



Pós-Graduação em Ciência da Computação

**Paulo Henrique Muniz Ferreira**

**Controles Baseados em Mapas Auto-organizáveis para Sistemas Não-lineares  
Variantes no Tempo**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
<http://cin.ufpe.br/~posgraduacao>

Recife  
2020

**Paulo Henrique Muniz Ferreira**

**Controles Baseados em Mapas Auto-organizáveis para Sistemas Não-lineares  
Variantes no Tempo**

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

**Área de Concentração:** inteligência computacional

**Orientador:** Aluizio Fausto Ribeiro Araújo

Recife  
2020

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

F383c Ferreira, Paulo Henrique Muniz  
Controles baseados em mapas auto-organizáveis para sistemas não-lineares variantes no tempo / Paulo Henrique Muniz Ferreira. – 2020.  
134 f.: il., fig., tab.

Orientador: Aluizio Fausto Ribeiro Araújo.  
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.  
Inclui referências e apêndices.

1. Inteligência computacional. 2. Sistemas não-lineares. I. Araújo, Aluizio Fausto Ribeiro (orientador). II. Título.

006.31

CDD (23. ed.)

UFPE - CCEN 2020 - 144

**Paulo Henrique Muniz Ferreira**

**“Controles Baseados em Mapas Auto-organizáveis para  
Sistemas Não-lineares Variantes no Tempo”**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Ciência da Computação.

Aprovado em: 06/03/2020.

---

**Orientador: Prof. Aluizio Fausto Ribeiro Araújo**

**BANCA EXAMINADORA**

---

Prof. Dr. Hansenclever de França Bassani  
Centro de Informática /UFPE

---

Prof. Dr. Ronaldo Ribeiro Barbosa de Aquino  
Departamento de Engenharia Elétrica / UFPE

---

Prof. Dr. Guilherme de Alencar Barreto  
Departamento de Engenharia de Teleinformática / UFC

---

Prof. Dr. Adrien Durand-Petiteville  
Departamento de Engenharia Mecânica / UFPE

---

Prof. Dr. Adrião Duarte Doria Neto  
Departamento de Engenharia da Computação / UFRN

*À minha amada mãe, Socorro Muniz, que, com muito amor e sabedoria, me proporcionou tudo para a conquista desta vitória.*

## AGRADECIMENTOS

Agradeço a minha adorável namorada, noiva e, hoje, esposa, Beatriz Moura. Seu incentivo e apoio fizeram com que eu tivesse estímulo para superar as dificuldades, por vezes, tão recorrentes. Como se não bastasse, ela ainda ajudou na revisão do texto (e das equações!). À ela, toda a minha imensa gratidão e amor. Também agradeço a família dela por ter me ajudado nessa dura batalha acadêmica.

Agradeço imensamente a minha mãe, Socorro Muniz, por todo seu esforço e amor incondicional. Mesmo não estando presente até o final do doutorado, esteve sempre e sempre estará no meu coração. Agradeço também a meu pai, Henrique Ferreira, por terem me apoiado, dia após dia. Além deles, meus sinceros agradecimentos a minha irmã, Ágata Kellen. Sem eles na minha vida, de forma alguma, eu teria conseguido realizar todas as minhas conquistas.

Aos meus amigos, em especial Adriano, Bruno Almeida, Bruno Maciel, Filipe, Hallison, Monique, Ruan, Renê, Sidartha, Silas e Wanessa, e a todos os colegas do laboratório da Associação de Pós-graduação, por todas as contribuições e discussões, além dos momentos de descontração e dos cafezinhos no intervalo.

Aos professores que contribuíram para minha formação e aos funcionários do Centro de Informática, em especial Roberto e Suilan.

Gostaria de expressar a minha gratidão aos professores da banca avaliadora, Adrião Duarte (UFRN), Guilherme Barreto (UFC), Ronaldo de Aquino (UFPE), Adrian Durand-Petiteville (UFPE) e Hansenclever Bassani (UFPE), pelo tempo dedicado à leitura e apreciação do meu trabalho, e pelas diversas sugestões e críticas que contribuíram para melhorar substancialmente do trabalho.

Por fim, mas com uma imensa importância, agradeço ao meu orientador, Aluísio Fausto Ribeiro Araújo. Desde da graduação, me auxiliou nessa jornada acadêmica e, durante o doutorado, sua presença, paciência, persistência e confiança no meu trabalho, possibilitou a conclusão desta tese.

Obrigado

## RESUMO

Em vários sistemas de controles reais, é exigido que o controlador atenda a requisitos complexos. Em particular, há uma demanda de estratégias de controle capazes de operar sistemas não-lineares sem o conhecimento do modelo matemático do comportamento do sistema. Também, há uma necessidade em considerar que o sistema a ser controlado pode apresentar um comportamento variando consideravelmente ao longo do regime de funcionamento. Exemplos de sistemas não-lineares desconhecidos variantes no tempo são encontrados em diversas áreas como: sistemas servo pneumáticos, veículos elétricos, e sistemas robóticos. Controladores convencionais como o controlador proporcional integral derivativo (PID) não tem todos os atributos necessários requeridos nesse contexto. Outras técnicas adequadas para controlar sistemas não-lineares como linearização por realimentação tem sua efetividade significativamente comprometida quando aplicada a um sistema não-linear desconhecido. Um controle adaptativo baseado em aprendizagem pode ser um alternativa para lidar com comportamentos não-lineares e operar mesmo sem o conhecimento do modelo matemático do sistema variante no tempo, em virtude da sua capacidade de aprender e adapta-se a novas situações. Nesta tese de doutorado, foram investigados trabalhos aplicando o mapa auto-organizável de Kohonen ou algum mapa derivado dessa rede em problemas de controle. Em seguida, estratégias de controles baseado em SOM para o contexto de sistemas não-lineares desconhecidos variantes no tempo foram desenvolvidas. Mais especificamente, as estratégias de controle usam um mapa inspirado no SOM, o mapa auto-organizável com campo receptivo adaptativo local (LARFSOM), em uma abordagem de controle em múltiplos modelos. O LARFSOM é capaz de dividir o espaço de operação do sistema e aprender ações de controle apropriadas para cada comportamento do sistema. Durante o regime de operação, o LARFSOM pode identificar o atual comportamento do sistema e selecionar a ação de controle. Adicionalmente, o LARFSOM foi estendido através de técnicas de mapeamento contínuo por partes, proporcionando uma ação de controle mais precisa e mais suave. Três sistemas não-lineares foram usados para avaliar o desempenho do controle: o sistema do pêndulo invertido, o sistema do manipulador robótico, e o oscilador de Van der Pol. Nos experimentos, os sistemas foram simulados com seus parâmetros variando ao longo do tempo e os resultados foram comparados com outros métodos. Além disso, a capacidade de mapeamento contínuo através do LARFSOM estendido foi validada em situações de aproximação de função estática, variante no tempo e em problemas reais. Os resultados sugerem que as técnicas de controle baseadas em mapas auto-organizáveis podem controlar sistema não-lineares variantes no tempo e podem ser eficazes ao lidar com configurações de sistema não apresentadas na etapa de treinamento.

**Palavras-chave:** Sistemas não-lineares. Sistemas variantes no tempo. Mapa Auto-organizável (SOM). Controle Adaptativo baseado em Aprendizagem. Controle baseado no SOM.

## ABSTRACT

In several control systems, a controller must be designed to meet complex requirements. Particularly, it is required control strategies for nonlinear systems without prior knowledge of the mathematical model of the system dynamics. Also, it is necessary to take into consideration that the controlled system can have a behavior that varies considerably over the operating regime. Examples of unknown time-varying nonlinear systems are found in many fields as Servo-pneumatic systems, electric vehicles, and robotic systems. Conventional controllers such as proportional-integral-derivative (PID) controllers do not have all the attributes necessary to meet the requirements. Other techniques suitable for controlling nonlinear systems, as feedback linearization, have the performance significantly decreased when applied in unknown nonlinear systems. Learning-based adaptive controllers can be suitable alternatives to deal with nonlinearities, and unknown time-varying systems because they can learn and adapt themselves to new situations. In this doctoral work, self-organizing maps (SOMs) applied for control system problems were investigated. Then, Self-Organizing Maps based control strategies were developed to deal with unknown nonlinear time-varying systems. Specifically, the proposed control strategies use a SOM-based model, local adaptive receptive field self-organizing map (LARFSOM), in a multiple model control approach. LARFSOM can split the operating space of the system and learns suitable control actions for each system behavior. During the operating regime, LARFSOM can identify the current system behavior and selects the control action. Additionally, the LARFSOM was extended using piecewise continuous mapping techniques to produce more precise and smoother control actions. Three nonlinear systems were used to evaluate the control performance: the inverted pendulum system, the robotic manipulator system, and the Van der Pol oscillator. In the experiments, the systems were simulated in a time-varying way. The results of the proposed methods were compared against the results of conventional methods. Furthermore, the properties of the extended LARFSOM were validated in simulations. The results suggest that the control techniques based on self-organizing maps can control time-varying nonlinear systems and can be effective when dealing with system configurations not presented during the training phase.

**Keywords:** Nonlinear systems. Time-varying systems. Self-organizing Map. Learning-based adaptive control. SOM-based control.

## LISTA DE FIGURAS

Figura 1 – Esquema de um controle adaptativo, adaptado de (ÅSTRÖM; WITTENMARK, 1995). . . . .	28
Figura 2 – Estrutura de um controle adaptativo neural indireto. . . . .	31
Figura 3 – Estrutura de um controle adaptativo direto com modelo de referência. . . . .	32
Figura 4 – Exemplo de topologia usada para SOM. As unidades da camada de entrada são conectadas a todos os nodos da camada de saída. . . . .	34
Figura 5 – Estrutura variante no tempo do LARFSOM. (a) Estrutura inicial do LARFSOM com dois nodos conectados. (b) Amostra de treinamento $\xi$ é apresentada ao mapa. (c) Um novo nodo é adicionado ao mapa. (d) Vencedor é ajustado. . . . .	46
Figura 6 – Diagrama de fluxo do procedimento de treinamento do LARFSOM estendido para aproximação de função. . . . .	48
Figura 7 – Os nodos e as conexões traçados em verde e azul são usados para calcular a saída aproximada através do D-LARFSOM, xD-LARFSOM, LWD-LARFSOM e LWxD-LARFSOM, respectivamente. O nodo $W_c$ (em verde) é o nodo vencedor em todas as ilustrações. . . . .	52
Figura 8 – Diagrama de blocos do controle por realimentação de estados. . . . .	62
Figura 9 – Diagrama de blocos do sistema de controle usando LARFSOM. . . . .	63
Figura 10 – O mapa auto-organizável treinado com padrões de entrada concatenados. No exemplo, três padrões de entrada são concatenados para formar o vetor de entrada da rede. Adaptado da Figura 2 de (KANGAS, 1990). . . . .	63
Figura 11 – Diagrama de blocos do sistema de controle adaptativo direto baseado no LARFSOM. . . . .	67
Figura 12 – Função a ser aproximada. . . . .	72
Figura 13 – Curva de aprendizagem média na aproximação de função com saída estática com vetor de entrada 2D (a) e 10D (b). . . . .	78
Figura 14 – Curva de aprendizagem média na aproximação de função com saída estática com vetor de entrada 20D (a) e tempo médio de execução na aproximação de função com saída estática em todos os experimentos (b). Os resultados apresentados nas Figuras 13a, 13b e 14a sugerem que os métodos não são sensíveis a dados redundantes e irrelevantes. Detalhes estão descritos na Seção 6.2.1. . . . .	79
Figura 15 – Curva de aprendizagem média para a aproximação de função com saída variante no tempo no experimento <i>Exp2.1</i> . Esses resultados sugerem que os métodos podem lidar com funções variante no tempo. Mais detalhes na Seção 6.2.2. . . . .	81

Figura 16 – Curva de aprendizagem média para a aproximação de função com saída variante no tempo no experimento <i>Exp2.2</i> . Resultados apresentados em b sugerem que os métodos podem lidar com funções com saída variante no tempo, e em a, mostram que os métodos foram sensíveis a fortes ruídos. Detalhes são descritos na Seção 6.2.2. . . . . .	82
Figura 17 – Resultados no experimento da aproximação de função com distribuição de entrada variante no tempo ( <i>Exp3.1</i> ). Em (a), curva de aprendizagem média e em , média do número de nodos usando o LARFSOM para todas as 100 repetições, para repetições com $ c_1 - c_2  < 0,25$ e para repetições com $ c_1 - c_2  > 0,75$ (b). Resultados apresentados em a sugerem que os métodos podem lidar com funções variante no tempo. Mais detalhes nas Seções 6.2.2 e 6.2.3. . . . . .	83
Figura 18 – Parâmetros da planta usados para treinamento e teste. Figura adaptada de (MINATOHARA; FURUKAWA, 2011). . . . . .	90
Figura 19 – Posição desejada do carrinho (em vermelho) e configurações do sistema durante a fase de teste. . . . . .	91
Figura 20 – Resultado da simulação do LQR para um caso de conhecimento total das configurações da planta. . . . . .	92
Figura 21 – Resultado da simulação do $LQR_{p_4}$ . No início, há uma resposta não-amortecida. Porém, o sistema fica instável a partir da configuração de teste $p_B$ . . . . . .	93
Figura 22 – Resultado da simulação do $LQR_{p_4}$ usando apenas as configurações $p_A$ , $p_C$ , $p_D$ , $p_E$ e $p_I$ do conjunto de teste. . . . . .	93
Figura 23 – Resultado do LARFSOM para simulação de controle do sistema usando as configurações de treinamento. (a) mostra o ângulo do pêndulo $\theta$ , a posição do carrinho $x$ e o sinal de controle $u$ . (b) mostra o valor estimado para o sinal de controle em malha aberta $v$ e o sinal de realimentação $\mathbf{K} \times \mathbf{x}$ . (c) mostra os ganhos estimados. . . . . .	95
Figura 24 – Simulações de teste do pêndulo invertido usando o LARFSOM e o D-LARFSOM. (a) e (b) mostram o ângulo do pêndulo $\theta$ , a posição do carrinho $x$ e o sinal de controle $u$ . (c) e (d) mostram o valor estimado para o sinal de controle em malha aberta $v$ e o sinal de realimentação $\mathbf{K} \times \mathbf{x}$ . (e) e (f) mostram os ganhos estimados. . . . . .	96
Figura 25 – Simulações de teste do pêndulo invertido usando o xD-LARFSOM. (a) mostra o ângulo do pêndulo $\theta$ , a posição do carrinho $x$ e o sinal de controle $u$ . (b) mostra o valor estimado para o sinal de controle em malha aberta $v$ e o sinal de realimentação $\mathbf{K} \times \mathbf{x}$ . (c) mostra os ganhos estimados. . . . . .	97

Figura 26 – (a) diagrama esquemático do manipulador robótico de único elo. (b) distúrbio externo ocorrido durante uma simulação do sistema com variação da massa de carga durante o tempo. . . . .	99
Figura 27 – Diagrama de blocos do sistema controlado pelo LQR mais dois integradores para garantir um erro pequeno no estado estacionário. . . . .	101
Figura 28 – Resultado da simulação para as configurações do sistema para treinamento. . . . .	103
Figura 29 – Resultado das simulações do manipulador robótico usando o LARFSOM e D-LARFSOM. (a) e (b) mostram o estado do sistema $\mathbf{x}$ e o sinal de controle $u$ . (c) e (d) mostram o sinal de controle $u$ , o sinal da malha de realimentação $h$ e os ganhos da malha de realimentação $\mathbf{K}$ . (e) e (f) mostram os erros de rastreamento. . . . .	106
Figura 30 – Resultado das simulações do manipulador robótico usando o xD-LARFSOM. (a) mostra o estado do sistema $\mathbf{x}$ e o sinal de controle $u$ . (b) mostra o sinal de controle $u$ , o sinal da malha de realimentação $h$ e os ganhos da malha de realimentação $\mathbf{K}$ . (c) mostram os erros de rastreamento. . . . .	107
Figura 31 – Resultado do controle adaptativo direto baseado no LARFSOM. Resposta do sistema e desejada, sinal de controle, e aproximação de $f$ por $\tilde{f}$ usando o D-LARFSOM. . . . .	110
Figura 32 – Resultado usando o controle adaptativo direto baseado no LARFSOM. (a) Resposta transitório no intervalo $t \in [0s\ 15s]$ . (b) Resposta no estado estacionário no intervalo $t \in [235s\ 250s]$ . . . . .	111
Figura 33 – Diagrama de bloco do controle linear quadrático para rastreamento. . . . .	128
Figura 34 – (a) Diagrama esquemático do sistema mecânico do pêndulo invertido em um carrinho. (b) Diagrama esquemático do corpo livre do pêndulo. . . . .	132

## LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos aplicando SOM para problemas de controle usando a abordagem de múltiplos modelos. . . . .	41
Tabela 2 – Resumo dos outros trabalhos aplicando SOM para problemas de controle. . . . .	42
Tabela 3 – Principais características das bases de dados de problemas de <i>benchmarks</i> . . . . .	75
Tabela 4 – Parâmetros e valores dos métodos utilizados na aproximação de funções de problemas de <i>benchmarks</i> . . . . .	76
Tabela 5 – Média do nMSE dos métodos após 100k passos de treinamento. . . . .	80
Tabela 6 – Valores para os parâmetros de treinamento usados nos problemas de <i>benchmarks</i> . . . . .	85
Tabela 7 – Média do erro RMSE, média do tempo de treinamento, média do tempo de resposta para aproximação de funções de problemas de <i>benchmarks</i> . . . . .	86
Tabela 8 – Vantagens do LWD-LARFSOM e LWxD-LARFSOM em comparação ao SVR e LWI-LARFSOM. . . . .	87
Tabela 9 – Valores dos parâmetros $m$ e $l$ usado para testar . . . . .	90
Tabela 10 – Valor dos parâmetros para o controle LQR. . . . .	91
Tabela 11 – Valores dos parâmetros usados no treinamento dos LARFSOMs . . . . .	94
Tabela 12 – Resultado dos índices de desempenho na simulação do pêndulo invertido. . . . .	94
Tabela 13 – Ganhos dos integradores para todas as massas de carga de treinamento ( $m_r$ ). . . . .	102
Tabela 14 – Medidas de desempenho avaliadas nas simulações para gerar o conjunto de treinamento. . . . .	102
Tabela 15 – Valores dos parâmetros usados no treinamento das LARFSOMs . . . . .	104
Tabela 16 – Resultado das medidas de desempenho nas simulações usando controle por realimentação de estados baseado no LARFSOM. . . . .	105
Tabela 17 – Parâmetros de treinamento do LARFSOM. . . . .	109
Tabela 18 – Resultado das medidas de desempenhos. . . . .	110
Tabela 19 – Controle Rastreador Linear Quadrático . . . . .	129

## LISTA DE SIGLAS

AFC	Adaptive Fuzzy Control ( <i>Controle Adaptativo Fuzzy</i> )
ASOM	Adaptive Self-Organizing Map ( <i>Mapa Auto-organizável Adaptativo</i> )
BIBO	Bounded-input/bounded-output ( <i>Entrada limitada/saída limitada</i> )
D	Difference Method ( <i>Método da Diferença</i> )
IAE	Integral Absolute Error ( <i>Integral do Erro Absoluto</i> )
ITAE	Integral Time-weighted Absolute Error ( <i>Integral do Erro Absoluto Ponderado no Tempo</i> )
I-SOM	Interpolating Self-Organizing Map ( <i>Mapa Auto-organizável com Interpolação</i> )
I-LARFSOM	Interpolating Local Adaptive Receptive Field Self-Organizing Map ( <i>Mapa Auto-organizável com Campo Receptivo Adaptativo Local com Interpolação</i> )
IVK	Induced Voronoi Kernel ( <i>Kernel de Voronoi Induzido</i> )
LARFSOM	Local Adaptive Receptive Field Self-Organizing Map ( <i>Mapa Auto-organizável com Campo Receptivo Adaptativo Local</i> )
LLM	Local Linear Maps ( <i>Mapas Lineares Locais</i> )
LQR	Linear-Quadratic Regulator ( <i>Regulador Linear Quadrático</i> )
LWD	Locally Weighted Difference Method ( <i>Método da Diferença Localmente Ponderado</i> )
LWI-LARFSOM	Locally Weighted Interpolating Local Adaptive Receptive Field Self-Organizing Map ( <i>Mapa Auto-organizável com Campo Receptivo Adaptativo Local com Interpolação Localmente Ponderada</i> )
LWIGNG	Locally Weighted Interpolating Growing Neural Gas ( <i>Gás Neural Crescente com Interpolação Localmente Ponderada</i> )
LWxD	Locally Weighted Extended Difference Method ( <i>Método da Diferença Estendido Localmente Ponderado</i> )
MLP	Multilayer Perceptron ( <i>Perceptron Multicamadas</i> )
mnSOM	Modular Network SOM ( <i>Rede Modular SOM</i> )
MIMO	Multiple-input multiple-output ( <i>Múltiplas entradas múltiplas saídas</i> )
nMSE	Normalised Mean Square Error ( <i>Erro Quadrático Médio normalizado</i> )
PDA	Proportional-Derivative-Acceleration Controller ( <i>Controlador Proporcional Derivativo Acelerativo</i> )

PID	Proportional-Integral-Derivative Controller ( <i>Controlador Proporcional Integral Derivativo</i> )
QSMC	Quasi-Sliding Mode Controller ( <i>Controlador por Modo Quase-deslizante</i> )
RBF	Radial-Basis Function Networks ( <i>Redes Neurais com Função de Base Radial</i> )
SOAC	Self-Organizing Adaptive Controller ( <i>Controlador Adaptativo Auto-organizável</i> )
SOLIM	Self-Organising Locally Interpolating Maps ( <i>Mapas Auto-organizáveis Interpolado Localmente</i> )
SOFM	Self-Organizing Feature Map ( <i>Mapa Auto-organizável de Característica</i> )
SOM-TVS	Self-Organizing Maps with Time-Varying Structure ( <i>Mapas Auto-organizáveis com Estrutura Variante no Tempo</i> )
SOI	Self-Organizing Input ( <i>Auto-organização da Entrada</i> )
SOM	Self-Organizing Map ( <i>Mapa Auto-organizável</i> )
SVR	Support Vector Regression ( <i>Máquina de Vetores de Suporte para Regressão</i> )
VQTAM	Vector-Quantized Temporal Associative Memory ( <i>Memória Associativa Temporal por Quantização Vetorial</i> )
WNN	Wavelet Neural Network ( <i>Rede Neural de Wavelet</i> )
xD	Extended Difference Method ( <i>Método da Diferença Estendido</i> )

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
1.1	MOTIVAÇÃO E JUSTIFICATIVA	18
1.2	ESTADO DO PROBLEMA	19
1.3	OBJETIVO	21
1.4	CONTRIBUIÇÕES	21
1.5	ESTRUTURA DA PROPOSTA	21
<b>2</b>	<b>FUNDAMENTOS DE CONTROLE PARA SISTEMAS NÃO-LINEARES E VARIANTES NO TEMPO</b>	<b>23</b>
2.1	SISTEMA DE CONTROLE	23
2.2	DEFINIÇÃO DO PROBLEMA	25
2.3	MÉTODOS DE CONTROLE	26
<b>2.3.1</b>	<b>Controlador Proporcional Integral Derivativo</b>	<b>26</b>
<b>2.3.2</b>	<b>Controle Não-Linear</b>	<b>26</b>
2.3.2.1	Método de Linearização por Realimentação	26
2.3.2.2	Método <i>Backstepping</i>	27
<b>2.3.3</b>	<b>Controle Adaptativo</b>	<b>28</b>
<b>2.3.4</b>	<b>Controle Ótimo</b>	<b>28</b>
<b>2.3.5</b>	<b>Controle Inteligente</b>	<b>29</b>
<b>2.3.6</b>	<b>Controle Neural</b>	<b>30</b>
2.3.6.1	Controle Adaptativo Indireto usando Redes Neurais	31
2.3.6.2	Controle Adaptativo Direto usando Redes Neurais	32
2.4	CONCLUSÃO	32
<b>3</b>	<b>ESTRATÉGIAS DE CONTROLE BASEADO EM SOM</b>	<b>33</b>
3.1	CONTROLE BASEADO EM SOM	33
<b>3.1.1</b>	<b>Mapa Auto-organizável - SOM</b>	<b>33</b>
3.2	CARACTERÍSTICAS DAS ESTRATÉGIAS DE CONTROLE BASEADO EM SOM	35
3.3	TRABALHOS RELACIONADOS	36
<b>3.3.1</b>	<b>SOM Aplicado em Controle Adaptativo baseado em Múltiplos Mo- delos</b>	<b>37</b>
<b>3.3.2</b>	<b>Outros Trabalhos Relacionados</b>	<b>39</b>
3.4	CONCLUSÃO	40

<b>4</b>	<b>MAPA AUTO-ORGANIZÁVEL PARA SISTEMA NÃO-LINEAR VARIANTE NO TEMPO . . . . .</b>	<b>43</b>
4.1	LARFSOM - MAPA AUTO-ORGANIZÁVEL COM CAMPO RECEPTIVO ADAPTATIVO LOCAL . . . . .	43
<b>4.1.1</b>	<b>Estrutura do LARFSOM . . . . .</b>	<b>44</b>
<b>4.1.2</b>	<b>Algoritmo de Treinamento do LARFSOM . . . . .</b>	<b>45</b>
4.2	MAPEAMENTO CONTÍNUO POR PARTES COM LARFSOM . . . . .	48
<b>4.2.1</b>	<b>Trabalhos Relacionados . . . . .</b>	<b>49</b>
4.2.1.1	Mapa Linear Local . . . . .	50
4.2.1.2	Interpolação Localmente Ponderada . . . . .	51
<b>4.2.2</b>	<b>LARFSOM com Interpolação Linear por Partes - I-LARFSOM . . .</b>	<b>51</b>
<b>4.2.3</b>	<b>Método da Diferença - D-LARFSOM . . . . .</b>	<b>54</b>
<b>4.2.4</b>	<b>Método da Diferença Estendido - xD-LARFSOM . . . . .</b>	<b>55</b>
<b>4.2.5</b>	<b>LARFSOM para Mapeamento Contínuo por Partes Localmente Ponderado . . . . .</b>	<b>56</b>
4.2.5.1	Interpolação Linear por Partes Localmente Ponderada - LWI-LARFSOM . .	57
4.2.5.2	Método da Diferença Localmente Ponderado - LWD-LARFSOM . . . . .	57
4.2.5.3	Método da Diferença Estendido Localmente Ponderado - LWxD-LARFSOM	57
4.2.5.4	Adaptação da Abrangência dos Modelos Locais . . . . .	58
<b>4.2.6</b>	<b>Treinamento do LARFSOM com Métodos de Mapeamento Contínuo por Partes . . . . .</b>	<b>58</b>
4.3	CONCLUSÃO . . . . .	60
<b>5</b>	<b>ESTRATÉGIAS DE CONTROLE BASEADO NO LARFSOM . . . .</b>	<b>61</b>
5.1	CONTROLE POR REALIMENTAÇÃO DE ESTADOS BASEADO NO LARFSOM . . . . .	61
<b>5.1.1</b>	<b>Vetor Concatenado de Entrada . . . . .</b>	<b>62</b>
<b>5.1.2</b>	<b>Discussão . . . . .</b>	<b>64</b>
5.2	CONTROLE ADAPTATIVO DIRETO BASEADO NO LARFSOM . . . . .	66
<b>5.2.1</b>	<b>Classe de Sistemas Considerados . . . . .</b>	<b>66</b>
<b>5.2.2</b>	<b>Controle Adaptativo Direto Baseado no LARFSOM . . . . .</b>	<b>66</b>
<b>5.2.3</b>	<b>Análise de Estabilidade . . . . .</b>	<b>68</b>
5.2.3.1	Incluindo erro de aproximação . . . . .	70
<b>6</b>	<b>VALIDAÇÃO . . . . .</b>	<b>71</b>
6.1	DESCRIÇÃO DO EXPERIMENTOS . . . . .	71
<b>6.1.1</b>	<b>Dados Sintéticos . . . . .</b>	<b>71</b>
6.1.1.1	Função de Saída Estática . . . . .	72
6.1.1.2	Função com Saída Variante no Tempo . . . . .	73
6.1.1.3	Distribuição de Entrada Variante no Tempo . . . . .	74

6.1.1.4	Ajuste de Parâmetros . . . . .	74
<b>6.1.2</b>	<b>Problemas de <i>Benchmarks</i></b> . . . . .	<b>75</b>
6.1.2.1	Ajuste dos Parâmetros . . . . .	76
6.2	RESULTADOS EXPERIMENTAIS . . . . .	77
<b>6.2.1</b>	<b>Resultados Comparativos em Aproximação de Função com Saída Estática</b> . . . . .	<b>77</b>
<b>6.2.2</b>	<b>Resultados Comparativos em Aproximação de Função com Saída Variante no Tempo</b> . . . . .	<b>80</b>
<b>6.2.3</b>	<b>Resultados Comparativos em Aproximação de Função com Distribuição Variante no Tempo</b> . . . . .	<b>81</b>
<b>6.2.4</b>	<b>Resultados Comparativos em Benchmarks</b> . . . . .	<b>84</b>
6.3	CONCLUSÃO . . . . .	87
<b>7</b>	<b>EXPERIMENTOS NO PÊNDULO INVERTIDO</b> . . . . .	<b>89</b>
7.1	PÊNDULO INVERTIDO COM CARRINHO . . . . .	89
7.2	DESCRIÇÃO DO EXPERIMENTO . . . . .	89
7.3	RESULTADOS DOS EXPERIMENTOS . . . . .	91
<b>7.3.1</b>	<b>Resultados LQR</b> . . . . .	<b>91</b>
<b>7.3.2</b>	<b>Resultados do Controlador por Realimentação de Estado Baseado no LARFSOM</b> . . . . .	<b>92</b>
<b>8</b>	<b>EXPERIMENTOS NO MANIPULADOR ROBÓTICO</b> . . . . .	<b>98</b>
8.1	SISTEMA DINÂMICO DO MANIPULADOR ROBÓTICO . . . . .	98
8.2	LINEARIZAÇÃO DO SISTEMA . . . . .	99
8.3	DESCRIÇÃO DO EXPERIMENTO . . . . .	99
8.4	RESULTADOS DOS EXPERIMENTOS . . . . .	100
<b>8.4.1</b>	<b>Descrição do LQR com Integradores</b> . . . . .	<b>100</b>
<b>8.4.2</b>	<b>Resultados LQR</b> . . . . .	<b>101</b>
<b>8.4.3</b>	<b>Resultados do Controlador por Realimentação de Estado Baseado no LARFSOM</b> . . . . .	<b>102</b>
<b>9</b>	<b>EXPERIMENTOS NO OSCILADOR DE VAN DER POL</b> . . . . .	<b>108</b>
9.1	SISTEMA OSCILANTE DE VAN DER POL . . . . .	108
9.2	MODELO DE REFERÊNCIA - OSCILADOR DE DUFFING . . . . .	108
9.3	DESCRIÇÃO DO EXPERIMENTO . . . . .	108
9.4	RESULTADOS DO CONTROLADOR ADAPTATIVO DIRETO BASEADO NO LARFSOM . . . . .	109
<b>10</b>	<b>CONCLUSÃO</b> . . . . .	<b>112</b>
10.1	TRABALHO PUBLICADO . . . . .	114
10.2	LIMITAÇÕES DAS ABORDAGENS . . . . .	115

10.3	TRABALHOS FUTUROS . . . . .	115
	<b>REFERÊNCIAS . . . . .</b>	<b>117</b>
	<b>APÊNDICE A – ESTABILIDADE . . . . .</b>	<b>125</b>
	<b>APÊNDICE B – REGULADOR LINEAR QUADRÁTICO . . . . .</b>	<b>128</b>
	<b>APÊNDICE C – DEDUÇÃO DAS REGRAS DE APRENDIZAGEM</b>	<b>130</b>
	<b>APÊNDICE D – MODELAGEM DA DINÂMICA DO PÊNDULO</b>	
	<b>INVERTIDO . . . . .</b>	<b>132</b>

## 1 INTRODUÇÃO

A atual tendência dos sistemas de controle é caracterizada por um aumento na complexidade dos requisitos. Por exemplo, um sistema de controle deve operar um sistema não-linear desconhecido com múltiplas entradas e múltiplas saídas (MIMO), e ao mesmo tempo, deve-se considerar a possibilidade de múltiplos comportamentos do ambiente, do processo ou da planta (WU; CHEN; LI, 2015; TAN et al., 2017). Diferentes comportamentos podem ser caracterizados por mudanças repentinas nos valores paramétricos do sistema, falha de sensores ou subsistemas, e até distúrbios externos (NARENDRA; BALAKRISHNAN, 1997). Naturalmente, há uma demanda de controles capazes de repostas rápidas e acuradas às mudanças do sistema (NARENDRA; BALAKRISHNAN, 1997).

Segundo (NARENDRA, 2016), controladores usuais não são capazes de satisfazer tais requisitos complexos. Por exemplo, os controladores proporcionais integrativos derivativos convencionais (PID) não alcançam resultados satisfatórios para sistemas não-lineares variantes no tempo devido à sua estrutura linear com valores de ganhos constantes (CHEN; GONG, 2017). Outras técnicas apropriadas para sistemas não-lineares, como a linearização por realimentação e o *backstepping*, tem seus desempenhos significativamente reduzidos quando aplicadas em um sistema no qual a não-linearidade é desconhecida ou há incertezas sobre os distúrbios (SOLTANPOUR; KHOOBAN; KHALGHANI, 2016). Em alguns casos, essas técnicas podem tornar o sistema em malha fechada instável (SLOTINE; LI, 1991; KHALIL, 2002).

Uma possível via para desenvolver um controle capaz de responder adequadamente em um contexto de sistemas não-lineares variantes no tempo desconhecidos é utilizar métodos da computação inteligente. Através destes métodos, o controlador terá as habilidades de aprender, identificar e se adaptar a novas situações (NARENDRA; BALAKRISHNAN, 1997). Estes predicados podem permitir a capacidade de lidar com não-linearidades, sistemas desconhecidos, e mudanças repentinas durante o regime de operação do sistema. Considerando este contexto, este doutorado apresenta propostas de controle usando técnicas de computação inteligente, mais especificamente, os Mapas Auto-organizáveis de Kohonen.

### 1.1 MOTIVAÇÃO E JUSTIFICATIVA

Segundo Passino (2010), sistemas de controle inteligente é um dos campos mais investigados dos controles modernos e têm atraído interesse tanto do ponto de vista acadêmico quanto industrial. Esta importância vem em virtude da sua aplicabilidade com sucesso em diversos tipos de sistemas tais como industriais, aeroespaciais, robóticos, automotivos e químicos. Essa classe é referente aos controles que utilizam alguma técnica de inteligência artificial ou inteligência computacional. Entre os principais tipos de sistemas de controle

inteligente estão: os controles nebulosos (*fuzzy*), os controles especialistas (*expert*), os controles evolucionários e os controles neurais.

No campo dos controles inteligentes, ainda há tópicos desafiadores tanto em termos teóricos quanto em aspectos práticos. Um deles é a área de conhecimento sobre os controladores fundamentados no mapa auto-organizável de Kohonen (SOM, do inglês self-organizing map) (KOHONEN, 1982; KOHONEN, 1990). Segundo Kohonen (2013), o mapa auto-organizável é uma rede neural artificial relacionada ao clássico problema de quantização vetorial e é amplamente aplicada em problemas de agrupamentos e exploração de dados. Mas, há trabalhos reportados em outros campos como na engenharia de controle (BARRETO; ARAÚJO; RITTER, 2003; HÜLSEN, 2007).

O SOM possui características interessantes para ser aplicado no campo de sistemas de controle. Uma delas é o paradigma de aprendizagem intrínseca a essa rede neural. O mapa auto-organizável de Kohonen tem uma aprendizagem não-supervisionada (KOHONEN, 2013). Nesse paradigma, a estrutura neural aprende através de dados não-rotulados e o objetivo é identificar os diferentes padrões dos dados e construir representações internas desses padrões. Esse tipo de aprendizagem pode simplificar o controlador e, conseqüentemente, implicar em uma carga menor na programação de um sistema de controle (BARRETO; ARAÚJO; RITTER, 2003). Ao mesmo tempo, pode melhorar a robustez para lidar com grande volume de dados e redundância de informações (HÜLSEN, 2007). Outro ponto sobre modelo não-supervisionado é que ele pode ser projetado para aprender incrementalmente (BARRETO; ARAÚJO; RITTER, 2003). Desta forma, o sistema de controle pode, ao longo da execução, aprimorar seu desempenho. Essa característica é encontrada em sistemas controlados por um operador humano. À medida que o operador humano vai ganhando experiência, o sistema vai sendo melhor controlado. Na Seção 3.2, as propriedades do SOM em sistemas de controle são detalhas.

## 1.2 ESTADO DO PROBLEMA

A grande maioria dos sistemas de controle encontrados na prática exibem uma dinâmica não-linear (SLOTINE; LI, 1991). Outro ponto também importante que ocorre na prática é que os parâmetros que definem o comportamento do sistema podem mudar ao longo do tempo (NARENDRA, 2016). Por consequência destas duas características, o sistema pode apresentar uma dinâmica tão complexa que um sistema de controle de modelo único apresentará dificuldades para operar adequadamente (CHO et al., 2007). Na prática, isso também se traduz em uma inviabilização da modelagem através da linearização em um ponto de operação, pois o sistema pode requerer operações em regiões do espaço de estados onde a aproximação linear não é válida.

Para esses sistemas complexos, modelos locais de controle vêm sendo aplicados e eles podem simplificar bastante tanto a modelagem quanto o projeto do controlador (CHO et al., 2007; CHO et al., 2006). A ideia é usar vários modelos para representar a dinâmica

do sistema ao longo de todo o regime de funcionamento e então utilizar um sistema de identificação para selecionar o controlador mais apropriado para o comportamento atual do sistema. Esta estratégia de identificar e controlar simultaneamente para controle de sistemas não-lineares foi inicialmente proposta por Narendra e Parthasarathy (1990). Posteriormente, Narendra, Balakrishnan e Ciliz (1995), Narendra e Balakrishnan (1997) reformularam em um conceito de múltiplos modelos do sistema com a capacidade de comutação.

A ideia de controles de múltiplos modelos se encaixa bem com a estrutura de mapas auto-organizáveis. O SOM pode ser utilizado para dividir o espaço de operação em subespaços e selecionar um modelo de controle apropriado em cada região (CHO et al., 2007). Por sua vez, este modelo de controle local pode ser potencialmente mais simples do que um único controle global. A abordagem de múltiplos modelos está relacionada, também, ao sucesso do uso de Mapas Auto-organizáveis em identificação de sistemas dinâmicos não-lineares (BARRETO; ARAÚJO, 2004; BARRETO, 2007). A identificação do sistema pode ser realizada pela rede SOM. Isto é possível porque o SOM é capaz de criar uma representação da dinâmica do sistema e discriminar diferentes comportamentos a partir da ativação de diferentes regiões do mapa topológico. O aspecto mais interessante em modelar um controle de múltiplos modelos com o SOM é que essa rede neural pode ser usada para generalizar os modelos locais de controle aprendidos. O controle poderá ser robusto para variações de parâmetros no sistema e, além disso, poderá também operar para parâmetros não considerados na etapa de treinamento.

Porém, controles baseados no SOM podem apresentar uma perda no desempenho devido a quantização do mapeamento (HÜLSEN, 2007). Como a rede SOM realiza uma quantização vetorial, o seu processo de treinamento faz uma discretização tanto do espaço de entrada quanto do de saída (KOHONEN, 1990). Conseqüentemente, o mapeamento construído não é contínuo. Em um problema que as entradas e as saídas podem assumir valores contínuos, a chance destes valores serem exatamente iguais aos armazenados nos nodos é ínfima. Isso pode implicar em uma menor acurácia na identificação e na ação de controle do sistema, e dificultar a generalização a partir de poucos exemplos. Uma possível solução para isto seria utilizar alguma técnica capaz de construir um mapeamento contínuo ou um mapeamento contínuo por partes sobre o SOM.

Além de um mapeamento contínuo, essa técnica aplicada entre os nodos de uma rede SOM pode proporcionar as seguintes propriedades: coerência no mapeamento entrada-saída visto que os cálculos podem ser realizados de modo similar tanto no espaço de entrada quanto no espaço de saída; a técnica que gera valores intermediários pode ser realizada usando somente o nodo vencedor e seus vizinhos topológicos ao invés de todos os nodos da rede; e resultados satisfatórios podem ser obtidos mesmo com um número menor de nodos e de exemplos treinamento em comparação ao treinamento de uma rede sem mapeamento contínuo (LUDWIG et al., 1995; WALTER; RITTER, 1996). Portanto, o foco

---

desta pesquisa é explorar a estratégia de controle baseado em SOM com a capacidade de mapeamento contínuo por partes em uma abordagem de múltiplos modelos para controlar sistemas não-lineares e com mudanças desconhecidas em tempo de operação.

### 1.3 OBJETIVO

O objetivo principal deste doutorado é propor, implementar e analisar sistemas de controle inteligentes baseados em mapas auto-organizáveis capazes de controlar sistemas desconhecidos não-lineares com parâmetros variantes no tempo.

### 1.4 CONTRIBUIÇÕES

As contribuições deste trabalho são:

- Extensões aplicáveis em mapas auto-organizáveis para aproximação de função não-linear variante no tempo;
- Um controle inteligente baseado em SOM sob a estratégia de realimentação de estados capaz de identificar e gerar uma ação de controle apropriada para controlar sistemas desconhecidos não-lineares variantes no tempo;
- Um controle adaptativo direto baseado em SOM aplicável a uma classe de sistemas não-lineares variantes no tempo com prova de convergência;

### 1.5 ESTRUTURA DA PROPOSTA

O restante deste documento aborda os seguintes assuntos:

O Capítulo 2 descreve uma visão geral dos conceitos relacionados a sistema de controle e uma definição formal do problema de controle. Uma introdução de diferentes métodos de controle também é feita ao longo desse capítulo.

O Capítulo 3 apresenta investigação de sistemas de controle baseados em mapas auto-organizáveis. São descritas as principais características positivas e as limitações para esse tipo de controle. Também, há um levantamento de trabalhos que propuseram controladores baseados no SOM.

As abordagens de controle desenvolvidas nesta tese são baseadas no mapa auto-organizável com campo receptivo adaptativo local (LARFSOM). No Capítulo 4, há uma descrição do LARFSOM para modelagem de funções não-lineares variantes no tempo. Também, são apresentadas técnicas da literatura e propostas desenvolvidas durante este doutorado para a construção de mapeamento contínuo por partes sobre mapas auto-organizáveis.

O Capítulo 5 apresenta as duas propostas de controle baseado em SOM para problemas de controle de sistemas não-lineares variantes no tempo desenvolvidas. Primeiramente, é detalhada a estratégia de controle sob o arcabouço de realimentação de estados e como

o LARFSOM leva em conta o fator temporal da dinâmica do sistema para identificar o comportamento atual. Em segundo momento, a segunda proposta de controle baseado em SOM é apresentada. Essa segunda proposição consiste de um controle adaptativo direto em múltiplos modelos para uma classe de sistemas não-lineares variantes no tempo. Para a segunda proposição, esse capítulo apresenta uma demonstração de estabilidade no sentido de Lyapunov.

O Capítulo 6 reporta os resultados das validações realizadas no LARFSOM para mapeamento contínuo por partes. Diferentes técnicas para mapeamento contínuo sobre o LARFSOM foram avaliadas. As avaliações foram realizadas utilizando funções não-lineares sob ruídos com dados irrelevantes e redundantes. Também foram realizados experimentos sob condições de mudança da saída ao longo do tempo e mudança na distribuição de entrada ao longo do tempo. Por fim, validações foram executadas sobre problemas de dados do mundo real.

O controle proposto baseado no LARFSOM em realimentação de estados foi simulado nos sistemas do pêndulo invertido e manipulador robótico para avaliar a capacidade de controle. Nos capítulos 7 e 8, há a descrição matemática de cada sistema, a descrição do experimentos e os resultados obtidos.

O controle adaptativo direto baseado no LARFSOM foi simulado no oscilador de Van der Pol para avaliar a capacidade de controle. O Capítulo 9 traz a descrição matemática do sistema, a descrição do experimentos e os resultados obtidos.

O Capítulo 10 trata das discussões finais e trabalhos futuros.

## 2 FUNDAMENTOS DE CONTROLE PARA SISTEMAS NÃO-LINEARES E VARIANTES NO TEMPO

### 2.1 SISTEMA DE CONTROLE

Segundo Nise (2011), um sistema de controle consiste em subsistemas e processos (ou plantas) construído para obter uma resposta de saída desejada com uma performance específica para uma dada entrada. Em geral, sistemas de controle são encontrados em aplicações de diversas engenharias tais como elétrica, da computação, de controle, mecânica, espacial, biomédica e química. A evolução de sistemas de controle nasce da necessidade de desenvolver estratégias de controle para operar em problemas cada vez mais complexos como, por exemplo, sistemas de múltiplas entradas e múltiplas saídas, variantes no tempo e sem o conhecimento *a priori* da modelagem matemática. Para atender os requisitos cada vez mais demandantes no desempenho dos sistemas de controle, vêm sendo empregados diferentes esquemas de controle avançado. Alguns deles são: *controle não-linear*, *controle adaptativo*, *controle ótimo* e *controle inteligente*.

Neste trabalho de doutorado, foram considerados sistemas que podem ser representados pela seguinte modelagem matemática:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_f(t)) \quad (2.1)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.2)$$

onde os vetores  $\mathbf{y}(t)$ ,  $\mathbf{x}(t)$  e  $\mathbf{u}(t)$  representam a resposta, o estado atual e a entrada do sistema, respectivamente. A função  $\mathbf{h}$  representa o mapeamento estado e entrada do sistema para a resposta do sistema. Já a função  $\mathbf{f}$  representa o comportamento do sistema com suas não-linearidades e pode variar ao longo do tempo  $t$ . Pode-se denotar  $\mathbf{p}_f(t)$  para se referir aos parâmetros da função  $\mathbf{f}$  quando eles variam ao longo do tempo. Abaixo, são apresentados os conceitos básicos de sistemas dinâmicos.

**Estado** O estado do sistema dinâmico é constituído por um conjunto de variáveis, conhecidas por *variáveis de estado*, de modo que com o conhecimento destas variáveis no instante  $t_0$  e a entrada no instante  $t$  ( $t \geq t_0$ ), pode-se determinar completamente a saída do sistema no instante  $t$ .

Sendo assim, o estado do sistema dinâmico em  $t$  é determinado univocamente pelos valores das variáveis no instante  $t_0$  e a entrada no instante  $t$  ( $t \geq t_0$ ), e é independente do estado e entrada para  $t < t_0$ . Normalmente, ao lidar com sistemas lineares e invariantes no tempo, é escolhido o tempo  $t_0$  como zero.

**Vetor de Estado** O estado do sistema dinâmico é constituído por um conjunto de  $n$  variáveis de estado. Essas variáveis podem ser consideradas como as  $n$  componentes de um vetor  $\mathbf{x}(t)$ . Tal vetor é chamado de *vetor de estado*.

**Espaço de Estado** O espaço de estado é um espaço  $n$ -dimensional onde  $n$  é o número de variáveis de estado. Cada eixo de coordenada representa uma variável de estado e qualquer estado do sistema pode ser representado por um ponto no espaço de estado.

**Estabilidade** A estabilidade do sistema pode significar que as saídas e os sinais internos de um sistema de controle são delimitados por limites admissíveis. Esse tipo de perspectiva de estabilidade é analisada por um critério de estabilidade de entrada-limitada/saída-limitada BIBO (*bounded-input/bounded-output*). De uma maneira estrita, a estabilidade do sistema pode significar que as saídas do sistema convergem a um estado de equilíbrio. Esta última é chamada de estabilidade assintótica ou estabilidade no sentido de *Lyapunov*. A teoria de estabilidade tem um papel importante na análise e no projeto de sistemas de controle, pois fornece ferramentas capazes de assegurar que o sistema opere de maneira segura e com garantias de alcançar o resultado esperado mesmo na presença de variações, perturbações e incertezas durante a operação. Detalhes formais sobre estabilidade estão apresentadas no Apêndice A.

**Controlabilidade** Um sistema é dito controlável no instante  $t_0$  se for possível, usando-se um vetor de controle arbitrário, sem qualquer restrição, transferir o sistema de qualquer estado inicial  $\mathbf{x}(t_0)$  para qualquer outro estado em um intervalo de tempo finito.

**Observabilidade** Um sistema é dito observável no instante  $t_0$  se for possível determinar o estado inicial  $\mathbf{x}(t_0)$  do sistema a partir da observação da saída em um intervalo de tempo finito.

**Robusto** Um sistema de controle é robusto quando no projeto de controle é considerado que na modelagem da planta a ser controlada haverá incertezas. Essa metodologia se volta para o fato que a planta e elementos do ambiente no qual o sistema de controle irá operar não podem ser modelados com exatidão, e, podem, até mesmo, se alterar de modo imprevisível ao longo da execução. Logo, o objetivo do projeto de controle robusto é garantir desempenho desejado mesmo na presença de incertezas significativas no processo de modelagem. Pode-se dizer que um sistema de controle é robusto quando apresenta mudanças aceitáveis do desempenho predefinido devido as inexatidões ou possíveis mudanças no sistema a ser controlado (DORF; BISHOP, 2010).

