REFERENCES

Telephony:

1. Digital Simulation Test Tape." Bell Communication Research Technical Reference TR-TSY-D00763, Issue 1, July 1987.

2. "Dual-Tone Multifrequency Receiver Generic Requirements for End-to-End Signaling Over Tandem-Switched Voice Links." Bell Communications Research Technical Reference TR-TSY-000181, Issue 1, March 1987.

3. Noll, A. Michael. *Introduction to Telephones and Telephone Systems*. Artech House, Inc., 1986.

4. Keiser, Bernhard E. and Strange, Eugene. *Digital Telephony and Network Integration*. Van Nostrand Reinhold Company, 1985, pp. 289-90, 306-7.

5. Freeman, Roger L. *Telecommunication System Engineering: Analog and Digital Network Design*. John Wiley & Sons, 1980, pp. 377-8.

Digital Signal Processing:

6. Blahut, Richard E. *Fast Algorithms For Digital Signal Processing*. Addison-Wesley Publishing Co., Inc., 1985, pp. 131-2.

7. Manolakis, Dimitris G. and John G. Proakis. *Digital Signal Processing*. 2nd edition. Macmillan Publishing Co., 1992.

8. Burrus, C. S. and T. W. Parks. *DFT/FFT and Convo lution Algorithms: Theory and Implementations*. Texas Instruments, Inc., 1985, pp. 32-36.

9. Marple, S. Lawrence Jr. *Digital Spectral Analysis*. Prentice-Hall, Inc., 1987, pp. 136-44.

10. Oppenheim, Alan V. and Schafer, Ronald W. *Digital Signal Processing*. Prentice-Hall of India, 1989.

Systems:

11. Fannin, D. Ronald, Tranter, William H., Ziemer, Rodger E. *Signals and Systems: Continuous and Discrete, 3rd Ed*. MacMillan Publishing Incorporated, 1993, pp. 486-501, 523-525.

12. Tranter, W. H. and Ziemer, R. E. *Principles of Communications Systems, Modulation and Noise*. Houghton Mifflin Company, 1990.

WWW Sites:

13. TI. http://www.ti.com.

14. Programming in C. http://arachnid.cs.sf.ac.uk/Dave/C/CE.html.

15. Introduction to Signal Processing. http://www-ece.rutgers.edu/gopher/ftp/pub/sjo/intro2sp.html.

The DTMF detector was also evaluated for real time operation. Data was taken from a touch-tone telephone via a microphone and an A/D converter. The narecord program was used to set the channel preference and sample frequency and pipe the sampled data to the detector. The pipe command was also utilized to send files to the input of the detector for real time testing.

The two parameters that affected real time operation were N(total number of points computed) and the size of sliding window(number of points computed each iteration). Real time performance deteriorates as N grows large(>400) or the sliding window size grows small(<64). This highlights the interdependence between parameters. Decrease N for real time and the bandwidth grows large accordingly.

6. SUMMARY

Testing and evaluation resulted in the Hamming window being chosen for this detector. The frame size used is N=330. For this value: the bandwidth is acceptable, real time operation is achieved, and the adjacent frequencies fall in or very near to the null.A sliding window size of 80 was chosen because this value was small enough ensure that the detector would always detect signal durations of 40ms. Lower values of N compromise speech and noise immunity. Higher values of N slow processing and decrease the bandwidth to the extent that telephones with any deviation from ideal DTMF signal frequencies cannot be detected.

Testing revealed inadequacies in the generation of DTMF frequencies by signalling devices. It was found that several telephones that were tested actually produced frequencies that were significantly out of specified ranges. Because of this fact, we began to look for windows with wider bandwidths to compensate for this deviation and increase detector accuracy.

This detector was tested extensively with computer generated and real data. It performed flawlessly on all data in table 3. It registered zero misses on data with signal-to-noise ratios as low as 10dB. The detector experienced no failures due to music or talk-off.

This detection scheme is more robust and cost-effective than conventional analog detection techniques. The algorithm was tested with a variety of data including speech, music, and DTMF tones. We find that this detector is efficient, reliable, and exceeds BellCore standards for DTMF detection.

7. ACKNOWLEDGEMENTS

We would like to thank Dr. Joseph Picone for his guidance and patience throughout this project. We would also like to thank Dr. and Mrs. Picone for their help in data collection and the use of their telephones and recording equipment.

We wish to thank Dr. William J. Ebel for his support, ideas, and the use of his personal library. We also want to acknowledge the staff of ISIP for their technical support, and especially Mary Weber for her help in scanning images.

