



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**  
**DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA**

**JÉSSYCA ALMEIDA BESSA**

**MODELOS LOCAIS CRESCENTES E ROBUSTOS A OUTLIERS PARA  
IDENTIFICAÇÃO RECURSIVA DE SISTEMAS DINÂMICOS**

**FORTALEZA**

**2021**

JÉSSYCA ALMEIDA BESSA

MODELOS LOCAIS CRESCENTES E ROBUSTOS A OUTLIERS PARA IDENTIFICAÇÃO  
RECURSIVA DE SISTEMAS DINÂMICOS

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia em Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto

Coorientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B465m Bessa, Jéssyca Almeida.

Modelos Locais Crescentes e Robustos a outliers para Identificação Recursiva de Sistemas Dinâmicos /  
Jéssyca Almeida Bessa. – 2021.  
123 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação  
em Engenharia de Teleinformática, Fortaleza, 2021.

Orientação: Prof. Dr. Ajalmar Rêgo da Rocha Neto.

Coorientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Identificação de sistemas. 2. Modelos locais. 3. Modelos Crescentes. 4. Outliers. 5. Robustez. I. Título.  
CDD 621.38

---

JÉSSYCA ALMEIDA BESSA

MODELOS LOCAIS CRESCENTES E ROBUSTOS A OUTLIERS PARA IDENTIFICAÇÃO  
RECURSIVA DE SISTEMAS DINÂMICOS

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia em Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 28 de abril de 2021

BANCA EXAMINADORA

---

Prof. Dr. Ajalmar Rêgo da Rocha  
Neto (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Guilherme de Alencar  
Barreto (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. João Paulo Pordeus Gomes  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Danton Diego Ferreira  
Universidade Federal de Lavras (UFLA)

---

Prof. Dr.Leandro dos Santos Coelho  
Pontifícia Universidade Católica do Paraná  
(PUC-PR)

Pai e mãe, por vossa capacidade de acreditar e investir em mim. Darlan, por seu cuidado e dedicação que me deram a esperança para seguir. Filho, por sua presença que significou segurança e certeza de que não estou sozinha nessa caminhada.

## **AGRADECIMENTOS**

Primeiro, agradeço a Deus por todas as oportunidades a mim dadas;

Ao meu orientador, Prof. Dr. Ajalmar R. Rocha Neto, pela parceria desde o início do meu mestrado, pelo incentivo nessa reta final de doutorado e pela confiança depositada na minha proposta de projeto.

Ao meu coorientador, Prof. Dr. Guilherme Alencar Barreto, pela confiança do início ao fim, pelos ensinamentos, pela paciência e por todo o apoio e por me proporcionar um conhecimento não apenas racional, mas uma manifestação de afetividade no processo de formação profissional, por tanto que se dedicou a mim, não somente por ter me ensinado, mas por ter me feito aprender.

Ao meu esposo, Darlan, por sempre me fazer olhar para frente, seguir um passo de cada vez e dizer que tudo é possível;

Ao meu filho, Miguel, que me mostrou, desde 2019, o que é o amor incondicional de mãe e, é por ele, que me sinto apta a realizar qualquer coisa todos os dias;

Aos meus pais, Farnésio e Vanda, pelo contínuo incentivo de ir à frente mesmo com todas as adversidades;

À minha irmã, Laryssa, por todo companheirismo;

Ao meu amigo, Prof. Dr. Auzuir Ripardo de Alexandria do IFCE pelo acompanhamento em toda a minha vida como profissional;

À minha amiga, Profa. Dra. Jayna Kátia Dionisio dos Santos do IFCE pelo incentivo constante e por ser um exemplo a ser seguido;

À UFC, Universidade Federal do Ceará, e ao IFCE, Instituto Federal de Educação, Ciência e Tecnologia do Ceará, por todo o repasse de conhecimento;

Aos amigos e professores do curso de Pós Graduação em Engenharia de Teleinformática, por todo o acolhimento desde minha entrada no Doutorado;

Aos meus alunos e amigos que no IFCE cultivei;

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

“Noventa por cento do sucesso se baseia simplesmente em insistir”

(Woody Allen)



## RESUMO

As tarefas de identificação e modelagem de sistemas são fundamentais em controle moderno. Por exemplo, modelos matemáticos são usados para obter aproximações adequadas das respostas dos sistemas que se deseja controlar. De forma sucinta, esses modelos podem ser classificados em dois grupos: modelos globais ou modelos locais. Modelos globais fazem uso apenas de uma única estrutura matemática para representar todo o domínio do problema. Enquanto modelos locais particionam o domínio do problema em regiões menores de modo que um modelo mais simples seja adotado para cada região. Uma grande desvantagem dos modelos locais é a necessidade de especificar o número de submodelos antes do treinamento. Uma possível solução é a utilização de modelos crescentes (*growing models*). Independente da estrutura do modelo, é desejável que a estrutura seja capaz de lidar com amostras discrepantes, *outliers*, conferindo-lhe robustez em cenários não ideais de uso. Isto posto, esta tese tem como objetivo revistar o projeto de modelos locais baseados em quantização vetorial visando estender a aplicabilidade de tais modelos, sejam de tamanho fixo ou crescentes, para identificação recursiva robusta de sistemas dinâmicos não lineares. Os modelos utilizados para este fim utilizam variantes da regra LMS (*least mean square*), escolhida por seu baixo custo computacional para estimação de parâmetros, e por uma regra robusta a *outliers*, baseada no arcabouço teórico de estimação  $M$ , obtida diretamente da regra LMS sem custo computacional extra. Os modelos escolhidos foram os seguintes: o mapeamento linear local (LLM, *local linear map*), a rede de funções de base radial (RBFN, *radial basis functions network*), a rede de modelos locais (LMN, *local model networks*). A partir destes modelos, variantes incrementais e robustas para identificação recursiva de sistemas dinâmicos foram desenvolvidas, a saber, RAN-LMS, G-LMN e ORG-LMN. Os modelos propostos têm como características estrutura online crescente, regras de atualização recursiva rápida, melhor uso de memória (não é necessário estimar a inversa da matriz de covariância como na regra RLS) e robustez a *outliers*. Nesse sentido, eficiência no desempenho e simplicidade de implementação são as qualidades essenciais das abordagens propostas. Os modelos foram testados em um cenário de simulação livre e foram avaliados por duas métricas: (i) erro médio quadrático e (ii) teste de Kolmogorov-Smirnov. Uma avaliação abrangente envolvendo dois conjuntos de dados sintéticos e cinco conjuntos de dados reais, dentre estes um conjunto de dados de larga escala e um conjunto de dados MIMO (*multiple input, multiple output*), corrobora o desempenho preditivo superior da abordagem proposta em cenários contaminados por *outliers*

em comparação a modelos alternativos.

**Palavras-chave:** Identificação de sistemas. Modelos Locais. Modelos Crescentes. Outliers. Robustez. Estimação- $M$ .

## ABSTRACT

The tasks of identification and modeling of dynamical systems are fundamental in modern control. For example, models are used to obtain adequate approximations of the response of the system to be controlled. Roughly, these models can be classified into two groups: global models or local models. Global models use only a single mathematical structure to represent the entire domain of the problem. Local models, in their turn, partition the problem domain into smaller regions so that a simpler model can be adopted for each region. A major disadvantage of local models is the need to specify the number of submodels before training. A possible solution involves the use of growing models. Regardless of the model's structure, it is desirable that the structure be capable of handling outlying samples, providing robustness to the model robustness in real-world applications. That said, this thesis aims to review the design of local models based on vector quantization in order to extend the applicability of such models, whether of fixed or increasing size, for robust recursive identification of nonlinear dynamic systems. The models chosen for this purpose use variants of the LMS (least mean square) rule, chosen due to its low computational cost for parameter estimation, and an outlier-robust rule, based on the framework of  $M$  estimation, which is obtained directly from the LMS rule without additional cost. The chosen models were the following: the local linear mapping (LLM), the radial basis functions network (RBFN), the local model network (LMN). From these models, incremental and robust variants for recursive identification of dynamic systems were developed, namely, RAN-LMS, G-LMN and ORG-LMN. The proposed models have the following characteristics: increasing online structure, fast recursive update rules, better memory usage (it is not necessary to estimate the inverse of the covariance matrix as required in the RLS rule) and robustness to outliers. In this sense, efficiency in performance and simplicity of implementation are the essential qualities of the proposed approaches. The models were tested in a free simulation scenario and were evaluated by two metrics: (i) mean squared error and (ii) Kolmogorov-Smirnov test. A comprehensive assessment involving two sets of synthetic data and five sets of real data, including a large-scale data set and a MIMO data set, corroborates the superior predictive performance of the proposed approaches in scenarios contaminated by outliers compared to alternative models.

**Keywords:** System Identification. Local Models. Growing Models. Outliers. Robustness. M-Estimation

## LISTA DE FIGURAS

Figura 1 – Abordagens de modelos de sistemas. . . . .	32
Figura 2 – Exemplo de um problema de identificação de sistemas (SISO) em que o modelo é adaptado para representar o processo com dinâmica desconhecida. Adaptado de Santos (2017). . . . .	33
Figura 3 – Representação do modelo LLM. . . . .	38
Figura 4 – Representação do modelo RBF. . . . .	41
Figura 5 – Representação do modelo LMN. . . . .	45
Figura 6 – Esquema de um modelo NARX MIMO, em que $\mathbf{G}(\cdot)$ é uma função não linear genérica. . . . .	63
Figura 7 – Abordagens para tratamento de <i>outliers</i> . Adaptado de Blazquez-García <i>et al.</i> (2021) . . . . .	69
Figura 8 – Cenários para validação do modelo. Adaptado de Nelles (2020). . . . .	78
Figura 9 – Valores medidos de entrada e saída do conjunto de dados Sintético 1. . . . .	81
Figura 10 – Valores medidos de entrada e saída do conjunto de dados Sintético 2. . . . .	81
Figura 11 – Valores medidos de entrada (posição da válvula) e saída (pressão do óleo) do conjunto de dados atuador hidráulico. . . . .	82
Figura 12 – Valores medidos de entrada(tensão aplicada) e saída(temperatura) do conjunto de dados secador de cabelo. . . . .	83
Figura 13 – Valores medidos de entrada(fluxo da solução de base) e saída( <i>pH</i> ) do conjunto de dados neutralização de <i>pH</i> . . . . .	83
Figura 14 – Valores medidos de entrada e saída do conjunto de dados <i>silverBox</i> . . . . .	84
Figura 15 – Sequência de entradas e saídas do conjunto de dados MIMO secador industrial. . . . .	85
Figura 16 – RMSE no conjunto de teste <i>versus</i> número de modelos durante o treinamento com contaminação de 20% de <i>outliers</i> (dados sintéticos 2) (versões robustas). . . . .	87
Figura 17 – RMSE no conjunto de teste para diferentes proporções de <i>outliers</i> durante o treinamento com 5 modelos locais (dados sintéticos 1): (a) LLM, (b) RBF e (c) LMN. . . . .	88
Figura 18 – RMSE no conjunto de teste para diferentes proporções de <i>outliers</i> durante o treinamento com 5 modelos locais (dados sintéticos 2): (a) LLM, (b) RBF e (c) LMN. . . . .	89

Figura 19 – Séries preditas por simulação livre (conjunto de teste) pelos modelos LLM versão original (esquerda) e versão robusta (direita) com 5 modelos locais e 10% de <i>outliers</i> (sintético 1). . . . .	89
Figura 20 – Séries preditas por simulação livre (conjunto de teste) pelos modelos LMN versão original (esquerda) e versão robusta (direita) com 5 modelos locais e 10% de <i>outliers</i> (sintético 2). . . . .	90
Figura 21 – Valores de RMSE na fase de teste em função da quantidade de <i>outliers</i> dos dados de treinamento (atuador hidráulico). . . . .	91
Figura 22 – Séries preditas por simulação livre (conjunto de teste) pelos modelos LMN com estrutura fixa (original e robusto)(superior), G-LMN versão original (inferior esquerda) e ORG-LMN versão robusta (inferior direita) com 15% de <i>outliers</i> (atuador hidráulico). . . . .	92
Figura 23 – Evolução do número de modelos locais em função das épocas de treinamento para modelos propostos: versões originais (linha azul sólida) e versões robustas (linha vermelha tracejada) (atuador hidráulico). . . . .	93
Figura 24 – RMSE <i>versus</i> número de modelos locais com 15% <i>outliers</i> : versões originais (linha azul sólida) e versões robustas (linha vermelha tracejada)(atuador hidráulico - conjunto de treinamento). . . . .	93
Figura 25 – RMSE na fase de treinamento <i>versus</i> número de modelos locais com 15% <i>outliers</i> : G-LMN (linha azul sólida) e ORG-LMN (linha vermelha tracejada)(secador de cabelo). . . . .	95
Figura 26 – Séries preditas por simulação livre (conjunto de teste) pelos modelos LMN com estrutura fixa (original e robusto)(superior) e com estrutura crescente (original e robusto)(inferior) com 15% de <i>outliers</i> (secador industrial). . . . .	95
Figura 27 – Evolução do número de modelos locais em função das épocas de treinamento para modelos propostos: versão original (linha azul sólida) e versão robusta (linha vermelha tracejada) (secador de cabelo). . . . .	96
Figura 28 – Valores de RMSE na fase de teste em função da quantidade de <i>outliers</i> dos dados de treinamento (conjunto de dados de neutralização de <i>pH</i> ). . . . .	97

Figura 29 – Séries preditas por simulação livre (conjunto de teste) pelos modelos incrementais (versões originais - esquerda e robustas - direita): INC-LLM (superior), RAN (meio), G-LMN (inferior) com 15% de <i>outliers</i> (conjunto de dados de neutralização de <i>pH</i> ). . . . .	98
Figura 30 – RMSE na fase de treinamento versus número de modelos locais com 15% <i>outliers</i> : G-LMN (linha azul sólida) e ORG-LMN (linha vermelha tracejada)(conjunto de dados de neutralização de <i>pH</i> ) . . . . .	99
Figura 31 – Desempenho do modelo proposto ORG-LMN durante as primeiras 200 amostras de treinamento para o conjunto de dados <i>Silverbox</i> . A figura superior mostra a série temporal de saída estimada, enquanto a figura inferior mostra a evolução correspondente dos valores RMSE. . . . .	100
Figura 32 – Séries preditas (conjunto de teste) pelos modelos G-LMN (superior) e ORG-LMN (inferior) em cenários contaminados com 15% de <i>outliers</i> (Conjunto de dados <i>Silverbox</i> ). . . . .	101
Figura 33 – CDFs empíricas dos resíduos gerados (dados de treinamento) pelo modelo G-LMN (linha sólida em azul) e o modelo ORG-LMN (linha vermelha tracejada) para o conjunto de dados <i>Silverbox</i> . Cenário livre de <i>outliers</i> (superior). Cenário contaminado por <i>outliers</i> (inferior). . . . .	102
Figura 34 – Valores de RMSE em função da quantidade de <i>outliers</i> dos dados de treinamento para as saídas $y_1$ , $y_2$ e $y_3$ (Conjunto de dados MIMO secador industrial). . . . .	105
Figura 35 – Evolução do número de modelos locais em função das épocas de treinamento para modelos incrementais robustos (conjunto de dados MIMO secador industrial). . . . .	106
Figura 36 – Melhores séries preditas por simulação livre ( $y_1$ , $y_2$ e $y_3$ ) pelo modelo proposto ORG-LMN em cenários contaminados com 15% de <i>outliers</i> (Conjunto MIMO Secador Industrial). . . . .	107
Figura 37 – Modelos implementados nesta tese. . . . .	107

## LISTA DE TABELAS

Tabela 1 – Regras de aprendizado das duas propostas: variante da RAN e G-LMN. . . . .	60
Tabela 2 – Regras recursivas robustas desenvolvidas a partir da regra LMM. . . . .	75
Tabela 3 – Descrição dos conjuntos de dados sintéticos usados nos experimentos computacionais. $U(A, B)$ é um número aleatório uniforme entre $A$ e $B$ . . . . .	81
Tabela 4 – Características dos conjuntos de dados avaliados. . . . .	82
Tabela 5 – RMSE na fase de teste utilizando conjunto de dados sintéticos 1. . . . .	88
Tabela 6 – RMSE na fase de teste utilizando conjunto de dados sintéticos 2. . . . .	88
Tabela 7 – RMSE na fase de teste para o conjunto de dados atuador hidráulico para dois níveis de contaminação de <i>outliers</i> (estrutura fixa, $S = 5$ ). . . . .	91
Tabela 8 – RMSE na fase de teste para o conjunto de dados atuador hidráulico para dois níveis de contaminação de <i>outliers</i> (estruturas crescentes). . . . .	91
Tabela 9 – RMSE na fase de teste para o conjunto de dados secador de cabelo para dois níveis de contaminação de <i>outliers</i> . . . . .	94
Tabela 10 – RMSE na fase de teste para o conjunto de dados de neutralização de <i>pH</i> para dois níveis de contaminação com <i>outliers</i> . . . . .	97
Tabela 11 – RMSE na fase de teste para o conjunto de dados <i>Silverbox</i> para dois níveis de contaminação com <i>outliers</i> . . . . .	100
Tabela 12 – Teste KS aplicado em cenários sem presença de <i>outliers</i> . . . . .	102
Tabela 13 – Teste KS aplicado em cenários contaminados com 15% <i>outliers</i> . . . . .	102
Tabela 14 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo INC-LLM-LMS. . . . .	104
Tabela 15 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo INC-LLM-LMM. . . . .	104
Tabela 16 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo RAN-LMS. . . . .	104
Tabela 17 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo RAN-LMM. . . . .	104
Tabela 18 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo G-LMN. . . . .	104
Tabela 19 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo ORG-LMN. . . . .	105

Tabela 20 – Valores de RMSE em função da estratégia adotada para a fase de teste em cenários contaminados com 15% de <i>outliers</i> na fase de estimação para o modelo ORG-LMN. . . . .	106
Tabela 21 – Teste KS aplicado utilizando o modelo LLM em cenários sem a presença de <i>outliers</i> utilizando o conjunto de Dados Sintéticos 1. . . . .	120
Tabela 22 – Valores de RMSE para a fase de teste utilizando o modelo LLM em cenários sem a presença de <i>outliers</i> utilizando o conjunto de Dados Sintéticos 1. . . . .	120
Tabela 23 – Teste KS aplicado em cenários sem presença de <i>outliers</i> (Atuador Hidráulico).	121
Tabela 24 – Teste KS aplicado em cenários contaminados com 15% <i>outliers</i> (Atuador Hidráulico). . . . .	121
Tabela 25 – Teste KS aplicado em cenários sem presença de <i>outliers</i> (Secador de Cabelo).	121
Tabela 26 – Teste KS aplicado em cenários contaminados com 15% <i>outliers</i> (Secador de Cabelo). . . . .	122
Tabela 27 – Teste KS aplicado em cenários sem presença de <i>outliers</i> (pH). . . . .	122
Tabela 28 – Teste KS aplicado em cenários contaminados com 15% <i>outliers</i> (pH). . . . .	122



## LISTA DE ALGORITMOS

Algoritmo 1 – Modelo INC-LMM. . . . .	53
Algoritmo 2 – Modelo RAN-LMS . . . . .	57
Algoritmo 3 – Modelo G-LMN . . . . .	61
Algoritmo 4 – Modelo ORG-LMN . . . . .	74

## LISTA DE ABREVIATURAS E SIGLAS

FIR	<i>Finite Impulse Response</i>
G-LMN	<i>Growing Local Model Network</i>
INC-LLM	<i>Incremental Local Linear Mapping</i>
LLM	<i>Local Linear Mapping</i>
LMN	<i>Local Model Network</i>
LMM	<i>Least Mean M-estimate</i>
LMS	<i>Least Mean Squares</i>
M	Máxima Verossimilhança
MIMO	<i>Multiple Input, Multiple Output</i>
MLP	<i>Multi Layer Perceptron</i>
OLS	<i>Ordinary Least Squares</i>
ORG-LMN	<i>Outlier Robust Growing Local Model Network</i>
RAN	<i>Resource Allocating Network</i>
RBF	<i>Radial Basis Function</i>
RBFN	<i>Radial Basis Function Network</i>
SISO	<i>Single Input, Single Output</i>
SOM	<i>Self Organizing Maps</i>
TS	<i>Takagi-Sugeno</i>
AR	<i>Auto-Regressive</i>
ARX	<i>Auto-Regressive with eXogenous inputs</i>
MA	<i>Moving Average</i>
ARMA	<i>Auto-Regressive Moving Average</i>

ARMAX	<i>Auto-Regressive Moving Average with eXogenous inputs</i>
OE	<i>Output Error</i>
BJ	<i>Box-Jenkins</i>
NARX	<i>Nonlinear AutoRegressive with eXogenous inputs</i>
NARMAX	<i>Nonlinear Auto-Regressive Moving Average with eXogenous inputs</i>
NOE	<i>Nonlinear Output Error</i>
NFIR	<i>Nonlinear Finite Impulse Response</i>
NOBF	<i>Nonlinear Orthonormal Basis Functions</i>
LOLIMOT	<i>Local Linear Model Trees</i>
MDE	<i>Modified Differential Evolution</i>
RNA	Rede Neural Artificial
GK	<i>Gustafson-Kessel</i>
LARFSOM	<i>Local Adaptive Receptive Fields Self-Organizing Map</i>
RLS	<i>Recursive Least Squares</i>
GNG	<i>Growing Neural Gas</i>
ADAIN	<i>Adaptive Incremental Learning</i>
SRAN	<i>Self-adaptive Resource Allocation Network</i>
PID	Proporcional Integrador Derivativo
AGV	<i>Automated Guided Vehicle</i>
RLM	<i>Recursive Least M-estimate</i>
MSE	<i>Mean Square Error</i>
HM	Huber modificada
QV	Quantização vetorial

## LISTA DE SÍMBOLOS

$t$	Tempo
$u(t)$	Entrada de um sistema
$y(t)$	Saída de um sistema
$\hat{y}(t)$	Saída estimada de um sistema
$e(t)$	Erro entre saída observada e estimada de um sistema
$n(t)$	Ruído
$N$	Quantidade de amostras
$W$	Modelo do sistema
$G(\cdot)$	Função que descreve o sistema
$\hat{G}(\cdot)$	Função estimada para descrever o sistema
$L_u$	Ordem de memória da entrada
$L_y$	Ordem de memória da saída
$\mathbb{R}$	Números reais
$\mathbf{w}$	Vetor protótipo no espaço de entrada
$\hat{y}_j$	$j$ -ésimo protótipo no espaço de saída
$\mathbf{a}$	Vetor de coeficientes
$S$	Número de modelos locais
$j^*$	Índice do neurônio vencedor
$T$	Transposição de matriz/vetor
$h(\cdot)$	Função de vizinhança
$\alpha$	Taxa de aprendizagem
$\alpha'$	Taxa de aprendizagem
$\varepsilon$	Termo de correção para evitar divisão por zero
$\mathbf{c}$	Centro da função de base
$d(\mathbf{x}, \mathbf{y})$	Distância entre os vetores $\mathbf{x}$ e $\mathbf{y}$ .
$z_i$	Saída da $i$ -ésima função de base

$\sigma$	Raio ou largura da função de base
$f_i$	Função local associada à $i$ -ésima função de base
$\varepsilon_b$	Constante usada no algoritmo LLM Incremental
$\varepsilon_n$	Constante usada no algoritmo LLM Incremental
$a_{max}$	Idade máxima dos <i>links</i> no algoritmo LLM Incremental
$\kappa$	Fator de sobreposição
$\delta(t)$	Limiar decrescente para detecção de novidade
$J$	Função objetivo do modelo
$\partial$	Derivada parcial
$\rho(\cdot)$	Função custo na Estimação $M$
$E$	Esperança
$\beta$	Limiar usado para mitigar a influência de outliers
$\nabla$	Vetor gradiente
$\psi$	Função de ponderação do erro na estimação $M$
$\sigma$	Desvio padrão dos resíduos
$\infty$	Infinito

## SUMÁRIO

1	INTRODUÇÃO . . . . .	24
1.1	Objetivo Geral e Específicos . . . . .	28
1.2	Produção Científica . . . . .	28
1.3	Organização da Tese . . . . .	29
2	MODELOS LOCAIS EM IDENTIFICAÇÃO DE SISTEMAS . . . . .	31
2.1	Introdução . . . . .	31
2.2	O modelo LLM . . . . .	35
2.2.1	<i>Revisão bibliográfica</i> . . . . .	36
2.2.2	<i>Formalização matemática</i> . . . . .	37
2.3	O modelo RBFN . . . . .	39
2.3.1	<i>Revisão bibliográfica</i> . . . . .	39
2.3.2	<i>Formalização matemática</i> . . . . .	41
2.4	O modelo LMN . . . . .	44
2.4.1	<i>Revisão bibliográfica</i> . . . . .	44
2.4.2	<i>Formalização matemática</i> . . . . .	44
2.5	Considerações Finais . . . . .	46
3	MODELOS LOCAIS CRESCENTES . . . . .	48
3.1	Introdução . . . . .	48
3.2	Mapeamento linear local incremental . . . . .	49
3.3	Proposta 1: Modelo RAN-LMS . . . . .	54
3.4	Proposta 2: Modelo LMN Crescente (caso SISO) . . . . .	58
3.5	Proposta 3: O caso MIMO . . . . .	62
3.6	Considerações finais . . . . .	66
4	MODELOS LOCAIS ROBUSTOS A <i>OUTLIERS</i> . . . . .	68
4.1	Introdução . . . . .	68
4.2	Estimação Recursiva Robusta via Regra LMM . . . . .	70
4.3	Proposta 4: <i>Outlier-robust growing</i> LMN . . . . .	72
4.4	Outras variantes robustas . . . . .	75
4.5	Considerações finais . . . . .	76
5	METODOLOGIA . . . . .	77

5.1	<b>Validação dos modelos</b>	77
5.2	<b>Medidas de Avaliação de Desempenho</b>	79
5.2.1	<i>Erro médio quadrático</i>	79
5.2.2	<i>Teste de Kolmogorov-Smirnov</i>	79
5.3	<b>Conjuntos de dados avaliados</b>	80
5.3.1	<i>Dados sintéticos</i>	80
5.3.2	<i>Dados reais</i>	80
5.3.2.1	<i>Atuador hidráulico</i>	82
5.3.2.2	<i>Secador de cabelo</i>	82
5.3.2.3	<i>Neutralização de pH</i>	83
5.3.2.4	<i>Silverbox</i>	83
5.3.2.5	<i>Secador Industrial - caso MIMO</i>	84
5.4	<b>Considerações finais</b>	84
6	<b>RESULTADOS</b>	86
6.1	<b>Resultados nos conjuntos de dados sintéticos</b>	86
6.2	<b>Resultados no conjunto atuador hidráulico</b>	90
6.3	<b>Resultados no conjunto secador de cabelo</b>	93
6.4	<b>Resultados no conjunto neutralização de <i>pH</i></b>	96
6.5	<b>Resultados no conjunto <i>Silverbox</i></b>	99
6.5.1	<i>Avaliação via Teste KS</i>	101
6.6	<b>Resultados no conjunto Secador Industrial (MIMO)</b>	103
6.7	<b>Considerações finais</b>	106
7	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	108
7.1	<b>Conclusões e Discussão</b>	108
7.2	<b>Resumo das Contribuições da Tese</b>	109
7.3	<b>Sobre a continuidade desta pesquisa</b>	110
	<b>REFERÊNCIAS</b>	111
	<b>APÊNDICES</b>	120
	<b>APÊNDICE A – COMPARAÇÃO ENTRE ALGORITMOS DE QUANTIZAÇÃO VETORIAL PARA O ESPAÇO DE ENTADA <math>\chi</math></b>	120
	<b>APÊNDICE B – TESTES DE KOLMOGOROV-SMIRNOV</b>	121

<b>ANEXOS</b> .....	121
---------------------	-----



# 1 INTRODUÇÃO

Um sistema dinâmico é um processo natural ou artificial cujas entradas e saídas apresentam dependência temporal (ou causal). Por meio da utilização de dados observados na forma de séries temporais de entrada-saída em vez de equações da Física, o objetivo da identificação de sistemas é capturar essa dependência temporal, ou dinâmica, entre as variáveis de entrada e saída (LJUNG, 1999). O modelo construído por meio do processo de identificação de sistemas pode então ser útil para descrever, controlar, monitorar e simular os sistemas de interesse (SJÖBERG *et al.*, 1995). Conhecer um modelo que descreva o comportamento de um sistema dinâmico é essencial em diversas áreas do conhecimento técnico-científico, principalmente nas ciências e nas engenharias (BRUNTON; KUTZ, 2019).

Quanto à escolha do modelo, deve-se normalmente buscar um compromisso entre o custo de se ter o modelo, isto é, o tempo e o esforço requeridos para obtê-lo e verificar o nível de detalhes do modelo, bem como os benefícios esperados de sua aplicação. Pode-se dividir os modelos matemáticos em grupos de acordo com certas características que os distinguem, tais como determinísticos e estocásticos, paramétricos e não paramétricos, lineares e não lineares, globais e locais (AGUIRRE, 2019).

Em um modelo determinístico, a saída pode ser calculada de forma exata tão logo se conheça o sinal de entrada e as condições iniciais. Em contraste, um modelo estocástico contém termos aleatórios que dificultam ou impossibilitam o cálculo exato da saída. Normalmente, o modelo determinístico engloba apenas o processo, enquanto o estocástico considera também as perturbações e ruídos (AGUIRRE, 2019).

Já modelos paramétricos e não paramétricos se diferenciam em relação ao resultado da modelagem. Os modelos paramétricos resultam em um modelo matemático tal como uma função de transferência, enquanto os modelos não paramétricos resultam em uma representação gráfica que caracteriza a dinâmica do sistema em estudo (AGUIRRE, 2019).

Em relação a sistemas lineares e não lineares, a ampla utilização de técnicas de controle baseadas em modelos lineares é, em parte, devido à simplicidade dos modelos empregados para representar o comportamento do processo; no entanto, isto também constitui uma deficiência potencial porque modelos lineares são, muitas vezes, inadequados quando se requer uma melhor aproximação de um processo. Por outro lado, esquemas de controle não linear, os quais empregam modelos mais complexos para a descrição de processos não lineares, eliminam a simplicidade associada às técnicas lineares. Os modelos não lineares possibilitam um retrato

mais fiel do processo quando este se faz necessário (AGUIRRE, 2019; NELLES, 2020).

Isto posto, é possível afirmar que existem várias maneiras de descrever um sistema a partir dos dados o que se deve fazer é escolher o modelo mais adequado ao problema de interesse. O campo da inteligência computacional, por exemplo, tem contribuído com vários métodos para identificação de sistemas dinâmicos, começando provavelmente com o modelo *fuzzy* TS (TAKAGI; SUGENO, 1985; ANGELOV; FILEV, 2004; RONG *et al.*, 2006) introduzido em meados da década de 1980. Continua-se com abordagens baseadas em redes neurais, tais como a bem conhecida rede perceptron multicamadas (*multilayer perceptron*, MLP) (NARENDRA; PARTHASARATHY, 1990; LI; YU, 2002), as redes de funções de base radial (*radial basis function network*, RBFN) (CHEN *et al.*, 1990) e os mapas auto-organizáveis (*self organizing maps*, SOM) (WALTER *et al.*, 1990; MOSHOU; RAMON, 1997; BARRETO; ARAÚJO, 2004; BARRETO; SOUZA, 2016; ZHAO *et al.*, 2016).

Uma tarefa muito importante em um processo de identificação é a de selecionar uma estrutura de modelo (LJUNG, 1998). De maneira sucinta, estas estruturas podem ser classificadas em duas categorias: modelos globais ou modelos locais. Modelos globais são projetados para caracterizar todo o domínio do problema usando apenas uma única estrutura matemática. Estas estruturas podem ser polinomiais, racionais, e até neurais (*feedforward* ou recorrentes, uni- ou multicamadas, etc.) (LAWRENCE *et al.*, 1996). Os modelos baseados na rede MLP, por exemplo, implementam uma abordagem global para identificação do sistema, uma vez que todo o conjunto de treinamento é usado para construir um único modelo que aproxima a relação entrada-saída do problema.

Modelos locais, por outro lado, particionam o domínio do problema em regiões menores de modo que um modelo estruturalmente mais simples seja adotado para cada região do espaço. Os modelos baseados em TS, RBF e SOM implementam uma abordagem de modelo local, no sentido de que vários submodelos localizados em subregiões do domínio do problema são construídos usando apenas uma partição dos dados (FOSS; JOHANSEN, 1993; CHUANG, 2007). As saídas dos vários submodelos podem ser agregadas ou utilizadas individualmente para prever a saída do modelo. Para esta tese, tem-se um particular interesse em três abordagens locais: *local linear mapping* (LLM), *radial basis functions networks* (RBFN) e *local model network* (LMN). Tal interesse surge principalmente em razão da simplicidade dos modelos originais, que utilizam técnicas de quantização vetorial do espaço de entrada para encontrar as subregiões onde os modelos locais serão construídos. Para estimar os parâmetros dos submodelos é comum usar

as regras LMS (*least mean squares*) e as regras RLS (*recursive least squares*) (HAYKIN *et al.*, 2003).

A ideia básica do modelo LLM (WALTER *et al.*, 1990) é aproximar uma função por meio de um conjunto de hiperplanos, cada um dos quais limitado a uma região local do espaço de entrada  $\mathbb{R}^n$ . Cada região local está associada a um neurônio e cada neurônio está associado a um filtro de resposta ao impulso finita (*finite impulse response*, FIR) cujos parâmetros são ajustados pelo algoritmo LMS. Esta matriz é usada para quantizar o espaço de entrada em um número reduzido de vetores de protótipos, enquanto o filtro associado ao neurônio vencedor fornece uma estimativa local da saída a ser aproximada.

A rede RBF é uma arquitetura clássica de rede neural de uma camada oculta (MUSAVI *et al.*, 1992). Os neurônios ocultos, designados como funções de base radial, possuem uma função de ativação não linear, enquanto, os neurônios de saída tem, em geral, uma função de ativação linear. Existem várias maneiras de projetar a rede RBF mas este trabalho se restringe ao procedimento descrito por Moody e Darken (1989). Estes autores treinam uma rede RBF em três etapas executadas sequencialmente. (i) Determinação dos centros por através de um algoritmo de quantização vetorial, tal como a rede SOM; (ii) Determinação do raio  $\sigma$  das funções de base. Para este fim, o raio de um neurônio tem seu valor definido como metade da distância entre o centro do neurônio  $j$  e o centro mais próximo; (iii) Determinação do vetor de pesos de saída  $\mathbf{w} \in \mathbb{R}^S$ .

O modelo LMN foi introduzido por Murray-Smith e Johansen (1995) como uma generalização da rede RBF. Esta abordagem pode ser vista como um sistema não linear construído a partir de um conjunto de submodelos locais que são, então, combinados linearmente pelas ativações das funções de base associadas. Isso significa que um número menor de modelos locais pode abranger áreas maiores do espaço de entrada, quando comparado com a rede RBF simples.

A ideia de usar modelos locais dentro de uma rede RBF foi sugerida em vários trabalhos que remontam ao início da década de 1990 (JONES *et al.*, 1990; STOKBRO *et al.*, 1990; BARNES *et al.*, 1991; JACOBS *et al.*, 1991), mas aplicações mais recentes deste modelo local continuam a surgir (KÖNIG *et al.*, 2014; COSTA *et al.*, 2015; MÜNKER *et al.*, 2017; BELZ *et al.*, 2017; MAIER *et al.*, 2018; BLAZIC; SKRJANC, 2020; WEINSTEIN; HUBBARD, 2020). Na literatura, pode-se encontrar alguns trabalhos utilizando o modelo LMN para identificação do sistema (BESSA; BARRETO, 2019a; BESSA; BARRETO, 2019b; BELZ *et al.*, 2017; MÜNKER *et al.*, 2017; COSTA *et al.*, 2015; PRASAD *et al.*, 1998).

Uma desvantagem dos modelos locais supracitados é a necessidade de especificar o número de submodelos antes do treinamento. Trabalhos tais como Gao (2004), Souza (2012) e Belz *et al.* (2017) abordam esta limitação de modelos locais. Uma abordagem para mitigar esta desvantagem envolve a utilização de modelos crescentes (*growing models*) nos quais os submodelos são progressivamente inseridos até que um determinado requisito de desempenho baseado em erro ou quantidade máxima permitida de submodelos seja alcançado. Esta abordagem também é adequada para lidar com a não estacionariedade, uma vez que a dinâmica do sistema pode mudar ao longo do tempo e o modelo pode adaptar sua estrutura para lidar com as novas demandas da tarefa (KALHOR *et al.*, 2013).

A abordagem de modelo crescente para identificação de sistema e tarefas relacionadas como previsão de séries temporais, detecção de anomalias, diagnóstico de falhas e controle, tem recebido contribuições de diferentes áreas da inteligência computacional. Por exemplo, um modelo crescente baseado em TS, no qual novas regras *fuzzy* são inseridas, é proposto em Angelov e Filev (2004). Modelos baseados em SOM são introduzidos em Liu *et al.* (2009), Kiumarsi *et al.* (2015), enquanto modelos baseados na rede RBF são propostos em Fritzke (1994), Okamoto *et al.* (2003) e Han *et al.* (2010). Um modelo de crescimento baseado em LMN foi recentemente introduzido em Blazic e Skrjanc (2020) e aplicado à previsão de séries temporais.

Sabido quais modelos locais interessam a esta tese, tem-se que destacar que um outro requisito desejável para a identificação do sistema é a robustez a *outliers*, aqui entendido em um sentido amplo como uma medição anômala. Estas amostras anormais são comuns em aplicações do mundo real e podem até ser removidas antes da construção do modelo em configurações *offline*. No entanto, em cenários de identificação de sistema *online*, *outliers* devem ser adequadamente tratados por métodos robustos de estimação de parâmetros (HAMPEL *et al.*, 2011). Independente da estrutura do modelo (global ou local) é desejável que a estrutura seja capaz de lidar com estas amostras discrepantes, conferindo-lhe robustez em cenários não-ideais de uso.

Os métodos padrões de estimativa de parâmetros utilizados na identificação de sistemas, como os mínimos quadrados ordinários (*ordinary least squares*, OLS) e os algoritmos recursivos LMS e RLS são ideais sob a suposição de gaussianidade dos erros de estimativa (ZOU *et al.*, 2001; KOVACEVIC *et al.*, 2016). No entanto, tal suposição não é realista em aplicações com amostras discrepantes, e a aplicação cega de métodos de estimativa de parâmetros não

robustos frequentemente leva a estimativas viesadas dos parâmetros do modelo.

Dentre os trabalhos citados anteriormente, o tratamento de *outliers* ou não é abordado ou quando o é, a abordagem utilizada não é satisfatória. Um dos poucos trabalhos que abordam diretamente o efeito da presença de *outliers* nos dados de estimação é o de Bessa e Barreto (2019b), com os modelos LLM, RBF e LMN de tamanho fixo utilizando modificação das regras de aprendizado com o auxílio do arcabouço de estatística robusta conhecido como *estimação-M* (HUBER, 1964).

