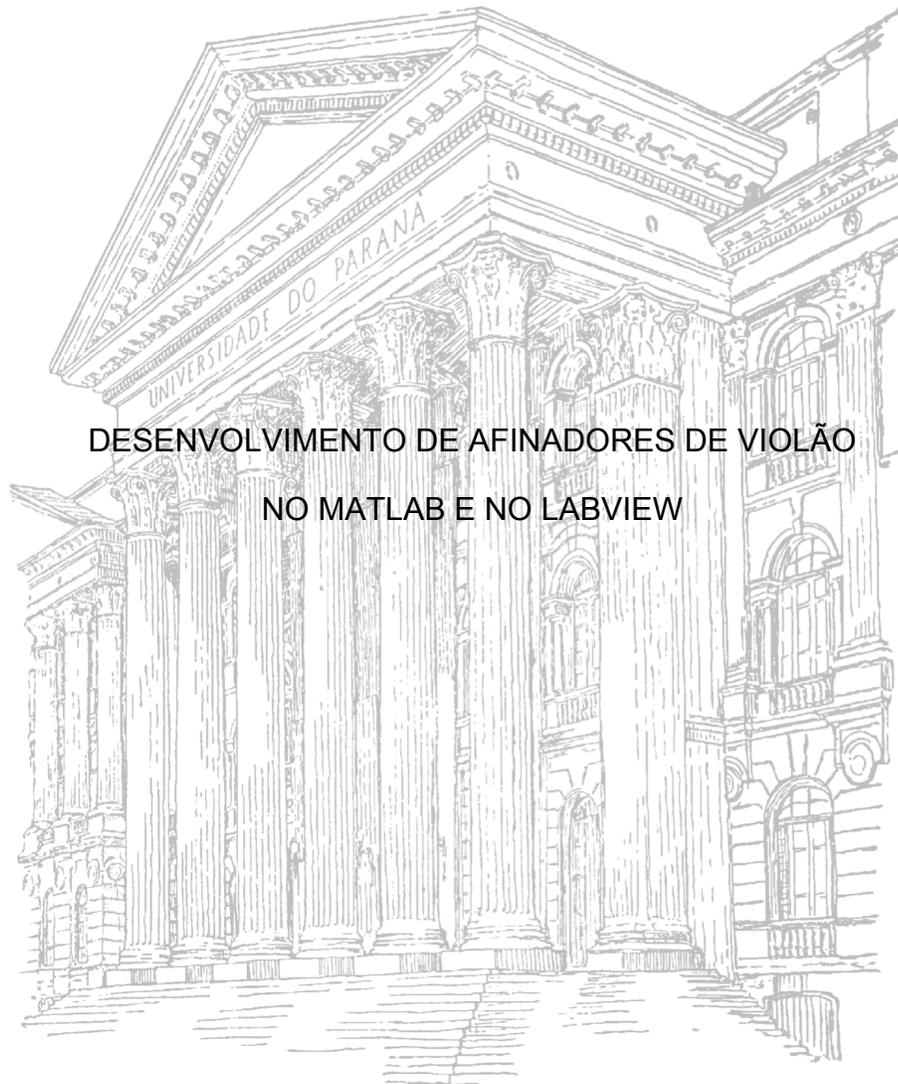


UNIVERSIDADE FEDERAL DO PARANÁ

FABIO LUIS REWAY



DESENVOLVIMENTO DE AFINADORES DE VIOLÃO  
NO MATLAB E NO LABVIEW

CURITIBA  
2014

FABIO LUIS REWAY

DESENVOLVIMENTO DE AFINADORES DE VIOLÃO  
NO MATLAB E NO LABVIEW

Relatório apresentado junto ao  
Curso de Engenharia Elétrica da  
UFPR como requisito parcial  
para a aprovação na disciplina de  
Processamento Digital de Sinais II.