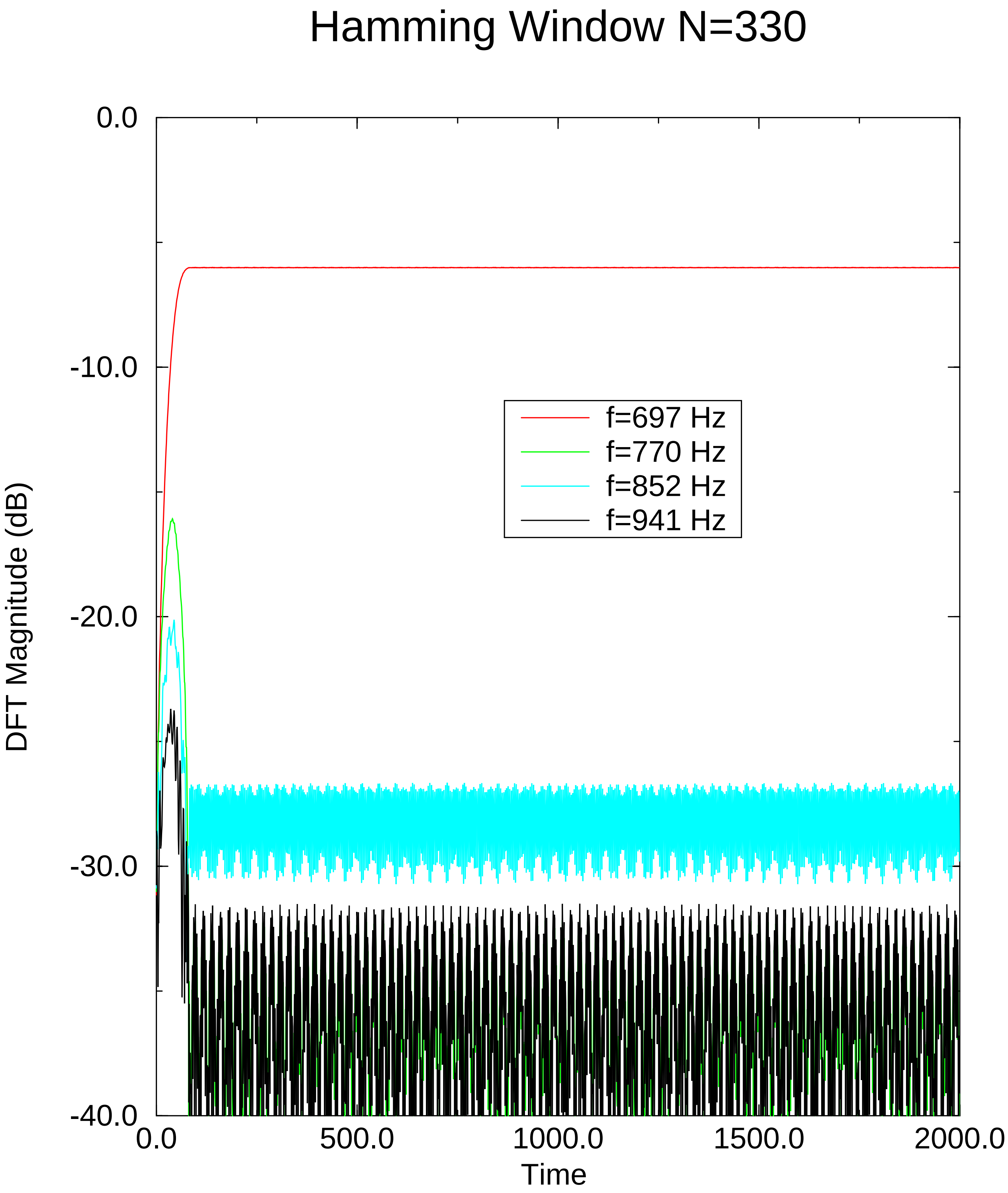

Figure 6: Hamming Window N=330

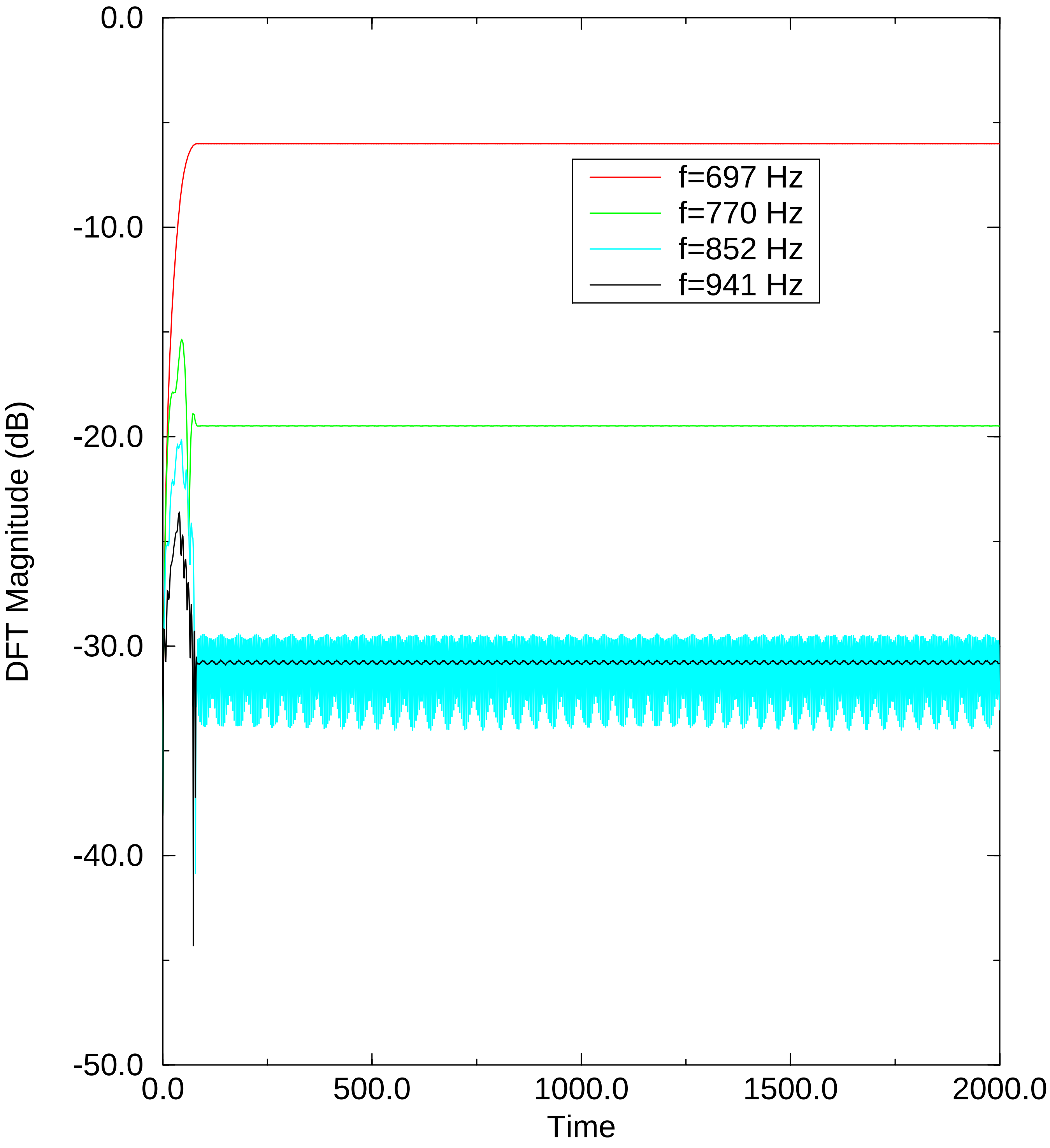

Figure 5: Triangular Window N=330

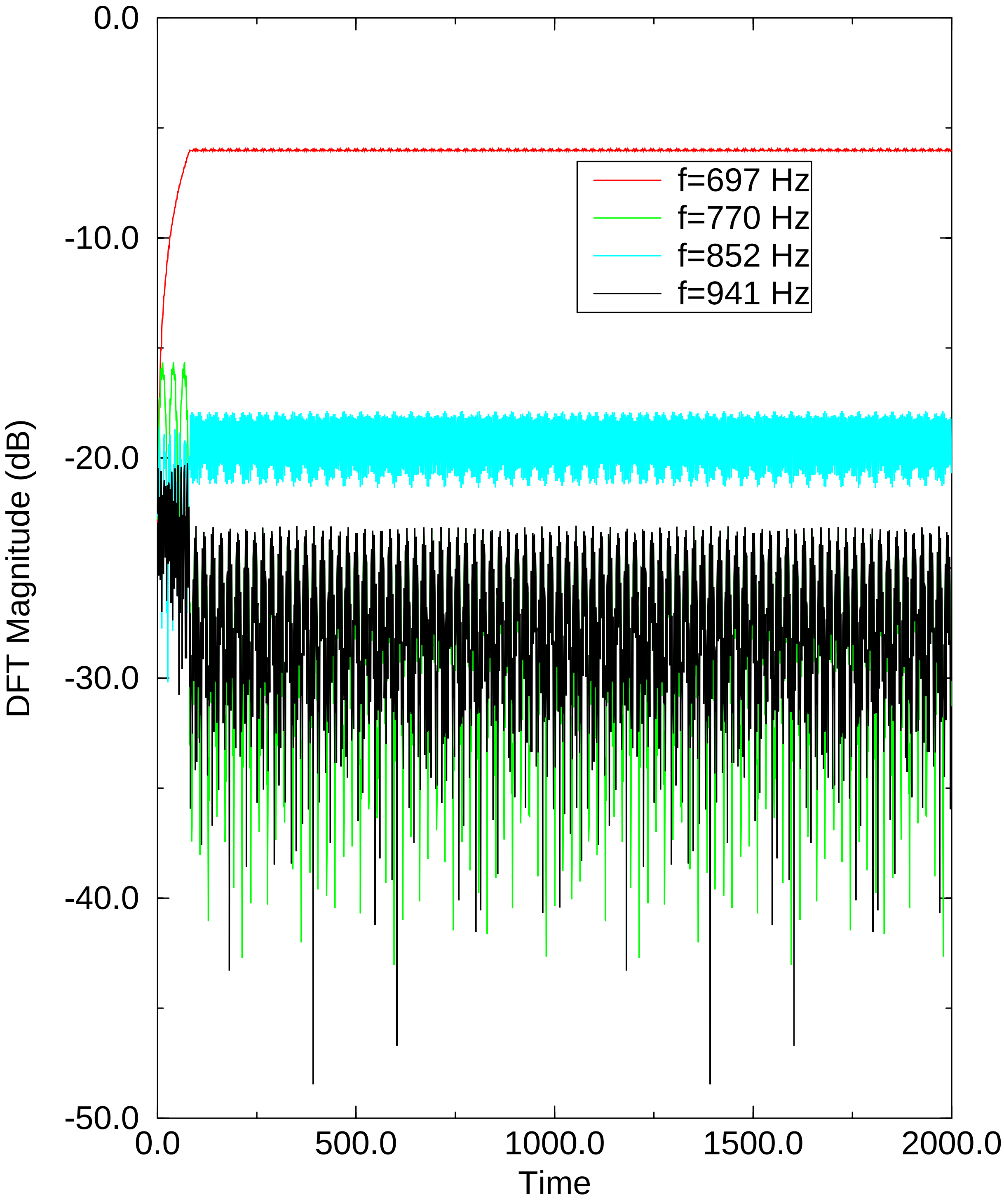

Figure 4: Rectangular Window

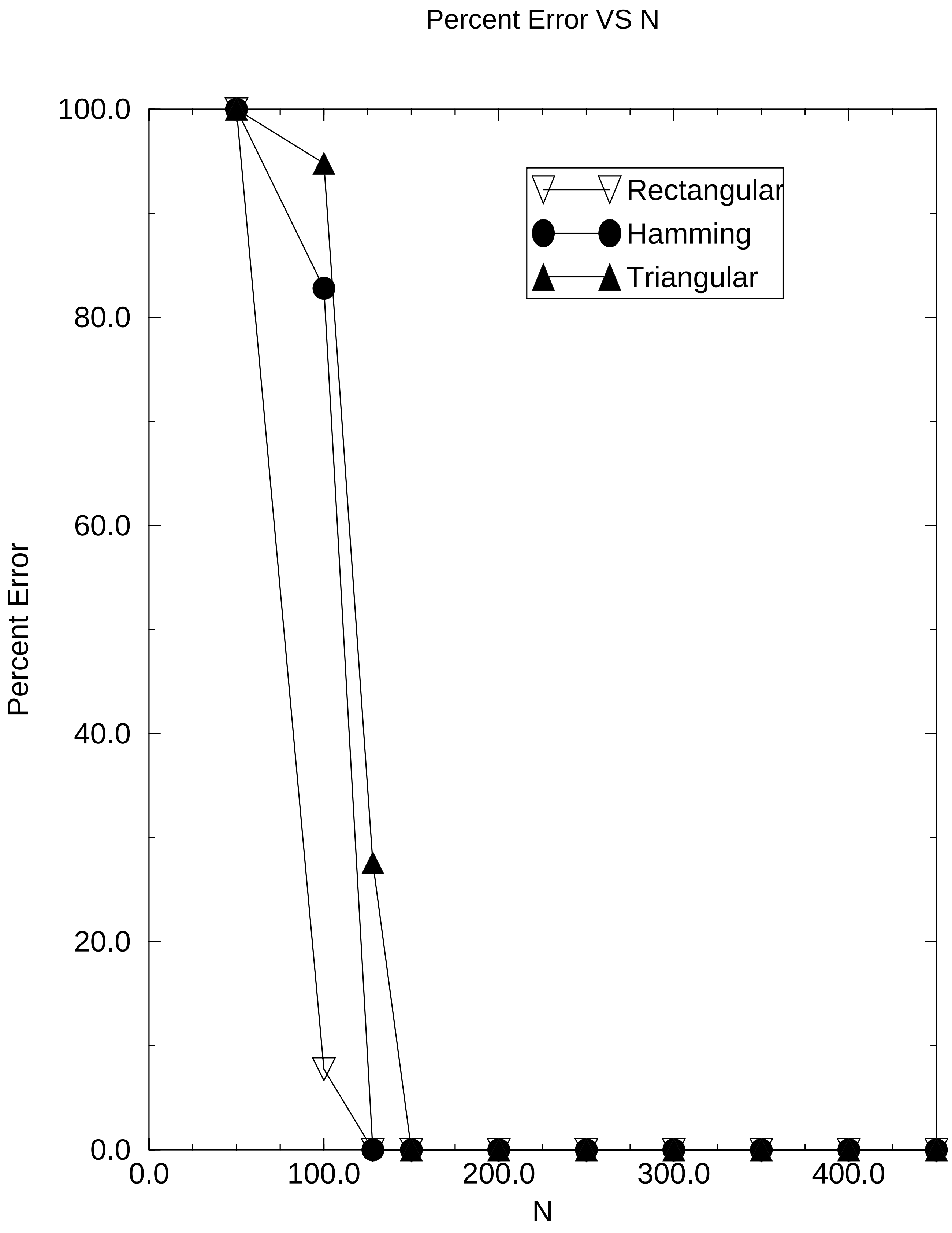

Figure 3: Percent Error VS N

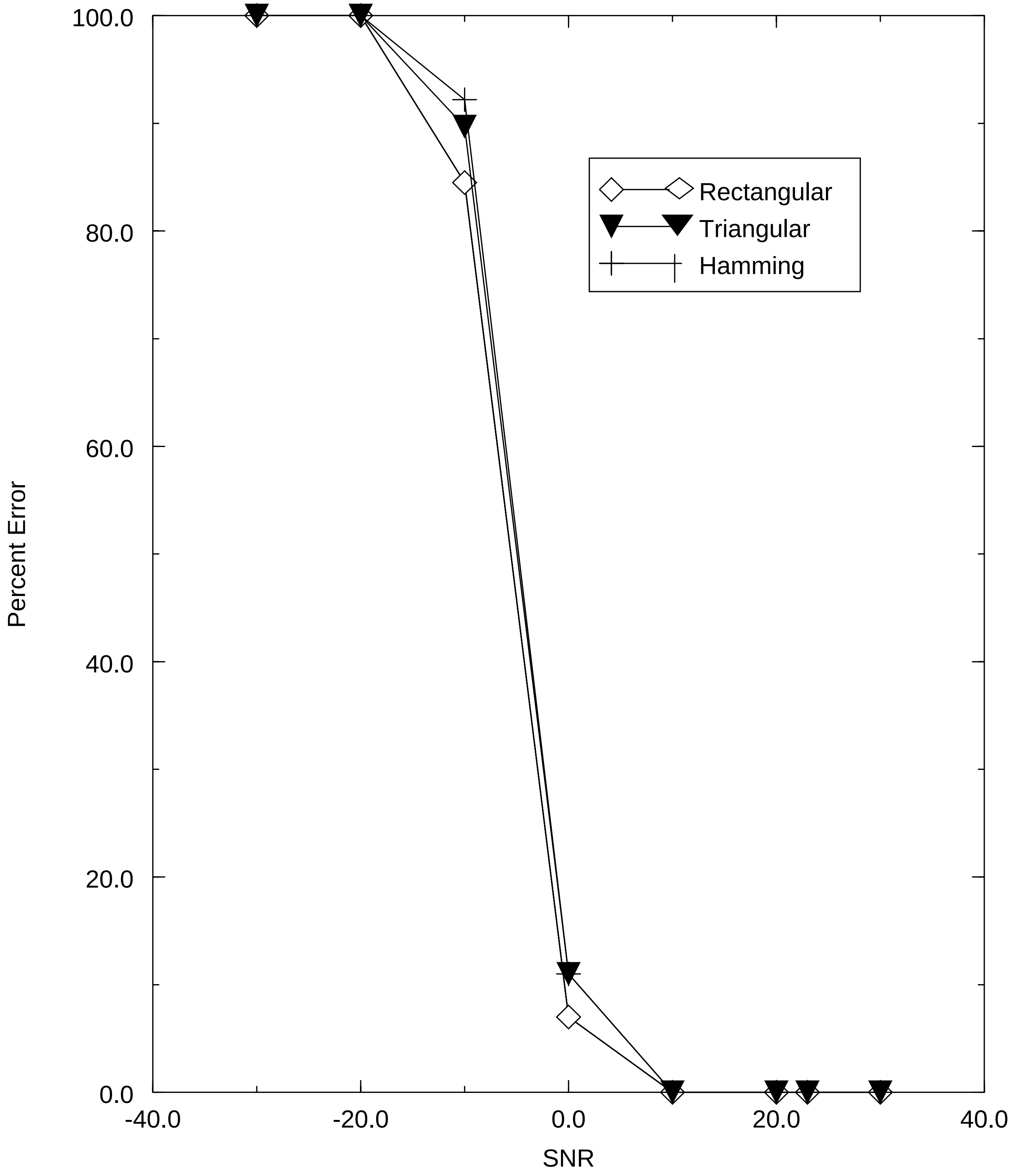

Figure 2: Percent Error VS SNR

sequence x(n) from a circular buffer. The original Goertzel algorithm assumes that the data is stored in a vector with x(0) located in element 0 of the vector. LiNear storage of the data proved to be awkward, at best. The entire vector had to be reconstructed each time the DFT was to be recomputed. Using a circular buffer, only `stale' data need be changed as new data is received, while updating the pointers appropriately.

5. EVALUATION

5.1. Different Window Functions

One of the most important issues that was contended with was which window yields the best results. The key parameter focused on during testing was the window size(N)[11]. We tested each window using values