Utilizing Spread Spectrum in Satellites Communications

Niruba Ratnavale

University of New South Wales at the Australian Defence Force Academy

Satellite communication systems, whether for telecommand and telemetry or as a communication service, have historically been heavily planned. This strategy has been viable as the number of satellites, and satellite networks have been low in the past, and new developments typically progress over the course of a decade. Dramatic reductions in launch cost and improvements in spacecraft capability promise to significantly increase the range and number of systems in orbit and reduce the development timescale to a year or two. Formal coordination of spectrum is therefore less viable in the future. In this report we explore the use of Direct Sequence Spread Spectrum or Code Division Multiple Access (CDMA) for satellite communication, since it offers the potential to optimally handle intermittent interference in non-coordinated operations. The simulation implements Direct Sequence Spread Spectrum and Binary Phased Shift Keying Modulation to create a signal with varying spreading rates. The simulations display increasing the spreading rate provides more protection against signals. Measuring the noise $\frac{E_b}{N_0}$ is useful as the spreading rate does not affect the results whereas calculating the Signal to Noise Ratio weakens the signal. Analyzing the effect of two signals examines the percentage of errors, the ability of transmitting and receiving data and resilience to interference. It is observed that with perfect power balancing, data can be transmitted and received where the signal and interference are well isolated from each other. Without power balancing, the signals are subject to interfering into each other and loss of data. Overall, CDMA should be implemented allowing many users to use the spectrum with little to no coordination. With spreading, users will have more protection against interference allowing data to be transmitted and received with minimum loss.

Contents

I. Introduction 2
   A. Aim 2
   B. Motivation 2
II. Literature Review 3
    A. Frequency Hopping Spread Spectrum 3
    B. Direct Sequence Spread Spectrum 3
    C. Modulation 3
    D. Interference 3
III. Methodology 6
IV. Results 7
V. Conclusions 10
VI. Recommendations 10
    Acknowledgements 10
    References 11
APPENDICES
Appendix A. 12

Nomenclature

$BER$ = Bit Error Rate
$DSSS$ = Direct Sequence Spread Spectrum
$EB/No$ = Energy per Bit/ Noise
$FHSS$ = Frequency Hopping Spread Spectrum
$PRN$ = Pseudo Random Number
$PN$ = Pseudo Noise
$SINR$ = Signal to Interference to Noise Ratio
$SNR$ = Signal to Noise Ratio

---

1 Niruba Ratnavale, School of Engineering & Information Technology. ZEIT4501

Final Project Report 2018, UNSW Canberra at ADFA
I. Introduction

A. Aim

The aim of this study is to investigate the utility of a spread spectrum waveform to facilitate a less coordinated approach to Satellite Communications.

B. Motivation

For decades, humans have been involved in space activities to understand more about the universe [1] [3]. With technology continuing to advance, research has helped develop capabilities such as satellite telecommunications to overcome global challenges. Currently there are 4,857 satellites orbiting the Earth with an increase of 4.79% from last year. A record of the number of satellites that continue to operate are kept by the Union of Concerned Scientists (UCS). As at April 2018, there are 1,980 active satellites continuing to orbit with an increase of 13.92% from last year. In space, there are many debris interfering with capabilities in space and the different orbits. There is a total of 2,877 pieces of metal in space which is 20 less from the previous year. It is likely that each week a satellite returns to Earth due to the debris, deorbiting or being burnt up in space. For example, in 2017, 453 objects were launched with only 390 remaining by April 2018 with a 14% reduction.

In the previous era, there were only a few satellites that were expensive but now that access has become cheaper and technology is improving, moving past and letting go of coordination will enable rapid innovation [3]. Satellite Communications has been around for a long time as telecommunication networks are taking part our day to day lives [3] [5]. For communications, satellites use radio frequency to enable a sufficient amount of bandwidth where each satellite is allowed to operate on a set frequency band. This approach is time consuming and highly inefficient. There are not enough frequency bands to be allocated for all satellites and the process of attaining approval takes time.

There are many advantages and disadvantages and one disadvantage is the lack of coordination between satellites. Satellites share one frequency band where coordination is a critical procedure to ensure minimum interference [3] [4]. This procedure involves registration and approval by the International Telecommunication Union (ITU). This results in a long process to allow new satellites to be coordinated amongst the existing satellites. There is only a certain amount of bandwidth availability and when sharing, the performance of the satellites is degraded.

