

DANIEL VIDAL

Medição de rotação de motores de combustão interna por meio de vibração

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de
São Carlos, da Universidade de São
Paulo.

Curso de Engenharia Elétrica com ênfase em eletrônica

ORIENTADOR: Evandro Luis Linhari Rodrigues

São Carlos

2008

Resumo

O trabalho apresentado tem como objetivo o desenvolvimento e implementação de um método indireto para medida de rotação de motores a explosão utilizando a vibração inerente a seu funcionamento. O trabalho desenvolvido teve parte dele realizado durante o estágio supervisionado. Para a determinação do algoritmo para a medida levantou-se a origem da vibração de motores a combustão interna e sua relação com a rotação do eixo utilizando análise no domínio da frequência. Para a implementação do algoritmo foi necessário o desenvolvimento de hardware e software embarcado utilizando técnicas de processamento digital de sinais. O equipamento encontra-se em estágio de protótipo para testes, ajustes e melhorias. Os primeiros resultados obtidos foram considerados animadores apesar de algumas falhas de medidas, que podem ser contornadas utilizando um algoritmo de medida mais robusto, sugerido para continuação do trabalho.

Palavras chave: MEDIDA, TACÔMETRO, ROTAÇÃO, MOTOR, VIBRAÇÃO, DSP,

Abstract

The present work goals are the development and implementation of an indirect measurement method for internal combustion engine speed, using the vibration that is an intrinsic characteristic. The work had some parts developed during the internship. The algorithm development needed some studies about the nature of the engine vibration and its relation to the engine's speed. The implementation has been done on an embedded hardware and software system with some digital signal processing techniques. The developed system is at the prototyping stage for tests and adjustments. The results seen on the first tests are good despite some fault results, which can be resolved with a more robust algorithm suggested as future work.

Keywords: MEASUREMENT, *TACHOMETER*, *ROTATION*, *VIBRATION*, *ENGINE*, *DSP*

Sumário

| | |
|--|------|
| Resumo..... | i |
| Abstract | ii |
| Sumário | iii |
| Lista de figuras..... | v |
| Lista de tabelas..... | vii |
| Lista de Siglas | viii |
| 1 Introdução..... | 1 |
| 1.1 Métodos convencionais para medição de rotação | 1 |
| 1.2 Método proposto – Vibração..... | 2 |
| 2 Introdução teórica..... | 5 |
| 2.1 Representação de números com ponto fixo e ponto flutuante..... | 5 |
| 2.2 Transformada de Fourier | 7 |
| 3 Especificações do equipamento..... | 11 |
| 4 Desenvolvimento de hardware | 13 |
| 4.1 Arquitetura | 13 |
| 4.2 Processador..... | 13 |
| 4.3 Acelerômetro..... | 14 |
| 4.4 Chaves | 14 |
| 4.5 Módulo LCD | 15 |
| 4.6 Pulso..... | 16 |
| 4.7 RS-232..... | 17 |
| 4.8 Diagrama esquemático e layout | 18 |
| 5 Desenvolvimento do software | 20 |
| 5.1 Ambiente de desenvolvimento | 20 |
| 5.2 Algoritmo geral de funcionamento..... | 20 |
| 5.3 Inicialização do sistema | 21 |
| 5.4 Aquisição dos sinais..... | 23 |
| 5.5 Processamento dos sinais | 23 |
| 5.6 Análise em frequência..... | 24 |
| 5.7 Interface LCD..... | 25 |
| 5.8 Interface UART..... | 25 |
| 5.9 Geração da saída pulso..... | 26 |
| 5.10 Interface das chaves..... | 27 |

| | | |
|-----|---|----|
| 6 | Resultados e discussão | 29 |
| 6.1 | Resultados obtidos..... | 29 |
| 6.2 | Discussão..... | 38 |
| 7 | Conclusões..... | 39 |
| 8 | Trabalhos futuros..... | 40 |
| 8.1 | Algoritmo mais robusto para detecção..... | 40 |
| 8.2 | Aumento da resolução da medida | 40 |
| | Referências bibliográficas | 41 |

Lista de figuras

| | |
|---|----|
| Figura 1: Medição da rotação por relutância variável..... | 1 |
| Figura 2: Diagrama de eixos no bloco do motor..... | 4 |
| Figura 3: Representação numérica em ponto fixo..... | 5 |
| Figura 4: Representação numérica em ponto flutuante..... | 6 |
| Figura 5: Redução da DFT de 8 pontos para 4 pontos..... | 8 |
| Figura 6: Redução da DFT de 4 pontos para 2 pontos..... | 8 |
| Figura 7: DFT de 2 pontos..... | 8 |
| Figura 8: Estágio borboleta do algoritmo FFT..... | 8 |
| Figura 9: Diagrama completo do algoritmo FFT de 8 pontos com decimação no tempo..... | 9 |
| Figura 10: Diagrama completo do algoritmo FFT de 8 pontos com decimação em frequência ... | 9 |
| Figura 11: Saída tipo pulso..... | 11 |
| Figura 12: Arquitetura do hardware..... | 13 |
| Figura 13: Chave táctil..... | 14 |
| Figura 14: Interface para as chaves..... | 15 |
| Figura 15: Módulo LCD HD44780..... | 15 |
| Figura 16: Interface com módulo LCD..... | 16 |
| Figura 17: Módulo para saída pulso..... | 17 |
| Figura 18: Sinais do módulo de saída pulso..... | 17 |
| Figura 19: Conexão do transceiver RS232..... | 18 |
| Figura 20: Módulo de comunicação RS232..... | 18 |
| Figura 21: Layout PCI protótipo..... | 19 |
| Figura 22: Fluxograma geral de funcionamento..... | 20 |
| Figura 23: Diagrama do PLL..... | 22 |
| Figura 24: Fluxograma das rotinas de aquisição dos sinais..... | 23 |
| Figura 25: Fluxograma do processamento dos sinais..... | 24 |
| Figura 26: Diagrama dos sinais para escrita de comandos e dados no módulo LCD..... | 25 |
| Figura 27: Fluxograma para escrita de comandos e dados no módulo LCD..... | 25 |
| Figura 28: Fluxograma das rotinas UART..... | 26 |
| Figura 29: Diagrama de estados da saída pulso..... | 27 |
| Figura 30: Procedimentos para controle da saída pulso..... | 27 |
| Figura 31: Fluxograma para detecção das chaves..... | 28 |
| Figura 32: Diagrama dos estados da navegação pelas chaves..... | 28 |
| Figura 33: Placa do protótipo..... | 29 |

| | |
|--|----|
| Figura 34: Montagem do acelerômetro | 29 |
| Figura 35: Medidas de vibração do veículo Ford Ranger com rotação 840 RPM | 31 |
| Figura 36: Medidas de vibração do veículo Ford Ranger com rotação 1250 RPM | 32 |
| Figura 37: Medidas de vibração do veículo Ford Ranger com rotação 1990 RPM | 33 |
| Figura 38: Medidas de vibração do veículo Ford Ranger com rotação 2680 RPM | 34 |
| Figura 39: Medidas de vibração do veículo Toyota Hilux com rotação 1000 RPM..... | 35 |
| Figura 40: Medidas de vibração do veículo Toyota Hilux com rotação 1700 RPM..... | 36 |
| Figura 41: Medidas de vibração do veículo Toyota Hilux com rotação 2400 RPM..... | 37 |

Lista de tabelas

| | |
|---|----|
| Tabela 1: Reorganização de um vetor de 8 posições por inversão da ordem dos bits de índice | 9 |
| Tabela 2: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 840 RPM..... | 31 |
| Tabela 3: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1250 RPM..... | 32 |
| Tabela 4: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1990 RPM..... | 33 |
| Tabela 5: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1990 RPM..... | 34 |
| Tabela 6: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 1000 RPM..... | 35 |
| Tabela 7: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 1700 RPM..... | 36 |
| Tabela 8: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 2400 RPM..... | 37 |

Lista de Siglas

- FFT – *Fast Fourier Transform*
- DFT – *Discrete Fourier Transform*
- DMA – *Direct Memory Access*
- DSC – *Digital Signal Controller*
- DSP – *Digital Signal Processor*
- LCD – *Liquid crystal display*
- PCI – *Placa de circuito impresso*
- RPM – *Rotações por minuto*
- ULA – *Unidade lógica e aritmética*

1 Introdução

Rotação é o nome dado à velocidade angular de um corpo submetido a movimento circular. Constitui um dos parâmetros de funcionamento dos motores, estabelecendo uma relação de interdependência entre potência e torque, desempenho e eficiência.

Os motores, de modo geral, são projetados para operar dentro de uma faixa de rotação ótima, onde tem desempenho e eficiência otimizada para um faixa de torque estabelecida, sendo portanto a rotação um dos parâmetros que devem ser monitorados em testes e diagnósticos de motores.

A maneira mais simples e intuitiva para se medir a rotação é contar o número de voltas do eixo num determinado período de tempo. Trata-se de um método simples, porém inconveniente, pois se deve instalar algum dispositivo no eixo do motor, e em alguns casos, quando não há acesso ao eixo motriz esta abordagem é impraticável.

O trabalho apresentado tem como objetivo o desenvolvimento de um tacômetro, equipamento para medida de rotação, a ser utilizado em motores a combustão interna utilizando um método indireto não convencional: análise da vibração inerente ao funcionamento do motor.

A seguir é mostrada uma explanação sobre os métodos convencionais e o método proposto para a medição de rotação de motores.

1.1 Métodos convencionais para medição de rotação

Relutância variável

Os módulos de injeção eletrônica encontradas nos automóveis medem a rotação através de um sensor indutivo localizado junto a um volante com ranhuras, acoplado ao eixo motriz. A relutância variável vista no sensor indutivo faz com que sua saída seja um sinal de frequência proporcional ao número de ranhuras e a rotação do motor. A montagem do sensor é mostrada na figura a seguir:

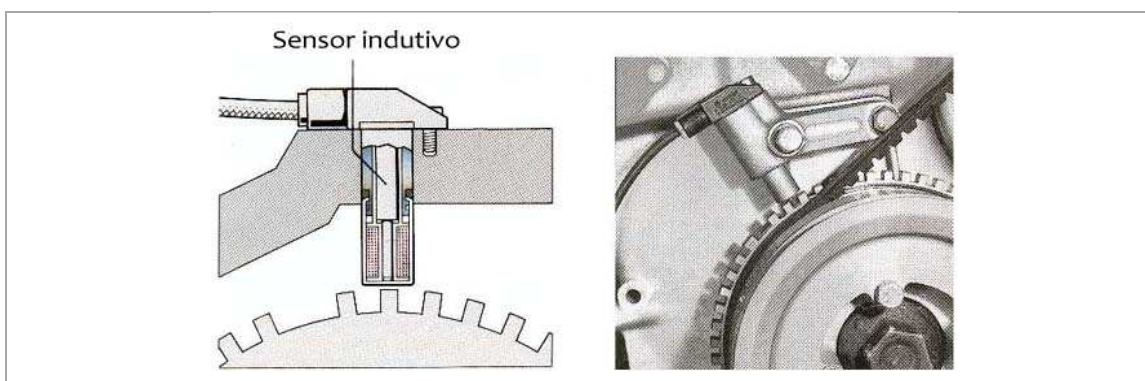


Figura 1: Medição da rotação por relutância variável

Esse método também possibilita a determinação da posição angular do eixo através da ausência de um dos dentes do volante. Esse dado é utilizado pela central de injeção eletrônica para a sincronização dos processos do motor (injeção de combustível, disparo das velas, etc.)

Tacômetros comerciais convencionais

Os tacômetros convencionais para uso geral também tem de ter acesso ao eixo motriz para contar-se o número de revoluções do eixo. Os métodos utilizados para as medidas nos aparelhos comerciais são diversos:

- **Contato:** o equipamento possui um *encoder*, o qual é acoplado mecanicamente ao eixo motriz.
- **Óptico:** o equipamento lança um feixe de *laser* no eixo motriz e conta o número de reflexões do feixe de luz através de uma fita adesiva reflexiva fixada pelo usuário no eixo.
- **Luz estroboscópica:** fixa-se uma marca de referência no eixo motriz e varre-se a frequência da fonte de luz estroboscópica até que a marca de referência pareça estacionária, nesta condição a frequência de rotação é um múltiplo da frequência da fonte de luz.

Todos os métodos diretos dos tacômetros apresentados são invasivos de algum modo, ou seja, necessitam de acesso ao eixo motriz para o caso dos tacômetros comerciais ou do acesso ao sensor indutivo caso esteja presente no automóvel.

O acesso ao eixo pode necessitar a remoção de peças do motor o que torna o processo inconveniente e em alguns casos impossível: quando não há acesso ao eixo motriz. Outro ponto negativo é que o acesso ao eixo e manuseio de instrumentos de medida acoplados ao mesmo constituem uma provável fonte de acidentes pessoais para o operador.

A utilização do sensor indutivo também apresenta o inconveniente de não ser padronizada. Cada fabricante de central de injeção eletrônica utiliza seu próprio sensor e a variação de sensores pode ocorrer inclusive dentro de um mesmo fabricante para modelos diferentes de automóveis.

1.2 Método proposto – Vibração

Vibração é o movimento oscilatório de um corpo em torno de um ponto de referência. A oscilação pode ser periódica, como num pêndulo, ou ser aleatória, como o movimento de uma roda em uma estrada não pavimentada.

Em alguns casos a vibração é um efeito desejável, como ocorre nas cordas e palhetas de instrumentos musicais e nos alto-falantes, mas em geral a vibração é um efeito não desejável, pois equivale a desperdício de energia e está associada à geração de ruído sonoro além de ser um dos fatores que levam a fadiga das peças móveis numa máquina. A vibração indesejada pode ser causada por desbalanceamento nos componentes móveis de uma máquina ou por atrito em superfícies irregulares.

Nas máquinas com partes móveis sempre há a presença de vibração, inerente ao seu funcionamento, sendo que a frequência de vibração é relacionada à frequência do movimento da máquina. Valendo-se deste princípio pode-se medir a frequência de rotação de um motor a explosão por sua vibração.

Para a medida da rotação do motor através da vibração basta fixar um sensor de vibração ao bloco do motor, não sendo necessário o acesso ao eixo motriz, o que constitui um método pouco invasivo.

Natureza da vibração dos motores

Para estabelecer a relação entre vibração supõe-se que vibração inerente ao funcionamento do motor tem como origem a realização de trabalho pelo motor na forma de impulsos de torque no eixo motriz.

A súbita expansão dos gases causada pela combustão do combustível força o pistão para baixo e aplica um impulso de torque ao eixo de manivelas (virabrequim), que é responsável por transmitir o movimento do motor.

Partindo-se dessa suposição desenvolve-se a relação entre rotação e frequência de vibração: a fase ativa em cada pistão ocorre uma vez por volta, no caso de motores de dois tempos e uma vez a cada duas voltas, para o caso de motor de quatro tempos.

Portanto para o caso do motor de dois tempos a frequência de vibração é dada por:

$$f_{vibração(2T)(Hz)} = \frac{f_{rotação(2T)(RPM)}}{60} \times n_{cilindros}$$

Onde $f_{vibração(2T)(Hz)}$ é a frequência de vibração predominante em *Hertz*, $f_{rotação(2T)(RPM)}$ é a rotação do eixo em *RPM* e $n_{cilindros}$ é o número de cilindros do motor.

Isolando-se o termo da rotação do motor tem-se:

$$f_{rotação(2T)(RPM)} = 60 \times \frac{f_{vibração(2T)(Hz)}}{n_{cilindros}}$$

No motor de quatro tempos ocorre um impulso a cada duas voltas do eixo, portanto para obter-se a relação entre a frequência de vibração e a rotação aplica-se um fator de 2 na equação obtida anteriormente:

$$f_{rotação(4T)(RPM)} = 60 \times \frac{2 \times f_{vibração(4T)(Hz)}}{n_{cilindros}}$$

A vibração de natureza rotacional pode ser observada nos eixos lineares tangentes ao movimento circular ou em qualquer combinação linear destes eixos (x e z na Figura 2). No eixo y, na figura 2, nenhum movimento causado pela vibração deve ser detectado, pois o eixo é coincidente ao eixo do movimento circular.

Além da frequência fundamental causada pelos impulsos de torque, outras frequências de vibração podem, eventualmente, ser detectadas devido ao desbalanceamento do conjunto das partes móveis do motor (pistões, bielas, virabrequim, válvulas, etc.) e pode haver frequências harmônicas dependendo da fixação do motor ao chassi do automóvel.

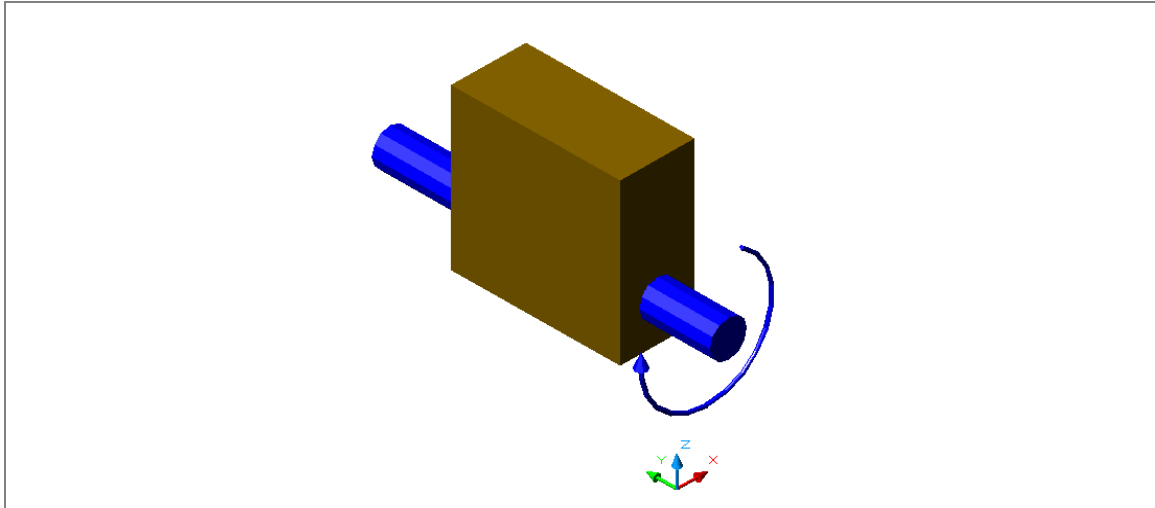


Figura 2: Diagrama de eixos no bloco do motor.

A relação entre rotação e vibração desenvolvida é uma relação entre frequências, portanto para a implementação do algoritmo foram escolhidas técnicas de processamento digital no domínio da frequência.

No capítulo 2 há uma introdução teórica a respeito do algoritmo para transformar sinais do domínio do tempo no domínio da frequência, assim como alguns aspectos práticos para implementação em processadores digitais de sinais.