## 1.1 Objetivo Geral e Específicos

Diante do exposto até agora e em um sentido mais amplo, o objetivo geral desta tese é revisitar o projeto de modelos locais baseados em quantização vetorial visando estender a aplicabilidade de tais modelos, sejam de tamanho fixo ou crescentes, para identificação recursiva robusta de sistemas dinâmicos não lineares.

Quanto aos objetivos específicos desta tese, pode-se listar os seguintes itens:

- Estender a aplicação de modelos locais de estrutura fixa para identificação recursiva robusta de sistemas dinâmicos.
- Realizar uma breve comparação de algoritmos de quantização vetorial do espaço de entrada para identificação robusta de sistemas com modelos de estrutura fixa.
- Avaliar o estado da arte em modelos locais crescentes em identificação recursiva robusta de sistemas dinâmicos.
- Propor modelo para identificação robusta recursiva com estrutura crescente.
- Avaliar o desempenho dos modelos locais propostos para conjuntos de dados *single input, single output* (SISO) disponíveis na literatura especializada de identificação de sistemas.
- Avaliar o desempenho dos modelos locais propostos para identificação de sistemas (crescente e robustos) em conjunto de dados *multiple input, multiple output* (MIMO) e em conjunto de larga escala.

## 1.2 Produção Científica

Ao longo do desenvolvimento desta tese de doutorado, os seguintes trabalhos foram desenvolvidos:

- **Bessa, J. A.**, Barreto, G. A. (2019). Recursive System Identification Using Outlier-Robust

Local Models. IFAC-PapersOnLine, 52(1), 436-441.

- **Bessa, J. A.**, Barreto, G. A. (2018). Identificação Recursiva de Sistemas Dinâmicos usando Modelos Locais Robustos a Outliers. XXII Congresso Brasileiro de Automática (CBA 2018). Anais da Sociedade Brasileira de Automática.
- **Bessa, J. A.**, Barroso, D. A., da Rocha Neto, A. R., de Alexandria, A. R. (2015). Global location of mobile robots using artificial neural networks in omnidirectional images. IEEE Latin America Transactions, 13(10), 3405-3414.

Os trabalhos abaixo foram submetidos:

- **Bessa, J. A.**, Barreto, G. A. An Outlier-Robust Growing Local Model Network for Recursive System Identification. Neural Processing Letters (submetido).
- **Bessa, J. A.**, Barreto, G. A. Outlier-Robust Growing Local Models for System Identification: A performance comparison. IEEE Latin America Transactions (submetido).

### 1.3 Organização da Tese

O restante da tese está organizado conforme a lista de capítulos apresentada a seguir.

O **Capítulo 2** apresenta uma abordagem do problema de Identificação de Sistemas. Neste capítulo, comentou-se sobre as etapas que envolvem o processo de Identificação de Sistemas, classes de modelos existentes na literatura, e descreveu-se a formulação do vetor de regressores (NARX, *nonlinear autoregressive with exogenous inputs*) que será utilizada no restante dos capítulos. Além disso, esse capítulo contém as idéias básicas dos três modelos locais de estrutura fixa não robustos a *outliers* implementados nesta tese: LLM, RBFN e LMN.

O **Capítulo 3** é dedicado a apresentação dos modelos crescentes e a apresentação da nossa proposta crescente da rede de modelos locais (*growing local model network*, G-LMN). Apresentou-se uma discussão de algoritmos de modelos locais crescentes encontrados na literatura como: mapeamento linear local incremental (*incremental local linear mapping*, INC-LLM) e a rede de alocação de recursos (*resource allocating network*, RAN). Ao final do capítulo, realizou-se a apresentação da formulação MIMO para os modelos incrementais.

O **Capítulo 4** traz uma abordagem sobre robustez a *outliers*. Neste capítulo, apresenta-se a fundamentação teórica de estimadores- $M$ , que são apresentados como uma alternativa aos métodos baseados na regra LMS. A seguir, são apresentadas as versões robustas do modelos descritos com a substituição da regra LMS pela regra LMM (baseada em estimação  $M$ ). Todos os modelos robustos são tidos como propostas desta tese. Destaca-se o modelo *outlier*

*robust growing local model network* (ORG-LMN) que é uma das contribuições desta tese.

No **Capítulo 5**, a metodologia adotada para os experimentos computacionais realizados nesta tese é apresentada. Uma breve discussão sobre os bancos de dados utilizados para avaliarmos os modelos é realizada.

No **Capítulo 6**, são apresentados os resultados de experimentos computacionais com as abordagens propostas em diversas tarefas de regressão/identificação de sistemas utilizando conjuntos de dados sintéticos, conjuntos de dados reais, conjunto de dados de larga escala e conjunto de dados MIMO.

O **Capítulo 7** é dedicado às conclusões deste trabalho, bem como às orientações para pesquisas futuras sobre questões relacionadas à esta tese.

## 2 MODELOS LOCAIS EM IDENTIFICAÇÃO DE SISTEMAS

Neste capítulo, as etapas que cercam o problema de identificação de sistemas são destacadas. Para este problema, utilizou-se a classe de modelos locais e, dentre estes modelos, os de estrutura fixa não robustos LLM, RBFN e LMN são descritos com detalhes.

### 2.1 Introdução

A descrição matemática de sistemas reais, sejam estes naturais ou construídos pelo ser humano, para entendê-los e/ou resolver problemas a eles relacionados é uma tarefa essencial nas engenharias e ciências. Nos dias atuais, a competitividade de mercado faz com que as indústrias, principalmente, necessitem cada vez mais de análise e controle acurado de seus sistemas, sendo assim, existe uma busca crescente pela obtenção de modelos matemáticos que representem os sistemas de interesse, sejam estáticos ou dinâmicos (KARNOPP *et al.*, 2012).

Sistema é um conjunto de componentes interconectados que apresentam certas relações de causa e efeito e que atuam como um todo com um determinado objetivo. É importante diferenciar um sistema estático de um sistema dinâmico. De maneira geral, o sistema estático é aquele em que as propriedades descritivas do sistema não possuem dependência temporal; ou seja, memória. Já no sistema dinâmico, estas propriedades possuem tal dependência temporal (MONTEIRO, 2006).

O modelo de um sistema é uma representação mental, física, gráfica ou matemática de observações feitas no mundo real. A modelagem de sistemas é fundamental para o desenvolvimento da ciência e tecnologia (NOWAK, 2002). Três abordagens podem ser utilizadas para se identificar um modelo de um sistema conforme ilustra a Figura 1:

- Caixa branca, que consiste da análise física e matemática de um processo utilizando o conhecimento de leis físicas conhecidas;
- Caixa preta, que consiste em obter um modelo matemático de um sistema já existente e em operação utilizando para isto medidas de entradas, saídas e/ou estados do sistema e;
- Caixa cinza que é um método onde se realiza uma união entre técnicas puramente analíticas de modelagem com ferramentas baseadas em dados experimentais das plantas (KERSCHEN *et al.*, 2006).

Com o passar dos anos, com a digitalização de muitos processos de aquisição de dados industriais e a disponibilidade de tais dados bancos de dados de larga escala, houve a



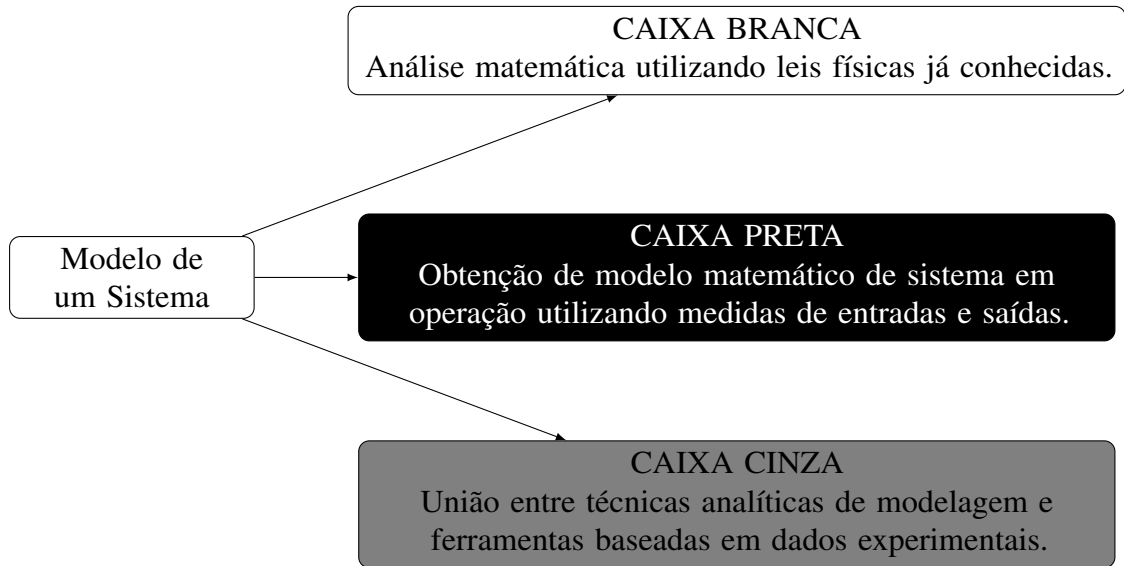


Figura 1 – Abordagens de modelos de sistemas.

necessidade de desenvolver formas de obtenção desses modelos a partir dos dados observados do sistema e não exclusivamente a partir de equações que descrevem o processo. Esse procedimento é conhecido como identificação de sistemas (AGUIRRE, 2015). A busca de representações matemáticas de sistemas a partir de dados obtidos através de experiências e ensaios a serem realizados com o sistema cujo modelo se quer obter é de grande interesse não só para ajudar no entendimento do fenômeno de interesse como também por permitir uma crescente melhora na eficiência dos processos de monitoramento e de controle de plantas industriais. Pode-se ver exemplos provenientes das mais diversas áreas do conhecimento (AGUIRRE, 2004; GIANNAKIS; SERPEDIN, 2001): populações de insetos, bombeamento de água, caldeira industrial (PUTRI, 2019), fornos elétricos, conversores estáticos (BAUMANN *et al.*, 2019), batimentos cardíacos (DURAN-HERNANDEZ *et al.*, 2019), oscilações não lineares, controle de processos (NAUNG *et al.*, 2018), análise dinâmica de plantas existentes, detecção, diagnóstico e prognóstico de falhas, processamento de sinais (FELDMAN; BRAUN, 2017), sistemas biológicos (HE; YANG, 2020), economia (ZHANG *et al.*, 2018) e circuitos elétricos (JAMSHIDI; ALIBEIGI, 2017). Assim, muitas aplicações diferentes deram origem a um grande número de tipos de modelos e técnicas para descrever um sistema. Certas abordagens são específicas para a respectiva aplicação, outras têm um uso mais amplo.

O problema de identificação de um sistema dinâmico do tipo SISO, sem perda de generalidade, é representado na Figura 2 e tem como entradas e saídas  $u(t)$  e  $y(t)$ , respectivamente. As previsões do modelo devem representar o comportamento dinâmico do processo o mais fielmente possível. O desempenho do modelo é normalmente avaliado em termos da função erro

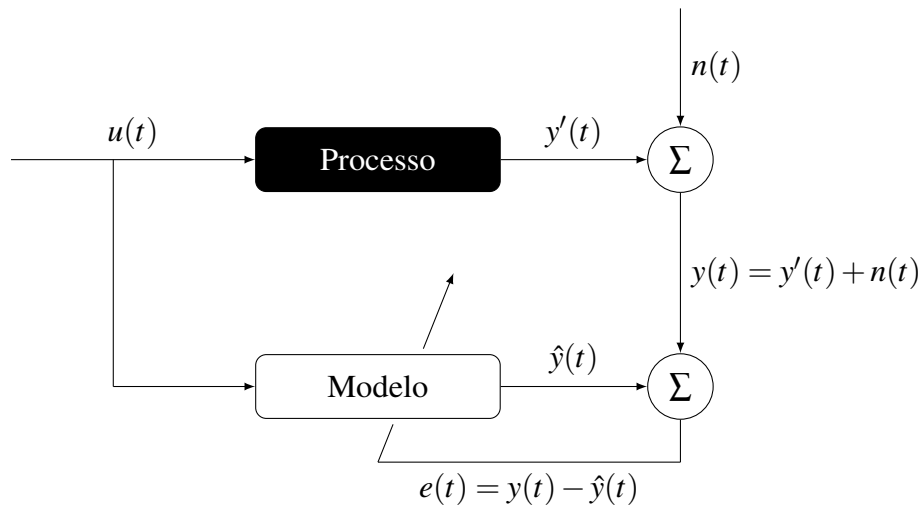


Figura 2 – Exemplo de um problema de identificação de sistemas (SISO) em que o modelo é adaptado para representar o processo com dinâmica desconhecida. Adaptado de Santos (2017).

$e(t)$  que é a diferença entre a saída do processo com ruído  $y(t)$  e a saída do modelo  $\hat{y}(t)$ . Então, esse erro é usado para ajustar os parâmetros do modelo. O problema envolve, no geral, as etapas a seguir (LJUNG, 1998):

1. A primeira etapa envolve a aquisição dos dados. Para este procedimento deve-se escolher quais saídas e entradas são necessárias para construção de um modelo matemático, quantos  $N$  pares de entrada e/ou saída são necessários, quais as taxas de amostragem devem ser usadas, número de canais, assim como a aplicação de técnicas de pré-processamento dos sinais, tais como filtragem de ruído de instrumentação e eliminação de medidas anômalas ou incompletas.
2. A segunda etapa é decidir qual estrutura do modelo  $M(\theta)$  deve ser empregada, onde  $M(\cdot)$  representa o modelo e  $\theta$  representa o vetor de parâmetros do modelo.
3. Em seguida, deve-se escolher os valores dos parâmetros  $\theta$  de tal forma que a saída estimada  $\hat{y}(t)$  do modelo tenda ao valor real de saída medido  $y(t)$ , ou seja, que um índice de desempenho baseado no erro  $e(t)$  tenda a zero. Para isto, deve-se ter que  $M(\theta)$  deve ter sua estrutura adequada para representar o processo real que gerou os dados.
4. Por fim, a última etapa consiste em realizar uma validação e verificação do modelo. Este procedimento consiste em analisar eventuais erros de modelagem, vieses na estimação dos parâmetros e autocorrelação dos resíduos (erros de estimação).

Quanto à escolha do modelo, deve-se normalmente buscar um compromisso entre o custo de se ter o modelo, ou seja, o tempo e o esforço requeridos para obtê-lo e verificar o nível de detalhes do modelo, bem como os benefícios esperados de sua aplicação conforme dito

anteriormente (AGUIRRE, 2004).

Isto posto, uma importante classificação dos modelos matemáticos para sistemas dinâmicos se baseia no conceito de linearidade. Modelos lineares constituem uma classe de modelos com estruturas mais simples, tanto de estimação de parâmetros quanto de validação. Dentre estes, uma série de modelos lineares diferentes estão disponíveis na literatura (LJUNG, 1999; BILLINGS, 2013; AGUIRRE, 2015), tais como autorregressivos (AR, *auto-regressive*), autorregressivos com entrada exógenas (ARX, *auto-regressive with exogenous inputs*), médias móveis (MA, *moving averages*), autorregressivo de médias móveis com entradas exógenas (ARMAX, *auto-regressive moving average with exogenous inputs*), erros na saída (OE, *output error*), resposta finita ao impulso (FIR, *finite impulse response*) e Box-Jenkins (BJ).

Entretanto, a maioria dos sistemas reais são, em algum grau, não-lineares, o que exige aproximações e/ou imposição de restrições para a aplicação de um modelo linear. Desta forma, devido a complexidade da dinâmica dos sistemas envolvidos e pela necessidade de um modelo o mais fiel possível à realidade, os modelos não lineares já vem há algum tempo ganhando espaço na literatura (AGUIRRE, 2019; AYALA *et al.*, 2020). As estruturas de modelos não lineares se diferenciam entre si quanto aos elementos utilizados na composição do vetor de regressores do modelo (SANTOS, 2017).

Dentre os modelos não lineares comuns, destacam-se os seguintes (NELLES, 2020): *nonlinear ARX* (NARX), *nonlinear ARMAX* (NARMAX), *nonlinear OE* (NOE), *nonlinear FIR* (NFIR), *nonlinear orthonormal basis functions* (NOBF), além dos modelos Wiener e Hammerstein. Dentre os modelos dinâmicos não lineares, o modelo NARX cobre uma ampla classe de sistemas não lineares e é normalmente uma das abordagens mais utilizadas na maioria das aplicações de engenharia (NELLES, 2020). Desta forma, o modelo NARX será utilizado para todas as aplicações de identificação de sistemas ao longo desta tese.

Para todas os modelos a serem apresentados neste e nos próximos capítulos, definiu-se o tipo de modelo dinâmico de entrada-saída em que tem-se interesse. Neste sentido, a dinâmica dos sistemas de interesse é considerada adequadamente descrita por um modelo NARX cuja formulação geral obedece à seguinte equação (LJUNG, 1998):

$$y(t) = G(y(t-1), \dots, y(t-L_y), u(t-1), \dots, u(t-L_u)) + n(t), \quad (2.1)$$

em que  $L_u \geq 1$  e  $L_y \geq 1$  denotam as ordens de memória de entrada e saída, respectivamente, e  $n(t)$  é uma variável aleatória do tipo AWGN<sup>1</sup>. A função não linear  $G(\cdot) : \mathbb{R}^{L_u+L_y} \rightarrow \mathbb{R}$  é considerada

<sup>1</sup> *Additive white Gaussian noise*. Ou seja, ruído branco (sem memória) e Gaussiano de média zero e variância  $\sigma_n^2$ .

desconhecida parcial ou totalmente. Os dados observados na forma de séries temporais de pares entrada-saída  $(u(t), y(t)) \in \mathbb{R} \times \mathbb{R}$  utilizados para construir um modelo aproximado  $\hat{G}(\theta)$  para a função  $G(\cdot)$  por meio de qualquer modelo de aproximação de função, tais como redes neurais (LAWRENCE *et al.*, 1996) e aquelas baseadas em *fuzzy* (FOSS; JOHANSEN, 1993; CHUANG, 2007). Qualquer que seja a escolha, o  $t$ -ésimo vetor de regressão de entrada  $\mathbf{x}(t) \in \mathbb{R}^{L_u+L_y}$  é montado pela concatenação de  $L_y$  saídas observadas passadas e  $L_u$  entradas anteriores, tudo isso em um único vetor de regressores da forma

$$\mathbf{x}(t) = \begin{pmatrix} y(t-1) \\ \vdots \\ y(t-L_y) \\ u(t-1) \\ \vdots \\ u(t-L_u) \end{pmatrix}. \quad (2.2)$$

Dessa forma, os dados de de entrada-saída observados devem ser utilizados para construir um modelo aproximado  $\hat{G}(\cdot)$  para a função  $G(\cdot)$ , que é expresso por

$$\hat{y}(t) = \hat{G}(y(t-1), \dots, y(t-L_y), u(t-1), \dots, u(t-L_u)). \quad (2.3)$$

O erro no instante  $t$  é dado por

$$e(t) = y(t) - \hat{y}(t), \quad (2.4)$$

sendo chamado de erro de estimação ou resíduo quando são usados os dados de treinamento; ou erro de predição quando forem usados os dados de teste.

Conforme já mencionado, existem várias maneiras de descrever um sistema a partir dos dados, escolhendo o modelo mais adequado ao problema de interesse. O processo de identificação equivale a selecionar uma estrutura de modelo (global ou local), estimar seus parâmetros e avaliar a qualidade deste modelo (LJUNG, 1998). Nas próximas seções, serão feitas uma descrição e uma discussão sobre os modelos locais a serem utilizados nesta tese, a saber, LLM, RBFN e LMN. A partir destes modelos, segue-se com as propostas de variantes crescentes e robustas.

## 2.2 O modelo LLM

O primeiro modelo local a ser abordado nesta tese é o modelo LLM (WALTER *et al.*, 1990). Inicialmente, uma revisão bibliográfica é realizada. A formulação matemática do modelo

é descrita em seguida.

### 2.2.1 Revisão bibliográfica

O modelo LLM tem sido utilizado na literatura nos mais diversos problemas. Métodos baseados no modelo LLM foram utilizados para o aprendizado da coordenação visuomotora ou coordenação olho-mão de um robô manipulador (RITTER *et al.*, 1992). Littmann e Ritter (1992) apresentaram uma nova arquitetura de rede em cascata baseada na minimização de erro combinada com o modelo LLM. Neste trabalho, o bom desempenho da rede é resultante de várias camadas de LLMs que são treinados para uma determinado fim.

Martinetz *et al.* (1993) utilizaram o mapeamento linear local para a previsão de séries temporais. O modelo também foi utilizado para aproximação de função e identificação de um sistema de um atuador hidráulico com base em medições de entrada-saída (MOSHOU; RAMON, 1997). Moshou *et al.* (1998) adaptaram o modelo LLM como método de reconhecimento de padrões e então aplicaram na classificação de culturas e ervas daninhas.

Souza (2012) aplicou o modelo LLM para utilização em identificação robusta *offline* de sistemas dinâmicos. Neste último trabalho, a sensibilidade do modelo local com relação ao algoritmo de quantização vetorial usado no treinamento do modelo é realizada. A aplicação dessa arquitetura em robótica foi retomada em Fontinele *et al.* (2016) com a utilização da LLM para o aprendizado de cinemática inversa. Uma abordagem similar à do modelo LLM foi usada para realizar a segmentação automática do hipocampo, com o método proposto avaliado em imagens cerebrais de 35 indivíduos (PANG *et al.*, 2017; PANG *et al.*, 2019).

Aplicações mais atuais da rede LLM continuam a surgir na literatura. Miyajima *et al.* (2019) aplicaram mapeamento linear local para solucionar problemas de agrupamento e classificação em Internet das Coisas (*internet of things*, IoT) e Miyajima *et al.* (2020) utilizaram a rede *neural gas* (NG), uma variante do modelo LLM, e a aplicaram na modelagem *fuzzy*.

Com esse apanhado de trabalhos encontrados na literatura, é possível observar que o mapeamento linear local é versátil quanto a utilização e é revisitado desde sua proposição no início da década de 90 até os dias atuais. Assim, este modelo será utilizado para posteriores discussões nessa tese. Para isso, é necessário um detalhamento formal do modelo.

### 2.2.2 Formalização matemática

Como dito anteriormente, a ideia básica do modelo LLM (WALTER *et al.*, 1990) é aproximar uma função multidimensional por meio de um conjunto de hiperplanos, cada um dos quais limitado a uma região local do espaço de entrada  $\mathbb{R}^{n_x}$ , onde  $n_x = L_u + L_y$ . Cada região local está associada a um vetor protótipo e cada vetor protótipo está associado a um filtro linear cujos parâmetros são ajustados por uma variante da regra LMS. As regiões locais são parametrizadas por um número reduzido de vetores protótipos  $\mathbf{w}_j$  de um algoritmo de quantização vetorial qualquer, por exemplo, a rede SOM (KOHONEN, 1982), enquanto os filtros lineares são parametrizados por vetores de coeficientes locais  $\mathbf{a}_j$ .

Formalmente, a quantização vetorial do espaço de entrada  $\chi$  é realizada pelo modelo LLM, com cada neurônio  $j$  possuindo um vetor protótipo  $\mathbf{w}_j \in \mathbb{R}^{n_x}$ ,  $j = 1, \dots, S$ , com  $S$  denotando o número de neurônios e  $n_x = L_u + L_y$ . Associado a cada vetor  $\mathbf{w}_j$ , existe um vetor de coeficientes  $\mathbf{a}_j(t) \in \mathbb{R}^{n_x}$  com caso particular onde  $n_y = 1$  que desempenha o papel dos coeficientes de um modelo ARX local:

$$\mathbf{a}_j(t) = \begin{pmatrix} a_{j,1} \\ \vdots \\ a_{j,n_x} \end{pmatrix}_{n_x \times 1} \quad (2.5)$$

Assim, os parâmetros ajustáveis do modelo LLM são o conjunto de vetores protótipos  $\mathbf{w}_j$  e seus respectivos vetores de coeficientes  $\mathbf{a}_j$ , com  $j = 1, \dots, S$ .

Seguindo a lógica do algoritmo SOM, a escolha do neurônio vencedor baseia-se na menor distância euclidiana do vetor protótipo ao vetor de entrada:

$$j^*(t) = \arg \min \|\mathbf{x}(t) - \mathbf{w}_j(t)\|, \quad (2.6)$$

de modo que o valor predito da saída do modelo baseado em LLM é então calculado como

$$\hat{y}(t) = \mathbf{a}_{j^*}^T(t) \mathbf{x}(t), \quad (2.7)$$

em que  $\mathbf{a}_{j^*}(t)$  é o vetor coeficientes do filtro linear associado ao neurônio vencedor  $j^*(t)$  e é usado para construir uma aproximação local para a saída.

Conforme esperado, o processo de adaptação dos vetores protótipos e do vetor de coeficientes associados a todos os neurônios segue a lógica cooperativa da rede SOM (KOHONEN,

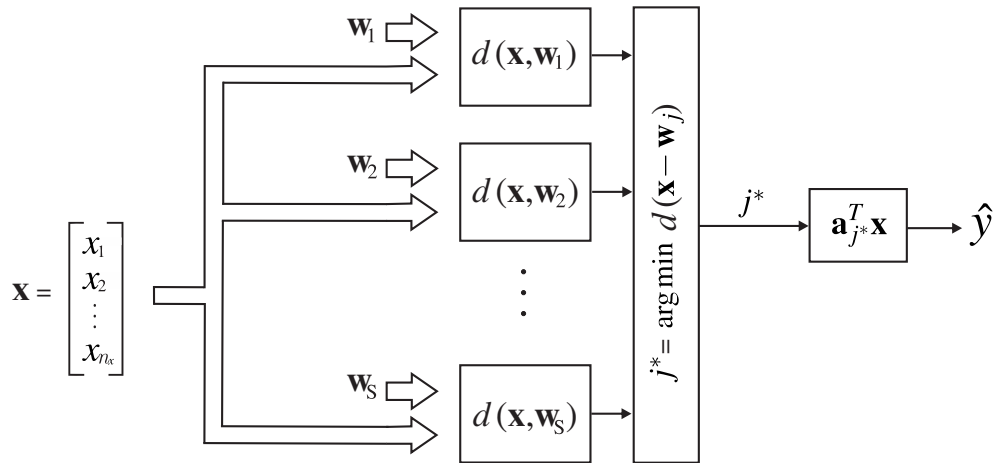


Figura 3 – Representação do modelo LLM.

1982), em que não apenas os parâmetros do neurônio vencedor, mas também os parâmetros dos neurônios vizinhos, são ajustados a cada iteração do algoritmo.

A regra de aprendizado dos vetores protótipos  $\mathbf{w}_j$  é dada por

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)h(j^*, j; t)[\mathbf{x}(t) - \mathbf{w}_j(t)], \quad (2.8)$$

em que  $h(j^*, j; t)$  é a função de vizinhança em relação ao neurônio vencedor. Já a regra de ajuste do vetor de coeficientes  $\mathbf{a}_j(t)$  é uma extensão do algoritmo LMS normalizado (WIDROW, 2005):

$$\mathbf{a}_j(t+1) = \mathbf{a}_j(t) + \alpha'(t)h(j^*, j; t)\Delta\mathbf{a}_j(t), \quad (2.9)$$

em que  $0 < \alpha, \alpha' \ll 1$  são as taxas de aprendizado dos vetores protótipos e dos vetores de coeficientes, respectivamente. O termo de correção  $\Delta\mathbf{a}_j(t)$  é computado como:

$$\Delta\mathbf{a}_j(t) = [y(t) - \mathbf{a}_{j^*}^T(t)\mathbf{x}(t)] \frac{\mathbf{x}(t)}{\varepsilon + \|\mathbf{x}(t)\|^2}, \quad (2.10)$$

tal que  $y(t)$  é a saída observada e  $\varepsilon$  é uma constante de correção para evitar divisão por zero.

Após o detalhamento do algoritmo, pode-se ilustrar o princípio de funcionamento da rede LLM. Com base na Figura 3, pode-se notar que inicialmente o algoritmo particiona o espaço de entrada e atribui a cada espaço um vetor protótipo  $\mathbf{w}_j$ . Encerrada a etapa de estimação, a cada novo vetor de entrada  $\mathbf{x}(t)$  apresentado à rede, há o mapeamento do mesmo em relação ao vetor protótipo mais próximo  $\mathbf{w}_{j^*}(t)$ . Para cálculo da saída estimada  $y(t)$ , é utilizado o vetor de coeficientes  $\mathbf{a}_{j^*}(t)$  associado ao vetor protótipo vencedor.

## 2.3 O modelo RBFN

A rede RBF pertence a uma classe de modelos de redes neurais na qual a ativação de uma unidade oculta é uma função não linear da distância entre o vetor de entrada e seu vetor protótipo. As RBFNs unificam diversas teorias importantes envolvendo aproximação de funções, regularização, interpolação, estimação de densidade, classificação ótima e funções de potencial. Como consequência, os procedimentos de treinamento destas redes costumam ser muito mais rápidos que os métodos usados para treinar redes MLP, por exemplo. Abaixo, realizou-se uma revisão bibliográfica sobre o modelo RBFN e logo após a formulação matemática é apresentada.

### 2.3.1 Revisão bibliográfica

O modelo RBFN é baseado em uma arquitetura clássica de rede neural de uma camada oculta largamente conhecida por pesquisadores da área de inteligência computacional (MOODY; DARKEN, 1989; CHEN *et al.*, 1990; DASH *et al.*, 2016) com várias aplicações em tarefas de regressão (FONTINELE *et al.*, 2016; R; T, 2018; TU *et al.*, 2018; KIANI, 2020) e classificação (MUSAVI *et al.*, 1992; OYANG *et al.*, 2005; WU *et al.*, 2012; SHAO, 2020).

Inúmeras são as aplicações, nas mais diversas áreas, da rede RBF encontradas na literatura. Aqui, alguns trabalhos que utilizam a rede RBF para a tarefa de identificação de sistemas foram destacados por ser o foco desta tese. Neste sentido, Chen *et al.* (1990) investigaram a identificação de sistemas não lineares utilizando funções de base radial. Neste trabalho, um algoritmo de regressão com base em uma decomposição ortogonal da matriz de regressão é empregado para selecionar um conjunto adequado de centros de função de base radial a partir de um grande número de candidatos possíveis.

Ye e Loh (1993) apresentaram uma estrutura de rede neural local denominada rede espaço-temporalmente local, combinando a rede de função de base radial (RBFN) e redes locais recorrentes. Três estruturas locais são propostas e os algoritmos são comparados para identificação de sistemas dinâmicos não lineares. A dinâmica do sistema pode ser modelada com o aprendizado rápido da estrutura da rede neural proposta.

Em Cho e Wang (1996), os autores descrevem um sistema neuro-*fuzzy* com capacidade adaptativa para extrair regras *fuzzy* SE-ENTÃO de dados de amostra de entrada e saída por meio do aprendizado. O sistema proposto, denominado sistema *fuzzy* adaptativo e baseado em função de base radial (RBF), emprega as funções gaussianas para representar as funções de



pertinência das premissas.

Nelles e Isermann (1996) propõem um algoritmo denominado *local linear model trees* (LOLIMOT) para identificação de sistemas dinâmicos não lineares com modelos lineares locais. O espaço de entrada é particionado por um algoritmo de construção de árvore. Os modelos locais são interpolados por funções de base local sobrepostas. A estrutura resultante é equivalente a um sistema *fuzzy* Takagi-Sugeno. O algoritmo LOLIMOT é simples, fácil de implementar e rápido. Além disso, esta abordagem tem a seguinte propriedade atraente: não está sujeita à maldição da dimensionalidade, pois revela entradas irrelevantes e detecta entradas que influenciam a saída.

Zemouri *et al.* (2003) propõem uma rede RBF Recorrente (RRBFN) que pode ser aplicada ao monitoramento dinâmico e prognóstico. A aplicação diz respeito a um problema de prognóstico industrial. A identificação do sistema não linear foi feita utilizando os dados do forno a gás Box e Jenkins (BOX *et al.*, 2015). A rede é capaz de prever as duas séries temporais e obtém bons resultados para a identificação do sistema não linear. A vantagem da rede RRBF proposta é combinar a flexibilidade de aprendizagem do modelo RBF com os desempenhos dinâmicos da recorrência local dada pelos neurônios com laços de realimentação.

Em Xue *et al.* (2012), um novo esquema de adaptação de parâmetros de controle é introduzido no algoritmo de evolução diferencial clássico (STORN; PRICE, 1997). O método combina evolução diferencial modificada (*modified differential evolution*, MDE) e redes RBF, que podem autoconfigurar a estrutura de redes RBF e obter os parâmetros do modelo. A estrutura e os parâmetros da rede RBF podem ser determinados simultaneamente com base em dados de entrada e saída sem conhecimento a priori.

Oliveira *et al.* (2017) propuseram um algoritmo de aprendizagem baseado em gradiente para uma RBFN. Uma abordagem de estágio único é empregada para o processo de aprendizagem, onde os parâmetros livres da rede (a posição dos centróides, a largura das funções de base e os pesos de saída) são atualizados por meio de um método supervisionado. A capacidade de identificação precisa é examinada pelo uso de dois conjuntos de dados não lineares e o desempenho da proposta do método é comparado com as técnicas tradicionais.

Naing *et al.* (2020) apresentaram uma síntese de uma rede RBF utilizando algoritmo genético para determinar o número de neurônios em camada oculta e os parâmetros da rede neural RBF. Essa adaptação foi utilizada para o processo de identificação de sistemas de sistemas não lineares.

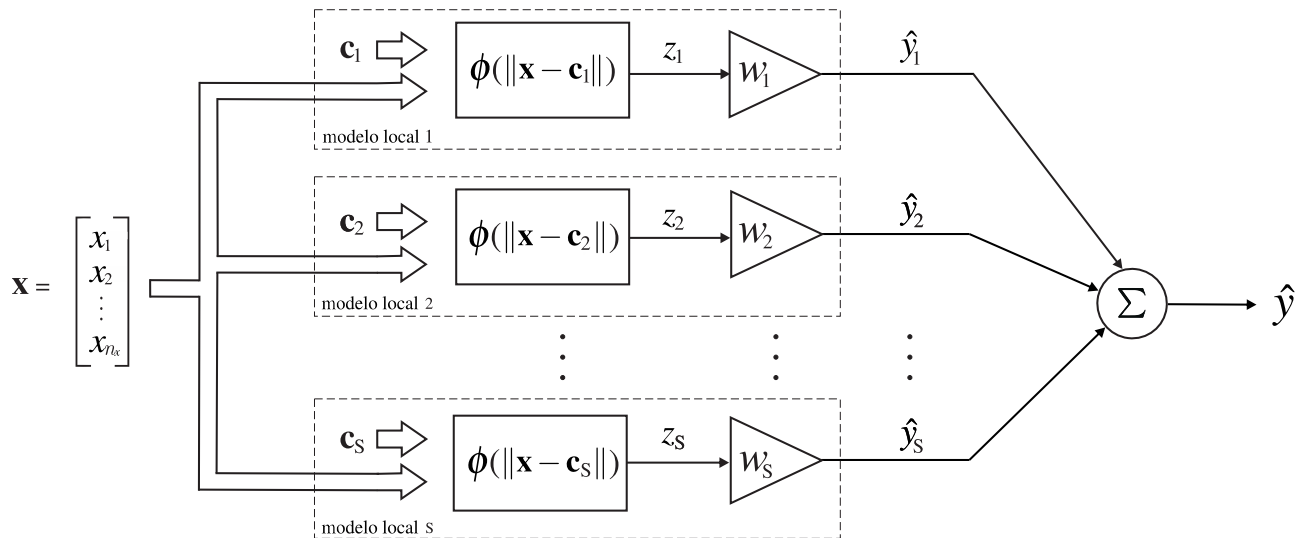


Figura 4 – Representação do modelo RBF.

Ayala *et al.* (2020) aplicaram algoritmos coevolucionários e a rede RBF com o objetivo de estimar parâmetros para modelos de amortecedor magneto-reológico, de reator de polimerização e de micromanipulador robótico piezoelétrico.

### 2.3.2 Formalização matemática

Com essa breve revisão sobre trabalhos que utilizaram a RBFN para a tarefa de identificação de sistemas, nota-se que a utilização da rede para a tarefa em questão vem desde sua proposição. Assim, faz-se necessária uma descrição formal de como a rede funciona. O projeto de uma rede neural é visto como um problema de ajuste de curva multidimensional. De forma particular, uma rede RBF tem componentes fundamentais, a saber (MOODY; DARKEN, 1989):

1. As unidades de entrada que conectam a rede ao seu ambiente.
2. Uma única camada de entrada que promove uma transformação não linear do espaço de entrada para o espaço oculto.
3. A camada de saída é linear, fornecendo a resposta da rede ao sinal de ativação aplicado na entrada.

Os neurônios ocultos da segunda camada, designados como funções de base radial, possuem uma função de ativação não linear, enquanto, os neurônios de saída tem, em geral, uma função de ativação linear conforme já mencionado. Tipicamente, em uma rede RBF, a forma das funções de base é escolhida a priori, com a função gaussiana sendo uma escolha muito comum e suficientemente adequada a uma gama de problemas. A Figura 4 ilustra uma representação

do modelo RBF, com destaque aos seguintes elementos: vetor de entrada  $\mathbf{x}$ , camada oculta que promove a transformação não linear do espaço de entrada para o espaço oculto e uma camada de saída linear.

Isto posto, o problema consiste, então, em localizar os centros e outros parâmetros das funções de base e determinar os pesos de saída usando o conjunto de treinamento. No modelo RBF, considere o centro da  $j$ -ésima função de base como  $\mathbf{c}_j \in \mathbb{R}^{n_x}$ ,  $j = 1, \dots, S$ . A  $j$ -ésima função de base é composta de dois elementos:

- Uma métrica de distância  $d_j(\mathbf{x})$ ; em geral, a distância euclidiana ou a de Mahalanobis:

$$d_j(\mathbf{x}) = \text{dist}(\mathbf{x}; \mathbf{c}_j). \quad (2.11)$$

- A ativação da  $j$ -ésima função de base radial:

$$z_j = \phi(d_j(\mathbf{x})), \quad (2.12)$$

com  $j = 1, \dots, S$ , e  $S$  denotando o número de neurônios ocultos.

As funções de base radial são uma classe especial de funções. A sua característica principal é que sua resposta diminui (aumenta) monotonamente com o aumento (diminuição) da distância ao seu centro  $\mathbf{c}_j$ . O centro, a métrica de distância e a forma da função radial são parâmetros do modelo.

Como já mencionado, uma métrica de distância bastante comum é a distância euclidiana quadrática dada por

$$d_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}_j\|^2, \quad (2.13)$$

e como escolha rotineira, que fez-se uso nesta tese, como função de base usa-se a função gaussiana:

$$\begin{aligned} z_j &= \exp\left(-\frac{d_j(\mathbf{x})}{2\sigma_j^2}\right) \\ &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2\sigma_j^2}\right), \end{aligned} \quad (2.14)$$

com  $\sigma$  denotando o raio (ou largura) da função de base.

