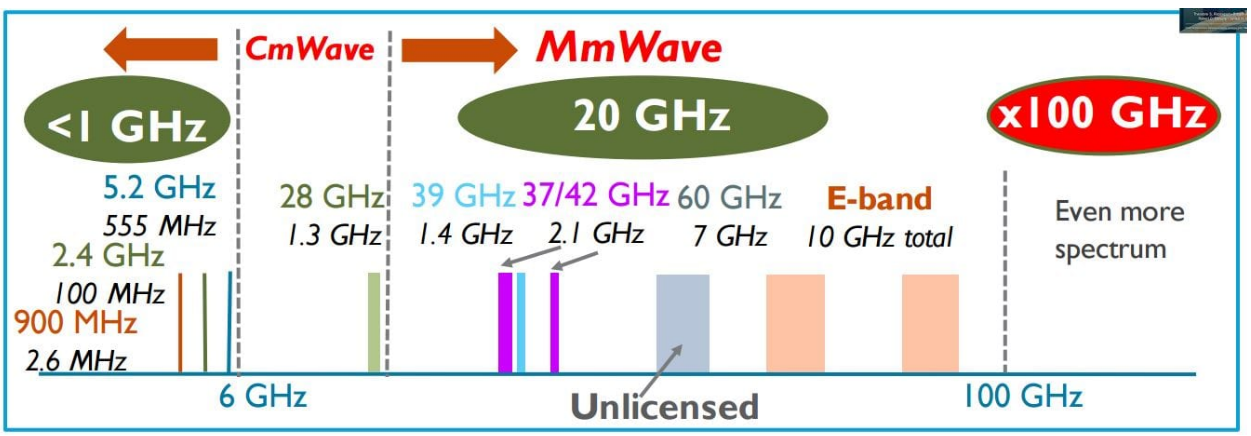
mmWave Lab

ECE 146 Lab 4

December 14, 2019



Student: Ivan Arevalo
Perm Number: 5613567
Email: ifa@ucsb.com

Department of Electrical and Computer Engineering, UCSB

0 Introduction

This lab is an introduction into the fundamentals of radar sensing with mm waves for various applications. We will start by deriving some basic equations and eventually test our design with MATLAB.

1 Section 1 - 3: Derving basic equations

mmWave Radar Sensing Lab

(1) $y(t) = 2S_p(t-\tau)S_p(t)$

$S_p(t) = \cos(2\pi f_c t + \pi S t^2) \quad 0 \le t \le T_d \quad T_d = 20\mu s$

$* \; 2\cos\phi\cos\theta = \cos(\phi-\theta) + \cos(\phi+\theta)$

$$y(t) = 2\cos(\underbrace{2\pi f_c(t-\tau) + \pi S(t-\tau)^2}_{\phi})\cos(\underbrace{2\pi f_c t + \pi S t^2}_{\theta})$$

$$(2\pi f_c t - 2\pi f_c\tau + (t^2 - 2t\tau + \tau^2)\pi S)$$

$$y(t) = \cos(\cancel{2\pi f_c t} - 2\pi f_c\tau + \cancel{\pi S t^2} - 2\pi S t\tau + \pi S\tau^2 \cancel{- 2\pi f_c t} \cancel{- \pi S t^2})$$

$$+\cos(2\pi f_c t - 2\pi f_c\tau + \pi S t^2 - 2\pi S t\tau + \pi S\tau^2 + 2\pi f_c t + \pi S t^2)$$

$$y(t) = \cos(-2\pi f_c\tau - 2\pi S t\tau + \pi S\tau^2)$$

$\boxed{\text{filtered by LPF}}$

$$\cancel{+\cos(4\pi f_c t + 2\pi S t^2 + \pi S\tau^2 - 2\pi f_c\tau - 2\pi S t\tau)}$$

$$y(t) = \cos(-(2\pi f_c\tau + 2\pi S t\tau - \pi S\tau^2))$$

$\boxed{\text{Cos even function}}$

$$y(t) = \cos(2\pi f_c\tau + 2\pi S t\tau - \pi S\tau^2)$$

$$\boxed{y(t) = \cos(2\pi f_0 t + \phi_0) \quad f_0 = S\tau \quad \phi_0 = 2\pi f_c\tau - \pi S\tau^2}$$

In practice, a single chirp doesn't give us enough information because since it is periodic, it will give us the same phase for different distance).

