

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Thiago Liquita Sávio

**Aplicação de Redes Neurais em um Manipulador Paralelo
de Elos Flexíveis para a Extração de Modelos**

São Carlos

2023

Thiago Liquita Sávio

**Aplicação de Redes Neurais em um Manipulador Paralelo
de Elos Flexíveis para a Extração de Modelos**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Ciências - Programa de Pós-Graduação em Engenharia Mecânica.

Área de concentração: Dinâmica e Mecatrônica

Orientador: Profa. Dra. Maíra Martins da Silva

VERSÃO CORRIGIDA

**São Carlos
2023**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

S267a Sávio, Thiago Liquita
Aplicação de Redes Neurais em um Manipulador
Paralelo de Elos Flexíveis para a Extração de Modelos /
Thiago Liquita Sávio; orientadora Maíra Martins da
Silva. São Carlos, 2023.

Dissertação (Mestrado) - Programa de
Pós-Graduação em Engenharia Mecânica e Área de
Concentração em Dinâmica e Mecatrônica -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2023.

1. Manipuladores paralelos flexíveis. 2. Redes
Neurais Artificiais (RNA). 3. 3RRR. 4. TLFN. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **THIAGO LIQUITA SÁVIO**.

Título da dissertação: "Aplicação de redes neurais em um manipulador paralelo de elos flexíveis para a extração de modelos".

Data da defesa: 24/03/2023

Comissão Julgadora

Resultado

Profa. Associada **Maíra Martins da Silva**

APROVADO

(Orientadora)

(Escola de Engenharia de São Carlos – EESC/USP)

Prof. Dr. **Wesley Angelino de Souza**

APROVADO

(Universidade Tecnológica Federal do Paraná/UTFPR)

Prof. Dr. **Rodrigo Elias Bianchi**

APROVADO

(Instituto Federal de Educação, Ciência e Tecnologia de São Paulo/IFSP)

Coordenador do Programa de Pós-Graduação em Engenharia

Mecânica:

Prof. Associado **Adriano Almeida Gonçalves Siqueira**

Presidente da Comissão de Pós-Graduação:

Prof. Titular **Carlos De Marqui Junior**

Dedico este trabalho a todas as pessoas que me ajudaram em minha jornada até aqui.

AGRADECIMENTOS

Primeiramente agradeço a minha orientadora Profa. Dra. Máira Martins da Silva por ter me auxiliado durante todo o desenvolvimento do projeto.

Agradeço a minha mãe Gercilia A. L. Sávio e ao meu pai José C. Sávio por me darem todo o apoio necessário para que fosse possível chegar até aqui. E também a minha irmã Tatiane L. Sávio.

Agradeço ao Prof. Dr. Wesley Angelino de Souza e ao Prof. Dr. Rodrigo Elias Bianchi pelas contribuições dadas a esse trabalho.

Agradeço também a todos os meus parentes e amigos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES). Por meio do programa de demanda social. Dessa forma para essa agência também vão meus agradecimentos.

*“O estudo, a busca da verdade e da beleza são domínios
em que nos é consentido sermos crianças por toda a vida.”*

Albert Einstein

RESUMO

Sávio, T.L. **Aplicação de Redes Neurais em um Manipulador Paralelo de Elos Flexíveis para a Extração de Modelos**. 2023. 126p. Dissertação - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Os Manipuladores Paralelos (*Parallel Manipulator* - PM) estão sendo mais utilizados a cada dia, como uma alternativa viável de projeto para aplicações industriais. Fatores que contribuem para isso, são algumas vantagens provenientes desse tipo de manipulador, como: a leveza, elevadas relações velocidades/acelerações, rigidez elevada, capacidade de carga e alta compactação. No entanto, esta opção de projeto pode gerar vibrações indesejadas devido à flexibilidade de seus componentes, exigindo a implementação de novas estratégias de controle de juntas e espaço de tarefas. Dois desafios principais surgem ao projetar uma estratégia de controle para um PM: a falta de uma medição direta da pose do efetuador final e sua dinâmica de acoplamento. Este trabalho propõe um estimador para avaliar a pose do efetuador final de um PM usando Redes Neurais Artificiais (RNAs) estáticas (*multilayer perceptron* - MLP) e com atraso de tempo (*focused time lagged feedforward network* - TLFN focada, e *distributed time lagged feedforward network* - TLFN distribuída). Encoders, extensômetros e câmera foram utilizados para realizar as medições, os encoders medem o deslocamento angular das juntas ativas do manipulador, os extensômetros a deformação dos elos e a câmera a posição do efetuador final. A proposta é validada usando dados experimentais de um PM 3RRR com elos flexíveis. No trabalho em questão foi possível treinar uma RNA com 22 trajetórias que foi capaz de prever o posicionamento de uma trajetória que não participou do treinamento, através dos dados de entrada, com uma média do MAPE (*mean absolute percentage error*) para as 3 saídas do efetuador final de 6,53 % para uma MLP, 3,43 % para uma TLFN focada e 6,52 % para uma TLFN distribuída, a média do MSE (*mean aquare error*) foram respectivamente, $1,0 * 10^{-3}m^2$, $1,6 * 10^{-4}m^2$ e $7,4 * 10^{-4}m^2$.

Palavras-chave: Manipuladores paralelos flexíveis. Redes Neurais Artificiais (RNA). 3RRR. TLFN.

ABSTRACT

Sávio, T.L. **Application of Neural Networks in a Parallel Flexible Link Manipulator for Model Extraction.** 2023. 126p. Dissertação - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Parallel Manipulators (*Parallel Manipulator* - PM) are being used more daily, as a viable design alternative for industrial applications. Factors that contribute to this are some advantages of this type of manipulator, such as: lightness, high speed/acceleration ratios, high rigidity, load capacity and high compaction. However, this design option can generate unwanted vibrations due to the flexibility of its components, requiring the implementation of new joint control strategies and task space. Two main challenges arise when designing a control strategy for a PM: the lack of a direct measurement of the end-effector pose and its coupling dynamics. This work proposes an estimator to evaluate the final effector pose of a PM using Artificial Neural Networks (ANNs) static (*multilayer perceptron* - MLP) and with time delay (*focused time lagged feedforward network* - focused TLFN, and *distributed time lagged feedforward network* - distributed TLFN). Encoders, strain gauges and a camera were used to perform the measurements, the encoders measure the angular displacement of the active joints of the manipulator, the strain gauges the deformation of the links and the camera the position of the end effector. The proposal is validated using experimental data from a PM 3RRR with flexible links. In the work in question, it was possible to train an ANN with 22 trajectories that was able to predict the positioning of a trajectory that did not participate in the training, through the input data, with an average of the MAPE (*mean absolute percentage error*) for the 3 end-effector outputs of 6.53 % for an MLP, 3.43 % for a focused TLFN, and 6.52 % for a distributed TLFN, the mean of the MSE (*mean square error*) were, respectively, $1.0 * 10^{-3}m^2$, $1.6 * 10^{-4}m^2$ and $7.4 * 10^{-4}m^2$.

Keywords: Flexible parallel manipulators. Artificial Neural Networks (ANN). 3RRR. TLFN.

LISTA DE FIGURAS

Figura 1 – Manipuladores	29
Figura 2 – (a) Protótipo Par2 construído no LIRMM e (b) Comparação entre duas estratégias de controle.	30
Figura 3 – Duas formas de representar um neurônio artificial	36
Figura 4 – Funções de ativação	38
Figura 5 – Rede Alimentada Adiante com Camada Única	39
Figura 6 – MLP	40
Figura 7 – Rede Recorrente	40
Figura 8 – Ilustração do fluxo de sinais básicos em um perceptron de múltiplas camadas	42
Figura 9 – TLFN focada	43
Figura 10 – Protótipo atual	45
Figura 11 – Evolução do protótipo	46
Figura 12 – Aparato Experimental	48
Figura 13 – Instrumentação e seu esquema de comunicação	49
Figura 14 – Extensômetros	50
Figura 15 – Metodologia da medida, técnica de medição	50
Figura 16 – Modelo do manipulador	51
Figura 17 – Esquemático da RNA	53
Figura 18 – Esquemático de atuações no manipulador para obtenção de dados	55
Figura 19 – Posição do efetuador final (dados brutos) e deslocamentos incrementais dos encoders (dados brutos) para uma das atuações feitas no experimento	57
Figura 20 – Medidas de tensão para os extensômetro (dados brutos) para uma das atuações feitas no experimento	58
Figura 21 – Posição do efetuador final (filtrado x não-filtrado) e deslocamentos dos encoders para uma das atuações feitas no experimento	60
Figura 22 – Medidas de tensão para os extensômetro (filtrado x não-filtrado) para uma das atuações feitas no experimento	61
Figura 23 – Esquemático da RNA para testes com uma única trajetória como entrada	62
Figura 24 – Esquemático da matriz do erro	62
Figura 25 – Esquemático da RNA atrasada focada no tempo	64
Figura 26 – Esquemático da RNA atrasada distribuída no tempo	65
Figura 27 – RNA_{22} aplicada à trajetória 23 para os três tipos de RNAs	67
Figura 28 – RNA_{21} aplicada à trajetória 22 para os três tipos de RNAs	68
Figura 29 – RNA_{21} aplicada à trajetória 23 para os três tipos de RNAs	69
Figura 30 – RNA_{20} aplicada à trajetória 21 para os três tipos de RNAs	70

Figura 31 – RNA_{20} aplicada à trajetória 22 para os três tipos de RNAs	71
Figura 32 – RNA_{20} aplicada à trajetória 23 para os três tipos de RNAs	72
Figura 33 – Parte I da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	80
Figura 34 – Parte II da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	81
Figura 35 – Parte III da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	82
Figura 36 – Parte IV da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	83
Figura 37 – Parte V da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	84
Figura 38 – Parte I da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	86
Figura 39 – Parte II da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	87
Figura 40 – Parte III da matriz de erro (MAPE) para treinamento com uma traje- tória, RNA de Múltiplas Camadas	88
Figura 41 – Parte IV da matriz de erro (MAPE) para treinamento com uma traje- tória, RNA de Múltiplas Camadas	89
Figura 42 – Parte V da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas	90
Figura 43 – Parte I da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas	92
Figura 44 – Parte II da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas	93
Figura 45 – Parte III da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas	94
Figura 46 – Parte IV da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas	95
Figura 47 – Parte V da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas	96
Figura 48 – Parte I da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas	98
Figura 49 – Parte II da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas	99
Figura 50 – Parte III da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas	100

Figura 51 – Parte IV da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas	101
Figura 52 – Parte V da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas	102
Figura 53 – Parte I da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA Focada	104
Figura 54 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	105
Figura 55 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	106
Figura 56 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	107
Figura 57 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	108
Figura 58 – Parte I da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	110
Figura 59 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	111
Figura 60 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	112
Figura 61 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	113
Figura 62 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada	114
Figura 63 – Parte I da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	116
Figura 64 – Parte II da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	117
Figura 65 – Parte III da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	118
Figura 66 – Parte IV da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	119
Figura 67 – Parte V da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	120
Figura 68 – Parte I da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	122
Figura 69 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	123

Figura 70 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	124
Figura 71 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	125
Figura 72 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA	126

LISTA DE TABELAS

Tabela 1 – Constantes Geométricas	52
Tabela 2 – Parâmetros do treinamento.	62
Tabela 3 – Representação de parte das matrizes de erro.	63
Tabela 4 – Representação de parte das matrizes de erro.	63
Tabela 5 – Parâmetros do treinamento RNAs atrasadas focadas no tempo.	64
Tabela 6 – Parâmetros do treinamento RNAs atrasadas distribuídas no tempo.	65
Tabela 7 – Comparação da previsão entre os tipos de RNA para a RNA_{22} aplicada à trajetória 23.	66
Tabela 8 – Comparação da previsão entre os tipos de RNA para a RNA_{21} aplicada à trajetória 22 e 23.	67
Tabela 9 – Comparação da previsão entre os tipos de RNA para a RNA_{20} aplicada à trajetória 21, 22 e 23.	70

LISTA DE QUADROS

Quadro 1 – Descrição dos componentes do protótipo	47
---	----

LISTA DE ABREVIATURAS E SIGLAS

PM	<i>Parallel Machines</i>
RNA	Redes Neurais Artificiais
MSE	<i>Mean Squared Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
TLFN	<i>focused time lagged feedforward network</i>
MLP	<i>Multilayer Perceptron</i>
ECU	<i>Electronic Control Unity</i>
EESC-USP	Escola de Engenharia de São Carlos - São Paulo
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

LISTA DE SÍMBOLOS

x_j	Entrada de sinal em uma sinapse j de um neurônio k
w_{kj}	Peso sináptico do neurônio k na sinapse j
b_k	Bias aplicado a um neurônio k
ϕ	Função de ativação
v_k	Potencial de ativação
w_{k0}	Transformação do bias em uma entrada sináptica do neurônio com entrada sempre igual a +1 e peso igual ao valor do bias
n	número de amostras
y_i	Valor aproximado
y'_i	Valor absoluto
$M1$	Motor 1
$M2$	Motor 2
$M3$	Motor 3
x	Posição do efetuador final no eixo x
y	Posição do efetuador final no eixo y
α	Rotação do efetuador final
θ_i	Ângulo de rotação de uma junta ativa
β_i	Ângulo de rotação de uma junta passiva
λ_i	Ângulo em relação à normal de cada lado do triângulo formado pela área de trabalho do manipulador
A_i	Junta ativa do manipulador
B_i	Junta passiva intermediária do manipulador
C_i	Junta passiva próxima ao efetuador final
μ_i	Variável auxiliar

ρ_i	Variável auxiliar
l_i	Comprimento do elo i
h	Distância entre o ponto central do efetuador e a segunda junta passiva
r_i	Vetor posição do elo i
e_1	medida de deslocamento angular realizada pelo encoder 1
e_2	medida de deslocamento angular realizada pelo encoder 2
e_3	medida de deslocamento angular realizada pelo encoder 3
s_1	medida de tensão associada a deformação do elo 1
s_2	medida de tensão associada a deformação do elo 2
s_3	medida de tensão associada a deformação do elo 3
s_4	medida de tensão associada a deformação do elo 4
s_5	medida de tensão associada a deformação do elo 5
s_6	medida de tensão associada a deformação do elo 6

SUMÁRIO

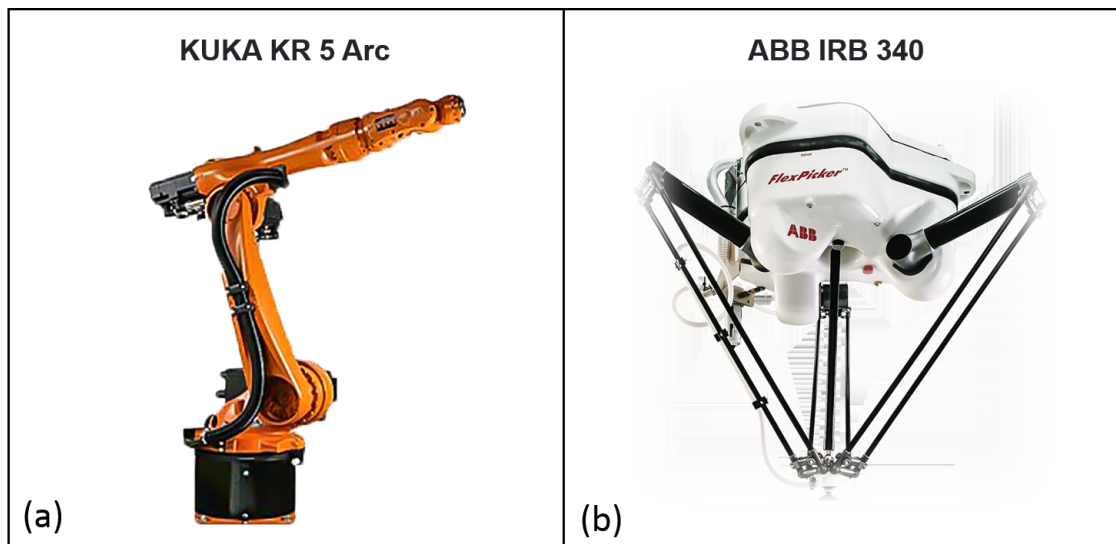
1	INTRODUÇÃO	29
1.1	Objetivos	32
1.2	Estrutura do texto	33
2	REDES NEURAIAS ARTIFICIAIS	35
2.1	Redes Alimentadas Adiante com Atraso de Tempo	42
2.2	Tipos de erro	43
3	MATERIAIS E MÉTODOS	45
3.1	Materiais	45
3.1.1	Histórico de desenvolvimento do protótipo	46
3.1.2	Comunicação e Instrumentação do protótipo atual	47
3.2	Métodos	51
3.2.1	Modelo Cinemático	51
3.2.2	Descrição do problema e metodologia de resolução	52
4	RESULTADOS E DISCUSSÃO	55
4.1	Obtenção dos Dados Experimentais	55
4.2	Tratamento dos Dados Experimentais	56
4.3	Treinamento das Redes Neurais	59
4.3.1	Redes de Múltiplas Camadas Alimentadas Adiante	59
4.3.1.1	Treinamento da RNA com uma trajetória	59
4.3.1.2	Treinamento da RNA com múltiplas trajetórias	63
4.3.2	Redes de Múltiplas Camadas Alimentadas Adiante Atrasadas no Tempo	64
4.3.2.1	Treinamento de TLFN focadas com Múltiplas Trajetórias	64
4.3.2.2	Treinamento de TLFN distribuídas com Múltiplas Trajetórias	65
4.4	Comparação dos resultados entre os diferentes tipos de RNAs	66
4.4.1	RNA_{22} aplicada à trajetória 23	66
4.4.2	RNA_{21} aplicada à trajetória 22 e 23	66
4.4.3	RNA_{20} aplicada à trajetória 21, 22 e 23	68
5	CONCLUSÃO	73
	REFERÊNCIAS	75