A rede RBF é projetada para realizar um mapeamento não linear do espaço de entrada para o espaço oculto, seguido de um mapeamento linear do espaço oculto para o espaço de saída. Existem várias maneiras de projetar a rede RBF, mas nesta tese opta-se pelo procedimento descrito por Moody e Darken (1989). Estes autores recomendam treinar uma rede RBF em três etapas executadas sequencialmente:

1. Determinação das posições dos centros por meio de algoritmo de quantização vetorial, tal como a rede SOM (KOHONEN, 1982);
2. Determinação do raio  $\sigma$  das funções de base. Para este fim, o raio de um neurônio tem seu valor definido como metade da distância entre o centro do neurônio  $j$  e o centro mais próximo;
3. Cálculo dos pesos de saída. Se houver uma única saída (sistema SISO), tem-se um vetor de pesos de saída. Caso haja várias saídas (sistema MIMO), tem-se uma matriz de pesos de saída.

Para aplicações que envolvem aproximação de funções, como identificação de sistemas, normalmente a função de ativação do neurônio de saída é linear<sup>2</sup>:

$$\hat{y}(t) = \mathbf{w}^T(t)\mathbf{z}(t) = \sum_{j=1}^S w_j(t)z_j(t), \quad (2.15)$$

em que  $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_S]^T$  é o vetor de pesos do neurônio de saída e  $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_S]^T$  é o vetor com as ativações das funções de base.

O vetor de pesos  $\mathbf{w}$  pode ser atualizado recursivamente. Neste caso, a regra LMS é usada para este fim; ou seja

$$\begin{aligned} \mathbf{w}(t+1) &= \mathbf{w}(t) + \alpha(t)e(t)\mathbf{z}(t), \\ &= \mathbf{w}(t) + \alpha(t)[y(t) - \hat{y}(t)]\mathbf{z}(t), \end{aligned} \quad (2.16)$$

em que  $e(t)$  é o erro de estimação e  $0 < \alpha \leq 1$  é a taxa de aprendizado do vetor de pesos  $\mathbf{w}$ .

**Observação 1** - O modelo RBFN pode ser entendido como uma implementação neural de um modelo *fuzzy* de Takagi-Sugeno (TS) de ordem zero (TAKAGI; SUGENO, 1985), com as saídas das  $S$  funções de base expressando as ativações das  $S$  regras.

Com isso, finalizou-se aqui a descrição da rede RBF. Na próxima Seção, o modelo LMN que pode ser entendida como uma versão generalizada da rede RBF será descrito.

---

<sup>2</sup> Aqui assumiu-se apenas um neurônio de saída,  $n_y = 1$ .

## 2.4 O modelo LMN

A idéia de usar modelos locais em uma rede RBF foi sugerida em vários trabalhos que remontam ao início da década de 90. Uma revisão bibliográfica e a formulação matemática do modelo são realizadas a seguir.

### 2.4.1 Revisão bibliográfica

Jones *et al.* (1990) propuseram um modelo muito similar ao LMN para uso em aplicações de estimação de funções e predição de séries temporais. A predição de séries temporais caóticas utilizando uma arquitetura similar a de Jones *et al.* (1990) também foi vista em Stokbro *et al.* (1990). Em Barnes *et al.* (1991), foi relatado o uso da arquitetura supracitada para modelagem e controle de processos. Já em Jacobs *et al.* (1991), notou-se o uso para reconhecimento de padrões, mais especificamente, de vogais. O modelo LMN foi, de fato, introduzido por Johansen e Foss (1992) como uma generalização da rede RBF.

Notou-se na revisão de literatura que aplicações mais recentes deste modelo local continuam a surgir. Teslic *et al.* (2011) descrevem uma nova abordagem de identificação iterativa, onde a aprendizagem supervisionada e não supervisionada são combinadas para otimizar a estrutura do modelo LMN. Com o objetivo de ajustar os centros é aplicado o agrupamento *fuzzy*, ou seja, aprendizagem não supervisionada.

Hametner e Jakubek (2013) propõem um algoritmo de otimização para o treinamento online do modelo LMN o que permite a estimação recursiva dos parâmetros do modelo. Em König *et al.* (2014), o modelo local em questão é utilizado para obter modelos de baterias em tempo real. Costa *et al.* (2015) apresentam uma avaliação experimental de uma classe de controladores preditivos baseados em LMN onde os modelos locais são utilizados para descrever o processo não linear em diferentes pontos de operação. O modelo LMN também foi utilizado para controle preditivo de plantas de pH (MÜNKER *et al.*, 2017), identificação de sistemas dinâmicos (BELZ *et al.*, 2017; MAIER *et al.*, 2018), como estratégia de controle preditivo de uma turbina (ZHU *et al.*, 2019) e modelagem aerodinâmica (WEINSTEIN; HUBBARD, 2020).

### 2.4.2 Formalização matemática

Esta abordagem pode ser vista como um sistema não linear em um conjunto de submodelos lineares locais que são, então, combinados linearmente pelas ativações das funções

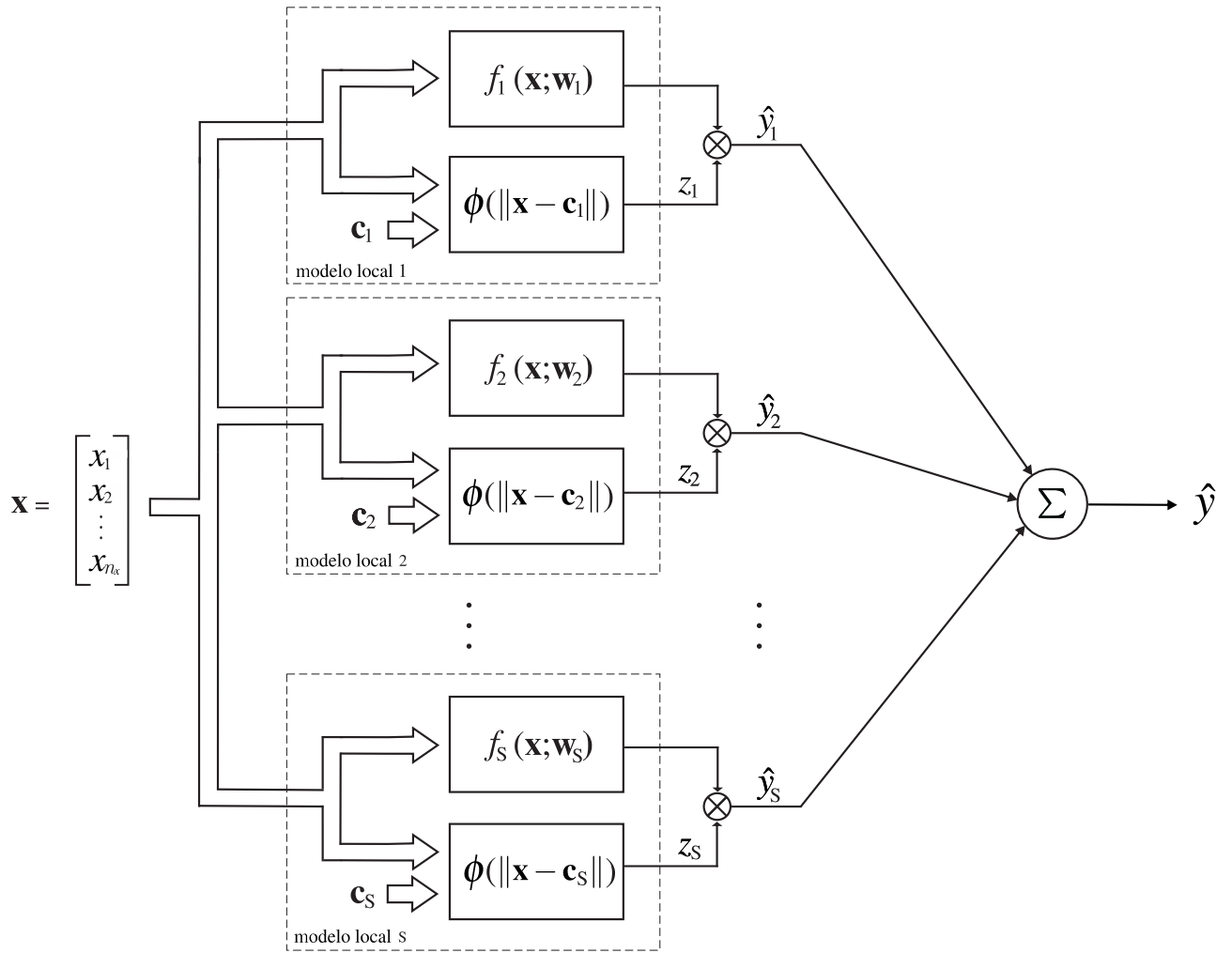


Figura 5 – Representação do modelo LMN.

de base radial associadas. Isso significa que um número menor de modelos locais pode abranger áreas maiores do espaço de entrada, quando comparado com a rede RBF simples. Uma representação da rede LMN é ilustrada na Figura 5. Nota-se que a diferença entre o modelo LMN e o modelo RBF está na redução de  $f_j(\mathbf{x}, \mathbf{w}_j; t)$  a  $\mathbf{w}_j(t)$ .

Assim, a Equação (2.15) do modelo RBF é generalizada para envolver não apenas um peso constante associado a cada função de base, mas também uma função do vetor de entrada  $\mathbf{x}$  e de um vetor pesos  $\mathbf{w}_j$  associado a cada função de base, passando a ser escrita como

$$\hat{y}(t) = \sum_{j=0}^S z_j(t) f_j(\mathbf{x}, \mathbf{w}_j; t), \quad (2.17)$$

em que  $z_j(t)$  é calculado conforme Equação (2.12) e  $f_j(\mathbf{x}; \mathbf{w}_j)$  é a função local associada à

$j$ -ésima função de base, como por exemplo uma função linear <sup>3</sup>:

$$f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j^T(t) \mathbf{x}(t) + w_{j,0}(t), \quad (2.18)$$

$$= w_{j,1}(t)x_1(t) + \cdots + w_{j,n_x}(t)x_{n_x}(t) + w_{j,0}(t). \quad (2.19)$$

A rede LMN reduz-se à rede RBF quando a função local é dada por  $f_j(\mathbf{x}, \mathbf{w}_j; t) = w_{j,0}(t)$ . A regra LMS normalizada (WIDROW; LEHR, 1990) também é utilizada para adaptar os pesos de saída  $\mathbf{w}_j$

$$\begin{aligned} \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) + \alpha(t)e(t) \frac{\mathbf{x}(t)}{\varepsilon + \|\mathbf{x}(t)\|^2}, \\ &= \mathbf{w}_j(t) + \alpha(t)[y(t) - \hat{y}(t)] \frac{\mathbf{x}(t)}{\varepsilon + \|\mathbf{x}(t)\|^2}, \end{aligned} \quad (2.20)$$

tal que  $e(t) = y(t) - \hat{y}(t)$  é o erro de estimação,  $0 < \varepsilon \ll 1$  é uma pequena constante para evitar a divisão por zero. A norma quadrática de  $\mathbf{x}(t)$  é comumente implementada como  $\|\mathbf{x}(t)\|^2 = \mathbf{x}^T(t)\mathbf{x}(t)$  por uma questão de menor custo computacional.

**Observação 2** - Conforme mencionado, o modelo LMN foi introduzido por Johansen e Foss (1992) como uma generalização da rede RBF (CHEN *et al.*, 1990). Ele pode ser visto como a implementação de uma decomposição do mapeamento global de entrada-saída não linear em um conjunto de submodelos locais que são, então, agregados de maneira suave pelas ativações das funções de base. O processo de suavização permite que um número menor de modelos locais cubra áreas maiores do espaço de entrada do que o modelo RBFN padrão.

**Observação 3** - Conforme mencionado na Observação 1, o modelo RBFN pode ser pensado como uma espécie de implementação neural de um modelo TS de ordem 0. Da mesma forma, o modelo LMN pode ser entendido como sua implementação de 1ª ordem, com as funções de base desempenhando o papel das regras do modelo TS e a função local  $f_j(\mathbf{x}; \mathbf{w}_j) = \mathbf{w}_j^T \mathbf{x}$  sendo os consequentes das regras.

## 2.5 Considerações Finais

Os modelos descritos neste capítulo possuem estrutura fixa, ou seja, quantidade de unidades locais fixa. Vale ratificar que todos os modelos compartilham da limitação de especificar a priori a quantidade de modelos locais. Essa limitação será resolvida com a utilização de modelos crescentes (*growing models*). Aqui, os modelos locais serão inseridos de maneira progressiva até

<sup>3</sup> Na verdade, como existe o bias, esta é uma função afim e não uma função linear.

que um requisito escolhido pelo usuário seja alcançado. Isso é importante pois a dinâmica do sistema pode mudar ao longo do tempo e o modelo pode precisar adaptar sua estrutura para lidar com as novas demandas da tarefa. No Capítulo 3, os modelos crescentes utilizados nessa tese serão descritos e discutidos e a proposta para integrar a função de crescimento ao modelo LMN será apresentada.



### 3 MODELOS LOCAIS CRESCENTES

Neste capítulo, os modelos crescentes utilizados nesta tese serão descritos. Dentre estes, estão duas das propostas deste trabalho, a saber: RAN-LMS e G-LMN.

#### 3.1 Introdução

Conforme mencionado na introdução desta tese, uma desvantagem dos modelos locais descritos no Capítulo 2 é a necessidade de especificar o número de submodelos antes do treinamento. Essa limitação pode ser resolvida com a utilização de modelos incrementais, ou seja, os submodelos são progressivamente inseridos até que um determinado requisito de desempenho seja alcançado.

Em Ferreira e Araújo (2020), aprender incrementalmente significa que os pesos de um sistema de aprendizagem são atualizados sem degradar o conhecimento anterior. Assim, três questões precisam ser consideradas:

1. um cenário de memória limitada de tal forma que, após uma nova amostra ser aprendida, ela é descartada e não pode ser reutilizada;
2. as distribuições de entrada e saída são desconhecidas;
3. essas distribuições podem variar no tempo (SCHAAL; ATKESON, 1998).

Ferreira e Araújo (2020) exploram diferentes abordagens para realizar a aproximação de funções com base no mapa auto-organizável com campo receptivo adaptativo local (*local adaptive receptive fields self-organizing map*, LARFSOM). Os modelos locais são construídos pelo cálculo entre a saída associada ao neurônio vencedor e o vetor diferença entre o vetor de entrada e o vetor de peso. Esses modelos são combinados usando uma soma ponderada para produzir o valor aproximado final. A topologia é adaptada de forma auto-organizável e os vetores de peso são ajustados em um algoritmo de aprendizado não supervisionado modificado para problemas supervisionados.

Em suma, busca-se um modelo com os seguintes requisitos:

- estrutura online crescente (ou seja, não há necessidade de definir o número de unidades ocultas de antemão),
- regra de atualização recursiva rápida; ou seja, com poucas operações matemáticas,
- com poucos requisitos de memória, e
- resiliente a *outliers*.

Nesse sentido, tanto a acurácia no desempenho quanto a simplicidade de implementação são os principais pontos a serem considerados. O estado da arte em algoritmos de estimação recursiva robusta a *outliers* voltados para identificação de sistemas possuem alguns dos recursos listados acima, mas não todos. A maioria deles utiliza o algoritmo RLS, que requer muito mais espaço de memória do que o algoritmo LMS devido ao uso de matrizes de covariância, uma para cada neurônio da rede.

Isto posto, neste capítulo, destacam-se dois modelos que satisfazem boa parte dos requisitos desejados e já estão bem consolidados na literatura. Além disso, a proposta de modelo incremental desta tese é descrita. Os modelos que serão descritos estão listados abaixo.

- Mapeamento linear local incremental (*incremental local linear mapping*, INC-LLM) (FRITZKE, 1995a) que é uma versão crescente do modelo LLM descrito no capítulo anterior;
- Rede de alocação de recursos (*resource allocating network*, RAN) (PLATT, 1991) que pode ser considerado um modelo crescente baseado no modelo RBF;
- *Growing local model network* (G-LMN) que é uma das propostas dessa tese e tem como inspiração o modelo LMN e o modelo RAN.

### 3.2 Mapeamento linear local incremental

Abordagens baseadas no modelo LLM geralmente assumem um número fixo de centros que são distribuídos no espaço de entrada por algum método de quantização vetorial. Um modelo de rede incremental para aprendizagem supervisionada foi proposto em Fritzke (1995b). Nessa abordagem, as informações de erro obtidas durante o treinamento são utilizadas para determinar quando e onde inserir novas unidades. A teoria da rede GNG (*growing neural gas*) (FRITZKE, 1995a) foi utilizada para dar um caráter incremental ao modelo LLM.

O modelo GNG descrito em Fritzke (1995a) não é supervisionado e insere novas unidades para reduzir a distância entre o posicionamento dos vetores protótipos e o vetor de entrada. Por esta razão, o erro de distorção é acumulado localmente e novas unidades são inseridas perto da unidade com maior erro acumulado.

Isto posto, é necessário antes descrever o algoritmo GNG (FRITZKE, 1995a) que é um algoritmo de quantização vetorial, mas quando usado com a regra de Hebb é capaz de preservar a topologia dos dados de entrada de modo similar à rede SOM. A rede GNG constrói, de forma incremental, uma representação gráfica de um determinado conjunto de dados que é

$n$ -dimensional. O método GNG distribui um conjunto de centros em  $\mathbb{R}^{n_x}$ . Isso é parcialmente feito em etapas de adaptação, mas principalmente por interpolação de novos centros entre os já existentes. Entre dois centros pode haver um *link* indicando vizinhança indicando vizinhança em  $\mathbb{R}^{n_x}$ . Esses *links* são utilizados para interpolação e são inseridos de acordo com as regras do aprendizado de Hebb (MARTINETZ, 1993) durante as etapas de adaptação.

A rede GNG começa com dois neurônios  $a$  e  $b$  em posições aleatórias  $w_a$  e  $w_b$ . Para o vetor de entrada  $\mathbf{x}(t)$ , deve-se encontrar as duas unidades mais próximas  $s_1$  e  $s_2$  utilizando a Equação (2.6). Após isso, deve-se incrementar a idade de todos os *links* que saem de  $s_1$ . Agora, pode-se adicionar a distância quadrática entre o sinal de entrada  $\mathbf{x}(t)$  e a unidade mais próxima no espaço de entrada ( $\mathbf{w}_{s_1}$ ) para uma variável de erro local de  $s_1$  dada por

$$\Delta e_{s_1}(t) = \|\mathbf{x}(t) - \mathbf{w}_{s_1}(t)\|^2. \quad (3.1)$$

Agora, move-se o neurônio de índice  $s_1$  e seus vizinhos topológicos diretos conforme as seguintes regras:

$$\Delta \mathbf{w}_{s_1}(t) = \varepsilon_b(\mathbf{x}(t) - \mathbf{w}_{s_1}(t)), \quad (3.2)$$

$$\Delta \mathbf{w}_n(t) = \varepsilon_n(\mathbf{x}(t) - \mathbf{w}_n(t)), \quad (3.3)$$

em que  $0 < \varepsilon_b \leq 1$  e  $0 < \varepsilon_n \leq 1$  são taxas de aprendizado para a unidades  $s_1$  e seus  $n$  vizinhos topológicos.

Se  $s_1$  e  $s_2$  estiverem conectados por *link*, define-se a idade desse *link* como zero. Se o *link* não existe, deve ser criado. Removem-se os *links* com uma idade maior que  $a_{max}$ . Se isso resultar em unidades sem *links*, estas devem ser removidas também. Se o número de sinais de entrada fornecido até agora for um múltiplo inteiro de um parâmetro  $\lambda$ , insire-se uma nova unidade conforme o procedimento descrito a seguir.

- Determina-se a unidade  $q$  com o maior erro acumulado.
- Insire-se uma nova unidade  $r$  entre  $q$  e seu vizinho  $f$  com o maior erro acumulado;

$$\mathbf{w}_r = 0,5(\mathbf{w}_q + \mathbf{w}_f). \quad (3.4)$$

- Insire-se *links* conectando a nova unidade  $r$  com unidades  $q$  e  $f$ , e remova o *link* original entre  $q$  e  $f$ .
- Atualizam-se as variáveis de erro de  $q$  e  $f$  multiplicando-as por uma constante  $0 < \phi \leq 1$ .
- Inicia-se a variável de erro de  $r$  com o novo valor da variável de erro de  $q$ .

Com isso, decrementam-se todas as variáveis de erro multiplicando-as por uma constante de decaimento  $d$ . Todos estes procedimentos devem ser realizados até um critério de parada (por exemplo, tamanho da rede ou alguma medida de desempenho) ser cumprido.

Nota-se que o modelo GNG foi criado com o intuito de ser não supervisionado e insere novas unidades para reduzir o erro médio entre os vetores de entrada e os vetores protótipos. Fritzke (1995b) descreveu como esse princípio poderia ser utilizado para o aprendizado supervisionado. Primeiro tem-se que definir qual é a saída da rede (que não era necessário para o aprendizado não supervisionado). Então, faz-se uso da diferença entre a saída real e saída desejada para orientar as inserções de novas unidades. O problema em questão é aproximar uma função  $G(\cdot) : \mathbb{R}^{L_u+L_y} \rightarrow \mathbb{R}^{n_y}$ , dado pares de entrada-saída, com o caso particular onde  $n_y = 1$ .

Associado a cada neurônio  $j$  da rede (posicionado em  $\mathbf{w}_j$  no espaço de entrada), tem-se uma saída  $\hat{y}_j$  e um vetor  $\mathbf{a}_j$  de dimensão  $n_x \times 1$  (ver Equação (2.5)) associada a esta saída, onde  $n_x = L_u + L_y$ . O escalar  $\hat{y}_j$  é a saída da rede para os casos em que os vetores de entrada coincidem com um dos centros, ou seja,  $\mathbf{x}(t) = \mathbf{w}_j$ . Para um vetor de entrada qualquer, o centro mais próximo  $s_1$  é determinado conforme a Equação (2.6) e a saída  $\hat{y}(t)$  da rede é calculada da seguinte forma:

$$\hat{y}(t) = \hat{y}_{s_1}(t) + \mathbf{a}_{s_1}^T(t)(\mathbf{x}(t) - \mathbf{w}_{s_1}(t)), \quad (3.5)$$

em que o termo  $\hat{y}_{s_1}$  é uma primeira aproximação local da saída com o segundo termo promovendo uma correção de primeira ordem, à guisa de uma expansão de Taylor em torno do ponto  $\hat{y}_{s_1}$ .

Agora, tem-se que promover uma ligeira mudança no algoritmo GNG original para permitir o ajuste adequado dos modelos lineares locais, uma vez que se está interessado em reduzir o erro quadrático médio. Muda-se então a Equação (3.1) do modelo GNG para

$$\Delta e_{s_1}(t) = |y(t) - \hat{y}(t)|^2, \quad (3.6)$$

em que  $|\cdot|$  denota o valor absoluto.

A Equação (3.6) significa que agora o erro é acumulado localmente em relação à função a ser aproximada. Novas unidades são inseridas quando essa aproximação for considerada ruim.

Os mapeamentos lineares locais  $\mathbf{a}_j$  associados às unidades da rede são inicialmente definidos de forma aleatória. Em cada etapa de adaptação, o par de dados entrada-saída é utilizado duas vezes: (i) o vetor de entrada é utilizado para a adaptação do centro e (ii) todo o

par é utilizado para melhorar o vetor de coeficientes  $\mathbf{a}_{s_1}$  do centro mais próximo  $s_1$ . Isso é feito utilizando, mais uma vez, a regra LMS (WIDROW, 2005):

$$\hat{y}_{s_1}(t+1) = \hat{y}_{s_1}(t) + \alpha e_{s_1}(t), \quad (3.7)$$

$$\mathbf{a}_{s_1}(t+1) = \mathbf{a}_{s_1}(t) + \alpha e_{s_1}(t)(\mathbf{x}(t) - \mathbf{w}_{s_1}), \quad (3.8)$$

onde  $0 < \alpha \ll 1$  é a taxa de aprendizado.

Quando uma nova unidade  $r$  é inserida, um mapeamento linear local é interpolado entre os vizinhos  $q$  e  $f$ :

$$\hat{y}_r(t) = 0,5(\hat{y}_q(t) + \hat{y}_f(t)) \quad (3.9)$$

$$\mathbf{a}_r(t) = 0,5(\mathbf{a}_q(t) + \mathbf{a}_f(t)). \quad (3.10)$$

Um critério de parada deve ser definido para terminar o processo de crescimento. Isso pode ser escolhido arbitrariamente, dependendo da aplicação. Uma escolha possível é observar o desempenho da rede em um conjunto de validação durante o treinamento e parar quando esse desempenho começar a diminuir. Alternativamente, o erro no conjunto de treinamento pode ser usado ou simplesmente o número de unidades na rede se, por algum motivo, um tamanho de rede específico for desejado. Um pseudocódigo para esse modelo é apresentado no Algoritmo 1.

O modelo proposto por Fritzke (1995b) se tornou um propulsor para a abordagem incremental de mapeamentos lineares locais. Alguns trabalhos encontrados na literatura foram propostos baseados no modelo apresentado (VACHKOV, 2004; HE *et al.*, 2011; FUANGKHON, 2014; LU *et al.*, 2017). Em Vachkov (2004), a proposta é bem semelhante à de Fritzke (1995b), a cada iteração, a informação de realimentação do erro de aproximação do modelo atual é utilizada para tomar a decisão de inserção de novos modelos locais na área de entrada com maior erro. He *et al.* (2011) propuseram uma estrutura de aprendizagem incremental adaptativa chamada de *adaptive incremental learning* (ADAIN), que é capaz de aprender a partir dos dados, acumulando experiência ao longo do tempo e usando esse conhecimento para melhorar o desempenho futuro de aprendizagem e previsão. Em Lu *et al.* (2017), os autores apresentam uma nova estrutura de aprendizagem de mapeamentos lineares baseados em *deep learning* para reconstrução de imagens. Nesta abordagem, o modelo é capaz de aprender a função de mapeamento sem negligenciar a natureza multicamadas de algumas aplicações.

---

**Algoritmo 1: Modelo INC-LMM.**


---

**Início:**

Definir:  $a_{max}$ ,  $d$ ,  $\varepsilon_b$ ,  $\varepsilon_n$ ,  $\delta_{min}$ ,  $N$ ,  $\lambda$  e  $\alpha$ .

Atribuir:  $S = 2$ .

**Iteração:** apresentação dos  $i$  pares entrada-saída

**1. Busca dos centros mais próximos  $s_1$  e  $s_2$ .**

**2. Incremento das idades dos *links* que saem de  $s_1$ .**

**3. Predição:** Para cada vetor de entrada  $\mathbf{x}(t)$ , calcular

$$3.1. \hat{y}(t) = \hat{y}_{s_1}(t) + \mathbf{a}_{s_1}^T(t)(\mathbf{x}(t) - \mathbf{w}_{s_1}(t));$$

**4. Cálculo e acúmulo de erro local:**

$$4.1. \Delta e_{s_1}(t) = |y(t) - \hat{y}(t)|^2$$

**5. Atualização de  $\hat{y}_{s_1}$  e  $a_{s_1}$  :**

$$5.1. \hat{y}_{s_1}(t+1) = \hat{y}_{s_1}(t) + \alpha e_{s_1}(t);$$

$$5.2. \mathbf{a}_{s_1}(t+1) = \mathbf{a}_{s_1}(t) + \alpha e_{s_1}(t)(\mathbf{x}(t) - \mathbf{w}_{s_1}(t));$$

**6. Movimentação de  $s_1$  e dos seus vizinhos topológicos  $n$ :**

$$6.1. \Delta \mathbf{w}_{s_1}(t) = \varepsilon_b(\mathbf{x}(t) - \mathbf{w}_{s_1}(t));$$

$$6.2. \Delta \mathbf{w}_n(t) = \varepsilon_n(\mathbf{x}(t) - \mathbf{w}_n(t));$$

**7. Criação ou remoção de *links*:**

**SE**  $s_1$  e  $s_2$  estão conectados por *link*,

**ENTÃO** Defina idade do *link* = 0;

**SENÃO** Crie o *link*.

Remova os *links* com idade  $> a_{max}$

**8. Decisão se o modelo deve alocar nova unidade  $r$ :**

**SE**  $\text{MOD}(i, \lambda) = 0$  e  $S \leq N$ , **ENTÃO**

{ % Crescer

8.1. Determine a unidade  $q$  com maior erro acumulado  $\Delta e_q = \max(\Delta e)$  ;

8.2. Insira a unidade  $r$  entre  $q$  e seu vizinho  $f$  com  $\max(\Delta e)$  usando a Equação (3.4);

8.3. Insira  $\hat{y}_r$  e  $\mathbf{a}_r$  usando as Equações (3.9) e (3.10);

8.4. Insira *links* de  $r$  para  $q$  e  $f$ ;

8.5. Remova *link* de  $q$  para  $f$ ;

8.6.  $\Delta e_r = \Delta e_q$ ;

}

**9. Atualização dos erros:**

9.1.  $\Delta e_q = d\Delta e_q$ ;

9.2.  $\Delta e_f = d\Delta e_f$ ;

---

### 3.3 Proposta 1: Modelo RAN-LMS

A rede de alocação de recursos (*resource allocation network*, RAN) foi proposta por Platt (1991) e é uma rede de duas camadas, que se assemelha a uma rede RBF, porém com uma estrutura crescente. A primeira camada consiste em unidades ocultas que respondem apenas a uma região local do espaço de entrada, ou seja, seus neurônios possuem um campo receptivo localizado. A segunda camada contém unidades de saída que agregam saídas dessas unidades e criam a função que se aproxima do mapeamento de entrada-saída em todo o espaço.

Esta rede visa construir representações locais do mapeamento de entrada-saída subjacente. Como já foi dito, as unidades ocultas da rede RAN respondem apenas a uma região local do espaço de entrada. A rede aprende alocando novas unidades e ajustando os parâmetros das unidades existentes. Se a rede funcionar mal para um determinado vetor de entrada, conforme medido pelo erro de aproximação do neurônio de saída, uma nova unidade oculta é alocada para corrigir a resposta da rede a esse vetor de entrada. Se a rede funcionar bem para um determinado vetor de entrada, os parâmetros das unidades de saída existentes serão atualizados usando a regra LMS padrão já descrita em seções anteriores.

A RAN começa com uma folha em branco: nenhum padrão foi apresentado ainda. Conforme os padrões são apresentados a ela, a rede escolhe armazenar alguns deles. Em outras palavras, nenhum padrão foi escolhido para ser armazenado ainda. Os pares de entrada-saída são apresentados e a rede identifica um padrão que não está bem representado no momento e aloca uma nova unidade oculta que memoriza essa amostra. A saída da nova unidade oculta se estende à camada de saída. Depois que a nova unidade é alocada, a saída da rede é igual à saída desejada:  $\hat{y}(t) = y(t)$ . Seja o índice desta nova unidade  $n$ . Uma formalização do procedimento de inserção de uma nova unidade oculta é fornecida abaixo.

1. O centro da nova unidade é definido como o  $\mathbf{c}_n = \mathbf{x}(t)$ .
2. O peso que conecta a nova unidade oculta à unidade de saída é definido com  $w_n = y(t) - \hat{y}(t)$ , em que  $\hat{y}(t)$  pode ser calculado conforme a Equação (2.15) do modelo RBF. Vale ressaltar que o modelo RAN se assemelha a um modelo crescente da rede RBF.
3. O raio da nova unidade oculta é dado por

$$\sigma_n(t) = \kappa \|\mathbf{x}(t) - \mathbf{c}_{nearest}(t)\|, \quad (3.11)$$

tal que  $\kappa > 0$  é um fator de sobreposição e  $\mathbf{c}_{nearest}$  é o centro mais próximo de  $\mathbf{c}_n$ .

Para decidir sobre a inserção de uma nova unidade oculta, o modelo RAN verifica duas condições de detecção de novidade.

**Condição 1** - Verificar se o vetor de entrada está muito distante dos centros existentes, o que é verificado pela seguinte regra:

$$\|\mathbf{x}(t) - \mathbf{c}_{nearest}(t)\| > \delta(t), \quad (3.12)$$

em que  $\delta(t)$  é um limiar decrescente para detecção de novidade.

**Condição 2** - Verificar se o erro entre a saída desejada e a saída atual da rede é grande, ou seja,

$$|y(t) - \hat{y}(t)| > \varepsilon, \quad (3.13)$$

em que  $|\cdot|$  é o operador de valor absoluto e  $\varepsilon$  é a acurácia desejada para o problema de aproximação

**Observação 4** - Para estabilizar o processo de crescimento da rede, o limiar de novidade começa com um valor alto  $\delta_{max}$ , diminuindo com o tempo até atingir um valor mínimo  $\delta_{min}$ , que é mantido constante pelo tempo restante do processo de aprendizagem. A este respeito, a função de decaimento sugerida em Platt (1991) é usada para o limiar de novidade:

$$\delta(t) = \max[\delta_{max} \exp(-t/\tau), \delta_{min}], \quad (3.14)$$

em que  $\tau$  é uma constante de decaimento que define a velocidade de inserção de novas unidades ocultas e, portanto, o número de unidades ocultas que podem ser inseridas. Valores pequenos de  $\tau$  levam a um rápido decaimento de  $\delta(t)$ , implicando em um aumento na velocidade de inserção de novas unidades ocultas.

Quando uma nova unidade não é alocada, os centros das unidades ocultas existentes e os pesos e o limiar do neurônio de saída são ajustados por meio de variantes da regra LMS desenvolvidas a seguir. Vale ressaltar que estas equações foram desenvolvidas nesta tese e diferem bastante das equações do modelo RAN original. Por isso, optou-se por colocar esta variante como uma proposta desta tese. Para a aplicação desta regra, a função objetivo global do modelo é o erro quadrático instantâneo do neurônio de saída:

$$J_{LMS}(t) = \frac{1}{2}e^2(t) = \frac{1}{2}(y(t) - \hat{y}(t))^2, \quad (3.15)$$

$$= \frac{1}{2} \left( y(t) - \sum_{j=1}^S z_j(t)w_j(t) \right)^2, \quad (3.16)$$



em que  $w_j(t)$  é o peso associado à  $j$ -ésima unidade oculta e  $z_j(t)$  é a saída da unidade oculta associada e calculada conforme Equação (2.14). Portanto, para atualizar o vetor de peso da  $j$ -ésima função local, a seguinte regra de atualização é usada:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial \mathbf{w}(t)}, \quad (3.17)$$

$$= \mathbf{w}(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \mathbf{w}(t)}, \quad (3.18)$$

$$= \mathbf{w}(t) - \alpha(t) e(t) \cdot (-1) \cdot \mathbf{z}(t), \quad (3.19)$$

$$= \mathbf{w}(t) + \alpha(t) e(t) \mathbf{z}(t), \quad (3.20)$$

em que se pode notar que a atualização dos vetores de peso dos submodelos locais é modulada pelas saídas das unidades ocultas  $z_j(t)$ ,  $j = 1, \dots, S$ . Esta modulação dá um caráter localizado ao processo de atualização dos parâmetros. Quanto maior o valor de  $z_j(t)$ , maior é a variação no vetor de peso  $\mathbf{w}(t)$ .

A atualização dos centros das unidades ocultas segue um procedimento semelhante. Para isto, utilizou-se mais uma vez a função objetivo global do modelo supracitada. Neste caso, aplicou-se a seguinte regra:

$$\mathbf{c}_j(t+1) = \mathbf{c}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial \mathbf{c}_j(t)}, \quad (3.21)$$

$$= \mathbf{c}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial z_j(t)} \frac{\partial z_j(t)}{\partial \mathbf{c}_j(t)}, \quad (3.22)$$

$$= \mathbf{c}_j(t) - \alpha(t) e(t) \cdot (-1) \cdot w_j(t) \cdot \left( \frac{z_j(t)}{\sigma_j^2} (\mathbf{x}(t) - \mathbf{c}_j(t)) \right), \quad (3.23)$$

$$= \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} z_j(t) e(t) w_j(t) (\mathbf{x}(t) - \mathbf{c}_j(t)), \quad (3.24)$$

em que a Equação (3.24) é derivada usando a função de base gaussiana, conforme definido na Equação (2.14). As Equações (3.21), (3.22) e (3.23) foram desenvolvidas para esta tese buscando um melhor entendimento da atualização dos centros das unidades ocultas realizada no trabalho de Platt (1991).

Para os experimentos desenvolvidos nesta tese, obteve-se melhores resultados usando a regra LMS normalizada (WIDROW; LEHR, 1990). Para implementar esta variante da regra LMS, substitui-se a taxa de aprendizagem original  $\alpha(t)$  com  $\alpha'(t) = \alpha(t) / \|\mathbf{x}(t)\|^2$ , onde  $\|\mathbf{x}(t)\|^2$  é a norma quadrática do vetor de entrada atual. O modelo RAN está disposto na forma de pseudocódigo no Algoritmo 2.

---

**Algoritmo 2: Modelo RAN-LMS**


---

**Início:**

Definir:  $\kappa$ ,  $\varepsilon$ ,  $\tau$ ,  $\delta_{max}$ ,  $\delta_{min}$ ,  $\sigma_j$ , e  $\alpha$ .

Atribuir:  $\delta_0 = \delta_{max}$ , e  $S = 1$ .

**Iteração:** apresentação dos pares entrada-saída**1. Predizer:** Para cada vetor de entrada  $\mathbf{x}(t)$ , calcular

$$1.1. z_j(t) = \phi(d(\mathbf{x}(t), \mathbf{c}_j(t))); \quad j = 1, \dots, S.$$

$$1.2. \hat{y}(t) = \sum_{j=1}^S z_j(t)w_j(t);$$

**2. Calcular o erro:**

$$2.1. e(t) = y(t) - \hat{y}(t);$$

**3. Encontrar as distâncias até o centro mais próximo:**

$$3.1. d_{near}(t) = \|\mathbf{x} - \mathbf{c}_{near}(t)\|;$$

**4. Decidir entre crescer ou atualizar:**

**SE**  $|e(t)| > \varepsilon$  e  $d_{near}(t) > \delta(t)$ , **ENTÃO**

{ % Crescer

Aloca nova unidade  $\mathbf{c}_n = \mathbf{x}(t)$  e  $w_n(t) = e(t)$ ;

**SE** essa é a primeira unidade a ser alocada,

**ENTÃO**  $\sigma_n = \kappa\delta(t)$ ;

**SENÃO**  $\sigma_n = \kappa\|\mathbf{x}(t) - \mathbf{c}_{nearest}(t)\|$ .

}

**SENÃO**

{ % Atualizar

Atualiza vetores de peso usando Equação (3.20).

Atualiza centro usando Equação (3.24).

Atualiza limiar de novidade usando Equação (3.14).

}

---

A facilidade de implementação e o baixo custo computacional são dois grandes atrativos do modelo RAN. Pai *et al.* (2001) aplicaram a RAN para estimar o desgaste da ferramenta em operações de fresamento. Sinais de emissão acústica, parâmetros de rugosidade da superfície e condições de corte (velocidade de corte, avanço) foram utilizados para formular padrões de entrada. O desempenho do RAN foi comparado com a rede MLP.

Mahanand *et al.* (2012) apresentaram uma nova abordagem para a identificação de regiões do cérebro responsáveis pela doença de Alzheimer utilizando imagens de ressonância magnética. A abordagem incorporou uma versão da RAN, a SRAN (*self-adaptive resource allocation network*) (SURESH *et al.*, 2010) para a classificação da doença de Alzheimer. O classificador SRAN usa um algoritmo de aprendizagem sequencial, empregando limites auto-

adaptativos para selecionar as amostras de treinamento apropriadas e descartar as amostras redundantes para evitar o excesso de treinamento. Essas amostras de treinamento selecionadas são então utilizadas para desenvolver a arquitetura de rede de forma eficiente.