## 2.2 DEFINIÇÃO DO PROBLEMA

O problema se caracteriza em determinar uma estratégia de controle capaz de gerar entradas para o sistema  $\mathbf{u}$  (comando de controle) de tal forma que a saída do sistema  $\mathbf{y}(t)$  (Eq. 2.1) siga um comportamento desejado  $\mathbf{y}_d(t)$ . Os sistemas de controle nos quais a especificação é um estado estacionário são classificados como *reguladores* automáticos. Caso a especificação seja um comportamento dinâmico das variáveis de estado do sistema é dito que é um problema de *rastreamento* (ASSUNÇÃO; ANDREA; TEIXEIRA, 2004), também denominado na literatura por problema de *seguimento* (FLORES; SILVA, 2010) ou problema de *acompanhamento* (LEONARDI; CRUZ, 2005).

Um ponto crucial para o projeto de controle é a necessidade de lidar com sistemas não-lineares e variantes no tempo. Para isso, normalmente é adequado projetar um controle com duas fases (NARENDRA; PARTHASARATHY, 1990): A primeira fase é a identificação. Nesta fase, é identificada a dinâmica do sistema. Já a segunda fase é a de controle. Nesta fase, é gerada uma ação de controle apropriada para atender as especificações de saída do sistema.

Outro fato é o conhecimento sobre a função  $\mathbf{f}$  e seus parâmetros. Não é sempre que  $\mathbf{f}$  e  $\mathbf{p}_f(t)$  são conhecidos. Para esse caso, as abordagens neurais de controles adaptativos são bastante convenientes visto que há apenas as informações de entrada e saída da função  $\mathbf{f}$ . Essas abordagens também são aplicadas adequadamente para o caso no qual  $\mathbf{f}$  é uma função vetorial não-linear que torna o projeto de controle difícil. Em ambos os casos,  $\mathbf{f}$  e  $\mathbf{p}_f(t)$  são desconsiderados e é construída uma aproximação da função  $\mathbf{f}$  para projetar o controlador.

Segundo Lawrence, Tsoi e Back (1996), há duas abordagens para conceber a função aproximada: Uma é aproximar  $\mathbf{f}$  através de um único modelo não-linear  $\tilde{\mathbf{f}}$ , conhecida como *modelagem global*. A outra abordagem é dividir o espaço em regiões locais e modelar localmente a dinâmica da função  $\mathbf{f}$ . Matematicamente, é feita uma decomposição da função  $\mathbf{f}$  em uma série de modelos locais  $\tilde{\mathbf{f}}_k$ , com  $k = 1, \dots, N$ . Dessa forma, o modelo “global” é constituído pela união de todos os modelos locais  $\tilde{\mathbf{f}}_k$  definido somente na região  $R_k$ .

$$\tilde{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t)) = \bigcup_{k=1, \dots, N} \tilde{\mathbf{f}}_k(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.3)$$

onde

$$R_i \cap R_j = \emptyset, \quad \forall i \neq j \quad (2.4)$$

## 2.3 MÉTODOS DE CONTROLE

A seguir, será descrita sucintamente cada abordagens de controle mencionada. Também, é apresentado resumidamente o controle proporcional integral derivativo (PID) e estratégias de controle aplicadas a sistemas não-lineares.

### 2.3.1 Controlador Proporcional Integral Derivativo

Controle direto e retroalimentado com ganho fixo (*fixed-gain feedback and feed-forward control*) é o controle mais simples e mais usado nos diferentes problemas de automação. Acredita-se que mais de 90% dos processos de controle atuais são do tipo proporcional integral derivativo (PID) (ÅSTRÖM; HÄGGLUND, 2001). É comum encontrar essa classe de controle na forma *proporcional mais derivativo*, *proporcional mais integrativo* ou *proporcional mais integral mais derivativo*. Esta última forma é apresentada na Equação 2.5.

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t)dt + K_d \cdot \frac{d}{dt}e(t) \quad (2.5)$$

onde  $u(t)$  é o sinal de comando de controle,  $e(t)$  é o sinal de erro atuante entre o valor de referência (a saída desejada) e a saída do sistema e  $K_p$  é denominado sensibilidade proporcional ou ganho. Já  $K_i = \frac{K_p}{T_i}$  e  $K_d = K_p \cdot T_d$  onde  $T_i$  e  $T_d$  representam os tempos integrativo e derivativo, respectivamente.

Uma das principais vantagens ligadas a essa classe de controle é a simplicidade. Isso reflete na sua fácil implementação e na sua capacidade de ser utilizado em esquemas sofisticados de controle.

### 2.3.2 Controle Não-Linear

Nesta seção, são apresentadas duas abordagens de controle não-linear. Ambas são aplicáveis quando há uma modelagem matemática que captura o comportamento dinâmico do sistema.

#### 2.3.2.1 Método de Linearização por Realimentação

A primeira estratégia de controle não-linear a ser apresentada é *método de linearização por realimentação*. Essa técnica consiste em tornar um sistema não-linear em linear a partir da realimentação e de uma lei de controle adequada. Porém, pode levar a um projeto de controle ineficiente já que controladores de linearização por realimentação podem gerar um grande esforço desnecessário para cancelar a não-linearidade (HADDAD; CHELLABOINA, 2011).

Considere um sistema não-linear com a seguinte forma:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad t \geq 0, \quad (2.6)$$



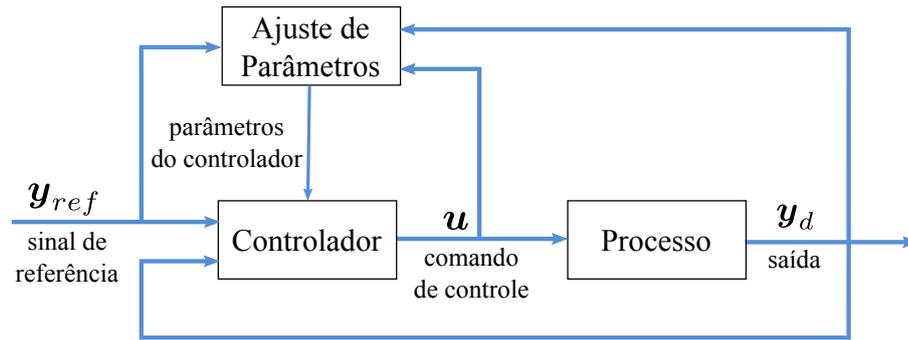


Figura 1 – Esquema de um controle adaptativo, adaptado de (ÅSTRÖM; WITTENMARK, 1995).

$x_{2d}$ . Esse procedimento é repetido, até, finalmente, escolher um  $u(t)$  tal que  $x_m$  rastreie  $x_{md}$ .

### 2.3.3 Controle Adaptativo

Usualmente, as características do processo ou da planta que determinam a dinâmica do sistema, isto é, a função  $f$  da Equação 2.1, não são constantes. Isso pode ocorrer por diversos motivos. Alguns deles são: deterioração dos componentes ao longo do tempo, variações nas condições do ambiente, e, até mesmo, a própria natureza dos componentes do sistema.

Normalmente, os efeitos provocados por pequenas variações nas características dinâmicas podem ser atenuados por um sistema de controle com realimentação. Contudo, para variações significativas nas características do sistema, o controle deve possuir a habilidade de adaptação. Esta habilidade consiste na capacidade de se auto-ajustar ou automodificar para se adequar às novas características do ambiente ou da estrutura.

Segundo Åström e Wittenmark (1995), um controlador adaptativo é um controlador com parâmetros modificáveis através de mecanismo de ajuste. A Figura 1 apresenta um diagrama representando o controle adaptativo. Pode-se notar que em um sistema de controle adaptativo há a necessidade do monitoramento a todo instante do comportamento dinâmico do processo para que se possa modificar corretamente os parâmetros do controle. O conceito de controle adaptativo é bastante atrativo, pois pode responder a variações ambientais e a erros ou incertezas do projeto do controlador. Segundo Ogata (2011), também pode se adaptar para compensar falhas de componentes, aumentando, assim, a confiabilidade no sistema.

### 2.3.4 Controle Ótimo

Controle ótimo é um importante componente na teoria de controle moderno. Foi inicialmente desenvolvido na década de 1960 e seu início está ligado a era da corrida espacial, dos computadores e dos robôs (LEWIS, 2010). Pode-se afirmar que é um ramo da teoria

de controle moderno que lida com o projeto de controle para sistemas dinâmicos através da otimização de um índice de performance que, por sua vez, depende das variáveis do sistema. As aplicações do controle ótimo foram bastante significativas no controle de aplicações espaciais, aeroespaciais e militares.

Tipicamente, para o problema de controle ótimo, deve-se definir os seguintes elementos:

- Uma modelagem de equações de estados;
- Condições iniciais do sistema;
- Conjunto de objetivos definidos;

O objetivo central na confecção desta abordagem é encontrar uma lei de controle viável tal que o sistema parta das condições atuais do sistema e atinja o estado alvo otimizando um índice de desempenho.

O controle ótimo consiste de uma extensão do cálculo de variações. Pois, esse problema de otimização está definido em espaços de funções de dimensões infinitas e tem a função objetivo definida por um operador de integração. Há duas poderosas ferramentas para resolver os problemas de conjunto de restrições: princípio do máximo (ou mínimo) de Pontryagin e programação dinâmica de Bellman (LOCATELLI, 2001).

O controle ótimo é muito bem adaptado para problemas aeroespaciais e aplicações militares. Aplicações dessa abordagem apareceram no controle de pouso na Lua de uma nave espacial (LIU; DUAN; TEO, 2008) e no controle de voo de um foguete (TEWARI, 2011). No Apêndice B, é apresentado um controle ótimo que foi implementado e utilizado neste doutorado.

### 2.3.5 Controle Inteligente

Um campo do controle moderno interessante e estudado atualmente é o controle inteligente. Há mais de uma definição para esta classe de controle. Objetivamente, pode-se defini-lo como um paradigma de controle que propõe empregar uma ou mais técnicas de inteligência computacional no projeto do sistema de controle (PASSINO, 2010). Técnicas de computação inteligente podem ser empregadas tanto na modelagem da planta ou identificação do modelo de planta quanto no controlador que gerar as ações de controle. Ainda mais, o controlador inteligente pode ser projetado sem a necessidade de empregar um modelo ou um identificador do sistema. A classe desses controles inteligentes é denominada de controle dirigido por dados (HOU; WANG, 2013). Em geral, os controles inteligentes podem apresentar as seguintes características (ALBUS, 1990):

- Aprendizagem automática;
- Autorreconfiguração;

- Mecanismos de inferência;
- Elaboração e uso de planejamentos;
- Mecanismos de tomada de decisão;
- Capacidade de extrair as informações mais relevantes de dados sem prévio processamento, isto é, de dados brutos sem estarem classificados ou estruturados e até mesmo com ruído.

Todas essas características dão a esse tipo de controle um alto grau de autonomia. Os sistemas de controle inteligente podem ter a capacidade de detectar alterações na planta ou até mesmo no ambiente. Podem tomar decisões corretas mesmo na falta de parte da informação ou proveniente de múltiplos sensores. E são capazes de se reconfigurarem na situação em que apresentem falhas internas. Os controles inteligentes são comumente divididos em algumas subclasses. Algumas delas são: controle nebuloso (*fuzzy*), controle especialista (*expert*), controle evolucionário e controle neural (PASSINO, 2010).

### 2.3.6 Controle Neural

Segundo Blondin, Sáez e Pardalos (2019), a medida que deseja-se atuar em sistema cada vez mais complexo, o modelo matemático passar a ser mais difícil de construir. Em algumas aplicações, as informações *a priori* do modelo não são suficientemente precisas para projetar um controle fixo que satisfaça todas as especificações do projeto (BLONDIN; SÁEZ; PARDALOS, 2019; POLYCARPOU; FARRELL, 2010). Nesse caso, uma abordagem é projetar um controlador capaz de se ajustar durante a execução para melhorar seu desempenho à medida que os dados de entrada e saída fornecem mais informações sobre incertezas do modelo do sistema.

O controle adaptativo, apresentado na Subseção 2.3.3, foi desenvolvido para essa classe de problemas e as redes neurais podem ser adequadas a um projeto de um controlador adaptativo. Isso pode ser justificado pelas características dos modelos conexionistas (HUNT et al., 1992). Os modelos neurais são aproximadores capazes de operar sobre ambiente impreciso e com parâmetros variantes no tempo. Outro ponto atrativo, na utilização de metodologias usando redes neurais, é a capacidade de identificação de mudanças na dinâmica em tempo de execução e adapta para funcionar sobre esse novo comportamento do sistema. Os modelos neurais são capazes de aprender as informações relevantes do ambiente. Então, o termo controle neural é o nome dado na literatura para os sistemas de controle que empregam uma rede neural no projeto de controle.

Os esquemas de controle neural podem ser classificados de maneira geral como adaptativo direto ou indireto. Ambos seguem noções básicas oriundas da estratégia de controle adaptativo clássico. O esquema adaptativo indireto usa um modelo neural explícito da planta para ajustar os parâmetros do controlador. Enquanto o esquema direto usa um

controlador parametrizado em termos de uma rede neural e a regra de atualização dos pesos, também chamada de lei de adaptação, é derivada de tal forma que a combinação controle neural e planta em malha fechada seja estável.

Antes de explicar as abordagens de controle neural indireta e direta, é importante definir três termos inerentes ao campo dos controle neurais, são eles: *adaptação*, *aprendizagem* e *auto-organização*. Na literatura, diferentes autores podem usar diferentes significados para esses termos. Aqui, os significados serão os definidos por Polycarpou e Farrell (2010). Segundo esses autores, adaptação refere-se às mudanças temporais. A aprendizagem significa que o método retém informações como variáveis mesuráveis aplicadas em uma função. Portanto, a aprendizagem tem uma conotação espacial enquanto a adaptação tem um sentido de efeito temporal. Já a auto-organização é um termo usado para designar a característica que uma estratégia de controle neural é capaz de modificar a estrutura de aproximação da função durante a fase de operação (ZHAO; FARRELL, 2007).

### 2.3.6.1 Controle Adaptativo Indireto usando Redes Neurais

O arcabouço conceitual do controle adaptativo indireto para sistemas de tempo contínuo é apresentado na Figura 2.

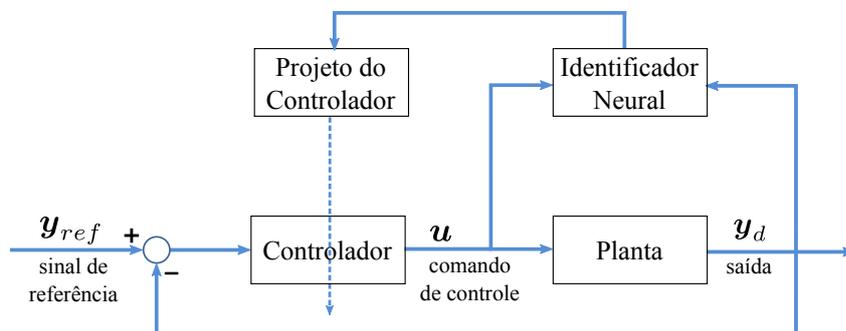


Figura 2 – Estrutura de um controle adaptativo neural indireto.

O comportamento da planta é capturado usando uma rede neural. Assumindo que a rede neural se comporta suficientemente igual à planta, o controlador é projetado de tal forma que estabilize o modelo neural da planta usando a malha realimentação partindo da planta real.

Sendo assim, o controle adaptativo indireto é implementado seguindo estes dois estágios:

- **Identificação do sistema:** O modelo neural é explicitamente treinado para se comportar suficientemente igual ao sistema a ser controlado;
- **Controle:** O controlador é projetado para que o modelo explícito identificado no passo 1 tenha o comportamento desejado.

### 2.3.6.2 Controle Adaptativo Direto usando Redes Neurais

Em certos cenários, não há conhecimento exato da não-linearidade do sistema e isso inviabiliza a aplicação de técnicas como a *linearização por realimentação*. A abordagem adaptativa direta é apropriada para tais cenários. Ela consiste de usar um modelo neural capaz de estimar a não-linearidade do sistema e, através do modelo treinado, construir uma estratégia de controle capaz de linearizar e controlar o sistema adequadamente, assim como seria feito através da técnica linearização por realimentação.

No esquema adaptativo direto, o controlador é projetado em uma rede neural e a lei de ajuste dos pesos é derivada de tal forma que a combinação do controlador neural e a planta na configuração de malha fechada tenha erro de rastreamento convergindo para zero. O arcabouço conceitual do controle adaptativo direto para sistemas de tempo contínuo é apresentado na Figura 3.

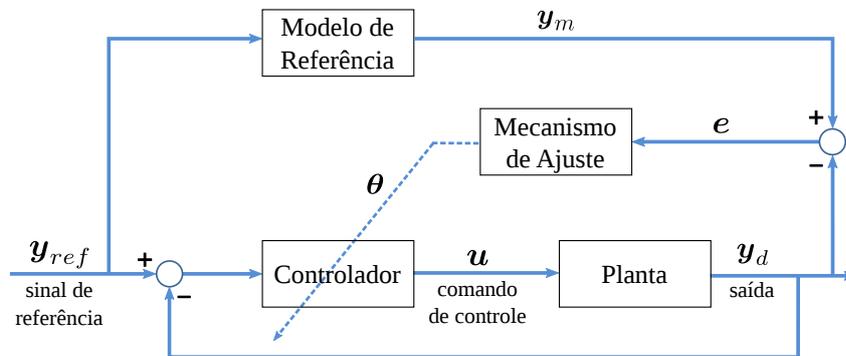


Figura 3 – Estrutura de um controle adaptativo direto com modelo de referência.

## 2.4 CONCLUSÃO

Este capítulo teve três objetivos principais: (1) definir formalmente o problema de controle no contexto de sistemas não-lineares variantes no tempo, (2) apresentar brevemente conceitos fundamentais do contexto de sistemas de controle, e (3) sumarizar estratégias de controle fundamentadas na literatura importantes no desenvolvimento deste trabalho de doutorado. Os conceitos, as ideias e as estratégias serão empregadas nas estratégias de controle propostas neste doutorado.

### 3 ESTRATÉGIAS DE CONTROLE BASEADO EM SOM

Neste capítulo, é apresentada uma análise da literatura existente sobre sistemas de controle baseados no mapa auto-organizável de Kohonen (KOHONEN, 1982; KOHONEN, 1990) ou algum mapa variante dele. Primeiramente, é definido o termo controle baseado em SOM. Logo em seguida, é descrito o SOM e são tratadas as principais características e as limitações das estratégias de controle baseado em SOM. Depois, são descritos sucintamente alguns sistemas de controle baseado em SOM. Por último, uma conclusão com análise dos modelos existentes servirão de base para este trabalho de doutorado.

#### 3.1 CONTROLE BASEADO EM SOM

Os trabalhos aqui apresentados foram concebidos empregando mapas auto-organizáveis. Logo, este trabalho adotou o termo *controle baseado em mapa auto-organizável* para designar as estratégias de controle inteligente baseado no mapa auto-organizável de Kohonen ou em qualquer mapa auto-organizável derivado do SOM original.

##### 3.1.1 Mapa Auto-organizável - SOM

O SOM também chamado de mapa auto-organizável de característica (*self-organizing feature map* - SOFM) foi desenvolvido por Teuvo Kohonen na década de 1980 (KOHONEN, 1982) e tem sido usado com sucesso para resolver um grande número de aplicações em diversos campos, tais como agrupamento de dados, reconhecimento de padrões, processamento de sinais, otimização, controle de processos e controle motor (KOHONEN, 2001b; KOHONEN, 2001a; KOHONEN, 2013). O SOM é um modelo de rede neural competitivo com a propriedade de reduzir padrões de alta dimensionalidade em um mapa de dimensão inferior. Em geral, o mapa é constituído por uma grade uni ou bidimensional de nodos (Figura 4). O mapa topológico é automaticamente organizado através de um processo cíclico de comparação dos padrões de entrada com o vetor de pesos de cada nodo e ajuste do nodo mais semelhante e de sua vizinhança. Dessa forma, o algoritmo de aprendizagem organiza os nodos em uma malha onde cada região desta malha atua como um “agrupador de características” dos dados.

O SOM funciona basicamente da seguinte forma: quando um padrão de entrada  $\xi$  é apresentado, a rede procura o *nodo com melhor correspondência*, isto é, o nodo cujo vetor de pesos a ele associado é o mais próximo do padrão de entrada segundo uma medida de distância. Esse nodo é chamado de *vencedor*. Durante o processo treinamento, a rede aumenta a semelhança do nodo vencedor e de seus vizinhos ao padrão  $\xi$  através de uma regra de ajuste que está especificada na Equação 3.1. O processo termina quando as modificações no mapa são insignificantes. Em síntese, a rede constrói um mapa topológico onde

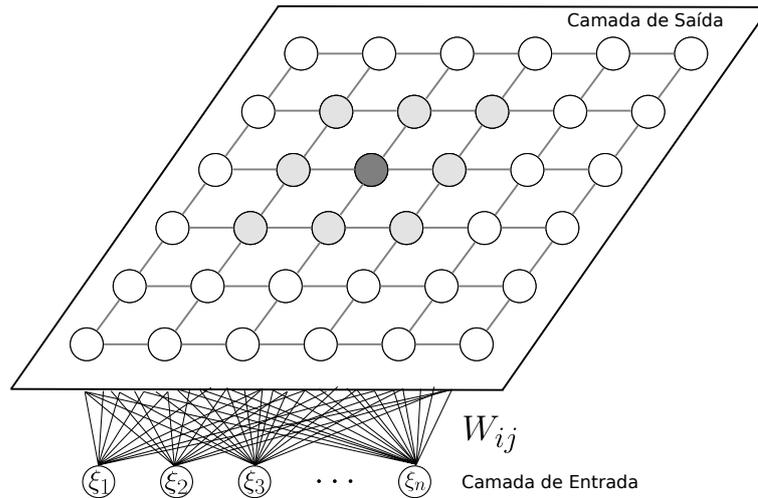


Figura 4 – Exemplo de topologia usada para SOM. As unidades da camada de entrada são conectadas a todos os nodos da camada de saída.

nodos que estão topologicamente próximos respondem de forma semelhante a padrões de entrada parecidos (KOHONEN, 1982).

A regra de ajuste dos pesos tem a seguinte forma:

$$\mathbf{w}_k[t + 1] = \mathbf{w}_k[t] + \alpha[t] h_{s,k}[t] (\boldsymbol{\xi}[t] - \mathbf{w}_k[t]) \quad (3.1)$$

no qual  $\mathbf{w}_k[t]$  é o vetor de pesos do nodo  $n_k$  no instante  $t$ . Normalmente, o vetor inicial  $\mathbf{w}_k[0]$  é um vetor de valores aleatórios.  $\boldsymbol{\xi}[t]$  é o padrão de entrada apresentado no instante  $t$ .  $\alpha(t)$  é a taxa de aprendizagem representada por uma função que decresce monotonicamente à medida que  $t$  cresce. Outro ponto importante é que  $0 < \alpha[t] \leq 1$  e  $\alpha[t] \rightarrow 0$  quando  $t \rightarrow \infty$ .  $h_{s,k}[t]$  é a função vizinhança cujo valor depende do nodo vencedor  $n_s$ , do nodo atual  $n_k$  e do instante  $t$ . A função  $h$  tem valor máximo unitário e mínimo nulo e decresce à medida que se distancia do nó vencedor. Para assegurar a convergência, para  $n_s \neq n_k$  tem-se  $h_{s,k}[t] \rightarrow 0$  quando  $t \rightarrow \infty$ . Normalmente, é escolhida uma função gaussiana para representar  $h_{s,k}[t]$ :

$$h_{s,k}[t] = \exp\left(-\frac{\|\mathbf{r}_k[t] - \mathbf{r}_s[t]\|^2}{2\sigma^2[t]}\right) \quad (3.2)$$

no qual  $\mathbf{r}_k[t]$  e  $\mathbf{r}_s[t]$  são as coordenadas dos nodos  $n_k$  e  $n_s$  no gradil, respectivamente.  $\sigma[t]$  define a região de vizinhança. Assim como  $\alpha[t]$ ,  $\sigma \rightarrow 0$  quando  $t \rightarrow \infty$ . As funções  $\alpha[t]$  e  $\sigma[t]$  podem ser modeladas das seguintes formas:

$$\alpha[t] = \alpha_\infty + (\alpha_0 - \alpha_\infty) \exp\left[-\frac{t}{\tau_\alpha}\right] \quad (3.3)$$

$$\sigma[t] = \sigma_\infty + (\sigma_0 - \sigma_\infty) \exp\left[-\frac{t}{\tau_\sigma}\right] \quad (3.4)$$

### 3.2 CARACTERÍSTICAS DAS ESTRATÉGIAS DE CONTROLE BASEADO EM SOM

Estratégias de controle baseados no mapa de Kohonen são soluções úteis para sistemas de controle visto que o SOM combina as seguintes propriedades:

**Aprendizagem Não-supervisionada** A primeira característica sobre essa classe de controle está no seu paradigma de aprendizagem. Esse tipo de controle herda, normalmente, o paradigma de aprendizagem da rede SOM, que é chamado de aprendizagem não-supervisionada. Por sua vez, na aprendizagem não-supervisionada, a rede neural não necessita de um “professor” externo para informar as respostas corretas. A rede SOM é capaz de se *auto-organizar* para construir representações internas dos padrões de entrada sem um “supervisor”. Esta habilidade de *auto-organização* pode simplificar o controlador e, conseqüentemente, implicar em uma carga menor na programação de um sistema de controle. Conseqüentemente, o custo da implementação do controle pode ser reduzido (BARRETO; ARAÚJO; RITTER, 2003).

**Aprendizagem Incremental** Outro ponto sobre modelos não-supervisionados é que eles podem ser confeccionados para uma aprendizagem incremental. Desta forma, o sistema de controle pode, ao longo da execução, aprimorar seu desempenho. Assim como um operador humano, o sistema é mais abrangentemente controlado a medida que o controlador vai adquirindo mais experiência (BARRETO; ARAÚJO; RITTER, 2003).

**Modelo Baseado em Dados** O SOM é uma rede aplicável sem muitas suposições e pode ser aplicável universalmente a partir de dados do sistema. Isso implica que um controlador baseado na rede SOM pode controlar um sistema com modelo desconhecido somente usando dados mensuráveis do sistema (RITTER; MARTINETZ; SCHULTEN, 1992).

**Modelo Não-linear** O SOM tem a capacidade de modelar relações não-lineares em simples mapeamentos espaciais. Visto que seu objetivo principal é construir uma transformação não-linear dos padrões de entrada de dimensão arbitrária em um mapa discreto normalmente unidimensional ou bidimensional (HAYKIN, 2009). Ele é capaz de construir a transformação adaptativamente e com preservação da ordenação topológica dos padrões de entrada.

**Resolução de Ambiguidades** A rede SOM é adequada para problemas com grande volume de dados e com redundância de informações, propriedade está presente em alguns sistemas de controle como sistemas de controle robóticos (RITTER; MARTINETZ; SCHULTEN, 1992; HÜLSEN, 2007). Esta característica pode ser útil para resolver ambiguidades de possíveis mapeamentos.

**Interpretabilidade** A rede SOM pode ter um certo grau de interpretabilidade, pois cada região do mapa aprende a responder a um determinado padrão de entrada (WALTER, 1997).

**Poder de Generalização** Em virtude do poder de generalização do SOM, sistemas de controle baseados nessa rede podem ter a capacidade de se adaptar a mudanças significativas nos parâmetros do sistema, a alterações do ambiente do sistema ou distúrbios externos. Essa particularidade é também importante em diversos sistemas de controle. O controle pode ser capaz de operar o sistema, mesmo para parâmetros não apresentados na etapa de treinamento (MINATOHARA; FURUKAWA, 2011).

Uma limitação presente nesta classe de controle é uma perda na acurácia devido a uma discretização do espaço de entrada e de saída. Porém, essa perda na acurácia pode ser superada ao menos de duas maneiras: (1) pode-se ampliar o conjunto de treinamento aumentando a taxa de amostragem e associar a um número maior de protótipos na rede (2) ou pode-se projetar o controle a partir de um SOM estendido ou uma rede derivada do SOM que possui a característica de saída contínua ou contínua por partes. Para essa segunda alternativa, mais detalhadamente, pode ser utilizada:

- Alguma técnica de interpolação apropriada para rede SOM (GÖPPERT; ROSENSTIEL, 1995; GÖPPERT; ROSENSTIEL, 1997; HÜLSEN, 2007);
- Uma camada extra capaz de gerar um saída contínua na rede SOM (SBARBARO; BASSI, 1995; PRINCIPE; WANG; MOTTER, 1998);
- Uma rede derivada do SOM com capacidade de mapeamento contínuo, ou contínuo por partes (WALTER; RITTER, 1996; FLENTGE, 2006).

Outra limitação em estratégias de controle baseados na rede SOM é ter um mapa de tamanho fixo e conexões pré-definidas. Sendo assim, a rede pode não se adequar bem ao espaço de entrada do sistema ou pode ser necessário várias tentativa para encontrar o valor adequado do número de nodos e da topologia da vizinhança. A solução deste ponto está em utilizar as redes SOMs que tem a capacidade de ajustar o número de nodos e a topologia a medida que é necessário (ARAÚJO; RÊGO, 2013). Tais redes são conhecidas por mapas auto-organizáveis com estrutura variante no tempo (SOM-TVS).

### 3.3 TRABALHOS RELACIONADOS

Foi feito um levantamento da literatura de sistemas de controle inteligentes baseado em SOM. Foi observado que alguns trabalhos aplicaram SOM em uma abordagem de múltiplos modelos. Então, primeiramente, são descritos sucintamente sistemas de controle na abordagem múltiplos modelos baseado em SOM. Logo em seguida, são apresentados trabalhos da literatura sob outras abordagens.

### 3.3.1 SOM Aplicado em Controle Adaptativo baseado em Múltiplos Modelos

Nesta forma de abordagem, os projetistas normalmente utilizam a rede SOM para modelar a dinâmica do sistema. O SOM pode ter o papel de dividir e escolher o espaço de operação de cada controle local. Normalmente, cada nodo tem um modelo local que aproxima o comportamento do sistema para um determinado espaço de operação e, por sua vez, esse modelo local é usado para definir o controlador local. Nestes casos, o SOM identifica o comportamento do sistema e seleciona o modelo local que mais se aproxima da dinâmica do sistema, e conseqüentemente, pode ser também responsável pela comutação dos controladores locais. Trabalhos com essa abordagem estão relacionados à arquitetura de controle de múltiplos modelos (NARENDRA; PARTHASARATHY, 1990; NARENDRA; BALAKRISHNAN, 1997). Alguns trabalhos estão descritos nos próximos parágrafos.

Sbarbaro e Bassi (1995) apresentaram um controle baseado no SOM. O papel da rede era particionar as regiões de operação de um processo químico não-linear e organizar modelos locais capazes de prever o comportamento do sistema. A partir desses modelos locais, controladores locais foram projetados para determinar o comando de controle para o sistema.

Principe, Wang e Motter (1998) e Motter (2000) propuseram uma estratégia de controle preditivo de múltiplos modelos locais através de uma modelagem dinâmica local com várias redes SOMs. O modelo é capaz de comutar os múltiplos modelos para estimar a resposta do sistema com acurácia. Inspirado na arquitetura de múltiplos modelos, cada rede SOM foi aplicada para desempenhar os seguintes papéis: (1) Particionar as regiões de operação para obter uma modelagem da dinâmica global do sistema sobre uma condição específica de alimentação do sistema; (2) derivar os modelos lineares locais a partir dos pesos da rede e indiretamente organizar topologicamente os modelos locais; e (3) selecionar o modelo local mais apropriado para simular a aplicação de uma sequência de controles candidatos. A rede SOM juntamente com os modelos locais associados a cada nodo do mapa podem ser interpretados como um modelo de comportamento entrada/saída da planta para uma classe fixa de entradas. Esse controle foi implementado em *hardware* e aplicado ao problema não-linear de identificação e controle de túnel de vento durante testes aerodinâmicos em três diferentes aeronaves. Essa estratégia de controle obteve resultados que satisfizeram as especificações do projeto, mas, como próprios autores concluem, esse controle não é de proposição geral devido ao fato que as ações de controle candidatas são específicas para o problema de controle de túnel de vento.

Thampi et al. (2002) atacaram o problema de modelagem e controle de sistemas discretos não-lineares utilizando mapas de Kohonen. O método proposto foi esquematizado em múltiplos módulos de identificadores e controladores. O papel do SOM consistiu em modelar a dinâmica através do seu poder de representação dos dados de entrada. Cho et al. (2006) propuseram um controle no esquema de modelagem linear local através de um Mapa Auto-Organizável. A função da rede era identificar a região de operação do

sistema e selecionar adequadamente o controle linear para controlar o sistema. Cho et al. (2007) também trabalharam em um controle para uma classe geral de sistemas não-lineares de tempo discreto utilizando a rede SOM para dividir o espaço de estados em regiões de operação. Em cada região, há um controlador local por modo quase-deslizante (*local quasi-sliding mode controller* - QSMC).

Minatohara e Furukawa (2011) trabalharam em um controlador que atendesse a dois requisitos importantes para sistema de controle complexos: (1) resposta rápida para mudanças repentinas no objeto controlado e (2) aquisição de um comportamento geral a partir de um pequeno número de padrões de treinamento. Para isso, propuseram o controlador adaptativo auto-organizável (*self-organizing adaptive controller* - SOAC). Esse controlador foi projetado sob a arquitetura de múltiplos módulos para atender ao primeiro requisito. Por sua vez, o segundo requisito foi atendido através do poder de generalização da rede SOM. Mais precisamente nesse ponto, o SOAC foi elaborado sob uma extensão do SOM chamada rede modular SOM (mnSOM) (TOKUNAGA; FURUKAWA, 2009). Cada módulo consiste de um par preditor/controlador. O preditor pode ser visto como um identificador do comportamento da planta. Já o controle a ser aplicado é determinado a cada instante e é escolhido através do preditor que mais se aproxima da planta.

Dayuan, Zheng e Hai (2013) projetaram um controle auto-organizável no esquema de múltiplos módulos para manobra de veículo aéreo não-tripulado. O mapa auto-organizável foi responsável por dividir o espaço de operação. Em cada região, havia um modelo linear aproximado da cinemática não-linear do veículo. Por sua vez, controladores locais foram projetados para cada modelo linear local. Também, foi proposto um comutador adaptativo fundamentado no SOM. Os resultados simulados sugeriram um aumento na velocidade de resposta e na acurácia do controle.

Kiumarsi, Lewis e Levine (2015) atacaram o problema de controle de sistemas discretos variantes no tempo através de uma abordagem de múltiplos modelos com mapa auto-organizável. No caso, inspirados em modelos neuro-cognitivos, propuseram o mapa auto-organizável adaptativo (ASOM). O ASOM foi aplicado para detectar mudanças na dinâmica do sistema, e adicionava novos modelos locais à medida que novos comportamentos do sistema eram detectados. O ASOM também alimentava uma estrutura de aproximação de função com um sinal de responsabilidade. Essa estrutura de aproximação é adaptada de maneira *online* através de nova técnica de aprendizagem por reforço e ponderada localmente a partir do sinal de responsabilidade. A ação de controle é gerada a partir da sequência temporal dos estados do sistema e dos comandos de controle. Este sistema de controle foi simulado no controle discreto do sistema massa-mola-amortecedor.

Zhao et al. (2016) propuseram um sistema de controle para um manipulador flexível piezoelétrico-pneumático. Esse sistema híbrido foi decomposto em dois subsistemas: o subsistema piezoelétrico e o subsistema pneumático. Para o subsistema pneumático, eles investigaram um controle inverso em múltiplos modelos baseado no SOM para controle

da vibração. O SOM foi utilizado para dividir o espaço de saída em regiões locais. O conjunto de treinamento de cada região local foi usado para estimar um modelo linear local do subsistema pneumático. Mais especificamente, o método dos mínimos quadrados ponderados foi usado para obter o modelo local. No caso, os ponderadores foram calculados a partir da distância entre o nodo do modelo local e nodo representante do conjunto dos dados de treinamento. A partir dos modelos lineares locais, controladores inversos locais são construídos. Durante a fase de operação, uma janela de tempo de tamanho 3 no sinal de saída do subsistema é usada como entrada do SOM. O processo de competição seleciona um controlador inverso local. O controlador inverso vencedor mais um controlador proporcional em realimentação negativa são aplicados para gerar o sinal de controle apropriado. O sistema de controle foi avaliado para um manipulador flexível real e os resultados sugerem que o controlador foi efetivo em suspender a vibração.

### 3.3.2 Outros Trabalhos Relacionados

Delgado (2000) apresentou uma rede SOM com modificações aplicável ao contexto de problemas de controle não-linear. A principal modificação foi a adição de um camada extra de ganhos, onde cada ganho está associado a um nodo da rede SOM. A entrada da rede é o vetor composto pelo erro de rastreamento e pela derivada do erro. A saída da rede é calculada a partir da soma da ativação dos nodos da rede ponderada pelos ganhos da última camada. Delgado também propôs uma aprendizagem híbrida. A camada da rede SOM é treinada pelos ajustes tradicionais da rede SOM e a camada de ganhos é adaptada através de um treinamento estocástico inspirado na quimiotaxia (*chemotaxis*). O controlador é proposto em duas abordagens: linearização pela realimentação e controle ótimo. Para avaliar o desempenho, esse controlador foi simulado em um manipulador de único elo.

Barreto e Araújo (2004) apresentaram uma técnica de modelagem geral chamada memória associativa temporal por quantização vetorial (*vector-quantized temporal associative memory* - VQTAM) para identificação e controle de sistemas dinâmicos. Liu, Xiao e Yu (2008) aplicaram o conceito de modelagem entrada-saída do VQTAM sobre um mapa auto-organizável intervalar. Esse mapa considera os pesos nos nodos como intervalos numéricos, e aplicação do mapa proposto foi na implementação de um controle inverso. Barreto, Araújo e Ritter (2003) também apresentaram uma revisão de redes SOMs aplicadas em modelagem adaptativas e controle de manipuladores robóticos. Barreto, Araújo e Ritter (2003) argumentam que as redes SOMs podem ser mais simples em termos função de aproximação e projeto do controle que redes tradicionais como redes perceptron de multicamadas (MLP) e de função de base radial (RBF).

Hülßen (2007) propôs uma variante da rede SOM com requisitos adequados para aprendizagem do modelo inverso do sistema. Ele apresentou o mapa auto-organizável interpolado localmente (*self-organising locally interpolating map* - SOLIM). O SOLIM é uma rede

---

inspirada no SOM com capacidade de realizar interpolação e extrapolações localmente. Esta interpolação é uma extensão da interpolação do simplex, que é contínua, local e interpretável. O SOLIM foi simulado para validação de propriedades importantes dentro do contexto de modelagem inversa de sistema e foi avaliado em um sistema de controle de micro-robôs.

Laffamme, Slotine e Connor (2011) apresentaram um controle neural de wavelet para operar em sistemas incertos com atuadores semi-ativo. Esta condição aparece em cenário de controle de estrutura civil de larga escala como pontes e viadutos. Esse controle herda a característica da auto-organização do SOM. Em outro trabalho, Laffamme, Slotine e Connor (2012) também propuseram um novo modelo de rede neural inspirado nos mapas auto-organizáveis para controle estrutural. O ponto fundamental desse trabalho esteve no algoritmo de auto-organização da entrada (algoritmo self-organizing input - SOI). Ele foi responsável por ajustar a camada do espaço de entrada. Já a única camada escondida chamada de rede neural de wavelet (*wavelet neural network* - WNN) foi organizada pelo Mapa Auto-Organizável.

### 3.4 CONCLUSÃO

As abordagens de controle usando SOM mais importantes para o desenvolvimento deste trabalho estão sintetizadas nas Tabelas 1 e 2. Foi observado nos trabalhos principalmente o papel da rede SOM no projeto de controle. Também, levou em conta o contexto do problema de controle e os detalhes da planta usados para avaliar. Foi feito um levantamento das modificações e extensões na rede para o contexto de problemas de controle. Isto inclui melhorias no processo de treinamento, extensões no vetor de entrada, adição de camada extra, métodos de interpolação e extrapolação, ou aplicação de uma variante da rede SOM. Foi analisado o desenho de controle investigando os esquemas de controle aplicados, os detalhes dos modelos matemáticos e os controladores auxiliares utilizados. Por fim, foi constatado se há garantias de estabilidade nas estratégias de controle propostas na literatura.

Tabela 1 – Resumo dos trabalhos aplicando SOM para problemas de controle usando a abordagem de múltiplos modelos.

Trabalho	Papel do SOM	Problema	Sistema	Prova	Modificação	Modelos
Local dynamic modeling with SOM and applications to nonlinear system identification and control (PRINCIPE; WANG; MOTTER, 1998)	(1) Particionar espaço de operação; (2) Derivar modelos locais; (3) Organizar modelos locais; (4) Selecionar modelo local	Não-linear; Discreto	Túnel de vento	Não	Múltiplos SOMs; Adição de camada extra; Modificação no treino; camada de entrada - concatenação temporal	Modelos lineares; Comandos de controle pré-definidos
Adaptive inverse control using SOM based multiple models (THAMPI et al., 2002)	(1) Particionar espaço de operação; (2) Organizar modelos locais; (3) Selecionar modelo local; (4) Comutar controladores locais	Não-linear; Discreto	Simulação de sistemas discretos gerais	Não	Adição de camada extra; camada de entrada - concatenação temporal da resposta e controle	Modelos lineares; Controladores locais adaptativos
Modeling and inverse controller design for an unmanned aerial vehicle based on SOM (CHO et al., 2006)	(1) Particionar espaço de operação; (2) Organizar modelos locais; (3) Selecionar modelo local; (4) Comutar controladores locais	Não-linear; Contínuo;	Simulação de veículo aéreo não tripulado	Não	Adição de camada extra; camada de entrada - concatenação temporal da resposta	Modelos lineares; Controle Inverso
Quasi-sliding mode control strategy based on multiple-linear models (CHO et al., 2007)	(1) Particionar espaço de operação; (2) Organizar modelos locais; (3) Selecionar modelo local; (4) Comutar controladores locais	Não-linear; Discreto	Simulação de sistemas discretos gerais	Sim	Adição de camada extra; camada de entrada - concatenação temporal da resposta	Controle modo <i>Quasi-sliding</i>
The self-organizing adaptive controller (MINATOHARA; FURUKAWA, 2011)	(1) Organizar modelos locais; (3) Ponderar controladores locais; (4) Estimar parâmetros do sistema	Não-linear; Contínuo; Variante no Tempo	Simulação dos sistemas massa mola amortecedor e pêndulo invertido	Não	rede modular SOM	LQR, PDA;
Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure (KIUMARSI; LEWIS; LEVINE, 2015)	(1) Particionar espaço de saída; (2) Organizar modelos locais; (3) Comutar controladores locais	Não-linear; Discreto	sistema massa mola amortecedor	Não	Adição de camada extra de saída; sinal de responsabilidade	Aprendizagem por reforço
Vibration control of a pneumatic driven piezoelectric flexible manipulator using SOM based multiple models (ZHAO et al., 2016)	(1) Particionar espaço de saída; (2) Organizar modelos locais; (3) Comutar controladores locais	Não-linear; Contínuo	Manipulador flexível	Não	Adição de camada extra; camada de entrada - concatenação temporal da resposta	Modelos lineares; Controle Inverso

Tabela 2 – Resumo dos outros trabalhos aplicando SOM para problemas de controle.

Trabalho	Papel do SOM	Problema	Sistema	Prova	Modificação	Controle	Modelos
Control of Nonlinear Systems Using a Self-Organising Neural Network (DELGADO, 2000)	(1) Mapear o erro e a deriva do erro em ações ótimas de controle	Não-linear; Contínuo	Simulação de manipulador de único elo	Sim	Adição de camada extra; aprendizagem híbrida (tradicional + apr. estocástica)	Controle ótimo; linearização pela realimentação	Ganhos associado a cada nodo;
Identification and Control of Dynamical System Using the SOM (BARRETO; ARAÚJO, 2004)	(1) mapeamento temporal	Não-linear; Contínuo	Simulação de um sistema não-linear	Não	Concatenação temporal da entrada	Controle preditivo	modelo auto-regressivo linear
Self-Organising Locally Interpolating Maps in Control Engineering (SOLIM) (HÜLSEN, 2007)	(1) Aprender modelo inverso do sistema	Não-linear; Contínuo; Variante no tempo	Sistema real de micro-robôs; simulação de manipulador de dois elos	Não	SOLIM; Interpolação <i>simplex</i> ; Extrapolação; Auto-organização no espaço de entrada separada da auto-organização no espaço de saída	Controle Inverso	Vetores de suporte de entrada pré-definidos;

## 4 MAPA AUTO-ORGANIZÁVEL PARA SISTEMA NÃO-LINEAR VARIANTE NO TEMPO

As estratégias de controle propostas foram projetadas para atender alguns requisitos importantes de controle de sistemas não-lineares variantes no tempo. São eles: resposta rápida e acurada a mudanças inesperadas na planta ou processo controlado e aprendizagem da dinâmica do sistema a partir de uma amostra pequena de exemplos (MINATOHARA; FURUKAWA, 2011). As propostas de controle construídas neste trabalho buscaram atender a esses requisitos. Para isso, elas foram fundamentadas em uma variante da rede SOM chamada mapa auto-organizável com campo receptivo adaptativo local (LARFSOM). Mais detalhadamente, as soluções de controle foram projetadas usando uma arquitetura de múltiplos modelos. Nesse sentido, o controlador em múltiplos modelos deve ser capaz de identificar o status atual do processo ou da planta, e selecionar uma ação de controle apropriada. Ao utilizar LARFSOM, esse mapa pode provê ao controlador uma estratégia de rápida resposta a mudanças no sistema controlado já que o processo de competição entre nodos pode ser empregado para comandar uma rápida mudança na geração de ações de controle. O LARFSOM também pode ser capaz de aprender diferentes comportamentos do sistema e ações de controle, e aplicá-los adequadamente inclusive para configurações paramétricas do sistema a ser controlado não apresentadas na etapa de treinamento.

Como apontado no Capítulo 3, mapas auto-organizáveis como o LARFSOM apresenta uma degradação na sua acurácia devido ao processo de quantização vetorial. Para construir um modelo mais acurado, o LARFSOM foi estendido para ter a capacidade de mapeamento contínuo por partes. Técnicas de interpolação e extrapolação podem melhorar a acurácia do LARFSOM na resposta, reduzir o número de nodos no mapa, e viabilizar aprendizagem a partir de um número reduzido de exemplos de treinamento (WALTER; RITTER, 1996).

Na primeira seção, a rede LARFSOM é descrita. Logo após, são apresentadas técnicas da literatura e outras desenvolvidas durante o doutorado para a construção de um mapeamento contínuo por partes sobre mapas auto-organizáveis. Por fim, é feita uma discussão sobre os métodos propostos.

### 4.1 LARFSOM - MAPA AUTO-ORGANIZÁVEL COM CAMPO RECEPTIVO ADAPTATIVO LOCAL

O LARFSOM é uma rede inspirada na versão original do SOM e foi proposto originalmente por Araújo e Costa (2009) para o problema de quantização de cores de imagens. A principal diferença para a rede SOM está no fato do LARFSOM ter uma estrutura topológica variante no tempo (ARAÚJO; RÉGO, 2013). O LARFSOM inicia com um mapa de dois nodos e, durante o treinamento, insere novos nodos e novas conexões entre eles no

mapa. A regra de inserção de nodos é baseada na similaridade entre as amostras em um agrupamento e seu protótipo. Se o mapa possuir algum vetor de pesos associado a um nodo suficientemente semelhante ao padrão de treinamento apresentado, esse mapa não necessita de um novo protótipo para representar esse padrão. Caso contrário, esse mapa necessita de um novo nodo para representar adequadamente tal padrão. Uma questão importante é *como definir similaridade entre o padrão de entrada e os protótipos do mapa*. Para resolver isso, o LARFSOM considera grau de similaridade determinado por um limiar de ativação, calculado com respeito a um valor de campo receptivo de vizinhança local (ARAÚJO; COSTA, 2009).

O LARFSOM funciona da seguinte forma: quando um padrão de entrada  $\xi$  é apresentado, a rede procura os dois nodos com vetores de pesos mais próximos de  $\xi$ . Logo em seguida, a rede calcula o grau de ativação deste nodo mais próximo, o nodo vencedor, para decidir entre inserção de um novo nodo ou atualização dos pesos do vencedor. Esse processo de treinamento é repetido várias vezes e se encerra quando a rede converge. O critério de convergência pode ser determinado por um número máximo de iterações ou um nível máximo aceitável do erro não-supervisionado. Por fim, são filtrados todos os nodos desconectados.

O LARFSOM tem as seguintes características:

- Número adaptativo de nodos;
- Topologia variante no tempo;
- Inserção de novos nodos a medida que não há protótipos suficientemente semelhantes ao padrão apresentado;
- Similaridade decidida através de um limiar calculado a partir de uma função de ativação do tipo base radial;
- Campo receptivo ajustado por nodo, calculado considerando a distância dos dois nodos mais próximos;
- Taxa de aprendizagem de cada nodo dependendo de seu número de vitórias;
- Liberdade na construção da vizinhança.

