Matlab code 5.7: Matlab file “Figure5-19ab.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 5.19 (a-b)

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% Esplanorteta60.mat

% planorteta60\_2\_xyout.mat

clear all

close all

c=.3; % speed of light

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%Find spatial resolutions

BWx = 12;

BWy = 16;

M = 32;

N = 64;

fc = 6;

phic = 0;

% Image resolutions

dx = BWx/M;

dy = BWy/N;

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

%Find resoltions in freq and angle

df = c/(2\*BWx);

dk = 2\*pi\*df/c;

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)];

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];

% Load the backscattered data

load Esplanorteta60

load planorteta60\_2\_xyout

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ POST PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ISAR = fftshift(fft2(Es.’));

ISAR = ISAR/M/N;

%---Figure 5.19(c)---------------------------------------------------–

h = figure;

matplot2(X(32:-1:1),Y,ISAR,20);

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);%grid on;

h = line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’,’

MarkerSize’,3);

%windowing;

w = hamming(M)\*hamming(N).’;

Ess = Es.\*w;

%zero padding;

Enew = Ess;

Enew(M\*4,N\*4) = 0;

XX = X(1):dx/4:X(1)+dx/4\*(4\*M-1);

YY = Y(1):dy/4:Y(1)+dy/4\*(4\*N-1);

% ISAR image formatiom

ISARnew = fftshift(fft2(Enew.’));

ISARnew = ISARnew/M/N;

%---Figure 5.19(d)---------------------------------------------------–

load planorteta60\_2\_xyout.mat

h = figure;

matplot2(XX(4\*M:-1:1),YY,abs(ISARnew),20);

colormap(1-gray);

colorbar

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’,’Marke

rSize’,3);

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);

Matlab code 5.8: Matlab file “Figure5-19cd.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 5.19 (c-d)

%----------------------------------------------------------------

% This file requires the following files to be present in the same

% directory:

%

% ucak.mat

clear all

close all

c=.3; % speed of light

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%Find spatial resolutions

BWx = 18;

BWy = 16;

M = 64;

N = 64;

fc = 8;

phic = 0;

% Image resolutions

dx = BWx/M;

dy = BWy/N;

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

%Find resoltions in freq and angle

df = c/(2\*BWx);

dk = 2\*pi\*df/c;

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)];

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];

K = 2\*pi\*F/c;

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FORM RAW BACKSCATTERED DATA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

load ucak

l = length(xx);

Es = zeros(M,N);

for m=1:l;

Es=Es+1.0\*exp(j\*2\*K’\*(cos(PHI)\*xx(m)+sin(PHI)\*yy(m)));

end

%\_\_\_\_\_ POST PROCESSING OF ISAR (Small BW Small

angle)\_

ISAR = fftshift(fft2(Es.’));

ISAR = ISAR/M/N;

h = figure;

matplot2(X(M:-1:1),Y,ISAR,25);

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);%grid on;

colormap(1-gray);

%windowing;

w = hamming(M)\*hamming(N).’;

Ess = Es.\*w;

%-------------zero padding with 4 times----------

Enew = Ess;

Enew(M\*4,N\*4) = 0;

% ISAR image formatiom

ISARnew = fftshift(fft2(Enew.’));

ISARnew = ISARnew/M/N;

h = figure;

matplot2(X(M:-1:1),Y,ISARnew,25);

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);%grid on;

2-

Matlab code 7.2: Matlab file ‘Figure7-9thru7-13.m’

%----------------------------------------------------------------

% This code can be used to generate Figure 7.9 thru 7.13

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% Es60.mat

% planorteta60\_2\_xyout.mat

clear all

close all

c = .3; % speed of light

fc = 6; % center frequency

phic = 0\*pi/180; % center of azimuth look angles

thc = 90\*pi/180; % center of elevation look angles

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BWx = 12; % range extend

M = 32; % range sampling

BWy = 16; % xrange extend

N = 64; % xrange sampling

dx = BWx/M; % range resolution

dy = BWy/N; % xrange resolution

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

XX = -dx\*M/2:dx/4:-dx\*M/2+dx/4\*(4\*M-1);

