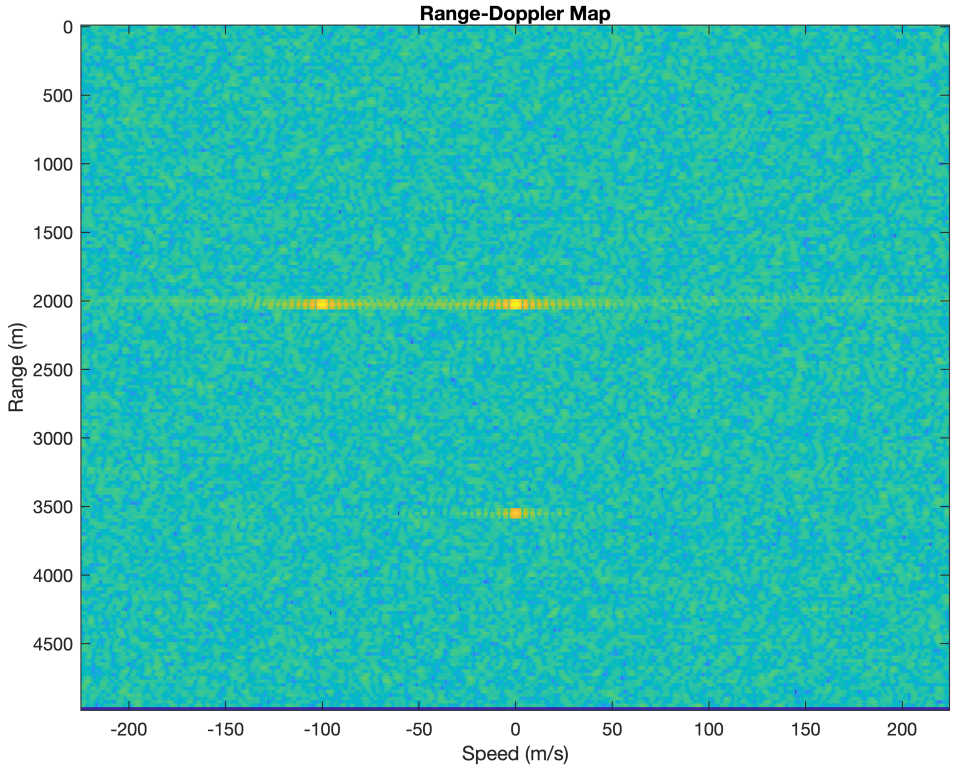
```
%Jesus Martinez A01283785
%Genaro Gallardo A01382459
%Carlos López A01283785
%Claudia González A01411506
%Mario Veccio A00826824

%%% Range-Doppler response, two radar systems

%%% Pulse radar systems

%% Range Doppler response using matched filter
load RangeDopplerExampleData;

% Create a range-Doppler response object
response = phased.RangeDopplerResponse('DopplerFFTLengthSource','Property', ...
   'DopplerFFTLength',RangeDopplerEx_MF_NFFTDOP, ...
   'SampleRate',RangeDopplerEx_MF_Fs,'DopplerOutput','Speed', ...
   'OperatingFrequency',RangeDopplerEx_MF_Fc);

% Calculate the range-Doppler response.
[resp,rng_grid,dop_grid] = response(RangeDopplerEx_MF_X, ...
    RangeDopplerEx_MF_Coeff);

% Plot the range-Doppler response.
figure(1)
imagesc(dop_grid,rng_grid,mag2db(abs(resp)));
xlabel('Speed (m/s)');
ylabel('Range (m)');
title('Range-Doppler Map');
```



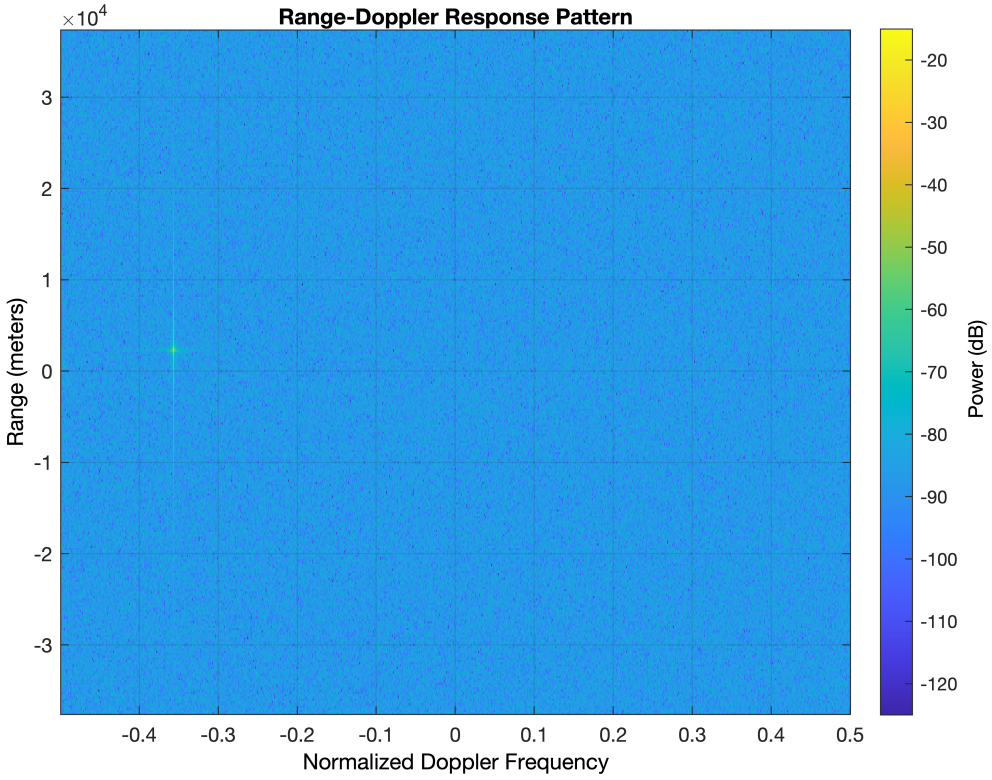

```
%% Estimate Doppler and range from range-Doppler response.