No capítulo 3 apresenta-se uma série de especificações as quais o equipamento desenvolvido deve atender

Nos capítulos 4 e 5 são apresentadas as implementações adotadas para hardware e software embarcado de modo a atender as especificações apresentadas no capítulo 3.

2 Introdução teórica

Para o desenvolvimento do trabalho são necessários conhecimentos sobre processamento digital de sinais: representação de números, algoritmo para análise em frequência e aspectos práticos para a implementação.

2.1 Representação de números com ponto fixo e ponto flutuante

Números representados em ponto fixo não são necessariamente números inteiros. Os processadores de ponto fixo representam os números tanto em formato fracionário, usado principalmente em processamento de sinais, ou em formato inteiro, usado principalmente para operações de controle e endereçamento do processador.

O formato inteiro representa números de zero ao maior número que pode ser representado pela largura de bits da palavra e é normalmente utilizada a representação por complemento de dois com o sinal no bit mais significativo.

O formato fracionário representa a faixa de números $[-1,1)$, sendo que o bit mais significativo é o bit de sinal. Neste formato o ponto decimal está implícito logo depois do bit mais significativo. A Figura 3 exemplifica as representações de número inteiro e número fracionário para uma palavra de 8 bits.

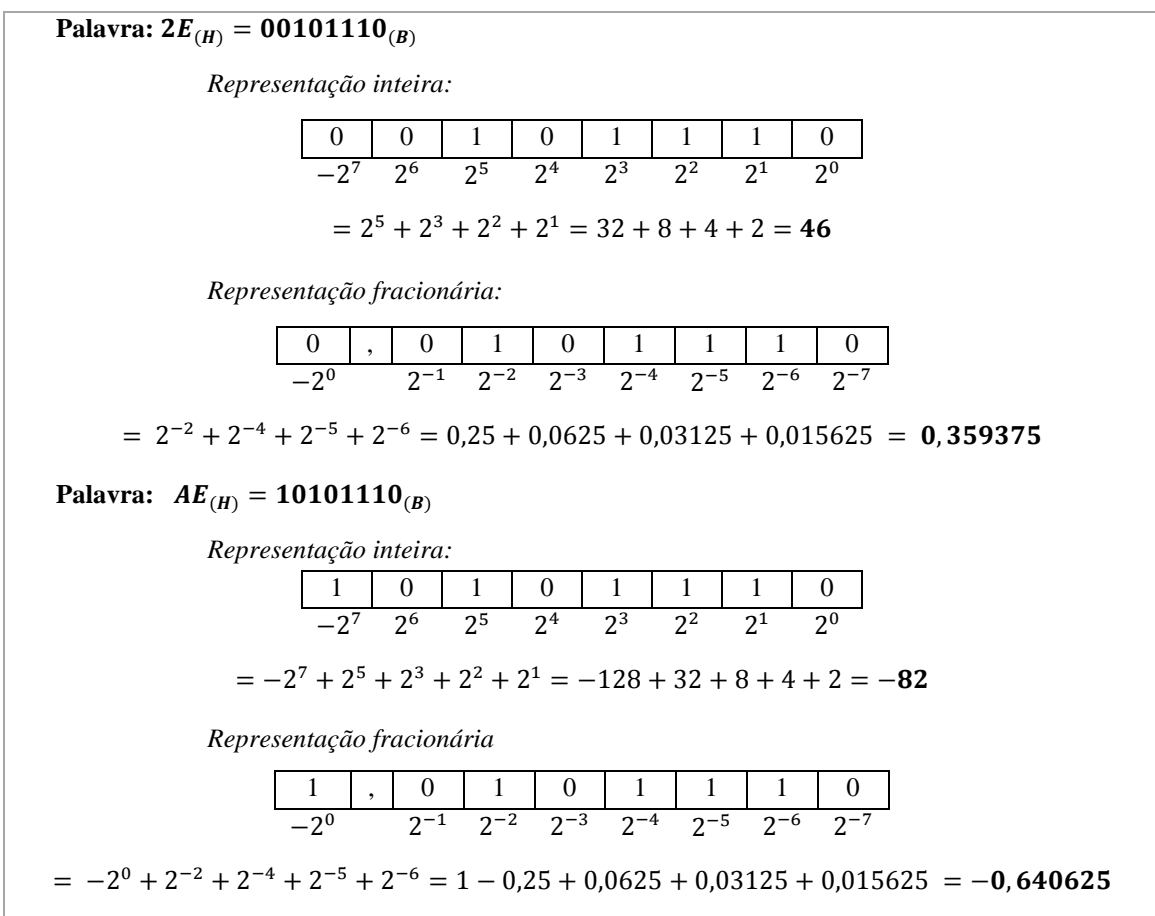


Figura 3: Representação numérica em ponto fixo

A representação em ponto flutuante é feita através de dois componentes: a mantissa e um expoente. A mantissa é geralmente um valor fracionário, similar a um número de ponto fixo. O expoente é um número inteiro que representa o número de deslocamentos do ponto decimal para obter-se o número representado. A Figura 4 ilustra um exemplo de número de ponto flutuante de 12 bits. O ponto decimal da mantissa está implícito depois do segundo bit mais significativo.

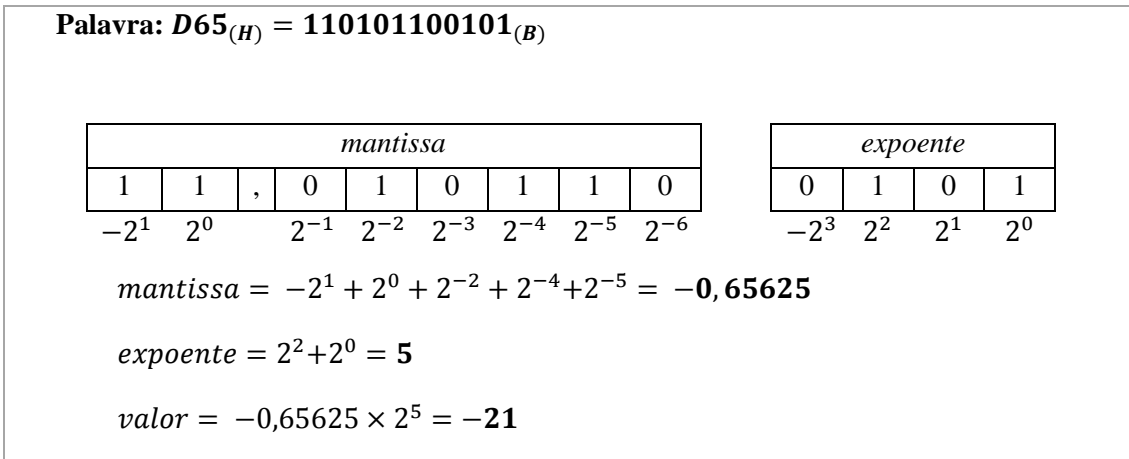


Figura 4: Representação numérica em ponto flutuante

Precisão e faixa dinâmica

Dois meios para medir o desempenho das representações em ponto fixo ou ponto flutuante são a faixa dinâmica e a precisão.

A precisão define a resolução da representação do sinal; é medida pelo peso do bit menos significativo do número fracionário. Na representação de ponto fixo a precisão é dada pelo tamanho da palavra, já na representação por ponto flutuante é o tamanho da mantissa que determina a precisão do número.

Na representação de ponto flutuante a precisão é a mínima diferença entre dois números de mesmo expoente. Uma vantagem desta representação é que o hardware escala automaticamente os números para que usem toda a faixa disponível na mantissa: se o número é muito grande o hardware escala-o deslocando os bits da mantissa para a direita e se o número não ocupa todos os bits da mantissa o hardware desloca os bits para a esquerda. O expoente armazena os deslocamentos ocorridos em ambas as direções.

A faixa dinâmica é a razão entre o menor e o maior valor que pode ser representado. A faixa dinâmica de uma representação em ponto flutuante é determinada pelo tamanho do expoente. Para o mesmo tamanho de palavra, a faixa dinâmica é sempre maior para representação em ponto flutuante. Por outro lado para o mesmo tamanho de palavra a representação em ponto fixo tem maior precisão.

A maior faixa dinâmica dos processadores de ponto flutuante tem seu preço: custam mais caro e dissipam mais energia do que os processadores de ponto fixo devido ao fato do seu hardware mais volumoso para a implementação das operações aritméticas.

2.2 Transformada de Fourier

A transformada de Fourier é uma transformada que leva uma função a uma função de base senoidal, ou seja, transforma uma função em uma soma ou integral de funções senoidais multiplicadas por coeficientes de amplitude.

Transformada contínua de Fourier

Dada uma função contínua no tempo $f(t)$, aplicando-se a transformada de Fourier obtém-se uma função de amplitudes complexas no domínio da frequência através da relação:

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

Transformada discreta de Fourier

Também conhecida por DFT (Discrete Fourier Transform). Para uso em computadores e processamento digital de sinais, os sinais são discretizados (amostrados), portanto deve-se utilizar uma versão discreta da transformada. Dado um sinal discreto $x[n]$, obtém-se as amplitudes complexas $X[k]$ no domínio da frequência discreta k através da relação:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk}, k = 0, \dots, N-1$$
$$W_N^{nk} = e^{-\frac{j2\pi}{N}nk}$$

A implementação direta, tal como descrita acima, não é muito utilizada devido a sua complexidade de N^2 multiplicações e N^2 somas. Manejando-se a equação da transformada é possível implementá-la com menor custo computacional. Essas implementações são chamadas de transformadas rápidas de Fourier ou mais comumente de FFT (Fast Fourier Transform). O algoritmo FFT mais utilizado é o Cooley-Tukey, que é brevemente explicado a seguir.

Algoritmo FFT

O algoritmo FFT *Cooley-Tukey* baseia-se em desdobramentos sucessivos, onde uma DFT de N pontos é decomposta por DFTs com menos pontos. A implementação mais utilizada é radix-2, onde a DFT de N pontos é reduzida a DFTs de tamanho 2 o que limita a implementação desse algoritmo a sinais cujo tamanho seja uma potência de 2.

A implementação do algoritmo por decimação no tempo separa o sinal de entrada em duas partes, reduzindo a DFT de N pontos para duas DFTs de $N/2$ pontos. As duas DFTs de $N/2$ pontos são divididas em quatro DFTs de $N/4$ pontos e assim por diante até que se chega a DFTs de dois pontos. Devido à decimação feita em cada etapa do algoritmo o sinal de entrada $x[n]$ deve ser modificado de tal modo que os bits dos índices n fiquem na ordem reversa.

A decimação também pode ser feita em frequência, neste caso o sinal de entrada tem a ordem original e a saída é organizada com os bits do índice em ordem reversa.

As figuras 5, 6, 7, 8 e 9 ilustram os passos para a dedução do algoritmo de uma FFT de um sinal $x[n]$ de 8 pontos com decimação no tempo. As deduções e provas são demonstrados por HAYES,1999.

Primeiro reduz-se a DFT de 8 pontos para duas DFTs de 4 pontos:

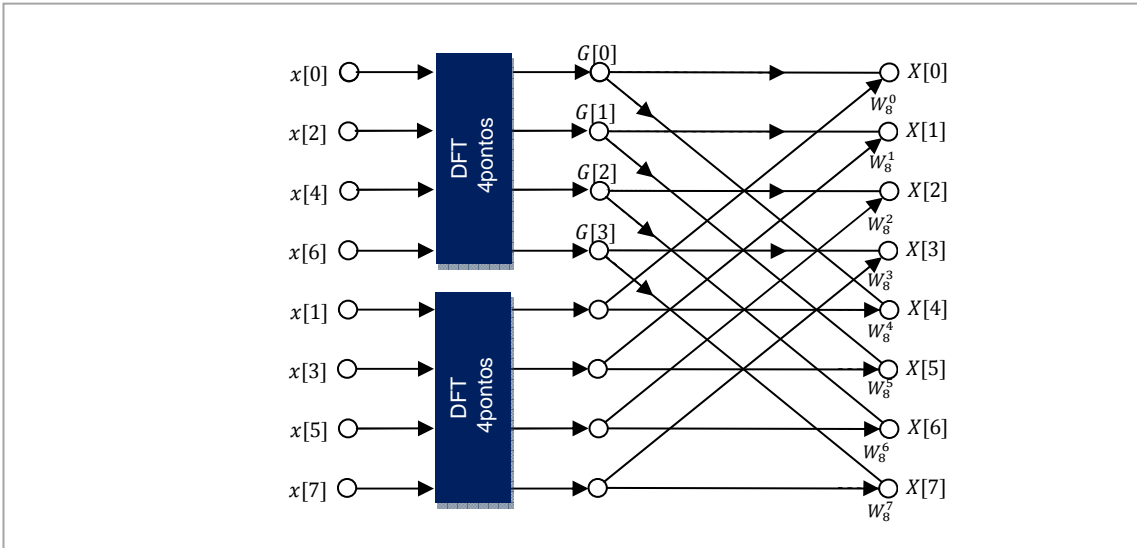


Figura 5: Redução da DFT de 8 pontos para 4 pontos

A seguir, as DFTs de 4 pontos são separadas em DFTs de 2 pontos:

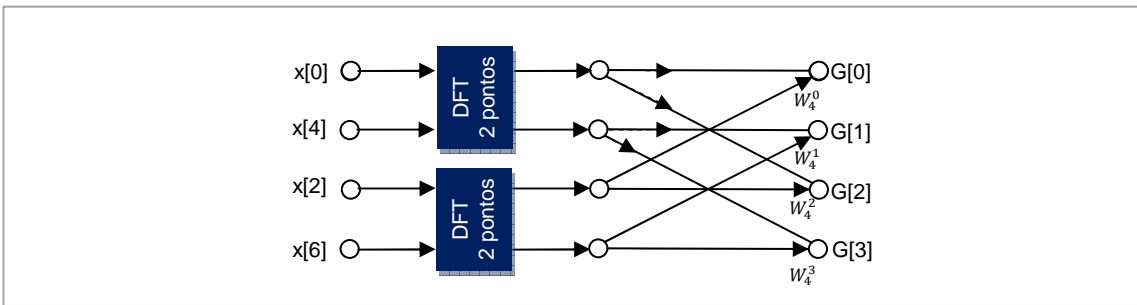


Figura 6: Redução da DFT de 4 pontos para 2 pontos

Por fim, abre-se cada DFT de 2 pontos:

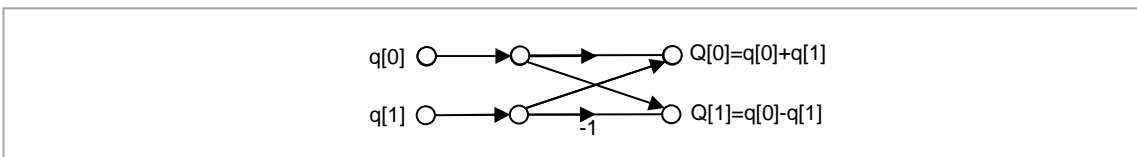


Figura 7: DFT de 2 pontos

Cada uma das etapas descritas de separação podem ser representadas por operações semelhantes, chamada de borboleta, devido ao seu formato no diagrama:

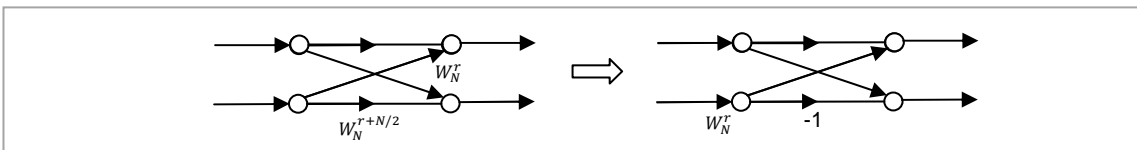


Figura 8: Estágio borboleta do algoritmo FFT

Reescrevendo o diagrama do algoritmo substituindo as DFTs por estágios borboleta:

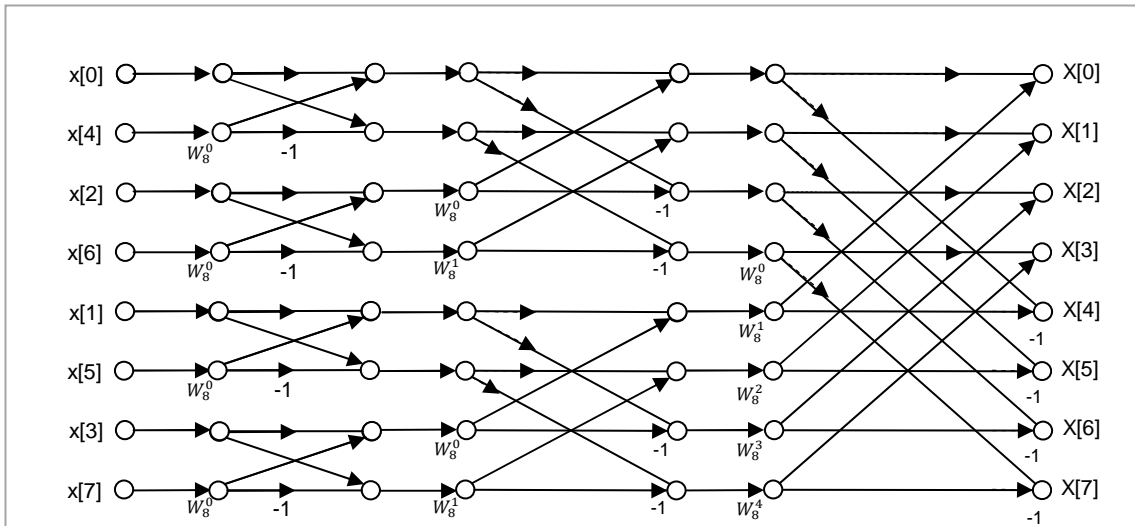


Figura 9: Diagrama completo do algoritmo FFT de 8 pontos com decimação no tempo

Deve-se observar a reorganização do sinal de entrada para a utilização deste algoritmo, os bits do índice do vetor são invertidos. Essa reorganização é ilustrada na tabela a seguir:

Tabela 1: Reorganização de um vetor de 8 posições por inversão da ordem dos bits de índice

| n | n(binário) | n'(binário inverso) | n' |
|---|------------|---------------------|----|
| 0 | 000 | 000 | 0 |
| 1 | 001 | 100 | 4 |
| 2 | 010 | 010 | 2 |
| 3 | 011 | 110 | 6 |
| 4 | 100 | 001 | 1 |
| 5 | 101 | 101 | 5 |
| 6 | 110 | 011 | 3 |
| 7 | 111 | 111 | 7 |

Este algoritmo pode ser implementado também com a decimação em frequência, onde o vetor de sinal entra em ordem e é a saída que tem os bits do índice invertidos. Como demonstrado por HAYES, 1999 o algoritmo com decimação em frequência é ilustrado pelo diagrama da Figura 10.

