

## INSTRUMENTAÇÃO DO PROCESSO DE APRENDIZAGEM DAS DISCIPLINAS DE SINAIS E SISTEMAS LINEARES BASEADO NO PROJETO DE UM FILTRO DIGITAL

FILIPE TAVARES CARNEIRO<sup>1</sup>; GUSTAVO COLPES RIBEIRO<sup>2</sup>; MARCELO ESPOSITO<sup>3</sup>

<sup>1</sup>Universidade Federal de Pelotas – filipctavares@gmail.com

<sup>2</sup> Universidade Federal de Pelotas – colpesgustavo@gmail.com

<sup>3</sup> Universidade Federal de Pelotas – marcelo.esposito@ufpel.edu.br

### 1. INTRODUÇÃO

Ao contrário de filtros em tempo contínuo, o desempenho de filtros em tempo discreto não é afetado pela variação dos componentes, temperatura, umidade ou tempo de uso. Além disso, o *hardware* digital é facilmente reprogramado, o que permite a mudança adequada da função do dispositivo. Por exemplo, alguns aparelhos de audição são programados em função da resposta necessária para um dado usuário (LATHI, 2007).

Filtros digitais podem processar sinais contínuos no tempo através de sistemas em tempo discreto, usando interfaces apropriadas na entrada e saída do sistema. A resposta em frequência de um sistema representa sua característica de filtragem. Neste contexto, o gráfico de resposta em magnitude ajuda a avaliar o desempenho e a qualidade de um filtro, no caso de um filtro ideal a resposta em magnitude é uma função retangular com ganho unitário na faixa de passagem e uma atenuação perfeita na banda filtrada.

Tipicamente, filtros em tempo discreto são categorizados como resposta infinita ao impulso (IIR–*Infinite Impulse Response*) ou resposta finita ao impulso (FIR–*Finite Impulse Response*). Um método popular para a obtenção de um filtro IIR em tempo discreto é a partir da transformação de um projeto de filtro correspondente em tempo contínuo. Filtros FIR são flexíveis, sempre estáveis e relativamente fáceis de serem implementados, são essas qualidades que os tornam interessantes. Levando-se em consideração a equação diferença de um filtro FIR a questão é como determinar os coeficientes do filtro da resposta em magnitude desejada (LATHI, 2007).

O objetivo principal do presente trabalho foi aproximar e conciliar de forma prática os conceitos de análise no domínio do tempo de sinais em tempo contínuo usando também o domínio da frequência e sinais em tempo discreto. Para tanto é apresentado o projeto de um filtro FIR.

### 2. METODOLOGIA

A Figura 1(a) mostra parte do algoritmo de projeto do filtro digital no ambiente computacional Matlab (ESPOSITO, 2010). O presente trabalho é baseado no comando “*cfirpm*” que permite especificar restrições arbitrárias no domínio da frequência para o projeto de um filtro FIR. Em “*coef=cfirpm(n,[0 fcutsNorm 1],@lowpass)*”, *n* é a ordem do filtro, *[0 fcutsNorm 1]* é o vetor referente aos limites da banda de frequências, *fcutsNorm* corresponde à frequência de Nyquist normalizada. De acordo com a Figura 1(a), a ordem do filtro utilizado foi 12. Embora a frequência de amostragem dos canais de entrada analógica do *hardware* utilizado (Arduino UNO) tenha como valor padrão a frequência de 10kHz, devido à aplicação

em tempo real do filtro em um algoritmo de controle com atuação na frequência de 4kHz, optou-se por usar como frequência de amostragem o valor de 4kHz. De acordo com o software Matlab neste caso a frequência de Nyquist normalizada deve ser de 2kHz, logo o vetor  $[0 \ 0.002 \ 0.500 \ 1]$  está de acordo com o gráfico de magnitude mostrado na Figura 1(b) para um filtro passa baixa. A Figura 1(d) mostra o projeto mudando-se somente  $fcuts=[4 \ 45]$ , ou seja, reduzindo-se a banda de passagem do filtro.

