Digital Signal Processing (DSP) Project

Name: *Rekha Lokesh*

Roll No: *19EC10052*

Keypad GUI - DTMF (Dual Tone Multifrequency) coder/decoder

This is an addendum to the DTMF experiment in Digital Signal Processing Laboratory. The self-project is about designing a ***Keypad GUI*** in ***MATLAB*** to achieve the given goals.

Goals: The aim of the experiment is:

- Using ***DTMF technique to encode the keys*** {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D}.
- Design a ***bandpass FIR filter bank*** and ***decode the keys*** from the signal sent and analyze the noise performance and filtering.

Theory:

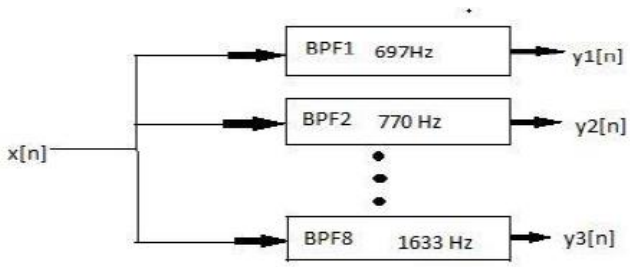
- Dual-tone multi-frequency signalling (***DTMF***) is a ***telecommunication signalling system*** using the voice-frequency band over telephone lines between telephone equipment and other communications devices and switching centres.
- The landline communication channel used in ***primitive telephone communication***, was ***analog*** in nature. For establishing a proper communication between the caller and the callee, the telephone exchange office had ***to decode the dialled digital number*** from analog information. Thus DTMF (Dual to Multi-Frequency) was used. If any key is dialled, a dual tone sinusoidal signal containing ***two distinct frequencies below and above 1000Hz*** is generated and transmitted over the telephone channel.
- The ***DTMF Encoding Scheme*** used for the keypad:

Hz	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

- Thus, based on the sequence of keys pressed, ***sinusoidal signals with frequencies corresponding to the keys pressed***, length of each bit or time segment being noted is ***generated***.

Decoding the DTMF Signals:

- ***FIR filter bank*** is used to ***decode DTMF signals***. The filter bank consists of ***8 BPFs***, each of which are designed to pass only one frequency component among the DTMF frequencies. ***The input signal for all filters is the same DTMF signal***.
- For a particular time-length L (which is priorly known in this case), when input to the filter bank is DTMF signal, ***two outputs will have relatively higher magnitude of frequencies from all the BPFs*** or in case of noise corrupted signals, two cases have dominant rms values of the output from the filter bank and ***those frequencies*** are ***used as row & column pointers*** to determine the ***key from DTMF code***. A filter bank consisting of 8 BPFs is shown below:



- ***Bandpass Filter Design:***

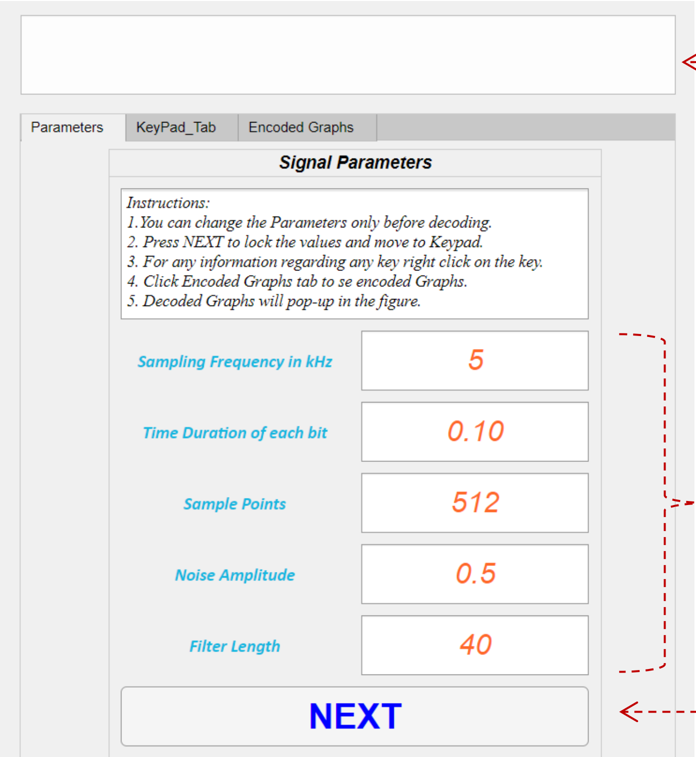
  The L-point average filter is a low pass filter with bandwidth inversely proportional to L (length of the filter). The BPF with L-points, gain in pass band and centre frequency w_c is defined by:

