Performance Evaluation and Real-Time Implementation of Subspace, Adaptive, and DFT Algorithms for Multi-tone Detection

Güner Arslan† *Brian L. Evans*‡ *F. Ayhan Sakarya*† *José Luis Pino*‡ *

† Dept. of Electronics and Comm. Eng., Yıldız Technical University, Istanbul, Turkey
‡ Dept. of EECS, University of California, Berkeley, CA 94720, USA

Abstract - **We evaluate MUSIC, LMS, and the Goertzel algorithms for dual-tone multiple-frequency estimation. We rate their computational requirements, estimation errors, and their compliance with the CCITT Q.23 and Q.24 standards through MATLAB simulations and real-time implementations on the ADSP-2101 and Motorola 56001 processors.**

I. INTRODUCTION

Estimating the frequencies of multiple signals in noise is an important problem in signal processing. We can estimate these frequencies, using subspace techniques such as Multiple Signal Classification (MUSIC) [1], adaptive techniques such as Least Mean-Square (LMS) estimation [2], or fast implementations of discrete Fourier transform (DFT) such as the Goertzel algorithm [3]. The choice of the technique is based on a trade-off between the observation time and the available computational resources.

We apply MUSIC, LMS-based normalized Direct adaptive Frequency Estimation Technique (DFET) [4] (NDFET) [5], and the Goertzel algorithm to the detection of dual-tone multiple frequencies (DTMF). We evaluate their performance and compliance with standards through MATLAB simulations and real-time embedded digital signal processor implementations. We coded the ADSP-2101 implementations manually, but we used the Ptolemy 0.6 rapid prototyping environment [6] to generate the Motorola 56001 implementations. A key benefit of using a tool like Ptolemy is that the DTMF decoders can be reused in other designs and retargeted to VHDL, C, and other implementations.

II. DUAL-TONE MULTIPLE FREQUENCIES

DTMF signaling has many applications such as telephone dialing, data entry, credit checking, and voice mail system control. A DTMF signal consists of two superimposed sinusoidal waveforms with frequencies chosen from a set of eight standardized frequencies. These frequencies should be generated and detected according to the CCITT Recommendation Q.23 [7] and Q.24 [8]:

1. Signal frequencies
   a. Low group: 697, 770, 852, 941 Hz
   b. High group: 1209, 1336, 1477, 1633 Hz
2. Frequency tolerance
   a. Operation: $\leq 1.5\%$
   b. Non-operation: $\geq 3.5\%$
3. Signal reception timing
   a. Signal duration/operation: 40 ms min.
   b. Signal duration/non-operation: 23 ms max.
   c. Pause duration: 40 ms min.
   d. Signal interruption: 10 ms max.

III. GOERTZEL ALGORITHM

The Goertzel algorithm is more efficient than the Fast Fourier Transform in computing an N-point DFT if less than $2 \log_2 N$ DFT coefficients are required [9]. In DTMF detection, we only need 8 of, for example, 205 DFT coefficients to detect the first harmonics of the 8 possible tones, and then apply decision logic to choose the strongest touch tone. Since DTMF signals do not have second harmonics, we could compute another 8 DFT coefficients to compute the second harmonics to detect the presence of speech [3].

The Goertzel algorithm computes the kth DFT coefficient of the input signal $x[n]$ using a second-order filter

$$s_k[n] = x[n] + 2\cos(\frac{2\pi k}{N})\, s_k[n-1] - s_k[n-2] \quad (1)$$

$$y_k[n] = s_k[n] - W_N^k\, s_k[n-1] \quad (2)$$

where $s_k[-2] = s_k[-1] = 0$. The kth DFT coefficient is produced after the filter has processed N samples: $X[k] = y_k[n]|_{n=N}$. The key in an implementation is to run $s_k[n]$ for N samples and then evaluate $y_k[N]$. The computation for $s_k[n]$ takes one add ($x[n] - s_k[n-2]$) and one multiply-accumulate per sample. In DTMF detection, we are only concerned with the power of the kth coefficient, $y_k[N]y_k^*[N]$:

$$s_k[N]s_k[N] - 2\cos(\frac{2\pi k}{N})s_k[N]s_k[N-1] + s_k[N-1]s_k[N-1]$$

---

*G. Arslan and F. A. Sakarya were supported by the Yıldız Technical University (YTÜ) Research Foundation under project number #94-B-04-03-01. B. L. Evans and J. L. Pino were supported by the Ptolemy project which is supported by ARPA, U.S. Air Force, Semiconductor Research Corporation, and various companies. The Ptolemy World Wide Web page is located at http://ptolemy.eecs.berkeley.edu. F. A. Sakarya is also with the Marmara Research Center of TÜBİTAK, Gebze, Turkey.

