

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 discussões	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	22
6	Controlador PID	23
6.1	Objetivos	23
6.2	Fundamentação Teórica	23
6.3	Procedimentos	23
6.3.1	Exercício 1	23
6.3.2	Exercício 2	23
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.	24
6.3.2.2	Inclua saturação na ação de controle em 150% da referência e analise o comportamento do sistema de controle	24
6.3.3	Exercício 3	24
6.4	Resultados e discursões	24
6.4.1	Exercício 1	24
7	Controlador Repetitivo	26
7.1	Objetivos	26
7.2	Fundamentação Teórica	26
7.3	Procedimento e resolução	26
7.3.1	Circuito de condicionamento de sinais	28
7.3.2	ADC de 10 bits	28
7.3.3	Controlador repetitivo	28
7.3.4	Proporcional-Integral-Derivativo (PID)	29
7.3.5	PWM	29
7.4	Resultados e discussões	30
8	Conclusão	31

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.

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

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

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.

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

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

2.3 PROCEDIMENTOS

Foram resolvidos os exercícios 1 e 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

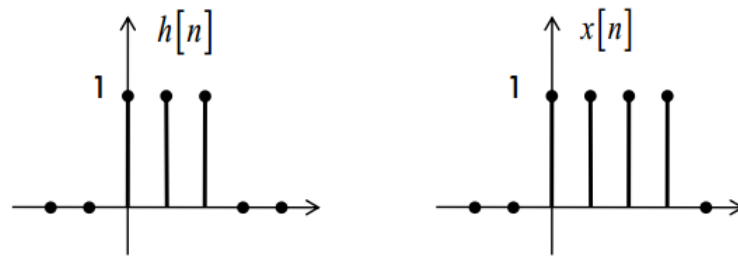


Figura 1: Sinais do Exercício 1, prática 1.

3,3V.

- A) Determine a resposta (sinal de saída) do sistema para 3 períodos do sinal de entrada considerando que o período 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:

```

1 %% -----
2 %   Universidade Tecnológica Federal do Parana
3 %   Engenharia Eletrica
4 %   Controle Digital
5 %
6 %   Aluno: Victor Emanuel Soares Barbosa
7 %
8 %   Aula 3: Transformada Z
9 %   Exercício 1:
10 %   Determine a saída do sistema com
11 %   resposta ao impulso h[n] e para um sinal de entrada
12 %   x[n]:
13 %   C) convolucao utilizando ferramenta
14 %   computacional: script Matlab
15 % -----
16 %% Inicializacao do programa

```

```

17  clc;
18  clear all;
19  close all;
20
21  %% Variaveis gerais
22  numero_pontos = 8; % Numero de pontos simulados
23  h = [1 1 1 zeros(1,numero_pontos-3)]; % resposta ao impulso
24  x = [1 1 1 1 zeros(1,numero_pontos-4)]; % sinal de entrada
25  y = [zeros(1,numero_pontos)]; % resposta do sistema a entrada x
26  amostras = zeros(1,numero_pontos); % valor de amostras
27
28  %% Letra c)
29
30  % Execucao
31  for n=0:(numero_pontos-1)
32      for k = 0:3
33          if (n-k)>0
34              y(n+1) = y(n+1) + x(k+1)*h(n-k);
35          end
36      end
37      amostras(n+1) = n-1;
38  end
39
40  % Graficos
41  figure
42  stem(amostras,y)
43  title('Aula 2 - Exercício 2 - Letra c');
44  legend('Sinal y[n]');
45  ylabel('Amplitude');
46  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

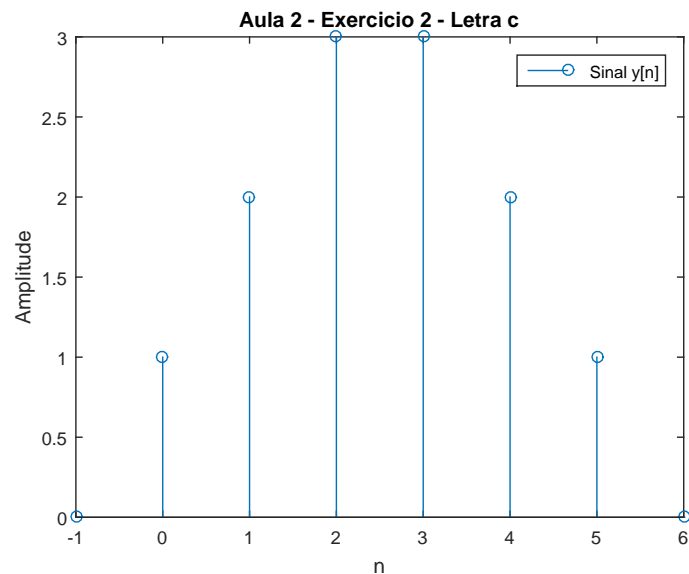


Figura 2: Gráfico de saída do código do Exercício 1, letra C, prática 1.

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 eficácia do algoritmo desenvolvido.

2.4.2 Exercício 2

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