APÊNDICES	77
APÊNDICE A – MATRIZ DE ERRO (MSE) PARA TREINAMENTO COM UMA TRAJETÓRIA, RNA DE MÚLTIPLAS CAMADAS	80
APÊNDICE B – MATRIZ DE ERRO (MAPE) PARA TREINAMENTO COM UMA TRAJETÓRIA, RNA DE MÚLTIPLAS CAMADAS	86
APÊNDICE C – MATRIZ DE ERRO (MSE) PARA TREINAMENTO COM MÚLTIPLAS TRAJETÓRIAS, RNA DE MÚLTIPLAS CAMADAS	92
APÊNDICE D – MATRIZ DE ERRO (MAPE) PARA TREINAMENTO COM MÚLTIPLAS TRAJETÓRIAS, RNA DE MÚLTIPLAS CAMADAS	98
APÊNDICE E – MATRIZ DE ERRO (MSE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA FOCADA	104
APÊNDICE F – MATRIZ DE ERRO (MAPE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA FOCADA	110
APÊNDICE G – MATRIZ DE ERRO (MSE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA DISTRIBUÍDA	116
APÊNDICE H – MATRIZ DE ERRO (MAPE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA DISTRIBUÍDA	122

1 INTRODUÇÃO

Os manipuladores são constituídos por um conjunto de corpos denominados elos, que quando unidos por juntas adquirem a condição de cadeia cinemática, ou seja, cadeia cinemática pode ser interpretada como um conjunto de elos e juntas. Um manipulador de arquitetura cinemática serial apresenta uma única cadeia cinemática, e esta se encontra conectada à base. Um exemplo desse tipo de manipulador é apresentado na Fig. 1a, enquanto que um manipulador de arquitetura cinemática paralela é aquele que possui mais de uma cadeia cinemática conectada ao efetuador final, um exemplo deste tipo de manipulador é apresentado na Fig. 1b.

Figura 1 – Manipuladores



Fonte: (a) <https://www.robots.com/robots/kuka-kr-5-arc> e (b) <https://www.robots.com/robots/abb-irb-340>

Os manipuladores de arquitetura cinemática paralela, também conhecidos como manipuladores paralelos (*Parallel Machines* - PM) apresentam diversas vantagens em relação às arquiteturas tradicionais de robôs manipuladores (MERLET, 2005). Dentre essas vantagens pode-se citar a leveza, rigidez elevada, capacidade de carga, elevadas relações velocidades/aceleração e alta compactação. Com isso, esse tipo de arquitetura vem se mostrando uma ótima escolha, quando se trata de projeto de manipuladores de alto desempenho dinâmico e elevada eficiência energética (LI; BONE, 2001). Entretanto, como é de se esperar fazer uso dessas vantagens no projeto de manipuladores, também acarreta certas desvantagens, e essas tratam-se da presença de singularidades no espaço de trabalho (MERLET, 2005) e dificuldades com relação ao projeto de sistemas de controle

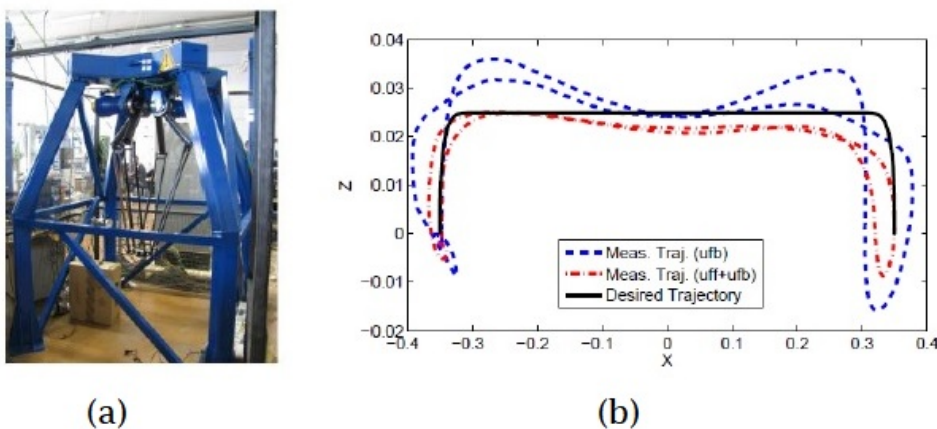
do mecanismo (PACCOT; ANDREFF; MARTINET, 2009).

Alguns trabalhos foram desenvolvidos com o objetivo de se obter um manipulador cinemático e paralelo que tivesse elevada capacidade dinâmica e alta eficiência energética, além de resolver os problemas mencionados acima. Os temas mais recorrentes nesses trabalhos serão apresentados adiante, e suas ideias principais podem ser listadas, abaixo:

- Redução da inércia dos componentes móveis;
- Uso de modelos cinemáticos e dinâmicos para a realização do controle;
- Uso de visão computacional;
- Uso de Redes Neurais Artificiais (RNAs).

Segundo os autores Nabat *et al.* (2005) e Zhang *et al.* (2015), o projeto de manipuladores de alto desempenho dinâmico e alta eficiência energética pode ser beneficiado pela redução da inércia de componentes móveis. Uma estratégia desse tipo foi adotada para o desenvolvimento de um manipulador paralelo de alto desempenho dinâmico, denominado Par2, construído no LIRMM/Montpellier. Nesse trabalho, dois controladores são projetados para o manipulador, sendo um PID e um PID+*feedforward* nas juntas ativas. Para a obtenção do sinal de *feedforward*, implementou-se um modelo multicorpos flexível, no entanto, esse modelo não apresentou um desempenho satisfatório. Ao implementar-se o PID+*feedforward*, ele obteve um resultado melhor para a mesma trajetória do primeiro, porém ainda insatisfatória (SILVA *et al.*, 2010). Na Figura 2 é apresentada uma imagem que sintetiza essa ideia.

Figura 2 – (a) Protótipo Par2 construído no LIRMM e (b) Comparação entre duas estratégias de controle.



Fonte: (SILVA *et al.*, 2010)

Utilizar de modelos cinemáticos e dinâmicos no sistema de controle é de extrema importância, uma vez que adicionando esses modelos à malha de controle, contribui-se para o sistema apresentar um alto desempenho dinâmico e uma elevada precisão. No entanto, quando o objeto de estudo são manipuladores paralelos, os modelos desses sistemas são muito complexos, e essa complexidade se torna ainda maior quando esses modelos são flexíveis. Abordando esse tipo de problema, foram encontrados trabalhos dos autores Zhang *et al.* (2015), Silva *et al.* (2010) e Wang *et al.* (2009). Os trabalhos possuem em comum, o fato de utilizar modelos dinâmicos de manipuladores paralelos com a finalidade de obter sinais de *feedforward* que auxiliam na precisão e no desempenho desse tipo de manipulador. Esses autores utilizaram como estratégia a implementação de modelos dinâmicos reduzidos para o cálculo das entradas necessárias para a atuação do manipulador que estava sendo estudado. A técnica de Parametrização Global, que leva em consideração termos não-lineares foi utilizada no trabalho de Silva *et al.* (2010), enquanto Zhang *et al.* (2015) e Wang *et al.* (2009) utilizaram a técnica dos Métodos dos Modos Assumidos. Esses modelos, quanto mais preciso forem, mais ajudam para que os sinais *feedforward* que são calculados contribuam efetivamente para o aumento do desempenho dinâmico do sistema.

Conforme já foi mencionado, os modelos cinemáticos e dinâmicos apresentam um certo nível de complexidade, buscando contornar esse problema foram desenvolvidos trabalhos que fazem o uso de visão computacional. Com isso é possível medir a posição do efetuador final de maneira direta, dessa forma são feitos projetos de sistemas de controle retroalimentados para manipuladores paralelos implementados no espaço das juntas. Paccot, Andreff e Martinet (2009), Colombo, Fontes e Silva (2019) e Yu *et al.* (2011) utilizaram métodos de visão computacional para a medição direta da postura do efetuador de manipuladores de elos rígidos. O trabalho de Paccot, Andreff e Martinet (2009) exigiu que a taxa de aquisição e processamento de imagens garanta a largura de banda do controlador, no entanto o trabalho realizado por Colombo, Fontes e Silva (2019) fez o uso de redundância de medidas para que esse tipo de exigência fosse aliviada. Por fim, a estratégia adotada por Yu *et al.* (2011) fez o uso de um controle descentralizado e não baseado em um modelo, para isso utilizou um sensor de deslocamento atuado por cabo para a medição atual de um manipulador paralelo flexível 3RRR.

No trabalho de Colombo, Fontes e Silva (2019), implementou-se uma estratégia de controle por torque calculado no espaço de tarefa, usando servo visão para mapear a postura do efetuador final. A câmera utilizada foi capaz de medir a postura do efetuador final de maneira adequada. A câmera foi suficiente para adquirir 20 imagens por segundo (20 fps), no entanto essa taxa de aquisição não se mostrou adequada para fornecer a largura de banda necessária para realizar o projeto de um controlador para um manipulador de alta capacidade dinâmica. Então, fez-se necessário a utilização de estratégias de controle com diferentes níveis de autoridade e redundância de medidas, para isso fez-se o uso de *encoder* e câmera, a estratégia se mostrou eficaz para o controle de um manipulador

paralelo de elos rígidos. Este trabalho deixou claro que fazer medidas diretas e indiretas do efetuador final é essencial para o projeto de um controlador para um manipulador de alto desempenho dinâmico.

Outros estudos que tentam contornar o problema acarretado pela complexidade dos modelos cinemático e dinâmico dos manipuladores são os estudos que fazem uso de Redes Neurais Artificiais (RNAs). Nestes trabalhos, os autores procuram deduzir os modelos de estudo com base no aprendizado de máquina. No trabalho de Bidokhti e Enferadi (2015) é tratado o problema da cinemática direta de um manipulador robótico planar $3R\underline{R}R$, esse problema é resolvido usando dois métodos diferentes de RNAs, um trata-se de uma Rede Neural de de Múltiplas Camadas, enquanto o outro uma Rede Neural de Base radial. Ambas as redes utilizaram como conjunto de dados de treinamento a resolução da cinemática inversa do robô, no trabalho a eficácia de solução é demonstrada comparando um caminho simulado em espiral com um caminho real, e para ambas as redes o erro total durante o caminho simulado foi aceitável, o que confirma a confiabilidade dos métodos.

As aplicações de RNAs também são feitas para se obter a cinemática inversa de robôs. O trabalho de Moori, Khoramdel e Moosavian (2019) que trata-se de um robô paralelo $3R\underline{R}R$ esférico tem como um de seus objetivos fazer cálculos rápidos e superar as incertezas de fabricação presentes no robô. Para isso as equações cinemáticas inversas presentes no problema são resolvidas por uma RNA de Múltiplas Camadas. Os autores Csiszar, Eilers e Verl (2017) realizaram um trabalho propondo uma abordagem supervisionada para resolver o problema da cinemática inversa e da calibração. Neste trabalho foi proposto que ao invés de se criar um modelo ideal para cada robô, e depois calibrar cada um individualmente, treiná-lo através de uma RNA para aprender a função da cinemática inversa, uma vez que essa já viria incluindo os erros devido às tolerâncias de fabricação e/ou montagem.

No trabalho de Félix (2021) é feito o estudo de um manipulador planar $3R\underline{R}R$ de elos flexíveis, nesse estudo dados experimentais são extraídos usando encoders, extensômetros e uma câmera, durante a execução de tarefas pré-determinadas. Para estimar a pose do efetuador final, neste trabalho, é feito o uso de três RNAs de Múltiplas Camadas, uma para cada coordenada da posição do efetuador final. Os resultados obtidos foram satisfatórios.

1.1 Objetivos

O presente trabalho tem como objetivo obter um modelo por intermédio de RNAs que auxilie na construção de sistemas de controle para o manipulador paralelo flexível $3R\underline{R}R$. Espera-se que esse modelo funcione como um estimador que seja capaz de prever a postura do efetuador final, a ponto de que obtido esse modelo seja possível dispensar o uso da câmera para a realização de medições da posição do efetuador final. Para a obtenção desse modelo é feito o uso de encoders, câmera e extensômetros. Os dados dos encoders

e dos extensômetros funcionam como entrada para o modelo, enquanto que os dados da câmera como os dados de saída para o modelo. Afim de alcançar esse objetivo é necessário a realização dos seguintes objetivos secundários, abaixo:

- Realizar coleta de dados do manipulador paralelo flexível 3RRR, presente no Laboratório de Dinâmica da EESC, para produção de um dataset;
- Utilizar Redes Neurais Artificiais (RNAs) (MLP, TLFN focada e TLFN distribuída) para aplicar as trajetórias do manipulador;
- Encontrar uma RNA que seja suficiente para modelar qualquer trajetória do 3RRR, dados os dados de entrada.

1.2 Estrutura do texto

O presente trabalho se divide da seguinte forma:

- O Capítulo 2 apresenta uma breve revisão sobre RNAs para facilitar o entendimento do trabalho;
- O Capítulo 3 apresenta os materiais e os métodos utilizados no projeto, além de fazer uma descrição do problema em questão;
- O Capítulo 4 descreve como foram adquiridos os dados utilizados no trabalho, como foi o seu tratamento e quais foram os resultados obtidos após realizar os treinamentos com as RNAs;
- O Capítulo 5 apresenta as conclusões do trabalho, além de propor sugestões para trabalhos futuros.

2 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais (RNAs) tem como inspiração a estrutura e o funcionamento do sistema nervoso, buscando com isso a capacidade de simular a forma como o cérebro humano adquire conhecimento (FACELI, 2021). Segundo Haykin (2009) uma rede neural trata-se de um processador maciçamente paralelamente distribuído que possui uma unidade de processamento simples e tem como função natural armazenar o conhecimento adquirido através da experiência e torná-lo disponível para uso. Ele afirma que uma RNA se assemelha ao cérebro em dois aspectos, sendo que o primeiro é relativo ao fato da rede adquirir conhecimento do ambiente a sua volta, a partir de um processo de aprendizagem. O segundo aspecto está relacionado ao fato da RNA possuir forças de conexão entre os neurônios que são conhecidas como pesos sinápticos, utilizadas para armazenar o conhecimento adquirido.

