CDMA Simulation

By:

Alison Hightman
Elaine Chung
Alejandra Galvan

CDMA Simulation

By:
Alison Hightman
Elaine Chung
Alejandra Galvan

Online:
< http://cnx.org/content/col11388/1.1/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: December 19, 2011
PDF generated: December 19, 2011
For copyright and attribution information for the modules contained in this collection, see p. 22.

Table of Contents

Chapter 1

Why does CDMA matter?[1]

CDMA is a crucial part of today's communications technologies especially cellphone/wireless communication. CDMA allows us to have as many users as we need in a limited range of channels. It is also the basis for 3G technologies which use wide-band CDMA and many major cellphone companies, such as Verizon, utilize CDMA in their phones.

In general CDMA is an effective way to transfer data, such as voice and multimedia. These uses are seen in cellphone communications as well as other applications. Other applications include transfer of high resolution images using compression techniques on the image and then CDMA to quickly and safely (without losing resolution/data) transfer said images. Seeing the importance and value of understanding CDMA concepts as an electrical engineer, especially if one is interested in wireless communications, we saw the need to develop a versatile simulation to model a CDMA system and illustrate the effects of various parameters.

[1]This content is available online at <http://cnx.org/content/m41808/1.1/>.

Chapter 2

Our Project[1]

Our objective in this project was to improve our own understanding of CDMA and create a tool for others to use for the same purpose. The user would be able to see how the efficiency and error rate of a CDMA system changes based on its input parameters. By observing and analyzing the trends in the error rates the user would be able to better understand the effects of altering specific parameters on the system. Our simulation is designed to be a versatile model that can simulate many different scenarios and parameters. Our simulation aims to optimize efficiency of data transmission and minimize error rate. In order to create this toolbox we must be able understand all possible effects of changing parameters in our simulation and establish some limitations for the system and work within those constants.

Our finished code is available for download here[2].

[1] This content is available online at <http://cnx.org/content/m41801/1.1/>.
[2] See the file at <http://cnx.org/content/m41801/latest/cdma_sim_long_sequence.m>

Chapter 3

CDMA Background [1]

Code division multiple access, or CDMA, is a channel access method often used in radio/cell phone communication. CDMA employs the spread-spectrum modulation format, which has the advantage of making the signals resistant to narrowband interference, since it would only affect a small portion of the spread spectrum signal it is easy to filter out without a significant loss of data, difficult to intercept and resistant to jamming.