```

1 %% -----
2 %   Universidade Tecnológica Federal do Parana
3 %   Engenharia Eletrica
4 %   Controle Digital
5 %
6 %   Aluno: Victor Emanuel Soares Barbosa
7 %
8 %   Aula 3: Transformada Z
9 %   Exercício 2:
10 %   Considere um sistema que possui resposta
11 %   ao impulso  $h[n]=2-nT$  e o sinal de entrada he uma onda
12 %   retangular (razao ciclica 40%,  $D=0,4$ ) com periodo
13 %   10s e amplitude 3,3V.
14 %   A) Determine a resposta (sinal de saída) do sistema para
15 %   3 periodos do sinal de entrada considerando que o

```

```
16 % periodo de amostragem he T=0.2s.
17 % B) Considere um ruido de 10% no sinal de entrada e
18 % repita o item A.
19 % C) Considere um sinal de entrada senoidal com mesmo
20 % periodo, amplitude e ruido dados acima.
21 % -----
22 %% Inicializacao do programa
23 clc;
24 clear all;
25 close all;
26
27 %% Variaveis gerais
28 periodos = 3; % Quantidade de periodos
29 R = 0.1; % Nivel de ruido
30 T = 0.2; % Periodo de amostragem
31 T_entrada = 10; % Periodo do sinal de entrada
32 A_entrada = 3.3; % Amplitude do sinal de entrada
33 D = 0.4; % Razao ciclica do sinal de entrada
34 total_pontos = periodos*T_entrada/T; % Total de pontos simulados
35 pontos_periodo = T_entrada/T; % Total de pontos por periodo
36 amostras = zeros(1,total_pontos); % Vetor de pontos de simulacao
37 h = zeros(1,total_pontos); % vetor da resposta ao impulso
38 x = zeros(1,total_pontos); % Vetor do sinal de entrada
39 y = zeros(1,total_pontos); % Vetor do sinal de saida
40 cont = 0;
41
42 %% Letra a)
43
44 % Execucao
45 for n = 0:total_pontos-1
46
47     if cont < (pontos_periodo*D)
48         x(n+1) = A_entrada;
49     else
50         x(n+1) = 0;
51     end
52
53     if cont == pontos_periodo
54         cont = 0;
55     end
56
```

```
57     cont = cont + 1;
58
59     for k = 0:pontos_perodo*3
60         h(n+1) = 2^(-n*T);
61
62         if (n-k)> 0
63             y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
64         end
65     end
66
67     amostras(n+1) = (n)*T;
68 end
69
70 % Graficos
71 figure
72 stem(amostras,y)
73 hold
74 stem(amostras,x)
75 stem(amostras,h)
76 title('Aula 2 - Exercicio 2 - Letra a');
77 legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
78 ylabel('Amplitude');
79 xlabel('Tempo (s)');
80
81 %% Letra b)
82
83 % Execucao
84 for n = 0:total_pontos-1
85
86     if cont < (pontos_perodo*D)
87         x(n+1) = A_entrada*(1+0.1*rand);
88     else
89         x(n+1) = 0.1*rand;
90     end
91
92     if cont == pontos_perodo
93         cont = 0;
94     end
95
96     cont = cont + 1;
97
```

```

98     for k = 0:pontos_periodo*3
99         h(n+1) = 2^(-n*T);
100
101         if (n-k)> 0
102             y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
103         end
104     end
105
106     amostras(n+1) = (n)*T;
107 end
108
109 % Graficos
110 figure
111 stem(amostras,y)
112 hold
113 stem(amostras,x)
114 stem(amostras,h)
115 title('Aula 2 - Exercicio 2 - Letra b');
116 legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
117 ylabel('Amplitude');
118 xlabel('Tempo (s)');
119
120 %% Letra c)
121
122 % Execucao
123 for n = 0:total_pontos-1
124
125     x(n+1) = A_entrada*sin(2*pi*n/(T_entrada/T))*(1+0.1*rand);
126
127     if cont == pontos_periodo
128         cont = 0;
129     end
130
131     cont = cont + 1;
132
133     for k = 0:pontos_periodo*3
134         h(n+1) = 2^(-n*T);
135
136         if (n-k)> 0
137             y(n+1) = y(n+1) + x(k+1)*h(n-k)*T;
138         end

```

