Computer Assignment 3

Assigned: Wednesday, May 10, 2006
Due: Wednesday, May 17, 2006

In this computer assignment, we will learn about Code-Division Multiple Access (CDMA) by using Matlab to look at some examples. The only operations you will need to perform are element-by-element multiplication (.* ) and array multiplication (*).

What to turn in:
• Source file for programs (.m files), answers to questions in parts 1 and 2.

Introduction:
CDMA puts all users desiring access to a trunked resource on the same RF channel at the same time. Cellular CDMA systems overlay their CDMA schemes on the FDMA (Frequency Division Multiple Access) plan. Users of a common frequency channel are separated from each other by superimposing a user-specific high-speed code on the modulation of each user (PN Sequences). Since the separating code has the effect of spreading the occupied bandwidth of each user's transmissions, the system is called a spread spectrum system.

We will generate two CDMA channels, and use a correlator (sum the result) to distinguish one channel, or signal, from the other. We will also show that without the correct high-speed code (also known as a key), to decode (or unlock) the common frequency channel and obtain data is difficult. We will consider two pairs of different length codes (PN sequences), one pair of length seven, and another of length 31. We will initially transmit only data from “User 1” and observe the results, and then transmit data from both “User 1” and “User 2”, superimpose them, and observe the results.

Encoding
We start with two different sets of binary information: data1 for User 1 and data2 for User 2. We wish to transmit the data stream for User 1 initially and subsequently both data streams together. Consider the following data sequences:
- data1 = [ 1 1 1 1 ];
- data2 = [-1 1 -1 -1 ];

Now, to give each data stream its own character, we need to multiply the elements of the information sequences in the previous step with a PN sequence. Consider the following sequences as the corresponding keys to the above information sequences:
- PN7_data1 = [-1 -1 1 1 1 1 1 ];
- PN7_data2 = [1 -1 1 1 1 1 -1 ];
- PN31_data1 = [-1 -1 -1 -1 1 1 1 1 1 1 -1 -1 1 1 1 1 -1 1 1 1 -1 -1 1 1 -1 1 1 ];
- PN31_data2 = [1 -1 1 -1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 ];

In order to spread the information sequence, we need to multiply each element in the information sequence with its corresponding PN sequence and create a signal in matrix form. To do this, simply take the transpose of your data sequence (the ‘{ }’ operator in Matlab) and matrix-multiply
this with the appropriate PN sequence \{\ast\}. For the length seven PN sequence, each signal should be a \(4 \times 7\) matrix and for the length 31 PN Sequence, each signal should be a \(4 \times 31\) matrix. Refer to these matrices as \(S_{1,7}\), \(S_{2,7}\), \(S_{1,31}\), \(S_{2,31}\) respectively (where \(S_{1,7}\) for example refers to the signal obtained by spreading the first data sequence by the PN sequence of length seven), so you may want to label your signals similarly for this assignment.

Transmission and Recovery

Part 1:
For this first part, suppose that only \(data_1\) is transmitted even though we've encoded everything already in the previous section.

We will first consider the signal with the PN sequence of length seven as an example. Hence, take the transmitted signal to be \(S_{1,7}\). To recover the original data, we need to first perform an element-by-element multiplication on each row of \(S_{1,7}\) by \(PN_{7,\text{data1}}\).

Sum up the rows of the previous step, and scale this by the length of the PN sequence (divide each sum by the length of the PN sequence).

1. Give the sequence you recover when you use \(PN_{7,\text{data1}}\) to recover data from \(S_{1,7}\).
2. Give the sequence you recover when you use \(PN_{7,\text{data2}}\) to recover data from \(S_{1,7}\).
3. Give the sequence you recover when you use \(PN_{31,\text{data1}}\) to recover data from \(S_{1,31}\).
4. Give the sequence you recover when you use \(PN_{31,\text{data2}}\) to recover data from \(S_{1,31}\).

Part 2
For this part, we transmit both data sequences by summing up the encoded signals we generated from the “Encoding” section. Hence, if we were to transmit the signal that was spread with a PN sequence of length 7, the transmitted signal would merely be \(S_{1,7} + S_{2,7}\). Let us recover the data sequences given the fact that both data sequences are transmitted.

5. Give the sequence you recover when you use \(PN_{7,\text{data1}}\) to recover data from \(S_{1,7} + S_{2,7}\).
6. Give the sequence you recover when you use \(PN_{7,\text{data2}}\) to recover data from \(S_{1,7} + S_{2,7}\).
7. Give the sequence you recover when you use \(PN_{31,\text{data1}}\) to recover data from \(S_{1,31} + S_{2,31}\).
8. Give the sequence you recover when you use \(PN_{31,\text{data2}}\) to recover data from \(S_{1,31} + S_{2,31}\).