

Desenvolvimento de filtros no Processador Digital de Sinais TMS320C5502 e Matlab.

Daniel da Rocha Carneiro

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
carneiro.daniel@ufba.br

Gabriel Souza de Andrade

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
gsandrade@ufba.br

Leon Lima Santos

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
leon.lima@ufba.br

Mauricio dos Santos de Jesus

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
mauriciosj@ufba.br

Osmar Pinto Oliveira Junior

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
osmar.pinto@ufba.br

Yago Cunha Martins Rodrigues

DEEC - Departamento de Engenharia Elétrica e da computação
UFBA - Universidade Federal da Bahia
Salvador-BA, Brasil
yago.martins@ufba.br

Resumo — O trabalho detalha a implementação de dois filtros utilizados em guitarra elétrica em ambientes diferentes, sendo eles: o matlab, um código em C para computadores de propósito geral, e a utilização de um DSP.

Palavras-Chave — Transformada de Fourier, ruído, gaussiano, alta frequência, filtro, média móvel, Butterworth.

I. INTRODUÇÃO

O processamento digital de sinais tem sido intensamente utilizado no campo da engenharia de áudio. Uma das suas aplicações recorrentes é a alteração de sinais específicos de áudio para gerar efeitos não inerentes ao sinal. O domínio digital oferece métodos rápidos e mais simples de processamento de áudio, especialmente com o aumento da velocidade e redução de custos de microprocessadores e DSPs.

Neste contexto, existe um grande interesse no desenvolvimento de algoritmos eficientes para plataformas DSP de ponto fixo, como a arquitetura TMS320C55x. A geração de efeitos para guitarra é apenas um dos exemplos

de aplicação, e este problema objetiva a concepção de um processador multiefeitos para o instrumento.

Alguns desses efeitos são Echo e flanger. O Echo é um efeito que repete amostras anteriores de um determinado áudio, mas com atenuação de intensidade. Esse efeito possui alguns parâmetros de controles como *delay* (tempo de atraso), *feedback* (quantidade de repetições) e *effect level* (volume do efeito). O Flanger é uma derivação do Phaser e consiste em interferências destrutivas e construtivas que interagem com a frequência do som emitido, criando novas cristas e vales para onda sonora.

II. EFEITOS DE ÁUDIO

A. Echo

O Delay é um dos efeitos sonoros mais simples de se implementar e é a base para diversos outros efeitos como o Echo.

O Echo consiste em causar um atraso no áudio original, aplicar um ganho, que normalmente é menor do

que 1, e por fim combinar o áudio original e o áudio modificado. Esse processo pode ser visualizado no diagrama de blocos da figura 1.

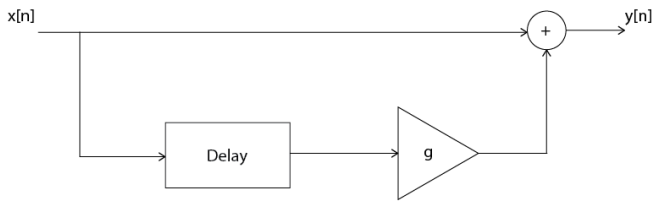


Figura 1 - diagrama de blocos do efeito delay

A implementação do Echo tem a mesma estrutura da implementação do delay porém com a possibilidade de ter mais de um atraso combinado ao áudio original, ou seja, sua implementação consiste em somar ao áudio original N áudios atrasados com diferentes ganhos. É possível ainda entender o delay como um caso específico do Echo onde o N é igual a 1. O processamento do efeito pode ser visualizado no diagrama de blocos da figura 2.

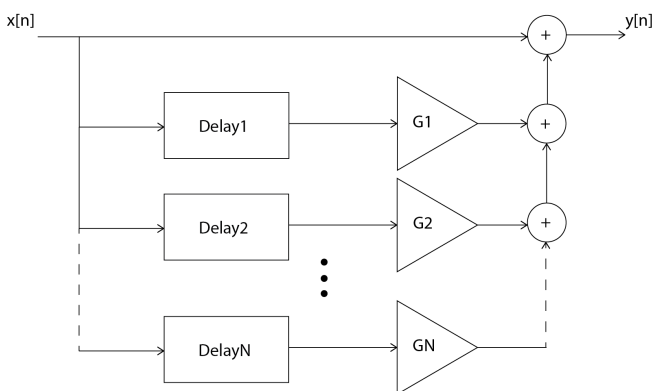


Figura 2 - Diagrama de blocos do efeito echo

B. Flanger

O efeito Flanger é uma combinação de efeitos onde o delay não é constante.

