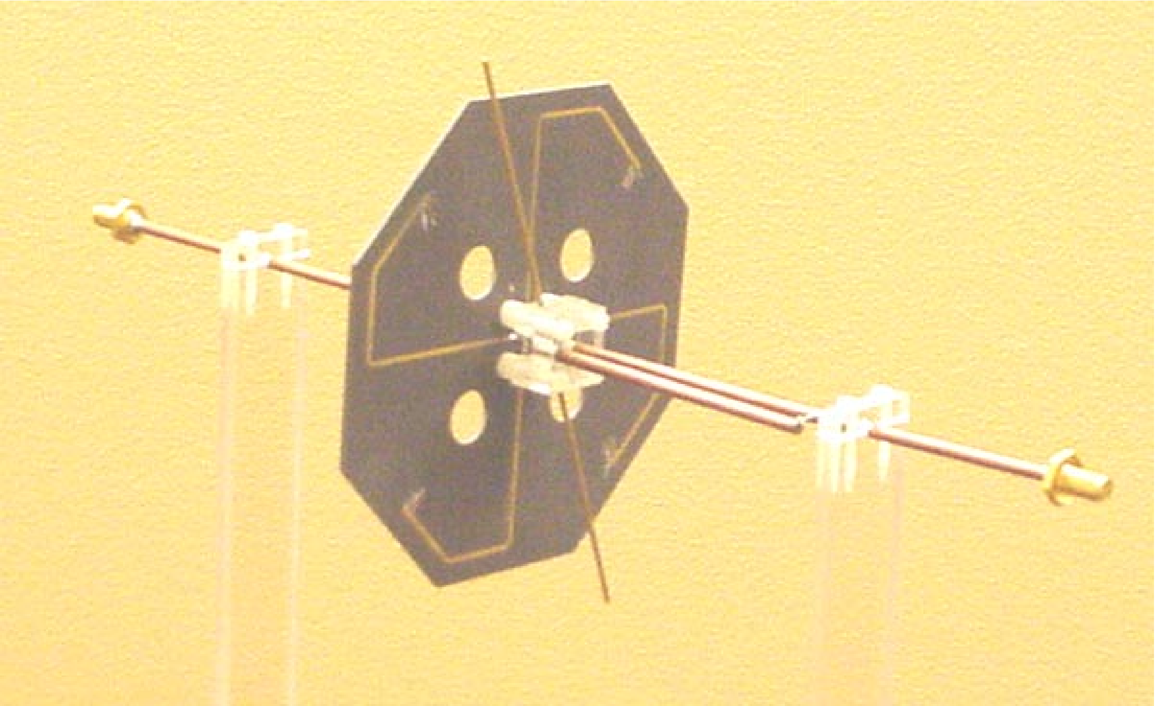
DOUBLING CHANNEL CAPACITY USING POLARIZATION MULTIPLEXING

18-551 Final Report
Spring 2001



Group Eight
Parag Patel <ppatel@andrew>
Shalin Sheth <ssheth@andrew>
J.P. Van't Hof <vanthof@andrew>

Table of Contents

Motivation

Wireless communication has become a fundamental part of our current information infrastructure. However, wireless bandwidth is costly which prompts us to more closely examine the data channel available using electromagnetic waves. It has been well noted that two polarization sates of planar waves allow two distinct information channels using a technique called polarization diversity. In a well-publicized article published in January 2001 by *Nature* entitled, "Tripling the Capacity of Wireless Communications Using Electromagnetic Polarization," mentions largely how wireless channel capacity can be increased using polarization multiplexing. The article suggests that extra channel capacity can be obtained because there are six distinguishable electric and magnetic states of polarization at any given point [1].

The article cites previous experimental work conducted at Bell Labs that demonstrated the increase in capacity originating from the three electric polarization states in a scattering environment. This article has caused much excitement in the wireless communication research industry and we hope to further the research for polarization multiplexing.

The motivation behind our project is to demonstrate the multiplexing across a electric and magnetic polarization state within a scattering environment. This is simply to expand on the idea of polarized antennae diversity using multiple input/multiple output (MIMO) communications because multiple antenna are required to access polarization or spatial channels. We propose the use to two identical co-located magnetic and electric dipole antennas used at the transmitter and receiver to demonstrate electric and magnetic multiplexing. We will use the Texas Instruments C67 Digital Signal Processor (DSP) to extract two distinct data sets from the two independent co-interfering transmitted sources.

Target demo

In order to demonstrate the wireless system using the two antennae at both the transmitter and the receiver we propose the wireless system shown in Fig. 1. The target demo will require that all the processing be done on the DSP and the transmitted signal will be sent from the DSP through the *line out* jack off of the codec on the EVM and then amplified and mixed using external equipment and then transmitted through the antennae. The interfering signal will then be received through the receiving antennae and sent through a mixer and amplified before it is received by the codec through the *line in* jack on the DSP. The received signal is then processed for bit error rate checking.