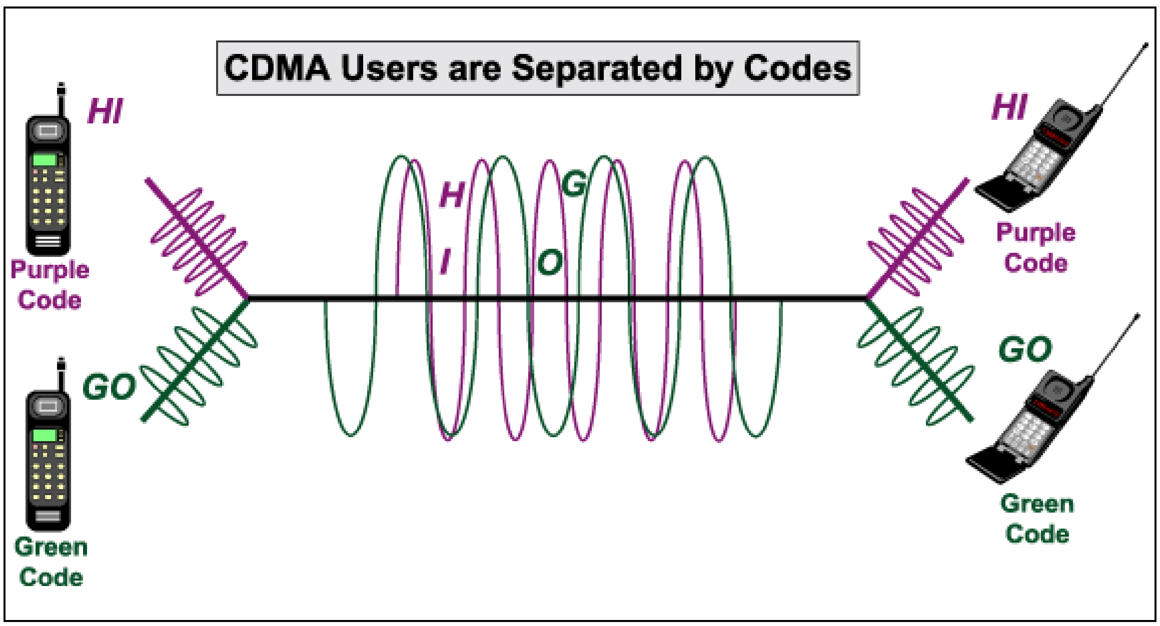


Figure 3.1

Direct-sequence spread spectrum for CDMA involves multiplying the data to be transmitted by a pseudo-random sequence of -1's and 1's. This spreads the original signal into a much wider band, hence the term "spread-spectrum." The original data can be restored by the receiver using a de-spreading technique. This involves de-spreading the received signal with the pseudo-random sequence respective to the original signal. To do this, the transmitted and received sequences must be synchronized.

[1] This content is available online at <http://cnx.org/content/m41832/1.1/>.

CDMA uses orthogonal coding to allow multiple users to access a channel simultaneously. An advantage of CDMA is that it is possible to keep increasing the number of users for a system. In comparison, other techniques for multiple access, such as frequency and time division, rely on assigning a specific frequency or time slot to each user. This greatly limits the efficiency of a system, since the number of frequencies available for wireless communication is limited, and the use of time division only allows for one user to access the channel at a time. Essentially FDMA and TDMA set hard limits on the number of users a system can have because there is no way to continue making divisions past a certain point. On the other hand, with CDMA it is possible to keep adding users. The limiting factor of CDMA is the error rate, which increases with each added user.

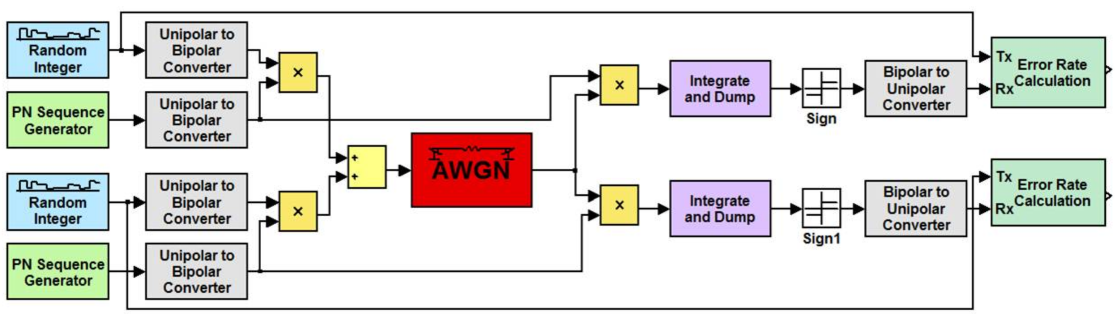
For our model we assume that we know when the signal arrives at the transmitter, so we can assume that the modulation and de-modulation codes are synchronized. Each user has a pseudo-random noise (PN) sequence that is orthogonal to other users' PN sequences. Their cross-correlation is close to 0. The orthogonality of the codes ensures that each signal's transmitter-receiver pair will only deal with its corresponding signal; in other words, the receiver for a particular signal will see any other signal going through the channel as simply noise, and this will eliminate multiple-access interference.

