

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Engenharia Elétrica – DEE
Coordenação de Pós-Graduação em Engenharia Elétrica – COPELE

Matheus Ferreira da Silva

Trabalho de Dissertação

**Identificação de Sistemas Não Lineares
Utilizando Aprendizagem Profunda e
Modelagem Orientada a Blocos**

Campina Grande, Brasil

Março de 2022

Matheus Ferreira da Silva

Trabalho de Dissertação

**Identificação de Sistemas Não Lineares
Utilizando Aprendizagem Profunda e
Modelagem Orientada a Blocos**

Trabalho de Dissertação apresentado à Coordenação de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

Orientador:

Péricles Rezende Barros, Ph.D

Campina Grande, Brasil

Março de 2022

S586i

Silva, Matheus Ferreira da.

Identificação de sistemas não Lineares utilizando aprendizagem profunda e modelagem orientada a blocos / Matheus Ferreira da Silva. – Campina Grande, 2022.

137 f. : il. color.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2022.

"Orientação: Prof. Dr. Péricles Rezende Barros".

Referências.

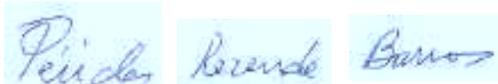
1. Identificação de Sistemas. 2. Aprendizagem Profunda. 3. Modelos de Hammerstein. 4. Modelos de Wiener I. Barros, Péricles Rezende. II. Título.

CDU 519.71(043)

Identificação de Sistemas Não Lineares Utilizando Aprendizagem Profunda e Modelagem Orientada a Blocos

MATHEUS FERREIRA DA SILVA

DISSERTAÇÃO APROVADA EM 24/02/2022

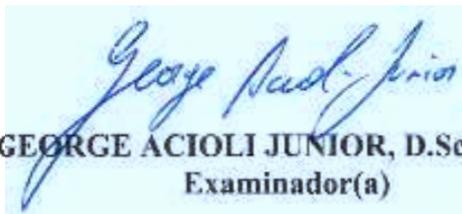


PÉRICLES REZENDE BARROS, Ph.D., UFCG
Orientador(a)

**José Sérgio da
Rocha Neto**

Assinado de forma digital
por José Sérgio da Rocha
Neto
Dados: 2022.03.04
09:16:03 -03'00'

JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Examinador(a)



GEORGE ACIOLI JUNIOR, D.Sc., UFCG
Examinador(a)

CAMPINA GRANDE - PB

Agradecimentos

Agradeço a Deus e a minha família por terem me dado suporte para chegar até aqui.

Em especial, agradeço ao meu pai, João Pedro, e minha mãe, Maria José, por sempre terem feito tanto com tão pouco. Agradeço aos meus irmãos Márcio, Magno, Maxwellton e irmãs Ana Márcia e Ana Maria, além das minhas sobrinhas Isabella e Mariana, por toda a ajuda, exemplos e bons momentos vividos. Sou imensamente grato também a minha namorada, Mikaely, por todo o amor, carinho e momentos compartilhados, por todo o apoio, compreensão e companheirismo, especialmente nos momentos mais difíceis ao longo desses anos de mestrado.

Agradeço ao professor Péricles pela sua excelente orientação e comprometimento ao longo deste trabalho. Sou grato também aos professores George e José Sérgio pelas sugestões e comentários para elaboração da versão final deste documento.

Sou muito grato aos meus estimados amigos que contribuíram de maneira importante para a realização deste trabalho de dissertação. Em especial, agradeço a Júnior, Tadeu, Ulisses, além do Professor Rafael, Pedro, William, Lucas e Ravi.

Agradeço também a todos os outros excelentes amigos que fiz ao longo destes anos de universidade e a todos que de alguma forma me incentivaram e ajudaram ao longo desta caminhada.

Resumo

Nesta dissertação de mestrado são exploradas conexões entre as áreas de identificação de sistemas e aprendizagem profunda. Neste contexto, é proposto um método de identificação de sistemas não lineares baseado em redes neurais profundas e em modelos não lineares orientados a blocos.

A abordagem proposta é formulada a partir da separação das partes linear e não linear de modelos de *Hammerstein* e *Wiener*. Nesse sentido, é utilizada uma estrutura de rede neural que reflete essa estratégia de separação e o procedimento de identificação é realizado a partir do treinamento da rede neural profunda com dados de entrada e saída dos sistemas de interesse.

Para testar o método proposto foram simulados seis diferentes sistemas de *Hammerstein* e seis sistemas de *Wiener* de diferentes complexidades a fim de gerar dados de entrada e saída de cada um dos casos. Como referência foi adotado outro método de identificação que, embora também seja baseado na separação das partes linear e não linear de modelos orientados a blocos, emprega funções de base ortonormais e radiais para a tarefa de identificação. A comparação dos resultados obtidos revela que o método proposto, de maneira geral, fornece modelos mais precisos.

Palavras-chave: *identificação de sistemas, aprendizagem profunda, modelos de Hammerstein, modelos de Wiener.*

Abstract

In this masters dissertation, connections between the areas of systems identification and deep learning are explored. In this context, a method for nonlinear systems identification based on deep neural networks and nonlinear block-oriented models is proposed.

The proposed approach is formulated from the separation of linear and non-linear parts of the *Hammerstein* and *Wiener* models. In this sense, a neural network structure that reflects this separation strategy is used and the identification procedure is performed by training the deep neural network with input and output data from the systems of interest.

To test the proposed method, six different *Hammerstein* systems and six *Wiener* systems of different complexities were simulated in order to generate input and output data for the application of identification techniques. As a reference, another identification method was adopted which, although also based on the separation of linear and non-linear parts of block-oriented models, employs orthonormal and radial basis functions for the identification task. The comparison of the results obtained reveals that the proposed method, in general, provides more accurate models.

Keywords: *system identification, deep learning, Hammerstein models, Wiener models.*

Sumário

1	Introdução	2
1.1	Contexto	2
1.2	Revisão Bibliográfica	3
1.3	Objetivos	4
1.4	Organização do texto	5
2	Conceitos Básicos de Identificação de Sistemas	6
2.1	Introdução	6
2.2	Sistemas dinâmicos	7
2.3	Modelos	7
2.4	Tipos de identificação	8
2.5	O problema de identificação paramétrica de sistemas	9
2.6	Alguns modelos de sistemas dinâmicos	12
2.6.1	Modelos lineares	13
2.6.2	Modelos não lineares	14
2.6.3	Modelagem orientada a blocos	14
2.7	Conclusão	16
3	Aprendizado de Máquina e Aprendizagem Profunda	17
3.1	Introdução	17
3.2	Dados, modelos e algoritmos	17
3.2.1	Dados	18
3.2.2	Modelos	18
3.2.3	Funções objetivo	18
3.2.4	Algoritmos de otimização	19
3.3	Aprendizagem profunda	20
3.3.1	A construção de redes neurais profundas	20
3.3.2	Treinamento de redes neurais	23
3.4	Conclusão	31

4	Identificação, <i>Deep Learning</i> e Modelagem Orientada a Blocos	32
4.1	Introdução	32
4.2	Identificação com base em modelos orientados a blocos	32
4.3	Método Proposto	39
4.3.1	Sistemas de <i>Hammerstein</i>	42
4.3.2	Sistemas de <i>Wiener</i>	47
4.4	Conclusão	53
5	Resultados de Simulações	54
5.1	Introdução	54
5.2	<i>Setup</i> de simulações	54
5.3	Identificação utilizando redes neurais profundas	57
5.4	Resultados de identificação	59
5.4.1	Caso de estudo 1 (<i>Hammerstein</i>)	59
5.4.2	Caso de estudo 2 (<i>Hammerstein</i>)	71
5.4.3	Caso de estudo 3 (<i>Hammerstein</i>)	80
5.4.4	Caso de estudo 4 (<i>Wiener</i>)	89
5.4.5	Caso de estudo 5 (<i>Wiener</i>)	99
5.4.6	Caso de estudo 6 (<i>Wiener</i>)	106
5.5	Análise dos resultados	113
5.6	Conclusão	115
6	Conclusões e Sugestões para Trabalhos Futuros	116
6.1	Conclusões	116
6.2	Sugestões para trabalhos futuros	117
	Referências Bibliograficas	118
A	Caso de estudo 1	120
B	Caso de estudo 2	124
C	Caso de estudo 3	127
D	Caso de estudo 4	130
E	Caso de estudo 5	133
F	Caso de estudo 6	136

Lista de Tabelas

5.1	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 1).	64
5.2	Valores de erro sobre o conjunto de teste para diferentes quantidades de camadas e neurônios (Caso 1).	64
5.3	Valores real e estimado para o coeficiente da parte linear de acordo com o método proposto (Caso 1).	67
5.4	Parâmetros utilizados nas RBF.	69
5.5	Coeficientes OBF obtidos (Caso 1).	69
5.6	Coeficientes RBF (Caso 1).	69
5.7	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 2).	73
5.8	Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 2).	74
5.9	Valores reais e estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 2).	76
5.10	Coeficientes OBF obtidos (Caso 2).	78
5.11	Coeficientes RBF (Caso 2).	78
5.12	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 3).	82
5.13	Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 3).	83
5.14	Valores reais e estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 3).	85
5.15	Coeficientes OBF obtidos (Caso 3).	87
5.16	Coeficientes RBF (Caso 3).	87
5.17	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 4).	92
5.18	Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 4).	92

5.19	Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 4).	95
5.20	Coeficientes OBF obtidos (Caso 4).	97
5.21	Coeficientes RBF (Caso 4).	97
5.22	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 5).	100
5.23	Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 5).	101
5.24	Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 5).	103
5.25	Coeficientes OBF obtidos (Caso 5).	104
5.26	Coeficientes RBF (Caso 5).	104
5.27	Tipos de dados de entrada a serem utilizados para o método proposto (Caso 6).	107
5.28	Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 6).	108
5.29	Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 6).	110
5.30	Coeficientes OBF obtidos (Caso 6)	111
5.31	Coeficientes RBF (Caso 6)	111
5.32	Valores de erro médio quadrático (MSE) obtidos por meio do método proposto e do método de referência para os casos de estudo abordados	113
5.33	Quantidades de camadas e neurônios para cada caso de estudo.	114
A.1	Caso 1 - X_{in}	122
A.2	Caso 1 - b	122
A.3	Caso 1 - W_{in}	122
A.4	Caso 1 - W_1	122
A.5	Caso 1 - W_2	122
A.6	Caso 1 - W_3	122
A.7	Caso 1 - W_4	122
A.8	Caso 1 - W_5	123
A.9	Caso 1 - W_6	123
A.10	Caso 1 - W_7	123
A.11	Caso 1 - W_8	123
A.12	Caso 1 - W_9	123
A.13	Caso 1 - W_{out}	123

B.1	Caso 2 - X_{in}	124
B.2	Caso 2 - b	124
B.3	Caso 2 - W_{in}	124
B.4	Caso 2 - W_1	125
B.5	Caso 2 - W_2	125
B.6	Caso 2 - W_3	125
B.7	Caso 2 - W_4	125
B.8	Caso 2 - W_5	126
B.9	Caso 2 - W_6	126
B.10	Caso 2 - W_7	126
B.11	Caso 2 - W_{out}	126
C.1	Caso 3 - X_{in}	127
C.2	Caso 3 - b	127
C.3	Caso 3 - W_{in}	127
C.4	Caso 3 - W_1	128
C.5	Caso 3 - W_2	128
C.6	Caso 3 - W_3	128
C.7	Caso 3 - W_4	129
C.8	Caso 3 - W_5	129
C.9	Caso 3 - W_{out}	129
D.1	Caso 4 - X_{in}	130
D.2	Caso 4 - b	130
D.3	Caso 4 - W_{in}	130
D.4	Caso 4 - W_1	131
D.5	Caso 4 - W_2	131
D.6	Caso 4 - W_3	131
D.7	Caso 4 - W_4	131
D.8	Caso 4 - W_{out}	132
E.1	Caso 5 - X_{in}	133
E.2	Caso 5 - b	133
E.3	Caso 5 - W_{in}	133
E.4	Caso 5 - W_1	134
E.5	Caso 5 - W	134
E.6	Caso 5 - W_3	134
E.7	Caso 5 - W_4	134
E.8	Caso 5 - W_5	135

E.9	Caso 5 - W_{out}	135
F.1	Caso 6 - X_{in}	136
F.2	Caso 6 - b	136
F.3	Caso 6 - W_{in}	136
F.4	Caso 6 - W_1	137
F.5	Caso 6 - W_2	137
F.6	Caso 6 - W_3	137
F.7	Caso 6 - W_{out}	137

Lista de Figuras

2.1	Representação do esquema empregado para obtenção de um modelo $M(\theta)$ de um sistema S , a partir de dados de entrada u e de saída y do sistema.	10
2.2	Representação esquemática de um processo genérico de identificação baseado na busca iterativa pelo melhor modelo.	12
2.3	Representação esquemática de um sistema de <i>Hammerstein</i>	15
2.4	Representação esquemática de um sistema de <i>Wiener</i>	15
3.1	Representação esquemática de como a aprendizagem profunda se diferencia de outras técnicas de aprendizado de máquina.	21
3.2	Representação do comportamento não linear da função de ativação ReLU.	22
3.3	Representação esquemática do funcionamento de um neurônio artificial com uma função de ativação <i>ReLU</i>	24
3.4	Representação de uma rede neural simples com dois nós de entrada, uma camada de entrada composta por dois neurônios com função de ativação <i>ReLU</i> , além de uma camada de saída constituída por um neurônio com função de ativação linear.	25
4.1	Representação gráfica de entrada e saída para o procedimento de identificação proposto por (STANISŁAWSKI; LATAWIEC, 2008)	36
4.2	Representação gráfica do equacionamento da saída estimada como uma soma de uma componente linear e outra não linear para um sistema de <i>Hammerstein</i>	36
4.3	Representação gráfica do equacionamento da saída estimada em termos de funções de bases ortonormais e funções de base radiais para um sistema de <i>Hammerstein</i>	37
4.4	Representação gráfica de como a saída de sistema <i>Wiener</i> pode ser dada em termos de funções de bases ortonormais e funções de base radiais	39
4.5	Representação da estrutura geral aplicada no identificação proposto.	40
4.6	Ilustração da estrutura de rede neural profunda empregada no método de identificação proposto.	42

5.1	Exemplo de um sinal PRBS de amplitude $a = 1$ e um sinal APBRS correspondente com amplitudes entre 0 e 1, projetado para os sistemas de primeira ordem.	56
5.2	Esquema geral utilizado para as simulações dos sistemas de <i>Hammerstein</i>	56
5.3	Esquema geral utilizado para as simulações dos sistemas de <i>Wiener</i>	57
5.4	Representação gráfica da evolução do valor da função de custo e do custo de validação da rede para as primeiras 300 épocas de um treinamento.	59
5.5	Resposta de $G_1(q)$ a um degrau unitário (Caso 1).	60
5.6	Sinal <i>multidegraus</i>	60
5.7	Gráfico comparativo entre o sinal <i>multidegraus</i> original e o resultado da aplicação de $f_H(\cdot)$	61
5.8	Comportamento não linear de $f_H(\cdot)$	61
5.9	Resposta ao degrau de $G_1(q)$ destacando se (da direita para a esquerda) o valor de regime e os tempos correspondentes a 95% e 67% desse valor.	62
5.10	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 1000 pontos de dados (Caso 1).	63
5.11	Gráfico de valores MSE para diferentes quantidades de camadas ocultas e número de neurônios por camada (Caso 1).	65
5.12	Valores de erro médio quadrático para 10 camadas ocultas e diferentes números de neurônios (Caso 1).	65
5.13	Ilustração da estrutura de rede adotada (Caso 1).	66
5.14	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 1).	67
5.15	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 1).	68
5.16	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 1).	70
5.17	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 1).	70
5.18	Resposta de $G_2(q)$ a um degrau unitário	71
5.19	Resposta ao degrau de $G_2(q)$ destacando-se os tempos utilizados para a definição do sinal de excitação.	72
5.20	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 2)	73
5.21	Gráfico de valores MSE para diferentes quantidades de camadas ocultas e número de neurônios (Caso 2).	74

5.22	Gráfico de erro para diferentes números de neurônios para os caso de 7, 8 e 9 camadas (Caso 2).	75
5.23	Ilustração da estrutura de rede adotada (Caso 2).	76
5.24	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 2).	77
5.25	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 2).	78
5.26	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 2)	79
5.27	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 2).	79
5.28	Resposta de $G_3(q)$ a um degrau unitário	80
5.29	Resposta ao degrau de $G_3(q)$ destacando-se os tempos utilizados para a definição do sinal de excitação.	81
5.30	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 3).	82
5.31	Gráfico de Valores de erro para diferentes quantidades de camadas e neurônios (Caso 3).	83
5.32	Valores de erro para diferentes números de neurônios para de 6 camadas (Caso 3)	84
5.33	Ilustração da estrutura de rede adotada (Caso 3)	85
5.34	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 3).	86
5.35	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 3).	86
5.36	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 3).	88
5.37	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 3).	88
5.38	Sinal <i>multidegraus</i>	89
5.39	Gráfico comparativo entre o sinal <i>multidegraus</i> original e o resultado da aplicação de $f_W(\cdot)$	90
5.40	Comportamento não linear de $f_W(\cdot)$	90
5.41	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 200 pontos de dados (Caso 4).	91
5.42	Gráfico de valores de erro para diferentes quantidades de camadas e neurônios (Caso 4).	93

5.43	Gráfico de valores de erro para diferentes números de neurônios por camada para 5, 6 e 7 camadas (Caso 4).	94
5.44	Ilustração da estrutura de rede adotada (Caso 4).	94
5.45	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 4).	96
5.46	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 4).	96
5.47	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 4).	98
5.48	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 4).	98
5.49	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 5).	100
5.50	Gráfico de valores de erro para diferentes quantidades de camadas e neurônios (Caso 5).	101
5.51	Ilustração da estrutura de rede adotada (Caso 5).	102
5.52	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 5).	103
5.53	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 5).	104
5.54	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ até $t = 300$ (Caso 5).	105
5.55	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 5).	105
5.56	Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 6).	107
5.57	Valores de erro para diferentes quantidades de camadas e de neurônios (Caso 6).	108
5.58	Ilustração da estrutura de rede adotada (Caso 6).	109
5.59	Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 6).	110
5.60	Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 6).	111
5.61	Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 6).	112
5.62	Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 6).	112

Capítulo 1

Introdução

1.1 Contexto

A *identificação de sistemas* diz respeito à obtenção de modelos para sistemas dinâmicos a partir de dados por meio de técnicas lineares ou não lineares. Em particular, a modelagem de sistemas dinâmicos não lineares é um problema importante em muitas áreas da ciência e engenharia, pois os modelos permitem a predição do comportamento de um sistema, fazendo com que eles sejam especialmente importantes para muitos algoritmos de controle modernos (SCHÜSSLER; MÜNKER; NELLES, 2019).

Por outro lado, a *aprendizagem profunda* (em inglês, *deep learning*), que integra o chamado *aprendizado de máquina*, é um tópico bastante presente em pesquisas atuais sobre modelagem e estimação. Assim, abordagens baseadas em aprendizagem profunda têm sido amplamente aplicadas para a modelagem de sistemas de alta complexidade em razão da sua capacidade de identificar e capturar padrões em problemas não lineares e de alta dimensão (GEDON et al., 2021). Assim como a identificação de sistemas, a aprendizagem profunda também lida com o problema fundamental de *inferir modelos a partir de dados observados*. Portanto, existem várias conexões entre esses dois tópicos. Isso possibilita que avanços importantes na área de *deep learning* possam ser empregados no desenvolvimento ou no aprimoramento de técnicas de identificação de sistemas, sobretudo no que diz respeito a sistemas não lineares (LJUNG et al., 2020).

No campo da identificação de sistemas não lineares uma abordagem amplamente utilizada trata-se da modelagem orientada a blocos, principalmente por meio de modelos de *Hammerstein* e de *Wiener* (NELLES, 2020). Esta é, portanto, uma abordagem que atrai esforços de pesquisa consideráveis nos ambientes industrial e acadêmico (STANISŁAWSKI; LATAWIEC, 2008).

Nesse contexto, propõe-se, no presente trabalho, uma estratégia de identificação que se baseia no uso de redes neurais profundas para a identificação de sistemas não lineares a

partir de estruturas de modelos orientados a blocos para os casos de *Hammerstein* e de *Wiener*.

1.2 Revisão Bibliográfica

Em (LJUNG et al., 2020) são exploradas as conexões entre *deep learning* e a área de identificação de sistemas. Como consideração inicial, é ressaltado que redes neurais profundas comuns, tais como as redes *feedforward* e *cascadeforward* são modelos ARX não lineares (NARX), e assim podem facilmente serem incorporadas em códigos e práticas de identificação de sistemas. Além disso, o uso de aprendizagem profunda, em particular para a estimação de sistema dinâmicos, tem fortes ligações com a prática de identificação de sistemas. Alguns pontos enfatizados nesta contribuição são:

- ferramentas estatísticas clássicas, como validação cruzada, compromisso de polarização/variância e análise residual também são essenciais para o aprendizado profundo;
- o fluxo de trabalho em *deep learning* e identificação de sistemas tem muitas semelhanças;
- os resultados obtidos para aplicações de identificação de sistemas são compatíveis com os melhores resultados alcançados mesmo com soluções sofisticadas que não envolvem técnicas de aprendizagem.

Em (SCHÜSSLER; MÜNKER; NELLES, 2019) também são exploradas relações entre a área de *deep learning* e abordagens convencionais de modelagem de sistemas dinâmicos. São empregadas redes neurais recorrentes profundas (*Deep Recurrent Neural Networks* - DRNNs) como um meio para a identificação de sistemas não lineares e é mostrado que modelos de espaço de estado podem ser transformados em redes neurais recorrentes e vice-versa. Os resultados obtidos com esta abordagem indicam que, se grandes quantidades de dados estiverem disponíveis, as redes neurais recorrentes padrão podem alcançar desempenhos comparáveis ao estado da arte dos métodos de identificação de sistemas convencionais.

Em (ANDERSSON et al., 2019), considera-se que os desenvolvimentos recentes em *deep learning* são relevantes para problemas de identificação de sistemas não lineares. Além disso, boa parte dos avanços nesse sentido estão associados ao chamado *sequence learning* (aprendizado sequencial). Com base nisso, foram exploradas as relações explícitas entre, a recentemente proposta, rede convolucional temporal e duas estruturas de modelo clássicas de identificação de sistema. Os resultados alcançados indicam que esta técnica têm potencial para fornecer bons resultados em problemas de identificação de sistemas, mesmo que sejam necessárias alterações relacionadas ao seu treinamento e regularização.

Já em (GEDON et al., 2021), é considerado o fato de a aprendizagem de máquina, e especial o *deep learning*, surgir como um novo padrão em muitas disciplinas para a modelagem de sistema de elevada complexidade. Com esta motivação é utilizada uma abordagem para mostrar a utilidade de modelos em espaço de estados (*State Space Models - SSMS*) profundos aplicados à identificação de sistemas não lineares. Busca-se, então, preencher a lacuna entre os dois campos de modo que um possa se beneficiar com os avanços do outro e vice-versa. Os resultados obtidos indicam que a classe de SSMS profundos representa uma alternativa competitiva em relação aos métodos clássicos de identificação, devido a sua alta flexibilidade e capacidade de capturar uma ampla gama de dinâmicas de um sistema. Características estas conferidas pelo uso de redes neurais (*neural networks - NNs*) profundas.

A abordagem apresentada em (WU; JAHANSHAH, 2019) é motivada pelos avanços frequentes na área de aprendizagem profunda. É argumentado que, contrário das técnicas convencionais de *machine learning*, esse tipo de solução geralmente se refere ao estabelecimento de rede neurais maiores e mais profundas, complementado pela disponibilidade de grandes quantidades de dados. É, então, proposto o uso de redes neurais convolucionais (Convolutional Neural Networks - CNNs) para a estimação da resposta dinâmica de sistemas lineares e não lineares de diferentes complexidades. De acordo com os resultados, a abordagem proposta, baseada em CNNs, se mostrou capaz de prever respostas estruturais com precisão e robustez ao ruído.

1.3 Objetivos

Os objetivos pretendidos neste trabalho são:

- a avaliação de técnicas de identificação de sistemas não lineares;
- a avaliação de técnicas de aprendizagem profunda aplicáveis à identificação de sistemas;
- a proposição de um esquema de identificação baseado nas técnicas estudadas;
- a aplicação da abordagem proposta para sistemas não lineares simulados a partir de modelos conhecidos;
- a avaliação do método proposto quanto à identificação dos sistemas de teste adotados;

- a comparação entre a abordagem de identificação proposta e um método da literatura que não emprega técnicas de aprendizagem.

1.4 Organização do texto

No capítulo 2 é apresentado o problema padrão de identificação de sistemas, bem como algumas formas de solução comumente empregadas. No capítulo 3 são apresentados conceitos e técnicas de aprendizado de máquina e aprendizagem profunda aplicáveis à solução de problemas de identificação de sistemas. No capítulo 4 é realizada a revisão de trabalhos que tratam da aplicação de técnicas de aprendizagem profunda para a solução de problemas de identificação de sistemas não lineares, bem como é proposto um método de identificação de sistemas não lineares utilizando aprendizagem profunda e modelagem orientada a blocos. No capítulo 5 trata-se do teste do método proposto, da análise comparativa com um método de referência e da discussão dos resultados obtidos. Por fim, no capítulo 6 é apresentada a conclusão do trabalho.

Capítulo 2

Conceitos Básicos de Identificação de Sistemas

2.1 Introdução

A identificação de sistemas trata do problema de construção de modelos matemáticos de sistemas dinâmicos com base em dados observados de um sistema de interesse. O assunto é, portanto, parte da metodologia científica básica, e uma vez que sistemas dinâmicos são abundantes em nosso meio, as técnicas de identificação de sistemas têm uma ampla área de aplicação (LJUNG, 1998).

O conceito de modelo matemático é fundamental em muitos ramos da ciência e da engenharia. Praticamente todos os sistemas em que podemos pensar podem ser descritos por um modelo matemático (BILLINGS, 2013).

Os modelos de sistemas reais são de fundamental importância em praticamente todas as disciplinas, sendo úteis para a obtenção de uma melhor compreensão acerca de algum sistema de interesse. Assim, os modelos podem ser encarados como descrições de sistemas que tornam possível prever ou simular o seu comportamento. Na engenharia, essas representações são necessários para o projeto de novos processos e para a análise dos processos já existentes. Além disso, técnicas avançadas de projeto de controladores, otimização, supervisão e detecção de falhas também são baseadas em modelos de processos (NELLES, 2020).

Em particular, tendo em vista que grande parte dos sistemas do mundo real são não lineares, o estudo e o desenvolvimento de técnicas de identificação voltadas para sistemas desse tipo são de grande importância nas mais diversas áreas de aplicação. Nesse contexto, o uso de modelos de *Hammerstein* e de *Wiener*, isto é, modelagem orientada a blocos, para a representação de sistemas não lineares é uma abordagem amplamente utilizada para dar suporte a aplicações de identificação (NELLES, 2020).

2.2 Sistemas dinâmicos

Em termos gerais, um *sistema* é um objeto no qual variáveis de diferentes tipos interagem e produzem sinais observáveis. Dentre estes sinais, aqueles que se tem interesse, são denominados de *saídas*. Existem também sinais externos manipuláveis capazes de afetar o sistema, os quais são chamados de *entradas*. Além disso, um sistema está sujeito também a perturbações. Algumas dessas podem ser medidas diretamente e outras apenas podem ser observadas mediante da sua influência na saída (LJUNG, 1998).

Geralmente, os fenômenos do mundo real são descritos por meio dos chamados *sistemas dinâmicos*, os quais fornecem o suporte matemático para a modelagem das interações entre variáveis, bem como sua evolução no tempo. Formalmente, os sistemas dinâmicos contemplam a análise, predição, e compreensão do comportamento de um sistema de equações diferenciais ou mapeamentos iterativos que descrevem a evolução do estado de um sistema. A análise de sistemas dinâmicos é realizada com alguns objetivos principais (BRUNTON; KUTZ, 2019):

- **predição de estado futuro:** em muitas áreas de aplicação busca-se obter um conhecimento prévio de estados futuros de um sistema para finalidades de planejamento, por exemplo;
- **projeto e otimização:** no que se refere ao ajuste de parâmetros de um sistema buscando uma melhoria de performance ou estabilidade;
- **interpretabilidade e compreensão física:** talvez um dos objetivos mais fundamentais dos sistemas dinâmicos seja fornecer uma visão física e permitir a interpretação do comportamento de um sistema por meio da análise de trajetórias e soluções para equações de movimento;
- **estimação e controle:** frequentemente é possível controlar ativamente um sistema dinâmico por meio de realimentação por meio de um sinal de atuação destinado a modificar o seu comportamento e, em razão disso, muitas vezes é necessário estimar o estado completo do sistema a partir de medições limitadas.

2.3 Modelos

Nos campos da ciência e engenharia objetiva-se construir modelos na forma de sistemas dinâmicos de acordo com seu uso posterior para diferentes propósitos, dentre os quais pode-se destacar os projetos de controle. Ao lidar com modelos nesta área, é possível distinguir os seguintes tipos diferentes de modelos (HOF, 2012):

- *modelos mentais*: refletem a noção intuitiva que as pessoas têm do comportamento de um sistema a exemplo de um operador humano que é capaz de controlar muito bem um sistema complicado apenas por experiência;
- *modelos de software*: dizem respeito às descrições do sistema contidas em programas de computador, incluindo possíveis regras *if-then*, mecanismos a eventos discretos e tabelas de consulta (*look-up tables*);
- *modelos gráficos*: são descrições do sistema na forma de características (não lineares) e gráficos, diagramas de Bode e de Nyquist, respostas ao impulso e ao degrau;
- *modelos matemáticos*: são descrições de um sistema na forma de relações matemáticas como, por exemplo, equações diferenciais e equações de diferenças, modelos do tipo *fuzzy* e redes neurais. Em razão do uso de parâmetros para a construção de modelos desse tipo.

Construindo modelos

Basicamente, um modelo deve ser construído a partir de dados observados. O modelo mental da dinâmica de direção de um carro, por exemplo, é desenvolvido por meio da experiência adquirida ao dirigir. Já os modelos matemáticos podem ser desenvolvidos de acordo com duas abordagens distintas (ou uma combinação delas).

Um dos caminhos consiste em dividir o sistema, figurativamente falando, em subsistemas, cujas propriedades são bem compreendidas mediante experiência prévia, isto é, trabalhos empíricos anteriores. Esses subsistemas são então unidos matematicamente e um modelo de todo o sistema é obtido. Esta rota é conhecida como *modelagem* e não envolve, necessariamente, qualquer experimentação no sistema real. Essa abordagem geralmente envolve a estruturação do processo em diagramas de blocos compostos por elementos simples. A reconstrução do sistema a partir desses blocos simples pode ser realizada via computador, resultando em um modelo de *software* ao invés de um modelo matemático.

O outro caminho, destinado à obtenção de modelos matemáticos e gráficos, é diretamente baseado na experimentação. Sinais de entrada e saída do sistema são registrados e submetidos à análise de dados para a inferência de um modelo. Este procedimento é denominado de *identificação de sistemas* (LJUNG, 1998).

2.4 Tipos de identificação

Segundo (ISERMANN; MÜNCHHOF, 2010), a descrição do comportamento dinâmico de um sistema mediante modelo pode ser realizada por meio de funções que relacionam

entrada e saída ou que expressam relações entre os estados internos. Além disso, os modelos podem também ser estabelecidos em termos de tabelas ou curvas características.

No primeiro caso, são empregadas equações onde os parâmetros do modelo são incluídos explicitamente. Já no último caso, não. Dessa forma, os modelos de sistemas dinâmicos podem, então, ser classificados em:

- *modelos paramétricos*: os quais têm estrutura definida e número finito de parâmetros;
- *modelos não paramétricos*: aqueles sem estrutura específica e com um número infinito de parâmetros.

Para o caso das técnicas de identificação não paramétricas como, por exemplo, a análise Fourier e a análise de correlação, uma certa estrutura de modelo não precisa ser assumida. Esse tipo de abordagem é geralmente utilizada para a validação de modelos derivados de considerações teóricas, uma vez que, nesta área particular de aplicação, não se tem interesse em fazer quaisquer suposições *a priori* sobre a estrutura do modelo.

Já para os métodos paramétricos, uma estrutura de modelo deve ser assumida. A escolha adequada dessa estrutura tende a proporcionar resultados de identificação mais precisos devido à maior quantidade empregada de conhecimento prévio a respeito do sistema de interesse.

O método mais simples para a obtenção de parâmetros é a determinação de valores característicos. Com base em medições de resposta ao degrau ou resposta ao impulso, os valores característicos, como o atraso de tempo, podem ser determinados.

Em abordagens mais sofisticadas, os métodos de estimativa de parâmetros têm como base equações diferenciais ou equações de diferenças de ordem arbitrária. Esse tipo de solução é voltada para a minimização de certos sinais de erro. Existe uma ampla gama de métodos para a estimação de parâmetros capazes de lidar com excitações arbitrárias e pequenas relações sinal-ruído, além de poderem ser estendidos a sistemas não lineares e para *identificação em malha fechada*.

2.5 O problema de identificação paramétrica de sistemas

A construção de um modelo a partir de dados envolve três entidades básicas:

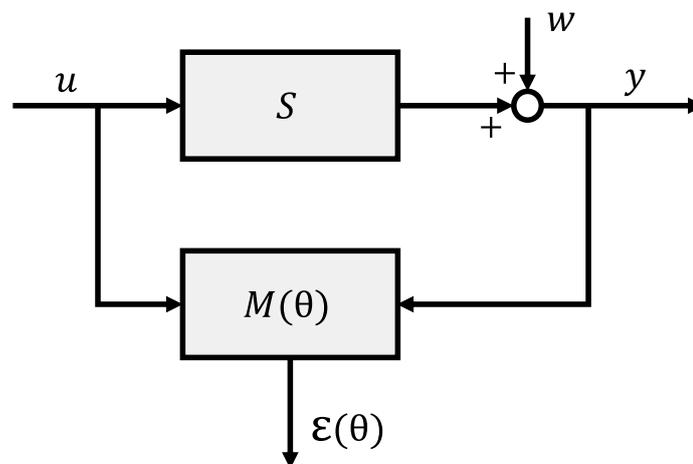
- um *conjunto de dados* de entrada e saída;
- um conjunto de modelos candidatos - uma *estrutura de modelo*;

- uma regra pela qual os modelos candidatos podem ser acessados usando os dados - um *critério de identificação*.

Considerando o diagrama apresentado na Figura 2.1, onde u e y refletem os sinais de entrada e saída, respectivamente. O sinal v representa uma perturbação não mensurável na saída. S diz respeito ao sistema a ser modelado. $M(\theta)$ reflete um conjunto parametrizado de modelos, onde o parâmetro θ pode variar ao longo de um conjunto de parâmetros $\Theta \in \mathbb{R}^d$. Para cada $\theta_1 \in \Theta$, $M(\theta_1)$ é um modelo que relaciona que relaciona os sinais de entrada e saída entre si.

Entretanto, quando um modelo é confrontado com os dados de medição (y, u) , as relações do modelo geralmente não são satisfeitas exatamente, pois o ruído nos dados impede que isso aconteça. Isso implica que, ao confrontar um modelo com dados de medição, será obtido um erro ou sinal residual. Isso é refletido na Figura 2.1 pelo sinal residual $\varepsilon(\theta)$.

Figura 2.1: Representação do esquema empregado para obtenção de um modelo $M(\theta)$ de um sistema S , a partir de dados de entrada u e de saída y do sistema.



Fonte: Adaptada de (HOF, 2012).

Nesse contexto, em um problema geral de identificação, devem ser feitas as seguintes escolhas principais (HOF, 2012):

- **Experimento:** medições de sinais de entrada e saída devem ser coletadas, levando a um conjunto de dados $\{y(t), u(t)\}_{t=1, \dots, N}$. Ao se projetar um experimento, as características do sinal de entrada u devem ser escolhidas de modo a estimular as dinâmicas relevantes do sistema S , a fim de se obter informações sobre essas dinâmicas particulares no sinal de saída medido.

- **Conjunto de modelos:** cada modelo em um conjunto específico pode, por exemplo, ser representado por uma equação a diferenças

$$y(t) + a_1y(t-1) + a_2y(t-2) = b_1u(t-1) + b_2u(t-2) + \varepsilon(t) \quad (2.1)$$

com $t = 1, 2, \dots$ e onde o vetor de parâmetros $\theta \in \mathbb{R}^4$ é dado de acordo com a equação 2.2

$$\theta = (a_1 \ a_2 \ b_1 \ b_2)^T. \quad (2.2)$$

Após a definição da quantidade de parâmetros (e, conseqüentemente, da ordem da equação de diferenças) um modelo pode ser obtido variando θ ao longo do conjunto de parâmetros $\Theta \subset \mathbb{R}^4$. De modo geral, quando da escolha do conjunto de modelos, várias propriedades básicas devem ser consideradas, como linearidade/não linearidade, variação no tempo, tempo discreto/contínuo e outras propriedades estruturais tal qual a ordem dos modelos.

- **Critério de identificação:** de posse dos dados de medição $\{y(t), u(t)\}_{t=1, \dots, N}$, para cada modelo (e, portanto, para cada parâmetro $\theta \in \Theta$) é possível gerar um sinal

$$\varepsilon(t, \theta) = y(t) + a_1y(t-1) + a_2y(t-2) - b_1u(t-1) - b_2u(t-2) \quad (2.3)$$

Este sinal residual $\varepsilon(t, \theta)$ pode ser usado como base para o ajuste do(s) modelo(s) identificado(s), e isso pode ser feito de diferentes maneiras.

A abordagem padrão consiste em construir uma função de custo para ser minimizada. A escolha mais popular é um critério de *mínimos quadrados*:

$$J_N(\theta) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (2.4)$$

a partir da qual, uma estimativa de parâmetro $\hat{\theta}_N$ é obtida ao encontrar o valor de θ que minimiza J_N :

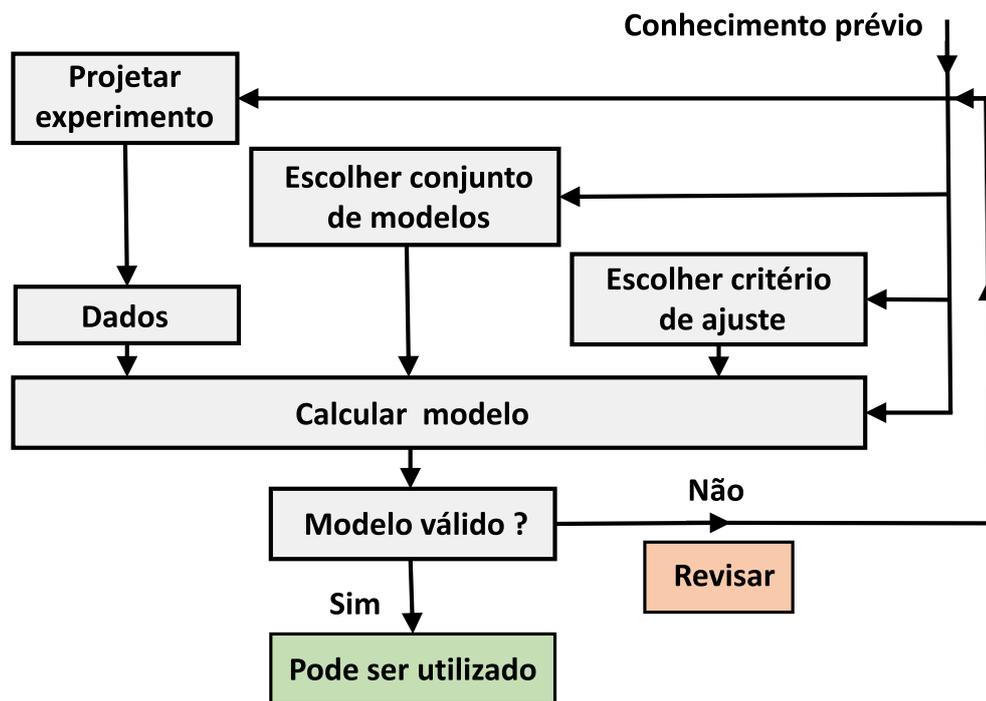
$$\hat{\theta}_N = \arg \min_{\theta \in \Theta} J_N(\theta) \quad (2.5)$$

- **Validação:** de acordo com (LJUNG, 1998), após a realização das etapas anteriores, chega-se a um modelo particular, que melhor descreve os dados de acordo com o critério escolhido. Resta, então, testar se esse modelo obtido é "bom o suficiente", isto é, se ele é válido para o seu propósito. Esses testes, que compõem a chamada *validação do modelo*, envolvem vários procedimentos para avaliar como o modelo se relaciona com os dados observados, com o conhecimento prévio e com o uso

pretendido.

Dessa forma, ainda segundo (LJUNG, 1998), o procedimento de identificação de sistemas tem um fluxo lógico natural ilustrado na Figura 2.2. Na verdade, a maior parte de uma aplicação de identificação consiste em realizar o procedimento em questão de forma iterativa, guiada por informações prévias e pelos resultados de tentativas anteriores.

Figura 2.2: Representação esquemática de um processo genérico de identificação baseado na busca iterativa pelo melhor modelo.



Fonte: Adaptada de (LJUNG, 1998).

2.6 Alguns modelos de sistemas dinâmicos

Em se tratando de sistemas dinâmicos, o modelo autorregressivo de médias móveis com entradas exógenas (*autoregressive moving average with exogenous inputs* - ARMAX), incluindo os subconjuntos de modelos AR (autorregressivo), ARMA (autorregressivo de médias móveis) e ARX (autorregressivo com entradas exógenas), são as representações mais comumente empregadas para modelos lineares voltados para a identificação de sistemas. Já os modelos não lineares autorregressivos de médias móveis com entradas exógenas (NARMAX) são as representações mais populares para identificação de sistemas dinâmicos não lineares no domínio do tempo discreto (BILLINGS, 2013).

2.6.1 Modelos lineares

Os modelos lineares são as representações mais frequentemente usadas para sistemas com estrutura inerentemente linear ou cujo comportamento pode ser caracterizado por uma aproximação linear em um ponto de operação.

A estrutura geral do modelo de sistemas lineares de ordem finita pode ser formulada como:

$$A(z^{-1})y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k) + \frac{C(z^{-1})}{D(z^{-1})}\xi(k) \quad (2.6)$$

onde

$$\begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + \dots + a_{na}z^{-na} \\ B(z^{-1}) &= b_1z^{-1} + \dots + b_{nb}z^{-nb} \\ C(z^{-1}) &= 1 + c_1z^{-1} + \dots + c_{nc}z^{-nc} \\ D(z^{-1}) &= 1 + d_1z^{-1} + \dots + d_{nd}z^{-nd} \\ F(z^{-1}) &= 1 + f_1z^{-1} + \dots + f_{nf}z^{-nf} \end{aligned} \quad (2.7)$$

$y(k)$ e $u(k)$ ($k = 1, 2, \dots$) são os sinais de entrada e de saída do sistema, respectivamente; na , nb , nc , nd , e nf são inteiros não negativos que indicam as ordens dos polinômios que compõem o modelo; $\xi(k)$ é uma sequência de ruído assumida independente e identicamente distribuída (i.i.d) com média zero e variância finita. O símbolo z^{-1} denota o operador de atraso, tal que $z^{-1} = x(k-1)$, onde x é um sinal qualquer.

Outros modelos mais simples podem ser obtidos a partir da estrutura geral da equação 2.6, com por exemplo, o modelo de resposta ao impulso finita (*finite impulse response - FIR*), o modelo ARX e o modelo ARMAX.

Modelo FIR: na equação (2.6), se $A(z^{-1}) = 1$, $F(z^{-1}) = 1$, $C(z^{-1}) = 0$, o modelo se torna FIR

$$y(k) = b_1u(k-1) + b_2u(k-2) + \dots + b_{nb}u(k-nb) \quad (2.8)$$

Modelo ARX: fazendo-se $C(z^{-1}) = D(z^{-1}) = F(z^{-1}) = 1$ é obtido

$$y(k) + a_1y(k-1) + \dots + a_{na}y(k-na) = b_1u(k-1) + \dots + b_{nb}u(k-nb) + \xi(k) \quad (2.9)$$

que corresponde ao modelo ARX **Modelo ARMAX:** de maneira semelhante, se $D(z^{-1}) = F(z^{-1}) = 1$ é obtido o modelo ARMAX

$$\begin{aligned} y(k) + a_1y(k-1) + \dots + a_{na}y(k-na) &= b_1u(k-1) + \dots + b_{nb}u(k-nb) \\ &+ \xi(k) + c_1\xi(k-1) + \dots + c_{nc}\xi(k-nc) \end{aligned} \quad (2.10)$$

2.6.2 Modelos não lineares

Para o caso de sistemas não lineares, são necessários modelos capazes de representar seus comportamentos complexos. Existem abordagens como o modelo ARMAX não linear (*Nonlinear* ARMAX - NARMAX) que são frequentemente usadas para esse tipo de problema (BILLINGS, 2013).

