
Part 1: SSB Generation

Table of Contents

| | |
|-------------------------------------|---|
| Parameters | 1 |
| PSS | 1 |
| SSS | 1 |
| PBCH Pilot | 2 |
| PBCH Data | 2 |
| Concatenation | 2 |
| Channel and Noise | 2 |
| (a) $s(n)$ Transmitted Signal | 3 |
| (b) $y(n)$ Received Signal | 3 |
| Function Definitions | 4 |

Parameters

```
FFT_size = 4096;
CP_length = 288;
CP_OFDM_length = FFT_size+CP_length;
SCS = 30e3;
Ts = 1/FFT_size/SCS;
delta_f = 2.03e3; % kHz
```

```
QAM_mod = 4;
num_sc = 240;
N_id_1 = 77;
N_id_2 = 2;
c_init = 120897;
```

```
SNR = 20;
h = [0 0 0 0 1 0.5]';
```

PSS

```
% OFDM Modulation
PSS_stream = generate_PSS_BPSK(N_id_2);
% Map symbol to subcarrier
d_PSS = [zeros(56,1);PSS_stream;zeros(FFT_size-183,1)];
% FFT
OFDM_PSS_body = ifft(d_PSS)*sqrt(FFT_size);
% Add CP
CP_OFDM_PSS = [OFDM_PSS_body(end-CP_length+1:end);OFDM_PSS_body];
```

SSS

```
% OFDM Modulation
SSS_stream = generate_SSS_BPSK(N_id_1, N_id_2);
```

```
% Map symbol to subcarrier
d_SSS = [zeros(56,1);SSS_stream;zeros(FFT_size-183,1)];
% FFT
OFDM_SSS_body = ifft(d_SSS)*sqrt(FFT_size);
% Add CP
CP_OFDM_SSS = [OFDM_SSS_body(end-CP_length+1:end);OFDM_SSS_body];
```

PBCH Pilot

```
% OFDM Modulation
PBCH_pilot_stream = generate_PBCH_pilot(c_init);
% Map symbol to subcarrier
d_PBCH_pilot = zeros(FFT_size,1);
k = 0:59;
d_PBCH_pilot(1+4*k+1) = PBCH_pilot_stream;
% FFT
OFDM_PBCH_pilot_body = ifft(d_PBCH_pilot)*sqrt(FFT_size);
% Add CP
CP_OFDM_PBCH_pilot = [OFDM_PBCH_pilot_body(end-
CP_length+1:end);OFDM_PBCH_pilot_body];
```

PBCH Data

```
% OFDM Modulation
PBCH_data_stream = generate_PBCH_data(num_sc, QAM_mod);
% Map symbol to subcarrier
d_PBCH_data = [PBCH_data_stream;zeros(FFT_size-num_sc,1)];
% FFT
OFDM_PBCH_data_body = ifft(d_PBCH_data)*sqrt(FFT_size);
% Add CP
CP_OFDM_PBCH_data = [OFDM_PBCH_data_body(end-
CP_length+1:end);OFDM_PBCH_data_body];
```

