

Bernardo Barancelli Schwedersky

**AVALIAÇÃO EXPERIMENTAL DE REDES NEURAIS ARTIFICIAIS
RECORRENTES PARA IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS:
UM ESTUDO DE CASO**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Rodolfo César Costa Flesch, Dr. Eng.

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Schwedersky, Bernardo Barancelli
Avaliação experimental de redes neurais
artificiais recorrentes para identificação de
sistemas dinâmicos : um estudo de caso / Bernardo
Barancelli Schwedersky ; orientador, Rodolfo César
Costa Flesch, 2017.
123 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia de Automação e Sistemas,
Florianópolis, 2017.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2.
identificação de sistemas não lineares. 3. redes
neurais artificiais recorrentes. 4. rede de estado
de eco. 5. NARX. I. Flesch, Rodolfo César Costa .
II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia de Automação e
Sistemas. III. Título.

Bernardo Barancelli Schwedersky

**AVALIAÇÃO EXPERIMENTAL DE REDES NEURAIS ARTIFICIAIS
RECORRENTES PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS: UM ESTUDO DE CASO**

Esta Dissertação foi julgada adequada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 28 de agosto de 2017.

Prof. Daniel Ferreira Coutinho, Dr. Eng.
Coordenador do Curso
Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Rodolfo César Costa Flesch, Dr. Eng.
Orientador
Departamento de Automação e Sistemas - UFSC

Prof. Julio Elías Normey Rico, Dr.
Presidente
Departamento de Automação e Sistemas - UFSC

Prof. Jomi Fred Hübner, Dr. Eng.
Departamento de Automação e Sistemas - UFSC

Prof. Mauro Roisenberg, Dr. Eng.
Departamento de Informática e Estatística - UFSC

AGRADECIMENTOS

O autor gostaria de agradecer a todos que contribuíram para o desenvolvimento deste trabalho, em especial ao seu orientador Rodolfo César Costa Flesch, o qual demonstrou dedicação ímpar para o enriquecimento deste trabalho.

A todos os colegas do LIAE, em especial ao Hiago Danguí, Ahryman Nascimento e Eric Antonello, que contribuíram no desenvolvimento e foram essenciais para alcançar os objetivos da dissertação.

Agradecimentos especiais aos meus pais, Werner e Rosa, irmãos, Mateus e Maurício e à Patricia, pelo incentivo e apoio durante todo o curso de mestrado.

Por fim, o autor agradece à Universidade Federal de Santa Catarina e ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas pela oportunidade de realizar este curso e à CAPES pelo suporte financeiro.

RESUMO

Este trabalho apresenta um estudo experimental de redes neurais artificiais recorrentes em problemas de identificação de sistemas dinâmicos não lineares. Três arquiteturas recorrentes encontradas na literatura foram empregadas em um estudo de caso que consiste na identificação do comportamento dinâmico de uma bancada de ensaios que reproduz as possíveis condições de operação de compressores de refrigeração em sistemas (refrigeradores, congeladores, expositores, entre outros). As três arquiteturas exploradas no trabalho são: a rede não linear autorregressiva com entradas exógenas, a rede *perceptron* de múltiplas camadas recorrente e a rede de estado de eco. Os modelos treinados empregando cada arquitetura foram testados em tarefas de simulação e predição e os resultados experimentais mostram que as três arquiteturas são capazes de modelar de forma satisfatória os principais elementos da resposta dinâmica do sistema empregado como estudo de caso. Os modelos que empregaram redes de estado de eco apresentaram tempos de treinamento cerca de mil vezes menores que os apresentados pelas demais redes, o que facilita a otimização do processo de treinamento e permite que sejam alcançados resultados melhores tanto para tarefas de simulação quanto predição. O melhor resultado foi obtido por uma das abordagens que empregou redes de estado de eco, a qual alcançou erros para a tarefa de simulação 10% menores que os obtidos pela rede *perceptron* de múltiplas camadas recorrente e até 55% menores que os obtidos pela rede não linear autorregressiva com entradas exógenas.

Palavras-chave: identificação de sistemas não lineares; redes neurais artificiais recorrentes; rede de estado de eco; NARX; RMLP

ABSTRACT

This work presents an experimental study about the use of recurrent artificial neural networks for the identification of nonlinear dynamic systems. Three recurrent architectures from literature are used in a case study, which consists in the identification of the dynamic behavior of a test rig that can impose to a refrigeration compressor the possible operating conditions found in cooling systems (refrigerators, freezers, beverage coolers, among others). The three architectures explored in the work are: the nonlinear autoregressive network with exogenous inputs, the recurrent multilayer perceptron network, and the echo state network. The trained models were tested in simulation and prediction tasks and the experimental results showed that the three architectures are able to satisfactorily model the main dynamics of the system used as case study. The models that used echo state networks presented training times about a thousand times shorter than the ones presented by the other architectures, which makes the optimization of the training process easier and allows better results to be obtained for both simulation and prediction tasks. The best results were obtained by an echo state network implementation, which presented an error 10% lower than a recurrent multilayer perceptron implementation and 55% lower than a nonlinear autoregressive with exogenous input network, for the simulation task.

Keywords: nonlinear system identification; artificial neural networks; echo state network; NARX; RMLP

LISTA DE FIGURAS

2.1	Exemplo da realização de um preditor de duas amostras à frente.	32
2.2	Estratégia da dinâmica externa.	34
2.3	Modelos nas configurações série-paralelo, usado para predição, e paralelo, usado para simulação	35
2.4	Ajuste de três modelos polinomiais para dados de exemplo.	39
2.5	Representação do dilema da relação entre bias e variância. O erro de modelagem é decomposto em uma parcela de erro de bias e uma parcela de erro de variância. (a) Exemplo com variância alta. (b) Exemplo com variância reduzida em comparação a (a).	41
2.6	Exemplo de sinais de excitação PRBS e APRBS.	43
2.7	Modelo geral de um neurônio artificial.	47
2.8	Arquiteturas recorrentes básicas (nem todas as conexões são mostradas). A rede de Jordan apresenta “entradas de estado”, enquanto a rede de Elman apresenta uma “camada de contexto”, que apresentam funcionamento semelhante.	49
2.9	Arquitetura da rede NARX (nem todas as conexões são mostradas).	50
2.10	Diagrama de blocos para as camadas da rede NARX. Os blocos tracejados representam matrizes treináveis do modelo.	52
2.11	Diagrama de blocos genérico para uma camada i da rede RMLP. Os blocos tracejados representam matrizes treináveis do modelo.	54
2.12	Desdobramento no tempo efetuado pelo BPTT	55
2.13	Arquitetura da rede ESN. Linhas sólidas representam conexões fixas, enquanto tracejados representam pesos treináveis.	58
2.14	Diagrama de blocos completo da rede ESN.	59
2.15	Diagrama de blocos para unidades integradoras com vazamento. Por simplicidade, são ignoradas as matrizes de realimentação externa e bias nessa representação.	61
3.1	Fluxograma do laço da identificação de sistemas usando redes neurais recorrentes.	66
3.2	Diagrama simplificado de processo e instrumentos para bancada de emulação de condições de operação aplicada a compressores herméticos de refrigeração.	69
3.3	Porção do sinal de excitação APRBS.	72

3.4	Porções do ensaio de identificação com as respostas do sistema à excitação APRBS. São mostradas a pressão de sucção, P_s , a pressão de descarga, P_d , a tensão na válvula de sucção, V_s , e a tensão na válvula de descarga, V_d . . .	73
4.1	Busca em grade para escolha da melhor configuração de atrasos para a entrada e a saída realimentada para a rede NARX. Células mais escuras representam menores erros quadráticos médios. O círculo branco indica o menor erro.	78
4.2	Validação cruzada realizada durante o treinamento da rede NARX. A rede associada à época com menor erro no conjunto de validação é selecionada.	79
4.3	Resultados da rede NARX para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo NARX.	81
4.4	Resultados da rede NARX para a tarefa de predição de N instantes à frente. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo NARX.	82
4.5	Validação cruzada realizada durante o treinamento da rede RMLP. A rede associada à época com menor erro no conjunto de validação é selecionada.	83
4.6	Resultados da rede RMLP para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo RMLP.	85
4.7	Configuração da rede ESN sem realimentações externas.	86
4.8	Diagrama de blocos da rede ESN simplificada. Blocos tracejados representam as matrizes treináveis do modelo. .	88
4.9	Comportamento do erro quadrático médio associado a diferentes valores do parâmetro de regularização. . . .	89
4.10	Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNS. Células mais escuras representam menores erros quadráticos médios. Círculo branco indica a célula com melhor desempenho.	91
4.11	Estados do reservatório para diferentes redes ESN. Erros de projeto podem ser identificadas ao analisar o comportamento dos estados.	93
4.12	Resultados da aplicação da rede ESN no conjunto de testes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNS.	95
4.13	Configuração da rede ESN com realimentação da saída com a saída.	97

4.14 Diagrama de blocos da rede ESN com realimentação da saída para a saída. Blocos tracejados representam matrizes treináveis do modelo.	98
4.15 Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNRS. Células mais escuras representam menores erros quadráticos médios.	100
4.16 Resultados da rede ESNRS para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa as saída para o modelo ESNRS.	102
4.17 Resultados da rede ESNRS para a tarefa de previsão de N instantes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRS. . . .	103
4.18 Configuração da rede ESN com realimentação da saída com o reservatório.	104
4.19 Diagrama de blocos da rede ESN com realimentação da saída para o reservatório. Blocos tracejados representam matrizes treináveis do modelo.	105
4.20 Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNRR. Células mais escuras representam menores erros quadráticos médios.	107
4.21 Resultados da rede ESNRR para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRR.	108
4.22 Resultados da rede ESNRR para a tarefa de previsão de N instantes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRR. . . .	109
4.23 Comparação entre as arquiteturas de redes recorrentes. .	114

LISTA DE TABELAS

2.1	Elementos da figura 2.14	60
4.1	Tempo médio de treinamento para as arquiteturas de redes.	110
4.2	Tempos médio e máximo para a utilização dos modelos treinados. Resultados representam tempos para a execução de 500 predições por cada modelo.	111
4.3	Soma dos erros quadráticos para as diversas configurações de RNA testadas para simulação.	111
4.4	Soma dos erros quadráticos para as diversas configurações de RNA testadas para a predição de N instantes à frente.	112

LISTA DE SIGLAS

APRBS	sinal binário pseudo-aleatório, modulado em amplitude - do inglês <i>Amplitude modulated Pseudo-Random Binary Signal</i>
BPDC	retropropagação decorrelacionada - do inglês <i>BackProagation-DeCorrelation</i>
BPTT	retropropagação através do tempo - do inglês <i>BackPropagation Through Time</i>
EQM	Erro Quadrático Médio
ESN	rede de estado de eco - do inglês <i>Echo State Network</i>
ESNRR	ESN com Realimentação para o Reservatório
ESNRS	ESN com Realimentação para a Saída
ESNS	ESN Simplificada
IA	Inteligência Artificial
LSM	máquina de estado líquido - do inglês <i>Liquid State Machine</i>
LSTM	memória de longo curto prazo - do inglês <i>Long Short-Term Memory</i>
MLP	<i>perceptron</i> de múltiplas camadas - do inglês <i>MultiLayer Perceptron</i>
MPC	controle preditivo baseado em modelo - do inglês <i>Model Predictive Control</i>
NARX	modelo não linear, autorregressivo de média móvel com entrada exógena - do inglês <i>Nonlinear AutoRegressive Moving Average model with eXogenous input</i>
NARX	modelo não linear, autorregressivo com entrada exógena - do inglês <i>Nonlinear AutoRegressive model with eXogenous input</i>
NOE	modelo não linear do erro de saída - do inglês <i>Nonlinear Output Error</i>
PRBS	sinal binário pseudo-aleatório, modulado em amplitude - do inglês <i>Pseudo-Random Binary Signal</i>
RBF	função de base radial - do inglês <i>Radial Basis Function</i>
ReLU	unidade linear retificadora - do inglês <i>Rectifier Linear Unit</i>
RMLP	<i>perceptron</i> de múltiplas camadas recorrente - do inglês <i>Recurrent MultiLayer Perceptron</i>
RNA	Rede Neural Artificial
RTRL	aprendizado recorrente em tempo real - do inglês <i>Real Time Recurrent Learning</i>

SRN rede recorrente simples - do inglês *Simple Recurrent Network*
TDL linha de atrasos discretos - do inglês *Tap Delay Line*
UFSC Universidade Federal de Santa Catarina

LISTA DE SÍMBOLOS

$a(\cdot)$	aproximador não linear estático
b_k	bias do neurônio artificial k
c_{de}^{para}	taxa de conexão da matriz de pesos \mathbf{W}_{de}^{para}
$e(t)$	erro temporal entre a resposta do processo e a saída do modelo
$f(\cdot)$	função de transferência dos neurônios da rede ESN
$g(\cdot)$	função de transferência para pós-processamento da rede ESN
q^{-1}	atraso discreto simples
$h(\cdot)$	função de ativação do neurônio artificial
\mathbf{I}	matriz identidade
N	tamanho do horizonte de predição
N_{var}	dimensão da variável var
n_i	número de padrões de entrada do modelo
n_x	número de atrasos discretos associados à entrada x
n_y	número de atrasos discretos associados à saída y
$q(\cdot)$	função que mapeia as entradas no estado interno do modelo
r	rede neural genérica
$s(\cdot)$	função que mapeia as o estado interno nas saídas do modelo
T	número de amostras de tempo usados no treinamento
$u(t)$	entrada temporal do modelo
\mathbf{u}	vetor de entradas para rede ESN
$\mathbf{u}_d(t)$	vetor formado pelas entradas do modelo no instante t até $t+n_u$
v_{de}^{para}	escalamento da matriz de pesos \mathbf{W}_{de}^{para}
v_k	soma das entradas ponderadas e bias do neurônio artificial k
\mathbf{W}^{out}	matriz de pesos sinápticos para a camada de saída

$\mathbf{W}_{\text{bias}}^i$	matriz diagonal de pesos sinápticos de bias para os neurônios artificiais da camada i
$\mathbf{W}_{\text{de}}^{\text{para}}$	matriz de pesos sinápticos que conecta duas camadas da rede ESN
\mathbf{W}_i^i	matriz de pesos sinápticos de realimentação da camada i
$w_i^{\text{alvo}}(t)$	pesos sinápticos do mecanismo de saída associados à saída i
\mathbf{W}_{i-1}^i	matriz de pesos sinápticos da camada $i - 1$ para a camada i
w_{km}	peso sináptico m do neurônio artificial k
$\mathbf{W}_{\text{out}}^1$	matriz de pesos da realimentação entre a saída e a entrada
$\hat{x}(t)$	estado interno do modelo
\mathbf{x}	vetor de estados do reservatório da rede ESN
\mathbf{X}	matriz formada pelos vetores de estados do reservatório, em um período de T amostras
x_{km}	entrada m do neurônio artificial k
$y(t)$	resposta temporal do processo
\mathbf{y}	vetor de saídas para rede ESN
\mathbf{Y}	matriz formada pelo vetor de saídas em um período de T amostras
$\hat{y}(t)$	predição do modelo para o valor da saída no instante t
$\mathbf{y}_{dy}(t)$	vetor formado pelas saídas do modelo no instante t até $t-n_y$
\mathbf{Y}^{alvo}	matriz formada pelo vetor de saídas alvo em um período de T amostras
$\mathbf{y}_i(t)$	vetor de saídas da camada i do modelo RMLP
$y_i(t)$	saída i do modelo ESN
$y_i^{\text{alvo}}(t)$	saída alvo i
y_k	saída do neurônio artificial k
α	taxa de vazamento do reservatório
β	coeficiente de regularização da função objetivo

$\varphi(t)$ vetor de regressão

$\rho(\cdot)$ raio espectral

SUMÁRIO

1	Introdução	25
1.1	Objetivos	28
1.1.1	Objetivo Geral	28
1.1.2	Objetivos Específicos	28
1.2	Estrutura do Documento	28
2	Revisão da literatura	31
2.1	Identificação de sistemas dinâmicos não lineares . . .	31
2.1.1	Dinâmica externa	34
2.1.2	Dinâmica interna	37
2.1.3	Aprimoramento da complexidade do modelo .	37
2.1.4	Excitação de processos não lineares	41
2.2	Redes neurais artificiais	42
2.2.1	Redes <i>perceptron</i> de múltiplas camadas dinâmicas	45
2.2.1.1	Neurônio MLP	46
2.2.1.2	Redes de Jordan e Elman	48
2.2.1.3	Rede neural NARX	49
2.2.1.4	Rede neural RMLP	53
2.2.1.5	Treinamento de arquiteturas MLP recorrentes	54
2.2.2	Computação de reservatório	56
2.2.2.1	Redes neurais de estado de eco . . .	57
2.2.2.2	Treinamento da ESN	61
2.3	Comentários Finais	63
3	Estudo de Caso	65
3.1	Metodologia	65
3.2	Estudo de caso	68
3.2.1	Descrição do processo	68
3.2.2	Ensaio de identificação	70
3.2.3	Série de dados e a definição das entradas dos modelos	72
3.3	Comentários Finais	74
4	Identificação com redes neurais recorrentes	77
4.1	Rede Neural NARX	77
4.1.1	Construção do modelo e treinamento	77
4.1.2	Resultados	80
4.2	Rede Neural RMLP	82

4.2.1	Construção do modelo e treinamento	83
4.2.2	Resultados	83
4.3	Rede Neural ESN	84
4.3.1	Arquitetura ESN simplificada	86
4.3.1.1	Construção do modelo e treinamento	89
4.3.1.2	Resultados	94
4.3.2	Arquiteturas com realimentações externas . .	94
4.3.3	ESN com realimentação para a saída	96
4.3.3.1	Construção do modelo e treinamento	99
4.3.3.2	Resultados	101
4.3.4	ESN com realimentação para o reservatório . .	102
4.3.4.1	Construção do modelo e treinamento	106
4.3.4.2	Resultados	106
4.4	Comparação dos resultados	108
5	Considerações Finais	115
5.1	Conclusões	115
5.2	Sugestões para trabalhos futuros	117
	Referências	119

1 INTRODUÇÃO

Existe um grande otimismo em torno da inteligência artificial (IA) e do aprendizado de máquina, devido ao seu potencial para melhorar a vida das pessoas, auxiliando na resolução de alguns dos maiores desafios e incapacidades encontrados no mundo moderno. Hoje, a IA é um campo promissor, com diversas linhas de pesquisa ativas, e possui um potencial impacto transformativo muito grande, comparado aos avanços recentes decorrentes da popularização da computação móvel. A IA já avança a passos largos em diversas áreas, como medicina e engenharia, com avanços expressivos em áreas como a identificação de imagens e fala (NATIONAL SCIENCE AND TECHNOLOGY COUNCIL, 2016; KRITTANAWONG, 2017; DETTERMAN, 2011; GOODFELLOW et al., 2016).

Desenvolvimentos recentes da IA, em especial na área denominada do aprendizado profundo (do inglês *deep learning*), possibilitaram um intenso crescimento nas pesquisas e no interesse público em tais técnicas. A resolução de problemas desafiadores, como a visão computacional, o processamento de linguagem natural, o reconhecimento de áudio e a tradução automática, os quais apresentavam avanços lentos com baixa perspectiva de solução, já é uma realidade no mundo moderno, possível apenas devido ao aprendizado profundo (KRIZHEVSKY et al., 2012; DENG et al., 2013; GOLDBERG, 2015; GOODFELLOW et al., 2016).

A área do aprendizado profundo engloba várias arquiteturas de redes neurais artificiais (RNA), que se assemelham pelo emprego de estruturas com mais de uma camada escondida. Esses modelos vêm se popularizando, principalmente na última década, com estruturas específicas se destacando em determinadas aplicações. Uma das mais conhecidas, a rede convolucional, é aplicada extensivamente em tarefas de visão computacional, com resultados largamente superiores aos de outras abordagens. Outro modelo que se destaca é a rede recorrente denominada memória de longo curto prazo (LSTM, do inglês *Long Short-Term Memory*), a qual apresenta resultados expressivos em tarefas de identificação de áudio. Outras arquiteturas recorrentes apresentam um interesse crescente, como as redes de estado de eco (ESN, do inglês *Echo-State Networks*), usadas com sucesso para modelar problemas dinâmicos (LECUN et al., 1998; HOCHREITER; SCHMIDHUBER, 1997; JAEGER; HAAS, 2004).

No ramo da engenharia, diversas áreas podem ser beneficiadas por tais avanços da IA. A identificação de sistemas dinâmicos, em especial, é uma delas. Essa é uma área de estudos im-

portante, que auxilia no desenvolvimento de soluções de controle e otimização, além de permitir a detecção de falhas. O modelamento fenomenológico de sistemas reais não é uma tarefa trivial, devido à alta complexidade do desenvolvimento, demandando muito tempo e representando uma das tarefas mais custosas das etapas de projeto. Para contornar essas limitações, o uso de técnicas de identificação consiste em uma alternativa interessante (NELLES, 2001; ESTRADA-FLORES et al., 2006).

Na campo da identificação de sistemas dinâmicos, os maiores desafios se encontram na vertente que explora soluções para sistemas não lineares. A obtenção de modelos não lineares é uma tarefa complexa, o que faz com que aplicações dependentes de modelos não lineares apresentem, ainda, baixa aplicação em problemas industriais. Em muitas soluções reais, modelos lineares aproximados ainda são usados, mesmo que não representem bem o sistema real e suas não linearidades. A falta de técnicas gerais para a obtenção de modelos não lineares é também um desafio, fazendo com que aplicações dependentes desses modelos sejam ainda pouco exploradas em comparação com alternativas que empregam aproximações lineares. Esse é o cenário, por exemplo, no controle preditivo baseado em modelo (ADENIRAN; FERIK, 2017; CAMACHO, 2007).

Modelos caixa-preta, como as RNAs, são uma alternativa para a obtenção de modelos não lineares através de identificação. Aplicações utilizando RNAs surgiram no início da década de 1990, após a apresentação de estratégias para criar e treinar arquiteturas capazes de realizar processamento temporal, por Werbos (1990). De forma geral, os trabalhos presentes na literatura exploram, em especial, uma classe de arquiteturas de redes neurais artificiais. Essa classe estende o modelo *perceptron* de múltiplas camadas (MLP, do inglês *MultiLayer Perceptron*), modelo mais popular de RNA para problemas estáticos, para que ele possa tratar problemas dinâmicos. Tais arquiteturas apresentam conexões recorrentes, podendo ser empregadas para modelar problemas com contexto temporal, inclusive sistemas dinâmicos não lineares. Essa classe de arquiteturas apresenta diversos modelos distintos, desde versões simplificadas como as apresentadas por Jordan (1986) e Elman (1990), até versões genéricas, que possibilitam o treinamento dos pesos de realimentação, caso da rede não linear autorregressiva com entrada exógena (NARX, do inglês *Nonlinear AutoRegressive with exogenous inputs*) e da rede *perceptron* de múltiplas camadas recorrente (RMLP, do inglês *Recurrent MultiLayer Perceptron*) (NARENDRA; PARTHASARATHY, 1990; PARLOS et al., 1991; PUSKORIUS;

FELDKAMP, 1994).

A aplicação dessas classes de RNAs apresentaram resultados promissores, mas que foram, de certa forma, limitados por algumas características indesejadas, como o alto tempo do treinamento dessas redes recorrentes, e o fenômeno do desaparecimento do gradiente (do inglês *vanishing gradient*), o qual afeta redes MLP com diversas camadas escondidas e também redes recorrentes derivadas do modelo MLP. Tal problema decorre da utilização do algoritmo da retropropagação do erro (do inglês *error backpropagation*) para o treinamento da rede, o qual é a alternativa comumente aplicada na literatura. Esse algoritmo propaga os sinais de erro da saída para a entrada, os quais decaem exponencialmente, fazendo com que o erro nas camadas iniciais desapareça ou cresça indefinidamente, em casos que a rede é profunda (SCHMIDHUBER, 2015; HOCHREITER, 1998).

Pesquisas mais recentes exploram um novo paradigma de aprendizado para problemas dinâmicos, chamado de computação de reservatório. Esse paradigma apresenta uma nova visão para o treinamento de redes recorrentes, buscando eliminar muitos dos problemas encontrados no treinamento de modelos MLP recorrentes, como o desaparecimento do gradiente. As principais arquiteturas que se aproveitam desse paradigma são as redes de estado de eco (ESN, do inglês *Echo-State Network*), proposta por Jaeger (2002) e a máquina de estado líquido (LSM, do inglês *Liquid State Machine*), proposta por Maass et al. (2002).

Os trabalhos que utilizam computação de reservatório em tarefas de identificação de sistemas são ainda incipientes. Alguns exemplos de aplicações são o trabalho de Pan e Wang (2012), que identificou um sistema não linear genérico para uso em controle preditivo baseado em modelo, e o de Antonelo et al. (2017), que aplicou computação de reservatório para construir um modelo para a pressão de fundo de poço em uma aplicação na indústria do petróleo. Tais exemplos mostram que a utilização dessas arquiteturas nessa classe de problemas é promissora, motivando a pesquisa nessa área.

Pela contextualização apresentada, a avaliação experimental de técnicas de IA, em especial de redes neurais artificiais recorrentes, aplicadas em problemas de identificação de sistemas representa um tema de pesquisa relevante. O emprego da computação de reservatório para tarefas de identificação de sistemas, em especial, se mostra interessante e a comparação experimental desse paradigma com as arquiteturas de RNA recorrentes derivadas do modelo MLP

consiste em uma atividade de interesse para auxiliar na definição do tipo de abordagem que pode ser empregado com maior probabilidade de sucesso em cada aplicação. Tal comparação possibilita um estudo aprofundado da capacidade de cada modelo, provendo meios para comparações não apenas da qualidade dos modelos gerados, mas também dos aspectos inerentes ao projeto de cada modelo.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo deste trabalho é avaliar experimentalmente a utilização de diferentes modelos e arquiteturas de redes neurais artificiais recorrentes no contexto de identificação de sistemas dinâmicos não lineares, através do desenvolvimento de um estudo de caso consistindo na identificação de um processo real, multivariável e não linear.

1.1.2 Objetivos Específicos

As atividades associadas ao objetivo geral do trabalho foram divididas em objetivos específicos, consistindo em:

- revisar a literatura para selecionar arquiteturas de redes neurais artificiais recorrentes capazes de identificar sistemas dinâmicos não lineares;
- modelar um processo não linear real utilizando as arquiteturas de redes neurais artificiais apresentadas;
- comparar o desempenho dos modelos desenvolvidos em diferentes tarefas de identificação de sistemas.

1.2 ESTRUTURA DO DOCUMENTO

O capítulo 2 apresenta uma revisão da literatura. Ele é dividido em duas seções, a primeira explorando conceitos de identificação de sistemas dinâmicos aplicados a RNAs, enquanto na segunda são apresentadas as arquiteturas de RNAs recorrentes que são avaliadas no decorrer do trabalho, sendo apresentado um detalhamento do equacionamento dos modelos e dos métodos de treinamento para as redes NARX, RMLP e ESN.

No capítulo 3 é apresentada a metodologia para identificação de sistemas dinâmicos não lineares empregada neste trabalho. Nesse capítulo é detalhado o estudo de caso desenvolvido, sendo apresentado o processo que é identificado usando a metodologia apresentada. É apresentado o desenvolvimento do primeiro passo da metodologia, o qual engloba a realização do ensaio na bancada, almejando a obtenção da série de dados que é utilizada na identificação do processo, e a análise da mesma, a qual é usada na definição dos conjuntos que são empregados no desenvolvimento e análise dos modelos.

No capítulo 4 é apresentado o desenvolvimento dos modelos para o processo do estudo de caso, empregando as RNAs apresentadas na revisão da literatura, e a análise do desempenho das mesmas. O capítulo é dividido em quatro seções, com a primeira apresentando o desenvolvimento do modelo empregando uma rede NARX, a segunda através de uma rede RMLP, a terceira apresentando o desenvolvimento de três modelos utilizando estruturas variadas de ESN, enquanto a quarta seção realizando uma comparação dos resultados de todos os modelos.

