EE 430 Term Project, Part 2

In this part of the project, you are going to experiment with dual-tone multi-frequency (DTMF) signaling. More specifically, you are going to implement signal processing methods in MATLAB to generate, transmit, receive, and decode audio DTMF signals. Guidelines are given below.

Prepare a descriptive and clearly written (in language and format) report. You are going to upload your report and MATLAB codes to ODTUClass.

1- Background

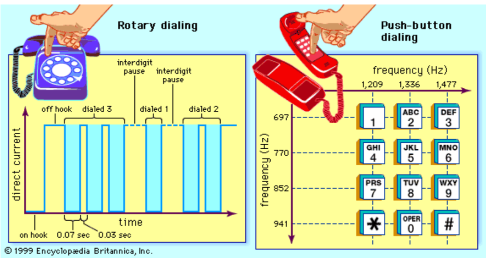
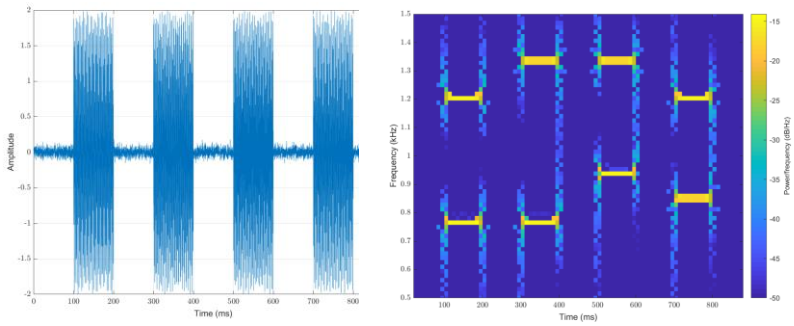


Figure 1. Rotary dialing and push-button dialing.

Signaling is the exchange of information via the use of signals between network devices to manage communication sessions. Some examples of signaling are addressing, such as dialing the phone number of the called party, and controlling automated equipment, such as navigating on the provider's menu.

When a user dials a phone number, they start a signaling process, where the dialed digits are transmitted to the telecommunications networks and are used to route the call to the correct destination. Between the 1950s and 1960s, the rotary phone was considered as the standard for telephone communications. On rotary phones, the address of the called party was signaled to the switching office via pulse dialing, where each dialed digit was coded by the interruptions in the current flow proportional to the rotation of the dial, see Fig. 1. However, as each digit is represented by a sequence of pulses, the rotary dialing was slow and was not efficient. Furthermore, although it worked fine for point-to-point telephone calls, it did not allow for much else. To address these issues, AT&T (a telecommunications company) developed the push-button dialing system that utilizes dual-tone multi-frequency (DTMF) signaling. By the 1960s, rotary dials gradually lost their prominence and were eventually replaced with push-button dialing.

On DTMF signaling, each key is encoded using 8 tones. These 8 tones are divided into 4 high and 4 low-frequency groups, as illustrated in Fig.1. Accordingly, for each push of a button, a signal as a superposition of two constant frequencies (tones) is generated. This process is repeated for each key press until the input sequence is fully encoded. An example DTMF signal for the sequence [4,5,0,7] is shown in Fig.2.



(a) Time domain **(b)** Spectrogram

Figure 2: Signaling [4,5,0,7] with DTMF (100ms/100ms)

Accordingly, one can express a DTMF-encoded key as follows,

$$s^{(k)}(t;T_d) = \left(sin\left(f_L^{(k)}t\right) + sin\left(f_H^{(k)}t\right)\right) \cdot \left(u(t) - u(t-T_d)\right) \qquad (1)$$

where s is time domain signal corresponding to the pressed key, the superscript k represents the index of that key, $f_L^{(k)}$ and $f_H^{(k)}$ are the respective low and high frequencies from the table in Fig.1 and T_d is the signaling duration per key. Then, given a discrete sequence of numbers $x[n] = [x_0, x_1, \dots, x_{N-1}]$ (such as your phone number) one can express the DTMF-encoded squence, m, as,

$$m(t; T_d, T_r) = \sum_{k=0}^{N-1} s^{(x[k])}(t - k(T_d + T_r)\,;\, T_d) \qquad (2)$$

where T_r is the resting duration between two consecutively pressed keys. For example, in Fig. 2 both T_d and T_r are set as 100ms and $x[n] = [4, 5, 0, 7]$.

2- Deliverables

For all of the following items, you are required to write your own MATLAB codes and also must appropriately design a graphical user interface (GUI) using the app designer.

