

UTFPR
Engenharia de Computação
Controle Digital

Projeto 2: PID

Aluno: Deivid da Silva Galvão RA: 2408740

Aluno: João Vitor Levorato De Souza R.A: 2419890

Aluno: João Vitor N. Yoshida RA: 2419904

Aluno: Thiago Berto Minson, RA: 2270412

Professor orientador: Adalberto Zanatta Neder Lazarini

Dezembro
2024

UTFPR
Engenharia de Computação
Controle Digital

Projeto 2: PID

Relatório do Trabalho Prático Disciplinar apresentado como requisito parcial à obtenção de nota na disciplina de Controle Digital do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Aluno: Deivid da Silva Galvão RA: 2408740

Aluno: João Vitor Levorato De Souza R.A: 2419890

Aluno: João Vitor N. Yoshida RA: 2419904

Aluno: Thiago Berto Minson, RA: 2270412

Professor orientador: Adalberto Zanatta Neder Lazarini

Dezembro
2024

Conteúdo

1	Introdução	1
2	Implementação	1
3	Conclusão	12

1 Introdução

No projeto 2, vamos aplicar os conceitos aprendidos anteriormente para projetar três controladores, onde 2 são PID sendo um via Ziegler-Nichols, o outro por meio da transformação discreta e por fim um controlador Deadbeat do sistema massa mola amortecedor a baixo:

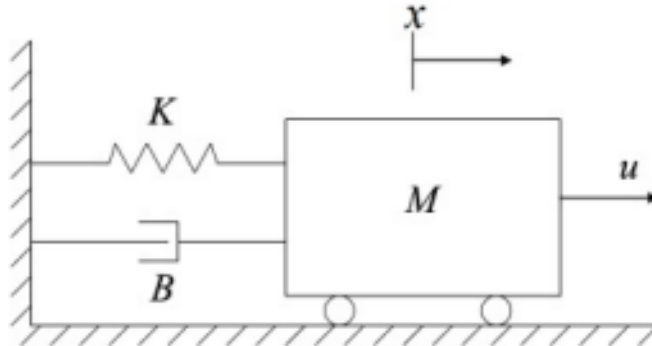


Figura 1: Sistema massa mola amortecedor

Descrito pela função transferência:

$$\frac{X(s)}{U(s)} = G(s) = \frac{1}{10s^2 + 11s + 120} \quad (1)$$

2 Implementação

Código Matlab

```
1 Alunos :
2 Deivid da Silva Galv o RA: 2408740
3 Jo o Vitor Levorato De Souza R.A: 2419890
4 Jo o Vitor N. Yoshida RA: 2419904
5 Thiago Berto Minson, RA: 2270412
6 %par metros gerais
7 M = 10; %massa do sistema massa mola
8 B = 11; %constante do amortecedor
9 K = 120; %constante de rigidez da mola
10
11 %fun o de transferencia
12 %X(s)/U(s) = G(s) = 1/(Ms^2 + Bs + K)
13 G = tf(1, [M B K]);
14
15 %Parte 1 - Ziegler-Nichols
16 %vari veis utilizadas:
17 ts = 0.1; %tempo de amostragem do enunciado
18
19 %visualiza o da resposta degrau
20 [a, b] = step(G);
21
22 %constantes encontradas
```

```

23 L = 0.157;
24 T = 0.455;
25
26 %opera es na tf
27 G_dig = c2d(G, 0.1);%converter a tf para digital
28 G_dig_mf = feedback(G_dig, 1); %malha fechada
29
30 %visualiza o
31 t = 0:ts:10; %janela de observa o
32 [stp_G_dig, t] = step(G_dig, t);
33 [stp_G_dig_mf, t] = step(G_dig_mf, t);
34 [imp_G_dig, t] = impulse(G_dig, t);
35 [imp_G_dig_mf, t] = impulse(G_dig_mf, t);
36
37 %degrau, malha aberta
38
39 figure
40 plot(t, stp_G_dig, 'k');
41 grid on
42 title('Resposta Entrada Degrau Malha Aberta');
43 xlabel('Tempo (s)');
44 ylabel('Amplitude');
45
46 %degrau, malha fechada
47 figure
48 plot(t, stp_G_dig_mf, 'g');
49 grid on
50 title('Resposta Entrada Degrau Malha Fechada');
51 xlabel('Tempo (s)');
52 ylabel('Amplitude');
53
54 %impulso, malha aberta
55 figure
56 plot(t, imp_G_dig, 'r');
57 grid on
58 title('Resposta Entrada Impulso Malha Aberta');
59 xlabel('Tempo (s)');
60 ylabel('Amplitude');
61
62 %impulso, malha fechada
63 figure
64 plot(t, imp_G_dig_mf, 'b');
65 grid on
66 title('Resposta Entrada Impulso Malha Fechada');
67 xlabel('Tempo (s)');
68 ylabel('Amplitude');
69
70 %valores de ganho
71 Kp = 1.2*(T/L);
72 Td = 0.5*L;

