

Experiência 2 - Fundamentos de Realimentação - Controle em Malha Aberta e Fechada (Proporcional)

Rodrigo Seiji Piubeli Hirao (186837)

20 de abril de 2021

Conteúdo

1	Preliminares para a Simulação do Modelo	2
2	Função de transferência do problema do regulador	2
3	Iniciando Matlab	2
3.a	2
3.b	3
3.c	3
3.d	4
3.e	4
4	Erro em malha aberta	5
5	Erro em malha fechada	6
6	Simulação	6
7	Simulação com diferentes valores de k_p	12
8	Discussão	20

1 Preliminares para a Simulação do Modelo

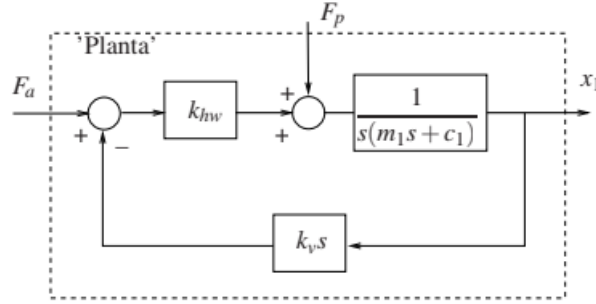


Figura 1: Planta compensada com simulação de atrito viscoso adicional. F_a : força de acionamento (entrada servo); F_p : força de perturbação (entrada de perturbação).

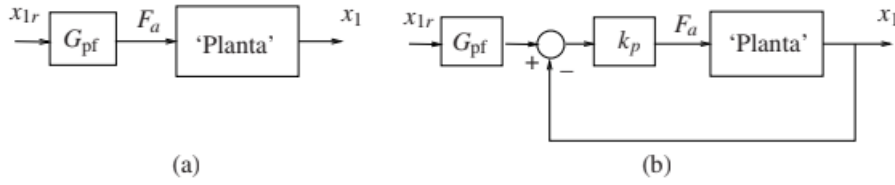


Figura 2: a) Controle em malha aberta; b) Controle em malha fechada do tipo proporcional. x_{1r} : entrada de referência, G_{pf} : função de transferência do pré-filtro, k_p : ganho do controlador proporcional.

A função de transferência da planta compensada com simulação de atrito viscoso adicional da Figura 1 para o problema do servo pode ser calculada como:

$$X_1 = \frac{1}{s(m_1s + c_1)} k_{hw} (F_a - k_v s X_1)$$

$$X_1 m_1 s^2 + X_1 c_1 s + k_{hw} k_v s X_1 = k_{hw} F_a$$

$$\frac{X_1(s)}{F_a(s)} = G'_{ps}(s) = \frac{k_{hw}}{m_1 s^2 + (c_1 + k_{hw} k_v) s}$$

2 Função de transferência do problema do regulador

Já para o problema do regulador será:

$$X_1 = \frac{1}{s(m_1s + c_1)} (F_p - k_{hw} k_v s X_1)$$

$$X_1 m_1 s^2 + X_1 c_1 s + k_{hw} k_v s X_1 = F_p$$

$$\frac{X_1(s)}{F_p(s)} = G'_{pr}(s) = \frac{1}{m_1 s^2 + (c_1 + k_{hw} k_v) s}$$

3 Iniciando Matlab

3.a

Colocando a função G'_{ps} no matlab e calculando seu ganho DC, ou seja, $G'_{ps}(0)$ obtemos:

```
>> G_ps = tf ([k_hw], [m_1 c_1 + k_hw * k_v 0])
```

```
G_ps =
```

$$\frac{2088}{0.2376 s^2 + 21.11 s}$$

```
Continuous-time transfer function.
```

```
>> dcgain(G_ps)
```

```
ans =
```

```
Inf
```

E como temos que:

$$e_r = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s(F_a(s)G'_{p_s}(s) - F_a(s)) = \lim_{s \rightarrow 0} sF_a(s)(G'_{p_s}(s) - 1)$$

Temos que o erro é infinito para uma entrada unitária pois:

$$e_r = \lim_{s \rightarrow 0} s \frac{1}{s} (G'_{p_s}(s) - 1) = \lim_{s \rightarrow 0} G'_{p_s}(s) - 1 = G'_{p_s}(0) - 1 = \infty$$

3.b

E para G'_{p_r} temos o mesmo ganho em regime:

```
>> G_pr = tf ([1], [m_1 c_1 + k_hw * k_v 0])
```

```
G_pr =
```

$$\frac{1}{0.2376 s^2 + 21.11 s}$$

```
Continuous-time transfer function.
```

```
>> dcgain(G_pr)
```

```
ans =
```

```
Inf
```

3.c

Usando a função feedback e considerando G_{p_f} como 1, podemos ter a função de transferência da figura 2b:

```
>> H = feedback(k_p * G_ps, 1)
```

```
H =
```

$$\frac{4177}{0.2376 s^2 + 21.11 s + 4177}$$

```
Continuous-time transfer function.
```

3.d

```
>> dcgain(H)
```

```
ans =
```

```
1
```

Como já calculamos o ganho em regime da planta anteriormente temos que os ganhos são ∞ para a malha aberta e 1 para malha fechada.

E utilizando o mesmo procedimento da seção 2a temos que seus erros em regime a uma entrada de degrau unitário é igual ao seu ganho em regime -1, logo ∞ para a malha aberta e 0 para a malha fechada.

3.e

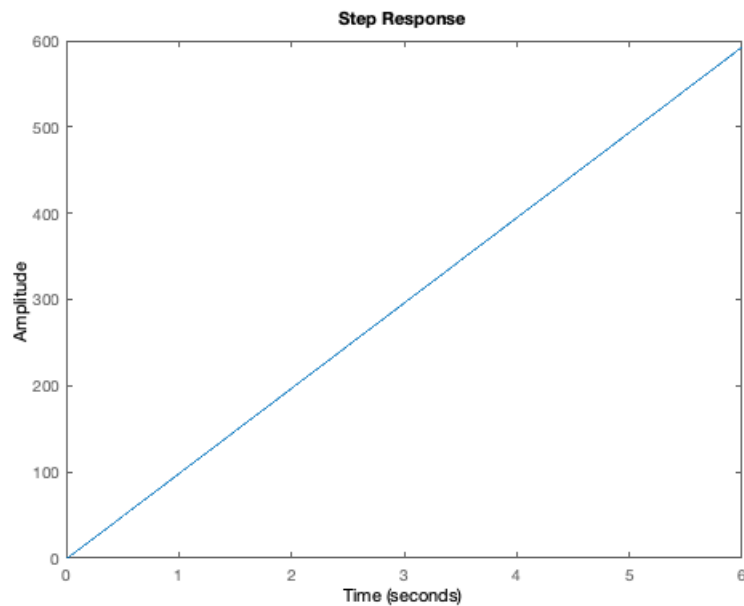


Figura 3: Resposta ao degrau unitário em função do tempo do sistema de malha aberta

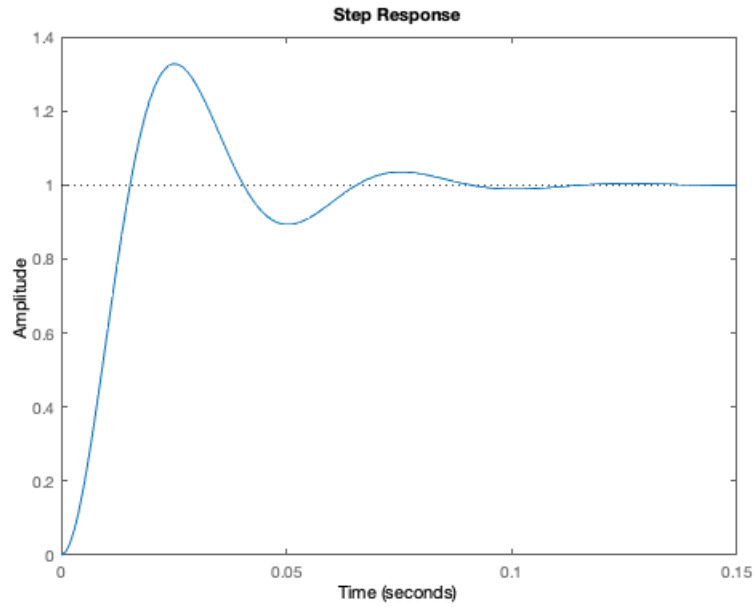


Figura 4: Resposta ao degrau unitário em função do tempo do sistema de malha fechada

