LAMPIRAN

List Program

1. “maketraj.m”

```
function [times , trajectory] = maketraj(start_loc, xvelocity, yamp,
yperiod, zamp, zperiod, samplingtime, deltat)
% CARA MENGGUNAKAN: [times , trajectory] = maketraj(start_loc,
xvelocity, yamp, yperiod, zamp, zperiod, samplingtime, deltat)
%
% CATATAN: keterangan dari semua koordinat pada radar.
% INPUT
%
% nama dimensi keterangan satuan
%------ ------ --------------- -------
% start_loc 3 X 1 lokasi awal target m
% xvelocity 1 kecepatan target m/s
% yamp 1 amplituda dari osilasi arah y m
% yperiod 1 perioda dari osilasi arah y m
% zamp 1 amplituda dari osilasi arah z m
% zperiod 1 peroida dari osilasi arah z m
% samplingtime 1 waktu jeda lintasan sec
% deltat 1 waktu antara sample sec
%
% OUTPUTS
%
% nama dimensi keterangan satuan
%------ ---------- --------------- ------
% times 1 X samplingtime/deltat vektor dari kores-
% ponding ke sample sec
% trajectory 3 X samplingtime/deltat lintasan x,y,z m
%
times = 0: deltat: samplingtime ;
x = start_loc(1)+xvelocity.*times ;
if yperiod~=0
y = start_loc(2)+yamp*cos(2*pi*(1/yperiod).*times) ;
else
y = ones(1, length(times))*start_loc(2) ;
end
if zperiod~=0
z = start_loc(3)+zamp*cos(2*pi*(1/zperiod).*times) ;
else
z = ones(1, length(times))*start_loc(3) ;
end
trajectory = [x ; y ; z] ;
```

2.“addnoise.m”

```
function [noisytraj ] = addnoise(trajectory, sigmaaz, sigmael, sigmarange )
%
% CARA MENGGUNAKAN: [noisytraj ] = addnoise(trajectory, sigmaaz, sigmael,
sigmarange )
% INPUT
%
% namA dimensi keterangan satuan
%------ ------ --------------- -------
% trajectory 3 X POINTS lintasan pada koordinat radar [m;m;m]
```

```
% sigmaaz 1 standar deviasi dari eror azimuthr radians
% sigmael 1 standar deviasi dari eror elevasi radians
% sigmarange 1 standar deviatsi dari eror jarak m
%
% OUTPUTS
%
% nama dimensi keterangan satuan
%------ ------ --------------- -------
% noisytraj 3 X POINTS lintasan noise [m;m;m]
noisytraj = zeros(3, size(trajectory,2)) ;
for loop = 1 : size(trajectory,2)
x = trajectory(1,loop);
y = trajectory(2,loop);
z = trajectory(3,loop);
azimuth_corrupted = atan2(y,x) + sigmaaz*randn(1) ;
elevation_corrupted = atan2(z, sqrt(x^2+y^2)) + sigmael*randn(1) ;
range_corrupted = sqrt(x^2+y^2+z^2) + sigmarange*randn(1) ;
x_corrupted = range_corrupted*cos(elevation_corrupted)*cos(azimuth_corrupted) ;
y_corrupted = range_corrupted*cos(elevation_corrupted)*sin(azimuth_corrupted) ;
z_corrupted = range_corrupted*sin(elevation_corrupted) ;
noisytraj(:,loop) = [x_corrupted ; y_corrupted; z_corrupted ] ;
        end % loop berikutnya
```

3. “Rayleigh dan Gaussian”

```
clear all
close all
xg = linspace(-6,6,1500); % randowm variable between -4 and 4
xr = linspace(0,6,1500); % randowm variable between 0 and 8
mu = 0; % zero mean Gaussain pdf mean
sigma = 1.5; % standard deviation (sqrt(variance)
ynorm = normpdf(xg,mu,sigma); % use MATLAB funtion normpdf
yray = raylpdf(xr,sigma); % use MATLAB function raylpdf
plot(xg,ynorm,'k',xr,yray,'k-.');
grid
legend('Gaussian pdf','Rayleigh pdf')
xlabel('x')
ylabel('Probability density')
gtext('\mu = 0; \sigma = 1.5')
gtext('\sigma =1.5')
```

