MIMO System Identification and Deconvolution

Author:

Pistone Santi Pier 10867402

Topic of HW

MIMO-OFDM (Multiple Input Multiple Output - Orthogonal Frequency Division Multiplexing), is a new wireless broadband technology has gained great popularity for its capability of high rate transmission and its toughness against multipath fading and other channel impairments. In MIMO, multiple antennas are employed both at the transmitter and the receiver. Various signals are transmitted from different antennas at the transmitter using the same frequency and separated in space. In this paper, we analyze and implement various channel estimation techniques such as Least Squares (LS), Minimum Mean Square Error (MMSE) for MIMO-OFDM System. The MSE (Mean Square Error) performance characteristics of channel are investigated for BPSK, M-ary QAM modulation schemes over the AWGN and Rayleigh fading channel. The topic of the HW is to get practice and evaluate the MSE in MIMO estimation and deconvolution using Matlab. We'll use the QPSK modulation that semplify the problem of estimation and deconvolution for multi-carrier communication system.

Introduction

The MIMO channel can be general rapresented as

$$y_i[k] = \sum_{\ell=0}^{N} h_{i,\ell}[k] * x_\ell[k] + w_i[k],$$

or more explicitly :

$$\begin{bmatrix} y_1[k] \\ \vdots \\ y_M[k] \end{bmatrix} = \begin{bmatrix} h_{11}[k] & \dots & h_{1N}[k] \\ \vdots & \ddots & \vdots \\ h_{M1}[k] & \dots & h_{MN}[k] \end{bmatrix} * \begin{bmatrix} x_1[k] \\ \vdots \\ x_N[k] \end{bmatrix} + \begin{bmatrix} w_1[k] \\ \vdots \\ w_M[k] \end{bmatrix}$$

Where:

- Y[k] is the output of k-th sample
- H[k] is the filter
- X[k] is the data of k-th sample
- W[k] is the noise.

Objective

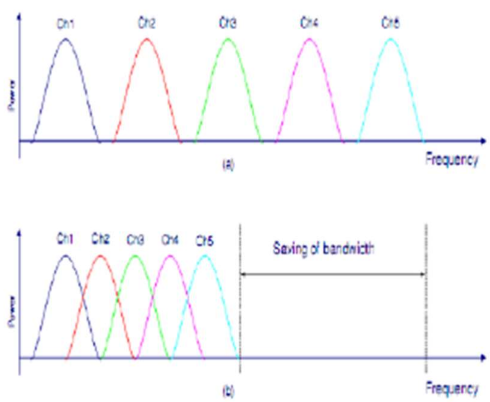
Our final objective is to estimate the MSE of LSE channel estimation and BER respect to SNR.

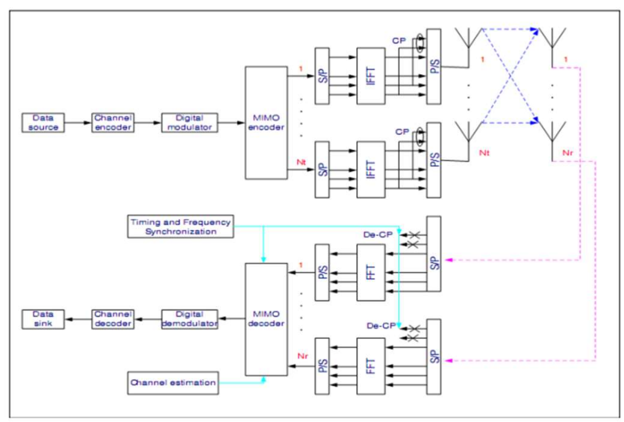
Theory explanation

MIMO-OFDM is used in 4G and 5G communication. We have N transmitter and M receiver. Is a combination between MIMO technology with ortogonal frequency division multiplexing. OFMD modulation divides main band into subchannel (128 in our case). The conversion of channel in k sub-channels allow us to semplify structure of receiver. This also allow us to increase the throughput of Tx-Rx system. OFMD is not only a frquency-modulation technique but a frequency-division multiplexing.