```

139     end
140
141     amostras(n+1) = (n)*T;
142 end
143
144 % Graficos
145 figure
146 stem(amostras,y)
147 hold
148 stem(amostras,x)
149 stem(amostras,h)
150 title('Aula 2 - Exercício 2 - Letra c');
151 legend('Sinal y[n]', 'Sinal x[n]', 'Sinal h[n]');
152 ylabel('Amplitude');
153 xlabel('Tempo (s)');

```

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

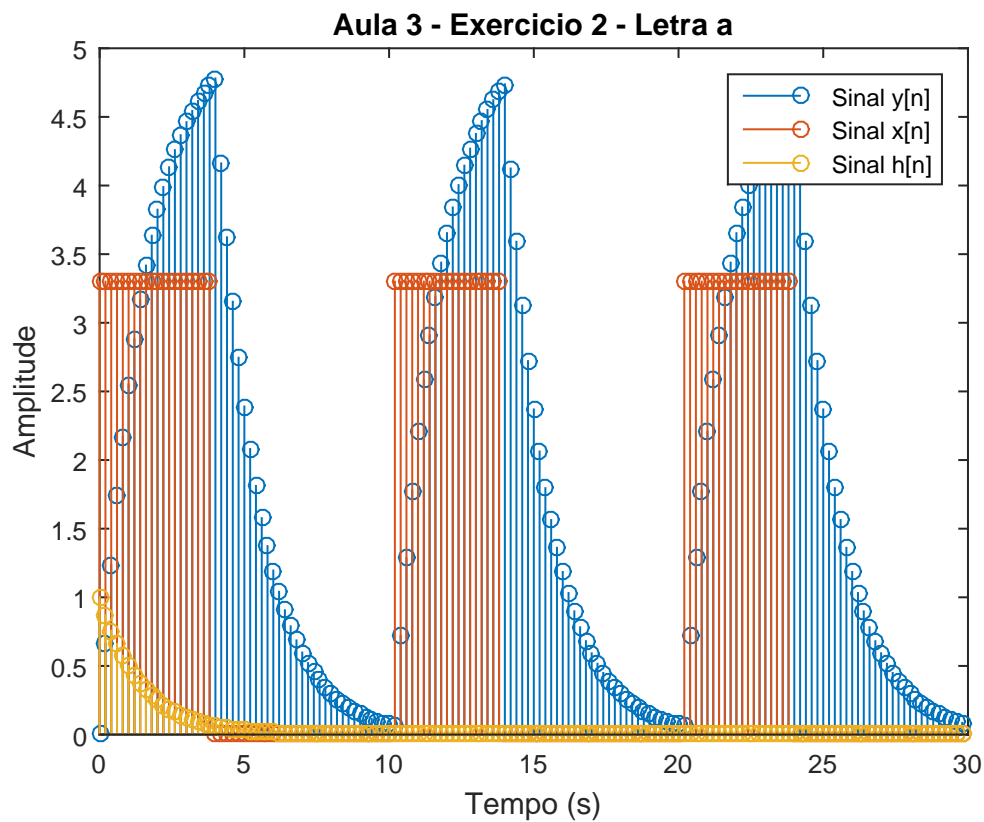


Figura 3: Gráfico de saída do código do Exercício 2, letra A, prática 1.

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

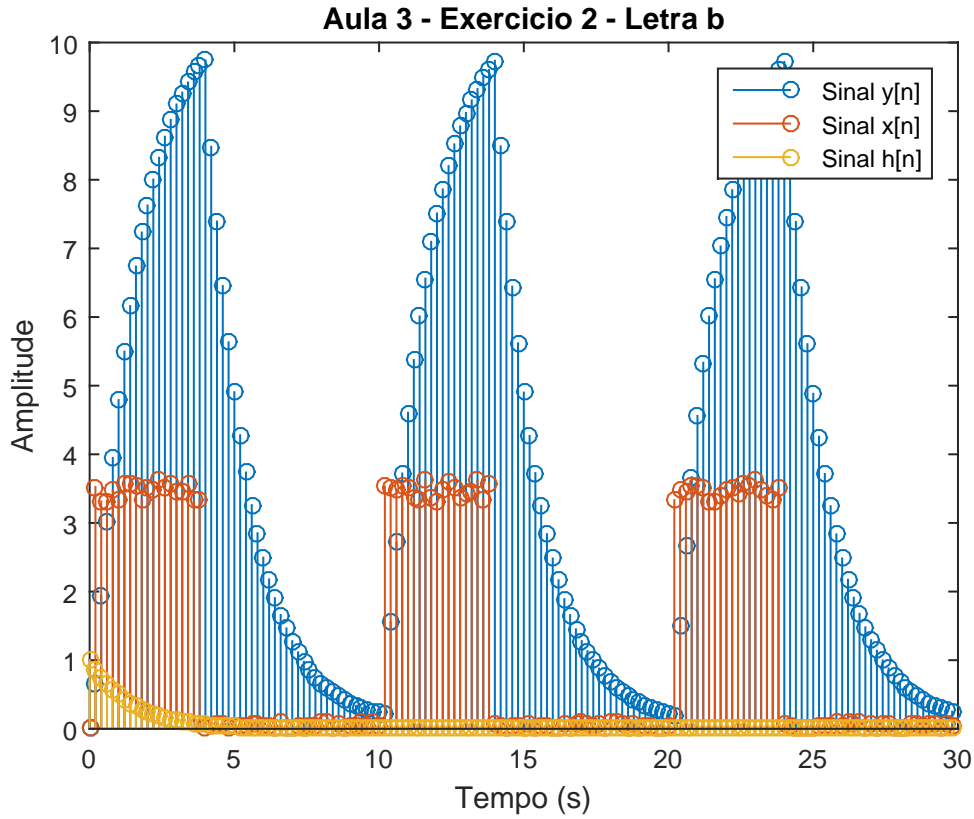


Figura 4: Gráfico de saída do código do Exercício 2, letra B, prática 1.

Como temos um período de amostragem de 0,2 logo temos 5 pontos simulados entre valores inteiros. Assim se fixarmos o número 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 número de amostras realizado. Para que se possa ter um sinal de amplitude normalizada independente do período de amostragem, o sinal convolucionado é dividido pelo número de amostras entre intervalo entre dois inteiros, ou número de amostras por ciclo.

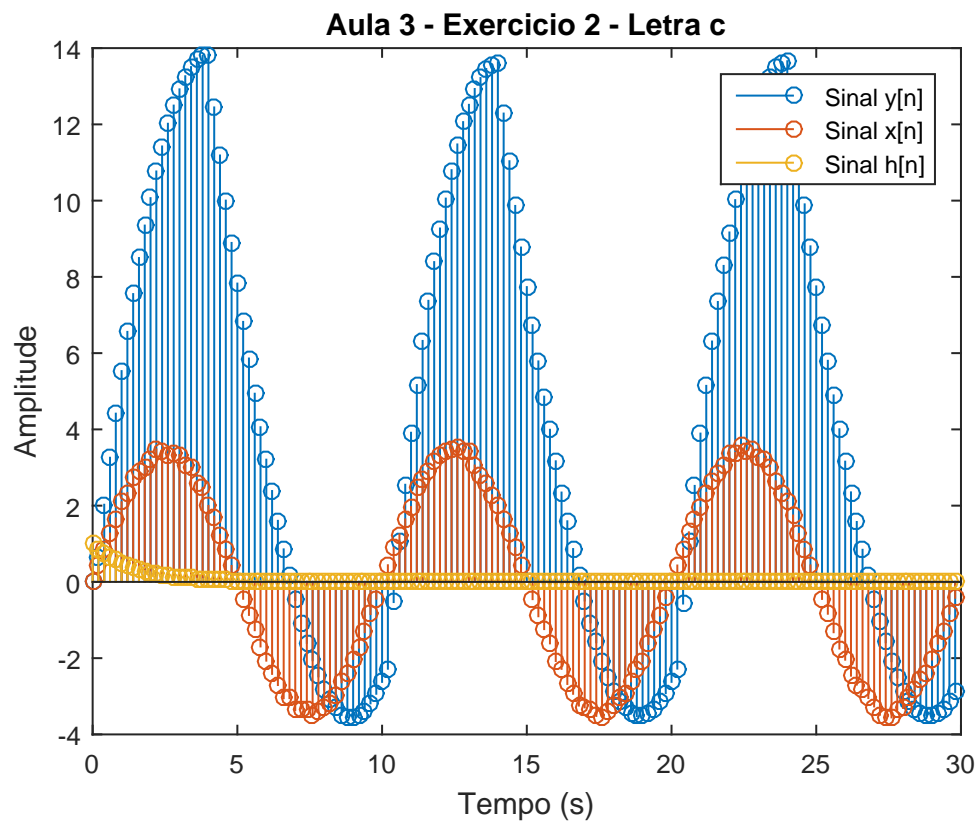


Figura 5: Gráfico de saída do código do Exercício 2, letra C, prática 1.

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 transformada 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)

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

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

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

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

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:

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

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

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

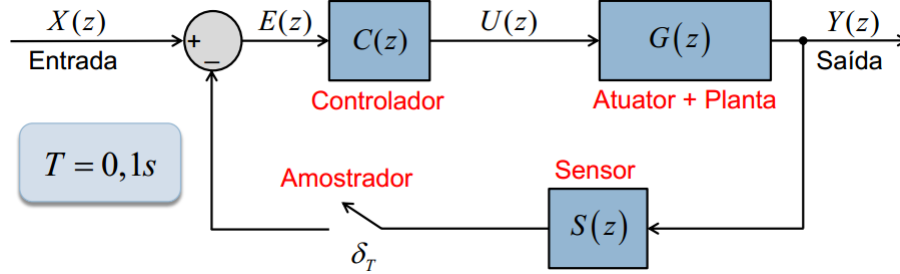


Figura 6: Diagrama de blocos da prática 2.

3.4 RESULTADOS E DISCUSSÕES

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

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

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

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

Através da equação (7) aplica-se o método da equação de diferença conforme apresentado na equação (2), concebendo a equação (??) 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:

```

1 %% -----
2 %   Universidade Tecnológica Federal do Parana
3 %   Engenharia Eletrica
4 %   Controle Digital
5 %
6 %   Aluno: Victor Emanuel Soares Barbosa
7 %
8 %   Aula 5: Transformada Z Inversa
9 %   Exercício 2:
10 %   Considere o diagrama de controle em tempo discreto.
11 %   Determinar o grafico da saída considerando:
12 % -----
13 %% Inicializacao do programa
14 clc;
15 clear all;
16 close all;
17
18 %% Variaveis gerais
19 ciclos = 4; % Quantidade de ciclos da onda de entrada
20 T_entrada = 10; % Periodo da onda de entrada
21 Ta = 0.1; % Periodo de amostragem
22 total_pontos = T_entrada/Ta*ciclos; % Total de pontos de simulacao
23 A_entrada = 5; % Amplitude de entrada
24 x = zeros(1,total_pontos); % Vetor da entrada
25 y = zeros(1,total_pontos); % Vetor de saída
26 tempo = zeros(1,total_pontos); % Vetor de tempo
27 cont = 0; % contador para auxiliar
28
29 for n = 1:total_pontos
30     if cont < ((T_entrada/Ta)/2)
31         x(n) = A_entrada*(1+0.02*rand);

```

