# Table of Contents

Project Summary ........................................................................................................ 3

Function Description and System Block Diagram ........................................ 4-9

  *Mixer Subsystem* ............................................................................................. 5

  *Data Acquisition Subsystem* .......................................................................... 6

  *Frequency Synthesizer Subsystem* ................................................................. 6

    DDS Theory of Operation ............................................................................. 6-7

Mathematical Verification .................................................................................... 10-11

Preliminary Experimental Results ................................................................. 12-15

  *Matlab and Simulink Simulations* ................................................................. 12-13

    AD9854 Evaluation Board ........................................................................... 13-15

Preliminary Datasheet ......................................................................................... 16

Equipment List .................................................................................................. 16

Related Standards and Patents ........................................................................ 17

Schedule ............................................................................................................. 18

References ......................................................................................................... 19

Appendix A ......................................................................................................... 20-21
Project Summary

The Scanning Digital Radar Receiver is a scaled down variation of an early warning radar system. The receiver uses a frequency synthesizer to create a variable local oscillator signal. Using this signal the radar frequency can be down converted and digitalized by the data acquisition board. Once digitalized, the signal is loaded into Matlab and DSP techniques are used to characterize the radar signal. The system will output the following time domain characteristics: Pulse Repetition Rate (PRR), Pulse Width (PW), and RF Frequency. This information can be used in other application such as signal jamming.
Introduction

The Scanning Digital Radar Receiver is a project supported by Northrop Grumman Corporation. The objective of the system is to be able to detect a pulsed radar signal over a large frequency range. The frequency range is from 50 MHz to 120 MHz. After a pulsed radar signal has been identified, the system will output the characteristics of the signal to the user.

High Level Block Diagram

Figure 1 is the high level system block diagram. The input, user input mode, is strictly a software input and the various modes are discussed in the software section. The RF signal is mixed with the very stable frequency synthesizer, which is acting like a local oscillator.

The internal signal IF, is filtered and then fed into the A/D card. The signal is then processed using DSP methods, and if a radar signal is present the signals characteristics are sent as outputs.

(Figure 1 High Level System Block Diagram)
To help understand the systems outputs, Figure 2 is a pulsed radar signal in the time domain. By processing the signal in the frequency domain, the system can determine time domain characteristics. The following is the list of outputs and how they relate to Figure 2:

The Pulse Width (PW) corresponds to \( \tau \) in the figure below.

The Pulse Repetition Rate (PRR) is the inverse of the period, \( T \), of the time domain signal.

The Freq RF is the \( \text{F}_{\text{RF}} \) of the sinusoid under the pulse enveloper.

(Figure 2 Time Domain Representation of the Pulsed Radar Signal)

**High Level Block Diagram Subsystems**

*Mixer*

Figure 3 is the mixer subsystem. The mixer multiplies the RF with LO and then outputs IF. The intermediate frequency (IF) contains frequencies of \( \text{F}_{\text{RF}}-\text{F}_{\text{LO}} \) and \( \text{F}_{\text{RF}}+\text{F}_{\text{LO}} \). The purpose of this subsystem is to convert the RF signal to a lower frequency so that it can be processed by the computer.

(Figure 3 Mixer Subsystem)
**Data Acquisition**

Figure 4 is the Data Acquisition subsystem. It is composed of an analog filter and the data acquisition board. The IF signal is filtered so that the down converted signal, $F_{RF}-F_{LO}$, remains. The signal is then digitalized by the data acquisition card.

The data acquisition card is the I/O tech 3005USB, it has a maximum sampling rate of 1MS/sec. To assure that the correct data is recorded a 400 KHz low-pass filter is used.

![Diagram of Data Acquisition Subsystem]

(Figure 4 Data Acquisition Subsystem)

**Frequency Synthesizer**

Figure 5 is the Frequency Synthesizer subsystem. The control word is sent to the frequency synthesizer, either by serial or parallel ports. Based off this control word the output LO is generated.

The Analog Digital DDS 9854 is going to be used as the frequency synthesizer. It uses direct digital synthesis to create a very stable sinusoid wave form. The chip runs at 300 MHz but its output must be filtered at 120 MHz to prevent aliasing.