In direct sequence spread spectrum, there are two types of spreading sequences: short and long. In short spreading sequences, the spreading sequence is periodic with a period equal to the processing gain. As a result, the same short sequence is used to modulate each symbol which is transmitted. Examples of short spreading sequences are Walsh Codes and Gold Codes. In long spreading sequences, the spreading sequence is either aperiodic or has a period which is much longer than the processing gain. We chose to use long spreading sequences in our model because long spreading sequences are used in the IS-95 standard. Long spreading PN sequences can be generated using maximum length shift registers. To acquire sequences for each user which are close to mutually orthogonal, each user is assigned a shifted version of the PN sequence.

Chapter 4

Overview of Our Model[1]

Our CDMA simulation will help users understand the basic concepts of CDMA and the effects of the various parameters on error rate. Our code is available for download here[2]. The MATLAB code is based on this simple model of CDMA.



We initially used Simulink to model CDMA, but since each user requires their own random sequence and PN sequence, we had no easy way to add multiple users. We based the code for our MATLAB model off of the setup of our Simulink model.

The inputs to our MATLAB simulation are the processing gain, length of sequence to be transmited, signal-to-noise ratio (SNR) of our additive white Gaussian noise (AWGN) channel, number of users, and the degree of the generator polynomial of our MLSR, which affects the period of the PN sequence. Each user will transmit a data sequence of 0s and 1s, determined by a random number generator. These random sequences are determined independently. Each user will also be assigned to a distinct pseudo-random noise (PN) sequence of 0s and 1s. Since we are using a long spreading sequence, the PN sequence will be periodic with a period which is much longer than the processing gain. The generation of pseudo-random sequences will be discussed in a later section.

Each user's data sequence must be modulated by a corresponding PN sequence, so we used a unipolar to bipolar converter to change the 0s and 1s of the data and PN sequences to -1s and 1s. To get the baseband waveform which will be transmitted, we take the product of the converted data and PN sequences. Since the idea of CDMA is for multiple users transmit information over a single channel, we sum the baseband waveforms of all of the users. Our simulation is conducted at baseband, so we will now send the accumulated sum of waveforms through the AWGN channel.

We are interested in the errors caused by the channel, so we need to compare the transmitted signals to the received signals. Since each user will experience approximately the same error rate, so we only need to demodulate the data for one user to estimate the bit error rate of our system. To demodulate, we multiply

[1] This content is available online at <http://cnx.org/content/m41829/1.1/>.
[2] See the file at <http://cnx.org/content/m41829/latest/cdma_sim_long_sequence.m>

the received output of the channel by the PN sequence corresponding to the user whose sequence we want to receive. Then, we integrate and dump for each transmitted bit. Since the PN sequences are roughly orthogonal, integrating will isolate the waveform corresponding to the PN sequence we used to demodulate. Then, we can use the sign of the integration result to determine whether the received bit is -1 or 1. We can convert this result back to 0s and 1s with a bipolar to unipolar converter.

We determine the number of errors in our system by comparing the received and transmitted sequences. Going through both sequences bit by bit, we count an error whenever the two do not match.

Chapter 5

Generating the Spreading Sequence[1]

To generate a PN sequence in MATLAB, we used the built-in PN sequence generator, which uses a linear feedback shift register (LFSR) to determine the sequence. An LFSR is a set of shift registers connected with a feedback configuration. The register values update simultaneously during each cycle. An important input parameter of an LFSR is the generator polynomial. The degree of the generator polynomial determines the number of registers in the LFSR. For example, an LFSR with a generator polynomial of degree 3 will have three registers. The coefficients of the generator polynomial, which are either 0 or 1, determine the feedback connections. A coefficient of 1 for a particular term means the feedback is connected to the corresponding register, while a coefficient of 0 means there is no feedback connected. The coefficients of the first and last term of the generator polynomial must be 1.

Another important parameter for an LFSR is the initial state of the registers. The initial state cannot be 0 for all registers because if they were, no amount of feedback could make the registers change to any other value. By default, the initial state of the register representing the least significant bit is 1 and all other registers are 0.

