Simple CDMA: Decode Secret Message

Sophie Jaro
The Cooper Union
April 1, 2020

***Abstract*—The purpose of this project was to decode a secret message that had been received from a simple CDMA system. Encoding the signal required framing the data, BPSK modulating, spreading using an 8-ary Hadamard transform, spreading and oversampling an m sequence, baseband filtering, upconverting and a raised cosine filter. Decoding required undoing these steps. The decoded secret message in this attempt was " a£ˆ EHx@ ˆ U] ¨ a".**

***Index Terms*—CDMA, Hadamard, m-sequence**

I. Introduction

Code Division Multiple Access is a channel access method that relies on spread spectrum technology and coding schemes to allow several transmitters to send information over one communication channel simultaneously. In direct sequence CDMA, the user signal is multiplied by a pseudo-noise sequence of high bandwidth. Common spreading sequences are M sequences and Walsh/Hadamard sequences. M or maximum length sequences are generated with a linear feedback shift register. The taps that generated the m sequence used to spread this signal were [8 7 6 1] corresponding to a generator polynomial of $G(X) = X^8 + X^7 + X^6 + X + 1$. Walsh/Hadamard spreading was also used to spread this signal. Walsh/Hadamard codes are perfectly orthogonal, making them optimal for avoiding user interference.

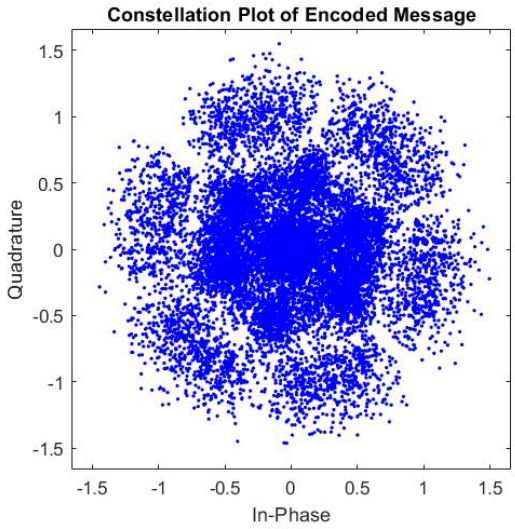
II. Methods

A. Filtering and Downsampling

The received "Rcvd" was filtered using the root raised cosine filter taps given. One copy of the filtered signal was then correlated with an upsampled version of the m sequence to attempt to find offset. Another copy of the filtered signal was immediately downsampled by 4. This downsampled copy was also used to attempt to find offset.

B. Generating M Sequence

The Fibonacci implementation of the m sequence generator was implemented by xoring (modulo 2 addition) the taps given and shifting. This was compared to an m sequence generated with the MATLAB Comms toolbox to ensure that a reasonable result was obtained. Several correlation methods were also used to check that the generated m sequence was valid but it cannot be confirmed that this is the correct m sequence for decoding this message.



lot of Encoded Message.jpg

Fig. 1. Constellation plot of the encoded message

C. Finding the Offset with Correlation

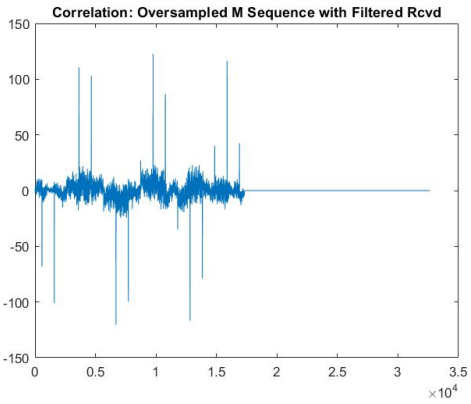
Two attempts were made to find the offset (which should have been 261). The first method was using the MATLAB xcorr() function to find the autocorrelation of the oversampled m sequence that was generated and the entire filtered received signal. The resulting plot featured its first impulse at about and x of 156. The other method attempted was finding the autocorrelation of the M sequence and one frame length of the filtered, downsampled received signal. The reasoning for examining only one frame is that the offset should not be longer than one pilot, which has length 255. The x value of the first impulse was interpreted as the offset. The offset found through that method was 144 so the M sequences was shifted by 144 before being applied to the signal.

D. Applying the Offset M Sequence

The apply the shifted M sequences to the signal, they were repeated by the number of frames of the downsampled received signal, which was that signal's length divided by the length of each frame which is 255. The repmatted m sequence was then element-wise multiplied to the processed recieved signal.