4 Erro em malha aberta

Para que o erro o sistema de malha aberta da Figura 2a seja nulo, temos que:

$$\lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} H(s) - 1 = \lim_{s \rightarrow 0} G_{pf}(s)G'_{ps}(s) - 1 = 0$$

$$\lim_{s \rightarrow 0} G_{pf}(s)G'_{ps}(s) = 1$$

$$\lim_{s \rightarrow 0} \frac{k_{hw}}{m_1 s^2 + (c_1 + k_{hw}k_v)s} \frac{k_{pf}s}{1 + 0.01s} = 1$$

$$\lim_{s \rightarrow 0} \frac{k_{hw}}{m_1 s + c_1 + k_{hw}k_v} \frac{k_{pf}}{1 + 0.01s} = 1$$

$$\frac{k_{hw}k_{pf}}{c_1 + k_{hw}k_v} = 1$$

$$k_{pf} = \frac{c_1 + k_{hw}k_v}{k_{hw}} = \frac{c_1}{k_{hw}} + k_v = 0.0101$$

Que, jogando no matlab temos que:

```
>> G_pf = tf([k_pf 0], [0.01 1])
```

G_pf =

```
0.01011 s
-----
0.01 s + 1
```

Continuous-time transfer function.

```
>> H_f = series(G_pf, G_ps)
```

H_f =

```

21.11 s
-----
0.002376 s^3 + 0.4487 s^2 + 21.11 s

Continuous-time transfer function.

>> dcgain(H_f)

ans =

1

```

Com resposta ao degrau:

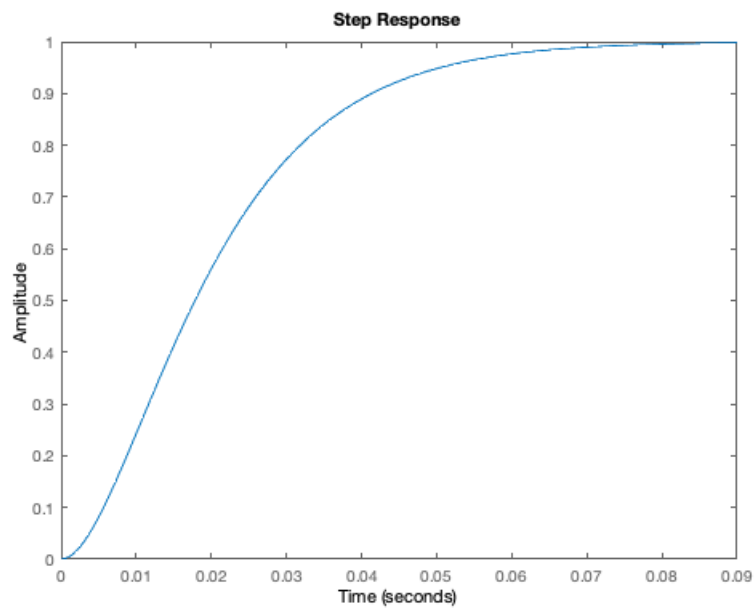


Figura 5: Resposta ao degrau unitário em função do tempo do sistema de malha aberta com pré-filtro

5 Erro em malha fechada

Temos que o erro em malha fechada é:

$$\begin{aligned}
 \lim_{s \rightarrow 0} sE(s) &= \lim_{s \rightarrow 0} G'_{p_s m_f}(s) - 1 \\
 &= \lim_{s \rightarrow 0} \frac{k_{pf} k_p k_{hw}}{m_1 s^2 + (c_1 + k_{hw} k_v) s + k_p k_{hw}} - 1 = \frac{k_{pf} k_p k_{hw}}{k_p k_{hw}} - 1 = k_{pf} - 1
 \end{aligned}$$

Assim podemos ver que o erro depende apenas de k_{pf}

6 Simulação

Foram montados os seguintes modelos no simulink das imagens 6, 7 e 8

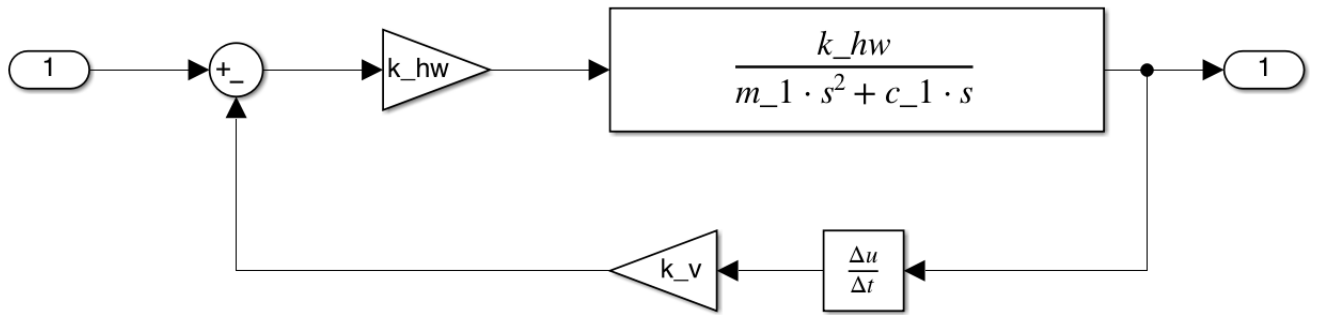


Figura 6: Simulação do sistema com a haste rotacional travada

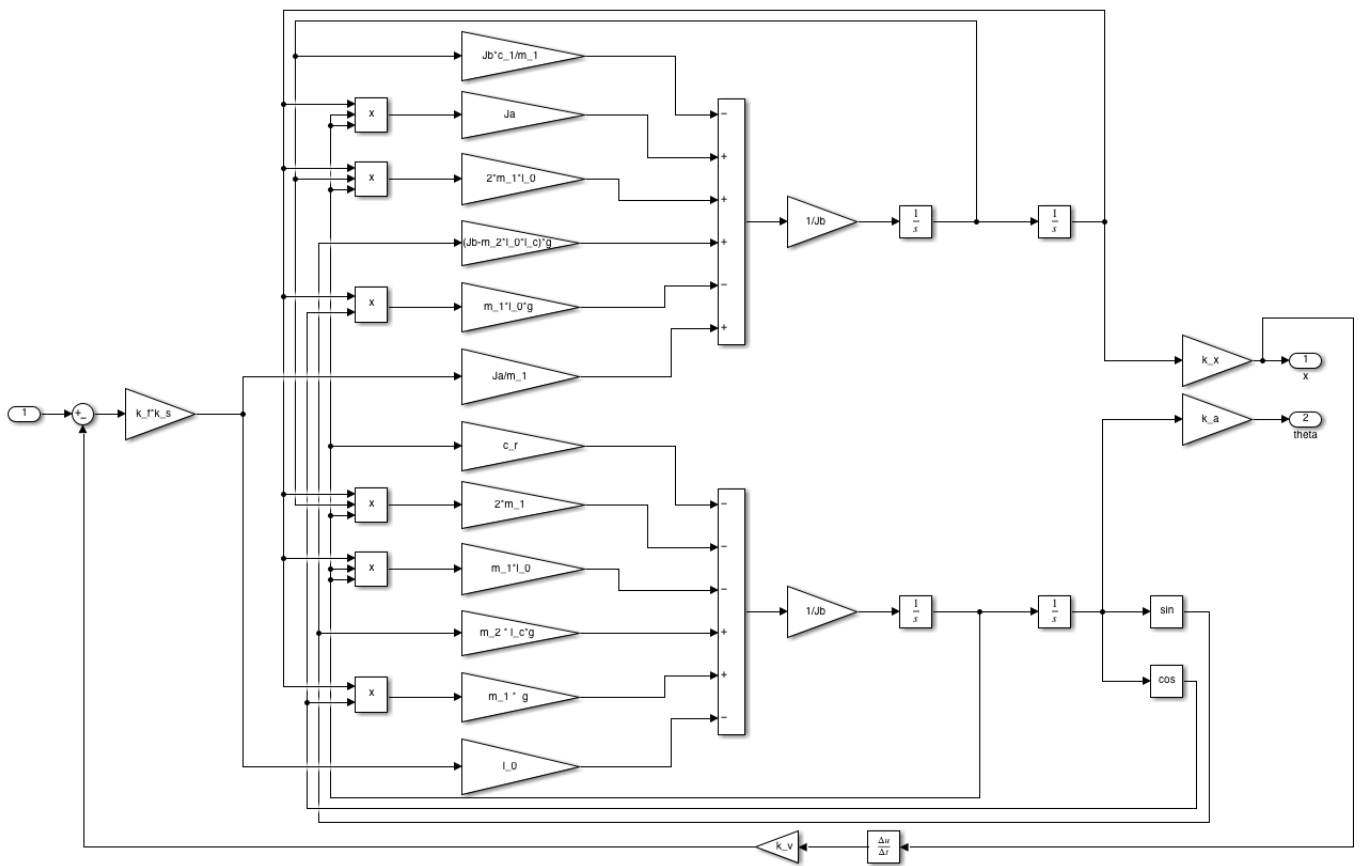


Figura 7: Simulação do sistema não linear

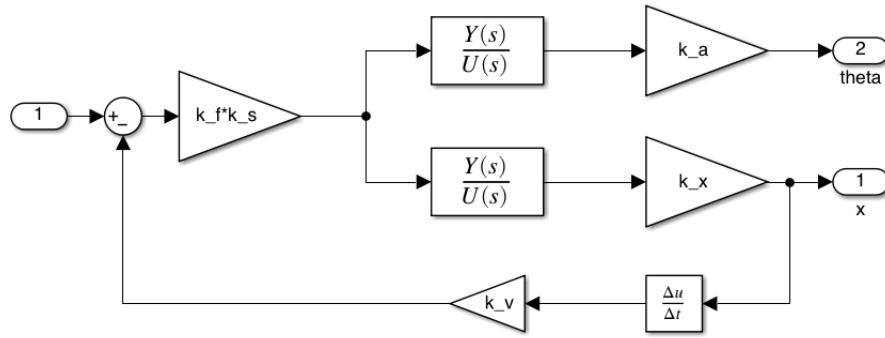


Figura 8: Simulação do sistema linearizado

E usados nos sistemas de malha aberta (Figura 9) e malha fechada (Figura 10)