#### 4.1.1 Estrutura do LARFSOM

O modelo apresentado neste trabalho é uma versão estendida do LARFSOM apropriada para aproximação de funções  $f : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$ . Esta versão estendida é modelada como uma memória associativa temporal por quantização vetorial (VQTAM) (BARRETO; ARAÚJO, 2004), então o vetor de pesos do nodo  $n_i$ ,  $\mathbf{w}_i$ , é dividido em duas partes: vetor de pesos de entrada  $\mathbf{w}_i^{in}$  e vetor de pesos de saída  $\mathbf{w}_i^{out}$ . No caso, o vetor  $\mathbf{w}_i^{in}$  armazena informações sobre o espaço de entrada enquanto o vetor  $\mathbf{w}_i^{out}$ , sobre o espaço de saída.

O LARFSOM estruturalmente consiste de um conjunto de nodos conectados entre si. O conjunto de nodos é denotado por  $N$  enquanto o conjunto de conexões por  $C$ . Cada nodo  $n_i = (\mathbf{w}_i^{in}, \mathbf{w}_i^{out}, v_i)$  é composto por um vetor de peso de entrada  $\mathbf{w}_i^{in} \in \mathbb{R}^{d_{in}}$ , um vetor de pesos de saída  $\mathbf{w}_i^{out} \in \mathbb{R}^{d_{out}}$  e um contador de vitórias do nodo  $v_i$ . Uma conexão  $c_{i,j}$  corresponde uma ligação entre os nodos  $n_i$  e  $n_j$ .

#### 4.1.2 Algoritmo de Treinamento do LARFSOM

Os detalhes do procedimento de treinamento do LARFSOM serão apresentados nesta subseção. A Figura 5 ilustra a estrutura variante no tempo do LARFSOM.

Passo 1 - *Inicializar Mapa*: Como dito anteriormente, o mapa inicial é constituído por dois nodos conectados (Fig 5a). O vetores de pesos iniciais são escolhidos aleatoriamente do conjunto de treinamento. Também, são estabelecidos os valores dos seguintes parâmetros:

- Taxa de aprendizagem final ( $\rho_f$ );
- Modulador da taxa de aprendizagem ( $\varepsilon$ );
- Limiar de atividade ( $a_T$ );
- Máxima quantidade de vitórias de um nodo ( $v_{max}$ );
- Erro mínimo ( $e_{min}$ ) ou número máximo de iterações ( $t_{max}$ );
- Número máximo de nodos ( $N_{max}$ ).

Passo 2 - *Selecionar Amostra*: Uma amostra  $\boldsymbol{\xi}$  é aleatoriamente selecionada do conjunto de treinamento e apresentada à rede como estímulo de treinamento (Fig. 5b). A amostra tem a mesma dimensão do vetor de pesos  $\mathbf{w}_i$ , definidas como

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}^{in} \\ \boldsymbol{\xi}^{out} \end{bmatrix} \quad \text{e} \quad \mathbf{w}_i = \begin{bmatrix} \mathbf{w}_i^{in} \\ \mathbf{w}_i^{out} \end{bmatrix} \quad (4.1)$$

Passo 3 - *Competição entre Nodos*: O LARFSOM determina os dois nodos com melhores correspondência com a amostra. Esses nodos são denotados por  $n_{s_1}$  e  $n_{s_2}$  e chamados de nodos vencedores. No caso, os nodos vencedores são os dois nodos que seus vetores de pesos de entrada são os mais próximos do vetor do espaço de entrada  $\boldsymbol{\xi}^{in}$  da amostra. Os dois nodos vencedores são determinados por:

$$d(\mathbf{w}_{s_1}^{in}, \boldsymbol{\xi}^{in}) \leq d(\mathbf{w}_{s_2}^{in}, \boldsymbol{\xi}^{in}) \leq d(\mathbf{w}_i^{in}, \boldsymbol{\xi}^{in}), \quad \forall n_i \in N \quad (4.2)$$

onde  $d(\mathbf{w}, \boldsymbol{\xi}) = \|\mathbf{w} - \boldsymbol{\xi}\|^2$  e  $N$  é o conjunto de todos os nodos da rede.

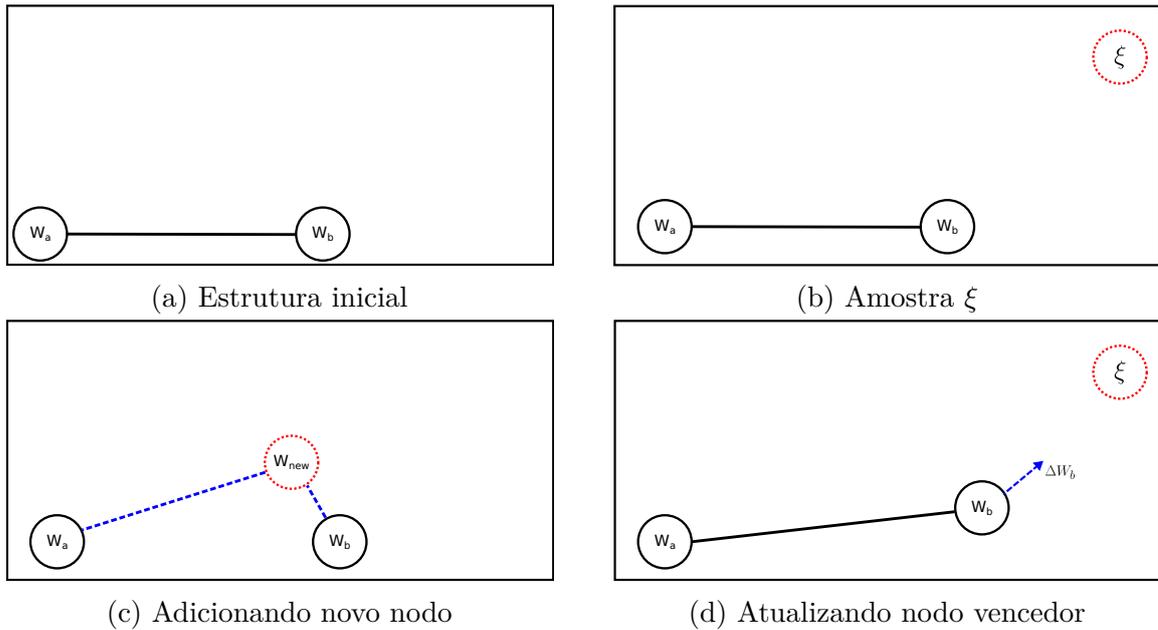


Figura 5 – Estrutura variante no tempo do LARFSOM. (a) Estrutura inicial do LARFSOM com dois nodos conectados. (b) Amostra de treinamento  $\xi$  é apresentada ao mapa. (c) Um novo nodo é adicionado ao mapa. (d) Vencedor é ajustado.

Passo 4 - *Calcular Atividade*: O LARFSOM calcula o sinal de ativação do estímulo de entrada sobre o nodo vencedor. Primeiramente, o campo receptivo do nodo vencedor  $r_{s_1}$  é calculado:

$$r_{s_1} = \left\| \mathbf{w}_{s_1}^{in} - \mathbf{w}_{s_2}^{in} \right\| \quad (4.3)$$

Em seguida, é calculada a atividade de  $n_{s_1}$  por meio de uma *função de base radial*:

$$a_{s_1} = \frac{\exp(-\left\| \xi_{in} - \mathbf{w}_{s_1}^{in} \right\|)}{r_{s_1}} \quad (4.4)$$

Passo 5 - *Inserir Novo Nodo*: A rede insere um novo nodo ( $n_{new}$ ) se a ativação do nodo vencedor é menor que um limiar de ativação ( $a_T$ ) e o número de nodos no mapa  $|N|$  é menor que o número máximo de nodos  $N_{max}$  (Fig. 5c). Nessa situação, a rede:

- Adiciona novo nodo com vetor de pesos  $\mathbf{w}_{new} = \frac{\xi + \mathbf{w}_{s_1} + \mathbf{w}_{s_2}}{3}$ ;
- Remove as conexões entre os dois vencedores  $n_{s_1}$  e  $n_{s_2}$  (caso exista);
- Calcula as distâncias:  $d(\mathbf{w}_{s_1}^{in}, \mathbf{w}_{new}^{in})$ ,  $d(\mathbf{w}_{s_2}^{in}, \mathbf{w}_{new}^{in})$  e  $d(\mathbf{w}_{s_1}^{in}, \mathbf{w}_{s_2}^{in})$ ;
- Insere as conexões relacionadas as duas menores distâncias.

Passo 6 - *Atualizar Nodo Vencedor*: A rede atualiza o vencedor se a atividade  $a_{s_1}$  é maior ou igual ao limiar  $a_T$ , ou o atual número do nodos na rede  $|N|$  é igual ao número máximo de nodos  $N_{max}$  (Fig. 5d). Nessa situação, a rede:

- Insere uma conexão entre os vencedores  $n_{s_1}$  e  $n_{s_2}$  (caso não exista);
- Incrementa o contador de vitória do vencedor:  $v_{s_1}[t] = v_{s_1}[t - 1] + 1$ ;
- Atualiza o vetor de pesos de entrada do vencedor:  $\mathbf{w}_{s_1}^{in}[t + 1] = \mathbf{w}_{s_1}^{in}[t] + \Delta\mathbf{w}_{s_1}^{in}$ ;
- Atualiza o vetor de pesos de saída do vencedor:  $\mathbf{w}_{s_1}^{out}[t + 1] = \mathbf{w}_{s_1}^{out}[t] + \Delta\mathbf{w}_{s_1}^{out}$ .

onde,

$$\Delta\mathbf{w}_{s_1}^{in} = \rho_{in} \times (\boldsymbol{\xi}^{in} - \mathbf{w}_{s_1}^{in}) \quad (4.5)$$

$$\Delta\mathbf{w}_{s_1}^{out} = \rho_{out} \times (\boldsymbol{\xi}^{out} - \mathbf{w}_{s_1}^{out}) \quad (4.6)$$

e

$$\rho_{in} = \rho_{out} = \begin{cases} \varepsilon \times \rho_f^{\frac{v_{s_1}}{v_{max}}} & \text{se } v_{s_1} \leq v_{max} \\ \varepsilon \times \rho_f & \text{caso contrário} \end{cases} \quad (4.7)$$

Passo 7 - *Atualizar Número de Iterações*: A rede incrementa o número de iterações  $t = t + 1$ .

Passo 8 - *Critério de Parada*: O procedimento de treinamento termina quando a rede atinge algum critério de parada. O critério de parada é definido pela Equação 4.8 ou por atingir o número máximo de iterações  $t_{max}$ . Caso o critério não tenha sido atingido, o procedimento retorna para o passo 2.

$$e = \frac{1}{|N|} \sum_{n_i \in N} \|\mathbf{w}_i[t] - \mathbf{w}_i[t - 1]\|^2 \leq e_m = 10^{-4} \quad (4.8)$$

O valor de  $e_m = 10^{-4}$  é sugerido pelo o artigo original do LARFSOM (ARAÚJO; COSTA, 2009). Nos experimentos deste trabalho, foi utilizado apenas o número máximo de iterações como critério de parada.

Passo 9 - *Remover Nodos*: Por fim, a rede remove todos os nodos desconectados.

A Figura 6 mostra o diagrama de fluxo com subprocedimentos utilizados na aprendizagem.

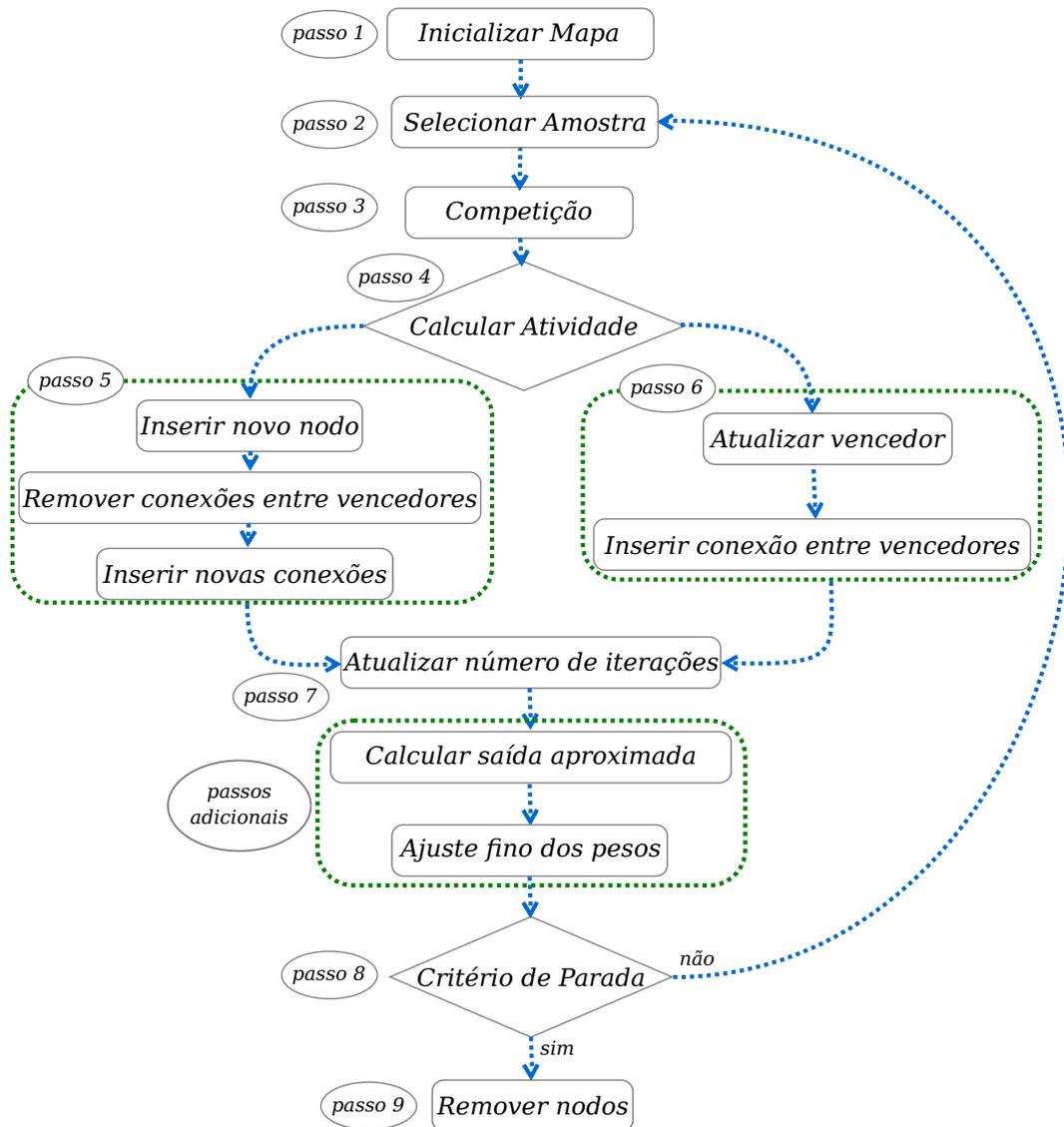


Figura 6 – Diagrama de fluxo do procedimento de treinamento do LARFSOM estendido para aproximação de função.

## 4.2 MAPEAMENTO CONTÍNUO POR PARTES COM LARFSOM

No domínio de sistemas de controle, há sistemas que tem um comportamento que varia consideravelmente ao longo da fase de execução. Sabendo que grande parte desses sistemas envolve também não-linearidades, construir um modelo ou um controlador para tais sistemas pode ser visto como construir um modelo que aproxima o comportamento de uma função não-linear variante no tempo. Dentro deste contexto, como apontado em Flentge (2006) e Vijayakumar, D’Souza e Schaal (2005), um algoritmo de aprendizagem deve cumprir três requisitos: (1) permitir aprendizagem incremental para lidar com a saída de funções variante no tempo e com distribuição de entrada não-estacionária (2) apresentar custo computacional baixo e (3) realizar estimativas acuradas.

Neste trabalho, foi adotado o cenário de aprendizagem incremental descrito por Schaal e Atkeson (1998). Aprendizagem incremental corresponde a um sistema de aprendizagem

que tem seus pesos atualizados sem degradação dos conhecimentos previamente aprendidos e leva em conta três considerações: (1) um cenário de memória limitada de modo que não há uma memória extra para armazenar e reutilizar as instâncias de treinamento. Essa consideração é importante visto que há abordagens de armazenamento e reuso de dados de treinamento previamente apresentados ao modelo de aprendizagem para promover uma aprendizagem incremental, porém essas abordagens são ineficientes em ambiente de tempo real (PARISI et al., 2019). (2) As distribuições de entrada e de saída são desconhecidas, e (3) essas distribuições podem variar ao longo do tempo.

O SOM e outros mapas derivados dele como o LARFSOM constroem um mapeamento discreto do espaço de entrada no espaço de saída (CAMPOS; CARPENTER, 2000). Mesmo com uma perda na acurácia devido a discretização, muitos autores propuseram técnicas de aproximação de função aplicando SOMs (BARRETO, 2007; CAMPOS; CARPENTER, 2000; FLENTGE, 2006; GÖPPERT; ROSENSTIEL, 1995; GÖPPERT; ROSENTIEL, 1997; HECHT; LEFORT; GEPPERTH, 2015; LUDWIG et al., 1995; THAMPI et al., 2002; WALTER; RITTER, 1996). A principal motivação para aplicar SOMs em um problema de aproximação de função é que o SOM tem a capacidade de aprender um mapeamento não-linear de um espaço de entrada de alta dimensão para um espaço de saída de baixa dimensão preservando a topologia do espaço (GÖPPERT; ROSENTIEL, 1997; KOHONEN, 1982).

Modelar uma função contínua através de um mapeamento discreto pode fornecer uma modelagem insuficientemente acurada para o problema alvo. A partir disto, nasce a necessidade de projetar e utilizar técnicas de interpolação e extrapolação sobre redes do tipo SOM para criar um mapeamento contínuo, ou um mapeamento contínuo por partes (CAMPOS; CARPENTER, 2000; GÖPPERT; ROSENTIEL, 1997; WALTER; RITTER, 1996). Tais técnicas aplicadas sobre SOM podem apresentar as seguintes características:

- Consistência no mapeamento entre entrada e saída: A distribuição espacial dos nodos no espaço de saída é similar à distribuição das amostras no espaço de entrada (LUDWIG et al., 1995);
- Cooperação topológica: O vetor de saída pode ser computado através da cooperação do nodo vencedor com os nodos da sua vizinhança topológica (LUDWIG et al., 1995);
- Resultados acurados usando um número reduzido de nodos e padrões de treinamentos. As técnicas de interpolação e extrapolação aplicadas sobre os nodos de um mapa auto-organizável podem proporcionar resultados mais acurados com um número menor de nodos e ajustados a partir de um conjunto de treinamento menor do que na versão discreta (WALTER; RITTER, 1996).

#### 4.2.1 Trabalhos Relacionados

Técnicas de interpolação e extrapolação propostas na literatura aplicadas para mapas auto-organizáveis incluem diversas abordagens. Entre elas, destacam-se o mapa linear

local (*local linear map*) (LLM) (MARTINETZ; RITTER; SCHULTEN, 1990; RITTER; MARTINETZ; SCHULTEN, 1992) e interpolação localmente ponderada (*locally weighted interpolation*) (FLENTGE, 2006).

#### 4.2.1.1 Mapa Linear Local

O mapa linear local se baseia na mesma estrutura do SOM descrita na Seção 3.1.1. Há um conjunto de nodos  $n_i \in N$ , no qual cada nodo tem um vetor de pesos de entrada  $\mathbf{w}_i^{in}$  e um vetor de pesos de saída  $\mathbf{w}_i^{out}$ . Esses nodos estão conectados em uma estrutura topológica predefinida  $C$ . Em cada nodo, há um modelo linear local que é descrito pelo vetor de pesos de saída  $\mathbf{w}_i^{out}$ , e uma matriz jacobiana  $\mathbf{A}_i$ , que, neste caso, vai representar estimativas das derivadas parciais da função a ser aproximada. O mapeamento de uma amostra do espaço de entrada  $\boldsymbol{\xi}^{in}$  para o espaço de saída será construído utilizando apenas o nodo vencedor. Logo, a saída aproximada será calculada da seguinte forma:

$$\widetilde{\mathbf{M}}_s(\boldsymbol{\xi}^{in}) = \mathbf{w}_s^{out} + \mathbf{A}_s (\boldsymbol{\xi}^{in} - \mathbf{w}_s^{in}) \quad (4.9)$$

onde o nodo vencedor  $n_s$  é determinado por:

$$\|\mathbf{w}_s^{in} - \boldsymbol{\xi}^{in}\| \leq \|\mathbf{w}_i^{in} - \boldsymbol{\xi}^{in}\|, \quad \forall n_i \in N \quad (4.10)$$

Os vetores de pesos  $\mathbf{w}^{in}$  e  $\mathbf{w}^{out}$ , e a matriz jacobiana  $\mathbf{A}$  são ajustados pela a regra de auto-organização herdada do SOM. Então, o nodo vencedor  $n_s$  e seus nodos vizinhos são ajustados pela mesma regra apresentada na Eq. 3.1:

$$\mathbf{w}_k^{in}[t+1] = \mathbf{w}_k^{in}[t] + \alpha^{in}[t] h_{s,k}^{in}[t] (\boldsymbol{\xi}^{in} - \mathbf{w}_k^{in}[t]) \quad (4.11)$$

$$\mathbf{w}_k^{out}[t+1] = \mathbf{w}_k^{out}[t] + \alpha^{out}[t] h_{s,k}^{out}[t] (\boldsymbol{\xi}_*^{out} - \mathbf{w}_k^{out}[t]) \quad (4.12)$$

$$\mathbf{A}_k[t+1] = \mathbf{A}_k[t] + \alpha^A[t] h_{s,k}^A[t] (\mathbf{A}_* - \mathbf{A}_k[t]) \quad (4.13)$$

as funções de vizinhança  $h_{s,k}^{in}[t]$ ,  $h_{s,k}^{out}[t]$  e  $h_{s,k}^A[t]$  são descritas de acordo com a Eq. 3.2 e as taxas de aprendizagem  $\alpha^{in}[t]$ ,  $\alpha^{out}[t]$  e  $\alpha^A[t]$ , pela Eq. 3.3. Os termos  $\boldsymbol{\xi}^e$  e  $\mathbf{A}^e$  são estimados através das seguintes equações:

$$\mathbf{A}_* = \mathbf{A}_s + \left( (\boldsymbol{\xi}_*^{out} - \widetilde{\mathbf{M}}_s(\boldsymbol{\xi}^{in})) - \mathbf{A}_s \mathbf{d}_s \right) \frac{(\mathbf{d}_s)^T}{\|\mathbf{d}_s\|^2} \quad (4.14)$$

$$\boldsymbol{\xi}_*^{out} = \mathbf{w}_s^{out}[t] + \mathbf{A}_s[t+1] (\boldsymbol{\xi}^{in} - \mathbf{w}_k^{in}[t]) \quad (4.15)$$

onde  $\mathbf{d}_s = \boldsymbol{\xi}^{in} - \mathbf{w}_s^{in}$ . Outros trabalhos no contexto de aproximação de função que se basearam nesse método são (MOSHOU; RAMON, 1997; WINTER; METTA; SANDINI, 2000; CHO; PRINCIPE; MOTTER, 2003).

#### 4.2.1.2 Interpolação Localmente Ponderada

Flentge propôs o gás neural crescente com interpolação localmente ponderada (*locally weighted interpolating growing neural gas* - LWIGNG) para aproximação de funções não-linear com aprendizagem incremental (FLENTGE, 2006). Neste trabalho, Flentge generalizou a interpolação linear por partes proposta por Göppert e Rosenstiel (1995) para o SOM utilizando a regularização de Tikhonov. Flentge aplicou também uma técnica de aprendizagem localmente ponderada para melhorar as aproximações calculadas pelos modelos locais e chamou esse método de interpolação linear por partes localmente ponderada (*locally weighted piecewise linear interpolation* - LWI). LWIGNG foi aplicado com sucesso para aproximar funções variantes no tempo. No entanto, a técnica LWI consome um tempo excessivo tanto no treinamento quanto na execução e isto pode inviabilizar uma aplicação em problemas de tempo real.

Na próxima seção, são apresentadas adaptações sobre o LARFSOM para ter a capacidade de mapeamento contínuo por partes. As duas técnicas propostas por Flentge foram aplicadas no LARFSOM e são denominadas de LARFSOM com Interpolação (I-LARFSOM) e LARFSOM com interpolação linear por partes localmente ponderada (LWI-LARFSOM). Também, são descritas duas técnicas alternativas para a interpolação linear de modo que a solução final tenha um custo computacional baixo. Essas alternativas são chamadas de *método da diferença* - D e *método da diferença estendido* - xD. Esses dois métodos também são melhorados através da aprendizagem localmente ponderada e são denominados por *método da diferença localmente ponderado* - LWD e *método da diferença estendido localmente ponderado* - LWxD. Essas técnicas aplicadas no LARFSOM são chamadas de D-LARFSOM, xD-LARFSOM, LWD-LARFSOM e LWxD-LARFSOM. A Figura 7 ilustra os nodos e as conexões usadas na geração da aproximação por cada método de saída contínua por partes.

#### 4.2.2 LARFSOM com Interpolação Linear por Partes - I-LARFSOM

Flentge parte do princípio de que a maneira mais direta de aproximar uma função  $f : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$  com uma rede SOM é associar cada nodo  $n_i$  a um valor de saída  $w_{out}$ . No caso, Flentge trabalhou apenas com  $d_{out} = 1$ . Outro ponto, herdado do Mapa Auto-organizável com Interpolação (I-SOM) proposto por Göppert e Rosenstiel (1995), consiste no cálculo da interpolação através de um sistema de coordenadas locais. Esse sistema de coordenadas tem o centro representado pelo vencedor e emprega algumas conexões para os vizinhos do vencedor como eixos de coordenadas. Nesta abordagem, o primeiro passo é representar a amostra  $\xi$  nesse sistema de coordenadas locais. Para tal, o vetor de entrada  $\xi^{in}$  é decomposto em uma combinação linear do vetor de pesos de entrada  $w_s^{in}$  do nodo vencedor  $n_s$  e dos vetores diferença normalizados  $l_{s,i}$  gerados a partir dos nodos vizinhos

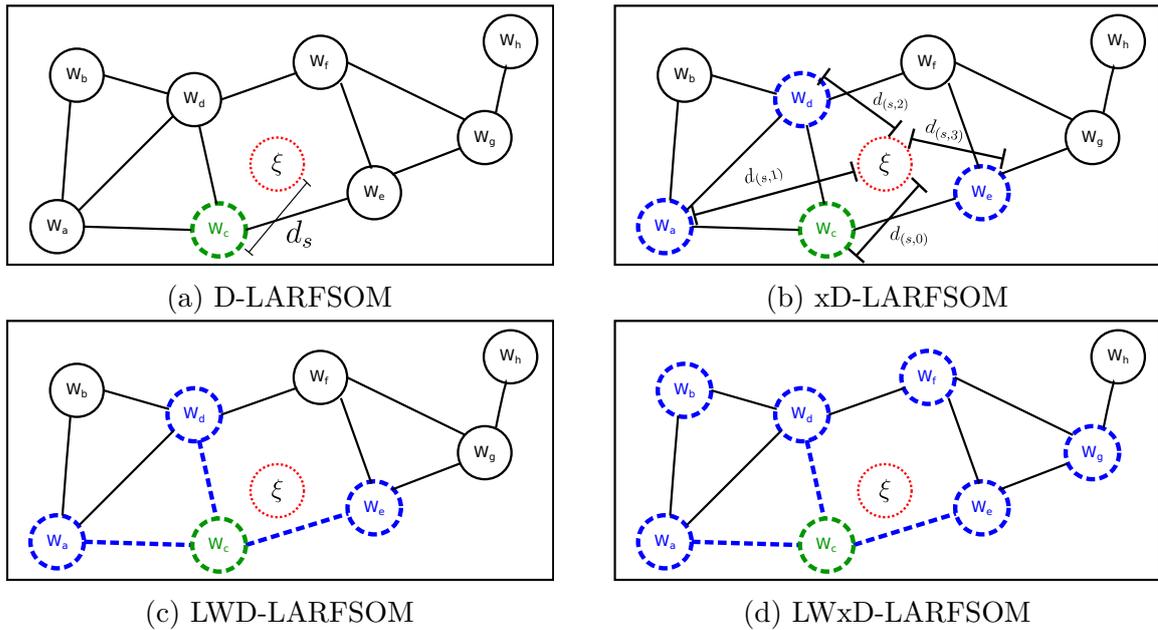


Figura 7 – Os nodos e as conexões traçados em verde e azul são usados para calcular a saída aproximada através do D-LARFSOM, xD-LARFSOM, LWD-LARFSOM e LWxD-LARFSOM, respectivamente. O nodo  $W_c$  (em verde) é o nodo vencedor em todas as ilustrações.

$n_{(s,i)}$ :

$$\boldsymbol{\xi}^{in} = \mathbf{w}_s^{in} + \sum_{i=1}^{N_s} a_i \mathbf{l}_{s,i} \quad (4.16)$$

com

$$\mathbf{l}_{s,i} = \frac{(\mathbf{w}_{(s,i)}^{in} - \mathbf{w}_s^{in})}{\|\mathbf{w}_{(s,i)}^{in} - \mathbf{w}_s^{in}\|} \quad (4.17)$$

onde a tupla  $(s, i)$  para  $i = 1, \dots, N_s$  corresponde ao índice do  $i$ -ésimo nodo vizinho do vencedor  $n_s$ , logo,  $\mathbf{w}_{(s,i)}^{in}$  é o vetor pesos de entrada do nodo vizinho  $n_{(s,i)}$  e  $a_i$  é o valor da coordenada sobre o eixo  $\mathbf{l}_{s,i}$ .

A Equação 4.16 pode não ter solução, ter uma única solução ou várias soluções. Isto vai depender do número de vizinhos  $N_s$ , da dimensão do vetor de entrada  $\boldsymbol{\xi}^{in}$  e da posição dos vizinhos. Assim, esse problema pode ser formulado como um sistema linear:

$$\mathbf{D}\mathbf{a} = \mathbf{d}_s \quad (4.18)$$

com

$$\mathbf{D} = [\mathbf{l}_{s,1} \ \cdots \ \mathbf{l}_{s,N_s}] \quad \mathbf{a} = [a_1 \ \cdots \ a_{N_s}]^T \quad \mathbf{d}_s = \boldsymbol{\xi}^{in} - \mathbf{w}_s^{in} \quad (4.19)$$

Flentge resolveu a Equação 4.18 através do uso da regularização de Tikhonov. Esse método calcula  $\mathbf{a}$  de maneira a minimizar a seguinte soma:

$$\|\mathbf{d}_s - \mathbf{D}\mathbf{a}\|^2 + \mu \|\mathbf{a}\|^2 \quad (4.20)$$

O termo  $\mu$  é um ponderador e penaliza valores grandes de  $|a_i|$  dentro da equação de erro quadrático. Como a solução depende da escala da entrada, deve-se considerar os vetores diferença entre o vencedor e seus vizinhos de maneira normalizada (Eq. 4.17). A solução da regularização de Tikhonov é a seguinte<sup>1</sup>:

$$\mathbf{a} = (\mathbf{D}^T \mathbf{D} + \mu \mathbf{I})^{-1} \mathbf{D}^T \mathbf{d}_s \quad (4.21)$$

onde  $0 < \mu \ll 1$  e  $\mathbf{I}$  é a matriz identidade.

Para evitar uma extrapolação excessiva, um procedimento é realizado para garantir que não haverá um vetor  $a_i \mathbf{l}_{s,i}$  na decomposição aproximada do vetor de entrada  $\boldsymbol{\xi}^{in}$  que seja maior que o vetor diferença correspondente ao vizinho  $n_i$ . Isto significa, enquanto a condição 4.22 não for satisfeita, um procedimento é realizado para remoção de alguns vetores diferença.

$$\max_i \{ \|a_i \mathbf{l}_{s,i}\| \} \equiv \max_i \{ |a_i| \} \leq \left\| \mathbf{w}_{(s,i)}^{in} - \mathbf{w}_s^{in} \right\| \quad (4.22)$$

A remoção é realizada através da seleção de vetores diferenças normalizados com direções similares ou opostas de modo a manter os máximo de vetores diferenças com diferentes direções.

Este procedimento é realizado da seguinte forma até atingir a condição através do recálculo dos  $a_i$ :

1. Selecione os vetores diferenças normalizados  $\mathbf{l}_{s,i}$  e  $\mathbf{l}_{s,j}$  com maior produto escalar absoluto  $|\mathbf{l}_{s,i} \cdot \mathbf{l}_{s,j}|$ .
2. Remova a coluna de  $\mathbf{D}$  correspondente ao vetor diferenças normalizado que minimiza o produto escalar  $\mathbf{l}_{s,k} \cdot \mathbf{d}_s$  onde  $k = i$  ou  $k = j$ .

O passo 1, seleciona os vetores mais “dependentes” entre si. O passo 2, mantém o vetor que aponta mais para a direção de  $\mathbf{d}_s$ .

### Interpolação Linear por Partes

A saída do modelo local  $\widetilde{M}_s(\boldsymbol{\xi}^{in})$  é calculada através de um interpolação entre o valor de saída do nodo vencedor  $w_s^{out}$  e dos valores de saída correspondentes aos vizinhos do

<sup>1</sup> Não houve a necessidade de calcular a matriz inversa. A solução da regularização de Tikhonov pode ser encontrada considerando a resolução dos sistemas lineares através do método LU. Isso é feito através do operador \ presente nas linguagens Matlab, Octave e Julia.

vencedor  $w_{(s,i)}^{out}$  que não foram excluídos de  $\mathbf{D}$ . Para simplificar a explicação através de formas matemáticas, considera-se o parâmetro  $a_{s,j} = 0$ , se o vetor diferença normalizado  $\mathbf{l}_{s,j}$  foi removido de  $\mathbf{D}$  e  $a_{s,j} = a_i$ , caso o vetor diferença normalizado  $\mathbf{l}_{s,j}$  está associado ao valor  $a_i$  da coluna  $i$  do vetor  $\mathbf{D}$ . Sendo assim, o cálculo da saída do modelo local fica:

$$\widetilde{M}_s(\boldsymbol{\xi}^{in}) = w_s^{out} + \sum_{i=1}^{N_s} a_{s,i} \frac{(w_{(s,i)}^{out} - w_s^{out})}{\|\mathbf{w}_{(s,i)}^{in} - \mathbf{w}_s^{in}\|} \quad (4.23)$$

Melhores resultados de aproximação podem ser alcançados através do gradiente descendente para ajustes dos valores de saída dos nodos. Para usar o gradiente descendente foi considerada a função de erro  $E(\boldsymbol{\xi}^{in}) = (1/2)(\xi^{out} - \widetilde{M}_s(\boldsymbol{\xi}^{in}))^2$ . Então, a regra de ajuste do valor  $w_s^{out}$  do nodo vencedor e dos seus vizinhos  $w_{(s,i)}^{out}$  fica da seguinte forma

$$\Delta w_s^{out} = \alpha_{out} \left( 1 - \sum_{i=1}^{N_s} \frac{a_{s,i}}{\|\mathbf{w}_{(s,i)}^{in} - \mathbf{w}_s^{in}\|} \right) (\xi^{out} - \widetilde{M}_s(\boldsymbol{\xi}^{in})) \quad (4.24)$$

$$\Delta w_{(s,k)}^{out} = \alpha_{out} \frac{a_{s,k}}{\|\mathbf{w}_{(s,k)}^{in} - \mathbf{w}_s^{in}\|} (\xi^{out} - \widetilde{M}_s(\boldsymbol{\xi}^{in})) \quad (4.25)$$

onde  $\alpha_{out}$  corresponde a taxa de aprendizagem.<sup>2</sup>

### 4.2.3 Método da Diferença - D-LARFSOM

O método da diferença parte da ideia de usar o vetor diferença  $\mathbf{d}_s$  entre o vetor de entrada  $\boldsymbol{\xi}^{in}$  e o vetor de pesos de entrada  $\mathbf{w}_i^{in}$  para gerar uma saída aproximada (Fig. 7a). Nesse caso, o vetor diferença, calculado pela expressão  $\mathbf{d}_s = \boldsymbol{\xi}^{in} - \mathbf{w}_s^{in}$ , onde  $s$  referencia o nodo vencedor  $n_s$ , é ponderado por um vetor de pesos  $\mathbf{k}$  para produzir um fator de ajuste sobre a saída do nodo vencedor  $w_s^{out}$  para o padrão de entrada  $\boldsymbol{\xi}^{in}$ .

Diferentemente da técnica da interpolação linear por partes previamente descrita, o método da diferença aproxima uma saída vetorial, isto é,  $\boldsymbol{\xi}^{out} \in \mathbb{R}^{d_{out}}$ , onde  $d_{out} \geq 1$ . Para isso, cada nodo  $n_i$  passa a ter  $d_{out}$  vetores de pesos denotados por  $\mathbf{k}_i^l$ , onde  $l = 1, \dots, d_{out}$ . O vetor de saída do modelo local  $\widetilde{M}_s(\boldsymbol{\xi}^{in})$  é calculado através de uma combinação do vetor de pesos de saída do vencedor  $\mathbf{w}_s^{out}$  com o vetor diferença  $\mathbf{d}_s$ . Então, o cálculo da saída do modelo local fica:

$$\widetilde{M}_s[l](\boldsymbol{\xi}^{in}) = w_s^{out}[l] + \mathbf{k}_s^l \cdot \mathbf{d}_s, \quad l = 1, \dots, d_{out} \quad (4.26)$$

onde  $\mathbf{k}_s^l$  é o vetor de pesos do nodo vencedor  $n_s$  usado no cálculo do valor da posição  $l$  do vetor da aproximação da saída  $\widetilde{M}_s(\boldsymbol{\xi}^{in})$ .

O vetor de pesos  $\mathbf{k}_s^l$  pode ser ajustado usando o método do gradiente descendente sobre o erro quadrático médio. A função de custo a ser minimizada é dada por  $\mathbf{E}(\boldsymbol{\xi}^{in})[l] =$

<sup>2</sup> Os cálculos das derivadas para dedução das Equações 4.24 e 4.25 estão apresentados em (FLENTGE, 2006).

$(1/2)(\boldsymbol{\xi}^{out}[l] - \widetilde{\mathbf{M}}_s[l](\boldsymbol{\xi}^{in}))^2$  para  $l = 1, \dots, d_{out}$ . Logo, a regra de atualização dos pesos é dada por:

$$\Delta \mathbf{k}_s^l = \alpha_k \cdot \mathbf{d}_s \cdot (\boldsymbol{\xi}^{out}[l] - \widetilde{\mathbf{M}}_s[l](\boldsymbol{\xi}^{in})) \quad (4.27)$$

onde  $\alpha_k$  é a taxa de aprendizagem para os vetores  $\mathbf{k}$ .<sup>3</sup>

É importante ressaltar que o método da diferença é semelhante ao mapa linear local (LLM) (MARTINETZ; RITTER; SCHULTEN, 1990) visto que ambos são baseados em uma modelagem linear em cada região local. A diferença parte da função de erro a ser otimizada. No mapa linear local (LLM), a função de erro é dada por  $\mathbf{E}(\boldsymbol{\xi}^{in}) = (1/2)((\boldsymbol{\xi}^{out} - \widetilde{\mathbf{M}}_s(\boldsymbol{\xi}^{in})) - \mathbf{d}_s)^T \cdot (\boldsymbol{\xi}^{out} - \widetilde{\mathbf{M}}_s(\boldsymbol{\xi}^{in})) - \mathbf{d}_s)$ , enquanto no método da diferença, a função de erro é  $\mathbf{E}(\boldsymbol{\xi}^{in})[l] = (1/2)(\boldsymbol{\xi}^{out}[l] - \widetilde{\mathbf{M}}_s[l](\boldsymbol{\xi}^{in}))^2$ . Os vetores  $\mathbf{k}_i^l$  do D-LARFSOM podem ser interpretados como fatores de correção da aproximação local mais diretamente relacionados com o erro do modelo local.

Outro ponto a se observar é que o mapa linear local utiliza o processo de auto-organização para adaptar o vetor de pesos de entrada, o vetor de pesos de saída e a matriz jacobiana, e também, utiliza o gradiente descendente sobre o vetor de pesos de saída e a matriz jacobiana. Já o D-LARFSOM utiliza o processo de auto-organização para criar representações no espaço de entrada e no espaço de saída interligadas pela própria estrutura do mapa. Logo, o processo de auto-organização adapta apenas os vetores de pesos de entrada e de saída. Já o método do gradiente descendente é utilizado apenas para ajustar os vetores de pesos  $\mathbf{k}_i^l$ . Enquanto a matriz jacobiana do LLM é atualizada pelo processo de auto-organização e pelo gradiente descendente. Essa segunda diferença proporciona um custo computacional menor ao D-LARFSOM em comparação ao LLM.

#### 4.2.4 Método da Diferença Estendido - xD-LARFSOM

O método da diferença pode ser estendido usando os nodos da vizinhança local. No caso, o vetor de saída estimado é calculado usando os vetores diferença entre o vetor de entrada e o vetor de pesos de entrada do vencedor e dos seus vizinhos (Fig. 7b). Os vetores diferença são dados por  $\mathbf{d}_{s,i} = \boldsymbol{\xi}^{in} - \mathbf{w}_{(s,i)}^{in}$  onde a tupla  $(s, i)$  para  $i = 1, \dots, |N_s|$  representa o índice do  $i$ -ésimo vizinho do vencedor  $n_s$  e a tupla  $(s, 0)$  denota o vencedor  $n_s$ , logo,  $\mathbf{w}_{(s,0)}^{in}$  denota o vetor de pesos de entrada  $\mathbf{w}_s^{in}$ .

Em cada nodo  $n_i$ , novos vetores de pesos denotados por  $\mathbf{k}_{i,j}^l$  são adicionados para cada conexão  $c_{i,j}$  que nodo  $n_j$  possui e para cada valor no vetor de saída,  $l = 1, \dots, d_{out}$ . Adicionalmente, os vetores de pesos  $\mathbf{k}_{i,0}^l$  para  $l = 1, \dots, d_{out}$  também são inclusos no nodo  $n_i$ . Então, a saída vetorial aproximada calculada pelo modelo local é dada por:

$$\widetilde{\mathbf{M}}_s[l](\boldsymbol{\xi}^{in}) = \mathbf{w}_s^{out}[l] + \sum_{j=0}^{|N_s|} \mathbf{k}_{s,j}^l \cdot \mathbf{d}_{s,j}, \quad l = 1, \dots, d_{out} \quad (4.28)$$

<sup>3</sup> Os cálculos das derivadas para dedução da Equação 4.27 estão apresentados no Apêndice C.

onde  $s$  denota o nodo vencedor  $n_s$ .

Cada vetor de pesos  $\mathbf{k}_{s,j}^l$  pode ser ajustado através do método do gradiente descendente. Usando a função de custo dada pelo erro quadrático médio  $\mathbf{E}(\boldsymbol{\xi}_{in})[l] = (1/2)(\boldsymbol{\xi}_{out}[l] - \widetilde{M}_s[l](\boldsymbol{\xi}_{in}))^2$ , a regra de ajuste dos pesos é calculada por:

$$\Delta \mathbf{k}_{s,j}^l = \alpha_k \cdot \mathbf{d}_{s,j} \cdot (\boldsymbol{\xi}_{out}[l] - \widetilde{M}_s[l](\boldsymbol{\xi}_{in})) \quad (4.29)$$

onde  $\alpha_k$  é a taxa de aprendizagem.<sup>4</sup>

#### 4.2.5 LARFSOM para Mapeamento Contínuo por Partes Localmente Ponderado

Através de aprendizagem localmente ponderada (*locally weighted learning*), é possível aprimorar o valor estimado por qualquer um dos métodos descritos nas subseções anteriores. O novo valor de saída  $\widetilde{F}_s(\boldsymbol{\xi}^{in})$  vai ser calculado através de uma combinação do valor de saída  $\widetilde{M}_{(s,0)}(\boldsymbol{\xi}^{in})$  do nodo vencedor com os valores de saída dos seus vizinhos  $\widetilde{M}_{(s,i)}(\boldsymbol{\xi}^{in})$ . O cálculo dos valores de saída  $\widetilde{M}_{(s,i)}(\boldsymbol{\xi}^{in})$  pode ser realizado pelo método da interpolação linear por partes, pelo método da diferença ou pelo método da diferença estendido. A combinação dessas técnicas produz novas versões do LARFSOM estendido para mapeamento contínuo:

- LWI-LARFSOM: LARFSOM com interpolação linear por partes localmente ponderada. A saída do modelo local  $\widetilde{M}$  é calculada através Eq. 4.23;
- LWD-LARFSOM: LARFSOM com método da diferença localmente ponderado. A saída do modelo local  $\widetilde{M}$  é calculada através Eq. 4.26 (Fig. 7c);
- LWxD-LARFSOM: LARFSOM com o método da diferença estendido localmente ponderado. A saída do modelo local  $\widetilde{M}$  é calculada através Eq. 4.28 (Fig. 7d).

Para o cálculo da ponderação dos modelos locais, é adicionado um novo parâmetro  $\lambda_{i,j}$  para cada conexão  $c_{i,j}$  existente entre os nodos da rede. O valor de  $\lambda_{i,j}$  vai corresponder ao valor de “abrangência” do modelo local do nodo  $n_j$  em direção ao nodo  $n_i$ . Adicionalmente, existirá o parâmetro  $\lambda_{i,0}$  que significará a abrangência do modelo local do nodo  $n_i$  sobre si mesmo. Estes  $\lambda_{i,j}$  são usados para calcular pesos normalizados  $m$ , que também vão depender da distância entre o vetor de entrada e o vetor de pesos de entrada do nodo  $n_j$ . Então, o cálculo dos pesos normalizados fica:

$$m_{s,k}(\boldsymbol{\xi}^{in}) = \frac{e^{-\lambda_{s,(s,k)}} \|\boldsymbol{\xi}^{in} - \mathbf{w}_{(s,k)}^{in}\|}{\sum_{l=0}^{|N_s|} e^{-\lambda_{s,(s,l)}} \|\boldsymbol{\xi}^{in} - \mathbf{w}_{(s,l)}^{in}\|}, \quad k = 0, \dots, |N_s| \quad (4.30)$$

onde a tupla  $(s, i)$  para  $i = 1, \dots, N_s$  corresponde ao índice do  $i$ -ésimo nodo vizinho do vencedor  $n_s$  e a tupla  $(s, 0)$  corresponde ao índice do nodo vencedor  $n_s$ .

<sup>4</sup> Os cálculos das derivadas para dedução da Equação 4.29 estão apresentados no Apêndice C.

## Saída dos Modelos Locais Localmente Ponderados

A aproximação final  $\tilde{F}_s(\boldsymbol{\xi}^{in})$  pode ser calculada através da ponderação dos modelos locais utilizando os pesos normalizados  $m_{s,k}(\boldsymbol{\xi}^{in})$ . Sendo assim, o cálculo da aproximação final fica<sup>5</sup>:

$$\tilde{F}_s(\boldsymbol{\xi}^{in}) = \sum_{k=0}^{|N_s|} m_{s,k}(\boldsymbol{\xi}^{in}) \tilde{M}_{(s,k)}(\boldsymbol{\xi}^{in}) \quad (4.31)$$

Para cada modelo local  $\tilde{M}_{(s,k)}$  utilizado para o cálculo da saída aproximada  $\tilde{F}_s(\boldsymbol{\xi}^{in})$ , podem ser realizados ajustes finos. Neste caso, a regra de ajuste vai depender da técnica usada no cálculo do modelo local  $\tilde{M}_{(s,k)}$ .

### 4.2.5.1 Interpolação Linear por Partes Localmente Ponderada - LWI-LARFSOM

Usando a interpolação linear por partes como modelo local  $\tilde{M}_{(s,k)}$ , o valor de saída armazenado em cada nodo pode ser ajustados da seguinte forma:

$$\Delta w_{(s,k)}^{out} = \alpha_{out} m_{s,k}(\boldsymbol{\xi}^{in}) \left( 1 - \sum_{j=1}^{N_{(s,k)}} \frac{a_{(s,k),j}}{\|\mathbf{w}_{((s,k),j)}^{in} - \mathbf{w}_{(s,k)}^{in}\|} \right) (\xi^{out} - \tilde{F}_s(\boldsymbol{\xi}^{in})) \quad (4.32)$$

$$\Delta w_{((s,k),j)}^{out} = \alpha_{out} m_{s,k}(\boldsymbol{\xi}^{in}) \frac{a_{(s,k),j}}{\|\mathbf{w}_{((s,k),j)}^{in} - \mathbf{w}_{(s,k)}^{in}\|} (\xi^{out} - \tilde{F}_s(\boldsymbol{\xi}^{in})) \quad (4.33)$$

para todos os  $j = 1, \dots, |N|_{(s,k)}$  vizinhos do nodo  $n_{(s,k)}$ .<sup>6</sup>

### 4.2.5.2 Método da Diferença Localmente Ponderado - LWD-LARFSOM

Usando o método da diferença como modelo local  $\tilde{M}_{(s,k)}$ , os vetores  $\mathbf{k}_{(s,k)}^l$  armazenados no nodo  $n_{(s,k)}$  podem ser ajustados por:

$$\Delta \mathbf{k}_{(s,k)}^l = \alpha_k \cdot m_{s,k} \cdot \mathbf{d}_{(s,k)} \cdot (\boldsymbol{\xi}^{out}[l] - \tilde{M}_{(s,k)}[l](\boldsymbol{\xi}^{in})) \quad (4.34)$$

onde  $\alpha_k$  é a taxa de aprendizagem dos vetores  $\mathbf{k}$  e  $l = 1, \dots, d_{out}$ .<sup>6</sup>

### 4.2.5.3 Método da Diferença Estendido Localmente Ponderado - LWxD-LARFSOM

Usando o método da diferença estendido como modelo local  $\tilde{M}_{(s,k)}$ , os vetores  $\mathbf{k}_{(s,k),j}^l$  armazenados no nodo  $n_{((s,k),j)}$  podem ser ajustados por:

$$\Delta \mathbf{k}_{(s,k),j}^l = \alpha_k \cdot m_{s,k} \cdot \mathbf{d}_{(s,k),j} \cdot (\boldsymbol{\xi}^{out}[l] - \tilde{M}_{(s,k)}[l](\boldsymbol{\xi}^{in})), \quad n_j \in n_{(s,k)} \cup N_{(s,k)} \quad (4.35)$$

<sup>5</sup> A Equação 4.31 é semelhante ao mecanismo de inferência aplicado no modelo fuzzy Takagi-Sugeno (Takagi; Sugeno, 1985), porém, a função equivalente ao  $\tilde{M}_{(s,k)}$  no modelo Takagi-Sugeno se limita, normalmente, a funções polinomiais de primeiro grau ou constantes (SIDDIQUE, 2014).