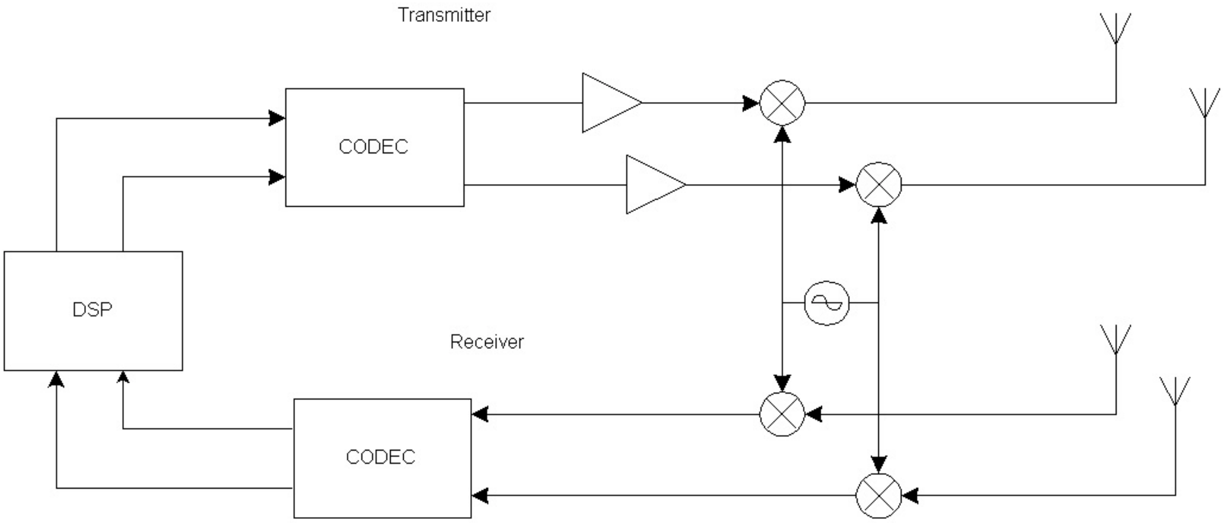


Figure 1: Target System Setup

Overview of project

DSP work

The DSP will be used to for data generation, modulation across the two antennae, data transmission and reception over the C67 codec, adaptive equalization and training, synchronization, demodulation and bit error rate testing. Figure 2 shows a detailed schematic of the processing being done on the DSP and detailed explanation of the algorithms for each component of the system can be found in section 3 of this report.

The transmitted signal is either a randomly generated data bit or a predetermined pilot sequence to retrain the channel estimation and equalization. The data is modulated on the DSP using a binary frequency shift keying (BFSK) modulation scheme. The received signal is then demodulated or synchronized and retrained. The demodulated data is then compared for bit error testing on the DSP.

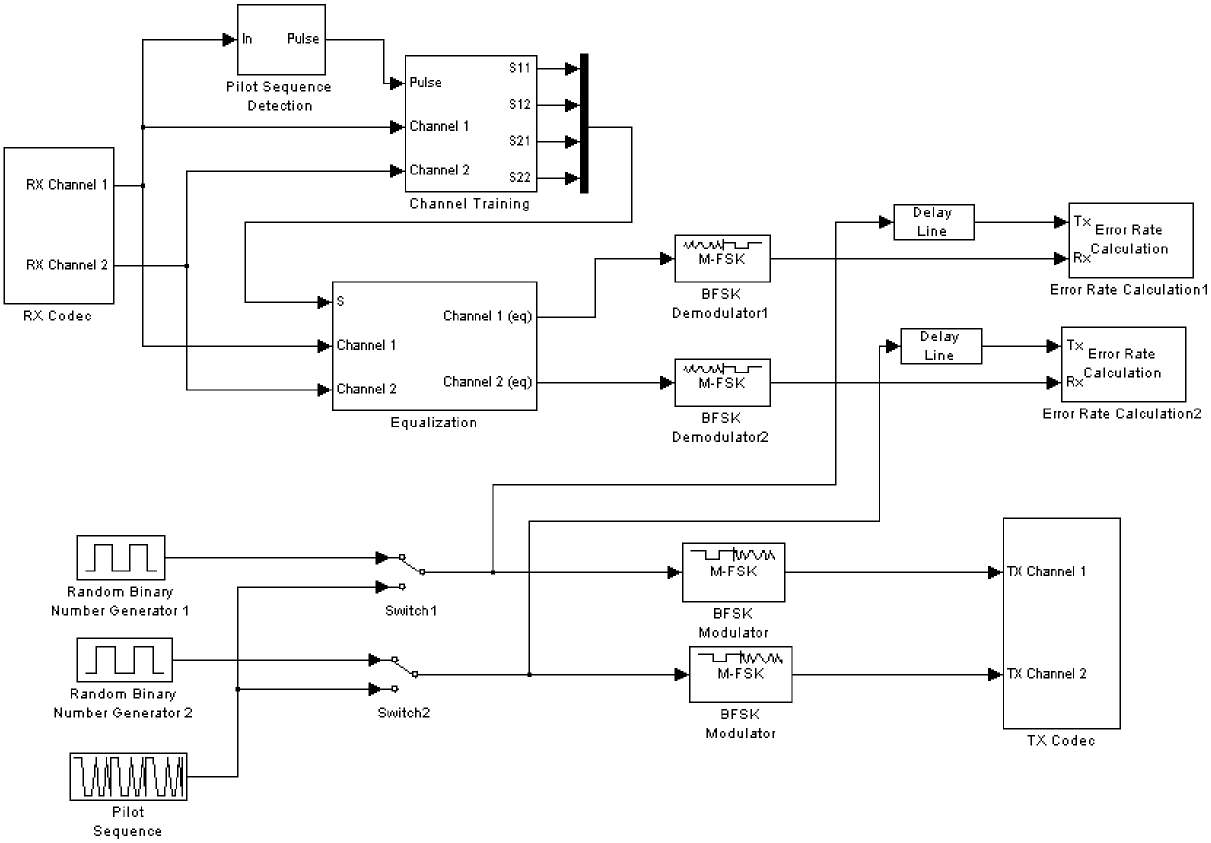


Figure 2: C67 DSP Block Diagram

Lab setup

Due to time constraints the complete RF channel was not able to be setup for the class demo. However, the current lab setup for the demo was conducted using a Simulink simulation of the target scenario. A diagram showing both scenarios is shown in Fig. 3. The current lab setup involves a direct wire connection from the *line in / line out* jacks of the DSP to the *line in / line out* jacks of the second computers sounds card. The second computer will be running the Simulink simulation. The connection between the two computers was also monitored on the oscilloscope. The target scenario will involve a actual RF link instead of the Simulink simulation