A RNA possui como unidade básica uma estrutura chamada de neurônio que é fundamental para o seu funcionamento. Na Fig. 3a está presente em forma de diagrama de blocos um modelo do neurônio. Esse modelo é composto basicamente por três elementos, segundo Haykin (2009):

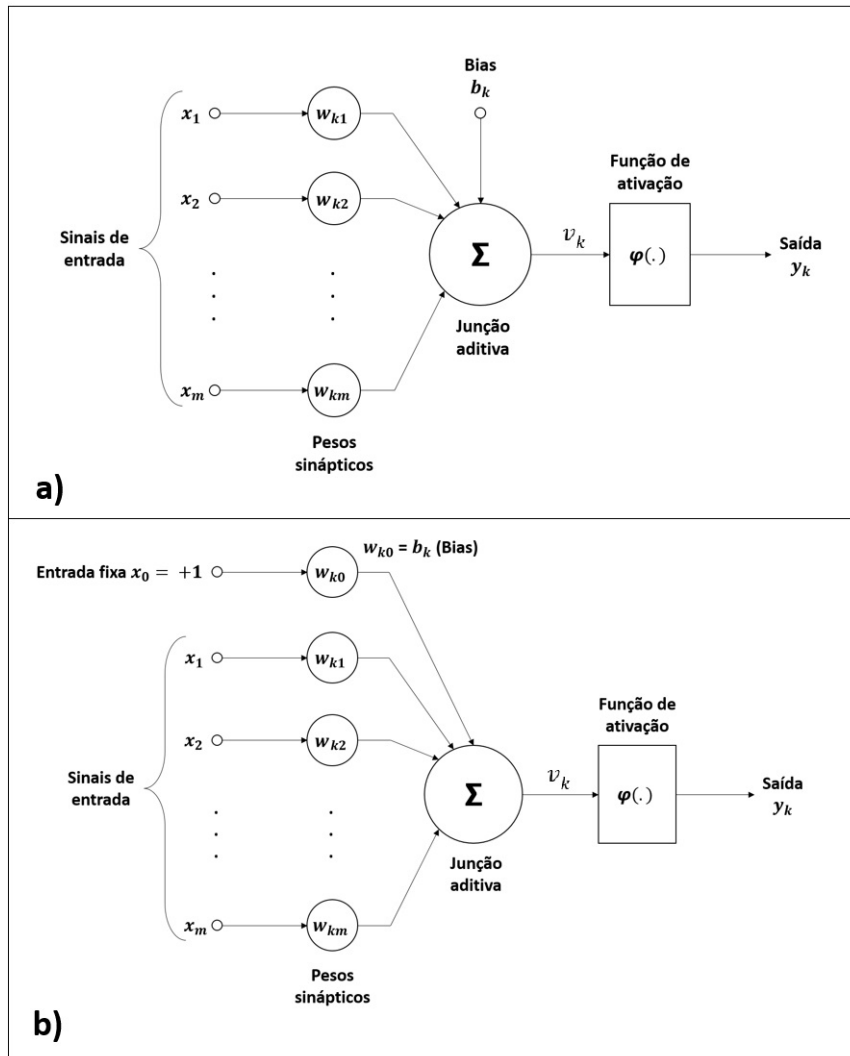
- Um conjunto de sinapses ou elos de conexão: cada uma possui um peso ou força própria. Pode-se dizer que uma entrada x_j na sinapse j que está presente em um neurônio k é multiplicada pelo peso sináptico w_{kj} .
- Um somador: os sinais de entrada são ponderados pelas respectivas sinapses do neurônio, o somador é responsável por somá-los. Essas operações constituem um combinador linear.
- Uma função de ativação: restringe a amplitude da saída de um neurônio, ou seja, o intervalo permissível de amplitude do sinal de saída é limitado a um valor finito. Normalmente, o intervalo normalizado de saída do neurônio pode ser escrito como o intervalo unitário fechado $[0,1]$ ou de uma maneira alternativa $[-1,1]$.

Ao observar a Fig. 3a é possível perceber a existência de um *bias* que é aplicado externamente, representado por b_k . A presença do *bias* tem como efeito aumentar (*bias* positivo) ou diminuir (*bias* negativo) a entrada líquida de uma função de ativação.

Matematicamente, é possível equacionar um determinado neurônio k , através do par de equações:

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (2.1)$$

Figura 3 – Duas formas de representar um neurônio artificial



Fonte: adaptado (HAYKIN, 2009)

$$y_k = \phi(u_k + b_k) \quad (2.2)$$

Na Eq. 2.1 x_1, x_2, \dots, x_m tratam-se dos sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ; a saída do combinador linear devido aos sinais de entrada é u_k ; o bias é b_k ; a função de ativação é denotada por $\phi(\cdot)$; e y_k é o sinal de saída do neurônio. O bias possui o efeito de aplicar uma transformação afim à saída u_k , ao somar o *bias* com a saída, temos o potencial de ativação v_k , como mostrado por:

$$v_k = u_k + b_k \quad (2.3)$$

O *bias* pode ser modelado como uma entrada (Fig. 3b). Para isso adicionamos uma

nova sinapse ao sistema $x_0 = +1$, e um peso é atribuído a essa nova entrada $w_{k0} = b_k$. As Eq. 2.1 e 2.2 podem ser reescritas como segue:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.4)$$

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

As funções de ativação $\phi(\cdot)$ presentes nos neurônios da RNA possuem três tipos básicos (HAYKIN, 2009):

- Função de Limiar: Esse tipo de função pode assumir apenas dois valores 0 ou 1, assume valor 0 se $v < 0$ e 1 se $v \geq 0$. Essa função pode ser visualizada na Fig. 4a.
- Função Linear por Partes: Essa função assume valores entre 0 e 1 para $\frac{1}{2} > v > -\frac{1}{2}$, assume valor 1 para $v \geq \frac{1}{2}$ e valor 0 para $v \leq -\frac{1}{2}$. Essa função pode ser visualizada na Figura 4b.
- Função Sigmóide: Este tipo de função possui um formato que se assemelha a letra "s", sendo de longe a forma mais comum de função de ativação, usada no projeto de RNA. Trata-se de uma função estritamente crescente que exhibe um balanceamento adequado entre comportamento linear e não-linear. Como exemplo de função sigmóide (Fig. 4c), pode-se citar a função logística, essa função possui um parâmetro a que é denominado parâmetro de inclinação, responsável por fazer a função apresentar diferentes inclinações. Uma outra forma correspondente de uma função sigmóide é uma função tangente hiperbólica, ambas as funções apresentadas aqui são diferenciáveis e apresentam saídas no espectro contínuo, as duas equações são apresentadas, abaixo:

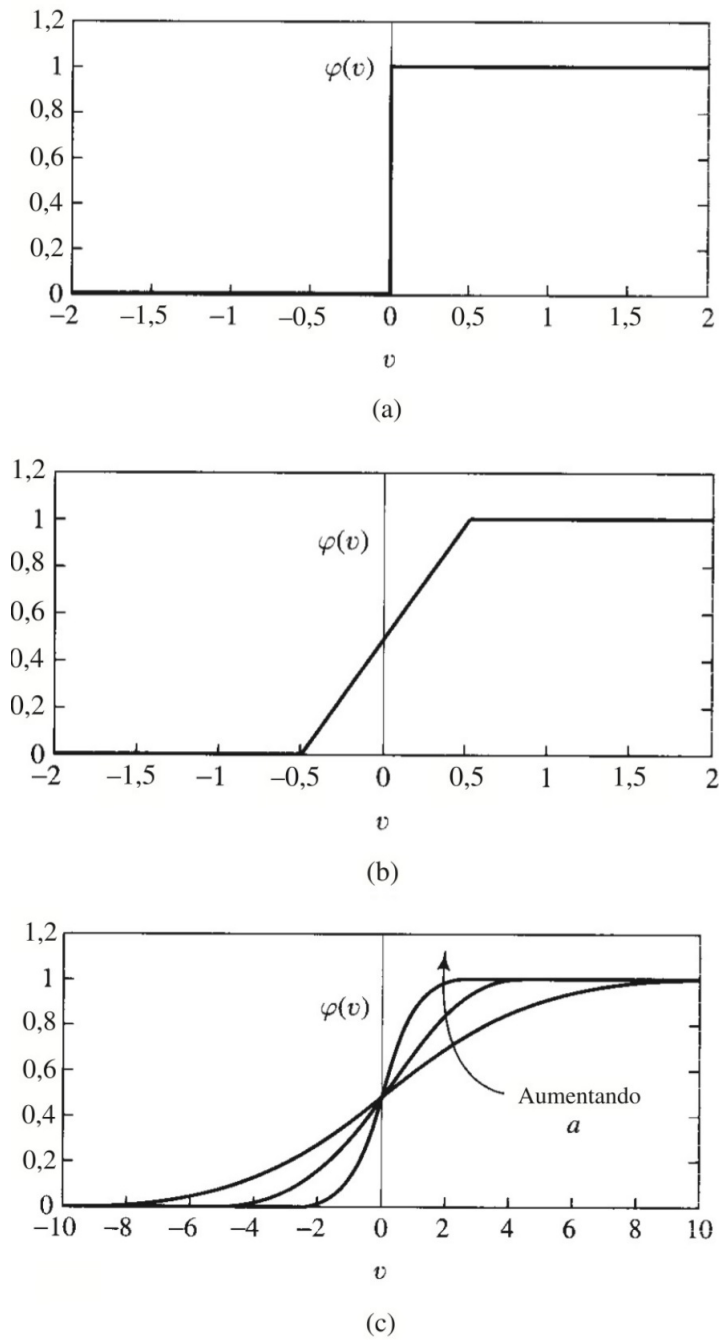
$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (2.6)$$

$$\phi(v) = \tanh(v) \quad (2.7)$$

Além desses tipos de funções de ativação existem outros tipos, tais como: ReLU, Leaky ReLU, PReLU, ReLU6, ELU, SELU, Swish e Mish. Detalhes sobre essas funções são apresentados no artigo de Rasamoelina, Adjailia e Sinčák (2020).

Quando se trata de RNAs, dois princípios básicos da construção dessas redes devem vir a mente: arquitetura e aprendizado. A arquitetura de uma RNA diz respeito ao tipo, número de unidades de processamento, e por fim a forma com que os neurônios estão

Figura 4 – Funções de ativação



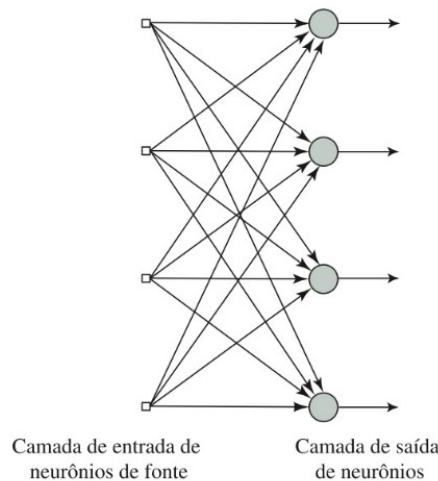
Fonte: adaptado (HAYKIN, 2009)

ligados, enquanto o aprendizado está relacionada às regras que são utilizadas para o ajuste de pesos nas RNAs, e qual a informação é utilizada pelas regras (FACELI, 2021).

Em relação a arquitetura de RNAs pode-se geralmente identificar três tipos (HAYKIN, 2009):

- **Redes Alimentadas Adiante com Camada Única:** Essa é a forma mais simples de uma rede de camadas, esse tipo de rede tem esse nome justamente por ser organizada em camadas. Em uma rede de camada única há uma camada de entrada de nós de fonte que é conectada a uma camada de saída de neurônios (nós computacionais), o contrário não acontece. É dito que a rede é estritamente do tipo alimentada adiante ou acíclica. Um fato importante de frisar aqui é que apesar da rede possuir duas camadas (nós de fonte e nós computacionais) ela é nomeada como de camada única, isso acontece pelo fato dos nós de fonte não realizarem nenhum tipo de computação, dessa forma ela não é levada em consideração. Esse tipo de RNA pode ser visualizado na Figura 5.

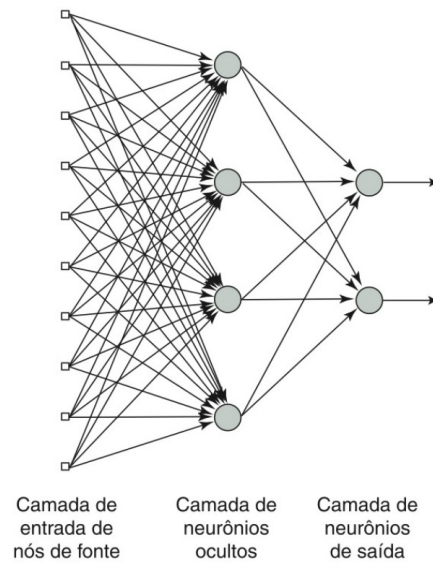
Figura 5 – Rede Alimentada Adiante com Camada Única



Fonte: adaptado (HAYKIN, 2009)

- **Redes Alimentadas Diretamente com Múltiplas Camadas (*multilayer perceptron* - MLP):** Esse tipo de rede se difere da anterior pelo fato de possuir uma ou mais camadas ocultas, os nós computacionais dessas camadas são chamados de neurônios ocultos ou unidades ocultas. Eles têm como função intervir de uma maneira útil entre a entrada e a saída de uma RNA, com uma ou mais camadas desses neurônios é possível fazer com que a rede extraia estatísticas de ordem elevada. Nessa arquitetura de rede os nós de fonte são ligados a primeira camada oculta, as saídas dessa camada são utilizadas de entrada para a próxima camada, e assim sucessivamente até chegar na camada de saída. Os sinais de entrada dos neurônios em cada camada de rede provem da camada precedente, os sinais de saída da última camada da rede correspondem a resposta global da rede. Esse tipo de RNA pode ser visualizado na Figura 6.
- **Redes Recorrentes:** Esse tipo de rede se difere de uma rede alimentada adiante

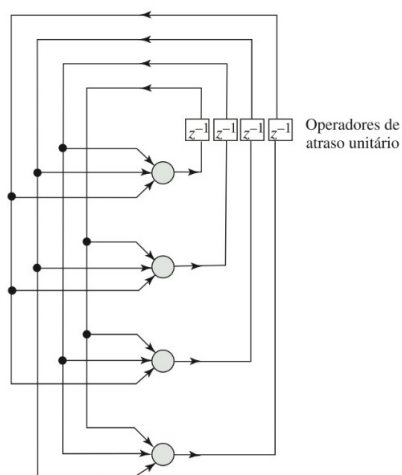
Figura 6 – MLP



Fonte: adaptado (HAYKIN, 2009)

porque possui no mínimo um laço de retroalimentação, e podem ter de uma a várias camadas. Quando elas possuem uma camada única, a saída de um neurônio pode realimentar os outros neurônios da mesma camada. Além disso, um neurônio pode realimentar ele mesmo, esse processo se chama auto-realimentação. Esse tipo de RNA pode ser visualizado na Figura 7.

Figura 7 – Rede Recorrente



Fonte: adaptado (HAYKIN, 2009)

Dos tipos de arquitetura de rede aqui mencionados, um tipo que certamente não

foi utilizado neste projeto é o dito como Redes Alimentadas Adiante com Camada Única, pois segundo Faceli (2021) esse tipo de rede não serve para resolver problemas não-lineares e o problema de estudo desse projeto é de natureza não-linear.

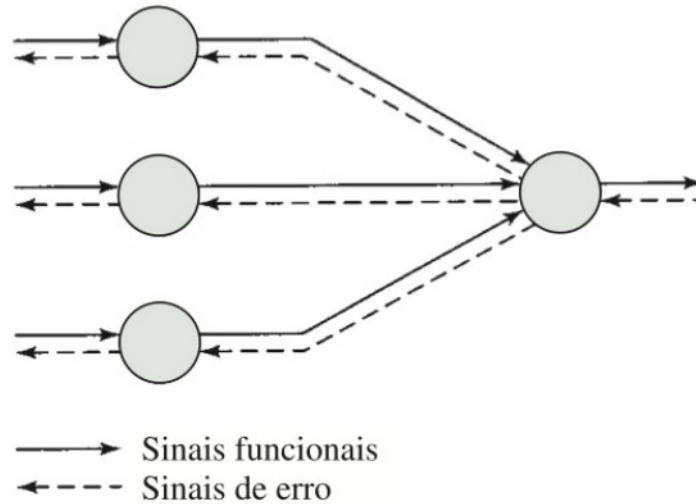
Definida a arquitetura da rede é necessário fazê-la aprender. Segundo Haykin (2009) o aprendizado é feito por meio dos ajustes de parâmetros de uma RNA, principalmente pela definição dos pesos que estão associados às conexões da rede que fazem com que o modelo obtenha melhor desempenho. Algoritmos que executam essa função podem ser chamados de algoritmos de treinamento, esses são formados por um conjunto de regras bem específicas que ditam como e quando devem ser alterados os valores de cada peso. Esse tipo de algoritmo pode ser dividido em quatro grupos:

- Correção de erro: Os erros cometidos pela rede são reduzidos, para isso procura-se ajustar os pesos da RNA. Esse tipo é geralmente utilizado em aprendizado supervisionado;
- Hebbiano: Esse tipo é usado frequentemente em aprendizado não-supervisionado, aqui faz-se o uso da regra de Hebb que sugere que se dois neurônios se encontram simultaneamente ativos, a conexão entre eles deve ser reforçada;
- Competitivo: Esse tipo é usado para aprendizado não-supervisionado. Neste tipo é feita uma competição entre neurônios, assim é definido qual ou quais neurônios devem ter seus pesos ajustados;
- Termodinâmico: Tratam-se de algoritmos estocásticos que possuem seus princípios baseados na metalurgia.

