

Pedaleira Digital de Efeitos Para Guitarras

Gabriel da Silva Soares

Faculdade do Gama

Universidade de Brasília

Gama - DF, Brasil

Ricardo Vieira Borges

Faculdade do Gama

Universidade de Brasília

Gama - DF, Brasil

I. RESUMO

O projeto proposto consiste na implementação de uma pedaleira na placa Raspberry Pi 3B que adicione efeitos em tempo real ao som de uma guitarra, buscando a inovação no que se refere a efeitos já existentes no mercado.

II. INTRODUÇÃO

Um pedal ou pedaleira digital é um equipamento capaz de modificar o sinal natural proveniente de uma guitarra, baixo, violão e etc. (cada equipamento voltado para o instrumento utilizado). A ressonância das cordas do instrumento é captada pelos captadores que geram um sinal elétrico que é apenas amplificado ou modificado pela pedaleira para depois ser amplificado dependendo da necessidade. Estes são bastante utilizados em apresentações ou estúdios para gravação e basicamente adicionam efeitos ao sinal por meio de um microprocessador embarcado no equipamento, dependendo da modificação desejada, são adicionados os tipos de filtros aplicados no sinal, que são representados por equações implementadas digitalmente no computador, após a passagem do sinal por estes filtros, é disponibilizada uma saída com o sinal modificado para o amplificador.

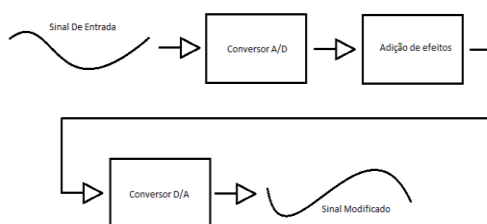


Figura 1 – caminho do sinal.

III. DESENVOLVIMENTO

Primeiramente optou-se pela realização de diversos testes com sinais genéricos de guitarra (formato .WAV) em softwares consolidados como Matlab e Scilab, tentando adicionar edições simples ao sinal como multiplicar por funções, aplicar filtros como o de média móvel, funções de atraso e etc, gerando outro arquivo de mesmo formato, reproduzindo e observando se o resultado sonoro é satisfatório.

No anexo 1 há um código que demonstra a modificação de um sinal por modulação em amplitude e aplicação de raiz quadrada, dessa forma, a saída Y é a combinação do sinal atual e sua cópia atrasada seguindo a equação:

$$Y = \sqrt{x_1 * \cos(\omega_1 t)} + \sqrt{x_1 Z^{-n} * \cos(\omega_2 t)}$$

Alguns efeitos sonoros foram estudados, tais como chorus, delay, flanger, vibrato e os mais comuns [2] para que os efeitos criados tenham alguma distinção dos já existentes.

Nos testes propostos foi utilizado apenas um notebook contendo o software Scilab para modificação dos sinais em formato .WAV.

IV. RESULTADOS

É evidente que o processamento em tempo real envolve outras dificuldades que não foram transpassadas com o teste em arquivo pronto, porém, o intuito da simulação computacional é descobrir efeitos úteis e interessantes para posterior aplicação no equipamento.

Depois de diversos testes de aplicações de funções matemáticas ao sinal, percebeu-se que este só começou a demonstrar diferenças sonoras

significativas com a adição de funções trigonométricas ou atrasos.

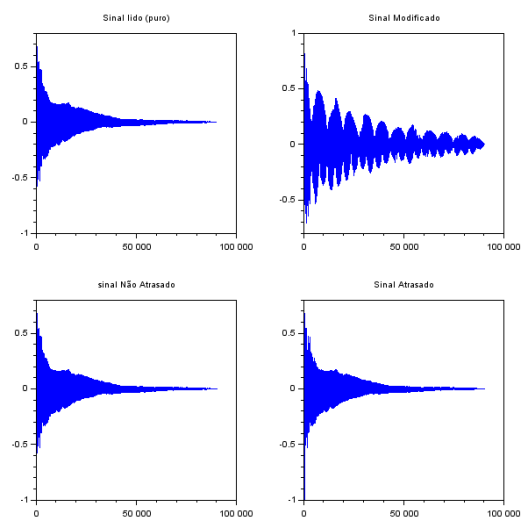


Figura 2: Demonstração gráfica do sinal modificado pela expressão Y.

V. REVISÃO BIBLIOGRÁFICA

[1] Mitch Gallagher. *Guitar Tone:: Pursuing the Ultimate Guitar Sound*. Cengage Learning; p. 81.

[2] A. W. FRANÇA. *Uso de Processamento Digital de Áudio na Implementação de Efeitos em Instrumentos Musicais*. Julho de 2015. http://bdm.unb.br/bitstream/10483/13268/1/2015_AndreWagnerFranca.pdf

```

1 n = 94803; //quantidade-de-amostras
2 T = n - 1; //tamanho-do-vetor
3 step = 1000; //define-a-precisão
4 t_x = 0:(1/step):(T/step); //tempo-para-o-som-normal
5 N = n/10; //atraso
6 t_z = 0:(1/step):((T/step)+(N/step)); //tempo-para-a-escala-atrasada-em-N
7
8
9 s = loadwave("C:\Faculdade\Proc. Sinais\A2.wav");
10 //playsnd(s);
11
12 x1 = [s, zeros(1,N)];
13 x2 = [zeros(1,N), s];
14 z = sqrt(x1.*cos(0.4*t_z)) + sqrt(x2.*sin(0.8*t_z));
15 playsnd(z);
16
17 clf();
18 subplot(221);
19 plot(s);
20 xtitle("Sinal lido (puro)");
21 subplot(222);
22 plot(z);
23 xtitle("Sinal Modificado");
24 subplot(223);
25 plot(x1);
26 xtitle("sinal Não Atrasado");
27 subplot(224);
28 plot(x2);
29 xtitle("Sinal Atrasado");

```

Figura 3: Código gerado para testar a expressão matemática no Scilab.