Implementation of a DTMF dialing voicemail server and client

Luis G. Leon-Vega
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
luis.leon@ieee.org

Allan Navarro-Brenes
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
allannabre01@estudiantec.cr

***Abstract*—This work implemented a Dual-tone multi-frequency signaling dialling-based voicemail application using basic Digital Signal Processing concepts to generate the tones and the Goertzel Algorithm to detect them. Our results show that our implementation was successful by creating a simple server/client-based application using the microphone and speaker of a computer.**

***Index Terms*—DTMF dialing, Goertzel algorithm, voicemail application, DSP**

I. Introduction

The telephone is a system that has become crucial for modern communication [1]. It has enabled voice communication for peer-to-peer communication, customer service, and computer interconnection. In the customer service field, in particular, the automation of customer service has been possible thanks to the integration of digital signal processing (DSP) over the telephone service, allowing people to use the dial pad to interact with an automated system to ask for a service or call the proper department in a telephone central [2].

The Dual-tone multi-frequency signaling (DTMF) is one of the most popular techniques utilised to encode the dial buttons into frequency. It is composed of low frequencies in the rows and high frequencies in the columns. Each button triggers a couple of frequency that encodes the row and the column [3]. In the server side, the frequencies are received, detected, and decoded to reconstruct the dial buttons sent by the client. For detection, the Goertzel algorithm is widely used by this kind of system because of its ease of implementation. It can be seen as a simplified version of the FFT for a range of frequencies [4].

This document is a report of a sample implementation of a voicemail system. The implementation is focused on the frequency generator, encoding, frequency detector and decoding. Additional features such as text-to-speech (TTS) are also implemented to add functionality.

II. Background

After moving manual call re-directions to be automated, the services were expanded to support a primitive way of "typing on a call" for automated customer service. This system is based on DTMF, assigning frequencies to the rows and columns of the dial pad, as shown in Table I. Usually, the encoding system triggers a sine-wave signal generator at the required frequency. Practically, the trigger system consists of a modulator (usually a soft window to avoid artefacts in the frequency domain (the ideal interpolator phenomenon) and a sine-wave generator to turn the tone on and off. The evaluation of the quality of a generator can be quantified through the total-harmonic distortion (THD), given by

$$\text{THD} = \frac{\text{All Spurious Frequencies' Power}}{\text{Total Power}} \tag{1}$$

TABLE I
Frequency assignation of an extended numeric phone dial pad. Frequencies are triggered in pairs to identify a column and a row. Based on [3]

		High-group Frequencies			
		1209 Hz	1336 Hz	1477 Hz	1633 Hz
Low-group Frequencies	697 Hz	1	ABC 2	DEF 3	A
	770 Hz	GHI 4	JKL 5	MNO 6	B
	852 Hz	PQRS 7	TUV 8	WXYZ 9	C
	941 Hz	*	0	#	D

The decoding system, on the other hand, consists of a module to detect the frequency. Two alternatives to perform tone detection is the Discrete Fourier Transform (usually by using the Fast Fourier Transform) [5] or Goertzel [4], which is a domain-specific version of the FFT for a bounded frequency range.

The Goertzel algorithm is a two-stage cascade filter connected with a parameter ω_0 (between 0 and π to avoid aliasing) which is the frequency to analyse. The first stage generates an intermediate sequence $s[n]$ from the input $x[n]$ (see (4), and the second stage generates the output $y[n]$ (see (5)).

$$s[n] = x[n] + 2\cos(\omega_0)s[n-1] - s[n-2] \tag{2}$$

$$y[n] = s[n] - e^{-j\omega_0}s[n-1] \tag{3}$$

Nevertheless, in this work's application, we require to analyse a single component of the spectrum. Thus, it can be simplified as

$$s[N] = 2\cos(\omega_0)s[N-1] - s[N-2] \tag{4}$$

$$y[N] = s[N] - e^{-j2\pi\frac{k}{N}}s[N-1] \tag{5}$$

where k is the frequency bin and N are the frequency bins, s.t. $k = \{0, 1, 2, \ldots, N-1\}$. Then, each bin can be represented by a frequency within the range of interest (if not all are interesting). Thus, the decoding can be performed directly.

