

JUNIO CESAR FERREIRA

**DESEMPENHO COMPARATIVO DAS TRANSFORMADAS DE
FOURIER E HARTLEY E APLICAÇÃO EMBARCADA PARA
ANÁLISE DE VIBRAÇÕES MECÂNICAS**

**Trabalho de Conclusão de Curso
apresentado como exigência parcial, para
obtenção de grau no curso de Engenharia
Mecatrônica, da Universidade de Franca.**

**Orientador(a): Prof. Dr. Maurício Garcia
Chiarello**

**FRANCA
2016**

JUNIO CESAR FERREIRA

DESEMPENHO COMPARATIVO DAS TRANSFORMADAS DE FOURIER E HARTLEY
E APLICAÇÃO EMBARCADA PARA ANÁLISE DE VIBRAÇÕES MECÂNICAS

Orientador: _____

Nome: Prof. Dr. Maurício García Chiarello
Instituição: Universidade de Franca, UNIFRAN.

Examinador(a): _____

Nome: Profa. Dra. Lucinda Maria de Fátima Rodrigues Coelho
Instituição: Centro Universitário Municipal de Franca, Uni-FACEF.

Examinador(a): _____

Nome: Prof. Dr. Antônio Carlos Marangoni
Instituição: Universidade de Franca, UNIFRAN.

Franca, 06/12/2016

DEDICO este estudo aos meus pais, que sempre acompanharam de perto minha jornada escolar e acadêmica, oferecendo todo o apoio e colocando sempre meu aprendizado e minha formação profissional como prioridade em suas vidas.

AGRADEÇO:

Aos meus pais, pelo amor, incentivo e apoio incondicional.

À minha namorada Amanda, por ter sido minha grande companheira nesta fase da vida.

Aos professores Henrique José da Silva e Ricardo David por seu apoio e incentivo de longa data, e em particular durante esta graduação.

Agradeço especialmente ao professor Maurício García Chiarello por seus ensinamentos e por sua paciência, que me permitiram chegar até aqui.

“Uma grande descoberta envolve a solução de um grande problema, mas há uma semente de descoberta na solução de qualquer problema. Seu problema pode ser modesto; porém, se ele desafiar a sua curiosidade e fizer funcionar sua capacidade inventiva, e caso você o resolva sozinho, então você poderá experimentar a tensão e o prazer do triunfo da descoberta.”

George Polya

RESUMO

FERREIRA, Junio Cesar. **Desempenho comparativo das Transformadas de Fourier e Hartley e aplicação embarcada para análise de vibrações mecânicas.** 2016. 113 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade de Franca, Franca.

Este trabalho avalia o desempenho comparativo das transformadas rápidas de Fourier e de Hartley, servindo-se do algoritmo Cooley-Tukey *radix-2* implementado no microcontrolador ESP8266. No desempenho comparativo considerou-se o tempo de execução do algoritmo para comprimentos diferentes dos dados de entrada. Os resultados obtidos apresentam um ganho na velocidade de processamento da ordem de 34% com o emprego da transformada rápida de Hartley em relação à de Fourier. Foi desenvolvido também um protótipo de analisador de vibrações mecânicas empregando a transformada rápida de Hartley e o acelerômetro MPU6050.

Palavras-chave: Transformadas rápidas; Transformada de Hartley; Transformada de Fourier; Algoritmo Cooley-Tukey; Análise de Vibrações.

ABSTRACT

FERREIRA, Junio Cesar. **Comparative performance of Fourier Transforms and Hartley Transforms with embedded application for analysis of mechanical vibrations.** 2016. 113 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade de Franca, Franca.

This study evaluates the comparative performance of Fourier and Hartley's rapid transformed using the radix-2 Cooley-Tukey algorithm implemented in ESP8266 microcontroller. In the comparative performance, we were considered the execution time of the algorithm for different lengths of the input data. The results indicate a gain of 34% in processing speed with use of fast Hartley transform in relation to the fast Fourier transform. In addition it was developed a prototype mechanical vibration analyzer employing the fast Hartley transform and the MPU6050 accelerometer.

Keywords: Fast transform; Hartley transform; Fourier transform; Cooley-Tukey algorithm; Vibration Analysis.

SUMÁRIO

INTRODUÇÃO	10
1 REVISÃO DE LITERATURA	13
1.1 TRANSFORMADAS DE FOURIER E HARTLEY	13
1.1.1 Função Delta de Dirac e Distribuições	13
1.1.2 Teoria dos Sinais	15
1.1.3 Séries de Fourier	16
1.1.4 Transformada de Fourier	19
1.1.5 Transformada de Hartley	20
1.1.6 Algumas transformadas e teorema da derivada	22
1.2 TRANSFORMADAS DISCRETAS	25
1.2.1 Transformada Discreta de Fourier	25
1.2.2 Transformada Discreta de Hartley	26
1.2.3 Janelamento	26
1.2.4 Complexidade de Algoritmos	28
1.2.5 Algoritmos Rápidos para Transformadas Discretas	30
1.3 ANÁLISE DE VIBRAÇÕES MECÂNICAS	39
1.3.1 Fundamentos de vibrações	39
1.3.2 Vibrações de meios contínuos	44
1.3.3 Sensores de vibração	45
2 DESEMPENHO COMPARATIVO ENTRE AS TRANSFORMADAS	47
2.1 MICROCONTROLADOR ESP8266 E IDE ARDUINO	47
2.2 DESEMPENHO DAS TRANSFORMADAS DISCRETAS	50
2.2.1 Tempo de processamento	55
2.2.2 Consumo de memória	57
2.2.3 Comparação com a transformada contínua	59
2.3 DESEMPENHO DAS TRANSFORMADAS RÁPIDAS	63
2.3.1 Tempo de processamento	67
2.3.2 Consumo de memória	71

2.3.3	Validação dos resultados	72
3	ANALISADOR DE VIBRAÇÕES MECÂNICAS	75
3.1	ACELERÔMETRO.....	76
3.2	<i>HARDWARE</i> UTILIZADO	78
3.3	PLATAFORMAS EXPERIMENTAIS	81
3.3.1	Estrutura mecânica com atuadores	81
3.3.2	Sistema Massa-mola com amortecimento	83
3.4	CONSIDERAÇÕES DE <i>FIRMWARE</i>	84
3.4.1	Leitura do sensor	85
3.4.2	Janelamento, janela de Hamming e FHT	87
3.4.3	Envio de dados	88
3.5	RESULTADOS EXPERIMENTAIS	89
3.5.1	Análise de vibração em estruturas mecânicas	89
3.5.2	Análise de vibração em sistema massa-mola-amortecedor	93
3.5.2.1	Resultado referente à aplicação de um impulso	95
3.5.2.2	Resultado referente à aplicação excitação senoidal	98
	CONCLUSÃO.....	100
	CONSIDERAÇÕES FINAIS.....	103
	REFERÊNCIAS	104
	APÊNDICE A – Teorema de convolução para transformada de Fourier	108
	APÊNDICE B – Resposta de um sistema massa-mola amortecido a um impulso	110
	APÊNDICE C – Espectro de frequências do sistema massa-mola amortecido em resposta a função impulso	112

INTRODUÇÃO

Atualmente, existe uma infinidade de aplicações móveis e embarcadas que fazem o uso de algoritmos de processamento de sinais, para, por exemplo, reconhecimento de padrão em imagens, som ou outros sinais provenientes de sensores diversos. A evolução constante da eletrônica/informática promove um aumento incessante da capacidade de processamento dos processadores e microcontroladores, pode parecer, enganosamente, não haver necessidade de algoritmos rápidos. No entanto, programadores, cientistas e engenheiros estão sempre procurando formas de aperfeiçoar o uso dos algoritmos, tornando-os mais eficientes de forma a consumir o mínimo possível de processamento e memória. Pois assim, com o avanço da tecnologia, problemas mais complexos podem ser computados em menor tempo.

Na decomposição de sinais para o domínio da frequência, uma ferramenta matemática largamente conhecida e empregada é a transformada de Fourier. Esta transformada decompõe o sinal em suas componentes harmônicas. Em sua forma discreta, a transformada de Fourier possui o inconveniente de exigir muito tempo de processamento, além de operar com números complexos. Visando solucionar este problema, foi desenvolvida a *Fast Fourier Transform* (FFT), ou Transformada rápida de Fourier, que reduziu consideravelmente o tempo de processamento para decomposição de sinais. No processamento da FFT, embora o número total de operações tenha sido reduzido, estas ainda se realizam sobre números complexos. A transformada de Hartley, também empregada na decomposição de sinais, não utiliza números complexos, o que reduz a complexidade computacional. Com algumas modificações, muitos dos algoritmos utilizados para computar a FFT também podem ser utilizados para a Transformada rápida de Hartley ou *Fast Hartley Transform* (FHT). Neste trabalho, buscamos uma relação teórica entre as duas transformadas e a comparação de eficiência computacional de suas transformadas rápidas aplicadas a um sistema embarcado, cuja unidade de processamento é um microcontrolador ESP8266. Para a comparação, foram simulados sinais unidimensionais analíticos, de forma a obter uma comparação de fidelidade da resposta além da eficiência computacional.

Os trabalhos de Cunha (2013) e Castilho (2005) estabelecem algumas comparações entre os resultados da eficiência computacional da FFT e da FHT, sendo que o primeiro estabelece uma comparação entre os algoritmos Cooley-Tukey *radix-2*, *radix-4* e *Split-radix*, enquanto que o segundo estabelece uma comparação entre o Cooley-Tukey *radix-2 decimation in time* (DIT) e o DIT-FHT Cooley-Tukey *radix-2*. Nestes dois trabalhos, os experimentos foram realizados em computadores com sistemas operacionais Windows e Linux. Nestes experimentos, perceberam-se algumas vantagens no uso da FHT em relação à FFT. Estas vantagens se verificam principalmente em termos de velocidade de processamento, porém, em alguns resultados apresentados em Castilho (2005), a transformada discreta de Fourier (DFT) se mostrou mais eficiente que a transformada discreta de Hartley (DHT).

Piccinin (1988) realiza uma revisão dos algoritmos Cooley-Tukey *radix-2* para FFT e FHT, propondo que o uso da FHT para computar a FFT pode apresentar algum ganho computacional. Em Rao, Kim e Hwang (2010) é comentado que a FHT pode ser utilizada como forma de aumentar a velocidade de computação da DFT.

No IEEE, encontra-se o *conference paper* de Orintara (2002), onde se propõe uma teoria unificada para computar as transformadas de Fourier, Hartley, Cossenos e Senos através da FFT. Esta teoria se justifica nas situações em que se faz uso de microcontroladores que possuem um núcleo próprio para computar a FFT.

É possível encontrar no *Github*¹ algumas bibliotecas de FFT para aplicações microcontroladas. Algumas destas aplicações fazem uso direto ou indireto da FHT, porém ainda existe carência de informações sobre a FHT e seu uso, especialmente em microcontroladores.

O presente trabalho estabelece comparações de eficiência computacional entre as transformadas rápidas de Hartley e de Fourier implementadas em um microcontrolador que não possui um núcleo próprio para computar a FFT – o ESP8266. Os resultados obtidos permitem concluir que o uso da FHT é vantajoso, fornecendo um ganho na velocidade de processamento bastante significativo, da ordem de 34% em relação à FFT.

No capítulo 1, é apresentada uma revisão teórica sobre as transformadas de Fourier e de Hartley. Inicialmente, se apresenta uma formalização matemática para a compreensão das transformadas de Fourier e Hartley, além de se demonstrar alguns resultados

¹ Repositório *on-line* de livre acesso para compartilhamento de projetos e códigos.

importantes, que serão úteis a comparação entre resultados teóricos e práticos. Em seguida, são apresentadas as transformadas discretas e algumas de suas propriedades computacionais, fazendo uma breve revisão sobre a análise de complexidade computacional. Na sequência, é apresentado o princípio de funcionamento do algoritmo Cooley-Tukey para a transformada rápida de Fourier, assim como o funcionamento do Cooley-Tukey *radix-2* aplicado à FFT e a FHT. No último item, são revisados alguns conceitos básicos de análise de vibrações mecânicas, assim como o princípio de funcionamento dos sensores de vibrações do tipo acelerômetro.

O capítulo 2 inicia-se com uma apresentação das características da plataforma utilizada para a comparação de desempenho entre as transformadas. Em seguida, são apresentados os resultados de desempenho das transformadas discretas, utilizando algoritmos otimizados, mas ainda de ordem assintótica $O(n^2)$. Uma comparação em termos de tempo de processamento e memória consumida é realizada. Nesta seção, também é apresentada a comparação entre os resultados obtidos pela transformada discreta em relação à transformada teórica em tempo contínuo. Finalizando o capítulo, apresentam-se algumas considerações para a codificação dos algoritmos e os resultados de desempenho comparativo entre as transformadas rápidas de Fourier e Hartley.

O capítulo 3 é dedicado a alguns experimentos de aplicação em análise de vibrações, através de duas plataformas mecânicas com um circuito eletrônico, concebidas para análise de vibrações mecânicas. Apresenta-se o desenvolvimento de um dispositivo para análise de vibrações mecânicas, utilizando como sensor um acelerômetro do tipo *Micro Electro-Mechanical Systems* (MEMS) – o MPU6050 e um sistema microcontrolado de baixo custo. Dispositivos como esse são extremamente interessantes para auxiliar na modelagem de sistemas vibratórios ou para aplicações em manutenção preditiva mediante análise espectral. Ao final, apresentam-se alguns resultados espectrais para diferentes tipos de vibrações forçadas aplicadas a uma estrutura mecânica que simula uma máquina qualquer. Além disso, são apresentados resultados da análise de vibrações mecânicas de um alto-falante, modelado como um sistema massa-mola-amortecedor.

1 REVISÃO DE LITERATURA

1.1 TRANSFORMADAS DE FOURIER E HARTLEY

Segundo Stewart (2013, p. 117), a transformada de Fourier é de extrema importância para a tecnologia atual, por ser uma ferramenta que decompõe um sinal em frequências componentes que podem ser usadas para analisar os padrões, levar-lhes ordem, extrair características importantes ou remover ruído aleatório. Nesta subseção é apresentada a teoria das transformadas de Fourier e Hartley em tempo contínuo.

1.1.1 Função Delta de Dirac e Distribuições

Uma estrutura matemática necessária à definição da transformada de Fourier é a classe das distribuições ou funções generalizadas, construídas a partir da função delta de Dirac.

A função delta de Dirac, também conhecida como função impulso unitário, pode ser definida de vários modos. Usualmente ela é expressa pela relação encontrada em Hsu (1972, p. 37), representada pelas equações (1) e (2).

$$\delta(t) = \begin{cases} 0 & \forall t \neq 0 \\ \infty & t = 0 \end{cases} \quad (1)$$

$$\int_{-\infty}^{\infty} \delta(t) dt = \int_{-\varepsilon}^{\varepsilon} \delta(t) dt = 1, \quad \varepsilon > 0 \quad (2)$$

Sendo usualmente representada conforme o gráfico da figura 1.

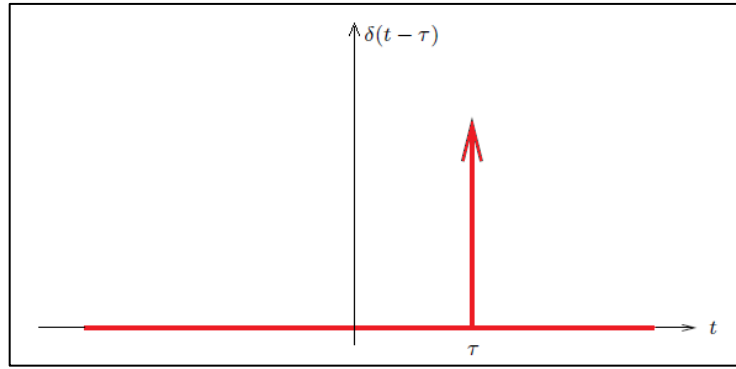


Figura 1 – Gráfico função delta de Dirac, $\delta(t - \tau)$.

Fonte: TAKAHASHI, 2002, p. 20.

Em Figueiredo (2014, p. 72), encontra-se a definição:

Se $\psi(t)$ for uma função real contínua que se anule fora de um intervalo limitado, então:

$$\int_{-\infty}^{\infty} \delta(t) \psi(t) dt = \psi(0) \quad (3)$$

A função generalizada $\mathcal{U}(t)$ definida pela equação (4) é conhecida como função unitária de Heaviside ou função degrau unitário.

$$\mathcal{U}(t) = \int_{-\infty}^t \delta(t) dt = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases} \quad (4)$$

Segundo Takahashi (2002 apud RUDIN, 1993, p. 20), as distribuições, cujos domínios são subconjuntos abertos dos reais, ou ainda o próprio conjunto dos reais, são extensões do conjunto das funções diferenciáveis, tais que:

- Toda função contínua é uma distribuição;
- Toda distribuição possui derivadas que são distribuições;
- As regras formais e usuais do cálculo aplicáveis às funções deriváveis devem funcionar para as distribuições;
- Deve haver um conjunto de teoremas de convergência que seja adequado para dar suporte às análises de limites usuais.

Segundo Takahashi (2002, p. 20), para os propósitos do que será apresentado a seguir, é suficiente definir o conjunto de distribuições \mathbb{D} como o espaço das funções contínuas continuamente diferenciáveis acrescido do espaço de funções gerado pela função delta de Dirac e por suas derivadas e integrais de ordens quaisquer. Logo:

$$\mathbb{D} = \mathcal{C}^\infty + \Delta \quad (5)$$

Onde Δ é o conjunto de todas as funções que podem ser geradas como combinações da função delta de Dirac, de suas derivadas e de suas integrais e \mathcal{C}^∞ designa o conjunto das funções contínuas por partes infinitamente diferenciáveis em um intervalo não degenerado dos complexos, esta teoria pode ser encontrada com maiores detalhes em Ferreira (1993).

Estarão neste conjunto \mathbb{D} os sinais de entrada, assim como suas transformadas de Fourier.

1.1.2 Teoria dos Sinais

Os sinais deverão corresponder, ao menos hipoteticamente, à medição de uma grandeza física num sistema fisicamente realizável. Isso implica uma considerável simplificação, que poderá ser assumida aqui; todos os sinais irão satisfazer às condições de convergência da transformada de Fourier, esta definição é encontrada em Takahashi (2002).

Um sinal $f(t) \in \mathbb{D}$ é definido como uma distribuição que satisfaz às condições de Dirichlet, enunciadas a seguir:

1. $f(t)$ possui um número finito de discontinuidades em todo intervalo T , sendo T um intervalo finito em \mathbb{R} , e cada uma dessas discontinuidades é finita.
2. $f(t)$ possui um número finito de máximos e mínimos em todo intervalo T .
3. $\int_T |f(t)| dt < \infty \quad \forall T$

As condições de Dirichlet podem ser encontradas em Hsu (2012, p. 195, 1972, p. 16), Spiegel (1976, p. 30) e Takahashi (2002, p. 22).

Hsu (2012) e Takahashi (2002) definem as operações utilizadas sobre os sinais conforme a seguir:

Seja $f(t), g(t) \in \mathbb{D}$, tal que, f e g satisfaçam às condições de Dirichlet.

- Soma de sinais

$$f(t) + g(t) = c(t) \quad c \in \mathbb{D} \quad (6)$$

- Multiplicação de um elemento escalar

$$a \cdot f(t) = af(t) \quad a \in \mathbb{C} \quad (7)$$

- Convolução

$$f(t) \odot g(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = c(t) \quad c \in \mathbb{D} \quad (8)$$

- Multiplicação de sinais

$$f(t).g(t) = f(t)g(t) \quad (9)$$

Em aplicações práticas não dispomos de infinitas medições ou de uma função que descreva o sinal em tempo contínuo, mas sim de amostras espaçadas em períodos de tempo de amostragem, por isso é necessário introduzir a análise de sinais em tempo discreto. Essa teoria é encontrada com bastante critério no capítulo 1 [Sinais e sistemas] de Hsu (2012).

1.1.3 Séries de Fourier

“Embora a série de Fourier tenha sido desenvolvida como subsídio matemático ao estudo da transferência de calor, a aplicação desta soma de senos e cossenos estendeu-se a todos os ramos da física, engenharia e matemática” (RICIERI, 1988, p. 11).

Segundo Zill e Cullen (2001), a série de Fourier de uma função f periódica de período $2p$, definida no intervalo $(-p, p)$ é dada por:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi t}{P}\right) + b_n \sin\left(\frac{n\pi t}{P}\right) \right] \quad (10)$$

$$a_0 = \frac{1}{P} \int_{-P}^P f(t)dt \quad (11)$$

$$a_n = \frac{1}{P} \int_{-P}^P f(t) \cos\left(\frac{n\pi t}{P}\right) dt \quad (12)$$

$$b_n = \frac{1}{P} \int_{-P}^P f(t) \sin\left(\frac{n\pi t}{P}\right) dt \quad (13)$$

Um estudo mais detalhado sobre a convergência das séries de Fourier é encontrado em Figueiredo (2014), bem como a dedução dos coeficientes a_n e b_n , e as considerações sobre ortogonalidade, que também são encontrados em Hsu (1972).

Segundo Spiegel (1976, p. 30), as condições de Dirichlet são suficientes, mas não necessárias para a convergência das séries Fourier em $f(t)$.

Em um ponto de descontinuidade, o qual pode ser $t = t_1$, a série de Fourier converge para:

$$\frac{1}{2}[f(t_1 -) + f(t_1 +)] \quad (14)$$

Onde,

$$f(t_1 -) = \lim_{t \rightarrow t_1^-} f(t) \quad e \quad f(t_1 +) = \lim_{t \rightarrow t_1^+} f(t) \quad (15)$$

Esta afirmação é demonstrada em Hsu (1972, p. 17).

Outra forma importante de expressar as séries de Fourier é em termos de exponenciais complexas. Utilizando esta forma, encontramos o espectro de frequências complexas. Segundo Figueiredo (2014), a dedução da série complexa de Fourier pode ser realizada como expomos a seguir.

Usa-se a fórmula de Euler,

$$e^{i\theta} = \cos \theta + i \sin \theta, \quad (16)$$

e suas consequências,

$$\cos \theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \quad e \quad \sin \theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}, \quad (17)$$

para escrever

$$a_n \cos \frac{n\pi t}{P} + b_n \sin \frac{n\pi t}{P} = \left(\frac{a_n}{2} + \frac{b_n}{2i}\right) e^{in\pi t/P} + \left(\frac{a_n}{2} - \frac{b_n}{2i}\right) e^{-in\pi t/P}.$$

Logo, o coeficiente c_n de $e^{\pm in\pi t/P}$ é dado por

$$c_{\pm n} = \frac{a_n}{2} \pm \frac{b_n}{2i} = \frac{1}{2}(a_n \pm ib_n) = \frac{1}{2P} \int_{-P}^P f(t) \left(\cos \frac{n\pi t}{P} \pm i \sin \frac{n\pi t}{P} \right) dt,$$

ou seja,

$$c_{\pm n} = \frac{1}{2P} \int_{-P}^P f(t) e^{\pm in\pi t/P} dt \quad (18)$$

Verifica-se também que:

$$c_0 = \frac{a_0}{2} = \frac{1}{2P} \int_{-P}^P f(t) dt \quad (19)$$

Finalmente, mostra-se que, se $f: \mathbb{R} \rightarrow \mathbb{R}$ for periódica de período $2P$, integrável e absolutamente integrável, então a série de Fourier de f poderá ser escrita na forma:

$$\begin{aligned} f(t) &= c_0 + \sum_{n=1}^{\infty} (c_n e^{\frac{in\pi t}{P}} + c_{-n} e^{\frac{-in\pi t}{P}}) = c_0 + \sum_{n=1}^{\infty} c_n e^{\frac{in\pi t}{P}} + \sum_{n=-\infty}^{-1} c_n e^{\frac{in\pi t}{P}} \\ \therefore f(t) &= \sum_{n=-\infty}^{\infty} c_n e^{\frac{in\pi t}{P}} \end{aligned} \quad (20)$$

Onde,

$$c_{\pm n} = \frac{1}{2P} \int_{-P}^P f(t) e^{\frac{\pm in\pi t}{P}} dt, \quad \forall n = 0, \pm 1, \pm 2, \dots$$

No entanto, se $f(t) \in \mathbb{R}$ então: $c_{-n} = c_n^*$, onde o $*$ indica o complexo conjugado. Logo,

$$c_n = \frac{1}{2P} \int_{-P}^P f(t) e^{\frac{-in\pi t}{P}} dt, \quad \forall n = 0, \pm 1, \pm 2, \dots \quad (21)$$

Além disso, se

$$c_{-n} = c_n^* = |c_n| e^{-i\phi_n} \quad (22)$$

Então

$$|c_n| = \frac{1}{2} \sqrt{a_n^2 + b_n^2} \quad (23)$$

Tal que

$$\phi_n = \tan^{-1} \left(-\frac{b_n}{a_n} \right) \quad (24)$$

Para todo valor de n , exceto para $n=0$. Neste caso, c_0 é real e

$$c_0 = \frac{1}{2} a_0 \quad (25)$$

Relativamente à representação gráfica dos coeficientes c_n e dos ângulos de fase ϕ_n , encontramos a seguinte definição em Hsu (1972, p. 58):

Um gráfico onde se marcam no eixo horizontal as frequências angulares e no eixo vertical os valores dos coeficientes c_n das séries de Fourier chamamos de espectro de amplitudes da função periódica $f(t)$. Um gráfico onde se marcam no eixo vertical os ângulos de fase ϕ_n de c_n e no eixo horizontal as frequências angulares chamamos espectro de fases de $f(t)$. Como os índices n assumem somente valores inteiros, os espectros de amplitudes e de fases não são curvas contínuas, mas somente aparecem na variável discreta $n\omega_0$. A tais valores discretos chamamos, então, espectro de frequências discretas ou linha espectral. A representação dos coeficientes complexos c_n , contra a variável discreta $n\omega_0$ especifica a função periódica $f(t)$ no chamado domínio de frequências, do mesmo modo que a função $f(t)$ contra t a especifica no domínio do tempo.

1.1.4 Transformada de Fourier

A definição da transformada de Fourier pode ser encontrada em livros de matemática, ciências e engenharia. Aqui se optou por fazer uma demonstração heurística, suficiente aos nossos propósitos, com base em Hsu (2012) e Hsu (1972).

Seja $f(t)$ não periódica, ou seja:

$$f(t) \neq f(t + \tau) \quad \forall t \in \mathbb{R} \text{ tal que } f(t) = 0 \quad \forall |t| > T_1$$

Definimos $f_{T_0}(t) = f_{T_0}(t + T_0)$, ou seja $f_{T_0}(t)$ é a extensão periódica de $f(t)$ com período $T_0 = 2T_1$. Tal que $f_{T_0}(t) = f(t)$ se somente se $t \in [-T_1, T_1]$.

Como $f_{T_0}(t)$ é uma função periódica e atende as condições de Dirichlet dentro do intervalo $[-T_1, T_1]$, podemos representá-la em série de Fourier por:

$$f_{T_0}(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t} \quad \text{onde} \quad \omega_0 = \frac{2\pi}{T_0} \quad (26)$$

Com o coeficiente c_n calculado por:

$$c_n = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f_{T_0}(t) e^{-in\omega_0 t} dt \quad (27)$$

Agora se fizermos $T_0 \rightarrow \infty \Rightarrow f_{T_0}(t) = f(t)$, substituindo nos extremos da integral da equação (27), temos:

$$c_n = \frac{1}{T_0} \int_{-\infty}^{\infty} f(t) e^{-in\omega_0 t} dt \quad (28)$$

Neste ponto é extremamente conveniente formularmos a função $F(\omega)$ da seguinte maneira:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (29)$$

Substituindo $F(\omega)$ em (26), temos que:

$$f_{T_0}(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T_0} F(n\omega_0) e^{in\omega_0 t} = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \omega_0 F(n\omega_0) e^{in\omega_0 t} \quad (30)$$

Mas como $T_0 \rightarrow \infty$ temos que $\omega_0 = \Delta\omega_0 \rightarrow 0$, por tanto:

$$f(t) = \lim_{T_0 \rightarrow \infty} f_{T_0}(t) = \lim_{\Delta\omega \rightarrow 0} \left[\frac{1}{2\pi} \sum_{n=-\infty}^{\infty} F(n\Delta\omega) e^{in\Delta\omega t} \Delta\omega \right] \quad (31)$$

Onde $\Delta\omega$ se tornou um diferencial $d\omega$ logo $n\Delta\omega = \omega$, por tanto:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \quad (32)$$

A equação (32) define a transformada inversa de Fourier, pois estamos transformando a função $F(\omega)$, que é uma função do domínio da frequência (ou, no caso, da velocidade angular) em $f(t)$, que é uma função do domínio do tempo. A equação (29) é a definição da transformada de Fourier.

Neste ponto, passaremos a utilizar a seguinte notação para o par de transformadas de Fourier:

Para transformada de Fourier:

$$\mathcal{F}\{f(t)\} = F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (33)$$

Para transformada inversa de Fourier:

$$\mathcal{F}^{-1}\{F(\omega)\} = f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \quad (34)$$

1.1.5 Transformada de Hartley

A transformada de Hartley possui uma estreita relação com a transformada de Fourier. Esta transformada foi proposta por Ralph Vinton Lyon Hartley em seu artigo intitulado “*A More Symmetrical Fourier Analysis Applied to Transmission Problems*”, de 1942. A transformada de Hartley foi proposta como alternativa à transformada de Fourier para algumas aplicações. Sua relação estreita com a transformada de Fourier remonta a sua origem como sendo uma nova formulação da identidade integral de Fourier.

O resultado a seguir pode ser encontrado em Hartley (1942), onde o autor estabelece uma formulação simétrica para a integral de Fourier.

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(\omega) \text{cas}(\omega t) d\omega \quad (35)$$

$$\psi(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \text{cas}(\omega t) dt \quad (36)$$

Onde *cas* é a função *cosine and sine* (cosseno e seno).

$$cas \theta = \cos \theta + \sen \theta \quad (37)$$

Com um pouco de manipulação trigonométrica é fácil mostrar que:

$$cas \theta = \sqrt{2} \cos \left(\theta - \frac{\pi}{4} \right) = \sqrt{2} \sen \left(\theta + \frac{\pi}{4} \right) \quad (38)$$

Neste ponto, passaremos a utilizar a seguinte notação para o par de transformadas de Hartley, com o intuito de obter uma maior semelhança com a notação das transformadas de Fourier:

$$\mathcal{H}\{f(t)\} = H(\omega) = \int_{-\infty}^{\infty} f(t) cas(\omega t) dt \quad (39)$$

E sua transformada inversa:

$$\mathcal{H}^{-1}\{H(\omega)\} = f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) cas(\omega t) d\omega \quad (40)$$

Como a transformada de Hartley é uma variante da identidade integral de Fourier, sua condição de existência está ligada à convergência da integral de Fourier, ou seja, a própria transformada de Fourier deve existir.

Se separarmos a transformada de Hartley em suas componentes pares e ímpares teremos as seguintes relações, encontradas em Poularikas (1999) e em Olejniczak (2000). Sendo $H_e(\omega)$ a componente par de $H(\omega)$ e $H_o(\omega)$ a componente ímpar de $H(\omega)$.

$$\mathcal{H}_e\{f(t)\} = H_e(\omega) = \frac{H(\omega) + H(-\omega)}{2} = \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt \quad (41)$$

$$\mathcal{H}_o\{f(t)\} = H_o(\omega) = \frac{H(\omega) - H(-\omega)}{2} = \int_{-\infty}^{\infty} f(t) \sen(\omega t) dt \quad (42)$$

A relação entre as transformadas de Hartley e Fourier é dada por:

$$\mathcal{F}\{f(t)\} = F(\omega) = \frac{H(\omega) + H(-\omega)}{2} - i \frac{H(\omega) - H(-\omega)}{2} \quad (43)$$

No capítulo 14 [*The Hartley Transform*] de Poularikas (1999) e no capítulo 4 [*The Hartley Transform*] de Olejniczak (2000) também encontramos a relação inversa.

$$\mathcal{H}\{f(t)\} = \Re\{F(\omega)\} - \Im\{F(\omega)\} \quad (44)$$

No lado direito da equação (44), $\Re\{F(\omega)\}$ e $\Im\{F(\omega)\}$ são respectivamente a componente real e imaginária da transformada de Fourier.

1.1.6 Algumas transformadas e teorema da derivada

Neste ponto, é importante apresentar algumas transformadas que servirão de referência para as comparações teóricas com os resultados experimentais.

A primeira transformada de interesse é a transformada da função delta de Dirac, conforme segue:

$$\mathcal{F}\{\delta(t - a)\} = \int_{-\infty}^{\infty} \delta(t - a) e^{-i\omega t} dt = e^{-ia\omega} \quad (45)$$

O resultado de (45) é uma consequência direta do resultado (3) e uma translação em (1).

Semelhantemente para transformada de Hartley, tem-se:

$$\mathcal{H}\{\delta(t - a)\} = \int_{-\infty}^{\infty} \delta(t - a) \cos(\omega t) dt = \cos(a\omega) - \sin(a\omega) \quad (46)$$

A segunda transformada interessante de se demonstrar é da exponencial complexa, ou seja, o próprio núcleo da transformada inversa de Fourier.

$$\mathcal{F}\{e^{i\omega_0 t}\} = \int_{-\infty}^{\infty} e^{i\omega_0 t} e^{-i\omega t} dt = \int_{-\infty}^{\infty} e^{it(\omega_0 - \omega)} dt \quad (47)$$

Substituindo $(\omega_0 - \omega)$ por t' e t por ω' tem-se:

$$\begin{aligned} \int_{-\infty}^{\infty} e^{it'\omega'} d\omega' &= \mathcal{F}^{-1}\{2\pi\} = 2\pi\delta(t') = 2\pi\delta(\omega_0 - \omega) = 2\pi\delta[-(\omega - \omega_0)] = 2\pi\delta(\omega - \omega_0) \\ \therefore \mathcal{F}\{e^{i\omega_0 t}\} &= \int_{-\infty}^{\infty} e^{it(\omega_0 - \omega)} dt = 2\pi\delta(\omega - \omega_0) \end{aligned} \quad (48)$$

Para transformada de Hartley temos que:

$$\begin{aligned}\mathcal{H}\{e^{i\omega_0 t}\} &= \int_{-\infty}^{\infty} e^{i\omega_0 t} \text{cas}(\omega t) dt = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} e^{it(\omega+\omega_0)} + e^{-it(\omega-\omega_0)} - ie^{it(\omega+\omega_0)} + ie^{-it(\omega-\omega_0)} dt\end{aligned}\quad (49)$$

Utilizando o resultado (48) em (49) e lembrando que o resultado deve ser real, conclui-se que:

$$\begin{aligned}\mathcal{H}\{e^{i\omega_0 t}\} &= \frac{1}{2} [2\pi\delta(\omega + \omega_0) + 2\pi\delta(\omega - \omega_0) - 2\pi\delta(\omega + \omega_0) + 2\pi\delta(\omega - \omega_0)] \\ &= 2\pi\delta(\omega - \omega_0)\end{aligned}\quad (50)$$

A terceira transformada importante é a da função seno. Para Fourier temos:

$$\begin{aligned}\mathcal{F}\{\text{sen}(\omega_0 t)\} &= \int_{-\infty}^{\infty} \text{sen}(\omega_0 t) e^{-i\omega t} dt = \int_{-\infty}^{\infty} \frac{e^{i\omega_0 t} - e^{-i\omega_0 t}}{2i} e^{-i\omega t} dt \\ &= \frac{1}{2i} \int_{-\infty}^{\infty} (e^{-it(\omega-\omega_0)} - e^{-it(\omega+\omega_0)})\end{aligned}\quad (51)$$

Utilizando o resultado (45) na expressão final de (51), conclui-se que:

$$\mathcal{F}\{\text{sen } \omega_0 t\} = -i\pi[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)] \quad (52)$$

Para a transformada de Hartley temos:

$$\begin{aligned}\mathcal{H}\{\text{sen } \omega_0 t\} &= \int_{-\infty}^{\infty} \text{sen}(\omega_0 t) \text{cas}(\omega t) dt = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} i(e^{-i\omega_0 t} - e^{i\omega_0 t})(e^{i\omega t} + e^{-i\omega t} - ie^{i\omega t} + ie^{-i\omega t}) dt = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} (e^{it(\omega-\omega_0)} - e^{it(\omega+\omega_0)}) dt = \pi[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)] \\ \therefore \mathcal{H}\{\text{sen } \omega_0 t\} &= \pi[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]\end{aligned}\quad (53)$$

Outra transformada relevante para a compreensão de alguns tópicos que são apresentados neste trabalho é a transformada de Fourier de um sinal retangular, ou função retangular. A função retangular é definida por:

$$g(t) = A[\mathcal{U}(t+d) - \mathcal{U}(t-d)] = \begin{cases} 0, & |t| < d \\ A, & |t| = d \\ 0, & |t| > d \end{cases} \quad \forall A, d \in \mathbb{R}$$

Logo, temos a seguinte transformada:

$$\mathcal{F}\{g(t)\} = \int_{-\infty}^{\infty} g(t) e^{-i\omega t} dt = A \int_{-d}^d e^{-i\omega t} dt = A \frac{e^{i\omega d} - e^{-i\omega d}}{i\omega}$$

$$\therefore \mathcal{F}\{g(t)\} = \frac{2A}{\omega} \text{sen}(\omega d) = 2Ad \cdot \text{sinc}(\omega d) \quad (54)$$

A figura 2 apresenta o gráfico da função retangular e sua respectiva transformada de Fourier.