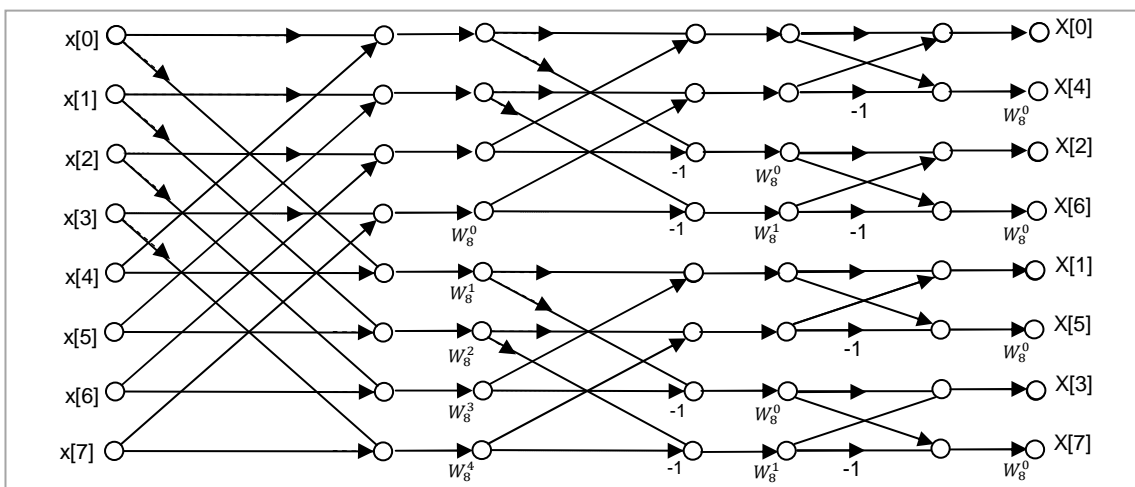


Figura 10: Diagrama completo do algoritmo FFT de 8 pontos com decimação em frequência

Vantagens do algoritmo FFT para implementação

A principal vantagem deste algoritmo é o custo computacional bem inferior a aplicação direta de DFTs. A complexidade é de $N/2 \log_2^N$ multiplicações complexas e $N \log_2^N$ somas complexas contra N^2 multiplicações complexas e N^2 somas complexas da implementação direta da DFT, consistindo num grande ganho computacional, principalmente para grandes valores de N .

Outra característica interessante é que após um estágio *borboleta* o sinal não precisa mais ser guardado e o resultado do estágio pode ser guardado no próprio registrador do sinal de entrada, essa característica é chamada de cálculo *in place*. A desvantagem do cálculo *in place* da FFT é que o sinal de entrada (ou saída) precisa ser reorganizado com os bits de endereçamento na ordem inversa. Para lidar com isso alguns dos processadores digitais de sinais possuem a característica de endereçamento *bit reverse* que facilita muito essa implementação: não é necessário efetivamente reorganizar os dados, o processador já trata os endereços com os bits na ordem invertida.

Consideração sobre a implementação em processamento de ponto fixo

Na aritmética de ponto fixo, o resultado das somas pode ter um bit a mais que os operandos e portanto em cada estágio borboleta a largura do dado pode aumentar, o que leva à saturação dos acumuladores de cálculo. Uma técnica utilizada para contornar este problema é rotacionar todos os elementos um bit à direita a cada estágio.

Este artifício, porém reduz a faixa dinâmica dos sinais e leva os sinais de saída a ter um fator de escala de $1/N$. Para se ter uma faixa dinâmica maior sem aumentar o número de bits pode-se empregar uma implementação FFT chamada de *Block floating point FFT*, que verifica os valores dos sinais entre cada estágio da FFT e aplica um fator de escala de modo que sejam preenchidos todos os bits possíveis para que não haja saturação nos cálculos do estágio. Esta implementação tem o custo computacional de analisar os valores dos sinais, porém aumenta significativamente a faixa dinâmica dos sinais analisados.

3 Especificações do equipamento

Para o desenvolvimento de um equipamento comercial para a medida de rotação determinaram-se algumas especificações, listadas a seguir.

Faixas de medida e tipos de motores

O equipamento deve ser capaz de medir rotações de marcha lenta até 9999 RPM, de motores de 3 a 12 cilindros nas configurações em linha, em V e *boxer*.

Comunicação serial UART

O equipamento deve possuir uma interface serial UART para comunicação com outros equipamentos de diagnóstico que possuam este tipo de interface. Os dados são enviados sem nenhum tipo de *handshaking* e deve ser usada a codificação *ASCII*. No protocolo a ser seguido deve-se enviar continuamente a *string*:

| | | | | | | | | | | |
|-----|------|------|------|------|------|-----|---|---|---|----|
| 'G' | rot4 | rot3 | rot2 | rot1 | rot0 | 'T' | x | x | x | CR |
|-----|------|------|------|------|------|-----|---|---|---|----|

- 'G' e 'T' – Caracteres G e T, respectivamente
- rot4 a rot 0 representam os dígitos da medida da rotação em RPM
- x – caracteres não utilizados
- CR – caractere especial *Carrier Return*.

Além da especificação acima, foi adicionada como especificação um modo *debug* que quando ativado utiliza a comunicação UART para transmitir os dados adquiridos a um computador pessoal para visualização e armazenamento dos sinais.

Saída pulso

A comunicação digital por pulso é um padrão utilizado para comunicação com outros equipamentos de diagnóstico automotivo que fazem uso da informação da rotação do motor.

Trata-se de um pulso simétrico de 5V a -5V com largura t de 5 ms e período T equivalente a duas vezes o período de rotação do motor. Na Figura 11 observa-se como deve ser o sinal no domínio do tempo.

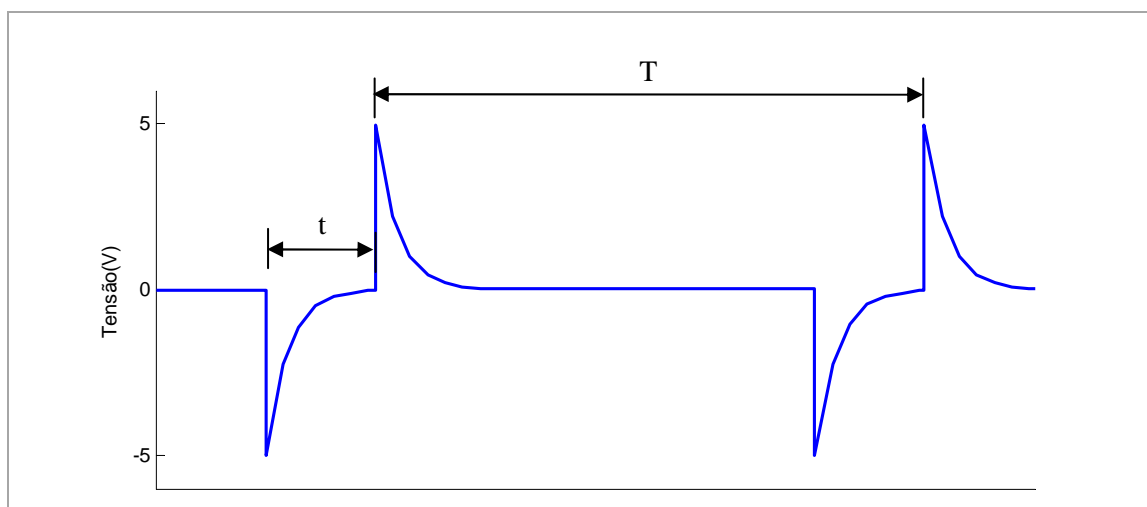


Figura 11: Saída tipo pulso

Interface com usuário

O equipamento deve contar com as seguintes interfaces com o usuário:

- *Display* LCD onde é mostrado o valor da rotação medida em RPM e os parâmetros de configuração do equipamento. Quando em modo de configuração o display deve mostrar claramente qual dos parâmetros está sendo configurado
- *LED* indicador de medida – ligado à saída pulso para que haja uma verificação visual por parte do usuário quando o equipamento está em modo de medida. Quando em modo de configuração, o *LED* deve permanecer apagado.
- *Chaves* para navegação e configuração de parâmetros. São utilizadas em conjunto com o *display* para a navegação e configuração dos parâmetros do equipamento.

4 Desenvolvimento de hardware

4.1 Arquitetura

A arquitetura do hardware implementado consiste num processador com interface analógica onde é ligado o sensor de vibração e uma interface digital onde são ligados os elementos de interface com o usuário. O diagrama da Figura 12 ilustra a arquitetura do hardware implementado. Cada um dos elementos é explicado a seguir.

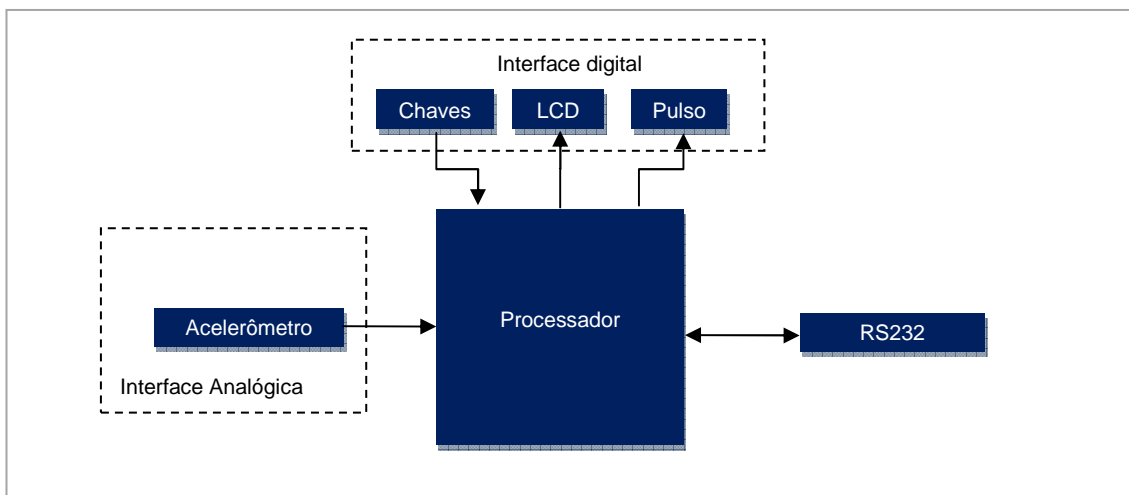


Figura 12: Arquitetura do hardware

4.2 Processador

O processador foi escolhido levando-se em conta as tarefas que deve desempenhar: conversão analógica/digital, interface de comunicação serial (SPI e UART), número de entradas e saídas suficientes para a implementação de interface com o usuário (botões e display LCD) e capacidade para implementação de técnicas de processamento digital de sinais. Além das tarefas a desempenhar deve-se levar em conta disponibilidade no mercado, preço, ferramentas de desenvolvimento disponíveis e suporte.

O processador escolhido foi da família *dsPIC33FJ128MC706*. Trata-se de um microcontrolador com recursos DSP e tem como características relevantes para o projeto:

- Tensão de alimentação 3,3V
- Recursos DSP
 - ALU com barramento de dados duplo
 - *Instruction set* com diversas instruções de ciclo único, como deslocamento de múltiplos bits, multiplicação, *MAC* e outros.
- Acesso direto à memória pelos periféricos (DMA)
- Memória de programa de 128kB e memória de dados com 16kB incluindo 2kB de DMA
- 53 pinos de entrada/saída tolerantes a 5V e possibilidade de dreno-aberto.
- 2 conversores analógico digitais com 4 canais de amostragem cada.

- Módulos de comunicação SPI e UART.

Estes periféricos e especificações também estão presentes em outras famílias de *DSC* da *Microchip* que são pino a pino compatíveis e podem ser utilizados sem problema no caso da falta em mercado do modelo escolhido.

4.3 Acelerômetro

Para a medida de vibração de um corpo usa-se medir sua aceleração. Os acelerômetros são constituídos por uma massa sísmica e um meio de medir a força aplicada por ela quando lhe é imposta alguma aceleração.

O acelerômetro mais comum é do tipo piezelétrico, onde a força aplicada pela massa sísmica é medida por piezeletricidade, porém este tipo de acelerômetro tem como limitação a faixa de frequência e a necessidade de utilizar um amplificador de carga casado com o acelerômetro, o que encarece sua aplicação.

A alternativa a esse tipo de acelerômetro é a utilização de acelerômetros integrados que utilizam tecnologia *MEM* (*microelectromechanical*). A massa sísmica, o transdutor utilizado para a medida de força e os circuitos de condicionamento de sinais são encapsulados em um único circuito integrado. O acelerômetro escolhido para este projeto é o *MMA7620*, fabricado por *Freescale Semiconductor*.

Trata-se de um acelerômetro tridimensional onde se pode medir a aceleração em 3 eixos dispostos ortogonalmente, com saída analógica, alimentação de 3,3V e sensibilidade ajustável de 1,5g/2g/4g/6g.

Como a saída de tensão fica sempre na faixa de 0 a 3,3V o acelerômetro pode ser ligado diretamente às entradas analógicas do processador sem necessidade de conversão de nível de tensão. Deve-se utilizar, porém um filtro passa-baixas para *anti-aliasing* na conversão analógico-digital.

4.4 Chaves

Como interface para a entrada dos dados e escolha de parâmetros pelo usuário, foi implementado um conjunto de chaves tácteis, como a da figura 13, que suportam elevado número de manobras.



Figura 13: Chave táctil

A chave é do tipo normal aberta e deve ser ligada a entrada digital do processador por meio de *pull-up* ou *pull-down*. Neste projeto é utilizado o método com *pull-up* já que o processador escolhido possui algumas entradas digitais com *pull-up* interno, que reduz o número de componentes do hardware. Estas entradas também têm como característica a geração

de interrupção por troca de estado facilitando a implementação do software de tratamento para as chaves.

Em paralelo com a chave é utilizado um capacitor, que junto à resistência de *pull-up* constitui um filtro passa-baixas. Esta é uma técnica de anti-ripique (*debouncing*) que visa suprimir a oscilação da tensão de saída durante o fechamento ou abertura do contato mecânico da chave.

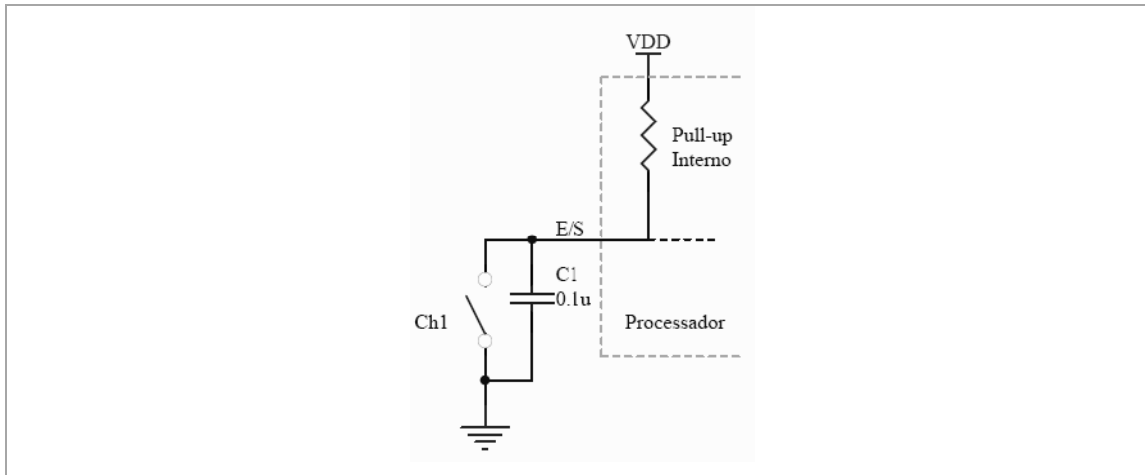


Figura 14: Interface para as chaves

4.5 Módulo LCD

Para interface de saída para usuário foi escolhido um módulo LCD tipo HD44780. O módulo é constituído por um mostrador tipo LCD com matrizes de caractere 5x10 dispostos em 8, 16 ou 20 caracteres por 1, 2 ou 4 linhas. O módulo escolhido é o de 16 caracteres com 2 linhas pois é suficiente para atender a especificação de interface com o usuário e tem maior disponibilidade no mercado que os outros formatos. Um exemplo de módulo LCD pode ser visto na figura 15.

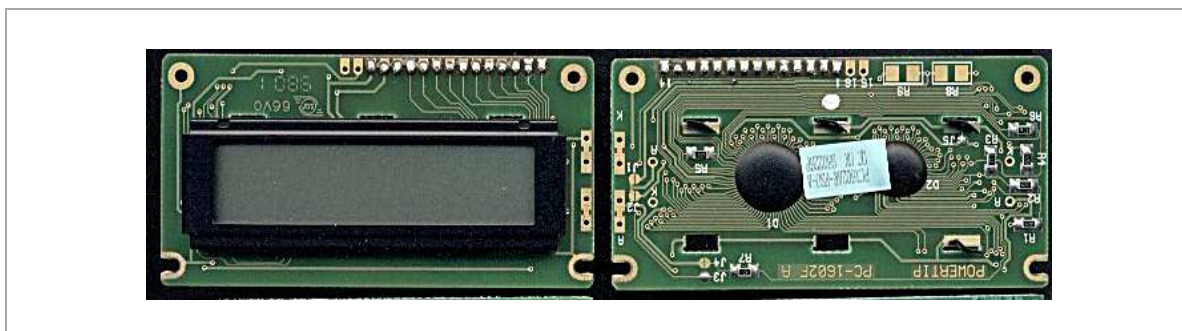


Figura 15: Módulo LCD HD44780

Funcionamento

No módulo há um sistema microprocessado responsável pelo controle dos caracteres através de um barramento de 8 bits e 3 sinais de controle. Além dos Sinais de dados e controle encontram-se no módulo os pinos de alimentação, um pino para ajuste do contraste do LCD e, caso haja *backlight*, os pinos dos LEDs do *backlight*.

O barramento tem a função de receber os caracteres em formato ASCII e também tem a função de receber os comandos de controle do módulo (limpar tela, posicionar cursor, etc). Para distinguir a natureza dos dados no barramento é utilizado um dos pinos de controle, *RS*. Para que cada dado no barramento seja processado no módulo é necessário um pulso num outro sinal de controle, *E*. O terceiro sinal de controle, *R/W*, determina a direção do barramento para possibilitar a leitura de status do módulo. Neste projeto não é utilizado o recurso de leitura no módulo, sendo portanto utilizado somente o modo de escrita.

Para economia de pinos E/S do processador há a possibilidade de utilizar somente 4 bits do barramento de dados, dividindo os dados em *nibbles* e enviando um de cada vez.

