

Implementação de um Detector DTMF utilizando o algoritmo JCO

G. Jerônimo da Silva Jr. e R. M. Campello de Souza

Resumo—Este artigo apresenta uma implementação eficiente para um detector DTMF, no sentido de reduzir a complexidade aritmética do procedimento, em relação às técnicas conhecidas. Para isso, a implementação utiliza o novo algoritmo JCO e o algoritmo de Goertzel. Resultados de simulações são apresentados.

Palavras-Chave—DTMF, algoritmo JCO, algoritmo de Goertzel, transformada discreta de Fourier, DFT.

Abstract—This paper introduces a new and efficient implementation of a DTMF detector, in the sense that it has low arithmetic complexity. The implementation is based on the new JCO algorithm and in the Goertzel algorithm. Simulation results are presented.

Keywords—DTMF, JCO algorithm, Goertzel algorithm, discrete Fourier transform, DFT.

I. INTRODUÇÃO

A transformada de Fourier é uma das ferramentas mais usadas da área de processamento de sinais [1], [2]. Uma função $X(j\Omega)$ é dita ser a transformada de Fourier da função $x(t)$ se

$$X(j\Omega) \triangleq \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt, \quad (1)$$

onde $j = \sqrt{-1}$. Nesse caso, a função $x(t)$ pode ser escrita em função de $X(j\Omega)$ por

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega. \quad (2)$$

As equações (1) e (2) são conhecidas como a transformada de Fourier e a transformada inversa de Fourier, respectivamente.

Uma de suas aplicações mais simples é a detecção de uma determinada frequência em um sinal. Com a evolução da eletrônica digital, técnicas baseadas em processamento de tempo discreto tornaram-se atrativas para resolver esse problema. Surgem então a transformada de Fourier de tempo discreto (DTFT, do inglês *discrete time Fourier transform*), a transformada discreta de Fourier (DFT, do inglês *discrete Fourier transform*) e algoritmos rápidos para sua computação, as chamadas transformadas rápidas de Fourier (FFT, do inglês *fast Fourier transform*) [3],[4]. A função $X(e^{j\omega})$ é dita ser a transformada de Fourier de tempo discreto da sequência $x[n]$ se

$$X(e^{j\omega}) \triangleq \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \quad (3)$$

G. J. da Silva Jr. e R. M. Campello de Souza, Grupo de Processamento de Sinais, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, PE, E-mails: gilsonjr@gmail.com, ricardo@ufpe.br

A sequência $x[n]$ pode ser escrita como

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n} d\omega. \quad (4)$$

As equações (3) e (4) são conhecidas como a DTFT e a DTFT inversa, respectivamente. De forma geral, a sequência $x[n]$ é obtida de uma função ou sinal $x(t)$ por $x[n] = x(Tn)$. A DFT de uma sequência $x[n]$ de N pontos, $n = 0, \dots, N - 1$, denotada por $X[k]$, é dada por

$$X[k] \triangleq \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad (5)$$

$k = 0, \dots, N - 1$, em que $W_N \triangleq e^{-j\frac{2\pi}{N}}$. A sequência $x[n]$ pode ser escrita como

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-kn}. \quad (6)$$

As equações (5) e (6) são a DFT e a DFT inversa, respectivamente. A DFT se relaciona com a DTFT pela relação [3]

$$X[k] = X(e^{j\frac{2\pi}{N}k}). \quad (7)$$

Existem algoritmos especializados na computação de uma única componente da DFT, dentre eles destacam-se o algoritmo de Goertzel [4]-[6], sliding Goertzel [7],[8] e o JCO [9]. O algoritmo de Goertzel computa a componente $X[k]$ através de uma divisão polinomial pelo polinômio $p_k(\chi)$, definido por

$$\begin{aligned} p_k(\chi) &\triangleq (\chi - W_N^k)(\chi - W_N^{-k}) \\ &= 1 - 2\cos\left(\frac{2\pi k}{N}\right)\chi + \chi^2. \end{aligned} \quad (8)$$

