Reliable Underwater Feedback Transmission via Software-Defined Acoustic Modems

Caroline Abel
c_abel200@yahoo.com

Ray Chen
raychen2019@gmail.com

James Gallicchio
jamesgallicchio@gmail.com

Grace Zhang
gelei.zhang@gmail.com

Kathryn Zhou
kathrynzhou878@gmail.com

Abstract—Underwater wireless communication provides great potential through semi-autonomous aquatic robots that have use in scientific research, pollution monitoring, and maintenance of underwater facilities. However, the underwater setting poses a unique array of difficulties for wireless communication, including path attenuation, colored noise, Doppler shifting, multipath propagation, and bandwidth limitations. Current underwater robot communication systems lack a robust and reliable feedback channel that mitigates the impact of underwater transmission challenges. This research concludes that the use of low-rate binary frequency shift keying (BFSK) maximizes accuracy of the feedback channel.

Index Terms—Modulation, underwater acoustic communications, software-defined modems.

I. INTRODUCTION

A. Overview

As the world of communication technologies expands, underwater wireless transmission has become an emerging area of research. Underwater transmission is an invaluable piece of technology to develop because of the growing number of aquatic applications including maintenance, discovery, and safety. Applications of underwater transmission include, but are not limited to, remote oil rig inspections, oceanographic data collection, pollution monitoring, offshore exploration, tactical surveillance applications, and marine life research [1]–[3]. Autonomous Underwater Vehicles (AUVs) find applications in underwater monitoring and exploration as an individual vehicle or as a team of vehicles. For instance, Rutgers University has AUVs called “Challenger Gliders,” seen in Figure 1, which collect data from over 128,000 km of ocean basin [4]. Rather than requiring gliders to rise towards the water surface in order to transmit their data, the creation of a reliable underwater transmission system would allow these gliders to send data while they are traveling to the next site; by simultaneously transmitting the data and traveling to the next site, the AUVs save time and the researchers get their data earlier than they would have beforehand. Overall, underwater transmission allows vehicles to communicate information between themselves and the user without the dangerous and arduous task of sending divers or relaying the data physically through another process.

Figure 1. The Rutgers Challenger Glider can take advantage of underwater communication by sending live data to remote sites during travel [4].

New Jersey’s Governor’s School of Engineering and Technology
July 27, 2018

*Corresponding Author
For many of the mentioned applications, a reliable transmission is a challenging problem in such a harsh environment, in which Radio-Frequency (RF) waves cannot travel above a few tens of meters because of absorption in the water. Optical waves could be another option for transmission, but they require narrow laser beams and works for the very short distances. Acoustic waves are the only feasible solution as they are able to propagate up to several tens of kilometers [5]. However, due to high transmission loss and very dynamic channel, it leads to error prone, spectrum limited, and slow communications [6].

Consistency and accuracy are crucial in underwater data transmission, where the properties of the underwater environment pose unique challenges that could potentially result in communication errors [7], [8]. Miscommunication often has critical implications in many applications, such as sending faulty commands to costly underwater robots or receiving incorrect data transmitted from underwater drones. Thus, relevant research is necessary to address these concerns. To achieve these goals, the possibility of replacing the traditional underwater hardware-based acoustic modems with the software-defined modems should be explored. Currently, traditional commercial acoustic modems with their fixed-hardware designs are more common. Meanwhile, software-defined modems are highly flexible, reconfigurable, and reprogrammable [9]. These modems use minimal hardware, and depend on the host's, i.e a computer, cpu and other resources to process the data.