Your GUI must include the following items;

a) Transmitter Panel
   i. A push-button keypad,
   ii. A display field to show the pressed keys as text,
   iii. A reset button to clear the pressed keys,
   iv. Input fields for signaling-duration-per-key (T_d) and resting duration (T_r).
   v. An input field to adjust the amplitude of the time domain signal.
   vi. A time domain signal panel to display the DTMF-encoded squence, $m(t; T_d, T_r)$.
   vii. Display of the spectrogram of the generated signal $m(t; T_d, T_r)$.
   viii. Save and play buttons for saving and playing the generated signal $m(t; T_d, T_r)$ as audio.

b) Receiver Panel
   i. Input fields for signaling-duration-per-key (T_d) and resting duration (T_r).
   ii. Start/stop listening button,
   iii. A display to plot the received time domain signal,
   iv. A display to plot its spectrogram,
   v. A switch to select the decoding algorithm,
   vi. A display field to show the decoded signal,

You are required to write the program for both the transmit and the receive end:

- On the transmitter side, the program must be able to generate arbitrary length DTMF encoded signals which will be input via a push button keypad. The program must be able to display the input sequence as a text, its encoding as a time domain signal, and its frequency content as a spectrogram. Moreover, the user must be able to adjust the parameters such as the amplitude of the time domain signal, T_d, T_r and the parameters of the spectrogram. If needed, the generated signal must be able to be played or saved as audio.
- On the receiver side, the program must be able to listen (not necessarily record) via a microphone. The received audio signal, its spectrogram, and the decoded signal must also be displayed to the user (*optionally in real-time).* To decode the received signal, the user must be able to select between a spectrogram-based method and another algorithm of your choice (see the section 'Further Reading'). Moreover, the user must be able to adjust the parameters such as T_d, T_r from the receiver side as well. (Note: For real-time displays you can plot the signals at a separate figure from the GUI. You can also create checkbox elements to enable/disable plotting in order to balance the load on processing if needed.)

On your report, explain the GUI of your application thoroughly, by providing screenshots whenever necessary. You must demonstrate that you deliver all the requirements. Explain and summarize the different approaches for estimating and decoding DTMF signals in detail. Also, further elaborate on the two algorithmic approaches you implemented, and demonstrate their performance by using your student numbers. That is, for the transmitter side, you must display original the time domain signals corresponding to your student numbers and their original spectrograms. Similarly for the receiver side you must display the received time domain signals and their spectrogram along with the decoded text/digits. You must also compare the two algorithms and compare their strengths and weaknesses. For your trials, you are required to at least test the performance for $(T_d, T_r) \in \{(250ms, 250ms), (100ms, 100ms), (40ms, 40ms)\}$. You must discuss your results and must provide all the necessary figures. You must comment on the algorithm performances for different (T_d, T_r) pairs.

3- Project Demonstration

In the project demonstration (which will be held face-to-face), a user will test your program to check if it satisfies all the set requirements.

4- Further Reading & Useful Links

- **DTMF:** https://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling
- **Signalling:** https://en.wikipedia.org/wiki/Signaling_(telecommunications)

- **Filter Banks:** https://en.wikipedia.org/wiki/Filter_bank
- **Goertzel algorithm:** https://en.wikipedia.org/wiki/Goertzel_algorithm
- **MUSIC algorithm:** https://en.wikipedia.org/wiki/MUSIC_(algorithm)

Useful MATLAB Links:

- **Real Time Audio:** https://www.mathworks.com/help/audio/gs/real-time-audio-in-matlab.html
- **Audio I/O Buffering**: https://www.mathworks.com/help/audio/gs/audio-io-buffering-latency-and-throughput.html

Disclaimer about the codes in MATLAB/Online!

Although there are MATLAB codes for DTMF, Filter banks, the Goertzel algorithm, and the various forms of MUSIC algorithm (as exemplified below), you are required to write your own code whenever possible. Therefore, you are not allowed to use prewritten high-level code. However, you are allowed to use low-level functions such as fft(), dct(), filter() etc. If you are in doubt about whether to use a built-in function of MATLAB you can ask via e-mail (ookyanus@metu.edu.tr).

- **DTMF Generator and Receiver:** https://www.mathworks.com/help/dsp/ug/dtmf-generator-and-receiver.html
- **Filter Banks:** https://www.mathworks.com/help/dsp/ug/overview-of-filter-banks.html
- **Goertzel algorithm:** https://www.mathworks.com/help/signal/ref/goertzel.html