Analog Modulation Simulation

Brian Doan

Prof. Frost

October 12, 2020

Table of contents

Introduction

This project explores the effects of noise on different analog modulation schemes including frequency modulation (FM), phase modulation (PM), conventional amplitude modulation and single side band amplitude modulation (SSB AM) through the use of Matlab. The script takes in a short audio clip and evaluates the experiential performance of the frequency and amplitude modulation schemes on noise. The main script applies Matlab implementations of the modulation schemes to return modulated versions and then to implementations of demodulation to return the demodulated signal. The noise is additive white gaussian noise (AWGM) at three different variances. The script calculates the signal to noise ratio across these three variances for all the schemes. Finally, the script plots the output of the scheme modulate and demodulate implementations in the frequency domain for analysis.

Phase Modulation is a type of angle modulation. The phase of the carrier signal varies while the frequency and amplitude stay constant. Theoretically, this scheme is easy to implement and is better suited for noise. This scheme however, requires two signals with a phase variation. Frequency modulation varies the carrier frequency in relation to the frequency of the receiving signal. FM is theoretically better suited for noise and has less power consumption than AM. However, FM is expensive to build and has a larger bandwidth. The upper single side band modulation was used for the simulation. It is a derivative of amplitude modulation and has a lower bandwidth than the other schemes. Also the scheme is better suited for noise. The implementation is complex Amplitude modulation Amplitude modulation is simple to implement and can work as an RC circuit. The scheme is also very cheap. However, the less complex scheme is more sensitive to noise.

Results

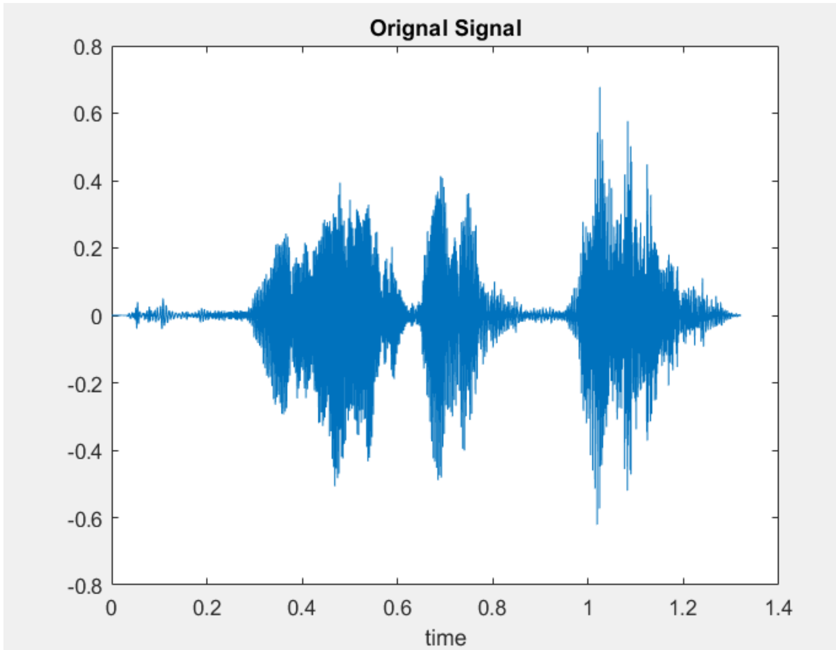


Figure 1: Original Signal from audioread

The plot in Figure 1 is the original signal in the time domain. The input is approximately a one second clip in Spanish. The full clip was six seconds long. However, while simulating the later sessions the size of the vectors were too large so it was shortened. Immediately after the audioread function execution, the output m(t) is transposed for computational convenience. Then, the second output of the audioread was reused to replay the sound with soundsc and to calculate the cutoff frequency.

