

## LAMPIRAN A

Berikut ini adalah dokumentasi kode program yang ditulis pada MATLAB<sup>®</sup> versi 7.0.1 yang terdiri dari beberapa *file* atau fungsi untuk menjalankan simulasi ini. Pada umumnya setiap studi kasus terdiri dari 2 fungsi utama dan beberapa fungsi tingkat rendah serta fungsi bawaan dari MATLAB<sup>®</sup>.

### **Forward\_I.m, Forward\_II.m, Forward\_III.m, Forward\_IV.m**

*File* ini merupakan fungsi utama untuk menjalankan simulasi standar IS-95 untuk *forward link*.

### **Reverse\_I.m, Reverse\_II.m, Reverse\_III.m, Reverse\_IV.m**

*File* ini merupakan fungsi utama untuk menjalankan simulasi standar IS-95 untuk *reverse link*.

### **Simulasi\_CDMA.m**

*File* ini merupakan fungsi utama untuk menjalankan simulasi pengiriman data gambar pada sistem CDMA standar IS-95.

### **uplink\_IS\_95.m**

Fungsi ini merupakan pemodelan pemancar dan penerima untuk *uplink* standar IS-95.

### **downlink\_IS\_95.m**

Fungsi ini merupakan pemodelan pemancar dan penerima untuk *downlink* standar IS-95.

### **Mobile\_Transmitter.m**

Fungsi ini adalah pemodelan struktur pemancar pada *mobile unit* standar IS-95 yang digunakan dalam simulasi ini.

### **Base\_Receiver.m**

Fungsi ini adalah pemodelan struktur penerima pada *base station* standar IS-95 yang digunakan dalam simulasi ini.

### **Base\_Transmitter.m**

Fungsi ini adalah pemodelan struktur pemancar pada *base station* standar IS-95 yang digunakan dalam simulasi ini.

**Mobile\_Receiver.m**

Fungsi ini merupakan pemodelan struktur penerima pada *mobile unit* standar IS-95 yang digunakan dalam simulasi ini.

**LFSR.m**

Fungsi ini merupakan fungsi untuk membangkitkan *short PN codes* atau *long PN codes* yang digunakan dalam simulasi ini.

**Jammer.m**

Fungsi ini merupakan fungsi untuk membangkitkan sinyal penginterferensi yang akan ditambahkan pada sinyal yang dipancarkan.

**Data\_Source.m**

Fungsi ini merupakan fungsi untuk membangkitkan data acak atau membaca *file* gambar kedalam data biner untuk digunakan dalam simulasi ini.

**IS95\_Transmitter.m**

Fungsi ini merupakan pemodelan struktur pemancar dan modulator standar IS-95 baik *uplink* maupun *downlink* yang digunakan dalam simulasi ini.

**Conv29.m, Conv39.m**

Fungsi ini merupakan pemodelan untuk proses pengkodean konvolusi yang digunakan dalam simulasi ini.

**AWGNChannel.m**

Fungsi ini merupakan fungsi untuk memodelkan kanal AWGN, dimana terjadi penambahan *white noise* pada bagian ini.

**RayleighFading.m**

Fungsi ini merupakan fungsi untuk memodelkan kanal *rayleigh fading* yang digunakan dalam simulasi ini.

**Interference.m**

Fungsi ini merupakan pemodelan proses interferensi *cochannel* untuk kasus spesial/terburuk.

**IS95\_Receiver.m**

Fungsi ini merupakan pemodelan struktur penerima dan demodulator standar IS-95 baik *uplink* maupun *downlink* yang digunakan dalam simulasi ini.

**Data\_Sink.m**

Fungsi ini merupakan fungsi untuk mengembalikan data kedalam bentuk semula, data biner menjadi *file* gambar.

**deci2bin.m, bin2deci.m**

Fungsi ini merupakan pemodelan untuk proses konversi dari bilangan desimal ke bilangan biner dan sebaliknya.

**imag.m**

Merupakan fungsi bawaan yang berfungsi untuk memisahkan bagian bagian imajiner dari suatu variabel kompleks.

**length.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung panjang atau lebar dari suatu variabel.

**mean.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung nilai rata-rata dari suatu variabel.

**mod.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung nilai sisa dari suatu proses pembagian.

**ones.m**

Merupakan fungsi bawaan yang berfungsi untuk membangkitkan variabel yang semuanya bernilai satu.