Modelo NARMAX: é definido como

$$\begin{aligned}
 y(k) = & F [y(k-1), y(k-2), \dots, y(k-n_y), \\
 & u(k-d), u(k-d-1), \dots, u(k-d-n_u) \\
 & e(k-1), e(k-2), \dots, e(k-n_e)] + e(k)
 \end{aligned} \tag{2.11}$$

onde $y(k)$, $u(k)$ e $e(k)$ são as sequências de saída, entrada e ruído, respectivamente; n_y , n_u , n_e são os máximos deslocamentos no tempo para os respectivos sinais; d é um atraso de tempo tipicamente definido como $d = 1$, e $F[\cdot]$ é alguma função não linear.

Os termos de ruído $e(k)$ no modelo (2.11) são normalmente definidos como os erros de predição $e(k) = y(k) - \hat{y}(k|k-1)$. Esses termos são incluídos para acomodar os efeitos do ruído de medição, erros de modelagem e/ou distúrbios não medidos.

Quanto ao mapeamento desconhecido $F[\cdot]$, existem muitos tipos estruturas de modelos disponíveis aproximá-lo, como por exemplo, modelos polinomiais em forma de potências, modelos racionais, modelos baseados em lógica *fuzzy* e, em particular, **redes neurais**.

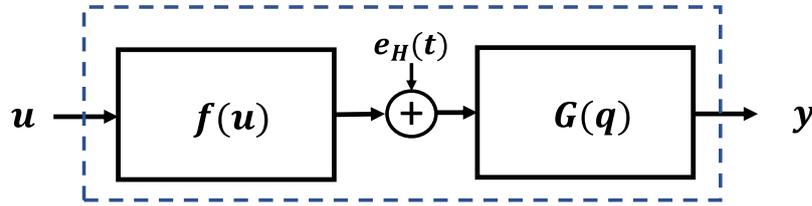
2.6.3 Modelagem orientada a blocos

Tanto na indústria quanto no ambiente acadêmico, esforços de pesquisa consideráveis têm sido direcionados para o estudo de sistemas não lineares orientados a blocos, em especial sistemas de *Hammerstein* e *Wiener*. O conceito de modelagem não linear orientada a blocos consiste, de maneira geral, na conexão entre submodelos lineares dinâmicos e blocos não lineares estáticos (STANISŁAWSKI; LATAWIEC, 2008).

No caso particular de um sistema de *Hammerstein*, cuja representação esquemática é apresentada na Figura 2.3, uma parte não linear estática é colocada em cascata com um bloco linear dinâmico.

É possível também descrever o sistema em questão por meio da equação (2.12), onde temos a entrada $u(t)$, a saída $y(t)$, o mapeamento não linear $f(u(t))$ e a função de trans-

Figura 2.3: Representação esquemática de um sistema de *Hammerstein*.



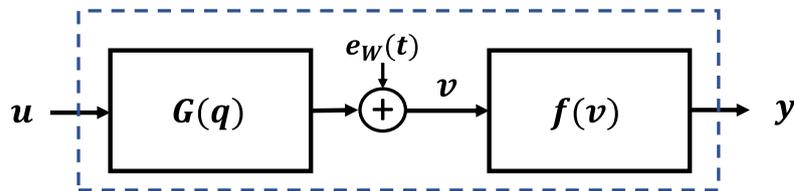
Fonte: Adaptada de (STANISLAWSKI; HUNEK; LATAWIEC, 2008).

ferência linear $G(q)$, além de um termo de erro/perturbação $e_H(t)$.

$$y(t) = G(q)[f(u(t)) + e_H(t)] \quad (2.12)$$

Já para o caso de um modelo *Wiener*, assim como representado na Figura 2.4, um bloco linear dinâmico é colocado em cascata com uma parte não linear estática.

Figura 2.4: Representação esquemática de um sistema de *Wiener*.



Fonte: Adaptada de (STANISLAWSKI; HUNEK; LATAWIEC, 2008).

É possível também descrever o este sistema utilizando a equação (2.13), onde $u(t)$ a entrada do sistema, $y(t)$ a saída, $f(\cdot)$ é o mapeamento não linear, $G(q)$ é a função de transferência linear e $e_W(t)$ é um termo de erro/perturbação .

$$y(t) = f[G(q)(u(t)) + e_W(t)] \quad (2.13)$$

Esse tipo de modelagem baseada na separação entre em blocos lineares e não lineares torna possível a aplicação de uma ampla gama de técnicas de identificação para a modelagem de cada uma das partes, permitindo, por exemplo, a aplicação de técnicas de aprendizado de máquina ou, mais especificamente, de aprendizagem profunda para a obtenção de modelos referentes a parte não linear.

2.7 Conclusão

A identificação de sistemas é um tópico clássico bem estabelecido em muitos livros-texto e que conta com uma estrutura geral e com técnicas bem definidas, mas que é alvo constante de pesquisas que buscam inovar ou aprimorar procedimentos já existentes, principalmente no que diz respeito a modelagem de sistemas não lineares utilizando-se de técnicas de *aprendizagem baseadas em dados*.

Capítulo 3

Aprendizado de Máquina e Aprendizagem Profunda

3.1 Introdução

O aprendizado de máquina (*machine learning* - ML) é uma estrutura algorítmica unificada projetada para identificar modelos computacionais que descrevam, com precisão, dados empíricos e os fenômenos associados a eles, com pouco ou nenhum envolvimento humano (WATT; BORHANI; KATSAGGELOS, 2020). Assim, *machine learning* máquina compreende o estudo de técnicas poderosas capazes de aprender a partir de experiência, isto é, o desempenho do algoritmo de aprendizagem melhora conforme ele acumula mais experiência, na forma de dados observacionais ou interações com um ambiente (ZHANG et al., 2021).

Em particular, a aprendizagem profunda (*deep learnig*, em inglês), que é um campo do aprendizado de máquina, compreende um conjunto de técnicas de aprendizagem baseada em dados capazes de alcançar grande flexibilidade de aprendizado a partir de grandes quantidades de dados e por meio de estruturas complexas ("*profundas*") como, por exemplo, as rede neurais multicamadas (BRUNTON; KUTZ, 2019).

3.2 Dados, modelos e algoritmos

Em se tratando de problemas de *machine learning*, existem alguns componentes principais a serem considerados (ZHANG et al., 2021):

1. os *dados* a partir dos quais se dará o aprendizado;
2. um *modelo* de como transformar os dados;
3. uma *função de custo* para quantificar a precisão do modelo;

4. um *algoritmo* para ajustar os parâmetros do modelo a fim de minimizar a função de custo.

3.2.1 Dados

No que diz respeito aos dados necessários à aprendizagem, geralmente, estamos preocupados com uma coleção de exemplos (também chamados de *pontos de dados*, *amostras* ou *instâncias*).

Para trabalhar com dados de maneira útil, normalmente é necessário encontrar uma representação numérica adequada. Cada exemplo normalmente consiste em uma coleção de atributos numéricos chamados de *características*. No que diz respeito à quantidade de dados, em geral, quanto mais dados estão disponíveis, mais fácil se torna o trabalho, pois neste caso, é possível treinar modelos mais poderosos e são menos necessárias suposições prévias.

Por fim, não basta apenas ter muitos dados e processá-los com inteligência. São necessários os dados certos. Se os dados contiverem muitos erros ou se as características escolhidos não possibilitarem a predição do alvo de interesse, o aprendizado irá falhar.

3.2.2 Modelos

Com relação aos modelos associados ao processo e aprendizagem, de maneira geral, o aprendizado de máquina envolve a transformação dos dados de alguma forma. Em particular, tem-se interesse em modelos estatísticos que podem ser estimados a partir de dados.

A complexidade do modelo empregado deve ser condizente com a complexidade do problema que se deseja resolver. Enquanto modelos simples são perfeitamente capazes de resolver problemas simples de maneira apropriada, situações mais complexas exigem abordagens mais elaboradas como *deep learning*. Esse tipo de solução se diferencia das abordagens convencionais pelo conjunto de modelos poderosos utilizados. Nesses casos são realizadas muitas transformações sucessivas dos dados com o intuito de conferir, aos modelos, uma maior "*profundidade*" e, conseqüentemente, uma maior capacidade de generalização dos dados. Daí advém a denominação *aprendizagem profunda*.

3.2.3 Funções objetivo

Para desenvolver um sistema matemático formal de aprendizado de máquina, são necessárias medidas formais de quão bons (ou ruins) são os modelos empregados. No aprendizado de máquina, e na otimização de maneira mais geral, essas medidas são denominadas de

funções objetivo. Por convenção, geralmente as funções objetivo são definidas de tal maneira que quanto menor seja o seu valor, melhor. Em razão disso, essas funções são normalmente chamadas de *funções de perda* ou *funções de custo*.

Ao tentar prever valores numéricos, a função objetivo mais comum é o erro quadrático, ε , dado pelo quadrado da diferença entre o valor verdadeiro y , de uma variável e o seu valor estimado, \hat{y} , de acordo com a equação (3.1).

$$\varepsilon = (y - \hat{y})^2 \quad (3.1)$$

Normalmente, a função de perda é definida em relação aos parâmetros do modelo e depende do conjunto de dados. Os melhores valores dos parâmetros do nosso modelo são aprendidos minimizando o custo obtido em um determinado conjunto de treinamento.

No entanto, ter um bom ajuste aos dados de treinamento não garante que será alcançado um bom desempenho com os dados de teste (não vistos durante o treinamento). Portanto, normalmente os dados disponíveis são divididos em duas partes:

- os *dados de treinamento*, usados para ajustar os parâmetros do modelo, e
- os *dados de teste*, que são utilizados para avaliação do modelo.

Como resultado, são obtidos o erro de treinamento e o erro de teste. Este último é o erro obtido para um conjunto de teste desconhecido. Esse valor pode variar significativamente em relação do erro de treinamento. Nesse caso, quando um modelo tem um bom desempenho sobre os dados de treinamento, mas não consegue generalizar para dados não vistos anteriormente, diz-se que ocorreu um *sobreajuste* (em inglês, *overfitting*).

3.2.4 Algoritmos de otimização

Uma vez que se tenha alguma fonte de dados, um modelo e uma função objetivo bem definidos, é preciso um algoritmo capaz de buscar os melhores parâmetros possíveis para minimizar a função de custo.

Os algoritmos de otimização mais populares para redes neurais seguem uma abordagem chamada *gradiente descendente*. Nessa abordagem, a cada etapa, é verificada, para cada parâmetro, de que maneira o custo total do conjunto de treinamento seria alterado por uma pequena variação nesse parâmetro. Dessa maneira, cada parâmetro tem seu valor atualizado segundo a direção que reduz o valor da função de custo.

3.3 Aprendizagem profunda

A aprendizagem profunda ou aprendizado profundo (em inglês, *deep learning*) é um tipo particular de aprendizado de máquina, que envolve uma quantidade maior de composições de funções aprendidas ou conceitos aprendidos do que as abordagens tradicionais de *machine learning*. Em razão disso, o *deep learning* atinge grande poder e flexibilidade ao aprender a representar um problema como uma hierarquia aninhada de conceitos, onde há a definição de conceitos complexos com base em conceitos mais simples e representações mais abstratas são computadas em termos de outras menos abstratas (GOODFELLOW; BENGIO; COURVILLE, 2016). Na Figura 3.1 é apresentada uma comparação entre a aprendizagem profunda e outros campos do aprendizado de máquina, com relação as suas estruturas básicas.

A utilidade do aprendizagem profunda já foi provada em diversas áreas de aplicação, incluindo visão computacional, processamento de áudio, processamento de linguagem natural, robótica, bioinformática e química, videogames, motores de busca, publicidade online e finanças. Esses resultados se devem, em particular, à sua ampla capacidade de lidar com situações de elevada complexidade (GOODFELLOW; BENGIO; COURVILLE, 2016).

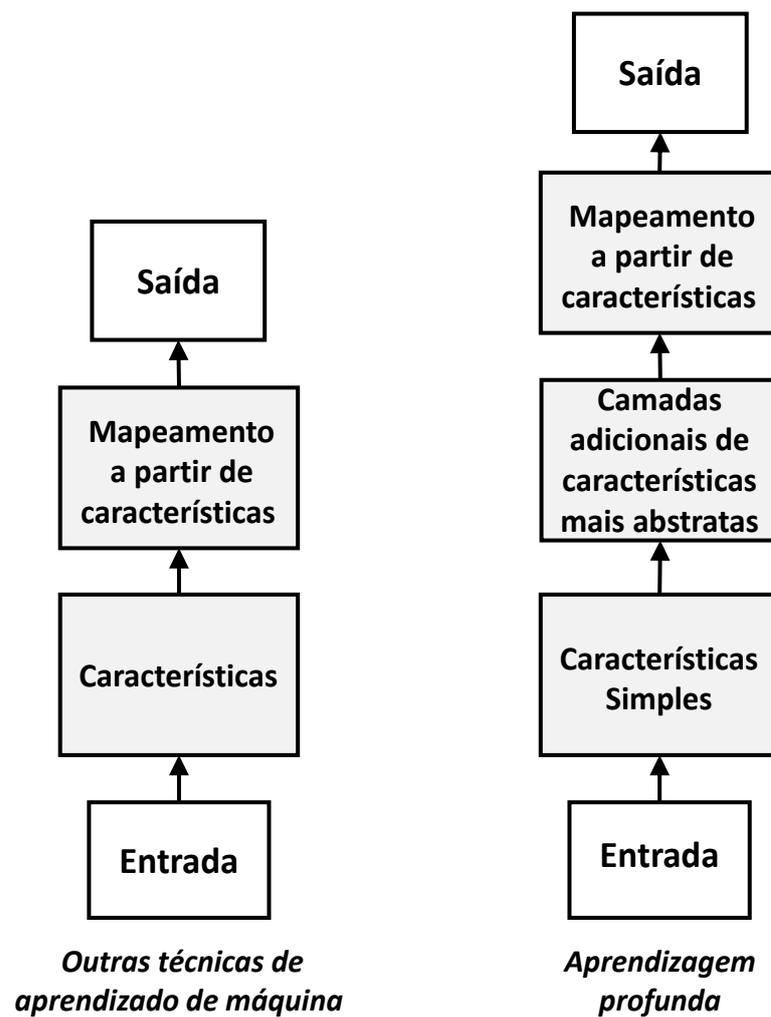
Para fins de compreensão, um exemplo de modelo de *deep learning* é a rede neural *feed-forward* profunda ou *perceptron* multicamadas (em inglês, *multilayer perceptron* - MLP). Um *perceptron* multicamadas é uma função matemática que mapeia algum conjunto de valores de entrada para valores de saída. Essa função é formada pela composição de muitas funções mais simples, onde cada aplicação de uma função matemática diferente fornece uma nova representação da entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

3.3.1 A construção de redes neurais profundas

De acordo com o exemplo apresentado em (STRANG, 2019), as peças que compõem uma estrutura de *deep learning* se juntam em uma *função de aprendizagem* $F(W, x)$ com pesos W que capturam informações dos dados de treinamento x para que essa função possa ser aplicada posteriormente sobre dados de teste. Alguns elementos importantes para a criação dessa função F são:

1. **Operação-chave** - composição: $F = F_3(F_2(F_1(W, x)))$.
2. **Regra-chave** - regra da cadeia para o cálculo das derivadas de F .
3. **Algoritmo-chave** - gradiente descendente estocástico para encontrar os melhores pesos W .

Figura 3.1: Representação esquemática de como a aprendizagem profunda se diferencia de outras técnicas de aprendizado de máquina.



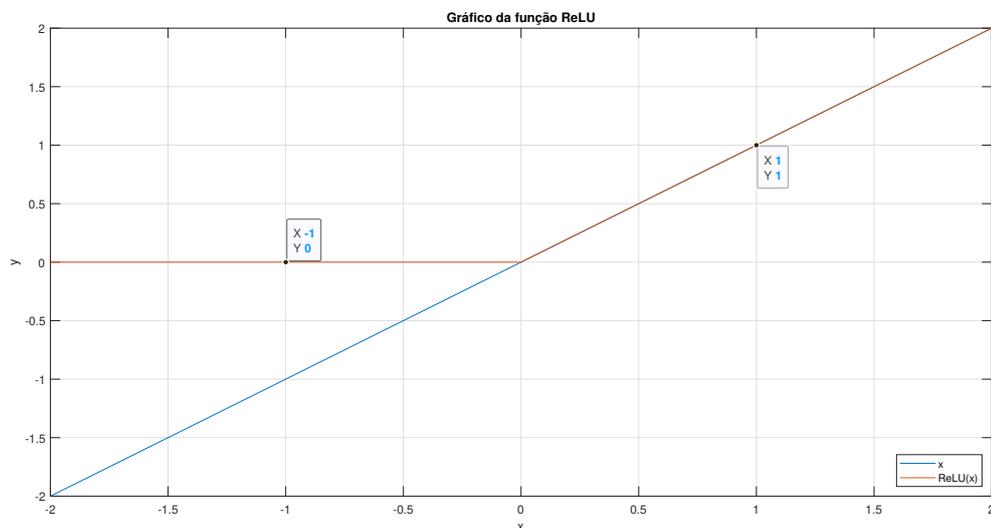
Fonte: Adaptada de (GOODFELLOW; BENGIO; COURVILLE, 2016).

4. **Sub-rotina-chave** - *backpropagation* para executar a regra da cadeia.
5. **Não linearidade-chave** - unidade linear retificada (Rectified Linear Unit - ReLU):

$$ReLU(y) = \max(y, 0)$$

O comportamento da função $ReLU$ é mostrado na Figura 3.2. Em razão da sua simplicidade e dos resultados que sua aplicação é capaz de proporcionar, essa é uma das funções de ativação mais utilizadas na construção de redes neurais profundas (STRANG, 2019). Nesse caso, para a construção da rede neural profunda, as funções F_k , com $k =$

Figura 3.2: Representação do comportamento não linear da função de ativação ReLU.



Fonte: Autoria própria.

$1, 2, \dots$, são funções contínuas lineares por parte. A escolha de funções desse tipo se dá pelos seguintes motivos:

- *linear* devido à simplicidade;
- *por partes* para que se possa alcançar, mediante composição, a não linearidade necessária para se lidar com problemas complexos;

Para a construção de uma função linear por partes de um vetor de dados x , é escolhida uma matriz de pesos A_1 e um vetor de polarizações b_1 . Em seguida é realizada uma operação não linear (mapeamento fornecido pela função ReLU), que consiste em definir como zero todos os elementos negativos de $A_1x + b_1$. O que pode ser escrito como $F_1 = (A_1x + b_1)_+$, onde esse vetor resultante forma uma *camada oculta* entre a entrada e a saída da rede.

Dessa forma, seja a função de aprendizagem F e seja cada função F_k o resultado da aplicação de uma função linear (ou afim) seguida pela aplicação da função não linear ReLU, isto é, $F_k(x) = (A_k x + b_k)_+$.

Assim, a composição dessas funções resulta na função de aprendizagem, de modo que $F(v) = F_L(F_{L-1}(\dots F_2(F_1(x))))$. Onde existem, então, $L - 1$ camadas ocultas antes da camada de saída. E a rede se torna mais profunda à medida que L aumenta.

O grande problema de otimização em se tratando de *deep learning* é computar os pesos A_k e b_k que farão a saída $F(x)$ ser próxima de uma referência $R(x)$.

O problema de minimizar alguma medida da diferença $F(x) - R(x)$ é resolvido por meio de gradientes descendentes. O gradiente, por sua vez, é calculado utilizando-se *backpropagation*, que executa a regra da cadeia.

Com relação aos pesos da rede, o algoritmo de aprendizagem deve, então, encontrar uma função F com resultados próximos a uma referência não apenas para as amostras de treinamento, mas também para amostras de teste posteriores. Caso a função F obtida seja extremamente precisa em relação ao conjunto de treinamento, mas fora de controle com os dados de teste, diz-se que houve um *sobreajuste*, isto é, foi produzida uma função que não consegue generalizar os dados.

3.3.2 Treinamento de redes neurais

De acordo com (BRUNTON; KUTZ, 2019), O treinamento de redes neurais, de modo geral, é computacionalmente custoso. Mesmo redes de tamanho modesto podem se gerar um custo de treinamento proibitivo se as rotinas de otimização usadas para treinamento não forem adequadas. Dois algoritmos são especialmente importantes para viabilizar o treinamento de redes neurais e, principalmente, redes neurais profundas: o *backpropagation* e o *gradiente descendente estocástico* (Stochastic Gradient Descent - SGD). O *backpropagation* (retropropagação) permite um cálculo eficiente do gradiente da função de custo, enquanto o SGD fornece uma definição mais rápida dos pesos ideais da rede.

Backpropagation

Para o treinamento de uma rede neural, é necessário que os pesos que a compõem sejam atualizados ao longo do tempo. Dado que a rede é composta por um conjunto de pesos \mathbf{W} e tem uma função de composição F , então esta rede é capaz de fornecer uma saída \hat{Y} para uma dada entrada x . Sabendo que o valor verdadeiro esperado para uma entrada x é um valor de referência Y , então, é possível mensurar o quão precisa é a previsão da rede por meio de uma função de custo J definida a partir da diferença entre Y e \hat{Y} . Dessa forma, adotando, por exemplo, a função de custo dada pela equação (3.2),

$$J = \frac{1}{2}[Y - \hat{Y}]^2 \quad (3.2)$$

a atualização dos pesos na iteração de treinamento $[k + 1]$ em relação à iteração $[k]$ será dada de acordo com a equação (3.3), utilizando-se o gradiente descendente.

$$\mathbf{W}[k + 1] = \mathbf{W}[k] - \alpha \nabla J \quad (3.3)$$

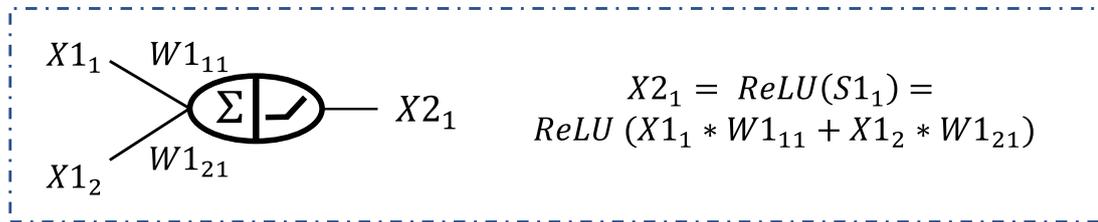
onde α é taxa de aprendizagem e ∇J é o gradiente da função de custo.

Dessa forma, para a atualização de um peso específico W_{ij} , então tem-se que

$$W_{ij}[k + 1] = W_{ij}[k] - \alpha \frac{\partial J}{\partial W_{ij}} \quad (3.4)$$

Do ponto de vista construtivos de uma rede neural, tem-se como unidades elementares, os *neurônios artificiais*. Um exemplo neurônio artificial é lustrado na Figura 3.3,

Figura 3.3: Representação esquemática do funcionamento de um neurônio artificial com uma função de ativação *ReLU*



Fonte: Autoria própria.

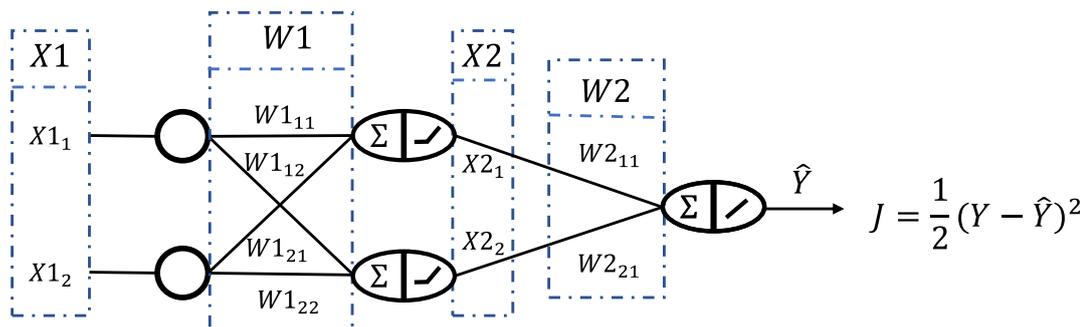
Nessa representação, a parte esquerda a representação elíptica do neurônio artificial se refere à operação de soma das entradas ponderadas pelos seus respectivos pesos. Já a parte direita da representação diz respeito à aplicação de uma função de ativação, neste caso, a função *ReLU*. Dessa maneira,

- $X1_1$ é a entrada número 1 do conjunto de entradas $X1$ que é referente à camada 1;
- $X1_2$ é a entrada número 2 do conjunto de entradas $X1$;
- o peso $W1_{11}$ pertence ao conjunto de pesos $W1$ referente à camada 1 e, de acordo com o subscrito 11, liga a entrada 1 ao neurônio 1;

- de maneira análoga, o peso W_{121} pertence ao conjunto de pesos $W1$ e liga a entrada 2 ao neurônio 1;
- a soma $S_{11} = W_{111} * X_{11} + W_{121} * X_{12}$ é referente a camada 1 e o subscrito 1 indica que ela é realizada pelo primeiro neurônio da camada;
- por fim, $X_{21} = ReLU(S_{11})$ é a saída do neurônio em questão, a qual vai compor o conjunto de entradas $X2$ da camada subsequente.

Com base no exemplo de neurônio artificial mostrado, é possível construir estrutura neural simples, tal qual ilustrado na Figura 3.4, com apenas uma camada de entrada e uma camada de saída, a fim de exemplificar o funcionamento do *backpropagation*.

Figura 3.4: Representação de uma rede neural simples com dois nós de entrada, uma camada de entrada composta por dois neurônios com função de ativação *ReLU*, além de uma camada de saída constituída por um neurônio com função de ativação linear.



Fonte: Autoria própria.

Nesta representação,

- $X1$ é o vetor de entradas da primeira camada da rede (camada de entrada);
- $X2$ é o vetor de entradas da segunda camada (camada de saída);
- $W1$ é a matriz de pesos associada à camada de entrada;
- $W2$ é a matriz de pesos referente à camada de saída;
- \hat{Y} é a saída fornecida pela rede; e
- J é a função de custo definida para se realizar o treinamento.

Dessa maneira, fornecido um determinado conjunto de entradas para a rede, então, percorrendo a rede desde a entrada em direção a saída, tem-se que

$$X2_1 = ReLu(S1_1) = ReLu(W1_{11} \times X1_1 + W1_{21} \times X1_2) \quad (3.5)$$

$$X2_2 = ReLu(S1_2) = ReLu(W1_{12} \times X1_1 + W1_{22} \times X1_2) \quad (3.6)$$

e, por fim,

$$\hat{Y} = S_{\hat{Y}} = (W2_{11} \times X2_1 + W2_{21} \times X2_2) \quad (3.7)$$

Onde na equação (3.6), por simplicidade foi utilizada uma função de saída linear do tipo $f(x) = x$, daí $\hat{Y} = S_{\hat{Y}}$. De posse do valor de saída da rede, \hat{Y} , é possível calcular de custo a partir da equação (3.8), de modo que

$$J = \frac{1}{2}(Y - \hat{Y})^2 \quad (3.8)$$

Assim, o processo realizado neste estágio consiste em percorrer a estrutura da rede realizando as multiplicações dos dados de entrada pelos respectivos pesos, bem como a realização das somas dos valores ponderados e da aplicação das funções de ativação para cada camada da rede, partindo da entrada e indo em direção a saída, isto é "indo para frente".

Entretanto, tendo em vista a necessidade de se ajustar os pesos da rede visando a minimização da diferença entre a saída obtida por meio da rede e o valor verdadeiro da saída, é preciso que se tenha informação a respeito de como o valor de cada peso deve ser alterado para se minimizar o erro em questão. Para o conjunto de pesos como um todo, as informações de como eles devem ser ajustados são fornecidas pelo gradiente da função de custo J .

Dessa forma, para a realização do ajuste de cada peso específico, é necessário o cálculo, a cada iteração do processo de treinamento, do valor da derivada parcial da função de custo em relação a cada peso em particular. É justamente para o cálculo dessas derivadas que é aplicada a estratégia de *backpropagation*, que consiste em percorrer a rede neural no sentido reverso, isto é, desde a saída até a entrada, realizando ao longo do caminho o cálculo da influencia de cada peso no valor total da função de custo. O algoritmo de *backpropagation* é baseado na possibilidade de decompor o cálculo de uma determinada derivada em cálculos de derivadas intermediárias utilizando-se a *regra da cadeia*.

Considerando-se, por exemplo, a situação em que a função de custo J depende de $\hat{Y} = S_{\hat{Y}} = (W2_{11} \times X2_1 + W2_{21} \times X2_2)$, então, para o cálculo do termo $\frac{\partial J}{\partial W2_{11}}$, é possível, a partir da regra da cadeia, decompor a derivada parcial em questão como mostrado na

equação (3.9)

$$\frac{\partial J}{\partial W_{211}} = \frac{\partial J}{\partial \hat{Y}} \times \frac{\partial \hat{Y}}{\partial S_{\hat{Y}}} \times \frac{\partial S_{\hat{Y}}}{\partial W_{211}} \quad (3.9)$$

Tomando como base esta decomposição é possível então dividir o cálculo da derivada parcial de interesse em três passos mais simples, como mostrado nas equações a seguir.

Adotando a função de custo J como sendo

$$J = \frac{1}{2}(Y - \hat{Y})^2 \quad (3.10)$$

então

$$\frac{\partial J}{\partial \hat{Y}} = 2 \times \frac{1}{2} \times (Y - \hat{Y}) \frac{\partial(Y - \hat{Y})}{\partial \hat{Y}} \quad (3.11)$$

$$\frac{\partial J}{\partial \hat{Y}} = (Y - \hat{Y}) \times (-1) \quad (3.12)$$

logo, para o termo $\frac{\partial J}{\partial \hat{Y}}$, tem-se que

$$\frac{\partial J}{\partial \hat{Y}} = (\hat{Y} - Y) \quad (3.13)$$

Já com relação a $\frac{\partial \hat{Y}}{\partial S_{\hat{Y}}}$:

$$\hat{Y} = S_{\hat{Y}} = (W_{211} \times X_{21} + W_{221} \times X_{22}) \quad (3.14)$$

$$\frac{\partial \hat{Y}}{\partial S_{\hat{Y}}} = 1 \quad (3.15)$$

E para o termo $\frac{\partial S_{\hat{Y}}}{\partial W_{211}}$:

$$\frac{\partial S_{\hat{Y}}}{\partial W_{211}} = \frac{\partial(W_{211} \times X_{21} + W_{221} \times X_{22})}{\partial W_{211}} = X_{21} \quad (3.16)$$

Assim,

$$\frac{\partial J}{\partial W_{211}} = \frac{\partial J}{\partial \hat{Y}} \times \frac{\partial \hat{Y}}{\partial S_{\hat{Y}}} \times \frac{\partial S_{\hat{Y}}}{\partial W_{211}} \quad (3.17)$$

$$\frac{\partial J}{\partial W_{211}} = (\hat{Y} - Y) \times 1 \times X_{21} \quad (3.18)$$

Esse conjunto de passos exemplifica como o algoritmo de *backpropagation* possibilita o cálculo das influências dos pesos no custo total da rede a partir de cálculos mais simples de como os resultados de uma camada são influenciados pelos resultados das camadas imediatamente anteriores.

De maneira análoga, para o cálculo de $\frac{\partial J}{\partial W_{221}}$, é possível obter a formulação mostrada na equação (3.19).

$$\frac{\partial J}{\partial W_{221}} = (\hat{Y} - Y) \times 1 \times X_{22} \quad (3.19)$$

onde $X_{21} = ReLu(S_{21}) = ReLu(W_{111} \times X_{11} + W_{121} \times X_{12})$.

À medida que se avança em direção à entrada da rede, é realizada a continuação da aplicação do algoritmo para toda a rede. Para os pesos referentes a camada de entrada tem-se, por exemplo, para o cálculo do termo $\frac{\partial J}{\partial W_{111}}$:

$$\frac{\partial J}{\partial W_{111}} = \frac{\partial J}{\partial \hat{Y}} \times \frac{\partial \hat{Y}}{\partial S_{\hat{Y}}} \times \frac{\partial S_{\hat{Y}}}{\partial X_{21}} \times \frac{\partial X_{21}}{\partial S_{21}} \times \frac{\partial S_{21}}{\partial W_{111}} \quad (3.20)$$

Entretanto, há que se considerar que o cálculo da parcela $\frac{\partial J}{\partial \hat{Y}} \times \frac{\partial \hat{Y}}{\partial S_{\hat{Y}}}$ já foi realizada aplicação do algoritmo para os pesos da camada de saída. Dessa forma, é preciso apenas calcular os termos $\frac{\partial S_{\hat{Y}}}{\partial X_{21}}$, $\frac{\partial X_{21}}{\partial S_{21}}$ e $\frac{\partial S_{21}}{\partial W_{111}}$. Este que é um dos grandes benefícios proporcionados pela algoritmo de *backpropagation* é possível justamente pela abordagem de se percorrer a rede indo da saída para a entrada realizando a atualização dos pesos camada após camada.

Com relação ao termo $\frac{\partial S_{\hat{Y}}}{\partial X_{21}}$, como

$$S_{\hat{Y}} = (W_{211} \times X_{21} + W_{221} \times X_{22}) \quad (3.21)$$

então,

$$\frac{\partial S_{\hat{Y}}}{\partial X_{21}} = W_{211} \quad (3.22)$$

Já com relação à derivada $\frac{\partial X_{21}}{\partial S_{21}}$, sabendo que

$$X_{21} = ReLu(S_{21}) = ReLu(W_{111} \times X_{11} + W_{121} \times X_{12}) \quad (3.23)$$

logo,

$$\frac{\partial X_{21}}{\partial S_{21}} = \frac{\partial ReLu(S_{21})}{\partial S_{21}} = ReLu'(S_{21}) \quad (3.24)$$

de tal maneira que, $ReLu'(S_{21}) = S_{21}$, se $S_{21} > 0$ e $ReLu'(S_{21}) = 0$, se $S_{21} \leq 0$

por fim, para o termo $\frac{\partial S_{21}}{\partial W_{111}}$, tem-se

$$S_{21} = (W_{111} \times X_{11} + W_{121} \times X_{12}) \quad (3.25)$$

$$\frac{\partial S_{21}}{\partial W_{111}} = \frac{\partial (W_{111} \times X_{11} + W_{121} \times X_{12})}{\partial W_{111}} = X_{11} \quad (3.26)$$

Desse modo, como

$$\frac{\partial J}{\partial W_{111}} = \frac{\partial J}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial S_{\hat{y}}} \times \frac{\partial S_{\hat{y}}}{\partial X_{21}} \times \frac{\partial X_{21}}{\partial S_{21}} \times \frac{\partial S_{21}}{\partial W_{111}} \quad (3.27)$$

então,

$$\frac{\partial J}{\partial W_{111}} = (\hat{Y} - Y) \times 1 \times (W_{211}) \times (ReLU'(S_{21})) \times (X_{11}) \quad (3.28)$$

De maneira semelhante podemos obter os outros pesos W_{1ij} :

$$\frac{\partial J}{\partial W_{111}} = (\hat{Y} - Y) \times 1 \times (W_{211}) \times (ReLU'(S_{21})) \times (X_{11}) \quad (3.29)$$

$$\frac{\partial J}{\partial W_{112}} = (\hat{Y} - Y) \times 1 \times (W_{221}) \times (ReLU'(S_{22})) \times (X_{11}) \quad (3.30)$$

$$\frac{\partial J}{\partial W_{121}} = (\hat{Y} - Y) \times 1 \times (W_{211}) \times (ReLU'(S_{21})) \times (X_{12}) \quad (3.31)$$

$$\frac{\partial J}{\partial W_{122}} = (\hat{Y} - Y) \times 1 \times (W_{221}) \times (ReLU'(S_{22})) \times (X_{12}) \quad (3.32)$$

Assim como exemplificado para a estrutura neural simples adotada, o algoritmo back-propagation pode ser aplicado para redes múltiplas camadas utilizando-se a mesma lógica mostrada, de modo a proporcionar o cálculo de gradientes de uma maneira simples e pouco custosa computacionalmente (GOODFELLOW; BENGIO; COURVILLE, 2016). Além disso, à medida que torna-se maior o número de entradas, de camadas e de pesos de uma rede, isto é, à medida que as redes se tornam mais complexas, os ganhos de eficiência no processo de treinamento proporcionados pelo *backpropagation* tornam-se mais significativos.

Gradiente descendente estocástico

De acordo com (STRANG, 2019), o conceito de gradiente descendente é fundamental no treinamento de uma rede neural profunda. De modo geral, é estabelecido um passo de atualização do conjunto de pesos W da rede de acordo com a equação (3.33)

$$W[k+1] = W[k] - \alpha \nabla \mathbf{L}(W[k]) \quad (3.33)$$

Esse abordagem deve nos levar ao conjunto de pesos W^* para o qual a função de perda $L(W)$ é minimizada para os dados de teste x . No entanto, para redes grandes com muitas amostras no conjunto de treinamento, esse algoritmo, isto é, o gradiente descendente clássico apresenta dois problemas principais:

1. Calcular $\nabla \mathbf{L}$ em cada passo de descida - as derivadas da perda total \mathbf{L} em relação

a todos os pesos W na rede - é muito custoso. Essa perda total é o resultado da soma das perdas individuais $\ell(W, x_i)$ para cada componente x_i no conjunto de treinamento, de modo que pode ser necessário o cálculo de milhões de perdas separadas para serem somadas resultando no custo total \mathbf{L} a cada passo de treinamento.

2. Para um número de pesos muito grande, é possível que $\nabla \mathbf{L} = 0$ para muitas escolhas diferentes W^* dos pesos, em razão da possibilidade de existência de muitos mínimos locais. Algumas dessas escolhas podem dar resultados ruins em dados de teste não vistos no treinamento. A função de aprendizagem F pode, então, falhar em "generalizar". Mas ao se utilizar o chamado *gradiente descendente estocástico* (*Stochastic Gradient Descent* - SGD), de maneira geral, são encontrados pesos W^* capazes de generalizar o aprendizado para vetores de dados não vistos x de uma população semelhante a de treinamento.

Para a execução do algoritmo do gradiente descendente estocástico é, então, usado apenas um "*mini lote*" dos dados de treinamento em cada etapa. Para a composição desse subconjunto adotado em cada etapa, são escolhidos aleatoriamente B elementos do conjunto total de treinamento de tamanho n . A substituição do lote completo de todos os dados de treinamento por um mini lote altera $\mathbf{L}(W) = \frac{1}{n} \sum_{i=1}^n \ell_i(W)$ para um soma de apenas B perdas. Esta estratégia resolve ambas as dificuldades mencionadas anteriormente. Dessa maneira, o sucesso do aprendizado profundo quanto ao uso do SGD se baseia em dois fatos principais:

1. O cálculo de $\nabla \ell_1$ usando *backpropagation* para B amostras é muito mais rápido. Frequentemente $B = 1$.
2. O algoritmo estocástico produz pesos W^* que também são bem-sucedidos quando aplicados a dados não vistos.

Um ponto importante a se destacar é que não se deve ajustar os dados de treinamento com perfeição. Isso muitas vezes ocasiona o chamado sobreajuste ou *overfitting*. Neste caso, a função F torna-se hipersensível e ao invés de aprender com os dados de treinamento, eles foram "memorizados", o que conduz a um mal desempenho frente a dados não vistos no treinamento. A generalização obtida por meio do SGD é a capacidade de fornecer a classificação correta para dados de teste não vistos x_t , com base nos pesos W que foram aprendidos com os dados de treinamento x .

No que se refere a evitar o *overfitting*, uma estratégia fundamental no treinamento de uma rede neural (o que significa encontrar uma função que aprenda com os dados de treinamento e generalize bem para os dados de teste) é o realizar o chamado *early stopping*, ou, em uma tradução livre, parar cedo. Esta estratégia consiste, de maneira geral, em finalizar o processo de treinamento antes que o ajuste dos pesos ocasione *overfitting*.

3.4 Conclusão

Tendo em vista as semelhanças entre o procedimento geral de identificação de sistemas e a estrutura das soluções de *machine learning* e, considerando também os avanços que têm sido alcançados no campo da aprendizagem profunda com relação à capacidade de resolver problemas altamente não lineares e de elevada complexidade, é bastante razoável a investigação da possibilidade de aplicação desse tipo de técnica à identificação de sistemas não lineares de maneira ampla.

Capítulo 4

Identificação, *Deep Learning* e Modelagem Orientada a Blocos

4.1 Introdução

A aplicação de técnicas de aprendizagem profunda têm sido bastante difundida em diversas áreas de pesquisa. Em especial, resultados importantes têm sido alcançados, principalmente no que se refere à obtenção de modelos apropriados associados a dados complexos (LJUNG et al., 2020). Dessa forma, tendo em vista as conexões entre as áreas de identificação de sistemas e de aprendizagem baseada em dados, várias pesquisas têm sido desenvolvidas no sentido de empregar técnicas de aprendizagem profunda para a modelagem de sistemas dinâmicos e, em especial, para a identificação de sistemas não lineares.

Desta forma, busca-se incorporar avanços da área de aprendizagem profunda em estratégias de identificação de sistemas. Para isto, é proposta a aplicação de redes neurais profundas em um esquema de identificação baseado na abordagem de modelagem não linear orientada a blocos apresentada em (STANISŁAWSKI; LATAWIEC, 2008).

4.2 Identificação com base em modelos orientados a blocos

Nos trabalhos de (LATAWIEC et al., 2003), (STANISŁAWSKI; HUNEK; LATAWIEC, 2008) e (STANISŁAWSKI; LATAWIEC, 2008) são propostos esquemas de identificação baseados na separação da partes linear e não linear de sistemas com modelos orientados a blocos. Nestas abordagens, a parte linear é aproximada por funções de base ortonormal (*Orthonormal Basis Functions* - OBF) e para a aproximação da parte não linear são utilizadas funções de base radiais (*Radial Basis Functions* - RBF).

Mais detalhadamente, de acordo com (STANISLAWSKI; HUNEK; LATAWIEC, 2008), considerando um sistema linear de tempo discreto com uma entrada $u(t)$, uma saída $y(t)$ e que tem seu comportamento descrito por uma função de transferência $G(q)$, tal que $y(t) = G(q)u(t)$, é possível obter uma função $\hat{G}(q)$ como uma estimativa para $G(q)$. Dessa forma, uma saída estimada $\hat{y}(t)$ pode ser obtida como expresso na equação (4.1).

$$\hat{y}(t) = \hat{G}(q)u(t) \quad (4.1)$$

A função $\hat{G}(q)$, por sua vez, pode ser escrita utilizando funções de bases ortonormais (*Orthonormal Basis Functions* - OBF), de modo que

$$\hat{G}(q) = \sum_{i=1}^M c_i B_i(q, p) \quad (4.2)$$

onde B_i , $i = 1, 2, 3, \dots$ são funções de uma base ortonormal e c_i , $i = 1, 2, 3, \dots$ são coeficientes de ponderação. Dessa maneira, $\hat{y}(t)$ pode ser obtido de acordo com a equação (4.3).

$$\hat{y}(t) = \sum_{i=1}^M c_i B_i(q, p) u(t) \quad (4.3)$$

Essa abordagem tem como base a chamada OBF Regular, porém em situações onde se deseja estimar uma função inversa, como por exemplo, $\hat{G}^{-1}(q)$, é possível usar o conceito de OBF Inversa introduzido no trabalho de (STANISLAWSKI; LATAWIEC, 2008).

