Comparative Study of MIMO Systems with Linear Detection and Error-correction Code

Shih-an Yu, Student Member, IEEE

Abstract—This term paper presents comparative study of several MIMO systems. A group of simulated results of various MIMO systems from 1x1 to 4x4 with and without convolutional coding is also demonstrated and analyzed. Zero-forcing (ZF) and Minimum mean-squared error (MMSE) MIMO detection methods are utilized throughout the simulations.

Index Terms— SISO, MIMO, linear detection, zero-forcing, MMSE, error-correction, 16-QAM, QPSK.

I. INTRODUCTION

EVER since the first introduction of VBLAST system [1] in Bell laboratories in 1996 by Gerard J. Foschini, the multiple-inputs and multiple output (MIMO) systems have become a popular approach to boost the bandwidth efficiency of the most advanced wireless and mobile communication systems which are usually used in an environment where the wireless channels are fading ones. For an additive white Gaussian noise (AWGN) channel, the probability of bit error (or bit error rate, BER) for an MQAM system can be approximated as:

\[
P_b \approx 2\left(\frac{\sqrt{M-1}}{\sqrt{M \cdot \log_2 M}}\right) \cdot \left(\frac{3\gamma_s \cdot \log_2 M}{M-1}\right)
\]

where \(\gamma_s\) is the average energy per symbol in the M-QAM constellation. In AWGN the probability of bit error depends on received SNR or, equivalently, on \(\gamma_s\). However, in a fading environment, the received signal power varies randomly over distance or time due to shadowing and/or multipath fading. Thus, the received signal power is a random variable with certain distribution, and therefore the probability of bit error is also a random variable. Due to the characteristics of randomness, there are three different performance criteria to characterize the random variable \(P_b\):

- The outage probability, \(P_{\text{out}}\), defined as the probability that \(\gamma_s\) falls below a given value corresponding to the maximum allowable \(P_b\).
- Combined average error probability and outage, defined as the average probability of error that can be achieved some percentage of time or some percentage of spatial locations.

When the signal fading is on the order of a symbol time \((T_c-T_s)\), so that the signal fade level is constant over roughly over symbol time. Since many error-correction coding techniques can recover from a few bit errors, the average BER is a reasonably good figure of merit (FOM) for the channel quality under these conditions.

If the signal power is changing slowly \((T_c>>T_s)\), then a deep fade will affect many simultaneous symbols. Thus, fading may lead to large error bursts, which cannot be corrected for with coding of reasonable complexity. As a consequence, these error bursts can seriously degrade end-to-end performance. In this case acceptable performance cannot be guaranteed over all time unless the transmit power is increased drastically. Under these situations, an outage probability is used such that the channel is deemed unusable for some fraction of time or space.

Sometimes outage and average probability of error are combined when the channel is modeled as a combination of fast and slow fading, for example, log-normal shadowing with fast Rayleigh fading. However, even though the outage is a good measure for a system in slow fading scenario, we focus only on the average probability of error to simplify the comparison.

This term paper is organized as follows: In section II, several MIMO detection techniques are introduced and discussed. The simulation setup for bit error rate from various MIMO systems under different channel conditions is described in section III. In section IV, the simulated results with different setups are presented, compared, and discussed. And the conclusion of this work is given in section V.

II. MIMO DETECTION FOR FADING CHANNELS

A. Rayleigh Fading Channel

Fading channel causes drastic degradation in probability of error. Rayleigh fading is one of the worst-case fading scenarios. In Fig. 1, the comparison of average bit error rate of a 16-QAM system in both AWGN and Rayleigh fading channels. It can be
seen that when the wireless channel suffers from Rayleigh fading, the BER performance is drastically degraded compared to those with AWGN channels. For QAM-16 system at $10^{-3}$ of BER, the SNR requirement for fading channels is 10dB higher than a system operating in AWGN channel without error correction code, or 13dB higher with error correction code.

