 Received SNR level as a function of range

Chart

Description automatically generated

Maximum detectable range of the system for the different considered targets.

tgtrng1 = 1.4555e+04 = 14.55 km

tgtrng2 = 2.5883e+04 = 25.88 km

Obtain the range profile of this scenario for multiple iterations. What is the effect of the Swerling model?

If the target velocity is low compared to the observation time, then the target can be non-moving. Scanning radar systems are fast-acting devices that send out a signal that sweeps past the target in a short period of time. The motion of the target is only seen from scan-to-scan, not from within a scan.

In Swerling model, target whose magnitude of the backscattered signal is relatively constant during the dwell time. It varies according to a Chi-square probability density function with two degrees of freedom (m = 1). The radar cross section is constant from pulse-to-pulse but varies independently from scan to scan. The density of probability of the RCS is given by the Rayleigh-Function.

Chart, histogram

Description automatically generated

Obtain the histograms for the two possible hypotheses.

Chart, histogram

Description automatically generated

Analyse how the threshold level affects to the false alarm and detection probabilities of the system

Chart

Description automatically generated with medium confidence

Obtain the Receiver Operating Characteristics (ROC) curve (Pd vs. Pfa)

Chart

Description automatically generated

How is the system performance affected if we consider now a target with a RCS of 0.05 square meters?

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart

Description automatically generated with medium confidenceChart

Description automatically generated

Part 1 -A

wavelen = 0.3; % Wavelength (m)

pwidth = 5e-07; % Pulse width (s)

sysloss = 2; % System losses (dB)

noisetemp = 290; % Noise temperature (K)

rcs1=0.1;

rcs2 = 1; % Target radar cross section (m^2)

gain = 30; % Gain (dB)

tgtrng = 25000; % Target range (m)

pkpow = 5000; % Peak transmit power (W)

for i = 500:35000

snr1(i) = radareqsnr(wavelen, i, pkpow, pwidth,'rcs', rcs1, 'gain', ...

gain, 'loss', sysloss, 'Ts', noisetemp);

end

for i = 500:35000

snr2(i) = radareqsnr(wavelen, i, pkpow, pwidth,'rcs', rcs2, 'gain', ...

gain, 'loss', sysloss, 'Ts', noisetemp);

end

figure

plot(snr1(500:end))

hold on

plot(snr2(500:end))

hold off

title('SNR Vs Distance');

xlabel('Distance(m)');

ylabel('SNR(dB)');

legend("RCS =0.1m^2","RCS=1m^2");

Part 1-B

wavelen = 0.3; % Wavelength (m)

pwidth = 5e-07; % Pulse width (s)

sysloss = 2; % System losses (dB)

noisetemp = 290; % Noise temperature (K)

rcs1=0.1;

rcs2 = 1; % Target radar cross section (m^2)

gain = 30; % Gain (dB)

pkpow = 5000; % Peak transmit power (W)

snr = 16; % SNR (dB)

tgtrng1 = radareqrng(wavelen, snr, pkpow, pwidth,'rcs', rcs1, 'gain', ...

gain, 'loss', sysloss, 'Ts', noisetemp)

tgtrng2 = radareqrng(wavelen, snr, pkpow, pwidth,'rcs', rcs2, 'gain', ...

gain, 'loss', sysloss, 'Ts', noisetemp)

Part 2

clear all,

close all,

clc

waveform = phased.RectangularWaveform('PulseWidth',1e-6,'PRF',5e3,'OutputFormat','Pulses','NumPulses',1);

antenna = phased.IsotropicAntennaElement('FrequencyRange',[1e9 10e9]);

target = phased.RadarTarget('Model','Swerling1','MeanRCS',0.5, 'PropagationSpeed',physconst('LightSpeed'),'OperatingFrequency',4e9);

antennaplatform = phased.Platform('InitialPosition',[0;0;0],'Velocity',[0;0;0]);

targetplatform = phased.Platform('InitialPosition',[6895.2; 0; 0], 'Velocity',[-15;-10;0]);

[tgtrng,tgtang] = rangeangle(targetplatform.InitialPosition,antennaplatform.InitialPosition);

Pd = 0.9;

Pfa = 1e-6;

numpulses = 10;

SNR = albersheim(Pd,Pfa,10);

%target\_model="noncoherent";

target\_model = 'Swerling1';

maxrange = 1.5e4;

lambda = physconst('LightSpeed')/target.OperatingFrequency;

tau = waveform.PulseWidth;

Ts = 290;

rcs = 0.05;

tgt\_range = 6895.2;

Gain = 20;

dbterm = db2pow(SNR - 2\*Gain);

Pt = (4\*pi)^3\*physconst('Boltzmann')\*Ts/tau/rcs/lambda^2\*maxrange^4\*dbterm;

transmitter = phased.Transmitter('PeakPower',10e3,'Gain',20,'LossFactor',0, ...

'InUseOutputPort',true,'CoherentOnTransmit',true);

radiator = phased.Radiator('Sensor',antenna,...

'PropagationSpeed',physconst('LightSpeed'),'OperatingFrequency',4e9);

collector = phased.Collector('Sensor',antenna,...

'PropagationSpeed',physconst('LightSpeed'),'Wavefront','Plane', ...

'OperatingFrequency',4e9);

receiver = phased.ReceiverPreamp('Gain',20,'NoiseFigure',2, ...

