EXPERIMENT AND SIMULATION OF DIGITAL SIGNAL TRANSMISSION IN RELATION TO SIGNAL-TO-NOISE RATIO (SNR)

By Jeremiah K. Jones

Introduction

As technology progresses we are seeing that correct and efficient transmission of information is an increasingly important concern. In particular, digital communications systems are highly common and require complex circuitry in order to properly transmit a signal. For this reason, we can see that it would be valuable to have a basic understanding of some of the important issues relating to digital communications systems. One of the most prominent problems that arise in such a system is that of the Signal-to-Noise ratio (SNR).

SNR is of primary concern in determining such factors as available bandwidth (using Shannon's theorem), necessary power output, and Bit Error Rate (BER). In order to create the most efficient system possible, each of these factors must be taken into account. Therefore, it becomes increasingly important to understand how SNR works in a digital system.

The purpose of this experiment will be to acquire a better understanding of SNR and how it relates to digital signals. Unfortunately, a true understanding of these concepts cannot be fully gained with just one experiment, and would instead necessitate a series of experiments and simulations. Therefore, for the intents of this lab, only the basic analysis of SNR on a small digital signal will be processed. This includes understanding how Shannon's Theorem is affected by the SNR of a system. Shannon's

Theorem states that: capacity of a transmission line = bandwidth x $\log_2$ (1 + signal power/noise power). From this formula we can see that as the power of the noise increases, or as the power of the signal decreases, the total capacity of the line will suffer. Although the ratio of the signal power to noise power does exist in this equation, this is not the formula that is typically implied when discussing SNR. Having made it clear that the amount of signal power compared with noise power is important we must now seek to define this in terms of SNR.

Because signals in communications systems span a very wide range, it becomes difficult to represent them appropriately. In order to compress the range, ratios and logarithms are used as a decibel scale. This is the scale that is used to describe SNR. The final equation that is used to calculate this when using a voltage signal is:

20 log (signal power / noise power).

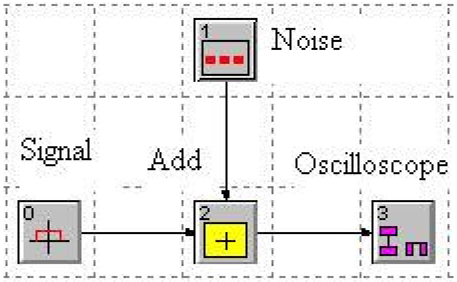
The final result of this experiment will be to calculate the SNR of a simulated digital signal. In order to accomplish this, it is proposed that the student will use LabView to create a square wave to represent a digital signal. Then, using an external signal generator, an extraneous signal will be added into the simulated square wave signal to simulate noise in the system. By isolating the power of the various signals, the student should then be able to calculate the SNR of the digital system. The experiment should be run at various noise and signal strengths in order to make a proper analysis of the results.

Procedure

We started this lab by verifying that we had the appropriate equipment, which included an oscilloscope, a signal generator, connecting wires, and the LabView computer. We then proceeded to work on creating the square-wave generator with LabView. In order to simulate a digital signal, we needed to create a function that would toggle between high and low states continuously. We also wanted to be able to vary the voltage level, as well as the function frequency.

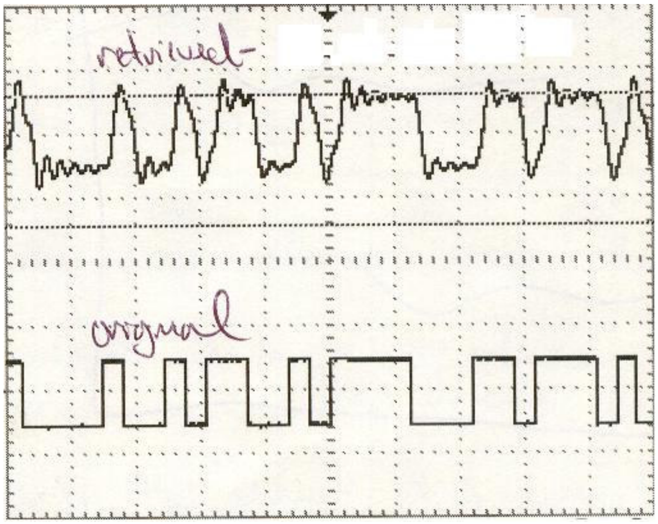
We attempted several different techniques using LabView, but had little success in being able to appropriately vary the frequency. After some time, we concluded that we should have a set frequency and set voltage and should focus more time on the creation of the square wave. Again, we attempted several different methods to create a square wave, including imbedded looping, and logic gates, but we were not able to create a steady and continuous signal. This may have been due to some limitations with LabView. We were able to find a square wave VI on the computer, but this VI did not function exactly as we desired, and did not provide us with the necessary signal.

We then decided to use a second function generator to create the simulated digital signal, as well as the noise signal. Upon obtaining a second function generator from the TA, we were then able to successfully implement the system (diagram shown below).



We set the transmission signal generator to 1 kHz and the noise signal to 2 kHz. We then put both of these signals into the oscilloscope and used the "adder" function on the oscilloscope to add the two signals together. As a note, we tried various frequency levels of noise, and noted that at higher frequencies, the noise did not interfere as much with the signal, and thus we kept the noise at an appropriate level. It is interesting to note that the higher frequency noise does not seem to have as much of an effect on a digital system. We needed to take great care in making sure that the oscilloscope was configured correctly. This included making sure that the ranges of both channels were equal, so that we could easily see an accurate representation of what was going on. We also had some trouble with the trigger mechanism, but can conclude that this was due to the natural wear and tear of the machinery.

