

UTFPR
Engenharia de Computação
Controle Digital

Projeto 1: Root Locus

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

Novembro
2024

UTFPR
Engenharia de Computação
Controle Digital

Projeto 1: Root Locus

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

Outubro
2024

Conteúdo

1	Introdução	1
2	Implementação	1
3	Conclusão	8

1 Introdução

No projeto 1, vamos aplicar os conceitos aprendidos anteriormente para projetar um controlador digital para o sistema descrito pela função de transferência contínua:

$$\frac{s + 20}{s^2 + 1.5s + 35} \quad (1)$$

O Controlador deve atender os parametros de overshoot $\leq 20\%$, Tempo de subida $\leq 0,2s$ e Tempo de estabelecimento $\leq 0,4s$. Para alcançar esses objetivos, será realizado um estudo do comportamento dinâmico do sistema, seguido pela aplicação de técnicas de controle digital. Este processo envolve a discretização da função de transferência contínua, a seleção de um método apropriado de projeto de controlador digital, e a verificação de que o controlador atende às especificações de desempenho estabelecidas. Este relatório apresenta o desenvolvimento do controlador digital, desde a modelagem inicial até a validação final do desempenho do sistema controlado.

2 Implementação

Pode-se observar que o sistema contínuo é estável, uma vez que seus polos estão localizados na parte negativa do plano s . Essa conclusão foi alcançada ao calcular as raízes do denominador da função de transferência, que são: $-0.75 \pm 5,8684i$. As duas Figuras abaixo representam a saída do sistema contínuo com malha aberta, tanto para a entrada impulso quanto para a entrada degrau.

