



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
APRENDIZAGEM COMPUTACIONAL
MESTRADO EM ENGENHARIA BIOMÉDICA

Sistemas Fuzzy e Neuro-Fuzzy

Prática Laboratorial 1

Mariana Montenegro 2019245964

Pedro Ferreira 2019170165

9 de dezembro 2022

Δe_k e_k	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

Figure 2: Conjunto de 25 regras utilizado na construção dos controladores.

Source: Theoretical content: Chapters 6,7,8

Δe_k e_k	N	ZE	P
N	N	N	Z
ZE	N	Z	P
P	Z	P	P

Figure 3: Conjunto de 9 regras utilizado na construção dos controladores.

Source: Theoretical content: Chapters 6,7,8

3 Sistema

3.1 Caracterização do Sistema

A função de transferência usada no sistema dinâmico foi a seguinte (G4PL1):

$$G(z) = Y(z)U(z) = \frac{25}{s^3 + 9s^2 + 28s + 30}$$

Um sistema é considerado estável se toda a entrada limitada tem uma resposta limitada. Assim, segundo o conceito de BIBO-estabilidade (bounded input-bounded output), um sistema é estável se os polos da função de transferência têm parte real negativa, isto é, estão localizados no semiplano esquerdo (SPE). Através da função roots do Matlab, foram calculados os polos do sistema e foram obtidos os valores: -3 (polo real), -3+i, -3-i (complexos conjugados). Pode-se verificar que todos os polos estão no SPE, logo o sistema é estável. De seguida, através da função pzmap, representamos os polos no plano S, e o resultado encontra-se na figura 4.

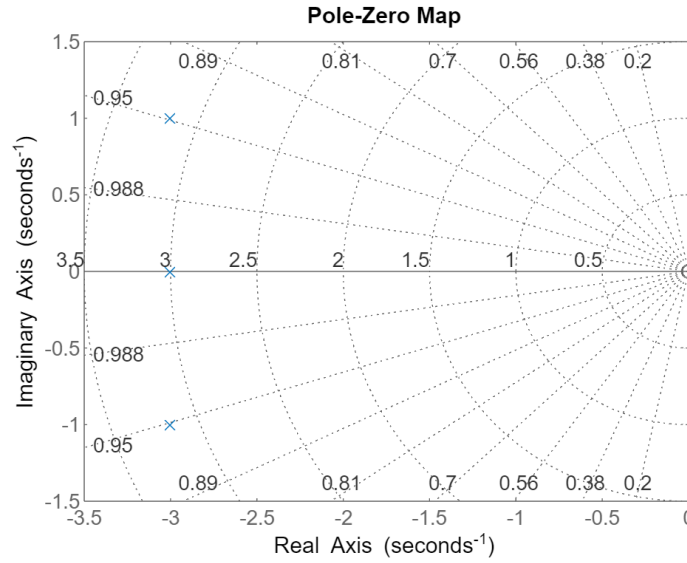


Figure 4: Polos da função de transferência do sistema.

Outro dado importante a ter em conta, é o facto de a função de transferência não possuir nenhum zero. Assim, estaremos perante uma maior inércia em comparação com um sistema com cuja função de transferência possui zeros. A inércia tem valores superiores quando o grau do denominador é maior do que o grau do numerador, ou seja, o sistema tem mais polos do que zeros, sendo que isto vai exigir um maior esforço por parte do controlador.

3.2 Avaliação da performance

Para avaliar os controladores, utilizou-se um diagrama em Simulink, que está representado na figura 5, com a função de transferência do ponto anterior. A performance foi obtida através do cálculo da integral do erro quadrático entre a entrada de referência e a saída do sistema. Uma outra medição importante na avaliação do sistema é o esforço do controlador, sendo que para tal foi calculado o máximo do controlo de ação e o integral quadrado do controlo de ação. Para calcular o máximo, usou-se o bloco MinMax Running Resettable apenas com a ligação u. Foi usada uma entrada de referência (Reference) com amplitude 1 e frequência 0.05 rad/s e variou-se a forma da onda, sendo utilizadas ondas do tipo sinusoidais, quadradas e sawtooth.

O sistema foi simulado durante 300s e, nos casos de avaliação dos controladores na presença de perturbações, estas foram inseridas nos instantes $T=100s$ (Actuator Disturbance) e $T=200s$ (Load Disturbance). Estas perturbações foram colocadas em instantes temporais relativamente afastados de modo a que o controlador tivesse tempo de recuperar. Nos valores iniciais e finais e no tempo de amostra das perturbações foram mantidos os valores default do Matlab, sendo estes 0, 1 e 0, respetivamente. Por outro lado, na análise sem perturbações, a ligação destes blocos ao sistema foi removida. Relativamente aos dois fatores de escala, foram testadas diversas combinações de valores de modo a encontrar o conjunto que permita um melhor controlo do sinal e, consequentemente, o melhor desempenho possível. Os melhores pares de valores encontrados estão apresentados nas tabelas 5-10.

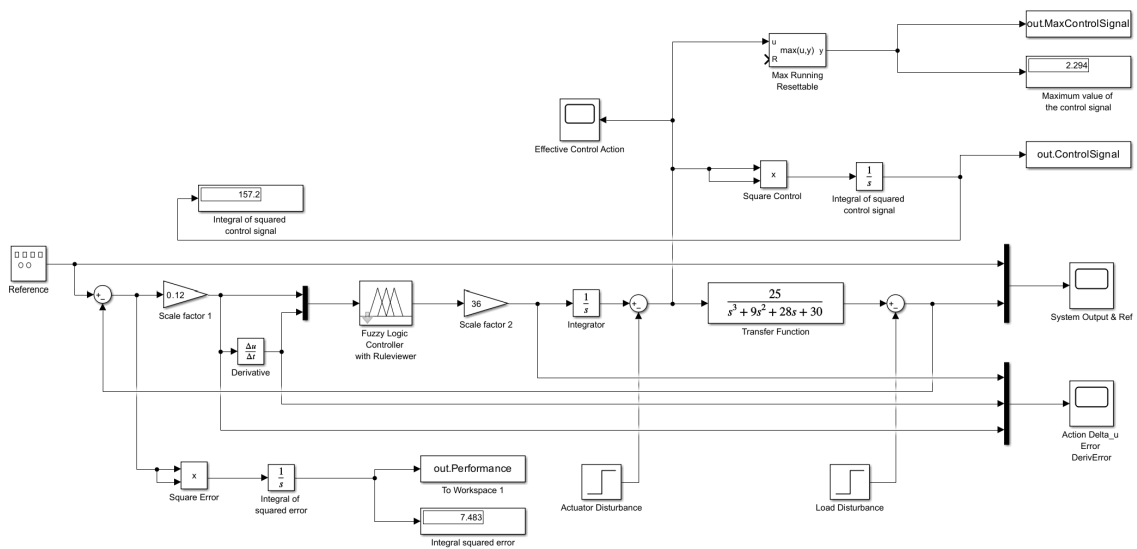


Figure 5: Diagrama de blocos do modelo em Simulink.