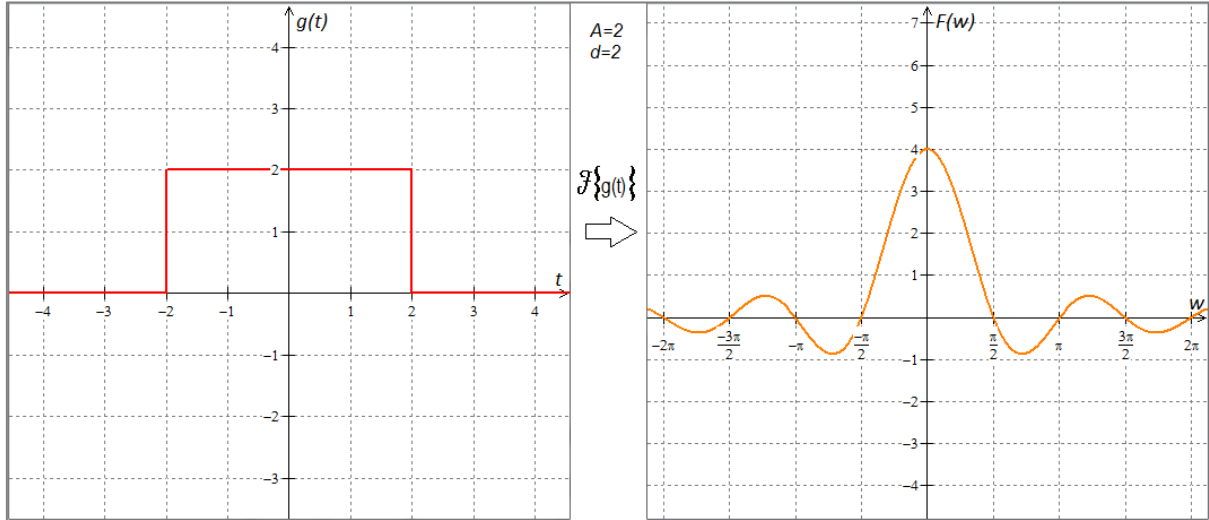


Figura 2 – Gráfico função retangular no domínio do tempo e sua transformada de Fourier .

Fonte: Autor.

O teorema da transformada da derivada é definido da seguinte forma:

Seja $f \in \mathbb{D}$ conforme definido na equação (5), $F(\omega) = \mathcal{F}\{f(t)\}$, então:

$$\mathcal{F}\left\{\frac{d^n f}{dt^n}\right\} = (i\omega)^n F(\omega) \quad (55)$$

A demonstração do teorema acima pode ser realizada utilizando a definição da transformada de Fourier, equação (33). Aplicando uma solução por partes na integral da transformada, conforme a seguir:

$$\mathcal{F}\left\{\frac{d^n f}{dt^n}\right\} = \int_{-\infty}^{\infty} \frac{d^n f}{dt^n} e^{-i\omega t} dt = \lim_{\varepsilon \rightarrow \infty} \left[\frac{d^{n-1} f}{dt^{n-1}} e^{-i\omega t} \right]_{-\varepsilon}^{\varepsilon} + i\omega \int_{-\infty}^{\infty} \frac{d^{n-1} f}{dt^{n-1}} e^{-i\omega t} dt$$

Onde o $\lim_{t \rightarrow \pm\infty} \left[\frac{d^{n-1} f}{dt^{n-1}} e^{-i\omega t} \right] = 0$, este limite existe e converge, pois $f(t)$ e suas derivadas atendem as condições de Dirichlet. Aplicando a solução por partes n vezes verifica-se o resultado (55).

1.2 TRANSFORMADAS DISCRETAS

Em aplicações computacionais o tempo é discreto, por isso devem-se definir os sinais e as transformadas em termos discretos. Define-se neste tópico a transformada discreta de Fourier (DFT), bem como um algoritmo para computar esta transformada discreta.

A partir deste ponto, utilizaremos as seguintes notações para sinais discretos:

$$x[n] = x_n = f(nT) \quad \forall n \in \mathbb{Z}$$

Ou seja, $x[n]$ é a versão discreta da função contínua $f(t)$ e T é o intervalo entre as amostras.

1.2.1 Transformada Discreta de Fourier

De forma generalizada, dado $x[n] \in \mathbb{R}, n \in \mathbb{Z}$, a transformada de Fourier de tempo discreto de $x[n]$ pode ser definida partindo de (33) como:

$$X[\omega] = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-i\omega n} \quad (56)$$

Segundo Takahashi (2002), é possível construir um algoritmo capaz de computar a transformada de Fourier discreta perfazendo um número finito de operações, desde que os sinais em questão atendam alguns requisitos, listados a seguir:

- 1- $f(t) = f(t + \tau) \quad \forall t$
- 2- Função amostragem $f_s(t) = f(t) \cdot d(t, T)$
Onde $d(t, T) = \sum_{k=-\infty}^{\infty} \delta(t - kT)$
 $T = \frac{\tau}{N}$, onde T é o período de amostragem e N a quantidade de amostras.
- 3- Frequência de Nyquist: $\omega_s > 2\omega_{max}$, ou seja, a frequência de amostragem deve ser maior que o dobro da maior frequência contida em $f(t)$.
- 4- Devemos dispor de N amostras do sinal.

Embora esse tópico seja encontrado em diversas outras literaturas, o capítulo 6 [Análise de Fourier de Sinais e Sistemas de Tempo Discreto] de Hsu (2012) apresenta os fundamentos teóricos necessários à compreensão da transformada discreta de Fourier (DFT). Segundo este autor, a DFT é definida da seguinte maneira.

Seja $x[n]$ uma sequência de tamanho finito de comprimento N .

Definimos a DFT de $x[n]$ como sendo:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad k = 0, 1, \dots, N-1 \quad (57)$$

Onde, $W_N = e^{-i\frac{2\pi}{N}}$

A transformada inversa de Fourier em sua forma discreta é dada por:

$$x[n] = \sum_{k=0}^{N-1} X[k] W_N^{-kn} \quad n = 0, 1, \dots, N-1 \quad (58)$$

1.2.2 Transformada Discreta de Hartley

A transformada discreta de Hartley (DHT) é definida em Bracewell (1984), de maneira análoga a DFT.

Segundo este autor, dada uma sequência real $f[n]$ para $n = 0, 1, \dots, N-1$, podemos definir a DHT $H[k]$ como a soma das transformadas senos e cossenos.

$$H[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] \text{cas} \left(\frac{2\pi nk}{N} \right) \quad k = 0, 1, \dots, N-1 \quad (59)$$

Sua transformada inversa é:

$$f[n] = \sum_{k=0}^{N-1} H[k] \text{cas} \left(\frac{2\pi nk}{N} \right) \quad n = 0, 1, \dots, N-1 \quad (60)$$

1.2.3 Janelamento

Quando um sinal é amostrado, inicia-se esta amostragem em um dado instante, finalizando após um intervalo de tempo. Esta operação retira apenas uma “janela” retangular do sinal no domínio do tempo. A operação de janelamento pode ser vista como uma multiplicação de sinais no domínio do tempo, conforme definida no tópico 1.1.2, ou como uma convolução no domínio da frequência, consequência do teorema da convolução (ver Apêndice A). Conforme apresentado na expressão (61), utilizando as definições de operações (8) e (9), onde $f(t)$ é o sinal e $w(t)$ é a janela, que também pertence à classe dos sinais.

O teorema de convolução pode ser expresso pela equivalência:

$$f(t) \cdot w(t) \leftrightarrow F(\omega) \circledast W(\omega) \quad (61)$$

O tamanho da janela varia com o tamanho do vetor amostra e a frequência de amostragem. Neste caso, surge uma janela simples, esta janela é conhecida como janela retangular, e pode ser definida da seguinte forma:

$$w(t, T) = \begin{cases} 0, & \forall |t| > T/2 \\ 1, & \forall |t| < T/2 \end{cases} \quad (62)$$

Onde T é uma constante que define a largura da janela, ou ainda, o período da amostragem. Aplicando a transformada de Fourier em $w(t, T)$. Obtém-se:

$$\mathcal{F}\{w(t)\} = W(\omega) = \int_{-\infty}^{\infty} w(t) e^{-i\omega t} dt = \int_{-T/2}^{T/2} e^{-i\omega t} dt = 2T \operatorname{sinc}\left(\frac{\omega T}{2}\right) \quad (63)$$

Se $f(t)$ é um sinal periódico de período τ , onde $\omega_0 = \frac{2\pi}{\tau}$, temos a transformada de Fourier dada por:

$$\mathcal{F}\{f(t)\} = F(\omega) = \sum_{n=0}^{\infty} A_n (\delta(\omega - n\omega_0) + \delta(\omega + n\omega_0)) \quad (64)$$

Combinando as equações (3) e (8) temos a seguinte propriedade:

$$a(t) \otimes \delta(t - \phi) = a(t - \phi) \quad (65)$$

Então aplicando (65) ao teorema de convolução de (62) e utilizando o resultado (64), chega-se ao seguinte resultado:

$$W(\omega) \otimes F(\omega) = \sum_{n=0}^{\infty} A_n (W(\omega - n\omega_0) + W(\omega + n\omega_0)) \quad (66)$$

Ou seja, a janela quadrada definida em (62) cuja transformada de Fourier é uma função *sinc*, produzirá “cópias” com características da *sinc* sobre o espectro de $f(t)$. Para solucionar este espalhamento indesejável no espectro podem ser utilizadas janelas diferentes da quadrada. Em geral essas janelas tendem a atenuar as extremidades da janela, evitando o degrau da janela quadrada. Dentre estas janelas selecionou-se empiricamente a janela de Hamming para ser utilizada neste trabalho. A definição da janela de Hamming é dada pela equação (67). O efeito desta janela sobre o sinal pode ser visto nos dois gráficos da figura 3, na qual, nota-se a diferença no espectro gerado pela janela retangular em relação à janela de Hamming. Os dois gráficos superiores são de um sinal amostrado por uma janela quadrada, (os gráficos correspondem respectivamente ao sinal no tempo e seu espectro à direita), enquanto na linha de baixo são mostrados, para, o mesmo sinal, o efeito da janela de Hamming no domínio do tempo e no espectro de frequências.

$$w(t) = 0,54 - 0,46 \cos\left(t \frac{2\pi}{N}\right) \quad (67)$$

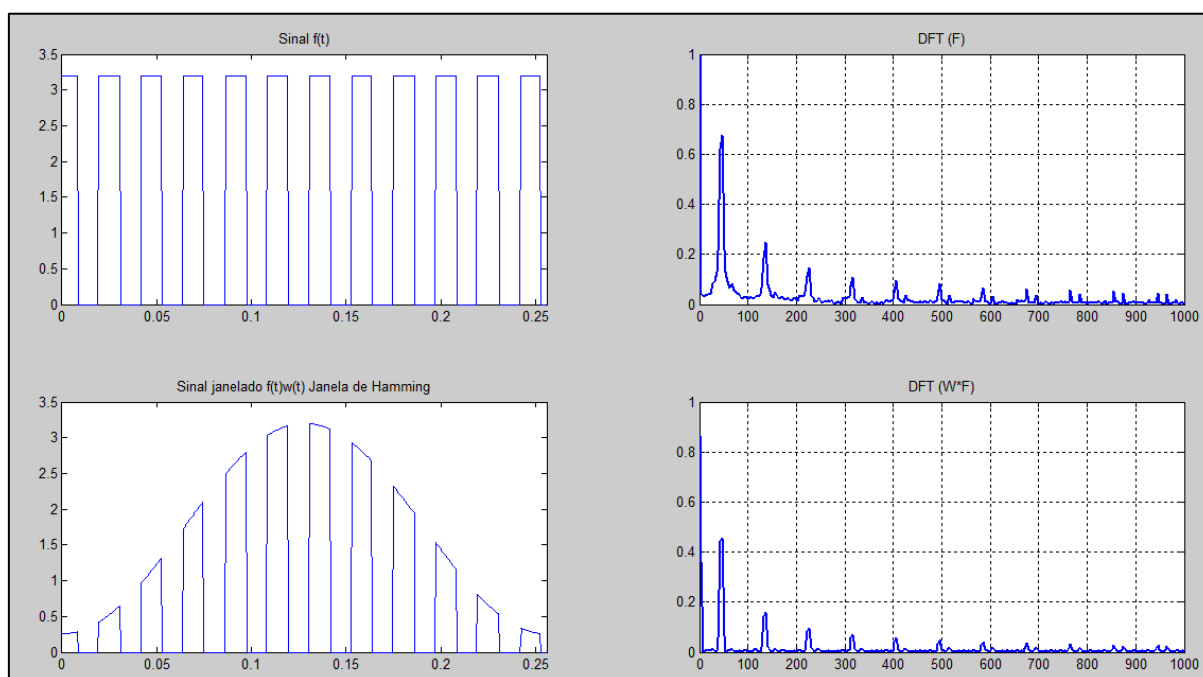


Figura 3 – Efeito do janelamento nos domínios do tempo e da frequência, utilizando janela de Hamming.
Fonte: Autor.

1.2.4 Complexidade de algoritmos

Neste tópico, é feita uma breve revisão de alguns conceitos básicos referentes à análise de complexidade de algoritmos, pois este conteúdo é evidentemente importante para a compreensão do desempenho comparativo entre algoritmos.

Segundo Toscani e Veloso (2012), a complexidade de um algoritmo reflete o tamanho do esforço computacional necessário para executá-lo. Este esforço computacional reflete a quantidade de trabalho, em termos de tempo e ou memória requeridos para a execução do algoritmo. Entre as diversas medidas de desempenho, uma das mais valiosas é o tempo de execução do algoritmo, sendo, por isso, a medida mais frequente na literatura em geral.

Toscani e Veloso (2012, p. 24) concebem a complexidade dos algoritmos a partir da noção de complexidade assintótica. Segundo estes autores,

O comportamento assintótico de um algoritmo é o mais procurado, já que, para um volume grande de dados, a complexidade torna-se mais importante. O algoritmo assintoticamente mais eficiente é o melhor para todas as entradas, exceto talvez para entradas relativamente pequenas.

Além disso, a complexidade exata costuma conter expressões de difícil manipulação e análise, devido ao excesso de informação que se torna desprezível conforme

crece o tamanho da entrada; por isso, a análise de complexidade concentra-se no comportamento assintótico.

Para melhor ilustrar o conceito, suponhamos um algoritmo cuja complexidade exata seja de $10n+7$. Para uma entrada suficientemente grande, a parcela linear $10n$ se torna dominante em relação à constante 7, por isso simplificamos para $10n$. Percebe-se que $10n$ cresce mais rapidamente que n , mas sua razão é constante. Isto é similar a uma mudança de escala. Deste modo, simplificamos $10n$ para n , a menos de uma constante multiplicativa, $10n$ e n apresentam o mesmo comportamento assintótico.

Ainda segundo Toscani e Veloso (2012), várias comparações de complexidade assintótica podem ser definidas, dentre estas, será destacada neste trabalho a notação ‘O’. Porém, antes de introduzir a notação ‘O’, necessita-se de alguns conceitos de análise de complexidade assintótica de algoritmos.

Uma cota assintótica superior (CAS) é uma função que cresce mais rapidamente que outra a partir de certo ponto.

Define-se que g é CAS para f se:

$$(\exists n_0 \in \mathbb{N})(\forall n \geq n_0): f(n) \leq g(n)$$

O conceito é ilustrado no gráfico da figura 4.

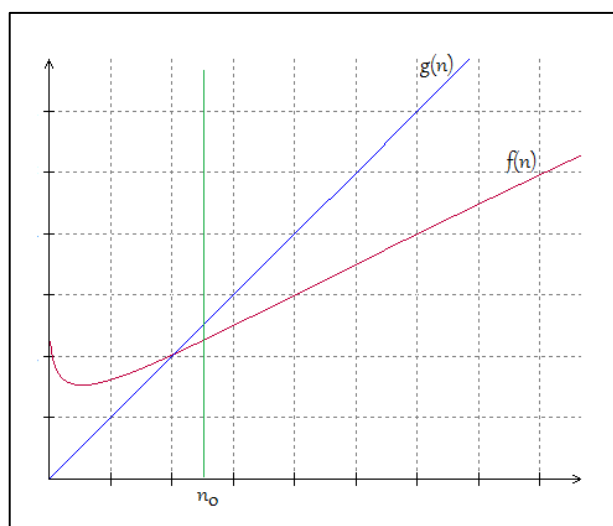


Figura 4 – $g(n)$ é CAS para $f(n)$.

Fonte: Autor.

A notação ‘O’ é comumente utilizada para estabelecer a ordem assintótica em diversas literaturas, por isso nos ateremos apenas a esta notação neste trabalho.

A ordem O designa uma cota assintótica superior a menos de constantes. Definindo que $f(n)$ é de ordem $O(g(n))$, se para alguma constante $c \in \mathbb{R}_+$, então $c[g(n)]$ é CAS para $f(n)$, ou seja: $(\exists c \in \mathbb{R}_+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0): f(n) \leq c[g(n)]$.

A complexidade da DFT definida em (57), é dada por:

$$C(N) = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} 1 = \sum_{k=0}^{N-1} N := O(N^2)$$

Segundo Toscani e Veloso (2012), para uma complexidade dada por um polinômio $p(n) = a_m n^m + a_{m-1} n^{m-1} + \dots + a_1 n + a_0$ de grau m , onde os coeficientes a_m são constantes, sua complexidade absoluta é dada por $O(n^m)$.

Demonstração:

$$\begin{aligned} |p(n)| &\leq |a_m|n^m + |a_{m-1}|n^{m-1} + \dots + |a_1|n + |a_0| \\ |a_m|n^m + |a_{m-1}|n^{m-1} + \dots + |a_1|n + |a_0| &= n^m \left[|a_m| + \frac{|a_{m-1}|}{n} + \dots + \frac{|a_1|}{n^{m-1}} + \frac{|a_0|}{n^m} \right] \\ &\text{para } n \neq 0 \\ \Rightarrow n^m \left[|a_m| + \frac{|a_{m-1}|}{n} + \dots + \frac{|a_1|}{n^{m-1}} + \frac{|a_0|}{n^m} \right] &\leq n^m [|a_m| + |a_{m-1}| + \dots + |a_1| + |a_0|] \\ \text{Então, escolhendo } c = |a_m| + |a_{m-1}| + \dots + |a_1| + |a_0| \text{ e } n_0 = 1, \text{ tem-se:} \\ |p(n)| &\leq c \cdot n^m \quad \forall n \geq n_0 \\ \therefore p(n) &:= O(n^m) \end{aligned} \tag{68}$$

1.2.5 Algoritmos Rápidos para as Transformadas Discretas

A transformada discreta de Fourier de um vetor v pode ser escrita como:

$$F[k] = \sum_{n=0}^{N-1} \omega^{nk} v[n] \quad | \quad \omega = e^{-i\frac{2\pi}{N}}, \quad k = 0, 1, \dots, N-1 \tag{69}$$

Segundo Blahut (2010, p. 68), a DFT conforme está escrita em (69), requer N^2 multiplicações e N^2 adições para ser computada (ordem $O(N^2)$). Se N é um número composto, existem formas de mudar esta transformada unidimensional em uma transformada que poderíamos chamar bidimensional. Esta mudança torna a computação muito mais eficiente. Algoritmos deste tipo são conhecidos como Transformadas Rápidas de Fourier ou FFT (*Fast Fourier Transform*).

A ideia proposta por Cooley-Tukey é utilizar-se de fatorações algébricas que subdividam a transformada discreta, reduzindo sua complexidade computacional.

Partindo da DFT conforme a expressão (69), considerando que N é composto, logo, $N = N'N''$. Então se tem:

$$n = n' + N'n'' \quad (70)$$

$$\begin{aligned} n' &= 0, 1, \dots, N' - 1 \\ n'' &= 0, 1, \dots, N'' - 1 \end{aligned} \quad (71)$$

Pois,

$$\begin{aligned} n = 0, \dots, N - 1 &\Rightarrow \max\{n\} = N - 1 = N'N'' - 1 = \\ &= N'N'' - 1 + N' - N' = N' - 1 + N'(N'' - 1) \end{aligned} \quad (72)$$

Analogamente, tem-se que:

$$\begin{aligned} k = 0, \dots, N - 1 &\Rightarrow \max\{k\} = N - 1 = N'N'' - 1 = \\ &= N'N'' - 1 + N'' - N'' = N'' - 1 + N''(N' - 1) \\ \therefore k &= k'' + N''k' \mid \begin{array}{l} k' = 0, \dots, N' - 1 \\ k'' = 0, \dots, N'' - 1 \end{array} \end{aligned} \quad (73)$$

Substituindo os resultados de (73) e (70) em (69), tornando-a uma transformada dupla, com a seguinte forma:

$$F_{(k''+N''k')} = \sum_{n'=0}^{N'-1} \sum_{n''=0}^{N''-1} \omega^{(n'+n''N')(k''+N''k')} v_{(n'+N'n'')} \quad (74)$$

Aplicando a distributiva no expoente de ômega da expressão (74):

$$(n' + n''N')(k'' + N''k') = n'k'' + n'N''k' + n''N'k'' + n''N'N''k' \quad (75)$$

No lado esquerdo da equação (75), temos um termo cuja exponencial tem em sua composição o $N'N''$, este termo vai à zero, pois $N'N'' > N - 1$, devido a periodicidade de e^{-in} , logo, $N'N'' = 0 \Rightarrow n''N'N''k' = 0$. Utilizando este resultado e fazendo $\gamma = \omega^{N'}$ e $\beta = \omega^{N''}$, chega-se ao seguinte resultado:

$$F_{(k''+N''k')} = \sum_{n'=0}^{N'-1} \left[\beta^{n'k'} \omega^{n'k''} \sum_{n''=0}^{N''-1} \gamma^{n''k''} v_{(n'+N'n'')} \right] \quad (76)$$

Fazendo uma mudança de variável de F_N para $F_{N'N''}$, finalmente encontra-se a expressão que computa o algoritmo de Cooley-Tukey:

$$F_{k'k''} = \sum_{n'=0}^{N'-1} \left[\beta^{n'k'} \omega^{n'k''} \sum_{n''=0}^{N''-1} \gamma^{n''k''} v_{n'n''} \right] \quad (77)$$

Ainda segundo Blahut (2010, p. 72), muitas aplicações do Cooley-Tukey FFT usa bloco de tamanho N que é uma potência de dois ou de quatro. O bloco de tamanho 2^m é fatorado como $2 \cdot 2^{m-1}$, a cada iteração. A FFT então é chamada de *radix-2* Cooley-Tukey FFT. Similarmente, o bloco de tamanho 4^m é fatorado em $4 \cdot 4^{m-1}$. A FFT então é conhecida como *radix-4* Cooley-Tukey FFT. Em Rao, Kim e Hwang (2010) são definidos algoritmos para diferentes composições de N , inclusive para caso N seja um número primo.

A definição da FFT *radix-2* Cooley-Tukey de 2^m pontos, ou seja, utilizando n' igual a 2 e n'' igual a 2^{m-1} , é realizada substituindo $n' = 2$ e $n'' = n/2$ na expressão (77), podendo ser simplificada para a seguinte forma.

$$F_k = \sum_{n=0}^{\left(\frac{N}{2}\right)-1} \omega^{2nk} v_{2n} + \omega^k \sum_{n=0}^{\left(\frac{N}{2}\right)-1} \omega^{2nk} v_{2n+1} \quad (78)$$

$$F_{k+\frac{N}{2}} = \sum_{n=0}^{\left(\frac{N}{2}\right)-1} \omega^{2nk} v_{2n} - \omega^k \sum_{n=0}^{\left(\frac{N}{2}\right)-1} \omega^{2nk} v_{2n+1} \quad (79)$$

Em (78) e (79) $k = 0, \dots, \frac{N}{2} - 1$, onde foi utilizado o fato que $\beta = \omega^{n/2} = -1$, devido a simetria da matriz de coeficientes. Em termos computacionais, isto equivale a dividir em duas transformadas menores, dadas por:

$$F'_k = \sum_{n=0}^{\frac{N}{2}-1} \mu^{nk} v_{2n} \quad e \quad F''_k = \sum_{n=0}^{\frac{N}{2}-1} \mu^{nk} v_{2n+1} \quad (80)$$

Onde $\mu = \omega^2$. Cada uma dessas expressões corresponde a uma transformada de Fourier de comprimento $N/2$. Onde F é obtido pelas seguintes equações.

$$F_k = F'_k + \omega^k F''_k \quad e \quad F_{k+\frac{N}{2}} = F'_k - \omega^k F''_k \quad (81)$$

Para $k = 0, \dots, \frac{N}{2} - 1$.

Neste ponto podemos bisseccionar as transformadas de $N/2$ pontos em transformadas menores de $N/4$ pontos, e continuar com esse processo até que as transformadas se reduzam a apenas um ponto.

A operação básica para computar o Cooley-Tukey conforme mostrado nas equações (78) e (79), é conhecida como “borboleta”, esta operação é ilustrado na figura 5.

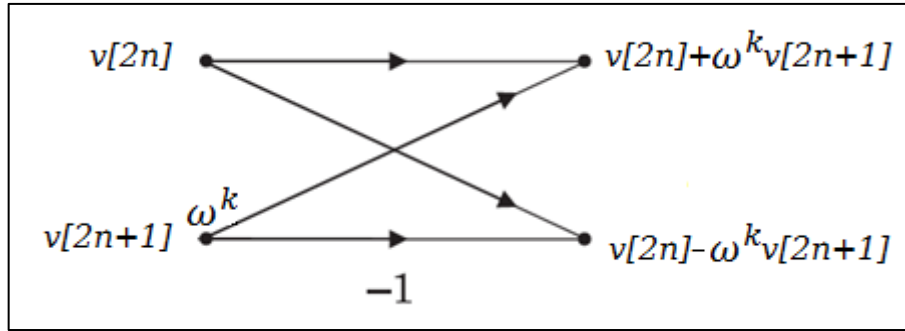


Figura 5 – Borboleta para computar Cooley-Tukey base 2 FFT *decimation-in-time*.

Fonte: Autor.

Por exemplo, o caso em que $N = 8 = 2^3$, segue que:

$$F[k] = \sum_{n=0}^7 v[n] \omega^{nk} \quad k = 0, \dots, 7$$

Onde a complexidade é da ordem de $O(N^2)$, ou seja, 64 operações.

Bisseccionando a expressão:

$$F[k] = \sum_{n=0}^3 v[2n] \omega^{nk} + \omega^k \sum_{n=0}^3 v[2n+1] \omega^{nk} \quad k = 0, \dots, 3$$

$$F\left[k + \frac{N}{2}\right] = \sum_{n=0}^3 v[2n] \omega^{nk} - \omega^k \sum_{n=0}^3 v[2n+1] \omega^{nk} \quad k = 0, \dots, 3$$

Neste ponto temos duas transformadas menores que compõe a “borboleta”.

$$A[k] = \sum_{n=0}^3 v[2n] \omega^{nk} \quad k = 0, \dots, 3$$

$$B[k] = \sum_{n=0}^3 v[2n+1] \omega^{nk} \quad k = 0, \dots, 3$$

Bisseccionando em transformadas menores:

$$A[k] = \sum_{n=0}^1 v[4n] \omega^{nk} + \omega^k \sum_{n=0}^1 v[4n+2] \omega^{nk} \quad k = 0, 1$$

$$A\left[k + \frac{N}{2}\right] = \sum_{n=0}^1 v[4n] \omega^{nk} - \omega^k \sum_{n=0}^1 v[4n+2] \omega^{nk} \quad k = 0, 1$$

$$B[k] = \sum_{n=0}^1 v[4n+1]\omega^{nk} + \omega^k \sum_{n=0}^1 v[4n+3]\omega^{nk} \quad k = 0,1$$

$$B\left[k + \frac{N}{2}\right] = \sum_{n=0}^1 v[4n+1]\omega^{nk} - \omega^k \sum_{n=0}^1 v[4n+3]\omega^{nk} \quad k = 0,1$$

Bisseccionando-se mais uma vez ficaram apenas os pontos individuais, cujas transformadas são eles próprios.

No diagrama apresentado na figura 6, é possível observar o comportamento do índice n a cada etapa do exemplo acima. Na última camada inferior, aparece o índice de cada ponto, com uma permutação conhecida como reversão de bits, esta permutação corresponde ao espelhamento dos bits do índice, por exemplo, $4_{base\ 10} = 100_{base\ 2}$, aplicando a reversão ou espelhamento dos bits, temos $001_{base\ 2} = 1_{base\ 10}$. Esta permutação também pode ser observada nos índices do vetor de entrada na figura 7.

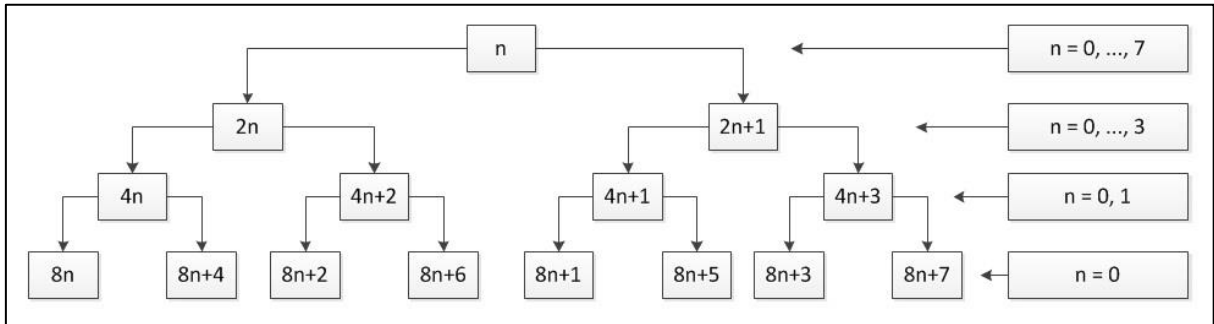


Figura 6 – Diagrama de desenvolvimento dos índices do vetor durante bissecções.

Fonte: Autor.

Na figura 7 é apresentado o fluxograma com as operações borboletas para $N=8$. Este algoritmo é conhecido como DIT (*Decimation in time*) FFT, pois a resolução da frequência durante a decomposição é sempre a mesma. Outra resolução possível seria o DIF (*Decimation in frequency*), onde se conserva a resolução no tempo.

A quantidade de bissecções é dada pelo $\log_2 N$; em cada camada temos ordem assintótica de N operações, por isso a ordem de complexidade para a FFT é dada por $O(N \log_2 N)$, no caso em que $N=8$, teremos 24 operações. Conforme foi apresentado no início do exemplo, para DFT de ordem $O(N^2)$ são necessárias 64 operações. Ou seja, o número de operações foi reduzido em 62,5% utilizando a FFT no lugar da DFT por definição.

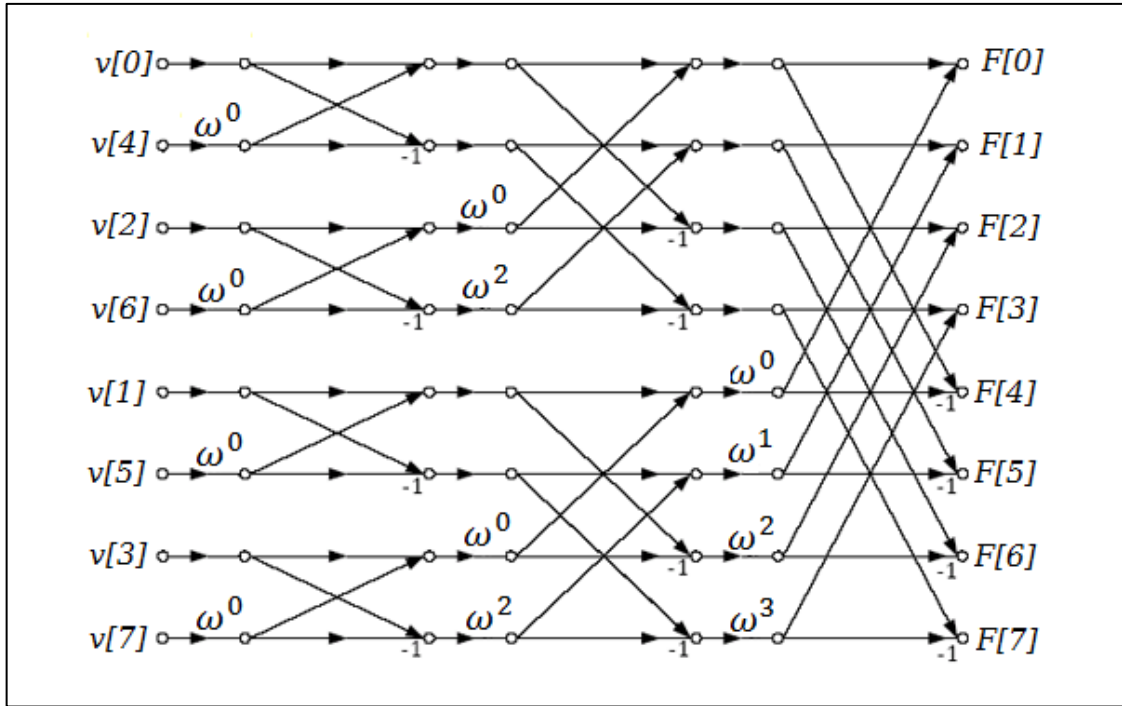


Figura 7 – Fluxograma para FFT, N=8 (*radix-2 DIT-FFT*).

Fonte: Autor.

A transformada rápida de Hartley foi apresentada por Bracewell (1984), utilizando o mesmo princípio do Cooley-Tukey para FFT, porém, com uma diferença significativa na fatoração que leva à redução das transformadas.

Partindo da DHT definida por:

$$H[k] = \sum_{n=0}^{N-1} v[n] \text{cas}(nk\theta) \quad k = 0, 1, \dots, N-1 \quad (82)$$

Onde, $\theta = \frac{2\pi}{N}$.

Para esta fatoração é utiliza-se uma propriedade da transformada de Hartley. A equação (83), apresenta uma propriedade da transformada de Hartley que pode ser encontrada em Poularikas (1999) e Olejniczak (2000).

$$H(\omega + k) = H(\omega) \cos(k) + H(-\omega) \text{sen}(k) \quad (83)$$

É conveniente neste ponto, inserir a relação para o caso discreto:

$$H[-k] = \begin{cases} H[0] & \rightarrow k = 0 \\ H[N-k] & \rightarrow k > 0 \end{cases} \quad (84)$$

Separando (82) em parte par e parte ímpar.

$$H[k] = \sum_{n=0}^{\frac{N}{2}-1} v[2n] \text{cas}(2nk\theta) + \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \text{cas}(2nk\theta + k\theta) \quad k = 0, \dots, \frac{N}{2}-1 \quad (85)$$

Aplicando a propriedade (83) na subtransformada ímpar de (85), resulta em:

$$H[k] = \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos(2nk\theta) + \cos(k\theta) \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos(2nk\theta) \\ + \sin(k\theta) \sum_{n=0}^{\frac{N}{2}-1} v[-(2n+1)] \cos(2nk\theta) \quad k = 0, \dots, \frac{N}{2} - 1 \quad (86)$$

$$H\left[k + \frac{N}{2}\right] = \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos(2nk\theta) - \cos(k\theta) \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos(2nk\theta) \\ - \sin(k\theta) \sum_{n=0}^{\frac{N}{2}-1} v[-(2n+1)] \cos(2nk\theta) \quad k = 0, \dots, \frac{N}{2} - 1 \quad (87)$$

Os sinais negativos na expressão (87) se devem as equações (88) e (89), onde foram utilizadas as relações trigonométricas para soma dos ângulos.

$$\cos\left(k\theta + \frac{N}{2}\theta\right) = \cos(k\theta + \pi) = \cos(k\theta) \cos(\pi) - \sin(k\theta) \sin(\pi) = -\cos(k\theta) \quad (88)$$

$$\sin\left(k\theta + \frac{N}{2}\theta\right) = \sin(k\theta + \pi) = \sin(k\theta) \cos(\pi) + \cos(k\theta) \sin(\pi) = -\sin(k\theta) \quad (89)$$

Sendo que as mesmas propriedades de periodicidade encontradas na FFT podem ser consideradas para a FHT.

Partindo das expressões (86) e (87) é possível construir a operação borboleta para a FHT, conforme ilustrada o fluxograma na figura 8.

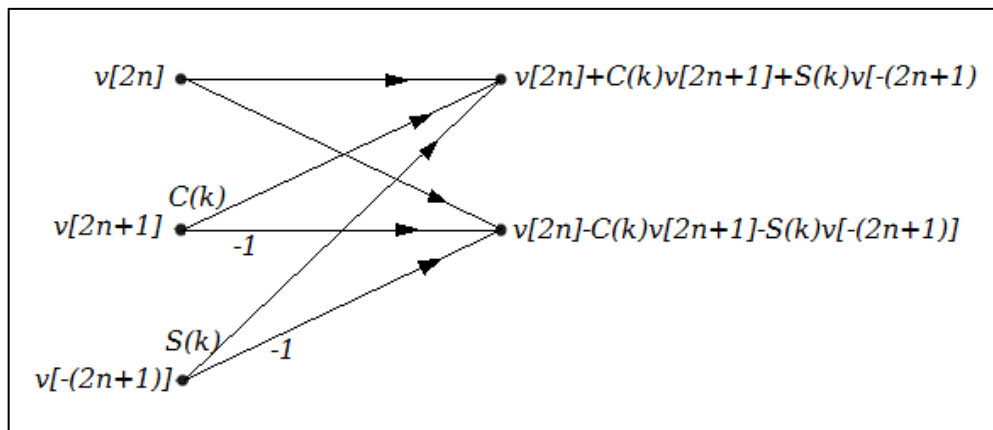


Figura 8 – Representação da borboleta DIT-FHT.

