# Doppler Filtering and Adaptive Clutter Cancellation for Radar

University of Oklahoma

ECE 3793-001

Dr. Nathan Goodman

December 3rd, 2023

Ged Miller

Joshua Philip

## Introduction

The goal of this assignment was to perform frequency analysis which converted the given data set compromised of "range bins" to the Doppler domain. These "range bins" are reflections collected from a radar and separated according to distance. Each "range bin" was measured by transmitting 64 pulses at uniform intervals. For the first part, the simulated data was processed to create a basic Range-Doppler Map. In the second part, a window function was applied to enhance the previously produced Range-Doppler Map. For the third part, adaptive filtering methods were employed to suppress the clutter and emphasize the detection of moving targets on the Range-Doppler map.

Doppler data analysis allows for detection of target velocity and target tracking. In tandem with filtering, these techniques can be used to detect targets in military and civilian domains, such as aviation, meteorology, and maritime operations. The purpose of this experiment is to use pulse-doppler radar methods to extra target data from a radar dataset.

## Methods

To extract usable data from the given idealized radar data, Pulse-Doppler radar techniques are utilized to process the data. First, basic Range-Doppler processing is used, then windowing techniques are used for side-lobe reduction, and lastly adaptive filtering reduces clutter.

Basic Range-Doppler Processing on the data requires to first create a Range-Doppler Map that represents the target velocity and range, in other words, the Doppler shift. The Discrete Time Fourier Transform is used on the received radar pulses by using the Fast Fourier Transform. The Fast Fourier Transform (*Figure 1*) computes the Discrete Time Fourier Transform at discrete frequency points, converting the signal from time domain to the frequency domain.

$$X(f) = \sum_{n=0}^{N-1} x[n]e^{-j2\pi fn}$$

Figure 1: The Fast Fourier Transform.

Next, the relative motion, or velocity, between the radar and the target can be determined via Doppler shift analysis; the doppler shift in the frequency of the reflected signal. Doppler shift, as seen in *Figure 2*, is not only sensitive to velocity but also direction, enabling the ability to determine if a target is moving closer or farther away from the radar. Lastly, Doppler shift is instrumental in reducing clutter, in this instance, stationary or slow moving objects.

$$f_d = \frac{2v}{\lambda}$$

Figure 2: Doppler shift equation.

An RDM map is then created when plotting the magnitude of the output of the Fast Fourier Transform against the range and Doppler frequency. This map is pivotal in visualizing the relative velocity between targets.

Since weak signals near strong reflectors can be obscured by sidelobes, windowing techniques like the Hamming, Blackman, and Flattopwin windows were tested. Ultimately, the Hamming window was decided upon due to it having the narrowest doppler pattern, meaning it was the most accurate allowing for better target tracking and velocity measurement. The Hamming window (*Figure 3*) is multiplied by the time domain signal to reduce the sidelobes of the signal in the frequency domain. However, since the Hamming window smooths out the ends of signal to zero, reducing it's clutter, it also slightly increases the width of the main lobe, reducing accuracy.

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad \therefore for\ 0 \leq n \leq N-1$$

Figure 3: Hamming window equation.

Adaptive filtering is added to reduce clutter from slow and stationary objects whilst improving detection of moving objects. An estimation of the pulse data is put into a matrix called the Covariance Matrix to estimate the statistical properties of clutter and noise. Then, adaptive weights are assigned to each frequency bin using the inverse of the Covariance Matrix and Doppler manifold matrices to enhance the signals apart from the clutter as seen in *Figure 4*.

$$w(f_k) = \frac{R^{-1}v(f_k)}{\sqrt{v^H(f_k)R^{-1}v(f_k)}}$$

Figure 4: Adaptive Weight Formula.

Lastly, the adaptive weights are used to generate the final RDM which is significantly more accurate and has greater target visibility than previous versions.