YY = -dy\*N/2:dy/4:-dy\*N/2+dy/4\*(4\*N-1);

%Find resoltions in freq and angle

df = c/(2\*BWx); % frequency resolution

dk = 2\*pi\*df/c; % wavenumber resolution

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);% azimuth resolution

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)]; % frequency vector

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];% azimuth vector

K = 2\*pi\*F/c; % wavenumber vector

dk = K(2)-K(1); % wavenumber resolution

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_GET THE DATA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

load Es60 % load E-scattered

load planorteta60\_2\_xyout.mat % load plane outline

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MATCHING PURSUIT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

collectedData = zeros(200,3); %initilize scattering center

info

ES = Es;

Power1 = sum(sum(Es).^2); % initial power of the data

axisX = X(1):dx/4:X(32);

axisY = Y(1):dy/4:Y(64);

cosPhi = cos(PHI);

sinPhi = sin(PHI);

for N = 1:250; % extract 250 scattering centers

Amax = 0;

p1Max = zeros(size(ES));

for Xn = axisX

for Yn = axisY

p1 = exp(-j\*2\*K.’\*(cosPhi.\*Xn+sinPhi.\*Yn));

A = sum(sum(ES.\*p1))/(size(ES,1)\*size(ES,2));

if A > Amax

Amax = A;

collectedData(N,1:3) = [A Xn Yn];

p1Max = conj(p1);

end

end

end

ES = ES-(Amax.\*p1Max);

end

%------–Field Reconsctruction---------–

Esr = zeros(32,64);

for N = 1:250

A = collectedData(N,1)

x1 = collectedData(N,2);

y1 = collectedData(N,3);

Esr = Esr+A\*exp(j\*2\*K.’\*(cosPhi.\*x1+sinPhi.\*y1));

end

%---Figure 7.9(a)------------------------------------------------

%---SCATTERING CENTER INFO DISPLAY-------------------------------

load planorteta60\_2\_xyout.mat

SSs = collectedData;

h = figure;

plot(abs(SSs(1:250,1)),’square’,’MarkerSize’,4,’MarkerFaceCo

lor’, [0, 0, 1]);

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

xlabel(‘Scattering Center #’);

ylabel(‘Amplitude [mV/m]’);

%---Figure 7.9(b)------------------------------------------------

h = figure;

hold

for m=1:150

t = round(abs(SSs(m,1))\*20/abs(SSs(1,1)))+1

plot(-SSs(m,2),SSs(m,3),’o’,’MarkerSize’,t,’MarkerFaceColor’,

[0, 0, 1]);

end

hold

line( -xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’);

axis([min(X) max(X) min(Y) max(Y)])

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

xlabel(‘Range [m]’);

ylabel(‘X-Range [m]’);

title(‘Locations of scattering centers with relative

amplitudes ‘)

%–ISAR IMAGE COMPARISON------------------------------------------

Enew = Es;

Enew(M\*4,N\*4) = 0;

ISARorig = fftshift(fft2(Enew));

ISARorig = ISARorig/M/N;

h = figure;

matplot(X,Y,abs(ISARorig(4\*M:-1:1,:).’),20);

colorbar;

colormap(1-gray);

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’);

xlabel(‘Range [m]’);

ylabel(‘X-Range [m]’);

title(‘Original ISAR image’)

Enew = Esr;

Enew(M\*4,N\*4) = 0;

ISARrec = fftshift(fft2(Enew));

ISARrec = ISARrec.’/M/N; % reconstructed ISAR image

%---Figure 7.13 ------------------------------------------------–

h = figure;

matplot(X,Y,abs(ISARrec(:,4\*M:-1:1)),20);

colorbar;

colormap(1-gray);

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’);

xlabel(‘Range [m]’);

ylabel(‘X-Range [m]’);

title(‘Reconstructed ISAR image’)