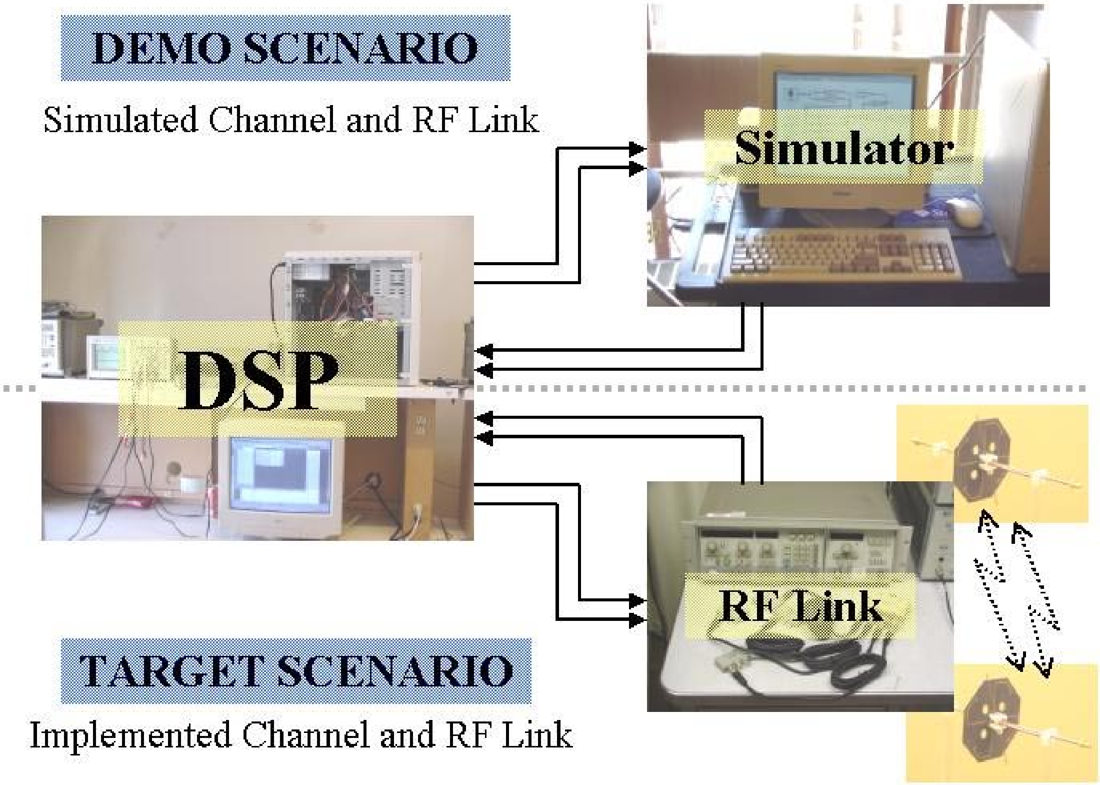

DEMO SCENARIO
Simulated Channel and RF Link
Simulator
DSP
RF Link
TARGET SCENARIO
Implemented Channel and RF Link

Figure 3: 18-551 In-Lab Test Setup

Test procedure

The first test procedure used was to test the wireless system for its comparability to a non-interfering channel wireless system for noise profiling. Then the system was tested for possible lab conditions using the electromagnetic channel estimation parameters for noise profiling.

Project details, algorithms

Transmitter Design

Data Generation and Frame Format

Each frame transmitted has a length of 110 symbols of which 88 contain actual data and the remaining 22 are used for training and synchronizations purposes. During the first 11 symbols of the frame, the 11 bit pilot sequence is transmitted on channel 1 and channel 2 is turned off. After the pilot sequence is transmitted on channel 1, channel 1 is turned off and the same pilot sequence is transmitted on channel 2. 88 bits of data are then transmitted on each channel.

For the pilot sequence, we chose to use an 11 bit Barker sequence. The Barker sequence offers the best peak-to-side lobe ratio, and therefore can be detected by the matched filter receiver with the greatest confidence.

The data bits are generated using a random binary number generator. After generating a bit, the bit is copied into an expect queue and then sent to the modulator. The expect queues store transmitted bits that are later retrieved by the bit error checker for comparison against the received bits.

Modulation

Our communications system uses binary frequency shift keying (BFSK) modulation. In BFSK, a low bit is transmitted as a sinusoid of a particular frequency and a high bit is transmitted as a sinusoid of another frequency. At the sampling rate of 22.05 kHz that our system uses, we determined through simulations in Matlab that we could achieve good performance at a rate of 2205 bits per second per channel using frequencies of 2205 kHz for a low bit and 4410 kHz for a high bit. At these rates, each bit is 10 samples long. The 10 sample symbol patterns for both bits are generated and stored in registers at the start of the program.

All DSP activity related to the transmitter occurs within the transmit interrupt. At the end of each symbol transmission, a request is made to the data generation function for the next pair of bits. The modulator will receive a "high bit", a "low bit", or a "no bit" for each channel. When a "no bit" is received, the channel is turned off and nothing is transmitted. For all other cases, the appropriate symbol pattern is copied from the into the transmit register and sent out one sample at a time on an interrupt-by-interrupt basis. After the last sample of the bit is sent, the request is made for the next bits is made.