4 Resultados

A partir do modelo Simulink acima descrito, foram obtidos os valores de erro e do controle de ação dos vários controladores construídos, a fim de avaliar as suas performances. Estes valores foram obtidos com e sem perturbações e para inputs com três referências diferentes: sinusoidal, quadrada e sawtooth.

Os valores de erro obtidos encontram-se expostos nas seguintes tabelas, sendo que as primeiras 3 tabelas correspondem a controladores do tipo Mamdani e as seguintes 3 a controladores do tipo Sugeno.

Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Mamdani com Referência Quadrada							
Mamdani_9rules	Triangular	0.18	34	19.39	569	17.77	419.5
Mamdani_9rules	Gaussiana	0.23	34	19.78	561.4	17.89	414.6
Mamdani_25rules	Triangular	0.15	25	16.35	586.4	15.87	432.6
Mamdani_25rules	Gaussiana	0.27	20.45	15.91	592.8	15.2	441
Mamdani_49rules	Triangular	0.54	5.7	18.02	583.3	17.84	430.4
Mamdani_49rules	Gaussiana	0.5	7.9	16.75	589	16.11	434.3

Table 1: Erros obtidos para o controlador Mamdani com referência quadrada.

Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Mamdani com Referência Sinusoidal							
Mamdani_9rules	Triangular	1.59	6.94	0.99250	366.1	1.024	223.6
Mamdani_9rules	Gaussiana	1.5	8.5	1.304	365	0.446	222.6
Mamdani_25rules	Triangular	1	4.4	0.9647	367.5	0.04155	223.6
Mamdani_25rules	Gaussiana	0.1	70	0.9258	366.8	0.05438	223.5
Mamdani_49rules	Triangular	0.4	6.45	0.9877	367.1	0.02591	223.6
Mamdani_49rules	Gaussiana	0.6	6.5	0.912	367.7	0.02975	223.6

Table 2: Erros obtidos para o controlador Mamdani com referência sinusoidal.

Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Mamdani com Referência Sawtooth							
Mamdani_9rules	Triangular	0.6	17	8.536	286.7	8.251	149
Mamdani_9rules	Gaussiana	0.13	151	9.772	283.6	8.023	148.2
Mamdani_25rules	Triangular	0.1	35.99	8.714	287.9	8.077	150.6
Mamdani_25rules	Gaussiana	0.11	37.28	9.224	284	8.481	145.9
Mamdani_49rules	Triangular	0.25	10	9.774	282.6	9.346	146.2
Mamdani_49rules	Gaussiana	0.25	10.5	9.656	282.6	9.077	145.4

Table 3: Erros obtidos para o controlador Mamdani com referência sawtooth.

Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Sugeno com Referência Quadrada							
Sugeno_9rules	Triangular	0.5	10.1	16.79	585.6	16.05	432.8
Sugeno_9rules	Gaussiana	0.6	12.05	15.4	591.8	14.44	443
Sugeno_25rules	Triangular	0.5	6.9	17.21	606.6	16.97	454.4
Sugeno_25rules	Gaussiana	0.48	6.9	16.26	581.1	15.35	428.6
Sugeno_49rules	Triangular	0.3	10.5	17.12	589.8	15.91	436
Sugeno_49rules	Gaussiana	0.3	10.55	16.06	580	15.32	427.8

Table 4: Erros obtidos para o controlador Sugeno com referência quadrada.

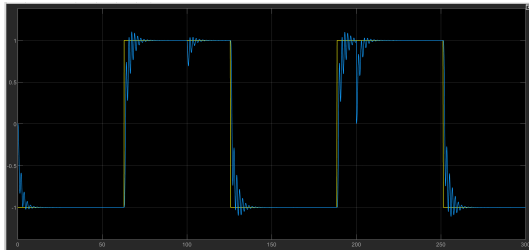
Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Sugeno com Referência Sinusoidal							
Sugeno_9rules	Triangular	0.55	7.6	0.9991	365.8	0.06093	223.3
Sugeno_9rules	Gaussiana	0.7	10	0.8917	366.9	0.04891	223.6
Sugeno_25rules	Triangular	0.4	7	1.025	367.4	0.02517	223.7
Sugeno_25rules	Gaussiana	0.4	10	0.9	367.2	0.03849	223.5
Sugeno_49rules	Triangular	0.55	5.2	0.9866	367.3	0.0244	223.7
Sugeno_49rules	Gaussiana	0.55	7	0.9169	367.3	0.04195	223.6

Table 5: Erros obtidos para o controlador Sugeno com referência sinusoidal.

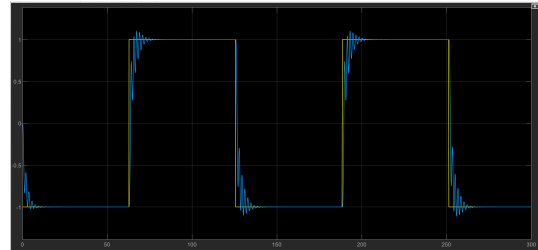
Controlador	Membership Function	Scale Factor 1	Scale Factor 2	Com Perturbação		Sem Perturbação	
				Integral do Erro Quadrado	Integral do Control Action Quadrado	Integral do Erro Quadrado	Integral do Control Action Quadrado
Controlador Sugeno com Referência Sawtooth							
Sugeno_9rules	Triangular	0.1	47	9.126	286.9	8.204	148.9
Sugeno_9rules	Gaussiana	0.45	13	8.534	286.2	7.784	148
Sugeno_25rules	Triangular	0.4	7	9.718	284.7	8.819	147.6
Sugeno_25rules	Gaussiana	0.15	28	8.223	294.6	7.405	154.3
Sugeno_49rules	Triangular	0.09	30	9.792	283.5	8.843	145.9
Sugeno_49rules	Gaussiana	0.12	36	8.343	298.2	7.483	157.2

Table 6: Erros obtidos para o controlador Sugeno com referência sawtooth.