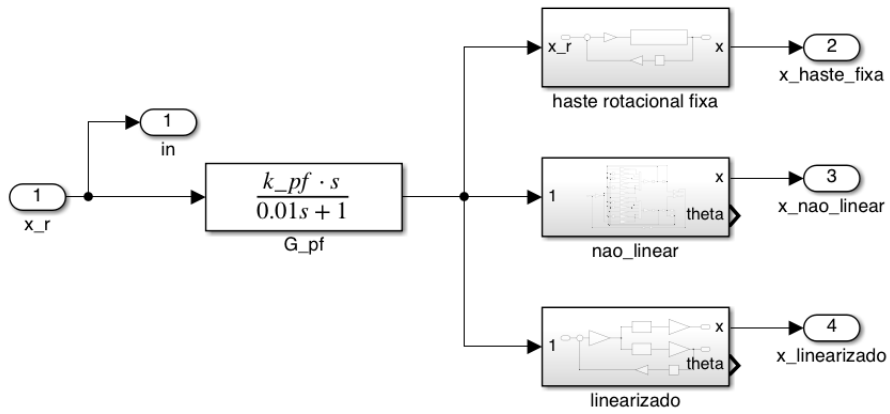


Figura 9: Simulação dos sistemas em malha aberta

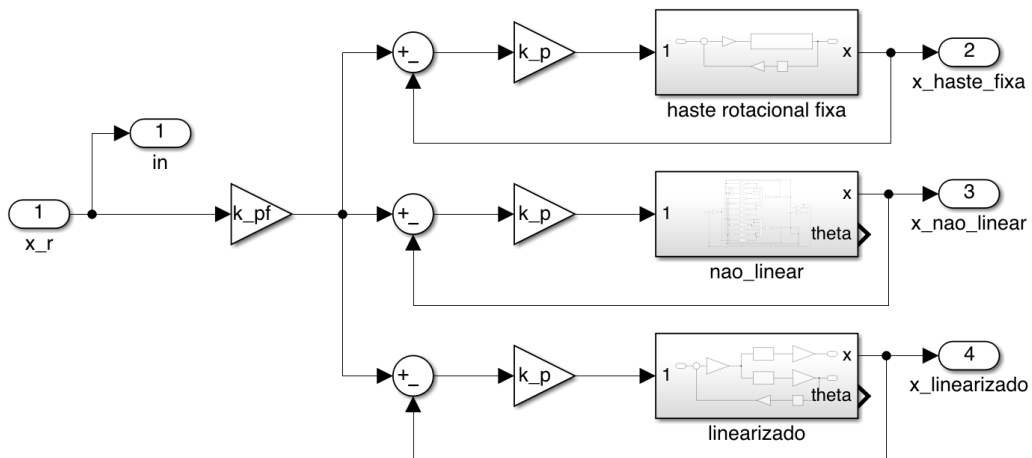


Figura 10: Simulação dos sistemas em malha fechada

E ao simular com os o degrau de 500 [count] até 2s, 0 [count] até 4s, -500 [count] até 6s e 0 [count] no resto com o código (e em seguida com amplitude 2000):


```
>> in = Simulink.SimulationInput('lab02_1').setExternalInput([t u500])
```

```
in =
```

```
SimulationInput with properties:
```

```

    ModelName: 'lab02_1'
   InitialState: [0×0 Simulink.op.ModelOperatingPoint]
  ExternalInput: [10001×2 double]
 ModelParameters: [0×0 Simulink.Simulation.ModelParameter]
BlockParameters: [0×0 Simulink.Simulation.BlockParameter]
    Variables: [0×0 Simulink.Simulation.Variable]
    PreSimFcn: []
    PostSimFcn: []
    UserString: ''

```

```
>> plot(sim(in))
```

Foi obtido os resultados da imagem 11 e 12 para a malha aberta e 13 e 14 para a malha fechada

