

Universidade Federal da Bahia (UFBA)
Departamento de Engenharia Elétrica - Escola Politécnica
Matéria: ENGG63 – Processamento Digital de Sinais
Professor: Antônio Carlos L. Fernandes Jr.

**PROJETO DE FILTRO DISCRETO PASSA-BAIXA A PARTIR DE UM FILTRO
PASSA-BAIXA BUTTERWORTH PELO MÉTODO DA INVARIÂNCIA DA
RESPOSTA IMPULSIVA**

Caetano P. da Silva Neto
Acácio Souza de Azevedo

I. RESUMO

Este documento tem como principal objetivo apresentar os passos realizados para construção do filtro descrito no título de acordo com as especificações apresentadas no Problema 1 proposto, bem como descrever as técnicas e ferramentas utilizadas no processo. A ideia é utilizar a ferramenta MATLAB como forma para apresentar os gráficos obtidos, verificando o comportamento do Filtro Butterworth nas etapas do projeto.

II. INTRODUÇÃO

Os Filtros são sistemas lineares invariantes no tempo capazes de modificar as características dos sinais conectados em sua entrada, de modo que, apenas uma parcela específica dos componentes de frequência do sinal chega à saída do filtro. Para os filtros digitais, utilizam um processador digital para executar cálculos numéricos em valores amostrados do sinal de entrada. O processador pode ser um computador ou um DSP. Em um processo de filtragem digital, o sinal analógico deve ser primeiramente digitalizado usando um ADC. Isto quer dizer que, a cada intervalo de tempo previamente definido é retirada uma amostra do sinal de entrada que vai ser codificada em forma binária e este procedimento é aplicado sucessivamente a cada novo intervalo de tempo.

O processo de filtragem de sinais pode ser realizado digitalmente, na forma esquematizada pelo diagrama apresentado a seguir:

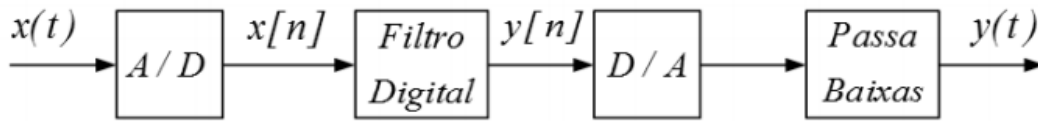


Figura 1. Diagrama de Bloco da Filtragem de Sinais Analógicos para Digitais.

Os Filtros Digitais podem ser caracterizados de duas formas distintas, dependendo da duração da sequência $y[n]$ quando aplicado em sua entrada um sinal do tipo impulso. A primeira, cuja resposta ao impulso apresenta resposta finita (FIR – Finite Impulse Response), para uma entrada um sinal do tipo impulso e a segunda, cuja resposta ao impulso apresenta resposta infinita (IIR – Infinite Response Impulse), para uma entrada do mesmo tipo.

Para esse trabalho será utilizado o filtro IIR dada à especificação que é pelo método da invariância impulsiva. A projeção desse será dada utilizando-se uma abordagem clássica onde será construído um filtro analógico, então realizada uma transformação para filtro digital usando mapeamento de valores.

Em nossas discussões que serão apresentadas neste projeto, focaremos principalmente os filtros passas-baixas seletivos em frequência utilizando o filtro Butterworth, este servirá como protótipo de filtro analógico para obtenção do filtro digital.

III. FILTRO DIGITAL IIR

A resposta de um filtro de Resposta Infinita ao Impulso (IIR) é função dos sinais de entrada presentes e passados, e dos sinais de saída passados. A equação diferencial representa um filtro IIR porque é uma função dos elementos de excitação e resposta. A dependência das saídas passadas (recursividade) faz com que a duração da resposta seja infinita, mesmo quando cessaram os sinais de entrada. As equações a seguir retratam esse enunciado:

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k];$$

[eq. 01]

Utilizando uma representação recursiva temos que:

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k] = \sum_{k=0}^N b_k x[n-k] - \sum_{k=1}^M a_k y[n-k]; \quad [\text{eq. 02}]$$

Sendo a_k e b_k os coeficientes do filtro.

Podemos encontrar a função transferência $H(z) = Y(z)/X(z)$ para o filtro IIR através da transformada z, onde vamos obter:

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}} \quad [\text{eq. 03}]$$

Desta forma, resumidamente, nos filtros IIR a saída do sistema no instante t depende não somente das entradas anteriores, mas também das saídas. Conforme mostrado no diagrama de bloco desse filtro:

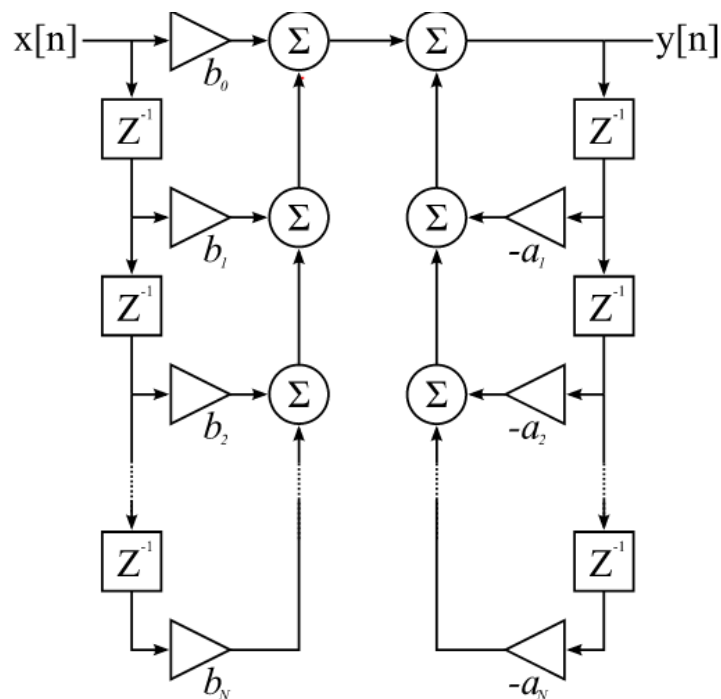


Figura 2. Diagrama Bloco Filtro IIR.