Implementação

O módulo LCD tem alimentação em 5V e para que possa ser ligado ao processador deve haver conversão dos sinais de 3,3V para 5V. A conversão de nível neste caso é feita utilizando a opção de E/S do processador em dreno aberto com *pull-up* em 5V. A implementação do módulo LCD é ilustrada na Figura 16.

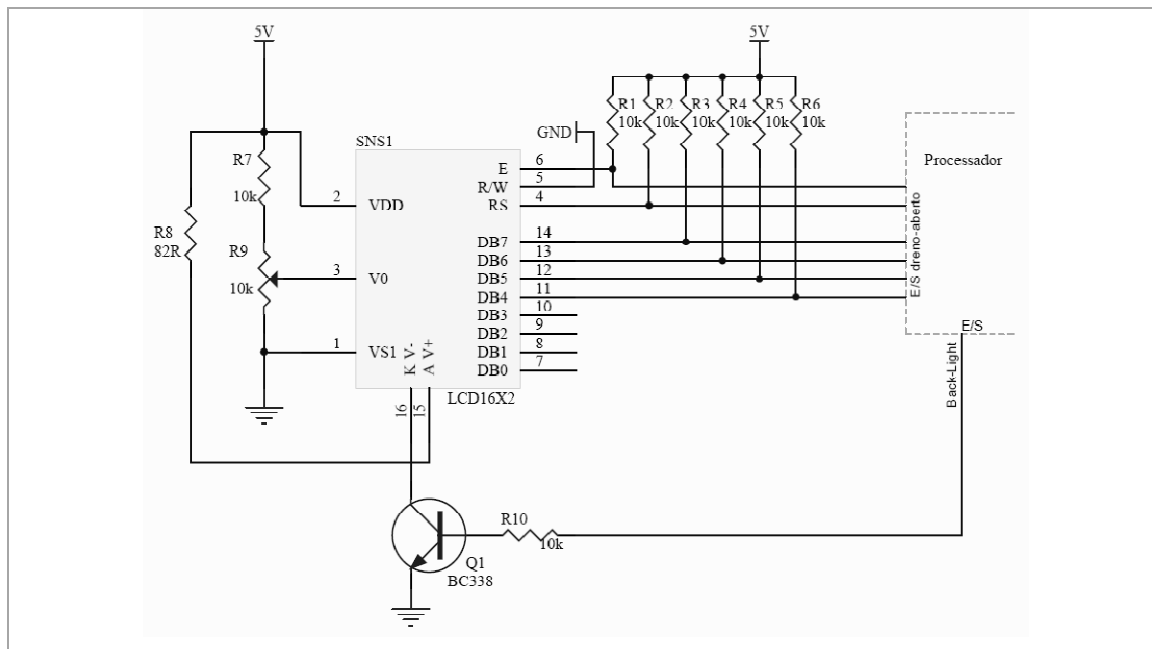


Figura 16: Interface com módulo LCD

4.6 Pulso

Dada a especificação da saída pulso utiliza-se a carga e descarga de um capacitor comandado por pulsos retangulares, que são facilmente gerados numa saída do processador.

O circuito adotado para a saída é uma configuração *push-pull* com deslocamento de nível. O par *push-pull* foi adotado para uma baixa impedância de saída e deve-se utilizar um conversor de nível pois a saída do microcontrolador é de 3,3V e o estágio *push-pull* opera em 5V. O diagrama esquemático do estágio de saída tipo pulso e os sinais em cada estágio são mostrados na Figura 17 e na Figura 18, respectivamente. Deve-se observar o valor da resistência da base de Q1 para que a corrente não ultrapasse o limite de saída do processador.

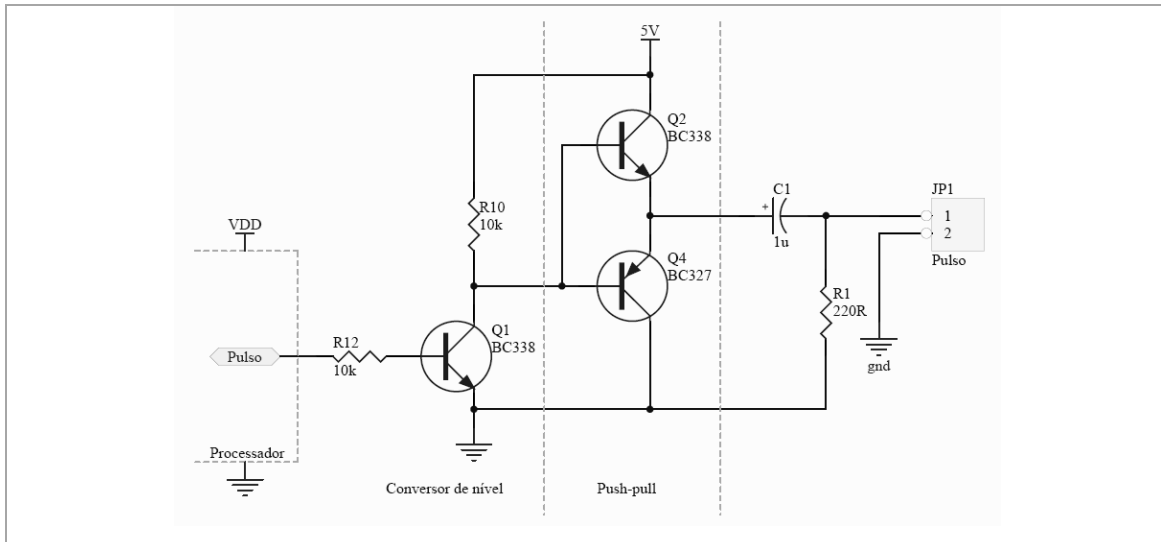


Figura 17: Módulo para saída pulso

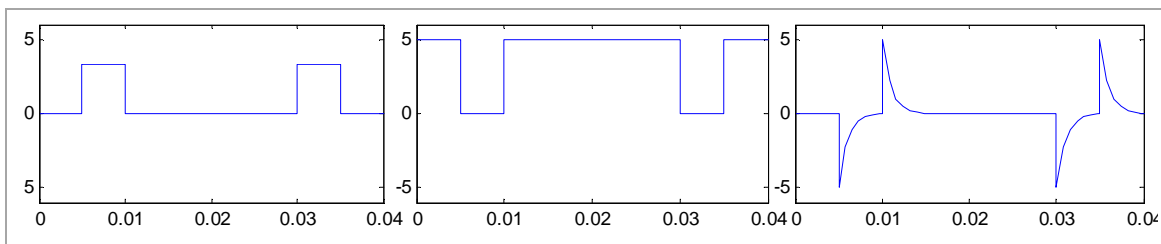


Figura 18: Sinais do módulo de saída pulso

4.7 RS-232

O conjunto de especificações RS232 é um dos padrões recomendados para comunicação digital entre equipamentos. Vem sendo substituído pelo padrão USB, mas ainda é muito utilizado devido à compatibilidade com antigos equipamentos e à simplicidade na implementação tanto de hardware quanto de software.

No padrão RS-232 o dado é enviado como uma série de bits. De acordo com a norma a comunicação pode ser tanto síncrona como assíncrona. A norma também define as características elétricas dos sinais (tensão, *slew-rate*, curto-circuito) e ainda uma série de sinais de controle para gerir a comunicação entre os equipamentos (*handshaking*). A norma não define codificação de caracteres (ASCII, EBCDIC, etc..)

Características elétricas

A faixa de tensão permitida é de -15V a -3V ou de 3V a 15V. A faixa ao redor do zero não caracteriza um sinal válido neste padrão. O nível lógico um é representado por uma tensão negativa e o nível lógico zero é representado por uma tensão positiva.

Os *transceivers* constituem uma forma de isolamento protegendo o circuito contra transitórios na linha e curtos-circuitos e tem capacidade de corrente suficiente para cumprir a especificação de *slew-rate*. O *transceiver* escolhido foi o MAX3232, que pode operar com alimentação simples de 3,3V, possui conversor *charge-pump* que funciona com capacitores pequenos para montagem compacta e possui dois canais completos (*Tx/Rx*) para comunicação.

As conexões do *transceiver* podem ser vistas na figura 19, conforme indicado na folha de dados do fabricante

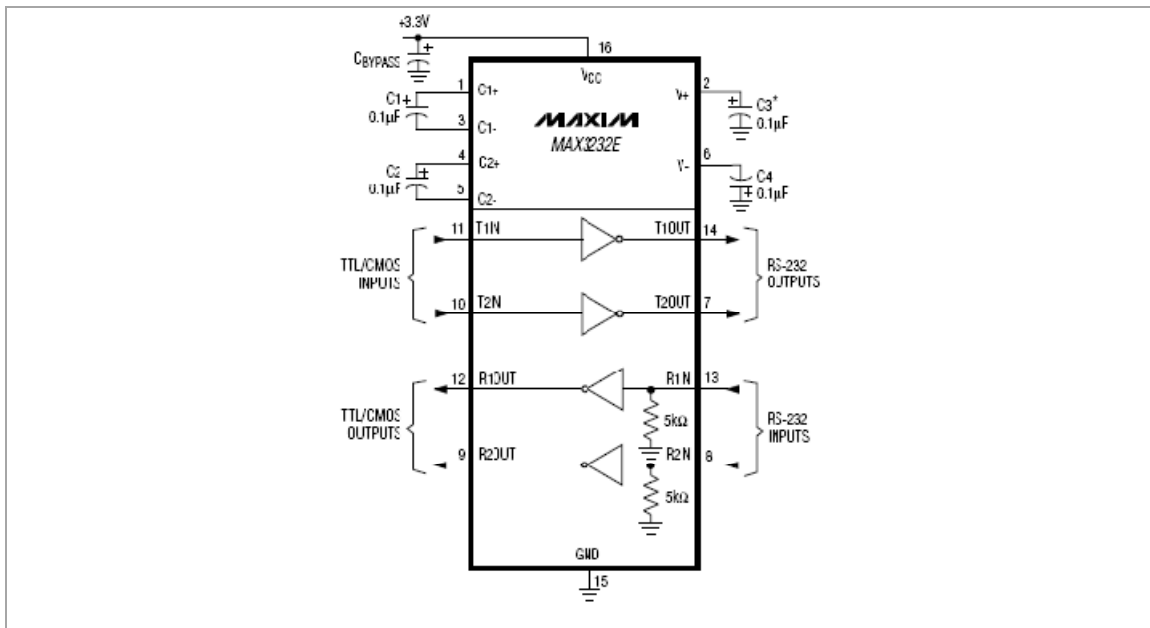


Figura 19: Conexão do transceiver RS232

O modo de comunicação utilizado é assíncrono, sem *handshaking*. A transmissão e recepção dos dados e gestão do *baud-rate* ficam a cargo do módulo UART do processador, conectado ao *transceiver*. A figura 20 mostra o diagrama da implementação do módulo de comunicação RS232.

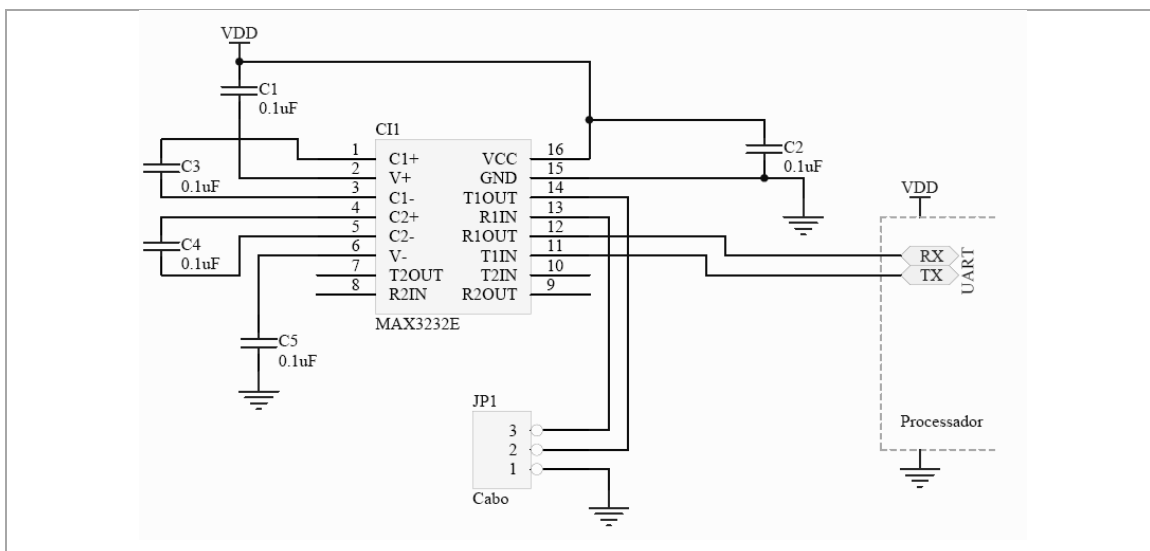


Figura 20: Módulo de comunicação RS232

4.8 Diagrama esquemático e layout

O hardware foi todo implementado utilizando o pacote *EDA (Electronic Design Automation) Altium Designer*. Este pacote inclui ferramentas para desenho de diagrama esquemático e *layout* da placa de circuito impresso.

Design flow

O *design flow* básico consiste no desenho do diagrama esquemático, contendo todos os componentes do sistema. Feito o diagrama é gerada a *netlist*, documento contendo a informação dos componentes (identificação, tipo e conexões). Com a *netlist* pode-se começar o desenho do *layout* da placa de circuito impresso.

A ferramenta de *layout* verifica continuamente as regras de desenho (*DRC – Design Rule Check*), definidas para o processo de fabricação escolhido como espessura mínima das trilhas, distância mínima entre trilhas de sinais diferentes, diâmetro de furos, etc. A ferramenta verifica também se o *layout* desenhado corresponde ao circuito do diagrama esquemático através da *netlist* gerada pelo diagrama.

Depois de finalizado o *layout* a ferramenta gera os arquivos de manufatura que são enviados a empresa responsável pela fabricação de placas. O hardware é considerado acabado depois que as placas são testadas. O teste consiste na implementação de software básico para cada um dos módulos descritos. Os erros mais comuns são ocasionados por falta de ligações ou ligações erradas. Como a verificação durante a elaboração do *layout* é baseada no diagrama esquemático, os erros dessa natureza devem-se a erros na elaboração do diagrama, é por esse motivo que normalmente há uma ou mais pessoas encarregadas de revisar o diagrama esquemático. O *layout* da placa desenvolvida para protótipo é mostrado na figura 21.

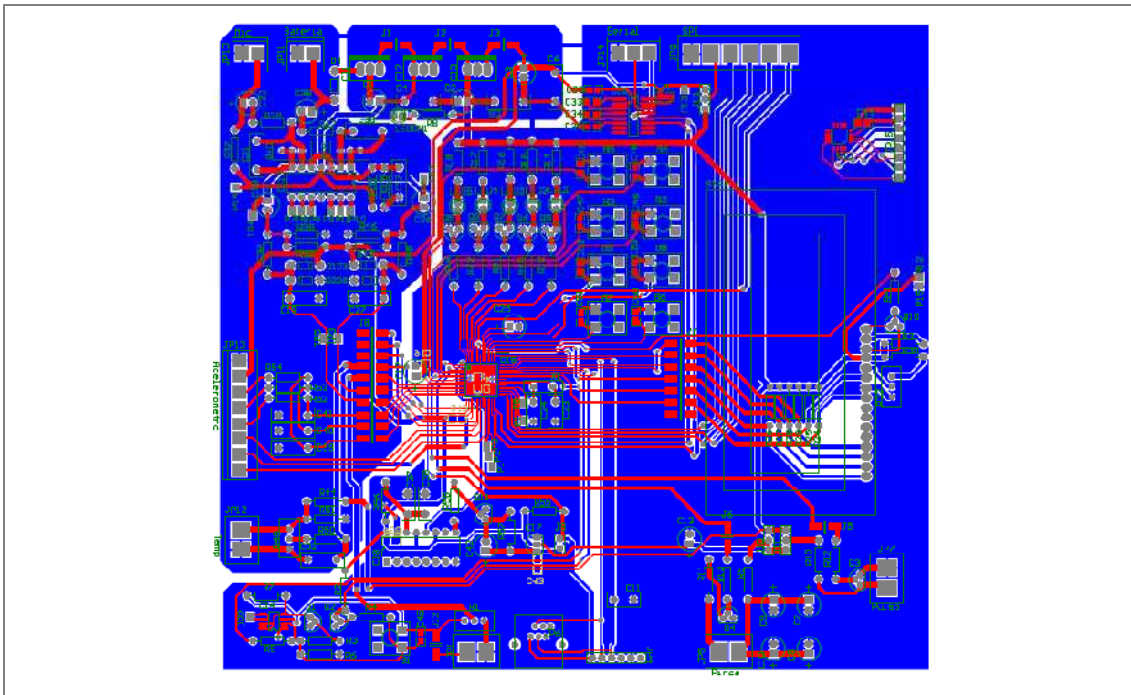


Figura 21: Layout PCI protótipo

5 Desenvolvimento do software

5.1 Ambiente de desenvolvimento

O ambiente de desenvolvimento utilizado é o *MPLAB*, fornecido pela *Microchip*, fabricante do processador escolhido. Este ambiente faz a interface entre um editor de código (editor de texto), compilador, *assembler*, *linker*, simulador e programador *in-circuit*. A linguagem padrão do ambiente é *assembly*, porém o fabricante disponibiliza compilador para linguagem C que pode ser instalado para funcionar junto ao ambiente *MPLAB*.

O pacote do ambiente de desenvolvimento possui uma biblioteca DSP com funções escritas em *assembly* que podem ser chamadas diretamente pelo código escrito em C.

5.2 Algoritmo geral de funcionamento

O algoritmo de funcionamento do equipamento consiste numa seqüência onde há um laço principal que é executado infinitamente. O fluxo do laço principal é controlado por variáveis globais de sincronização, que são controladas por rotinas de interrupção.

O sinal a ser processado é gerado automaticamente por interrupção e é passado à rotina de processamento através de uma variável global. O fluxograma do sistema, com as variáveis de dados e sincronização é mostrado na Figura 22.