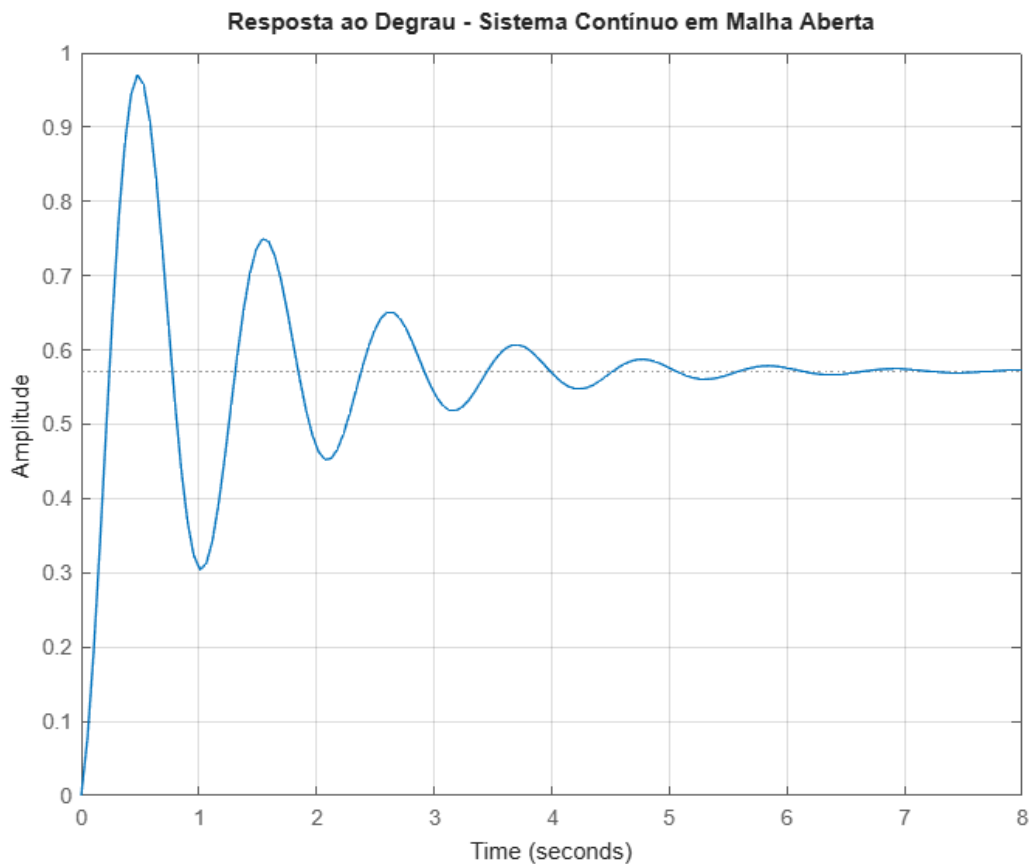


Figura 1: Sistema Continuo em Malha Aberta

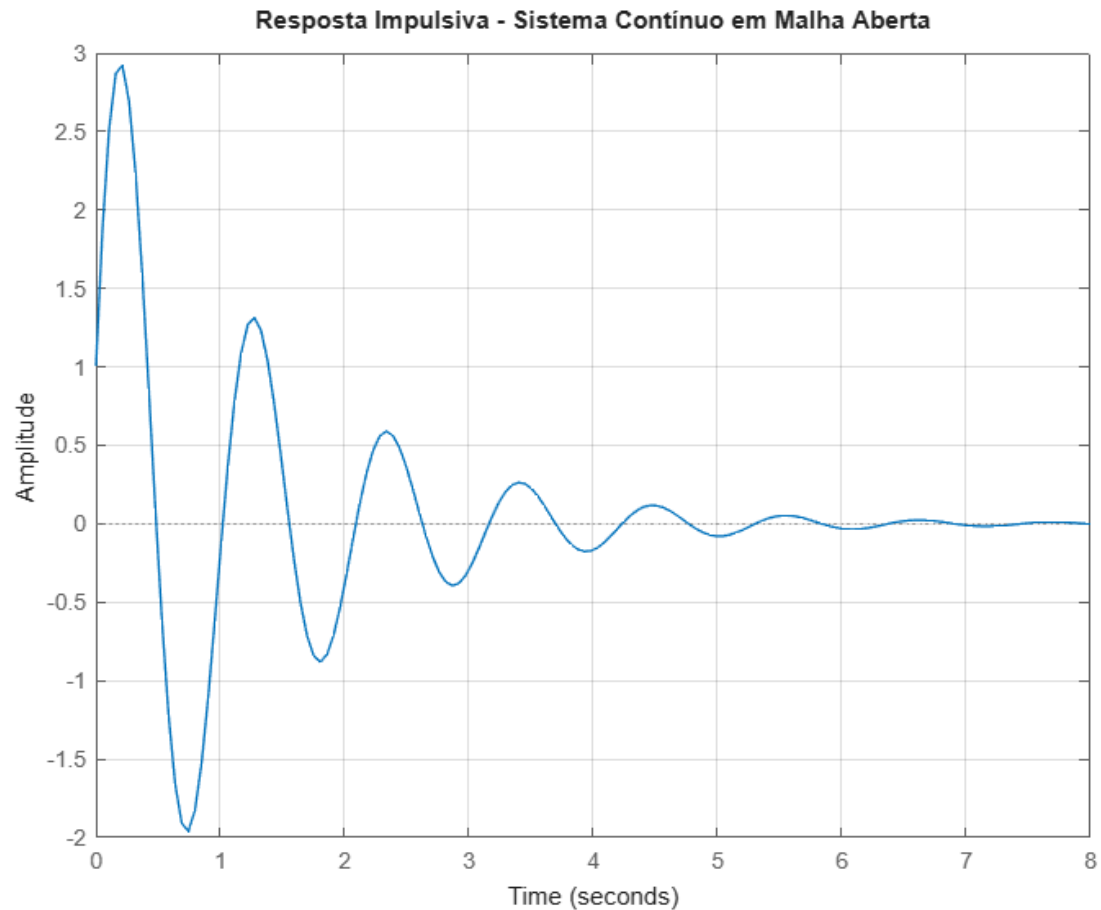


Figura 2: Sistema Contínuo em Malha Aberta

```

1 -----
2 % Definir a funcao de transferencia continua do sistema
3 num = [1 20];
4 den = [1 1.5 35];
5 G_s = tf(num, den);
6 sysd=c2d(G_s,0.01)
7 rlocus(sysd)
8 rltool(sysd)
9 -----

```

Inicialmente foi feito a conversão do sistema contínuo para um sistema discreto, onde por meio do rlocus e rltool verificamos que o tempo de amostragem de 0,1 que foi escolhido , não era possível "juntar" os polos no rltool, portanto usamos uma frequência maior com o tempo de amostragem de 0,01.