Existem diversas implementações digitais e estruturas para um mesmo problema, sendo elas equivalentes quando ao cálculo da saída do filtro. Iremos abordar e implementar um filtro IIR utilizando estruturas nas Formas Direta e em Cascata. A seguir será apresentado essas estruturas de forma simplificada.

a. FILTRO IIR – FORMA DIRETA

Para a forma Direta podemos representar ao mesmo tempo $N(z)$ e $1/D(z)$, na qual os coeficientes do filtro estão explícitos. Segue o diagrama da Forma Direta:

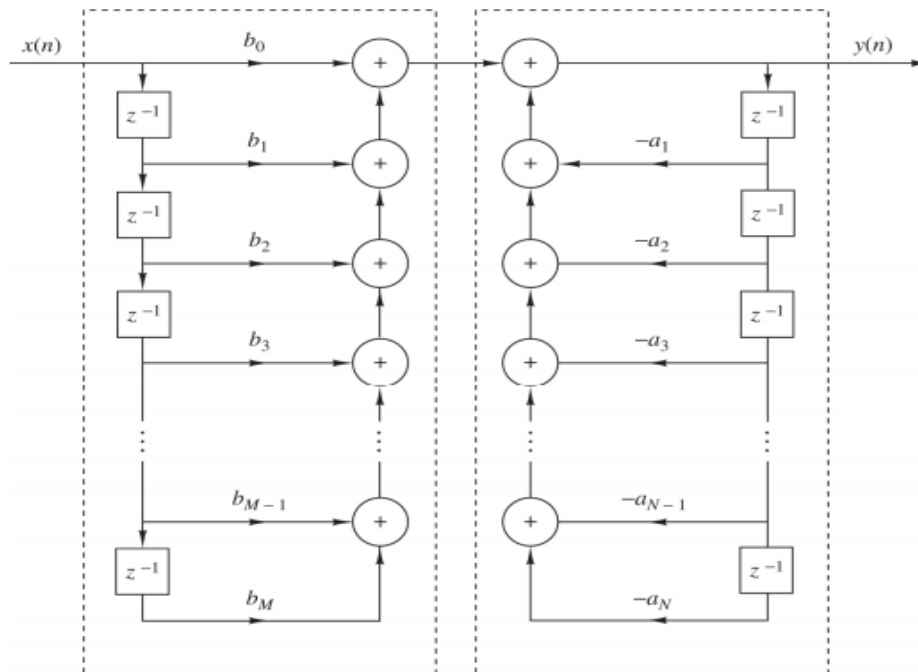


Figura 3. Diagrama de Bloco Forma Direta.

A partir do diagrama teremos a seguinte equação:

$$y(n) = - \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k)$$

[eq. 04]

Podemos destacar para as estruturas na Forma Direta os aspectos que elas são sensíveis a parâmetros de quantização e quando N é grande, uma pequena mudança em um coeficiente

do filtro devido ao parâmetro de quantização resulta em uma grande mudança na locação dos polos e zeros do sistema.

b. FILTRO IIR – FORMA EM CASCATA

Considerando um sistema IIR assumindo que $N \geq M$ e o sistema pode ser fatorado em subsistemas de segunda ordem em cascata, tal como expresso em $H(z)$:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad H(z) = H_1(z)H_2(z) \quad H(z) = \prod_{k=1}^K H_k(z)$$

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}}$$

[eq. 05]

Onde k é a parte inteira de $(N+1)/2$ e O parâmetro b_0 pode ser distribuído igualmente entre as k seções do filtro, tal que $b_0 = b_{10} b_{20} \dots b_{k0}$, dessa forma teremos o seguinte esquema:

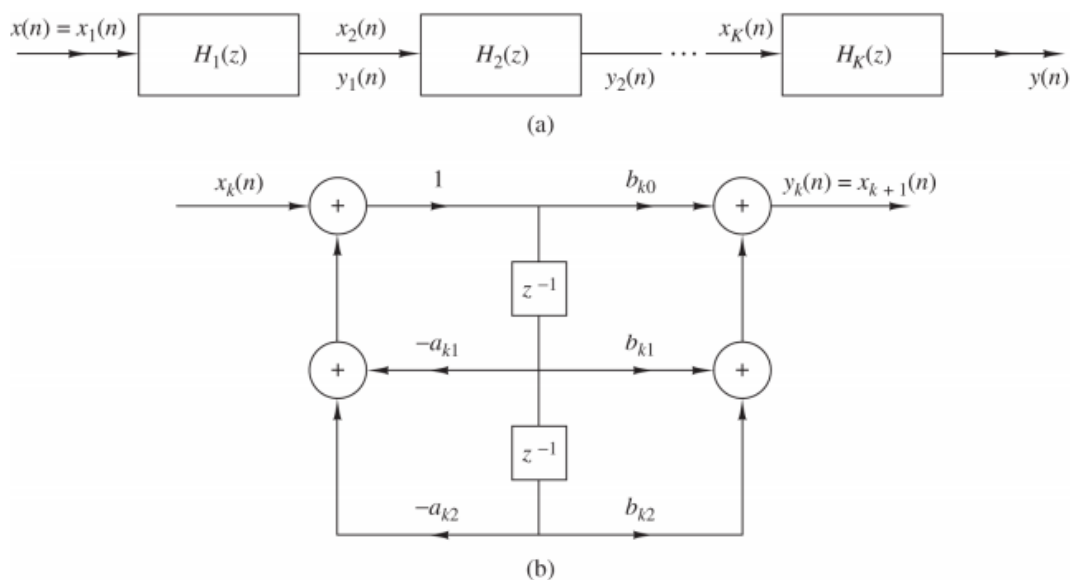


Figura 4. Diagrama de Bloco Forma em Cascata.

Utilizaremos esses conceitos para aplicação na implementação levantada pelo problema. Agora iremos entender os filtros Butterworth.

IV. FILTROS BUTTERWORTH

Os filtros passa-baixa Butterworth são especificados de modo a terem uma função de transferência com o mínimo de oscilações tanto na banda passante como na banda de corte.

A principal característica desse filtro é que sua resposta em magnitude é flat (plana) na banda de passagem e de corte. Sua resposta quadrática de magnitude de N ésima ordem será dada pela expressão abaixo.

$$|H_a(j\Omega)|^2 = 1 / 1 + (\Omega / \Omega_c)^{2N} \quad [\text{eq. 06}]$$

Observando o gráfico a seguir podemos tomar algumas observações com relação a esse filtro.

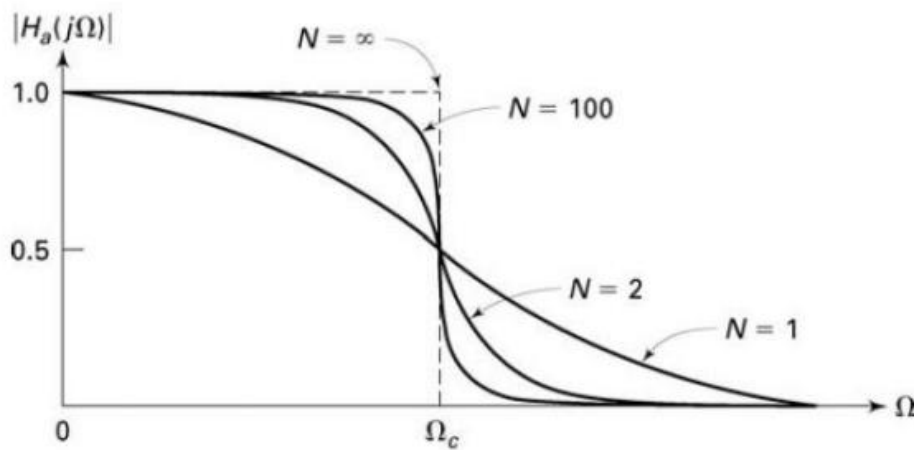


Figura 5. Gráfico da Equação 06.