O resultado é um algoritmo com N multiplicações reais (supondo $x[n] \in \mathbb{R}$) [4]. A Figura 1 mostra a implementação da divisão polinomial por $p_k(\chi)$. O algoritmo JCO difere do Goertzel por utilizar o polinômio ciclotômico, $\Phi_N(\chi)$, no lugar de $p_k(\chi)$. Nesse caso, a divisão polinomial não tem multiplicações, mas são necessárias $2[\phi(L) - 1]$ multiplicações finais, onde $\phi(\cdot)$ é a função aritmética de Euler e L é a ordem de W_N^k [10],[9]. Esses algoritmos tem como uma de suas principais aplicações a detecção de sinais DTMF (do inglês *Dual Tone MultiFrequency*). Esses sinais são largamente utilizados em telefonia digital e podem ser reproduzidos, de forma audível, em qualquer aparelho celular. O objetivo é implementar de forma eficiente o detector DTMF, no sentido de reduzir o número de multiplicações e adições do sistema.

O processamento digital de um detector DTMF é mostrado na Figura 2. Nele existem um filtro anti-sobreposição seguido

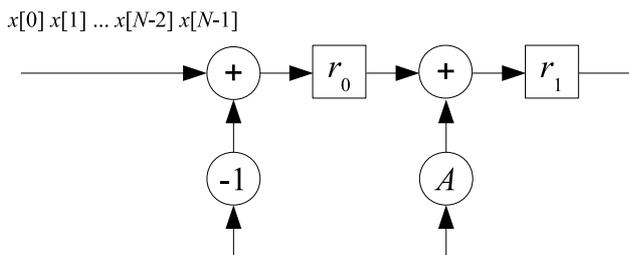


Fig. 1. Filtro autoregressivo para computar a divisão polinomial por $p_k(\chi)$. A componente $x[N-1]$ é a primeira a entrar no registrador de deslocamento e $A = 2\cos(2\pi k/N)$. A componente da DFT é computada por $X[k] = r_0 + r_1 W_N^k$.

de um processamento digital para a detecção do tom DTMF. A Tabela I mostra os caracteres usados em um detector DTMF em função das frequências do sinal. O tom de um determinado caractere é produzido pelo sinal $v(t) = A[\cos(2\pi f_1 t) + \sin(2\pi f_c t)]$, onde f_1 e f_c são, respectivamente, as frequências de linha e coluna do caractere.

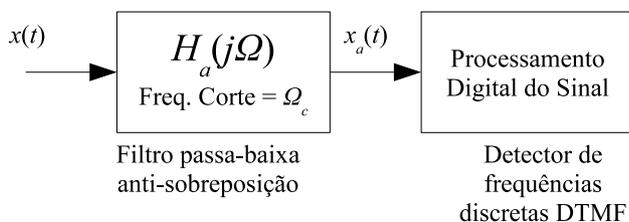


Fig. 2. Diagrama geral de um detector DTMF utilizando processamento de tempo discreto.

TABELA I

CARACTERES DTMF EM FUNÇÃO DAS FREQUÊNCIAS DO SINAL

$f_l \setminus f_c$ (Hz)	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Este artigo utiliza a seguinte notação:

t : variável temporal em segundos (s);

Ω : frequência contínua em rads/s;

$x(t)$: função ou sinal de entrada;

$X(j\Omega)$: transformada de Fourier de $x(t)$, definida conforme a relação (1);

$H_a(j\Omega)$: resposta em frequência do filtro passa-baixa de anti-sobreposição;

Ω_c : frequência de corte do filtro anti-sobreposição em rads/s, considerando que $H_a(j\Omega) = 0$ para $|\Omega| \geq \Omega_c$. Também define-se $f_c \triangleq \Omega_c/(2\pi)$;

$x_a(t)$: saída do filtro anti-sobreposição;

$X_a(j\Omega)$: transformada de Fourier de $x_a(t)$;

$x[n]$: sequência de N pontos, obtida por $x[n] = x_a(Tn)$, para $n = 0, \dots, N-1$, sendo T o período de amostragem em segundos;

$X(e^{j\omega})$: DTFT de $x[n]$, definida conforme a relação (3);

$X[k]$: DFT de $x[n]$, definida conforme a relação (5);

τ : tempo total de processamento, $\tau \triangleq NT$;

Ω_0 : frequência a ser analisada em rads/s, $\Omega_0 < \Omega_c$. Também define-se $f_0 \triangleq \Omega_0/(2\pi)$.

