# Sound-Based Morse Code Modem

Annie Chu, Sparsh Bansal, and Thomas Jagielski
Olin College of Engineering, Needham, MA
Email: {achu, sbansal, tjagielski}@olin.edu

## I. Abstract

This project explored Morse code, its common modulation keying methods, and how to build a Morse code modem over software. The project aimed to transmit and receive the text "hello" over Morse code. It was implemented in MATLAB with the Digital Signal Processing and Audio Toolboxes and utilized two computers (one to transmit and one to receive). By applying Audio Frequency-Shift Keying (AFSK) principles on the transmission side and extracting frequency, power, and time from a spectrogram on the receiving side, this project was able to send and receive "hello", as well as several other messages, over audio as Morse code.

## II. Introduction

Morse code is a method of encoding text characters into standard sets of dots and dashes, two signals of differing durations. Founded in the 1830s by Samuel Morse, Morse code in conjunction with the invention of the telegraph revolutionized long distance communication, allowing for messages to be sent quickly via wire over long distances using pulses of electrical current. Once received, these pulses were translated back into their corresponding text characters. In addition to this electrical medium, Morse code was also commonly transmitted over radio in the form of sound. Prior to the invention of the Morse code, people used drumbeats and smoke signals transmit critical information [6]. A popular, now standard, variant of Morse code is International Morse code, shown below in Figure 1.

FIGURE 1: International Morse Code [2]

Adopted in 1851, International Morse Code became the standard and was heavily used for marine communications by battleships and the shipping industry during World War II, as well as the Korean and Vietnam Wars. In this standard, a dash is three times the duration of a dot. A pause with the duration of a dot is used in-between dots and dashes. A pause with the duration of a dash indicates a new letter, and a pause with the duration of seven dots indicates a new word. [8] International Morse code transmitted over radio, also known as Continuous Wave (CW), remains popular in ham radio as the simplest transmission form found over many RF bands. CW utilizes On-Off Keying modulation (OOK) and is conducted via a keyer. (Figure 2).

FIGURE 2: Straight Morse Keyer (for transmission) [3]

There are several different variations of Morse code keyers, of which the straight keyer is most common. The straight keyer is composed of a vertical lever serving as an electromechanical switch to connect and disconnect from the transmitter, which could be radio or a simple telegraph wire. In its natural untouched state, the keyer's electrical connection is normally open; only when the lever is pushed down is the connection closed and turns on the transmitter. [9]

In CW, this on-off connection driven by the lever is basis of the modulation scheme OOK. Figure 3 describes the process of OOK modulation.

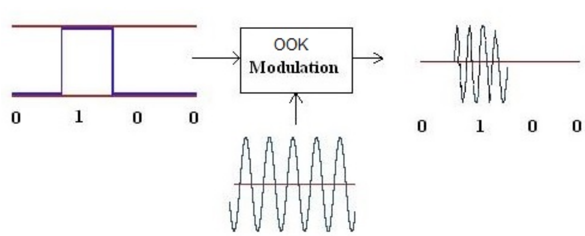


FIGURE 3: OOK Modulation [4]

In OOK modulation, the sinusoidal carrier signal is turned on and off with the keyer lever, representing binary states.

OOK Modulation is the most basic version of Amplitude Shift Keying (ASK), which is described in Figure 4.

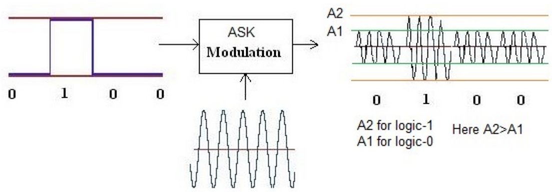


FIGURE 4: ASK Modulation [4]

As shown, ASK can generally be described with a logical 0 (LOW) represented by a lower amplitude of the carrier frequency and a logical 1 (HIGH) represented by a higher amplitude of the same carrier frequency. Some common applications of ASK include wireless base stations, tire pressure monitoring services, and digital communication links [7].