Fonte: Autor.

Assim como foi realizado para DIT-FFT, considerando o exemplo $N=8$, segue:

$$H[k] = \sum_{n=0}^3 v[2n] \cos(2nk\theta) + \cos(k\theta) \sum_{n=0}^3 v[2n+1] \cos(2nk\theta) + \sin(k\theta) \sum_{n=0}^3 v[-(2n+1)] \cos(2nk\theta)$$

$$k = 0, \dots, 3$$

Sendo:

$$A[k] = \sum_{n=0}^3 v[2n] \cos(2nk\theta) \quad e \quad B[k] = \sum_{n=0}^3 v[2n+1] \cos(2nk\theta)$$

Então:

$$A[k] = \sum_{n=0}^1 v[4n] \cos(4nk\theta) + \cos(2k\theta) \sum_{n=0}^1 v[4n+2] \cos(4nk\theta) + \sin(2k\theta) \sum_{n=0}^1 v[-(4n+2)] \cos(4nk\theta)$$

$$A\left[k + \frac{N}{2}\right] = \sum_{n=0}^1 v[4n] \cos(4nk\theta) - \cos(2k\theta) \sum_{n=0}^1 v[4n+2] \cos(4nk\theta) - \sin(2k\theta) \sum_{n=0}^1 v[-(4n+2)] \cos(4nk\theta)$$

$$B[k] = \sum_{n=0}^1 v[4n+1] \cos(4nk\theta) + \cos(2k\theta) \sum_{n=0}^1 v[4n+3] \cos(4nk\theta) + \sin(2k\theta) \sum_{n=0}^1 v[-(4n+3)] \cos(4nk\theta)$$

$$B\left[k + \frac{N}{2}\right] = \sum_{n=0}^1 v[4n+1] \cos(4nk\theta) - \cos(2k\theta) \sum_{n=0}^1 v[4n+3] \cos(4nk\theta) - \sin(2k\theta) \sum_{n=0}^1 v[-(4n+3)] \cos(4nk\theta)$$

para $k = 0, 1$

Notando que para $k=0$ e $k=1$, generalizando para $m = \log_2 N$. Tem-se:

$$\sin(2^{m-1}k\theta) = \sin\left(2^{m-1}k \frac{2\pi}{N}\right) = \sin(k\pi) = 0 \rightarrow k = 0, 1$$

$$\cos(2^{m-1}k\theta) = \cos\left(2^{m-1}k \frac{2\pi}{N}\right) \Rightarrow \cos(0) = 1 \wedge \cos(\pi) = -1$$

Ainda nas transformadas $A[k]$ e $B[k]$ onde $k=(0,1,2,3)$, nestes casos os senos também zeram e os cossenos aparecem como 1 e -1, por isso, nestas duas primeiras camadas pode-se representar o fluxograma idêntico ao utilizado no caso da FFT da figura 7.

Utilizando as considerações supracitadas, torna-se possível representar o fluxograma das borboletas para DIT-FHT com $N=8$, conforme pode ser visto na figura 9. Neste fluxograma, as flechas em azul indicam segmentos multiplicados por cosseno e as flechas vermelhas, por seno. A figura 10 representa a borboleta da FHT de forma a evidenciar a multiplicação dos coeficientes trigonométricos, isto simplifica a programação do algoritmo, bem como a representação do fluxograma.

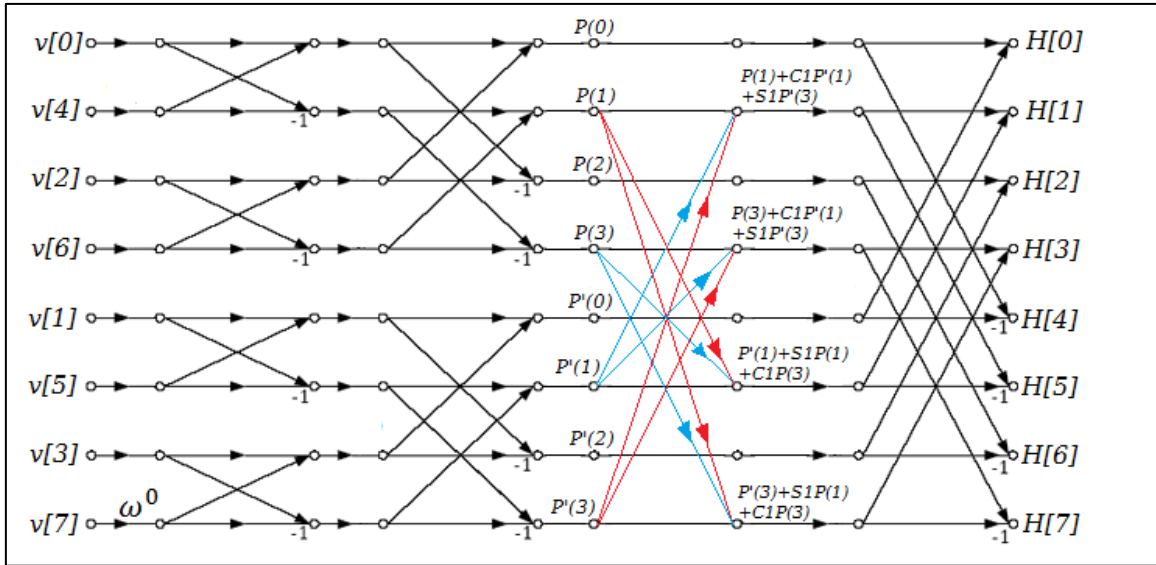


Figura 9 – Fluxograma borboletas DIT-FHT com N=8.

Fonte: Autor.

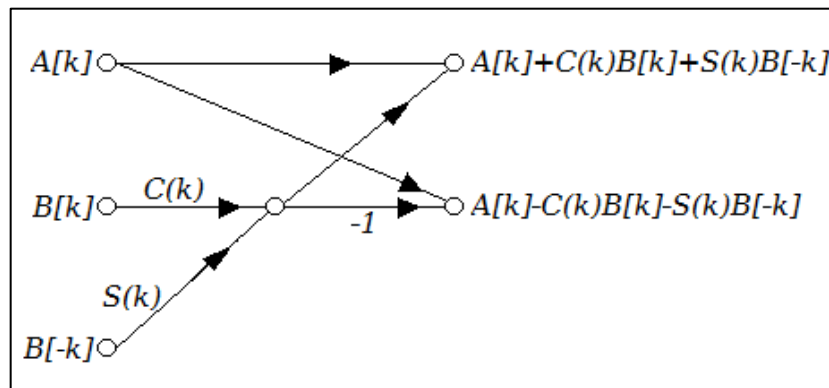


Figura 10 – Operação Borboleta para FHT com parte trigonométrica separada.

Fonte: Autor.

Segundo Blahut (2010), o algoritmo de Cooley-Tukey é recursivo e o número de multiplicações e adições complexas para a transformada rápida de Fourier (FFT) é dado pelas equações recursivas a seguir:

$$M_c(N) = 2M_c\left(\frac{N}{2}\right) + \frac{N}{2} \quad (90)$$

$$A_c(N) = 2A_c\left(\frac{N}{2}\right) + N \quad (91)$$

Onde N é potência de 2, estas equações recursivas (90) e (91) podem ser reescritas da seguinte forma:

$$M_c(N) = \frac{N}{2} \log_2(n) \quad (92)$$

$$A_c(N) = N \log_2(n) \quad (93)$$

Mas cada multiplicação complexa corresponde a quatro multiplicações reais e duas adições reais bem como a adição complexa correspondem a duas adições reais.

Portanto,

$$M_R(N) = 2N \log_2(n) \quad (94)$$

$$A_R(N) = 4N \log_2(n) \quad (95)$$

Em contra partida, para a transformada rápida de Hartley (FHT) verifica-se a seguinte quantidade de operações reais.

$$M_C(N) = N \log_2(n) \quad (96)$$

$$A_C(N) = 2N \log_2(n) \quad (97)$$

Observando estes resultados, excluindo a ordem de complexidade que é idêntica para ambos os casos $O(N \log_2 N)$, temos então seis operações da FFT contra três operações para a FHT, o que apresentaria um ganho de 50% no uso da FHT em relação à FFT, mas esta conta considerou apenas os números de operações por borboleta, o que não é realista, pois a implementação da FHT exige operações extras devido ao fato de sua borboleta não ser simétrica.

1.3 ANÁLISE DE VIBRAÇÕES MECÂNICAS

1.3.1 Fundamentos de vibrações

As vibrações mecânicas são de fundamental importância para os seres humanos, pois estas estão ligadas a um dos nossos sentidos básicos, a audição. Por isso acredita-se que as pessoas começaram a se interessar pelo assunto quando foram desenvolvidos os primeiros instrumentos musicais. Segundo Rao (2008), o matemático grego Pitágoras (582-507 a.C.) é considerado o primeiro a investigar as vibrações das cordas de um instrumento musical utilizando métodos científicos.

Um sistema vibratório é um sistema dinâmico para o qual as variáveis de entrada e as de saídas são dependentes do tempo, e possuem características oscilatórias. Qualquer movimento que se repita ao longo do tempo é denominado oscilação ou ainda vibração.

A teoria de vibrações baseia-se no estudo de movimentos oscilatórios de corpos e das forças envolvidas nesta dinâmica. Em Rao (2008), encontramos a teoria básica para a compreensão destes sistemas de maneira simplificada. Muitas aplicações no âmbito da engenharia podem ser encontradas em Crede (1972) e em Inman (2008). Nestas obras são

analisadas as partes elementares destes sistemas, a quantidade de graus de liberdade, e suas classificações, tais como sistemas discretos ou contínuos, com vibração livre ou forçada, amortecida ou não amortecida, linear ou não linear, determinística ou aleatória. Neste tópico serão abordados apenas sistemas de vibração forçada determinística e linear. Neste caso, a análise harmônica é aplicada e servirá de base para a compreensão dos resultados experimentais. Por isso, é imprescindível revisar alguns conceitos básicos sobre vibrações mecânicas.

Para atingir uma boa compreensão dos sistemas contínuos, apresenta-se aqui o movimento harmônico simples, que pode ser facilmente exemplificado com um sistema discreto com um grau de liberdade, como, por exemplo, o pendulo simples. Seto (1971, p. 4) concebe a seguinte definição:

Para uma partícula em movimento retilíneo, se uma aceleração é sempre proporcional à distância da partícula a um ponto fixo na trajetória e é dirigida para o ponto fixo, dizemos que a partícula tem um movimento harmônico simples, ou simplesmente, MHS. MHS é a forma mais simples do movimento periódico. O movimento periódico de vibração quer simples, quer complexo, pode ser considerado como composto de MHS ou um número de MHS de várias amplitudes e frequências por meio de uma série de Fourier.

Em forma de equação diferencial utilizando a lei do movimento de Newton, temos:

$$a = -Kx \quad (98)$$

$$\ddot{x} + Kx = 0 \quad (99)$$

Resultando em:

$$x = A. \sin \sqrt{K}t + B. \cos \sqrt{K}t = C. \sin(\sqrt{K}t + \phi) \quad (100)$$

Onde, $K = \frac{k}{m}$, tal que m é a massa e k a constante elástica da mola.

O sistema amortecido sob ação de uma força harmônica é aquele em que a força pode ser descrita pela função oscilante $f(t) = f_0 \cos \omega t$. Neste caso, a equação de movimento torna-se:

$$m\ddot{x} + \beta\dot{x} + kx = f_0 \cos \omega t \quad (101)$$

Onde β é o coeficiente de atrito viscoso.

Como se espera que a solução particular para a equação (101) também seja harmônica, assumimos que esta tenha a forma:

$$x_p(t) = A. \cos(\omega t - \phi), \quad (102)$$

onde A e ϕ são constantes a determinar. Onde o A é o módulo da função de transferência harmônica e ϕ é o ângulo de fase, pois:

$$A[(k - m\omega^2) \cos(\omega t - \phi) - \beta\omega \sin(\omega t - \phi)] = f_0 \cos \omega t \quad (103)$$

Utilizando as mesmas relações trigonométricas utilizadas em (88) e (89), porém para a diferença dos ângulos, e igualando os coeficientes de $\cos \omega t$ e $\sin \omega t$ em ambos os lados da equação resultante, obtemos o seguinte sistema:

$$\begin{cases} A[(k - m\omega^2) \cos \phi + \beta\omega \sin \phi] = f_0 \\ A[(k - m\omega^2) \sin \phi - \beta\omega \cos \phi] = 0 \end{cases} \quad (104)$$

A solução de (104) para A e ϕ , é:

$$A = \frac{f_0}{\sqrt{(k - m\omega^2)^2 + \beta^2\omega^2}} \quad (105)$$

$$\phi = \tan^{-1} \left(\frac{\beta\omega}{k - m\omega^2} \right) \quad (106)$$

Utilizando os resultados de (105) e (106) para substituir A e ϕ na equação (102), obtemos a solução particular da equação (101). A figura 11 ilustra os gráficos típicos deste sistema em regime permanente. Para fins práticos as seguintes substituições são utilizadas: $\omega_n = \sqrt{\frac{k}{m}}$ é a frequência natural não amortecida, $\zeta = \frac{\beta}{\beta_c} = \frac{\beta}{2m\omega_n} = \frac{\beta}{2\sqrt{mk}}$; $\frac{\beta}{m} = 2\zeta\omega_n$.

A equação de um sistema de segunda ordem ao ser excitado por um impulso, modelado pelo delta de Dirac, adquire a seguinte forma.

$$m\ddot{x} + \beta\dot{x} + kx = \delta(t) \quad (107)$$

A solução da equação diferencial (107) é apresentada em mais detalhes no Apêndice B, mas sua solução final para $x(t)$ é dada por:

$$x(t) = \frac{1}{m\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t) \mathcal{U}(t) \quad (108)$$

Onde, $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ e $\mathcal{U}(t)$ é a função degrau unitário conforme definida pela equação (4). A figura 12 ilustra um gráfico típico da resposta a este tipo de entrada.

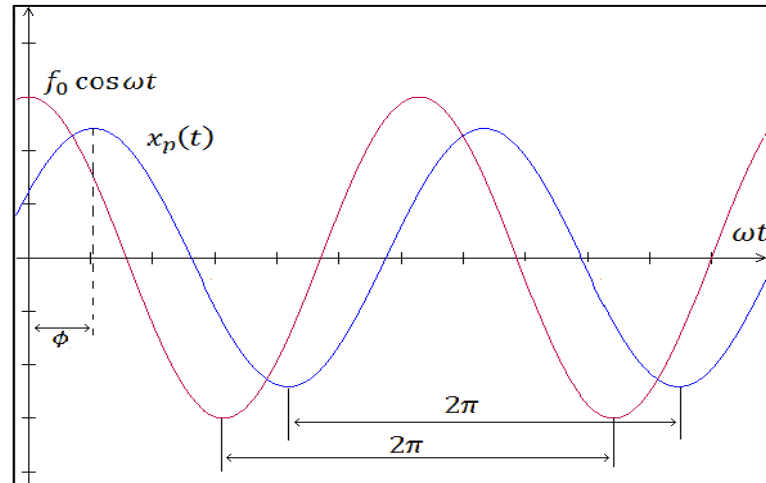


Figura 11 – Representação resposta à força harmônica.

Fonte: Autor

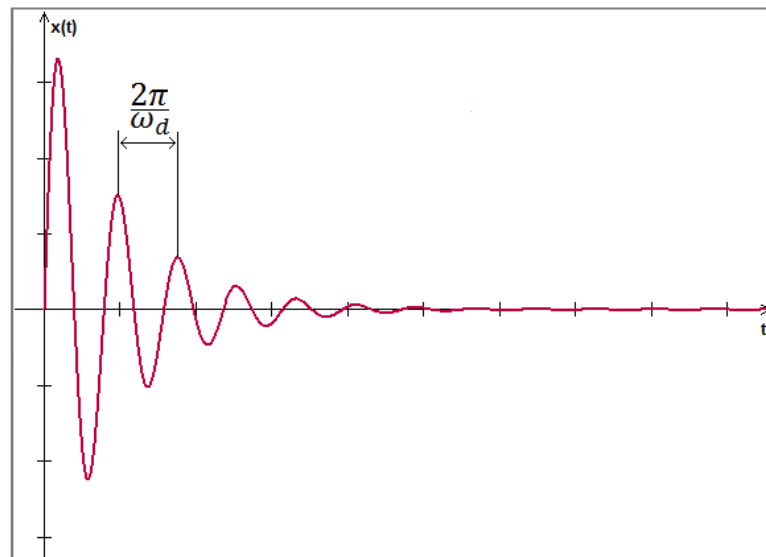


Figura 12 – Representação resposta ao impulso no instante zero.

Fonte: Autor

A ressonância ocorre quando $\omega = \omega_n$, ou seja, quando a frequência de excitação é igual à frequência natural do sistema. Quando esta ocorre, a amplitude aumenta sem limite no modelo teórico sem amortecimento, tal que, na prática a amplitude é limitada apenas pela quantidade de amortecimento presente no sistema, neste caso a ressonância é detectada quando $\omega = \omega_d$. Por isso é de fundamental importância conhecer a frequência natural de um sistema, pois uma vibração de grande amplitude pode provocar efeitos desastrosos a um sistema mecânico.

Segundo Rao (2008), quando a força externa é periódica, esta pode ser expandida em séries de Fourier e a equação de movimento do sistema pode ser expressa por:

$$m\ddot{x} + \beta\dot{x} + kx = f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (109)$$

O lado direito da equação (109) é uma constante mais uma soma de funções harmônicas. Pelo princípio da superposição, podemos escrever a solução como a soma das soluções em regime permanente das seguintes equações.

$$m\ddot{x} + \beta\dot{x} + kx = \frac{a_0}{2} \quad (110)$$

$$m\ddot{x} + \beta\dot{x} + kx = a_n \cos(n\omega t) \quad (111)$$

$$m\ddot{x} + \beta\dot{x} + kx = b_n \sin(n\omega t) \quad (112)$$

A solução da equação (110) é dada por:

$$x_p(t) = \frac{a_0}{2k} \quad (113)$$

Utilizando os resultados (102), (105) e (106), pode-se expressar os resultados de (111) e (112), como:

$$x_p(t) = \frac{a_n k^{-1}}{\sqrt{(1 - n^2 r^2)^2 + (2\zeta nr)^2}} \cos(n\omega t - \phi_n) \quad (114)$$

$$x_p(t) = \frac{b_n k^{-1}}{\sqrt{(1 - n^2 r^2)^2 + (2\zeta nr)^2}} \sin(n\omega t - \phi_n) \quad (115)$$

Onde:

$$\phi_n = \tan^{-1} \left(\frac{2\zeta nr}{1 - n^2 r^2} \right) \text{ e } r = \frac{\omega}{\omega_n} \quad (116)$$

Sendo assim a solução completa da equação diferencial (109) para regime permanente é dada por:

$$x_p(t) = \frac{a_0}{2k} + \sum_{n=1}^{\infty} \left[\frac{a_n k^{-1}}{\sqrt{(1 - n^2 r^2)^2 + (2\zeta nr)^2}} \cos(n\omega t - \phi_n) + \frac{b_n k^{-1}}{\sqrt{(1 - n^2 r^2)^2 + (2\zeta nr)^2}} \sin(n\omega t - \phi_n) \right] \quad (117)$$

Outra informação de interesse é o fator de ampliação $F(r)$ das amplitudes de vibração de um sistema massa-mola-amortecedor em resposta à frequência, em Rao (2008, p.105), encontramos o modelo que pode ser representado pela equação a seguir:

$$F(r) = \frac{\delta_{st}}{\sqrt{(1 - r^2)^2 + 4(\zeta r)^2}} \quad (118)$$

Onde $\delta_{st} = \frac{f_0}{k}$ é a deflexão sob a força estática f_0 .

Na figura 13, temos algumas curvas de resposta do fator de ampliação para um sistema massa-mola-amortecedor com diferentes valores para ζ .

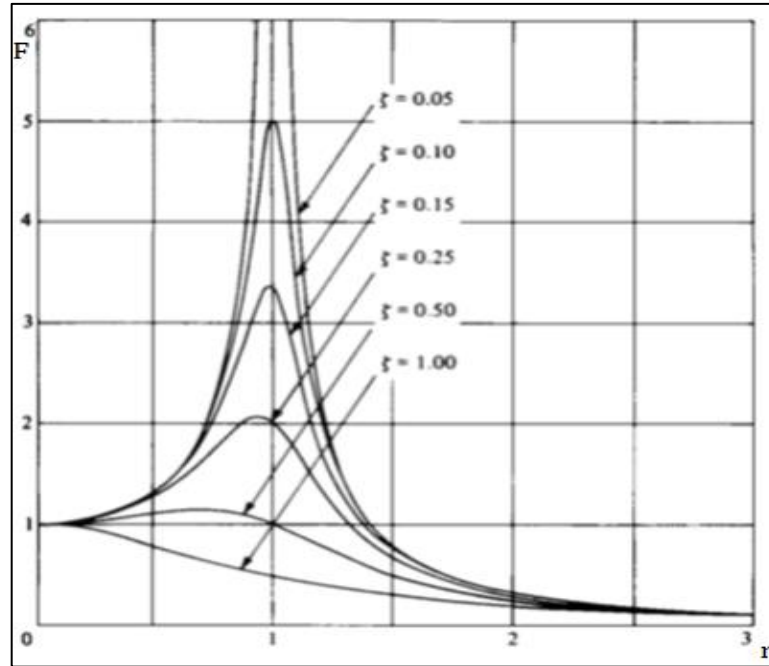


Figura 13 – Resposta em frequência para um massa-mola-amortecedor.

Fonte: <https://www.ime.usp.br/~oda/16_Resp_Sist_Excit_Harm.pdf>

1.3.2 Vibrações de meios contínuos

Segundo Seto (1971), os sistemas mecânicos que tem suas massas e forças elásticas distribuídas, no lugar de agrupados por molas e massas concentradas, pertencem à classe de sistemas contínuos. Estes sistemas são compostos por um número infinitamente grande de partículas e requerem um número infinitamente grande de coordenadas para especificar suas configurações. Sistemas mecânicos desta classe têm um número infinito de frequências naturais e modos naturais de vibração.

Analisando o caso, para a vibração longitudinal de uma barra, tem-se a seguinte equação diferencial de movimento:

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} \quad (119)$$

Onde μ é o deslocamento de uma secção transversal qualquer, a^2 é dado pela relação entre o produto da gravidade e o módulo de elasticidade da barra pelo peso específica da barra, e x é a coordenada ao longo do eixo longitudinal.

A solução geral é dada por:

$$u(x, t) = \sum_{k=1}^{\infty} (A_k \cos \omega_k t + B_k \sin \omega_k t) \left(C_k \cos \frac{\omega_k x}{a} + D_k \sin \frac{\omega_k x}{a} \right) \quad (120)$$

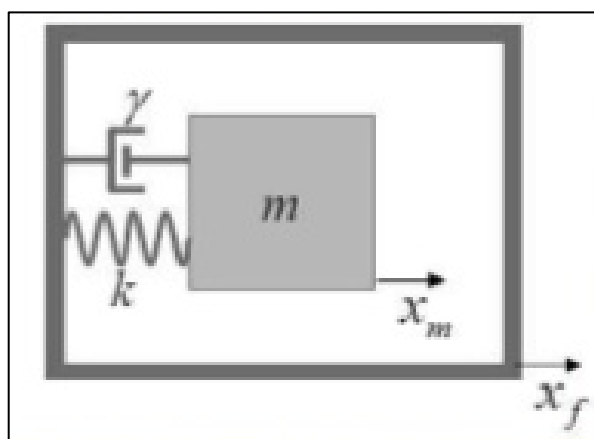
Onde A_k , B_k , C_k , D_k são constantes a serem definidas segundo as condições de contorno e ω_k é a frequência natural do sistema. Além disso, vale notar que quanto maior o valor de k , menor se tornam as amplitudes dos harmônicos, por isso, na prática é possível estabelecer esta solução com limite finito para o somatório da equação (120).

A equação (119) é tipicamente um exemplo de equação diferencial parcial harmônica para a qual se pode encontrar a solução utilizando o método de Fourier para separação de variáveis. Em Figueiredo (2007) e em Iório Jr e Iório (1988) são apresentados os métodos para obtenção desta solução.

1.3.3 Sensores de vibração

Segundo Rao (2008) e Inman (2008), um sensor de vibração é um conjunto de pelo menos um transdutor e um circuito mecânico acoplado ao transdutor, sendo que o transdutor é um dispositivo que transforma valores de grandezas físicas em sinais elétricos.

Os sensores de vibrações são comumente conhecidos como instrumentos sísmicos. Um instrumento sísmico consiste basicamente de um sistema massa-mola-amortecedor disposto sobre um corpo vibratório, conforme ilustrado na figura 14. O movimento vibratório é medido através da determinação do deslocamento da massa em relação à base na qual está montada.



γ – Amortecedor; k – Mola; m – Massa; x_m , x_f – Referenciais.

Figura 14 – Instrumento sísmico.

Fonte: www.mdpi.com/1424-8220/13/5/5720/pdf.

Existem diversos transdutores e dispositivos que podem ser montados para compor um instrumento sísmico e, conseqüentemente, um sensor de vibração. Alguns que se destacam na literatura são os transdutores resistivos, piezelétricos e eletrodinâmicos. O sensor

construído utilizando estes princípios supracitados e qualquer um destes transdutores é dito um Acelerômetro.

O acelerômetro é um sensor que mede a vibração na forma de aceleração do movimento de uma estrutura. A construção do acelerômetro é semelhante à configuração mostrada na figura 14. Por isso o acelerômetro pode ser utilizado como um sensor sísmico.

Figueiredo et al. (2007) ressalta que existem vários tipos de acelerômetros. Cada qual possui características particulares. Entre os acelerômetros mecânicos, os mais comuns são os capacitivos, os piezoelétricos e os piezoresistivos. Porém, recentemente os acelerômetros mecânicos vêm sendo substituídos por acelerômetros eletrônicos, que utilizam a tecnologia *Micro Electro-Mechanical Systems* (MEMS). A família dos sensores MEMS inclui acelerômetros que podem fazer medições de aceleração e vibração em um, dois ou três eixos. O sensor MEMS é baseado numa estrutura de silício, com interligações em forma de pente, composta por dedos fixos e móveis. A aceleração é obtida da medição dos deslocamentos dos elementos móveis que estão associados a cada eixo. O movimento medido pelo transdutor é então convertido em um sinal analógico, que pode estar integrado a um circuito de conversão analógico/digital.

Como exemplo, na figura 15, é apresentado o diagrama esquemático interno do acelerômetro MMA7361L, produzido pela *Freescall Semiconductor*.

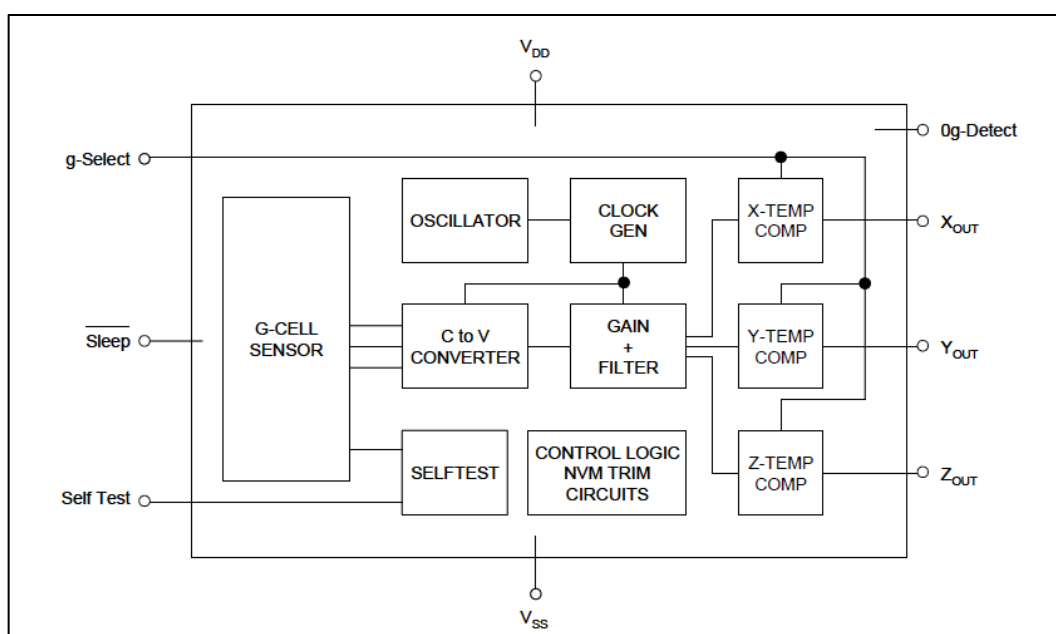


Figura 15 – Diagrama em blocos da estrutura eletrônica do acelerômetro MMA7361L.

Fonte: Datasheet MMA7361L Versão 0, 04/2008.

2. DESEMPENHO COMPARATIVO ENTRE AS TRANSFORMADAS

Neste capítulo são discutidos os aspectos técnicos e teóricos da comparação de eficiência computacional da FFT em relação à FHT aplicadas em um microcontrolador ESP8266, apresentando o desenvolvimento do ferramental para tal comparação e seus resultados.

2.1 MICROCONTROLADOR ESP8266 E IDE ARDUINO

Para a comparação de eficiência computacional, foi escolhido o microcontrolador ESP8266 produzido pela *Espressif Systems*. Este microcontrolador possui diversos recursos de hardware, inclusive Wi-Fi, mas o que mais chamou a atenção para selecioná-lo, dentre vários outros, foi o seu reduzido custo, alta capacidade de processamento e facilidade para programação, utilizando-se a IDE do Arduino (versão 1.6.5) para programação em linguagem C++.

Seguem abaixo as principais características do microcontrolador ESP8266, encontradas em seu *datasheet* ESP8266 (2012):

- *802.11 b/g/n protocol*
- *Wi-Fi Direct (P2P), soft-AP*
- *Integrated TCP/IP protocol stack*
- *Integrated TR switch, balun, LNA, power amplifier and matching network*
- *Integrated PLL, regulators, and power management units*
- *+19.5dBm output power in 802.11b mode*
- *Integrated temperature sensor*
- *Supports antenna diversity*
- *Power down leakage current of $< 10\mu A$*
- *Integrated low power 32-bit CPU could be used as application processor*
- *SDIO 2.0, SPI, UART*
- *STBC, 1×1 MIMO, 2×1 MIMO*
- *A-MPDU & A-MSDU aggregation & 0.4us guard interval*

- *Wake up and transmit packets in $< 2ms$*
- *Standby power consumption of $< 1.0mW$ (DTIM3)*
- *CPU frequency: 80/160MHz*

O microcontrolador ESP8266, vem sendo comercializado em placas de circuito impresso (PCI), com memória RAM de 80 kB e memória *flash* de 4 MB, possuindo também uma antena impressa para facilitar o uso do Wi-Fi. Entre os vários modelos disponíveis no mercado atual, um que vem se popularizando e é facilmente encontrado no mercado brasileiro é o ESP12E. Esta PCI ainda pode ser encontrado em kits de desenvolvimento, como é o caso da PCI NodeMCU v2, que foi utilizada para os experimentos. O NodeMCU é um kit simplificado, onde o ESP12E vem montado, de modo a facilitar o acesso aos seus pinos de entrada e saída. Além disso, o kit conta com um conversor USB/Serial para a programação do ESP12E e depuração do programa durante execução.

A figura 16 é um diagrama onde se tem representada as entradas e saídas do kit utilizado e suas funções, mais informações sobre este módulo podem ser encontradas em Amica (2015).

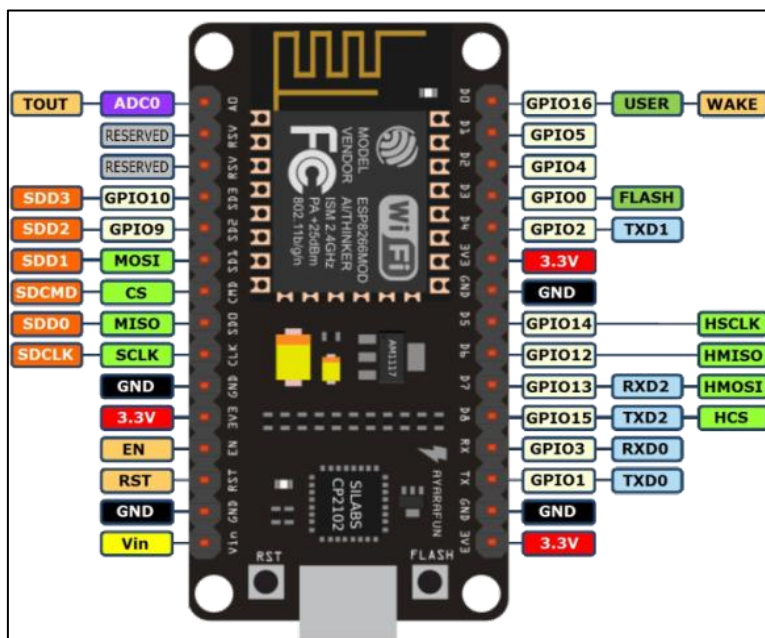


Figura 16 – NodeMCU V2 com ESP12E.

Fonte: https://alexbloggt.com/wp-content/uploads/2015/10/nodemcu_pinout_700.png.

A programação do ESP8266 pode ser feita em seu ambiente nativo com uma SDK (*Software Development kit*) e a IDE (*Integrated Development Environment*) Eclipse, ou

ainda utilizando outras IDEs, inclusive, em linguagem Lua². Mais informações sobre este assunto pode ser encontrada em Kolban (2015) e no site do fabricante Espressif (2016). Dado que este microcontrolador e esta arquitetura são relativamente recentes, existe certa carência de informações mais profundas sobre o assunto. Em contra partida, o que vem sendo extremamente relevante na disseminação desta nova tecnologia, ESP8266, se deve a integração de sua SDK à plataforma Arduino. A IDE Arduino é um ambiente de programação que utiliza linguagem C/C++, cuja tecnologia é conhecida por viabilizar a programação a pessoas não familiarizadas com desenvolvimento de *software* ou *firmware*.

A arquitetura do ESP8266 é conhecida como Xtensa. Em Xtensa (Manual de Referência) encontramos muitos pormenores sobre este assunto, porém, neste ponto é relevante citar apenas que, esta arquitetura deriva várias características RISC (*Reduced Instruction Set Computer*) em um núcleo de 32 bits. Por isso, é muito natural a necessidade do uso de um sistema operacional de tempo real (RTOS – *Real Time Operation System*), principalmente se tratando de um microcontrolador com tal poderio de processamento e recursos avançados como protocolos TCP/IP e acesso Wi-Fi, integrados. No site do fabricante Espressif (2016), encontra-se duas SDKs para a programação do ESP8266, uma com o uso do RTOS e outra sem RTOS.

A SDK utilizada na plataforma Arduino é a que faz uso do RTOS, o que vem a ser uma vantagem, pois, como já foi mencionado, este microcontrolador possui muitos recursos avançados que necessitam de um sistema operacional para facilitar o desenvolvimento e obter uma maior versatilidade e estabilidade do sistema. Mas, esta informação, é extremamente relevante no momento de programar utilizando tal SDK, pois, deve se considerar durante a programação, que o *loop* principal do código programado no Arduino é uma *task* de prioridade máxima no RTOS, porém, esta *task* não deve enterrar o sistema. Para tanto, foram introduzidas algumas funções que fazem a limpeza do *WatchDog*. A sub-rotina *yield()* definida nas bibliotecas do ESP8266 para Arduino, tem justamente esta função (estas informações podem ser encontradas nos próprios códigos e documentos da interface, no repositório [github/esp8266](https://github.com/espressif/esp8266)). Porém, ainda assim, o *loop* principal não deve exceder o tempo de aproximadamente 550 ms para execução de um laço *for*, por exemplo, esta informação foi obtida empiricamente durante os testes dos algoritmos de transformadas

² Lua é uma linguagem de script de multiparadigma, pequena, reflexiva e leve, projetada para expandir aplicações em geral. Por ser uma linguagem extensível (que une partes de um programa feitas em mais de uma linguagem), é utilizada para prototipagem e para ser embarcada em softwares complexos.

discretas. Esta limitação acaba por depor a favor da necessidade de algoritmos rápidos e eficientes para uso em sistemas embarcados com RTOS.

2.2 DESEMPENHO DAS TRANSFORMADAS DISCRETAS

Esta etapa iniciou-se com o desenvolvimento dos algoritmos primeiramente utilizando o compilador Dev-c++ versão 5.11 da *Bloodshed Software*, a fim de testar e verificar os resultados em *prompt*, antes de aplicar os algoritmos em um microcontrolador. Estas validações foram feitas utilizando o software Matlab (versão 2010) e sua biblioteca de FFT. Após verificado os resultados, os códigos em linguagem C foram apenas adaptados para o microcontrolador. Além do ESP8266, os códigos foram também testados no microcontrolador PIC18F2550. Todos os códigos desenvolvidos durante este trabalho estão disponíveis no *Github*³.