Receiver Design

Pilot Sequence Detection

The pilot sequence detector is used to detect the arrival of the pilot sequence on channel 1 and synchronize the receiver. At the start of transmission, the sequence detection is run on each interrupt until the first instance of the pilot sequence is found on channel 1. After the sequence is found, the detector is shut off and a clock is started that is used to trigger events for the rest of the frame. The clock is updated on each interrupt. The detector is turned back on at the point at which the receiver expects the next pilot sequence to arrive on channel 1 to make sure that the receiver is still synchronized.

When each new sample is received, it is copied into a buffer that stores the 110 most recent samples. The detector performs a correlation between the received signal on channel 1 and the expected pilot sequence to determine if the signal in the buffer is a pilot sequence. Rather than performing a direct correlation between the two signals, we use a modified algorithm that only looks at whether each sample has a positive or negative value and increments a counter each time the signs correspond and decrements the counter each time they do not. When the value of the counter exceeds a predetermined threshold value, the pilot sequence has been positively detected.

Equalization

The estimated channel has both channel noise and co-channel interference, which act together affecting the data transmission system in a combined manner. In order to account for the co-channel interference within the estimated channel we decided to use a zero-forcing equalizer (ZFE). The objective of a ZFE is to have the co-channel interference is equal to zero at all times when the channel output is being sampled except during the training process [2]. The training process occurs during the pilot sequence detection period. During the training process the coefficients of the four assumed channels (further discussed in the S-value calculations section) are derived for the S matrix that will be used for equalization.

A major design flaws of the ZFE is that it ignores channel noise and it leads to overall performance degradation due to noise enhancement, a phenomenon that is inherent to the ZFE. The noise enhancement due to the channel estimation is attributed to the singularity of the ZFE equalizing matrix. Equation (1) describes the equalization process where S is the channel estimation matrix, $\underline{x_k}$ are the transmitted bits, $\underline{y_k}$ are the received bits and $\underline{n_k}$ is the added noise per bit.

$$\underline{y_k} = S\underline{x_k} + \underline{n_k} \qquad (1)$$

After testing the wireless system in our simulated channel it is noted that the noise enhancement is much greater than expected and future work in this project should look at other equalization methods. The mean-square error criterion is a more refined approach for receiver designs that is aimed at providing a balanced solution to the problem of reducing effects of channel noise and ISI.

Demodulation

The demodulator, which follows the equalizer in the system signal flow, demodulates the two individual extracted data channels. The input is in the form of BFSK symbols and the output is in the form of bits. Bit decisions are made using a hard-decision method. Hard decisions are made based on a pre-defined threshold; if a sample of the symbol is above the threshold, then it is decoded as a "1." Likewise, if a sample is below the threshold, then that sample is decoded as a "0." The demodulation algorithm used was chosen for its elegance [3]. Shown in Fig. 4, the algorithm accomplishes its task using only a delayed product and a 32nd order FIR low-pass filter. The inner workings of the demodulator are detailed below. The time-domain signals received by the demodulator

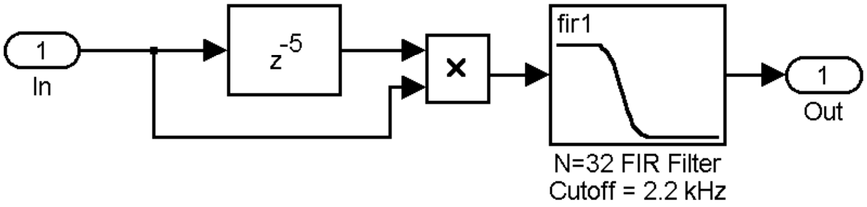


Figure 4: Demodulator Block Diagram

are of the form,

$$s(t) = A\sin(2\pi f_0 t)\,, \tag{2}$$

where f_0 is the frequency of the tone received (recall there are two possible tones in BFSK modulation, f_0 & f_1). After the delayed product, the signal takes the form,

$$s(t)\cdot s(t-t_0) = A^2 \sin(2\pi f_0 t)\cdot \sin(2\pi f_0(t-t_0))\,, \tag{3}$$

which, using trigonometric identities, reduces to,

$$\begin{gathered} s(t)\cdot s(t-t_0) = \frac{A^2}{2}\left[\cos(2\pi f_0 t_0) - \cos(2\pi(2f_0)t - 2\pi f_0 t_0)\right] = \\ \frac{A^2}{2}\left[\cos(2\pi f_0 t_0) - \cos(2\pi(2f_0)t - \phi)\right] \end{gathered}\;. \tag{4}$$

Applying the low-pass filter to (4) using a cutoff frequency of the symbol rate (2.205 kHz), the second cosine term of (4) is attenuated because it is a sinusoid of twice the tone f_0. Recall that f_0 is 2.205 kHz and f_1 is 4.41 kHz. The signal after the filter is now of the form,

$$h_{LPF}(t) * \left[s(t)\cdot s(t-t_0)\right] = \frac{A^2}{2}\cos(2\pi f_0 t_0)\,, \tag{5}$$

which depends only on the delay, t_0 ,and the frequency of the modulating tone, f_0. This suggests that the signal at the output of the demodulator depends only of the modulating tone, once the delay, t_0, has been fixed. Since the modulating tone is chosen based on the bit being sent, the demodulator has recovered the transmitted bit from the received signal.

To ensure that the demodulator performs optimally, the value of (5) when considering both f_0 and f_1 needs to be maximized. This is done by solving the following relation:

$$K = \max_k \left|\cos(2\pi f_0 k T_s) - \cos(2\pi f_1 k T_s)\right|\,, \tag{6}$$

where K is the optimal delay, k is the delay over which the relation is varied and T_s is the sampling period of the system (1/22.050 kHz). When (6) is solved, an optimal delay of 5 is found; thus, the signal will be delayed by 5 samples in the demodulator before being multiplied by a zero-delay version of itself. After demodulation and detection, bits are obtained and are ready for comparison with the transmitted stream in the bit error rate checker (BER).