![Diagram of Frequency Synthesizer Subsystem]

(Figure 5 Frequency Synthesizer Subsystem)

**DDS Theory of Operation**

DDS or Direct Digital Synthesis is the ability to use a digital system to create and analog output. Figure 6 is a block diagram of the core of every DDS system. The frequency tuning word is loaded into the Phase Accumulator. The values in the Phase Accumulator correspond to points on the cycle of the output sinewave. Resolution of the Phase Accumulator is determined by $N$ and can range from 24 – 48 bits.
The output of the phase accumulator is linear and cannot directly be used to generate a sinewave or any other waveform except a ramp. Therefore, a phase-to-amplitude lookup table is used to convert a truncated version of the phase accumulator’s instantaneous output value into the sinewave amplitude information that is presented to the D/A converter.

**Signal Processing**

The signal processing subsystem consists of a computer running Matlab and Simulink. After the IF signal has been sampled by the data acquisition card it is loaded into Matlab to do the digital signal processing. Here a Fast Fourier Transform (FFT) is taken and the various radar characteristics are determined.

**Modes of Operation**

There will be three modes of operation that the user can choose from. Scan mode is where the system is continuously scanning and displaying signal information to the user. In Scan & Hold mode, the system stops once a signal has been detected. This mode could be used to allow the user to enter the signal characteristics into a jamming device. Manual mode would allow the user to pick a frequency and check it for a radar signal. This mode might be used to calibrate or test the system.
Figure 7 is the high level flowchart illustrating the various modes of operation.

(Figure 7 High Level Flow Chart)
**DSP Flow Chart**

Figure 8 is the DSP flow chart. The signal processing begins by checking if a signal was detected. If no signal is detected the frequency synthesizer is incremented and the program exits. However, if a signal is detected, then its FFT is taken. The FFT point with the maximum value is the frequency of the IF signal. The maximum value following the IF frequency is the pulse repetition rate. Finally, the zero crossing that comes after the IF frequency is the pulse width.

(Figure 8 DSP Flow Chart)

All code is written in Matlab in the form of m-files. To see detailed examples of this code please refer to Appendix A.
Mathematical Verification

The purpose of this section is to verify how the FFT of a pulsed radar section will yield time domain results. Figure 9 is the time domain representation of a pulsed radar signal. To obtain the FFT the signal is broken up into two components, which are then multiplied together.

(Figure 9 Time Domain Representation of Pulsed Radar Signal)

The two time domain components are:

\[ x_1(t) = \cos(2\pi f_{RF} T) \]

\[ x_2(t) = A \sum \prod \left( \frac{t - nT \tau}{\tau} \right) \]

Then;

\[ g(t) = x_1(t)x_2(t) \]

Use Multiplication Theorem (table 2-1)\(^1\)

\[ x_1(t)x_2(t) = X_1(f) \ast X_2(f) \]

Using Fourier Transform (table 2-2)\(^1\) and Time Delay Theorem (table 2-1)\(^1\)

\[ X_1(f) = \frac{1}{2} \delta(f - f_{RF}) + \frac{1}{2} \delta(f + f_{RF}) \]

\[ X_2(f) = A \sum_n \text{sinc}(\eta f) e^{j(-j2\pi fnT)} \]

\[ G(f) = X_1(f) \ast X_2(f) \]
\[ G(f) = \frac{1}{2} \delta(f - f_{RF}) + \frac{1}{2} \delta(f + f_{RF}) * A \sum_n \text{sinc}(\pi f) e^{j2\pi fnT} \]

\[ G(f) = \frac{A\tau}{2} \sum_n \left[ \delta(f - f_{RF}) \text{sinc}(\tau(f - f_{RF})) e^{j2\pi fnT} + \delta(f + f_{RF}) \text{sinc}(\tau(f + f_{RF})) e^{j2\pi fnT} \right] \]

Using Frequency Shifting Property. 2 pg.38-39

\[ G(f) = \frac{A\tau}{2} \text{sinc}(\tau(f - f_{RF})) \sum_n e^{j2\pi fnT} + \frac{A\tau}{2} \text{sinc}(\tau(f + f_{RF})) \sum_n e^{j2\pi fnT} \]

