Department of Electrical and Computer Engineering
Wireless Communication Systems
Instructor: Dr. Dario Pompili

Analog Joint Source Channel Coding for Wireless Sensor Networks

Angela Ciummo, Lidan Wang, Lulu Jiang

November 22, 2015

1 Introduction

With the world in the midst of the so-called "Technology Age," one of the biggest challenges faced today is finding inexpensive, efficient implementations of various technologies. Originally motivated by military applications, wireless sensor networks (WSNs) have become a big topic of interest in telecommunications and their applications have expanded to industrial and consumer use. Today's WSNs are employed for many things from environmental and earth sensing to healthcare sensing, however one common theme that these wireless sensor networks share is that they are implemented via digital, expensive, power-hungry systems. The purpose of this research is to propose Analog Joint Source Channel Coding (AJSCC) as an inexpensive, efficient method to implement wireless sensor networks. [4]

Joint Source Channel Coding(JSCC) is the encoding of a redundant information source for transmission over a noisy channel, and the corresponding decoding, using a single code. Analog Joint Source Channel Coding is the analog-circuit based implementation of JSCC. This project proposes using a multi-tier architecture for WSNs. Tier 1 is comprised of high-density analog sensor nodes, tier 2 is comprised of low-density digital clusterheads, and tier 3 contains a fusion center. This 3-tier architecture splits the sensing and computing functionality between high-density, inexpensive, low-power analog sensors, expensive, power-hungry digital clusterheads, and powerful fusion centers. The analog sensors have shannon mapping capabilities, as shannon mapping is resilient to channel impairments. [4]

Shannon mapping was developed by Claude Shannon in his paper "Communication in the Presence of Noise," as a method of encoding two signals into one for efficient transmission. Shannon mapping achieves 2:1 compression by mapping two signals onto either a rectangular map or a spiral map. Figure 1 is an illustration of Shannon mapping on a rectangular map. Note that the x and y axes are x_1 and x_2 respectively, with x_1 and x_2 being signals of length 1. The point (x_1, x_2) is mapped to the closest point on the parallel lines, with the distance from the origin, or total length of the bold line on figure 1, being the single value that the two signals are mapped to. Δ represents the distance between parallel lines and is taken into account when calculating the distance between the origin and (x_1, x_2). A fixed value called D_{max} represents the amplitude constraint, and the number of levels, or number of parallel lines, is also a fixed value. The demapped point ($\hat{x}_1$, $\hat{x}_2$) is found using the total length, however the demapped point will be slightly different from the original signal as the compressed signal, or Vd_1, encounters noise at the receiver. Shannon mapping is a good method of encoding because it is robust to channel distortion. [3]

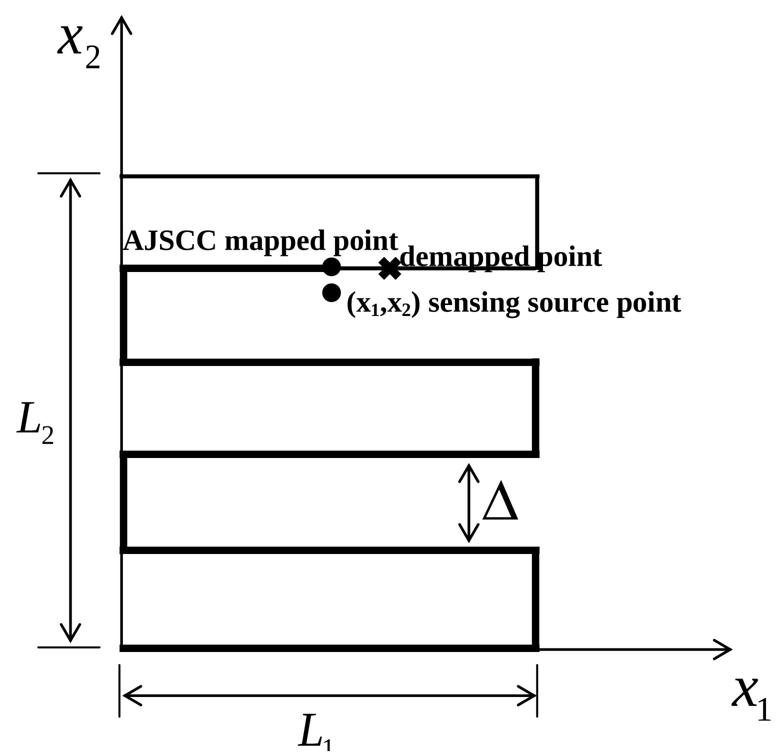
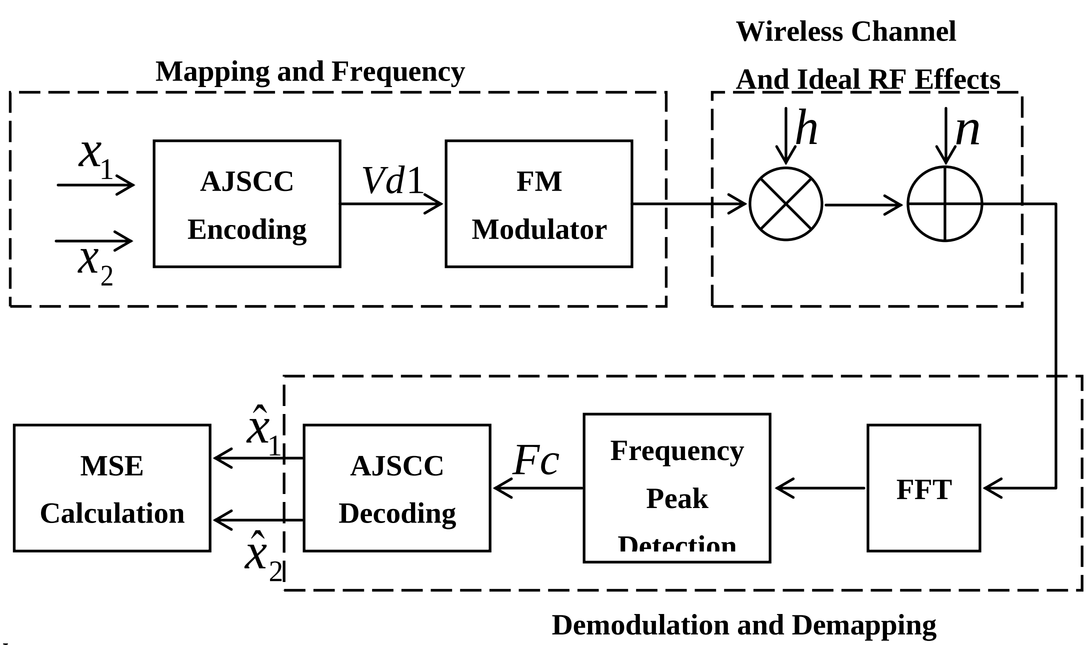


Fig 1. Shannon's Rectangular Mapping. Sensed point(x_1, x_2) is mapped to a point closest on the rectangular curve(x_1', x_2'), thus two signals will be compressed to one signal which is the length of the bold lines. The value of length will be transmitted instead of two sensed signals.