Orientador:  
Prof. Dr. Eduardo Parente Ribeiro

CURITIBA

2014

## SUMÁRIO

1 INTRODUÇÃO .....	4
2 DESENVOLVIMENTO .....	4
2.1 MATLAB .....	5
2.2 LABVIEW .....	6
3 CONCLUSÕES .....	8
APÊNDICE I – BLOCOS DO LABVIEW .....	9

## 1 INTRODUÇÃO

O tema deste trabalho é o desenvolvimento de um afinador de violão no MATLAB e, também, outro no LABVIEW. Ambos captam o som das cordas do violão através da placa de som do computador.

Objetiva-se com este trabalho aplicar os conhecimentos da disciplina de Processamento Digital de Sinais II em um projeto prático e que tenha uma aplicação direta nas atividades do dia-a-dia de músicos e interessados por música.

A motivação deste trabalho é o fato de existirem equipamentos de afinação automáticos no mercado e de alto valor comercial e, dessa forma, este trabalho busca demonstrar o valor e a importância do conhecimento aplicado à um produto existente no mercado musical.

A afinação de um violão é um processo que envolve a comparação da frequência da corda tocada com a frequência da mesma corda devidamente afinada. A diferença entre as duas frequências deve ser o mais próxima de zero para que se tenha uma afinação perfeita.

A tabela 1 lista as cordas de um violão com as suas respectivas frequências.

Nota musical (2a Harmonica)	Frequência (Hz)
<b>E</b>	164
<b>B</b>	219
<b>G</b>	293
<b>D</b>	392
<b>A</b>	493
<b>E</b>	659

Tabela 1 – Cordas de um violão e suas respectivas frequências

## 2 DESENVOLVIMENTO

### 2.1 MATLAB

O desenvolvimento do programa do MATLAB do afinador segue o fluxograma da figura 1.

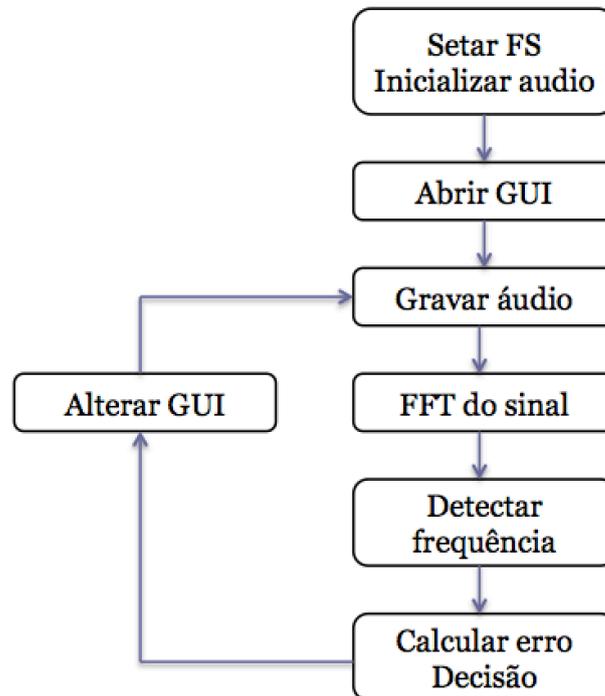


Figura 1 – Fluxograma do programa do afinador em MATLAB

A primeira parte do programa seta a frequência de amostragem igual à 44100 amostras por segundo e, em seguida, inicializa um objeto retornado pela função ***audiorecorder()***, capaz de armazenar o som captado pela placa de som do computador.

Em seguida, é inicializada a *interface* gráfica do usuário, cujo código foi codificado em um arquivo à parte – chamado `afinador_gui.m` – e é totalmente desvinculado das funções matemáticas do afinador.

A *interface* gráfica pode ser observada na Figura 2.