No caso da OBF Inversa, considerando que $\hat{G}^{-1}(q)$ existe, é possível reescrever a equação (4.1) como:

$$\hat{G}^{-1}(q)\hat{y}(t) = u(t) \quad (4.4)$$

onde $\hat{G}^{-1}(q)$ pode ser escrita na forma de uma resposta finita ao impulso com N termos. E, considerando-se que o sistema tem um atraso conhecido d , é possível escrever $\hat{G}^{-1}(q)$ como mostrado nas equações (4.5) e (4.6).

$$\hat{G}^{-1}(q) = q^d(\alpha_0 + \alpha_1 q^{-1} + \dots + \alpha_d q^{-d} + \dots + \alpha_{N-1} q^{-(N-1)}) \quad (4.5)$$

$$\hat{G}^{-1}(q) = \alpha_0 q^d + \alpha_1 q^{d-1} + \dots + \alpha_d q^0 + \dots + \alpha_{N-1} q^{-(N-1)+d} \quad (4.6)$$

É possível, ainda, reescrever a equação (4.6) da seguinte forma:

$$\frac{\hat{G}^{-1}(q)}{\alpha_0 q^d} = 1 + \beta_1 q^{-1} + \dots + \beta_d q^{-d} + \dots + \beta_{N-1} q^{-(N-1)} \quad (4.7)$$

sendo $\beta_i = \frac{\alpha_i}{\alpha_0}$, com $i = 1, 2, \dots, N - 1$

Dividindo ambos os membros da equação (4.4) por $\alpha_0 q^d$, tem-se que

$$\frac{\hat{G}^{-1}(q)}{\alpha_0 q^d} \hat{y}(t) = \frac{u(t)}{\alpha_0 q^d} \quad (4.8)$$

Reescrevendo-se a equação (4.8) em termos da representação de $\hat{G}^{-1}(q)$ na forma de uma resposta finita ao impulso, tem-se

$$\hat{y}(t)(1 + \beta_1 q^{-1} + \dots + \beta_d q^{-d} + \dots + \beta_{N-1} q^{-(N-1)}) = \beta_0 u(t) q^{-d}, \quad \beta_0 = \frac{1}{\alpha_0} \quad (4.9)$$

$$\hat{y}(t) + \beta_1 \hat{y}(t) q^{-1} + \dots + \beta_d \hat{y}(t) q^{-d} + \dots + \hat{y}(t) \beta_{N-1} q^{-(N-1)} = \beta_0 u(t) q^{-d} \quad (4.10)$$

onde $\beta_1 \hat{y}(t) q^{-1} + \dots + \beta_d \hat{y}(t) q^{-d} + \dots + \hat{y}(t) \beta_{N-1} q^{-(N-1)}$ pode ser aproximada por $\sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t)$, com $L_i(q^{-1}, p)$, $i = 1, 2, 3, \dots$ sendo funções de base ortonormal, por exemplo, funções de *Laguerre* e *Kautz*. Dessa forma,

$$\hat{y}(t) + \sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t) = \beta_0 u(t) q^{-d} \quad (4.11)$$

e, finalmente,

$$\hat{y}(t) + \sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t) = \beta_0 u(t - d) \quad (4.12)$$

Sistemas de *Hammerstein*

Considerando-se agora um modelo de *Hammerstein* representado pela equação (4.13)

$$y(t) = G(q)[f(u(t)) + e_H(t)] = G(q)[x(t) + e_H(t)] \quad (4.13)$$

e, de maneira aproximada,

$$\hat{y}(t) = \hat{G}(q)[f(u(t))] \quad (4.14)$$

pode-se, então, escrever

$$\hat{G}^{-1}(q) \hat{y}(t) = f(u(t)) \quad (4.15)$$

Conforme o procedimento de aproximação de $\hat{G}^{-1}(q)$ por meio de funções de base ortonormal mostrado anteriormente, é possível escrever

$$\hat{y}(t) + \sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t) = \beta_0 f(u(t)) q^{-d} \quad (4.16)$$

Como $f(\cdot)$ é uma não linearidade estática, isto é, não varia com o tempo, então

$$\hat{y}(t) + \sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t) = \beta_0 f(u(t-d)) \quad (4.17)$$

Em (STANISŁAWSKI; LATAWIEC, 2008) e (STANISŁAWSKI; HUNEK; LATAWIEC, 2008) são utilizadas funções de *Laquerre* para a aproximação da parte linear via OBF, de modo que cada função $L_i(q^{-1}, p)$ é dada por:

$$L_i(q, p) = \frac{\sqrt{1-p^2}}{q-p} \left[\frac{1-pq}{q-p} \right]^{i-1} \quad i = 1, \dots, M \quad (4.18)$$

Essas funções são caracterizadas pela dinâmica dominante do sistema, a qual é dada por um único polo real p que pode ser selecionado de forma experimental.

Nos trabalhos em questão, é realizada também a aproximação da parte não linear por meio de uma rede de funções de base radiais (*Radial Basis Functions Network* - RBFN). A saída dessa rede é dada pela equação 4.19:

$$x(t) = \sum_{j=1}^m w_j \phi_j(u(t)) \quad (4.19)$$

onde $w_j, j = 1, \dots, m$ são coeficientes de ponderação e $\phi_j(u(t)), j = 1, \dots, m$ são as saídas de cada nó da rede. As funções $\phi_j(\cdot), j = 1, \dots, m$ são mapeamentos não lineares. Neste caso foram adotadas funções gaussianas, de modo que $\phi_j(u(t))$ é dada por

$$\phi_j(u(t)) = \exp(-\|u(t) - \alpha_j\|^2 / \sigma_j^2), \quad j = 1, \dots, m \quad (4.20)$$

Dessa forma, realizando a aproximação da parte linear via OBF e aproximando a parte não linear por meio de uma rede RBF, é possível escrever

$$\hat{y}(t) = - \sum_{i=1}^M c_i L_i(q^{-1}, p) \hat{y}(t) + \beta_0 \sum_{j=1}^m w_j \phi_j(u(t-d)) \quad (4.21)$$

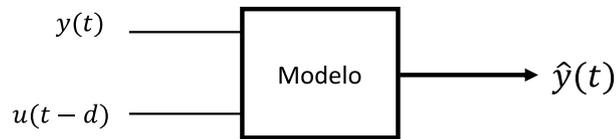
Assumindo que $\underline{w}_j = \beta_0 w_j, j = 1, \dots, m$ e utilizando valores de $y(t)$ na aproximação da parte linear, a saída do sistema de *Hammerstein* é dado finalmente por

$$\hat{y}(t) = - \sum_{i=1}^M c_i L_i(q^{-1}, p) y(t) + \sum_{j=1}^m \underline{w}_j \phi_j(t-d) \quad (4.22)$$

Dessa maneira, assim como mostrado na Figura 4.1, o método de identificação em questão tem por objetivo a obtenção de um modelo que forneça um valor estimado $\hat{y}(t)$ da saída a partir de informação de entrada, $u(t-d)$, e saída, $y(t)$, do sistema a ser identificado.

Para a obtenção do modelo é realizada uma formulação que expresse a saída estimada

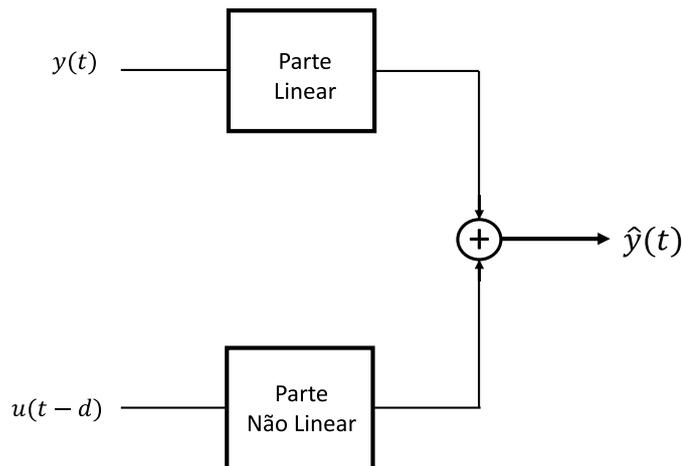
Figura 4.1: Representação gráfica de entrada e saída para o procedimento de identificação proposto por (STANISŁAWSKI; LATAWIEC, 2008)



Fonte: Autoria própria.

$\hat{y}(t)$ simplesmente como uma soma de uma parte linear com uma parte não linear, tal qual é mostrado na Figura 4.2

Figura 4.2: Representação gráfica do equacionamento da saída estimada como uma soma de uma componente linear e outra não linear para um sistema de *Hammerstein*.



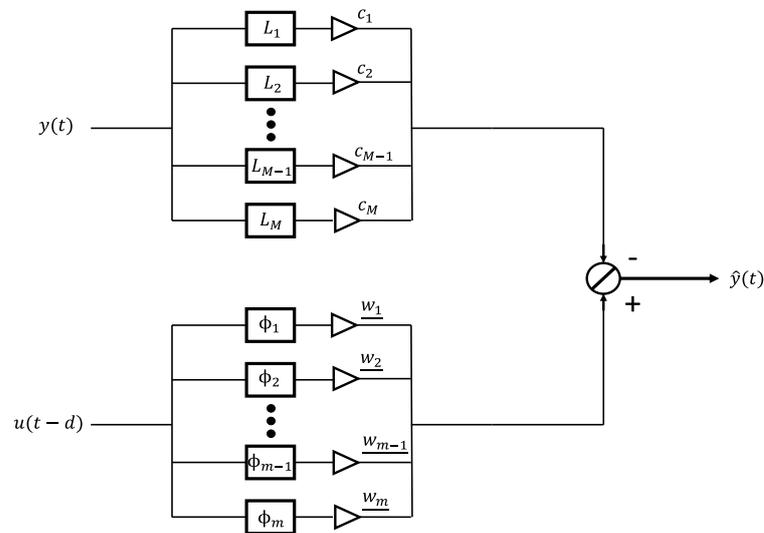
Fonte: Autoria própria.

Dessa forma, de acordo com a equação (4.22), tem-se uma estrutura de modelo baseada em funções de bases ortonormais e em funções de bases radiais, além de coeficientes de ponderação, os quais são os parâmetros do modelo. Na Figura 4.4 é apresentada uma ilustração de como a saída do modelo é formulada utilizando-se OBF e RBF.

Para permitir o ajuste dos parâmetros do modelo, a equação (4.22) pode ainda ser escrita na forma de uma regressão linear

$$\hat{y}(t) = \varphi^T(t)\theta \quad (4.23)$$

Figura 4.3: Representação gráfica do equacionamento da saída estimada em termos de funções de bases ortonormais e funções de base radiais para um sistema de *Hammerstein*.



Fonte: Autoria própria.

onde $\varphi^T = [-v_1(t) \cdots -v_M(t) \quad \phi_1(t-d) \cdots \phi_m(t-d)]$, com $v_i(t) = L_i(q^{-1}, p)y(t)$ e $\theta = [c_1 \cdots c_M \quad \underline{w}_1 \cdots \underline{w}_m]$. O conjunto de parâmetros desconhecidos θ do modelo pode ser estimado por meio do algoritmo de mínimos quadrados recursivos (*Recursive Least Squares* - RLS).

Sistemas de Wiener

Para o caso de um modelo de *Wiener*, (STANISŁAWSKI; LATAWIEC, 2008) emprega uma abordagem semelhante ao que foi mostrado anteriormente para um modelo de *Hammerstein* com uma formulação que expressa a saída estimada do sistema como uma soma de uma parte linear e outra não linear.

Considerando-se um sistema de *Wiener* representado pela equação (4.24),

$$y(t) = f[G(q)u(t) + e_W(t)] \quad (4.24)$$

pode-se, então, escrever

$$f^{-1}[y(t)] = G(q)u(t) \quad (4.25)$$

Para este caso, é possível aproximar a função $G(q)$ utilizando o conceito de OBF regular, de forma que

$$G(q)u(t) = \sum_{i=1}^M c_i L_i(q, p)u(t) \quad (4.26)$$

onde $L_i(q, p)$, $i = 1, 2, 3, \dots, M$ são, para este caso, funções de *Laguerre* e c_i , $i =$

$1, 2, 3, \dots, M$ são coeficientes de ponderação. Além disso, o mapeamento não linear $f^{-1}[y(t)]$ pode ser aproximado por meio de uma expansão polinomial de forma que

$$f^{-1}[y(t)] = \bar{d}_1 y(t) + \bar{d}_2 y^2(t) + \dots + \bar{d}_m y^m(t) \quad (4.27)$$

e, sem perda de generalidade, é possível fazer o primeiro coeficiente, \bar{d}_1 , igual a um, resultando em

$$f^{-1}[y(t)] = y(t) + d_2 y^2(t) + \dots + d_m y^m(t) \quad (4.28)$$

A partir dessa expansão, $f^{-1}[y(t)]$ pode ser escrita como

$$f^{-1}[y(t)] = y(t) + \psi(t) \quad (4.29)$$

onde $\psi(t)$ é um mapeamento não linear correspondente a $d_2 y^2(t) + \dots + d_m y^m(t)$. Dessa forma, de acordo com a equação 4.25,

$$y(t) + \psi(t) = \sum_{i=1}^M c_i L_i(q, p) u(t) + e_W(t) \quad (4.30)$$

$$y(t) - e_W(t) = \sum_{i=1}^M c_i L_i(q, p) u(t) - \psi(y(t)) \quad (4.31)$$

Fazendo $y(t) - e_W(t) = \hat{y}(t)$ e escrevendo o mapeamento não linear $\psi(y(t))$ como uma expansão usando funções de base radiais (RBF), por fim, tem-se a formulação apresentada na equação (4.32)

$$\hat{y}(t) = \sum_{i=1}^M c_i L_i(q, p) u(t) - \sum_{j=1}^m w_j \phi_j(y(t)) \quad (4.32)$$

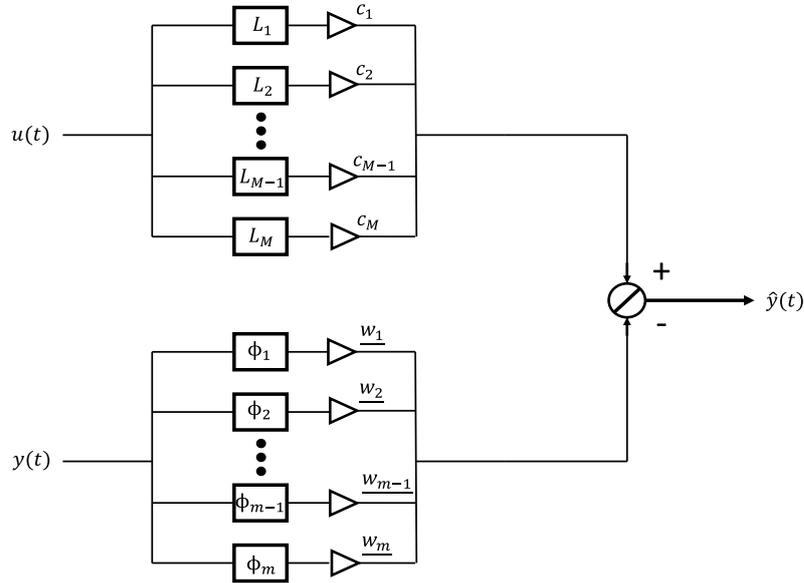
em que $\phi_j(y(t))$, $j = 1, 2, \dots, m$ são funções de base radiais e w_j , $j = 1, 2, \dots, m$ são coeficientes de ponderação.

Dessa forma, de acordo com a equação (4.32), tem-se uma estrutura de modelo baseada em funções de bases ortonormais e em funções de bases radiais, além de coeficientes de ponderação, os quais são os parâmetros do modelo. Na Figura 4.4 é apresentada uma ilustração de como a saída do modelo é formulada utilizando-se OBF e RBF.

Para permitir o ajuste dos parâmetros do modelo, a equação 4.32 pode também ser escrita na forma de uma regressão linear

$$\hat{y}(t) = \boldsymbol{\varphi}^T(t) \boldsymbol{\theta} \quad (4.33)$$

Figura 4.4: Representação gráfica de como a saída de sistema *Wiener* pode ser dada em termos de funções de bases ortonormais e funções de base radiais



Fonte: Autoria própria.

onde tem-se $\varphi^T = [x_1(t) \cdots x_M(t) \ \phi_1(y(t)) \cdots \phi_m(y(t))]$, com $x_i(t) = L_i(q, p)u(t)$ e $\theta = [c_1 \cdots c_M \ w_1 \cdots w_m]$. O conjunto de parâmetros desconhecidos θ do modelo pode ser estimado dessa forma por meio do algoritmo de mínimos quadrados recursivos (*Recursive Least Squares* - RLS).

4.3 Método Proposto

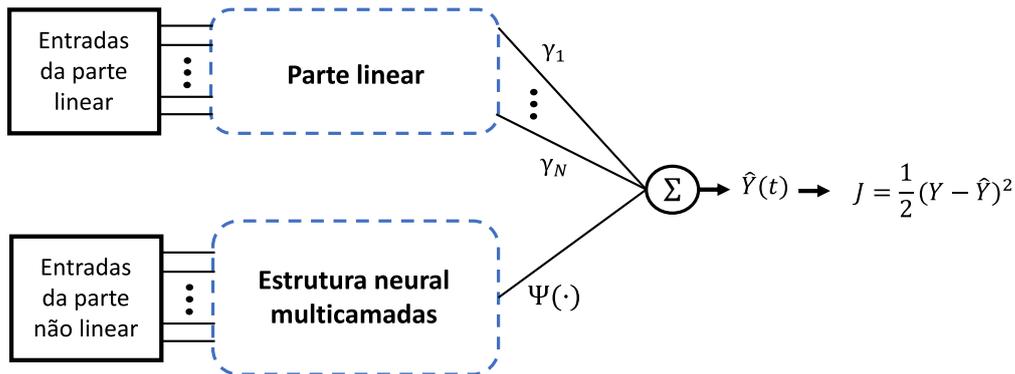
O esquema de identificação apresentado em (STANISŁAWSKI; LATAWIEC, 2008) baseia-se, em termos gerais, na separação das partes linear e não linear para a identificação de sistema a partir de modelos de *Hammerstein* e *Wiener* por meio da estimação de parâmetros via mínimos quadrados recursivos. Com base nessa abordagem, é proposta no presente trabalho uma estratégia de identificação para sistemas não lineares baseada em aprendizagem profunda a partir de modelos orientados a blocos tomando como ponto de partida a separação entre as partes linear e não linear.

Diante da possibilidade de representar um sistema não linear orientado a blocos como a soma de componentes lineares e não lineares, isto é, a possibilidade representar o sistema por uma parte linear e uma parte não linear de maneira dissociada, propõe-se, então, um esquema de identificação baseado na utilização de redes neurais profundas (multicamadas) para a aproximação da parte não linear do sistema e no aproveitamento do processo de

treinamento da rede multicamadas para a realização simultânea do ajuste dos parâmetros referentes à parte linear. Dessa forma, objetiva-se, após o processo de treinamento da rede como um todo, a obtenção de um modelo capaz de aproximar a dinâmica de um sistema de interesse. Na Figura 4.5 é mostrada a representação esquemática da abordagem proposta.

Figura 4.5: Representação da estrutura geral aplicada no identificação proposto.

A



Fonte: Autoria própria.

Ao longo do treinamento da rede, é realizado o ajuste dos parâmetros da parte linear e, simultaneamente são ajustados os pesos da parte multicamadas. Desse modo, ao final do processo de treinamento é obtida uma estrutura capaz de fornecer uma estimativa $\hat{y}(t)$ da saída do sistema no instante t a partir dos valores anteriores de entrada e saída do sistema.

Para o treinamento da rede, adotando-se uma função de custo J da pela equação (4.34), onde $y(t)$ é o valor de referência da saída real do sistema e $\hat{y}(t)$ é a saída estimada pela rede a partir dos valores fornecidos como entradas.

$$J(t) = \frac{1}{2}[y(t) - \hat{y}(t)]^2 \quad (4.34)$$

A atualização dos pesos da rede, tanto da parte linear quanto da parte multicamada, é realizada com base na derivada parcial da função de custo J em relação ao peso específico. Para o parâmetro γ_1 , por exemplo, se faz necessário o cálculo da derivada parcial $\frac{\partial J}{\partial \gamma_1}$, que é dada por

$$\frac{\partial J}{\partial \gamma_1} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \gamma_1} \quad (4.35)$$

Como $J(t) = \frac{1}{2}[Y(t) - \hat{Y}(t)]^2$, então

$$\frac{\partial J}{\partial \hat{y}} = \frac{2}{2}[y - \hat{y}] \frac{d[y - \hat{y}]}{d\hat{y}} \quad (4.36)$$

$$= [y - \hat{y}](-1) \quad (4.37)$$

$$= [\hat{y} - y] \quad (4.38)$$

Já com relação a $\frac{\partial \hat{y}}{\partial a_1}$, supondo $\hat{y}(t) = \gamma_1 y(t-1) + \Psi(t)$, então

$$\frac{\partial \hat{y}}{\partial \gamma_1} = y(t-1) = y(t-1) \quad (4.39)$$

Assim,

$$\frac{\partial J}{\partial \gamma_1} = [\hat{y} - y] * y(t-1) \quad (4.40)$$

Dessa forma, a cada iteração do treinamento da rede, o valor do peso a_1 é atualizado utilizando-se a técnica do gradiente descendente por meio do termo $\frac{\partial J}{\partial \gamma_1}$ de modo que se $\gamma_1[n]$ é o valor do peso γ_1 na iteração de treinamento n , então, assim como são atualizados os pesos da rede neural multicamadas, o valor do peso γ_1 sofre a seguinte modificação:

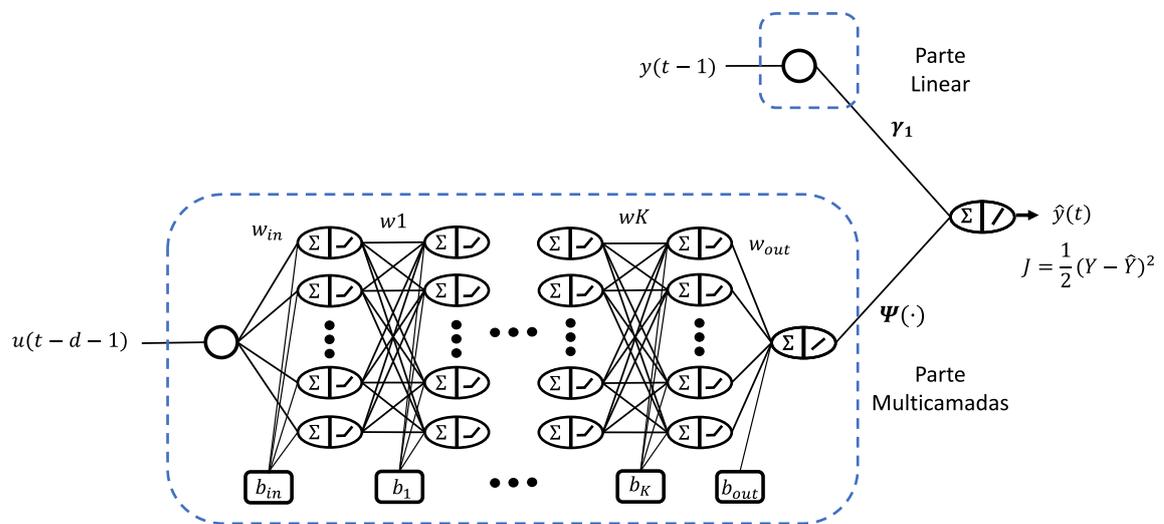
$$\gamma_1[n+1] = \gamma_1[n] - \frac{\partial J}{\partial \gamma_1} \quad (4.41)$$

No que diz respeito aos pesos que compõem a parte multicamadas da rede, eles são atualizados também com base no algoritmo do gradiente descendente, de tal maneira que para um determinado peso W_{ij} , o seu valor no próximo passo de treinamento é dado por

$$W_{ij}[n+1] = W_{ij}[n] - \frac{\partial J}{\partial W_{ij}} \quad (4.42)$$

Para a composição da parte multicamadas indicada na Figura 4.6 é proposto o uso de estrutura do tipo *feedforward* com múltiplas camadas ocultas e um mesmo número de neurônios artificiais em todas as camadas, além de valores de polarização unitários. Como função de ativação propõe-se utilizar a função *ReLU* em todas as camadas ocultas. Assim como ilustrado na Figura para o caso de se aplicar $u(t-d-1)$ como entrada da parte multicamadas e $y(t-1)$ como entrada na parte linear, por exemplo. Onde os símbolos circulares são os nós de entrada, os símbolos elípticos representam os neurônios artificiais com a operação de soma e função de ativação, $b_{in}, b_1, \dots, b_K, b_{out}$ são os termos de polarização e $W_{in}, W_1, \dots, W_2, W_{out}$ são as matrizes que reúnem os pesos que interligam as camadas da rede.

Figura 4.6: Ilustração da estrutura de rede neural profunda empregada no método de identificação proposto.



Fonte: Autoria própria.

4.3.1 Sistemas de *Hammerstein*

Sistema de primeira ordem com atraso

Considerando um sistema de primeira ordem com uma entrada $u(t)$, uma saída $y(t)$, um atraso d e uma função de transferência $G_1(q)$ dada por

$$G_1(q) = q^{-d} \frac{K}{q - p_1} \quad (4.43)$$

$$= q^{-d} \frac{K}{q - p_1} \times \frac{q^{-1}}{q^{-1}} \quad (4.44)$$

$$= \frac{K}{1 - p_1 q^{-1}} q^{-(d+1)} \quad (4.45)$$

e $G_1^{-1}(q)$ sendo

$$G_1^{-1}(q) = \frac{1 - p_1 q^{-1}}{K} q^{(d+1)} \quad (4.46)$$

com

$$G_1^{-1}(q)y(t) = u(t) \quad (4.47)$$

então

$$\frac{1 - p_1 q^{-1}}{K} q^{(d+1)} y(t) = u(t) \quad (4.48)$$

$$(1 - p_1 q^{-1}) y(t) = K u(t) q^{-(d+1)} \quad (4.49)$$

$$y(t) + (-pq^{-1})y(t) = Ku(t - d - 1) \quad (4.50)$$

Para um sistema de *Hammerstein* formado pela função linear $G_1(q)$ e por um mapeamento não linear estático da entrada $f_1(u(t))$, além de um perturbação $e_H(t)$ é possível escrever

$$y(t) = G_1(q)[f_1(u(t)) + e_H(t)] \quad (4.51)$$

e assumindo que $G_1^{-1}(q)$ existe

$$G_1^{-1}(q)y(t) = [f_1(u(t)) + e_H(t)] \quad (4.52)$$

substituindo $G_1^{-1}(q)$

$$\frac{1 - p_1q^{-1}}{K}q^{(d+1)}y(t) = [f_1(u(t)) + e_H(t)] \quad (4.53)$$

$$(1 - p_1q^{-1})y(t) = Kq^{-(d+1)}[f_1(u(t)) + e_H(t)] \quad (4.54)$$

$$y(t) - p_1q^{-1}y(t) = Kq^{-(d+1)}[f_1(u(t))] + e_1(t) \quad (4.55)$$

onde $e_1 = Kq^{-(d+1)}e_H(t)$.

$$y(t) - e_1(t) = p_1q^{-1}y(t) + K[f_1(u(t - d - 1))] \quad (4.56)$$

e, por fim, fazendo $\hat{y}(t) = y(t) - e_1(t)$

$$\hat{y}(t) = p_1q^{-1}y(t) + K[f_1(u(t - d - 1))] \quad (4.57)$$

De maneira similar à formulação proposta por (STANISŁAWSKI; LATAWIEC, 2008) apresentada na equação (4.22), tem-se na equação (4.57), a saída do sistema de *Hammersstein* dada em termos da soma de um componente linear com outro não linear. Quanto à parte linear, é possível escrevê-la em termos da saída do sistema, $y(t)$, do operador de atraso, q^{-1} , além de um coeficiente de ponderação $a_1 = p_1$. Já com relação à parte não linear, é possível expressá-la como um mapeamento $\Psi(\cdot)$, de modo que

$$\hat{y}(t) = [a_1q^{-1}y(t)] + \Psi(u(t - d - 1)) \quad (4.58)$$

Com base na equação (4.58), tem-se o objetivo obter o mapeamento $\Psi(\cdot)$ por meio de uma rede neural proposta. Dessa forma, fazendo $a_1 = \gamma_1$, e escrevendo $q^{-1}y(t)$ como $y(t - 1)$

$$\hat{y}(t) = \gamma_1y(t - 1) + \Psi(u(t - d - 1)) \quad (4.59)$$

Assim, é possível usar o método de identificação proposto com $u(t-d-1)$ como entrada da parte não linear e $y(t-1)$ como entrada da parte linear.

Sistema de segunda ordem com atraso

Considerando um sistema de *Hammerstein* com parte linear dada pela função de transferência de segunda ordem $G_2(q)$, com um mapeamento não linear $f_2(u(t))$ e um perturbação $e_H(t)$. E, sendo $G_2(q)$ dada por

$$G_2(q) = q^{-d} \frac{K}{(q-p_1)(q-p_2)} \quad (4.60)$$

$$= q^{-(d+2)} \frac{K}{(1-p_1q^{-1})(1-p_2q^{-1})} \quad (4.61)$$

ou ainda

$$G_2(q) = q^{-(d+2)} \frac{K}{(1+(-p_1-p_2)q^{-1}+(p_1p_2)q^{-2})} \quad (4.62)$$

$$= q^{-(d+2)} \frac{K}{1+a_1q^{-1}+a_0q^{-2}} \quad (4.63)$$

onde $a_1 = -p_1 - p_2$ e $a_0 = (p_1 * p_2)$. Sendo $G_2^{-1}(q)$ dada por

$$G_2^{-1}(q) = q^{(d+2)} \frac{1+a_1q^{-1}+a_0q^{-2}}{K} \quad (4.64)$$

então, se $y(t)$ é

$$y(t) = G_2(q)[f_2(u(t)) + e_H(t)] \quad (4.65)$$

e

$$G_2^{-1}(q)y(t) = [f_2(u(t)) + e_H(t)] \quad (4.66)$$

logo

$$\frac{1+a_1q^{-1}+a_0q^{-2}}{K} q^{(d+2)} y(t) = [f_2(u(t)) + e_H(t)] \quad (4.67)$$

$$(1+a_1q^{-1}+a_0q^{-2})y(t) = Kq^{-(d+2)}[f_2(u(t)) + e_H(t)] \quad (4.68)$$

$$y(t) + [a_1q^{-1} + a_0q^{-2}]y(t) = Kq^{-(d+2)}[f_2(u(t))] + e_2(t) \quad (4.69)$$

onde $e_2 = Kq^{-(d+2)}e_H(t)$.

$$y(t) - e_2(t) = -[a_1q^{-1} + a_0q^{-2}]y(t) + K[f_2(u(t-d-2))] \quad (4.70)$$

e, por fim, fazendo $\hat{y}(t) = y(t) - e_2(t)$

$$\hat{y}(t) = -[a_1q^{-1} + a_0q^{-2}]y(t) + K[f_2(u(t-d-2))] \quad (4.71)$$

De maneira semelhante à formulação encontrada para o caso de primeira ordem, um sistema de *Hammerstein* com uma função linear de segunda ordem pode ter sua saída dada em termos da soma de uma componente linear e outra não linear. Neste caso, a parte linear do sistema também é dada em termos da saída, $y(t)$, em instantes de tempo anteriores ponderadas pelos coeficientes $-a_1$ e $-a_0$. Já no que se refere à parte não linear, é possível reescrevê-la como um mapeamento $\Psi(\cdot)$, de modo que

$$\hat{y}(t) = -[a_1q^{-1} + a_0q^{-2}]y(t) + \Psi(u(t-d-2)) \quad (4.72)$$

Assim, com base na equação (4.72), fazendo $-a_1 = \gamma_1$, $-a_0 = \gamma_2$ e escrevendo $q^{-1}y(t)$ como $y(t-1)$ e $q^{-2}y(t)$ como $y(t-2)$

$$\hat{y}(t) = \gamma_1y(t-1) + \gamma_2y(t-2) + \Psi(u(t-d-2)) \quad (4.73)$$

Neste caso, é possível usar o método de identificação proposto com $u(t-d-2)$ como entrada da parte da parte multicamadas e com $y(t-1)$ e $y(t-2)$ sendo as entradas da parte linear.

Sistema de segunda ordem com atraso com a adição de um zero

Seja um sistema de *Hammerstein* com parte linear dada pela função de transferência de segunda ordem comum zero $G_3(q)$, com um mapeamento não linear estático da entrada $f_3(u(t))$ e um perturbação $e_H(t)$. Sendo $G_3(q)$ dado por

$$\begin{aligned} G_3(q) &= q^{-d} \frac{K(q-r_1)}{(q-p_1)(q-p_2)} \\ &= q^{-d} \frac{Kq(1-r_1q^{-1})}{q(1-p_1q^{-1})q(1-p_2q^{-1})} \\ &= q^{-d-1} \frac{K(1-r_1q^{-1})}{(1-p_1q^{-1})(1-p_2q^{-1})} \end{aligned} \quad (4.74)$$

ou ainda

$$\begin{aligned} G_2(q) &= q^{-(d+1)} \frac{K(1-r_1q^{-1})}{(1+(-p_1-p_2))q^{-1} + (p_1p_2)q^{-2}} \\ &= q^{-(d+1)} \frac{K(1-b_1q^{-1})}{(1+a_1q^{-1}+a_0q^{-2})} \end{aligned} \quad (4.75)$$

onde $a_1 = -p_1 - p_2$, $a_0 = (p_1 * p_2)$ e $b_1 = r_1$ Sendo $G_3(q)^{-1}$ dada por

$$G_3^{-1}(q) = q^{(d+1)} \frac{(1 + a_1 q^{-1} + a_0 q^{-2})}{K(1 - b_1 q^{-1})} \quad (4.76)$$

então, se $y(t)$ é

$$y(t) = G_3(q)[f_3(u(t)) + e_H(t)] \quad (4.77)$$

e

$$G_3^{-1}(q)y(t) = [f_3(u(t)) + e_H(t)] \quad (4.78)$$

logo

$$[1 + a_1 q^{-1} + a_0 q^{-2}]q^{(d+1)}y(t) = [K(1 - b_1 q^{-1})][f_2(u(t)) + e_H(t)] \quad (4.79)$$

$$[1 + a_1 q^{-1} + a_0 q^{-2}]y(t) = [K(1 - b_1 q^{-1})]q^{-(d+1)}[f_2(u(t)) + e_H(t)] \quad (4.80)$$

$$y(t) + [a_1 q^{-1} + a_0 q^{-2}]y(t) = [K(1 - b_1 q^{-1})]q^{-(d+1)}[f_2(u(t))] + e_2(t) \quad (4.81)$$

onde $e_2 = [K(1 - b_1 q^{-1})]q^{-(d+1)}e_H(t)$.

$$y(t) - e_2(t) = -[a_1 q^{-1} + a_0 q^{-2}]y(t) + [K(1 - b_1 q^{-1})][f_3(u(t - d - 1))] \quad (4.82)$$

e, por fim, fazendo $\hat{y}(t) = y(t) - e_2(t)$

$$\hat{y}(t) = -[a_1 q^{-1} + a_0 q^{-2}]y(t) + [K(1 - b_1 q^{-1})][f_3(u(t - d - 1))] \quad (4.83)$$

$$\hat{y}(t) = -[a_1 q^{-1} + a_0 q^{-2}]y(t) + K[f_3(u(t - d - 1)) - b_1 f_3(u(t - d - 2))] \quad (4.84)$$

Assim, como nos caso anteriores, a formulação encontrada expressa a separação entre as partes linear e não linear. Entretanto, devido à presença de um zero em $G_3(q)$, a parte não linear passa a ser dada em termos da entrada do sistema em dois instantes de tempo anteriores, $u(t - d - 1)$ e $u(t - d - 2)$. Neste caso, é possível expressar esta parte como um mapeamento $\Psi(\cdot)$, de modo que

$$\hat{y}(t) = -[a_1 q^{-1} + a_0 q^{-2}]y(t) + \Psi(u(t - d - 1), u(t - d - 2)) \quad (4.85)$$

Essa alteração, por sua vez, implica no fornecimento de duas entradas para a parte multicamadas da rede e na manutenção do número de entradas para a parte linear. Assim, com base na equação (4.85), fazendo $-a_1 = \gamma_1$, $-a_0 = \gamma_2$ e escrevendo $q^{-1}y(t)$ como $y(t - 1)$ e $q^{-2}y(t)$ como $y(t - 2)$, tem-se

$$\hat{y}(t) = \gamma_1 y(t - 1) + \gamma_2 y(t - 2) + \Psi(u(t - d - 1), u(t - d - 2)) \quad (4.86)$$

Dessa maneira, é possível utilizar o método de identificação proposto com $y(t-1)$ e $y(t-2)$ sendo as entradas da parte linear além de $u(t-d-1)$ e $u(t-d-2)$ como entrada da parte da parte multicamadas.

4.3.2 Sistemas de Wiener

Sistema de primeira ordem

Considerando um sistema de Wiener com uma função de transferência de primeira ordem com atraso, $G_1(q)$, um mapeamento não linear estático da entrada $f_1(\cdot)$ e um perturbação $e_W(t)$, é possível escrever

$$y(t) = f_1[G_1(q)(u(t)) + e_W(t)] \quad (4.87)$$

e, assumindo que $f_1^{-1}(q)$ existe,

$$f_1^{-1}[y(t)] = [G_1(q)u(t) + e_W(t)] \quad (4.88)$$

substituindo $G_1(q)$

$$f_1^{-1}[y(t)] = \frac{K}{1 - p_1q^{-1}}q^{-(d+1)}u(t) + e_W(t) \quad (4.89)$$

$f_1^{-1}[y(t)]$ pode ser escrita como uma aproximação polinomial $y(t) + d_2y^2(t) + d_3y^3(t) + d_4y^5(t) + \dots + d_my^m(t)$, de modo que a parte $d_2y^2(t) + d_3y^3(t) + d_4y^5(t) + \dots + d_my^m(t)$ pode ser escrita como um mapeamento não linear $\phi_1(y(t))$. Assim $f_1^{-1}[y(t)]$ pode ser escrito como $f_1^{-1}[y(t)] = y(t) + \phi_1(y(t))$ Dessa forma,

$$y(t) + \phi_1(y(t)) = \frac{K}{1 - p_1q^{-1}}q^{-(d+1)}u(t) + \frac{1 - p_1q^{-1}}{1 - p_1q^{-1}}e_W(t) \quad (4.90)$$

e

$$(1 - p_1q^{-1})[y(t) + \phi_1(y(t))] = K[q^{-(d+1)}u(t) + e_1(t)] \quad (4.91)$$

onde $e_1(t) = K(1 - p_1q^{-1})e_W(t)$ e

$$[y(t) + \phi_1(y(t))] - p_1[y(t-1) + \phi_1(y(t-1))] = Ku(t-d-1) + e_1(t) \quad (4.92)$$

fazendo $\hat{y}(t) = y(t) - e_1(t)$

$$\hat{y}(t) + \phi_1(y(t)) = Ku(t-d-1) + p_1[y(t-1) + \phi_1(y(t-1))] \quad (4.93)$$

e, assim,

$$\hat{y}(t) = -\phi_1(y(t))Ku(t-d-1) + p_1y(t-1) + p_1\phi_1(y(t-1)) \quad (4.94)$$

Entretanto, para o cálculo de $\phi_1(y(t))$ e, conseqüentemente, de $\hat{y}(t)$, é necessário saber o valor de $y(t)$, que é a própria saída do sistema no instante t . Essa condição é um fator limitante do ponto de vista prático, uma vez que se faz necessário saber o valor da saída no tempo t para se calcular uma estimativa da saída também no instante de tempo t . Todavia, colocando a equação anterior na seguinte forma,

$$\hat{y}(t) + \phi_1(y(t)) = Ku(t-d-1) + p_1y(t-1) + p_1\phi_1(y(t-1)) \quad (4.95)$$

é possível notar que o resultado da soma $\hat{y}(t) + \phi_1(y(t))$ é dado por uma combinação linear de $u(t-d-1)$, $y(t-1)$ e $\phi_1(y(t-1))$. Assim, reescrevendo o termo $Ku(t-d-1)$ como uma soma da forma

$$Ku(t-d-1) = \alpha_1[Ku(t-d-1)] + (1-\alpha_1)[Ku(t-d-1)] \quad (4.96)$$

e de maneira similar, escrevendo $p_1y(t-1)$ e $p_1\phi_1(y(t-1))$ como

$$p_1y(t-1) = \alpha_2[p_1y(t-1)] + (1-\alpha_2)[p_1y(t-1)] \quad (4.97)$$

e

$$p_1\phi_1(y(t-1)) = \alpha_3[p_1\phi_1(y(t-1))] + (1-\alpha_3)[p_1\phi_1(y(t-1))] \quad (4.98)$$

Deste modo, é possível escrever $\phi_1(y(t))$ como

$$\phi_1(y(t)) = (1-\alpha_1)[Ku(t-d-1)] + (1-\alpha_2)[p_1y(t-1)] + (1-\alpha_3)[p_1\phi_1(y(t-1))] \quad (4.99)$$

e, por fim,

$$\hat{y}(t) = \alpha_1[Ku(t-d-1)] + \alpha_2[p_1y(t-1)] + \alpha_3[p_1\phi_1(y(t-1))] \quad (4.100)$$

A partir da equação (4.100), fazendo $\alpha_1K = \gamma_1$, $\alpha_2p_1 = \gamma_2$ e escrevendo $\alpha_3[p_1\phi_1(y(t-1))]$ como um mapeamento não linear $\Psi(y(t-1))$, tem-se:

$$\hat{y}(t) = \gamma_1[u(t-d-1)] + \gamma_2[y(t-1)] + \Psi(y(t-1)) \quad (4.101)$$

Dessa forma, é possível usar o método de identificação proposto com $u(t-d-1)$ e $y(t-1)$ como entradas da parte linear e $y(t-1)$ como entrada da parte multicamadas.

Sistema de segunda ordem

Para o caso de um sistema de *Wiener* que tenha como parte linear uma função de transferência $G_2(q)$ de segunda ordem com um atraso conhecido d , um mapeamento não linear estático $f_2(\cdot)$ e um termo de erro/perturbação $e_W(t)$, é possível escrever

$$y(t) = f_2[G_2(q)(u(t)) + e_W(t)] \quad (4.102)$$

e considerando que $f_2^{-1}(\cdot)$ existe

$$f_2^{-1}[y(t)] = [G_2(q)u(t) + e_W(t)] \quad (4.103)$$

sendo $G_2(q)$ dada por

$$G_2(q) = q^{-(d+2)} \frac{K}{1 + a_1q^{-1} + a_0q^{-2}} \quad (4.104)$$

então

$$f_2^{-1}[y(t)] = q^{-(d+2)} \frac{K}{1 + a_1q^{-1} + a_0q^{-2}} u(t) + e_W(t) \quad (4.105)$$

considerando que o mapeamento $f_2^{-1}[y(t)]$ pode ser aproximada por meio de uma expansão polinomial $y(t) + d_2y^2(t) + d_3y^3(t) + \dots + d_my^m(t)$, de modo que a parte $d_2y^2(t) + d_3y^3(t) + \dots + d_my^m(t)$ pode representado como um mapeamento não linear $\phi_2(y(t))$. Dessa forma, $f_2^{-1}[y(t)] = y(t) + \phi_2(y(t))$

$$y(t) + \phi_2(y(t)) = q^{-(d+2)} \frac{K}{1 + a_1q^{-1} + a_0q^{-2}} u(t) + \frac{1 + a_1q^{-1} + a_0q^{-2}}{1 + a_1q^{-1} + a_0q^{-2}} e_W(t) \quad (4.106)$$

e, fazendo $(1 + a_1q^{-1} + a_0q^{-2})e_W(t) = e_1(t)$

$$(1 + a_1q^{-1} + a_0q^{-2})[y(t) + \phi_2(y(t))] = Kq^{-(d+2)}u(t) + e_1(t) \quad (4.107)$$

$$[y(t) + \phi_2(y(t))] + (a_1q^{-1} + a_0q^{-2})[y(t) + \phi_2(y(t))] = Ku(t - d - 2) + e_1(t) \quad (4.108)$$

$$Ku(t-d-2) + e_1(t) = [y(t) + \phi_2(y(t))] + a_1[y(t-1) + \phi_2(y(t-1))] + a_0[y(t-2) + \phi_2(y(t-2))] \quad (4.109)$$

$$y(t) + \phi_2(y(t)) = Ku(t-d-2) - a_1y(t-1) - a_0y(t-2) - a_1\phi_2(y(t-1)) - a_0\phi_2(y(t-2)) + e_1(t) \quad (4.110)$$

e fazendo $\hat{y}(t) = y(t) - e_1(t)$

$$\hat{y}(t) + \phi_2(y(t)) = Ku(t-d-2) - a_1y(t-1) - a_0y(t-2) - a_1\phi_2(y(t-1)) - a_0\phi_2(y(t-2)) \quad (4.111)$$

de maneira similar ao caso do primeiro grau é possível escrever $Ku(t-d-2)$ como

$$Ku(t-d-2) = \alpha_1[Ku(t-d-2)] + (1 - \alpha_1)[Ku(t-d-2)] \quad (4.112)$$

e, da mesma forma,

$$a_1y(t-1) = \alpha_2[a_1y(t-1)] + (1 - \alpha_2)[a_1y(t-1)] \quad (4.113)$$

$$a_0y(t-2) = \alpha_3[a_0y(t-2)] + (1 - \alpha_3)[a_0y(t-2)] \quad (4.114)$$

$$a_1(\phi_2(y(t-1))) = \alpha_4[a_0\phi_2(y(t-1))] + (1 - \alpha_4)[a_1\phi_2(y(t-1))] \quad (4.115)$$

$$a_0(\phi_2(y(t-2))) = \alpha_5[a_0\phi_2(y(t-2))] + (1 - \alpha_5)[a_0\phi_2(y(t-2))] \quad (4.116)$$

Dese modo, é possível escrever $\phi_2(y(t))$ como

$$\begin{aligned} \phi_2(y(t)) &= (1 - \alpha_1)[Ku(t-d-2)] - (1 - \alpha_2)[a_1y(t-1)] - \\ &\quad - (1 - \alpha_3)[a_0y(t-2)] - (1 - \alpha_4)[a_1\phi_2(y(t-1))] - \\ &\quad - (1 - \alpha_5)[a_0\phi_2(y(t-2))] \end{aligned} \quad (4.117)$$

e, por fim,

$$\begin{aligned} \hat{y}(t) &= \alpha_1[Ku(t-d-2)] - \alpha_2[a_1y(t-1)] - \alpha_3[a_0y(t-2)] - \\ &\quad - \alpha_4[a_1\phi_2(y(t-1))] - \alpha_5[a_0\phi_2(y(t-2))] \end{aligned} \quad (4.118)$$

A partir da equação (4.118) e fazendo $\alpha_1 K = \gamma_1$, $-\alpha_2 a_1 = \gamma_2$, $-\alpha_3 a_0 = \gamma_3$ e escrevendo $-\alpha_4[a_1\phi_2(y(t-1))] - \alpha_5[a_0\phi_2(y(t-2))]$ como um único mapeamento não linear $\Psi(y(t-1), y(t-2))$, tem-se:

$$\hat{y}(t) = \gamma_1[u(t-d-2)] + \gamma_2[y(t-1)] + \gamma_3[y(t-2)] + \Psi(y(t-1), y(t-2)) \quad (4.119)$$

Assim, é possível aplicar o método proposto com $u(t-d-2)$, $y(t-1)$ e $y(t-2)$ como entradas da parte linear e com $y(t-1)$ e $y(t-2)$ sendo as entradas da parte multicamadas.