B. Purpose and Motivation

The purpose of this research is to understand the complexities of underwater transmission and explore factors that can lead to robustness, reliability, and accuracy. Dr. Dario Pompili, director of the Cyber-Physical Systems Laboratory (CPS Lab) and an associate professor of Electrical and Computer Engineering at Rutgers University, Mehdi Rahmati, a Rutgers PhD candidate, and their research team concern the development of a real-time software-defined multi-antenna communication system between autonomous underwater robots and remote transmitters/receivers on the buoy/land station, seen in Figure 2. The main goal is to create high-speed acoustic links between the robots and the buoy/land station, so that underwater videos, captured by the robots, can be processed, compressed, and transmitted through these high-speed links in a timely manner and under a certain level of Quality of Service (QoS). Achieving this goal requires a closed-loop algorithm that creates a reliable feedback delivery system for underwater drone usage, where different encoding tools can be used to reduce potential errors. In their research [10], a test-bed is formed based on software-defined Universal Software Radio Peripheral (USRP) with high processing capabilities. The system is cable of transmitting videos, which are captured by underwater robots, processed locally and coded with Scalable Video Coding (SVC) H.264/MPEG-4 AVC compression standard.

To have a bidirectional communication and to guarantee data delivery, a reliable feedback system is required in the reverse direction—from buoy/land station to robots—to acknowledge the received packets and to issue the up-to-date commands, accordingly. However, without a robust communication system, the feedback system would compound error upon error while attempting to detect or correct previous mistakes. The current research aims at investigating and optimizing the most stable and reliable methods possible to guarantee the reliability of the mentioned system.

C. Paper Organization

The remainder of the paper is organized as follows. Sect. II introduces the fundamental concepts and the basics, which are used through this paper. Sect. III includes the proposed solution for a reliable underwater acoustic transmission. Sect. IV presents the performance evaluations and results based on the simulations; finally, Sect. V concludes the paper and discusses the future path of this research.

II. BACKGROUND AND THE RELATED WORK

A. General Telecommunication

![Figure 3. The overall process for general telecommunication transmissions. It can be summarized as the collection and conversion of data into a wave, the traversal of the wave through space, the interception of the wave by an antenna, and the conversion of the wave back into data [13, pp. 2-3].](image)

The information to be transmitted (text file, audio, video, etc.) starts as a series of 1s and 0s which occupies units of memory called bits. The information is compressed, such
as in a zip file, to reduce the quantity of information to be transmitted; this is the source encoder phase in Figure 3. Afterwards, redundant information is added in the channel encoder phase. These redundant bits are relevant in the channel decoder phase, where they are used to correct potential errors that occurred during the transmission process. Then, during the modulation phase, the encoded string of binary bits is converted into a voltage wave in which the waves’ amplitudes, phases, and/or frequencies contain the transmitted information [14]. After modulation, the transmitter antenna outputs the wave generated, and this wave travels through a medium or field called the channel. In the process of traveling through the channel, external noises interfere with the transmitted waves. Specifically, the channel does not differentiate between the transmitted wave and external waves. Thus, the frequencies of the external sound waves are added on top of the transmitted wave to form a singular superimposed wave.

Concerning the receiver phases, there is a receiving antenna that intercepts the superimposed transmitted wave and delivers it to the demodulator. The demodulator converts the incoming wave into a string of bits. However, due to the superimposed wave’s inclusion of external sounds, the demodulator may incorrectly convert the wave, producing a series of bits that does not match the original wave. The channel decoding phase is used to fix these transmission errors. Then, the source decoder takes the string of bits and translates it back to the original file for the receiver to view [14].

1) Symbols: Specifically in modulation and demodulation, there is an intermediate phase between the bit string and the transmitted wave. Initially, the bit string is split into n-bit words (e.g. 2-bit words: 00, 01, 10, or 11. For 3-bit words, there are 8 possible words). In modulation, these words are mapped to a series of symbols that describes the physical wave to be transmitted. Time variant sinusoids are defined in general by three parameters: amplitude (the peak intensity of the wave, A), frequency (the number of cycles of the wave per unit time, f), and phase (the time offset between some starting time and the waves first cycle, φ). The generalized sinusoid parent function can be summarized as follows,

\[ I(t) = A \sin(2\pi ft + \phi). \]  

Each possible symbol is associated with a distinct combination of A, f, and φ in Equation (1), giving the symbol a distinct waveform that can be demodulated at the receiver. The mapping of bit words to symbols is known as a modulation scheme. There are a multitude of modulation scheme types that vary by the defining parameters.