<sup>6</sup> Os cálculos das derivadas para dedução das Equações 4.32 e 4.33 estão apresentados em (FLENTGE, 2006).

onde  $\alpha_k$  é a taxa de aprendizagem dos vetores  $\mathbf{k}$  e  $l = 1, \dots, d_{out}$ .<sup>7</sup>

#### 4.2.5.4 Adaptação da Abrangência dos Modelos Locais

Os valores para os pesos de abrangência  $\lambda_{i,j}$  podem ser calculados através do método do gradiente descendente de maneira a refinar as aproximações feitas pelo modelo final. Isto pode ser feito de modo que modelos locais com uma boa aproximação tenha um peso maior, enquanto aqueles que tem aproximações ruins, passem a ter pesos menores. Sendo assim, os ajustes podem ser calculados por:

$$\Delta\lambda_{s,(s,k)} = -\alpha_\lambda \left\| \boldsymbol{\xi}^{in} - \mathbf{w}_{(s,k)}^{in} \right\| m_{s,k}(\boldsymbol{\xi}^{in}) \left( \tilde{F}_s(\boldsymbol{\xi}^{in}) - \tilde{M}_{(s,k)}(\boldsymbol{\xi}^{in}) \right) \left( \tilde{F}_s(\boldsymbol{\xi}^{in}) - \xi^{out} \right) \quad (4.36)$$

onde  $\alpha_\lambda$  é um taxa de aprendizagem para os pesos  $\lambda_{i,j}$ .<sup>8</sup> Flentge (2006) sugere que o valor inicial para os pesos de abrangência  $\lambda_{i,j}$  seja nulo.

#### 4.2.6 Treinamento do LARFSOM com Métodos de Mapeamento Contínuo por Partes

O treinamento do LARFSOM com alguma das técnicas de mapeamento contínuo por partes é executado de modo *online*. No laço de treinamento apresentado na Seção 4.1.2, dois passos são adicionados após o passo 7: (1) calcular a saída através do método escolhido e (2) ajuste dos pesos do método. O diagrama de fluxo apresentado na Figura 6 mostra os passos adicionais para o treinamento do LARFSOM com um método de mapeamento contínuo. O procedimento de treinamento do LARFSOM estendido de modo incremental é apresentado em Pseudocódigo 4.1.

<sup>7</sup> Os cálculos das derivadas para dedução das Equações 4.34 e 4.35 estão apresentados no Apêndice C.

<sup>8</sup> Os cálculos das derivadas para dedução da Equação 4.36 estão apresentados em (FLENTGE, 2006).

---

**Algoritmo 4.1:** Treinamento da rede LARFSOM estendido de modo incremental
 

---

**Dados:**  $S_{treinamento} = \{\xi_i\}, (i = 1, \dots, I)$  conjunto de treinamento

**Início**

$\rho_f \leftarrow$  taxa de aprendizagem final;  $\varepsilon \leftarrow$  modulador da taxa de aprendizagem;

$a_T \leftarrow$  limiar de atividade;  $v_{max} \leftarrow$  número máximo de vitórias;

$e_{min} \leftarrow$  erro mínimo;  $t_{max} \leftarrow$  número máximo de iterações;

$N_{max} \leftarrow$  número máximo de nodos;

$Mapa \leftarrow$  Inicialização(); //mapa com 2 nodos e conectados;

**para**  $n_i \in Mapa$  **fazer**

$\xi \leftarrow$  selecionar aleatoriamente de  $S_{treinamento}$ ;

$w_i \leftarrow \xi$ ; //Inicializar pesos

$v_i \leftarrow 0$ ; //Número de vitórias

**fim**

//Treinamento online;

**repita**

$\xi_i \leftarrow$  Selecionar um padrão de treinamento de  $S_{treinamento}$ ;

$n_{s_1} \leftarrow$  Selecionar o nodo mais próximo ao padrão de treinamento;

$n_{s_2} \leftarrow$  Selecionar o 2º nodo mais próximo ao padrão de treinamento;

$l \leftarrow$  comprimento do vetor  $w^{in}$ ;

$$r_{s_1} = \sqrt{\sum_{i=1}^l (w_{s_1}^{in}[i] - w_{s_2}^{in}[i])^2};$$

    //Calcular o campo receptivo de  $n_{s_1}$ ;

$$a_{s_1} = \frac{\exp(-\|\xi_i^{in} - w_{s_1}^{in}\|)}{r_{s_1}}; //Calcular a atividade de  $n_{s_1}$ ;$$

    //Inserir um nodo se a ativação do vencedor estiver abaixo do limiar;

**se**  $a_{s_1} < a_T$  e  $|N| < N_{max}$  **então**

$w_{novo} \leftarrow \xi_i$ ; //Inserir novo nodo;

        Remover conexões entre  $n_{s_1}$  e  $n_{s_2}$ ;

        Calcular distâncias entre  $n_{novo}$ ,  $n_{s_1}$  e  $n_{s_2}$ ;

        Inserir as duas conexões com menores distâncias;

**fim**

    //Caso contrário, atualizar os pesos do vencedor;

**senão faça**

        Inserir uma conexão entre  $n_{s_1}$  e  $n_{s_2}$  se ela não existir;

$v_{s_1} = v_{s_1} + 1$ ; //Incrementar contador de vitórias do vencedor;

$$\rho = \begin{cases} \varepsilon \times \rho_f^{\frac{v_{s_1}}{v_{max}}} & \text{se } v_{s_1} \leq v_{max} \\ \varepsilon \times \rho_f & \text{caso contrário} \end{cases}$$

$$\Delta w_{s_1} = \rho \times (\xi_i - w_{s_1}(t));$$

$$w_{s_1}(t+1) = w_{s_1}(t) + \Delta w_{s_1};$$

**fim**

$\xi_i^{out} \leftarrow$  mapeamentoContínuo();

    ajusteFinoMapeamentoContínuo();

até não existir novas amostras de treinamento;

**Fim**

---

### 4.3 CONCLUSÃO

Este capítulo apresenta o desenvolvimento de diferentes versões do LARFSOM para o contexto de sistemas não-lineares variantes no tempo. Nesse contexto, o modelo do sistema ou do controlador pode ser visto como um modelo que aproxima o comportamento de um função não-linear variante no tempo. Como apontado por Vijayakumar, D'Souza e Schaal (2005), um algoritmo de aprendizagem nesse contexto deve cumprir três requisitos: (1) permitir aprendizagem incremental para lidar com a saída de funções variante no tempo e com distribuição de entrada não estacionária (2) custo computacional baixo e (3) realizar aproximações acuradas.

Para construção do modelo que atenda esses requisitos, o LARFSOM foi estendido como uma memória associativa temporal por quantização vetorial (VQTAM), e métodos de mapeamento contínuo foram aplicados sobre o LARFSOM estendido para realizar aproximações mais acuradas. Dois métodos de mapeamento contínuo foram selecionados da literatura, que são a interpolação linear por partes generalizada (I) e interpolação linear localmente ponderada (LWI) (FLENTGE, 2006). Este trabalho apresenta duas técnicas, método da diferença (D) e método da diferença estendido (xD), para substituir a interpolação linear a fim de obter melhores aproximações com um menor esforço computacional. Essas duas técnicas com aprendizagem localmente ponderada são chamadas de método da diferença localmente ponderado (LWD) e método da diferença estendido localmente ponderado (LWxD). Desse modo, o processo de treinamento incremental combina a abordagem autossupervisionada<sup>9</sup> do LARFSOM estendido, o gradiente descendente e a aprendizagem localmente ponderada. Pode-se afirmar que o esquema de redução do erro da aprendizagem auto-supervisionada contribui para a aprendizagem de representações do espaço de entrada e saída. Enquanto o método do gradiente descendente e a aprendizagem localmente ponderada são utilizadas para melhorar a qualidade da aproximação final.

As principais contribuições apontadas neste capítulo são: (i) o desenvolvimento de uma versão estendida do LARFSOM para aprendizagem incremental em problemas de aproximação de funções não-lineares variantes no tempo e (ii) proposições de alternativas ao método LWI, estado da arte em termos de técnica de mapeamento contínuo para redes SOMs.

---

<sup>9</sup> Segundo Barreto e Araújo (2004), o termo *abordagem autossupervisionada* refere-se à aplicação da aprendizagem não-supervisionada no problema da aprendizagem supervisionada sem a necessidade da computação explícita de um sinal de erro.

## 5 ESTRATÉGIAS DE CONTROLE BASEADO NO LARFSOM

Neste capítulo, são apresentadas propostas de estratégias de controle apropriada para sistemas com modelos desconhecidos e com mudanças repentinas durante a execução. Primeiramente, é descrito o *controle por realimentação de estados baseado no LARFSOM*. Essa solução foi proposta a partir de uma investigação das possíveis utilizações do SOM em estratégias de controle. Foram considerados os modelos LARFSOM e LARFSOM com capacidade de mapeamento contínuo por partes para a construção de uma estratégia de controle em múltiplos modelos. Esse controle também é visto como um controle adaptativo por realimentação de estados. Nessa proposta, são utilizados dois LARFSOMs. Um LARFSOM é responsável por estimar o comando de controle direto enquanto que um segundo LARFSOM estima os ganhos da malha de realimentação de estados. Então, o sinal de controle direto é combinado com sinal da malha de realimentação para produzir a ação de controle a ser aplicada no sistema.

Em um segundo momento, é apresentado o *controle adaptativo direto baseado no LARFSOM*. A motivação desta segunda proposição é o desenvolvimento de um controlador baseado no LARFSOM com garantias de convergência. Essa segunda estratégia de controle consiste de um controle adaptativo direto sob a arquitetura de múltiplos modelos. Analogamente à primeira estratégia, na etapa de treinamento, o LARFSOM adquire informações sobre o comportamento do sistema e divide o espaço de operação em regiões. Nessa etapa, também é aprendido uma ação de controle para cada região. Na etapa de operação, o LARFSOM identifica o comportamento do sistema e seleciona o controlador local apropriado para o comportamento atual. Diferentemente da primeira estratégia, este controlador local consiste de um controlador adaptativo direto. A teoria de estabilidade de Lyapunov é usada para demonstrar a convergência do erro de rastreamento e todos os outros sinais do sistema em malha fecha. O desenvolvimento desta abordagem de controle considerou uma classe de sistemas não-lineares variantes no tempo.

### 5.1 CONTROLE POR REALIMENTAÇÃO DE ESTADOS BASEADO NO LARFSOM

Nessa seção, é apresentada uma estratégia de controle desenvolvida nesse trabalho. A descrição do modelo do controle nesta seção considera apenas o LARFSOM original, mas a abordagem apresentada é facilmente aplicável para as versões do LARFSOM com mapeamento contínuo por partes.

O sistema de controle proposto está esquematizado sobre a estratégia de controle por realimentação de estados. Um esquema genérico de um controle por realimentação de estados está apresentada na Figura 8.

Em uma etapa de pré-treinamento, controladores em realimentação de estados são de-

desenvolvidos para controlar a planta. No caso, os controladores são projetados considerando o sistema conhecido e diferentes controladores podem ser desenvolvidos para diferentes situações paramétricas do sistema. Estes sistemas de controle são simulados para produzir dados de treinamento. Neste trabalho, empregou-se o controlador linear-quadrático (LQR) para sistemas de controle para aquisição dos dados de treinamento.

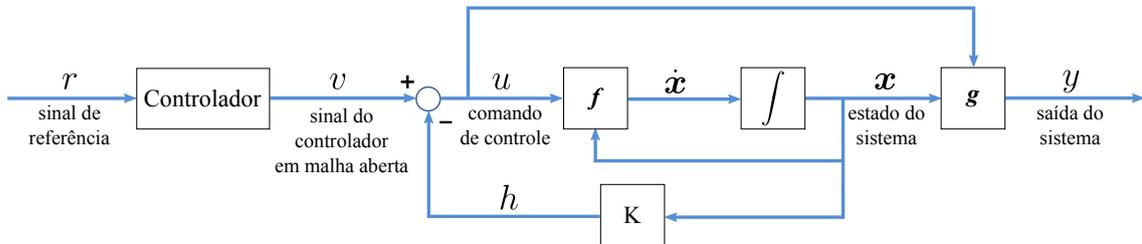


Figura 8 – Diagrama de blocos do controle por realimentação de estados.

A ideia central desta estratégia de controle é utilizar LARFSOMs para generalizar os diferentes sistemas de controle projetados para situações específicas nas quais se conheciam os parâmetros do sistema (apenas para fins de geração de dados para treinamento). Isso significa que os LARFSOMs, através do processo de treinamento, devem ser capazes de criar protótipos de diferentes dinâmicas do sistema e aprender ações de controle para cada caso. Logo, o novo sistema de controle baseado no LARFSOM deve poder controlar o sistema para as diferentes situações de controle simuladas na etapa de geração de dados de treinamento, e, também, devido à sua capacidade de generalização, para novas situações.

O controle adaptativo inteligente emprega dois blocos de LARFSOMs. O primeiro deles é aplicado para generalizar o controlador em malha aberta. Portanto, o propósito desse LARFSOM é estimar o sinal de controle em malha aberta  $v$ , que assegura que o erro de estado estacionário esteja em um nível aceitável. Esse LARFSOM é denotado por  $\text{LARFSOM}^v$ . Um segundo bloco de LARFSOM é responsável por estimar os ganhos da malha de realimentação  $\mathbf{K}$ . Ele é denotado por  $\text{LARFSOM}^k$ . A escolha por utilizar dois LARFSOMs se deve às diferentes características utilizadas para estimar o sinal  $v$  e para estimar os ganhos  $\mathbf{K}$ . A arquitetura do sistema de controle proposto está apresentada na Figura 9.

### 5.1.1 Vetor Concatenado de Entrada

Para que os LARFSOMs possam identificar a dinâmica do sistema, é preciso levar em consideração o fator temporal do comportamento do sistema. Isso pode ser feito através de um mecanismo de *memória de curto prazo*. Uma estrutura com atrasos no tempo, também conhecida por *sliding window*, é utilizada como mecanismo para armazenar uma sequência temporal de amostras dos sinais que é apresentada aos LARFSOMs como sua entrada. Isso permite que a sequência temporal de amostra seja tratada com uma amostra

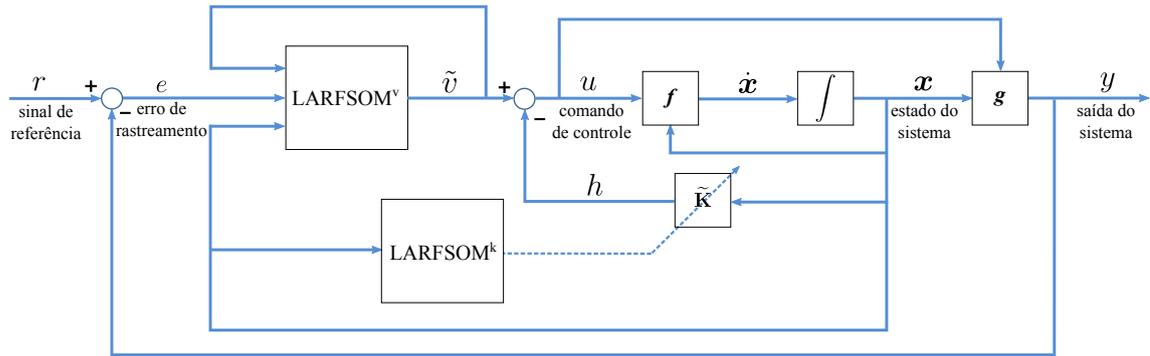


Figura 9 – Diagrama de blocos do sistema de controle usando LARFSOM.

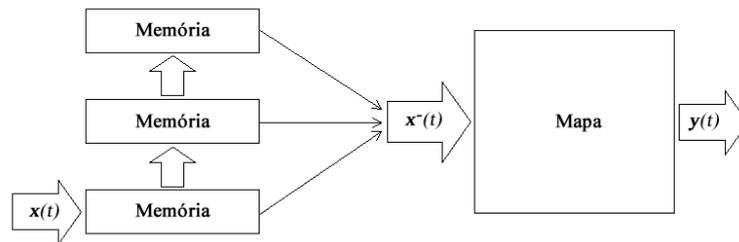


Figura 10 – O mapa auto-organizável treinado com padrões de entrada concatenados. No exemplo, três padrões de entrada são concatenados para formar o vetor de entrada da rede. Adaptado da Figura 2 de (KANGAS, 1990).

única ao invés de receber os padrões um a um. Como notação, essa estrutura é referida com  $\xi_{in}^-$  e será chamada de vetor concatenado de entrada. A Figura 10 apresenta a estrutura de armazenamento temporal.

Como dois LARFSOMs são aplicados, tem-se dois vetores concatenados de entrada distintos. Como descrito anteriormente, esses dois LARFSOMs demandam diferentes discriminantes, logo dois vetores concatenados de entrada distintos são construídos. A notação  $\xi_{in}^{-v}$  se referir ao vetor concatenado de entrada para o LARFSOM<sup>v</sup>. Enquanto a notação  $\xi_{in}^{-k}$  remete ao vetor LARFSOM<sup>k</sup>. Especificamente, os vetores  $\xi_{in}^{-v}$  e  $\xi_{in}^{-k}$  no instante  $t$  são definidos por:

$$\xi_{in}^{-v}[t] = \begin{bmatrix} \mathbf{x}^-[t] \\ \mathbf{e}^-[t] \\ \mathbf{v}^-[t] \end{bmatrix} \quad \xi_{in}^{-k}[t] = \mathbf{x}^-[t] \quad (5.1)$$

onde

$$\mathbf{x}^-[t] = \begin{bmatrix} \mathbf{x}[t] \\ \vdots \\ \mathbf{x}[t-p+1] \end{bmatrix} \quad \mathbf{e}^-[t] = \begin{bmatrix} \mathbf{e}[t] \\ \vdots \\ \mathbf{e}[t-p+1] \end{bmatrix} \quad \mathbf{v}^-[t] = \begin{bmatrix} \mathbf{v}[t-1] \\ \vdots \\ \mathbf{v}[t-p+1] \end{bmatrix} \quad (5.2)$$

onde  $\mathbf{x}[t]$  é o vetor de estado,  $\mathbf{e}[t]$  é o sinal de erro, e  $\mathbf{v}[t]$  é o sinal do controlador em malha

aberta. O tamanho da janela de tempo  $p$  é definido previamente. Técnicas podem auxiliar na escolha do parâmetro  $p$  como o índice de Lipschitz (HE; ASADA, 1993), porém, nos experimentos, a escolha foi feita através tentativas preliminares de usar o menor tamanho possível.

### Fase de Aprendizagem do Controle

No treinamento, tem-se um vetor  $\boldsymbol{\xi}_{out}[t]$  associado a cada  $\boldsymbol{\xi}_{in}^-[t]$ . No caso dos vetores  $\boldsymbol{\xi}_{in}^{-v}[t]$ , o vetor  $\boldsymbol{\xi}_{out}^{-v}[t]$  é composto pelo sinal  $\boldsymbol{v}[t]$  no instante  $t$ . Já para os vetores  $\boldsymbol{\xi}_{in}^{-k}[t]$ , o vetor  $\boldsymbol{\xi}_{out}^{-k}[t]$  é dado pelo ganhos de realimentação  $\boldsymbol{K}[t]$  do tempo  $t$ . Após os procedimentos treinamentos apresentados no Capítulo 4, o vetor de pesos de saída do LARFSOM<sup>v</sup> compreende uma estimativa do sinal  $\tilde{\boldsymbol{v}}$ , e o vetor de pesos de saída do LARFSOM<sup>k</sup>, uma estimativa do ganhos  $\tilde{\boldsymbol{K}}$ .

### Fase de Execução do Controle

O sinal de controle  $\boldsymbol{u}[t]$  é calculado usando o LARFSOM<sup>v</sup> e LARFSOM<sup>k</sup>. Assumindo que  $s_1^v$  denota o vencedor de LARFSOM<sup>v</sup> e  $s_1^k$  denota o vencedor de LARFSOM<sup>k</sup>, o sinal de controle  $\boldsymbol{u}[t]$  é definido por:

$$\boldsymbol{u}[t] = \tilde{\boldsymbol{v}}[t]_{s_1^v} - \tilde{\boldsymbol{K}}_{s_1^k}^T[t] \cdot \boldsymbol{x}[t] \quad (5.3)$$

onde  $\tilde{\boldsymbol{v}}[t]_{s_1^v}$  e  $\tilde{\boldsymbol{K}}_{s_1^k}^T[t]$  são as estimativas do sinal de controle em malha aberta e os ganhos de realimentação calculadas pelos modelos locais vencedores dos LARFSOM<sup>v</sup> e LARFSOM<sup>k</sup>, respectivamente.

#### 5.1.2 Discussão

A principal contribuição do *Controle por Realimentação de Estados Baseado no LARFSOM* está em generalizar controladores projetados para diferentes configurações paramétricas do sistema sobre a arquitetura de controle por realimentação de estados. O projeto de controle LARFSOM utiliza da auto-organização e do gradiente descendente para aprender os padrões de comportamento e as ações de controle a serem aplicadas em sistemas com configurações paramétricas apresentadas ou não na etapa de treinamento. Esse controle é capaz de responder as mudanças comportamentais inesperadas do sistema a ser controlado, mesmo não tendo informações sobre os novos parâmetros do sistema durante a execução. Além disso, esse controle pode também mapear comportamentos em ações que não estavam presentes no conjunto de amostras de treinamento através do poder de generalização do LARFSOM.

Narendra e Balakrishnan (1997) afirmaram que as características de estabilidade de sistemas com múltiplos modelos e comutação de controladores são substancialmente mais

---

difíceis de analisar. Mas, para garantir a estabilidade do sistema com comutação de controladores locais, cada possível configuração da planta em tempo de operação deve ser estabilizada por ao menos um dos controladores locais. Este trabalho assume que os controladores locais aprendidos são robustos e suficientes para garantir essa primeira afirmação. Isto pode ser assumido sem grandes consequências negativas, pois os reguladores lineares quadráticos possuem garantias de robustez (LEWIS, 2010). Então, o ponto central da estabilidade deste controle está na identificação do sistema. O LARFSOM aprende comandos de controle em malha aberta e ganhos da malha de realimentação capazes de controlar um determinado comportamento do sistema. Porém, identificações incorretas do comportamento ao longo da execução vão selecionar ações de controle incapazes de levar o sistema de controle a convergência, pois os comandos de controle foram gerados para serem estáveis para o outro ponto de operação. Então, o ponto crucial na estabilidade desse sistema de controle baseado no LARFSOM está na qualidade da discriminação do comportamento do sistema.

Um ponto importante a ser discutido é a competição do controle LARFSOM. Os LARFSOMs são capazes de aprender diferentes comportamentos do sistema. O processo de competição também vai desempenhar o papel de identificação do comportamento do sistema, e conseqüentemente, comutar as ações de controle assim como em alguns trabalhos citados na Subseção 3.3.1. A competição usa somente dados provenientes do espaço de entrada  $\xi_{in}^-(t)$  e os vetores de pesos de entrada  $w^{in}$  tanto na fase de treinamento quanto na fase de execução. Desta forma, a rede divide o espaço de operação de modo que comportamentos do sistema semelhantes, mesmo sendo provenientes de configurações paramétricas diferentes, possam ser sintetizadas em um mesmo módulo.

## 5.2 CONTROLE ADAPTATIVO DIRETO BASEADO NO LARFSOM

Nesta seção, é apresentada a segunda estratégia de controle para sistema com modelos desconhecidos. Esse segundo método de controle considerou um classe de sistemas de não-lineares variantes no tempo.

### 5.2.1 Classe de Sistemas Considerados

Seja a classe de sistemas não-lineares variantes no tempo expressos pela forma de Brunovsky (WANG; HILL, 2006):

$$\begin{aligned} \dot{x}_i &= x_{i+1}, & 1 \leq i \leq n-1 \\ \dot{x}_n &= f(\mathbf{x}, \mathbf{p}_f) + u \\ y &= x_1 \end{aligned} \quad (5.4)$$

onde  $y$  é a saída do sistema,  $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$  é o vetor de variáveis de estado,  $u$  é a entrada de controle e  $f(\mathbf{x}, \mathbf{p}_f)$  é uma função não-linear. O vetor  $\mathbf{p}_f$  consiste dos parâmetros da função  $f$  que variam ao longo do tempo  $t$ . No contexto de sistema com modelo desconhecido, tanto a função  $f$  quanto os parâmetros  $\mathbf{p}_f$  são desconhecidos.

Considere o seguinte modelo de referência:

$$\begin{aligned} \dot{x}_{di} &= x_{d(i+1)}, & 1 \leq i \leq n-1 \\ \dot{x}_{dn} &= f_d(\mathbf{x}_d) \\ y_d &= x_{d1} \end{aligned} \quad (5.5)$$

onde  $y_d$  representa a saída do modelo de referência,  $\mathbf{x}_d = [x_{d1} \ x_{d2} \ \cdots \ x_{dn}]^T$  é o estado desejado, e  $f_d(\cdot)$  é uma função não-linear conhecida. O objetivo é projetar uma lei de controle  $u$  de modo que a saída do sistema  $y = x_1$  siga a saída do modelo de referência  $y_d = x_{d1}$ .

Se a função  $f(\cdot)$  e os parâmetros  $\mathbf{p}_f$  fossem conhecidos, poderia ser aplicada a estratégia de controle *linearização por realimentação* através da seguinte lei de controle:

$$u = -f(\mathbf{x}, \mathbf{p}_f) + k_v r + \lambda_1 e^{(n-1)} + \cdots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn} \quad (5.6)$$

onde  $e = x_{d1} - x_1$  é o erro de rastreamento,  $r = e^{(n-1)} + \lambda_1 e^{(n-2)} + \cdots + \lambda_{n-1} e$  é erro filtrado e  $e^{(i)}$  é a  $i$ -ésima derivada de  $e$ . Com essa estratégia, o sistema em malha fechada se torna  $\dot{r} = -k_v r$ , isto é, linear e possivelmente estável (KAR; BEHERA, 2009).

### 5.2.2 Controle Adaptativo Direto Baseado no LARFSOM

Esta estratégia de controle utiliza apenas um LARFSOM em duas etapas: treinamento e operação. Na etapa de treinamento, o LARFSOM divide o espaço de comportamentos do sistema em regiões locais e organiza as informações sobre diferentes dinâmicas

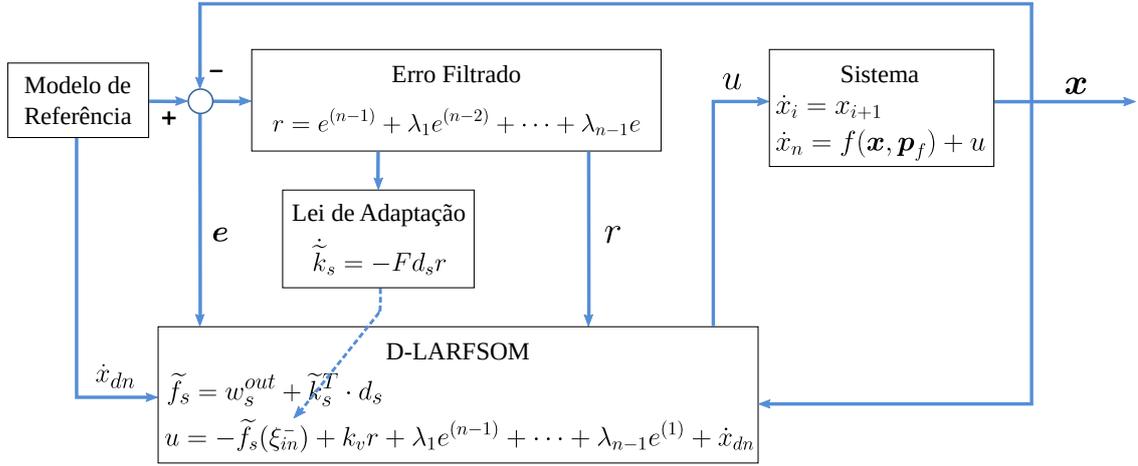


Figura 11 – Diagrama de blocos do sistema de controle adaptativo direto baseado no LARFSOM.

do sistema. Nessa etapa, o LARFSOM aprende uma ação de controle para cada região local. Na etapa de operação, o controlador baseado no LARFSOM atua considerando duas fases. Na primeira fase, o LARFSOM identifica a dinâmica do sistema e seleciona um controlador local. Na segunda fase, o controlador local se adapta e gera uma ação de controle apropriada para o comportamento identificado. A principal diferença dessa abordagem para a apresentada na Seção 5.1 está na adaptação direta dos controladores locais. Isso proporciona ao controlador um refinamento contínuo das ações de controle durante o regime de operação. A Figura 11 apresenta o diagrama de blocos do sistema de controle adaptativo direto baseado no LARFSOM.

Mais especificamente, a estratégia de controle adaptativo direto baseado no LARFSOM consiste em aplicar o D-LARFSOM para aproximar a função  $f(\mathbf{x}, \mathbf{p}_f)$  (Eq. 5.4). Considerando um sistema de uma única saída, o controlador local do nodo vencedor  $n_s$  pode ser expresso da seguinte forma<sup>1</sup>:

$$\tilde{f}_s(\boldsymbol{\xi}_{in}^-) = w_s^{out} + \tilde{\mathbf{k}}_s^T \cdot \mathbf{d}_s \quad (5.7)$$

onde  $\tilde{\mathbf{k}}_s$  é o vetor de pesos do nodo vencedor  $n_s$ , e  $\mathbf{d}_s$  é o vetor diferença ( $\mathbf{d}_s = \boldsymbol{\xi}_{in}^- - \mathbf{w}_s^{in}$ ). No caso, o vetor concatenado de entrada  $\boldsymbol{\xi}_{in}^-$  consiste de uma sequência temporal de estados do sistema mais os parâmetros do sistema que gerou a sequência, isto é:

$$\boldsymbol{\xi}_{in}^-[t] = \begin{bmatrix} \mathbf{x}[t] \\ \vdots \\ \mathbf{x}[t-p+1] \\ \mathbf{p}_f \end{bmatrix} \quad (5.8)$$

<sup>1</sup>  $\tilde{f}_s(\boldsymbol{\xi}_{in}^-) = w_s^{out} + \tilde{\mathbf{k}}_s^T \cdot \mathbf{d}_s$  seria expresso por  $\tilde{M}_s(\boldsymbol{\xi}_{in}^-) = w_s^{out} + \mathbf{k}_s^T \cdot \mathbf{d}_s$  na notação do Capítulo 4. Uma nova notação foi escolhida para facilitar o entendimento da prova de convergência.

Vale ressaltar que os parâmetros do sistema  $\mathbf{p}_f$  são desconsiderados durante a fase de operação, visto que o sistema é desconhecido. Essa informação é apenas usada durante o treinamento. Sendo assim, a distância  $d$  na Equação 4.2 durante o treinamento foi alterada para:

$$d(\mathbf{w}_i^{in}, \boldsymbol{\xi}_{in}^-) = \begin{cases} \|\mathbf{w}_i^{in} - \boldsymbol{\xi}_{in}^-\|^2 & \text{se } \|\tilde{\mathbf{p}}_i - \mathbf{p}_f\|^2 = 0 \\ \infty & \text{caso contrário} \end{cases} \quad (5.9)$$

Essa modificação foi feita para promover uma divisão de possíveis configurações do sistema, e partir disso, uma divisão do espaço de comportamento do sistema. Isso pode proporcionar uma aprendizagem mais detalhada de possíveis comportamentos já que dinâmicas semelhantes de configurações de sistemas diferentes serão modelados em nodos diferentes.

### 5.2.3 Análise de Estabilidade

Seguindo os mesmos passos de Kar e Behera (2009), o Teorema 1 sumariza o resultado da análise de estabilidade.

**Teorema 1.** *Suponha que a função  $f(\mathbf{x}, \mathbf{p}_f)$  da Equação 5.4 é aproximada por  $\tilde{f}_s(\boldsymbol{\xi}_{in}^-)$  usando o D-LARFSOM. Então, a lei de controle  $u = -\tilde{f}_s(\boldsymbol{\xi}_{in}^-) + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn}$  estabiliza o sistema da Eq. 5.4 no sentido de Lyapunov caso o vetor de pesos  $\tilde{\mathbf{k}}_s$  seja ajustado pela lei de adaptação  $\dot{\tilde{\mathbf{k}}}_s = -\mathbf{F} \mathbf{d}_s r$ , onde  $\mathbf{F}$  é uma matriz definida positiva.* —

*Demonstração.* Seja o erro de rastreamento filtrado  $r$  definido como:

$$r = e^{(n-1)} + \lambda_1 e^{(n-2)} + \dots + \lambda_{n-1} e \quad (5.10)$$

onde  $e = x_{d1} - x_1$  é o erro de rastreamento e  $e^{(i)}$  é a  $i$ -ésima derivada de  $e$ . Os parâmetros  $\lambda_1, \lambda_2, \dots, \lambda_{n-1}$  são escolhidos de tal modo que o sistema representado pela Eq. 5.10 apresente um comportamento convergente. Considerando a seguinte lei de controle  $u$ :

$$u = -\tilde{f}_s(\boldsymbol{\xi}_{in}^-) + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn} \quad (5.11)$$

O objetivo é garantir uma propriedade de convergência uniforme para o erro de rastreamento  $r$  e para o vetor de pesos  $\tilde{\mathbf{k}}_i$ .

Suponha que existe um vetor de pesos  $\mathbf{k}_s$  que a função  $f(\mathbf{x}, \mathbf{p}_f)$  pode ser modelada com  $f(\mathbf{x}, \mathbf{p}_f) = w_s^{out} + \mathbf{k}_s^T \cdot \mathbf{d}_s$ , onde  $s$  denota o nodo vencedor  $n_s$ . Aplicando a lei de controle  $u$  no sistema (Eq. 5.4):

$$\dot{x}_n = f(\mathbf{x}, \mathbf{p}_f) + [-\tilde{f}_s(\boldsymbol{\xi}_{in}^-) + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn}] \quad (5.12)$$

$$\dot{x}_n = w_s^{out} + \mathbf{k}_s^T \mathbf{d}_s - w_s^{out} - \tilde{\mathbf{k}}_s^T \mathbf{d}_s + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn} \quad (5.13)$$

Definindo  $\mathbf{E}_{\mathbf{k}_s}^T = \mathbf{k}_s^T - \tilde{\mathbf{k}}_s^T$ , a Equação 5.13 pode ser escrita da seguinte maneira:

$$\dot{x}_n = \mathbf{E}_{\mathbf{k}_s}^T \mathbf{d}_s + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dot{x}_{dn} \quad (5.14)$$

e

$$e^{(n)} = \dot{x}_{dn} - \dot{x}_n \quad (5.15)$$

$$e^{(n)} = -\mathbf{E}_{\mathbf{k}_s}^T \mathbf{d}_s - k_v r - \lambda_1 e^{(n-1)} - \dots - \lambda_{n-1} e^{(1)} \quad (5.16)$$

Derivando o erro de rastreamento filtrado (Eq. 5.10):

$$\dot{r} = e^{(n)} + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} \quad (5.17)$$

Substituindo  $e^{(n)}$  da Eq. 5.17 na Eq. 5.16:

$$\dot{r} = -k_v r - \mathbf{E}_{\mathbf{k}_s}^T \mathbf{d}_s \quad (5.18)$$

Considerando a seguinte função de Lyapunov candidata:

$$V = \frac{1}{2} r^2 + \frac{1}{2} \mathbf{E}_{\mathbf{k}_s}^T \mathbf{F}^{-1} \mathbf{E}_{\mathbf{k}_s} \quad (5.19)$$

Derivando  $V$  tem-se:

$$\dot{V} = r \dot{r} + \mathbf{E}_{\mathbf{k}_s}^T \mathbf{F}^{-1} \dot{\mathbf{E}}_{\mathbf{k}_s} \quad (5.20)$$

Substituindo  $\dot{r}$  da Eq. 5.18,

$$\dot{V} = r \left( -k_v r - \mathbf{E}_{\mathbf{k}_s}^T \mathbf{d}_s \right) + \mathbf{E}_{\mathbf{k}_s}^T \mathbf{F}^{-1} \dot{\mathbf{E}}_{\mathbf{k}_s} \quad (5.21)$$

Como  $\mathbf{k}_s$  é constante,  $\dot{\mathbf{E}}_{\mathbf{k}_s} = -\dot{\tilde{\mathbf{k}}}_s$

$$\dot{V} = -k_v r^2 - \mathbf{E}_{\mathbf{k}_s}^T \left( \mathbf{d}_s r + \mathbf{F}^{-1} \dot{\tilde{\mathbf{k}}}_s \right) \quad (5.22)$$

Fazendo o segundo termo da Eq. 5.22 igual a zero,

$$0 = \mathbf{d}_s r + \mathbf{F}^{-1} \dot{\tilde{\mathbf{k}}}_s \quad (5.23)$$

$$\dot{\tilde{\mathbf{k}}}_s = -\mathbf{F} \mathbf{d}_s r \quad (5.24)$$

Usando a regra de adaptação Eq. 5.24, a Eq. 5.20 fica

$$\dot{V} = -k_v r^2 \quad (5.25)$$

Assim, a lei de controle descrita pela Equação 5.11 e a lei de adaptação descrita pela Equação 5.24 garante que  $V > 0$ ,  $\dot{V} \leq 0$  e  $V \rightarrow \infty$  quando  $\|r\| \rightarrow \infty$  e  $\|\mathbf{E}_{\mathbf{k}_s}\| \rightarrow \infty$  tornando a Equação 5.19 uma função de Lyapunov. Assim, temos garantias que o sistema é globalmente estável no sentido de Lyapunov.

□

### 5.2.3.1 Incluindo erro de aproximação

Agora, suponha que a função  $f(\mathbf{x}, \mathbf{p}_f)$  pode ser modelada por  $f(\mathbf{x}, \mathbf{p}_f) = w_s^{out} + \mathbf{k}_s^T \cdot \mathbf{d}_s + \epsilon$ , onde  $\epsilon$  é um erro de aproximação diferente de zero que é limitado por um valor positivo suficientemente pequeno  $\|\epsilon\| < \epsilon_N$ . Considerando essa modificação, a derivada de  $V$  fica:

$$\dot{V} = r \left( -k_v r - \mathbf{E}_{\mathbf{k}_s}^T \mathbf{d}_s - \epsilon \right) + \mathbf{E}_{\mathbf{k}_s}^T \mathbf{F}^{-1} \dot{\mathbf{E}}_{\mathbf{k}_s} \quad (5.26)$$

Usando a lei de adaptação  $\dot{\mathbf{k}}_s = -\mathbf{F} \mathbf{d}_s r$

$$\dot{V} = -k_v r^2 - \epsilon r \quad (5.27)$$

$\dot{V}$  pode ser expresso por:

$$\dot{V} \leq -k_v r^2 + \|\epsilon r\| \quad (5.28)$$

e

$$-k_v r^2 + \|\epsilon r\| = -k_v r^2 + \epsilon_N \|r\| \quad (5.29)$$

$$= -\|r\| (k_v \|r\| - \epsilon_N) \quad (5.30)$$

$\dot{V}$  será negativa se o termo  $k_v \|r\| - \epsilon_N$  for positivo. Então,

$$\begin{aligned} k_v \|r\| - \epsilon_N &> 0 \\ \|r\| &> \frac{\epsilon_N}{k_v} \end{aligned} \quad (5.31)$$

Isso implica que  $\dot{V}$  será definida negativa enquanto  $\|r\| = \epsilon_N/k_v$ . Logo, o erro de rastreamento filtrado  $r$  pode vir a ser arbitrariamente pequeno escolhendo um  $k_v$  suficientemente grande.

## 6 VALIDAÇÃO

Neste capítulo, há uma investigação da qualidade da aproximação realizada pelos métodos de mapeamento contínuo incorporados ao LARFSOM. Simulações foram executadas em diferentes casos a fim de avaliar a acurácia do mapeamento e aprendizagem. Foram considerados os seguintes cenários:

1. Aproximação de função com saída estática (Experimentos 1.1, 1.2 e 1.3);
2. Aproximação de função com saída variante no tempo (Experimentos 2.1 e 2.2);
3. Distribuição de entrada variante no tempo (Experimento 3);
4. *Benchmark* de aproximação de função (Experimentos em *benchmarks*).

### 6.1 DESCRIÇÃO DO EXPERIMENTOS

#### 6.1.1 Dados Sintéticos

A função usada para avaliar a acurácia do mapeamento foi:

$$f(x_1, x_2) = \max\{e^{-10x_1^2}, e^{-50x_2^2}, 1,25 \cdot e^{-5(x_1^2+x_2^2)}\} \quad (6.1)$$

Um ruído gaussiano  $\mathcal{N}(0; 0,01)$  foi adicionado na saída da função durante o treinamento para gerar uma leve dificuldade em aproximar a função. Essa função ruidosa foi aplicada por Vijayakumar, D'Souza e Schaal (2005), e Flentge (2006) em trabalhos anteriores para avaliar o LWPR e LWINGG, respectivamente. Na Figura 12, é apresentada a superfície gerada por essa função sem o ruído. Pode-se observar que a função gera uma superfície com valores maiores que 1 na região central, com valores constantes em certas áreas, e regiões com rápida mudança de valores.

Para avaliar a aproximação de função em um cenário de aprendizagem incremental, cada instância de treinamento é gerada aleatoriamente de modo *online*, considerando potenciais mudanças na função durante o tempo de treinamento. É importante ressaltar que o exemplo de treinamento processado é descartado após o mapa utilizá-lo.

Em cada experimento, 100k passos de treinamento são realizados. A cada 1k passos, o erro quadrático médio normalizado (*normalised Mean Square Error* - nMSE) é calculado sobre um gradil de 41x41 em  $[-1, 1]^2$ . Este experimento é repetido 100 vezes. Mais detalhadamente, cada iteração de treinamento consiste em:

1. Um vetor de atributos de entrada  $[x_1 \ x_2]^T$  é aleatoriamente gerado no espaço  $[-1, 1]^2$ ;
2. A saída  $f(x_1, x_2)$  é calculada e o exemplo de treinamento com ruído  $\xi = [x_1 \ x_2 \ (f(x_1, x_2) + \mathcal{N}(0, 0,01))]^T$  é gerado;

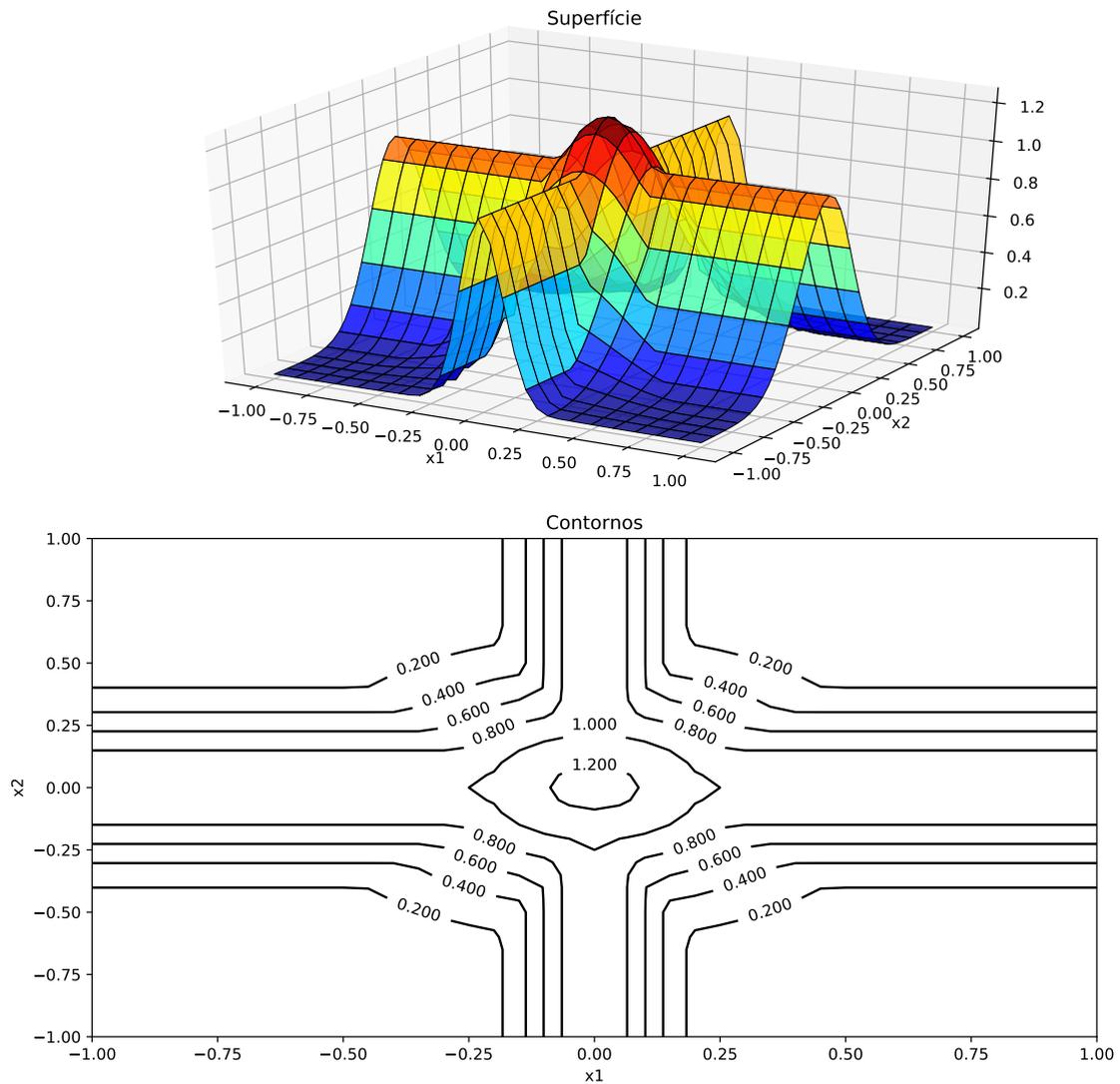


Figura 12 – Função a ser aproximada.

3. A rede é treinada com o padrão  $\xi$ ;
4. A saída aproximada é computada através de uma técnica de mapeamento contínuo;
5. Os pesos do método de mapeamento contínuo são refinados;
6. O exemplo de treinamento  $\xi$  é descartado.

De acordo com Maltoni e Lomonaco (2019), os experimentos de dados sintéticos realizados nesta tese podem ser vistos como cenários de aprendizagem incremental do tipo Novas Instâncias (*New Instances* - NI).

#### 6.1.1.1 Função de Saída Estática

Para aproximação de função de saída estática, três experimentos foram considerados:

**Exp 1.1** - O vetor de atributos de entrada sendo um vetor bidimensional composto por  $\xi^{in} = [x_1 \ x_2]^T$ ;

**Exp 1.2** - O vetor de entrada do Exp 1.1 é concatenado com um vetor de oito valores constantes, e o vetor 10-D resultante é multiplicado por uma matriz de rotação aleatória<sup>1</sup>. Isso é feito para avaliar sobre dados de entrada redundantes. O vetor de atributos de entrada fica  $\xi^{in} = R \times [[x_1 \ x_2][c \ c \ c \ c \ c \ c \ c \ c]]^T$ ;

**Exp 1.3** - O vetor de entrada do Exp 1.2 é concatenado com um vetor com dez valores aleatórios de uma distribuição gaussiana  $\mathcal{N}(0, 0,025)$ . Isso é feito para avaliar os métodos sobre dados de entrada irrelevantes.

Os Experimentos 1.2 e 1.3 objetivam avaliar a capacidade dos métodos em recuperar a função original de entrada bidimensional, que agora está incorporada em espaço de maior dimensão, sem a utilização de técnicas de redução de dimensionalidade adicionais. Uma técnica de aprendizagem robusta deverá preservar a acurácia das aproximações feitas no Experimento 1.1.