O efeito de flanging é obtido através da mistura de dois sinais de áudio idênticos, onde um deles é atrasado por um intervalo de tempo curto e variável, normalmente menor do que 20 ms. O efeito audível é semelhante a uma turbina de avião por trás do sinal de entrada.

O termo flanging surgiu da forma como o efeito era produzido originalmente. O áudio, até então gravado em fitas, era tocado a partir de duas fontes e simultaneamente.

O engenheiro de áudio então pressionava uma das fitas com o dedo, fazendo com que uma das fontes ficasse com o sinal levemente atrasado em relação à outra. No período de desaceleração da fita, nota-se o efeito do flanging, devido à variação na altura do som. Pressionando a outra fita, o sinal original se atrasa também. A medida que o sinal se aproxima do outro, pode-se notar também o efeito de angling, no outro sentido.

III. SIMULAÇÕES EM MATLAB

A fim de se realizar a comparação da precisão obtida ao se trabalhar com diferentes formas de representação numérica, em especial, entre as de ponto flutuante e ponto fixo, realizou-se a implementação dos filtros escolhidos em Matlab.

Para ambos os filtros selecionados, foram implementadas soluções no formato numérico padrão do Matlab, o *double*, que consiste na representação em ponto flutuante de 64 bits e no formato de ponto fixo de 16 bits. Os resultados finais eram gravados em arquivos de áudio no formato “.wav “, para comparação da percepção sonora, bem como em vetor e forma de onda, para comparação através de erro quadrático médio e gráfica.

Inicialmente, tinha-se como proposta realizar a simulação com pontos flutuantes de 16 bits, a fim de comparar de forma mais próxima da aplicação real em DSP. Porém, como o objetivo inicial do problema é avaliar os erros oriundos da quantização em diferentes formas de representação numérica, manteve-se o formato padrão do Matlab, esperando-se obter diferenças mais significativas no resultado final.

A. Echo

Para o efeito Echo, simulou-se uma condição de apenas uma realimentação com atraso de 0,6 segundos, com um mix de 60% para o sinal original e 40% para o sinal retornado.

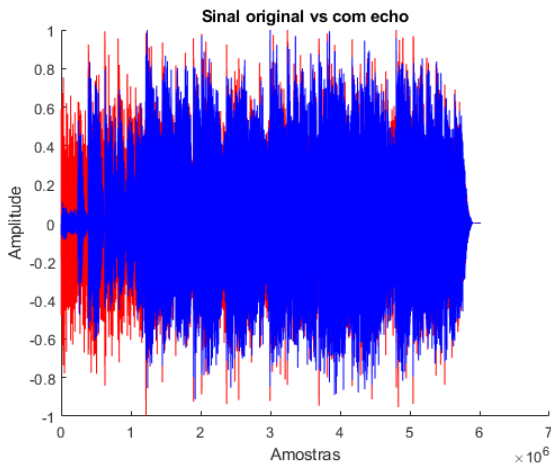


Figura 3. Sinal original (vermelho) e sinal com echo (azul)

Para este efeito em específico, também foi elaborada sua implementação via simulink.

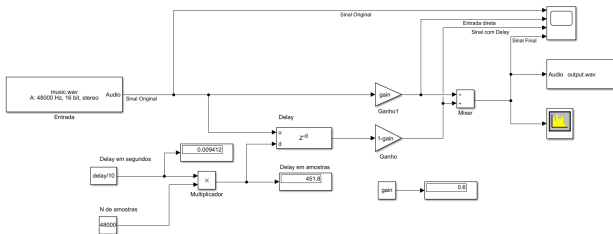


Figura 4. Diagrama de blocos em Simulink

Através do Simulink, foi possível obter em tempo real a aplicação do efeito, sendo possível visualizar sua resposta em frequência, bem como a sobreposição das ondas com e sem atraso.

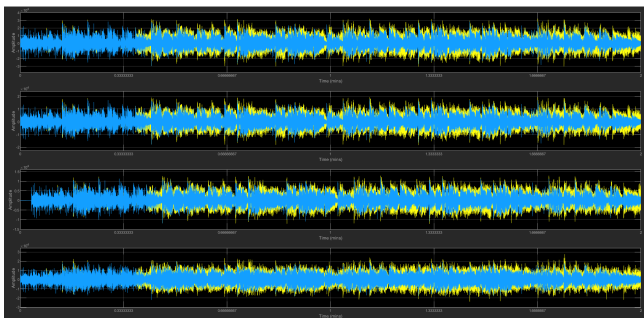


Figura 5. Formas de onda via Simulink

Na imagem acima, estão dispostos em ordem de cima para baixo: o sinal original, o sinal com um ganho de 60% da intensidade original, o sinal com atraso e ganho de 40%

da intensidade original, o sinal final oriundo da união dos dois anteriores.