Este artigo está organizado da seguinte forma: Na Seção II, são apresentadas duas implementações utilizando-se o algoritmo JCO, uma livre de multiplicações e outra com apenas duas multiplicações, para detectar a frequência f_0 em um sinal de entrada $x(t)$. Na Seção III, demonstra-se como utilizar a implementação da Seção II para construir um detector DTMF completo. Na Seção IV, são mostrados resultados de simulações utilizando-se várias entradas. As conclusões do trabalho são apresentadas na Seção V.

II. DETECÇÃO DE UMA FREQUÊNCIA

O diagrama para detecção de uma frequência está mostrado na Figura 3. A sequência $x[n]$ é obtida através da amostragem de N pontos de $x_a(t)$ com período de amostragem

$$T = \frac{1}{Mf_0} = \frac{2\pi}{M\Omega_0}, \quad (9)$$

com $M \in \mathbb{N}^+$. Ao mesmo tempo da amostragem, o bloco JCO computa uma componente da DFT utilizando o algoritmo JCO.

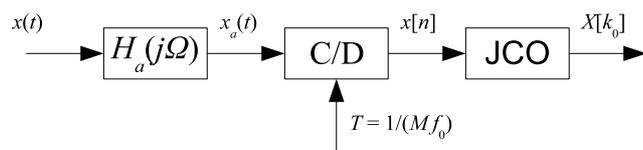


Fig. 3. Diagrama de um processamento digital para detecção de uma frequência. A sequência $x[n]$ é obtida através da amostragem de N pontos de $x_a(t)$ com período de amostragem $T = 1/(Mf_0)$. A frequência é detectada computando-se uma componente da DFT com a implementação JCO.

A primeira condição que deve ser obedecida é o critério de Nyquist,

$$M > 2\frac{f_c}{f_0} = 2\frac{\Omega_c}{\Omega_0}. \quad (10)$$

Isso evita o problema de sobreposição da frequência [3], garantindo assim a relação

$$X(e^{j\omega}) = \frac{1}{T} X_a(j\omega/T). \quad (11)$$

Utilizando (7), pode-se escrever

$$X[k] = \frac{1}{T} X_a(j\frac{2\pi}{NT}k). \quad (12)$$

Considerando (9) e o fato de que $X_a(j\Omega) = H_a(j\Omega)X(j\Omega)$, tem-se

$$X(j\Omega_0 \frac{M}{N}k) = \frac{2\pi}{M\Omega_0 H_a(j\Omega_0 \frac{M}{N}k)} X[k]. \quad (13)$$

Agora, escolhe-se N múltiplo de M , isto é,

$$N = k_0 M, \quad (14)$$

onde $k_0 \in \mathbb{N}^+$. Fazendo $k = k_0$ na Equação (13), tem-se

$$X(j\Omega_0) = \frac{2\pi}{M\Omega_0 H_a(j\Omega_0)} X[k_0]. \quad (15)$$

A frequência Ω_0 pode ser detectada computando-se a componente k_0 de uma DFT de N pontos. Agora, pode-se projetar o filtro digital que implementa o algoritmo JCO.

A computação da componente $X[k_0]$, pode ser feita utilizando o filtro IIR

$$H(z) = \frac{1}{1 - W_N^{-k_0} z^{-1}}, \quad (16)$$

com entrada $x[n]$, $n = 0, \dots, N-1$ e saída $y[n]$ [3],[6]. A componente de interesse é computada por

$$X[k_0] = y[N]. \quad (17)$$

A implementação do algoritmo JCO é feita escrevendo-se $H(z)$ como

$$H(z) = \frac{\prod_{\text{ord}(W_N^i)=L, i \neq k_0} (1 - W_N^{-i} z^{-1})}{\prod_{\text{ord}(W_N^i)=L, i \neq k_0} (1 - W_N^i z^{-1})}, \quad (18)$$

que pela definição de polinômio ciclotômico [4],[11], torna-se

$$H(z) = \frac{\prod_{\text{ord}(W_N^i)=L, i \neq k_0} (1 - W_N^{-i} z^{-1})}{\Phi_L(z^{-1})}. \quad (19)$$

Os graus do denominador e do numerador são $\phi(L)$ e $\phi(L)-1$, respectivamente. Dessa forma, $H(z)$ pode ser expresso na forma

$$H(z) = \frac{1 + a_1 z^{-1} + \dots + a_{\phi(L)-1} z^{-\phi(L)+1}}{1 + b_1 z^{-1} + \dots + z^{-\phi(L)}}. \quad (20)$$

A Figura 4 mostra a implementação em hardware do algoritmo JCO no caso geral.