Foi calculado o ω_n e ξ a partir das especificações dos projeto

$$Ts = \frac{2,4}{\omega_n} \rightarrow 0.2 = \frac{2,4}{\omega_n} \rightarrow \omega_n \geq 12 \quad (2)$$

$$Te = \frac{4,6}{\xi \omega_n} \rightarrow 0.4 = \frac{4,6}{\xi 12} \rightarrow \xi \geq 0,9383 \quad (3)$$

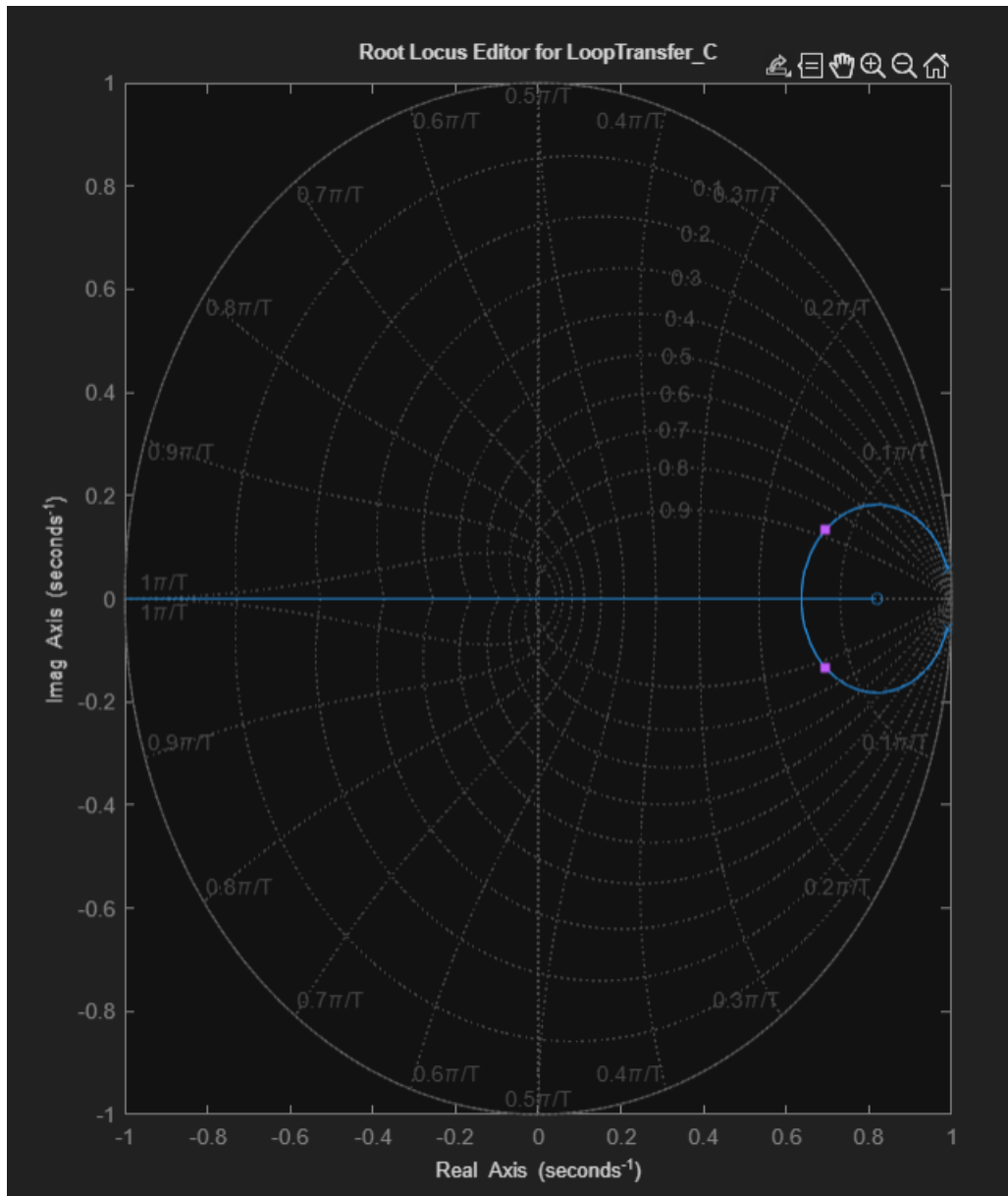


Figura 3: Rltool

Os polos foram colocados em pontos 0.94 aproximadamente onde satisfaz as especificações do projeto com um overshoot próximo de 4 e o ganho obtido foi de 54,4.

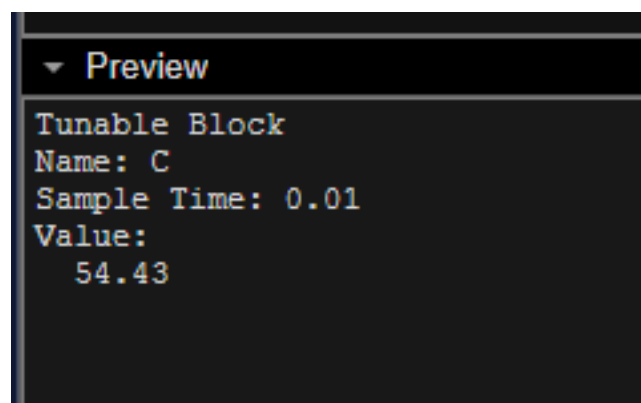


Figura 4: Ganho k

Com o valor de ganho definido foi plotado o grafico root locus, Comparando o gráfico com os valores obtidos percebe-se que o ganho aplicado ao sistema colocou realmente as raízes nos lugares corretos.

