LTE -PILOT ESTIMATION

Yara Mohsen Mahmoud
ITI intake41 | wireless communication | 7 April 2021

```
% clear all;
rand('seed',7020);
E_N_dB=0:2:20;
E_N=10.^(E_N_dB/10);


%%data
bitsNum=10000;
num_of_ch=72;
pilot=3:6:69;
sig=1:72;

index_data=1:72;
bits =randsrc(num_of_ch,bitsNum,[0 1 ;0.5 0.5]);
Symb =bits*2-1;
N0 =randn(167,bitsNum)+1i*randn(167,bitsNum);


for t=3:6:69
    Symb(t,:)=1;
    index_data(1,t)=0;
end

index_data2=index_data(index_data>0);
index_data2=reshape(index_data2,60,1);


BER_OFDM_pilot1=[];
BER_OFDM_pilot2=[];
BER_OFDM_pilot3=[];
BER_OFDM_pilot4=[];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%1

for v=0:2:20
    error=0;
    error1=0;

    N=1/(10^(v/10));
    n=N0.*sqrt(N/2);

   for k=1:bitsNum
    S=bits(:,k);
    S_no_pilot=S(index_data2);
    S=reshape(S,1,72);
    S_no_pilot=reshape(S_no_pilot,1,60);


    H=sqrt(1/2)*sqrt(1/8)*(randn(8,1)+1i*randn(8,1));

    %reshape the data to 128 bit and add zeros
    TX_B=zeros(128,1);
    TX_B(1:36,1)=Symb(37:72,k);
```