Bit Error Checker

The bit error checker is necessary to determine how well our communications link performs. The bit error checker compares received bit values against transmitted bit values and gives information about error rates on each channel. Each time a data bit is sent out by the transmitter, it is placed into an expect queue. A separate expect queue is maintained for each channel. When a bit is demodulated by the receiver, it is given to the bit error checker. The checker compares the value of the bit against the next bit in the expect queue for that channel and updates the error counter and total bit counter appropriately.

Once during each second of operation, the bit count and error count information is sent to a program running on the PC using an asynchronous FIFO transfer and the information is displayed in a window.

Simulator

Reasons for using simulator

As the end of the project quickly approached, we realized that a demo that included a real transmitter and receiver, antennas and propagating waveforms was not entirely realistic for our in-lab demonstration. Thus, a simulation that accurately modeled our channel was needed as a substitute for a real system. Such a simulation was devised using the SIMULINK simulation environment, a part of the MATLAB computation suite.

Description of algorithm

The simulation, diagramed in Fig. 5, operates on the audio-range signals created by the transmit side of the C67 DSP in real-time. After the signals have been processed using the simulator, they are sent back to the C67 DSP, where they will be equalized, demodulated and detected. As such, the simulator serves as a functionally even swap for the real channel system mentioned previously. The algorithm used in the simulator is described below.

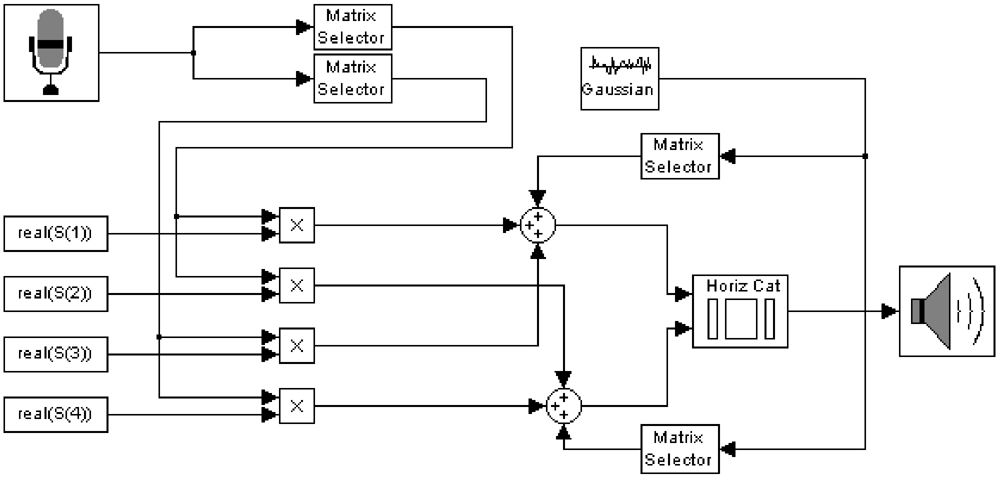


Figure 5: Simulink Channel Simulator

The channel simulated by SIMULINK is represented by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{7}$$

where $\underline{y}$ is the received signal at the DSP (before equalization and demodulation), $\underline{x}$ is the transmitted signal from the DSP (after modulation) and $\underline{n}$ is the noise presented by the channel ($n \sim \mathrm{CN}(0, \sigma^2)$). The 2x2 matrix $\underline{\underline{S}}$ represents the gains of the four different signal channels represented in Fig. 6. This channel representation depicts the gains of the four "effective" channels realized between the two different antennas on the transmitter and receiver. (The reader should keep in mind that there are only two "real" channels, though, one for each receiver.) Thus, the SIMULINK model implements (7) in real time, after it has been given $\underline{\underline{S}}$ and the desired noise power, σ^2. Real-time data is inputted to the model using the host PC's sound card A/D (represented by the microphone block in Fig. 5) and outputted from the model using the host's sound card D/A (represented by the speaker block in Fig. 5). The values of $\underline{\underline{S}}$ are read from the MATLAB workspace using the constant blocks on the left of Fig. 5, and the noise power is inputted into the Gaussian Noise block on the right of Fig. 5.

S-Value Calculations

The values of the matrix $\underline{\underline{S}}$ are calculated using the two-ray ground reflection model for RF wave propagation [4]. This model is represented in Fig. 7. Equations (8) – (11) yield