Results:

After successfully implementing the Fast Fourier Transform of the radar data, the basic Range Doppler Map is produced as seen in *Figure 5*. The basic RDM presents a strong narrow signal at only zero Hz with errant signals at different ranges and frequencies, since the signal is so strong, there a likely multiple moving targets obscured within the main lobe at zero Hz. Changing $N_{fft}$ significantly increases the number of signals and background noise. The computation time also increases, so much so that it can overload the testing computers. Since there is strong clutter at zero Hz and the main lobe is inaccurate, further processing is required to obtain accurate results. The main challenge in this process will be to separate the moving targets from the background clutter without removing relevant moving targets.

Next, a Hamming window function was applied to radar pulses altering the RDM as seen in *Figure 6*. The Hamming window function greatly reduces clutter and signal sidelobes especially targets near stationary objects, however, some clutter and sidelobes still remain. Using the Blackman window produces a slightly wider main lobe with less clutter than the Hamming window. On the other hand, the Flattopwin window produces a much wider main lobe with much great sidelobes and clutter. For this experiment, it seems the Hamming window is the most accurate with the least clutter and smallest sidelobes of the three window functions.

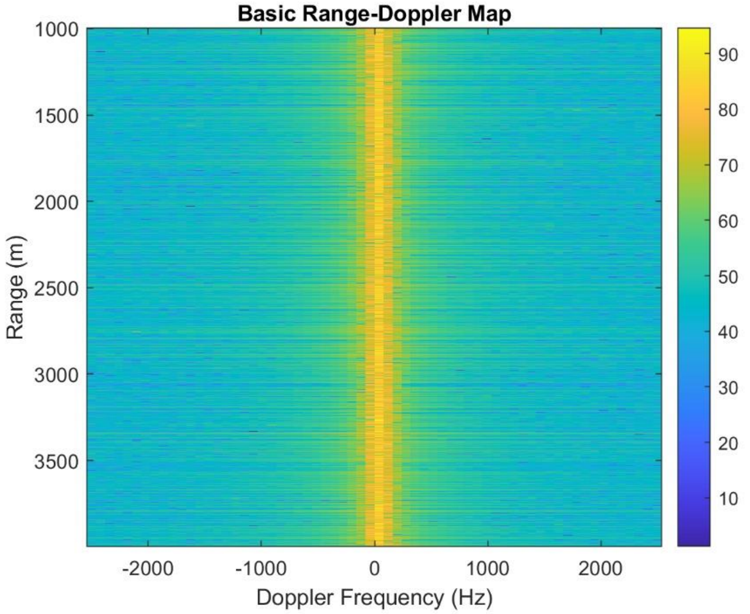

Figure 5: Basic Range-Doppler Map of radar data.

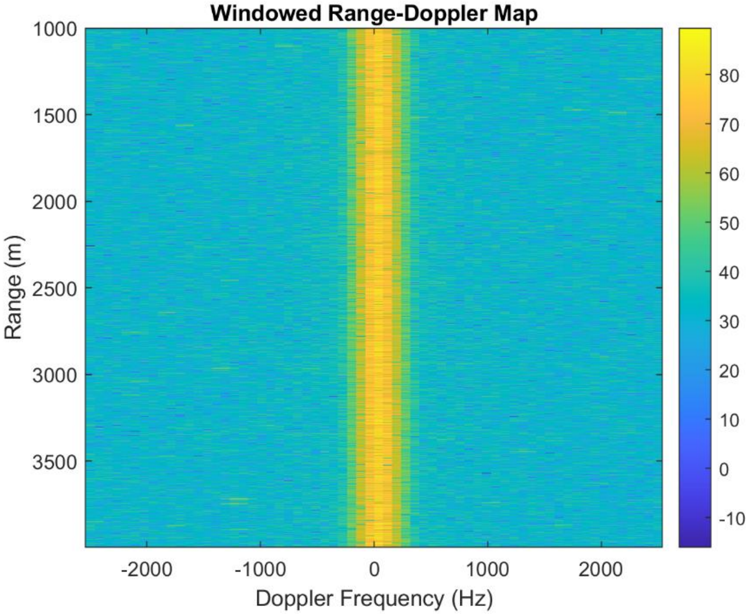

Figure 6: Windowed Range-Doppler Map of radar data.