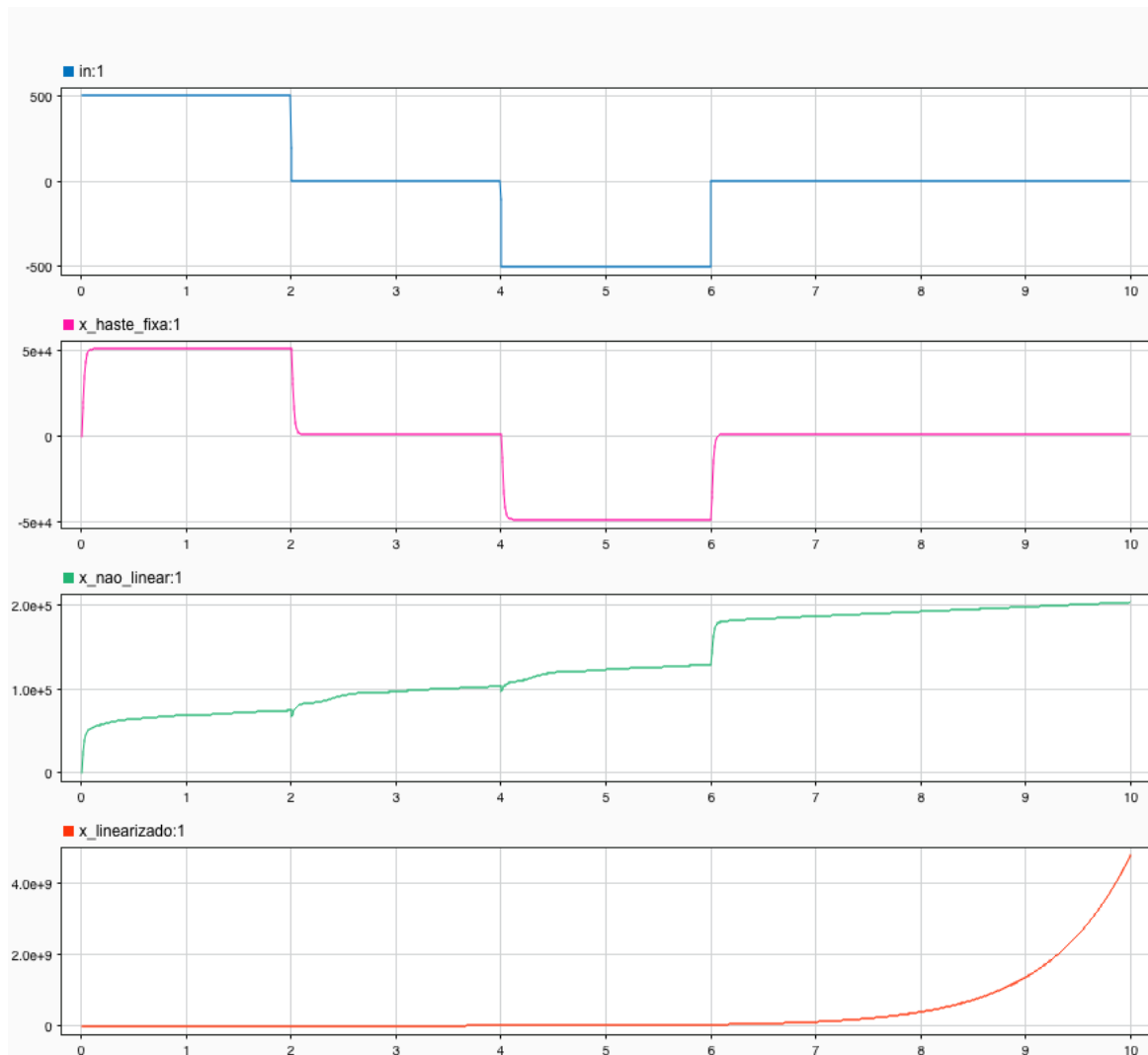


Figura 11: Simulação dos sistemas em malha aberta para degrau com amplitude 500

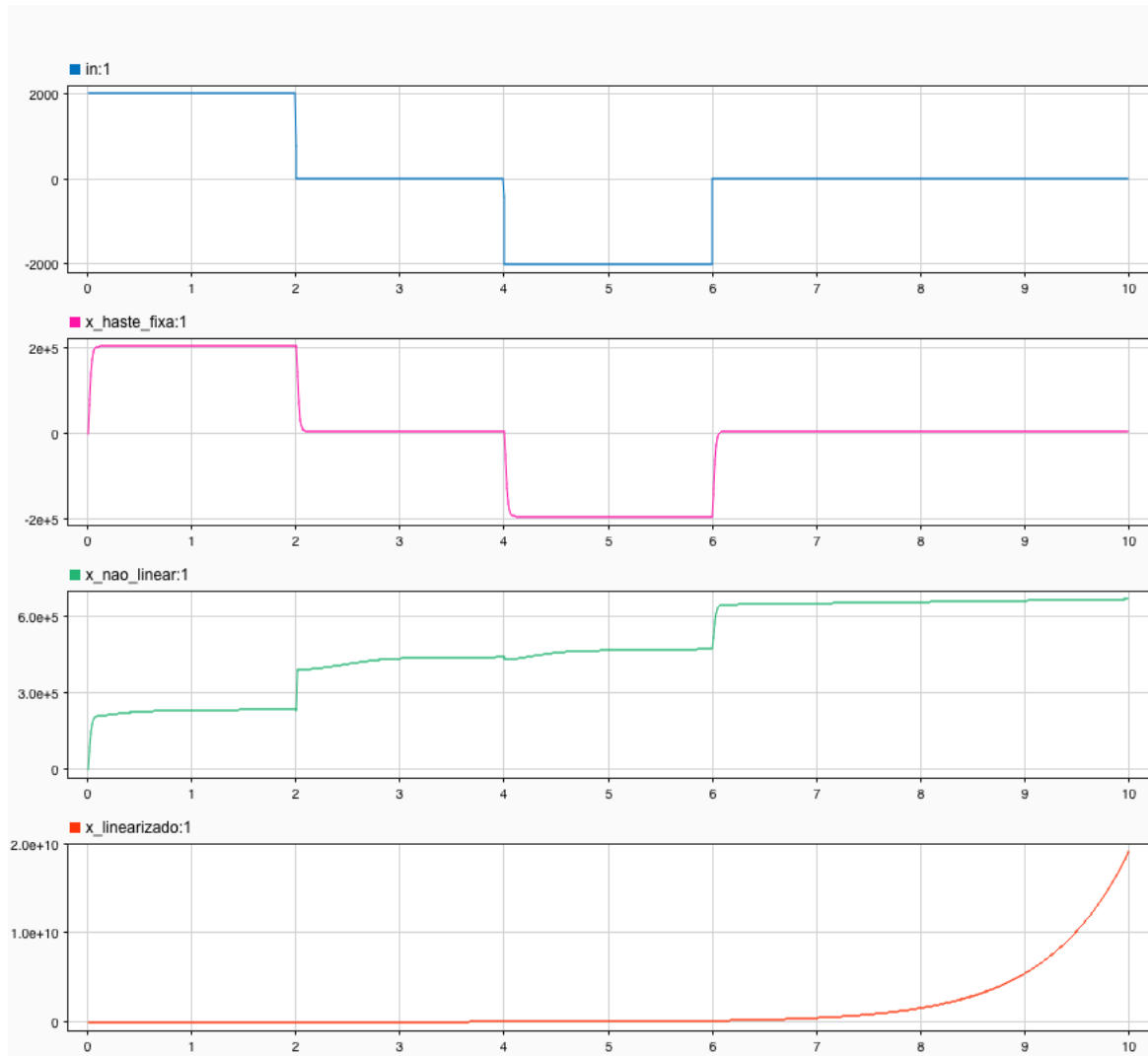


Figura 12: Simulação dos sistemas em malha aberta para degrau com amplitude 2000

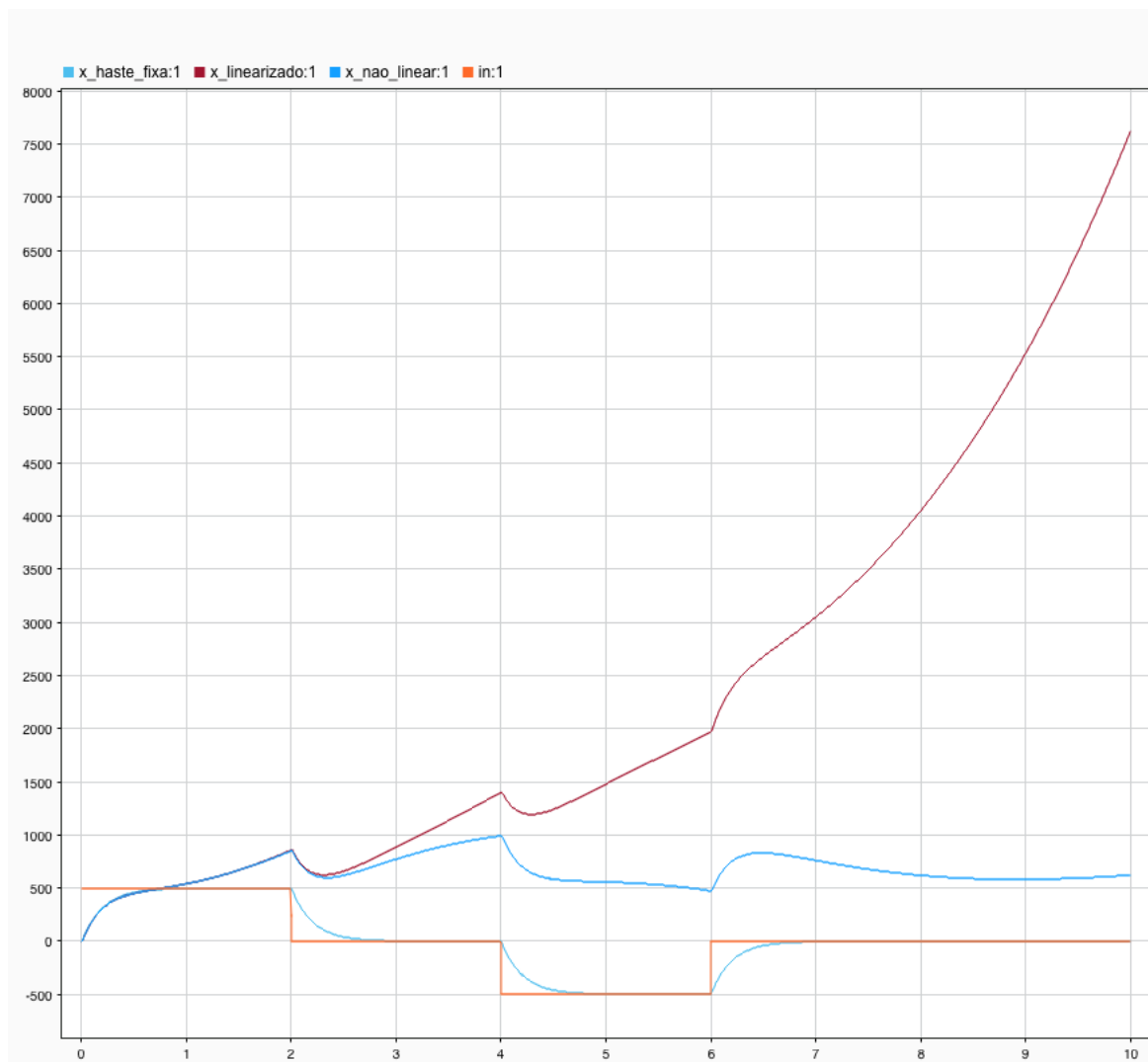


Figura 13: Simulação dos sistemas em malha aberta para degrau com amplitude 500

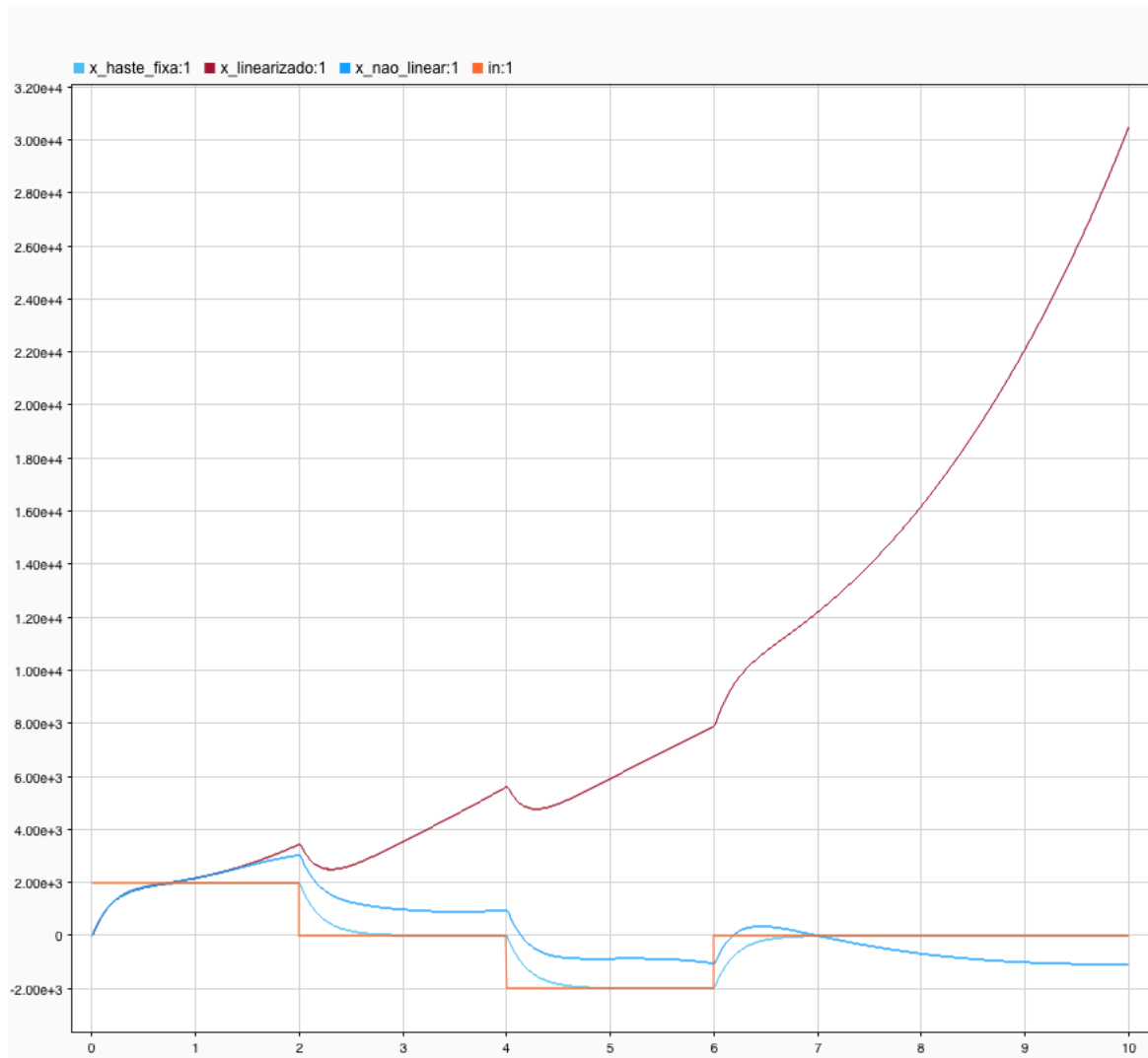