Sistema de segunda ordem com um zero

Considerando um sistema de Wiener, com parte linear dada por uma função de segunda ordem com um zero, $G_3(q)$, além de uma parte não linear estática seja $f_3(\cdot)$ e um termo de erro/perturbação $e_W(t)$ relacionados da forma mostrada na equação (4.120)

$$y(t) = f_3[G_3(q)(u(t)) + e_W(t)] \quad (4.120)$$

e considerando que $f_3^{-1}(\cdot)$ existe

$$f_3^{-1}[y(t)] = [G_3(q)u(t) + e_W(t)] \quad (4.121)$$

sendo $G_3(q)$ dada por

$$G_3(q) = q^{-(d+1)} \frac{K(1 - b_1 q^{-1})}{1 + a_1 q^{-1} + a_0 q^{-2}} \quad (4.122)$$

então

$$f_3^{-1}[y(t)] = q^{-(d+1)} \frac{K(1 - b_1 q^{-1})}{1 + a_1 q^{-1} + a_0 q^{-2}} u(t) + e_W(t) \quad (4.123)$$

considerando que o mapeamento $f_3^{-1}[y(t)]$ pode ser aproximado por meio de uma expansão polinomial $y(t) + d_2 y^2(t) + d_3 y^3(t) + \dots + d_m y^m(t)$, de modo que a parte $d_2 y^2(t) + d_3 y^3(t) + \dots + d_m y^m(t)$ pode, por sua vez, ser representada como um mapeamento não linear $\phi_3(y(t))$. Dessa forma, $f_3^{-1}[y(t)] = y(t) + \phi_3(y(t))$

$$y(t) + \phi_3(y(t)) = q^{-(d+1)} \frac{K(1 - b_1 q^{-1})}{1 + a_1 q^{-1} + a_0 q^{-2}} u(t) + \frac{1 + a_1 q^{-1} + a_0 q^{-2}}{1 + a_1 q^{-1} + a_0 q^{-2}} e_W(t) \quad (4.124)$$

e, fazendo $(1 + a_1 q^{-1} + a_0 q^{-2})e_W(t) = e_1(t)$

$$(1 + a_1q^{-1} + a_0q^{-2})[y(t) + \phi_3(y(t))] = q^{-(d+1)}(K(1 - b_1q^{-1})u(t) + e_3(t)) \quad (4.125)$$

$$[y(t) + \phi_3(y(t))] + (a_1q^{-1} + a_0q^{-2})[y(t) + \phi_3(y(t))] = (K(1 - b_1q^{-1})u(t-d-1) + e_3(t)) \quad (4.126)$$

$$(K(1 - b_1q^{-1})u(t-d-1) + e_3(t) = [y(t) + \phi_3(y(t))] + a_1[y(t-1) + \phi_3(y(t-1))] + a_0[y(t-2) + \phi_3(y(t-2))]) \quad (4.127)$$

$$y(t) + \phi_3(y(t)) = (K(1 - b_1q^{-1})u(t-d-1) - a_1y(t-1) - a_1\phi_3(y(t-1))) - a_0y(t-2) - a_0\phi_3(y(t-2)) + e_3(t) \quad (4.128)$$

e fazendo $\hat{y}(t) = y(t) - e_3(t)$

$$\hat{y}(t) + \phi_3(y(t)) = Ku(t-d-1) - Kb_1u(t-d-2) - a_1y(t-1) - a_0y(t-2) - a_1\phi_3(y(t-1)) - a_0\phi_3(y(t-2)) \quad (4.129)$$

de maneira similar ao caso do primeiro grau é possível escrever $Ku(t-d-1)$ como

$$Ku(t-d-1) = \alpha_1[Ku(t-d-1)] + (1 - \alpha_1)[Ku(t-d-1)] \quad (4.130)$$

e, da mesma forma,

$$Kb_1u(t-d-2) = \alpha_2[Kb_1u(t-d-2)] + (1 - \alpha_2)[Kb_1u(t-d-2)] \quad (4.131)$$

$$a_1y(t-1) = \alpha_3[a_1y(t-1)] + (1 - \alpha_3)[a_1y(t-1)] \quad (4.132)$$

$$a_0y(t-2) = \alpha_4[a_0y(t-2)] + (1 - \alpha_4)[a_0y(t-2)] \quad (4.133)$$

$$a_1(\phi_3(y(t-1))) = \alpha_5[a_1\phi_3(y(t-1))] + (1 - \alpha_5)[a_1\phi_3(y(t-1))] \quad (4.134)$$

$$a_0(\phi_3(y(t-2))) = \alpha_6[a_0\phi_3(y(t-2))] + (1 - \alpha_6)[a_0\phi_3(y(t-2))] \quad (4.135)$$

Dese modo, é possível escrever $\phi(y(t))$ como

$$\begin{aligned} \phi(y(t)) = & (1 - \alpha_1)[Ku(t-d-1)] - (1 - \alpha_2)[Ku(t-d-2)] - \\ & - (1 - \alpha_3)[a_1y(t-1)] - (1 - \alpha_4)[a_0y(t-2)] - \\ & - (1 - \alpha_5)[a_1\phi_3(y(t-1))] - (1 - \alpha_6)[a_0\phi_3(y(t-2))] \end{aligned} \quad (4.136)$$

e, por fim,

$$\begin{aligned} \hat{y}(t) = & \alpha_1[Ku(t-d-1)] - \alpha_2[Kb_1u(t-d-2)] - \alpha_3[a_1y(t-1)] - \\ & - \alpha_4[a_0y(t-2)] - \alpha_5[a_1\phi_3(y(t-1))] - \alpha_6[a_0\phi_3(y(t-2))] \end{aligned} \quad (4.137)$$

A partir da equação (4.137) e fazendo $\alpha_1K = \gamma_1$, $-\alpha_2Kb_1 = \gamma_2$, $-\alpha_3a_1 = \gamma_3$, $-\alpha_4a_0 = \gamma_4$ e escrevendo $-\alpha_5[a_1\phi_3(y(t-1))] - \alpha_6[a_0\phi_3(y(t-2))]$ como um único mapeamento não linear $\Psi(y(t-1), y(t-2))$, tem-se:

$$\begin{aligned} \hat{y}(t) = & \gamma_1[u(t-d-1)] + \gamma_2[u(t-d-2)] + \gamma_3[y(t-1)] + \\ & + \gamma_4[y(t-2)] + \Psi(y(t-1), y(t-2)) \end{aligned} \quad (4.138)$$

Dessa maneira, é possível aplicar o método proposto com $u(t-d-1)$, $u(t-d-2)$, $y(t-1)$ e $y(t-2)$ como entradas da parte linear e com $y(t-1)$ e $y(t-2)$ sendo as entradas da parte multicamadas.

4.4 Conclusão

Em razão das semelhanças entre os campos da aprendizagem profunda e da identificação de sistemas, muitas pesquisas têm sido desenvolvidas no sentido de utilizar técnicas de aprendizagem baseadas em dados para aplicações de identificação. Além disso, uma outra abordagem possível consiste em empregar técnicas de aprendizagem profunda para modificar e aprimorar estratégias de identificação já existentes no sentido de melhorar o seu desempenho.

Capítulo 5

Resultados de Simulações

5.1 Introdução

Neste capítulo são descritos os procedimentos e requisitos associados à aplicação tanto do método de proposto, quanto do método de referência adotado para a identificação de sistemas simulados de *Hammerstein* e *Wiener*. Para isso foram definidos seis casos de estudo para os quais são apresentados em detalhes os procedimentos de obtenção de dados de entrada e saída, além da aplicação dos métodos de identificação em questão para cada um dos casos.

5.2 *Setup* de simulações

Np que se refere à identificação de sistemas, é fundamental que os sinais de entrada empregados sejam capazes de excitar as dinâmicas de interesse do sistema a ser identificado. De maneira mais específica, em se tratando de sistemas não lineares, é preciso que o sinal de excitação utilizado contemple diversos valores de amplitude dentro de uma faixa de interesse para que sejam obtidos dados de saída referentes à vários pontos de operação diferentes. Apenas com informações dessa natureza se torna possível a obtenção de modelos capazes de reproduzir o comportamento não linear de interesse.

Excitação APRBS

Para a realização das simulações e obtenção de dados foram adotadas excitações do tipo APRBS (*Amplitude Modulated PRBS* - PRBS Modulado em Amplitude). O sinal PRBS (*Pseudo-Random Binary Sequence* - Sequência Binária Pseudo-Aleatória), por sua vez, é amplamente utilizado para a obtenção de dados para aplicações de identificação de sistemas lineares. Entretanto, por ser um sinal binário, isto é, só apresentar duas amplitudes

possíveis, o sinal PRBS não é adequado para a obtenção de dados de sistemas não lineares para fins de identificação de sistemas. Em razão disso, para se ter um sinal com características semelhantes ao PRBS, mas que permita a identificação de comportamentos não lineares, é construído, a partir de um PRBS, um sinal com variações aleatórias de amplitude dentro de uma faixa estabelecida, resultando assim no chamado APRBS, o qual é amplamente utilizado em aplicações de identificação de sistemas não lineares (ISERMANN; MÜNCHHOF, 2010).

Para a obtenção de um sinal APBRS, de acordo com (ISERMANN; MÜNCHHOF, 2010), é necessário tomar como base um sinal PRBS e realizar os seguintes procedimentos:

- definir limites inferior a_{min} e superior a_{max} para a amplitude do sinal APRBS;
- realizar a contagem do número k de variações da amplitude do sinal PRBS original;
- dividir o intervalo $[a_{min}, a_{max}]$ em k diferentes amplitudes;
- atribuir aleatoriamente os k valores de amplitudes definidos aos k intervalos correspondentes aos passos do sinal PRBS.

Para gerar um sinal do tipo PRBS, ainda de acordo com (ISERMANN; MÜNCHHOF, 2010), é preciso atender aos seguintes requisitos:

- a amplitude a do sinal deve ser sempre escolhida como sendo a maior possível levando em conta os limites para a entrada $u(t)$ e a saída $y(t)$ do processo;
- quanto ao tempo de ciclo escolhido, λ , é sugerido que $\lambda \leq T_i/5$, onde T_i é a menor constante de tempo de interesse do processo;
- para tempo discreto, a escolha do tempo de ciclo λ deve ser condicionada a escolha do tempo de amostragem T_s de modo que λ seja um múltiplo inteiro de T_s

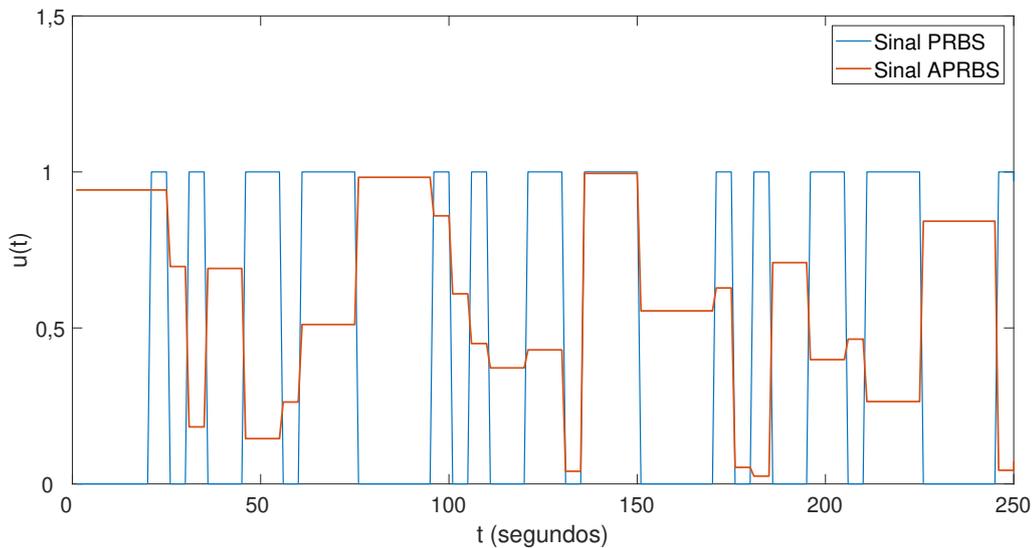
$$\lambda = \mu T_s \quad , \quad \text{com } \mu = 1, 2, \dots \quad (5.1)$$

- a frequência de corte do sinal será então $\omega_c = 1/\lambda$;
- o período de tempo, T_p , do sinal não deve ser menor que o tempo de transiente T_{95} (tempo em que o sinal atinge 95% do seu valor de regime) do sistema sobre investigação, sendo recomendado $T_p \approx 1,5T_{95}$;
- o tempo total de medição requerido do sinal, T_t , deve ser dado por um múltiplo M do período T_p

$$T_t = MT_p$$

Adotando-se por exemplo $[a_{min}, a_{max}] = [0, 1]$ é possível obter sinais APRBS a partir de sinais PRBS como mostrado na Figura 5.1, onde são apresentados os primeiros 250 pontos de dados dos sinais PRBS e APRBS .

Figura 5.1: Exemplo de um sinal PRBS de amplitude $a = 1$ e um sinal APRBS correspondente com amplitudes entre 0 e 1, projetado para os sistemas de primeira ordem.

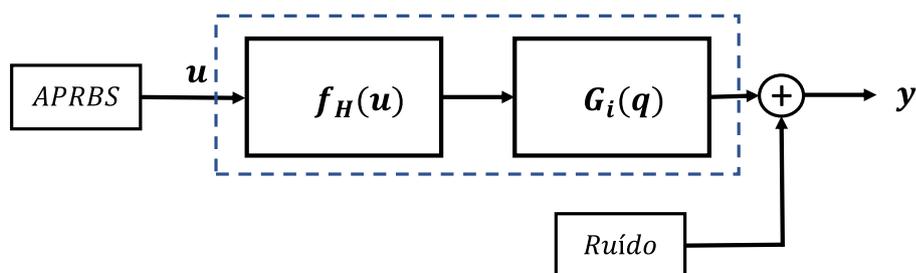


Fonte: Autoria própria.

Sistemas simulados

Com relação a simulação dos sistemas de interesse utilizando as excitações projetadas, foi utilizado o ambiente *Simulink*. Para o caso dos sistemas de *Hammerstein* foi utilizado o esquema geral apresentado na Figura 5.2, de modo que $f_H(u)$ e $G_i(q)$ são substituídas pela não linearidade e a função de transferência linear correspondentes para cada sistema simulado.

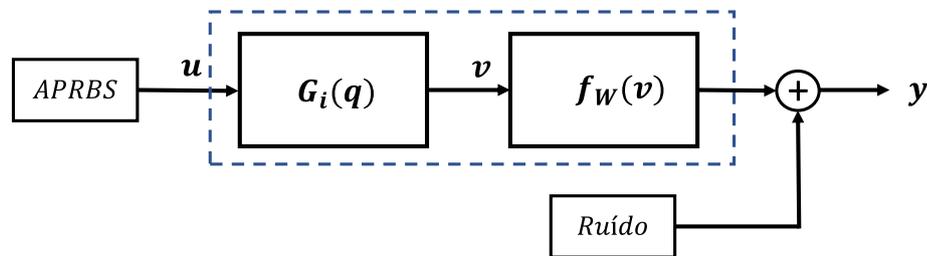
Figura 5.2: Esquema geral utilizado para as simulações dos sistemas de *Hammerstein*.



Fonte: Autoria própria.

Já no caso das simulações dos sistemas de *Wiener* foi utilizado o esquema apresentado na Figura 5.3, sendo $G_i(q)$ e $f_W(u)$ correspondentes às funções de transferências lineares e não linearidades de cada um dos casos utilizados.

Figura 5.3: Esquema geral utilizado para as simulações dos sistemas de *Wiener*.



Fonte: Autoria própria.

5.3 Identificação utilizando redes neurais profundas

Uma vez obtidos dados de entrada e saída para cada um dos sistemas escolhidos é possível, então, identificá-los utilizando a estrutura de rede neural profunda proposta. No entanto, para realizar o treinamento da rede é necessário que sejam definidos parâmetros estruturais e de treinamento da rede utilizada.

Treinamento das redes

De maneira geral, para o treinamento da rede foi adotada uma taxa de aprendizagem fixa, $\alpha = 0,00005$, escolhida de maneira empírica, e foi utilizado o algoritmo do gradiente descendente estocástico com mini-lotes de tamanho $l = 1$.

Em razão da possibilidade de geração de dados de maneira simulada, foram gerados dois conjuntos de dados para cada um dos sistemas de interesse mediante simulações realizadas separadamente de tal maneira que um é destinado apenas ao treinamento da rede e outro é utilizado exclusivamente para testar a precisão do modelo obtido.

De início, com relação apenas ao processo de ajuste dos pesos da rede, o conjunto de treinamento é separado em um subconjunto de treino propriamente dito e em um outro subconjunto de validação, de modo que o primeiro é usado para realizar o ajuste dos pesos da rede e o segundo é utilizado para acompanhar a evolução do processo de treinamento.

É importante lembrar que para o treinamento da rede se fazem necessários tanto os dados de entrada para a rede bem como o respectivo valor de referência da saída, isto é, o valor real que se espera obter em correspondência a um determinado conjunto de entradas. Assim, o procedimento de validação consiste no cálculo de um valor de saída a

partir de entradas pertencentes ao conjunto de validação e usando os valores correntes do pesos da rede, de modo que a diferença entre o valor obtido e o valor de referência pode ser tomado como uma métrica do quão bom está o ajuste dos pesos.

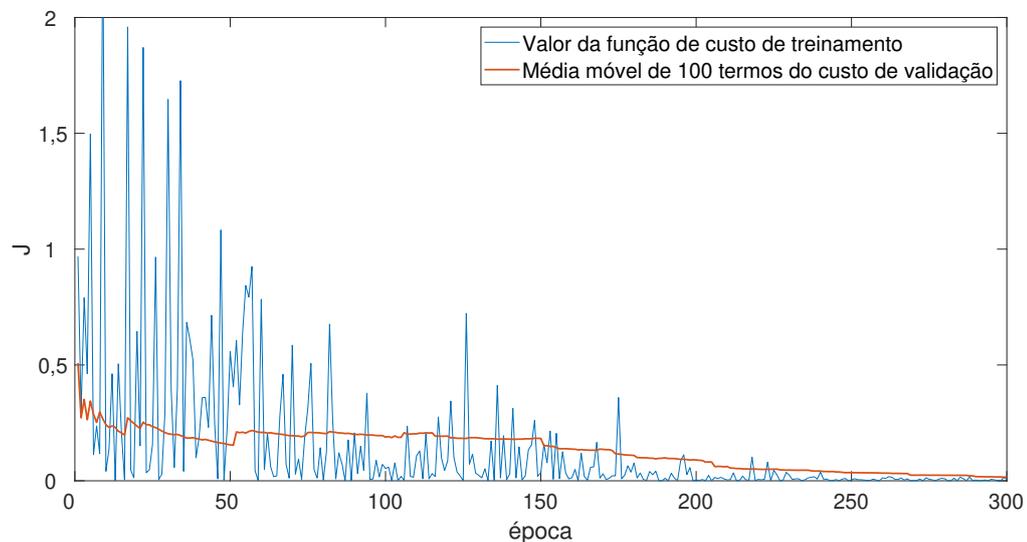
Uma vez definido o subconjunto de treino, são selecionados lotes de dados de tamanho $L = 500$ ao longo do processo de treinamento e é realizado o ajuste dos pesos da rede com base nos dados presentes em cada lote. Cada um desses ciclos de atualização dos pesos com base em "amostras" do subconjunto de treino recebe o nome de *época de treinamento*. Desse modo, para cada época são realizados os seguintes passos:

1. seleção aleatória de dados de entrada e referência correspondentes para a composição do lote referente à época;
2. realização da atualização dos pesos da rede de acordo com os dados presentes no conjunto selecionado, com base no algoritmo de *backpropagation* e utilizando o gradiente descendente;
3. seleção aleatória de dados do conjunto de validação para testar a precisão obtida com o conjunto de pesos obtidos para a época corrente;
4. cálculo do erro de validação e cálculo da média móvel do seu valor para as últimas cem épocas de treinamento com o objetivo de avaliar se, em média, o desempenho da rede tem melhorado ou não ao longo da épocas.

Na Figura 5.4 é mostrado um exemplo da evolução de um processo de treinamento. São mostrados graficamente os valores da função de custo de treinamento, J , da rede e os valores correspondentes à média móvel de 100 termos do custo de validação para as primeiras 300 épocas de treinamento. O uso da média móvel de cem termos do erro de validação é justificado pelo fato de que o emprego do gradiente descendente estocástico, isto é, o uso de mini-lotes selecionados aleatoriamente pode provocar variações consideráveis nos valores de erro obtidos em épocas subsequentes. Essa variação inerente ao uso do gradiente descendente estocástico também tem o efeito de dificultar que a rede realize o sobreajuste aos conjunto de treino, uma vez que, a cada época, o mini-lote de treino muda consideravelmente em relação à época anterior.

Com base nessa metodologia de treinamento e observando as entradas adequadas para cada caso, foi realizado o procedimento de identificação proposto para cada um dos conjuntos de dados referentes ao sistemas de *Hammerstein* e *Wiener* adotados para o teste do método.

Figura 5.4: Representação gráfica da evolução do valor da função de custo e do custo de validação da rede para as primeiras 300 épocas de um treinamento.



Fonte: Autoria própria.

5.4 Resultados de identificação

Com o intuito de avaliar o método proposto, bem como de compará-lo com o método tomado como referência, foram adotados seis casos de estudo, sendo três sistemas de *Hammerstein* e outros três de *Wiener*. Para cada um dos casos foram gerados dois conjuntos de dados independentes, sendo um conjunto de treinamento de 6000 pontos de dados e um conjunto de teste com 1000 amostras. Para gerar cada conjunto foi utilizado um sinal de excitação APRBS diferente. Para todos os casos, foi adicionado à saída um ruído branco gaussiano (média zero) com uma variância igual a 0,002.

Já com relação ao treinamento das redes neurais profundas, foi estabelecido um processo de treinamento de 1000 épocas para todos os casos.

5.4.1 Caso de estudo 1 (*Hammerstein*)

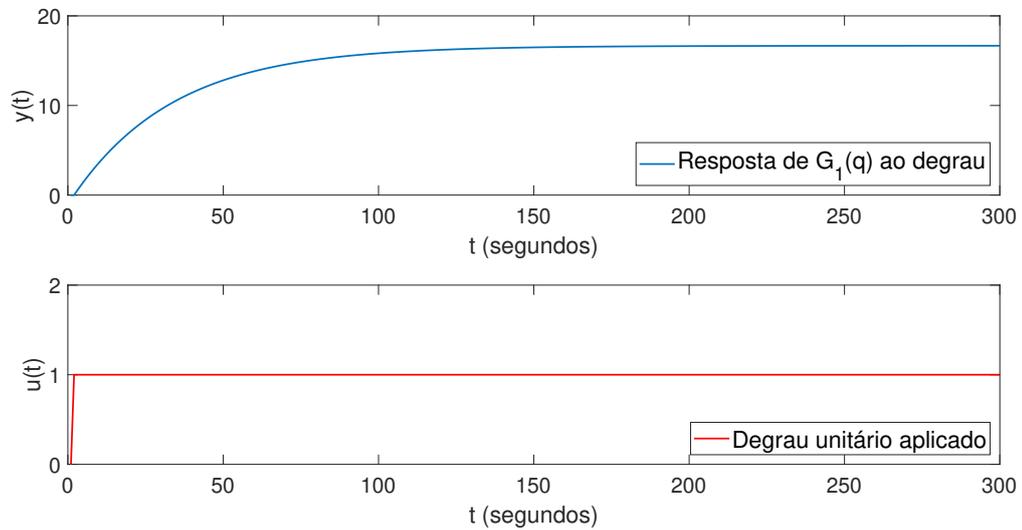
Sendo um sistema de *Hammerstein* formado por um bloco linear $G_1(q)$ e uma não linearidade $f_H(\cdot)$. Onde $G_1(q)$ e $f_H(\cdot)$ são funções utilizadas em (ZHU, 2001), e são dadas de acordo com as equações (5.23) e (5.22), respectivamente.

$$G_1(q) = (q^{-1}) \frac{0,5}{1 - 0,97q^{-1}} \quad (5.2)$$

$$f_H(v) = v + 3v^2 + 2v^3 \quad (5.3)$$

Na Figura 5.5 é mostrada a resposta de $G_1(q)$ a um impulso unitário.

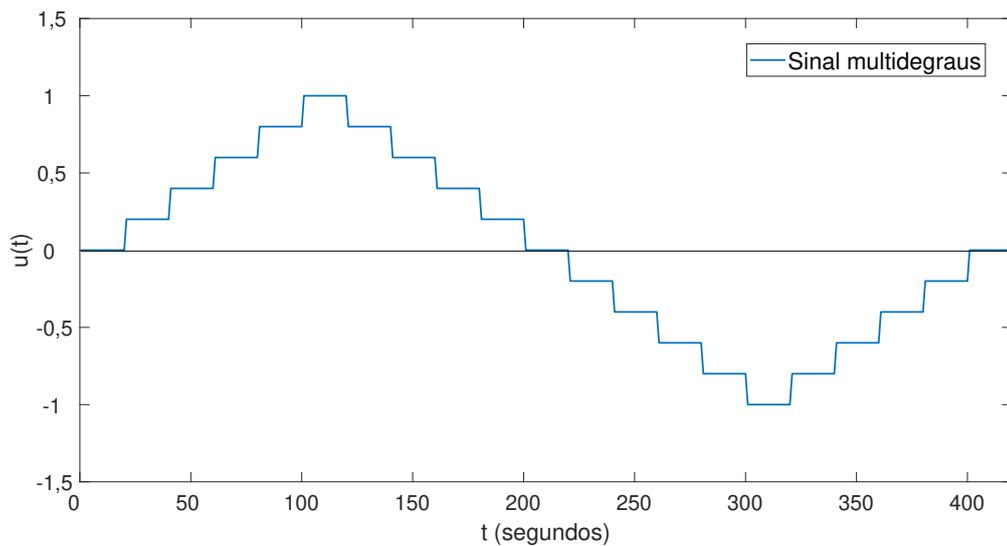
Figura 5.5: Resposta de $G_1(q)$ a um degrau unitário (Caso 1).



Fonte: Autoria própria.

Quanto à parte não linear, de acordo com (ZHU, 2001), é possível usar um sinal multidegraus (*staircase signal* - sinal "escadaria") para checar uma não linearidade. Utilizando um sinal tal qual mostrado na Figura 5.6, é possível verificar a característica não linear associada a $f_H(\cdot)$ a partir do gráfico apresentado na Figura 5.7

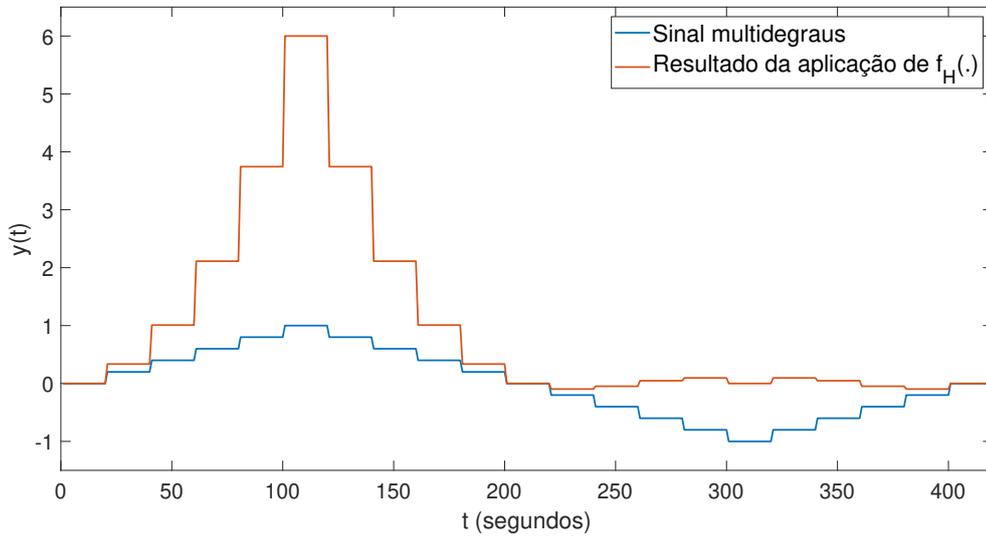
Figura 5.6: Sinal *multidegraus*.



Fonte: Autoria própria.

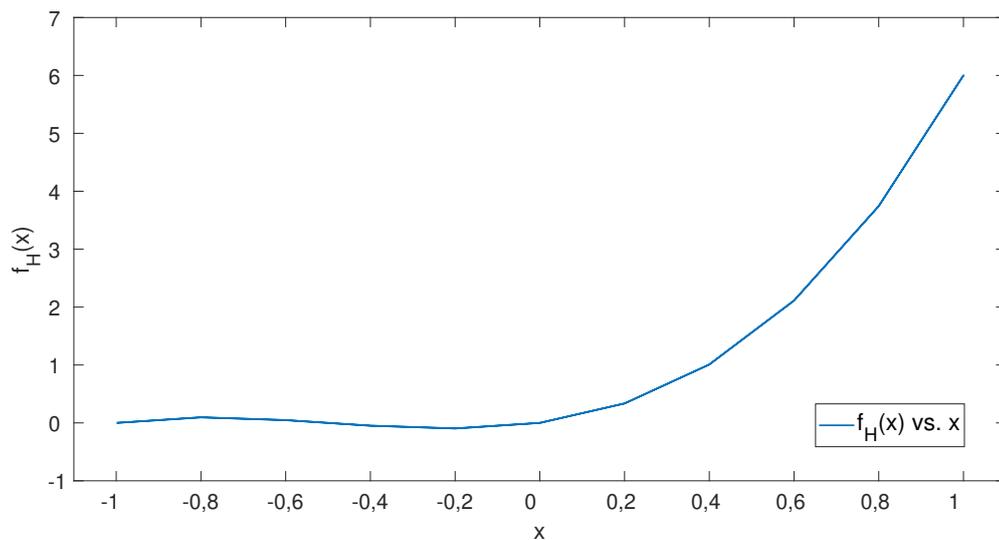
Desse modo, para valores entre -1 e 1 , a função $f_H(\cdot)$ se comporta de acordo com o gráfico apresentado na Figura 5.8

Figura 5.7: Gráfico comparativo entre o sinal *multidegraus* original e o resultado da aplicação de $f_H(\cdot)$



Fonte: Autoria própria.

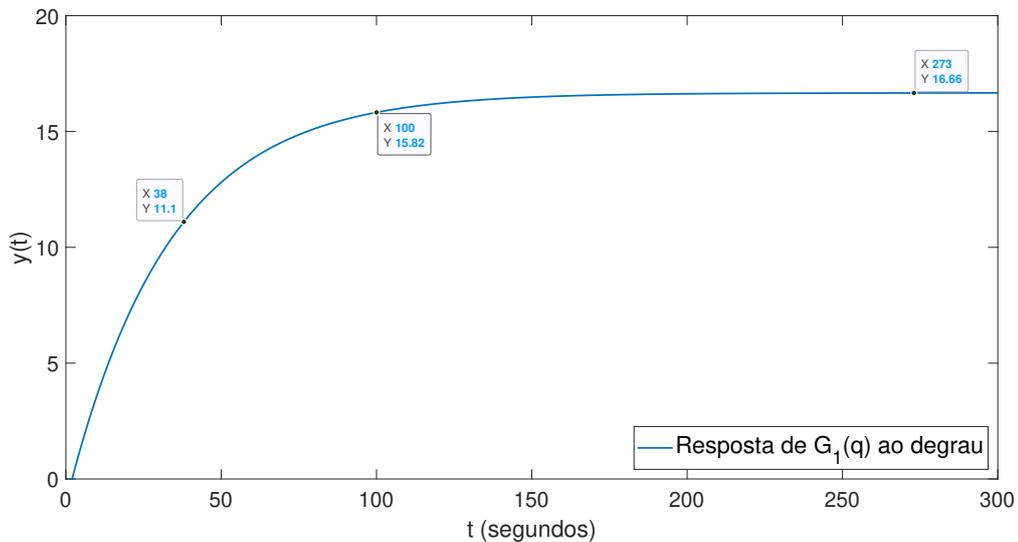
Figura 5.8: Comportamento não linear de $f_H(\cdot)$



Fonte: Autoria própria.

Com relação ao sinal de excitação, considerando a resposta ao degrau de $G_1(q)$, assim como mostrado na Figura 5.9, tem-se um valor de regime igual a 16,66 e, conseqüentemente, 67% desse valor sendo equivalente a 11,16. Para um tempo de amostragem T_s , tem-se em uma de tempo $T_i \approx 38 T_s$. Como o tempo de ciclo $\lambda \leq T_i/5$, então foi adotado $\lambda = 5 T_s$, com $T_s = 1$ segundo. Além disso, para 95% do valor de regime, tem-se um

Figura 5.9: Resposta ao degrau de $G_1(q)$ destacando se (da direita para a esquerda) o valor de regime e os tempos correspondentes a 95% e 67% desse valor.



Fonte: Autoria própria.

tempo $T_{95} \approx 100 T_s$. Como $T_p \approx 1,5 T_{95}$, então foi adotado para a definição do sinal de excitação um período $T_P = 150 T_s$. Para a obtenção de uma quantidade considerável de pontos de dados, foi arbitrado um número de período $M = 80$.

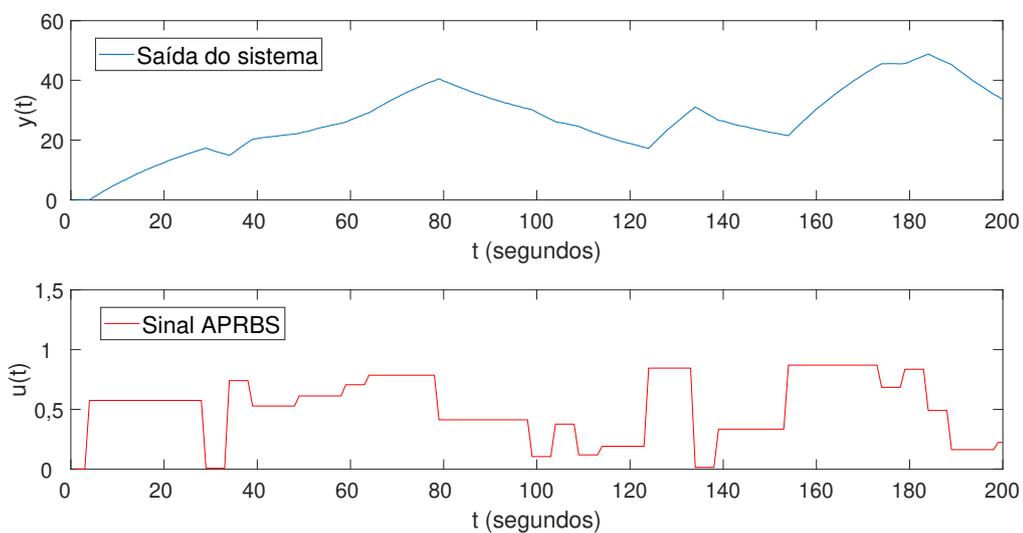
Foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 1$, respectivamente. A escolha desses valores se deu com o intuito de obter valores moderados tanto de saída quanto de entrada para evitar, em um primeiro momento, possíveis dificuldades de otimização.

A partir dos parâmetros estabelecidos foi utilizada a função $idinput()$ do software MATLAB para gerar um sinal PRBS e, a partir dele, gerar um sinal APRBS de acordo com o procedimento descrito na seção 5.2. Quanto ao sinal de excitação obtido, embora tenha sido especificado um período $T_P = 150 T_s$, o sinal PRBS só pode ser gerado com um período pertencente a um conjunto preestabelecido. Em razão disso a função anteriormente referida substituiu automaticamente o valor fornecido como entrada pelo valor mais próximo dentre os períodos possíveis, neste caso $T_P = 75 T_s$. Dessa forma, foi gerado um sinal PRBS a partir dos parâmetros

- $\lambda = 5 T_s$;
- $T_P = 75 T_s$;
- $M = 80$.

com um total de 6000 pontos de dados. O sinal APRBS resultante para uma faixa de amplitude $[0, 1]$, bem como a saída do sistema incluindo o ruído adicionado, são mostrados na Figura 5.10

Figura 5.10: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 1000 pontos de dados (Caso 1).



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Hammerstein* composto por uma função linear de primeira ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.1, onde $u(t - d - 1)$ é o sinal de excitação atrasado de $d + 1$ instantes de tempo e $y(t - 1)$ é o sinal de saída atrasado de um instante de tempo. Neste caso d é um atraso conhecido e para a simulação foi adotado $d = 2$. Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.10 para treinar da rede e posteriormente foi calculado o erro médio quadrático (*Mean Squared Error* - MSE) sobre um conjunto de teste para ser usado como métrica de desempenho. Esse procedimento de identificação foi realizado para diferentes escolhas projeto no que se refere à quantidade de camadas ocultas e o número de neurônios por

Tabela 5.1: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 1).

Entrada da parte linear	$y(t - 1)$
Entrada da parte multicamadas	$u(t - d - 1)$

Fonte: Autoria própria.

camada. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.2.

Tabela 5.2: Valores de erro sobre o conjunto de teste para diferentes quantidades de camadas e neurônios (Caso 1).

Número de camadas ocultas	Número de neurônios por camada							
		4	5	6	7	8	9	10
7		0,51700	0,74900	0,005230	0,00584	0,00502	0,00490	0,00498
8		0,00612	0,00715	0,00458	0,58900	0,00483	0,00450	0,00520
9		0,51400	0,00460	0,00768	0,00892	0,00579	0,00493	0,00538
10		0,00466	0,00504	0,00526	0,00908	0,00490	0,00495	0,00530

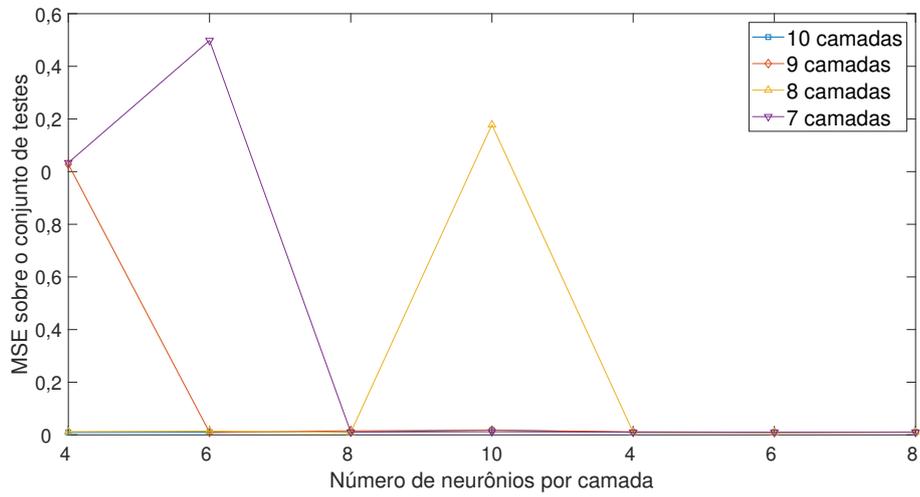
Fonte: Autoria própria.

Na Figura 5.11 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede.

A partir da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que, de maneira geral, os valores de erro decrescem à medida que é aumentado o número de neurônios para as diferentes quantidades de camadas. Entretanto para as redes com 10 camadas ocultas são obtidos valores de erro relativamente pequenos mesmo para quantidades pequenas de neurônios por camada oculta. O que indica que uma quantidade maior de camadas permitiu, para este caso uma melhor generalização para os dados de teste.

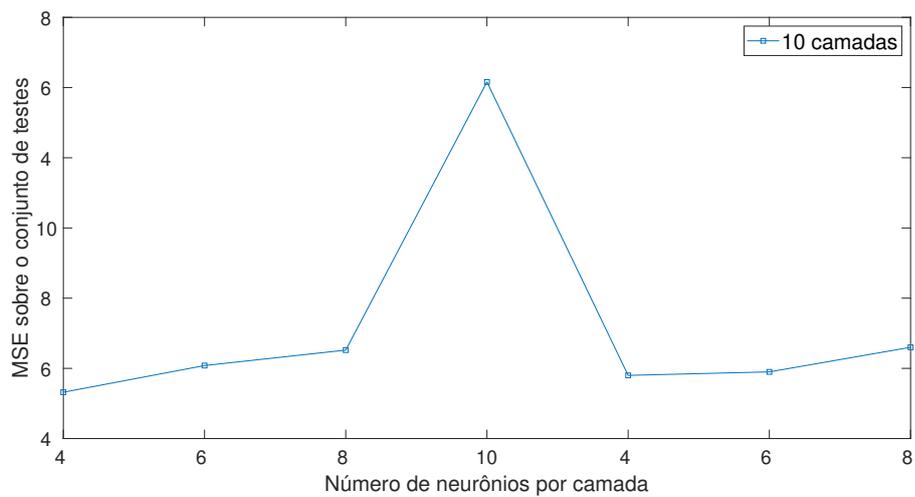
Na Figura 5.12, são mostrados os valores de erro obtidos para diferentes número de neurônios por camada para o caso específico da rede com 10 camadas ocultas. Com base no comportamento do erro apresentado na Figura 5.43, foi selecionada como modelo para o sistema de *Hammerstein* em questão, a estrutura com 10 camadas ocultas, além das camadas de entrada e saída, e 4 neurônios por camada, além dos termos de polarização unitários. Essa escolha se deu em razão de essa ser uma combinação capaz fornecer um dos menores erros sobre o conjunto de teste, assim como mostrado na Tabela 5.2, e também

Figura 5.11: Gráfico de valores MSE para diferentes quantidades de camadas ocultas e número de neurônios por camada (Caso 1).



Fonte: Autoria própria.

Figura 5.12: Valores de erro médio quadrático para 10 camadas ocultas e diferentes números de neurônios (Caso 1).

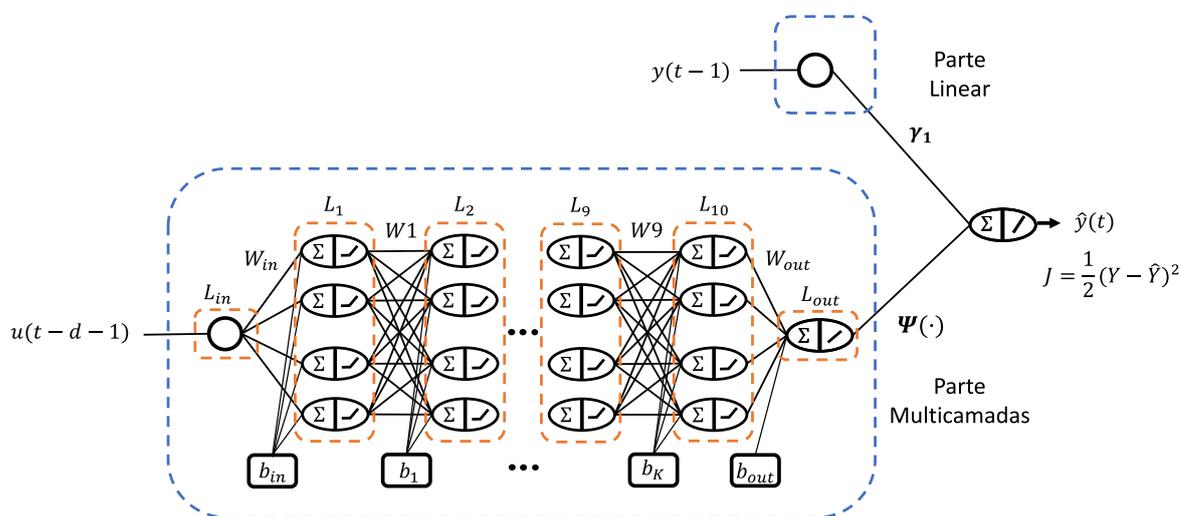


Fonte: Autoria própria.

resultar em uma quantidade menor de parâmetros a serem ajustados.