Por fim, no capítulo 5, são apresentadas as considerações finais do trabalho, compostas pelas principais conclusões obtidas com o desenvolvimento do trabalho e por propostas para trabalhos futuros.

2 REVISÃO DA LITERATURA

A revisão da literatura é dividida em duas grandes seções, as quais abordam os temas fundamentais a este trabalho, a identificação de sistemas dinâmicos não lineares e as redes neurais artificiais.

A identificação de sistemas dinâmicos não lineares é apresentada na seção 2.1, sendo detalhados os modelos com dinâmica interna e externa, nas subseções 2.1.2 e 2.1.1 respectivamente. Na subseção 2.1.3, são apresentadas alternativas para o aprimoramento da complexidade de modelos e, na seção 2.1.4, são detalhados aspectos acerca da excitação de processos não lineares.

Redes neurais artificiais são apresentadas na seção 2.2, com um enfoque nos modelos capazes de resolver problemas dinâmicos. É explorado, na subseção 2.2.1, o paradigma de redes recorrentes derivadas do modelo MLP, as quais são alternativas clássicas para a identificação de sistemas utilizando redes neurais artificiais. Um segundo paradigma para a resolução de problemas dinâmicos usando redes neurais, o da computação de reservatório, é apresentado na subseção 2.2.2. Nessa seção é apresentada a principal arquitetura de computação de reservatório, a rede ESN, sendo detalhados os aspectos construtivos do modelo, bem como seu método de treinamento.

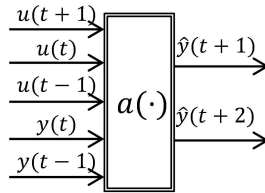
2.1 IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS NÃO LINEARES

Ljung (1999, p. 1) definiu: “a identificação de sistemas lida com o problema da construção de modelos matemáticos de sistemas dinâmicos, baseados em dados observados do sistema”. Através de análise experimental, baseada em medidas ou observações do sistema, são obtidos modelos que representam a sua dinâmica, podendo ser utilizados para diversas finalidades, destacando-se predição e simulação. Outros exemplos típicos de aplicações são controle, análise, otimização e detecção de falhas.

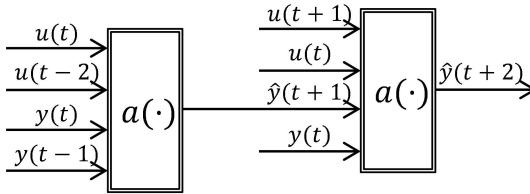
Na tarefa de predição, para sistemas dinâmicos de tempo discreto, as entradas do sistema, $u(t - j)$, $j \in \mathbb{Z}^+$, e suas saídas, $y(t - j)$, são usadas para a predição da saída do processo, $y(t+1)$, no próximo período de amostragem. Neste trabalho é usada a notação para a predição de um instante à frente realizada no instante atual como $\hat{y}(t+1)$. Tipicamente predições são representadas por $\hat{y}(t+j|t+k)$, indicando a predição para o instante $t+j$ realizada no instante $t+k$, porém neste trabalho optou-se por simplificar a

notação e omitir o instante em que é realizada a predição quando for o instante atual, t . No caso mais simples, apenas um instante à frente é predito. Se mais de um instante à frente é predito, a tarefa é denominada predição N instantes à frente e o número de instantes preditos, N , é denominado horizonte de predição. São apresentadas na figura 2.1 duas alternativas para a tarefa de predição de vários instantes à frente (NELLES, 2001).

Figura 2.1: Exemplo da realização de um preditor de duas amostras à frente.



(a) Preditor estendido com entrada adicional



(b) Preditor de apenas um instante replicado

Fonte: adaptado de Nelles (2001).

A primeira abordagem consiste na construção de um modelo que prediga l instantes à frente, como exemplificado na figura 2.1a. Para isso, o modelo necessita de um conjunto maior de entradas, proporcional a N , já que necessitará de entradas passadas e futuras, além das saídas passadas. A desvantagem é que essa abordagem aumenta a dimensão da entrada do modelo e também o tamanho da camada de saída, fazendo com que o modelo necessite de uma complexidade maior. A segunda abordagem é a construção de um modelo de predição para um instante à frente. Após a primeira predição, se t for substituído por $t + 1$ e for empregado $\hat{y}(t + 1)$

da predição anterior, pode-se prever o próximo instante, como mostrado na figura 2.1b. Pode-se repetir o procedimento N vezes e prever N instantes à frente. A desvantagem dessa abordagem é que ela se baseia em previsões anteriores e, por isso, os erros de previsão se acumulam (NELLES, 2001).

Quando o horizonte de previsão tende ao infinito ($N \rightarrow \infty$), a tarefa é equivalente à simulação. Tal tarefa contrasta com a de previsão pelo fato de apenas as variáveis de entrada estarem disponíveis. Dessa forma, com um modelo de simulação é possível realizar experimentos sem a utilização do processo real. Tipicamente, a tarefa de simulação é mais frequente que a de previsão, sendo utilizada principalmente para auxiliar o projeto de controladores e a otimização de malhas de controle. A diferença fundamental entre previsão e simulação está na possibilidade de utilização das saídas passadas medidas, o que acontece apenas em tarefas de previsão, possibilitando a correção periódica do erro acumulado durante a execução do modelo (LJUNG, 1999; NELLES, 2001).

Diversos métodos de identificação são descritos na literatura. Esses métodos podem ser sumarizados pelo tipo de modelo resultante, podendo ser paramétrico ou não-paramétrico. Um modelo paramétrico é caracterizado quando é assumida uma estrutura para o modelo a ser identificado. Para as técnicas baseadas em modelos não-paramétricos, por outro lado, não é necessária qualquer suposição *a priori* acerca da estrutura do modelo ou de possíveis não linearidades. (LEVINE, 2010) Isermann e Münchhof (2011, p. 13) definiram os modelos paramétricos como: “equações que explicitamente contenham os parâmetros do processo”. Por sua vez, modelos não paramétricos foram definidos por Isermann e Münchhof (2011, p. 13) como: “modelos que proporcionam uma relação entre certa entrada e sua correspondente resposta por meio de tabelas ou amostras de curvas características”.

As redes neurais artificiais fazem parte da subclasse dos modelos paramétricos. Apesar de a aplicação de RNA exigir a definição de uma estrutura para o modelo neural, tal técnica é capaz de modelar processos a partir de pouco ou nenhum conhecimento *a priori* sobre a física que rege o processo. A grande desvantagem do seu uso é o fato de que, na maioria das arquiteturas de redes neurais, os parâmetros das redes não podem ser interpretados e nenhum significado físico dos resultados do processo de identificação pode ser extraído (ISERMANN; MÜNCHOF, 2011; LEVINE, 2010).

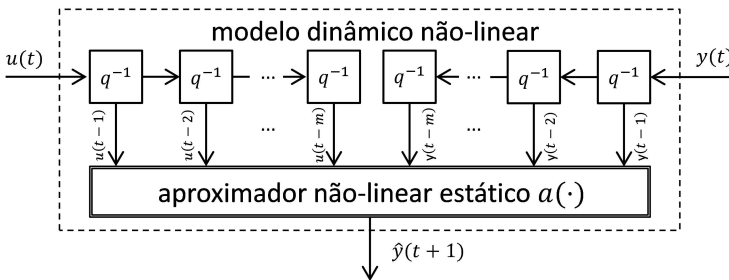
De forma geral, os modelos entrada-saída não lineares são divididos em duas classes, dinâmica externa e dinâmica interna, as

quais são detalhadas nas seções 2.1.1 e 2.1.2.

2.1.1 Dinâmica externa

A estratégia da dinâmica externa é a mais frequentemente aplicada a sistemas dinâmicos não lineares. O termo, dinâmica externa, decorre do fato de que sistemas dinâmicos não lineares podem ser claramente separados em duas partes: um modelo não linear estático e um banco de filtros dinâmicos externos, como mostrado na figura 2.2. Tipicamente, o modelo de filtro escolhido é um atraso simples q^{-1} no tempo, formando uma linha de atrasos. Em princípio, qualquer modelo de arquitetura pode ser escolhido para o aproximador $a(\cdot)$, por exemplo um sistema não-linear estático, que pode gerar um sistema Hammerstein ou Wiener, ou uma rede neural estática, que gera um modelo neural dinâmico.

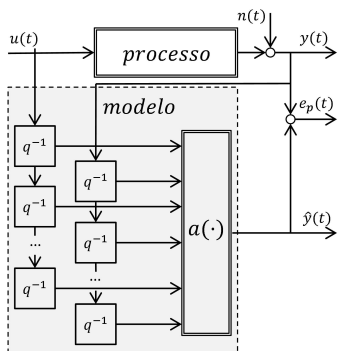
Figura 2.2: Estratégia da dinâmica externa.



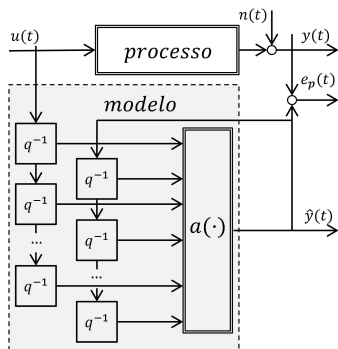
Fonte: adaptado de Nelles (2001).

Como mostrado na seção 2.1, os modelos dinâmicos não lineares podem ser utilizados em duas configurações: para predição e para simulação. Essas configurações são tradicionalmente denominadas como: modelo série-paralelo, quando é realizada predição, e modelo paralelo, quando é realizada simulação. As duas configurações são mostradas na figura 2.3. No modelo série-paralelo, a informação é puramente alimentada para frente, já que o modelo tem acesso à medição real do sistema. No caso do modelo paralelo, a alimentação acontece de forma recorrente, pois o modelo não possui informação da saída real do processo, que geralmente não está operando junto com a simulação.

Figura 2.3: Modelos nas configurações série-paralelo, usado para predição, e paralelo, usado para simulação



(a) Modelo série-paralelo



(b) Modelo paralelo

Fonte: adaptado de Nelles (2001).

Todos os modelos dinâmicos entrada-saída não lineares podem ser escritos da seguinte forma:

$$\hat{y}(t) = a(\varphi(t)), \quad (2.1)$$

sendo que o vetor de regressão $\varphi(t)$ pode conter entradas passadas e possivelmente entradas futuras, saídas passadas do processo ou do modelo e erros passados de predição. Os modelos que apresen-

tam realimentação da saída são os mais comuns, com destaque para: modelo não linear autorregressivo com entrada exógena (NARX, do inglês *Nonlinear AutoRegressive model with eXogenous inputs*); modelo não linear autorregressivo de média móvel com entrada exógena (NARMAX, do inglês *Nonlinear AutoRegressive Moving Average model with eXogenous inputs*); e modelo não linear do erro de saída (NOE, do inglês, *Nonlinear Output Error*), que possuem os seguintes vetores de regressão (considerando $e(t) = y(t) - \hat{y}(t)$) (NELLES, 2001):

$$\begin{aligned} \text{NARX :} \quad & \varphi(t) = [u(t-1), \dots, u(t-m), y(t-1), \\ & \quad \quad \quad \dots, y(t-m)]^T \\ \text{NARMAX :} \quad & \varphi(t) = [u(t-1), \dots, u(t-m), y(t-1), \\ & \quad \quad \quad \dots, y(t-m), e(t-1), \dots, e(t-m)]^T \\ \text{NOE :} \quad & \varphi(t) = [u(t-1), \dots, u(t-m), \hat{y}(t-1), \\ & \quad \quad \quad \dots, \hat{y}(t-m)]^T \end{aligned}$$

O modelo NARX pode ser identificado na configuração série-paralela, quando a saída real é conhecida. Por sua vez, o modelo NOE deve ser identificado na configuração paralela, por utilizar explicitamente a saída predita em seu vetor de regressão. O modelo NARMAX necessita tanto da saída do processo quanto da do modelo, e o uso explícito de um modelo de perturbação faz com que o aproximador $a(\cdot)$ necessite de entradas adicionais, o que aumenta a dimensão do problema, fazendo essa solução ser menos usada que as duas anteriores (NELLES, 2001).

Modelos sem a realimentação da saída também existem, mas, por possuírem apenas entradas passadas ou filtradas, o número de regressores em $\varphi(t)$ acaba sendo significativamente maior. Isso consiste em uma grande desvantagem em modelos que são especialmente sensíveis à dimensionalidade do problema, caso de algumas arquiteturas de redes neurais (NELLES, 2001).

A classe de modelos com dinâmica externa, apesar de poder descrever uma grande classe de problemas não lineares, não é tão geral quanto os modelos com dinâmica interna. As limitações aparecem em problemas nas quais o processo apresenta mais de uma não linearidade, estados internos não-mensuráveis com uma grande não linearidade ou em casos nos quais as não linearidades não são inversíveis (NELLES, 2001).

2.1.2 Dinâmica interna

Modelos não lineares com dinâmica interna são baseados nos modelos estáticos com memória interna. Essa classe de modelos pode ser descrita através da seguinte representação no espaço de estados:

$$\hat{x}(t + 1) = q(\hat{x}(t), u(t)) \quad (2.2)$$

$$\hat{y}(t) = s(\hat{x}(t)), \quad (2.3)$$

em que as variáveis de estados, $\hat{x}(t)$, descrevem os estados internos do modelo, estados esses que normalmente não apresentam relação direta com os estados físicos do sistema. O número de estados internos do sistema está ligado tipicamente à estrutura do modelo, podendo ser escolhidos desde valores próximos à ordem esperada do processo, até valores maiores que a ordem real do processo (NELLES, 2001).

Em contraste com os modelos de dinâmica externa, não é necessário o uso de entradas ou saídas passadas. Isso faz com que os modelos com dinâmica interna apresentem uma redução na dimensionalidade do espaço de entrada. Outra diferença é que, devido à inexistência de realimentação externa, apenas o modelo paralelo pode ser aplicado. Conseqüentemente esse modelo não é particularmente adequado à tarefa de predição de um ou mais instantes à frente.

2.1.3 Aprimoramento da complexidade do modelo

A complexidade de um modelo, apesar de ser um parâmetro de difícil definição, pode ser associada ao número de parâmetros que o modelo possui. Quanto maior o número de parâmetros, mais complexo o modelo se torna. Conseqüentemente, se for possível retirar parâmetros do modelo, ele se tornará mais simples (BISHOP, 2006).

A complexidade do modelo é um parâmetro importante, pois modelos com complexidade insuficiente são incapazes de resolver problemas complexos. Por outro lado, modelos com alta complexidade podem resolver tarefas complexas, mas quando a complexidade é mais alta que o necessário para resolver a tarefa pode haver superajuste do modelo. Por conseqüência, espera-se que exista uma complexidade ótima para o modelo, a qual dependerá de vários fa-

tores, em especial dos dados disponíveis e da natureza específica do modelo (GOODFELLOW et al., 2016; NELLES, 2001).

A importância da otimização da complexidade do modelo é, usualmente, ilustrada pelo dilema da relação entre bias e variância. O conceito explorado por esse dilema leva em conta a decomposição do erro de modelagem em dois termos, denominados erro de bias e erro de variância (GOODFELLOW et al., 2016).

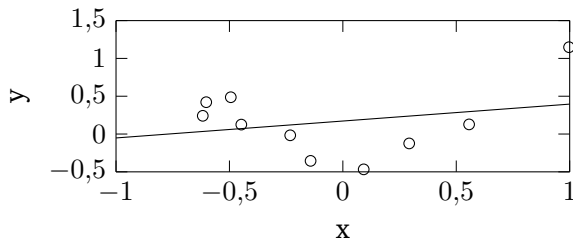
O erro de bias pode ser entendido intuitivamente como a parte do erro de modelagem decorrente da restrição na flexibilidade do modelo. Em termos gerais, os processos reais são mais complexos que os modelos tipicamente utilizados, fazendo com que os modelos não sejam capazes de modelar o processo exatamente. Mesmo que os parâmetros estimados para o modelo sejam ótimos, existirá um erro de bias, decorrente da inflexibilidade do modelo. O erro de bias, tipicamente, diminui com o aumento da complexidade do modelo (NELLES, 2001).

Por sua vez, o erro de variância é associado à parte do erro de modelagem decorrente do desvio entre os parâmetros estimados e o valor ótimo dos mesmos. Levando em conta que em aplicações reais os parâmetros do modelo são estimados a partir de dados finitos e ruidosos, esses parâmetros geralmente apresentam um desvio em relação ao seu valor ótimo. O erro introduzido por esse desvio é o erro de variância. Essa parcela de erro, tipicamente, aumenta linearmente com a complexidade do modelo (NELLES, 2001).

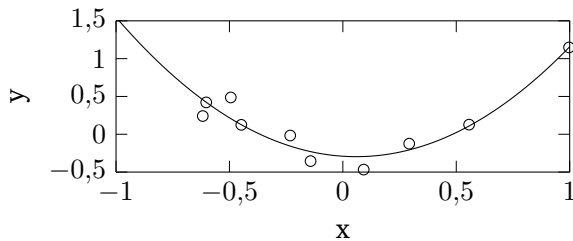
Esse conceito pode ser visualizado através do exemplo da figura 2.4. Um conjunto de dados gerado através de uma função quadrática com ruído foi ajustada usando modelos polinomiais de ordens diferentes. Na figura 2.4a é mostrado o ajuste por um modelo com complexidade inferior à ótima, caso em que ocorre subajuste. Esse modelo apresenta alto erro de bias, por ter uma complexidade limitada pelo problema. Por sua vez, na figura 2.4b é mostrado o ajuste por um modelo com complexidade ótima. Por fim, na figura 2.4c é utilizado um modelo com complexidade maior que a ótima, resultando em superajuste do modelo. Nesse caso, o erro de bias é menor comparado ao dos demais modelos, porém o erro de variância é alto. Para uma realização de ruído diferente, o modelo apresentado na figura 2.4c apresentará resultados ruins (GOODFELLOW et al., 2016; BISHOP, 2006).

Existe então um compromisso na escolha da complexidade do modelo. Se ela for aumentada, será induzido um menor erro de bias, já se for reduzida, induzirá um menor erro de variância. A dificuldade na obtenção desse compromisso, que leva à complexidade

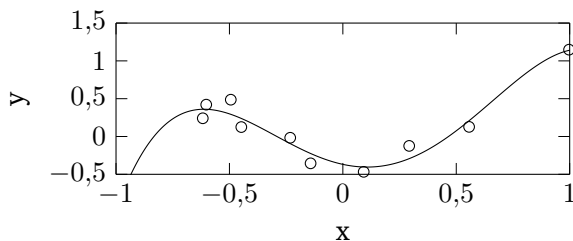
Figura 2.4: Ajuste de três modelos polinomiais para dados de exemplo.



(a) Ajuste com modelo polinomial de 1ª ordem



(b) Ajuste com modelo polinomial de 2ª ordem



(c) Ajuste com modelo polinomial de 4ª ordem

Fonte: autor.

ótima do modelo, reside no fato de as parcelas do erro serem desconhecidas. Uma forma de estimar os erros de bias e variância consiste no treinamento de diversos modelos com complexidades diferentes. O erro de bias pode ser identificado através do erro no conjunto de treinamento, no qual a parcela decorrente da variância não pode ser detectada. O erro de variância, por ser causado pelas incertezas dos parâmetros do modelo, é identificado apenas em um conjunto

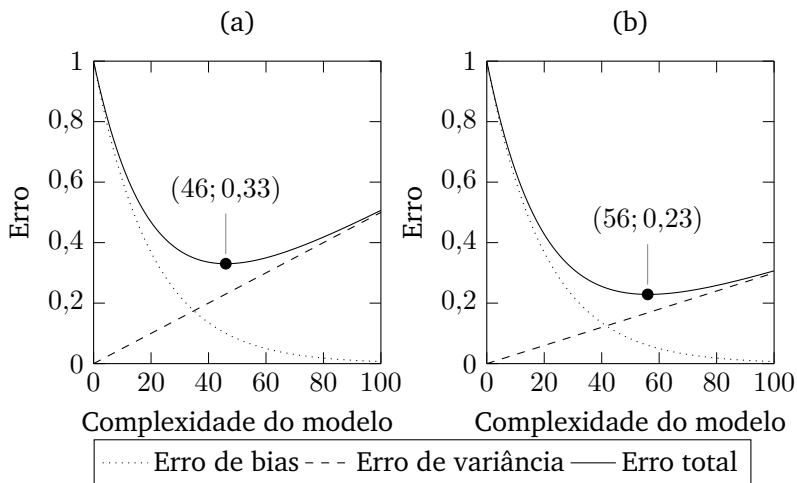
de validação, que apresenta uma realização de ruído diferente da realização apresentada ao modelo durante a etapa de treinamento. Essa estratégia é efetiva apenas para modelos que requerem pouco esforço computacional (NELLES, 2001).

O erro no conjunto de treinamento, aproximadamente igual ao erro de bias, é o erro de generalização. Ele diminui à medida que a complexidade do modelo aumenta, ao mesmo tempo que o erro no conjunto de testes, aproximadamente igual ao erro de bias somado ao de variância, diminui até a complexidade do modelo atingir a complexidade ótima, a partir da qual o erro de variância aumenta. Isso pode ser verificado na figura 2.5, que apresenta dois exemplos do comportamento das parcelas do erro de modelagem. O primeiro exemplo da figura, mostrado em 2.5(a), ilustra um caso em que a variância é alta, fazendo com que a complexidade ótima do modelo seja 46. Já no caso ilustrado na figura 2.5(b), é exemplificado o caso em que a variância do conjunto de dados é reduzida, o que pode decorrer de menos ruído nos dados ou uma quantidade maior dos mesmos. A variância reduzida propicia a utilização de modelos com complexidades maiores, fazendo com que a complexidade ótima seja 56, com erro menor que o obtido no caso (a) (NELLES, 2001).

No contexto de redes neurais artificiais, a otimização da estrutura do modelo é realizada, usualmente, de forma implícita através de regularização. Regularização pode ser definida como qualquer modificação realizada no algoritmo de treinamento com a intenção de reduzir o erro de generalização sem modificar o erro de treinamento. Técnicas de regularização permitem um controle da complexidade do modelo, sem modificar a quantidade real de parâmetros do modelo. Na prática, o modelo se torna menos flexível que a quantidade de parâmetros aparenta, característica alcançada principalmente de duas formas (GOODFELLOW et al., 2016; NELLES, 2001):

- Regularização por penalidades: essa técnica utiliza penalização para comportamentos não suaves, forçando então o modelo a ser suave. Isso é alcançado ao otimizar uma função custo composta de um critério relativo ao desempenho da rede e um critério de penalidade para a não suavidade. Esse tipo de regularização é implementado em técnicas como a regularização de Thikonov (*ridge regression*), aplicada a problemas de programação linear, e decaimento de pesos (do inglês, *weight decay*), que expande o conceito para problemas

Figura 2.5: Representação do dilema da relação entre bias e variância. O erro de modelagem é decomposto em uma parcela de erro de bias e uma parcela de erro de variância. (a) Exemplo com variância alta. (b) Exemplo com variância reduzida em comparação a (a).



Fonte: adaptado de Nelles (2001).

de otimização não linear.

- Regularização por parada precoce: a regularização por parada precoce (do inglês, *early stopping*) é aplicada em métodos de otimização iterativos. Nessa abordagem, o modelo é treinado iterativamente e o desempenho é verificado em um conjunto de validação, independente do conjunto de treinamento. O treinamento é encerrado quando o desempenho para o conjunto de validação atinge um mínimo local, normalmente identificado pela piora sucessiva do desempenho.

2.1.4 Excitação de processos não lineares

A série de dados usada para identificar o sistema tem grande impacto na qualidade dos resultados. Um dos aspectos importantes é a escolha do sinal de excitação a ser utilizado, pois a qualidade alcançável do modelo será limitada pela qualidade dos dados obtidos pelo ensaio de identificação, sendo altamente influenciada pela escolha do sinal de excitação. A escolha do sinal de excitação

é mais crítica para sistemas dinâmicos não lineares, pois eles são geralmente mais complexos que os lineares, necessitando consideravelmente de mais informações para que o processo possa ser adequadamente descrito (NELLES, 2001).

Como sinal de excitação, é possível utilizar sinais que ocorrem durante a operação normal do processo, ou então sinais projetados específicos para a tarefa de identificação. Os sinais encontrados na operação convencional do processo são adequados, apenas, se excitarem as dinâmicas do processo de forma suficiente, o que geralmente não se verifica. Devido a isso, sempre que possível, é recomendado o uso de sinais artificialmente criados para essa tarefa (ISERMANN; MÜNCHOF, 2011).

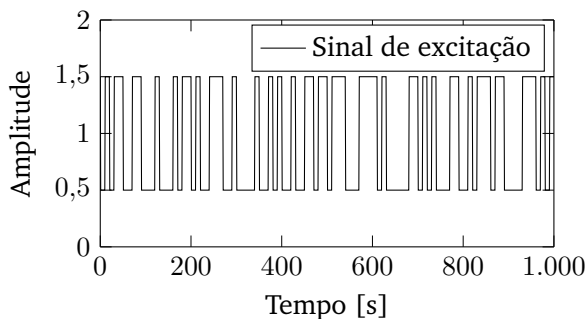
Sinais de excitação são adequados, tipicamente, quando excitam as dinâmicas de interesse do processo de forma contínua e intensa. Em sistemas lineares, a amplitude do sinal deve ser a maior possível, respeitando as limitações impostas pelos atuadores e pelos limites de operação do processo. Já para sistemas não lineares, é importante que o sistema seja levado a operar nas diferentes regiões de operação, com amplitudes do sinal condizentes com o ponto de operação. A presença de variações abruptas excita as altas frequências do processo, enquanto pulsos pequenos nesse sinal resultam em excitação de médias e altas frequências. Por sua vez, pulsos longos excitam as baixas frequências (ISERMANN; MÜNCHOF, 2011).

Para sistemas lineares, um sinal de excitação geralmente aplicado é o sinal binário pseudo aleatório (PRBS, do inglês, *Pseudo-Random Binary Signal*), cujo espectro pode ser escolhido de acordo com a dinâmica do processo. Para sistemas não lineares, além do espectro frequencial, a amplitude deve ser escolhida cuidadosamente, de modo a cobrir todas as condições de operação de interesse. Para isso, uma alternativa é utilizar um sinal PRBS em que cada degrau apresenta uma amplitude diferente, o que é conhecido como PRBS modulado em amplitude (APRBS, do inglês *Amplitude-modulated PRBS*). Na figura 2.6 são mostrados um sinal PRBS convencional e um APRBS (NELLES, 2001).

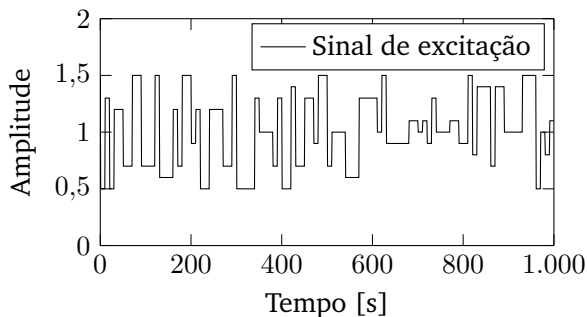
2.2 REDES NEURAIS ARTIFICIAIS

A inteligência artificial (IA) é uma área de estudo vasta, apresentando historicamente diversas linhas de pesquisa, com destaque para a linha simbólica e a conexionista. A linha conexionista foi definida por Bittencourt (2006, p. 309) como:

Figura 2.6: Exemplo de sinais de excitação PRBS e APRBS.



(a) Sinal PRBS



(b) Sinal APRBS

Fonte: autor.

O conexionismo é uma das duas grandes linhas de pesquisa da IA e tem por objetivo investigar a possibilidade de simulação de comportamentos inteligentes através de modelos baseados na estrutura e funcionamento do cérebro humano.

A seguinte definição para redes neurais artificiais (RNAs) foi oferecida por Haykin (2001, p. 28):

Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso.

As RNAs são modelos que buscam imitar alguns dos métodos básicos de processamento de informação encontrados no cérebro humano. As redes neurais artificiais apresentam elementos em comum com sua correspondente biológica como: suas unidades básicas de processamento, conhecidas como neurônios; a arquitetura baseada em processamento distribuído; e os mecanismos de aprendizagem (SAMARASINGHE, 2016).