It can also be found that for a SISO system in slow fading channel, which cause burst errors. These burst errors cause a long series of symbols of signal to suffer from low SNR. Simple error correct code such as convolutional code cannot correct the wrong bits from low SNR scenarios. In addition, the BER of a system in fading environment decreases only linearly with increasing SNR. In other words, in order to reduce the BER by 50%, twice the original power is required if the system operates in this fading channel.

B. A MIMO Slow and Flat-fading Channel Model

In this term paper, the channel is assumed to be slow and flat fading. In flat-fading channels, all the frequency components of the signal experience the same magnitude of fading. In slow-fading channels, the amplitude and phase change imposed by the path gain from the fading environment can be considered constant over the period of use.

Consider a MIMO system which has $N_t$ transmitters and $N_r$ receivers, where $N_r$ is no less than $N_t$, the signals are transmitted and arrive at the array of receivers through a slow and flat-fading environment. The system can be treated as transmitting an $N_t \times 1$ vector signal $x$ through an $N_r \times N_t$ channel matrix $H$, with $N_r \times 1$ Gaussian noise vector $v$ added at the input of the receiver. And it results in a received $N_r \times 1$ vector $y$.

$$ y = Hx + v $$

(2)

The $(n_r, n_t)$-th element of $H$, $h_{n_r, n_t}$, is the complex channel response from the $n_t$-th transmitter to the $n_r$-th receiver. Vector $x$ and $v$ are zero-mean and their covariance matrices are listed as follows:

$$ R_x = E[xx^*] = \sigma_x^2 \cdot I $$

$$ R_v = E[vv^*] = \sigma_v^2 \cdot I $$

(3)

(4)

C. MIMO Detections

In a fading channel, the path gain of the channel for a given instance of transmission is a random vector, which introduces magnitude and phase errors. If there is only one pair of transmitter and receiver, the random path gain can result an SNR reduction lasting for a significant portion of the entire transmission. If multiple transmitters and receivers are used, the probability for all path gain vectors to be low can be greatly reduced if these path gain vectors are independent.

For the detection of MIMO signal, the most straightforward approach to recover the transmitted vector $x$ from the received $y$ is to use an $N_t \times N_r$ matrix $W$ to linearly combine the elements of $y$ to estimate $x$. Two popular approaches to achieve linear MIMO detection are described as follows:

1. Zero-forcing (ZF) Method: The zero-forcing algorithm attempts to null out the interference from the channel matrix by inverting the channel with the weight of matrix

$$ W_{ZF} = (H^* H)^{-1} H $$

(5)

where $H^*$ is the Hermitian of $H$.

2. Minimum Mean-squared Error (MMSE) Method: different from the zero-forcing method, the MMSE method attempts to minimize the mean-squared error between the estimated $x$ and $x$, and results in an optimal linear combination

$$ W_{MMSE} = R_{xy} R_y^{-1} = (H^* H + \frac{\sigma_y^2}{\sigma_x^2} I)^{-1} H^* $$

(6)

where $R_{xy}$ is the covariance matrix of $x$ and $y$.

D. Linear Adaptive MIMO Detection

Without the assumption of known channel matrix $H$, which usually requires channel probing and then computing $W$ before each transmission, adaptive algorithms are used to estimate $W$ directly through iteration through sending and receiving known training sequence at the beginning of each transmission.
E. Nonlinear MIMO Detection
Two major nonlinear MIMO detection method are described as follows:
1. VBLAST: VBLAST is the acronym of vertical Bell Labs layers space-time algorithm. It uses the detect-and-cancel strategy that is similar to decision-feedback equalizer. The performance of VBLAST is generally better than ZF and MMSE.

2. Maximum Likelihood (ML): ML detection has the best performance among all the MIMO detection algorithms presented here. It finds the $x$, which minimizes

$$
\| y - Hx \| = e_c^* * e_c = (y - Hx) * (y - Hx)
$$

In other words, the most likely transmitted signal that cause the smallest squared error from the received signal. However, the computational complexity increases exponentially with increasing signal constellation, and therefore the cost can be prohibitively high for practical use.