O aumento do número de coeficientes do filtro torna necessário o armazenamento e o computo em tempo real de uma maior quantidade de valores medidos e não apenas 12, como mostrado na Figura 1 (c), que se refere ao software Arduino usado para confeccionar a aplicação em *hardware* em tempo real. Isso significa que se o usuário deseja que a ação de controle que em malha fechada (MF) ocorre somente após a leitura do sensor, filtragem (se necessária), implementação da lei de controle e escrita no canal de saída, seja da ordem de uma fração de segundo o número de amostras passadas será um impeditivo, pois levará a um atraso de grupo (LATHI, 2007).

```

function coef=ProjFiltro
figure
n=12;
fsamp=4000; % (Hz)
fNiq=fsamp/2; % (Hz)
fcuts=[4 1000]; % (Hz)
fcutsNorm=fcuts/fNiq;
coef=cfirpm(n,[0 fcutsNorm 1],@lowpass);
coef=coef/sum(coef);

freqz(coef);

//Coeficientes do filtro FIR.
//n=12; fsamp=4000; fcuts=[4 45];
//float COEF_FIR [] = { 0.461695784390783, 0.0069059;
//n=12; fsamp=4000; fcuts=[4 1000];
float COEF_FIR [] = { 0.001631651451766, 0.009792467;

datatofilter[12] = datatofilter[11];
datatofilter[11] = datatofilter[10];
datatofilter[10] = datatofilter[9];
datatofilter[9] = datatofilter[8];
datatofilter[8] = datatofilter[7];
datatofilter[7] = datatofilter[6];
datatofilter[6] = datatofilter[5];
datatofilter[5] = datatofilter[4];
datatofilter[4] = datatofilter[3];
datatofilter[3] = datatofilter[2];
datatofilter[2] = datatofilter[1];
datatofilter[1] = datatofilter[0];
datatofilter[0] = analogRead(PIN_HALL_SENSOR);

filteredata = datatofilter[0]*COEF_FIR [0] + datatofilter[1]*COEF_FIR [1] + datatofilter[2]*COEF_FIR [2] + datatofilter[3]*COEF_FIR [3] + datatofilter[4]*COEF_FIR [4] + datatofilter[5]*COEF_FIR [5] + datatofilter[6]*COEF_FIR [6] + datatofilter[7]*COEF_FIR [7] + datatofilter[8]*COEF_FIR [8] + datatofilter[9]*COEF_FIR [9] + datatofilter[10]*COEF_FIR [10] + datatofilter[11]*COEF_FIR [11] + datatofilter[12]*COEF_FIR [12];

```

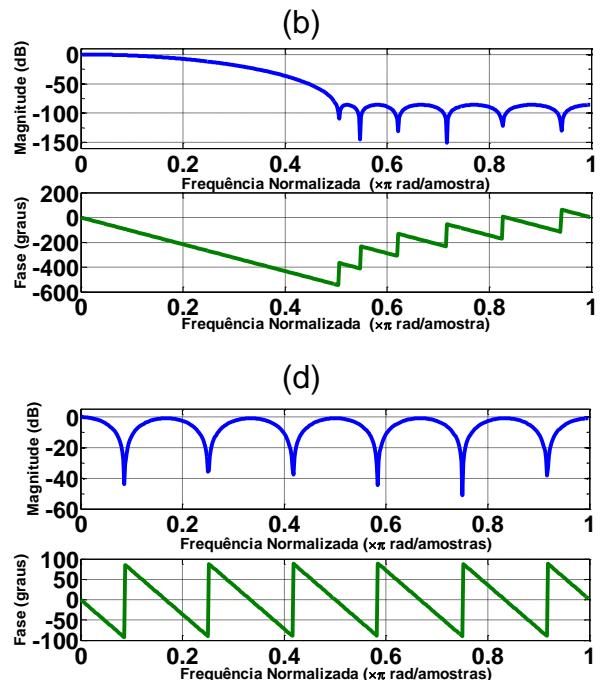


Figura 1 – (a) Projeto de um filtro FIR no Matlab, (b) e (d) gráficos de módulo e fase de um filtro passa baixa usando o comando *cfirpm* do Matlab, 1kHz e 45Hz, respectivamente e (c) parte do algoritmo do filtro digital usado na aplicação em tempo real com o software Arduino.

Na Figura 2(a) o número (1) indica o módulo Analog Discovery que em conjunto com o software WaveForms, disponibiliza ao usuário um osciloscópio e um gerador de funções portátil, de dois canais, entre muitas outras funções. O número (2) indica o kit de desenvolvimento Arduino UNO. Um canal do osciloscópio foi ligado no pino 11 que de acordo com o código utilizado corresponde a uma saída digital com modulação por largura de pulso (PWM - *Pulse Width Modulation*) o outro canal do osciloscópio foi ligado em um dos canais do gerador de funções e no pino A1 do Arduino que é uma entrada analógica. Todos os terminais de terra (GND) foram conectados a um mesmo nó. Na Figura 2(b)

Na Figura 2(b) o número (1) indica a fonte de alimentação de corrente contínua (CC), (2) é um amplificador de instrumentação, (3) é uma esfera magnética (4) é um sensor magnético de efeito Hall (5) é o Arduino UNO. Este circuito é semelhante ao mostrado na Figura 2(a), a diferença é a substituição do sinal proveniente do gerador de funções pelo obtido a partir do sensor magnético. Com isso pode-se analisar o comportamento dinâmico de diversas variáveis como o tempo de processamento necessário para a aquisição de dados, o processamento do filtro, a execução da rotina de controle em MF entre outras rotinas que se façam necessárias para o emprego de um filtro digital em um sistema microcontrolado. No presente trabalho algoritmo em MF utilizado é o apresentado por ASSENZA (2011).