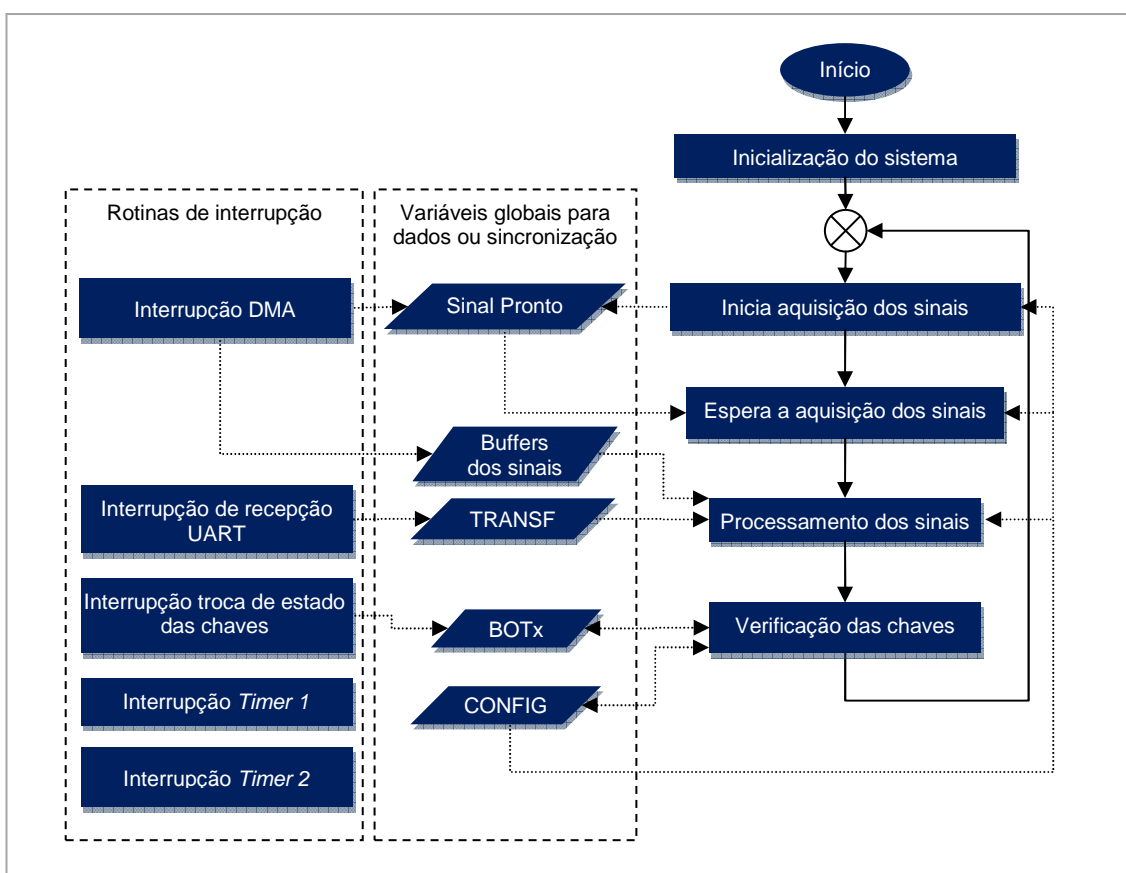


Figura 22: Fluxograma geral de funcionamento

Quando o equipamento é ligado todos os módulos são inicializados. A seguir o algoritmo entra no laço normal de funcionamento onde a medida da rotação é feita.

O *loop* principal de funcionamento liga o conversor analógico digital a seguir o fluxo fica paralisado, esperando que os buffers com os sinais de vibração sejam preenchidos por interrupção. Para sincronizar o preenchimento dos buffers de sinal e a liberação do fluxo há uma variável de sincronização (*Sinal Pronto*). Esta variável é ativada pela rotina de interrupção assim que os buffers de sinal estejam preenchidos.

A seguir, no fluxo principal, os sinais de aceleração, armazenados nos buffers de sinais, são processados e a rotação é determinada. Neste passo o *display* é atualizado com a rotação medida, a saída de pulso tem o período atualizado e envia-se a rotação pela interface RS232 conforme a especificação. Pode-se fazer a aquisição dos sinais e espectro da vibração para *debug* do equipamento, que é controlado pela variável de sincronização *TRANSF*. Esta variável é ativada por interrupção de *UART* somente quando uma instrução específica é enviada ao equipamento.

O próximo passo é processar as teclas, caso o usuário tenha pressionado alguma. Quando uma tecla muda de estado ocorre uma interrupção e rotina de interrupção verifica se a tecla foi pressionada ou solta e caso tenha sido pressionada uma variável de sincronização *BOTx*, onde *x* representa a tecla, é acionada. A rotina de verificação das chaves verifica as variáveis *BOTx* e de acordo com a tecla pressionada muda os estados do equipamento conforme descrito na seção 5.10. Quando o equipamento entra em modo de configuração uma variável *CONFIG* é ativada e faz com que as rotinas de aquisição e processamento de sinais sejam ignoradas.

As seções a seguir explicam detalhadamente os algoritmos e funções do equipamento.

5.3 Inicialização do sistema

Durante a inicialização é feita a configuração do oscilador e dos módulos do processador utilizados na implementação.

Configurar oscilador/PLL

O processador possui diversas opções de oscilador e foi escolhido utilizar o oscilador a cristal interno com o PLL *on chip* para alcançar alta taxa de cálculos nas rotinas de processamento dos sinais. À partida, o processador começa a funcionar na frequência do oscilador a cristal e depois o PLL é configurado por registradores internos via software.

A frequência do *clock* do sistema é configurada através de três divisores de frequência do PLL interno, sempre respeitando os limites de frequência em cada estágio do módulo, conforme o diagrama da Figura 23, obtida da folha de dados do fabricante.

Uma característica interessante do hardware implementado é que a frequência do clock pode ser modificada dinamicamente durante o funcionamento do sistema, tanto através do PLL quanto pela escolha de outro oscilador o que torna possível a redução do consumo de energia quando não é necessário alto desempenho.

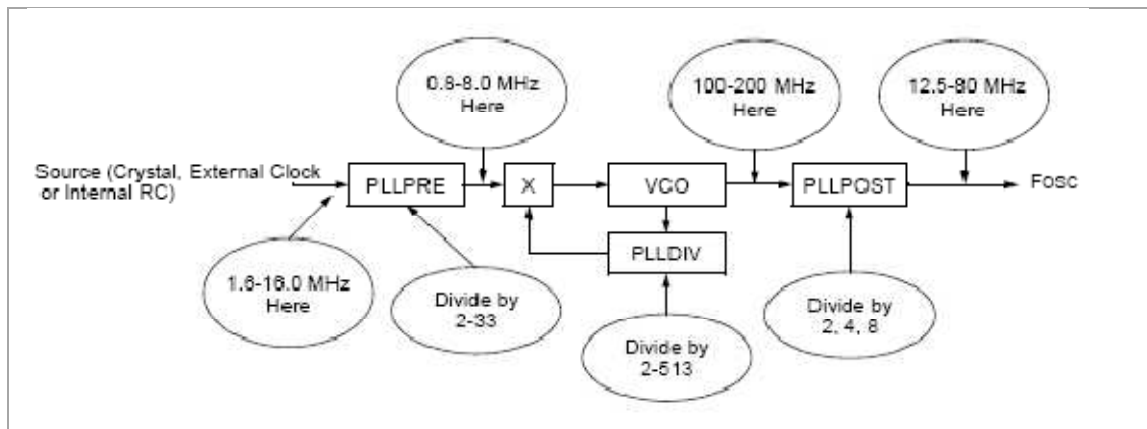


Figura 23: Diagrama do PLL

Inicialização da UART

A inicialização do módulo UART consiste em configurar os diversos registradores do módulo de acordo com as especificações:

- Divisor de frequência de acordo com o *baud-rate* da especificação
- tipo de dado: *8 bits*
- Stop bits: *1 stop bit*
- Controle de fluxo(*handshaking*): sem controle de fluxo
- Interrupção para recebimento de dados

Inicialização E/S e interrupção para as chaves

Os pinos onde são ligados os *LEDs* devem ser configurados como saída e os pinos ligados às chaves devem ser configurados como entradas com *pull-up*.

Em seguida habilita-se a interrupção por troca de estado. O tratamento das chaves é feito através da função de atendimento a interrupção e de uma função no *loop* principal.

Inicialização conversor analógico digital

A inicialização do conversor analógico digital consiste em:

- Configuração dos pinos analógicos como entradas analógicas
- *Sample/hold* automático através do *Timer3*
- Inicialização do *Timer 3* de acordo com a taxa de amostragem
- Scan automático das entradas analógicas – as entradas são amostradas e convertidas seqüencialmente de modo automático
- Modo *DMA* – As amostras são colocadas automaticamente numa área de memória para posterior processamento.
- Inicializar controlador *DMA*

Inicialização módulo do LCD

A inicialização do módulo LCD consiste em:

- Configurar pinos como saídas com dreno aberto (*open-drain*)
- Executar comandos de inicialização – modo de operação, limpeza da tela, etc.

5.4 Aquisição dos sinais

A aquisição dos dados é iniciada ligando-se o módulo de conversão AD.

Os sinais resultantes da conversão analógico-digital são transferidos para uma área temporária da memória por acesso direto (DMA). Quando este espaço temporário é todo preenchido ocorre uma interrupção de DMA.

Na interrupção por DMA os dados da área temporária são transferidos para o buffer principal e seu ponteiro é incrementado.

Quando o buffer de sinais está completo, verificado através do ponteiro, o conversor analógico-digital é desligado, portanto a aquisição dos sinais também é desligada e a variável de sincronização *Sinal Pronto* é ativada, liberando o fluxo no laço principal do sistema.

As rotinas que controlam a aquisição dos sinais são ilustradas no fluxograma da Figura 24.

5.5 Processamento dos sinais

A rotina do processamento dos sinais toma os buffers onde estão armazenados os sinais de vibração, que são passados a uma rotina de análise de frequência, que retorna a magnitude da maior componente de frequência e a frequência cuja componente possui maior magnitude. As frequências e magnitudes são armazenadas para serem comparadas entre si posteriormente.

Comparam-se então as três magnitudes, e a frequência que tem maior magnitude é passada adiante para as rotinas de atualização do display, da saída pulso e saída UART. O fluxograma da função de processamento de sinais é mostrado na figura 25.

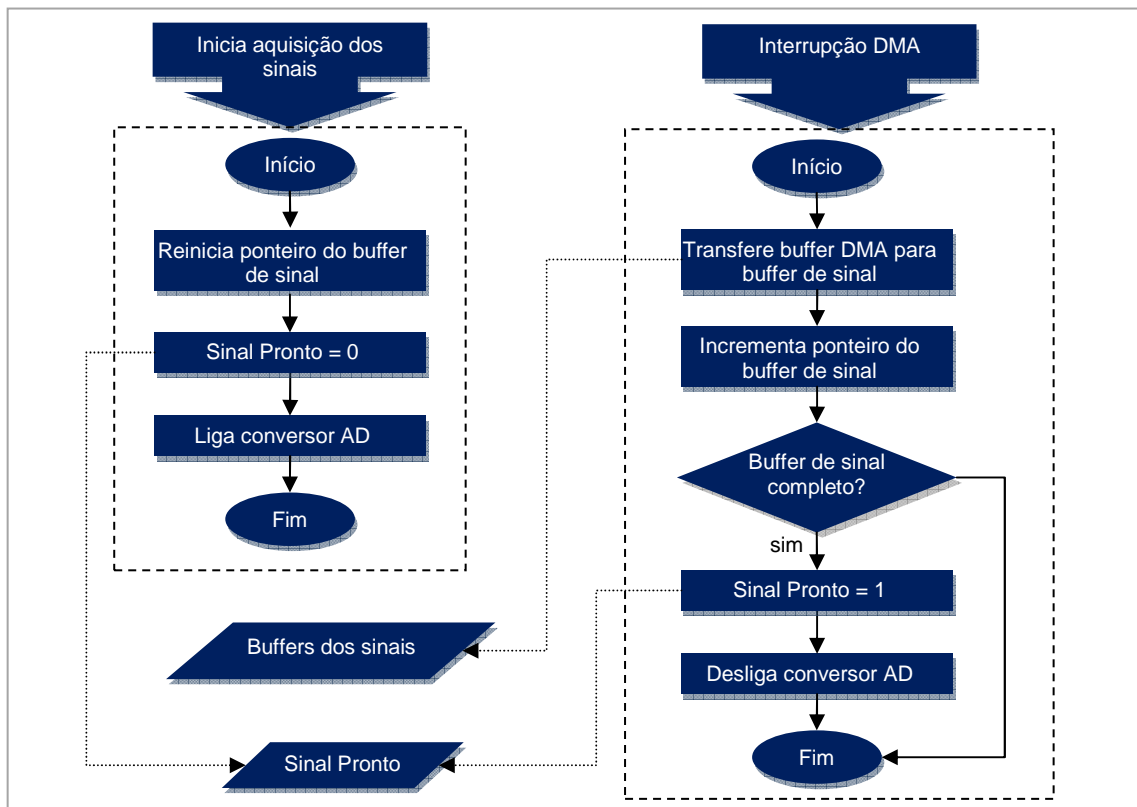


Figura 24: Fluxograma das rotinas de aquisição dos sinais

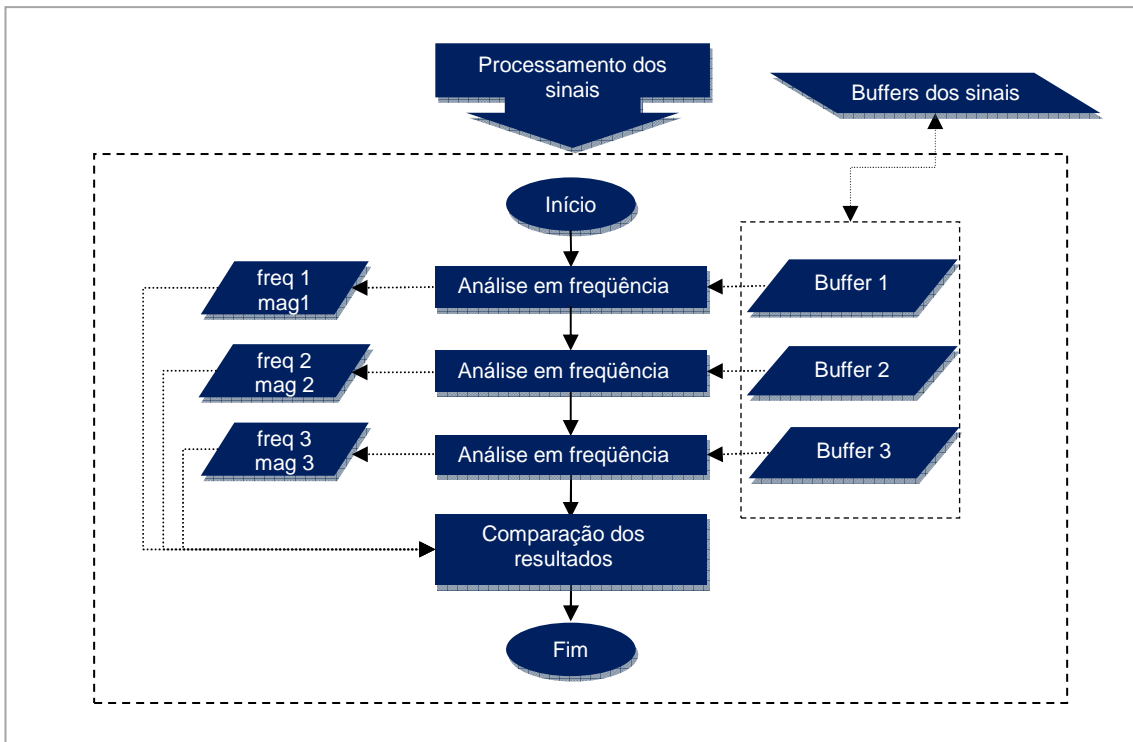


Figura 25: Fluxograma do processamento dos sinais

5.6 Análise em freqüência

O algoritmo utilizado para a análise em freqüência toma como entrada um dos buffers de sinal, transforma-o para o domínio da freqüência, acha a freqüência de maior componente e retorna o valor da magnitude e freqüência da maior componente encontrada. É possível implementar essa rotina utilizando as funções disponíveis na biblioteca *DSP* do compilador:

- *FFT* – função de FFT com decimação em freqüência, trata os dados como números complexos e a saída tem os bits de endereçamento invertidos.
- *BitReverseComplex* – Reordena o vetor de sinais invertendo os bits de endereçamento de um vetor de números complexos.
- *SquareMagnitudeComplex* – calcula o valor ao quadrado dos módulos do número complexo
- *VectorMax* – Procura pelo máximo valor dentro de um vetor, retorna o valor máximo e seu índice.

A função *FFT* da biblioteca é fornecida com escalonamento fixo por estágio, o que leva a uma faixa dinâmica pequena. A função foi, portanto, modificada utilizando o conceito de *Block float point FFT* como descrito no capítulo 2.2, para se conseguir uma faixa dinâmica maior. Também embutiu-se nessa função a rotina de ordenação dos bits de endereçamento, e a rotina para o cálculo dos módulos dos números complexos.

A modificação da função de FFT foi feita no código original em linguagem *assembly* para que tivesse o melhor desempenho tanto em tamanho de código como em velocidade de processamento.

5.7 Interface LCD

O controle do módulo LCD é feito através do envio de comandos e dados de 1 *byte*. Como descrito na implementação do hardware, o barramento utilizado é de 4 bits, sendo portanto necessário enviar os comandos e dados separados por *nibbles*. O envio de comandos e dados é informado ao módulo LCD por meio do sinal RS. As figuras 26 e 27 ilustram, respectivamente, o diagrama de sinais e o fluxograma para escrita de comandos e caracteres. O módulo de LCD possui um modo de leitura destinado a verificar se o último comando ou dado foi processado. Optou-se por utilizar os pinos do barramento como unidirecional, portanto o modo leitura não é utilizado e por conta disso necessita-se adicionar um pequeno atraso entre o envio dos comandos para que haja tempo para o processamento dos mesmos pelo módulo.