of N ranging from 50 to 450(figure 4). These were the values that we determined would keep us within bandwidth specification and still allow us to run in real time.

Certain other values of N were chosen to optimize the placement of the zero locations in the frequency domain. Figures 4-6 illustrate the relative sidelobe attenuation for the rectangular, triangular, and Hamming windows. It can be seen that the Hamming window provides superior sidelobe attenuation at the adjacent DTMF signal frequencies[14].

5.2. Data Generation

Several methods were used to produce data which was used to test the DTMF detector. Computer programs were written in C++ to generate a sampled sinusoidal signal sequence. The amplitudes and frequencies of the sinusoidal signals were varied and approximately 100 files were generated containing a variety of tones, tone spacings, and tone durations.

A simulation was then done using each file as input data to the DTMF detector and the results were recorded. Next, a noise file of approximately ten second duration was recorded using an inexpensive clock-radio tuned to no station as a noise source. This file was then summed with each of the computer-generated signal files to form noisy data files of specified signal-to-noise ratio. Simulations were done with SNRs ranging from -30 to 30, in steps of 5 dB, and the results were recorded. Real DTMF data was then collected by originating a normal voice call over the public switched network, and recording the keypresses using the Sony DAT system. Several recordings were made, using several different telephone sets. DTMF tones were generated by cycling through the keypad several times, as well as by entering valid seven- and eleven-digit telephone numbers. The algorithm was simulated using these recorded files, and the results were noted. Each of these files was then combined in various ratios with the noise file previously described to again create noisy data. Simulations were done with SNRs ranging from -30 to 30 and the results were noted.

Software Generated Data	100 files 7 windows 100 files with noise 7 windows
Recorded Data	Phone 1 3 files(92 key presses) Phone 2 2 files(124 key presses) Phone 3 1 file(99 key presses)
Speech Data	151 files(0 key presses)
Music and Speech Data	25 files(0 key presses)

Table 3: Testing Database

Figure 3 shows the signal-to-noise ratio versus percent error for various windows. The detector performs perfectly for SNRs down to 10 dB as can be seen in figure 2.

	Windows		
SNR	Rectangle	Triangle	Hamming
30	0	0	0
23	0	0	0
20	0	0	0
10	0	0	0
0	7	11	11
-10	84.5	89.7	92.2
-20	100	100	100
-30	100	100	100

Table 4: Percent Error for different SNRs and Windows

$$w(n) = 0.54 + 0.46\cos\left(\omega\frac{n}{N}\right)$$

Equation 4: Hamming(Raised Cosine)Window

Window	Highest Sidelobe Level	Equivalent BW	1/2 Power BW
Rectangle	-13.3 dB	1.00	0.89
Triangle	-26.5 dB	1.33	1.28
Hamming	-43 dB	1.36	1.30

Table 2: Characteristics of Windows

were tested. This table shows why the sidelobe attenuation characteristics of the Hamming window were preferable to the others given the slight difference in bandwidth.