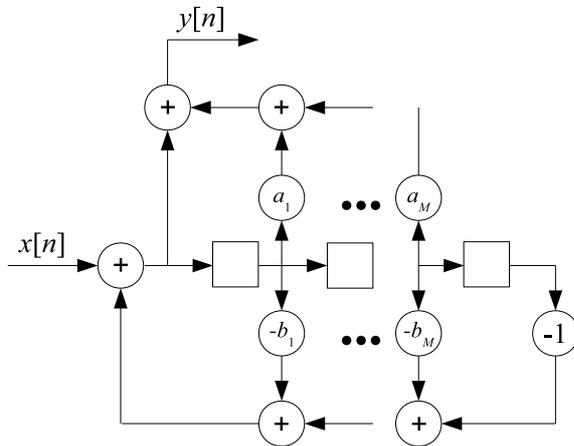


Fig. 4. Diagrama do filtro que implementa o algoritmo JCO. A componente da DFT é computada por $X[k_0] = y[N]$.

A complexidade multiplicativa do JCO é proporcional a

$$L = \phi\left(\frac{N}{\text{MDC}(N, k_0)}\right), \quad (21)$$

e k_0 divide N , pois $N/k_0 = M \in \mathbb{N}^+$. Nesse caso, a Equação (21) torna-se

$$L = \phi(M). \quad (22)$$

Quanto menor o valor de M , menor a complexidade do procedimento. Entretanto, o critério de Nyquist apresentado

na Inequação (10) impõe uma restrição no projeto. Filtros com Ω_0 próximo a Ω_c são mais difíceis de implementar, o que implica que diminuir a complexidade do algoritmo JCO significa aumentar a complexidade do filtro anti-sobreposição. Tipicamente, o filtro anti-sobreposição é um filtro Butterworth de primeira ordem, implementado com capacitores e resistores [12]. É necessário, portanto, escolher de forma adequada o parâmetro M , de acordo com o projeto. A seguir, são apresentadas duas escolhas eficientes para o parâmetro M .

A. Implementação JCO com $M = 4$

Nesse caso, o filtro digital que implementa o JCO, utilizando (19), é

$$H(z) = \frac{1 + jz^{-1}}{\Phi_4(z^{-1})} = \frac{1 + jz^{-1}}{1 + z^{-2}}. \quad (23)$$

que coincide com o filtro encontrado utilizando-se o algoritmo de Goertzel. A implementação está mostrada na Figura 5. Este projeto computa a componente $X[k_0]$ utilizando apenas N adições com a restrição de que $\Omega_0 < \Omega_c < 2\Omega_0$.

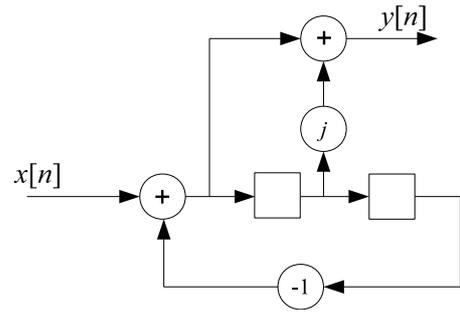


Fig. 5. Implementação do algoritmo JCO com $M = 4$. Existem apenas multiplicações triviais.

B. Implementação JCO com $M = 8$

Nesse caso, o filtro JCO é

$$H(z) = \frac{(1 - W_8 z^{-1})(1 - W_8^3 z^{-1})(1 - W_8^5 z^{-1})}{\Phi_8(z^{-1})}, \quad (24)$$

que leva a

$$H(z) = \frac{1 + \frac{\sqrt{2}}{2}(1+j)z^{-1} + jz^{-2} - \frac{\sqrt{2}}{2}(1-j)z^{-3}}{1 + z^{-4}}. \quad (25)$$

A implementação desse filtro é mostrada na Figura 6. Esse projeto utilizando o algoritmo JCO computa a componente $X[k_0]$ utilizando apenas duas multiplicações e aproximadamente N adições, o que é muito mais eficiente do que o algoritmo de Goertzel, que computa a mesma componente utilizando aproximadamente N multiplicações e $2N$ adições. A restrição para a frequência de corte do filtro anti-sobreposição, neste caso, é $\Omega_0 < \Omega_c < 4\Omega_0$.