% Create a range-Doppler response object.
hrdresp = phased.RangeDopplerResponse(...
   'RangeMethod','FFT',...
   'PropagationSpeed',RangeDopplerEx_Dechirp_PropSpeed,...
   'SampleRate',RangeDopplerEx_Dechirp_Fs,...
   'DechirpInput',true,...
   'SweepSlope',RangeDopplerEx_Dechirp_SweepSlope);

% Obtain the range-Doppler response data.
[resp,rng_grid,dop_grid] = step(hrdresp,...
   RangeDopplerEx_Dechirp_X,RangeDopplerEx_Dechirp_Xref);

% Estimate the range and Doppler by finding the location of the maximum response.
[x_temp,idx_temp] = max(abs(resp));
[~,dop_idx] = max(x_temp);
rng_idx = idx_temp(dop_idx);
dop_est = dop_grid(dop_idx) % Doppler shift
```

```
dop_est = -712.8906
```

```
rng_est = rng_grid(rng_idx) % Distance of target
```

```
rng_est = 2250
```

```
%%% FMCW Radar System

%% Range Doppler Response of FMCW Signal

% Create a range-Doppler response object.
hrdresp = phased.RangeDopplerResponse(...
    'RangeMethod','FFT',...
    'PropagationSpeed',RangeDopplerEx_Dechirp_PropSpeed,...
    'SampleRate',RangeDopplerEx_Dechirp_Fs,...
    'DechirpInput',true,...
    'SweepSlope',RangeDopplerEx_Dechirp_SweepSlope);

% Plot the range-Doppler response.
figure(2)
plotResponse(hrdresp,...
   RangeDopplerEx_Dechirp_X,RangeDopplerEx_Dechirp_Xref,...
    'Unit','db','NormalizeDoppler',true)
```




```
%%% Range-Speed response pattern of target

%%
% Initial settings
```

```
antenna = phased.IsotropicAntennaElement(...
    'FrequencyRange',[5e9 15e9]);
transmitter = phased.Transmitter('Gain',20,'InUseOutputPort',true);
fc = 10e9;
target = phased.RadarTarget('Model','Nonfluctuating',...
    'MeanRCS',1,'OperatingFrequency',fc);
txloc = [0;0;0];
tgtloc = [5000;5000;10];
antennaplatform = phased.Platform('InitialPosition',txloc);
targetplatform = phased.Platform('InitialPosition',tgtloc);
[tgtrng,tgtang] = rangeangle(targetplatform.InitialPosition,...
    antennaplatform.InitialPosition);

% Creating rectangular pulse
waveform = phased.RectangularWaveform('PulseWidth',2e-6,...
    'OutputFormat','Pulses','PRF',1e4,'NumPulses',1);
c = physconst('LightSpeed');
maxrange = c/(2*waveform.PRF);
SNR = npwgnthresh(1e-6,1,'noncoherent');
lambda = c/target.OperatingFrequency;
maxrange = c/(2*waveform.PRF);
tau = waveform.PulseWidth;
Ts = 290;
dbterm = db2pow(SNR - 2*transmitter.Gain);
Pt = (4*pi)^3*physconst('Boltzmann')*Ts/tau/target.MeanRCS/lambda^2*maxrange^4*dbterm;

% Set the peak transmit power to the value obtained from the radar equation.
transmitter.PeakPower = Pt;

radiator = phased.Radiator(...
    'PropagationSpeed',c,...
    'OperatingFrequency',fc,'Sensor',antenna);
channel = phased.FreeSpace(...
    'PropagationSpeed',c,...
    'OperatingFrequency',fc,'TwoWayPropagation',false);
collector = phased.Collector(...
    'PropagationSpeed',c,...
    'OperatingFrequency',fc,'Sensor',antenna);
receiver = phased.ReceiverPreamp('NoiseFigure',0,...
    'EnableInputPort',true,'SeedSource','Property','Seed',2e3);

numPulses = 25;
rx_puls = zeros(100,numPulses);

for n = 1:numPulses
    wf = waveform();
    [wf,txstatus] = transmitter(wf);
    wf = radiator(wf,tgtang);
    wf = channel(wf,txloc,tgtloc,[0;0;0],[0;0;0]);
    wf = target(wf);
```

```
    wf = channel(wf,tgtloc,txloc,[0;0;0],[0;0;0]);
    wf = collector(wf,tgtang);
    rx_puls(:,n) = receiver(wf,~txstatus);
end

rangedoppler = phased.RangeDopplerResponse(...
    'RangeMethod','Matched Filter',...
    'PropagationSpeed',c,...
    'DopplerOutput','Speed','OperatingFrequency',fc);
figure(3)
plotResponse(rangedoppler,rx_puls,getMatchedFilter(waveform))
```