Figura 14: Simulação dos sistemas em malha aberta para degrau com amplitude 2000

7 Simulação com diferentes valores de k_p

Foram feitos diferentes testes para o sistema de malha fechada com o k_p tendo os valores de 0.1, 0.2, 0.4 e 0.8:

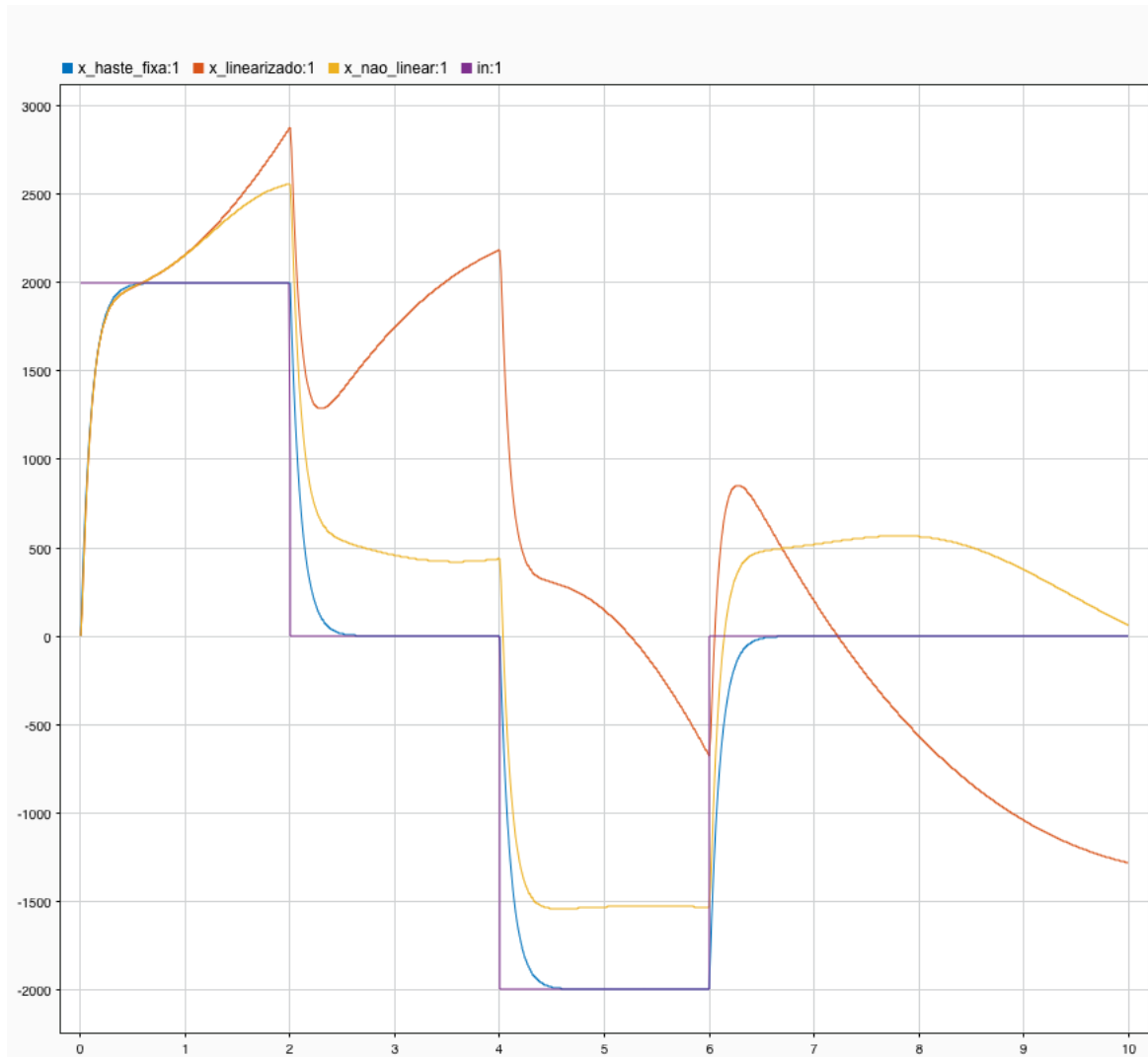


Figura 15: Simulação dos sistemas em malha fechada para degrau com amplitude 500 e $k_p = 0.1$

