{LMFC}} = \frac{f_s}{K \times S \times M} \quad (2.9)$$

To achieve the desired LMFC, the SYSREF signal must be configured to the desired LMFC frequency. The LMFC controls the timing of all groups of samples between the ADC and FPGA in a JESD204B link, and as such, a new data capture must begin on the rising edge of the LMFC. It follows that the trigger signal to the FPGA initiating a data capture must deterministically align with the edge of the LMFC. Otherwise, there will be a non-deterministic latency between the trigger signal being issued and capture taking place, leading to digital incoherence discussed in Section 2.1.2.

Chapter 3

Initial System Design

The work done in this thesis is based on an original radar design developed in [7] at the University of Oklahoma. In this work, an RF analog front-end is designed, and components of a digital back-end are selected. This chapter contains an explanation of the original system's theory of operation, including a brief discussion of the analog front-end, the digital components of the RF transceiver, and the digital control system. With respect to the digital back-end section of this chapter, the focus is on system-level requirements driving the selection of hardware. The specific details of the digital hardware implementation are discussed in further detail in Chapter 5.

3.1 Theory of Operation

The radar is designed to be used in a synthetic aperture radar (SAR) application on board a lightweight aircraft. The radar operates at Ku-band with a carrier frequency of 16.6 GHz. The waveform is an LFM pulse with 300 MHz of bandwidth and a pulse width of $1\ \mu\text{s}$ transmitted with 2 W of power. The bandwidth of the waveform results in a range resolution of 0.5 m at boresight using (2.3), and in a SAR application with a 30° look angle it results in a cross-track resolution of 1 m.

To form an image with this resolution, the radar must capture pulses over a synthetic aperture length of 253.6 m, which is accomplished over the course of 4.5 seconds of flight time by a Piper PA-28-161 Warrior III flying at 57 m/s. In a typical monostatic radar application, the 1 μ s pulse width would result in a 150 meter blind range. However, this radar system utilizes a quasi-monostatic configuration with two antennas with high isolation, allowing for simultaneous-transmit-and-receive (STaR) operation, which eliminates the blind range.

The system operates with a PRF of 3 kHz. This frequency was chosen to absorb the entire Doppler bandwidth of a stationary SAR scene observed at an altitude of 1.5 km with a look-angle of 30° and an aircraft velocity of 57 m/s.

This section is concerned with the original design of the Ku-band radar system. The original radar block diagram is shown in Figure 3.1, and an image of the full radar hardware is shown in 3.2. The theory and design of both the RF front-end and the digital back-end are detailed in the following sections.

3.2 The RF Front-End

The RF front-end accepts the signal from the digital back-end as an LFM pulse with 150 MHz of bandwidth centered at 1 GHz. The LFM pulse is amplified and upconverted by a 2x frequency multiplier to 300 MHz of bandwidth at 2 GHz. The harmonics of this multiplication are then filtered and the signal is mixed up to the carrier frequency of 16.6 GHz by an LO of 14.6 GHz generated by a dielectric resonating oscillator (DRO). The output RF signal is then amplified to a 2 W transmit power and transmitted by the transmit antenna. It should be noted that the 2 W amplifier is not pulsed: as such, the pulsing of the waveform is accomplished digitally rather than at the output of the RF transmitter.