Adaptive filtering utilizing the covariance matrix and doppler manifold matrix resulted in significantly reduced clutter and distinction of moving targets as seen in *Figure 7*. The background noise is uniform across the RDM demonstrating effective clutter suppression. The adaptive filtering also suppressed signals a zero Hz, allowing for clear distinction of moving targets.

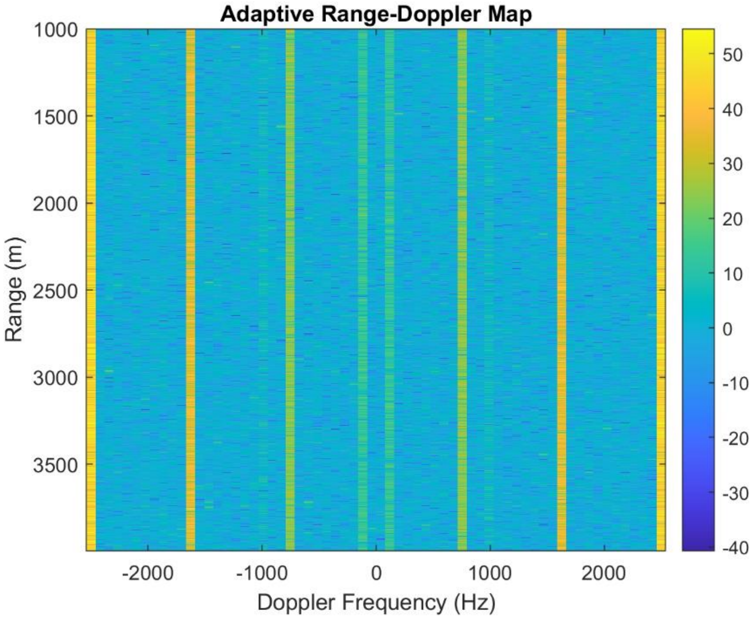

Figure 7: Adaptive Range Doppler Map of radar data.

The frequency response of adaptive filter weights at zero Hz and far away from zero Hz seen in *Figure 8* and *Figure 9* can help determine the systems response to moving targets and the balance between clutter rejection and target detection. As seen in *Figure 8*, the frequency response of the target frequency at -10 Hz, there is a significant dip in magnitude at zero Hz, signaling that the system is effectively attenuating signals at zero frequency, meaning that the system reduces clutter like slow or stationary objects well. There is however, a strong sinc like signal at around 1700 Hz, indicates that the system is sensitive to signals at that frequency or that there is a harmonic response or interference at this frequency, however, since a target in *Figure 7* is at this frequency, it seems that the target has a direct reflection back to the radar.

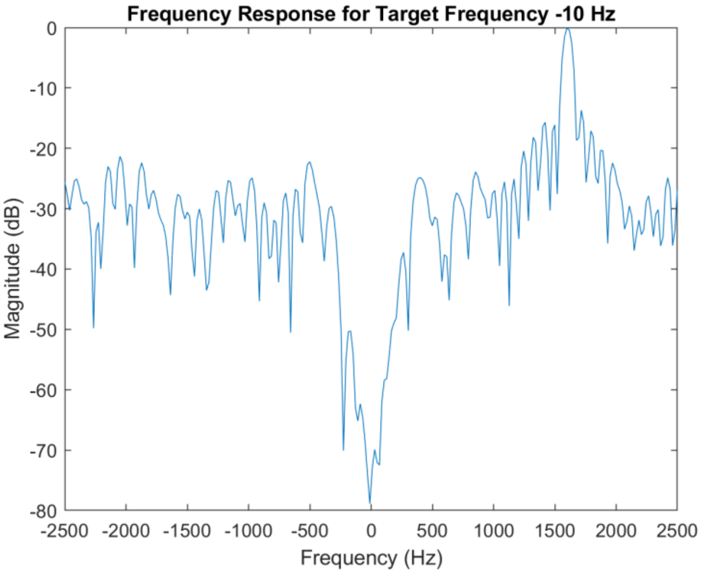

Figure 8: Frequency Response for the Target Frequency of -10 Hz.