After configuring the oscilloscope, we were able to verify the two signals as separate channels, and were then able to add them together with the add function. This produced a signal that appeared as a corrupted digital signal. Because we could not print from the oscilloscope, the diagram shown below is not the actual output, but is a similar output that has been re-created by an external computer.



The next step was to isolate the strengths of both the signal and the noise. In order to accomplish this, we viewed both the signal and noise simultaneously and then adjusted the strength of the signal until the two signals were just barely free from overlapping. We did this because at the point where the two signals begin to overlap is where the digital signal becomes so corrupted that it no longer can be retrieved by the receiver. We then isolated each channel separately and measured the amplitude according to voltage strength.

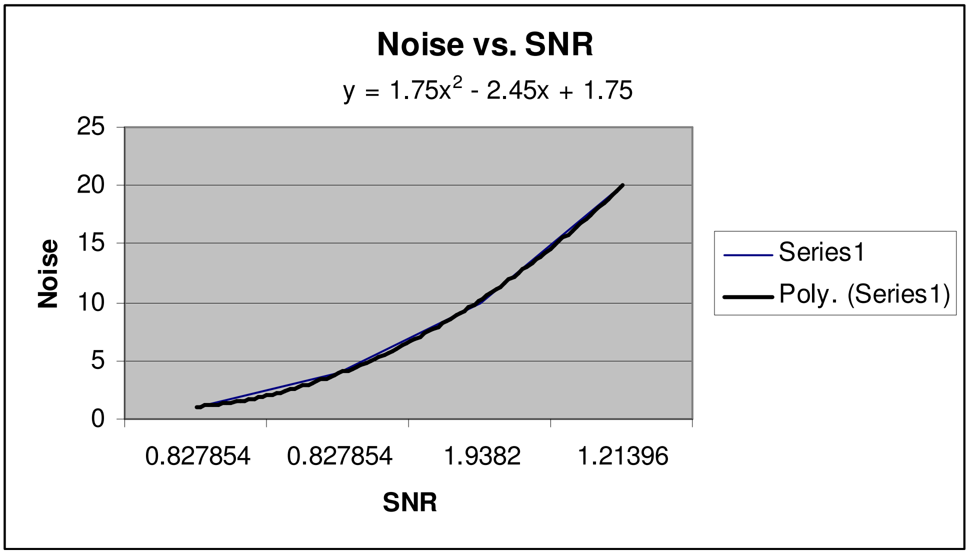
We repeated this procedure a few times, and then recorded the results. The data, as well as the calculation of the SNR of the system is included in the following section.

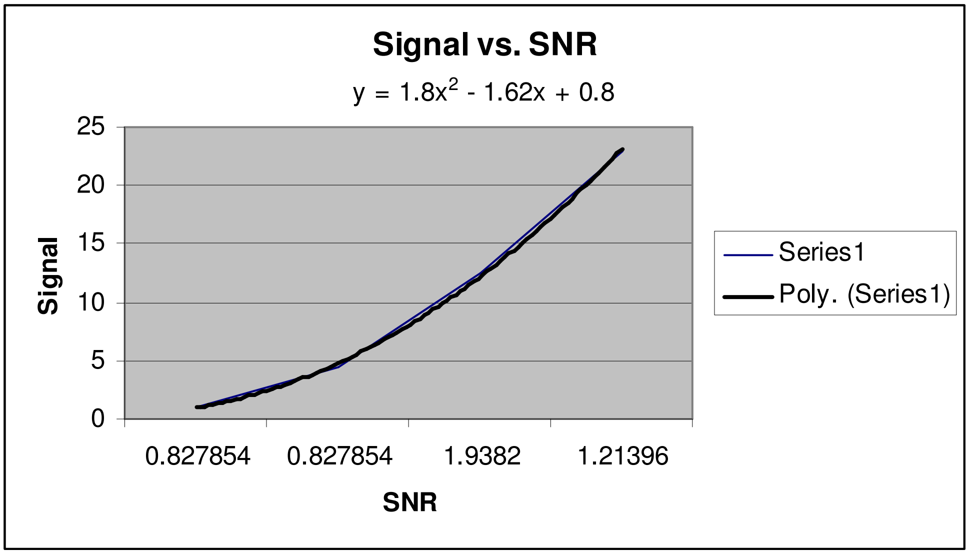
Data, Calculations, and Comments

Although we were confronted by some problems with machine limitation, we were able to take some data points at various noise levels in order to help calculate the overall SNR of the system. The results are listed in the chart below.

Noise Level (V)	Signal Level (V)	Calculated SNR (dB)
1	1.1	0.827854
4	4.4	0.827854
10	12.5	1.9382
20	23	1.21396

As is shown, the SNR stayed within a fairly close range, as should be expected where all other variables in the system were held relatively constant. In order to help visualize the relationship between the Signal strengths and the varying SNR we charted the data as shown below. We also added trend lines with the corresponding equations that are involved in the trend of the system.


Noise vs. SNR
y = 1.75x^2 - 2.45x + 1.75
25
20
15
10
5
0
Noise
0.827854
0.827854
1.9382
1.21396
SNR
Series1
Poly. (Series1)


Signal vs. SNR
y = 1.8x^2 - 1.62x + 0.8
25
20
15
10
5
0
Signal
0.827854
0.827854
1.9382
1.21396
SNR
Series1
Poly. (Series1)

Conclusions and Summary

From this lab, we were able to gain some experience in working with SNR. This included a deepened understanding of Shannon's Theorem. Although we ran into a few technical problems with the lab equipment, all problems were overcome, and the final objective of the lab was met. For future experimentation it is recommended that an oscilloscope with printing capability be used. We also recommend further research into the area of attenuation and bandwidth limitation.