O neurônio artificial, usado nas RNAs, é um modelo matemático simplificado que busca emular o funcionamento do neurônio biológico. A primeira formulação, apresentada por McCulloch e Pitts (1943) no modelo *perceptron*, apresenta uma estrutura que recebe diversas entradas e retorna uma ou mais saídas, que consistem em uma função não linear da soma das entradas do neurônio (HAYKIN, 2009).

O conhecimento é adquirido a partir do ambiente através de um processo de aprendizagem. Ele acontece através de exposição da rede a exemplos, de forma repetida, até que a rede apresente a saída desejada. Esse processo consiste na modificação das forças de conexão entre os neurônios (pesos sinápticos) através de um algoritmo de aprendizagem, de forma que o conhecimento adquirido seja armazenado. (SAMARASINGHE, 2016; HAYKIN, 2009).

Existem na literatura diversos modelos de redes neurais artificiais. Esses modelos apresentam, em geral, uma das seguintes estruturas: arquitetura de uma camada; arquitetura de múltiplas camadas; ou arquitetura recorrente. Enquanto as duas primeiras caracterizam-se pelo fluxo unidirecional adiante (*feedforward networks*), com a informação fluindo da entrada para a saída, a terceira apresenta, ao menos, um caminho cíclico conectando seus neurônios (HAYKIN, 2001).

As arquiteturas alimentadas adiante englobam os modelos de redes mais tradicionais e utilizados, as redes *perceptron* de múltiplas camadas (MLP, do inglês *MultiLayer Perceptron*) e as redes com função de ativação de base radial (RBF, do inglês *Radial Basis Function*). Elas diferem entre si pelo modelo de neurônio artificial que utilizam, mas são empregadas no mesmo contexto, principalmente para a resolução de problemas de estáticos de aprendizado supervisionado (HAYKIN, 2001).

Os modelos tradicionais de redes neurais foram desenvolvidos com capacidade para processar apenas padrões de entrada estáticos. Para lidar com estímulos variantes no tempo e padrões dinâmicos, esses modelos podem incorporar algumas estratégias capazes de contornar as limitações inerentes a eles. Uma dessas estratégias

é a utilização do tempo como uma dimensão espacial adicional, dividindo os estímulos da rede em janelas temporais. Essa estratégia resulta em um alto número de parâmetros de entrada, pois, por exemplo, para uma janela temporal de 5 amostras, a dimensão da camada de entrada será $5n_i$, sendo n_i o número de padrões de entrada. Uma segunda estratégia consiste na utilização de conexões recorrentes, conceito explorado pelas redes de Elman e Jordan. As conexões recorrentes proveem mecanismos de contexto temporal, permitindo que estados passados da rede sejam usados em conjunto com os estímulos de entrada (ANTONELO, 2012).

As redes neurais recorrentes se comportam como um sistema dinâmico. Em geral, as arquiteturas recorrentes derivam de arquiteturas estáticas, em especial da MLP. Destacam-se arquiteturas como a rede MLP com realimentação entre saída e entrada, comumente denominada de rede NARX, e o modelo MLP recorrente (RMLP, do inglês *Recurrent MLP*), que consiste em um modelo recorrente geral em que cada camada apresenta realimentação apenas com sua própria entrada (TUTSCHKU, 1995).

São encontrados também, na literatura, modelos puramente recorrentes, ou seja, que não são derivações de arquiteturas estáticas. Esses modelos apresentam capacidade intrínseca para lidar com problemas dinâmicos, pois incorporam em sua arquitetura estruturas recorrentes. Uma classe de arquiteturas recorrentes, em específico, utiliza o paradigma de aprendizado denominada como computação de reservatório (do inglês, *reservoir computing*), o qual caracteriza-se pela utilização de um reservatório, composto por neurônios interconectados de forma recorrente, para gerar um modelo dinâmico. As duas principais arquiteturas que se utilizam do reservatório são a rede de estado de eco (ESN, do inglês *Echo State Network*) e a máquina de estado líquido (LSM, do inglês *Liquid State Machine*) (JAEGER; HAAS, 2004; MAASS, 2011).

Na seção 2.2.1 é apresentada em detalhes a classe de redes MLP, com destaque para suas variantes recorrentes. Por sua vez, na seção 2.2.2 é abordado o paradigma da computação de reservatório, sendo explorada em detalhes a arquitetura ESN.

2.2.1 Redes *perceptron* de múltiplas camadas dinâmicas

A rede MLP é a arquitetura de RNA mais conhecida e amplamente utilizada, sendo apresentada em algumas publicações, inclusive, como sinônimo de RNA. Esse modelo usa o aprendizado supervisionado, ou seja, aprende através de exemplos rotulados por

um supervisor, e pode desempenhar tanto tarefas de classificação quanto de regressão¹ (NELLES, 2001).

Para uma rede MLP ser considerada recorrente é necessário haver, ao menos, um caminho cíclico de conexões sinápticas (JAEGER, 2002). O modelo MLP é uma RNA estática, capaz de implementar qualquer função não linear, por ser um aproximador universal (HORNIK et al., 1989). Por sua vez, as redes neurais recorrentes são capazes de implementar sistemas dinâmicos não lineares. A ideia utilizada pelas arquiteturas MLP recorrentes para o aprendizado de problemas dinâmicos é a mesma mostrada nas seções 2.1.1 e 2.1.2. (HAYKIN, 2001).

Nas subseções a seguir são apresentados aspectos básicos do modelo MLP e as principais arquiteturas recorrentes encontradas na literatura. A subseção 2.2.1.1 detalha o modelo de neurônio usado na estrutura MLP, enquanto as seções 2.2.1.2, 2.2.1.3 e 2.2.1.4 abordam as arquiteturas recorrentes baseadas no modelo MLP mais usuais, as redes de Elman e Jordan e as redes NARX e RMLP, respectivamente. Na subseção 2.2.1.5 são discutidos os métodos para o treinamento de estruturas MLP recorrentes e suas peculiaridades.

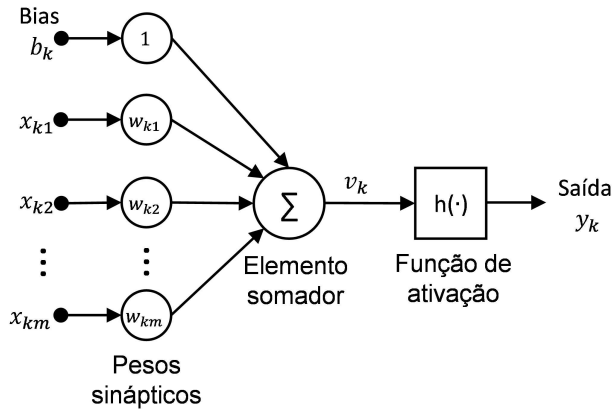
2.2.1.1 Neurônio MLP

Os neurônios artificiais, elementos distribuídos de processamento das RNAs, apresentam uma estrutura básica e têm sua morfologia e funcionamento inspirados no neurônio biológico. Eles apresentam três elementos básicos: um conjunto de pesos sinápticos, um somador e uma função de ativação. O modelo geral do neurônio artificial é mostrado na figura 2.7 (HAYKIN, 2001).

Os pesos sinápticos armazenam grande parte do conhecimento do neurônio. Esses pesos sinápticos são responsáveis por ponderar as informações que entram no neurônio, que serão somadas no elemento somador. O parâmetro bias, aplicado externamente, é obtido durante a fase de aprendizado da rede. A função de ativação, $h(\cdot)$, mapeia a soma dos dados de entrada, ponderados pelos pesos

¹Classificação e regressão são duas das tarefas básicas do aprendizado de máquina. Enquanto na classificação o objetivo é rotular um novo exemplo como pertencente ou não a uma determinada classe, na regressão busca-se atribuir ao novo exemplo um valor contínuo, permitindo inferir uma relação entre uma variável dependente e as variáveis independentes (SMOLA, 2008).

Figura 2.7: Modelo geral de um neurônio artificial.



Fonte: adaptado de Haykin (2001).

sinápticos, para o valor de saída, determinando o grau de ativação do neurônio.

É possível descrever um neurônio k , matematicamente, pelas equações (2.4) e (2.5):

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (2.4)$$

$$y_k = h(v_k) \quad (2.5)$$

onde:

- $x_{k1}, x_{k2}, \dots, x_{km}$ são os sinais de entrada;
- $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ;
- b_k é o bias;
- v_k é a saída do elemento somador, contendo a soma das entradas ponderadas e o bias b_k ;
- $h(\cdot)$ é a função de ativação;
- y_k é o sinal de saída do neurônio.

Diversas funções podem ser escolhidas para a ativação do neurônio. A saturação é uma função clássica, presente na formulação original do *perceptron*. Outras funções comumente usadas são a logística e a tangente hiperbólica, as quais são do tipo sigmoide e deriváveis, característica importante para os algoritmos de treinamento, que de forma geral necessitam computar o gradiente do erro da rede (HAYKIN, 2001).

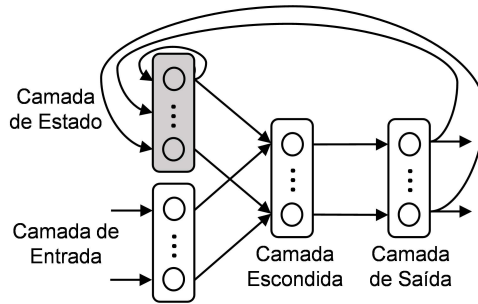
2.2.1.2 Redes de Jordan e Elman

Jordan (1986) definiu que se uma rede apresenta um ou mais ciclos em que é possível seguir um caminho entre uma unidade e ela mesma, então essa rede pode ser denominada recorrente. Nesse trabalho foi apresentada a primeira arquitetura recorrente, que ficou conhecida como rede de Jordan, mostrada na figura 2.8a. Essa rede apresenta laços de realimentação entre a saída da rede e a entrada, a qual é denominada camada de estado. Nessa arquitetura, a camada de estado apenas replica as saídas da rede e pode apresentar realimentações entre seus neurônios. De forma geral, os pesos de entrada da camada de estado não são treináveis, fazendo com que esses neurônios se comportem como uma entrada qualquer, não necessitando de métodos de treinamento alternativos aos tradicionalmente usados em redes MLP.

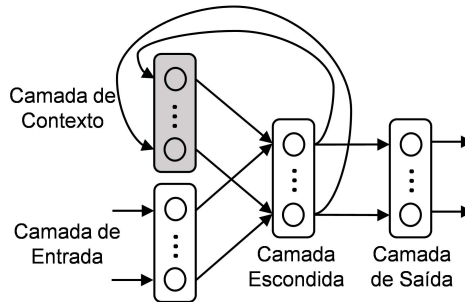
Outra arquitetura recorrente encontrada na literatura, denominada rede recorrente simples (SRN, do inglês *Simple Recurrent Network*) e comumente tratada como rede de Elman, foi apresentada por Elman (1990). Essa arquitetura expande a arquitetura MLP através da adição de uma camada de contexto, a qual consiste em uma realimentação entre a camada escondida e a entrada da rede. Essa camada de contexto faz com que rede de Elman seja capaz de lidar com problemas temporais. De forma semelhante à rede de Jordan, a camada de contexto apenas replica os valores da saída dos neurônios da camada escondida, não sendo possível treinar esses pesos (HAYKIN, 2009).

De maneira geral, as redes de Jordan e Elman podem ser utilizadas em problemas dinâmicos, sendo empregadas principalmente para o processamento de linguagem natural. Na prática, ambos os modelos são pouco aplicados, em detrimento de modelos mais gerais, que permitem o treinamento dos pesos das realimentações, como a rede neural NARX e a rede RMLP, modelos que serão apresentados nas próximas seções.

Figura 2.8: Arquiteturas recorrentes básicas (nem todas as conexões são mostradas). A rede de Jordan apresenta “entradas de estado”, enquanto a rede de Elman apresenta uma “camada de contexto”, que apresentam funcionamento semelhante.



(a) Rede de Jordan



(b) Rede de Elman

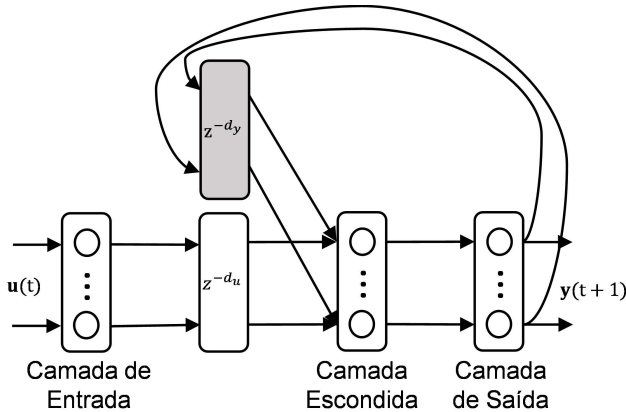
Fonte: autor.

2.2.1.3 Rede neural NARX

Além das redes de Jordan e Elman, outras arquiteturas recorrentes, originárias do modelo MLP, são apresentadas na literatura. As principais são a rede NARX e a rede RMLP, que diferem das anteriores pelo fato de apresentarem pesos treináveis nas realimentações. A primeira delas, a rede NARX, utiliza a realimentação dos neurônios de saída para a camada de entrada,

como mostrado na seção 2.1.1, gerando um modelo recorrente de entrada-saída, comumente denominado como modelo auto-regressivo não linear com entradas exógenas (NARX). Esse modelo apresenta sua entrada aplicada a uma linha de atrasos com tamanho d_u , sua saída realimentada para a camada de entrada, aplicada a uma linha de atrasos com tamanho d_y , e utiliza um *perceptron* de múltiplas camadas como bloco construtivo básico. Esse modelo é mostrado na figura 2.9 (HAYKIN, 2001).

Figura 2.9: Arquitetura da rede NARX (nem todas as conexões são mostradas).



Fonte: autor.

Essa arquitetura é caracterizada por incorporar partes do modelo *perceptron* de múltiplas camadas, associado com linhas de atrasos discretos (TDL, do inglês *Tap Delay Line*). Essas linhas de atrasos são conectadas a cada entrada, preservando a informação temporal contida nas entradas. Esse modelo incorpora, adicionalmente, a sua própria saída realimentada como uma entrada, associada também a uma linha de atrasos discretos. Além disso, o modelo conta com uma ou mais camadas escondidas, responsáveis pelo processamento da informação entre as camadas de entrada e saída da rede (HAYKIN, 2009).

O funcionamento da rede NARX é determinado por (2.6) e (2.7), equações que definem o funcionamento da camada 1, que contém as entradas da rede, e das camadas $i > 1$, respectivamente,

sendo que a última camada corresponde à camada de saída.

A equação para a camada 1 da rede neural NARX é definida em (2.6).

$$\mathbf{y}_1(t+1) = h(\mathbf{W}_0^1 \mathbf{u}_{du}(t) + \mathbf{W}_{out}^1 \mathbf{y}_{dy}(t) + \mathbf{W}_{bias}^1 \mathbf{1}), \quad (2.6)$$

\mathbf{W}_0^1 é a matriz de pesos da entrada para a camada 1, com dimensão $(N_u d_u) \times N_1$; \mathbf{W}_{out}^1 é a matriz de pesos da realimentação entre a saída e a entrada, com dimensão $(N_y d_y) \times N_1$; \mathbf{W}_{bias}^1 é a matriz diagonal de pesos para o bias da camada 1, com dimensão $N_1 \times N_1$; $\mathbf{u}_{du}(t)$ é um vetor formado pelas entradas do modelo no instante t até $t - d_u$; $\mathbf{y}_{dy}(t)$ é um vetor composto pela saída realimentada da rede nos instantes t até $t - d_y$; e $h(\cdot)$ é a função de ativação dos neurônios (NARENDRA; PARTHASARATHY, 1990; TUTSCHKU, 1995).

A equação para as demais camadas da rede neural NARX é apresentada em (2.7).

$$\mathbf{y}_i(t+1) = h(\mathbf{W}_{i-1}^i \mathbf{y}_{i-1}(t+1) + \mathbf{W}_{bias}^i \mathbf{1}). \quad (2.7)$$

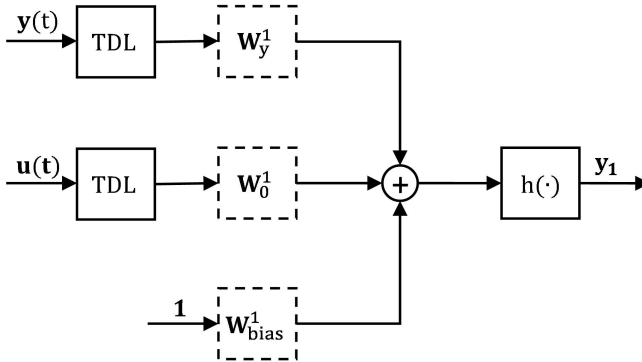
na qual \mathbf{W}_{i-1}^i é a matriz de pesos entre a camada $i - 1$ e i , com dimensão $N_{i-1} \times N_i$; \mathbf{W}_{bias}^i é a matriz diagonal de pesos para o bias da camada i , com dimensão $N_i \times N_i$; $\mathbf{y}_{i-1}(t+1)$ é o vetor de saídas da camada $i - 1$ do modelo; e $h(\cdot)$ é a função de ativação dos neurônios (NARENDRA; PARTHASARATHY, 1990; TUTSCHKU, 1995).

A representação em diagrama de blocos para a camadas de entrada da rede é apresentada na figura 2.10a. Por sua vez, o diagrama de blocos para as demais camadas é mostrado na figura 2.10b.

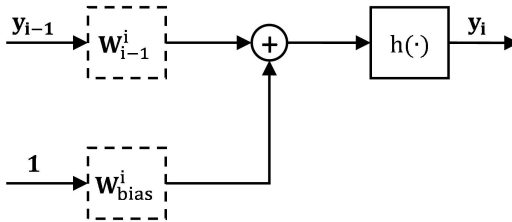
Uma rede NARX, em geral, apresenta mais de uma camada. O modelo mais simples, com apenas uma camada, apresenta a estrutura mostrada na figura 2.10a, com a saída \mathbf{y}_1 consistindo na saída do modelo $\mathbf{y}(t+1)$. Estruturas com mais de uma camada são construídas através da justaposição de uma camada 1, que contém a entrada, com as camadas genéricas $i > 1$, sendo que a saída da última camada representa a saída do modelo $\mathbf{y}(t+1)$.

O número de neurônios usado nas camadas de entrada da rede é dependente do número de entradas e saídas do modelo, e também do número de atrasos d_u e d_y , associados a $u(t)$ e $y(t)$, respectivamente. O tamanho dessa linha de atrasos é um dos parâmetros ajustados na etapa de definição da estrutura do modelo. O

Figura 2.10: Diagrama de blocos para as camadas da rede NARX. Os blocos tracejados representam matrizes treináveis do modelo.



(a) Camada 1, que incorpora a entrada da rede NARX.



(b) Camada $i > 1$ da rede NARX.

Fonte: autor.

número de neurônios da camada de saída é igual ao número de saídas do modelo. Os demais parâmetros ajustáveis são o número de camadas escondidas e o número de neurônios em cada uma dessas camadas (TUTSCHKU, 1995).

A rede neural NARX implementa um modelo construído através da técnica da realimentação externa. Como o apresentado na seção 2.1.1, modelos com recorrência externa são mais adequados ao emprego em tarefas de predição. A utilização em tarefas de simulação é possível, mas é esperado que modelos com realimentação interna apresentem resultados melhores que a rede NARX (NELLES, 2001).

2.2.1.4 Rede neural RMLP

A rede RMLP consiste em uma arquitetura de rede neural MLP com realimentação em todas as camadas do modelo. Essa arquitetura incorpora o comportamento dinâmico, em seu modelo, através da construção de um modelo com realimentação interna (NELLES, 2001).

Uma rede neural RMLP pode ser interpretada como uma sequência de sub-redes. Cada sub-rede consiste em um conjunto de camadas de neurônios, com alimentação adiante e realimentação apenas na última camada. Uma rede é composta por um conjunto de sub-redes, as quais se conectam apenas com a sub-rede adiante, sem conexões recorrentes (PUSKORIUS; FELDKAMP, 1994).

Essa representação generalizada para a rede RMLP é capaz de representar desde o modelo MLP totalmente alimentado adiante, até redes recorrentes totalmente conectadas. Em aspectos gerais, para ser utilizada na prática, a arquitetura precisa sofrer uma restrição no tamanho das sub-redes, as quais podem apresentar apenas uma camada. Dessa forma, cada sub-rede é equivalente a cada camada da rede RMLP (TUTSCHKU, 1995).

Cada camada i é totalmente conectada, estando ligada à camada subsequente com alimentação adiante por meio de pesos treináveis. Cada neurônio apresenta conexões recorrentes treináveis ligados à sua própria entrada e à entrada dos neurônios de sua própria camada (TUTSCHKU, 1995).

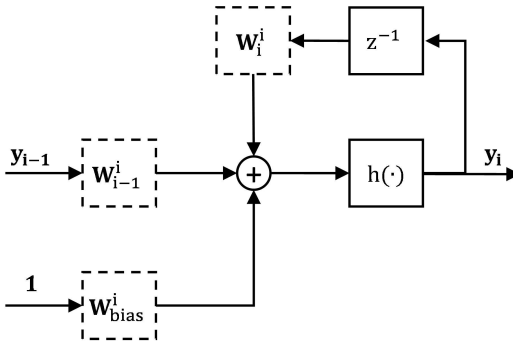
Em uma rede RMLP, a saída de cada camada, \mathbf{y}_i , pode ser descrita pela equação:

$$\mathbf{y}_i(t+1) = h(\mathbf{W}_{i-1}^i \mathbf{y}_{i-1}(t+1) + \mathbf{W}_i^i \mathbf{y}_i(t) + \mathbf{W}_{\text{bias}}^i \mathbf{1}), \quad (2.8)$$

no qual \mathbf{W}_{i-1}^i é a matriz de pesos que liga a camada $i-1$ com a camada i , com dimensão $N_i \times N_{i-1}$; \mathbf{W}_i^i é a matriz de pesos da realimentação da saída da camada i com sua própria entrada; $\mathbf{W}_{\text{bias}}^i$ é a matriz diagonal de pesos bias para a camada i , com dimensão $N_i \times N_i$; e $h(\cdot)$ é a função de ativação dos neurônios. Na figura 2.11 é mostrado o diagrama de blocos para uma camada da rede RMLP (PUSKORIUS; FELDKAMP, 1994).

A estrutura de uma rede RMLP é composta por três tipos de camadas com funções distintas. A camada de entrada coleta apenas as entradas da rede, tendo um neurônio para cada entrada do modelo. Ela não consiste em uma sub-rede, sendo geralmente denominada como camada 0. A camada de saída tem dimensão igual ao

Figura 2.11: Diagrama de blocos genérico para uma camada i da rede RMLP. Os blocos tracejados representam matrizes treináveis do modelo.



Fonte: autor.

número de saídas do modelo, apresentando a arquitetura da figura 2.11. O terceiro tipo consiste nas camadas escondidas, que ligam a camada de entrada à de saída. É usada, em geral, uma camada escondida, a qual pode ter uma quantidade de neurônios arbitrária. Aplicações com mais de uma camada escondida são menos comuns, devido ao alto custo computacional demandado pelos métodos de treinamento. A seleção do número de neurônios da camada escondida também sofre limitações devido ao custo computacional do treinamento, tornando difícil a obtenção de valores ótimos (PUSKORIUS; FELDKAMP, 1994).

O modelo RMLP consiste em um modelo com realimentação interna. Como mostrado na seção 2.1.2, modelos com realimentação interna adequam-se bem à tarefa de simulação, com um potencial grande para a obtenção de modelo com alta qualidade. Por sua vez, ele não pode ser usado na configuração série-paralelo, fazendo com esse modelo presente desvantagens quando utilizado em tarefas de predição.

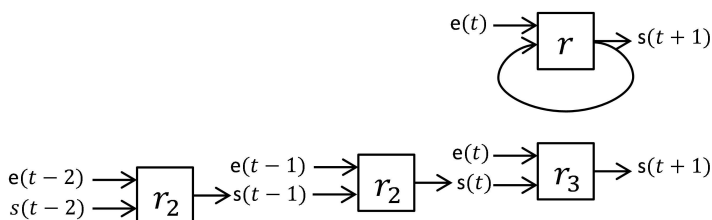
2.2.1.5 Treinamento de arquiteturas MLP recorrentes

De modo diferente do que ocorre com o modelo MLP alimentado adiante, que apresenta um algoritmo de treinamento

muito popular e amplamente utilizado, denominado algoritmo da retropropagação do erro (do inglês, *backpropagation*) (RUMELHART et al., 1986), para os modelos MLP com conexões recorrentes existem vários algoritmos de treinamento, cada um com suas particularidades e sem um deles apresentar clara vantagem sobre os outros (JAEGER, 2002; HAYKIN, 2001).

Um deles é algoritmo da retropropagação através do tempo (BPTT, do inglês *BackPropagation Through Time*), proposto por Werbos (1990), consistindo em uma extensão do algoritmo da retropropagação do erro. Ele opera com a premissa de que a operação temporal de uma rede recorrente pode ser desdobrada em uma rede alimentada adiante. Esse desdobramento no tempo acontece como mostrado na figura 2.12 (HAYKIN, 2001).

Figura 2.12: Desdobramento no tempo efetuado pelo BPTT



Fonte: autor.

Para uma rede neural r , com realimentação de sua saída com sua entrada, o algoritmo BPTT desdobra a rede k instantes no tempo, nesse exemplo $k = 3$, replicando a rede r em r_1 , r_2 e r_3 , todas com os pesos de r replicados. Após esse desdobramento, a rede pode ser treinada com o algoritmo da retropropagação do erro, sendo que ao final de cada época² os pesos de r_1 , r_2 e r_3 são igualados usando a média dos três. São usados porções das séries de dados, de forma a obter as sequências de e e s utilizadas em cada época.

²No treinamento por época, o algoritmo apresenta todos exemplos à rede, para então atualizar os pesos. Uma época consiste, então, na apresentação de todo o conjunto de entrada à rede (SAMARASINGHE, 2016).

Outro método da aprendizagem para arquiteturas MLP recorrentes é o aprendizado recorrente em tempo real (RTRL, do inglês *real time recurrent learning*), proposto por Williams e Zipser (1989), o qual ajusta os pesos de uma rede recorrente totalmente conectada, em tempo real. Diferente do algoritmo BPTT que desdobra a rede no tempo e calcula um gradiente aproximado, o RTRL calcula as derivadas dos estados e saídas com respeito a todos os pesos durante a etapa de avanço. Em essência, o algoritmo mede a sensibilidade do valor de saída de um neurônio k no instante de tempo t a uma pequena variação de um peso sináptico, levando em conta o efeito dessa mudança no peso sobre toda a trajetória desde um t_0 até t . É importante notar que até os pesos que não estão conectados ao neurônio k podem afetar sua saída, devido à natureza recorrente da rede. Esse algoritmo apresenta uma complexidade muito maior que o BPTT, demandando um poder computacional maior e, conseqüentemente, um tempo de treinamento superior para o treinamento da mesma rede. Ambos se baseiam no método do gradiente, pelo qual o valor instantâneo de uma função de custo é minimizado em relação aos pesos sinápticos da rede, e ambos apresentam uma convergência lenta (HAYKIN, 2001).

2.2.2 Computação de reservatório

Um paradigma alternativo ao dos modelos tradicionais de redes neurais é o da computação de reservatório, apresentado simultaneamente por Jaeger (2002) e Maass et al. (2002). Esse paradigma se baseia na utilização de um sistema dinâmico não linear com pesos fixos gerados aleatoriamente, por exemplo uma rede neural, para mapear as entradas em um espaço de alta dimensionalidade, na qual a tarefa de classificação ou de regressão é realizada eficientemente (LUKOŠEVIČIUS, 2012).

