
Part 3: Timing synchronization

Table of Contents

Parameters	1
PSS and PBCH_pilot	1
Concatenation	1
Channel and Noise	2
Timing Synchronization	2
(a) Plot Correlation for all SNR values	2

Parameters

```
FFT_size = 4096;
CP_length = 288;
SCS = 30e3;
Ts = 1/FFT_size/SCS;
CP_OFDM_length = FFT_size+CP_length;
num_sc = 240;
N_id_2 = 2;
c_init = 120897;
```

PSS and PBCH_pilot

```
% OFDM Modulation
PSS_stream = PSS_BPSK(N_id_2);
PBCH_pilot_stream = PBCH_QPSK(c_init);

% Map symbol to subcarrier
d_PSS = [zeros(56,1);PSS_stream;zeros(FFT_size-183,1)];
d_PBCH_pilot = zeros(FFT_size,1);
k = 0:59;
d_PBCH_pilot(1+4*k+1) = PBCH_pilot_stream(1:60);

% FFT
OFDM_PSS_body = ifft(d_PSS)*sqrt(FFT_size);
OFDM_PBCH_pilot_body = ifft(d_PBCH_pilot)*sqrt(FFT_size);

% Add CP
CP_OFDM_PSS = [OFDM_PSS_body(end-CP_length+1:end);OFDM_PSS_body];
CP_OFDM_PBCH_pilot = [OFDM_PBCH_pilot_body(end-CP_length+1:end);OFDM_PBCH_pilot_body];
```

Concatenation

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

Channel and Noise

```
h = [0 0 0 0 1 0.5]';
signal_after_channel = conv(CP_OFDM_chain,h);

SNR_values = [-5, 20]; % SNR values in dB
corr_all = zeros(length(SNR_values), FFT_size + 1);

for snr_idx = 1:length(SNR_values)

    SNR = SNR_values(snr_idx);
    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 + noise;
```

Timing Synchronization

```
corr = zeros(FFT_size+1,1);
for m = 0:FFT_size
    corr(m+1) = abs(received_signal(m+1:m+CP_OFDM_length)' *
CP_OFDM_PSS);
end
[~,m_STO] = max(corr);
m_STO = m_STO - 1; % adjusting the index
disp(['mSTO for SNR = ', num2str(SNR), ' dB: ', num2str(m_STO)]);

corr_all(snr_idx, :) = corr;

mSTO for SNR = -5 dB: 4
mSTO for SNR = 20 dB: 4

end
```

(a) Plot Correlation for all SNR values

```
figure;
plot(0:FFT_size, corr_all(1, :), 'r', 'DisplayName', 'SNR = -5 dB');
hold on;
plot(0:FFT_size, corr_all(2, :), 'b', 'DisplayName', 'SNR = 20 dB');
xlabel("Timing Offset (sample)")
ylabel("Correlation")
title('(3a) Correlation for Different SNR Values')
legend show;
grid on;

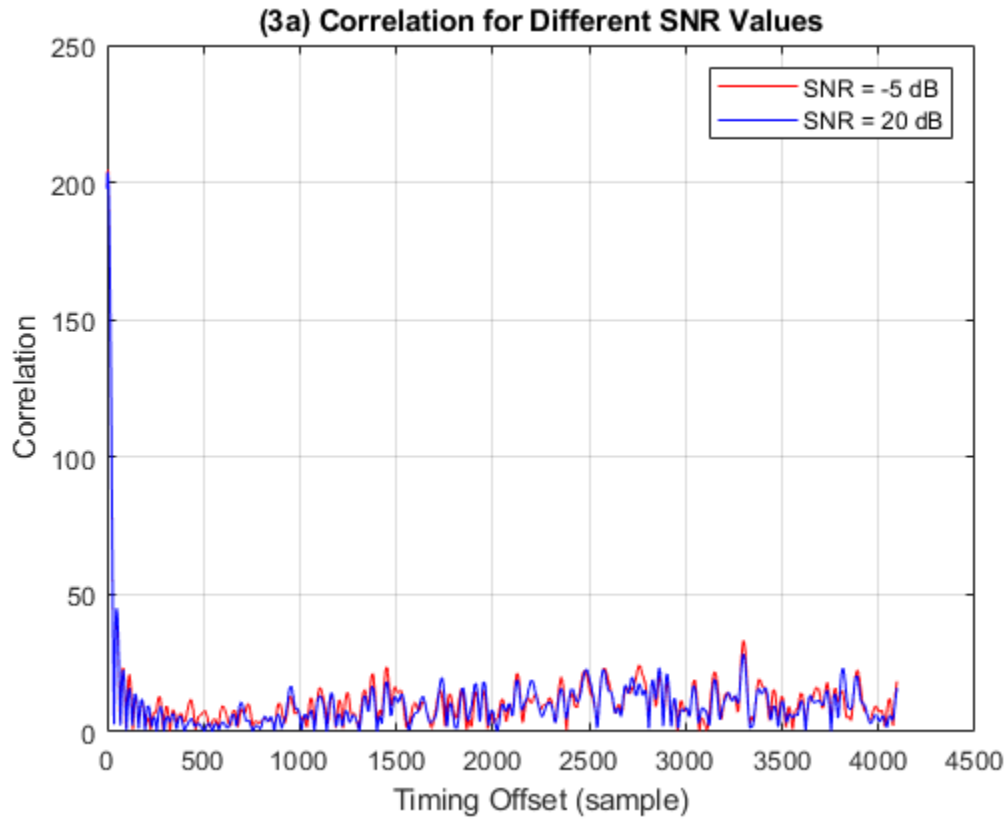
function BPSK_stream = PSS_BPSK(N_id_2)
    x = zeros(127,1);
    BPSK_stream = zeros(127,1);
    x_init = [0 1 1 0 1 1 1];
```

```

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);
    BPSK_stream(n+1) = 1-2*x(m+1);
end
end

function QPSK_stream = PBCH_QPSK(c_init)
    c = zeros(120,1);
    QPSK_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_stream(n+1) = 1/sqrt(2)*(1-2*c(2*n+1)) + 1j/
sqrt(2)*(1-2*c(2*n+2));
    end
end
end

```



Published with MATLAB® R2024a