Foram então escolhidos os controladores de dentro de cada tipo (Mamdani e Sugeno), com o melhor desempenho para cada tipo de referência, ou seja, o controlador com melhores resultados em cada tabela cujo erro com perturbação fosse menor (sublinhados a verde). Para cada um destes o gráfico da respetiva simulação encontra-se representado nas seguintes figuras devidamente legendadas. Notar que a amarelo, está o gráfico da referência utilizada, o input, e a azul, o gráfico da resposta do controlador, ou seja, o output.

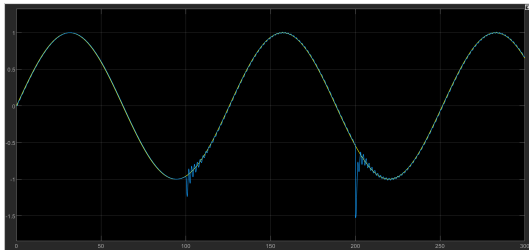


(a) Com perturbação

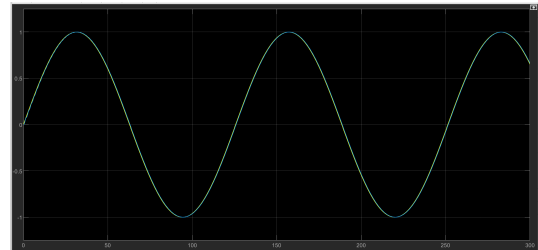


(b) Sem perturbação

Figure 6: Melhor controlador Mamdani com referência quadrada.

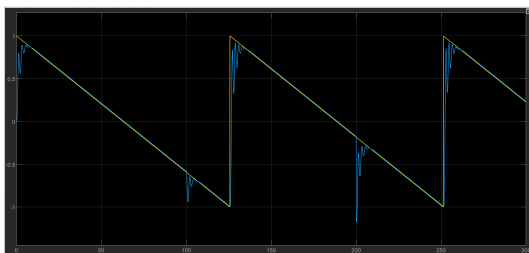


(a) Com perturbação

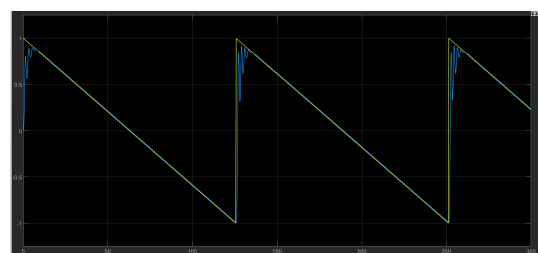


(b) Sem perturbação

Figure 7: Melhor controlador Mamdani com referência sinusoidal.

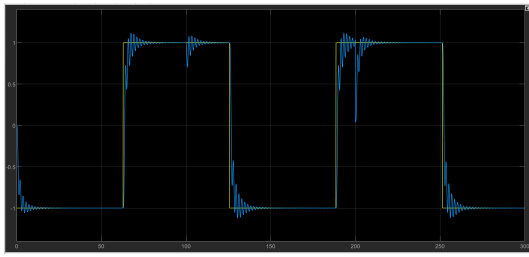


(a) Com perturbação

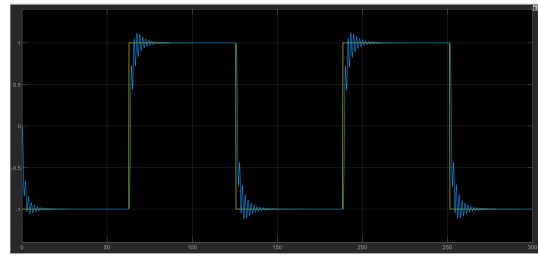


(b) Sem perturbação

Figure 8: Melhor controlador Mamdani com referência sawtooth.

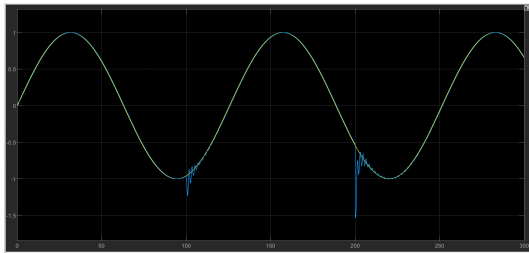


(a) Com perturbação

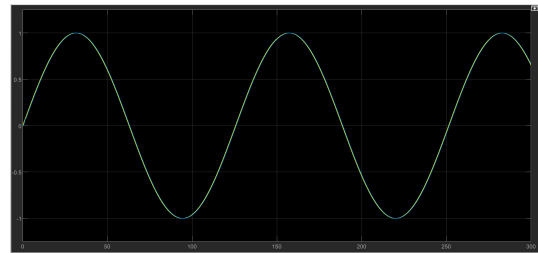


(b) Sem perturbação

Figure 9: Melhor controlador Sugeno com referência quadrada.

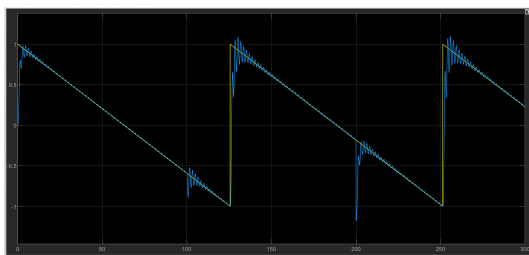


(a) Com perturbação

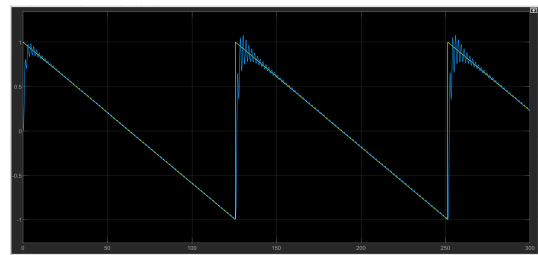


(b) Sem perturbação

Figure 10: Melhor controlador Sugeno com referência sinusoidal.