O conjunto de estados desse sistema dinâmico compõe a estrutura chamada de reservatório (tradução livre para *reservoir*). Esse reservatório pode ser entendido como um *kernel*³ temporal, projetando a entrada em um espaço dinâmico não linear. Durante a operação desse reservatório, os seus estados formam uma trajetória

³*Kernels* são utilizados na teoria do aprendizado estatístico para a expansão dos algoritmos de vetor suporte para problemas não lineares. O *kernel* é empregado para mapear a entrada em um espaço de alta dimensionalidade, fazendo com que o problema possa ser resolvido mais facilmente (SMOLA; SCHÖLKOPF, 2004).

dependente dos estímulos externos e também da memória de estímulos passados. A computação é feita pela camada de saída, que lê os estados instantâneos do reservatório (ANTONELO, 2012).

As principais arquiteturas de redes neurais que utilizam esse paradigma são a rede de estado de eco (ESN, do inglês *Echo State Network*), proposta por Jaeger e Haas (2004), a máquina de estado líquido (LSM, do inglês *Liquid State Machine*), proposta por Maass et al. (2002), e a rede *BackPropagation-DeCorrelation* (BPDC), proposta por Steil (2004). Os modelos ESN e BPDC utilizam um reservatório composto por neurônios analógicos, idênticos ao utilizados em redes MLP, e são utilizados principalmente em aplicações de engenharia. Já o modelo LSM utiliza neurônios pulsantes (tradução livre para *spiking neurons*), sendo usado principalmente em aplicações de neurociência computacional, por ser um dos modelos mais fiéis ao comportamento de algumas áreas do cérebro humano (MAASS, 2001; MAAS et al., 2013).

Na subseção 2.2.2.1 é apresentada em detalhes a arquitetura da ESN, a qual recebe destaque por ser comumente utilizada em tarefas de engenharia, caso do tarefa explorada neste trabalho. Por fim, na subseção 2.2.2.2 é apresentado o método de treinamento utilizado pelo mecanismo de leitura do reservatório da rede ESN.

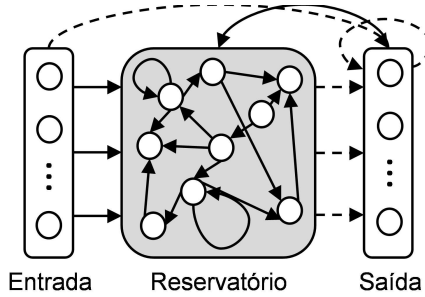
2.2.2.1 Redes neurais de estado de eco

A estrutura básica de uma ESN é ilustrada na figura 2.13. O reservatório é constituído de neurônios com ativação sigmoide, conectados de forma recorrente, característica responsável por seu comportamento dinâmico. As conexões entre os neurônios do reservatório são fixas e geradas de forma aleatória. Diversas conexões são opcionais, podendo ser omitidas dependendo da aplicação desejada. Na prática, as únicas conexões obrigatórias são as internas ao reservatório, e o mecanismo de leitura do reservatório, o qual liga o reservatório com a camada de saída (LUKOŠEVIČIUS, 2012).

A figura 2.14 apresenta um diagrama de blocos para o modelo ESN completo, no qual os blocos tracejados representam matrizes treináveis. Na tabela 2.1 são detalhados os elementos apresentados no diagrama de blocos.

As matrizes de conexão não treináveis, \mathbf{W}_{res}^{res} , \mathbf{W}_{in}^{res} , \mathbf{W}_{out}^{res} e \mathbf{W}_{bias}^{res} , são geradas, geralmente, usando uma distribuição aleatória, por exemplo a distribuição Gaussiana. Para sua inicialização são utilizados dois parâmetros, a taxa de conexão (c_{de}^{para}), que corresponde à porcentagem de pesos não nulos na matriz de conexão, e o

Figura 2.13: Arquitetura da rede ESN. Linhas sólidas representam conexões fixas, enquanto tracejados representam pesos treináveis.



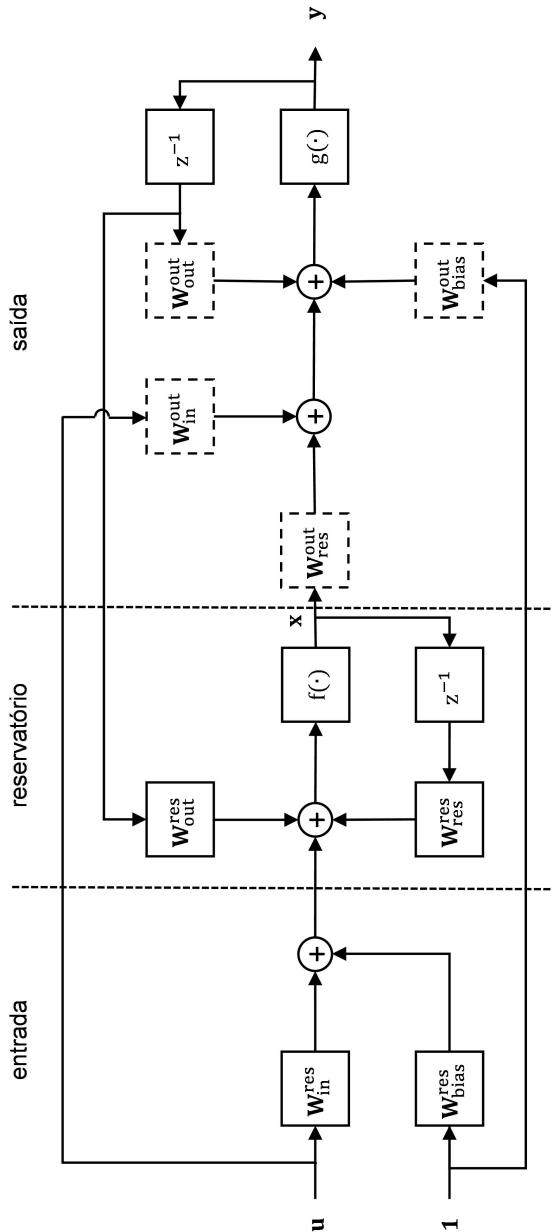
Fonte: adaptado de Antonelo (2012).

escalamento (v_{de}^{para}), que corresponde aos valores com os quais são escaladas as matrizes de conexão. O parâmetro v apresenta grande influência na dinâmica do reservatório, devendo ser ajustado para a obtenção de um desempenho ótimo (ANTONELO, 2012).

Os valores iniciais das matrizes afetam o comportamento geral do modelo, sendo que algumas propriedades dessas matrizes apresentam maior importância no comportamento dinâmico do reservatório. A matriz \mathbf{W}_{res}^{res} , em especial, apresenta grande importância, sendo gerada de forma que o modelo mantenha a propriedade de estado de eco, que pode ser entendida como a propriedade do reservatório que faz com que a memória desvaneça assintoticamente apagando qualquer informação das condições iniciais. Isso é alcançado com um reservatório estável, cujos autovalores estejam dentro do círculo de raio unitário no plano complexo. Para isso, o valor absoluto do autovalor com maior norma $\rho(\mathbf{W}_{res}^{res})$, denominado raio espectral, deve ser menor que a unidade, ou seja, $\rho(\mathbf{W}_{res}^{res}) < 1$. Na prática, o reservatório é inicializado com valores aleatórios, e a matriz de pesos é escalada, de forma a ajustar o maior autovalor da matriz ao valor desejado (LUKOŠEVIČIUS, 2012).

Para ajustar a escala de tempo do reservatório, de forma que acompanhe a dinâmica temporal dos sinais de entrada, é utilizado o conceito do reservatório com unidades integradoras com vazamento (do inglês *leaky integrator units*). Esse conceito pode ser compreendido como se cada unidade do reservatório armazenasse seu

Figura 2.14: Diagrama de blocos completo da rede ESN.



Fonte: adaptado de Antonelo (2012).

Tabela 2.1: Elementos da figura 2.14

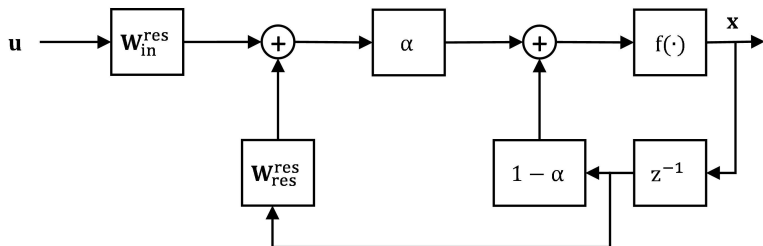
Sinais	
\mathbf{u}	sinal de entrada
\mathbf{y}	sinal de saída
\mathbf{x}	estado do reservatório
$\mathbf{1}$	vetor de elemento iguais a 1
Pesos	
\mathbf{W}_{in}^{res}	matriz de conexão entre a entrada e o reservatório
\mathbf{W}_{bias}^{res}	matriz com pesos de bias para a matriz de conexão do reservatório
\mathbf{W}_{res}^{res}	matriz de conexão do reservatório
\mathbf{W}_{out}^{res}	matriz de conexão entre a saída e o reservatório
\mathbf{W}_{in}^{out}	matriz de conexão entre a entrada e a saída
\mathbf{W}_{res}^{out}	matriz de conexão entre o reservatório e a saída
\mathbf{W}_{out}^{out}	matriz de conexão da saída
\mathbf{W}_{bias}^{out}	matriz com pesos de bias para a matriz de conexão da saída
Funções	
$f(\cdot)$	função de transferência f
$g(\cdot)$	função de transferência g

estado líquido, mas estivesse vazando a uma taxa α (ANTONELO, 2012).

Com a utilização dessa taxa de vazamento $\alpha \in (0,1]$, é possível alterar a memória do reservatório. Considerando a estrutura apresentada na figura 2.15, para um mesma matriz \mathbf{W}_{res}^{res} , a taxa de vazamento ajustada em valores pequenos produzem reservatórios com maior memória, enquanto valores próximos a 1 fazem o reservatório ter pouca ou nenhuma memória (ANTONELO, 2012).

O funcionamento geral da rede ESN pode ser descrito, então, pelas equações (2.9) e (2.10). A primeira descreve o comportamento dinâmico do reservatório, sendo a equação de atualização dos estados. Já a segunda define o mecanismo de leitura do reservatório, ou seja, o mapeamento entre os estados internos do reservatório e a saída da rede. Os parâmetros empregados nas equações são os mesmos descritos na tabela 2.1 (LUKOŠEVIČIUS, 2012).

Figura 2.15: Diagrama de blocos para unidades integradoras com vazamento. Por simplicidade, são ignoradas as matrizes de realimentação externa e bias nessa representação.



Fonte: autor.

$$\mathbf{x}(t+1) = f((1-\alpha)\mathbf{x}(t) + \alpha(\mathbf{W}_{in}^{res}\mathbf{u}(t) + \mathbf{W}_{res}^{res}\mathbf{x}(t) + \mathbf{W}_{bias}^{res}\mathbf{1} + \mathbf{W}_{out}^{res}\mathbf{y}(t))) \quad (2.9)$$

$$\mathbf{y}(t+1) = g(\mathbf{W}_{res}^{out}\mathbf{x}(t+1) + \mathbf{W}_{bias}^{out}\mathbf{1} + \mathbf{W}_{in}^{out}\mathbf{u}(t) + \mathbf{W}_{out}^{out}\mathbf{y}(t)) \quad (2.10)$$

2.2.2.2 Treinamento da ESN

A saída da rede ESN é formada através de uma combinação linear dos estados internos do reservatório, das saídas passadas no instante anterior, das entradas atuais e pesos de bias. O resultado dessa combinação linear passa pela função de pós-processamento $g(\cdot)$, que é utilizada, geralmente, em problemas de classificação, para mapear a saída contínua em valores discretos. Como na maioria dos casos, as matrizes \mathbf{W}_{in}^{out} e \mathbf{W}_{out}^{out} não são utilizadas, o treinamento será apresentado apenas para o mecanismo de leitura do reservatório, \mathbf{W}_{res}^{out} , e o bias \mathbf{W}_{bias}^{out} , sendo facilmente expandido para os demais casos (LUKOŠEVIČIUS, 2012).

Se a saída da rede for representada como

$$\mathbf{Y} = \mathbf{W}^{out}\mathbf{X}, \quad (2.11)$$

onde $\mathbf{Y} \in \mathfrak{R}^{N_y \times T}$ é definida como

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(t) & \mathbf{y}(t+1) & \dots & \mathbf{y}(t+T-1) \end{bmatrix}, \quad (2.12)$$

e $\mathbf{X} \in \mathfrak{R}^{(1+N_u+N_x) \times T}$ é uma matriz formada por

$$\mathbf{X} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \dots & \mathbf{1} \\ \mathbf{u}(t) & \mathbf{u}(t+1) & \dots & \mathbf{u}(t+T-1) \\ \mathbf{x}(t) & \mathbf{x}(t+1) & \dots & \mathbf{x}(t+T-1) \end{bmatrix}, \quad (2.13)$$

ambas geradas ao excitar o reservatório com $\mathbf{u}(t)$ em um período de treinamento de T amostras (LUKOŠEVIČIUS, 2012).

A tarefa de aprendizado consiste em encontrar a solução ótima de $\mathbf{W}^{\text{out}} \in \mathfrak{R}^{N_y \times (1+N_u+N_x)}$ que minimize o erro quadrático entre $\mathbf{y}(t)$ e $\mathbf{y}^{\text{alvo}}(t)$, para os instantes $t, t+1, \dots, t+T-1$. $\mathbf{y}^{\text{alvo}}(t)$ é uma matriz formada da mesma forma que \mathbf{Y} , representando o conjunto de saídas do conjunto de treinamento, usados como alvo almejado para o treinamento. O problema se resume então a encontrar a solução do sistema de equações lineares

$$\mathbf{Y}^{\text{alvo}} = \mathbf{W}^{\text{out}} \mathbf{X}, \quad (2.14)$$

o qual é sobredeterminado, ja que tipicamente $T \gg 1 + N_u + N_x$ (LUKOŠEVIČIUS, 2012).

A alternativa comumente utilizada para encontrar \mathbf{W}^{out} é conhecida como a regressão com regularização (do inglês *ridge regression*), a qual otimiza a seguinte função objetivo:

$$\mathbf{W}^{\text{out}} = \underset{\mathbf{W}^{\text{out}}}{\operatorname{argmin}} \frac{1}{N_y} \sum_{i=1}^{N_y} \left(\sum_{n=1}^T (y_i(n) - y_i^{\text{alvo}}(n))^2 + \beta \|\mathbf{w}_i^{\text{out}}\|^2 \right). \quad (2.15)$$

Essa função objetivo busca o argumento \mathbf{W}^{out} que minimiza o erro quadrático médio entre a saída da rede e a saída alvo, ao mesmo tempo que penaliza valores altos para \mathbf{W}^{out} utilizando o coeficiente de regularização β . Isso representa um compromisso entre um erro de treinamento pequeno e pesos de saída pequenos (LUKOŠEVIČIUS, 2012).

A solução desse problema é dada por (BISHOP, 2006):

$$\mathbf{W}^{\text{out}} = (\mathbf{X}^T \mathbf{X} + \beta \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}^{\text{alvo}}. \quad (2.16)$$

Esse método é indicado para o treinamento de ESN, pois valores muito elevados para \mathbf{W}^{out} podem ser um indicativo de uma solução muito sensível ou instável. Pesos elevados fazem com que pequenas diferenças nos estados do reservatório sejam amplificadas, fazendo com que a rede seja muito sensível à operação em condições levemente diferentes das observadas no treinamento. Devido a isso é utilizado o termo de regularização β , que resulta em uma penalização para os valores elevados de \mathbf{W}^{out} , fazendo com que a solução tenha um compromisso entre um erro pequeno de treinamento e pesos pequenos na matriz \mathbf{W}^{out} (LUKOŠEVIČIUS, 2012).

Pelo fato de serem ajustados apenas os pesos do mecanismo de leitura do reservatório, o treinamento dessa arquitetura de rede evita diversas desvantagens encontradas no treinamento de redes MLP com conexões recorrentes. Dentre essas desvantagens, destaca-se o elevado tempo de treinamento e a ocorrência do fenômeno do desaparecimento do gradiente, que afeta redes profundas que utilizam métodos de treinamento baseados no cômputo do gradiente, caso das redes MLP com conexões recorrentes (LUKOŠEVIČIUS; JAEGER, 2009).

2.3 COMENTÁRIOS FINAIS

A revisão da literatura apresentou os fundamentos para a avaliação de arquiteturas de redes neurais artificiais aplicadas em problemas de identificação de sistemas. Arquiteturas derivadas do modelo MLP são as mais comuns, configurando-se como uma escolha padrão. Por sua vez, arquiteturas que empregam o conceito da computação de reservatório apresentam uma alternativa às arquiteturas MLP recorrentes, com grande potencial para aplicação em problemas de identificação.

Os conceitos de identificação e RNA são combinados no capítulo 3 para definir uma metodologia de identificação de sistemas usando RNAs recorrentes. No capítulo 4, as redes NARX, RMLP e ESN são avaliadas experimentalmente para identificação de uma planta real, possibilitando uma avaliação da aplicação de tais arquiteturas em tarefas de identificação.

3 ESTUDO DE CASO E METODOLOGIA

O trabalho foi desenvolvido seguindo uma metodologia para identificação de sistemas dinâmicos não lineares. Essa metodologia foi aplicada almejando a obtenção de modelos adequados, que possibilitem a avaliação das arquiteturas de RNAs em estudo, bem como a comparação do desempenho entre os modelos. Para possibilitar a avaliação de arquiteturas de RNA recorrentes em tarefas de identificação de sistemas, é definido um estudo de caso, no qual são aplicadas as arquiteturas, possibilitando uma avaliação experimental das mesmas.

O presente capítulo é dividido em três seções. A primeira delas, 3.1, apresenta a metodologia para identificação de sistemas com RNA que foi aplicada no desenvolvimento deste trabalho. Em seguida, a seção 3.2 apresenta uma descrição da bancada utilizada como estudo de caso. Já na seção 3.3, são feitos os comentários finais do capítulo.

3.1 METODOLOGIA PARA IDENTIFICAÇÃO DE SISTEMAS COM REDES NEURAIS

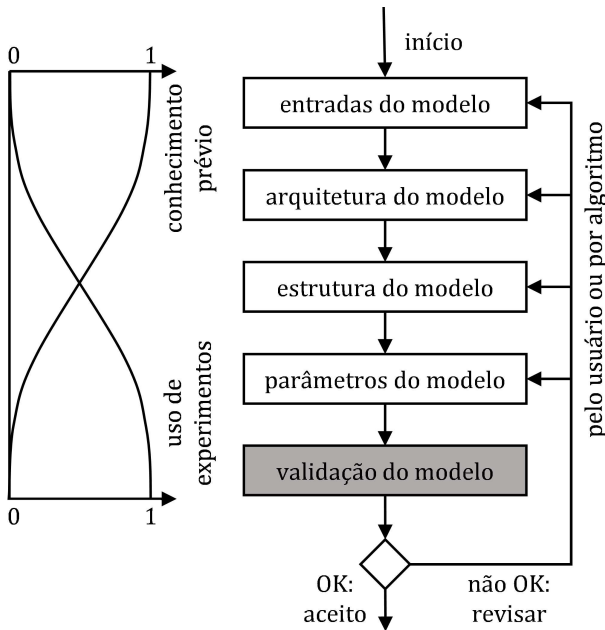
A identificação de sistemas usando redes neurais compartilha diversas características e desafios apresentados por outras famílias de modelos não lineares. Metodologias gerais para identificação de sistemas não lineares podem ser utilizadas naturalmente com redes neurais, sendo que uma delas, apresentada em Nelles (2001), é discutida nesta seção. Devido ao trabalho aplicar apenas modelos que utilizam redes neurais recorrentes, são apresentados apenas os passos necessários para as ferramentas em questão.

O principais passos para a identificação de um sistema dinâmico usando redes neurais artificiais recorrentes estão sumarizados no fluxograma da figura 3.1. Nesse fluxograma, é apresentado o laço de tarefas executadas para a identificação de um sistema genérico através de redes neurais artificiais.

As tarefas iniciais necessitam de um alto conhecimento prévio acerca do processo que será identificado. A definição das entradas do modelo e da sua arquitetura são, de maneira geral, efetuadas pelo projetista. As etapas posteriores, como a definição da estrutura do modelo, dos seus parâmetros e sua validação, são realizadas através de, principalmente, experimentos.

Na etapa de escolha da arquitetura do modelo, é escolhida a arquitetura específica de rede neural recorrente. Essa etapa é impor-

Figura 3.1: Fluxograma do laço da identificação de sistemas usando redes neurais recorrentes.



Fonte: adaptado de (NELLES, 2001)

tante para a seleção de um modelo adequado ao tipo de problema de identificação para o qual ele será empregado. O tipo de rede neural e a arquitetura escolhidos podem limitar a gama de aplicações, devido a aspectos inerentes da arquitetura escolhida.

Com base na arquitetura escolhida, é necessária a definição de uma estrutura específica, o que abrange, dependendo do modelo: número de camadas, número de neurônios por camada, função de ativação dos neurônios, parâmetros construtivos, taxa de conectividade, taxa de vazamento, entre outros. Esse ajuste pode ser feito diretamente pelo projetista, requerendo um alto conhecimento tanto do problema, quanto do funcionamento e ajuste do modelo, ou através de métodos experimentais, como a busca em grade. Na busca

em grade, os principais parâmetros são ajustados através do treinamento extensivo de redes, avaliando o desempenho do modelo para diferentes combinações de parâmetros construtivos e de ajuste dos modelos. A utilização desse método possibilita a obtenção de regiões em que os parâmetros do modelo apresentam melhor desempenho, possibilitando um melhor ajuste do modelo (GOODFELLOW et al., 2016; LUKOŠEVIČIUS, 2012).

A etapa de obtenção dos parâmetros do modelo consiste no treinamento dos pesos sinápticos da rede neural. Nessa etapa são definidos os parâmetros utilizados para o ajuste do processo de treinamento, que podem englobar desde a seleção do algoritmo de ajuste dos pesos até os parâmetros de ajuste do algoritmo escolhido, como a seleção de parâmetros de regularização, critérios de parada, entre outros. Com esses parâmetros de treinamento é possível realizar o treinamento do modelo, obtendo então os pesos sinápticos.

A validação do modelo é realizada através da utilização de um conjunto de testes, contendo uma porção do ensaio de identificação diferente da utilizada no treinamento do modelo. Para a aceitação do modelo são usados parâmetros de desempenho, que propiciam tanto a comparação entre o desempenho de modelos, quanto a avaliação geral da qualidade de ajuste do modelo. Entre os parâmetros geralmente utilizados na literatura, destacam-se a soma dos erros quadráticos, o erro quadrático médio e o coeficiente de determinação R^2 (ISERMANN; MÜNCHOF, 2011).

Caso o modelo estimado não seja aprovado, a sua revisão é feita, em geral, verificando as etapas no sentido inverso do ciclo, ou seja, verificando primeiro as etapas de ajuste de parâmetros e da arquitetura, em detrimento das etapas de escolha das entradas e arquitetura do modelo, as quais mais dificilmente necessitam de revisão.

Em seguida, na seção 3.2 é descrita a bancada, a qual foi utilizada para o desenvolvimento de um estudo de caso. Esse estudo de caso emprega os passos da metodologia apresentada, sendo que as seções 3.2.2 e 3.2.3 apresentam o primeiro passo da metodologia, que é a seleção das entradas do modelo. Nessas duas seções é explicitado o ensaio de identificação realizado no processo e é detalhada a série de dados obtida, passos necessários para realizar a seleção dos conjuntos de entrada do processo de identificação.

3.2 ESTUDO DE CASO: IDENTIFICAÇÃO DE UMA BANCADA DE CONTROLE DE PRESSÕES

A indústria de refrigeração realiza um investimento constante em pesquisa e desenvolvimento buscando a qualidade de seus produtos. Para a avaliação dos compressores herméticos são realizados diversos ensaios, utilizados para finalidades distintas, em especial para pesquisa e desenvolvimento e para o controle de qualidade do produto. De forma geral, ensaios para avaliação de compressores apresentam a necessidade de manter o compressor operando em regimes específicos, com ao menos uma variável do compressor controlada. O controle automático da operação desse tipo de bancada é importante, podendo influenciar diretamente no tempo de execução do ensaio. Isso faz com que a modelagem e a identificação desse tipo de sistema apresentem grande importância, por darem suporte ao desenvolvimento de aplicações de controle, bem como por auxiliarem a análise das mesmas (FLESCHE; NORMEY-RICO, 2010; SCHMITZ, 2016).

O modelamento de sistemas de refrigeração é um exercício complexo. A realização de um modelamento muito detalhado pode até resultar em modelos com alta exatidão, mas que geralmente apresentam custos de desenvolvimento proibitivos, com longos tempos de computação e que necessitam de vastas quantidades de dados provenientes do usuário. Devido a isso, técnicas caixa-cinza ou caixa-preta surgem como uma alternativa importante para suplanar essas limitações (ESTRADA-FLORES et al., 2006).

3.2.1 Descrição do processo

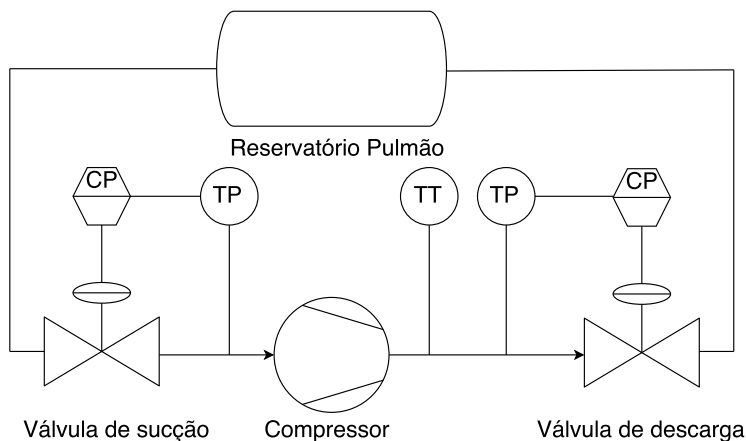
O processo em estudo consiste em uma bancada empregada para impor condições de pressão de sucção e de descarga a compressores de refrigeração para que possam ser medidas características de operação em diferentes condições. Dessa forma, podem ser emuladas em bancada diferentes condições de operação a que compressores herméticos de refrigeração podem ser submetidos durante seu uso em campo. Essa bancada foi projetada no trabalho de Schmitz (2016), onde podem ser encontrados mais detalhes sobre a construção da mesma.

A tarefa realizada pela bancada de testes consiste em impor condições termodinâmicas específicas, de forma que possam ser realizadas as medições de interesse. Em muitos casos, as condições termodinâmicas de operação do compressor podem ser caracteriza-

das bem pelos pares de pressões associadas às linhas de sucção e descarga do compressor, além da temperatura de sucção, que geralmente está próxima à temperatura ambiente, não sendo tratada no presente trabalho.

A bancada contém válvulas tanto na linha de entrada de fluido refrigerante quanto na linha de saída, com o objetivo de manipular as pressões de sucção e descarga do compressor. Essas duas válvulas são conectadas através de um reservatório pulmão, o qual tem como função desacoplar parcialmente as duas pressões. O diagrama de processo e instrumentos é mostrado na figura 3.2.

Figura 3.2: Diagrama simplificado de processo e instrumentos para bancada de emulação de condições de operação aplicada a compressores herméticos de refrigeração.



Fonte: autor.

A válvula localizada na linha de sucção é normalmente fechada, fazendo com que a pressão na sucção do compressor seja diretamente proporcional à tensão aplicada à válvula. Já a válvula associada à descarga do compressor é normalmente aberta, fazendo com que a pressão de descarga seja também diretamente proporcional à tensão aplicada à respectiva válvula. É importante ressaltar que, pela natureza do processo, existe um acoplamento entre as pressões de sucção e descarga, fazendo com que mudanças na

posição da válvula associada à sucção afetem também a pressão de descarga. Já o acoplamento no sentido descarga-sucção é desprezível, sendo quase totalmente compensado pelo reservatório pulmão.