III. SIMULATION SETUP AND IMPLEMENTATION

The simulations of MIMO systems are conducted assuming slow and flat fading channel. In practical systems, it means that after the system estimates the channel before the transmission, the channel will not change during the period of use. In the simulation configuration, this property assumes that the channel matrix is constant during each realization when the sequence of bits is transmitted. For the Rayleigh-fading channel, an array that contains 1000 channel matrices is generated for 1000 realizations. The weighting matrix $W_{ZF}$ and $W_{OPT}$ are computed based on those channel matrices and signal SNR according to equation (5), (6), and (8). The number of realizations can have effect of the quality of average BER simulation for a Rayleigh fading channel because of the statistical nature of all the elements in the channel matrix. In Fig. 2, it can be shown that without sufficient number of realizations, the BER curve can deviate from the line that decreases linearly with increasing SNR.

In order to achieve fair comparison, the channel is set to have unity average signal energy at the receiver array and the SNR is determined by

$$
SNR = \frac{\sigma^2_x}{\sigma^2_e}
$$

In each realization, 1000 symbols are transmitted from the $N_t \times 1$ transmitter array such that the total sample population is large enough to make sure at least 100 errors can be detected for each BER point. In addition, the bits stream for transmission is also encoded by convolutional code with code rate of 1/2 and decoded by Viterbi decoder with hard decision and proper traceback length. Performance of MIMO systems with and without coding is simulated and compared in various scenarios.

IV. SIMULATION AND DISCUSSIONS

Different combinations of parameter setups for MIMO systems in fading channels have been simulated. The results are presented and discussed as follows:

A. Performance of 2x2 MIMO Systems in Rayleigh-fading Channels with Different Linear Detection

In the first experiment, zero-forcing detection is used in 1x1, 2x2, and 4x4 MIMO 16-QAM systems. Compared to the reference performance of a QPSK system in a Rayleigh-fading channel, a 1x1 16-QAM system needs about 7dB higher SNR to achieve a BER of $10^{-3}$. Furthermore, 2x2 and 4x4 systems need about 3dB and 6dB higher SNR than 1x1 systems to achieve a BER of $10^{-3}$.

If MMSE method detection, which has been described in section II is used to detect MIMO signals in a fading channel, the SNR corresponding to certain BER performance can be reduced. In Fig. 4, the simulated BER performance of a 4x4 16-QAM systems with MMSE detection shows a slight improvement while no observable improvement can be found for the 1x1 and 2x2 system.
B. Performance of Error Correction Code in 2x2 MIMO Systems

The next experiment is to simulate the BER performance while error-correction code is used. The code rate used here is 1/2, which means that if N-bit long message is encoded, the resulting coded message is 2N-bit long. In Fig. 5, the error correction code can improve the BER performance of 2x2 and 4x4 systems slightly. For 1x1 systems, however, no obvious improvement can be found. The convolutional code used here is not designed to combat burst errors, and therefore if the path gain in certain realization turns out to be low, the detection quality for the entire realization will be jeopardized. Once the number of continuous errors in the message exceeds the correction capability, the functionality of error-correction breaks down.

In 2x2 and 4x4 systems, the probability of having all the path gain vectors in the channel matrix to be low simultaneously is greatly reduced since these vectors are assumed independent. Even though the channel is still slow-fading, the probability for a burst error to happen is now reduced, and the error-correction code can still function effectively.

C. Performance of Asymmetric MIMO Systems

In the previous simulations, only the results of MIMO systems with same numbers of transmitters and receivers are presented. In some cases, the number of receivers can be more than the number of transmitters. This configuration can greatly boost the BER performance of the communication systems operating in fading channels.

QPSK(4-QAM) and 16-QAM systems are simulated with unequal numbers of receivers and transmitters, and the simulated results are show in Fig. 6 and Fig. 7, respectively. In these simulations, the ratio between the numbers of the receivers and transmitters are kept to be two. In Fig. 6, a significant improvement of BER performance can be observed. The required SNR to achieve a BER of $10^{-3}$ is reduced at least 10dB when compared to the reference QPSK system.

