Binary PAM Communication

David Harper Armstrong, *Member*, Kaan Dincer, *Member*

***Abstract*—Provide a short, 5-6 line abstract.**
Communication systems today operate at very high-frequency ranges. Basic signals are converted into these higher frequency bands by upconverting their information. This process allows users to send and receive multiple messages at once as long as each message stores its data in separate frequency bands (Ex: channels on a TV, stations on the radio). One implementation of this process is the binary pulse-amplitude modulation, which we simulate in this case study. In the process of completing this project, we compared various pulse shapes to measure their performance of accurately transferring data, and finally, we use our simulator to send multiple messages at once using three separate frequency bands.

I. Background

Communication systems are widely used in our day to transfer messages and information between people, systems, or computers. In every communication system, the information or message is transformed into a form usable for the communication system, and then the signal is recovered through processing at the receiver because every communication system has a specific frequency range in which it works best. Modulation is the process of embedding an information-carrying signal into another signal. Modulation not only allows to transmission of signals that carry information, but it also allows multiple information-carrying signals with overlapping spectra to be transmitted at the same time. One of the ways modulation is used in communication systems is amplitude modulation, in which a signal is used to modulate the amplitude of another signal. In this case study, we used this technique, pulse amplitude modulation, to communicate certain texts by converting those into binary messages and then to signals to be transmitted through our system.

II. Methods

A. Pulse Creation

In our case study, we use three different pulse shapes: the sinc function, a rectangle, and a normal curve. The sinc function is 1x101, the square is 1x10, and the normal curve is 1x101.

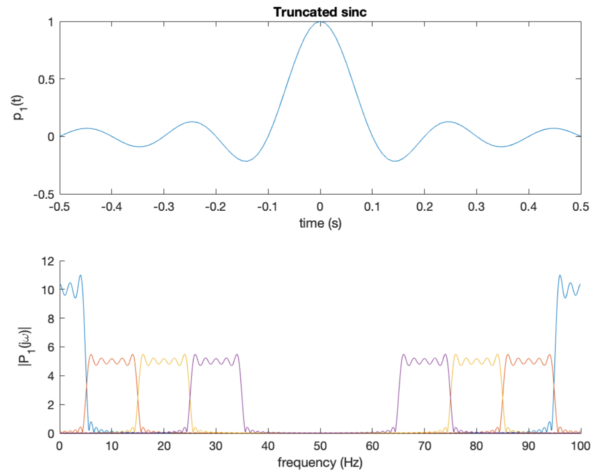

Fig. 1. Plot of sinc pulse shape in the time & frequency domain.

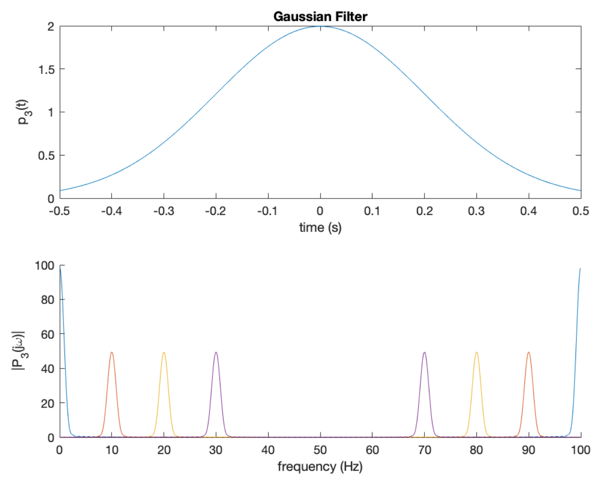

Fig. 2. Plot of Gaussian pulse shape in the time & frequency domain.