```
32     else
33         x(n) = 0.02*rand*A_entrada;
34     end
35
36     if n > 3
37         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);
38     end
39
40     tempo(n) = n*Ta;
41     cont = cont + 1;
42
43     if cont > T_entrada/Ta
44         cont = 0;
45     end
46 end
47
48 % Grafico
49 figure
50 stem(tempo,y)
51 hold
52 stem(tempo,x)
53 title('Aula 5 - Exercício 2');
54 legend('Saída y[n]', 'Entrada x[n]');
55 ylabel('Amplitude');
56 xlabel('Amostrs');
```

Obtendo como saída o gráfico da Figura 7.

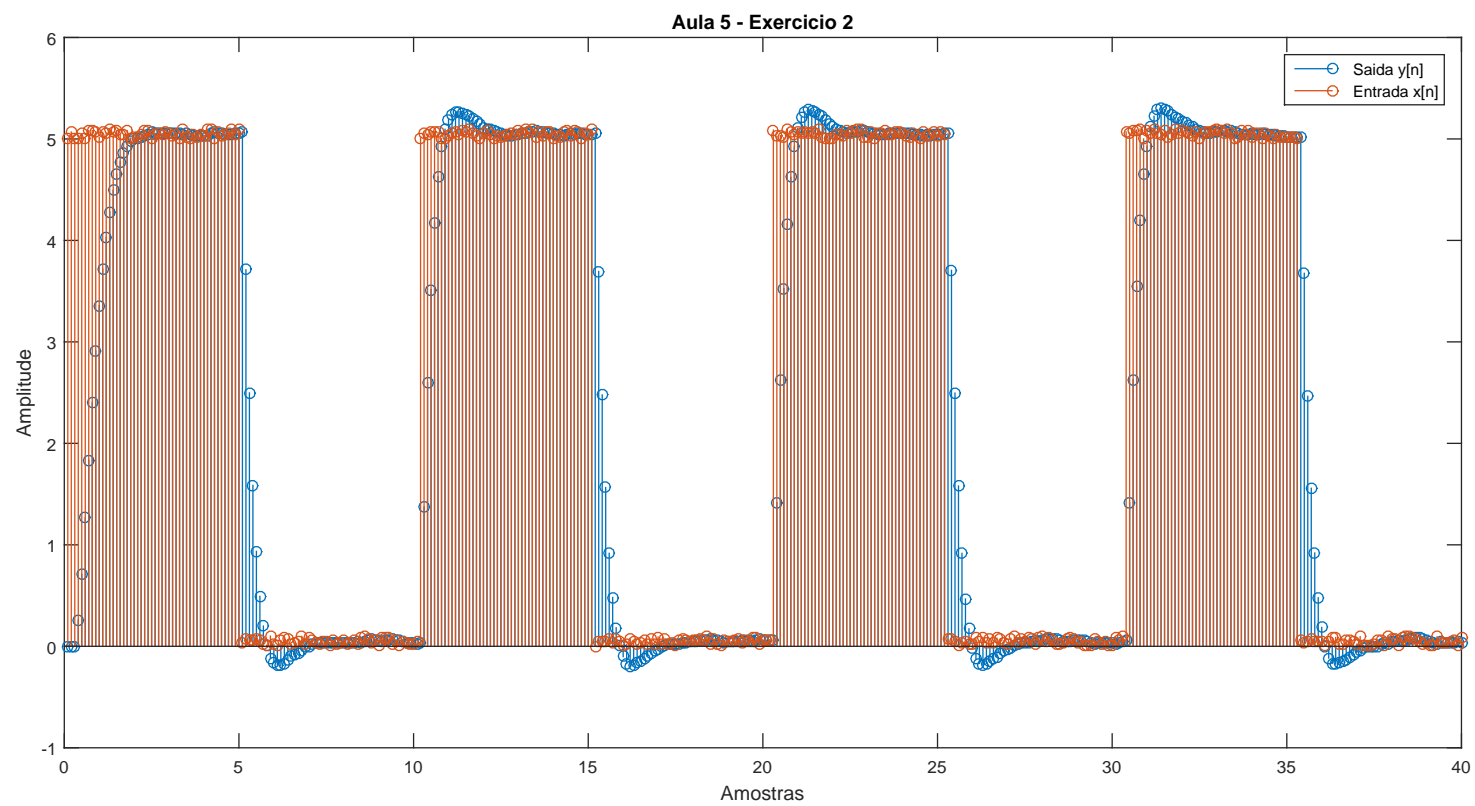


Figura 7: Gráfico de saída do código do exercício da prática 2.

COMENTAR E DISCUTIR RESULTADOS

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

5.2 FUNDAMENTAÇÃO TEÓRICA

5.3 PROCEDIMENTOS

5.4 RESULTADOS E DISCURSÕES

6 CONTROLADOR PID

6.1 OBJETIVOS

6.2 FUNDAMENTAÇÃO TEÓRICA

6.3 PROCEDIMENTOS

6.3.1 Exercício 1

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

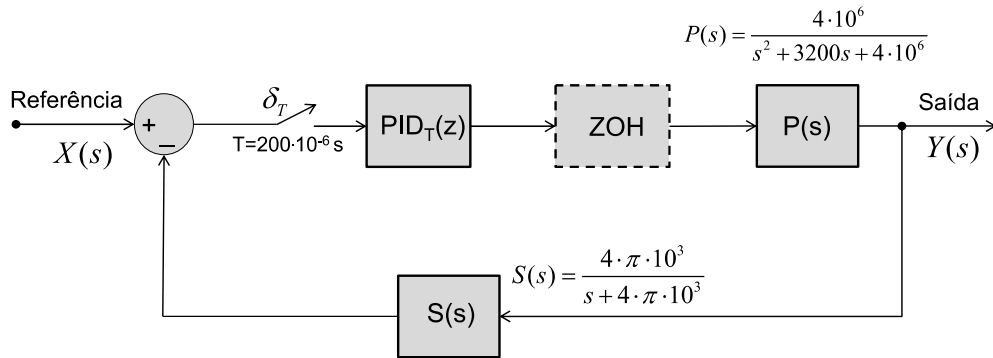


Figura 8: Sistema - Exercício 1

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 K_p , K_i , e K_d 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) $V_{in}=0-5V$ $T=200.10^{-6} s$
- 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 8, monta-se o circuito no Simulink®. A Figura 9 apresenta o dado circuito implementado.