  $$h[n] = \beta \cos(\omega_c n) \quad \text{where } 0 \leq n < L$$

  β is chosen such that maximum value of the frequency response magnitude will be one.

Insight into GUI Design

- This is a GUI designed in MATLAB which accepts the input parameters, key sequence as inputs, and displays decoded key, encoded and decoded signals.



Main Screen

Displays the Final Result

When, the app or code is run, this tab appears. This is the parameters tab.

Parameters

These are the default values. You can edit the values before giving input or directly move to input

NEXT

Click to save the above values if changed before encoding

Clear, Back and Enter buttons convey their regular meanings. For any other info, right click on those buttons

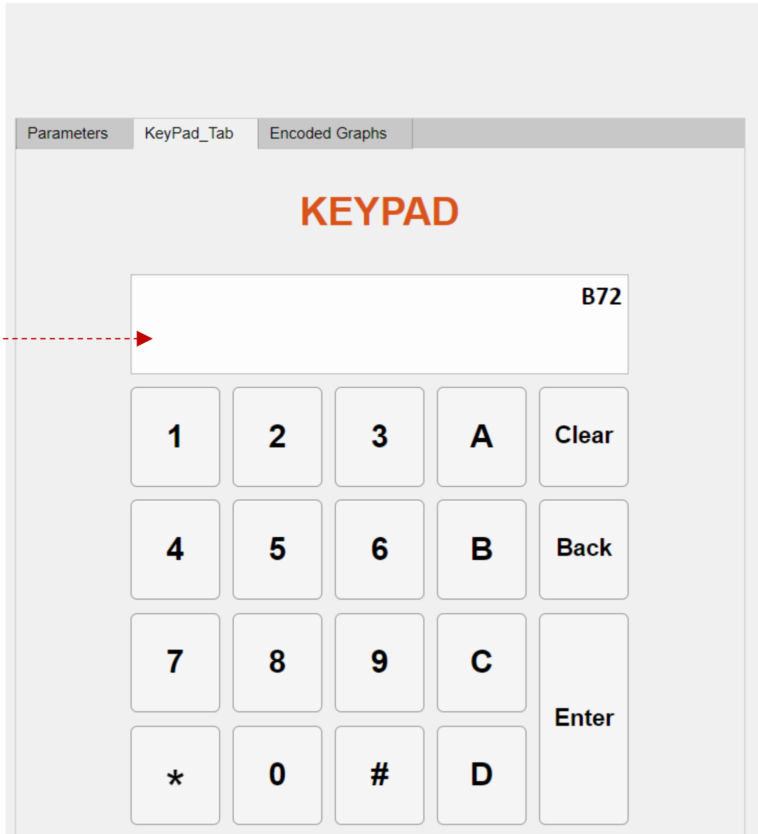
*Click **Enter** to start decoding at the other end*