Neste trabalho, o aprendizado é supervisionado, sendo assim foi utilizado um algoritmo de aprendizagem de correção de erro. O algoritmo de aprendizado desse tipo mais utilizado é o *Backpropagation*. Segundo Parker (1987), dois tipos de sinais são identificados na rede para esse tipo de treinamento. Um é dito como Sinal Funcional, sendo este um sinal de entrada (estímulo), este incide no terminal de entrada da rede e propaga para frente (neurônio por neurônio), até emergir no terminal de saída da rede. O outro é dito como Sinal de Erro, este se origina em um neurônio de saída da rede e se propaga para trás (camada por camada). Na Figura 8 são ilustrados os dois tipos de sinais, em um perceptron de múltiplas camadas.

O Sinal Funcional ao atravessar a rede gera uma resposta com base nos seus pesos sinápticos atuais, esta fase é conhecida como “propagação adiante” (*forward*). A saída encontrada é comparada com uma resposta desejada, e o Sinal de Erro é propagado para trás corrigindo os pesos sinápticos, esta fase é conhecida como “propagação reversa” (*backward*). Este processo é repetido até que o erro na saída esteja dentro da faixa

Figura 8 – Ilustração do fluxo de sinais básicos em um perceptron de múltiplas camadas



Fonte: Adaptado (HAYKIN, 2009)

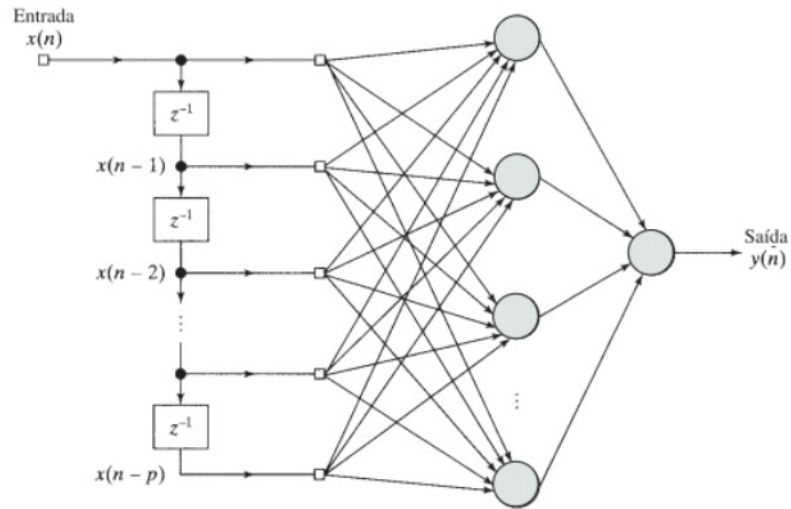
imposta no início do processo de treinamento. Os pesos sinápticos finais obtidos pela rede, equivalem a rede treinada, neste ponto espera-se que ao entrar com uma determinada entrada, obtenha-se a resposta desejada.

2.1 Redes Alimentadas Adiante com Atraso de Tempo

Quando os problemas em questão apresentam variações temporais, uma forma interessante de se utilizar as RNAs é adicionar atrasos de tempo. Uma maneira de se trabalhar com esses atrasos é utilizar as Redes Alimentadas Adiante Focadas Atrasadas no Tempo (*focused time lagged feedforward network* - TLFN focada) e outra as Redes Alimentadas Adiante Atrasadas no Tempo Distribuídas (*distributed time lagged feedforward network* - TLFN distribuída).

Uma TLFN focada concentra todos os seus atrasos de tempo na camada de entrada, enquanto que uma TLFN distribuída tem os seus atrasos distribuídos ao longo da rede. Um esquemático de uma TLFN focada pode ser visualizado na Fig. 9 e 25, um exemplo de TLFN distribuída pode ser visualizado na Fig. 26. Segundo Haykin (2009) é através da incorporação do tempo em uma RNA que é possível suprir uma rede estática, como a rede MLP, com características dinâmicas. As TLFNs utilizam como base um RNA estática, como as MLPs, por exemplo.

Figura 9 – TLFN focada



Fonte: Adaptado (HAYKIN, 2009)

2.2 Tipos de erro

Para verificar as performances das RNAs são utilizados erros. Nesse trabalho foram utilizados erros de dois tipos: o erro quadrático médio (*mean square errors* - MSE) e a média do erro percentual absoluto médio (*mean absolute percentual error* - MAPE). O MSE é dado pela eq. 2.8, enquanto o MAPE é dado pela eq. 2.9.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (2.8)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|(y_i - y'_i)|}{|y_i|} \quad (2.9)$$

Para a eq. 2.8 e eq. 2.9, y_i trata-se do valor aproximado, y'_i o valor absoluto e n do número de amostras. Como o problema em questão apresenta 3 saídas de dados, o erro utilizado para avaliar o desempenho foi uma média simples tanto do valor do MAPE para as três saídas como do valor do MSE.

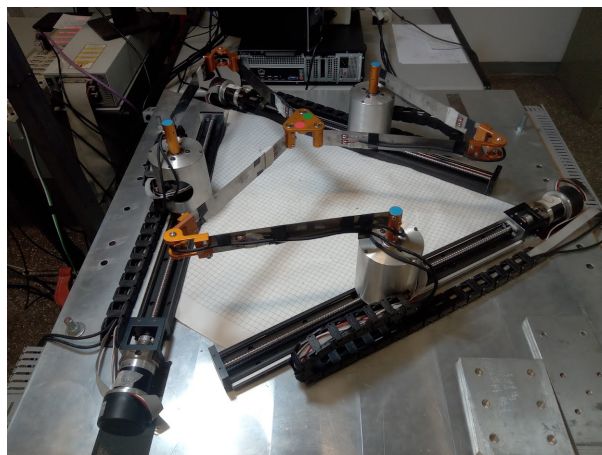
3 MATERIAIS E MÉTODOS

Neste capítulo são discutidos os materiais e os métodos envolvidos no presente trabalho. Em materiais é feita uma breve abordagem sobre o histórico de desenvolvimento do protótipo em estudo, além da comunicação e instrumentação do protótipo. Em métodos são mostradas as principais teorias envolvidas no trabalho.

3.1 Materiais

O manipulador em estudo trata-se do 3(P)RRR (Fig. 10), presente no laboratório de Dinâmica da EESC-USP. Este é um manipulador paralelo planar de cadeia cinemática fechada que possui três cadeias cinemáticas idênticas que são ligadas ao efetuador final, sendo que cada uma dessas cadeias possui quatro juntas, uma delas prismática e ativa (P) e as três outras (RRR) rotacionais, sendo a junta mais distante do efetuador final, ativa, e as outras duas passivas. A presença de sublinhado abaixo das letras indica que a junta é ativa, enquanto sua ausência indica que a junta é passiva. Para esse trabalho, apesar do manipulador em questão poder ser utilizado como 3(P)RRR, isso não é feito, uma vez que as juntas prismáticas estão bloqueadas. Dessa forma para esse trabalho o manipulador em questão pode ser tratado como 3RRR. Nas subseções a seguir são tratados, o histórico de desenvolvimento do protótipo e a comunicação e instrumentação necessárias para se trabalhar com um manipulador flexível.

Figura 10 – Protótipo atual


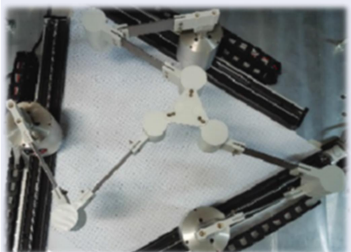



Fonte: Próprio Autor

3.1.1 Histórico de desenvolvimento do protótipo

O protótipo do manipulador 3RRR inicialmente era constituído por elos rígidos, o que impossibilitava a modelagem de um manipulador flexível e ocasionava um aumento de inércia do sistema, dificultando a aplicação de sistemas de controle. Com a presença desses empecilhos, modificações foram feitas no protótipo, os elos rígidos foram substituídos por elos de PLA, fabricados por meio do processo de impressão 3D. Ocorreu com essa modificação, uma redução da inércia do sistema e um aumento da flexibilidade, no entanto, houve o aparecimento de folga no manipulador. Atualmente, trocou-se os elos flexíveis de PLA por réguas metálicas de alumínio, com isso o sistema adquiriu a inércia e a flexibilidade adequada para a realização do estudo. Na Fig. 11 a evolução do protótipo é sintetizada.

Figura 11 – Evolução do protótipo

Evolução do protótipo	Modificações	Peculiaridades
	(Não se aplica)	<ul style="list-style-type: none"> • Impossibilidade de modelar um manipulador flexível, devido a presença de elos rígidos; • Inércia elevada que prejudica a implementação de sistemas de controle.
	<ul style="list-style-type: none"> • Troca de elos rígidos por elos flexíveis de PLA; • Troca das juntas rotacionais por juntas de PLA. 	<ul style="list-style-type: none"> • Apresenta flexibilidade, o que possibilita a modelagem de um manipulador flexível, porém há aparecimento de folga; • Redução da inércia.
	<ul style="list-style-type: none"> • Troca de elos flexíveis de PLA por régua metálicas flexíveis; • Troca das juntas rotacionais de PLA por juntas de alumínio. 	<ul style="list-style-type: none"> • Apresenta flexibilidade, o que possibilita a modelagem de um manipulador flexível; • Não há folga; • Inércia adequada.

Fonte: Próprio autor

3.1.2 Comunicação e Instrumentação do protótipo atual

Nesta seção são descritos a comunicação e instrumentação do protótipo atual, no Quadro 1 é apresentada a descrição dos componentes aqui mencionados e na Fig. 12 é apresentado o aparato experimental utilizado com a indicação de cada um dos seus componentes. Como este trabalho utilizará apenas o modo 3RRR do protótipo, os únicos motores ativos são $M1$, $M2$ e $M3$ (Fig. 13). Estes motores são os responsáveis por fornecer movimento às articulações motoras, e para o seu correto funcionamento eles possuem controladores.

Quadro 1 – Descrição dos componentes do protótipo

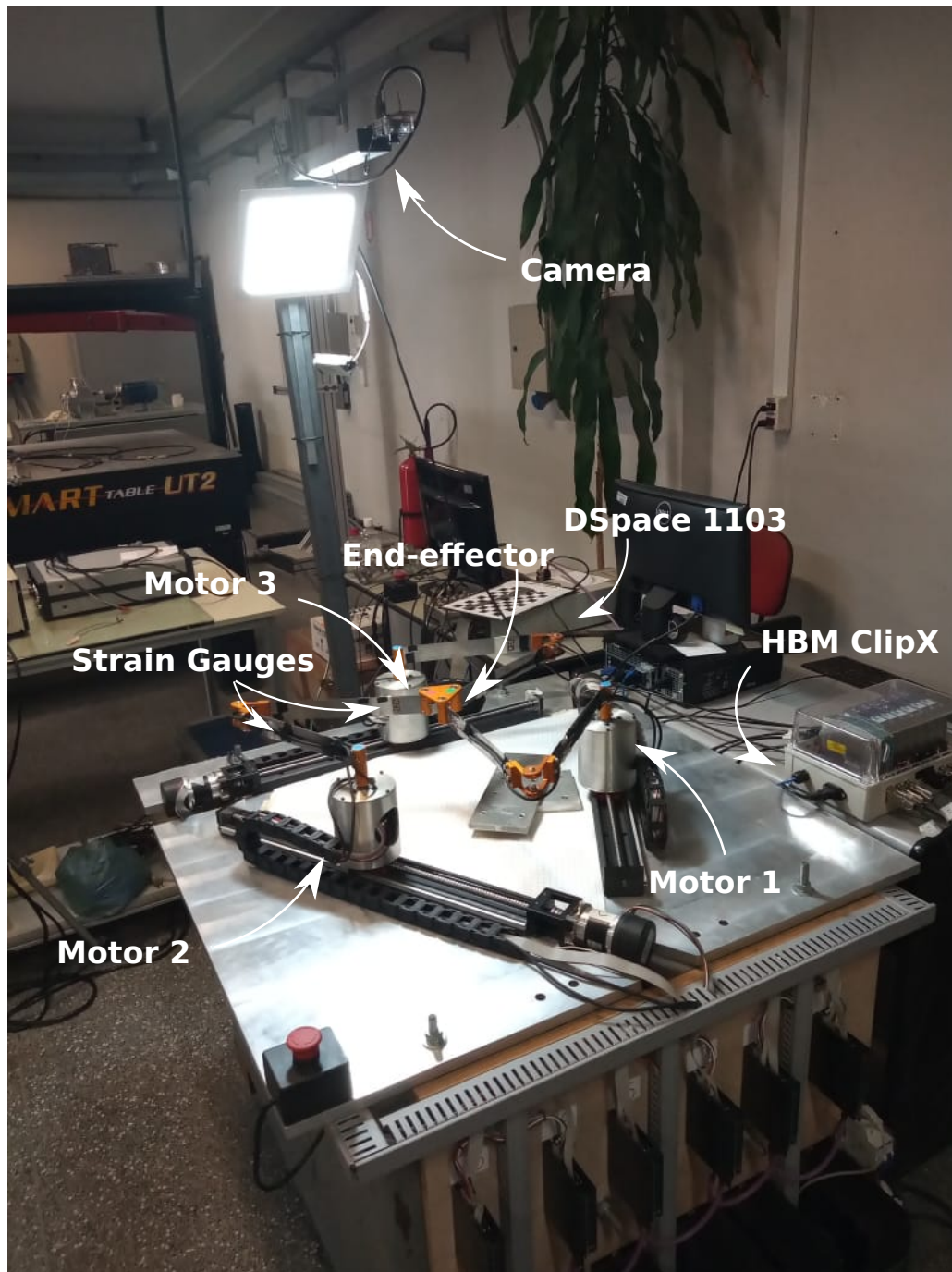
Componente	Descrição
Motores ($M1$, $M2$ e $M3$)	<i>EC60 flat da Maxon</i> sem escova, com 100W de potência e um valor nominal de corrente de 2,3A, acoplados a redutores planetários <i>GP52C</i> com uma redução de 3,5 : 1, proporcionando uma rotação nominal de 1200RPM e torque nominal de 0,82Nm.
Controlador	Maxon <i>EPOS250/5</i> , alimentação de até 50Vdc e corrente de 5A.
Placa de aquisição de dados	Placa do tipo ECU (<i>Electronic Controller Unity</i>), modelo DSPACE 1103
Extensômetros	HBM 1-LA11K3/350-E
Amplificador de sinal	HBM BM40
Câmera	<i>oCam – 5CRO – U</i> , interface USB 3.0, sensor de imagem CMOS e <i>rolling shutter</i> , com resolução de 640x480 pixels e <i>framerate</i> máximo de 120 fps

Fonte: Próprio Autor

Na Fig. 13 pode ser visto um diagrama da comunicação. A comunicação dos motores com a placa de aquisição de dados é feita via protocolo CAN, com uma taxa de transmissão de 250 kbit/s. Este protocolo é responsável tanto pelo acionamento do motor, quanto pela aquisição de dados do codificador. Para medida de tensão são utilizados extensômetros (Fig. 14a) em uma configuração de ponte cheia (Fig. 14c), esses são anexados a cada elo do manipulador, conforme a Fig. 14b para realizar as medidas de tensão relativas a deformação. Os sinais colhidos pela ponte de extensômetros precisam ser lidos pelas entradas A/D da placa de aquisição de dados, no entanto, para que isso seja possível precisam passar por um amplificador de sinal.

