LAMPIRAN A
LISTING PROGRAM

```
clc;
clear;
close all;
clc;

N_1=;
N_3=;
sigma_0_2=0.7697; % light = 0.7697
kappa_0=0.4045;  % light = 0.4045;
theta_0=164*pi/180;  % light = 164*pi/180; 164*pi/180)
f_max=91;
sigma_3=0.0062; % light =0.0062;
m_3=0.3861; % light =-0.3861;
rho=1.567;  % light =1.567;
theta_rho=127*pi/180;    % light =127*pi/180; 127*pi/180)
K_c=1.735;
f_c=f_max./K_c;
T_s=1.8e-4;
T_sim=3;
PLOT=1;


N_1_s=ceil(N_1/(2/pi*asin(kappa_0)));


[f1,c1,th1]=parameter_Jakes('es_j',N_1_s,sigma_0_2,f_max,'rand',
0);
f1 =f1(1:N_1);
c1 =c1(1:N_1);
th1=th1(1:N_1);

f_i_n=f1;
c_i_n=c1;
theta_i_n=th1;
N_i=N_1;
K=K_c;

figure(1);
subplot(1,2,1);
stem([-f_i_n(N_i:-1:1);f_i_n],1/4*[c_i_n(N_i:-1:1);c_i_n].^2);
title('Estimasi untuk rapat spektral Jakes');
grid;
xlabel('f(Hz)');
ylabel('Rapat spektral daya');
tau_max=N_i/(K*f_max);
tau=linspace(0,tau_max,500);
% r_mm=sigma_0^2*besselj(0,2*pi*f_max*tau);
```

```
r_mm=sigma_0_2^2*besselj(0,2*pi*f_max*tau);
r_mm_tilde=acf_mue(f_i_n,c_i_n,tau);
subplot(1,2,2);
plot(tau,r_mm,'r-',tau,r_mm_tilde,'b--');
title('Estimasi untuk autokorelasi dengan Jakes');
legend('asli','estimasi');
grid;
xlabel('tau (s)');
ylabel('Fungsi autokorelasi');
 [f3,c3,th3]=parameter_Gauss('es_g',N_3,1,f_max,f_c,'rand',0);

gaMma=(2*pi*f_c/sqrt(2*log(2)))^2;
f3(N_3)=sqrt(gaMma*N_3/(2*pi)^2-sum(f3(1:N_3-1).^2));

f_i_n=f3;
c_i_n=c3;
theta_i_n=th3;
N_i=N_3;
K=K_c;

figure(2);
subplot(1,2,1);
stem([-f_i_n(N_i:-1:1);f_i_n],1/4*[c_i_n(N_i:-1:1);c_i_n].^2);
title('Estimasi untuk rapat spektral Gaussian');
grid;
xlabel('f (Hz)');
ylabel('Rapat spektral daya');
% tau_max=N_i/(K*kappa_c*f_c);
tau_max=N_i/(K*K_c*f_c);
tau=linspace(0,tau_max,500);
r_mm=sigma_0_2*exp(-(pi*f_c/sqrt(log(2))*tau).^2);
r_mm_tilde=acf_mue(f_i_n,c_i_n,tau);
subplot(1,2,2);
plot(tau,r_mm,'r-',tau,r_mm_tilde,'b--');
title('Estimasi untuk autokorelasi dengan Gaussian');
legend('asli','estimasi');
grid;
xlabel('tau (s)');
ylabel('Fungsi autokorelasi');


N=ceil(T_sim/T_s);
t=(0:N-1)*T_s;

xi_t=abs(Mu_i_t(c1,f1,th1,T_s,T_sim)+rho*cos(theta_rho)+...
         j*(Mu_i_t(c1,f1,th1-theta_0,T_s,T_sim)+...
          rho*sin(theta_rho) ) );
```

```
lambda_t=exp(Mu_i_t(c3,f3,th3,T_s,T_sim)*sigma_3+m_3);

eta_t=xi_t.*lambda_t;

figure(3);
if PLOT==1,
    plot(t,20*log10(eta_t),'b-');
    xlabel('t (s)');
    ylabel('20 log eta(t)');
    title('Estimasi Proses Suzuki tipe II');
    grid;
    legend('heavy shadowing');
    %    legend('light shadowing');
end


%-----------------------------------------------------------------
-----
% parameter_Jakes.m --------------------------------------------
-----
%
% Program untuk komputasi frekuensi Doppler diskrit dan phasa
Doppler dengan % menggunakan jakes rapat spectral
% Used m-files: LPNM_opt_Jakes.m, fun_Jakes.m,
%                           grad_Jakes.m, acf_mue.m
%-----------------------------------------------------------------
-----
%
[f_i_n,c_i_n,theta_i_n]=parameter_Jakes(METHOD,N_i,sigma_0_2,...