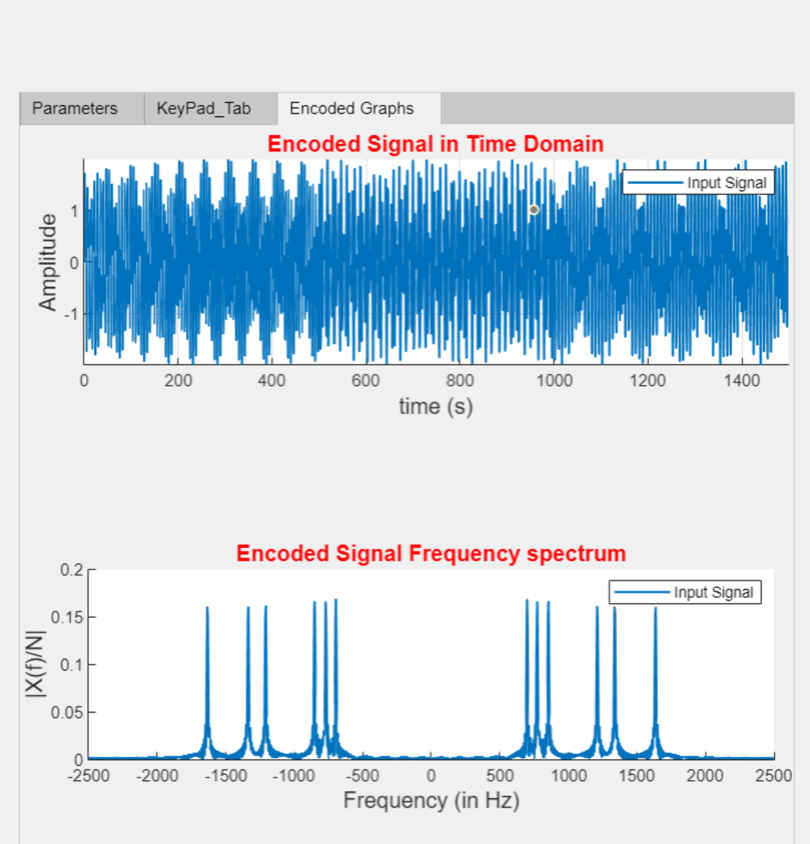
Keypad Screen

This is the KEYPAD tab. Used to give input

*The Digits 0-9, A-D, * and # are the keys to be pressed*

The entered keys are displayed on the keypad-screen simultaneously as pressed (with some time lag)





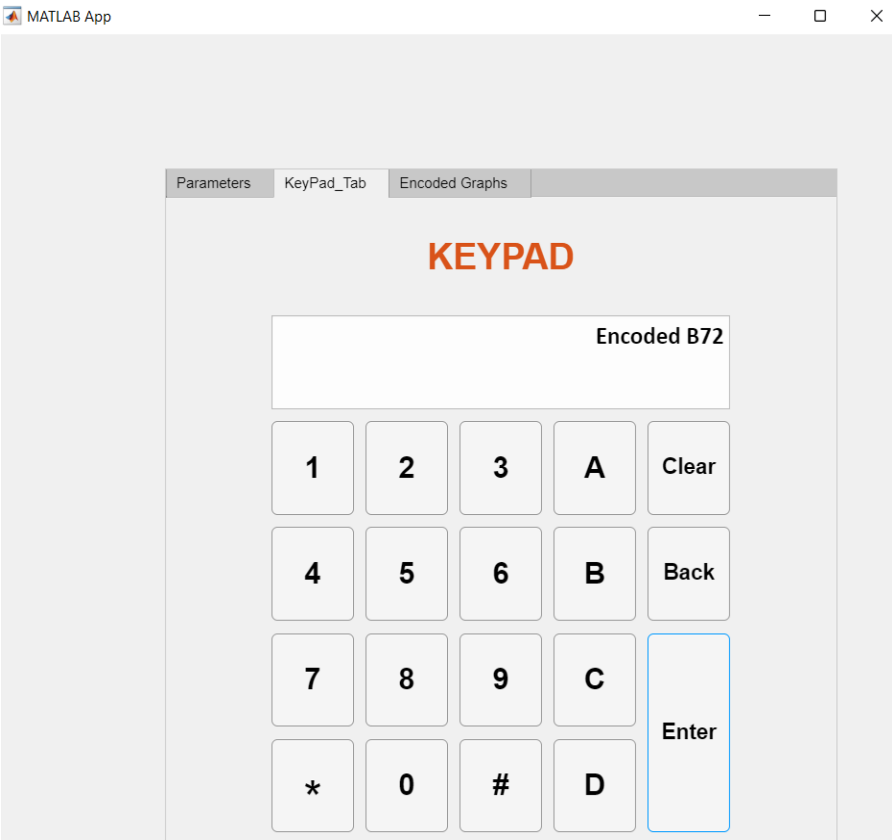
As the input is being given in the keypad tab, the corresponding encoded signals will be plotted here simultaneously. These change with every key pressed