This project focuses on taking the AJSCC encoded signals, x_1 and x_2, and performing FM modulation over a wireless channel. The details of the project include: performing FM modulation of the of the AJSCC encoded signal onto a cosine wave with constant power, generating a Rayleigh-fading channel to transmit the FM modulated signal over, performing FFT-based FM demodulation, and plotting the MSE vs. L, and MSE vs. SNR curves to evaluate the performance. The following report will discuss the simulation code and results, however for more detail please refer to the attached code.

2 Implementation

2.1 General Project



Overview

Fig 2. The block diagram detailing the AJSCC encoding/decoding of two signals, x_1 and x_2

Figure 2 illustrates the AJSCC system via a block diagram. This implementation aims at providing promising results that can prove that Analog Joint Source Channel Coding is a solution to costly, power-hungry digital sensor networks currently used today. Firstly, there are two signals x_1 and x_2 sent through the AJSCC encoding block. AJSCC encoding uses Shannon mapping capabilities, so the output of the AJSCC encoding block is Vd_1. Then, the output signal is frequency modulated by the FM modulator. Subsequently, the FM modulated signal is sent through the Rayleigh fading channel and additive white Gaussian noise (AWGN) is added to the received signal. At the receiver, the Fast Fourier Transform (FFT) is performed on the received signal to analyze the signal in the frequency domain. From the frequency domain view of the signal we can detect the peaks of the FFT, which should occur at the carrier frequency of the received signal. After determining the base-band frequency the reverse mapped signals $\hat{x}_1$ and $\hat{x}_2$ can be obtained. Finally, the Mean Square Error (MSE) can be computed and the performance of the AJSCC system can be evaluated.

2.2 AJSCC Encoding

```
Dmax=5;
N=20;
level=100;
beta=1000;
SNR=-20;
for Levels=1:level;
w=[];
v=[];
u=[];


for i=1:N

x1=rand(1);           %generate signal x1 between [0,1]
x2=rand(1);           %generate signal x2 between [0,1]
L1= (Dmax-1)/Levels;      %range of x1
L2=1;                 %range of x2
delta=L2/(Levels-1);
x10=x1*L1;      %change the range of x1 from generated [0,1] to [0,L1]
k=round(x2/delta);
if (rem(k,2)== 0)          % Vd1's value when k is even
    Vd1= (k*L1) + x10 + (k*delta);
else
    Vd1=(k*L1) + L1 - x10 + (k*delta);    % Vd1's value when k is odd
end
```

Fig 3. MATLAB Implementation of AJSCC encoding (and Shannon Mapping)

Figure 3 illustrates the MATLAB implementation of AJSCC encoding. The pure-analog sensor nodes that AJSCC employs have Shannon mapping capabilities, so a large part of implementing AJSCC encoding is Shannon mapping. Note that the values of D_{max} and the number of Levels are set values. The signals x_1 and x_2 are generated using MATLAB's built-in function rand, with each signal being of length 1. The compressed signal, Vd_1, was found using the following equation:

$$Vd1 = \begin{cases} kL_1 + x_1 & k \text{ is even} \\ kL_1 + L_1 - x_1 & k \text{ is odd} \end{cases}$$ **(Equation 1)**

where k=round(x_2/Δ). [4]

2.3 FM Modulation

In order to perform transmission, the signal to be transmitted has to be modulated. Modulation is the process of varying one or more properties of a periodic waveform, called a carrier signal, with a modulating signal that has information to be transmitted. Common modulation methods include amplitude modulation, frequency modulation, phase modulation, etc. This project uses frequency modulation (FM modulation) for transmission.

FM modulation is the encoding of information in a carrier wave by varying the instantaneous frequency of the wave. FM modulation can be given by the equation:

$$FM \quad Modulated \quad Signal = \sqrt{P_0} * \cos(2\pi * \beta * Vd_1 * t)$$ **(Equation 2)**

where P_0 is the power of the signal, β is the scaling factor, Vd_1 is the AJSCC encoded signal, and t represents the time sample. FM modulation is widely used and has many benefits. Some of the benefits of FM modulation include resilience to noise, resilience to signal strength variations, and a good anti-noise performance that can operate at a low Signal-to-Noise Ratios (SNRs).

Since D_{max} was set to 5 the range of Vd_1, the AJSCC encoded signal, was [0, 5]. A linear conversion is performed in FM modulation, which is represented by $Vd_1*\beta$. Thus, the carrier frequency ranges from 0 to 5 kHz. When a signal is transmitted through a static channel only the phase and amplitude change, and the frequency does not. Therefore, it is easy to get a accurate demodulation result at the receiver. [6]

```
Fs = 30*beta*Vd1;          % Sampling frequency
T = 1/Fs;             % Sampling period
L = 100000;             % Length of signal
t = (0:L-1)*T;        % Time vector
FM=cos(2*pi*beta*Vd1*t);   % FM modulation, beta*Vd1 is the carrier frequency
```

Fig 4. MATLAB implementation of FM modulation

Figure 4 shows the MATLAB implementation of the FM modulation. It is implemented identically to the equation given in equation 1. For the purposes of this project β was set to 1000.

2.4 Rayleigh Fading Channel and Additive White Gaussian Noise(AWGN) channel

The Rayleigh fading channel is a statistical model of the radio signal propagation environment, such as that used by wireless devices. Rayleigh fading channels are most applicable to instances where there are different signal paths, none of which are dominant. This makes the Rayleigh fading channel a perfect model for the channel because it most closely resembles a cellular communication channel. One of the features of this model is that when a signal is sent through this channel the amplitude is random i.e. fading, and the envelope of the signal obey Rayleigh distribution. [1]

Additive White Gaussian Noise is a basic and generally accepted model for thermal noise in communication channels. The set of assumptions are:

1) the noise is additive, i.e. the received signal equals the transmit signal plus some noise, where the noise is statistically independent of the signal.

2) the noise is white, i.e. the power spectral density is flat, so the autocorrelation of the noise in time domain is zero for any non-zero time offset.

3) the noise samples have a Gaussian distribution.

```
% rayleigh channel fading
 chan=rayleighchan(1/100,0);
 Vd0=filter(chan,FM);

% add noise
sigPower = 10*log10(sum(abs(Vd0(:)).^2)/length(Vd0(:)));
Vd2 = awgn(Vd0,SNR,sigPower); % Add Gaussian noise to Vd1 with SNR=sigPower/noisePower SNRdB=sigPowerdB-noisePowerdB
```

Fig 5. MATLAB implementation of the Rayleigh Fading Channel and Additive White Gaussian Noise

Figure 5 shows the MATLAB implementation for the Rayleigh fading channel and the addition of white Gaussian noise. MATLAB’s built-in function “rayleighchan” was used to generate the channel. Rayleighchan accepts sampling frequency and maximum doppler shift as input, and outputs the channel. For the purposes of this project the sampling frequency was set to 0.01, and the maximum doppler shift was set to zero, so as to resemble static channel. By filtering the FM modulated signal by the Rayleigh fading channel, Vd_0, or the received signal is found. At the receiver, white Gaussian noise is added to the signal and to implement this the function "awgn" was used to obtain Vd_2, or the received AJSCC encoded, FM modulated signal with noise.