Sabendo-se que N é ordem do filtro e Ω_c é a frequência de corte, temos:

- Em $\Omega = 0$, $|H_a(j0)|^2 = 1$, $\forall N$;
- Em $\Omega = \Omega_c$, $|H_a(j\Omega_c)|^2 = 0,5$, $\forall N$, o que implica 3db de atenuação e Ω_c ;
- $|H_a(j\Omega_c)|^2$ é uma função monotonicamente decrescente em Ω ;
- $|H_a(j\Omega_c)|^2$ se aproxima de um filtro passa-baixa ideal em $N \rightarrow \infty$;
- $|H_a(j\Omega_c)|^2$ é maximamente plano em $\Omega = 0$.
- É visualmente observável que a faixa transição diminui de forma inversamente proporcional ao aumento da ordem do filtro.

Para a realização do projeto de um filtro digital a partir de um protótipo analógico se faz necessário converter $h(t)$ em $h(n)$ ou $H(s)$ em $H(z)$. Isso pode ser escrito como:

$$H(z) = H_a(s)|_{s=m(z)} \quad [\text{eq. 07}]$$

Onde $H(z)$ é a função sistema no domínio discreto, $H_a(s)$ é a função sistema no domínio analógico e $s=m(z)$ é uma função de mapeamento.

No filtro passa-baixa butterworth sua função $H_a(s)$ é definida como:

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2 \quad |\Omega=s/j = 1/(1+(s/j\Omega_c)^{2n}) = (j\Omega)^{2n}/s^{2n} + (j\Omega_c)^{2n} \quad [\text{eq. 08}]$$

Podemos destacar algumas características do filtro butterworth como as suas Respostas suaves em todas as frequências, decréscimo monotônico na frequência de corte, a frequência de 3 dB corresponde à frequência de corte e a resposta plana nas bandas de corte e de passagem.

V. TRANSFORMAÇÕES DE FILTROS ANALÓGICOS EM DIGITAIS

As transformações são capazes de preservar diferentes características de filtros analógicos em projetos de filtros digitais (MITRA, 1998). Temos diferentes tipos de técnicas para realizar tais transformações. A técnica que desenvolveremos nesse trabalho é a Invariância ao Impulso, que segue a explicação de sua implementação a seguir.

a. TÉCNICA DE INVARIÂNCIA AO IMPULSO

Para conseguir uma representação de resposta ao impulso o mais próximo possível de um filtro analógico seletivo em frequência fazemos uso de uma técnica conhecida como invariância ao impulso. A amostragem resultante é definida por:

$$H(n) = H_a(nT) \quad [\text{eq. 09}]$$

Tomando $\omega = \Omega T$ ou $e^{j\omega} = e^{j\Omega T}$, onde ω é a representação da frequência em tempo discreto e Ω é a frequência de tempo contínuo. Uma vez que $z=e^{sT}$ se encontra no círculo de raio unitário e $s = j\Omega$ no eixo imaginário, tem-se a seguinte transformação do plano s (analógico) no plano z (digital):

$$z = e^{sT} \quad [\text{eq.10}]$$

A consequência desse mapeamento, no plano complexo, apresenta as seguintes considerações:

- Para $\sigma < \text{Re}(s)$, temos:
 1. $\sigma < 0$, mapeamento $|z| < 1$ (interior do círculo unitário)
 2. $\sigma = 0$, mapeamento $|z| = 1$ (no círculo de raio unitário)
 3. $\sigma > 0$, mapeamento $|z| > 1$ (fora do círculo unitário)
- Todo semi-plano s de comprimento $2\pi/T$ é mapeado em $|z| < 1$. Este mapeamento não é único, mais de vários para um;
- "Uma vez que todo o semiplano s esquerdo é mapeado dentro do círculo de raio unitário, um filtro analógico causal e estável é mapeado em um filtro digital também causal e estável"
- Se $H_a(j\Omega) = H_a(j\omega/T) = 0$ para $|\Omega| \geq \pi/T$, então $H(e^{j\omega}) = (1/T)H_a(j\omega/T)$, $|\omega| \leq \pi$.

VI. ETAPAS DO PROJETO DE UM FPB PELA TÉCNICA DE INVARIÂNCIA AO IMPULSO

As etapas para construção são dadas na ordem a seguir:

1. Escolher T e determinar as frequências de corte Ω_p da faixa de passagem e Ω_s da faixa de rejeição, ambas dada por $\omega = \Omega T$;
2. Projetar um filtro analógico $H_a(s)$ usando as especificações Ω_p , Ω_s , δ_p , A_s . Em que δ_p é o ripple da faixa de passagem e A_s é a atenuação na faixa de rejeição;
3. Usando expansão em frações parciais, expandir $H_a(s)$ a em:

$$H_a(s) = \sum_{k=1}^N \left(\frac{R_k}{s - P_k} \right),$$

4. Projetar um filtro analógico passa-baixas com as frequências de corte encontradas no passo 1 e com o ripple da faixa de passagem e a atenuação na faixa de rejeição do passo 2;
5. Aplicar a técnica de invariância ao impulso ao filtro projetado no passo 3.

VII. ENUNCIADO PROJETO 1

Projetar, a partir de um filtro passa-baixas Butterworth de sexta ordem, um filtro discreto passa-baixas com frequência de corte (-3dB) igual a $\omega_c=2\pi/3$, usando o método da invariância da resposta impulsiva. Fazer a implementação sob forma direta e sob forma em cascata. Represente os coeficientes em ponto flutuante (ex.: $0,00423578 = 0,423578 \times 10^{-2}$) e vá diminuindo o número de casas decimais após a vírgula nas forma direta e em cascata para verificar a sensibilidade a quantização de parâmetros. Trace a curva do módulo de resposta em frequência em dB para os casos de precisão infinita e precisão finita. Em seguida, para a representação em forma direta, escolha duas das transformações em frequência a seguir ($Z^{-1} = -z^{-1}$; $Z^{-1} = z^2$ ou $Z^{-1} = -z^{-2}$) e trace a curva do módulo em dB da resposta em frequência resultante.

VIII. IMPLEMENTAÇÃO E CÓDIGO FONTE MATLAB

Para desenvolvimento desse projeto foi utilizado o matlab em conjunto aos conceitos adquiridos em sala de aula bem como os provindos de fontes de pesquisas.