%-------FIELD COMPARISON---------------------

%-------------------------------------------

h = figure;

matplot(F,PHI\*180/pi,abs((Es.’)),20);

colorbar;

colormap(1-gray)

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

ylabel(‘Angle [Degree]’);

xlabel(‘Frequency [GHz]’);

title(‘Original back-scattered field’)

%---Figure 7.10 ------------------------------------------------–

h = figure;

matplot(F,PHI\*180/pi,abs((Esr.’)),20);

colorbar;

colormap(1-gray)

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

ylabel(‘Angle [Degree]’);

xlabel(‘Frequency [GHz]’);

title(‘Reconstructed back-scattered field’)

%------–RECONSTRUCT THE FIELD PATTERN x10-------------

Esr = zeros(320,640);

k = K;

kk = k(1):(k(32)-k(1))/319:k(32);

pp = PHI(1):(PHI(64)-PHI(1))/639:PHI(64);

csP = cos(pp);

snP = sin(pp);

for N = 1:250

A = collectedData(N,1);

x1 = collectedData(N,2);

y1 = collectedData(N,3);

Esr = Esr+A\*exp(j\*2\*kk.’\*(csP.\*x1+snP.\*y1));

end

%---Figure 7.11 ------------------------------------------------–

h = figure;

matplot(F,PHI\*180/pi,abs((Esr.’)),20);

colorbar;

colormap(1-gray)

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

ylabel(‘Angle [Degree]’);

xlabel(‘Frequency [GHz]’);

title(‘Reconstructed field (x10 upsampled)’)

%---Figure 7.12(a)------------------------------------------------

nn = 7;

h = figure;

plot(PHI\*180/pi,abs(Es(nn,:)),’k-.\*’,’MarkerSize’,8,’LineWi

dth’,2);

hold;

plot(pp\*180/pi,abs(Esr(10\*(nn-1)+1,:)),’k-’,’LineWidth’,2);

hold;

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

xlabel(‘PHI [Degree]’); ylabel(‘Scat. field [V/m]’);

tt = num2str(F(nn));

ZZ = [‘@ f = ‘ tt ‘ GHz’];

axis([PHI(1)\*180/pi PHI(64)\*180/pi 0 0.35])

title(ZZ);

drawnow;

legend(‘with brute force computation’,’with scattering

centers’)

%---Figure 7.12(b)------------------------------------------------

nn = 4;

h = figure;

plot(F,abs(Es(:,nn)),’k-.\*’,’MarkerSize’,8,’LineWidth’,2);

hold;

plot(kk\*c/2/pi,abs(Esr(:,10\*(nn-1)+1)),’k-’,’LineWidth’,2);

hold;

set(gca,’FontName’, ‘Arial’, ‘FontSize’,14,’FontWeight’,

‘Bold’);

xlabel(‘Frequency [GHz]’); ylabel(‘Scat. field [V/m]’);

tt = num2str(PHI(nn));

ZZ = [‘@ PHI = ‘ tt ‘ Deg.’];

title(ZZ);

drawnow;

axis([F(1) F(32) 0 0.5])

legend(‘with brute force computation’,’with scattering

centers’)

3

Matlab code 5.2: Matlab file “Figure5-10ab.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 5.10 (a-b)

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% Esplanorteta60.mat

% planorteta60\_2\_xyout.mat

clear all

close all

c=.3; % speed of light

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%Find spatial resolutions

BWx = 12;

BWy = 16;

M = 32;

N = 64;

fc = 6;

phic=0;

dx = BWx/M;

dy = BWy/N;

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

%Find resoltions in freq and angle

df = c/(2\*BWx);

dk = 2\*pi\*df/c;

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)];

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];

k = 2\*pi\*F/c;

% Load the backscattered data

load Esplanorteta60

load planorteta60\_2\_xyout

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ POST PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ISAR = fftshift(fft2(Es.’));

ISAR = ISAR/M/N;

