Appendix C

Uplink Transmitter by Using Matlab v6.5

Program Description

Output:

- Transmitted signal before passing through the multipath channel model.

```
%%%%%%%%%%%%%%%%%%
%%% References %%%
%%%%%%%%%%%%%%%%%%

%%% Case 1: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Test when information bit rate at 12.2 kbps %%
% DPDCH = 60 kbps, slot # 2, SF = 64, bits/frame = 600, N_data = 40 bits/slot
% DPCCH = 15 kbps, slot # 0, SF = 256, bits/frame = 150, N_pilot = 6 bits/slot,
% N_tpc = 2, N_tfci = 2, N_fbi = 0
% power ratio = DPCCH/DPDCH = -5.46 dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Case 2: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Test when information bit rate at 64 kbps %%
% DPDCH = 240, slot # 4, SF = 16, bits/frame = 2400, N_data = 160 bits
% DPCCH = 15, slot # 0, SF = 256, bits/frame = 150, N_pilot = 6 bits,
% N_tpc = 2, N_tfci = 2, N_fbi = 0
% power ratio = DPCCH/DPDCH = -9.54 dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Case 3: %%%
%% Test when information bit rate at 144 kbps %%
% DPDCH = 480, slot # 5, SF = 8, bits/frame = 4800, N_data = 320 bits
% DPCCH = 15, slot # 0, SF = 256, bits/frame = 150, N_pilot = 6 bits,
% N_tpc = 2, N_tfci = 2, N_fbi = 0
% power ratio = DPCCH/DPDCH = -11.48 dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Case 4: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Test when information bit rate at 384 kbps %%
% DPDCH = 960, slot # 6, SF = 4, bits/frame = 9600, N_data = 640 bits
% DPCCH = 15, slot # 0, SF = 256, bits/frame = 150, N_pilot = 6 bits,
% N_tpc = 2, N_tfci = 2, N_fbi = 0
% power ratio = DPCCH/DPDCH = -11.48 dB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Note: 1 frame = 15 slots
%% N_pilot = 6 %%
%% N_pilot --> TFCI --> FBI --> TPC %%

pilot_slot_0 = [1 1 1 1 1 0]; pilot_slot_1 = [1 0 0 1 1 0];
pilot_slot_2 = [1 0 1 1 0 0]; pilot_slot_3 = [1 0 0 1 0 0];
pilot_slot_4 = [1 1 0 1 0 1]; pilot_slot_5 = [1 1 1 1 1 0];
pilot_slot_6 = [1 1 1 1 0 0]; pilot_slot_7 = [1 1 0 1 0 0];
pilot_slot_8 = [1 0 1 1 1 0]; pilot_slot_9 = [1 1 1 1 1 1];
pilot_slot_10 = [1 0 1 1 0 1]; pilot_slot_11 = [1 1 0 1 1 1];
pilot_slot_12 = [1 1 0 1 0 0]; pilot_slot_13 = [1 0 0 1 1 1];
pilot_slot_14 = [1 0 0 1 1 1];

%%%%%%%%%%%%%%%%%%%%%
%%%% Transmitter %%%%
%%%%%%%%%%%%%%%%%%%%%

% defined variables (can choose from follow the specs)
N = 10; % no. of frames
bpfi = 600; % bits/frame of I
sfi = 64; % spreading factor of I
bpfq = 150; % bits/frame of Q
sfq = 256; % spreading factor of Q
gain_Q = 10^(-5.46/10); % Power in Watts follow specs

%% I_data --> OVSF (spreading) --> gain --> I_channel %%
```

```
% I channel DPDCH (data channel)

I_data = round(rand(1,N.*bpfi));
I_binary = -2*(I_data)+1; % mapped to binary

% OVSF for I %
OVSF_I = dlmread('C:\MATLAB6p5\work\ovsf_64_16.txt');

I_sf = reshape(((I_binary)'*OVSF_I)',1,N.*bpfi.*sfi);
gain_I = 1;
I_part = I_sf*gain_I;

%% Q_data --> OVSF (spreading) --> gain --> Q_channel %%
% Q channel DPCCH (control channel)

%% slot formats %%
TPC = round(rand(15,2));
TFCI = round(rand(15,2));

slot_0 = [pilot_slot_0 TFCI(1,:) TPC(1,:)];

slot_1 = [pilot_slot_1 TFCI(2,:) TPC(2,:)];

slot_2 = [pilot_slot_2 TFCI(3,:) TPC(3,:)];

slot_3 = [pilot_slot_3 TFCI(4,:) TPC(4,:)];

slot_4 = [pilot_slot_4 TFCI(5,:) TPC(5,:)];

slot_5 = [pilot_slot_5 TFCI(6,:) TPC(6,:)];

slot_6 = [pilot_slot_6 TFCI(7,:) TPC(7,:)];

slot_7 = [pilot_slot_7 TFCI(8,:) TPC(8,:)];

slot_8 = [pilot_slot_8 TFCI(9,:) TPC(9,:)];

slot_9 = [pilot_slot_9 TFCI(10,:) TPC(10,:)];

slot_10 = [pilot_slot_10 TFCI(11,:) TPC(11,:)];

slot_11 = [pilot_slot_11 TFCI(12,:) TPC(12,:)];

slot_12 = [pilot_slot_12 TFCI(13,:) TPC(13,:)];

slot_13 = [pilot_slot_13 TFCI(14,:) TPC(14,:)];

slot_14 = [pilot_slot_14 TFCI(15,:) TPC(15,:)];

sfm = [slot_0 slot_1 slot_2 slot_3 slot_4 slot_5 slot_6 slot_7
slot_8 slot_9 slot_10 slot_11 slot_12 slot_13 slot_14];

Q_data = [];

for frame = 1:N
    Q_data = [Q_data sfm];
end

Q_binary= -2*(Q_data)+1; % mapped to binary

% OVSF for Q %
OVSF_Q = dlmread('C:\MATLAB6p5\work\ovsf_256_0.txt');

Q_sf = reshape(((Q_binary)'*OVSF_Q)',1,N.*sfq.*bpfq);
```

```
Q_part = Q_sf*gain_Q;

% Summation I&Q (I+jQ) %
iq_sum = I_part + (Q_part)*j;

% scrambling code (Gold code)
scramb = []; for n = 1:N
    scramb = [scramb c_long2];
end

% multiplied with scrambling code in complex form %
s = iq_sum.*scramb;

fid = fopen('Tx_real.txt','w');
fprintf(fid,'%f ',(real(s).'));
fclose(fid);

fid = fopen('Tx_imag.txt','w');
fprintf(fid,'%f ',(imag(s).'));
fclose(fid);

fid = fopen('I_binary.txt','w');
fprintf(fid,'%f ',(I_binary.'));
fclose(fid);

fid = fopen('Q_binary.txt','w');
fprintf(fid,'%f ',(Q_binary.'));
fclose(fid);
```