O diagrama mostrado na figura 3.2 apresenta uma estrutura simplificada do processo real. Existem adicionalmente várias medições de temperatura e pressão, que são empregadas para diferentes ensaios, mas não serão utilizadas no escopo deste trabalho.

O problema de identificação consiste no modelamento das pressões associadas à sucção e à descarga do compressor, que serão denominada neste trabalho como P_s e P_d , respectivamente. Essas pressões são manipuladas através da abertura das válvulas posicionadas na sucção e na descarga do compressor, denominadas V_s e V_d , respectivamente. As pressões de sucção e descarga estão intimamente ligadas às temperaturas do fluido refrigerante nos pontos em questão, fazendo com que fontes de calor perturbem as pressões. Esse é o caso do próprio compressor, cuja temperatura durante o ensaio varia com o tempo, perturbando as pressões, em especial P_d , a qual é medida diretamente após o fluido ser passar pelo compressor. Devido a isso, a temperatura associada à linha de descarga é utilizada na forma de uma perturbação mensurável do processo.

A seção 3.2.2 apresenta o ensaio realizado na bancada descrita na seção atual. Esse ensaio faz parte do primeiro passo da metodologia apresentada na seção 3.1, sendo necessário para a obtenção da série de dados que será usada como entrada para o processo de identificação. A primeira etapa da metodologia prossegue na seção 3.2.3, na qual é apresentada a série de dados que foi usada no decorrer do desenvolvimento, bem como o pré-processamento aplicado nos dados.

3.2.2 Ensaio de identificação

Para a identificação de um sistema através de redes neurais recorrentes, é necessário obter uma série de dados representativa do processo. Para a obtenção de tal série, foi realizado um ensaio de identificação, no qual as pressões foram excitadas através de um sinal APRBS aplicado às válvulas.

A utilização de redes neurais artificiais exige alguns cuidados adicionais no projeto do sinal de excitação. Para redes neurais, uma condição básica para uma boa generalização são séries de dados suficientemente grandes. As RNAs são notoriamente boas interpoladoras, mas geralmente não são capazes de extrapolar confiavelmente (BARNARD; WESSELS, 1992). Por isso, é importante haver

dados de treinamento suficientes, de forma a evitar a necessidade de extrapolação. É importante, então, excitar o processo em todas as possíveis regiões de operação, de forma a garantir que a rede neural seja aplicada apenas a casos que não extrapolem o conjunto de treinamento (SVOZIL et al., 1997).

Com base no estudo da bancada apresentado por Schmitz (2016), foi gerado um sinal APRBS capaz de excitar o sistema de modo que as várias regiões de operação fossem varridas. Nesse estudo, foi apresentado o perfil de ganho para as pressões de sucção e descarga, ao serem excitadas pelas variações de tensão nas válvulas de sucção e descarga. A pressão de sucção apresenta ganhos não nulos na região entre 1 V e 6 V, enquanto a descarga apresenta ganhos não nulos na região entre 5 V e 10 V.

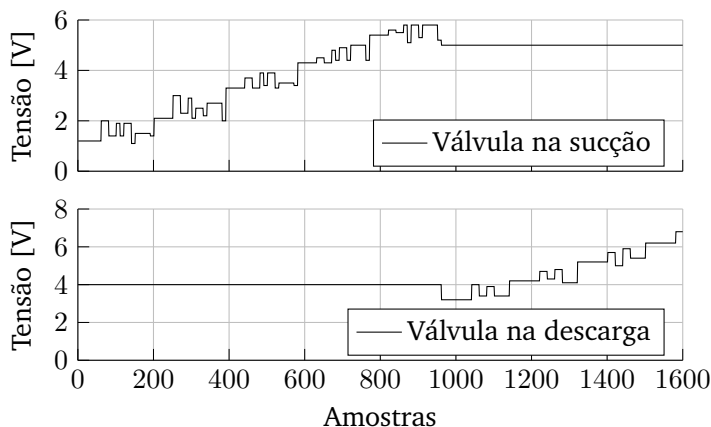
Para a válvula de sucção, foi gerado um sinal APRBS, o qual foi gerado com amplitude 1 V. Esse sinal foi escalado usando *offsets* variando entre 1,5 V e 5,5 V, de forma a percorrer os vários pontos de operação para a pressão de sucção, enquanto a válvula de descarga foi mantida em um patamar constante. Esse sinal APRBS foi replicado para diversos patamares constantes da válvula de descarga, buscando excitar todos os pontos de operação para a pressão de descarga. A duração de cada patamar foi projetada considerando o período de amostragem de 200 ms. O mesmo procedimento foi realizado para a excitação da válvula de descarga, nesse caso com a válvula de sucção sendo mantida em patamares constantes e a válvula de descarga sendo excitada com um sinal APRBS com amplitude 1 V, escalados usando *offsets* que variaram entre 3 V e 7 V. Uma porção do sinal de excitação é mostrada na figura 3.3.

A resposta do processo à excitação aplicada é mostrada na porção do ensaio de identificação apresentada na figura 3.4. A primeira porção, apresentada na figura 3.4a, exhibe a resposta do sistema à excitação da válvula de sucção, enquanto a segunda porção, apresentada na figura 3.4b, mostra a resposta à excitação da válvula de descarga.

O comportamento dinâmico apresentado pelo processo durante o ensaio de identificação é o esperado. A pressão de sucção é manipulada através da válvula de sucção, não havendo perturbação significativa provocada por variações na pressão de descarga devido ao reservatório pulmão, o qual funciona como um desacoplador. Já a pressão de descarga é manipulada pela válvula localizada na descarga, apresentando um forte acoplamento com a pressão de sucção, decorrente da natureza do funcionamento do compressor.

A temperatura medida na descarga do compressor é uma

Figura 3.3: Porção do sinal de excitação APRBS.



Fonte: autor.

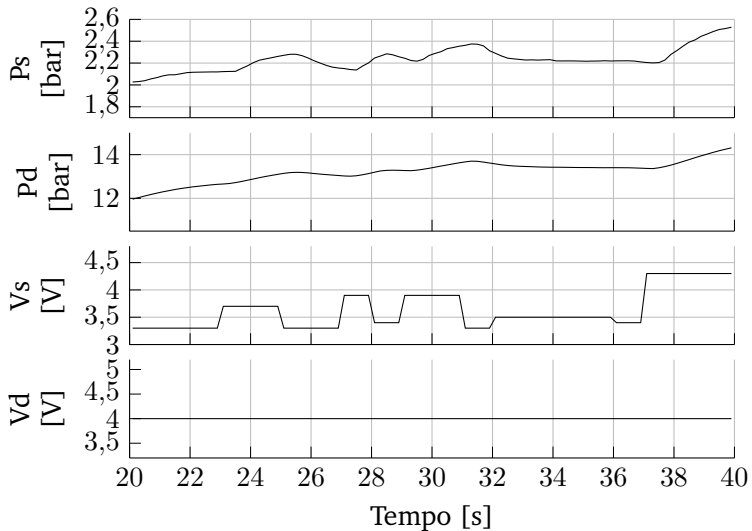
perturbação mensurável do sistema. Por não ser possível excitar diretamente essa variável, foi tomado o cuidado para produzir, no ensaio de identificação, uma série de dados que contenha valores de temperatura na descarga em toda a faixa possível de valores. Ela apresenta variação, durante o ensaio, entre a temperatura ambiente, a qual é controlada no ambiente laboratorial, mantida próximo de 20 °C, e uma temperatura máxima de 80 °C, atingida após horas de operação. Essa temperatura apresenta, também, comportamento dinâmico proveniente das variações do ponto de operação da pressão de descarga. É importante ressaltar que, durante a operação convencional da bancada, essas variações na temperatura geralmente se verificam, com um ensaio típico iniciando em temperaturas baixas e finalizando em valores elevados.

3.2.3 Série de dados e a definição das entradas dos modelos

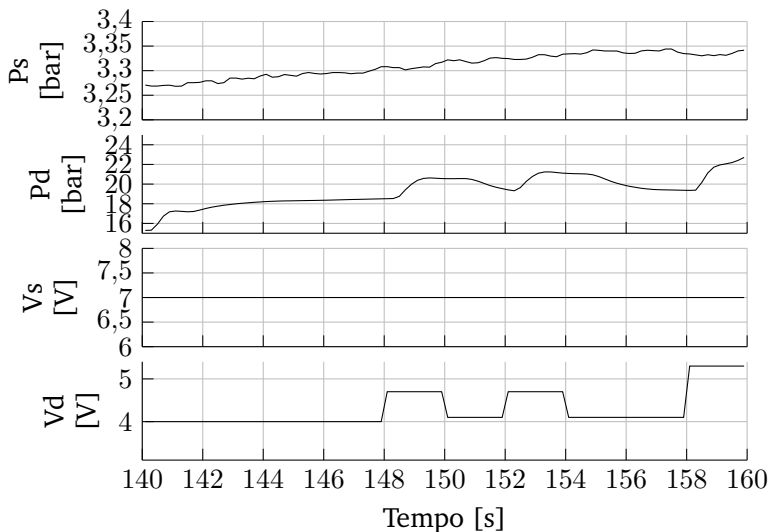
O ensaio de identificação produziu uma série de dados, que passou por uma etapa de seleção e pré-processamento, a qual é geral para todos modelos. Esse procedimento permite a obtenção dos conjuntos usados no treinamento, validação e teste dos modelos, sendo importante empregar os mesmos conjuntos para todos os modelos, de forma a possibilitar a comparação dos resultados.

A divisão da série de dados em conjuntos separados, um

Figura 3.4: Porções do ensaio de identificação com as respostas do sistema à excitação APRBS. São mostradas a pressão de sucção, P_s , a pressão descarga, P_d , a tensão na válvula de sucção, V_s , e a tensão na válvula de descarga, V_d .



(a) Resposta à variação da válvula de sucção



(b) Resposta à variação da válvula de descarga

Fonte: autor.

para o treinamento do modelo e um para o teste do mesmo, é um procedimento usual para RNAs, o qual foi adotado neste trabalho. A série de dados obtida no ensaio de identificação corresponde a três ensaios com excitação APRBS, realizados de forma subsequente, contendo 1686, 1681 e 1697 amostras, respectivamente. O período de amostragem usado corresponde a 0,2 s, e cada ensaio dura 340 s, aproximadamente. Para o conjunto de treinamento foram selecionados o primeiro e o terceiro ensaio, enquanto o segundo foi designado para testes. Do conjunto de treinamento, foi retirada uma porção para auxiliar na escolha do modelo por meio de regularização, conjunto que foi denominado como conjunto de validação. Isso resultou em aproximadamente, 55% dos dados para treinamento, 15% para validação e 30% para testes. Os dados foram selecionados de forma a garantir que o conjunto de treinamento contivesse os exemplos extremos, bem como exemplos intermediários, enquanto os conjuntos de validação e testes fossem compostos por exemplos intermediários.

A série de dados passou por uma etapa de pré-processamento, de forma a evitar problemas no aprendizado dos modelos. Nessa etapa, foi realizada a normalização tanto do conjunto de entrada, quanto de saída, consistindo na normalização dos valores máximos e mínimos das séries de dados em uma escala entre -1 e $+1$. O processo de normalização dos conjuntos de entrada de redes neurais é importante para melhorar a estabilidade dos algoritmos de treinamento, a qual pode ser afetada por entradas de valores elevados saturando os neurônios da camada de entrada. A normalização do conjunto de saídas apresenta importância especial neste trabalho, pois alguns modelos utilizam conexões recorrentes a partir da camada de saída, fazendo com que a saída atue como uma entrada adicional do modelo, necessitando então passar pelo mesmo processo de normalização que o conjunto de entradas.

3.3 COMENTÁRIOS FINAIS

O capítulo atual apresentou a metodologia que é utilizada para a avaliação das arquiteturas de RNA recorrentes. Além da metodologia, o capítulo apresentou o estudo de caso que serve de base para a avaliação experimental das arquiteturas de RNA. Dentro do estudo de caso, foi apresentado o processo que é identificado no decorrer deste trabalho, sendo detalhado o ensaio de identificação, realizado no processo para a obtenção da série de dados que foi

usada nas etapas de identificação.

Enquanto o capítulo atual apresenta o primeiro passo da metodologia, o qual é geral para todos modelos, o capítulo 4 detalha os passos subsequentes, os quais são apresentados para cada arquitetura de RNA avaliada neste trabalho. É mostrado o desenvolvimento dos modelos empregando desde redes derivadas do modelo MLP, como as redes NARX e RMLP, até arquiteturas de rede que utilizam o paradigma da computação de reservatório, com três implementações distintas de ESN. Para cada arquitetura selecionada, foi definida uma estrutura, a qual passou por uma etapa de treinamento para gerar um modelo para o processo. Todos os modelos foram testados usando o conjunto de testes, passando por uma comparação de seus resultados.

4 IDENTIFICAÇÃO COM REDES NEURAIS RECORRENTES

O presente capítulo apresenta o desenvolvimento dos modelos para a identificação do processo em estudo neste trabalho. São apresentadas as etapas da metodologia, a qual foi introduzida na seção 3.1, que são dependentes da escolha do modelo. As etapas independentes do modelo, como a de seleção das entradas do modelo foram apresentadas na seção 3.2.3.

Foram selecionadas três arquiteturas distintas de redes neurais recorrentes, com as quais foram desenvolvidos os modelos do processo. O desenvolvimento de cada arquitetura é explorado em seções distintas deste capítulo. A primeira arquitetura, apresentada na seção 4.1, é a rede neural NARX. Em seguida, é apresentado o desenvolvimento do modelo usando a rede neural RMLP, na seção 4.2. Por fim, na seção 4.3 é apresentado o desenvolvimento de três modelos empregando estruturas ESN distintas.

A última seção do capítulo, 4.4, aborda a comparação do desempenho dos modelos desenvolvidos. Essa comparação é realizada através dos resultados obtidos pelos modelos para as tarefas de simulação e predição e também de aspectos qualitativos, como o tempo de treinamento e versatilidade do modelo.

4.1 REDE NEURAL NARX

A construção de um modelo para o processo em estudo, utilizando a arquitetura de rede neural NARX, é apresentada nesta seção. São apresentados os demais passos da metodologia para identificação de sistemas dinâmicos não lineares, que consistem na definição da estrutura do modelo e dos parâmetros, os quais são abordados na seção 4.1.1, enquanto a etapa de validação é detalhada na seção 4.1.2.

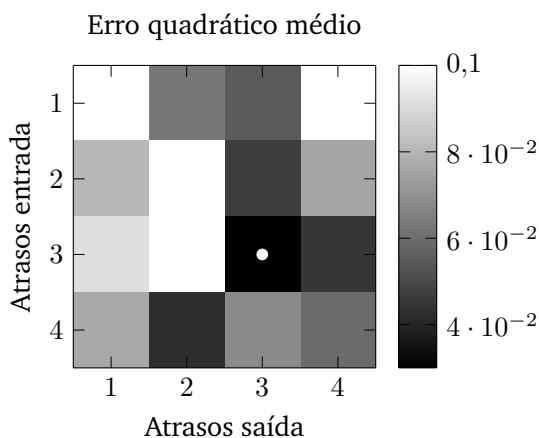
4.1.1 Construção do modelo e treinamento

A construção de um modelo utilizando a rede neural NARX inicia com a definição da morfologia da rede, tarefa que resume-se à escolha do número de camadas, neurônios por camada e número de atrasos no tempo associados à entrada da rede e à saída realimentada.

Optou-se pela definição de uma arquitetura inicial, contendo duas camadas escondidas, com 5 e 3 neurônios, respectivamente. Essa arquitetura foi usada para a escolha do número de atrasos associados à entrada e à saída realimentada. Para a seleção do valor

desses parâmetros foi utilizado um algoritmo de busca em grade, o qual treina redes variando os parâmetros que estão sendo selecionados, computando o desempenho dos modelos treinados em um conjunto de validação. Através desse procedimento, é possível escolher uma configuração adequada. Esse algoritmo testou valores distintos para os atrasos associados à entrada e à saída realimentada, variando cada atraso individualmente dentro do intervalo de 1 a 4 atrasos. Uma representação gráfica dos erros quadráticos médios das duas saídas da rede, obtidos pela busca em grade, é mostrado na figura 4.1.

Figura 4.1: Busca em grade para escolha da melhor configuração de atrasos para a entrada e a saída realimentada para a rede NARX. Células mais escuras representam menores erros quadráticos médios. O círculo branco indica o menor erro.



Fonte: autor.

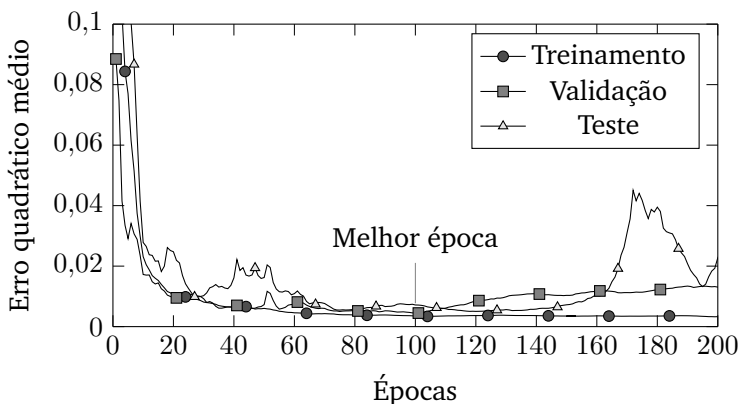
Pode-se verificar que as redes treinadas com um número maior de atrasos obtiveram resultados melhores que os apresentados por configurações com número baixo de atrasos. A rede com melhor desempenho foi treinada com 3 atrasos associados tanto à entrada quanto à saída realimentada, sendo esse par de atrasos selecionado para o modelo final.

O treinamento das redes NARX foi realizado utilizando o algoritmo BPTT, por meio do *toolbox* Pyrenn (ATABAY, 2016). Para o modelo final foram testados valores maiores para o número de

neurônios de cada camada, o que não se converteu em melhorias no desempenho da rede. Por isso, optou-se por manter os mesmos valores usados na etapa de escolha dos atrasos, resultando na arquitetura final com 3 neurônios na entrada, duas camadas escondidas com 5 e 3 neurônios, respectivamente, e 2 neurônios na camada de saída. Para arquiteturas recorrentes, em especial, a utilização de camadas com quantidades grandes de neurônios é desencorajada, devido ao alto custo demandado para o treinamento de tais modelos, o que motivou a manutenção de tal arquitetura.

Para evitar o superajuste da rede neural, foi utilizada regularização por parada precoce, através de validação cruzada (PRECHELT, 2012). A representação da evolução do erro quadrático médio nos três conjuntos da série de dados é mostrada na figura 4.2. A melhor época para o conjunto de validação foi a de número 100, com a rede associada a essa época sendo utilizada como o modelo final NARX. É interessante notar que a validação cruzada foi de extrema importância, já que apesar de o erro no conjunto de treinamento reduzir após a época 100, os erros, tanto no conjunto de validação quanto no de teste, aumentaram, com piora significativa no conjunto de testes.

Figura 4.2: Validação cruzada realizada durante o treinamento da rede NARX. A rede associada à época com menor erro no conjunto de validação é selecionada.



Fonte: autor.

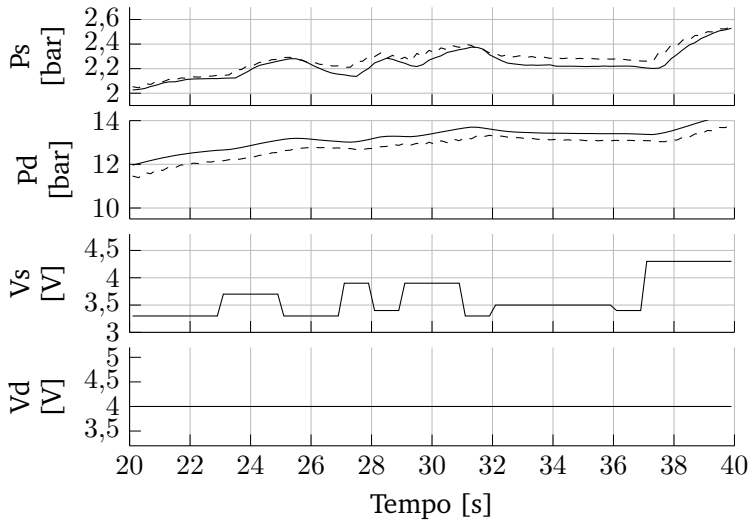
4.1.2 Resultados

O modelo NARX desenvolvido foi testado em dois contextos distintos, o de uma tarefa de simulação e o de predição. O comportamento do modelo NARX em uma tarefa de simulação é mostrado na figura 4.3. Na figura 4.3a é mostrada uma porção do conjunto de testes, composta de 200 amostras entre os instantes 20 s e 40 s do ensaio, correspondendo a um trecho do ensaio em que a válvula de sucção está sendo excitada. Por sua vez, em 4.3b é mostrado um trecho entre os instantes 140 s e 160 s do ensaio, no qual a válvula de descarga sofre excitação. É mostrada a saída do modelo, representada pela linha tracejada, comparada à saída real do processo, representado pela linha sólida. Os sinais de excitação do processo são mostrados em V_s e V_d .

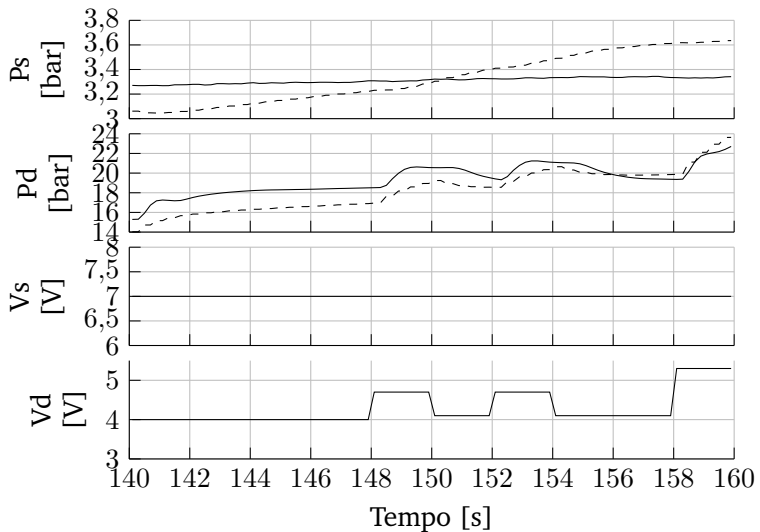
Os resultados obtidos pelo modelo NARX foram considerados satisfatórios, pois o modelo foi capaz de modelar o comportamento dinâmico do processo, sendo capaz de aprender o comportamento característico de cada variável. Os resultados apresentados pelo modelo para a pressão de descarga foram superiores aos da pressão de sucção, o que é evidenciado pelo erro quadrático médio (EQM). Enquanto o modelo NARX apresentou um EQM de 0,0229 para a sucção, foi encontrado 0,0070 para a descarga. Isso resultou em um R^2 de 0,8678 para a sucção, e 0,9639 para a descarga.

O comportamento do modelo NARX em uma tarefa de predição de N instantes à frente é mostrado na figura 4.4. A figura ilustra os resultados obtidos pelo modelo realizando a predição de 100 instantes à frente, com a predição iniciando nos instantes 20 s e 140 s do ensaio, figuras 4.4a e 4.4b respectivamente. O modelo foi excitado na configuração série-paralelo até o instante inicial da predição, no qual foram realizadas as 100 predições, com o modelo operando em paralelo. Nessa figura, assim como nas demais que são apresentadas na sequência deste trabalho, foram omitidos os sinais de entrada do modelo por não apresentarem informações novas. O modelo se beneficiou da possibilidade de operação do modelo na configuração série-paralelo para realizar predições mais precisas nos instantes iniciais do horizonte. Essa característica fica mais evidente analisando a pressão de sucção na figura 4.4b, a qual apresenta resultados expressivamente melhores em comparação aos verificados na tarefa de simulação, mostrados na figura 4.3b.

Figura 4.3: Resultados da rede NARX para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo NARX.



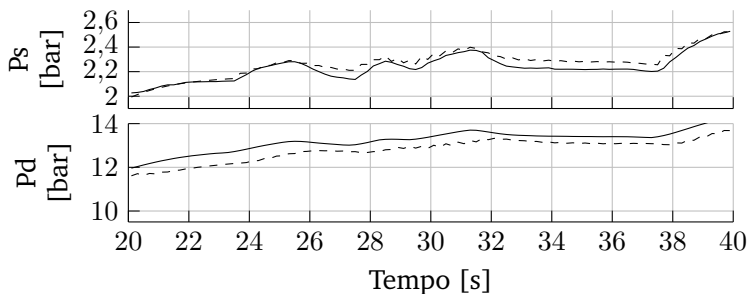
(a) Resposta a variação da válvula de sucção



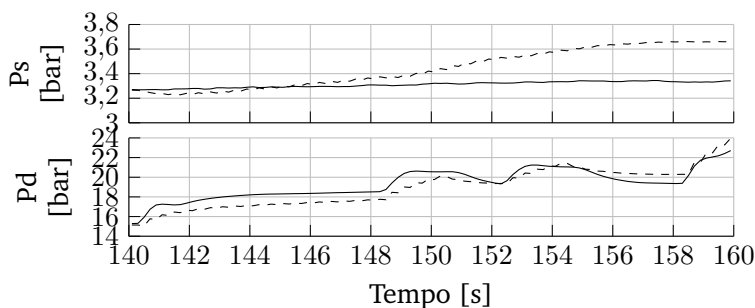
(b) Resposta a variação da válvula de descarga

Fonte: autor.

Figura 4.4: Resultados da rede NARX para a tarefa de predição de N instantes à frente. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo NARX.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

Fonte: autor.

4.2 REDE NEURAL RMLP

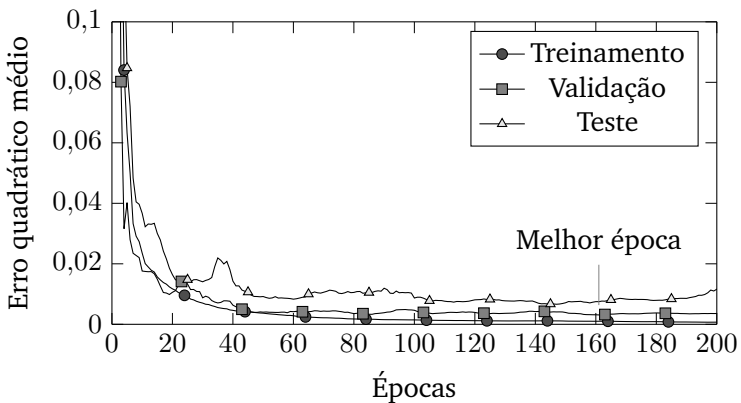
Da mesma forma como apresentado para a rede NARX, nesta seção é detalhado o desenvolvimento de um modelo para o processo utilizando a rede neural RMLP. São apresentados na seção 4.2.1 a construção do modelo e o treinamento, correspondendo às etapas de definição da estrutura e dos parâmetros do modelo. Em seguida são apresentados os resultados, na seção 4.2.2, correspondendo à validação do modelo.

4.2.1 Construção do modelo e treinamento

A arquitetura da rede RMLP não apresenta grande flexibilidade. Os principais parâmetros construtivos são o número de camadas e de neurônios por camada. Foi definido para o modelo RMLP final uma arquitetura com duas camadas, uma escondida e uma de saída, sendo que a camada escondida empregou 10 neurônios. A escolha do número de neurônios não é tão crítica, pois, geralmente, com um número pequeno de neurônios é alcançado um modelo com alta complexidade, a qual é regularizada através da utilização de métodos de parada precoce no algoritmo de treinamento.

O treinamento do modelo RMLP foi realizado usando o algoritmo BPTT, o qual empregou validação cruzada para a regularização do modelo, procedimento mostrado na figura 4.5. O menor erro quadrático médio para o modelo aplicado no conjunto de validação foi verificado na época de número 161, com o modelo associado a essa época sendo escolhido.

Figura 4.5: Validação cruzada realizada durante o treinamento da rede RMLP. A rede associada à época com menor erro no conjunto de validação é selecionada.