Para fins de comparação, considerando apenas os pesos associados às camadas ocultas, uma outra combinação com bom resultado, como por exemplo, 9 camadas e 5 neurônios. Entretanto, assim como ilustrado na Figura 5.44, para um número de camada C e um número de neurônios N são necessárias, apenas entre as camadas ocultas, $C - 1$ matrizes de pesos, cada qual com dimensão $N \times N$. Dessa forma, a rede com 9 camadas ocultas e 5 neurônios por camada resultaria em $(9 - 1) \times (5 \times 5) = 200$ pesos associados às camadas ocultas. Já para a estrutura selecionada, são obtidos $(10 - 1) \times (4 \times 4) = 144$ pesos relativos às camadas intermediárias. Dessa forma, a estrutura de rede escolhida para este caso de estudo é como ilustrada na Figura 5.44, onde $L_1, L_2, \dots, L_9, L_{10}$ são as camadas

Figura 5.13: Ilustração da estrutura de rede adotada (Caso 1).



Fonte: Autoria própria.

ocultas da rede e L_{in}, L_{out} são, respectivamente, as camadas de entrada saída da rede. Já $b_{in}, b_1, b_2, \dots, b_k, \dots, b_{out}$ são os vetores de polarização de valores unitários e com comprimento igual ao número de neurônios por camada.

De acordo com a estrutura de modelo apresentada na equação 5.4, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

$$\hat{y}(t) = \gamma_1 y(t-1) + \Psi(u(t-d-1)) \quad (5.4)$$

tem-se como coeficiente da parte linear, o parâmetro γ_1 , que corresponde, para este caso de estudo, ao negativo do coeficiente de primeira ordem do denominador da função $G_1(q)$, isto é, 0.97. Na Tabela 5.3 são mostrados o valor real desse coeficiente e o valor estimado a partir da estrutura de rede escolhida.

Para $\gamma_1 = 0,9702$ e tendo em vista que o mapeamento não linear $\Psi(\cdot)$ pode ser obtido

Tabela 5.3: Valores real e estimado para o coeficiente da parte linear de acordo com o método proposto (Caso 1).

	Valor real	Valor estimado
γ_1	0,9700	0,9702

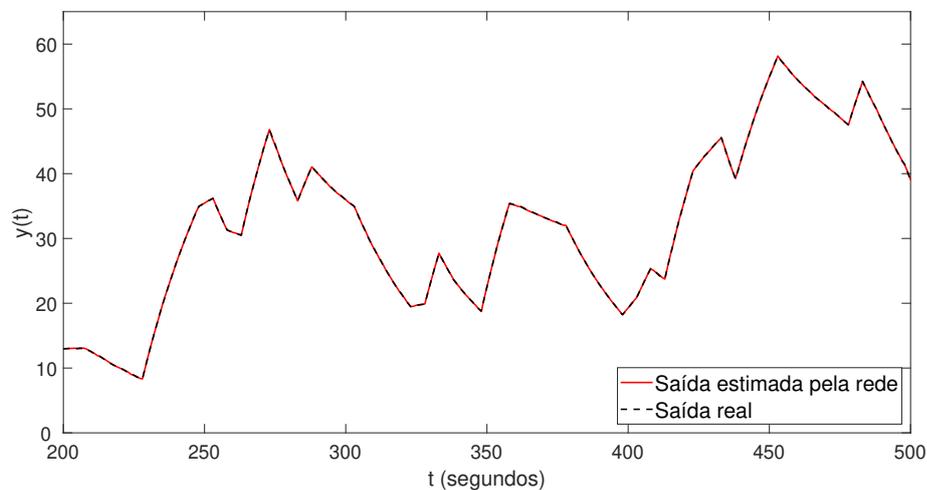
Fonte: Autoria própria.

a partir das matrizes de pesos apresentadas no Apêndice A, então

$$\hat{y}(t) = 0,9702y(t - 1) + \Psi(u(t - d - 1)) \quad (5.5)$$

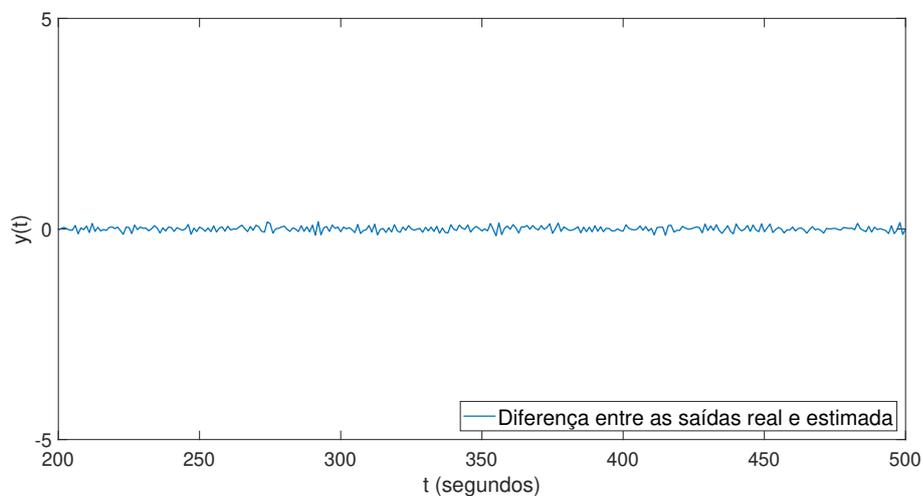
O erro médio quadrático obtido com a estrutura escolhida foi de 0,00466. Uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido para o teste realizado pode ser vista na Figura 5.14 para um intervalo específico do conjunto de teste. Na Figura 5.15 é apresentado o gráfico dos valores de erro de estimação para o subconjunto de teste adotado para comparação.

Figura 5.14: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 1).



Fonte: Autoria própria.

Figura 5.15: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 1).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Para fins de comparação, foi implementado o método de identificação apresentado por (STANISŁAWSKI; LATAWIEC, 2008), no qual são utilizadas funções de base ortonormais e radiais (OBF/RBF). Assim como definido no trabalho em questão, foi utilizada uma quantidade $M = 6$ de funções de *Laguerre* para o caso das OBF, e $m = 8$ funções de base radiais.

Dessa forma, sendo as funções de *Laguerre*, $L_i(q, p)$ definidas em termos do polo p como mostrado na equação (5.6)

$$L_i(q, p) = \frac{\sqrt{1-p^2}}{q-p} \left[\frac{1-pq}{q-p} \right]^{i-1} \quad i = 1, \dots, M \quad (5.6)$$

Com relação ao polo p das função de *Laguerre*, para cada caso, foi feito o ajuste de maneira iterativa por meio várias execuções de modo que a cada execução seu valor foi alterado de 0,05, sendo adotado como resultado da execução do método o menor valor do erro médio quadrático em relação ao conjunto de teste após as várias execuções com o ajuste do valor do polo.

Já no que se refere às funções de base radiais, $\phi_j(u(t))$, definidas como

$$\phi_j(u(t)) = \exp(-b_j \|u(t)\|^2), \quad j = 1, \dots, m \quad (5.7)$$

a definição dos parâmetros b_j foi realizada de maneira empírica, tendo em vista que, de acordo com (STANISŁAWSKI; LATAWIEC, 2008), essa escolha tem pouca influência na precisão do método. Assim, foram definidos os valores mostrados na Tabela 5.4. Foi

Tabela 5.4: Parâmetros utilizados nas RBF.

j	1	2	3	4	5	6	7	8
b_j	0,05	0,1	0,2	0,4	0,8	1	2	4

Fonte: Autoria própria.

utilizado um valor para $p = 0,1$ para as funções de *Laguerre*.

Considerando a estrutura de modelo adotada por (STANISŁAWSKI; LATAWIEC, 2008), apresentada na equação (5.8)

$$\hat{y}(t) = - \sum_{i=1}^M c_i L_i(q^{-1}, p) y(t) + \sum_{j=1}^m w_j \phi_j(t - d) \quad (5.8)$$

foi utilizado um o método dos mínimos quadrados recursivos para o ajuste dos coeficientes c_i e w_j a partir dos dados de entrada e saída adotados para este caso de estudo. Os resultados obtidos são mostrados na Tabelas 5.5 e 5.6.

Tabela 5.5: Coeficientes OBF obtidos (Caso 1).

i	1	2	3	4	5	6
c_i	-0,54961	-0,29132	-0,10616	-0,00858	0,02987	0,04250

Fonte: Autoria própria.

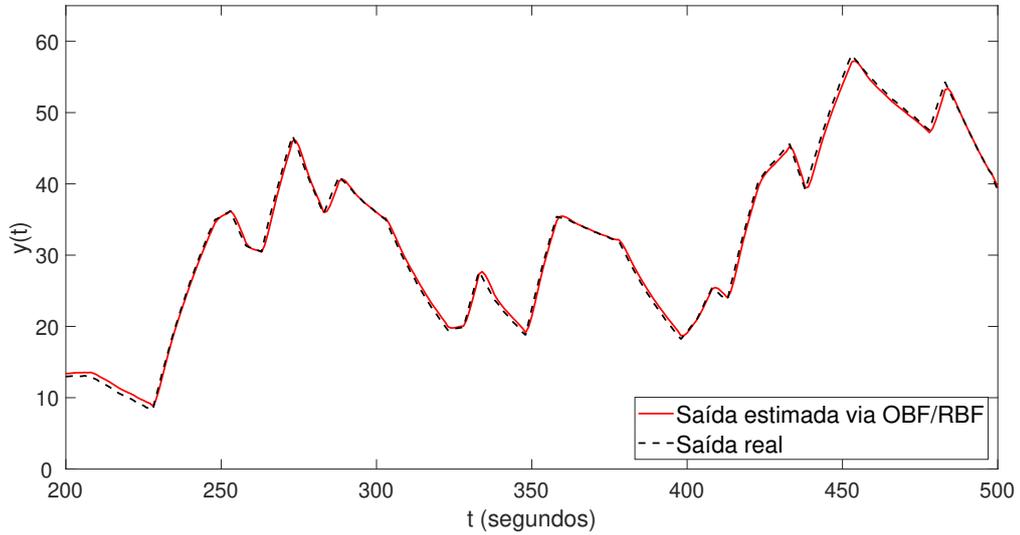
Tabela 5.6: Coeficientes RBF (Caso 1).

j	1	2	3	4	5	6	7	8
w_j	0,17647	0,17106	0,16074	0,14189	0,11050	0,09748	0,05169	0,01346

Fonte: Autoria própria.

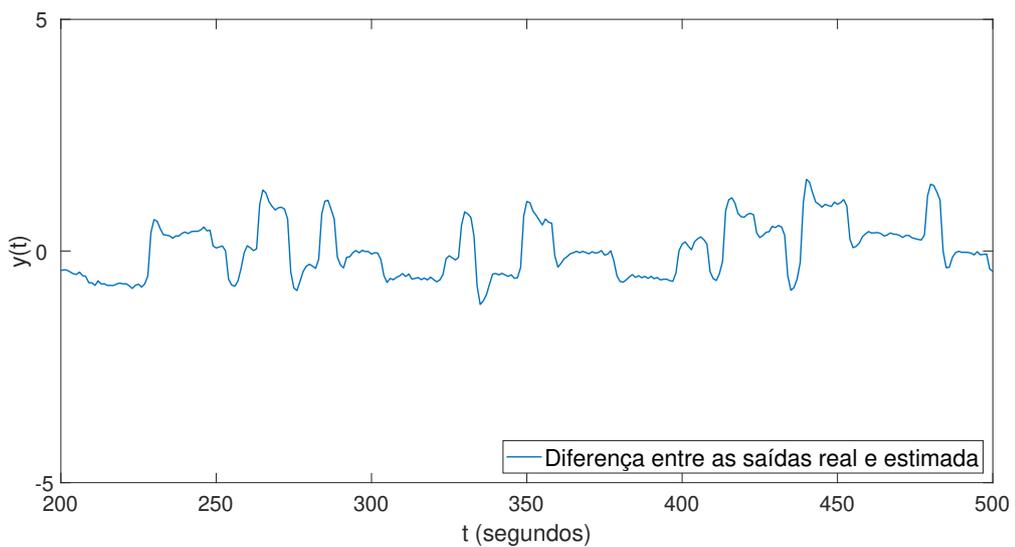
Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 0,37020$. Na Figura 5.16 são apresentados os gráficos da saída real do sistema e da saída estimada a partir do modelo OBF/RBF obtido para um subconjunto dos dados de teste. Além disso, na 5.17 são apresentados graficamente os valores de erro de estimação para um intervalo específico do conjunto de teste.

Figura 5.16: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 1).



Fonte: Autoria própria.

Figura 5.17: Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 1).



Fonte: Autoria própria.

5.4.2 Caso de estudo 2 (*Hammerstein*)

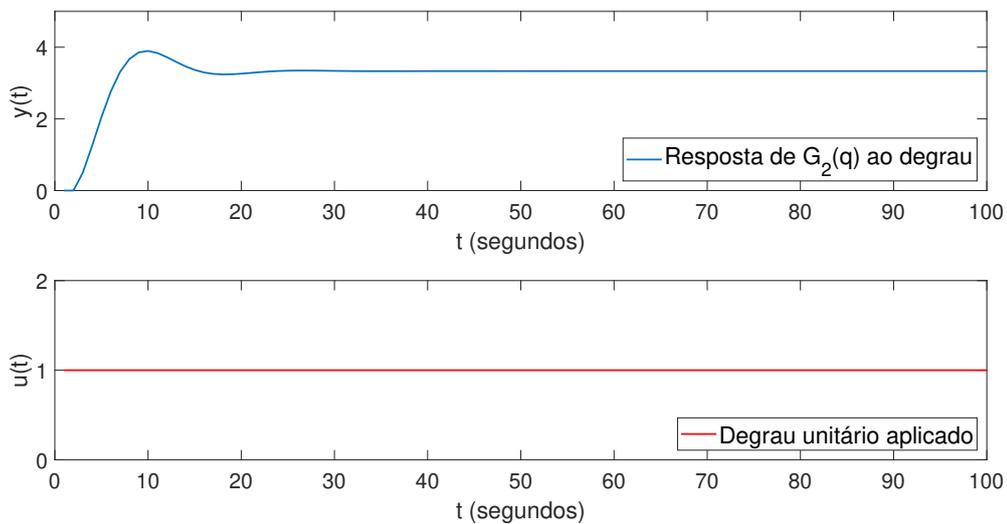
Considerando agora um sistema de *Hammerstein* formado por um bloco linear $G_2(q)$ e a mesma não linearidade $f_H(\cdot)$ utilizada no caso de estudo anterior. Onde $G_2(q)$ é uma função de segunda ordem adaptada de (ZHU, 2001). As funções em questão são dadas de acordo com as equações (5.9) e (5.10), respectivamente.

$$G_2(q) = (q^{-2}) \frac{0,5}{1 - 1,5q^{-1} + 0,65q^{-2}} \quad (5.9)$$

$$f_H(v) = v + 3v^2 + 2v^3 \quad (5.10)$$

Na Figura 5.18 é mostrada a resposta de $G_2(q)$ a um impulso unitário.

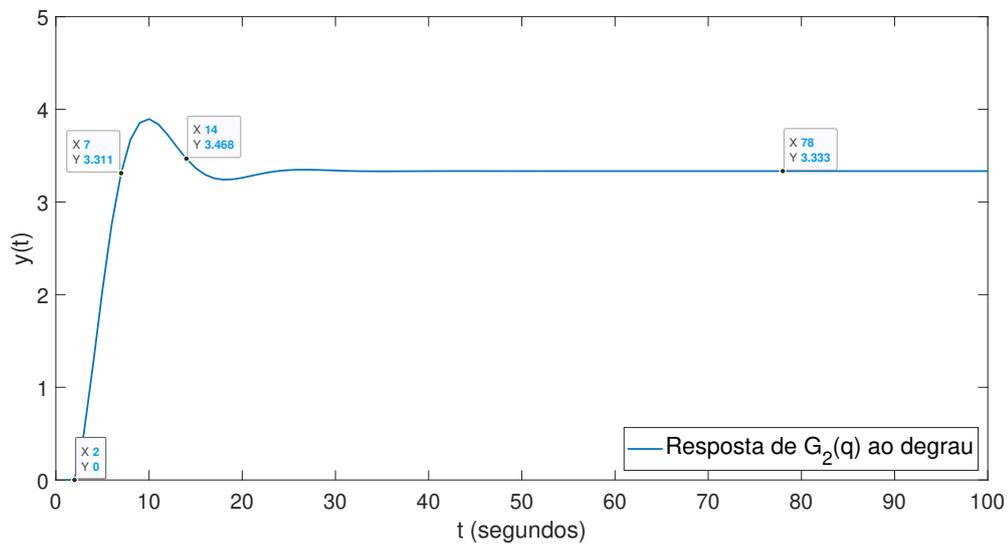
Figura 5.18: Resposta de $G_2(q)$ a um degrau unitário



Fonte: Autoria própria.

Com relação ao sinal de excitação, considerando a resposta ao degrau de $G_2(q)$, assim como mostrado na Figura 5.19, tem-se um valor de regime igual a 3,33. Diante da informações disponíveis foi adotado como referência do quão ingreme é a subida do sinal o período de tempo em que o sinal parte de 0% até aproximadamente 100% do seu valor de regime. Este tempo entre $t = 2$ e $t = 7$ foi utilizado para a definição do tempo de ciclo λ do sinal PRBS, de modo que $\lambda \leq (7 - 2)/5$. Dessa forma, para um tempo de amostragem $T_s = 1$ segundo, foi escolhido $\lambda = 1 T_s$. Além disso, ainda de acordo com a Figura 5.19, a partir do $t = 14$ o sinal não varia mais do que 5% do seu valor de regime permanente. Desse modo, o período do sinal de excitação foi definido com base na consideração de $T_p \approx 1,5 \times 14 T_s$, assim, $T_p = 21 T_s$. Para a obtenção de uma quantidade de pontos de dados equivalente ao caso anterior, foi escolhido um número de períodos $M = 400$.

Figura 5.19: Resposta ao degrau de $G_2(q)$ destacando-se os tempos utilizados para a definição do sinal de excitação.



Fonte: Autoria própria.

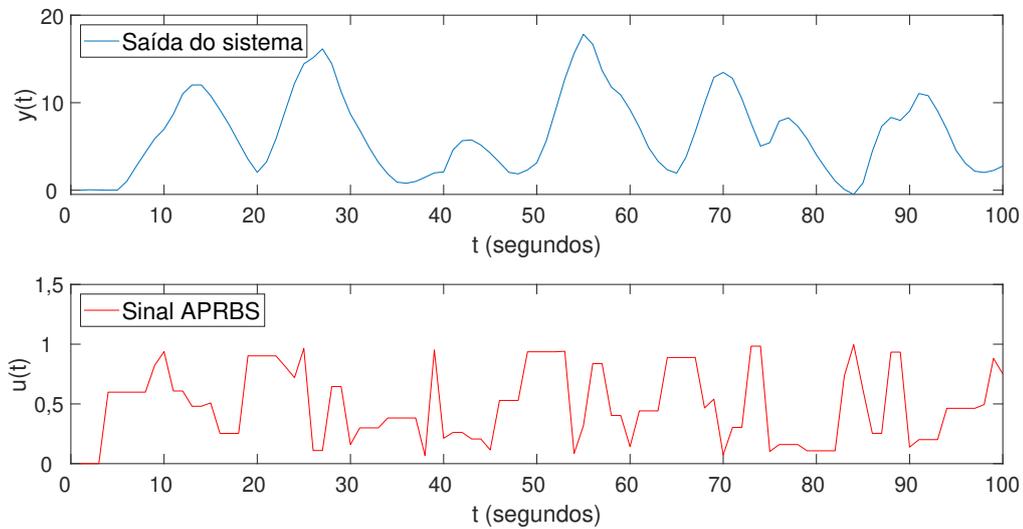
A partir dos parâmetros estabelecidos foi utilizada a função *idinput()* do *software* MATLAB para gerar um sinal PRBS e, a partir dele, gerar um sinal APRBS de acordo com o procedimento descrito na seção 5.2. Quanto ao sinal de excitação obtido, embora tenha sido especificado um período $T_P = 21 T_s$, a função utilizada fez a substituição desse valor pelo mais próximo dentre os valores possíveis de período, neste caso $T_P = 15 T_s$. Dessa forma, foi gerado um sinal PRBS a partir dos parâmetros

- $\lambda = 1 T_s$;
- $T_P = 15 T_s$;
- $M = 400$.

com um total de 6000 pontos de dados.

Com relação a obtenção do sinal APRBS, foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 1$, respectivamente, assim como no caso de estudo anterior. Uma amostra do sinal APRBS resultante para uma faixa de amplitude $[0, 1]$, bem como a saída do sistema incluindo o ruído adicionado, são mostrados na Figura 5.20

Figura 5.20: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 2)



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Hammerstein* composto por uma função linear de segunda ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.7, onde $u(t - d - 2)$ é o sinal de excitação atrasado de

Tabela 5.7: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 2).

Entradas da parte linear	$y(t - 1)$	$y(t - 2)$
Entradas da parte multicamadas	$u(t - d - 2)$	–

Fonte: Autoria própria.

$d + 2$ instantes de tempo, $y(t - 1)$ e $y(t - 2)$ correspondem ao sinal de saída atrasado de um e dois instantes de tempo, respectivamente. Neste caso d é um atraso conhecido e para a simulação foi adotado $d = 2$. Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.20 para treinar da rede e posteriormente foi calculado o erro médio quadrático sobre o conjunto de teste para ser usado como métrica de desempenho. Esse procedimento foi realizado para diferentes escolhas projeto no que se refere à quantidade de camadas ocultas e o número de neurônios

por camada. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.8.

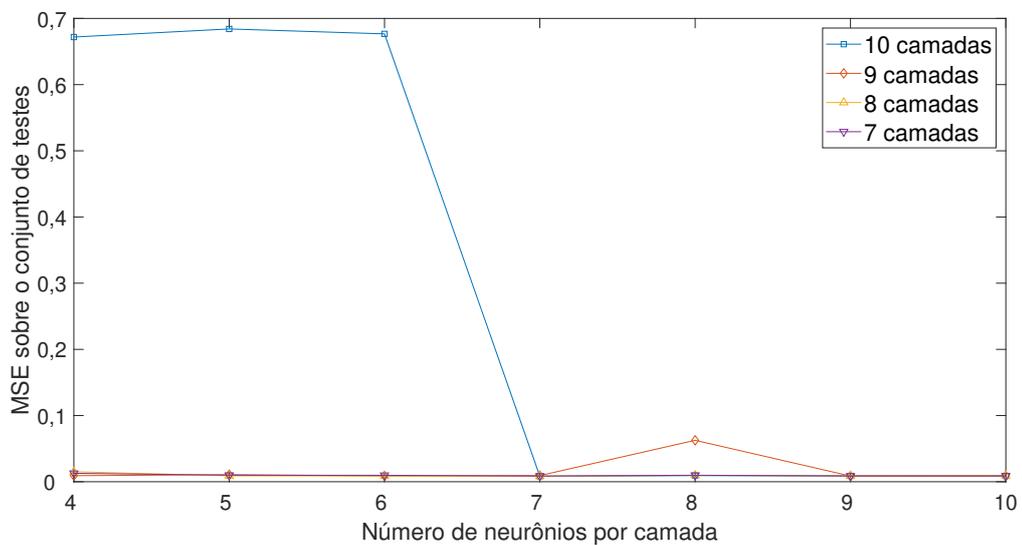
Tabela 5.8: Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 2).

		Número de neurônios por camada						
		4	5	6	7	8	9	10
Número de camadas ocultas	7	0,01273	0,00989	0,00972	0,00864	0,00982	0,00871	0,00879
	8	0,01518	0,00889	0,00863	0,00824	0,00969	0,00890	0,00912
	9	0,00913	0,01066	0,00868	0,00930	0,06259	0,00860	0,00878
	10	0,67189	0,68409	0,67669	0,00829	0,00930	0,00881	0,00874

Fonte: Autoria própria.

Na Figura 5.21 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede. A partir

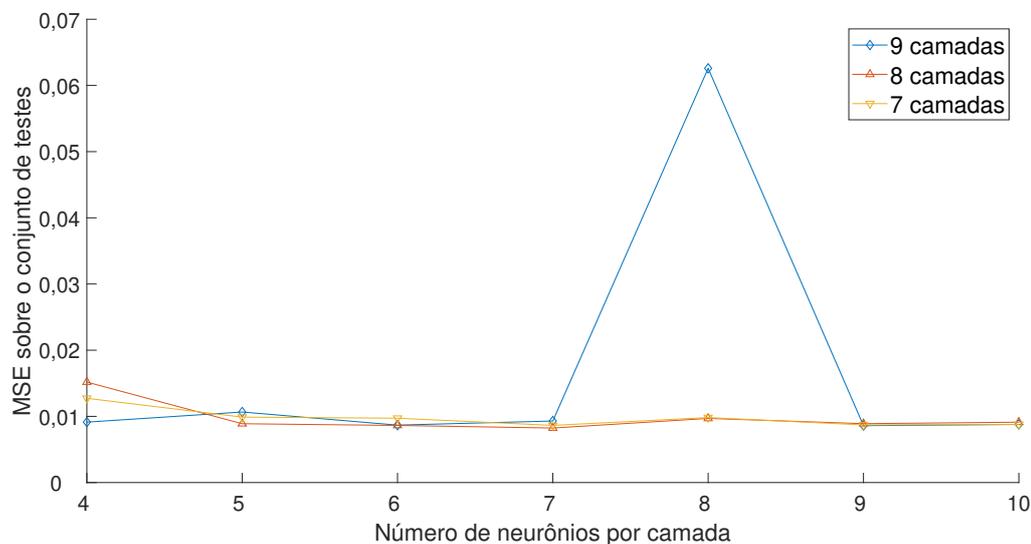
Figura 5.21: Gráfico de valores MSE para diferentes quantidades de camadas ocultas e número de neurônios (Caso 2).



Fonte: Autoria própria.

da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que, de maneira geral, os valores de erro decrescem com o aumento do número de neurônios por camada, porém verifica-se, para este caso uma maior dificuldade generalização para 10 camadas ocultas, de modo que os melhores resultados foram alcançados para o caso em que foram utilizadas 7, 8 e 9, camadas ocultas. Na Figura 5.22, são mostrados os valores de erro obtidos para diferentes número de neurônios por camada para os casos em que foram utilizadas as quantidade de camadas ocultas em

Figura 5.22: Gráfico de erro para diferentes números de neurônios para os caso de 7, 8 e 9 camadas (Caso 2).



Fonte: Autoria própria.

questão. Com base no comportamento do erro apresentado na Figura 5.22, foi selecionada como modelo para o sistema de *Hammerstein* deste caso de estudo, a estrutura com 8 camadas ocultas, além das camadas de entrada e saída, e 7 neurônios por camada, além dos termos de polarização unitários. Essa escolha se deu em razão de essa ser uma combinação capaz fornecer um o menor erro sobre o conjunto de teste assim como mostrado na Tabela 5.2, e também resultar em uma quantidade de parâmetros semelhante a outras combinações com resultados semelhantes. Para a estrutura selecionada, são obtidos $(8 - 1) \times (7 \times 7) = 343$ parâmetros relativos às camadas intermediárias. Dessa forma, a estrutura de rede escolhida para o caso este caso de estudo é como ilustrada na Figura 5.24, onde L_{in} , L_1 , L_2, \dots, L_7 , L_8 , L_{out} são as camadas da rede.

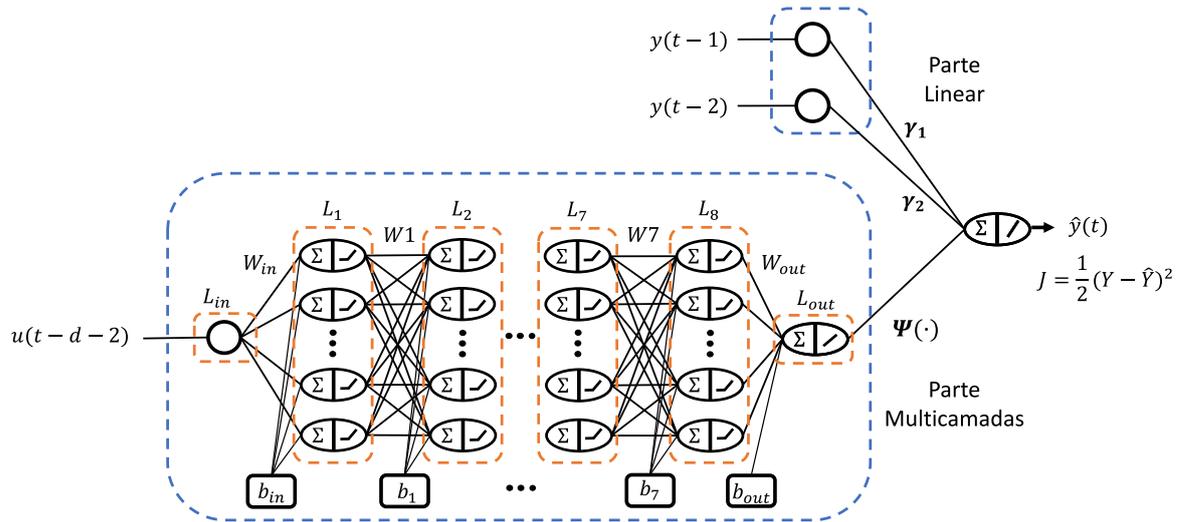
De acordo com a estrutura de modelo apresentada na equação 5.11, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

$$\hat{y}(t) = \gamma_1 y(t - 1) + \gamma_2 y(t - 2) + \Psi(u(t - d - 2)) \quad (5.11)$$

tem-se como coeficientes da parte linear, os parâmetros γ_1 e γ_2 , que correspondes, para este caso de estudo, aos negativo dos coeficientes de primeira e segunda ordem do denominador da função $G_2(q)$, isto é, 1,5 e $-0,65$, respectivamente. Na Tabela 5.9 são mostrados o valor real desse coeficiente e o valor estimado a partir da estrutura de rede escolhida.

Para $\gamma_1 = 1,4970$ e $\gamma_2 = -0,6481$ e tendo em vista que o mapeamento não linear $\Psi(\cdot)$ pode ser obtido a partir das matrizes de pesos presentes no Apêndice B, de modo

Figura 5.23: Ilustração da estrutura de rede adotada (Caso 2).



Fonte: Autoria própria.

Tabela 5.9: Valores reais e estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 2).

	Valor real	Valor estimado
γ_1	1,5000	1,4970
γ_2	-0,6500	-0,6481

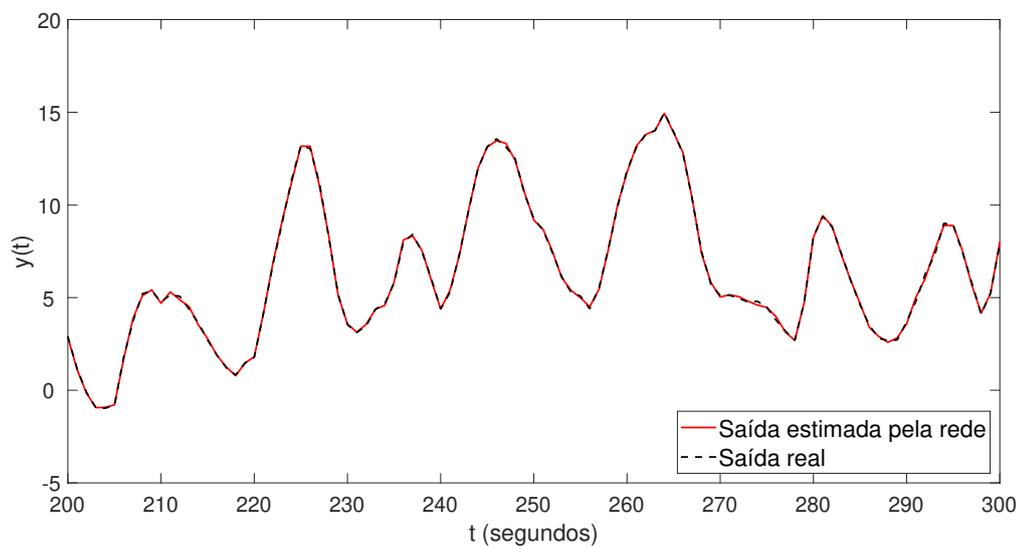
Fonte: Autoria própria.

que

$$\hat{y}(t) = 1,4970y(t-1) - 0,6481y(t-2) + \Psi(u(t-d-2)) \quad (5.12)$$

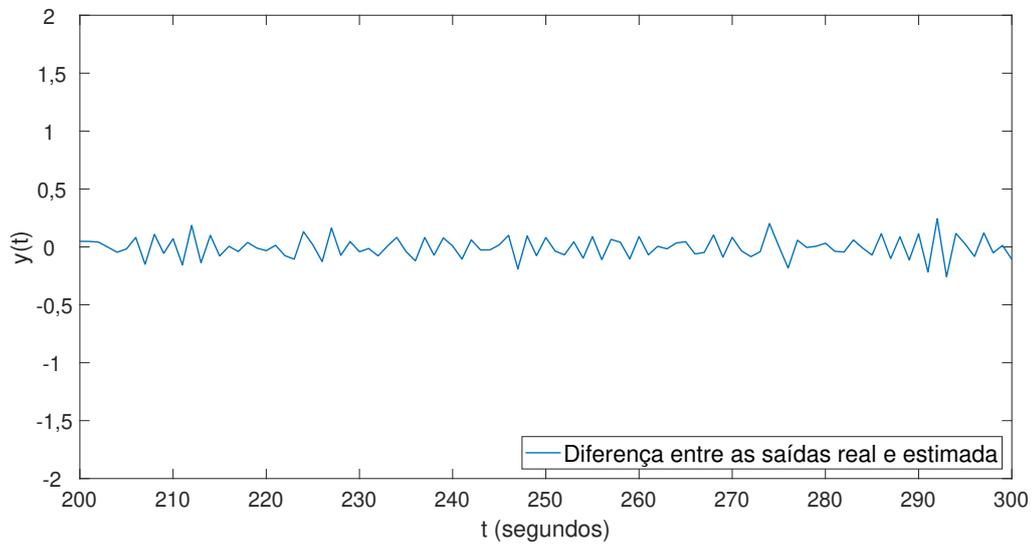
O erro médio quadrático obtido no teste foi de 0.00824. Uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido pode ser vista na Figura 5.24. Além disso, na Figura 5.25 é apresentado o conjunto de valores de erro de estimação para o subconjunto de dados de teste adotado para comparação.

Figura 5.24: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 2).



Fonte: Autoria própria.

Figura 5.25: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 2).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Para todos os casos de estudo foram utilizadas as mesmas quantidades, $M = 6$, de funções de *Laguerre* e $m = 8$ funções de base radiais. Foi também usada a mesma estratégia de ajuste do valor do polo p , resultando para este caso em $p = 0, 1$.

Para os dados de entrada e saída deste caso de estudo, foram ajustados os coeficientes c_i e w_j mostrados na Tabelas 5.10 e 5.11.

Tabela 5.10: Coeficientes OBF obtidos (Caso 2).

i	1	2	3	4	5	6
c_i	-0,85867	-0,12270	0,11113	0,02800	-0,03731	0,02894

Fonte: Autoria própria.

Tabela 5.11: Coeficientes RBF (Caso 2).

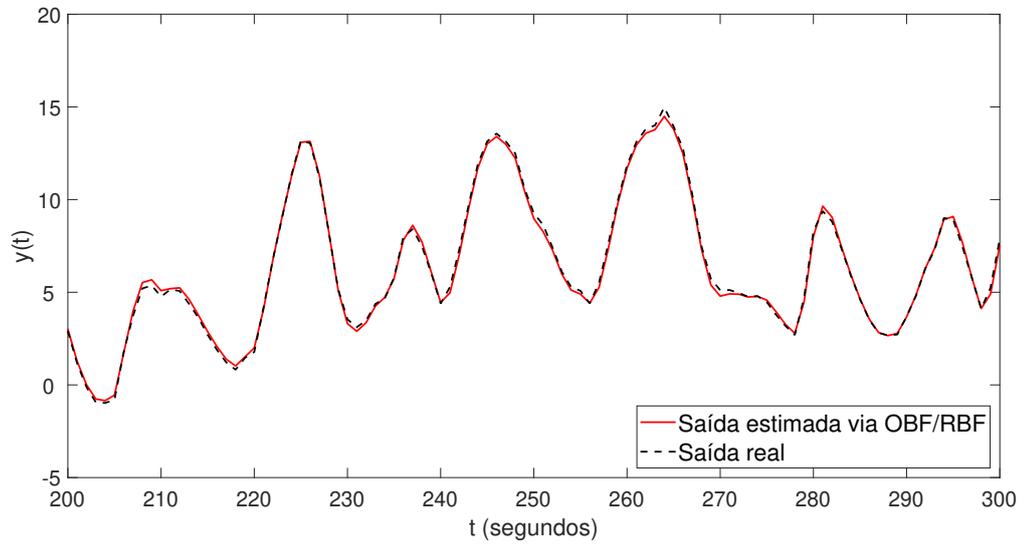
j	1	2	3	4	5	6	7	8
w_j	0,10258	0,09705	0,08661	0,06801	0,03846	0,02682	-0,01003	-0,03141

Fonte: Autoria própria.

Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 0,04111$. Nas Figuras 5.26 e 5.27 são apresentadas a comparação entre a saída

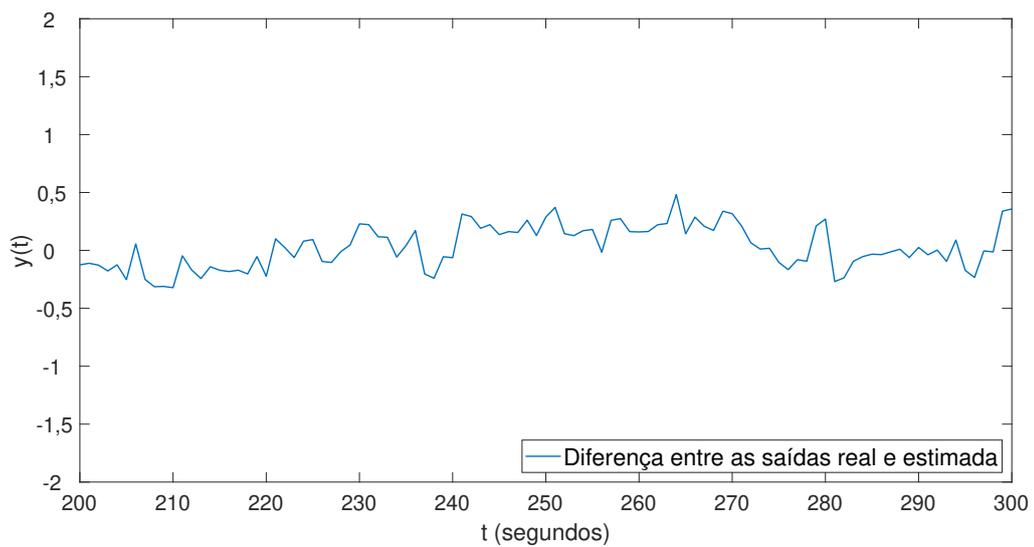
real do sistema e a saída estimada a partir do método OBF/RBF, além do conjunto de valores de erro estimacão para o intervalo de dados adotado.

Figura 5.26: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 2)



Fonte: Autoria própria.

Figura 5.27: Erro de estimacão utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 2).



Fonte: Autoria própria.

5.4.3 Caso de estudo 3 (*Hammerstein*)

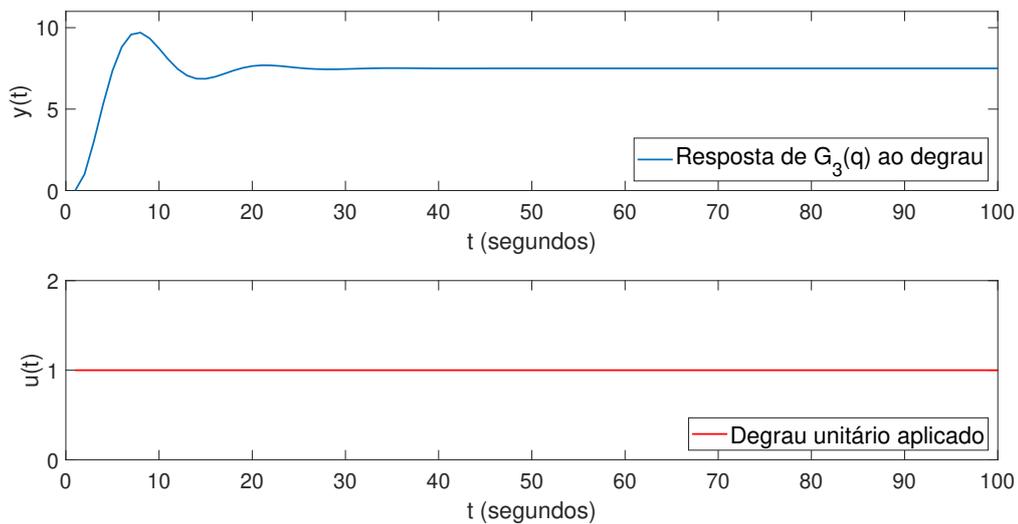
Considerando agora um sistema de *Hammerstein* formado por um bloco linear $G_3(q)$ e a mesma não linearidade $f_H(\cdot)$ utilizada nos casos de estudo anteriores. Onde $G_3(q)$ é uma função de segunda ordem com um zero utilizada em (ZHU, 2001). As funções em questão são dadas de acordo com as equações (5.13) e (5.14), respectivamente.

$$G_3(q) = (q^{-1}) \frac{1 + 0,5q^{-1}}{1 - 1,5q^{-1} + 0,7q^{-2}} \quad (5.13)$$

$$f_H(v) = v + 3v^2 + 2v^3 \quad (5.14)$$

Na Figura 5.28 é mostrada a resposta de $G_3(q)$ a um impulso unitário.

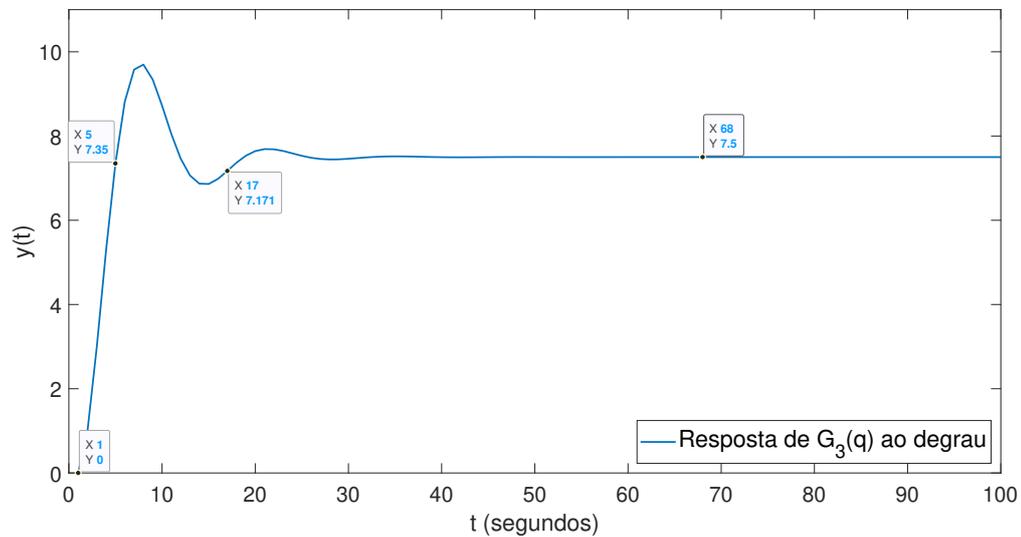
Figura 5.28: Resposta de $G_3(q)$ a um degrau unitário



Fonte: Autoria própria.