Antes da implementação dos algoritmos rápidos, foram implementados os algoritmos discretos de complexidade $O(N^2)$, utilizando algumas simplificações a fim de aumentar seu desempenho. Para deixar a comparação equilibrada, tomamos como referência de saída o espectro absoluto de frequências positivas de Fourier, ou seja, o módulo da DFT para frequências positivas.

A DFT foi implementada apenas para a parte positiva do espectro, o que deverá reduzir o custo de processamento.

A partir destas considerações, foi então implementado o seguinte algoritmo para a DFT:

$$F[k] = \sum_{n=0}^{\frac{N}{2}-1} v[2n]e^{i\frac{4\pi nk}{N}} + \sum_{n=0}^{\frac{N}{2}-1} v[2n+1]e^{i\frac{2\pi k(2n+1)}{N}} \mid k = 0, 1, \dots, \frac{N}{2} - 1 \quad (121)$$

$$V[k] = \frac{1}{N} \sqrt{\text{Re}\{F[k]\}^2 + \text{Im}\{F[k]\}^2} \mid k = 0, 1, \dots, \frac{N}{2} - 1 \quad (122)$$

³ <<https://github.com/JunioCesarFerreira/Fourier-and-Hartley-transforms-for-embedded-applications>>

Enquanto para DHT foi elaborado o seguinte algoritmo:

$$H[k] = \sqrt{2} \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos\left(\frac{4\pi nk}{N} - \frac{\pi}{4}\right) + \sqrt{2} \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos\left(\frac{2\pi k(2n+1)}{N} - \frac{\pi}{4}\right) \quad (123)$$

$$H\left[k + \frac{N}{2}\right] = \sqrt{2} \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos\left(\frac{4\pi nk}{N} - \frac{\pi}{4}\right) - \sqrt{2} \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos\left(\frac{2\pi k(2n+1)}{N} - \frac{\pi}{4}\right) \\ | \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (124)$$

$$V[k] = \frac{\sqrt{2}}{2N} \sqrt{H[k]^2 + H\left[\frac{N}{2} - k\right]^2} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (125)$$

As expressões (123) e (124) são consequências da relação (38), enquanto o resultado (125) vem do desenvolvimento da relação (43) em domínio discreto. Pois:

$$Re\{F[k]\}^2 = \frac{\left(H[k] + H\left[\frac{N}{2} - k\right]\right)^2}{4} = \frac{1}{4} \left(H[k]^2 + 2H[k]H\left[\frac{N}{2} - k\right] + H\left[\frac{N}{2} - k\right]^2\right) \quad (126)$$

$$Im\{F[k]\}^2 = \frac{\left(H[k] - H\left[\frac{N}{2} - k\right]\right)^2}{4} = \frac{1}{4} \left(H[k]^2 - 2H[k]H\left[\frac{N}{2} - k\right] + H\left[\frac{N}{2} - k\right]^2\right) \quad (127)$$

Utilizando a primeira parte da igualdade (38), temos:

$$H[k] = \sum_{n=0}^{N-1} v[n] \cos\left(nk \frac{2\pi}{N}\right) = \sqrt{2} \sum_{n=0}^{\frac{N}{2}-1} v[n] \cos\left(nk \frac{2\pi}{N} - \frac{\pi}{4}\right) = \sqrt{2} H'[k] \quad (128)$$

Substituindo (126) e (127) em (125) considerando também a igualdade (128).

Temos que:

$$V[k] = \frac{1}{2N} \sqrt{2(\sqrt{2}H'[k])^2 + 2\left(\sqrt{2}H'\left[\frac{N}{2} - k\right]\right)^2} = V[k] = \frac{1}{N} \sqrt{H'[k]^2 + H'\left[\frac{N}{2} - k\right]^2} \quad (129)$$

Assim gerou-se por fim o seguinte algoritmo composto pelas expressões (130), (131) e (132):

$$H'[k] = \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos\left(\frac{4\pi nk}{N} - \frac{\pi}{4}\right) + \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos\left(\frac{2\pi k(2n+1)}{N} - \frac{\pi}{4}\right) \quad (130)$$

$$H'\left[k + \frac{N}{2}\right] = \sum_{n=0}^{\frac{N}{2}-1} v[2n] \cos\left(\frac{4\pi nk}{N} - \frac{\pi}{4}\right) - \sum_{n=0}^{\frac{N}{2}-1} v[2n+1] \cos\left(\frac{2\pi k(2n+1)}{N} - \frac{\pi}{4}\right) \quad (131)$$

$$V[k] = \frac{1}{N} \sqrt{H'[k]^2 + H'\left[\frac{N}{2} - k\right]^2} \quad \forall k = 0, 1, \dots, \frac{N}{2} - 1 \quad (132)$$

Nas figuras 17 e 18, podem ser analisadas as seções principais dos algoritmos codificados; os códigos completos encontram-se no *Github*.

Nesta etapa foi feita a comparação dos resultados espectrais de cada transformada, confrontando os resultados obtidos computacionalmente com os resultados teóricos. Foram utilizados sinais virtuais, ou seja, inicializou-se o sistema já com os vetores de entrada prontos, de modo a garantir que as entradas seriam as mesmas para ambas as transformadas.

Além disso, foi desenvolvido um *software* de interface que é executado em um computador e comunica-se com o microcontrolador através do canal de comunicação USB/Serial do próprio kit NodeMCU. Quando o microcontrolador termina o processamento de um vetor de sinal, ele envia o vetor de entrada e o vetor resultante da transformada para o computador; a interface exibe então os gráficos de ambos, podendo também, armazenar os resultados em planilhas.

```

1  double v[N];           // Vetor de entrada
2  double Vreal[N];       // Vetor Saída Real
3  double Vimag[N];       // Vetor Saída Imaginária
4  double pi_N=2*pi/N;    // Auxiliar Pi*n
5  double wre, wie, wro, wio; // Componentes da transformada
6  unsigned int i;        // Indexador de uso geral
7  unsigned int n, k;      // Indexadores da transformada
8  // Realiza amostragem do sinal
9  for (i=0;i<N;i++){
10     digitalWrite(Teste,1);
11     yield();
12     digitalWrite(Teste,0);
13     v[i]=(double)analogRead(A0)*3.3/1023;
14     delayMicroseconds(585); // Sample 1KHz
15 }
16 digitalWrite(Check,HIGH); // Ponto de checagem
17 // Computa a DFT partes Par e Impar
18 for (k=0;k<N/2;k++){
19     Vreal[k]=0;
20     Vimag[k]=0;
21     Vreal[k+N/2]=0;
22     Vimag[k+N/2]=0;
23     yield();
24     for (n=0;n<N/2;n++){
25         wre = cos(pi_N*2*n*k);
26         wie = sin(pi_N*2*n*k);
27         wro = cos(pi_N*k*(2*n+1));
28         wio = sin(pi_N*k*(2*n+1));
29         Vreal[k]+=(wre*v[2*n]+wro*v[2*n+1]);
30         //Vreal[k+N/2]+=(wre*v[2*n]-wro*v[2*n+1]);
31         Vimag[k]-=(wie*v[2*n]+wio*v[2*n+1]);
32         //Vimag[k+N/2]-=(wie*v[2*n]-wio*v[2*n+1]);
33     }
34     Vreal[k]=Vreal[k]/N;
35     Vimag[k]=Vimag[k]/N;
36     //Vreal[k+N/2]=Vreal[k+N/2]/N;
37     //Vimag[k+N/2]=Vimag[k+N/2]/N;
38 }
39 yield();
40 // Computa módulo da DFT
41 for (i=1;i<N/2;i++){
42     Vreal[i]=sqrt(Vreal[i]*Vreal[i]+Vimag[i]*Vimag[i]);
43     //Vreal[N-i]=Vreal[i];
44 }
45 digitalWrite(Check,LOW);

```

Figura 17 – Código DFT utilizado para comparação.

Fonte: autor.

```

1  double v[N];           // Vetor de entrada
2  double V[N];           // Vetor de saída
3  double pi_N=2*pi/N;    // Auxiliar Pi*n
4  double wre, wro;       // Componentes Pares e impares
5  unsigned int i;        // Indexador de uso geral
6  unsigned int n, k;     // Indexadores das transformadas
7
8  // Realiza amostragem do sinal
9  for (i=0;i<N;i++){
10     digitalWrite(Teste,1);
11     yield();
12     digitalWrite(Teste,0);
13     v[i]=(double)analogRead(A0)*3.3/1023;
14     delayMicroseconds(585); // Sample 1KHz
15     //delayMicroseconds(80); // Sample 2KHz
16 }
17
18 digitalWrite(Check,HIGH); // Ponto de checagem
19 // Computa DHT
20 for (k=0;k<N/2;k++){
21     V[k]=0;
22     V[k+N/2]=0;
23     yield();
24     for (n=0;n<N/2;n++){
25         wre = cos(pi_N*2*n*k-pi/4);
26         wro = cos(pi_N*k*(2*n+1)-pi/4);
27         V[k]+=(wre*v[2*n]+wro*v[2*n+1]);
28         V[k+N/2]+=(wre*v[2*n]-wro*v[2*n+1]);
29     }
30     V[k]= V[k]/N;
31     V[k+N/2]= V[k+N/2]/N;
32 }
33 yield();
34 // Computa T. de Fourier atraves da T. Hartley
35 for (i=1;i<N/2;i++){
36     V[i]=sqrt(V[i]*V[i]+V[N-i]*V[N-i]);
37     //V[N-i]=V[i];
38 }
39 digitalWrite(Check,LOW);

```

Figura 18 – Código DHT utilizado para comparação.

Fonte: autor.

Na figura 19, é mostrada a tela de interface do software, onde se pode observar sua exibição gráfica e informações de parametrização. A aplicação foi desenvolvida na plataforma Visual Studio da Microsoft 2012, utilizando a linguagem C#. Este código também se encontra disponível no *Github* supracitado. Esta aplicação permite ainda, salvar os resultados em forma de planilhas para análises posteriores, além de gerar um vetor de saída que pode ser utilizado no *software* Matlab (2010) para comparações. Os resultados são enviados com precisão de centésimos de hertz.

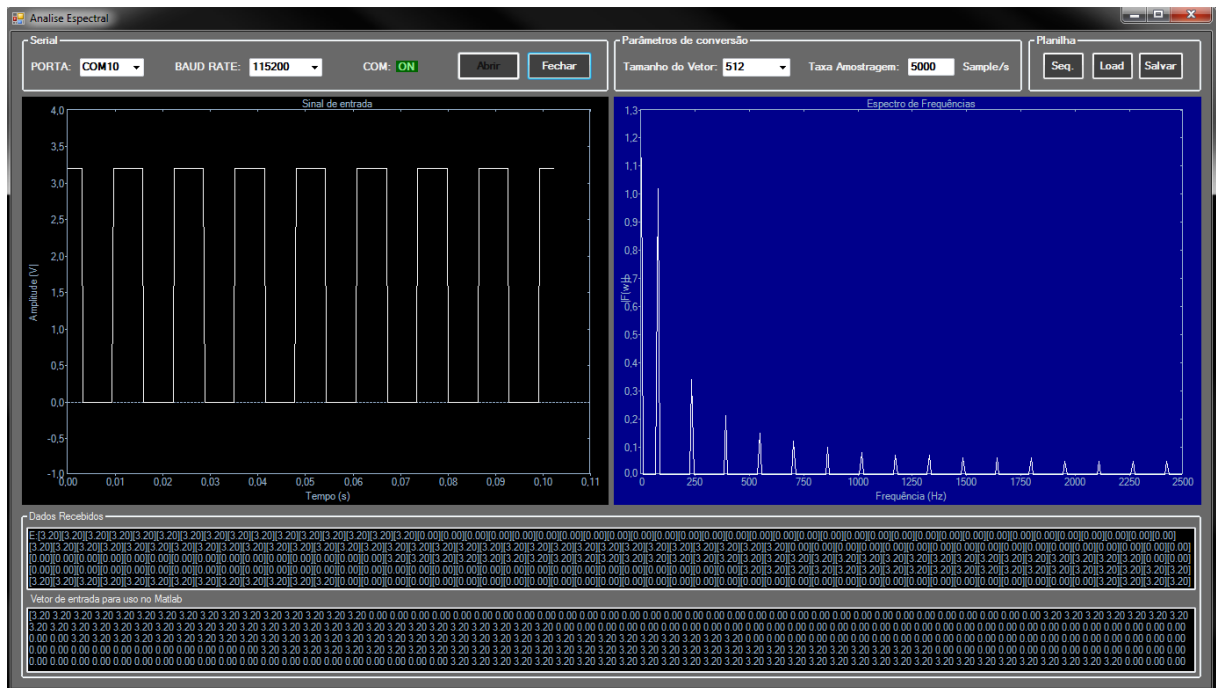


Figura 19 – Interface desenvolvida para auxiliar na verificação dos resultados.
Fonte: Autor.

2.2.1 Tempo de processamento

A medição do tempo de processamento foi realizada por intermédio de um osciloscópio digital, da marca Minipa, modelo MO-2100, 100 MHz. Como pode ser verificado nos códigos apresentados nas figuras 17 e 18, há um ponto de checagem, por meio da mudança de estado de um pino do microcontrolador. Este muda seu estado para alto no início do processamento e retorna para baixo no final. Através do sinal de saída deste pino, foi mensurado o tempo de processamento. Durante a mensuração, foram feitos processamentos de diferentes comprimentos de vetores de entrada. Os resultados podem ser vistos na tabela 1 e no gráfico da figura 20. Ao notar que conforme explicado no tópico 2.1, o algoritmo não deve ultrapassar o tempo de 550 ms para ser executado, por isso a partir de $N=144$, não constam os tempos da DFT, pois o sistema fica instável e trava. Além disso, para ambos os códigos foram retiradas as partes que calculam a segunda parte do vetor de saída, pois para esta aplicação seu resultado é redundante, devido ao fato de que a entrada sempre é real e nos interessa apenas o espectro de frequências em valores absolutos, porém, a DHT é computada completa, conforme pode ser verificado na figura 18, somente no momento da conversão é que existe a remoção da parte negativa.

Tabela 1 – Resultado da comparação de eficiência entre DFT e DHT.

N	DFT(ms)	DHT(ms)
16	5,08	3,48
32	23,3	14,3
48	54,6	33
64	100	58,4
80	159	93,2
96	232	134
112	320	184
128	420	243
144	-	310
160	-	380
176	-	464
192	-	548

Fonte: Autor.

Ao fazer um ajuste de curvas para o gráfico da figura 20, utilizando a ferramenta de linhas de tendência do Microsoft Excel (2010), obtemos as seguintes funções:

Para DFT:

$$t(n) = 0,0271n^2 - 0,1928n + 1,4418 \quad (133)$$

Para DHT:

$$t(n) = 0,0152n^2 - 0,0491n + 0,0623 \quad (134)$$

As funções (133) e (134) confirmam a ordem de complexidade esperada $O(N^2)$, conforme visto no resultado (68). Ao utilizar os coeficientes dos termos quadráticos das funções (133) e (134), obtemos a seguinte relação de ganho percentual (G_t) para tempo de processamento, DHT para DFT:

$$G_t = \frac{0,0271 - 0,0152}{0,0271} 100 = 43,9\% \quad (135)$$

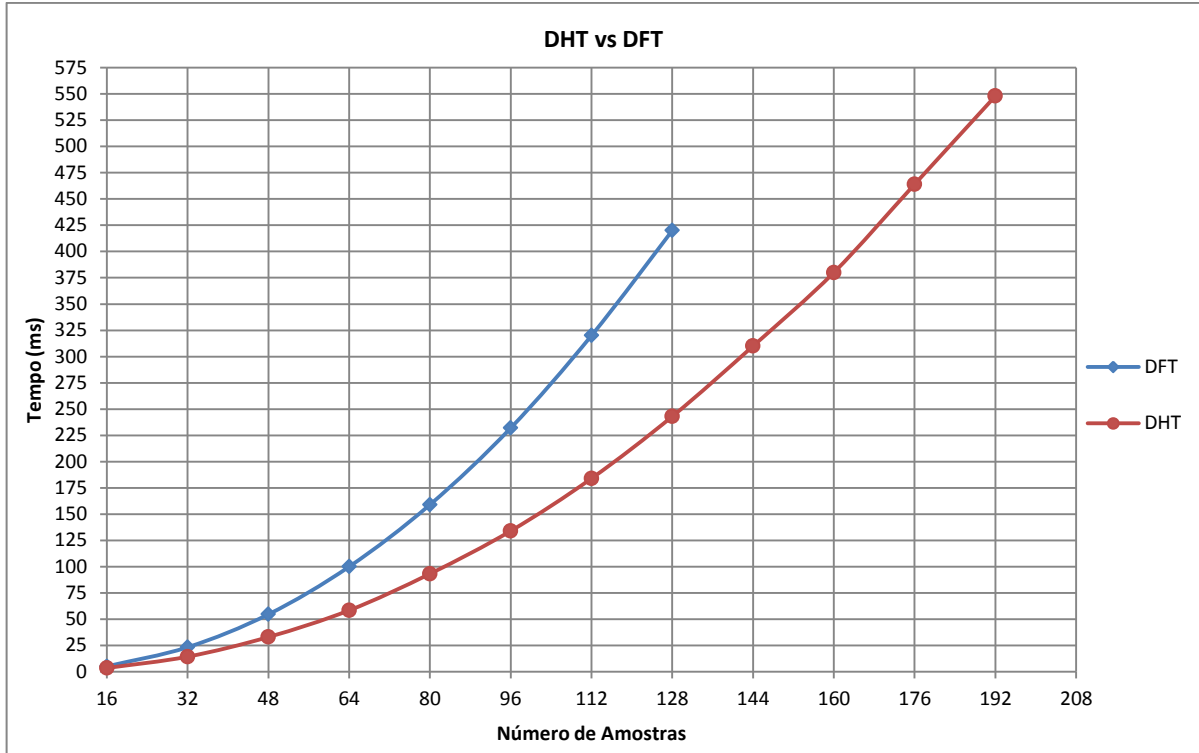


Figura 20 – Resultado da comparação de eficiência entre DFT e DHT.

Fonte: Autor.

2.2.2 Consumo de memória

O RTOS do sistema ocupa 33018 bytes de memória RAM e 198648 bytes de memória de programa, independente do programa ou do *loop* principal. Os valores apresentados na tabela 2 representam os consumos subtraídos o consumo do RTOS. Na figura 21, imprime-se a tabela 2 em gráfico, correspondente ao consumo de memória RAM.

Ajustando retas ao gráfico da figura 21 temos a equação (136) para DHT e a equação (137) para a DFT:

$$m(n) = 16n + 832 \quad (136)$$

$$m(n) = 24n + 848 \quad (137)$$

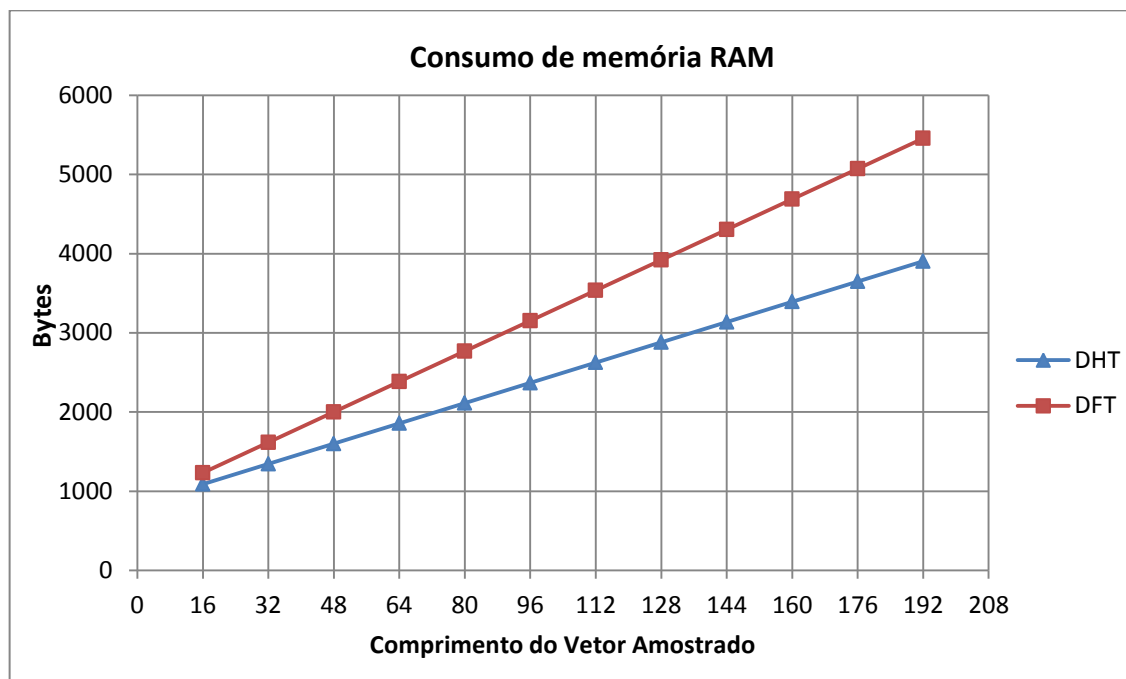
Utilizando os coeficientes angulares das equações (136) e (137) podemos computar o ganho de memória da DHT em relação à DFT programadas:

$$G_m = \frac{24 - 16}{24} 100 = 33,3\% \quad (138)$$

Tabela 2 – Consumo de memória para transformadas discretas

N	DHT		DFT	
	PROG(Bytes)	RAM(Bytes)	PROG(Bytes)	RAM(Bytes)
16	15048	1088	15416	1232
32	15048	1344	15432	1616
48	15048	1600	15432	2000
64	15048	1856	15432	2384
80	15048	2112	15432	2768
96	15048	2368	15432	3152
112	15064	2624	15432	3536
128	15064	2880	15432	3920
144	15064	3136	15432	4304
160	15064	3392	15432	4688
176	15064	3648	15432	5072
192	15064	3904	15432	5456

Fonte: Autor

**Figura 21** – Resultado da comparação de eficiência entre DFT e DHT.

Fonte: Autor.

2.2.3 Comparação com a transformada contínua

Os resultados das transformadas foram armazenados em planilhas por meio do *software* de interface supracitado.

Optou-se por fazer a comparação dos resultados teóricos ao empregar um sinal de onda quadrada, por este ser de fácil análise utilizando a transformada de Fourier de tempo contínuo, e possuir uma boa quantidade de harmônicos dentro da faixa estabelecida pela DFT.

Os gráficos apresentados nas figuras 22 e 23 são referentes a um sinal quadrado de 62,5 Hz, este sinal foi amostrado a 1 kHz, em um vetor de comprimento 128. Na figura 22, temos o gráfico do sinal amostrado, sendo que no eixo das abcissas temos os valores do tempo em milissegundos, enquanto nas ordenadas, temos o valor de tensão elétrica em mili-Volts; na figura 23, o gráfico de sua DFT ou espectro de frequências em Hertz.

O resultado apresentado na figura 23 foi obtido por ambos os algoritmos, validando assim que o algoritmo que utilizou a DHT para computar a DFT, descrito pelas equações (130), (131) e (132), produz os mesmos resultados que a DFT calculada por coeficientes complexos.

A função correspondente ao sinal que aparece na figura 21 pode ser expressa como extensão periódica de período T da seguinte função real e limitada:

$$f_0(t) = \begin{cases} A, & |t| < \frac{T}{4} \\ 0, & |t| > \frac{T}{4} \end{cases} \quad (139)$$

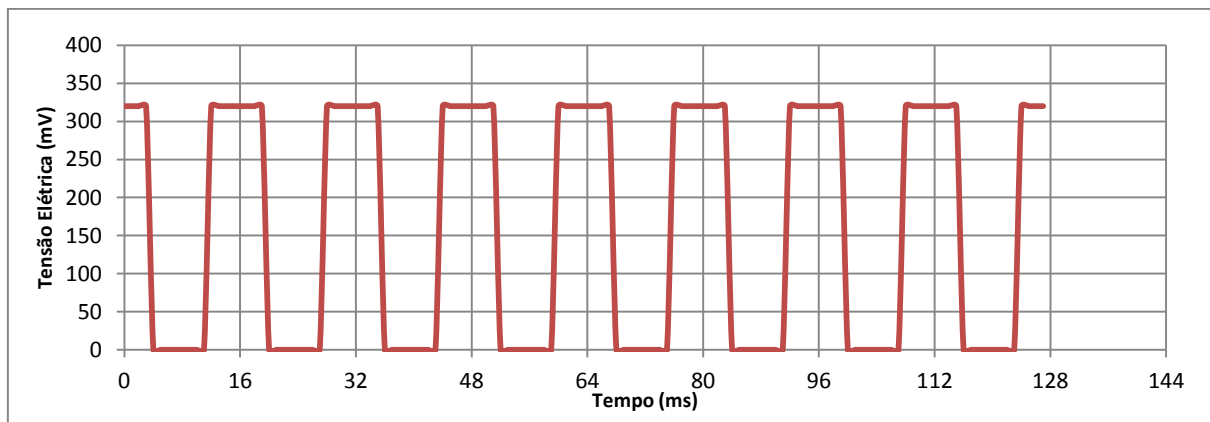


Figura 22 – Sinal amostrado, onda quadrada de 62,5Hz.

Fonte: Autor.

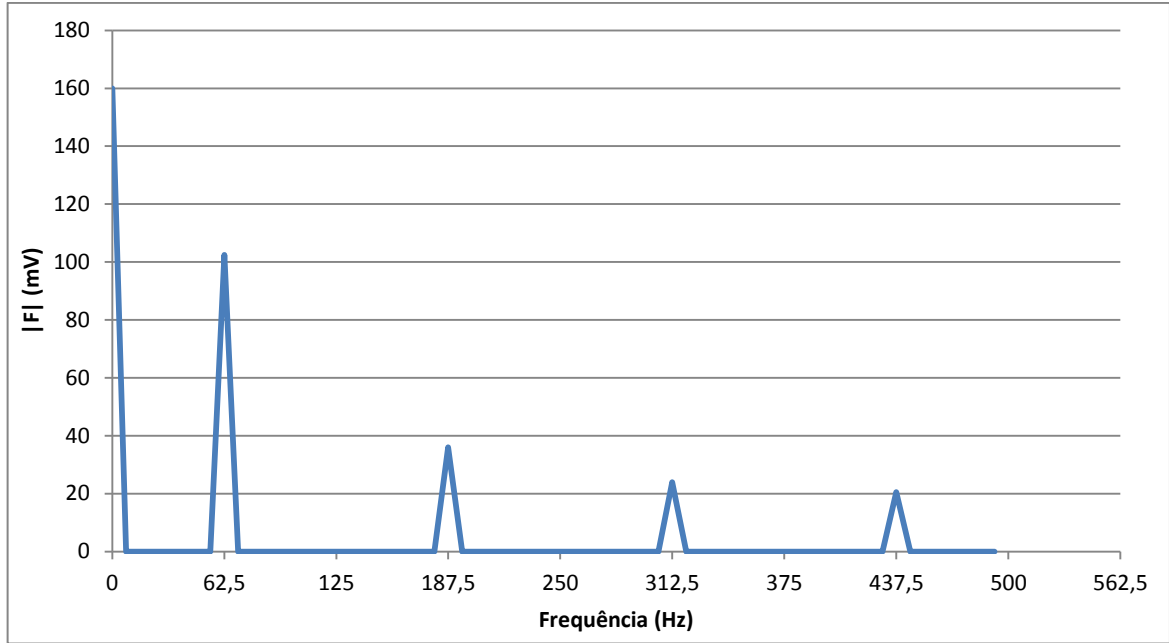


Figura 23 – Espectro de frequências de Fourier, resultado da DFT.

Fonte: Autor.

Na expressão (139), temos que A é a amplitude do sinal, estendendo $f_0(t)$ periodicamente, obtemos $f(t)$ que pode ser representado por séries de Fourier. Servindo-se da formula (21) para calcular os coeficientes c_n , tem-se:

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f_0(t) e^{-2\pi i n t / T} dt$$

Por meio da relação (29), pode-se escrever que:

$$F_0(\omega) = \int_{-\infty}^{\infty} f_0(t) e^{-i\omega t} dt \Rightarrow F_0(n\omega_0) = \int_{-\frac{T}{2}}^{\frac{T}{2}} f_0(t) e^{-in\omega_0 t} dt$$

Portanto,

$$c_n = \frac{1}{T} F_0(n\omega_0) \quad (140)$$

Então,

$$\begin{aligned} F_0(\omega) &= \mathcal{F}\{f_0(t)\} = \int_{-\infty}^{\infty} f_0(t) e^{-i\omega t} dt = A \int_{-\frac{T}{4}}^{\frac{T}{4}} e^{-i\omega t} dt \\ &= \frac{A}{-i\omega} e^{-i\omega t} \Big|_{-\frac{T}{4}}^{\frac{T}{4}} = \frac{A}{i\omega} \left[e^{i\omega \frac{T}{4}} - e^{-i\omega \frac{T}{4}} \right] = \frac{2A}{\omega} \operatorname{sen}\left(\frac{\omega T}{4}\right) \\ &= A \frac{T \operatorname{sen}\left(\frac{\omega T}{4}\right)}{\left(\frac{\omega T}{4}\right)} = A \frac{T}{4} \operatorname{sinc}\left(\frac{\omega T}{4}\right) \end{aligned} \quad (141)$$

Substituindo o resultado (141) em (140),

$$c_n = \frac{A}{2} \text{sinc}\left(\frac{\omega T}{4}\right) \text{ onde } \omega = n\omega_0 \text{ e } \omega_0 = \frac{2\pi}{T}$$

Assim pode-se representar $f(t)$ por:

$$f(t) = \frac{A}{2} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\pi}{2}\right) e^{\frac{i2\pi nt}{T}}$$

Utilizando o resultado (49) vem que a transformada de Fourier $\mathcal{F}\{f(t)\}$ é:

$$\mathcal{F}\{f(t)\} = A\pi \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\pi}{2}\right) \delta(\omega - n\omega_0)$$

Porém deve-se considerar uma normalização por 2π para gerar o espectro de frequências em Hertz. Chega-se a seguinte equação:

$$\mathcal{F}\{f(t)\} = \frac{A}{2} \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\pi}{2}\right) \delta(\omega - n\omega_0) \quad (142)$$

Tomando $T = \frac{1}{62,5}$ (s), então $\omega_0 = 125\pi$. Considerando ainda que $A = 3,2(V)$. Finalmente chega-se a solução da transformada:

$$F(\omega) = 1,6 \sum_{n=-\infty}^{\infty} \text{sinc}\left(\frac{n\pi}{2}\right) \delta(\omega - 125\pi n) \quad (143)$$

Na figura 23, é apresentado o gráfico da transformada de Fourier (TF) resultante da equação (143), onde a curva tracejada é o valor absoluto da função $\text{sinc}\left(\frac{n\pi}{2}\right)$. A tabela 3 apresenta as frequências e os valores da DFT ao lado da TF.

Tabela 3 – Comparativo resultado da DFT em relação a TF

F(Hz)	DFT	F(Hz)	TF
0	1,6	0	1,6
62,5	1,02	60	1,02
187,5	0,36	180	0,34
312,5	0,24	300	0,2
437,5	0,2	420	0,15

Fonte: Autor

O gráfico ilustrado na figura 25 apresenta o erro percentual da transformada discreta computada em relação à transformada contínua, ou seja, o erro percentual da diferença do gráfico da figura 23 em relação ao gráfico da figura 24, considerando apenas o espectro positivo.

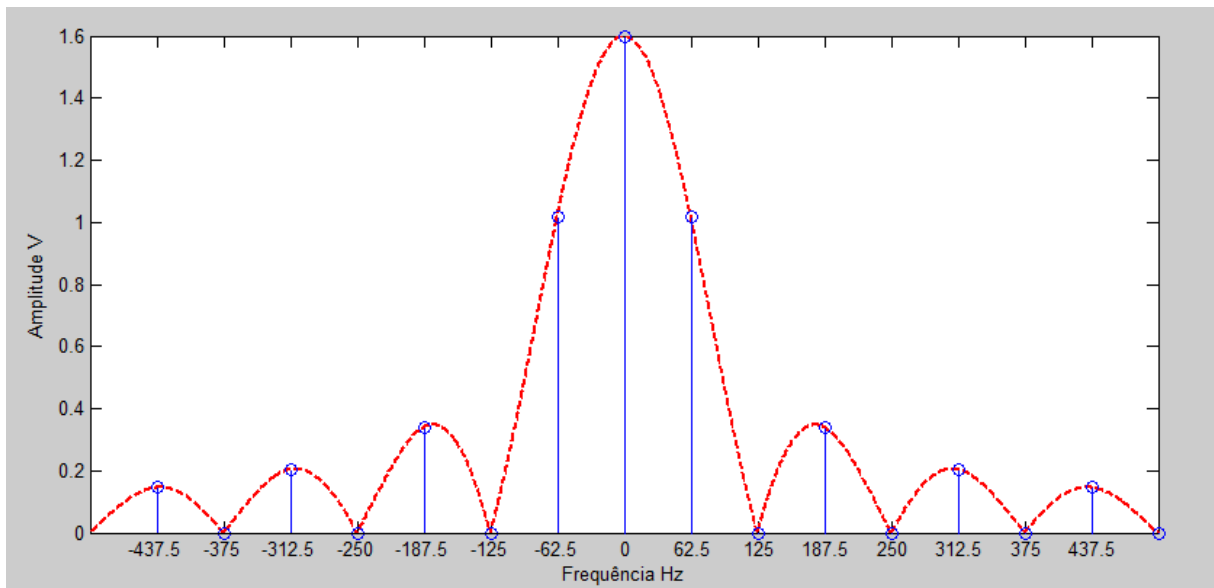


Figura 24 – Espectro de frequências de Fourier, módulo do resultado da TF.
Fonte: Autor.

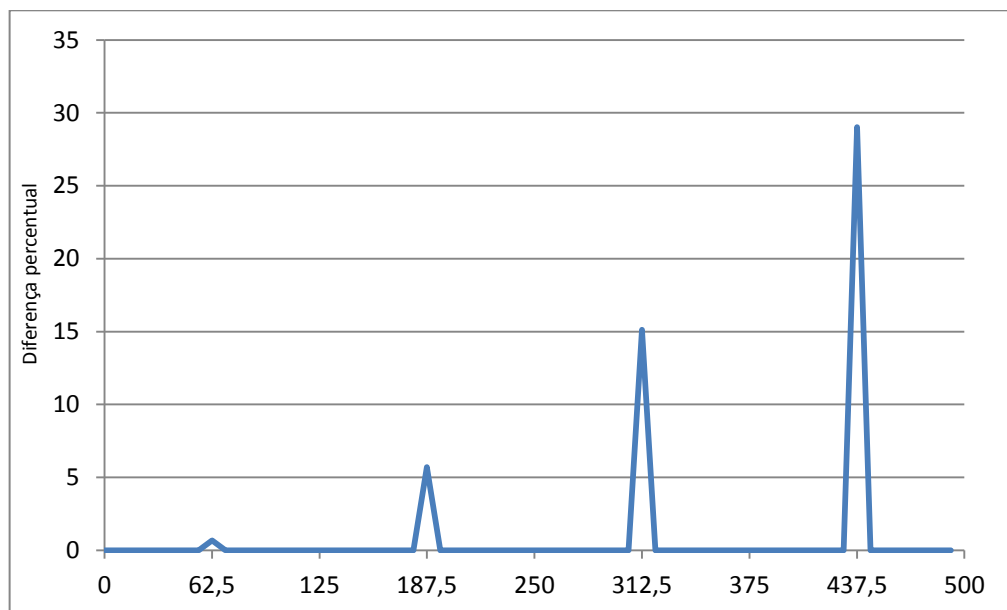


Figura 25 – Erro percentual em cada frequência do espectro gerado pela DFT em relação a TF.
Fonte: Autor.

Nota-se no gráfico do erro percentual, apresentado na figura 25, que existe um erro considerável para os harmônicos mais distantes da origem. Acredita-se, que para fins práticos, esse erro se torna desprezível, visto que ele se concentra em harmônicos de baixa intensidade. Embora seja necessário um estudo mais abrangente para verificar essa suspeita. Outra consideração são os limites dos extremos das transformadas.

2.3 DESEMPENHO DAS TRANSFORMADAS RÁPIDAS

As transformadas rápidas foram implementadas por meio do algoritmo de *Cooley-Tukey radix-2*, valendo-se das noções apresentadas no tópico 1.2.5. Porém, é importante fazer algumas considerações sobre codificação dos algoritmos propostos.

Ao codificar as transformadas discretas, conforme visto nas figuras 17 e 18, foram utilizadas as funções trigonométricas da própria biblioteca de matemática do Arduino. Em geral, estas funções são calculadas por meio de aproximações por polinômios de Taylor, o que apresenta bons resultados, porém demanda considerável processamento. Outro método seria utilizar tabelas predefinidas. Em particular, no caso dos coeficientes trigonométricos para computar a FFT e FHT *Cooley-Tukey radix-2*, observa-se que existe uma simetria natural no círculo trigonométrico que permite gerar uma tabela relativamente pequena, mas que atenda todas as funções trigonométricas necessárias. Por exemplo, quando $N=8$, na figura 26 temos os pontos dos coeficientes para este caso dispostos sobre o plano complexo normalizado. Os números, 0 a 7, em cada ponto, são os indexadores que aparecem no expoente do coeficiente $e^{-i\frac{2\pi}{N}k}$, ou ainda, as raízes sem repetição. Na tabela 4, são dispostos os valores para cada valor de k sobre o eixo dos reais e sobre o eixo dos imaginários. Logo, para o caso $N=8$, temos apenas três valores diferentes, que são $\{1; 0,7071; 0\}$, basta indexá-los de maneira adequada e alternar o sinal quando necessário.