The value of N must be shorter than the samples in half of a DTMF signaling interval ($N < 400$), be large enough for good frequency resolution ($N > 516$), and meet the relative error specification 2(a). We used a conventional value of N = 205 [3, 10] because it is roughly half of 400 samples. Decision logic can be added to give a valid DTMF signal if the same two DTMF tones are detected in a row to add robustness against noise.

Table 1: Chosen k values to minimize the error in k.

Tone in Hz	floating point k	k	Absolute error	Relative error
697	17.861	18	0.139	0.0078
770	19.731	20	0.269	0.0136
852	21.833	22	0.167	0.0077
941	24.113	24	0.113	0.0047
1209	30.981	31	0.019	0.0006
1336	34.235	34	0.235	0.0069
1477	37.848	38	0.152	0.0040
1633	41.846	42	0.154	0.0037

IV. NDFE ALGORITHM

The Direct adaptive Frequency Estimation Technique [4] is based on the Least Mean Squares (LMS) algorithm [2]. We introduce an adaptive step size to normalize the result and called the algorithm NDFET [5].

Using trigonometric identities, we can rewrite a sinusoidal signal of frequency ω as

$$\sin(\omega \cdot n) = 2\cos(\omega)\sin(\omega \cdot (n-1)) - \sin(\omega \cdot (n-2))$$

Letting $\sin(\omega \cdot n) = x[n]$ and $a = \cos(\omega)$ yields

$$x[n] = 2\,a\,x[n-1] - x[n-2] + e[n] \tag{3}$$

where $e[n]$ is the estimation error or additive noise. When $x[n]$ consists of multiple sinusoids, $e[n]$ includes the sum of the additional sinusoidal terms. We apply LMS with step size μ to minimize $e[n]$ and estimate a:

$$\hat{a}[n] = \hat{a}[n-1] + 2\,\mu[n]\,x[n-1]\,e[n-1] \tag{4}$$

From equations (3) and (4), the LMS filter output is

$$e[n] = x[n] - 2\,\hat{a}[n]\,x[n-1] + x[n-2] \tag{5}$$

For a fixed step size of $\mu[n] = 0.04$, the NDFET algorithm in equation (5) converged within 20 samples (2.5 ms) for each of the eight single tones used in DTMF signals.

This algorithm can be expanded for signals including more than one sinusoid [4, 5]:

$$\hat{a}_1[n] = \hat{a}_1[n-1] + 2\,\mu[n]\,e[n-1]\,x[n-1] \tag{6}$$

$$\hat{a}_2[n] = \hat{a}_2[n-1] + 2\,\mu[n]\,e[n-1]\,x^{'}[n-1] \tag{7}$$

$$x^{'}[n] = x[n] - 2\,\hat{a}_1[n]\,x[n-1] + x[n-2] \tag{8}$$

$$x^{''}[n] = x^{'}[n] - 2\,\hat{a}_2[n]\,x^{'}[n-1] + x^{'}[n-2] = e[n] \tag{9}$$

We use a periodic adaptive step size $\mu[n] = \mu_0/(1 + (n \bmod N))$ where N is the period and $\mu_0 = 0.04$.

V. MUSIC ALGORITHM

MUSIC detects frequencies in a signal by performing an eigen decomposition on the covariance matrix of a data vector $\mathbf{y}$ of M samples obtained from the samples of the received signal. The key to MUSIC is its data model

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{v} \tag{10}$$

where $\mathbf{v}$ is a vector of M noise samples, $\mathbf{s}$ is a vector of N signal amplitudes ($N = 2$ for DTMF tones), and $\mathbf{A}$ is the $M \times N$ Vandermonde matrix of samples of the signal frequencies. If we assume a zero-mean signal and white noise, then the covariance of $\mathbf{y}$ has the form

$$\mathbf{R_y} = E\{\mathbf{y}\mathbf{y}^H\} = \mathbf{A}\mathbf{R_s}\mathbf{A}^H + \sigma^2\mathbf{I}. \tag{11}$$

