UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CALLEBE SOARES BARBOSA

RAFAEL DA COSTA BONOTTO

RAPHAEL HENRIQUE SOARES MACHADO

VICTOR EMANUEL SOARES BARBOSA

RELATÓRIOS, EXERCÍCIOS E FUNDAMENTAÇÃO TEÓRICA RELACIONADOS A DISCIPLINA DE CONTROLE DIGITAL

PATO BRANCO

2016

CONTEÚDO

1 Introdução **3**

2 Implementação da convolução **4**

2.1 Objetivos 4

2.2 Fundamentação Teórica 4

2.2.1 Integral de convolução 4

2.2.2 Soma de convolução 5

2.3 Procedimentos 5

2.3.1 Exercício 1 5

2.3.2 Exercício 2 5

2.4 Resultados e discussões 6

2.4.1 Exercício 1 6

2.4.1.1 C) 6

2.4.2 Exercício 2 8

3 Simulação de um sistema discreto com equações diferenças **15**

3.1 Objetivos 15

3.2 Fundamentação Teórica 15

3.3 Procedimentos 16

3.4 Resultados e discussões 16

4 Modulador PWM e Sistema de Condicionamento de Sinais e ADC . . . **21**

4.1 Objetivos 21

4.2 Fundamentação Teórica 21

4.3 Procedimentos 21

4.4 Resultados e discursões 21

5 Amostragem de Sinais e Análise em Frequência de Sinais Amostrados **22**

5.1 Objetivos 22

5.2 Fundamentação Teórica 22
5.3 Procedimentos 22
5.4 Resultados e discursões 23
6 Controlador PID 27
6.1 Objetivos 27
6.2 Fundamentação Teórica 27
6.3 Procedimentos 27
6.3.1 Exercício 1 27
6.3.2 Exercício 2 27
6.3.2.1 Repita o Exercício 1 para esta função de transferência comparando os resultados de simulação de ambos os casos para os mesmos ganhos. 28
6.3.2.2 Inclua saturação na ação de controle em 150% da referência e analise o comportamento do sistema de controle 28
6.3.3 Exercício 3 28
6.4 Resultados e discursões 28
6.4.1 Exercício 1 28
6.4.2 Exercício 2 30
6.4.3 Exercício 3 33
7 Controlador Repetitivo 37
7.1 Objetivos 37
7.2 Fundamentação Teórica 37
7.3 Procedimento e resolução 37
7.3.1 Circuito de condicionamento de sinais 39
7.3.2 ADC de 10 bits 39
7.3.3 Controlador repetitivo 39
7.3.4 Proporcional-Integral-Derivativo (PID) 40
7.3.5 PWM 40
7.4 Resultados e discussões 41
8 Conclusão 42

1 INTRODUÇÃO

2 IMPLEMENTAÇÃO DA CONVOLUÇÃO

2.1 OBJETIVOS

O objetivo principal desta prática é a implementação da convolução como ferramenta matemática para obtenção da saída de um sistema dada uma entrada qualquer e a resposta ao impulso.

2.2 FUNDAMENTAÇÃO TEÓRICA

A convolução, pode-se assim dizer, é o equivalente entre sinais da multiplicação. Ela pode ser descrita em tempo contínuo, sendo chamada de integral de convolução e em tempo discreto de soma de convolução.

2.2.1 Integral de convolução

A resposta $y(t)$ a uma entrada $x(t)$ aplicada a um sistema T, sendo este linear e invariante no tempo, pode ser dado pela Equação 2, onde $h(t)$ é a resposta do sistema ao impulso. Esta dedução partiu da propriedade da função impulso dada na Equação 1.

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau \tag{1}$$

$$\begin{aligned}
y(t) &= Tx(t) \\
y(t) &= T\int_{-\infty}^{\infty} x(\tau)\delta t - \tau d\tau \\
y(t) &= \int_{-\infty}^{\infty} x(\tau)T\delta t - \tau d\tau \\
y(t) &= \int_{-\infty}^{\infty} x(\tau)ht - \tau d\tau \\
y(t) &= x(t) * h(t) \qquad (2)
\end{aligned}$$

2.2.2 Soma de convolução

A resposta em tempo discreto, $y[n]$, a uma entrada $x[t]$ aplicada a um sistema T, sendo este linear e invariante no tempo, pode ser dado pela Equação 4, onde $h[n]$ é a resposta do sistema ao impulso. Esta dedução partiu da propriedade da função impulso dada na Equação 3.

$$x[n] = \sum_{-\infty}^{\infty} x[k]\delta(n-k) \tag{3}$$

$$\begin{aligned} y[n] &= Tx[n] \\ y[n] &= T\sum_{-\infty}^{\infty} x[k]\delta[n-] \\ y[n] &= \sum_{-\infty}^{\infty} x[k]T\delta[n-k] \\ y[n] &= \sum_{-\infty}^{\infty} x[k]h[n-k] \\ y[n] &= x[n] * h[n] \end{aligned} \tag{4}$$

2.3 PROCEDIMENTOS

Foram resolvidos os exercícios 1 2 da apresentação de slides referente a transformada Z, com código implementado em Matlab.

2.3.1 Exercício 1

Determine a saída do sistema com resposta ao impulso h[n] e para um sinal de entrada x[n], ambos sinais estão mostrados na Figura 1:
A) análise gráfica por impulsos
B) cálculo/tabela de convolução
C) convolução utilizando ferramenta computacional: *script* Matlab

2.3.2 Exercício 2

Considere um sistema que possui resposta ao impulso h[n]=2^{-nT} e o sinal de entrada é uma onda retangular (razão cíclica 40%, D=0,4) com período 10s e amplitude

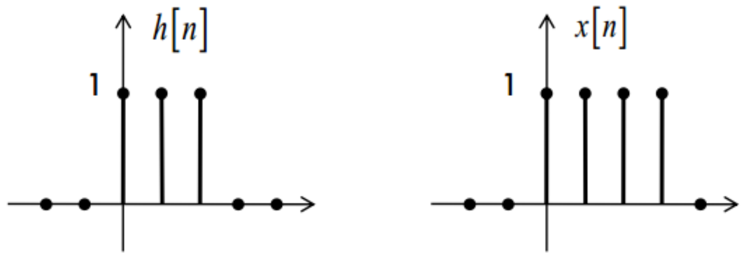

Figura 1: Sinais do Exercício 1, prática 1.

3,3V.

A) Determine a resposta (sinal de saída) do sistema para 3 períodos do sinal de entrada considerando que o periodo de amostragem é T=0.2s.

B) Considere um ruído de 10% no sinal de entrada e repita o item A.

C) Considere um sinal de entrada senoidal com mesmo período, amplitude e ruído dados acima.

2.4 RESULTADOS E DISCUSSÕES

2.4.1 Exercício 1

2.4.1.1 C)

O código implementado em Matlab para a resolução da letra C do exercício 1 está logo abaixo:

```
%% ----------------------------
%   Universidade Tecnologica Federal do Parana
%   Engenharia Eletrica
%   Controle Digital
%
%   Aluno: Victor Emanuel Soares Barbosa
%
%   Aula 3: Transformada Z
%   Exercicio 1:
%   Determine a saida do sistema com
%   resposta ao impulso h[n] e para um sinal de entrada
%   x[n]:
%  C) convolucao utilizando ferramenta
%   computacional: script Matlab
% ----------------------------
%% Inicializacao do programa
```

```
clc;
clear all;
close all;

%% Variaveis gerais
numero_pontos = 8; % Numero de pontos simulados
h = [1 1 1 zeros(1,numero_pontos-3)]; % resposta ao impulso
x = [1 1 1 1 zeros(1,numero_pontos-4)]; % sinal de entrada
y = [zeros(1,numero_pontos)]; % resposta do sistema a entrada x
amostras = zeros(1,numero_pontos); % vator de amostras

%% Letra c)

% Execucao
for n=0:(numero_pontos-1)
    for k = 0:3
        if (n-k)>0
            y(n+1) = y(n+1) + x(k+1)*h(n-k);
        end
    end
    amostras(n+1) = n-1;
end

% Graficos
figure
stem(amostras,y)
title('Aula 2 - Exercicio 2 - Letra c');
legend('Sinal y[n]');
ylabel('Amplitude');
xlabel('n');
```

Obtendo como saída o gráfico da Figura 2

Observa-se que os intervalos de simulação para uma convolução teórica deveria ser estabelecido entre $-\infty$ e $+\infty$, com uma valor de passo unitário. Porém computacionalmente esses intervalos são impraticáveis, necessitando de uma adequação nos valores de simulação.