Using Poisson Sum Formula. Equ. 3-38¹

\[ \sum_{n=-\infty}^{\infty} e^{j2\pi fnT} = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T}) \]

\[ G(f) = \frac{A\tau}{2} \text{sinc}(\tau(f - f_{RF})) \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta((f - f_{RF}) - \frac{n}{T}) + \frac{A\tau}{2} \text{sinc}(\tau(f + f_{RF})) \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta((f + f_{RF}) - \frac{n}{T}) \]

Figure 10 below is a plot of the above equation. It shows the doubled sided sinc functions and how they are made up of the delta functions.

(Figure 10 Plot of G(f))


Preliminary Experimental Results

Matlab and Simulink Simulations

Figure 11 below is the Simulink block diagram for the system. It includes the RF front end, local oscillator, and the IF filter. The filter is set for 400 KHz and the signal coming from the RF_signal block is the pulsed radar signal.

(Figure 11 System Simulation in Simulink)

For the simulation the RF signal is a 105 MHz sinusoid mixed with a 2 ms period, 20% duty cycle pulse train. The local oscillator signal is at 104.9 MHz, the output signal will be at $F_{IF} = F_{RF} - F_{LO}$. Figure 12 is the time domain plot of the filtered IF signal.

(Figure 12 Time Domain Plot of Simulated IF Signal)
The idea is to be able to extract time domain information mathematically from the frequency domain. By taking an FFT of the signal, the time domain characteristics can be obtained as shown in Figure 10. The code to perform the calculations is in Appendix A. To find the original RF signal the LO frequency is added to the IF frequency. So 104.9 MHz added to 100 KHz is a 105 MHz.

In Figure 13 the FFT of the IF signal is shown. The plot shows where all the time domain characteristics are calculated from. It should be noted that this plot is a plot of the magnitude of the FFT. Matlab cannot plot complex numbers in vector form; therefore, the zero crossing in now the lowest value after the maximum at which the IF frequency is found.

![Figure 13 Frequency Domain Plot of the IF Signal](image)

(Figure 13 Frequency Domain Plot of the IF Signal)

**AD9854 Evaluation Board**

The AD9854 Evaluation Board has been successfully connected and tested using the provided computer software. To test the evaluation board a 15 MHz 2 V pk-pk sine wave was used as the reference clock. The board was powered by at 3.3 V for appropriate CMOS logic levels. The evaluation board comes with a standard printer connection for the PC’s parallel port, and the software allows for testing of all the AD9854’s functions.
Figure 14 and Figure 15 are scope plots taken of the board I DAC output. Figure 14 is a 50 MHz sine wave and Figure 15 is a 120 MHz sine wave. For testing both of the output sine waves are at the full amplitude the DAC will allow.

(Figure 14 Scope Plot of 50 MHz sine wave from AD9854)

(Figure 15 Scope Plot of 120 MHz sine wave from AD9854)

Communication with AD9854 Evaluation Board

Early on communication with the AD9854 was going to be done with external logic using Matlab code and the PC’s parallel port. There were too many problems interfacing the logic to the PC and this idea was aborted.
The new communication strategy is to use the provided PC connector and modifying the Visual Basic source code of the AD9854. Matlab can be setup in a server mode were it can be called to perform operations from an external program. In the case of this project Visual Basic will be used.

**Results of Preliminary Work**

All of the Matlab code works great; however, it cannot truly be tested until the data acquisition board arrives. Right now the problem with the code is that you have to change the sampling rate every time you change the RF frequency.

The AD9854 Evaluation Board works very well; however, at this time only the provided software application has been used to test it. The only area of concern is that as the frequency of the output is increased the amplitude of the sine wave decreases. This can be seen from Figures 14 and 15. In Figure 14 with a 50 MHz sine wave the amplitude is at 996 mV, when the frequency is increase to 120 MHz, as in Figure 15, the amplitude decreased 420 MHz. This will be a concern when spcing out the mixer subsystem.
Preliminary Datasheet

Table 1 is the preliminary datasheet for the Scanning Digital Radar Receiver. It describes input requirements and individual subsystem requirements.