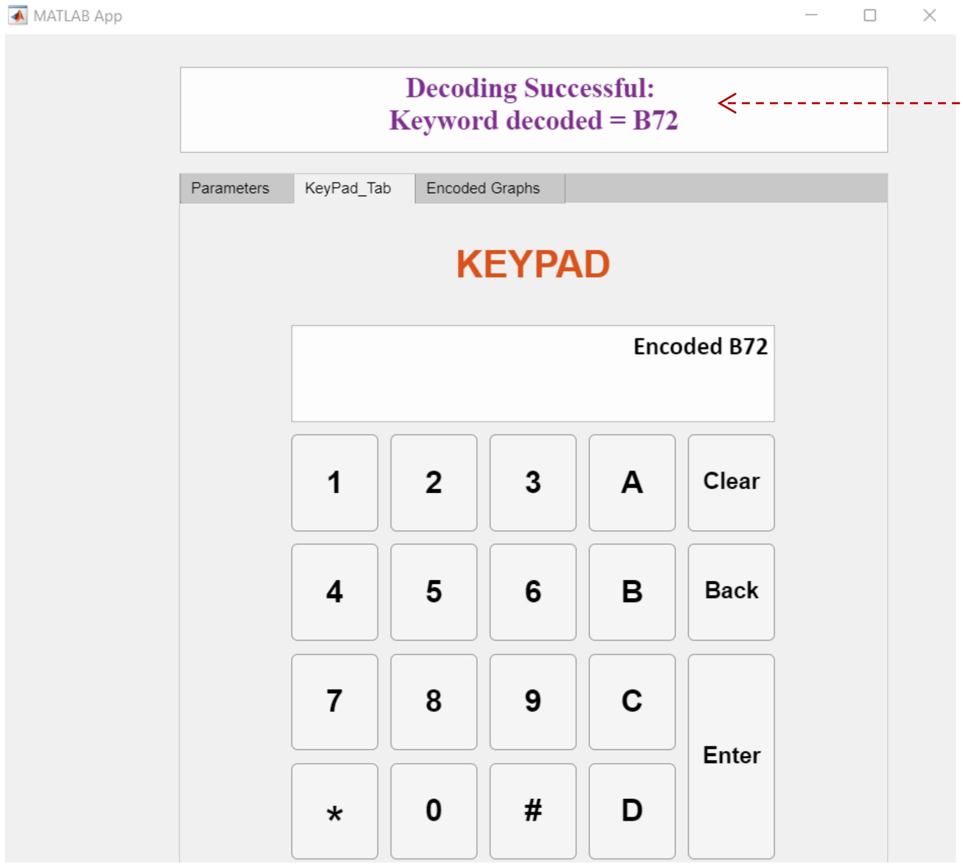
At any time, you can switch between the tabs

*When pressed **Enter,** the program goes to the decoding side, and it starts decoding key by key and the relevant plots will start popping up in the new page.*

Note: *For a sequence having n keys, there will be (4×n+1) decoding related plot figures with each figure having 9 subplots*

For each key, there will be four graphs related to time, frequency, noise corrupted time domain and noise corrupted frequency domain and in each case, there are 8 subplots: input signal segment and the 8 bandpass filter outputs





Final Result

After decoding completely, final result will be displayed on the Main Screen.