Como os sinais h e x possuem valores diferentes de zero apenas a partir de $n = 0$ até $n = 3$ se faz necessário convolucionar apenas neste intervalo. O numero total de pontos convolucionados é dado pela soma dos valores no intervalo da convolução dos

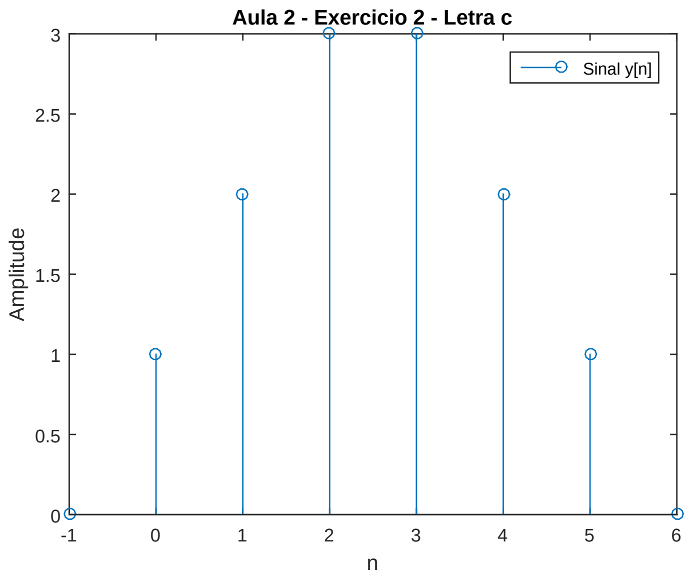

Figura 2: Gráfico de saída do código do Exercício 1, letra C, prática 1.

dois sinais, sendo portanto no total 8 pontos.

A figura 2 apresenta o resultado da simulação de forma coerente com a teoria, e portanto comprova a eficacia do algoritmo desenvolvido.

2.4.2 Exercício 2

O código implementado em Matlab para a resolução do exercício 2 está logo abaixo:

```
%% ----------------------------------------
%   Universidade Tecnologica Federal do Parana
%   Engenharia Eletrica
%   Controle Digital
%
%   Aluno: Victor Emanuel Soares Barbosa
%
%   Aula 3: Transformada Z
%   Exercicio 2:
%   Considere um sistema que possui resposta
%   ao impulso h[n]=2-nT e o sinal de entrada he uma onda
%   retangular (razao ciclica 40%, D=0,4) com periodo
%   10s e amplitude 3,3V.
%   A) Determine a resposta (sinal de saida) do sistema para
%   3 periodos do sinal de entrada considerando que o
```

```
%   periodo de amostragem he T=0.2s.
%   B) Considere um ruido de 10% no sinal de entrada e
%   repita o item A.
%   C) Considere um sinal de entrada senoidal com mesmo
%   periodo, amplitude e ruido dados acima.
% ---------------------------------------------------
%% Inicializacao do programa
clc;
clear all;
close all;

%% Variaveis gerais
periodos = 3; % Quantidade de periodos
R = 0.1; % Nivel de ruido
T = 0.2; % Periodo de amostragem
T_entrada = 10; % Periodo do sinal de entrada
A_entrada = 3.3; % Amplitude do sinal de entrada
D = 0.4; % Razao ciclica do sinal de entrada
total_pontos = periodos*T_entrada/T; % Total de pontos simulados
pontos_periodo = T_entrada/T; % Total de pontos por periodo
amostras = zeros(1,total_pontos); % Vetor de pontos de simulacao
h = zeros(1,total_pontos); % vetor da resposta ao impulso
x = zeros(1,total_pontos); % Vetor do sinal de entrada
y = zeros(1,total_pontos); % Vetor do sinal de saida
cont  = 0;

%% Letra a)

% Execucao
for n = 0:total_pontos-1

    if cont < (pontos_periodo*D)
         x(n+1) = A_entrada;
    else
        x(n+1) = 0;
    end

    if cont == pontos_periodo
        cont = 0;
    end

```

```
    cont = cont + 1;

    for k = 0:pontos_periodo*3
        h(n+1) = 2^(-n*T);

        if (n-k)> 0
            y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
        end
    end

    amostras(n+1) = (n)*T;
end

% Graficos
figure
stem(amostras,y)
hold
stem(amostras,x)
stem(amostras,h)
title('Aula 2 - Exercicio 2 - Letra a');
legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
ylabel('Amplitude');
xlabel('Tempo (s)');

%% Letra b)

% Execucao
for n = 0:total_pontos-1

    if cont < (pontos_periodo*D)
         x(n+1) = A_entrada*(1+0.1*rand);
    else
        x(n+1) = 0.1*rand;
    end

    if cont == pontos_periodo
        cont = 0;
    end

    cont = cont + 1;

```

```
    for k = 0:pontos_periodo*3
        h(n+1) = 2^(-n*T);

        if (n-k)> 0
            y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
        end
    end

    amostras(n+1) = (n)*T;
end

% Graficos
figure
stem(amostras,y)
hold
stem(amostras,x)
stem(amostras,h)
title('Aula 2 - Exercicio 2 - Letra b');
legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
ylabel('Amplitude');
xlabel('Tempo (s)');

%% Letra c)

% Execucao
for n = 0:total_pontos-1

    x(n+1) = A_entrada*sin(2*pi*n/(T_entrada/T))*(1+0.1*rand);

    if cont == pontos_periodo
        cont = 0;
    end

    cont = cont + 1;

    for k = 0:pontos_periodo*3
        h(n+1) = 2^(-n*T);

        if (n-k)> 0
            y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
        end
```

```
        end

        amostras(n+1) = (n)*T;
end

% Graficos
figure
stem(amostras,y)
hold
stem(amostras,x)
stem(amostras,h)
title('Aula 2 - Exercicio 2 - Letra c');
legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
ylabel('Amplitude');
xlabel('Tempo (s)');
```

Obtendo como saída os gráfico das Figuras 3, 4 e 5.

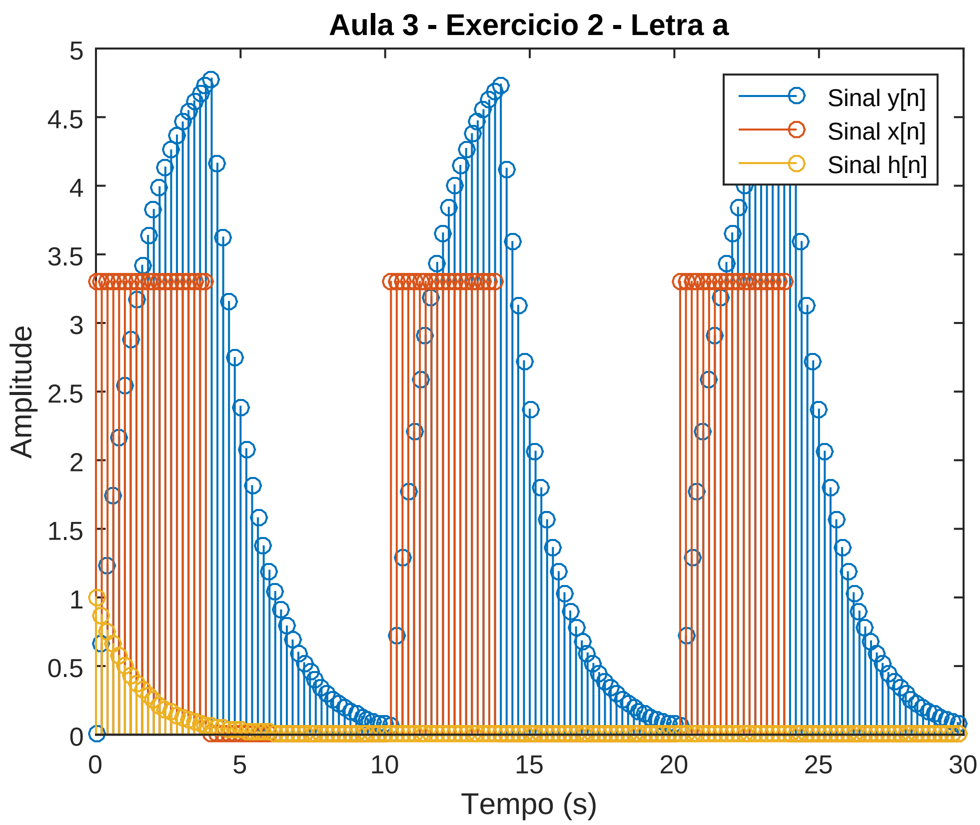

Figura 3: Gráfico de saída do código do Exercício 2, letra A, prática 1.

Neste exercício deve se ter umas atenção especial em relação ao passo da convolução, ou seja a quantidade de pontos simulados entre valores inteiros.