(Table 1 Scanning Digital Radar Receiver Preliminary Datasheet)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Min</th>
<th>Max</th>
<th>Typical</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SYSTEM SPECIFICATIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel Scanning Time</td>
<td></td>
<td></td>
<td>-</td>
<td>[Sec]</td>
</tr>
<tr>
<td><strong>INPUT SPECIFICATIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carrier Frequency ($f_{RF}$)</td>
<td>50</td>
<td>120</td>
<td>-</td>
<td>[MHz]</td>
</tr>
<tr>
<td>Pulse Repition Rate (PRR)</td>
<td>500</td>
<td>5000</td>
<td>500</td>
<td>[Hz]</td>
</tr>
<tr>
<td>Pulse Width (PW)</td>
<td>80</td>
<td>400</td>
<td>-</td>
<td>[us]</td>
</tr>
<tr>
<td>MIXER (unknown at this time)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL OSCILATOR (AD9854)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Frequency</td>
<td>0</td>
<td>300</td>
<td>120</td>
<td>[MHz]</td>
</tr>
<tr>
<td>Step Size</td>
<td>0.1</td>
<td>-</td>
<td>1.00E+06</td>
<td>[Hz]</td>
</tr>
<tr>
<td>Clock Jitter (100 MHz)</td>
<td></td>
<td></td>
<td>7</td>
<td>[Ps rms]</td>
</tr>
<tr>
<td>(for more specs refer to AD9854 datasheet)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DATA ACQUISITION (IOTech 3005USB)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Rate</td>
<td></td>
<td></td>
<td>1</td>
<td>[MHz]</td>
</tr>
<tr>
<td>(for more specs refer to IOTech 3005 datasheet)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Equipment List

- 1, IOTech 3005USB data acquisition board
- 1, AD9854ASQ Evaluation Board
- 1, PC running Matlab Version 7.01
- 1, Agilent E3630A Power Supply
- 1, Agilent 33220A Function Generator
Standards Relating to Project

The following is a list of standards documents that pertain to this project. Since the standards must be purchased, only the basic information is listed. Information was gathered from www.nssn.org.

IEEE Standard Radar Definitions
Author: IEEE
Published: 4/27/1998
Document #: 686-1997

Early Warning Systems
Author: Department of Defense (DoD)
Published: 8/31/1995
Document #: QAP-124 ED.1

Electronic Counter-countermeasures in Radar Systems Acquisition
Author: Department of Defense (DoD)
Published: 6/05/1987
Document #: MIL-HDBK-293

Patents Relating to Project

The following is a list of patents with abstracts that pertain to this project. Information was gathered from www.uspto.gov.

Patents

5,010,341 High Pulse Repetition Frequency Early Warning Receiver
4,860,013 Automatic Threshold Multi-channel Digital Radar Early Warning Sys.
6,879,281 Pulse Radar Detection System
6,587,072 Pulse Radar Detection System
4,504,830 Synthetic Pulse Radar System and Method
6,693,578 Mixer Optimization for Active Radar Warning Receiver

Patent Applications

20050179582 Radar Detection Method and Apparatus
20050134502 Pulse Wave Radar Device
20030193430 Pulse Radar Detection System
20040212358 Scanning RF Receiver
Schedule

Table 2 is an outline of work that will be completed each week. The schedule is expected to change to some degree.

(Table 2 Schedule of Tasks)