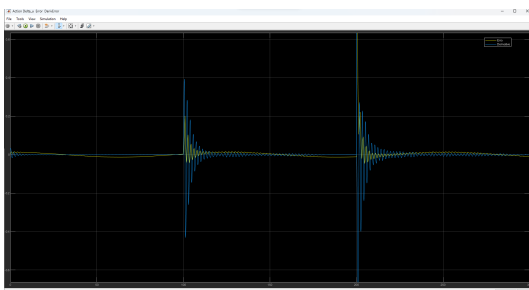
(a) Com perturbação



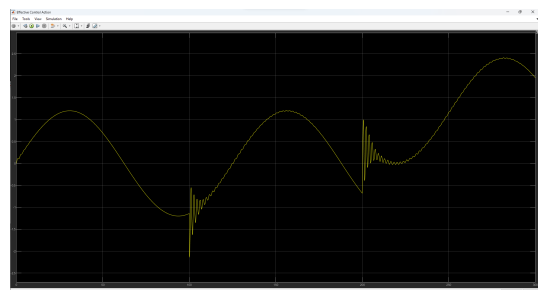
(b) Sem perturbação

Figure 11: Melhor controlador Sugeno com referência sawtooth.

Com vista a observar o controlo efetivo do controlador dinâmico inclui-se a seguir 2 gráficos relativo à evolução do erro e da sua derivada para 3 bons controladores anteriormente identificados. Ao lado do gráfico do erro e da sua derivada apresentamos também o gráfico do Controlo Efetivo do Sistema para cada um dos gráficos.

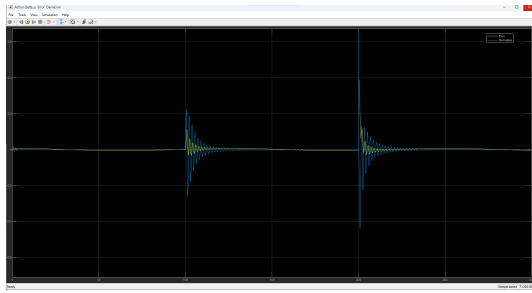


(a) Erro e Derivada

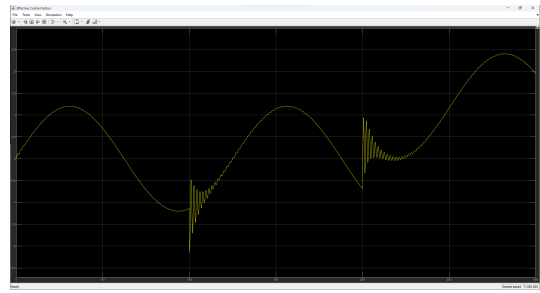


(b) Controlo Efetivo

Figure 12: Evolução do Erro e Derivada e do Controlo Efetivo para controlador Sugeno Com 9 Regras e MF Gaussiana com referência Sinusoidal.

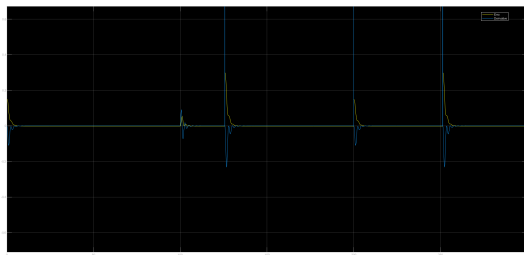


(a) Erro e Derivada

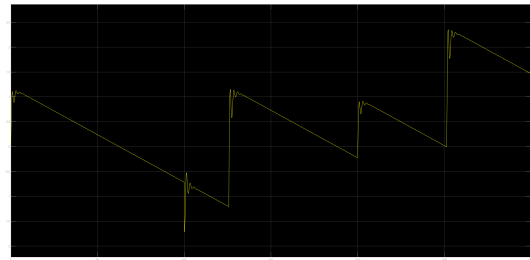


(b) Controle Efetivo

Figure 13: Evolução do Erro e Derivada e do Controle Efetivo para controlador Sugeno Com 25 Regras e MF Gaussiana com referência Sinusoidal.



(a) Erro e Derivada



(b) Controle Efetivo

Figure 14: Evolução do Erro e Derivada e do Controle Efetivo para controlador Sugeno Com 9 Regras e MF Gaussiana com referência Sawtooth.

5 Conclusão

Para avaliar a performance dos nossos controladores, foram tidos em conta os valores dos integrais quadrados do erro e o desvio da trajetória do gráfico output em relação à referência. Foi ainda feito o balanceamento entre estes dados e a taxa de esforço do controlador, ou por outras palavras, o consumo energético (parâmetros: máximo do control action e integral quadrado do control action).

Analisando as tabelas 1-6 é possível observar que, geralmente, quando aplicados os controladores de 9 regras, os resultados foram melhores, isto é, foram apresentados os menores erros. No entanto, as diferenças de resultados obtidos entre os controladores de 49 e de 25 regras não são significativos, o que nos permite concluir que ao aumentarmos o número de regras, acima de um certo patamar, apenas estamos a aumentar a complexidade do sistema, sem ganhos significativos na sua performance.

Comparando agora a performance dos controladores tendo por base a sua membership function, apercebemo-nos de que na globalidade dos resultados a função gaussiana e a função triangular, apresentam um desempenho semelhante.

Relativamente ao tipo de controlador, os resultados entre Mamdani e Sugeno são relativamente semelhantes. Foi ainda possível aferir que a resposta dos controladores é melhor quando a referência é uma onda sinusoidal, intermédia quando é uma onda sawtooth e apresentam os piores resultados para as ondas quadradas. Este dado é esperado teoricamente, uma vez que, o controlador consegue ajustar-se mais facilmente à referência quando a variação da amplitude da onda é mais suave. No caso de ondas com mudanças bruscas, como é o caso das ondas quadradas, o controlador tem mais dificuldade em ajustar-se à resposta, o que se reflete em erros superiores.

