

UNIVERSIDADE FEDERAL DA  
PARAÍBA  
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA

## Dissertação de Mestrado



IDENTIFICAÇÃO E PREVISÃO NEURAL DE  
ATUADORES COM MEMÓRIA DE FORMA  
NA PRESENÇA DE DEFORMAÇÃO  
RESIDUAL SIGNIFICATIVA

Lucas Feitosa Nicolau

João Pessoa-PB, Brasil  
Julho – 2018

**LUCAS FEITOSA NICOLAU**

**IDENTIFICAÇÃO E PREVISÃO NEURAL DE ATUADORES COM MEMÓRIA  
DE FORMA NA PRESENÇA DE DEFORMAÇÃO RESIDUAL SIGNIFICATIVA**

**Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.**

**Orientador: Cícero da Rocha Souto**

**JOÃO PESSOA**

**2018**

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

N639i Nicolau, Lucas Feitosa.

IDENTIFICAÇÃO E PREVISÃO NEURAL DE ATUADORES COM  
MEMÓRIA DE FORMA NA PRESENÇA DE DEFORMAÇÃO RESIDUAL  
SIGNIFICATIVA / Lucas Feitosa Nicolau. - João Pessoa,  
2018.

72 f. : il.

Orientação: Cícero da Rocha Souto.  
Dissertação (Mestrado) - UFPB/CEAR.

1. Ligas com Memória de Forma. 2. Redes Neurais  
Artificiais. 3. Redes Neurais Recorrentes. 4.  
Identificação de Sistemas. I. Souto, Cícero da Rocha.  
II. Título.

UFPB/BC

UNIVERSIDADE FEDERAL DA PARAÍBA – UFPB  
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS – CEAR  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGEE

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

IDENTIFICAÇÃO E PREVISÃO NEURAL DE ATUADORES COM MEMÓRIA DE  
FORMA NA PRESENÇA DE DEFORMAÇÃO RESIDUAL SIGNIFICATIVA

Elaborado por

LUCAS FEITOSA NICOLAU

como requisito parcial para obtenção do grau de  
**Mestre em Engenharia Elétrica.**

COMISSÃO EXAMINADORA

  
PROF. DR. CÍCERO DA ROCHA SOUTO (Presidente)

  
PROF. DR. JUAN MOISES MAURICIO VILLANUEVA

  
PROF. DR. ALEXSANDRO JOSÉ VIRGÍNIO DOS SANTOS

João Pessoa/PB, 11 de julho de 2018.

## **AGRADECIMENTOS**

Agradeço a Deus pela minha saúde e pela minha família maravilhosa.

Aos meus pais, Marcos e Roseane, por terem me educado, apoiado e amado. São exemplos de pais e profissionais que eu mejo seguir.

Ao meu irmão e cunhada, Vítor e Raquel, pelas palavras de apoio, pelas risadas e pelas aventuras.

A minha namorada, Érika, por estar ao meu lado nos bons momentos e nos nem tanto.

Agradeço ao meu orientador, Cícero, por não apenas me orientar no mestrado, mas também me guiar na direção certa durante toda a minha vida acadêmica.

A todos da minha família, pelos momentos de felicidade e comunhão.

Aos meus colegas de laboratório do LaSEA, por terem colaborado no desenvolvimento deste trabalho.

Aos meus amigos, que me auxiliaram durante diversas etapas da minha vida.

Ao CNPq e CAPES, pelo suporte financeiro.

## SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES</b> .....	<b>IV</b>
<b>LISTA DE TABELAS</b> .....	<b>V</b>
<b>RESUMO</b> .....	<b>VI</b>
<b>ABSTRACT</b> .....	<b>VII</b>
<b>1 INTRODUÇÃO</b> .....	<b>9</b>
1.1 OBJETIVO GERAL .....	10
1.2 OBJETIVOS ESPECÍFICOS .....	10
1.3 CONTRIBUIÇÕES CIENTÍFICAS .....	11
1.4 ESTRUTURA DO TRABALHO.....	11
<b>2 REVISÃO DE LITERATURA</b> .....	<b>14</b>
2.1 LIGAS COM MEMÓRIA DE FORMA.....	14
2.2 DEFORMAÇÃO IRREVERSÍVEL DAS LMF .....	17
2.3 REDES NEURAIS ARTIFICIAIS .....	18
2.4 ALGORÍTIMO DE LEVENBERG-MARQUARDT COM REGULARIZAÇÃO BAYESIANA.....	21
2.5 MULTIPLAS PREVISÕES COM REDES NEURAIS ARTIFICIAIS .....	24
<b>3 MATERIAL E MÉTODO</b> .....	<b>27</b>
3.1 PROCEDIMENTO EXPERIMENTAL DO ATUADOR LINEAR .....	27
3.2 DESENVOLVIMENTO DA REDE NEURAL .....	29
3.2.1 <b>Preparação dos dados</b> .....	30
3.2.2 <b>Normalização</b> .....	30
3.2.3 <b>Inicialização dos pesos</b> .....	30
3.2.4 <b>Treinamento e teste da rede neural artificial</b> .....	30
3.2.5 <b>Validação</b> .....	32
3.2.6 <b>Desnormalização</b> .....	33
3.2.7 <b>Comparação e avaliação</b> .....	33
3.3 REDE NEURAL DE MÚLTIPLAS PREVISÕES.....	33
<b>4 RESULTADOS</b> .....	<b>36</b>
4.1 RESPOSTA DO TREINAMENTO DA LMF.....	36
4.2 RESPOSTA DA CICLAGEM DO ATUADOR.....	42
4.3 RESPOSTA DA REDE NEURAL DE MÚLTIPLAS SAÍDAS .....	48
<b>5 DISCUSSÃO</b> .....	<b>55</b>
<b>6 CONCLUSÕES</b> .....	<b>58</b>
<b>REFERÊNCIAS</b> .....	<b>59</b>
<b>APÊNDICE 1</b> .....	<b>61</b>
<b>APÊNDICE 2</b> .....	<b>64</b>

## LISTA DE ILUSTRAÇÕES

FIGURA 1 - TRANSFORMAÇÃO DE FASE .....	15
FIGURA 2 - EFEITO DE MEMÓRIA DE FORMA .....	16
FIGURA 3 - DIAGRAMA TEMPERATURA-TENSÃO DAS LMF .....	16
FIGURA 4 - EVOLUÇÃO DA DEFORMAÇÃO DAS LMF .....	17
FIGURA 5 - DIAGRAMA DO NEURÔNIO ARTIFICIAL .....	19
FIGURA 6 - FUNÇÃO TANGENTE HIPERBÓLICA .....	20
FIGURA 7 - ARQUITETURA NEURAL NARX .....	21
FIGURA 8 - REDE NEURAL DE MULTIPLAS PREVISÕES .....	24
FIGURA 9 - ETAPAS DO DESENVOLVIMENTO DA RNA .....	29
FIGURA 10 - EVOLUÇÃO DA DEFOMAÇÃO NO TREINAMENTO DA LMF .....	37
FIGURA 11 - EVOLUÇÃO DA HISTERESE NO TREINAMENTO DA LMF .....	38
FIGURA 12 - VALIDAÇÃO EM FUNÇÃO DO TEMPO NO TREINAMENTO DA LMF .....	39
FIGURA 13 - VALIDAÇÃO DA HISTERESE NO TREINAMENTO DA LMF .....	40
FIGURA 14 - MÁXIMOS E MÍNIMOS DA DEFORMAÇÃO DURANTE A CICLAGEM .....	43
FIGURA 15 - RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 50MPA .....	44
FIGURA 16 – RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 100MPA .....	44
FIGURA 17 - RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 150MPA .....	45
FIGURA 18 - MÁXIMOS E MÍNIMOS DA DEFORMAÇÃO PARA VALIDAÇÃO .....	46
FIGURA 19 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 50MPA .....	47
FIGURA 20 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 100MPA .....	47
FIGURA 21 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 150MPA .....	48
FIGURA 22 - TREINAMENTO DA RNA PREDITIVA DE 4 SAÍDAS .....	49
FIGURA 23 - TREINAMENTO DA RNA PREDITIVA DE 6 SAÍDAS .....	50
FIGURA 24 - VALIDAÇÃO DA RNA PREDITIVA DE 4 SAÍDAS .....	51
FIGURA 25 - VALIDAÇÃO DA RNA PREDITIVA DE 6 SAÍDAS .....	52

## LISTA DE TABELAS

TABELA 1 – TOPOLOGIA DAS REDES DE MÚLTIPLAS PREVISÕES .....	34
TABELA 2 - DESEMPENHO DA RNA PARA O TREINAMENTO DA LMF .....	41
TABELA 3 - DESEMPENHO DA RNA PARA CICLAGEM DA LMF .....	48
TABELA 4 - DESEMPENHO DA RNA DE MÚLTIPLAS PREVISÕES .....	53

## RESUMO

Este trabalho visa o desenvolvimento de redes neurais artificiais para a identificação do comportamento de ligas com memória de forma aplicada como atuadores termomecânicos, apresentando deformação irreversível considerável. O acúmulo da deformação residual durante a transformação de fase dos atuadores com memória de forma prejudica a estabilidade dimensional e modifica o seu comportamento histerético, fato este que dificulta o projeto e controle destes materiais. Para o aprendizado do modelo neural, fios de liga Níquel-Titânio são expostos a duas condições distintas, o treinamento do material com memória de forma através de vinte e cinco ciclos de transformação de fase e a ciclagem térmica subsequente ao treino para mil e quinhentos ciclos. Os dados de temperatura e deformação destas duas situações são utilizados para identificação e para validação dos dois modelos implementados, de previsão a um passo à frente e de múltiplas previsões da evolução da deformação irreversível. A partir da comparação entre os dados experimentais e os estimados é observado que a rede neural a um passo à frente caracteriza adequadamente o alongamento do atuador, a diminuição da máxima deformação de transformação e a degradação da histerese do atuador. Ademais, apesar do desempenho da arquitetura neural de múltiplas saídas exibir erro absoluto relativo médio próximo de 3% para as condições estudadas, é observado uma menor precisão na identificação do material com memória de forma, principalmente nos valores de máximo e mínimo da deformação em cada ciclo.

**Palavras-chave:** Ligas com Memória de Forma, Redes Neurais Artificiais, Redes Neurais Recorrentes, Identificação de Sistemas.

## ABSTRACT

This work aims to develop of artificial neural networks to identify the behavior of shape memory alloys applied as thermomechanical actuators, presenting considerable irreversible deformation. The accumulation of the residual deformation during the phase transformation of the shape memory actuators degrades the dimensional stability and modifies its hysteretic behavior, which difficults the design and control of these materials. For the learning of the neural model, nickel-titanium wires are exposed to two distinct conditions, training the shape memory material through twenty-five phase transformation cycles and the subsequent thermal cycling for one thousand five hundred cycles. The data of temperature and deformation of these two situations are used for identification and for validation of the two implemented models, the one-step-ahead prediction and the multiple predictions of the evolution of the irreversible deformation. From the comparison between the experimental data and the estimates value, it is observed that the one-step-ahead neural network adequately characterizes the elongation of the actuator, the decrease of the maximum transformation strain and the degradation of the actuator hysteresis. Further, although the performance of the multiple output neural architecture exhibits an average relative absolute error close to 3% for the studied conditions, a smaller accuracy in the identification of the shape memory material is observed, especially in the maximum and minimum values of the deformation in each cycle.

**Keywords:** Shape Memory Alloys, Artificial Neural Networks, Recurrent Neural Networks, Systems Identification.

# 1 INTRODUÇÃO

## 1 INTRODUÇÃO

O desenvolvimento de sistemas mais compactos e eficientes pode ser concebido por meio da utilização de materiais e estruturas inteligentes. Um exemplo é a utilização de atuadores termomecânicos que empregam Ligas com Memória de Forma (LMF), de composição Ni-Ti, permitindo que atuadores lineares convencionais sejam substituídos por atuadores com menor peso e tamanho (MOHD JANI et al., 2014). Este material ativo tem a capacidade de recuperar elevadas deformações através da transformação de fase sólida entre a austenita e a martensita. Esta transformação pode ser induzida por tensão ou por temperatura e, conseqüentemente, apresenta os fenômenos de pseudoelasticidade e efeito de memória de forma (EMF), respectivamente.

Na pseudoelasticidade, a transformação das fases é provocada por uma força externa, normalmente a temperatura constante. Enquanto que, no efeito de memória de forma, a mudança de martensita para austenita é dada pela variação da temperatura, a tensão invariável. Ambos os processos são praticamente reversíveis, ou seja, grande parte da deformação imposta pode ser recuperada em um ciclo de transformação de fase, possibilitando que este material seja utilizado como atuadores termomecânicos de considerável força por volume do material (KUMAR; LAGOUDAS, 2008).

O elevado número de ciclos em que os atuadores com ligas de memória de forma estão submetidos proporciona um acúmulo de imperfeições que prejudica o seu desempenho. Segundo Gao (2016), um das questões críticas que dificulta o emprego das LMF é a fadiga funcional durante a atuação repetida de ciclos térmicos ou mecânicos. Estas imperfeições na transformação de fase provocam o aparecimento de deformações não reversíveis e afetam o comportamento histerético, a máxima deformação de transformação e a estabilidade dimensional destes termoatuadores (KUMAR; LAGOUDAS, 2008).

Com a finalidade de reduzir os efeitos da deformação plástica irreversível, o atuador termomecânico é submetido a um procedimento de treinamento o qual consiste na acumulação significativa de defeitos cristalinos que aumentam o estresse interno residual, fato que inibe o aumento de mais defeitos e, portanto, estabiliza a resposta da liga. Este treinamento também é responsável pela ativação do efeito de

memória de forma duplo que se resume a presença da memorização de forma em ambos as fases, de baixa e alta temperaturas (LUO; ABEL, 2007). No entanto, a estabilização da deformação residual e a presença da memória de forma de dois sentidos ocorre em condições específicas e para um número limitado de ciclos de transformação, o que dificulta o desenvolvimento e controle dos atuadores com LMF.

Para modelar a degradação da resposta do atuador, principalmente em duas condições críticas, no treinamento quando o acúmulo de deformação residual é acentuado e na ciclagem que uma grande quantidade de ciclos gera alterações graduais, faz-se necessário a utilização de Redes Neurais Artificiais (RNA). Estas redes que, pelo teorema de Kolmogorov-Nielsen, possuem a capacidade de aproximar qualquer função arbitrária, não constante e limitada, sendo apropriadas para tarefas de identificação e previsão (FERRONATO, 2008). Além do mais, materiais que apresentam comportamento não linear como por exemplo as ligas com memória de forma e os materiais piezoelétricos são modelados com relativa simplicidade através de redes neurais recorrentes (WANG; SONG, 2014).

## 1.1 OBJETIVO GERAL

O propósito geral desta pesquisa é desenvolver um modelo computacional baseado em redes neurais recorrentes para atuadores com memória de forma apresentando crescimento acentuado da deformação irreversível, de modo a não apenas aprender as modificações que ocorrem no material, mas também prever adequadamente a evolução da histerese e perda da estabilidade dimensional.

## 1.2 OBJETIVOS ESPECÍFICOS

Para analisar a capacidade de identificação e previsão das redes neurais artificiais nos atuadores termomecânicos, este trabalho apresenta três casos distintos, previsão a um passo à frente durante a etapa de treinamento, previsão um passo à frente ao longo da ciclagem, e múltiplas previsões para os dados do treinamento do

atuador com memória de forma. Portanto, todas as etapas que definem estes objetivos específicos são:

- Implementar uma rede neural artificial no software MATLAB;
- Realizar a identificação do treinamento e da ciclagem da LMF por meio da rede neural implementada;
- Desenvolver uma RNA de múltiplas previsões e realizar a identificação da LMF;
- Discutir os resultados obtidos dos diferentes modelos propostos e comparar com os dados experimentais do atuador.

### 1.3 CONTRIBUIÇÕES CIENTÍFICAS

Diferentemente da maioria dos trabalhos envolvendo atuadores com memória de forma que considera a aplicação da LMF estabilizada, este trabalho visa a implementação de uma rede neural não linear auto regressiva de entrada exógena, conhecida como *Nonlinear Autoregressive network with exogenous inputs* (NARX), para aprender e prever, um passo a frente e até múltiplos passos a frente, os efeitos do acúmulo da deformação residual. Uma vez que este fenômeno impõe um desafio significativo na durabilidade e confiabilidade das LMF, principalmente no processo de treinamento e funcionalidade a longo prazo (NESPOLI et al., 2010).

Diferentes condições de carregamento externo são analisadas devido ao acúmulo da deformação residual ser proporcional ao carregamento externo exercido no atuador, de maneira a observar a habilidade do sistema neural em acompanhar essas modificações.

### 1.4 ESTRUTURA DO TRABALHO

O presente trabalho está dividido em cinco capítulos. Neste primeiro capítulo de introdução é exposto as motivações, os objetivos e a contribuição científica desta pesquisa. No segundo apresenta-se a fundamentação teórica, destacando-se as

principais informações e referências sobre as ligas com memória de forma, a deformação plástica residual e as redes neurais artificiais. No capítulo seguinte, trata-se dos materiais e métodos usados, destacando o experimento para aquisição dos dados do atuador termomecânico e a implementação da rede neural em MATLAB. O quarto capítulo representa os resultados obtidos com as simulações e as suas discussões. Por último, as conclusões finais e proposta para trabalhos futuros.

## **2 REVISÃO DE LITERATURA**

## 2 REVISÃO DE LITERATURA

Neste capítulo é abordado de forma detalhada as principais características das ligas com memória de forma, os efeitos da deformação irreversível nos atuadores destes materiais e a utilização de redes neurais artificiais para a identificação e previsão de sistemas não lineares.