*To try again, Click **Clear** and start giving input.*

*The parameter values are locked when clicked **NEXT** and can be unlocked by clicking **Clear** after which parameter values can be changed*

- For the above entered sequence ***B72,*** the obtained plot figures are made available here due to space constraints.
- The above app can be run inside the MATLAB app section or run the corresponding MATLAB code (run like any .m script file) whose links are KEYPAD_APP and *KEYPAD_APP_exported respectively.*

Note: The above mentioned are the drive links to the files and plot figures and all the links to the files are made available in the *Reference* section at last.

Discussion:

- ***DTMF*** is a signalling system used for identifying the keys or the number dialled on a pushbutton or DTMF keypad. The early telephone systems used pulse dialling or loop disconnect signalling. This was replaced by multi frequency (MF) dialling. ***DTMF is a multi-frequency tone dialling system*** used by the push button keypads in telephone and mobile sets to convey the number or key dialled by the caller. DTMF has enabled the long-distance signalling of dialled numbers in voice frequency range over telephone lines. This has eliminated the need of telecom operator between the caller and the callee and evolved automated dialling in the telephone switching centres.
- ***DTMF*** (Dual tone multi frequency) as the name suggests uses a ***combination of two sine wave tones*** to represent a key. These tones are called row and column frequencies as they correspond to the layout of a telephone keypad.
- The frequencies used are 697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1336 Hz, 1477Hz, and 1633 Hz. The ***frequencies were carefully chosen*** in such a way as to ***prevent harmonics***. ***No frequency is a multiple of another*** and the ***difference or sum between any two frequencies is not equal*** to any other frequency. In practical systems, the higher frequencies are transmitted at 3 dB louder to compensate for any high frequency roll-off in the channel.
- The signal thus generated corresponding to the input sequence of keys is analyzed ***window by window (bit by bit or time segment by segment)*** where in for each segment, ***maximum frequency response and maximum rms value*** of the output of the filter banks is noted and the ***maximum two of those*** are ***detected to decode the key***.
- There are several advantages to this scheme. One is that by using this scheme ***we can encode 16 keys by only 8 frequencies***. If we had used a single frequency to encode each key, then we would have needed 16 frequencies and hence, ***16 FIR filters leading to large overheads and more resolution***. Also, the frequencies chosen for DTMF tones cannot be interpreted by the human ear but can be easily decoded by a phone system and computer. Thus, sensitive information can be isolated from the agents as well as from call recording systems.
- The ***FIR Band Pass*** filter design is ***derived from the FIR Ideal pass*** by ***multiplying it with cosine pulse*** and ***rectangular windowing***. Ideally, it should be of infinite length but due to practical constraints, it is truncated using a rectangular window of length filt_len. ***As filt_len decreases***, bandwidth increases which allows adjacent frequencies to pass and hence may lead to ***incorrect decoding***.

- In the given scheme, there are ***5 main parameters which can be varied***. They are ***time segment length (L), Sampling frequency (F_s), Number of Sample Points (N), Noise amplitude (noise_amp) and filter length(filt_len)*** whose optimal values and reasons are discussed in the above observations.
- In my design, it is better to choose ***values of L and F_s*** such that the ***value of $L \times F_s$ is an integer*** since its ***rounded off value is used as an index of a vector***.
- Choosing the ***number of sample points (N) as power of 2 yields faster results*** as ***computing fft will be faster*** in those cases. (Since radix 2 fft algos will be faster)
- In ***most practical*** cases or ***updated cases***, ***prior knowledge*** of the ***time segment length of each key*** will ***not*** be ***available*** at the ***decoding side***, or the length may keep changing. In those cases, a ***change in key*** is ***identified*** by ***detecting the white noise*** present between the ***time gap of*** pressing ***two keys*** where ***no DTMF frequency*** is present.

For Further Reference:

GITHUB Link – *Link to the report, MATLAB Code*

DRIVE LINK - *Link to my plots, figures, reports, script files and app related to the experiment.*

Additional DOC – *contains the MATLAB code written in doc*

MATLAB Code script – *MATLAB script file .m extension*