O manipulador atual possui marcações, as quais com o auxílio da câmera são responsáveis pelo monitoramento da posição do efetuador final. Para que isso seja possível, as marcações em azul (Fig. 10) feitas nos atuadores são utilizadas para encontrar o centro do espaço de trabalho. Enquanto que as marcações em vermelho e verde (Fig. 10) estão relacionadas a encontrar a posição (x, y, α) em relação ao centro, isso é feito com uma

Figura 12 – Aparato Experimental

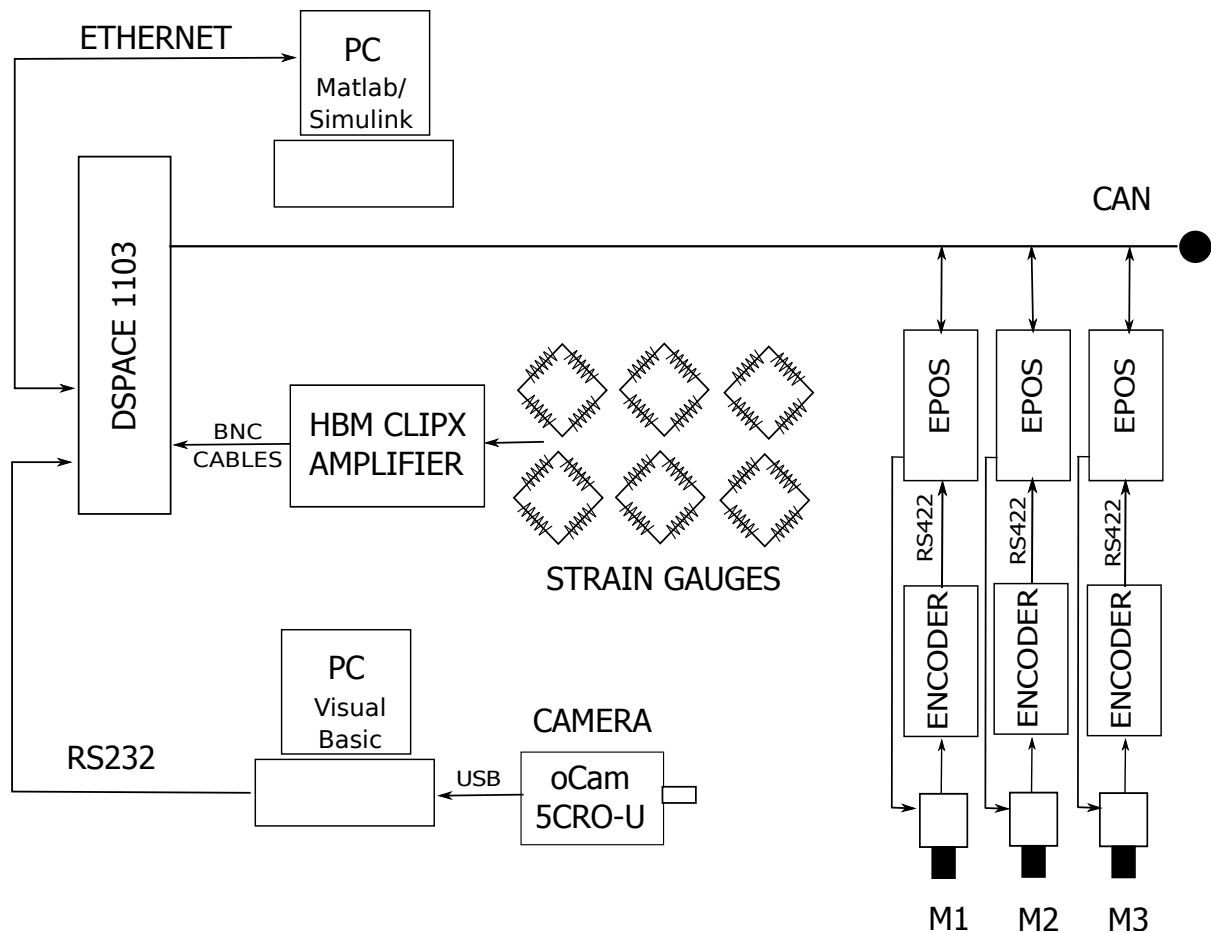


Fonte: Silva, 2023

técnica de Visão Computacional. Na Figura 15 é possível visualizar um esquemático ilustrando o processo. Nos itens abaixo estão listados os principais pontos desse processo:

- (A) Tirar várias fotografias da imagem do xadrez;
- (B) Extrair as matrizes dos parâmetros extrínsecos e intrínsecos;

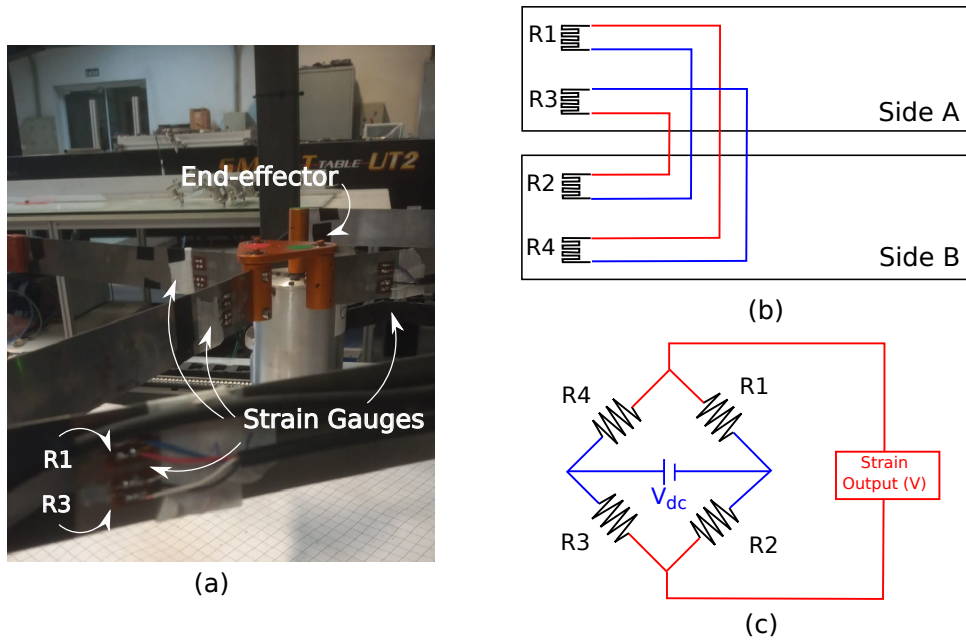
Figura 13 – Instrumentação e seu esquema de comunicação



Fonte: Silva, 2023

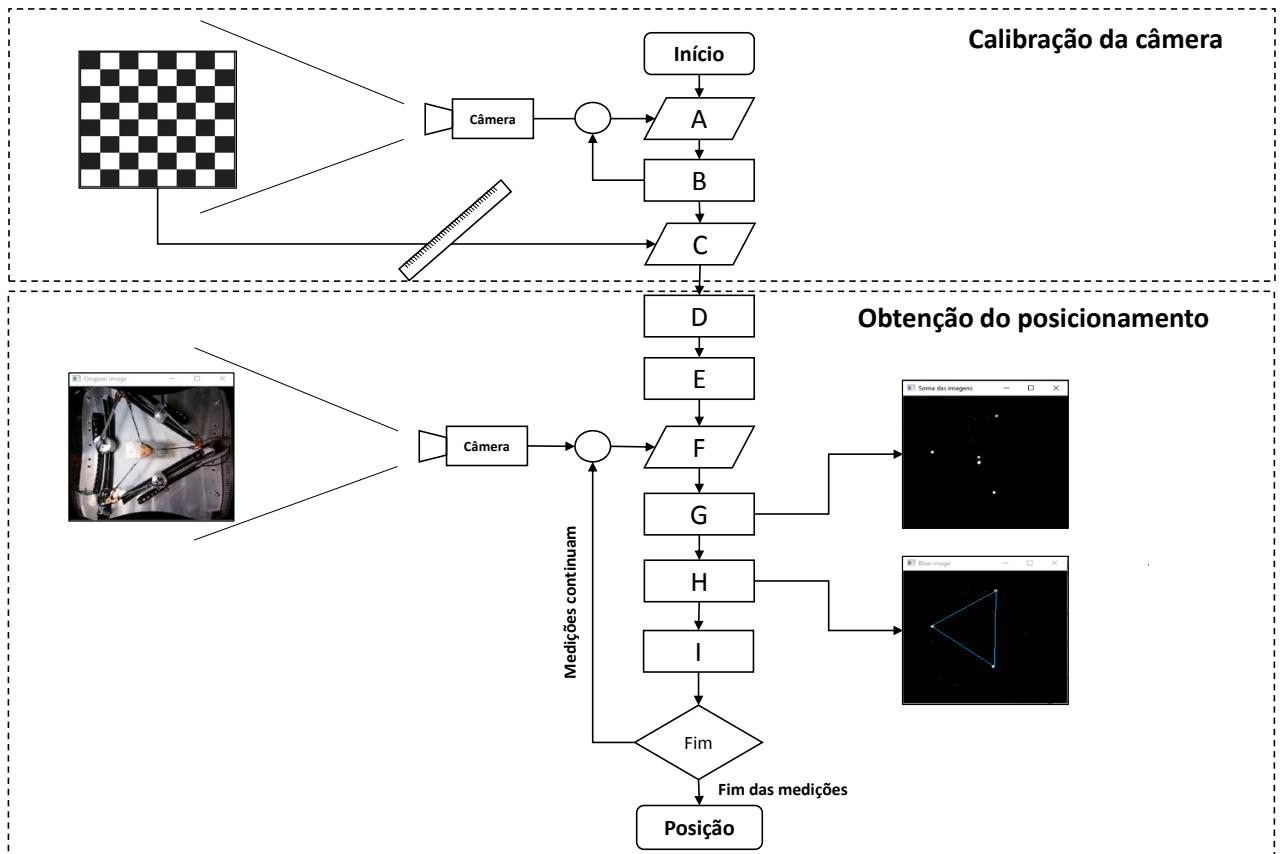
- (C) Medir tamanho dos quadrados do xadrez manualmente, e encontrar a relação real com a imagem, entrando no sistema com a medida encontrada manualmente;
- (D) carregar OpenCV;
- (E) Habilitar USB da câmera;
- (F) Adquirir imagem da câmera;
- (G) Aplicar filtros para encontrar as cores vermelho, verde e azul (decomposição em RGB ou HSM);
- (H) Encontrar o centro do contorno azul esse é o zero;
- (I) Ligar os pontos verde e vermelho, encontrar o centro, e conseqüentemente, as coordenadas do efetuador final.

Figura 14 – Extensômetros



Fonte: Silva, 2023

Figura 15 – Metodologia da medida, técnica de medição



Fonte: Próprio Autor

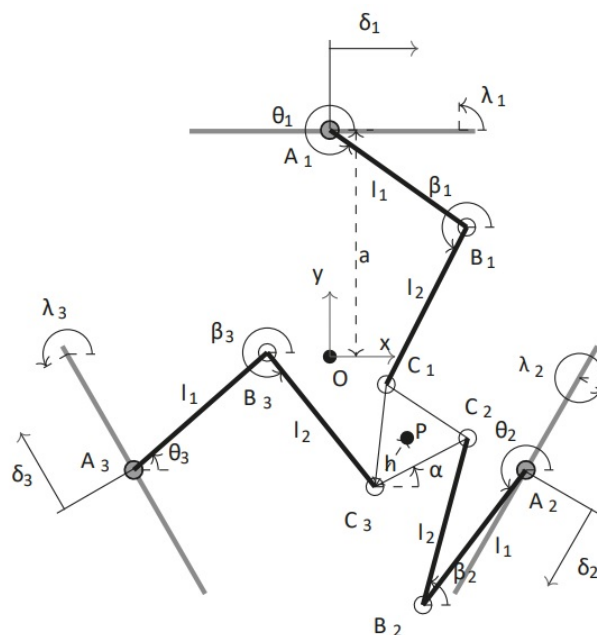
3.2 Métodos

Nesta seção são apresentados os métodos que são utilizados durante a pesquisa.

3.2.1 Modelo Cinemático

O modelo de um manipulador paralelo de elos rígidos pode ser visualizado na Figura 16. A origem do sistema de coordenadas $[x, y, z]$, coincide com centro geométrico O do triângulo equilátero que compreende a área de trabalho do manipulador. Para o braço $A_i B_i$, o comprimento de cada elo é definido por l_1 e o ângulo de rotação por θ_i , enquanto para o braço $B_i C_i$ o comprimento do elo é l_2 e o ângulo de rotação é β_i , sendo para ambos os casos $i = 1, 2, 3$ que corresponde a cada uma das cadeias cinemáticas do manipulador. O efetuador, também trata-se de um triângulo equilátero, ele possui um centro P que coincide com seu centro geométrico, e sua translação é definida pela posição do ponto P em relação ao ponto O , através do sistema de coordenadas $[x, y, z]$, sendo $z = 0$ sempre, por se tratar de um manipulador planar, enquanto sua rotação é medida pelo ângulo α entre o lado $C_2 C_3$ e o eixo x de referência. Dessa forma, a posição do efetuador final pode ser completamente definida por x, y, α . O comprimento entre a origem do sistema de coordenadas O_{xyz} e as juntas ativas A_i é definida por a , h trata-se da distância entre o ponto P e as juntas passivas C_i , e por fim λ_i trata-se dos ângulos em relação à normal de cada lado do triângulo formado pela área de trabalho.

Figura 16 – Modelo do manipulador



Segundo Fontes (2015), o manipulador 3RRR com elos rígidos pode ser modelado cinematicamente considerando a restrição geométrica $\|\overrightarrow{B_i C_i}\| = \|\mathbf{r}_{C_i} - \mathbf{r}_{B_i}\| = l_2$, onde \mathbf{r}_{C_i} e \mathbf{r}_{B_i} são os vetores de posição das juntas passivas C_i e B_i . Essa restrição impõe:

$$\left\| \begin{bmatrix} \mu_i - l_1 \cos(\theta_i) \\ \rho_i - l_1 \sin(\theta_i) \end{bmatrix} \right\| = \left\| l_2 \begin{bmatrix} \cos(\beta_i) \\ \sin(\beta_i) \end{bmatrix} \right\| = l_2, \quad (3.1)$$

sendo μ_i e ρ_i definidos a seguir como

$$\begin{bmatrix} \mu_i \\ \rho_i \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - h_i \begin{bmatrix} \cos(\alpha + \eta_i) \\ \sin(\alpha + \eta_i) \end{bmatrix} - a_i \begin{bmatrix} \cos(\lambda_i) \\ \sin(\lambda_i) \end{bmatrix} - \delta_i \begin{bmatrix} \cos(\gamma_i) \\ \sin(\gamma_i) \end{bmatrix}. \quad (3.2)$$

Dessa forma θ_i pode ser calculado como

$$\theta_i = 2 \arctan\left(\frac{-e_{i1} \pm \sqrt{e_{i1}^2 + e_{i2}^2 - e_{i3}^2}}{e_{i3} - e_{i2}}\right), \quad (3.3)$$

em que $e_{i1} = -2l_{1i}\rho_i$, $e_{i2} = -2l_{1i}\mu_i$ e $e_{i3} = \mu_i^2 + \rho_i^2 + l_{1i}^2 - l_{2i}^2$.

E também temos que β_i pode ser dado por

$$\beta_i = \arctan\left(\frac{\rho_i - l_1 \sin \theta_i}{\mu_i - l_1 \cos \theta_i}\right), \quad (3.4)$$

O modelo cinemático inverso para um manipulador rígido é necessário para este trabalho, devido à necessidade de referenciar a posição dos encoders, uma vez que esses registram apenas a variação do deslocamento angular. Na Tabela 1, se encontram os valores das constantes do manipulador.

Tabela 1 – Constantes Geométricas

Variável	Valor [m]
h	34,9133*10 ⁻³
a	259,8076*10 ⁻³
l ₁	0,3188
l ₂	0,3188

Fonte: Próprio Autor

3.2.2 Descrição do problema e metodologia de resolução

O problema em questão consiste em encontrar um modelo para o manipulador paralelo flexível 3RRR. Esse modelo tem como objetivo funcionar como um previsor para

a postura do efetuador final. Para isso foi utilizado um sistema de controle em malha aberta para a aquisição dos dados (Fig. 13), onde a câmera foi responsável pela aquisição da postura do efetuador final, os encoders pela variação angular das juntas ativas e os extensômetros por capturar a deformação dos elos. Matematicamente podemos descrever o problema pela eq. 3.5.

$$(x, y, \alpha) = F(e_1, e_2, e_3, s_1, s_2, s_3, s_4, s_5, s_6) \quad (3.5)$$

As variáveis e_1 , e_2 e e_3 são referentes a variação angular nos encoders, enquanto que s_1 , s_2 , s_3 , s_4 , s_5 e s_6 são referentes as deformações nos extensômetros, e por fim x , y , α se referem a posição do efetuador final. A função F que prevê a posição do efetuador final, quando dados os valores dos encoders e dos extensômetros, foi modelada com o auxílio de uma RNA (Fig. 17), detalhes sobre a estrutura da RNA são apresentados na Seção 4.3. O software MATLAB foi utilizado para modelar essa RNA.

Figura 17 – Esquemático da RNA



Fonte: Próprio Autor

A metodologia que foi utilizada afim de encontrar essa RNA é dividida em três etapas:

- Obtenção dos dados experimentais;
- Tratamento dos dados experimentais;
- Treinamento das RNAs.