2) Modulation Schemes: The most basic modulation schemes utilize only one of the three parameters: phase, amplitude, or frequency.

For amplitude shift keying (ASK), a sinusoidal carrier wave is modulated to have distinct amplitudes for each symbol [16]. The frequency and phase of the wave remain constant, while the amplitude jumps between distinct values. When the antenna receives the transmission, it demodulates the signal via its amplitudes. In Figure 4, the data in the first line corresponds to the ASK signal in the second line; a bit value of 1 corresponds to a high amplitude wave segment while a bit value of 0 corresponds to a low amplitude wave segment.

Similarly, another class of modulators is frequency shift keying (FSK). FSK operates in the same manner as ASK, but by varying frequencies between distinct values to represent each symbol instead of using amplitudes [16]. In Figure 4, the data in the first line corresponds to the FSK signal in the second line; a 1 corresponds to a high frequency wave segment while a 0 corresponds to a low frequency wave segment.

Meanwhile, phase shift keying (PSK) relates the data to the transmitted waves’ phases. In the string of binary data bits, if the bits alternate - either a 1 followed by a 0 or vice versa - then the carrier signal will have a 180 degree phase shift [16]. This is seen in Figure 4 between the first line of data and the corresponding PSK wave in the fourth line.

Each modulation scheme has its strengths and weaknesses depending on the situation or task at hand. In the underwater environment, acoustic FSK shows promise as a medium-range communication method. The underwater signal waves often encounter and reflect off surfaces, including the ocean surface, sea bed, and aquatic landmarks, which produce echoes that change direction and interfere with the original transmission. Known as multipath propagation, this interference shifts the amplitude and phase of the signal, rendering ASK and PSK very unreliable. FSK circumvents this problem by solely focusing on frequency. Since the demodulator only needs to distinguish one symbol from another, or one frequency from another, in order to determine the correct tone, the changes in amplitude or phase due to underwater echoes have no effect on FSK.

The general form of FSK is M-FSK, where M indicates the number of distinct frequencies used. In M-FSK, there are M symbols, with each symbol corresponding to a single unique tone. Given the binary nature of computers, M is almost
exclusively a power of 2; in this configuration, $\log_2 M$ bits can be sent in a single symbol. For example, in binary frequency shift keying (BFSK, $M=2$), data is represented by patterns of two values, 0 and 1. In quadrature frequency shift keying (QFSK, $M=4$), there are four possible 2-bit combinations. The symbol 00 represents one frequency ($f_1$), 01 another ($f_2$), 10 ($f_3$), and 11 ($f_4$), while amplitude and phase remain constant. The same technicalities apply to modulation schemes other than FSK, such as M-PSK (BPSK, QPSK, etc.) and M-ASK (BASK, QASK, etc).

B. Underwater Difficulties

1) Time Variant Factors: The complexity of underwater transmission lies in the variances of the acoustic channel environments. Multiple factors, such as temperature, salinity, and pressure, can greatly impact the characteristics of the aquatic channel.

2) Bandwidth Concerns: The available bandwidth for underwater communications is 100 kHz. This is an extremely small range compared to the bandwidth for radio communications in air, which is on the order of gigahertz. A small bandwidth limits the maximum bit rate, the speed that bits are transmitted.

3) Noise: All transmissions will be affected by external noises from the environment, but underwater communications deal with additional noises that do not occur in general transmissions. For terrestrial communications, external noise can be modeled by additive white Gaussian noise (AWGN), which is normally distributed and has uniform power across the frequency band.