Here, $\mathbf{R_s} = E\{\mathbf{s}\mathbf{s}^H\}$ is the $N \times N$ signal autocorrelation matrix, $\mathbf{I}$ is the $M \times M$ identity matrix, and σ^2 is the noise variance. From the eigen decomposition of $\mathbf{R_y}$, we use the eigenvectors associated with the N maximum eigenvalues to define the signal subspace (the column space of $\mathbf{A}$), and use the other eigenvectors to define the noise subspace, $\mathcal{U}_v$. From the orthogonality of the signal and noise subspaces, finding the peaks in the estimator function

$$J(\omega) = \frac{1}{[\mathbf{a}(\omega)]^H \mathcal{U}_v \mathcal{U}_v^H [\mathbf{a}(\omega)]} \tag{12}$$

for various ω values yields the strongest frequencies [1], where $\mathbf{a}(\omega)$ refers to the columns of $\mathbf{A}$.

MUSIC assumes that the number of samples M and the number of frequencies N are known. The efficiency of MUSIC is the ratio of the theoretical smallest variance, given by the Cramer-Rao Lower Bound (CRLB) [11], to the variance of the MUSIC estimator:

$$\text{eff} = \text{var}_{CRLB(\hat{\omega}_i)} / \text{var}_{MUSIC(\hat{\omega}_i)} \tag{13}$$

The efficiency does not depend on the total number of samples, m, (Figure 1) but does depend on M (Figure 2). As M increases, efficiency and computation time increase. We pick $M = 8$, because larger values do not significantly improve the efficiency.

VI. MATLAB SIMULATIONS

MATLAB simulations indicate that the Goertzel algorithm does not meet standard 2(b), and that the NDFE algorithm is very fast compared to the other two but gives large estimation errors. MUSIC meets the standards on the estimation error because it is a high-resolution technique [5]. Execution times in MATLAB are: Goertzel 11.8 s, NDFET 2.1 s, and MUSIC 3.2 s. MATLAB, however, is optimized for floating-point matrix-vector calculations, and it is not necessarily a good indicator of execution times on fixed-point DSP processors.

VII. ADSP-2101 IMPLEMENTATION

When implementing the DTMF decoder algorithms on the fixed-point ADSP-2101 (16.67 MHz) processor, we sampled touch tones generated by a telephone at 8 KHz and 8 bits. For each algorithm, we compute the estimation errors for each touch tone digit and manually compared it against standards given in Section II. The required programming memory (PM) in words, data memory (DM) in words, and execution time (ET) in μs/sample follow. A word is 2 bytes for DM and 3 bytes for PM, and samples arrive every 125 μs.

- Goertzel DTMF detector hand-coded in assembly: PM = 167, DM = 51, ET = 33.2 μs/sample
- NDFET DTMF detector hand-coded in assembly: PM = 162, DM = 13, ET = 9.1 μs/sample
- MUSIC DTMF detector cross-compiled from C: PM = 2071, DM = 1824, ET = 1302 μs/sample

A. *The Goertzel Algorithm*

Table 2 shows the error percentage in detecting DTMF tones for the Goertzel algorithm on the ADSP-2101. Unfortunately, its resolution was not high enough. For example, for $N = 205$ with 8 KHz sampling rate, resolution in frequency domain is 8000/205 $=$ 39 Hz. This resolution causes an error in meeting standard 2(b). The DTMF receiver must eliminate the tones outside the specified ranges. To be more specific, consider a 770 Hz tone. Standard 2(b) enforces the elimination of frequencies outside $743.05 - 796.95$ Hz for 770 Hz. However, 39 Hz resolution enforces the detection of signals in $760.98 - 800.0$ Hz to yield 780.48 Hz, which is impossible with 39 Hz resolution in frequency domain.

Table 2: Goertzel algorithm tests on the ADSP-2101

	1209	1336	1477
697	702, 1209	702, 1326	702, 1482
(e%)	0.72, 0.00	0.72, 0.74	0.72, 0.34
770	780, 1209	780, 1326	780, 1482
(e%)	1.30, 0.00	1.30, 0.74	1.30, 0.34
852	858, 1209	858, 1326	858, 1482
(e%)	0.70, 0.00	0.70, 0.74	0.70, 0.34
941	936, 1209	936, 1326	936, 1482
(e%)	0.53, 0.00	0.53, 0.74	0.53, 0.34

B. *Normalized Direct Frequency Estimation Technique*

Table 3 shows the detected tones with NDFET implemented on the ADSP-2101. For the implementation, we used an adaptive step size over each $\mu[n] = \mu_0/n$ interval of 300 samples, $n = 1, ..., N$, with $\mu_0 = 0.04$. As shown on Table 3, estimation errors are very high since this algorithm is not robust in the presence of noise. Although NDFET is much faster than the Goertzel algorithm, it does not meet the standard 2(a).