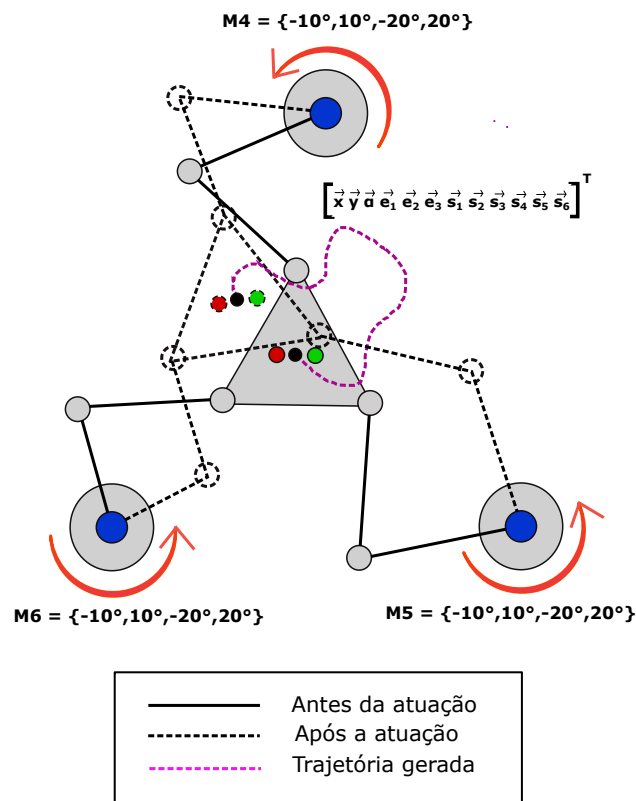
Cada uma dessas três etapas é tratada em detalhes na Seção 4.

4 RESULTADOS E DISCUSSÃO

4.1 Obtenção dos Dados Experimentais

Para a obtenção dos dados experimentais foi feito o uso do aparato experimental apresentado na Fig. 13. Esses dados foram obtidos submetendo o manipulador a um esquema de malha aberta, onde foram realizadas atuações nos motores. As atuações eram feitas acionando um motor, dois motores simultaneamente, ou os três motores simultaneamente. As entradas nos motores eram dadas em deslocamento angular, os módulos e direções utilizados para esse procedimento poderiam ser: 10° , -10° , 20° , -20° (Fig. 18). Essas entradas foram escolhidas porque na prática, verificou-se que aplicando deslocamentos dessa magnitude e direção, e com o efetuador final próximo a origem do sistema de coordenadas, o manipulador entrava mais raramente em regiões de singularidades que poderiam danificar o aparato experimental em questão, quando o manipulador atingia essa região uma parada imediata era necessária.

Figura 18 – Esquemático de atuações no manipulador para obtenção de dados



Fonte: Próprio Autor

A escolha do número de motores a serem acionados, e os valores de deslocamento

angulares desses, foram feitos de maneira aleatória pelo operador. Uma escolha possível poderia ser $10^\circ M1 -10^\circ M2 0^\circ M3$. Para essa ação era gerado um arquivo, onde eram registrados os valores de deslocamento angular incremental nos encoders, valores de tensão nos extensômetros e valores de posição para as coordenadas do efetuador final. Esse registro não era feito de forma automática pelo sistema, antes do início da atuação, o operador deveria iniciar a gravação que duraria por um período pré-determinado de 10,001s. Ao todo para esse projeto foram feitas 18 atuações (dessas 10 foram válidas), cada uma contando com 10001 pontos de captura de dados, cada captura era feita a cada 1 *ms* para os encoders e extensômetros, e a 20 *fps* pela câmera. Por fim, o arquivo era trabalhado e convertido em outro arquivo, esse arquivo foi o utilizado na etapa de Tratamento dos Dados Experimentais. Uma atuação, é apresentada na Fig. 19 e 20 em sua forma bruta.

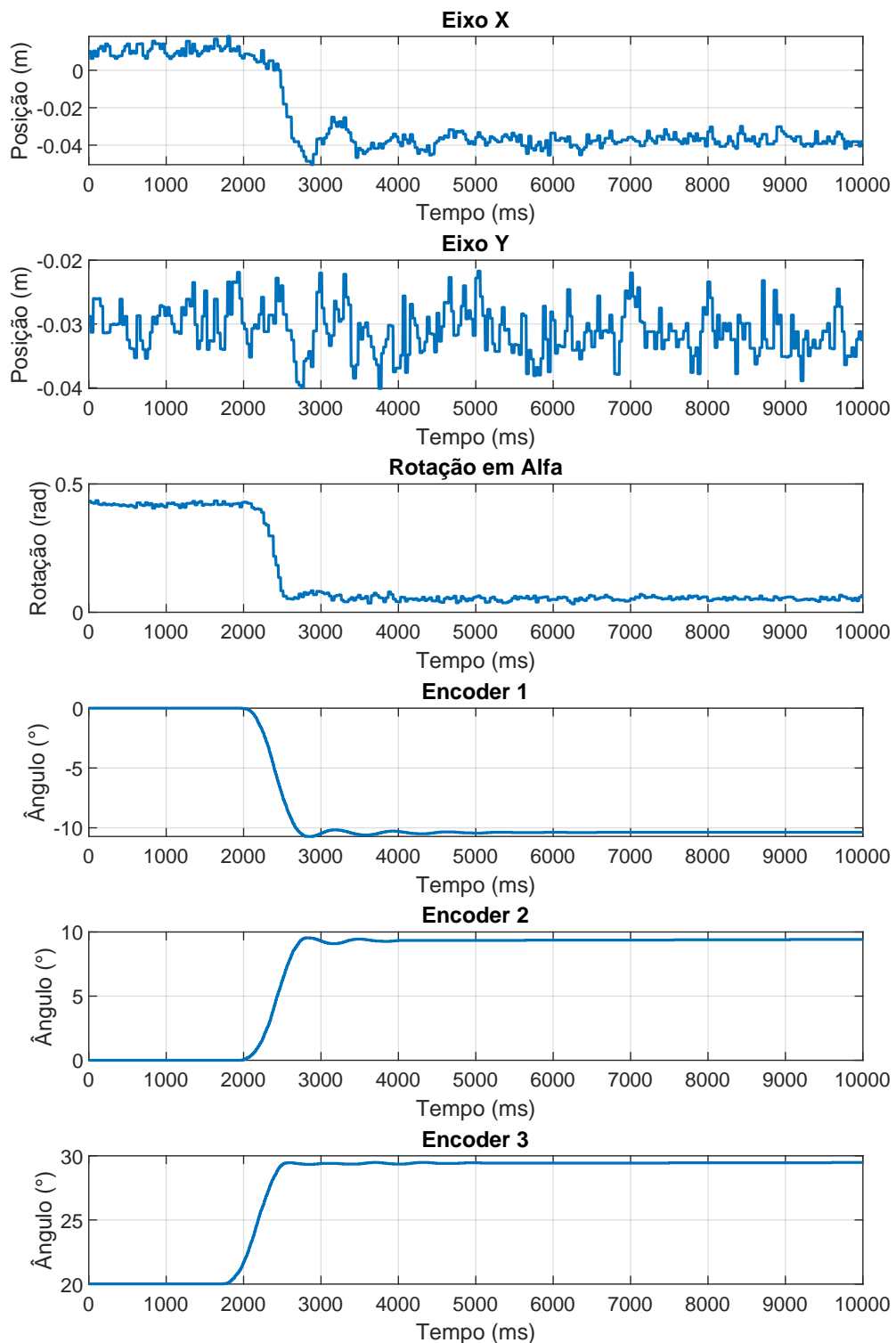
As 10 trajetórias válidas coletada foram unidas a outras 13 trajetórias válidas (coletadas no trabalho de Félix (2021)), coletadas nesse mesmo manipulador, no entanto em um momento distinto. Para esse trabalho as coletas mais recentes de trajetórias são chamadas de Geração 2 (G2), enquanto que as trajetórias mais antigas de Geração 1 (G1). Juntando as duas gerações de trajetórias, obteve-se um conjunto com 23 trajetórias para estudo.

4.2 Tratamento dos Dados Experimentais

Essa etapa tem como objetivo eliminar os ruídos presentes nos dados, retirar zonas de dados desnecessárias e adequar os dados aos fins desejados. Para isso cada trajetória foi analisada. Os dados em questão podem ser divididos em três zonas para cada trajetória: pré-operação, operação e pós-operação. Isso acontece devido a imprecisão no momento de gravar os dados, conforme descrito em 4.1. Abaixo, é feita uma breve descrição dessas três zonas.

- Pré-operação: compreende o momento em que o aparato experimental já começou a gravar os dados, no entanto, a atuação ainda não teve início. Em uma primeira análise, esses dados parecem apenas causar volume ao conjunto de dados, no entanto, eles são muito úteis para definir com mais precisão a posição angular dos encoders.
- Operação: é onde de fato a gravação coincide com a operação do manipulador e trata-se da zona de maior interesse, onde será feito o tratamento dos dados.
- Pós-operação: trata-se de uma zona morta, onde o manipulador já encerrou a sua atuação, no entanto, o sistema ainda segue gravando. Essa zona diferente das outras duas não possui funcionalidade alguma, apenas faz volume nos dados, dessa forma ela é descartada.

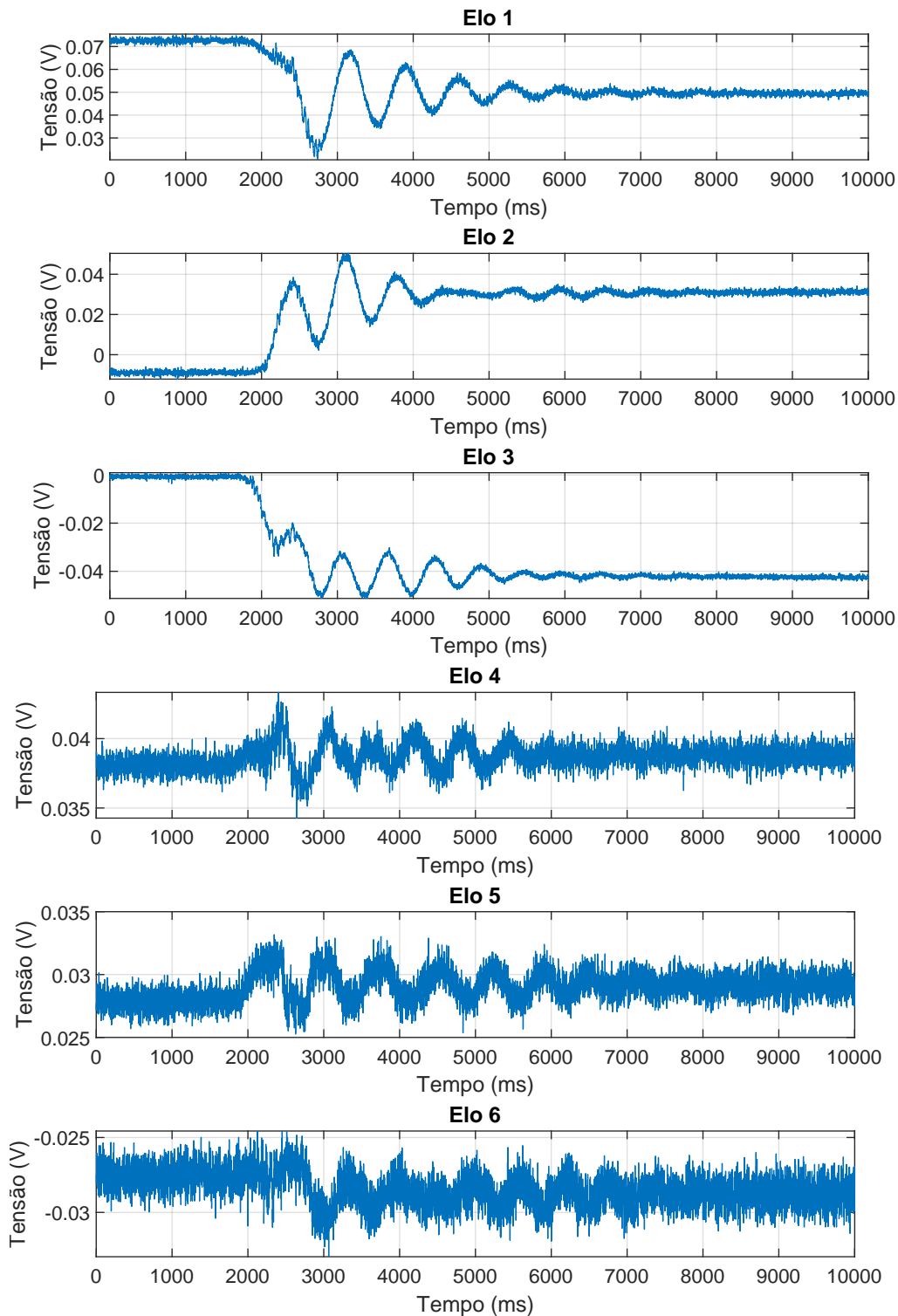
Figura 19 – Posição do efetuador final (dados brutos) e deslocamentos incrementais dos encoders (dados brutos) para uma das atuações feitas no experimento



Fonte: Próprio Autor

Para o efetuador final e extensômetros, utilizou-se apenas a zona de operação, e aplicou-se a ela um filtro digital, seguindo os mesmos passos apresentados no trabalho de

Figura 20 – Medidas de tensão para os extensômetro (dados brutos) para uma das atuações feitas no experimento



Fonte: Próprio Autor

Félix (2021). Neste trabalho, o autor aplica filtros digitais usando uma função chamada de *designfilt*, presente no MATLAB, utilizando essa função é necessário entrar com os

parâmetros de ordem do filtro, frequência de corte e taxa de amostragem do filtro para com isso poder se obter os parâmetros do filtro.

Para os dados provenientes da câmera foi utilizado um filtro passa-baixa com os seguintes parâmetros: ordem do filtro de 35, frequência de corte de 2 Hz e taxa de amostragem de 0,001 s. Afim de eliminar a característica retangular presente nas curvas capturadas por esse aparelho, aspecto este que ocorre devido a taxa de *frames* por segundo (*fps*) da câmera ser inferior à taxa de aquisição de dados da DSPACE. Para os dados provenientes dos extensômetros os parâmetros se deram da seguinte forma: ordem do filtro de 150, frequência de corte de 2 Hz e taxa de amostragem de 0,001 s. Neste caso, o filtro tem como objetivo atuar na retirada dos ruídos presentes na medição.

Os encoders utilizados nesse trabalho, apenas registravam a variação incremental do deslocamento angular das juntas. Dessa maneira, foi necessário referenciar a posição inicial das juntas ativas, para isso, fez-se o uso da zona de pré-operação. Uma média aritmética foi feita com os pontos dessa zona, a cinemática inversa para um manipulador rígido (3.2.1), foi aplicada a esse valor para poder encontrar a posição aproximada inicial dos encoders. Esse valor foi somado aos valores incrementais dos encoders.

Para aumentar o desempenho de treinamento da RNA tanto os dados de saída quanto os dados de entrada foram normalizados no intervalo $[0,1]$. A mesma atuação apresentada na Fig. 19 e Fig. 20 é apresentada com o tratamento na Fig. 21 e Fig. 22, sem a normalização.