The frequency response of *Figure 9* is very similar to the response seen in *Figure 8* even though it's target frequency is much farther away at 1250 Hz. This means the system may have a broadband filter response, and may suppress clutter at a wide range of frequencies. The system also appears to be symmetric, with positive frequencies mirroring behavior of negative frequencies. This can be from the native properties of symmetry the Fourier transform creates, the presence of similar target velocities at both target frequencies, or even limitations of the algorithm's range.

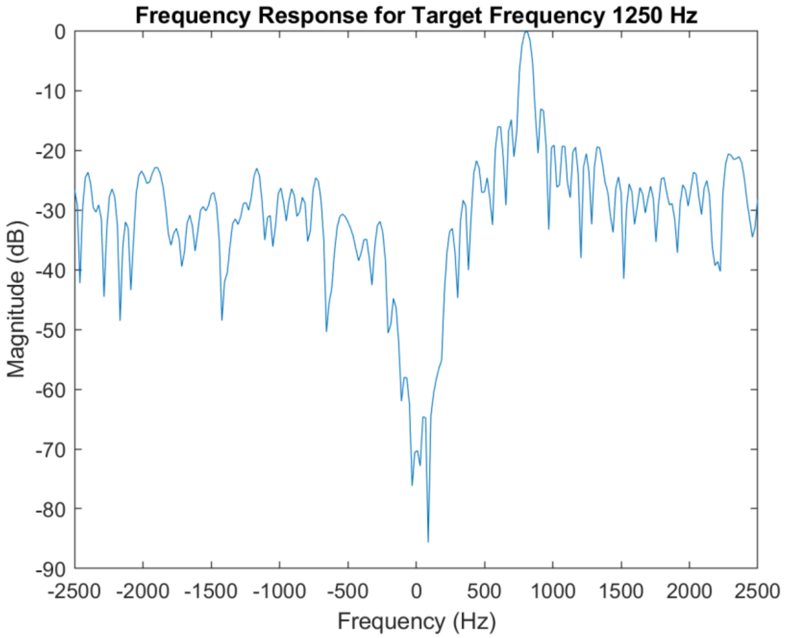

Figure 9: Frequency Response for the Target Frequency of 1250 Hz.

Conclusion:

The MATLAB implementation of Pulse Doppler radar signal processing has demonstrated its ability to distinguish moving signals from clutter and stationary objects. Using methods like the Fast Fourier transform to create a Range Doppler Map allowed for visualization of targets and clutter that was instrumental in providing scope and direction for the experiment. The Hamming window function allowed for clutter to be reduced and the adaptive clutter reduction enabled the distinction of discrete moving targets in the RDM. Analyzing frequency response characteristics allowed for the system's behavior at frequencies near and far from 0 Hz to be discovered, like its broadband capability.

Pulse Doppler radar signal processing has not only shown to be effective for the given data set, but also has demonstrated its utility for solving real world problems. This technology would be beneficial in commercial and military applications and future studies into the technology, like applications of Pulse Doppler radar signal processing to Synthetic Aperture Radar, could enhance spatial resolution and help provide processing in complex environments.