A implementação desse filtro se deu a partir de um filtro butterworth analógico de 6º ordem que conseguimos implementar baseado em comandos nativos do matlab como segue descrito abaixo:

```

% -----
% PROJETO DO FPB ANALÓGICO BUTTERWORTH
% -----
%Valores base
N = 6;
Wc = 2.094395102;

% O parâmetro Td (sampling interval) cancela no metodo de invariância ao
% impulso e portanto seu valor é 1
Td = 1;

% Projetando o filtro butterworth.
% b: array de zeros
% a: array de polos
% O s indica que está trabalhando com um filtro analógico e portante Wc
% pode ser maior que 1
[B,A] = butter(N, Wc, 's');

% função de transferência do filtro analógico
ft_Hz_analogico = tf(B, A);

% geração do gráfico
figure
title(sprintf('Gráfico do FPB Butterworth Analógico '))
set(gcf, 'NumberTitle', 'off');
set(gcf, 'Name', 'Grafico do FPB Butterworth Analógico');
freqs(B,A)

```

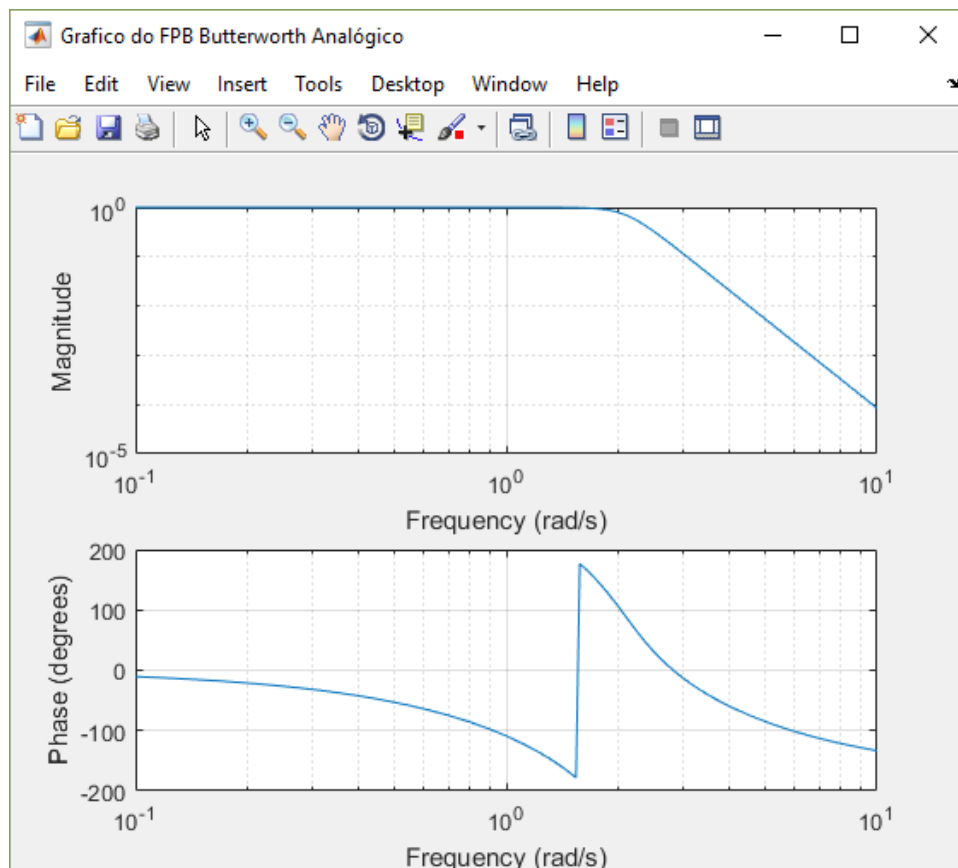


Figura 6. Gráfico do filtro butterworth analógico

```
%-----  
% REPRESENTAÇÃO DO FILTRO NO TEMPO DISCRETO  
% -----  
% >>>>>>>>>>> Implementação pela forma direta <<<<<<<<<<<<  
% -----
```

% Construção da junção de transferência a partir dos numeradores e % denominadores extraídos pelo tfdata

ft_Hz_Direta = tf(Bz_discreto, Az_discreto, Td);

```
figure
set(gcf, 'NumberTitle', 'off');
set(gcf, 'Name', 'Grafico do Filtro Discreto - Forma Direta');
title(sprintf('Grafico do Filtro Discreto - Forma Direta'))
freqz(Bz_discreto,Az_discreto)
```



A implementação da forma em cascata se deu através do artifício de criar uma nova função que realizasse a codificação necessária para tal. É possível encontrar a implementação

```
% -----  
% >>>>>>>>>>> Implementação pela forma em cascata <<<<<<<<<<<<  
% -----  
  
% chamada da funcao que realiza a implementação em forma de cascata  
[Bz_cascata, Az_cascata] = impinvarByCascata(Bz_discreto,Az_discreto, Td);  
  
  
% Função de transferência dada pela forma em cascata  
ft_Hz_cascata = tf(Bz_cascata, Az_cascata, Td);  
  
  
% gerando o gráfico  
figure  
set(gcf, 'NumberTitle', 'off');  
set(gcf, 'Name', 'Grafico do Filtro Discreto - Forma Cascata');  
title(sprintf('Grafico do Filtro Discreto - Forma Cascata'));  
freqz(Bz_cascata,Az_cascata);
```



Aqui é observado que ambos os métodos são sensíveis à variação das casas decimais e portanto, ambas as curvas são sensíveis ao erro de quantização de forma semelhante, ao menos seguindo a observação do gráfico.

```
% -----
% SENSIBILIDADE E QUANTIZAÇÃO DOS PARÂMETROS
% -----

% Loop para redução dinâmica das casas decimais.
for j=5:-1:1
    % Limitação do denominador e numerador para forma direta
    [Bz_discreta_direta, Az_discreta_direta]= tfdata(ft_Hz_Direta, 'v');
    Bz_direto_rounded = round(Bz_discreta_direta, j);
    Az_direto_rounded = round(Az_discreta_direta, j);
    % Limitação do denominador e numerador para forma em cascata
    [Bz_discreta_cascata, Az_discreta_cascata]= tfdata(ft_Hz_cascata, 'v');
    [Bz_cascata_rounded, Az_cascata_rounded] =impinvarByCascataRounded(Bz_discreta_cascata, Az_discreta_cascata, Td, j);
    % geração dos gráficos - Comparador em ordem de casas decimais
    graph_comp_decimals = fvtool(Bz_direto_rounded, Az_direto_rounded, Bz_cascata_rounded, Az_cascata_rounded);
    title(sprintf('Comparação dos filtros para %d casas decimais',j))
    set(gcf,'NumberTitle','off');
    legend(graph_comp_decimals, 'Forma direta', 'Forma cascata');
    hold on
end
hold off;
```