Analisando as formas de onda, fica evidente a atuação do efeito, replicando alguns pontos característicos da onda com o atraso esperado.

Com o efeito corretamente configurado, buscou-se obter as diferenças entre a aplicação em ponto flutuante de 64 bits e a de ponto fixo de 16 bits. Graficamente e ao escutar o arquivo gerado, não são perceptíveis diferenças. Calculou-se o erro quadrático médio entre os 2 sinais filtrados, obtendo-se o valor de $6,1379 \times 10^{-10}$.

O valor relativamente baixo revela uma similaridade satisfatória entre as duas implementações, indicando que para a aplicação em questão, a solução em ponto fixo de 16 bits, utilizada no DSP, atende ao problema sem maiores desvantagens.

B. Flanger

O mesmo procedimento foi adotado para o efeito Flanger, com exceção da implementação em Simulink.

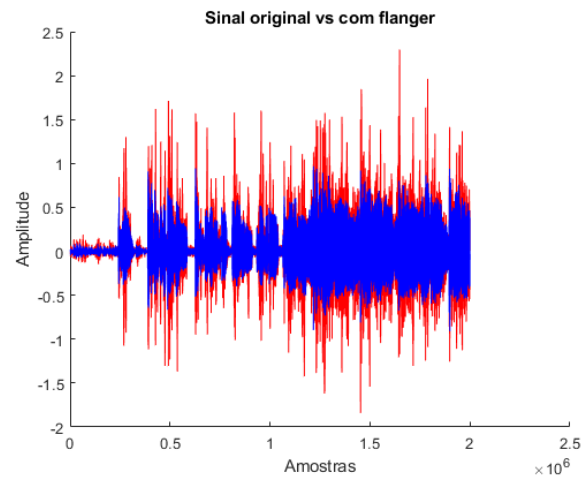


Figura 6. Sinal original (vermelho) e sinal com flanger (azul)

Ajustou-se os parâmetros que interagem com a onda dente de serra a fim de se obter um resultado agradável aos ouvidos. Sendo possível comparar através da imagem acima o sinal original e o filtrado.

Ao se comparar a implementação em *double* e *fixed-point*, obtiveram-se resultados similares aos do efeito Echo. Gráficamente e ao reproduzir o arquivo de áudio, não foram detectadas diferenças perceptíveis. O erro quadrático médio também foi calculado, obtendo-se um valor de $4,1444 \times 10^{-7}$.

O erro obtido foi poucas ordens de grandeza maior em comparação ao do outro efeito. Esta era uma consequência esperada, diante a maior complexidade das operações para aplicação desse efeito, aumentando os erros de quantização. Porém, ainda assim esses erros são praticamente imperceptíveis, também validando o ponto fixo de 16 bits do DSP como uma forma satisfatória para implementação deste efeito.

IV. IMPLEMENTAÇÃO NO DSP.

Em geral, a abordagem no DSP pode ter dois métodos: a utilização de uma memória flash para armazenar o áudio e o resultado da filtragem, ou utilizando o codec AIC3204 presente no kit para gerar o resultado em tempo real.

Para implementar algo que fosse mais voltado a aplicação de filtros de guitarra, foi escolhida a opção de utilizar o codec presente na placa, juntamente com os exemplos já existentes de "line in" que podem ser alterados para gerar um efeito no output ao invés de simplesmente conduzir o sinal.

Para o cenário de testes, foi montado um setup com as conexões dos cabos de áudio ao kit, sendo um de entrada, que pode ser conectado na saída de um computador, e um de saída que pode ser conectado em um fone de ouvido ou caixa de som.

A. Echo

Para implementação do Echo utilizamos um buffer circular que deixa as amostras num mesmo lugar, movimentando apenas os ponteiros de leitura e escrita ao longo do buffer. Desta forma, é necessário apenas manter controle dos valores dos ponteiros. Caso as operações de leitura e escrita sejam realizadas numa mesma posição de

memória. Para implementar o Echo, aloca-se o número necessário de amostras e, lê-se e escreve-se posteriormente na memória, incrementando o ponteiro após a operação de escrita.