Single Side Band without Noise

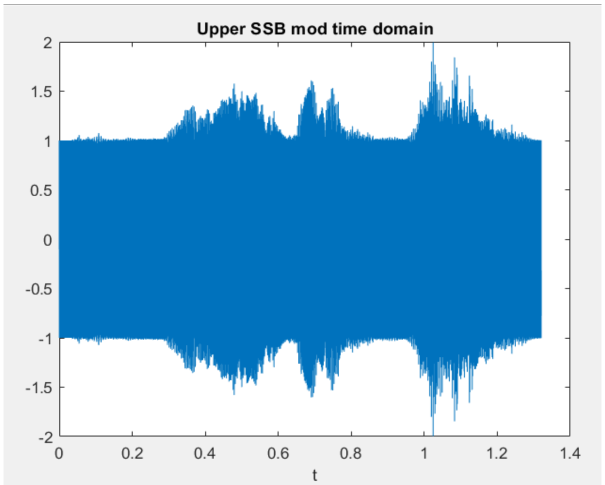


Figure 2

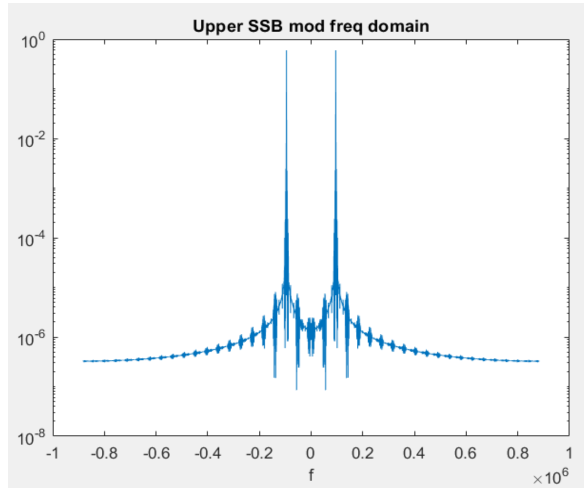


Figure 3

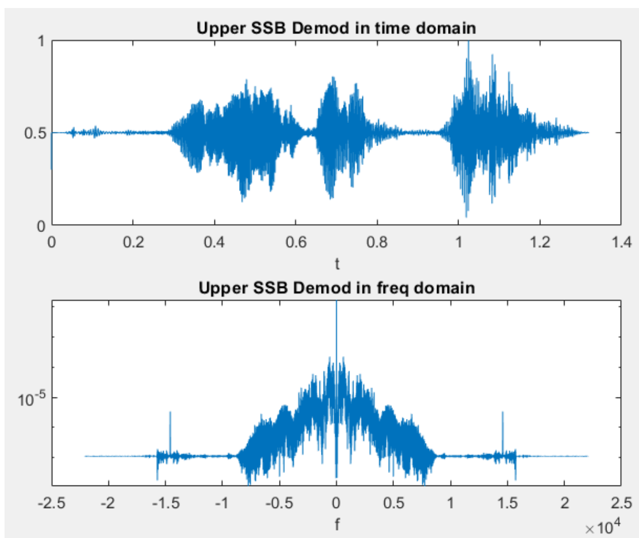


Figure 4

The SSB modulation scheme design consisted of an upscaled sampling frequency and an upscaled time. The upsample factor, L, was chosen as 40. The original signal was scaled by the max amplitude of the original signal then linearly interpolated. The modulation scheme design takes in a signal with modulation index A_c which in that case was 1. The signal power at the output did not require a change in A_c. Then, the real component of the Hilbert transform was taken at the upsampled frequency in order to apply the upper single sideband equation. Finally, a pilot tone was added at the same cutoff frequency. The carrier frequency as, W_c, was 600000 Hz. Figure 1 shows the output of this stage of the scheme in the time domain while Figure 2 shows the output in the frequency domain. The demodulate stage of the scheme multiplies the input signal by the pilot tone then lowpasses at the cutoff frequency. Then, the output is downsampled by L.

Conventional Amplitude without Noise

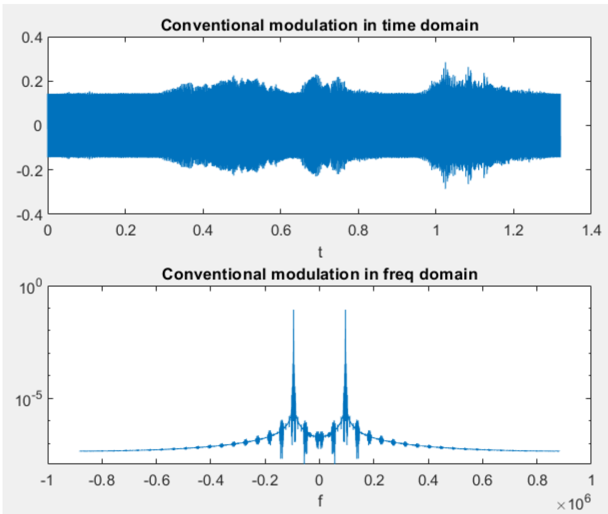


Figure 5

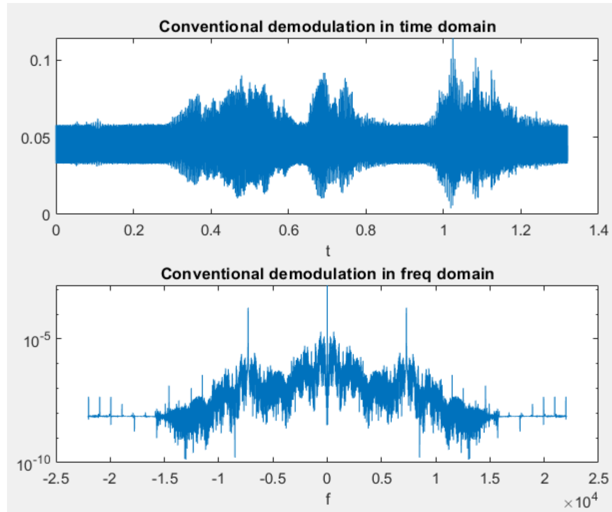


Figure 6

The conventional amplitude modulation scheme first starts with upsampling the time and normalizing the original signal. Then the normalized signal is linearly interpolated. Finally, the output of the stage is the product of the modulation index by the carrier and by one plus the interpolated signal. In the demodulate stage the input is rectified much like in a RC circuit. Then the rectified signal is filtered with a lowpass filter. The output of this stage is the downsample of the lowpassed signal.