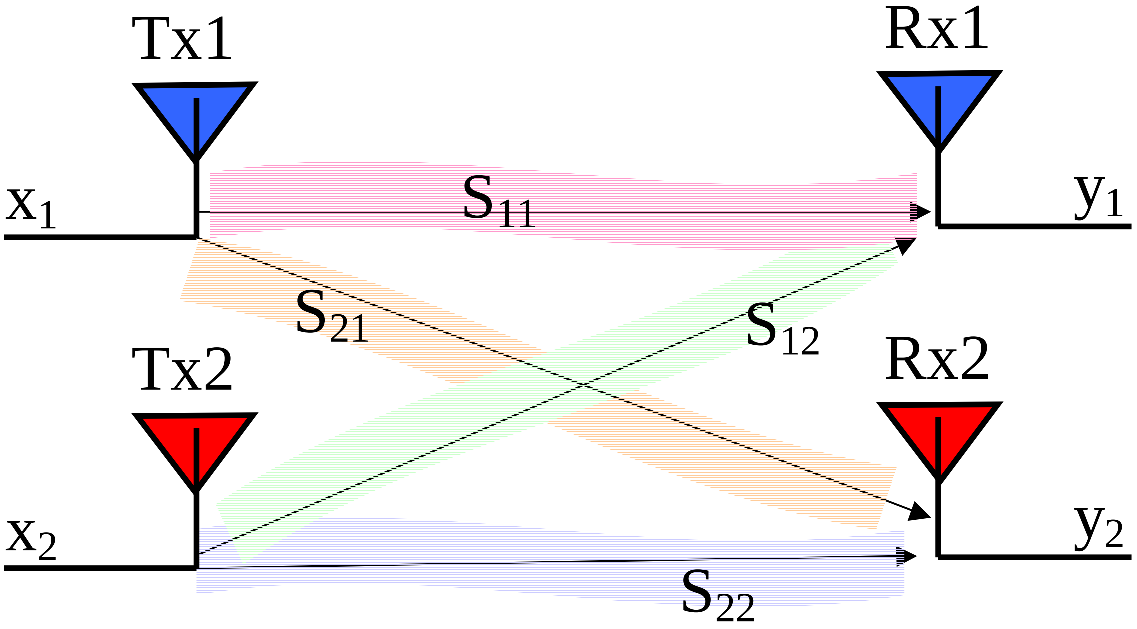


Figure 6: Signal Processing Channel Representation

the values derived from this model; we acknowledge Professor Dan Stancil for assistance in deriving these equations.

$$S_{11} = G_{dipole} \cdot \left[\frac{e^{jkd_1}}{d_1} + \Gamma \cdot \cos^2(\Theta) \cdot \frac{e^{jkd_2}}{d_2} \right] \quad (8)$$

$$S_{12} = \sqrt{G_{loop} \cdot G_{dipole}} \cdot \left[\frac{e^{jkd_1}}{d_1} + \Gamma \cdot \cos(\Theta) \cdot \frac{e^{jkd_2}}{d_2} \right] \quad (9)$$

$$S_{21} = \sqrt{G_{loop} \cdot G_{dipole}} \cdot \left[\frac{e^{jkd_1}}{d_1} + \Gamma \cdot \cos(\Theta) \cdot \frac{e^{jkd_2}}{d_2} \right] \quad (10)$$

$$S_{22} = G_{loop} \cdot \left[\frac{e^{jkd_1}}{d_1} + \Gamma \cdot \frac{e^{jkd_2}}{d_2} \right] \quad (11)$$

In (8) – (11), G_{dipole} and G_{loop} are the gains of the dipole and loop antennas, respectively. The reflection coefficient of the reflecting surface is represented by Γ. The propagation constant, *k*, equals $\frac{2\pi}{\lambda}$, where λ is the wavelength of the carrier (915 MHz), in meters. Using these generated values, SIMULINK simulates the channel, presenting realistic data back to the DSP to be processed in the system's receiver.

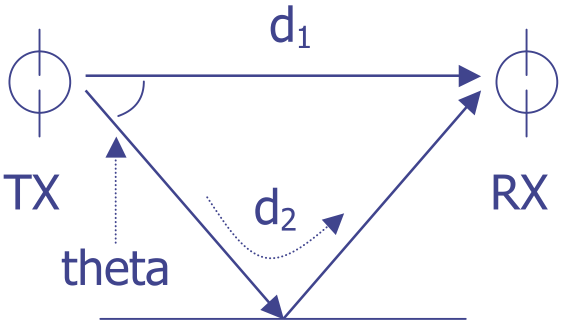


Figure 7: 2-Ray Ground Reflection Model

Simulator non-idealities

While the SIMULINK model does successfully model the channel intended for our system, it does not completely model the real channel. In the real channel, a very large number of signal paths (>>2) will exist in the indoor propagation environment. These additional paths should yield slightly different, and hopefully better, values in the $\underline{S}$ matrix. Currently, the model yields a fairly singular matrix for $\underline{S}$, giving a large amount of noise enhancement (discussed later in this report). A less singular matrix, hopefully resulting from the additional multipaths, would thereby increase the performance of the system by reducing noise enhancement.

It should also be mentioned that SIMULINK is not entirely designed for a real-time environment. As such, there may be some parasitic phenomena present in the simulation, which would be seen as a decrease in performance from the DSP's perspective. Indeed, such performance decreases were noted in test simulations; a restart of the host PC seemed to solved such problems.

Resource Issues

Optimization

Because of the time sensitive nature of our program, optimizing our code was extremely important. All transmitter and receiver activity occurs during interrupts, and in order to ensure proper operation, all processing must be completed during the limited time in between interrupt arrivals. A number of modifications were made from our initial plans to guarantee that we would meet our time constraints.

The greatest speed improvement came in cases where we were able to spread out a large amount of processing over time. In particular, we were able to do this for the channel training and the demodulation parts of the receiver. For the channel training, instead of waiting until the entire pilot sequence had been received, a portion of the computation was done as each sample of pilot sequence arrived. A similar improvement was seen for the demodulation where instead of waiting to collect all the samples that make up a particular bit before running the demodulator, it was run continuously during each sample.

Another optimization method we used was to modify the correlation technique in the pilot sequence detector. A normal correlation would have required the completion of 110 multiplication operations during a single interrupt which turned out to require more computation time than was available. The alternate method of only looking at the sign of each sample allowed us to replace the multiplications with less computationally intensive comparison operations but still providing us with similar performance.

Other optimization methods we implemented included generating test data directly on the EVM instead of transferring it over from the PC, turning off the pilot sequence detector when the demodulator was operating, and performing bit error checking and EVM to PC transfers outside of interrupts.

Memory Usage

The table below summarizes information about the memory requirements of the DSP software. The amount of data memory used is included for both the normal case and the case where no bit error calculations are being performed. The reason for the discrepancy is when BER calculation is disabled, the need for long expect queues that store information about the transmitted data is eliminated. In a real system, the transmitter and receiver would be independent and bit error checking would not be performed.