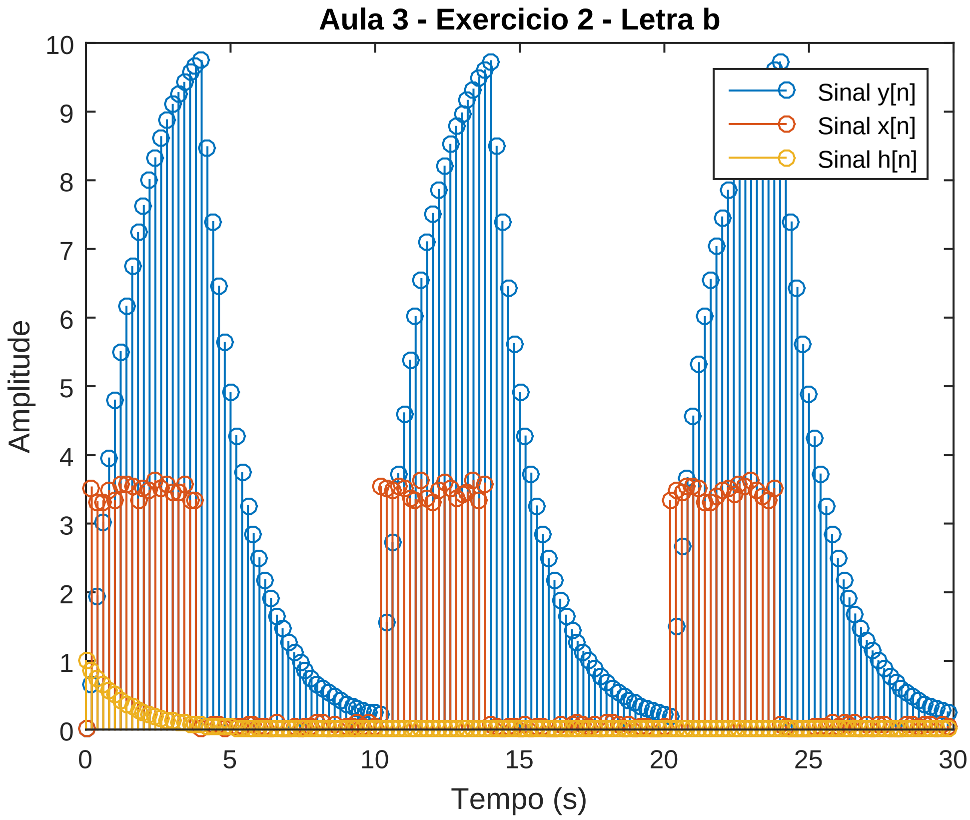

Figura 4: Gráfico de saída do código do Exercício 2, letra B, prática 1.

Como temos um período de amostragem de $0,2$ logo temos 5 pontos simulados entre valores inteiros. Assim se fixarmos o numero de períodos e variarmos o número de amostragem variamos também o número de pontos a serem convolucionados. Como a convolução pode ser representada pela soma na forma da equação 4, logo a amplitude de cada ponto n do sinal convolucionado depende do numero de amostras realizado. Para que se possa ter um sinal de amplitude normalizada independente do período de amostragem, o sinal convolucionado é dividido pelo numero de amostras entre intervalo entre dois inteiros, ou numero de amostras por ciclo.

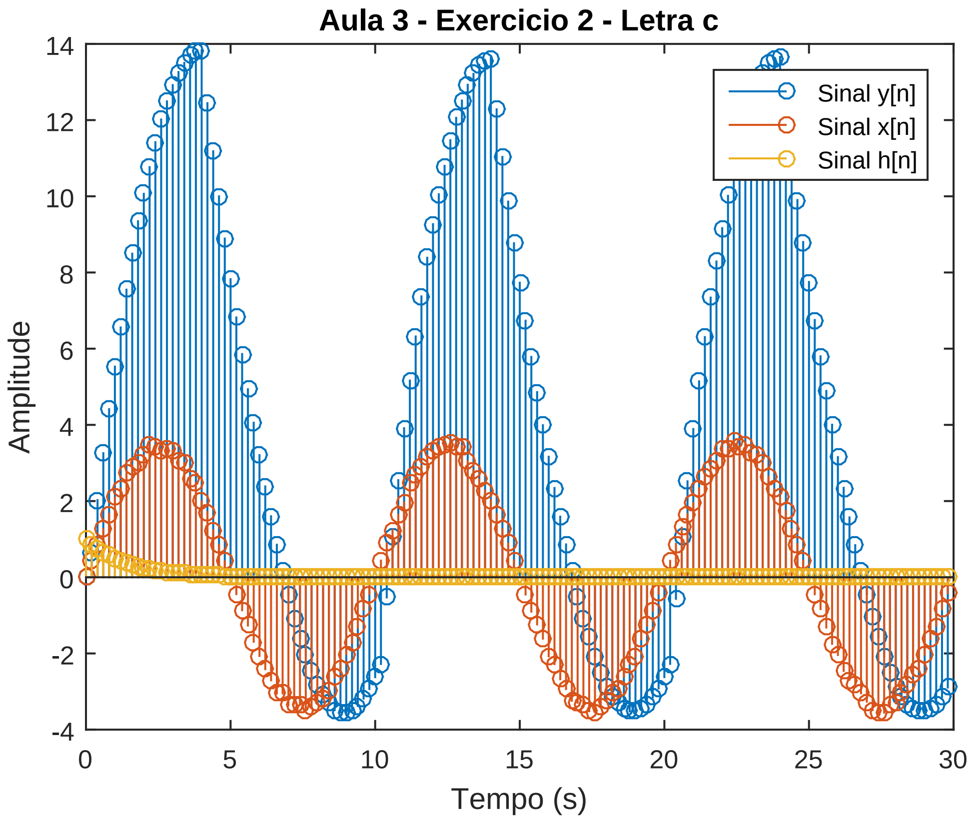

Figura 5: Gráfico de saída do código do Exercício 2, letra C, prática 1.

3 SIMULAÇÃO DE UM SISTEMA DISCRETO COM EQUAÇÕES DIFERENÇAS

3.1 OBJETIVOS

O objetivo principal desta prática é a simulação em Matlab de um sistema de controle digital completo dado as equações em Z que descrevem os blocos constituintes do sistema em malha fechada.

3.2 FUNDAMENTAÇÃO TEÓRICA

Para realização desta prática se utilizou a teoria da transformada Z inversa, em específico o método das equações de diferenças. Este método é facilmente utilizado e computadores digitais por fornecer a equação em tempo discreto da transforma inversa de z .

Quando obtemos a transformada inversa de z, assumimos que a sequência $x(kT)$ ou $x(k)$ é zero para $k < 0$. Nota-se que em aplicação de engenharia de controle e processamento de sinais, $X(z)$ é frequentemente expressado com a razão polinomial de z^{-1}, como apresentado na equação (1)

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z^{-(n-m)} + b_1 z^{-(n-m+1)} + ... + b_m z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} \quad (m \leq n) \tag{1}$$

Pelo método aproximado de equações de diferenças convertemos a equação (1) para a equação (2),

$$(1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n})Y(z) = (b_0 z^{-(n-m)} + b_1 z^{-(n-m+1)} + ... + b_m z^{-n})X(z)$$

$$\begin{aligned} y(k) + a_1 y(k-1) + a_2 y(k-2) + ... + a_n y(k-n) &= b_0 x(k-(n-m)) + ... \\ &...b_1 x(k-(n-m+1)) + ... + b_m x(k-n) \\ y(k) = -a_1 y(k-1) - a_2 y(k-2) + ... + -a_n y(k-n) &+ b_0 x(k-(n-m)) + ... \\ &...b_1 x(k-(n-m+1)) + ... + b_m x(k-n) \end{aligned} \tag{2}$$

Achando a transformada inversa z de $Y(z)$, resolve-se a equação de diferença

$y(k)$ facilmente por linguagem de programação.

3.3 PROCEDIMENTOS

O exercício consistiu em simular um sistema de tempo discreto para período de amostragem de 0,1 s para sinal de entrada uma onda quadrada de amplitude 0 - 5 V, com período de 10s e 2% de ruído randômico. O sistema está mostrado na Figura 6, bem como as equações dos blocos estão descritas abaixo:

$$C(z) = 0,9 * \frac{z - 0,8}{z - 1} \tag{3}$$

$$G(z) = \frac{0,3z}{(z - 0,5)(z - 0,2)} \tag{4}$$

$$S(z) = \frac{0,2}{z - 0,8} \tag{5}$$

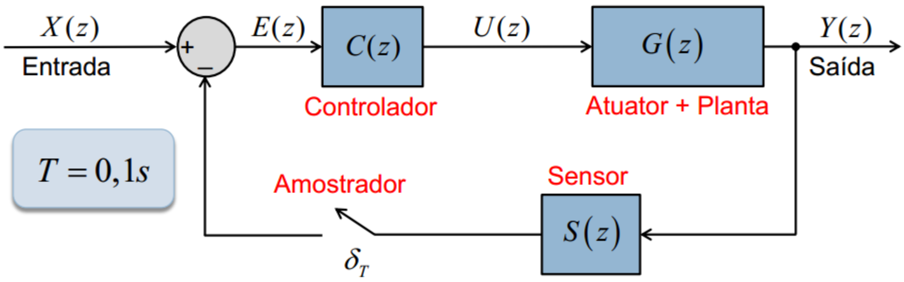

Figura 6: Diagrama de blocos da prática 2.