2.5 FM Demodulation

When a modulated signal is transmitted the job of the receiver is to demodulate the signal to get the original signal back. Demodulation is the act of extracting the original signal from the carrier wave. Since FM modulation was used for transmission, FM demodulation must be used to extract the original signal. One common method of FM modulation is to pass the signal through a low pass filter, and then pass the signal through an envelope detector. An envelope detector is an electronic circuit that takes the signal as input and outputs the envelope of the signal. It is in this way that we can obtain the FM demodulated signal. During the MATLAB simulation, in every loop we get a constant value of Vd1, thus the FM modulated signal will only have one constant frequency during the transmission. [7]. Unlike normal FM demodulation, we only need to detect this frequency at the receiver for every iteration and a vector of different demodulated signals will achieved. An FFT-based FM demodulation was performed for this project. The FFT, or Fast Fourier Transform, is an algorithm that computes the discrete Fourier Transform and converts the time domain signal into a frequency domain signal. When the Fourier Transform of the signal is found, the function has two peaks that occurs at the carrier frequency. Using the carrier frequency found using a peak detection algorithm, the demodulated signal can be AJSCC decoded.

```
% fft
y=fft(Vd2);
% figure;plot(abs(y));

P2=abs(y/L);
P1= P2(1:L/2 +1);
P1(2:end-1)= 2*P1(2:end-1);
f=Fs*(0:(L/2))/L;
% figure;
% plot(f,P1);

% detect peak frequency
[pks,locs] = max(P1);
Fc=locs*Fs/L;
```

Fig 6. MATLAB implementation of FFT-based FM demodulation and peak detection

Figure 6 shows the MATLAB implementation of FFT-based FM demodulation and the peak detection algorithm. MATLAB's built-in function "fft" was used to calculate the FFT of Vd_2. In order to use the peak detection algorithm, the fft of Vd_2 was normalized by the length of the signal. P2 is the two-sided spectrum of fft signal and P1 is the single-sided spectrum based on P2 and the even-valued signal length L. By defining the frequency domain f we got the single-sided amplitude spectrum of the fft signal. [8] As we found during the simulation, longer signals produce better frequency approximations. Subsequently, using MATLAB's built-in function "max" the peak of the FFT was found and the index of the max value corresponds to the carrier frequency.

2.6 AJSCC Decoding

```
Yy=Fc/beta;
k1 = floor(Yy/(L1+delta));          %demapping
if (rem(k1,2)== 0)
    x11= Yy-k1*(L1+delta); %x1 hat (demapped x1)
else
    x11=(k1+1)*(L1+delta)-Yy-delta;
end
x110=x11/L1;        % change the range of x11 from [0,L1] back to [0,1]

x21 = k1*delta;  % x2 hat (demapped x2)

w(i)=((x1-x110).^2) + ((x2-x21).^2); %|x1-x11|^2 - |x2-x21|^2
v(i)=((x1-x110).^2) ; %|x1-x11|^2
u(i)=((x2-x21).^2); %|x2-x21|^2
end


p0=1;                   % power of the FM modulation
Fs = 20000;             % Sampling frequency
T = 1/Fs;               % Sampling period
L = 1000;               % Length of signal
t = (0:L-1)*T;          % Time vector
MSE(Levels)=mean(w); % MSE= E[|x1-x11|^2 - |x2-x21|^2]
MSEx1(Levels)=mean(v); % MSE= E[|x1-x11|^2]
MSEx2(Levels)=mean(u); % MSE= E[|x2-x21|^2]

end

levels=1:2:level;
semilogy(levels,MSE(levels),'-+'); hold on;
semilogy(levels,MSEx1(levels),'-o'); hold on;
semilogy(levels,MSEx2(levels),'-square');
title('MSE vs. L' );
xlabel('Parameter L, maximum number of parallel lines');
ylabel('Mean Square Error (MSE)');
legend('Sum MSE of [x1,x2]','MSE of x1','MSE of x2');
```

Fig 7. MATLAB implementation of AJSCC decoding and MSE Calculation

Figure 7 shows the MATLAB implementation of AJSCC decoding. After the received signal is demodulated and the carrier frequency, F_c, is found the carrier frequency is then normalized by the scaling factor β. In order to decode the compressed signal the following relation is used to find $\hat{x}_1$:

$$\begin{cases} \hat{x}_1 = (k_1 + 1) * L_1 - y & k_1 \, is\, odd \\ \hat{x}_1 = y - k_1 * L_1 & k_1 \, is\, even \end{cases} \quad \textbf{(Equation 3)}$$

where k_1=floor(y/L1+Δ), and $\hat{x}2 = k_1 * \Delta$. Next, we calculated the mean squared error (MSE). The MSE measures the average of the squares of the errors, and it can be calculated via equation 4. Where the square of the difference between x_1 and $\hat{x}_1$ corresponds to the MSE of x_1, and the square of the difference between x_2 and $\hat{x}_2$ correspond to the MSE of x_2. [4]

$$MSE = E[| x_1 - \hat{x}_1 |^2 + | x_2 - \hat{x}_2 |^2] \quad \textbf{(Equation 4)}$$