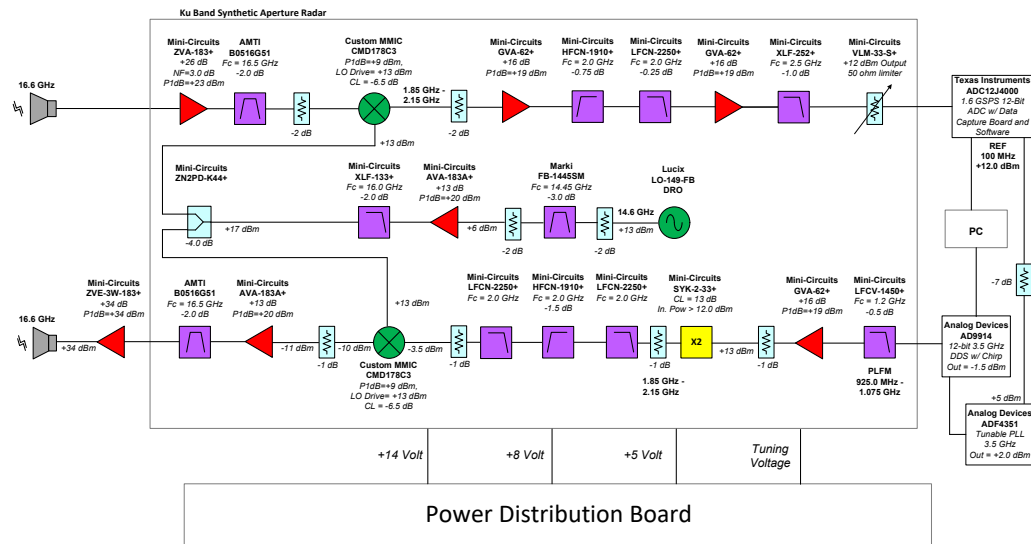


Figure 3.1: The original radar block diagram

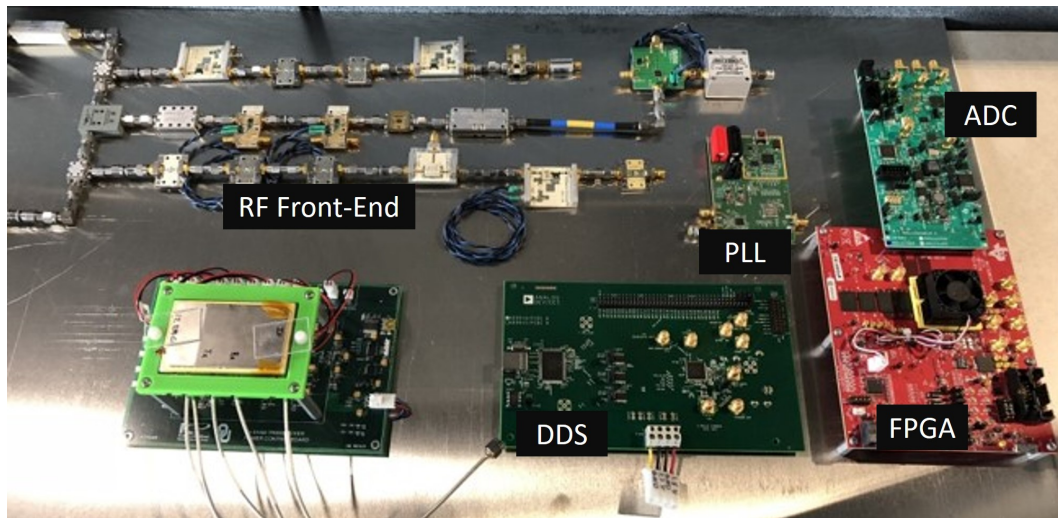


Figure 3.2: The original radar hardware

The receiver accepts the reflected LFM signal with a minimum detectable signal (MDS) at -110 dBm and an input saturation power of -30 dBm. The signal is amplified by a low-noise amplifier (LNA) and filtered, and then mixed back down

to an IF of 2 GHz using the same LO as the transmitter. The IF signal is further amplified and filtered, and the final signal is input directly to the ADC at 2 GHz.

3.3 The Digital Back-End

The digital back-end circuitry is selected with the goal of incorporating and testing state-of-the-art digital transceiver hardware. The digital signal generator is the Analog Devices AD9914 DDS with integrated 12-bit DAC, which has a maximum reconstruction frequency of 3.5 GHz [8]. The receiver subsystem is the Texas Instruments ADC12J4000 ADC, which has a maximum sampling frequency of 4 GSPS [9] with 12-bit sampling resolution. The ADC mates with the Texas Instruments TSW14J56 FPGA capture board through a high pin-count (HPC) FPGA mezzanine card (FMC) connector [10]. The capture board contains an Altera Arria V GZ FPGA and 4 GB of onboard RAM. Data is collected from the ADC by the FPGA using the JESD204B interface through the FMC connector, and is then off-loaded to a PC for post-processing through a USB cable. For simplicity, each digital device is purchased on an evaluation board to allow the immediate development of a system.

