EE430 Term Project

In this project, you will implement a Frequency-Hopping (FH) Spread Spectrum (FHSS) transmitter and receiver. The idea of spread spectrum is to spread the information signal over a wider bandwidth in order to make jamming and interception more difficult. Frequency-hopping spread spectrum is a method of transmitting radio signals by rapidly changing the carrier frequency among many frequencies occupying a large spectral band. The frequency band is divided into smaller sub-bands. Signals rapidly change ("hop") their carrier frequencies among the center frequencies of these sub-bands. Thus, interference at a specific frequency affects the signal only during a short interval.
FHSS offers four main advantages over a fixed-frequency transmission:

- FHSS signals are highly resistant to narrowband interference because the signal hops to a different frequency band.
- Signals are difficult to intercept if the frequency-hopping pattern is not known.
- Jamming is also difficult if the pattern is unknown; the signal can be jammed only for a single hopping period if the spreading sequence is unknown.
- FHSS transmissions can share a frequency band with many types of conventional transmissions with minimal mutual interference. FHSS signals add minimal interference to narrowband communications, and vice versa.

Mathematical representation of a frequency hopping signal is shown by Equation 1.

$$x(t) = \sum_k rect_{T_h}(t - kT_h) e^{j(2\pi(f_k + \Delta f m_k)(t-kT_h))}. \tag{1}$$

By using the signal model in Equation 1, you will implement a transmitter that sends digital data over FHSS, as well as a receiver that decodes the received signal. In Equation 1, $rect_{T_h}(t)$ is a rectangular signal with a width of T_h such that $rect_{T_h}(t) = 1, 0 \leq t \leq T_h$. T_h represents the 'hop period', which is the time spent in a single hop frequency, measured in seconds. f_k represents the 'hop frequencies'. Hop frequencies will be determined by selection from a table via pseudo-random manner (Details will be explained in the later sections). We use M-ary Frequency Shify Keying to modulate the digital data over frequency. Basically, FSK is a frequency modulation scheme in which digital information is encoded on a carrier signal by shifting the frequency of the carrier between M different frequencies. In Equation 1, Δf is the amount of shift to separate frequency determined by the digital data from the center frequency (hop frequency). This separation is controlled by digital data according to the m_k in Equation 1. Now, to clarify this, let's investigate the 2-FSK (binary), 4-FSK, and 8-FSK cases that you will work on this project:

1) 2-FSK: In this case, digital data can be in the set $\{0, 1\}$. Depending on the input digital data, our exponential in the Equation 1 becomes $e^{j(2\pi(f_k \pm \Delta f)(t-kT_h))}$. In other words, if the digital data is 0, $m_k = -1$; if the digital data is 1, $m_k = +1$.
2) 4-FSK: In this case, digital data can be in the set $\{00, 01, 10, 11\}$. We send two data bits in a single hop period. Similar to the previous case:
   - $00 \rightarrow m_k = -2$
   - $01 \rightarrow m_k = -1$
   - $10 \rightarrow m_k = +1$
   - $11 \rightarrow m_k = +2$
3) 8-FSK: This case can be found by understanding the previous cases.

I. TRANSMITTER

You must create a graphical user interface (GUI) for the transmitter side using MATLAB's App Designer. Through this GUI, user should be able to enter a text message and transmit a FH signal for communication. Then, GUI should display the spectrogram of the generated signal.

A. *Data generation*

For data generation, we will follow a module similar to the one shown in Figure 1. Figure 1 illustrates a widely used FHSS transmitter module. In MATLAB, we will generate pseudo-random numbers and use these numbers to select frequencies from a table. Next, you will form the signal model shown in Equation 1 **according to the input data**.