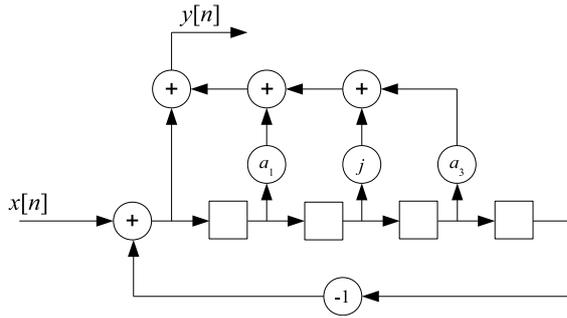


Fig. 6. Implementação do algoritmo JCO com $M = 8$, $a_1 = \frac{\sqrt{2}}{2}(1 + j)$ e $a_3 = \frac{\sqrt{2}}{2}(-1 + j) = -a_1^*$.

III. DETECTOR DTMF

O detector DTMF é implementado de acordo com a Figura 7. Como é utilizado apenas um filtro anti-sobreposição, nem todos os filtros JCO podem ser tipo $M = 4$ (observando as restrições para todos). Outra observação é que os tempos de computação τ devem ser compatíveis, de modo que o processamento de todas as frequências termine simultaneamente; para isso basta escolher $k_i \approx \tau f_i$ com $k_i \in \mathbb{N}^+$. Assim é necessário escolher o número de pontos N_i para cada frequência f_i analisada.

Adotando $\tau = 40ms$, que é um valor encontrado nos circuitos integrados comerciais, temos a Tabela II com os parâmetros do projeto do detector DTMF para $f_c = 2,4kHz$. A frequência de corte do filtro anti-sobreposição deve satisfazer a condição de projeto para todas as frequências.

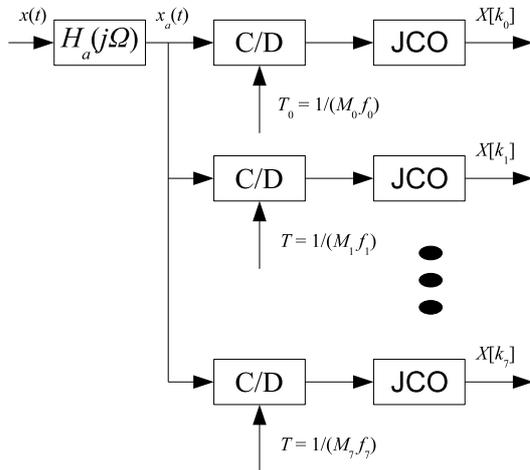


Fig. 7. Implementação do detector DTMF utilizando o algoritmo JCO.

O projeto detecta todas as frequências DTMF num sinal $x(t)$ com duração de $40ms$ com aproximadamente 10 multiplicações, 2130 adições e 26 registradores utilizados pelos filtros JCO (5 filtros de ordem 4 e 3 filtros de ordem 2). As frequências são detectadas utilizando-se a Equação (15), que pode ser reescrita por

$$X(j\Omega_i) = \frac{2\pi}{M_i \Omega_i H_a(j\Omega_i)} X[k_i]. \quad (26)$$

TABELA II

PARÂMETROS DE PROJETO DO DETECTOR DTMF IMPLEMENTADO UTILIZANDO-SE O ALGORITMO JCO: A FREQUÊNCIA ANALISADA f_i , O ÍNDICE DA COMPONENTE DFT $k_i \approx f_i \tau$, O TEMPO DE COMPUTAÇÃO $\tau_i = k_i / f_i$, O PARÂMETRO M_i E O NÚMERO DE PONTOS AMOSTRADOS POR COMPUTAÇÃO $N_i = M_i k_i$, ASSUMINDO $f_c = 2,4kHz$.

f_i (Hz)	k_i	τ_i (ms)	M_i	N_i
697	27	38,7	8	216
770	31	40,3	8	248
852	34	39,9	8	272
941	38	40,4	8	304
1209	48	39,7	8	384
1336	53	39,7	4 / 8	212 / 424
1477	59	39,9	4 / 8	236 / 472
1633	65	39,8	4 / 8	260 / 520

A Tabela III apresenta uma comparação da complexidade aritmética do projeto entre os algoritmos JCO e Goertzel.