The DDS is operated at the maximum frequency of 3.5 GHz. Operating at a higher frequency reduces the harmonic output power of the DDS. The DDS generates the 150 MHz LFM pulse over the course of $1\ \mu\text{s}$, which is then multiplied to 300 MHz of bandwidth in the RF front-end. As stated in Section 3.2, the RF amplifier is not pulsed. As such, the amplitude window of the pulse must be applied by the DDS.

To save FPGA memory during the capture of the waveform, the ADC is operated at a sampling frequency of 1.6 GSPS. This undersamples the 2 GHz waveform

from the IF of the receiver, resulting in the captured signal being an aliased signal at 400 MHz. Because the sampling frequency is still higher than the twice the bandwidth of the signal per the Nyquist Theorem, the full signal spectrum is preserved.

To derive the digital clocks used in the system, a 100 MHz crystal oscillator is multiplied by various phase-locked loops (PLL) to the desired frequencies. The ADC evaluation board utilizes two onboard PLLs to synthesize the sampling clock and the JESD204B clocks: the TRF3765 and the LMK04828 [11, 12, 13]. The precise operation of these devices will be discussed in Chapter 5. The DDS contains an internal PLL that can be used to synthesize a clocking frequency internally - however, it has a maximum clock output of 2.5 GHz, meaning that the 3.5 GHz clock must be synthesized off-chip by an external PLL. The PLL selected for this is the Analog Devices ADF4351 [14]. The 100 MHz clock reference on the ADC evaluation board is tapped off to the ADF4351, which multiplies the signal by x35 to obtain the necessary clocking frequency.

In the initial back-end design, an NI DAQ 6251 was used as a controller platform to generate the PRF signal, simultaneously triggering the DDS to synthesize a single pulse and the FPGA to begin saving samples from the pulse on each cycle of the PRF. The configuration of each device is implemented using a USB interface to each evaluation board. The original digital back-end diagram is shown in Figure 3.3.

3.3.1 Evaluation Software

Because the digital components are purchased as evaluation boards, it is convenient to utilize evaluation software to interface with them. Each digital component (with the exception of the FPGA, discussed later) is internally configured by a set