III. Implementation

A. Software Architecture

Fig. 1 and Fig. 2 show the high-level software architecture for the server and client sides, respectively. Both programs are implemented in Python and run on independent processes.

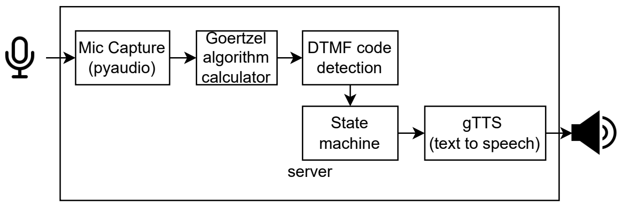

Fig. 1. High level software architecture, server side

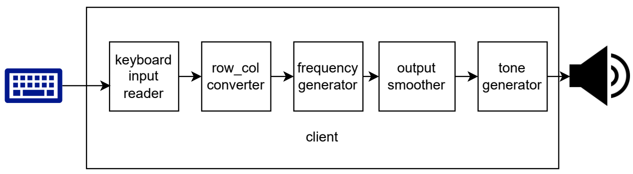

Fig. 2. High level software architecture, client side

IV. Results

Using a reference DTMF detector [6], a recording of the tone generator was made and uploaded to the site using the sequences "0123456789ABCD#*" and "2468013579BD*AC#" without any delay in between tones. The online application detects both sequences without errors as shown in Fig **??** and 4.

Detect DTMF Tones

no graphic available at this time (child process exited abnormally)

Sample Format RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 32000 Hz

Sample Size 239,694 bytes
approximately 119,784 usable samples
3.7 seconds

Tones Found

Tone	Start Offset [ms]	End Offset [ms]	Length [ms]
0	543 ± 15	694 ± 15	150 ± 30
1	694 ± 15	815 ± 15	120 ± 30
2	815 ± 15	935 ± 15	120 ± 30
3	935 ± 15	1,056 ± 15	120 ± 30
4	1,056 ± 15	1,177 ± 15	120 ± 30
5	1,177 ± 15	1,298 ± 15	120 ± 30
6	1,328 ± 15	1,449 ± 15	120 ± 30
7	1,449 ± 15	1,569 ± 15	120 ± 30
8	1,569 ± 15	1,690 ± 15	120 ± 30
9	1,690 ± 15	1,811 ± 15	120 ± 30
A	1,811 ± 15	1,932 ± 15	120 ± 30
B	1,962 ± 15	2,052 ± 15	90 ± 30
C	2,082 ± 15	2,173 ± 15	90 ± 30
D	2,203 ± 15	2,324 ± 15	120 ± 30
#	2,324 ± 15	2,445 ± 15	120 ± 30
*	2,445 ± 15	2,565 ± 15	120 ± 30

Fig. 3. Results of reference DTMF decoder for sequence "0123456789ABCD#*"

Detect DTMF Tones

no graphic available at this time (child process exited abnormally)

Sample Format RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 32000 Hz

Sample Size 319,566 bytes
approximately 159,390 usable samples
5.0 seconds

Tones Found

Tone	Start Offset [ms]	End Offset [ms]	Length [ms]
2	2,445 ± 15	2,596 ± 15	150 ± 30
4	2,596 ± 15	2,716 ± 15	120 ± 30
6	2,716 ± 15	2,837 ± 15	120 ± 30
8	2,837 ± 15	2,958 ± 15	120 ± 30
0	2,958 ± 15	3,079 ± 15	120 ± 30
1	3,109 ± 15	3,199 ± 15	90 ± 30
3	3,230 ± 15	3,320 ± 15	90 ± 30
5	3,350 ± 15	3,471 ± 15	120 ± 30
7	3,471 ± 15	3,592 ± 15	120 ± 30
9	3,592 ± 15	3,713 ± 15	120 ± 30
B	3,713 ± 15	3,833 ± 15	120 ± 30
D	3,833 ± 15	3,954 ± 15	120 ± 30
*	3,954 ± 15	4,075 ± 15	120 ± 30
A	4,105 ± 15	4,196 ± 15	90 ± 30
C	4,226 ± 15	4,316 ± 15	90 ± 30
#	4,347 ± 15	4,467 ± 15	120 ± 30