Adicionalmente, importa referir a capacidade que os controladores revelam a dar resposta a perturbações rápida e eficazmente. É ainda relevante mencionar que a maior dificuldade encontrada na realização desta parte do trabalho foi a escolha dos fatores de escala. Foram realizadas inúmeras tentativas com diferentes combinações de valores de modo a encontrar o par que apresentasse um menor erro e, simultaneamente, mantivesse um esforço reduzido por parte do controlador. Todavia, existe uma infinidade de combinações possíveis, o que torna muito difícil encontrar a combinação de valores ideais para estes parâmetros.

Por fim, é de notar que, os resultados obtidos dependem ainda do tempo de simulação escolhido, da frequência e da amplitude da referência, e da função de transferência do sistema.

Em situações reais é importante saber qual é a aplicação final de um controlador, de modo a que este seja detentor das características adequadas ao sistema em que estará inserido. Como no nosso caso apenas possuímos a função de transferência e não sabemos a contextualização do problema, optámos por desenvolver o trabalho com o objetivo de diminuir simultaneamente o erro e as oscilações do sistema. Tendo em conta os resultados apresentados, podemos concluir que o objetivo do projeto foi cumprido de forma satisfatória.

Parte B: Sistemas Neuro-Fuzzy

1 Introdução

Na segunda parte do trabalho, o objetivo passa por utilizar sistemas Neuro-Fuzzy para modelar sistemas dinâmicos. Nestes sistemas o output é, frequentemente, dependente de instantes anteriores, ou seja, além de depender dos inputs do presente, estes sistemas dependem ainda de inputs e outputs passados. Pode-se, portanto, dizer que, os sistemas Neuro-Fuzzy são caracterizados pela presença de memória e são descritos por modelos autoregressivos não lineares com entradas exógenas (NARX).

A geração destes sistemas está, deste modo, dependente da determinação da membership function dos antecedentes e das constantes dos consequentes. É na fase de aprendizagem que se procede à determinação destes dados. Em seguida, através da utilização técnicas de clustering é possível obter as regras, sendo a configuração otimizada a posteriori.

2 Dataset

Para ser possível determinar o conjunto de regras recorrendo às técnicas de clustering, existiu a necessidade de gerar um dataset representativo da dinâmica do sistema. Com esse intuito procedeu-se à criação de um modelo em Simulink.

Inicialmente foi necessário passar a função de transferência atribuída ao nosso grupo (G4PL1) para a sua forma discreta, para poder ser utilizada no nosso modelo em Simulink. Para fazer esta discretização recorreu-se à função `c2dm` do MATLAB, com o método de discretização `zoh` e com um intervalo de discretização de 0.0833 segundos, que foi calculado da seguinte maneira: $\frac{1}{4}$ da menor constante temporal $\tau_1 = -\frac{1}{\rho_i}$ sendo ρ_i os polos reais do sistema. Com este passo foram obtidos os coeficientes do numerador e do denominador da função de transferência discreta, que ficou com a seguinte forma:

$$F(z) = \frac{0,002z^2 + 0,0066z + 0,0014}{z^3 - 2,331z^2 + 1,8154z - 0,4724}$$

Foi então elaborado o modelo de Simulink. Este consiste essencialmente num gerador de números aleatórios ligado à função de transferência na forma acima explicitada. Para esta simulação o solver utilizado foi a `ode4` (Runge-Kutta), com passo fixo equivalente ao sample time, 0.0833, e o tempo de simulação escolhido foi de 625 segundos (múltiplo do passo 0.125). O diagrama de blocos deste modelo está reproduzido na figura seguinte:

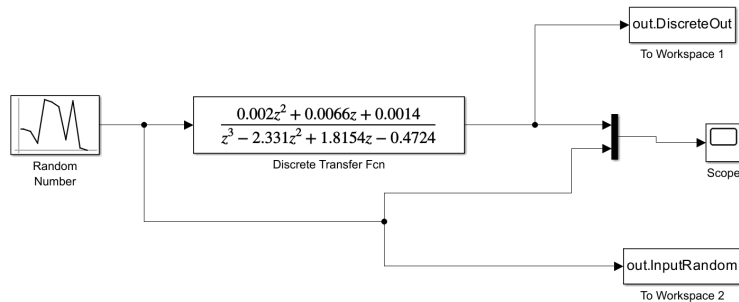


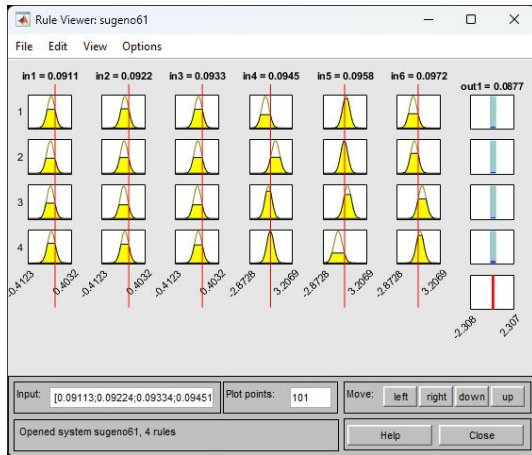
Figure 15: Diagrama de blocos do modelo Simulink utilizado para gerar os dados.

Com este modelo é possível a obtenção do valor de input gerado aleatoriamente e do output correspondente, após a passagem pela função de transferência, em cada instante temporal. As séries temporais obtidas foram então utilizadas para elaborar a matriz de dados que constitui o nosso dataset. Matriz esta que é formada por 7 colunas, onde a última coluna corresponde ao consequente, ou seja, o nosso target, e as primeiras 6 correspondem aos antecedentes, as features. Os dados obtidos foram então divididos em grupo de treino, 70% dos dados, e grupo de teste, os restantes 30%.