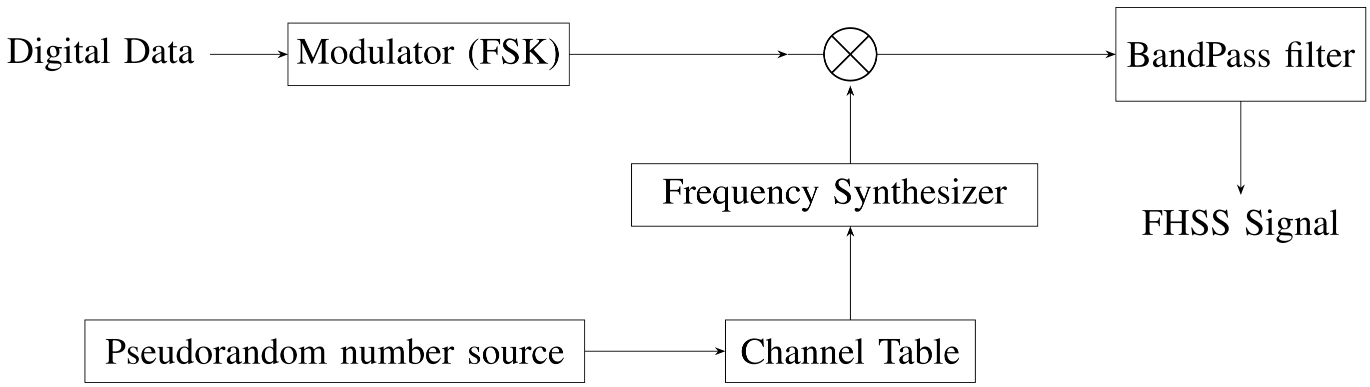

Fig. 1. Practical model of a FHSS transmitter

B. *Spectrogram*

After generating the signal, you will plot its spectrogram. A spectrogram is a color or greyscale plot of the magnitude of the short-time Fourier transform (STFT) on the time-frequency plane. One axis of the time-frequency plane represents time, and the other represents frequency. The third dimension is for the magnitude of the short-time Fourier transform. The STFT is a sequence of Fourier transforms of "short" consecutive segments of a "long" signal and can be used to analyze the temporal frequency content of the signal. Our textbook discusses the STFT and related topics in Chapter 10 entitled "Fourier Analysis Using Discrete Fourier Transform". In the textbook, STFT is also referred to as "Time Dependent Fourier Transform". You can read the first 4 sections of this chapter. **You can use MATLAB's built-in `stft` command**. You can choose another method to plot the spectrogram if you wish.

C. *Implementation of the transmitter*

For creating the transmitter, follow the steps below:

- In GUI, create an input field for entering a text message. This text message is the message that we want to transmit. In the background, encode this text message to digital data (Your receiver should know this rule to decode). According to the digital data and modulation order (M-FSK), you can obtain the m_k values.
- In this GUI, the sampling rate is crucial for generating the signal. Therefore, the user should be able to adjust the sampling rate through the GUI.

