

1. (3 pontos) Para a questão do pêndulo invertido da Parte 1, simular o controlador *sliding mode* utilizando os sinais de controle encontrados (usando a lei descontínua *sign* e a proposta para substituí-la). Tome como condição inicial  $x_1 = 0.1$  e insira uma perturbação externa em  $t = 5$  s por meio de um impulso aplicado sinal de controle. Os valores das variáveis são:  $M = 1$  kg,  $m = 0.1$  kg e  $l = 0.5$  m. Para uma referência  $x_d = A \sin(t)$  com  $\frac{\pi}{50} \leq A \leq \frac{\pi}{30}$ , pede-se:

a) Em um mesmo gráfico, plotar a saída  $\theta$  para os dois sinais de controle.

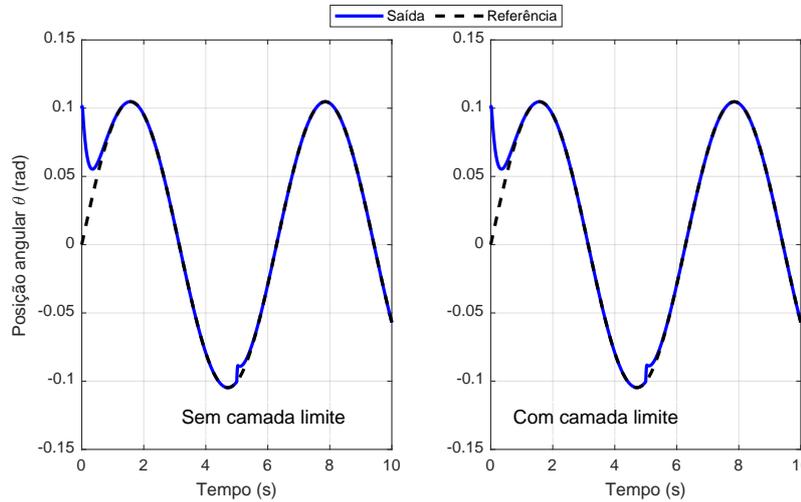


Figura 1: Posição angular do pêndulo invertido para  $A = \frac{\pi}{30}$ .

Fonte: Gerada utilizando MATLAB.

b) Em um mesmo gráfico, plotar os sinais de controle  $u$ .

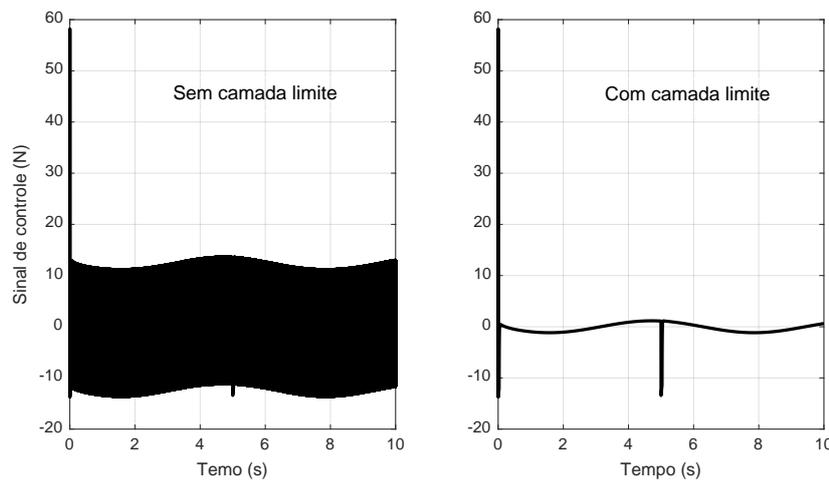


Figura 2: Sinal de controle do pêndulo invertido para  $A = \frac{\pi}{30}$ .

Fonte: Gerada utilizando MATLAB.

c) Discutir os resultados quanto à precisão do controlador e características do sinal de controle.

É possível perceber que, em relação ao desempenho do controlador, o uso da camada limite não trouxe uma piora significativa, uma vez que a largura utilizada foi escolhida pequena. Contudo, para valores maiores, o desempenho pode ser piorado. Além disso, o uso da

camada limite torna o sinal de controle mais suave, praticamente eliminando as oscilações de alta frequência.