### 2.1 LIGAS COM MEMÓRIA DE FORMA

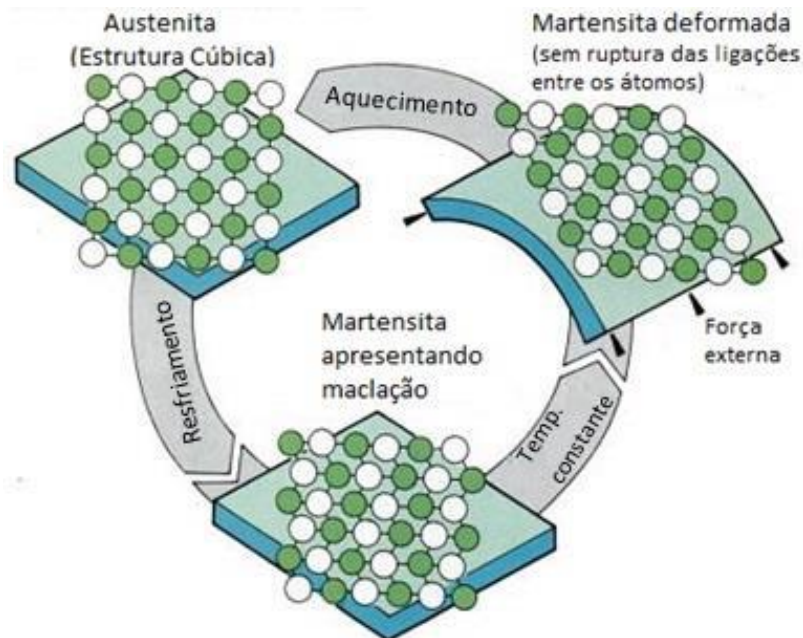
A capacidade de recuperação de forma da liga níquel-titânio foi observada por Buehler e Wiley, em 1962, enquanto estudavam a transformação da fase martensita dessa liga (OLIVEIRA, 2013). A partir desta descoberta as ligas metálicas que apresentam esse comportamento foram nomeadas de ligas com memória de forma. Além desta capacidade de recuperação de elevadas deformações por aquecimento também foi identificado o fenômeno da superelasticidade, que confere resistência a elevadas tensões mecânicas, que não se restringiam as ligas níquel-titânio mas se abrangiam a diversas outras composições de ligas (AURICCHIO; TAYLOR, 1997).

O efeito de memória de forma e a superelasticidade são fenômenos que ocorrem devido à transformação de fase sólida-sólida nas LMF. No efeito de memória de forma a transformação das fases austenita e martensita é induzida por temperatura. A liga é submetida à deformação plástica em baixas temperaturas, fazendo com que a martensita maclada se torne não maclada, ou seja, ocorre uma orientação dos grãos do material (GARCIA, 2015). Em seguida, o material é aquecido a uma determinada temperatura em que sucede a mudança de fase para a austenita e, conseqüentemente, a recuperação da forma memorizada. Na Figura 1, observa-se este fenômeno de transformação de fase envolvendo também o processo de orientação ocorrido na martensita.

A superelasticidade é a transformação de fase induzida por tensão mecânica. Inicialmente, o material se apresenta no estado de austenita e, em seguida, é carregado mecanicamente provocando a transformação para a martensita, que é uma fase menos rígida que a primeira (ZHOU, 2012). Em seguida, a força externa aplicada é removida e a liga retorna ao anterior estado de austenita. Esta mudança de fase

permite que o material seja consideravelmente deformado sem que ocorra deformação plástica irreversível significativa.

FIGURA 1 - TRANSFORMAÇÃO DE FASE



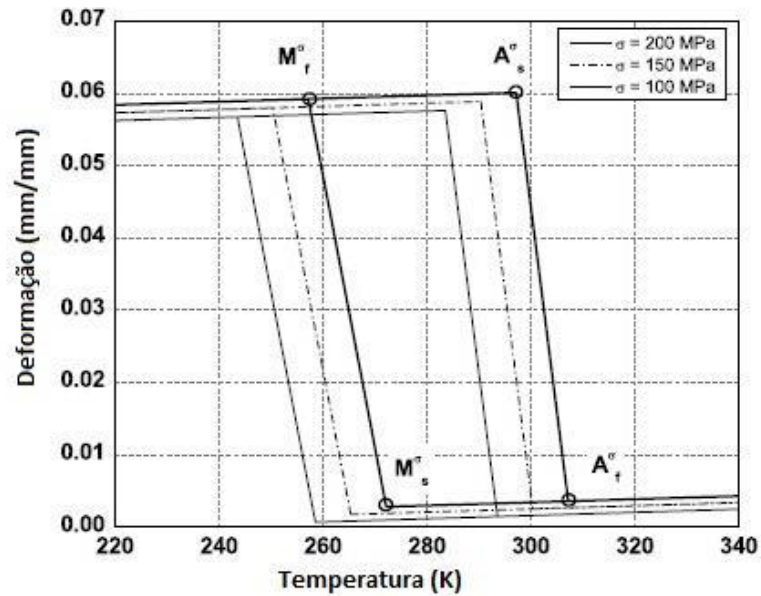
Fonte: <http://engenheirodemateriais.com.br/mecanismo-memoria-de-forma/>

Em ambos os fenômenos, o efeito de memória de forma e a superelasticidade, os materiais com memória de forma apresentam histerese, que representa a energia dissipada no ciclo de transformação de fase (KUMAR; LAGOUDAS, 2008). Utilizando o EMF como exemplo, é possível construir a curva de histerese da liga com memória de forma a partir da variação da deformação em função da temperatura, comportamento observado na Figura 2.

A partir do fenômeno de recuperação de forma, as LMF são utilizadas como atuadores termomecânicos lineares ou rotativos além de serem aplicados em diversas áreas como, por exemplo, na odontologia, na engenharia automotiva e aeronáutica (MOHD JANI; LEARY; SUBIC, 2017). Entretanto, estes atuadores são significativamente influenciados pela tensão mecânica em que estão submetidos, devido ao aumento da temperatura de transformação de fase ser proporcional ao crescimento do carregamento externo (BASÍLIO SOBRINHO et al., 2017). Esta relação pode ser visualizada na Figura 3, em que as temperaturas de início e fim da

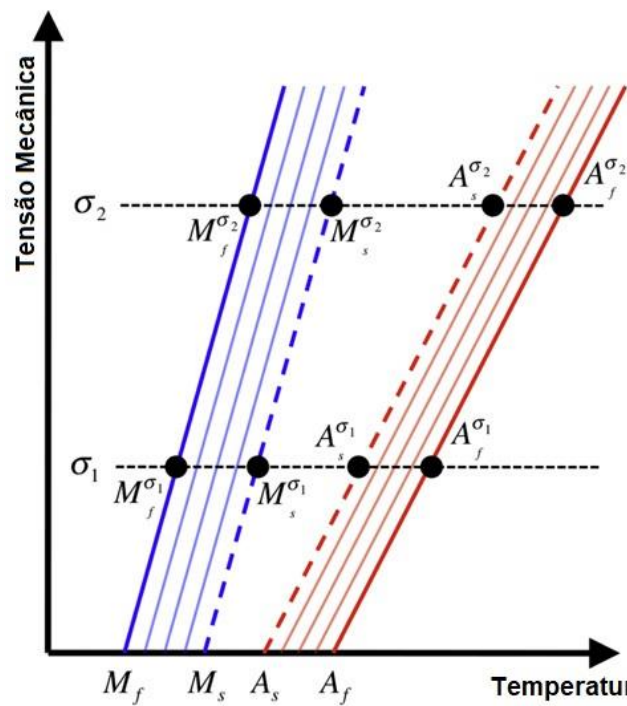
transformação de martensita e austenita são elevadas com a tensão aplicada ao atuador.

FIGURA 2 - EFEITO DE MEMÓRIA DE FORMA



Fonte: (KUMAR; LAGOUDAS, 2008) adaptado

FIGURA 3 - DIAGRAMA TEMPERATURA-TENSÃO DAS LMF

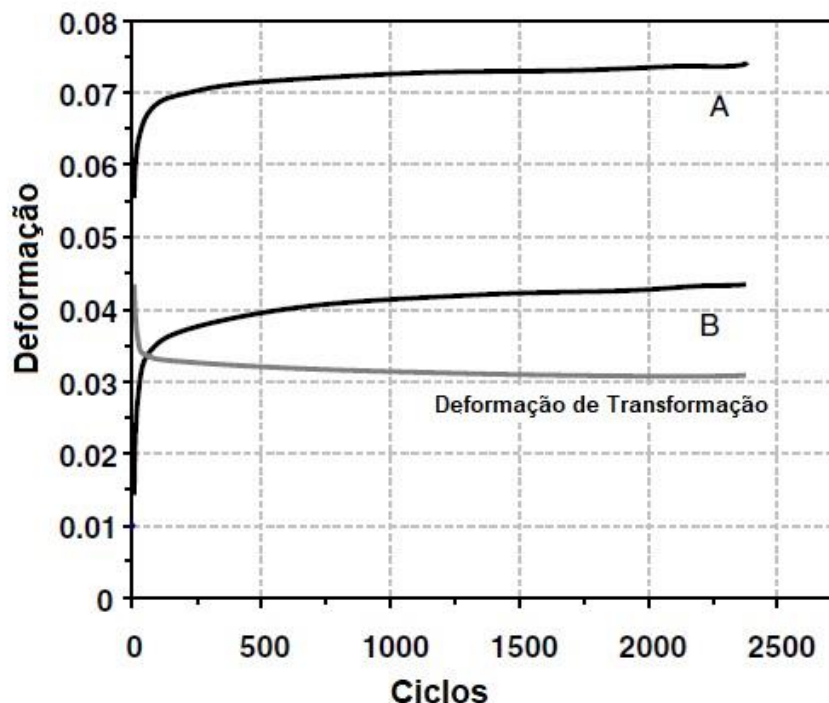


Fonte: (KUMAR; LAGOUDAS, 2008) adaptado

## 2.2 DEFORMAÇÃO IRREVERSÍVEL DAS LMF

No funcionamento dos atuadores com memória de forma, o material é sujeito a vários ciclos de transformação de fase que favorece o acúmulo da deformação residual. A deformação residual ou deformação plástica irreversível é formada pelo crescimento de defeitos cristalinos nas ligas com memória de forma e prejudica o desempenho deste material por conta da perda de estabilidade dimensional e alteração na curva histerética e na máxima deformação de transformação (BENAFAN et al., 2014). Esta evolução do alongamento do atuador como as mudanças dos valores mínimos, B, e máximos, A, de deformação são identificados na Figura 4. Além do mais, a fadiga funcional, produzida pelos defeitos durante a ciclagem termomecânica, influencia significativamente a longevidade e a aplicabilidade da LMF, além de formas efetivas de caracterizar e controlar este fenômeno ainda estão em desenvolvimento (SCIRÈ MAMMANO; DRAGONI, 2014).

FIGURA 4 - EVOLUÇÃO DA DEFORMAÇÃO DAS LMF



Fonte: (KUMAR; LAGOUDAS, 2008) adaptado

Com a finalidade de reduzir os efeitos da deformação residual e que proporciona maior estabilidade dimensional às ligas com memória de forma são submetidas a um treinamento termomecânico. Este processo de treinamento também é conhecido como fenômeno de ativação do efeito de memória de forma duplo, ou seja, o material é sujeito a alguns ciclos térmicos, com tensão mecânica constante, para memorizar a forma em baixas temperatura e em altas temperaturas (LUO; ABEL, 2007). Visto que o acúmulo de deformação residual é acentuado durante os primeiros ciclos de funcionamento do atuador de LMF, o treinamento provoca um elevado acúmulo de defeito cristalinos, permitindo que o material seja utilizado durante vários ciclos com maior estabilidade em relação as suas propriedades.

Apesar da ativação do efeito de memória de forma duplo facilitar o desenvolvimento e controle destes atuadores, o crescimento da deformação plástica irreversível é contínuo e significativo à determinação da vida útil destes dispositivos. De acordo com Gao (2017), a origem física da formação da deformação residual não está completamente esclarecida e, conseqüentemente, maneiras efetivas de desenvolver e gerir os atuadores com memória de forma são necessárias.

### 2.3 REDES NEURAIS ARTIFICIAIS

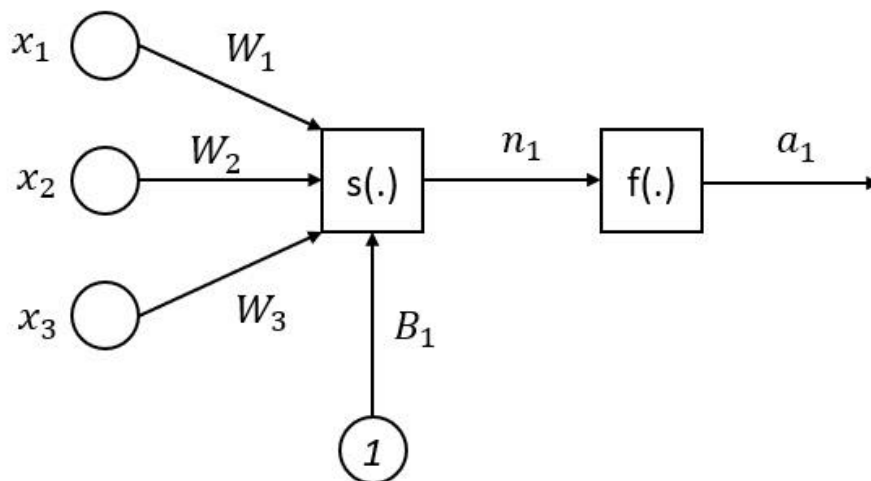
Incentivados pela capacidade de adaptação e aprendizado das redes neurais biológicas, modelos matemáticos de neurônios artificiais foram desenvolvidos de modo a emular a ação de um neurônio biológico e conseqüentemente reconhecer e prever diversos padrões diferentes. Segundo Wang (2014), comportamentos não lineares como a histerese de ligas com memória de forma e materiais piezoelétricos podem ser modelados com relativa simplicidade através de redes neurais recorrentes. Além do mais, redes neurais NARX, por possuírem uma arquitetura neural dinâmica e recorrente, podem ser utilizadas tanto para previsões à um passo à frente como também para previsões de longo prazo (CHATTERJEE et al., 2012).

Os neurônios artificiais são compostos por uma função de soma e uma função de ativação, e seu diagrama pode ser visualizado na Figura 5. O modelo que descrevem estas duas funções é descrito nas equações (1) e (2).

$$n_i = \sum_{i=1}^{N_i} W_i x_i + B_1 \quad (1)$$

$$a_i = f(n_i) \quad (2)$$

FIGURA 5 - DIAGRAMA DO NEURÔNIO ARTIFICIAL



Fonte: Autor

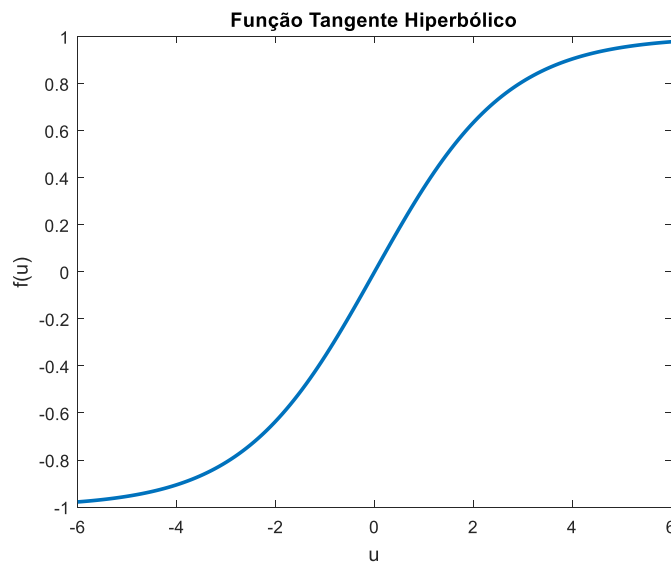
Onde  $W_i$ ,  $x_i$  e  $B_1$  representam os pesos, as entradas e o bias da camada intermediária.  $N_i$  é definido como o número de neurônios da camada intermediária e  $a_i$  é a saída do respectivo neurônio. Para facilitar o aprendizado de comportamentos não lineares, é empregado uma função de ativação tangente hiperbólica na camada intermediária apresentada na equação (3). Além do mais, a representação gráfica dessa função de ativação pode ser observada na Figura 6.

$$F(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}} \quad (3)$$

Na camada de saída o mesmo modelo de neurônio artificial é utilizando, a única diferença é a aplicação da função de ativação linear descrita pela equação (4).

$$L(n) = n \quad (4)$$

FIGURA 6 - FUNÇÃO TANGENTE HIPERBÓLICA



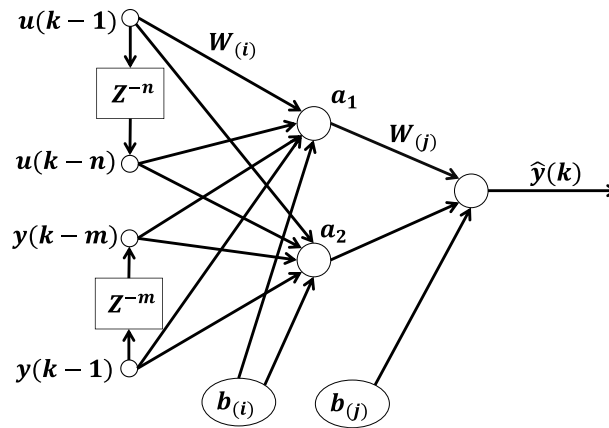
Fonte: Autor

A arquitetura da rede neural não linear auto regressiva com entrada externa é constituída de um sistema dinâmico, que emprega valores passados das series temporais da entrada exógena e da realimentação da saída, para caracterizar as mudanças que ocorrem em função do tempo como o comportamento histerético de materiais com memória de forma (WANG; SONG, 2014). A arquitetura desta rede NARX é identificada pela equação 5 e pode ser visualizada na Figura 7, em que sua entrada externa é representada por  $u$ , as respostas anteriores do atuador como  $y$ , e a sua saída estimada por  $\hat{y}$ .

$$\hat{y}(k) = F(u(k-1), \dots, u(k-n), y(k-1), \dots, y(k-m)) \quad (5)$$

Além destes aspectos topológicos da rede neural como a função de ativação, o número de neurônios e o número de entradas afetam consideravelmente o desempenho do sistema neural. Por exemplo, com o aumento destas variáveis o número de pesos cresce proporcionalmente, permitindo uma maior capacidade de aprendizado e previsão de comportamentos não lineares. Todavia, a carga computacional necessária para o treinamento é influenciada por estas escolhas topológicas e um equilíbrio entre estes aspectos deve ser almejado (KRIESEL, 2005).