Phase Modulation without Noise

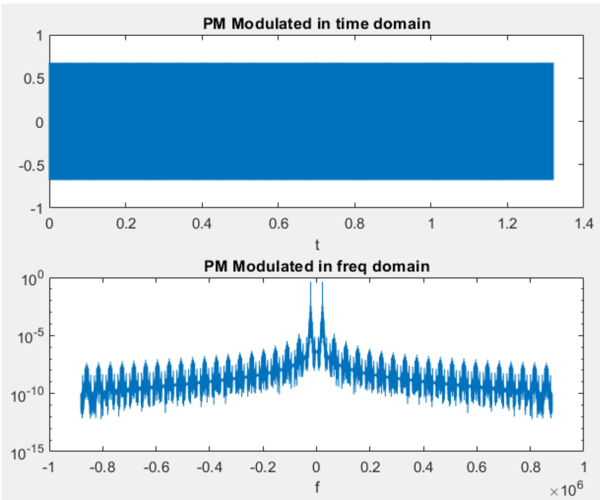


Figure 7

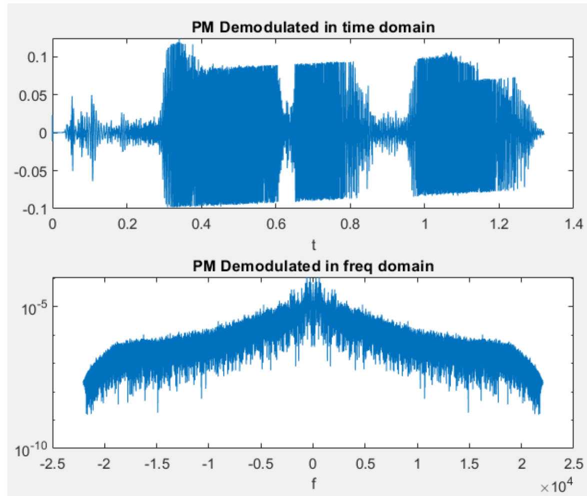
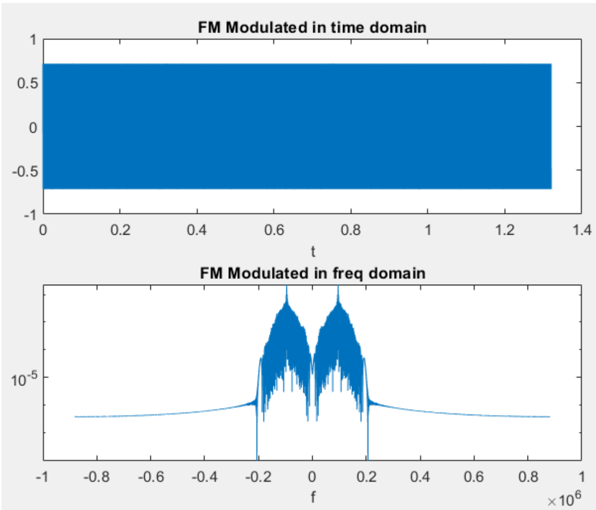


Figure 8

The Phase Modulation (PM) scheme design comprises the modulation and demodulation stages. In the modulation stage, the signal is scaled by the max amplitude and while the time is upsampled. Then the signal is interpolated linearly. The value for the constant that controls frequency deviation, k, was 2 for Figures 7 and 8. The k had to be a low value compared to that of the FM scheme. Finally, the output of the interpolation is the product of the modulation index by the phase shift of the carrier by the interpolated output times k. Figure 7 is that output of the modulation stage in the time and frequency domains. This output was then fed into the demodulation stage which was implemented by IQ decomposition. A lowpass filter was used instead of the default Matlab function. The components of the input signal are determined by a sin and cos which are then filtered. The arctan of the ratio is taken between the two and scaled by the same k. Finally, the quotient is decimated.

The box like plot is just a varying phase. The carrier frequency is very large, initially given as 600000 Hz, so the block is a high frequency cosine.

Frequency Modulation without Noise



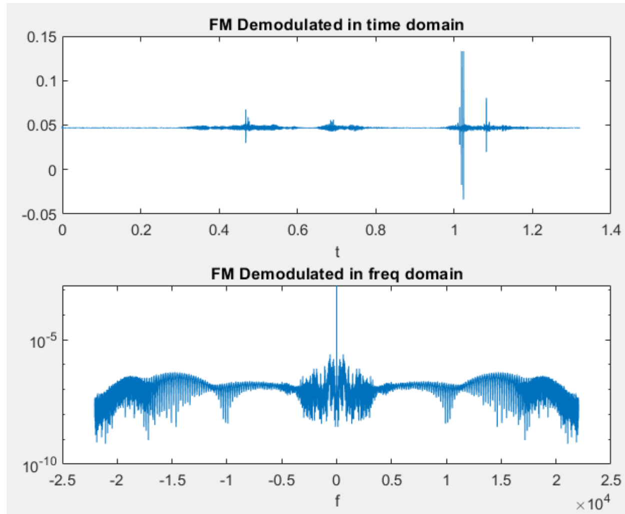


Figure 9

Figure 10

The frequency modulation scheme calculates the phase of the interpolated signal. The phase is the product between the frequency deviation constant k and the integral of the linearly interpolated input signal, all scaled by the upscaled sampling frequency. The k value for this scheme is much higher compared to that of the phase modulation scheme. For the figures above 100000 was the k value. The output of the modulation stage is the product between the modulation index and the phase shifted carrier signal. The demodulation stage of this scheme passes the input signal through a jw block. The real part of the block output is then lowpassed and decimated. The output of the modulated divided by W_c is the input of the demodulated. The division is the result of a comparison between AM and FM where one plus the received signal is different then the W_c plus the received signal. This made the signal power at the output more reasonable.

The box like plot is just a varying phase. The carrier frequency is very large, initially given as 600000 Hz, so the block is a high frequency cosine.