%---Figure 5.10(a)---------------------------------------------------–

matplot2(X(32:-1:1),Y,ISAR,20);

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,14,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);%grid on;

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’,’Marke

rSize’,3);

%zero padding;

Enew = Es;

Enew(M\*4,N\*4) = 0;

XX = X(1):dx/4:X(1)+dx/4\*(4\*M-1);

YY = Y(1):dy/4:Y(1)+dy/4\*(4\*N-1);

% ISAR image formatiom

ISARnew = fftshift(fft2(Enew.’));

ISARnew = ISARnew/M/N;

figure;

%---Figure 5.10(b)---------------------------------------------------–

matplot2(XX(4\*M:-1:1),YY,abs(ISARnew),20);

colormap(1-gray);

colorbar

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’,’Marke

rSize’, 3);

set(gca,’FontName’, ’Arial’, ’FontSize’,14,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’);%grid on;

Matlab code 5.3: Matlab file “Figure5-10cd.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 5.10 (c-d)

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% Esairbus.mat

% airbusteta80\_2\_xyout.mat

clear all

close all

c=.3; % speed of light

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%Find spatial resolutions

BWx = 80;

BWy = 66;

M = 32;

N = 64;

fc = 4;

phic = 0;

dx = BWx/M;

dy = BWy/N;

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

%Find resoltions in freq and angle

df = c/(2\*BWx);

dk = 2\*pi\*df/c;

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)];

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];

k = 2\*pi\*F/c;

% Load the backscattered data

load Esairbus

load airbusteta80\_2\_xyout

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ POST PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ISAR = fftshift(fft2(Es.’));

ISAR = ISAR/M/N;

%---Figure 5.10(c)---------------------------------------------------–

matplot2(X(32:-1:1),Y,ISAR,30); colormap(1-gray); colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,14,’FontWeight’,

’Bold’);

xlabel(’Range [m]’); ylabel(’Cross - range [m]’);%grid on;

colormap(1-gray);

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’);

%zero padding with 4 times;

Enew = Es;

Enew(M\*4,N\*4) = 0;

figure;

% ISAR image formatiom

ISARnew = fftshift(fft2(Enew.’));

ISARnew = ISARnew/M/N;

%ISARnew(1,1)=2.62

load airbusteta80\_2\_xyout.mat;

%---Figure 5.10(d)---------------------------------------------------–

matplot2(X(32:-1:1),Y,ISARnew,30);

colormap(1-gray);

line(-xyout\_xout,xyout\_yout,’Color’,’k’,’LineStyle’,’.’);

set(gca,’FontName’, ’Arial’, ’FontSize’,14,’FontWeight’,

’Bold’);

xlabel(’Range [m]’); ylabel(’Cross - range [m]’);

5

Matlab code 4.4: Matlab file “Figure4-15.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 4.15

%----------------------------------------------------------------

% This file requires the following files to be present in the same

% directory:

%

% PLANORPHI45\_Es.mat

% planorphi45\_2\_xyout.mat

clear all

close all

c = .3; % speed of light

fc = 6; % center frequency

phic = 45\*pi/180; % center of azimuth look angles

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BWx = 13; % range extend

M = 32; % range sampling

BWy = 13; % xrange extend

N = 64; % xrange sampling

dx = BWx/M; % range resolution

dy = BWy/N; % xrange resolution

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);

Y = -dy\*N/2:dy:dy\*(N/2-1);

%Find resolutions in freq and angle

df = c/(2\*BWx); % frequency resolution

dk = 2\*pi\*df/c; % wavenumber resolution

kc = 2\*pi\*fc/c;

dphi = pi/(kc\*BWy);% azimuth resolution

%Form F and PHI vectors

F = fc+[-df\*M/2:df:df\*(M/2-1)]; % frequency vector

PHI = phic+[-dphi\*N/2:dphi:dphi\*(N/2-1)];% azimuth vector

K=2\*pi\*F/c; % wanenumber vector

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_GET THE DATA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

load PLANORPHI45\_Es.mat; % load E-scattered

load planorphi45\_2\_xyout.mat; % load target outline

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ POST PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%windowing;

w=hanning(M)\*hanning(N).’;

