Mini-Project 5

DTMF Decoder

5.1 Synopsis

Listen to the sound of your telephone keypad when you dial a number. These distinctive sounds known as *DTMF tones* provide a way for your phone to communicate dialing information to the telephone switching system. No doubt you are familiar with the numbers 0 to 9 and the asterisk and pound keys, however the DTMF standard defines four additional keys to make a total of 16 unique tones.

In this project you will create a complete DTMF decoder system based on FFT spectral analysis. Your decoder system will accept an audio recording of a DTMF tone sequence as input and display the detected tones on a grid of visual indicators.

Ulaby/Yagle Sections 8.6–8.7

5.2 Objectives

1. Interpret a signal's magnitude spectrum to search for periodic signals

2. Investigate windowing as a means to reduce spectral leakage

3. Process audio signals and work with audio files

4. Implement an edge detector to automatically identify the location of DTMF tone bursts in an audio waveform

5.3 Deliverables

1. Hardcopy of all LabVIEW block diagrams and front panels that you create
2. Screen shots of requested spectrum/time plots
3. Lab report or notebook formatted according to instructor's requirements

5.4 Required Resources

1. NI LabVIEW with MathScript RT Module
2. Audio file of a mystery DTMF digit sequence provided by instructor

5.5 Preparation

DTMF decoder block diagram

Figure 5.1 shows the high-level block diagram of the DTMF decoder system that you will create in this project. The decoder loads an audio file that contains a DTMF tone sequence, extracts an audio subrange for a single tone, calculates the discrete Fourier transform (DFT) magnitude spectrum for the tone, and decodes the spectral information to determine the corresponding DTMF digit.

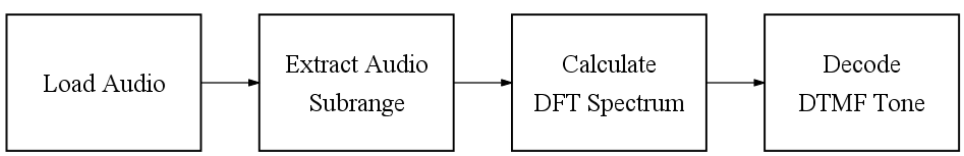

Figure 5.1: High-level block diagram DTMF decoder system.

DTMF tones

Learn about DTMF tones and create your own audio test sequence.

☐$_1$ Do an Internet search to determine the following information about DTMF tones:

1. Meaning of the acronym "DTMF,"
2. The year in which telephones first began to use DTMF,
3. Labels of the 16-key DTMF keypad,
4. Frequencies of the keypad rows, and
5. Frequencies of the keypad columns.

Include URLs for the articles from which you obtained this information.

► Install the Audacity software available at `http://audacity.sourceforge.net`. Run Audacity to create a new project. Set the "Project Rate" (system sampling frequency) to 8 kHz; see the lower left corner of the application. Select the menu command `Generate | DTMF Tones` and then enter the following information into the dialog box:

- "DTMF Sequence" = `123A456B789C*0#D`,
- "Amplitude" = 0.8,
- "Duration" = 5 seconds, and
- "Tone/silence ratio" = 50% (you will see a tone duration of 161 ms with these settings).

Select `File | Export` to save the audio as a WAV 16-bit audio file.

► Alternatively, search for a web service that converts the DTMF digits and letters to a .wav audio file, e.g., `http://www.dialabc.com/sound/generate`. Remember to select 8 kHz for the sampling frequency.

DFT-based spectral analysis

In this project you will use DFT spectral analysis as the heart of your DTMF decoder.

► Study Sections 8-6 and 8-7 in your text.

▶ Study the tutorial video at `http://youtu.be/z7X6jgFnB6Y` for an overview of DFT spectral analysis.

▶ Study the tutorial video at `http://youtu.be/UGUceuhSuUE` to learn more about the relationship between the time-domain waveform and its frequency-domain spectrum computed by the *fast Fourier transform* (FFT) algorithm. Note that the terms "DFT" and "FFT" are often used interchangeably.

5.6 Spectral Analysis

In this section you will analyze individual DTMF tones to better understand how to design your DTMF decoder.

▶ Create an empty top-level VI to serve as the starting point of your DTMF decoder application.

▶ Follow along with the video `http://youtu.be/ChSVf9lZ44k` to learn how to read a .wav audio file, extract the 1-D signal array, and play the audio to your soundcard. Use the DTMF test sequence audio you created earlier.