4.3 Treinamento das Redes Neurais

4.3.1 Redes de Múltiplas Camadas Alimentadas Adiante

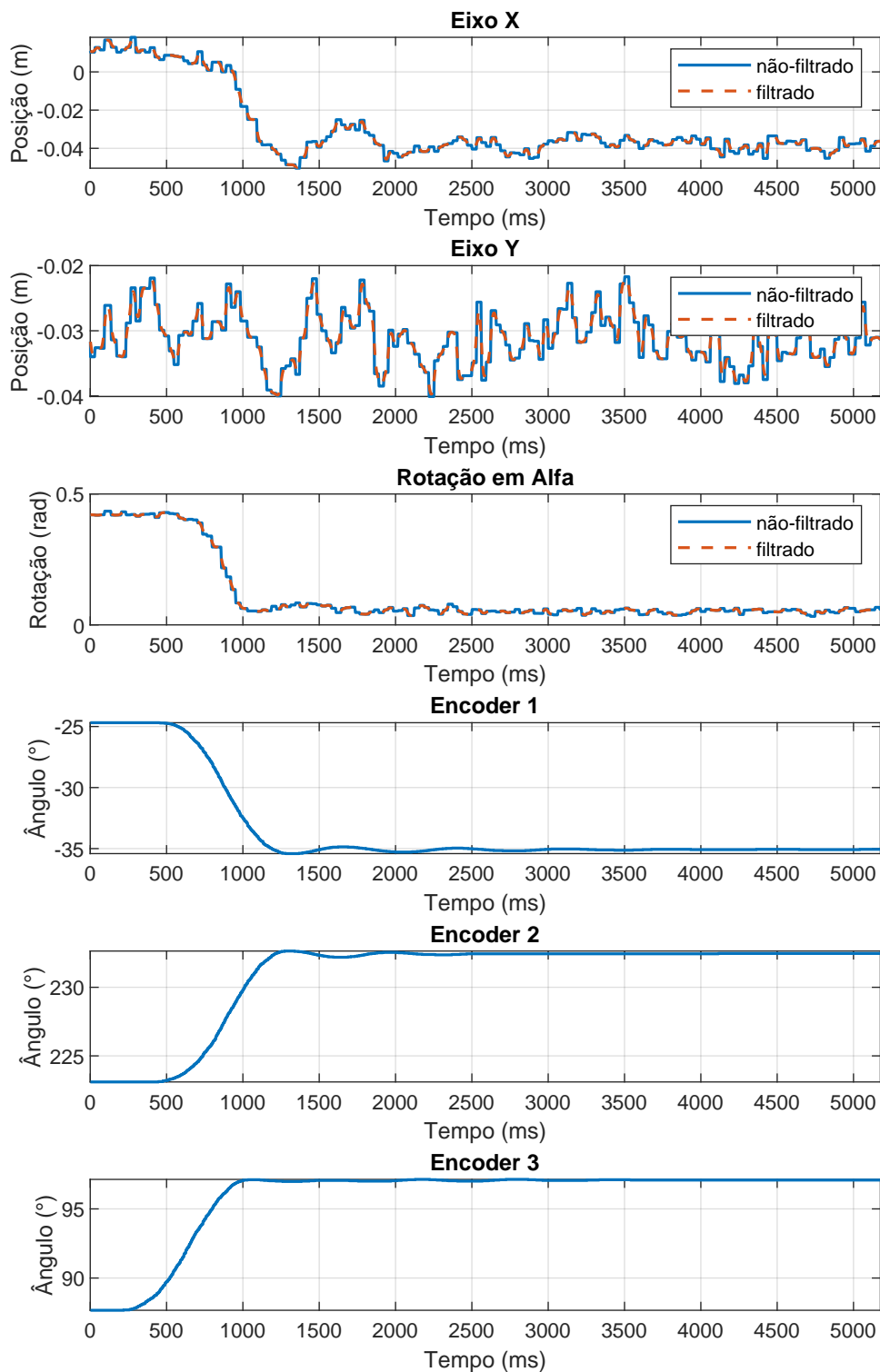
Para cada uma das RNAs foram utilizadas três camadas, sendo: duas dessas camadas internas e uma de saída (Fig. 23), o motivo de se utilizar duas camadas internas para a RNA está no fato de que fazendo verificações experimentais, percebeu-se que a curva se ajusta melhor e de maneira mais suave quando utilizada com duas camadas internas.

Duas estratégias diferentes foram utilizadas para treinar as RNAs, a primeira estratégia consistiu em utilizar apenas uma trajetória como entrada para o treinamento, enquanto a outra em utilizar mais de uma trajetória como entrada para o treinamento, ambas a técnicas e os seus resultados são descritos nas subseções seguintes.

4.3.1.1 Treinamento da RNA com uma trajetória

Essa estratégia funciona da seguinte maneira. Para cada trajetória foi treinada uma RNA com os parâmetros apresentados na Tab. 2. Realizados os treinamentos era feita a verificação se alguma das RNAs obtidas era suficiente para generalizar todas as

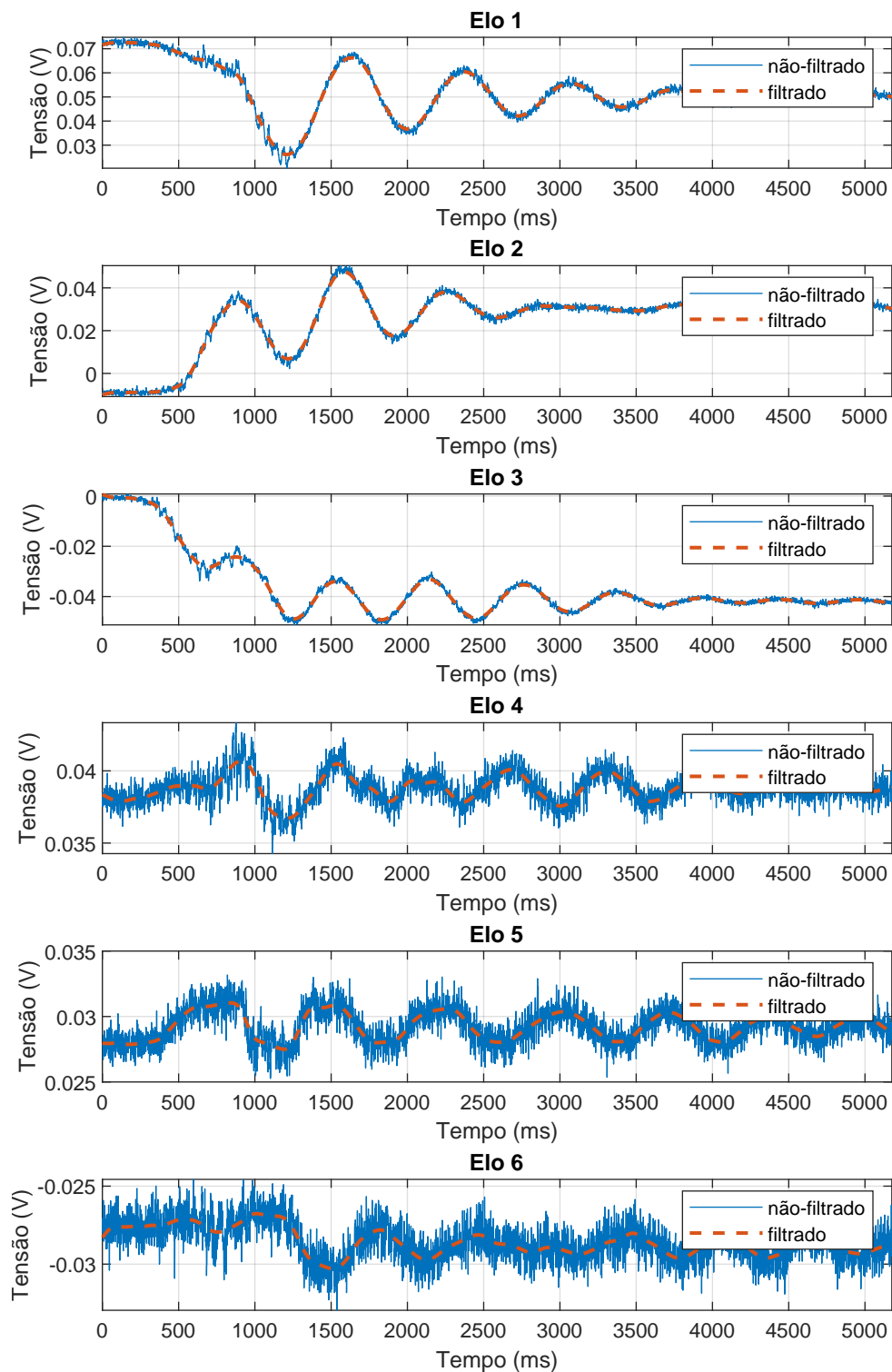
Figura 21 – Posição do efetuador final (filtrado x não-filtrado) e deslocamentos dos encoders para uma das atuações feitas no experimento



Fonte: Próprio Autor

trajetórias, a verificação foi realizada através da média do MSE das três saídas da rede, e também com a média do MAPE. Os erros foram todos organizados em uma matriz,

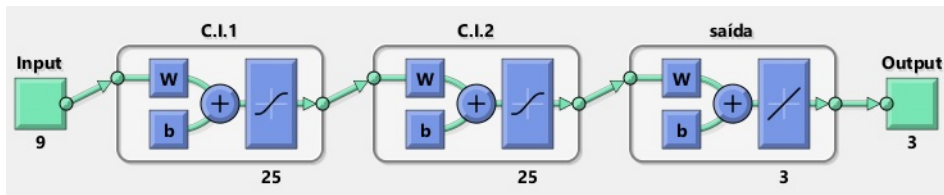
Figura 22 – Medidas de tensão para os extensômetros (filtrado x não-filtrado) para uma das atuações feitas no experimento



Fonte: Próprio Autor

denominada matriz de erro, onde a cada coluna é atribuída uma RNA obtida durante os treinamentos, enquanto que em cada linha é atribuída uma trajetória, dessa forma cada

Figura 23 – Esquemático da RNA para testes com uma única trajetória como entrada



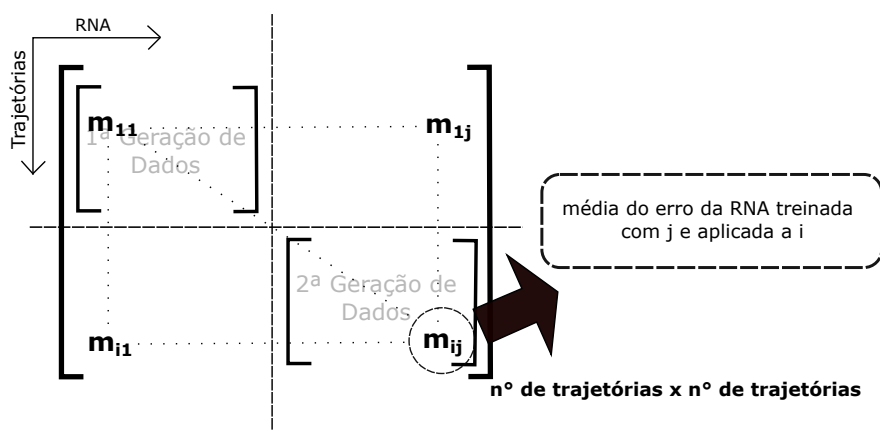
Fonte: Próprio Autor

Tabela 2 – Parâmetros do treinamento.

Parâmetro	Tipo/Valor
Treinamento	<i>trainlm</i>
Performance	<i>mse</i>
Divisão dos dados	<i>dividerand</i>
Função de ativação camadas internas	<i>tansig</i>
Função de ativação camada de saída	<i>purelin</i>
Dados de treinamento	70 %
Dados de validação	15 %
Dados de teste	15 %
n° de camadas	2
n° de neurônios por camada	25
n° de épocas	1000

entrada da matriz se refere ao erro obtido, aplicando-se uma determinada RNA a uma determinada trajetória. O esquemático ilustrado na Fig. 24 ajuda a entender como os dados são organizados e seu significado.

Figura 24 – Esquemático da matriz do erro



Fonte: Próprio Autor

Na Tab. 3 é apresentada parte dos resultados da matriz de erro do MSE e da matriz

do erro do MAPE. Os resultados de ambas as medidas de erro são fornecidos lado a lado separados por uma barra vertical. A matriz completa de ambos os tipos de erro pode ser apreciada no Apêndice A e B.

Tabela 3 – Representação de parte das matrizes de erro.

Traj.	$RNA_1 [m^2 \%]$	$RNA_2 [m^2 \%]$	$RNA_3 [m^2 \%]$	$RNA_4 [m^2 \%]$
1 ^a	$4,7 * 10^{-8} 0,03$	$6,5 * 10^{-2} 27,40$	$9,4 * 10^{-1} 63,66$	$8,4 * 10^{-1} 59,41$
2 ^a	$3,71 * 10^{-2} 32,62$	$5,3 * 10^{-8} 0,05$	$3,9 93,53$	$4,2 * 10^{-1} 57,13$
3 ^a	$1,3 * 10^{-1} 47,10$	$5,9 * 10^{-2} 41,13$	$5,3 * 10^{-5} 1,49$	$4,9 * 10^{-1} 62,81$
4 ^a	$2,1 * 10^{-1} 34,71$	$3,6 * 10^{-2} 33,25$	$7,8 * 10^{-1} 124,88$	$1,2 * 10^{-6} 0,06$

Analisando apenas os resultados apresentados na Tab. 3 é possível concluir que essa metodologia de entrada para os dados não foi suficiente, uma vez que apenas as trajetórias que participaram do treinamento apresentaram um bom resultado quando submetidas a RNA, as trajetórias que não participaram do treinamento apresentaram um média do MAPE que algumas vezes chega a ultrapassar a casa dos 100%.

4.3.1.2 Treinamento da RNA com múltiplas trajetórias

Para essa segunda forma de análise dos dados foram treinadas 23 RNAs cada vez adicionando uma nova trajetória para treinar a RNA seguinte. Dessa forma, a RNA_1 foi treinada apenas com a trajetória 1, a RNA_2 foi treinada com as trajetórias 1 e 2, a RNA_3 com as trajetórias 1, 2 e 3, e assim sucessivamente, até que a RNA_{23} foi treinada com todas as trajetórias. Os parâmetros de treinamento são os mesmos apresentados na Tab. 2, a única diferença está no fato da rede possuir agora 15 neurônios por camada. Na Tab. 4 é exibida parte da matriz do erro a partir da coluna 18, uma vez que é próximo a essa coluna que começam a ocorrer os melhores resultados. A matriz completa pode ser apreciada no Apêndice C e D.

Tabela 4 – Representação de parte das matrizes de erro.

Traj.	$RNA_{18} [m^2 \%]$	$RNA_{19} [m^2 \%]$	$RNA_{20} [m^2 \%]$	$RNA_{22} [m^2 \%]$
18 ^a	$1,4 * 10^{-5} 1,30$	$1,1 * 10^{-5} 1,43$	$9,9 * 10^{-6} 1,15$	$1,8 * 10^{-5} 1,93$
19 ^a	$2,3 * 10^{-4} 7,10$	$1,0 * 10^{-5} 2,16$	$8,1 * 10^{-6} 1,4$	$5,0 * 10^{-6} 1,14$
20 ^a	$5,6 * 10^{-3} 36,64$	$1,8 * 10^{-2} 43,75$	$5,6 * 10^{-6} 2,05$	$3,9 * 10^{-6} 1,44$
21 ^a	$1,1 * 10^{-4} 4,44$	$1,4 * 10^{-4} 3,29$	$8,7 * 10^{-5} 3,18$	$6,6 * 10^{-6} 0,89$
22 ^a	$2,7 * 10^{-3} 17,23$	$3,8 * 10^{-3} 28,92$	$3,4 * 10^{-3} 15,24$	$6,5 * 10^{-6} 0,63$
23 ^a	$3,0 * 10^{-3} 19,23$	$4,2 * 10^{-3} 24,31$	$2,6 * 10^{-3} 15,48$	$1,0 * 10^{-3} 6,53$

Analisando as matrizes de erro é possível perceber que trajetórias que participam do treinamento apresentam uma média do MAPE máxima entorno de 6% em raras exceções, a maioria dos erros são abaixo disso, e uma média do MSE máxima da ordem de $10^{-4}m^2$. Com isso, é possível concluir que as redes aprendem de maneira satisfatória os exemplos a ela submetidos, e conseguem reproduzi-los posteriormente. No entanto, a previsão de

trajetórias que não participaram do treinamento só começa a ficar interessantes após alimentar a RNA com 17 trajetórias, uma vez que a maior média do MAPE é de 43,75% e a menor é de 3,29% (Tab. 4), esses resultados não satisfatórios e também não servem para o propósito desse projeto, no entanto mostra que ao inserir um número considerável de trajetórias é possível prever bem algumas trajetórias, dados os seus dados de entrada.

4.3.2 Redes de Múltiplas Camadas Alimentadas Adiante Atrasadas no Tempo

Para esses tipos de RNA somente aplicou-se a metodologia de entradas múltiplas, tendo como base os resultados desastrosos que se obteve com as MLPs aplicando apenas uma trajetória.

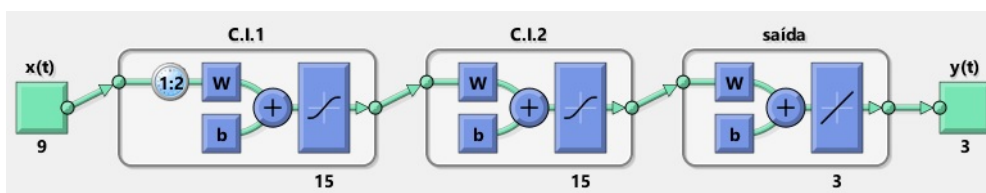
4.3.2.1 Treinamento de TLFN focadas com Múltiplas Trajetórias

Na Tab. 5 podem ser visualizados os parâmetros utilizados para o treinamento desse tipo de rede, na Fig. 25 a rede treinada, no Apêndice E a matriz do erro para o MSE e no Apêndice F a matriz do erro para o MAPE.

Tabela 5 – Parâmetros do treinamento RNAs atrasadas focadas no tempo.