Ess=Es.\*w;

%zero padding;

Enew=Ess;

Enew(M\*4,N\*4)=0;

% ISAR image formatiom

ISARnew=fftshift(ifft2(Enew));

h=figure;

matplot2(X,Y,abs(ISARnew),22); % form the image

colormap(1-gray);colorbar

line(xyout\_yout,xyout\_xout,’LineWidth’,.25,’LineStyle’,’.’,’Co

lor’,’k’);

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’); ylabel(’Cross-Range [m]’);

6

Matlab code 4.6: Matlab file “Figure4-20.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 4.20

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% Esairbus.mat

% airbusteta80\_2\_xyout.mat

clear all

close all

c = .3; % speed of light

fc = 4; % center frequency

phic = 0\*pi/180; % center of azimuth look angles

thc = 80\*pi/180; % center of elevation look angles

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_PRE PROCESSING OF ISAR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BWx = 80; % range extend

M = 32; % range sampling

BWy = 66; % x-range extend

N = 64; % x-range sampling

dx = BWx/M; % range resolution

dy = BWy/N; % xrange resolution

% Form spatial vectors

X = -dx\*M/2:dx:dx\*(M/2-1);% range vector

XX = -dx\*M/2:dx/4:-dx\*M/2+dx/4\*(4\*M-1); % range vector (4x

upsampled)

Y = -dy\*N/2:dy:dy\*(N/2-1); % x-range vector

YY = -dy\*N/2:dy/4:-dy\*N/2+dy/4\*(4\*N-1); % range vector (4x

upsampled)

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_GET THE DATA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

load Esairbus % load E-scattered

load airbusteta80\_2\_xyout.mat % load target outline

% ISAR 4x UPSAMPLED-------------------

%zero padding;

Enew = Es;

Enew(M\*4,N\*4)=0;

% ISAR image formatiom

h = figure;

ISARnew = fftshift(ifft2(Enew));

matplot2(X,Y,abs(ISARnew.’),30); % form the image

colormap(1-gray);colorbar

line(-xyout\_xout,xyout\_yout,’LineWidth’,.25,’LineStyle’,’.’,’Color

’,’k’);

set(gca,’FontName’, ’Arial’, ’FontSize’,14,’FontWeight’,

’Bold’);

xlabel(’Range [m]’); ylabel(’Cross-Range [m]’)

6

Matlab code 4.9: Matlab file “Figure4-26thru4-28.m”

%----------------------------------------------------------------

% This code can be used to generate Figure 4.26 thru 4.28

%----------------------------------------------------------------

% This file requires the following files to be present in the

same

% directory:

%

% fighterSC.mat

clear all

close all

c = .3; % speed of light

fc = 8; % center frequency

fMin = 6; % lowest frequency

fMax = 10; % highest frequency

phic = 0\*pi/180; % center of azimuth look angles

phiMin = -30\*pi/180; % lowest angle

phiMax = 30\*pi/180; % highest angle

%------------------------------------------------

% WIDE BW AND WIDE ANGLE ISAR

%------------------------------------------------

% B- POLAR REFORMATTING

%------------------------------------------------

nSampling = 1500; % sampling number for integration

% Define Bandwidth

f = fMin:(fMax-fMin)/(nSampling):fMax;

k = 2\*pi\*f/.3;

kMax = max(k);

kMin = min(k);

% Define Angle

phi = phiMin:(phiMax-phiMin)/(nSampling):phiMax;

kc = (max(k)+min(k))/2;

kx=k.’\*cos(phi);

ky=k.’\*sin(phi);

kxMax = max(max(kx));

kxMin = min(min(kx));

kyMax = max(max(ky));

kyMin = min(min(ky));