In addition to ASK, in which the amplitude is shifted, another modulation technique is frequency-shift keying (FSK), in which frequency of the carrier wave is shifted depending on its binary value (Figure 5).

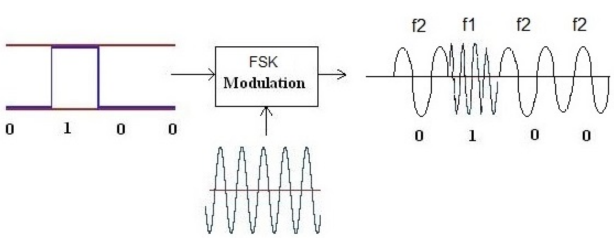


FIGURE 5: FSK Modulation [4]

FSK is advantageous over ASK regarding signal interference, but is more expensive to implement. This technique is widely used for applications such as garage door openers, telemetry for communication systems, and low-frequency radio transmission on the Very Low Frequency (VLF) and Extremely Low Frequency (ELF) bands. [4]

This project further explores the concept of audio frequency-shift keying (AFSK) - the digital data is encoded in binary using changes in the pitch of an audio tone. Relative to other forms of FSK, AFSK is usually not employed for high-speed data communications due to inefficiencies in power and bandwidth requirement. An important application of AFSK is the U.S.-based Emergency Alert System which is used to convey emergencies to affected locations without the need for transmitting plain text. Other applications include Weatheradio (NWR) by the National Oceanographic and Atmospheric Administration (NOAA) and shortwave radio transmission [1].

## III. METHODS

The MVP goal of this project was to transmit and receive a simple encoded message such as "hello" using two computers: one as the transmitter, the other as the receiver. This program was implemented in MATLAB utilizing the Digital Signal Processing and Audio Toolboxes. A block diagram describing this process is shown below in Figure 6.

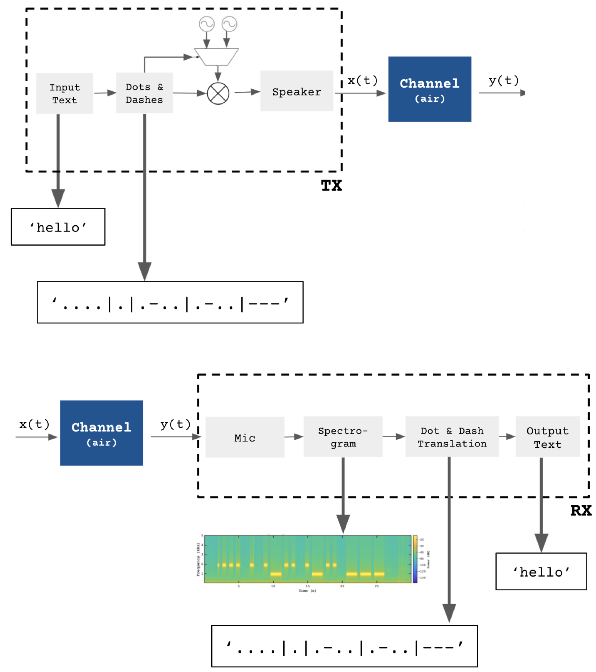


FIGURE 6: Full System Block Diagram Transmitting (top), Receiving (bottom)

Depicted in the TX-labeled box on the left above, the process of transmitting Morse code begins with an text input ("hello"). The text is then converted character by character to International Morse using the Rosetta Code helper script [5].

Once the text is encoded into a sequence of dots and dashes, AFSK is employed. The dots and dashes are split, and a dot is modulated to a carrier frequency of 2 kHz for 0.5s, a dash to 1 kHz for 1.5s. The two are then mixed together and transmitted auditorily through the speaker of the first computer.

The transmitted signal $x(t)$ travels through the air, which serves as the channel, and is received and recorded by microphone connected to a second computer.