Other studies previously done include spatial reuse with frequencies, isolation using high gain antennas [3] [6]. This paper will be focusing on isolation between spacecraft using spread spectrum. The study on spatial reuse implements regulation to reduce interference by separating the satellites physically and transmission through antenna with high gains at the ground station. Satellites will be able to transmit and receive at the same time using the same frequency reusing the frequency band. The motivation for this study is to show that spread spectrum is a method to modulate and demodulate signals with little to no data loss from other emitters on the same channel.

Using Direct Sequence Spread Spectrum or Code Division Multiple Access will help deal with the lack of coordination [3]. By implementing spreading, the channel is shared amongst users without any need for coordination and allocation of time slots. Each user is only required to have a unique Pseudo Random Number (PRN) i.e. the spreading code. This allows more users to use the bandwidth with more flexibility. With the increase in bandwidth, there will be minimum data loss and interference, showing coordination is unnecessary.

II. Literature Review

A. Frequency Hopping Spread Spectrum

Spread spectrum uses the idea of expanding the bandwidth beyond the required amount. In doing this, the signal power density is low, making detection potentially difficult [2]. There are a few different methodologies used for spread spectrum such as FHSS (Frequency Hopping Spread Spectrum) and DSSS (Direct Sequence Spread Spectrum). FHSS implements a carrier frequency where the frequency hops in different intervals. The transmission is controlled by a Pseudo Noise (PN) generator. The PN generated for transmission must be synchronized to successfully receive the signal.
B. Direct Sequence Spread Spectrum

DSSS is more commonly used due to limitations in FHSS such as difficulty in maintaining synchronization in the carrier frequency [2]. In DSSS, a PN code which is made up of 1’s and 0’s, also known as a chip, is used to spread the signal. The nature of the PN presents difficulties for users to intercept the signal. Multiplying the signal with spreading with a known signal i.e. a matched filter results with the received signal. Code Division Multiple Access (CDMA) implements DSSS to allow multiple users with a unique PN code for the signal. This provides access to many users without wasting the bandwidth.

C. Modulation

Different modulation techniques may be applied such as Binary Phased Shift Keying (BPSK). BPSK is made up of two phases separated by an angle of 180˚ which map to the following symbols ‘1’ and ‘0’ on a constellation diagram as shown in Figure 1 [2] [3].

Using the constellation diagram, bit ‘0’ will be mapped to ‘+1’ and bit ‘1’ will be mapped to ‘-1’ to implement BPSK [3]. Only 2 decisions are required to be made. If it is greater than 0, bit ‘0’ will be transmitted and if it less than 0, bit ‘1’ will be transmitted. The advantages of BPSK include the simplicity and efficiency in power.

This modulation is preferred over Quadrature Phase Shift Keying (QPSK) because it will require more decisions in each of the quadrants as shown in Figure 2 [3]. If it is transmitted in [1,+]j, bit ‘0’ will be transmitted, if it is transmitted in [1,-j], bit ‘1’ will be transmitted, if it is transmitted in [-1,-j], bit ‘3’ will be transmitted and if it is transmitted in [1,-j], bit ‘4’ will be transmitted. These decisions increase the complexity in detecting errors.

D. Interference

An issue in spread spectrum is the near-far problem where transmitters nearby interfere with the desired (distant) transmitter whilst receiving. This is overcome by filtering out weak signals whilst keeping the strong signals present [2] [3]. For example, if there are two sources and one receiver, the transmitter closer to the receiver will output higher power because the intensity of the power is inversely proportional to the distance i.e. 

$$P = \frac{1}{Distance^2}$$

The near-far problem will eliminate the transmitter that is the furthest away because of its weak output signal.
Signals propagating can be affected by many factors resulting in attenuation in the signal [3] [7]. Figure 3 shows that the received amplitude of Source 1 is 100 times larger than the amplitude in Source 2. Source 1 is the stronger signal due to the relationship of power and distance mentioned above which will hide it from the weak signal, Source 2. The distance of the sources affects misdetection by the receiver.

![Figure 3: Sources at different distances for the near far problem [3] [7]](image)

Even if the sources are the same distance apart from the receiver, the direction of the source effect misdetection of the source as shown in Figure 4 [3] [7].