3.4 RESULTADOS E DISCUSSÕES

Realizando o equacionamento genérico a malha fechada do sistema apresentado na Figura 6 encontra-se a equação (6)

$$\frac{Y(z)}{X(z)} = \frac{C(z)G(z)}{1 + C(z)G(z)S(z)} \tag{6}$$

Substituindo as equações (3), (4) e (5) na equação (6) e expressado com a razão polinomial de z^{-1} temos

$$\frac{Y(z)}{X(z)} = \frac{0,27z^{-1} - 0,216z^{-2}}{1 - 1,7z^{-1} + 0,854z^{-2} - 0,1z^{-3}} \tag{7}$$

Através da equação (7) aplica-se o método da equação de diferença conforme apresentado na equação (2), concebendo a equação (8) que será simulada com o respectivo sinal de entrada.

$$y(k) = 1,7y(k-1) - 0,854y(k-2) + 0,1y(k-3) + 0,27x(k-1) - 0,216x(k-2) \quad (8)$$

O código implementado em Matlab está mostrado abaixo:

```
%% ————————————————————
%   Universidade Tecnologica Federal do Parana
%   Engenharia Eletrica
%   Controle Digital
%
%   Aluno: Victor Emanuel Soares Barbosa
%
%   Aula 5: Transformada Z Inversa
%   Exercicio 2:
%   Considere o diagrama de controle em tempo discreto.
%   Determinar o grafico da saida considerando:
% ————————————————————
%% Inicializacao do programa
clc;
clear all;
close all;

%% Variaveis gerais
ciclos = 4; % Quantidade de ciclos da onda de entrada
T_entrada = 10; % Periodo da onda de entrada
Ta = 0.1; % Periodo de amostragem
total_pontos = T_entrada/Ta*ciclos; % Total de pontos de simulacao
A_entrada = 5; % Amplitude de entrada
x = zeros(1,total_pontos); % Vetor da entrada
y = zeros(1,total_pontos); % Vetor de saida
tempo = zeros(1,total_pontos); % Vetor de tempo
cont = 0; % contador para auxiliar

for n = 1:total_pontos
    if cont < ((T_entrada/Ta)/2)
        x(n) = A_entrada*(1+0.02*rand);
```

```
    else
        x(n) = 0.02*rand*A_entrada;
    end

    if n > 3
        y(n) = 0.27*x(n-1) - 0.216*x(n-2) + 1.7*y(n-1) - 0.854*y(n-2) ...
            + 0.1*y(n-3);
    end

    tempo(n) = n*Ta;
    cont = cont + 1;

    if cont > T_entrada/Ta
        cont = 0;
    end
end

% Grafico
figure
stem(tempo,y)
hold
stem(tempo,x)
title('Aula 5 - Exercicio 2');
legend('Saida y[n]','Entrada x[n]');
ylabel('Amplitude');
xlabel('Amostras');
```

Obtendo como saída o gráfico da Figura 7.

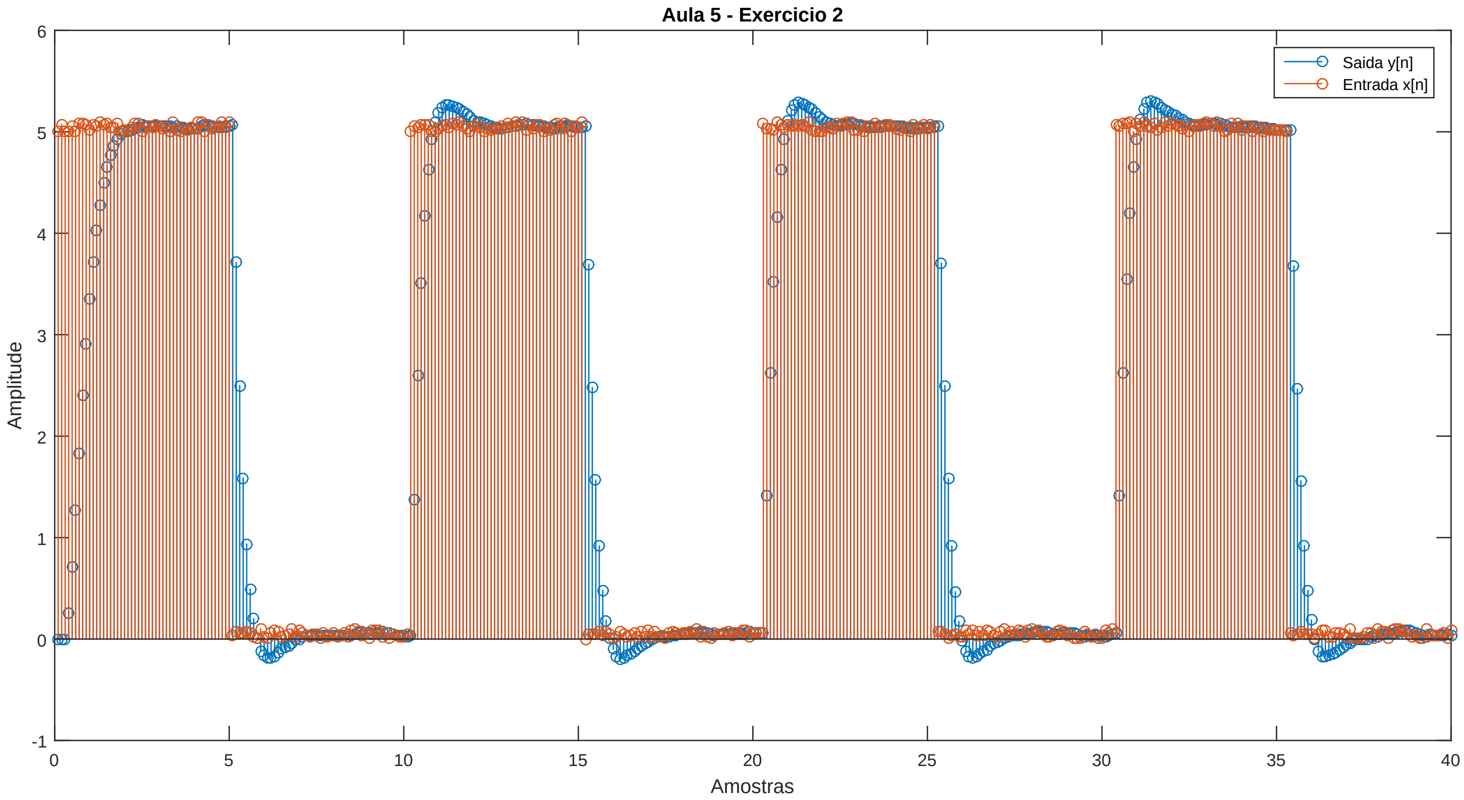

Figura 7: Gráfico de saída do código do exercício da prática 2.

A transformada z é uma ferramenta comumente utilizada para análise e síntese de sistemas de controle em tempo discreto. Análoga a transformada de Laplace, possui a equação de diferenças como ferramenta matemática para a obtenção da resposta do sistema a uma dada entrada.

Como a natureza exprime sistemas variáveis no tempo de forma contínua, a transformada inversa z permitiu através do método de equações de diferenças fornece a saída em tempo discreto fazendo uso do tempo de amostragem que simule um sistema pseudo-contínuo.

Este método recursivo possibilitou a implementação digital, conforme o exercício apresentado, de forma simplista utilizando laço de controle e vetores. Na Figura 7, dado a entrada de uma onda quadrada, o PID, a planta e o sensor interagem na malha fornecendo uma resposta com baixo sobresinal e rápido tempo de assentamento.

4 MODULADOR PWM E SISTEMA DE CONDICIONAMENTO DE SINAIS E ADC

4.1 OBJETIVOS

4.2 FUNDAMENTAÇÃO TEÓRICA

4.3 PROCEDIMENTOS

4.4 RESULTADOS E DISCURSÕES

5 AMOSTRAGEM DE SINAIS E ANÁLISE EM FREQUÊNCIA DE SINAIS AMOSTRADOS

5.1 OBJETIVOS

Nesta prática objetiva-se apresentar os princípios elementares para amostragem de sinais

5.2 FUNDAMENTAÇÃO TEÓRICA

Sendo o sinal analógico contínuo no tempo e amplitude, contém uma infinidade de valores. O aparelhos de processamento digital possuem uma banda passante limitada, acarretando na finitude de amostras utilizada para formação do sinal.

A amostragem de um sinal analógico é caracterizada como o processo pelo qual o mesmo é representado por um conjunto discreto de números. Este número, ou amostras, são iguais ao valor no sinal neste respectivo instante. A quantidade de amostras é definida pela frequência de amostragem (frequência de coleta dos valores) comparado a frequência fundamental do sinal.

A escolha da frequência de amostragem é baseado no *Teorema de amostragem de Nyquist e Shannon*. Este teorema basicamente estabelece que o sinal é precisamente reconstruído por amostras sob a condição que componente de maior frequência não deve ser maior que metade da taxa de amostragem.

Quando este teorema não é obedecido, ocorre *folding* e/ou *aliasing*, fenômeno de sobreposição do espectro de frequência conhecido. Um filtro anti-*aliasing* analógico é frequentemente colocado entre o sensor e o conversor A/D tendo como função, a redução das componentes de ruídos em alta frequência

5.3 PROCEDIMENTOS

O exercício consistiu em utilizar o *Simulink*® para a construção do sinal contínuo no tempos com as seguintes componentes frequências:

- $f_0 = 1$ Hz;

- $f_1 = 5$ Hz;
- $f_2 = 100$ Hz;

Sendo estes sinais amostrados por um filtro *Zero-order-hold* (ZOH) com as seguintes frequências de amostragem (f_s):

- $f_{s1} = 10$ Hz;
- $f_{s2} = 100$ Hz;
- $f_{s3} = 1000$ Hz;

Na análise do espectro foi utilizado o comando FFT do $Matlab^{®}$, de forma a verificar o fenômeno de *aliasing* para cada caso da frequência de amostragem. Para as frequências onde ocorreram foi projeto um filtro passa-baixa.