**OBS:** Um arquivo do SIMULINK (*penduloinvertido.rar*) com a base da simulação está disponível no e-disciplinas, no tópico "Avaliação".

2. (3 pontos) Considere o sistema exibido na questão 4 da Parte 1:

$$\begin{cases} \dot{x}_1 = x_1(x_1^2 + x_2^2 - 2) - 4x_1x_2^2 \\ \dot{x}_2 = 4x_1^2x_2 + x_2(x_1^2 + x_2^2 - 2) \end{cases} \quad (1)$$

a) Utilizando a função *ppplane8*, plote o plano de fase exibindo todos os pontos de equilíbrio.

O plano de fase utilizando a função *ppplane8* pode ser visto na Figura 3.

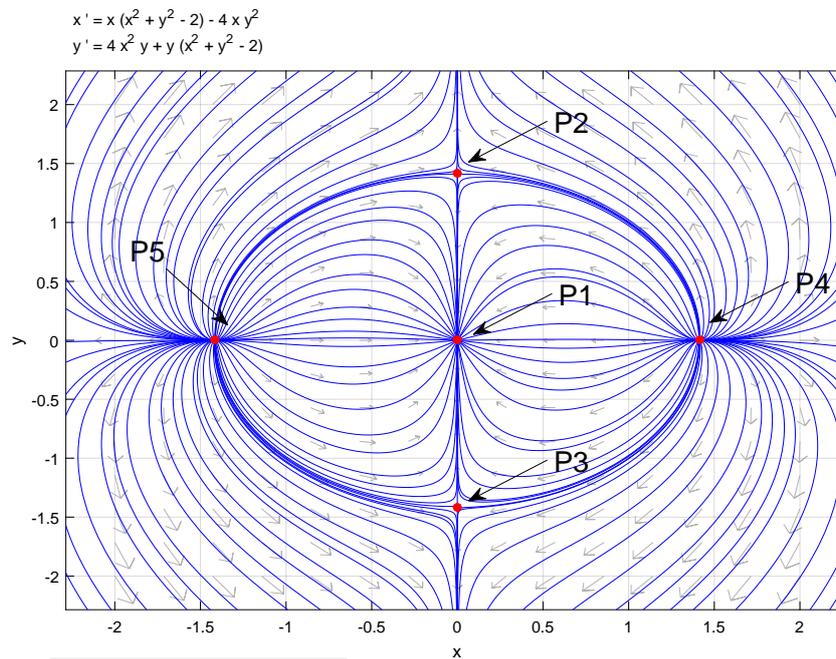


Figura 3: Plano de fase do sistema em (1).

Fonte: Gerada pelo MATLAB.

b) No plano  $x_1 \times x_2$ , escolha uma condição inicial (diferente da origem) dentro da bola de raio  $\sqrt{2}$  centrada na origem e mostre que a trajetória evolui até alcançar a origem. Escolha uma condição inicial fora da bola de raio  $\sqrt{2}$  centrada na origem e analise a evolução da trajetória.

A Figura 4 mostra que, considerando uma condição inicial dentro do conjunto  $\omega_\ell$ , as trajetórias evoluem para a origem. Todavia, para a condição externa escolhida, a trajetória não convergiu para a origem.

c) Para a condição inicial (diferente da origem) dentro da bola de raio  $\sqrt{2}$  centrada na origem escolhida anteriormente, plote a resposta temporal de  $x_1$  e  $x_2$ .

