Simple CDMA

Gian Angelo Tria
ECE 404:Wireless Communications
The Cooper Union
New York City, USA
gianangelotria@gmail.com

***Abstract*—The objective was to implement a CDMA uplink that would be able to decode a secret message on MATLAB. The input data stream was a string of 8-bit ASCII characters encoded MSB first in a serial data stream. Then each bit was spread on Walsh channel 5 using an 8-ary Hadamard transform. An 8th order PN sequence was then used for spreading. This was then filtered and upsampled. In order to decode the secret message, each of these steps had to be undone starting with the up sampling and ending with converting the bits into characters.**

***Index Terms*—CDMA**

I. Sampling and Filtering

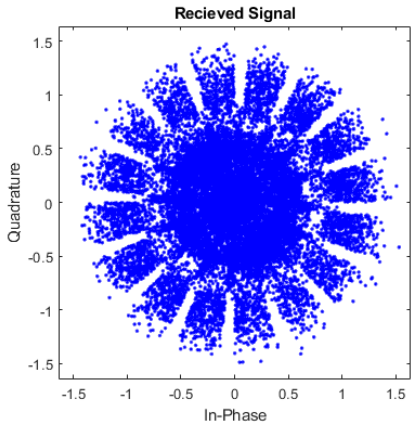

Fig. 1. Received Signal

The starting signal is shown in Figure 1. First, filtering was done to the signal using a Root Raised Cosine Filter. The specific filter used was given and implemented by using the filter function on MATLAB. The results of this filtering is shown in Figure 2

Because the signal went through an upconversion before being sent out, it must be downsampled first in order to be decoded. The signal was downsampled by a factor of 4, the same rate as the upconversion.The result can be seen in Figure 3, and it is clear that the number of samples has gone down.

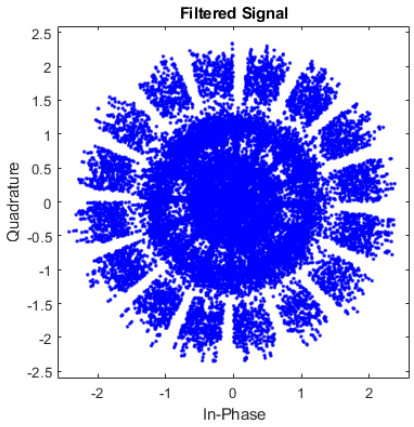

Fig. 2. Root Raise Cosine Filtered Signal

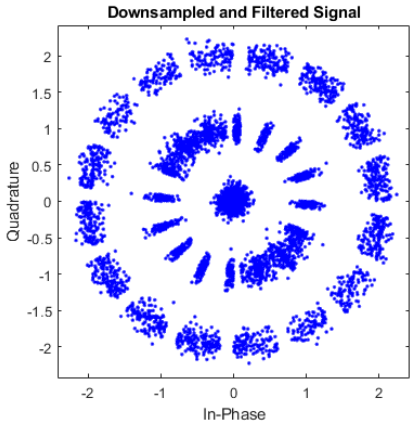

Fig. 3. Received Signal

II. PN Sequence Generation

Once the downsampling and filtering is done, the next step is to generate a PN sequence. The PN sequence generated must be the same as that used during transmission, and this was done using the generator polynomial given by the taps. The order of this generator polynomial was 8, and the sequence was generated by replacing the correct indexes with the correct XOR outputs. The generator type used was Galois although Fibonacci produced the same output when tested.

III. PN Sequence Application

Although the m sequence has been verified to be correct, it still may not be in the right orientation to be applied to the filtered and demodulated signal. Then, the correlation was checked between the m sequence and the filtered and demodulated signal. The correct shape of this should be noise at all places that are not modulo 256, and so the graph of Figure 4 is offset by a little. By checking the index of the impulse, the received signal was shifted by the correct amount.