**poly2trellis.m**

Merupakan fungsi bawaan yang berfungsi untuk membuat struktur *trellis* untuk keperluan viterbi *decoding* .

**rand.m**

Merupakan fungsi bawaan yang berfungsi untuk membangkitkan data acak antara 0 s.d. 1 berdistribusi seragam.

**randn.m**

Merupakan fungsi bawaan yang berfungsi untuk membangkitkan data acak antara -1 s.d. 1 berdistribusi normal.

**real.m**

Merupakan fungsi bawaan yang berfungsi untuk memisahkan bagian riil dari suatu variabel kompleks.

**reshape.m**

Merupakan fungsi bawaan yang berfungsi untuk mengubah orientasi kolom dan baris dari suatu matriks.

**round.m**

Merupakan fungsi bawaan yang berfungsi untuk membulatkan bilangan menjadi bilangan bulat terdekat.

**semilogy.m**

Merupakan fungsi bawaan yang berfungsi untuk *plotting* dengan skala *axis x* linier dan *axis y* logaritmik.

**sign.m**

Merupakan fungsi bawaan yang berfungsi untuk membulatkan bilangan: bila lebih besar dari 0 maka dibulatkan menjadi 1, bila sama dengan 0 tetap bernilai 0, bila lebih kecil dari 0 maka dibulatkan menjadi -1.

**size.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung ukuran kolom dan baris dari suatu variabel.

**sqrt.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung nilai akar kuadrat dari suatu variabel.

**sum.m**

Merupakan fungsi bawaan yang berfungsi untuk menghitung nilai jumlah dari setiap elemen suatu variabel.

**vitdec.m**

Merupakan fungsi bawaan yang berfungsi untuk pemodelan viterbi *decoding* yang digunakan dalam simulasi ini.

**zeros.m**

Merupakan fungsi bawaan yang berfungsi untuk membangkitkan variabel yang semuanya bernilai nol.

## LAMPIRAN B

### PROGRAM SIMULASI MATLAB

#### 1. Studi Kasus I

```
%===== %
% Simulation of IS-95 Standard for Forward Link %
% Channel model : AWGN %
% User data Rate : 9600 bps %
% Study Case I : Performance IS-95 at AWGN Channel %
% % %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%===== %

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

%--- set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');
longPNcode = longPNcode(1:64:end);

%generate Hadamard matrix & choose the eighth row.
WalshCode = hadamard(64);
WalshCode = WalshCode(8,:);

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
```

```

%--- IS-95 Transmitter ---%
Txsignal = IS95_Transmitter(TxData);
%--- AWGN Channel ---%
Rxsignal = AWGNChannel(Txsignal,EbNoIndex);
%--- IS-95 Receiver ---%
RxData = IS95_Receiver(Rxsignal);
%--- Error Estimation ---%
error = sum(TxData ~= RxData);
BER(iters) = error./length(RxData);
end
%--- Average Simulation BER ---%
AvBER(EbNoIndex) = sum(BER)/nIters;
%--- Theoretical QPSK BER ---%
z = 2*10^(EbNo_dB(EbNoIndex)/10);
TheoBER(EbNoIndex) = 0.5*erfc(sqrt(z)/sqrt(2));
end

%--- Save in file ---%
save AvBER.mat
save TheoBER.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at AWGN Channel')
semilogy(EbNo_dB, AvBER, 'r-');
hold on; grid on;
semilogy(EbNo_dB, TheoBER, 'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('Forward Link IS-95', 'Baseband QPSK');

%=====
% Simulation of IS-95 Standard for Reverse Link %
% Channel model : AWGN %
% User data Rate : 9600 bps %
% Study Case I : Performance IS-95 at AWGN Channel %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%=====

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

```

```

%--- set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code.
longPNcode = LFSR('Long PN code');

%generate 64x64 Hadamard matrix.
WalshCode = hadamard(64);

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        Txsignal = IS95_Transmitter(TxData);
        %--- AWGN Channel ---%
        Rxsignal = AWGNChannel(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        RxData = IS95_Receiver(Rxsignal);
        %--- Error Estimation ---%
        error = sum(TxData ~= RxData);
        BER(iters) = error./length(RxData);
    end
    %--- Average Simulation BER ---%
    AvBER(EbNoIndex) = sum(BER)/nIters;
    %--- Theoretical QPSK BER ---%
    z = 2*10^(EbNo_dB(EbNoIndex)/10);
    TheoBER(EbNoIndex) = 0.5*erfc(sqrt(z)/sqrt(2));
end

%--- Save in file ---%
save AvBER.mat
save TheoBER.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at AWGN Channel');
semilogy(EbNo_dB, AvBER, 'r-');
hold on; grid on;