Memory Type	*Used*	*Available*
On-chip program memory	64064 bytes	65536 bytes
On-chip data memory	43612 bytes	65536 bytes
On-chip data memory (w/ BER disabled)	17508 bytes	65536 bytes
Off-chip memory	0 bytes	8MB

Timing Information

Timing and profile information is given below for the major parts of our program.

Transmitter
    Average interrupt time=0.75ms (125 CPU cycles)
    Maximum interrupt time=2.14ms (354 CPU cycles)

Receiver
    Average interrupt time=21.25ms (3541 CPU cycles)
    Maximum interrupt time=25.43ms (4238 CPU cycles)
        FSK Demodulation time=16.62ms (2770 cycles)
        Pilot sequence detection time=24.42ms (4070 cycles)

Sample time = 45.35 ms (~7558 cycles)

Testing

Test cases

First, we tested the wireless system with 2 non-interfering channels with a simulated identity matrix showing two direct channels only affected by noise. In this test we varied the noise that was being added to learn about the system limits of non-interfering channels. For this scenario, $\underline{\underline{S}}$ equals $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

After testing the non-interfering case, we tested the wireless system for the potential test setting in the labs. The test setting used was an antenna height of 2.5 meters and an antenna separation of 6 meters. This signal power was multiplied by 10 in the Simulink simulator to provide sufficient amplification for the EVM's input. The gamma value of 0.6 was assumed as a reflection coefficient of the lab area. For this scenario, $\underline{\underline{S}}$ equals $\begin{bmatrix} -0.529 & -0.402 \\ -0.402 & -0.264 \end{bmatrix}$.

Test Results

The results obtained from the first (non-interfering) scenario are plotted in Fig. 8. One can see that the performance of the system without interference between the channels seems to be good above 20 dB SNR, but deteriorates quickly below 15 dB SNR. It is likely that this lower limit is dominated by the SNR requirements of the synchronization and demodulation blocks.

The results obtained from the realistic lab test scenario are plotted in Fig. 9. Here, performance can been seen as decreasing appreciably below 45 dB SNR, qualitatively a

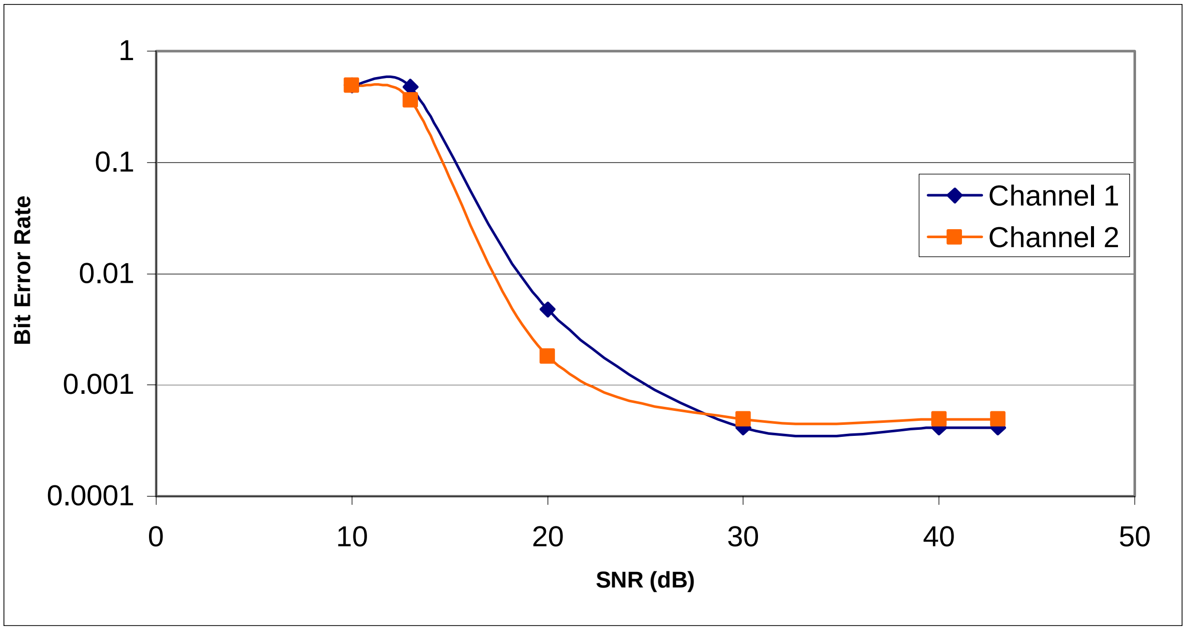


Figure 8: Performance in Non-Interfering Channel

very large value. This increase in required SNR is clearly related to the fact that the channel now is interfering, i.e. each transmitted signal can now be found in each of the received signals. The mechanism of this reduction in performance is presented in the next section.

Noise Enhancement

When using a zero-forcing equalizer, such as has been implemented in this project, a phenomenon called noise enhancement can cause significant losses in performance. When (7) is received by the C67 DSP, the first operation performed is equalization. In the equalization operation, the recovery of $\underline{x}$, the transmitted signal, is attempted. The equalized signal, $\underline{z}$, can then be written as the following:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underline{\underline{S}}^{-1} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underline{\underline{S}}^{-1} \cdot \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{12}$$