Concatenation

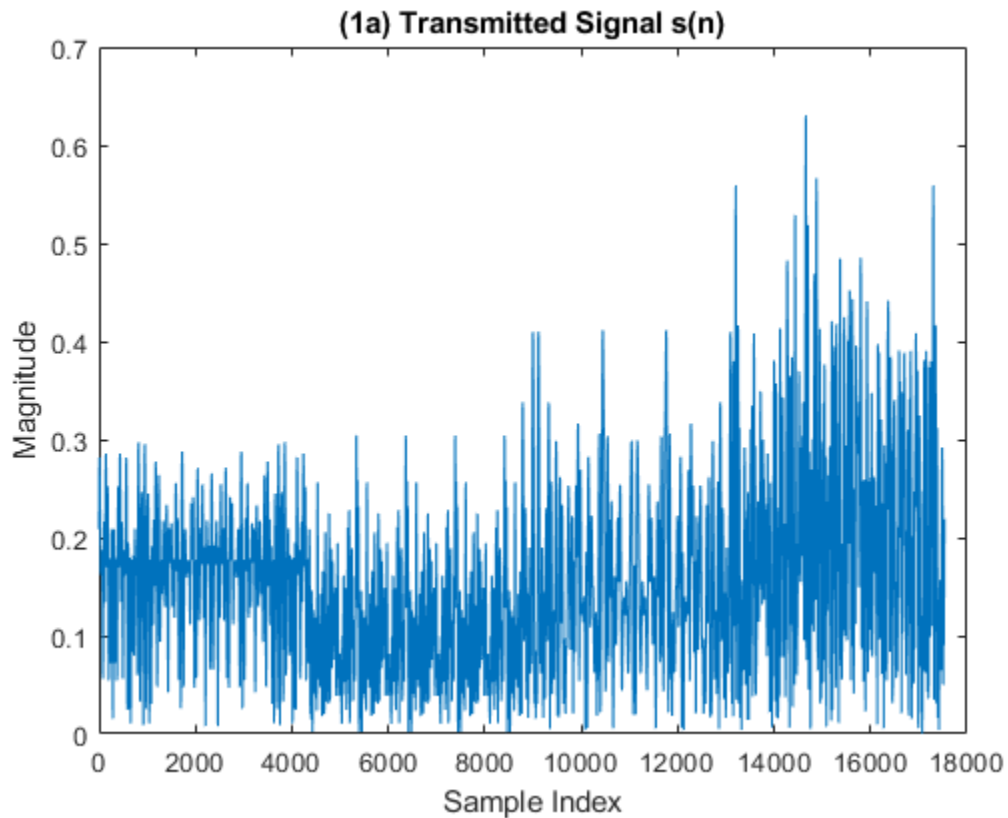
```
CP_OFDM_chain = [CP_OFDM_PSS; CP_OFDM_PBCH_pilot; CP_OFDM_SSS;
CP_OFDM_PBCH_data];
```

Channel and Noise

```
signal_after_channel = conv(CP_OFDM_chain,h);
N_0 = 10^(-SNR/10) * (norm(signal_after_channel)^2/
length(signal_after_channel));
noise = sqrt(N_0/2)*(randn(length(signal_after_channel),1) +
1j*randn(length(signal_after_channel),1));
received_signal = signal_after_channel .*
exp(1j*2*pi*delta_f*(0:length(signal_after_channel)-1)*Ts) + noise;
```

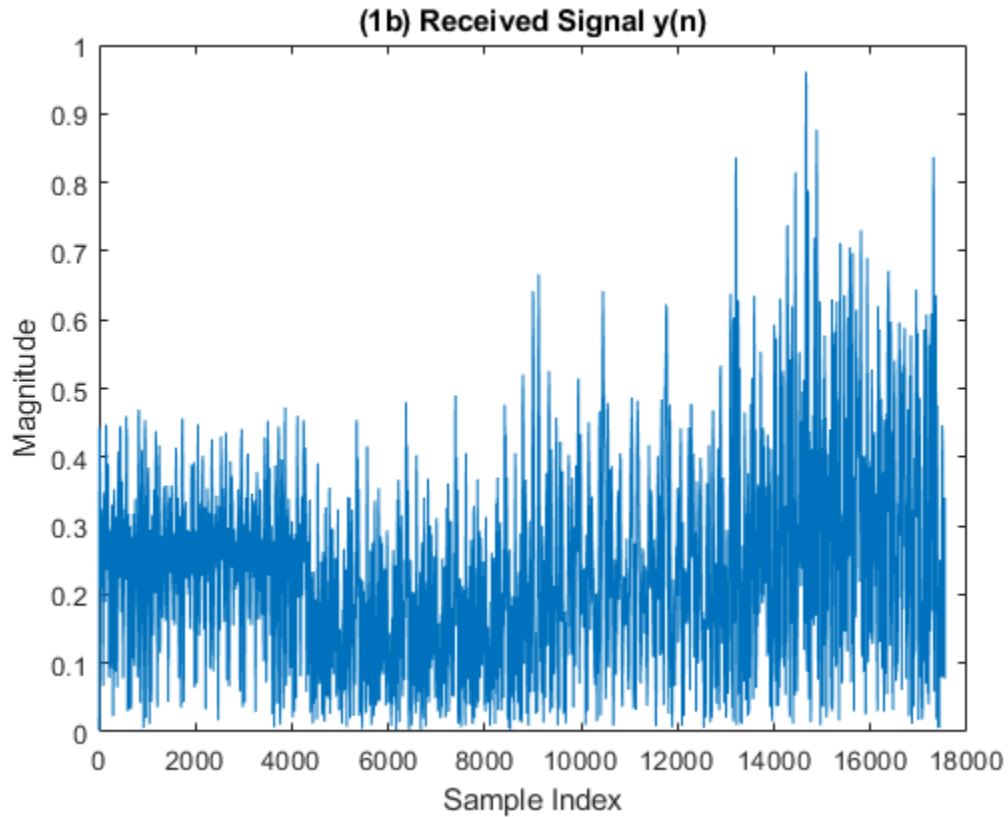
(a) $s(n)$ Transmitted Signal

```
figure;  
plot(abs(CP_OFDM_chain));  
xlabel('Sample Index'); % X-axis label  
ylabel('Magnitude'); % Y-axis label  
title('(1a) Transmitted Signal s(n)');
```