Table 3: NDFE tests on the ADSP-2101

	1209	1336	1477
697	894, 1178	917, 1200	925, 1241
(e%)	28.29, 2.57	31.57, 10.15	32.78, 16
770	927, 1171	944, 1224	962, 1269
(e%)	20.44, 3.15	22.6, 8.38	24.98, 14.08
852	953, 1182	981, 1238	999, 1282
(e%)	11.88, 2.25	15.17, 7.31	17.25, 13.18
941	980, 1198	1017, 1256	1043, 1304
(e%)	4.18, 0.90	8.11, 6.02	10.85, 11.7

C. *MUSIC Algorithm*

We implemented MUSIC in C and then cross-compiled it to assembly. We run 20 iterations of the QR algorithm [12] to compute the eigenvalue decomposition of the covariance matrix $\mathbf{R_y}$. $\mathbf{R_y}$ is estimated by

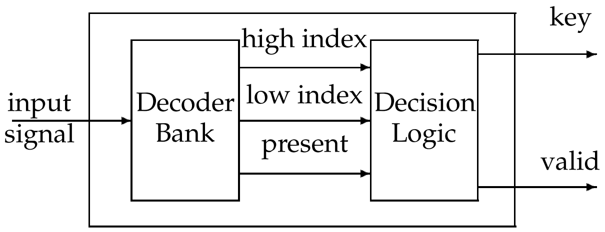
$$\frac{1}{2(m-M)} \sum_{n=M+1}^{m} \mathbf{y}[n]\mathbf{y}^H[n] + \mathbf{y}^{B*}[n]\mathbf{y}^{BT}[n] \qquad (14)$$

where B denotes the backward vector operation, with $M = 8$ and $m = 200$. The implementation meets the frequency resolution standards, and we expect to develop a MUSIC algorithm by hand that can keep up with the incoming samples.

VIII. MOTOROLA 56001 IMPLEMENTATION

We add decision logic to detect valid DTMF signals and generate DTMF decoders for the Motorola 56001 processor (24-bit, 40 MHz) using the Ptolemy software environment [6]. Ptolemy is a freely distributable, extensible, graphical block diagram environment that interfaces with compilers, assemblers, hardware synthesis tools, and other external programs. Goertzel and NDFET decoders are included in Ptolemy 0.6, due for release on April 15, 1996, via the ftp site `ptolemy.eecs.berkeley.edu`. A MUSIC decoder has not yet been implemented.

The DTMF decoders have the following structure:



The decoder bank, after some observation time, outputs the index of the low and high frequency components and whether a DTMF signal might be present. Decision logic decodes the touch tone key and decides whether or not the detection is valid. Like the ADSP implementation, processing is performed on blocks of 205 samples for Goertzel and 300 for the NDFET.

We match the intensive signal processing computations and the accompanying decision logic to the Synchronous Dataflow (SDF) model of computation [13] in Ptolemy. A valid SDF subsystem *always* has a static implementation, i.e., one that does not require any run-time scheduling. This holds true whether we ultimately generate an implementation for a DSP processor, for multiple processors, as C code, or as a VLSI chip. Using knowledge of the target implementation, we can compute whether or not the system meets the real-time constraints. We specified the DTMF decoders as SDF block diagrams and verified them through simulation, and converted them to a Motorola 56001 implementation. Extra memory and instructions are generated to handle the I/O for each block.

On the 56001, a word is 3 bytes. Samples arrive every 125 μs. Input data buffers take up 205 and 300 words, respectively, and are not included in data memory usage.

- Goertzel DTMF decoder (detector + logic): PM = 805, DM = 218, ET = 19.4 μs/sample
- NDFET DTMF decoder (detector + logic): PM = 329, DM = 273, ET = 20.9 μs/sample

IX. SUMMARY

We implemented three real-time DTMF detection schemes, and analyzed their performance. We chose an observation time (block size) to optimize the performance subject to real-time constraints. The MUSIC DTMF decoder complies with CCITT Q.24 standards, but is the most expensive to implement on an embedded DSP processor. Although easier to implement, neither the Goertzel nor the NDFET DTMF decoders can satisfy the standards, due to poor frequency resolution and high SNR requirements (50 dB), respectively. The hand-coded implementations are faster and require less memory, but the Ptolemy designs can be reused and retargeted to a variety of implementations.

