When running the code, I was not able to get the first two plots and Matlab has given below warning.

MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL

Table

Description automatically generated with medium confidence

Chart

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, chart

Description automatically generated

Chart

Description automatically generated

clear;

close all;

clc;

addpath('data/');

c = physconst('LightSpeed');

%% SDR parameters

filename = 'Meas\_3.dat'; % File name

fileID = fopen(filename,'r');

dataArray = textscan(fileID,'%f');

fclose(fileID);

radarData = dataArray{1};

clearvars fileID dataArray ans;

fc = radarData(1); % Center frequency

Tsweep = radarData(2)\*1e-3; % Sweep time in ms

NTS = radarData(3); % Number of time samples per sweep

Bw = radarData(4); % FMCW Bandwidth. For FSK, it is frequency step; For CW, it is 0.

Data = radarData(5:end); % raw data in I+j\*Q format

Data\_1 = Data(1:2:end); % Data of channel 1

Data\_2 = Data(2:2:end); % Data of channel 2

lambda=c/fc; % seconds

%%

samp\_rate=NTS/Tsweep;

FFT\_size=1024;

dopp\_FFT\_size=128;

numberSweeps = length(Data\_1)/NTS;

data\_block1=reshape(Data\_1,[NTS,numberSweeps]);

Rng = c\*linspace(0, samp\_rate, FFT\_size/2+1)/2\*(Tsweep\*1e-3)/(2\*Bw);

psd\_matrix=zeros(FFT\_size/2+1,numberSweeps);

data\_matrix=zeros(FFT\_size/2+1,numberSweeps);

t=(1:numberSweeps)\*Tsweep;

for index=1:numberSweeps

signal=data\_block1(:,index);

signal=signal.\*hamming(NTS);

fsignal1 = fft(signal, FFT\_size);

fsingal=fsignal1(1:FFT\_size/2+1);

data\_matrix(:,index)=fsingal;

psdx=(1/length(fsingal)).\*abs(fsingal).^2;

psd = 10\*log10(psdx);

psd\_matrix(:,index) = psd;

end

figure(1)

surf(Rng,t,psd\_matrix')

xlim([0 50])

ylim([0 10])

bar=colorbar('EastOutside');

shading interp

colormap(jet)

caxis([30 80])

view(2)

xlabel('Distance (m)')

ylabel('Intensity (dB)')

ylabel(bar,'Normalized intenstiy(dB)','FontName','Arial','FontSize',12)

title('Range Waterfall')

%%

vel\_vector=linspace(-1/(2\*Tsweep),1/(2\*Tsweep),dopp\_FFT\_size)\*lambda/2;

PN=30;

pro\_vel\_prof=round(numberSweeps/PN);

psd\_rng\_vel=zeros(pro\_vel\_prof,length(Rng),dopp\_FFT\_size);

psd\_vel=zeros(pro\_vel\_prof,dopp\_FFT\_size);

for index=0:1:(pro\_vel\_prof-1)

signal=data\_matrix(:,PN\*index+1:PN\*(index+1));

window=hamming(PN).';

window\_matrix=repmat(window,length(signal(:,1)),1);

signal=signal.\*window\_matrix;

fsingal=fft(signal,dopp\_FFT\_size,2);

psdx=(1/length(fsingal)).\*abs(fsingal).^2;

psdx=fftshift(psdx,2);

psd\_rng\_vel(index+1,:,:)=10\*log10(psdx);

plot\_image(:,:)=psd\_rng\_vel(index+1,:,:);

psd\_vel(index+1,:)=mean(plot\_image,1);

figure(2)

surf(-vel\_vector,Rng,plot\_image)

xlim([-6 6])

ylim([0 100])

colorbar('EastOutside');

shading interp

colormap(jet)

caxis([0 100])

view(2)

xlabel('Velocity (m/s)')

ylabel('Range (m)')

end

wfall\_t=linspace(0,Tsweep\*numberSweeps,pro\_vel\_prof);

figure(3)

surf(-vel\_vector,wfall\_t,psd\_vel)

xlim([-6 6])

ylim([0 10])

bar=colorbar('EastOutside');

colormap(jet)

shading interp

caxis([0 100])

view(2)

xlabel('Velocity (m/2)')

ylabel('Range (m)')

%%

Fc=5.8e9;

Sps=512;

Fs=2\*Bw;

t=0:1/Fs:Tsweep-1/Fs;

y=chirp(t,0,Tsweep,Bw,'linear',0);

figure(4)

subplot(211)

plot(t,real(y));

xlabel('Time(s)')

ylabel('Amplitude')

title('FMCW Signal')

subplot(212)

spectrogram(y,1024,0,1024,Fs)

xlabel('Frequency(MHz)')

ylabel('Time(s)')

title('Spectrum of FMCW')

%%

f1=10e6;

f2=50e6;

t=0:1/Fs:Tsweep-1/Fs;

bits=randi([0 1],numel(t)/10,1);

bits=repmat(bits,1,10).';

bits=bits(:);

y=zeros(size(bits));

for ii=1:length(y)

if bits(ii)==1

y(ii)=exp(1j\*2\*pi\*f1\*t(ii));

else

y(ii)=exp(1j\*2\*pi\*f2\*t(ii));

end

end

figure(5)

subplot(211)

plot(t,bits);

xlabel('Time(s)')

ylabel('Amplitude')

title('Digital Signal')

xlim([0 2.5e-6])

ylim([0 1])

subplot(212)

plot(t,real(y));

xlabel('Time(s)')

ylabel('Amplitude')

title('Digital Signal')

xlim([0 2.5e-6])

ylim([0 1])