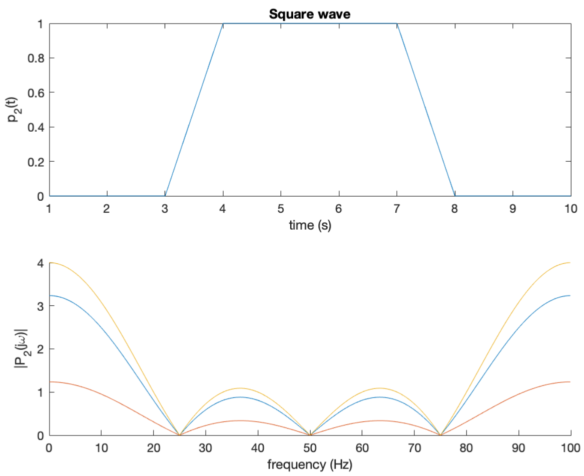

Fig. 3. Plot of square wave pulse shape in the time & frequency domain

As we can see in the plots of the frequency domain of each pulse shape, the gaussian filter and the truncated sinc have a bandwidth of less than 10 Hz which satisfies the Nyquist filtering criterion. The lack of overlap in adjacent bands allows messages to be sent simultaneously without interference. However, the square wave has a very large bandwidth so the upconverted versions overlap and would be a worse fit than the other pulses for the PAM simulator.

B. *Binary PAM Simulation*

The text message is converted into a binary message using the provided functions in the following way: `str2num(reshape(dec2bin(string message)',1,[])')`, so it converts a string into a vector of 1s and 0s. Then, the 0s in the vector of 1s and 0s are converted in –1 because our matched filter is assigned +1 or –1 depending if the index is positive or negative, and when we want to convolve +1s and –1s because it is harder to see the effect of the 0s on the signal and it is harder for the system to detect and visualize the 0s. Then this binary message with +1 and -1s is embedded into a new vector: the new vector takes in each index of the binary message vector and spaces it out by the symbol rate. This spaced-out vector is then convolved with our pulse shape to create the signal to be transmitted through our system.

C. *Up and Down Converting the Signals*

Up-converting (modulate) the signal is done by multiplying the signal by a cosine function at a specific frequency. Down-converting the signal is done by multiplying the up-converted signal with the same cosine function at the same frequency. The down-converted signals were also passed through a lowpass filter to get the original signal back. The frequencies 20 Hz, 30 Hz, and 40 Hz were chosen for the cosine functions multiplied by signal 1, signal 2 and signal 3 respectively. These frequencies were determined as such because the Nyquist filtering criteria requires a bandwidth of $1/Ts = 1/0.1 = 10$. So, the frequencies of each cosine multiplied by each signal were spaced by 10 Hz.

D. *Adding Noise*

Noise is created by multiplying a variable sigma, with $0 \leq$sigma ≤ 1, with a vector of random numbers that has the size of the biggest vector out of the three signals used. This noise vector is then added to the combination of the up-converted signals.

E. *Creation of Matched Filter*

First, a z vector was created through the convolution of the down-converted, filtered signal and the pulse shape. Then the matched filter was created by assigning +1 to the specific index of the matched filter if the index at each multiple of the symbol period of the z vector was positive and –1 if the index was negative.

F. *Error Rate Calculations*

The whole binary message vector was looped through and if a value at a certain index of the binary vector wasn't equal to the matched filter vector then total errors were increased by one. After it was looped through the whole binary message vector the total errors were divided by the number of elements checked (number of elements in binary message vector), which gave us the error rate.

III. Results

A. *Optimal Pulse Shape and Frequency Conversion*

The three pulse shapes used (sinc, square, and normal curve) each give different levels of accuracy for the final message received. We also experimented with choosing different frequencies to upconvert our signals to. The results of these two factors, combined with variable added noise, can be seen in the following figures.