5.4 RESULTADOS E DISCURSÕES

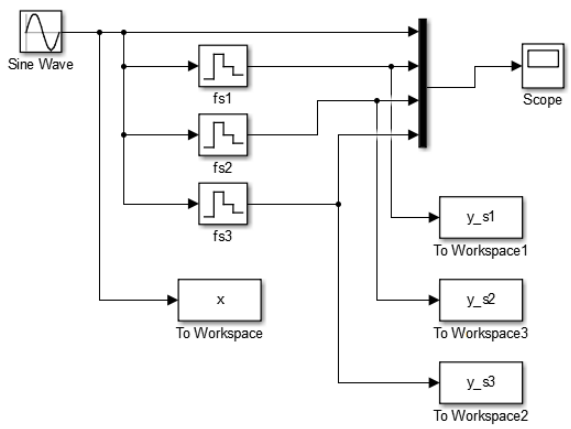

Figura 8: Diagrama de blocos da prática 4.

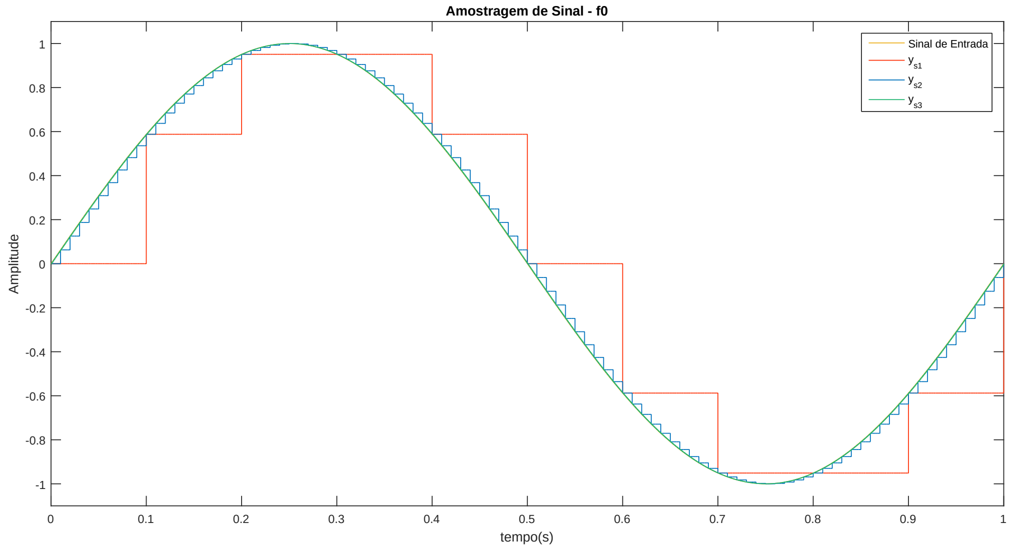

Figura 9: Diagrama de blocos da prática 4.

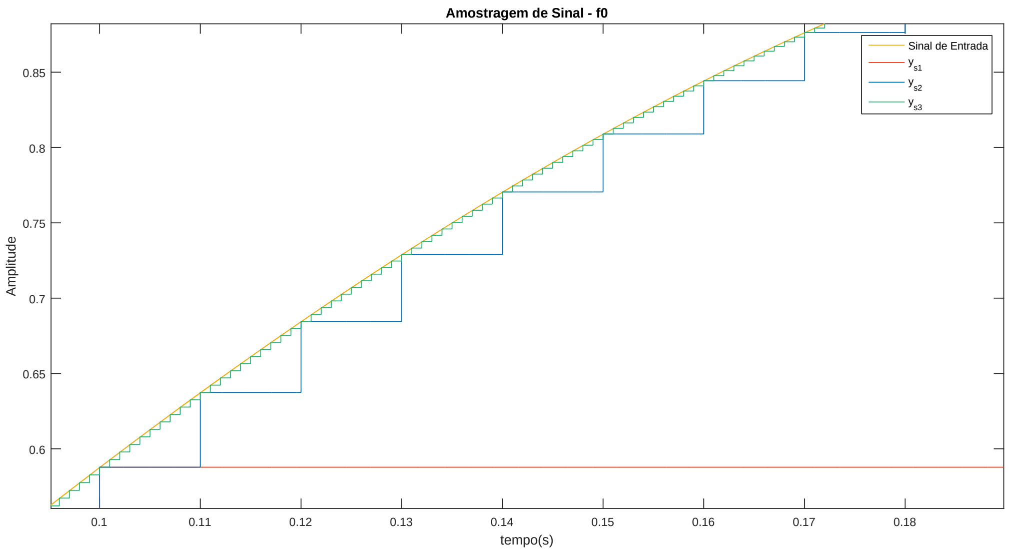

Figura 10: Diagrama de blocos da prática 4.

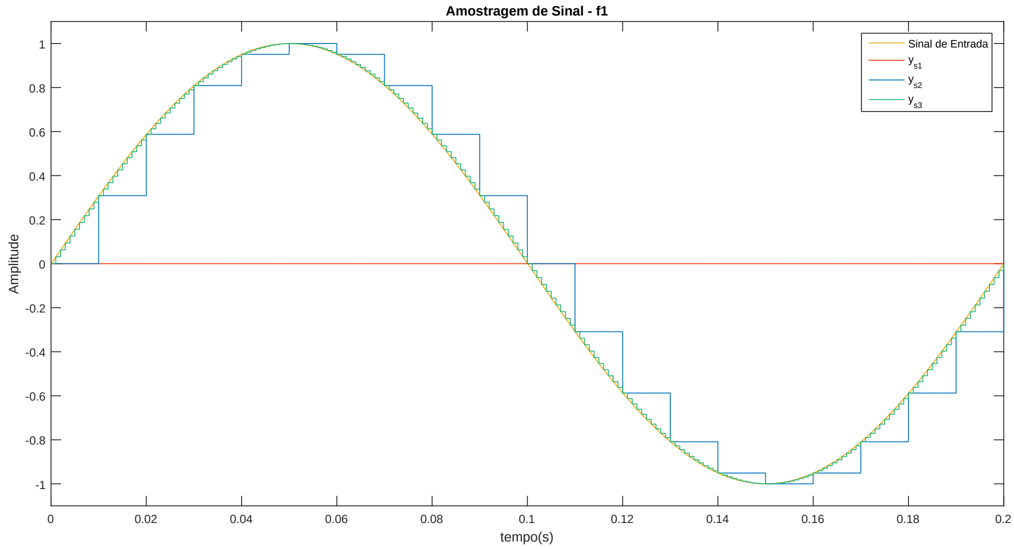

Figura 11: Diagrama de blocos da prática 4.

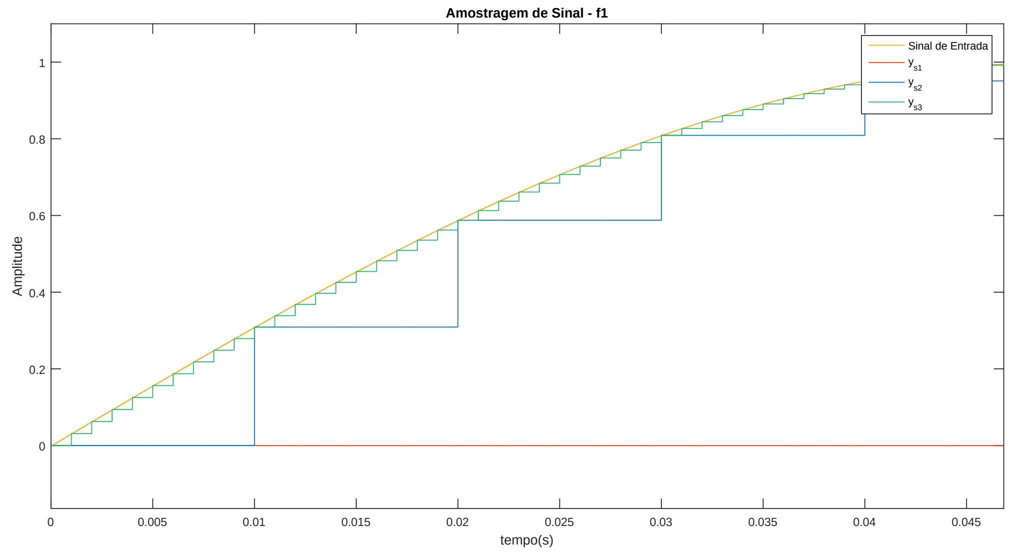

Figura 12: Diagrama de blocos da prática 4.

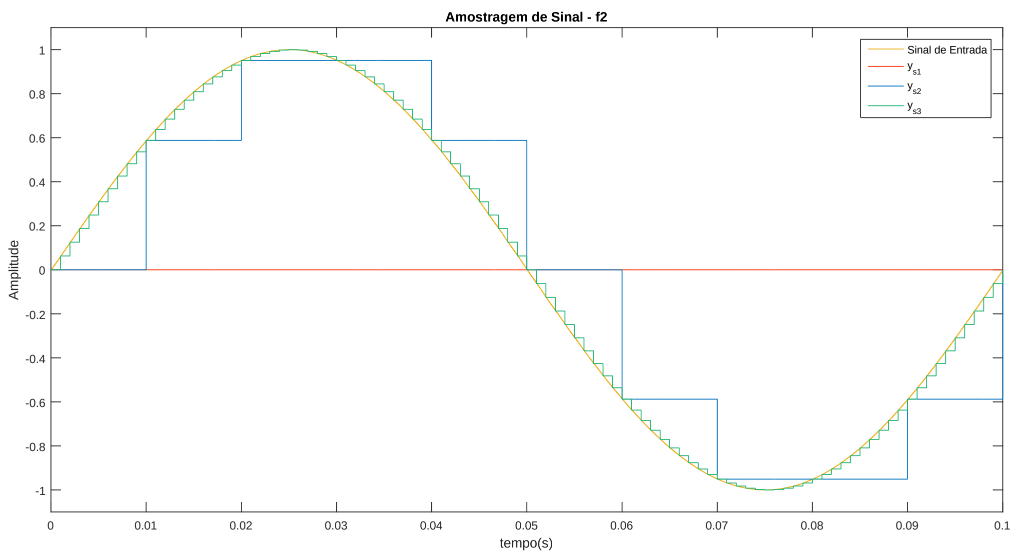

Figura 13: Diagrama de blocos da prática 4.