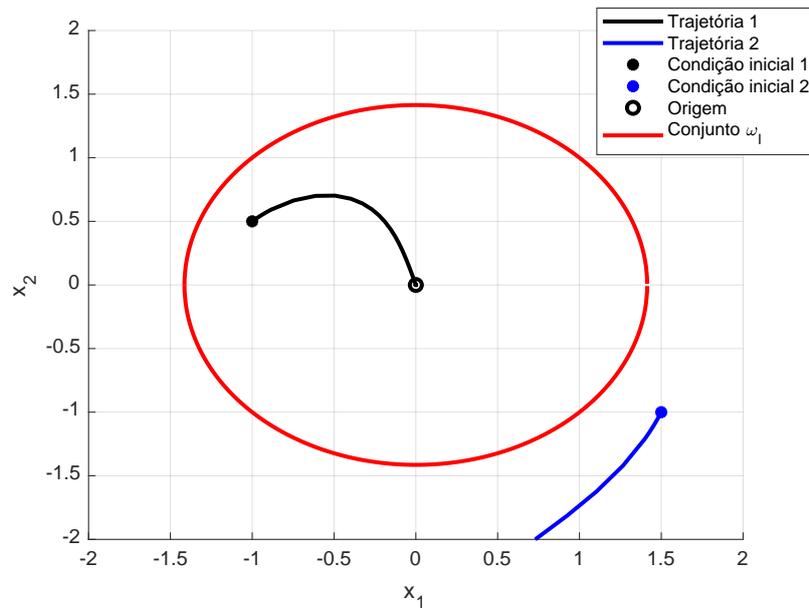


Figura 4: Evolução das trajetórias do sistema em (1) no plano  $x_1 \times x_2$ .  
Fonte: Gerada pelo MATLAB.

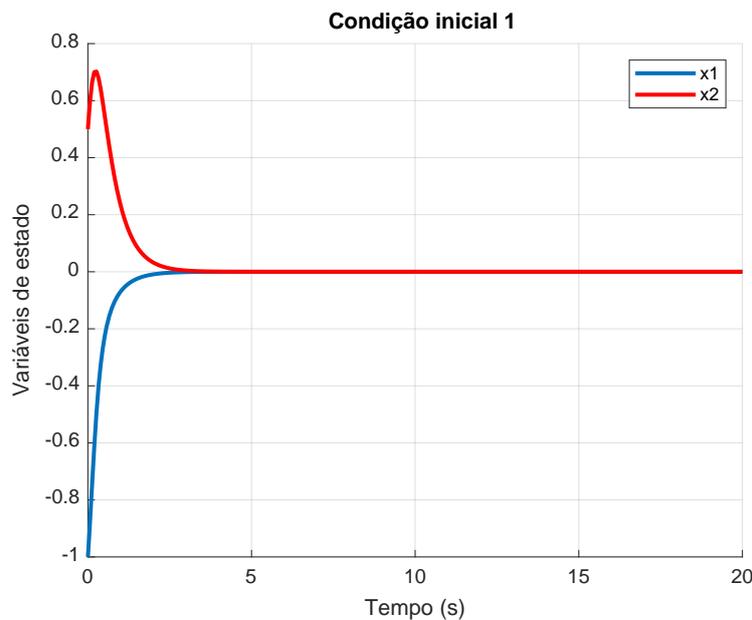


Figura 5: Resposta temporal das variáveis de estado do sistema em (1).  
Fonte: Gerada pelo MATLAB.

3. (4 pontos) A função de transferência que relaciona a velocidade de saída  $\omega(t)$  de um dado motor CC com a tensão aplicada à sua armadura  $V_a(t)$  é dada a seguir, representando a planta a ser controlada.

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{138,5}{s^2 + 1,48s + 12,81}. \quad (2)$$

- a) Projetar um controlador fuzzy PD para manter a saída em  $\omega_{ref} = 1 \text{ rad/s}$ . Verifique a existência de um pequeno erro de regime na resposta ao degrau. Exibir as variáveis de entrada e saída, funções de pertinência, base de regras e superfície *fuzzy*. Exibir o diagrama completo de simulação. Plotar a curva de velocidade e sinal de controle.

Projetou-se um controlador linear, o qual será descrito a seguir. O sistema criado possui

duas entradas, erro (eP) e variação do erro (eD), com três funções de pertinência para cada entrada, e cinco funções de pertinência para a saída, sendo estas exibidas na Figura 6. Na Tabela 4 pode-se analisar o conjunto de regras adotadas para o sistema. É possível perceber a existência de uma simetria da diagonal secundária com relação às variáveis linguísticas de saída que é necessária para obter-se uma superfície de controle linear, exibida na Figura 7. Para garantir a linearidade, torna-se necessário adotar alguns critérios no desenvolvimento do controlador *fuzzy*, dados a seguir.