Example of difference between FDM and OFDM.

(FMD upper and OFDM lower)





Scheme of a MIMO OFDM transmission system

Basically we have MIMO encoder, Nt transmitter, serial-to-parallel conversion, pilot insertion, IFFT and cyclic prefix insertion; the receiver has a parallel-to-serial converter with a CP remover, synchronized in time and frequency and a decoder for channel estimation. The basic principle that underlies OFDM is the insertion of guard intervall (cyclic prefix) which is copy of last part of OFDM symbol. This modify the convolution from linear to cyclic convolution; the resulting overall transfer function can be diagonalized through the use of an IFFT at the transmitter and an FFT at the receiver. The main advantage of OFDM-MIMO are :

- High flexibility for link adaptation
- High spectral efficency
- Simple implementation

Code explanation

Parameters

In this project is possible to modify some parameters to customize the estimation. We can modify parameters like:

- Number of transmitters
- Number of receivers
- Number of blocks
- Number of subcarriers

```
ofdm.Nb      = 1e2;                  % number of blocks
ofdm.Nt      = 2;                    % number of transmit antenna
ofdm.Nr      = 3;                    % number of receive antenna
ofdm.K       = 128;                  % number of subcarriers
ofdm.G       = 1/4;                  % Guard interval percentage
ofdm.Mod     = 4;                    % QPSK Modulation
ofdm.PSpace  = 1;                    % pilot space between two pilots

% channel parameters
chan.SNR_dB  = 15;                   % signal to noise ratio
chan.L       = 6;                    % number of channel taps between each transmit-receive antenna
```

Normalization of energy for costellation

```
temp          = 0:ofdm.Mod-1;              % possible symbols
temp          = qammod(temp,ofdm.Mod);     % modulated symbols
temp          = abs(temp).^2;              % power of each point in the constellation
temp          = mean(temp);                % average energy of the constellation
ofdm.ModNorm = 1/sqrt(temp);               % normaliztion factor
```

Data generation

We start generate data from

```
ofdm.d        = randi(ofdm.Mod,ofdm.DL,ofdm.Nb,ofdm.Nt)-1;
```

The data is generated using randi function using

- Modulation: QSPK
- Length of data subcarriers
- Number of block
- Number of transmitter

Data Modulation

Now we start the modulation of data. We allocate the memory for block transmitted from each antenna. Then we modulate the data for each transmitter multiplying the normalization factor of the energy of costellation with the qammod of data generated and modulation type (in this case is a QPSK).

```
for nt = 1 : ofdm.Nt
    ofdm.dMod(ofdm.DPos,:,nt) = ofdm.ModNorm*qammod(ofdm.d(:,:,nt),ofdm.Mod);
end
```

Pilot insertion

```
%% Pilot insertion
   for nt = 1 : ofdm.Nt
       ofdm.dMod(ofdm.PPos,:,nt) = repmat(exp(-sqrt(-1)*2*pi*(nt-1)*chan.L*(1:ofdm.PL).'/ofdm.PL),1,ofdm.Nb);
   end

       ofdm.pow = var(ofdm.dMod(:))+abs(mean(ofdm.dMod(:)))^2;
```

For each transmitter we use repmat to creates a convolution matrix of pilot input; after that we calculate the energy of pilot, making sure the power is normalized (equal to 1).