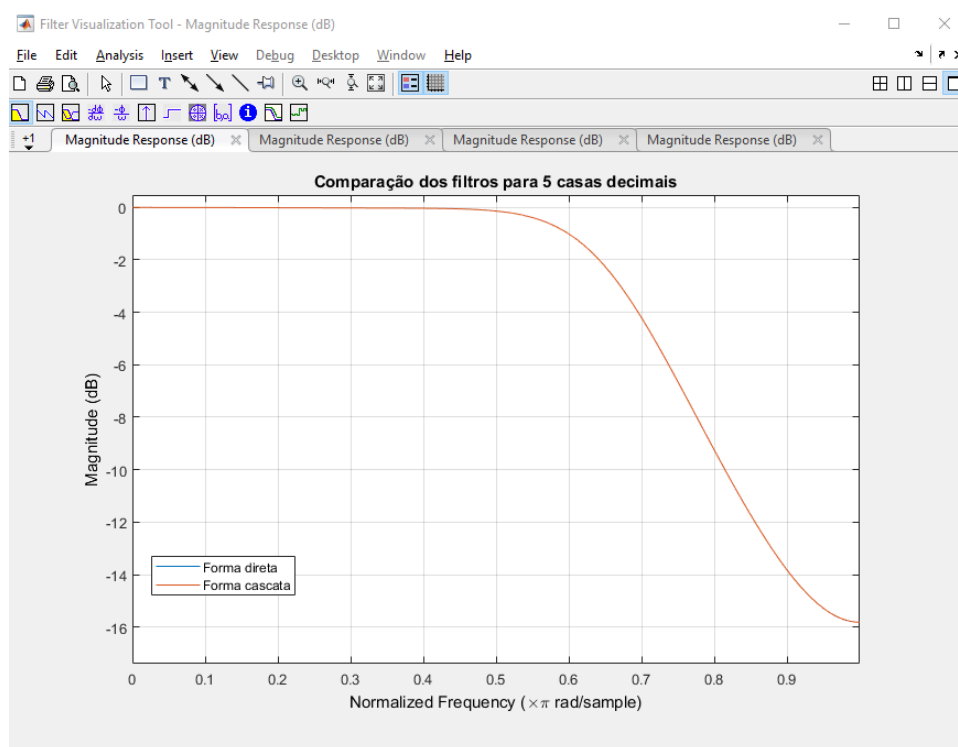


Figura 9.1. Comparação forma direta com 5 casas decimais

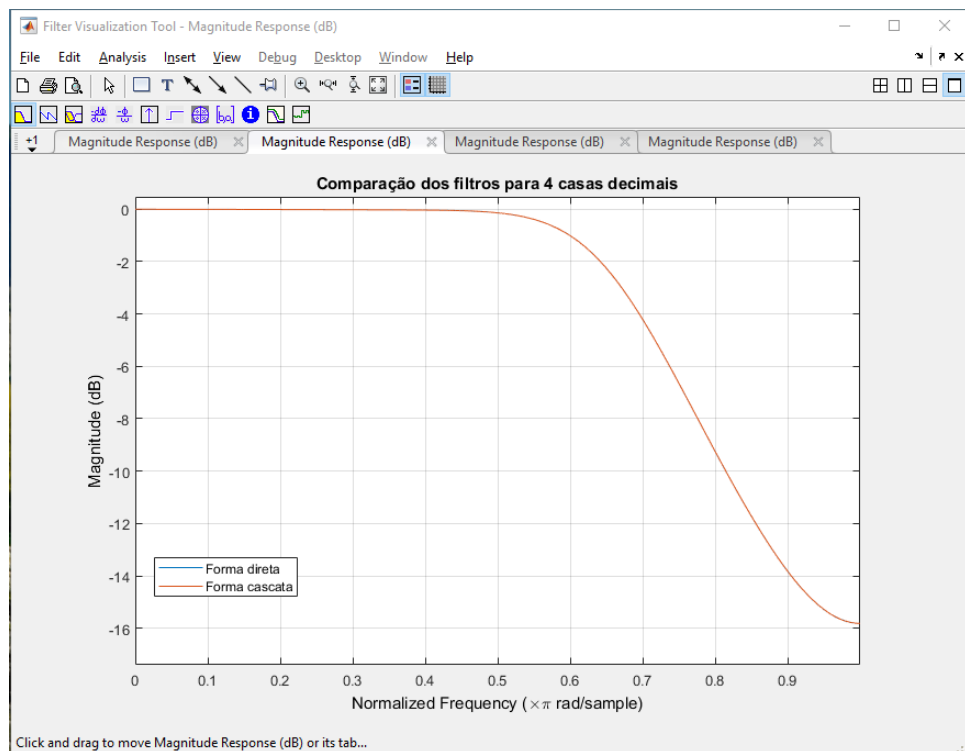


Figura 9.2. Comparação forma cascata com 4 casas decimais

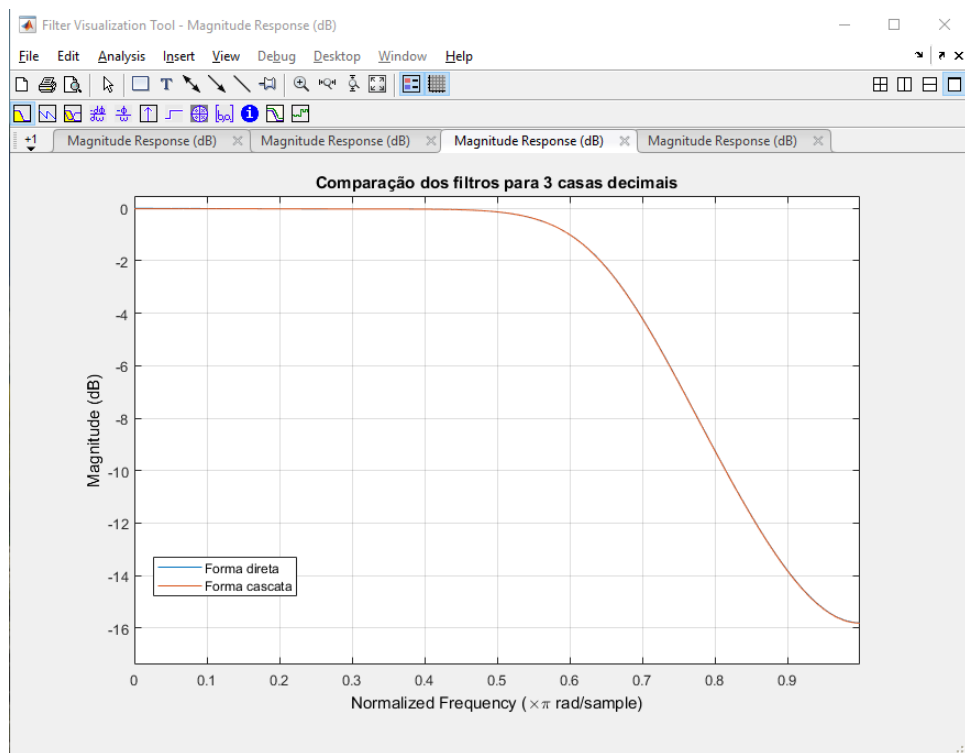


Figura 9.3. Comparação forma direta com 3 casas decimais.