4. GOERTZEL'S ALGORITHM

4.1. The Algorithm

The algorithm we used to compute the Fourier transform X(w) is based substantially on Goertzel's algorithm. Goertzel's Algorithm. This algorithm models the computation of the DFT as a linear filtering operation [9]. This operation takes the form of a parallel bank of resonators, where each resonator selects one of the frequencies of the DFT [8]. The output of each of theses filters at n=N yields the value of the DFT at the frequency wk = 2*PI*k/N.

The advantages of this approach over other algorithms are two-fold: 1) it is computationally more efficient, and 2) the value of the DFT can be computed at any frequency desired. The extent to which the Goertzel algorithm is more efficient than other algorithms (such as Fast Fourier Transforms) depends on the number of frequencies at which the DFT is to be computed[9]. Each iteration of the Goertzel algorithm requires one real multiplication and two real additions. If the value of the DFT is required at M points, then the total cost for computing the DFT is M*N multiplications and 2*M*N additions. For values of M < Log N, Goertzel's algorithm is less expensive computationally than an FFT algorithm which is O(N*Log N).

A second advantage to Goertzel's algorithm is that, unlike FFT algorithms require N and N to be equal and a power of 2[9], Goertzel's algorithm places no artificial constraints on these values. Hence, Goertzel's algorithm may be used to compute the value of the DFT at any frequency, and with any value of N. This is critical in DTMF detection because the resolution between frequencies decreases as the value of N decreases. If the FFT were used, the value of the DFT could be computed only at N equally spaced frequencies, which may or may not correspond to the frequencies of interest. The fact that any value of N may be used with Goertzel's algorithm is also highly beneficial, because, as will be shown later in this paper, the optimal range of N is about 150 - 400.

Three primary factors led to the selection of this technique. First, the DFT has been shown by Proakis [9] to be a linear operation, i.e. any noise that is contained in the time-domain signal representation is not amplified in the frequency-domain by the DFT. Secondly, the DFT is well-studied, and several efficient algorithms exist to compute the DFT.

4.2. The Modifications

As previously stated, the algorithm presented here is based heavily on Goertzel's algorithm. Several substantial modifications were made, however, to improve the overall computational efficiency of the DTMF detector. First, we recognize that we are interested in computing the DFT at a very small set of frequencies. We create an array of size eight containing only the values of k corresponding to the eight frequencies with which we are interested. The algorithm was then modified to compute the DFT at only these frequencies.

Secondly, we recognized that the values of k need not be restricted to integers. This results directly from the fact that X(w) is continuous in w [9]. By allowing k to take on floating point values, the DFT can be computed at precisely the frequency of interest, within roundoff error, regardless of the value of N. This single modification eliminated altogether the problems associated with frequency resolution.

A third improvement came from the realization that Goertzel's algorithm was designed to handle a complex input data sequence x(n). However, the real and complex portions of the computations are separated in the algorithm. Because samples of a real signal are real-valued, many of the computations were unnecessary and were eliminated. This effectively halved the number of computations required for each iteration.

A fourth modification that was made was to precompute the phase factors Wk. This saves sixteen trigonometric evaluations per DFT computation, at a cost of the storage of sixteen floating point numbers.