%                                       f_max,PHASE,PLOT)
%---------------------------------------------------------------
% beberapa parameter input:
%
% METHOD:
% |----------------------------------------------|-------------------|
% | Methods for the computation of the discrete  |      Input        |
% | Doppler frequencies and Doppler coefficients |                   |
% |----------------------------------------------|-------------------|
% |----------------------------------------------|-------------------|
% | Method of equal distances (MED)              |     'ed_j'        |
% |----------------------------------------------|-------------------|
% | Mean square error method  (MSEM)             |     'ms_j'        |
% |----------------------------------------------|-------------------|
% | Method of equal areas (MEA)                  |     'ea_j'        |
% |----------------------------------------------|-------------------|
% | Monte Carlo method (MCM)                     |     'mc_j'        |
% |----------------------------------------------|-------------------|
% | Lp-norm method (LPNM)                        |     'lp_j'        |
% |----------------------------------------------|-------------------|
% | Method of exact Doppler spread (MEDS)        |     'es_j'        |
```

```
% |-----------------------------------------------|------------------|
% | Jakes method (JM)                             |      'jm_j'      |
% |-----------------------------------------------|------------------|
%
% N_i: jumlah fungsi harmonik
% sigma_0_2: daya rata-rata proses real Gaussian mu_i(t)
% f_max: frekuensi maximum doppler
%
% PHASE:
% |-----------------------------------------------|------------------|
% | Methods for the computation of the Doppler    |      Input       |
% | phases                                        |                  |
% |-----------------------------------------------|------------------|
% |-----------------------------------------------|------------------|
% | Random Doppler phases                         |      'rand'      |
% |-----------------------------------------------|------------------|
% | Permuted Doppler phases                       |      'perm'      |
% |-----------------------------------------------|------------------|
%
% PLOT: plot of the ACF and the PSD of mu_i(t), if PLOT==1

function [f_i_n,c_i_n,theta_i_n]=parameter_Jakes(METHOD,N_i,...
                                  sigma_0_2,f_max,PHASE,PLOT)

if nargin<6,
   error('Not enough input parameters')
end

%-------------------------------------------------------------------
% parameter_Gauss.m ------------------------------------------------
%
% Program untuk komputasi frekuensi Doppler diskrit, koefisien Doppler phasa
% Doppler menggunakan rapat spectral daya
%
% Gunakan m-files: LPNM_opt_Gauss.m, fun_Gauss.m,
%                         grad_Gauss.m, acf_mue.m
%-------------------------------------------------------------------
% [f_i_n,c_i_n,theta_i_n]=parameter_Gauss(METHOD,N_i,sigma_0_2,...
%                                      f_max,f_c,PHASE,PLOT)
%-------------------------------------------------------------------
% Explanation of the input parameters:
%
% METHOD:
% |-----------------------------------------------|------------------|
% | Methods for the computation of the discrete   |      Input       |
% | Doppler frequencies and Doppler coefficients  |                  |
% |-----------------------------------------------|------------------|
% |-----------------------------------------------|------------------|
% | Method of equal distances (MED)               |      'ed_g'      |
% |-----------------------------------------------|------------------|
% | Mean square error method  (MSEM)              |      'ms_g'      |
% |-----------------------------------------------|------------------|
% | Method of equal areas (MEA)                   |      'ea_g'      |
% |-----------------------------------------------|------------------|
% | Monte Carlo method (MCM)                      |      'mc_g'      |
```

```
% |-----------------------------------------------|-------------------|
% | Lp-norm method (LPNM)                         |      'lp_g'       |
% |-----------------------------------------------|-------------------|
% | Method of exact Doppler spread (MEDS)         |      'es_g'       |
% |-----------------------------------------------|-------------------|
%
% N_i: jumlah fungsi harmonik
% sigma_0_2: daya rata-rata of the real deterministic Gaussian
%            process mu_i(t)
% f_max: maximum Doppler frequency
% f_c: 3-dB-cutoff frequency
%
% PHASE:
% |-----------------------------------------------|-------------------|
% | Methods for the computation of the Doppler    |       Input       |
% | phases                                        |                   |
% |-----------------------------------------------|-------------------|
% |-----------------------------------------------|-------------------|
% | Random Doppler phases                         |      'rand'       |
% |-----------------------------------------------|-------------------|
% | Permuted Doppler phases                       |      'perm'       |
% |-----------------------------------------------|-------------------|
%
% PLOT: plot of the ACF and the PSD of mu_i(t), if PLOT==1

function [f_i_n,c_i_n,theta_i_n]=parameter_Gauss(METHOD,N_i,...