```

```

73 Ti = 2*L;
74
75 %Ki e Kd originais
76 %Ki = (Kp/Ti);
77 %Kd = (Kp * Td);
78
79 %com ajuste
80 Ki = (Kp/Ti) + 120;
81 Kd = (Kp * Td) + 30;
82
83 %sistema de controle
84 ZiegNic_cont = pid(Kp, Ki, Kd);
85 ZiegNic_sistComp = feedback(G*ZiegNic_cont, 1);
86
87 t = 0:ts:8;
88 [stp_ZiegNic, t] = step(ZiegNic_sistComp, t);
89 c = stepinfo(stp_ZiegNic, t);
90 disp(c)
91 [imp_ZiegNic, t] = impulse(ZiegNic_sistComp, t);
92
93 %degrau, ziegnic
94 figure
95 plot(t, stp_ZiegNic, 'b');
96 grid on
97 title('Resposta Entrada Degrau Ziegler-Nichols');
98 xlabel('Tempo (s)');
99 ylabel('Amplitude');
100
101 %impulso, ziegnic
102 figure
103 plot(t, imp_ZiegNic, 'g');
104 grid on
105 title('Resposta Entrada Impulso Ziegler-Nichols');
106 xlabel('Tempo (s)');
107 ylabel('Amplitude');
108
109 %Parte 2 - Controlador discreto
110 ContDisc_cont = tf([Ki*ts^2+2*Kp*ts+2*Kd Ki*ts^2-2*Kp*ts-4*Kd 2*
    Kd], [2*ts -2*ts 0], ts);
111 ContDisc_sistComp = feedback(ContDisc_cont * G_dig, 1);
112
113 t = 0:ts:8;
114 [stp_ContDisc, t] = step(ContDisc_sistComp, t);
115 [imp_ContDisc, t] = impulse(ContDisc_sistComp, t);
116
117 %degrau, discreto
118 figure
119 plot(t, stp_ContDisc, 'b');
120 grid on
121 title('Resposta Entrada Degrau Discreto');

```

```

122 xlabel('Tempo (s)');
123 ylabel('Amplitude');
124
125 %impulso, discreto
126 figure
127 plot(t, imp_ContDisc, 'g');
128 grid on
129 title('Resposta Entrada Impulso Discreto');
130 xlabel('Tempo (s)');
131 ylabel('Amplitude');
132
133 %Parte 3 - Deadbeat
134 DeadBeat_deg = tf(1,[1 0 0], ts); %definir um degrau
135 DeadBeat_cont = (1/G_dig) * (DeadBeat_deg/(1-DeadBeat_deg));
136 DeadBeat_sistComp = feedback(DeadBeat_cont * G_dig, 1);
137
138 %degrau, deadbeat
139 %DeadBeat_stp =
140 step(DeadBeat_sistComp, t);
141 %impulso, deadbeat
142 %DeadBeat_imp =
143 impulse(DeadBeat_sistComp);
144
145 %comparacao ts=0.1
146 t = 0:ts:8;
147 figure
148 plot(t, stp_ZiegNic, 'r');
149 grid on
150 hold on
151 title('Compara o PID Cont nuo e Discreto - T = 0.1s');
152 xlabel('Tempo (s)');
153 ylabel('Amplitude');
154 plot(t, stp_ContDisc, 'g');
155 legend('Contínuo', 'Discreto');
156
157
158 %comparacao ts=0.001
159 t = 0:0.001:8;
160 G_dig = c2d(G, 0.001);
161 ts2 = 0.001
162 ContDisc_cont2 = tf([Ki*ts2^2+2*Kp*ts2+2*Kd Ki*ts2^2-2*Kp*ts2-4*
    Kd 2*Kd], [2*ts2 -2*ts2 0], ts2);
163 ContDisc_sistComp2 = feedback(ContDisc_cont2 * G_dig, 1);
164 [stp_ZiegNic2, t] = step(ZiegNic_sistComp, t);
165 [stp_ContDisc2, t] = step(ContDisc_sistComp2, t);
166
167 figure
168 plot(t, stp_ZiegNic2, 'r');
169 grid on
170 hold on

```

```

171 title('Compara o PID Contínuo e Discreto - T = 0.001s');
172 xlabel('Tempo (s)');
173 ylabel('Amplitude');
174 plot(t, stp_ContDisc2, 'g');
175 legend('Contínuo','Discreto');

```

Para projetar o controlador PID via Ziegler-Nichols inicialmente foi simulado a resposta ao degrau em malha aberta, onde foi estimado o ponto de maior derivada da resposta por meio da análise da curva, no caso os valores estimados de L e T obtidos foram 0.157s e 0.455s respectivamente.