Underwater transmissions introduce complications since there are additional forms of noises that cannot be modeled by white noise. These include but are not limited to man-made sources of noise from sea vessels and underwater machinery, ambient noise from currents, tides, and storms, and noise from sea life [1]. These noises are highly dependent on the specific environment. However, they can be generalized as Brownian, meaning that lower frequency noise is more pronounced than higher frequency noise.

Sea animals send and detect various frequencies for communicating with others, acquiring food, and navigating through their environments. For example, the bottlenose dolphin uses clicks of frequencies between 40 and 150 kHz for echolocation [2, Fig. 6]. Since this research will use a frequency band of 50 to 150 kHz, the frequencies transmitted by sea life could potentially interfere with the signals transmitted by humans, and vice versa.

4) Path Loss/Attenuation: As the carrier wave propagates, the amplitude of the signal decreases, a phenomenon known as path loss or path attenuation [17]. Since water is heavy and therefore takes more energy to move as compared to air, the underwater acoustic waves’ acoustic energy converts to heat energy as the distance traveled increases. With modulation schemes that do not rely on the wave amplitude (e.g. PSK and FSK), a reduction in amplitude is negligible, so long as the signal’s amplitude is large enough to be differentiable from the background noise.

5) Multipath: The most substantial difficulty of underwater acoustic communication is multipath propagation. Multipath propagation is caused by underwater reverberation effects, which are signal reflections from the surface and the ocean floor. The multiple reflections, or echoes, can interfere with one another. For ASK and PSK, this interference could have extreme consequences because signal reflections affect both the amplitude and phase. Frequency, in contrast, is unaffected by these reflections, making FSK seem a good candidate to avoid the effects of multipath propagation. However, echoes can interfere either constructively or destructively, potentially making demodulation difficult. Furthermore, inter-symbol interference (ISI) may occur if the multipath propagation spread is longer than the signal duration [19]. In this scenario, symbols carried by reflected signals are delayed enough to interfere...
with subsequent symbols and can confuse the demodulator by appearing to show multiple symbols simultaneously.

6) Doppler Effect: The Doppler effect refers to frequency shifts in underwater transmissions as the result of the transmitter and receiver moving relative to the water [20]. As witnessed in Figure 7, an observer will hear different pitches depending on the relative velocity of the transmitter (in this diagram, the ambulance) and the receiver (the person listening). If the ambulance approaches the listener, the observer hears a higher frequency than the actual frequency. Similarly, if the ambulance distances itself from the listener, the observer hears a lower frequency. For frequency dependent modulation schemes, such as FSK, this frequency shift could be large enough to introduce demodulation errors.

Figure 7. If the sound source is approaching the observer, the perceived signal has a higher frequency than the original. If the sound source is leaving the observer, the perceived signal has a lower frequency than the original [21].

III. PROCEDURE

The overall research process consisted of simulating three underwater-specific issues, analyzing their effects on the transmissions’ accuracy, and utilizing the results to optimize underwater acoustic modulation for accuracy. Given the advantages posed by FSK schemes in handling multipath propagation— the most substantial difficulty for underwater transmission—this research tested multiple FSK schemes (BFSK, QFSK, 8-FSK, and 16-FSK), as well as BPSK as a control to compare these FSK schemes to a non-FSK scheme. All five modulation schemes were tested under 1) Brownian Noise, 2) Doppler Effect with Brownian Noise, and 3) Multipath with Brownian Noise. These three simulations were representative of most of the difficulties presented within the underwater environment.

Additionally, the inherent limitations of the channel regarding frequency band and power were obeyed in all simulations. Because underwater channels generally operate between 50kHz and 150kHz, all tests used an average bandwidth value of 100kHz. Similarly, signal-to-noise ratios below 20 dB were varied to model relatively small transmitters with limited power for acoustic output.

Lastly, a signal was generated using FSK and sent through a 20 gallon fish tank testbed in a Rutgers Engineering Lab measuring 3 feet by 1 foot by 1 foot; this was the closest thing to testing the modulation schemes in the real world.