Fonte: autor.

4.2.2 Resultados

A avaliação da qualidade do modelo RMLP treinado foi realizada usando o conjunto de testes, para o modelo executando as

tarefas de simulação e predição de N instantes à frente. Pelo fato de o modelo RMLP apresentar uma arquitetura com realimentações internas, ele apresenta-se como uma opção interessante para a tarefa de simulação. Para a tarefa de predição, a rede RMLP apresenta desvantagens, já que não pode ser aplicada na configuração série-paralelo, por apresentar realimentações internas ao modelo.

O comportamento das saídas do modelo RMLP para o conjunto de validação em uma tarefa de simulação é mostrado na figura 4.6. Na figura 4.6a, é mostrada a porção do conjunto de testes em que a válvula de sucção está sendo excitada, com o modelo RMLP apresentando resultados satisfatórios. Para a porção em que a válvula de descarga é excitada, mostrada em 4.6b, a rede RMLP apresentou resultados destacadamente bons, sendo capaz de modelar com boa qualidade ambas as variáveis.

O bom resultado para a tarefa de simulação, verificado na figura 4.6, reflete-se nos indicadores de desempenho do modelo. A rede RMLP apresentou erro quadrático médio de 0,0101 para a sucção e 0,0049 para a descarga. Esses EQM resultam em um R^2 de 0,9405 associado à pressão de sucção, e 0,9728 associado à pressão de descarga.

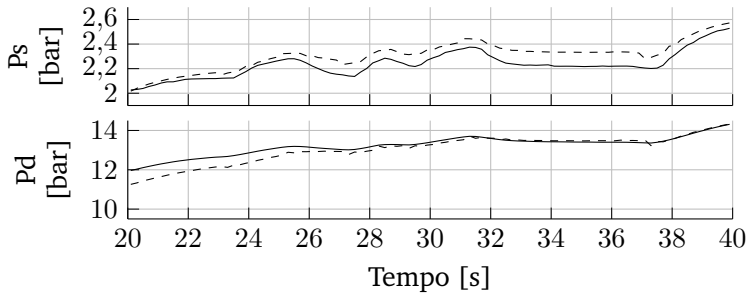
Os resultados para a tarefa de predição foram iguais aos apresentados para simulação. Isso acontece pela incapacidade da rede RMLP de operar em série-paralelo, fazendo com que esse modelo seja incapaz de beneficiar-se do conhecimento das saídas passadas do processo para realizar predições com maior exatidão.

4.3 REDE NEURAL ESN

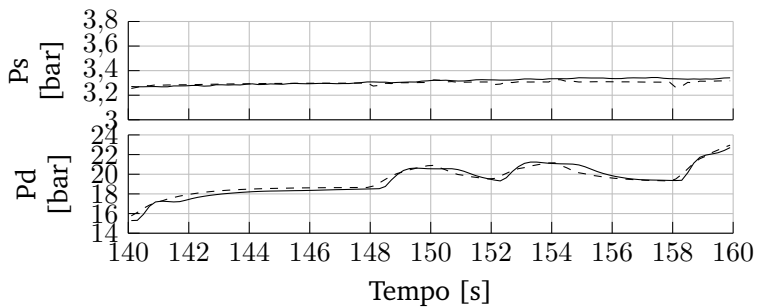
Em contraste aos modelos NARX e RMLP, que apresentam estruturas bem definidas, a rede ESN apresenta uma arquitetura altamente configurável, apresentando diversas conexões opcionais. A utilização dessas conexões é capaz de alterar o funcionamento geral do modelo, fazendo com que diferentes arquiteturas ESN possam se adaptar a tipos de problemas específicos.

A arquitetura completa da rede ESN, a qual é apresentada na figura 2.13, contempla todas as conexões possíveis no modelo. Essa arquitetura é raramente aplicada em sua plenitude, sendo que grande parte dos trabalhos na literatura utilizam arquiteturas simplificadas. Uma arquitetura específica, denominada nesse trabalho como ESN simplificada, é a mais comumente encontrada na literatura, servindo como um ponto de partida para arquiteturas mais

Figura 4.6: Resultados da rede RMLP para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo RMLP.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

Fonte: autor.

complexas, as quais utilizam uma ou mais das conexões opcionais, possibilitando uma melhora no desempenho geral do modelo para aplicações específicas. Os modelos foram criados utilizando o *toolbox Reservoir Computing* (SCHRAUWEN et al., 2007).

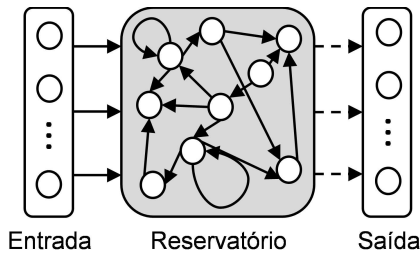
Nas seções subsequentes são apresentadas três arquiteturas distintas para a construção de um modelo ESN para o processo, a arquitetura ESN simplificada e sua extensão por meio de conexões adicionais. A seção 4.3.1 aborda a arquitetura simplificada, apresentando as simplificações dessa arquitetura e a configuração para a tarefa de identificação de sistemas. As seções 4.3.3 e 4.3.4 apresentam duas extensões da arquitetura ESN simplificada, que adicionam

realimentações, entre a saída e a própria saída ou entre a saída e o reservatório, respectivamente. Para cada uma dessas ESN é apresentada a construção do modelo e treinamento, que corresponde às etapas da metodologia de definição da estrutura e parâmetros do modelo, e também a validação do modelo.

4.3.1 Arquitetura ESN simplificada

A estrutura apresentada na figura 4.7 é a versão de ESN mais empregada na literatura. Ela é composta por um reservatório com unidades integradoras com vazamento, excitado por entradas externas, o qual é mapeado para saídas por um mecanismo de leitura.

Figura 4.7: Configuração da rede ESN sem realimentações externas.



Fonte: autor.

Esse modelo é capaz de preservar a informação do comportamento dinâmico da entrada através do estado interno de seu reservatório. A quantidade de memória armazenada depende das características construtivas desse reservatório, em especial de seu raio espectral e sua taxa de vazamento.

O equacionamento dessa arquitetura é dado por (os símbolos foram definidos na tabela 2.1):

$$\mathbf{x}_{\text{esn}}(t+1) = f((1-\alpha)\mathbf{x}_{\text{esn}}(t) + \alpha(\mathbf{W}_{\text{in}}^{\text{res}} \mathbf{u}(t) + \mathbf{W}_{\text{res}}^{\text{res}} \mathbf{x}_{\text{esn}}(t) + \mathbf{W}_{\text{bias}}^{\text{res}} \mathbf{1})), \quad (4.1)$$

$$\mathbf{y}_{\text{esn}}(t+1) = g(\mathbf{W}_{\text{res}}^{\text{out}} \mathbf{x}_{\text{esn}}(t+1) + \mathbf{W}_{\text{bias}}^{\text{out}} \mathbf{1}). \quad (4.2)$$

A equação (4.1) é denominada equação de estados, definindo o estado futuro do modelo. Por sua vez, a equação (4.2) é denominada equação de saída, mapeando o estado interno do reservatório na saída futura da rede. Esse par de equações pode ser representado na forma de um diagrama de blocos, o qual é ilustrado pela figura 4.8.

Por consistir em um modelo com recorrência interna, é esperado que exista uma dependência grande do histórico de amostras da entrada, sendo importante o ajuste do reservatório de forma a propiciar uma manutenção adequada dos seus estados passados, o que é obtido através do ajuste da taxa de vazamento.

O modelo é treinado através do ajustes da matriz de pesos conectados à saída do modelo, \mathbf{W}^{out} , que para a arquitetura em análise consiste em uma matriz com dois blocos, compostos pelas matrizes $\mathbf{W}_{\text{bias}}^{\text{out}}$ e $\mathbf{W}_{\text{res}}^{\text{out}}$, construída conforme a equação (4.3)

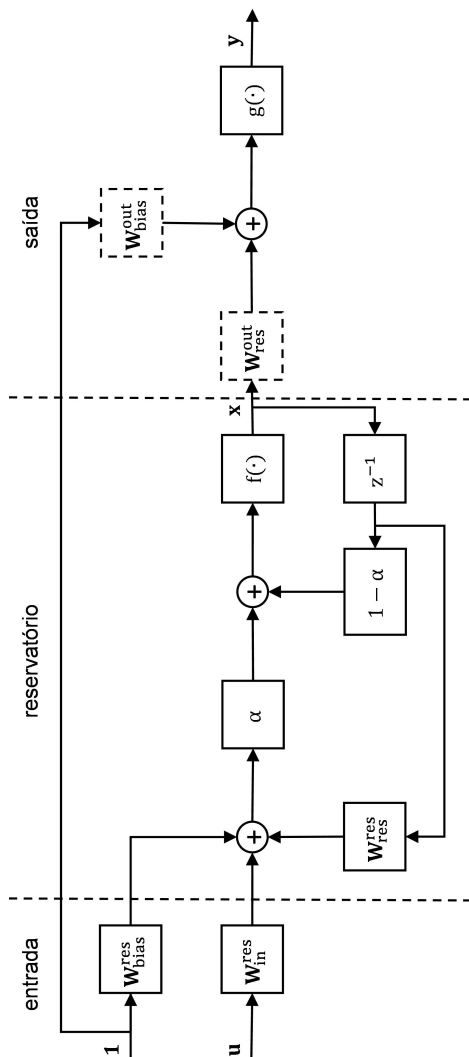
$$\mathbf{W}^{\text{out}} = \begin{bmatrix} \mathbf{W}_{\text{bias}}^{\text{out}} \\ \mathbf{W}_{\text{res}}^{\text{out}} \end{bmatrix}, \quad (4.3)$$

a qual pode ser ajustada através do método de regularização de Tikhonov, apresentado na seção 2.2.2.2.

Devido à sua arquitetura contendo apenas recorrências internas, a rede ESN simplificada se adequa bem à tarefa de simulação. Na tarefa de simulação, o modelo é executado em paralelo ao processo, não podendo tirar proveito do conhecimento de saídas passadas do processo. Por não apresentar realimentações externas, a rede ESN simplificada apresenta vantagens em relação às arquiteturas com recorrência externa. Isso decorre do fato das arquiteturas em que o modelo utiliza explicitamente as saídas passadas do processo como entrada necessitarem utilizar a própria saída do modelo realimentada, já que a saída real da planta não é conhecida em uma tarefa de simulação.

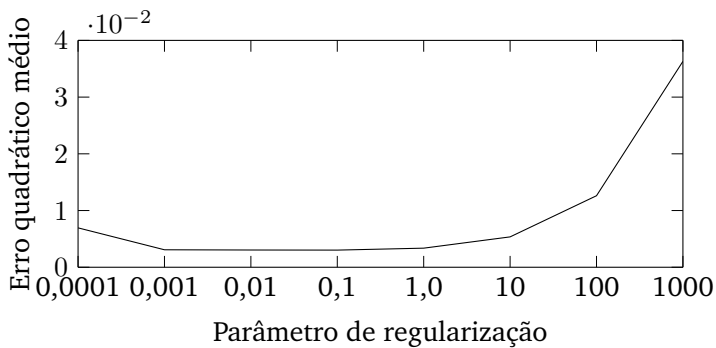
Em contrapartida, para a tarefa de predição, tanto de 1 quanto de N instantes à frente, a arquitetura ESN simplificada apresenta desvantagens em relação às arquiteturas com recorrência externa. Por não utilizar explicitamente as saídas passadas do processo, o modelo não pode ser executado na configuração série-paralelo, a qual é possível e vantajosa em tarefas de predição. Enquanto um modelo com realimentação externa se beneficia da operação em série-paralelo para fazer predições com uma precisão elevada, um modelo com recorrência interna não é capaz de se beneficiar disso, pois a saída atual e as passadas não são explicita-

Figura 4.8: Diagrama de blocos da rede ESN simplificada. Blocos tracejados representam as matrizes treináveis do modelo.



Fonte: autor.

Figura 4.9: Comportamento do erro quadrático médio associado a diferentes valores do parâmetro de regularização.



Fonte: autor.

mente informadas ao modelo.

4.3.1.1 Construção do modelo e treinamento

O primeiro passo para a construção de um modelo usando a arquitetura ESN simplificada, denotado simplificada neste trabalho como ESNS, consiste na determinação das entradas e saídas do modelo. Foram usados os mesmos vetores de entradas e saídas apresentados no desenvolvimento das demais redes.

O primeiro parâmetro que foi ajustado foi o de regularização, utilizado no algoritmo de treinamento, empregado para o ajuste dos pesos para as ESNs implementadas neste trabalho. É importante encontrar um valor adequado para esse parâmetro, e para isso foram treinadas redes com valores variados desse parâmetro, associados a um ajuste preliminar dos parâmetros do reservatório, os quais sofreram um ajuste refinado nos passos seguintes. Para esse experimento, um reservatório composto por 200 neurônios foi utilizado, ajustado com valores de 0,95 para a taxa de vazamento e 0,85 para o raio espectral, os quais são razoáveis para o problema. As redes treinadas tiveram seu desempenho testado no conjunto de validação, com o erro quadrático médio (média dos erros quadráticos médios individuais de cada saída do modelo) mostrado na figura 4.9.

É possível verificar que, para o problema em questão, o parâmetro de regularização induz resultados satisfatórios em uma faixa

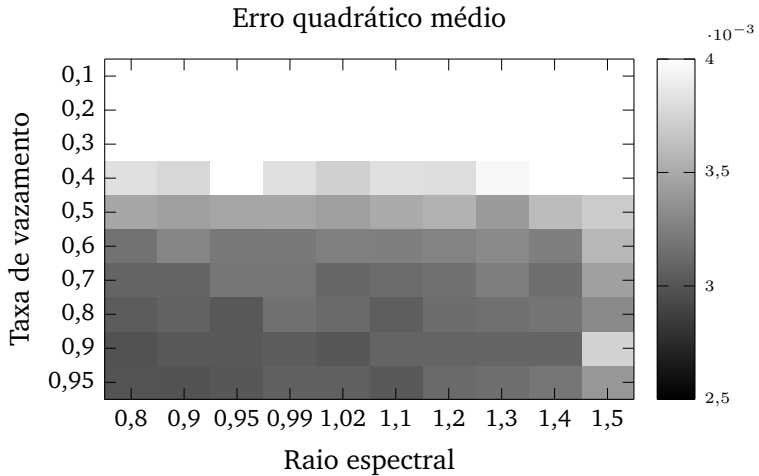
grande de valores, sendo que ajustes entre 0,001 e 1 apresentam resultados semelhantes. Valores muito pequenos fazem com que a rede sofra pouca regularização, fazendo com que ela possa apresentar grande sensibilidade a ruídos. Por sua vez, valores muito grandes fazem com que a rede não consiga se ajustar adequadamente ao problema. O valor selecionado foi 0,1, o qual é utilizado no decorrer deste trabalho para todas as implementações de ESN. A escolha desse valor resulta em modelos cuja resposta é rápida o suficiente para acompanhar as variações das dinâmicas modeladas, ao mesmo tempo que é suave, sendo capaz de filtrar o ruído. A escolha se mostrou adequada, apresentando resultados compatíveis com o desejável para a aplicação para os variados ajustes que são apresentados no decorrer do trabalho.

O número de neurônios do reservatório foi fixado, inicialmente, em 200 neurônios, para a etapa de ajuste do modelo.¹ Para a seleção dos parâmetros construtivos do reservatório foi utilizado um algoritmo de busca em grade. Esse algoritmo treina modelos variando os dois principais parâmetros do reservatório, a taxa de vazamento e o raio espectral. Para cada par de parâmetros foram treinados 10 modelos, sendo usada para a análise a média do desempenho dos modelos treinados, mostrada graficamente através da figura 4.10a. A taxa de vazamento foi testada para valores no intervalo de 0,1 até 0,95, enquanto o raio espectral foi testado para valores no intervalo de 0,8 até 1,5, com uma resolução mais fina para valores próximos a 1,0.

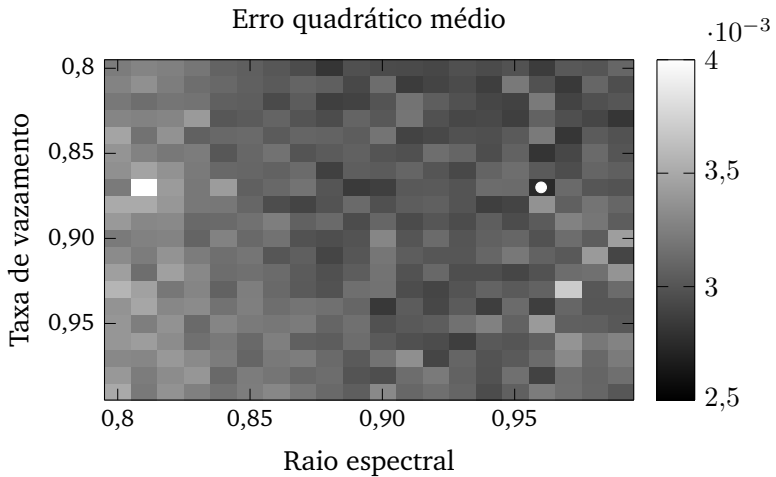
O desempenho dessa arquitetura é afetado sensivelmente pela escolha da taxa de vazamento. Valores menores que 0,5 resultam em desempenhos insatisfatórios. O raio espectral tem um impacto menos crítico, com melhores resultados induzido por valores inferiores a 1,1. Apesar de serem obtidos resultados satisfatórios com raios espectrais maiores que 1, a literatura recomenda a utilização de valores inferiores à unidade, de forma a garantir a propriedade de estado de eco. Devido a isso, foi realizada uma nova busca em grade, explorando apenas a região cujo desempenho apre-

¹A literatura apresenta tamanhos variados para o reservatório, com o entendimento que quanto maior o reservatório melhores será o desempenho alcançável, provendo métodos apropriados de regularização (LUKOŠEVIČIUS, 2012). São encontrados na literatura valores moderados, como 500 neurônios, em Antonelo et al. (2017), até valores grandes, como 20000 neurônios, em Triefenbach et al. (2010).

Figura 4.10: Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNS. Células mais escuras representam menores erros quadráticos médios. Círculo branco indica a célula com melhor desempenho.



(a) Busca em grade com variações alta dos parâmetros.



(b) Busca em grade na região de interesse.

Fonte: autor.

sentou melhores resultados. Foram testados valores entre 0,8 e 0,99, tanto para a taxa de vazamento, quanto para o raio espectral. A representação gráfica do resultado desse procedimento é mostrado na figura 4.10b.

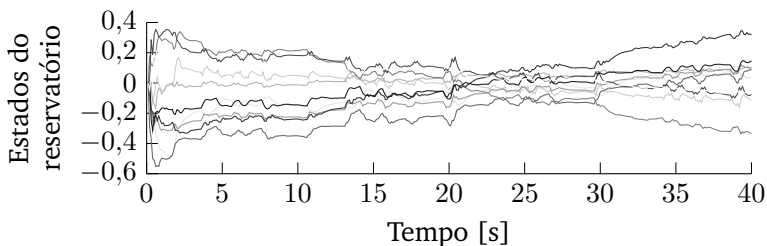
Foi selecionado para o modelo final, os valores de 0,96 para o raio espectral e 0,87 para a taxa de vazamento, por terem induzido o melhor desempenho no conjunto de validação. Para o modelo final, foi utilizado um reservatório com 500 neurônios, já que a literatura apresenta evidências de que reservatórios ajustados com um número fixo de neurônios e escalados para uma quantidade maior de neurônios apresentam, em geral, resultados tão bons ou melhores que os obtidos com a quantidade inicial (LUKOŠEVIČIUS, 2012).

O modelo ESN apresenta um mecanismo interessante para a identificação de alguns problemas no funcionamento do modelo. Através da verificação dos estados internos do reservatório, podem ser identificados problemas na construção do modelo e no ajuste dos parâmetros. Um exemplo é mostrado na figura 4.11. Na figura 4.11a é mostrado um reservatório “saudável”, verificado pela riqueza das dinâmicas e pela ausência de comportamentos instáveis e caóticos ou de estados saturados.

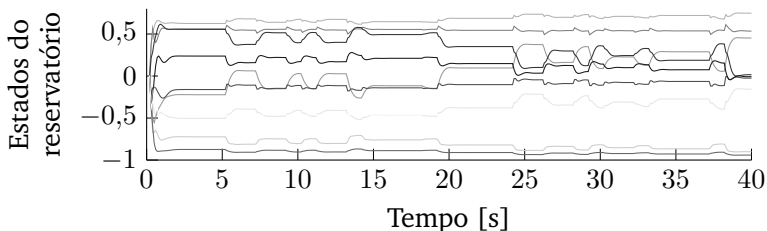
Por sua vez, os reservatórios apresentados nas figuras 4.11b, 4.11c e 4.11d apresentam anomalias indesejadas. No primeiro caso, a rede foi treinada sem a realização da normalização das entradas da rede, o que resultou em estados insensíveis a pequenas variações nas entradas da rede. Isso é causado por valores altos na entrada dos neurônios, fazendo com que opere na região de saturação da função de ativação. Esse comportamento pode ser verificado quando os estados operam oscilando entre dois valores fixos. O segundo reservatório anômalo foi treinado com uma taxa de vazamento inadequada para o problema. Isso resultou em estados com muita memória, os quais apresentam pouca sensibilidade às variações da entrada, resultando em estados com comportamento rico em dinâmicas de baixa frequência, mas ao mesmo tempo pobre de dinâmicas de alta frequência. No terceiro caso, é mostrado um reservatório com comportamento instável. Ele decorre do valor elevado do raio espectral, o qual introduziu comportamentos dinâmicos que crescem indefinidamente, que levou à instabilidade dos estados do reservatório. É importante notar que os estados não cresceram até o infinito devido à função de ativação dos neurônios do reservatório, a qual limita a resposta dos mesmos entre ± 1 .

Para o treinamento desses modelos, a matriz de pesos \mathbf{W}^{out}

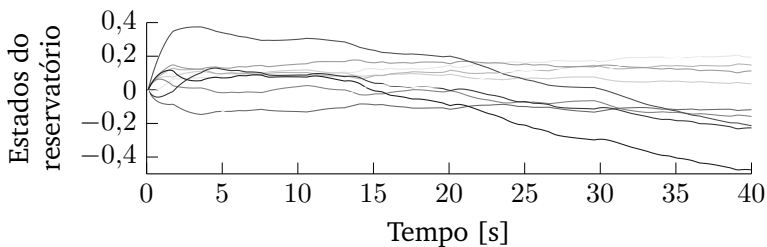
Figura 4.11: Estados do reservatório para diferentes redes ESN. Erros de projeto podem ser identificadas ao analisar o comportamento dos estados.



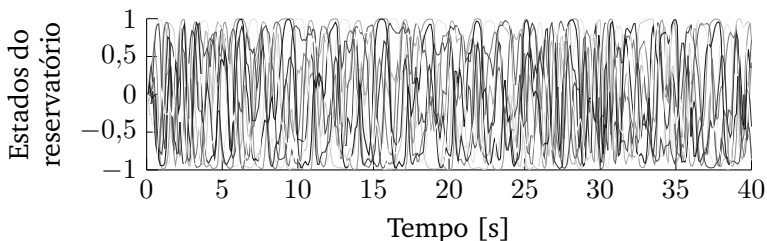
(a) Treinamento com parâmetros adequados.



(b) Treinamento com entradas não normalizadas.



(c) Treinamento com taxa de vazamento 0,1.



(d) Treinamento com raio espectral 1,8.

Fonte: autor.

foi ajustada utilizando a regularização de Tikhonov, com parâmetro de regularização fixado em 0,1.

4.3.1.2 Resultados

Assim como foi realizado para as demais arquiteturas, o modelo treinado foi avaliado para as tarefas de simulação e predição de N instantes à frente. Devido ao fato da arquitetura ESNS ser construída apenas com recorrências internas, ela sempre opera na configuração paralela, apresentando o mesmo funcionamento e resultados para as duas tarefas em questão.

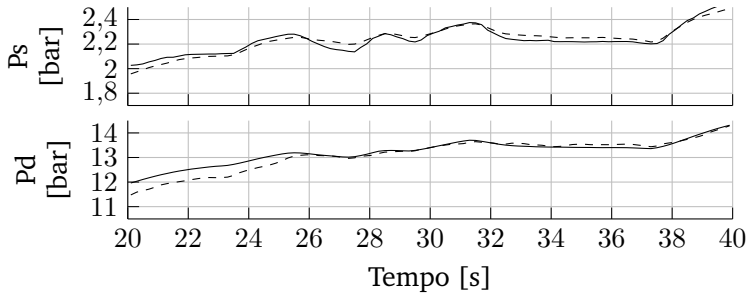
A avaliação do desempenho foi realizada no conjunto de testes, com o modelo sendo executado na configuração paralela, usando os valores reais da perturbação mensurável. A qualidade do modelo identificado pode ser verificada na figura 4.12, na qual o modelo ESN treinado é comparado com a saída real do processo. São apresentadas duas porções do conjunto de testes, correspondendo a instantes nos quais a válvula de sucção foi excitada, entre os instantes 20 s e 40 s do ensaio, e entre os instantes 140 s e 160 s, nos quais a válvula de descarga foi excitada.

O modelo apresenta resultados bons na primeira porção do ensaio, durante a excitação da válvula na sucção, mostrada na figura 4.12a. O comportamento apresentado é satisfatório para ambas as pressões, pois o modelamento, tanto da pressão de sucção, P_s , manipulada diretamente pela válvula na sucção, quanto do acoplamento entre a sucção e a descarga, foi correto. Na segunda porção do ensaio, período em que houve a excitação da válvula na descarga, mostrado na figura 4.12b, os resultados são satisfatórios. Apesar de o ensaio encontrar-se em sua porção final, o modelo apresentou boa capacidade de modelagem para a pressão de sucção e teve resultados satisfatórios para a descarga. É interessante notar que a escala da pressão de sucção, P_s , em 4.12b é reduzida, fazendo com que a amplitude do desvio entre o modelo e o alvo aparente seja pequena, comparada à variação da pressão de sucção durante o ensaio.

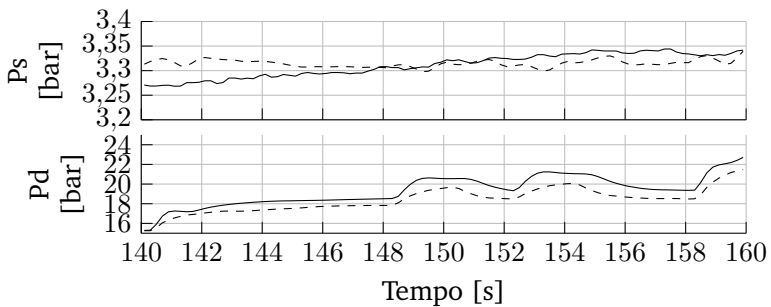
4.3.2 Arquiteturas com realimentações externas

A utilização de realimentações da saída é uma alternativa para expansão da arquitetura ESN simplificada, propiciando um mecanismo adicional de processamento temporal. Ao ser realimentada a saída, o modelo resultante tem tanto realimentação interna quanto externa. Isso faz com que o modelo incorpore algumas caracte-

Figura 4.12: Resultados da aplicação da rede ESN no conjunto de testes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNS.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

Fonte: autor.

terísticas inerentes aos modelos com recorrência externa, as quais não são verificadas na arquitetura ESN simplificada, com destaque à capacidade de utilização do modelo na configuração série-paralelo.

Essa adição ao modelo ESN é semelhante à estratégia utilizada pelas redes NARX e RMLP para realizarem processamento temporal. Em ambos os casos, um bloco construtivo é associado às realimentações externas de forma a realizar processamento temporal. No caso das redes NARX e RMLP, o bloco construtivo é um modelo estático, já no caso da rede ESN com realimentações externas, o bloco construtivo é um modelo dinâmico.

A presença de realimentações é capaz de modificar significati-

vamente o funcionamento do modelo. No contexto da identificação de sistemas, esse tipo de arquitetura faz com que o modelo utilize em seu vetor de regressão, φ , as saídas passadas do processo, em adição ao estado interno do reservatório, trazendo mudanças na utilização do modelo em tarefas de identificação.