4 “ Peringatan palsu”

```
close all
clear all
logpfa = linspace(.01,250,1000);
var = 10.^(logpfa ./ 10.0);
vtnorm =  sqrt( log (var));
semilogx(logpfa, vtnorm,'k')
grid
```

5 “Deteksi”

```
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
   index = index +1;
   prob1(index) =  pd_swerling3 (nfa, 1, snr);
   prob10(index) =  pd_swerling3 (nfa, 10, snr);
   prob50(index) =  pd_swerling3(nfa, 50, snr);
   prob100(index) =  pd_swerling3 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', ...
   x, prob100,'k-.');
axis([-10 30 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

6 “Peningkatan Pulse Integration Non Koheren”

```
clear all
close all
pfa1 = 1.0e-2;
pfa2 = 1.0e-6;
pfa3 = 1.0e-10;
pfa4 = 1.0e-13;
pd1 = .5;
pd2 = .8;
pd3 = .95;
pd4 = .999;
index = 0;
for np = 1:1:1000
  index = index + 1;
  I1(index) = improv_fac (np, pfa1, pd1);
  I2(index) = improv_fac (np, pfa2, pd2);
  I3(index) = improv_fac (np, pfa3, pd3);
  I4(index) = improv_fac (np, pfa4, pd4);
end
np = 1:1:1000;
semilogx (np, I1, 'k', np, I2, 'k--', np, I3, 'k-.', np, I4, 'k:')
%set (gca,'xtick',[1 2 3 4 5 6 7 8  10 20 30 100]);
xlabel ('Number of pulses');
ylabel ('Improvement factor in dB')
legend ('pd=.5, nfa=e+2','pd=.8, nfa=e+6','pd=.95, nfa=e+10','pd=.999, nfa=e+13');
grid
```

7 “Kerugian Pulse Integration Non Koheren”

```
clear all
close all
pfa1 = 1.0e-12;
pfa2 = 1.0e-12;
pfa3 = 1.0e-12;
pfa4 = 1.0e-12;
pd1 = .5;
pd2 = .8;
pd3 = .95;
pd4 = .99;
index = 0;
for np = 1:1:1000
   index = index+1;
   I1 = improv_fac (np, pfa1, pd1);
```

```
    i1 = 10.^(0.1*I1);
    L1(index) = -1*10*log10(i1 ./ np);
    I2 = improv_fac (np, pfa2, pd2);
    i2 = 10.^(0.1*I2);
    L2(index) = -1*10*log10(i2 ./ np);
    I3 = improv_fac (np, pfa3, pd3);
    i3 = 10.^(0.1*I3);
    L3(index) = -1*10*log10(i3 ./ np);
    I4 = improv_fac (np, pfa4, pd4);
    i4 = 10.^(0.1*I4);
    L4 (index) = -1*10*log10(i4 ./ np);
end
np = 1:1:1000;
semilogx (np, L1, 'k', np, L2, 'k--', np, L3, 'k-.', np, L4, 'k:')
axis tight
xlabel ('Number of pulses');
ylabel ('Integration loss - dB')
legend ('pd=.5, nfa=e+12','pd=.8, nfa=e+12','pd=.95, nfa=e+12','pd=.99, nfa=e+12');
grid
```

8 “Gamma”