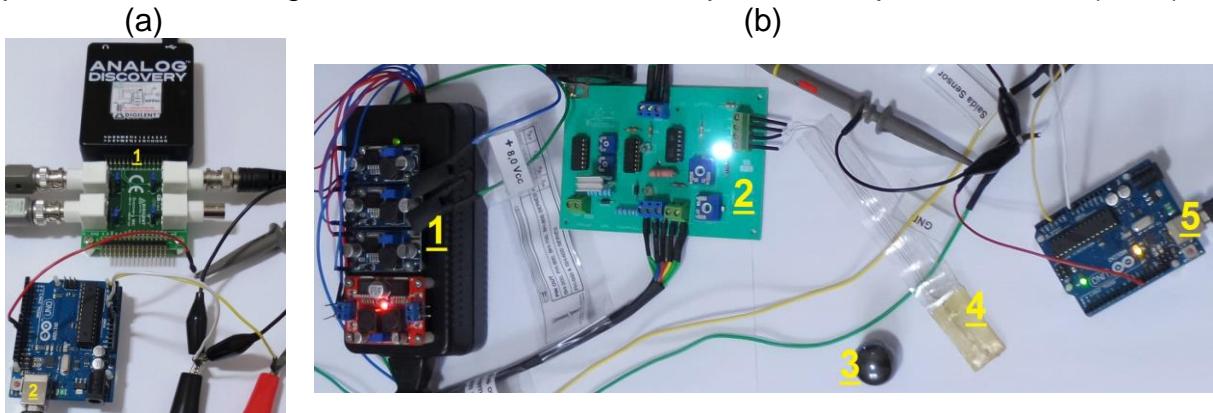


Figura 2 – (a) Circuito utilizado para a simulação de uma fonte CC com ruído adicionado ao sinal e (b) unidade experimental utilizada na análise da qualidade do filtro digital projetado.

### 3. RESULTADOS E DISCUSSÃO

A Figura 3 (a) mostra os parâmetros utilizados para se chegar à forma do sinal CC mostrado na Figura 3 (b). Trata-se de um sinal em corrente contínua de 3,5V que possui adicionada a ele uma onda triangular modificada, com frequência de 20Hz. A onda triangular representa uma fonte de ruído e serve de teste para o filtro FIR projetado. Utilizando o comando *Serial.print* (Figura 1 (c)) para a aquisição de dados e observando os gráficos de magnitude na Figura 1(b) e 1(d) percebe-se que o resultado da leve atenuação dos picos mostrado na Figura 4(a) está coerente com o fato de que neste caso a banda de passagem é estreita, até 45Hz e como o filtro não é ideal a atenuação está presente. No caso da banda de passagem com limite máximo de 1000 Hz (Figura 4(b)) o filtro não exerceu sua função. Reduzindo o valor máximo da banda para 5Hz o resultado foi similar ao mostrado na Figura 4(a), ou seja, o ruído não foi totalmente atenuado, indicando que o número de coeficientes do filtro precisa ser ajustado caso este seja o objetivo. Os resultados da adição do

filtro FIR (Figura 2(b)) no trabalho de ASSENZA (2011) forneceram a mesma conclusão.