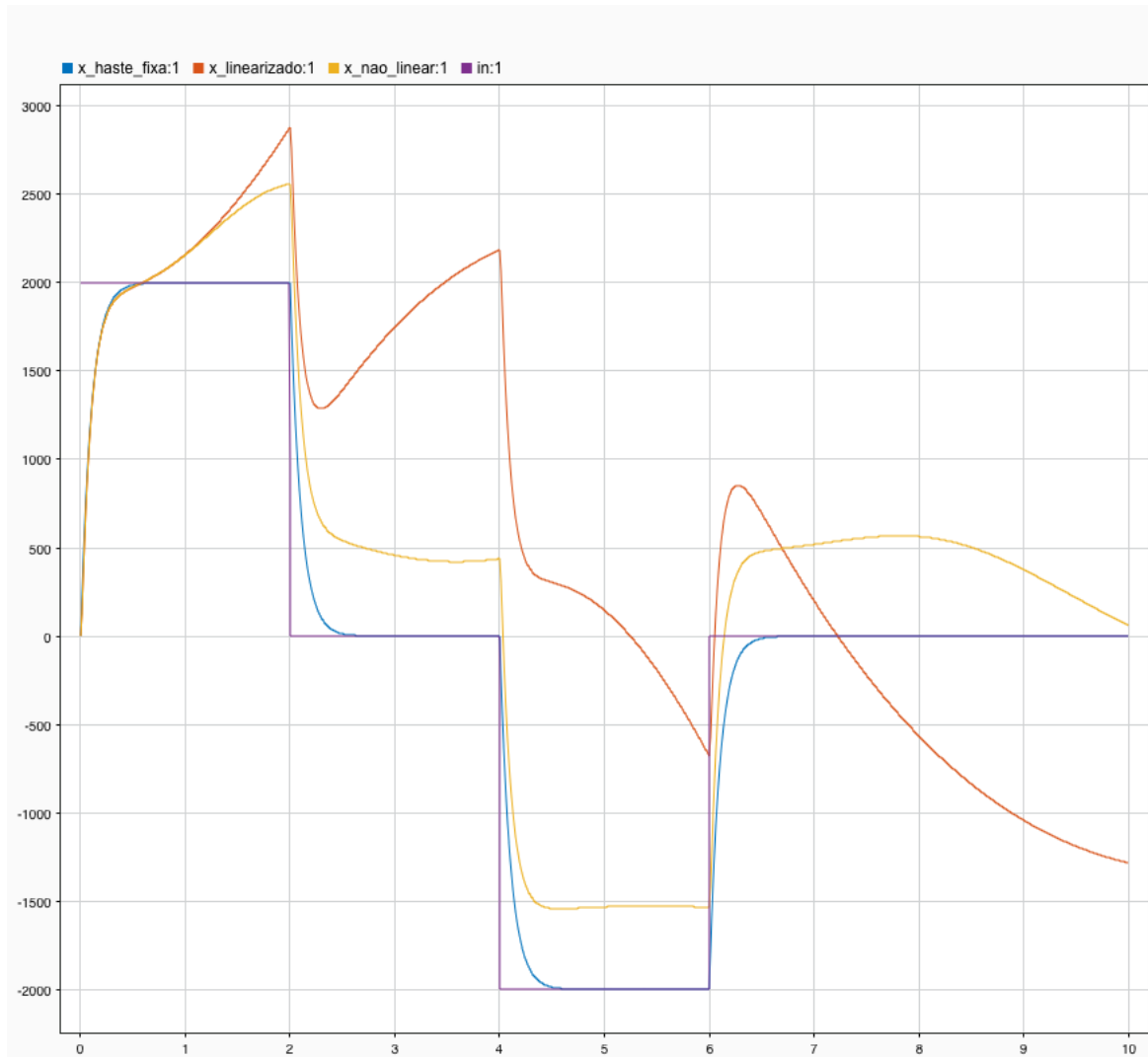


Figura 16: Simulação dos sistemas em malha fechada para degrau com amplitude 2000 e $k_p = 0.1$

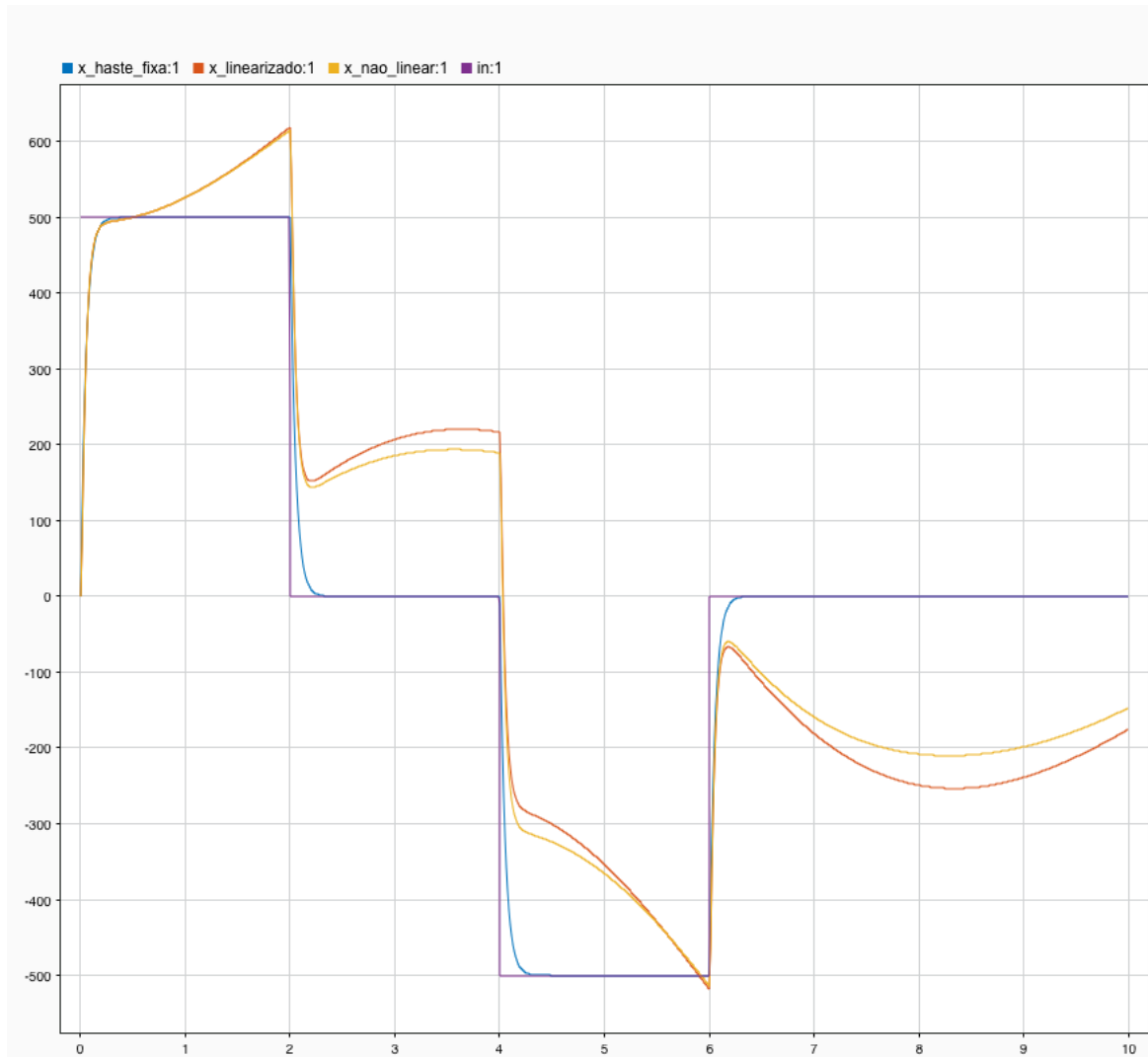


Figura 17: Simulação dos sistemas em malha fechada para degrau com amplitude 500 e $k_p = 0.2$

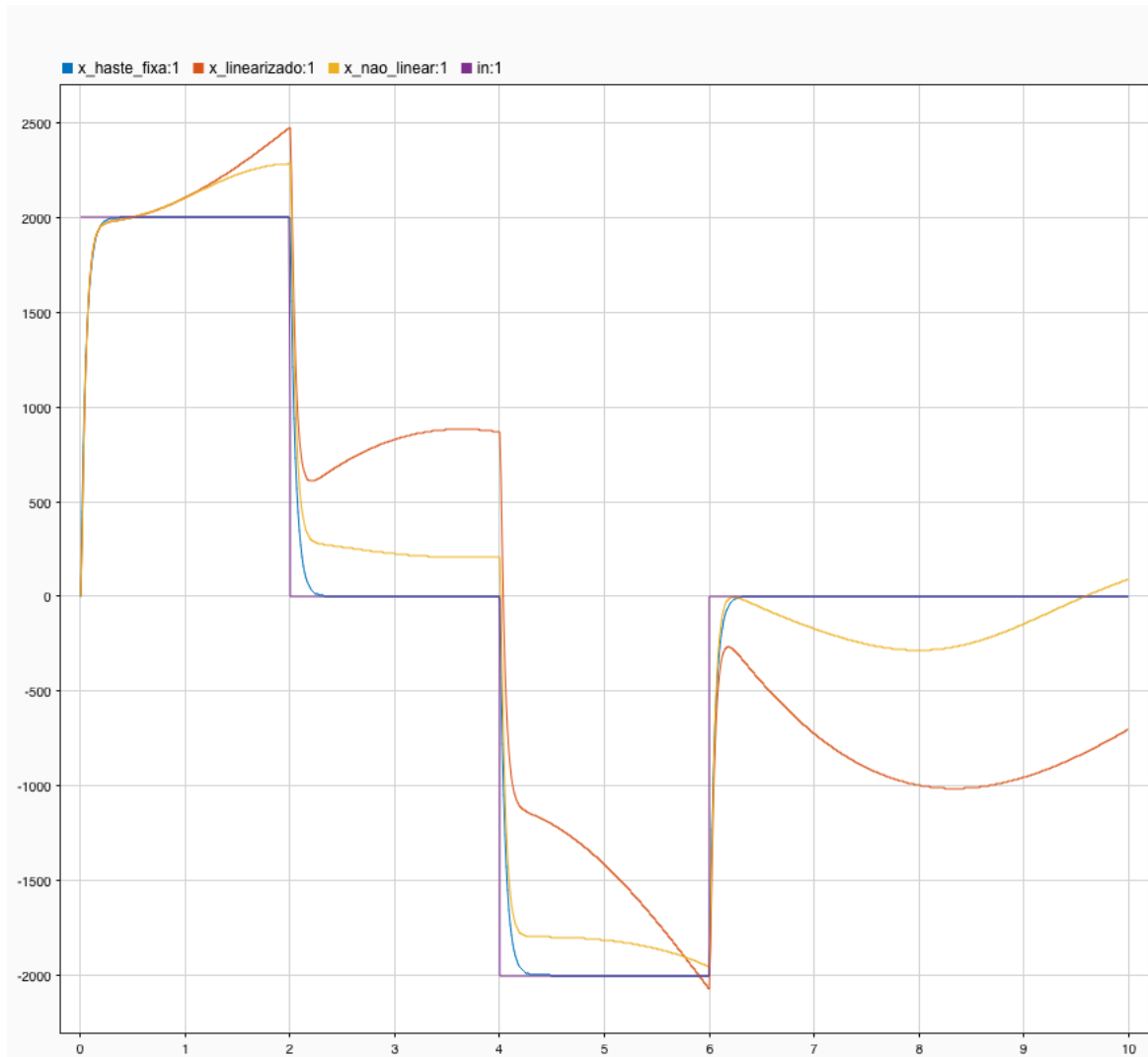


Figura 18: Simulação dos sistemas em malha fechada para degrau com amplitude 2000 e $k_p = 0.2$