#### 6.1.1.2 Função com Saída Variante no Tempo

Para aproximação de função com saída variante no tempo, dois experimentos foram considerados:

**Exp 2.1** - *Saída aumentando lentamente*: A saída da função com ruído  $f(x_1, x_2) + \mathcal{N}(0, 0,01)$  é multiplicada pelo fator  $t/50000$  durante os primeiros 50k passos, onde  $t$  é a quantidade de passos de treinamento executados até o momento. Depois de 50k passos de treinamento, o mapa é treinado com função sem ruído  $f(x_1, x_2)$ . Então, a função com saída variante no tempo é dada por:

$$f_{exp2.1}(x_1, x_2, t) = \begin{cases} (t/50000) \cdot (f(x_1, x_2) + \mathcal{N}(0, 0,01)) & t \leq 50000 \\ f(x_1, x_2) & t > 50000 \end{cases} \quad (6.2)$$

onde  $f(x_1, x_2)$  é dada pela Eq. 6.1. Nesse experimentos, o nMSE é calculado usando a função  $f_{exp2.1}$  sem o ruído  $\mathcal{N}(0, 0,01)$ .

**Exp 2.2** - *Ruído decrescendo lentamente*: Um ruído gaussiano variante no tempo  $\mathcal{N}(0, 1 - t/50000)$  é adicionado a saída da função  $f(x_1, x_2)$  durante os primeiros 50k passos. Esse ruído gaussiano decresce lentamente a variância de 1 até zero. Depois de 50k passos de treinamento, o mapa é treinado com a função sem ruído  $f(x_1, x_2)$ . Nesse

<sup>1</sup> A matriz de rotação aleatória  $R$  é calculada pelo comando `qr(randn(10))` presente nas linguagens Matlab, Octave, e Julia.

caso, o nMSE é calculado com a função sem ruído  $f(x_1, x_2)$  durante toda a fase de treinamento. Então, a função com saída variante no tempo é dada por:

$$f_{exp2.2}(x_1, x_2, t) = \begin{cases} f(x_1, x_2) + \mathcal{N}(0, 1 - t/50000) & t \leq 50000 \\ f(x_1, x_2) & t > 50000 \end{cases} \quad (6.3)$$

onde  $f(x_1, x_2)$  é dada pela Eq. 6.1. Esse experimento avalia a interferência do conhecimento ruidoso aprendido previamente na adaptação de mudanças sobre a saída da função.

### 6.1.1.3 Distribuição de Entrada Variante no Tempo

**Exp 3.1** - Esse experimento se baseia no vetor de entrada de 10-D do *Exp 1.2*. Nesse experimento, a constante  $c$  se torna um valor variante no tempo durante os primeiros 50k passos de treinamento. Depois de 50k passos,  $c$  se torna uma constante. Matematicamente, pode-se representar pela Equação 6.4.

$$c(t) = \begin{cases} \frac{t}{50000}c_1 + \left(1 - \frac{t}{50000}\right)c_2 & t \leq 50000 \\ c_2 & t > 50000 \end{cases} \quad (6.4)$$

onde  $c_1$  e  $c_2$  são valores aleatórios em  $[0, 1]$ . O valor  $c(t)$  é usado para construir o vetor de entrada  $\xi^{in} = R \times [[x_1 \ x_2][c(t) \ c(t) \ c(t)]]^T$ , onde  $R$  é uma matriz de rotação aleatoriamente gerada.

### 6.1.1.4 Ajuste de Parâmetros

Para escolher os parâmetros de treinamento para LARFSOM, diversos valores foram experimentados para limiar de ativação  $a_T$ , taxa de aprendizagem final  $\rho_f$  e modulador da taxa de aprendizagem  $\varepsilon$  utilizando apenas o LARFSOM no *Exp1.1*. Os valores  $a_T = 1,65$ ,  $\rho_f = 0,001$  e  $\varepsilon = 0,4$  foram selecionados e aplicados em todos os outros experimentos. Também, foram usados os valores sugeridos no artigo original (ARAÚJO; COSTA, 2009) para a máxima quantidade de vitórias  $v_{max}$  nos experimentos *Exp1.1*, *Exp1.2* e *Exp1.3*. Entretanto, esse parâmetro foi alterado para  $v_{max} = 2000$  nos *Exp2.1*, *Exp2.2* e *Exp3.1* visto que o mapa precisa acompanhar o comportamento variante durante a execução do experimento. Para os experimentos *Exp1.1*, *Exp1.2*, *Exp1.3*, *Exp2.1* e *Exp2.2*, o número máximo de nodos  $N_{max}$  é 64 (mesmo valor aplicado em (FLENTGE, 2006)), e para *Exp3.1*,  $N_{max} = 225$ . Os valores sugeridos  $\alpha_\lambda = 1000$  e  $\lambda_{init} = 30$  de (FLENTGE, 2006) foram empregados para os métodos localmente ponderados. Também, a taxa de aprendizagem  $\alpha_k$  dos métodos da diferença e da diferença estendido foi variada entre  $[0,01 \ 1]$ . As taxas de aprendizagem usadas foram  $\alpha_k = 0,7$  para o método da diferença, e  $\alpha_k = 0,1$  para o método da diferença estendido.

Tabela 3 – Principais características das bases de dados de problemas de *benchmarks*.

Problemas	# de instâncias	# de atributos	Descrição
Abalone	4177	7 contínuas e 1 nominal <sup>a</sup>	idade do abalone a partir das medidas físicas
Boston Housing	506	13 contínuas	valores imobiliários no subúrbio de Boston
Auto Price	159	14 contínuas e 1 inteira	preço do carro a partir de suas características
Servo	167	4 nominais <sup>a</sup>	fenômeno não-linear em um experimento servomecânico
Machine CPU	209	6 contínuas	desempenho relativo da CPU
Auto-MPG	398	4 contínuas e 3 nominais <sup>a</sup>	consumo de combustível na cidade em milhas por galão (mpg)
Bank (8FM)	8192	8 contínuas	taxa de rejeições em um banco por causa da fila cheia
Kin. of Robot Arm (Kin-8nm)	8192	8 contínuas	Cinemática direta de um braço robótico

<sup>a</sup> Cada atributo nominal foi transformado em um vetor binário.

### 6.1.2 Problemas de *Benchmarks*

As abordagens propostas e da literatura foram desenvolvidas para aproximação de funções não-lineares variantes no tempo que naturalmente demandam aprendizagem incremental. No entanto, pode-se usar no modo tradicional da aprendizagem em lote (*off-line*) para análise. Oito problemas de *benchmarks* de regressão foram considerados: seis são do repositório de dados para aprendizagem de máquina da UCI (BACHE; LICHMAN, 2013), e dois outros são do repositório DELVE (RASMUSSEN et al., 1996).<sup>2</sup> Foram consideradas bases de dados de regressão populares tais como as bases *Auto-MPG* e *Boston Housing*, como também, bases ruidosas como *Bank* (8FM) e *Kinematic of Robot Arm* (Kin-8nm). As principais características das bases de dados usadas estão descritas na Tabela 3.

Esses experimentos utilizam uma análise de aprendizagem em lote. Então, cada experimento consiste de 30 repetições independentes de treinamento do mapa. Em cada repetição, foi empregada validação cruzada do tipo *k-fold*, com  $k = 5$ . Sendo assim, quatro partições de dados foram aplicadas para treinamento e uma, para teste. Todos os valores de entrada foram normalizadas dentro do intervalo  $[-1, 1]$  enquanto a saída foi normalizada entre  $[0, 1]$ . O desempenho foi mensurado através da raiz do erro quadrático médio (RMSE). Aqui, os métodos usando o LARFSOM também foram comparados com a máquina de vetores de suporte para regressão (SVR), uma técnica de aprendizagem de máquina aplicada com sucesso em aproximação de funções (BASAK; PAL; PATRANABIS, 2007; SMOLA; SCHÖLKOPF, 2004). A função de *kernel* utilizada foi a função de base radial  $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$ . O valor de  $\gamma$  é igual a  $1/(\text{número de atributos})$  e o parâmetro de margem  $C$  é selecionado empiricamente. Foram tentados os seguintes valores para parâmetro de margem  $C \in \{2^{12}, 2^{11}, \dots, 2^{-1}, 2^{-2}\}$ . O valor de  $C$  que apresentou melhor resultado foi selecionado. Os resultados foram avaliados comparativamente através do teste estatístico não paramétricos de Friedman e o teste de Wilcoxon *post-hoc* com intervalo de confiança de 95%.

Todos os experimentos para o LARFSOM com mapeamento contínuo por partes foram implementados em Julia v0.5.2 (BEZANSON et al., 2012; BEZANSON et al., 2017). Os expe-

<sup>2</sup> As bases de dados podem ser acessadas através do site <<http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html>>

Tabela 4 – Parâmetros e valores dos métodos utilizados na aproximação de funções de problemas de *benchmarks*.

Método	Parâmetro	valor	Descrição
SVR	$K(\mathbf{x}_i, \mathbf{x}_j)$	$e^{-\gamma\ \mathbf{x}_i-\mathbf{x}_j\ ^2}$	função de kernel
	$\gamma$	$1/(\# \text{ de atributos})$	valore de gama
	$C$	$C \in \{2^{12}, 2^{11}, \dots, 2^{-1}, 2^{-2}\}$	valor de margem
LARFSOM	$\rho_f$	0,001	taxa de aprendizagem final
	$\epsilon$	0,4	modulador da taxa de aprendizagem
	$a_T$	1,65	limiar de atividade
	$v_m$	100	máxima quantidade de vitórias de um nodo
	$t_{max}$	$t_{max} \in \mathbb{N}$	número máximo de iterações*
	$N_{max}$	$N_{max} \in \mathbb{N}$	número máximo de nodos*
LWI	$\alpha_{w^{out}}$	0,1	taxa de aprendizagem do valor de saída
	$\mu$	0,1	parâmetro da regularização de Tikhonov
	$\lambda_{init}$	$\lambda_{init} \in \mathbb{N}$	valor inicial para o parâmetro de abrangência*
	$\alpha_l$	$\alpha_l \in \mathbb{R}_{\geq 0}$	taxa de aprendizagem do parâmetro de abrangência*
LWD e LWxD	$\alpha_k$	[0,1 0,01 0,001]	taxa de aprendizagem do vetor de pesos $\mathbf{k}^*$
	$\lambda_{init}$	$\lambda_{init} \in \mathbb{N}$	valor inicial para parâmetro de abrangência*
	$\alpha_l$	$\alpha_l \in \mathbb{R}_{\geq 0}$	taxa de aprendizagem do parâmetro de abrangência*

Os valores exatos para os parâmetros com \* estão na tabela 6.

rimentos usando SVR foram executados através de um *wrapper* do LIBSVM 3.22 para Julia.<sup>3</sup> A LIBSVM (CHANG; LIN, 2011) é uma biblioteca codificada em C/C++ de SVM e contém duas versões básicas SVM para regressão: epsilon-SVR e nu-SVR (SCHÖLKOPF et al., 2000). Nos experimentos, foi aplicado o epsilon-SVR. Mais detalhes sobre epsilon-SVR podem ser encontrados em (SMOLA; SCHÖLKOPF, 2004).

É importante mencionar que os experimentos utilizando LIBSVM vão ser naturalmente mais rápidos do que os experimentos envolvendo o LARFSOM. Há dois motivos para isso, a implementação do LIBSVM é em C/C++ e foi bastante otimizada ao longo das diferentes versões. Já os algoritmos implementados neste trabalho não foram otimizados. Isso poderia ser feito, por exemplo, através busca otimizada do nodo vencedor usando método de acesso espacial (DENNY; GOZALI; MANURUNG, 2016) e codificação otimizada para a linguagem Julia (EKRE et al., 2019).

### 6.1.2.1 Ajuste dos Parâmetros

Os parâmetros podem ser divididos em duas categorias principais: os parâmetros referente ao LARFSOM e os parâmetros dos métodos usados para mapeamento contínuo. O procedimento para ajuste dos parâmetros consiste em:

1. Encontrar os parâmetros do LARFSOM que minimizam o erro de aproximação (RMSE);

<sup>3</sup> Link do *wrapper* usado nos experimentos <https://github.com/madsjulia/SVR.jl>.

2. Encontrar as taxas de aprendizagem  $\alpha_{w^{out}}$  (LWI) ou  $\alpha_k$  (LWD e LWxD) que minimizam o erro do mapeamento contínuo do LARFSOM sem a aprendizagem localmente ponderada;
3. Encontrar os parâmetros para a aprendizagem localmente ponderada  $\lambda_{init}$  e  $\alpha_l$  utilizando os parâmetros selecionados nas passos anteriores.

Para os parâmetros do LARFSOM, diferentes valores para os parâmetros *número máximo de nodos*  $N_{max}$  e *número máximo de iterações*  $t_{max}$  foram testados para cada base de dados. Experimentos preliminares mostraram que o LARFSOM é mais sensível a escolha desses dois parâmetros em cenários estacionários de aprendizagem em lote. Parâmetros como *taxa de aprendizagem final*  $\rho_f$ , *modulador da taxa de aprendizagem*  $\epsilon$  e *máxima quantidade de vitórias*  $v_m$  são mais importantes para aprendizagem incremental. A Tabela 4 sumariza todos os parâmetros.

Para os parâmetros dos métodos de computação da saída:

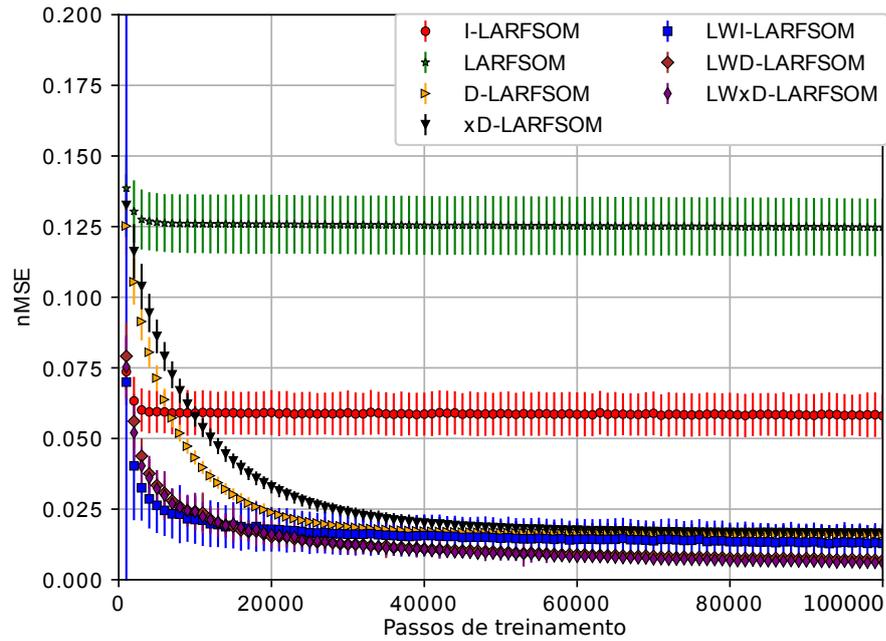
- A taxa de aprendizagem  $\alpha_k$  pode ser diferente entre os métodos da diferença e da diferença estendido. Preliminarmente, os experimentos usaram dois valores diferentes de  $\alpha_k$  e escolheu-se o valor que minimizou o erro de aproximação dos métodos sem a aprendizagem localmente ponderada. Os valores considerados foram:  $\alpha_k = 0,1$  e  $\alpha_k = 0,01$  para o método da diferença, e  $\alpha_k = 0,01$  e  $\alpha_k = 0,001$  para o método da diferença estendida.
- A escolha do valor inicial de  $\lambda_{init}$  e da taxa de aprendizagem  $\alpha_l$  foi conduzida da maneira sugerida em (FLENTGE, 2006). Primeiramente, valores diferentes para  $\lambda_{init}$  usando  $\alpha_l = 0$  foram testados e foi selecionado o melhor valor. Depois, ajustamos o parâmetro  $\alpha_l$  para obter a melhor aproximação.

## 6.2 RESULTADOS EXPERIMENTAIS

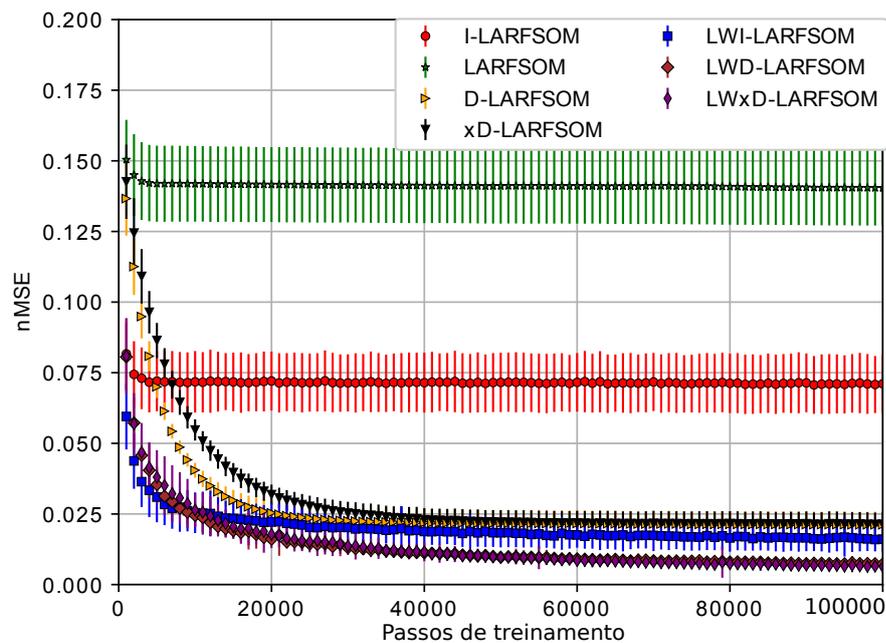
Esta seção apresenta os resultados do LARFSOM para aproximação de funções usando diferentes técnicas.

### 6.2.1 Resultados Comparativos em Aproximação de Função com Saída Estática

Os desempenhos dos LARFSOMs para aproximação de função foram primeiramente analisadas comparativamente sobre os experimentos *Exp1.1*, *Exp1.2* e *Exp1.3*. As Figuras 13a, 13b e 14a mostram a curva de aprendizagem média, e a Figura 14b, o tempo médio gasto em 100 simulações dos respectivos experimentos ao longo de 100k passos. Os resultados dos três experimentos são similares. A média do nMSE de cada método após 100k passos de treinamento estão apresentadas na Tabela 5.



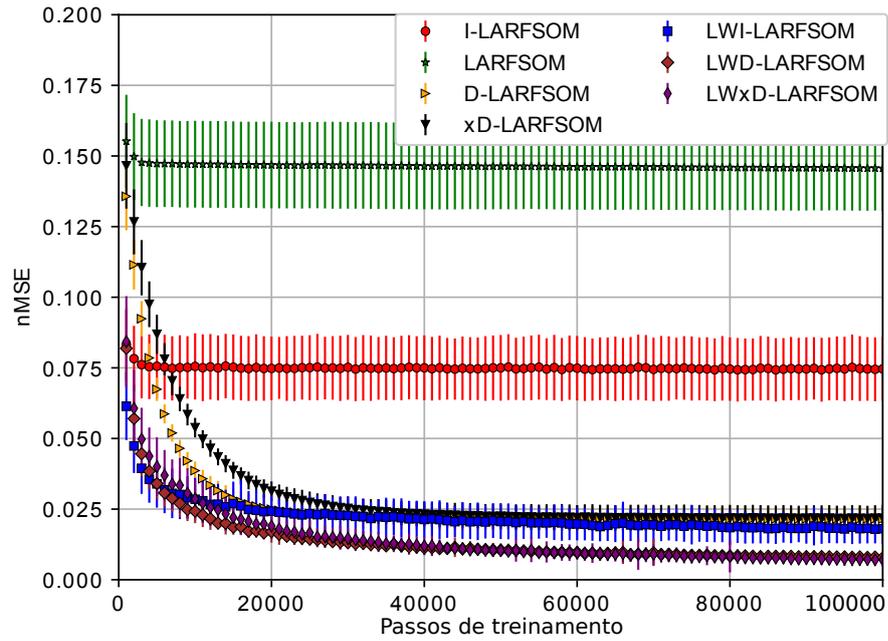
(a) Entrada 2D



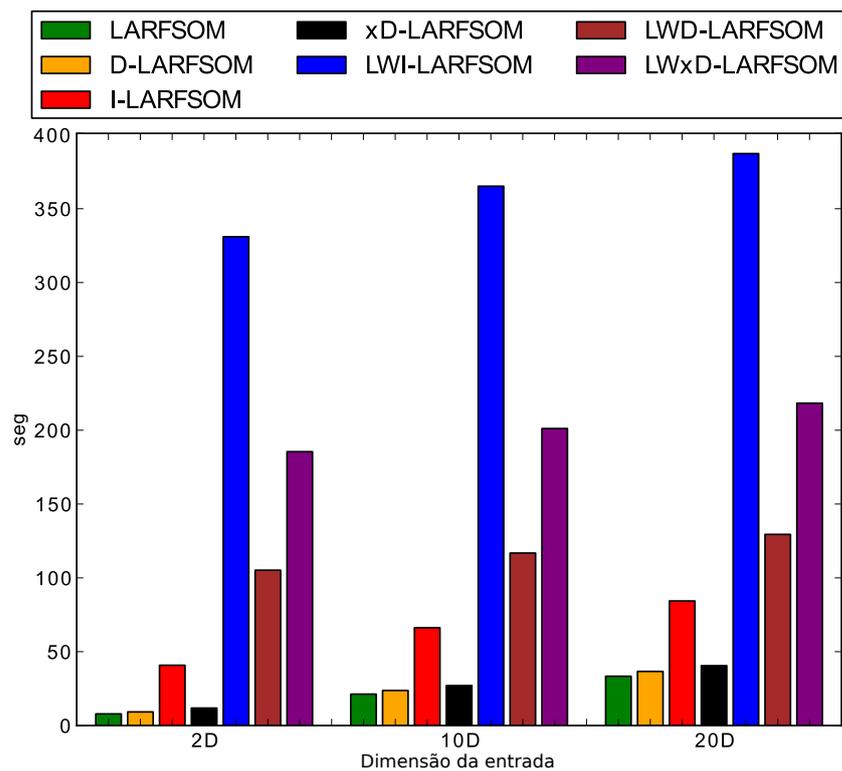
(b) Entrada 10D

Figura 13 – Curva de aprendizagem média na aproximação de função com saída estática com vetor de entrada 2D (a) e 10D (b).

Como mostrado nas Figuras 13a, 13b e 14a, o LARFSOM com saída discreta naturalmente tem uma aproximação pior que qualquer LARFSOM com algum método de mapeamento contínuo. O LWxD-LARFSOM e LWD-LARFSOM aproximaram com mais acurácia que os outros métodos. Em um primeiro momento, o LWI-LARFSOM é o mais acurado, mas, depois de cerca de  $\sim 20k$  passo de treinamento, o LWxD-LARFSOM e LWD-LARFSOM são os mais acurados. Também cabe ressaltar que o nMSE final do LWxD-LARFSOM e LWD-LARFSOM são menores do que o nMSE final relatado em (FLENTGE,



(a) Entrada 20D



(b) Tempo de execução

Figura 14 – Curva de aprendizagem média na aproximação de função com saída estática com vetor de entrada 20D (a) e tempo médio de execução na aproximação de função com saída estática em todos os experimentos (b). Os resultados apresentados nas Figuras 13a, 13b e 14a sugerem que os métodos não são sensíveis a dados redundantes e irrelevantes. Detalhes estão descritos na Seção 6.2.1.

2006) usando LWINGG (aproximadamente 0,006 e 0,007 dos LWxD-LARFSOM e LWD-

Tabela 5 – Média do nMSE dos métodos após 100k passos de treinamento.

Método	2-D 100k	10-D 100k	20-D 100k
LARFSOM	0,125 ± 0,01	0,14 ± 0,013	0,146 ± 0,015
I-LARFSOM	0,058 ± 0,008	0,071 ± 0,01	0,075 ± 0,011
D-LARFSOM	0,016 ± 0,003	0,021 ± 0,005	0,021 ± 0,004
xD-LARFSOM	0,016 ± 0,003	0,021 ± 0,004	0,022 ± 0,005
LWI-LARFSOM	0,013 ± 0,007	0,016 ± 0,004	0,018 ± 0,005
LWD-LARFSOM	<b>0,007 ± 0,001</b>	<b>0,007 ± 0,002</b>	<b>0,008 ± 0,002</b>
LWxD-LARFSOM	<b>0,006 ± 0,001</b>	<b>0,006 ± 0,001</b>	<b>0,007 ± 0,002</b>

LARFSOM, respectivamente, contra 0,009 do LWINGG). É importante frisar que os métodos LWD e LWxD, assim como o LWI, conseguiram recuperar a função bidimensional original quando confrontados com dados redundantes e irrelevantes (*Exp1.2* e *Exp1.3*).

A Figura 14b mostra o tempo médio das 100 execuções dos 100k passos de treinamento mas 100x41x41 calculos da saída aproximada. O LWxD-LARFSOM é quase duas vezes mais rápido do que LWI-LARFSOM e o LWD-LARFSOM é quase três vezes mais rápido do que LWI-LARFSOM. A principal razão para isso pode ser atribuída ao custo computacional menor dos métodos da diferença e da diferença estendido em comparação ao método da interpolação linear por partes.

### 6.2.2 Resultados Comparativos em Aproximação de Função com Saída Variante no Tempo

Como dito anteriormente, para os experimentos *Exp2.1* e *Exp2.2*, o número máximo de vitórias foi ajustado para  $v_m = 2000$  visto que o mapa precisa acompanhar as mudanças na função a ser aproximada. As Figuras 15, 16 e 17 mostram as curvas de aprendizagem média de 100 simulações dos experimentos *Exp2.1*, *Exp2.2* e *Exp3.1* ao longo de 100k passos.

Como mostrado na Figura 15, o LWxD-LARFSOM e LWD-LARFSOM podem aproximar com mais acurácia do que os outros métodos. Em um primeiro momento, o LWI-LARFSOM é o mais acurado, mas o LWxD-LARFSOM passa a ser o mais acurado após aproximadamente 14k passos de treinamento. D-LARFSOM e xD-LARFSOM aproximam melhor do que I-LARFSOM após cerca de 18k passos.

Os resultados do *Exp2.2* estão apresentados na Figura 16. Durante os primeiros 10k passos, o ruído é forte, e todos os métodos não conseguem aproximar com um nMSE menor do que 0,5. D-LARFSOM e LWD-LARFSOM parecem ser os menos sensíveis a ruído durante os primeiros 50k passos. Após 50k passos, os métodos LARFSOM e I-LARFSOM convergem, e os outros conseguem reduzir o nMSE. Ao fim do experimento, o LWxD-LARFSOM e LWD-LARFSOM mostram ser os mais acurados.

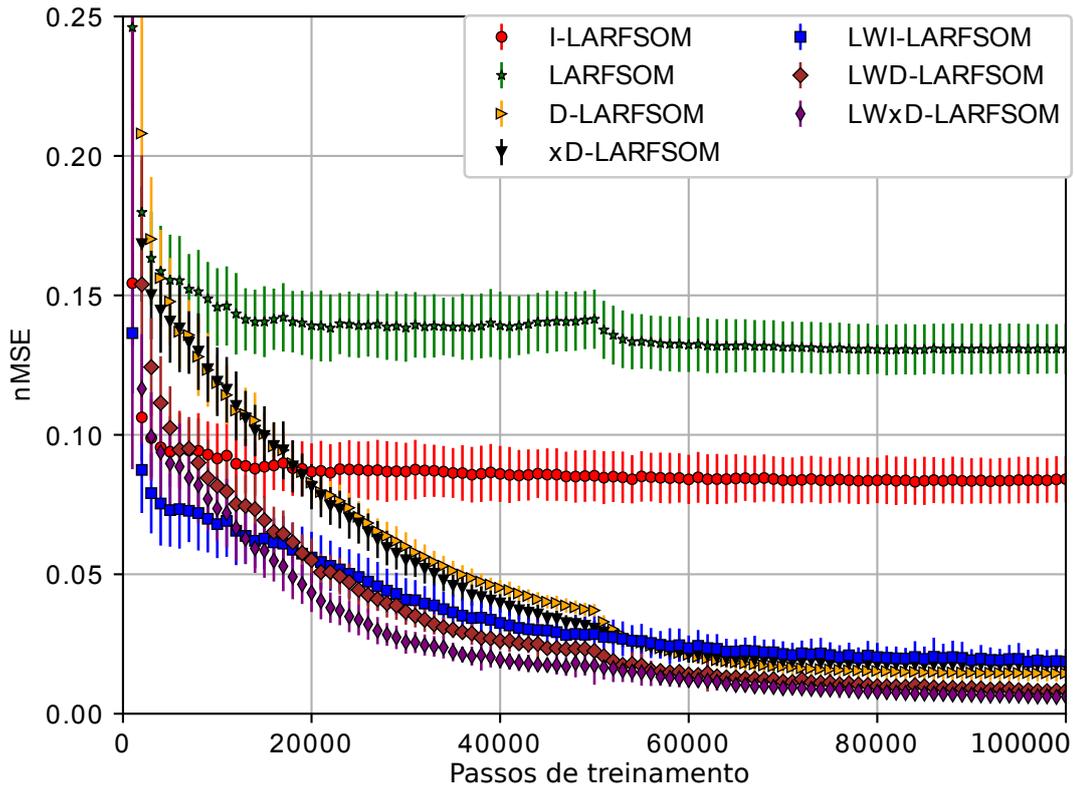
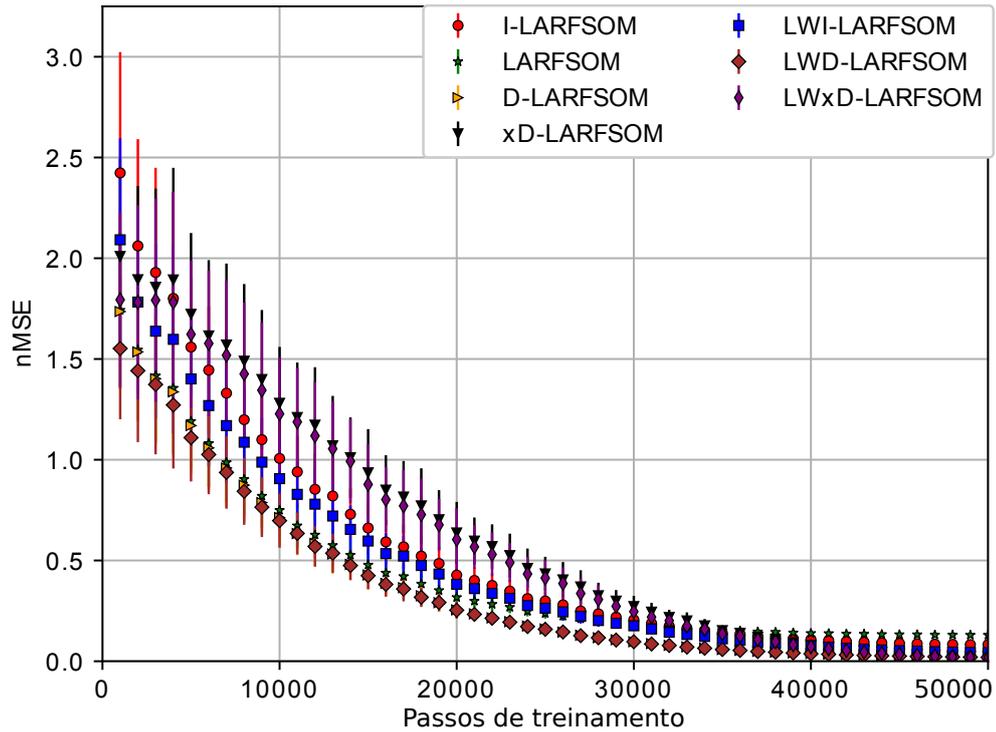


Figura 15 – Curva de aprendizagem média para a aproximação de função com saída variante no tempo no experimento *Exp2.1*. Esses resultados sugerem que os métodos podem lidar com funções variante no tempo. Mais detalhes na Seção 6.2.2.

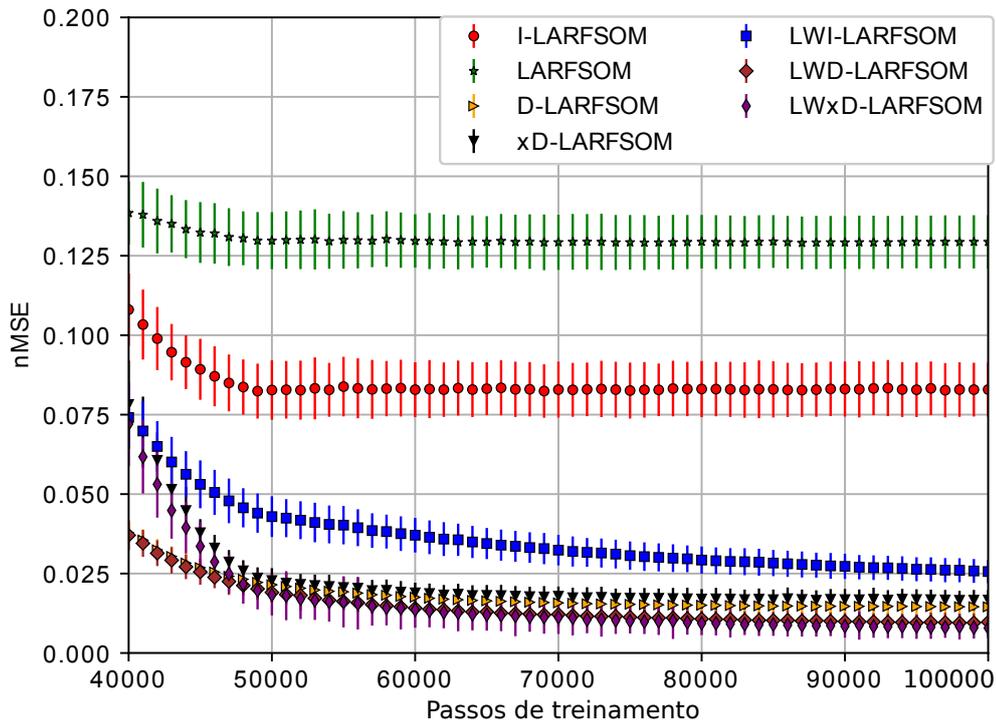
### 6.2.3 Resultados Comparativos em Aproximação de Função com Distribuição Variante no Tempo

Para o *Exp3.1*, o número máximo de nodos  $N_{max}$  foi 225. Na Figura 17a, pode-se observar que D-LARFSOM, xD-LARFSOM, LWI-LARFSOM, LWD-LARFSOM, e LWxD-LARFSOM reduzem continuamente o nMSE desde do início do treinamento até por volta de 35k passos de treinamento. Também, pode-se observar que todos os métodos sofrem uma perda significativa da acurácia logo em seguida ao instante de descontinuidade do comportamento variante no tempo da distribuição de entrada (Eq. 6.4).

Entre 35k e 50k passos de treinamento, a variância do erro de todos os métodos aumenta (Fig. 17a). Para analisar esse fenômeno, foi plotado o número médio de nodos durante o treinamento para as repetições, para repetições com  $|c_1 - c_2| < 0,25$  e com  $|c_1 - c_2| > 0,75$  (Fig. 17b). Pode-se constatar, através da Figura 17b, que o número de nodos também aumenta entre 35k e 50k passos de treinamento para todas as repetições e para repetições com  $|c_1 - c_2| > 0,75$ . Porém, a média do número de nodos para as repetições com  $|c_1 - c_2| < 0,25$  não aumenta entre 35k e 50k passos de treinamento. Isso sugere que a variância do nMSE entre 35k e 50k passos pode estar relacionada com o valor aleatório da diferença entre  $c_1$  e  $c_2$ . Então, quando a diferença é grande, o LARFSOM decide adicionar novos nodos. Esses novos nodos não estão treinados o suficiente para

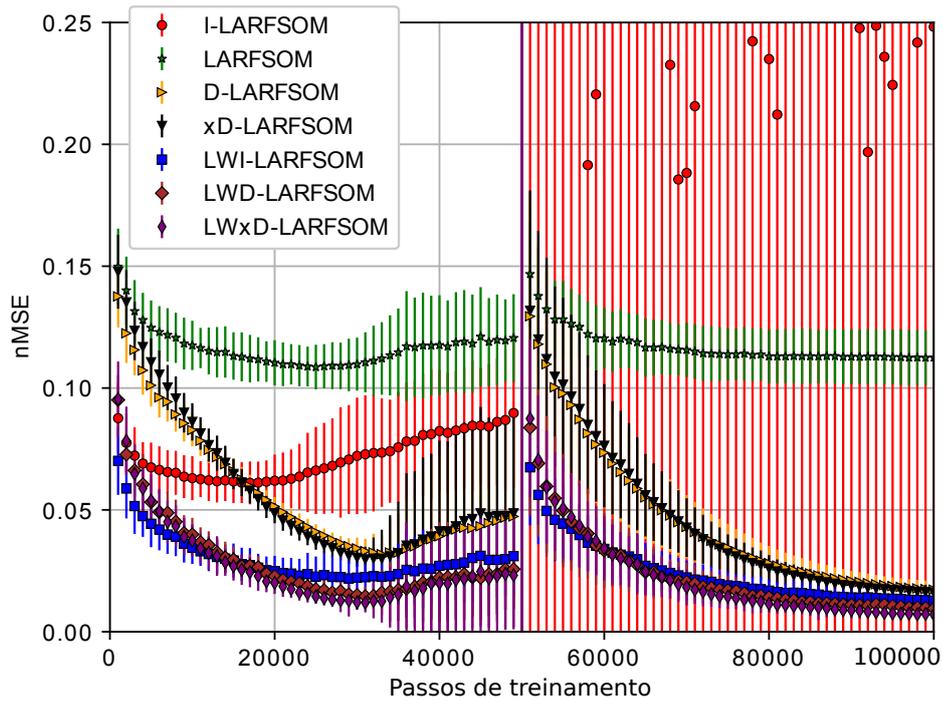


(a) Exp2.2 - primeiros 50k passos

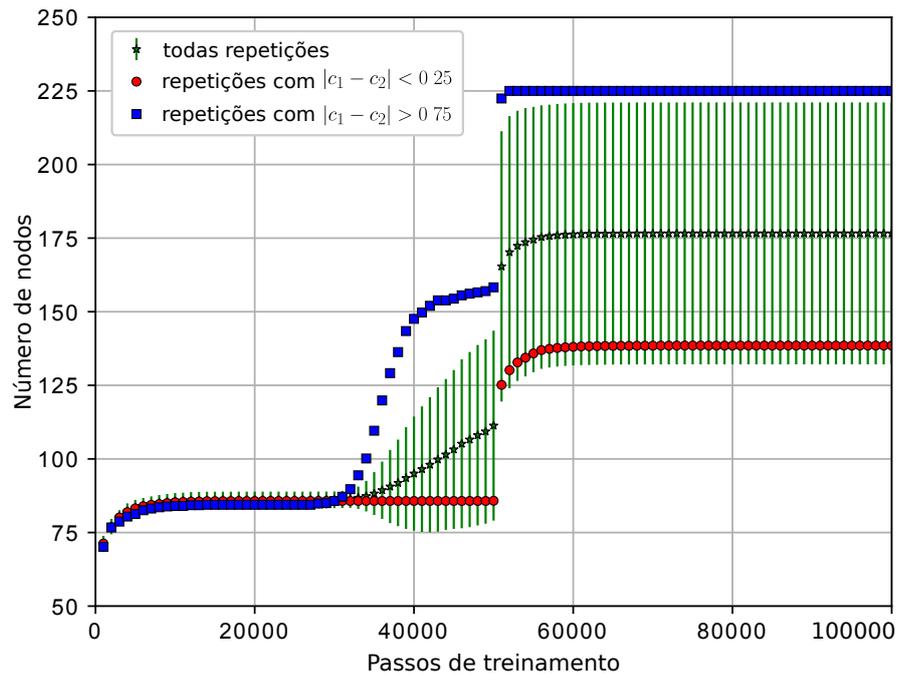


(b) Exp2.2 - de 40k até 100k passos

Figura 16 – Curva de aprendizagem média para a aproximação de função com saída variante no tempo no experimento *Exp2.2*. Resultados apresentados em b sugerem que os métodos podem lidar com funções com saída variante no tempo, e em a, mostram que os métodos foram sensíveis a fortes ruídos. Detalhes são descritos na Seção 6.2.2.



(a) Exp3.1 - Curva de aprendizagem



(b) Exp3.1 - Número de nodos

Figura 17 – Resultados no experimento da aproximação de função com distribuição de entrada variante no tempo (*Exp3.1*). Em (a), curva de aprendizagem média e em (b), média do número de nodos usando o LARFSOM para todas as 100 repetições, para repetições com  $|c_1 - c_2| < 0,25$  e para repetições com  $|c_1 - c_2| > 0,75$  (b). Resultados apresentados em (a) sugerem que os métodos podem lidar com funções variante no tempo. Mais detalhes nas Seções 6.2.2 e 6.2.3.

produzir saídas acuradas, logo, o nMSE é afetado negativamente. Contudo, se a diferença for pequena, poucos nodos são adicionados, portanto, mais nodos previamente treinados são utilizados no cálculo da saída aproximada.

No experimento *Exp3.1*, o LWI-LARFSOM aprende mais rapidamente do que o LWxD-LARFSOM e LWD-LARFSOM no começo do treinamento e logo após a descontinuidade. Porém, o LWxD-LARFSOM e LWD-LARFSOM produzem melhores aproximações do que LWI-LARFSOM ao longo das duas fase, antes e depois da descontinuidade. Vale ressaltar que o I-LARFSOM perde acurácia significativamente logo após a descontinuidade, e em seguida, não consegue reduzir o erro de aproximação consistentemente. Esse fato sugere que a interpolação linear é sensível a mudanças abruptas na distribuição de entrada.

#### 6.2.4 Resultados Comparativos em Benchmarks

Foi feita uma análise comparativa de três versões do LARFSOM para aproximação de função em problemas de *benchmarks*. Na análise, foram considerados a taxa de erro, o tempo de treinamento e o tempo de resposta (tempo gasto para calcular a saída aproximada). Selecionou-se as três técnicas com melhores resultados de aproximação para os dados sintéticos: LWI-LARFSOM, LWD-LARFSOM e LWxD-LARFSOM. O objetivo deste estudo comparativo é responder duas questões: (1) Os LARFSOMs estendidos para aproximação de função são competitivos em comparação a um algoritmo de aprendizagem supervisionada em termos de acurácia em problemas de *benchmarks*? (2) Os métodos da diferença e da diferença estendido com aprendizagem localmente ponderada são mais acurados e menos custosos computacionalmente do que a interpolação linear por partes localmente ponderada?

O algoritmo de aprendizagem supervisionada usado nesta tese foi a máquina de vetores de suporte para regressão (SVR). Essa técnica foi selecionada para comparação, pois é largamente aplicada em diversos problemas de regressão (AL-MUSAYLH et al., 2018; DÍAZ-VICO et al., 2017; LI; FANG; LIU, 2018; SALCEDO-SANZ et al., 2014; WANG; LI, 2018). O parâmetro ajustado para cada experimento usando o SVR foi o parâmetro de margem  $C$ . Esse parâmetro foi selecionado para cada base de dados e o valor específico está apresentado na Tabela 6. Essa tabela também apresenta os parâmetros de treinamento para os LARFSOMs com os métodos LWI, LWD e LWxD. Os demais parâmetros de treinamento foram  $a_T = 1,65$ ,  $\rho_f = 0,001$ ,  $\epsilon = 0,4$  e  $v_m = 100$ .

Para as bases de dados *Kinematic* e *Bank*, ao utilizar todos os dados de treinamento, o LARFSOM com LWI demandou muitos nodos e muitas iterações de treinamento. Essa condição inviabilizou os experimentos do LWI-LARFSOM nessas bases, visto que o LWI-LARFSOM necessitava de muito tempo de treinamento e muito tempo de resposta.<sup>4</sup> Tal constatação sugere que o LWI é sensível à presença de ruído já que essas duas bases

<sup>4</sup> O LWI-LARFSOM gasta 150 vezes e 5 vezes mais de tempo de treinamento e resposta, respectivamente, do que o LWxD-LARFSOM.

Tabela 6 – Valores para os parâmetros de treinamento usados nos problemas de *benchmarks*.

Base de dados	SVR	LWI-LARFSOM				LWD-LARFSOM					LWxD-LARFSOM				
	$C$	$N_{max}$	$t_{max}$	$\lambda_{init}$	$\alpha_\lambda$	$N_{max}$	$t_{max}$	$\alpha_k$	$\lambda_{init}$	$\alpha_\lambda$	$N_{max}$	$t_{max}$	$\alpha_k$	$\lambda_{init}$	$\alpha_\lambda$
Boston Housing	$2^3$	90	75	5	10	25	200	0,1	5	10	25	200	0,01	5	10
Kinematic	$2^7$	100	150	1	1	40	100	0,01	5	1	40	100	0,001	5	1
Auto MPG	$2^3$	60	100	5	100	20	50	0,1	5	100	20	50	0,01	5	100
Abalone	$2^7$	100	25	15	50	15	100	0,1	15	50	15	100	0,01	15	50
Price	$2^0$	15	200	5	1	5	100	0,01	1	0,1	5	100	0,001	1	1
Servo	$2^5$	30	200	5	5	10	100	0,1	5	50	10	150	0,01	5	1
Machine CPU	$2^1$	15	100	1	1	2	20	0,1	30	250	2	100	0,01	20	30
Bank	$2^4$	50	100	5	1	5	50	0,01	1	0,1	5	50	0,01	5	0,1

são consideradas ruidosas. Logo, para realizar uma análise comparativa com o LWI, foram realizados experimentos com essas bases de dados reduzidas para apenas um *fold* de treinamento (escolhido de maneira aleatória a cada treinamento de um lote). Esses experimentos estão referenciados com \* na Tabela 7.

Pode-se ver na Tabela 7 que o LWD e LWxD conseguiram atingir resultados significativamente melhores ou equivalente aos do SVR e do LWI. O LWD atingiu a menor taxa RMSE para 5 bases de dados e o LWxD, para 3 bases de dados (ambos foram o menor erro para a base de dados *servo*). Para as bases *Abalone*, *Price*, *Bank reduzida*, as taxas RMSE do LWD-LARFSOM são estatisticamente menores do que os outros métodos. O SVR atingiu o menor RMSE para 3 bases, *kinematic*, *kinematic reduzida* e *bank*. Respondendo a primeira questão deste estudo comparativo, os resultados sugerem que o LWD é equivalente ao SVR em termos de acurácia.

A Tabela 7 também mostra que o SVR é treinando mais rapidamente do que os LWD-LARFSOM e LWxD-LARFSOM para todas as bases de dados. Entretanto, em termos do tempo de resposta, o LWD e LWxD são mais rápidos do que o SVR em 5 bases (*Auto MPG*, *Abalone*, *Price*, *Servo* e *Machine CPU*). O custo computacional dos métodos baseado no LARFSOM aumenta principalmente em virtude do número de nodos no mapa e do número de iterações na fase de treinamento, e, na fase de operação, aumenta em virtude do número de nodos. Como mencionado na Seção 6.1.2, as implementações dos métodos baseado no LARFSOM não são otimizadas. O tempo de treinamento e de resposta poderiam ser otimizados utilizando métodos de acesso espacial na busca do nodo vencedor.

Quando os métodos LWD e LWxD são comparados ao LWI, os LWD e LWxD conseguem aproximações mais acuradas do que LWI para todas as base (Tabela 7). É importante salientar que LWD e LWxD conseguem esse feito com um menor número de nodos e iterações de treinamento (Tabela 6). Também, o LWI falhou para as base *kinematic* e *bank* quando utiliza todo conjunto de treinamento na fase de aprendizagem. Por sua vez, os LWD e LWxD foram soluções viáveis visto que não demandam aumento de memória como o LWI. Esse fato sugere que LWD e LWxD podem evitar uma limitação importante

Tabela 7 – Média do erro RMSE, média do tempo de treinamento, média do tempo de resposta para aproximação de funções de problemas de *benchmarks*.