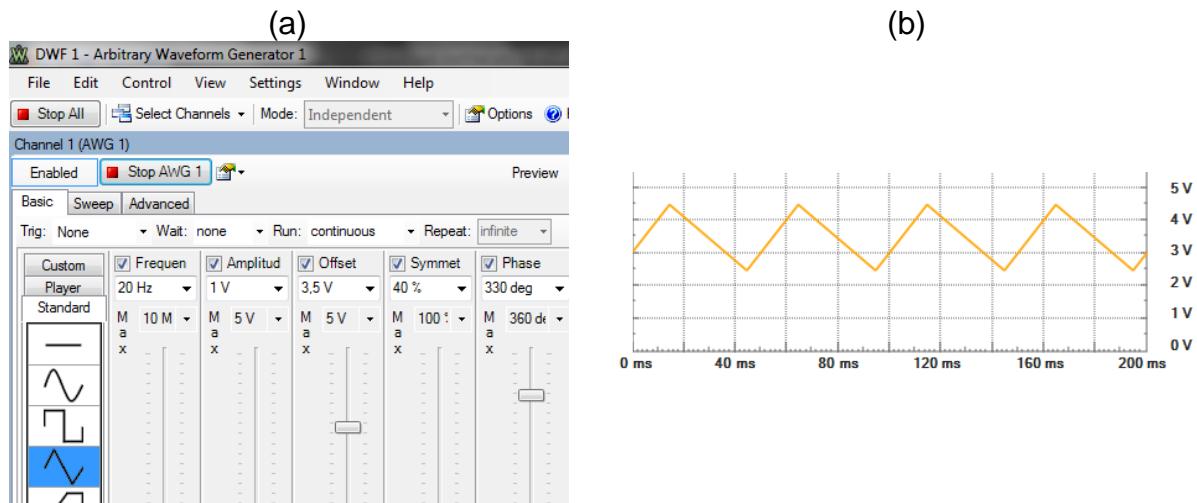


Figura 3 – (a) Gerador de funções no Waveform e (b) sinal gerado.

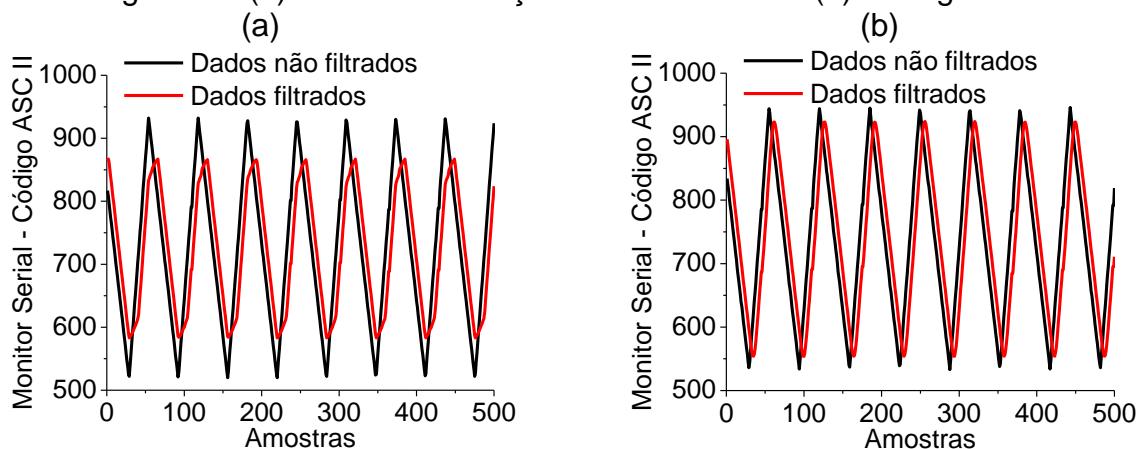


Figura 4 – Resposta do FIR:  $n=12$ ,  $fsamp=4000$ , (a)  $fcuts=[4 \ 45]$  e (b)  $fcuts=[4 \ 1000]$ .

#### 4. CONCLUSÕES

Abordaram-se de forma ampla e com sucesso todos os conceitos mencionados nos planos de ensino das disciplinas de sinais e sistemas lineares, um e dois, ministradas para os cursos de Engenharia de Controle e Automação e Engenharia Eletrônica da Universidade Federal de Pelotas. A base conceitual utilizada foi o próprio livro texto da disciplina. Em termos práticos a deficiência ficou por conta da utilização do monitor serial do hardware Arduino como forma de verificação dos dados com e sem filtro. Uma vez que foi cumprida a exigência de aplicação do filtro digital em um sistema em malha fechada em tempo real, a melhora desta verificação torna-se apenas um objetivo para trabalhos futuros.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- ASSENZA, E., **Magnetic Levitation \* MagneticLevitation.ino**, Acessado em 13 jul. 2015. Disponível em: <http://forum.arduino.cc/index.php?topic=89241.0>
- ESPOSITO, M., **Instrumentação e Monitoramento em Tempo Real de Reatores de Polimerização Através de Técnicas Calorimétricas**. 2010. 155f. Tese (Doutorado em Engenharia Química). Universidade Federal de Santa Catarina.
- LATHI, B. P. **Sinais e Sistemas Lineares**. São Paulo, Bookman, 2007.