FIGURA 7 - ARQUITETURA NEURAL NARX



Fonte: Autor

#### 2.4 ALGORÍTIMO DE LEVENBERG-MARQUARDT COM REGULARIZAÇÃO BAYESIANA

O treinamento das RNAs é aplicado para minimizar uma função de custo estabelecida e conseqüentemente aproximar a resposta estimada pelo sistema neural aos dados experimentais coletados. O algoritmo de Levenberg Marquardt com regularização Bayesiana é definido para reduzir o número máximo de épocas de treinamento, aprimorar a capacidade de generalização e diminuir o sobreajuste (sobretraining), ou seja, evitar que ocorra o excesso de treinamento e por conseqüência a perda da habilidade de previsão do sistema neural (HAGAN; DEMUTH; BEALE, 1995). Além do mais, as principais vantagens em utilizar a regularização Bayesiana são a robustez conferida ao modelo e a validação cruzada ser desnecessária (WALKER et al., 2008). A função de custo é composta pelo somatório do erro quadrático,  $E_D$ , como também o somatório do quadrado dos pesos da rede neural,  $E_W$ , apresentada na equação 6.

$$E(W) = \beta \sum_{k=1}^S \sum_{i=1}^N (d_{ki} - y_{ki})^2 + \alpha \sum_j^{N_W} W_j^2 = \beta E_D + \alpha E_W \quad (6)$$

Onde  $d_i$  representa o valor da resposta desejada obtida experimentalmente, e  $S, N$  e  $N_W$  simbolizam, respectivamente, o número de saídas, o número de padrões de treinamento e o número de pesos da rede neural. Os hiperparâmetros  $\alpha$  e  $\beta$  são inseridos para minimizar o erro da rede neural como também reduzir o valor dos pesos de modo a atenuar a resposta da rede neural (KAYRI, 2016). A partir da função de custo é determinada a matriz hessiana do algoritmo de Levenberg-Marquardt, que é apresentada na equação 7 (KOKOL et al., 2000).

$$H = \nabla \nabla E + \mu I = 2\beta J^T J + 2\alpha \mu I \quad (7)$$

Onde  $\nabla$  representa o gradiente da função de custo,  $I$  é a matriz identidade e  $J$  é a matriz Jacobiana do erro, ou seja, a derivada do erro em função de todos os pesos da rede neural. A derivação do divergente da função de custo,  $\nabla E$ , como também a equação detalhada da matriz Jacobiana podem ser visualizadas no Apêndice I. No intuito de encontrar os valores dos hiperparâmetros  $\alpha$  e  $\beta$  que otimizam o treinamento é necessário que os mesmos sejam atualizados pelo método de David Mackay através do número efetivo de parâmetros que é definido pela equação 8 (WALKER et al., 2008).

$$\gamma = N_W - 2\alpha \text{tr}(H^{-1}) \quad (8)$$

Em seguida, as equações 9, descrevem a equação de atualização dos hiperparâmetros em função do número efetivo de parâmetros.

$$\alpha = \frac{\gamma}{2E_W} \text{ e } \beta = \frac{N - \gamma}{2E_D} \quad (9)$$

Onde  $N$  é o número de padrões utilizados para o treinamento da rede neural. Finalmente, os passos empregados durante o algoritmo de treinamento são:

1. Inicializa os parâmetros com seus respectivos valores iniciais,  $\alpha = 0.01$ ,  $\beta = 1$  e  $\mu = 10$ , além dos pesos da rede neural de maneira aleatória. É iniciada a contagem de épocas de treinamento.

2. Realiza o processo de propagação das informações de entrada e calcula a resposta da rede neural e a matriz Jacobiana.
3. Compara o erro quadrático médio admissível com o valor estimado. Se o erro obtido for menor que o erro admissível, o treinamento é encerrado.
4. Se o erro for maior que o admissível, os pesos são recalculados a partir da equação:

$$W_i(k + 1) = W_i(k) - (H)^{-1}\nabla E \quad (10)$$

5. Em seguida, o processo de propagação é repetido e o erro anterior é comparado com o erro posterior. Se o erro subsequente for maior, o valor de  $\mu$  é incrementado e a época de treinamento é aumentada, retornando ao passo 2.
6. Se o erro posterior for menor, o valor de  $\mu$  é diminuído. Posteriormente, o número efetivo de parâmetros e os hiperparâmetros são atualizados, e os pesos recalculados são substituídos pelos iniciais da época de treinamento. Por fim, o treinamento retorna ao segundo passo, incrementando o valor da época de treinamento.

Paralelamente ao estágio de treinamento, uma parte dos dados é usado para testar a rede neural, de modo a examinar a capacidade de generalização do modelo. Os dados são divididos aleatoriamente em dois conjuntos, sem modificação de suas ordens, dos quais setenta e cinco por cento é empregado para treinamento enquanto vinte e cinco por cento é utilizado para testes. Como o processo de aprendizado não ocorre com os dados separados para o teste, é esperado um erro maior em relação ao treinamento. Contudo, se durante o processo de treino da rede neural ocorrer uma divergência significativa entre o erro de teste e o erro de treinamento, o mesmo deve ser encerrado, caracterizando o fenômeno de parada antecipada do treinamento (HAGAN; DEMUTH; BEALE, 1995). Este evento, também conhecido como *Early stopping*, evita a ocorrência do sobre-treinamento, aprimorando a capacidade de generalização da rede neural.

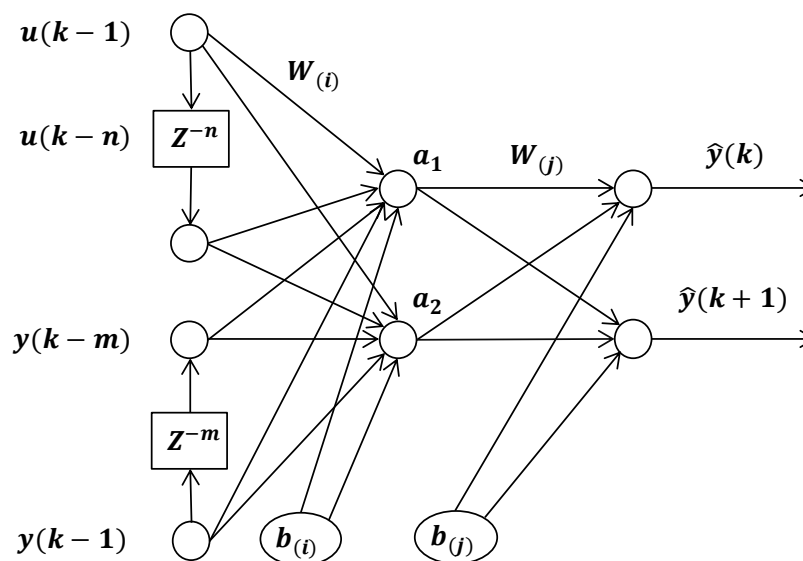
Durante cada época de treinamento, os valores estimados para os conjuntos de treinamento e de teste são comparados com os valores admissíveis definidos, a fim de completar a fase de aprendizagem da rede neural artificial. O número de atrasos

empregados nas entradas e o número de neurônios são encontrados por um processo de tentativa e erro, verificando a configuração que apresenta rápida convergência para os valores admissíveis do erro quadrático médio do treinamento e teste.

## 2.5 MULTIPLAS PREVISÕES COM REDES NEURAIS ARTIFICIAIS

O modelo NARX apresentado na Figura 7 é caracterizado por uma forma de circuito aberto, também conhecido como *open-loop*, por ser um modelo preditivo de um passo à frente que utiliza informações anteriores para estimar uma resposta futura do sistema. As redes NARX também podem apresentar uma forma de circuito fechado, chamado de *closed-loop*, que consiste em realimentar os valores de saída estimados na entrada do sistema neural para previsão de futuras respostas. De acordo com Chatterjee (2011), estes sistemas de malha fechada tentam emular recursivamente o comportamento dinâmico do sistema que gerou as séries temporais não lineares e que, em caso de um horizonte de previsão tendendo ao infinito, a resposta prevista será gerada utilizando estimativas, acumulando o erro de previsão em cada laço. Estas redes *closed-loop* apresentam maior complexidade em relação ao modelo NARX *open-loop*, principalmente quanto ao treinamento empregado, e limitações quando ao horizonte de previsão escolhido.

FIGURA 8 - REDE NEURAL DE MULTIPLAS PREVISÕES



Fonte: Autor

Outra abordagem para realizar múltiplas previsões são redes neurais compostas por várias saídas, semelhantes aos sistemas *Multiple-Input and Multiple-Output* (MIMO), em que cada resposta estimada consiste em uma previsão a um passo à frente da resposta anterior. A arquitetura desta rede pode ser visualizada na Figura 8. Estas redes surgiram pela necessidade de valores futuros da resposta de determinados processos de engenharia, sem que medições no horizonte de interesse fossem realizadas (PARLOS; RAIS; ATIYA, 2000). Em outras palavras, aproximações das respostas futuras do sistema poderiam ser obtidas necessitando somente de informações passadas. Estas redes podem melhorar o aprendizado e antecipação de comportamentos complexos, apesar de que a capacidade de generalização da rede é prejudicada (CHTOUROU; CHTOUROU; HAMMAMI, 2008). Além do mais, comparando esta metodologia com as redes NARX de malha fechada, é observável uma maior carga computacional para uma menor deterioração da precisão de estimação (BAO; XIONG; HU, 2014).

### **3 MATERIAL E MÉTODO**

### 3 MATERIAL E MÉTODO

Nesta seção é discutido o procedimento experimental para a obtenção dos dados do atuador linear de liga com memória de forma. Posteriormente, as etapas de identificação do treinamento e da ciclagem termomecânica, e, por fim, o processo de validação do modelo neural para previsão à um passo a frente e para estimativa de várias saídas posteriores do comportamento do atuador.

#### 3.1 PROCEDIMENTO EXPERIMENTAL DO ATUADOR LINEAR

Os dados experimentais utilizados para treinar e validar a rede neural artificial foram extraídos de uma pesquisa realizada em fios de níquel-titânio para analisar o comportamento de estabilidade de atuadores de memória de forma sob diferentes condições termomecânicas (LIMA, 2017). Este estudo foi realizado envolvendo fios de LMF com 0,305 mm de espessura e com uma composição nominal de 52,95% Ni e 47,05% Ti (porcentagem de massa) fornecidos pela empresa Memory Metalle. Antes dos ciclos de transformação de fase, os fios foram recozidos por 60 min a 450 °C e resfriados por têmpera em água até a temperatura ambiente (27 °C), que resultou na seguinte temperaturas de transformação:  $A_S = 88,4^\circ\text{C}$ ,  $A_F = 99,7^\circ\text{C}$ ,  $M_S = 63,7^\circ\text{C}$ , and  $M_F = 54,4^\circ\text{C}$ .

Na realização do experimento, as três amostras dos fios atuadores de LMF, com comprimento ativo de 140 mm e sem deformação residual prévia, foram submetidos inicialmente a 25 ciclos térmicos sob um carregamento mecânico de 100 MPa, 200 MPa e 300MPa, respectivamente. Esta ciclagem térmica inicial aplicada ao material sem deformação residual prévia é conhecida como treinamento termomecânico. Posteriormente, mais três amostras de ligas com memória de forma foram sujeitas ao treinamento de 200 MPa para que, em seguida, fossem aplicados mais 1500 ciclos térmicos sob um carregamento mecânico de 50 MPa, 100 MPa e 150MPa, respectivamente, chamado de ciclagem termomecânica.

A liga de memória de forma foi posicionada em um guia mecânico linear e deformada pela força gravitacional através de pesos. Além disso, um sistema de aquisição de dados NI cRIO-9076 conectado a um circuito de acionamento controlava

o aquecimento da corrente elétrica e a contração do atuador. Os dados de temperatura são medidos por um termopar tipo K conectado ao SMA enquanto a informação de deformação é calculada a partir da deformação medida por um Transformador Diferencial de Variável Linear (LVDT).

O aquecimento do atuador no processo de treinamento e ciclagem foi dado a partir do efeito Joule, ou seja, a temperatura da liga com memória de forma foi elevada através da passagem de corrente elétrica no material. A corrente elétrica alternada, do tipo onda quadrada de frequência constante de 0,0333 Hz, aqueceu a liga acima da temperatura inicial da austenita e permitiu que o ar circundante resfriasse o material abaixo da temperatura inicial da martensita. A frequência para o caso do treinamento de 100 MPa, foi de 0,025 Hz, para permitir um maior tempo de aquecimento do material e a transformação completa entre as fases.

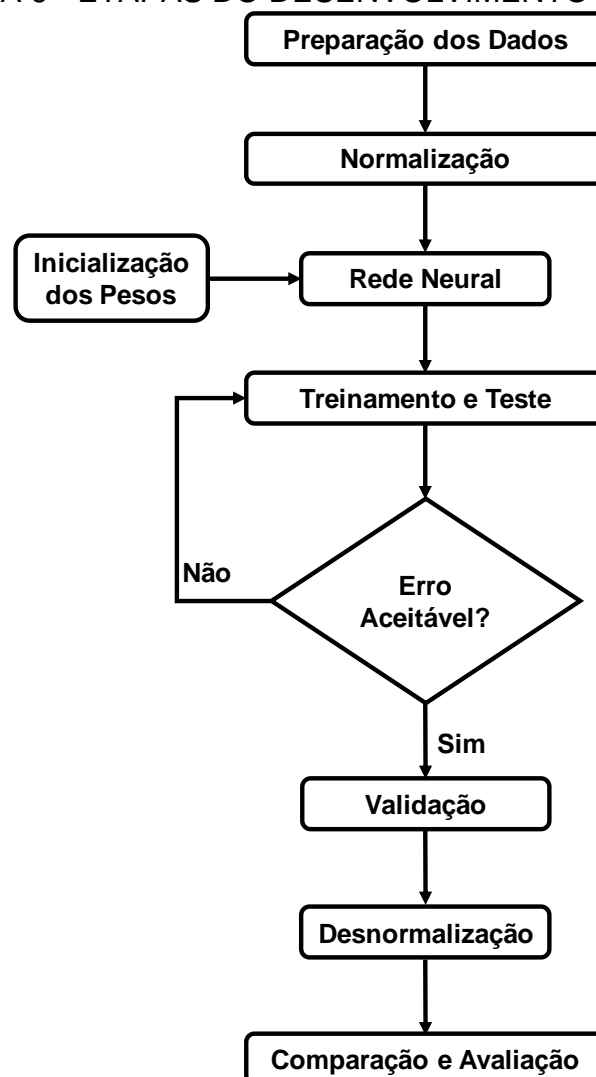
A amplitude e o ciclo de trabalho foram selecionados para gerar a contração máxima do atuador e executar um ciclo de transformação completo com a máxima tensão de transformação. Portanto, no treinamento de 100 MPa, foi empregado um ciclo térmico de 20 s para o aquecimento e 20 s para o resfriamento, com amplitude de 0,84 mA. Para o treinamento de 200 MPa e 300 MPa, foi submetido um ciclo de transformação de fase com 10 s para o aquecimento e 20 s para o resfriamento, com amplitude de 1,1 mA. Ademais, para a ciclagem foi utilizada uma metodologia do tempo de aquecimento e resfriamento semelhante, de ciclo térmico de 15 s para o aquecimento e de 15 s para o resfriamento, com a amplitude da corrente elétrica de 0,84 mA para as três condições de carregamento externo.

A aquisição e controle dos dados experimentais foram realizados através do NI cRIO-9076, que se comunica com o software LabVIEW e MATLAB. Os dados de temperatura e deformação foram armazenados em um computador pessoal com uma taxa de aquisição de 3,33 Hz. Além disso, para reduzir os ruídos presentes nos sinais adquiridos foi aplicado um filtro de média móvel com cinco elementos nos dados medidos do fio atuador, para que em seguida, fossem empregados no treinamento offline da rede neural.

### 3.2 DESENVOLVIMENTO DA REDE NEURAL

O sistema neural baseado no modelo NARX com o treinamento por meio do algoritmo de Levenberg-Marquardt com regularização Bayesiana, destacado na seção 2.4, foi implementado no software MATLAB. Seu respectivo código pode ser visualizado no Apêndice 2. Todavia, antes e após o treinamento, as redes neurais artificiais passam por etapas de pré-processamento e pós-processamento, além da etapa de validação que permite, posteriormente, a avaliação do desempenho dos pesos obtidos. O esquema completo das etapas envolvidas no treinamento, validação e avaliação da RNA pode ser visualizado na Figura 9.

FIGURA 9 - ETAPAS DO DESENVOLVIMENTO DA RNA



Fonte: Autor

### 3.2.1 Preparação dos dados

A preparação dos dados e a normalização fazem parte do pré-processamento da rede neural. Esta preparação consiste em dividir os dados em dois conjuntos, para o teste e para o treinamento da rede neural, e em construir o vetor de entradas da RNA a partir dos atrasos aplicados na série temporal da entrada externa, temperatura, e dos valores passados de deformação do atuador. Este vetor possui o número de colunas respectiva ao número de entradas na rede neural e ao número de linhas respectivas ao número de padrões empregados.

### 3.2.2 Normalização

As entradas de uma rede neural podem apresentar diferentes faixas de valores máximos e mínimos, fato que pode aumentar significativamente o tempo de computação do treinamento. A normalização dos dados, *a priori* do treinamento, acelera consideravelmente o tempo dos cálculos das RNA (SOLA; SEVILLA, 1997). Portanto, as entradas e alvos da rede neural são normalizadas para uma faixa semelhante, entre -1 e 1, de acordo com a função de ativação utilizada, que é a tangente hiperbólica.

### 3.2.3 Inicialização dos pesos

Os pesos e bias da rede neural são inicializados com valores aleatórios no intervalo de  $[0,1]$  para facilitar a computação da propagação de informação na rede neural, como também facilitar na correção dos seus valores durante o treinamento.

### 3.2.4 Treinamento e teste da rede neural artificial

A capacidade de aprendizado das redes neurais é advinda do processo de treinamento que ocorre nestes sistemas. Os pesos iniciais são modificados em cada época de treino de modo a reduzir uma função de custo e, conseqüentemente, reconhecimento de comportamentos não lineares como os das ligas de memória de forma. Conjuntamente com o treinamento da rede ocorre o estágio de teste para evitar

o fenômeno de sobre-treinamento, ou seja, uma parte dos dados é separada do treinamento e utilizada para analisar a capacidade de generalização da rede neural. Os dados divididos aleatoriamente para o teste devem ter a mesma distribuição de probabilidade que os do treinamento e devem ser utilizados somente para avaliação do desempenho da rede. Portanto, foram definidas as percentagens de 25% e 75% dos dados experimentais para o teste e treinamento, respectivamente, de modo a proporcionar um treinamento da rede apropriado e uma quantidade de dados satisfatória para o teste.

