LAMPIRAN A

LISTING PROGRAM

```
clear;
close all;
clc;

co=3*1e8;
fo=9*1e8;

% sprintf('%s','0.1')
% sigma_0_2=input('Masukkan daya rata-rata dari proses Gaussian real
% deterministik mu_i(t) = ');
sigma_0_2=1;
disp('              ');

sprintf('%s','ja ','untuk Jakes atau ', 'ri ' ,'untuk Rice atau' ,' G1 ', 'untuk Gaussian 1 atau '
,'G2 ', 'untuk Gaussian 2')
D_S_T=input('Masukkan jenis Doppler PSD yang dipilih = ');
disp('              ');
sigma_0=sqrt(sigma_0_2);
% sprintf('%s','25')
% N_i=input('Masukkan jumlah fungsi harmonik = ');
N_i=15;
disp('              ');
v=input('Masukkan kecepatan unit mobile = ');
disp('              ');
% fo ditentukan untuk 900 Mhz
% fo=input('Masukkan frekuensi pemancar = ');
% disp('        ');
f_max=(v./co).*fo;
PHASE='rand';
PLOT=1;
f_c=sqrt(log(2)).*f_max;

[f1,f2,c1,c2,th1,th2,rho,f_rho,f01,f02]=pCOST207(D_S_T,N_i);
f_i_n=f1;
c_i_n=c1;
K=1.675;
    figure;
    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);
    stem([-f1(N_i:-1:1);f1],1/4*[c1(N_i:-1:1);c1].^2);
     grid on;
     xlabel('f/Hz');
     ylabel('LDS');
     subplot(1,2,2)
     stem([-f_i_n(N_i:-1:1);f_i_n],1/4*[c_i_n(N_i:-1:1);c_i_n].^2);
     stem([-f2(N_i+1:-1:1);f2],1/4*[c2(N_i+1:-1:1);c2].^2);
     grid on;
     xlabel('f/Hz');
     ylabel('LDS');
     legend('Estimasi 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_tilde=acf_mue(f_i_n,c_i_n,tau);
    subplot(1,2,2)
    plot(tau,r_mm,'r-',tau,r_mm_tilde,'b--')
    grid on;
    xlabel('tau (s)');
    ylabel('ACF');
    legend('Nilai autokorelasi sebenarnya (teoritis)','Nilai estimasi autokorelasi (acf)');
```

```
%-----------------------------------------------------------------
% pCOST207.m ------------------------------------------------------
%
% Program for the derivation of the channel parameters of the
% Doppler PSDs defined by COST 207.
%
%-----------------------------------------------------------------
%[f1,f2,c1,c2,th1,th2,rho,f_rho,f01,f02]=pCOST207(D_S_T,N_i)
%-----------------------------------------------------------------
% Explanation of the input parameters:
%
% D_S_T: type of the Doppler PSD:
%           Jakes:    D_S_T='JA'
%           Rice:     D_S_T='RI'
%           Gauss  I: D_S_T='G1'
%           Gauss II: D_S_T='G2'
% N_i: number of harmonic functions

function [f1,f2,c1,c2,th1,th2,rho,f_rho,f01,f02]=pCOST207(D_S_T,N_i)

if    all(lower(D_S_T)=='ri'), % RICE
      n=(1:N_i);
      f1=sin(pi/(2*N_i)*(n-1/2));
      c1=0.41*sqrt(1/N_i)*ones(1,N_i);
      th1=rand(1,N_i)*2*pi;
      n=(1:N_i+1);
      f2=sin(pi/(2*(N_i+1))*(n-1/2));
      c2=0.41*sqrt(1/(N_i+1))*ones(1,N_i+1);
      th2=rand(1,N_i+1)*2*pi;
      f01=0;f02=0;
      rho=0.91;f_rho=0.7;
elseif all(lower(D_S_T)=='ja'), % JAKES
      n=(1:N_i);
      f1=sin(pi/(2*N_i)*(n-1/2));
      c1=sqrt(1/N_i)*ones(1,N_i);
      th1=rand(1,N_i)*2*pi;
      n=(1:N_i+1);
      f2=sin(pi/(2*(N_i+1))*(n-1/2));
```

```
        c2=sqrt(1/(N_i+1))*ones(1,N_i+1);
        th2=rand(1,N_i+1)*2*pi;
        f01=0;f02=0;
        rho=0;f_rho=0;
elseif all(lower(D_S_T)=='g1'), % GAUSS I
        n=(1:N_i);
        sgm_0_2=5/6;
        c1=sqrt(sgm_0_2*2/N_i)*ones(1,N_i);
        f1=sqrt(2)*0.05*erfinv((2*n-1)/(2*N_i));
        th1=rand(1,N_i)*2*pi;
        sgm_0_2=1/6;
        c2=[sqrt(sgm_0_2*2/N_i)*ones(1,N_i),0]/j;
        f2=[sqrt(2)*0.1*erfinv((2*n-1)/(2*N_i)),0];
        th2=[rand(1,N_i)*2*pi,0];
        f01=0.8;f02=-0.4;
        rho=0;f_rho=0;
elseif all(lower(D_S_T)=='g2'), % GAUSS II
        n=(1:N_i);
        sgm_0_2=10^0.5/(sqrt(10)+0.15);
        c1=sqrt(sgm_0_2*2/N_i)*ones(1,N_i);
        f1=sqrt(2)*0.1*erfinv((2*n-1)/(2*N_i));
        th1=rand(1,N_i)*2*pi;
        sgm_0_2=0.15/(sqrt(10)+0.15);
        c2=[sqrt(sgm_0_2*2/N_i)*ones(1,N_i),0]/j;
        f2=[sqrt(2)*0.15*erfinv((2*n-1)/(2*N_i)),0];
        th2=[rand(1,N_i)*2*pi,0];
        f01=-0.7;f02=0.4;
        rho=0;f_rho=0;

end



%----------------------------------------------------------------------
% F_S_K_p.m ------------------------------------------------------------
%
% Program for the generation of the matrices used in F_S_K.m.