- Base de regras contendo todas as combinações possíveis das entradas;
- Funções de pertinências triangulares para as entradas, sendo igualmente espaçadas e se cruzando em pontos nos quais o valor da pertinência é 0,5;
- Uso de multiplicação para o conectivo E;
- Funções de pertinência para a saída do tipo *singletons* posicionadas nos picos das funções de pertinência da entrada;
- Acumulação por soma;
- Método de *defuzzificação* por centro de área.

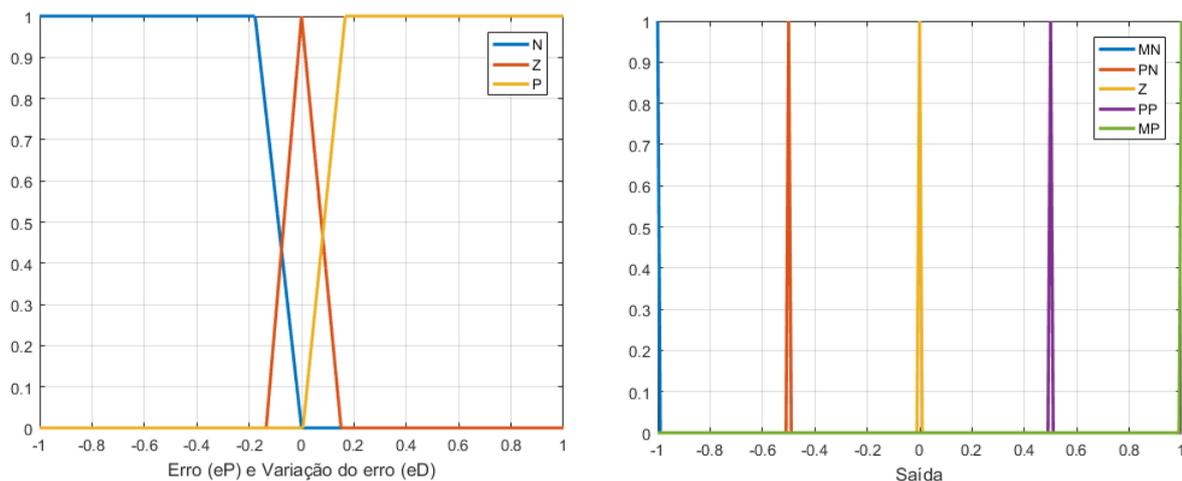


Figura 6: Entradas e saída do sistema *fuzzy*.  
Fonte: Gerada pela *toolbox fuzzy* do MATLAB.

As variáveis linguísticas escolhidas e seus respectivos significados são dados nas Tabelas 2 e 3.

Tabela 2: Variáveis linguísticas adotadas para as entradas do sistema *fuzzy*.

Variável linguística	Significado
P	“Positivo”
Z	“Zero”
N	“Negativo”

A sintonia dos ganhos foi feito por meio da sintonia manual, observando algumas regras.

- O ajuste de GE deve maximizar o universo de discurso, mas a escolha de um valor muito grande torna o sistema menos estável;
- O ajuste de GCE deve ser um valor tão grande quanto possível preserva a estabilidade do sistema;

Tabela 3: Variáveis linguísticas adotadas para a saída do sistema *fuzzy*.

Variável linguística	Significado
MP	“Muito positivo”
PP	“Pouco positivo”
Z	“Zero”
PN	“Pouco negativo”
MN	“Muito negativo”

Tabela 4: Regras de inferência do sistema *fuzzy* linear.