6 CONTROLADOR PID

6.1 OBJETIVOS

O objetivo principal desta prática é tanto a dedução da equação em tempo discreto de um PID, como a sua implementação prática em ambiente Simulink®e em código básico de Matlab®.

6.2 FUNDAMENTAÇÃO TEÓRICA

6.3 PROCEDIMENTOS

6.3.1 Exercício 1

Considere um sistema em malha fechada com PID T(z) sendo:

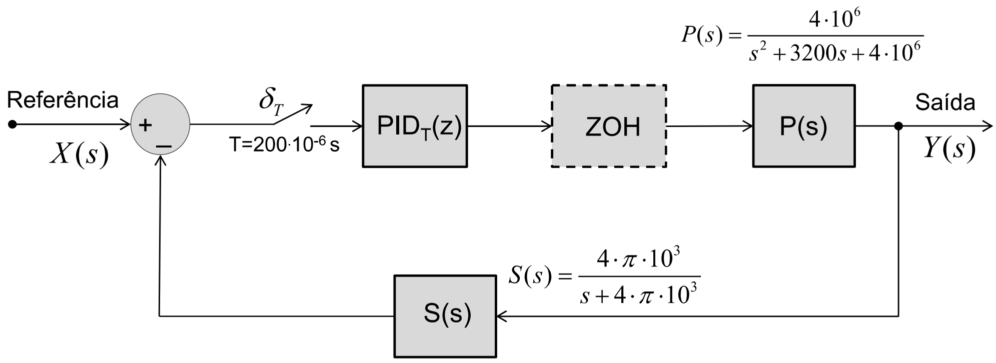

Figura 14: Sistema - Exercício 1

Utilize simulações computacionais (Simulink®) para projetar ganhos para o controlador PID considerando a entrada um degrau unitário.

6.3.2 Exercício 2

Obtenha a função de transferência do PID de tempo discreto utilizando o método de discretização Forward para a parcela integral e considere Kp, Ki, e Kd como ganhos paralelos do controlador PID.

6.3.2.1 Repita o Exercício 1 para esta função de transferência comparando os resultados de simulação de ambos os casos para os mesmos ganhos.

6.3.2.2 Inclua saturação na ação de controle em 150% da referência e analise o comportamento do sistema de controle

6.3.3 Exercício 3

Considerando o sistema descrito no exercício 2, desenvolva um script em Matlab para implementar o PID com os seguintes parâmetros.

- Sinal de referência: Onda quadrada Amplitude 40 V_{pp} Offset 0 V Período 10ms
- Controlador: PID 'Digital' (equação de diferenças) Saturação do PID (Sat = 0.98 V_{cc})
- Atuador: sinal PWM Resolução 8 bits (2n divisões) $V_{cc} = 40\ V$
- Ruído: Randômico Amplitude 2% da saída
- Conversor A/D: Resolução 10 bits (2n níveis) Vin=0-5V T=200.10-6 s
- Planta $P(s) = \frac{4 \cdot 10^6}{s^2+3200s+4 \cdot 10^6}$
- Sensor $S(s) = \frac{4 \cdot \pi \cdot 10^3}{s+4 \cdot \pi \cdot 10^3}$

6.4 RESULTADOS E DISCURSÕES

6.4.1 Exercício 1

A partir do diagrama da Figura 14, monta-se o circuito no Simulink®. A Figura 15 apresenta o dado circuito implementado.

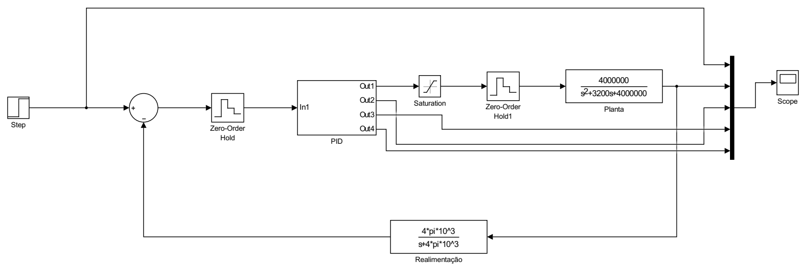

Figura 15: Diagrama de Blocos Exercício 1

A Figura 16 está apresentado o circuito PID implementado. Os valores K_P, K_D e K_I são obtidos a partir dos método de projeto Ziegler-Nichols usando o procedimento 2. Ou seja foi obtido um ganho critico em malha fechada de modo que provoque um sinal de saída com oscilações constantes, e com base neste ganho e no período das oscilações constantes se defini os valores dos ganhos do PID.

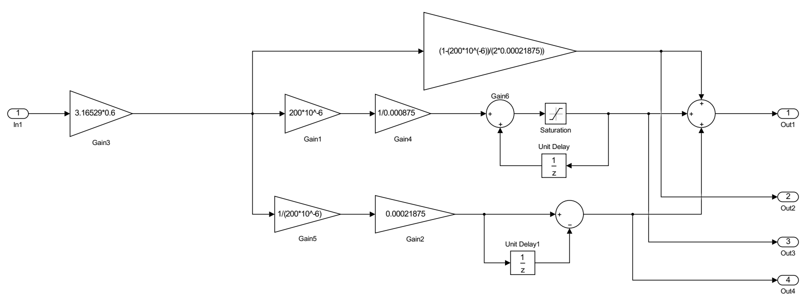

Figura 16: Diagrama de Blocos Exercício 1 - PID

A Figura 17 apresenta o resultado do circuito deste exercício simulado. As curvas contidas na Figura 17 são referentes aos 5 sinais na entrada do bloco *scope*, sendo: curva em amarelo representa o sinal de referência, curva em roxo representa a resposta do sistema em malha fechada com PID, curva em azul a ação proporcional do PID, curva em vermelho a ação integrativa do PID e por fim a curva verde representa ação derivativa do PID.

Figura 17: Resultado da Simulação do Exercício 1 - PID

6.4.2 Exercício 2

Para a obtenção da função transferência do PID de tempo discreto usando o método de discretização *Forward* se parte da equação básica do PID, definida pela Equação 1.

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{d}{dt}e(t)\right) \tag{1}$$

Seguindo a discretização, tornando $u_k = u(k)$ é possível aproximar a parcela de erro pela Equação 2, a parcela integral pela Equação 3, proveniente da discretização

Forward. E por fim a parcela derivativa pela Equação 4.

$$e(t) \rightarrow e_k \tag{2}$$

$$\int_0^t e(t)dt \rightarrow \sum_{i=0}^{k} e_i T \tag{3}$$

$$\frac{d}{dt}e(t) \rightarrow \frac{e_k - e_{k-1}}{T} \tag{4}$$

Logo pode-se definir a forma discreta do PID pela Equação 5.

$$u_k = K\left(e_k + \frac{T}{T_i}\sum_{i=0}^{k}(e_i) + \frac{T_d}{T}(e_k - e_{k-1})\right) \tag{5}$$

Para que se possa eliminar o somatório da integral, que torna a implementação prática inaplicável, pode-se utilizar a transformada Z. Logo pelas Equações 7, 9 e 12.

$$Z\{e_k\} = E(z) \tag{6}$$

$$Z\{K \cdot e_k\} = K \cdot E(z) \tag{7}$$

$$Z\{e_{k-1}\} = z^{-1}E(z) \tag{8}$$

$$Z\{K_D(e_k - e_{k-1})\} = K_D(1 - z^{-1})E(z) \tag{9}$$

$$Z\left\{\sum_{i=0}^{k} f_i\right\} = \frac{1}{1 - z^{-1}} Z f_i \tag{10}$$

$$Z\left\{K_I \sum_{i=0}^{k} e_i\right\} = K_I \frac{1}{1 - z^{-1}} E(z) \tag{11}$$

$$\tag{12}$$

Logo temos a transformada Z para o PID se torna a Equação 13.

$$U(z) = K\left(E(z) + \frac{T}{T_i}\frac{1}{1-z^{-1}}E(z)\frac{T_d}{T}(1-z^{-1})E(z)\right) \tag{13}$$

Por fim tem-se pela propriedade do deslocamento a Equação 17, que defini a equação discretizada do PID.

$$u_P(k) = K_P\ e(k) \tag{14}$$

$$u_I(k) = K_I\ e(k) + u_I\ (k-1) \tag{15}$$

$$u_D(k) = K_D\ (e(k) - e(K-1)) \tag{16}$$

$$u_{PID} = u_P(k) + u_I(k) + u_D(k) \tag{17}$$

Logo para se repetir o mesmo procedimento feito no exercício 2 aplicando a função do PID discretizada utilizando o método de *Forward* para parcela integral basta apenas modificar o valor do ganho K_I, já que ele é o único fator modificado pela mudança no método de discretização da integral. Portanto a figura 18 apresenta o resultado da simulação do exercício 2, sendo que as curvas contidas nesta figura tem a mesma representação daquelas contidas na Figura 17.