More antennas create more diversity gain even though the ratio is kept the same. When compared to the results presented in the previous sub-sections, the effects of the increased diversity gain can be more pronounced while the number of receivers is more than that of transmitters. The BER from the best to worst is 4x8, 2x4, and 1x2 for the QPSK MIMO system presented in Fig. 6 and 2x4, 1x2 for the 16-QAM MIMO systems in Fig. 7. Except for the effects of the increased diversity gain, the performance of the error-correction codes is also improved, while the performance improvement by the coding is not obvious in the previous sub-sections where the simulations are set to have the same numbers of receivers and transmitters in the MIMO systems.

V. CONCLUSION

In this work, a series of simulations are conducted to compare the BER performance of different MIMO systems in fading channels. In order to have correct simulations, the number of realizations and transmission bits are chosen to be large enough. The configurations are summarized as follows:

1. QPSK and 16-QAM.
2. Zero-forcing (ZF) and minimum mean-squared error (MMSE) detections.
3. Numbers of transmitters and receivers.

From the simulated results, it can be concluded that larger constellation and small numbers of transmitters and receivers are less efficient compared to simple constellation but with larger number of antennas. In addition, the performance of the error-correction codes is better when the constellation is small or when the number of receivers is more than that of transmitters.
REFERENCE


APPENDIX

The matlab codes for simulation:

```matlab
function[y]=qam16_map(x);
% This function adopts gray code 16-QAM
%% 0111(-3, +3) 0101(-1, +3) 1101(+1, +3) 1111(+3, +3)
%% 0110(-3, +1) 0100(-1, +1) 1100(+1, +1) 1110(+3, +1)
%% 0010(-3, -1) 0000(-1, -1) 1000(+1, -1) 1010(+3, -1)
%% 0011(-3, -3) 0001(-1, -3) 1001(+1, -3) 1011(+3, -3)

x_sign=sign(x-0.5);
i=x_sign(:,1)+2*x_sign(:,1).*x(:,3);
q=x_sign(:,2)+2*x_sign(:,2).*x(:,4);
y=[i,q];
```

```matlab
function[y]=qam16_inv(x);
a1=(x(:,1)>0);
a2=(x(:,2)>0);
a3=(x(:,1)>2)+(x(:,1)<-2);
a4=(x(:,2)>2)+(x(:,2)<-2);
y=[a1,a2, a3, a4];
```

```matlab
function[y]=qam4_map(x);
% This function adopts gray code 4-QAM
%% 01(-1, +1) 11(+1, +1)
%% 00(-1, -1) 10(+1, -1)

x_sign=sign(x-0.5);
i=x_sign(:,1);
q=x_sign(:,2);
y=[i,q];
```

```matlab
function[y]=qam4_inv(x);
a1=(x(:,1)>0);
a2=(x(:,2)>0);
y=[a1,a2];
```

%Main code starts from here
```matlab
clear;

%mmse/zf 1/0 switch
mmse=1;

%convolutional coding switch 1/0 = on/off
coding=1;

%plot SNR or Eb/N0
plot_SNR=1;

%fading 1/0=on/off
fading=1;

%mod parameter
bit requires that number_rx >= number_tx
number.tx=2;
number_rx=2;

%constellation
mqam=16;

%data volume
number_symbol=10000;
bit_per_symbol=log2(mqam);
number_realization=1000;
number_total_bits=number_symbol*bit_per_symbol*number_realization;

%seed control
seed=269451; % best seed so far 123456
seed_master=randint(5,1,[0,100000000],seed);
seed_ch1=randint(2,1,[1,1000000000],seed_master(1));
seed_bl1=randint(number_realization,1,[1,1000000000],seed_master(1));
seed_ch2=randint(2,1,[1,1000000000],seed_master(2));
seed_bl2=randint(number_realization,1,[1,30000000000],seed_master(2));
seed_ch3=randint(2,1,[1,30000000000],seed_master(3));
seed_bl3=randint(number_realization,1,[1,30000000000],seed_master(3));

%rayleigh channel generation
if(fading==0)
 rayl_ch=1;
 else
 rayl_coeff = wgn(number_realization,2*number_rx*number_tx,-3,1,seed_ch1(1),'real');
 rayl_ich = wgn(number_realization,number_rx*number_tx,-3,1,seed_ch1(1),'real');
 rayl_qch = wgn(number_realization,number_rx*number_tx,-3,1,seed_ch1(1),'real');
 rayl_ch=rayl_ich+j*rayl_qch;
 end
```