A Figura 10 está apresentado o circuito PID implementado.

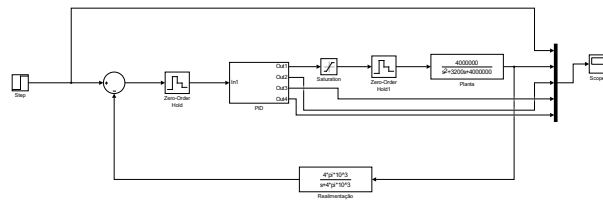


Figura 9: Diagrama de Blocos Exercício 1

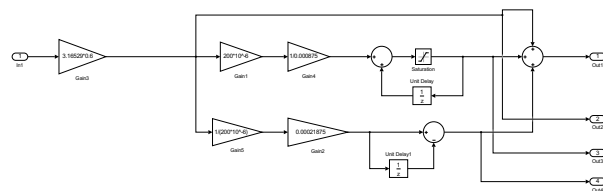


Figura 10: Diagrama de Blocos Exercício 1 - PID

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:

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

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

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

- **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:

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

$$d = 3$$

$$N = 100$$

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

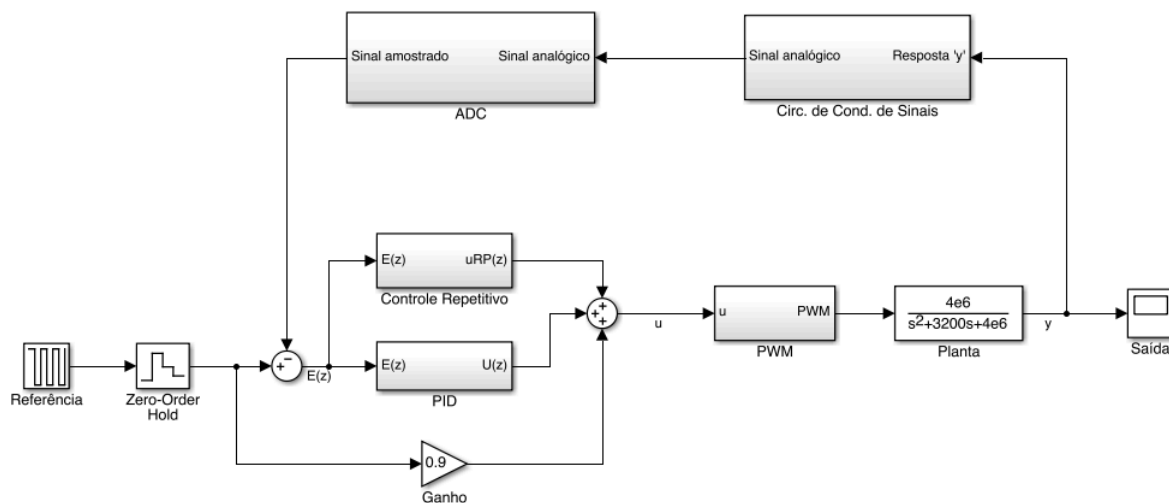


Figura 11: Sistema completo implementado em Simulink®

No ambiente de simulação **Simulink®**, esquematizou o sistema solicitado da seguinte maneira, conforme a Figura 11.

O sinal de referência, conforme solicitado nas especificações do exercício, foi implementado conforme a Figura 12.

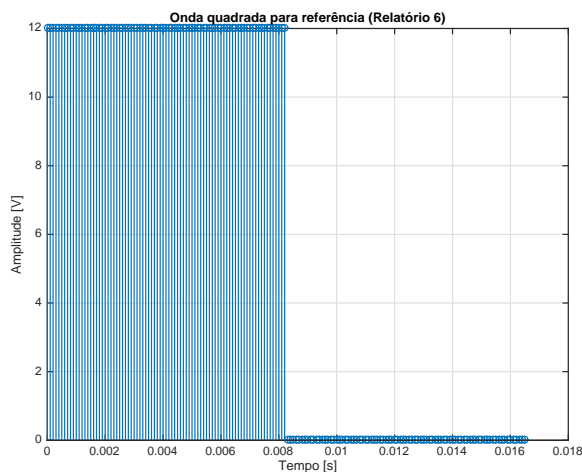


Figura 12: Sinal de referência, implementado em MATLAB®

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

O sistema implementado na Figura 11 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 13.

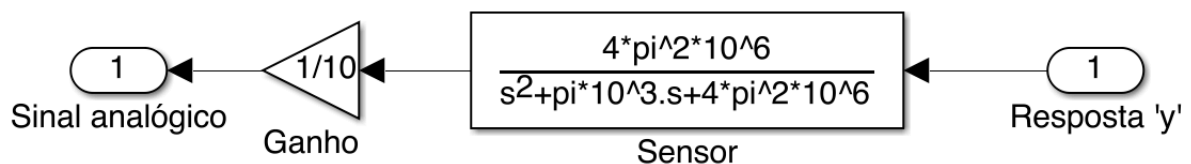


Figura 13: Circuito de condicionamento de sinais implementado em Simulink®