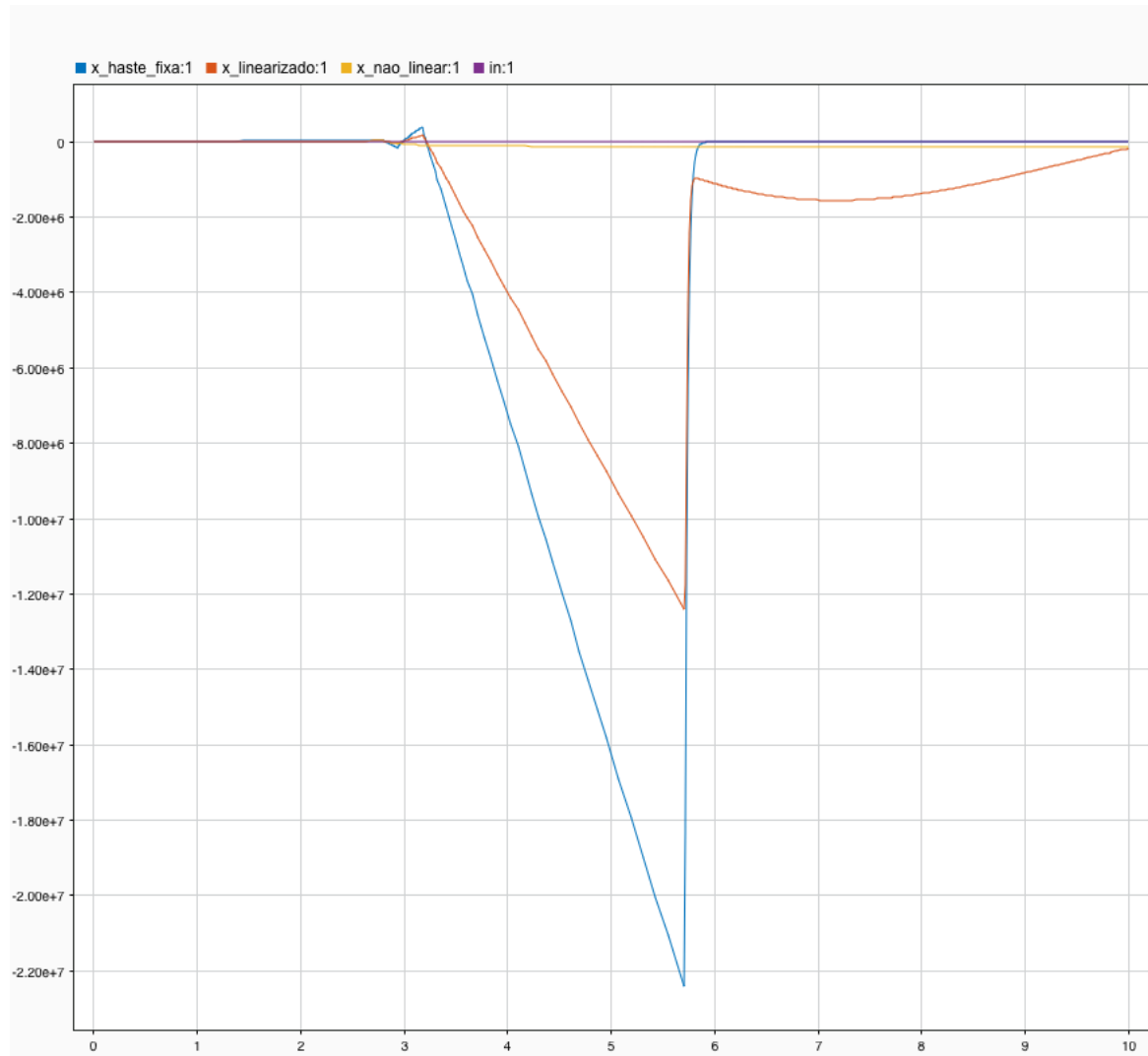


Figura 19: Simulação dos sistemas em malha fechada para degrau com amplitude 500 e $k_p = 0.4$

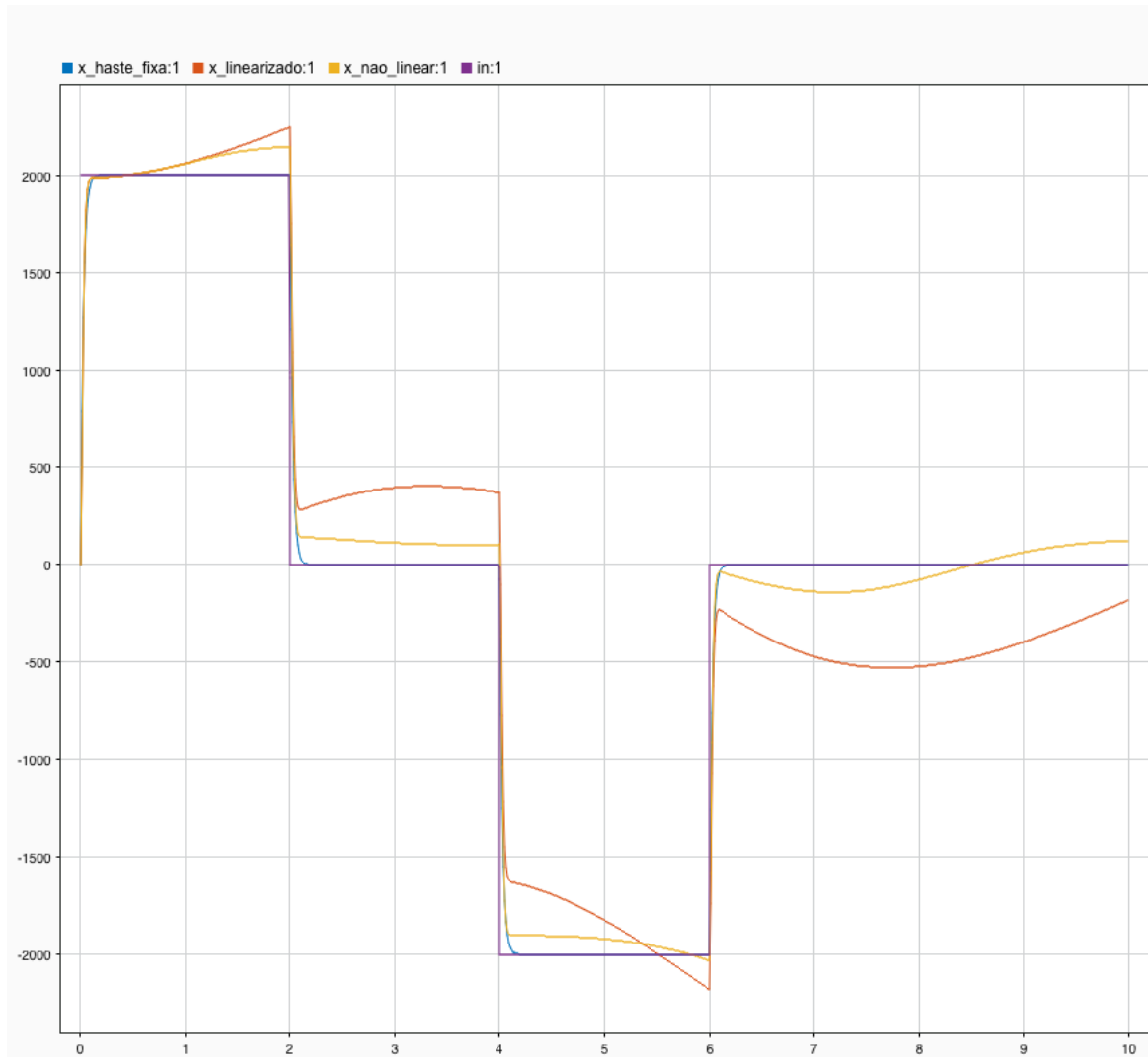


Figura 20: Simulação dos sistemas em malha fechada para degrau com amplitude 500 e $k_p = 0.4$