![Figure 4: Sources at the same distance for the near far problem [3] [7]](image)

Detecting the near far problem is achieved by using a matched filter for the output of the source and normalizing the energy of the input signal [3] [7]. Using this process, the output peak of the strongest pulse will have an amplitude of approximately 1 and the output peak amplitude of the weakest pulse will be 0. If both the pulses have the same energy, the output peak will be 0.5. Comparing the output peak of the pulses enables detection of the near far problem. A threshold can help identify the presence of the near far problem. If a pulse does not pass the threshold, the near far problem is applicable.

The output of a matched filter for a weak pulse results in noise whilst a strong pulse is present leading to misdetection [3] [7]. This is due to the amount of energy in the weak pulse being much stronger than the strong pulse as shown in Figure 5.

![Figure 5: Output of a matched filter [7]](image)
These results change when the energy is normalized for the matched filter [7]. The weak pulse will not be present and does not affect the decision-making process. As shown in Figure 6, the peak amplitude of the strong pulse is approximately 1 allowing this pulse to be detected.

The advantage of using spread spectrum is the amount of data protection provided [2]. The signal is much more resilient to noise because the amount of spreading PN code is unknown. A narrowband jammer is more efficient than a broadband jammer because it only covers a small range of frequencies with a high-power density. The correlation between these are minimal where the signal after de-spreading has little or no effect. However, broadband jammers will cover a large range of frequencies and do not work efficiently.

The Minimum Mean Squared Error (MMSE) can help reduce the interference when DSSS and CDMA are implemented [8]. Schemes to utilize multiuse detection are complex to implement to reduce the near-far problem where this approach can maintain resistance to the near far problem. To receive a signal, a matched filter is used by applying correlation. Matched filters are used to control the power receiving feedback from the receiver.

A scheme that was proposed sampled the output at the PRN using a Finite Impulse Response (FIR) filter to reduce the mean squared error (MSE) between transmitting the detected symbols [8]. An MMSE linear detector uses an infinite length of delay line where the symbols and PRN are synchronized at the transmitter. If the PRN’s are not synchronized, the spacing of the delay line needs to be reduced. If the symbols are not synchronized, an infinite number of taps are required. Results claim that despite the symbols or the PRN not being synchronized, there is an improvement using the N-tap MMSE detector compared to the matched filter detector. The Cyclically Shifted Filter Bank (CSFB) scheme samples the filter output at a symbol rate and combining it with taps, where these filters have cyclic shifts. This scheme is similar to sampling a matched filter, $D$ times for each symbol period.

Numerical results using a spreading gain of 31 and oversampling, the schemes with the N-tap detector and CSFB can be compared [8]. Figure 7 shows the results where the N-tap MMSE detector is lightly sensitive to delay and is much higher than the CSFB method. The N-tap scheme consists of many delays providing an advantage of resisting interference. Using either of these schemes compared to the matched filter is recommended. These schemes help minimize interference as well as intersymbol interference.
III. Methodology

Spread spectrum is used in communication to secure communication by spreading the signal by a length, \(L\). This is achieved by using a PRN (Pseudo Random Number) to expand the bandwidth. The PRN is multiplied with each data bit using the Exclusive OR (XOR) logic gate.

For example,

Data = \([1, 0, 0, 1]\)
PRN = \([1, 0]\)
Signal = \([0, 1, 1, 0, 1, 0, 0, 1]\)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: XOR Logic [3]

Figure 8: Transmitting and Receiving a Signal

Implementing the spreading of a signal, transmitting and receiving a signal is depicted in Figure 8. Using random 1’s and 0’s for the input data and PRN, the transmitted signal is created by using the XOR function for the data and PRN. To receive the signal, the XOR function is used between the transmitted signal and the PRN. It is noticeable that both the input data and received signal are the same which is expected.

The signal length is the data length multiplied by the PRN length i.e. the spreading of the signal. Different types of modulation may be implemented however in this instance BPSK will be used. BPSK modulation consists of two symbols, ‘1’ and ‘0’ which are mapped to +1 and -1 respectively using bipolar signals. To use this concept in the simulation, the signal must change from 0’s and 1’s to +1’s and -1’s by the operation \(2 \times \text{signal} - 1\). Continuing the example above, the signal will be \([-1, 1, 1, -1, 1, -1, 1]\).