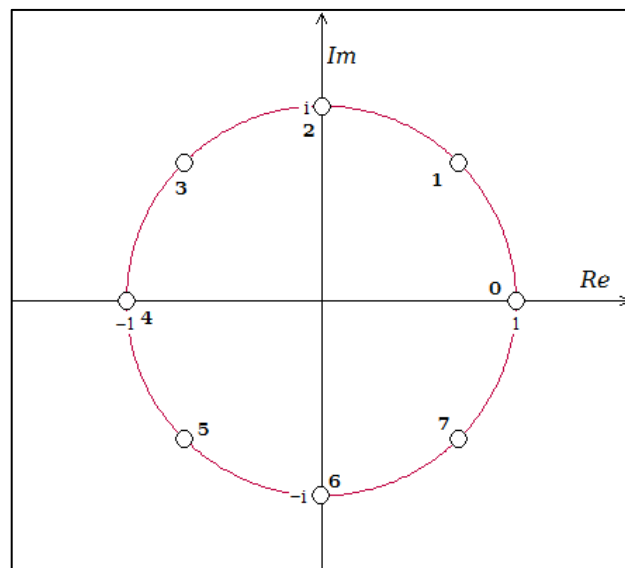


Figura 26 – Distribuição dos coeficientes da DHT sobre o plano complexo.
Fonte: Autor.

Tabela 4 – Valores sobre os eixos real e imaginário da figura 22.

k	Re	Im
0	1	0
1	0,7071	0,7071
2	0	1
3	-0,7071	0,7071
4	-1	0
5	-0,7071	-0,7071
6	0	-1
7	0,7071	-0,7071

Fonte: Autor.

No exemplo anterior, se mudarmos o valor de N para 16, os valores de k que são números pares corresponderão ao caso em que N é igual a 8, e assim por diante, pois estamos trabalhando com comprimentos apenas na forma 2^m , onde m é um número natural. Limitando N com valor máximo de 512, é possível gerar um vetor de apenas 129 posições, que atenderá a todos os coeficientes necessários para N=512. Este vetor atende aos casos N={8, 16, 32, 64, 128, 256, 512}, apenas adequando a forma de indexar os coeficientes. O trecho de código da figura 27, mostra como é feita esta indexação. Nas linhas 4,5 e 6 estão em comentário as funções trigonométricas usuais, que são sub-rotinas que computam seus resultados utilizando-se de tabelas ou polinômios finitos de Taylor. O vetor ‘tab_trig’ pode ser visto na figura 28.

```

1 // Carrega vetor de coeficientes
2 f=512/N;
3 for (k=0;k<N/4;k++){
4     //aux1=2*pi*k/N
5     //Wreal[i] = cos(aux1);
6     //Wimag[i] = -sin(aux1);
7     aux1=i*f;
8     Wreal[i] = tab_trig[aux1];
9     Wreal[i+N/4]= -tab_trig[128-aux1];
10    Wimag[i] = -tab_trig[128-aux1];
11    Wimag[i+N/4]= -tab_trig[aux1];
12 }
```

Figura 27 – Trecho de código que carrega coeficientes para FFT.**Fonte:** Autor.


```

1  const double tab_trig[] =
2  {
3      1,          0.999924702, 0.999698819, 0.999322385, 0.998795456, 0.998118113, 0.997290457, 0.996312612,
4      0.995184727, 0.99390697,  0.992479535, 0.990902635, 0.98917651,  0.987301418, 0.985277642, 0.983105487,
5      0.98078528,  0.978317371, 0.97570213,  0.972939952, 0.970031253, 0.966976471, 0.963776066, 0.960430519,
6      0.956940336, 0.95330604,  0.949528181, 0.945607325, 0.941544065, 0.937339012, 0.932992799, 0.92850608,
7      0.923879533, 0.919113852, 0.914209756, 0.909167983, 0.903989293, 0.898674466, 0.893224301, 0.88763962,
8      0.881921264, 0.876070094, 0.870086991, 0.863972856, 0.85772861,  0.851355193, 0.844853565, 0.838224706,
9      0.831469612, 0.824589303, 0.817584813, 0.810457198, 0.803207531, 0.795836905, 0.788346428, 0.780737229,
10     0.773010453, 0.765167266, 0.757208847, 0.749136395, 0.740951125, 0.732654272, 0.724247083, 0.715730825,
11     0.707106781, 0.698376249, 0.689540545, 0.680600998, 0.671558955, 0.662415778, 0.653172843, 0.643831543,
12     0.634393284, 0.624859488, 0.615231591, 0.605511041, 0.595699304, 0.585797857, 0.575808191, 0.565731811,
13     0.555570233, 0.545324988, 0.53499762,  0.524589683, 0.514102744, 0.503538384, 0.492898192, 0.482183772,
14     0.471396737, 0.460538711, 0.44961133,  0.438616239, 0.427555093, 0.41642956,  0.405241314, 0.39399204,
15     0.382683432, 0.371317194, 0.359895037, 0.34841868,  0.336889853, 0.325310292, 0.31368174,  0.302005949,
16     0.290284677, 0.278519689, 0.266712757, 0.25486566,  0.24298018,  0.231058108, 0.21910124,  0.207111376,
17     0.195090322, 0.183039888, 0.170961889, 0.158858143, 0.146730474, 0.134580709, 0.122410675, 0.110222207,
18     0.09801714,  0.085797312, 0.073564564, 0.061320736, 0.049067674, 0.036807223, 0.024541229, 0.012271538,
19     0
20 };

```

Figura 28 – Trecho de código vetor de tabela trigonométrica.

Fonte: Autor.

Ao empregar a relação de Euler, equação (16), verifica-se que o mesmo acontece com o núcleo da transformada de Hartley discreta. Por isso utilizou-se este vetor para as duas transformadas rápidas, FFT e FHT.

Baseado na teoria do tópico 1.2.5, foi codificado o algoritmo da FFT. A parte principal do código pode ser vista na figura 29. Diferentemente dos algoritmos aplicados para computar as transformadas discretas, o algoritmo de Cooley-Tukey exige que todo o espectro seja computado, pois as bisseções fazem uso de todos os pontos do vetor, conforme visto no fluxograma da figura 7.

```

1 // Computa FFT Cooley-Tukey radix-2
2 for (P=N;P>=2;P=P/2){
3     yield();
4     NP = (N/P); // Comprimento da DFT decomposta
5     L=P/2; // Limite para deslocamentos da Butterfly do nível atual
6     desl = 2*NP; // Fator de deslocamento interno do nível
7     // Deslocamento da Butterfly principal
8     for (k=0;k<NP;k++){
9         aux2=k*P/2;
10        // Deslocamento das Butterflies semelhantes
11        for (i=0;i<L;i++){
12            // Prepara entradas da Butterfly
13            aux1=k+i*desl;
14            aar=Xreal[aux1];
15            aai=Ximag[aux1];
16            bbr=Xreal[NP+aux1];
17            bbi=Ximag[NP+aux1];
18            // Computa Butterfly DIT ...
19            Br=Wreal[aux2];
20            Ar=Wimag[aux2];
21            Bi=Wimag[aux2];
22            Ai=Wreal[aux2];
23            Br*=bbr;
24            Ar*=bbi;
25            Bi*=bbr;
26            Ai*=bbi;
27            Br-=Ar;
28            Bi+=Ai;
29            Xreal[aux1]=aar+Br;
30            Ximag[aux1]=aai+Bi;
31            Xreal[NP+aux1]=aar-Br;
32            Ximag[NP+aux1]=aai-Bi;
33        }
34    }
35 }
36 // Normaliza resultado
37 yield();
38 for (i=0;i<N/2;i++){
39     Xreal[i]=sqrt(Xreal[i]*Xreal[i]+Ximag[i]*Ximag[i])/N;
40 }

```

Figura 29 – Trecho principal do código que computa FFT Cooley-Tukey Radix-2.

Fonte: Autor.

Para codificar a FHT, tomou-se por base a operação borboleta da figura 7; os mesmos mecanismos supramencionados para computar os coeficientes da FFT foram também utilizados. A estrutura principal do código pode ser vista na figura 30. A equação (125) é codificada nas linhas 36, 37 e 38. É neste ponto em que a DFT é computada a partir do resultado da FHT.

```

1 // Carrega vetor para processamento
2 for (i=0;i<N;i++) X[Table_Reverse(i,N)]=v[i];
3 // Carrega vetor de coeficientes
4 k=512/N;
5 for (i=0; i<N/4; i++){
6     aux1=i*k; // Base do indexador
7     C[i] = tab_trig[aux1];
8     C[i+N/4]= -tab_trig[128-aux1];
9     S[i] = tab_trig[128-aux1];
10    S[i+N/4]= tab_trig[aux1];
11 }
12 // Computa FHT Cooley-Tukey radix-2
13 for (P=N; P>=2; P=P/2){
14     NP = N/P; // Comprimento da DHT decomposta
15     L = P/2; // Limite para deslocamentos da Butterfly do nível atual
16     desl = 2*NP; // Fator de deslocamento interno do nível
17     // Deslocamento da Butterfly principal
18     for (k=0;k<NP;k++){ // Desloca grupo de butterflys
19         aux2 = Complemento(k,NP)+NP-k; // Indexador complementar
20         aux3 = k*P/2; // Indexador para vetores de coeficientes
21         // Deslocamento Butterflys do nível
22         for (i=0;i<L;i++){ // Desloca Butterfly
23             aux1 = k + i*desl; // Indexador da Butterfly
24             // Prepara entradas da Butterfly
25             a = X[aux1];
26             b = X[NP+aux1] * C[aux3];
27             c = X[aux2+aux1] * S[aux3];
28             // Computa Butterfly DIT
29             Ha[aux1] = a + b + c;
30             Ha[NP+aux1] = a - b - c;
31         }
32     }
33     for (i=0;i<N;i++) X[i]=Ha[i];
34 }
35 // Computa T. de Fourier através da T. Hartley
36 for (i=0;i<N/2;i++){
37     X[i] = (sqrt(2)/2) * sqrt(X[i]*X[i] + X[N-i]*X[N-i])/N;
38 }

```

Figura 30 – Código principal que computa FHT Cooley-Tukey Radix-2.

Fonte: Autor.

2.3.1 Tempo de processamento

Para obtenção dos resultados práticos, assim como no caso das transformadas discretas, a medição do tempo de processamento foi realizada por meio de um osciloscópio digital. Além disso, o sinal utilizado para a medição do tempo foi o mesmo, uma onda quadrada de 62,5 Hz. Outros sinais foram testados, mostrando variações no tempo de processamento. Isto ocorre devido ao fato de estarmos trabalhando com números com ponto

flutuante padrão IEEE754⁴, do tipo *double* (64 bits); suas multiplicações e somas podem demandar tempos diferentes, dependendo dos valores no vetor. Porém, este fator é apenas uma constante de escala com relação à ordem assintótica, por isso utilizou-se como parâmetro o mesmo sinal já utilizado nas comparações anteriores.

Na tabela 5 são apresentados os resultados das medições. A coluna à direita da tabela 5 apresenta o ganho relativo de cada medição, calculado utilizando a formula (147).

$$G_t = \frac{Tempof_{ft} - Tempof_{ht}}{Tempof_{ft}} * 100 \quad (147)$$

Tabela 5 – Resultados dos tempos de processamento das transformadas rápidas.

N	T(ms) FFT	T(ms) FHT	Ganho(%)
8	0,16	0,144	10
16	0,256	0,208	18,8
32	0,464	0,364	21,5
64	0,936	0,696	25,6
128	2	1,42	29
256	4,36	3,04	30,3
512	9,5	6,56	31
1024	21,8	14,2	34,9

Fonte: Autor.

Na figura 32, encontra-se a representação dos dados da tabela 5 com duas curvas ajustadas por $T(n) = K_j n \log_2(n)$, conservando o resultado da integral neste intervalo. Onde K_j é o coeficiente ajustado para cada curva.

As funções utilizadas nos ajustes de curvas representados na figura 32 são:

$$T_{fft} = 2,19 \cdot 10^{-3} \cdot n \cdot \log_2(n) \quad (148)$$

$$T_{fht} = 1,44 \cdot 10^{-3} \cdot n \cdot \log_2(n) \quad (149)$$

Utilizando os coeficientes K_j das funções (148) e (149), pode-se calcular o ganho percentual da resolução via FHT em relação à FFT complexa, por:

$$G_t = \frac{2,19 - 1,44}{2,19} 100 = 34,7\% \quad (150)$$

⁴ O padrão IEEE 754, definido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), foi adotado em 1985 e desde então sofreu algumas modificações. Este padrão define algumas regras de normalização a serem seguidas nas operações e representações com ponto flutuante representado em binário.

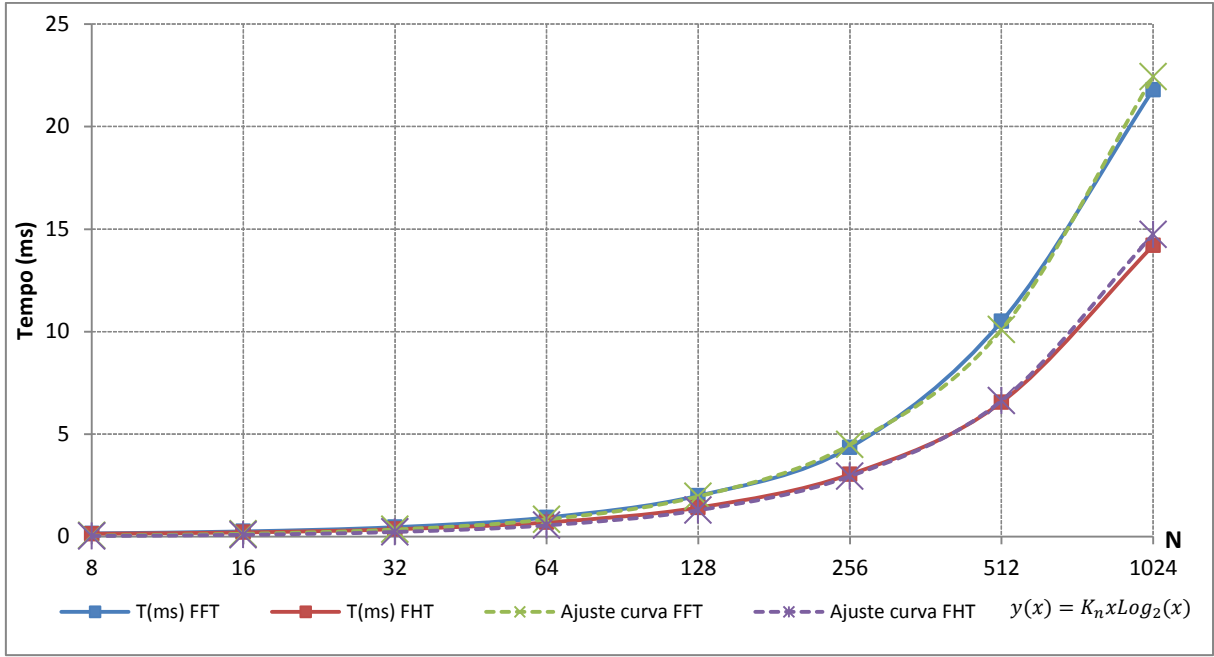


Figura 32 – Tempo de processamento das transformadas rápidas com ajuste de curvas.

Fonte: Autor.

Conforme indicado no final do tópico 1.2, o ganho esperado é variável segundo o comprimento do vetor de entrada.

Valendo-se de uma análise das estruturas dos algoritmos para FFT e FHT implementados neste trabalho, pode-se procurar explicar o ganho experimental obtido nos tempos de processamento da seguinte maneira.

Se designamos por O_{BFFT} o número de operações reais executada para cômputo da FFT, então:

$$O_{BFFT} = S_{\mathbb{R}} + M_{\mathbb{R}} = 4 + (2 + 4) + 2 = 12 \quad (144)$$

Onde $S_{\mathbb{R}}$ designa o número de somas e $M_{\mathbb{R}}$ o número de multiplicações. Isto porque são necessárias quatro somas reais, correspondentes as duas somas complexas utilizadas, além de duas somas reais mais quatro multiplicações reais, referentes a uma multiplicação complexa. Devemos ainda acrescentar duas somas reais, que são realizadas durante a preparação das entradas da borboleta da FFT.

Se designarmos por O_{BFHT} o número de operações por borboleta da FHT implementada neste trabalho, então pode-se modelar este número por:

$$O_{BFHT}(m) = (S_{\mathbb{R}} + M_{\mathbb{R}}) + \frac{2}{mN} \left[\left(\sum_{n=0}^{m-1} 2^{n+2} \right) + \frac{mN}{2} + \frac{N}{2} \right] \quad (145)$$

Onde, $N = 2^m \forall m \in \mathbb{N}$ e $S_{\mathbb{R}} + M_{\mathbb{R}} = 4 + 2 = 6$.

Isto porque considerou-se quatro operações a mais para cada deslocamento de borboleta dentro de um ciclo, e a cada mudança de ciclo temos $N/2$ operações a mais. No final do ultimo ciclo, durante a conversão dos resultados, ainda existe mais $N/2$ operações. Portanto, o ganho será variável em relação ao comprimento dos vetores de entrada, resultando na expressão $\frac{2}{mN} \left[\left(\sum_{n=0}^{m-1} 2^{n+2} \right) + \frac{mN}{2} + \frac{N}{2} \right]$ dependente de m .

Deste modo, pode-se definir a função ganho $\Gamma(m)$ dada por:

$$\Gamma(m) = \frac{O_{BFFT} - O_{BFHT}(m)}{O_{BFFT}} \quad (146)$$

Na figura 31, temos o gráfico de $\Gamma(m)$ para $m = 1, \dots, 12 \Rightarrow N = 2, \dots, 5096$.

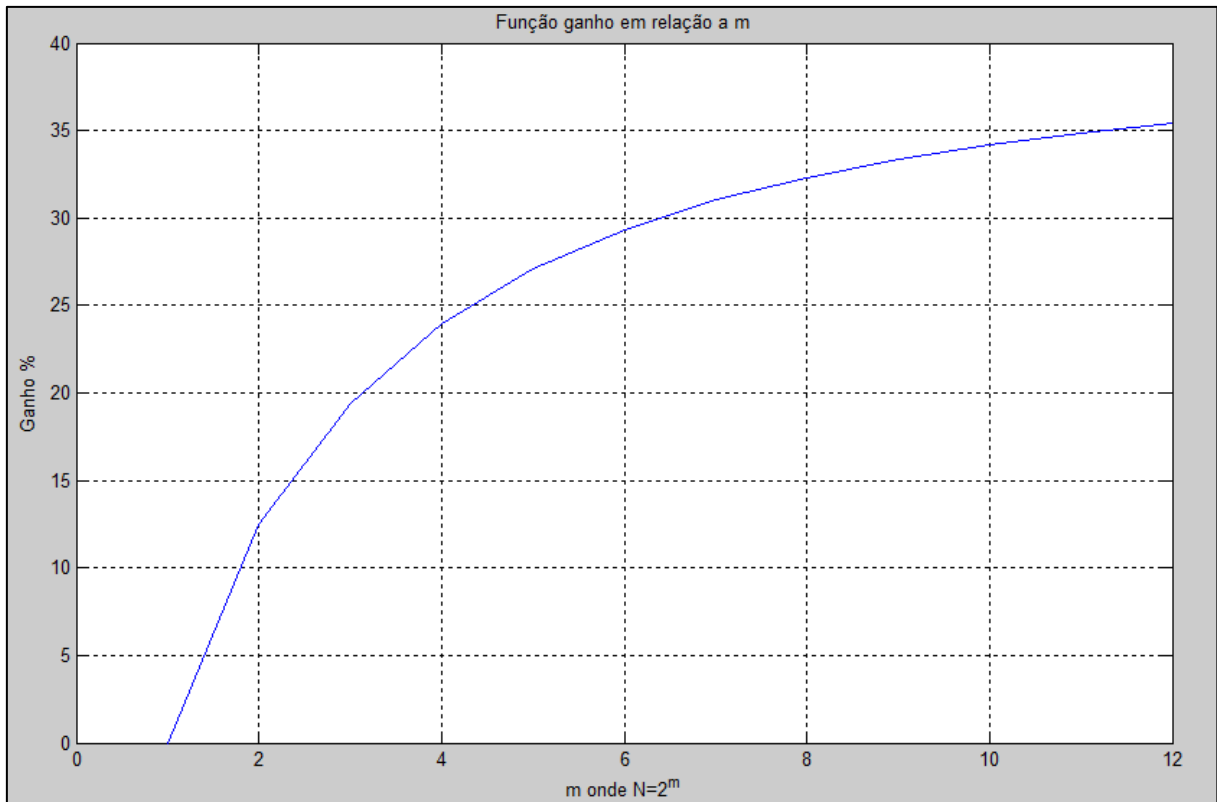


Figura 31 – Função de ganho teórico $\Gamma(m)$ da FHT em relação à FFT.

Fonte: Autor.

Ao comparar o resultado experimental com o gráfico da figura 31, nota-se que existe uma grande discrepância para os valores iniciais, isto se deve à simplicidade da análise realizada, uma análise mais detalhada deverá ser realizada em trabalhos futuros.

2.3.2 Consumo de memória

Assim como no tópico 2.2.2, aqui foi desconsiderado o consumo do RTOS (Sistema operacional de tempo real). A tabela 6 apresenta os resultados de consumo de memória. Embora a DHT não precise de componente imaginária, para implementar a FHT é necessário um vetor de comprimento N , que é utilizado como auxiliar durante a computação das borboletas. Isso se deve ao deslocamento da parte trigonométrica ter um elemento em complemento, conforme ilustrado na figura 10. Por isso, o consumo de RAM da FHT e da FFT é o mesmo. Este consumo é ilustrado no gráfico da figura 33.

Tabela 6 – Consumo de memória para transformadas rápidas.

N	FHT		FFT	
	PROG(Bytes)	RAM(Bytes)	PROG(Bytes)	RAM(Bytes)
16	8356	1740	8452	1740
32	8356	2252	8468	2252
64	8356	3276	8468	3276
128	8356	5324	8468	5324
256	8356	9420	8468	9420
512	8356	17612	8468	17612
1024	7308	32956	7420	32956

Fonte: Autor.

Conforme já mencionado foram utilizadas variáveis do tipo *double*. O que pode ser feito para reduzir o consumo de memória pela metade é utilizar um pouco menos de precisão na representação de ponto flutuante, passando as variáveis para o tipo *float*. Isto reduz o consumo de memória pela metade, porém não provoca alterações significativas na comparação entre os algoritmos relativamente ao tempo de processamento.

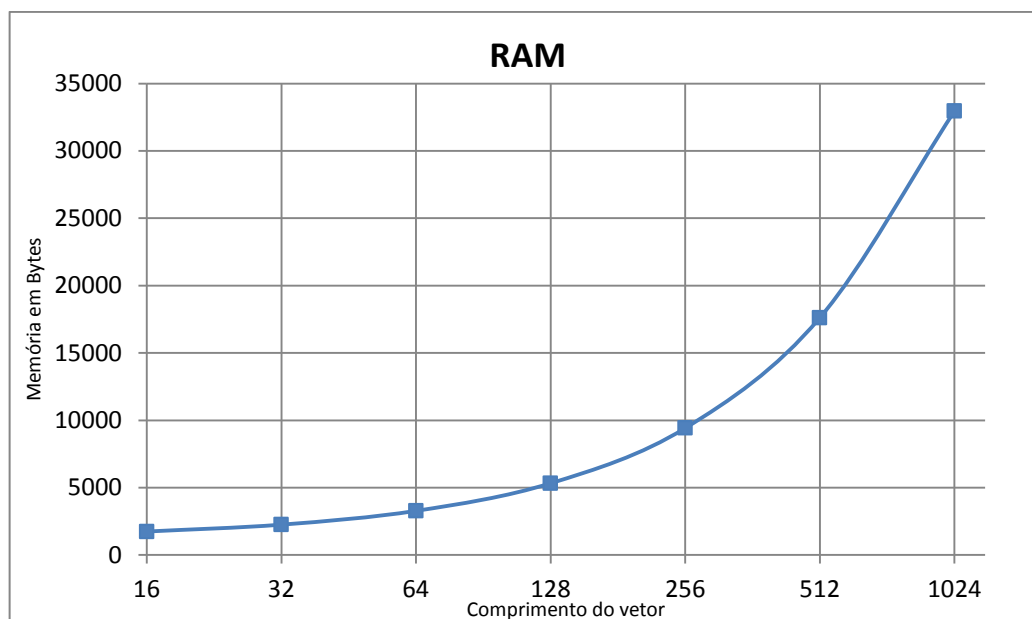


Figura 33 – Gráfico consumo de memória RAM transformadas rápidas utilizando *double*.

Fonte: Autor.

2.3.3 Validação dos resultados

Utilizando a interface desenvolvida para recepção de dados, tanto o resultado via FFT quanto via FHT para computar a DFT, foram armazenados em planilhas e confrontados com os resultados anteriores do tópico 2.2.3. Aplicou-se como critério uma comparação com precisão de dois dígitos decimais, os resultados se mostram idênticos para os sinais utilizados no experimento, validando assim os resultados das transformadas rápidas. As figuras 34, 35 e 36 exibem gráficos de resultados computados no ESP12E. Estes resultados foram comparados com os resultados obtidos no Matlab; exceto por um erro de ordem de 10^{-14} , os resultados são os mesmos. Este erro pode ser justificado pelo erro de truncamento⁵ no formato com ponto flutuante utilizado pelo Matlab e pelo microcontrolador serem diferentes. Porém, um estudo aprofundado sobre esta questão fugiria ao escopo deste trabalho.

⁵ O truncamento de uma representação binária para n bits é severamente tendencioso, já que sempre conduz a um número cujo valor absoluto é inferior ou igual ao valor exato, assim, possivelmente causando acumulação significativa de erro durante uma longa sequência de cálculos.

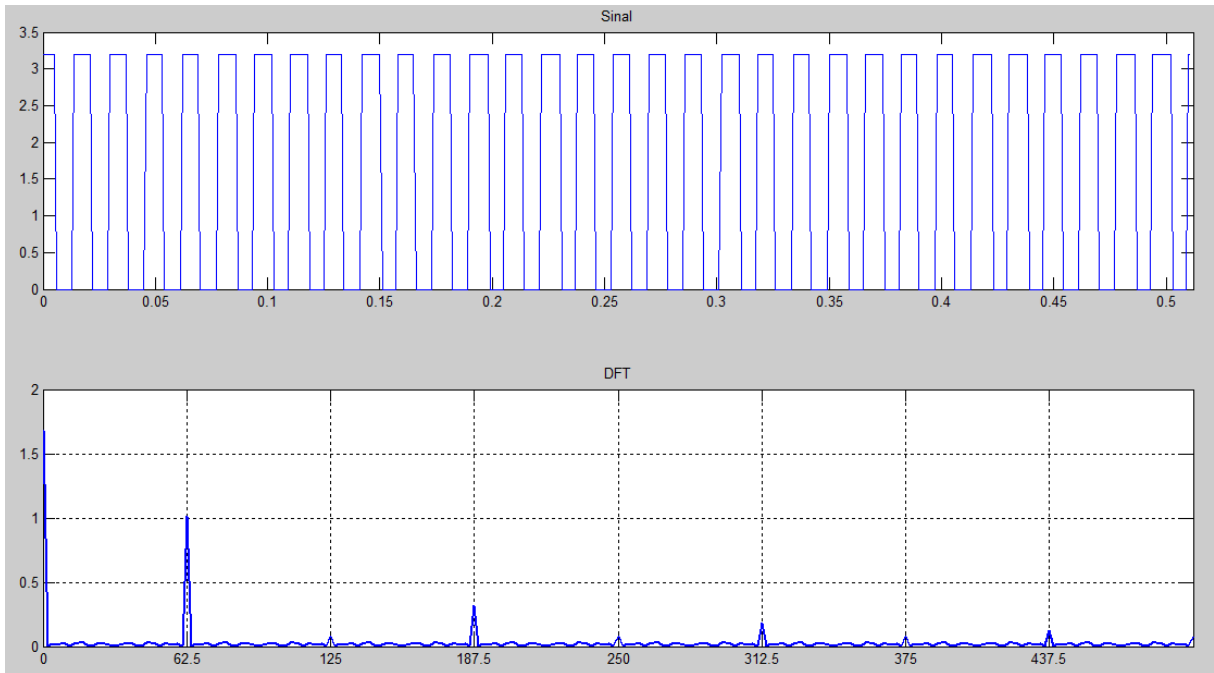


Figura 34 – Sinal quadrado no tempo e resultado da FFT no domínio da frequência.
Fonte: Autor.

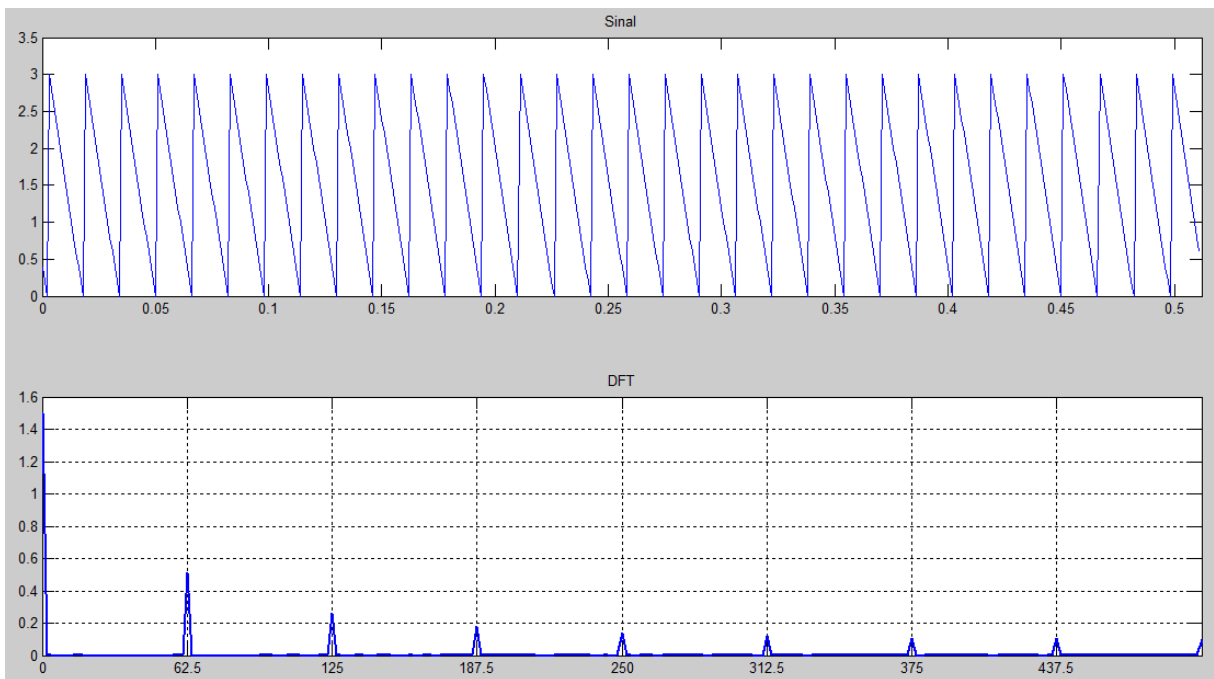


Figura 35 – Sinal dente de serra no tempo e resultado da FFT no domínio da frequência.
Fonte: Autor.

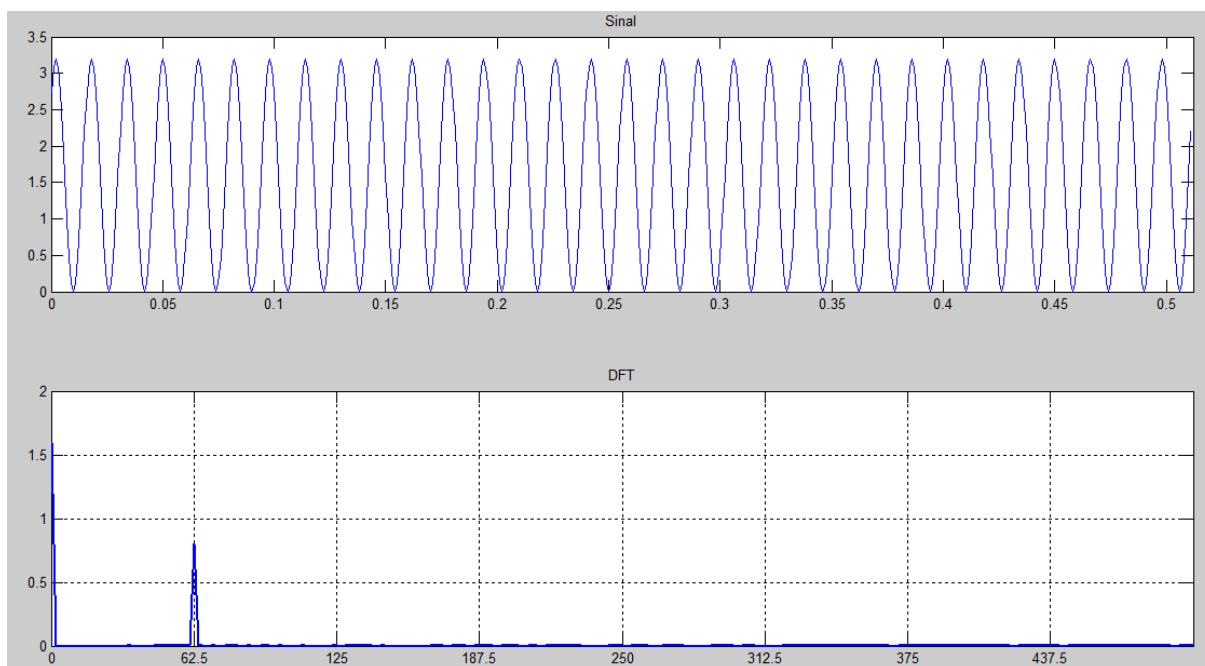


Figura 36 – Senoide no domínio do tempo e resultado da FFT no domínio da frequência.

Fonte: Autor.

Os resultados exibidos nas figuras 34, 35 e 36 foram amostrados com uma taxa de 1000 amostras por segundo, e armazenados em vetores de 512 posições, sendo que cada posição corresponde a um valor com ponto flutuante representado no formato de 64 bits. A figura 37 apresenta os resultados de um teste realizado com um sinal menos convencional, composto por frequências distintas entre si. Todos estes resultados verificaram-se corretos.

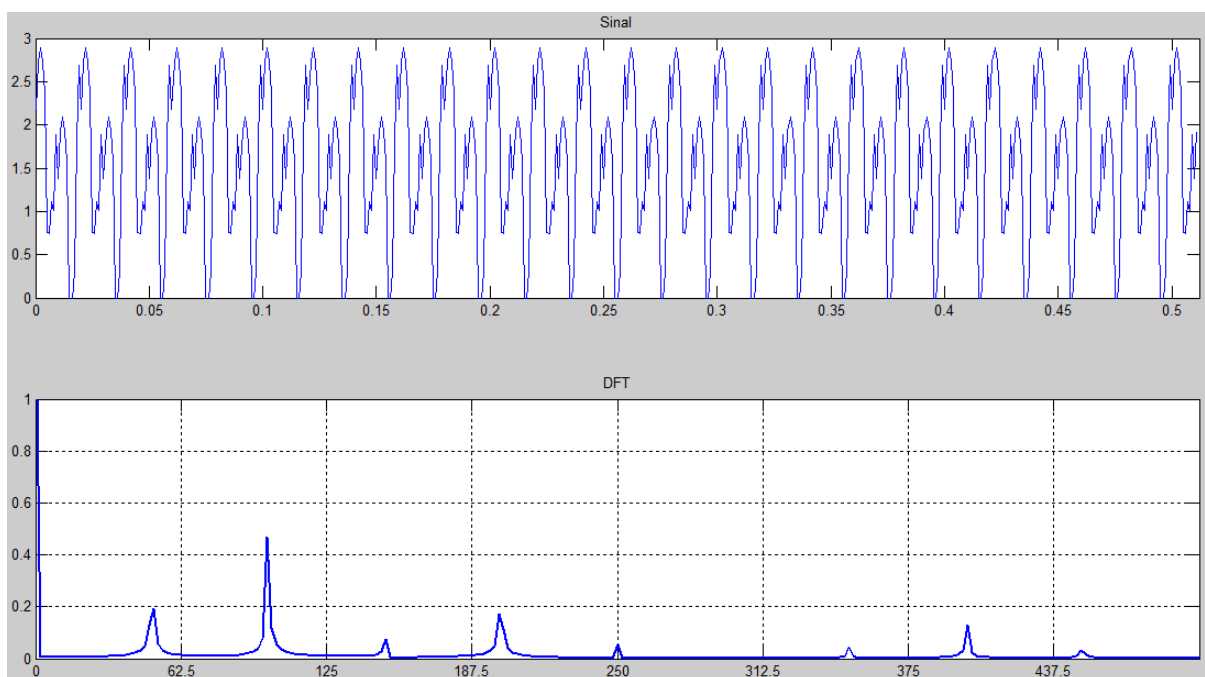


Figura 37 – Mistura de sinais no domínio do tempo e resultado da FFT no domínio da frequência.

Fonte: Autor.