Digital Block Diagram

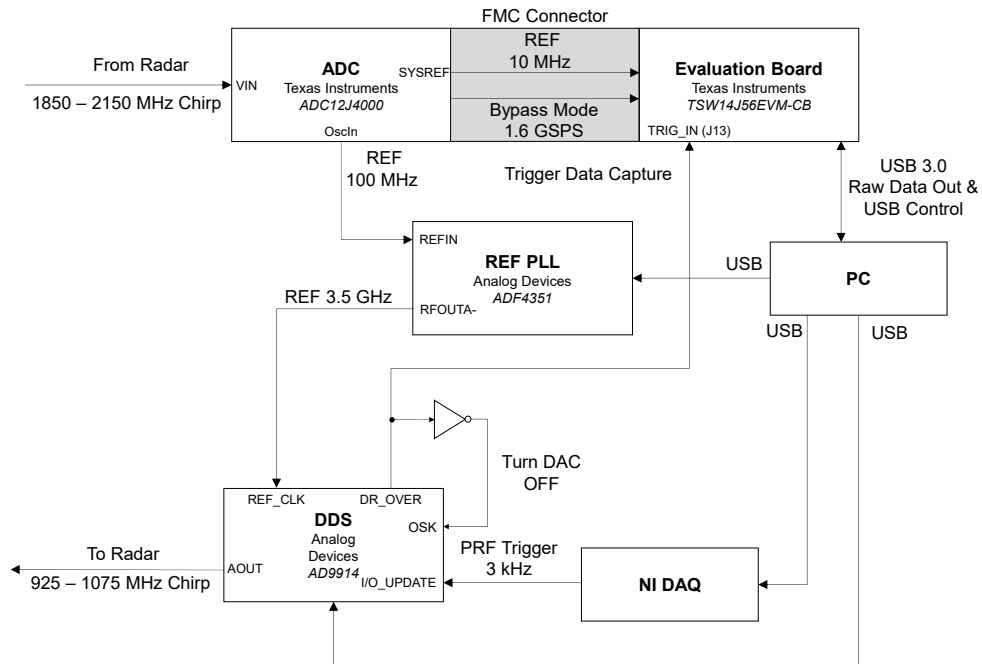


Figure 3.3: The original digital back-end block diagram

of registers, and the registers are written to through an SPI interface. The evaluation boards each have an SPI programmer chip that is controlled by a user's PC through USB. The evaluation software is essentially a user-friendly way of interfacing with the devices, allowing for changes in device operation to be made without a full understanding of the device registry (although register level interface is also available through the evaluation software). Because the hardware can be interfaced directly through SPI, the evaluation software is used with the expectation that an SPI interface will eventually be developed to program all the devices on startup with one single piece of software, rather than using a different piece of software for each device.

To interface with the FPGA, a piece of software developed by TI called *HSDC*

Pro is used. The HSDC Pro GUI is shown in Figure 3.4. HSDC Pro allows for firmware to be ported to the FPGA and it provides an interface for loading captured data from the FPGA RAM into the computer over a USB cable. The software also provides some basic digital signal processing features, including frequency domain analysis and tapering windows. Raw data can also be exported from HSDC Pro in a comma-separated value (CSV) or raw binary file for further processing in MATLAB.

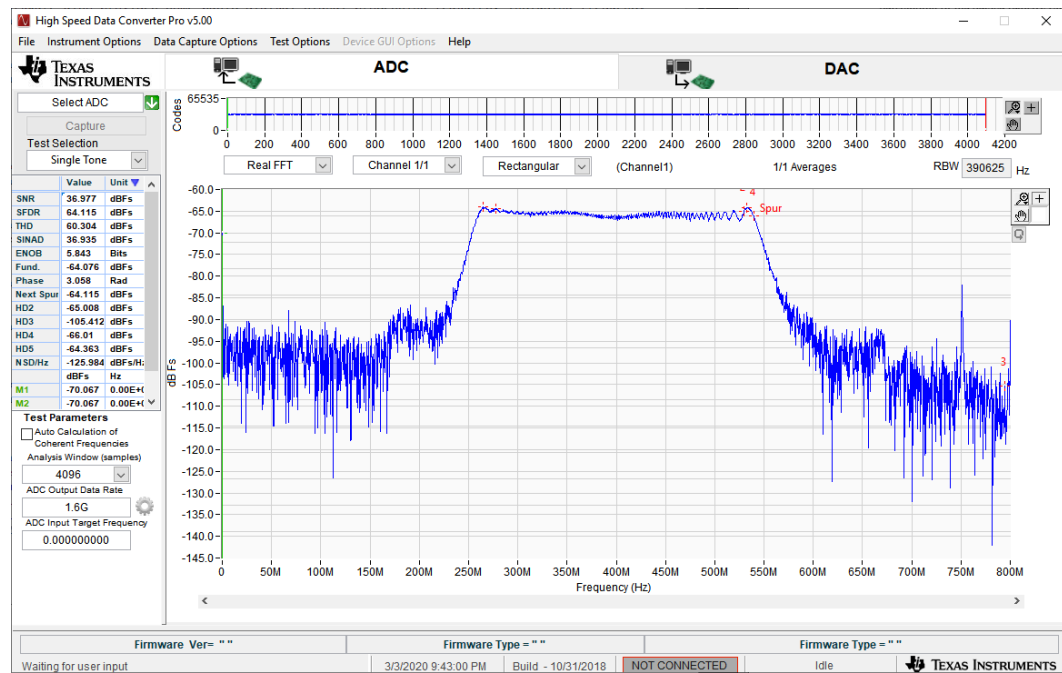


Figure 3.4: The HSDC Pro GUI

The software is useful in that it enables the basic use of the FPGA without writing custom firmware. The major drawback to HSDC Pro in a pulse-Doppler application is that it does not allow for pulsed data captures. Instead, the software only allows for the manual retrieval of a single data capture at a time. In conjunction with the unmodified FPGA firmware shipped with the capture board, the capture of radar pulses is only possible through the continuous capture of samples. At a sample

rate of 1.6 GHz and 2 bytes of memory required for each sample, the FPGA memory will fill up after 1.25 seconds, falling short of the 4.5 seconds required to form a full imaging CPI [7]. Solving this problem requires some minor modifications to the FPGA firmware to allow pulsed data captures without pulsed interfacing with HSDC Pro.