7.3.2 ADC de 10 bits

O sistema que representa a inserção do conversor analógico-digital está apresentado na Figura 14.

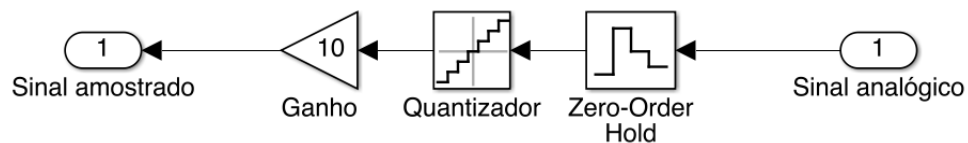


Figura 14: Sistema que representa a inserção do conversor analógico-digital implementado em Simulink®

7.3.3 Controlador repetitivo

O sistema que representa a aplicação do controlador repetitivo na malha de controle está apresentado na Figura 15.

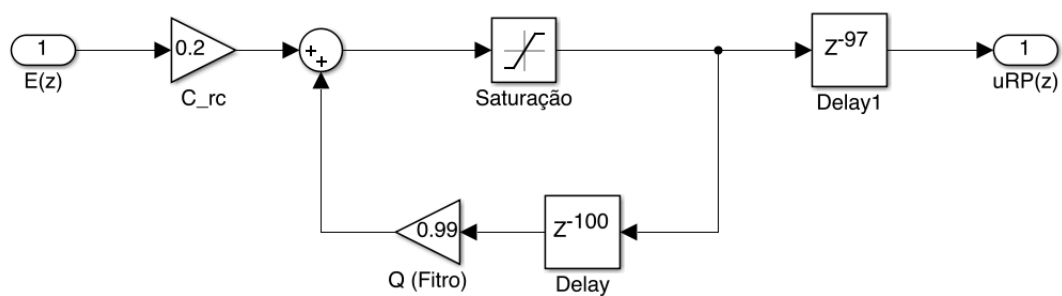


Figura 15: Sistema que representa a aplicação do controlador repetitivo na malha de controle implementado em Simulink®

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 16.

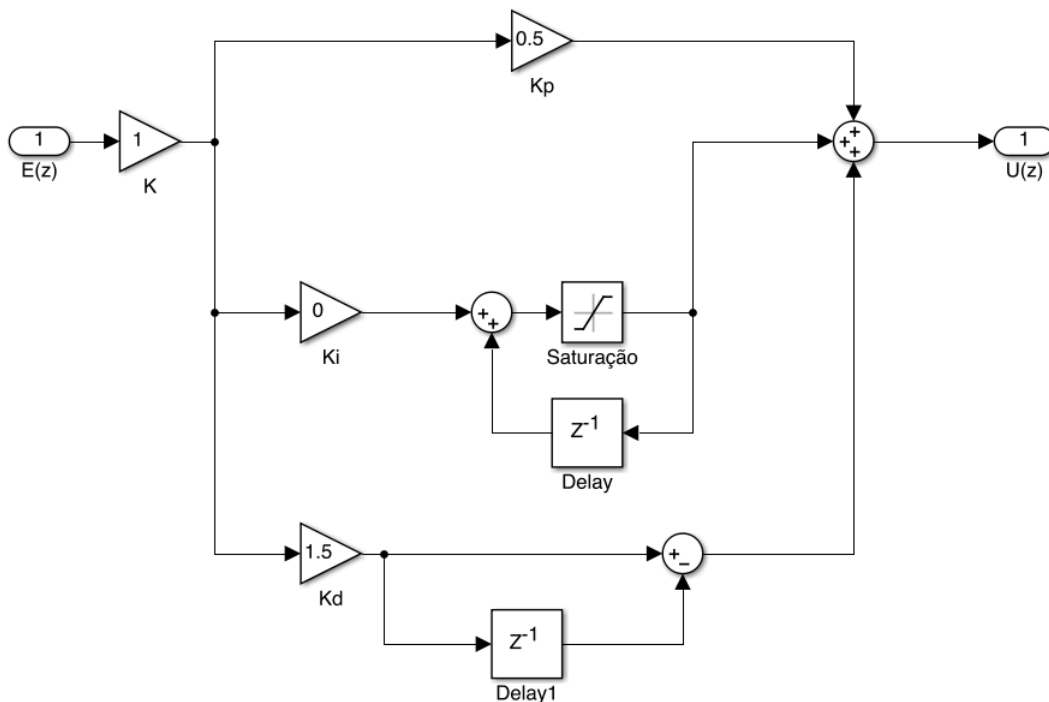


Figura 16: Sistema que representa a aplicação do controlador proporcional-integral-derivativo na malha de controle implementado em Simulink®

7.3.5 PWM

O sistema que representa a criação do sinal PWM está apresentado na Figura 17.

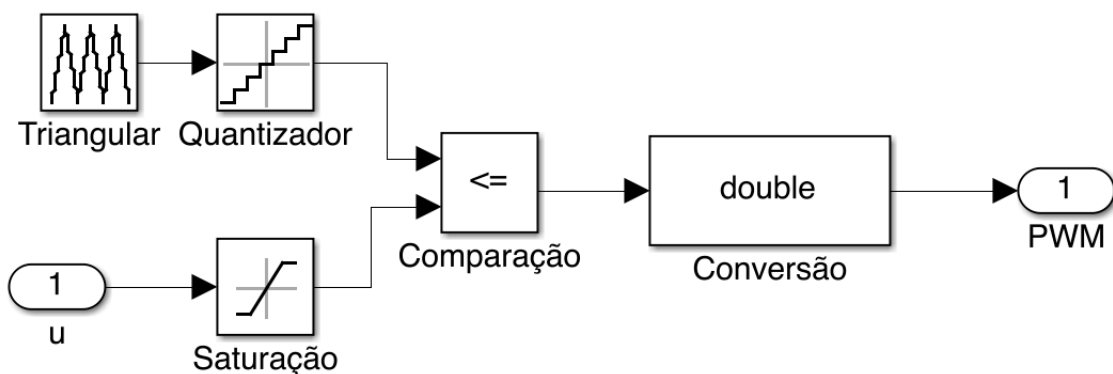


Figura 17: Circuito para criação de sinal PWM implementado em Simulink®

O sinal de onda triangular que será utilizado para comparação é apresentado

conforme a Figura 18.

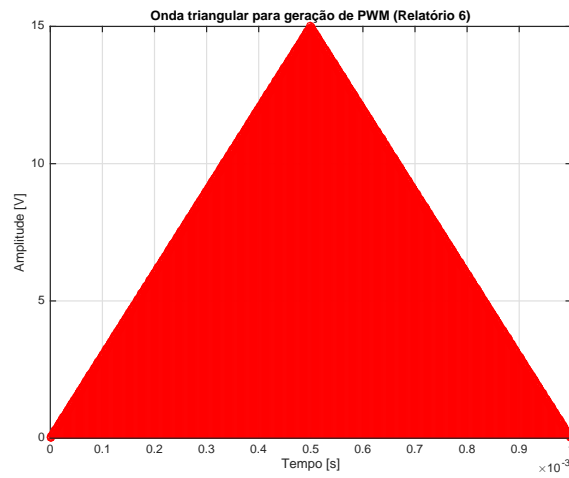


Figura 18: Sinal da onda triangular para comparação, implementado em MATLAB®

Percebe-se pela Figura 18 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

REFERÊNCIAS