Jain e Bandyopadhyay (2017) demonstraram a aplicabilidade do modelo RAN por meio de três exemplos de diversos domínios: rede de resfriamento de água, planejamento do setor de energia com restrição de carbono e rede de alocação de água. Em Lv *et al.* (2018), o modelo RAN é treinado *offline* para determinar uma estrutura inicial. A partir daí, o modelo RAN inicial é utilizado para a previsão *online* de energia fotovoltaica e é posteriormente atualizado. Os resultados da simulação mostraram que o modelo RAN de dois estágios pode melhorar efetivamente a precisão da previsão da saída de energia fotovoltaica.

Wang *et al.* (2019) utilizaram o modelo RAN para estimação dos parâmetros de um controlador proporcional integrador derivativo (PID) para veículos guiados automatizados (*automated guided vehicle*, AGV). Os resultados da simulação mostram que o método tem alto desempenho, boa capacidade de rastreamento e alta precisão para aproximação.

Pode-se notar que, apesar da formulação ter se dado no início da década de 90, aplicações recentes da rede RAN ainda são encontradas. Esta rede será usada, com algumas modificações, para o desenvolvimento da primeira proposta desta tese na próxima seção.

### 3.4 Proposta 2: Modelo LMN Crescente (caso SISO)

Nesta seção, uma das propostas desta tese, o modelo *growing* LMN (G-LMN), é apresentada. A abordagem proposta é desenvolvida a partir da variante do modelo RAN (PLATT, 1991) introduzida na seção anterior. A estratégia de alocação de unidades da rede RAN é usada aqui para desenvolver uma variante do modelo LMN que é capaz de crescer com o tempo. O algoritmo proposto é descrito a seguir.

**O Processo de Aprendizagem.** Da mesma forma que a RAN, o modelo G-LMN começa com uma única unidade. As amostras são apresentadas à rede e esta que vai decidir se as amostras estão bem representadas no momento. Da mesma forma da seção anterior, considera-se o índice  $n$  de uma suposta nova unidade que deverá ser alocada caso a amostra apresentada não esteja bem representada pela rede.

1. O centro da nova unidade é definido como o  $\mathbf{c}_n(t) = \mathbf{x}(t)$ .
2. O peso que conecta a nova unidade oculta à unidade de saída é definido com  $w_n =$

$$y(t) - \hat{y}(t).$$

3. O raio da nova unidade oculta é calculado conforme Equação (3.11).

O modelo G-LMN, baseado na RAN, também verifica duas condições de detecção de novidade para decidir sobre a inserção de uma nova unidade oculta conforme as Equações (3.12) e (3.13). A respeito da função de decaimento, utilizou-se a mesma sugerida para o modelo RAN (ver Equação (3.14)). Esse decaimento é necessário para estabilizar o processo de crescimento da rede conforme dito anteriormente. Vale ratificar que os valores de  $\tau$  influenciam diretamente no decaimento de  $\delta(t)$ .

Quando não existe a alocação de uma nova unidade, a rede será ajustada através do ajuste dos pesos das unidades ocultas existentes, dos pesos e do limiar do neurônio de saída por meio de variantes da regra LMS (WIDROW; LEHR, 1990). A função objetivo global do modelo G-LMN difere um pouco da função objetivo global do modelo RAN. Ambas são calculadas pelo erro quadrático instantâneo do neurônio de saída, mas no modelo G-LMN é dada por

$$J_{LMS}(t) = \frac{1}{2}e^2(t) = \frac{1}{2}(y(t) - \hat{y}(t))^2, \quad (3.25)$$

$$= \frac{1}{2} \left( y(t) - \sum_{j=1}^S z_j(t) f_j(\mathbf{x}, \mathbf{w}_j; t) \right)^2, \quad (3.26)$$

em que  $f_j(\mathbf{x}, \mathbf{w}_j; t)$  é a função local associada a  $j$ -ésima função de base  $z_j(t)$ . Desta forma, para atualizar o vetor de pesos da  $j$ -ésima função local, a seguinte regra de atualização é usada:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial \mathbf{w}_j(t)}, \quad (3.27)$$

$$= \mathbf{w}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{y}_j(t)} \frac{\partial \hat{y}_j(t)}{\partial \mathbf{w}_j(t)}, \quad (3.28)$$

$$= \mathbf{w}_j(t) - \alpha(t) e(t) \cdot (-1) \cdot z_j(t) \cdot \mathbf{x}(t), \quad (3.29)$$

$$= \mathbf{w}_j(t) + \alpha(t) z_j(t) e(t) \mathbf{x}(t), \quad (3.30)$$

em que  $\hat{y}_j(t) = f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j(t)^T \mathbf{x}(t)$  e pode notar que a atualização dos vetores de pesos dos submodelos locais é modulada pelas saídas das unidades ocultas  $z_j(t)$ ,  $j = 1, \dots, S$ . Da mesma forma que a variante RAN proposta, esta modulação confere um caráter localizado ao processo de atualização dos parâmetros. Quanto maior o valor de  $z_j(t)$ , maior é a correção no vetor de pesos  $\mathbf{w}_j(t)$ .

A atualização dos centros das unidades ocultas segue um procedimento semelhante ao modelo de Platt (1991):

Tabela 1 – Regras de aprendizado das duas propostas: variante da RAN e G-LMN.

Modelos propostos	Regras de aprendizado
RAN-LMS	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)e(t)\mathbf{z}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2}z_j(t)e(t)w_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t))$
G-LMN	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)z_j(t)e(t)\mathbf{x}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2}z_j(t)e(t)f_j(\mathbf{x}, \mathbf{w}_j; t)(\mathbf{x}(t) - \mathbf{c}_j(t))$

$$\mathbf{c}_j(t+1) = \mathbf{c}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial \mathbf{c}_j(t)}, \quad (3.31)$$

$$= \mathbf{c}_j(t) - \alpha(t) \frac{\partial J_{LMS}(t)}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}_j(t)} \frac{\partial \hat{y}_j(t)}{\partial z_j(t)} \frac{\partial z_j(t)}{\partial \mathbf{c}_j(t)}, \quad (3.32)$$

$$= \mathbf{c}_j(t) - \alpha(t)e(t) \cdot (-1) \cdot \hat{y}_j(t) \cdot \left( \frac{z_j(t)}{\sigma_j^2} (\mathbf{x}(t) - \mathbf{c}_j(t)) \right), \quad (3.33)$$

$$= \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} z_j(t)e(t)\hat{y}_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t)), \quad (3.34)$$

em que  $\hat{y}_j(t) = f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j(t)^T \mathbf{x}(t)$  e a Equação (3.34) é derivada usando a função de base gaussiana, conforme definido na Equação (2.14).

As regras de aprendizado das duas propostas, RAN-LMS e G-LMN, são apresentadas na Tabela 1. Quanto à atualização dos pesos  $\mathbf{w}_j$ , notou-se que o vetor de entrada  $\mathbf{x}(t)$  possui uma influência maior no modelo G-LMN. Quanto à atualização dos centros  $\mathbf{c}_j$ , notou-se a que no modelo RAN, o vetor de pesos  $\mathbf{w}_j$  tinha uma influência direta na atualização dos centros. Já no modelo G-LMN, a atualização dos centros  $\mathbf{c}_j$  recebe uma influência do valor de  $f_j(\mathbf{x}, \mathbf{w}_j; t)$ , ou seja, mais uma vez, o vetor de entrada  $\mathbf{x}$  influencia mais fortemente na adaptação.

Aqui também utilizou-se a regra LMS normalizada (WIDROW; LEHR, 1990), ou seja, substituiu-se a taxa de aprendizagem original  $\alpha(t)$  por  $\alpha'(t) = \alpha(t)/\|\mathbf{x}(t)\|^2$ , onde  $\|\mathbf{x}(t)\|^2$  é a norma quadrática do vetor de entrada atual. O pseudocódigo do modelo G-LMN é apresentado no Algoritmo 3.

**Observação 5** - A escolha pela regra LMS normalizada para atualizar os parâmetros dos modelos RAN e G-LMN se baseia em dois aspectos importantes.

1. Em primeiro lugar, é uma regra de baixo custo computacional para estimativa de parâmetros em comparação com a regra RLS, que também é uma escolha comum para atualização de parâmetro recursiva (Mandic *et al.*, 2015), uma vez que não há necessidade de estimar  $S$  matrizes de covariância inversa, um para cada unidade oculta, que são necessárias para atualizar os vetores de ganhos de Kalman associados.

---

**Algoritmo 3: Modelo G-LMN**


---

**Início:**

Definir:  $\kappa$ ,  $\varepsilon$ ,  $\tau$ ,  $\delta_{max}$ ,  $\delta_{min}$ ,  $\sigma_j$ , e  $\alpha$ .

Atribuir:  $\delta_0 = \delta_{max}$ , e  $S = 1$ .

**Iteração:** apresentação dos pares entrada-saída**1. Predizer:** Para cada vetor de entrada  $\mathbf{x}(t)$ , calcular

$$1.1. z_j(t) = \phi(d(\mathbf{x}(t), \mathbf{c}_j(t))); \quad j = 1, \dots, S.$$

$$1.2. f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j(t)^T \mathbf{x}(t) + w_{j,0}(t); \quad j = 1, \dots, S.$$

$$1.3. \hat{y}(t) = \sum_{j=1}^S z_j(t) f_j(\mathbf{x}, \mathbf{w}_j; t);$$

**2. Calcular o erro:**

$$2.1. e(t) = y(t) - \hat{y}(t);$$

**3. Encontrar as distâncias até o centro mais próximo:**

$$3.1. d_{near}(t) = \|\mathbf{x} - \mathbf{c}_{near}(t)\|;$$

**4. Decidir entre crescer ou atualizar:**

**SE**  $|e(t)| > \varepsilon$  e  $d_{near}(t) > \delta(t)$ , **ENTÃO**

{ % Crescer

Aloca nova unidade  $\mathbf{c}_n = \mathbf{x}(t)$  e  $w_n = e(t)$ ;

**SE** essa é a primeira unidade a ser alocada,

**ENTÃO**  $\sigma_n = \kappa \delta(t)$ ;

**SENÃO**  $\sigma_n = \kappa \|\mathbf{x}(t) - \mathbf{c}_{nearest}(t)\|(t)$ .

}

**SENÃO**

{ % Atualizar

Atualiza vetores de peso usando Equação (3.30).

Atualiza centro usando Equação (3.34).

Atualiza limiar de novidade usando Equação (3.14).

}

---

2. Em segundo lugar, uma regra de aprendizado robusta para *outliers* é obtida diretamente da regra LMS original sem nenhum custo computacional extra, como será mostrado no próximo capítulo.

Vale destacar aqui que propostas incrementais do modelo LMN foram abordadas em Teshnehlab *et al.* (2019) e Blazic e Skrjanc (2020). Em Teshnehlab *et al.* (2019), um novo algoritmo incremental foi proposto baseado no algoritmo *local linear model trees* (LOLIMOT) (NELLES, 1996). Os autores chamaram esse modelo de *growing local linear model trees* (GLOLIMOT). Este modelo e sua versão original são testados em um protótipo de turbina a gás industrial de eixo único, que possui alta complexidade e grande dimensão.

Já no trabalho proposto por Blazic e Skrjanc (2020), o algoritmo adiciona os modelos locais de forma incremental e adapta recursivamente os parâmetros do modelo local. A estrutura do modelo é tratada como um o modelo LMN na forma Takagi-Sugeno e o particionamento do espaço de entrada-saída é baseado em métricas nas quais esses modelos locais são definidos como protótipos dos agrupamentos. O algoritmo proposto é testado em três exemplos para demonstrar suas principais características. O primeiro é um exemplo em que há sobreposição de agrupamentos; o segundo é um *benchmark* (a série temporal Mackey-Glass); o terceiro é um exemplo que mostra a classificação dos dados de um telêmetro a laser.

Após buscas exaustivas, as poucas abordagens incrementais do modelo LMN encontrados na literatura deixam uma lacuna para a aplicação de modelos incrementais em identificação recursiva de sistemas dinâmicos. A teoria do modelo RAN para o crescimento de um modelo LMN não foi encontrado em trabalhos anteriores. Dessa forma, pode-se ainda destacar que o modelo G-LMN proposto possui as seguintes características que foram citadas no início desse capítulo:

1. estrutura *online* crescente para lidar com a não estacionariedade de processos;
2. regra de atualização recursiva rápida para rastreamento rápido de mudanças na dinâmica do sistema;
3. melhor uso de memória, uma vez que nenhum armazenamento de matrizes de covariância é necessário, conforme exigido nas regras de aprendizagem do tipo RLS.

### 3.5 Proposta 3: O caso MIMO

Nesta tese, tem-se um particular interesse em identificação de sistemas dinâmicos MIMO; portanto, é necessário desenvolver uma formulação para estes casos. Para identificação de sistemas MIMO, o vetor de regressão deve conter informação sobre todas as variáveis de entrada e saída de interesse (LJUNG, 1999). A este respeito, definiu-se o subvetor de regressão de entrada  $\mathbf{u}_a(t-1)$  associado à  $a$ -ésima entrada como

$$\mathbf{u}_a(t-1) = [u_a(t-1) \ u_a(t-2) \ \cdots \ u_a(t-L_u)]^T, \quad a = 1, 2, \dots, n_u, \quad (3.35)$$

em que  $L_u$  denota a ordem de memória das variáveis de entrada, mantida igual para todas as variáveis. Da mesma forma, definiu-se o subvetor de regressão de saída  $\mathbf{y}_b(t-1)$  associado a  $b$ -ésima saída como

$$\mathbf{y}_b(t-1) = [y_b(t-1) \ y_b(t-2) \ \cdots \ y_b(t-L_y)]^T, \quad b = 1, 2, \dots, n_y, \quad (3.36)$$

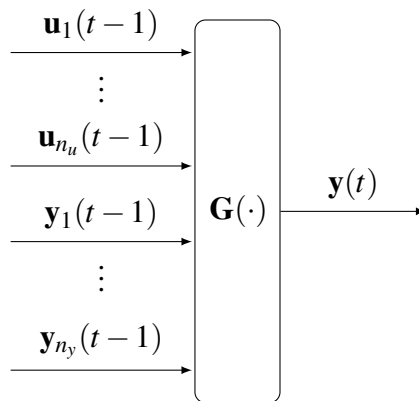


Figura 6 – Esquema de um modelo NARX MIMO, em que  $\mathbf{G}(\cdot)$  é uma função não linear genérica.

em que  $L_y$  denota a ordem de memória das variáveis de saída, mantida a mesma para todas as variáveis. Isto posto, o vetor de regressores para o caso MIMO é então dado por (BILLINGS, 2013)

$$\mathbf{x}(t) = [\mathbf{y}_1(t-1) \ \mathbf{y}_2(t-1) \ \cdots \ \mathbf{y}_{n_y}(t-1), \ \mathbf{u}_1(t-1) \ \mathbf{u}_2(t-1) \ \cdots \ \mathbf{u}_{n_u}(t-1)] \quad (3.37)$$

em que  $\mathbf{x}(t) \in \mathbb{R}^{(n_y \cdot L_y + n_y \cdot L_u)}$  e  $\mathbf{y}(t) = \mathbf{G}(\mathbf{x}(t))$  conforme ilustrado na Figura 6.

A função não linear  $\mathbf{G}(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$  é considerada desconhecida e  $n_x = n_y \cdot L_y + n_u \cdot L_u$ . Os dados observados na forma de séries temporais de entrada-saída são usados para construir um modelo aproximado  $\hat{\mathbf{G}}(\cdot)$  para a função de alvo  $\mathbf{G}(\cdot)$  da mesma forma dos sistemas SISO.

Assim, pode-se realizar uma generalização dos modelos descritos nessa tese para um sistema MIMO. Quanto ao modelo LLM, o vetor de coeficientes descrito pela Equação (2.5) é reescrito como uma matriz de coeficientes

$$\mathbf{A}_j(t) = \begin{pmatrix} a_{j,(1,1)} & \cdots & a_{j,(1,n_y)} \\ & \vdots & \\ a_{j,(n_x,1)} & \cdots & a_{j,(n_x,n_y)} \end{pmatrix}_{n_x \times n_y}, \quad (3.38)$$

de modo que o valor predito da saída do modelo baseado em LLM descrito pela Equação (2.7) é então reescrito como

$$\begin{pmatrix} \hat{y}_1(t) \\ \vdots \\ \hat{y}_{n_y}(t) \end{pmatrix}_{n_y \times 1} = \begin{pmatrix} a_{j^*,(1,1)} & \cdots & a_{j^*,(1,n_y)} \\ & \vdots & \\ a_{j^*,(n_x,1)} & \cdots & a_{j^*,(n_x,n_y)} \end{pmatrix}_{n_y \times n_x}^T \begin{pmatrix} x_1(t) \\ \vdots \\ x_{n_x}(t) \end{pmatrix}_{n_x \times 1}. \quad (3.39)$$



Quanto ao modelo INC-LLM, o valor predito descrito pela Equação (3.5) é generalizado para

$$\begin{pmatrix} \hat{y}_1(t) \\ \vdots \\ \hat{y}_{n_y}(t) \end{pmatrix} = \begin{pmatrix} \hat{y}_{s_1,1}(t) \\ \vdots \\ \hat{y}_{s_1,n_y}(t) \end{pmatrix} + \begin{pmatrix} a_{s_1,(1,1)} & \cdots & a_{s_1,(1,n_y)} \\ & \vdots & \\ a_{s_1,(n_x,1)} & \cdots & a_{s_1,(n_x,n_y)} \end{pmatrix}_{n_y \times n_x}^T \left[ \begin{pmatrix} x_1(t) \\ \vdots \\ x_{n_x}(t) \end{pmatrix} - \begin{pmatrix} w_{s_1,1}(t) \\ \vdots \\ w_{s_1,n_x}(t) \end{pmatrix} \right]_{n_x \times 1}, \quad (3.40)$$

em que  $s_1$  é o centro mais próximo determinado conforme a Equação (2.6).

Como o modelo RAN é muito semelhante ao modelo G-LMN, será descrito com mais detalhes o modelo G-LMN para o problema MIMO. O modelo G-LMN é usado para aprender a dinâmica de um determinado sistema de entrada-saída usando os vetores de entrada conforme definido na Equação (3.37).

Como o interesse é identificar um sistema MIMO, considerou-se uma camada de saída com  $n_y$  neurônios. Assim, a saída do modelo G-LMN associa uma função vetorial local  $\mathbf{f}_j(\mathbf{x}; \mathbf{w}_j)$  para cada função de base, a saber,

$$\hat{\mathbf{y}}(t) = \sum_{j=1}^S z_j(t) \hat{\mathbf{y}}_j(t) = \sum_{j=1}^S z_j(t) \mathbf{f}_j(\mathbf{x}, \mathbf{w}_j; t), \quad (3.41)$$

em que  $\hat{\mathbf{y}}_j(t) \in \mathbb{R}^{n_y}$  é a predição multidimensional local associada a  $\mathbf{f}_j \in \mathbb{R}^{n_y}$ , cujas componentes são dadas por

$$\begin{aligned} \hat{y}_1(t) &= \sum_{j=1}^S z_j(t) f_{j,1}(\mathbf{x}, \mathbf{w}_j; t), \\ \hat{y}_2(t) &= \sum_{j=1}^S z_j(t) f_{j,2}(\mathbf{x}, \mathbf{w}_j; t), \\ &\vdots \\ \hat{y}_{n_y}(t) &= \sum_{j=1}^S z_j(t) f_{j,n_y}(\mathbf{x}, \mathbf{w}_j; t). \end{aligned} \quad (3.42)$$

Para este caso, o processo de aprendizagem segue os mesmos passos do caso SISO. A formalização da inserção de uma nova unidade é descrita a seguir:

1. O centro da nova unidade é dado por  $\mathbf{c}_n = \mathbf{x}(t)$ .

2. O peso  $\mathbf{w}_n$  que conecta a nova unidade à camada de saída é dado por uma matriz com  $n_x$  linhas com cada linha sendo igual a  $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ .
3. O raio da nova unidade é dado pela Equação (3.11).

Para o caso MIMO, os critérios de novidade para inserção de uma nova unidade são formalizados através da Equação (3.12) e a substituição da Equação (3.13) por

$$\|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\| > \varepsilon, \quad (3.43)$$

em que  $\|\cdot\|$  denota a norma euclidiana.

Quando uma nova unidade não é alocada, os centros, os pesos e os limiares devem ser ajustados através da regra LMS ou da regra LMM. Para aplicação dessa regra no caso MIMO, a função objetivo global do modelo, descrita na Equação (3.15) para o modelo SISO, é substituída por

$$J_{MIMO}(t) = \frac{1}{2} \|\mathbf{e}(t)\|^2 = \frac{1}{2} \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|^2, \quad (3.44)$$

$$= \frac{1}{2} \|\mathbf{y}(t) - \sum_{j=1}^S z_j(t) \hat{\mathbf{y}}_j(t)\|^2, \quad (3.45)$$

onde  $\hat{\mathbf{y}}_j(t) \in \mathbb{R}^{n_y}$  é a saída do  $j$ -ésimo modelo local e  $z_j(t)$  é a saída da unidade oculta associada.

Portanto, para atualizar o vetor de peso da  $j$ -ésima função local, a regra de atualização da Equação (3.30) é modificada para

$$\begin{pmatrix} \mathbf{w}_{j,1}(t+1) \\ \vdots \\ \mathbf{w}_{j,n_y}(t+1) \end{pmatrix}_{n_y \times n_x} = \begin{pmatrix} \mathbf{w}_{j,1}(t) \\ \vdots \\ \mathbf{w}_{j,n_y}(t) \end{pmatrix}_{n_y \times n_x} - \alpha(t) z_j(t) \begin{pmatrix} e_1(t) \\ \vdots \\ e_{n_y}(t) \end{pmatrix}_{n_y \times 1} \begin{pmatrix} x_1(t) \\ \vdots \\ x_{n_x}(t) \end{pmatrix}_{1 \times n_x}^T. \quad (3.46)$$

A atualização dos centros das unidades ocultas prevista na Equação (3.34) recebe modificações semelhantes e é dada por

$$\begin{aligned}
\begin{pmatrix} c_{j,1}(t+1) \\ \vdots \\ c_{j,n_x}(t+1) \end{pmatrix}_{n_x \times 1} &= \begin{pmatrix} c_{j,1}(t) \\ \vdots \\ c_{j,n_x}(t) \end{pmatrix}_{n_x \times 1} \\
&+ \frac{\alpha(t)}{\sigma_j^2} z_j(t) \begin{pmatrix} \hat{y}_1(t) \\ \vdots \\ \hat{y}_{n_y}(t) \end{pmatrix}_{1 \times n_y}^T \begin{pmatrix} e_1(t) \\ \vdots \\ e_{n_y}(t) \end{pmatrix}_{n_y \times 1} \left( \begin{pmatrix} x_1(t) \\ \vdots \\ x_{n_x}(t) \end{pmatrix} - \begin{pmatrix} c_{j,1}(t) \\ \vdots \\ c_{j,n_x}(t) \end{pmatrix} \right)_{n_x \times 1}
\end{aligned} \tag{3.47}$$

Uma vez que as soluções para o caso MIMO foram encontradas, os erros de predição devido a cada saída são calculados por

$$\begin{aligned}
e_1(t) &= y_1(t) - \sum_{j=1}^S z_j(t) f_{j,1}(\mathbf{x}, \mathbf{w}_j; t), \\
e_2(t) &= y_2(t) - \sum_{j=1}^S z_j(t) f_{j,2}(\mathbf{x}, \mathbf{w}_j; t), \\
&\vdots \\
e_{n_y}(t) &= y_{n_y}(t) - \sum_{j=1}^S z_j(t) f_{j,n_y}(\mathbf{x}, \mathbf{w}_j; t).
\end{aligned} \tag{3.48}$$

### 3.6 Considerações finais

Neste capítulo, um modelo de estrutura crescente bem conhecido na literatura foi descrito, a saber, o modelo INC-LLM. Em seguida, duas propostas de modelos crescentes para identificação recursiva de sistemas desenvolvidas nesta tese são apresentadas, ambas construídas a partir do critério de novidade da rede RAN. Assim, a limitação dos modelos locais de ter que especificar a quantidade de unidades a priori foi resolvida. Nas abordagens descritas neste capítulo, os modelos locais são inseridos de maneira progressiva até que um requisito escolhido pelo usuário seja alcançado. Isso é importante pois a dinâmica do sistema pode mudar ao longo do tempo e o modelo pode precisar adaptar sua estrutura para lidar com as novas demandas da tarefa. Lançou-se também a proposta de modelo crescente (G-LMN) desta tese que é um modelo crescente baseado em duas abordagens bastante conhecidas na literatura: LMN e RAN e, assim, conseguiu-se integrar a função de crescimento ao modelo LMN para a tarefa de identificação de

sistemas dinâmicos. No Capítulo 4, a regra LMM será descrita e discutida e como ela poderá ser utilizada para se incluir robustez a *outliers* aos modelos propostos neste capítulo.

## 4 MODELOS LOCAIS ROBUSTOS A *OUTLIERS*

Neste capítulo, uma discussão sobre a importância do tratamento de *outliers* na tarefa de identificação de sistemas será realizada e a regra LMM será apresentada. Com este arcabouço teórico, mais uma proposta deste trabalho será apresentada, a saber, ORG-LMN. Os modelos apresentados anteriormente também terão suas regras de atualização alteradas via regra LMM.

### 4.1 Introdução

O adjetivo robusto, que deriva do latim *robustus*, faz referência àquilo que é vigoroso, forte ou resiliente. A robustez pode referir-se a características físicas ou simbólicas. Uma construção que se mantém em pé por muitos séculos demonstra a sua robustez por ter resistido ao passar do tempo. Um sistema de *software*, por outro lado, pode vangloriar-se da sua robustez se conseguir executar diversos processos em simultâneo sem causar problemas ou bloquear. No problema de identificação de sistemas, robustez está intimamente relacionada ao modelo estimado ser capaz de lidar com medidas anômalas que fazem com que o modelo preditivo gere um resíduo de alto valor.

Como dito anteriormente, as amostras de dados coletadas em aplicações no mundo real estão comumente contaminadas com *outliers*. Estes, geralmente, são entendidos como observações que diferem claramente de outras observações dentro do mesmo conjunto de dados. Segundo Hampel *et al.* (2011), algumas das principais fontes de *outliers* são as seguintes:

- erros de medição,
- erros humanos,
- erros de arredondamento,
- falhas de sistema (comportamento inesperado) e
- mudanças no *set-point* no sistema.

Yu *et al.* (2015) afirmam que essas observações parecem ser inconsistentes com todo o restante do conjunto de dados original. Não existe uma definição matemática formal do que constitui um *outlier*, e determinar se uma observação é ou não um desses pontos é, em última análise, um exercício um tanto quanto subjetivo.

Grosso modo, conforme Barros (2013), um *outlier* é uma observação (escalar ou vetorial) que difere marcadamente de outras observações da amostra a que pertence, ou seja, *outliers* são pontos fora da curva pelo simples fato de não partilharem a mesma distribuição

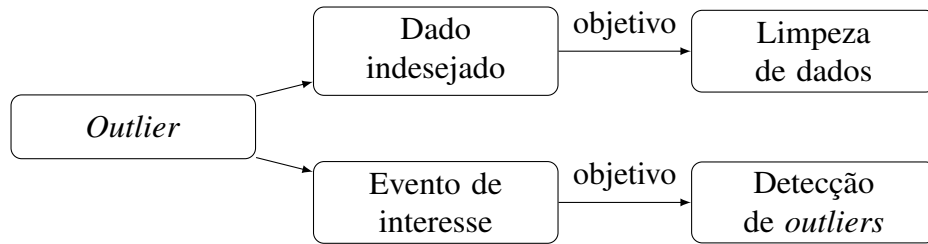


Figura 7 – Abordagens para tratamento de *outliers*. Adaptado de Blazquez-García *et al.* (2021)

de probabilidades que o restante das observações. Embora não haja tal definição universal do que seja um *outlier*, a conceituação acima é adequada para problemas de regressão robusta e, portanto, é adotada a partir de agora.

É frequentemente assumido que os *outliers* seguem uma distribuição de ruído diferente da gaussiana (por exemplo, *t*-Student). Neste sentido, as regras de estimativa padrão baseadas em erros quadráticos, como o OLS e o LMS, levam a modelos com desempenho abaixo do ideal, uma vez que essas regras são desenvolvidas sob a suposição de gaussianidade do ruído. Essa suposição é adequada para o ruído de medição usual, mas não para *outliers* ou ruído impulsivo. Segundo Blazquez-García *et al.* (2021), esse tipo de ruído é um dado que se comporta de maneira incomum em um determinado instante de tempo quando comparados com os outros valores na série temporal ou com seus pontos vizinhos.

A má influência dos *outliers* é ainda mais pronunciada na estimativa recursiva *online*, onde um processo de remoção de *outliers* é mais difícil ou muitas vezes impossível de se realizar em comparação a uma abordagem *offline*. Isso é importante de ser ressaltado porque a busca é por modelos que sejam capazes de aprender diretamente a dinâmica do sistema de interesse mesmo com a presença de *outliers* nos dados, ao contrário de metodologias que visam removê-los dos dados antes do treinamento, ou seja, uma etapa conduzida *offline* visando a limpeza dos dados (PEARSON, 2002; MATTOS, 2017). As formas para tratamento de *outliers* estão dispostas na Figura 7. A detecção de *outliers* se tornou um campo de interesse para muitos pesquisadores e profissionais e é agora uma das principais tarefas dentro da mineração de dados em identificação de sistemas (BLAZQUEZ-GARCÍA *et al.*, 2021).

A robustez que se busca está relacionada a modelos cujo o processo de estimação de parâmetros não seja afetado pela presença de *outliers* nos dados de treinamento. Esta robustez é uma propriedade desejada para métodos de identificação do sistema. Huber (1964) é um dos pioneiros no desenvolvimento e na implantação da área de estatística robusta. Uma estatística robusta é aquela não afetada por medidas discrepantes. O objetivo é produzir estimadores que não sejam afetados por observações que não satisfazem às hipóteses que levaram à construção

do modelo matemático de interesse (PIRES *et al.*, 2007).

Huber (1964) também se posiciona contra a abordagem de limpeza de dados, enfatizando que o fato de lidarmos diretamente com a presença de *outliers* evita uma remoção incorreta de amostras de treinamento que poderiam ser extremamente valiosas.

A partir do exposto, a necessidade de minimizar o impacto que esses valores atípicos provocam nas estimativas de parâmetros, acarretou no surgimento de métodos mais robustos, ou seja, menos sensíveis; para o nosso caso em particular, a abordagem de identificação de sistemas robusta busca desenvolver estimadores que são resilientes a *outliers*. Para isto, Huber (1964) desenvolveu uma classe de estimadores conhecida como estimadores  $M$ , com a letra  $M$  servindo para simbolizar que os estimadores robustos são do tipo *máxima verossimilhança*.

Com base nesse arcabouço teórico, os estimadores recursivos robustos são encontrados em Zou *et al.* (2000) e Chan e Zhou (2010), que introduziram modificações na regra LMS padrão a fim de melhorar seu desempenho em dados contaminados com *outliers*. A regra resultante foi então chamada de LMM (*least mean M-estimate*) e é tão simples quanto a regra LMS original, no sentido de que nenhuma carga computacional extra é adicionada ao processo de estimativa de parâmetros. Na próxima seção, a regra LMM será descrita com mais detalhes.

## 4.2 Estimação Recursiva Robusta via Regra LMM

Muitas técnicas têm sido empregadas para reduzir o efeito hostil no sistema devido à interferência de *outliers*. Os exemplos incluem algoritmos baseados no filtro LMS da mediana (HAWHEEL; CLARKSON, 1992) e algoritmos baseados em estimação  $M$ , como o já mencionado algoritmo LMM (ZOU *et al.*, 2000) e *recursive least M-estimate* (RLM) (CHAN; ZOU, 2004). Em particular, os dois últimos algoritmos foram desenvolvidos minimizando outras funções objetivo em vez do erro quadrático médio (*mean square error*, MSE) convencional. A seguir, o conceito da função objetivo da estimativa  $M$  será brevemente revisada e o algoritmo LMM apresentado.

Considera-se o problema de identificação do sistema mostrado na Figura 2. Os sinais  $\mathbf{x}(t)$  e  $\hat{y}(t)$  são, respectivamente, os sinais de entrada e da saída predita do modelo. O erro de estimação é dado por  $e(t) = y(t) - \mathbf{w}^T(t)\mathbf{x}(t)$ , tal que  $\mathbf{w}(t)$  e  $\mathbf{x}(t)$  são os vetores de coeficiente e de entrada, respectivamente. O sinal  $y(t)$  é a saída medida no instante  $t$ . Em aplicações práticas,  $\mathbf{x}(t)$  e  $y(t)$  podem ser corrompidos por *outliers*. Baseado em estatística robusta para estimação

(HUBER, 1964), a seguinte função objetivo para modelo é proposta (ZOU *et al.*, 2000)

$$J_p = E[\rho(e(t))], \quad (4.1)$$

em que  $\rho(\cdot)$  é uma função para mitigar o efeito de (*outliers*). Diversas são as funções que podem ser escolhidas para este fim (SANTOS, 2017), tais como OLS, Hampel, Huber, Logistic, Myriad. Porém, alguns requisitos são impostos na escolha da função  $\rho(\cdot)$ ; ou seja,

1.  $\rho(e) \geq 0$ ;
2.  $\rho(0) = 0$ ;
3.  $\rho(e) = \rho(-e)$ ;
4.  $\rho(e(t_1)) \geq \rho(e(t_2))$ , se  $|e(t_1)| > |e(t_2)|$ .

Para a tarefa de identificação de sistemas, duas funções se destacam, a saber, função Hampel (ZOU *et al.*, 2000) e Huber modificada (HM) (CHAN; ZHOU, 2010). Nesta tese, utilizou-se a função Huber Modificada descrita como

$$\rho(e(t)) = \begin{cases} e(t)^2/2, & \text{se } |e(t)| < \beta, \\ \beta^2/2, & \text{caso contrário,} \end{cases} \quad (4.2)$$

em que  $\beta$  é o limiar de anormalidade, que define o intervalo em que o resíduo é considerado dentro ou fora dos padrões usuais. Da mesma forma do algoritmo LMS, a função objetivo  $J_p$  será minimizada atualizando  $\mathbf{w}(t)$  na direção negativa do vetor gradiente. Portanto, o vetor gradiente  $\nabla_{\mathbf{w}} J_p$  a saber,

$$\nabla_{\mathbf{w}} J_p(e(t)) = E \left[ + \frac{\partial \rho(e(t))}{\partial \mathbf{w}(t)} \right] \quad (4.3)$$

que pode ser aproximado pelo gradiente instatâneo  $\nabla J_p(e(t))$  da seguinte forma

$$\hat{\nabla}_{\mathbf{w}} = - \frac{\partial \rho(e(t))}{\partial \mathbf{w}(t)} = + \frac{\partial \rho(e(t))}{\partial e(t)} \cdot \frac{\partial e(t)}{\partial \mathbf{w}(t)}, \quad (4.4)$$

$$= + \frac{\partial \rho(e(t))}{\partial e(t)} \cdot \frac{\partial e(t)}{\partial \hat{y}(t)} \cdot \frac{\partial \hat{y}(t)}{\partial \mathbf{w}(t)}, \quad (4.5)$$

$$= \psi(e(t))(-1)\mathbf{x}(t), \quad (4.6)$$

$$= -q(e(t))e(t)\mathbf{x}(t), \quad (4.7)$$

em que  $q(e) = \frac{1}{e} \psi(e)$  e  $\psi(e)$  é a função de ponderação do erro. Para a função Huber modificada, tem-se



$$q(e(t)) = \begin{cases} 1, & \text{se } |e(t)| < \beta, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.8)$$

Finalmente, o algoritmo LMM proposto em Chan e Zhou (2010) é dado por:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \hat{\mathbf{V}}_{w\rho} \quad (4.9)$$

$$= \mathbf{w}(t) + \alpha q(e(t))e(t)\mathbf{x}(t). \quad (4.10)$$

Geralmente, quando  $|e(t)| < \beta$ , a função de ponderação  $q(e(t))$  é igual a 1 e, assim, a Equação (4.10) se torna idêntica ao algoritmo LMS. Porém, quando  $|e(t)| > \beta$ ,  $q(e(t))$  se tornará zero e impedirá a atualização do vetor de pesos. Recomenda-se usar  $\beta = 1,345\sigma$  a fim de produzir 95% de eficiência quando os erros têm distribuição gaussiana, enquanto ainda oferece proteção contra *outliers*, com  $\sigma$  sendo uma estimativa robusta da dispersão dos resíduos. (ROUSSEEUW; LEROY, 2005).

Assim, o algoritmo LMM pode reduzir efetivamente o efeito adverso de grandes erros de estimação na atualização dos vetores de coeficientes. Na próxima seção, utilizando a regra LMM apresentada, a terceira proposta desta tese será apresentada.

### 4.3 Proposta 4: *Outlier-robust growing LMN*

A fim de aplicar o princípio subjacente da regra LMM às regras de atualização que são baseadas na regra LMS, toma-se como base o modelo G-LMN proposto no capítulo anterior para desenvolver uma variante robusta a *outliers*, chamada de modelo ORG-LMN (*outlier-robust G-LMN*). Pode-se redefinir a função objetivo da Equação (3.15) para uma expressão mais geral dada por

$$\begin{aligned} J_{LMM}(t) &= \rho(e(t)) \\ &= \rho(y(t) - \hat{y}(t)) \\ &= \rho\left(y(t) - \sum_{j=1}^S z_j(t)\hat{y}_j(t)\right), \end{aligned} \quad (4.11)$$

em que função  $\rho(\cdot)$  calcula a contribuição do erro de estimação atual  $e(t)$  para a função objetivo. Se  $\rho(e(t)) = \frac{1}{2}e^2(t)$ , obtém-se a função custo do erro quadrático instantâneo.

Isto posto, uma versão robusta da regra recursiva para atualização do vetor de coeficientes da  $j$ -ésima função local mostrada na Equação (3.30), é dada por

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) - \alpha(t) \frac{\partial J_{LMM}(t)}{\partial \mathbf{w}_j(t)}, \quad (4.12)$$

$$= \mathbf{w}_j(t) - \alpha(t) \frac{\partial \rho(e(t))}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial \hat{y}_j(t)} \frac{\partial \hat{y}_j(t)}{\partial \mathbf{w}_j(t)}, \quad (4.13)$$

$$= \mathbf{w}_j(t) - \alpha(t) q(e(t)) e(t) \cdot (-1) \cdot z_j(t) \cdot \mathbf{x}(t), \quad (4.14)$$

$$= \mathbf{w}_j(t) + \alpha(t) z_j(t) q(e(t)) e(t) \mathbf{x}(t), \quad (4.15)$$

em que  $q(e(t))$  é uma função com o papel de penalizar altos valores de resíduos  $e(t)$  que são produzidos devido a *outliers* e  $\hat{y}_j(t) = f_j(\mathbf{x}, \mathbf{w}_j; t)$ . É importante destacar que a única diferença entre as Equações (4.15) e (3.30) é o fator de ponderação  $q(e(t))$ .