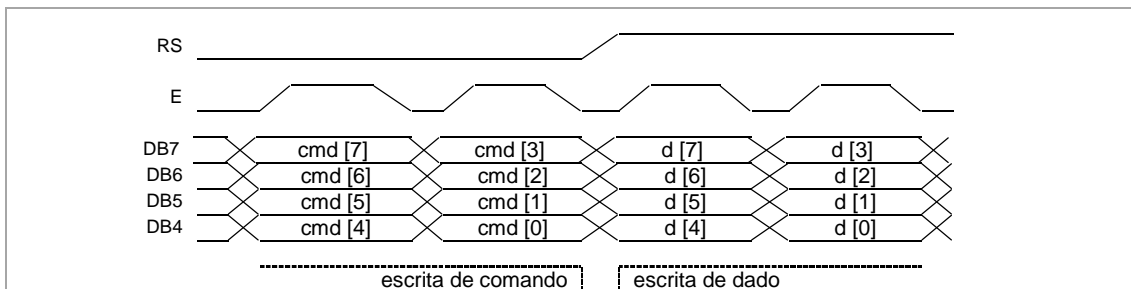


Figura 26: Diagrama dos sinais para escrita de comandos e dados no módulo LCD

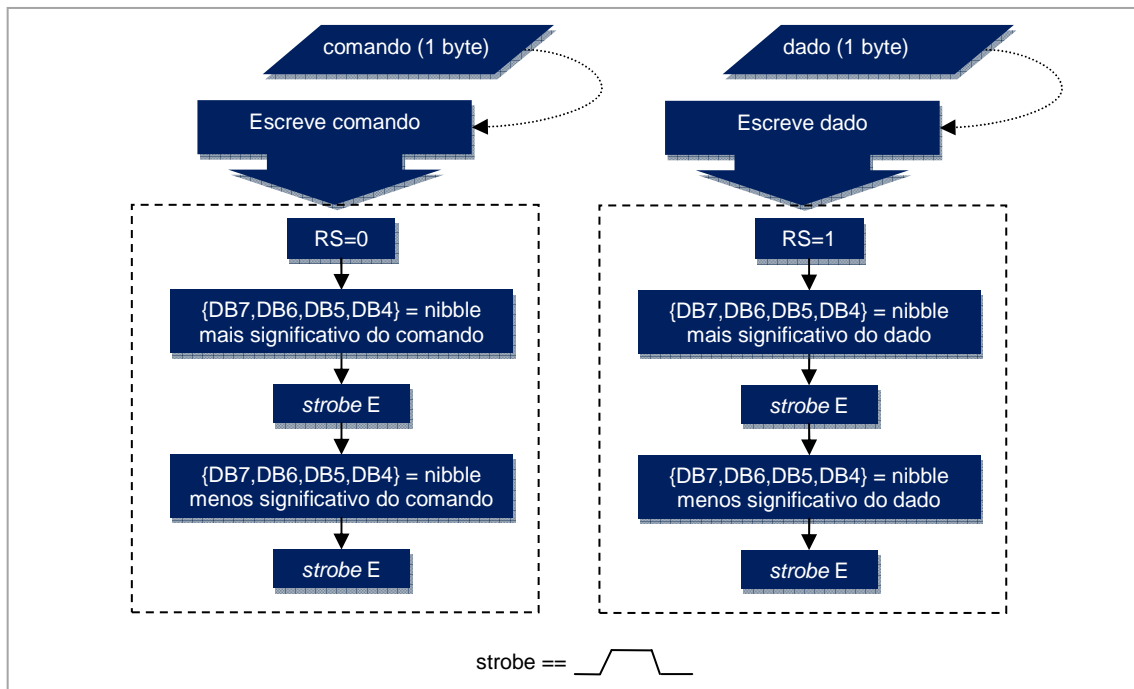


Figura 27: Fluxograma para escrita de comandos e dados no módulo LCD

Os comandos determinam o modo de funcionamento e são utilizados para ajustar as configurações do módulo como tipo de cursor, limpeza do *display*, etc. A lista de todos os comandos e modos de funcionamento pode ser encontrada no manual do módulo.

5.8 Interface UART

Depois de o módulo ser inicializado as operações são feitas diretamente através dos registradores de leitura e escrita.

A interface UART, deve ser utilizada para comunicação com outros equipamentos através do envio de uma *string* contendo o valor da rotação do motor, obedecendo a formatação especificada no capítulo 3.

Modo debug

O modo debug é uma especificação de implementação e consiste em enviar os sinais e respectivos espectros pela interface serial, para que possam ser visualizados e documentados em um computador.

Este modo é ativado quando o equipamento recebe o caractere 't' pela interface UART, que ativa uma variável de sincronização. Esta variável é monitorada a cada ciclo de aquisição/processamento de sinal e caso esteja ativada, todos os elementos do vetor de sinal são enviado pela interface UART.

A Figura 28 mostra os fluxogramas das funções de UART.

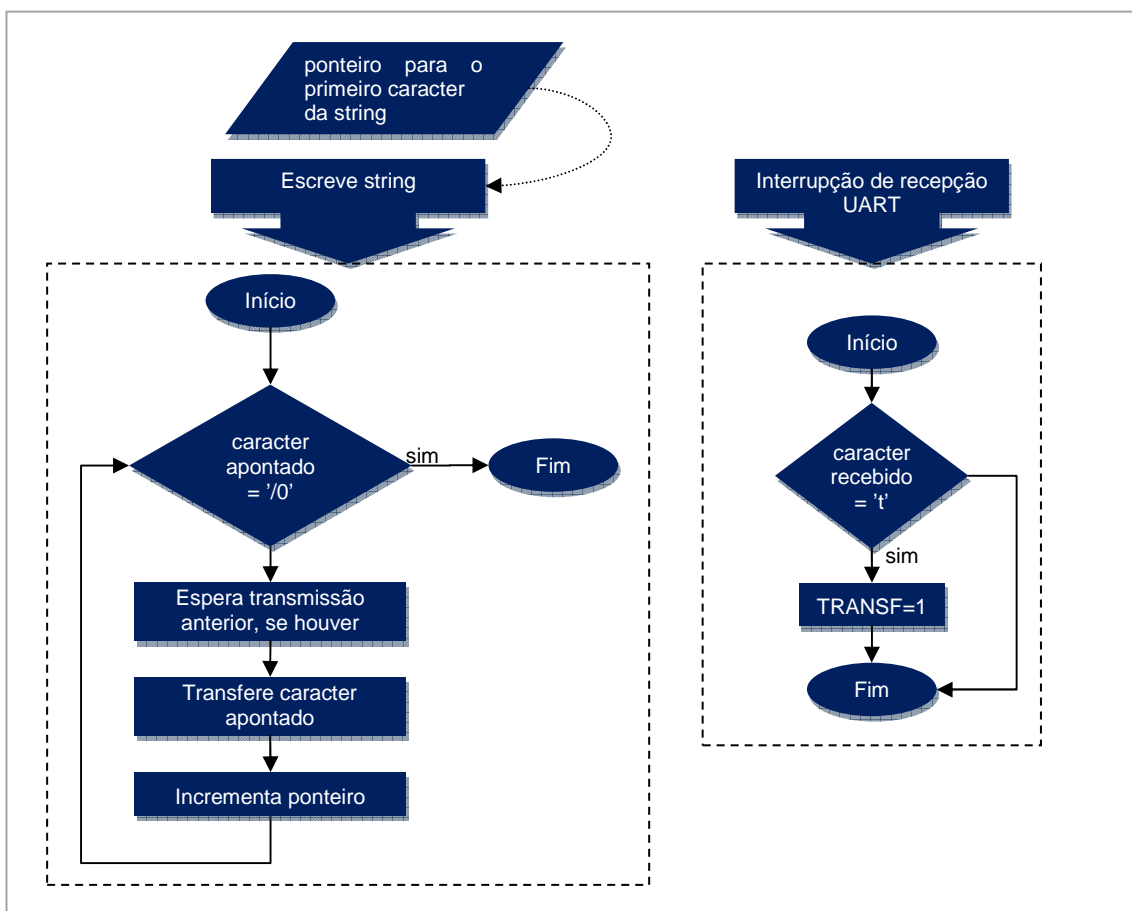


Figura 28: Fluxograma das rotinas UART

5.9 Geração da saída pulso

O controle da saída pulso, tal como descrito nas especificações, é implementado utilizando interrupção de dois *timers* (Timer 1 e Timer 2). O funcionamento é ilustrado pelo diagrama de estados da Figura 29.

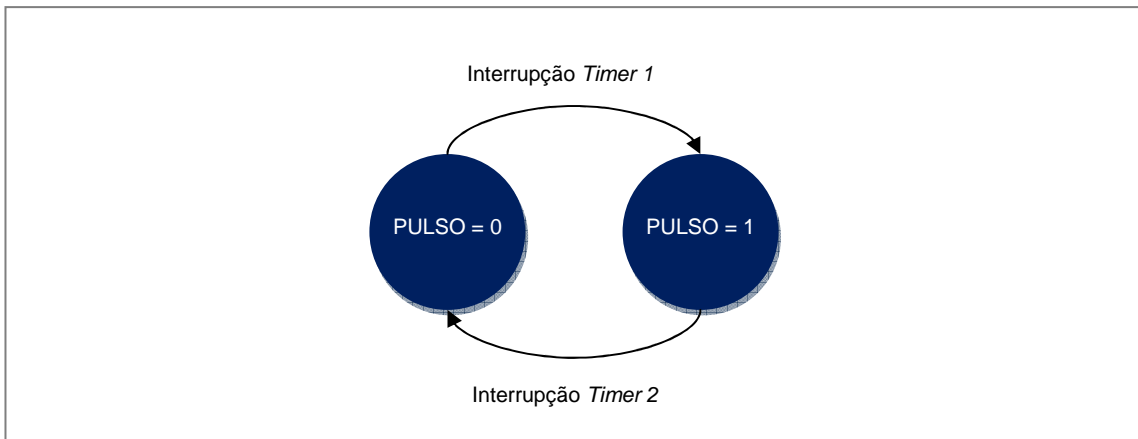


Figura 29: Diagrama de estados da saída pulso

Não é necessária nenhuma função no *loop* principal para controlar o pino da saída pulso, basta somente que o registrador de tempo do *Timer 1* seja atualizado para cada nova medida da rotação. O fluxograma das rotinas relacionadas à saída pulso é mostrado Figura 30.

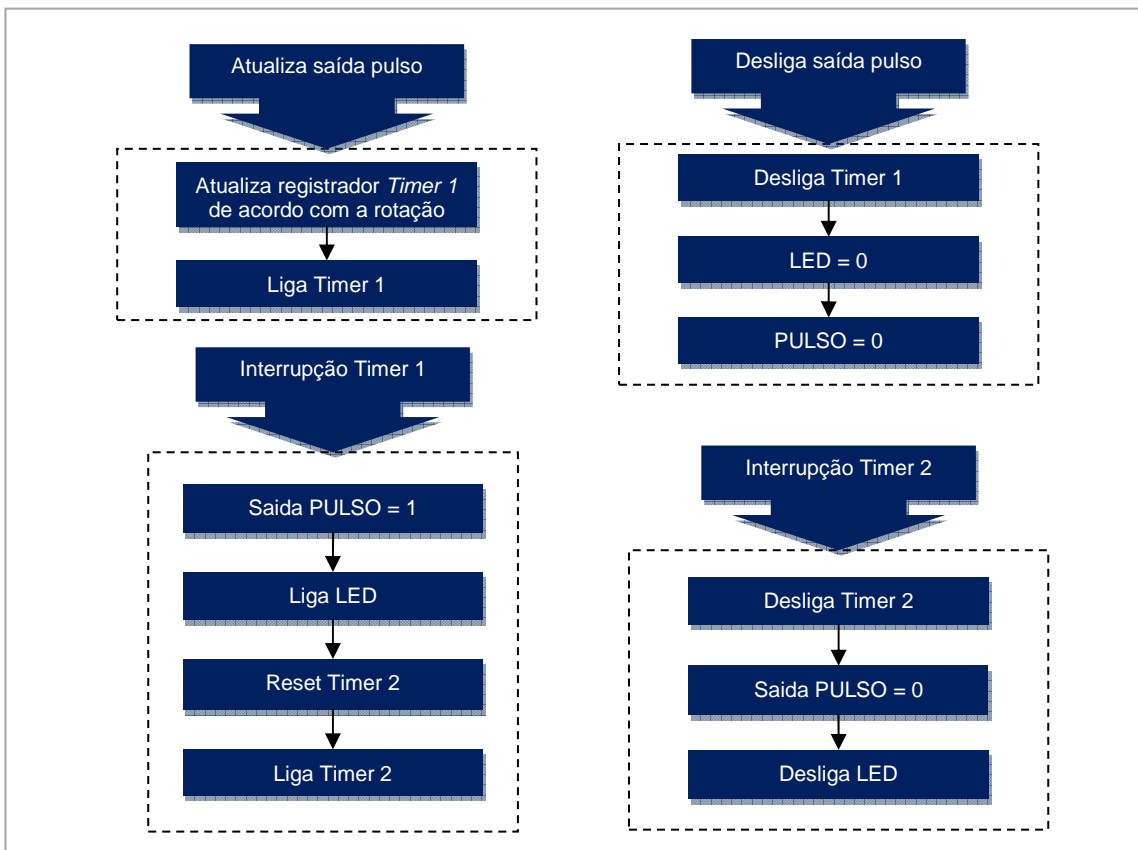


Figura 30: Procedimentos para controle da saída pulso

5.10 Interface das chaves

O tratamento das chaves é feito utilizando a interrupção por troca de estado em conjunto com uma rotina no laço principal.

Sempre que ocorre uma troca de estado numa das entradas das chaves a rotina de interrupção verifica se alguma das chaves foi pressionada, caso positivo uma variável BOT_x é ativada (onde x é o número da chave). As variáveis BOT_x dizem se há alguma tecla pressionada

que ainda não teve tratamento. A Figura 31 mostra o fluxograma do algoritmo de detecção das chaves.

A rotina de verificação das chaves no laço principal faz a navegação das opções de configuração do equipamento de acordo com o diagrama de estados da Figura 32. Sempre que o botão pressionado é tratado, sua variável correspondente BOT_x deve ser desativada.

Sempre que o estado de navegação atual for diferente de “modo medição” a variável global $CONFIG$ é ativada, fazendo com que as rotinas de aquisição/processamento dos sinais, no laço principal, sejam ignoradas.

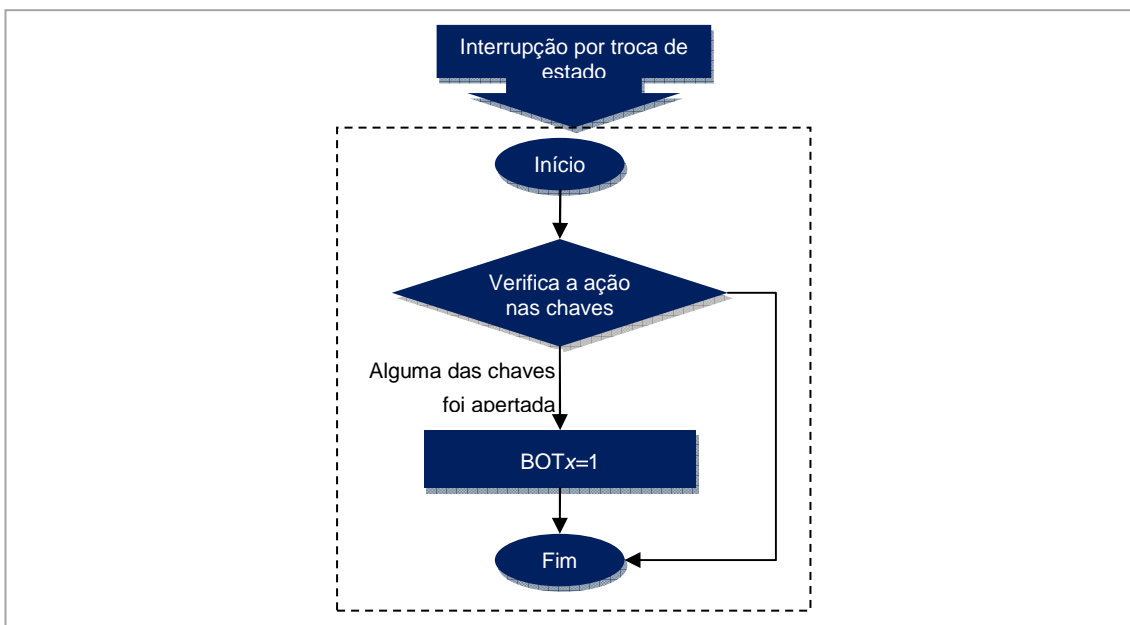


Figura 31: Fluxograma para detecção das chaves

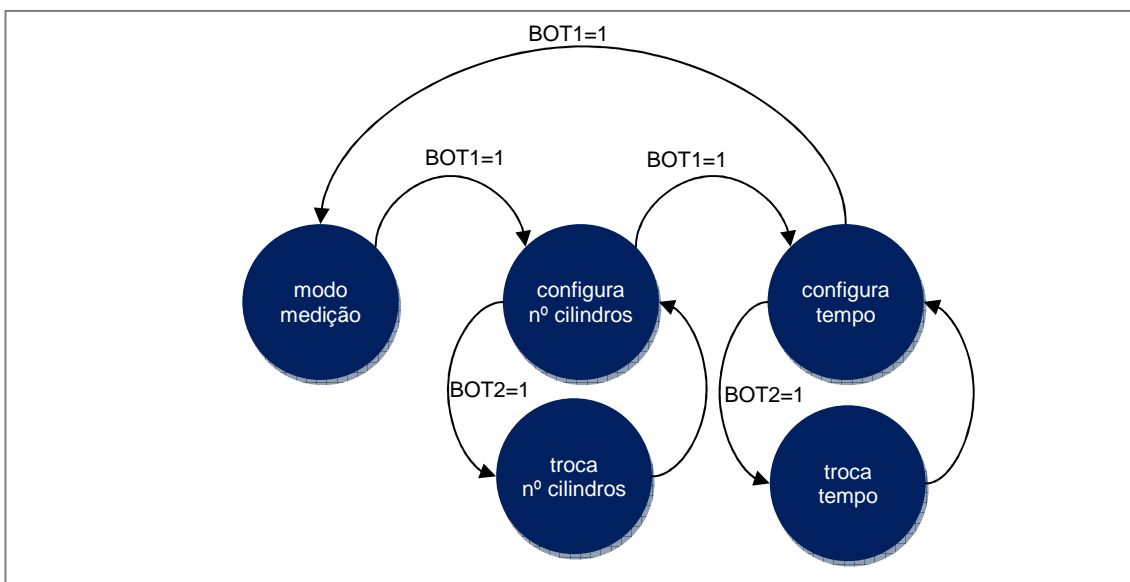


Figura 32: Diagrama dos estados da navegação pelas chaves

6 Resultados e discussão

6.1 Resultados obtidos

Os testes foram efetuados utilizando o protótipo do equipamento, descrito anteriormente. O acelerômetro foi montado em uma pequena placa de circuito impresso e fixado ao motor por um ímã. As figuras 36 e 37, a seguir, mostram a placa de protótipo e o acelerômetro.

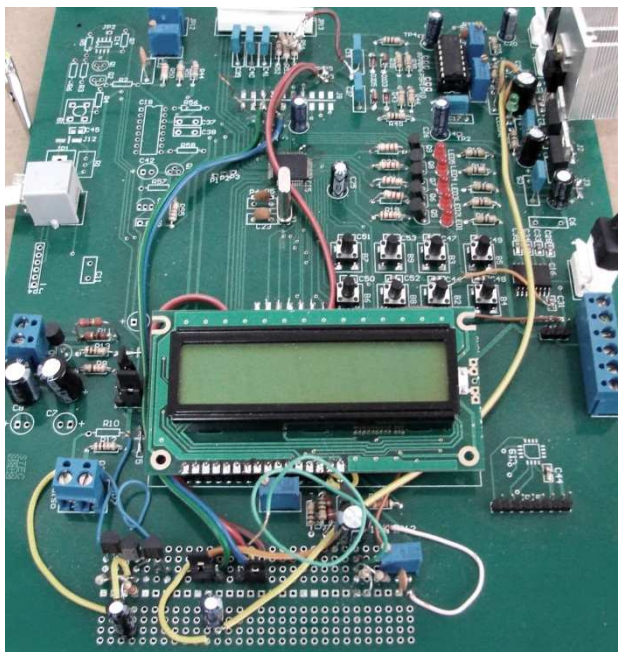


Figura 33: Placa do protótipo



Figura 34: Montagem do acelerômetro

A seguir apresentam-se os resultados de testes nos seguintes veículos:

- *Ford Ranger*: Veículo utilitário com motor diesel de 6 cilindros em V montado longitudinalmente ao chassi.