$1 \times 10^{6}\ \mu s$
$1\ s$

(2) $\tau = \frac{2R}{c}$ — R distance to object from ~~transceiver~~.
c Speed of light.
τ delay of recieved signal

(a) Max range 30m. What is largest value of τ we need to accomodate? Compare to frame duration $10\ \mu s$?

$$\tau = \frac{2(30)\,m}{3 \times 10^{8}\ m/s} = \boxed{2 \times 10^{-7}\ s} \quad \boxed{200\ ns}$$

It is 50 times smaller than the frame duration.

(b). What is largest f_0 to estimate?

$f_0 = \tau \mathcal{S}$; $\mathcal{S} = 50\ MHz/\mu s$

$$f_0 = 0.2\ \mu s \times 50\ MHz/\mu s = \boxed{10\ MHz}$$

$f_0 = 10\ MHz$ is 100 times smaller than the 1 GHz variation in instantaneous frequency over chirp duration.

(3) Generalization of mixed return and sent signals after LPF for K targets:

$$y(t) = \sum^{K} A_i \cos(2\pi f_K t + \phi_K) I_{[\tau_K, T_d]} + n(t)$$

$$y(t) \approx \sum_{K=1}^{K} A_i \cos(2\pi f_K t + \phi_K) + n(t), \quad 0 \le t < T_d$$

where $f_K = \mathcal{S}\tau_K$ $\quad \phi_K = 2\pi f_c \tau_K - \pi \mathcal{S} \tau_K^2$

We need to sample at twice the highest freq. present in our signal. Since $f_K = 10\ MHz$ is the highest frequency to accomodate frequencies corresponding to a range of up to 30m, we need to a least sample at a rate of 20 MHz