Once the signal has been generated, random noise is added to the signal because in real life situations noise is always present. The noise can be measured as Signal to Noise Ratio (SNR) or \(\frac{E_b}{N_0}\) (Energy per bit/Noise). The variable \(\frac{E_b}{N_0}\) is the ratio of the signal energy for each bit and the noise spectral density. The variable is calculated by taking the average of the signal and dividing it by the average of the noise and multiplying each energy bit by the spreading rate. To calculate the SNR, the average of the signal to noise ratio is calculated.

A matched filter is a known signal i.e. the signal with spreading is used to make decisions on which bit is being transmitted. The received signal is multiplied with this signal to calculate the errors. In digital communications, if the bits area represented with 1’s and -1’s, the dot product is used to obtain the result and this operation is equivalent to using an XOR. Due to the signal being sinusoidal, the ‘sign’ function is used to convert the received signal to +1’s and -1’s where if the transmitted signal is greater than zero, bit ‘1’ is transmitted and if the transmitted signal is less than zero, bit ‘0’ is transmitted. The error is calculated by subtracting the received
bits with the initial data input by using the following calculation \( \frac{\text{receive bits} - \text{data}}{2} \). The average of the ratio of the bit errors and the transferred bits calculates the Bit Error Rate (BER). The theory curve is generated using the formula \( 0.5erfc\sqrt{\frac{E_b}{2N_o}} \). Using tools such as Matlab and plotting the simulation and theory curve will show the performance of spread spectrum with varying spreading rates.

Another signal is included to analyse the effect of interference where two signals or more signals competing against each other. This is achieved by adding another set of input data and Pseudo Random Number (PRN). The length of the input data and PRN will be the same, but it will be a different sequence. The amplitude is a variable that can be varied for the interference signal however it is important to note that the amplitude is measured in volts. The transmitting signal is a combination of the signal with spreading and interference signal with spreading. This will show the effect of data transmission, occurrence of loss and errors. The Signal to Noise Interference Ratio (SINR) is not included in the calculations when calculating \( \frac{E_b}{N_o} \) and SNR.

IV. Results

Using the methodology above, the results produced BER plots by applying BPSK modulation with a spreading rate of 1, 10 and 100. The plots are measured with two types of noise where the first variable is \( \frac{E_b}{N_o} \) and the other is SNR. The spreading of the chip is implemented when measuring each energy bit for \( \frac{E_b}{N_o} \). For SNR, the spreading rate is ignored, and it is purely the signal to noise ratio. Observing Figures 8, 9 and 10, the plots show a theory and simulation curve for both \( \frac{E_b}{N_o} \) and SNR. The purpose of this is to ensure the theory matches the simulation to validate the data and to show that it is reliable.

Comparing the \( \frac{E_b}{N_o} \) and SNR, the spreading rate does not affect the \( \frac{E_b}{N_o} \) and each plot essentially is the same for each spreading rate. However, measuring the SNR is not efficient as it deteriorates as the spreading gain increases. The increase in spreading gain causes a weak SNR and a shift in the plot to the left. The amount of shift can be determined using \( 10 \log_{10}(\text{spreading rate}) \).

![Figure 9: Spreading Rate of 1](image)

Figure 9 shows the spreading rate with the x-axis is labelled as SNR. This is the appropriate label for the figure as there is no spreading applied to the signal. Figure 10 and 11 have an x-axis of SNR which is not necessarily correct due to the signal measured with two different types of noise i.e. \( \frac{E_b}{N_o} \) and SNR. It is difficult to decide the correct title for the x-axis but as the spreading rate increases, it should read as \( \frac{E_b}{N_o} \).
The noise is initially hard coded in the simulation to generate the x-axis for the signal to be between -10 to 10. This range is chosen to verify the plots against the theory curve in the textbook. However, at the end of the loop, $\frac{E_b}{N_0}$ and SNR are calculated for the simulation. This approach enables the noise power to be calculated using $\frac{E_b}{SNR(1)}$ in a loop and creates the noise using the noise power and the random function. The legend identifies the blue curve measuring $\frac{E_b}{N_0}$ and the red curve measuring SNR. The importance of calculating $\frac{E_b}{N_0}$ over SNR is also recognized in calculating the theory BER curve which is measured with the variable $\frac{E_b}{N_0}$.