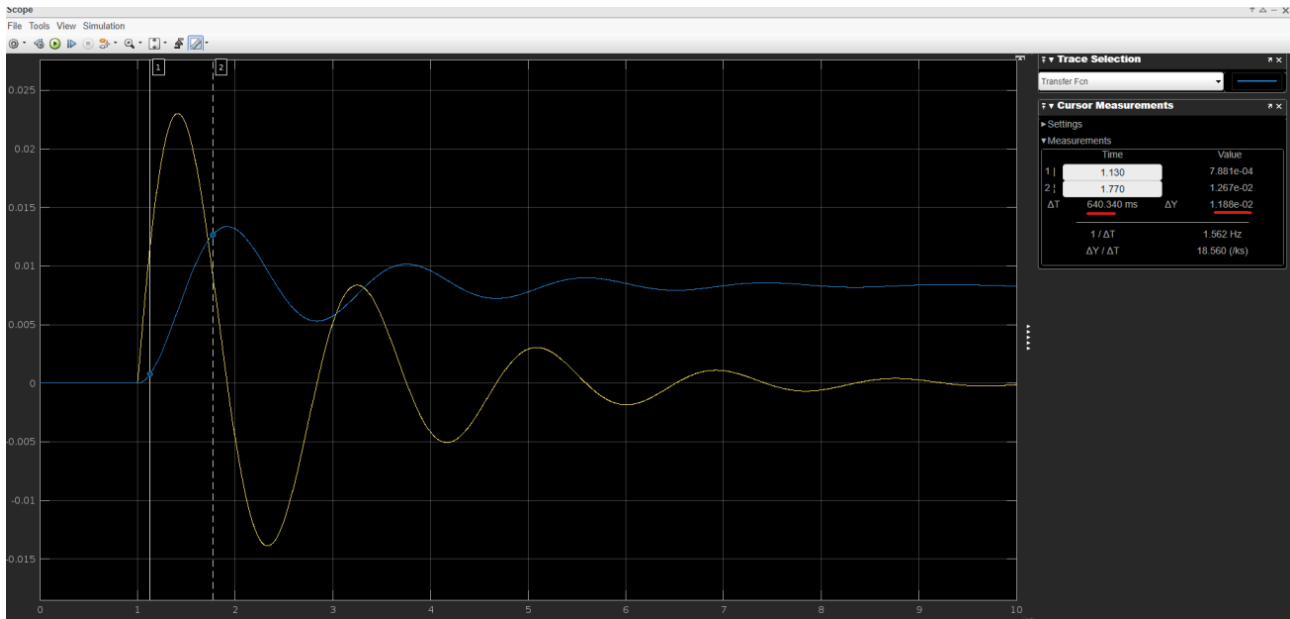


Figura 2: Simulação do sistema

Apos isso colocamos os valores obtidos de L e T nas formulas para encontrar k_p , t_i e t_d :

$$k_p = 1,2 \frac{T}{L} \rightarrow KP = 1,2 \frac{0,455}{0,157} \rightarrow 3,4777 \quad (2)$$

$$t_i = 2L \rightarrow t_i = 0,3140 \quad (3)$$

$$t_d = 0,5L \rightarrow t_d = 0,0785 \quad (4)$$

Com esses valores em mãos encontramos os valores de k_i e k_d :

$$k_i = \frac{k_p}{t_i} \rightarrow k_i = 11,0755 \quad (5)$$

$$k_d = k_p * t_d \rightarrow k_d = 0,2730 \quad (6)$$

Ao plotar o sistema com os valores obtidos, foram gerados gráficos para as respostas a uma entrada degrau e a um impulso:

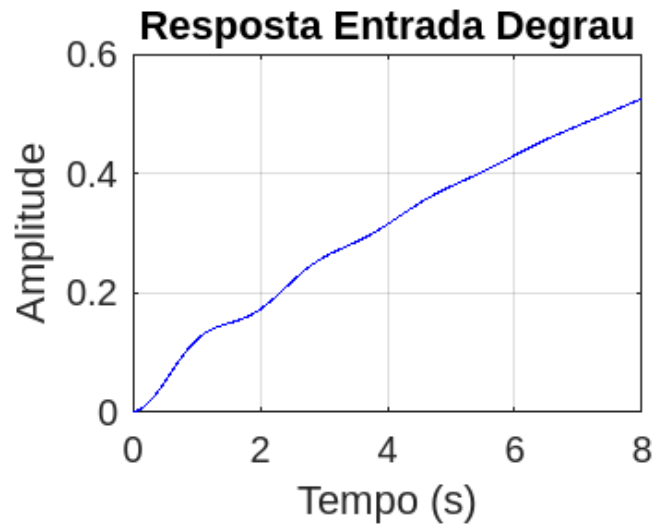


Figura 3: Resposta Entrada Degrau

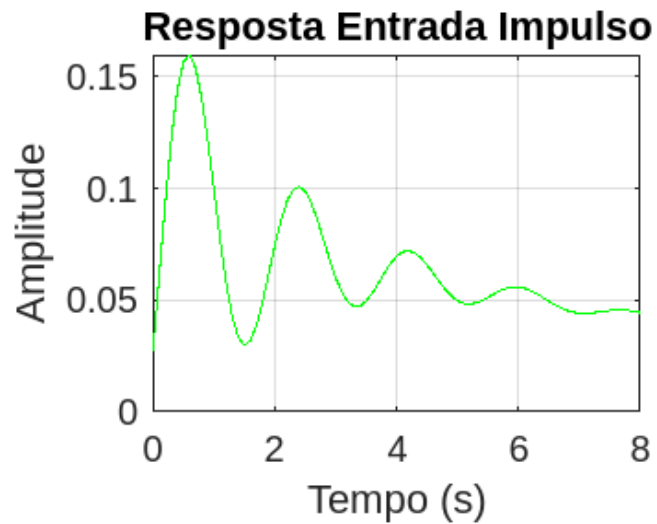


Figura 4: Resposta Entrada Impulso

Os gráficos mostram que o sistema não estabiliza e não atende aos parâmetros de projeto especificados: overshoot $\leq 20\%$ e tempo de estabelecimento $\leq 3s$.

Com isso por meio da tentativa e erro foi ajustado os valores de k_i e k_d para atender as especificações do projeto e foi obtido os valores de $k_i + 120$ e $k_d + 30$, e ao plotar novamente foi gerado os seguintes gráficos:

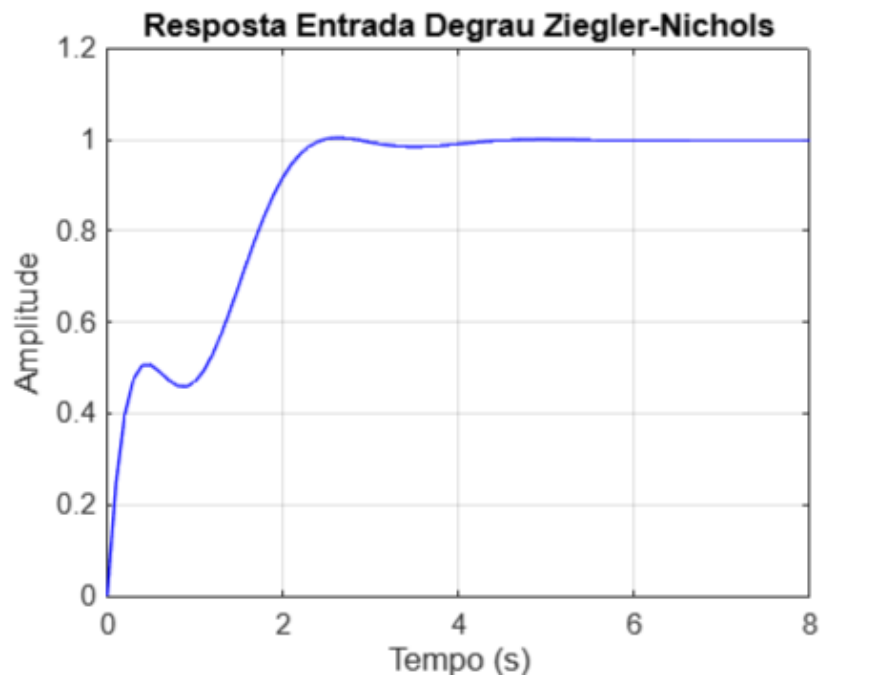


Figura 5: Resposta Entrada Degrau Corrigido

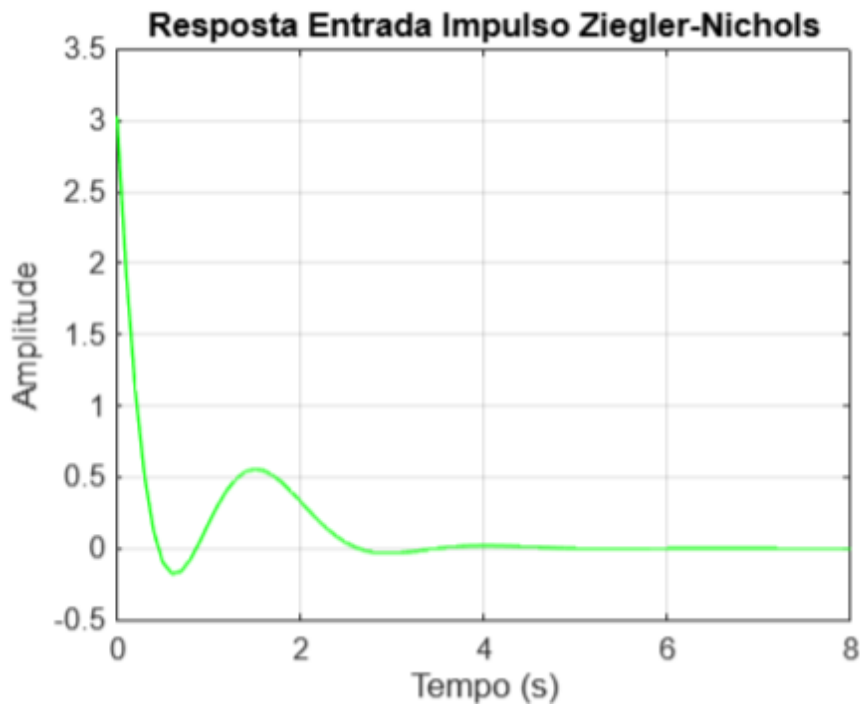


Figura 6: Resposta Entrada Impulso Corrigido

Nas figuras 5 e 6 é possível observar que o sistema converge para 1 com entrada em degrau e converge para zero quando entrada impulso, com um tempo de estabelecimento menor que 3 e um overshoot de 0.44 %.

Apos isso foi projetado o controlador discretizado onde foi usado a seguinte forma de função transferência:

$$\frac{U(s)}{E(s)} = \frac{(KiT + 2KpT + 2Kd)z + (KiT - 2KpT - 4Kd)z + 2Kd}{2Tz(z - 1)} \quad (7)$$

Usando os mesmos valores de k_p , k_i e k_d obtidos no método anterior e um tempo de amostragem de 0,1s foi obtido os seguintes gráficos:

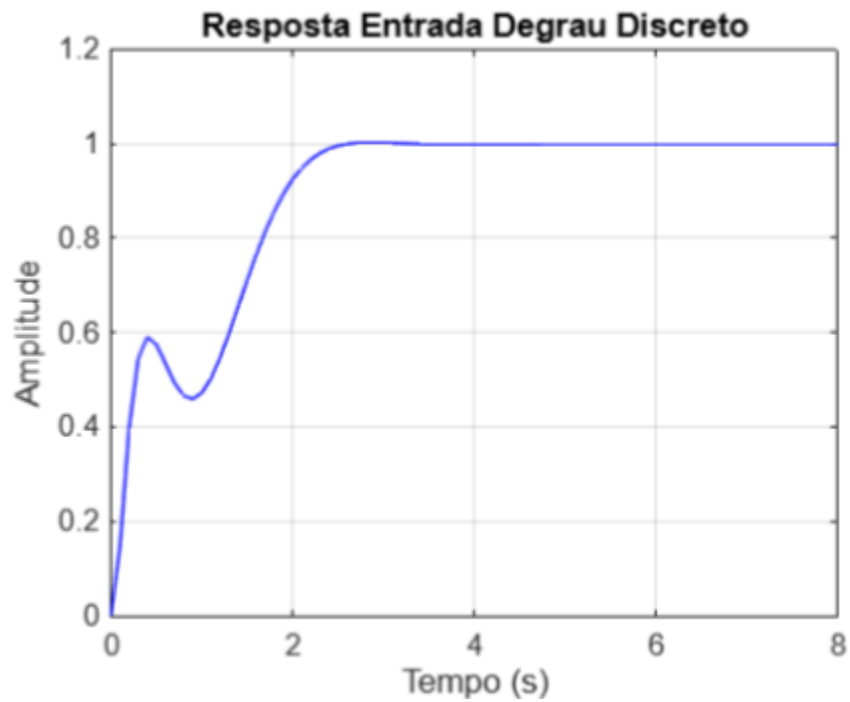


Figura 7: Resposta Entrada Degrau Corrigido

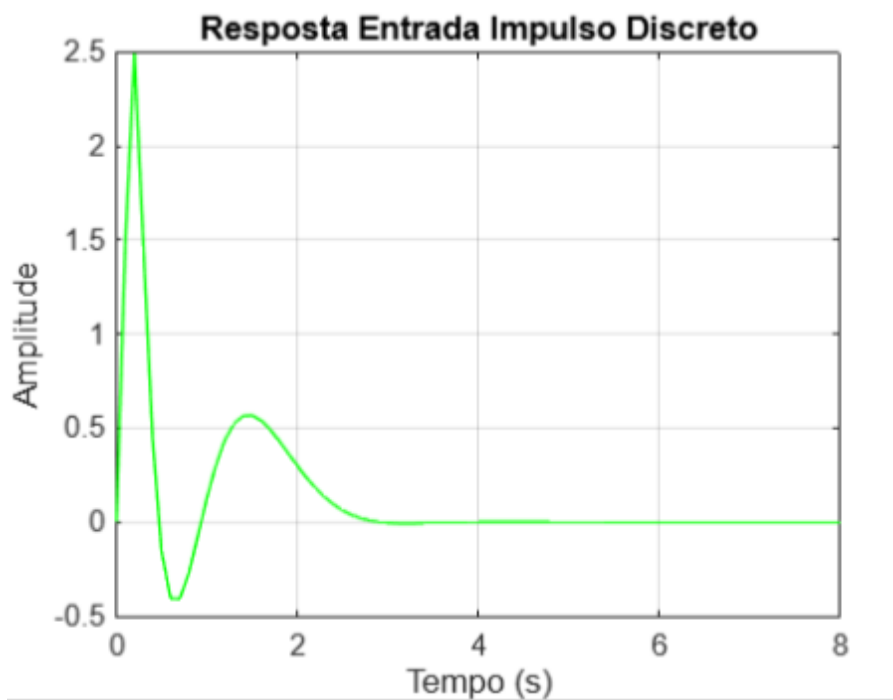


Figura 8: Resposta Entrada Impulso Corrigido

Ao comparar os gráficos de entrada degrau dos dois controladores verifica-se que são muito semelhantes com o tempo de amostragem de 0,1s, e caso o tempo de amostragem seja de 0,001 eles ficam praticamente sobrepostos confirmando que os resultados obtidos são satisfatórios.