A. BER vs. SNR

For each of these simulations, a BER vs. SNR graph was generated. BER or bit error rate is the number of bit errors per bit and SNR or signal-to-noise ratio is a measure of how loud the signal is in comparison to the background noise. Graphs of BER vs. SNR are a clear way to see at what point the quality of the channel is compromised as they show the accuracy at different SNRs. The most desirable outcome is to have the lowest SNR possible with acceptable error.

B. Generating Testbed Signals

Initially, it was necessary to obtain a 1 MB data file representing the transmission wave. After downloading the MATLAB communication toolbox, a MATLAB program was created to modulate the data using BFSK and transform it into a passband signal. This signal was subsequently sent through a testbed, a physical manifestation of underwater transmissions similar to the real world environment. The testbed was a 20 gallon fish tank that measured three feet by one foot by one foot.

C. Simulating Brownian Noise

Brownian noise was poorly supported in the MATLAB environment. More factors must be manually accounted for when using this type of noise. Raw Brownian noise was generated using the Digital Signal Processing toolbox. Next, the relative average amplitude of the noise was adjusted to conform to the selected sound-to-noise ratios. The root mean square function (rms) was then used to “measure” the average amplitude of the noise and of the signal, a common technique in signal processing. The raw noise was normalized using the formula:

\[ \text{noise}_{adj} = \text{noise}_{raw} \times \frac{\text{rms}_{signal}}{\text{rms}_{noise}} \text{snr} \]  

The adjusted noise (noise\textsubscript{adj}) was then added directly to the signal.

D. Simulating Doppler Shift

To simulate the Doppler effect in an underwater setting, the Doppler equation can be used to calculate frequency shifts. Given the velocity of the carrier wave, \( v_{wave} \), the velocity of the transmitter, \( v_{source} \), and the velocity of the receiver, \( v_{observer} \), the Doppler shift is described by:

\[ f_{observed} = \frac{(v_{wave} \pm v_{observer})}{(v_{wave} \pm v_{source})} f_{actual} \]

Sound travels at an average \( v_{wave} = 1500 \text{m/s} \) underwater. For this research, \( v_{source} \) was assumed to be 0, while \( v_{observer} \) was tested at -10 m/s, -5 m/s, -1 m/s, 10 m/s, 5 m/s, and 1 m/s. These selected values were representative of velocities in the real world, while also taking into account directionality and producing graphs with visually prominent shifts. With each velocity, the frequency shift was applied to a modulated
signal. The Brownian noise was added to the shifted signals according to the chosen signal to noise ratios, interpolating the noise to match the length of the signal. Next, the bit error rate was decimated, demodulated, and calculated for each signal to noise ratio and velocity combination. This was performed for all five modulation schemes.

E. Simulating Multipath

To simulate multipath signals and path attenuation, the Rician Channel function within the Communications toolbox was utilized. The sampling frequency of the path was set at 200 kHz. This was derived from the Nyquist rate, which dictates that the minimum sample rate for accurate demodulation is twice the bandwidth of the function. Since the bandwidth used in this project is 100 kHz, the sample rate was set to 200 kHz [22]. The path delay parameter describes the time, in seconds, it takes for an additional path to arrive at the receiver, relative to the main path. Path gain refers to the change, in decibels, that any additional path undergoes post multipath propagation reverberations. The PathDelays and AveragePathGains were set using empirically determined values that represent relatively heavy multipath propagation. In order to analyze the effects of the Doppler Shift on the signal separately, the Doppler Shift component of the Rician Channel was set to 0. The Doppler Effect was simulated using the frequency shift aforementioned. After running the signal through the Rician channel, the frequency distribution of the signal was displayed using the Welch Power Spectral Density Estimate, as seen in Figure 8. There is a noticeable shift in frequencies observed.