Para a tarefa de simulação, existem possíveis desvantagens no uso de realimentações externas. Em uma tarefa de simulação o modelo opera em paralelo ao processo, fazendo com que não existam saídas reais do processo para alimentar o modelo, o qual é alimentado então por sua própria saída, de forma recorrente. Esse processo implica em alguns cuidados adicionais, em especial no treinamento do modelo, de forma a evitar possíveis problemas de estabilidade. Em geral, é recomendado utilizar no treinamento métodos de regularização, os quais são importantes para tornar os modelos com realimentação estáveis (LUKOŠEVIČIUS, 2012).

Sob um viés de predição, as arquiteturas ESN com recorrência externa beneficiam-se da possibilidade de operação na configuração série-paralela, podendo obter resultados mais consistentes. Isso acontece pela utilização das saídas passadas reais do processo. Se a tarefa for de predição de N instantes à frente, o modelo passa a utilizar a saída realimentada da rede, com o resultado se aproximando ao de simulação à medida que o valor de N aumenta.

4.3.3 ESN com realimentação para a saída

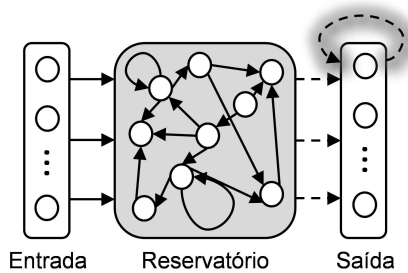
A primeira alternativa consiste no uso da realimentação da saída para a própria saída da rede. Como no modelo ESN todos os pesos ligados à camada de saída são treináveis, os pesos dessa realimentação também são treinados, em adição ao mecanismo de leitura do reservatório. Essa configuração é mostrada na figura 4.13.

A rede ESN com realimentação para a saída apresenta as seguintes equações para os estados, representados por $\mathbf{x}_{\text{esn-rs}}$, e para as saídas, representadas por $\mathbf{y}_{\text{esn-rs}}$:

$$\mathbf{x}_{\text{esn-rs}}(t+1) = f((1-\alpha)\mathbf{x}_{\text{esn-rs}}(t) + \alpha(\mathbf{W}_{\text{in}}^{\text{res}}\mathbf{u}(t) + \mathbf{W}_{\text{res}}^{\text{res}}\mathbf{x}_{\text{esn-rs}}(t) + \mathbf{W}_{\text{bias}}^{\text{res}}\mathbf{1})), \quad (4.4)$$

$$\mathbf{y}_{\text{esn-rs}}(t+1) = g(\mathbf{W}_{\text{res}}^{\text{out}}\mathbf{x}_{\text{esn-rs}}(t+1) + \mathbf{W}_{\text{bias}}^{\text{out}}\mathbf{1} + \mathbf{W}_{\text{out}}^{\text{out}}\mathbf{y}_{\text{esn-rs}}(t)). \quad (4.5)$$

Figura 4.13: Configuração da rede ESN com realimentação da saída com a saída.



Fonte: autor.

A equação (4.4), que define os estados da rede, é igual à equação de estados da rede ESN simplificada, apresentada em (2.9). Por sua vez, a equação de saída, (4.5), apresenta o termo adicional $\mathbf{W}_{\text{out}}^{\text{out}}\mathbf{y}(t)$, decorrente da realimentação que foi adicionada no modelo, conectando a saída com a própria saída, no qual $\mathbf{W}_{\text{out}}^{\text{out}}$ representa a matriz de realimentação, apresentando dimensão $\mathfrak{R}^{N_y \times N_y}$.

A figura 4.14 contém o diagrama de blocos que explicita o funcionamento da rede ESN com realimentação da saída para a própria camada de saída. Como todas as conexões ligadas à camada de saída são treinadas, a matriz \mathbf{W}^{out} deve ser escrita de forma a conter também os pesos $\mathbf{W}_{\text{out}}^{\text{out}}$:

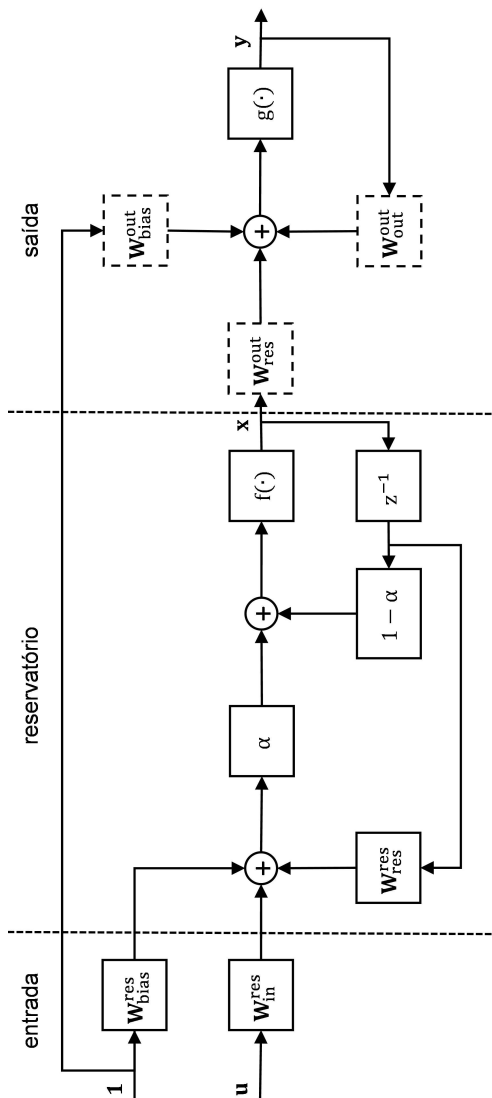
$$\mathbf{W}^{\text{out}} = \begin{bmatrix} \mathbf{W}_{\text{bias}}^{\text{out}} \\ \mathbf{W}_{\text{res}}^{\text{out}} \\ \mathbf{W}_{\text{out}}^{\text{out}} \end{bmatrix}. \quad (4.6)$$

A matriz \mathbf{X} , usada no treinamento, é formada pelos estados da rede, pelo bias e também pela saída realimentada, da seguinte forma:

$$\mathbf{X} = \begin{bmatrix} \mathbf{1} \\ \mathbf{X}_{\text{esn-rs}} \\ \mathbf{Y}_{\text{esn-rs}} \end{bmatrix}, \quad (4.7)$$

em que a matriz $\mathbf{1}$ apresenta dimensão $\mathfrak{R}^{1 \times T}$, a matriz $\mathbf{X}_{\text{esn-rs}}$

Figura 4.14: Diagrama de blocos da rede ESN com realimentação da saída para a saída. Blocos tracejados representam matrizes treináveis do modelo.



Fonte: autor.

apresenta dimensão $\mathfrak{R}^{N_x \times T}$ e a matriz $\mathbf{Y}_{\text{esn-rs}}$ apresenta dimensão $\mathfrak{R}^{N_y \times T}$.

O método recomendado para ajuste dessa matriz é o da regularização de Thikonov, apresentado na seção 2.2.2.2. Se o laço de realimentação for utilizado durante o treinamento, surgem alguns problemas, pois a matriz $\mathbf{Y}_{\text{esn-rs}}$ não é conhecida. Para lidar com o laço de realimentação, a estratégia denominada *teacher forcing* é usada. A ideia consiste em abrir o laço de realimentação, utilizando a saída desejada, ao invés da saída do modelo, durante a etapa de treinamento. Se for considerado que a saída alvo é aprendida com alta exatidão ($\mathbf{y}^{\text{alvo}}(n) \approx \mathbf{y}(n)$), o sistema recorrente se comportará de forma semelhante em sua operação real, quando usa $\mathbf{y}(n)$, ao comportamento durante o treinamento, fase na qual é usado $\mathbf{y}^{\text{alvo}}(n)$ (LUKOŠEVIČIUS, 2012).

É interessante notar que, apesar de ser adicionado um mecanismo de realimentação semelhante aos usados pelas redes NARX e RMLP, o treinamento continua sendo computacionalmente eficiente.

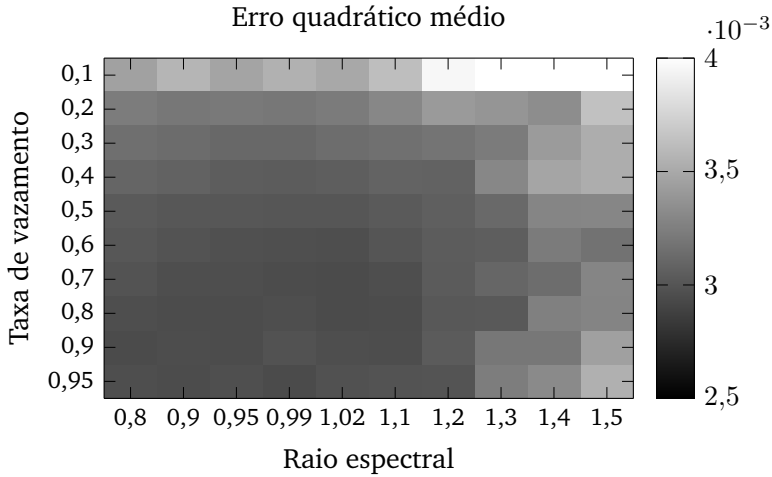
4.3.3.1 Construção do modelo e treinamento

O modelo ESN com realimentação da saída para a própria saída, o qual é denominado ESNRS, foi construído de forma semelhante ao modelo ESNS. Os vetores de entrada e saída são idênticos, bem como o tamanho do reservatório, fixado em 200 neurônios durante a etapa de escolha dos parâmetros.

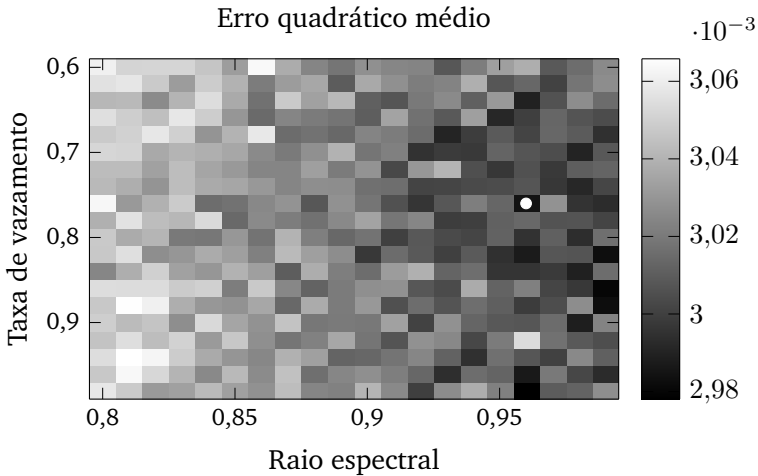
O mesmo procedimento para escolha dos parâmetros construtivos do reservatório, a busca em grade, foi utilizado. A figura 4.15a mostra os erros quadráticos médios para diferentes configurações do reservatório. A primeira etapa mostrou que o desempenho do modelo ESNRS é menos suscetível a valores pequenos da taxa de vazamento, apresentando melhores desempenhos em uma região mais extensa do plano de parâmetros. Devido a isso foi realizada uma segunda etapa de busca em grade, refinando os resultados em uma região menor, variando a taxa de vazamento entre 0,6 e 0,98, enquanto o raio espectral sofreu variações entre 0,8 e 0,99. Os resultados são mostrados na figura 4.15b, na qual o melhor desempenho é mostrado por um círculo branco. É interessante notar que o erro, em toda essa região explorada, sofre uma pequena variação, com desempenhos muito parecidos.

O modelo ESNRS final foi construído usando os valores associados à célula com melhor desempenho, consistindo em 0,76

Figura 4.15: Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNRS. Células mais escuras representam menores erros quadráticos médios.



(a) Busca em grade com variações alta dos parâmetros.



(b) Busca em grade na região de interesse.

Fonte: autor.

para a taxa de vazamento e 0,96 para o raio espectral. A rede final empregou um reservatório com 500 neurônios, que foi treinado, novamente, usando o método de regressão com regularização, empregando parâmetro de regularização fixado em 0,1. Para esse modelo em específico, é importante a utilização de métodos de regularização, pois, devido à presença do laço de realimentação externa, o modelo é mais suscetível a instabilização.

4.3.3.2 Resultados

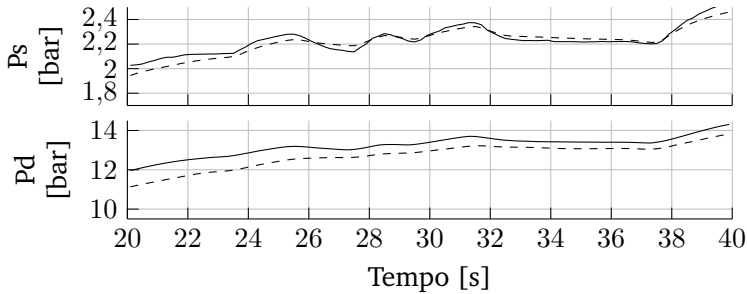
Assim como os demais modelos, o ESNRS foi testado realizando as tarefas de simulação e predição de N instantes à frente. Diferente do modelo ESNS, o ESNRS apresenta a um laço de realimentação externa, o que permite a execução do modelo na configuração série-paralela, possibilitando resultados mais exatos em tarefas de predição.

O resultado da utilização do modelo para um problema de simulação é mostrado na figura 4.16. O desempenho apresentado pelo modelo é satisfatório, já que foi possível o aprendizado das relações dinâmicas entre as variáveis, inclusive dos acoplamentos dinâmicos. Os resultados na porção inicial do ensaio, mostrada na figura 4.16a, comparado ao desempenho na porção próxima ao fim do ensaio de testes, mostrada na figura 4.16b, foram semelhantes, com ambos apresentando resultados satisfatórios.

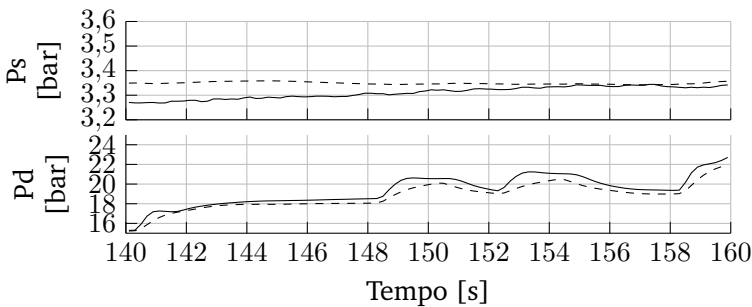
A grande melhoria no desempenho é verificada quando o modelo é testado na configuração série-paralelo, em uma tarefa de predição. Na figura 4.17 é apresentado o comportamento da rede ESNRS em uma tarefa de predição de N instantes à frente. Nessa tarefa, o modelo é executado em série-paralelo até o instante anterior ao início da predição, a partir do qual o modelo passa a operar em paralelo ao sistema. Os instantes iniciais foram 20 s para a figura 4.22a e 140 s para a figura 4.22b. A partir desse instante inicial, o modelo conhece apenas as entradas futuras, necessitando utilizar a saída do modelo realimentada, realizando 100 predições, iniciadas nesse instante inicial.

A operação em série-paralelo proporciona melhorias na qualidade do modelo durante os primeiros instantes da predição. Por sua vez, à medida que o número de amostras preditas cresce, o desempenho do modelo tende ao apresentado durante a tarefa de predição. Esse comportamento pode ser verificado, por exemplo, na figura 4.17b, na qual a pressão de descarga P_s modelada pela rede ESNRS apresenta erro muito pequeno durante os instantes iniciais

Figura 4.16: Resultados da rede ESNRS para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa as saída para o modelo ESNRS.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

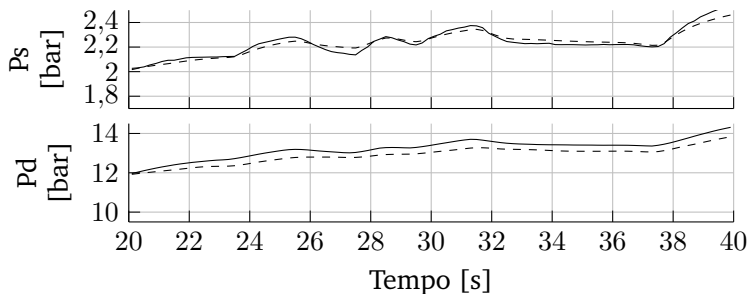
Fonte: autor.

da predição, o qual, após algumas amostras, retorna a um patamar próximo ao apresentado na tarefa de simulação.

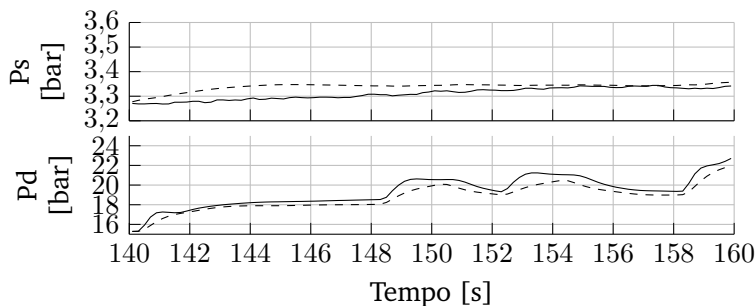
4.3.4 ESN com realimentação para o reservatório

A segunda opção para adição de realimentação externa ao modelo ESN consiste na realimentação da saída para o reservatório. Através de tal adição, a informação armazenada nas saídas passadas também é processada pelo reservatório, incorporando-se no estado interno do mesmo. Isso faz com que as saídas passadas passem também pela expansão não linear realizada pelo reservatório, e, da mesma forma que as entradas da rede, tenham seu comportamento

Figura 4.17: Resultados da rede ESNRS para a tarefa de predição de N instantes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRS.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

Fonte: autor.

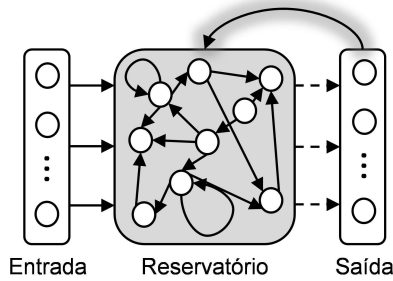
dinâmico armazenado pelo reservatório. A arquitetura é ilustrada na figura 4.18.

O modelo ESN com realimentação de sua saída para o reservatório é descrito pelas equações (4.8) e (4.9), que regem o estado interno do reservatório e saída da rede, respectivamente.

$$\mathbf{x}_{esn-rr}(t+1) = f((1-\alpha)\mathbf{x}_{esn-rr}(t) + \alpha(\mathbf{W}_{in}^{res}\mathbf{u}(t) + \mathbf{W}_{res}^{res}\mathbf{x}_{esn-rr}(t) + \mathbf{W}_{bias}^{res}\mathbf{1} + \mathbf{W}_{out}^{res}\mathbf{y}_{esn-rr}(t))) \quad (4.8)$$

$$\mathbf{y}_{esn-rr}(t+1) = g(\mathbf{W}_{res}^{out}\mathbf{x}_{esn-rr}(t+1) + \mathbf{W}_{bias}^{out}\mathbf{1}) \quad (4.9)$$

Figura 4.18: Configuração da rede ESN com realimentação da saída com o reservatório.



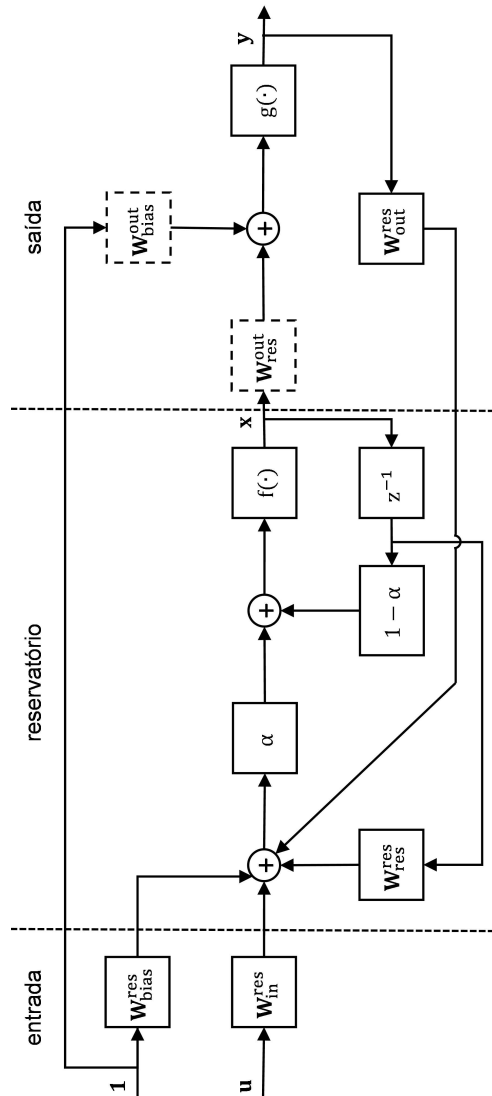
Fonte: autor.

Nesse caso, a equação de estados apresenta um elemento adicional, $\mathbf{W}_{\text{out}}^{\text{res}} \mathbf{y}_{\text{esn-rr}}(t)$, o qual adiciona o comportamento da saída da rede como entrada no reservatório. A matriz $\mathbf{W}_{\text{out}}^{\text{res}}$ define os pesos da realimentação e apresenta dimensão $\Re^{N_y \times N_x}$. A equação de saída é igual à apresentada para a ESN simplificada, mostrada na equação (2.10).

Um diagrama de blocos do funcionamento da ESN com realimentação para o reservatório é mostrado na figura 4.19. O termo que pondera a realimentação entre a saída e o reservatório, $\mathbf{W}_{\text{out}}^{\text{res}}$, modifica o funcionamento da equação de estado, como ilustrado na equação (4.8). O estado do reservatório passa a depender dos seus estados passados, das entradas atuais e também das saídas atuais do modelo.

O treinamento do modelo é realizado da forma semelhante ao da rede ESN com realimentação da saída para a camada de saída. É usado *teacher forcing*, eliminando o laço de realimentação externa ao utilizar a saída desejada durante o treinamento. A matriz \mathbf{W}^{out} que deve ser ajustada é a mesma da rede ESNS, apresentada na equação (4.3), dado que os pesos das conexões com sentido ao reservatório não são treinados em ESNs.

Figura 4.19: Diagrama de blocos da rede ESN com realimentação da saída para o reservatório. Blocos tracejados representam matrizes treináveis do modelo.



Fonte: autor.

4.3.4.1 Construção do modelo e treinamento

O desenvolvimento de um modelo ESN com realimentação da saída para o reservatório, tratado como ESNRR, é muito semelhante ao apresentado para o modelo ESNRS. O procedimento usado para o ajuste dos parâmetros construtivos do reservatório é o mesmo utilizado para as demais arquiteturas de ESN. Foi realizada a busca em grade no espaço de parâmetros, cujo resultado é ilustrado na figura 4.20a. De forma semelhante à rede ESNRS, valores menores da taxa de vazamento passam a apresentar desempenhos satisfatórios, mas, novamente, a região com melhores desempenhos foi a de alta taxa de vazamento e raio espectral menor que 1. Com base nesses dados, foi realizada uma segunda busca em grade, testando valores na região em que a rede apresentou melhores resultados. A taxa de vazamento sofreu variações entre 0,6 e 0,98, enquanto o raio espectral foi variado na faixa entre 0,8 e 0,99. Uma representação gráfica do resultado é mostrada na figura 4.20b.

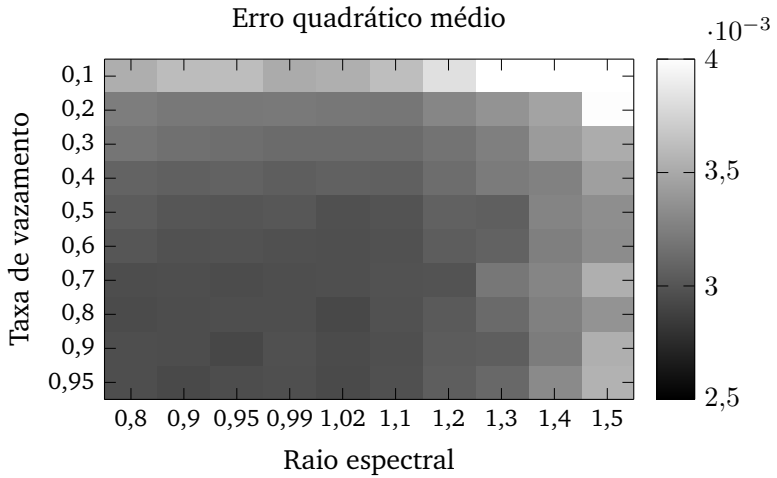
A célula com melhor desempenho teve seus parâmetros escolhidos para o modelo ESNRR final. Os parâmetros escolhidos foram, taxa de vazamento 0,92 e raio espectral 0,84. O modelo final foi construído com reservatório composto por 500 neurônios e o treinamento do mecanismo de leitura do reservatório foi realizado utilizando o valor de 0,1 para o parâmetro de regularização.

4.3.4.2 Resultados

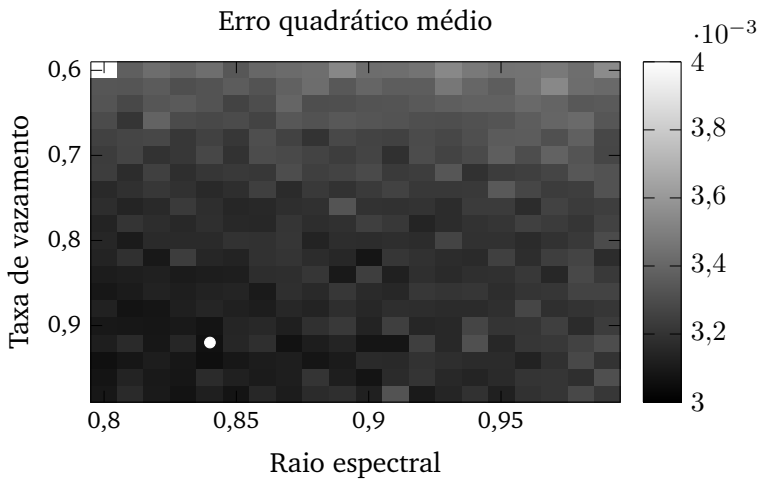
O desempenho modelo ESNRR foi testado através dos experimentos que o testaram executando as tarefas de simulação e predição. Os resultados apresentados pelo modelo no conjunto de teste foram satisfatórios. O modelo apresentou boa capacidade para o modelamento das dinâmicas, apresentando resultados semelhantes ao da rede ESNRS, porém com diferenças pontuais no seu desempenho. O comportamento do modelo em uma tarefa de simulação é mostrado na figura 4.21. É possível verificar um desempenho adequado em ambas as porções do conjunto de teste, com destaque para a pressão de sucção, P_s , mostrada na figura 4.21b, a qual apresentou resultados melhores que os demais modelos ESN.

O experimento que avaliou o desempenho do modelo ESNRR em uma tarefa de predição de N instantes à frente mostrou que, como o esperado, o modelo se beneficiou da operação em série-paralelo para realizar predições com maior exatidão, durante os instantes iniciais da processo de predição. O comportamento é se-

Figura 4.20: Busca em grade para escolha da melhor configuração de taxa de vazamento e raio espectral para rede ESNRR. Células mais escuras representam menores erros quadráticos médios.



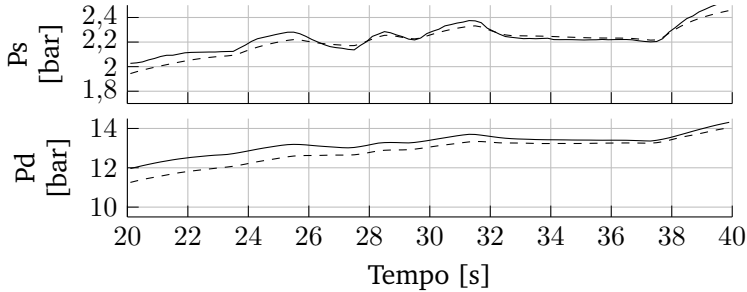
(a) Busca em grade com variações alta dos parâmetros.