'ReferenceTemperature',290,'SampleRate',1e6, ...

'EnableInputPort',true,'SeedSource','Property','Seed',2022);

channel = phased.FreeSpace(...

'PropagationSpeed',physconst('LightSpeed'), ...

'OperatingFrequency',4e9,'TwoWayPropagation',false, ...

'SampleRate',1e6);

ite = 5000;

T = 1/waveform.PRF;

% Get antenna position

txpos = antennaplatform.InitialPosition;

% Allocate array for received echoes

rxsig\_H0\_collection = zeros(waveform.SampleRate\*T, ite);

rxsig\_H1\_collection = zeros(waveform.SampleRate\*T, ite);

for ii = 1:ite

rxsig\_H0 = zeros(waveform.SampleRate\*T, numpulses);

rxsig\_H1 = zeros(waveform.SampleRate\*T, numpulses);

for n = 1:numpulses

% Update the target position

[tgtpos,tgtvel] = targetplatform(T);

% Get the range and angle to the target

[tgtrng,tgtang] = rangeangle(tgtpos,txpos);

% Generate the pulse

sig = waveform();

% Transmit the pulse. Output transmitter status

[sig,txstatus] = transmitter(sig);

% Radiate the pulse toward the target

sig = radiator(sig,tgtang);

% Propagate the pulse to the target in free space

sig = channel(sig,txpos,tgtpos,[0;0;0],tgtvel);

% Reflect the pulse off the target

if strcmp(target\_model, 'Swerling1')

sig\_H1 = target(sig, true);

else

sig\_H1 = target(sig);

end

% Propagate the echo to the antenna in free space

sig\_H0 = channel(sig, tgtpos, txpos, tgtvel, [0;0;0]);

sig\_H1 = channel(sig\_H1,tgtpos, txpos, tgtvel, [0;0;0]);

% Collect the echo from the incident angle at the antenna

sig\_H0 = collector(sig\_H0, tgtang);

sig\_H1 = collector(sig\_H1, tgtang);

% Receive the echo at the antenna when not transmitting

rxsig\_H0(:,n) = receiver(sig\_H0, ~txstatus);

rxsig\_H1(:,n) = receiver(sig\_H1, ~txstatus);

end

rxsig\_H0\_collection(:, ii) = pulsint(rxsig\_H0, 'noncoherent');

rxsig\_H1\_collection(:, ii) = pulsint(rxsig\_H1, 'noncoherent');

end

t = unigrid(0,1/receiver.SampleRate,T, '[)');

rangegates = physconst('LightSpeed')\*t/2;

tgt\_bin = find((rangegates - tgt\_range) == min(abs(rangegates - tgt\_range)));

figure

plot(rangegates/1e3, rxsig\_H1\_collection(:,1))

hold on

title("Range profile (" + target\_model + " model)")

xlabel('range (km)')

ylabel('Power')

xline(tgtrng/1e3, '--r')

axis tight

hold off

%% Part 2b

nbins = 30;

figure

h1 = histogram(rxsig\_H1\_collection(tgt\_bin,:) , nbins, 'Normalization', 'pdf', ...

'HandleVisibility', 'on');

hold on

h0 = histogram(rxsig\_H0\_collection(tgt\_bin,:) , nbins, 'Normalization', 'pdf', ...

'HandleVisibility', 'on');

xlabel("Amplitude")

ylabel("Probability")

title("Histograms for H0 and H1")

%% Part 2c

% H0

pd = fitdist(rxsig\_H0\_collection(tgt\_bin, :)', 'normal');

x\_interference = min(rxsig\_H0\_collection(tgt\_bin, :)):(10^(-9)):max(rxsig\_H0\_collection(tgt\_bin,:));

y\_interference = pdf(pd, x\_interference);

mu\_interference = pd.mu;

sigma\_interference = pd.sigma;

% H1

pd = fitdist(rxsig\_H1\_collection(tgt\_bin,:)','normal');

x\_target = min(rxsig\_H1\_collection(tgt\_bin, :)):(10^(-9)):max(rxsig\_H1\_collection(tgt\_bin,:));

y\_target = pdf(pd,x\_target);

my\_target = pd.mu;

sigma\_target = pd.sigma;

% PlotPDF

hold on

plot(x\_target, y\_target, 'b', 'LineWidth',2)

plot(x\_interference, y\_interference, 'r', 'LineWidth',2)

title("Histograms for H0 and H1 (RC = " + rcs + "m^2)")

legend(["H1" "H0" "Fit Gaussian PDF to H1" "Fit Gaussian PDF to H0"])

%% Part 2d

threshold = (0.2\*10^(-7)):(10^(-9)):(16\*(10^(-7)));

Pfa = 1 - cdf('normal', threshold, mu\_interference, sigma\_interference);

Pd = 1 - cdf('normal', threshold, my\_target, sigma\_target);

figure

plot(threshold, Pfa, threshold, Pd, 'LineWidth', 2)

xlabel("Threshold level")

ylabel("Probability")

title("Threshold vs. detection and false alarm rates")

legend(["P\_{fa}" "P\_d"])

figure

plot(Pfa,Pd, 'Linewidth',2)

title("ROC curve (RCS = " + rcs + " m^2)")

xlabel("P\_{fa}")

ylabel("F\_d")