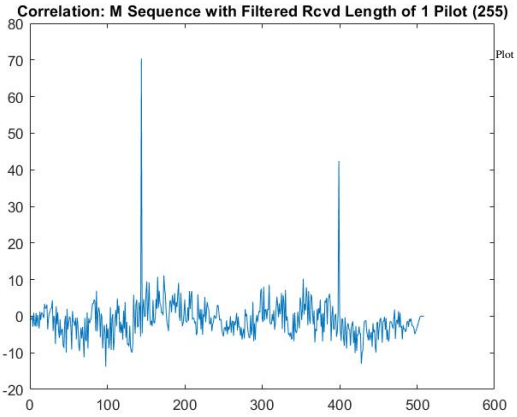
E. Adjusting for Frequency

The resulting constellation plot showed a similar arrangement of points as the original received signal. To continue



of Oversampled M Sequence and Filtered.jpg

Fig. 2. Correlation between the 4x oversampled m sequence and the filtered Rcvd signal. This shoul be used to find the offset.



of M Sequence and 1 Pilot.jpg

Fig. 3. Another attempt to find the offset. This is a plot of the correlation between the m sequence and one 255 length frame of the filtered, 4x downsampled Rcvd. The offset observed from this plot was 144, so the m sequence was shifted by 144 before it was applied to the signal

processing, they needed to be on the x-axis with no frequency component. This operation was completed in this step.

F. Apply 8-ary Hadamard Transform

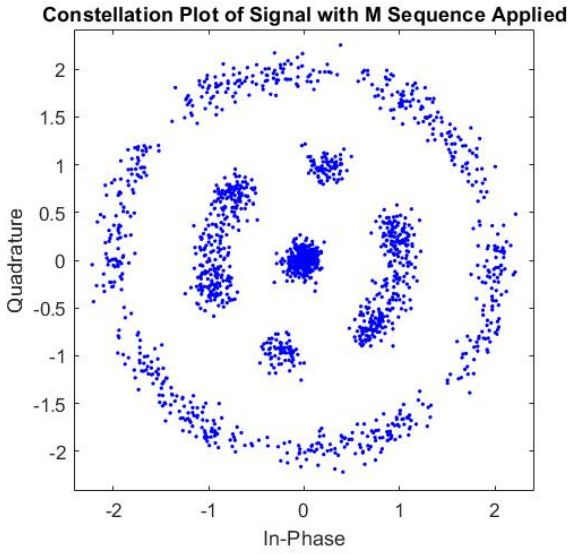
The 8-ary Hadamard matric was obtained with the MATLAB Hadamard function. The recieved signal vector was reshaped into an 8x510 matrix so that it could be multiplied by the Hadamard matrix. The 6th row (channel 5) was taken from the result of this multiplication. It was plotted, shifted to -1, 0, and 1, then demodulated.

G. BPSK Demodulate

In this step the data was demodulated manually using BPSK demodulation.

H. Decode to Characters

The resulting sequence of 0 and 1 was converted to characters with the char() function in MATLAB. The output of this step is the Secret Message.



Plot of Signal with M Applied.jpg

Fig. 4. After the received signal had been filtered and downsampled, the offset m sequence was applied. The result is shown here.

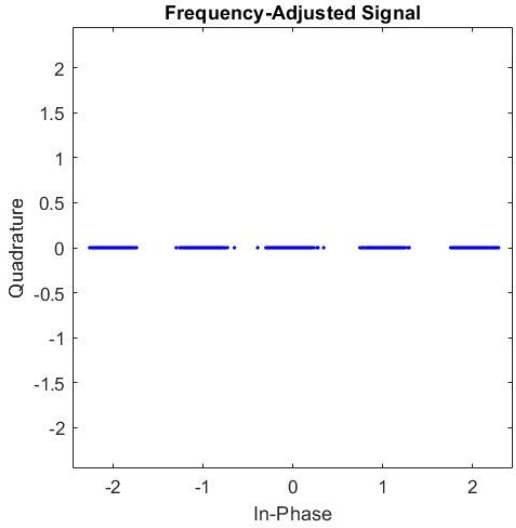


Fig. 5. Processing could not continue with the points offset from the x-axis. The offset was removed. This is the result. This signal was then modulated, Hadamard was applied, and BPSK was applied before the conversion to characters.

III. Results

Ultimately, a Secret Message of: $\hat{a}\hat{E}H \times x@\ddot{U}]\grave{a}\%Sx^{!}9z@`PBr\$`\ddot{A}9\ddot{O}\ \grave{A}9@\ddot{A}1mpd.f\grave{O}$

was obtained from the decoding attempt. This is obviously not the intended message. To obtain the intended Secret Message, I would spend more time checking that my m sequence matches that used to encode the message, using pilots to find the offset, and more carefully reshaping the data during the final decoding to characters step.

ACKNOWLEDGMENT

Gian Tria, Armaan Kohli, and Professor Hoerning answered many, many questions I had when I was lost in the process of this project.

REFERENCES

LFSR Reference:
$http://in.ncu.edu.tw/ncume_ee/digilogi/prbs.html$