Figure 8. After processing the signal through the generated Rician channel, the frequency distribution shifts. The blue line depicts the original frequency distribution of the signal, while the red line depicts the frequency distribution of the signal after being passed through the Rician Channel.

IV. RESULTS

A. Noise-Only Test

When Brownian noise was introduced to the signal, all of the tested modulation schemes handled the issue relatively well. However, the BER for BFSK dropped off significantly more quickly than other schemes. As seen in Figure 9, BPSK reached a BER below $10^{-4}$ at an SNR of around 4 dB. In comparison, Figure 10 shows that BFSK reached similar error rates at a much lower SNR around -5 dB.

The higher order FSK schemes are each less accurate than the previous. QFSK has a simulated error rate of around $10^{-3}$ by -3 dB; 8-FSK reaches a BER below $10^{-2}$ by an SNR of 0 dB; 16-FSK reaches a BER below 2% by 0 dB. Though the simulations do not offer a complete image of the BER for various SNRs across all schemes, the data can be extrapolated to find that BFSK, QFSK, and potentially 8-FSK all perform better than BPSK when Brownian noise is added.

B. Doppler Effect Test

BFSK was tested under the Doppler effect with a range of relative velocities between -10 m/s and 10 m/s. The BER vs. SNR graphs after applying four different frequency offsets were almost indistinguishable from each other, as shown in Figure 11. The similarity of each plot indicates that Doppler shifting at relatively low velocities has minimal impact on BFSK transmissions.

It is unclear whether the effect would be more pronounced on other modulation schemes, as only BFSK was tested in this case. With higher modulation orders, frequencies are closer
Figure 11. BER vs. SNR for BFSK after Doppler effect.

together, so these small frequency shifts may cause a more noticeable change.

C. Multipath Propagation Test

Figure 12. BER vs. SNR for BPSK with multipath propagation.

Figure 13. BER vs. SNR for BFSK with multipath propagation

Multipath propagation significantly impacted the accuracy of signals across the board. BPSK performed very poorly under multipath, with an error rate above 1% for SNRs as high as 10 dB (see Figure 12). BFSK performed noticeably better, dropping below $10^{-2}$, but still performed rather poorly, shown in Figure 13. Figure 13 shows BFSK at four different symbol rates, under the prediction that lower symbol rates (slower signals) would avoid inter-symbol interference and improve performance under multipath propagation. However, other than 10 kHz, these transmissions were affected similarly by multipathing. The comparative success of the 10 kHz transmission has no obvious explanation at the moment, and will require further testing to understand fully.

D. Testbed Signals Results

Figure 14. Frequency distribution analysis of BFSK signal after being sent through aquatic test bed.

After sending the BFSK signal through the test bed, a frequency distribution analysis was performed, depicted in Figure 14. The two frequency peaks characteristic of BFSK are clearly visible. However, more analysis on the signal must be done to determine if the signals are too distorted to be accurately demodulated.

V. CONCLUSIONS

Overall, simulation results indicate that FSK is effective at very low SNRs, with BFSK performing the most optimally and reaching an acceptable error rate at significantly below 0 dB. This suggests high reliability, as this indicates that the signal could resist high levels of noise. FSKs of higher degrees were less successful, but still achieved reliability at low SNRs. Underwater noise favors low frequencies well outside the 50kHz to 150kHz range used by these FSK schemes, which may have contributed to their success.

In simulation testing, BFSK was highly resistant to Doppler shifting, but it is currently unclear what results would be produced by other modulation schemes. However, given the low speeds of objects traveling underwater compared to the speed of sound underwater, it is probable that other versions of FSK would also be resilient.

Moreover, simulation results indicate that multipath propagation drastically increases error in transmissions across the board. FSKs are more resilient to multipath propagation, and we can reduce the impact of multipath propagation by...
decreasing overall inter-symbol interference. However, the combination of BFSK and low symbol rates could result in an unacceptably low transmission rate. Similar success, but with a higher transmission rate, can be achieved with QFSK.