Base de dados	Método	RMSE	Tempo de treinamento (seg)	Tempo de resposta (ms)
Boston Housing	SVR	0,0794 ± 0,0024 ✗	0,0178 ± 0,0111	0,742 ± 0,91
	LWI-LARFSOM	0,1063 ± 0,0056 ✓	13,2401 ± 0,2783	381,71 ± 20,136
	LWD-LARFSOM	0,0797 ± 0,0051 ✗	5,5338 ± 0,2017	69,109 ± 7,773
	LWxD-LARFSOM	<b>0,0789 ± 0,0054</b>	6,3134 ± 0,2376	94,967 ± 7,217
Kinematic	SVR	<b>0,0647 ± 0,0004</b>	19,9711 ± 0,4694	263,275 ± 2,578
	LWI-LARFSOM	-	-	-
	LWD-LARFSOM	0,0959 ± 0,0014 ✓	3223,6582 ± 57,0598	489,705 ± 11,073
	LWxD-LARFSOM	0,0925 ± 0,0014 ✓	4065,4198 ± 60,5011	1731,183 ± 23,679
Kinematic*	SVR	<b>0,0826 ± 0,0008</b>	1,096 ± 0,0387	200,082 ± 2,138
	LWI-LARFSOM	0,1236 ± 0,0091 ✓	6925,0085 ± 286,2433	9730,166 ± 729,793
	LWD-LARFSOM	0,0961 ± 0,0013 ✓	688,8619 ± 14,0455	461,764 ± 153,764
	LWxD-LARFSOM	0,1067 ± 0,0011 ✓	989,9604 ± 6,6839	1663,289 ± 52,641
Auto MPG	SVR	0,0745 ± 0,0015 ✗	0,0996 ± 0,0015	16,457 ± 0,337
	LWI-LARFSOM	0,0777 ± 0,0032 ✓	88,5572 ± 21,5555	60,531 ± 14,987
	LWD-LARFSOM	<b>0,0733 ± 0,0028</b>	3,7812 ± 0,1117	2,573 ± 0,813
	LWxD-LARFSOM	0,0774 ± 0,0026 ✓	4,9315 ± 0,1418	4,29 ± 0,9
Abalone	SVR	0,0780 ± 0,0003 ✓	1,0167 ± 0,013	179,940 ± 0,810
	LWI-LARFSOM	0,0807 ± 0,0006 ✓	561,6155 ± 48,6642	1129,723 ± 154,496
	LWD-LARFSOM	<b>0,0765 ± 0,0005</b>	60,1706 ± 3,075	26,213 ± 2,743
	LWxD-LARFSOM	0,0769 ± 0,0005 ✓	75,3861 ± 2,9586	34,742 ± 2,772
Price	SVR	0,0880 ± 0,0030 ✓	0,0291 ± 0,0009	9,075 ± 0,198
	LWI-LARFSOM	0,0956 ± 0,0068 ✓	81,047 ± 33,6651	42,174 ± 15,506
	LWD-LARFSOM	<b>0,0819 ± 0,0038</b>	1,8350 ± 0,0471	0,701 ± 0,461
	LWxD-LARFSOM	0,0847 ± 0,0033 ✓	2,3802 ± 0,1628	0,999 ± 0,424
Servo	SVR	0,1084 ± 0,0079 ✓	0,0669 ± 0,0023	9,414 ± 0,282
	LWI-LARFSOM	0,1039 ± 0,0134 ✓	94,2758 ± 37,8348	48,757 ± 16,571
	LWD-LARFSOM	<b>0,0893 ± 0,0096 ✗</b>	2,6632 ± 0,1019	0,910 ± 0,477
	LWxD-LARFSOM	<b>0,0893 ± 0,0075</b>	5,4766 ± 0,1469	2,146 ± 1,234
Machine CPU	SVR	0,0694 ± 0,0033 ✓	0,0226 ± 0,0008	6,799 ± 0,197
	LWI-LARFSOM	0,0583 ± 0,0133 ✓	18,8756 ± 2,3725	21,696 ± 4,788
	LWD-LARFSOM	0,0473 ± 0,0123 ✗	0,3068 ± 0,0232	0,646 ± 0,604
	LWxD-LARFSOM	<b>0,0464 ± 0,0114</b>	2,0317 ± 0,0229	0,747 ± 0,595
Bank	SVR	<b>0,0454 ± 0,0002</b>	0,5015 ± 0,0061	19,752 ± 0,294
	LWI-LARFSOM	-	-	-
	LWD-LARFSOM	0,048 ± 0,0002 ✓	42,1714 ± 0,2236	33,594 ± 1,231
	LWxD-LARFSOM	0,0489 ± 0,0005 ✓	54,5558 ± 0,176	51,109 ± 1,475
Bank*	SVR	0,0526 ± 0,0008 ✓	0,084 ± 0,0153	9,772 ± 1,914
	LWI-LARFSOM	0,1012 ± 0,0047 ✓	1070,2283 ± 95,1346	30584,379 ± 3242,278
	LWD-LARFSOM	<b>0,0488 ± 0,0004</b>	10,4734 ± 0,1463	34,208 ± 1,657
	LWxD-LARFSOM	0,0493 ± 0,0010 ✓	13,7693 ± 0,21	50,613 ± 1,493

Os experimentos com o conjunto de treinamento reduzido para apenas um *fold* estão indicados com \*. Os resultados dos testes de hipótese estão denotados pelos símbolos ✗ (não rejeita) e ✓ (rejeita com  $(p < 0, 05)$ )

do LWI.

Para todas as bases de dados, os LWD e LWxD demandaram menos nodos do que o LWI, e para cinco bases (*Kinematic*, *Auto MPG*, *Price*, *Servo* e *Bank*), os LWD e LWxD necessitaram de menos iterações de treinamento do que o LWI (Tabela 7). Isso pode significar que os métodos LWD e LWxD precisem de menos nodos e menos iterações de

Tabela 8 – Vantagens do LWD-LARFSOM e LWxD-LARFSOM em comparação ao SVR e LWI-LARFSOM.

Problema	SVR	LWI-LARFSOM	LWD-LARFSOM	LWxD-LARFSOM
aproximação de função estática <sup>a</sup> (Exp 1.1, Exp 1.2, Exp 1.3)	não ✗	sim	sim	sim
aprox. de função com saída variante no tempo <sup>a</sup> (Exp 2.1, Exp 2.2)	não ✗	sim	sim	sim
distribuição de entrada não estática <sup>a</sup> (Exp 3)	não ✗	sim	sim	sim
eficiência no treinamento (problemas de <i>benchmarks</i> )	rápida	lenta ✗	moderada	moderada
eficiência na resposta (problemas de <i>benchmarks</i> )	rápida	lenta ✗	rápido	rápido
desempenho acurado (problemas de <i>benchmarks</i> )	sim	não ✗	sim	sim

<sup>a</sup> De acordo com Flentge (FLENTGE, 2006), os resultados do SVR em um cenário de aprendizagem online não foram bons.

treinamento para aprender aproximações acuradas do que LWI precisaria. Então, isso pode ser interpretado como uma outra forma de otimizar o custo computacional além da redução através de procedimentos menos custosos. Logo, respondendo a segunda questão, os experimentos sobre problemas de *benchmarks* sugerem que os LWD e LWxD são mais rápidos do que o LWI tanto no tempo de treinamento quanto no tempo de resposta. A Tabela 8 sumariza vantagens dos LWD-LARFSOM e LWxD-LARFSOM.

### 6.3 CONCLUSÃO

As simulações apresentas neste capítulo sugerem que os LARFSOMs para mapeamento contínuo têm aspectos importantes para aplicações de controle em um cenário de mudanças no sistema. Para esse cenário, o modelo do sistema ou do controlador podem ser vistos como uma função variante no tempo. Então, os LARFSOMs foram avaliados na aproximação de uma função que podia ter a saída variante no tempo ou estar sobre uma distribuição de entrada variante no tempo. Um importante aspecto para esse cenário é a demanda de uma aprendizagem incremental. Há três mecanismos que suportam essas aprendizagem de maneira apropriada com a definição feita na Seção 4.2: (1) a estrutura variante no tempo e (2) a estratégia de regularização da taxa de aprendizagem do LARFSOM, e (3) a modelagem do mapeamento através de modelos locais. Os resultados sobre o cenário de mudanças ao longo do treinamento mostraram que o LARFSOM com técnicas de mapeamento contínuo tem a capacidade de aprender incrementalmente mesmo na presença de dados redundantes, irrelevantes e ruidosos.

A acurácia das aproximações e o custo computacional também foram avaliadas. Os experimentos sobre dados de problemas de *benchmarks* serviram de suporte para uma análise comparativa do LARFSOM para mapeamento contínuo. A máquina de vetores de suporte para regressão (SVR) foi selecionada para gerar resultados comparativos. Os LARFSOM para mapeamento contínuos com aprendizagem localmente ponderada, LWD-LARFSOM e LWxD-LARFSOM, são algoritmos de aprendizagem com capacidade de generalização e acurácia equivalente ao SVR. Porém, o custo computacional ainda é elevado em comparação ao do SVR. Mas, vale ressaltar que o custo computacional foi reduzido em comparação ao LWI-LARFSOM. Além da redução no custo, este trabalho satisfaz um

dos pontos levantados com trabalhos futuros em (FLENTGE, 2006). Os LWD-LARFSOM e LWxD-LARFSOM tem menos parâmetros para ajustar do LWINGG (9 parâmetros contra 14).

## 7 EXPERIMENTOS NO PÊNDBULO INVERTIDO

Esse capítulo apresenta e discute alguns experimentos para verificar a capacidade do *controle por realimentação de estados baseado no LARFSOM*. O primeiro sistema escolhido para fornecer uma visão desse controle foi o pêndulo invertido. Segundo Boubaker (2013), o pêndulo invertido é um dos problemas mais fundamentais da teoria de controle não-linear e nesse últimos 50 anos, tem sido o sistema mais usado para analisar e comparar os resultados de projetos de controle para problemas não-lineares. Neste experimento, a massa e o comprimento do pêndulo vão variar ao longo da execução provocando assim um sistema não-linear variante no tempo.

### 7.1 PÊNDBULO INVERTIDO COM CARRINHO

Considere um pêndulo invertido sobre um carrinho com a seguinte modelagem dinâmica (MINATOHARA; FURUKAWA, 2011):

$$\begin{aligned} (M + m)\ddot{x} + ml\ddot{\theta}\cos(\theta) - ml\dot{\theta}^2\sin(\theta) + f\dot{x} &= u \\ ml\ddot{x}\cos(\theta) + (I + ml^2)\ddot{\theta} - mgl\sin(\theta) + C\dot{\theta} &= 0 \end{aligned} \quad (7.1)$$

onde  $x$  é posição do carrinho [*metros*],  $\theta$  é a posição angular do pêndulo [*rad*],  $g = 9.8$  é aceleração da gravidade em [ $m/s^2$ ],  $M = 5.0kg$  é a massa do carrinho,  $m$  é a massa do pêndulo em [ $kg$ ] e  $l$  é o comprimento até o centro de massa do pêndulo [*metros*]. O estado do sistema é representado por  $\mathbf{x} = [x \ \theta \ \dot{x} \ \dot{\theta}]^T$ . A entrada de controle é denotada por  $u$ . O coeficiente de atrito do carrinho é  $f = 10.0kg/s$ , e coeficiente de atrito do pêndulo é  $C = 0.0004kgm^2/s$ . O momento de inercia é  $I = ml^2/3[kgm^2]$ . A massa do pêndulo  $m$  e o comprimento do pêndulo  $l$  serão os parâmetros variantes no tempo durante a execução do controle. Mais detalhes sobre a modelagem do pêndulo invertido e da linearização desse sistema estão apresentados no Apêndice D.

### 7.2 DESCRIÇÃO DO EXPERIMENTO

Nesta seção, são explicados os detalhes da execução do experimento do pêndulo invertido. Nesse experimento, foram usadas 9 configurações da planta para treinamento e 9 para teste. Duas das configuração foram usadas tanto para treinar quanto para testar. Os valores dos parâmetros  $m_i$  e  $l_i$  de treinamento e teste foram os mesmo usados em (MINATOHARA; FURUKAWA, 2011). As configurações do sistema usadas no treinamento foram as combinações de três valores de massas do pêndulo  $m \in \{0,2; 1; 1,8\}$  com três valores de comprimento  $l \in \{0,6; 1,2; 1,8\}$  e são representadas por  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_9$ . Os valores das configurações de teste estão apresentados na Tabela 9. Na Figura 18, são mostradas as

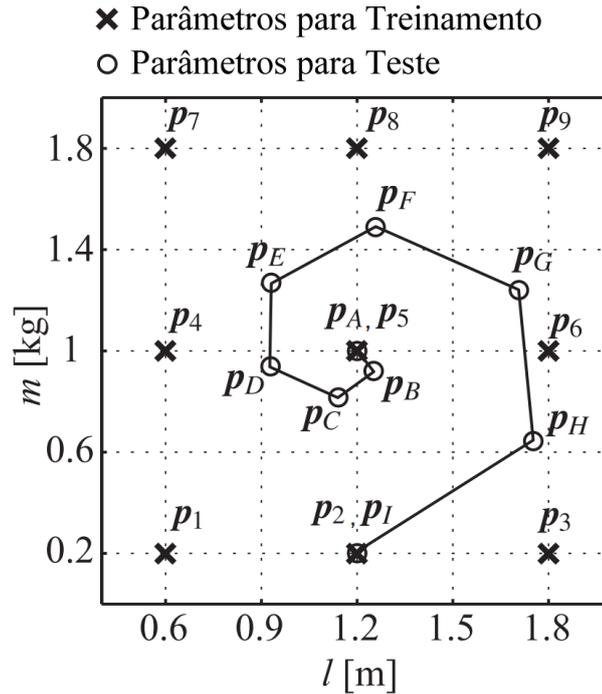


Figura 18 – Parâmetros da planta usados para treinamento e teste. Figura adaptada de (MINATOHARA; FURUKAWA, 2011).

configurações paramétricas do sistema usadas para o treinamento e teste neste experimento.

Na execução do teste, é feita uma mudança da posição desejada do carrinho  $x$ . Os possíveis valores desejados para  $x$  são  $\{1; -1; 0\}$  e a mudança ocorre a cada 10 segundos. O estado inicial é  $\mathbf{x}_0 = [0 \ 0 \ 0 \ 0]^T$  e o primeiro valor desejado para posição do carrinho é  $x_d = 1$ . Também ocorre a mudança dos valores das configurações da planta de teste a cada 10 segundos. A ordem das configurações de teste foi  $[\mathbf{p}_A, \mathbf{p}_B, \mathbf{p}_C, \mathbf{p}_D, \mathbf{p}_E, \mathbf{p}_F, \mathbf{p}_G, \mathbf{p}_H, \mathbf{p}_I]$ . As simulações foram executadas usando o método de Runge-Kutta de 4ª ordem com tamanho de passo  $h = 0,01$  [seg]. A Figura 19 mostra o sinal de referência para posição desejada do carrinho e as configurações do sistema usados durante a fase de teste.

Para construir os dados de treinamento, o sistema foi controlado usando o LQR projetado sabendo da configuração paramétrica da planta. Os parâmetros usados em todos os experimentos deste capítulo envolvendo o LQR estão apresentados na Tabela 10. Para cada configuração de treinamento, o sistema foi simulado por 30 segundos. Dessa forma,

Tabela 9 – Valores dos parâmetros  $m$  e  $l$  usado para testar

	$\mathbf{p}_A$	$\mathbf{p}_B$	$\mathbf{p}_C$	$\mathbf{p}_D$	$\mathbf{p}_E$	$\mathbf{p}_F$	$\mathbf{p}_G$	$\mathbf{p}_H$	$\mathbf{p}_I$
$m_i$	1	0,9	0,8	0,92	1,82	1,48	1,244	0,648	0,2
$l_i$	1,2	1,245	1,134	0,915	0,915	1,257	1,71	1,749	1,2

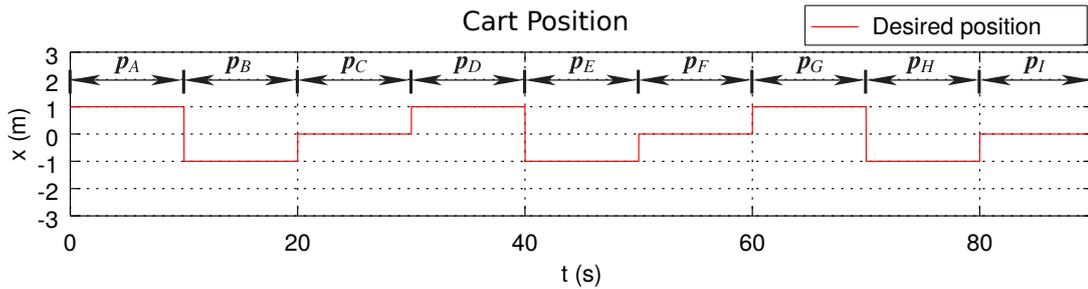


Figura 19 – Posição desejada do carrinho (em vermelho) e configurações do sistema durante a fase de teste.

Tabela 10 – Valor dos parâmetros para o controle LQR.

Parâmetro	Valor
$R$	1
$Q$	$diag\{500, 1, 1, 1\}$

cada valor desejado de posição do carrinho é simulado por 10 segundos. Para formar o conjunto de treinamento de fato, foram feitas tuplas com os dados de  $\{\text{vetor do estado do sistema } \mathbf{x}(t), \text{ sinal do erro } \mathbf{e}(t), \text{ saída do controle em malha aberta } \mathbf{v}(t), \text{ ganhos da malha de realimentação } \mathbf{K}(t)\}$  provenientes das simulações. Os dados de treinamento foram normalizados para que todas as variáveis ficassem com média igual a 0 e desvio padrão igual a 1. A partir desses dados, os vetores concatenados de entrada de tamanho  $p$  foram gerados.

## 7.3 RESULTADOS DOS EXPERIMENTOS

Nessa seção, são descritos os detalhes de cada modelo avaliado e são apresentados os resultados obtidos nas simulações do experimento do pêndulo invertido.

### 7.3.1 Resultados LQR

Nessa subseção, são mostrados os resultados obtidos no experimento do pêndulo invertido usando o regulador linear quadrático (LQR). Em um primeiro momento, é mostrado o resultado com LQR sendo projetado usando a informação da configuração do sistema. Dessa maneira, é gerado um LQR específico para cada configuração da planta. Na Figura 20, é apresentado o resultado dessa simulação. Este experimento é feito para mostrar um resultado comparativo para os sistema de controle que não utiliza a informação dos parâmetros da planta.

Também, foi feito um outro experimento usando um único LQR. Dessa vez, foi escolhida uma configuração de planta de treinamento e foi gerado um controle LQR específico para essa configuração. Foi selecionada a configuração de treinamento  $\mathbf{p}_4$ . Esse controle

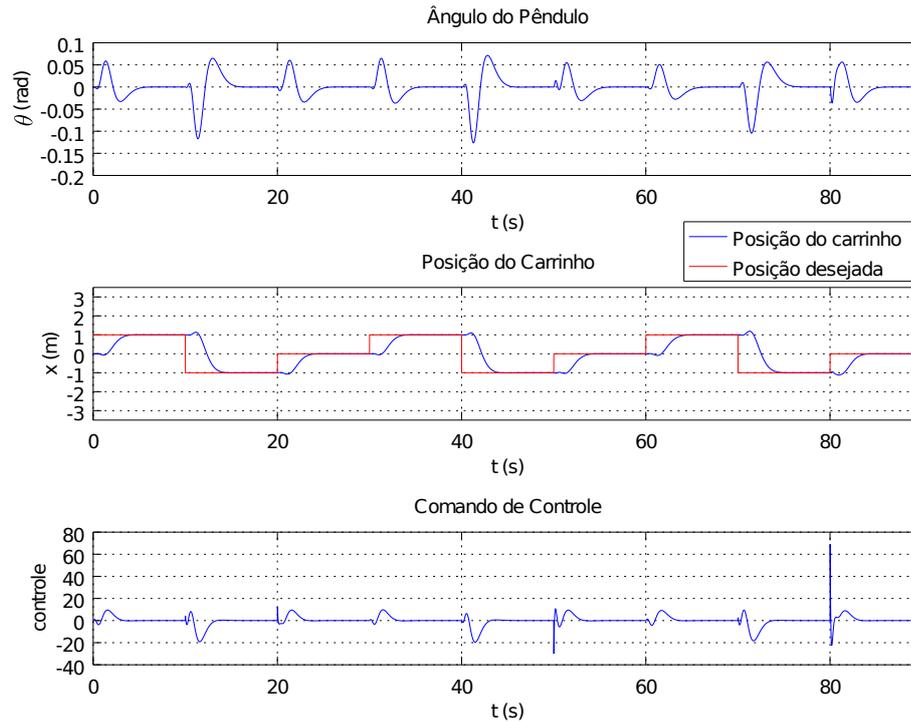


Figura 20 – Resultado da simulação do LQR para um caso de conhecimento total das configurações da planta.

foi usado para todas as configurações de teste. Para melhor entendimento do texto, vai-se denominar este controle de  $LQR_{p_4}$ . Na Figura 21, é apresentado o resultado deste experimento.

Como seria esperado, o controle  $LQR_{p_4}$  deixou de funcionar adequadamente, sem responder ou com instabilidade. Então, foram feitos outros experimentos para tentar identificar quais configurações de teste estavam apresentando um comportamento instável. O resultado foi que para as configurações de teste  $p_B$ ,  $p_F$ ,  $p_G$  e  $p_H$ , o  $LQR_{p_4}$  apresentou um comportamento instável. O experimento sem os casos de teste que o  $LQR_{p_4}$  falha está apresentado na Figura 22.

### 7.3.2 Resultados do Controlador por Realimentação de Estado Baseado no LARFSOM

Nessa subseção, são mostrados os resultados obtidos no experimento do Pêndulo Invertido usando LARFSOMs. Como pode ser observado no diagrama de blocos apresentado na Figura 9, há dois modelos de LARFSOM aplicados para controlar o sistema,  $LARFSOM^v$  e  $LARFSOM^k$ . O  $LARFSOM^v$  é responsável por estimar os comandos do controle em malha aberta  $v(t)$  e  $LARFSOM^k$  infere os ganhos da malha de realimentação  $\mathbf{K}$ . Então, foi realizado o treinamento de 10 pares de LARFSOM. As redes foram treinadas com os parâmetros apresentados na Tabela 11. Para este experimento, os seguintes índices de desempenho foram considerados: a integral do erro absoluto (IAE),  $J(\text{IAE}) = \int_0^t |\mathbf{e}(\tau)| d\tau$ , a integral do erro absoluto ponderado no tempo (ITAE),  $J(\text{ITAE}) = \int_0^t \tau |\mathbf{e}(\tau)| d\tau$ , a

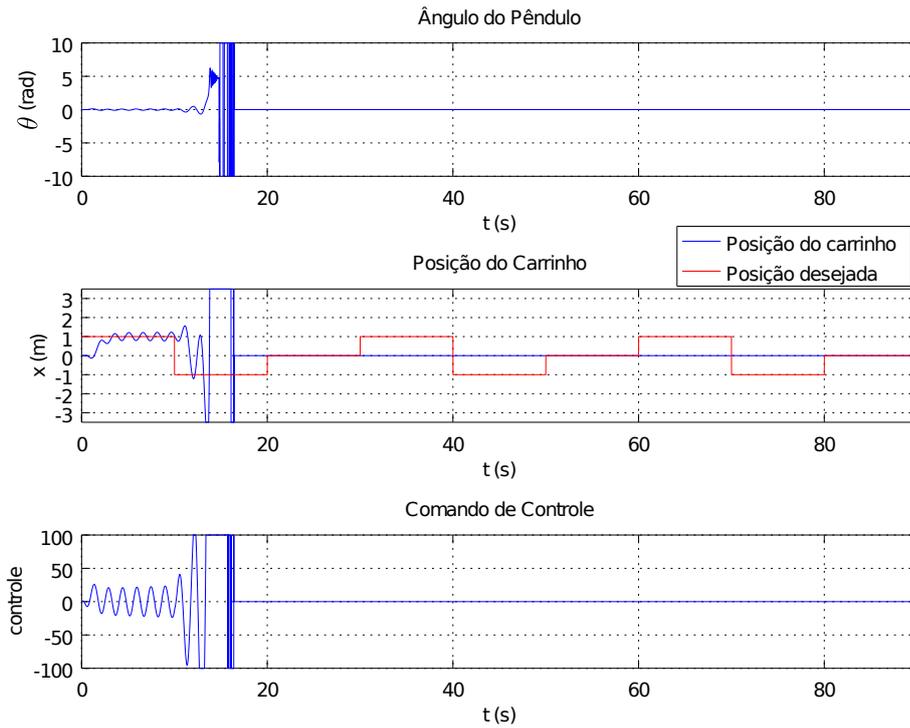


Figura 21 – Resultado da simulação do  $LQR_{p_4}$ . No início, há uma resposta não-amortecida. Porém, o sistema fica instável a partir da configuração de teste  $p_B$ .

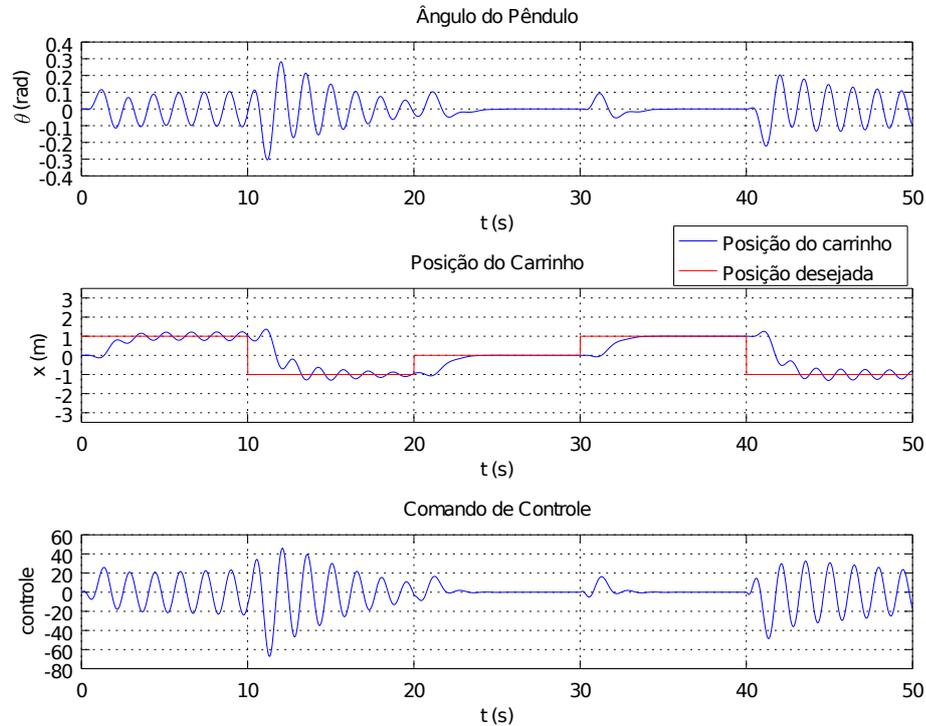


Figura 22 – Resultado da simulação do  $LQR_{p_4}$  usando apenas as configurações  $p_A$ ,  $p_C$ ,  $p_D$ ,  $p_E$  e  $p_I$  do conjunto de teste.

energia do sinal de controle  $E_c = \int_0^t u^2(\tau) d\tau$ , e o valor máximo do sinal de controle  $u_{max}$ .

Para avaliar o resultado do sistema de controle baseado nos LARFSOMs, foram com-

Tabela 11 – Valores dos parâmetros usados no treinamento dos LARFSOMs

Parâmetro	Valor					
	LARFSOM <sup>v</sup>	LARFSOM <sup>k</sup>	D-LARFSOM <sup>v</sup>	D-LARFSOM <sup>k</sup>	xD-LARFSOM <sup>v</sup>	xD-LARFSOM <sup>k</sup>
$\rho_f$	0,001	0,001	0,001	0,001	0,001	0,001
$\varepsilon$	0,4	0,4	0,4	0,4	0,4	0,4
$d_m$	100	100	100	100	100	100
$p$	9	9	9	9	9	9
$t_{max}$	5	5	5	5	5	5
$N_{max}$	3000	3000	600	1000	600	1000
$\alpha_T$	0,1	0,04	0,1	0,04	0,1	0,04
$\alpha_k$	-	-	0,001	0,001	0,001	0,001

Tabela 12 – Resultado dos índices de desempenho na simulação do pêndulo invertido.

Índice	LQR	LARFSOM	D-LARFSOM	xD-LARFSOM
$E_c$	1777	<b>1423,26</b>	2175,48	2111,31
$J(\text{IAE})$	<b>29,82</b>	31,62	30,61	30,36
$J(\text{ITAE})$	<b>1252,52</b>	1352,01	1298,62	1282,95
$u_{max}$	69,12	<b>23,84</b>	44,79	43,67

binados os  $LARFSOM^v$  e  $LARFSOM^k$  de mesmo índice de treinamento. Sendo assim, há 10 configurações de experimento. Nesta seção, todos os resultados apresentados foram obtidos pela configuração que obteve menor erro de rastreamento  $J(\text{IAE})$  para a simulação usando o conjunto de treinamento.

Na Figura 23, é apresentado o resultado do LARFSOM na simulação com o conjunto de parâmetros do sistema usados para treinamento. Na Figura 23b, são apresentados o sinal estimado do controle em malha aberta e o sinal da malha de realimentação aplicados no sistema durante a mesma simulação. Para um efeito comparativo, são mostrados também os comandos de controle em malha aberta e o sinal da malha de realimentação gerados por um controle LQR com o conhecimento dos parâmetros do sistema.

Na Figura 23a, pode-se notar que o sistema de controle LARFSOM controlou adequadamente, isto é, sem gerar instabilidade, para todos os parâmetros de treinamento. O único detalhe foi o comportamento transitório em torno do instante de tempo 73 segundos. Nesse caso, a resposta transitória teve um elemento oscilatório significativo. Isto também pode ser observado na Figura 23b.

Os  $LARFSOM^v$  e  $LARFSOM^k$  selecionados também foram simulados para os parâmetros de teste. Na Figura 24 e 25, são apresentados os resultados desta execução de teste. Os resultados dos índices de desempenho estão apresentados na Tabela 12. O resultados sugerem que o controle proposto baseado em três versões do LARFSOM tem potencial para controlar sistemas desconhecidos variante no tempo.

Foram realizadas simulações de 90 segundos usando unicamente cada parâmetro do conjunto de treinamento e de teste. Este experimento foi feito para verificar se o sistema

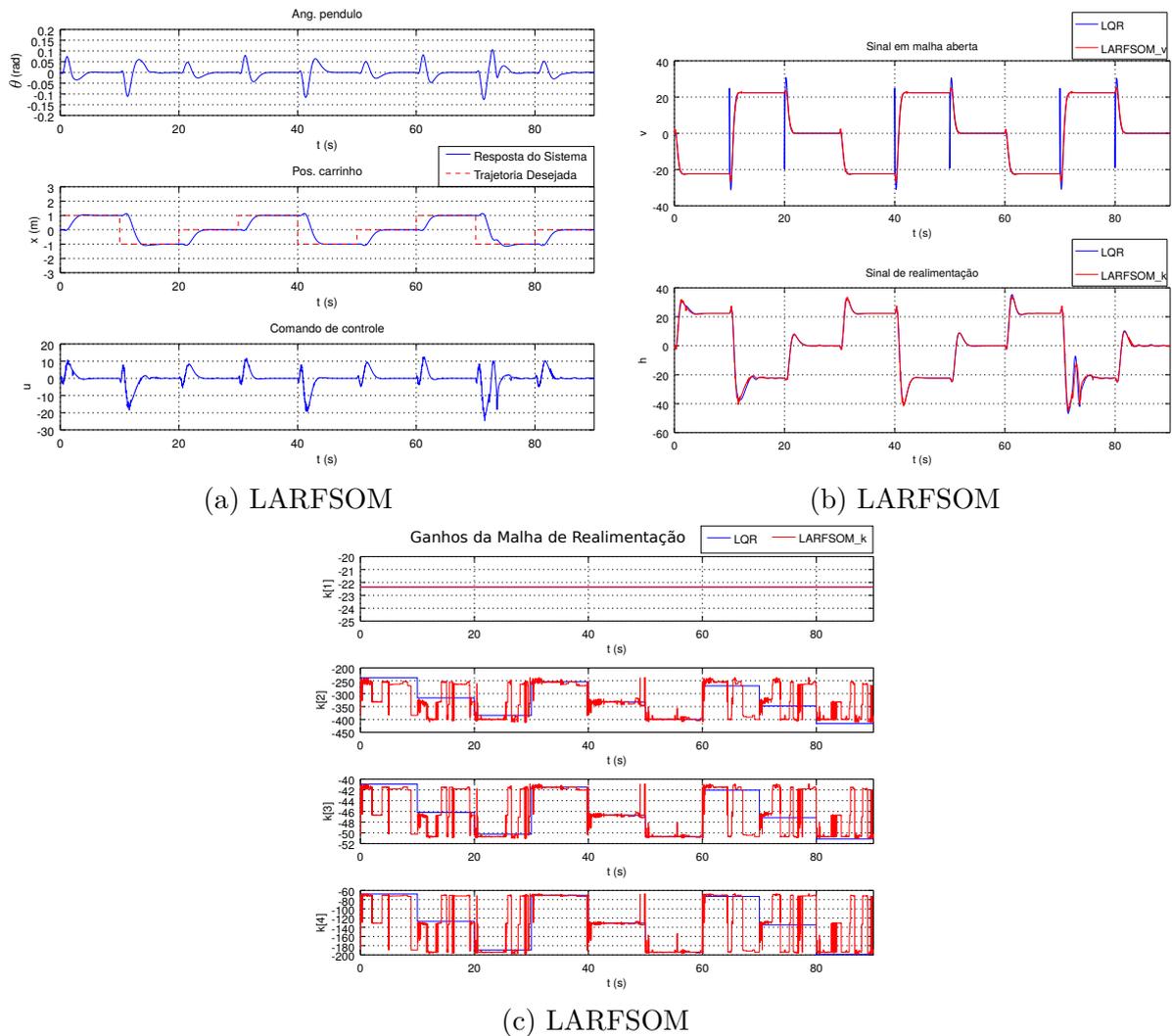
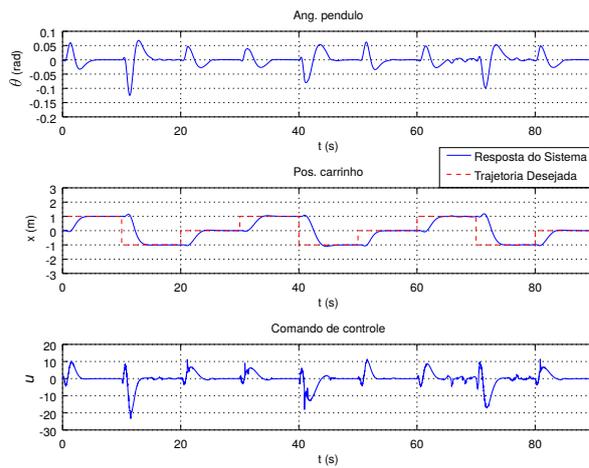
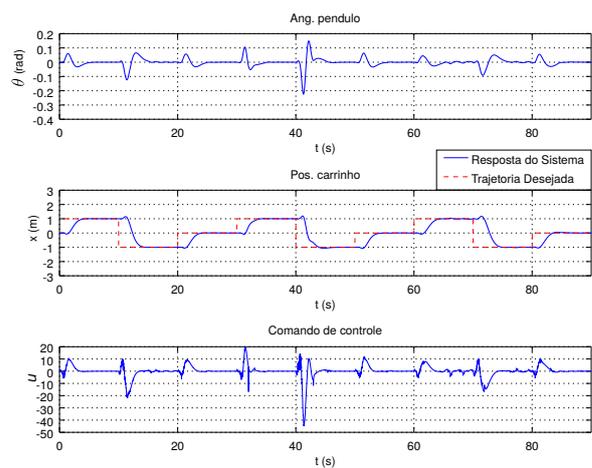


Figura 23 – Resultado do LARFSOM para simulação de controle do sistema usando as configurações de treinamento. (a) mostra o ângulo do pêndulo  $\theta$ , a posição do carrinho  $x$  e o sinal de controle  $u$ . (b) mostra o valor estimado para o sinal de controle em malha aberta  $v$  e o sinal de realimentação  $\mathbf{K} \times \mathbf{x}$ . (c) mostra os ganhos estimados.

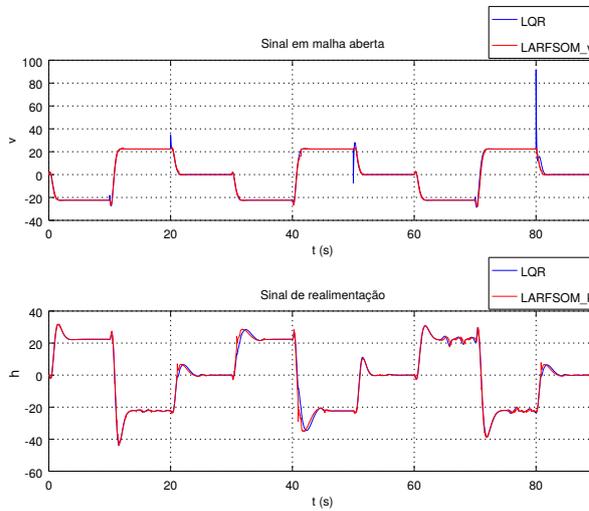
era capaz de controlar adequadamente cada parâmetro isoladamente e sem variar os parâmetros do sistema. Os controles LARFSOM controlaram adequadamente, isto é, sem um comportamento instável, todas as configurações de sistema simuladas.



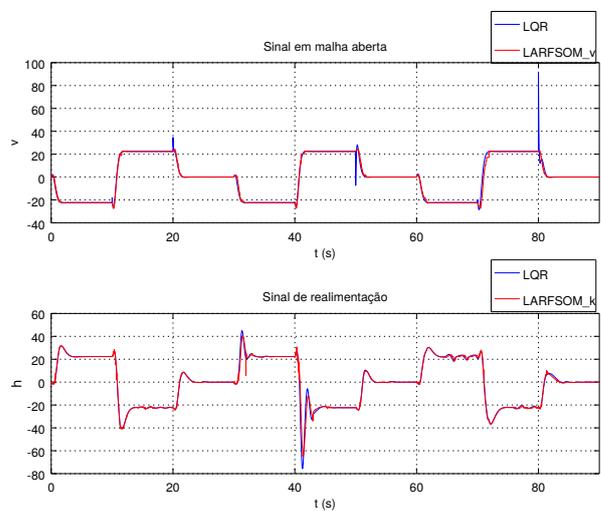
(a) LARFSOM



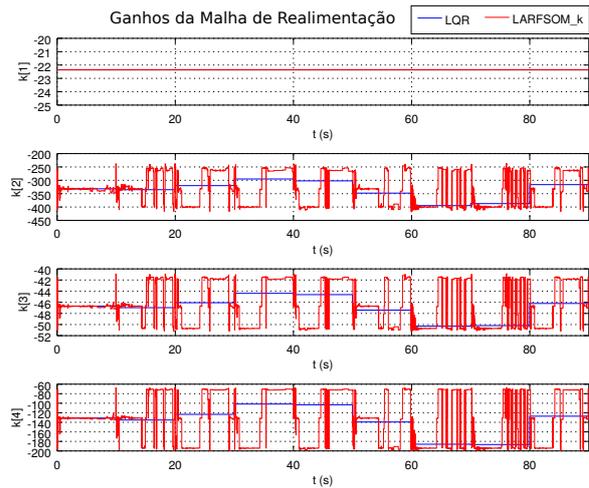
(b) D-LARFSOM



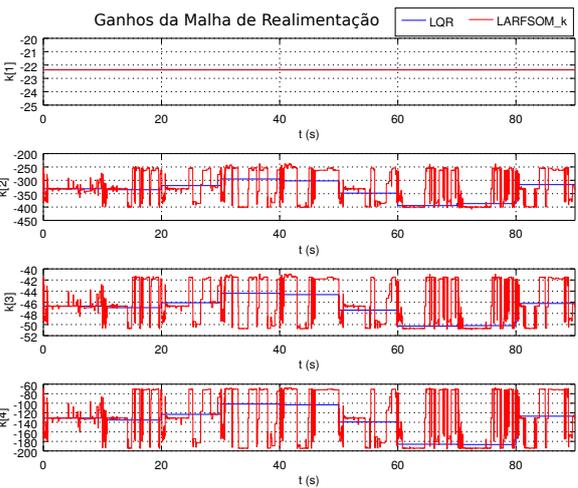
(c) LARFSOM



(d) D-LARFSOM

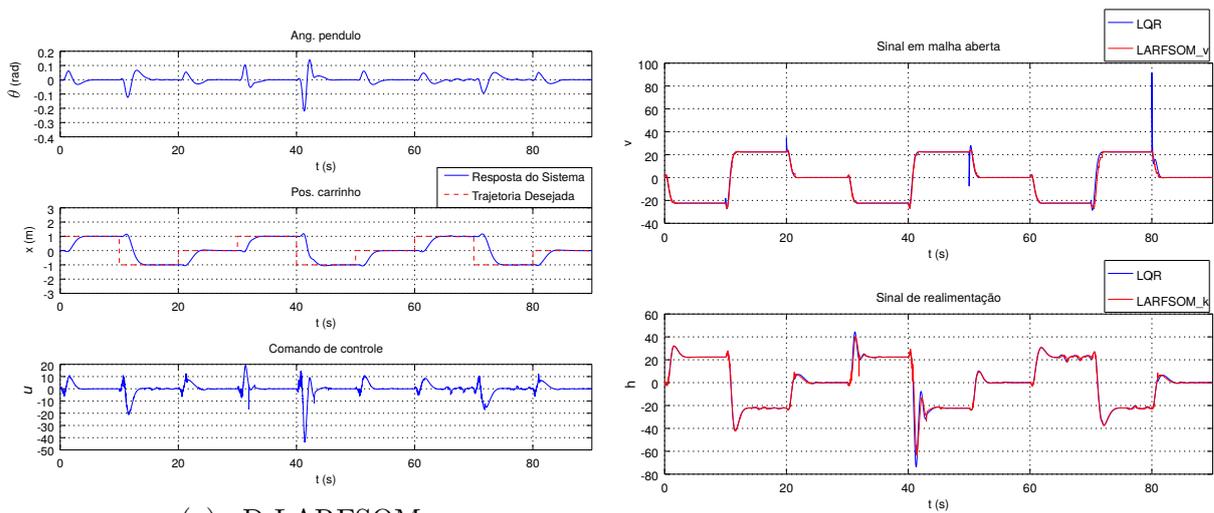


(e) LARFSOM



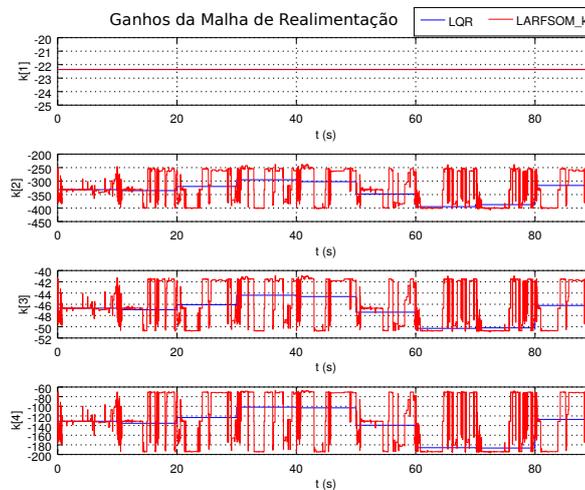
(f) D-LARFSOM

Figura 24 – Simulações de teste do pêndulo invertido usando o LARFSOM e o D-LARFSOM. (a) e (b) mostram o ângulo do pêndulo  $\theta$ , a posição do carrinho  $x$  e o sinal de controle  $u$ . (c) e (d) mostram o valor estimado para o sinal de controle em malha aberta  $v$  e o sinal de realimentação  $\mathbf{K} \times \mathbf{x}$ . (e) e (f) mostram os ganhos estimados.



(a) xD-LARFSOM

(b) xD-LARFSOM



(c) xD-LARFSOM

Figura 25 – Simulações de teste do pêndulo invertido usando o xD-LARFSOM. (a) mostra o ângulo do pêndulo  $\theta$ , a posição do carrinho  $x$  e o sinal de controle  $u$ . (b) mostra o valor estimado para o sinal de controle em malha aberta  $v$  e o sinal de realimentação  $\mathbf{K} \times \mathbf{x}$ . (c) mostra os ganhos estimados.

## 8 EXPERIMENTOS NO MANIPULADOR ROBÓTICO

A capacidade do *controle por realimentação de estados baseado no LARFSOM* também foi avaliada no sistema de manipulador robótico. Esse sistema não-linear afim vem sendo usado para avaliação de controladores não-lineares (PAN et al., 2011). Com este experimento, pode-se verificar o poder de generalização do controle baseado no LARFSOM para um problema não-linear afim com os parâmetros variando no tempo. Neste experimento, há, também, uma avaliação do controle na presença de distúrbios externos.

Primeiramente, serão mostradas as equações dinâmicas do sistema robótico. Em seguida, será calculada a linearização do sistema. Logo em seguida, haverá uma descrição do experimento. Depois, será descrita a abordagem LQR aplicada para controlar o sistema e serão apresentados os resultados obtidos pelo LQR. Por fim, serão descritos os detalhes dos LARFSOMs aplicados para controlar o sistema e seus resultados.

### 8.1 SISTEMA DINÂMICO DO MANIPULADOR ROBÓTICO

Considere um manipulador robótico de único elo com a seguinte modelagem dinâmica:

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2, u) = x_2 \\ \dot{x}_2 &= f_2(t, x_1, x_2, u) = -(d_r x_2 + m_r g_v l_r \cos(x_1))/J + (1/J)u + d \\ y &= g(t, x_1, x_2, u) = x_1\end{aligned}\tag{8.1}$$

onde  $x_1$  é a posição angular do manipulador,  $x_2$  é a velocidade angular,  $m_r$  é a massa de carga,  $l_r$  é o comprimento do manipulador,  $J = (1,33)m_r l_r^2$  é o momento de inércia e  $d_r$  é o fator de amortecimento. O distúrbio externo, representado por  $d$ , é um modelo de atrito composto por uma combinação de atrito de Coulomb e de atrito viscoso.

$$d = -(sgn(x_2)c_r + v_r x_2)/J\tag{8.2}$$

onde  $c_r \in \mathbb{R}^+$  é o torque provocado pelo atrito de Coulomb, e  $v_r \in \mathbb{R}^+$  é o coeficiente do atrito dinâmico. A Figura 26a apresenta o modelo do manipulador robótico e a Figura 26b mostra o distúrbio externo ao longo do tempo em uma simulação do sistema variante no tempo.

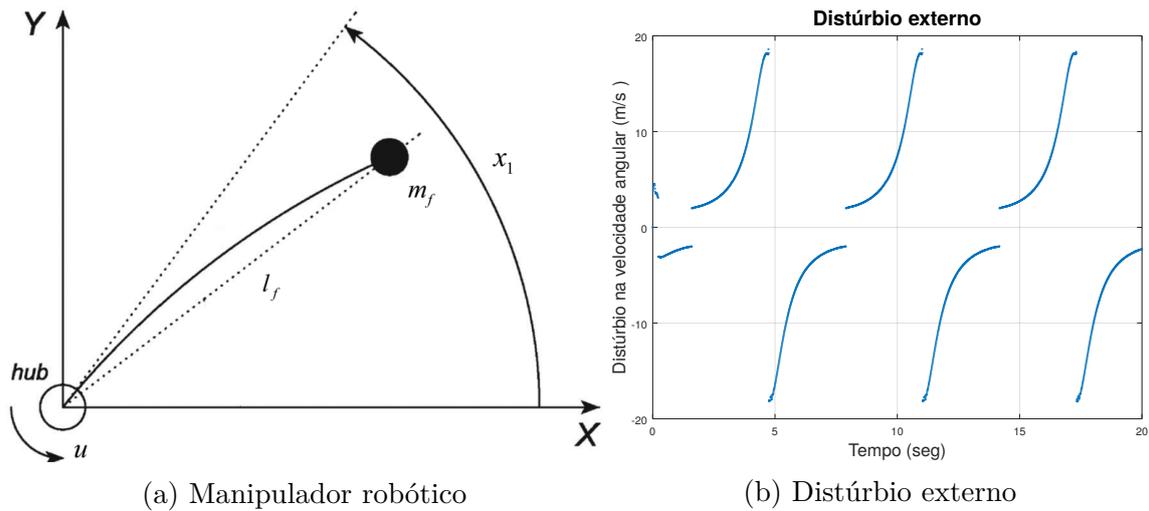


Figura 26 – (a) diagrama esquemático do manipulador robótico de único elo. (b) distúrbio externo ocorrido durante uma simulação do sistema com variação da massa de carga durante o tempo.

## 8.2 LINEARIZAÇÃO DO SISTEMA

Foi realizada a linearização do sistema para aplicação do controle LQR. A linearização foi feita através do cálculo da matriz Jacobiana do sistema. Então, o sistema linearizado é descrito da seguinte forma:

$$\delta \dot{\mathbf{x}} = A_j \cdot \delta \mathbf{x} + B_j \cdot \delta u \quad (8.3)$$

onde  $A_j$  é a matriz jacobiana. Então, definindo matematicamente para o sistema do manipulador robótico tem-se:

$$A_j = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ \frac{m_r g_v l_r \sin(x_1)}{J} & \frac{-d_r}{J} \end{vmatrix} \quad (8.4)$$

$$B_j = \begin{vmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{vmatrix} = \begin{vmatrix} 0 \\ \frac{1}{J} \end{vmatrix} \quad (8.5)$$

Foi desconsiderado o distúrbio externo no cálculo da linearização, pois esta linearização só é usada para projetar o controle LQR e ele é aplicado sem o conhecimento deste distúrbio.

## 8.3 DESCRIÇÃO DO EXPERIMENTO

Nesta seção, são explicados os detalhes da execução do experimento do manipulador robótico. No cenário de teste, a massa de carga vai variar no tempo conforme esta função:  $m_r = 5 + 4\text{sen}(t)\text{kg}$ . Sendo assim, o sistema do manipulador será um sistema

não-linear afim variante no tempo. Neste experimento, foram utilizadas 5 configurações de treinamento com diferentes valores de massa de carga. Os valores usados foram:  $m_r \in \{1; 2,5; 5; 7,5; 9\}$ . Para os demais parâmetros do sistema, os valores escolhidos foram:  $l_r = 0,25m$ ,  $d_r = 2kg \cdot m^2/s$ ,  $c_r = 1,5$ ,  $v_r = 0,3$ , e  $\mathbf{x}(0) = [\pi/6 \ 0]^T$ . O objetivo do controle é garantir que o ângulo do manipulador rastreie o sinal de referência  $\mathbf{r}(t) = [\text{sen}(t) \ \text{cos}(t)]^T$ . As simulações foram executadas usando o método de Runge-Kutta de 4ª ordem com tamanho de passo  $h = 0,01$  [seg]. Este experimento foi abordado da mesma forma que foi aplicado no trabalho de Pan et al. (2011).