Finally, the algorithm was modified to extract the input

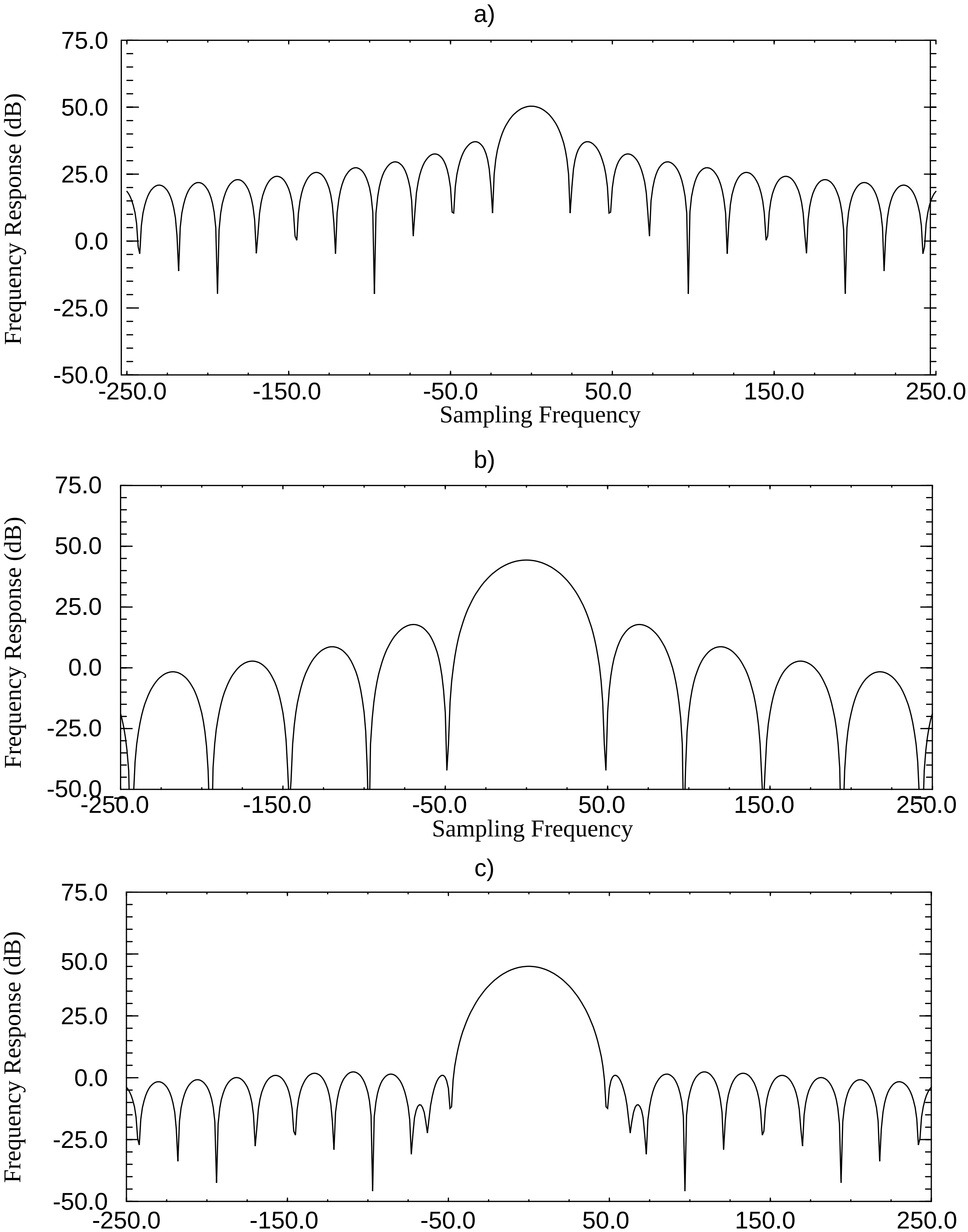

Figure 1: Frequency response for: a) Rectangular Window, b) Triangular Window, and c) Hamming Window

Which becomes[10]

$$X(\omega_0) = \frac{1}{2}AN$$

In this manner the amplitude of an individual frequency component can be computed.

$$A = \frac{2}{N}X(\omega_0)$$

The signal power is then $A^{2.}$

2.1. Finite Sampling Effects

A consequence of approximating the infinite duration signal x(t) with the finite duration sequence x(n) of length Nis distortion. Consider again the signal x(t) = A sin(w0t). In the frequency domain, this signal can be represented as X(F) = 0.5*A*delta(w+-w0). The Fourier Transform of this signal X(w) is

$$X(\omega) = \frac{A}{2}\sin c\left(\left(N + \frac{1}{2}\right)\omega - \omega_0\right)$$

Note that X(F)!= X(w). X(w) is a sinc function centered around w_0 with a main lobe width of 1/PI = 2*fs/N. Thus, X(w) approaches X(F) with increasing N.

2.2. Choice Of N

There are several factors which together constrain the choice of N. First, it is clearly evident in equation 3 that the bandwidth of X(w) is inversely proportional to N. Given the BellCore requirement that the bandwidth be less than 3% of the center frequency, N must be chosen not less than 380. Other constraints on the choice of N include the DTMF tone duration. Recall that N is the number of signal samples from which the DFT is computed. A large portion of the sampled signal may contain portions of multiple key presses, which could significantly alter the resulting spectrum. Because valid DTMF tones may be as brief as forty ms, with tone to tone separation of forty ms, a value of N samples greater than 321 samples may span multiple key presses. The extent to which errors are induced depend on a number of factors, including the relative power levels of the input signal, the amount of overlap onto the adjacent DTMF tone, and the window function used[14]

A third factor influencing the choice of N is the requirement that the DTMF detector operate in real time. Recall that the computation of the DFT is a summation of N terms. The time complexity of the best sequential algorithm for the computation of the DFT is O(N log N) [9]. Obviously, as N increases, the amount of computational effort required to compute the DFT increases as well. Therefore, the value of N must be chosen sufficiently small to ensure that the detector will run in real time.

3. WINDOWS

Windowing is an essential consideration when performing spectral estimation. It is used to control the sidelobes of the spectral estimator[11]. The magnitude of sidelobe attenuation is a large factor in DTMF detection. A DTMF detector cannot tolerate sidelobe leakage, but neither can it tolerate large mainlobe bandwidths. This is where a good balance of mainlobe bandwidth versus sidelobe attenuation must be achieved. Since the two are inversely related, this problem of windowing must be given serious consideration.

The windows that were considered with this DTMF detector were: rectangular(uniform), triangle(Bartlett), Hamming(raised cosine) and Hann(squared cosine). Complete testing was only performed using the rectangular, triangular, and Hamming windows as all of the others were close derivatives of the Hamming and therefore provided insignificant performance differences.

Equations 2, 3, and 4 are the weighting functions for the rectangle, triangle, and Hamming windows[11]. The frequency response of these functions can be seen in Figure 1.

Table 2 shows some characteristics of the windows that

$$w(n) = 1, n = 0 \ldots N-1$$

Equation 2: Rectangular (Uniform)Time Window

$$w(n) = \begin{cases} 2\frac{n}{N-1}, n < \frac{N}{2} \\ 2 - 2\frac{n}{N-1}, n > \frac{N}{2} \end{cases}$$

Equation 3: Triangular (Bartlett)Time Window

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

Table 1: DTMF Frequencies and Character Assignments

On the other hand, the design must be efficient enough to run in real time. So we have a trade-off between the number of DFT points that must be taken to give proper bandwidth(and accurate spectrum) compared with the amount of time it takes to evaluate that number of points. BellCore[2] specifies that the bandwidth of the detection filter be within +-1.5% of the designed center frequency and must accept frequencies that fall within that range. There is also what is referred to as a rejection bandwidth beyond which the filter must attenuate any frequencies. This bandwidth is +-3.5% of the designed center frequency.