Equation (12) suggests that there are two components of the signal exiting the equalizer. The first component is indeed the transmitted signal, $\underline{x}$. The second component is noise; however, the noise has been enhanced by the matrix $\underline{\underline{S}}^{-1}$. This suggests that the noise amplitude at the output of the equalizer is dependant on $\underline{\underline{S}}^{-1}$. In fact, it can be shown that the degree by which the noise, $\underline{n}$, is enhanced is directly proportional to the sum of the squares of the elements in the rows of $\underline{\underline{S}}^{-1}$. Thus, if $\underline{\underline{S}}$ is highly singular, the noise enhancement will become significantly large. Looking at the two sets of data collected and presented in the previous sections, one can see the effects of this noise enhancement. In the non-interfering test, the system operation deteriorated appreciably at 15 dB SNR. Likewise, the interfering case deteriorated at 45 dB SNR, leaving a difference of

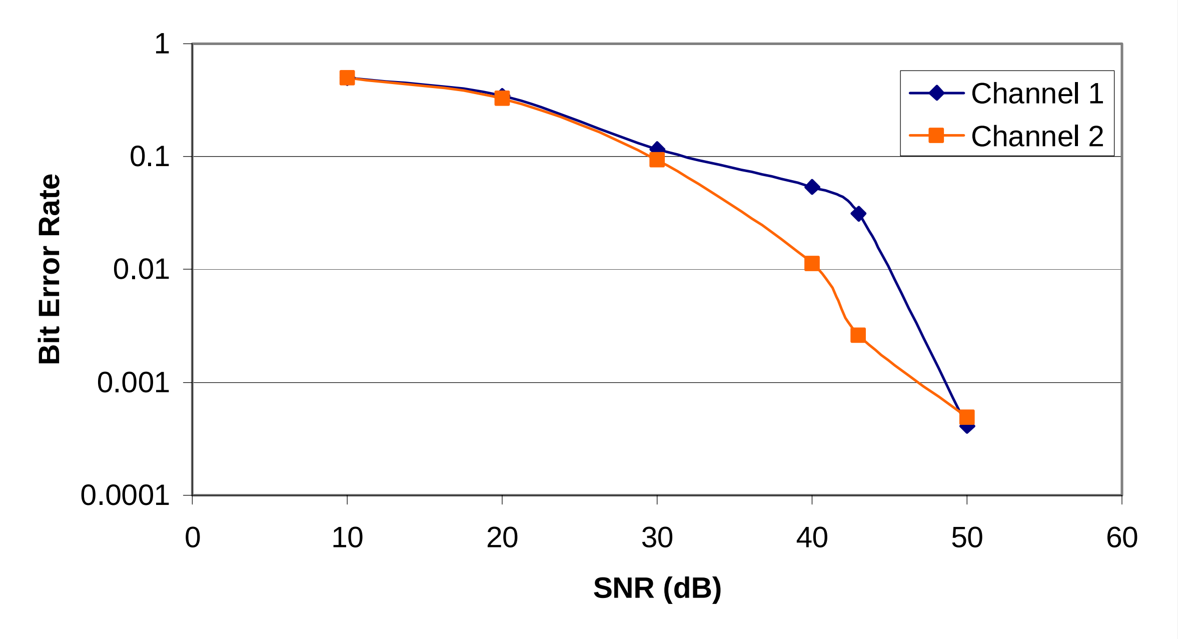


Figure 9: Performance in a Typical Indoor Wireless Channel

approximately 30 dB. When we calculated the expected noise enhancement expected from the simulation, using $\underline{\underline{S}}$ equal to $\begin{bmatrix} -0.529 & -0.402 \\ -0.402 & -0.264 \end{bmatrix}$, we found noise enhancements of 26 dB and 29dB for channels 1 and 2, respectively. As these values closely match the experimental value of 30 dB, we can conclude that noise enhancement presents a significant loss in performance in our system.

Conclusion

The results obtained from the setup described above suggest that the system performs as expected and is able to extract the data from the individual channels successfully. It is also noted that significant noise enhancement is being observed as a result of the zero-forcing equalizer. Furthermore, it appears that the enhancement factor expected is indeed that which is measured when operating the system with a real-time channel simulation. It is expected that the system will function with an even higher performance in the real channel, due to the significantly larger number of multipaths. It is the multipath that allows this system to function, which leads us to believe that this system will function well in any multipath-rich environment, such as the indoor wireless channel.

What's Next?

The next step for the system is experimental testing with the RF equipment described earlier in this report. This would include the use of the custom designed antennas designed for this project. In order to obtain a successful demo, it is almost certain that the equalization method be adapted to the minimum mean square error equalization

(MMSE) method. It is expected that this method will reduce the amount of noise enhancement experience in these initial lab demos. Along with this shift in equalization methods, a shift in demodulation schemes might also be necessary.

Provided a successful result of testing with the RF equipment, a potential paper might be written describing the experiment and processing used within it.

Acknowledgements

We are very grateful to several individuals who helped to make this project a reality. Professors Dan Stancil and Rohit Negi provided a wealth of advice and expertise without which our project would not have proceeded. Rick Burgett graciously provided an additional PC for exclusive use on our project to run the computationally demanding channel simulation using SIMULINK. Durmaz Berber has invested much time into the design and assembly of the RF link, including custom antennas, custom circuitry and RF communications components. Thanks also are extended to Ratish Punnoose, Pavel Nikitin and Richard Tseng, members of the ReH 220 office, whose impromptu advice and feedback were utilized too many times to count this semester.

References

[1] M.R. Andrews, et. al. “Tripling the capacity of wireless communications using electromagnetic polarization,” *Nature,* pg. 316-318, vol. 409, 18 Jan. 2001.

[2] S. Haykin, *Communication Systems.* 4th Ed. John Wiley and Sons, Inc.: NewYork, 2001.

[3] G. Baudoin, et. al., “Teaching DSP through the practical case study of an FSK modem,” Texas Instruments #SPRA347, September 1996.

[4] T.S. Rappaport, *Wireless Communications—Principles and Practices.* Upper Saddle River, NJ: Prentice Hall, Inc., 1996.