<table>
<thead>
<tr>
<th>Week</th>
<th>Tasks To Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter Break</td>
<td>Write Program to communicate with DDS board.</td>
</tr>
<tr>
<td></td>
<td>Begin Testing and Writing code for Data Acq Board</td>
</tr>
<tr>
<td></td>
<td>Begin Research of Mixer Component</td>
</tr>
<tr>
<td>Week 1</td>
<td>Research Mixer Components</td>
</tr>
<tr>
<td></td>
<td>Testing and Writing code for Data Acq Board</td>
</tr>
<tr>
<td>Week 2</td>
<td>Finish Testing and Writing code for Data Acq Board</td>
</tr>
<tr>
<td></td>
<td>Choose Mixer</td>
</tr>
<tr>
<td>Week 3</td>
<td>Integrate Data Acq Board and DDS Board Code</td>
</tr>
<tr>
<td>Week 4</td>
<td>Design and Implementation of IF filter</td>
</tr>
<tr>
<td></td>
<td>Filter Testing and Verification</td>
</tr>
<tr>
<td>Week 5</td>
<td>Design and Implementation of Pre-filter</td>
</tr>
<tr>
<td></td>
<td>Filter Testing and Verification</td>
</tr>
<tr>
<td></td>
<td>Spring Break</td>
</tr>
<tr>
<td>Week 6</td>
<td>Filter Testing and Verification</td>
</tr>
<tr>
<td></td>
<td>Mixer integration into system</td>
</tr>
<tr>
<td>Week 7</td>
<td>Integrate Filters and Mixer with DDS and Data Acq</td>
</tr>
<tr>
<td></td>
<td>Begin converting code to GUI</td>
</tr>
<tr>
<td>Week 8</td>
<td>Integrate Filters and Mixer with DDS and Data Acq</td>
</tr>
<tr>
<td></td>
<td>Continue converting code to GUI</td>
</tr>
<tr>
<td>Week 9</td>
<td>Integrate Filters and Mixer with DDS and Data Acq</td>
</tr>
<tr>
<td></td>
<td>Continue converting code to GUI</td>
</tr>
<tr>
<td>Week 10</td>
<td>Finish System Intergration</td>
</tr>
<tr>
<td></td>
<td>Begin Testing and Verification of entire system</td>
</tr>
<tr>
<td>Week 11</td>
<td>Writing Final Report</td>
</tr>
<tr>
<td></td>
<td>Begin Testing and Verification of entire system</td>
</tr>
<tr>
<td>Week 12</td>
<td>Writing Final Report</td>
</tr>
<tr>
<td></td>
<td>Prepare Oral Presentation</td>
</tr>
<tr>
<td>Week 13</td>
<td>Writing Final Report</td>
</tr>
<tr>
<td></td>
<td>Prepare Oral Presentation</td>
</tr>
<tr>
<td>Week 14</td>
<td>EXPO</td>
</tr>
</tbody>
</table>

18
References


Appendix A

Appendix A contains m-files that are used with my project. These m-files take a simulated pulse radar signal and determine its time domain characteristics.

%.m file for taking the fft of the filter IF signal
%from the Simulink system
%by: Ryan Hamor
%created: 11/1/05

```
fs=length(tout);
yq=fs/2;                   %nyquist number to cut off upper fft
freq=1e8*[0:fs-1]/fs;      %define frequency scale
fft_IF=abs(fft(IF));       %take the magnitude of the fft
```

%.m file for finding the frequency of the IF signal.
%Created by: Ryan Hamor
%Date: 11/03/05
%**********************************************************************
**
mx_mag=max(fft_IF(2:nyq)); %Cut out DC and only search for first peak.
j=1;                        %initialize counter
while  fft_IF(j)~=mx_mag    %find point in vector where maximum
   j=j+1;
end                         %save the value to a new count variable.
i=j;
IF_freq=freq(i);

%.m file to detect the Pulse Repetition Rate. It finds the second
maximum
%of the fft signal and what frequency it is at.
%Created by: Ryan Hamor
%Date: 11/3/05
%**********************************************************************
***

```
j=i+1;                      %move counter 1 past mx_frep
mx_mag2=max(fft_IF(j:nyq)); %find second maximum value
while  fft_IF(j)~=mx_mag2
   j=j+1
end
k=j;                        %k is value of location of second peak
mx2_freq=freq(k);
PRR=(mx2_freq-IF_freq)^(-1);
```
%.m file to find the Pulse width of the radar signal
%created by: Ryan Hamor
%data: 11/03/06
******************************************************************************
***
pl=i;                 %set pl to start at the location for Freq
IF.
p2=p1+1;              %p2 is always one value ahead of pl.

while fft_IF(p1)>fft_IF(p2)
    p2=p2+1;
    pl=pl+1;
end

pw_Freq=freq(p1);
PW=(pw_Freq-IF_freq)^(-1);
PW_percent=(PW/PRR)*100;