3 ANALISADOR DE VIBRAÇÕES MECÂNICAS

A ferramenta aqui proposta consiste de um sistema microcontrolado extremamente versátil, que possa ser utilizada para análise de vibrações em meios contínuos ou em circuitos do tipo massa-mola-amortecedor, possuindo ainda, dois modos de funcionamento, para facilitar modelagens ou para uso em manutenção preditiva. Na figura 38 encontra-se o fluxograma geral de funcionamento desta ferramenta.

Os dois modos de funcionamento diferem entre si por fazer uso ou não da excitação provocada pela própria ferramenta. O primeiro modo utiliza apenas excitações externas para análise de sistemas ou estruturas. No segundo modo, o próprio sistema produz uma excitação, um impulso, em seguida gera o espectro de resposta a esta excitação, sendo útil para estudos, tais como identificar frequências naturais de vibração.

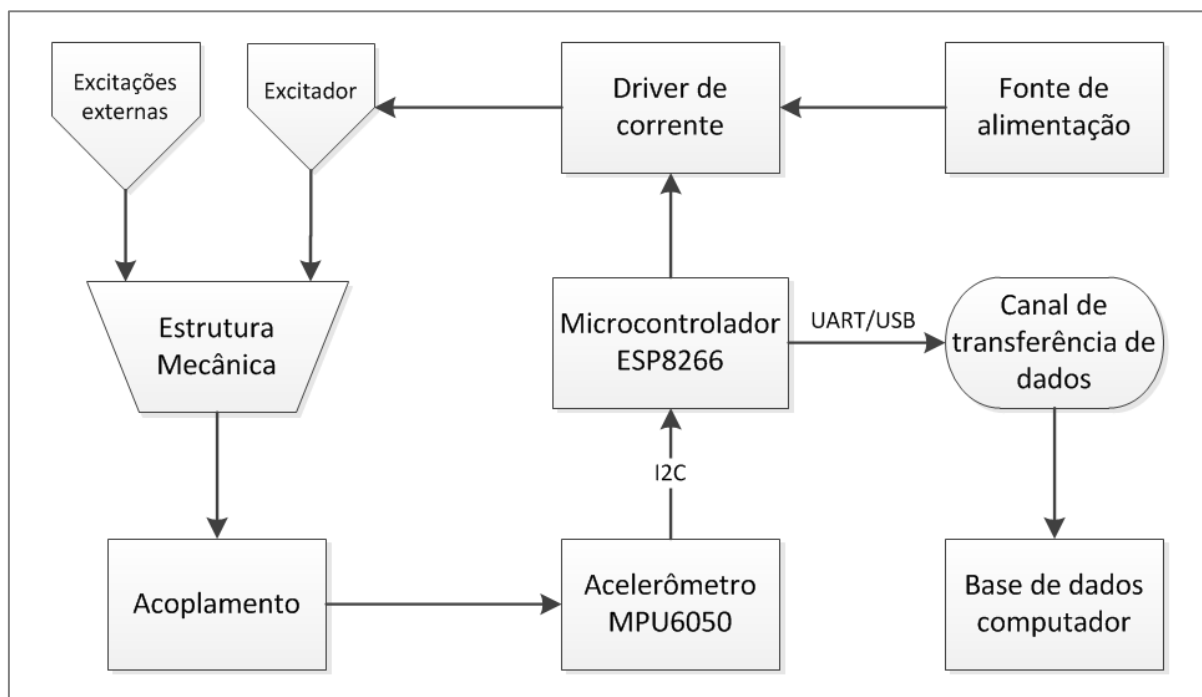


Figura 38 – Fluxograma do analisador de vibrações mecânicas acoplado a estrutura ou circuito mecânico.

Fonte: Autor.

No estudo de um circuito massa-mola como, por exemplo, um alto-falante, o excitador já está embutido na própria estrutura mecânica. Já em uma máquina qualquer, as excitações são provenientes de seus próprios elementos atuadores, neste caso o primeiro modo de funcionamento pode ser aplicado para estudos das vibrações identidades da estrutura ou para monitoramento e manutenção preditiva.

3.1 ACELERÔMETRO

Conforme visto no tópico 1.3.3, o acelerômetro é um sensor utilizado para medições de vibrações. Para esta aplicação foi selecionado o acelerômetro MPU-6050 produzido pela *InvenSense*. Este dispositivo é muito mais que um simples sensor, ele contém um acelerômetro de três eixos, um giroscópio de três eixos e mais alguns recursos avançados, integrados a um processador que codifica esses sinais e pode ser acessado através do protocolo de comunicação serial I2C⁶.

O diagrama interno geral da família MPU-60X0 é mostrado na figura 39. Nesta figura, os nomes de pinos entre parênteses são aplicáveis apenas ao MPU-6000, enquanto os nomes entre colchetes são aplicáveis apenas ao MPU-6050, o restante é válido para ambos.

Seguem as principais características retiradas do *datasheet* do acelerômetro MPU6050:

- *Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning*
- *Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning*
- *Digital Motion Processor (DMP) engine*
- *Auxiliary I2C serial interface for 3rd party magnetometer & other sensors*
- *Clocking*
- *Sensor Data Registers*
- *FIFO*
- *Interrupts*
- *Digital-Output Temperature Sensor*
- *Gyroscope & Accelerometer Self-test*
- *Bias and LDO*
- *Charge Pump*

⁶ I2C é um protocolo de comunicação serial utilizado em comunicações entre componentes digitais, geralmente dispostos na mesma placa de circuito impresso, ou ainda cabeados a pequenas distâncias. Este protocolo utiliza uma arquitetura simples de comunicação do tipo mestre-escravo, utilizando apenas dois canais para comunicação, sendo um de sincronização e o outro de dados.

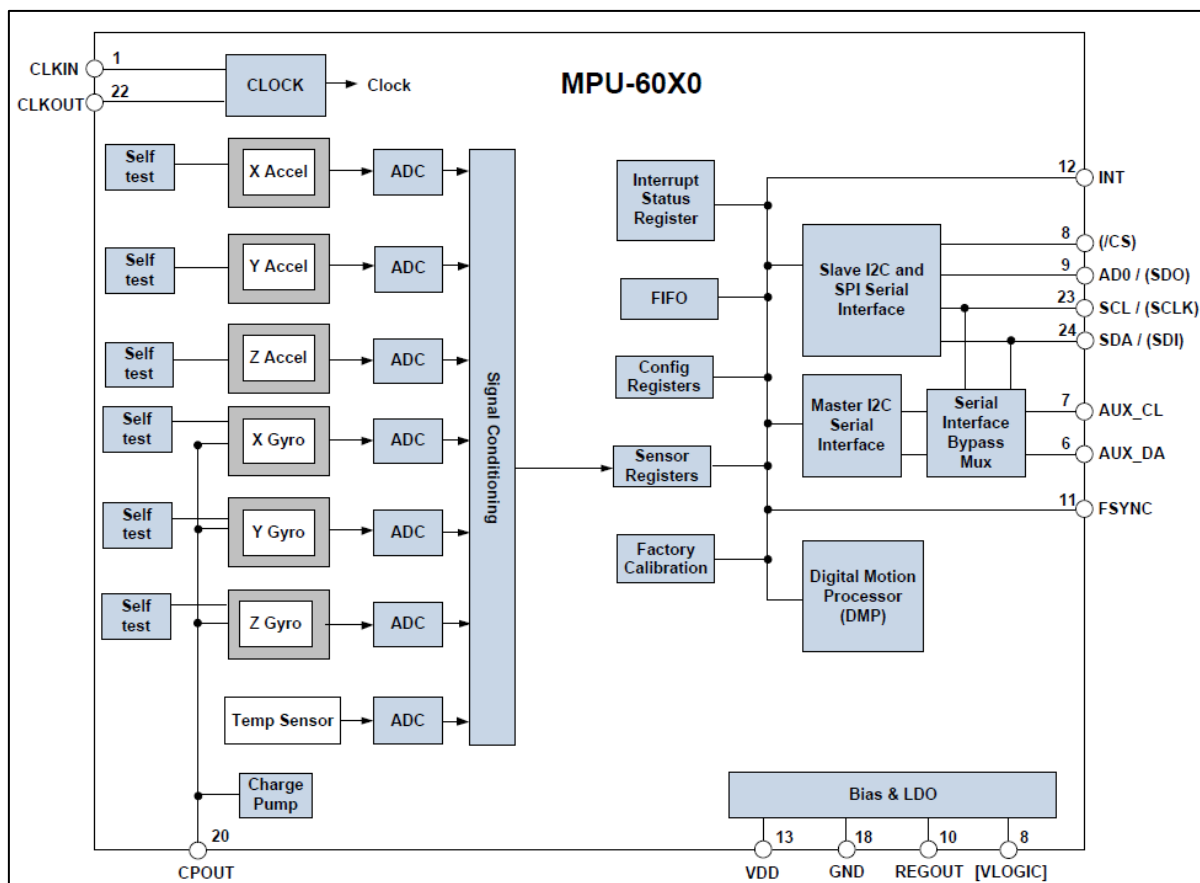


Figura 39 – Diagrama em blocos da família de sensores MPU-60X0.

Fonte: https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf

Especificamente, o acelerômetro de três eixos do tipo MEMS, que está incluso no MPU-6050, possui as seguintes características, também retiradas de seu *datasheet*:

- *Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$*
- *Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer*
- *Accelerometer normal operating current: $500\mu A$*
- *Low power accelerometer mode current: $10\mu A$ at $1.25Hz$, $20\mu A$ at $5Hz$, $60\mu A$ at $20Hz$, $110\mu A$ at $40Hz$*
- *Orientation detection and signaling*
- *Tap detection*
- *User-programmable interrupts*
- *High-G interrupt*
- *User self-test*

O circuito integrado MPU-6050 é fabricado em um formato fisicamente pequeno, sua soldagem deve ser feita na superfície da placa utilizando maquinário adequado. Por isso, soldá-lo manualmente em uma PCI é bastante difícil, sem os equipamentos adequados. Seguindo a filosofia de diversos outros componentes deste tipo, os fabricantes disponibilizam PCIs prontas com o circuito montado e com uma pinagem de acesso aos recursos do componente em questão; este é o caso do módulo MPU-6050, que foi utilizado neste trabalho. O módulo utilizado pode ser visto na figura 40.

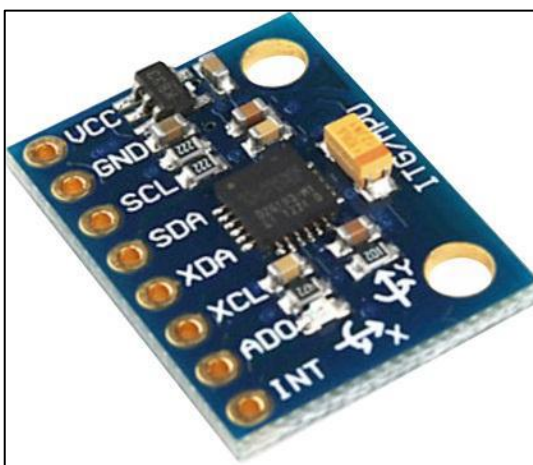


Figura 40 – Módulo PCI com MPU-6050.

Fonte: <https://trandi.wordpress.com/2016/04/24/e-spruino-mpu6050-digital-motion-processing/>.

3.2 HARDWARE UTILIZADO

O *hardware* utilizado é composto pelo módulo NodeMCU v2, o acelerômetro MPU-6050, um driver de corrente feito com transistores, um LED indicador de calibração e um botão para ativar a calibração. Na figura 41, temos o diagrama eletrônico das conexões do NodeMCU. Sendo que o MPU-6050 pode ser conectado através de um cabo *flat*, a fim de facilitar sua fixação sobre os elementos mecânicos para análise.

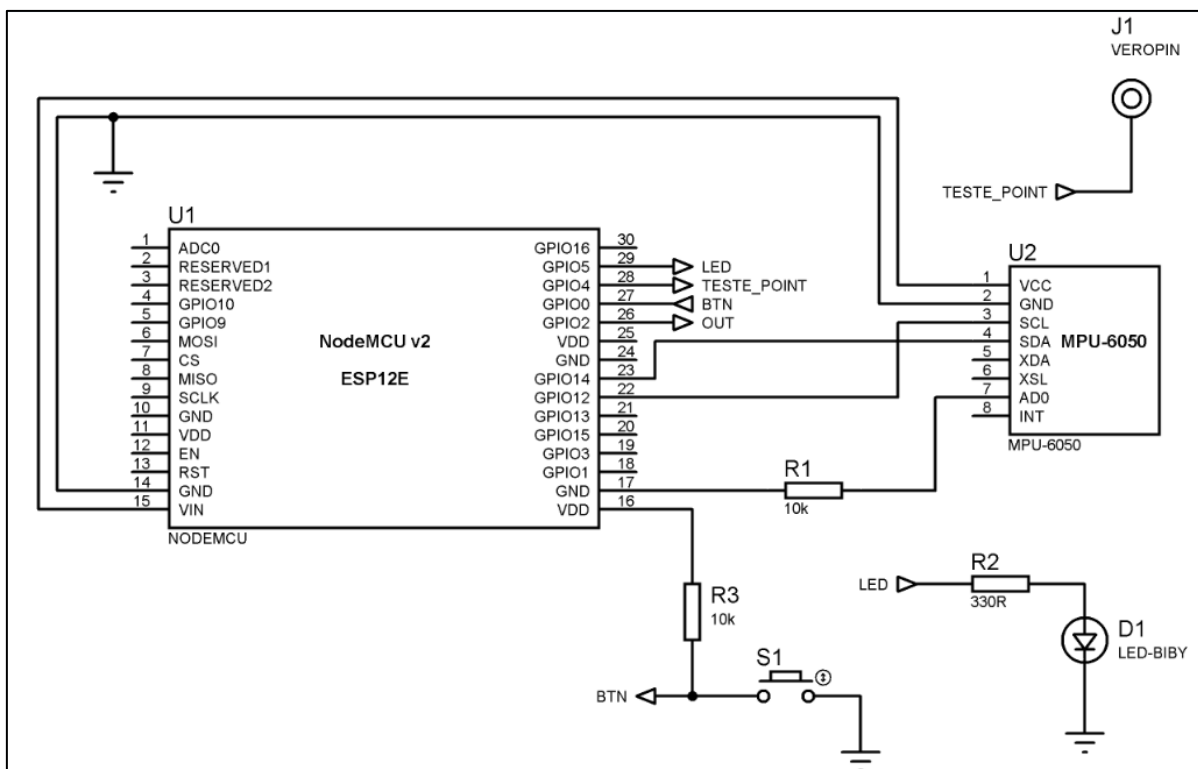


Figura 41 – Diagrama esquemático do circuito eletrônico do analisador de vibrações.

Fonte: Autor.

Na figura 42, tem-se o *driver* de corrente utilizado para gerar excitações em alto-falantes. A etiqueta OUT indica a conexão direta do microcontrolador à entrada do *driver*, passando por uma amplificação de tensão e corrente pelos transistores Q4 e Q3. O *jumper* JP1 na posição em que se encontra na figura 42, transfere o sinal amplificado para o driver de corrente que excita o alto-falante (a bobina do alto-falante é conectado no J3). Mas pode ser utilizado para ativar outros elementos de atuação. Para entrar com um sinal externo, por exemplo, de um gerador de função, muda-se a posição do JP1, e a saída de sinal do gerador deve ser conectada a J2. A fonte de alimentação do sistema deve ser uma fonte simétrica de ± 12 V. O diagrama elétrico da fonte utilizada encontra-se na figura 43.

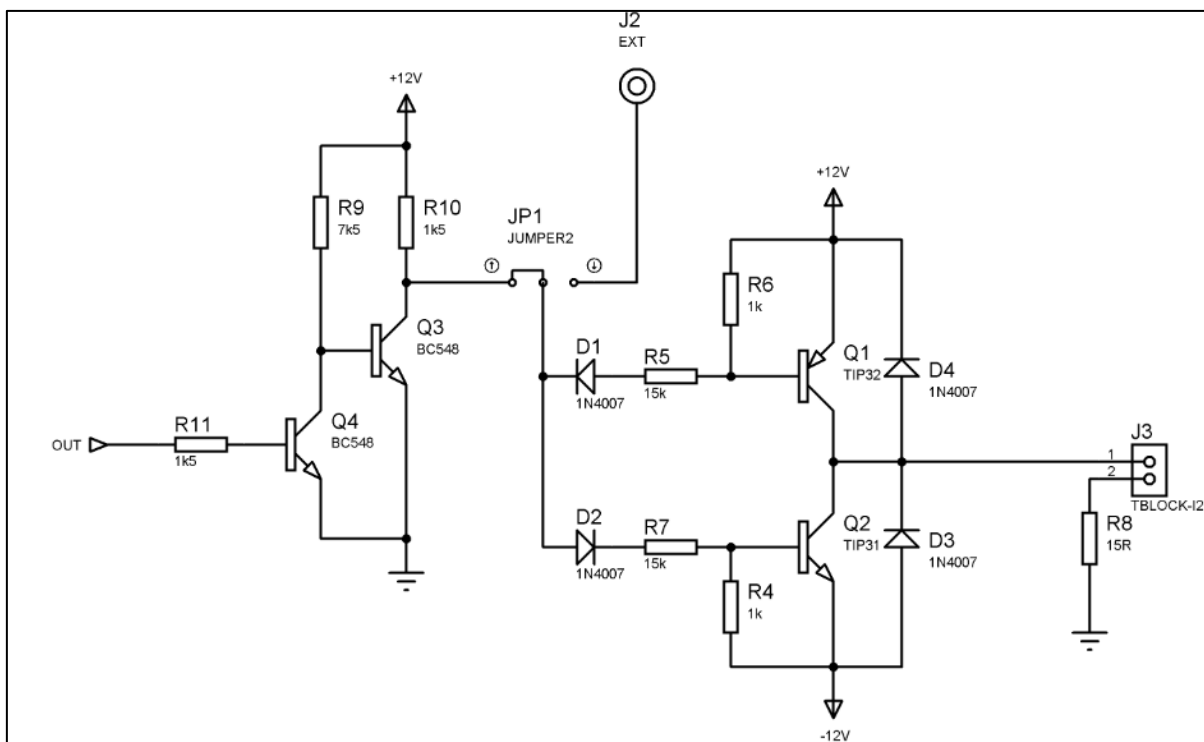


Figura 42 – Diagrama eletrônico do circuito do *driver* de corrente.

Fonte: Autor.

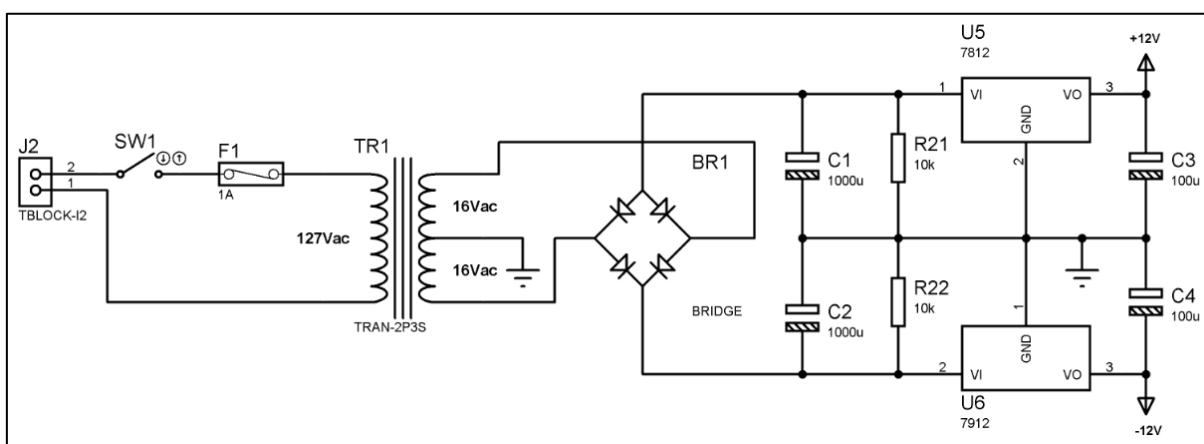


Figura 43 – Diagrama eletrônico da fonte de alimentação simétrica.

Fonte: Autor.

O *hardware* aqui proposto foi montado em *protoboard* para realização dos experimentos, embora uma PCI (placa de circuito impresso) possa ser facilmente fabricada para aplicações futuras.

3.3 PLATAFORMAS EXPERIMENTAIS

São apresentados, neste trabalho, dois modos de funcionamento do dispositivo de análise de vibrações, o que exige duas plataformas experimentais, apresentadas a seguir. A primeira é uma estrutura mecânica provida com atuadores; a segunda, consiste num alto-falante modelado com um sistema massa-mola amortecido.

3.3.1 Estrutura mecânica com atuadores

Esta plataforma experimental consiste de uma estrutura mecânica com alguns atuadores que serviram de fontes de vibrações. A estrutura foi montada em perfis de alumínio do fabricante Eng. Legho Systems⁷. Estes perfis são de fácil montagem, possuem rasgos para encaixe e fixação simples com parafusos. Tudo isso contribui para a construção de uma plataforma simples, porém, flexível, permitindo alterar suas configurações mecânicas para experimentos diversos.

Os atuadores utilizados para gerar vibrações são motores de corrente contínua (CC). Foram utilizados três motores de 12 W com tensão nominal de 3 V. Para fazer o controle de velocidade destes motores foram utilizadas três fontes ajustáveis de tensão CC. Assim, variando-se a tensão dos motores, teremos velocidades de rotação diferentes e conseqüentemente, vibrações diferentes impostas à estrutura física. Além disso, os motores utilizados não são novos, já possuem algum desgaste, por isso se espera algumas diferenças consideráveis nas vibrações de cada motor.

Na figura 44, temos as peças principais utilizadas, estas peças servirão para compor uma estrutura que foi selecionada aleatoriamente, com o único propósito de simular uma máquina qualquer, para isso adotou-se uma estrutura composta por uma base e três hastes iguais, onde estão fixos os motores e ainda uma quarta haste onde é fixo o acelerômetro.

Na figura 45 temos a estrutura montada na configuração em que foi utilizada durante os experimentos.

⁷ < <http://www.engelegho.com.br/> >

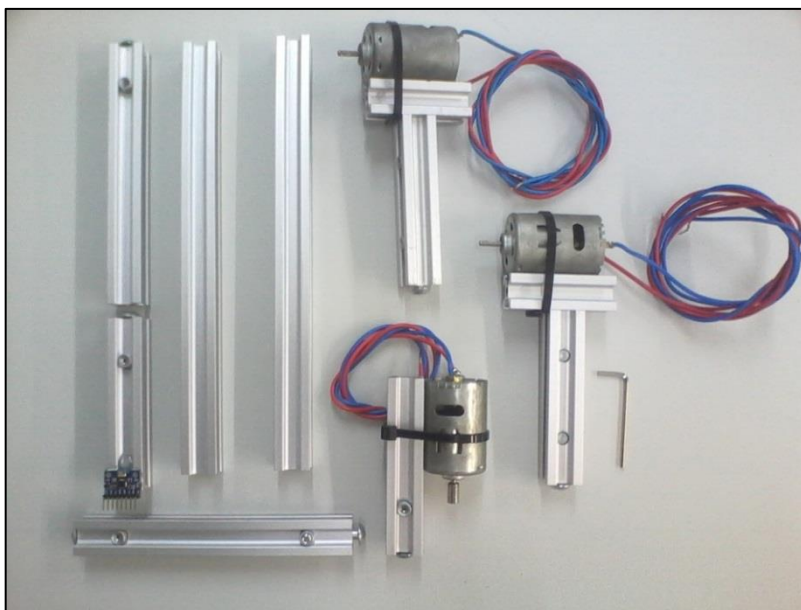


Figura 44 – Peças para montagem de estrutura mecânica analisador de vibrações.
Fonte: Autor.

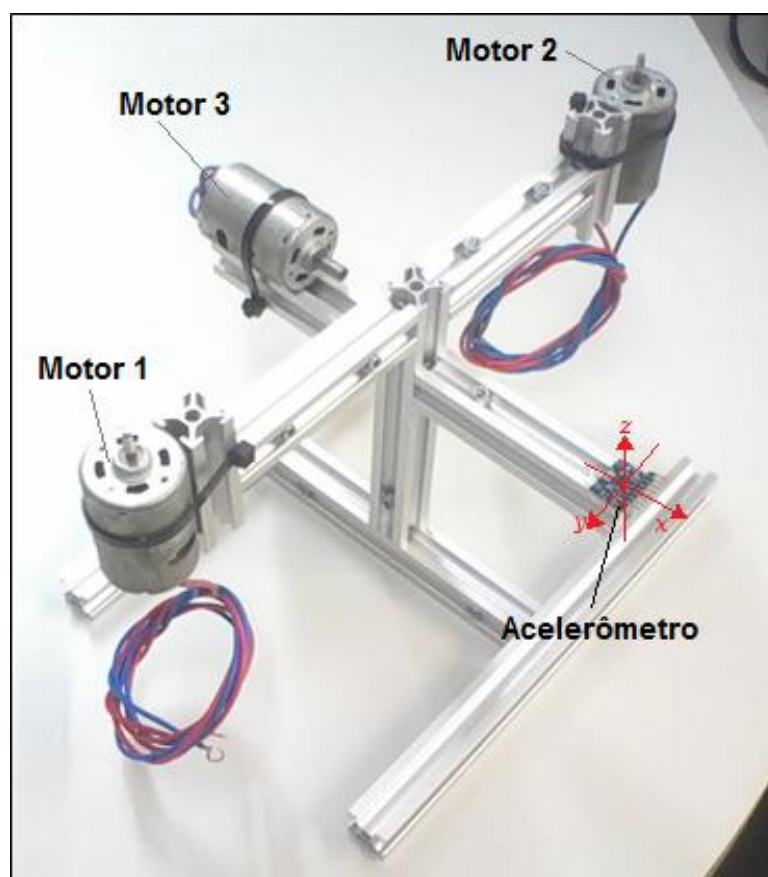


Figura 45 – Montagem da estrutura mecânica montada.
Fonte: Autor.

3.3.2 Sistema massa-mola com amortecimento

Para a análise de um sistema massa-mola-amortecedor foi utilizado um alto-falante do tipo *subwoofer* de $4\ \Omega$ de impedância e 35 W de potência RMS (*Root Mean Square*). Na figura 46, é apresentada a estrutura do alto-falante com as adaptações feitas para a fixação do acelerômetro em seu domo. A seta com a indicação z mostra a orientação do acelerômetro, sendo z o eixo do domo. Os eixos x e y, não serão considerados no experimento do massa-mola-amortecedor. Embora existam informações nestes eixos, em prol da simplicidade analisou-se apenas o eixo principal.

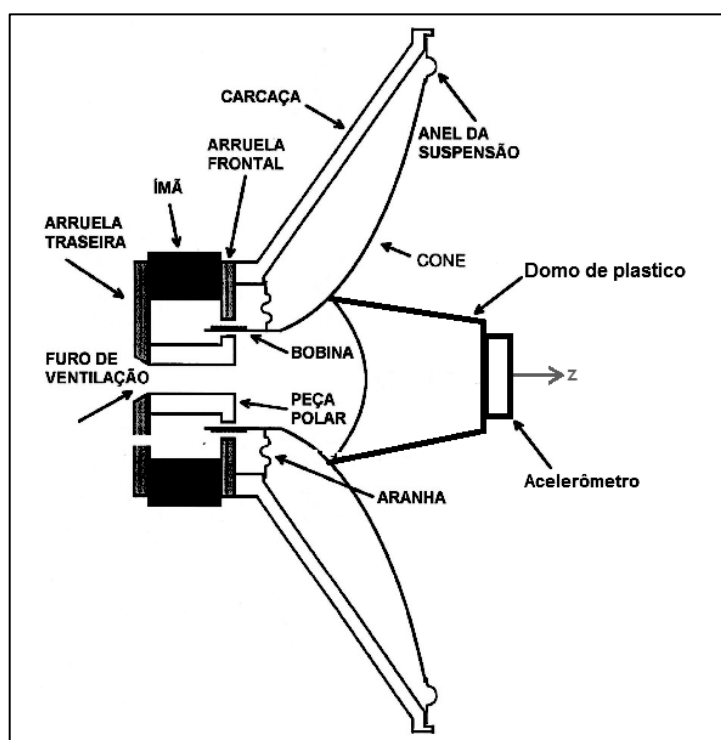


Figura 46 – Vista lateral esquemático do alto-falante com o acelerômetro.
Fonte: Autor.

Na figura 47, é apresentada a imagem do alto-falante utilizado durante os experimentos, montado com o acelerômetro MPU6050 e conectado com um *flat cable* ao módulo NodeMCU v2.

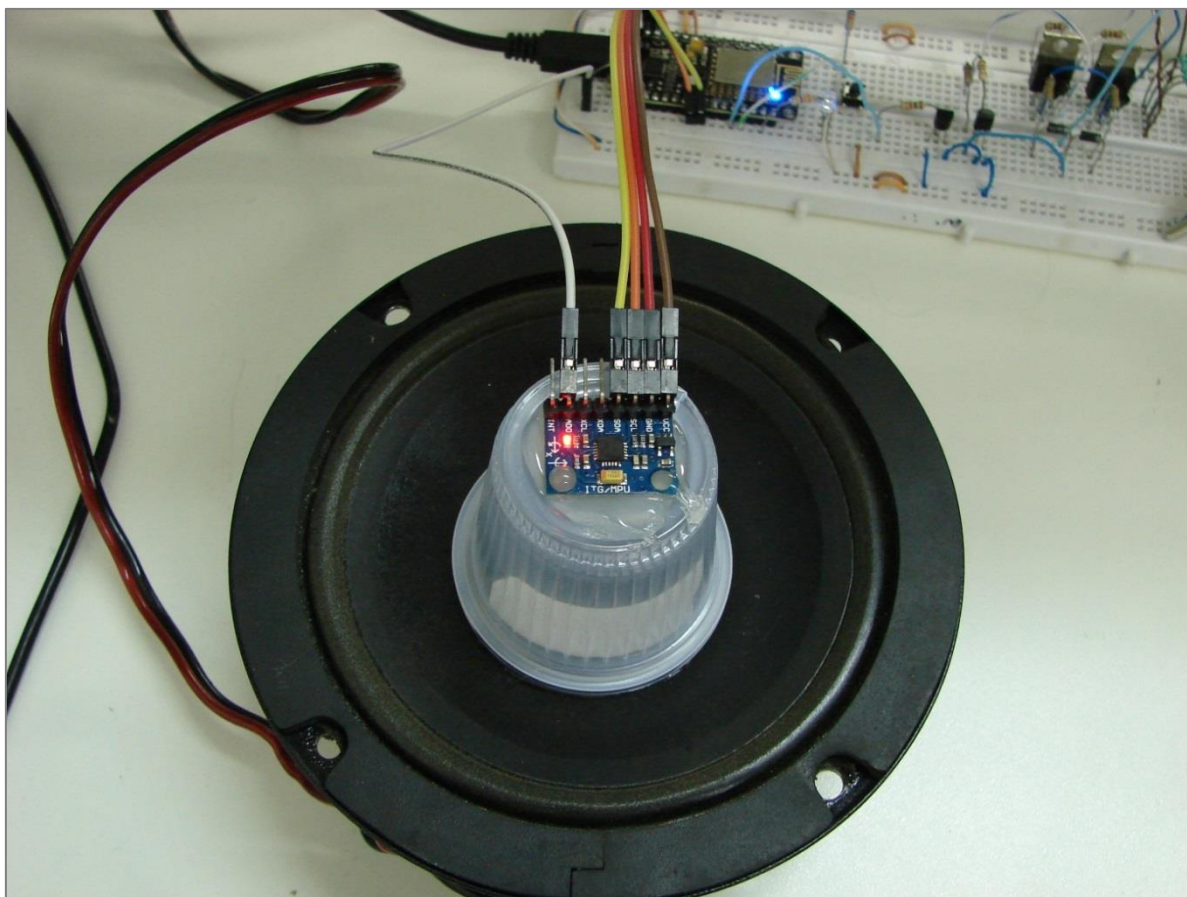


Figura 47 – Foto da vista superior do alto-falante com o acelerômetro no domo improvisado.

Fonte: Autor.

3.4 CONSIDERAÇÕES DE *FIRMWARE*

Uma visão geral do *firmware* é feita através do fluxograma da figura 48. Neste tópico, são apresentados os principais aspectos dos elementos do *firmware* exceto pelas configurações iniciais, pois estas são triviais e podem ser vistas no código do programa disponível no *Github*. No início do *looping* principal, faz-se a leitura dos dados do sensor acelerômetro, através do protocolo I2C. Após a leitura, é realizado o decremento de *offset*, para desconsiderar elementos contínuos já presentes no sistema, consequências da gravidade e desalinhamentos no sensor. Para atingir melhores resultados espectrais, também foi acrescentado um filtro janela de Hamming. Após o janelamento do sinal, a FHT é computada e deste resultado é obtida a FFT, utilizando o algoritmo mais eficiente da comparação realizada no capítulo 2. No final da operação, os resultados da FFT e o vetor amostrado são enviados para o computador.

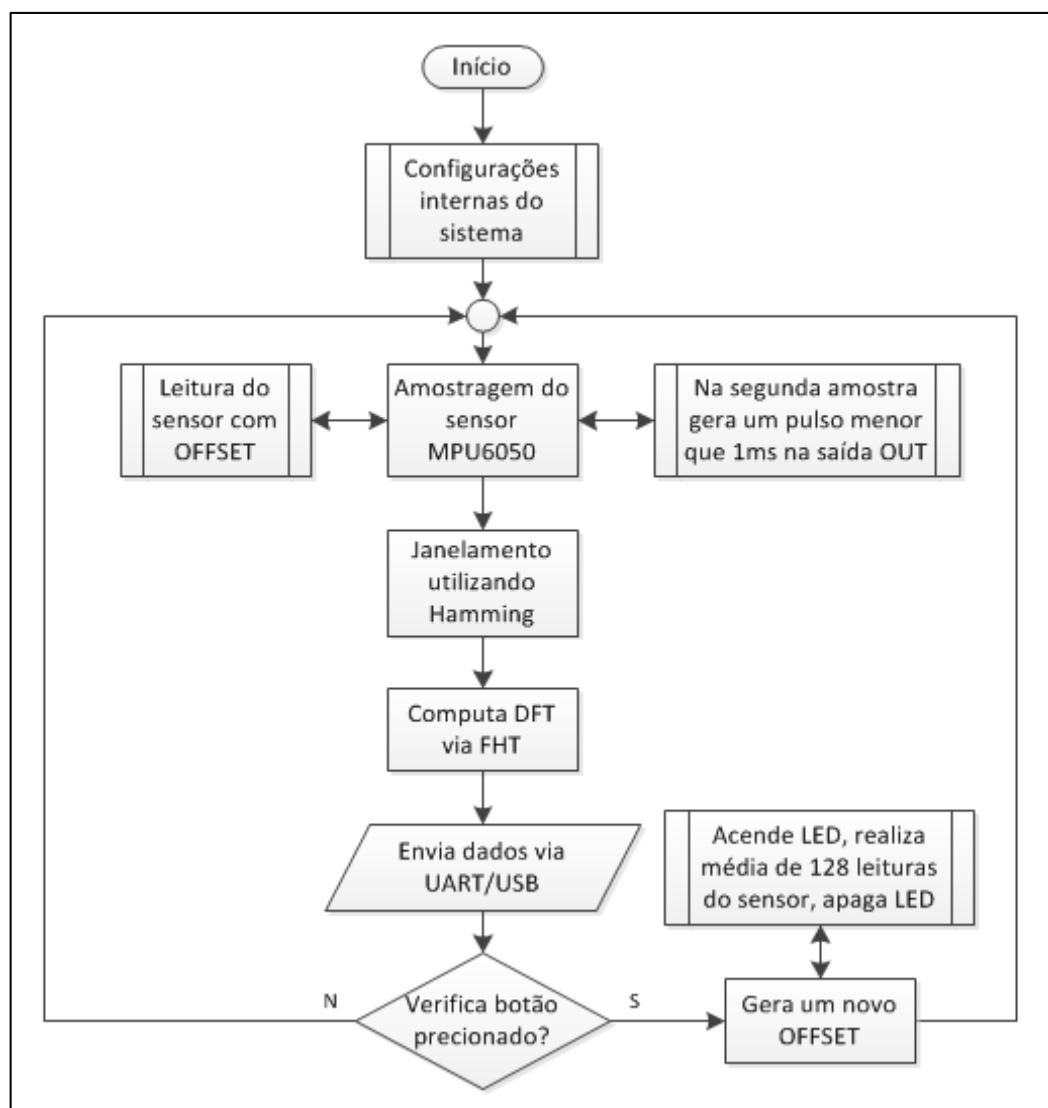


Figura 48 – Fluxograma geral do *firmware* utilizado.

Fonte: Autor.