                                   sigma_0_2,f_max,f_c,PHASE,PLOT)

if nargin<7,
   error('Not enough input parameters')
end
```

```
%--------------------------------------------------------------------
% Suzuki_Type_II.m ------------------------------------------------------
%
% program untuk simulasi deterministic extended Suzuki proses tipe II
%
% Gunakan m-files: parameter_Jakes.m, parameter_Gauss.m, Mu_i_t.m
%--------------------------------------------------------------------
% eta_t=Suzuki_Type_II(N_1,N_3,sigma_0_2,kappa_0,theta_0,f_max,...
%                   sigma_3,m_3,rho,theta_rho,f_c,T_s,T_sim,PLOT)
%--------------------------------------------------------------------
% Beberapa parameter input:
%
% N_1, N_3: jumlah fungsi harmonic dari proses real Gaussian nu_0(t) and
% nu_3(t)
% sigma_0_2: daya rata-rata  proses real gaussian mu_0(t)  (untuk kappa_0=1)
% kappa_0: perbandingan frekuensi f_min/f_max (0<=kappa_0<=1)
% theta_0: pergeseran phasa antara mu_1_n(t)dan mu_2_n(t)
% f_max: prekuensi maximum doppler
% sigma_3: kuadrat daya rata-rata proses real Gaussian nu_3(t)
% m_3: nilai rata-rata proses real Gaussian mu_3(t)
% rho: amplituda of the komponen LOS m(t)
```

```
% theta_rho: phasa komponen LOS m(t)
% f_c: 3-dB-frekuensi cut off
% T_s: sampling interval
% T_sim: durasi dari simulasi
% PLOT: plot of the deterministic extended Suzuki process eta(t) of
%       Type II, if PLOT==1

function
eta_t=Suzuki_Type_II(N_1,N_3,sigma_0_2,kappa_0,theta_0,...

f_max,sigma_3,m_3,rho,theta_rho,f_c,...
                            T_s,T_sim,PLOT)
if nargin==13,
   PLOT=0;
end

N_1_s=ceil(N_1/(2/pi*asin(kappa_0)));
[f1,c1,th1]=parameter_Jakes('es_j',N_1_s,sigma_0_2,f_max,'rand',
0);
f1 =f1(1:N_1);
c1 =c1(1:N_1);
th1=th1(1:N_1);

[f3,c3,th3]=parameter_Gauss('es_g',N_3,1,f_max,f_c,'rand',0);
gaMma=(2*pi*f_c/sqrt(2*log(2)))^2;
f3(N_3)=sqrt(gaMma*N_3/(2*pi)^2-sum(f3(1:N_3-1).^2));

N=ceil(T_sim/T_s);
t=(0:N-1)*T_s;

xi_t=abs(Mu_i_t(c1,f1,th1,T_s,T_sim)+rho*cos(theta_rho)+...
         j*(Mu_i_t(c1,f1,th1-theta_0,T_s,T_sim)+...
            rho*sin(theta_rho) ) );

lambda_t=exp(Mu_i_t(c3,f3,th3,T_s,T_sim)*sigma_3+m_3);

eta_t=xi_t.*lambda_t;


if PLOT==1,
   plot(t,20*log10(eta_t),'b-')
   xlabel('t (s)')
   ylabel('20 log eta(t)')
   grid
   legend('heavy shadowing')
   % legend('light shadowing')
end
```