%trellis description for convolutional code
trel=poly2trellis(3,[6 7]); % 1/2
```

%generate data bit in each realization
number_bit=number_symbol*bit_per_symbol;
```
```
%coding
if(coding==1)
 coded_bits = convenc(bits,trel); % Encode.
 else
 coded_bits = bits;
 end
```
```
%slice the data bit sequence into pieces with 4bit/symbol
symbol_code=reshape(coded_bits, bit_per_symbol*number_tx, 
)';
```
```
end
```
```matlab
fig.7 16-QAM systems in different MIMO configurations are simulated. The 2x4 MIMO achieves the best performance among all the configurations.

References


```
else
    for k=1:number_tx
        temp=qam_map(symbol_code(:,(k-1)*bit_per_symbol+1:k*bit_per_symbol));
        signal_c(:,k)=(temp(:,1)+j*temp(:,2))*normalization_factor;
    end
end
% construct channel elements
y_c = signal_c*h;
% measure the real power of the signal
s_power_meas=sum(y_c.*conj(y_c))/number_symbol;
% generate noise at receiver antenna
noise=wgn(length(coded_bits)/bit_per_symbol/number_tx,2*number_rx, N_power_in_dB-3, 1, seed_n1(SNR_in_dB)+seed_n2(realization));
noise_i=noise(:,1:number_rx);
n擁有 noise_q=noise(:,number_rx+1:end);
n_c=noise_i+j*noise_q;
% measure the real noise power from I/Q
n_power_meas=sum(n_c.*conj(n_c))/number_symbol;
% received noise to the Transmitted signal
r_c=y_c+n_c;
% inverse the channel
r_c_inv=r_c*h_inv;
% received signal symbols
if(mqam==16)
    for k=1:number_tx
        temp_i=real(r_c_inv)/normalization_factor;
        temp_q=imag(r_c_inv)/normalization_factor;
        symbol_code_r(:,(k-1)*bit_per_symbol+1:k*bit_per_symbol)=qam16_inv([temp_i(:,k), temp_q(:,k)]);
    end
else
    for k=1:number_tx
        temp_i=real(r_c_inv)/normalization_factor;
        temp_q=imag(r_c_inv)/normalization_factor;
        symbol_code_r(:,(k-1)*bit_per_symbol+1:k*bit_per_symbol)=qam4_inv([temp_i(:,k), temp_q(:,k)]);
    end
end
% shape back to number_bitsx1 array
coded_bits_r=reshape(symbol_code_r',1, []);
% decode
if(coding==1)
    bits_r=vitdec(coded_bits_r,trel,tblen,'cont','hard'); % Hard decision
    diff_bits=abs(bits(1:end-tblen)-bits_r(tblen+1:end));
    error_count(realization, SNR_in_dB)=sum(diff_bits);
else
    bits_r=coded_bits_r;
    diff_bits=abs(bits-bits_r);
    error_count(realization, SNR_in_dB)=sum(diff_bits);
end
% detect the difference
end
end
error_summary=sum(error_count);
error_rate=error_summary/number_total_bits;
figure(1);
semilogy(SNR_min:SNR_step:SNR_max)-(plot_SNR==0)*10*log10(bit_per_symbol), error_rate(SNR_min:SNR_step:SNR_max), '-db');
grid on;
xlabel('SNR [dB]');
ylabel('Bit Error Rate (BER)');
hold on;