Para construir os dados de treinamento, sistemas de controle foram simulados considerando individualmente cada configuração do sistema para treinamento. Nessas simulações, foi aplicado o controle LQR com integradores. Mais detalhes dessa estratégia de controle serão descritas na próxima seção.

Para cada configuração do sistema de treinamento, simulação de 20 segundos foram executadas considerando o mesmo sinal de referência,  $\mathbf{r}(t) = [\text{sen}(t) \ \text{cos}(t)]^T$ . Para formar o conjunto de treinamento de fato, foram construídas tuplas com os dados  $\{\text{vetor do estado do sistema } \mathbf{x}(t), \text{ sinal dos erros}^1 \mathbf{e}(t), \text{ o sinal de controle dos integradores } \mathbf{v}(t), \text{ ganhos da malha de realimentação } \mathbf{K}(t)\}$  do mesmo instante  $t$  provenientes das simulações. Os dados de treinamento foram normalizados para que todas as variáveis ficassem com média igual a 0 e desvio padrão igual a 1. A partir desses dados, os vetores concatenados de entrada de tamanho  $p$  foram gerados.

## 8.4 RESULTADOS DOS EXPERIMENTOS

Nesta seção, são descritos os detalhes de cada modelo avaliado e são apresentados os resultados obtidos nas simulações do experimento do manipulador robótico.

### 8.4.1 Descrição do LQR com Integradores

Para controlar o manipulador foi utilizada uma abordagem do LQR mais dois integradores. Em experimentos iniciais, usando somente o LQR, houve um valor de erro estacionário grande. Uma das formas de contornar esta situação é colocar elementos integrativos para que o controlador rastreie adequadamente o valor desejado.

O projeto do LQR utilizado para rastrear a saída desejada foi diferente do aplicado no experimento do pêndulo invertido. Neste experimento, foi implementada uma outra alternativa de adaptar o LQR em uma estratégia de controle para o problema de rastreamento. Os ganhos de realimentação foram calculados como estão explicadas na Subseção B e foi usada a linearização do sistema no estado atual do sistema para projeta-los. Isto é, a

<sup>1</sup> Como na etapa de projeto de controle usando LQR considerou integradores sobre os erros, o sinal de erros foi na prática as integrais dos erros de rastreamento das variáveis  $x_1$  e  $x_2$ .

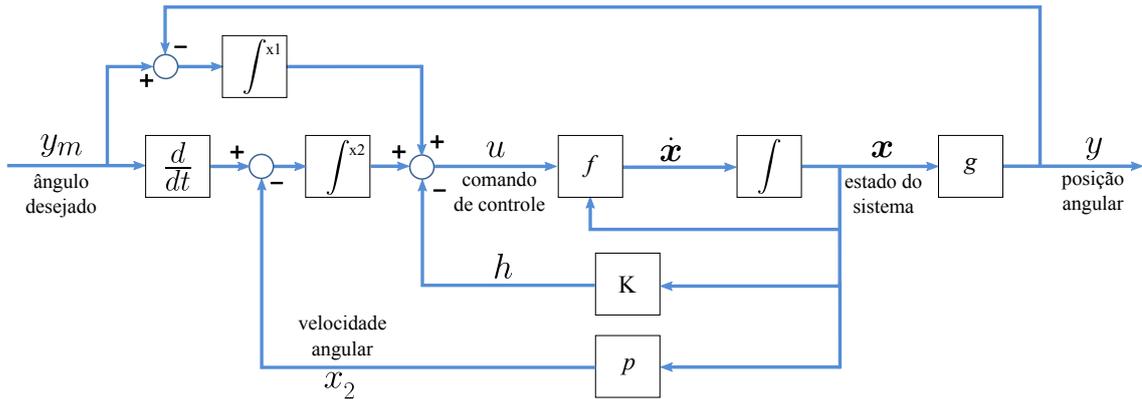


Figura 27 – Diagrama de blocos do sistema controlado pelo LQR mais dois integradores para garantir um erro pequeno no estado estacionário.

cada novo estado atual do sistema, o sistema é linearizado e novos valores de ganho de realimentação são computados.

Para converter o LQR em uma solução para o problema de rastreamento, foram adicionados dois integradores. Há dois integradores visto que melhores rastreamentos foram obtidos através de uma estratégia de controle da posição angular desejada e da velocidade angular desejada ao mesmo tempo. Então, o comando de controle aplicado à planta será da seguinte forma:

$$u = u_{I^{x_1}} + u_{I^{x_2}} - \mathbf{K}^T \cdot \mathbf{x} \quad (8.6)$$

Os termos  $u_{I^{x_1}}$  e  $u_{I^{x_2}}$  são os sinais de controle dos integradores do erro da posição angular e do erro da velocidade angular do sistema, respectivamente. Os dois termos  $u_{I^{x_1}}$  e  $u_{I^{x_2}}$  formam o sinal de controle  $\mathbf{v} = [u_{I^{x_1}} \ u_{I^{x_2}}]^T$ . Dessa maneira, o sistema de controle projetado usando o LQR com integradores ficou com a configuração mostrada no diagrama de bloco da Figura 27, na qual  $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, u)$  é uma função composta por  $f_1$  e  $f_2$  da Equação 8.1.

#### 8.4.2 Resultados LQR

Nesta subseção, são mostrados os resultados obtidos em simulações usando o controlador LQR com integradores. Foram feitos experimentos com este controle convencional para gerar um conjunto de treinamento para o controle LARFSOM. Os parâmetros usados no controle LQR foram  $Q = \text{diag}(250, 250)$ , onde  $\text{diag}$  significa matriz diagonal, e  $R = 1$  para todas configurações de treinamento exceto para configura de massa  $m_r = 1\text{kg}$ . Na configuração  $m_r = 1\text{kg}$  foi usada  $R = 10$ . Os valores dos ganhos dos integradores estão apresentados na Tabela 13. O ajuste dos parâmetros foi por tentativa e erro. Os resultados das simulações estão apresentados na Figura 28. Como foi descrito anteriormente, nestas simulações não houve mudança na massa de carga e não há presença dos distúrbios externos.

Tabela 13 – Ganhos dos integradores para todas as massas de carga de treinamento ( $m_r$ ).

Parâmetro	Valor
$K_{I^{x_1}}$	2550
$K_{I^{x_2}}$	550

A Tabela 14 exibe uma avaliação do desempenho do controle LQR com integradores para este experimento. Os índices de desempenho avaliados para este experimento foram:  $J(\text{IAE}) = \int_0^t |\mathbf{e}(\tau)| d\tau$ ,  $J(\text{ITAE}) = \int_0^t \tau |\mathbf{e}(\tau)| d\tau$ , a energia do sinal de controle  $E_c = \int_0^t u^2(\tau) d\tau$ , tempo de estabilização  $t_s$  (*settling time*) com um erro limite de 5% e os percentuais de erro no estado estacionário denotados por  $e_{s1}\%$  e  $e_{s2}\%$  referentes aos erros  $e_1$  e  $e_2$ , respectivamente.

Tabela 14 – Medidas de desempenho avaliadas nas simulações para gerar o conjunto de treinamento.

Medida	$m_r = 1$	$m_r = 2,5$	$m_r = 5$	$m_r = 7,5$	$m_r = 9$
$E_c$	119	514	1912	4236	6076
$J(\text{IAE})$	0,698	0,667	0,699	0,751	0,790
$J(\text{ITAE})$	1,800	1,477	1,779	2,282	2,659
$t_s$	0,52	0,46	0,44	0,42	0,41
$e_{s1}\%$	0,618	0,821	1,018	1,220	1,345
$e_{s2}\%$	0,511	0,281	0,782	1,401	1,819

#### 8.4.3 Resultados do Controlador por Realimentação de Estado Baseado no LARFSOM

Nesta subseção, são descritos alguns detalhes de como foram abordados os LARFSOMs para controlar o sistema do manipulador robótico e são apresentados os resultados. Dois LARFSOMs são usadas para controlar o sistema. O  $LARFSOM^k$  infere os ganhos de malha de realimentação  $\mathbf{K}$  e o  $LARFSOM^v$  estima sinal de controle gerado pelos integradores  $\mathbf{v} = [u_{I^{x_1}} \ u_{I^{x_2}}]^T$ . Os parâmetros de treinamento estão apresentados na Tabela 15.

Para avaliar o resultado do sistema de controle baseado nos LARFSOMs, foram combinados os LARFSOMs de mesmo índice de treinamento. Sendo assim, há 10 configurações de experimentos. Nesta seção, todos os resultados apresentados foram obtidos pela configuração que obteve menor  $J(\text{IAE})$ .

Nas Figuras 29 e 30, são apresentados os resultados das simulações usando a estratégia de controle LARFSOM. Pode-se notar que, em todos os casos, o rastreamento da posição angular foi muito próximo do valor de referência ( $sen(t)$ ) e apresentou um comportamento suave para esta variável (Figs 29a, 29b, e 30a). Em particular, o controle baseado

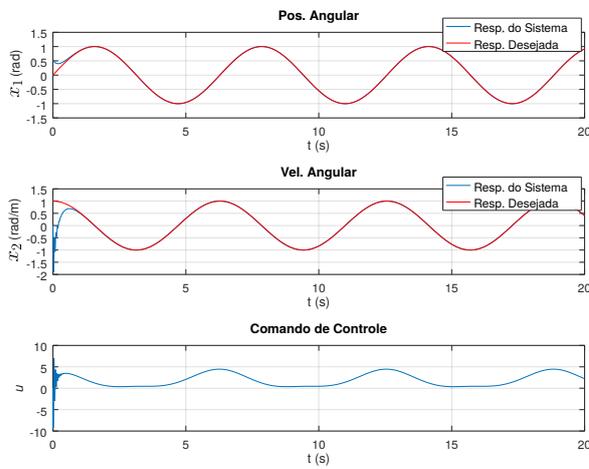
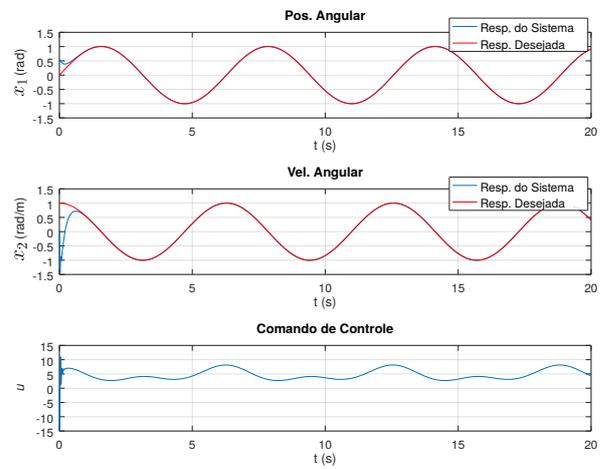
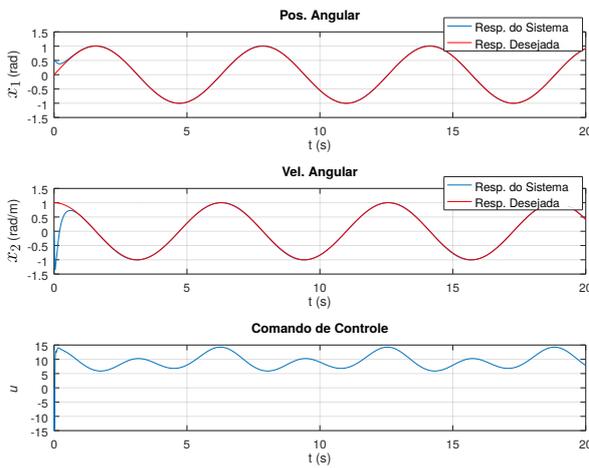
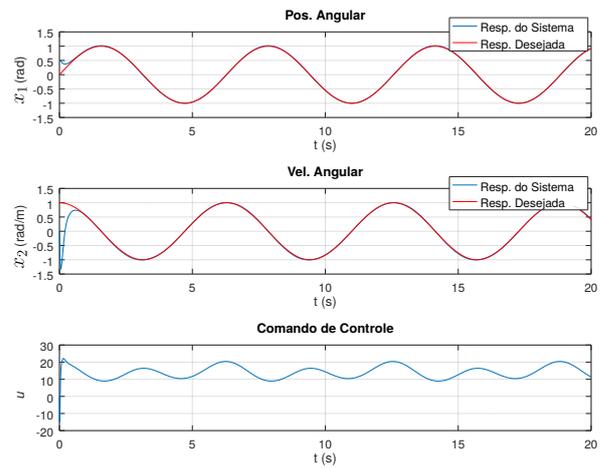
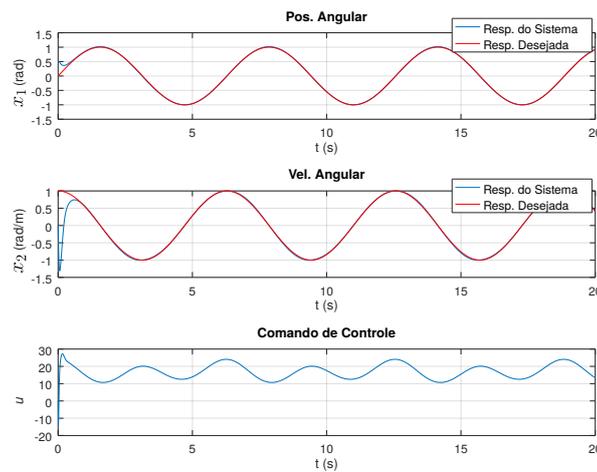
(a)  $m_r = 1 \text{ kg}$ (b)  $m_r = 2,5 \text{ kg}$ (c)  $m_r = 5 \text{ kg}$ (d)  $m_r = 7,5 \text{ kg}$ (e)  $m_r = 9 \text{ kg}$ 

Figura 28 – Resultado da simulação para as configurações do sistema para treinamento.

Tabela 15 – Valores dos parâmetros usados no treinamento das LARFSOMs

Parâmetro	Valor					
	LARFSOM <sup>v</sup>	LARFSOM <sup>k</sup>	D-LARFSOM <sup>v</sup>	D-LARFSOM <sup>k</sup>	xD-LARFSOM <sup>v</sup>	xD-LARFSOM <sup>k</sup>
$\rho_f$	0,001	0,001	0,001	0,001	0,001	0,001
$\varepsilon$	0,4	0,4	0,4	0,4	0,4	0,4
$d_m$	100	100	100	100	100	100
$p$	4	9	4	9	4	9
$t_{max}$	5	5	50	15	50	15
$N_{max}$	500	60	50	30	50	30
$\alpha_T$	0,7	0,03	0,2	0,015	0,2	0,015
$\alpha_k$	-	-	0,01	0,01	0,001	0,001

na versão discreta do LARFSOM não conseguiu seguir a velocidade angular desejada de uma maneira suave. Esse fenômeno está principalmente associado a qualidade da aproximação do sinal de controle  $\mathbf{v}$  que garante um erro de rastreamento em nível desejado. O LARFSOM versão discreta não foi suficientemente acurado para aproximar a ação de controle dos integradores, e conseqüentemente, a velocidade angular tem um comportamento não suave (Fig 29c). Entretanto, esse fenômeno foi significativamente reduzido usando o D-LARFSOM e xD-LARFSOM (Figs 29b e 30a) justamente pelo fato desses mapas aproximarem de maneira mais acurada as ações de controle dos integradores (Figs 29d e 30b). É importante também ressaltar sobre os ganhos da malha de realimentação inferidos pelos LARFSOMs. Os ganhos estimados não precisam ser necessariamente muito próximos dos valores produzidos pelo LQR. Como discutido na Seção 5.1.2, há uma boa margem de tolerância visto que eles são bem robustos. Isso pode ser observado pela diferença entre os valores inferidos pelos LARFSOMs e o LQR nas Figuras 29c, 29d e 30b.

Na Tabela 16, são mostrados os índices de desempenho obtidos pelos controles LARFSOMs. Esses resultados podem ser comparados com três controles adaptativos fuzzy (AFC). Os três controladores são caracterizados por utilizar um sistema de lógica fuzzy para modelar o controlador e estimar a ação de controle ideal. Também, todos os três AFC são controles adaptativos diretos com suas regras de adaptação derivadas do teorema de estabilidade de *Lyapunov*. Phan e Gale (2006) propuseram o primeiro AFC selecionado para comparação. Esse AFC se baseia no teorema da aproximação universal de sistema fuzzy e consegue estimar a ação de controle considerando o conhecimento do sinal do ganho de controle. Pan, Huang e Sun (2009) propuseram o segundo AFC escolhido para a análise comparativa. Nesse AFC, há um modulo fuzzy adicional que atua como um compensador adaptativo. Sua função é estimar o erro de aproximação ótimo e reduzir a influência do erro da aproximação no desempenho do sistema. Pan et al. (2011) também propuseram o terceiro AFC. Nessa proposição, eles reformularam seu trabalho anterior reprojando o controle adaptativo direto utilizando um sistema fuzzy com universo de discurso variável. O universo de discurso varia de acordo com fatores de contração-expansão dependentes das variáveis de entrada. Esse universo de discurso variável pode proporcionar mudanças

Tabela 16 – Resultado das medidas de desempenho nas simulações usando controle por realimentação de estados baseado no LARFSOM.

Índice	LARFSOM	D-LARFSOM	xD-LARFSOM	AFC <sup>a,b</sup>	AFC <sup>a,c</sup>	AFC <sup>a,d</sup>
$J(\text{IAE})$	4,479	0,837	0,849	1,065	0,817	<b>0,622</b>
$J(\text{ITAE})$	39,220	4,431	3,373	4,725	1,641	<b>0,877</b>
$E_c$	2909,8	2517,3	2524,4	2389	2386	<b>2384</b>
$t_s$	0,96	0,45	<b>0,44</b>	2,910	3,010	0,735
$e_{s1}\%$	2,191	1,081	1,049	4,012	0,779	<b>0,563</b>
$e_{s2}\%$	64,112	1,283	0,821	11,40	5,540	<b>0,405</b>

<sup>a</sup> Os resultados dos Controles Adaptativos Fuzzy AFC foram retirados do artigo de (PAN et al., 2011).

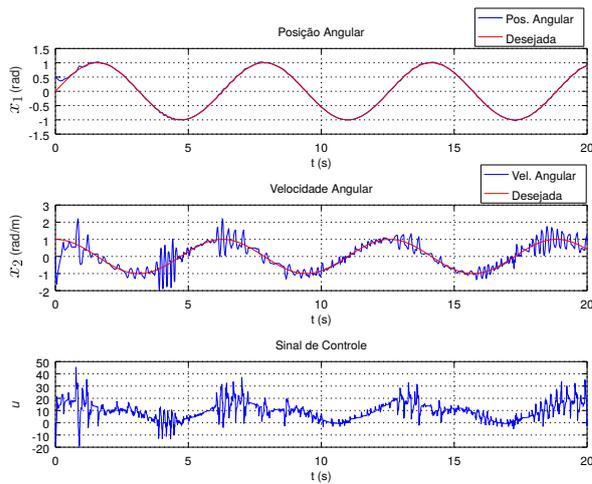
<sup>b</sup> AFC proposto por Phan e Gale (2006).

<sup>c</sup> AFC proposto por Pan, Huang e Sun (2009).

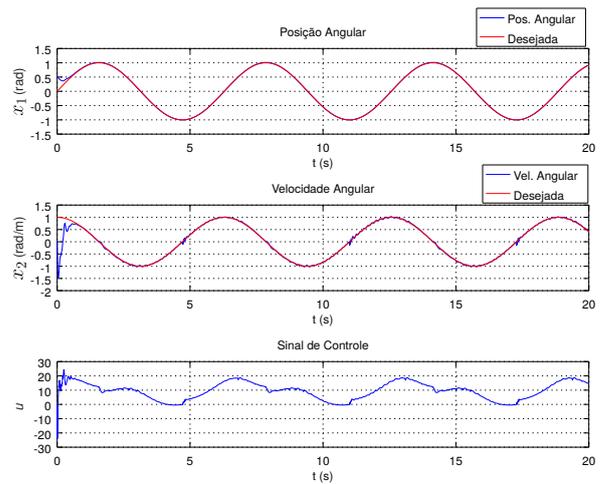
<sup>d</sup> AFC proposto por Pan et al. (2011).

nas regras fuzzy, que melhoravam a acurácia da saída do sistema fuzzy.

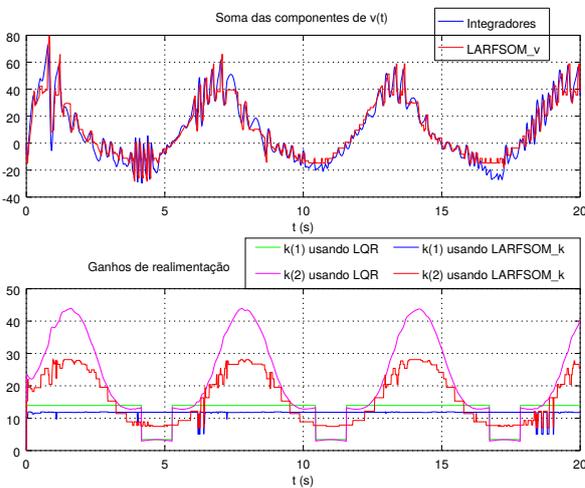
Os controles baseados nos D- e xD-LARFSOM conseguiram melhores índices de desempenho do pior AFC com exceção da energia do sinal de controle. Em comparação ao segundo melhor AFC, os controles baseados no LARFSOM foram melhores em termos de tempo de estabilização  $t_s$  e do erro de estado estacionário  $e_{s2}\%$ . Já em comparação ao melhor AFC, os controles baseado no LARFSOM foram melhores em termos de tempo de estabilização  $t_s$ . Esse tempo de estabilização reduzido é uma característica de controle em múltiplos modelos (NARENDRA; BALAKRISHNAN, 1997).



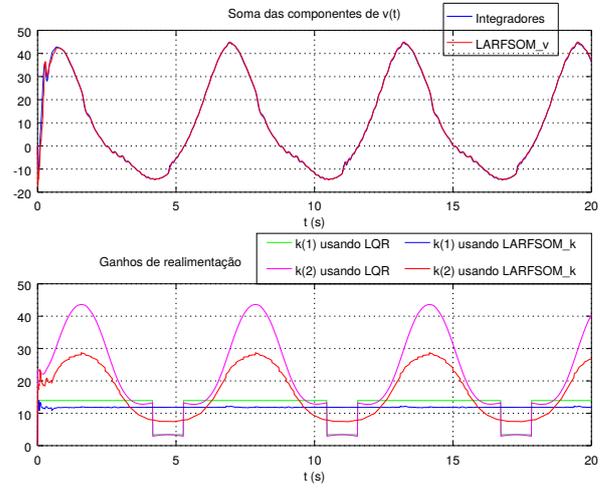
(a) LARFSOM



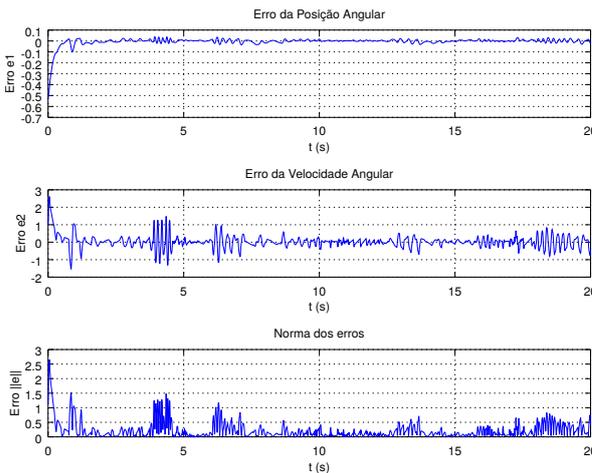
(b) D-LARFSOM



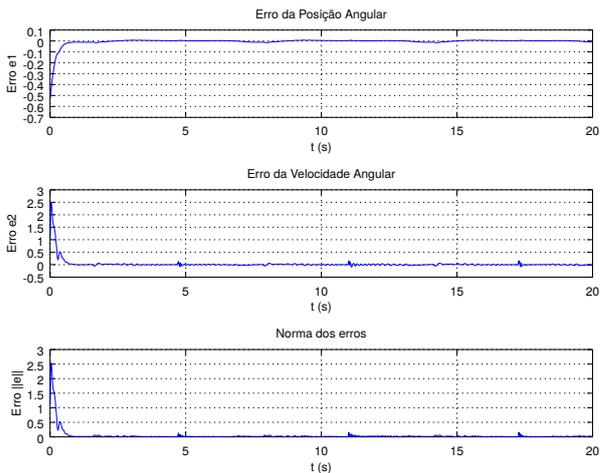
(c) LARFSOM



(d) D-LARFSOM

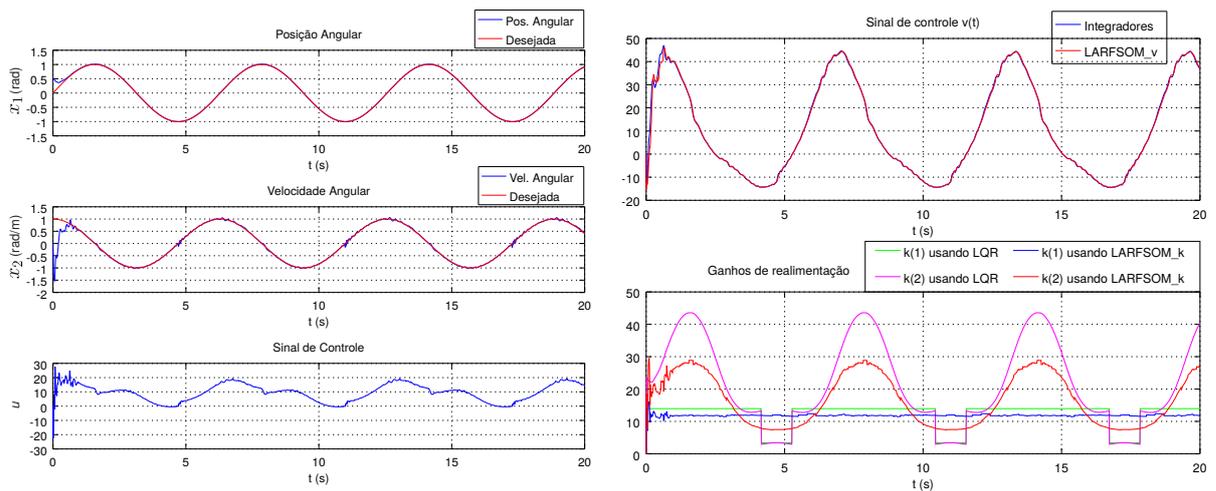


(e) LARFSOM



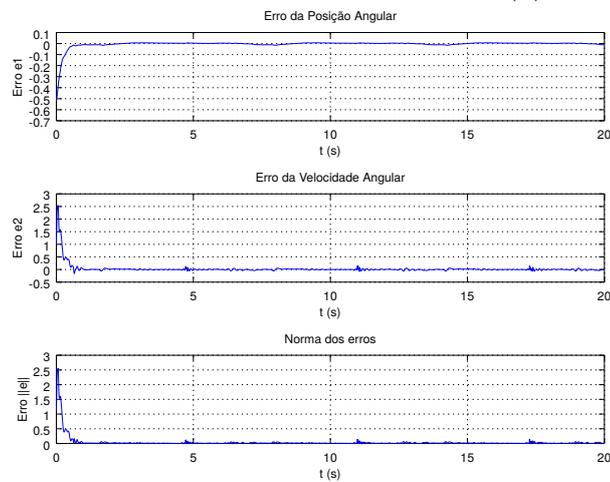
(f) D-LARFSOM

Figura 29 – Resultado das simulações do manipulador robótico usando o LARFSOM e D-LARFSOM. (a) e (b) mostram o estado do sistema  $\mathbf{x}$  e o sinal de controle  $u$ . (c) e (d) mostram o sinal de controle  $u$ , o sinal da malha de realimentação  $h$  e os ganhos da malha de realimentação  $\mathbf{K}$ . (e) e (f) mostram os erros de rastreamento.



(a) xD-LARFSOM

(b) xD-LARFSOM



(c) xD-LARFSOM

Figura 30 – Resultado das simulações do manipulador robótico usando o xD-LARFSOM. (a) mostra o estado do sistema  $\boldsymbol{x}$  e o sinal de controle  $u$ . (b) mostra o sinal de controle  $u$ , o sinal da malha de realimentação  $h$  e os ganhos da malha de realimentação  $\boldsymbol{K}$ . (c) mostram os erros de rastreamento.

## 9 EXPERIMENTOS NO OSCILADOR DE VAN DER POL

Para demonstrar a eficácia da abordagem de *controle adaptativo direto baseado no LARF-SOM*, foi considerado o sistema oscilante de Van der Pol. O oscilador de Van der Pol é um sistema de segunda ordem amortecido não-linearmente. Neste experimento, além da análise da capacidade de controle em um problema não-linear com os parâmetros variando no tempo, há uma avaliação do controle no rastreamento de um modelo de referência não-linear. Foi utilizado oscilador de Duffing como modelo de referência.

### 9.1 SISTEMA OSCILANTE DE VAN DER POL

O oscilador de Van der Pol pode ser modelado pelas seguintes equações dinâmicas (PAN; YU, 2017):

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\mathbf{x}, \beta) + u \\ y &= x_1\end{aligned}\tag{9.1}$$

onde  $f(\mathbf{x}, \beta) = -x_1 + \beta(1 - x_1^2)x_2$ . Os termos  $\mathbf{x} = [x_1 \ x_2]^T$ ,  $u$  e  $y$  são o vetor de estado, a entrada de controle e a saída do sistema, respectivamente. O parâmetro variante no tempo  $\beta$  é referente a força de amortecimento e pondera a não-linearidade do sistema.

### 9.2 MODELO DE REFERÊNCIA - OSCILADOR DE DUFFING

O modelo de referência escolhido foi oscilador de Duffing.

$$\begin{aligned}\dot{x}_{d1} &= x_{d2} \\ \dot{x}_{d2} &= -p_1 x_{d2} - p_2 x_{d1} - p_3 x_{d2}^3 + q \cos(wt)\end{aligned}\tag{9.2}$$

onde  $p_1$ ,  $p_2$ ,  $p_3$ ,  $q$  e  $w$  são parâmetros conhecidos. Para as simulações, os valores dos parâmetros foram  $p_1 = 0,4$ ,  $p_2 = -1,1$ ,  $p_3 = 1,0$ ,  $q = 1,498$  e  $w = 1,8$ . O estado inicial do modelo de referência  $\mathbf{x}_d(0) = [0,2 \ 0,3]^T$ . Esses valores foram os mesmos usados em (PAN; YU, 2017).

### 9.3 DESCRIÇÃO DO EXPERIMENTO

Nesta seção, são explicados os detalhes da execução do experimento do oscilador de Van der Pol. Neste experimento, foram utilizadas cinco configurações de treinamento com diferentes valores do parâmetro  $\beta$ . Os valores usados foram:  $\beta = [0,5 \ 1 \ 1,5 \ 2,0 \ 2,5]$ . Para gerar os dados de treinamento, o sistema é simulado para cada configuração de treinamento por 120 segundos. Nesse caso, o valor de  $f(\mathbf{x}, \beta)$  é conhecido e a lei de controle aplicada

é uma linearização por realimentação:

$$u = -f(\mathbf{x}, \beta) + k_v r + \lambda_1 \dot{e} + \dot{x}_{d2} \quad (9.3)$$

onde  $k_v = 10$  e  $\lambda_1 = 10$ . Os dados de treinamento provenientes das simulações são organizados em vetores concatenados de treinamento  $\boldsymbol{\xi}$  formado por *{vetores de estado concatenados  $\mathbf{x}^-$ , parâmetro do sistema  $\beta$ , valor da função  $f(\mathbf{x}, \beta)$ }*, isto é:

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{x}(t) \\ \vdots \\ \mathbf{x}(t-p+1) \\ \mathbf{p}_f \\ f(\mathbf{x}, \beta) \end{bmatrix} \quad (9.4)$$

onde  $p$  é o comprimento da janela de tempo. Os dados de treinamento foram normalizados de modo que todas variáveis ficassem com média igual a 0 e desvio padrão igual a 1.

Nas simulações de teste, o valor de  $\beta$  muda a cada 50 segundos e a ordem dos valores é  $[\beta_A = 0,6 \rightarrow \beta_B = 1 \rightarrow \beta_C = 1,4 \rightarrow \beta_D = 1,7 \rightarrow \beta_E = 2,2]$ . As simulações foram realizadas usando GNU Octave 4 e executadas através do método de Runge-Kutta de quarta ordem como o comprimento de passo  $h = 0,01$ .

O Controle adaptativo direto baseado no LARFSOM foi treinado com os parâmetros apresentados na Tabela 17. Para a lei de controle, também foram usados os valores  $k_v = 10$  e  $\lambda_1 = 10$ . Para a lei de adaptação, a matriz  $F$  é uma matriz diagonal de 0,1.

Tabela 17 – Parâmetros de treinamento do LARFSOM.

Parâmetros	Valor
$\rho_f$	0,001
$\varepsilon$	0,4
$a_T$	0,05
$v_m$	100
$t_{max}$	100
$N_{max}$	50
$p$	9

#### 9.4 RESULTADOS DO CONTROLADOR ADAPTATIVO DIRETO BASEADO NO LARFSOM

Nesta seção, são apresentados os resultados obtidos nas simulações do experimento do oscilador de Van der Pol. Na Figura 31, pode-se notar que o sistema de controle pode

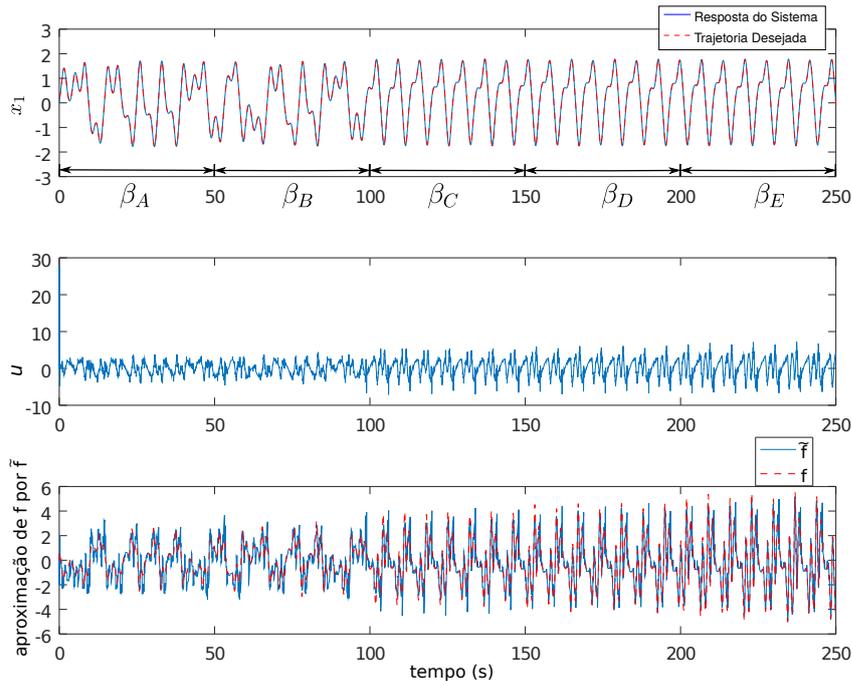


Figura 31 – Resultado do controle adaptativo direto baseado no LARFSOM. Reposta do sistema e desejada, sinal de controle, e aproximação de  $f$  por  $\tilde{f}$  usando o D-LARFSOM.

seguir o modelo de referência ( $x_{d1}$ ). As Figuras 32a e 32b mostram a resposta transiente e estacionária, respectivamente. Nota-se que o D-LARFSOM não aproxima de maneira acurada a função  $f$  durante o regime transitório (Fig. 32a). Porém, a acurácia aumenta significativamente durante o período de estado estacionário (Fig. 32b).

Para fazer uma análise comparativa, a lei de controle da linearização por realimentação (Eq. 9.3) foi escolhida como *benchmark*. Nesse caso, o valor da função  $f$  é conhecido, e os mesmos valores  $k_v = 10$  e  $\lambda_1 = 10$  foram usados. A Tabela 18 apresenta os resultados de medidas de desempenho. Os resultados do controle adaptativo direto baseado no LARFSOM foram similares ao da linearização por realimentação.

Tabela 18 – Resultado das medidas de desempenhos.

Medida	Benchmark	D-LARFSOM
$J_{x_1}$ (IAE)	2,438	2,323
$E_c$ (Energia)	1041	1090
$t_s$ (tempo de estabilização)	0,26	0,25

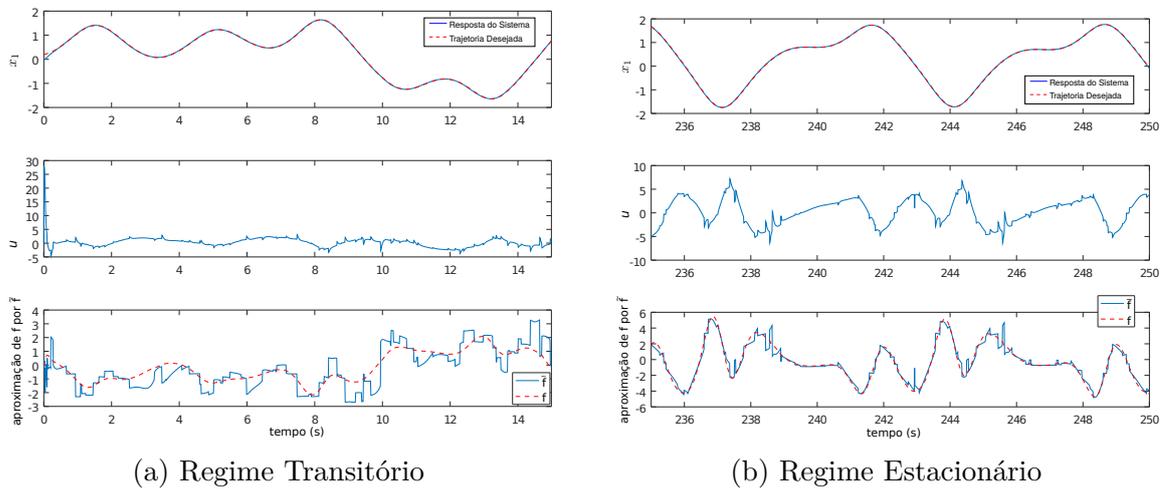


Figura 32 – Resultado usando o controle adaptativo direto baseado no LARFSOM. (a) Resposta transitório no intervalo  $t \in [0s \ 15s]$ . (b) Resposta no estado estacionário no intervalo  $t \in [235s \ 250s]$ .

## 10 CONCLUSÃO

Nesse trabalho de doutorado, foram propostas estratégias de controle baseadas em mapas auto-organizáveis para o problema de controle de sistema não-linear variante no tempo. Especificamente, as propostas foram concebidas empregando um mapa variante do SOM chamado de *mapa auto-organizável com campo receptivo adaptativo local* - LARFSOM (ARAÚJO; COSTA, 2009). O LARFSOM foi aplicado com o objetivo de dividir o espaço de operação do sistema, identificar o comportamento do sistema e adquirir ações de controle para cada comportamento do sistema. Nesse trabalho, o LARFSOM foi estendido para ter a capacidade de mapeamento contínuo por partes, e conseqüentemente, melhorar o desempenho de sistema de controle.

A inspiração de projetar controles a partir de mapa auto-organizável, chamados nesse trabalho de *controles baseados em SOM*, vem de alguns fatores presentes nessa classe de redes neurais. O primeiro deles é aprender sem supervisor (KOHONEN, 2013). A aprendizagem não-supervisionada pode simplificar o controle, e também, pode permitir aprendizagem incremental (BARRETO; ARAÚJO; RITTER, 2003). O SOM é um modelo não-linear baseado em dados capaz de aprender um mapeamento não-linear. Esse mapeamento, além de ser interpretável, pode ser aplicado para resolver certas ambigüidades existentes nos dados. Outro fator é que o SOM é adequado para a estratégia de controle com múltiplos modelos, e vem sendo aplicado com sucesso para projetar controladores capazes de terem respostas adequadas mesmo que haja mudanças repentinas no objeto controlado (PRINCIPE; WANG; MOTTER, 1998; CHO et al., 2007; KIUMARSI; LEWIS; LEVINE, 2015; ZHAO et al., 2016). O SOM também tem sido aplicado com sucesso na identificação de sistemas dinâmicos lineares e não-lineares (BARRETO, 2007). O último fator é o poder de generalização que tem o SOM e suas variantes. Essa característica dá a possibilidade de esquematizar o controle para ser robusto a comportamentos não apresentados na etapa de treinamento.

Assim, foi estudada, além das técnicas tradicionais de controles não-lineares, a subclasse de controle inteligente composta de controles neurais que usavam mapas auto-organizáveis. Foram identificadas as características relevantes desse tipo de controle para o problema de sistemas de controle. Também, foram apresentadas as limitações dessa subclasse com possíveis soluções para contorná-las.

O LARFSOM, assim como o SOM, é caracterizado por um processo de quantização vetorial. Isto, *a priori*, não impede a sua aplicação em aproximar funções contínuas. Mas, em problemas de controle de sistema, é importante atingir níveis altos de acurácia na identificação e na geração da ação de controle para sistemas com alta sensibilidade a variações paramétricas. Sendo assim, este trabalho de doutorado estudou e propôs técnicas de mapeamento contínuo por partes para mapas auto-organizáveis. Inicialmente, duas técnicas propostas por Flentge (2006), a interpolação linear por partes e a interpolação linear

por partes localmente ponderada, foram estudadas em detalhes e implementadas. Observando limitações dessas técnicas, foram desenvolvidas duas outras técnicas para substituir a interpolação linear por partes, chamadas de: método da diferença e método da diferença estendido. Esses dois métodos com *aprendizagem localmente ponderada* produzem outras duas técnicas: método da diferença localmente ponderado e método da diferença estendido localmente ponderado.

A partir dos estudos realizados, foram propostas duas estratégias de controles baseadas no LARFSOM. A primeira proposta foi projetada utilizando a estratégia de controle por realimentação de estados em um esquema de múltiplos modelos. Nessa proposta, dois LARFSOMs foram empregados. Ambos são capazes de dividir o espaço de operação do sistema e aprender ações de controle a partir de dados provenientes de simulações de controles em realimentação de estados para diferentes configurações do sistema. Então, os papéis dos LARFSOMs consistem em aprender comportamentos do sistema para diferentes situações de controle, aprender ações de controle para cada situação, identificar o atual comportamento do sistema desconhecido na presença de possíveis perturbações incertas e selecionar a ação de controle apropriada para o comportamento identificado.

A segunda proposta de controle nasce da necessidade de garantias de convergência. Assim como a primeira, o LARFSOM foi usado para dividir o espaço de comportamentos do sistema em regiões locais, identificar o comportamento do sistema e aprender ações de controle para cada comportamento. Mais especificamente, em cada região local, há um controlador adaptativo direto, que é responsável por inferir o comportamento não-linear do sistema. A partir dessa estimativa, é possível utilizar a técnica de linearização por realimentação para gerar o sinal de controle. A teoria de estabilidade de Lyapunov foi utilizada para garantir a convergência do erro de rastreamento e dos pesos do controlador local vencedor.

As técnicas da literatura e os métodos propostos para mapeamento contínuo usando o LARFSOM foram validados em experimentos sintéticos e em problemas de *benchmark*. Os resultados dos experimentos usando dados sintéticos sugerem que os métodos propostos têm as seguintes propriedades: aproximação acurada de funções não-lineares, capacidade de aprendizagem incremental, capacidade para aproximação de funções com saída variante no tempo e distribuição de entrada variante no tempo, e robustez a dados irrelevantes e redundantes. É interessante frisar que os LARFSOMs com os métodos de mapeamento contínuo por partes (D-, xD-, LWD-, e LWxD-LARFSOM) têm menos parâmetros para serem ajustados do que o LWINGG. Os resultados dos experimentos sobre os problemas de *benchmark* sugerem que os mapas LWD-LARFSOM e LWxD-LARFSOM são algoritmos de aprendizagem competitivos em termos de acurácia em comparação ao SVR. Assim como também nos resultados com dados sintéticos, foi observada uma significativa melhora no tempo de treinamento e no tempo de resposta nos LWD- e LWxD-LARFSOM em comparação ao LWI-LARFSOM. Isso pode ser explicado pelo fato que LWD- e LWxD-

LARFSOM são menos custosos computacionalmente, e precisaram de menos nodos e menos iterações de treinamento do que LWI-LARFSOM. Vale ressaltar também que os métodos desenvolvidos D-, xD-, LWD-, e LWxD-LARFSOM potencialmente podem ser empregados em outras classes de problemas como predição de séries temporais, problemas de classificação e até reconstrução de superfícies de objetos como foi feito por Ivriissimtzis, Jeong e Seidel (2003) e Rêgo, Ferreira e Araújo (2011).

Para realizar uma avaliação do controle por realimentação de estados baseado no LARFSOM, dois sistemas foram selecionados para simulações. Um deles foi o pêndulo invertido. Sistema conceituado como um dos mais fundamentais da teoria de controle não-linear e um dos mais usados para avaliar controles em sistema de natureza instável (BOUBAKER, 2013). Outro foi o manipulador robótico de um único elo. Em todas as simulações, os parâmetros dos sistemas variaram ao longo do tempo. Os resultados sugerem que o método proposto em realimentação de estados tem dois pontos importantes: (1) controla adequadamente em cenários de parâmetros variando abruptamente ou suavemente ao longo da execução e (2) ser eficaz ao lidar com configurações não apresentadas na etapa de treinamento. Além disso, este controlador também apresentou robustez a distúrbios externos nas simulações do manipulador robótico. Comparações entre os LARFSOMs com mapeamento contínuo por partes e o LARFSOM discreto, mostrou uma melhora na qualidade dos índices de desempenho ao utilizar os LARFSOMs contínuo por partes. Este controle proposto também foi comparado com três controles adaptativos fuzzy (AFC). O controle mostrou ser competitivo em comparação aos AFCs e apresentou melhores índices de tempo de estabilização.

O controle adaptativo direto baseado no LARFSOM foi avaliado em simulações do oscilador de Van der Pol. Neste experimento, o modelo de referência foi o oscilador de Duffing, que também é um sistema não-linear. Nas simulações, a componente não-linear do sistema de Van der Pol variou abruptamente ao longo do tempo. Esse controle foi comparado com o método de controle, linearização por realimentação, considerando o conhecimento exato da componente não-linear do sistema. Os resultados sugerem que o controlador adaptativo direto baseado no LARFSOM foi capaz de modelar a não-linearidade do sistema ao longo da fase de operação, e, com isso, controlar adequadamente o sistema. Os valores dos índices de desempenho, erro de rastreamento, energia do sinal e tempo de estabilização, foram semelhantes aos obtidos pela técnica de controle sabendo a não-linearidade do sistema.

## 10.1 TRABALHO PUBLICADO

O artigo relativo aos estudos e desenvolvimento do LARFSOM com mapeamento contínuo por partes para aproximação de função variante no tempo foi aceito para publicação no periódico *Neural Processing Letters*.

- Ferreira, P. H., Araújo, A. F., 2020. Growing Self-Organizing Maps for Nonlinear Time-Varying Function Approximation. *Neural Processing Letters*.

## 10.2 LIMITAÇÕES DAS ABORDAGENS

Acerca das técnicas de mapeamento contínuo por partes para SOM, o método da diferença localmente ponderado (LWD) e o método da diferença estendido localmente ponderado (LWxD) apresentam ainda um custo computacional elevado. Tanto o tempo de treinamento quanto o tempo de resposta do mapeamento podem ser reduzidos.

Controles em múltiplos modelos em geral são mais difíceis de avaliar matematicamente as características de estabilidade (NARENDRA; BALAKRISHNAN, 1997) e isso refletiu no *controle por realimentação de estados baseado no LARFSOM*. Então, a principal limitação é ausência de uma análise de estabilidade desse controle. Outra limitação importante é uma avaliação da robustez desse controle. Como o *controle por realimentação de estados baseado no LARFSOM* aprende a partir de dados do LQR para diferentes configurações do sistema, é esperado que esse controle seja capaz de controlar o sistema para as configurações usados para gerar os dados de treinamento e para novas configurações. Porém, até que ponto essas novas configurações podem ser diferentes das configurações apresentadas na etapa de treinamento? Então, é necessário uma análise da robustez dos controladores usados para etapa de treinamento para inferir um nível de robustez desse controle baseado no LARFSOM.

A segunda proposta de controle apresentada nesta tese visou atacar a principal limitação da primeira. Porém, houve a necessidade de delimitar uma classe de sistemas não-lineares variantes no tempo. Logo, a principal limitação é ser uma estratégia de controle aplicável somente a sistemas que podem ser modelados dentro da classe alvo. Uma outra limitação que atinge ambos métodos de controle propostos é a necessidade de gerar dados para treinamento a partir de sistemas de controle previamente estabelecidos.