IFFT

```
%% IFFT operation
    ofdm.ifft    = zeros(ofdm.K,ofdm.Nb,ofdm.Nt);
    for nt = 1 : ofdm.Nt
        ofdm.ifft(:,:,nt) = sqrt(ofdm.K)*ifft(ofdm.dMod(:,:,nt),ofdm.K);
    end
```

We allocate the memory for the ofdm blocks transmitted from each Tx antenna after ifft, then compute the ifft matrix multiplying the sqrt of number of subcarriers for ifft of data modulated matrix. This modify a linear convolution to cyclic convolution, increase semplicity of receiver.

```
% copy the end of signal to the begining of signal
ofdm.ifftG = [ofdm.ifft(ofdm.K*(1-ofdm.G)+1:ofdm.K,:,:);ofdm.ifft];
```

Channel

```
chan.Coeff = 1/sqrt(2)*1/sqrt(chan.L)*(randn(ofdm.Nt,ofdm.Nr,chan.L,ofdm.Nb)+sqrt(-1)*randn(ofdm.Nt,ofdm.Nr,chan.L,ofdm.Nb));
```

We generate for each block a rayleigh fading MIMO channel which is fixed over a block, then we define channel, filter and add noise.

```
ofdm.Y = zeros(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr);
for nb = 1 : ofdm.Nb
    for nt=1:ofdm.Nt
        for nr=1:ofdm.Nr
            ofdm.Y(:,nb,nr) = ofdm.Y(:,nb,nr) + filter(squeeze(chan.Coeff(nt,nr,:,nb)),1,ofdm.ifftG(:,nb,nt));
        end
    end
end
% adding noise
ofdm.Y = ofdm.Y + chan.sigma*1/sqrt(2)*(         randn(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr)+...
                                        sqrt(-1)*randn(ofdm.K*(1+ofdm.G),ofdm.Nb,ofdm.Nr)      );
```

Channel Estimation + Demodulation

```
ofdm.dDemod = zeros(ofdm.DL,ofdm.Nb,ofdm.Nt);
for nb = 1 : ofdm.Nb
    for dl = 1 : ofdm.DL
        ofdm.dDemod(dl,nb,:) = pinv(reshape(chan.CoeffEstFreq(ofdm.DPos(dl),:,:,nb),ofdm.Nt,ofdm.Nr).')...
                               *squeeze(ofdm.fft(ofdm.DPos(dl),nb,:));
    end
end
% detection
ofdm.dEst = zeros(ofdm.DL,ofdm.Nb,ofdm.Nt);
for nt = 1 : ofdm.Nt
    ofdm.dEst(:,:,nt) = qamdemod(1/ofdm.ModNorm * ofdm.dDemod(:,:,nt),ofdm.Mod);
end
% BER calculation
[~,ofdm.BER]  = biterr(ofdm.d(:),ofdm.dEst(:),log2(ofdm.Mod));
if ofdm.ifDisplayResults
    disp(['BER is = ',num2str(ofdm.BER)])
end
```

Then we calculate the MSE for channel in both way: Estimated (theory) and Simulated.

After that we demodulate, play detection and calculate the BER.

```
% building the first L columns of the fft matrix
F = dftmtx(ofdm.K);
F = F(:,1:chan.L);
% Memory allocation for the estimated channel coefficients
chan.CoeffEst = zeros(ofdm.Nt,ofdm.Nr,chan.L,ofdm.Nb);
for nb = 1 : ofdm.Nb
    for nr = 1 : ofdm.Nr
        % Building matrix A
        chan.A = zeros(ofdm.PL,chan.L*ofdm.Nt);
        for nt = 1 : ofdm.Nt
            chan.A(:,(1:chan.L)+(nt-1)*chan.L) = diag(ofdm.dMod(ofdm.PPos,nb,nt))*F(ofdm.PPos,:);
        end
        ChanEst = pinv(chan.A)*ofdm.fft(ofdm.PPos,nb,nr);
        for nt = 1 : ofdm.Nt
            chan.CoeffEst(nt,nr,:,nb) = ChanEst((1:chan.L)+(nt-1)*chan.L);
        end
    end
end
```