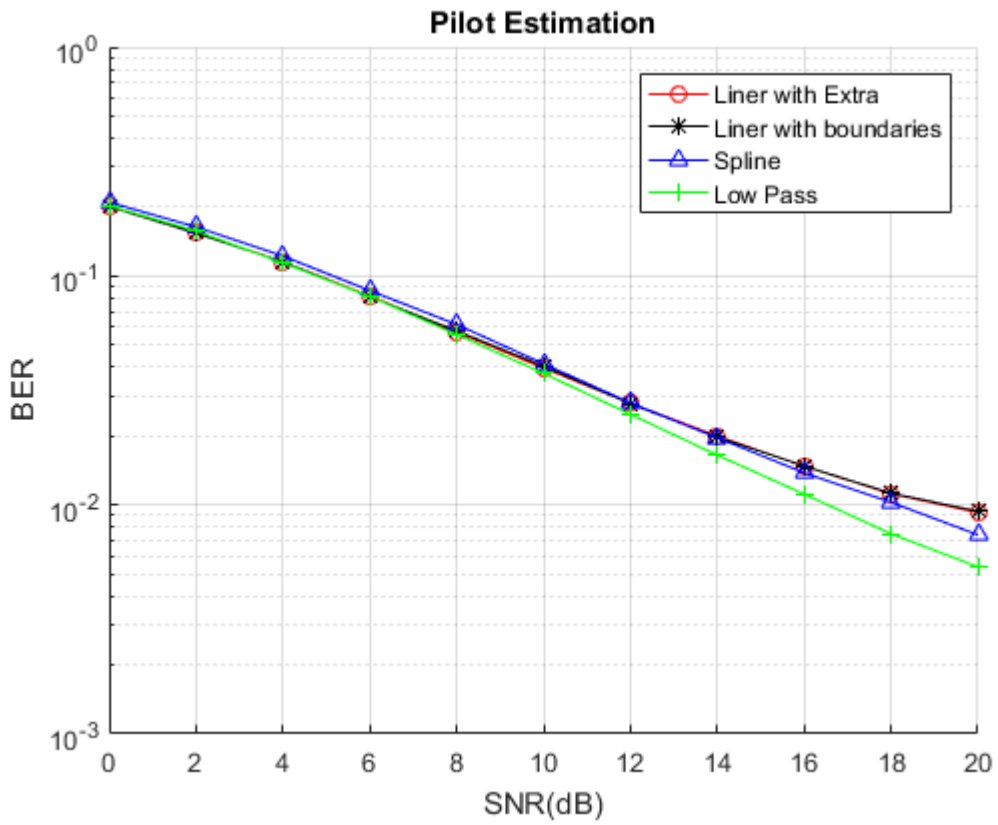
```
    TX_B(93:end,1)=Symb(1:36,k);
    TX_B_ifft = sqrt(128)*ifft(TX_B,128);

    %add cp to be 160
    cp=TX_B_ifft(97:128,1);
    signal_with_cp=[cp;TX_B_ifft];

    %convolution
    output_signal= conv(signal_with_cp,H) +n(:,k);

    %remove cp
    output_without_cp = output_signal(33:160,1);
    RX_B_fft= fft(output_without_cp);

    %THE ORIGINAL SIGNAL
    RX_B_fft_original=[RX_B_fft(93:128,1);RX_B_fft(1:36,1)];

    %the pilot
     H_estimate = RX_B_fft_original(pilot,:);

    slop_h = interp1(pilot,H_estimate,sig,'linear','extrap');


    RX_B_fft_original=reshape(RX_B_fft_original,1,72);
    final_sig=real(RX_B_fft_original./slop_h);
    final_sig_no_pilot=final_sig(index_data2);
    final_sig_no_pilot(final_sig_no_pilot<0)=0;
    final_sig_no_pilot(final_sig_no_pilot>0)=1;

    error = error+( sum(final_sig_no_pilot~=S_no_pilot));

     end
  error1=error/(72*bitsNum);
  BER_OFDM_pilot1=[BER_OFDM_pilot1 error1] ;
end
```

```
%%2
for v=0:2:20
    error=0;
    error1=0;

    N=1/(10^(v/10));
    n=N0.*sqrt(N/2);

   for k=1:bitsNum
    S=bits(:,k);
    S_no_pilot=S(index_data2);
    S=reshape(S,1,72);
    S_no_pilot=reshape(S_no_pilot,1,60);

    H=sqrt(1/2)*sqrt(1/8)*(randn(8,1)+1i*randn(8,1));

    %reshape the data to 128 bit and add zeros
    TX_B=zeros(128,1);
    TX_B(1:36,1)=Symb(37:72,k);
    TX_B(93:end,1)=Symb(1:36,k);
    TX_B_ifft = sqrt(128)*ifft(TX_B,128);

    %add cp to be 160
    cp=TX_B_ifft(97:128,1);
    signal_with_cp=[cp;TX_B_ifft];

    output_signal= conv(signal_with_cp,H) +n(:,k);

    %remove cp
    output_without_cp = output_signal(33:160,1);
    RX_B_fft= fft(output_without_cp);

    %THE ORIGINAL SIGNAL
    RX_B_fft_original=[RX_B_fft(93:128,1);RX_B_fft(1:36,1)];

    %the pilot
    H_estimate = RX_B_fft_original(pilot,:);
    slop_h = interp1(pilot,H_estimate,sig);
    slop_h(1,70:72) = RX_B_fft_original(69,1);
    slop_h(1,1:3) = RX_B_fft_original(3,1);

    RX_B_fft_original=reshape(RX_B_fft_original,1,72);
    final_sig=real(RX_B_fft_original./slop_h);
    final_sig_no_pilot=final_sig(index_data2);
    final_sig_no_pilot(final_sig_no_pilot<0)=0;
    final_sig_no_pilot(final_sig_no_pilot>0)=1;

    error = error+( sum(final_sig_no_pilot~=S_no_pilot));
     end
    error1=error/(72*bitsNum);

  BER_OFDM_pilot2=[BER_OFDM_pilot2 error1] ;
end
```

```
%%3

for v=0:2:20
    error=0;
    error1=0;

    N=1/(10^(v/10));
    n=N0.*sqrt(N/2);

   for k=1:bitsNum
    S=bits(:,k);
    S_no_pilot=S(index_data2);
    S=reshape(S,1,72);
    S_no_pilot=reshape(S_no_pilot,1,60);

    H=sqrt(1/2)*sqrt(1/8)*(randn(8,1)+1i*randn(8,1));

    %reshape the data to 128 bit and add zeros
    TX_B=zeros(128,1);
    TX_B(1:36,1)=Symb(37:72,k);
    TX_B(93:end,1)=Symb(1:36,k);

    TX_B_ifft = sqrt(128)*ifft(TX_B,128);

    %add cp to be 160
    cp=TX_B_ifft(97:128,1);
    signal_with_cp=[cp;TX_B_ifft];

    %convolution
    output_signal= conv(signal_with_cp,H) +n(:,k);

    %remove cp
    output_without_cp = output_signal(33:160,1);
    RX_B_fft= fft(output_without_cp);

    %THE ORIGINAL SIGNAL