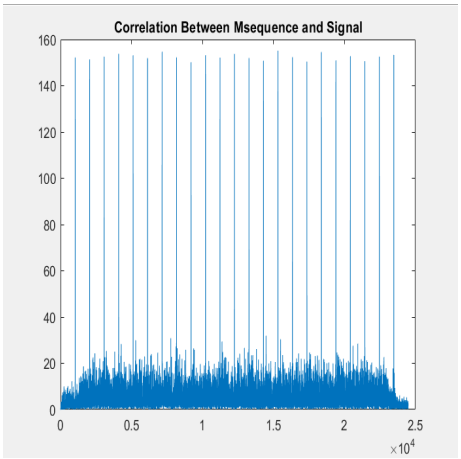

Fig. 4. Cross Correlation between the M Sequence and the Signal

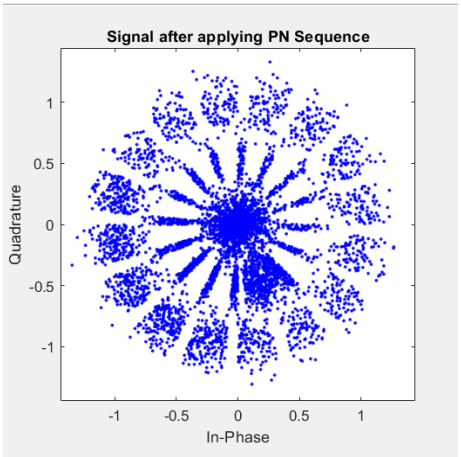

Fig. 5. Signal After Applying the PN Sequence

IV. Frequency and Phase Shifts

Assuming that the signal did not have frequency and phase shifts of more than pi radians,the signal seen in Figure 5 was converted into purely real values by multiplying each point by a phasor with the negative phase of its phase offset. The results of this can be seen in Figure 6.

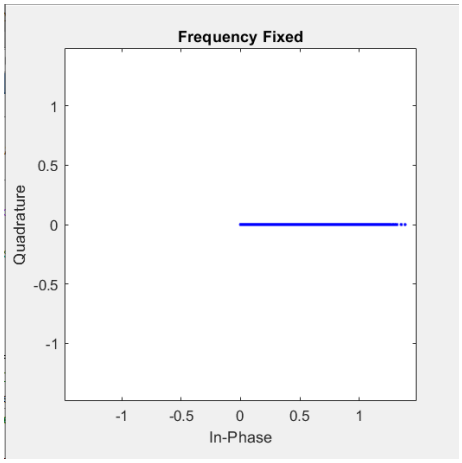

Fig. 6. Signal converted to all real values with the same magnitude

V. Walsh Codes and Demodulation

The Walsh Codes were made using the hadamard MATLAB function, and in order to multiply the signal matrix with the Hadamard matrix, the signal matrix had to be reshaped.This can be seen in Figure 7. Once this was done, each column corresponded to a channel of the Walsh codes, so in order to get the output signal, the correct channel and column was chosen and BPSK demodulated.

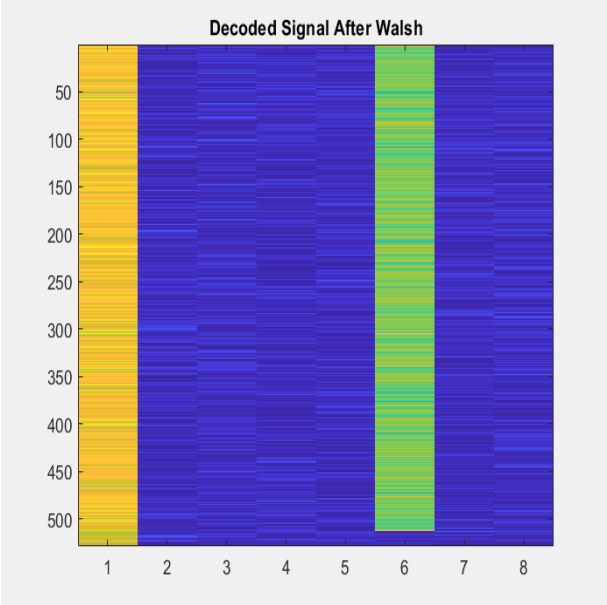

Fig. 7. Decoded Signal After Walsh

VI. Message and Conclusion

Finally the demodulated signal was separated into 8 bit parts. Each group of 8 bits was converted into a decimal before being converted into a character. The characters are correct and say:

```
decoded_message =

    'Your mother was a hamster and your father smelt of elderberries./Ÿ'
```

Fig. 8. Output Characters

ACKNOWLEDGMENTS

I would like to thank Armaan Kohli, Karol Wadolowski, Kevin Jiang and Sophie Jaro for reexplaining some of the concepts to me. They helped me figure out each next step and verified that each step taken was on the right track.