		eD			
		N	Z	P	
		N	MN	PN	Z
eP	Z	PN	Z	PP	
	P	Z	PP	MP	

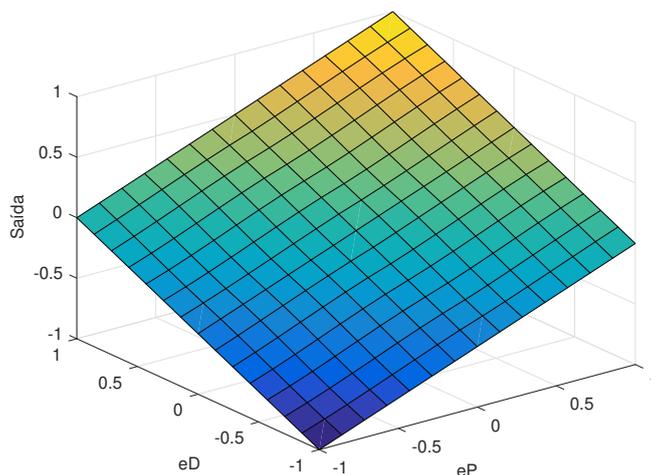


Figura 7: Superfície de controle *fuzzy* linear.  
 Fonte: Gerada pela *toolbox fuzzy* do MATLAB.

(c) O ganho  $GU$  deve ser grande o suficiente sem criar sobressinal. Caso seu valor seja pequeno, o sistema pode apresentar-se lento. Caso seu valor seja demasiadamente grande, o sistema pode apresentar-se instável.

Os ganhos foram  $GE = 1$ ,  $GCE = 0.3$  e  $GU = 15$ . A resposta da planta em malha fechada com o controlador *fuzzy* PD é exibida na Figura 8. Pode-se notar um pequeno erro de regime na saída da planta. OBS: o projeto aqui apresentado é apenas um exemplo de solução, não sendo necessário que a solução seja a mesma.

- b) Acrescentar uma ação integral para eliminar o erro de regime a uma perturbação degrau na entrada da planta. Use uma mudança na referência de  $\omega_{ref} = 1 \text{ rad/s}$  para  $\omega_{ref} = 0,7 \text{ rad/s}$  em  $t = 20 \text{ s}$ . Plotar a curva de velocidade e sinal de controle. O controlador deve garantir um sobressinal inferior a 10 %. O amplificador de potência do motor CC da questão possui um limite na entrada de 5 V.

Para inserir uma ação integral ao controlador *fuzzy*, deve-se projetar um *fuzzy* PD+I, cujo diagrama de blocos é exibido a seguir na Figura 9. O sistema *fuzzy* utilizado foi idêntico ao

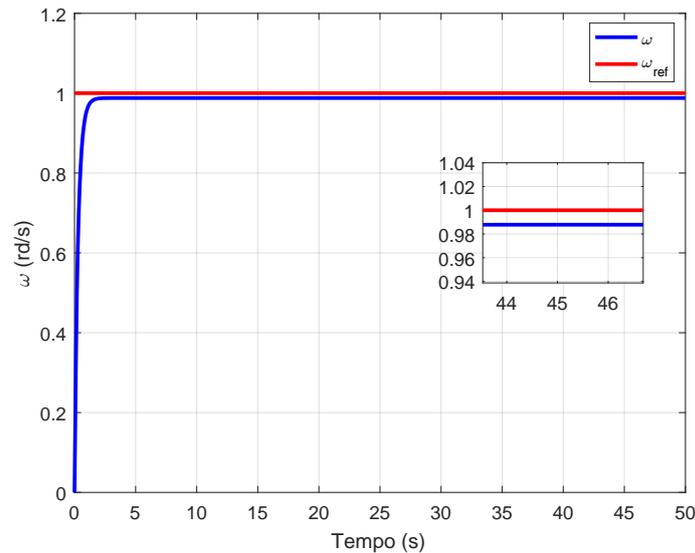


Figura 8: Resposta em malha fechada com o controlador linear *fuzzy* PD.  
Fonte: Gerada pelo MATLAB/Simulink.

descrito no item (a). A ação integradora foi implementada com um ganho de  $GIE = 0.3$ , sendo a sintonia dos ganhos realizada pelo método manual.