```

```

semilogy(EbNo_dB, TheoBER, 'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('Reverse Link IS-95', 'Baseband QPSK');

```

## 2. Studi Kasus II

```

%=====
% Simulation of IS-95 Standard for Forward Link %
% Channel model : AWGN %
% Interference : Worst case & omnidirectional antenna %
% User data Rate : 9600 bps %
% Study Case II : Performance IS-95 at AWGN Channel & %
% worst case cochannel interference %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
% %
%=====

```

```

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

```

```

%--- set as global parameter ---%
global EbNo_dB CI_dB shortPNcode longPNcode WalshCode

```

```

%--- Parameter ---%
EbNo_dB = [-2:.5:8]; %Eb/No measuring range [in dB]
CI_dB = 14.47; %C/I in dB for worst case cochannel interference
nIters = 50; %number of iteration

```

```

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

```

```

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');
longPNcode = longPNcode(1:64:end);

```

```

%generate Hadamard matrix & choose the eighth row.
WalshCode = hadamard(64);
WalshCode = WalshCode(2,:);

```



```

%generate Interferers data.
Interferer = Jammer();

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        Txsignal = IS95_Transmitter(TxData);
        %--- Cochannel Interference & AWGN Channel ---%
        Rxsignal_CI = Interference(Txsignal,Interferer,EbNoIndex);
        Rxsignal_NO = AWGNChannel(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        RxData_CI = IS95_Receiver(Rxsignal_CI); %Cochannel Interference
        RxData_NO = IS95_Receiver(Rxsignal_NO); %Noise Only
        %--- Error Estimation ---%
        %With Interference.
        error_CI = sum(TxData ~= RxData_CI);
        BER_CI(iters) = error_CI./length(RxData);
        %Without Interference.
        error_NO = sum(TxData ~= RxData_NO);
        BER_NO(iters) = error_NO./length(RxData);
    end
    %--- Average Simulation BER ---%
    AvBER_CI(EbNoIndex) = sum(BER_CI)/nIters;
    AvBER_NO(EbNoIndex) = sum(BER_NO)/nIters;
end

%--- Save in file ---%
save AvBER_CI.mat
save AvBER_NO.mat

%--- Plot simulation result ---%
figure('name','Effect of worst case cochannel interference')
semilogy(EbNo_dB, AvBER_CI, 'r-');
hold on; grid on;
semilogy(EbNo_dB, AvBER_NO, 'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('With Interference', 'Without Interference');

%=====
% Simulation of IS-95 Standard for Reverse Link %
% Channel model : AWGN %
% Interference : Worst case & omnidirectional antenna %

```

```

% User data Rate : 9600 bps %
% Study Case II : Performance IS-95 at AWGN Channel & %
% worst case cochannel interference %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%=====
%--- clear current workspaces and close any opened window ---%
clear all
close all
clc

%--- set as global parameter ---%
global EbNo_dB CI_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = [-2:.5:8]; %Eb/No measuring range [in dB]
CI_dB = 14.47; %C/I in dB
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code.
longPNcode = LFSR('Long PN code');

%generate 64x64 Hadamard matrix.
WalshCode = hadamard(64);

%generate Interferer data.
Interferer = Jammer();

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        Txsignal = IS95_Transmitter(TxData);
        %--- Cochannel Interference & AWGN Channel ---%
        Rxsignal_CI = Interference(Txsignal,Interferer,EbNoIndex);
        Rxsignal_NO = AWGNChannel(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        RxData_CI = IS95_Receiver(Rxsignal_CI); %Cochannel Interference
    end
end