## Code Appendix:

```
 close all;
%Load dataset
load('datafile_3793_proj2.mat');

%Define parameters
PRI = 0.2e-3;   %Pulse Repetition Interval in seconds
PRF = 1 / PRI;  %Pulse Repetition Frequency in Hz
Nr = 1001;      %Number of Range Bins
%Np = 64;        %Number of Pulses
Nfft = 2^nextpow2(Np); %FFT length as next power of 2 from Np

%Part I: Compute and Plot Basic Range-Doppler Map
RDM_basic = zeros(Nr, Nfft);
for i = 1:Nr
    pulse_data = Z(i, :);
    fft_result = fft(pulse_data, Nfft);
    fft_result_shifted = fftshift(fft_result);
    RDM_basic(i, :) = fft_result_shifted;
end

%Plotting the Basic RDM
figure;
imagesc(linspace(-PRF/2, PRF/2, Nfft), R_axis, 20*log10(abs(RDM_basic)));
xlabel('Doppler Frequency (Hz)');
ylabel('Range (m)');
title('Basic Range-Doppler Map');
```

```
colorbar;


%Part II: Apply Window Function and Compute Range-Doppler Map
%window = blackman(Np).';  %Blackman window didn't look any different.
window = hamming(Np).'; %Hamming window
%window = flattopwin(Np).'; %Produces a much wider doppler return centered at 0 Hz.
RDM_windowed = zeros(Nr, Nfft);
for i = 1:Nr
    pulse_data = Z(i, :) .* window;
    fft_result = fft(pulse_data, Nfft);
    fft_result_shifted = fftshift(fft_result);
    RDM_windowed(i, :) = fft_result_shifted;
end


%Plotting the Windowed RDM
figure;
imagesc(linspace(-PRF/2, PRF/2, Nfft), R_axis, 20*log10(abs(RDM_windowed)));
xlabel('Doppler Frequency (Hz)');
ylabel('Range (m)');
title('Windowed Range-Doppler Map');
colorbar;


%Part III: Adaptive Filtering
%Estimate Covariance Matrix R
R = zeros(Np, Np);
for i = 1:Nr
    xl = Z(i, :).';
    Xl = xl * xl';
    R = R + Xl;

end

R = R / Nr;



%Create Doppler Manifold Matrix V

f = linspace(-PRF/2, PRF/2, Nfft);

n = (0:Np-1).';

V = exp(1j * 2 * pi * f .* n);



%Compute Inverse of R

R_inv = inv(R);



%Compute Adaptive Weights

W = R_inv * V;

for k = 1:size(V, 2)

    W(:, k) = W(:, k) / (V(:, k)' * R_inv * V(:, k));

end



%Compute Adaptively-Filtered Range-Doppler Map

RDM_adaptive = Z * conj(W);



%Plotting the Adaptive RDM

figure;

imagesc(linspace(-PRF/2, PRF/2, Nfft), R_axis, 20*log10(abs(RDM_adaptive)));

xlabel('Doppler Frequency (Hz)');

ylabel('Range (m)');

title('Adaptive Range-Doppler Map');

colorbar;
```

```
%Frequency Response Analysis of Adaptive Filter Weights
selected_freqs = [-PRF/2500, PRF/4]; %Tested Frequencies
doppler_freqs = linspace(-PRF/2, PRF/2, Nfft); %Doppler frequencies based on Nfft

for i = 1:length(selected_freqs)
    %Find the index of the closest frequency in the Doppler frequency array
    [~, idx] = min(abs(doppler_freqs - selected_freqs(i)));

    %Use this index to get the corresponding column from W
    w = W(:, idx);
    w_fft = fft(w, Nfft*4); %Extended FFT length for smoother plot
    w_fft_shifted = fftshift(w_fft);

    %Plot frequency response
    extended_freq_array = linspace(-PRF/2, PRF/2, Nfft*4);
    figure;
    plot(extended_freq_array, 20*log10(abs(w_fft_shifted)));
    title(['Frequency Response for Target Frequency ' num2str(selected_freqs(i)) ' Hz']);
    xlabel('Frequency (Hz)');
    ylabel('Magnitude (dB)');
end
```