- Then, you need to decide hopping frequencies. For that, you will create a table such that this table will contain your hopping frequencies that the system will use during the communication. You can find the details of creating this table in the Section 'Frequency table'. Then, at each hop, you will 'choose' a frequency from the table. This 'choosing' will be determined by a pseudo-random number generator. Pseudo-random numbers are generated by deterministic algorithms. They are "random" in the sense that, on average, they pass statistical tests regarding their distribution and correlation. **You can use MATLAB's built-in functions to produce pseudo-random numbers. For this, you can set a seed and feed the seed with `rng` function. Then, you can use `randi` function to produce random numbers**. After generating the pseudo-random numbers, you will use these numbers as indices to choose frequencies (hop frequency) from the frequency table. This value is the f_k value in the Equation 1. Since pseudo-random numbers are generated by deterministic algorithms, the receiver should know this algorithm to follow the hopping frequency sequence to demodulate and decode the received signal. You can use the same specific seed in both transmitter and receiver to generate the same sequence. **Ensure that you feed the random number generator with the same seed before generating the sequence**.
- Now, there are some parameters the signal and communication should satisfy: T_h in the Equation 1 is the 'hop period' (the duration of a single hop-frequency occupies). 'Minimum frequency separation' is the **minimum frequency gap between hop frequencies ($\mathbf{f_k}$) of successive hops** in Hz. Δ_f is the frequency gap with center frequency in Hz. 'M-FSK' denotes the modulation order of FSK. 'Bandwidth' is the frequency range that your communication channel occupies in Hz. 'Minimum number of digital binary data' is a parameter that sets a threshold for the minimum length of the message (This depends on how you encode and decode your text message). We created some categories such that every category defines the parameter values that your signal and communication should obey. These categories are shown in Table I. Thus, you need to be able to create a signal and perform the communication for each of the categories. Hop period, M-FSK, and Δ_f should be adjustable in the GUI.
- In Table I, there is a parameter called 'Distance between transmitter-receiver'. This parameter will be used to adjust the Signal-to-Noise Ratio (SNR) level. It is the distance between two PCs in cm. Thus, you do not need to adjust anything to create a signal for this parameter (other than keeping the amplitude of the signal same).
- After generating the signal, you need to plot its spectrogram to the GUI. The user should be able to adjust the parameters of the STFT calculation if you use STFT to plot the spectrogram.
- After creating the signal, your GUI should be able to play it through the speakers of your PC. **Remember to adjust the sampling rate to a value suitable for audio signals**.

TABLE I
CATEGORY REQUIREMENTS FOR FH SIGNAL PARAMETERS

	Hop period (seconds)	M-FSK	Δ_f (Hz)	Minimum frequency separation (Hz)	Bandwidth (Hz)	Minimum number of digital binary data	Distance between transmitter-receiver (cm)
Category 1	1.0	2-FSK	100	500	4000	5	50
Category 2	0.75	4-FSK	150	1000	7000	20	100
Category 3	0.50	8-FSK	200	2000	10000	30	150

D. *Frequency table*

For each category in Table I, you will have a different frequency list from which you will select the hopping frequencies according to the pseudo-random number generator. In other words, to create a signal for 'category-i', you will use the 'row-i', which contains the corresponding frequencies. These frequencies will be determined based on your student numbers. Let's denote the student number of one of the group members as $N_1N_2N_3N_4N_5N_6N_7$.

- **Category 1**: 'row 1'$=[1000{+}bS,\ 1500{+}bS,\ 2000{+}bS,\ 2500{+}bS,\ 3000{+}bS,\ 3500{+}bS,\ 4000{+}bS,\ 4500{+}bS,\ 5000{+}bS]$ where $b = 1$ if N_6 is even, $b = -1$ otherwise. $S = 100 \cdot (N_7\%5)$, where $\%$ is the modulo operation.
- **Category 2**: 'row 2'$= [1500 + bS,\ 2500 + bS,\ 3500 + bS,\ 4500 + bS,\ 5500 + bS,\ 6500 + bS,\ 7500 + bS,\ 8500 + bS]$ where $b = 1$ if N_5 is even, $b = -1$ otherwise. $S = 100 \cdot N_7$.
- **Category 3**: 'row 3'$= [1000 + S,\ 3000 + S,\ 5000 + S,\ 7000 + S,\ 9000 + S,\ 11000 + S]$ where $S = 100 \cdot N_7$.

Please note that, due to the predetermined categories, the 'bandwidth' and 'minimum frequency separation' requirements are already satisfied. They are included in Table I to outline the general requirements.

Example 1. *Let the student number be* 1234567. *Then, the rows will be:*

- *'row 1'*$=$ [1200, 1700, 2200, 2700, 3200, 3700, 4200, 4700, 5200].
- *'row 2'*$=$ [800, 1800, 2800, 3800, 4800, 5800, 6800, 7800].
- *'row 3'*$=$ [1700, 3700, 5700, 7700, 9700, 11700].