2 Section 4: Simulation model

```
%% Section 4 Generate received Signal

sampling_rate = 20*10^6; % Sampling rate 20 Mhz. Derived in part 3;
chirp_duration = 20*10^-6; % Chirp duration 20 us. Given
f_c = 80*10^9; % 80 GHz Given
N = chirp_duration * sampling_rate; % Number of samples.
n = 0:N; % Samples vector
k = 3; % Number of targets -> received signals
target_distaces = [3; 10; 30];
SNR_closest_target = 80; % 30dB
slew_rate = 5*10^13; % 50 MHz / us --> 5*10^13 Hz / s
c = 3*10^8; % Speed of light

% SNR = (N * A_k^2)/variance
% Given that the SNR of the closest target is 30dB, determine magnituds A_k
% set the others assuming the same radar cross section for each target, and
% 1/R^4 loss
% 10log(raw_SNR_30m_target) = 30dB

% Noise
mean = 0;
variance = 1;
noise_vector = normrnd(mean, sqrt(variance), [int16(N), k]); %k = 3 Number of targets

raw_SNR_3m_target = 10^(SNR_closest_target/10);
A_k_reference = sqrt(raw_SNR_3m_target/N);
A_k = zeros(k, 1);
A_k(1) = A_k_reference;
for i=2:k
    A_k(i) = A_k_reference/((target_distaces(i)/target_distaces(1))^4);

    % Make noise proportional in weaker signals
    noise_vector(:,i) = noise_vector(:,i)./((target_distaces(i)/target_distaces(1))^4);

end
%%%%%

% Frequencies
time_delays = (2 * target_distaces)./c;
f_k = slew_rate * time_delays;

% Phases
phase = -2*pi*f_c*time_delays - pi*slew_rate.*(time_delays.^2);

% Generate Received Signals
y_k = A_k .* cos(2*pi*f_k.*(n/sampling_rate) + phase) + noise_vector';

figure('Name', 'Signals Received no noise');
plot(y_k(1,:)); hold on; plot(y_k(2,:));plot(y_k(3,:));

y_1_n = A_k(1) * cos(2*pi*f_k(1).*(n/sampling_rate) + phase(1)) + noise_vector(:,1)';
```

```
y_2_n = A_k(2) * cos(2*pi*f_k(2).*(n/sampling_rate) + phase(2)) + noise_vector(:,2)';
y_3_n = A_k(3) * cos(2*pi*f_k(3).*(n/sampling_rate) + phase(3)) + noise_vector(:,3)';

% Received Signal y[n]
y_n = sum(y_k);
figure('Name', 'Added Signals Received no noise');
plot(y_n);

%% Estimate the dominant frequencies in y[n]
fft_size = 2^(nextpow2(length(y_n))+1); % Use next power of 2 for efficient fft
y_n_fft = fft(y_n, fft_size);
fs=1/(fft_size*(1/sampling_rate)); %actual frequency resolution attained
%set of frequencies for which Fourier transform has been computed using DFT
f = ((1:fft_size)-1-fft_size/2)*fs;
figure; plot(f, fftshift(abs(y_n_fft)));

% Try windowing first
hann_window = hann(length(y_n));
windowed_y_n = y_n .* hann_window';
fft_windowed_y_n = fft(windowed_y_n, fft_size);
figure; plot(f,fftshift(abs(fft_windowed_y_n)));

%% Estimate the dominant frequencies and map them to range estimates
peak_frequencies = 1*10^6*[.99 3.16 4.57 4.805 8.403 9.141];

% find peaks gave me too much data.
% [pks, loc] = findpeaks(abs(fft_windowed_y_n));
% [pk, loc_ind] = maxk(pks,10);
% peak_frequencies = f(loc(loc_ind));

corresponding_time_delays = peak_frequencies/slew_rate;
corresponding_ranges = corresponding_time_delays*c/2;
figure('Name','Histogram Part c');
histogram(corresponding_ranges,100);
title('Estimated ranges histogram');xlabel('Distance (m)');

range_error = corresponding_ranges - target_distaces;
figure('Name','Histogram Part d');
histogram(range_error(1,:), 40);
title('Estimated range error for 3m target histogram');xlabel('Error Distance (m)');
figure('Name','Histogram Part d');
histogram(range_error(2,:), 40);
title('Estimated range error for 10m target histogram');xlabel('Error Distance (m)');
figure('Name','Histogram Part d');
histogram(range_error(3,:), 40);
title('Estimated range error for 30m target histogram');xlabel('Error Distance (m)');
```