▶ Add block diagram code to calculate the audio file length in seconds and the sampling rate in Hz, and display these values on the front panel.

▶ Create a waveform indicator to display the complete audio signal.

▶ Study the help page for "Array Subset." Add controls to specify the start time (in seconds) and the length N (in *samples* – note the difference) of a subrange of audio. The "Horizontal Pointer Slide" control style works well. Enable the digital display for fine control of the desired time and sample values; right-click on the control, choose "Visible Items," and then check "Digital Display."

▶ Use "Build Waveform" to convert the 1-D array subset to waveform datatype, and connect the waveform data type to a new waveform indicator that displays the audio subrange; ensure that the horizontal time axis is properly calibrated in seconds. Re-connect the "Play Waveform" Express subVI to listen to the audio subrange instead of the complete signal. Confirm that you can select any desired audio subrange.

▶ Study the tutorial video `http://youtu.be/9SL5_jY87IA` to learn how to use the "FFT" built-in subVI located in the `Signal Processing | Transforms` subpalette. Place and connect this VI to your audio subrange; use the *same* subrange length control value for the FFT length.

▶ Add an "Array Subset" node to extract the lower half of the FFT output array (the upper half contains a mirror image of the lower half and does not need to be plotted).

▶ Divide the FFT output by the subrange length N to normalize the values.

▶ Place a "Complex to Polar" node from the `Programming | Numeric | Complex` subpalette to compute the FFT magnitude, and then display this array on a waveform plot. Set the plot type to "stem" style by clicking the plot legend on the upper right corner of the indicator. Turn off autoscaling on the Y axis and set the maximum value to 0.12.

$\square_2$ Set the subrange length to 128, then adjust the start time to isolate a single DTMF tone. Run the VI to see the tone's waveform and spectrum, and also to listen to the tone. Create a screenshot of the spectrum and state the associated DTMF tone digit. Discuss the appearance of the DFT spectrum.

$\square_3$ Repeat the previous step for another DTMF tone, noting the differences that appear in the DFT spectrum.

$\square_4$ Try increasing the FFT length to values 256, 512, and 1024; note that the FFT algorithm works most efficiently with array lengths that are a power of two. Discuss all effects that you observe resulting from longer FFT lengths.

☐$_5$ Derive an equation to calculate the FFT spectral bin number for a given frequency f_0. Recall that the bin number is an integer. Calculate the frequency bin numbers for the two DTMF frequencies for a selected DTMF tone digit with $N = 1024$. Zoom in on the spectral lines in the spectrum plot (right-click on the waveform graph indicator, enable the "Graph Palette" option of "Visible Items," and then use the zoom controls). Create a screen shot of the zoomed plot and mark up the screenshot to demonstrate that your equation properly calculates the FFT bin number for a given DTMF frequency.

► Note the abundance of spectral lines that cluster around the base of the tall spectral line that corresponds to each DTMF frequency. This *spectral leakage* effect can be reduced by *windowing* the waveform array before it is applied to the FFT. Study the tutorial video `http://youtu.be/UGUceuhSuUE` to learn more about spectral leakage and windowing.

☐$_6$ Select the Hamming window subVI from the `Signal Processing | Windows` subpalette and insert this before the FFT. Discuss the reduction in spectral leakage that you observe.

5.7 Tone Detector and Display

By now you have developed a good feel for the spectral representation of DTMF tones, and you also know how to translate any particular DTMF tone frequency to an FFT bin number. In this section you will complete the tone detector and display panel.

☐$_7$ Determine a suitable threshold to distinguish between the presence and absence of all 8 DTMF tone frequencies. Use a "Greater?" node from the `Programming | Comparison` subpalette to threshold the FFT magnitude array; use a front-panel control for the threshold value. Report the threshold value you selected.

► Use the "Index Array" node from the `Programming | Array` subpalette to extract the thresholded frequency bins of the four DTMF keypad row frequencies; pull on the bottom handle of "Index Array" to create four index inputs and outputs. Create four constants for the DTMF keypad row frequencies (in Hz) and then add additional mathematical nodes to convert the frequencies to bin numbers for the "Index Array" inputs. In this way your DTMF decoder will work properly for any subrange length N that you choose. The end result of this step is four Boolean wires that indicate which DTMF keypad row is active, if any.

► Repeat the previous step to create four Boolean wires for the DTMF keypad columns. Arrange your block diagram layout to make a 4-by-4 grid of Boolean wires.