TABELA III

COMPARAÇÃO DA COMPLEXIDADE ARITMÉTICA DO PROJETO DTMF UTILIZANDO OS ALGORITMOS JCO E GOERTZEL. AS MULTIPLICAÇÕES CONTABILIZADAS SÃO TODAS NÃO TRIVIAIS. CONSIDERA-SE QUE $x(t) \in \mathbb{R}$ E $f_c = 2,4kHz$.

Complexidade	Goertzel	JCO
Adições	3556	2131
Multiplicações	1424	10
Registradores	16	26

IV. SIMULAÇÃO E RESULTADOS

Foram feitos dois tipos de simulação, um com sinais sintetizados, gerados digitalmente, e outro com sinais obtidos através de um microfone, sujeito a ruído e interferência. No primeiro teste, não há necessidade de se utilizar um filtro anti-sobreposição. No segundo teste, foi utilizado um filtro digital para simular o filtro anti-sobreposição. A seguir, são descritos alguns detalhes dos dois testes.

A. Teste com Sinais Sintetizados

Neste teste, um sinal de entrada $x(t)$ foi gerado pela relação

$$x(t) = \cos(2\pi 697t) + \sin(2\pi 1477t).$$

O sinal $x(t)$ foi inserido na estrutura da Figura 7 com $H_a(j\Omega) = 1$. As frequências foram computadas por (26). O resultado é apresentado no gráfico da Figura 8, em que é possível detectar claramente as duas frequências do sinal.

B. Teste com Sinais Amostrados de Ambiente com Ruído

Neste teste, foi utilizada uma sala com condicionador de ar e ventilador ligado e um celular para reproduzir o som da tecla 5 ($f_l = 770Hz$ e $f_c = 1336Hz$) em um microfone ligado a um computador pessoal. O sinal foi amostrado a uma taxa de $44100Hz$. Após o sinal ser amostrado, o filtro anti-sobreposição foi aplicado de forma digital, isto é, o filtro $H_a(j\Omega)$ é simulado. O ideal seria se utilizar um filtro analógico antes da amostragem. O filtro simulado escolhido foi

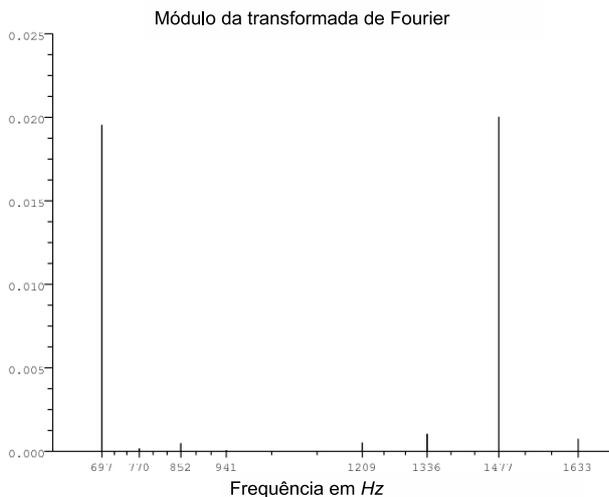


Fig. 8. Resultado do teste com o sinal sintetizado $x(t) = \cos(2\pi 697t) + \sin(2\pi 1477t)$ e $H_a(j\Omega) = 1$. O gráfico mostra a magnitude da transformada de Fourier, computada por (26), em função da frequência f_i .

um filtro de Butterworth de primeira ordem com frequência de corte $f_c = 2,4\text{kHz}$, isto é,

$$H_a(j\Omega) = \frac{2\pi f_c}{2\pi f_c + j\Omega}.$$

O resultado dessa simulação está mostrado na Figura 9. Mesmo sem implementar o filtro anti-sobreposição, é possível detectar as duas frequências que caracterizam a tecla 5 do DTMF.