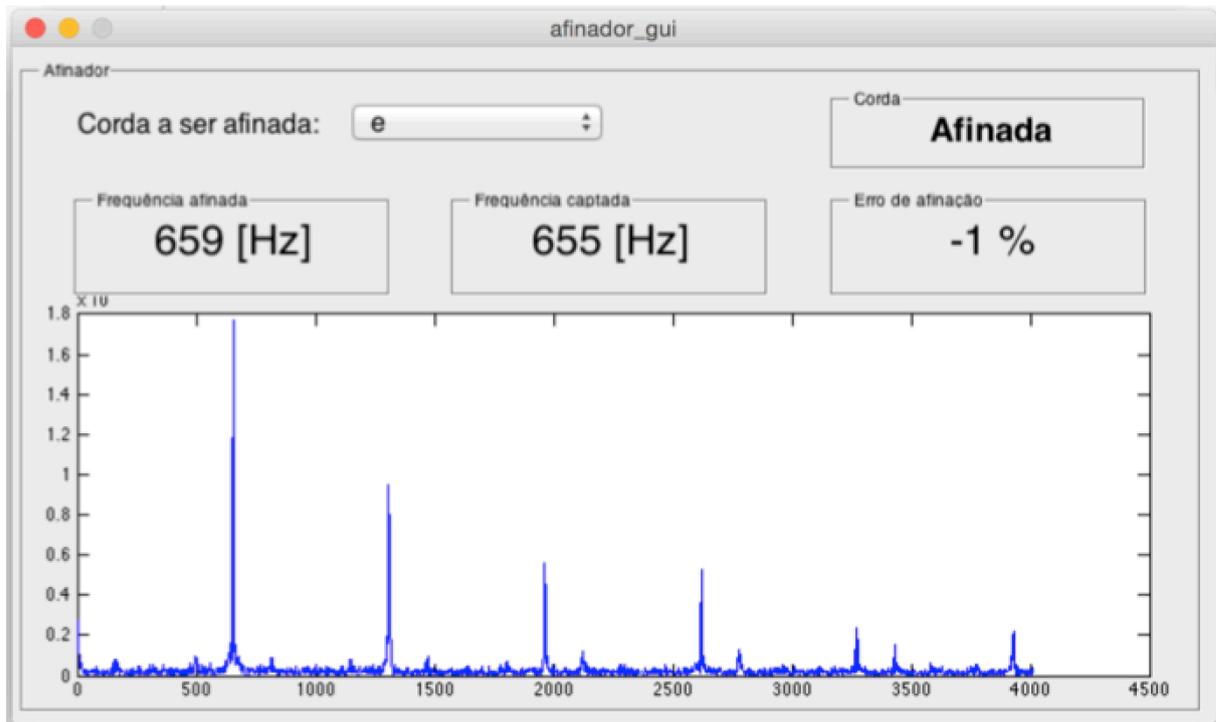


Figura 2 – Interface gráfica do usuário para o afinador desenvolvido no MATLAB

Esta *interface* possui um *menu* com o qual é possível escolher qual corda deseja-se afinar. Após feita a escolha, a *string* do campo “Frequência afinada” irá ser alterada para a frequência correspondente à Tabela 1. A escolha da corda dispara a função ***set\_note()*** que seta uma variável global para que todo o programa conheça qual é a nota escolhida pelo usuário.

Após captado o som através da função ***record\_sound()*** do programa, todas as amostras são enviadas para a função ***get\_fft\_and\_plot()***, que então calcula a FFT através do código:

```

Fs=44100;
[L,c] = size(data);
NFFT=L;
Y = fft(data,NFFT)/L;
f = Fs/2 * linspace(0,1,NFFT/2+1);

```

A função também é responsável por plotar a FFT no gráfico da interface gráfica do usuário.

Logo após de ser calculada a FFT, a função ***get\_signal\_frequency()*** é chamada para detectar a frequência do sinal através do pico, ou seja, valor mais alto presente no vetor de dados (de amostras) do som captado pelo microfone.