Single Side Band with Noise

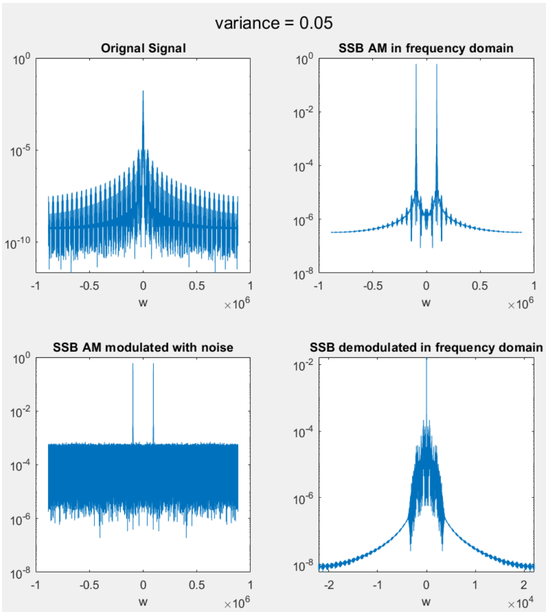


Figure 11

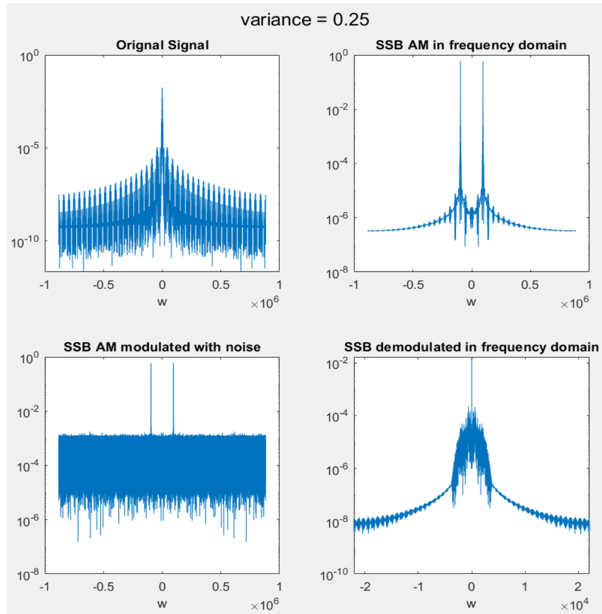


Figure 12

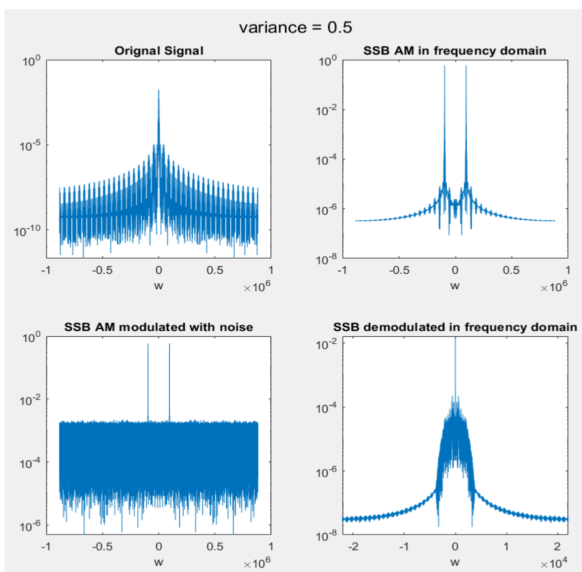


Figure 13

This scheme with the added white noise performed very well at the soundsc output. The noise was noticeable but was not close to overwhelming the recognizable original signal. The same input values were used for this scheme when there was not any noise.

Conventional Amplitude Modulation with Noise

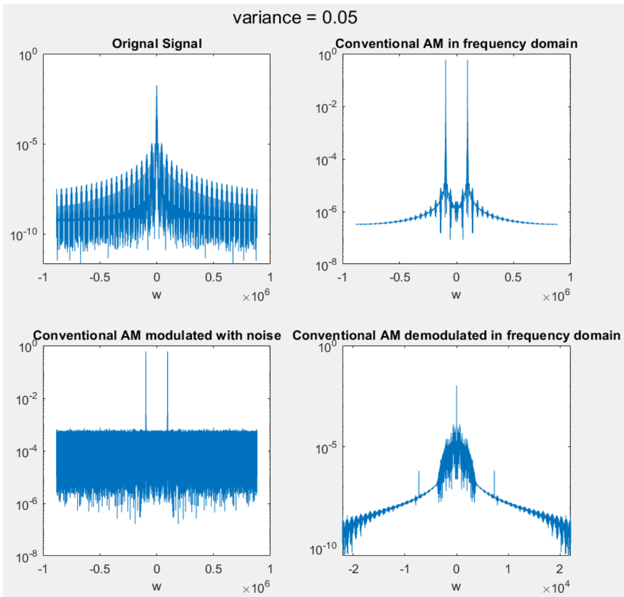


Figure 14

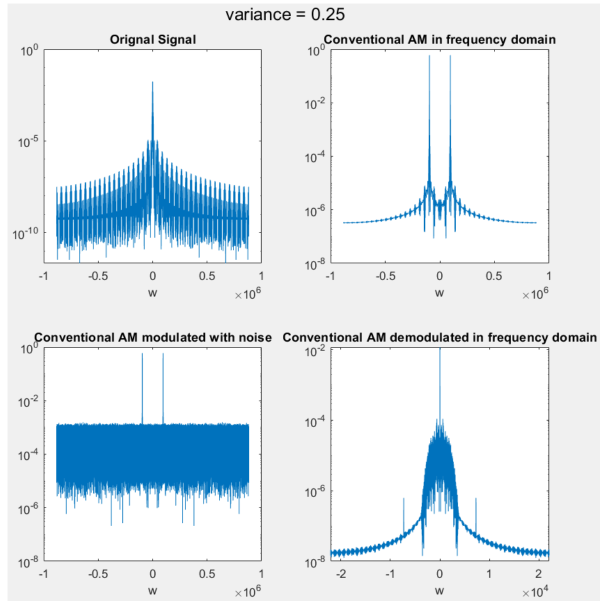


Figure 15

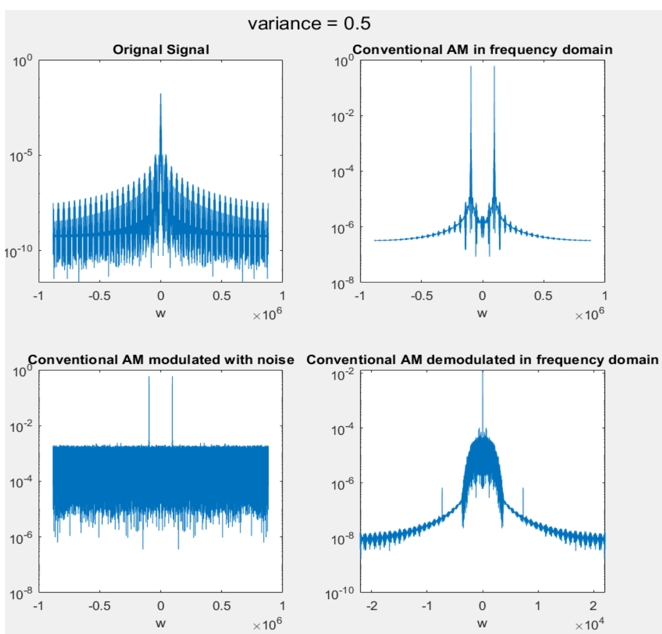


Figure 16

This scheme with the added white noise performed pretty well at the soundsc output. The noise was noticeable but was not close to overwhelming the recognizable original signal. The same input values were used for this scheme when there was not any noise.