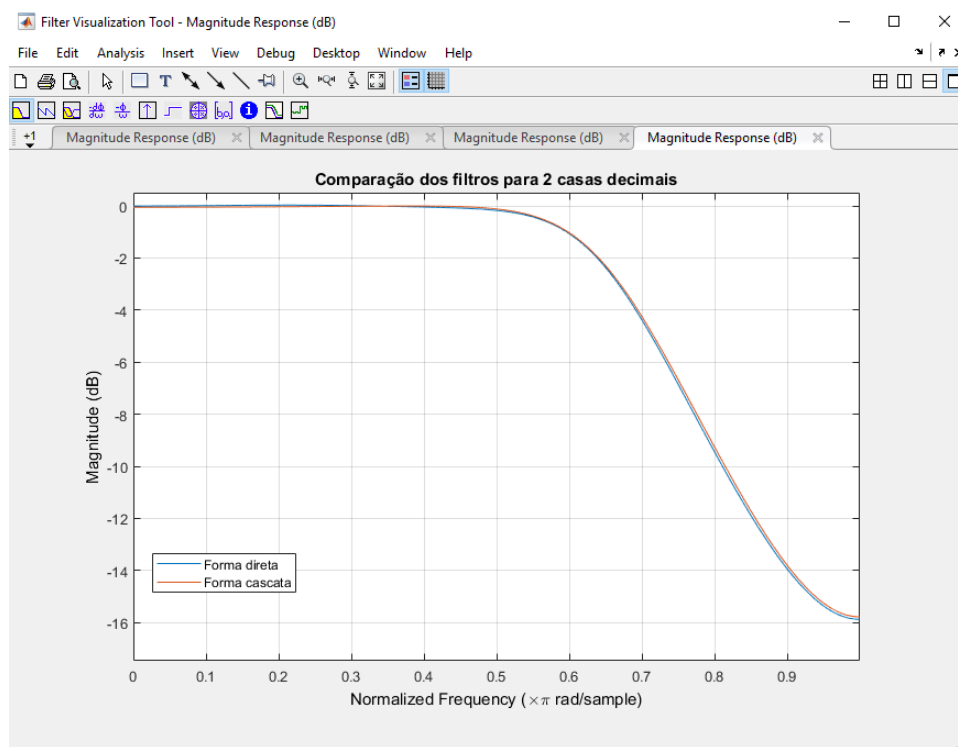


Figura 9.4. Comparação forma cascata com 2 casas decimais.

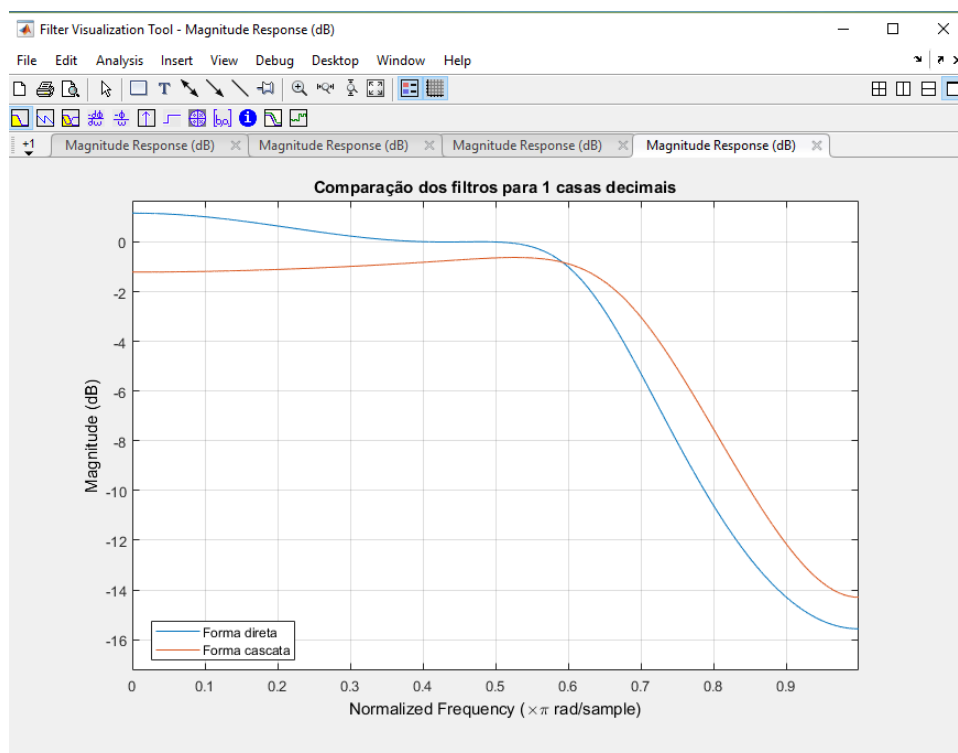


Figura 9.5. Comparação forma direta com 1 casa decimal.

Nesse trecho de código observamos o mesmo quanto a sua transformação em frequência. Nesse caso vemos que o filtro resultante é um filtro passa-alta e isso se dá pelo fato

que essa mudança na frequência gera uma distorção na função impulso onde as amostras par tem seu sinal invertido. Esse comportamento se observa na implementação abaixo, assim como o gráfico comprovando a transformação em um filtro passa-alta.

```
% -----
% TRANSFORMAÇÃO EM FREQUÊNCIA ( $z^{-1} = -z^{-1}$ ) PARA FORMA DIRETA
% -----

Bz_transf_freq_direta = Bz_direta;
Az_trasnf_freq_direta = Az_direta;

% Loop para os números par
for j=2:2:6
    %  $z^{-1} \Rightarrow -z^{-1}$ 
    Bz_transf_freq_direta(j) = -Bz_direta(j);
    Az_trasnf_freq_direta(j) = -Az_direta(j);
    hold on
end
hold off

% geração dos gráficos para transformação em frequência ( $z^{-1} = -z^{-1}$ ) - Forma Direta
figure
set(gcf,'NumberTitle','off');
set(gcf,'Name','Mudança da Transformação em frequência para forma direte:  $Z^{-1} = -z^{-1}$ ');
title('Mudança da Transformação em frequência para forma direte:  $Z^{-1} = -z^{-1}$ ')
freqz(Bz_transf_freq_direta, Az_trasnf_freq_direta);
```