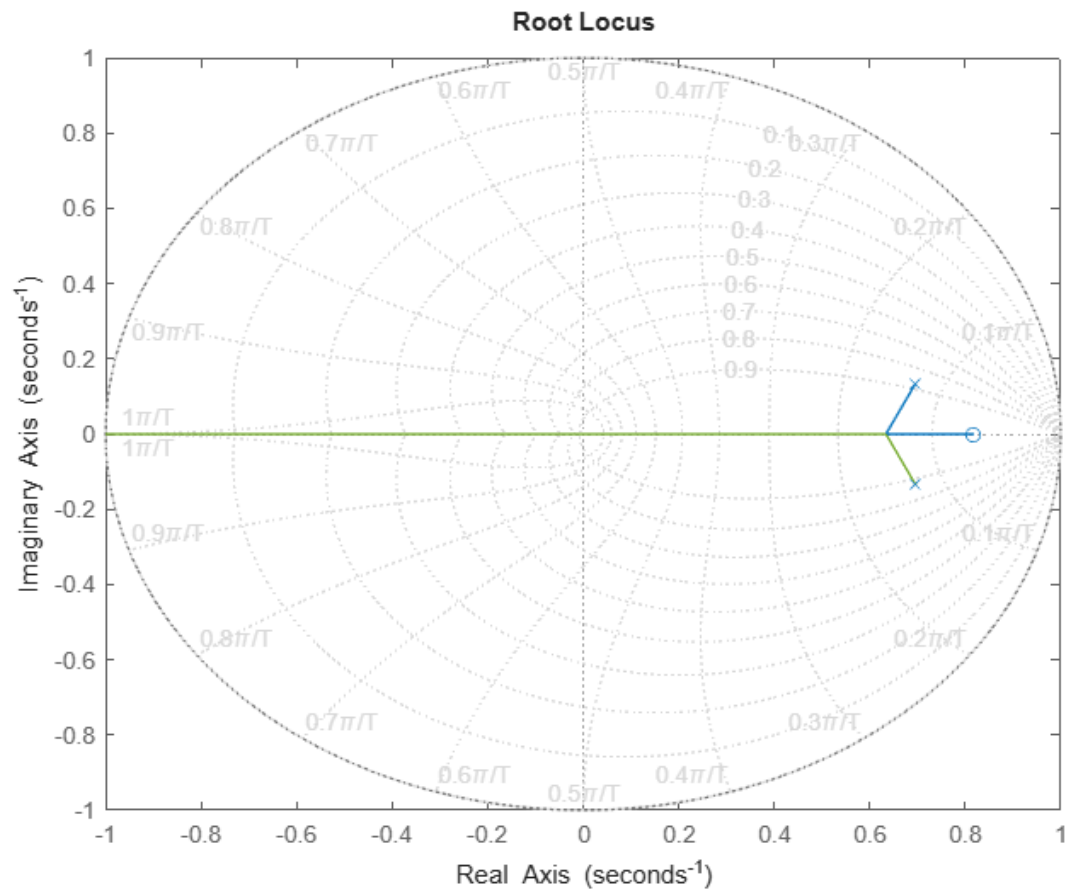


Figura 5: Rlocus

Em seguida, foi feita a implementação no Simulink por meio dos blocos com o intuito de representar o sistema na forma continua e discreta e poder compara-las de forma eficiente.

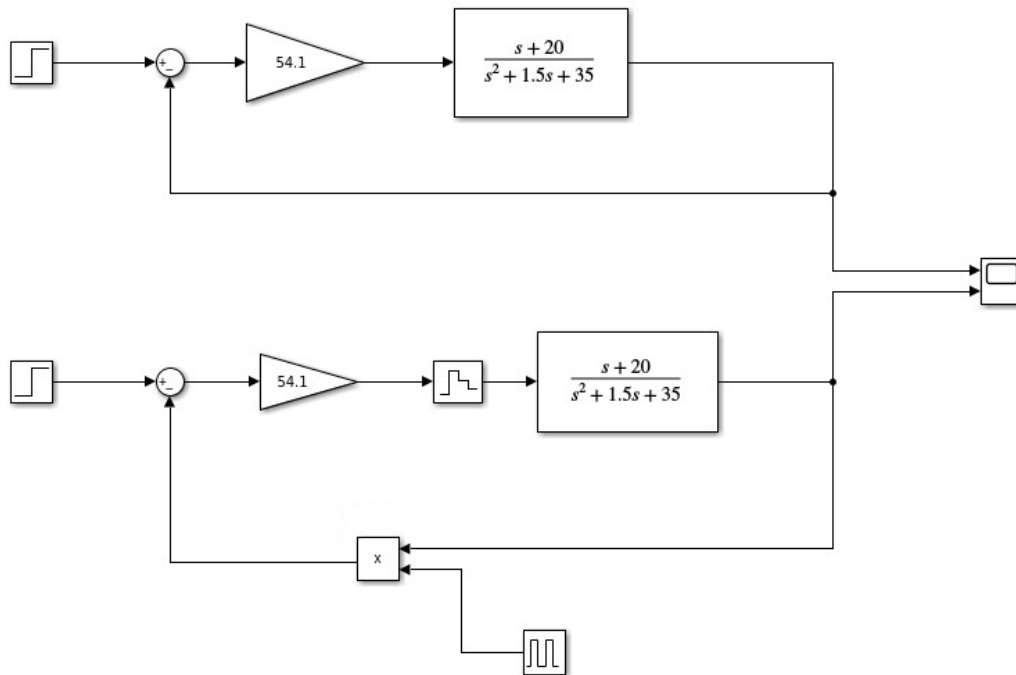


Figura 6: Diagrama de Blocos no Simulink

Com base nisso, foi criado um gráfico que mostra a resposta dos dois sistemas a uma entrada em degrau, utilizando um tempo de amostragem de 0,01.