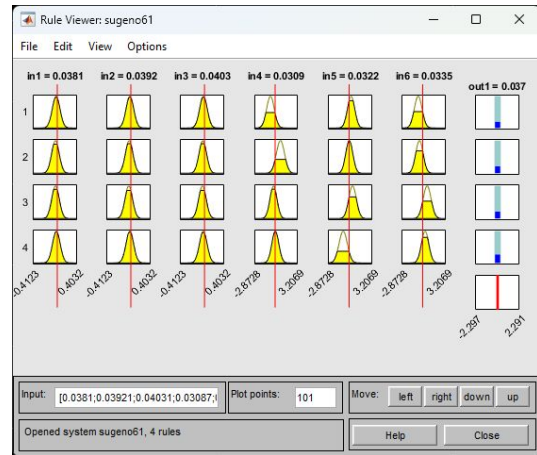
De referir que ao usar fuzzy c-means ou subtractive clustering, é gerada uma regra por cada fuzzy cluster enquanto que ao usar grid partitioning é gerada uma regra por cada combinação de membership function dos inputs. Nesta técnica de clustering, cada input tem membership functions distribuídas uniformemente pelos seus intervalos de valores.

3 Construção dos controladores

Foram gerados controladores neuro-difusos através da função `genfis` (permite gerar fis a partir dos dados de input e output e especificando qual a técnica de clustering) do MATLAB com recurso a varias técnicas de clustering: Fuzzy c-means, subtractive clustering e grid partitioning. Seguidamente, estes foram treinados utilizando 2 métodos de otimização: retropropagação e hybrid, através da função `anfis` (ajusta as membership functions para que o controlador neuro-fuzzy se adapte aos dados). Foram criados, portanto, 6 sistemas neuro-fuzzy, sendo que as regras de cada estão visíveis nas figuras 6 a 11 e foram obtidas através do ruleviewer do GUI Anfis (NeuroFuzzyDesigner).

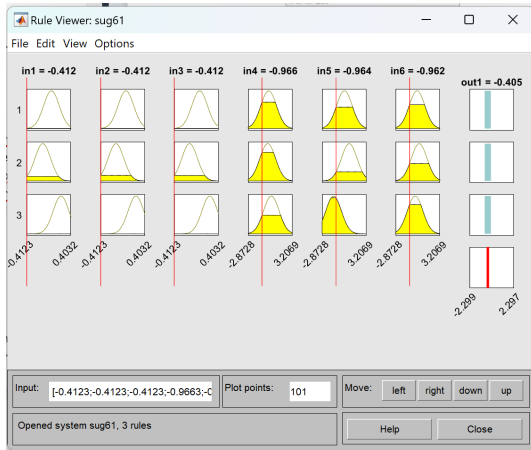


(a) Otimizado com Backpropagation.

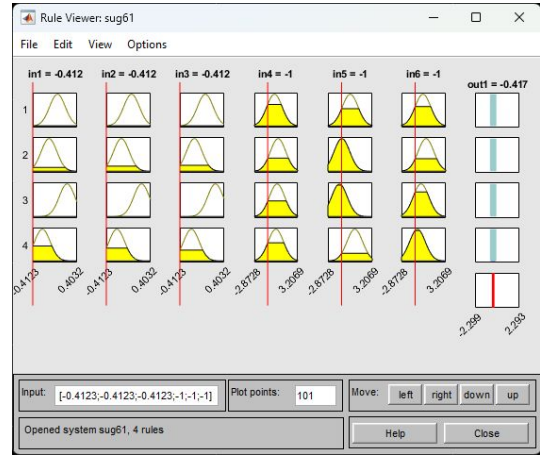


(b) Otimizado com Hybrid.

Figure 16: Regras do controlador gerado com clustering Fuzzy CMeans.

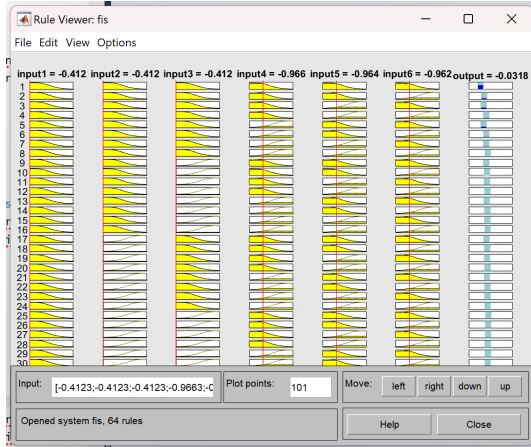


(a) Otimizado com Backpropagation.

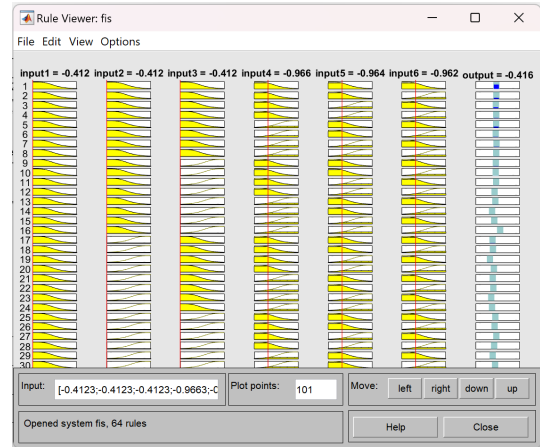


(b) Otimizado com Hybrid.

Figure 17: Regras do controlador gerado com clustering Subtractive.



(a) Otimizado com Backpropagation.



(b) Otimizado com Hybrid.

Figure 18: Regras do controlador gerado com clustering Grid Partition.

4 Avaliação de Performance

A função `evalfis` do Matlab permitiu a obtenção do output a partir dos dados de teste, sendo este comparado com os valores de target através do cálculo do erro quadrático médio (MSE). Os valores do erro para cada uma das avaliações dos sistemas estão apresentados na tabela 1 do ponto 5.

Por fim, foi construído o diagrama de blocos em Simulink da figura 8, de modo a avaliar a performance dos sistemas. As ondas de referência utilizadas foram a sinusoidal, quadrada e sawtooth, com amplitude 1 e frequência 0.05 rad/s. O sample time dos blocos delay foi de 0,0833 segundos, conforme calculado anteriormente.