- *Toyota Hilux*: Veículo utilitário com motor diesel de 4 cilindros em linha montado transversalmente ao chassi.

O acelerômetro foi orientado com o eixo x no sentido longitudinal e o eixo y no sentido transversal do veículo.

As medidas de referência foram feitas utilizando o tacômetro comercial *Minipa MDT-2238*.

As figuras 35, 36, 37, 38, 39, 40 e 41 mostram a vibração nos eixos do acelerômetro nos veículos de teste. Os eixos do acelerômetro seguem a orientação citada acima.

As tabelas 1, 2, 3, 4, 5, 6 e 7 apresentam as componentes de frequências de maior magnitude de cada eixo do acelerômetro assim como suas respectivas magnitudes. Apresenta-se também a conversão da frequência de vibração para frequência de rotação segundo a equação mostrada no capítulo 1.2.

A medida de rotação acusada pelo equipamento desenvolvido é a medida da maior componente de frequência do eixo cuja magnitude é a maior dentre os três eixos, indicada em destaque nas tabelas.

• *Ford Ranger com rotação de 840 RPM*

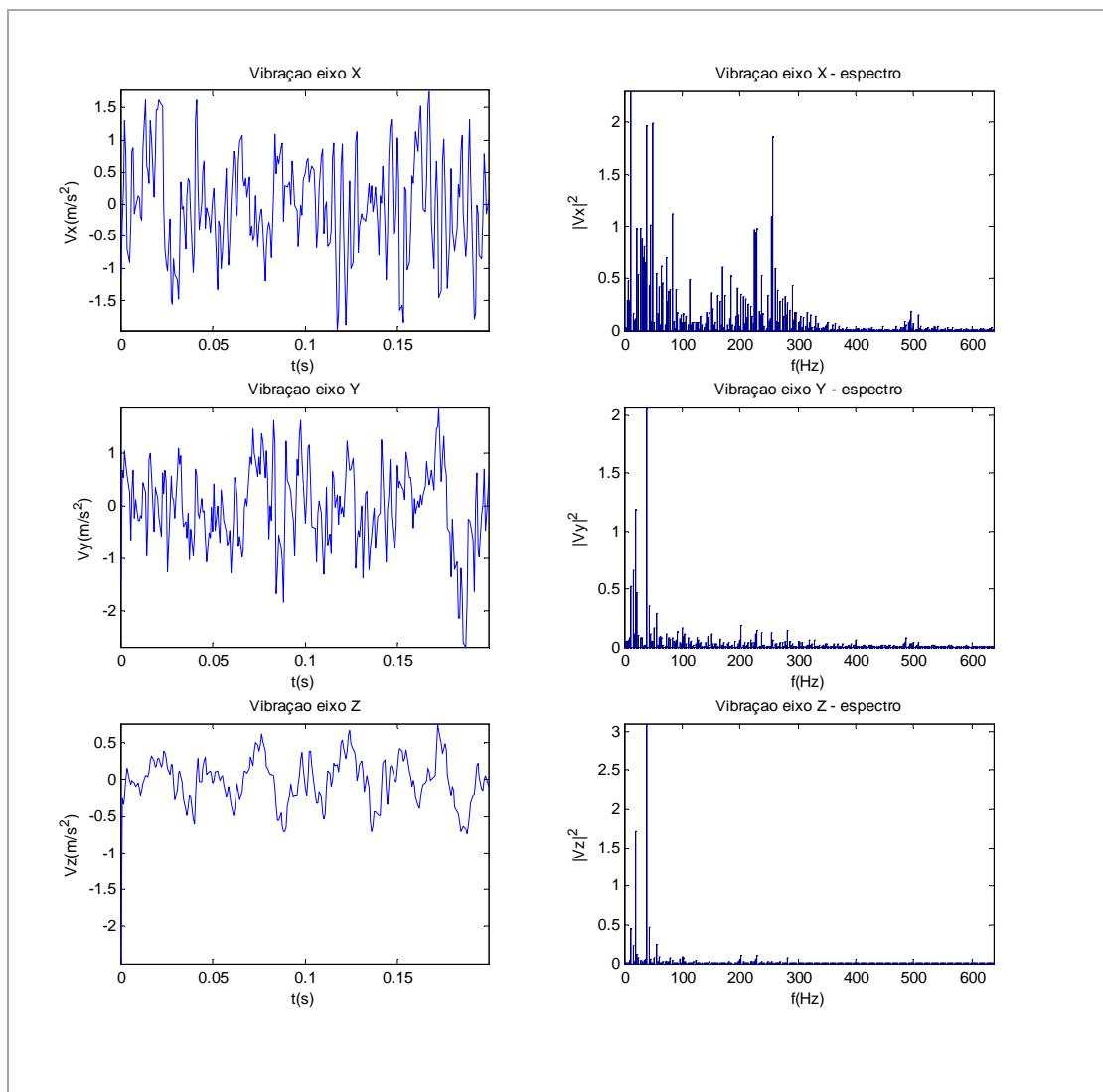


Figura 35: Medidas de vibração do veículo Ford Ranger com rotação 840 RPM

Tabela 2: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 840 RPM

| eixo | x | y | z |
|------------------|--------|--------|---------------|
| frequência (Hz) | 012.50 | 040.00 | 040.00 |
| frequência (RPM) | 250.00 | 800.00 | 800.00 |
| magnitude | 002.30 | 002.07 | 003.10 |

• *Ford Ranger com rotação de 1250 RPM*

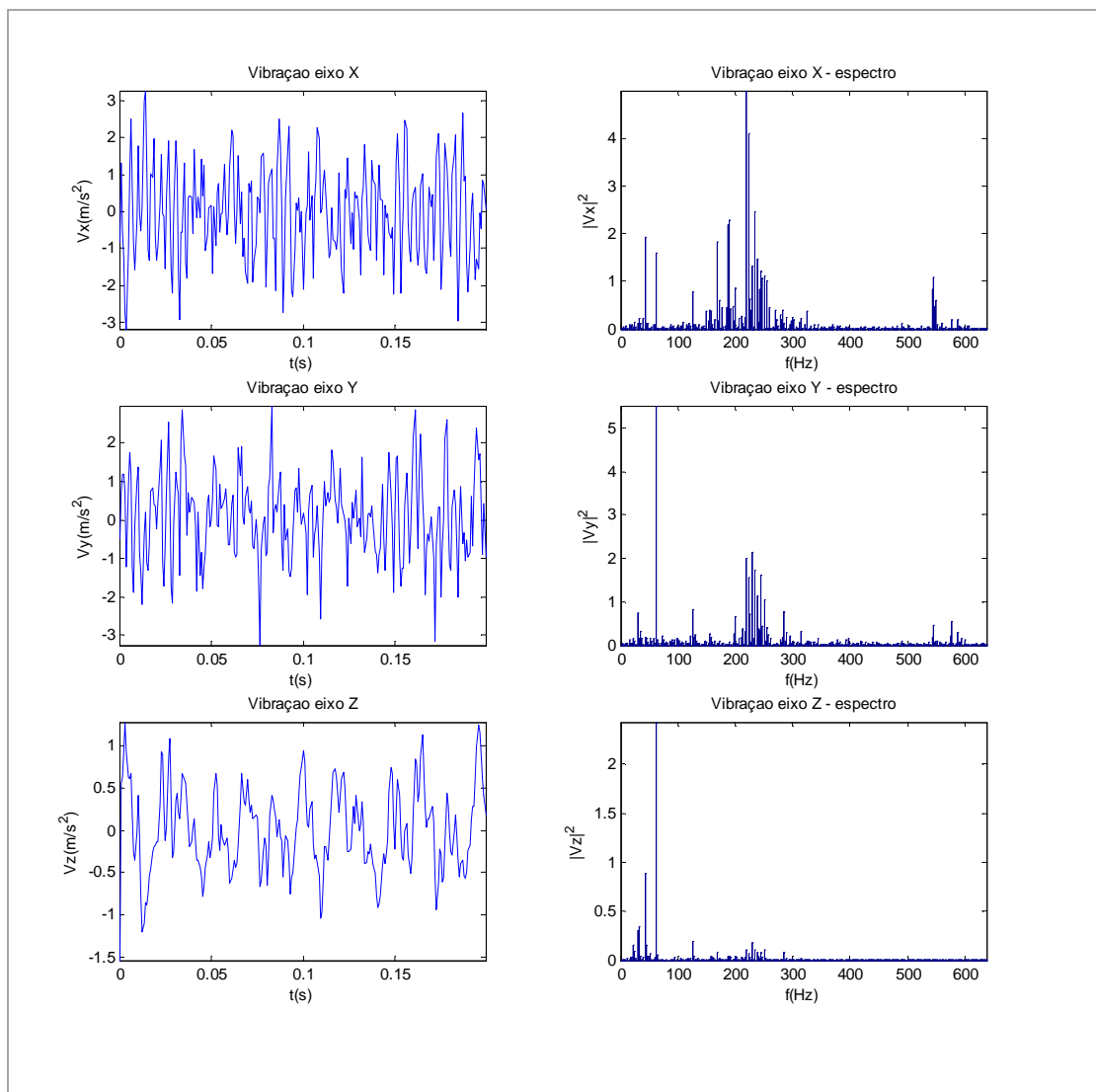


Figura 36: Medidas de vibração do veículo Ford Ranger com rotação 1250 RPM

Tabela 3: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1250 RPM

| eixo | x | y | z |
|------------------|---------|----------------|---------|
| frequência (Hz) | 220.00 | 062.50 | 062.50 |
| frequência (RPM) | 4400.00 | 1250.00 | 1250.00 |
| magnitude | 005.00 | 005.52 | 002.43 |

• *Ford Ranger com rotação de 1990 RPM*

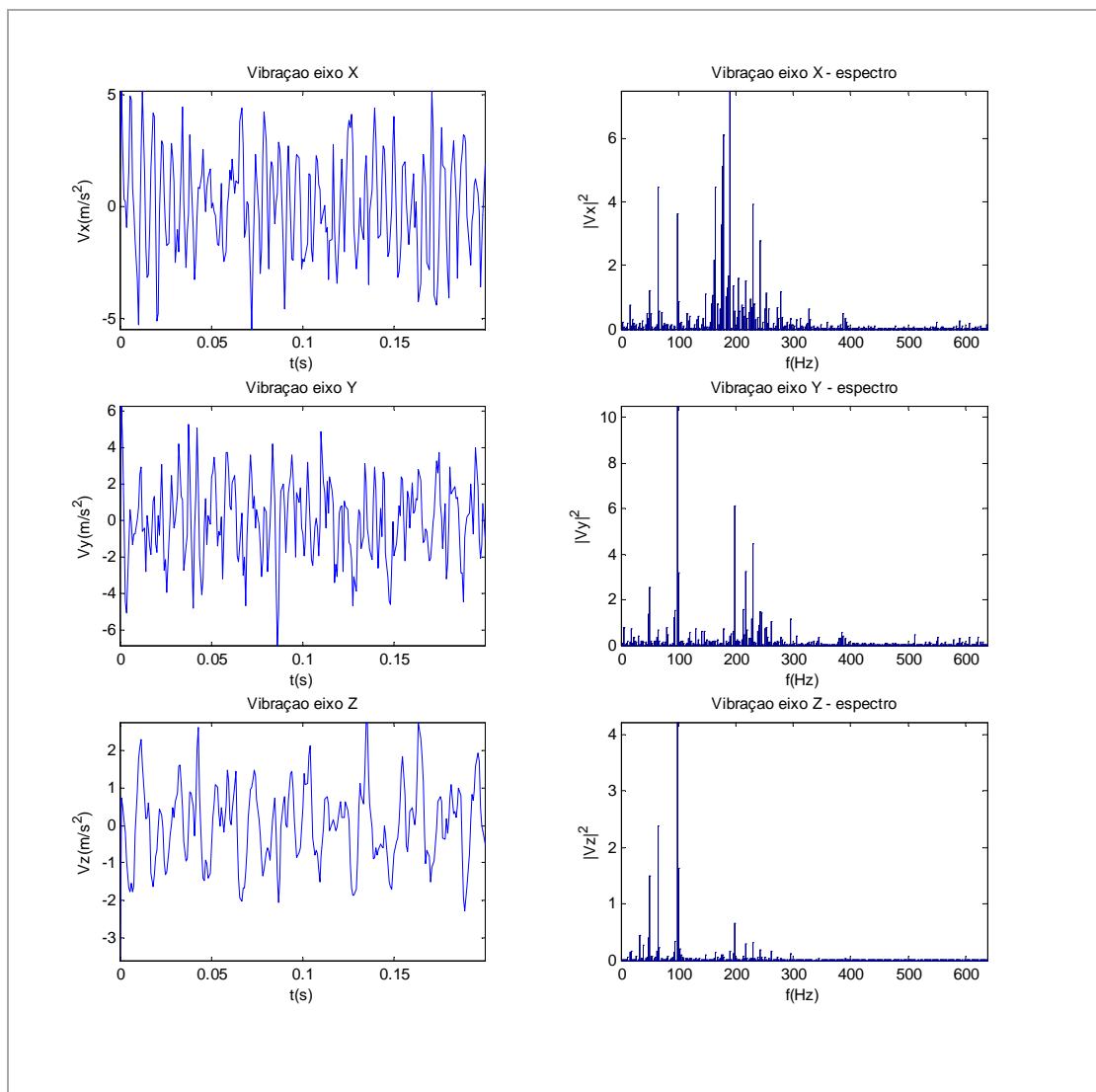


Figura 37: Medidas de vibração do veículo Ford Ranger com rotação 1990 RPM

Tabela 4: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1990 RPM

| eixo | x | y | z |
|------------------|---------|----------------|---------|
| frequência (Hz) | 190.00 | 097.50 | 097.50 |
| frequência (RPM) | 3800.00 | 1950.00 | 1950.00 |
| magnitude | 007.49 | 010.51 | 004.22 |

• *Ford Ranger com rotação de 2680 RPM*

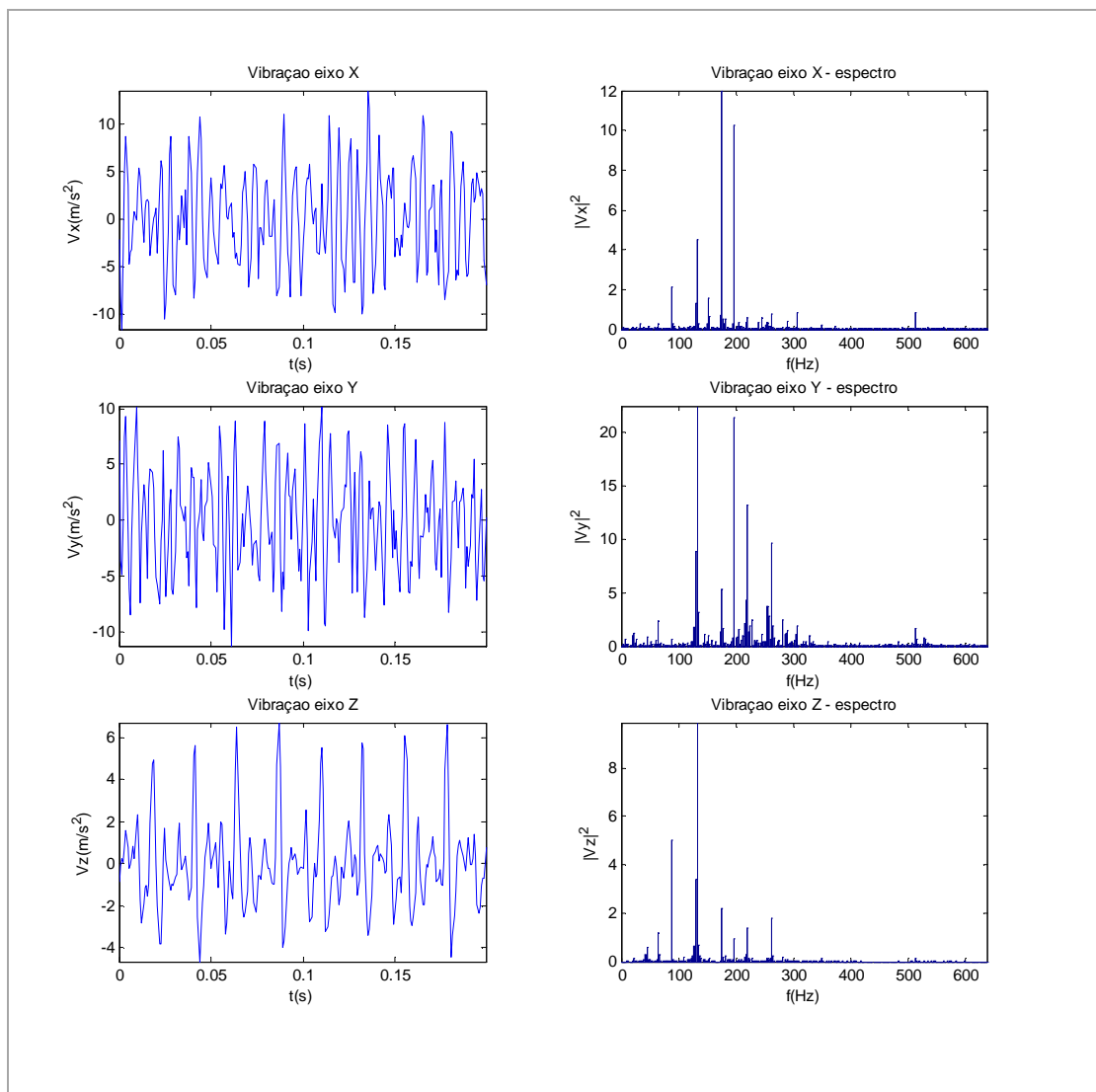


Figura 38: Medidas de vibração do veículo Ford Ranger com rotação 2680 RPM

Tabela 5: Maiores componentes do espectro da vibração do veículo Ford Ranger com rotação 1990 RPM

| eixo | x | y | z |
|------------------|---------|----------------|---------|
| frequência (Hz) | 175.00 | 132.50 | 132.50 |
| frequência (RPM) | 3500.00 | 2650.00 | 2650.00 |
| magnitude | 012.01 | 022.47 | 009.87 |