Once received signal $y(t)$ has finished being recorded, the audio is analyzed for frequency content over time. Using MATLAB's spectrogram function, the power of signals at a given frequency with respect to time can be computed. The computed values are extracted and visualized as a spectrogram for the entirety of the message. An example spectrogram is shown below in Figure 7.

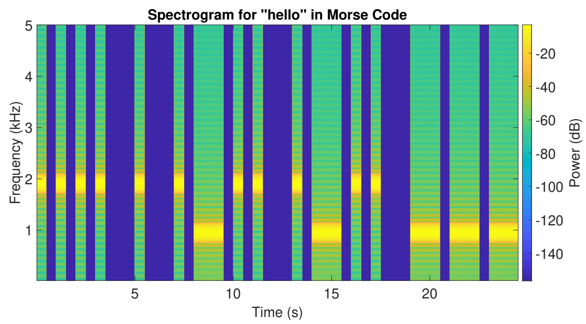


FIGURE 7: Spectrogram for received signal with ‘hello’ as the transmit signal.

The horizontal axis represents time in seconds, the vertical axis shows frequency in kHz, and the coloration depicts the power of the input signal in dB at a given frequency and time. When the color is lighter - closer to yellow - the strength of the signal is stronger. The short ‘dots’ and ‘dashes’ can be differentiated by yellow blobs at either 2 kHz or 1 kHz respectively.

The frequency with the most power at each timestep in the spectrogram is extracted to differentiate between a ‘dot’, ‘dash’, or character break. A visual of the extracted peaks is shown below in Figure 8.

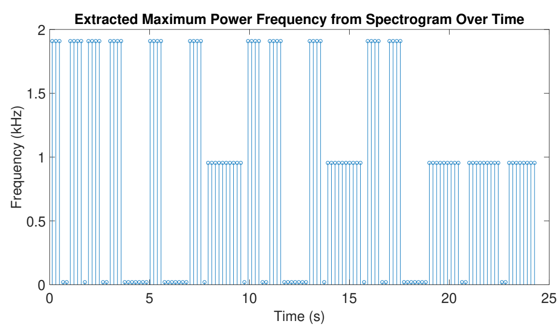


FIGURE 8: Extracted frequencies with maximum power over time with ‘hello’ as the input signal.

Similar to the spectrogram, the horizontal axis is time in seconds and the vertical axis shows frequency in kHz. The stem represents the frequency with the maximum power at that time within the signal. Considering both Figure 7 and 8, the stems align with the lighter frequencies contained in the spectrogram with respect to time.

The peaks visualized in Figure 8 are then processed to remove repetition in succeeding time samples. Firstly, peaks are assigned -1, 0, or 1 based on frequency thresholds. Samples with a peak greater than 1.5 kHz are assigned 1, peaks between 500 Hz and 1.5 kHz are assigned -1, and peaks less than 500 Hz are assigned 0. Duplicated symbols are eliminated with the requirement of no two subsequent points having the same -1 or 1 designator. Since 0 represents both a space between a new character in the original message and a new symbol in the ‘dot’-‘dash’ sequence, a counter is implemented to differentiate between the two. Given a sampling time window of 0.25s for the spectrogram, peaks are repeated with known replicate samples. The resulting sequence is converted to ‘dots’ and ‘dashes’ and the output message is translated using a Morse code to English dictionary.

## IV. RESULTS

Using over-the-air transmission between the two computers, “hello” was transmitted from the first to the second computer. Figure 7 depicts the sent message and the spectrogram of the received signal is shown in Figure 9.