(b) $y(n)$ Received Signal

```
figure;  
plot(abs(received_signal));  
xlabel('Sample Index'); % X-axis label  
ylabel('Magnitude'); % Y-axis label  
title('(1b) Received Signal y(n)');
```



Function Definitions

```
function PSS_BPSK = generate_PSS_BPSK(N_id_2)
    x = zeros(127,1);
    PSS_BPSK = zeros(127,1);
    x_init = [0 1 1 0 1 1 1];
    x(1:7) = x_init;
    for i = 1:120
        x(i+7) = mod(x(i+4)+x(i),2);
    end
    for n = 0:126
        m = mod(n + 43*N_id_2,127);
        PSS_BPSK(n+1) = 1-2*x(m+1);
    end
end

function SSS_BPSK = generate_SSS_BPSK(N_id_1,N_id_2)
    x_0 = zeros(127,1);
    x_1 = zeros(127,1);
    SSS_BPSK = zeros(127,1);
    x_init = [1 0 0 0 0 0 0];
    x_0(1:7) = x_init;
    x_1(1:7) = x_init;

    for i = 1:120
```

```
x_0(i+7) = mod(x_0(i+4)+x_0(i),2);
x_1(i+7) = mod(x_1(i+1)+x_1(i),2);
end
for n = 0:126
    m_0 = mod(n + 15* floor(N_id_1/112) + 5*N_id_2,127);
    m_1 = mod(n + mod(N_id_1,112),127);
    SSS_BPSK(n+1) = (1-2*x_0(m_0+1))*(1-2*x_1(m_1+1));
end
end

function QPSK_pilot_stream = generate_PBCH_pilot(c_init)
    c = zeros(120,1);
    QPSK_pilot_stream = zeros(60,1);
    x_1 = zeros(1800,1);
    x_2 = zeros(1800,1);
    x_1_init = [1; zeros(30,1)];
    x_1(1:31) = x_1_init;
    x_2_init = zeros(31,1);
    x_2_init_char = dec2bin(c_init);
    for i = 1:length(x_2_init_char)
        x_2_init(length(x_2_init_char)-i+1) = str2double(x_2_init_char(i));
    end
    x_2(1:31) = x_2_init;
    for n = 1:1800
        x_1(n+31) = mod(x_1(n+3)+x_1(n),2);
        x_2(n+31) = mod(x_2(n+3)+x_2(n+2)+x_2(n+1)+x_2(n),2);
    end
    for n = 0:119
        c(n+1) = mod(x_1(n+1600+1)+x_2(n+1600+1),2);
    end
    for n = 0:59
        QPSK_pilot_stream(n+1) = 1/sqrt(2)*(1-2*c(2*n+1)) + 1j/
sqrt(2)*(1-2*c(2*n+2));
    end
end

function QPSK_data_stream = generate_PBCH_data(num_sc, QAM_mod)
    data_bit_stream = randi([0 1],num_sc*log2(QAM_mod),1);
    QPSK_data_stream =
qammod(data_bit_stream,QAM_mod,InputType='bit',UnitAveragePower=true);
end
```

Published with MATLAB® R2024a