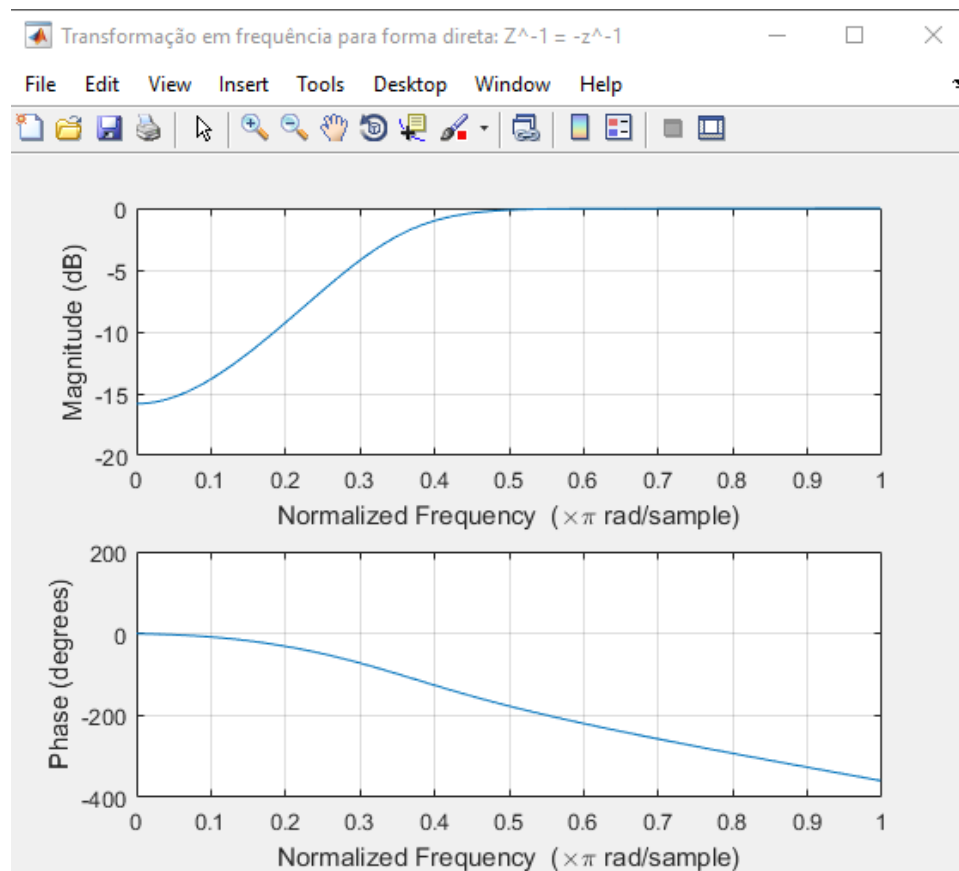


Figura 10. Gráfico da transformação em frequência $z^{-1} = -z^{-1}$

Essa transformação, de forma análoga à transformação anterior, modifica a característica do filtro, mas dessa vez tornando-o um passa-faixa. A implementação se deu através da manipulação dos coeficientes e do princípio que esse filtro é obtido através da multiplicação do filtro original passa-baixa com o filtro da transformação anterior passa-alta.

```
% -----  
% TRANSFORMAÇÃO EM FREQUÊNCIA ( $z^{-1} = z^{-2}$ ) PARA FORMA DIRETA  
% -----  
  
Bz_transf_freq_2_direta = Bz_direta;  
Az_transf_freq_2_direta = Az_direta;  
for j=2:2:6  
    Bz_transf_freq_2_direta(j) = 0;  
    Az_transf_freq_2_direta(j) = 0;  
    hold on  
end  
  
Bz_transf_freq_2_direta(3) = Bz_direta(2);  
Az_transf_freq_2_direta(3) = Az_direta(2);  
index = 5;  
isPar = false;  
for j=3:1:7  
    if isPar == true %quando ímpar  
        Bz_transf_freq_2_direta(index) = Bz_direta(j);  
        Az_transf_freq_2_direta(index) = Az_direta(j);  
    else  
        Bz_transf_freq_2_direta(index) = -Bz_direta(j);  
        Az_transf_freq_2_direta(index) = -Az_direta(j);  
    end  
    index = index + 2;  
    isPar = ~(isPar);  
    hold on  
end  
hold off  
  
% geração dos gráficos para transformação em frequência ( $z^{-1} = z^{-2}$ ) - Forma Direta  
figure  
set(gcf, 'NumberTitle', 'off');  
set(gcf, 'Name', 'Transformação em frequência para forma direta:  $Z^{-1} = -z^2$ ');  
title(sprintf('Transformação em frequência para forma direta:  $Z^{-1} = -z^2$ '))  
% correção do ganho para geração do gráfico.  
freqz(Bz_transf_freq_2_direta/2.82, Az_transf_freq_2_direta);
```