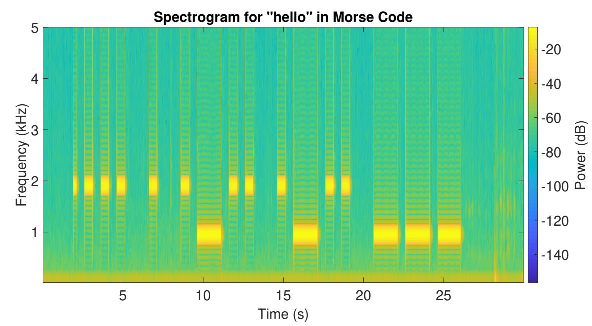


FIGURE 9: Spectrogram of “hello” received during over-the-air transmission.

With the same axis as Figure 7, Figure 9 indicates the presence of noise and signal impurities due to over-the-air transmission. Following the process described in the “Methods” section, the maximum power frequencies over time is shown in Figure 10.

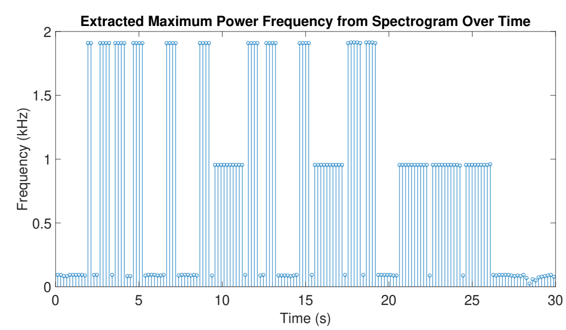


FIGURE 10: Extracted peaks from Figure 9 with respect to time.

With a horizontal axis of time in seconds and vertical axis of frequency in kHz, the stem at each timestep represents the maximum frequency present in the received signal during over-the-air transmission at a given time. With the noise present in the received spectrogram (Figure 9), the maximum power peaks filter the interference successfully such that the ‘dots’ and ‘dashes’ are distinguishable. The ‘dots’ and ‘dashes’ are converted to a received message of ‘hello’ with 100% accuracy. Other messages that were tested included “adcomms”, “happybirthday”, and “xylophone”, which were all received and translated with 100% accuracy.

The software testing was successful with a 100% accuracy rate consistently observed across all messages that were transmitted.

## V. Future Work

Morse code, in the nineteenth century when it was invented, was transmitted using mechanical keyers manually. This project implements that concept in software, which means there is more flexibility for controlling the data transmission rate. The challenge this incurs is that there is a certain limit to which the transmission speed would lead to no errors. This limit can be optimized by using different software techniques to minimize errors.

Secondly, this software could implement multi-word message decoding. Currently, a single word can be transmitted at a time without error, which is what Morse code was originally defined to be.

Lastly, Morse code uses a single frequency of sound for transmission. To simplify 'dot' versus 'dash' detection, the current approach used two separate frequencies to transmit either. A good next step would be to eliminate this simplification step and use one consistent pitch for transmitting 'dots' and 'dashes'.

## References

[1] Audio frequency-shift keying. URL https://academickids.com/encyclopedia/index.php/Audio_frequency-shift_keying.

[2] Morse code translator, decoder, alphabet. URL https://www.boxentriq.com/code-breaking/morse-code.

[3] Morse code telegraph transmitter, . URL https://kids.britannica.com/students/assembly/view/221683.

[4] Ook vs fsk vs ask, . URL https://www.rfwireless-world.com/Terminology/OOK-vs-FSK-vs-ASK.html.

[5] Rosetta code: Morse code. URL https://www.rosettacode.org/wiki/Morse_code.

[6] History of morse, 2013. URL https://nrich.maths.org/2198.

[7] Elprocus. Amplitude shift keying (ask) working and applications. URL https://www.elprocus.com/amplitude-shift-keying-ask-working-and-applications/.

[8] M. Mancini. How morse code works and still lives on in the digital age, 2021. URL https://science.howstuffworks.com/innovation/inventions/morse-code.htm.

[9] S. C. Phillips. Straight keys, bugs, paddles, sounders and keyers. URL https://morsecode.world/keys.html.

## VI. Appendix

The software files are available for viewing on a GitHub repository linked here.