Fig. 4. Results of reference DTMF decoder for sequence "2468013579bd*ac"

A. Tone generator THD

By calculating a signal's Power spectral density (PSD) it is possible to determine its spectral components. For the tone generator, several files were recorded with the

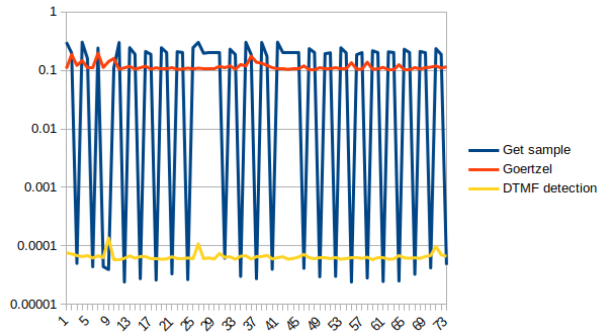

Fig. 6. Time consumption of the detection system. This case includes digit detection and noise detection. Both cases are seamless. Y-axis units are in time and X-axis are samples

basic fundamental frequencies for each tone, i.e: 1209, 1336, 1477, 1633,697, 770, 852, 941. Fig.5 shows the spectral components for each of the files recorded.

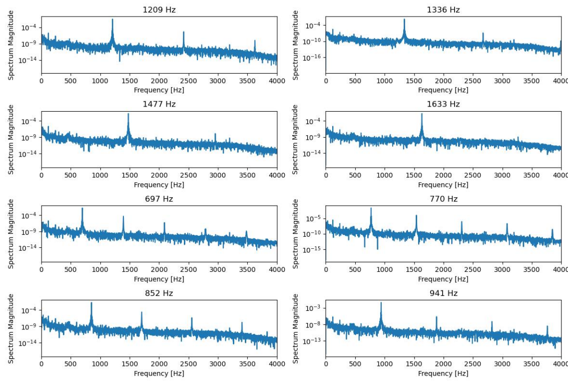

Fig. 5. FFT for each of the DTMF fundamental frequencies.

Using equation (1) and the PSD the THD of the signal can be determined by averaging the THD for each frequency, this results in a value of 22.96%

B. Response time

The response time is a function of the sample rate and the window size. In this case, the sample rate is given by how often a sample is taken and the number of samples to analyse per iteration. Thus, sampling a digit could take, at least, $T_d = N_s/F_s$, where N_s is the number of samples per window (or window size).

Fig. 6 shows the time consumption of the essential three parts of the DSP system. Goertzel takes about 150ms to process a single sample, the tone decoding takes less than 1ms and getting a sample takes around 250 ms. The oscillations are because of sample caching. The reason why it is 250 ms is that the window size is one-fourth of the sample rate. Overall, the worst-case take 400 ms.

C. Tone detection performance

To evaluate the performance of tone detection, we characterised our system by using an adaptation of the Signal-to-Noise ratio (SNR)

$$\text{SNR} = \frac{P_{\text{Signal}}}{P_{\text{Noise}}} \tag{6}$$

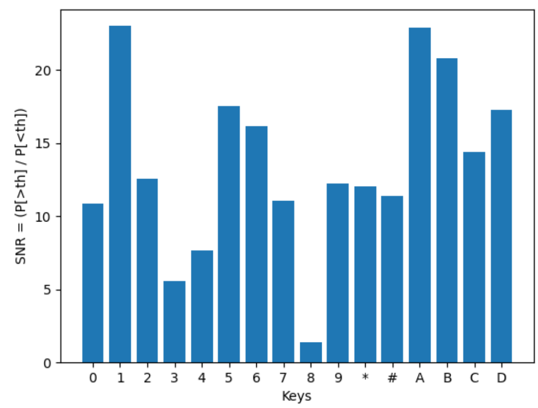

Fig. 7. Signal-to-noise ratio adapted to our use case. The signal power contemplates the following neighbours of the frequency pairs.