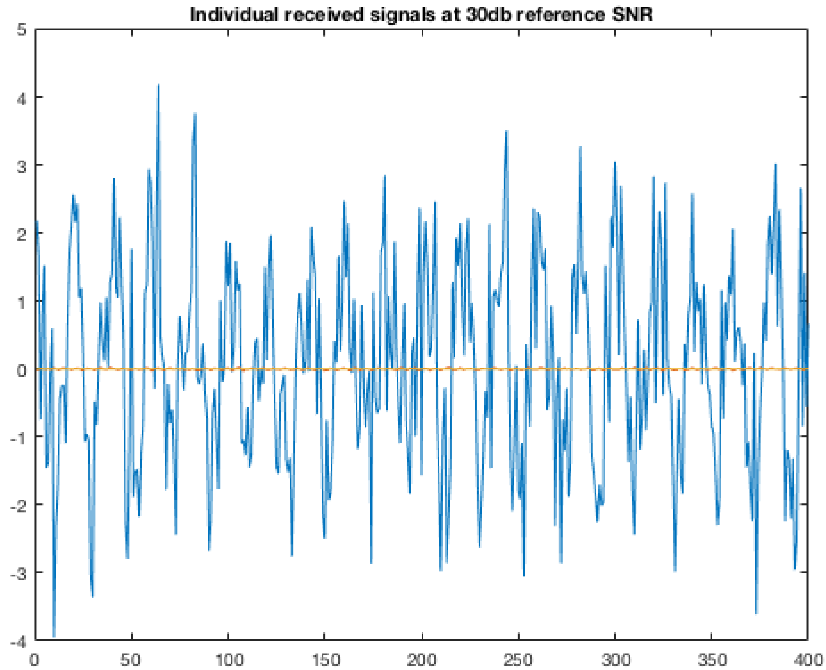


Figure 1: Individual received signals

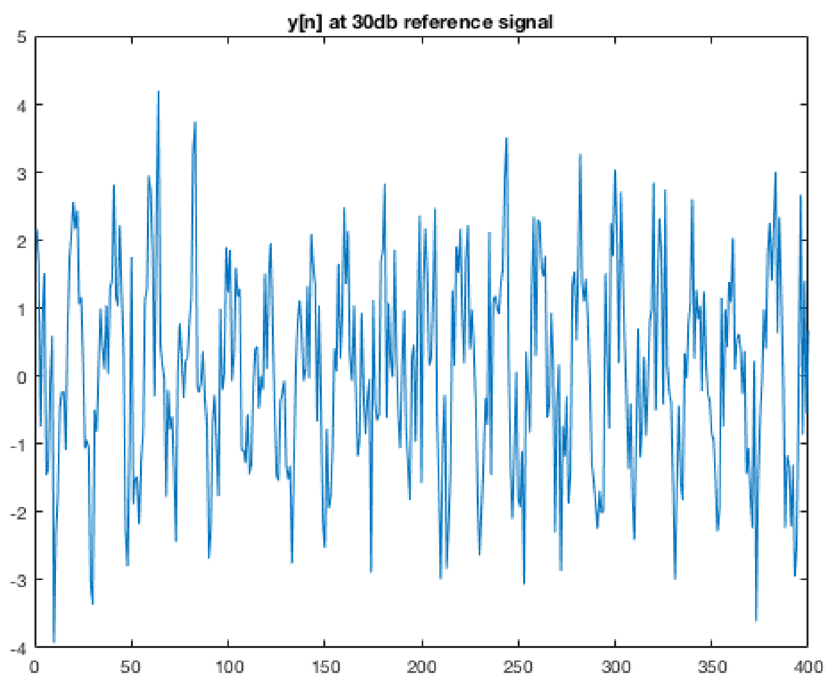


Figure 2: Summed received signals

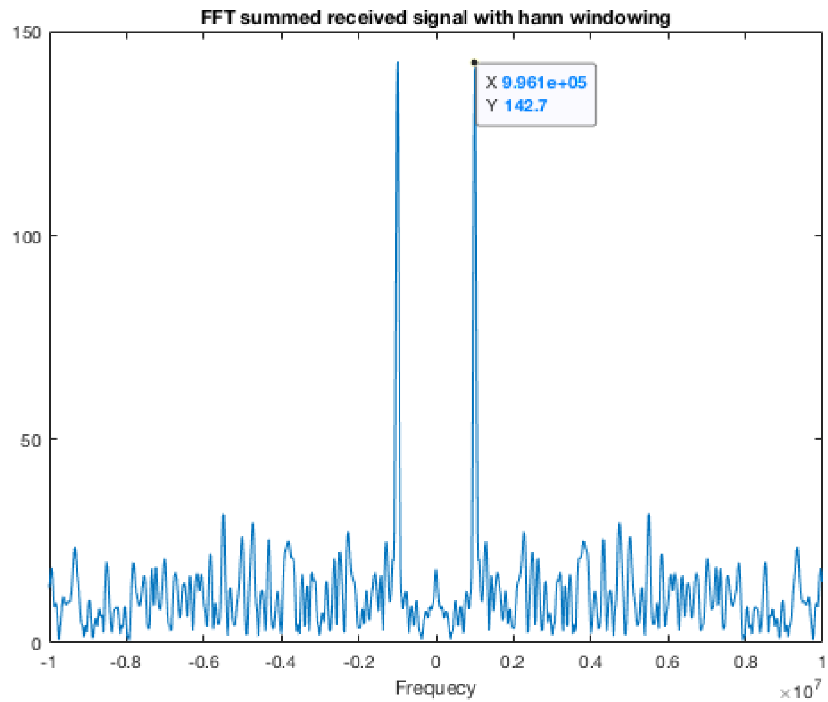


Figure 3: FFT of summed received signals with hann windowing

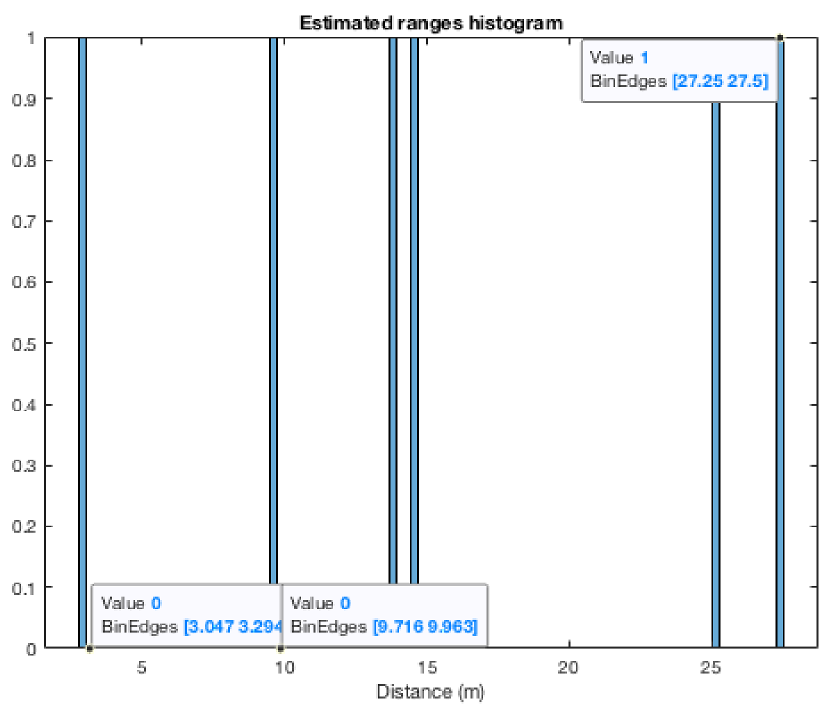


Figure 4: Histogram of estimated ranges