Um dos parâmetros de avaliação da qualidade do modelo computacional gerado através das redes neurais é a comparação entre o erro do treinamento com o erro do teste, durante esta etapa de aprendizagem. Por exemplo, um sinal de que está ocorrendo o sobre-treinamento é a verificação de uma diminuição do erro de treinamento e um aumento do erro de teste. Portanto, valores admissíveis para os erros de treinamento e de teste devem ser estabelecidos a partir da avaliação do desempenho da rede neural para permitir uma maior confiabilidade dos resultados.

Esta etapa de treinamento da rede neural também pode ser chamada de fase de identificação da liga com memória de forma. Na identificação do treinamento da LMF são empregados dados de temperatura e deformação dos 15 primeiros ciclos de transformação de fase. Esta divisão permite a construção do modelo computacional para a condição de crescimento acentuado da deformação residual em uma pequena quantidade de ciclos térmicos.

Como os efeitos da deformação residual variam significativamente com o carregamento externo aplicado ao atuador, foi aplicada a identificação da rede neural para três condições de tensão diferente, 100MPa, 200MPa e 300Mpa, de treinamento da liga. Para cada condição externa foi aplicada a mesma proporção de separação dos dados bem como os mesmos valores de 0,0005 e 0,02 do erro quadrático médio admissíveis para o treinamento e teste, respectivamente. Em seguida, através de um processo de tentativa e erro é selecionado a melhor topologia para o sistema neural que, por consequência, determinou-se os valores de 12 para o número de neurônios da camada interna e de 8 para os atrasos.

Semelhante à etapa de identificação do treinamento da liga com memória de forma é utilizada a rede neural artificial para reconhecer o comportamento do atuador durante a sua ciclagem termomecânica. Entretanto, diferentemente deste treinamento

da LMF a ciclagem é composta por um número muito maior de ciclos de transformação e é, geralmente, exposta a um carregamento mecânico inferior. Por ser uma etapa posterior, o comportamento do atuador é mais estável em relação ao alongamento e às mudanças histeréticas provocadas pela deformação residual.

A temperatura e deformação de 1000 ciclos térmicos do atuador foram utilizadas para a identificação e construção do modelo neural do sistema, para as três condições de carregamento externo de 50, 100 e 150 MPa. Os dados experimentais foram separados aleatoriamente na proporção de 25% e 75% para o teste e treinamento, respectivamente.

Os valores definidos para o erro admissível do treinamento e do teste, nesta etapa de identificação da ciclagem, foram de 0.0015 e 0.05, respectivamente, definindo-se os valores de 10 para o número de neurônios da camada interna e de 4 para os atrasos. Note-se que, tanto para a identificação do treinamento da LMF como para a ciclagem, o processo de aprendizado da rede neural foi finalizado apenas quando os valores de erro admissível para o teste e treinamento foram alcançados.

### **3.2.5 Validação**

O processo de correção dos pesos durante o treinamento pode prejudicar a capacidade de generalização da RNA, ou seja, o sistema neural pode apresentar desempenho insatisfatório na identificação de novos resultados do atuador termomecânico. Portanto, para verificar a performance do modelo quanto ao sobreajuste, informações subsequentes aos ciclos empregados na identificação são calculadas e, em seguida, comparadas com os valores experimentais adquiridos do atuador. Durante este processo, o sistema neural não passa por um processo de retreinamento, ou seja, os pesos determinados na fase anterior não são alterados. Consequente, as informações estimadas de deformação são utilizadas para a previsão da evolução da deformação residual, para a diminuição da máxima deformação de transformação e para as alterações na largura da curva histerética.

Na validação do treinamento da liga com memória de forma são empregados 10 ciclos térmicos subsequentes, enquanto que, na validação da ciclagem do atuador, são aplicados 500 ciclos de transformação de fase.

### 3.2.6 Desnormalização

A desnormalização é o processo inverso ao da normalização e é uma etapa de pós-processamento. Os dados retornam à sua faixa inicial de valores para permitir a comparação com a resposta atual do atuador de LMF.

### 3.2.7 Comparação e avaliação

Após a desnormalização, as respostas da identificação e da validação são comparadas com os valores desejados, dados experimentais do atuador termomecânico, a partir de índices de performance, como o Erro Absoluto Médio (EAM), o Erro Quadrático Médio (EQM) e a média percentual absoluta do erro, conhecida como Mean Absolute Relative Error (MAPE). Estes índices são analisados para confrontar as diferenças das respostas da rede neural para as diferentes condições de carregamento externo.

## 3.3 REDE NEURAL DE MÚLTIPLAS PREVISÕES

A rede neural de múltiplas previsões é capaz de estimar mais de uma resposta à frente do atuador com LMF, como visto na seção 2.5. Apesar de possuir mais de uma saída, estes sistemas são desenvolvidos de maneira semelhante às RNA anteriores. Para avaliar a capacidade preditiva desta rede foi considerada a situação de treinamento da LMF com a mesma condição de carregamento externo de 200MPa, utilizando 15 ciclos térmicos para o processo de aprendizado e os 10 ciclos subsequentes para a validação. Todavia, o número de saídas foi alterado para identificar o impacto do crescimento do horizonte de previsão em função do aprendizado da degradação da curva histerética, do acúmulo da deformação residual e da máxima deformação de transformação. Ademais, no intuito de obter os mesmos valores para os erros admissíveis definidos para a rede de única previsão, o novo sistema neural foi treinado para um número máximo de épocas de 500. Portanto, na tabela 1 são apresentadas as duas topologias analisadas e seus erros obtidos para treinamento e teste.

TABELA 1 – TOPOLOGIA DAS REDES DE MÚLTIPLAS PREVISÕES

Saídas	Atrasos	Neurônios	Erro do Treinamento	Erro do Teste
4	4	18	0,0022	0,0704
6	4	18	0,0034	0,1168

Fonte: Autor

## **4 RESULTADOS**

## 4 RESULTADOS

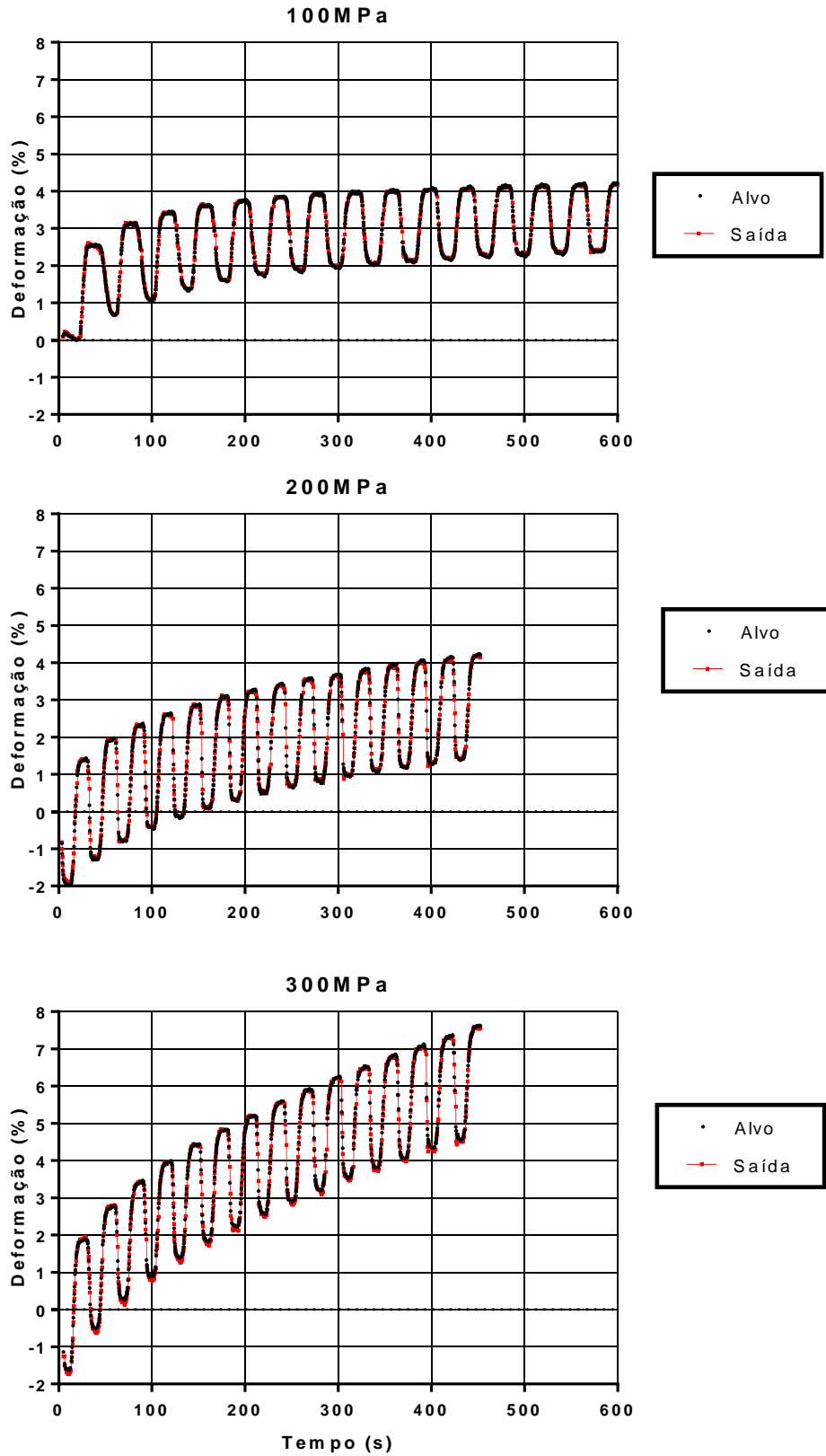
Nesta seção são apresentados os resultados da identificação e validação das fases de treinamento e ciclagem do atuador com memória de forma. A evolução da deformação residual e as mudanças nas curvas histeréticas são expostas graficamente. Os índices de desempenho são destacados para a comparação da rede neural em diferentes condições de carregamento externo.

### 4.1 RESPOSTA DO TREINAMENTO DA LMF

O processo de identificação das ligas com memória de forma é compreendido pelo reconhecimento de um valor de deformação a partir de valores anteriores de temperatura e deformação do material. Para avaliar a estabilidade dimensional do atuador durante o seu processo de treinamento, é representada a evolução da deformação em função do tempo dos 15 ciclos de transformação de fase. Além do mais, a degradação da curva histerética é destacada pela exibição da deformação em função da temperatura. As respostas gráficas destas duas funções para a três condições estudadas são apresentadas nas Figuras 10 e 11, respectivamente. A saída da rede neural é destacada em vermelho, enquanto que os dados experimentais, alvos, são expostos em cor preta.

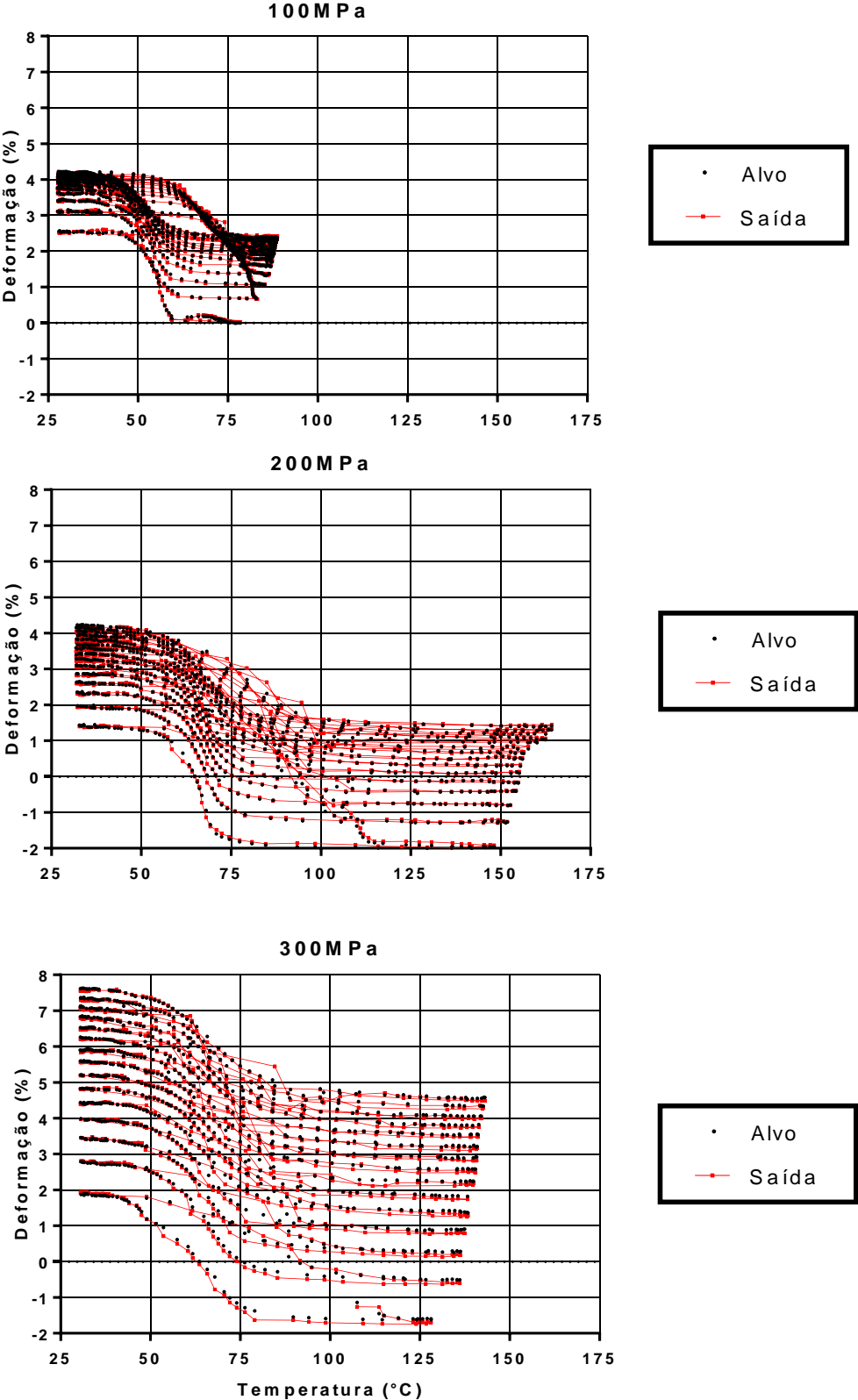
A validação do treinamento do material com memória de forma consiste em utilizar o modelo neural construído para prever o comportamento da liga sem que seja desenvolvido um novo aprendizado com os dados fornecidos. Semelhante à identificação, a previsão da evolução da deformação e da curva histerética executada pelo sistema neural é comparada com os valores experimentais para diferentes condições de carregamento externo. A previsão da estabilidade dimensional e crescimento da deformação para as tensões de 100 MPa, 200 MPa e 300 MPa podem ser visualizadas na Figuras 12. A estimativa das mudanças que ocorrem na curva histerética no treinamento do atuador como a redução da largura de histerese e a diminuição da máxima deformação de transformação são apresentadas na Figura 13, para as diferentes condições externas.