Chapter 4

Digital System Configuration

In this chapter, the implementation details of the digital system described in Section 3.3 are elaborated upon. Although the top-level hardware for the digital transceiver was selected in [7], the full operation of the system requires much more involved digital design and the development of additional hardware. In the preliminary stages of implementing the digital radar transceiver, the radar pulses were found to be incoherently timed with one another. Additionally, anomalies in the frequency spectra of the LFM pulses were observed, indicating improper waveform synthesis. These problems are presented in detail in this chapter, along with the preliminary and permanent solutions to these problems. Most of the modifications to the digital system detailed in this chapter are implemented in the final operation of the radar system.

4.1 DDS Configuration

The ADC evaluation board and software have several pre-defined configuration files enabling application-agnostic operation at several different sampling frequencies (including 1.6 GHz). Because the ADC does not require fine-tuning to enable the system to start running, the first important task in setting up the digital back-end

is using the DDS to correctly synthesize the LFM pulse. The key constraint that the DDS must fulfill is the synthesis of a 150 MHz LFM pulse with the output of the DDS windowed to 1 μ s. Additionally, the DDS must produce this LFM pulse each time it is triggered externally by the 3 kHz PRF.

4.1.1 LFM Pulse Generation

To generate the LFM pulse, a feature of the AD9914 called the Digital Ramp Generator (DRG) is employed. The DRG allows the controlled linear sweeping of the frequency, phase, or amplitude of the output waveform, enabling frequency, phase, or amplitude modulation. Using the DRG to sweep over frequency enables the generation of a 150 MHz LFM pulse at 1 GHz by sweeping from 925 MHz up to 1075 MHz.

The output frequency of the AD9914 is controlled by a *frequency tuning word* (FTW) 32-bit register. The FTW register is computed similarly to θ_{inc} in (2.8) for a given output frequency f_{OUT} and a DDS clock frequency f_{DDS} by

$$FTW = \text{round} \left(2^{32} \left(\frac{f_{OUT}}{f_{DDS}} \right) \right) \quad (4.1)$$

where the FTW becomes the number of cycles per sample scaled to fill a 32-bit integer. The DRG is configured by giving a lower frequency limit FTW f_{LO} , a higher frequency limit FTW f_{HI} , a frequency step size Δf , and a ramp rate Δt . Given these configuration values, the DRG will begin the frequency sweep with an output frequency of f_{LO} and will increase in frequency by Δf Hz every Δt s. The frequency sweep will end once f_{HI} is reached by the sweep. The registers representing f_{LO} and f_{HI} are given by FTW_{LO} and FTW_{HI} , respectively, and are solved for using (4.1). The 32-bit register value representing Δf is $STEP_P$ and is

given by

$$\text{STEP}_P = 2^{32} \frac{\Delta f}{f_{\text{DDS}}} \quad (4.2)$$

while the 16-bit register value representing Δt is P and is given by

$$P = \Delta t \frac{f_{\text{DDS}}}{24} \quad (4.3)$$

The factor of 24 in (4.3) is because the internal timer clock governing the DDS operations and the steps of the sweep runs at the frequency of the DDS sampling clock divided by 24. That is, the timer clock runs at a frequency f_{timer} given by

$$f_{\text{timer}} = \frac{f_{\text{DDS}}}{24} \quad (4.4)$$

The effects of this clock on system coherence are discussed later.