Esta função recebe como parâmetro a FFT do sinal, armazenada na variável *signal\_fft*.

```
[maxPeak, maxFreq] = max(signal_fft);
frequency = maxFreq * Fs / L;
```

A frequência do som captado irá alterar o campo “Frequência captada”. Com ambas as frequências, é calculado o erro percentual – através da função ***get\_error()*** - e, caso este seja dentro de um *range* de 10% pra mais ou pra menos que a frequência da corda devidamente afinada, a *string* do campo “Corda” irá ser alterado para “Afinada”. Caso contrário, será exibida a *string* “Desafinada”. Esta troca do valor das *strings* é comandado pela função ***set\_result()***

Este processo é repetido de forma a captar continuamente o som do microfone e interagir inúmeras vezes dentro do loop de afinação até que o usuário esteja satisfeito com a frequência real da corda tocada.

## 2.2 LABVIEW

O desenvolvimento do afinador no LABVIEW através da construção de blocos de programação do ambiente pode ser observado no Anexo I.

A configuração da captação do som através da placa de som do computador é possível através de blocos do tipo *Sound Input*.

A principal diferença em desempenho, se comparado com o programa de MATLAB, foi que a fácil implementação de um filtro *Butterworth* de ordem 20 - com frequências de corte inferior igual a 80 [Hz] e superior igual a 1000 [Hz] - permitiu a remoção de ruídos do som captado pelo microfone que se somavam à faixa de frequência de interesse para a afinação do violão, permitindo um processamento mais otimizado.

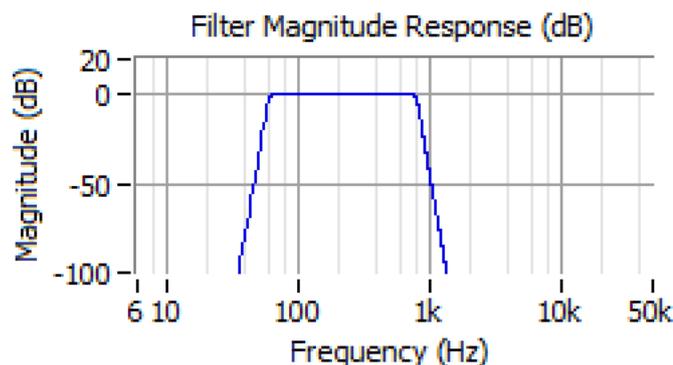


Figura 3 - Filtro Butterworth

A detecção da frequência foi feita por dois blocos. O primeiro, do tipo *Spectral Measurements*, retorna a FFT do sinal e foi cascateado com blocos do tipo *Unbundle by name* e *Power & Frequency estimate* para que a detecção da frequência fosse possível. O segundo bloco de detecção, do tipo *Tone Extraction*, já retorna a frequência detectada e a FFT do sinal.

O programa em LABVIEW também possui uma *interface* gráfica do usuário, que pode ser observada na figura 4.