► The Boolean "AND" gate uniquely detects the two frequencies for a given DTMF digit. Use sixteen AND gates from the `Programming | Boolean` subpalette placed at the intersections of the row and column Boolean wires. Create a Boolean indicator at the output of each AND gate. Label the indicators according to the sixteen DTMF digits and arrange the front-panel indicators in the shape of a standard DTMF keypad.

► Test your finished tone decoder display for each of the sixteen DTMF tones in your audio test sequence.

5.8 Complete DTMF Decoder

At this point in the project you have demonstrated that your design can successfully detect each of the sixteen DTMF tones in an audio sequence. However, you must manually set the start position of each tone. In this section you will add an *edge detector* to automatically find the leading edges of all of the DTMF tone bursts and then loop over all of the tone bursts detected in the audio signal.

Figure 5.2 on the next page shows the block diagram of the DTMF decoder system with a tone-burst edge detector to automatically identify the starting index of each tone burst in the sequence. The edge detector creates an array of start times that control a for-loop structure to repeatedly run the block diagram that you have developed so far.

► Enter the edge detector block diagram code shown in Figure 5.3 on the following page. The edge detector consists of an envelope detector, comparator, differentiator, and threshold detector. Place the needed components by typing "Ctrl+Space" for the "Quick Drop" menu and typing the following names:

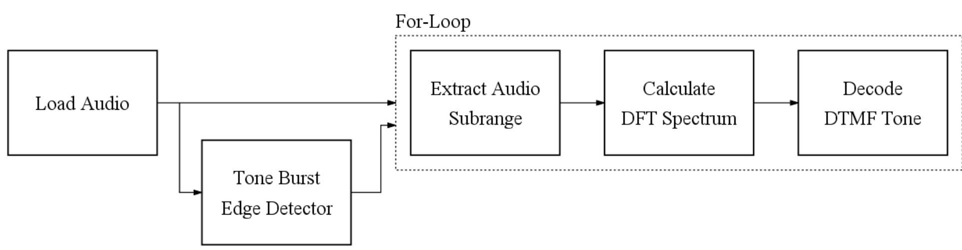

Figure 5.2: High-level block diagram DTMF decoder system with edge detector to automatically identify the starting times of the DTMF tone bursts.

- Absolute Value
- Butterworth Filter
- Greater?
- Boolean To (0,1)
- To Double Precision Float
- Derivative x(t)
- Threshold Detector

Right-click on each component and select "Help" to learn more details.

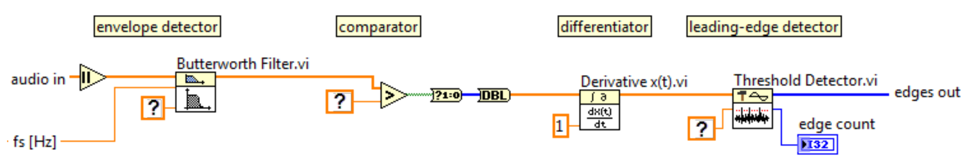

Figure 5.3: LabVIEW block diagram of the edge detector subsystem.

□8 Display the envelope detector output as a waveform plot. Experiment with the Butterworth filter's "low cutoff frequency" value. You want the filter output to accurately track the outline (envelope) of the DTMF tone bursts. Create a screen shot of the envelope detector output and report the final value you selected for the cutoff frequency.

$\Box_9$ Display the comparator output as a waveform plot. Select a threshold value that produces a clean squarewave representation of the DTMF tone burst envelope. Create a screen shot of the comparator output and report the final value you selected for the threshold value.

$\Box_{10}$ Display the differentiator output as a waveform plot; create a screen shot of the plot.

$\Box_{11}$ Add the "Threshold Detector" to determine the index values of the *leading edges* of the tone bursts. Select a threshold value to detect leading edges only; report your final value. Create an indicator for the "count" output to display the number of detected edges.

► Determine which parts of the existing block diagram to enclose in a for-loop structure. Wire the output of the "Threshold Detector" subVI into the for-loop structure using the default "Auto-Indexed Tunnel;" in this way the loop will execute one time for each DTMF tone burst. Include a "Wait" in the for-loop to set the pace of the tone playback and display.

► Test your complete DTMF decoder with your own audio test sequence and confirm that it works properly.

$\Box_{12}$ Run your decoder with the instructor's mystery DTMF sequence. Report the DTMF digit sequence you discovered.

5.9 Discussion

$\Box_{13}$ Discuss the overall performance of your DTMF decoder system.