Table 2: Input and output for PRN’s [3]

<table>
<thead>
<tr>
<th>PRN of Signal</th>
<th>PRN of Interference Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
In BPSK there are two bits transmitting where there are four possible outcomes as shown in Table 2 where the PRN’s are either orthogonal or parallel to each other [4]. There is a 50% chance to get 100% match between the two PRN sequences and a 50% chance to receive 50% match between the PRN sequences. The total match is calculated to be \((50\% \times 100\%) + (50\% \times 50\%) = 75\%\) match with expected errors of 25%. If the PRN sequences are orthogonal, interference is limited and if they don’t match, there is a 50:50 chance for perfect power balancing. If power balancing is not present, the interference dominates the signal.

Figure 12 shows a 1:1 ratio where the amplitude and rate are both set to 1. The BER vector is as follows: [0.3382, 0.3144, 0.2884, 0.2694, 0.2550, 0.2511, 0.2487, 0.2480, 0.2496, 0.2512]. This BER vector shows the plot has 25% errors because of perfect power balancing [4]. If two signals transmitting are competing against each other where they will go in the same direction where minimum data loss is otherwise maintained else, they go against each other. Without power balancing, only one signal is likely to transmit data successfully.

Figure 13 shows a spreading rate of 5 with interference signal.

Figure 14 shows a spreading rate of 10 with interference signal.

Figure 13 and 14 show an amplitude of 2 and the spreading rate increased from 5 to 10 respectively. With these results, there is a point in the curve where it is interference limited i.e. the theory and simulation curve match up and there is a point in the curve where the simulation curve diverges away from the theory curve as shown in Figure 13 where the theory curve is purely noise without any interference. Figure 14 shows perfect isolation between the signal and interference signal. The advantage of increasing the spreading rate ensures both signals transmit and receive data without being affected by each other. This also provides more protection against interference. Running this simulation multiple times results in different outcomes where this outcome shows the signals are orthogonal where the PRN’s cancel out.
V. Conclusions

Satellites have evolved over the past few decades continuing to be applied to commercial and military applications [3]. Spread spectrum can be implemented using CDMA and BPSK modulation. Using DSSS efficiently uses the whole bandwidth spreading it by a factor, L. CDMA implements DSSS to allow many users to use the bandwidth efficiently with minimum data loss and minimum interference. If this method is implemented, coordination is not required providing more flexibility for users. Once spreading is implemented, the signal measuring $\frac{E_b}{N_0}$ is not affected by the spreading rate whereas the signal measuring SNR is translated by a factor of $10 \log_{10}(\text{spreading rate})$ from right to left. With spreading, each signal is subject to $N$ times amount of protection where $N$ is the spreading rate.

All signals are subject to interference such as the near far problem [3]. The advantage of satellites is that the separation between satellites is so large, the near far problem will not impact them. Antennas at the ground base station point at the desired satellites to transmit and receive. However, power balancing will be maintained by ensuring the transmitters are balanced to avoid the near far problem. Observing the effect of two signals competing against each other, with power balancing, both signals can transmit and receive data without interfering with each other. If power balancing is not present, the interference signal will transmit and receive more data dominating the other signal. Implementing spreading provides isolation for both signals to transmit and receive data. This is because SNR cannot reach infinity. The signal expands with the spreading rate, where the interference signal will also increase, growing linearly with the signal.

VI. Recommendations

Recommendations for this project include validating the Matlab code. This includes to confirming that the simulation matches the theory curve. In the code, the input data and PRN are outside the loop therefore it is not generated a number of $n$ times. The same sequence for both the data and PRN is used for running the simulation, therefore the data and PRN are not purely random. If the input data and PRN are to be generated inside the loop, it could produce more bit errors.

Power balancing is vital to successfully transmit and receive information with minimum data loss and any interference [3]. The higher the power density, measured as $W/MHz/m^2$, the more resilient it is to noise. For every square metre on Earth, the designer can decide the amount of power that should be received by spacecrafts. This requires variations of power because everyone will be radiating large amounts of power which is inefficient as they will not be continuously using the spectrum. A reasonable value should be set by the ITU for regulation. This is a design trade to power balancing and to overcome it, a user wanting to radiate a low amount of power should have the ability to shrink the spectrum down as desired. Antennas are used to transmit and receive, and due to the directivity of the gain, less power is required at the earth’s surface. If this regulation is implemented and maintained by the ITU, it will require less coordination.