Conforme descrito na Seção 4.2, a fim de aplicar os estimadores  $M$  em problemas de regressão, muitas funções de peso tem sido propostas na literatura, especialmente para regressão linear (ROUSSEEUW; LEROY, 2005). Nesta tese, o foco está na simplicidade e no baixo custo computacional no cálculo das estimativas, a escolha para  $q(e(t))$  recai sobre a função Huber, mostrada na Equação (4.8). Para esta função, o fator de ponderação é dado por

$$q(e(t)) = \begin{cases} \frac{\beta}{|e(t)|}, & \text{se } |e(t)| > \beta, \\ 1, & \text{caso contrário.} \end{cases} \quad (4.16)$$

em que a constante  $\beta > 0$  é um limiar de anormalidade, a partir do qual a magnitude do erro é considerada alta demais para ser produzida por uma amostra comum.

Ratificando que se  $|e(t)| \leq \beta$ , então obtém-se  $q(e(t)) = 1$ , com a Equação (4.15) reduzindo à sua versão original mostrada na Equação (3.30). Se  $|e(t)| > \beta$ ,  $q(e(t))$  diminui exponencialmente para zero como  $|e(t)| \rightarrow \infty$ . Dessa forma, a regra de atualização do LMM reduz efetivamente o efeito de grandes erros, geralmente causados por *outliers*.

A fim de tornar o modelo proposto totalmente adaptativo, pode-se derivar uma versão robusta para atualizar o centro da  $j$ -ésima unidade oculta

---

**Algoritmo 4: Modelo ORG-LMN**


---

**Início**

Definir:  $p, q, \kappa, \varepsilon, \tau, \delta_{max}, \delta_{min}, \sigma_j, \alpha$ , e  $\beta$ .

Atribuir:  $\delta_0 = \delta_{max}$ , e  $S = 1$ .

**Iteração:** apresentação dos pares entrada-saída

**1. Predizer:** Para cada vetor de entrada  $\mathbf{x}(t)$ , calcular

$$1.1. z_j(t) = \phi(d(\mathbf{x}(t), \mathbf{c}_j(t))); \quad j = 1, \dots, S.$$

$$1.2. f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j(t)^T \mathbf{x}(t) + w_{j,0}(t); \quad j = 1, \dots, S.$$

$$1.3. \hat{y}(t) = \sum_{j=1}^S z_j(t) f_j(\mathbf{x}, \mathbf{w}_j; t);$$

**2. Calcular o erro:**

$$2.1. e(t) = y(t) - \hat{y}(t);$$

**3. Calcular o fator de ponderação a partir da função Huber:**

**SE**  $|e| > \beta$ ,

**ENTÃO**  $q(e) = \beta / |e|$ ;

**SENÃO**  $q(e) = 1$ .

**4. Encontrar as distâncias até o centro mais próximo::**

$$4.1. d_{near}(t) = \|\mathbf{x} - \mathbf{c}_{near}(t)\|;$$

**5. Decidir entre crescer ou atualizar:**

**SE**  $|e(t)| > \varepsilon$  and  $d_{near}(t) > \delta(t)$ , **ENTÃO**

{ % Crescer

Aloca nova unidade  $\mathbf{c}_n = \mathbf{x}(t)$  and  $w_n = e(t)$ ;

**SE** essa é a primeira unidade a ser alocada,

**ENTÃO**  $\sigma_n = \kappa \delta(t)$ ;

**SENÃO**  $\sigma_n = \kappa d_{near}(t)$ .

}

**SENÃO**

{ % Atualizar

Atualiza vetores de peso usando Equação (4.15).

Atualiza centro usando Equação (4.20).

Atualiza limiar de novidade usando Equação (3.14).

}

---

$$\mathbf{c}_j(t+1) = \mathbf{c}_j(t) - \alpha(t) \frac{\partial J_{LMM}(t)}{\partial \mathbf{c}_j(t)}, \quad (4.17)$$

$$= \mathbf{c}_j(t) - \alpha(t) \frac{\partial \rho(e(t))}{\partial e(t)} \frac{\partial e(t)}{\partial \hat{y}(t)} \frac{\partial \hat{y}(t)}{\partial z_j(t)} \frac{\partial z_j(t)}{\partial \mathbf{c}_j(t)}, \quad (4.18)$$

$$= \mathbf{c}_j(t) - \alpha(t) q(e(t)) e(t) \cdot (-1) \cdot \hat{y}_j(t) \cdot \left( \frac{z_j(t)}{\sigma_j^2} (\mathbf{x}(t) - \mathbf{c}_j(t)) \right), \quad (4.19)$$

$$= \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} z_j(t) q(e(t)) e(t) \hat{y}_j(t) (\mathbf{x}(t) - \mathbf{c}_j(t)), \quad (4.20)$$

em que  $\hat{y}_j(t) = f_j(\mathbf{x}, \mathbf{w}_j; t) = \mathbf{w}_j^T(t)\mathbf{x}(t) + w_{j,0}$  e para a função Huber definida na Equação (4.16), reduz para a Equação (3.34) sempre que  $|e(t)| \leq \beta$ . O pseudocódigo do modelo robusto G-LMN (ORG-LMN, de forma abreviada) é apresentado no Algoritmo 4.

Conforme mencionado anteriormente, o nosso objetivo era desenvolver um modelo LMN crescente de baixo custo computacional para identificação recursiva. Este objetivo levou ao desenvolvimento do modelo G-LMN. Como segundo objetivo, busca-se adicionar robustez a *outlier* ao modelo G-LMN, substituindo as regras LMS originais para atualizar os vetores de peso dos submodelos locais e os centros das unidades ocultas. Esse objetivo levou ao desenvolvimento do modelo ORG-LMN que, como será visto adiante nesta tese, terá desempenho superior ao modelo G-LMN em dados contaminados com *outliers*. Os experimentos computacionais que testam estas hipóteses e os resultados obtidos são apresentados no Capítulo 6.

#### 4.4 Outras variantes robustas

Vale ressaltar que, embora os modelos G-LMN e ORG-LMN foram apresentados como as principais contribuições desta tese, a simplicidade da regra LMM nos permite criar variantes robustas a *outliers* de todos os modelos locais descritos e avaliados nesta tese. Tais variantes estão descritas na Tabela 2 juntamente com suas versões originais. Todas as versões robustas apresentadas são também pequenas propostas desta tese.

É possível ver quão simples é possível criar uma versão robusta de qualquer modelo cuja estimação de parâmetros é baseada na regra LMS. Basicamente, o que muda nas regras de atualização é presença da função  $q(e(t))$ , que pode assumir várias expressões a depender da escolha da função  $\rho(e(t))$ .

Tabela 2 – Regras recursivas robustas desenvolvidas a partir da regra LMM.

Modelos Avaliados		
Modelo	Regra Original - LMS	Versão Robusta - LMM
LLM	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)h(j^*, j; t)[\mathbf{x}(t) - \mathbf{w}_j(t)]$ $\mathbf{a}_j(t+1) = \mathbf{a}_j(t) + \alpha'(t)h(j^*, j; t)e(t)\frac{\mathbf{x}(t)}{\varepsilon + \ \mathbf{x}(t)\ ^2}$	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)q(e(t))h(j^*, j; t)[\mathbf{x}(t) - \mathbf{w}_j(t)]$ $\mathbf{a}_j(t+1) = \mathbf{a}_j(t) + \alpha'(t)q(e(t))h(j^*, j; t)e(t)\frac{\mathbf{x}(t)}{\varepsilon + \ \mathbf{x}(t)\ ^2}$
RBFN	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)e(t)\mathbf{z}(t)$	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)q(e(t))e(t)\mathbf{z}(t)$
LMN	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)e(t)\frac{\mathbf{x}(t)}{\varepsilon + \ \mathbf{x}(t)\ ^2}$	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)q(e(t))e(t)\frac{\mathbf{x}(t)}{\varepsilon + \ \mathbf{x}(t)\ ^2}$
INC-LLM	$\hat{y}_{s_1}(t+1) = \hat{y}_{s_1}(t) + \alpha e_{s_1}(t)$ $\mathbf{a}_{s_1}(t+1) = \mathbf{a}_{s_1}(t) + \alpha e_{s_1}(t)(\mathbf{x}(t) - \mathbf{w}_{s_1})$	$\hat{y}_{s_1}(t+1) = \hat{y}_{s_1}(t) + \alpha q(e(t))e_{s_1}(t)$ $\mathbf{a}_{s_1}(t+1) = \mathbf{a}_{s_1}(t) + \alpha q(e(t))e_{s_1}(t)(\mathbf{x}(t) - \mathbf{w}_{s_1})$
RAN	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)e(t)\mathbf{z}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} z_j(t)e(t)w_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t))$	$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)q(e(t))e(t)\mathbf{z}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} q(e(t))z_j(t)e(t)w_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t))$
G-LMN	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)z_j(t)e(t)\mathbf{x}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} z_j(t)e(t)\hat{y}_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t))$	$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)q(e(t))z_j(t)e(t)\mathbf{x}(t)$ $\mathbf{c}_j(t+1) = \mathbf{c}_j(t) + \frac{\alpha(t)}{\sigma_j^2} q(e(t))z_j(t)e(t)\hat{y}_j(t)(\mathbf{x}(t) - \mathbf{c}_j(t))$

#### 4.5 Considerações finais

Neste capítulo, realizou-se uma breve discussão sobre a importância do tratamento de *outliers* na tarefa de identificação de sistemas e a regra LMM (CHAN; ZHOU, 2010) que é uma alternativa robusta a *outliers* à regra LMS foi descrita. Com esses fundamentos, introduziu-se o modelo ORG-LMN, que é uma variante robusta a *outliers* do modelo G-LMN descrito no capítulo anterior. O modelo ORG-LMN na seção anterior possui as seguintes características: (i) estrutura online crescente para lidar com a não estacionariedade de processos; (ii) regra de atualização recursiva para rastreamento *online* de mudanças na dinâmica do sistema; (iii) requisitos de uso de memória menores que os da regra RLS e (iv) robustez no tratamento adequado de amostras discrepantes de entrada-saída.

## 5 METODOLOGIA

Neste capítulo, a metodologia deste trabalho será apresentada. O capítulo será dividido da seguinte forma: (i) formas de validação dos modelos; (ii) medidas de avaliação de desempenho; e (iii) apresentação dos conjuntos de dados avaliados.

### 5.1 Validação dos modelos

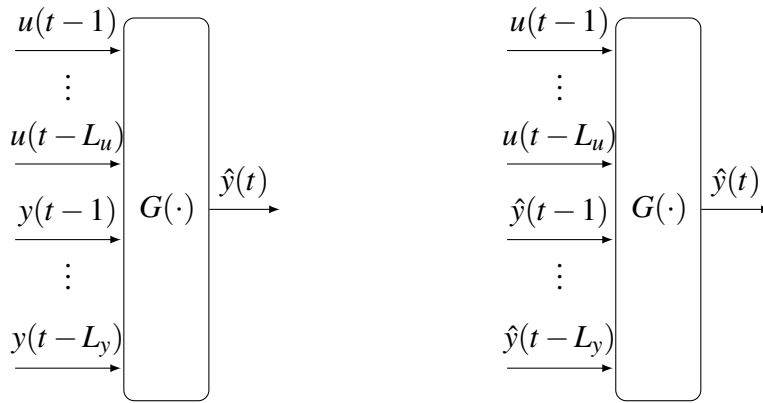
Com o intuito de avaliar o desempenho de um modelo preditivo em tarefas de identificação de sistemas, é comum a prática de dividir as amostras de dados disponíveis em duas partes: (i) estimação e (ii) validação. Dados de estimação são também conhecidos como de treinamento que é, grosso modo, a etapa de estimação dos parâmetros do modelo. Já a validação também conhecida como etapa de teste, ou seja, o modelo deve ser avaliado em um conjunto de dados de teste que é composto por amostras desconhecidas pelo modelo até então. É necessário verificar se eles incorporam ou não as características de interesse do sistema original (AGUIRRE, 2015).

A validação é a última etapa do processo de identificação de um sistema sendo a que certifica como o modelo estimado (ajustado) emula corretamente a dinâmica do sistema real. Uma importante observação é quanto aos dados utilizados. Os dados utilizados para a identificação e para a validação devem ser diferentes, pois o objetivo da identificação é obter o modelo mais geral possível. Após o modelo ser estimado, ele é utilizado para simular a dinâmica da saída do sistema. Para isto existem duas tarefas que diferem entre si na forma como as saídas passadas prevêm o futuro:

1. predição direta e
2. simulação livre.

Na predição, o modelo consegue prever um ou vários passos à frente com base nas entradas anteriores ( $u_{t-1} \cdots u_{t-L_u}$ ) e saídas observadas anteriores ( $y_{t-1} \cdots y_{t-L_u}$ ). Na simulação livre, o modelo prevê os resultados futuros com base nas entradas anteriores ( $u_{t-1} \cdots u_{t-L_u}$ ) e nas saídas preditas (em vez de saídas observadas) para prever a saída atual.

Dentre os cenários de predição, o mais comum é o cenário de um passo à frente (*one-step ahead*, OSA) representado na Figura 8 (a). Aqui, o vetor de regressores  $\mathbf{x}(t)$  no instante  $t$  é construído com base nos valores anteriores observados de entrada e saída e este vetor é utilizado para calcular a saída prevista  $\hat{y}(t)$  no próximo instante  $t$ , conforme definido na Equação



(a) Predição OSA.

(b) Simulação livre.

Figura 8 – Cenários para validação do modelo. Adaptado de Nelles (2020).

(2.2).

A saída prevista  $\hat{y}(t)$  não é utilizada para prever a próxima saída  $\hat{y}(t+1)$ . Nota-se que a predição de um passo à frente não depende do valor anterior a ela, sendo assim, o resíduo no instante  $t$  é  $e(t) = y(t) - \hat{y}(t)$ . Porém, esta metodologia de validação não implica necessariamente que o modelo capturou a dinâmica correta do sistema. Isto é particularmente verdade para sistemas dinâmicos não lineares, pois até mesmo modelos ruins podem apresentar boas predições de passo à frente. Isso ocorre pois algoritmos de estimação de parâmetros normalmente tendem a minimizar a soma do quadrado dos resíduos, computados em um esquema de predição do estilo OSA. Com isso, os erros de predição tendem a ser pequenos, mesmo para sistemas não lineares (AGUIRRE, 2015).

Já nos cenários de de validação via simulação livre, as predições anteriores são utilizadas para construir o vetor de regressores  $\mathbf{x}$ , e, assim, pode-se continuar fazendo previsões a partir do modelo obtido conforme representado na Figura 8 (b) . Neste caso, vetor de regressores é expresso tal como

$$\mathbf{x}(t) = \begin{pmatrix} \hat{y}(t-1) \\ \vdots \\ \hat{y}(t-L_y) \\ u(t-1) \\ \vdots \\ u(t-L_u) \end{pmatrix}. \quad (5.1)$$

Observe que agora as saídas previstas anteriormente  $\{\hat{y}(t-1), \dots, \hat{y}(t-L_y)\}$  são utilizadas para prever a saída  $\hat{y}(t)$ . A diferença fundamental entre simulação livre e predição OSA

é que a simulação requer a realimentação das saídas previstas para prever os valores subsequentes da saída  $\hat{y}(t)$ . Isto faz com que a etapa de validação se torne difícil e que se tenha mais cuidado para garantir a estabilidade do modelo (NELLES, 2020).

As tarefas de predição OSA e simulação livre são os dois os casos extremos. Nelles (2020) mostra que existe um caso intermediário entre esses dois cenários e é chamado de predição  $k$  passos à frente ou múltiplos passos à frente (AGUIRRE, 2015). E é exatamente o valor de  $k$  que vai definir o horizonte de predição, ou seja, quando  $k \rightarrow \infty$ , tem-se o caso de simulação livre novamente (horizonte de predição se aproxima do infinito). Nesta tese, na maioria das aplicações, utiliza-se a simulação livre para validação do modelo.

## 5.2 Medidas de Avaliação de Desempenho

### 5.2.1 Erro médio quadrático

Uma escolha comum para avaliar os modelos é através da raiz do erro quadrático médio (*root mean squared error*, RMSE), calculado a partir dos resultados previstos pelo modelo e as observações reais em todo o conjunto de dados de teste. Todos os modelos são numericamente avaliados por meio do RMSE expresso matematicamente por

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N e^2(t)}, \quad (5.2)$$

em que  $e(t) = y(t) - \hat{y}(t)$  e  $N$  é a quantidade de amostras do conjunto de estimação ou de validação.

### 5.2.2 Teste de Kolmogorov-Smirnov

De forma resumida, o teste de Kolmogorov-Smirnov (KS) avalia a distância entre as funções de distribuição acumulada empíricas (CDF, *cumulative distribution function*) de duas sequências de resíduos (SOONG, 2004). A hipótese nula a ser testada é que as sequências são retiradas da mesma distribuição (BARRETO; SOUZA, 2016). O conjunto final de experimentos visa avaliar o grau de similaridade, do ponto de vista estatístico, entre a sequência de resíduos gerada pelos modelos ajustados.

A utilização do teste KS no presente contexto justifica-se pela necessidade de avaliar a diferença entre os desempenhos dos modelos originais e robustos. Se dois modelos supostamente



diferentes geram sequências estatisticamente equivalentes de resíduos de acordo com o teste KS, então os dois modelos devem ser considerados equivalentes um ao outro na realidade.

Os resultados possíveis a serem obtidos a partir da aplicação deste teste de hipótese são os seguintes:

1. A aceitação da hipótese nula indica que as CDFs dos resíduos gerados pelos modelos são equivalentes.
2. Uma rejeição da hipótese nula indica que as CDFs dos resíduos gerados pelos modelos são diferentes.

### 5.3 Conjuntos de dados avaliados

Esta seção traz detalhes sobre cada conjunto de dados a ser utilizado nos experimentos computacionais realizados nesta tese e cujo os resultados serão reportados no próximo capítulo.

#### 5.3.1 *Dados sintéticos*

Detalhes sobre estes conjuntos estão na Tabela 3 conforme descritos em Narendra e Parthasarathy (1990). Além do ruído gaussiano adicionado a cada observação de saída dos dados de treinamento, uma proporção de 0%, 5%, 10%, 15%, 20%, 25% e 30% do total das saídas de treinamento também são contaminadas com *outliers* artificialmente gerados. Para isso, seleciona-se aleatoriamente uma certa proporção de valores de saída do total de dados de treinamento e adiciona-se a estes valores ruído não-gaussiano gerado a partir de uma distribuição *t*-Student de média nula e 2 graus de liberdade normalizados pelo desvio-padrão da saída. Esta metodologia de contaminação é a mesma utilizada em Mattos *et al.* (2017). Nas Figuras 9 e 10, são mostrados os valores de entrada e saída dos conjuntos de dados Sintéticos 1 e 2.

#### 5.3.2 *Dados reais*

O modelos implementados nesta tese são avaliados em cinco conjuntos de dados amplamente utilizados em estudos comparativos, a saber, atuador hidráulico, secador de cabelo, neutralização de pH, *silverbox* e secador industrial. Estes conjuntos estão disponíveis para uso público a partir do sítio do repositório DaISy (*Database for Identification of Systems*) na internet

Tabela 3 – Descrição dos conjuntos de dados sintéticos usados nos experimentos computacionais.  $U(A, B)$  é um número aleatório uniforme entre  $A$  e  $B$ .

#	Saída	Entrada/Amostras		
		Estimação	Teste	Ruído
1	$y_n = y_{n-1} - 0,5 \tanh(y_{n-1} + u_{n-1}^3)$	$u_n = \mathcal{N}(u_n 0, 1)$ $-1 \leq u_n \leq 1$ 150 amostras	$u_n = \mathcal{N}(u_n 0, 1)$ $-1 \leq u_n \leq 1$ 150 amostras	$\mathcal{N}(0, 0, 0025)$
2	$y_n = 0,3y_{n-1} + 0,6y_{n-2} + 0,3\text{sen}(3\pi u_{n-1}) + 0,1\text{sen}(5\pi u_{n-1})$	$u_i = U(-1, 1)$ 500 amostras	$u_i = \sin(2\pi i/250)$ $\sin(2\pi i/10)$ 500 amostras	$\mathcal{N}(0, 0, 18)$

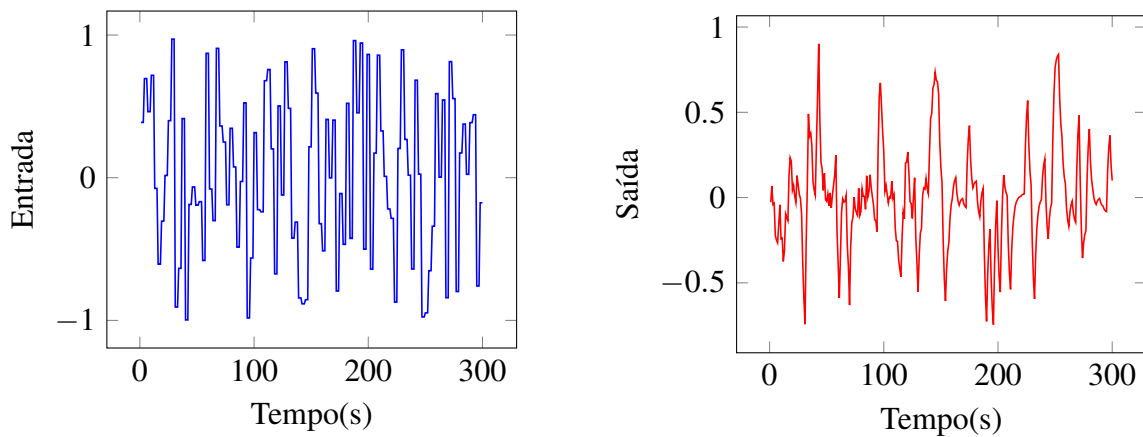


Figura 9 – Valores medidos de entrada e saída do conjunto de dados Sintético 1.

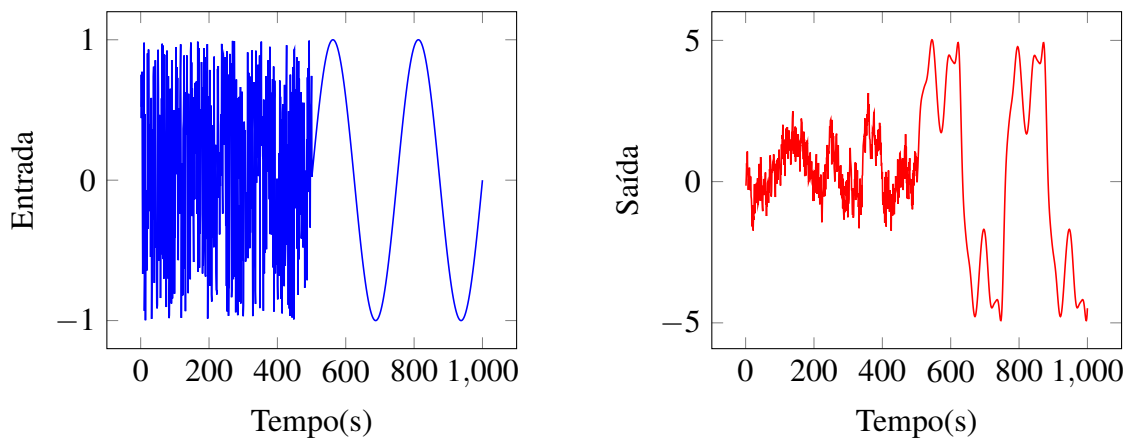


Figura 10 – Valores medidos de entrada e saída do conjunto de dados Sintético 2.

<sup>1</sup>. Algumas características importantes dos conjuntos de dados supracitados estão na Tabela 4.

Uma breve descrição de cada conjunto de dados é realizada a seguir.

<sup>1</sup> <https://homes.esat.kuleuven.be/~smc/daisy/>

Tabela 4 – Características dos conjuntos de dados avaliados.

Conjuntos Avaliados	Amostras de Estimação	Amostras de Teste	$\hat{L}_u$	$\hat{L}_y$
Atuador Hidráulico	512	512	4	4
Secador de Cabelo	500	500	5	5
Neutralização de pH	200	800	5	5
Silverbox	91072	40000	10	10
Secador Industrial (MIMO)	600	267	4	4

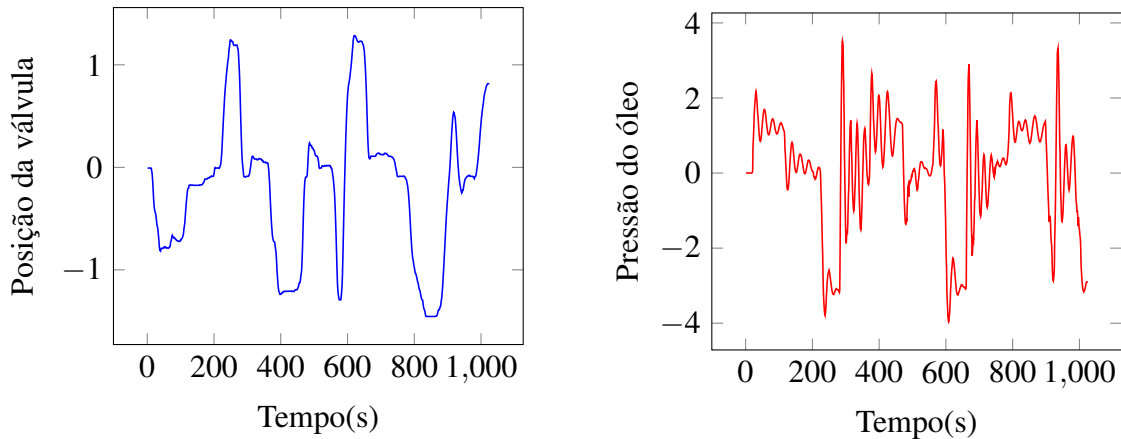


Figura 11 – Valores medidos de entrada (posição da válvula) e saída (pressão do óleo) do conjunto de dados atuador hidráulico.

### 5.3.2.1 Atuador hidráulico

Para a tarefa de identificação de sistemas, escolheu-se um conjunto de dados que tem como base um atuador hidráulico de uma estrutura mecânica hidráulica (grua). Este conjunto de dados consiste em duas séries temporais (entrada: abertura da válvula, saída: pressão do óleo), cada uma composta de 1024 medições coletadas de um atuador hidráulico em um guindaste (SJÖBERG *et al.*, 1995) conforme mostrado na Figura 11. Usou-se 512 amostras para construção de modelo e estimativa de parâmetros e a metade restante para validação de modelo (teste).

### 5.3.2.2 Secador de cabelo

Este conjunto de dados corresponde a um sistema eletromecânico SISO de uma configuração montada em laboratório para funcionar como um secador de cabelo. Neste sistema, a saída é a temperatura do ar que é medida por um termopar. A entrada é a tensão aplicada sobre o dispositivo de aquecimento. Das 1000 amostras disponíveis que são ilustradas na Figura 12, usou-se a primeira metade para a construção do modelo (treinamento) e o restante para avaliação do modelo (teste).

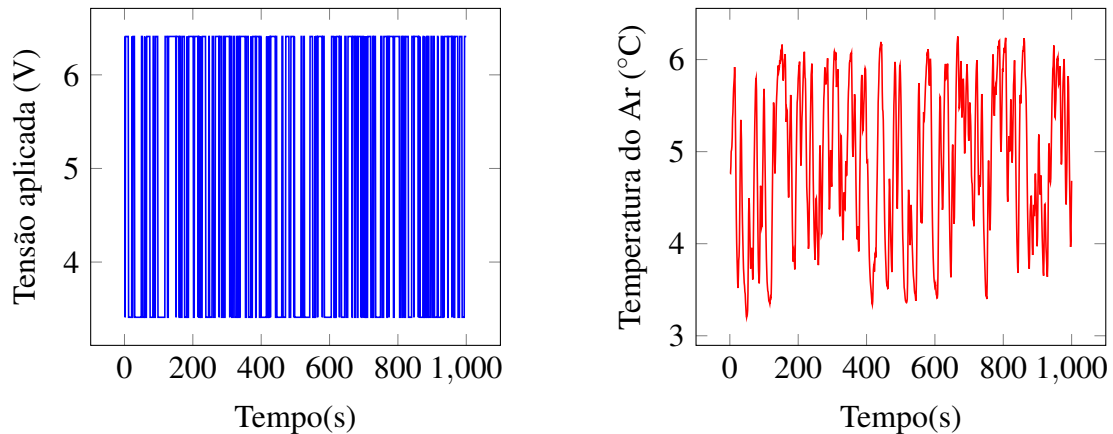


Figura 12 – Valores medidos de entrada(tensão aplicada) e saída(temperatura) do conjunto de dados secador de cabelo.

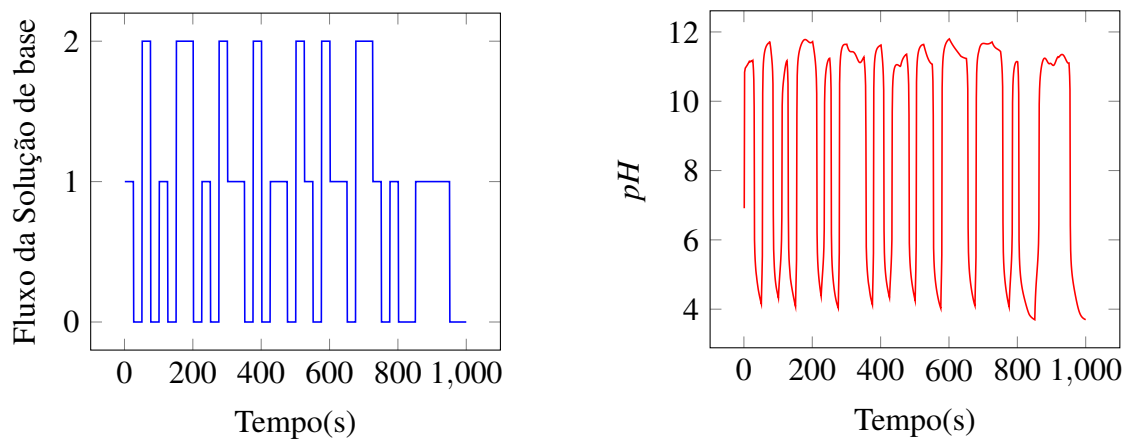


Figura 13 – Valores medidos de entrada(fluxo da solução de base) e saída( $pH$ ) do conjunto de dados neutralização de  $pH$ .

### 5.3.2.3 Neutralização de $pH$

As amostras desse conjunto de dados vêm de um processo de neutralização de  $pH$  em um tanque de agitação com volume constante. A entrada do sistema é o fluxo da solução de base e a saída é o valor  $pH$  da solução no tanque conforme exibido na Figura 13. Nesta tese, usou-se as primeiras 200 amostras para construção de modelo e estimativa de parâmetros e as próximas 800 amostras para validação de modelo (teste).

### 5.3.2.4 Silverbox

No conjunto de dados *Silverbox*, as amostras são obtidas a partir de um circuito elétrico simulando um sistema massa-mola-amortecedor, que corresponde a um sistema dinâmico não linear com realimentação e comportamento linear dominante (SCHOUKENS *et al.*, 2003).

Este conjunto de dados contém um total de 131072 amostras de cada sequência  $u(t)$

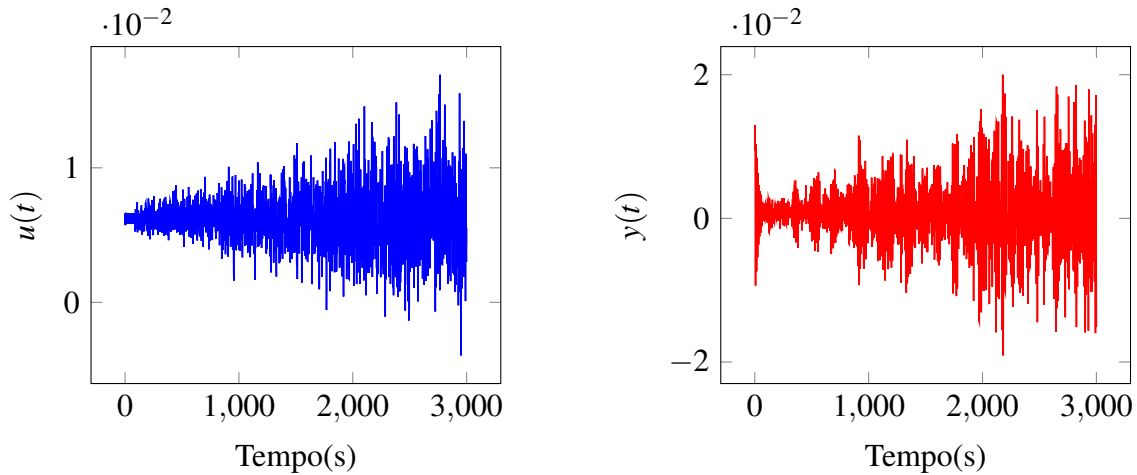


Figura 14 – Valores medidos de entrada e saída do conjunto de dados *silverBox*.

e  $y(t)$ , onde as primeiras 40000 amostras de  $u(t)$  e  $y(t)$  foram utilizadas para a construção do modelo e os 91072 restantes para a validação do modelo. Como o conjunto de dados *Silverbox* é extenso, o treinamento não se repete por várias épocas. Na verdade, o modelo é treinado em um modo totalmente *online* com estimativa de parâmetros recursiva, exigindo apenas uma passagem pelos dados para convergir. As primeiras 3000 amostras são exibidas na Figura 14.

#### 5.3.2.5 Secador Industrial - caso MIMO

Finalmente, os modelos propostos também foram testados em um conjunto de dados originado de um sistema MIMO. Para isso, selecionou-se o conjunto de dados secador industrial descrito em Bittanti e Picci (1996) e Chou e Maciejowski (1997). Este sistema tem 3 entradas ( $n_u = 3$ ), a saber: taxa de fluxo da matéria prima  $u_1(t)$ , taxa de fluxo de combustível  $u_2(t)$  e velocidade do ventilador de exaustão de gás quente  $u_3(t)$ . Além disso, o sistema conta com 3 saídas ( $n_y = 3$ ): temperatura do bulbo seco  $y_1(t)$ , temperatura do bulbo úmido  $y_2(t)$  e o teor de umidade do material de matéria-prima  $y_3(t)$ . Essas entradas e saídas são mostradas na Figura 15. Para melhor visualização das amplitudes da entrada  $u_1$ , estas foram divididas por um fator de 17.

## 5.4 Considerações finais

Este capítulo apresentou um panorama da metodologia utilizada nesta tese. Primeiramente, as duas formas utilizadas para simular a dinâmica da saída do sistema após ele ser estimado foram apresentadas: (i) predição e (ii) simulação livre. As medidas de avaliação de desempenho que serão utilizadas para validação dos modelos, a saber, erro médio quadrático

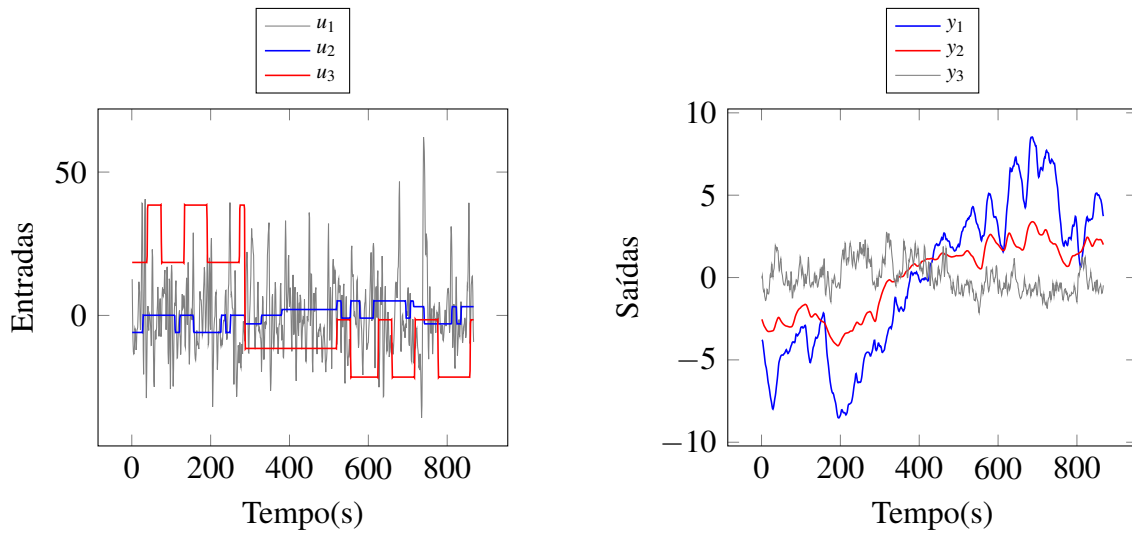


Figura 15 – Sequência de entradas e saídas do conjunto de dados MIMO secador industrial.

e teste de Kolmogorov-Smirnov (KS), também foram detalhadas. Dois conjuntos de dados sintéticos e cinco conjuntos de dados reais utilizados em estudos comparativos nesta tese também foram descritos neste capítulo. Isto posto, no próximo capítulo, os resultados de um amplo estudo comparativo entre os modelos reportados serão apresentados.

## 6 RESULTADOS

Neste capítulo, serão reportados os resultados de uma comparação abrangente de desempenho entre os seguintes modelos:

- **Modelos Locais de Tamanho Fixo:** LLM, RBF e LMN, descritos no Capítulo 2.
- **Modelos Locais Crescentes:** INC-LLM, RAN e G-LMN, descritos no Capítulo 3.
- **Variantes robustas a outliers,** descritas no Capítulo 4.

O objetivo dos experimentos é avaliar a capacidade dos modelos propostos em fornecer um modelo incremental e preciso para a identificação recursiva de sistemas dinâmicos na presença de *outliers*.

Todos os modelos avaliados foram implementados utilizando a linguagem do *software* Octave<sup>1</sup> e os códigos correspondentes estão disponíveis mediante solicitação. Algumas características importantes dos conjuntos de dados avaliados já foram resumidas na Tabela 3 e 4. A acurácia de todos os modelos é avaliada pelos valores de RMSE conforme a Equação (5.2).

*Outliers* foram introduzidos artificialmente em dados de treinamento em diferentes proporções. Para tanto, seguiu-se o mesmo procedimento utilizado por Mattos *et al.* (2017). Os resultados reportados correspondem à avaliação do modelo em dados de validação livres de *outliers*. A justificativa para essa abordagem é avaliar como os *outliers* afetam o processo de estimação de parâmetros (treinamento) e seu impacto na fase de validação (teste) do modelo. Os modelos treinados foram testados sob o regime de *simulação livre*, no qual os valores de saída preditos são realimentados para o vetor de regressão de entrada - ver Equação (5.1).

### 6.1 Resultados nos conjuntos de dados sintéticos

Estes conjuntos de dados foram inicialmente utilizados nas primeiras simulações com os modelos locais de estrutura fixa (LLM, RBF e LMN) e suas versões robustas para avaliarmos a inclusão da regra LMM (ZOU *et al.*, 2000). Alguns dos resultados desta tese se encontram em Bessa e Barreto (2019a).