```

```

RxData_NO = IS95_Receiver(Rxsignal_NO); %Noise Only
%--- Error Estimation ---%
% With Interference.
error_CI = sum(TxData ~= RxData_CI);
BER_CI(iters) = error_CI./length(RxData);
% Without Interference.
error_NO = sum(TxData ~= RxData_NO);
BER_NO(iters) = error_NO./length(RxData);
end
%--- Average Simulation BER ---%
AvBER_CI(EbNoIndex) = sum(BER_CI)/nIters;
AvBER_NO(EbNoIndex) = sum(BER_NO)/nIters;
end

%--- Save in file ---%
save AvBER_CI.mat
save AvBER_NO.mat

%--- Plot simulation result ---%
figure('name','Effect of worst case cochannel interference')
semilogy(EbNo_dB, AvBER_CI, 'r-');
hold on; grid on;
semilogy(EbNo_dB, AvBER_NO, 'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('With Interference', 'Without Interference');

```

### 3. Studi Kasus III

```

%=====
% Simulation of IS-95 Standard for Forward Link %
% Channel model : AWGN & Fading Channel %
% User data Rate : 9600 bps %
% Study Case III : Performance IS-95 at AWGN Channel & %
% Flat Reyleigh-Fading Channel %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%=====

```

```

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

```

```

%--- Set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 5; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');
longPNcode = longPNcode(1:64:end);

%generate walsh code & choose the eighth row.
WalshCode = hadamard(64);
WalshCode = WalshCode(8,:);

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        Txsignal = IS95_Transmitter(TxData);
        %--- Reyleigh Fading & AWGN Channel ---%
        FadedNoisy = ReyleighFading(Txsignal,EbNoIndex);
        Noisy = AWGNChannel(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        RxData_FN = IS95_Receiver(FadedNoisy);
        RxData_NO = IS95_Receiver(Noisy);
        %--- Error Estimation ---%
        %for Rayleigh Fading, FN = Faded and Noisy.
        errors_FN = sum(TxData ~= RxData_FN);
        BER_FN(iters) = errors_FN./length(RxData_FN);
        %for AWGN, NO = Noise Only.
        errors_NO = sum(TxData ~= RxData_NO);
        BER_NO(iters) = errors_NO./length(RxData_NO);
    end
    %--- Average Simulation BER ---%
    AvBER_FN(EbNoIndex) = sum(BER_FN)/nIters;
    AvBER_NO(EbNoIndex) = sum(BER_NO)/nIters;
end

%--- Save in file ---%
save AvBER_FN.mat

```

```

save AvBER_NO.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at Reyleigh-Fading Channel')
semilogy(EbNo_dB,AvBER_FN,'r-');
hold on; grid on;
semilogy(EbNo_dB,AvBER_NO,'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('Rayleigh Fading', 'AWGN');

%=====
% Simulation of IS-95 Standard for Reverse Link %
% Channel model : AWGN & Fading Channel %
% User data Rate : 9600 bps %
% Study Case III : Performance IS-95 at AWGN Channel & %
% Flat Reyleigh-Fading Channel %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%=====

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

%--- Set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');

%generate 64x64 Hadamard matrix.
WalshCode = hadamard(64);

%--- Simulation loop ---%

```

```

for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        Txsignal = IS95_Transmitter(TxData);
        %--- Reyleigh Fading & AWGN Channel ---%
        FadedNoisy = ReyleighFading(Txsignal,EbNoIndex);
        Noisy = AWGNChannel(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        RxData_FN = IS95_Receiver(FadedNoisy);
        RxData_NO = IS95_Receiver(Noisy);
        %--- Error Estimation ---%
        %for Rayleigh Fading, FN = Faded and Noisy.
        errors_FN = sum(TxData ~= RxData_FN);
        BER_FN(iters) = errors_FN./length(RxData_FN);
        %for AWGN, NO = Noise Only.
        errors_NO = sum(TxData ~= RxData_NO);
        BER_NO(iters) = errors_NO./length(RxData_NO);
    end
    %--- Average Simulation BER ---%
    AvBER_FN(EbNoIndex) = sum(BER_FN)/nIters;
    AvBER_NO(EbNoIndex) = sum(BER_NO)/nIters;
end

%--- Save in file ---%
save AvBER_FN.mat
save AvBER_NO.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at Reyleigh-Fading Channel')
semilogy(EbNo_dB,AvBER_FN,'r-');
hold on; grid on;
semilogy(EbNo_dB,AvBER_NO,'b-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('Rayleigh Fading', 'AWGN');