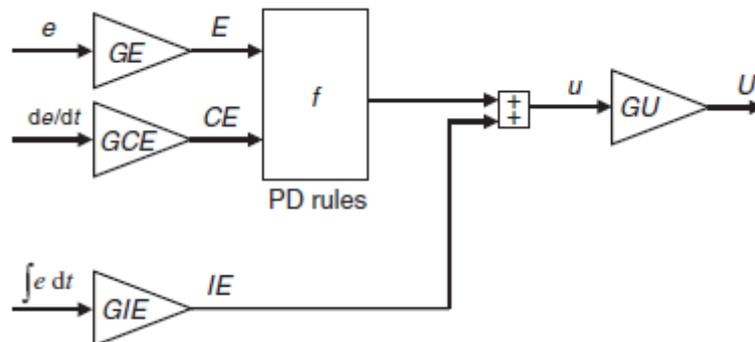


Figura 9: Controlador *fuzzy* PD+I.  
Fonte: Jantzen (2007).

Aplicou-se uma perturbação degrau na entrada da planta em  $t = 20s$  para a planta com realimentação unitária negativa. A resposta temporal é exibida na Figura 10. Pode-se notar que após aplicação da perturbação na entrada da planta, o controlador foi capaz de eliminar o erro de regime gerado retornando a saída para a referência. O valor de pico atingiu  $\omega = 1.098$  rad/s, caracterizando um *overshoot* de 9,8%. A tensão de armadura aplicada ao motor CC é exibida na Figura 11. Pode-se perceber que o valor não ultrapassa o limite de 5 V estabelecido.

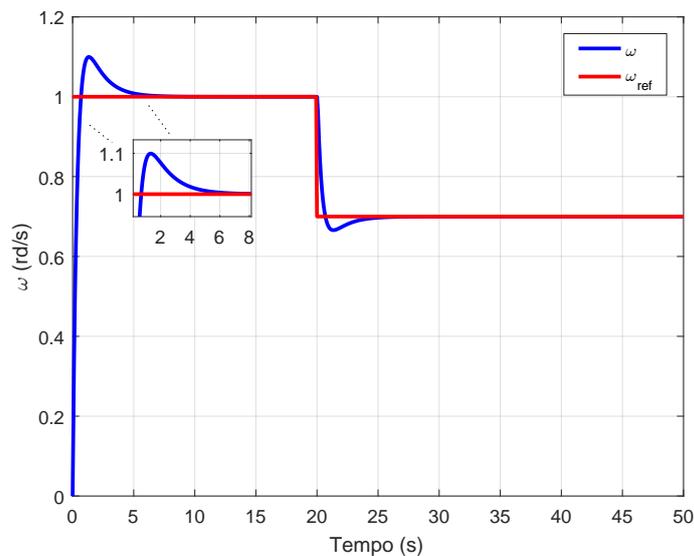


Figura 10: Resposta em malha fechada com o controlador linear *fuzzy* PD+I para uma perturbação em  $t = 20$ s.

Fonte: Gerada pelo MATLAB/Simulink.

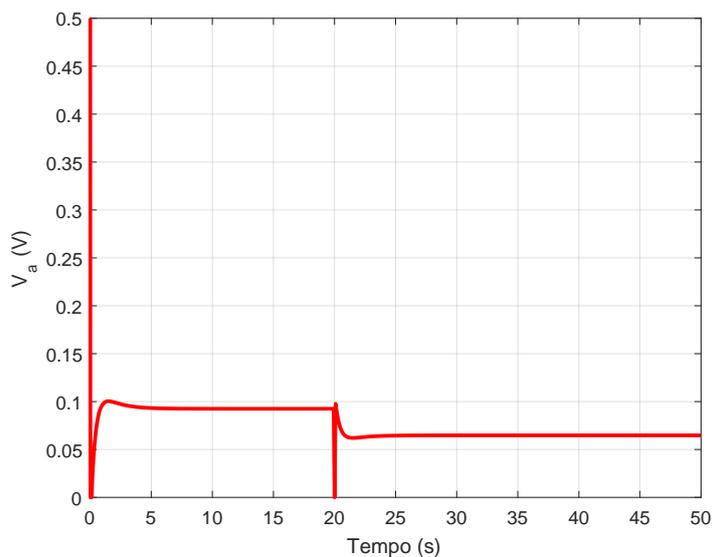


Figura 11: Tensão de armadura aplicada ao motor.

Fonte: Gerada pelo MATLAB/Simulink.