Antes de avaliarmos o desempenho dos modelos robustos em relação a suas versões originais, realizou-se alguns testes com diferentes algoritmos para quantização vetorial (QV) do espaço de entrada, visto que as abordagens testadas inicialmente possuem uma quantidade pré-definida de modelos locais. Algoritmos como WTA (*winner take all*), SOM (*self organizing*

---

<sup>1</sup> [www.gnu.org/software/octave/](http://www.gnu.org/software/octave/)

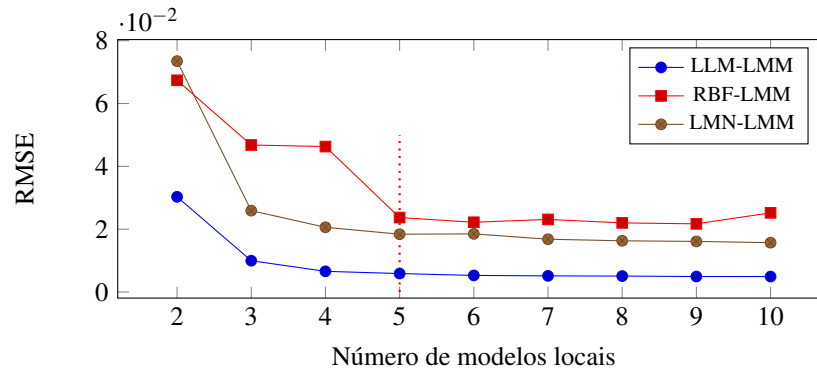


Figura 16 – RMSE no conjunto de teste *versus* número de modelos durante o treinamento com contaminação de 20% de *outliers* (dados sintéticos 2) (versões robustas).

*maps*), FSCL (*frequency-sensitive competitive learning*) e NG (*neural gas*) foram testados nesta fase. Para estes testes, utilizou-se o modelo LLM com a regra LMS e o conjunto de dados sintéticos 1. Os resultados dessas comparações são apresentados no Apêndice A. Conforme observado, a utilização de diferentes algoritmos na fase de quantização vetorial do espaço de entrada não influencia no comportamento dos resíduos, ou seja, a distribuição dos resíduos é semelhante. O algoritmo SOM obteve melhores resultados de RMSE nesta fase, dessa forma, ele será utilizado em todos os testes com os modelos de estrutura fixa daqui em diante.

Definiu-se 200 épocas de treinamento em 5 realizações diferentes para todos os modelos. O melhor modelo é, então, escolhido. O número de neurônios foi definido em  $S = 5$  após alguma experimentação inicial com os dados dos conjuntos sintéticos 1 e sintéticos 2. Valores de  $S$  maiores que 5 não melhoraram significativamente os resultados de um modo geral, enquanto valores menores conduziram a desempenho inferiores de todos os modelos avaliados, robustos ou não. Um exemplo desse desempenho *versus* o número de modelos locais no conjunto de dados sintéticos 2 utilizando as abordagens robustas com os dados de treinamento contaminados numa proporção de 20% é mostrado na Figura 16.

As taxas de aprendizado inicial e final foram definidas em  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos.

Os valores de RMSE para o conjunto de dados sintéticos 1 estão mostrados na Tabela 5. Ao analisar esta tabela, pode-se observar o melhor desempenho para as versões robustas dos modelos locais avaliados. A tendência de as versões robustas sofrerem menor influência dos *outliers* durante a fase de treinamento e, conseqüentemente, apresentarem melhor desempenho no conjunto de teste, pode ser melhor apreciada na Figura 17. Nestas, os desempenhos dos modelos de estrutura fixa original e robusto são avaliados em função da proporção de *outliers* no conjunto de treinamento aumenta. Pode-se observar que as versões robustas sentem menos a



Tabela 5 – RMSE na fase de teste utilizando conjunto de dados sintéticos 1.

Modelos Locais Avaliados						
Outliers	LLM-LMS	LLM-LMM	RBF-LMS	RBF-LMM	LMN-LMS	LMN-LMM
0%	4,50E-03	4,44E-03	2,07E-02	2,20E-02	1,99E-02	2,30E-02
5%	5,37E-03	4,10E-03	2,74E-02	2,22E-02	1,96E-02	2,13E-02
10%	5,86E-03	4,48E-03	2,58E-02	2,27E-02	3,51E-02	1,85E-02
15%	7,98E-03	4,73E-03	2,59E-02	2,24E-02	3,70E-02	1,98E-02
20%	9,53E-03	6,04E-03	2,48E-02	2,34E-02	5,39E-02	2,07E-02
25%	1,35E-02	1,15E-02	2,90E-02	2,43E-02	5,42E-02	2,70E-02
30%	1,66E-02	1,23E-02	2,93E-02	2,49E-02	6,68E-02	2,78E-02

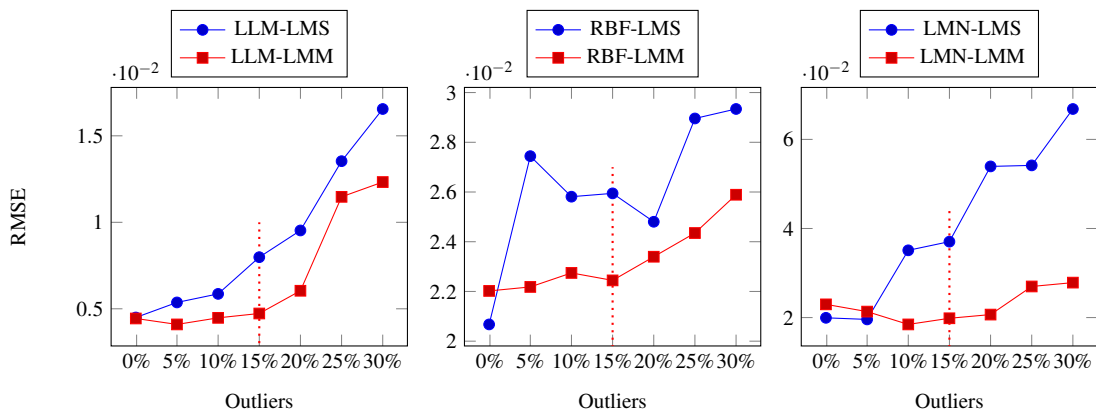
Figura 17 – RMSE no conjunto de teste para diferentes proporções de *outliers* durante o treinamento com 5 modelos locais (dados sintéticos 1): (a) LLM, (b) RBF e (c) LMN.

Tabela 6 – RMSE na fase de teste utilizando conjunto de dados sintéticos 2.

Modelos Locais Avaliados						
Outliers	LLM-LMS	LLM-LMM	RBF-LMS	RBF-LMM	LMN-LMS	LMN-LMM
0%	3,22E-03	3,50E-03	1,44E-02	1,30E-02	1,61E-02	1,51E-02
5%	5,00E-03	3,62E-03	2,04E-02	1,36E-02	1,41E-02	1,47E-02
10%	6,07E-03	4,80E-03	3,28E-02	2,23E-02	3,03E-02	1,43E-02
15%	8,43E-03	4,76E-03	3,38E-02	2,36E-02	3,36E-02	1,55E-02
20%	1,16E-02	5,87E-03	4,37E-02	2,37E-02	5,02E-02	1,84E-02
25%	1,52E-02	9,51E-03	4,40E-02	3,04E-02	5,15E-02	2,42E-02
30%	2,11E-02	1,08E-02	5,26E-02	3,19E-02	6,13E-02	2,52E-02

presença dos *outliers* pois tendem a apresentar menor RMSE que suas versões originais, com especial destaque para o modelo LMN robusto - ver Figura 17 (c).

Os desempenhos dos modelos avaliados para o conjunto de dados sintéticos 2 seguem o mesmo padrão de comportamento que aqueles obtidos para o conjunto de dados sintéticos 1, com as versões robustas apresentando melhor desempenho que as versões originais, especialmente quando o número de *outliers* é expressivo. Na Tabela 6 são mostrados os resultados de RMSE para todos os modelos em vários níveis de contaminação por *outliers* para os dados sintéticos 2. A curva do erro na fase de teste *versus* a contaminação do conjunto de treinamento é mostrada na Figura 18 para todos os modelos (originais e robustos) de estrutura fixa.

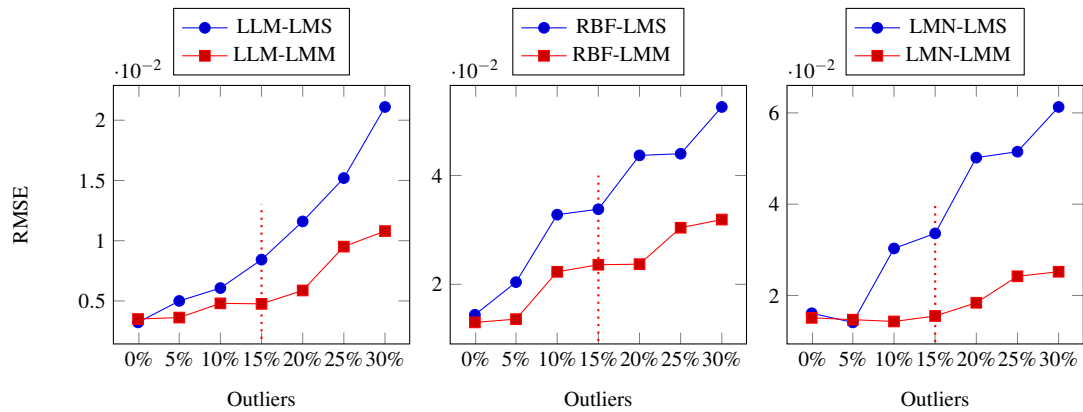


Figura 18 – RMSE no conjunto de teste para diferentes proporções de *outliers* durante o treinamento com 5 modelos locais (dados sintéticos 2): (a) LLM, (b) RBF e (c) LMN.

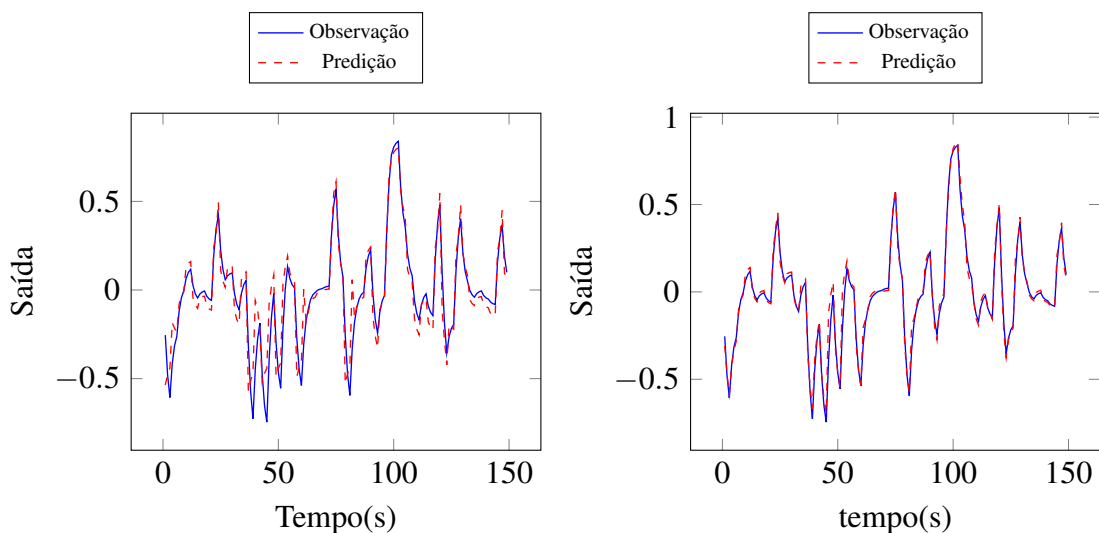


Figura 19 – Séries preditas por simulação livre (conjunto de teste) pelos modelos LLM versão original (esquerda) e versão robusta (direita) com 5 modelos locais e 10% de *outliers* (sintético 1).

Na Figura 19, são mostradas as séries preditas pelos modelos LLM-LMS (original) e LLM-LMM (robusto), na fase de teste para o conjunto de dados sintéticos 1, quando ambos os modelos são treinados com 10% de *outliers*. Como esperado, o desempenho da versão robusta do modelo LLM é superior ao de sua versão original (treinada com a regra LMS). Da mesma forma, na Figura 20, as séries preditas pelos modelos LMN (versão original e robusta) para o conjunto de dados sintéticos 2 foram reportadas. Ambos os modelos foram treinados com 10% de *outliers*. A versão que utiliza a regra LMM, mais uma vez, se mostra superior a versão original. A escolha por reportar nas Figuras 19 e 20 os resultados da predição do modelo LLM e LMN (versões originais e robustas) é arbitrária. Porém, pode-se ratificar que as estruturas RBF e LMN possuem semelhanças em suas formulações.

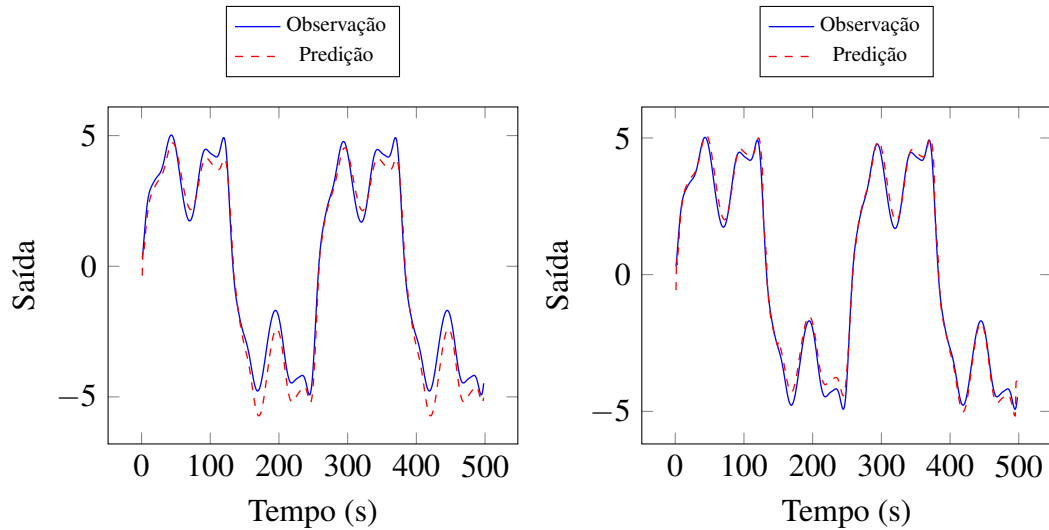


Figura 20 – Séries previstas por simulação livre (conjunto de teste) pelos modelos LMN versão original (esquerda) e versão robusta (direita) com 5 modelos locais e 10% de *outliers* (sintético 2).

## 6.2 Resultados no conjunto atuador hidráulico

Este conjunto de dados consiste em duas séries temporais (entrada: abertura da válvula, saída: pressão do óleo), cada uma composta por 1024 medições coletadas de um atuador hidráulico em um guindaste (SJÖBERG *et al.*, 1995). Usou-se 512 amostras para construção de modelo e estimativa de parâmetro e a metade restante para validação de modelo (teste). Durante o teste, os parâmetros dos modelos não são atualizados.

Definiu-se 100 épocas para treinar todos os modelos. O número de neurônios é inicialmente definido como  $S = 5$  para todos os modelos com tamanho fixo (originais e robustos), isso também se deve ao fato de valores maiores não mostrarem desempenho expressivamente superiores. Definiu-se o número de modelos locais iniciais como  $S = 2$  para o modelo crescente INC-LLM (original e robusto) e como  $S = 1$  para os modelos crescentes RAN e G-LMN (originais e robustos). As taxas de aprendizagem inicial e final foram definidas como  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos. Quanto à escolha dos hiperparâmetros, para o modelo LLM Incremental, utilizou-se: a idade máxima para remoção dos links  $a_{max} = 50$ ;  $\epsilon_b = 0,99$  e  $\epsilon_n = 0,005$  - ver Equação (3.2); número máximo de modelos  $S = 10$ ;  $\lambda = 100$ . Uma grande dificuldade enfrentada pelo modelo incremental LLM é a quantidade de hiperparâmetros a ser definida antes da estimação do modelo. Para os modelos RAN e GLMN, definiu-se  $\kappa = 0,9$ ,  $\delta_{max} = \max(\|\mathbf{x} - \mathbf{c}_0\|)$ ,  $\delta_{min} = \delta_{max}/10$ ,  $\tau = 10$ , e  $\epsilon = 10^{-4}$ . Perceba, que parte dos parâmetros dos dois últimos modelos foram calculados em função dos próprios dados de entrada.

Os valores RMSE em função da porcentagem de *outliers* são mostrados nas Tabelas

Tabela 7 – RMSE na fase de teste para o conjunto de dados atuador hidráulico para dois níveis de contaminação de *outliers* (estrutura fixa,  $S = 5$ ).

Modelos	0% outliers		15% outliers	
	RMSE	STD	RMSE	STD
LLM-LMS	3,21E-02	2,56E-03	2,17E-01	6,87E-03
LLM-LMM	2,10E-02	8,02E-03	8,23E-02	5,93E-03
RBF-LMS	3,04E-01	3,75E-03	4,37E-01	3,62E-03
RBF-LMM	2,76E-01	1,56E-03	2,76E-01	4,90E-03
LMN-LMS	3,98E-01	3,15E-03	6,10E-01	1,24E-03
LMN-LMM	3,68E-01	1,12E-03	1,19E-01	1,13E-03

Tabela 8 – RMSE na fase de teste para o conjunto de dados atuador hidráulico para dois níveis de contaminação de *outliers* (estruturas crescentes).

Modelos	0% outliers		15% outliers	
	RMSE	STD	RMSE	STD
INC-LLM-LMS	3,14E-02	1,08E-04	2,12E-01	3,65E-04
INC-LLM-LMM	2,06E-02	1,01E-04	6,06E-02	8,04E-04
RAN-LMS	4,63E-02	2,14E-05	6,45E-01	6,34E-05
RAN-LMM	2,38E-02	1,89E-05	1,18E-01	2,17E-05
G-LMN	1,36E-02	4,68E-05	2,97E-01	3,98E-05
ORG-LMN	2,19E-02	1,78E-05	4,07E-02	1,33E-05

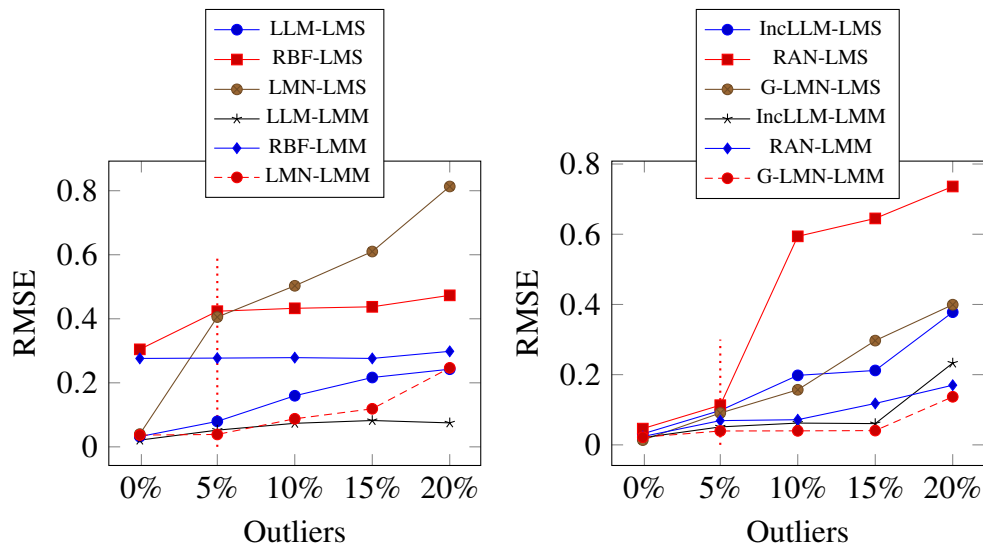


Figura 21 – Valores de RMSE na fase de teste em função da quantidade de *outliers* dos dados de treinamento (atuador hidráulico).

7 e 8. Como esperado, os modelos robustos tiveram um desempenho melhor do que as versões não robustas. Além disso, o modelo ORG-LMN foi capaz de funcionar melhor do que o modelo LMN-LLM de tamanho fixo e do que o modelo RAN-LMM. O desempenho do modelo ORG-LMN proposto é praticamente insensível à presença de *outliers*. O desempenho dos modelos

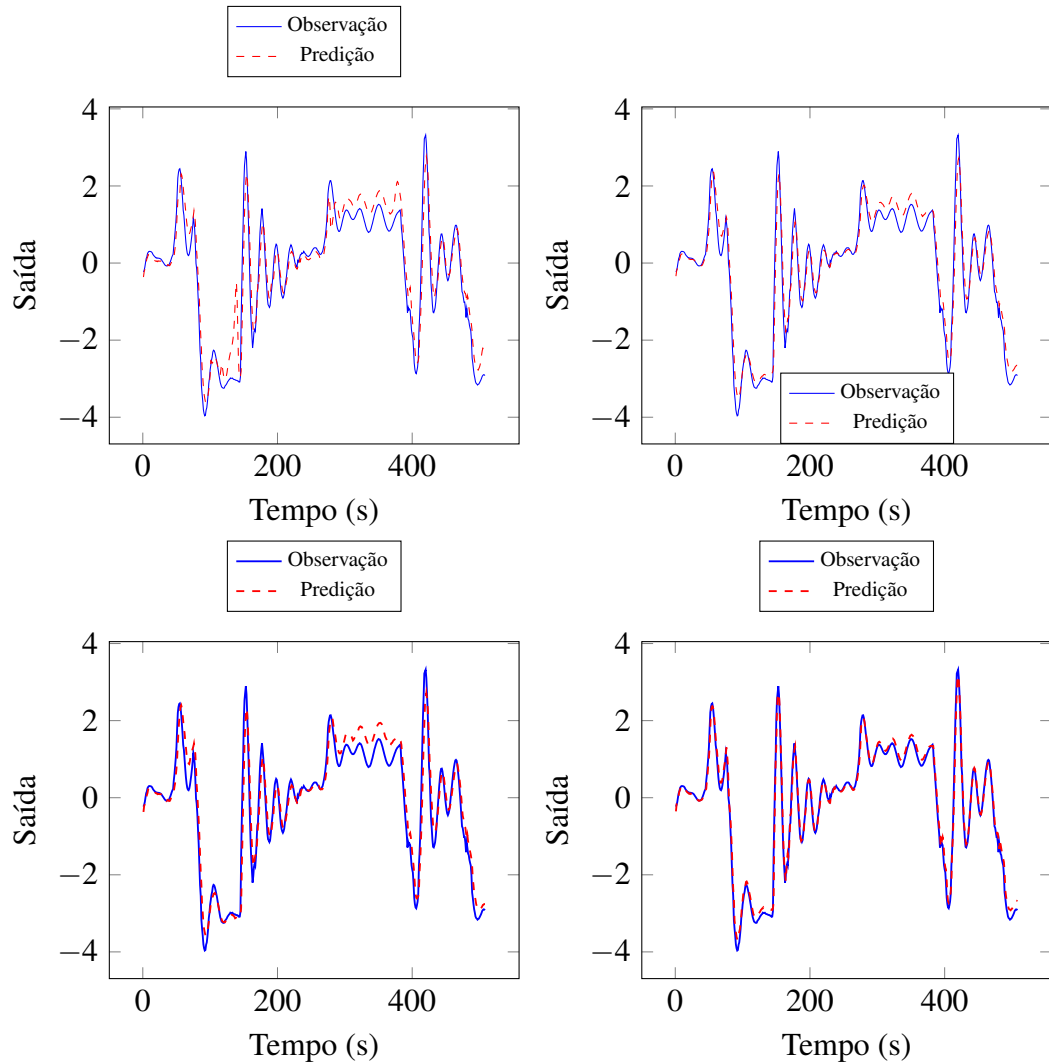


Figura 22 – Séries previstas por simulação livre (conjunto de teste) pelos modelos LMN com estrutura fixa (original e robusto)(superior), G-LMN versão original (inferior esquerda) e ORG-LMN versão robusta (inferior direita) com 15% de *outliers* (atuador hidráulico).

crecentes, de modo geral, foi melhor do que os modelos de tamanho fixo. Isso ratifica a nossa hipótese que modelos crescentes são melhores para lidar com uma possível não estacionariedade do sistema.

A melhoria no desempenho alcançada pelo modelo ORG-LMN em relação a sua abordagem baseada em LMS (ambos propostos nessa tese) para o conjunto de dados do atuador hidráulico pode ser visualizada na Figura 22. A escolha dos dois modelos para visualização das séries previstas foi feita pelo motivos de ser os dois modelos propostos nessa tese. Esta figura mostra a série temporal de saída prevista típica em linhas vermelhas tracejadas, enquanto a série temporal de saída realmente observada é mostrada em linhas sólidas azuis.

A evolução correspondente do número de modelos locais para os dois modelos propostos é mostrada na linha superior da Figura 23. Esse número se estabilizou em  $S = 7$

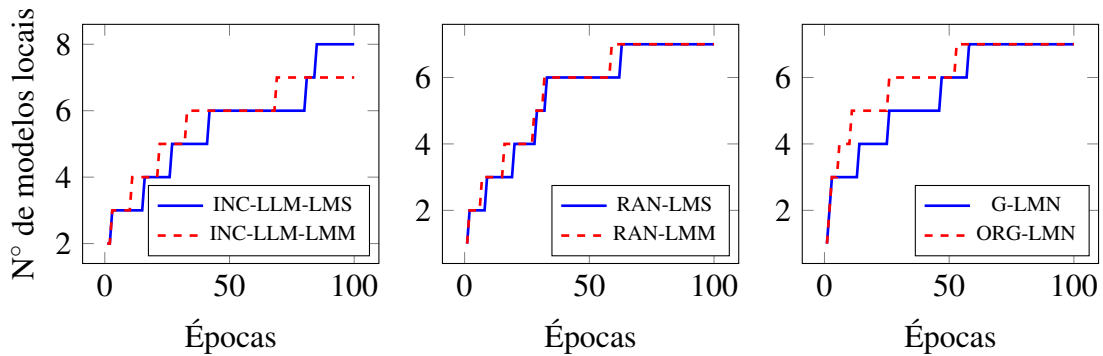


Figura 23 – Evolução do número de modelos locais em função das épocas de treinamento para modelos propostos: versões originais (linha azul sólida) e versões robustas (linha vermelha tracejada) (atuador hidráulico).

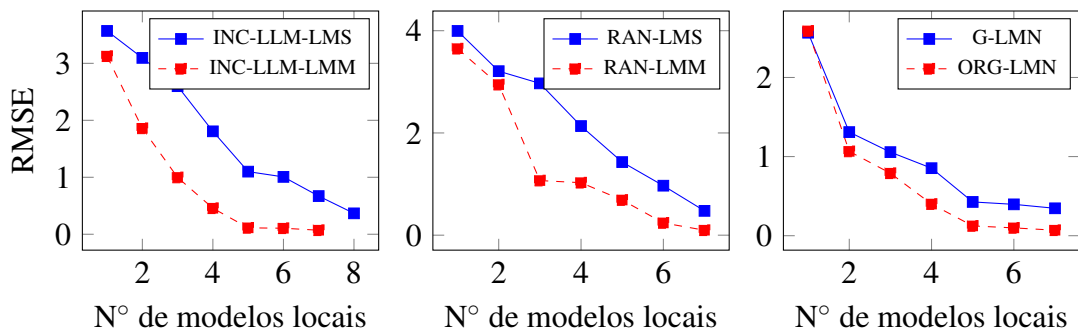


Figura 24 – RMSE *versus* número de modelos locais com 15% *outliers*: versões originais (linha azul sólida) e versões robustas (linha vermelha tracejada)(atuador hidráulico - conjunto de treinamento).

para ambos os modelos. Deve-se notar que o número de modelos locais cresce mais rápido para o modelo ORG-LMN robusto, que usa a regra de aprendizado LMM. Esta regra trata automaticamente os *outliers*, evitando sua influência negativa no processo de estimativa de parâmetros.

Um efeito colateral positivo é que o mecanismo de detecção de *outliers* da regra LMM também sinaliza para a estratégia de crescimento adotada que novos modelos locais devem ser incluídos o mais rápido possível para lidar adequadamente com a influência negativa dos *outliers*. Os efeitos desta propriedade interessante dos modelo robusto ORG-LMN podem ser vistos na taxa mais rápida de decaimento dos valores RMSE correspondentes, como mostrado na Figura 24.

### 6.3 Resultados no conjunto secador de cabelo

Este conjunto de dados corresponde a um sistema SISO mecânico de uma configuração de laboratório projetado para atuar como um secador de cabelo. Neste sistema, a temperatura

Tabela 9 – RMSE na fase de teste para o conjunto de dados secador de cabelo para dois níveis de contaminação de *outliers*.

Modelos	0% outliers		15% outliers	
	RMSE	STD	RMSE	STD
LMN-LMS	1,66E-02	3,75E-03	2,68E-02	1,63E-03
LMN-LMM	1,76E-02	1,52E-03	1,26E-02	3,13E-03
G-LMN	8,04E-03	1,08E-05	1,04E-02	1,46E-05
ORG-LMN	7,97E-03	1,02E-05	9,85E-03	1,22E-05

do ar  $y(t)$  é medida por um termopar na saída. A entrada  $u(t)$  é a tensão sobre o dispositivo de aquecimento. Das 1000 amostras disponíveis de  $u(t)$  e  $y(t)$ , a primeira metade é usada para a construção do modelo, enquanto a outra metade é usada para a validação do modelo.

Esse conjunto de dados será utilizado para efetuar uma comparação mais detalhadas nos 4 modelos baseados no modelo LMN. As taxas de aprendizagem inicial e final foram definidas como  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos. Quanto à escolha dos hiperparâmetros, para esse conjunto de dados, definiu-se  $\kappa = 0,9$ ,  $\delta_{\max} = \max(\|\mathbf{x} - \mathbf{c}_0\|)$ ,  $\delta_{\min} = \delta_{\max}/10$ ,  $\tau = 10$ , e  $\varepsilon = 10^{-4}$ . Os resultados das simulações são comentados a seguir.

Os valores RMSE para dois níveis de contaminação com *outliers* são mostrados na Tabela 9 para os dois modelos de tamanho fixo (LMN-LMS e LMN-LMM) e os modelos de crescimento propostos (G-LMN e ORG-LMN).

Dois cenários são testados, a saber, cenário 1 (livre de *outliers*) e cenário 2 (15 % de contaminação). Para o cenário 1, os desempenhos dos modelos de crescimento propostos são uma ordem de magnitude melhores (em termos de RMSE) do que os dos modelos de tamanho fixo. Para o cenário 2, o modelo ORG-LMN teve desempenho pelo menos uma ordem de magnitude melhor que os outros três modelos. As taxas de decaimento dos valores RMSE devido à inserção progressiva de modelos locais no estágio de treinamento são mostradas na Figura 25. Nesta figura, pode-se observar facilmente que a taxa de decaimento do modelo ORG-LMN é muito mais rápida do que a do modelo G-LMN.

Na Figura 26, mostra-se uma série temporal típica de saída prevista produzida pelos quatro modelos propostos para o conjunto de dados secador de cabelo. Uma inspeção visual não revela nenhuma diferença notável entre os desempenhos preditivos dos dois modelos propostos nessa tese (G-LMN e ORG-LMN). Isso acontece porque o resultado alcançado pelo modelo G-LMN já foi muito bom. Mesmo assim, o modelo robusto a *outlier* (ORG-LMN) foi capaz de melhorar ainda mais seu desempenho na tarefa de predição, como foi observado na Tabela 9.

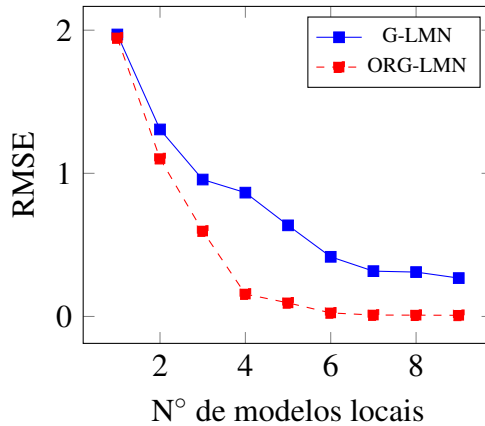


Figura 25 – RMSE na fase de treinamento versus número de modelos locais com 15% outliers: G-LMN (linha azul sólida) e ORG-LMN (linha vermelha tracejada)(secador de cabelo).

A melhoria no desempenho do modelo LMN devido à substituição da regra LMS original pela regra LMM robusta a outlier é facilmente observada nesta tabela.

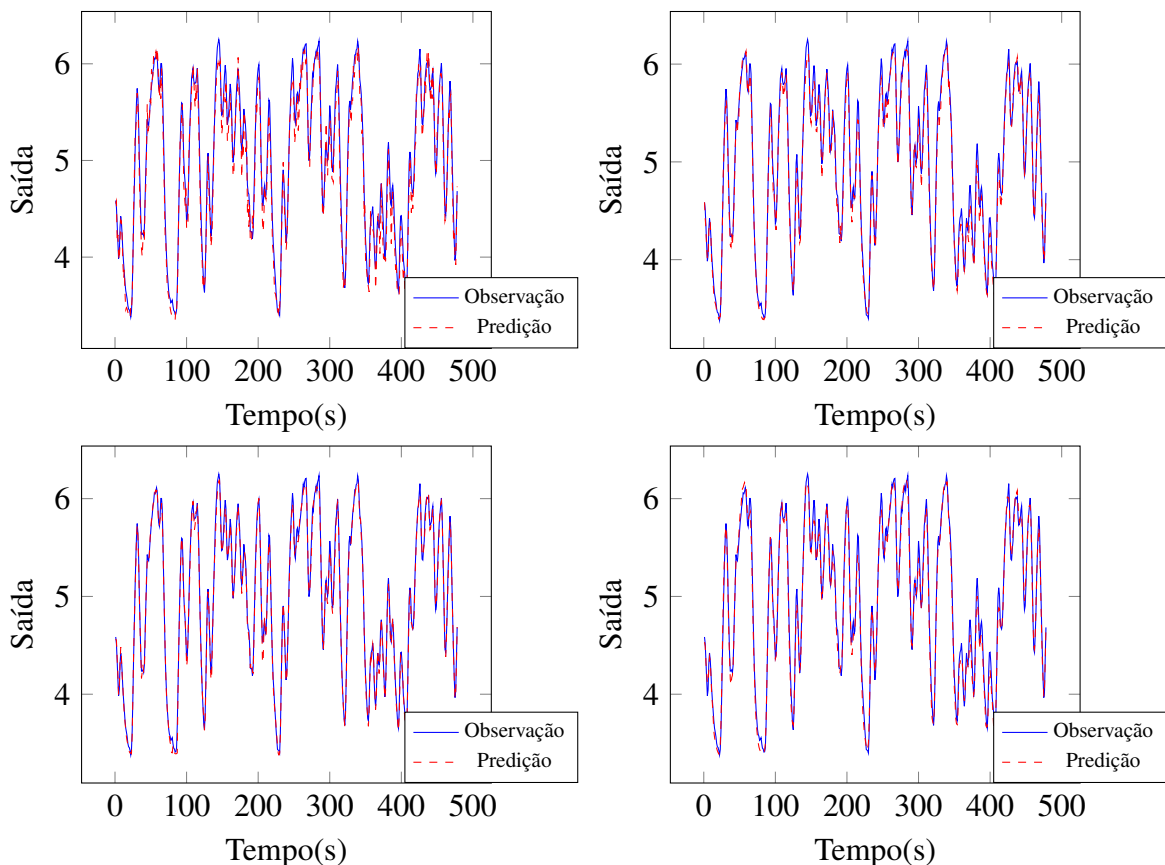


Figura 26 – Séries previstas por simulação livre (conjunto de teste) pelos modelos LMN com estrutura fixa (original e robusto)(superior) e com estrutura crescente (original e robusto)(inferior) com 15% de outliers (secador industrial).

Mais uma vez, nota-se a influência dos outliers na velocidade do crescimento no modelo conforme a Figura 27. O modelo robusto possui uma evolução do número de modelos



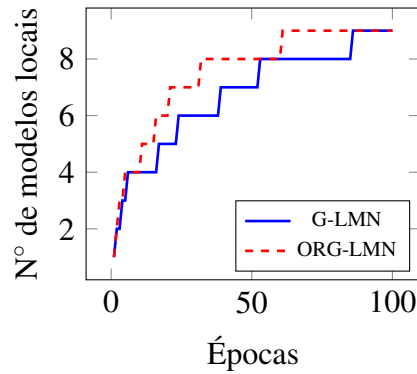


Figura 27 – Evolução do número de modelos locais em função das épocas de treinamento para modelos propostos: versão original (linha azul sólida) e versão robusta (linha vermelha tracejada) (secador de cabelo).

locais maior que o as versão original.

#### 6.4 Resultados no conjunto neutralização de $pH$

As amostras vêm de um processo de neutralização de  $pH$  em um tanque de agitação de volume constante. A entrada de controle é o fluxo da solução de base e a saída é o valor do  $pH$  da solução no tanque. Usou-se as primeiras 200 amostras para construção de modelo e estimativa de parâmetro e as próximas 800 amostras para validação de modelo (teste). Aqui, nesta seção, o foco é a comparação dos modelos crescentes versões originais (INC-LLM-LMS, RAN-LMS e G-LMN) e versões robustas propostas nesta tese (INC-LLM-LMM, RAN-LMM, ORG-LMN).

Para os testes com esse conjunto de dados, as taxas de aprendizagem inicial e final foram definidas como  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos. Para o modelo LLM incremental, utilizou-se: a idade máxima para remoção dos links  $a_{max} = 50$ ;  $\varepsilon_b = 0,99$  e  $\varepsilon_n = 0,005$  (Equação 3.2); número máximo de modelos  $N = 10$ ;  $\lambda = 100$ . Para os modelos RAN e GLMN, definiu-se  $\kappa = 0,9$ ,  $\delta_{max} = \max(\|\mathbf{x} - \mathbf{c}_0\|)$ ,  $\delta_{min} = \delta_{max}/10$ ,  $\tau = 10$ , e  $\varepsilon = 10^{-4}$ .

Uma comparação em termos de valores RMSE médios dos três modelos que utilizam regra LMS para atualização e os três modelos robustos para o conjunto de dados  $pH$  é mostrado na Figura 28 para diferentes cenários de contaminação com *outliers*. Como esperado para o cenário livre de *outliers*, os desempenhos dos modelos é praticamente o mesmo. Para este cenário, pode-se destacar o modelo INC-LLM (versão original e robusta). Estes dois modelos alcançaram menores valores de RMSE.

No entanto, conforme o nível de contaminação de valores discrepantes aumenta, pode-se observar novamente que as versões robustas alcançam um desempenho melhor do que

Tabela 10 – RMSE na fase de teste para o conjunto de dados de neutralização de  $pH$  para dois níveis de contaminação com *outliers*.