3 Results

3.1 Under Additive White Gaussian Noise Channel

3.1.1 MSE vs. L with different SNRs

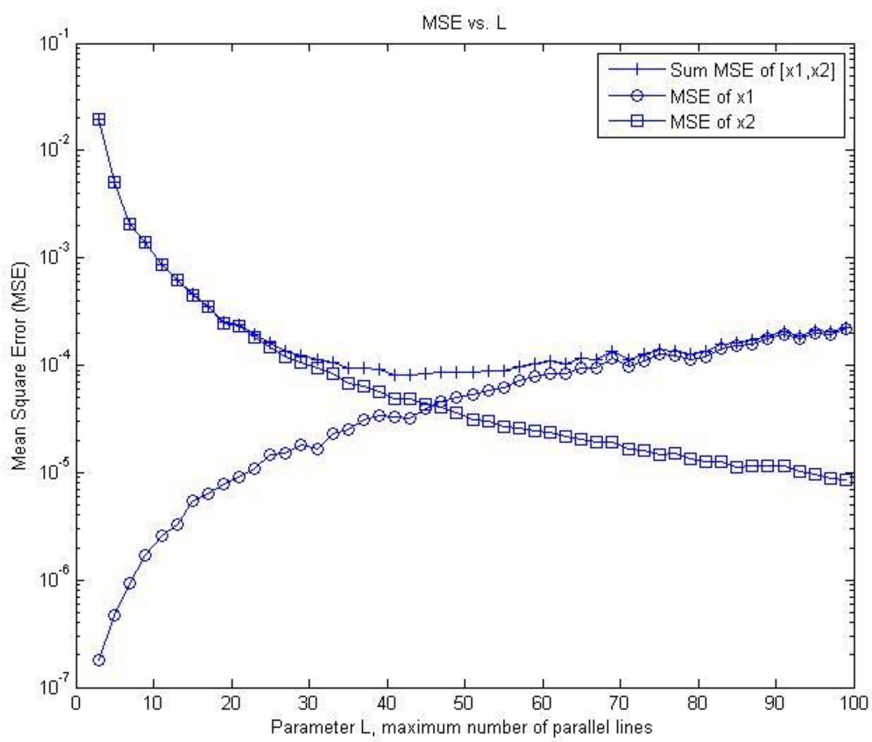


Fig 8. MSE vs. L for SNR = -20 dB

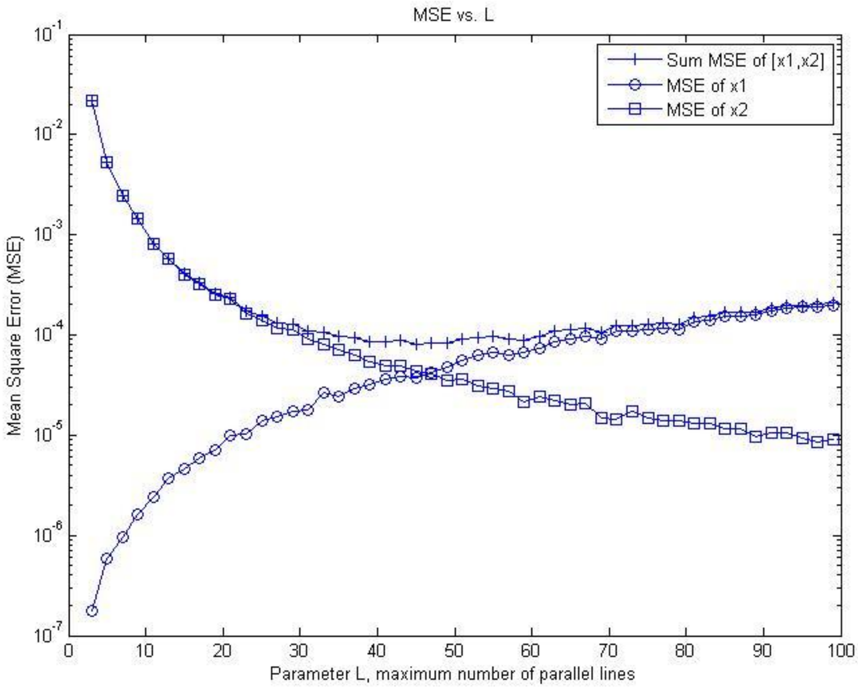


Fig 9. MSE vs. L for SNR = -10 dB

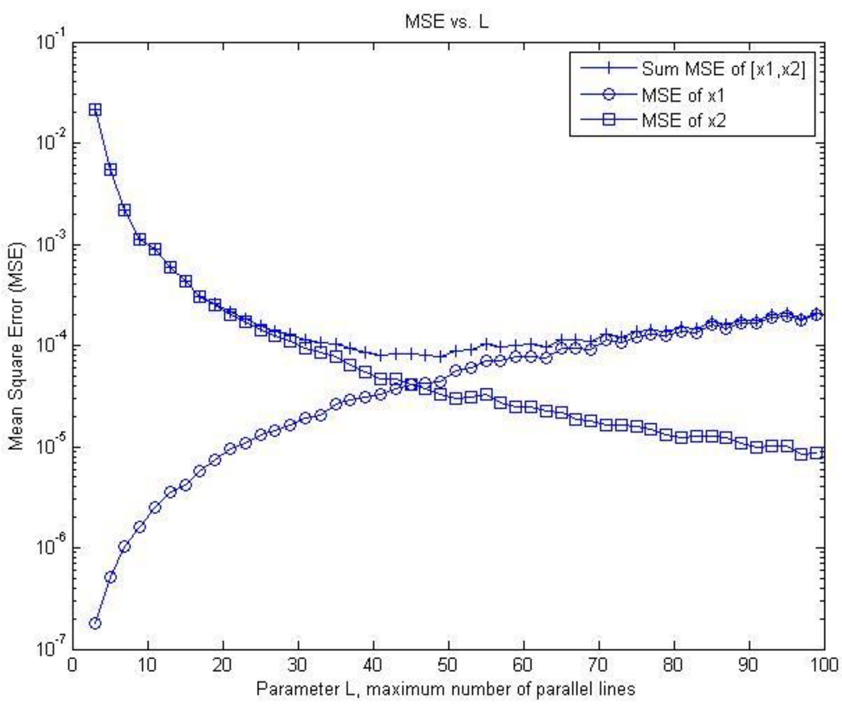


Fig 10. MSE vs. L for SNR = 0 dB

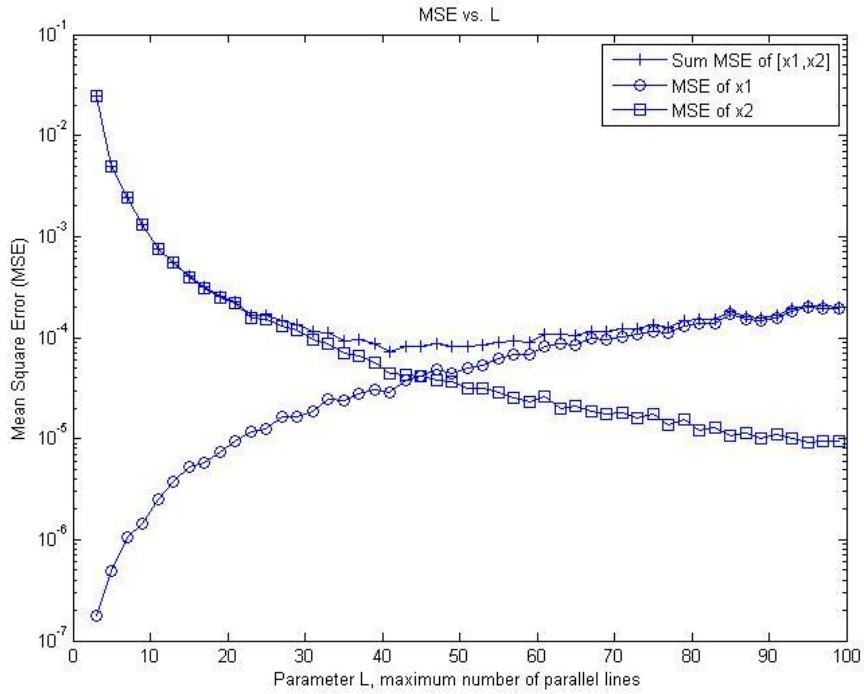


Fig 11. MSE vs. L for SNR = 10 dB

3.1.2 MSE vs. SNR with different levels

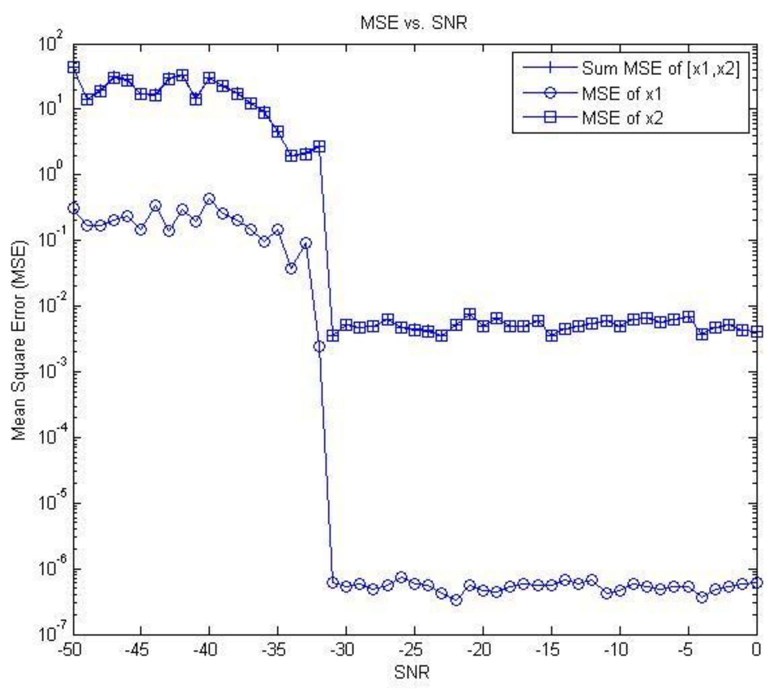