II. RECEIVER

For implementing the receiver, you will design another GUI. The role of this GUI will be recording the audio signal generated by the transmitter, plotting its spectrogram, and decoding the message embedded in the FH signal. One important note before processing the received signal is that it will not be exactly the same as the signal transmitted by the transmitter due to noise and environmental effects. You might need to apply some signal processing techniques to reduce the noise effects. One of the simplest methods is to use a bandpass filter so that frequencies out-of interest will be removed. You can apply other techniques as needed.

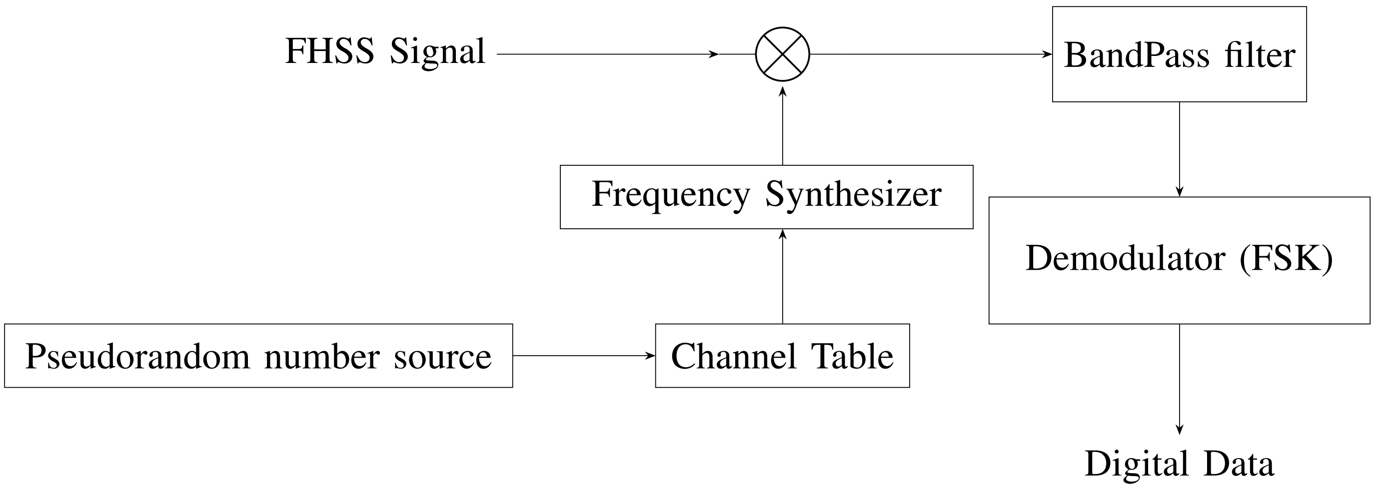

Fig. 2. Practical model of a FHSS receiver

Figure 2 shows a pratical receiver for FHSS communication. To implement a receiver in MATLAB, you can follow the steps below:

- You need to create another GUI for the receiver. This GUI should be able to record audio signals using the microphone of your PC. The user should be able to adjust both the sampling rate and the duration of the recording.
- Please adjust the distance between two PCs according to the Table I.
- After recording the received signal, you will plot its spectrogram to the GUI as you did for the transmitter. If you apply any techniques to reduce noise, you should also display the spectrogram after implementing those techniques.
- Now, you need to decode the received signal to retrieve the message. First, the receiver must know the encoding-decoding rules between the text message and the digital data. Moreover, the receiver should be familiar with the deterministic algorithm used to produce pseudo-random numbers and the table containing the hop frequencies. It is crucial for the transmitter and receiver to be synchronized so that the receiver can accurately follow the hop frequencies in the signal. There are several ways to achieve synchronization. One method is to send a pilot signal with a specific frequency before the actual transmission, allowing the receiver to recognize that the transmission is about to begin. The receiver can listen for the frequency of the pilot signal and start its internal clock to synchronize with the transmitter. However, even without synchronization, you can track the hop frequency by detecting the current frequency of the signal. By analyzing the frequency changes in the signal, you can determine which hop the signal is in. Most methods require the detection of the signal's frequency, so you can utilize the time-frequency analysis applied in the spectrogram. There are various techniques for estimating the frequency of a frequency-hopping signal using the time-frequency analysis.
- Now, you need to demodulate the FSK-modulated signal. There are several methods for demodulating the received FSK signal. One approach you can apply for this project is to detect the current frequency of the signal. Since the current hopping frequency is known to the receiver (previous step), subtracting the hopping frequency from the detected frequency will give you the frequency difference determined by the digital data (this also depends on the modulation order).
- Based on the digital data, the text message can be decoded and displayed in the GUI.