%
% Used m-file: pCOST207.m
%----------------------------------------------------------------------
% [C1,F1,TH1,C2,F2,TH2,F01,F02,RHO,F_RHO,q_l,T]=
%                       F_S_K_p(N_1,AREA,f_max)
%----------------------------------------------------------------------
% Explanation of the input parameters:
%
% N_1: minimum number of discrete Doppler frequencies
% AREA: according to COST 207, 4 types of channels are specified:
%            1) Rural Area:    'ra'
%            2) Typical Urban: 'tu'
%            3) Bad Urban:     'bu'
%            4) Hilly Terrain: 'ht'
% f_max: maximum Doppler frequency
```

```
function [C1,F1,TH1,C2,F2,TH2,F01,F02,RHO,F_RHO,q_l,T]=...
         F_S_K_p(N_1,AREA,f_max)

% The greatest common divisor of the discrete propagation delays
% defines the sampling interval T_s:
T_s=0.2E-6;

if     all(lower(AREA)=='ra'),
       a_l=[1,0.63,0.1,0.01];
       tau_l=[0,0.2,0.4,0.6]*1E-6;
       DOPP_KAT=['RI';'JA';'JA';'JA'];
elseif all(lower(AREA)=='tu'),
       a_l=[0.5,1,0.63,0.25,0.16,0.1];
       tau_l=[0,0.2,0.6,1.6,2.4,5]*1E-6;
       DOPP_KAT=['JA';'JA';'G1';'G1';'G2';'G2'];
elseif all(lower(AREA)=='bu'),
       a_l=[0.5,1,0.5,0.32,0.63,0.4];
       tau_l=[0,0.4,1.0,1.6,5.0,6.6]*1E-6;
       DOPP_KAT=['JA';'JA';'G1';'G1';'G2';'G2'];
elseif all(lower(AREA)=='ht'),
       a_l=[1,0.63,0.4,0.2,0.25,0.06];
       tau_l=[0,0.2,0.4,0.6,15,17.2]*1E-6;
       DOPP_KAT=['JA';'JA';'JA';'JA';'G2';'G2'];
end

% Generate the parameters and assign them to the matrices:
num_of_taps=length(DOPP_KAT);
F1=zeros(num_of_taps,N_1+2*num_of_taps-1);
F2=F1;C1=F1;C2=F1;TH1=F1;TH2=F1;
F01=zeros(1,num_of_taps);F02=F01;
RHO=zeros(1,num_of_taps);F_RHO=RHO;
NN1=N_1+2*(num_of_taps-1):-2:N_1;
for k=1:num_of_taps,
   [f1,f2,c1,c2,th1,th2,rho,f_rho,f01,f02]=...
   pCOST207(DOPP_KAT(k,:),NN1(k));
   F1(k,1:NN1(k))=f1;
   C1(k,1:NN1(k))=c1*sqrt(a_l(k));
   TH1(k,1:NN1(k))=th1;
   F2(k,1:NN1(k)+1)=f2;
   C2(k,1:NN1(k)+1)=c2*sqrt(a_l(k));
   TH2(k,1:NN1(k)+1)=th2;
   F01(k)=f01;F02(k)=f02;
   RHO(k)=rho;F_RHO(k)=f_rho;
end

% Determine indices of the delay elements of the FIR filter:
q_l=tau_l/T_s+1;

% Initialization of the delay elements of the FIR filter:
T=zeros(1,max(q_l));
```