Fig 12. MSE vs. SNR for number of lines = 5

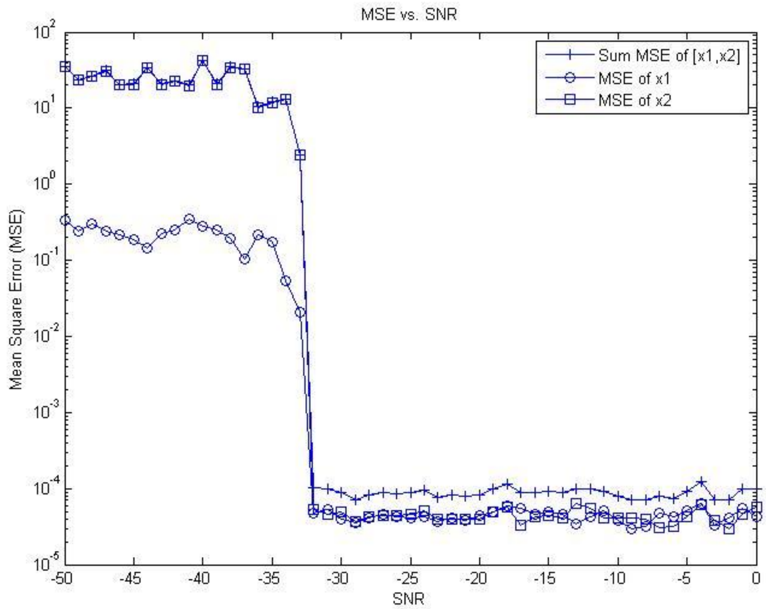


Fig 13. MSE vs. SNR for number of lines = 45

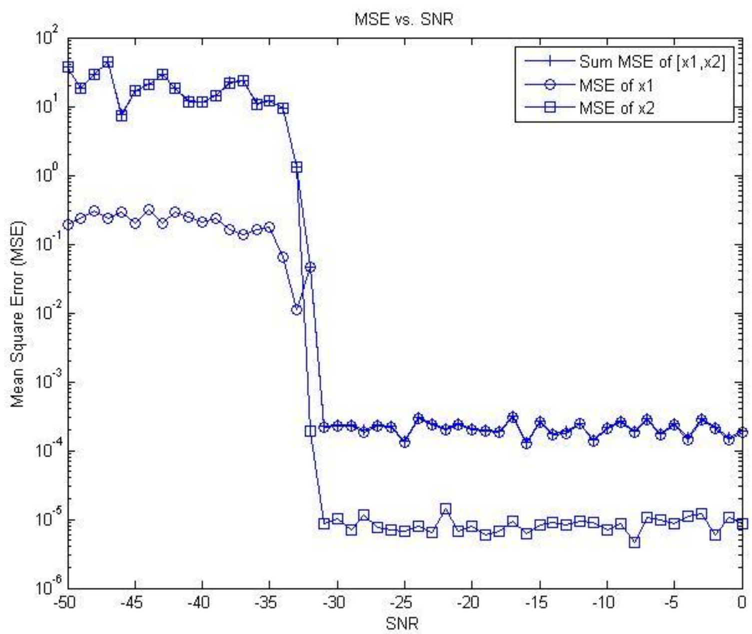


Fig 14. MSE vs. SNR for number of lines = 100

3.2 Under Additive White Gaussian Noise Channel and Rayleigh fading channel

3.2.1 MSE vs. L with different SNRs

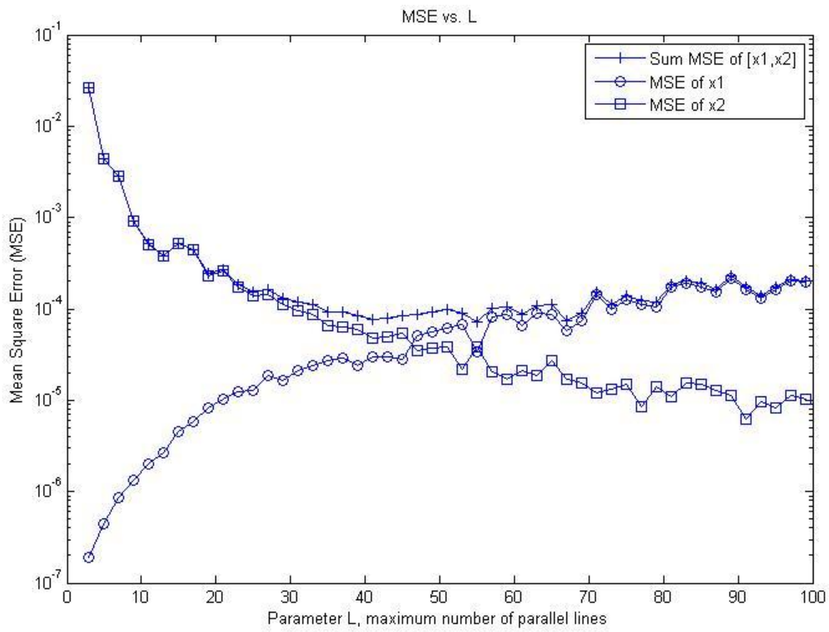


Fig 15. MSE vs. L for SNR = -20 dB

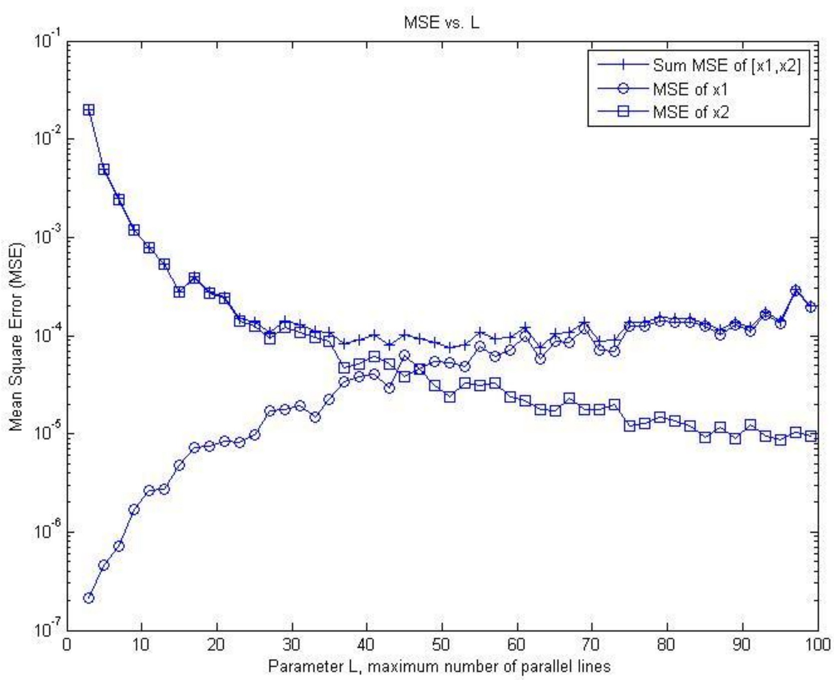


Fig 16. MSE vs. L for SNR = -10 dB