FIGURA 10 - EVOLUÇÃO DA DEFOMAÇÃO NO TREINAMENTO DA LMF



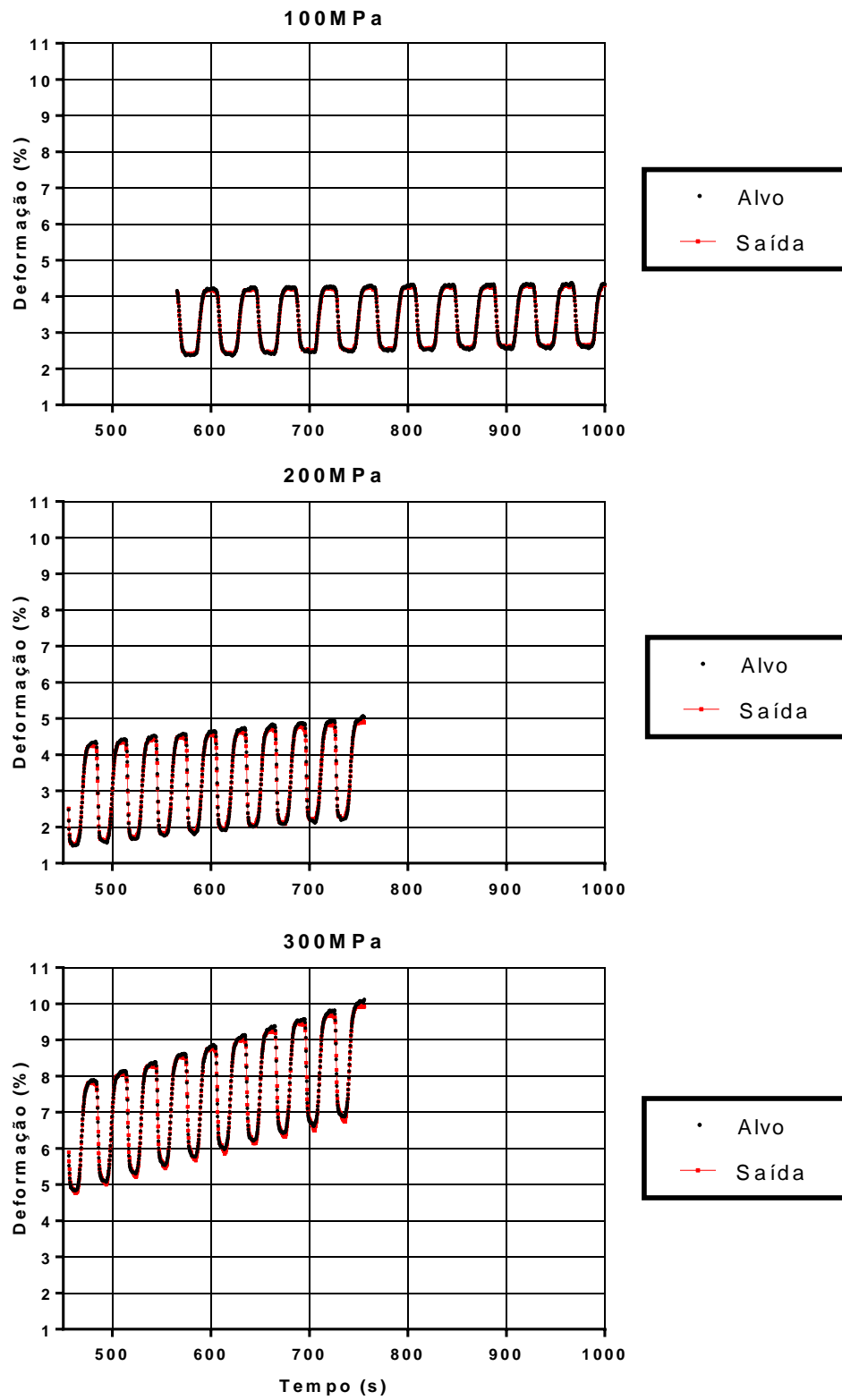
Fonte: Autor

FIGURA 11 - EVOLUÇÃO DA HISTERESE NO TREINAMENTO DA LMF



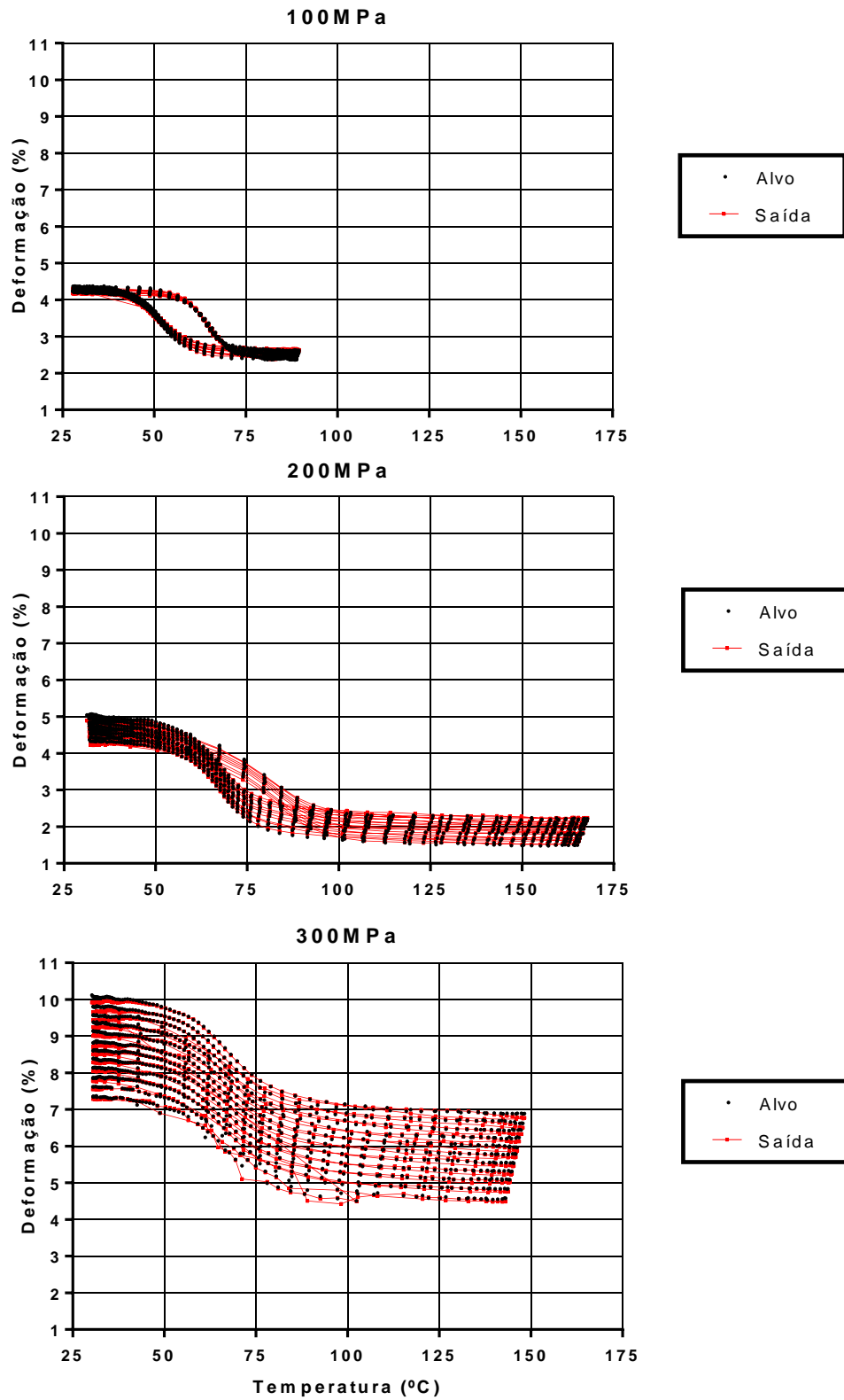
Fonte: Autor

FIGURA 12 - VALIDAÇÃO EM FUNÇÃO DO TEMPO NO TREINAMENTO DA LMF



Fonte: Autor

FIGURA 13 - VALIDAÇÃO DA HISTERESE NO TREINAMENTO DA LMF



Fonte: Autor

Examinando a evolução da deformação residual nas fases de treinamento e validação da rede neural é possível observar a fidelidade da resposta estimada com a resposta efetiva dos atuadores, para as três diferentes cargas. Além disso, o modelo de rede neural reconhece com sucesso a redução da largura histerética e a máxima deformação de transformação, apesar do erro acentuado em alguns ciclos histeréticos, especialmente para cargas maiores.

A tabela 2 expõe o desempenho da rede neural artificial em aprender o comportamento da liga com memória de forma durante o seu treinamento, ou seja, a ativação do efeito de memória de forma duplo. Este desempenho é apresentado a partir dos índices EQM, EAM e MARE da fase de aprendizado e da validação da RNA.

TABELA 2 - DESEMPENHO DA RNA PARA O TREINAMENTO DA LMF

Carga	Treinamento			Validação		
	EQM	EAM	MARE	EQM	EAM	MARE
<b>100MPa</b>	0,002226	0,032903	0,027410	0,001195	0,029937	0,009148
<b>200MPa</b>	0,005977	0,046386	0,067189	0,003895	0,049399	0,015628
<b>300MPa</b>	0,010389	0,072304	0,057971	0,010271	0,077835	0,009942

Fonte: Autor

Apesar do elevado erro médio absoluto relativo (MARE) no treinamento, a validação mostra resultados consideráveis, o que pode ser atribuído principalmente pela divisão de dados aleatórios do teste que perturba a aprendizagem neural, embora isto seja necessário para evitar o ajuste excessivo. Além disso, comparando-se o erro absoluto médio (EAM), para a mesma carga, das fases de treinamento e validação, não foi encontrada diferença significativa, mostrando que as redes neurais com regularização bayesiana apresentam generalização substancial do modelo e podem prever o comportamento da liga com memória de forma sob a influência do crescimento acentuado da deformação residual.

## 4.2 RESPOSTA DA CICLAGEM DO ATUADOR

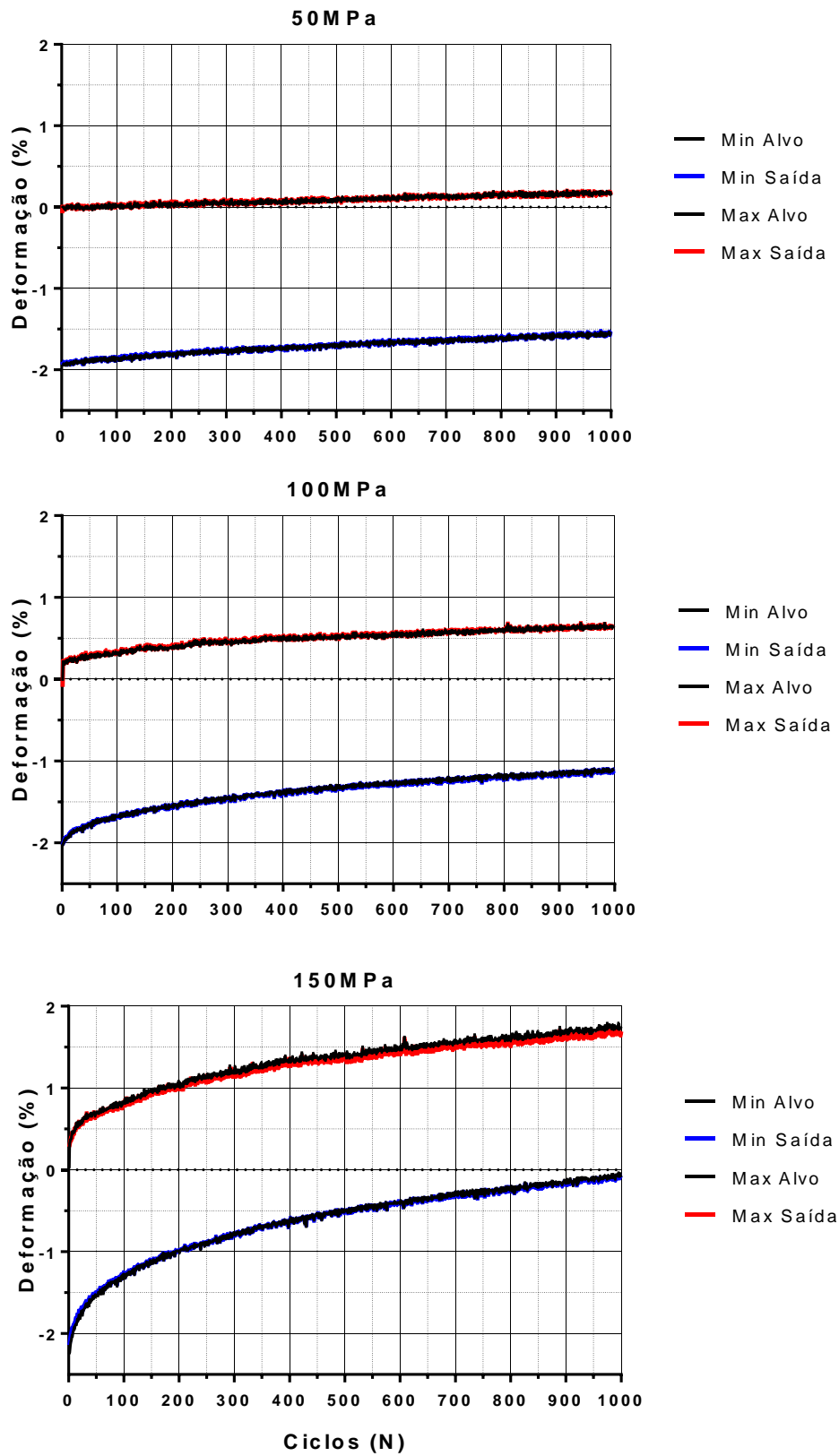
Quando o processo de treinamento do atuador é finalizado, observada-se um valor significativo de defeitos cristalinos no material e, conseqüentemente, uma estabilização da deformação residual. Em outras palavras, as mudanças nas dimensões do atuador e na sua resposta são pequenas para os ciclos seguintes de transformação. Entretanto, como os atuadores com memória de forma são geralmente submetidos a vários ciclos de transformação de fase após o processo de treinamento, o acúmulo destas mudanças dimensionais prejudica a aplicação destes transdutores termomecânicos.

A evolução da deformação para a LMF submetida a 1000 ciclos de transformação de fase para as condições de carregamento externo de 50, 100 e 150MPa é apresentada na Figura 14. Note-se que somente os valores de mínimo e máximo de deformação de cada ciclo foram destacados, para facilitar a visualização do crescimento da deformação residual e a diminuição da máxima deformação de transformação.

A resposta histerética do atuador com memória de forma para a ciclagem termomecânica é apresentada nas Figuras 15, 16 e 17, em que cada figura representa a resposta para uma condição de tensão externa diferente. Somente os ciclos 250, 500, 750 e 1000 foram apresentados para auxiliar na comparação entre os valores de deformação estimados pelo sistema neural e os dados experimentais, visto que as mudanças na curva histerética por ciclo de transformação são mais baixas quando comparadas com o treinamento do atuador.

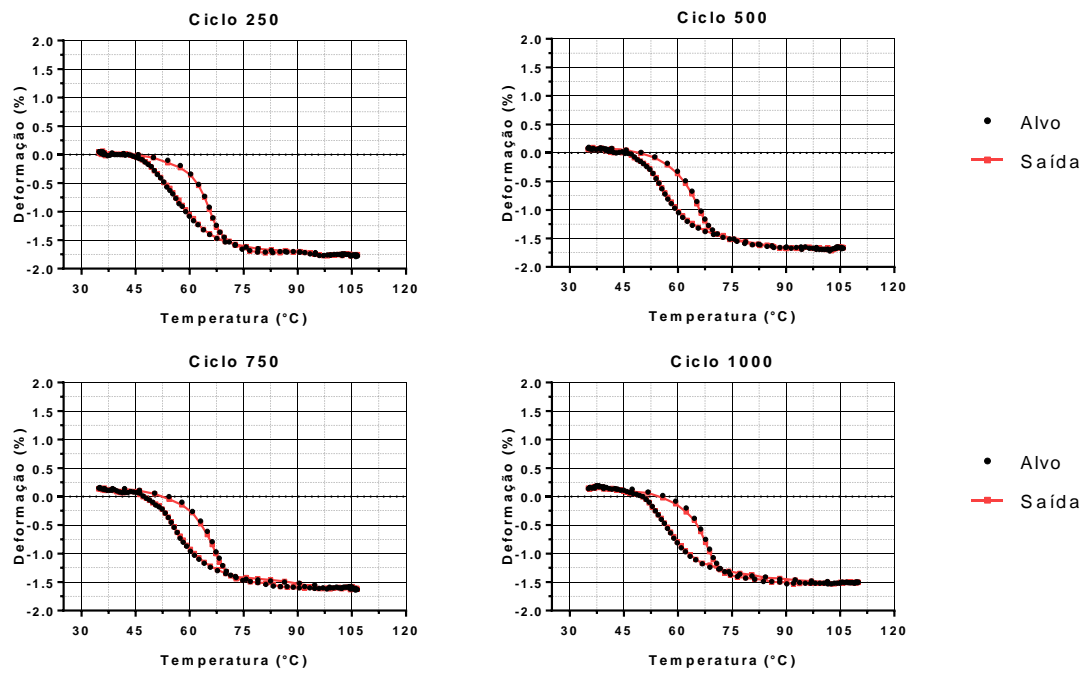
Semelhante ao treinamento do atuador, a deformação estimada pela rede neural apresenta concordância com os valores identificados, apresentando a capacidade de produzir a evolução da máxima deformação de transformação, que é um parâmetro fundamental no projeto de ligas com memória de forma.