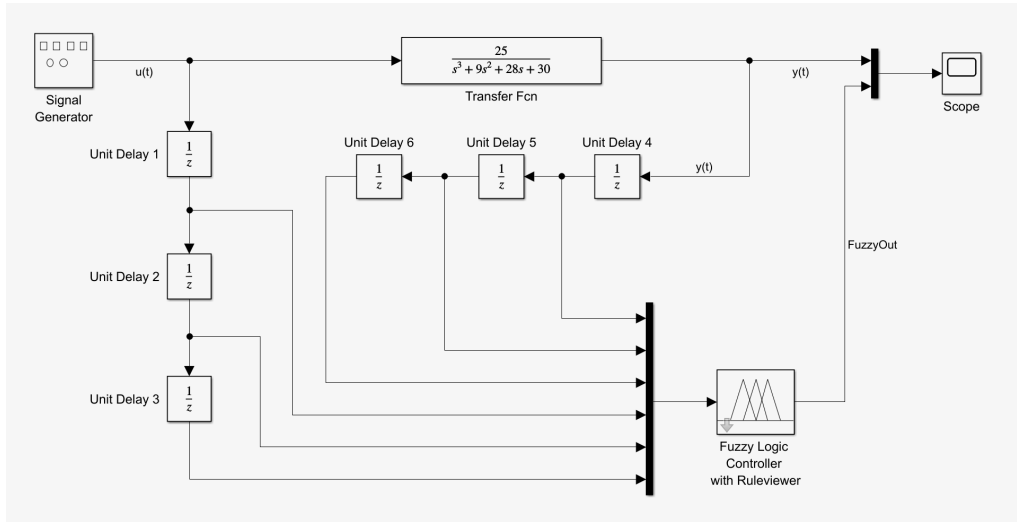


Figure 19: Diagrama de blocos do modelo Simulink utilizado para avaliar a performance.

5 Resultados

Procedeu-se então ao cálculo dos valores do Mean Square Error (MSE), utilizando a função `immse` do MATLAB no documento "test_performance.m", para os vários controladores criados, que se encontram na tabela 1. A performance dos vários sistemas difusos foi obtida através da simulação em simulink, utilizando o diagrama de blocos descrito no ponto anterior.

Método de clustering	Método de treino	Erro (MSE)	Número de regras obtido
Fuzzy C-means	Backpropagation	6.40301e-17	4
Fuzzy C-means	Hybrid	5.21118e-06	4
Subtractive	Backpropagation	1.41023e-16	3
Subtractive	Hybrid	6.27184e-06	4
Grid Partition	Backpropagation	0.136131	64
Grid Partition	Hybrid	0.000199167	64

Table 7: Erros obtidos para o controlador Sugeno com referência quadrada.

Os resultados obtidos foram melhores quando aplicados os métodos de clustering Fuzzy C-means e Backpropagation. Encontram-se em seguida expostos os gráficos quando utilizado o controlador gerado com o clustering Fuzzy C-Mean e com a otimização Hybrid, cujo erro foi ligeiramente menor, para as diferentes referências (sinusoidal, quadrada e sawtooth) utilizadas como input.

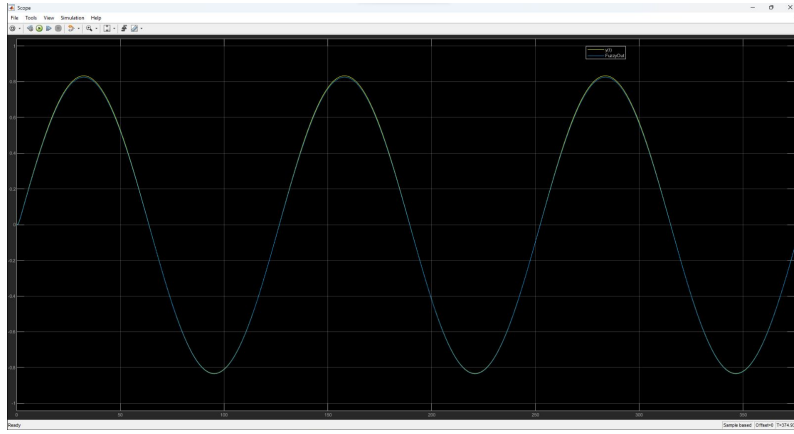


Figure 20: Performance do sistema com o controlador gerado com Fuzzy C-Means e otimização Backpropagation, para uma referência sinusoidal.

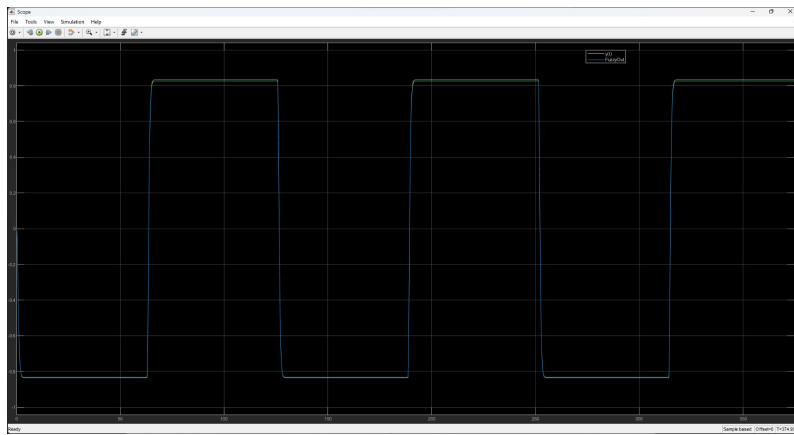


Figure 21: Performance do sistema com o controlador gerado com Fuzzy C-Means e otimização Backpropagation, para uma referência quadrada.

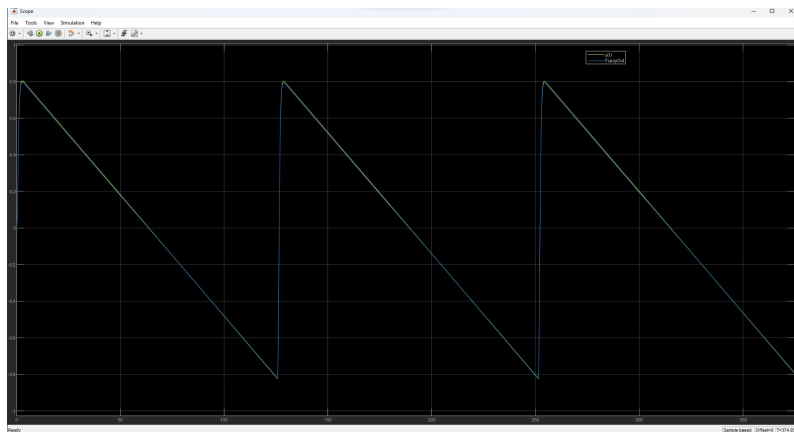


Figure 22: Performance do sistema com o controlador gerado com Fuzzy C-Means e otimização Backpropagation, para uma referência sawtooth.