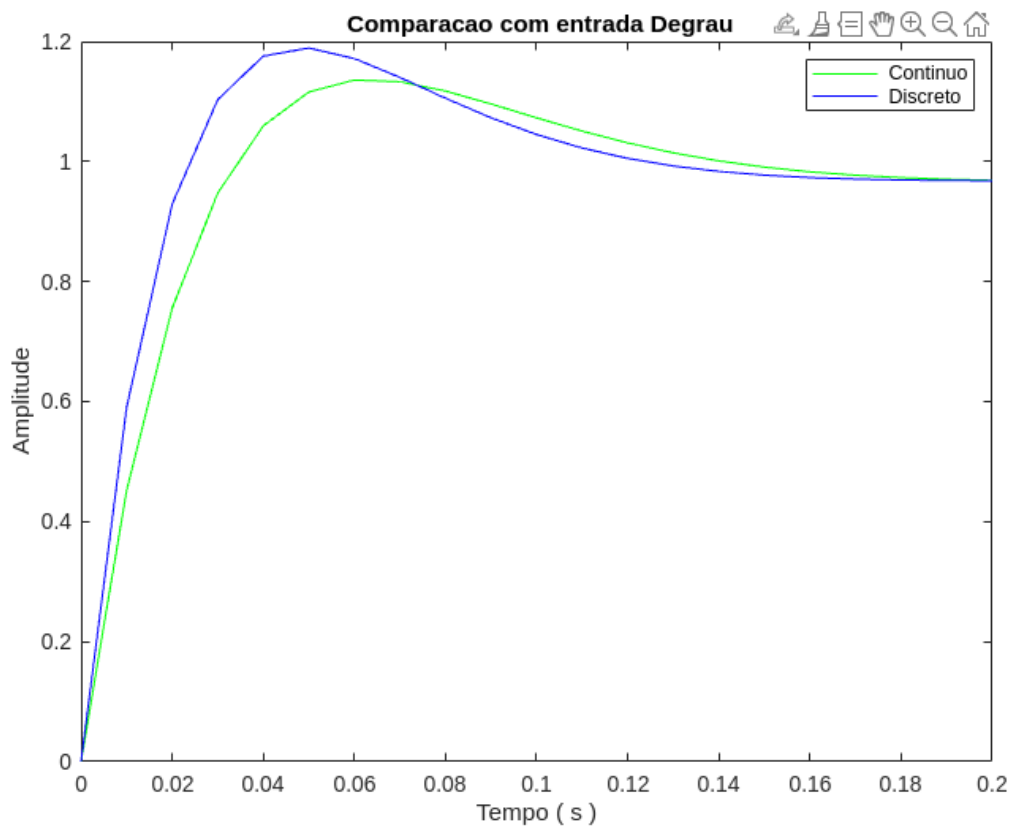


Figura 7: Saída da Entrada Degrau com tempo de amostragem 0,01

Ao aumentar a frequência e considerar um tempo de amostragem de 0,001, observa-se uma maior aproximação do sistema contínuo.