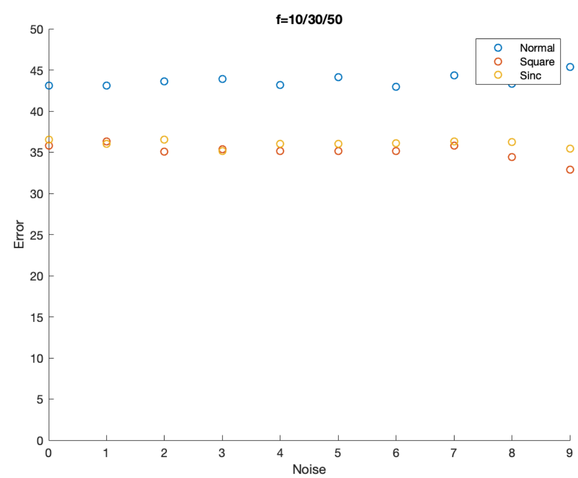

Fig. 4. Plot comparing the performance of the three pulse shapes with variable noise and target frequencies set at 10, 30, and 50 Hz for three binary messages.

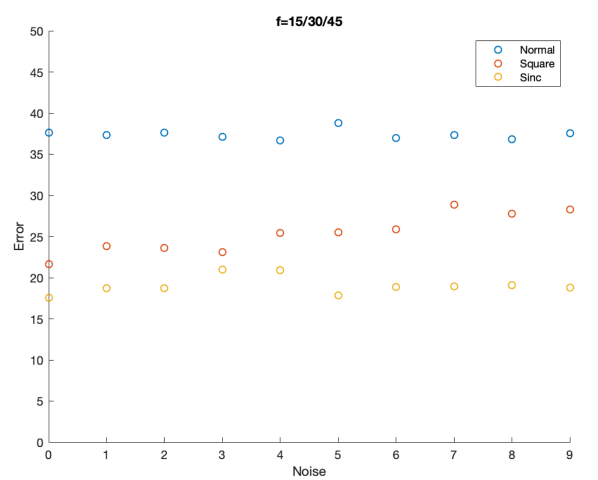

Fig. 5. Plot comparing the performance of the three pulse shapes with variable noise and target frequencies set at 15, 30, and 45 Hz for three binary messages.

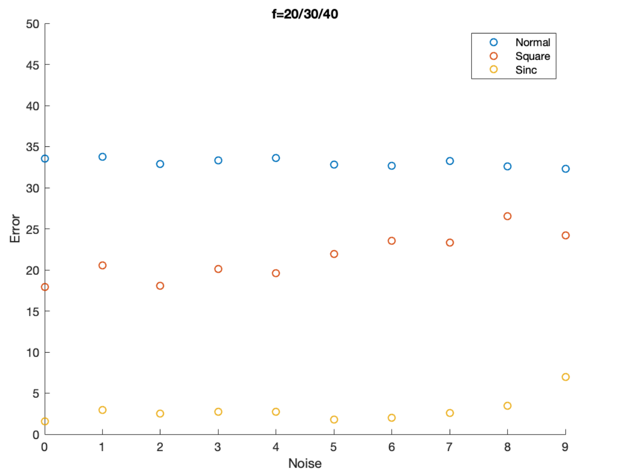

Fig. 6. Plot comparing the performance of the three pulse shapes with variable noise and target frequencies set at 20, 30, and 40 Hz for three binary messages.

These graphs portray how noise level, bandwidth, and pulse shape affect the error rate. It was determined from these graphs that increasing the noise increased the error rate. These graphs also show us that the optimal combination of pulse shape and up-conversion frequencies out of the three options we tested is the sinc function with the three messages being sent at 20, 30, and 40 Hz, respectively. We tried spreading out the frequencies to decrease overlap but it turns out that since the Nyquist criteria only requires a bandwidth of $\frac{1}{T_s} = 10$ Hz, spacing out the up-conversions by that exact amount gives us the best results.