Spreading code generally uses short codes however there are not enough combinations [3]. The code will not be orthogonal and instead it be parallel where no isolation can occur. Using a long sequence code is more efficient where there will be an infinite number of combinations, and the code is more likely to be orthogonal and will provide isolation. In the future, as the number of satellites in orbit increase, implementing long sequence PRN’s will be an advantage. A short simple code will not provide enough protection whereas a randomly generated long code means it is highly unlikely for another user to have the same code providing more independence between signals. Frequency Division Multiple Access (FDMA) has a limited set of codes where as CDMA is naturally non-coordinated.

Acknowledgements

I would like to acknowledge my supervisor Dr Craig Benson for assisting me, providing guidance and support throughout this project with his expertise in this field.
References


6. Stevens, D. (2017), Proposals for Allocation of 3.4GHz and 5.8GHz bands to Small Satellites in LEO, University of New South Wales at the Australian Defence Force Academy


Appendices A

Signal implementing spreading

numbits=100000;
rate = 10;
umruns = 1;
Eb = 1;
SNRdb = -10:2:10; %SNR dB
SNR = 10.^((SNRdb)/10); %SNR magnitude
BER = NaN(size(SNR));
BER_th = NaN(size(SNR));
EbonNo = NaN(size(SNR));
data = round(rand(numbits)); %random data of 1’s and 0’s
PRN1 = round(rand(1,rate)); %random PN of 1’s and 0’s

for i= 1:length(SNR)
    No = Eb./SNR(i); %noise power, 0.1
    avg = 0;
    for j = 1:numruns
        for m=1:length(data)
            for n=1:length(PRN1)
                sig((m-1)*length(PRN1)+n)=(xor(data(m),PRN1(n)));%xor
            end
        end
        sig = 2*sig - 1; %signal with +/- 1
        N = sqrt(rate)*sqrt(No/2)*randn(1,length(sig));
        sig_n = sig + N; %signal combined with noise
        for k= 1:numbits-1
            matched_filter=sig(1:rate);
            receive(k) = sum(sig_n(rate*k+1:rate*(k+1)).*matched_filter);
            receivebits(k)=sign((receive(k)));
        end
        errors = abs(receivebits-(2*data(2:end)-1))/2;
        EB = rate*sum(sig.^2)./numbits;
        S = sum(sig.^2)./numbits;
        Noise = sum(N.^2)./numbits;
        BER(i) = mean(errors);
        EbonNo(i)= EB/(2*Noise);
        SNR(i) = S/(2*Noise);
    end
    BER_th = (1/2)*erfc(sqrt(EbonNo));
end

Signal implementing spreading with an interference signal

numbits=100000;
rate = 5;
amplitude = 2;
umruns = 1;
Eb = 1;
SNRdb = -10:2:10; %SNR dB
SNR = 10.^((SNRdb)/10); %SNR magnitude
data = round(rand(1,numbits)); %random data of 1 and 0’s
data2 = round(rand(1,numbits)); %interference
PRN1 = round(rand(1,rate));
PRN2 = round(rand(1,rate));

for i= 1:length(SNR)
    No = Eb./SNR(i); %noise power, 0.1
avg = 0;
for j = 1:numruns
    for m=1:length(data)
        for n=1:length(PRN1)
            sig((m-1)*length(PRN1)+n)=(xor(data(m),PRN1(n)));
        end
    end
    for a=1:length(data2)
        for b=1:length(PRN2)
            sig2((a-1)*length(PRN2)+b)=(xor(data2(a),PRN2(b)));
        end
    end
    sig = 2*sig-1; %signal 2
    sig2 = 2*sig2-1; %interference signal
    sig_comb = sig+amplitude.*sig2;
    N = sqrt(rate)*sqrt(No/2)*randn(1,length(sig));
    sig_n = sig_comb + N; %signal combined with noise
    for k= 1:numbits-1
        matched_filter=sig(1:rate);
        receive(k) = sum(sig_n(rate*k+1:rate*(k+1)).*matched_filter);
        receivebits(k)=sign((receive(k)));
    end
    errors = abs(receivebits-(2*data(2:end)-1))/2;
    EB = rate*sum(sig.^2)/numbits;
    S = sum(sig.^2)/numbits;
    Noise = sum(N.^2)/numbits;
end
BER(i) = mean(errors);
EbonNo(i)= EB/(2*Noise); %explain noise or signal factor of p
SNR(i) = S/(Noise);
end
BER_th = (1/2)*erfc(sqrt(EbonNo));