Com relação ao sinal de excitação, considerando a resposta ao degrau de $G_3(q)$, assim como mostrado na Figura 5.29, tem-se um valor de regime igual a 7,5. Diante das informações disponíveis foi adotado como referência do quão ingreme é a subida do sinal o período de tempo em que o sinal parte de 0% até aproximadamente 100% do seu valor de regime. Este tempo entre $t = 1$ e $t = 5$ indica que a subida é ainda mais ingreme do que para o caso de estudo anterior. Dessa forma, para um tempo de amostragem $T_s = 1$ segundo, foi escolhido $\lambda = 1 T_s$, que é o menor valor possível para λ e que permite que o sinal varie de amplitude a cada passo de tempo. Além disso, ainda de acordo com a Figura 5.29, a partir do $t = 17$ o sinal não varia mais do que 5% do seu valor de regime permanente. Desse modo, o período do sinal de excitação foi definido com base na consideração de $T_p \approx 1,5 \times 17 T_s$, assim, $T_p = 25 T_s$. Para a obtenção de uma quantidade de pontos de

Figura 5.29: Resposta ao degrau de $G_3(q)$ destacando-se os tempos utilizados para a definição do sinal de excitação.



Fonte: Autoria própria.

dados equivalente ao caso anterior, foi escolhido um número de períodos $M = 400$.

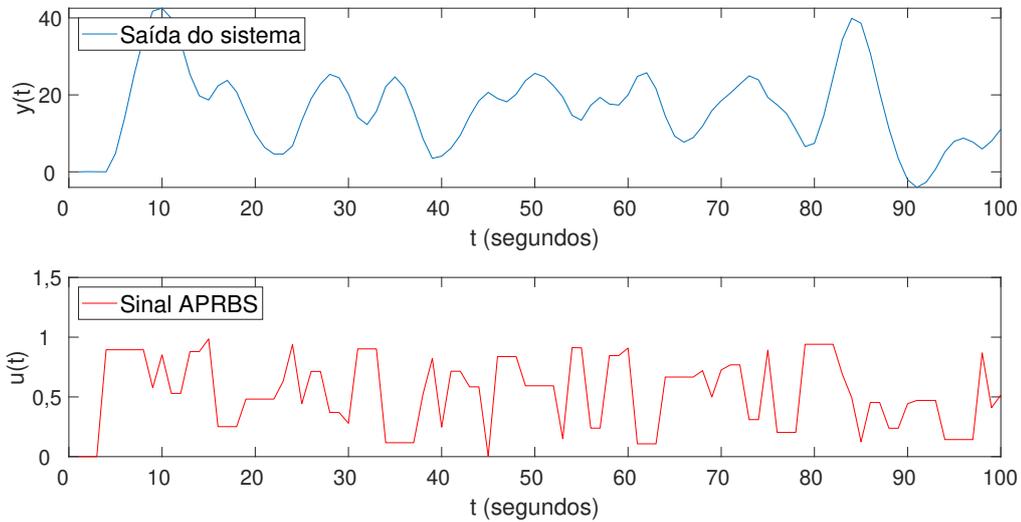
A partir dos parâmetros estabelecidos foi utilizada a função *idinput()* do *software* MATLAB para gerar um sinal PRBS e, a partir dele, gerar um sinal APRBS de acordo com o procedimento descrito na seção 5.2. Quanto ao sinal de excitação obtido, embora tenha sido especificado um período $T_P = 25 T_s$, a função utilizada fez a substituição desse valor pelo mais próximo dentre os valores possíveis de período, neste caso $T_P = 15 T_s$. Dessa forma, foi gerado um sinal PRBS a partir dos parâmetros

- $\lambda = 1 T_s$;
- $T_P = 15 T_s$;
- $M = 400$.

com um total de 6000 pontos de dados.

Com relação a obtenção do sinal APRBS, foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 1$, respectivamente, assim como no caso de estudo anterior. Uma amostra do sinal APRBS resultante para uma faixa de amplitude $[0, 1]$, bem como a saída do sistema incluindo o ruído adicionado, são mostrados na Figura 5.30

Figura 5.30: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 3).



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Hammerstein* composto por uma função linear de segunda ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.12, onde $u(t - d - 1)$ e $u(t - d - 2)$ representam o sinal

Tabela 5.12: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 3)

Entradas da parte linear	$y(t - 1)$	$y(t - 2)$
Entradas da parte multicamadas	$u(t - d - 1)$	$u(t - d - 2)$

Fonte: Autoria própria.

de excitação atrasado de $d + 1$ e $d + 2$ passos de tempo, respectivamente. Já $y(t - 1)$ e $y(t - 2)$ correspondem ao sinal de saída atrasado de 1 e 2 instantes de tempo, respectivamente. Foi utilizado um atraso conhecido $d = 2$ para a realização da simulação. Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.30 para treinar da rede e posteriormente foi calculado o erro médio quadrático sobre o conjunto de teste para ser usado como métrica de desempenho. Esse procedimento foi realizado para diferentes escolhas projeto no que se refere à quantidade

de camadas ocultas e o número de neurônios por camada. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.13.

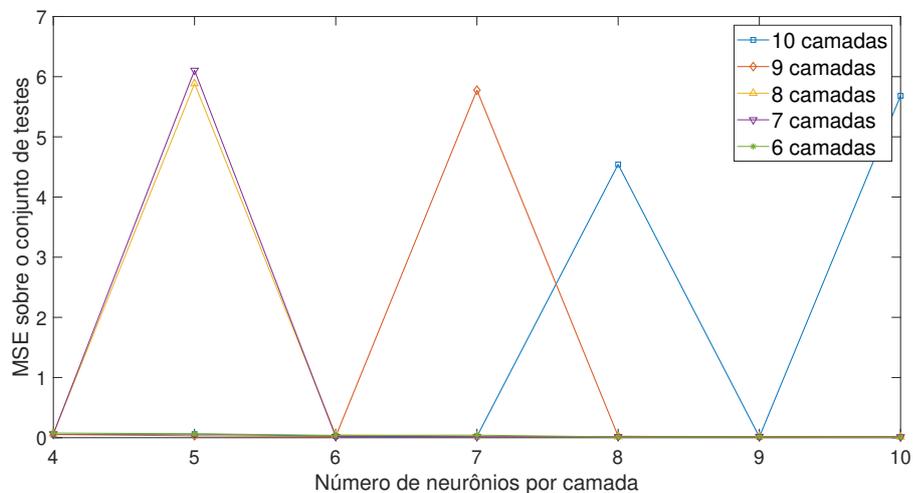
Tabela 5.13: Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 3).

		Número de neurônios por camada						
		4	5	6	7	8	9	10
Número de camadas ocultas	6	0,07893	0,06589	0,03962	0,04338	0,01324	0,01873	0,01037
	7	0,06288	6,10524	0,01715	0,01514	0,01559	0,01921	0,01453
	8	0,06444	5,88539	0,04677	0,02401	0,01651	0,01950	0,01323
	9	0,05254	0,03222	0,02160	5,77598	0,02413	0,01400	0,01348
	10	0,05352	0,06065	0,02658	0,02053	4,54189	0,01309	5,68167

Fonte: Autoria própria.

Na Figura 5.31 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede. A partir

Figura 5.31: Gráfico de Valores de erro para diferentes quantidades de camadas e neurônios (Caso 3).



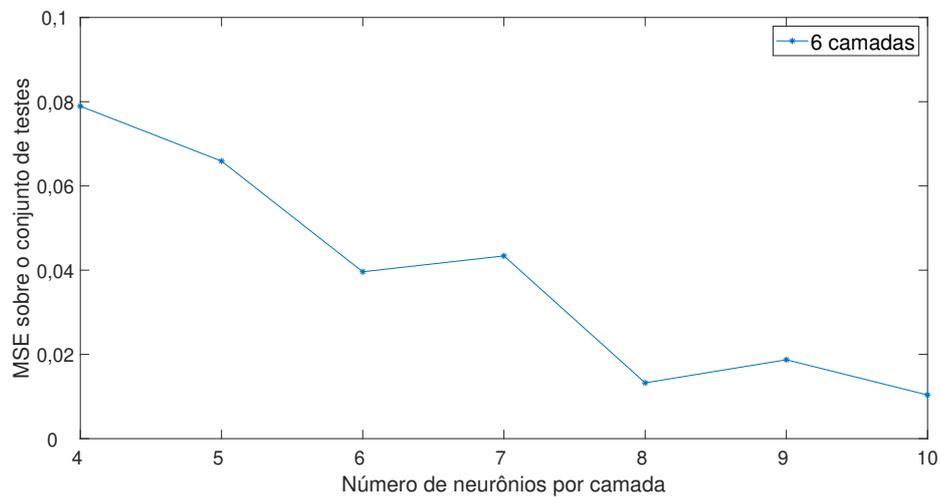
Fonte: Autoria própria.

da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que as estruturas com 7, 8, 9 e 10 camadas apresentaram dificuldade de generalizar os dados de teste para algumas quantidades de neurônios. Em particular, para as redes com 10 camadas ocultas tem-se um aumento nos valores de erro à medida que são utilizados mais neurônios por camada, o que é um indício de sobreajuste (*overfitting*).

Desse modo, foram também testadas estruturas com 6 camadas, as quais forneceram resultados mais consistentes e com o menor valor de erro alcançado para 10 neurônios por

camada oculta. Na Figura 5.32, são mostrados os valores de erro obtidos para diferentes número de neurônios para 6 camadas ocultas. Com base no comportamento do erro

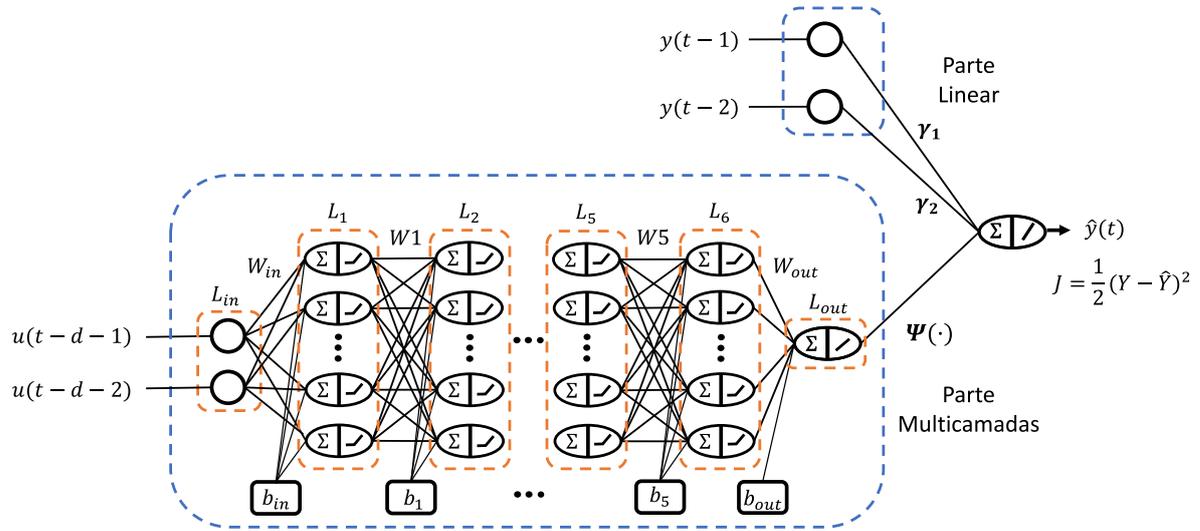
Figura 5.32: Valores de erro para diferentes números de neurônios para de 6 camadas (Caso 3)



Fonte: Autoria própria.

apresentado na Figura 5.32, foi selecionada como modelo para o sistema de *Hammerstein* deste caso de estudo, a estrutura com 6 camadas ocultas, além das camadas de entrada e saída, e 10 neurônios por camada, além dos termos de polarização unitários. O erro obtido sobre o conjunto de teste foi $MSE = 0,00103$ assim como mostrado na Tabela 5.13. Dessa forma, a estrutura de rede escolhida para o caso este caso de estudo é como ilustrada na Figura 5.33, onde L_{in} , L_1 , L_2, \dots, L_7 , L_8 , L_{out} são as camadas da rede.

Figura 5.33: Ilustração da estrutura de rede adotada (Caso 3)



Fonte: Autoria própria.

De acordo com a estrutura de modelo apresentada na equação 5.15, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

$$\hat{y}(t) = \gamma_1 y(t-1) + \gamma_2 y(t-2) + \Psi(u(t-d-1), u(t-d-2)) \quad (5.15)$$

tem-se como coeficientes da parte linear, os parâmetros γ_1 e γ_2 , que correspondes, para este caso de estudo, aos negativo dos coeficientes de primeira e segunda ordem do denominador da função $G_3(q)$, isto é, 1,5 e $-0,7$, respectivamente. Na Tabela 5.14 são mostrados o valor real desse coeficiente e o valor estimado a partir da estrutura de rede escolhida. Para $\gamma_1 = 1,5006$ e $\gamma_2 = -0,69994$. e tendo em vista que o mapeamento não

Tabela 5.14: Valores reais e estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 3).

	Valor real	Valor estimado
γ_1	1,5000	1,5006
γ_2	-0,7000	-0,69994

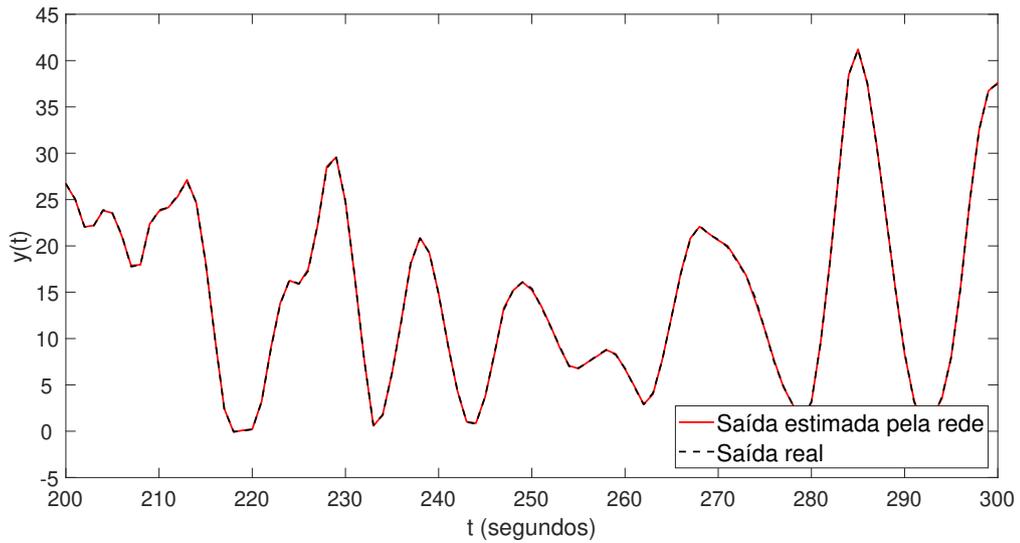
Fonte: Autoria própria.

linear $\Psi(\cdot)$ pode ser obtido a partir das matrizes de pesos presentes no Apêndice C, de modo que

$$\hat{y}(t) = 1,5006y(t-1) - 0,69994y(t-2) + \Psi(u(t-d-1), u(t-d-2)) \quad (5.16)$$

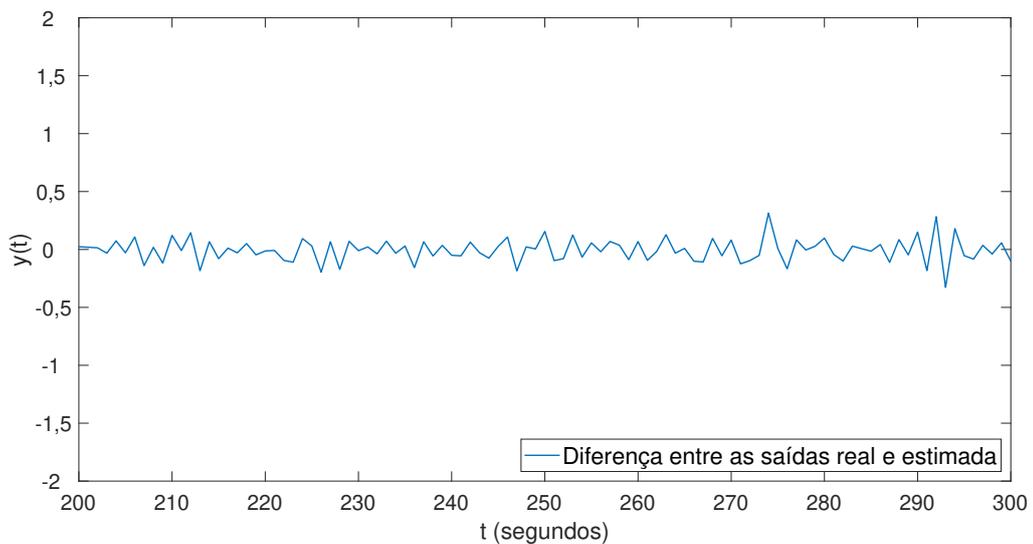
Para o erro médio quadrático obtido de 0,00103, uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido pode ser vista nas Figuras 5.34. Na Figura 5.35 é apresentado o gráfico dos valores de erro de estimação para o subconjunto de teste adotado.

Figura 5.34: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 3).



Fonte: Autoria própria.

Figura 5.35: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 3).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Para todos os casos de estudo foram utilizadas as mesmas quantidades, $M = 6$, de funções de *Laguerre* e $m = 8$ funções de base radiais. Foi também usada a mesma estratégia de ajuste do valor do polo p , resultando para este caso em $p = 0, 1$.

Para os dados de entrada e saída deste caso de estudo, foram ajustados os coeficientes c_i e w_j mostrados na Tabelas 5.15 e 5.16.

Tabela 5.15: Coeficientes OBF obtidos (Caso 3).

i	1	2	3	4	5	6
c_i	-0,94562	0,00913	0,06188	-0,01033	-0,02010	0,02073

Fonte: Autoria própria.

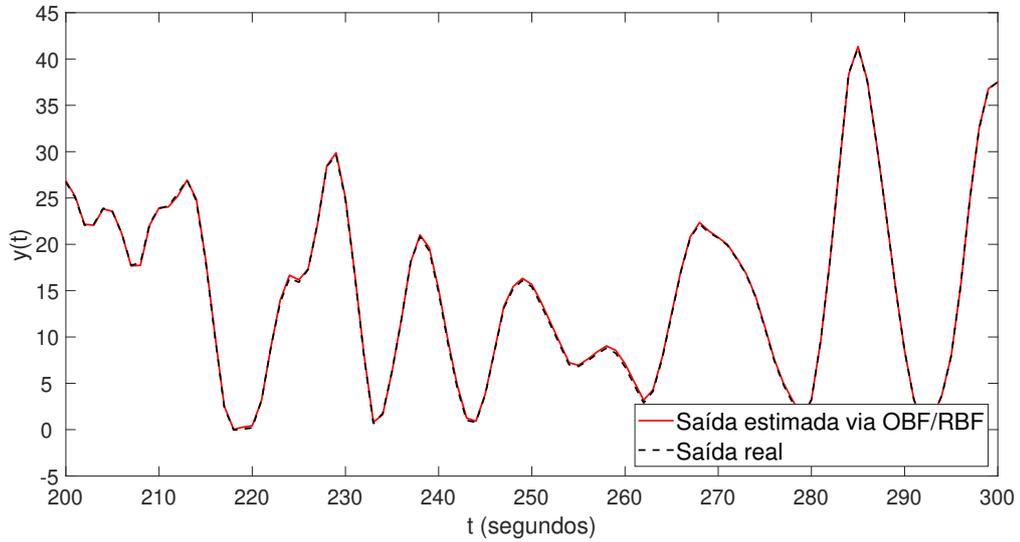
Tabela 5.16: Coeficientes RBF (Caso 3).

j	1	2	3	4	5	6	7	8
w_j	0,11323	0,10774	0,09748	0,07960	0,05240	0,04215	0,01205	-0,00308

Fonte: Autoria própria.

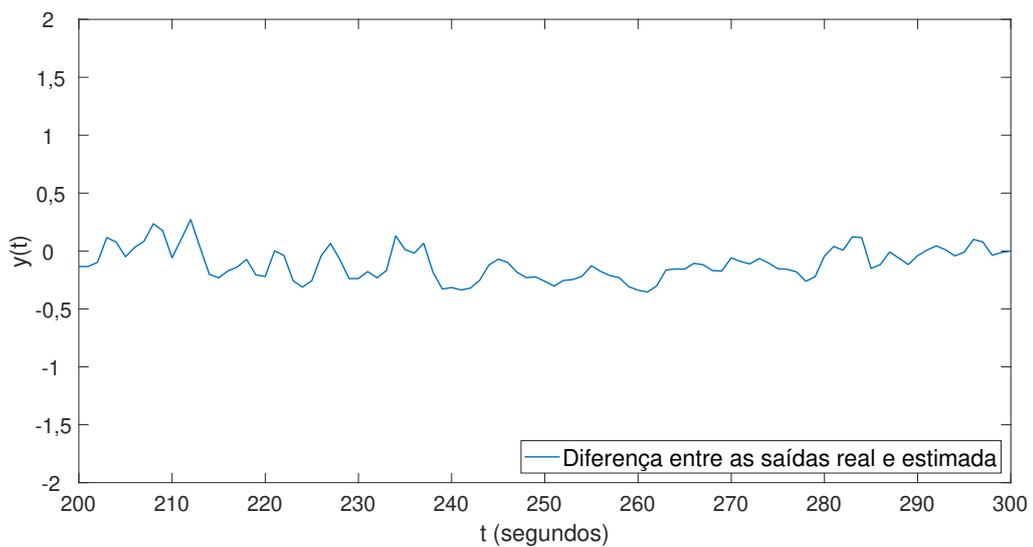
Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 0,04041$. Nas Figuras 5.36 e 5.37 são mostradas a comparação entre a saída real do sistema e a saída estimada a partir do método OBF/RBF, além do conjunto de valores de erro de estimação para um intervalo específico.

Figura 5.36: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 3).



Fonte: Autoria própria.

Figura 5.37: Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 3).



Fonte: Autoria própria.

5.4.4 Caso de estudo 4 (*Wiener*)

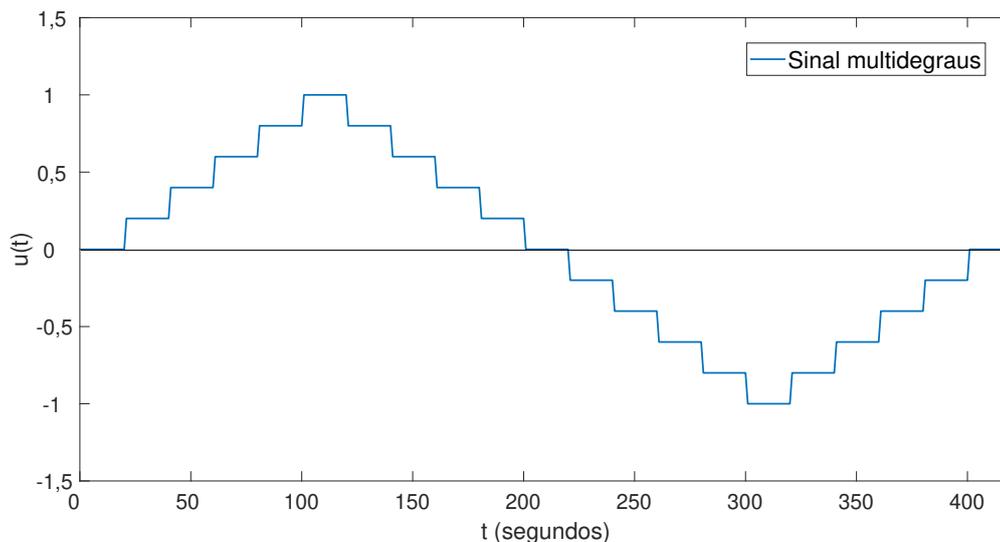
Sendo um sistema de *Wiener* formado por um bloco não linear $f_W(\cdot)$ e por uma função linear $G_1(q)$. Onde $f_W(\cdot)$ e $G_1(q)$ são funções utilizadas em (ZHU, 2001), e são dadas de acordo com as equações (5.17) e (5.18), respectivamente.

$$f_W(v) = v + 2v^2 \quad (5.17)$$

$$G_1(q) = (q^{-1}) \frac{0,5}{1 - 0,97q^{-1}} \quad (5.18)$$

A função $G_1(q)$ é a mesma utilizada no primeiro caso de estudo. Quanto à parte não linear, para checar não linearidade foi utilizado o sinal mostrado na Figura 5.38. Além disso, é possível verificar a característica não linear associada a $f_W(\cdot)$ a partir do gráfico apresentado na Figura 5.39

Figura 5.38: Sinal *multidegraus*.



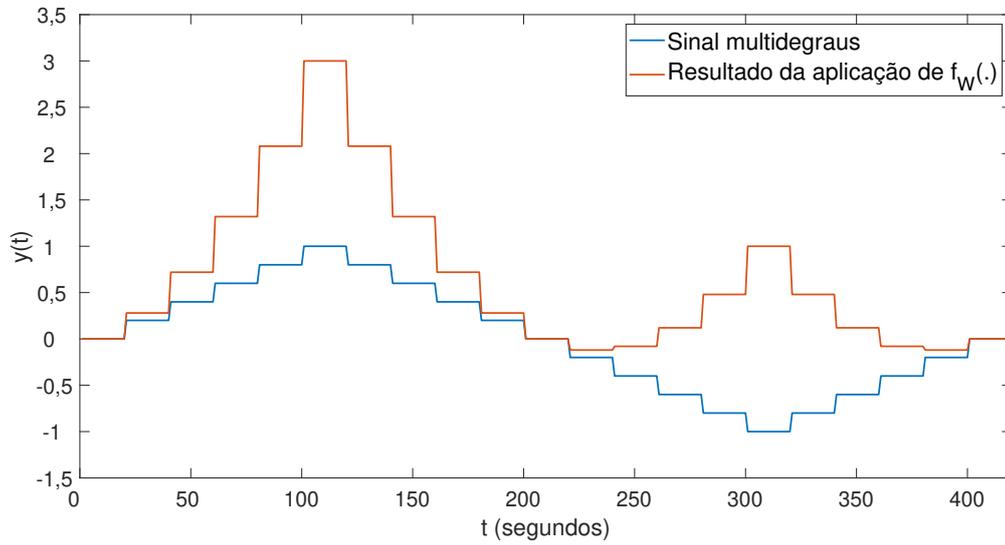
Fonte: Autoria própria.

Desse modo, para valores entre -1 e 1 , a função $f_W(\cdot)$ se comporta de acordo com o gráfico apresentado na Figura 5.40

Com relação ao sinal de excitação, considerando função $G_1(q)$, foram utilizados os mesmos parâmetros do caso de estudo 1 para a geração do sinal PRBS, sendo eles

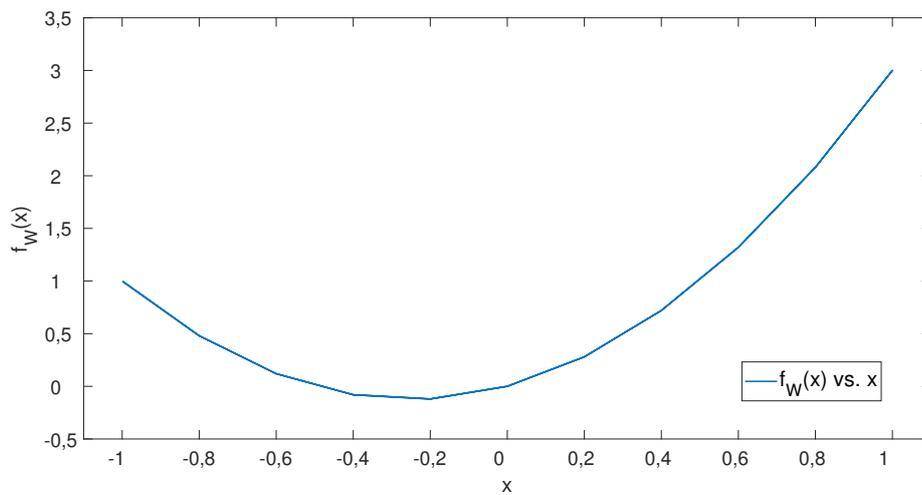
- $\lambda = 5 T_s$;
- $T_P = 75 T_s$;
- $M = 80$.

Figura 5.39: Gráfico comparativo entre o sinal *multidegraus* original e o resultado da aplicação de $f_W(\cdot)$



Fonte: Autoria própria.

Figura 5.40: Comportamento não linear de $f_W(\cdot)$

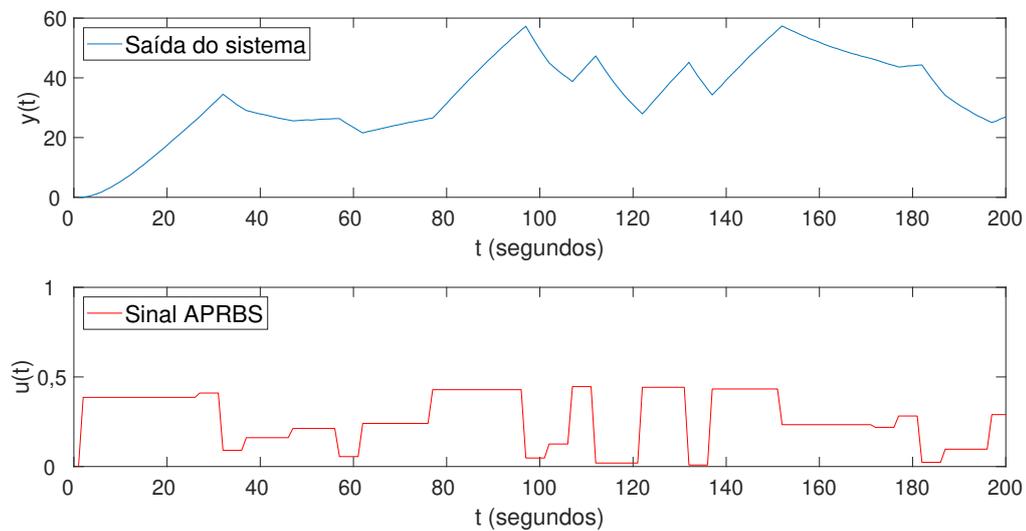


Fonte: Autoria própria.

Para o sinal APRBS, foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 0,5$, respectivamente. Essa escolha se deu de maneira empírica com o intuito de evitar valores de saída excessivamente elevados, visando evitar possíveis problemas de otimização associados ao treinamento da rede.

O resultado foi um sinal APRBS com 6000 pontos de dados. Os primeiros 200 valores do sinal de excitação, bem como a saída correspondente do sistema, incluindo o ruído adicionado, são mostrados na Figura 5.41

Figura 5.41: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 200 pontos de dados (Caso 4).



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Wiener* composto por uma função linear de primeira ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.17, onde $u(t - d - 1)$ é o sinal de excitação atrasado de $d + 1$

Tabela 5.17: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 4).

Entradas da parte linear	$u(t - d - 1)$	$y(t - 1)$
Entrada da parte multicamadas	$y(t - 1)$	–

Fonte: Autoria própria.

instantes de tempo e $y(t - 1)$ é o sinal de saída atrasado de 1 instante de tempo. Neste caso d é um atraso conhecido e para a simulação foi adotado $d = 0$. Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.41 para treinar a rede e foi utilizado o erro médio quadrático sobre o conjunto de teste como métrica de desempenho. Esse procedimento de identificação foi realizado para diferentes escolhas projeto no que se refere à quantidade de camadas ocultas e o número de neurônios por camada. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.18.

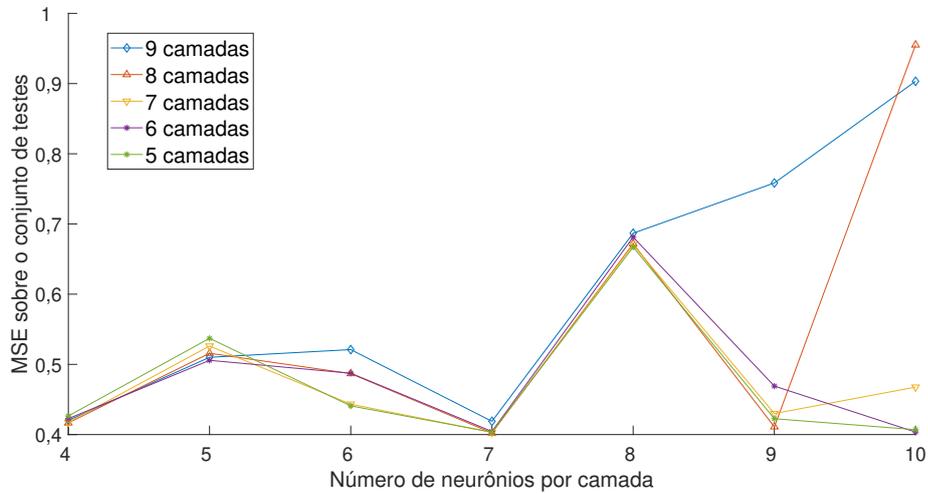
Tabela 5.18: Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 4).

	Número de neurônios por camada							
		4	5	6	7	8	9	10
Número de camadas ocultas	5	0,42647	0,53711	0,44085	0,40346	0,66705	0,42258	0,40701
	6	0,42264	0,50573	0,48739	0,40442	0,68128	0,46906	0,40358
	7	0,41749	0,52613	0,44331	0,40304	0,67039	0,42993	0,46759
	8	0,41662	0,51580	0,48686	0,40227	0,67194	0,41102	0,95508
	9	0,42023	0,51003	0,52105	0,41887	0,68685	0,75840	0,90328

Fonte: Autoria própria.

Na Figura 5.42 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede. A partir da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que, de maneira geral, são alcançados melhores resultados

Figura 5.42: Gráfico de valores de erro para diferentes quantidades de camadas e neurônios (Caso 4).



Fonte: Autoria própria.

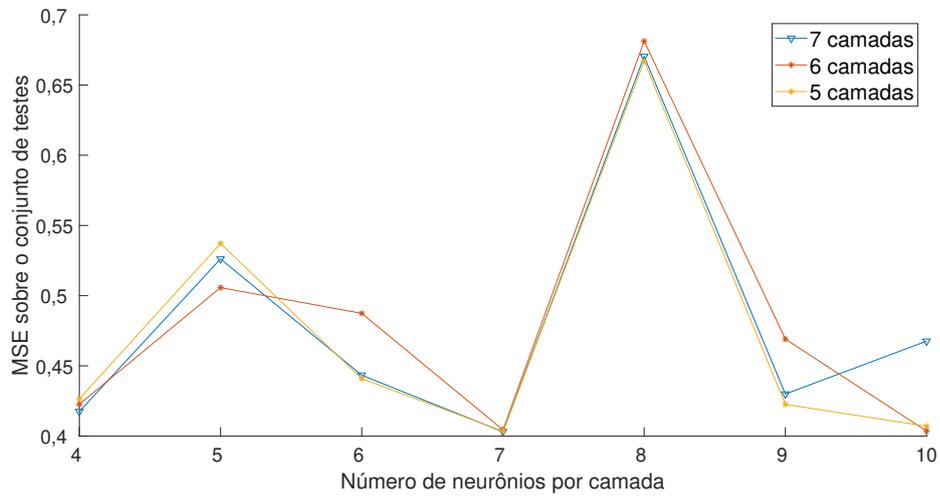
para os casos em que foram utilizadas 5, 6 e 7 camadas ocultas. É importante notar que ocorre o aumento dos valores de erro com o aumento do número de camadas. Em especial, para 10 neurônios por camada, os maiores erros são obtidos para maiores quantidades de camadas ocultas. Isso indica que embora essa redes com mais parâmetros tenha uma maior capacidade de ajuste, elas não conseguem obter bons resultados para um conjunto de dados não visto durante o treinamento, indicando a ocorrência de sobreajuste. Além disso, para o dados utilizados, é observado que quanto maior a capacidade de ajuste das redes mais acentuado é o sobreajuste.

Na Figura 5.43, são mostrados os valores de erro obtidos para diferentes número de neurônios para estes casos. Com base no comportamento do erro apresentado na Figura 5.43, foi selecionada como modelo para o sistema de *Wiener* em questão, a estrutura com 5 camadas ocultas e 7 neurônios por camada. Essa escolha se deu em razão de essa ser uma combinação capaz fornecer um dos menores erros sobre o conjunto de teste, assim como mostrado na Tabela 5.18, e também resultar em uma quantidade menor de parâmetros a serem ajustados, em relação a outras combinação com erros próximos, mas que empregam um número maior de camadas ou de neurônios. Dessa forma, a estrutura de rede escolhida para o caso este caso de estudo é como ilustrada na Figura 5.44, onde L_{in} , L_1 , L_2, \dots, L_4 , L_5 , L_{out} são as camadas da rede.

De acordo com a estrutura de modelo apresentada na equação 5.19, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

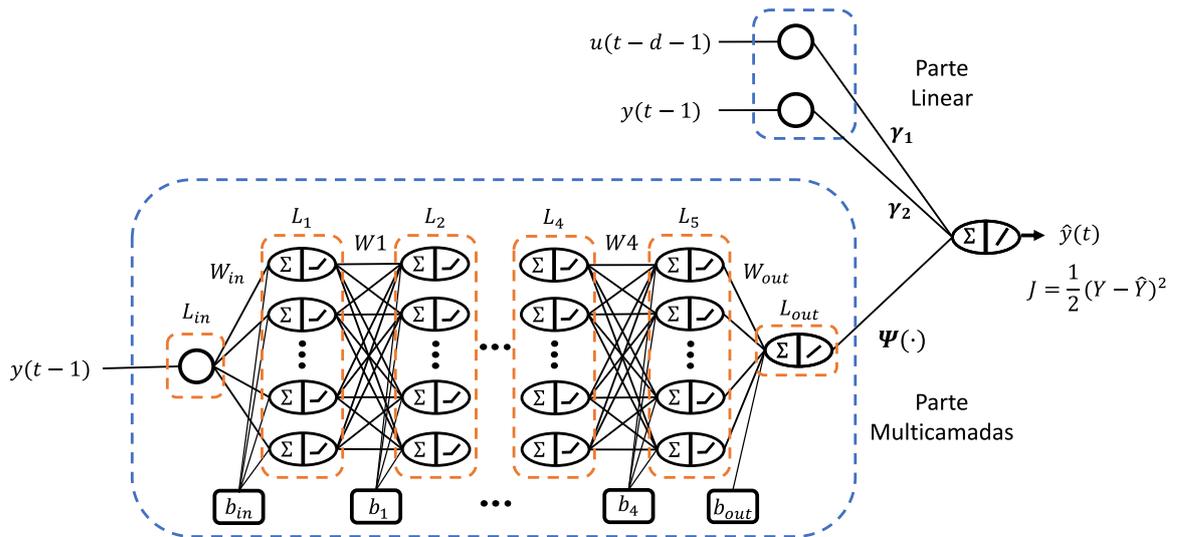
$$\hat{y}(t) = \gamma_1 u(t - d - 1) + \gamma_2 y(t - 1) + \Psi(y(t - 1)) \quad (5.19)$$

Figura 5.43: Gráfico de valores de erro para diferentes números de neurônios por camada para 5, 6 e 7 camadas (Caso 4).



Fonte: Autoria própria.

Figura 5.44: Ilustração da estrutura de rede adotada (Caso 4).



Fonte: Autoria própria.

tem-se como coeficientes da parte linear, os parâmetros γ_1 e γ_2 . Para os caso de sistemas de Wiener, não há uma correspondência direta entre esses coeficientes e os parâmetros da função $G_1(q)$. Ainda segundo o desenvolvimento da seção 4.3, para este caso de estudo, $\gamma_1 = \alpha_1 K$ e $\gamma_2 = \alpha_2 p_1$, onde K e p_1 são o ganho e o polo da função $G_1(q)$, respectivamente. Já α_1 e α_2 são coeficientes de ponderação supostos no desenvolvimento.

Na Tabela 5.29 são mostrados os valores estimados de γ_1 e γ_2 correspondentes aos quatro menores erros obtidos com as diversas estruturas de rede simuladas. Com base nos

Tabela 5.19: Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 4).

Estrutura de rede	5 camadas e 7 neurônios	6 camadas e 7 neurônios	7 camadas e 7 neurônios	8 camadas e 7 neurônios
MSE	0,4035	0,4044	0,4030	0,4023
γ_1	5,0312	5,0273	5,0303	5,0305
γ_2	1,0091	0,9888	0,9842	0,9851

Fonte: Autoria própria.

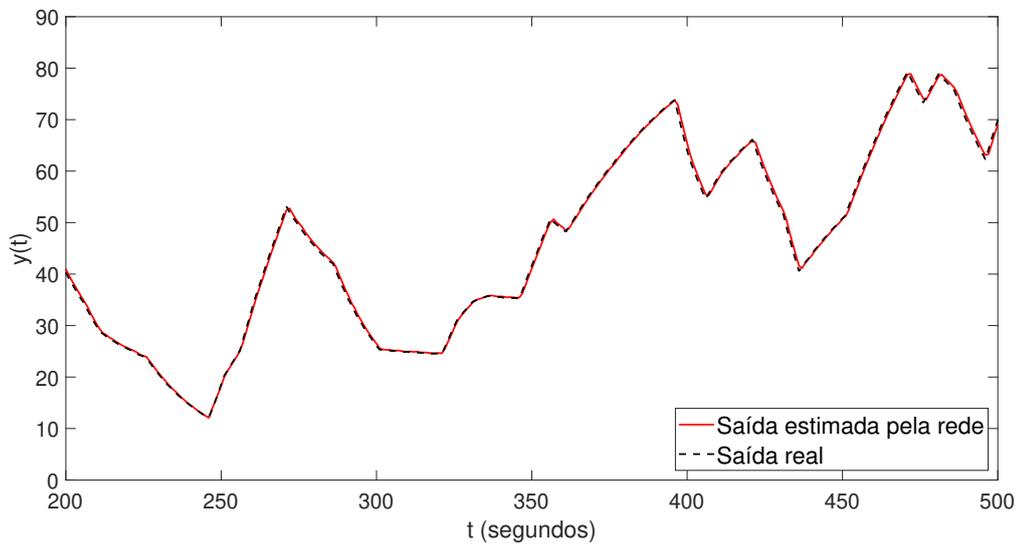
valores apresentados, é possível perceber que mesmo para estruturas de rede diferentes, são obtidos valores semelhantes para os parâmetros da parte linear. Isso indica a existência do parâmetros fixos de ponderação α_1 e α_2 considerados no desenvolvimento.

Para a estrutura de rede escolhida, foram obtidos $\gamma_1 = 5,0312$ e $\gamma_2 = 1,0091$. Dessa forma, tendo em vista que o mapeamento não linear $\Psi(\cdot)$ pode ser obtido a partir das matrizes de pesos apresentadas no Apêndice D, então

$$\hat{y}(t) = 5,0312u(t - d - 1) + 1,0091y(t - 1) + \Psi(y(t - 1)) \quad (5.20)$$

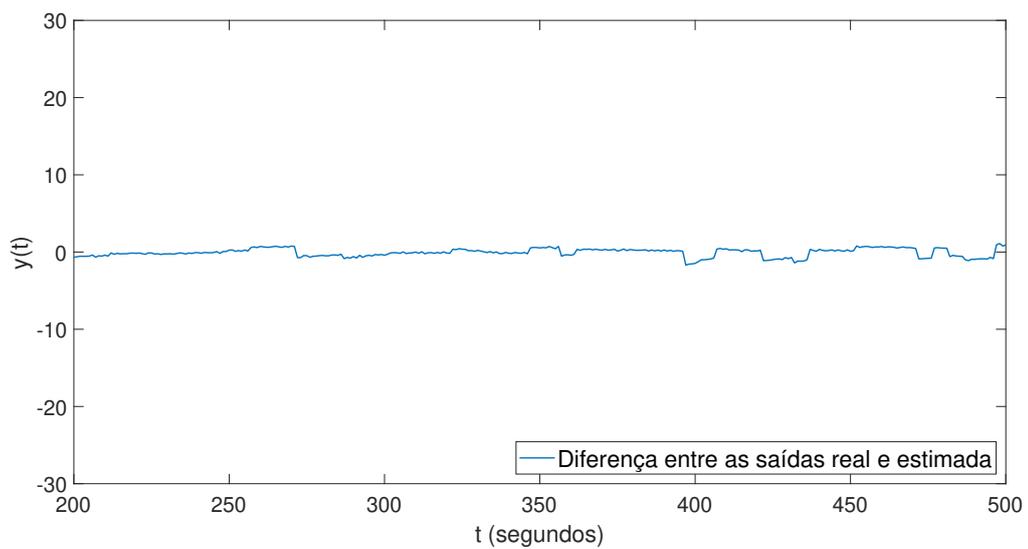
O erro médio quadrático obtido no teste foi de 0,40346. Uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido pode ser vista na Figura 5.45 para um subconjunto dos dados de teste. Na Figura 5.46 é apresentado o conjunto de valores de erro de estimação para o subconjunto adotado.

Figura 5.45: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 4).



Fonte: Autoria própria.

Figura 5.46: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 4).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Assim como nos casos anteriores foram utilizadas as mesmas quantidades, $M = 6$, de funções de *Laguerre* e $m = 8$ funções de base radiais. Foi também usada a mesma estratégia de ajuste do valor do polo p , resultando para este caso em $p = 0,8$.