REFERENCES

[1] R. Schmidt, "Multiple emitter location and signal parameter estimation," *RADC Spectrum Estimation Workshop*, pp. 243 – 258, 1979.

[2] B. Widrow and S. D. Stearns, *Adaptive Signal Proccessing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[3] P. Mock, "Add DTMF generation and decoding to DSP-μp designs," *EDN*, vol. 30, pp. 205–220, March 21 1985.

[4] J. Y. Cheung, "A direct adaptive frequency estimation technique," *Midwest Symposium on Circuits and Systems*, pp. 804 – 807, 1987.

[5] F. A. Sakarya and G. Arslan, "LMS ve MUSIC algoritmaları ile DTMF alıcı tasarımı," *Proc. 3. Sinyal İşleme ve Uygulamaları Kurultayı*, 1995.

[6] J. Pino, S. Ha, E. Lee, and J. Buck, "Software synthesis for DSP using Ptolemy," *Journal on VLSI Signal Processing*, vol. 9, pp. 7–21, Jan. 1995.

[7] CCITT Blue Book, *Recommendation Q.23: Technical Features of Push-Button Telephone Sets*. Geneva, 1989.

[8] CCITT Blue Book, *Recommendation Q.24: Multi-Frequency Push-Button Signal Reception*. Geneva, 1989.

[9] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, 1990.

[10] Analog Devices, *DSP Applications using the ADSP-2100 Family*. Prentice-Hall, 1992.

[11] P. Stoica and A. Nehorai, "MUSIC, Max. Likelihood, and Cramer-Rao Bound," *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, vol. 37, no. 5, pp. 720–741, 1989.

[12] D. S. Watkins, *Fundamentals of Matrix Computations*. John Wiley & Sons, 1991.

[13] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *Proc. of the IEEE*, Sept. 1987.

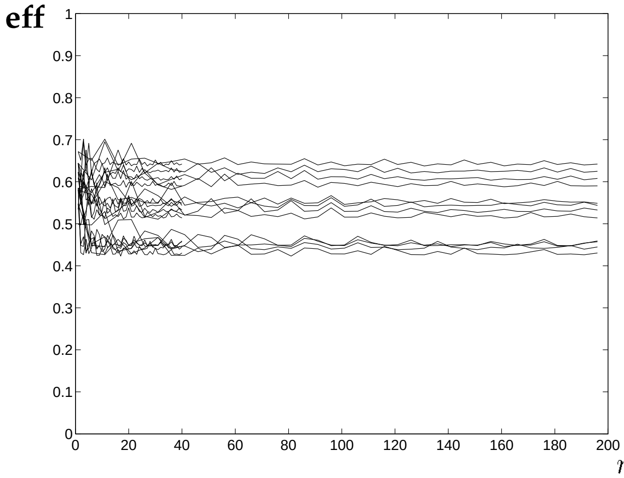


Figure 1. Effect of total number of samples m on the efficiency ratio **eff** for $M = 8$.

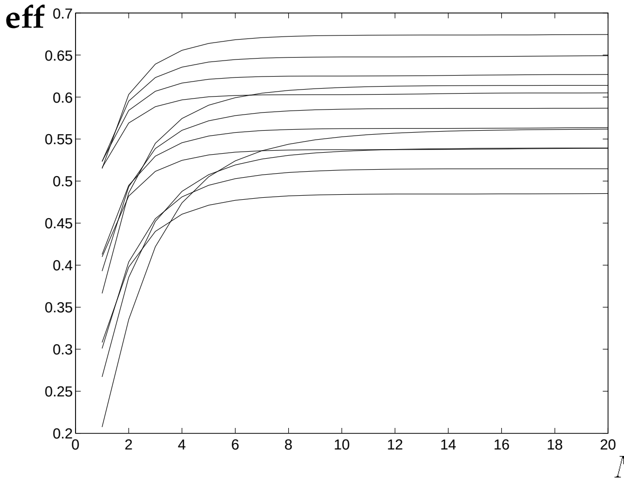


Figure 2. Change of efficiency ratio **eff** relative to the number of samples M with $m = 200$.

ACKNOWLEDGEMENT

We would like to thank Prof. Metin Yücel from Yıldız Technical University (YTÜ), Luis J. Gutierrez from University of California at Berkeley, and Onur Taçkın formerly from YTÜ, who indirectly contributed to this work.