MM=4; % up sampling ratio

clear kx ky;

kxSteps = (kxMax-kxMin)/(MM\*(nSampling+1)-1);

kySteps = (kyMax-kyMin)/(MM\*(nSampling+1)-1);

kx = kxMin:kxSteps:kxMax; Nx=length(kx);

ky = kyMin:kySteps:kyMax; Ny=length(ky);

kx(MM\*(nSampling+1)+1) = 0;

ky(MM\*(nSampling+1)+1) = 0;

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ FORM RAW BACKSCATTERED DATA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%load scattering centers

load fighterSC

l = length(xx);

%form backscattered E-field from scattering centers

Es = zeros((nSampling+1),(nSampling+1));

for n=1:length(xx);

Es = Es+exp(-j\*2\*k.’\*cos(phi)\*xx(n)).\*exp(-j\*2\*k.’\*sin(phi)\*

yy(n));

end

%---Figure 4.24---------------------------------------------------–

matplot2(f,phi\*180/pi,Es,40);

colormap(1-gray); colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Frequency [GHz]’);

ylabel(’Angle [Degree]’);

newEs = zeros(MM\*(nSampling+1)+1,MM\*(nSampling+1)+1);

t = 0;

v = 0;

for tmpk = k

t = t+1;

v = 0;

for tmpPhi = phi

v = v+1;

tmpkx = tmpk\*cos(tmpPhi);

tmpky = tmpk\*sin(tmpPhi);

indexX = floor((tmpkx-kxMin)/kxSteps)+1;

indexY = floor((tmpky-kyMin)/kySteps)+1;

r1 = sqrt(abs(kx(indexX)-tmpkx)^2+abs(ky(indexY)-tmpky)^2);

r2 =

sqrt(abs(kx(indexX+1)-tmpkx)^2+abs(ky(indexY)-tmpky)^2);

r3 = sqrt(abs(kx(indexX)-tmpkx)^2+abs(ky(ind

exY+1)-tmpky)^2);

r4 = sqrt(abs(kx(indexX+1)-tmpkx)^2+abs(ky(ind

exY+1)-tmpky)^2);

R = 1/r1+1/r2+1/r3+1/r4;

A1 = Es(t,v)/(r1\*R);

A2 = Es(t,v)/(r2\*R);

A3 = Es(t,v)/(r3\*R);

A4 = Es(t,v)/(r4\*R);

newEs(indexY,indexX) = newEs(indexY,indexX)+A1;

newEs(indexY,indexX+1) = newEs(indexY,indexX+1)+A2;

newEs(indexY+1,indexX) = newEs(indexY+1,indexX)+A3;

newEs(indexY+1,indexX+1) = newEs(indexY+1,indexX+1)+A4;

end

end

% down sample newEs by MM times

newEs=newEs(1:MM: size(newEs),1:MM: size(newEs));

%---Figure 4.25---------------------------------------------------–

% reformatted data

h = figure;

Kx = kx(1:Nx-1);

Ky = ky(1:Ny-1);

matplot2(Kx,Ky,newEs,40);

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’kx [rad/m]’); ylabel(’ky [rad/m]’);

% Find Corresponding ISAR window in Range and X-Range

kxMax = max(max(kx));

kxMin = min(min(kx));

kyMax = max(max(ky));

kyMin = min(min(ky));

BWKx = kxMax-kxMin;

BWKy = kyMax-kyMin;

dx = pi/BWKx;

dy = pi/BWKy;

X = dx\*(-nSampling/2:nSampling/2);

Y = dy\*(-nSampling/2:nSampling/2);

%---Figure 4.26---------------------------------------------------–

% Plot the resultant ISAR image

h = figure;

tt = nSampling/4:3\*nSampling/4;

ISAR3 = fftshift(ifft2(newEs));

matplot2(X,Y,ISAR3(:,tt),25);

axis([-8 8 -6 6])

colormap(1-gray);

colorbar

set(gca,’FontName’, ’Arial’, ’FontSize’,12,’FontWeight’,

’Bold’);

xlabel(’Range [m]’);

ylabel(’Cross - range [m]’)