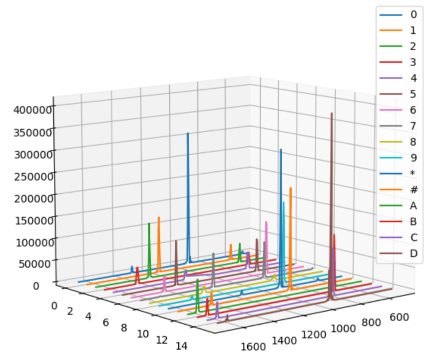

Fig. 8. Spectrum of the signal after measuring each digit with approximately the same setup. The setup was subject to movement and distance changes. X-axis represents the frequency values, Y-axis the series number, and Z-axis the tone power in Watts.

where the signal power is the sum of the powers of the frequency pair, and the noise is the sum of the power of the non-interesting frequencies (others than the frequency pair). For the frequency pair, also the following neighbour frequencies are taken into account, i.e., for $P_{f=697Hz} = \sum_{f=697-5}^{697+5} P_f$. The noise frequencies are the remaining ones in the sample bins.

Fig. 7 shows the SNR applied to one of the testing computers. From the SNR results, there are some digits which are hard to decode, such as the '3', '5', and '8', given that the SNR is lower than 5. The SNR must be greater than 1, in this case, to be reconstructible. Other digits are rather easier to find, such as '1' and 'A', whose SNR exceeds 20.

Fig. 8 complements the SNR analysis to determine the possible cause of their low values. It suggests that the low-SNR digits usually have low power in at least one of their frequencies. Another interesting finding is that the frequencies with the lowest values (rows) tend to have higher power peaks than the high-frequencies (columns). By definition, most reception systems have a low-pass filter behaviour, in particular, sound emission and amplifiers due to parasitic components. This fact can affect the results from Fig. 8.

V. Conclusions

Using a reference application shows that the tone generator can produce valid DTMF codes without delay between tones. The average THD for all the signals generated with the tone generator was 22.96% which means the percentage of spurious frequencies compared to the expected value in terms of power is only 22.96% concerning the total power.

The receptor side also presented interesting results at the level of SNR, having some issues decoding high frequencies. Nevertheless, the lowest SNR was 2, which is still functional for the purpose of tone detection. Overall, the receptor can decode a single sample in about 400ms, which is acceptable considering the typing speed on the dial pad.

On the DSP side, the Goertzel algorithm is a compact algorithm to detect frequency components without running the entire Fourier transform, making the algorithm efficient from the computational and memory perspective.

References

[1] J. Brooks, *Telephone: The first hundred years*. Harper Row, 1976.

[2] ITU, "F.902 : Interactive services design guidelines," 1995. [Online]. Available: https://www.itu.int/rec/T-REC-F.902-199502-I/en

[3] Frank Durda, "Dual Tone Multi-Frequency Reference," 1995. [Online]. Available: http://nemesis.lonestar.org/reference/telecom/signaling/dtmf.html

[4] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *The American Mathematical Monthly*, vol. 65, no. 1, pp. 34–35, 1958. [Online]. Available: http://www.jstor.org/stable/2310304

[5] J. G. Proakis and D. G. Manolakis, *Tratamiento Digital de señales.*, 2003, vol. 53, no. 9.

[6] DialABC. [Online]. Available: http://dialabc.com/sound/detect/index.html