Considerando a estrutura de modelo adotada por (STANISŁAWSKI; LATAWIEC, 2008) para sistemas de *Wiener*, apresentada na equação (5.21)

$$\hat{y}(t) = \sum_{i=1}^M c_i L_i(q, p) u(t) - \sum_{j=1}^m w_j \phi_j(y(t)) \quad (5.21)$$

foi utilizado um o método dos mínimos quadrados recursivos para o ajuste dos coeficientes c_i e w_j a partir dos dados de entrada e saída adotados para este caso de estudo. Os resultados obtidos são mostrados na Tabelas 5.20 e 5.21.

Tabela 5.20: Coeficientes OBF obtidos (Caso 4).

i	1	2	3	4	5	6
c_i	-12,60540	-15,44548	-7,98726	-8,47741	-2,43690	-2,66101

Fonte: Autoria própria.

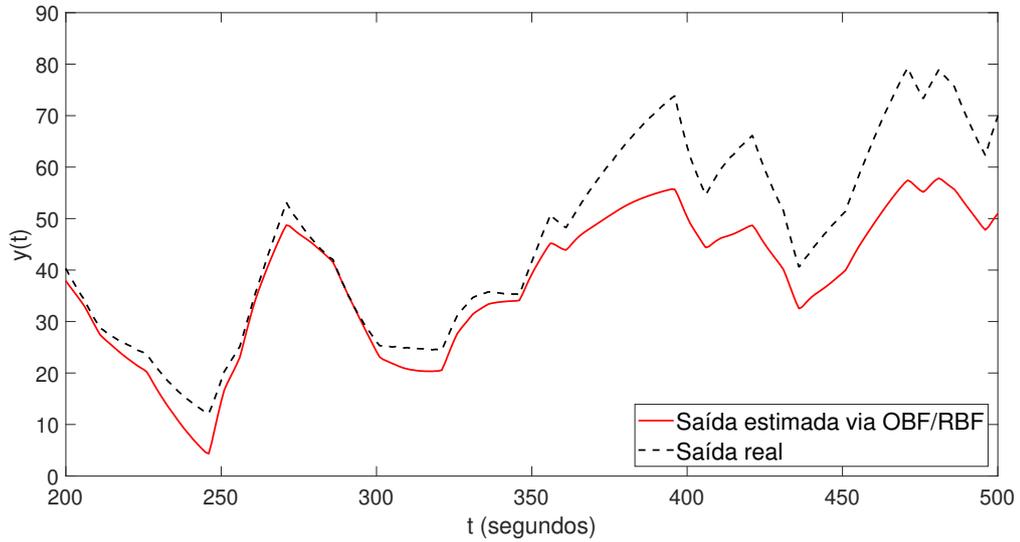
Tabela 5.21: Coeficientes RBF (Caso 4).

j	1	2	3	4	5	6	7	8
w_j	-16,80533	-4,43525	0,31245	0,37146	0,12138	0,08726	0,03517	0,01510

Fonte: Autoria própria.

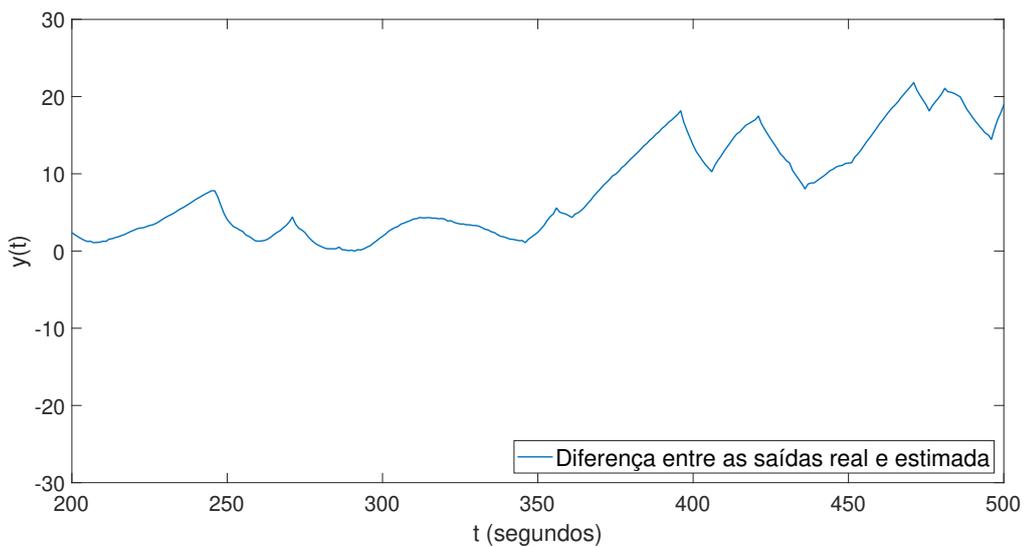
Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 89,63241$. Nas Figuras 5.47 e 5.48 são mostradas a comparação entre a saída real do sistema e a saída estimada a partir do modelo obtido OBF/RBF para um subconjunto dos dados de teste, além dos valores de erro de estimação para o intervalo adotado.

Figura 5.47: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 500$ (Caso 4).



Fonte: Autoria própria.

Figura 5.48: Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 4).



Fonte: Autoria própria.

5.4.5 Caso de estudo 5 (*Wiener*)

Sendo um sistema de *Wiener* formado por um bloco não linear $f_W(\cdot)$ e por uma função linear $G_2(q)$. Onde $f_W(\cdot)$ e $G_2(q)$ são funções adaptadas de (ZHU, 2001), e são dadas de acordo com as equações (5.22) e (5.23), respectivamente.

$$f_H(v) = v + 2v^2 \quad (5.22)$$

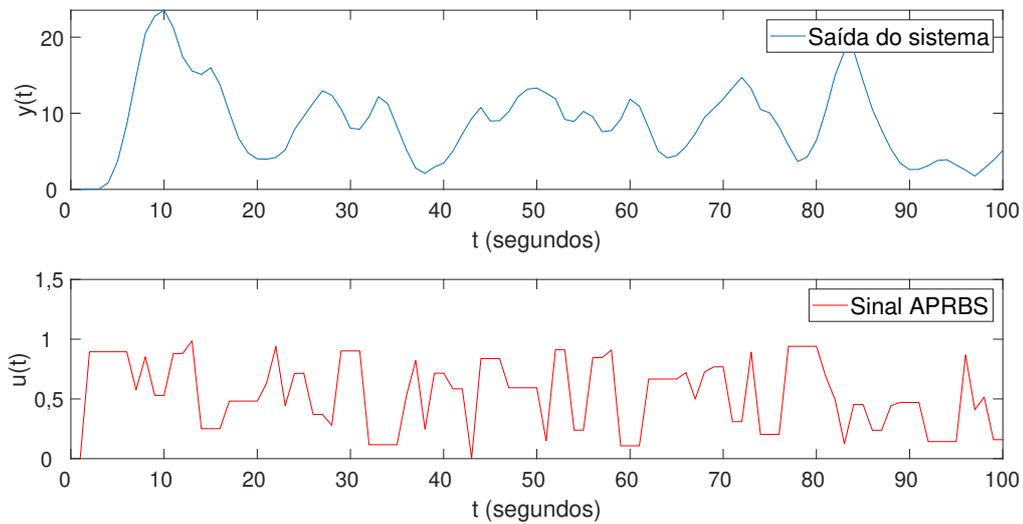
$$G_2(q) = (q^{-2}) \frac{0,5}{1 - 1,5q^{-1} + 0,65q^{-2}} \quad (5.23)$$

A função $G_2(q)$ é a mesma utilizada no caso de estudo 2. Quanto à parte não linear, $f_W(\cdot)$, trata-se da mesma função utilizada no caso de estudo 4. Com relação ao sinal de excitação, considerando função $G_2(q)$, foram utilizados os mesmos parâmetros do caso de estudo 2 para a geração do sinal PRBS, sendo eles

- $\lambda = 1 T_s$;
- $T_P = 15 T_s$;
- $M = 400$.

Para o sinal APRBS, foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 1$, respectivamente, com o intuito de evitar valores de saída muito elevados. O resultado foi um sinal APRBS com 6000 pontos de dados. Os primeiros 100 valores do sinal de excitação, bem como a saída correspondente do sistema, incluindo o ruído adicionado, são mostrados na Figura 5.49

Figura 5.49: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 5).



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Wiener* composto por uma função linear de primeira ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.22, onde $u(t - d - 2)$ é o sinal de excitação atrasado de $d + 2$

Tabela 5.22: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 5).

Entradas da parte linear	$u(t - d - 2)$	$y(t - 1)$	$y(t - 2)$
Entrada da parte multicamadas	$y(t - 1)$	$y(t - 2)$	–

Fonte: Autoria própria.

instantes de tempo, e $y(t - 1)$ e $y(t - 2)$ dizem respeito ao sinal de saída atrasado de 1 e 2 instantes de tempo, respectivamente. Neste caso d é um atraso conhecido e para a simulação foi adotado $d = 0$.

Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.49 para treinar a rede e posteriormente foi calculado o erro médio quadrático sobre um conjunto de teste para ser usado como métrica de desempenho. Esse procedimento foi realizado para diferentes escolhas de quantidades de

camadas ocultas e de números de neurônios por camada. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.23.

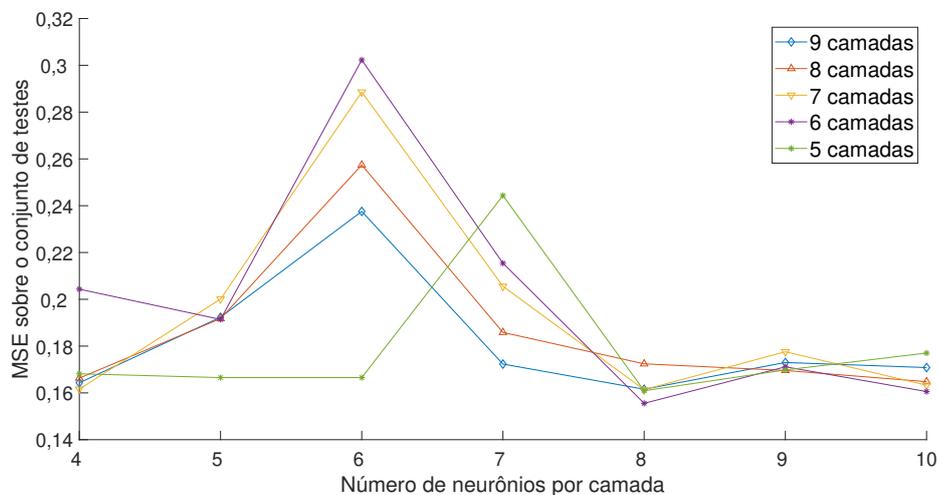
Tabela 5.23: Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 5).

Número de camadas ocultas	Número de neurônios por camada							
	4	5	6	7	8	9	10	
5	0,16816	0,16654	0,16654	0,24434	0,16092	0,16980	0,17702	
6	0,20433	0,19140	0,30228	0,21541	0,15554	0,17114	0,16057	
7	0,16162	0,20010	0,28856	0,20555	0,16138	0,17758	0,16329	
8	0,16630	0,19176	0,25740	0,18585	0,17241	0,16958	0,16479	
9	0,16431	0,19234	0,23756	0,17232	0,16157	0,17299	0,17082	

Fonte: Autoria própria.

Na Figura 5.50 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede. A partir

Figura 5.50: Gráfico de valores de erro para diferentes quantidades de camadas e neurônios (Caso 5).

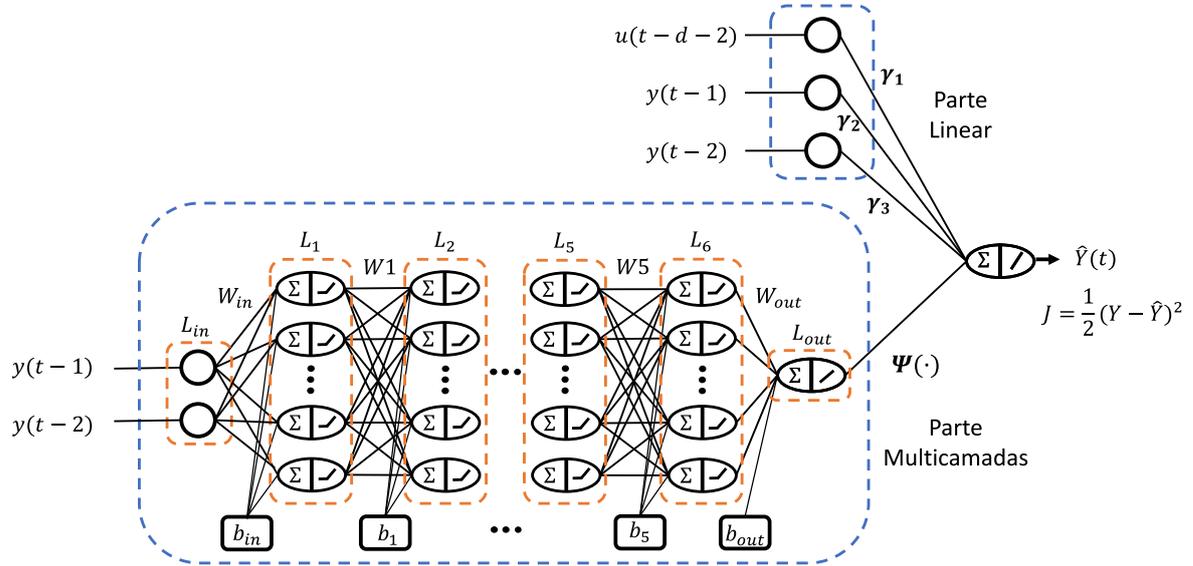


Fonte: Autoria própria.

da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que, de maneira geral, são obtidos resultados semelhantes para as várias quantidades de camadas ocultas. Para este caso, não foram observados indícios de *overfitting*, isto é, o aumento expressivo do erro de teste à medida que a complexidade das redes aumenta. Por outro lado, considerando que foram obtidos erros semelhantes para as diferentes configurações de rede, pode ser escolhida uma rede com uma menor quantidade de parâmetros, sem grandes prejuízos com relação a sua capacidade de representação do sistema. Assim, a estrutura com 6 camadas ocultas e 8 neurônios por

camada devido a essa combinação ter proporcionado o menor erro dentre as possibilidades analisadas. A estrutura de rede escolhida para o caso este caso de estudo é como ilustrada na Figura 5.51, onde L_{in} , L_1 , L_2, \dots, L_5 , L_6 , L_{out} são as camadas da rede.

Figura 5.51: Ilustração da estrutura de rede adotada (Caso 5).



Fonte: Autoria própria.

De acordo com a estrutura de modelo apresentada na equação 5.24, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

$$\hat{y}(t) = \gamma_1 u(t-d-2) + \gamma_2 y(t-1) + \gamma_3 y(t-2) + \Psi(y(t-1), y(t-2)) \quad (5.24)$$

tem-se como coeficientes da parte linear, os parâmetros γ_1 , γ_2 e γ_3 . Para os caso de sistemas de *Wiener*, não há uma correspondência direta entre esses coeficientes e os parâmetros da função $G_1(q)$. Ainda segundo o desenvolvimento da seção 4.3, para este caso de estudo, $\gamma_1 = \alpha_1 K$, $\gamma_2 = -\alpha_2 a_1$ e $\gamma_3 = -\alpha_3 a_0$, onde K é o ganho da função $G_2(q)$, a_1 e a_0 são os coeficientes de primeira e segunda ordem do denominador, respectivamente.

Na Tabela 5.29 são mostrados os valores estimados de γ_1 , γ_2 e γ_3 correspondentes aos quatro menores erros obtidos com as diversas estruturas de rede simuladas. Com base nos valores apresentados, é possível perceber que variações consideráveis nos valores de erro obtidos para diferentes estruturas provocam variações apreciáveis nos valores estimados dos coeficientes da parte linear. Isso confirma que a escolha de uma estrutura de rede deve se basear principalmente nos valores de erro obtidos e somente para casos que fornecem erros semelhantes é que se pode realizar a escolha com base na quantidade parâmetros de cada opção.

Tabela 5.24: Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 5).

Estrutura de rede	6 camadas e 8 neurônios	6 camadas e 10 neurônios	5 camadas e 8 neurônios	7 camadas e 8 neurônios
MSE	0,1555	0,1606	0,1609	0,1614
γ_1	3,6906	3,7053	3,6885	3,7096
γ_2	1,8467	1,8907	2,0257	1,7208
γ_3	0,4427	0,8152	0,5752	0,2858

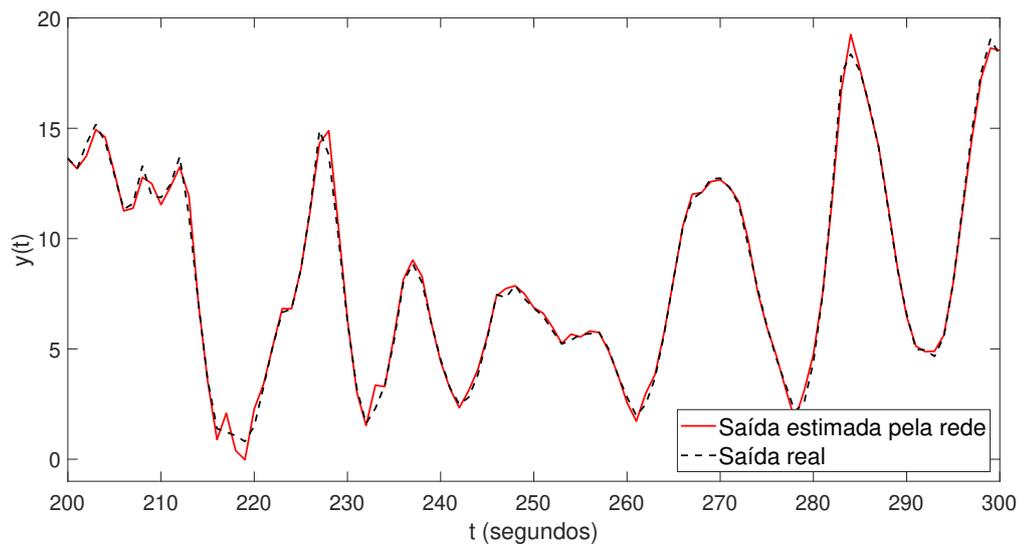
Fonte: Autoria própria.

Para a estrutura de rede escolhida de 6 camadas e 8 neurônios, foram obtidos $\gamma_1 = 3,6905$, $\gamma_2 = 1,8466$ e $\gamma_3 = 0.4427$. Já o mapeamento não linear $\Psi(\cdot)$ pode ser obtido a partir das matrizes de pesos apresentadas no Apêndice E, de modo que

$$\hat{y}(t) = 3,6905u(t - d - 2) + 1,8466y(t - 1) + 0.4427y(t - 2) + \Psi(y(t - 1), y(t - 2)) \quad (5.25)$$

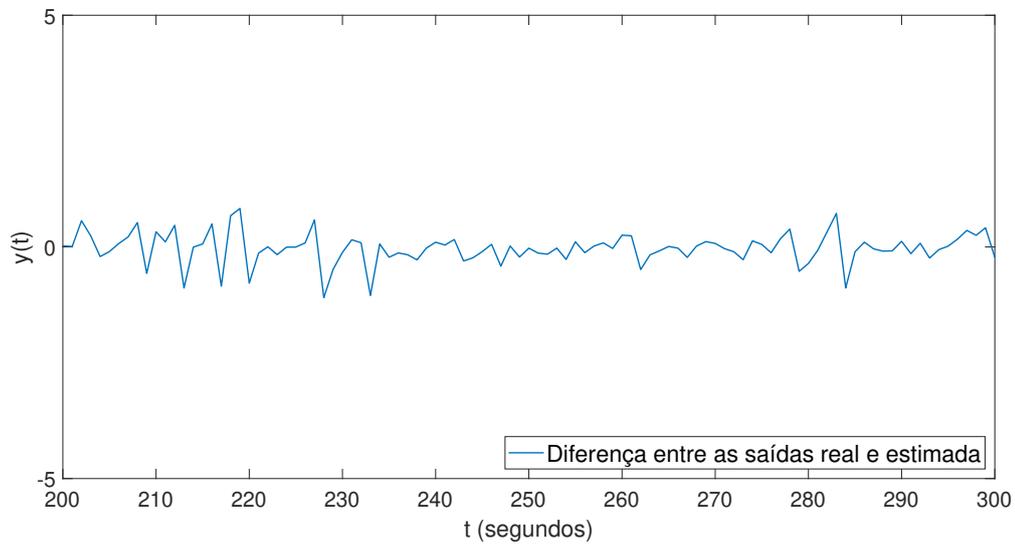
O erro médio quadrático obtido para este caso foi de 0,15554. Uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido pode ser vista na Figura 5.52 para um subconjunto dos dados de teste. Já na Figura 5.53 são apresentados os valores de erro de estimação para o subconjunto adotado.

Figura 5.52: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 5).



Fonte: Autoria própria.

Figura 5.53: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 5).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Assim como nos casos anteriores foram utilizadas $M = 6$, de funções de *Laquerre* e $m = 8$ funções de base radiais. Foi também usada a mesma estratégia de ajuste do valor do polo p , resultando para este caso em $p = 0,45$.

Foi utilizado um o método dos mínimos quadrados recursivos para o ajuste dos coeficientes c_i e w_j a partir dos dados de entrada e saída adotados para este caso de estudo. Os resultados obtidos são mostrados na Tabelas 5.25 e 5.26.

Tabela 5.25: Coeficientes OBF obtidos (Caso 5).

i	1	2	3	4	5	6
c_i	-1,44368	-6,60455	-8,44175	-1,22112	1,34787	2,42784

Fonte: Autoria própria.

Tabela 5.26: Coeficientes RBF (Caso 5).

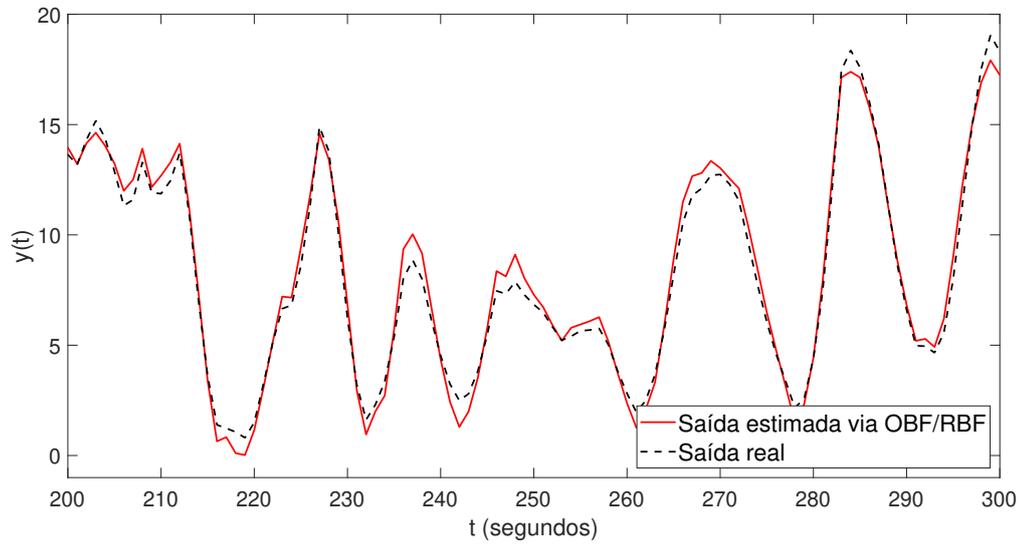
j	1	2	3	4	5	6	7	8
w_j	-3,06136	-3,02043	-0,38557	2,33222	2,39786	2,05507	0,98560	0,41158

Fonte: Autoria própria.

Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 0,57868$. Nas Figuras 5.54 e 5.55 são mostradas a comparação entre a saída

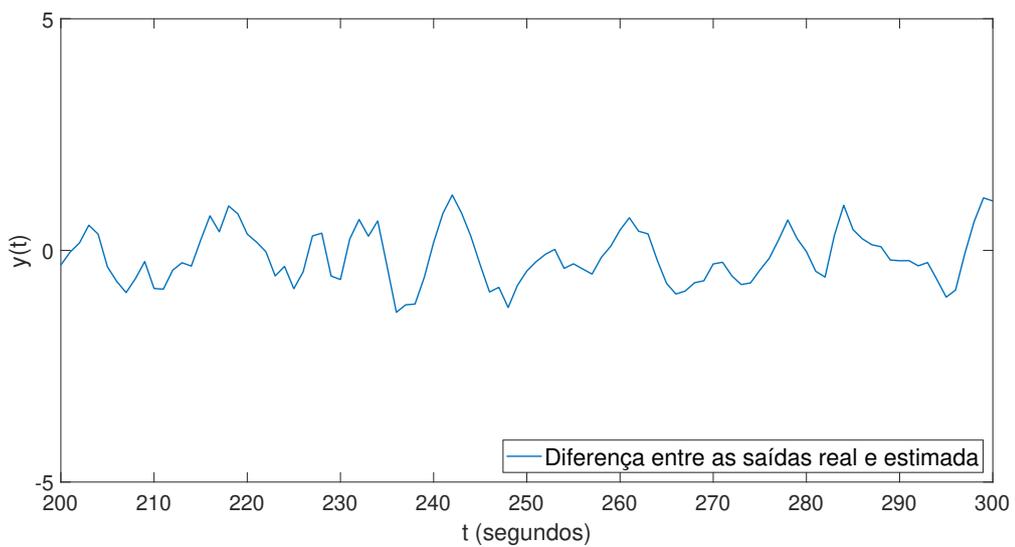
real do sistema e a saída estimada a partir do modelo obtido OBF/RBF, além do erro de estimação ponto a ponto para um subconjunto dos dados de teste.

Figura 5.54: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ até $t = 300$ (Caso 5).



Fonte: Autoria própria.

Figura 5.55: Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 5).



Fonte: Autoria própria.

5.4.6 Caso de estudo 6 (*Wiener*)

Sendo um sistema de *Wiener* formado por um bloco não linear $f_W(\cdot)$ e por uma função linear $G_3(q)$. Onde $f_W(\cdot)$ e $G_3(q)$ são funções utilizadas em (ZHU, 2001), e são dadas de acordo com as equações (5.26) e (5.27), respectivamente.

$$f_H(v) = v + 2v^2 \quad (5.26)$$

$$G_3(q) = (q^{-1}) \frac{1 + 0,5q^{-1}}{1 - 1,5q^{-1} + 0,7q^{-2}} \quad (5.27)$$

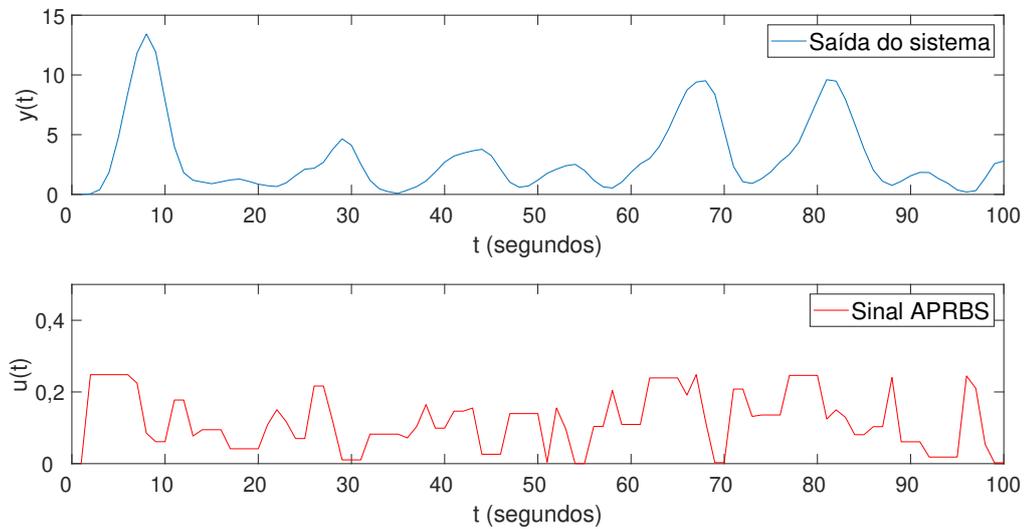
A função $G_3(q)$ é a mesma utilizada no caso de estudo 3. Quanto à parte não linear, $f_W(\cdot)$, trata-se da mesma função utilizada nos casos de estudo 4 e 5. Com relação ao sinal de excitação, considerando função $G_3(q)$, foram utilizados os mesmos parâmetros do caso de estudo 3 para a geração do sinal PRBS, sendo eles

- $\lambda = 1 T_s$;
- $T_P = 15 T_s$;
- $M = 400$.

Para o sinal APRBS, foram adotados valores de amplitude mínima e máxima $a_{min} = 0$ e $a_{max} = 0,25$, respectivamente, com o intuito de evitar valores de saída muito elevados.

O resultado foi um sinal APRBS com 6000 pontos de dados. Os primeiros 100 valores do sinal de excitação, bem como a saída correspondente do sistema, incluindo o ruído adicionado, são mostrados na Figura 5.56

Figura 5.56: Sinal de excitação (abaixo) e saída do sistema (acima) para os primeiros 100 pontos de dados (Caso 6).



Fonte: Autoria própria.

Aplicação do método proposto

Quanto à estrutura de rede utilizada, tendo em vista que neste caso de estudo trata-se de um sistema de *Wiener* composto por uma função linear de primeira ordem, então, de acordo com o desenvolvimento realizado na seção 4.3, o conjunto de entradas utilizado é como mostrado na Tabela 5.27, onde $u(t - d - 1)$ e $u(t - d - 2)$ são referentes ao sinal de

Tabela 5.27: Tipos de dados de entrada a serem utilizados para o método proposto (Caso 6)

Entradas da parte linear	$u(t - d - 1)$	$u(t - d - 2)$	$y(t - 1)$	$y(t - 2)$
Entrada da parte multicamadas	$y(t - 1)$	$y(t - 2)$	–	–

Fonte: Autoria própria.

excitação atrasado de $d + 1$ e $d + 2$ instantes de tempo. Já $y(t - 1)$ e $y(t - 2)$ dizem respeito ao sinal de saída atrasado de 1 e 2 instantes de tempo, respectivamente. Neste caso d é um atraso conhecido e para a simulação foi adotado $d = 0$. Para esta configuração de entrada, foram utilizados os dados de entrada e saída mostrados parcialmente na Figura 5.56 para treinar da rede. Foi usado como métrica de desempenho o erro médio quadrático sobre o conjunto de teste. Os valores de MSE obtidos para as diferentes combinações são apresentados na Tabela 5.28.

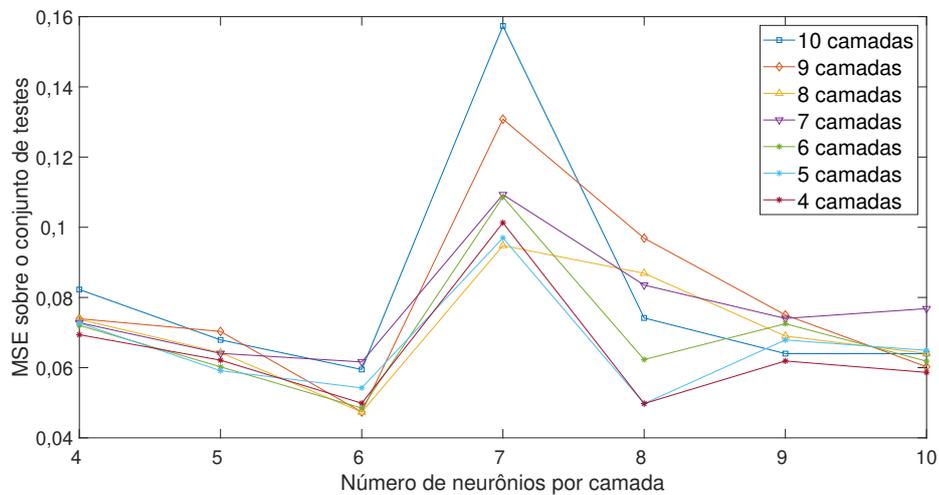
Tabela 5.28: Valores de erro obtidos sobre o conjuntos de teste para diferentes quantidades de camadas e neurônios (Caso 6).

		Número de neurônios por camada						
		4	5	6	7	8	9	10
Número de camadas ocultas	4	0,06939	0,06212	0,04987	0,10129	0,04976	0,06190	0,05865
	5	0,07271	0,05913	0,05424	0,09697	0,04976	0,06789	0,06493
	6	0,07205	0,06023	0,04847	0,10860	0,06231	0,07252	0,06184
	7	0,07282	0,06406	0,06163	0,10931	0,08353	0,07402	0,07682
	8	0,07377	0,06429	0,04734	0,09477	0,08689	0,06900	0,06406
	9	0,07394	0,07032	0,04736	0,13080	0,09688	0,07501	0,06025
	10	0,08226	0,06791	0,05947	0,15735	0,07416	0,06400	0,06400

Fonte: Autoria própria.

Na Figura 5.57 são mostrados de maneira gráfica os valores de erro médio quadrático sobre o conjunto de teste para diferentes configurações estruturais da rede. A partir

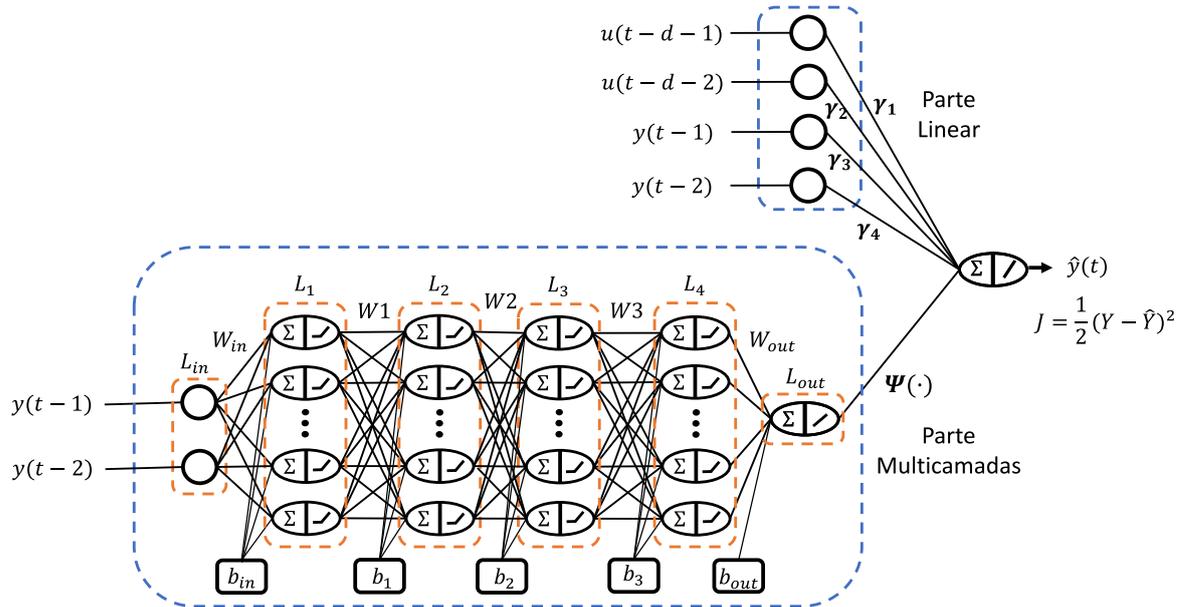
Figura 5.57: Valores de erro para diferentes quantidades de camadas e de neurônios (Caso 6).



Fonte: Autoria própria.

da análise da evolução do erro em relação às diferentes quantidades de camadas ocultas testadas, é possível perceber que, de maneira geral, são obtidos resultados semelhantes para as várias quantidades de camadas ocultas. Foi selecionada, então, a estrutura com 4 camadas ocultas e 6 neurônios por camada devido a ela ter proporcionado um dos melhores resultados e com a vantagem de ser uma estrutura com uma quantidade relativamente baixa de parâmetros. Dessa forma, a estrutura de rede escolhida para o caso este caso de estudo é como ilustrada na Figura 5.58, onde L_{in} , L_1 , L_2, \dots, L_3 , L_4 , L_{out} são as camadas da rede.

Figura 5.58: Ilustração da estrutura de rede adotada (Caso 6).



Fonte: Autoria própria.

De acordo com a estrutura de modelo apresentada na equação 5.28, que corresponde a este caso de estudo segundo o desenvolvimento realizado na seção 4.3,

$$\begin{aligned} \hat{y}(t) = & \gamma_1[u(t-d-1)] + \gamma_2[u(t-d-2)] + \gamma_3[y(t-1)] + \\ & + \gamma_4[y(t-2)] + \Psi(y(t-1), y(t-2)) \end{aligned} \quad (5.28)$$

tem-se como coeficientes da parte linear, os parâmetros γ_1 , γ_2 , γ_3 e γ_4 . Para os caso de sistemas de *Wiener*, não há uma correspondência direta entre esses coeficientes e os parâmetros da função $G_3(q)$. Ainda segundo o desenvolvimento da seção 4.3, para este caso de estudo, $\gamma_1 = \alpha_1 K$, $\gamma_2 = -\alpha_2 K b_1$, $\gamma_3 = -\alpha_3 a_1$ e $\gamma_4 = -\alpha_4 a_0$, onde K é o ganho da função $G_2(q)$, b_1 é o coeficiente de primeira ordem do numerador e a_1 e a_0 são os coeficientes de primeira e segunda ordem do denominador, respectivamente.

Na Tabela 5.29 são mostrados os valores estimados de γ_1 , γ_2 , γ_3 e γ_4 correspondentes aos quatro menores erros obtidos com as diversas estruturas de rede simuladas. Com base nos valores apresentados, é possível perceber pequenas variações nos valores de erro obtidos para as estruturas em questão e, conseqüentemente, variações moderadas nos valores estimados dos coeficientes da parte linear.

Para a estrutura de rede escolhida de 6 camadas e 8 neurônios, foram obtidos foram obtidos $\gamma_1 = 3,0167$, $\gamma_2 = 2,9318$, $\gamma_3 = 2,0138$ e $\gamma_4 = 2,7747$. Já o mapeamento não linear $\Psi(\cdot)$ pode ser obtido a partir das matrizes de peso apresentadas no Apêndice F, de

Tabela 5.29: Valores estimados para os coeficientes da parte linear de acordo com o método proposto (Caso 6).

Estrutura de rede	4 camadas e 6 neurônios	8 camadas e 6 neurônios	9 camadas e 6 neurônios	6 camadas e 6 neurônios	4 camadas e 8 neurônios
MSE	0,0499	0,0473	0,0474	0,0485	0,0498
γ_1	3,0168	3,0155	3,0171	3,0183	3,0170
γ_2	2,9319	2,9331	2,9368	2,9355	2,9390
γ_3	2,0139	1,9366	1,8866	1,8271	1,9879
γ_4	2,7748	2,6767	2,6338	2,5629	2,6953

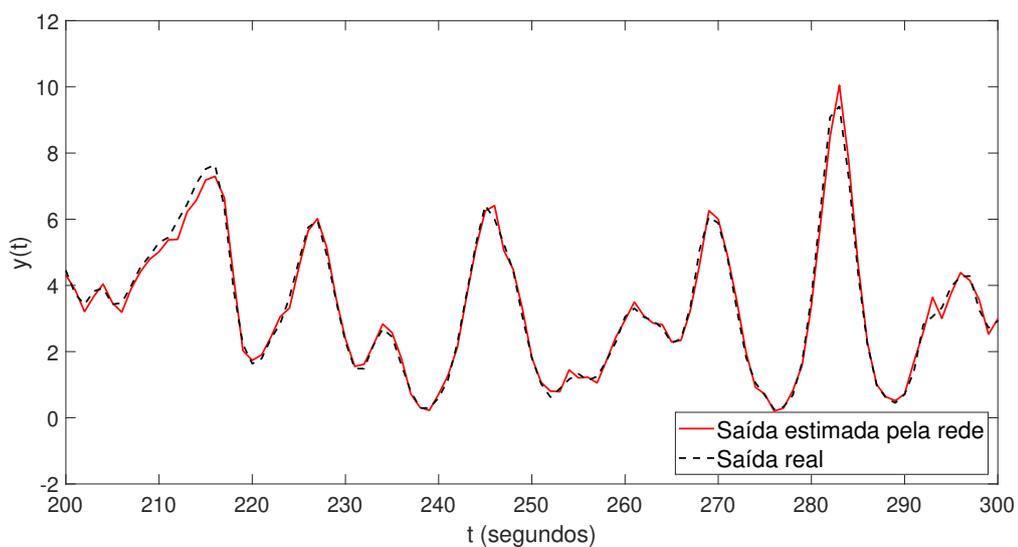
Fonte: Autoria própria.

modo que

$$\hat{y}(t) = 3,0167u(t - d - 1) + 2,9318u(t - d - 2) + 2,0138y(t - 1) + 2,7747y(t - 2) + \Psi(y(t - 1), y(t - 2)) \quad (5.29)$$

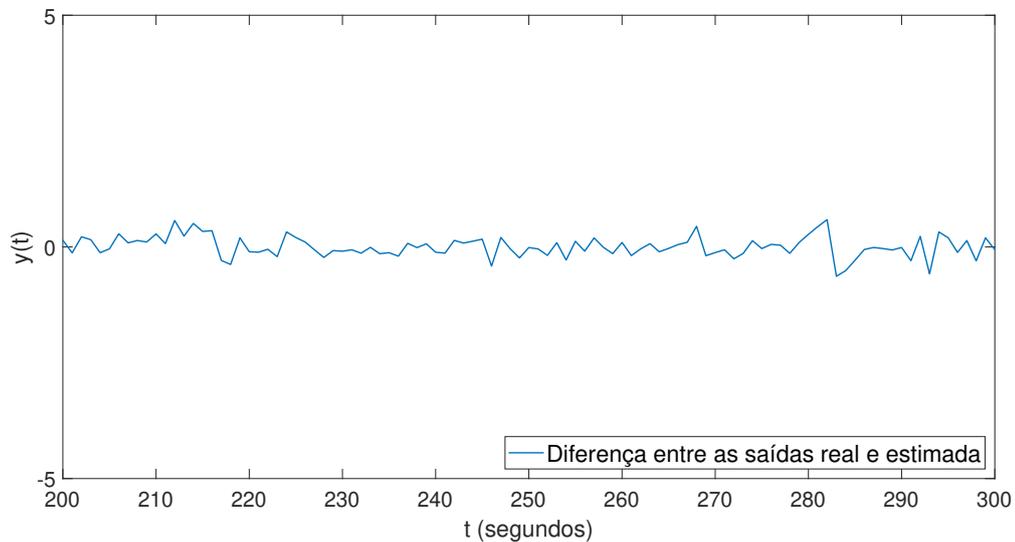
O erro médio quadrático obtido no teste foi de 0,04987. Uma comparação gráfica entre a saída real do sistema e a saída estimada a partir do modelo obtido por meio do método proposto para um subconjunto específico dos dados de teste pode ser vista na Figura 5.59. Já na Figura 5.60 é apresentado um gráfico do erro de estimação para o subconjunto de teste adotado.

Figura 5.59: Saída real e saída estimada por meio da rede para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 6).



Fonte: Autoria própria.

Figura 5.60: Erro de estimação utilizado o método proposto sobre o conjunto de teste no intervalo adotado (Caso 6).



Fonte: Autoria própria.

Aplicação do método OBF/RBF

Assim como nos casos anteriores foram utilizadas $M = 6$, de funções de *Laquerre* e $m = 8$ funções de base radiais. Foi também usada a mesma estratégia de ajuste do valor do polo p , resultando para este caso em $p = 0,35$.

Foi utilizado um o método dos mínimos quadrados recursivos para o ajuste dos coeficientes c_i e w_j a partir dos dados de entrada e saída adotados para este caso de estudo. Os resultados obtidos são mostrados na Tabelas 5.30 e 5.31.

Tabela 5.30: Coeficientes OBF obtidos (Caso 6)

i	1	2	3	4	5	6
c_i	-3,02331	-7,26750	-7,01988	-2,60926	1,10517	2,31180

Fonte: Autoria própria.