B. Flanger

Para poder gerar os efeitos de delay variável do filtro flanger alguns outros parâmetros precisam ser calculados e definidos, e um dos fatores que mais contribuem para o formato da saída é o sinal que controla a variância do delay.

Esse sinal em geral pode ser moldado como um sinal senoidal, um dente de serra, ou até um sinal triangular. A amplitude do sinal sempre fica vinculada ao tamanho máximo do delay.

Portanto, além da adição de alguns parâmetros para controlar a variância do delay, a principal diferença seria justamente o sinal modulador dessa variação.

V. RESULTADOS EXPERIMENTAIS COM DSP

A fim de mostrar a influência dos efeitos, foram criados parâmetros principais para realização do mesmo, tais parâmetros foi possível perceber a alteração nas curvas características do sinal sonoro. A frequência de amostragem utilizada em todas as implementações, f_s , é de 44,1 kHz. E os resultados foram obtidos em tempo real através da porta ADC do dispositivo.

Um dos empecilhos encontrados foi a limitação de memória do kit. Como os dois filtros são baseados em delay, a memória disponível afeta diretamente no tempo máximo que conseguimos dimensionar o delay na aplicação, sendo um limitante para filtros mais complexos.

VI. CONCLUSÃO

Este trabalho teve como principal objetivo projetar e implementar 2 efeitos de áudios para guitarra: Echo, Flanger. A placa TMS320C5502 é um dispositivo de relativo baixo custo, e com incríveis possibilidades de processamento de sinais. O Echo foi o primeiro a ser embarcado, por sua equação da diferença ser bastante

simples. A utilização do buffer circular foi muito importante, não só para este efeito, mas também para o Flanger, este foi mais complicado, pois foi necessário aplicar um LFO senoidal.

Os efeitos de distorção de áudio foram implementados e após alterações nas suas curvas características percebeu-se nos resultados experimentais, como os efeitos alteram o sinal original, tanto no domínio do tempo quanto no domínio da frequência.

VII.REFERÊNCIAS

- [1] Alan V. Oppenheim, Alan S. Willsky com S. Hamid Nawab ; tradução Daniel Vieira; revisão técnica Marcio Eisencraft; Maria D. Miranda. São Paulo : Pearson Prentice Hall, 2010.
- [2] Driemeier, Larissa. Alves, Marcilio e Moura, Rafael T. Análise de Sinais usando MatLab. Escola Politécnica da Universidade de São Paulo, São Paulo.
- [3] Habermeld, Gabriel Barsi. Tutorial de MATLAB para Sinais e Sistemas e Matérias Relacionadas, Junho, 2015.
- [4] Lathi, B.P. Sinais e Sistemas Lineares, 2ª edição, Bookman, 2007.
- [5] Ribeiro, Eduardo Parente. Processamento Digital de Sinais – Laboratório - Introdução ao Octave e Matlab, Setembro, 2021.
- [6] Vásquez, Alfredo Ricci; García, Juan Carlos Bucheli. Implementation of sound effects in DSP. Disponível em: https://www.eit.lth.se/fileadmin/eit/courses/etin80/2016/report_s/sound-effects.pdf. Acesso em: 22 out. 2022
- [7] LATHI, B. P. Modern, Digital and Analog Communication Systems. Inglaterra: Oxford University Press, 1998. Acesso em: 20 out. 2022
- [8] BOULANGER, R.; LAZZARINI, V. The Audio Programming Book. EUA: The MIT Press, 2010 .Acesso em: 20 out. 2022
- [9] SMITH, J. O. Physical Audio Signal Processing. 2010. Disponível em: <http://ccrma.stanford.edu/jos/pasp/i>.
- [10] CHASSAING, R.; REAY, D. Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK. EUA: Wiley-Interscience, 2008. Acesso em: 20 out. 2022
- [11] THOMAS, D. help with hello world, DSK6713. 2012. Disponível em: <https://e2e.ti.com/support/dsp/tms320c6000/high-performance-dsps/f/115/p/215890/763724i>. Acesso em: 30 out. 2022
- [12] SIAHAAN, B. Implementasi Real Time Effect Pada Gitar Berbasis Waktu Tunda /Delay Menggunakan Dsk TMS20C6713. Bacharelado em Engenharia Eletrica | Sepuluh Nopember Institute of Technology, Surabaya, Indonesia, 2010. Acesso em: 20 out. 2022