FIGURA 14 - MÁXIMOS E MÍNIMOS DA DEFORMAÇÃO DURANTE A CICLAGEM



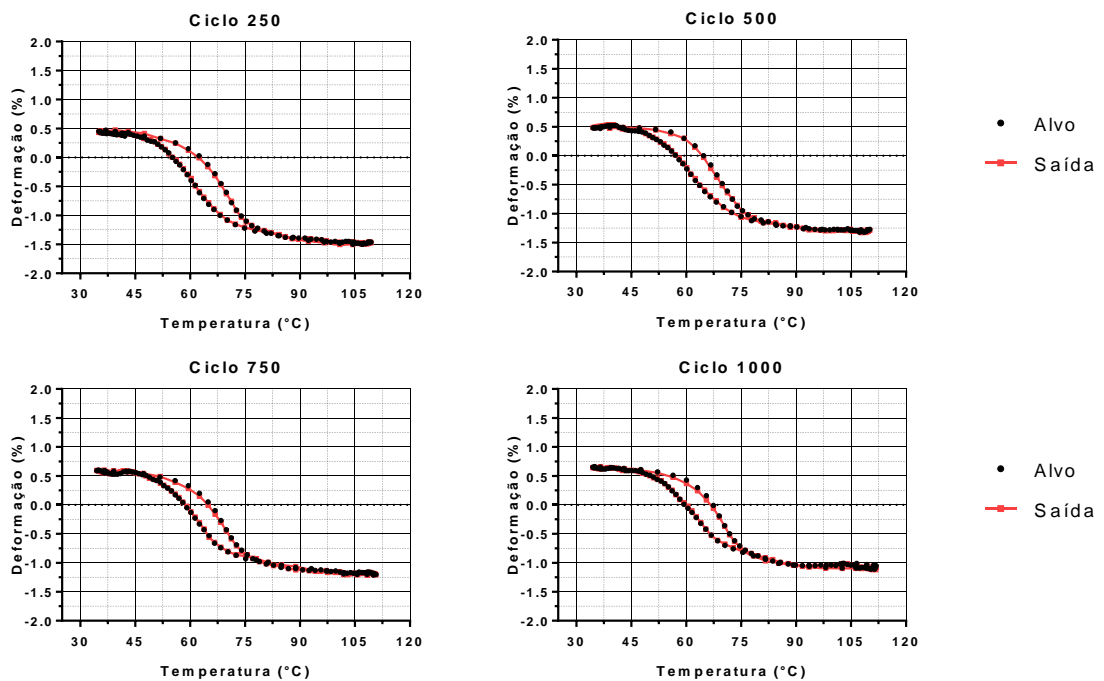
Fonte: Autor

FIGURA 15 - RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 50MPa



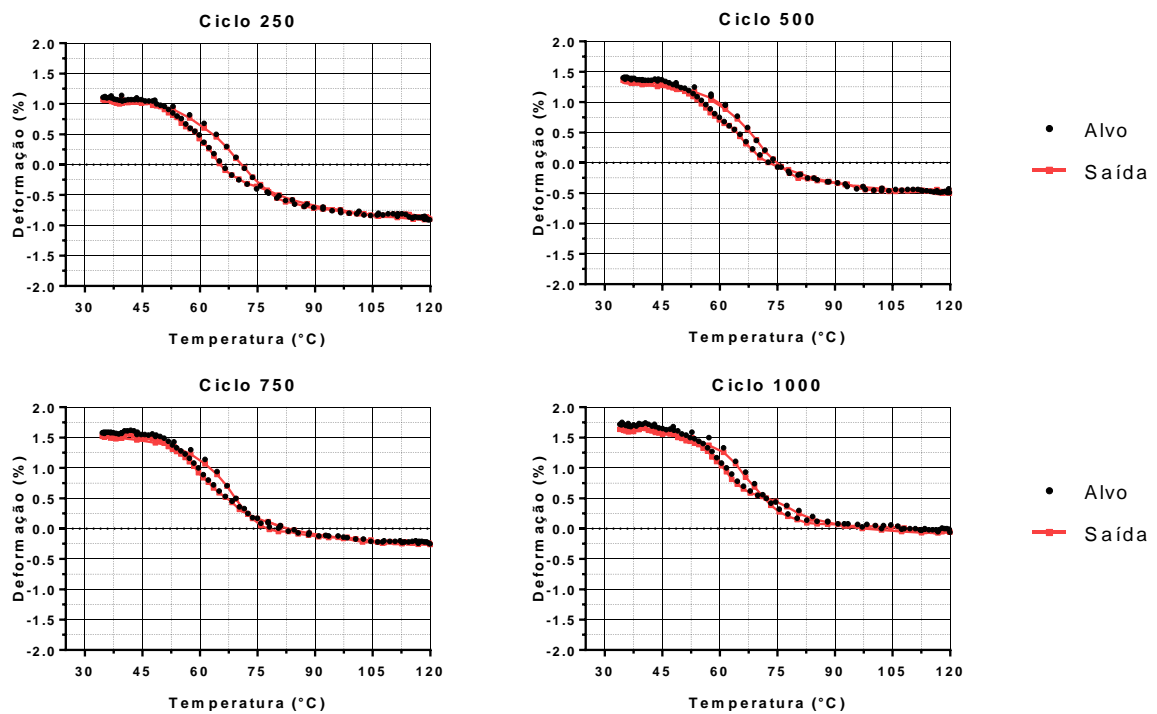
Fonte: Autor

FIGURA 16 – RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 100MPa



Fonte: Autor

FIGURA 17 - RESPOSTA HISTERÉTICA PARA A CICLAGEM DE 150MPa

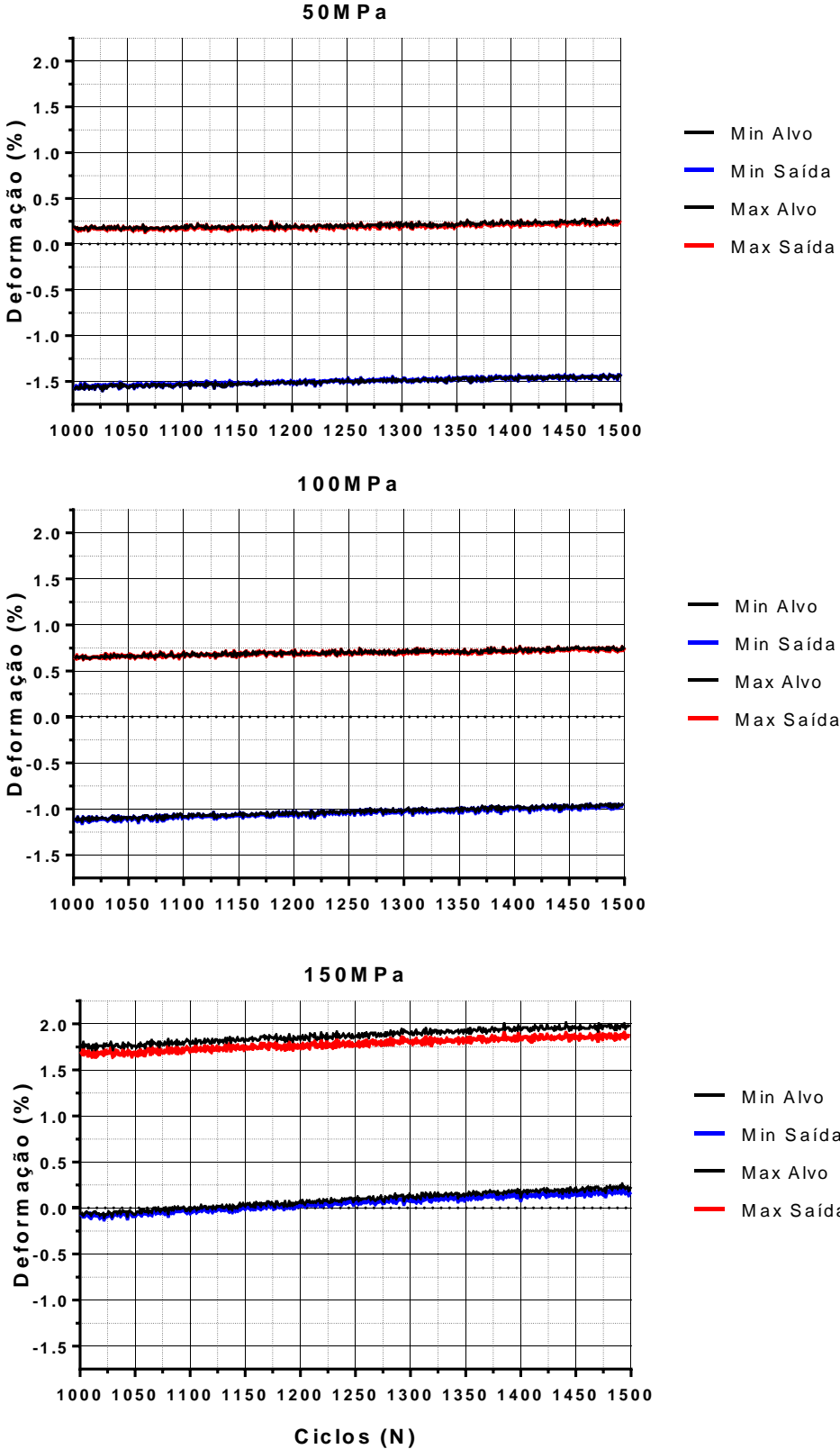


Fonte: Autor

A validação da ciclagem do atuador com memória de forma é realizada para 500 ciclos de transformação de fase subsequente aos 1000 ciclos empregados para o aprendizado da rede neural. Análogo a apresentação na fase de identificação, a resposta da evolução da deformação para a fase de validação é demonstrada pelos valores mínimos e máximos de cada ciclo, como se observa na Figura 18.

A previsão para a tensão mecânica de 150 MPa apresenta uma diferença acentuada em relação aos dados, principalmente próximo ao ciclo 1500. Esta dessemelhança verificada é causada pelo fato do acúmulo da deformação irreversível ser proporcional ao carregamento externo aplicado, apresentando uma maior variação e instabilidade, o que dificulta o processo de aprendizado do sistema neural.

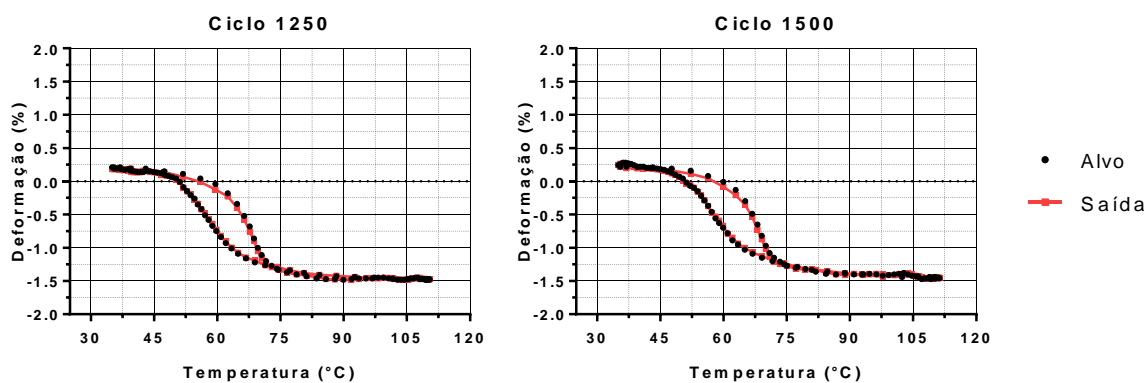
FIGURA 18 - MÁXIMOS E MÍNIMOS DA DEFORMAÇÃO PARA VALIDAÇÃO



Fonte: Autor

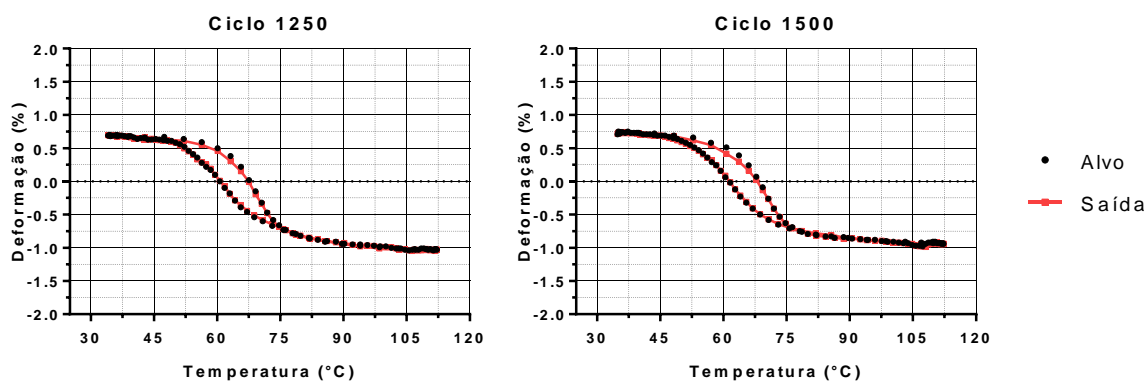
A validação da resposta histerética do atuador com memória de forma para a ciclagem termomecânica, considerando os diferentes carregamentos externos, é observada nas Figuras 19, 20 e 21. Com o intuito de facilitar a comparação da previsão da rede neural é apresentada a resposta histerética somente dos ciclos 1250 e 1500.

FIGURA 19 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 50MPA



Fonte: Autor

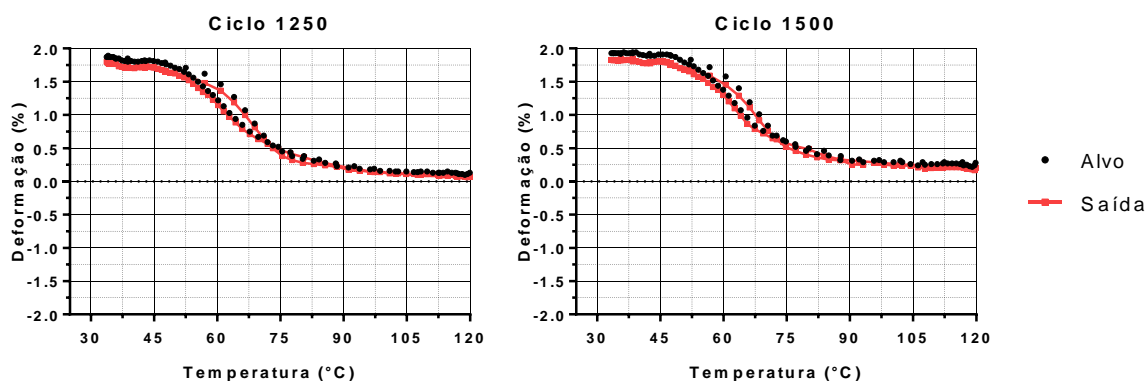
FIGURA 20 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 100MPA



Fonte: Autor

As modificações na dimensão do atuador e na resposta histerética da ciclagem para 50 MPa são significativamente menores que a ciclagem para 150 MPa, proporcionando uma resposta mais precisa para identificação e validação como observado nas figuras referentes à ciclagem. O erro quadrático médio e erro absoluto médio da identificação e da validação da ciclagem são expostos na tabela 4.

FIGURA 21 - CURVA HISTERÉTICA DA VALIDAÇÃO DA CICLAGEM PARA 150MPa



Fonte: Autor

TABELA 3 - DESEMPENHO DA RNA PARA CICLAGEM DA LMF

Load	Treinamento		Validação	
	EQM	EAM	EQM	EAM
50MPa	0,001675	0,014300	0,000467	0,015608
100MPa	0,001834	0,013000	0,000370	0,013652
150MPa	0,003487	0,004320	0,004322	0,056847

Fonte: Autor

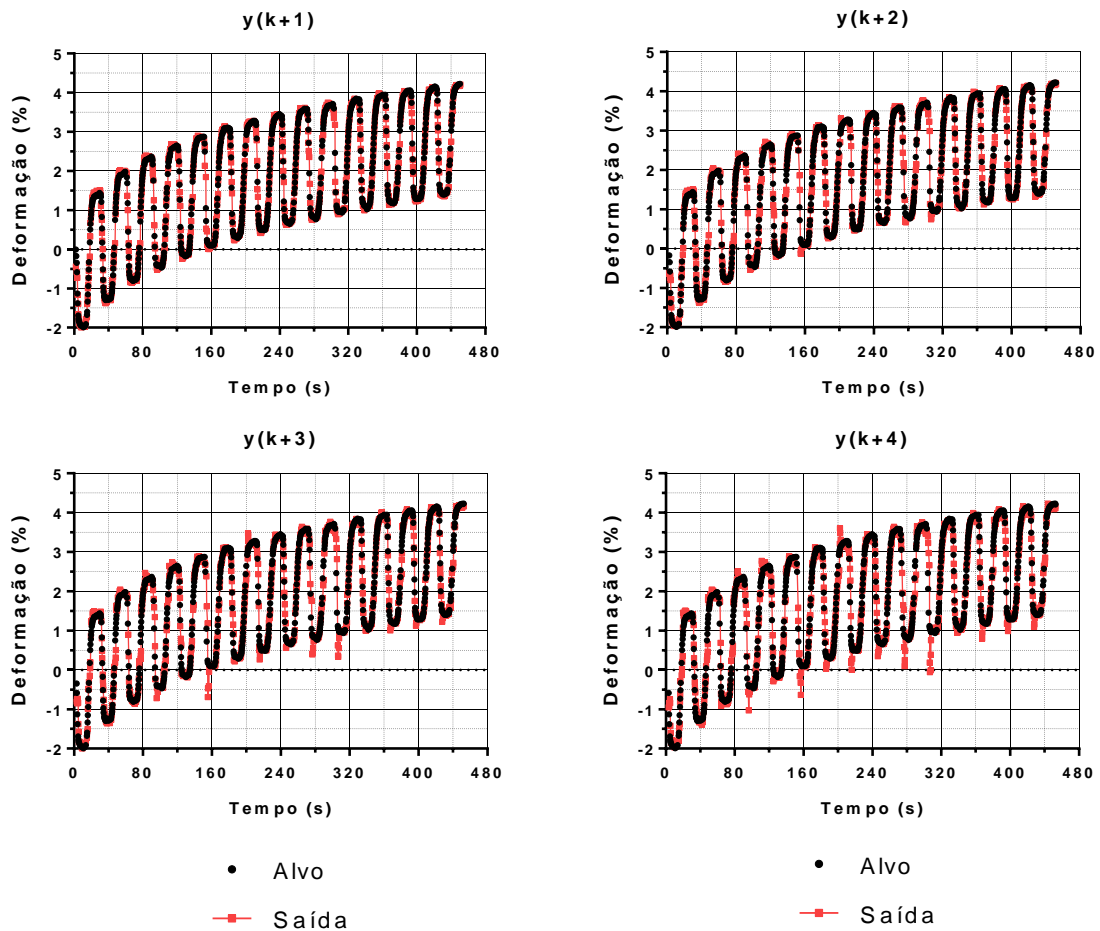
Confrontando os valores dos erros é notável as semelhanças para o treinamento e validação do sistema neural. Além do mais, os erros para as condições de 50 MPa e 100 MPa são próximos, contudo, o erro para a tensão de 150 MPa é relativamente maior para a validação como observado nas figuras anteriores.

#### 4.3 RESPOSTA DA REDE NEURAL DE MÚLTIPLAS SAÍDAS

A rede de múltiplas previsões é empregada para a identificação dos 15 primeiros ciclos do treinamento da liga com memória de forma, de maneira análoga à rede NARX um passo à frente apresentada anteriormente. Todavia, apenas o carregamento externo de 200 MPa é utilizado para a verificação do desempenho deste sistema preditivo. A resposta das deformações para o treinamento considerando a

previsão de 4 saídas e de 6 saídas à frente pode ser observada nas Figuras 22 e 23, respectivamente.