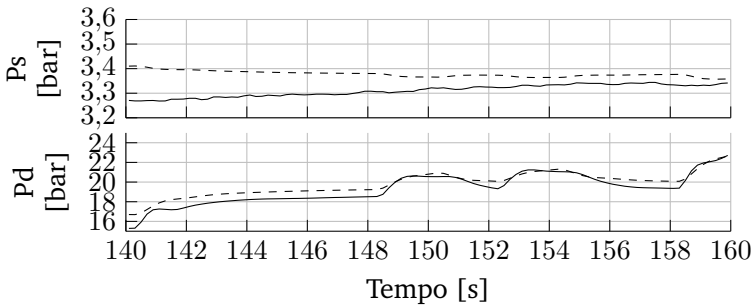
(b) Busca em grade na região de interesse.

Fonte: autor.

Figura 4.21: Resultados da rede ESNRR para a tarefa de simulação. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRR.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

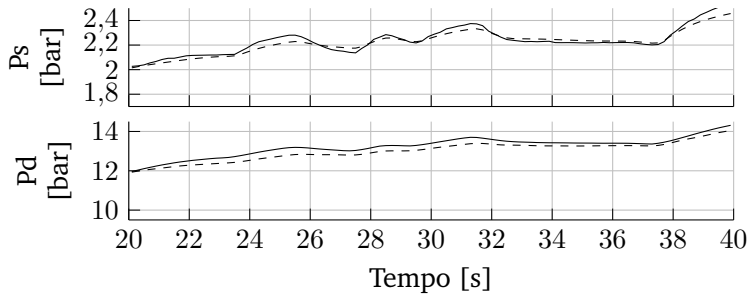
Fonte: autor.

melhante ao apresentado pela rede ESNRS, que também obteve tal vantagem, o que não se verificou para a rede ESN.

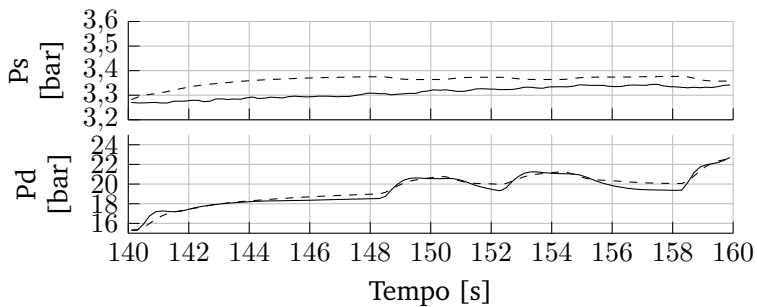
4.4 COMPARAÇÃO DOS RESULTADOS

No decorrer deste trabalho, foram desenvolvidos cinco modelos para a resolução do problema apresentado como estudo de caso. Na seção 4.1 é apresentado o desenvolvimento de um modelo empregando a rede NARX; na seção 4.2 é apresentado o desenvolvimento de um modelo empregando a rede RMLP; enquanto na seção 4.3 é apresentado o desenvolvimento de modelos empre-

Figura 4.22: Resultados da rede ESNRR para a tarefa de predição de N instantes. Linha sólida representa a saída real. Linha tracejada representa a saída para o modelo ESNRR.



(a) Resposta a variação da válvula de sucção



(b) Resposta a variação da válvula de descarga

Fonte: autor.

gando ESN, através de uma variante simplificada (ESNS), uma variante com realimentação da saída para a saída (ESNRS) e uma com realimentação da saída para o reservatório (ESNRR). Os resultados obtidos pelos cinco modelos são comparados nesta seção. Esses modelos foram treinados usando o mesmo conjunto de treinamento e foram testados no mesmo conjunto de testes, em experimentos para a realização de tarefas de simulação e predição.

A primeira comparação é a do tempo de treinamento das diferentes arquiteturas, o qual é mostrado na tabela 4.1. Os modelos NARX e RMLP apresentaram tempos de treinamento na ordem das centenas de segundos, sendo que cada época do treinamento

demandou em média 3 s. Apesar de NARX e RMLP terem apresentado tempos de treinamento da mesma ordem de grandeza, a rede NARX teve um tempo de treinamento inferior, pois o número total de épocas de treinamento é variável, decorrente do uso de validação cruzada. Por sua vez, as três arquiteturas de ESN apresentaram tempos totais de treinamento consideravelmente menores, inferiores a meio segundo, o que representa três ordens de grandeza menores que os tempos observados para as outras duas arquiteturas.

Tabela 4.1: Tempo médio de treinamento para as arquiteturas de redes.

	NARX	RMLP	ESNS	ESNRS	ESNRR
Tempo de treinamento (s)	319	524	0,42	0,39	0,43

A grande diferença de tempos de treinamento entre as duas classes de RNAs empregadas neste trabalho era esperada. O treinamento de redes recorrentes derivadas do modelo MLP apresenta uma função custo não linear, necessitando de um algoritmo iterativo, o qual é computacionalmente custoso. Por sua vez, o treinamento das arquiteturas ESN consiste em um problema de otimização cuja função custo é afim, possibilitando a obtenção de uma solução analítica, fazendo o custo computacional ser drasticamente inferior ao apresentado pelas redes NARX e RMLP.

Outro aspecto relevante é a quantidade de parâmetros do modelo, pois ela define a complexidade do modelo resultante e é um indicador do custo computacional que cada modelo demandará, para a utilização do mesmo em sua tarefa fim. As arquiteturas NARX e RMLP apresentam morfologia semelhante, com número de parâmetros dependente da quantidade de neurônios, camadas e laços de realimentação. A rede NARX apresenta 10 neurônios e 117 pesos sinápticos, enquanto a rede RMLP apresenta 12 neurônios e 174 pesos sinápticos. A arquitetura ESN, por sua vez, tem número de parâmetros dependente da dimensão do reservatório escolhido, sendo que as instâncias utilizadas neste trabalho contém 502 neurônios, em um total de 4002 pesos sinápticos. Com isso, a rede ESN apresenta maior complexidade na sua utilização após o treinamento, o que pode se configurar em uma desvantagem em aplicações na qual o tempo de cômputo do modelo é crítico. Para os modelos desenvolvidos neste trabalho, os tempos médios e máximos para a realização de uma predição de 500 instantes usando cada modelo é mostrada

na tabela 4.2. Foram realizadas 1000 séries de 500 predições para cada modelo, sendo mostrados na tabela os tempos médios e máximo, em segundos. Os modelos que implementam redes ESN apresentaram tempos médios para a predição de 500 amostras na ordem de 10 vezes maiores, resultado esperado considerando a quantidade maior de neurônios e parâmetros do modelo.

Tabela 4.2: Tempos médio e máximo para a utilização dos modelos treinados. Resultados representam tempos para a execução de 500 predições por cada modelo.

	NARX	RMLP	ESNS	ESNRS	ESNRR
Tempo médio (s)	0,043	0,055	0,312	0,320	0,350
Tempo máximo (s)	0,085	0,074	0,398	0,538	0,483

A comparação do desempenho dos modelos para a tarefa de simulação é apresentada na tabela 4.3. Os resultados de todos modelos foram satisfatórios, com destaque para os modelos com recorrência interna. O único modelo com apenas recorrência externa, o NARX, apresentou o erro quadrático médio mais elevado. Entre os modelos derivados da rede MLP, a rede RMLP apresentou resultados superiores aos da NARX, apresentando o melhor desempenho para a modelagem da pressão de descarga entre todos os modelos.

Tabela 4.3: Soma dos erros quadráticos para as diversas configurações de RNA testadas para simulação.

		NARX	RMLP	ESNS	ESNRS	ESNRR
EQM (10^{-2})	Sucção	2,29	1,01	1,31	0,74	0,71
	Descarga	0,70	0,49	0,81	0,64	0,63
	Média	1,50	0,75	1,06	0,69	0,67

Na comparação entre as arquiteturas ESN, os modelos que apresentam conexões adicionais, apresentando um modelo híbrido com realimentações internas e externas, ESNRS e ESNRR, apresentaram desempenho superior quando comparados à estrutura ESNS, que apresenta apenas realimentações internas. O melhor desempenho médio para a tarefa de simulação, entre todos modelos, foi apresentado pela ESNRR.

Devido ao fato de a rede RMLP ter apresentado o melhor resultado para o modelamento da pressão de descarga, e das arquite-

turas ESN terem apresentado os melhores resultados para o modelamento da pressão de sucção, em aplicações específicas, poderiam ser utilizados modelos distintos para cada pressão. Tal estratégia resultaria em uma solução mais complexa, porém com erros médios inferiores aos apresentados por uma solução única para ambas as pressões.

Para a comparação dos modelos em uma tarefa de predição de N instantes à frente, foram realizados experimentos com diferentes horizontes de predição. Os modelos foram testados em horizontes de 1, 5 e 100 amostras. Cada modelo executou 12 séries de predição, realizadas a partir de instantes iniciais que variaram entre os instantes 20 s e 140 s do ensaio, iniciando em intervalos separados por 10 s. Na tabela 4.4 são mostrados os resultados médios das séries de predição para cada horizonte.

Tabela 4.4: Soma dos erros quadráticos para as diversas configurações de RNA testadas para a predição de N instantes à frente.

(a) Horizonte $N = 1$

		NARX	RMLP	ESNS	ESNRS	ESNRR
EQM (10^{-3})	Sucção	0,182	7,59	2,20	0,0264	0,0269
	Descarga	0,192	2,22	2,10	0,0040	0,0026
	Média	0,187	4,90	2,15	0,0152	0,0147

(b) Horizonte $N = 5$

		NARX	RMLP	ESNS	ESNRS	ESNRR
EQM (10^{-3})	Sucção	0,407	8,54	2,30	0,122	0,114
	Descarga	0,332	2,22	2,10	0,047	0,038
	Media	0,246	5,38	2,20	0,085	0,075

(c) Horizonte $N = 100$

		NARX	RMLP	ESNS	ESNRS	ESNRR
EQM (10^{-3})	Sucção	14,62	7,60	1,82	1,62	2,02
	Descarga	4,04	2,19	1,79	1,01	1,20
	Media	9,33	4,89	1,80	1,31	1,61

Na tabela 4.4a é mostrado o desempenho para um horizonte de 1 instante à frente. Os dois modelos que não apresentam realimentações externas, RMLP e ESNS, não se beneficiam da

execução em uma configuração série-paralelo, apresentando os erros com a mesma ordem de grandeza que os apresentados para a tarefa de simulação. Por sua vez, os modelos que apresentam realimentações externas conseguiram se beneficiar da execução em série-paralelo, apresentando resultados muito superiores aos apresentados na tarefa de simulação. O destaque é para os modelos ESNRS e ESNRR, os quais apresentaram os menores erros. É interessante notar que, apesar de os modelos RMLP e ESNS não terem se adequados bem à tarefa de predição, é possível melhorar os resultados se o modelo for executado na configuração paralela e as predições forem corrigidas utilizando o erro entre a saída atual e a predição inicial no instante anterior.

Nas tabelas 4.4b e 4.4c são apresentados os resultados para os horizontes de 5 e 100 amostras, respectivamente. Enquanto os modelos RMLP e ESNS apresentam desempenhos semelhantes em todos os horizontes, com valores próximos aos apresentados na tarefa de simulação, os modelos NARX, ESNRS e ESNRR apresentam resultados melhores, que por sua vez pioram à medida que o horizonte testado aumenta. Isso era esperado, já que para horizontes grandes, a tarefa de predição se assemelha à de simulação. Para o horizonte 5, as redes ESNRS e ESNRR ainda apresentam resultados muito superiores. Por sua vez, para o horizonte de 100 amostras, os desempenhos de todos os modelos apresentam a mesma ordem de grandeza, com valores semelhantes aos de simulação. O destaque é para os modelos ESNRS e ESNRR, que apresentam comportamentos semelhantes e os menores erros médios.

Os resultados gerais foram sumarizados na figura 4.23, na qual as características identificadas para cada arquitetura de rede neural foram comparadas. A comparação foi realizada de forma qualitativa, por meio de uma comparação entre os resultados de cada arquitetura e evidenciando as vantagens de cada uma delas. Para os quatro primeiros parâmetros, o desejável são valores baixos, por se tratarem do EQM para as tarefas de predição e simulação, e também de quantidade de parâmetros e tempos de treinamento e execução dos modelos. Para os dois últimos critérios, é desejável valores altos, pois tratam de características como versatilidade de configuração e facilidade de ajuste da estrutura.

Figura 4.23: Comparação entre as arquiteturas de redes recorrentes.

	NARX	RMLP	ESN
Simulação (EQM)	Alto	Médio	Baixo
Predição (EQM)	Médio	Alto	Baixo
Quantidade de parâmetros	Médio	Médio	Alta
Tempo para treinamento	Alto	Alto	Baixo
Tempo para realização das predições	Baixo	Baixo	Alto
Versatilidade de configuração	Baixa	Baixa	Alta
Facilidade de ajuste da estrutura	Baixa	Baixa	Alta

Fonte: autor.

5 CONSIDERAÇÕES FINAIS

Este capítulo final apresenta as conclusões gerais para o trabalho e as sugestões para trabalhos futuros. A seção 5.1 elenca as contribuições do trabalho para a literatura e também as conclusões gerais para o problema explorado no trabalho. São apresentadas também as conclusões específicas, tanto relacionadas a cada modelo de rede neural artificial, quanto inerentes ao estudo de caso em que foram empregadas. Por sua vez, a seção 5.2 detalha as sugestões para o prosseguimento da pesquisa e a realização de trabalhos futuros.

5.1 CONCLUSÕES

Em um escopo geral, o trabalho trouxe algumas contribuições que podem ser destacadas. O estudo da literatura acerca do emprego de redes neurais artificiais recorrentes em tarefas de identificação mostrou que essa classe de técnicas é empregada na literatura com sucesso em diversos exemplos. Os resultados experimentais, obtidos através do estudo de caso em um processo não linear e multivariável real, apresentam contribuições à literatura ao empregar diversas arquiteturas de RNA com sucesso e ao estabelecer um caso real comum para comparação dessas arquiteturas. Todas as arquiteturas empregadas apresentaram resultados bons, sendo que algumas se adequaram de forma melhor a tarefas específicas, o que é discutido em detalhes nos próximos parágrafos.

Algumas conclusões específicas sobre a metodologia e as técnicas aplicadas podem ser feitas. Acerca da metodologia empregada no trabalho, é possível concluir que o emprego da mesma, para a identificação de sistemas dinâmicos não lineares, se mostrou efetivo, com tal metodologia se adequando bem à identificação empregando redes neurais artificiais recorrentes. Essa metodologia serviu de base para todo o processo de construção dos modelos, possibilitando uma adequada comparação entre o desempenho dos mesmos.

O desenvolvimento de um modelo empregando a rede neural NARX se mostrou desafiador. Essa é a arquitetura de RNA comumente empregada na literatura e, como o esperado, o modelo desenvolvido usando essa arquitetura apresentou resultados satisfatórios. O grande desafio do desenvolvimento foi o ajuste dos parâmetros estruturais do modelo, processo que apresentou limitações devido, principalmente, ao alto custo computacional do treinamento dessa classe de arquiteturas, realizado com o algoritmo BPTT. Esse alto

custo computacional se reflete no tempo total de treinamento dos modelos, fazendo com que a utilização de arquiteturas com muitos neurônios ou camadas escondidas seja desencorajada e que a escolha de estruturas ótimas para a aplicação seja uma tarefa desafiadora.

A rede RMLP apresentou bons resultados para a tarefa de simulação, próximos aos obtidos pelas arquiteturas baseadas no modelo ESN, sendo que a rede RMLP apresentou o melhor resultado individual para a pressão de descarga. Por sua vez, a rede RMLP mostrou-se a pior escolha em tarefas de predição, com resultados expressivamente piores que os demais modelos, o qual se acentua para horizontes de predição pequenos. O desenvolvimento de um modelo usando a rede RMLP apresenta os mesmos desafios encontrados pelo modelo NARX, em especial a alta complexidade do treinamento, a qual pode desencorajar o uso de tal técnica em problemas multivariáveis com grande quantidade de entradas e saídas.

O paradigma da computação de reservatório, neste trabalho representado pelas implementações de ESNs, apresentou diversas vantagens quando comparado ao paradigma de redes recorrentes derivadas do modelo MLP, representado pelas redes NARX e RMLP. A vantagem mais evidente é a complexidade do treinamento dos modelos, a qual, para o estudo de caso deste trabalho, se traduziu em tempos de treinamento da ordem de mil vezes menor, para modelos ESN comparados com modelos NARX e RMLP. Esse resultado é expressivo, pois, apesar da maior complexidade do modelo ESN, empregando estrutura com 500 neurônios, o tempo de treinamento foi significativamente menor que o verificado para os modelos NARX e RMLP, que empregaram poucos neurônios em suas camadas escondidas, não passando de 10 neurônios. Em contrapartida, o tempo demandado para a execução do modelo após treinado é maior, quando comparado aos apresentados pelas redes NARX e RMLP, o que pode desencorajar a utilização em problemas com restrições temporais.

O trabalho empregou três estruturas ESN: a estrutura ESN usualmente encontrada na literatura, a qual foi denominada ESN simplificada, e duas estruturas contando com realimentações da saída para a própria saída ou para o reservatório. A primeira arquitetura apresentou resultados satisfatórios para a tarefa de simulação, porém, por não apresentar realimentação externa, os resultados para a tarefa de predição foram piores que os apresentados pelas outras duas estruturas ESN. Essas duas estruturas com realimentação, por sua vez, apresentaram os melhores resultados tanto para a ta-

refa de simulação, quanto para a de predição. A arquitetura ESN com realimentação para o reservatório foi a que apresentou, no geral, os melhores resultados, os quais foram muito próximos aos apresentados pela ESN com realimentação para a saída.

De acordo com as conclusões apresentadas, pode ser afirmado que os objetivos do trabalho foram plenamente alcançados. As principais arquiteturas de RNA recorrentes foram identificadas na revisão da literatura e foram empregadas com sucesso para um estudo de caso. Os modelos obtidos foram comparados em dois contextos diferentes, o das tarefas de simulação e predição, e as melhores estruturas para cada tarefa foram identificadas.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

Diversas sugestões para trabalhos futuros são pertinentes. Em um contexto geral, é interessante a pesquisa acerca da implementação de métodos de treinamento específicos para a tarefa de predição. Neste trabalho, todos os modelos foram treinados empregando técnicas usuais, utilizando métodos que se adequam melhor para tarefas de simulação. É sugerido para trabalhos futuros o estudo de formas alternativas para a definição da função custo que é otimizada pelo algoritmo de treinamento, a qual poderia ser construída de forma a refletir melhor o erro encontrado em tarefas de predição de N instantes à frente.

A comparação dos resultados obtidos neste trabalho com técnicas clássicas de identificação de sistemas dinâmicos não lineares é uma sugestão natural. O desenvolvimento de um modelo Hammerstein, por exemplo, possibilitaria uma avaliação mais completa das qualidades e limitações das técnicas abordadas neste trabalho.

Outra sugestão geral é a avaliação de uma arquitetura de RNA recorrente denominada memória de longo curto prazo (LSTM, do inglês *Long Short-Term Memory*). Essa rede vem sendo aplicada na literatura para diversos problemas com contexto temporal, obtendo grande sucesso e, por isso, apresentando potencial para a aplicação na identificação de sistemas dinâmicos (HOCHREITER; SCHMIDHUBER, 1997).

Sob um ponto de vista mais específico, cada modelo é passível de melhorias individuais. Para os modelos NARX e RMLP é sugerida a avaliação da utilização de uma nova função de ativação para o modelo de neurônio artificial utilizado, denominada unidade linear retificadora (ReLU, do inglês *Rectifier Linear Unit*), a qual está

sendo utilizada com sucesso na literatura para redes neurais profundas e também para RNAs recorrentes. Essa função de transferência é usada, na literatura, buscando amenizar o fenômeno do desaparecimento do gradiente, propiciando a utilização de arquiteturas mais profundas e possivelmente melhorando os resultados (NAIR; HINTON, 2010; MAAS et al., 2013).

Para os modelos ESN são sugeridas, também, melhorias. Neste trabalho, foi utilizado para todas as implementações de ESN reservatório com taxa de vazamento única. Uma melhoria sugerida é a utilização de reservatório com mais de uma taxa de vazamento, o que pode ser interessante para a identificação de processos multivariáveis com dinâmicas diferentes, como o do estudo de caso. Taxas de vazamento diferentes podem fazer com que processos cujas dinâmicas apresentam tempos de acomodação diferentes sejam mais bem modelados.

Por fim, uma sugestão natural é a aplicação dos modelos desenvolvidos no decorrer deste trabalho em aplicações de controle preditivo baseado em modelo (MPC, do inglês *Model Predictive Control*). A literatura apresenta diversas implementações de MPC com redes neurais artificiais desempenhando o papel de modelo do processo. Algoritmos MPC que empregam redes NARX e RMLP são os mais comuns na literatura. Por sua vez, implementações empregando ESN ainda são incipientes, consistindo em uma alternativa promissora para trabalhos futuros.

REFERÊNCIAS

- ADENIRAN, A. A.; FERIK, S. E. Modeling and identification of nonlinear systems: A review of the multimodel approach - part 1. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 47, n. 7, p. 1149–1159, July 2017.
- ANTONELO, E. A. *Reservoir Computind Architectures for Modeling Robot Navigation Systems*. Tese (Doutorado) — Ingenieurswetenschappen: Computerwetenschappen, University Gent, Gent, 2012.
- ANTONELO, E. A.; CAMPONOGARA, E.; FOSS, B. Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells. *Neural Networks*, v. 85, p. 106 – 117, 2017.
- ATABAY, D. *pyrenn: First release*. 2016. Disponível em: <<https://doi.org/10.5281/zenodo.45022>>. Acesso em: 21 julho 2017.
- BARNARD, E.; WESSELS, L. Extrapolation and interpolation in neural network classifiers. *IEEE Control Systems*, IEEE, v. 12, n. 5, p. 50–53, 1992.
- BISHOP, C. M. *Pattern Recognition and Machine Learning*. Nova Iorque: Springer-Verlag, 2006. ISBN 0387310738.
- BITTENCOURT, G. *Inteligência artificial. Ferramentas e teorias*. 3. ed. Florianópolis: Editora da UFSC, 2006. 371 p.
- CAMACHO, C. B. E. F. *Model Predictive Control*. Londres: Springer Verlag, 2007. 405 p.
- DENG, L. et al. Recent advances in deep learning for speech research at microsoft. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2013.
- DETTERTMAN, D. K. A challenge to watson. *Intelligence*, v. 39, n. 2, p. 77 – 78, 2011.
- ELMAN, J. Finding structure in time. *Cognitive Science*, Wiley-Blackwell, v. 14, n. 2, p. 179–211, June 1990.
- ESTRADA-FLORES, S. et al. Development and validation of “grey-box” models for refrigeration applications: A review of key concepts. *International Journal of Refrigeration*, Elsevier BV, v. 29, n. 6, p. 931–946, Sep 2006.

FLESCH, R. C.; NORMEY-RICO, J. E. Modelling, identification and control of a calorimeter used for performance evaluation of refrigerant compressors. *Control Engineering Practice*, v. 18, n. 3, p. 254 – 261, 2010.

GOLDBERG, Y. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research (JAIR)*, v. 57, p. 345–420, 2015.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.

HAYKIN, S. *Redes neurais. Princípios e práticas*. 2. ed. Porto Alegre: Bookman, 2001. 900 p.

HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. New Jersey: Pearson, 2009. 900 p.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, World Scientific, v. 6, n. 02, p. 107–116, 1998.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks*, Elsevier, v. 2, n. 5, p. 359–366, 1989.

ISERMANN, R.; MÜNCHOF, M. *Identification of Dynamic Systems*. Berlin: Springer, 2011. 705 p.

JAEGER, H. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach. *GMD Report 159*, German National Research Center for Information Technology, 2002.

JAEGER, H.; HAAS, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, American Association for the Advancement of Science, v. 304, n. 5667, p. 78–80, 2004.

JORDAN, M. I. *Serial Order: A parallel Distributed Processing Approach*. San Diego, 1986.

KRITTANAWONG, C. The rise of artificial intelligence and the uncertain future for physicians. *European Journal of Internal Medicine*, 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. Lake Tahoe: Curran Associates, Inc., 2012. p. 1097–1105.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

LEVINE, W. S. *The Control Handbook, (Three Volume Set) (Electrical Engineering Handbook)*. 2nd. ed. Boca Raton: CRC Press, Inc., 2010.

LJUNG, L. *System identification*. 2. ed. Englewood Cliffs, New Jersey: Prentice Hall, 1999.

LUKOŠEVIČIUS, M. A practical guide to applying echo state networks. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). *Neural Networks: Tricks of the Trade: Second Edition*. Berlin: Springer, 2012. p. 659–686.

LUKOŠEVIČIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, v. 3, n. 3, p. 127 – 149, 2009.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: *Proceedings of ICML*. Atlanta: International Conference on Machine Learning, 2013. v. 30, n. 1.

MAASS, C. M. B. W. *Pulsed Neural Networks*. Cambridge: MIT Press, 2001.

MAASS, W. Liquid state machines: Motivation, theory, and applications. In: *Computability in Context: Computation and Logic in the Real World*. Londres: IMPERIAL COLLEGE PRESS, 2011. p. 275–296.

MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, MIT Press - Journals, v. 14, n. 11, p. 2531–2560, Nov 2002.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, 1943.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. Haifa: 27th International Conference on Machine Learning (ICML-2010), 2010. p. 807–814.

NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. *Trans. Neur. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 1, n. 1, p. 4–27, Mar 1990.

NATIONAL SCIENCE AND TECHNOLOGY COUNCIL. *Preparing for the future of artificial intelligence*. 2016. Relatório.

NELLES, O. *Nonlinear System Identification*. Londres: Springer, 2001. 785 p.

PAN, Y.; WANG, J. Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics*, v. 59, n. 8, p. 3089–3101, Aug 2012.

PARLOS, A. et al. Recurrent multilayer perceptron for nonlinear system identification. In: *IJCNN-91-Seattle International Joint Conference on Neural Networks*. Seattle: International Joint Conference on Neural Networks, 1991. v. 2, n. 3, p. 537–540.

PRECHELT, L. Early stopping — but when? In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer, 2012. p. 53–67.

PUSKORIUS, G.; FELDKAMP, L. Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, Institute of Electrical and Electronics Engineers (IEEE), v. 5, n. 2, p. 279–297, Mar 1994.

RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge: MIT Press, 1986.

SAMARASINGHE, S. *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Boca Raton: CRC Press, 2016.

- SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015.
- SCHMITZ, R. B. *Desenvolvimento de bancada automatizada para controle de pressões em compressores de refrigeração*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 2016.
- SCHRAUWEN, B.; VERSTRAETEN, D.; D’HAENE, M. *Reservoir computing toolbox manual*. 2007.
- SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004.
- SMOLA, S. V. A. *Introduction to Machine Learning*. Cambridge: Cambridge University Press, 2008.
- STEIL, J. Backpropagation-decorrelation: online recurrent learning with $o(n)$ complexity. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. Budapeste: IEEE, 2004. v. 1, p. 843–848.
- SVOZIL, D.; KVASNICKA, V.; POSPICHAL, J. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, Elsevier BV, v. 39, n. 1, p. 43–62, 1997.
- TRIEFENBACH, F. et al. Phoneme recognition with large hierarchical reservoirs. In: *Advances in Neural Information Processing Systems 23*. Vancouver: Advances in Neural Information Processing Systems 23 (NIPS 2010), 2010. p. 2307–2315.
- TUTSCHKU, K. Recurrent multilayer perceptrons for identification and control: The road to applications. *Univ. Würzburg, Germany, ser. Research Report Series*, 1995.
- WERBOS, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, IEEE, v. 78, n. 10, p. 1550–1560, 1990.
- WILLIAMS, R. J.; ZIPSER, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, MIT Press, v. 1, n. 2, p. 270–280, 1989.