3.4.1 Leitura do sensor

Conforme já mencionado no tópico 3.1, o acelerômetro utilizado é o modelo MPU-6050, o qual utiliza o protocolo I2C para entregar os dados de leitura ou ser configurado. O Arduino possui diversas bibliotecas prontas para protocolos de comunicação; neste trabalho, foi utilizada a biblioteca *wire.h*. O função de leitura dos dados do acelerômetro pode ser vista na figura 49, juntamente com a função que gera o *Offset* de cada eixo. Conforme pode ser visto nesse trecho de código, o recebimento de dados seriais do protocolo é utilizado através da classe instanciada com o nome *Wire*. Além disso, a função *GetAcel()* faz as conversões necessárias para a montagem dos dados recebidos, pois os bytes são recebidos separados e precisam ser unificados em uma mesma variável. No caso da rotina de

offset temos ainda implementada uma média aritmética para 128 amostras. Na figura 50, apresenta-se o trecho do código em que é feita amostragem do sinal para processamento.

```

1 // ****
2 // Função de Leitura dos dados do acelerômetro
3 // ****
4 signed int GetAcel(){
5     int16_t B;
6     uint8_t H,L;
7     H=Wire.read();
8     L=Wire.read();
9     B=(H<<8)+L;
10    return B;
11 }
12 // ****
13 // Função Offset
14 // ****
15 void MPU_Offset(){
16     signed long AcumX=0, AcumY=0, AcumZ=0;
17     unsigned short i;
18
19     for (i=0;i<128;i++){
20         Wire.beginTransmission(MPU); // Inicia comunicação com MPU
21         Wire.write(0x3B);           // Começa leitura pelo endereço 0x3B (ACCEL_XOUT_H)
22         Wire.endTransmission(false);
23         Wire.requestFrom(MPU,6,1); // Faz requisição para leitura de 6 registros
24         AcumX+=GetAcel();
25         AcumY+=GetAcel();
26         AcumZ+=GetAcel();
27         delay(3);
28     }
29     OfX=AcumX/128;
30     OfY=AcumY/128;
31     OfZ=AcumZ/128;
32 }

```

Figura 49 – Trecho de código de leitura dos dados individuais e *offset* do MPU-6040.

Fonte: Autor.

Nas linhas 20 e 21 do código apresentado na figura 50, pode-se ver o trecho do código responsável pela geração de um pulso de curta duração, logo no início da amostragem, entre a segunda e terceira amostras.

```

1 // ****
2 // Função que faz a amostragem
3 // ****
4 void Sample(){
5     unsigned short i;
6     for (i=0;i<N;i++){
7         if (i%16==0) yield();
8         digitalWrite(Teste, LOW); // Ponto de teste
9         Wire.beginTransaction(MPU); // Inicia comunicação com MPU
10        Wire.write(0x3B); // Começa leitura pelo endereço 0x3B (ACCEL_XOUT_H)
11        Wire.endTransmission(false);
12        Wire.requestFrom(MPU,6,1); // Faz requisição para leitura de 6 registros
13        digitalWrite(Teste, HIGH);
14        AcX=GetAcel(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
15        AcY=GetAcel(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
16        AcZ=GetAcel(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
17        v[X][i]=(double)(AcX-OfX);
18        v[Y][i]=(double)(AcY-OfY);
19        v[Z][i]=(double)(AcZ-OfZ);
20        if (i==1) digitalWrite(Output,HIGH); // Gera um pulso < 1ms na saída OUT
21        else digitalWrite(Output,LOW);
22    }
23    for (i=0;i<N;i++){
24        v[X][i]=v[X][i]*Fator_Conv; // 10 ~ 9.8 m/s^2
25        v[Y][i]=v[Y][i]*Fator_Conv;
26        v[Z][i]=v[Z][i]*Fator_Conv;
27    }
28 }

```

Figura 50 – Trecho de código de leitura dos dados do acelerômetro MPU-6040.

Fonte: Autor.

3.4.2 Janelamento, janela de Hamming e FHT

Na figura 51, está evidenciada em azul a linha do código que computa a janela de Hamming utilizando a fórmula (67). Empregou-se o algoritmo desenvolvido no capítulo 2 para computar a DFT via FHT, no qual foi inserida multiplicação da janela de Hamming sobre o sinal amostrado. Para economizar memória, utilizou-se o vetor auxiliar da FHT para receber o sinal janelado, pois o sinal original é conservado e enviado na íntegra para o computador, para eventuais verificações.

```

1 void FHT(unsigned short E){
2     double a, b, c;
3     unsigned int i, k, L, desl, P, NP, aux1, aux2, aux3;
4     yield();
5     // Janela de Hamming
6     for (i=0;i<N;i++) Hartley[i]=(0.54-0.46*cos(i*2*pi/N))*v[E][i];
7     yield();
8     // Carrega vetores
9     for (i=0;i<N;i++) Fourier[Table_Reverse(i,N)]=Hartley[i];
10    // Carrega vetor de coeficientes
11    k=512/N;
12    for (i=0;i<N/4;i++){
13        if (!i%8) yield();
14        aux1=i*k;
15        C[i] = tab_trig[aux1];
16        C[i+N/4]= -tab_trig[128-aux1];
17        S[i] = tab_trig[128-aux1];
18        S[i+N/4]= tab_trig[aux1];
19    }
20    // Computa FHT Cooley-Tukey radix-2

```

Figura 51 – Trecho do programa que computa FHT utilizando a janela de Hamming.

Fonte: Autor.

3.4.3 Envio de dados

Os dados de leitura são armazenados em matrizes 3x512, enquanto que suas transformadas são armazenadas em matrizes 3x256. Toda essa informação é montada em um pacote de dados e enviado via UART/USB para o computador.

Foi desenvolvida uma interface para plotar os gráficos referentes a estes dados enviados pelo microcontrolador, ou ainda, salvá-los em planilhas. A figura 52 mostra a tela principal onde são impressos do lado esquerdo os sinais provenientes do acelerômetro, sendo respectivamente de cima para baixo, os eixos x, y e z, enquanto à direita estão seus gráficos no domínio da frequência. Para os gráficos no domínio do tempo a unidade do eixo das ordenadas é mm/s², enquanto que o eixo das abcissas é segundo (s); para os espectros de frequência a unidade no eixo das ordenadas também é mm/s² e no eixo das abcissas tem-se a frequência em hertz (Hz).

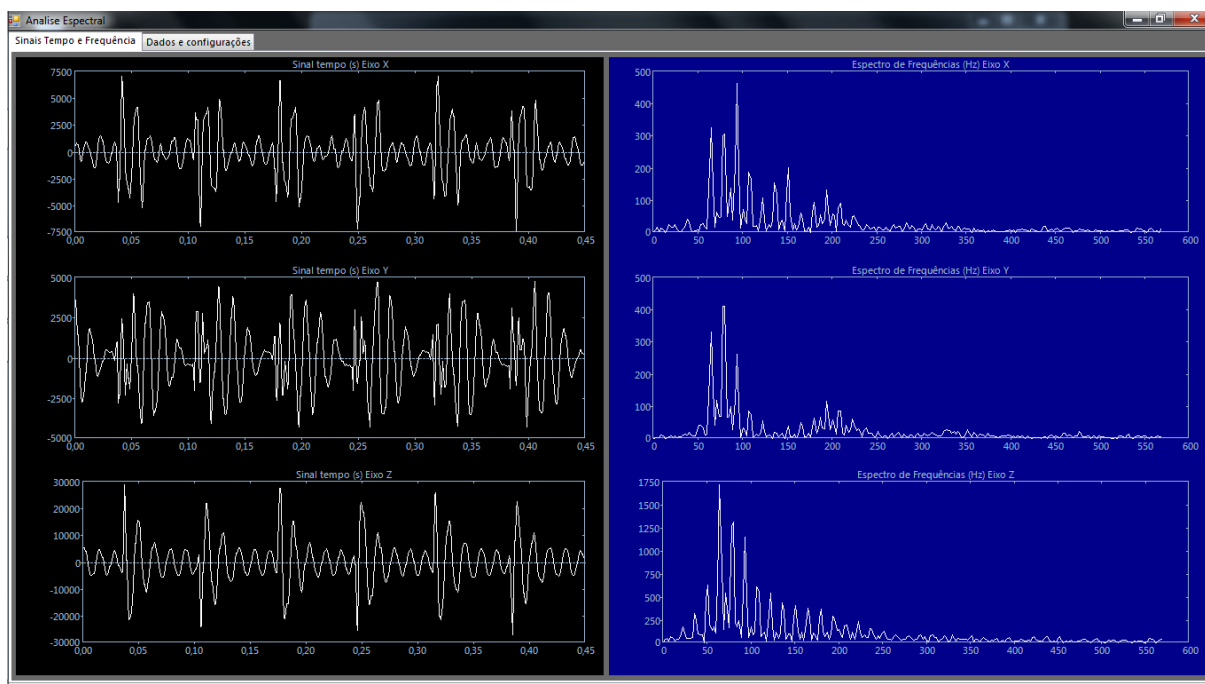


Figura 52 – Interface analisador de vibrações mecânicas.

Fonte: Print screen da tela da interface durante execução.

Na segunda aba da interface, também é possível alterar seus parâmetros de montagem dos gráficos, salvar os dados em planilhas e ainda configurar o funcionamento da comunicação com o microcontrolador.

3.5 RESULTADOS EXPERIMENTAIS

3.5.1 Análise de vibração em estruturas mecânicas

Para exemplificar o procedimento de análise de vibrações em estruturas mecânicas, foi utilizada a plataforma do tópico 3.3.1, empregando a configuração ilustrada na figura 45. Após todo o sistema estar conectado e pronto para os experimentos, foi feito o *Offset* do acelerômetro, que deve ser realizado sempre no início de novos experimentos, garantindo que os referenciais estejam em zero.

Ativando os motores 1 e 2, fazendo o uso de fontes ajustáveis, a fim de simular o funcionamento adequado de um equipamento qualquer, foi estabelecido que estes dois motores devem funcionar com 2 V. Para esta tensão, o motor 1 consome 2,3 A, enquanto o motor 2 consome 1,9 A. Sendo assim, o espectro para o equipamento funcionando com esta configuração é dito “espectro identidade da máquina”.

Na figura 53, é apresentado o espectro identidade para as configurações estabelecidas.

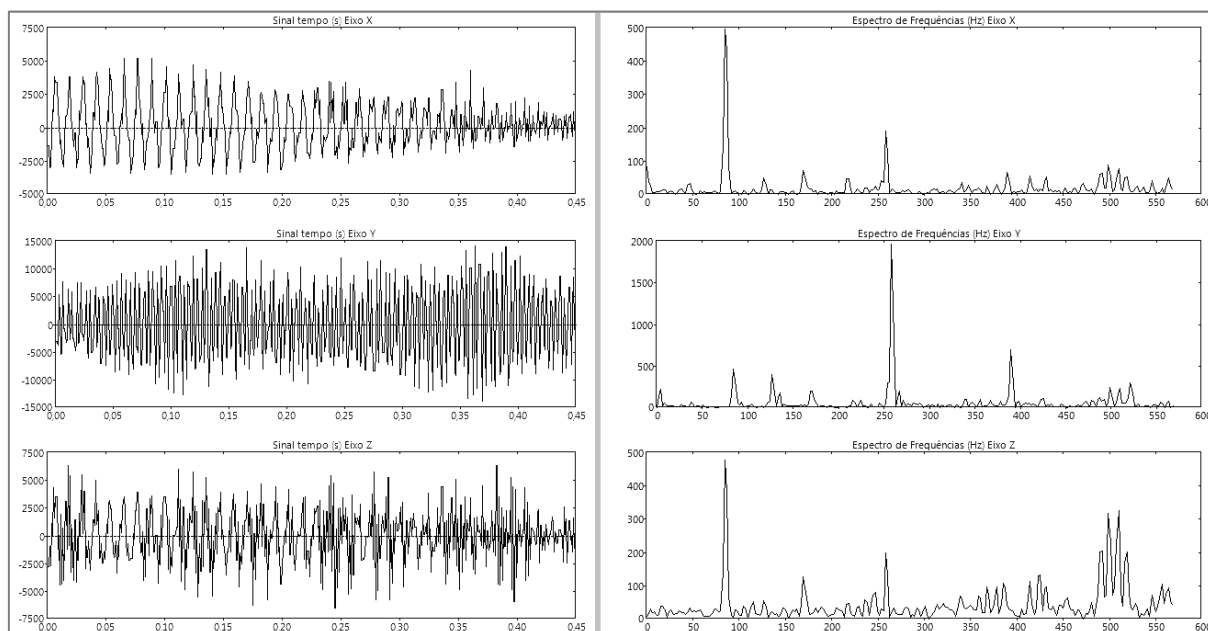


Figura 53 – Espectro identidade do equipamento em funcionamento normal.

Fonte: Autor.

Após algum tempo funcionando, pode haver pequenas variações, mas este espectro se mantém enquanto o equipamento funciona com os parâmetros supracitados. No espectro da figura 53, observa-se que existem duas frequências que se sobressaem às outras, sendo uma em torno de 90 Hz e a outra nos 260 Hz.

Para simular um defeito no funcionamento do equipamento, aplicou-se uma pequena queda de tensão no motor 1, baixando seu valor para 1,6 V. Na figura 54 encontra-se o resultado dos sinais amostrados e seus respectivos espectros.

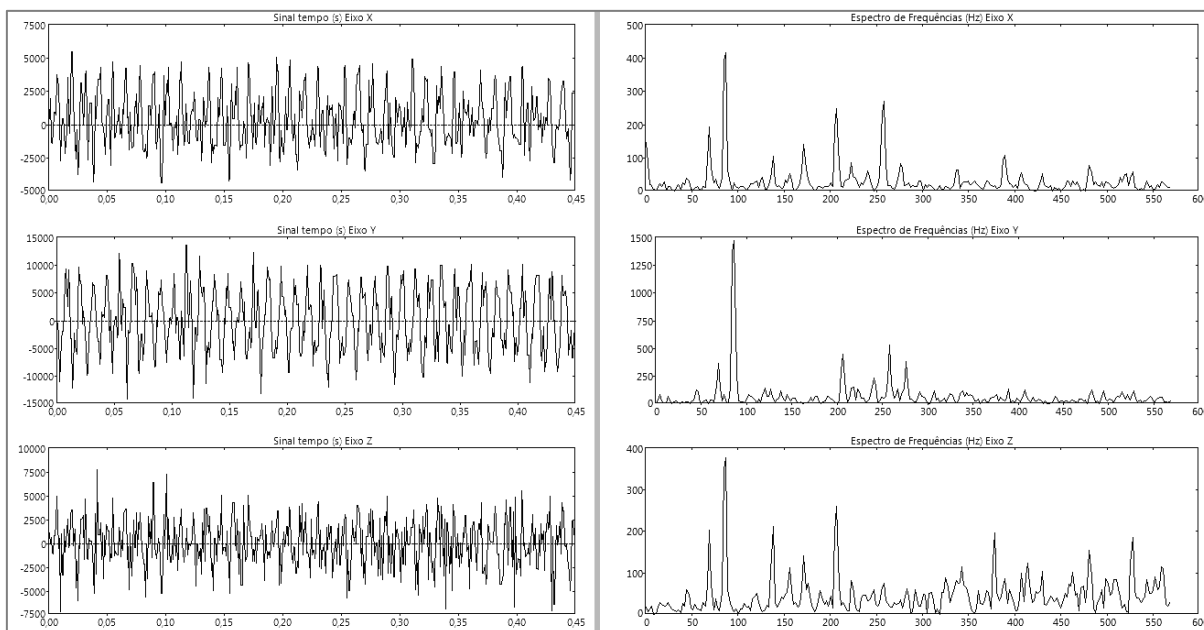


Figura 54 – Espectro para quando houve uma queda de tensão no motor 1.

Fonte: Autor.

Observando o sinal no domínio do tempo nas figuras 53 e 54, não se evidencia, visualmente, nenhuma informação valiosa sobre o funcionamento da máquina, porém, ao observar o espectro de frequências é bastante evidente que algo aconteceu, pois algumas frequências do espectro identidade desapareceram completamente e outras novas surgiram.

Foge do escopo deste trabalho fazer uma análise detalhada destes resultados, porém a ferramenta aqui desenvolvida pode ser modificada para uso em aplicações de manutenção preditiva, uma vez registrado os padrões espectrais de um equipamento em funcionamento adequado. O próprio microcontrolador pode estar comparando estes valores armazenados e emitir um aviso, caso haja uma variação significativa. Com um pouco mais de estudo pode ser possível até mesmo apontar qual o problema, utilizando como base o espectro de resposta.

Na figura 55, é possível observar o resultado de uma medição realizada com todos os motores desligados. Nesta condição, foi aplicado um pequeno impulso no sentido do eixo z sobre o motor 3. Ao aplicarmos um impulso em um sistema podemos verificar o aparecimento de suas frequências naturais de vibração no espectro.

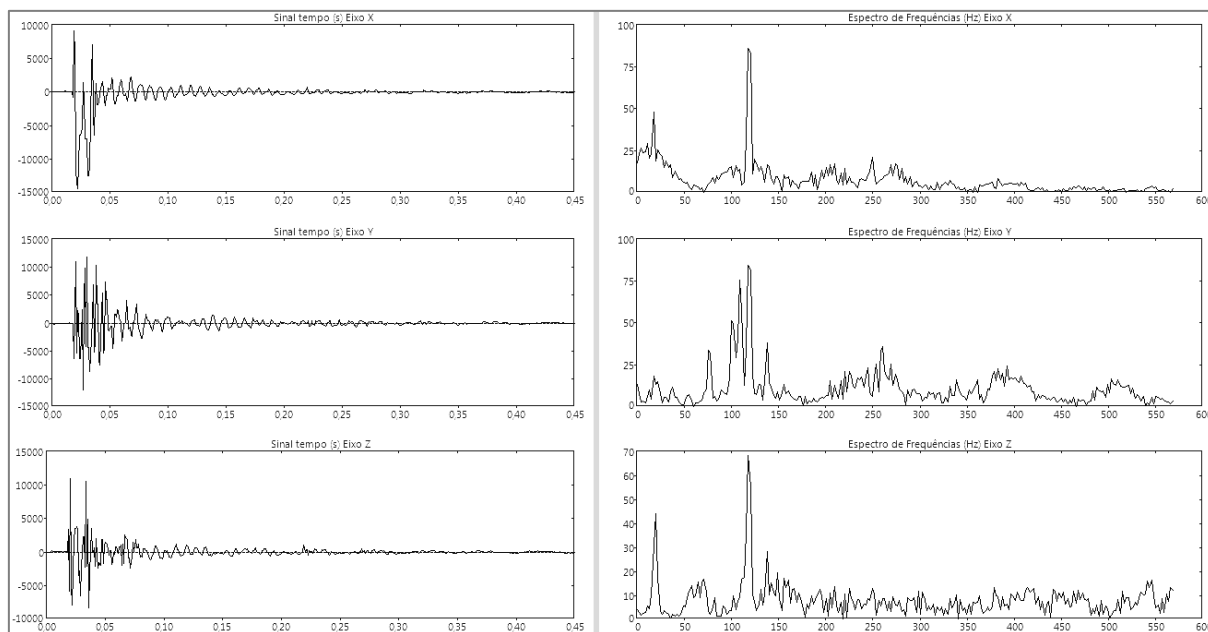


Figura 55 – Espectro durante impulso no motor 3 no sentido de z.

Fonte: Autor.

Na figura 56, tem-se o resultado para um impulso aplicado sobre o motor 3, porém no sentido do eixo x. Na figura 57, tem-se o resultado para um impulso aplicado sobre o motor 3 no sentido do eixo y. Este experimento foi realizado manualmente, ou seja, foram aplicados sucessivos golpes com um martelo adequado e registraram-se os resultados, dentre estes, selecionou-se aquele em que o gráfico no domínio do tempo apresentou maior semelhança ao segundo modo de funcionamento, onde um impulso de excitação ocorre próximo ao início da amostragem.

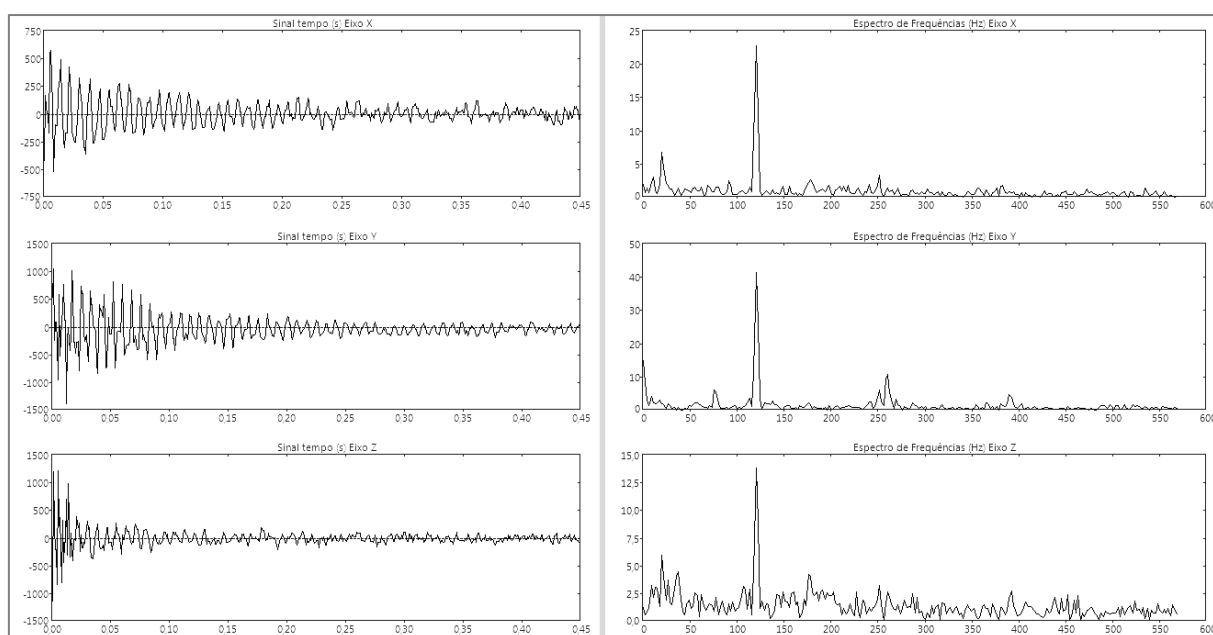


Figura 56 – Espectro durante impulso no motor 3 no sentido de x.

Fonte: Autor.

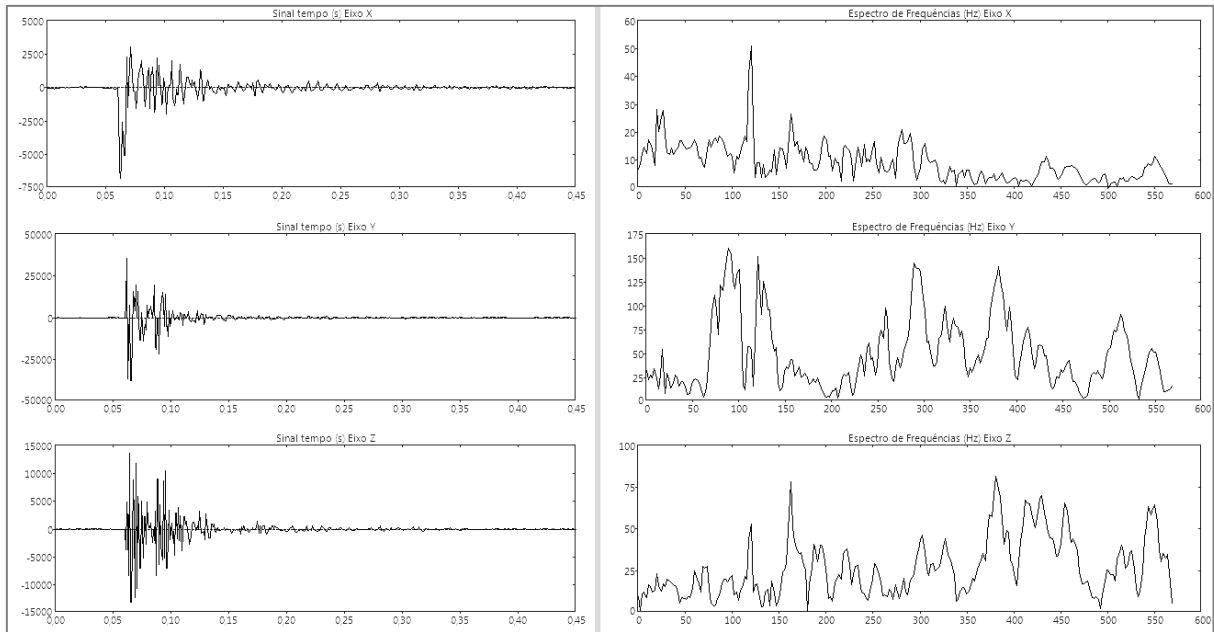


Figura 57 – Espectro durante impulso no motor 3 no sentido de y.

Fonte: Autor.

Os resultados apresentados nas figuras 55, 56 e 57 permitem supor que a frequência de ressonância desta estrutura esteja em torno dos 118 Hz, porém um estudo mais detalhado deve ser realizado para verificar este valor.

3.5.2 Análise de vibração em sistema massa-mola-amortecedor

Para a análise de vibrações em um sistema massa-mola-amortecedor foi utilizada a plataforma do tópico 3.3.2, empregando a configuração mostrada na figura 46. Utilizou-se, também, o circuito amplificador de corrente apresentado na figura 42 para gerar as excitações no alto-falante.

Conforme visto no tópico 1.3, o circuito massa-mola-amortecedor é modelado pela seguinte equação diferencial:

$$m \frac{d^2 z}{dt^2} + \beta \frac{dz}{dt} + kz = f(t) \quad (151)$$

Onde a variável z foi escolhida para deslocamento devido à montagem do acelerômetro, conforme apresentado na figura 45.

Na equação (151), $f(t)$ é a excitação externa que está sendo aplicada ao sistema. Neste tópico consideraram-se as duas situações a seguir:

$$1 - f(t) = \delta(t)$$

$$2 - f(t) = A \cdot \sin(\omega t)$$

Além das soluções apresentadas no tópico 1.3.1, durante os experimentos, utilizou-se o teorema para transformada de Fourier da derivada, para determinar qual o espectro de resposta era esperado teoricamente.

O primeiro caso, considerando que $f(t) = \delta(t)$, substituindo em (151), e aplicando a transformada de Fourier aos dois lados da equação, tem-se que:

$$(-m\omega^2 + i\beta\omega + k)Z(\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-i\omega t} dt = 1$$

Por tanto,

$$Z(\omega) = \frac{1}{(-m\omega^2 + i\beta\omega + k)} = \frac{1}{(k - m\omega^2) + (\beta\omega)i} = \frac{(k - m\omega^2) - (\beta\omega)i}{(k - m\omega^2)^2 + (\beta\omega)^2}$$

Logo,

$$|Z(\omega)| = \frac{\sqrt{(k - m\omega^2)^2 + (\beta\omega)^2}}{(k - m\omega^2)^2 + (\beta\omega)^2} \quad (152)$$

No Apêndice C, temos uma verificação de (152) partindo do resultado expresso em (108).

Para o segundo caso, em que $f(t) = A\sin(\varphi t)$, substituindo em (151), e aplicando a transformada de Fourier, tem-se que:

$$(-m\omega^2 + i\beta\omega + k)Z(\omega) = -i\pi A[\delta(\omega - \varphi) - \delta(\omega + \varphi)]$$

Portanto,

$$Z(\omega) = \frac{\pi A[\delta(\omega - \varphi) - \delta(\omega + \varphi)]}{(-im\omega^2 - \beta\omega + ik)} = \frac{\pi A[\delta(\omega - \varphi) - \delta(\omega + \varphi)]}{(-\beta\omega) + (k - m\omega^2)i}$$

Logo,

$$\begin{aligned} |Z(\omega)| &= \frac{\sqrt{\pi A[\delta(\omega - \varphi) - \delta(\omega + \varphi)][(-\beta\omega)^2 + (k - m\omega^2)^2]}}{(\beta\omega)^2 + (k - m\omega^2)^2} \\ &= \sqrt{\pi A}[\delta(\omega - \varphi) - \delta(\omega + \varphi)] \end{aligned} \quad (153)$$

Estes resultados são condizentes com a teoria apresentada no tópico 1.3. Ou seja, ao aplicar uma excitação senoidal a resposta do sistema será uma onda senoidal com a mesma frequência da senoide de excitação, porém, com amplitude e fase diferentes.

3.5.2.1 Resultado referente à aplicação de um impulso

O resultado a seguir foi obtido utilizando o alto-falante conectado ao circuito *driver* de corrente, que por sua vez recebeu um impulso com duração menor que 1 (ms) no início da amostragem. Ou seja, o comportamento esperado é semelhante ao primeiro caso em que a força de excitação é uma função delta de Dirac. A figura 58 apresenta o resultado, sendo que os três gráficos na coluna esquerda representam respectivamente os eixos x, y e z do acelerômetro no domínio do tempo, e os três gráficos na direita representam seus espectros de frequência computados no ESP8266.

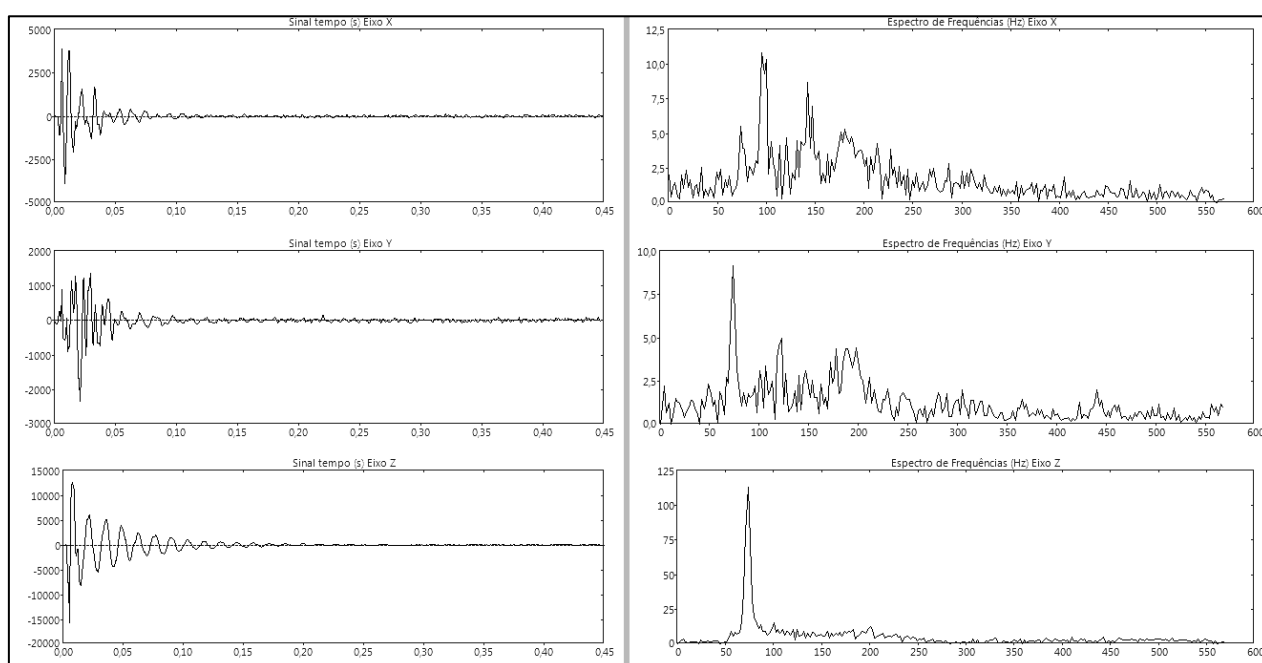


Figura 58 – Espectro de resposta ao impulso no alto-falante.

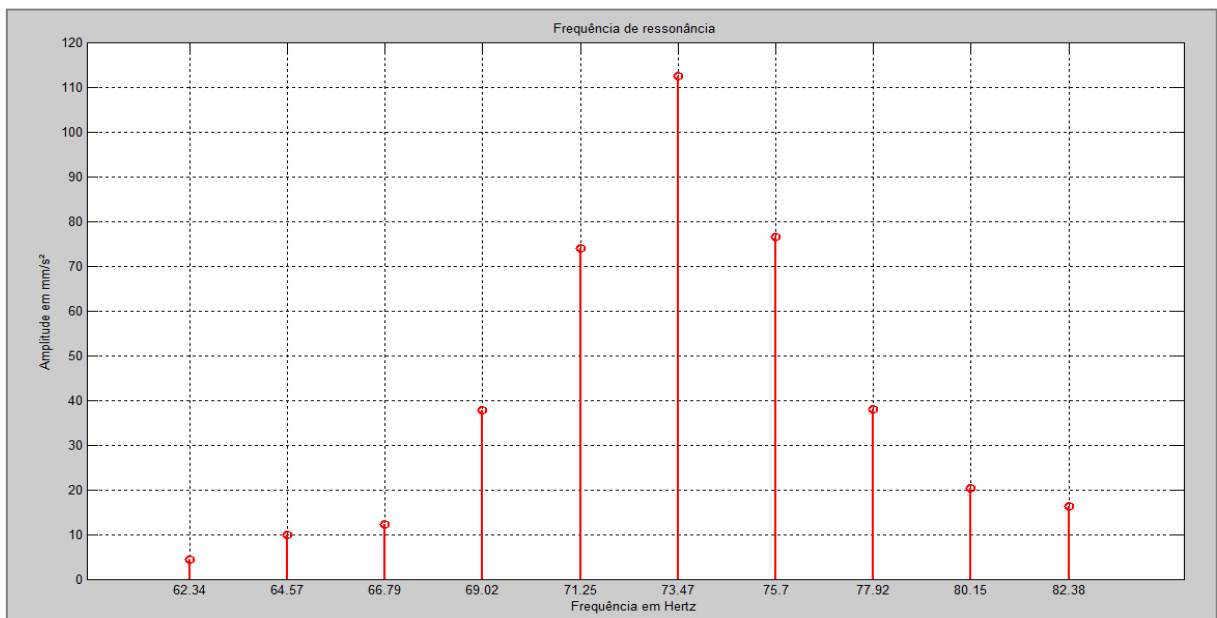
Fonte: Autor.

O eixo z do acelerômetro está alinhado com o eixo de movimento do cone do alto-falante. Fazendo uma inspeção visual no gráfico do espectro do eixo Z, na figura 58, fica evidente que a frequência de ressonância está entre 50 e 100 Hz, provavelmente na faixa dos 72 Hz (ver, a respeito, a figura 12 deste trabalho). Na figura 58, são apresentados graficamente os dados da tabela 7, sendo que estes dados correspondem ao intervalo de frequências entre 60 e 80 Hz do gráfico da figura 58. No gráfico da figura 59, verificamos que o pico que aparece no espectro do eixo z está entre as frequências de 71,2 Hz e 75,7 Hz.

Conforme visto no tópico 1.3.1, a frequência esperada como resposta ao impulso é a frequência natural do sistema (ω_d).

Tabela 7 – Amplitude de resposta próximo a frequência de ressonância.

F(Hz)	A(mm/s²)
62,34	4,4
64,57	9,98
66,79	12,32
69,02	37,67
71,25	74,01
73,47	112,54
75,7	76,5
77,92	38,03
80,15	20,29
82,38	16,2

Fonte: Autor.**Figura 59** – Espectro de amplitudes próximo à frequência de ressonância após impulso no eixo z.**Fonte:** Autor.

A utilização da equação (152), com ajuste de curva empírico dos parâmetros da equação (108), permitiu-nos gerar o gráfico exposto na figura 60, que representa uma aproximação teórica para o resultado obtido no experimento. Para estimar o parâmetro β utilizou-se o sinal no domínio do tempo, servindo-se dos dados em formato de planilha, realizou-se uma interpolação entre os pontos extremantes de ciclo, esperando-se que a curva resultante fosse uma exponencial na forma $s(t) = s_0 e^{-pt}$, mas não foi possível encontrar parâmetros s_0 e p de tal modo que a curva s convergisse sobre a curva interpolada. Após chegar nestes resultados utilizou-se o software Matlab (2010) para ajustar os parâmetros da equação (108) de modo que o resultado apresentado na figura 60, foi o que mais se aproximou do resultado obtido experimentalmente. No gráfico apresentado na figura 61,

temos o espectro da solução expressa pela equação (152) em azul e o espectro da DFT realizado sobre o ajuste feito na equação (108) em vermelho. Esta DFT foi computada pela definição no Matlab (2010) com um vetor de 1000 posições e uma taxa de amostragem de 1kHz. Os valores dos parâmetros utilizados foram:

$$m = 10^{-3}; \beta = 0,075; k = 200,4$$

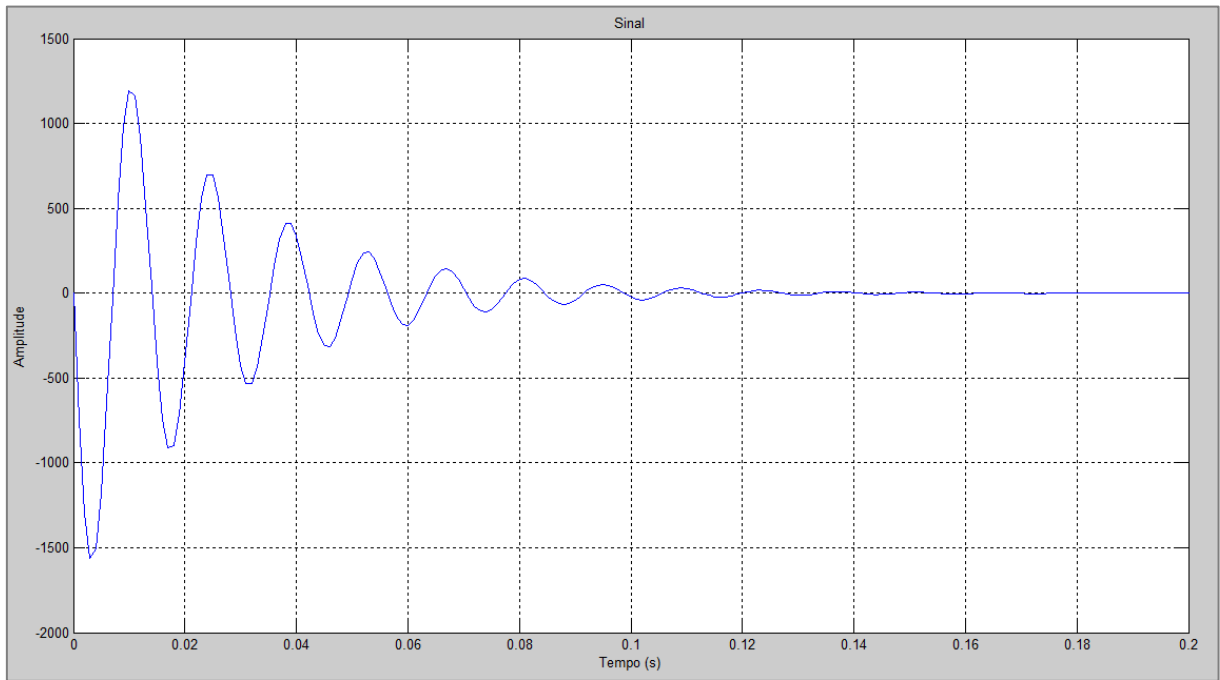


Figura 60 – Aproximação teórica para o resultado do experimento com impulso.