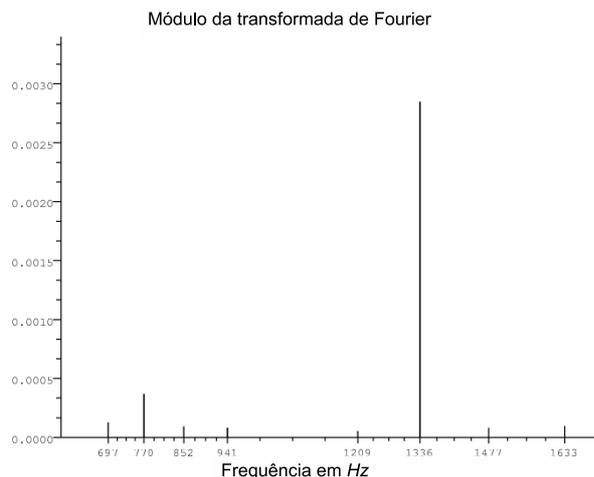


Fig. 9. Resultado do teste com o sinal obtido de um ambiente com ruído. O gráfico mostra a magnitude da transformada de Fourier, estimada pela Equação (26), em função da frequência f_i .

V. CONCLUSÕES

Esse artigo apresentou uma maneira eficiente de se implementar uma detecção de frequência de um sinal analógico

utilizando o algoritmo JCO, que para determinadas situações torna-se igual ao algoritmo de Goertzel. Também foi projetado um detector DTMF com essa implementação e simulações foram realizadas para demonstrar a validade da estrutura no processamento de sinais analógicos.

O critério usado para se caracterizar a eficiência do algoritmo foi o de reduzir o número de multiplicações e adições dos filtros digitais. Isso é importante para diminuir a complexidade do *hardware*, diminuir o número de operações por segundo e, conseqüentemente, diminuir a energia consumida em um processamento.

A detecção de sinais DTMF pode ser feita de outras maneiras [13],[14], aparentemente, não mais eficientes do que a apresentada nesse trabalho.

Embora este artigo mostre uma aplicação para detecção de sinais DTMF, é possível utilizar o algoritmo JCO em muitas outras aplicações, tais como análise espectral, filtragem de frequências específicas, estimação de resposta em frequência e equalização, entre outras.

AGRADECIMENTOS

Os autores agradecem ao Prof. Dr. Hélio M. de Oliveira por suas valiosas contribuições ao presente trabalho. O primeiro autor agradece ao CNPq pelo apoio recebido durante a realização desse trabalho.

REFERÊNCIAS

- [1] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, 2nd ed. Prentice Hall, 1996.
- [2] H. M. de Oliveira, *Análise de Fourier e Wavelets: sinais estacionários e não estacionários*. Univesitária da UFPE, 2007.
- [3] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.
- [4] R. E. Blahut, *Fast Algorithms for Digital Signal Processing*. Addison-Wesley Publishing Company, 1984.
- [5] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *The American Mathematical Monthly*, vol. 65, no. 1, pp. 34–35, 1958.
- [6] J. Beraldin and W. Steenaert, "Overflow analysis of a fixed-point implementation of the Goertzel algorithm," *Circuits and Systems, IEEE Transactions on*, vol. 36, no. 2, pp. 322–324, Feb 1989.
- [7] J. F. Chicharo and M. T. Kilani, "A sliding Goertzel algorithm," *Signal Processing*, vol. 52, no. 3, pp. 283 – 297, 1996.
- [8] R. Garcia-Retegui, S. Gonzalez, M. Funes, and S. Maestri, "Implementation of a novel synchronization method using sliding Goertzel DFT," *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pp. 1–5, Oct. 2007.
- [9] G. J. da Silva Jr., R. M. C. de Souza, and H. M. de Oliveira, "New algorithms for computing a single component of the discrete Fourier transform," *10th International Symposium on Communication Theory and Application, ISCTA'09*, July 2009, aceito para apresentação.
- [10] D. M. Burton, *Elementary Number Theory*, 6th ed. McGraw-Hill, 2007.
- [11] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*. Kluwer Academic Publishers, 1987.
- [12] A. S. Sedra and K. C. Smith, *Microeletrônica*, 4th ed. MAKRON Books, 2000.
- [13] R. Beck, A. Dempster, and I. Kale, "Finite-precision Goertzel filters used for signal tone detection," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 7, pp. 691–700, Jul 2001.
- [14] V. Vassilevsky, "Efficient multitone detection [dsp tips & tricks]," *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 144–147, March 2007.