Tabela 5.31: Coeficientes RBF (Caso 6)

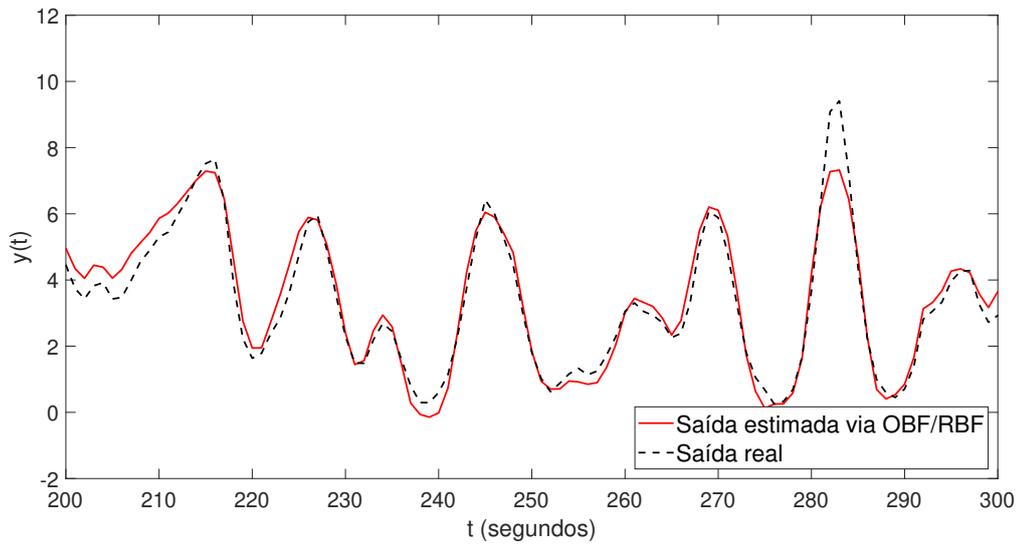
j	1	2	3	4	5	6	7	8
w_j	5,47805	-0,75523	-5,28848	-3,91451	0,82139	1,73835	1,61857	-0,17222

Fonte: Autoria própria.

Sobre o conjunto de teste adotado neste caso, foi obtido um erro médio quadrático, $MSE = 0,23936$. Na Figura 5.61 são apresentados os gráficos da saída real do sistema e

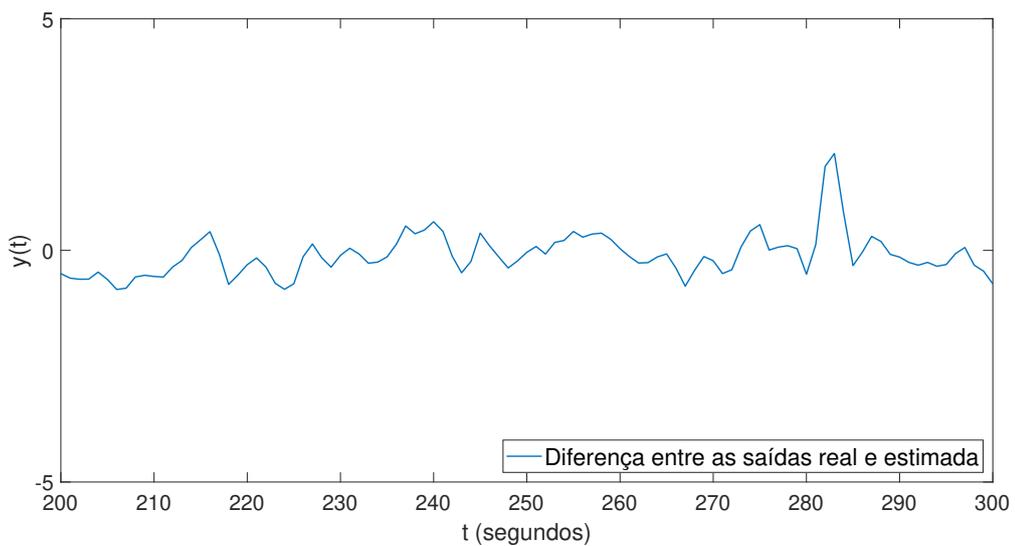
da saída estimada a partir do método OBF/RBF para um determinado subconjunto dos dados de teste. Já na Figura 5.62, é apresentado o gráfico do erro de estimação para cada ponto dentro deste subconjunto.

Figura 5.61: Saída real e saída estimada utilizando OBF/RBF para o conjunto de teste no intervalo de $t = 200$ a $t = 300$ (Caso 6).



Fonte: Autoria própria.

Figura 5.62: Erro de estimação utilizado o método OBF/RBF sobre o conjunto de teste no intervalo adotado (Caso 6).



Fonte: Autoria própria.

5.5 Análise dos resultados

Visando realizar uma comparação objetiva dos métodos de identificação testados, na Tabela 5.32 são reunidos os valores de erro médio quadrático obtidos sobre os conjuntos de teste para cada um dos casos de estudo adotados. Com base nessas informações, diversas comparações podem ser realizadas.

Tabela 5.32: Valores de erro médio quadrático (MSE) obtidos por meio do método proposto e do método de referência para os casos de estudo abordados

	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6
Método proposto	0,00466	0,00824	0,00103	0,40346	0,15554	0,04987
Método OBF/RBF	0,37020	0,04111	0,04041	89,63241	0,57868	0,23936

Fonte: Autoria própria.

Inicialmente, é importante ressaltar que os casos de estudo 1, 2 e 3 são referentes a sistemas de *Hammerstein*. Já os casos 4, 5 e 6 se referem a sistemas de *Wiener*. A partir da comparação entre os resultados dos sistemas com partes lineares de mesma ordem, é possível perceber que para os sistemas de *Wiener* são obtidos erros maiores tanto para o método proposto quanto para o método baseado em OBF/RBF.

Nesse sentido é importante ressaltar que para ambos os métodos, no caso de sistemas de *Wiener*, são utilizados como entradas das partes não lineares os valores de saída dos sistemas a serem identificados. Assim, o ajuste de parâmetros das partes não lineares sofre uma influência direta do ruído. Dessa forma, é de se esperar que, para sistemas de *Wiener*, a obtenção de modelos por meio dos métodos em questão seja prejudicada pela presença do ruído. Portanto, embora nos casos de estudo 1 e 4, 2 e 5, 3 e 6 sejam considerados sistemas de complexidades semelhantes, os valores de erro superiores obtidos nos casos 4, 5, 6 se devem, provavelmente, ao fato de estes se tratarem de sistemas de *Wiener*.

Com relação ao método proposto, particularmente no que se refere às estruturas escolhidas para as redes, em cada caso foi selecionada uma combinação diferente de quantidade de camadas ocultas e de neurônios, assim como mostrado na Tabela 5.33. Com base nisso, é possível perceber que os melhores resultados obtidos, os quais dizem respeito aos sistemas de *Hammerstein*, estão associados a maiores quantidades de camadas ocultas. O que é condizente com o resultado apresentado por (LJUNG et al., 2020) no sentido de que o número de camadas é um fator determinante (mais do que o número de neurônios) para a capacidade de aprendizado de uma rede. Desse modo uma maior quantidade de camadas tende a proporcionar uma maior capacidade de ajuste.

Tabela 5.33: Quantidades de camadas e neurônios para cada caso de estudo.

	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6
Número de camadas ocultas	10	8	6	5	6	4
Número de neurônios por camada	4	7	10	7	8	6

Fonte: Autoria própria.

No entanto, para o caso dos sistemas de *Wiener* foi verificado que o aumento no número de camadas, de maneira geral, não levou a obtenção de ajustes mais precisos. Chegando a haver, em alguns casos, uma piora nos resultados obtidos. Esse fato se deve, provavelmente, à influência direta do ruído nas entradas aplicadas às redes correspondentes aos sistemas de *Wiener*. Desse modo, mesmo estruturas de rede com maior capacidade de ajuste aos dados de entrada podem não conseguir produzir modelos precisos a partir de entradas ruidosas.

Nesse sentido, um ponto importante a se destacar é a possibilidade de que redes com um maiores capacidades de aprendizagem consigam se ajustar muito bem, mesmo aos dados com ruído. Porém este comportamento leva às redes a não alcançarem bom desempenho sobre conjuntos de teste não vistos no treinamento, o que caracteriza a ocorrência de um sobreajuste aos dados de treinamento, o chamado *overfitting*.

Uma observação importante diz respeito à discrepância entre os valores de MSE obtidos por meio do método OBF/RBF para os sistemas com parte linear de primeira ordem em relação aos demais casos. Isto é, observando as diferenças de resultados para o caso de estudo 1 em relação aos casos 2 e 3, para sistemas de *Hammerstein*, bem como para o caso 4 em relação aos casos 5 e 6. Com base nesta comparação é possível perceber que os valores de erro obtidos para os casos de estudo 1 e 4 são substancialmente maiores do que aqueles obtidos para os demais sistemas. O que se deve possivelmente aos dados utilizados nesses casos.

É importante ressaltar que para os sistemas 1 e 4 foram projetados sinais de excitação com uma menor quantidade de níveis de amplitude em relação aos demais casos. Dessa forma, os dados de entrada e saída obtidos trazem consigo menores quantidades de informação a respeito dos comportamentos dinâmicos de interesse, em relação aos sistemas de ordem superior. Para os quais foram projetados sinais de excitação que abrangem um número maior de níveis de amplitude.

Dessa forma, considera-se esta escassez relativa de informações a respeito do comportamento não linear dos sistemas como uma causa provável para os resultados mais imprecisos. Principalmente para o caso de estudo 4, que se trata de um sistema de *Wi-*

ener. É importante ainda observar que para os mesmos conjuntos de dados, o método proposto foi capaz de fornecer resultados de identificação compatíveis com aqueles obtidos para os demais casos.

Comparando o método proposto e o método baseado em OBF/RBF a partir dos resultados obtidos, é possível perceber que, para todos os casos de estudo, o método proposto foi capaz de fornecer modelos mais precisos para os sistemas de interesse. Esses resultados indicam que a abordagem proposta apresenta, de maneira geral, uma maior capacidade de modelagem de dinâmicas não lineares quando comparado com o método de referência para um mesmo conjunto de dados disponíveis.

Com relação à execução dos métodos, de maneira geral, o método baseado em OBF/RBF apresenta a vantagem de ter uma execução mais simples do que os processos de treinamento das redes. Entretanto, apresenta a desvantagem de necessitar do ajuste do parâmetro das funções OBF. Já no que se refere à flexibilidade estrutural e possíveis aperfeiçoamentos, o método proposto apresenta vantagens relacionadas à possibilidade de se utilizar diferentes estruturas de rede, além de ser possível empregar diversas outras técnicas de aprendizagem profunda.

5.6 Conclusão

Considerando os casos de estudo adotados, os resultados de identificação obtidos indicam que o método proposto baseado em redes neurais profundas é capaz de realizar ajustes mais precisos e consistentes do que o método de referência, para os diferentes sistemas simulados. Destaca-se, portanto, a verificação da capacidade de resolução de problemas não-lineares por parte das técnicas de aprendizagem baseada em dados. Além do mais, tendo em vista os grandes avanços de pesquisa na área de *deep learning*, tem-se um cenário bastante promissor com relação à aplicação de técnicas de aprendizagem para a identificação de sistemas, sobretudo para sistemas não lineares.

Capítulo 6

Conclusões e Sugestões para Trabalhos Futuros

6.1 Conclusões

Existem muitas semelhanças entre os campos da aprendizagem profunda e da identificação de sistemas, principalmente no que se refere ao fluxos de trabalho, aos problemas abordados e às estruturas algorítmicas de maneira geral. Nesse sentido, considerando os avanços expressivos alcançados em diversas áreas de aplicação a partir de técnicas de *deep learning*, é esperado que a integração de técnicas desses dois campos seja capaz de gerar resultados bastante expressivos.

Com base nessa premissa, a formulação proposta consiste justamente na integração de uma técnica de aprendizagem profunda com uma estratégia básica de modelagem amplamente empregada para tarefas de identificação de sistemas. E, para fins de comparação foi tomado como referência o método de identificação no qual a abordagem proposta teve sua formulação inicial baseada. Este método de referência, por sua vez, não faz uso de técnicas baseadas em aprendizado. Dessa forma sendo possível avaliar o efeito provocado pela incorporação de técnicas de aprendizagem profunda em um contexto de identificação de sistemas.

A partir dos testes comparativos realizados, foram observadas melhorias tanto no que se refere aos valores de erro obtidos, quanto à flexibilidade e robustez em relação aos diferentes casos de estudo adotados. Os resultados alcançados confirmam a capacidade que os métodos baseados em aprendizagem têm de promover melhoras nos resultados de identificação em relação àqueles obtidos por meio de técnicas convencionais.

Considera-se, por fim, que mesmo técnicas elementares de aprendizagem profunda são capazes de oferecer suporte para a obtenção de bons resultados em aplicações de identificação de sistemas não lineares. Todavia, há que se considerar a ampla gama de

técnicas e algoritmos de treinamento mais sofisticados e bem estabelecidos nesta área. Esse panorama, por sua vez, abre caminho para o emprego de abordagens dessa natureza na identificação de sistemas de elevada complexidade nas mais diversas áreas de aplicação.

6.2 Sugestões para trabalhos futuros

Diante do exposto, algumas pesquisas relevantes podem ser desenvolvidas a partir do presente trabalho. Tendo em vista os grandes avanços alcançados no campo da aprendizagem profunda, a abordagem de identificação proposta pode ser aperfeiçoada e estendida para casos mais complexos mediante o uso de outras técnicas mais sofisticadas de *deep learning*. No que se refere ao emprego de técnicas de aprendizagem baseada em dados em problemas de identificação de sistemas, uma pesquisa relevante que pode ser realizada diz respeito à incorporação de técnicas de aprendizagem por reforço (*reinforcement learning*) para o ajuste de modelos. Outro trabalho importante consiste no uso dessas técnicas para a obtenção de modelos capazes de oferecer suporte para aplicações de controle preditivo baseado em modelo (*Model Predictive Control* - MPC).

Referências Bibliográficas

- ANDERSSON, C. et al. Deep convolutional networks in system identification. In: IEEE. *2019 IEEE 58th Conference on Decision and Control (CDC)*. [S.l.], 2019. p. 3670–3676.
- BILLINGS, S. A. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. [S.l.]: John Wiley & Sons, 2013.
- BRUNTON, S. L.; KUTZ, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. [S.l.]: Cambridge University Press, 2019.
- GEDON, D. et al. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, Elsevier, v. 54, n. 7, p. 481–486, 2021.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.
- HOF, P. M. Van den. System identification-data-driven modelling of dynamic systems. *Lecture notes, Eindhoven University of Technology*, 2012.
- ISERMANN, R.; MÜNCHHOF, M. *Identification of dynamic systems: an introduction with applications*. [S.l.]: Springer Science & Business Media, 2010.
- LATAWIEC, K. J. et al. Linear parameter estimation and predictive constrained control of wiener/hammerstein systems. *IFAC Proceedings Volumes*, Elsevier, v. 36, n. 16, p. 345–350, 2003.
- LJUNG, L. System identification. In: *Signal analysis and prediction*. [S.l.]: Springer, 1998. p. 163–173.
- LJUNG, L. et al. Deep learning and system identification. *IFAC-PapersOnLine*, Elsevier, v. 53, n. 2, p. 1175–1181, 2020.
- NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. [S.l.]: Springer Nature, 2020.
- SCHÜSSLER, M.; MÜNKER, T.; NELLES, O. Deep recurrent neural networks for nonlinear system identification. In: IEEE. *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.], 2019. p. 448–454.
- STANISLAWSKI, R.; HUNEK, W. P.; LATAWIEC, K. J. Modeling of nonlinear block-oriented systems using orthonormal basis and radial basis functions. In: IEEE. *2008 19th International Conference on Systems Engineering*. [S.l.], 2008. p. 55–58.

STANISŁAWSKI, R.; LATAWIEC, K. J. Orthonormal basis and radial basis functions in modeling and identification of nonlinear block-oriented systems. *Automation and Robotics*, BoD–Books on Demand, p. 277, 2008.

STRANG, G. *Linear algebra and learning from data*. [S.l.]: Wellesley-Cambridge Press Cambridge, 2019.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. *Machine learning refined: Foundations, algorithms, and applications*. [S.l.]: Cambridge University Press, 2020.

WU, R.-T.; JAHANSHAHI, M. R. Deep convolutional neural network for structural dynamic response estimation and system identification. *Journal of Engineering Mechanics*, American Society of Civil Engineers, v. 145, n. 1, p. 04018125, 2019.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

ZHU, Y. *Multivariable system identification for process control*. [S.l.]: Elsevier, 2001.

Apêndice A

Caso de estudo 1

A seguir é apresentado o procedimento para a obtenção do mapeamento $\Psi(\cdot)$ para o caso de estudo 1 a partir da rede selecionada de 10 camadas ocultas e 4 neurônios por camada oculta.

Sejam

- X_{in} - o vetor de entrada da rede;
- b - o vetor de polarização;
- W_{in} - a matriz de pesos de entrada;
- W_1, \dots, W_9 - as matrizes de pesos das camadas ocultas;
- W_{out} - a matriz de pesos de saída.

Para a obtenção do mapeamento $\Psi(\cdot)$, seja S_1 o vetor de resultados das somas ponderadas realizadas pelos neurônios da camada 1. Além disso, seja $X = ReLU(S_1)$. Dessa forma

$$\begin{aligned} S_1 &= (W'_{in} X_{in}) + b \\ X_1 &= ReLU(S_1) \end{aligned}$$

$$\begin{aligned} S_2 &= (W'_1 X_1) + b \\ X_2 &= ReLU(S_2) \end{aligned}$$

$$\begin{aligned} S_3 &= (W'_2 X_2) + b \\ X_3 &= ReLU(S_3) \end{aligned}$$

$$S_4 = (W'_3 X_3) + b$$

$$X_4 = ReLU(S_4)$$

$$S_5 = (W'_4 X_4) + b$$

$$X_5 = ReLU(S_5)$$

$$S_6 = (W'_5 X_5) + b$$

$$X_6 = ReLU(S_6)$$

$$S_7 = (W'_6 X_6) + b$$

$$X_7 = ReLU(S_7)$$

$$S_8 = (W'_7 X_7) + b$$

$$X_8 = ReLU(S_8)$$

$$S_9 = (W'_8 X_8) + b$$

$$X_9 = ReLU(S_9)$$

$$S_{10} = (W'_9 X_9) + b$$

$$X_{10} = ReLU(S_{10})$$

$$S_{out} = (W'_{out} X_{10}) + b$$

$$\Psi(\cdot) = S_{out}$$

Nas tabelas A.1 até A.13 são mostradas as matrizes de pesos referentes ao caso de estudo 1.

Observação: em todas as tabelas foi utilizado o caractere '.' (ponto) como separador decimal, ao invés de ',' (vírgula) para facilitar a eventual inserção desses valores no *software* MATLAB.

Tabela A.1: Caso 1 - X_{in}

$u(t - d - 1)$

Tabela A.2: Caso 1 - b

1
1
1
1

Tabela A.3: Caso 1 - W_{in}

0.9000	-0.7195	0.7955	0.1866
--------	---------	--------	--------

Tabela A.4: Caso 1 - W_1

-0.8617	0.9539	0.0509	0.8078
-0.3400	-0.7075	0.6165	0.6078
0.0403	0.9643	-0.6393	0.8964
0.9210	0.9200	-0.1038	0.3064

Tabela A.5: Caso 1 - W_2

-0.9463	0.5432	-0.6663	-0.9077
0.7249	0.4411	0.5534	-0.8057
0.8286	-0.1532	-0.9795	0.6469
0.3334	0.3460	-0.3785	0.3897

Tabela A.6: Caso 1 - W_3

-0.3261	-0.2549	0.0649	0.4793
0.9263	0.5161	-0.0357	-0.4669
-0.9920	0.5890	0.3246	0.4091
-0.1225	-0.6263	0.4187	0.3102

Tabela A.7: Caso 1 - W_4

-0.6748	-0.4156	-0.4551	0.9934
-0.7620	0.1264	0.0337	0.0762
-0.0033	-0.5778	0.4087	-0.7201
0.9195	0.4565	0.8022	-0.7014

Tabela A.8: Caso 1 - W_5

-0.4850	-0.5130	-0.4978	0.6617
0.6681	0.8545	0.2415	0.1638
-0.5076	-0.3381	-0.0074	0.0802
0.6366	-0.6881	-0.3613	0.8421

Tabela A.9: Caso 1 - W_6

-0.4124	0.2265	0.5421	-0.0612
0.5934	-0.8553	0.8409	-0.9762
0.5468	-0.8656	-0.7699	-0.3258
-0.1787	0.1969	0.0890	-0.6756

Tabela A.10: Caso 1 - W_7

0.6777	0.1538	0.5510	0.8344
-0.4264	-0.3830	-0.1555	-0.8001
0.1046	0.3406	-0.8396	0.5441
-0.6687	0.3784	-0.5420	0.0767

Tabela A.11: Caso 1 - W_8

0.9813	0.9711	0.7364	0.7212
-0.8490	-0.9768	-0.8361	-0.0093
-0.1239	0.5518	-0.1985	0.8701
-0.7990	0.7015	-0.4826	-0.4526

Tabela A.12: Caso 1 - W_9

-0.4753	0.1630	0.2388	-0.8481
-0.6919	0.3251	-0.2216	-0.5879
-0.7277	-0.7088	0.0239	-0.7535
0.7499	0.8474	-0.0547	-0.6475

Tabela A.13: Caso 1 - W_{out}

-0.5905
0.6944
-0.9788
0.8028

Apêndice B

Caso de estudo 2

A seguir são apresentadas as matrizes de peso que compõem a rede seleccionada de 8 camadas ocultas e 7 neurônios por camada oculta.

Tabela B.1: Caso 2 - X_{in}

$u(t - d - 2)$

Tabela B.2: Caso 2 - b

1
1
1
1
1
1
1

Tabela B.3: Caso 2 - W_{in}

-0.7099	0.8152	0.2683	-0.8133	-0.4503	0.1267	0.8843
---------	--------	--------	---------	---------	--------	--------

Tabela B.4: Caso 2 - W_1

0.9330	-0.1465	0.8410	0.4121	-0.3671	-0.6536	0.3607
-0.6640	0.8234	0.3507	-0.9363	0.8911	-0.0571	0.3402
0.9554	0.5827	0.5015	-0.4462	-0.9375	-0.1419	-0.6551
0.9167	0.9298	0.4587	-0.9077	-0.1236	0.2657	-0.7618
-0.0228	0.3182	-0.2386	-0.8057	-0.2397	0.3896	0.0035
0.6132	-0.9282	0.2943	0.6469	0.5257	0.4769	0.9360
-0.6945	0.6891	-0.6629	0.3897	0.5803	-0.4849	-0.2875

Tabela B.5: Caso 2 - W_2

0.1640	0.9173	0.6253	-0.1143	0.4945	0.6000	-0.6756
-0.5069	0.0902	-0.4789	-0.3898	0.4777	0.9189	0.5886
0.5120	-0.7229	0.8635	0.5910	-0.2620	-0.6972	-0.3776
-0.4898	-0.7014	-0.3000	0.1705	0.1356	0.1376	0.0571
0.0526	-0.4880	-0.5764	0.0509	-0.8630	-0.0380	-0.6687
0.4023	0.6781	-0.4975	0.8107	-0.8974	-0.9631	0.2040
0.7918	-0.4886	0.2423	-0.4617	0.0447	-0.3046	-0.4741

Tabela B.6: Caso 2 - W_3

0.3064	-0.7150	0.9015	-0.4958	-0.6877	-0.2981	-0.5624
0.3795	0.6442	-0.9888	0.5976	0.7370	0.0265	-0.1537
0.4911	0.0633	0.5151	-0.1459	0.1666	-0.1964	-0.8975
-0.1011	0.9755	0.6170	0.8061	0.1019	-0.8481	0.7892
-0.8385	-0.8550	0.7050	-0.6440	-0.6976	-0.5202	0.8681
-0.5283	-0.1353	-0.7941	-0.4870	0.7768	-0.7534	-0.0638
0.8267	-0.7867	-0.2004	-0.7089	0.2441	-0.6322	-0.0215

Tabela B.7: Caso 2 - W_4

-0.3228	-0.1897	-0.5355	0.4647	0.3751	0.5487	0.6303
0.7961	-0.8070	-0.2825	0.2937	-0.6301	-0.0184	0.5949
-0.2449	-0.7169	0.5184	-0.1025	-0.3045	-0.2196	0.2480
-0.7789	0.8841	-0.9664	0.0940	0.2521	-0.1044	-0.2417
0.6040	0.9434	-0.9212	-0.3900	0.5455	-0.3966	0.6075
-0.2205	0.1504	-0.6620	0.4894	-0.8377	0.0170	0.0657
-0.5090	-0.8736	0.2844	-0.6265	0.8380	0.0137	-0.2900

Tabela B.8: Caso 2 - W_5

0.8876	-0.0581	-0.1450	-0.1134	0.4213	-0.8609	0.1560
0.7606	-0.5390	-0.3957	-0.7693	-0.5560	-0.5110	-0.5256
0.1101	0.6652	0.8070	-0.4851	-0.7654	0.5186	-0.0991
0.2459	-0.6105	-0.1413	-0.1808	-0.4078	-0.9417	0.9262
0.1963	-0.5549	-0.6392	0.2003	-0.3649	0.8036	0.0804
-0.5824	-0.6616	0.8091	-0.4757	-0.1518	0.4501	0.0391
-0.3751	-0.5552	0.9172	0.2219	0.0101	-0.1018	-0.5469

Tabela B.9: Caso 2 - W_6

-0.0979	0.7658	-0.7336	0.8698	-0.0607	0.2219	0.9033
0.2374	0.8249	0.4362	0.7818	-0.0646	-0.6342	0.4199
0.3172	0.5894	-0.8197	-0.3252	0.7229	-0.5117	-0.0179
-0.2139	-0.8015	0.3062	0.4019	0.2144	0.7778	-0.0593
-0.2658	-0.4760	-0.0117	-0.6032	0.2354	-0.9419	-0.8809
0.9545	-0.3306	0.5198	-0.9385	0.6697	-0.0164	0.3537
-0.9252	0.3595	0.4288	0.4881	0.6092	-0.6640	-0.9153

Tabela B.10: Caso 2 - W_7

-0.8467	0.2293	0.6394	-0.8791	-0.1367	-0.0138	-0.1943
0.1076	-0.0929	-0.8409	-0.2015	-0.9690	-0.2818	0.0500
-0.8065	0.9449	-0.7338	0.0538	0.9681	0.9033	0.8851
0.7048	0.2182	-0.6538	-0.1664	-0.6657	0.8818	-0.1859
0.6503	0.4795	-0.2326	0.3137	-0.7876	-0.8851	0.9129
0.4712	-0.1270	0.6622	0.2559	-0.2552	0.4911	-0.4065
-0.6254	-0.2635	0.6009	-0.4160	-0.6038	-0.4160	0.3616

Tabela B.11: Caso 2 - W_{out}

0.8638
-0.8594
-0.0585
-0.1513
-0.0782
0.5589
-0.1924

Apêndice C

Caso de estudo 3

A seguir são apresentadas as matrizes de peso que compõem a rede selecionada de 6 camadas ocultas e 10 neurônios por camada oculta.

Tabela C.1: Caso 3 - X_{in}

$u(t - d - 1)$
$u(t - d - 2)$

Tabela C.2: Caso 3 - b

1
1
1
1
1
1
1
1
1
1

Tabela C.3: Caso 3 - W_{in}

-0.6944	0.1610	-0.4127	0.9508	-0.5482	0.7915	0.6230	-0.2876	0.7128	0.1937
0.8284	-0.8200	0.1538	0.9204	0.9636	-0.0280	-0.7244	0.8076	0.8796	-0.9238

Tabela C.4: Caso 3 - W_1

0.7052	-0.4137	0.5225	0.2026	0.0020	0.6298	0.2489	0.0578	-0.4458	-0.4973
0.8868	-0.8894	0.5791	-0.6596	0.4362	-0.5282	0.1347	0.5547	0.0016	0.8261
0.3683	-0.7820	-0.6334	-0.8280	0.7863	0.8538	0.8926	0.8643	-0.7257	-0.6649
0.5421	0.7252	-0.0662	0.0397	0.9891	-0.3088	-0.3200	-0.7442	0.0617	0.6216
0.4947	0.4280	-0.1213	0.8171	0.0891	-0.6053	0.6001	0.1338	-0.5495	0.1173
-0.1890	-0.3122	0.2589	-0.2614	-0.6545	-0.5121	0.5798	-0.0651	0.1991	0.9635
0.3354	0.9318	0.3973	0.2209	-0.6431	0.2145	-0.1934	-0.9798	0.3051	-0.8621
-0.6448	-0.8866	0.4903	-0.6123	-0.4705	-0.0555	0.2197	-0.3297	0.4044	-0.0933
0.4353	-0.0516	-0.4874	0.5206	0.7386	-0.3039	-0.7445	-0.6796	-0.2281	-0.8039
-0.9161	-0.2188	0.3478	-0.4660	-0.4499	0.6454	-0.8595	0.5850	-0.8868	0.9200

Tabela C.5: Caso 3 - W_2

-0.9919	-0.6410	-0.2981	0.6975	-0.5066	0.7442	0.5597	-0.0646	0.6616	-0.0250
0.5499	-0.4724	0.0265	0.8875	-0.1926	-0.9676	-0.6314	-0.1262	0.0871	-0.5504
0.6326	-0.7134	-0.1964	-0.0801	-0.8006	-0.8619	0.3836	-0.1367	-0.2206	0.7293
0.7380	-0.7283	-0.8481	-0.0610	-0.7346	-0.6248	-0.5826	-0.3975	0.8336	-0.6232
-0.8310	0.7350	-0.5202	-0.4013	0.8912	0.3737	-0.2028	-0.0075	0.7568	-0.5377
-0.2014	0.1594	-0.7534	0.7939	0.9136	0.4671	0.2595	0.0135	0.0998	-0.6459
-0.4807	0.0997	-0.6322	-0.3169	0.1543	0.3660	0.6610	0.6271	0.1606	-0.5305
0.6001	-0.7101	-0.5201	-0.7770	-0.8804	-0.0989	-0.8376	0.5895	0.1724	-0.1278
-0.1364	0.7046	-0.1655	0.5605	-0.5304	0.0923	0.8491	0.2872	-0.5692	-0.3809
0.8200	0.2392	-0.9007	-0.2268	-0.2895	-0.4135	0.4769	-0.2567	-0.2509	0.8791

Tabela C.6: Caso 3 - W_3

-0.1383	0.2057	0.6136	0.0524	0.8266	-0.0270	0.0031	0.7678	-0.9099	0.9459
-0.6304	0.4224	-0.9415	-0.5368	0.5924	0.5580	-0.0401	-0.9427	-0.8571	0.2980
0.8098	-0.5565	0.8577	-0.0222	-0.8026	0.4301	0.8094	-0.0202	0.0433	0.6007
0.9690	-0.7652	0.4879	0.2741	-0.4763	0.7859	0.2063	-0.6708	-0.6967	-0.0924
-0.1221	-0.4066	-0.0195	0.3583	-0.3293	0.7818	0.2309	0.9574	0.6359	-0.1352
-0.7774	-0.3624	0.0843	-0.2092	0.3595	-0.3173	0.7865	0.4254	0.6984	0.6506
-0.4708	-0.1517	-0.4634	-0.2068	-0.7269	0.3499	0.6312	-0.0197	0.6178	-0.8331
-0.1919	0.0157	-0.1065	0.9760	0.4425	-0.5999	0.1720	-0.0578	-0.6837	-0.7337
0.1919	-0.8290	0.8142	-0.8881	-0.7865	-0.9568	-0.5322	-0.8875	0.5301	-0.6532
-0.4343	-0.4750	0.2225	0.8200	0.3075	0.4283	-0.6127	0.3471	0.0934	-0.2181

Tabela C.7: Caso 3 - W_4

0.6628	-0.9784	-0.8947	0.3015	0.7460	-0.6182	0.2392	0.6281	0.2208	-0.1829
0.6067	0.9681	0.4757	0.0783	0.3384	-0.1435	-0.4694	-0.4785	-0.0582	0.6400
-0.8791	-0.5985	-0.4735	0.2918	-0.5831	-0.0360	0.7088	0.1340	0.3638	0.4144
-0.2015	-0.7863	-0.1584	0.3331	-0.2722	-0.7588	0.9734	-0.9542	0.4038	0.9399
0.0538	-0.2552	0.0957	-0.6437	-0.0785	0.1790	0.4605	-0.1495	0.2771	0.0627
-0.1664	-0.6504	0.8722	-0.7440	0.9627	-0.5476	-0.2766	-0.3526	-0.8876	-0.2840
0.3137	-0.0000	-0.1674	0.9899	-0.6796	-0.2308	0.1740	-0.6887	-0.9208	-0.7981
0.2559	-0.3210	0.9668	-0.6578	0.7067	0.1660	-0.7834	-0.6425	-0.3608	0.2210
-0.4160	0.9486	-0.4094	-0.9820	0.3465	-0.4964	0.8691	-0.2110	-0.0103	0.5637
-0.1367	0.8407	0.4022	0.1224	-0.2475	-0.4191	0.7593	-0.8115	0.3089	-0.1531

Tabela C.8: Caso 3 - W_5

-0.8184	0.2754	0.3357	-0.0751	-0.0530	-0.6227	0.2716	0.5406	0.7399	0.0878
-0.4576	0.9287	0.7013	-0.1263	-0.6946	-0.4720	0.8916	-0.2997	-0.4752	0.4421
-0.6972	-0.5196	-0.3128	-0.0893	-0.3178	-0.7908	-0.5829	0.3175	-0.4022	0.0450
-0.4230	0.3523	0.5668	0.5404	0.2105	0.1349	0.4244	-0.1677	-0.7342	0.9874
-0.1257	-0.4343	0.3410	-0.3897	-0.6166	0.4588	-0.5291	0.6749	0.8071	-0.5626
0.0543	0.3436	-0.9866	0.5695	0.4769	0.0932	-0.7612	0.6658	0.2911	-0.7884
-0.1854	0.4343	0.2185	-0.0006	-0.5415	0.2472	0.1636	-0.4970	-0.3716	-0.7806
0.7180	-0.8611	-0.2339	-0.9258	0.8258	0.3716	-0.1148	0.2270	0.2289	-0.8728
-0.0256	-0.4624	0.8376	-0.6049	-0.4739	0.4043	-0.1084	0.1646	-0.0173	-0.1908
0.8145	-0.5270	-0.9945	0.4751	0.5141	0.6137	0.2898	0.0758	0.0859	-0.1033

Tabela C.9: Caso 3 - W_{out}

0.0369
-0.1630
-0.2224
-0.2552
0.1133
0.9168
-0.0065
-0.0888
-0.5821
-0.9511

Apêndice D

Caso de estudo 4

A seguir são apresentadas as matrizes de peso que compõem a rede selecionada de 5 camadas ocultas e 7 neurônios por camada oculta.

Tabela D.1: Caso 4 - X_{in}

$y(t - 1)$

Tabela D.2: Caso 4 - b

1
1
1
1
1
1
1

Tabela D.3: Caso 4 - W_{in}

-0.7440	0.2977	0.5097	-0.8046	-0.4438	-0.0236	0.3406
---------	--------	--------	---------	---------	---------	--------

Tabela D.4: Caso 4 - W_1

0.9297	-0.1558	0.8691	0.4121	-0.3660	-0.6277	0.3602
-0.6736	0.5796	0.4240	-0.9363	0.5950	-0.0240	0.3130
0.9613	0.3763	0.5908	-0.4462	-1.0590	-0.1127	-0.6715
0.9143	0.9195	0.4874	-0.9077	-0.1227	0.2913	-0.7613
-0.0295	0.3126	-0.2140	-0.8057	-0.2371	0.4167	-0.0022
0.5986	-0.9210	0.3083	0.6469	0.5269	0.5065	0.9215
-0.7049	0.4158	-0.5651	0.3897	0.2575	-0.4515	-0.3163

Tabela D.5: Caso 4 - W_2

0.1720	0.9134	0.6321	-0.0697	0.5019	0.5742	-0.6714
-0.5509	-0.0061	-0.5089	-0.2902	1.0111	0.3900	0.4360
0.5040	-0.7349	0.8627	0.6549	-0.2772	-0.7126	-0.3754
-0.4898	-0.7014	-0.3000	0.1705	0.1356	0.1376	0.0571
0.0126	-0.4868	-0.6064	0.1001	-0.4534	-0.2543	-0.7182
0.3989	0.6794	-0.4973	0.8351	-0.8898	-0.9763	0.2039
0.7825	-0.4933	0.2325	-0.4277	0.0637	-0.3259	-0.4741

Tabela D.6: Caso 4 - W_3

0.3010	-0.6936	0.9275	-0.4808	-0.7279	-0.2981	-0.5232
0.3752	0.6524	-0.9851	0.5999	0.7386	0.0265	-0.1631
0.4882	0.0786	0.5248	-0.1377	0.1594	-0.1964	-0.9272
-0.1063	0.9939	0.6166	0.8208	0.0997	-0.8481	0.7844
-0.8414	-0.8417	0.6579	-0.6369	-0.7679	-0.5202	0.5832
-0.5423	-0.1146	-0.8567	-0.4724	0.1443	-0.7534	0.2372
0.8267	-0.7867	-0.2156	-0.7089	0.0219	-0.6322	-0.0088

Tabela D.7: Caso 4 - W_4

-0.3246	-0.1922	-0.5297	0.4646	0.3736	0.5467	0.6357
0.8001	-0.8071	-0.2935	0.2959	-0.6330	-0.0279	0.5898
-0.2615	-0.7361	0.4365	-0.0887	-0.4310	-0.1518	0.3148
-0.7776	0.8841	-0.9690	0.0944	0.2512	-0.1082	-0.2426
0.7078	0.5659	-0.9140	-0.4074	0.4805	-0.3873	0.4675
-0.2205	0.1504	-0.6620	0.4894	-0.8377	0.0170	0.0657
-0.4993	-0.9101	0.0579	-0.6213	0.5859	0.0468	-0.2716

Tabela D.8: Caso 4 - W_{out}

0.8724
-0.6871
-0.1355
-0.1556
-0.0979
0.5300
-0.0901

Apêndice E

Caso de estudo 5

A seguir são apresentadas as matrizes de peso que compõem a rede selecionada de 6 camadas ocultas e 8 neurônios por camada oculta.

Tabela E.1: Caso 5 - X_{in}

$y(t - 1)$
$y(t - 1)$

Tabela E.2: Caso 5 - b

1
1
1
1
1
1
1
1

Tabela E.3: Caso 5 - W_{in}

0.6471	-0.7971	0.0921	0.8420	0.9013	-0.1730	-0.7352	0.7144
0.6665	-0.4260	0.8668	-0.7481	1.1325	0.1984	-0.2076	0.7543

Tabela E.4: Caso 5 - W_1

0.8997	-0.2155	0.5439	0.5583	0.4103	-0.3525	-0.8048	-0.1254
0.3140	0.3110	0.3863	-0.6285	0.3088	0.4977	-0.7083	-0.6094
-0.7393	-0.6576	-0.4447	-0.1027	-0.6242	-0.4554	-0.5504	-0.3675
0.1999	0.4121	0.8320	0.1257	-0.6454	0.6436	0.6179	0.5068
1.1880	-0.9363	-1.0772	0.0463	-0.0272	0.2788	-0.5989	0.0480
0.3549	-0.4462	-0.1662	0.4717	1.0007	0.9386	0.5839	-0.1665
0.5247	-0.9077	-0.2417	0.5069	-0.3213	0.9022	-0.5233	0.6435
0.5972	-0.8057	0.4156	-0.5515	0.1915	0.1600	0.7708	0.3081

Tabela E.5: Caso 5 - W

0.2863	-0.8921	-0.3258	0.6570	0.7996	0.4314	0.6829	-0.7101
0.8344	0.0616	-0.6756	0.3784	0.0767	0.6346	-0.6363	0.7061
-0.4455	0.5583	0.5886	0.4753	0.9716	0.7465	-0.4681	0.2441
0.6031	0.8680	-0.3776	-0.1503	-0.8781	-0.9044	-0.6915	-0.2981
0.5720	-0.7402	0.0571	-0.8606	-0.0995	-0.2570	-0.7442	0.0265
-0.1887	0.1376	-0.6687	-0.7110	-0.8939	-0.5393	0.7785	-0.1964
0.1286	-0.0612	0.2040	0.8184	0.9167	0.6027	0.1616	-0.8481
-0.8300	-0.9762	-0.4741	-0.7181	-1.0092	-0.1509	0.1095	-0.5202

Tabela E.6: Caso 5 - W_3

-0.8439	-0.0150	-0.3215	-0.3076	-0.0982	0.2423	-0.4702	0.0627
-0.6322	-0.3246	-0.8071	0.6424	0.0940	0.5605	0.0170	-0.2985
-0.5201	0.8001	-0.7361	-0.9692	-0.4074	-0.8377	0.0215	0.8780
-0.1185	-0.2613	0.7643	-0.9185	0.4894	0.7742	0.5281	0.7175
-0.8809	-0.7772	0.8431	-0.6675	-0.6221	0.5852	0.5406	0.1688
0.8183	0.5634	0.1138	0.2925	0.3736	-0.0135	0.2665	0.2837
0.8040	-0.1937	-0.9826	0.4280	-0.6330	-0.0543	-0.2960	0.3235
-0.0183	-0.5166	-0.5304	0.2955	-0.2630	-0.1064	0.6232	-0.5845

Tabela E.7: Caso 5 - W_4

-0.3975	-0.1224	-0.7587	-0.7651	-0.9226	0.0633	0.9757	0.3619
-0.0582	-0.3788	-0.4839	-0.4066	0.8575	0.0419	-0.9251	-0.7277
-0.5390	0.6424	-0.1027	-0.3624	0.3917	-0.6251	0.7703	0.4051
0.6886	-0.1209	0.1898	-0.1517	-0.0006	-0.0148	0.8254	-0.7754
-0.6105	-0.6304	-0.4756	0.0157	0.1571	0.2481	0.5924	0.3075
-0.5482	0.6446	0.3430	-0.8287	-0.4062	0.5217	-0.7941	-0.0412
-0.6586	0.8047	0.4887	-0.4750	-0.1026	-0.2165	-0.4770	0.5334
-0.5447	-0.2390	-0.4272	0.6023	1.0527	-0.1023	-0.3194	0.4116

Tabela E.8: Caso 5 - W_5

0.8074	-0.0402	-0.5201	-0.0578	0.6351	-0.0924	0.6067	-0.1367
0.7109	0.5781	0.6274	-0.8961	0.2906	-0.1317	-0.8823	-0.9690
-0.3366	0.1803	-0.9237	0.3639	-0.7520	0.6522	-0.2015	0.9681
0.3975	0.2353	-0.0202	-0.9151	0.3192	-0.8331	0.0538	-0.6657
-0.6454	0.5846	-0.7281	-0.8644	-0.0426	-0.7349	-0.1702	-0.7876
-0.9922	0.4723	0.9750	0.0304	0.7467	-0.6405	0.3102	-0.2552
0.4793	0.1515	0.4260	-0.8067	0.2960	-0.1951	0.2598	-0.6038
-0.0421	-0.7701	-0.0940	0.6083	0.5348	0.6806	-0.4205	-0.0206

Tabela E.9: Caso 5 - W_{out}

-0.0938
-0.2255
0.1917
-0.7008
-0.4009
0.7098
0.1508
0.8592

Apêndice F

Caso de estudo 6

A seguir são apresentadas as matrizes de peso que compõem a rede seleccionada de 4 camadas ocultas e 6 neurônios por camada oculta.

Tabela F.1: Caso 6 - X_{in}

$y(t - 1)$
$y(t - 1)$

Tabela F.2: Caso 6 - b

1
1
1
1
1
1

Tabela F.3: Caso 6 - W_{in}

0.2213	-0.2050	1.0342	-0.6161	0.7005	0.7741
-0.7756	0.1633	1.1789	0.8703	-0.3779	-0.6267

Tabela F.4: Caso 6 - W_1

-0.1662	0.7025	0.3267	-0.7931	-0.1244	-0.1277
0.8214	0.9382	-0.5574	0.6523	-0.3337	0.2591
0.6279	0.2453	0.7262	0.3117	0.7762	0.5360
0.9212	0.5444	-0.7074	-0.3855	0.5281	0.5041
0.3042	0.5335	-0.3997	0.9032	-0.6179	-0.4471
-0.9566	-0.1163	-0.9511	-0.9036	-0.1051	0.3261

Tabela F.5: Caso 6 - W_2

0.2272	0.1824	0.7927	0.7516	-0.6068	0.2098
-0.7375	-0.6232	0.9692	-0.5024	-0.4978	0.0961
-0.6621	0.5088	0.1003	0.6596	0.2321	0.8607
-0.0461	-0.5172	-0.7294	-0.4635	-0.0534	-0.3616
0.8909	0.0265	-0.7029	0.9068	-0.2967	0.5650
-0.3666	0.3668	-0.4430	-0.2958	0.6617	0.5074

Tabela F.6: Caso 6 - W_3

-0.2388	0.8440	-0.6294	-0.4725	-0.5130	-0.8314
0.1358	-0.7414	0.5833	0.3095	0.8225	-0.1167
-0.8463	0.1239	-0.4275	0.3935	-0.7520	-0.8064
-0.9077	-0.0918	0.1158	0.5306	0.6716	0.9283
0.0616	-0.9762	-0.6687	-0.0989	0.0767	-0.9907
0.5347	-0.3880	0.2308	-0.8196	0.9791	0.5410

Tabela F.7: Caso 6 - W_{out}

0.9022
0.8318
-0.8371
0.5014
-0.4859
-0.1887