Figura 18: Resultado da Simulação do Exercício 2 - PID

Os valores de K, T_i e T_d continuam sendo os mesmo daqueles usados no Exercício 1, sendo que o que muda neste exercício é a equação que defini a parcela K_i.

6.4.3 Exercício 3

Para a resolução do Exercício 3 é implementado em Matlab o código abaixo:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                AULA 12P             %

% Exerccio 1

%Considere um sistema em malha fechada com PID
%T(z) sendo: Utilize  simulaes computacionais (Simulink®) para ...
    projetar ganhos
%para o controlador PID considerando a entrada um degrau  unitrio.

% Exerccio 2
```

```

%Obtenha a  funo de  transferncia do PID de tempo
%discreto utilizando o mtodo de  discretizao Forward
%para a parcela integral e considere Kp, Ki, e Kd como
%ganhos paralelos do controlador PID.

% Exerccio 3

%Considerando o sistema descrito no  exerccio 2, desenvolva um script
%em Matlab para implementar o PID com os seguintes  parmetros:

clc;
clear;
close all;

PeriodoOndaQuadrada = 0.01;
AmplitudeOndaQuadrada = 40;
T = 0.0002;
tfinal = 10;
NumeroDePeriodosDeQuadrada = tfinal/PeriodoOndaQuadrada;
TempoDePeriodoDeQuadrada = tfinal/T/NumeroDePeriodosDeQuadrada;

U = zeros(tfinal/T,1);
PID = zeros(tfinal/T,1);
Planta = zeros(tfinal/T,1);
Erro = zeros(tfinal/T,1);
Realimentacao = zeros(tfinal/T,1);
Up = zeros(tfinal/T,1);
Ui = zeros(tfinal/T,1);
Ud = zeros(tfinal/T,1);

s=tf('s');
G = 4*10^6/(s^2 + 3200*s + 4*10^6);
S = 4*pi*10^3/(s+4*pi*10^3);

Gz = c2d(G,T);
Sz = c2d(S,T);

K = 3.16529*0.6;
Ti = 0.000875;
Td = 0.0005;
```

```

Kp = K - K*T/(2*Ti);
Ki = K*T/Ti;
Kd = K*Td/T;


for k = 0:(NumeroDePeriodosDeQuadrada-1)
    for i=1:TempoDePeriodoDeQuadrada
        if(i≤TempoDePeriodoDeQuadrada/2)
            U(i+TempoDePeriodoDeQuadrada*k) = AmplitudeOndaQuadrada;
        else
          U(i+TempoDePeriodoDeQuadrada*k) = -1*AmplitudeOndaQuadrada;
        end
    end
end

for k = 3:(tfinal/T -1)
     Erro(k) = U(k) - Realimentacao(k);
     Up(k) = Kp*Erro(k);
     Ui(k) = Ki*Erro(k) + Ui(k-1);
     Ud(k) = Kd*(Erro(k)-Erro(k-1));
     PID(k) = Up(k) + Ui(k) + Ud(i);
     Planta(k) = 0.06452*PID(k-1) + 0.0521*PID(k-2) + 1.411*Planta(k-1) ...
         - 0.5273*Planta(k-2);
     Realimentacao(k+1) = 0.912*Planta(k)+0.081*Realimentacao(k);
end