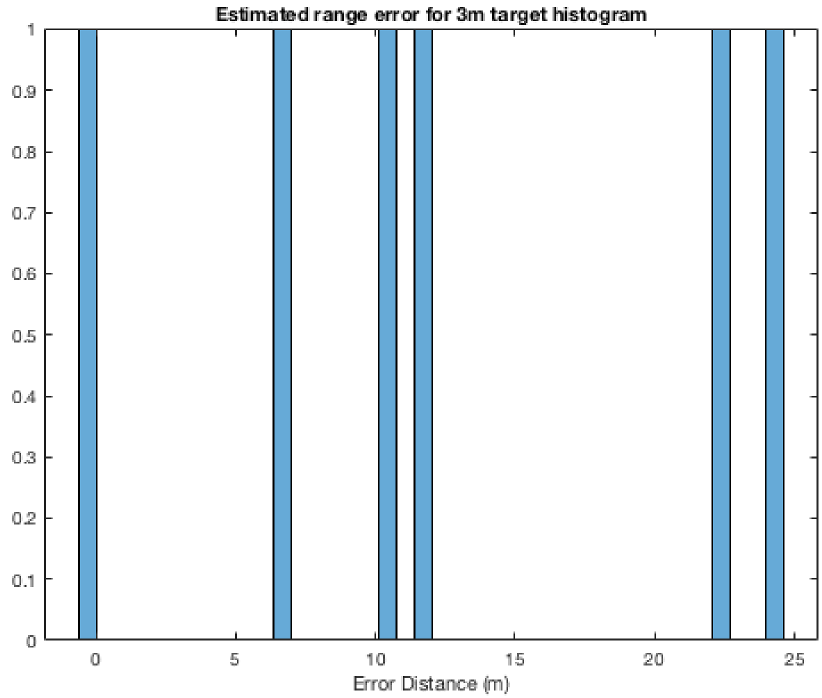


Figure 5: Histogram of estimated error for 3m target

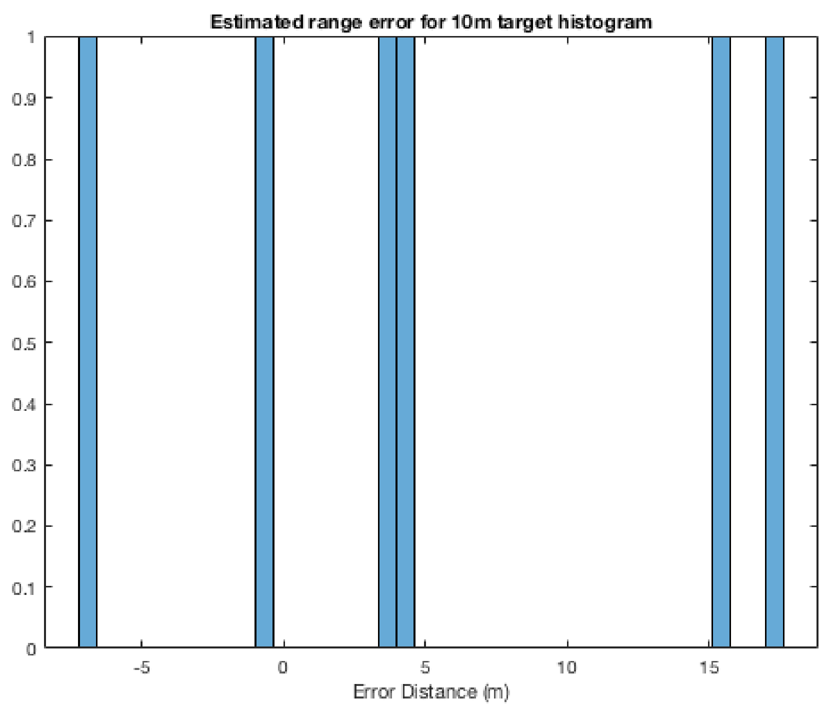


Figure 6: Histogram of estimated error for 10m target

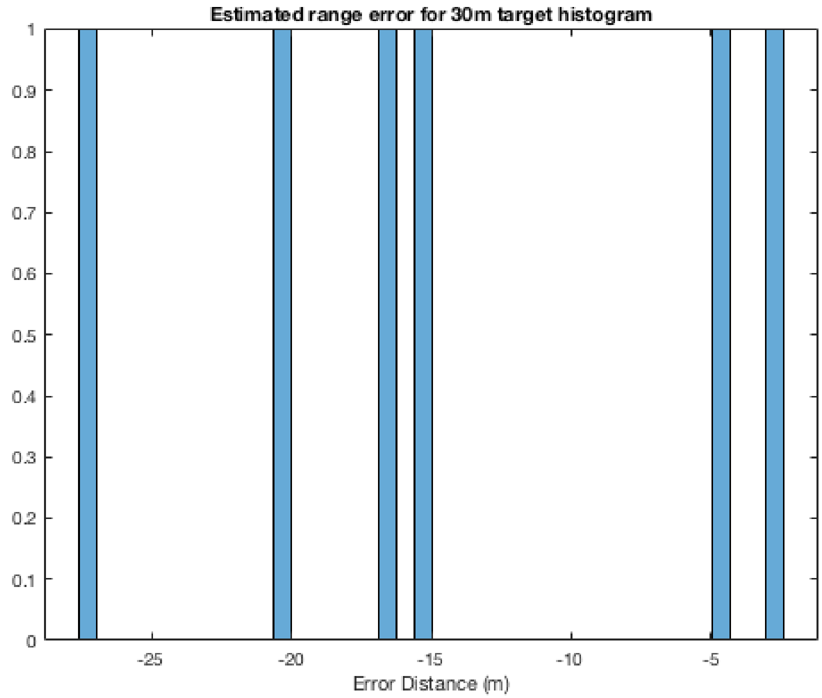


Figure 7: Histogram of estimated error for 10m target

At 30db refenrece SNR, i was able to get small range error by windowing with a hann window for each of the targets as seen in Figure 4. I tested other reference SNR's and observed that at 20dB I was not able to accurately find peaks that related to the correct ranges. Furthermore, at 60dB it got very easy to spot the correct peaks for the targets.