Fonte: Autor.

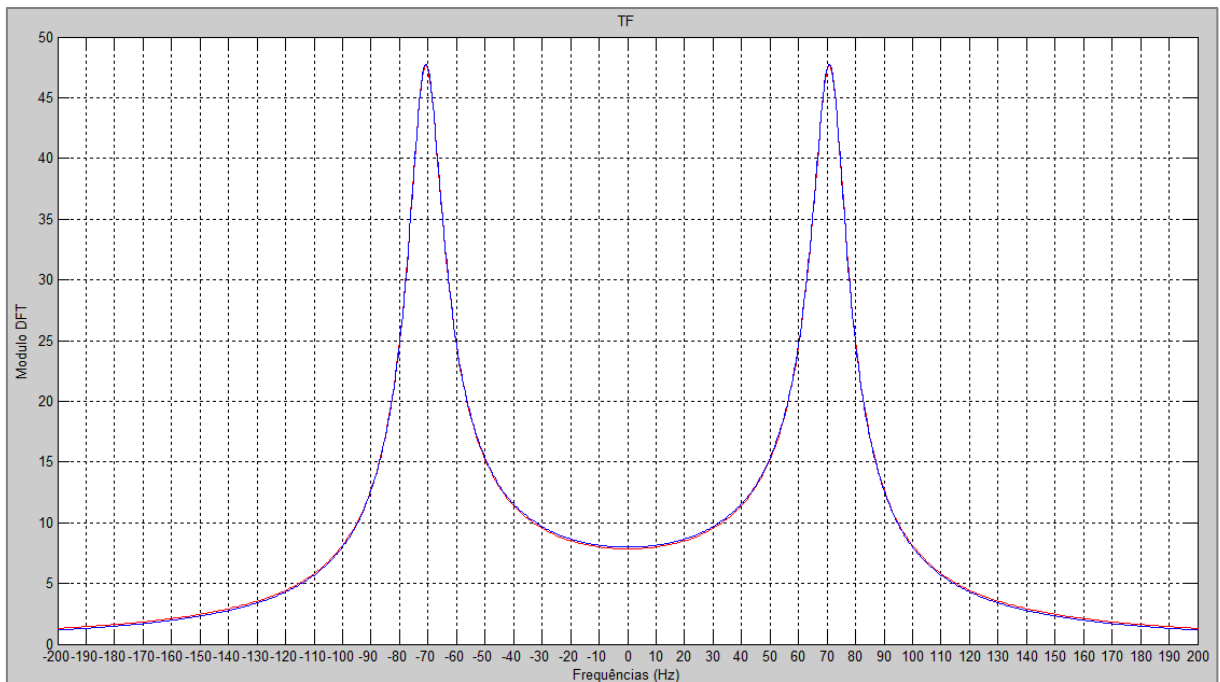


Figura 61 – Espectros do modelo teórico.

Fonte: Autor.

3.5.2.2 Resultado referente à aplicação excitação senoidal

Os resultados a seguir foram obtidos alterando a posição do *jumper* JP1 do circuito *driver* apresentado na figura 42, e conectando-se a saída de um gerador de função ao ponto indicado no circuito como J2. Utilizando o gerador de função variou-se a frequência de uma excitação senoidal de 10Hz até 500Hz, e registrou-se o espectro de frequência obtido utilizando a interface do analisador de vibrações mecânicas, salvando os resultados em planilhas. Em seguida, estes resultados foram reunidos em uma única tabela, registrando a variação da amplitude segundo a frequência. Estes resultados são apresentados na tabela 8 para o intervalo de 50 a 100 Hz. Na figura 62, temos o gráfico de resposta à frequência deste sistema massa-mola-amortecer (alto-falante) para frequências de 10 a 500 Hz. Este gráfico é de interesse para diversas aplicações, inclusive para análise de frequência de ressonância.

Tabela 8 – Resultados de amplitude em relação à frequência aplicada.

F (Hz)	A (mm/s²)
50	1221,33
60	3359,41
64	5388,85
68	8530,85
70	9617,27
72	9140,13
74	8555,47
78	7418,81
80	6320,29
90	4785,43
100	3940,41

Fonte: Autor.

Comparando o gráfico da figura 62 com o da figura 13, percebe-se que o comportamento destas curvas é discrepante. Em Schmith (2011) é mostrado que a dinâmica de um alto-falante não é linear, sendo assim, a modelagem de um alto falante por sistema massa-mola-amortecedor tem sérias limitações. Alguns dos motivos apontados são a falta de rigidez do cone, do anel de suspensão e da aranha, que tornam o comportamento do sistema não linear e até mesmo caótico. Além destes ainda temos a interferência provocada pelo *flat cable* de conexão do acelerômetro, embora durante os testes tomou-se o cuidado de manter a estrutura fixa, não havendo variações na posição do cabo, de modo a reduzir o erro.

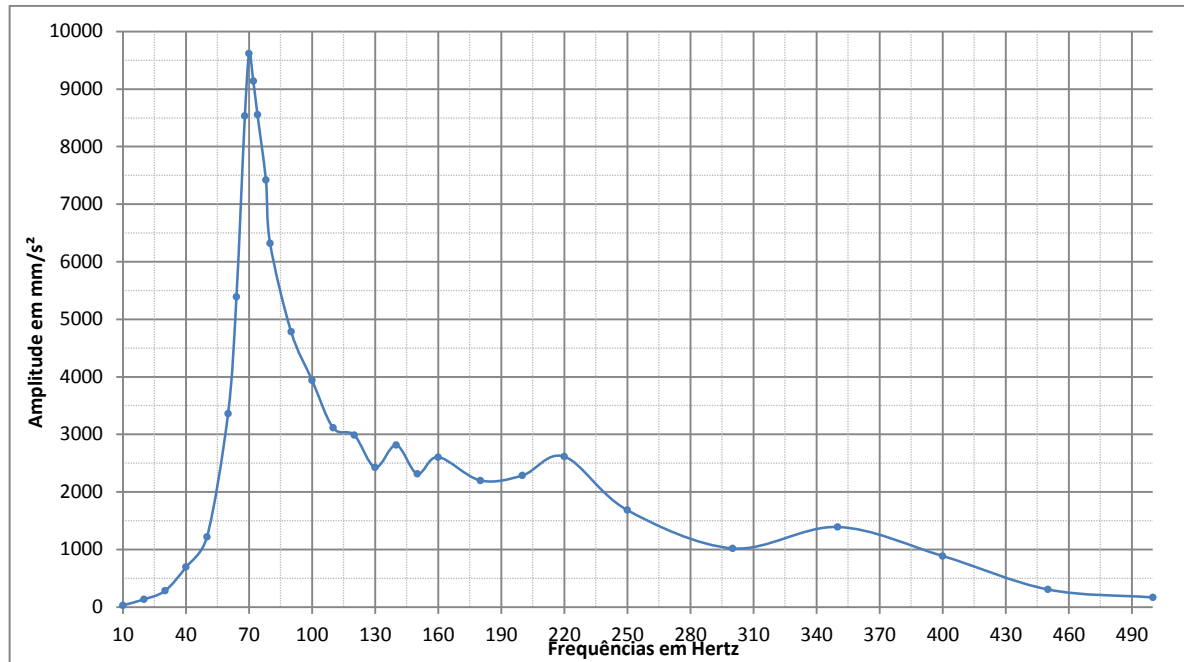


Figura 62 – Espectro de resposta do alto-falante utilizado durante os experimentos.

Fonte: Autor.

Aplicados os parâmetros aproximados empiricamente no tópico anterior e utilizando a equação (118), empregando $\delta_{st} = 270,4$ (calculado de modo que se obtenha a mesma amplitude de resposta na frequência de ressonância), pode-se verificar quão discrepante é a resposta do modelo linear em comparação ao resultado obtido. Na figura 63, é apresentado o gráfico que estabelece esta comparação. Para obter uma aproximação melhor seria preciso modelar a estrutura utilizando a teoria de sistemas dinâmicos não-lineares.

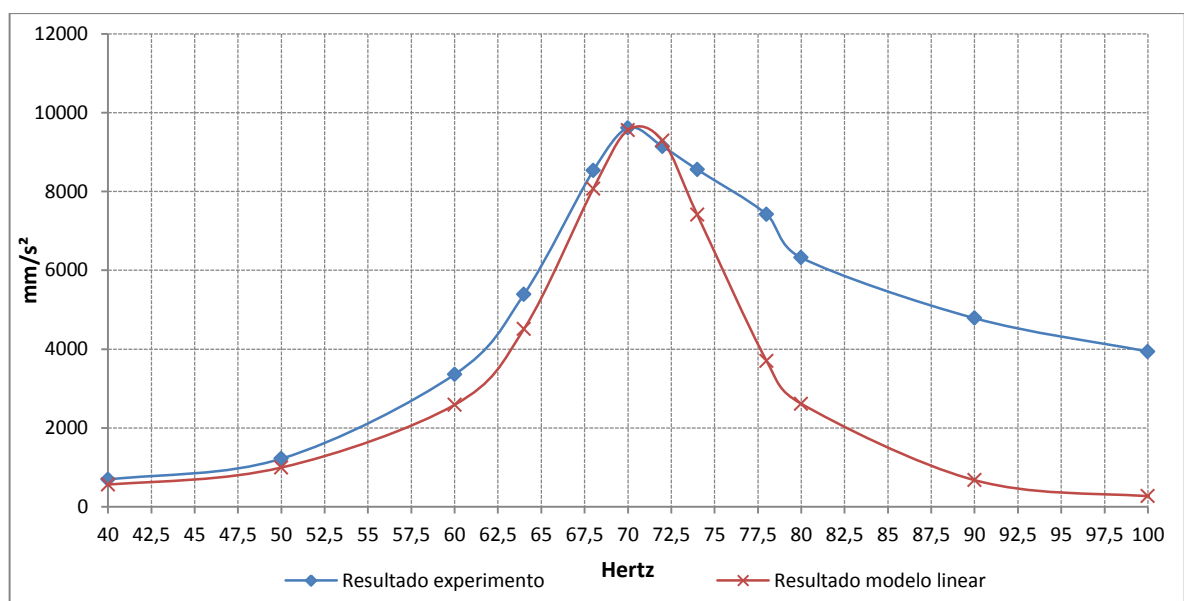


Figura 63 – Comparativo das respostas do modelo linear e da resposta obtida pelo experimento.

Fonte: Autor.

CONCLUSÃO

No início deste trabalho, foi apresentada uma introdução à análise de Fourier. Utilizando os conceitos básicos de funções generalizadas, procurou-se construir uma base matemática sólida para operar com a teoria dos sinais. Em seguida, foram definidas as transformadas contínuas de Fourier e de Hartley. Ainda na revisão de literatura, abordou-se a classificação da complexidade de algoritmos e foi analisado o modo de funcionamento dos algoritmos das transformadas rápidas de Fourier e Hartley, bem como sua estrutura operacional.

Após compreender o funcionamento dos algoritmos rápidos, passou-se à sua implementação. Primeiro, estes algoritmos foram implementados em linguagem C para serem executados em um *prompt* no computador, a fim de se verificar mais facilmente seu funcionamento. Em seguida, estes algoritmos foram implementados no microcontrolador ESP8266, programado através da IDE Arduino. Durante todo esse processo, algumas ideias para aperfeiçoar os códigos surgiram, como por exemplo, a tabela indexada apresentada no capítulo 2.

Durante o teste comparativo entre as transformadas discretas de Hartley (DHT) e de Fourier (DFT), implementadas pela definição, surgiu a ideia de computá-las separando os índices pares e ímpares, e considerando apenas a parte positiva do espectro, o que reduziu o tempo de processamento e o consumo de memória, mantendo, porém, a ordem assintótica $O(N^2)$ do algoritmo. Neste ponto, o emprego de algumas relações trigonométricas permitiu reduzir ainda algumas multiplicações por constantes no algoritmo da transformada de Hartley. Os resultados experimentais mostraram que o processamento da DFT via DHT apresentou um ganho de velocidade de 43,9% em relação à DFT e um ganho de 33,3% de alocação de memória RAM. Neste ponto, ainda foi realizada uma verificação da exatidão dos resultados computados, comparando-os com os resultados calculados através das transformadas contínuas. Uma consideração feita durante um primeiro momento foi a de utilizar sinais múltiplos inteiros da frequência de amostragem. Além disso, após a validação dos resultados iniciais, foram realizados outros testes utilizando sinais diversos, comparando seus resultados com os resultados computados no software Matlab.

Os mesmos testes experimentais realizados na primeira etapa com as transformadas discretas também foram realizados em seguida com as transformadas rápidas, verificando-se imediatamente o aumento da velocidade de processamento destas em relação às discretas. Durante esta comparação, verificou-se que, dada a estrutura do algoritmo Cooley-Tukey, ambos os algoritmos rápidos das transformadas consumiram a mesma quantidade de memória. Porém, observou-se um ganho de 34,8% na velocidade de processamento utilizando a transformada rápida de Hartley para computar o espectro de Fourier. Com isto, concluiu-se que este método pode ser de grande utilidade para implementar a DFT em um microcontrolador. Além disso, foi realizada uma análise do comportamento deste ganho, pois observou-se que devido à complexidade da implementação da FHT existe um adicional de operações que são distribuídas sobre as borboletas segundo o comprimento do vetor de entrada.

Para auxiliar os experimentos realizados, também surgiu a necessidade de desenvolver um *software* para receber os resultados e plotar os gráficos ou ainda salvar os resultados em planilhas. Esta ferramenta foi de grande utilidade durante a validação do funcionamento dos algoritmos no ESP8266 e também serviu de base para implementação da interface para análise de vibrações mecânicas. Todos os códigos desenvolvidos no decorrer deste trabalho estão disponíveis no repositório *on-line Github*, liberados para uso da comunidade.

No capítulo 3, foram apresentadas duas plataformas desenvolvidas para análise de vibrações, com *hardware* e um *firmware* que computa a DFT via FHT. Alguns resultados de análises espectrais aplicados às vibrações mecânicas foram obtidos e reportados. Não é novidade o uso da análise espectral em aplicações de manutenção preditiva. Espera-se que este dispositivo, que possui um baixo custo, encoraje uma maior difusão desta tecnologia, que pode ser aplicada em diversos setores. Conforme foi apresentado no subitem 3.5.1, a análise espectral pode ser uma ferramenta poderosa para manutenção preditiva, tanto para aplicações industriais quanto comerciais. Apresentamos também a modelagem de um alto-falante utilizando a EDO linear que descreve um sistema massa-mola amortecido. Embora esta modelagem tenha suas limitações, ainda assim foi possível verificar como a análise espectral é útil para verificação de modelos teóricos.

Como principais contribuições, este trabalho apresenta uma revisão teórica sobre as transformadas de Fourier e Hartley aplicadas à teoria dos sinais, bem como uma comparação entre as transformadas discretas de Fourier e Hartley, na perspectiva de utilizar-se a transformada de Hartley como uma forma de aumentar a velocidade de processamento da

DFT. O trabalho ainda apresenta uma aplicação prática das transformadas rápidas implementadas, que é um analisador de vibrações mecânicas desenvolvido utilizando um acelerômetro do tipo MEMS.

Este trabalho deixa como principais contribuições:

- Um estudo sistematizado e prático de comparação entre a eficiência do uso da FHT para computar a DFT em sistemas embarcados;
- Material literário sobre a transformada de Hartley;
- Diversos códigos e programas para divulgação do uso da FHT para computar a DFT, e aplicação de análise espectral. Todos códigos livres para o uso da comunidade em repositório *on-line*;
- Uma plataforma experimental para estudos de vibrações em estruturas mecânicas;
- Um dispositivo de análise de vibrações de baixo custo, com *hardware* e *firmware*, podendo esta servir de base para diversas situações de análise de vibrações mecânicas;
- Algumas comparações entre resultados analíticos e computados em forma discreta. Evidenciando a eficiência das transformadas discretas.

CONSIDERAÇÕES FINAIS

O presente estudo apresenta o desempenho comparativo entre as transformadas rápidas de Hartley e Fourier, servindo-se do algoritmo Cooley-Tukey para $N = 2^m$, diversos outros algoritmos podem ser implementados para esta mesma finalidade. Além disso, um estudo mais profundo sobre a aplicação da transformada de Hartley é de interesse para aplicações em sistemas embarcados, tais como analisadores de vibrações mecânicas.

Alguns temas de pesquisa futuros são propostos abaixo com o fim de aprimorar os resultados e as ferramentas obtidos neste trabalho:

- Utilizar o espectro de Hartley para análises de vibrações mecânicas;
- Fazer uma comparação mais abrangente relacionando os resultados das transformadas contínuas com os resultados das transformadas discretas;
- Utilizar a plataforma experimental para montar uma arquitetura de estudos de vibrações mecânicas proposta por Inman (2008, p. 545);
- Encontrar um algoritmo rápido baseado no algoritmo utilizado para computar a DHT com complexidade $O(N^2)$ no capítulo 2;
- Implementar os algoritmos de Cooley-Tukey *radix-4* e o *Split-radix* para verificar as diferenças de desempenho em um microcontrolador;
- Aprofundar o modelo que descreve a curva de ganho da transformada rápida de Hartley (FHT) em relação à transformada rápida de Fourier (FFT) apresentado no tópico 2.3.1 deste trabalho, pois este modelo pode mostrar que para N muito grande o ganho estabiliza;
- Alterar o software e o firmware do analisador de vibrações mecânicas para análise automática da resposta à aplicação de excitações senoidais conforme foi realizado no tópico 3.5.2.2 deste trabalho. Ou seja, o analisador gera o gráfico de resposta de um sistema massa-mola-amortecedor, semelhante ao da figura 62, variando automaticamente as frequências de excitação e registrando seus resultados;
- Utilizar o analisador de vibrações desenvolvido para testar modelos matemáticos não-lineares para dispositivos tais como alto-falantes.

REFERÊNCIAS

AMICA. Site do fabricante NodeMCU v2. Disponível em: <www.nodemcu.com>. Acesso em: 14 maio 2016.

BALACHANDRAN, B.; MAGRAB, E. B. **Vibrações mecânicas**. São Paulo: Cengage Learning, 2011.

BLAHUT, R. E.; **Fast Algorithms for Signal Processing**. Nova York: Cambridge University Press 2010. Disponível em: <<http://ir.nmu.org.ua/bitstream/handle/123456789/132674/6c6ca5b3d19a6f1ffd68415bef8a99e6.pdf?sequence=1>>. Acesso em: 17 ago. 2015.

BRACEWELL, R. N. **The Fast Hartley Transform**. Stanford: Stanford University, 1982. Proceedings of the IEEE, VOL. 72, NO. 8, AUGUST 1984. Disponível em: <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1457236&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1457236>. Acesso em: 06 Maio 2016.

CASTILHO, S. **Comparação de eficiência computacional entre as transformadas rápidas de Fourier (FFT) e de Hartley (FHT)**. TCC (Graduação em Ciências da computação); [orientado por] Prof. Júlio Felipe Szeremeta – Florianópolis, SC: UFSC, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_392/tcc.pdf>. Acesso em: 06 Maio 2016.

CREDE, C. E. **Choque e vibração nos projetos de engenharia**. Rio de Janeiro: Ao Livro Técnico, 1972.

CUNHA, I. **Análise comparativa de algoritmos que computam transformada discreta de Fourier e Hartley**. TCC (Graduação em Engenharia de computação);[orientado por] Prof. Dr. Raimundo Corrêa de Oliveira – Manaus, AM: UEA, 2013. Disponível em: <<http://tcc-computacao.tiagodemelo.info/monografias/2013/tcc-iasmim-cunha.pdf>>. Acesso em: 15 Jul. 2016.

ESP8266, *Datasheet*. Disponível em <https://cdn-shop.adafruit.com/datasheets/ESP8266_Specifications_English.pdf>. Acesso em: 06 Jun. 2016.

ESPRESSIF SYSTEMS – Site oficial do fabricante do ESP8266. Disponível em: <<https://espressif.com/en/products/hardware/esp8266ex/overview>>. Acesso em: 04 de Jun. de 2016.

FERREIRA, J. C. **Introdução à teoria das distribuições**. Lisboa: Fundação Calouste Gulbenkian, 1993.

FIGUEIREDO, D. G. **Análise de Fourier e equações diferenciais parciais**. 4. ed. Rio de Janeiro: IMPA, 2007.

FIGUEIREDO, L. J.; GAFANIZ A. R.; LOPES, G. S.; PEREIRA, R. **Aplicações de acelerómetros**. Lisboa: Monografia IAS, 2007. Disponível em: <<https://nebm.ist.utl.pt/repositorio/download/375>>. Acesso em: 25 Jul. 2016.

HARTLEY, R. V. L. **A More Symmetrical Fourier Analysis Applied to Transmission Problems**. Bell Telephone Laboratories, Inc., New York, N.Y., 1942. Disponível em: <<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1694454>>. Acesso em: 06 Maio 2016.

HSU, H. P. **Análise de Fourier**. Rio de Janeiro: L.T.C., 1972.

HSU, H. P. **Sistemas e Sinais**. 2ª Edição. São Paulo: Bookman Companhia editora Ltda, 2012.

INMAN, Daniel J. **Engineering vibration**. 3ª Edição. Nova Jersey: Pearson Prentice Hall 2008.

IÓRIO JR, R.; IÓRIO, V. M. **Equações diferenciais parciais: uma introdução**. Rio de Janeiro: IMPA, 1988.

KOLBAN, Neil. **Kolban's Book on ESP8266**. Ebook, 2015. Disponível em: <<http://neilkolban.com/tech/esp8266/>>. Acesso em: 04 Jun. 2016.

MPU-6050, *Datasheet*. Disponível em <https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf>. Acesso em: 09 Ago. 2016

OLEJNICZAK, K. J. **The Transforms and Applications Handbook**. 2. ed. Boca Raton: CRC Press LLC, 2000. Disponível em: < <http://dsp-book.narod.ru/TAH/ch04.pdf> >. Acesso em: 11 Maio 2016.

ORAINTARA, Soontorn. **The Unified Discrete Fourier-Hartley Transforms: Theory and Structure**. University of Texas at Arlington, EE Dept., Arlington, TX., 2002. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1010253&newsearch=true&queryText=UDFHT%20The%20unified%20discrete%20Fourier-Hartley%20transforms%20Theory%20and%20structure>>. Acesso em: 15 Jun. 2016.

POULARIKAS, A. D. **The Handbook of Formulas and Tables for Signal Processing**. Boca Raton: CRC Press LLC, 1999. Disponível em: <<http://dsp-book.narod.ru/HFTSP/8579ch14.pdf>>. Acesso em: 11 Maio 2016.

PUPIN, J. R. **Introdução às séries e transformadas de Fourier com aplicações no processamento de imagens**. Universidade Federal de São Carlos, 2011. Disponível em: <<http://www.dm.ufscar.br/dm/attachments/article/6/TCCJosiana.pdf>>. Acesso em: 16 Nov. 2015.

RAO, S. **Vibrações mecânicas**. 4. ed. São Paulo: Pearson Prentice Hall, 2008.

RAO, K.R.; KIM, D.N.; HWANG, J.J.; **Fast Fourier Transform: Algorithms and Applications**. Londres/Nova York: Springer Dordrecht Heidelberg, 2010.

RICIERI, A. P.. **Construindo a Série de Fourier**. Edições Prandiano, 1988.

RUDIN, W. **Functional Analysis**. Nova York: McGraw-Hill, 2ª Edição, 1993.

SCHMITH, J. **Modelagem e simulação da dinâmica de alto-falantes em caixas acústicas seladas**. Dissertação de mestrado (Ciências da computação); [orientado por] Prof. Dr. Luiz Paulo Luma de Oliveira – São Leopoldo, RS: UNISINOS, 2011. Disponível em: <http://www.repositorio.jesuita.org.br/bitstream/handle/UNISINOS/3365/modelagem_simulacao.pdf?sequence=1&isAllowed=y>. Acesso em: 08 Nov. 2016.

SETO, W.W. B. **Vibrações Mecânicas**. Rio de Janeiro: Mcgraw-Hill do Brasil, 1971.

SPIEGEL, M. R. **Análise de Fourier**. São Paulo: Mcgraw-Hill do Brasil/Pearson Makron Books, 1976.

STEWART, Ian. **17 Equações que Mudaram o Mundo**. 1.ed. Rio de Janeiro: Zahar, 2013.

TAKAHASHI, R. H. C. **Transformada Discreta de Fourier: Motivação e aplicações**. Notas de aula de Mini Curso, BH - I Bienal de Matemática – SBM, 2002. Disponível em: < <http://www.cnpq.br/documents/10157/9fe4bbf5-0bfe-4d3f-b9bb-17f9b827e228>>. Acesso em: 23 Fev. 2016.

TOSCANI, L. V.; VELOSO, P. A. S. **Complexidade de algoritmos**. 3. ed. Porto Alegre: Bookman, 2012.

XTENSA, Xtensa ISA Reference Manual. Disponível em: <<http://0x04.net/~mwk/doc/xtensa.pdf>>. Acesso em: 04 Jun. 2016.

ZILL, D. G; CULLEN, M. R. **Equações diferenciais**. Volume 2, 3. ed. São Paulo: Pearson Makron Books, 2001.

APÊNDICE A

Teorema da convolução para transformada de Fourier

Sejam $f(t)$ e $g(t)$ duas funções reais. A convolução de $f(t)$ e $g(t)$ é definida pela função:

$$h(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

Simbolicamente representada por:

$$h(t) = f(t) * g(t)$$

Um caso especial e importante é quando $f(t) = g(t) = 0 \forall t < 0$. Neste caso tem-se:

$$h(t) = \int_0^t f(\tau)g(t - \tau)d\tau$$

A convolução satisfaz a lei da comutatividade $f(t) * g(t) = g(t) * f(t)$.

No caso em que $g(t) = \delta(t)$, onde $\delta(t)$ é a função impulso.

$$f(t) * g(t) = \delta(t) * f(t) = \int_{-\infty}^{\infty} \delta(\tau)f(t - \tau)d\tau = f(t)$$

Teorema de convolução no tempo:

Se $\mathcal{F}\{f(t)\} = F(\omega)$ e $\mathcal{F}\{g(t)\} = G(\omega)$, então:

$$\mathcal{F}\{f(t) * g(t)\} = F(\omega)G(\omega)$$

Prova:

$$\mathcal{F}\{f(t) * g(t)\} = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \right] e^{-i\omega t} dt$$

Invertendo a ordem de integração, temos.

$$\mathcal{F}\{f(t) * g(t)\} = \int_{-\infty}^{\infty} f(\tau) \left[\int_{-\infty}^{\infty} g(t - \tau)e^{-i\omega t} dt \right] d\tau$$

Considerando a propriedade de deslocamento no tempo da transformada de Fourier, ou seja:

$$\mathcal{F}\{g(t - \tau)\} = \int_{-\infty}^{\infty} g(t - \tau) e^{-i\omega t} dt$$

Fazendo $t - \tau = u$ temos $dt = du$, logo:

$$\begin{aligned} \mathcal{F}\{g(t - \tau)\} &= \int_{-\infty}^{\infty} g(u) e^{-i\omega(\tau+u)} du = e^{-i\omega\tau} \int_{-\infty}^{\infty} g(u) e^{-i\omega u} du \\ \therefore \mathcal{F}\{g(t - \tau)\} &= e^{-i\omega\tau} G(\omega) \end{aligned}$$

Retomando a transformada da convolução entre $f(t)$ e $g(t)$, temos:

$$\begin{aligned} \mathcal{F}\{f(t) * g(t)\} &= G(\omega) \int_{-\infty}^{\infty} f(\tau) e^{-i\omega\tau} d\tau \\ \therefore \mathcal{F}\{f(t) * g(t)\} &= F(\omega) G(\omega) \quad c. q. d. \end{aligned}$$

Teorema de convolução na frequência:

Se $\mathcal{F}^{-1}\{F(\omega)\} = f(t)$ e $\mathcal{F}^{-1}\{G(\omega)\} = g(t)$, então:

$$\mathcal{F}\{f(t)g(t)\} = \frac{1}{2\pi} F(\omega) * G(\omega)$$

Prova:

$$\mathcal{F}\{f(t)g(t)\} = \frac{1}{2\pi} F(\omega) * G(\omega) \Rightarrow \mathcal{F}^{-1}\{F(\omega) * G(\omega)\} = 2\pi f(t)g(t)$$

Então,

$$\mathcal{F}^{-1}\{F(\omega) * G(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F(\Omega) G(\omega - \Omega) d\Omega \right] e^{i\omega t} d\omega$$

Fazendo uma substituição $v = \omega - \Omega$ e trocando a ordem de integração.

$$\mathcal{F}^{-1}\{F(\omega) * G(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega) e^{i\Omega t} \left[\int_{-\infty}^{\infty} G(v) e^{ivt} dv \right] d\Omega$$

$$\mathcal{F}^{-1}\{F(\omega) * G(\omega)\} = 2\pi \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \right] \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{i\omega t} d\omega \right]$$

$$\therefore \mathcal{F}^{-1}\{F(\omega) * G(\omega)\} = 2\pi f(t)g(t) \Rightarrow \mathcal{F}\{f(t)g(t)\} = \frac{1}{2\pi} F(\omega) * G(\omega) \quad c. q. d.$$

Podendo ser escrita também como:

$$\mathcal{F}\{f(t)g(t)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega) G(\omega - \Omega) d\Omega$$

APÊNDICE B

Resposta de sistema massa-mola amortecido a um impulso

Partindo da EDO que modela o massa-mola-amortecedor, onde m é a massa, β é o coeficiente de atrito viscoso e k é a constante de restauração da mola, qual seja:

$$m \frac{d^2x}{dt^2} + \beta \frac{dx}{dt} + kx = f(t)$$

É conveniente fazer a seguinte substituição:

$$\omega_n = \sqrt{\frac{k}{m}}; \zeta = \frac{\beta}{2m\omega_n}; g(t) = \frac{f(t)}{m}$$

Obtêm-se então:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = g(t)$$

Para o caso em que a função de excitação é um impulso na origem, ou seja

$$f(t) = \delta(t)$$

A EDO assim assume a forma:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = \frac{\delta(t)}{m}$$

Resolvendo a EDO utilizando a transformada de Laplace, tem-se:

$$\mathcal{L} \left\{ \frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x \right\} = \int_{0^-}^{\infty} \delta(t) e^{-st} dt = \frac{1}{m}$$

Considerando $x'(0) = 0$ e $x(0) = 0$, tem-se:

$$X(s) = \frac{1}{m(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{1}{m(s - s_1)(s - s_2)}$$

Onde $s_{1,2}$ são as raízes do polinômio do denominador:

$$s_{1,2} = \frac{-2\zeta\omega_n \pm \sqrt{4(\zeta\omega_n)^2 - 4\omega_n^2}}{2} = -\zeta\omega_n \pm i\omega_d$$

Onde: $\omega_d = \omega_n\sqrt{1 - \zeta^2}$

Definindo:

$$P(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$$

Derivando este polinômio, temos:

$$P'(s) = 2s + 2\zeta\omega_n$$

Utilizando o desenvolvimento de Heaviside para encontrar a transformada inversa de Laplace, tem-se que:

$$\mathcal{L}^{-1}\{X(s)\} = \frac{\mathcal{U}(t)}{m} \left[\frac{e^{s_1 t}}{P'(s_1)} + \frac{e^{s_2 t}}{P'(s_2)} \right]$$

Logo:

$$x(t) = \frac{\mathcal{U}(t)}{m} \left[\frac{e^{(-\zeta\omega_n + i\omega_d)t}}{2(-\zeta\omega_n + i\omega_d) + 2\zeta\omega_n} + \frac{e^{(-\zeta\omega_n - i\omega_d)t}}{2(-\zeta\omega_n - i\omega_d) + 2\zeta\omega_n} \right]$$

$$x(t) = \frac{\mathcal{U}(t)}{m} e^{-\zeta\omega_n t} \left(\frac{e^{i\omega_d t} - e^{-i\omega_d t}}{2i\omega_d} \right)$$

$$\therefore x(t) = \frac{\mathcal{U}(t)}{m\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t)$$

Que é a resposta procurada de um sistema massa-mola amortecido submetido a um impulso.

APÊNDICE C

Espectro de frequências do sistema massa-mola amortecido em resposta à função impulso

Partindo da função resposta de um sistema massa-mola amortecido submetido a um impulso encontrada no Apêndice B, qual seja:

$$x(t) = \frac{\mathcal{U}(t)}{m\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t)$$

Aplicando a transformada de Fourier em $x(t)$:

$$\mathcal{F}\{x(t)\} = \frac{1}{m\omega_d} [\mathcal{F}\{\mathcal{U}(t)e^{-\zeta\omega_n t}\} * \mathcal{F}\{\sin(\omega_d t)\}]$$

Onde o símbolo “*” denota a operação de convolução.

$$\therefore \mathcal{F}\{x(t)\} = \frac{1}{m\omega_d} \left\{ \left[\int_0^\infty e^{-t(\zeta\omega_n + i\omega)} dt \right] * [-i\pi(\delta(\omega - \omega_d) - \delta(\omega + \omega_d))] \right\}$$

Considerando $Re(\zeta\omega_n + i\omega) > 0$ tem-se:

$$\mathcal{F}\{x(t)\} = \frac{-i\pi}{2\pi m\omega_d} \int_{-\infty}^{\infty} \frac{\delta(\Omega - \omega_d) - \delta(\Omega + \omega_d)}{[\zeta\omega_n + i(\omega - \Omega)]} d\Omega$$

$$= \frac{1}{2im\omega_d} \left\{ \frac{i(\omega + \omega_d - \omega + \omega_d)}{[\zeta\omega_n + i(\omega - \omega_d)][\zeta\omega_n + i(\omega + \omega_d)]} \right\}$$

$$\therefore X(\omega) = \frac{1}{m[(\zeta\omega_n)^2 - (\omega^2 - \omega_d^2) + i2\zeta\omega_n\omega]} = \frac{(\zeta\omega_n)^2 - (\omega^2 - \omega_d^2) - i2\zeta\omega_n\omega}{m\{[(\zeta\omega_n)^2 - (\omega^2 - \omega_d^2)]^2 + 4(\zeta\omega_n\omega)^2\}}$$

$$\Rightarrow X(\omega) = \frac{(\zeta\omega_n)^2 - \omega^2 + \omega_n^2(1 - \zeta^2) - i2\zeta\omega_n\omega}{m\{[(\zeta\omega_n)^2 - \omega^2 + \omega_n^2(1 - \zeta^2)]^2 + 4(\zeta\omega_n\omega)^2\}}$$

$$\therefore X(\omega) = \frac{\omega_n^2 - \omega^2 - i2\zeta\omega_n\omega}{m[(\omega_n^2 - \omega^2)^2 + 4(\zeta\omega_n\omega)^2]} \quad (i)$$

Outra forma de obter o mesmo resultado consistiria em aplicar a transformada de Fourier diretamente na EDO que modela o sistema massa-mola amortecido submetido a um impulso:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = \frac{\delta(t)}{m}$$

Assim fazendo, tem-se:

$$\mathcal{F}\left\{\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x\right\} = \mathcal{F}\left\{\frac{\delta(t)}{m}\right\}$$

$$(-\omega^2 + i\omega 2\zeta\omega_n + \omega_n^2)X(\omega) = \frac{1}{m} \int_{-\infty}^{\infty} \delta(t)e^{-i\omega t} dt = \frac{1}{m}$$

$$\therefore F(\omega) = \frac{1}{m(\omega_n^2 - \omega^2 + i\omega 2\zeta\omega_n)} = \frac{\omega_n^2 - \omega^2 - i\omega 2\zeta\omega_n}{m[(\omega_n^2 - \omega^2)^2 + (\omega 2\zeta\omega_n)^2]} \quad (ii)$$

Comparando as equações (i) e (ii), verifica-se que $F(\omega) = X(\omega)$, como esperado.

Lembrando que $\omega_n = \sqrt{\frac{k}{m}}$ e $\zeta = \frac{\beta}{2m\omega_n}$, pode-se ainda verificar que:

$$|F(\omega)| = \frac{\sqrt{(\omega_n^2 - \omega^2)^2 + 4(\omega\zeta\omega_n)^2}}{m[(\omega_n^2 - \omega^2)^2 + 4(\omega\zeta\omega_n)^2]} = \frac{\sqrt{(k - m\omega^2)^2 + (\beta\omega)^2}}{(k - m\omega^2)^2 + (\beta\omega)^2}$$