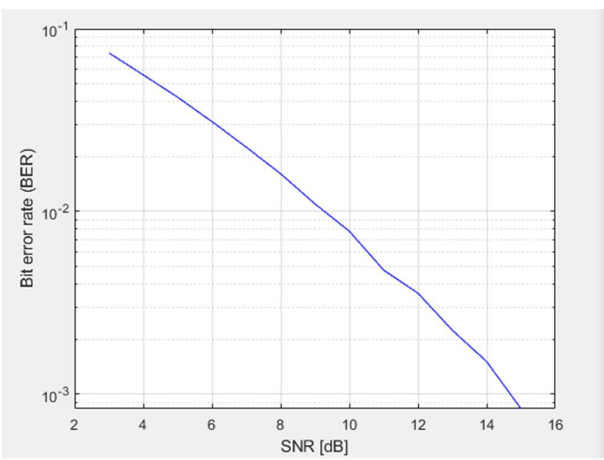
MC Estimation

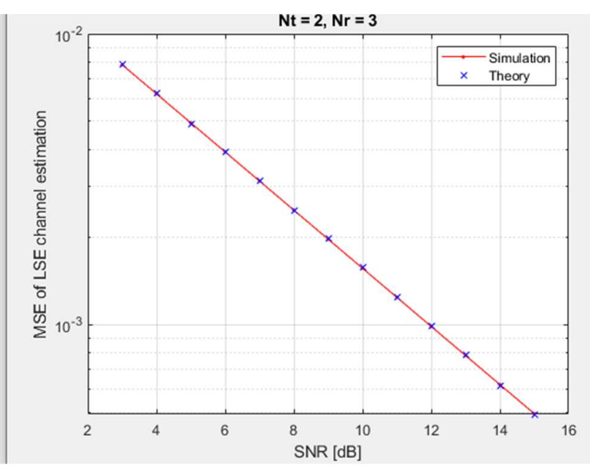
To reach an higher accuracy of MSE and BER we iterate a MC estimation .

```
cnt3 = 0;
for cnt1 = 1 : SNR_dBVL
    chanIn.SNR_dB              = SNR_dBV(cnt1); % load the SNR value
    for cnt2 = 1 : nMonteCarlo
        [ofdm chan] = MIMO_OFDM_LSE_CHAN_EST(ofdmIn,chanIn);
        MSE_CHAN_SIM(cnt2,cnt1) = chan.MSE_Simulation;
        MSE_CHAN_THR(cnt2,cnt1) = chan.MSE_Theory;
        MSE_CHAN_BER(cnt2,cnt1) = ofdm.BER;

        % update the loop counter
        cnt3 = cnt3 + 1;
        disp([num2str(round(cnt3/(SNR_dBVL*nMonteCarlo)*1000)/10),' is done...'])
    end
```

Output





Conclusion

MIMO-OFDM is a powerful technology that is the basis of 4G and 5G air interface. Is the most advanced technique for wireless LAN and mobile networks thank to spectral efficency and very high data throughput. Considering the Shannon upper bound limit we know that $C= \log_2(1 + (Ps/(Pi+Pn))$. We know that the throughput is equal to Spectral_efficency * Density * Available_Spectrum. MIMO allows us to increase the throughput in order to have ideally Pi -> 0 and allow multiple data-streams to co-exist in space division:

$C \rightarrow C_0 = \log_2(1 + Ps/Pn)$. Processing required by MIMO at higher speeds would be most manageable using OFDM modulation, because OFDM converts a high-speed data channel into a number of parallel lower-speed channels. MIMO-OFDM is a particularly powerful combination as MIMO does not attempt to mitigate multipath propagation and OFDM avoids the need for signal equalization. It can achieve very high spectral efficiency even when the transmitter does not possess channel state information (CSI). When the transmitter does possess CSI (which can be obtained through the use of training sequences), it is possible to approach the theoretical channel capacity. CSI may be used, for example, to allocate different size signal constellations to the individual subcarriers, making optimal use of the communications channel at any given moment of time. This allow us to use a QPSK channel modulation in addition.