## 10.3 TRABALHOS FUTUROS

As limitações das abordagens desenvolvidas durante o doutorado indicam diretrizes para trabalhos futuros. Para contornar a limitação do custo computacional do mapeamento contínuo por partes usando LWD-LARFSOM e LWxD-LARFSOM, futuros trabalhos envolve a otimização da busca do vencedor através de técnicas de acesso espacial (DENNY; GOZALI; MANURUNG, 2016) e uma implementação apropriada para GPU. Também, outras técnicas de interpolação e combinação de modelos locais podem ser consideradas em uma remodelagem do LWD- e LWxD-LARFSOM para redução do número de parâmetros. A técnica de interpolação e combinação dos modelos realizadas pelo kernel de Voronoi induzido (IVK) proposto por Aupetit, Couturier e Massotte (2001) podem ser uma diretriz para essa redução do número de parâmetros. Além desses, sugere-se: uma investigação

---

e análise comparativa da informação topológica como realizada por Aupetit (2006), e o desenvolvimento de um processo de auto-organização que incorpore o contexto semântico como desenvolvido por Hartono, Hollensen e Trappenberg (2015).

A partir dos LWD- e LWxD-LARFSOM otimizados, a proposta de controle por realimentação de estado baseada nesses dois mapas pode ser avaliada. Após validações dos modelos de mapeamento contínuo por partes propostos e realizadas as avaliações das estratégias de controle em simulações, o curso natural de futuros esforços está na implementação e avaliação do controle em um sistema real. Um possível sistema real seria o controle de movimento de uma mão robótica. Neste cenário, o sistema de controle tem o desafio de lidar com diferentes configurações da mão robótica. Algumas delas são: controle de movimento de precisão ou de força; controle de movimento no ato de agarrar diferentes tipos de objetos; e controle de movimento com ou sem objeto na mão.

Para evitar a necessidade de projetar controladores previamente para gerar o conjunto de treinamento, a técnica de *babbling* pode ser utilizada (ROLF; STEIL; GIENGER, 2010). Também, pode ser considerado um controle baseado no LARFSOM sem uma etapa prévia de treinamento. Conceitualmente é verificada através dos experimentos de validação, os LARFSOMs estendidos para tarefas de aproximação de funções variantes no tempo têm a característica de aprender incrementalmente. Logo, poderia ser considerado o desenvolvimento de um controle baseado no LARFSOM com uma aprendizagem acontecendo continuamente e somente durante a etapa de operação. Nesse sentido, a segunda proposta de controle tem um potencial maior, visto que já tem uma regra de ajuste para os controladores locais acontecendo continuamente durante o regime de operação. Necessitaria de modificações para adquirir e refinar as informações sobre a divisão do espaço de operação do sistema e identificação do sistema.

Além dos trabalhos futuros já mencionados, sugere-se para o controle adaptativo direto baseado no LARFSOM: (1) continuar o desenvolvimento do controlador seguindo os passos de Kar e Behera (2009) para controlar uma classe de sistemas não-lineares afins com múltiplas entradas e múltiplas saídas; e (2) remodelar a regra de ajuste dos controladores locais considerando uma classe de sistema não-lineares não-afim como foi feito por Park et al. (2005).

## REFERÊNCIAS

- AL-MUSAYLH, M. S.; DEO, R. C.; ADAMOWSKI, J. F.; LI, Y. Short-term electricity demand forecasting with mars, svr and arima models using aggregated demand data in queensland, australia. *Advanced Engineering Informatics*, v. 35, p. 1 – 16, 2018.
- ALBUS, J. A theory of intelligent systems. In: *Proceedings in IEEE International Symposium on Intelligent Control*. [S.l.: s.n.], 1990. p. 866–875 vol.2.
- ARAÚJO, A. F. R.; RÊGO, R. L. M. E. Self-organizing maps with a time-varying structure. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 1, p. 7:1–7:38, jul. 2013.
- ARAÚJO, A. R.; COSTA, D. C. Local adaptive receptive field self-organizing map for image color segmentation. *Image and Vision Computing*, Elsevier, v. 27, n. 9, p. 1229–1239, 2009.
- ASSUNÇÃO, E.; ANDREA, C. Q.; TEIXEIRA, M. C. M. Controle ótimo  $h_2$  e  $h_\infty$  com modificação de zeros para o problema de rastreamento usando lmi. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 15, n. 4, p. 413–422, 2004.
- ÅSTRÖM, K.; WITTENMARK, B. *Adaptive Control*. [S.l.]: Addison-Wesley, 1995.
- ÅSTRÖM, K. J.; HÄGGLUND, T. The future of pid control. *Control Engineering Practice*, v. 9, n. 11, p. 1163 – 1175, 2001. PID Control.
- AUPETIT, M. Learning topology with the generative gaussian graph and the em algorithm. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2006. p. 83–90.
- AUPETIT, M.; COUTURIER, P.; MASSOTTE, P. Induced voronoï kernels for principal manifolds approximation. In: *Advances in Self-Organising Maps*. [S.l.]: Springer, 2001. p. 73–80.
- BABAZADEH, R.; KHIABANI, A. G.; AZMI, H. Optimal control of segway personal transporter. In: *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*. [S.l.: s.n.], 2016. p. 18–22.
- BACHE, K.; LICHMAN, M. Uci machine learning repository. 2013.
- BARRETO, G. Time series prediction with the self-organizing map: A review. In: HAMMER, B.; HITZLER, P. (Ed.). *Perspectives of Neural-Symbolic Integration*. [S.l.]: Springer Berlin Heidelberg, 2007. v. 77, p. 135–158.
- BARRETO, G. A.; ARAÚJO, A. F.; RITTER, H. J. Self-organizing feature maps for modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, Springer, v. 36, n. 4, p. 407–450, 2003.
- 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, Sept 2004.

- BASAK, D.; PAL, S.; PATRANABIS, D. C. Support vector regression. *Neural Information Processing-Letters and Reviews*, v. 11, n. 10, p. 203–224, 2007.
- BENSON, T. *Rocket Stability*. : , 2014. <<https://www.grc.nasa.gov/WWW/K-12/rocket/rktstab.html>>. Acessado em: 10-01-2020.
- BEZANSON, J.; EDELMAN, A.; KARPINSKI, S.; SHAH, V. B. Julia: A fresh approach to numerical computing. *SIAM Review*, v. 59, n. 1, p. 65–98, 2017.
- BEZANSON, J.; KARPINSKI, S.; SHAH, V. B.; EDELMAN, A. *Julia: A Fast Dynamic Language for Technical Computing*. : , 2012.
- BLONDIN, M. J.; SÁEZ, J. S.; PARDALOS, P. M. Control engineering from classical to intelligent control theory—an overview. In: \_\_\_\_\_. *Computational Intelligence and Optimization Methods for Control Engineering*. Cham: Springer International Publishing, 2019. p. 1–30. ISBN 978-3-030-25446-9. Disponível em: <[https://doi.org/10.1007/978-3-030-25446-9\\_1](https://doi.org/10.1007/978-3-030-25446-9_1)>.
- BOUBAKER, O. The inverted pendulum benchmark in nonlinear control theory: A survey. *Int J Adv Robotic Sy*, v. 10, n. 233, 2013.
- CAMPOS, M. M.; CARPENTER, G. A. Building adaptive basis functions with a continuous self-organizing map. *Neural Processing Letters*, v. 11, n. 1, p. 59–78, Feb 2000.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>.
- CHEN, S.-Y.; GONG, S.-S. Speed tracking control of pneumatic motor servo systems using observation-based adaptive dynamic sliding-mode control. *Mechanical Systems and Signal Processing*, v. 94, p. 111 – 128, 2017.
- CHO, J.; PRINCIPE, J. C.; ERDOGMUS, D.; MOTTER, M. A. Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map. *Neural Networks, IEEE Transactions on*, IEEE, v. 17, n. 2, p. 445–460, 2006.
- CHO, J.; PRINCIPE, J. C.; ERDOGMUS, D.; MOTTER, M. A. Quasi-sliding mode control strategy based on multiple-linear models. *Neurocomputing*, Elsevier, v. 70, n. 4, p. 960 – 974, 2007.
- CHO, J.; PRINCIPE, J. C.; MOTTER, M. A. A local linear modeling paradigm with a modified counterpropagation network. In: *Proceedings of the International Joint Conference on Neural Networks, 2003*. [S.l.: s.n.], 2003. v. 1, p. 34–38 vol.1.
- DAYUAN, G.; ZHENG, M.; HAI, Z. Modeling and control of unmanned aerial vehicle using self-organizing map multiple models. In: *Control and Decision Conference (CCDC), 2013 25th Chinese*. [S.l.: s.n.], 2013. p. 4177–4182.
- DELGADO, A. Control of nonlinear systems using a self-organising neural network. *Neural Computing & Applications*, v. 9, n. 2, p. 113–123, Jul 2000.
- DENNY; GOZALI, W.; MANURUNG, R. Som training optimization using triangle inequality. In: MERÉNYI, E.; MENDENHALL, M. J.; O’DRISCOLL, P. (Ed.). *Advances in Self-Organizing Maps and Learning Vector Quantization*. Cham: Springer International Publishing, 2016. p. 61–71. ISBN 978-3-319-28518-4.

- DÍAZ-VICO, D.; TORRES-BARRÁN, A.; OMARI, A.; DORRONSORO, J. R. Deep neural networks for wind and solar energy prediction. *Neural Processing Letters*, v. 46, n. 3, p. 829–844, Dec 2017.
- DORF, R.; BISHOP, R. Sistemas de controle modernos. In: \_\_\_\_\_. [S.l.]: Livros Técnicos e Científicos, 2010. cap. Sistemas de Controle Robustos, p. 533–582. ISBN 9788521612421.
- EKRE, F. et al. *Performance Tips*. : , 2019. <<https://github.com/JuliaLang/julia/blob/master/doc/src/manual/performance-tips.md>>. Acessado em: 10-01-2020.
- FLENTGE, F. Locally weighted interpolating growing neural gas. *IEEE Transactions on Neural Networks*, v. 17, n. 6, p. 1382–1393, Nov 2006.
- FLORES, J. V.; SILVA, J. M. G. d. Estratégia de controle para o seguimento de referências em sistemas de tempo discreto com atuadores saturantes. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 21, n. 1, p. 14–28, 2010.
- GÖPPERT, J.; ROSENSTIEL, W. Interpolation in SOM: Improved generalization by iterative methods. In: SOULIÉ, F. F.; GALLINARI, P. (Ed.). *Proc. International Conference on Artificial Neural Networks (ICANN)*. Nanterre, France: EC2, 1995. II, p. 69–74.
- GÖPPERT, J.; ROSENTIEL, W. The continuous interpolating self-organizing map. *Neural Processing Letters*, v. 5, n. 3, p. 185–192, 1997.
- HADDAD, W.; CHELLABOINA, V. Nonlinear dynamical systems and control: A lyapunov-based approach. In: \_\_\_\_\_. [S.l.]: Princeton University Press, 2011. cap. Optimal Nonlinear Feedback Control, p. 511 – 554.
- HARTONO, P.; HOLLENSSEN, P.; TRAPPENBERG, T. Learning-regulated context relevant topographical map. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 10, p. 2323–2335, Oct 2015.
- HAYKIN, S. S. Neural networks and learning machines. In: \_\_\_\_\_. [S.l.]: Prentice Hall, 2009. cap. Self-Organizing Maps, p. 425 – 470.
- HE, X.; ASADA, H. A new method for identifying orders of input-output models for nonlinear dynamic systems. In: *1993 American Control Conference*. [S.l.: s.n.], 1993. p. 2520–2523.
- HECHT, T.; LEFORT, M.; GEPPERTH, A. Using self-organizing maps for regression: the importance of the output function. In: *European Symposium on Artificial Neural Networks (ESANN)*. Bruges, Belgium: , 2015.
- HOU, Z.-S.; WANG, Z. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, v. 235, n. 0, p. 3 – 35, 2013. Data-based Control, Decision, Scheduling and Fault Diagnostics.
- HÜLSEN, H. *Self-organising locally interpolating maps in control engineering*. Tese (Doutorado) — Universität Oldenburg, 2007.

- HUNT, K.; SBARBARO, D.; ŻBIKOWSKI, R.; GAWTHROP, P. Neural networks for control systems—a survey. *Automatica*, v. 28, n. 6, p. 1083 – 1112, 1992.
- IVRISSIMTZIS, I. V.; JEONG, W. .; SEIDEL, H. . Using growing cell structures for surface reconstruction. In: *Shape Modeling International*. [S.l.: s.n.], 2003. p. 78–86.
- KALMAN, R. E. Contributions to the theory of optimal control. *Boletín de la Sociedad Matemática Mexicana*, v. 5, n. 2, p. 102–119, 1960.
- KANGAS, J. Time-delayed self-organizing maps. In: IEEE. *Neural Networks, 1990., 1990 IJCNN International Joint Conference on. : , 1990*. p. 331–336.
- KAR, I.; BEHERA, L. Direct adaptive neural control for affine nonlinear systems. *Applied Soft Computing*, v. 9, n. 2, p. 756 – 764, 2009.
- KHALIL, H. *Nonlinear Systems*. [S.l.]: Prentice Hall, 2002.
- KIUMARSI, B.; LEWIS, F. L.; LEVINE, D. S. 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.
- KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464–1480, Sep 1990.
- KOHONEN, T. Applications. In: *Self-Organizing Maps*. [S.l.]: Springer Berlin Heidelberg, 2001. v. 30, p. 263–310.
- KOHONEN, T. An overview of SOM literature. In: *Self-Organizing Maps*. [S.l.]: Springer Berlin Heidelberg, 2001. v. 30, p. 347–371.
- KOHONEN, T. Essentials of the self-organizing map. *Neural Networks*, v. 37, p. 52 – 65, 2013. Twenty-fifth Anniversary Commemorative Issue.
- LAFLAMME, S.; SLOTINE, J. J. E.; CONNOR, J. J. Wavelet network for semi-active control. *Journal of Engineering Mechanics*, American Society of Civil Engineers, , v. 137, n. 7, p. 462–474, Jul 2011.
- LAFLAMME, S.; SLOTINE, J. J. E.; CONNOR, J. J. Self-organizing input space for control of structures. *Smart Materials and Structures*, IOP Publishing, , v. 21, n. 11, p. 115015, NOV 2012.
- LAWRENCE, S.; TSOI, A. C.; BACK, A. D. Function approximation with neural networks and local methods: Bias, variance and smoothness. In: *Proc. Australian Conference on Neural Networks (ACNN)*. [S.l.: s.n.], 1996. v. 1621.
- LEONARDI, F.; CRUZ, J. J. d. Projeto de controladores robustos com especificações temporais. *SBA: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 16, n. 3, p. 279–289, 2005.
- LEWIS, F. Control system advanced methods. In: \_\_\_\_\_. 2. ed. [S.l.]: CRC Press, 2010. cap. Optimal Control, p. 25–1–25–35.

- LI, S.; FANG, H.; LIU, X. Parameter optimization of support vector regression based on sine cosine algorithm. *Expert Systems with Applications*, v. 91, p. 63 – 77, 2018.
- LIU, L.; XIAO, J.; YU, L. Interval self-organizing map for nonlinear system identification and control. In: SUN, F.; ZHANG, J.; TAN, Y.; CAO, J.; YU, W. (Ed.). *Advances in Neural Networks - ISNN 2008*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 78–86.
- LIU, X.-L.; DUAN, G.-R.; TEO, K.-L. Optimal soft landing control for moon lander. *Automatica*, v. 44, n. 4, p. 1097 – 1103, 2008.
- LOCATELLI, A. Optimal control: An introduction. In: \_\_\_\_\_. [S.l.]: Springer, 2001. cap. Introduction, p. 1 – 5.
- LUDWIG, L.; KESSLER, W.; GÖPPERT, J.; ROSENSTIEL, W. Som with topological interpolation for the prediction of interference spectra. In: *In Proceedings of EANN*. Helsinki, Finland: [s.n.], 1995. p. 379–389.
- MALTONI, D.; LOMONACO, V. Continuous learning in single-incremental-task scenarios. *Neural Networks*, v. 116, p. 56 – 73, 2019.
- MARTINETZ, T. M.; RITTER, H. J.; SCHULTEN, K. J. Three-dimensional neural net for learning visuomotor coordination of a robot arm. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 131–136, March 1990.
- MINATOHARA, T.; FURUKAWA, T. The self-organizing adaptive controller. *International Journal of Innovative Computing, Information and Control (IJICIC)*, ICIC INT TOKAI UNIV, 9-1-1, TOROKU, KUMAMOTO, 862-8652, JAPAN, v. 7, n. 4, p. 1933–1947, 2011.
- MOSHOU, D.; RAMON, H. Extended self-organizing maps with local linear mappings for function approximation and system identification. In: *Proc. Workshop on Self-Organizing Maps (WSOM)*. [S.l.: s.n.], 1997.
- MOTTER, M. A. Predictive multiple model switching control with the self-organizing map. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. [S.l.: s.n.], 2000. v. 4, p. 317–322 vol.4.
- NARENDRA, K. Chapter two - hierarchical adaptive control of rapidly time-varying systems using multiple models. In: VAMVOUDAKIS, K. G.; JAGANNATHAN, S. (Ed.). *Control of Complex Systems*. [S.l.]: Butterworth-Heinemann, 2016. p. 33 – 66.
- NARENDRA, K. S.; BALAKRISHNAN, J. Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, v. 42, n. 2, p. 171–187, Feb 1997.
- NARENDRA, K. S.; BALAKRISHNAN, J.; CILIZ, M. K. Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems*, v. 15, n. 3, p. 37–51, Jun 1995.
- NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 4–27, Mar 1990.

- 
- NISE, N. S. Control systems engineering. In: \_\_\_\_\_. 6. ed. [S.l.]: John Wiley & Sons, Incorporated, 2011. cap. Introduction, p. 1 – 31.
- OGATA, K. Engenharia de controle moderno. In: \_\_\_\_\_. [S.l.]: Pearson Prentice Hall, 2011. cap. Introdução aos Sistemas de Controle, p. 1 – 18.
- PAN, Y.; ER, M. J.; HUANG, D.; WANG, Q. Adaptive fuzzy control with guaranteed convergence of optimal approximation error. *IEEE Transactions on Fuzzy Systems*, v. 19, n. 5, p. 807–818, Oct 2011.
- PAN, Y.; HUANG, D.; SUN, Z. Direct adaptive fuzzy control for a class of nonlinear systems with unknown bounds. In: *IEEE International Conference on Automation and Logistics*. [S.l.: s.n.], 2009. p. 949–954.
- PAN, Y.; YU, H. Biomimetic hybrid feedback feedforward neural-network learning control. *IEEE Transactions on Neural Networks and Learning Systems*, v. 28, n. 6, p. 1481–1487, June 2017.
- PARISI, G. I.; KEMKER, R.; PART, J. L.; KANAN, C.; WERMTER, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, v. 113, p. 54 – 71, 2019.
- PARK, J.-H.; HUH, S.-H.; KIM, S.-H.; SEO, S.-J.; PARK, G.-T. Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks. *IEEE Transactions on Neural Networks*, v. 16, n. 2, p. 414–422, March 2005.
- PASSINO, K. Control system advanced methods. In: \_\_\_\_\_. 2 ed.. ed. [S.l.]: CRC Press, 2010. cap. Intelligent Control, p. 54–1–54–12.
- PHAN, P. A.; GALE, T. J. Direct adaptive fuzzy control with less restrictions on the control gain. In: *International Conference on Computational Intelligence for Modelling Control and Automation*. [S.l.: s.n.], 2006. p. 168–168.
- POLYCARPOU, M.; FARRELL, J. Control system advanced methods. In: \_\_\_\_\_. 2 ed.. ed. [S.l.]: CRC Press, 2010. cap. Neural Control, p. 56–1–56–20.
- PRINCIPE, J. C.; WANG, L.; MOTTER, M. A. Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control. *Proceedings of the IEEE, IEEE*, v. 86, n. 11, p. 2240–2258, 1998.
- RASMUSSEN, C. E.; NEAL, R. M.; HINTON, G.; CAMP, D. van; REVOW, M.; GHAHRAMANI, Z.; KUSTRA, R.; TIBSHIRANI, R. Delve data for evaluating learning in valid experiments. 1996.
- RÊGO, R. L. M. E.; FERREIRA, P. H. M.; ARAÚJO, A. F. R. Reconstructing anatomical structures with growing self-reconstruction maps. In: *IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.: s.n.], 2011. p. 2976–2981.
- RITTER, H.; MARTINETZ, T.; SCHULTEN, K. *Neural Computation and Self-Organizing Maps; An Introduction*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.

- RIZK, R.; KRUT, S.; DOMBRE, E. Grasp-stability analysis of a two-phalanx isotropic underactuated finger. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. [S.l.: s.n.], 2007. p. 3289–3294.
- ROLF, M.; STEIL, J. J.; GIENGER, M. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, v. 2, n. 3, p. 216–229, Sep. 2010.
- SALCEDO-SANZ, S.; ROJO-ÁLVAREZ, J. L.; MARTÍNEZ-RAMÓN, M.; CAMPS-VALLS, G. Support vector machines in engineering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 4, n. 3, p. 234–267, 2014.
- SBARBARO, D.; BASSI, D. A nonlinear controller based on self-organizing maps. In: *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*. [S.l.: s.n.], 1995. v. 2, p. 1774–1777 vol.2.
- SCHAAL, S.; ATKESON, C. G. Constructive incremental learning from only local information. *Neural Computation*, v. 10, n. 8, p. 2047–2084, 1998.
- SCHÖLKOPF, B.; SMOLA, A. J.; WILLIAMSON, R. C.; BARTLETT, P. L. New support vector algorithms. *Neural Computation*, v. 12, n. 5, p. 1207–1245, 2000.
- SIDDIQUE, N. Intelligent control: A hybrid approach based on fuzzy logic, neural networks and genetic algorithms. In: \_\_\_\_\_. *Intelligent Control: A Hybrid Approach Based on Fuzzy Logic, Neural Networks and Genetic Algorithms*. Cham: Springer International Publishing, 2014. cap. Stability Analysis of Intelligent Controllers, p. 243–267.
- SLOTINE, J.; LI, W. *Applied Nonlinear Control*. [S.l.]: Prentice Hall, 1991.
- SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004.
- SOLTANPOUR, M. R.; KHOOBAN, M. H.; KHALGHANI, M. R. An optimal and intelligent control strategy for a class of nonlinear systems: adaptive fuzzy sliding mode. *Journal of Vibration and Control*, v. 22, n. 1, p. 159–175, 2016.
- Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15, n. 1, p. 116–132, 1985.
- TAN, C.; TAO, G.; QI, R.; YANG, H. A direct mrac based multivariable multiple-model switching control scheme. *Automatica*, v. 84, p. 190 – 198, 2017.
- TEWARI, A. Advanced control of aircraft, spacecraft and rockets. In: \_\_\_\_\_. [S.l.]: Wiley, 2011. cap. Optimal Guidance of Rockets, p. 195 – 275.
- THAMPI, G.; PRINCIPE, J. C.; CHO, J.; MOTTER, M. Adaptive inverse control using SOM based multiple models. In: *Proc. Portuguese Conf. Automatic Control*. [S.l.: s.n.], 2002. p. 278–282.
- TOKUNAGA, K.; FURUKAWA, T. Modular network som. *Neural Networks*, v. 22, n. 1, p. 82 – 90, 2009.

- 
- VIJAYAKUMAR, S.; D'SOUZA, A.; SCHAAL, S. Incremental online learning in high dimensions. *Neural Computation*, v. 17, n. 12, p. 2602–2634, Dec 2005.
- WALTER, J. *Rapid Learning in Robotics*. [S.l.]: Cuvillier Verlag, 1997.
- WALTER, J.; RITTER, H. Rapid learning with parametrized self-organizing maps. *Neurocomputing*, Elsevier, v. 12, n. 2, p. 131–153, 1996.
- WANG, C.; HILL, D. J. Learning from neural control. *IEEE Transactions on Neural Networks*, v. 17, n. 1, p. 130–146, Jan 2006.
- WANG, J.; LI, Y. Short-term wind speed prediction using signal preprocessing technique and evolutionary support vector regression. *Neural Processing Letters*, v. 48, n. 2, p. 1043–1061, Oct 2018.
- WINTER, M.; METTA, G.; SANDINI, G. Neural-gas for function approximation: a heuristic for minimizing the local estimation error. In: *Proc. Internacional Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2000. v. 4, p. 535–538 vol.4.
- WU, J.; CHEN, W.; LI, J. Fuzzy-approximation-based global adaptive control for uncertain strict-feedback systems with a priori known tracking accuracy. *Fuzzy Sets and Systems*, v. 273, p. 1 – 25, 2015. Theme: Control Engineering.
- ZHAO, Y.; FARRELL, J. A. Self-organizing approximation-based control for higher order systems. *IEEE Transactions on Neural Networks*, v. 18, n. 4, p. 1220–1231, 2007.
- ZHAO, Z.; QIU, Z.; ZHANG, X.; HAN, J. Vibration control of a pneumatic driven piezoelectric flexible manipulator using self-organizing map based multiple models. *Mechanical Systems and Signal Processing*, v. 70-71, p. 345 – 372, 2016.

## APÊNDICE A – ESTABILIDADE

As ferramentas matemáticas para demonstração de estabilidade de um sistema dinâmico são fundamentais na análise e no projeto de sistemas de controle. Através delas é possível encontrar garantias que o sistema opera de maneira segura e pode executar o resultado esperado mesmo em face de incertezas no modelo da planta, variações durante a execução e a presença de perturbações. As abordagens de estabilidade do sistema podem ser divididas em duas grandes classes (SIDDIQUE, 2014):

- Teoria Quantitativa;
- Teoria da Estabilidade Geral.

Na teoria quantitativa de sistemas dinâmicos, índices de estabilidade e robustez são usados. Há duas classes principais de índices. Uma delas é relacionada à estabilidade relativa e dá uma medida do grau de estabilidade no ponto de equilíbrio na origem. Já a outra classe é referente à estabilidade global do sistema e avalia quão distante o sistema está de uma bifurcação originada por um par atrator-repulsor. Esses índices já foram usados para analisar e projetar sistemas de controle (SIDDIQUE, 2014).

Já a teoria da estabilidade geral se divide em dois ramos principais, são eles:

1. Estabilidade no sentido de *Lyapunov*;
2. Estabilidade entrada-saída.

### A.1 TEORIA DE ESTABILIDADE DE LYAPUNOV

A Análise de estabilidade de Lyapunov é uma análise do comportamento do sistema próximo do seu ponto de equilíbrio. Considere um sistema dinâmico formulado por:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \quad (\text{A.1})$$

É dito que  $\mathbf{x}_e$  é um ponto de equilíbrio se  $\mathbf{f}(\mathbf{x}_e) = 0$ . O significado disso é que  $\dot{\mathbf{x}} = 0$ , e conseqüentemente, no ponto de equilíbrio, não há nenhuma tendência de mudança de estado. Um ponto de equilíbrio é dito *estável no sentido de Lyapunov* se para todo  $\epsilon \geq 0$  e qualquer  $t_0 \geq 0$  existe um raio  $\sigma(t_0, \epsilon)$  tal que:

$$|\mathbf{x}_0 - \mathbf{x}_e| < \sigma(t_0, \epsilon) \Rightarrow |\mathbf{x}(t) - \mathbf{x}_e| < \epsilon \quad (\text{A.2})$$

onde  $\mathbf{x}(t)$  é a solução do sistema (Eq. A.1) e  $\mathbf{x}_0 = \mathbf{x}(t_0)$ . Dizer que o sistema é estável pode ser interpretado da seguinte forma: se o estado inicial do sistema é suficientemente

próximo de um ponto de equilíbrio estável, os estados do sistema permanecem ao longo de todo tempo  $t$  tão próximos do ponto de equilíbrio quanto quisermos.

Já um ponto de equilíbrio  $\mathbf{x}_e$  é *assintoticamente estável* no sentido de Lyapunov se  $\mathbf{x}_e$  for estável e

$$|\mathbf{x}(t) - \mathbf{x}_e| \rightarrow 0, \quad t \rightarrow \infty \quad (\text{A.3})$$

Dizer que o sistema é assintoticamente estável pode ser interpretado da seguinte forma: se o estado inicial do sistema é suficientemente próximo de um ponto de equilíbrio assintoticamente estável, os estados do sistema convergirão para ele ao longo do tempo  $t$ . Também é dito que o ponto de equilíbrio  $\mathbf{x}_e$  é instável quando ele não for estável.

Um conceito importante para sistemas variantes no tempo, é a noção de *estabilidade uniforme*. É dito que o sistema é *uniformemente estável*, quando o raio  $\sigma(\epsilon)$  é independência em relação ao instante de tempo inicial  $t_0$ . O mesmo vale para *uniformemente assintoticamente estável*. Por último, tem-se a noção de *estabilidade assintótica global*. É dito que o sistema é globalmente assintoticamente estável, quando a condição de convergência para o ponto de equilíbrio assintoticamente estável é independente do estado inicial do sistema.

Há duas formas de determinar a estabilidade de um ponto de equilíbrio. Uma é o método indireto e a outra é o método direto. O método direto é a maneira mais robusta de determinar a estabilidade. Através dela, é capaz de analisar sistemas não-lineares e sistemas variantes no tempo. Também, é possível determinar a estabilidade assintótica e a estabilidade ordinária. Além disso, esse método pode estabelecer a região de estabilidade assintótica, como também, o domínio de atração de um equilíbrio. É uma ferramenta importante para o projeto de controle para sistemas não-lineares com garantias de estabilidade. O principal ponto negativo é que não há uma maneira sistemática de encontrar a função de Lyapunov, fato crucial para a análise da estabilidade.

## A.2 ESTABILIDADE ENTRADA-SAÍDA

A teoria de estabilidade entrada-saída é uma ferramenta para analisar a estabilidade e robustez do controlador, e requer menos suposições do sistema a ser controlado, como, por exemplo, a função de ganho. O sistema é visto com um mapeamento entrada-saída e esta técnica fornece um teste de estabilidade de “entrada limitada/saída limitada” para um controlador estático pré-sintetizado. O resultado final eventual da análise de estabilidade entrada-saída está no teorema do ganho pequeno. De acordo com o teorema do ganho, a condição suficiente para a estabilidade de um sistema em malha fechada é dada por:

$$\|G\| \cdot \|H\| < 1 \quad (\text{A.4})$$

onde  $G$  representa o sistema,  $H$  o controlador e  $\|\cdot\|$  denota a norma. Este critério também é conhecido por *critério de conicidade* e é uma condição suficiente, sendo assim, a não satisfação dessa condição não necessariamente significará que o sistema seja *instável*. O projetista pode adicionar e subtrair um bloco  $C$  no sistema de controle para aumentar a aplicabilidade da condição. Sendo assim, o critério de conicidade é reformulado para:

$$\|G(I + CG)^{-1}\| \cdot \|H - C\| < 1 \quad (\text{A.5})$$

onde o bloco  $C$  deve ser *linear e estático*, e a sua escolha vai garantir a estabilidade.

Esta técnica é bem adequada para sistemas estáveis em malha aberta e problemas de alta dimensionalidade. A principal desvantagem é que as vezes os limites cônicos podem ser muito “cautelosos” e não podem ser usados em perspectiva global.

A partir desta reformulação, é possível analisar o sistema em duas medidas com relação ao centro  $C$ . As duas medidas são: A derivação cônica  $d_H(C)$  de controlador não-linear  $H$  e a robustez cônica  $r_G(C)$  da planta linear  $G$ .

$$\begin{aligned} r_G(C) &= \|G(I + CG)^{-1}\| \\ d_H(C) &= \|H - C\| \end{aligned} \quad (\text{A.6})$$

Se  $C$  é escolhida de tal modo que a malha fechada tende a instabilidade, então o ganho tende a infinito e  $r_G(C)$  converge para zero. O critério de conicidade pode ainda levar a outro dois critérios de estabilidade, são eles: *critério de círculo* e *critério de Popov*.

## APÊNDICE B – REGULADOR LINEAR QUADRÁTICO

O *regulador quadrático linear* (LQR) é um controle por realimentação de estados proposto por Kalman (1960). O LQR é um projeto de controle ótimo para sistemas lineares (Eq. B.2) com índice de performance quadrático (Eq. B.5). Mais precisamente, é gerada uma lei de ganhos de realimentação, que otimiza o índice através da solução da equação diferencial de Riccati (LEWIS, 2010).

O objetivo do LQR é garantir a estabilidade do sistema mantendo-o em torno do ponto de equilíbrio. Neste doutorado, o controle LQR foi aplicado a problemas de rastreamento, portanto, é apresentada aqui a versão capaz de seguir um sinal de referência  $r(t)$ .

Considerando um sistema linear SISO representado pela Equação B.2, a saída do sistema  $y$  dado pela Equação B.3, e um sinal de referência  $r$ , a estratégia de controle é dada pela seguinte equação:

$$u = -Kx + v \quad (\text{B.1})$$

onde  $v$  é um sinal de controle requerido para garantir um rastreamento do sinal de referência  $r$  de modo que o erro no estado estacionário seja a um nível aceitável. Já  $K$  representa o vetor de ganhos de realimentação do estado. Os ganhos de realimentação  $K$  e o sinal  $v(t)$  são determinados pelas equações apresentadas na Tabela 19. O LQR também pode atuar no caso em que  $y$  e  $r$  são vetores, isto é, o caso de rastreamento de múltiplas saídas. A Tabela 19 apresenta as equações necessárias para implementação desse controle ótimo para o problema de rastreamento. Na Figura 33, é apresentado o diagrama de blocos do controle LQR capaz de rastrear o sinal  $r$ .

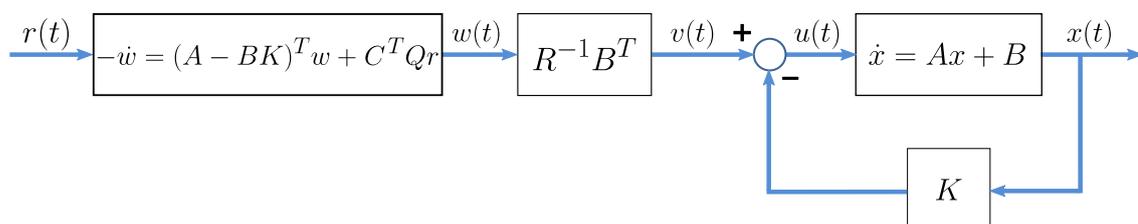


Figura 33 – Diagrama de bloco do controle linear quadrático para rastreamento.

Tabela 19 – Controle Rastreador Linear Quadrático

---



---

*Modelagem do sistema*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad t \geq t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (\text{B.2})$$

*Saída do sistema*

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (\text{B.3})$$

*Erro de rastreamento*

$$\mathbf{e} = \mathbf{r} - \mathbf{y} \quad (\text{B.4})$$

*Índice de desempenho*

$$J(t_0) = \frac{1}{2} \mathbf{e}^T(T) \mathbf{P} \mathbf{e}(T) + \frac{1}{2} \int_{t_0}^T (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (\text{B.5})$$

$$\mathbf{P} \geq 0, \quad \mathbf{Q} \geq 0, \quad \mathbf{R} > 0$$

*Controle ótimo para rastreamento*

*Equação de Riccati*

$$-\dot{\mathbf{S}} = \mathbf{A}^T \mathbf{S} + \mathbf{S} \mathbf{A} - \mathbf{S} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} + \mathbf{C}^T \mathbf{Q} \mathbf{C}, \quad t \leq T, \quad \mathbf{S}(T) = \mathbf{C}^T \mathbf{P} \mathbf{C} \quad (\text{B.6})$$

*Ganho de retroalimentação ótimo*

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} \quad (\text{B.7})$$

*Sistema para frente (Feedforward)*

$$-\dot{\mathbf{w}} = (\mathbf{A} - \mathbf{B}\mathbf{K})^T \mathbf{w} + \mathbf{C}^T \mathbf{Q} \mathbf{r}, \quad t \leq T, \quad \mathbf{w}(T) = \mathbf{C}^T \mathbf{P} \mathbf{r}(T) \quad (\text{B.8})$$

*Controle para frente (Feedforward) mais retroalimentação*

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) + \mathbf{R}^{-1} \mathbf{B}^T \mathbf{w}(t) \quad (\text{B.9})$$


---



---

## APÊNDICE C – DEDUÇÃO DAS REGRAS DE APRENDIZAGEM

Neste apêndice, são apresentadas as derivadas para obter as regras de aprendizagem utilizadas nos procedimentos de treinamento dos métodos para mapeamento contínuo por partes sobre o LARFSOM. Para simplificar os cálculos e a notação neste apêndice, será considerada a aproximação de uma função de saída unidimensional ( $d_{out} = 1$ ). Logo, a função de custo a ser minimizada pode ser expressa por  $E(\boldsymbol{\xi}^{in}) = (1/2)(\xi^{out} - \widetilde{M}_s(\boldsymbol{\xi}^{in}))^2$  para os métodos da diferença e da diferença estendido, e  $E(\boldsymbol{\xi}^{in}) = (1/2)(\xi^{out} - \widetilde{F}_s(\boldsymbol{\xi}^{in}))^2$  para os métodos localmente ponderados. Aplicando o método do gradiente descendente para encontrar a regra de ajustes dos pesos  $\mathbf{k}$ , tem-se:

$$\Delta \mathbf{k} = -\alpha_k \frac{\partial E(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}} = \alpha_k (\xi^{out} - \widetilde{M}_s(\boldsymbol{\xi}^{in})) \frac{\partial \widetilde{M}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}} \quad (\text{C.1})$$

onde  $\alpha_k$  é uma taxa de aprendizagem.

### C.1 DERIVADAS PARA REGRA DE APRENDIZAGEM DO MÉTODO DA DIFERENÇA

Para o vetor de pesos  $\mathbf{k}_s$  do nodo vencedor  $n_s$ :

$$\frac{\partial \widetilde{M}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_s} = \frac{\partial}{\partial \mathbf{k}_s} (w_s^{out} + \mathbf{k}_s \cdot \mathbf{d}_s) = \mathbf{d}_s \quad (\text{C.2})$$

### C.2 DERIVADAS PARA REGRA DE APRENDIZAGEM DO MÉTODO DA DIFERENÇA ESTENDIDO

Para o vetor de pesos  $\mathbf{k}_{s,j}$  do nodo vencedor  $n_s$ :

$$\frac{\partial \widetilde{M}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_{s,j}} = \frac{\partial}{\partial \mathbf{k}_{s,j}} \left( w_s^{out} + \sum_{k=0}^{|N_s|} \mathbf{k}_{s,k} \cdot \mathbf{d}_{s,k} \right) = \sum_{k=0}^{|N_s|} \frac{\partial \mathbf{k}_{s,k}}{\partial \mathbf{k}_{s,j}} \cdot \mathbf{d}_{s,k} \quad (\text{C.3})$$

visto que

$$\frac{\partial \mathbf{k}_{s,k}}{\partial \mathbf{k}_{s,j}} = \begin{cases} \mathbf{I} & k = j \\ \mathbf{0} & k \neq j \end{cases} \quad (\text{C.4})$$

então

$$\frac{\partial \widetilde{M}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_{s,j}} = \mathbf{d}_{s,j} \quad (\text{C.5})$$

### C.3 DERIVADAS PARA REGRA DE APRENDIZAGEM DOS MÉTODOS LOCALMENTE PONDERADOS

Para calcular a derivada  $\frac{\partial \widetilde{F}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i}$ , é preciso determinar  $\frac{\partial \widetilde{M}_{(s,k)}(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i}$

$$\frac{\partial \widetilde{F}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i} = \frac{\partial}{\partial \mathbf{k}_i} \left( \sum_{k=0}^{|N_s|} m_{s,k}(\boldsymbol{\xi}^{in}) \widetilde{M}_{(s,k)}(\boldsymbol{\xi}^{in}) \right) = \sum_{k=0}^{|N_s|} m_{s,k}(\boldsymbol{\xi}^{in}) \frac{\partial \widetilde{M}_{(s,k)}(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i} \quad (\text{C.6})$$

Há dois casos:

1. Caso:  $n_i = n_{(s,k)}$

$$\frac{\partial \tilde{M}_{(s,k)}(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i} = \frac{\partial}{\partial \mathbf{k}_{(s,k),j}} \left( w_{(s,k)}^{out} + \sum_{l=0}^{|N_{(s,k)}|} \mathbf{k}_{(s,k),l} \cdot \mathbf{d}_{(s,k),l} \right) = \sum_{l=0}^{|N_{(s,k)}|} \frac{\partial \mathbf{k}_{(s,k),l}}{\partial \mathbf{k}_{(s,k),j}} \cdot \mathbf{d}_{(s,k),l} \quad (\text{C.7})$$

visto que

$$\frac{\partial \mathbf{k}_{(s,k),l}}{\partial \mathbf{k}_{(s,k),j}} = \begin{cases} \mathbf{I} & l = j \\ \mathbf{0} & l \neq j \end{cases} \quad (\text{C.8})$$

então

$$\frac{\partial \tilde{M}_{(s,k)}(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_{(s,k),j}} = \mathbf{d}_{(s,k),j} \quad (\text{C.9})$$

2. Caso:  $n_i \neq n_{(s,k)}$

$$\frac{\partial \tilde{F}_s(\boldsymbol{\xi}^{in})}{\partial \mathbf{k}_i} = \mathbf{0} \quad (\text{C.10})$$

visto que

$$\frac{\partial \mathbf{k}_{(s,k),l}}{\partial \mathbf{k}_i} = \mathbf{0} \quad (\text{C.11})$$

## APÊNDICE D – MODELAGEM DA DINÂMICA DO PÊNULO INVERTIDO

Neste apêndice, é descrita, em detalhes, a modelagem da dinâmica do sistema pêndulo invertido utilizado nos experimentos do Capítulo 7. Também é mostrado como se deu a linearização desse problema. Por último, são apresentadas as matrizes da modelagem por variáveis de estado.

De maneira simples, o problema do pêndulo invertido consiste em controlar um pêndulo invertido fixado em um carrinho motorizado que pode se movimentar sobre um trilho. O objetivo do controle é conservar o pêndulo no estado invertido através de aplicações de forças no carrinho, mesmo que haja perturbações aplicadas ao sistema ou haja mudanças na posição desejada do carrinho. Um exemplo do mundo real que se relaciona diretamente com este sistema pêndulo invertido é o controle de inclinação do foguete na decolagem (BENSON, 2014). No caso, o objetivo é manter o foguete na posição vertical. Também há analogias com problemas, tais como, controle do dicitlo Segway, controle de um braço robótico, posicionamento de guindastes especiais e controle de dedos robóticos (RIZK; KRUT; DOMBRE, 2007; BABAZADEH; KHIABANI; AZMI, 2016).

Para modelar matematicamente, foi utilizado o esquema de forças apresentado na Figura 34a. Nesse caso, foi considerada uma versão bidimensional do problema. Sendo assim, o pêndulo só poderá se mover no plano  $xy$  mostrado na Figura 34b. Para esse sistema, a entrada de controle é a força  $u$ , que move o carrinho na horizontal, e os resultados são observados através dos valores da posição angular do pêndulo  $\theta$  e da posição horizontal do carrinho  $x$ .

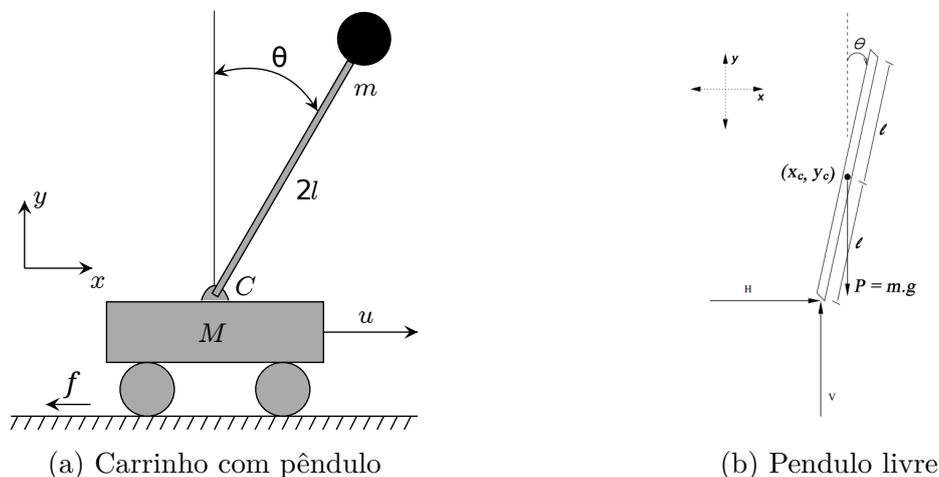


Figura 34 – (a) Diagrama esquemático do sistema mecânico do pêndulo invertido em um carrinho. (b) Diagrama esquemático do corpo livre do pêndulo.

## D.1 ANÁLISE DAS FORÇAS E SISTEMA DE EQUAÇÕES

Analisando a Figura 34b, obtêm-se a seguinte expressão para o centroide do pêndulo (Equações D.1 e D.2):

$$x_c = x + l \operatorname{sen}(\theta) \quad (\text{D.1})$$

$$y_c = l \operatorname{cos}(\theta) \quad (\text{D.2})$$

Somando as forças no diagrama de corpo livre do carro na direção horizontal, pode-se obter a seguinte equação de movimento:

$$M\ddot{x} + b\dot{x} + H = u \quad (\text{D.3})$$

Para encontrar uma expressão para força  $H$ , pode-se fazer a soma das forças no diagrama de corpo livre do pêndulo na direção horizontal.

$$H = m \frac{d^2}{dt^2}(x + l \operatorname{sen}(\theta)) = m\ddot{x} + ml\ddot{\theta} \operatorname{cos}(\theta) - ml\dot{\theta}^2 \operatorname{sen}(\theta) \quad (\text{D.4})$$

Substituindo  $H$  da Equação D.3 pela Equação D.4, pode-se obter a seguinte expressão:

$$(M + m)\ddot{x} + ml\ddot{\theta} \operatorname{cos}(\theta) - ml\dot{\theta}^2 \operatorname{sen}(\theta) + f\dot{x} = u \quad (\text{D.5})$$

Para obter a segunda equação de movimento para este sistema, soma-se as forças verticais no pêndulo. Sendo assim, pode-se obter a seguinte equação:

$$V - mg = m \frac{d^2}{dt^2}(x + l \operatorname{sen}(\theta)) = -ml\ddot{\theta} \operatorname{sen}(\theta) - ml\dot{\theta}^2 \operatorname{cos}(\theta) \quad (\text{D.6})$$

Fazendo a soma dos momentos de inércia no pêndulo, obtêm-se a seguinte expressão:

$$Vl \operatorname{sen}(\theta) - Hl \operatorname{cos}(\theta) = I\ddot{\theta} + C\dot{\theta} \quad (\text{D.7})$$

onde  $I$  é o termo de momento de inércia e é dada pela expressão  $I = \frac{ml^2}{3}$ .

Combinando as Equações D.4, D.6 e D.7 para eliminar os termos  $V$  e  $H$ , obtêm-se a seguinte expressão:

$$ml\ddot{x} \operatorname{cos}(\theta) + (I + ml^2)\ddot{\theta} - mgl \operatorname{sen}(\theta) + C\dot{\theta} = 0 \quad (\text{D.8})$$

As duas equações usadas para modelar o movimento do pêndulo invertido são as Equações D.5 e D.8.

## D.2 LINEARIZAÇÃO

Como as Equações D.5 e D.8 são não-lineares, foi feita uma linearização em torno de  $\theta = 0$ , pois esse sistema apresenta um comportamento próximo ao linear em torno desse ponto. Então, considerando que  $\theta$  tem pequenas variações em torno de zero, pode-se usar as seguintes aproximações:  $\text{sen}(\theta) = \theta$ ,  $\text{cos}(\theta) = 1$ ,  $\theta^2 = 0$ ,  $\dot{\theta}^2 = 0$  e  $\theta\dot{\theta} = 0$ . Sendo assim, as Equações D.5 e D.8 podem ser expressas da seguinte forma:

$$(M + m)\ddot{x} + m\ddot{\theta} + f\dot{x} = u \quad (\text{D.9})$$

$$m\ddot{x} + (I + ml^2)\ddot{\theta} - mgl\theta + C\dot{\theta} = 0 \quad (\text{D.10})$$

## D.3 REPRESENTAÇÃO POR VARIÁVEIS DE ESTADO

As equações linearizadas do movimento descritas anteriormente também podem ser representadas na forma de espaço de estado. Para isso, elas são reorganizadas em uma série de equações diferenciais de primeira ordem. Uma vez que as equações são lineares, elas poderão, então, ser colocadas na forma de matrizes, como descrito na Equação D.11.

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = A \cdot \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + B \cdot u(t) \quad (\text{D.11})$$

Após manipulações algébricas nas Equações D.9 e D.10, os valores obtidos para as matrizes  $A$  e  $B$  foram os seguintes:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-m^2gl^2}{Den} & \frac{-(I+ml^2)b}{Den} & \frac{mlc}{Den} \\ 0 & \frac{mgl(M+m)}{Den} & \frac{mlf}{Den} & \frac{-(M+m)C}{Den} \end{bmatrix} \quad (\text{D.12})$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{I+ml^2}{Den} \\ \frac{-ml}{Den} \end{bmatrix} \quad (\text{D.13})$$

$$Den = I(M + m) + Mml^2 \quad (\text{D.14})$$