When our generator polynomial has degree r and we have r registers, the registers can display at most 2^r-1 different states, because the state of all zeros can never come up if we want the states of our register to change. Since the output PN sequence will be read off one particular register, typically the right-most register, the period of the PN sequence will also be 2^r-1. Not all generator polynomials will yield the maximum length PN sequence. We assign a specific generator polynomial for each degree to achieve the maximum length PN sequence. We select these from a table of values provided for use with MATLAB which specifies which generator polynomial to use to achieve the maximum length.

The maximum length shift register is good for generating sequences with long periods. To create a spreading waveform for each of the users in a system, we assign each user a delayed version of the same sequence. A property of PN sequences is they are roughly orthogonal to delayed versions of themselves, so each user will have a spreading waveform which is approximately orthogonal to other users.

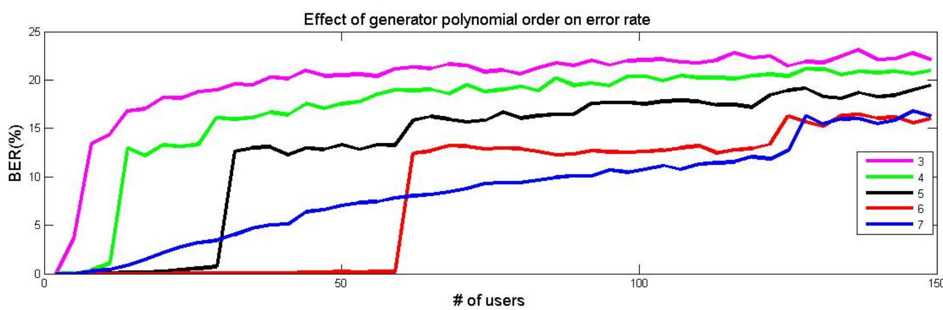
[1]This content is available online at <http://cnx.org/content/m41827/1.1/>.

Chapter 6

Results[1]

In order to design an effective analysis tool, we have to be familiar with the effects of all the parameters we make available (processing gain, sequence length, SNR of channel, number of users, degree of generator polynomial). We tested our model and concluded our results were consistent with what we would expect from a CDMA system. The following are some relationships between some parameters and the bit rate error we found especially interesting.

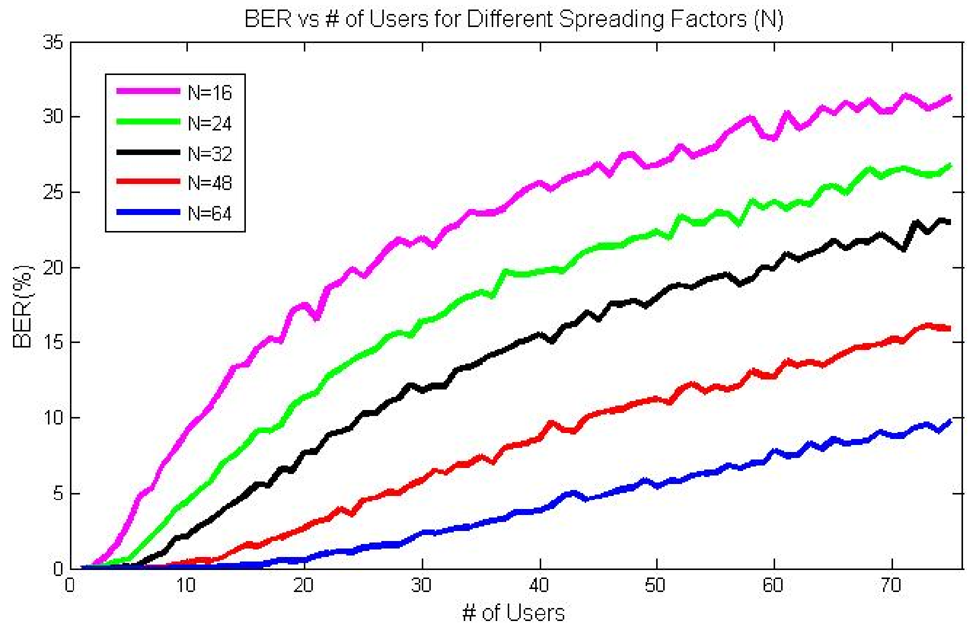
Generator polynomial vs number of users: For an polynomial of order x, the number of users must be less than 2x in order to minimize bit error ratio. For any number of users N, the minimum order polynomial that corresponds to the number of users >N must be used in order to minimize error.