- *Toyota Hilux com rotação de 1000 RPM*

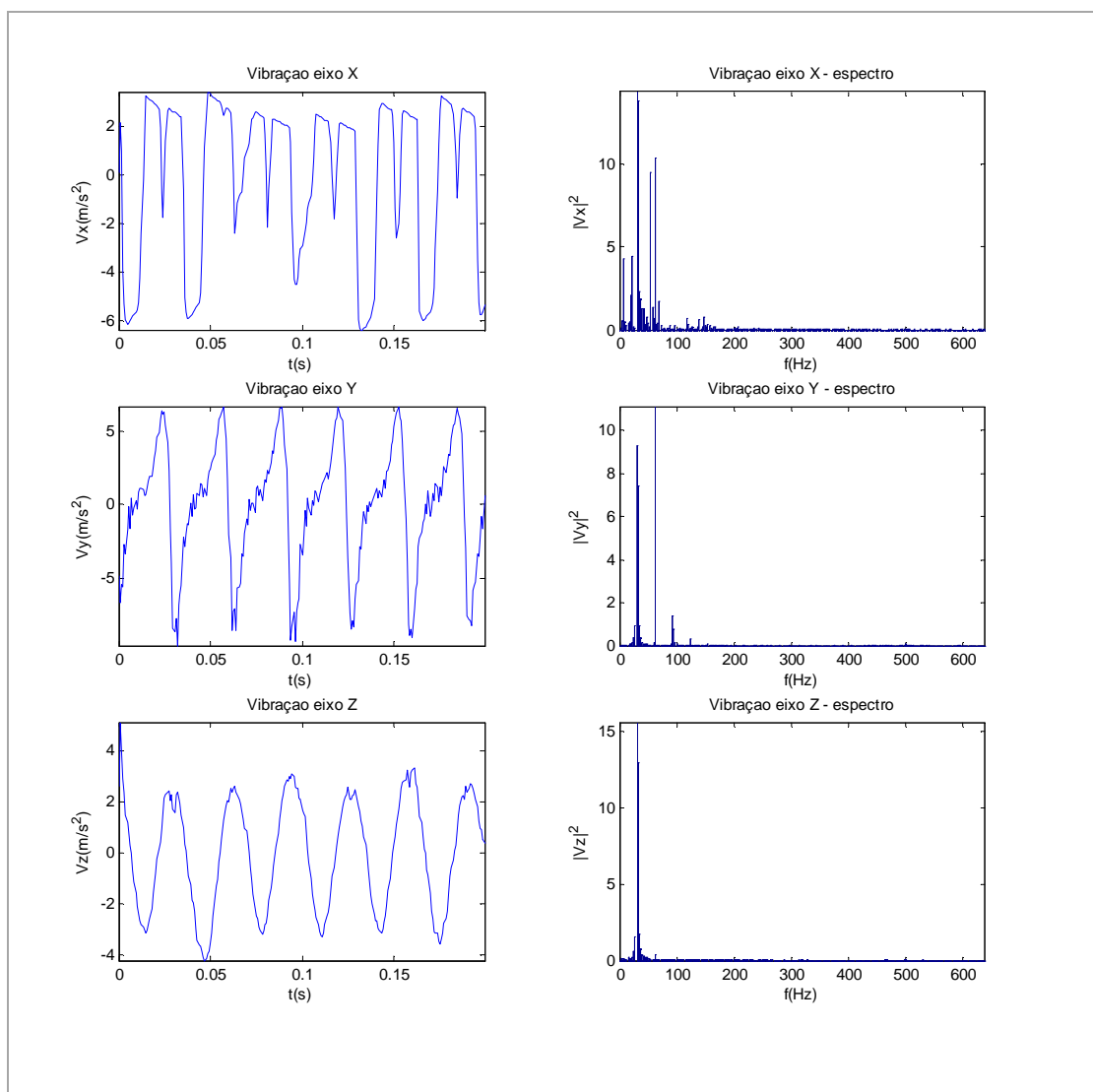


Figura 39: Medidas de vibração do veículo Toyota Hilux com rotação 1000 RPM

Tabela 6: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 1000 RPM

| eixo | x | y | z |
|------------------|--------|---------|---------------|
| frequência (Hz) | 030.00 | 062.50 | 030.00 |
| frequência (RPM) | 900.00 | 1875.00 | 900.00 |
| magnitude | 014.36 | 011.10 | 015.62 |

- *Toyota Hilux com rotação de 1700 RPM*

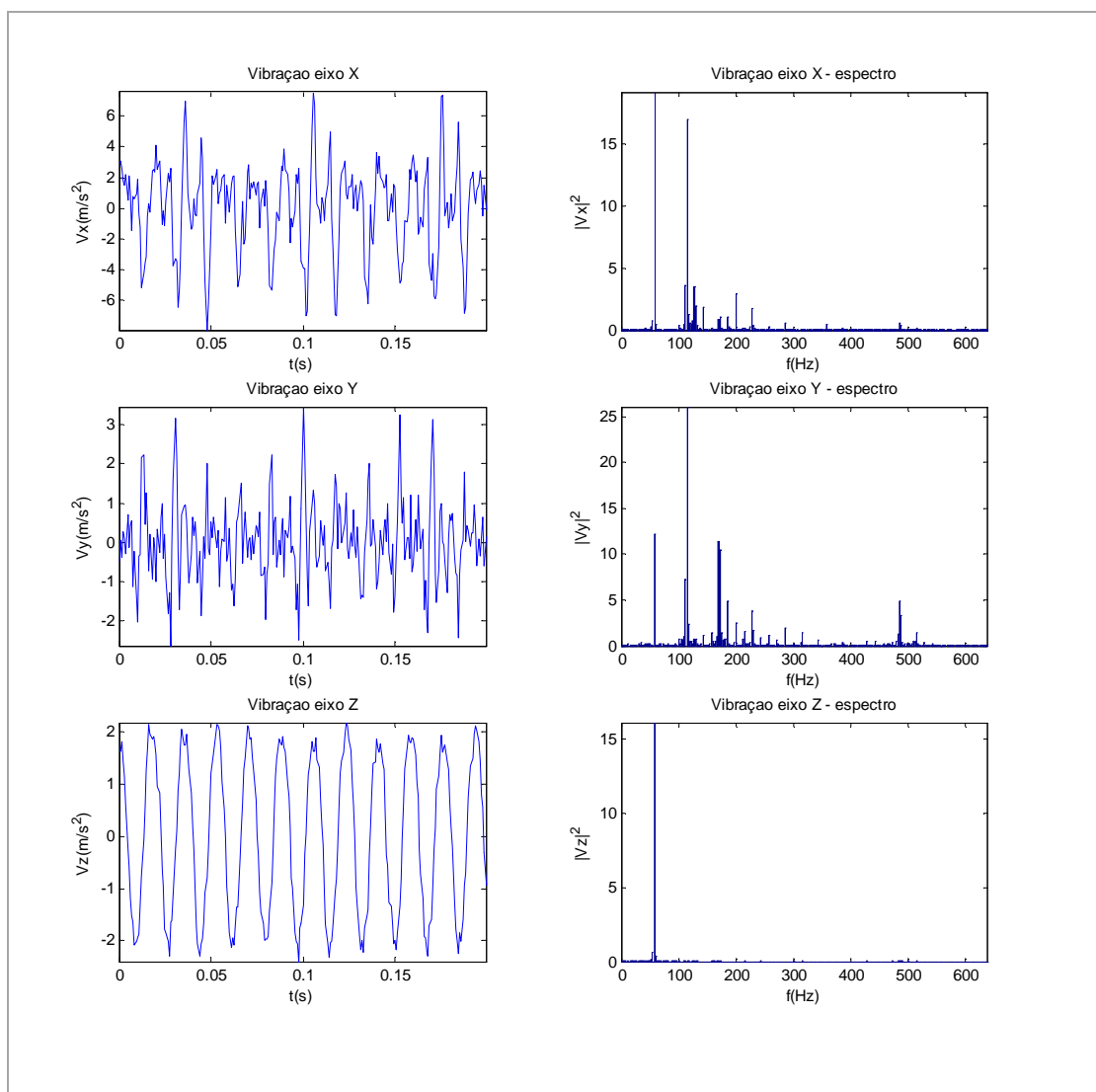


Figura 40: Medidas de vibração do veículo Toyota Hilux com rotação 1700 RPM

Tabela 7: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 1700 RPM

| eixo | x | y | z |
|------------------|---------|----------------|---------|
| frequência (Hz) | 057.50 | 115.00 | 057.50 |
| frequência (RPM) | 1725.00 | 3450.00 | 1725.00 |
| magnitude | 019.18 | 026.11 | 016.11 |

Neste caso a medida da rotação falha, o equipamento indica 3450 RPM. Analisando os gráficos do espectro dos três eixos, na Figura 40, observa-se a detecção correta nos eixos x e z. Com uma observação mais detalhada do eixo y, nota-se que a rotação detectada pelo equipamento na verdade é uma frequência harmônica da frequência da rotação esperada.

- *Toyota Hilux com rotação de 2400 RPM*

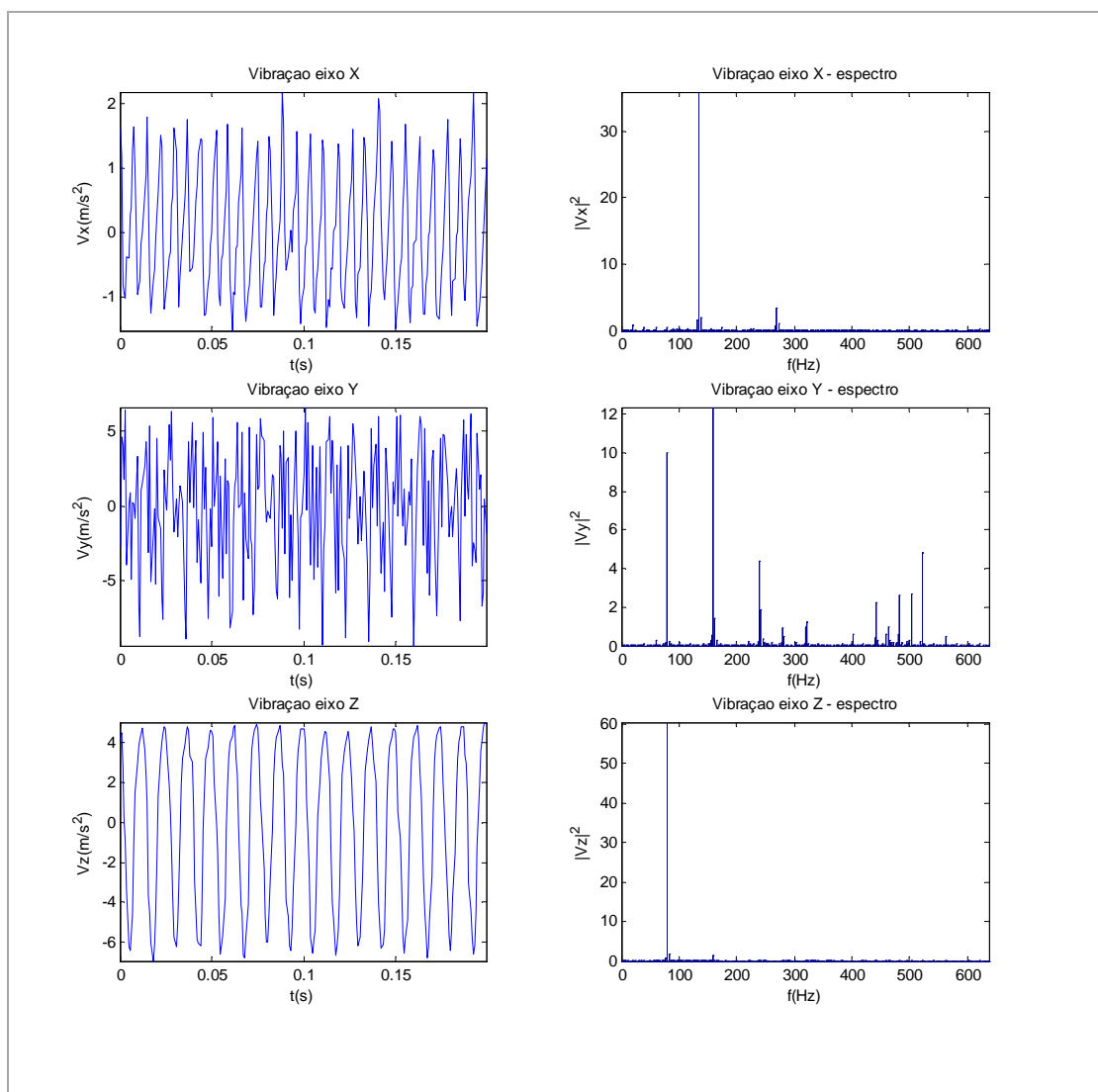


Figura 41: Medidas de vibração do veículo Toyota Hilux com rotação 2400 RPM

Tabela 8: Maiores componentes do espectro da vibração do veículo Toyota Hilux com rotação 2400 RPM

| eixo | x | y | z |
|------------------|---------|---------|----------------|
| frequência (Hz) | 135.00 | 160.00 | 080.00 |
| frequência (RPM) | 4050.00 | 4800.00 | 2400.00 |
| magnitude | 035.91 | 012.35 | 060.46 |

6.2 Discussão

A detecção da rotação falha em alguns casos para o algoritmo utilizado. Observando-se estes resultados mais atentamente nota-se que a frequência a qual é procurada aparece no espectro de outros eixos, ou mesmo no eixo escolhido pelo algoritmo, porém com potência menor que a da frequência escolhida pelo algoritmo. O fato de a frequência procurada estar presente em mais de um eixo pode ser utilizado para desenvolver um algoritmo mais robusto, como sugerido para continuação do trabalho no capítulo 8.

Observa-se também discrepância entre o valor padrão de rotação obtido através de um tacômetro comercial de uso geral. Esta discrepância deve-se à discretização da frequência na análise do espectro. Para um sinal com taxa de amostragem f_a com N pontos tem-se uma resolução em frequência de f_a/N . Como a conversão para RPM ainda depende do número de cilindros, de acordo com as equações mostradas em 1.2 tem-se que a resolução é de:

$$\Delta_{(2T)(RPM)} = 60 \times \frac{f_a(\text{Hz})}{N \times n_{\text{cilindros}}} \quad \text{e} \quad \Delta_{(4T)(RPM)} = 60 \times \frac{2 \times f_a(\text{Hz})}{N \times n_{\text{cilindros}}}$$

A resolução pode ser melhorada, conforme sugestão de continuação do trabalho no capítulo 8.

7 Conclusões

De acordo com os resultados obtidos, conclui-se que é possível determinar a rotação de um motor à explosão através da relação suposta no capítulo 1, constituindo em um excelente método para a medição da rotação de modo indireto, sem a necessidade de acessar o eixo motriz para contagem de voltas, utilizado pelos métodos convencionais.

O algoritmo, porém, falha em alguns casos porque a vibração nos motores não é uma característica bem comportada e não pode ser modelada para todos os casos, pois envolve um grande número de fatores como tipo do motor (gasolina ou diesel), cilindrada, número de cilindros, configuração (em linha, em V, etc), orientação em relação ao chassi (longitudinal ou transversal), tipo de fixação ao chassi dentre outros. Esses fatores podem fazer com que apareçam medidas erradas devidas à ressonância do sistema em frequências harmônicas da rotação, e também pode haver no espectro frequências relativas aos modos próprios de vibração dos corpos do sistema.

Pode-se diminuir as falhas com algoritmos mais robustos que levam em conta similaridades entre a vibração de dois eixos diferentes. A implementação de um algoritmo mais robusto é sugerida no capítulo a seguir como continuação do trabalho.

8 Trabalhos futuros

8.1 Algoritmo mais robusto para detecção

Pode-se desenvolver um algoritmo mais robusto para a detecção da frequência de vibração utilizando uma técnica de processamento digital de sinais chamada correlação cruzada.

Trata-se de um tipo de medida de similaridade de dois sinais quando um atraso no tempo é aplicado a uma delas. Pode ser utilizada para procurar padrões e em análises criptográficas.

Para funções contínuas f e g a correlação cruzada é definida como:

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

Para funções discretas f e g a correlação cruzada é definida por:

$$(f \odot g)[n] = \sum_{k=-\infty}^{\infty} f^*[k]g[n + k]$$

Para implementação prática de sinais discretos a dimensão do sinal é finita e não há posições negativas para as amostras, levando à definição prática da correlação cruzada entre os sinais discretos x e y :

$$(x \odot y)[n] = \sum_{k=0}^{N-1} x[k]y[n + k],$$
$$\begin{cases} N \rightarrow \text{dimensão } x[k] \\ M \rightarrow \text{dimensão } y[n] \\ 0 \leq n < N + M - 1 \\ M \leq N \end{cases}$$

A correlação cruzada equivale, no domínio da frequência, a multiplicar termo a termo as componentes em amplitude dos sinais. Por isso sua utilização deve praticamente eliminar as falsas medidas, observadas nos resultados.

8.2 Aumento da resolução da medida

A resolução das medidas é diretamente proporcionais à frequência de amostragem do sinal e inversamente proporcional ao número de pontos da medida e ao número de cilindros. O número de cilindros não é uma variável, portanto restam a frequência de amostragem e o número de pontos do sinal para melhorar a resolução. Uma sugestão interessante é utilizar uma técnica chamada de decimação, equivalente a diminuir a frequência de amostragem do sinal, com um fator dependente do número de cilindros. A decimação pode ser feita no estágio da aquisição dos sinais, não interferindo em nada nos estágios subsequentes, tornando a resolução indiferente ao número de cilindros do veículo.

Referências bibliográficas

- DE SILVA, C.W. **Vibration: fundamentals and practice**. CRC Press, 1999. 976p.
- HAYES, M.H. **Schaum's Outline of Digital Signal Processing**. McGraw-Hill, 1999. 432p.
- KESTER, W. **Mixed-Signal and DSP Design Techniques**. Analog Devices, 2003. 368p.
- TEXAS INSTRUMENTS **A Block Floating Point Implementation for an N-Point FFT on the TMS320C55x DSP**. Texas Instruments, 2003. 13p.
- FREESCALE SEMICONDUCTOR **MMA7260QT: XYZ axis accelerometer $\pm 1.5g/2g/4g/6g$** . Freescale Semiconductor, 2008. 12p.
- FREESCALE SEMICONDUCTOR **Using the Multi-Axis g-Select Evaluation Boards**. Freescale Semiconductor, 2006. 9p.
- MICROCHIP - **dsPIC33FJXXXMCX06/X08/X10 Motor Control Family Data Sheet**. Microchip, 2007. 342p.
- HITACHI – **HD44780U Dot Matrix Liquid Crystal Display Controller/Driver**. Hitachi, 1998. 60p.