Phase Modulation with Noise

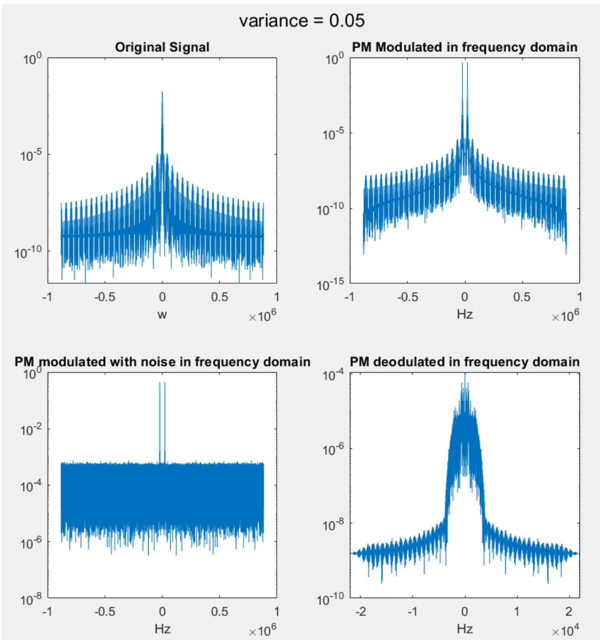


Figure 17

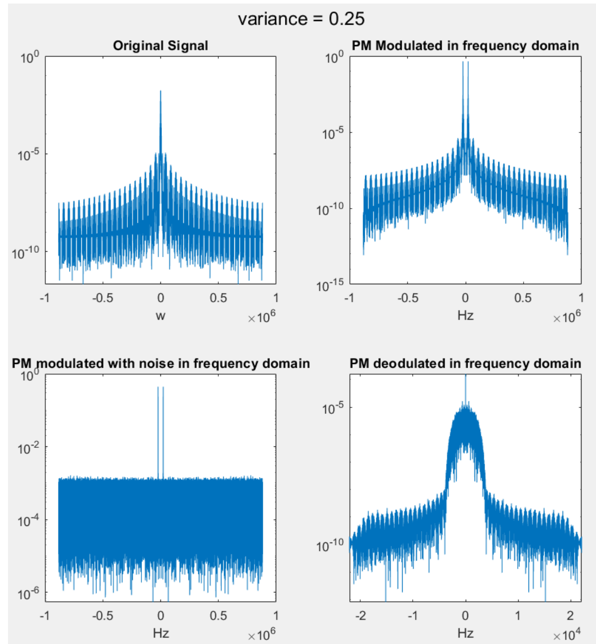


Figure 18

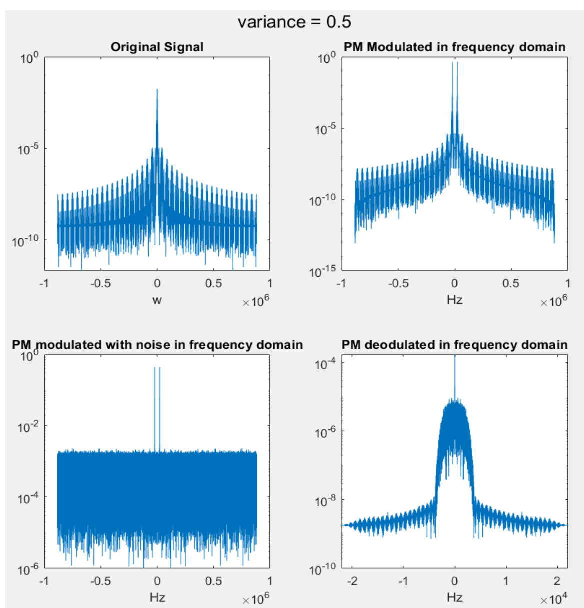


Figure 19

Of all the schemes, PM performed the best at the soundsc output after adding white gaussian noise. The same input values were used for this scheme when there was not any noise.