Number of users vs number of errors: As the number of users in a system increases, the bit error rate increases as well. Modulation sequences are assigned to each user by taking one PN sequence with a long period and assigning users delayed versions of the sequence. Increasing users will cause each user's sequence to be less orthogonal with the rest of the users, resulting in more errors at the receiving end.

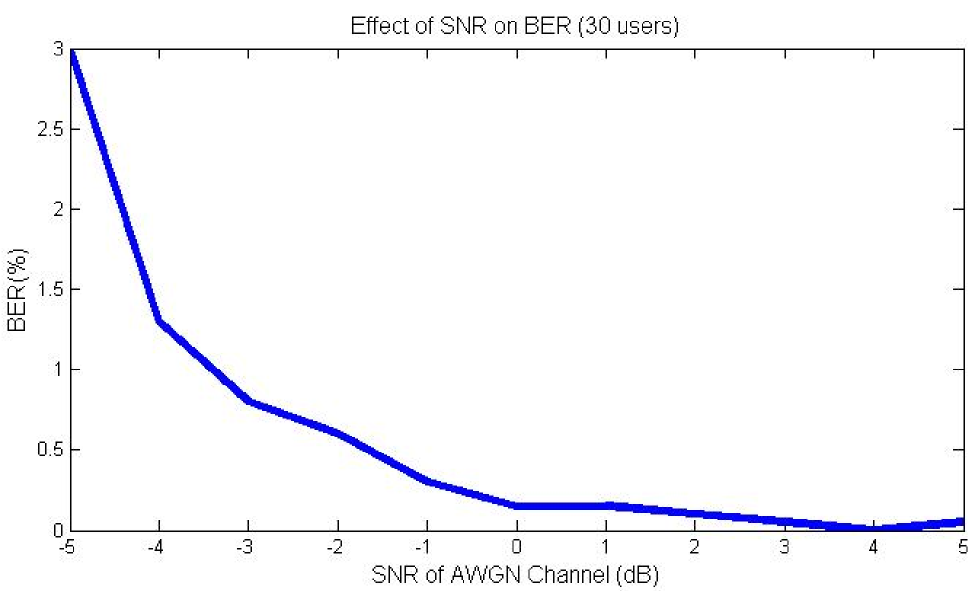
Processing Gain vs number of errors: The processor gain (also referred to as spreading factor) is the ratio of the bandwidth of the pseudo-random code to the bandwidth of our desired signal. The processing gain is also the number of "chips" used to transmit a single simple. A greater the processing gain results in fewer errors in our received signal. However, in practical applications, processing gain is limited by the available bandwidth. We plotted the effect of different processing gains against the bit error rate to observe the trade off between error rate and processing gain. This plot is shown below:

[1]This content is available online at <http://cnx.org/content/m41804/1.1/>.