Modelos	0% outliers		15% outliers	
	RMSE	STD	RMSE	STD
INC-LLM-LMS	2,93E-02	2,95E-05	6,90E-02	7,30E-05
INC-LLM-LMM	3,07E-02	6,17E-05	4,47E-02	8,36E-05
RAN-LMS	4,05E-02	2,69E-05	7,65E-02	4,16E-05
RAN-LMM	4,00E-02	8,97E-05	5,26E-02	6,37E-05
G-LMN	3,57E-02	7,56E-05	5,04E-02	8,81E-05
ORG-LMN	3,54E-02	6,69E-05	3,90E-02	7,01E-05

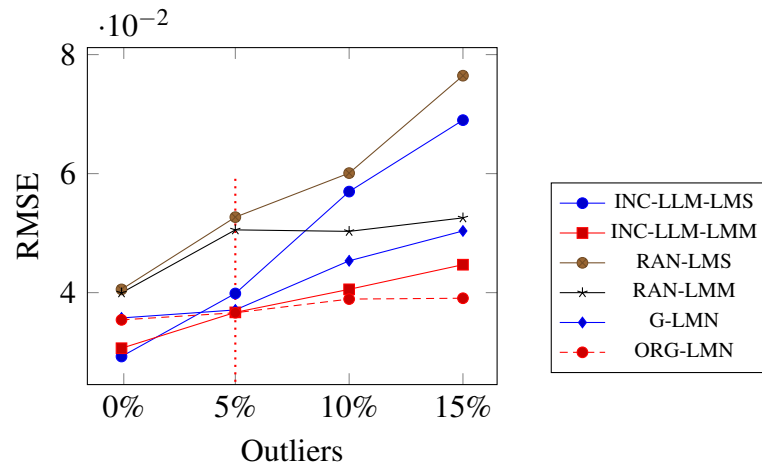


Figura 28 – Valores de RMSE na fase de teste em função da quantidade de *outliers* dos dados de treinamento (conjunto de dados de neutralização de  $pH$ ).

aqueles modelos que usam a regra de aprendizado baseada em LMS original. Para este conjunto de dados, o melhor desempenho foi alcançado pelo modelo ORG-LMN proposto, que, inclusive, possui uma curva de erro *versus* contaminação por *outliers* quase horizontal.

As séries temporais de saída previstas típicas produzidas pelos modelos propostos para o conjunto de dados neutralização de  $pH$  são mostradas na Figura 29. Como ocorreu com o conjunto de dados do secador de cabelo, uma diferença entre os desempenhos preditivos dos dois modelos é difícil de ser visualizada, mas o modelo robusto de *outlier* (ORG-LMN) foi capaz de melhorar ainda mais os resultados na tarefa de predição, como pode ser observado na Tabela 10. Os modelos robustos têm um desempenho melhor do que aqueles que usam a regra LMS usual. O modelo ORG-LMN proposto é o de melhor desempenho para este conjunto de dados.

A taxa de decaimento do erro em função da quantidade de modelos locais para as versões robustas é mostrada na Figura 30. Note que o modelo INC-LLM-LMM, que é o que mais se aproxima do nosso modelo proposto robusto ORG-LMN, precisa de 7 unidades locais para estimar o modelo e alcançar taxas de erros mais próximas do ORG-LMN. Enquanto isso, o

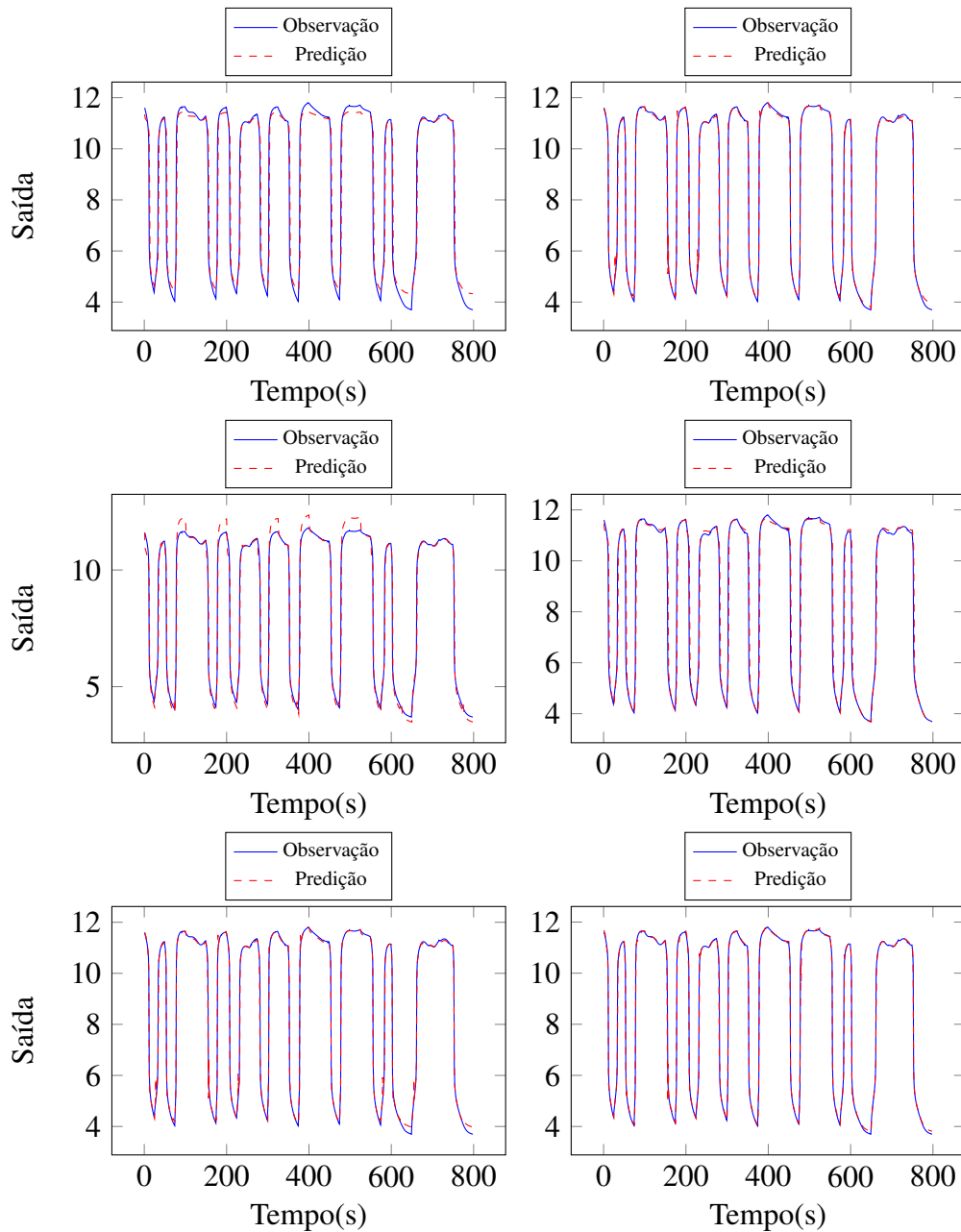


Figura 29 – Séries previstas por simulação livre (conjunto de teste) pelos modelos incrementais (versões originais - esquerda e robustas - direita): INC-LLM (superior), RAN (meio), G-LMN (inferior) com 15% de *outliers* (conjunto de dados de neutralização de *pH*).

modelo RAN, apesar de parar seu crescimento com 6 modelos locais, é o que possui maior taxa de erro durante o treinamento.

Realizada essas comparações com os 3 conjuntos de dados acima (atuador hidráulico, secador de cabelo, neutralização de *pH*), seguiu-se para uma avaliação mais datalhada dos nossos modelos propostos em um conjunto de dados com uma maior quantidade de amostras (*large scale*) e um conjunto de dados MIMO (*multiple input, multiple output*).

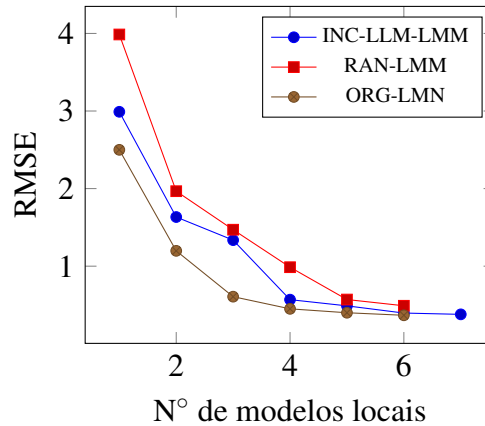


Figura 30 – RMSE na fase de treinamento versus número de modelos locais com 15% outliers: G-LMN (linha azul sólida) e ORG-LMN (linha vermelha tracejada)(conjunto de dados de neutralização de  $pH$ )

### 6.5 Resultados no conjunto *Silverbox*

Como experimento em conjuntos de dados de larga escala, decidiu-se avaliar o desempenho dos modelos LMN (com variantes crescentes e robustas). Para isso, seleciono-se o conjunto de dados *Silverbox* (SCHOUKENS *et al.*, 2003). Este conjunto dados foi obtido de um circuito elétrico simulando um sistema amortecedor massa-mola e contém um total de 131 072 amostras de cada sequência  $u(t)$  e  $y(t)$ , tal que as primeiras 40 000 amostras foram usadas para a construção do modelo e as 91 072 restantes para a validação do modelo.

Para o conjunto de dados *Silverbox*, as taxas de aprendizagem inicial e final foram definidas como  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos. Para o modelo LLM incremental, utilizou-se: a idade máxima para remoção dos links  $a_{max} = 50$ ;  $\varepsilon_b = 0,99$  e  $\varepsilon_n = 0,005$  (Equação 3.2); número máximo de modelos  $N = 10$ ;  $\lambda = 100$ . Para os modelos RAN e GLMN, definiu-se  $\kappa = 0,9$ ,  $\delta_{max} = \max(\|\mathbf{x} - \mathbf{c}_0\|)$ ,  $\delta_{min} = \delta_{max}/10$ ,  $\tau = 10$ , e  $\varepsilon = 10^{-4}$ . Para este conjunto de dados, decidiu-se, também, adotar, como neurônio inicial, a mediana dos valores de entrada. Assim, valores de desvio padrão irão tender a zero.

Como o conjunto de dados do *Silverbox* é muito longo, o treinamento não se repete por várias épocas. Em outras palavras, o modelo é usado no modo totalmente *online* com estimação de parâmetros recursiva, em que apenas uma passagem pelos dados é suficiente para o modelo convergir conforme Figura 31.

Uma comparação numérica em termos de valores RMSE das quatro variantes do modelo LMN (LMN-LMS, LMN-LMM, G-LMN e ORG-LMN) para o conjunto de dados *Silverbox* é mostrada na Tabela 11. Como feito anteriormente, dois cenários foram testados: um

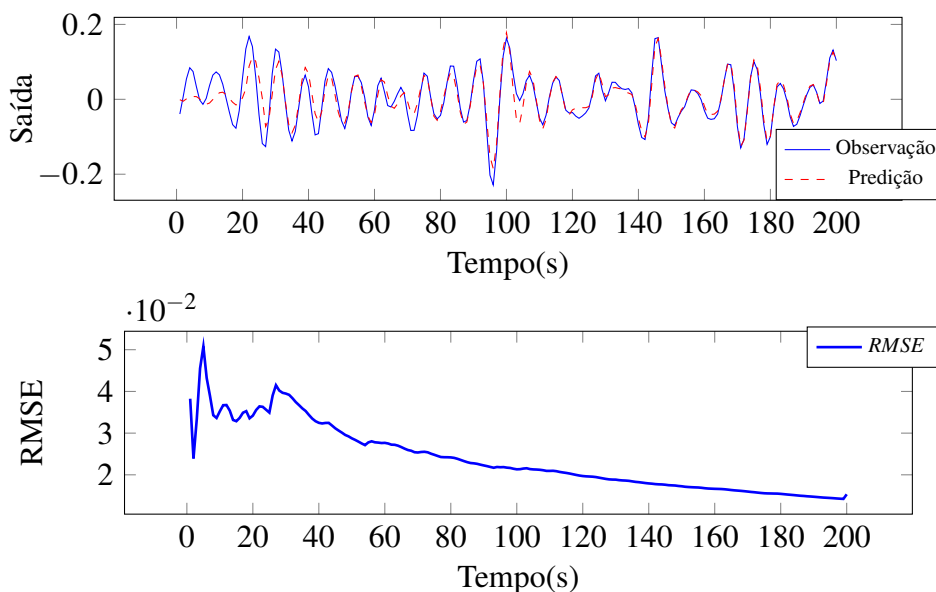


Figura 31 – Desempenho do modelo proposto ORG-LMN durante as primeiras 200 amostras de treinamento para o conjunto de dados *Silverbox*. A figura superior mostra a série temporal de saída estimada, enquanto a figura inferior mostra a evolução correspondente dos valores RMSE.

Tabela 11 – RMSE na fase de teste para o conjunto de dados *Silverbox* para dois níveis de contaminação com *outliers*.

Modelos	0% outliers		15% outliers	
	RMSE	STD	RMSE	STD
LMN-LMS ( $S = 5$ )	3,98E-05	3,62E-07	5,34E-05	2,73E-07
LMN-LMM ( $S = 5$ )	3,24E-05	2,36E-07	3,59E-05	1,13E-07
G-LMN	1,50E-05	0,00E0	3,05 E-05	0,00E0
ORG-LMN	1,02E-05	0,00E0	1,18 E-05	0,00E0

cenário livre de *outliers* e um com 15% de contaminação de *outliers*. Pode-se observar que a deterioração do desempenho com o aumento dos níveis de contaminação é muito menor para o modelo ORG-LMN robusto. Os desempenhos dos modelos propostos também são melhores do que os obtidos por modelos de tamanho fixo (LMN-LMS e LMN-LMM).

Exemplos típicos dos resultados de predição obtidos pelos modelos G-LMN e ORG-LMN sob regime de simulação livre para o conjunto de dados *Silverbox* com 15% de contaminação com *outliers* são mostrados na Figura 32. Para fins de visualização, mostrou-se apenas as primeiras 200 amostras de saída dos dados de validação. Para este conjunto de dados em larga escala, para o qual os modelos foram treinados de forma *online* (ou seja, sem treinamento repetido ao longo de várias épocas), o melhor desempenho do modelo ORG-LMN em comparação com sua versão não robusta é facilmente visto.

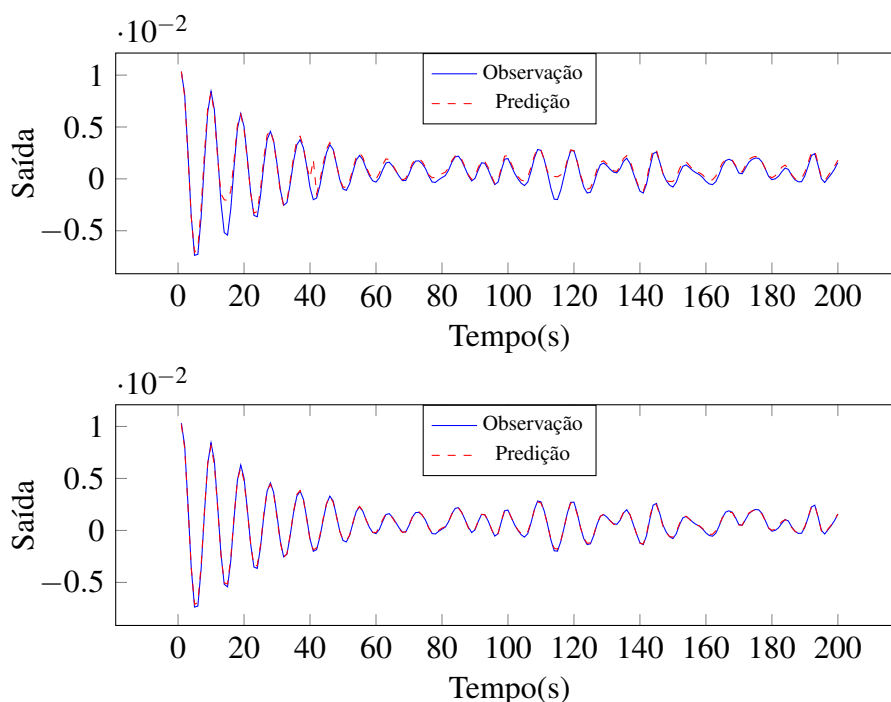


Figura 32 – Séries previstas (conjunto de teste) pelos modelos G-LMN (superior) e ORG-LMN (inferior) em cenários contaminados com 15% de *outliers* (Conjunto de dados *Siverbox*).

### 6.5.1 Avaliação via Teste KS

O teste KS avalia a distância entre as CDFs de duas sequências de resíduos (SONG, 2004). A hipótese nula a ser testada é que as sequências de resíduos provêm da mesma distribuição (BARRETO; SOUZA, 2016). O conjunto final de experimentos para esse conjunto de dados visa avaliar o grau de similaridade, do ponto de vista estatístico, entre a sequência de resíduos gerada pelos modelos G-LMN e ORG-LMN.

A aplicação do teste KS para as sequências de resíduos produzidos pelos dois modelos propostos, modelos G-LMN e ORG-LMN, no cenário livre de *outliers* indica que os modelos são estatisticamente equivalentes. Essa equivalência pode ser inferida qualitativamente pela comparação dos CDFs empíricas dos resíduos produzidos pelos dois modelos no cenário livre de valores discrepantes, conforme mostrado na linha superior da Figura 33. É muito difícil notar qualquer diferença entre os as duas CDFs.

No entanto, para o cenário contaminado por *outlier*, as CDFs empíricas correspondentes dos resíduos são mostradas na linha inferior da Figura 33, a partir do qual se pode observar claramente uma diferença entre as duas CDFs. Por esse resultado, pode-se inferir que os modelos propostos apresentam diferentes desempenhos preditivos para o cenário com contaminação *outlier*.

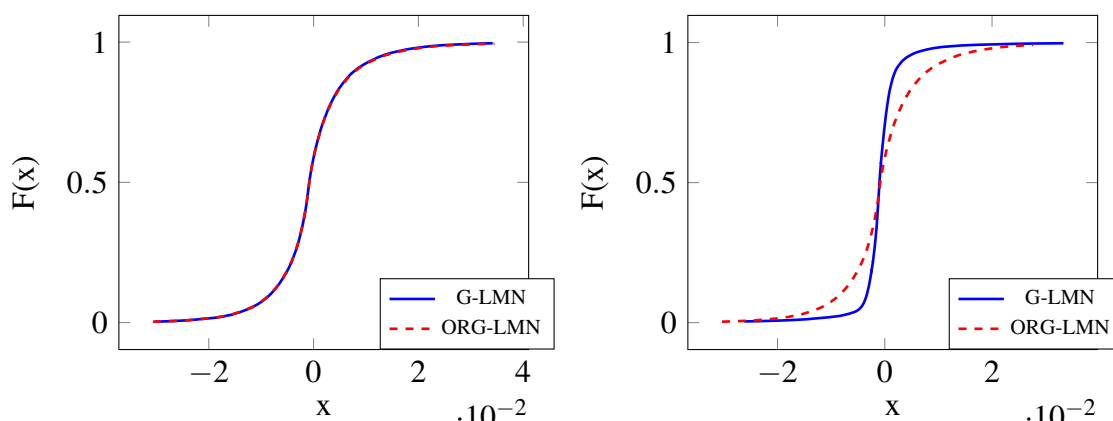


Figura 33 – CDFs empíricas dos resíduos gerados (dados de treinamento) pelo modelo G-LMN (linha sólida em azul) e o modelo ORG-LMN (linha vermelha tracejada) para o conjunto de dados Siverbox. Cenário livre de *outliers* (superior). Cenário contaminado por *outliers* (inferior).

Tabela 12 – Teste KS aplicado em cenários sem presença de *outliers*.

Hipótese Nula		Hipótese Nula	
G-LMN	Teste KS	ORG-LMN	Teste KS
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Aceita
RAN-LMM	Aceita	RAN-LMM	Aceita
ORG-LMN	Aceita	G-LMN	Aceita

Tabela 13 – Teste KS aplicado em cenários contaminados com 15% *outliers*.

Hipótese Nula		Hipótese Nula	
G-LMN	Teste KS	ORG-LMN	Teste KS
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Rejeita
RAN-LMM	Rejeita	RAN-LMM	Rejeita
ORG-LMN	Rejeita	G-LMN	Rejeita

Dessa forma, de acordo com a Tabela 12, notou-se que os modelos G-LMN e RAN (versões originais e robustas) geram resíduos equivalentes. Isso é aceitável pelo fato de estarmos em um cenário sem influência de *outliers*. Vale ratificar também que os modelos RAN e G-LMN possuem semelhanças em suas regras de aprendizagem e crescimento (ou seja, o custo computacional também é semelhante). Com isso, a melhor forma de avaliar os modelos é por meio do desempenho do sinal de erro.

Em cenários contaminados por *outliers* na fase de treinamento, nota-se que o que a maioria dos modelos possui sequências de resíduos estatisticamente não equivalentes, ou seja, não geram a mesma distribuição (ver Tabela 13). Apenas o modelos não robustos RAN-LMS e G-LMN que apresentaram semelhanças entre os resíduos gerado. Mais uma vez, pode-se afirmar

que essa semelhança corrobora a idéia das semelhanças entre os modelos baseados na regra LMS. Porém, quando os modelos são modificados pela regra LMM, a sequência de resíduos não se mostra estatisticamente equivalente.

## 6.6 Resultados no conjunto Secador Industrial (MIMO)

Com a formulação realizada, finalmente, também testou-se as propostas incrementais (versões originais e robustas) em um conjunto de dados *benchmarking* correspondente a um sistema MIMO. Para este propósito, selecionou-se o conjunto de dados secador industrial descrito em Bittanti e Picci (1996) e Chou e Maciejowski (1997) e disponível no repositório *DaISy*.

Todos os modelos incrementais com suas versões originais e robustas foram avaliados nesse conjunto de dados. Esse conjunto conta com 3 entradas ( $n_u = 3$ ): taxa de fluxo da matéria prima ( $u_1$ ), taxa de fluxo de combustível ( $u_2$ ) e velocidade do ventilador de exaustão de gás quente ( $u_3$ ); e 3 saídas ( $n_y = 3$ ): temperatura do bulbo seco ( $y_1$ ), temperatura do bulbo úmido ( $y_2$ ) e o teor de umidade do material de matéria-prima ( $y_3$ ).

Para os testes com esse conjunto de dados, as taxas de aprendizagem inicial e final foram definidas como  $\alpha_0 = 0,5$  e  $\alpha_T = 0,001$  para todos os modelos. Para o modelo LLM Incremental, utilizou-se: a idade máxima para remoção dos links  $a_{max} = 50$ ;  $\varepsilon_b = 0,99$  e  $\varepsilon_n = 0,005$  (Equação 3.2); número máximo de modelos  $N = 10$ ;  $\lambda = 100$ . Para os modelos RAN e GLMN, definiu-se  $\kappa = 0,9$ ,  $\delta_{max} = \max(\|\mathbf{x} - \mathbf{c}_0\|)$ ,  $\delta_{min} = \delta_{max}/5$ ,  $\tau = 5$ , e  $\varepsilon = 10^{-4}$ .

O conjunto de dados Secador Industrial possui valores da entrada  $u_1$  maiores que as demais entradas ( $u_1 \gg u_2$  e  $u_3$ ); por este motivo, optou-se pela normalização dos dados antes mesmo do treinamento para a faixa de valores entre 0 e 1.

Efetuiu-se uma comparação numérica em termos de valores RMSE das seis abordagens incrementais (INC-LLM-LMS, INC-LLM-LMM, RAN-LMS, RAN-LMM, G-LMN e ORG-LMN). Essa comparação é mostrada nas Tabelas 14, 15, 16, 17, 18 e 19. Como feito anteriormente, alguns cenários foram testados: um cenário livre de *outliers*, um com 5% de contaminação, um com 10% de contaminação e um com 15% de contaminação de *outliers*. Pode-se observar que a deterioração do desempenho com o aumento dos níveis de contaminação é muito menor para o modelo ORG-LMN robusto para todas as saídas  $y_1$ ,  $y_2$  e  $y_3$ .

A deterioração do desempenho com o aumento dos níveis de contaminação também pode ser observada na Figura 34. Os valores de RMSE para a saída  $y_1$  são mostrados no gráfico superior esquerdo, para a saída  $y_2$  estão no gráfico superior direito e  $y_3$  estão no gráfico inferior.



Tabela 14 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo INC-LLM-LMS.

INC-LLM-LMS			
Outliers	$y_1$	$y_2$	$y_3$
0%	9,11E-03 ± 0,00E0	6,54E-03 ± 0,00E0	9,65E-03 ± 0,00E0
5%	9,12E-03 ± 0,00E0	6,95E-03 ± 0,00E0	9,99E-03 ± 0,00E0
10%	1,22E-02 ± 0,00E0	7,69E-03 ± 0,00E0	1,36E-02 ± 0,00E0
15%	2,88E-02 ± 0,00E0	9,70E-03 ± 0,00E0	3,07E-02 ± 0,00E0

Tabela 15 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo INC-LLM-LMM.

INC-LLM-LMM			
Outliers	$y_1$	$y_2$	$y_3$
0%	9,10E-03 ± 0,00E0	6,54E-03 ± 0,00E0	9,66E-03 ± 0,00E0
5%	9,11E-03 ± 0,00E0	6,77E-03 ± 0,00E0	9,99E-03 ± 0,00E0
10%	9,90E-03 ± 0,00E0	6,89E-03 ± 0,00E0	1,04E-02 ± 0,00E0
15%	1,16E-02 ± 0,00E0	6,99E-03 ± 0,00E0	1,57E-02 ± 0,00E0

Tabela 16 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo RAN-LMS.

RAN-LMS			
Outliers	$y_1$	$y_2$	$y_3$
0%	1,11E-02 ± 0,00E0	7,45E-03 ± 0,00E0	1,26E-02 ± 0,00E0
5%	1,12E-02 ± 0,00E0	7,58E-03 ± 0,00E0	1,27E-02 ± 0,00E0
10%	2,22E-02 ± 0,00E0	8,22E-03 ± 0,00E0	2,28E-02 ± 0,00E0
15%	3,12E-02 ± 0,00E0	9,30E-03 ± 0,00E0	3,17E-02 ± 0,00E0

Tabela 17 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo RAN-LMM.

RAN-LMM			
Outliers	$y_1$	$y_2$	$y_3$
0%	1,09E-02 ± 0,00E0	7,34E-03 ± 0,00E0	1,26E-02 ± 0,00E0
5%	1,11E-02 ± 0,00E0	7,57E-03 ± 0,00E0	1,26E-02 ± 0,00E0
10%	1,82E-02 ± 0,00E0	7,82E-03 ± 0,00E0	1,31E-02 ± 0,00E0
15%	1,94E-02 ± 0,00E0	7,92E-03 ± 0,00E0	1,50E-02 ± 0,00E0

Tabela 18 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo G-LMN.

G-LMN			
Outliers	$y_1$	$y_2$	$y_3$
0%	2,35E-03 ± 0,00E0	4,43E-03 ± 0,00E0	2,27E-03 ± 0,00E0
5%	2,44E-03 ± 0,00E0	4,49E-03 ± 0,00E0	2,29E-03 ± 0,00E0
10%	4,54E-03 ± 0,00E0	6,40E-03 ± 0,00E0	3,03E-03 ± 0,00E0
15%	9,48E-03 ± 0,00E0	7,30E-03 ± 0,00E0	5,21E-03 ± 0,00E0

Mais uma vez o modelo ORG-LMN obtém melhores valores para o RMSE e se mostra quase insensível a presença de *outliers*, ou seja, a curva do modelo se mostra quase horizontal. O

Tabela 19 – RMSE na fase de teste utilizando conjunto de dados MIMO secador industrial para o modelo ORG-LMN.

ORG-LMN			
Outliers	$y_1$	$y_2$	$y_3$
0%	2,27E-03 ± 0,00E0	4,27E-03 ± 0,00E0	2,26E-03 ± 0,00E0
5%	2,41E-03 ± 0,00E0	4,41E-03 ± 0,00E0	2,31E-03 ± 0,00E0
10%	2,49E-03 ± 0,00E0	4,50E-03 ± 0,00E0	2,39E-03 ± 0,00E0
15%	2,98E-03 ± 0,00E0	4,68E-03 ± 0,00E0	2,59E-03 ± 0,00E0

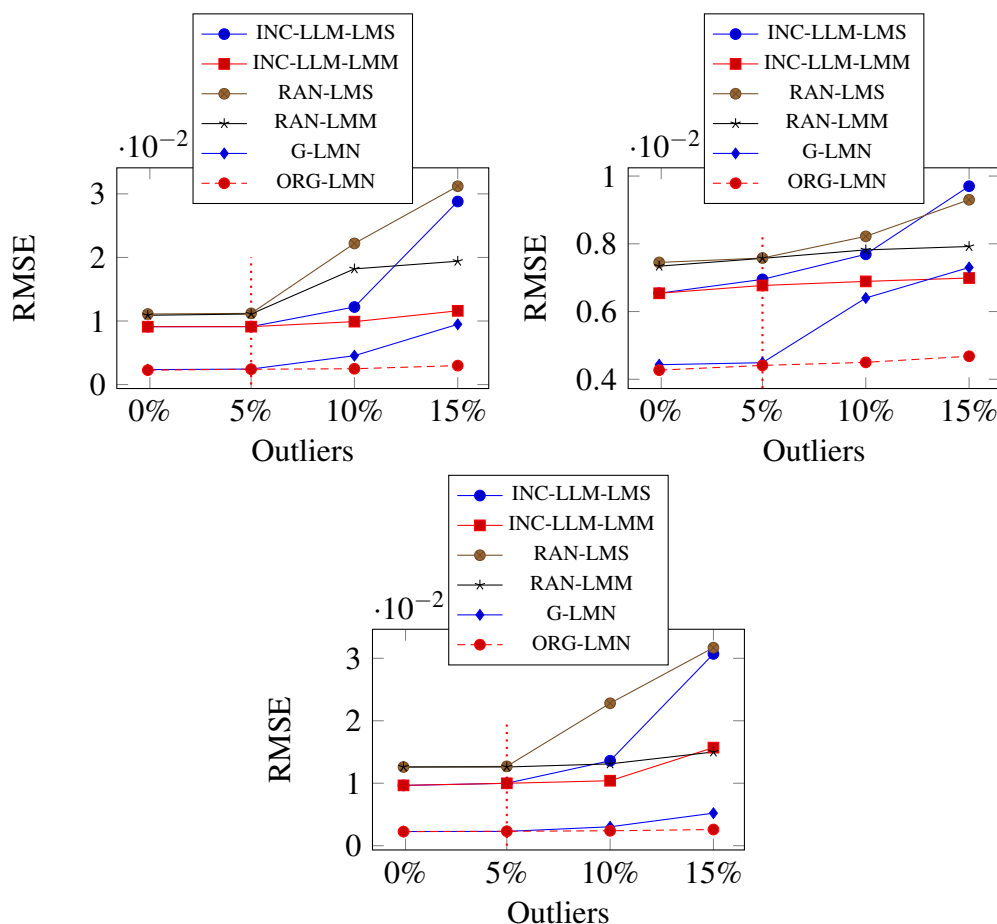


Figura 34 – Valores de RMSE em função da quantidade de *outliers* dos dados de treinamento para a as saídas  $y_1$ ,  $y_2$  e  $y_3$  (Conjunto de dados MIMO secador industrial).

modelo INC-LLM-LMM também merece destaque quanto à sensibilidade quanto a *outliers*.

Na Figura 35, pode-se ver a evolução do número de modelos locais em função das épocas de treinamento para os três modelos locais crescentes e robustos (INC-LLM-LMM, RAN-LMM e ORG-LMN). Os modelos RAN-LMM e ORG-LMN seguem um mesmo padrão de crescimento. Já o modelo INC-LLM-LMM precisa de um modelo local a mais para alcançar taxas de erros semelhantes aos outros dois modelos.

Para o conjunto de dados MIMO secador industrial, testes utilizando a metodologia de predição de  $k$  passos à frente foram realizados. Na Tabela 20, alguns desses resultados na

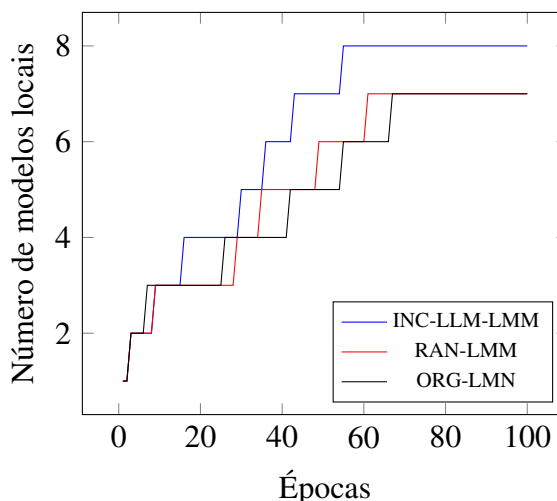


Figura 35 – Evolução do número de modelos locais em função das épocas de treinamento para modelos incrementais robustos (conjunto de dados MIMO secador industrial).

Tabela 20 – Valores de RMSE em função da estratégia adotada para a fase de teste em cenários contaminados com 15% de *outliers* na fase de estimação para o modelo ORG-LMN.

Outliers	ORG-LMN		
	$y^1$	$y^2$	$y^3$
1 passo à frente	1,58E-03 ± 0,00E0	2,17E-03 ± 0,00E0	1,12E-03 ± 0,00E0
2 passos à frente	2,11E-03 ± 0,00E0	2,98E-03 ± 0,00E0	1,94E-03 ± 0,00E0
3 passos à frente	2,59E-03 ± 0,00E0	3,17E-03 ± 0,00E0	2,04E-03 ± 0,00E0
Simulação Livre	2,98E-03 ± 0,00E0	4,68E-03 ± 0,00E0	2,59E-03 ± 0,00E0

fase de teste em cenários contaminados com 15% de *outliers* para o modelo ORG-LMN foram reportados. É possível notar que para o cenário de simulação livre ( $\infty$  passos à frente), a ordem de grandeza dos valores de RMSE continuou a mesma. Dessa forma, apresentou-se, mais uma vez, os resultados dos testes com simulação livre que se aproxima de um cenário *online*.

Por fim, de maneira a ilustrar a predição realizada, as melhores séries preditas ( $y_1$ ,  $y_2$  e  $y_3$ ) pelo modelo ORG-LMN em cenários contaminados com 15% de *outliers* para o conjunto de dados MIMO secador industrial são mostradas na Figura 36.

## 6.7 Considerações finais

Neste capítulo, realizou-se a avaliação dos modelos implementados nesta tese conforme ilustrado na Figura 36. O objetivo dos experimentos feitos foi avaliar a capacidade dos modelos propostos em fornecer um modelo incremental e robusto para a identificação recursiva de sistemas dinâmicos na presença de *outliers*. Para isto, utilizou-se dois conjuntos de dados sintéticos e cinco conjuntos de dados reais, dentre estes, um de larca escala e um conjunto MIMO. A inclusão da regra LMM nos modelos ofereceu robustez no tratamento adequado de amostras

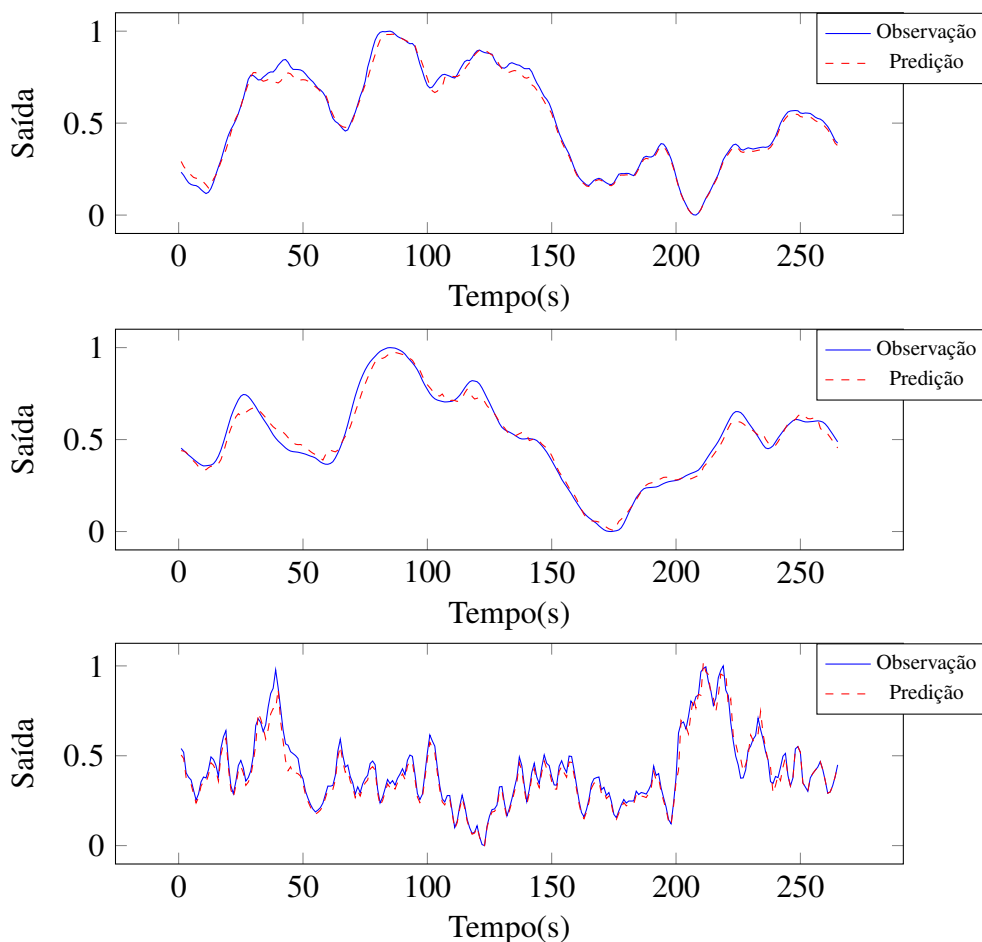


Figura 36 – Melhores séries preditas por simulação livre ( $y_1$ ,  $y_2$  e  $y_3$ ) pelo modelo proposto ORG-LMN em cenários contaminados com 15% de *outliers* (Conjunto MIMO Secador Industrial).

discrepantes de entrada-saída e a estrutura online crescente mostrou-se apta a lidar com a não estacionariedade de processos.