III. PROJECT PART-1

For the Part-1 of the project you need to implement all of the parts described above except that you do not need to implement decoding of 'category 2' and 'category 3' signals. Below, you can find the details of deliverables:

- In your report, you must explain the GUIs in detail and demonstrate their functionality by providing relevant screenshots and figures. For the spectrogram section, you should include the spectrograms of the FH signals for three different categories, both from the transmitter side and the receiver side.
- You need to explain the method you used for plotting the spectrogram. Also, you should describe the parameters of the time-frequency analysis and how they affect the plot and analysis, providing mathematical reasoning to support your explanations.
- Please explain the method you used for encoding and decoding the pairs of text messages and digital data.
- Please demonstrate that your transmitter can produce signals that meet the requirements of each category by providing spectrograms, explanations, and relevant details. Note that the 'Distance between transmitter and receiver' pertains to receiving, so there is no need to prove anything related to that for signal generation.
- On the receiver side, please plot the spectrograms of the signals that you used to prepare the previous sections of the report. Ensure to provide the spectrograms specifically for the receiver side signals.
- Please explain in detail the methods you used to reduce noise (including why these methods work and their limitations, if applicable). Also, provide the spectrogram of the received signal after applying the noise reduction methods.
- Please explain how you keep track of the hopping frequency.
- Please explain how you demodulate the signal in detail (This is the part that you explain how you find the $\Delta f m_k$).
- Please demonstrate that your receiver can decode the received signal while satisfying the requirements of 'Category 1' and successfully obtain the transmitted message.

Prepare a descriptive and clearly written report, ensuring clarity in both language and format. You will upload this report, along with the MATLAB codes, to ODTUClass. Include mathematical explanations where necessary. If you are unable to obtain a functioning receiver, please discuss the methods for decoding and demodulating the signal as described in the deliverables, detail any problems you encountered, and propose potential solutions.

IV. Example Case

As a reference, we provide an example case of FH signal that satisfies the 'category 1' requirements. In this example, we have a message of 5 bits, 10110. Let the student number be 1234567, which gives us the frequency list from 'row 1' in Example 1. Assume that the pseudo-random number generator produced the following sequence: 29464. This sequence is used to select frequencies from the frequency list. Thus, we use them as indices such that first element's index in the frequency list is 1. Figure 3 shows the signals produced during each hop. These signals should be generated by sampling in the MATLAB environment. The red-colored f_i values indicate the 'hop frequencies' selected from the table. Thus, $f_1 = 1700$, $f_2 = 5200$, $f_3 = 2700$, $f_4 = 3700$, and $f_5 = 2700$.