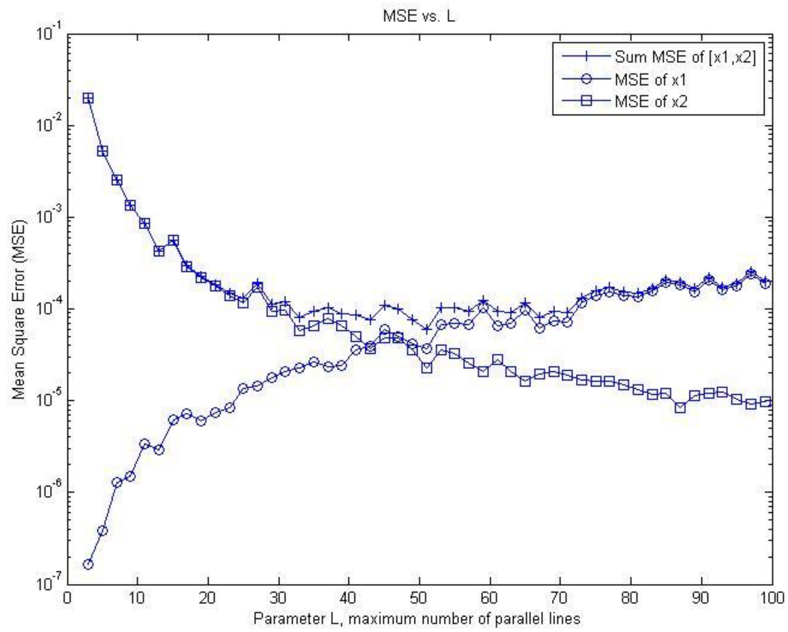


Fig 17. MSE vs. L for SNR = 0 dB

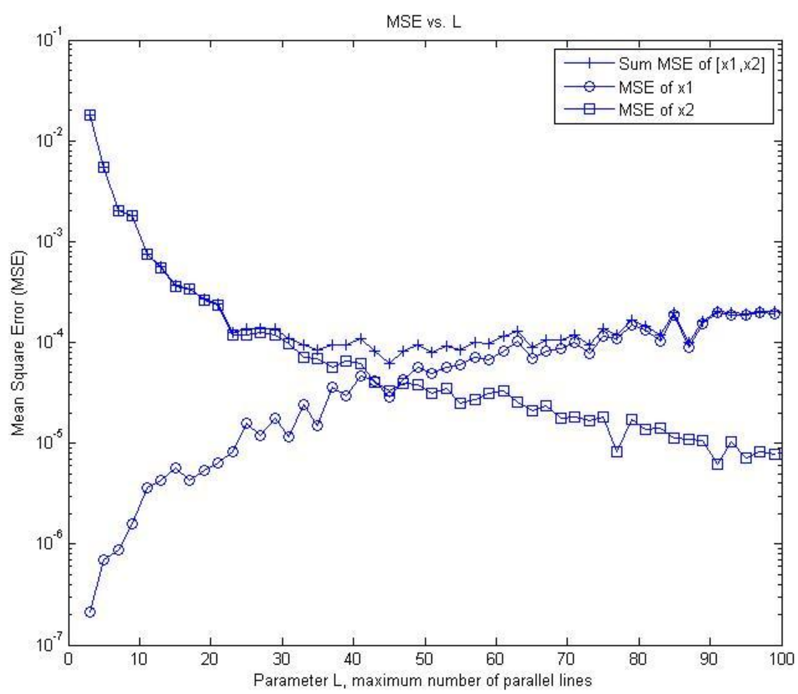


Fig 18. MSE vs. L for SNR = 10 dB

3.2.2 MSE vs. SNR with different levels

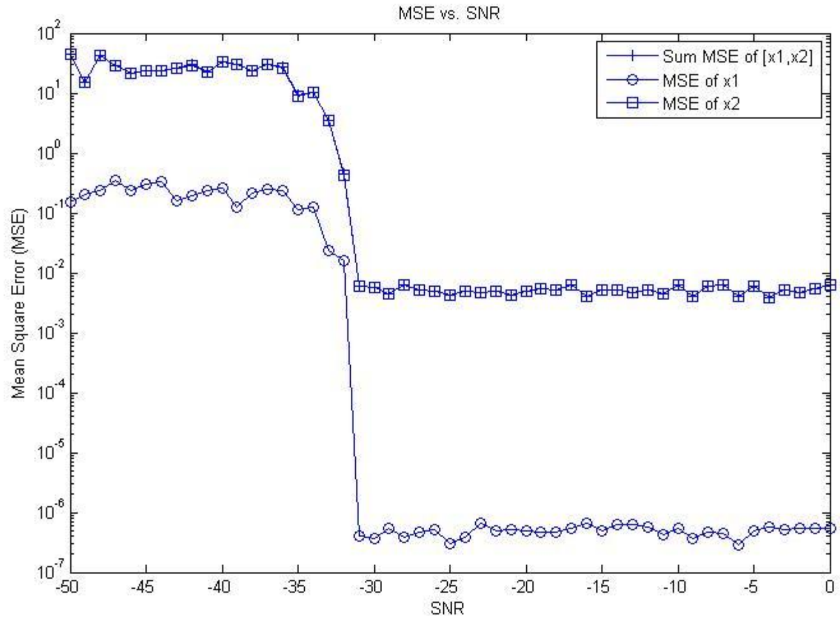


Fig 19. MSE vs. SNR for number of lines = 5

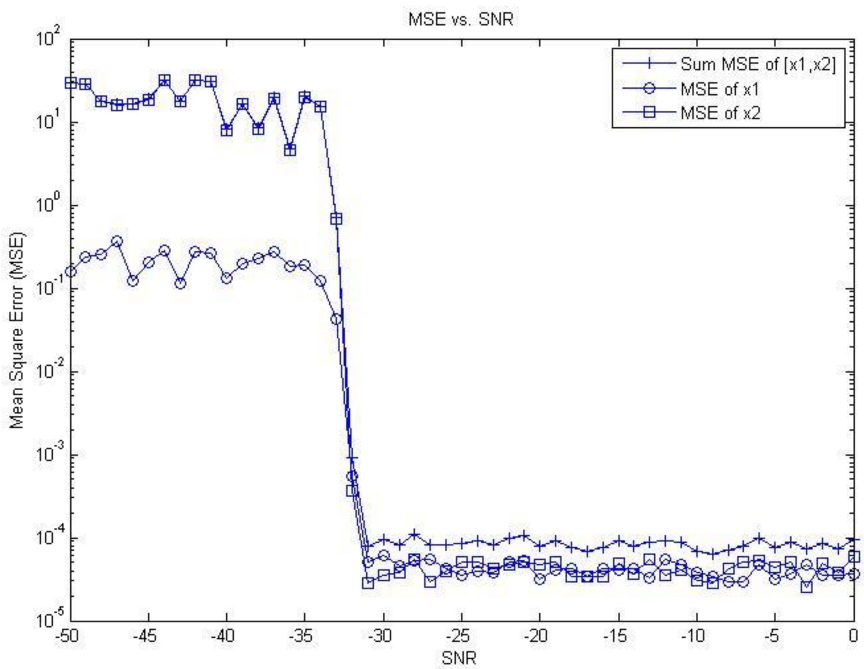


Fig 20. MSE vs. SNR for number of lines = 45

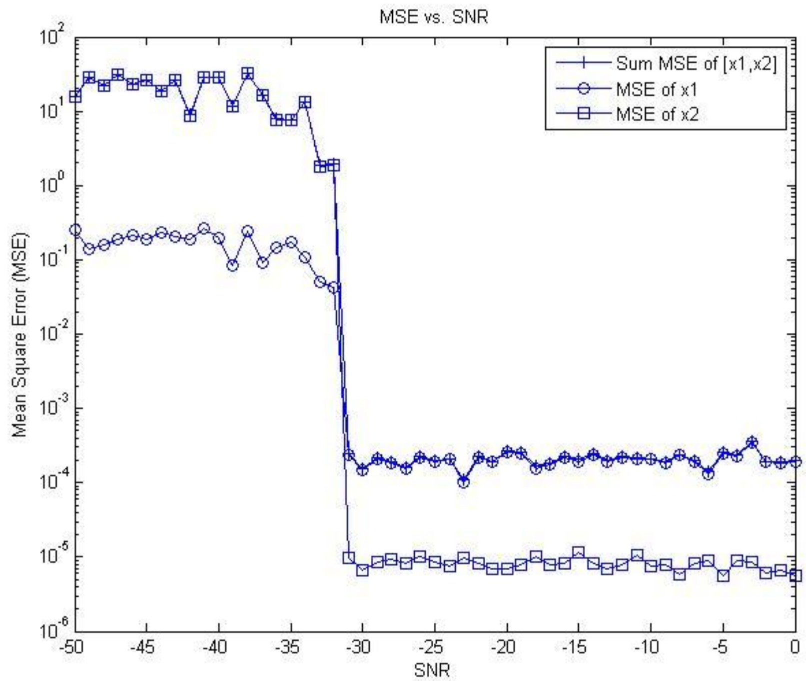


Fig 21. MSE vs. SNR for number of lines = 100