B. *Sending 3 "Text Messages"*

After determining the optimal settings for our simulator (sinc pulse with target frequencies of 20, 30, and 40), we transmit three separate strings through our code at once and then retrieve them at the end. Using noise at full strength, our receiver interprets

"I love ESE 351! :)" as "I ,ove ESEa#51! 2)"

"Trobaugh = goat !" as "Trobaegh - goat " and lastly,

"Systems > Electrical!" as " ystems > Elactrkc =h" with an average error rate of about 4%.

C. *Random bits*

Figure 7 below shows, in order, the noise-free PAM signal: y(t), up-converted signal: $y(t)\cos(w_c t)$, noisy received signal: r(t), and the down converted signal passed through the lowpass filter: $[r(t)\cos(w_c t)]*h_{LPF}(t)$.

Fig. 7. Plot of the graphing of the non-modulated, up-converted, and down-converted PAM signal, and of the combination of the 3 signals with addition of noise for the sinc function.

The process of the PAM simulation can be easier analyzed in the frequency domain, as seen below. The figure below shows our three signals in their up-converted form, as well as the combined version of all three with the gaussian noise added.

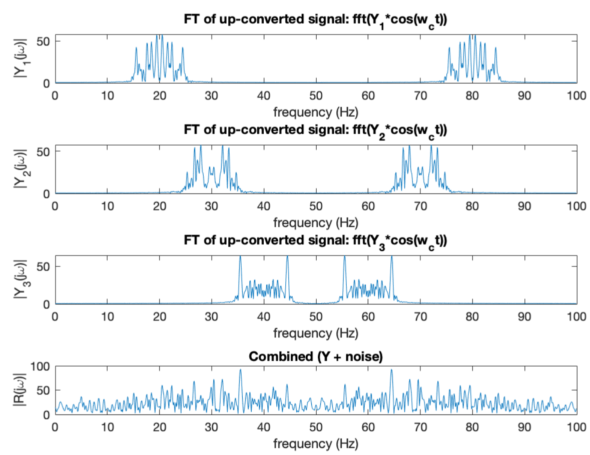

Fig. 8. Plot of upconverted signals at frequencies of 20, 30, and 40 Hz, as well as the combined signal of each Y and the gaussian noise. These vectors are shown in the frequency domain to best show the bands being used.

As seen below, a random message of bits can be precisely sent through our PAM simulator alongside two other messages and then it will be transformed back into the original message by the matched filter.

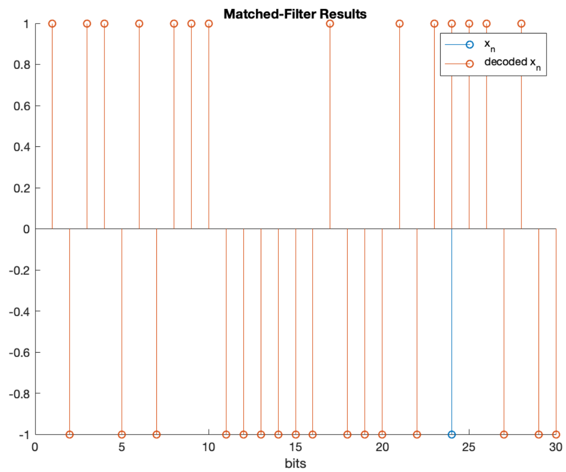

Fig. 9. Stem plot of our sent binary message x_n and the decoded message $\hat{x}_n$.

The small amount of error can be seen in the sole blue tick in the bottom right of the graph.

IV. Conclusions

In conclusion, we have demonstrated our results of a successful PAM simulator that can input multiple messages, up-convert and combine them with additive noise, and then use a match-based filter to decipher the original messages after down-conversion and a lowpass filter. Through performance analysis, we found that the optimal pulse shape for this task is the truncated sinc function and the optimal up-conversion frequency targets are 20, 30, and 40 Hz.

References

[1] A. W. Openheim, A. S. Willsky, and S. H. Nawab, “Signals and Systems” Second Edition, Prentice-Hall, Boston, MA, USA, 1996