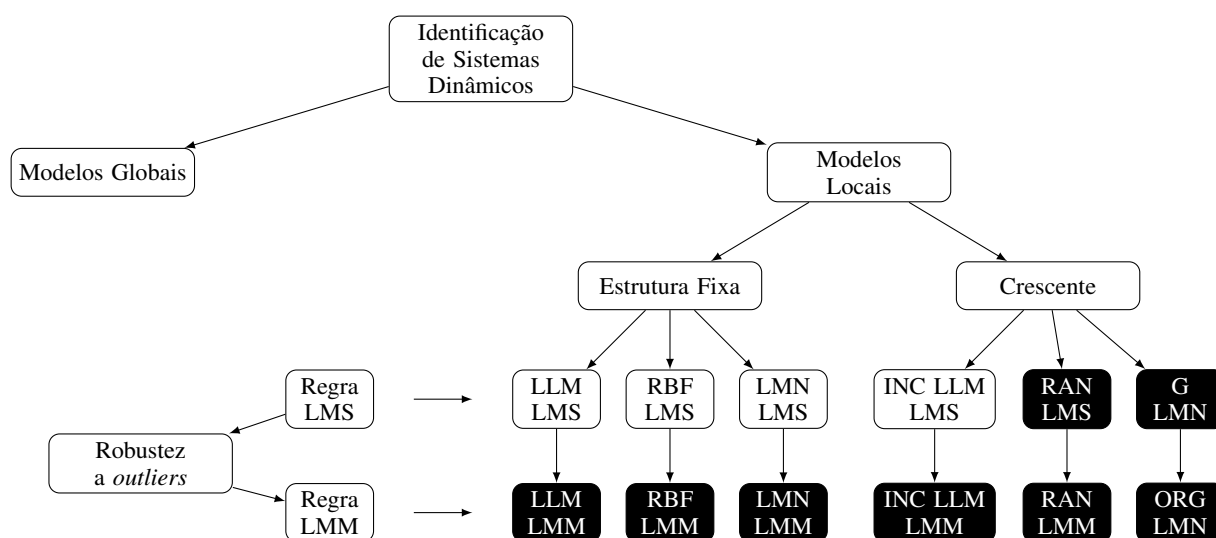


Figura 37 – Modelos implementados nesta tese.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo, uma visão geral das contribuições desta tese e as respectivas conclusões serão apresentadas. Algumas possibilidades de continuação deste trabalho foram apontadas.

### 7.1 Conclusões e Discussão

Nesta tese, abordou-se a tarefa de identificação de sistemas dinâmicos em cenários contaminados com *outliers*. Para este fim, revisitou-se uma classe de modelos de identificação conhecida como modelos locais e a questão da robustez na estimação recursiva de parâmetros (ou seja, aprendizagem) na presença de *outliers*.

A abordagem local é uma alternativa a modelos globais em problemas de identificação de sistemas não lineares. Dessa forma, o espaço consegue ser dividido em partições para as quais construiu-se modelos mais simples utilizando apenas os dados que participam destas regiões. O interessante é que esses modelos se tornam especializados (ou localizados) na dinâmica local do sistema referente a uma dada partição, ou seja, conseguiu-se aproximar uma relação global por meio de vários hiperplanos.

Os modelos locais discutidos e avaliados utilizam a conhecida regra de Widrow-Hoff (WIDROW, 2005), também chamada de regra LMS, na estimação recursiva de seus parâmetros. Um dos objetivos dessa tese foi, em primeiro lugar, avaliar o quanto *outliers* afetam o desempenho preditivo dos modelos locais de aproximação e, em segundo lugar, se a substituição da regra LMS original por uma versão robusta a *outliers*, chamada de regra LMM (ZOU *et al.*, 2000), é capaz de melhorar de alguma forma o desempenho dos modelos locais, proporcionando-lhes maior resiliência a *outliers*.

Todo o desenvolvimento apresentado nesta tese partiu de modelos locais de tamanho fixo, a saber, mapeamento linear local (LLM), rede de função de base radial (RBFN) e rede de modelos locais (LMN). Os resultados de um amplo estudo comparativo com dados de sistemas dinâmicos artificiais e reais mostraram que a simples alteração da regra LMS por sua versão robusta proporcionou uma melhora considerável no desempenho dos modelos locais avaliados em cenários ricos em *outliers*. Este estudo também mostrou uma melhora considerável para todas as versões robustas dos modelos locais avaliados e que a melhora ficava ainda mais evidente quando a proporção de *outliers* crescia.

Outro objetivo desta tese foi avaliar o desempenho de modelos locais robustos

crecentes, ou seja, modelos em que a especificação *a priori* do número de neurônios não fosse necessária, pois neurônios vão sendo alocados com o passar do tempo. Estas novas unidades são inseridas apenas quando exigidas pelo desempenho do modelo. Desenvolveu-se as equações de aprendizagem de dois modelos baseados em LMN. Este primeiro modelo, denominado G-LMN, lida bem com a não-estacionariedade, mas tem um desempenho ruim em termos de RMSE em cenários contaminados por *outliers*. Assim, uma variante robusta a *outliers*, denominada ORG-LMN, foi então desenvolvida introduzindo modificações adequadas em todas as regras de aprendizagem. As modificações envolveram a substituição da regra original do tipo LMS pela variante robusta supracitada (LMM). Esta modificação simples, mas relevante, foi capaz de oferecer mais resiliência aos *outliers* do que o modelo G-LMN original.

## 7.2 Resumo das Contribuições da Tese

Grosso modo, esta tese envolveu uma releitura da aplicação de modelos locais baseados em técnicas de quantização vetorial para identificação recursiva robusta de sistemas dinâmicos não lineares usando para isso o arcabouço da estimação  $M$ .

Esta ampla releitura levou ao desenvolvimento de contribuições particulares, algumas menores outras maiores, de modelos locais robustos a outliers. Dentre estas maiores contribuições, acredita-se que as listadas abaixo são as mais relevantes.

- Robustecer modelos locais de tamanho fixo, tais como aqueles baseados nas redes LLM, RBF e LMN, visando o tratamento adequado de outliers em problemas de identificação recursiva de sistemas SISO. Esta contribuição levou ao desenvolvimento dos modelos LLM-LMM, RBF-LMM, LMN-LMM.
- Desenvolver modelos locais crescentes e robustos a outliers, que possuam estrutura simples e de baixa complexidade computacional e aplicá-los em problemas de identificação recursiva de sistemas SISO. Esta contribuição levou ao desenvolvimento dos modelos RAN-LMM, INC-LLM-LMM e ORG-LMN.
- Desenvolver extensões dos modelos crescentes G-LMN e ORG-LMN voltados para problemas de identificação recursiva de sistemas MIMO.
- Promover uma ampla avaliação dos modelos propostos quanto às suas acurácias preditivas em simulação livre após etapa de estimação com outliers usando para isso conjuntos de dados amplamente usados na literatura, incluindo um conjunto de dados de larga escala (*silverbox*) e um conjunto MIMO.

- Promover uma ampla avaliação dos modelos crescentes propostos quanto a características específicas destes, tal como a evolução da inserção de modelos locais com o passar do tempo e o impacto desta inserção nos valores de RMSE.
- Promover uma validação estatística via teste de Kolmogorov-Smirnov do desempenho dos modelos robustos em relação aos modelos não robustos.

### 7.3 Sobre a continuidade desta pesquisa

Algumas propostas de trabalhos na linha de pesquisa desta tese podem ainda ser desenvolvidas. Dentre os potenciais temas para pesquisa futura, pode-se listar:

- Utilização de algoritmos de otimização de parâmetros para otimizar os parâmetros dos modelos crescentes.
- Utilização de outras metodologias de estatística robusta em substituição dos estimadores- $M$ .
- Avaliação dos modelos propostos na previsão de séries temporais.
- Estudo do modelo LWR (*locally weighted regression*) para estimação robusta.
- Aplicações dos modelos locais propostos em diversas estratégias de sistemas de controle (controle preditivo).
- Investigação dos modelos propostos em tarefas de classificação.
- Avaliação do modelo proposto ORG-LMN como um método de parametrização para a classificação de diferentes estados operacionais de sistemas dinâmicos para fins de detecção de falhas.
- Implementação do modelo RBFN considerando esparsidade.
- Estender a utilização do modelo NARX para o modelo NARMAX.

## REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais**. Editora UFMG, 2004.
- AGUIRRE, L. A. **Introdução à identificação de sistemas**. 4th. ed. Editora UFMG, 2015.
- AGUIRRE, L. A. A bird's eye view of nonlinear system identification. **arXiv preprint arXiv:1907.06803**, 2019.
- ANGELOV, P. P.; FILEV, D. P. An approach to online identification of takagi-sugeno fuzzy models. **IEEE Transactions on Systems, Man, and Cybernetics - part B**, v. 34, n. 1, p. 484–497, 2004.
- AYALA, H. V. H.; HABINEZA, D.; RAKOTONDRABE, M.; COELHO, L. dos S. Nonlinear black-box system identification through coevolutionary algorithms and radial basis function artificial neural networks. **Applied Soft Computing**, Elsevier, v. 87, p. 105990, 2020.
- BARNES, C.; BROWN, S.; FLAKE, G.; JONES, R.; O'ROURKE, M.; LEE, Y. **Applications of neural networks to process control and modeling**. 1991.
- BARRETO, G. A.; ARAÚJO, A. F. R. Identification and control of dynamical systems using the self-organizing map. **IEEE Transactions on Neural Networks**, v. 15, n. 5, p. 1244–1259, 2004.
- BARRETO, G. A.; SOUZA, L. G. M. Novel approaches for parameter estimation of local linear models for dynamical system identification. **Applied Intelligence**, v. 44, n. 1, p. 149–165, 2016.
- BARROS, A. L. B. d. P. **Revisitando o problema de classificação de padrões na presença de outliers usando técnicas de regressão robusta**. Tese (Doutorado) – Universidade Federal do Ceará, 2013.
- BAUMANN, M.; WEISSINGER, C.; HERZOG, H.-G. System identification and modeling of an automotive bidirectional dc/dc converter. In IEEE. **Vehicle Power and Propulsion Conference (VPPC)**. 2019. p. 1–5.
- BELZ, J.; MUNKER, T.; HEINZ, T. O.; KAMPMANN, G.; NELLES, O. Automatic modeling with local model networks for benchmark processes. **IFAC-PapersOnLine**, v. 50, n. 1, p. 470–475, 2017.
- BESSA, J. A.; BARRETO, G. A. Identificação recursiva de sistemas dinâmicos usando modelos locais robustos a outliers. **Anais da Sociedade Brasileira de Automática**, v. 1, n. 1, 2019.
- BESSA, J. A.; BARRETO, G. A. Recursive system identification using outlier-robust local models. **IFAC-PapersOnLine**, v. 52, n. 1, p. 436–441, 2019.
- BILLINGS, S. A. **Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains**. John Wiley & Sons, 2013.
- BITTANTI, S.; PICCI, G. **Identification, adaptation, learning: the science of learning models from data**. Springer Science & Business Media, 1996. v. 153.
- BLAZIC, S.; SKRJANC, I. Incremental Fuzzy C-Regression Clustering from Streaming Data for Local-Model-Network Identification. **IEEE Transactions on Fuzzy Systems**, v. 28, n. 4, p. 758–767, 2020. ISSN 19410034.



BLAZQUEZ-GARCÍA, A.; CONDE, A.; MORI, U.; LOZANO, J. A. A review on outlier/anomaly detection in time series data. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 54, n. 3, abr. 2021. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/34444690>.

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. **Time series analysis: forecasting and control**. John Wiley & Sons, 2015.

BRUNTON, S. L.; KUTZ, J. N. **Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control**. Cambridge Press, 2019.

CHAN, S.-C.; ZHOU, Y. On the performance analysis of the least mean M-estimate and normalized least mean M-estimate algorithms with gaussian inputs and additive gaussian and contaminated Gaussian noises. **Journal of Signal Processing Systems**, v. 60, n. 1, p. 81–103, 2010.

CHAN, S.-C.; ZOU, Y.-X. A recursive least m-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis. **IEEE Transactions on Signal Processing**, IEEE, v. 52, n. 4, p. 975–991, 2004.

CHEN, S.; BILLINGS, S. A.; COWAN, C. F. N.; GRANT, P. M. Non-linear systems identification using radial basis functions. **International Journal of Systems Science**, v. 21, n. 12, p. 2513–2539, 1990.

CHO, K. B.; WANG, B. H. Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction. **Fuzzy sets and systems**, Elsevier, v. 83, n. 3, p. 325–339, 1996.

CHOU, C. T.; MACIEJOWSKI, J. M. System identification using balanced parametrizations. **IEEE transactions on automatic control**, v. 42, n. 7, p. 956–974, 1997.

CHUANG, C.-C. Fuzzy weighted support vector regression with a fuzzy partition. **IEEE Transactions on Systems, Man, and Cybernetics - Part B**, v. 37, n. 3, p. 630–640, 2007.

COSTA, T. V.; FILETI, A. M. F.; OLIVEIRA-LOPES, L. C.; SILVA, F. V. Experimental assessment and design of multiple model predictive control based on local model networks for industrial processes. **Evolving Systems**, v. 6, n. 4, p. 243–253, 2015.

DASH, C. S. K.; BEHERA, A. K.; DEHURI, S.; CHO, S.-B. Radial basis function neural networks: a topical state-of-the-art survey. **Open Computer Science**, De Gruyter, v. 1, 2016.

DURAN-HERNANDEZ, C.; PEREZ-SANTIAGO, R.; ETCHEVERRY, G.; LEDESMA-ALONSO, R. Modeling of a simplified 2d cardiac valve by means of system identification. In SPRINGER. **Mexican Conference on Pattern Recognition**. 2019. p. 371–380.

FELDMAN, M.; BRAUN, S. Nonlinear vibrating system identification via hilbert decomposition. **Mechanical Systems and Signal Processing**, Elsevier, v. 84, p. 65–96, 2017.

FERREIRA, P. H.; ARAÚJO, A. F. Growing self-organizing maps for nonlinear time-varying function approximation. **Neural Processing Letters**, Springer, p. 1–26, 2020.

FONTINELE, H. I.; MELO, D. B.; BARRETO, G. A. Local models for learning inverse kinematics of redundant robots: A performance comparison. In **Advances in Self-Organizing Maps and Learning Vector Quantization**. Springer, 2016. p. 177–187.

FOSS, B. A.; JOHANSEN, T. A. On local and fuzzy modelling. In **Third International Conference on Industrial Fuzzy Control and Intelligent Systems (IFIS'93)**. 1993. p. 80–87.

FRITZKE, B. Fast learning with incremental RBF networks. **Neural Processing Letters**, v. 1, n. 1, p. 2–5, 1994.

FRITZKE, B. A growing neural gas network learns topologies. In **Advances in Neural Information Processing Systems 7**. MIT Press, 1995. p. 625–632.

FRITZKE, B. Incremental learning of local linear mappings. In **Proceedings of the International Conference on Artificial Neural Networks (ICANN'95)**. 1995. p. 217–222.

FUANGKHON, P. An incremental learning preprocessor for feed-forward neural network. **Artificial Intelligence Review**, Springer, v. 41, n. 2, p. 183–210, 2014.

GAO, R. **Local Model Network Application in Control**. Tese (Doutorado) – Dublin Institute of Technology, 2004.

GIANNAKIS, G. B.; SERPEDIN, E. A bibliography on nonlinear system identification. **Signal Processing**, Elsevier, v. 81, n. 3, p. 533–580, 2001.

HAMETNER, C.; JAKUBEK, S. Local model network identification for online engine modelling. **Information Sciences**, Elsevier, v. 220, p. 210–225, 2013.

HAMPEL, F. R.; RONCHETTI, E. M.; ROUSSEEUW, P. J.; STAHEL, W. A. **Robust statistics: the approach based on influence functions**. John Wiley & Sons, 2011. v. 114.

HAN, H.; CHEN, Q.; QIAO, J. Research on an online self-organizing radial basis function neural network. **Neural Computing and Applications**, v. 19, p. 667–676, 2010.

HAWHEEL, T. I.; CLARKSON, P. M. A class of order statistic LMS algorithms. **IEEE Transactions on Signal Processing**, v. 40, n. 1, p. 44–53, 1992.

HAYKIN, S. S.; WIDROW, B.; WIDROW, B. **Least-mean-square adaptive filters**. Wiley Online Library, 2003. v. 31.

HE, F.; YANG, Y. Nonlinear system identification of neural systems from neurophysiological signals. **bioRxiv**, Cold Spring Harbor Laboratory, 2020.

HE, H.; CHEN, S.; LI, K.; XU, X. Incremental learning from stream data. **IEEE Transactions on Neural Networks**, IEEE, v. 22, n. 12, p. 1901–1914, 2011.

HUBER, P. Robust estimation of a location parameter. **The Annals of Mathematical Statistics**, v. 35, n. 1, p. 73–101, 1964.

JACOBS, R. A.; JORDAN, M. I.; NOWLAN, S. J.; HINTON, G. E. Adaptive mixtures of local experts. **Neural computation**, MIT Press, v. 3, n. 1, p. 79–87, 1991.

JAIN, S.; BANDYOPADHYAY, S. Resource allocation network for segregated targeting problems with dedicated sources. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 56, n. 46, p. 13831–13843, 2017.

JAMSHIDI, M. B.; ALIBEIGI, N. Neuro-fuzzy system identification for remaining useful life of electrolytic capacitors. In **IEEE. 2nd International Conference on System Reliability and Safety (ICSRS)**. 2017. p. 227–231.

JOHANSEN, T. A.; FOSS, B. A. A NARMAX model representation for adaptive control based on local models. **Modeling, Identification and Control**, v. 13, n. 1, p. 25, 1992.

JONES, R. D.; LEE, Y.; BARNES, C.; FLAKE, G.; LEE, K.; LEWIS, P.; QIAN, S. Function approximation and time series prediction with neural networks. In **Proceedings of the 1990 IJCNN International Joint Conference on Neural Networks (IJCNN'1990)**. 1990. p. 649–665.

KALHOR, A.; ARAABI, B. N.; LUCAS, C. Evolving Takagi-Sugeno fuzzy model based on switching to neighboring models. **Applied Soft Computing Journal**, Elsevier B.V., v. 13, n. 2, p. 939–946, 2013.

KARNOPP, D. C.; MARGOLIS, D. L.; ROSENBERG, R. C. **System dynamics: modeling, simulation, and control of mechatronic systems**. John Wiley & Sons, 2012.

KERSCHEN, G.; WORDEN, K.; VAKAKIS, A. F.; GOLINVAL, J.-C. Past, present and future of nonlinear system identification in structural dynamics. **Mechanical systems and signal processing**, Elsevier, v. 20, n. 3, p. 505–592, 2006.

KIANI, M. On GNSS residual position time series prediction and analysis using radial basis function networks machine learning. **Geology**, 2020.

KIUMARSI, B.; LEWIS, F. L.; S.LEVINE, D. Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure. **Neurocomputing**, v. 156, p. 157–165, 2015.

KOHONEN, T. Self-organized formation of topologically correct feature maps. **Biological cybernetics**, Springer, v. 43, n. 1, p. 59–69, 1982.

KÖNIG, O.; HAMETNER, C.; PROCHART, G.; JAKUBEK, S. Battery emulation for power-HIL using local model networks and robust impedance control. **IEEE Transactions on Industrial Electronics**, IEEE, v. 61, n. 2, p. 943–955, 2014.

KOVACEVIC, B.; BANJAC, Z.; KOVACEVIC, I. K. Robust adaptive filtering using recursive weighted least squares with combined scale and variable forgetting factors. **EURASIP Journal on Advances in Signal Processing**, v. 2016, n. 37, p. 1–22, 2016.

LAWRENCE, S.; TSOI, A. C.; BACK, A. D. Function approximation with neural networks and local methods: Bias, variance and smoothness. In AUSTRALIAN NATIONAL UNIVERSITY. **Australian Conference on Neural Networks**. 1996. v. 1621.

LI, X.; YU, W. Dynamic system identification via recurrent multilayer perceptrons. **Information sciences**, Elsevier, v. 147, n. 1-4, p. 45–63, 2002.

LITTMANN, E.; RITTER, H. Cascade LLM networks. In CITESEER. **Proc. Int. Conf. Artificial Neural Networks**. 1992. v. 1, p. 253–257.

LIU, J.; DJURDJANOVIC, D.; MARKO, K.; NI, J. Growing structure multiple model systems for anomaly detection and fault diagnosis. **Journal of Dynamic Systems, Measurement, and Control**, v. 131, n. 5, p. 1–13, 2009.

LJUNG, L. System identification. In **Signal analysis and prediction**. Springer, 1998. p. 163–173.

- LJUNG, L. **System Identification: Theory for the User**. 2nd. ed. Prentice Hall, 1999.
- LU, T.; PAN, L.; JIANGS, J.; ZHANG, Y.; XIONG, Z. Dlm1: Deep linear mappings learning for face super-resolution with nonlocal-patch. In IEEE. **2017 IEEE International Conference on Multimedia and Expo (ICME)**. 2017. p. 1362–1367.
- LV, C.; ZHANG, T.; MA, F.; YUE, D. A very short-term online forecasting model for photovoltaic power based on two-stage resource allocation network. In IEEE. **2018 International Joint Conference on Neural Networks (IJCNN)**. 2018. p. 1–6.
- MAHANAND, B. S.; SURESH, S.; SUNDARARAJAN, N.; KUMAR, M. A. Identification of brain regions responsible for alzheimer’s disease using a self-adaptive resource allocation network. **Neural Networks**, Elsevier, v. 32, p. 313–322, 2012.
- MAIER, C. C.; SCHIRRER, A.; KOZEK, M. Real-time capable nonlinear pantograph models using local model networks in state-space configuration. **Mechatronics**, Elsevier, v. 50, p. 292–302, 2018.
- Mandic, D. P.; Kanna, S.; Constantinides, A. G. On the intrinsic relationship between the least mean square and kalman filters. **IEEE Signal Processing Magazine**, v. 32, n. 6, p. 117–122, 2015.
- MARTINETZ, T. Competitive hebbian learning rule forms perfectly topology preserving maps. In **Proceedings of the 4th International Conference on Artificial Neural Networks (ICANN’93)**. Springer, 1993. p. 427–434.
- MARTINETZ, T. M.; BERKOVICH, S. G.; SCHULTEN, K. J. Neural-gas network for vector quantization and its application to time-series prediction. **IEEE transactions on neural networks**, v. 4, n. 4, p. 558–569, 1993.
- MATTOS, C. L. C. **Recurrent gaussian processes and robust dynamical modeling**. Tese (Doutorado) – Universidade Federal do Ceará, 2017.
- MATTOS, C. L. C.; DAI, Z.; DAMIANOU, A.; BARRETO, G. A.; LAWRENCE, N. D. Deep recurrent gaussian processes for outlier-robust system identification. **Journal of Process Control**, v. 60, p. 82–94, 2017.
- MIYAJIMA, H.; MIYAJIMA, H.; SHIRATORI, N. Proposal of fast and secure clustering method for iot. In **International Multi Conference of Engineers and Computer Scientists**. 2019.
- MIYAJIMA, H.; SHIGEI, N.; MIYAJIMA, H. Fuzzy modeling using vector quantization and local linear mapping. **IAENG International Journal of Computer Science**, v. 47, n. 4, 2020.
- MONTEIRO, L. H. A. **Sistemas Dinâmicos, 2ª**. Ed. Livraria da Física. São Paulo, 2006.
- MOODY, J.; DARKEN, C. J. Fast learning in networks of locally-tuned processing units. **Neural computation**, MIT Press, v. 1, n. 2, p. 281–294, 1989.
- MOSHOU, D.; KETEIAERE, B. D.; VRINDTS, E.; KENNES, P.; BAERDEMAEKER, J. D.; RAMON, H. Local linear mapping neural networks for pattern recognition of plants. **IFAC Proceedings Volumes**, v. 31, n. 12, p. 61–66, 1998.

- MOSHOU, D.; RAMON, H. Extended self-organizing maps with local linear mappings for function approximation and system identification. In **Proceedings of the 1st Workshop on Self-Organizing Maps (WSOM'97)**. 1997. p. 1–6.
- MÜNKER, T.; HEINZ, T. O.; NELLES, O. Hierarchical model predictive control for local model networks. In **Proceedings of the American Control Conference (ACC'2017)**. 2017. p. 5026–5031.
- MUSAVI, M. T.; AHMED, W.; CHAN, K. H.; FARIS, K. B.; HUMMELS, D. M. On the training of radial basis function classifiers. **Neural networks**, Elsevier, v. 5, n. 4, p. 595–603, 1992.
- NAING, N. N.; KHAING, Z. M.; NAING, Z. M. Development of methodology for system identification of non-linear system using radial basis function neural network. In **IEEE. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)**. 2020. p. 2390–2394.
- NARENDRA, K.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **IEEE Transactions on Neural Networks**, v. 1, n. 1, p. 4–27, 1990.
- NAUNG, Y.; SCHAGIN, A.; OO, H. L.; YE, K. Z.; KHAING, Z. M. Implementation of data driven control system of dc motor by using system identification process. In **IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)**. 2018. p. 1801–1804.
- NELLES, O. Local linear model trees for on-line identification of time-variant nonlinear dynamic systems. In **International Conference on Artificial Neural Networks**. Springer, 1996. p. 115–120.
- NELLES, O. **Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes**. Springer Nature, 2020.
- NELLES, O.; ISERMANN, R. Basis function networks for interpolation of local linear models. In **IEEE. Proceedings of 35th IEEE Conference on Decision and Control**. 1996. v. 1, p. 470–475.
- NOWAK, R. D. Nonlinear system identification. **Circuits, Systems and Signal Processing**, Springer, v. 21, n. 1, p. 109–122, 2002.
- OKAMOTO, K.; OZAWA, S.; ABE, S. A Fast Incremental Learning Algorithm of RBF Networks with Long-Term Memory. **Proceedings of the International Joint Conference on Neural Networks**, v. 1, n. June, p. 102–107, 2003.
- OLIVEIRA, P. D. de; BRAGA, A. P. d. S.; REIS, L. L. dos; NOGUEIRA, F. G.; JÚNIOR, A. B. d. S. System identification through RBF neural networks: Improving accuracy by a numerical approximation method for the centroids and widths adjustment. In **XIII Brazilian Congress on Computational Intelligence (CBIC 2017)**. Rio de Janeiro, Brasil , 2017.
- OYANG, Y.-J.; HWANG, S.-C.; OU, Y.-Y.; CHEN, C.-Y.; CHEN, Z.-W. Data classification with radial basis function networks based on a novel kernel density estimation algorithm. **IEEE transactions on neural networks**, IEEE, v. 16, n. 1, p. 225–236, 2005.

- PAI, P. S.; NAGABHUSHANA, T.; RAO, P. R. Tool wear estimation using resource allocation network. **International Journal of Machine Tools and Manufacture**, Elsevier, v. 41, n. 5, p. 673–685, 2001.
- PANG, S.; JIANG, J.; LU, Z.; LI, X.; YANG, W.; HUANG, M.; ZHANG, Y.; FENG, Y.; HUANG, W.; FENG, Q. Hippocampus segmentation based on local linear mapping. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 1–11, 2017.
- PANG, S.; LU, Z.; JIANG, J.; ZHAO, L.; LIN, L.; LI, X.; LIAN, T.; HUANG, M.; YANG, W.; FENG, Q. Hippocampus segmentation based on iterative local linear mapping with representative and local structure-preserved feature embedding. **IEEE transactions on medical imaging**, v. 38, n. 10, p. 2271–2280, 2019.
- PEARSON, R. K. Outliers in process modeling and identification. **IEEE Transactions on control systems technology**, v. 10, n. 1, p. 55–63, 2002.
- PIRES, A. M.; BRANCO, J. A.; ESTATÍSTICA, C. A. da Sociedade Portuguesa de. **Introdução aos métodos estatísticos robustos**. 2007.
- PLATT, J. A resource-allocating network for function interpolation. **Neural Computation**, v. 3, n. 2, p. 213–225, 1991.
- PRASAD, G.; SWIDENBANK, E.; HOGG, B. W. A local model networks based multivariable long-range predictive control strategy for thermal power plants. **Automatica**, v. 34, n. 10, p. 1185–1204, 1998.
- PUTRI, A. N. Principal component analysis for system identification of boiler in mini batch distillation column. 2019.
- R, H. J. R.; T, A. A. V. Training radial basis function networks for wind speed prediction using pso enhanced differential search optimizer. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 5, p. e0196871, 2018.
- RITTER, H.; MARTINETZ, T.; SCHULTEN, K.; BARSKY, D.; TESCH, M.; KATES, R. **Neural computation and self-organizing maps: an introduction**. Addison-Wesley Reading, MA, 1992.
- RONG, H. J.; SUNDARARAJAN, N.; HUANG, G. B.; SARATCHANDRAN, P. Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. **Fuzzy Sets and Systems**, v. 157, n. 9, p. 1260–1275, 2006.
- ROUSSEEUW, P. J.; LEROY, A. M. **Robust regression and outlier detection**. John wiley & sons, 2005. v. 589.
- SANTOS, J. D. d. A. **Adaptive Kernel-based regression for robust system Identification**. Tese (Doutorado) – Universidade Federal do Ceará, 2017.
- SCHAAL, S.; ATKESON, C. G. Constructive incremental learning from only local information. **Neural computation**, MIT Press, v. 10, n. 8, p. 2047–2084, 1998.
- SCHOUKENS, J.; NEMETH, J. G.; CRAMA, P.; ROLAIN, Y.; PINTELON, R. Fast approximate identification of nonlinear systems. **Automatica**, v. 39, n. 7, p. 1267–1274, 2003.

SHAO, C. Data classification by quantum radial-basis-function networks. **Physical Review A**, APS, v. 102, n. 4, p. 042418, 2020.

SJÖBERG, J.; ZHANG, Q.; LJUNG, L.; BENVENISTE, A.; DELYON, B.; GLORENNEC, P.-Y.; HJALMARSSON, H.; JUDITSKY, A. Nonlinear black-box modeling in system identification: a unified overview. **Automatica**, v. 31, n. 12, p. 1691–1724, 1995.

SOONG, T. T. **Fundamentals of probability and statistics for engineers**. John Wiley & Sons, 2004.

SOUZA, L. G. M. **Modelos lineares locais para identificação de sistemas dinâmicos usando redes neurais competitivas**. Tese (Doutorado) – Universidade Federal do Ceará, 2012.

STOKBRO, K.; UMBERGER, D.; HERTZ, J. A. Exploiting neurons with localized receptive fields to learn chaos. **Complex Systems**, v. 4, n. 3, p. 603–622, 1990.

STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, v. 11, n. 4, p. 341–359, 1997.

SURESH, S.; DONG, K.; KIM, H. J. A sequential learning algorithm for self-adaptive resource allocation network classifier. **Neurocomputing**, Elsevier, v. 73, n. 16–18, p. 3012–3019, 2010.

TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modelling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 15, n. 1, p. 116–132, 1985.

TESHNEHLAB, M.; DELI, M. Aliyari-shore *et al.* Fault detection and identification of high dimension system by GLOLIMOT. **International Journal of Industrial Electronics, Control and Optimization**, University of Sistan and Baluchestan, v. 2, n. 4, p. 331–342, 2019.

TESLIC, L.; HARTMANN, B.; NELLES, O.; SKRJANC, I. Nonlinear system identification by gustafson–kessel fuzzy clustering and supervised local model network learning for the drug absorption spectra process. **IEEE transactions on neural networks**, v. 22, n. 12, p. 1941–1951, 2011.

TU, C.-S.; WU, D.-Y.; LEE, S.-J.; WU, C.-H. Regression estimation by radial basis function networks with self-constructing clustering. In IEEE. **2018 5th International Conference on Systems and Informatics (ICSAI)**. 2018. p. 327–332.

VACHKOV, G. Growing model algorithm for process identification based on neural-gas learning and local linear mapping. In IEEE. **Fourth International Conference on Hybrid Intelligent Systems (HIS'04)**. 2004. p. 222–227.

WALTER, J.; RITTER, H.; SCHULTEN, K. Nonlinear prediction with self-organizing maps. In **Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'90)**. 1990. p. 589–594.

WANG, Q.; WANG, X.; SUN, B. Resource allocation network and PID control based on automated guided vehicles. **ACADEMIC JOURNAL OF MANUFACTURING ENGINEERING**, v. 17, n. 2, 2019.

WEINSTEIN, R.; HUBBARD, J. E. Global aerodynamic modeling using automated local model networks in real time. In **AIAA Scitech 2020 Forum**. 2020. p. 0762.

- WIDROW, B. Thinking about thinking: The discovery of the LMS algorithm. **IEEE Signal Processing Magazine**, v. 22, n. 1, p. 100–106, 2005.
- WIDROW, B.; LEHR, M. A. 30 years of adaptive neural networks: Perceptron, madaline and backpropagation. **Proceedings of the IEEE**, v. 78, n. 9, p. 1415–1442, 1990.
- WU, Y.; WANG, H.; ZHANG, B.; DU, K.-L. Using radial basis function networks for function approximation and classification. **International Scholarly Research Notices**, Hindawi, v. 2012, 2012.
- XUE, X.; LÜ, J.; XIANG, W. Nonlinear system identification with modified differential evolution and RBF networks. In **IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)**. 2012. p. 332–335.
- YE, X.; LOH, N. Dynamic system identification using recurrent radial basis function network. In **IEEE 1993 American Control Conference**. 1993. p. 2912–2916.
- YU, C.; WANG, Q.-G.; ZHANG, D.; WANG, L.; HUANG, J. System identification in presence of outliers. **IEEE transactions on cybernetics**, IEEE, v. 46, n. 5, p. 1202–1216, 2015.
- ZEMOURI, R.; RACOCEANU, D.; ZERHOUNI, N. Recurrent radial basis function network for time-series prediction. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 16, n. 5-6, p. 453–463, 2003.
- ZHANG, W.; ZHANG, F.; ZHANG, J.; ZHANG, J.; ZHANG, J. Research and analysis on the identification model of multivariate economic system. In IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. 2018. v. 452, n. 2, p. 022061.
- ZHAO, Z.-L.; QIU, Z.-C.; ZHANG, X.-M.; HAN, J.-D. Vibration control of a pneumatic driven piezoelectric flexible manipulator using self-organizing map based multiple models. **Mechanical Systems and Signal Processing**, v. 70, p. 345–372, 2016.
- ZHU, H.; ZHAO, G.; SUN, L.; LEE, K. Y. Nonlinear predictive control for a boiler–turbine unit based on a local model network and immune genetic algorithm. **Sustainability**, Multidisciplinary Digital Publishing Institute, v. 11, n. 18, p. 5102, 2019.
- ZOU, Y.; CHAN, S.-C.; NG, T.-S. Least mean M-estimate algorithms for robust adaptive filtering in impulse noise. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, v. 47, n. 12, p. 1564–1569, 2000.
- ZOU, Y.; CHAN, S. C.; NG, T. S. Robust M-estimate adaptive filtering. **IEEE Proceedings - Vision, Image and Signal Processing**, v. 148, n. 4, p. 289–294, 2001.



## APÊNDICE A – COMPARAÇÃO ENTRE ALGORITMOS DE QUANTIZAÇÃO VETORIAL PARA O ESPAÇO DE ENTRADA $\chi$

A razão para a aplicação do teste KS no âmbito desta tese é avaliar, por exemplo, se os modelos implementados utilizando diferentes algoritmos de quantização vetorial geram sequências de resíduos estatisticamente equivalentes ou não. Se for equivalentes, então os desempenhos dos modelos locais são independentes dos algoritmos de quantização vetorial utilizados.

Tabela 21 – Teste KS aplicado utilizando o modelo LLM em cenários sem a presença de *outliers* utilizando o conjunto de Dados Sintéticos 1.

Modelos	Hipótese Nula			
	LLM-WTA	LLM-SOM	LLM-FSCL	LLM-NG
<b>LLM-WTA</b>	-	Aceita	Aceita	Aceita
<b>LLM-SOM</b>	Aceita	-	Aceita	Aceita
<b>LLM-FSCL</b>	Aceita	Aceita	-	Aceita
<b>LLM-NG</b>	Aceita	Aceita	Aceita	-

Analisando os resultados mostrados na Tabela 21, constata-se que o modelo LLM treinado via algoritmo SOM é estatisticamente equivalente a todos os modelos obtidos utilizando outros algoritmos de quantização vetorial. Dessa forma, podemos escolher o algoritmo de quantização vetorial que apresentou melhor desempenho conforme Tabela 22

Tabela 22 – Valores de RMSE para a fase de teste utilizando o modelo LLM em cenários sem a presença de *outliers* utilizando o conjunto de Dados Sintéticos 1.

Modelos	RMSE
<b>LLM-WTA</b>	4.62E-03
<b>LLM-SOM</b>	<b>4.50E-03</b>
<b>LLM-FSCL</b>	4.54E-03
<b>LLM-NG</b>	4.57E-03

## APÊNDICE B – TESTES DE KOLMOGOROV-SMIRNOV

A utilização do teste de Kolmogorov-Smirnov (KS) justifica-se pela necessidade de avaliar a diferença entre os desempenhos de dois modelos. Se dois modelos supostamente diferentes geram sequências estatisticamente equivalentes de resíduos (de acordo com o teste KS), então os dois modelos devem ser considerados equivalentes um ao outro. Caso contrário, os modelos não são considerados equivalentes, dessa forma, há uma necessidade de avaliação de desempenho em separado. Aqui, os resultados possíveis que foram obtidos para o Teste KS (cruzamento entre os modelos propostos e demais modelos) utilizando todos os conjuntos de dados reais SISO.

Tabela 23 – Teste KS aplicado em cenários sem presença de *outliers* (Atuador Hidráulico).

Hipótese Nula		Hipótese Nula	
G-LMN	Teste KS	ORG-LMN	Teste KS
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Aceita
RAN-LMM	Aceita	RAN-LMM	Aceita
ORG-LMN	Aceita	G-LMN	Aceita

Tabela 24 – Teste KS aplicado em cenários contaminados com 15% *outliers* (Atuador Hidráulico).

Hipótese Nula		Hipótese Nula	
G-LMN	Teste KS	ORG-LMN	Teste KS
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Rejeita
RAN-LMM	Rejeita	RAN-LMM	Rejeita
ORG-LMN	Rejeita	G-LMN	Rejeita

Tabela 25 – Teste KS aplicado em cenários sem presença de *outliers* (Secador de Cabelo).

Hipótese Nula		Hipótese Nula	
G-LMN	Teste KS	ORG-LMN	Teste KS
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Rejeita	RAN-LMS	Rejeita
RAN-LMM	Rejeita	RAN-LMM	Rejeita
ORG-LMN	Aceita	G-LMN	Aceita

Tabela 26 – Teste KS aplicado em cenários contaminados com 15% *outliers* (Secador de Cabelo).

<b>Hipótese Nula</b>		<b>Hipótese Nula</b>	
<b>G-LMN</b>	<b>Teste KS</b>	<b>ORG-LMN</b>	<b>Teste KS</b>
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Rejeita	RAN-LMS	Rejeita
RAN-LMM	Rejeita	RAN-LMM	Rejeita
ORG-LMN	Rejeita	G-LMN	Rejeita

Tabela 27 – Teste KS aplicado em cenários sem presença de *outliers* (pH).

<b>Hipótese Nula</b>		<b>Hipótese Nula</b>	
<b>G-LMN</b>	<b>Teste KS</b>	<b>ORG-LMN</b>	<b>Teste KS</b>
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Aceita
RAN-LMM	Aceita	RAN-LMM	Aceita
ORG-LMN	Aceita	G-LMN	Aceita

Tabela 28 – Teste KS aplicado em cenários contaminados com 15% *outliers* (pH).

<b>Hipótese Nula</b>		<b>Hipótese Nula</b>	
<b>G-LMN</b>	<b>Teste KS</b>	<b>ORG-LMN</b>	<b>Teste KS</b>
INC-LLM-LMS	Rejeita	INC-LLM-LMS	Rejeita
INC-LLM-LMM	Rejeita	INC-LLM-LMM	Rejeita
RAN-LMS	Aceita	RAN-LMS	Rejeita
RAN-LMM	Rejeita	RAN-LMM	Rejeita
ORG-LMN	Rejeita	G-LMN	Rejeita