The number of steps required to complete a sweep can be solved for both as a function of the sweep bandwidth and the pulse length. That is,

$$\# \text{ Steps} = \frac{f_{\text{HI}} - f_{\text{LO}}}{\Delta f} = \frac{T_p}{\Delta t} \quad (4.5)$$

where T_p is the pulse length discussed in Section 2.1. To give the most continuous sweep, Δt is assumed to be the smallest possible value. This is achieved when $P = 1$, leading to $\Delta t = \frac{24}{f_{\text{DDS}}}$. Rearranging the right-hand side of (4.5) to solve for Δf yields

$$\Delta f = \frac{\Delta t (f_{\text{HI}} - f_{\text{LO}})}{T_p} = \frac{24 (f_{\text{HI}} - f_{\text{LO}})}{f_{\text{DDS}} T_p} \quad (4.6)$$

Given $f_{\text{LO}} = 925 \text{ MHz}$, $f_{\text{HI}} = 1075 \text{ MHz}$, $f_{\text{DDS}} = 3.5 \text{ GHz}$, and $T_p = 1 \mu\text{s}$, Δf is determined to be 1.0286 MHz. Using $P = 1$ and rearranging (4.3) for Δt yields $\Delta t = 6.8571 \text{ ns}$. Using (4.1)-(4.3), the register values required for the desired DDS

Parameter	Register Name	Parameter Value	Register Value (Hex)
f_{LO}	FTW_{LO}	925 MHz	0x43A83A83
f_{HI}	FTW_{HI}	1075 MHz	0x4EA0EA0E
Δf	$STEP_P$	1.0286 MHz	0x00134272
Δt	P	6.8571 ns	0x0001

Table 4.1: The register values used to configure the DDS DRG

configuration are solved for. These configuration values are given in Table 4.1.1.

Once the DRG is configured to produce the LFM waveform, the signal must then be windowed in the output. Because the RF front-end does not utilize a pulsed amplifier, this windowing must be performed in the DDS. To perform the output windowing, three features of the DDS are utilized: the output shift keying (OSK) function, the DRG over (DROVER) digital output, and the DRG accumulator auto-clear function.

Chapter 5

Pulse-to-Pulse Incoherence

Chapter 6

Start-Up Incoherence and Loopback Calibration

Chapter 7

Sensing Results

Chapter 8

Conclusion and Future Work

Bibliography

- [1] M. I. Skolnik, *Introduction to Radar Systems*, 2nd ed. McGraw-Hill, 1980.
- [2] J. W. McDaniel, "Design, Integration, and Miniaturization of a Multichannel Ultra-Wideband Snow Radar Receiver and Passive Microwave Components," Master's thesis, University of Kansas, 2015.
- [3] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. McGraw-Hill Education, 2014.
- [4] E. Murphy and C. Slattery, "All About direct Digital Synthesis."
- [5] JEDEC, *Serial Interface for Data Converters*, December 2017.
- [6] S. Murthy, "Implementing JESD204B SYSREF and Achieving Deterministic Latency With ADC32RF45," Texas Instruments, Inc., 2016.
- [7] J. McDaniel, "Ku-Band Synthetic Aperture Radar: System Parameters Study," 2017.
- [8] *3.5 GSPS Direct Digital Synthesizer with 12-Bit DAC*, Analog Devices, Feb. 2017, Rev. F.
- [9] *ADC12J4000 12-Bit, 4-GSPS ADC with Integrated DDC*, Texas Instruments, Oct. 2017, Rev. D.
- [10] *TSW14J56 JESD204B High-Speed Data Capture and Pattern Generator Card User's Guide*, Texas Instruments, Jan. 2016, Rev. C.
- [11] J. Brinkhurst, *ADC12J4000EVM Schematic*, Texas Instruments, Sept. 2014.
- [12] *TRF3765 Integer-N/Fractional-N PLL With Integrated VCO*, Texas Instruments, Dec. 2015, Rev. E.
- [13] *LMK0482x Ultra Low-Noise JESD204B Compliant Clock Jitter Cleaner with Dual Loop PLLs*, Texas Instruments, Dec. 2015, Rev. AR.
- [14] *ADF4351 Wideband Synthesizer with Integrated VCO*, Analog Devices, Jan. 2017, Rev. A.