figure(1);
stem([0:T:tfinal-T],U,'blue');
hold on;
stem([0:T:tfinal-T],Planta,'red');
```

O resultado desta simulação esta expressa na Figura 19.

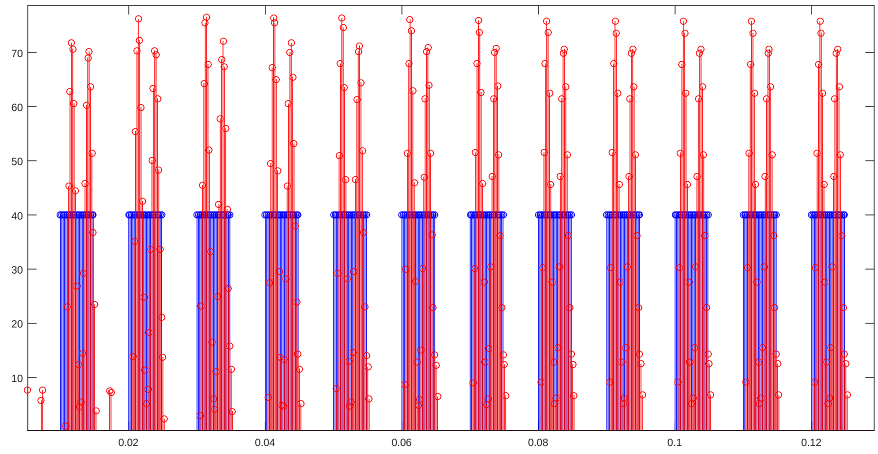

Figura 19: Simulação Exercicio 3 - Resposta em Malha Fechada

7 CONTROLADOR REPETITIVO

7.1 OBJETIVOS

7.2 FUNDAMENTAÇÃO TEÓRICA

7.3 PROCEDIMENTO E RESOLUÇÃO

O exercício proposto baseia-se em desenvolver uma simulação em ambiente Simulink® com as seguintes características:

- **Onda quadrada como referência**, com 12 V de *offset*, 12 Vpp de Amplitude e com frequência de 60Hz.
- **Planta** representada com a seguinte função de transferência:

$$P(s) = \frac{4 * 10^6}{s^2 + 3200s + 4 * 10^6} \tag{1}$$

- **Atuador** representado por um sinal PWM.
- **Medição do sinal**, ou sensor, representado pela seguinte função de transferência:

$$S(s) = \frac{4\pi^2 * 10^6}{s^2 + \pi 10^3 s + 4\pi^2 * 10^6} \tag{2}$$

- **Aquisição do sinal** efetuado por um A/D operando com 10 bits.
- **Controlador PID Digital**, conforme implementado no relatório anterior.
- **Controlador repetitivo** com os seguintes parâmetros estabelecidos:

  $C_{rp} = 0{,}92$

  $d = 3$

  $N = 100$

  $Q(z, z^{-1}) = 0{,}99$

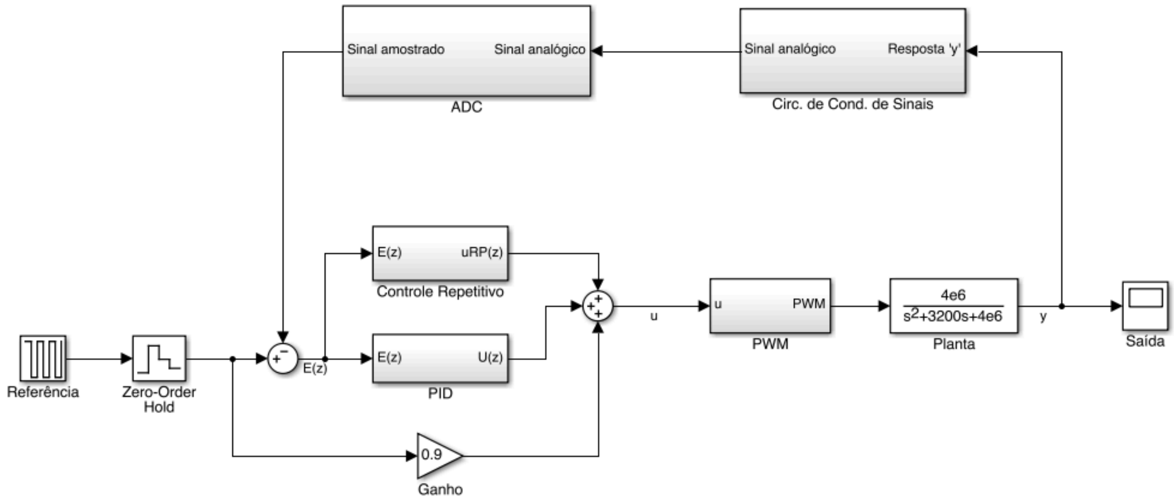

Figura 20: Sistema completo implementado em Simulink®

No ambiente de simulação **Simulink®**, esquematizou o sistema solicitado da seguinte maneira, conforme a Figura 20.

O sinal de referência, conforme solicitado nas especificações do exercício, foi implementado conforme a Figura 21.

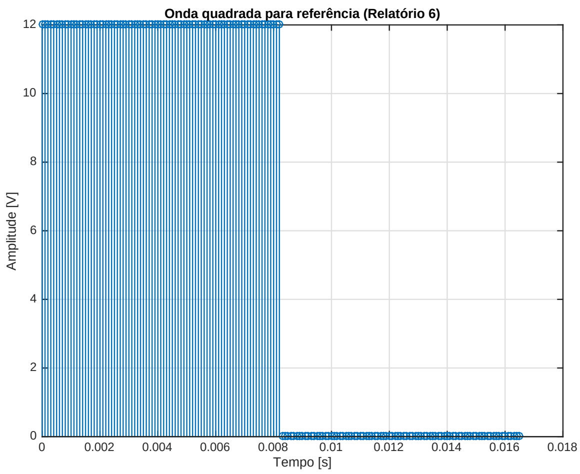

Figura 21: Sinal de referência, implementado em MATLAB®

Analisa-se pela Figura 21 que a onda quadrada tem 12 Vpp, período de $0,0167$ segundos (configurando em 60 Hz) e *offset* estabelecido.

O sistema implementado na Figura 20 é composto pelos seguintes blocos que serão explanados nas seguintes seções subsequentes:

7.3.1 Circuito de condicionamento de sinais

O sistema que representa o sensor, também apresentado como circuito de condicionamento de sinais, está apresentado na Figura 22.

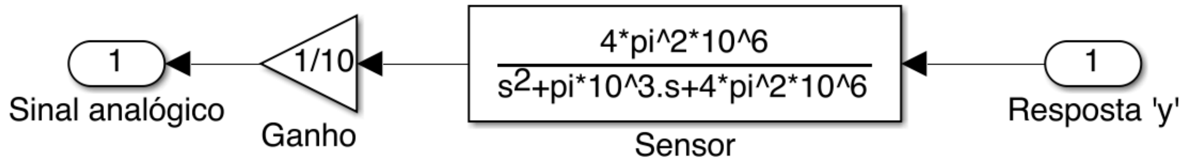

Figura 22: Circuito de condicionamento de sinais implementado em Simulink®

7.3.2 ADC de 10 bits

O sistema que representa a inserção do conversor analógico-digital está apresentado na Figura 23.

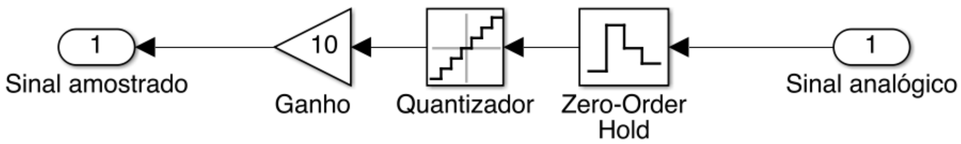

Figura 23: Sistema que representa a inserção do conversor analógico-digital implementado em Simulink®

7.3.3 Controlador repetitivo

O sistema que representa a aplicação do controlador repetitivo na malha de controle está apresentado na Figura 24.

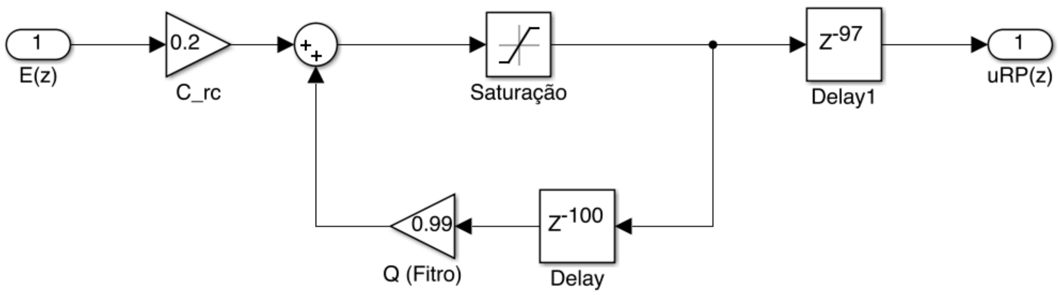

Figura 24: Sistema que representa a aplicação do controlador repetitivo na malha de controle implementado em Simulink®

7.3.4 Proporcional-Integral-Derivativo (PID)

O sistema que representa a aplicação do controlador proporcional-integral-derivativo na malha de controle está apresentado na Figura 25.

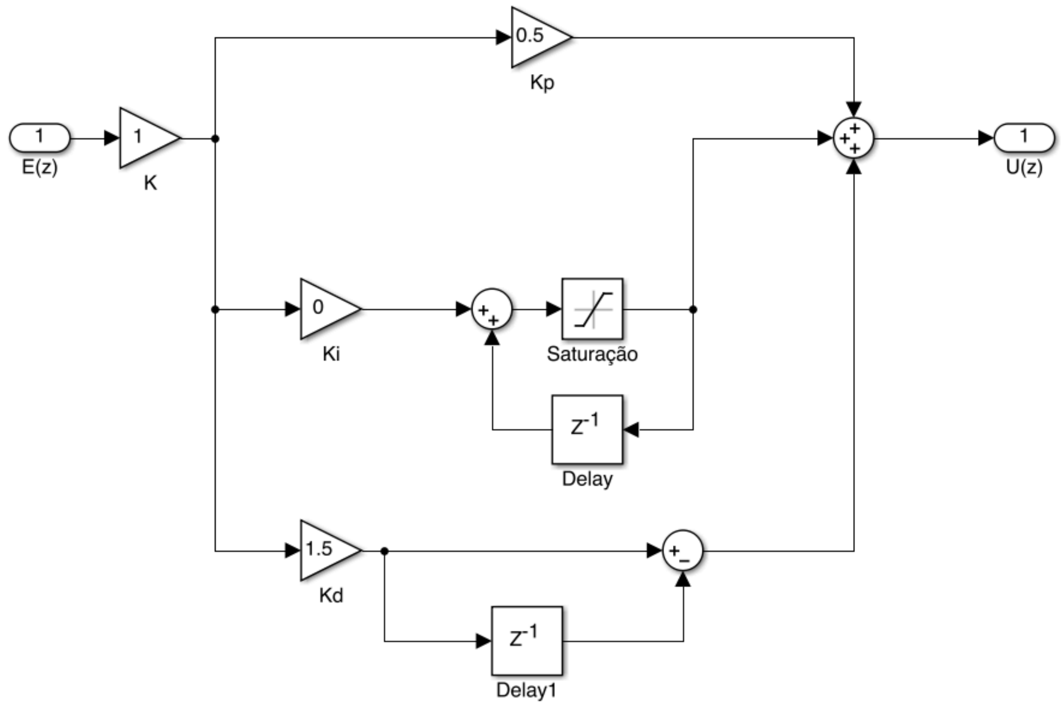

Figura 25: Sistema que representa a aplicação do controlador proporcional-integral-derivativo na malha de controle implementado em Simulink®

7.3.5 PWM

O sistema que representa a criação do sinal PWM está apresentado na Figura 26.

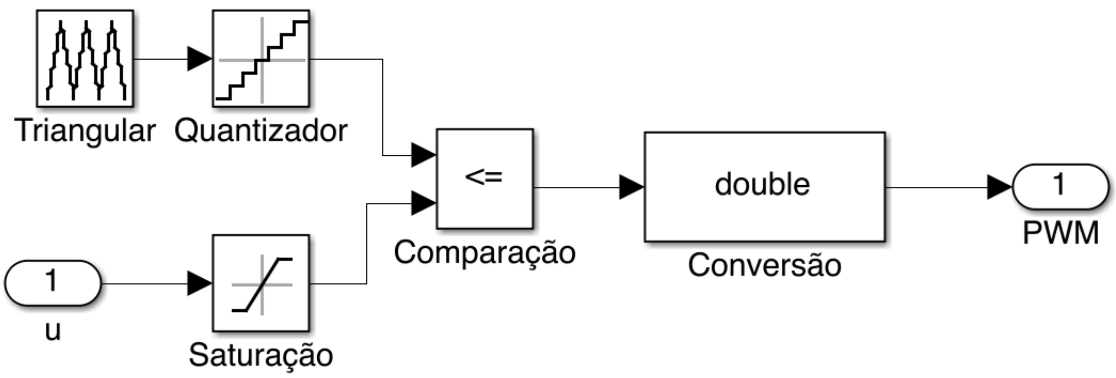

Figura 26: Circuito para criação de sinal PWM implementado em Simulink®

O sinal de onda triangular que será utilizado para comparação é apresentado conforme a Figura 27.

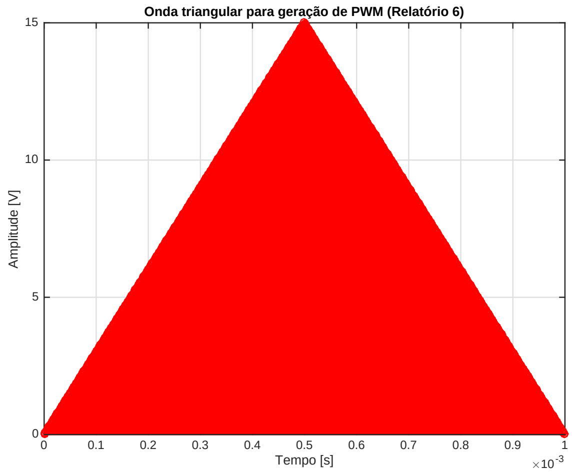

Figura 27: Sinal da onda triangular para comparação, implementado em MATLAB®

Percebe-se pela Figura 27 que a onda triangular tem amplitude de 15 V e período de 1 ms (Configurando frequência de 1 kHz).

7.4 RESULTADOS E DISCUSSÕES

8 CONCLUSÃO