Frequency Modulation with Noise

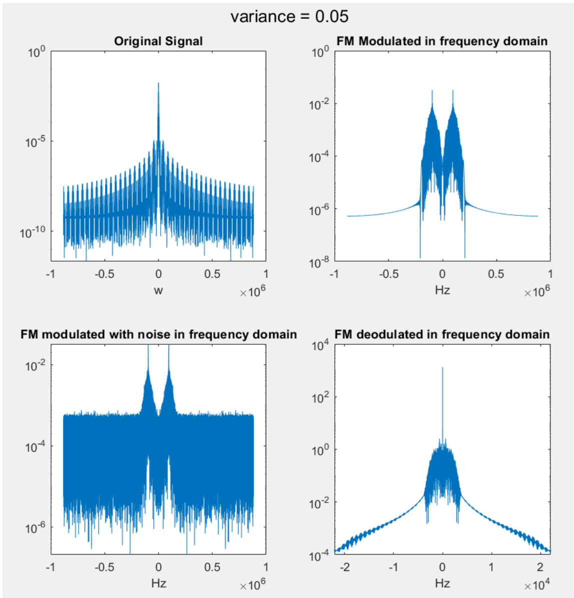


Figure 20

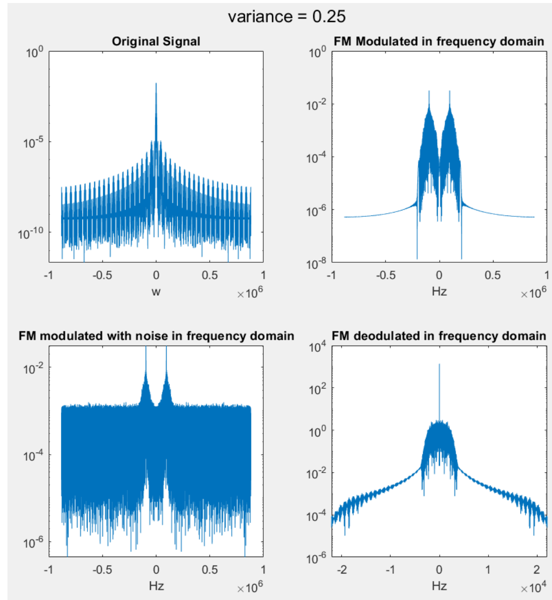


Figure 21

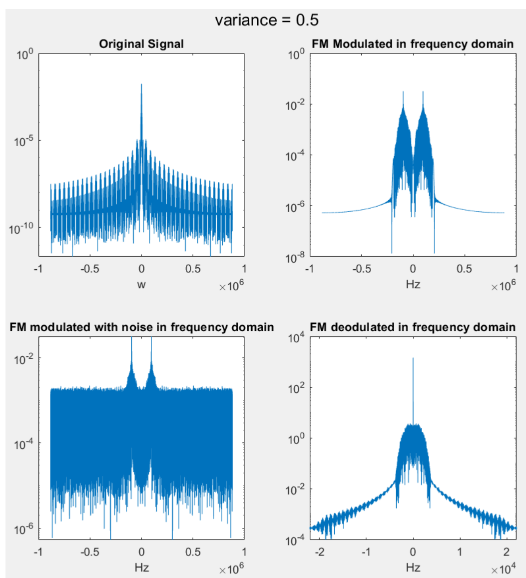


Figure 22

Unfortunately, the output of the FM from the soundsc does not sound very great. With noise, theoretically FM should sound the best but the output of the soundsc appears overwhelmed by the noise. The program generates noise based on the input variance vector.

Conventional Amplitude Modulation with varying Modulation Index

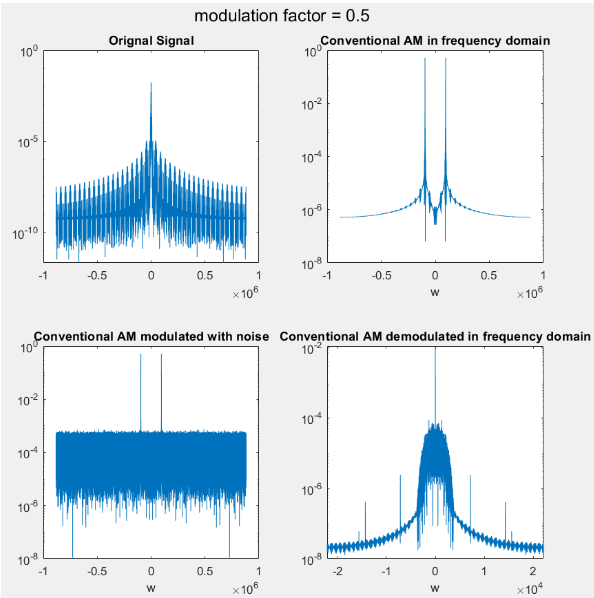


Figure 23

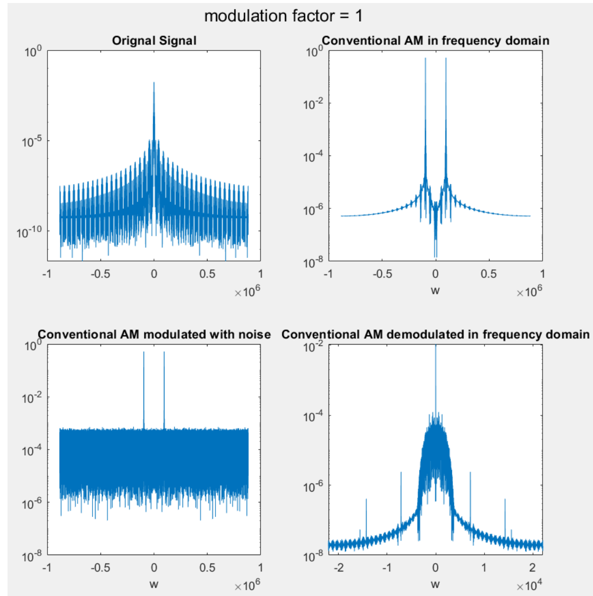


Figure 24

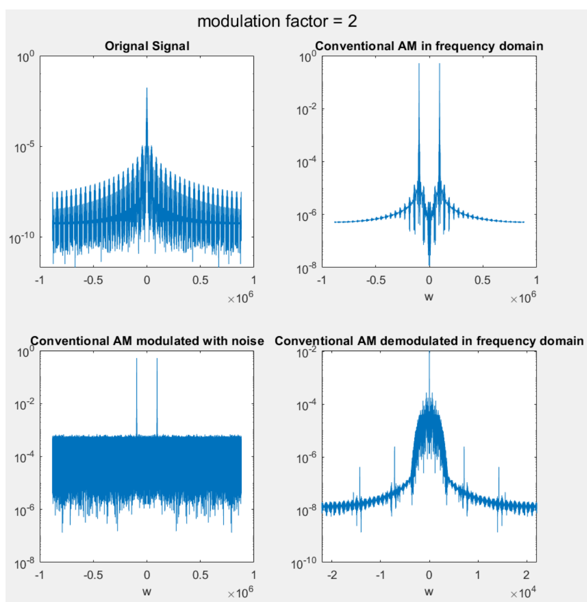


Figure 25

The same input values were used for this scheme as previously done. The difference here is that the max amplitude of the signal was fed in with a modulation index factor scale.