In the future, signals for modulation schemes other than BFSK could be generated and sent through the testbed. This research simply plotted the power spectral density of the modulated signal for visual comparison against the original transmission, with the natural next steps being attempting to decode the faulty signal and analyzing the corresponding error rates. Additionally, one other specific testbed shows promise for further testing: an indoor pool at Rutgers Science and Engineering Resource Center that is larger and more realistic than the fish tank used for this experiment. With this resource, more real life data could be produced in the lab instead of relying on mostly simulations.

The creation of a new modulation scheme is also worth consideration. Current modulation schemes (ASK, PSK, and FSK) have been designed for terrestrial transmissions. However in underwater environments, ASK and PSK are not as effective due to significant water-related external influences on amplitudes and phases. However, the frequency of the wave is less likely to be affected by the underwater setting, making FSK most advantageous for underwater settings. To further maximize the benefits of FSK underwater, creating a multitonal frequency shift keying (MTFSK) modulation scheme would have future potential. By using MTFSK schemes, the transmitter is able to send two signals or tones simultaneously through the same channel. While traditional FSK schemes use single wave frequencies that correspond to different symbols, MTFSK uses multiple frequency combinations. Since there are a greater number of possible frequency combinations than there are the number of individual frequency variants, MTFSK schemes will have more symbols than typical FSK schemes. Thus, more data can be stored through MTFSK than FSK. This makes data transmission faster compared to the other modulation schemes as the same number of symbols per second are sent, but more data per second is transmitted.

ACKNOWLEDGMENTS

The authors of this paper gratefully acknowledge the following: Project Mentors at the Cyber-Physical Systems Laboratory (CPS Lab), Rutgers University, Dr. Dario Pompli, Mehdi Rahmati, and Adam Gurney for their valuable knowledge of engineering and hands-on involvement; Residential Teaching Assistant Rajas Karajgikar for his invaluable assistance; Research Coordinator Brian Lai for his assistance in conducting proper research; Head Counselor Nicholas Ferraro for his guidance and support throughout the program; Dean Ilene Rosen, Ed. D., the Director of GSET and Dean Jean Patrick Antoine, the Associate Director of GSET for their management and guidance; Rutgers University, Rutgers School of Engineering, Rutgers’ Electrical and Computer Engineering Department, and the State of New Jersey for the chance to advance knowledge, explore engineering, and open up new opportunities; Lockheed Martin, Silver Line, Rubik’s, and other Corporate Sponsors for funding of our scientific endeavours; and lastly NJ GSET Alumni, for their continued participation and support.

REFERENCES

A. Appendix A

% MODULATOR TEST :: Tests a modulator for accuracy at a variety of SNRs, with a combination of additional issues (Doppler shift and multipath)

% Set up modulator and demodulator
mfsk = 8; % 2/4/8/16
mfskbits = round(log2(mfsk));
sampleRate = 200e3;
symbolRate = 10e3;
samplesPerSymbol = round(sampleRate/symbolRate);

mod = comm.FSKModulator();
mod.ModulationOrder = mfsk;
mod.FrequencySeparation = 100000/mfsk;
mod.SymbolRate = sampleRate/samplesPerSymbol;
mod.SamplesPerSymbol = samplesPerSymbol;

demod = comm.FSKDemodulator();
demod.ModulationOrder = mfsk;
demod.FrequencySeparation = 100000/mfsk;
demod.SymbolRate = sampleRate/samplesPerSymbol;
demod.SamplesPerSymbol = samplesPerSymbol;

% Get data to modulate
% Open the test data text file
file = fopen('data.raw');

% Make sure the file exist
if (file == -1)
    error("File not found!")
end

% Read from the file bits at a time
data = fread(file, Inf, num2str(mfskbits, '*ubit%-d'));
data = data(1:100000);
len = length(data);
fclose(file);

% Modulate the data
signal = mod(data);