    RX_B_fft_original=[RX_B_fft(93:128,1);RX_B_fft(1:36,1)];

    %the pilot
     H_estimate = RX_B_fft_original(pilot,:);
    slop_h = interp1(pilot,H_estimate,sig,'spline');

    RX_B_fft_original=reshape(RX_B_fft_original,1,72);
    final_sig=real(RX_B_fft_original./slop_h);
    final_sig_no_pilot=final_sig(index_data2);
    final_sig_no_pilot(final_sig_no_pilot<0)=0;
    final_sig_no_pilot(final_sig_no_pilot>0)=1;

    error = error+( sum(final_sig_no_pilot~=S_no_pilot));
     end
  error1=error/(72*bitsNum);
  BER_OFDM_pilot3=[BER_OFDM_pilot3 error1] ;
end
```

```
%% 4-low pass %%%%%%%%%%%%%%%%%%%%%%%

for v=0:2:20

    error=0;
    error1=0;

    N=1/(10^(v/10));
    n=N0.*sqrt(N/2);

   for k=1:bitsNum
    S=bits(:,k);
    S_no_pilot=S(index_data2);
    S=reshape(S,1,72);

    H=sqrt(1/2)*sqrt(1/8)*(randn(8,1)+1i*randn(8,1));

    %reshape the data to 128 bit and add zeros
    TX_B=zeros(128,1);
    TX_B(1:36,1)=Symb(37:72,k);
    TX_B(93:end,1)=Symb(1:36,k);
    TX_B_ifft = sqrt(128)*ifft(TX_B,128);

    %add cp to be 160
    cp=TX_B_ifft(97:128,1);
    signal_with_cp=[cp;TX_B_ifft];

    %convolution
    output_signal= conv(signal_with_cp,H) +n(:,k);

    %remove cp
    output_without_cp = output_signal(33:160,1);
    RX_B_fft= fft(output_without_cp);

    %THE ORIGINAL SIGNAL
    RX_B_fft_original=[RX_B_fft(93:128,1);RX_B_fft(1:36,1)];

    %the pilot
     H_estimate = RX_B_fft_original(pilot,:);
    slop_h = interp(H_estimate,6);
    slop_h(68:end)=[];
    ch=[H_estimate(1);H_estimate(1);slop_h;H_estimate(12);H_estimate(12);H_estimate(12)];

     %RX_B_fft_original=reshape(RX_B_fft_original,1,72);
    final_sig=real(RX_B_fft_original./ch);
    final_sig_no_pilot=final_sig(index_data2);
    final_sig_no_pilot(final_sig_no_pilot<0)=0;
    final_sig_no_pilot(final_sig_no_pilot>0)=1;

    error = error+( sum(final_sig_no_pilot~=S_no_pilot));

     end
```

```
  error1=error/(72*bitsNum);
  BER_OFDM_pilot4=[BER_OFDM_pilot4 error1] ;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fig = figure;
grid on;
hold on;
ax = fig.CurrentAxes;
ax.YScale = 'log';
xlabel('SNR(dB)');
ylabel('BER');
title('Pilot Estimation');

snr= 10*log10(E_N); %% snr_dB
semilogy(snr,BER_OFDM_pilot1,'r-o');
hold on
semilogy(snr,BER_OFDM_pilot2,'k-*');
hold on
semilogy(snr,BER_OFDM_pilot3,'b-^');
hold on
semilogy(snr,BER_OFDM_pilot4,'g-+');

legend('Liner with Extra','Liner with boundaries','Spline','Low Pass');
```



The 4 curves results are very near from each other's. The curve of the channel estimation of linear interpolation with extrapolation and the channel estimation with the linear interpolation where edge subcarriers are estimated to be the same as the closest pilot are the same.

From curve we can see that channel estimation with low-pass interpolation achieves the best error rate performance than other interpolation algorithms used while the linear interpolation shows worst performance in estimation.

It may result from the fact that lowpass interpolation does minimization of the mean-square error between the interpolated points and their ideal values