Determining whether DTMF signals are present or not is done using an adaptive level detection scheme. BellCore specifies that for adaptive thresholding, the signal frequency must be present at a power level of 9dB greater than that of the other signal frequencies in the same group(high or low, see Table 1). The other power level consideration is twist. Twist is the difference between the power of the signal frequency in the low group and that of the signal frequency in the high group. Twist is defined as positive when the high frequency power level exceeds that of the low frequency and negative if the opposite is true. BellCore‘s specification for twist is +4 dB to -8 dB. Once a valid level is found, it is desirable to know how long the signal is present at that level.

Timing is another important issue when doing DTMF detection. The duration of the signal is critical to the accuracy of the detector. BellCore. specifies several timing constraints: signal duration, interdigit time, and cycle time. According to BellCore, a DTMF detector should recognize tones of 40ms or greater. Alternatively, the detector can recognize tones with durations as low as 23ms. Any signal with a duration < 23ms must be rejected. This specification is instrumental for ensuring robustness to noise and speech. Interdigit time is the time between the end of one signal and the beginning of the next. Signals with interdigit times of at least 40ms must be recognized but the interdigit time may alternatively be allowed to approach zero. The cycle time is the time from the start of one signal to the start of the next. The minimum specification for cycle time is 93ms, but it also may be allowed to approach zero. We placed no bounds on interdigit or cycle time in our DTMF detector.

2. THEORY

The problem addressed in this paper is how, given a digital single-channel data stream, to detect the presence of valid DTMF tones. The algorithm must be capable of accurately determining 1) which of the eight DTMF frequencies are present, 2) the relative signal power at each of the frequencies, and 3) the duration that these signals are present. These signal characteristics must then be compared to industry standard criteria to determine whether a valid DTMF tone is present.

Several methods exist for determining the component of a signal at a given frequency[10]. The technique chosen here is the Discrete Fourier Transform (DFT). The DFT allows us to represent an aperiodic discrete signal as the algebraic sum of its frequency components. The DFT has the form

$$x(\omega) = \sum_{n=0}^{N-1} x(\omega)e^{-j\omega n}, n = 0, 1, \ldots, N-1$$

Equation 1: Discrete Fourier Transform

where x(n) is a finite duration sequence of length N, and w is the normalized radian frequency[9].

Consider a signal

$$x(t) = A\sin(\omega_0 t)$$

The sampled signal is then

$$x(n) = A\sin\left(\omega_0\frac{n}{f_s}\right) = A\sin\left(2\pi n\frac{f}{f_s}\right)$$

The DFT of this signal is given by

$$x(\omega_0) = \sum_{n=0}^{N-1} x(\omega)e^{-j\omega_0 n}, n = 0, 1, \ldots, N-1$$

A DISCRETE FOURIER TRANSFORM BASED DIGITAL DTMF DETECTION ALGORITHM

Jimmy H. Beard, Steven P. Given, and Brian J. Young

The Digital Signal Processing Group
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
{beard, given, byoung}@ee.msstate.edu

ABSTRACT

We present a new type of digital Dual-Tone Multifrequency(DTMF) detection scheme based on the Goertzel DFT algorithm. This detection scheme is more robust and cost-effective than conventional analog detection techniques. This algorithm is designed to provide optimal performance and exceed BellCore[2] specifications for DTMF detection. The problems associated with closely spaced signal frequencies and short tone duration are overcome by proper window and frame selection. Adaptive thresholding is provided to minimize false outputs due to noise and speech.

The algorithm was tested with a variety of data including speech, music, and DTMF tones. We find that this detector is efficient, reliable, and exceeds BellCore standards.

1. INTRODUCTION

The first touch tone telephone installation was in 1963[3]. DTMF signaling uses voiceband tones to send address signals and other digital information from push-button telephones and other devices such as modems and fax machines[2]. Analog DTMF detection is done using bandpass filter banks with center frequencies at the DTMF signal frequencies.

Analog receivers have wide tolerances to compensate for distortion caused by aging transmitters, variations in keying characteristics, and transmission line distortion. These distortions compound the problem of digital DTMF detection[4].

The introduction of digital pulse code modulation(PCM) switches in 1976 signaled the beginning of the end of the analog telephone network. In the past 20 years telephone networks have been rapidly moving from totally analog to totally digital. In digital switching systems is desirable to treat all signals uniformly, bringing all signals through A/D converters and switching them through the PCM system[4]. Therefore the need for digital DTMF detection is justifiable to avoid the costs of hardware and D/A conversion needed to use analog detectors. With the constant advances in VLSI driving DSP costs downward, it is economically sound to replace analog detectors with their digital counterparts which are more reliable, maintenance cost effective, and spatially minimal.

Several techniques for digital DTMF detection have been used, but most designers have settled on either digital filtering or discrete fourier transform(DFT)[4]. In digital filtering, DTMF signals are passed through digital bandpass filters centered at the signaling frequencies(shades of analog). The power at each frequency is then measured repeatedly to detect the DTMF tones. A DSP then interprets and translates them for the proper switching. The DFT also measures signal power at the signaling frequencies but has the additional need to check for signals of some minimum duration. This will help to ensure robustness toward speech and noise. The DFT is the technique that is outlined in this paper.

While the actual DFT is based on the aforementioned Goertzel algorithm, the windowing, level detection, and timing logic involved in developing a working design ensure that digital DTMF detection is a non-trivial problem.

As can be seen in Table 1, the DTMF signaling frequencies are very closely spaced. It is obvious that the bandwidth of the filter used for detection must be narrow enough to avoid any bleeding of adjacent frequencies. An even more limited bandwidth is introduced when one considers that some of today’s DTMF signal generators(phones, modems, etc.) are poorly made to such an extent that the actual frequencies generated are not as shown in Table 1.