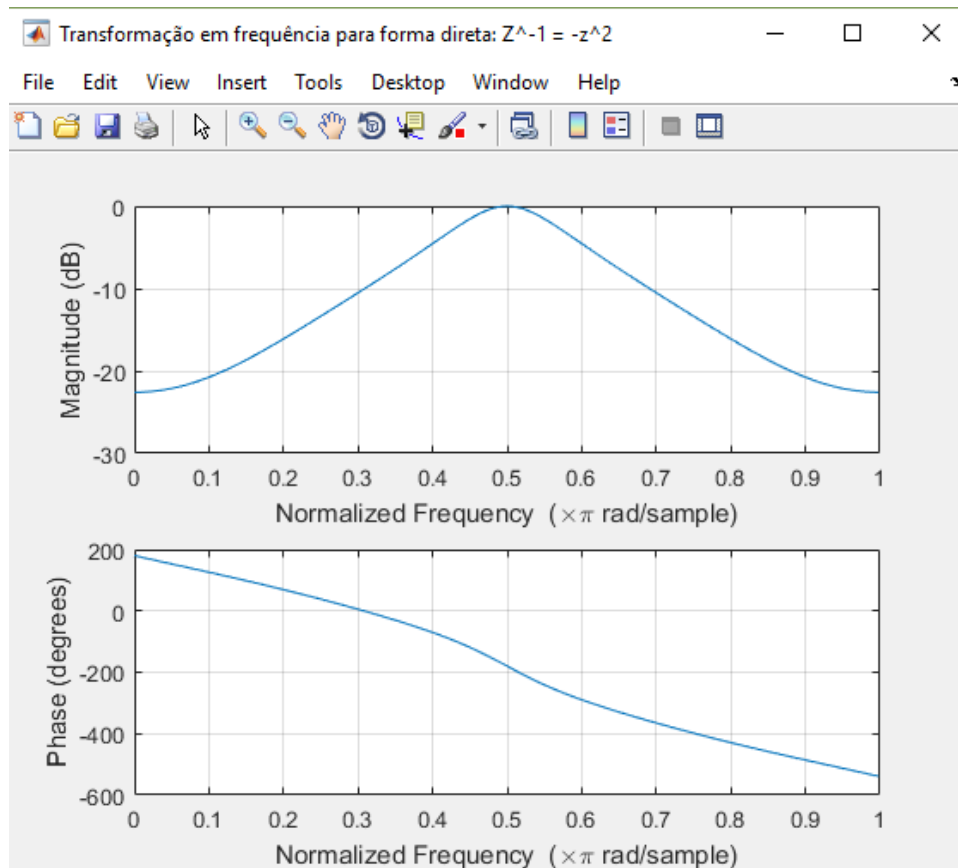


Figura 10. Gráfico da transformação em frequência $z^{-1} = z^2$

Aqui segue a implementação das funções para implementação do filtro discreto por meio da invariância ao impulso dada a forma em cascata. Uma vez extraído os polos, zeros e ganhos foram montados modelos para cada um dos polos e zeros. A função de transferência então é dada pela multiplicação desses vezes o ganho extraído no primeiro momento.

```

% -----
% FUNCTIONS
% -----
% função que projeta um filtro discreto pelo método do impulso a
% invariância pela forma em cascata
function [B, A] = impinvarByCascata(Denominador, Numerador, Td)
    % Aqui extraímos os zeros, os polos e o ganho:
    % Z: Zeros
    % P: Polos
    % K: Ganho
    [Z,P,K] = tf2zp(Denominador,Numerador);

    % Obtem n objetos modelo de classe @zpk que depois poderão ser usados para
    % montar Hz através da forma em cascata.

    SYS_1 = zpk(Z(1),P(1),1,Td);
    SYS_2 = zpk(Z(2),P(2),1,Td);
    SYS_3 = zpk(Z(3),P(3),1,Td);
    SYS_4 = zpk(Z(4),P(4),1,Td);
    SYS_5 = zpk(Z(5),P(5),1,Td);
    SYS_6 = zpk(Z(6),P(6),1,Td);

    % Hz pela forma em cascata
    Hz_cascata = SYS_1*SYS_2*SYS_3*SYS_4*SYS_5*SYS_6*K;

    % Extraindo os numeradores e denominador para montagem da função de
    % transferência
    % Bz: Numerador
    % Az: Denominador
    [Bz_cascata, Az_cascata] = tfdata (Hz_cascata,'v');
    B = Bz_cascata;
    A = Az_cascata;
end

```

Essa função só difere da anterior pelo fato de que recebe um parâmetro a mais para arredondamento, como é observado abaixo. Então seu resultado já é com base no arredondamento das casas decimais.

```
% função que projeta um filtro discreto pelo método do impulso a
% invariância pela forma em cascata limitado a um numero de casa decimais
% específicas.
function [B, A] = impinvarByCascataRounded(Denominador, Numerador, Td, Decimals)
% Aqui extraímos os zeros, os polos e o ganho:
% Z: Zeros
% P: Polos
% K: Ganho
[Z,P,K] = tf2zp(Denominador,Numerador);

% Obtem n objetos de class @zpk que depois poderão ser usados para
% montar Hz através da forma em cascata.

SYS_1 = zpk(round(Z(1), Decimals), round(P(1), Decimals),1,Td);
SYS_2 = zpk(round(Z(2), Decimals), round(P(2), Decimals),1,Td);
SYS_3 = zpk(round(Z(3), Decimals), round(P(3), Decimals),1,Td);
SYS_4 = zpk(round(Z(4), Decimals), round(P(4), Decimals),1,Td);
SYS_5 = zpk(round(Z(5), Decimals), round(P(5), Decimals),1,Td);
SYS_6 = zpk(round(Z(6), Decimals), round(P(6), Decimals),1,Td);

% Hz pela forma em cascata
Hz_cascata = SYS_1*SYS_2*SYS_3*SYS_4*SYS_5*SYS_6*K;

% Extraindo os numeradores e denominador para montagem da função de
% transferência
% Bz: Numerador
% Az: Denominador
[Bz_cascata, Az_cascata] = tfdata (Hz_cascata,'v');
B = Bz_cascata;
A = Az_cascata;
end
```

IX. CONCLUSÃO

O projeto de um filtro passa-baixa Butterworth implementado e simulado nesse projeto, através do MATLAB, teve como foco executar as etapas da filtragem de um sinal analógico em digital, etapas essas que foram abordadas em sala com o intuito do aprendizado do comportamento desses componentes, tentando-se chegar a um filtro ideal.

Foi possível analisar também o processo do filtro passa-baixas como podemos perceber no diagrama de espectro de frequência onde simulamos frequências abaixo da frequência de corte.

O uso dos filtros digitais ajuda no ganho da qualidade de informação, já que o mesmo poderá eliminar ruídos indesejados e também faixas do sinal original.

A utilização do MATLAB para o desenvolvimento desse projeto como uma programação intermediária para demonstração do comportamento e cálculos dos blocos apresentados dos filtros digitais, conseguimos obter resultados satisfatórios afim de comprovar na teoria o funcionamento dos Filtros Digitais IIR a partir do passa-baixas Butterworth.

X. REFERÊNCIAS BIBLIOGRÁFICAS

[2] DINIZ, P.S.R. Eduardo, A.B.S. SERGIO, L.N. Processamento Digital de Sinais: Projeto e Análise de Sistemas. 2. ed. São Paulo; Bookman, 2004.

[4] Elder E. C. O., Adaildo G. D'A., Ronaldo A. M. e João Bosco L. de O. PROJETO E ANÁLISE DO DESEMPENHO DOS FILTROS IIR POR MEIO DA TÉCNICA DE INVARIÂNCIA AO IMPULSO E TRANSFORMAÇÃO BILINEAR. Universidade Federal do Rio Grande do Norte, Natal – RN.

[5]http://paginapessoal.utfpr.edu.br/alessandro/disciplinas-do-semester/processamento-digital-de-sinais/aulas/terceira-prova/Projeto%20de%20filtros%20iir%20Ingle%20Proakis.pdf/at_download/file Acessado em 05/07/2017.

[6] OPPENHEIM, A. SCHAFER, V. BUCK, J.R. Discrete-Time Signal Processing. 3. ed. New Jersey; Prentice Hall, 1999.

[7] <http://www.ece.ufrgs.br/~eng04006/aulas/aula24.pdf>. Acessado em 02/09/2017.

[8] <https://www.mathworks.com/help/signal/ref/impinvar.html>. Acessado em 30/08/2017.

[9] http://www.dee.eng.ufba.br/home/simas/PDS_aula03.pdf. Acessado em 05/09/2017.