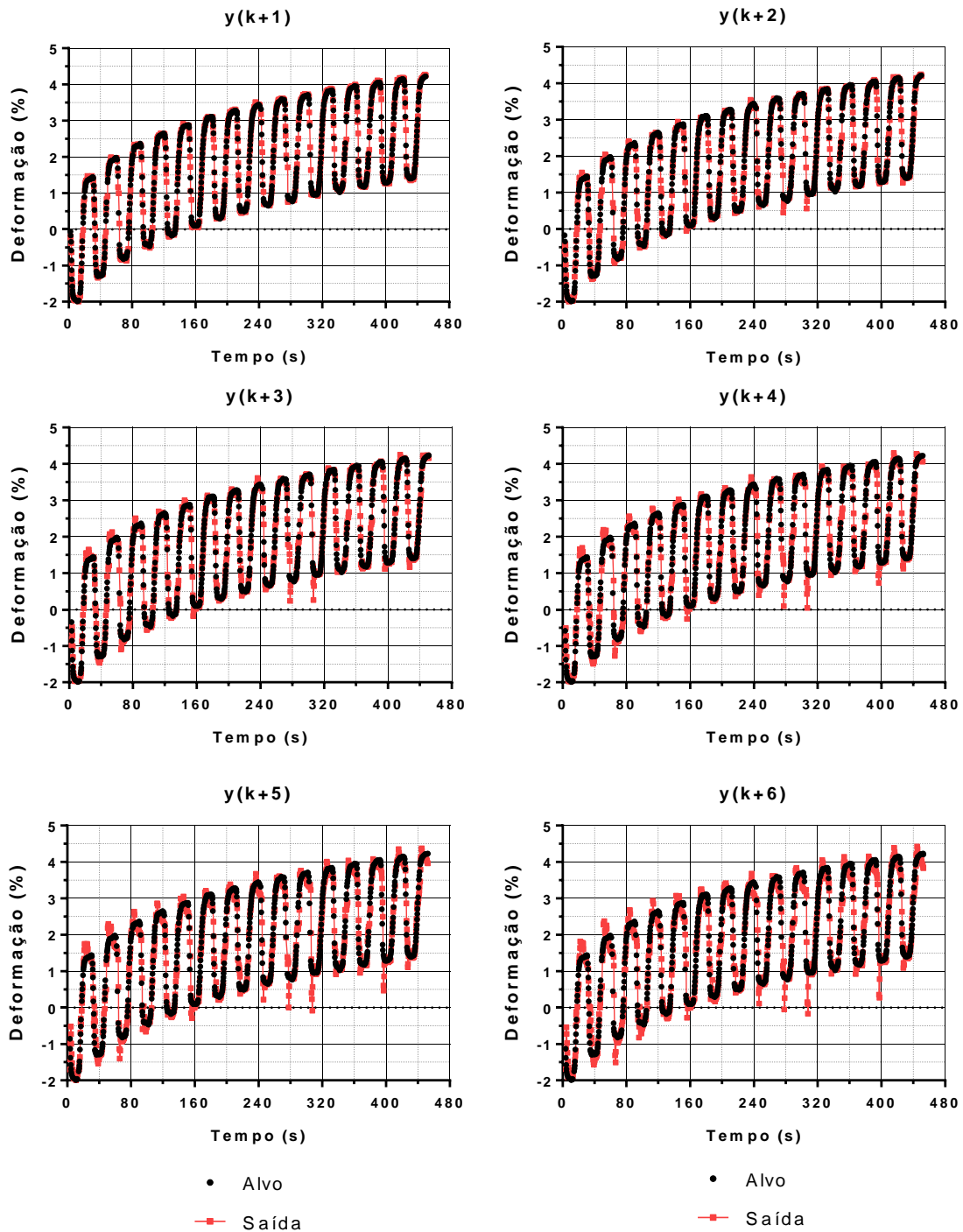
FIGURA 22 - TREINAMENTO DA RNA PREDITIVA DE 4 SAÍDAS



Analisando a evolução da deformação gerada por cada uma das saídas é observada a relação do crescimento do erro com o avançado da previsão. Em outras palavras, as saídas próximas ao final do horizonte de previsão apresentam maior divergências da resposta quando comparadas com os valores experimentais. Além disso, o crescimento do número de respostas estimadas também dificulta o treinamento para a obtenção de um mesmo valor de erro quadrático médio. Desta maneira, a RNA de múltiplas saídas tende a apresentar desempenho inferior para a identificação do comportamento do material com memória de forma quando comparado com os sistemas neurais de única resposta.

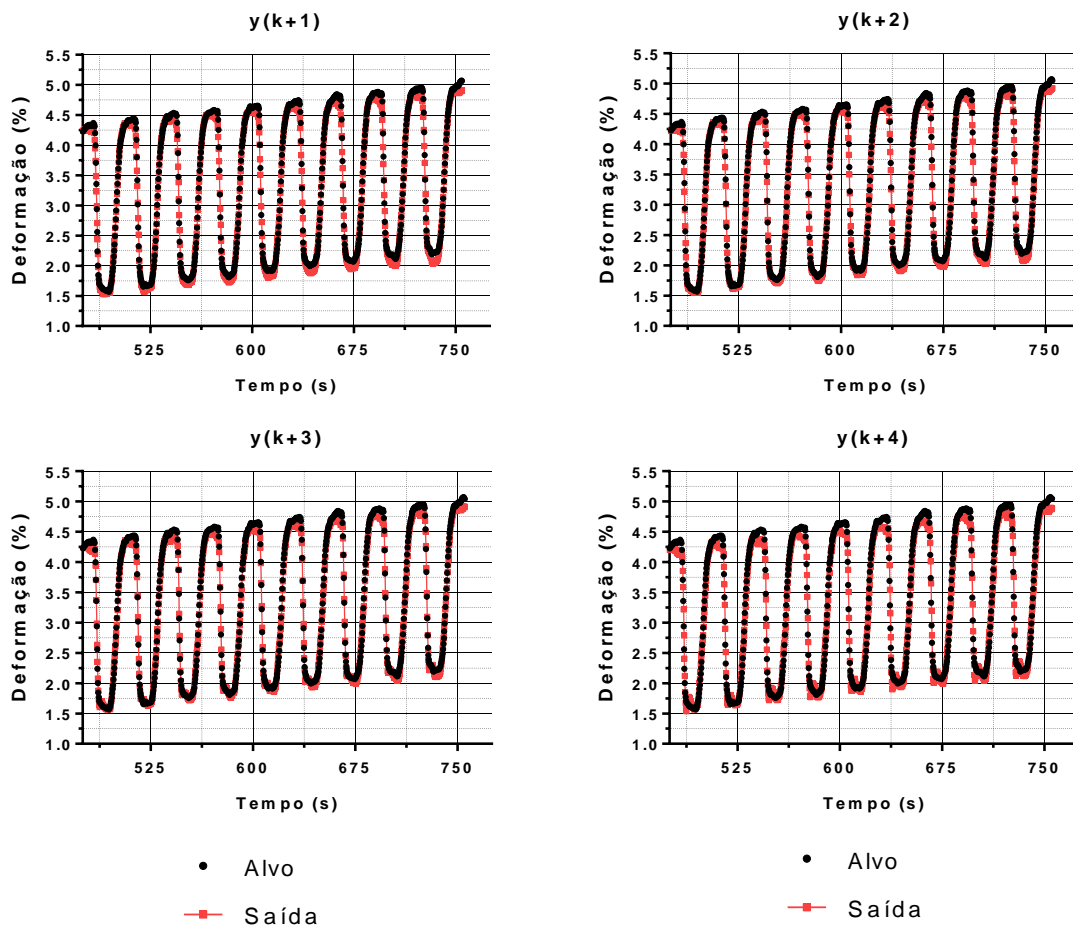
Os 10 ciclos seguintes ao treinamento são empregados para a validação da rede de múltiplos passos à frente, como pode ser observado nas Figuras 24 e 25.

FIGURA 23 - TREINAMENTO DA RNA PREDITIVA DE 6 SAÍDAS



Fonte: Autor

FIGURA 24 - VALIDAÇÃO DA RNA PREDITIVA DE 4 SAÍDAS

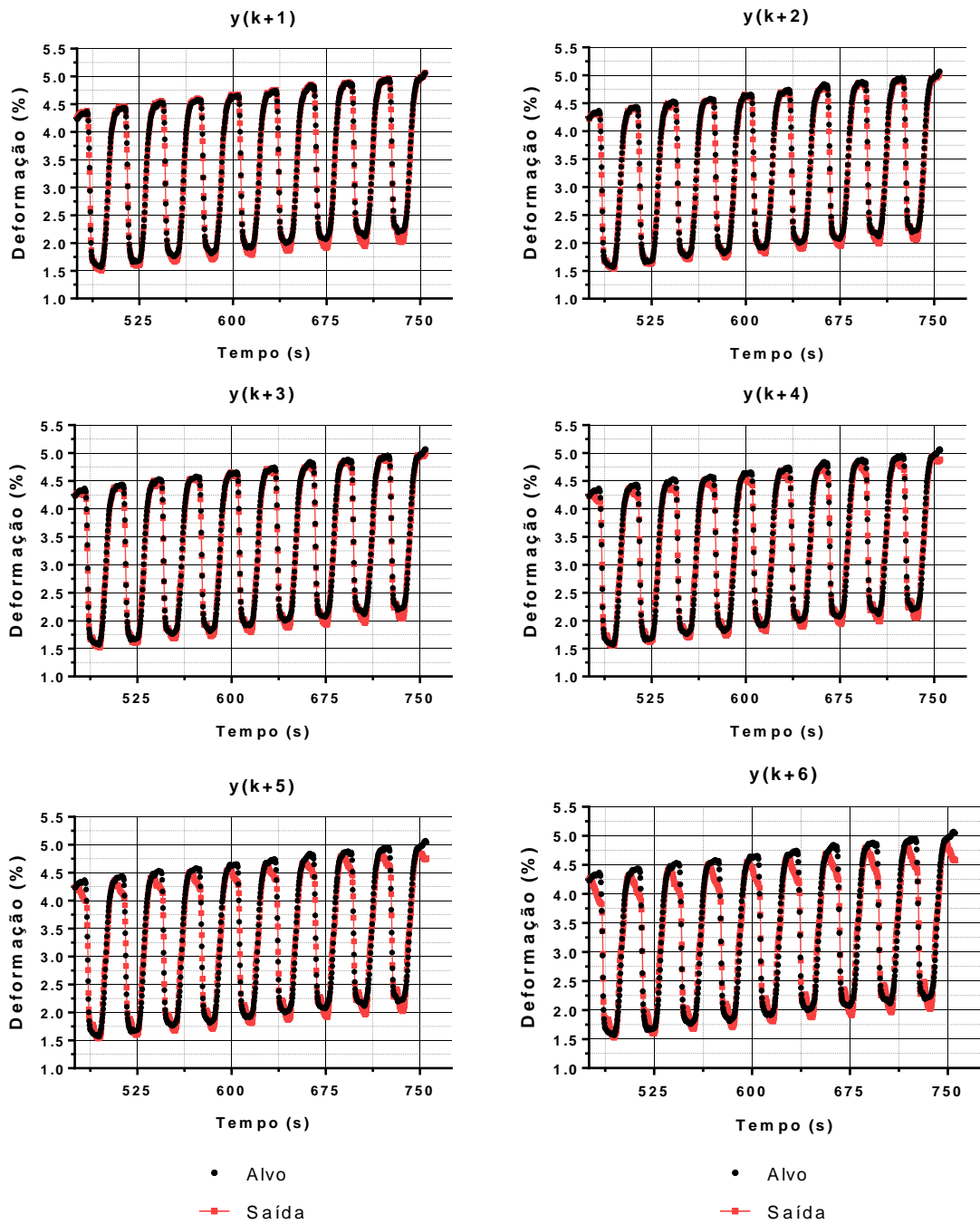


Fonte: Autor

O resultado da validação para a rede neural de 6 saídas apresenta considerável oscilação dos resultados, particularmente nos valores de máximos e mínimos de cada ciclo de transformação de fase. Este fato representa um obstáculo na estimativa da máxima deformação de transformação e na utilização destes sistemas preditivos para os atuadores com memória de forma.

A avaliação do desempenho da rede de múltiplas previsões é realizada a partir da análise dos índices do erro quadrático médio, erro absoluto médio e erro relativo absoluto médio. O resultado destes índices para os dois casos de rede neural desenvolvidas pode ser observado na Tabela 4.

FIGURA 25 - VALIDAÇÃO DA RNA PREDITIVA DE 6 SAÍDAS



Fonte: Autor

TABELA 4 - DESEMPENHO DA RNA DE MÚLTIPLAS PREVISÕES

Saídas	Treinamento			Validação		
	EQM	EAM	MARE	EQM	EAM	MARE
4	0,028584	0,098949	0,279989	0,043051	0,078872	0,026313
6	0,009492	0,126332	0,708788	0,017463	0,094924	0,032828

Fonte: Autor

Um erro consideravelmente elevado é encontrado quando comparando o desempenho da rede de múltiplas saídas com as a rede de um passo à frente, evidenciando que a vantagem de obter um maior número de estimativas do comportamento pode acompanhar uma menor precisão dos resultados. Além disso, o erro relativo do treinamento é significativamente maior que o erro relativo da validação, enquanto que o erro quadrático e absoluto médio não sofrem grandes alterações.

## **5 DISCUSSÃO**

## 5 DISCUSSÃO

O algoritmo de treinamento de Levenberg-Marquardt com Regularização bayesiana exibiu resultados satisfatória uma vez que em conjunto com uma arquitetura não linear auto regressiva conseguiu aprender o comportamento de condições distintas como o treinamento do atuador que apresenta elevado acúmulo de deformação residual em poucos ciclos de transformação de fase e a ciclagem que exhibe um crescimento considerável ao longo de vários ciclos térmicos. Além do mais, este algoritmo evidencia notável robustez por evitar o sobre-treinamento do modelo computacional e rápida convergência do erro de treinamento para os valores admissíveis.

A validação da rede neural à um passo à frente apresentou menor erro absoluto médio e erro quadrático médio que a etapa de aprendizagem, para ambos os casos de treinamento e ciclagem do atuador. Desta maneira, é possível utilizar o sistema neural para ciclos subsequentes aos empregados no reconhecimento do comportamento das ligas com memória de forma e, conseqüentemente, aplicar no desenvolvimento e controle destes atuadores sob o efeito do acúmulo da deformação irreversível. Além do mais, o erro mais elevado para a etapa de treinamento e teste pode indicar a necessidade de modificações na fase de pré-processamento, na metodologia de divisão dos dados ou até na modificação da topologia para aperfeiçoamento dos resultados. Por exemplo, a divisão dos dados entre as fases de treinamento e de teste após a normalização e construção do vetor de entradas pode manter uma maior consistência do comportamento da LMF, no lugar de desenvolver estas fases somente com suas informações próprias.

A identificação e previsão do atuador durante o processo de treinamento do material com memória de forma foi inferior à etapa de ciclagem. A capacidade de aprendizado da rede neural é consideravelmente afetada pelo acúmulo da deformação irreversível, evidenciando a importância da capacidade de adaptação do sistema de identificação e controle nos atuadores com memória de forma quando as modificações de dimensão são acentuadas.

Apesar da resposta histerética da LMF ser influenciada pelo carregamento externo aplicado ao material, o modelo neural desenvolvido para a previsão a um

passo à frente apresenta desempenho adequado em reconhecer a degradação da histerese. Em outras palavras, a RNA identifica e prevê a redução da máxima deformação de transformação e da largura da curva histerética, fatores essenciais no projeto de atuadores com estas ligas. Ademais, examinando os gráficos é possível identificar que quanto maior a tensão mecânica aplicada mais acentuado é a deformação residual e que a rede neural artificial consegue acompanhar este padrão. Portanto, visto que a mesma topologia foi empregada para diferentes condições de carregamento externo e os resultados do erro relativo absoluto médio foram inferiores a 1,6%, para o treinamento do atuador, o sistema neural apresenta considerável capacidade de generalização.

Analisando a rede neural de múltiplas saídas, verifica-se que o benefício de expandir o horizonte de precisão acompanha uma maior carga computacional e uma menor precisão de identificação e validação do modelo. Este erro não é tão significativo para as primeiras saídas, ou seja, para previsões próximas dos dados empregados. Todavia, conforme o número de saídas cresce e a resposta se distancia no futuro, é encontrado um resultado mais oscilatório e com erros acentuados das máximas e mínimas deformações de cada ciclo. Este fato dificulta a aplicação deste modelo para determinação da degradação da máxima deformação de transformação e da estabilidade dimensional do atuador. Portanto, faz-se necessário o aprofundamento da aplicação das redes neurais de múltiplos passos à frente para previsão dos atuadores de memória de forma na presença de deformação residual significativa, seja através da otimização da topologia empregada ou por meio do desenvolvimento de melhores metodologias para divisão dos dados e normalização.

Por fim, por mais que a rede neural de múltiplas previsões tenha apresentado erros superiores a previsão à um passo a frente, o seu resultado da validação foi considerável visto que o erro absoluto relativo médio foi de 2,63% e 3,28% para a previsão com quatro e seis saídas, respectivamente

## **6 CONCLUSÕES**

## 6 CONCLUSÕES

As redes neurais artificiais são conhecidas por serem aproximadores de função universal, permitindo o aprendizado de comportamentos contínuos não lineares até um grau de precisão determinado. Diferentemente de vários trabalhos de rede neural empregados em ligas com memória de forma, esta dissertação visa a utilização de uma rede neural recorrente para a identificação e previsão de atuadores com memória de forma considerando as influências da deformação residual.

As redes neurais de previsão a um passo à frente demonstra a capacidade de aprender coerentemente as modificações provocadas pela deformação residual que ocorrem nos materiais com memória de forma. Como por exemplo, o modelo NARX apresentado em conjunto com o treinamento de Levenberg-Marquardt com Regularização Bayesiana é capaz de identificar o acúmulo da deformação irreversível, a diminuição da máxima deformação de transformação e a redução da largura da curva histerética a partir da estimativa de deformação do atuador. Além do mais, para validar o modelo, ciclos subsequentes às etapas de aprendizado são empregados e estimados com resultado relevante, por apresentar erro quadrático médio inferior a 0.02 na etapa de treinamento do atuador, e 0.05 na etapa de ciclagem da liga com memória de forma.

Comparando os resultados para diferentes condições de carregamento externo é perceptível a influência da tensão mecânica aplicada com o desempenho da rede neural em prever a evolução da deformação e degradação da histerese. Estes resultados evidenciam a importância da utilização de sistemas de identificação e controle adaptativos para a obtenção de resultados satisfatórios, como também validam a utilização da rede neural artificial como modelo de referência para o desenvolvimento, construção e controle de transdutores termomecânicos baseados em ligas com memória de forma.

Finalmente, o sistema neural de vários passos à frente manifesta desempenho considerável, mesmo com índices superiores ao modelo de única previsão, com erro absoluto relativo médio próximo de 3% para as condições estudadas, de quatro e seis previsões.

## REFERÊNCIAS

- AURICCHIO, F.; TAYLOR, R. L. Shape-memory alloys: modelling and numerical simulations of the finite-strain superelastic behavior. **Computer Methods in Applied Mechanics and Engineering**, v. 143, n. 1–2, p. 175–194, 1997.
- BAO, Y.; XIONG, T.; HU, Z. Multi-step-ahead time series prediction using multiple-output support vector regression. **Neurocomputing**, v. 129, p. 482–493, 2014.
- BÁSILIO SOBRINHO, J. M. et al. Electronic instrumentation for the characterization of a rotary thermoelectric motor driven by shape memory alloy springs. **I2MTC 2017 - 2017 IEEE International Instrumentation and Measurement Technology Conference, Proceedings**, 2017.
- BENAFAN, O. et al. Thermomechanical cycling of a NiTi shape memory alloy-macroscopic response and microstructural evolution. **International Journal of Plasticity**, v. 56, p. 99–118, 2014.
- CHATTERJEE, S. et al. Software fault prediction using Nonlinear Autoregressive with eXogenous Inputs (NARX) network. **Applied Intelligence**, v. 37, n. 1, p. 121–129, 2012.
- CHTOUROU, S.; CHTOUROU, M.; HAMMAMI, O. A hybrid approach for training recurrent neural networks: Application to multi-step-ahead prediction of noisy and large data sets. **Neural Computing and Applications**, v. 17, n. 3, p. 245–254, 2008.
- FERRONATO, G. Universidade Federal De Santa Catarina Programa De Pós-Graduação Em Ciência Da. 2008.
- HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. H. Neural Network Design. **Boston Massachusetts PWS**, v. 2, p. 734, 1995.
- KAYRI, M. Predictive Abilities of Bayesian Regularization and Levenberg–Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data. **Mathematical and Computational Applications**, v. 21, n. 2, p. 20, 2016.
- KOKOL, P. et al. Intelligent Systems. p. 306–311, 2000.
- KRIESEL, D. A Brief Introduction to Neural Networks. **Retrieved August**, p. 244, 2005.
- KUMAR, P. K.; LAGOUDAS, D. C. **Introduction to shape memory alloys**. [s.l.: s.n.].
- LIMA, W. M. Study Of the Stability Of the NiTi Wire Applied To Thermomechanical Actuators. [s.d.].