```

#### 4. Studi Kasus IV

```

%=====
% Simulation of IS-95 Standard for Forward Link %
% Channel model : AWGN & Fading Channel %
% User data Rate : 9600 bps %
% Study Case IV : Coding Analysis of IS-95 Standard %
% at Flat Reyleigh-Fading Channel %

```

```

% Author :Meishkafadiah Alkaff
% 0222052
% bdg, October 06
%=====

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

%--- Set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.
shortPNcode = LFSR('Short PN code');

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');
longPNcode = longPNcode(1:64:end);

%generate walsh code & choose the eighth row.
WalshCode = hadamard(64);
WalshCode = WalshCode(8,:);

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        [TxSignal,Code] = IS95_Transmitter(TxData);
        %--- Reyleigh Fading Channel ---%
        FadedNoisy = ReyleighFading(TxSignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        [RxData,Deinter] = IS95_Receiver(FadedNoisy);
        %--- Error Estimation ---%
        %after Viterbi decoding.
        error_a = sum(TxData ~= RxData);
        BER_a(iters) = error_a./length(RxData);
        %befor Viterbi decoding.
        error_b = sum(Code ~= Deinter);
    end
end

```

```

        BER_b(iters) = error_b./length(Code);
    end
    %--- Average Simulation BER ---%
    AvBER_a(EbNoIndex) = sum(BER_a)/nIters;
    AvBER_b(EbNoIndex) = sum(BER_b)/nIters;
end

%--- Save in file ---%
save AvBER_a.mat
save AvBER_b.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at Reyleigh-Fading Channel')
semilogy(EbNo_dB,AvBER_a,'ok-');
hold on; grid on;
semilogy(EbNo_dB,AvBER_b,'dk-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('With Channel Coding', 'Without Channel Coding');

%=====
% Simulation of IS-95 Standard for Reverse Link %
% Channel model : AWGN & Fading Channel %
% User data Rate : 9600 bps %
% Study Case IV : Coding Analysis of IS-95 Standard %
% at Flat Reyleigh-Fading Channel %
% %
% Author :Meishkafadiah Alkaff %
% 0222052 %
% bdg, October 06 %
%=====

%--- clear current workspaces & close any opened window ---%
clear all
close all
clc

%--- Set as global parameter ---%
global EbNo_dB shortPNcode longPNcode WalshCode

%--- Parameter ---%
EbNo_dB = -2:.5:8; %Eb/No measuring range [in dB]
nIters = 50; %number of iteration

%--- Initialization ---%
%generate short PN code.

```



```

shortPNcode = LFSR('Short PN code');

%generate long PN code & decimation.
longPNcode = LFSR('Long PN code');

%generate 64x64 Hadamard matrix.
WalshCode = hadamard(64);

%--- Simulation loop ---%
for EbNoIndex = 1:length(EbNo_dB),
    for iters = 1:nIters,
        %--- Data Source ---%
        TxData = Data_Source('random');
        %--- IS-95 Transmitter ---%
        [Txsignal,Code] = IS95_Transmitter(TxData);
        %--- Reyleigh Fading & AWGN Channel ---%
        FadedNoisy = ReyleighFading(Txsignal,EbNoIndex);
        %--- IS-95 Receiver ---%
        [RxData,Deinter] = IS95_Receiver(FadedNoisy);
        %--- Error Estimation ---%
        %after Viterbi decoding.
        error_a = sum(TxData ~= RxData);
        BER_a(iters) = error_a./length(RxData);
        %befor Viterbi decoding.
        error_b = sum(Code ~= Deinter);
        BER_b(iters) = error_b./length(Code);
    end
    %--- Average Simulation BER ---%
    AvBER_a(EbNoIndex) = sum(BER_a)/nIters;
    AvBER_b(EbNoIndex) = sum(BER_b)/nIters;
end

%--- Save in file ---%
save AvBER_a.mat
save AvBER_b.mat

%--- Plot simulation result ---%
figure('name','Performance IS-95 at Reyleigh-Fading Channel')
semilogy(EbNo_dB,AvBER_a,'sk-');
hold on; grid on;
semilogy(EbNo_dB,AvBER_b,'dk-');
title('BER curve as function from Eb/No');
xlabel('Eb/No [dB]');
ylabel('BER');
legend('With Channel Coding', 'Without Channel Coding');

```