3.3 Simulation error caused by Shannon Mapping Algorithm

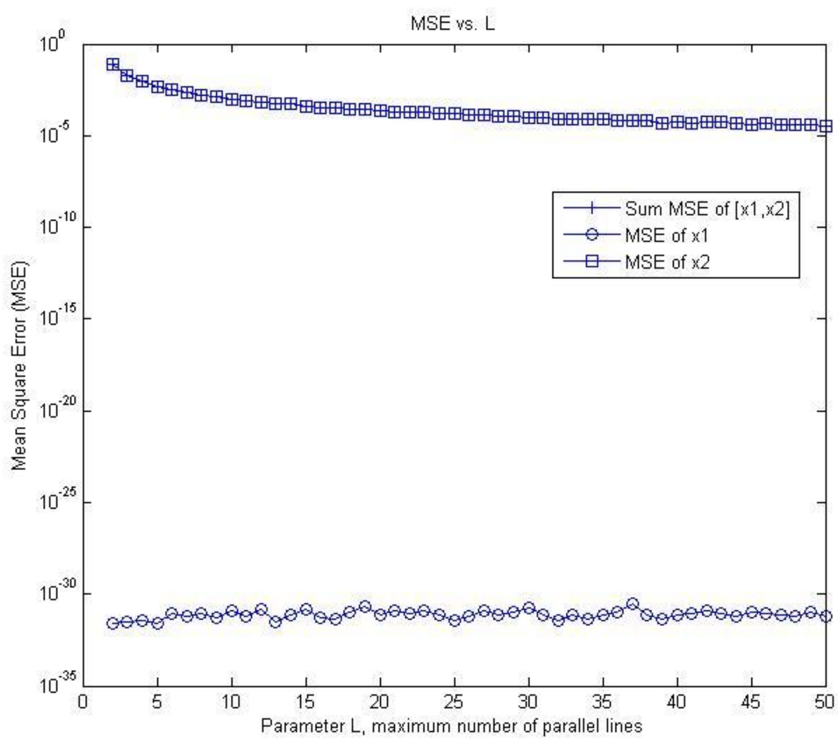


Fig 22. MSE caused by Shannon Mapping Algorithm

3.4 Simulation results with different signal length during the FM demodulation

Here are the results of MSE vs. L for signal length = 1000, 10000 and 1000000. The curves are not smooth because we reduced the number of loops for each L since large amount of iterations spend huge amount of time. But it doesn’t affect the observation of trend.