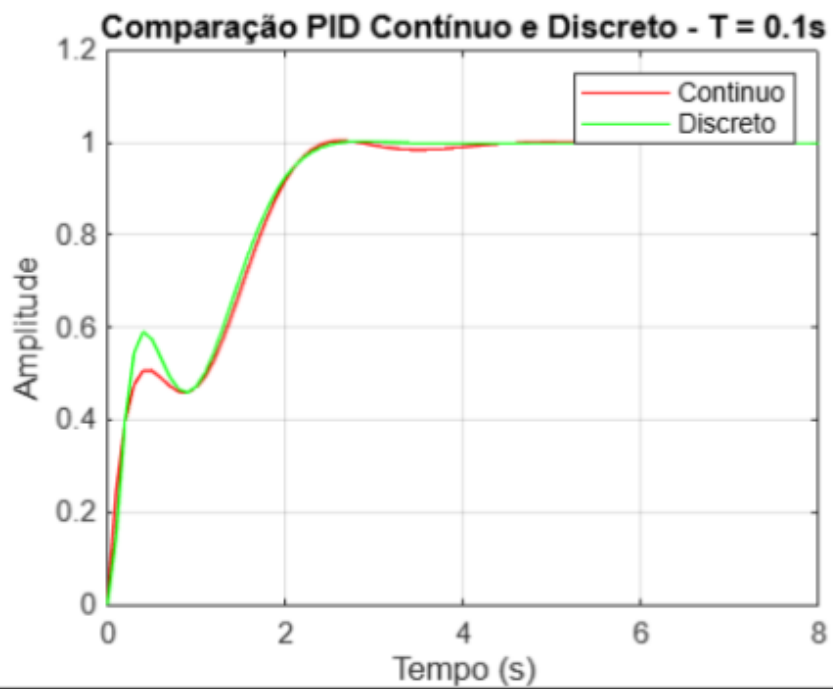


Figura 9: Comparação PID com $t = 0,1$

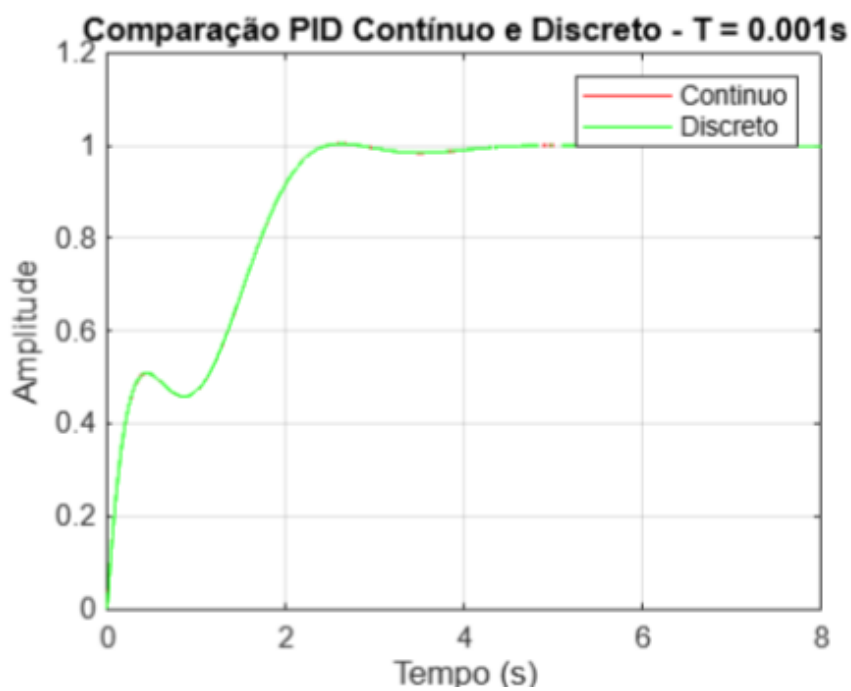


Figura 10: Comparação PID com $t = 0,001$

Por fim foi projetado o controlador Deadbeat, a função transferência deadbeat $M(z)$ varia dependendo do tipo de entrada, no caso para uma entrada degrau a equação é a seguinte:

$$M(z) = \frac{1}{Z^n} \quad (8)$$

E a partir da função transferência discreta chega-se a seguinte equação:

$$D(z) = \frac{1}{G(z)} \frac{M(z)}{1 - M(z)} \rightarrow D(z) = \frac{z^4 - 1,783z^3 + 0,8958z^2}{0,0004774z^5 + 0,0004601z^4 - 0,0004774z^3 - 0,0004601z^2} \quad (9)$$

Com isso foi verificado a resposta do sistema controlado em malha fechada para uma entrada degrau e também para fins de teste para entrada impulso. Os resultados desse processo podem ser observados nas figuras a seguir:

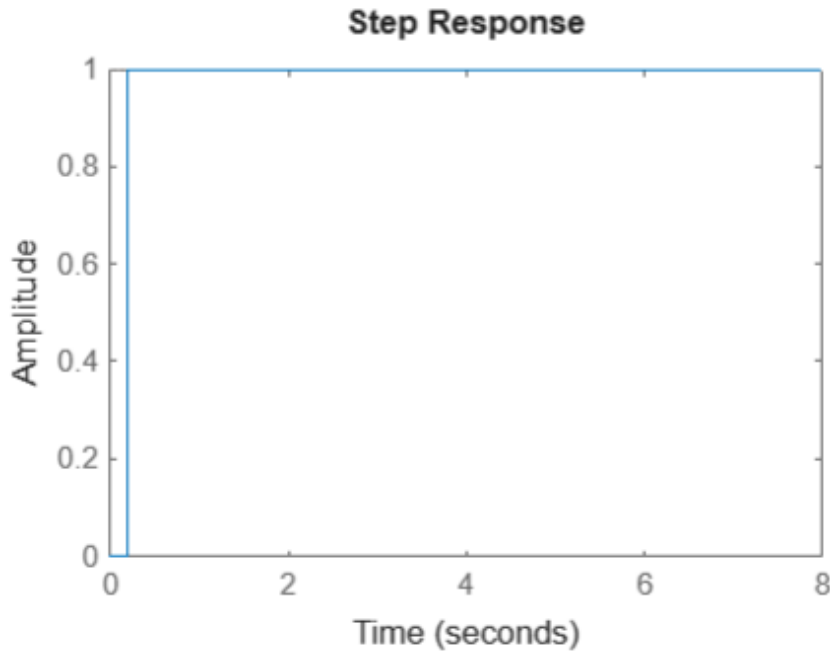


Figura 11: Resposta ao degrau para o controlador deadbeat

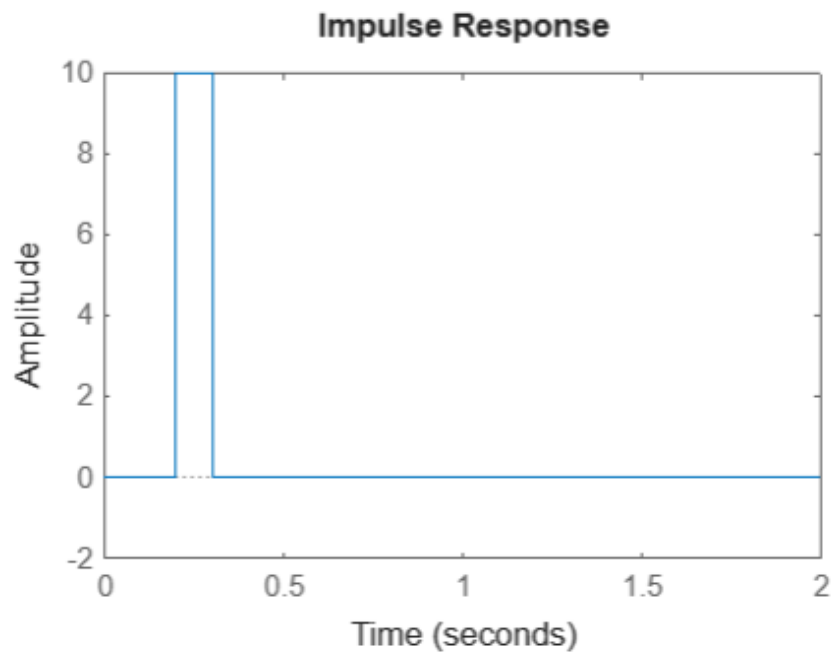


Figura 12: Resposta ao impulso para o controlador deadbeat

Por fim, na figura a seguir está o esquema completo do simulink dos 3 controladores:

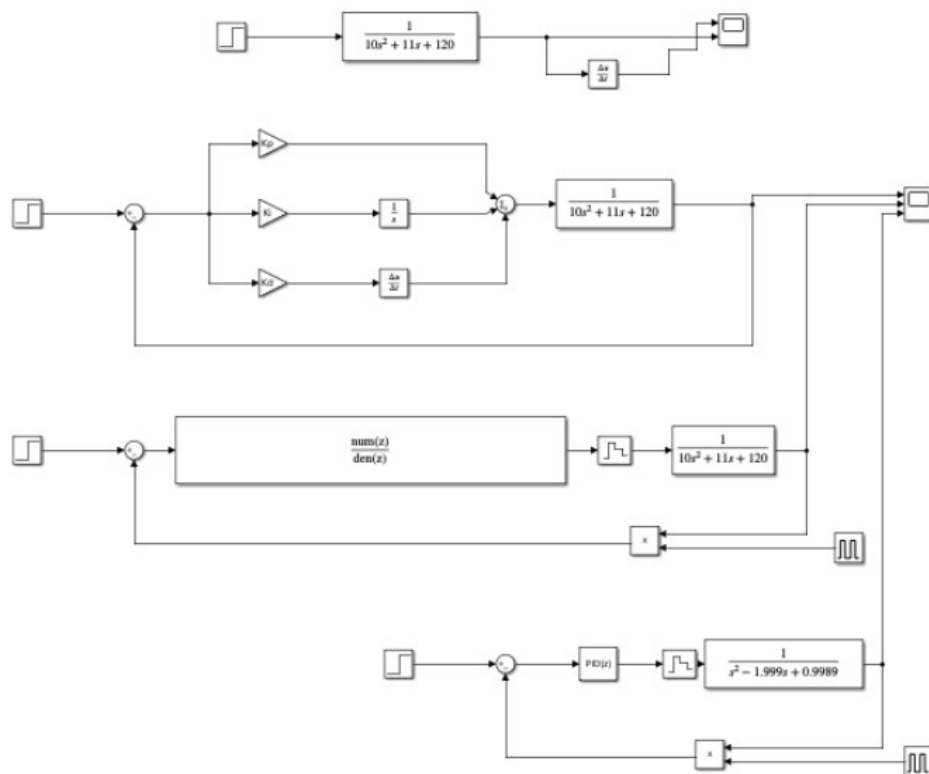


Figura 13: Simulink

3 Conclusão

Pode-se concluir que os controladores apresentaram resultados satisfatórios e foram devidamente ajustados conforme as especificações do projeto, porém vale destacar a importância de escolher um tempo de amostragem e valores de k_i e k_d adequados, pois causa um impacto significativo no desempenho do sistema.