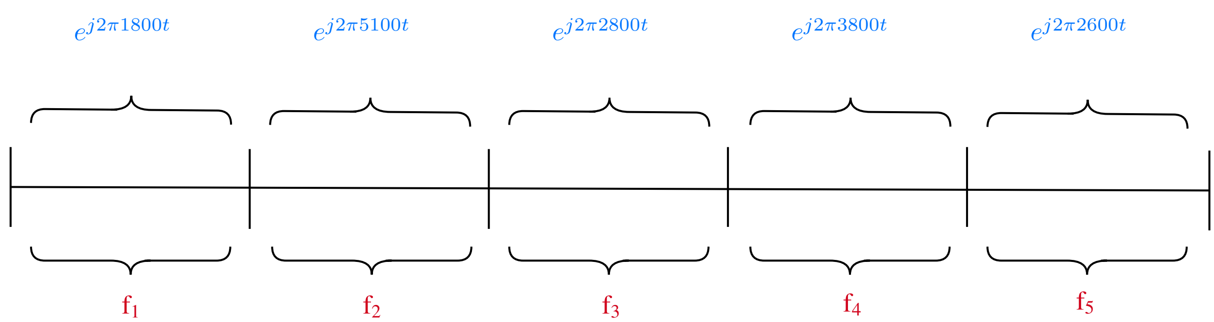

Fig. 3. Hop periods and signals produced in the hop periods

Figure 4 shows the spectrogram of the signal produced in this example.

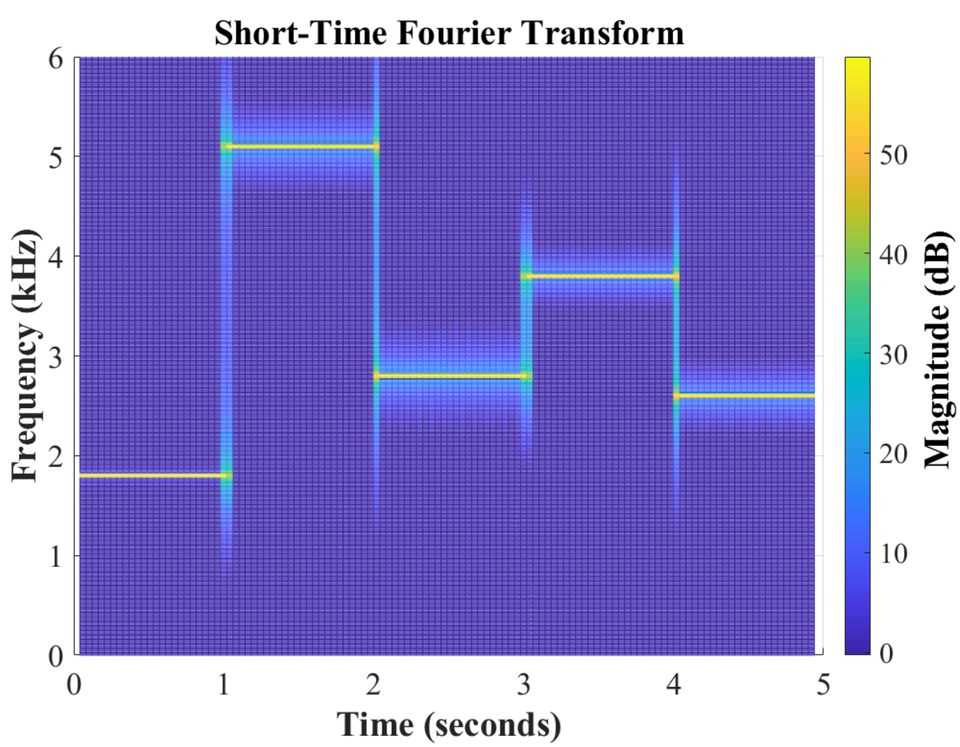

Fig. 4. Spectrogram of a signal in category 1.

REFERENCES

[1] Frequency-hopping spread spectrum. https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum.
[2] W. Stallings, T. (2003). Data and Computer Communications (5th ed.). Pearson.
[3] Y. Li, X. Guo, F. Yu and Q. Sun, "A New Parameter Estimation Method for Frequency Hopping Signals," 2018 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), Boston, MA, USA, 2018, pp. 51-52, doi: 10.1109/USNC-URSI.2018.8602599.
[4] S. Barbarossa and A. Scaglione, "Parameter estimation of spread spectrum frequency-hopping signals using time-frequency distributions," First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications, Paris, France, 1997, pp. 213-216, doi: 10.1109/SPAWC.1997.630288.