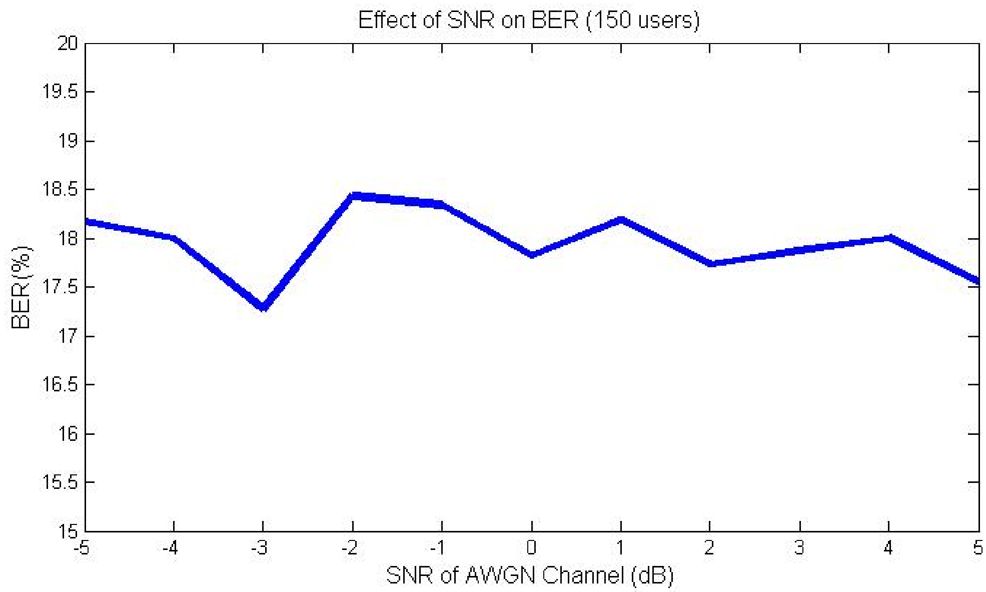


SNR vs number of users: In our model, SNR specifies the ratio of signal power to noise power in the additive white Gaussian noise channel. In a communication channel, we would like to have a high SNR. For the majority of our simulation, we set the SNR to 0(dB) to simulate a worst case scenario, where the power of the signal and the power of noise are equal. However, for a better understanding, we will show some of the effects of the SNR on few and multiple users.

First, we wanted to how SNR affected the bit error rate if the system had a small number of users (<40). Choosing 30 users for our simulation, we saw that SNR had a significant effect on bit error rate. As the SNR increases (e.x. from -5 to 5) we can see a close to quadratic relationship between the SNR and the number of errors.



We chose to include 150 users in our system to demonstrate how SNR affected bit error rate for a large number of uers. We observe that the SNR seems to have no effect on the bit error rate. Overall, the bit error rate is high for all SNR values. When there are many users, the SNR of the channel becomes less important because the waveforms transmitted by other users look like noise. Even with a high SNR, the channel will appear noisy because of the other users. Due to the randomness of the channel the BER varies by +/- 0.5%



Chapter 7

Conclusions and Applications[1]

Our CDMA simulation behaves as expected according to theory. While testing our simulation, we observed that bit error rate increases with more users in the system and decreases with a greater processing gain. We found that SNR had a significant effect on the bit error rate when there were few users in the system. With many users, the bit error rate was high regardless of SNR because the waveforms of the other users appear as noise in the channel.

We learned a lot while creating and testing our simulator. Through trial and error, we learned the importance of orthogonality in spreading sequences. The orthogonal property of long spreading sequences is only approximate, so increasing users reduces orthogonality and leads to more error. Orthogonality is important because it allows a particular receiver to disregard all waveforms except the particular one it wishes to decode, so reducing orthogonality means parts of other undesired waveforms are being read by the receiver.

Applications

This CDMA simulator is an educational tool which allows users to understand how bit error rate is affected by several different parameters. These parameters give the user multiple angles for variation and observation. The concepts and effects of variation of parameters can be observed on a simple level by a simple observation of cause and effect. Our hope is that users can use our simulation to gain a better understanding of the concepts behind CDMA.

[1] This content is available online at <http://cnx.org/content/m41831/1.1/>.