Como se pode ver pela tabela 1, à exceção da combinação (Grid Partition + Backpropagation), os controladores apresentam erros muito baixos, sendo a representação gráfica destes sinais idêntica às três figuras acima e, deste modo, optamos por não incluir as imagens no relatório. Contrariamente,

no caso do controlador gerado com clustering Grid Partition com otimização Backpropagation, o erro mais elevado reflete-se numa elevada diferença entre o output e o target.

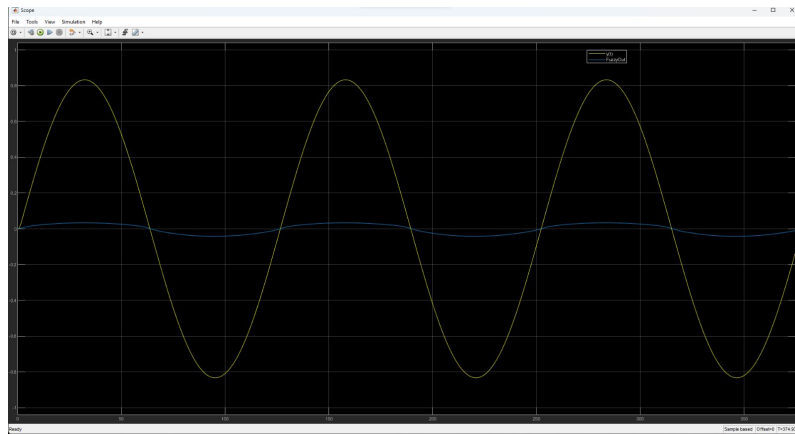


Figure 23: Performance do sistema com o controlador gerado com Grid Partition e otimização Backpropagation, para uma referência sinusoidal.

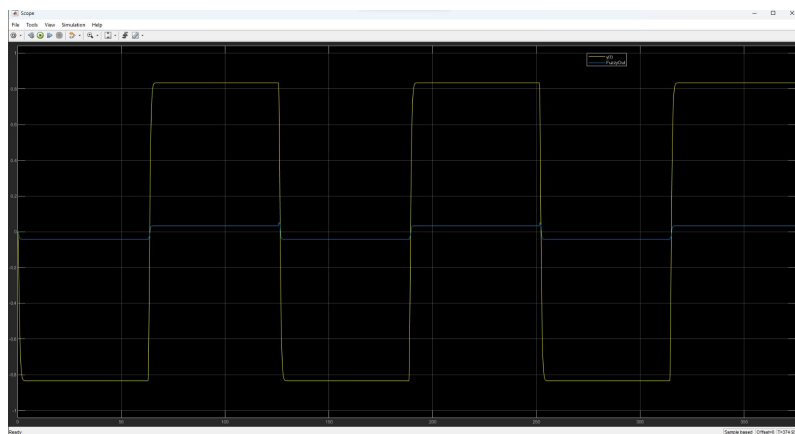


Figure 24: Performance do sistema com o controlador gerado com Grid Partition e otimização Backpropagation, para uma referência quadrada.

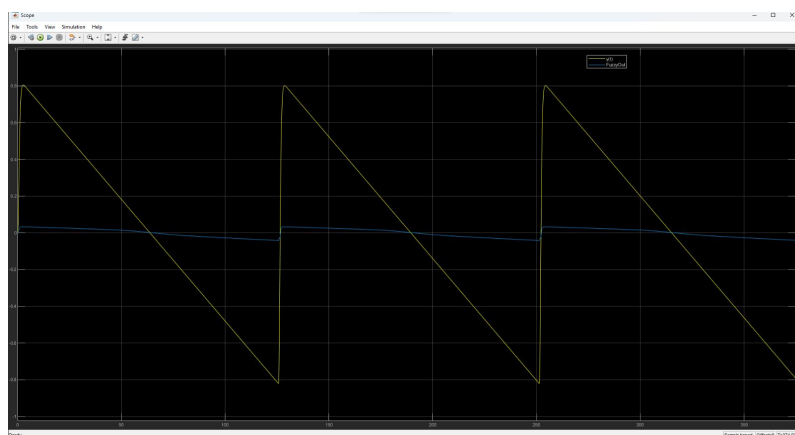


Figure 25: Performance do sistema com o controlador gerado com Grid Partition e otimização Backpropagation, para uma referência sawtooth.

6 Conclusão

Partindo para a análise da tabela 1 com os valores do erro quadrático médio obtidos utilizando os fis otimizados, observa-se que, tal como mencionado no ponto anterior, os métodos de clustering Fuzzy C-means e Subtractive apresentam resultados muito semelhantes.

Verifica-se também que o método de treino Backpropagation é mais adequado para o problema em estudo, uma vez que apresenta o menor erro em comparação com os controladores cujo método de treino foi o Hybrid. Nestes dois casos o erro foi suficientemente baixo ao ponto que o controlador se sobrepõe quase na totalidade à referência. Já o método de clustering Grid Partition, foi o que obteve piores resultados, com ambos os métodos de treino, e não o consideramos adequado para o problema em estudo.

Um dos pontos que mais influencia estes resultados é o dataset utilizado. Para equilibrar o erro em ambos os tipos de otimização, na criação do dataset, foi escolhido um tempo de simulação de 83 segundos, gerando o número de dados correspondente. Caso este tempo de simulação tivesse sido maior ou menor o número de dados também variaria, e tal iria influenciar o clustering e, portanto, as regras obtidas para os controladores.

Em suma, considerando os resultados obtidos da performance dos sistemas neurofuzzy conclui-se que estes são capazes de modelar eficazmente o sistema dinâmico.