Phase Modulation with varying Modulation Index

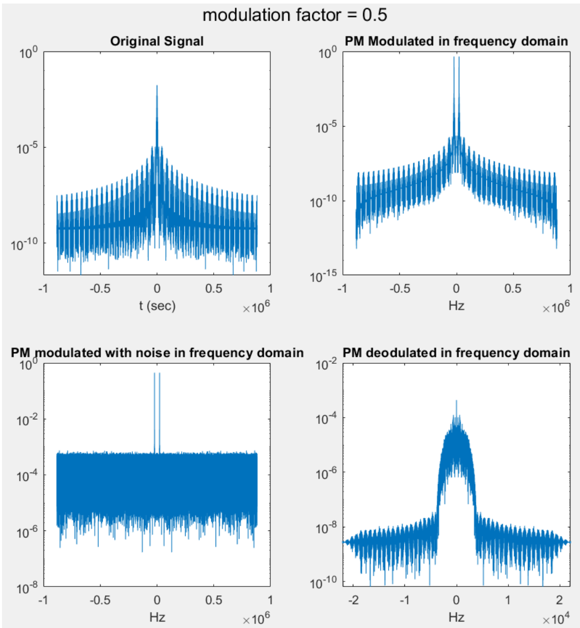


Figure 26

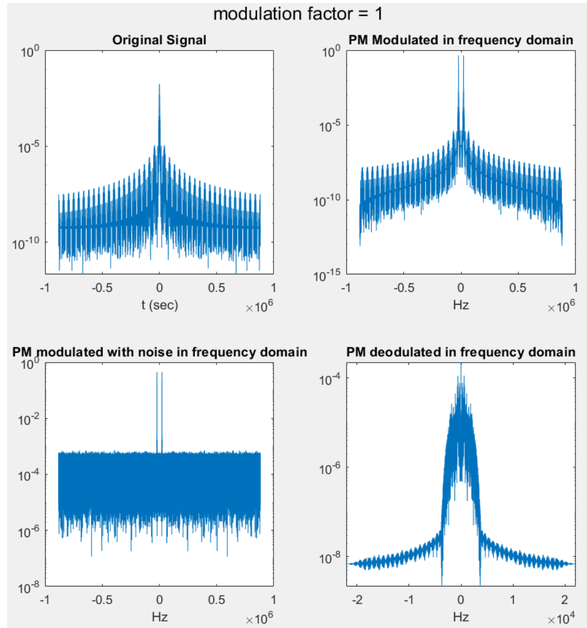


Figure 27

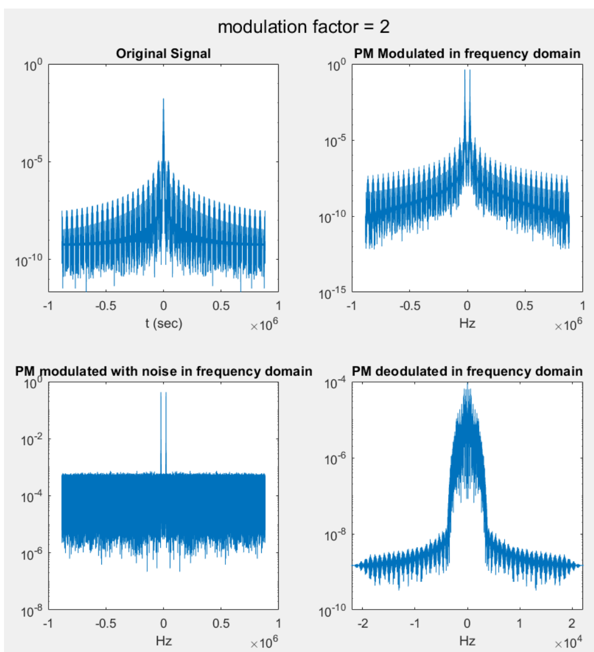


Figure 28

The same input values were used for this scheme as previously done. The difference here is that the k input was fed in with a modulation index factor scale since k is proportional to the modulation index. In the FM scheme, as shown below, the k was also scaled similarly.

Frequency Modulation with varying Modulation Index

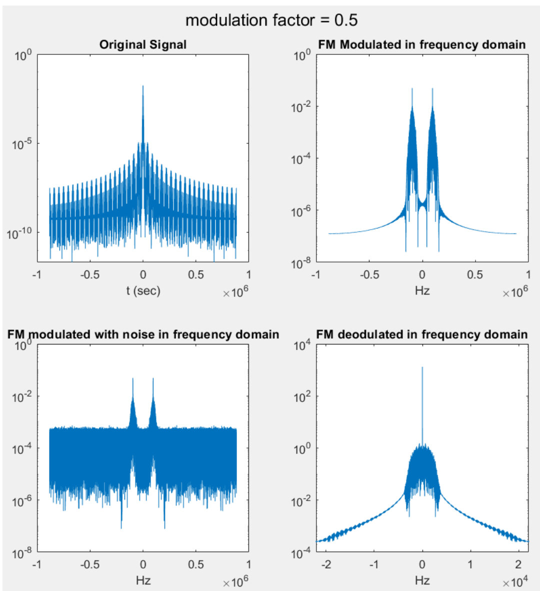


Figure 29

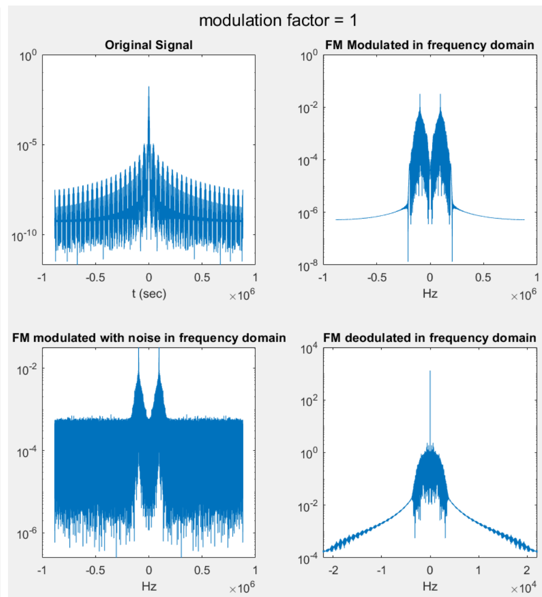


Figure 30

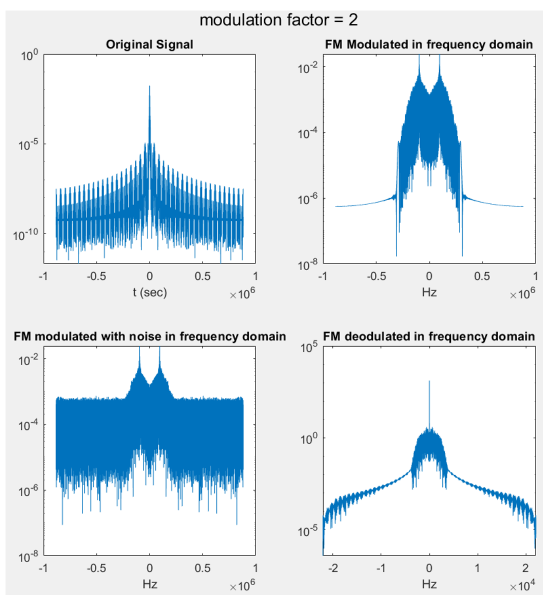


Figure 31

Signal Powers

Signal Powers	
Original Signal	0.010349
SSB	0.002556
Conventional AM	0.0021922
PM	0.0022691
FM	0.0022011