```
close all
clear all
format long
ii = 0;
for x = 0:.1:20
  ii = ii+1;
  val1(ii) = incomplete_gamma(x , 1);
  val2(ii) = incomplete_gamma(x , 3);
  val = incomplete_gamma(x , 6);
  val3(ii) = val;
  val = incomplete_gamma(x , 10);
  val4(ii) = val;
end
xx = 0:.1:20;
plot(xx,val1,'k',xx,val2,'k:',xx,val3,'k--',xx,val4,'k-.')
legend('N = 1','N = 3','N = 6','N = 10')
xlabel('x')
ylabel('Incomplete Gamma function (x,N)')
grid
```

9 “Batas Ambang Versus np”

```
clear all
for n= 1: 1:150
   [pfa1 y1(n)] = threshold(1000,n);
   [pfa2 y3(n)] = threshold(10000,n);
   [pfa3 y4(n)] = threshold(500000,n);
end
n =1:1:150;
loglog(n,y1,'k',n,y3,'k--',n,y4,'k-.');
axis([0 200 1 300])
xlabel ('Number of pulses');
ylabel('Threshold')
legend('nfa=1000','nfa=10000','nfa=500000')
grid
```

10 “Deteksi Versus SNR”

```
close all
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
b = sqrt(-2.0 * log(pfa));
index = 0;
for snr = 0:.1:20
   index = index +1;
   a = sqrt(2.0 * 10^(.1*snr));
   pro(index) = marcumsq(a,b);
   prob205(index) =  pd_swerling5 (pfa, 1, 10, snr);
end
x = 0:.1:20;
plot(x, pro,'k',x,prob205,'k:');
axis([0 20 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10')
grid
```

11 “Deteksi Versus SNR”

```
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
b = sqrt(-2.0 * log(pfa));
```

```
index = 0;
for snr = 0:.01:22
   index = index +1;
   a = sqrt(2.0 * 10^(.1*snr));
   pro(index) = marcumsq(a,b);
   prob(index) =  pd_swerling2 (nfa, 1, snr);
end
x = 0:.01:22;
plot(x, pro,'k',x,prob,'k:');
axis([2 22 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('Swerling V','Swerling I')
grid
```

12 "Sweling I"

```
clear all
pfa = 1e-11;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
   index = index +1;
   prob1(index) =  pd_swerling1 (nfa, 1, snr);
   prob10(index) =  pd_swerling1 (nfa, 10, snr);
   prob50(index) =  pd_swerling1 (nfa, 50, snr);
   prob100(index) =  pd_swerling1 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', ...
   x, prob100,'k-.');
axis([-10 30 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

13 "Sweling II"

```
clear all
pfa = 1e-10;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
   index = index +1;
```

```
  prob1(index) =  pd_swerling2 (nfa, 1, snr);
  prob10(index) =  pd_swerling2 (nfa, 10, snr);
  prob50(index) =  pd_swerling2 (nfa, 50, snr);
  prob100(index) =  pd_swerling2 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', ...
  x, prob100,'k-.');
axis([-10 30 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

14 "Sweling III"

```
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
  prob1(index) =  pd_swerling3 (nfa, 1, snr);
  prob10(index) =  pd_swerling3 (nfa, 10, snr);
  prob50(index) =  pd_swerling3(nfa, 50, snr);
  prob100(index) =  pd_swerling3 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', ...
  x, prob100,'k-.');
axis([-10 30 0 1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
```

15 "Sweling IV"

```
clear all
pfa = 1e-9;
nfa = log(2) / pfa;
index = 0;
for snr = -10:.5:30
  index = index +1;
```

```
   prob1(index) =  pd_swerling4 (nfa, 1, snr);
   prob10(index) =  pd_swerling4 (nfa, 10, snr);
   prob50(index) =  pd_swerling4(nfa, 50, snr);
   prob100(index) =  pd_swerling4 (nfa, 100, snr);
end
x = -10:.5:30;
plot(x, prob1,'k',x,prob10,'k:',x,prob50,'k--', ...
   x, prob100,'k-.');
axis([-10 30 0 1.1])
xlabel ('SNR - dB')
ylabel ('Probability of detection')
legend('np = 1','np = 10','np = 50','np = 100')
grid
axis tight
```