- LUO, H. Y.; ABEL, E. W. A comparison of methods for the training of NiTi two-way shape memory alloy. **Smart Materials and Structures**, v. 16, n. 6, p. 2543–2549, 2007.
- MOHD JANI, J. et al. A review of shape memory alloy research, applications and opportunities. **Materials and Design**, v. 56, p. 1078–1113, 2014.
- MOHD JANI, J.; LEARY, M.; SUBIC, A. Designing shape memory alloy linear actuators: A review. **Journal of Intelligent Material Systems and Structures**, v. 28, n. 13, p. 1699–1718, 2017.
- NESPOLI, A. et al. The high potential of shape memory alloys in developing miniature mechanical devices: A review on shape memory alloy mini-actuators. **Sensors and Actuators, A: Physical**, v. 158, n. 1, p. 149–160, 2010.
- PARLOS, A. G. G.; RAIS, O. T. T.; ATIYA, A. F. F. Multi-step-ahead prediction using dynamic recurrent neural networks. **Neural Networks**, v. 13, n. 7, p. 765–86, 2000.
- SCIRÈ MAMMANO, G.; DRAGONI, E. Functional fatigue of Ni-Ti shape memory wires under various loading conditions. **International Journal of Fatigue**, v. 69, p. 71–83, 2014.
- SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. **IEEE Transactions on Nuclear Science**, v. 44, n. 3 PART 3, p. 1464–1468, 1997.
- WALKER, J. M. et al. **Artificial Neural Networks Methods and Applications**. [s.l.: s.n.].
- WANG, H.; SONG, G. Innovative NARX recurrent neural network model for ultra-thin shape memory alloy wire. **Neurocomputing**, v. 134, p. 289–295, 2014.
- ZHOU, B. A macroscopic constitutive model of shape memory alloy considering plasticity. **Mechanics of Materials**, v. 48, p. 71–81, 2012.



## APÊNDICE 1 - DIVERGENTE DA FUNÇÃO DE CUSTO E DO ERRO

- Matriz Jacobiana

A matriz Jacobiana,  $J$ , pelo método de Levenberg-Marquardt auxilia na identificação da matriz Hessiana pro representar a derivada de primeira ordem do erro em função dos pesos da rede neural (KOKOL et al., 2000). Considerando uma rede neural com o número de padrões de treinamento representado por  $N$ , o número de pesos representado por  $N_W$  e apenas uma saída, a matriz Jacobinana é descrita por:

$$J = \frac{\partial e}{\partial W} = \begin{bmatrix} \frac{\partial e_1}{\partial W_1} & \frac{\partial e_1}{\partial W_2} & \dots & \frac{\partial e_1}{\partial W_{N_W}} \\ \frac{\partial e_2}{\partial W_1} & \frac{\partial e_2}{\partial W_2} & \dots & \frac{\partial e_2}{\partial W_{N_W}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_N}{\partial W_1} & \frac{\partial e_N}{\partial W_2} & \dots & \frac{\partial e_N}{\partial W_{N_W}} \end{bmatrix}$$

- Divergente da Função de Custo

A função de custo utilizando a regularização Bayesiana é:

$$E(W) = \beta \sum_{i=1}^N (d_i - y_i)^2 + \alpha \sum_j^{N_W} W_j^2 = \beta e^2 + \alpha w^2$$

Onde  $w$  apresenta a matriz composta pelos pesos e  $e$  simboliza a matriz do erro da resposta da rede neural. O divergente da Função de custo é a derivada dessa função em relação aos pesos que compõem a rede neural. Portanto, a equação que expressa esta derivada de primeiro grau é:

$$\nabla E = \frac{\partial E}{\partial W} = \frac{\partial (\beta e^2 + \alpha w^2)}{\partial W} = 2\beta J e + 2\alpha w$$

- Derivada de Segunda Ordem da Função de Custo

Derivando o divergente da função de custo é possível encontrar a derivada de segunda ordem:

$$\nabla\nabla E = \frac{\partial^2 E}{\partial W^2} = \frac{\partial (2\beta J e + 2\alpha w)}{\partial W} = 2\beta J^T J + 2\alpha$$

Onde  $T$  representa a transposta da matriz a que se refere.



## APÊNDICE 2 - CÓDIGO MATLAB DA IDENTIFICAÇÃO E VALIDAÇÃO

```

%-----
%Limpando o Workspace e Fechando todas as figuras
clear;close all;clc;
%-----
% Treinamento Levenberg-Marquardt com Regularização Bayesian
%-----
% Inicializando os arquivos que contem os dados do treinamento
load Training200MPa.txt;
load Training200MPaC15.txt;
load Training200MPaC25.txt;
%-----
%Dados Originais do experimento
time = Training200MPa(:,1); %tempo
temperature = Training200MPa(:,4); %temperature
strain = Training200MPa(:,3); %deformação
%Divisão dos dados entre Treinamento e Teste
[train,validation,test] = dividerand(Training200MPaC15',0.75,0,0.25);
%Preparação das variáveis de tempo, temperatura e deformação do experimento
time_train = train(1,:);
temperature_train = train(4,:);
strain_train = train(3,:);
temperature_test = test(4,:);
strain_test = test(3,:);
N_train = length(strain_train); %Número de dados do treinamento
N_test = length(strain_test); %Número de dados de teste
NumEpoch = 500; %Número Máximo de Épocas
Error_ADM_train = 2.5e-4; %Erro admissível para o treinamento
Error_ADM_test = 0.05; %Erro admissível para o teste
%Preparação das variáveis da validação
time_val = Training200MPaC25(:,1);
temperature_val = Training200MPaC25(:,4);
strain_val= Training200MPaC25(:,3);
N_val = length(strain_val);
%-----
%Normalização dos dados
temperature_train_norm = temperatureNorm(temperature_train,temperature);
strain_train_norm = targetNorm(strain_train,strain);
temperature_test_norm = temperatureNorm(temperature_test,temperature);
strain_test_norm = targetNorm(strain_test,strain);
%-----
%Inicializando os pesos com valores aleatórios entre 0 e 1
NumDelay = 4; %Número de Atrasos 10 - 8
NumNeurons = 18; %Número de Neurônios 18 - 22
NumInputs = NumDelay*2; %Número de Entradas
NumOutputs = 4; %Número de Saídas
W = rand(NumInputs,NumNeurons); %Pesos da Camada de Interna
Bw = rand(1,NumNeurons); %Pesos dos Bias de Interna
V = rand(NumNeurons,NumOutputs); %Pesos da Camada de Saída
Bv = rand(1,NumOutputs); %Pesos dos Bias de Saída
%-----
% Preparando o vetor de entrada da rede 2neural a partir do número de
atraso
% selecionado com os dados de temperatura e realimentação da saída.

```

```

[input_train,target_train] =
Preparation(temperature_train_norm,strain_train_norm,NumDelay,NumOutputs,N_
train);
[input_test,target_test] =
Preparation(temperature_test_norm,strain_test_norm,NumDelay,NumOutputs,N_te
st);
%-----
%Inicializando o coeficiente de combinação e os hiperparâmetros
mu = 10; alpha = 0.01; beta = 1;
%-----
% Preparação dos Vetores de Erro do Treinamento e do Teste
Error_Train = zeros(NumEpoch,1);
Error_Test = zeros(NumEpoch,1);
%Preparação da figura da evolução dos erros
figure(1);
loglog(Error_Train);grid on;hold on;
% Construindo os arquivos para o armazenamento dos erros
file_ID00 = fopen('Errors_Multi_200MPa.txt','wt');
%-----
% Treinamento para um valor máximo de épocas ou até que um valor admissível
% do erro seja alcançado.
for m = 1:NumEpoch
    %Evitando instabilidade devido ao valor de mu muito pequeno ou grande
    if (mu > 1e20 || mu < 1e-10)
        disp('Erro devido ao valor do mu');
        break
    end
    %Início do Loop de Treinamento: Minimização da Função de Custo
    weight = [W(:);Bw(:);V(:);Bv(:)]; %Vetor de pesos
    %Propagação dos dados na rede neural
    [y_train,J_train] =
    Propagation(input_train,weight,NumInputs,NumNeurons,NumOutputs);
    [y_test,J_test] =
    Propagation(input_test,weight,NumInputs,NumNeurons,NumOutputs);
    Error_Train(m) = mean(mean((target_train - y_train).^2)); %Erro do
    treinamento
    Error_Test(m) = mean(mean((target_test - y_test).^2)); %Erro do
    treinamento
    %Construindo Gráfico da evolução dos erros encontrados
    plot(m,Error_Train(m),'b*',m,Error_Test(m),'r*');
    %Testando se o erro foi minimizado ao valor admissível
    if (Error_Train(m) < Error_ADM_train) && (Error_Test(m) <
    Error_ADM_test)
        disp('Treinamento Encerrado');
        break
    else
        %Calculando a matriz Hessiana e realizando a correção dos pesos
        Hessian = 2*beta*(J_train'*J_train) +
        2*alpha*mu*eye(length(weight));
        error = (target_train - y_train)';
        dw = Hessian\ (2*beta*J_train'*error(:) + 2*alpha*weight);
        weight_pos = weight - dw;
        [y_pos,~] =
        Propagation(input_train,weight_pos,NumInputs,NumNeurons,NumOutputs);
        Error_Pos = mean(mean((target_train - y_pos).^2));
        %Se o erro for reduzido devido as mudanças nos pesos estas mudanças
        são
        %salvas, senão as correções são descartadas.
        if Error_Pos <= Error_Train(m)
            mu = mu/5;

```

```

weight = weight_pos;
gama = length(weight) - 2*alpha*(trace(inv(Hessian)));
alpha = gama/(2*(weight'*weight));
beta = (N_train - gama)/(2*sum(sum((target_train -
y_pos).^2)));
    %Reorganizando os vetores de pesos para o armazenamento da
configuração
    %da rede neural e para o processo de propagação
W =
reshape(weight(1:NumInputs*NumNeurons), [NumInputs, NumNeurons]);
Bw =
reshape(weight(NumInputs*NumNeurons+1:(NumInputs*NumNeurons)+NumNeurons), [1
, NumNeurons]);
V =
reshape(weight((NumInputs*NumNeurons)+NumNeurons+1:(NumInputs*NumNeurons)+N
umNeurons+(NumNeurons*NumOutputs)), [NumNeurons, NumOutputs]);
Bv =
reshape(weight((NumInputs*NumNeurons)+NumNeurons+(NumNeurons*NumOutputs)+1:
end), [1, NumOutputs]);
else
mu = mu*5;
end
end
Error_Test(m)
Error_Train(m)
    %Construindo Gráfico da evolução dos erros encontrados
loglog(m, Error_Train(m), 'b*', m, Error_Test(m), 'r*');
pause(0.1)
    %Armazenando os dados dos erros em arquivo txt
data_saved = [Error_Train(m); Error_Test(m)];
fprintf(file_ID00, '%8.20f %8.20f\n', data_saved');
end
    %Finalizando o armazenamento dos dados
fclose(file_ID00);
    %Desnormalização da resposta da rede neural e dos dados do treinamento
response_train_denorm = targetDenorm(y_train, strain);
strain_train_denorm = targetDenorm(target_train, strain);
[time_train_forward, temperature_train_forward] =
Forward(time_train, temperature_train, NumDelay, NumOutputs, N_train);
%-----
    %Previsão para 25 ciclos de transformação de fase
weight = [W(:); Bw(:); V(:); Bv(:)];
    %Normalização dos dados
temperature_val_norm = temperatureNorm(temperature_val, temperature);
strain_val_norm = targetNorm(strain_val, strain);
    %Preparação das entradas e da saída
[input_val, target_val] =
Preparation(temperature_val_norm, strain_val_norm, NumDelay, NumOutputs, N_val)
;
    %Propagação dos dados para a validação
[y_val, ~] = Propagation(input_val, weight, NumInputs, NumNeurons, NumOutputs);
    %Desnormalização da resposta da validação
response_val_denorm = targetDenorm(y_val, strain);
[time_val_forward, temperature_val_forward] =
Forward(time_val, temperature_val, NumDelay, NumOutputs, N_val);
strain_val_denorm = targetDenorm(target_val, strain);
%-----
    %Avaliação do treinamento com a validação
error_train = mean((strain_train_denorm -
response_train_denorm)./strain_train_denorm)

```

```
error_val = mean((strain_val_denorm -
response_val_denorm)./strain_val_denorm)
```

```
%-----
%Função de Normalização dos Dados de Temperatura
%Esta função recebe como dados de entrada o dado a ser normalizado e o
%parâmetro utilizado para a normalização.
function output = temperatureNorm(input,parameter)
    %Valores máximos e mínimos da normalização
    ymax = 1; ymin = -1;
    %Normalização
    output = ymin + (ymax - ymin)*(input -min(parameter))/(max(parameter) -
min(parameter));
end
```

```
%-----
%Função de Normalização dos Dados de Deformação
%Esta função recebe como dados de entrada o dado a ser normalizado e o
%parâmetro utilizado para a normalização.
function output = targetNorm(input,parameter)
    %Valores máximos e mínimos da normalização
    ymax = 1; ymin = -1;
    %Normalização
    output = ymin + (ymax - ymin)*(input -min(parameter))/(max(parameter) -
min(parameter));
end
```

```
%-----
% Preparação do vetor de entrada e de alvos para o treinamento da rede
function [inputs,targets] =
Preparation(temperature_norm,strain_norm,NumDelay,NumOutputs,N)

    for i = NumDelay:N-NumOutputs
        inputs(i-NumDelay+1,:) = [temperature_norm(i-
NumDelay+1:i);strain_norm(i-NumDelay+1:i)]';
        targets(i-NumDelay+1,:) = strain_norm(i+1:i+NumOutputs);
    end
end
```

```
%Descrição da função Propagation
function [y,J] = Propagation(Inputs,weight,NumInputs,NumNeurons,NumOutputs)
%Reorganizando a matriz dos pesos
W = reshape(weight(1:NumInputs*NumNeurons),[NumInputs,NumNeurons]);
Bw =
reshape(weight(NumInputs*NumNeurons+1:(NumInputs*NumNeurons)+NumNeurons),[1
,NumNeurons]);
V =
reshape(weight((NumInputs*NumNeurons)+NumNeurons+1:(NumInputs*NumNeurons)+N
umNeurons+(NumNeurons*NumOutputs)),[NumNeurons,NumOutputs]);
Bv =
reshape(weight((NumInputs*NumNeurons)+NumNeurons+(NumNeurons*NumOutputs)+1:
end),[1,NumOutputs]);
%Constructor
```

```

y = []; k = 0; n = length(Inputs);
JacobianW = [];JacobianBw = [];JacobianV =
zeros (n*NumOutputs,NumNeurons*NumOutputs);JacobianBv =
zeros (n*NumOutputs,NumOutputs);
%Resposta do Somatório e da Função de Ativação
for i = 1:n
    n1 = Inputs(i,:)*W+Bw; %Resposta do Somatório dos Neurônios da Camada
de Entrada
    a1 = tansig(n1); %Função de Ativação da Camada de Entrada
    n2 = a1*V+Bv; %Resposta do Somatório dos Neurônios da Camada Interna
    a2 = purelin(n2); %Função de Ativação da Camada Interna
    y = [y;a2]; %Resposta da Saída Estimada Pela Rede

%Calculo da matriz Jacobian
for j = 1:NumOutputs
    k = k +1;
    delta2 = -dpurelin(n2(j),a2(j));
    delta1 = delta2*V(j)'.*dtansig(n1,a1);
    %Calculo das componentes da matriz Jacobiana
    JV = a1'*delta2;
    JBv = delta2;
    JW = Inputs(i,:)'*delta1;
    JBw = delta1;

    JacobianW = [JacobianW;JW(:)'];
    JacobianBw = [JacobianBw;JBw(:)'];
    JacobianV(k,(j-1)*length(JV)+1:j*length(JV)) = JV(:);
    JacobianBv(k,(j-1)*length(JBv)+1:j*length(JBv)) = JBv(:);
end
end
%Reorganizando a matriz Jacobiana
J = [JacobianW JacobianBw JacobianV JacobianBv];
End

%-----
%Preparação dos dados de temperatura e tempo para a previsão de múltiplas
%saídas
function [time_forward,temperature_forward] =
Forward(time,temperature,NumDelay,NumOutputs,N)

for i = NumDelay:N-NumOutputs
    time_forward(i-NumDelay+1,:) = time(i+1:i+NumOutputs);
    temperature_forward(i-NumDelay+1,:) = temperature(i+1:i+NumOutputs);
end

%-----
%Função de Desnormalização dos Dados de Deformação
%Esta função recebe como entrada os dados normalizados e o vetor parâmetro
%para a normalização
function output = targetDenorm(input,parameter)
    %Valores máximos e mínimos da normalização
    ymax = 1; ymin = -1;
    %Normalização
    output = min(parameter) + (max(parameter) - min(parameter))*(input -
ymin)/(ymax - ymin);
end
%-----

```

---

```
%Função de Desnormalização dos Dados de Temperatura
%Esta função recebe como entrada os dados normalizados e o vetor parâmetro
%para a normalização
function output = temperatureDenorm(input,parameter)
    %Valores máximos e mínimos da normalização
    ymax = 1; ymin = -1;
    %Normalização
    output = min(parameter) + (max(parameter) - min(parameter))*(input -
ymin)/(ymax - ymin);
end
```