Table 1: Power of signals after each scheme

Signal to Noise Ratios

Experimental				
Variance	USSB AM	Conventional AM	PM	FM
0.05	28.9821	7.4555	3.9351	28.8515
0.25	23.9759	6.9806	0.67995	22.1177
0.5	21.3004	5.9026	0.17844	17.4173

Table 2: Experimental SNR Values

Theoretical				
Variance	USSB AM	Conventional AM	PM	FM
0.05	-35.9244	-38.9998	-57.361	61.9119
0.25	-42.9141	-45.9895	-64.3507	54.9222
0.5	-45.9244	-48.9998	-67.361	51.9119

Table 3: Theoretical SNR Values

Experimental (Variance = 0.05)			
Index Factor	Conventional AM	PM	FM
.5	2.1989	-2.7547	28.8515
1	2.1995	5.6745	22.1177
2	2.1959	4.5465	17.4173

Table 4: Experimental SNR values with varying Modulation Index Factors

Theoretical (Variance = 0.05)			
Index Factor	Conventional AM	PM	FM
.5	-35.9962	-60.5884	61.9119
1	-42.9137	-64.3507	54.9222
2	-48.6574	-69.4518	51.9119

Table 5: Theoretical SNR values with varying Modulation Index Factors

Discussion

After inputting the output of the modulate stage of the scheme into the demodulate state, the plots in Figure 4 were generated. In the time domain, the output from this scheme is almost the same as the original audio plot, the top plot of Figure 4. Similarly, the audio from Matlab's soundsc sounds very clear.

In the bottom plot of Figure 4, the bandwidth appears to be within the window. However, the drop off in the bandwidth is very small on Matlab so the simulation does not conflict with Carson's rule. The graph dies off into infinity so there is infinite bandwidth. About 98% is within the Carson bandwidth for three of the four schemes. For FM, 99% was inside the respective Carson bandwidth. The beta is equal to the product of *k* by the normalized input signal, all divided by the power of the input signal. The normalized input signal is just 1 since it was normalized or scaled by the max amplitude. The bandwidth is really infinite since the plot shows that the graph is mostly centered at the carrier frequency.

Even after demodulation, the time domain plot of Figure 6 does not look exactly the same as the original signal. AM output sound is special here since there was a rectifier used. The main difference is that the low frequencies that appear like a block are from the pilot tone. In the demodulation stage if there is a high pass filter at a low cutoff frequency like 5Hz then the plot will get rid of the low frequencies and the audio will sound much better without the high pitch. However, the plot will appear 'bouncy'. If the AM frequency is increased, then the plot will not be 'bouncy' and is very similar to the original signal.

Figure 8 shows the output of this stage in the frequency and time domains. The audio is not the best when fed through the soundsc function when compared to the original but is still quite clear. Theoretically, FM and PM should have the clearest audio. Adding noise to PM makes the instantaneous frequency or phase change, especially randomly with the noise.

The power of the output was taken after the demodulation. The values from the schemes in Table 1 were all calculated as the square of the root mean square of the output of the demoulation stage of the scheme. The A_c for each scheme was scaled in order to show that the output power of the schemes are equal or at least close to each other. In the PM scheme, the relationship is not very clear or linear between the A_c value and the power so after many iterations the PM power was chosen as the reference point rather than the original signal. The idea is still valid since in practical applications there would almost always be a gain amplifier anyways. Thus, the powers of the schemes had their A_c adjusted to match the output power of PM when that scheme's A_c was 1.

The theoretical SNRs were calculated based on constants including k, the variance, bandwidth, max amplitude of the signal and the modulation index A_c. For theoretical PM, the equation used was $\frac{k^2Ac^2}{2} * \frac{p}{Nw}$, where k is the lower version of FM scheme, p is the power of the modulated stage output and N_ow is the power of the noise output. For theoretical FM, the equation used was $\frac{3k^2Ac^2}{2} * \frac{p}{Nw}$ where k is the higher version of the PM scheme. For theoretical SSB the equation used was $\frac{a^2Ac^2}{1} * \frac{p}{Nw}$ where a is the max amplitude of the input signal. For theoretical conventional the equation used was $\frac{a^2Ac^2}{2} * \frac{p}{Nw}$. N_o can be calculated as twice the variance. All of the SNR are in dB based on the 10 log base 10.

The theoretical SSB SNR were close to theoretical conventional AM SNR which makes sense since the A_c was scaled to ensure the same signal power. The theoretical and experimental SNRs decreased with increasing variance. The SNRs for theoretical are very negative but could have been worse if the $2 * pi / fs$ factor was not taken into account. The negativity suggests an overwhelming effect of additive noise on the output.

The experimental SNR is calculated on the Matlab script as $10log_{10}(\frac{noiseless}{noisy})$ where noiseless is simply the power at the output of the demodulated stage with the noiseless input. Noisy is the power output of the difference between the output of the demodulated stage with the noiseless input and the output of the demodulated state with the noise input signal.

The experimental SNR values in Table 2 are decreasing with increasing noise variance which is fundamentally correct. The power of the noisy signal will increase. This increases the denominator. With that as the denominator the SNR value would decrease.

Acknowledgments

I would like to thank Derek Lee, Thodoris Kapouranis, Jonathan Lam, Anthony Belladonna, Daniel Kim, Steven Lee, Philip Blumin and Paul Cucchiara for helping me better understand the SNR calculations and staying up every night to work on the project with me while listening to my soundsc outputs.