% NOISE ONLY test

% Generate brown noise
cn = dsp.ColoredNoise('Color', 'brown', 'SamplesPerFrame', size(signal,1));
rawNoise = cn();
rms = dsp.RMS;
noiseRms = rms(rawNoise);
sigRms = rms(real(signal));
noise = rawNoise * (sigRms/noiseRms);

% Signal to Noise Ratios
snrDecibels = linspace(-20, 0, 8); % dB, dB, # points
snrRange = 10.^(snrDecibels/10);
E = zeros(size(snrRange)); % initializes an array for the error rates

%#
% Iterate through SNRs
for i = 1:size(snrRange,2)
    % Grab SNR
    snr = snrRange(i);
    % Add noise
    noisy = signal + noise / snr;

    % Demodulate and count error
    E(i) = sum(demod(noisy) == data);
end

clear noise;
clear noisy;
release(demod);

% Adjust for length of data to get per-bit error
E = E ./ (len*mfskbits);

% Display the error rate data with the SNR values in decibels
figure;
semilogy(snrDecibels, E);
title(num2str(mfsk,"-%dFSK Simulated BER vs. SNR"));
xlabel("SNR (dB)");
ylabel("BER (errors/bit)");

% Doppler shift + noise (1 graph, multiple lines)

% List of relative velocities between transmitter and receiver
relv = [-10;-2;2;10];

% Calculates frequency shifts for each velocity
freqs = (1500 + relv)./(1500.0);

figure;
title(num2str(mfsk,"-%dFSK Simulated BER vs. SNR with Doppler Shift "));
xlabel("SNR (dB)");
ylabel("BER (errors/bit)");
set(gca, 'YScale', 'log');
legend;
hold on;

% Factor to interpolate signals by (for better frequency shift)
% Somewhat uncommon prime to reduce aliasing
interpFactor = 11;
interpNoise = interp(rawNoise, interpFactor);
for idx = 1:length(freqs)
    fso = comm.PhaseFrequencyOffset('FrequencyOffset', freqs(idx), 'SampleRate', 200e3);
    shiftedSignal = fso(interp(signal, interpFactor));
    sigRms = rms(real(shiftedSignal));
    noise = interpNoise * (sigRms/noiseRms); % Interpolate noise to match length
    for i = 1:size(snrRange, 2)
        snr = snrRange(i);
        noisy = shiftedSignal + noise / snr;
        E(i) = sum(demod(decimate(noisy, interpFactor)) ~= data);
    end
    E = E ./ (len * mfskbts);
    plot(snrDecibels, E, 'DisplayName', num2str(relv(idx), '%−d_m/s'));
end
hold off

clear shiftedSignal;
clear interpNoise;
clear noise;
clear noisy;
release(demod);

% Multipath + noise

% Create a Rician Channel
pathDelays = [0 0.001 0.004 0.006 0.007 0.0075 0.008 0.0085 0.009 0.0095]; % sec
pathGains = [-5 -8 -11 -15 -16 -17 -18 -19 -20]; % dB
rician = comm.RicianChannel('SampleRate', 200e3, 'PathDelays', pathDelays, 'AveragePathGains',
multiSignal = rician(signal);

% Add noise
sigRms = rms(real(multiSignal));
noise = rawNoise * (sigRms/noiseRms);

% Iterate through SNRs
for i = 1:size(snrRange, 2)
...
% Grab SNR
snr = snrRange(i);
noisy = multiSignal + noise / snr;

% Demodulate and count error
E(i) = sum(demod(noisy) ~= data);
end

% Adjust for length of data to get per-bit error
E = E ./ (len*mfskbits);

% Display the error rate data with the SNR values in decibels
figure;
semilogy(snrDecibels, E);
title(num2str(mfsk,"%dFSK Simulated BER vs. SNR with Multipath Propagation"));
xlabel("SNR (dB)");
ylabel("BER (errors/bit)");