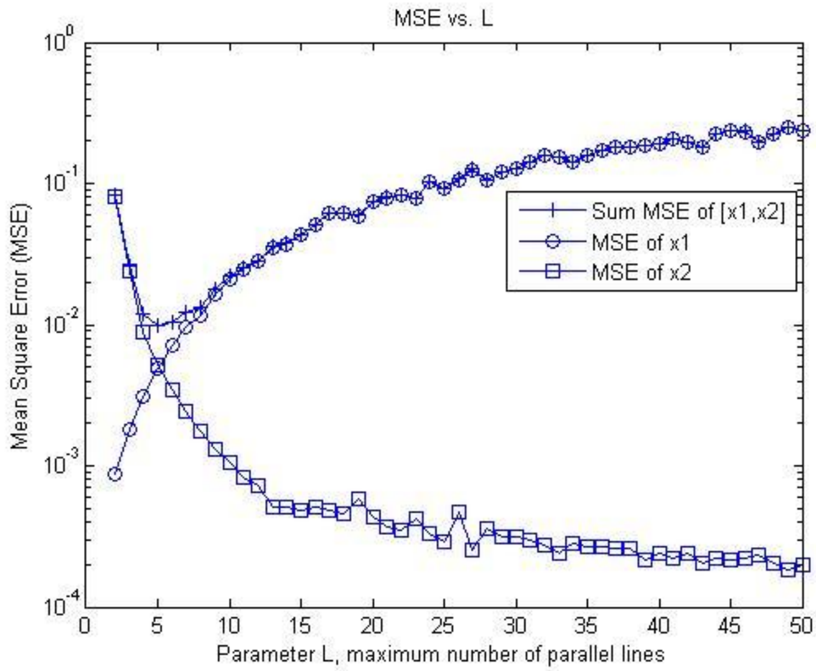


Fig 23. MSE vs. L for signal length = 1000

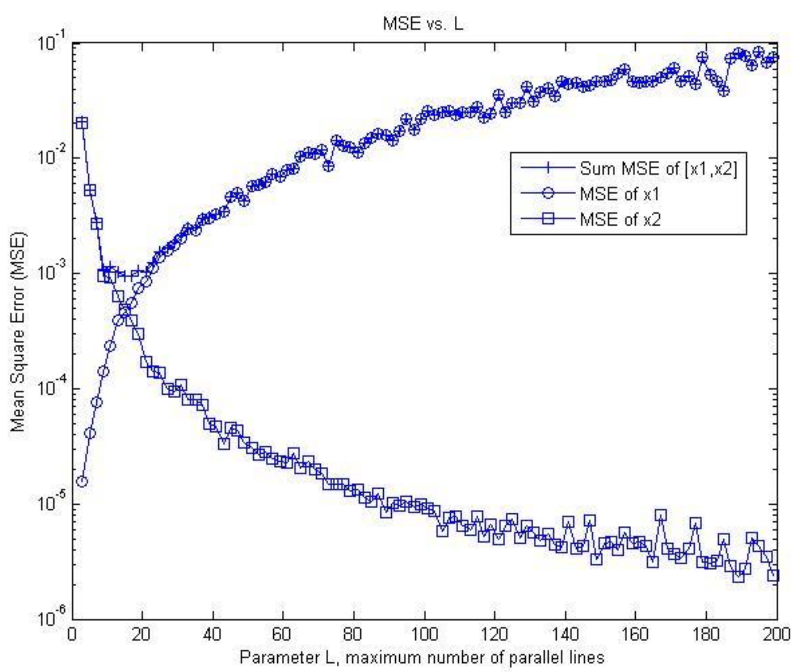


Fig 24. MSE vs. L for signal length = 10000

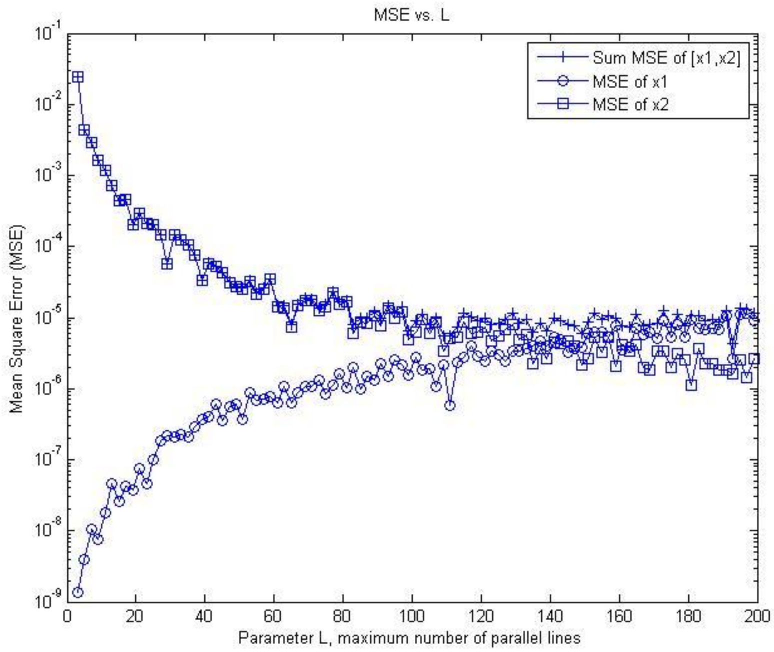


Fig 25. MSE vs. L for signal length = 1000000

4 Conclusion

Based on our previous works regarding AJSCC encoding and decoding, and Shannon mapping the focus of this project was more focused on FM modulation/demodulation and transmission over a noisy, Rayleigh fading channel. On observation of our results we found that FM modulation can result in a very low MSE, which is in the order of 10^{-4}, and the number of parallel lines required to achieve the minimum MSE is around 46.

From MSE vs. L curves under different SNR we found that the MSE stays relatively the same despite changes in the mean squared error. This may be attributed to the fact that FM modulation/demodulation is very resilient to noise, and signal strength variations.

From MSE vs. SNR curves with varying levels we found that the MSE decreases rapidly when SNR is around -30 dB. The system has a very good performance when SNR is higher than -30 dB. This result verified the claim from that FM modulation/ demodulation is very resilient to noise.

Comparing the results under Rayleigh fading channel and without the Rayleigh fading channel, the value of MSEs are very similar. That is because we simulated a scenario where the transmitter and the receiver are relatively static, which means there is no Doppler shift during the transmission. Thus the Rayleigh channel is a static channel that barely affects the results.

Figure 22 shows the MSE caused by the Shannon Mapping Algorithm. We observed that MSE of x_1 is very low, almost equal to 0, but MSE of x_2 and sum MSE are relatively high. Shannon Mapping itself will bring inevitable error that makes the value of sum MSE in the order of 10^{-5} to 10^{-4}. we think the value of sum MSE after the FM modulation and demodulation can not be smaller.

As mentioned in section 2.5, we also found that during the FM demodulation, the length of the signal affects the simulation results significantly whereas longer signals produce better frequency approximations, thus a better performance can be achieved. As shown in figures 23, 24 and 25 the MSE gets smaller while the length of signals increases, meanwhile, the optimal number of levels gets larger.

5 Applications of AJSCC in Wireless Sensor Networks

With Analog Joint Source Channel Coding being such an efficient, robust system for wireless sensor networks its applications, while not immediately foreseeable, can be widespread in the future. One application of interest is in bio sensing networks. A biosensor is an analytical device used for detection of various biomarkers. A biomarker is a measurable substance in an organism whose presence is indicative of some phenomenon such as disease, infection, or environmental exposure. One possible application for AJSCC is to create a biosensor that when implanted inside an organism will transmit data wirelessly to another device external to the organism. With the small, analog sensors being the implantable part of the sensor system, the Rayleigh fading model and FM modulation scheme would apply perfectly to this situation. The applications of AJSCC for biosensing networks are extremely preliminary and more work will be done in the future to figure out the best application for Analog Joint Source Channel Coding.

6 References

[1] Bernard Sklar. Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization. IEEE Communications Magazine. July 1997, 35 (7): 90–100.

[2] David Tse and Pramod Viswanath. Fundamentals of Wireless Communication. Cambridge University press. 2005. page 29-30.

[3] C. Shannon, “Communication in the Presence of Noise,” *Proceedings of the IRE*, 1949.

[4] X. Zhao, V. Sadhu, D. Pompili, “Analog Joint Source Channel Coding for Low-power Low-cost Wireless Sensor Networks.”

[5] X. Zhao, V. Sadhu, D. Pompili, “New Multiplexing Wireless Communication Methods Based on N-dimensional Shannon Mapping.”

[6] "Frequency Modulation." *Wikipedia*. Wikimedia Foundation, 5 Nov. 2015. Web. 22 Nov. 2015.

[7] Poole, Ian. "FM Demodulation / Detection Tutorial." *FM Demodulation*. Adrio Communications Ltd. Web. 22 Nov. 2015.

[8] Mathworks. Fast Fourier transform (R2015b). Retrieved November 10, 2015 from: http://www.mathworks.com/help/matlab/ref/fft.html#zmw57dd0e226281

7 Individual contribution

There was a pretty even division of work for this project. Everyone met and worked on the code and report together. However Lidan did a lot of the code debugging, Lulu worked on the original code and Angela worked on the original report.

8 Related links

8.1 Teamwork progress:

https://rutgers1.teamwork.com/projects/157157-remote-wireless-bio-sencing/overview

8.2 Git repository:

https://LidanWang@bitbucket.org/ajsccproject/matlab-code.git