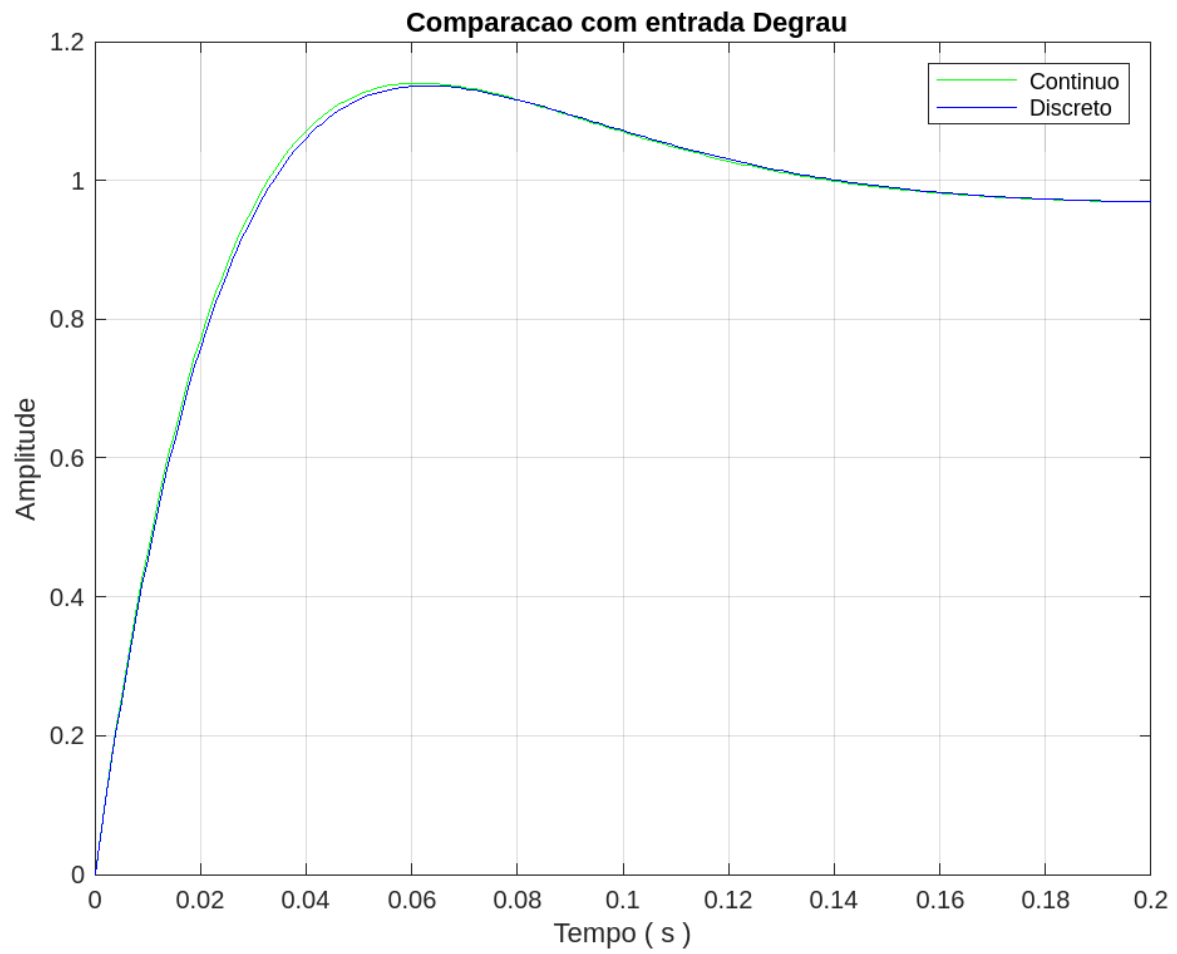


Figura 8: Saída da Entrada Degrau com tempo de amostragem 0,001

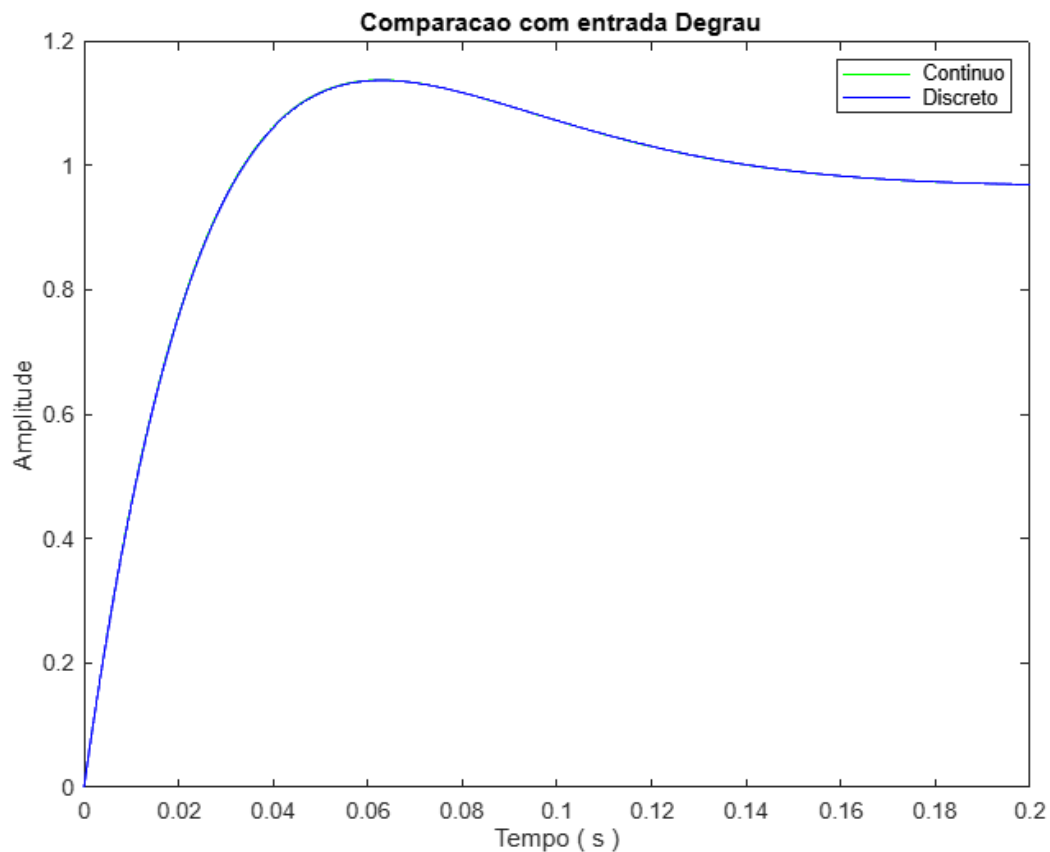


Figura 9: Saída da Entrada Degrau com tempo de amostragem 0,0001

Caso diminuir ainda mais o tempo de amostragem, percebe-se que fica ainda mais próximo do sistema contínuo. Dessa forma, o sistema passa a emular o comportamento do sistema contínuo, assumindo suas mesmas propriedades e características.

```

1 Ts=0.001;
2 % % Implementacao o ganho obtido para entrada degrau
3 t = 0:Ts:0.2;
4 sys_mf = feedback (54.4* G_s , 1) ;
5 sysd_mf = feedback (54.4* sysd , 1) ;
6 [ G_s_step , t ] = step ( sysd_mf , t ) ;
7 [ sysd_step , t ] = step ( sys_mf , t ) ;
8
9 % % Plotar para entrada degrau
10 figure
11 plot (t , G_s_step , 'g' , t , sysd_step , 'b') ;
12 title ( ' Comparacao com entrada Degrau' ) ;
13 xlabel ( ' Tempo ( s ) ' ) ;
14 ylabel ( ' Amplitude ' ) ;
15 legend ( ' Continuo ' , ' Discreto ' ) ;

```

3 Conclusão

Pode-se concluir que a modelagem do sistema contínuo para a forma discreta foi realizada com sucesso, atendendo a todos os requisitos especificados, onde a escolha correta do tempo de amostragem se mostrou bem importante. Com um tempo de amostragem de 0,01s, o sistema cumpre todos os requisitos, embora não seja exatamente igual ao sistema contínuo. Para obter uma semelhança total com o sistema contínuo, o tempo de amostragem deve ser de pelo menos 0,001s, permitindo que o sistema discreto emule o sistema contínuo. Por exemplo, ao aplicar uma entrada degrau com um tempo de amostragem de 0,0001s, as respostas dos dois sistemas são praticamente idênticas, tornando praticamente imperceptível a distinção entre a resposta discreta e a contínua.