Chapter 8

Further Work[1]

The simulation leaves a lot of room for improvements and additions. For our model, we use a long sequence pseudorandom spreading sequence generated by maximum length shift registers. However, there are many possible short spreading sequence codes for CDMA, including Walsh codes, Gold codes, and Barker codes, which could be used instead of the MSLR. Since the goal of our simulation is to provide flexibility for users to change parameters, we could add short spreading sequence functionality and other spread codes to our simulation.

Since our goal is to create a user-friendly model, our next steps would be to create a graphical user interface for our simulation. Based on user inputs, we could have our simulation automatically generate plots of the bit error rate. We could also include settings in our simulation which take into account practical limitations. For example, real world communication channels do not have unlimited bandwidth, and the spreading factor will be limited by the bandwidth of the channel. This limitation would be automatically reflected as the user is inputting values for the various parameters.

In the future one aspect we hope to add to our simulation to improve understanding is the use of audio. If we can use an audio signal instead of a random code of numbers and play the reconstructed audio signal back to the user, then the user can better understand the real effect of the error rate on the original signal. The use of audio files would also allow us to better compare different bit error rates by applying the simulation to a known input, rather than just observing pseudo-random number sequences.

[1] This content is available online at <http://cnx.org/content/m41815/1.1/>.

Chapter 9

Bibliography[1]

Charan Langton. "Intuitive Guide to Principles of Communications". *CDMA Tutorial.* 2006.<http://www.complextoreal.com/CDMA.pdf

Griffiths, Alison, and Amr El-Helw. "Matlab, Simulink Building a Direct Sequence Spread Spectrum Model." *Scribd.* N.p., 21 July 2009. Web. 29 Nov. 2011. <http://www.scribd.com/doc/16628894/Matlab-Simulink-Building-a-Direct-Sequence-Spread-Spectrum-Model>.

Madhow, Upamanyu. *Fundamentals of Digital Communication.* New York: Cambridge University Press, 2008. Print.

Simo, Ernest. "CDMA Interactive ." *CDMA Online* . N.p., 2009. Web. 1 Dec. 2011. <http://www.cdmaonline.com/interactive04/workshops/terms1/1008.htm>.

[1]This content is available online at <http://cnx.org/content/m41796/1.1/>.

Chapter 10

The Team[1]

10.1 Team Members

Nonso Anyigbo (Lovett '13) is specializing in Systems and Signals.

Elaine Chung (Lovett '13) is specializing in Systems and Signals.

Alejandra Galvan (Lovett '13) is specializing in Computer Engineering.

Alison Hightman (Baker '13) is specializing in Systems and Signals.

10.2 Acknowlegements

Thank you to Behnaam Aazhang and Chris Hunter for assisting and advising us through this project.

[1]This content is available online at <http://cnx.org/content/m41805/1.1/>.

Attributions

Collection: *CDMA Simulation*
Edited by: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/col11388/1.1/
License: http://creativecommons.org/licenses/by/3.0/

Module: "Why does CDMA matter?"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41808/1.1/
Page: 1
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Our Project"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41801/1.1/
Page: 3
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "CDMA Background "
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41832/1.1/
Pages: 5-6
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Overview of Our Model"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41829/1.1/
Pages: 7-8
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Generating the Spreading Sequence"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41827/1.1/
Page: 9
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Results"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41804/1.1/
Pages: 11-13
Copyright: Alison Hightman, Elaine Chung, Alejandra Galvan
License: http://creativecommons.org/licenses/by/3.0/

Module: "Conclusions and Applications"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41831/1.1/
Page: 15

Module: "Further Work"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41815/1.1/
Page: 17

Module: "Bibliography"
By: Alison Hightman, Elaine Chung, Alejandra Galvan
URL: http://cnx.org/content/m41796/1.1/
Page: 19

Module: "The Team"
By: Elaine Chung, Alejandra Galvan, Alison Hightman
URL: http://cnx.org/content/m41805/1.1/
Page: 21

CDMA Simulation
A simulation of CDMA which allows users to understand how different parameters affect the error rate of transmitted information. This module describes the approach taken to design this simulation and an analysis of simulation results.