3 Section 5 - 6: Velocity sensing

(3) $\theta(t) = 2\pi f_0 t + \phi_0$ function of time

(a) $\cos(\underbrace{2\pi f_0 t + \phi_0}_{\theta(t)})$ τ function of t

$\omega = 2\pi f \rightarrow 2\pi f(t) = \frac{d\theta(t)}{dt}$

$f(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt} = \frac{1}{2\pi}\frac{d}{dt}\left(2\pi f_0 t + \phi_0\right)$

$f(t) = \frac{1}{2\pi}\frac{d}{dt}\left(2\pi S'\tau(t)t - 2\pi f_c \tau(t) - \pi S'\tau(t)^2\right)$ → chainrule

$ab' + a'b$

$f(t) = \frac{1}{2\pi}\left(2\pi S'\tau(t) + 2\pi S'\dot{\tau}t - 2\pi f_c\dot{\tau} - 2\pi S'\tau(\dot{\tau})\right)$

$f(t) = \cancel{\frac{1}{2\pi}2\pi}\left(S'\tau(t) + S'\dot{\tau}(t)t - f_c\dot{\tau}(t) - S'\tau(t)\dot{\tau}(t)\right)$

$\boxed{f(t) = S'\tau\tau + S'\dot{\tau}t - f_c\dot{\tau}(t) - S'\tau(t)\dot{\tau}(t)}$

(b) $f(t) = S'\tau - f_c\dot{\tau} = S'\tau - f_D = S'\tau - \frac{2v_R}{c}f_c$

$f_D = \frac{2v_R}{c}f_c = \frac{2(3\,m/s)}{3\times10^8\,m/s}\,80\times10^9\,Hz$ $c = \lambda f$

3m) 20μs

$10^8/80\times10^9$

$\boxed{f_D = 1.6\,kHz}$ $\Delta\phi = \frac{4\pi v T_c}{\lambda} = \frac{4\pi v T_c}{c/f}$

$-\Delta\phi = \frac{4\pi v T_c}{\lambda}$ from TI paper

$\boxed{\Delta\phi = \frac{4\pi v T_c f_c}{c} = \frac{4\pi(3m)(20\mu s)(80\times10^9\,Hz)}{3\times10^8\,m/s} = 0.201}$

(c) Target moving at 3 m/s. How much would the range change over that duration?

$$\Delta\phi = \frac{4\pi v_R T_c f_c}{c} \approx 3(2\pi) \approx 6\pi$$

$$\frac{4\pi (3 m/s) T_c (80\times10^9 Hz)}{3\times10^8 m/s} = 6\pi$$

$$T_c = \frac{\cancel{6}^{2}\pi (3\times10^8 m/s)}{\cancel{4}\pi (3 m/s)(80\times10^9 Hz)} = 0.0025 s$$

$$\frac{0.0025 s}{20 \mu s} \approx \boxed{125 \text{ chirps}}$$

(b) $$\phi = \frac{4\pi (0.2 m/s) T_c (80\times10^9 Hz)}{3\times10^8 m/s} \approx 6\pi$$

$$T_c = \frac{6\pi (3\times10^8 m/s)}{4\pi (0.2 m/s)(80\times10^9 Hz)} = 0.0375$$

$$\frac{0.0375}{20\mu s} = \boxed{1875 \text{ chirps}}$$

4 Conclusion

In conclusion, we began this lab going over the theory of radar sensing and deriving some of the necessary equations. We then simulated a 3 target design with objects at 3, 10, and 30 meters. We generated the appropiate received signals including noise, and proceeded to derive the range estimates by windowing our function and observing the peaks of the frequency domain. We later moved on to derives some equations for velocity sensing and got insight into how this task might be performed