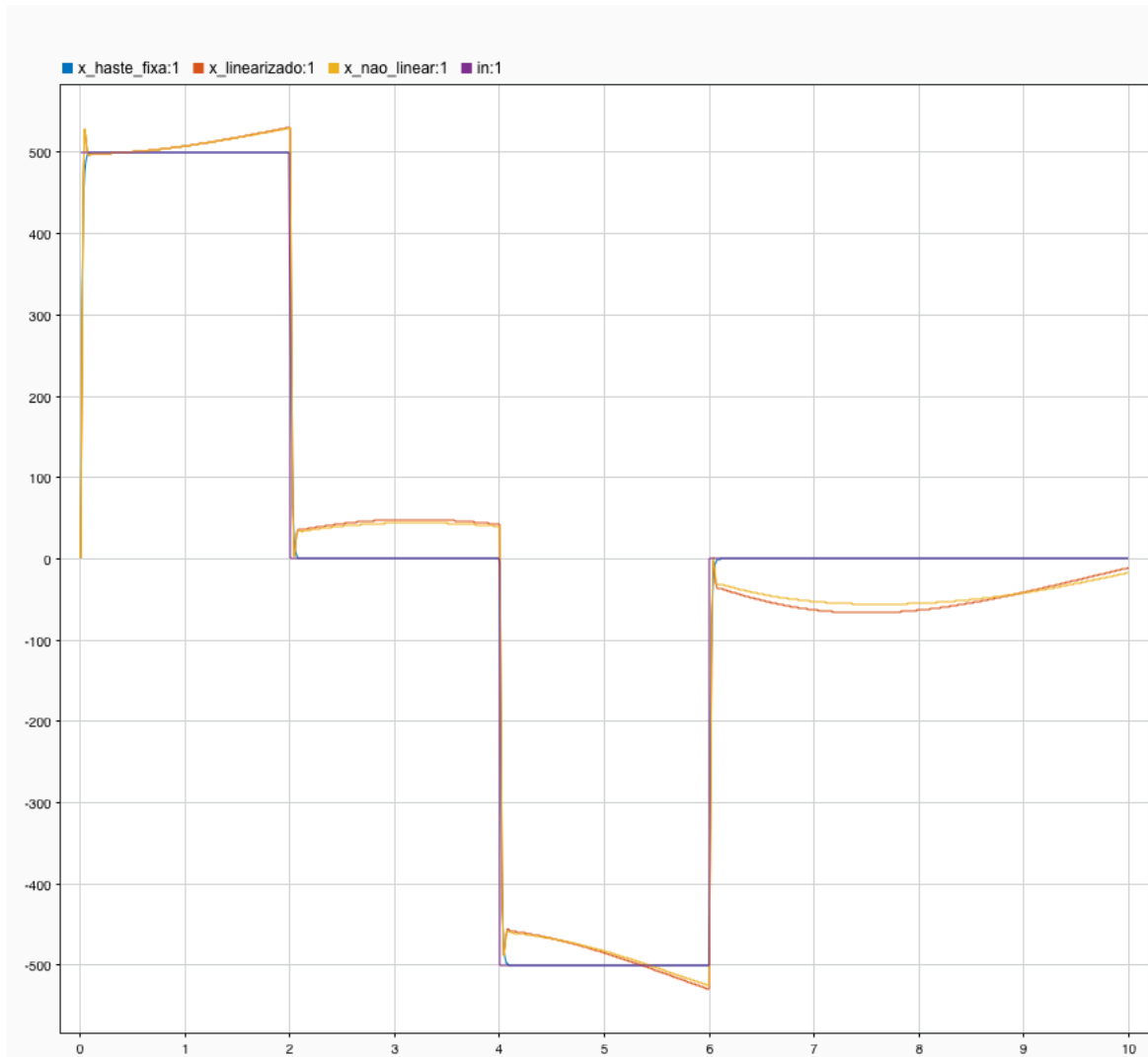


Figura 21: Simulação dos sistemas em malha fechada para degrau com amplitude 500 e $k_p = 0.8$

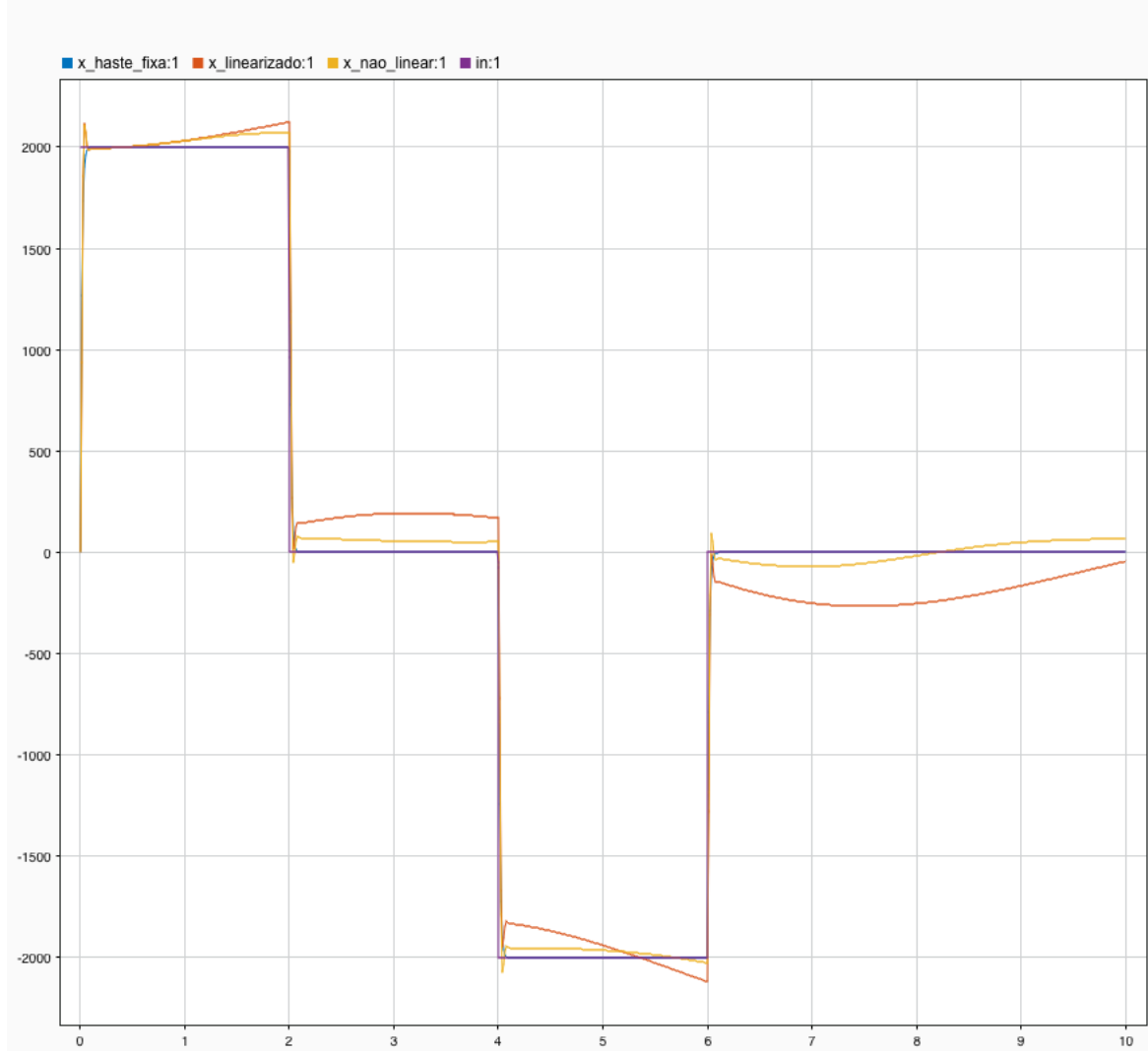


Figura 22: Simulação dos sistemas em malha fechada para degrau com amplitude 2000 e $k_p = 0.8$

8 Discussão

Pelos gráficos pode-se ver que as respostas em malha aberta com a haste rotacional livre não tem erros nulos, provavelmente pelo fato dele usar uma fórmula criada para um sistema com ângulo fixo em 0. Pode-se ver que o sistema de malha aberta não consegue lidar tão bem com as perturbações quanto o sistema de malha aberta tendo que o maior erro e regime presente foi o do sistema linearizado, que começa a divergir muito a partir do momento que o ângulo se afasta mais de 0, adotando um comportamento exponencial.

Como foi visto anteriormente na sessão 5, o erro da malha fechada depende apenas do parâmetro k_{pf} , mas o parâmetro k_p cuida do quão rápido o sistema entra em regime, podendo causar overshoot, porém também diminuído o erro em regime para o caso testado. Além de poder ser observado um comportamento estranho quando o degrau tem amplitude 500 e $k_p = 0.2$.

O erro do sistema não linear não é nulo pois ele não considera seu ângulo em sua realimentação, o que faz com que ele funcione melhor em ângulos pequenos apenas.