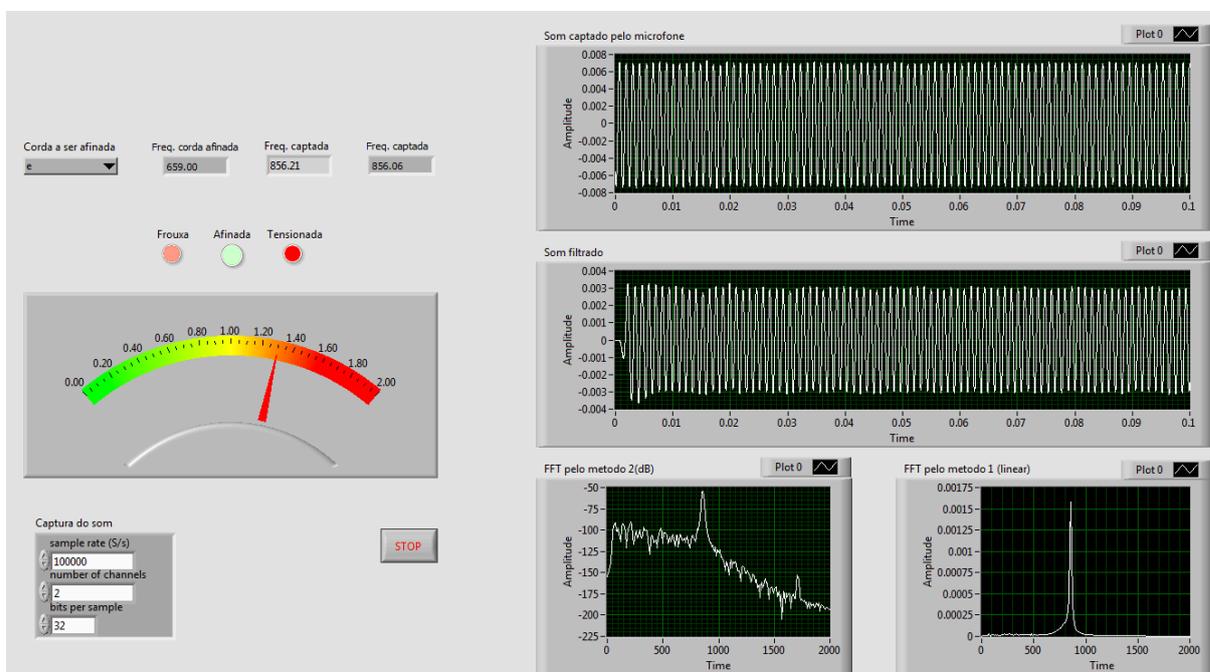


Figura 4 – Interface gráfica do afinador programa no LAVIEW.

Esta também possui um *menu popup* que permite a escolha da corda à ser afinada, possui indicadores da frequência da corda devidamente afinada conforme a

Tabela 1, indicadores das medições pelos blocos dos dois diferentes tipos já mencionados.

Além disso, possui indicadores visuais como “*LEDs*” que mostram se a corda está frouxa, afinada ou muito tensionada e, também, um medido de ponteiro, que indica através da normalização da frequência captada, ou seja, se a divisão desta frequência pela frequência da corda correspondente à tabela 1 é igual a um.

A margem de aceitação de que a corda está devidamente afinada, neste caso, é de 2%. Portanto, caso o cálculo da frequência captada esteja fora do range de  $\pm 2\%$  da frequência correspondente à tabela 1, o medidor de ponteiro e os *LEDs* indicarão que a corda ainda não está afinada.

Os gráficos representam o som captado, o som filtrado, a FFT do sinal captado por ambos os blocos. O primeiro tipo representado em escala linear e o outro em escala logarítmica.

### **3 CONCLUSÕES**

O desenvolvimento de uma mesma ideia em diferentes plataformas de programação agrega habilidades de software, já que a consolidação do algoritmo de funcionamento do projeto é bem semelhante. Desta forma, pode-se focar bastante na exploração do ambiente de desenvolvimento e não despende muito tempo desenvolvendo um novo algoritmo em um novo ambiente.

O MATLAB é uma poderosa ferramenta matemática que também permite a programação de Interfaces Gráficas do Usuário (GUI) e, que, desta forma, permite uma interação mais amigável com o usuário do que a interação através de linhas de comando.

O LABVIEW também demonstrou-se uma ótima ferramenta para processamento de sinais e muito atrativo para leigos em programação, já que o sistema de blocos facilita muito a visualização de variáveis, estruturas, dados etc. do seu programa.

Os afinadores desenvolvidos podem ser utilizados para exemplificar o funcionamento de um produto existente no mercado, porém, para serem embarcados e instalados junto à motores que façam uma afinação automática, ainda devem ser otimizados, principalmente no quesito de filtros de ruídos e do tempo de resposta.

# ANEXO I – BLOCOS DO LABVIEW