Parâmetro	Tipo/Valor
Treinamento	<i>trainlm</i>
Performance	<i>mse</i>
Divisão dos dados	<i>dividerand</i>
Função de ativação camadas internas	<i>tansig</i>
Função de ativação camada de saída	<i>purelin</i>
Dados de treinamento	70 %
Dados de validação	15 %
Dados de teste	15 %
n° de camadas	2
n° de neurônios por camada	15
Atrasos de tempo da 1ª camada	2
Atrasos de tempo da 2ª camada	0
Atrasos de tempo da camada de saída	0
n° de épocas	1000

Figura 25 – Esquemático da RNA atrasada focada no tempo



Fonte: Próprio Autor

Esse tipo de RNA assim como as MLPs consegue reproduzir bem os exemplos que participaram do treinamento, no entanto só começam a exibir resultados de previsão abaixo de 50 %, após receber 17 trajetórias para treinamento. A melhor previsão a partir desse momento é de 2,5 % com MSE de $6,5 * 10^{-5}m^2$ e a pior de 39,38 % e MSE de $1,0 * 10^{-2}$.

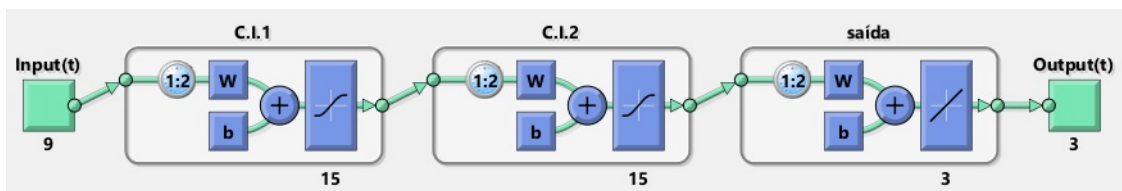
4.3.2.2 Treinamento de TLFN distribuídas com Múltiplas Trajetórias

Na Tab. 6 podem ser visualizados os parâmetros utilizados para o treinamento desse tipo de rede, na Fig. 26 a rede treinada, no Apêndice G a matriz do erro para o MSE e no Apêndice H a matriz do erro para o MAPE.

Tabela 6 – Parâmetros do treinamento RNAs atrasadas distribuídas no tempo.

Parâmetro	Tipo/Valor
Treinamento	<i>trainlm</i>
Performance	<i>mse</i>
Divisão dos dados	<i>dividerand</i>
Função de ativação camadas internas	<i>tansig</i>
Função de ativação camada de saída	<i>purelin</i>
Dados de treinamento	70 %
Dados de validação	15 %
Dados de teste	15 %
n° de camadas	2
n° de neurônios por camada	15
Atrasos de tempo da 1ª camada	2
Atrasos de tempo da 2ª camada	2
Atrasos de tempo da camada de saída	2
n° de épocas	1000

Figura 26 – Esquemático da RNA atrasada distribuída no tempo



Fonte: Próprio Autor

Esse tipo de RNA assim como as MLPs e TLFN focadas conseguem reproduzir bem os exemplos que participaram do treinamento, no entanto só começam a exibir resultados de previsão abaixo de 50 %, após receber 16 trajetórias para treinamento. A melhor previsão a partir desse momento é de 4,3 % com MSE de $1,4 * 10^{-4}m^2$ e a pior de 36,55 % e MSE de $1,2 * 10^{-2}$.

4.4 Comparação dos resultados entre os diferentes tipos de RNAs

Como pode ser verificado, as previsões das RNAs tem uma tendência a melhorar a cada nova trajetória que é adicionada ao treinamento, uma vez que para a MLP e TLFN focada as previsões começaram a ficar inferiores a 50 % a partir da trajetória 17 e para a TLFN distribuída a partir da trajetória 16. Observando essa tendência, é interessante analisar as últimas RNAs treinadas. Dessa forma, é feita uma análise das três últimas RNAs para cada um dos três tipos de RNAs.

4.4.1 RNA_{22} aplicada à trajetória 23

Na Tab. 7 é apresentada a média dos erros quando os dados de entrada da trajetória 23 são submetidos a RNA_{22} para as redes MLP, TLFN focada e TLFN distribuída.

Tabela 7 – Comparação da previsão entre os tipos de RNA para a RNA_{22} aplicada à trajetória 23.

Traj.	Tipo de RNA	Média do MSE [m^2]	Média do MAPE [%]
23 ^a	MLP	$1,0 * 10^{-3}$	6,53
	TLFN focada	$1,6 * 10^{-4}$	3,43
	TLFN distribuída	$7,4 * 10^{-4}$	6,52

Na Fig. 27 pode ser visualizada as curvas da RNA_{22} quando aplicada aos dados de entrada da trajetória 23 para os três tipos de RNA.

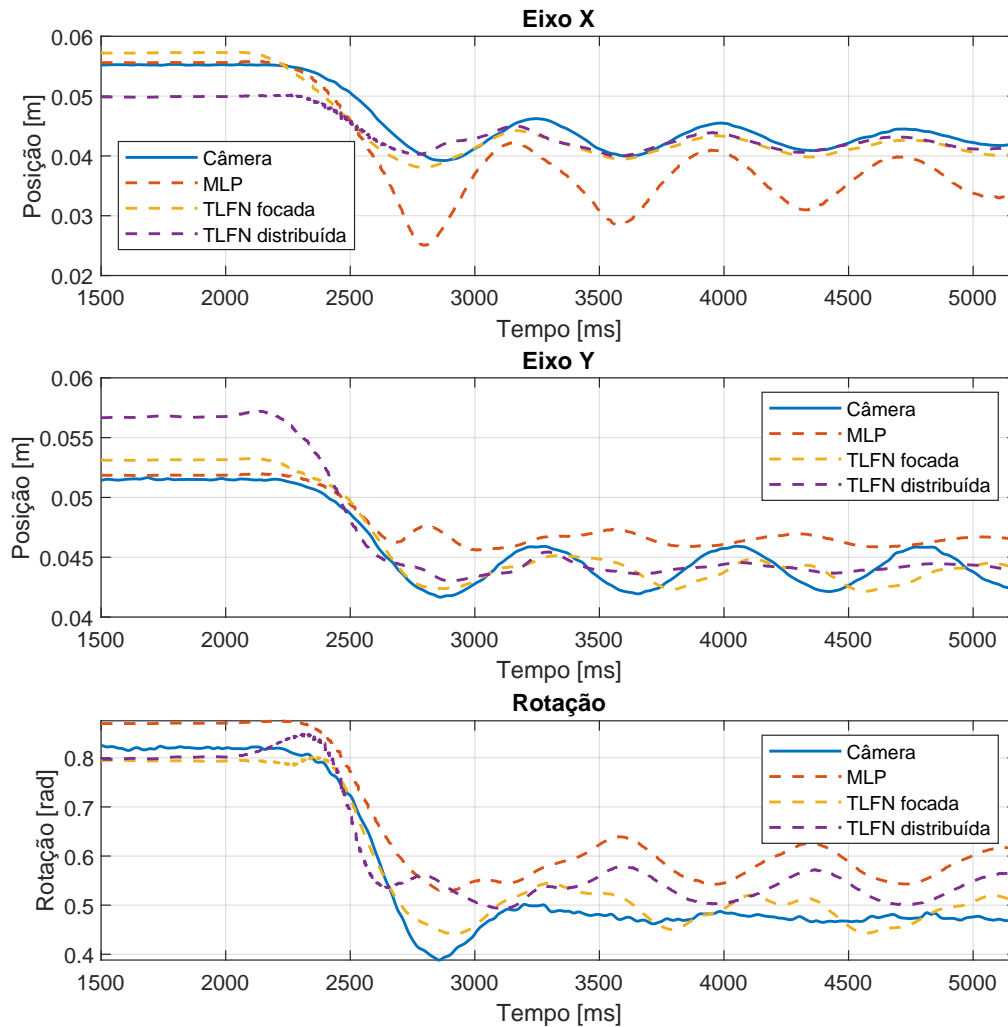
Levando em consideração tanto os valores de erro apresentados na Tab. 7, quanto as curvas apresentadas na Fig. 27. Para esse caso em específico é possível dizer que as TLFNs apresentam um desempenho superior a MLP, e TLFN focada o maior desempenho. No entanto, uma única trajetória é muito pouco para se chegar a uma conclusão do melhor tipo de rede. Dessa forma, é necessário continuar a análise para a RNA_{21} , com o uso dessa RNA fica disponível para análise de erro as trajetórias 22 e 23.

4.4.2 RNA_{21} aplicada à trajetória 22 e 23

Na Tab. 8 é apresentada a média dos erros quando os dados de entrada da trajetória 22 e 23 são submetidos a RNA_{21} para as redes MLP, TLFN focada e TLFN distribuída.

Na Fig. 28 e Fig. 29 podem ser visualizadas as curvas da RNA_{21} quando aplicada aos dados de entrada da trajetória 22 e 23 para os três tipos de RNA.

Com o uso RNA_{21} é possível observar que o desempenho de todas as RNAs foi comprometido, no entanto as TLFNs continuam a apresentar um resultado superior a MLP, e para esse caso em específico a TLFN distribuída apresentou um melhor desempenho, como pode ser visualizado na Tab. 8 e nas curvas da Fig. 28 e 29.

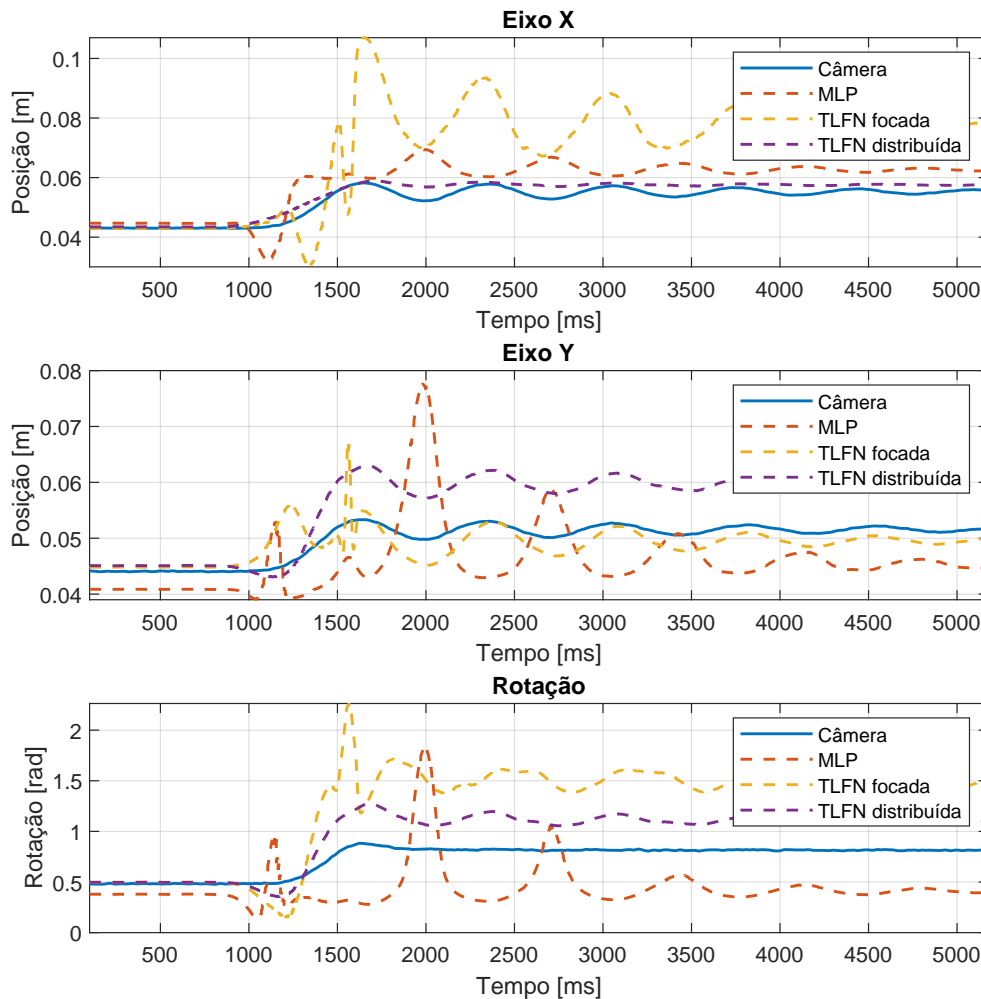
Figura 27 – RNA_{22} aplicada à trajetória 23 para os três tipos de RNAs

Fonte: Próprio Autor

Tabela 8 – Comparação da previsão entre os tipos de RNA para a RNA_{21} aplicada à trajetória 22 e 23.

Traj.	Tipo de RNA	Média do MSE [m^2]	Média do MAPE [%]
22 ^a	MLP	$5,4 * 10^{-3}$	29,43
	TLFN focada	$1,7 * 10^{-2}$	25,61
	TLFN distribuída	$2,7 * 10^{-3}$	13,73
23 ^a	MLP	$1,4 * 10^{-2}$	31,73
	TLFN focada	$2,9 * 10^{-3}$	14,95
	TLFN distribuída	$1,6 * 10^{-3}$	11,51

Figura 28 – RNA_{21} aplicada à trajetória 22 para os três tipos de RNAs



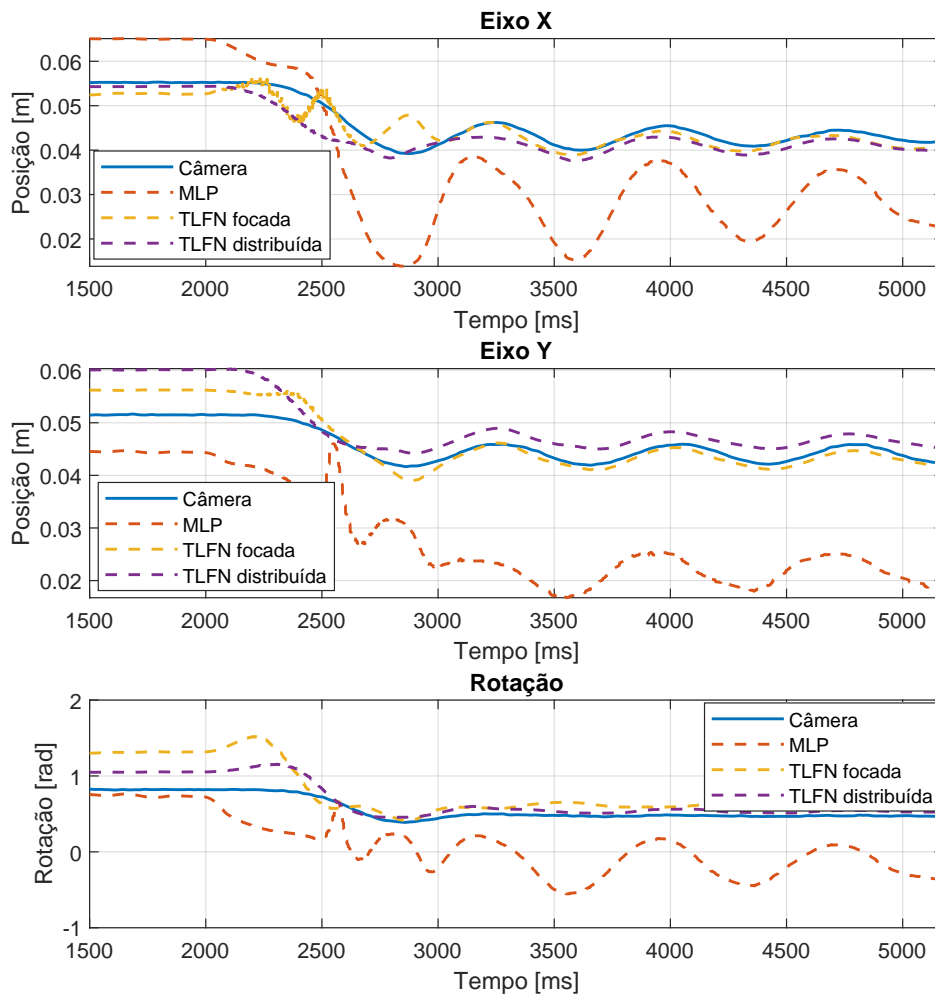
Fonte: Próprio Autor

4.4.3 RNA_{20} aplicada à trajetória 21, 22 e 23

Por fim, vamos retroceder mais um pouco e fazer as mesmas análises para a RNA_{20} , esta RNA apresenta disponíveis para comparação as trajetórias 21, 22 e 23. Na Tab. 9 é apresentada a média dos erros quando os dados de entrada da trajetória 21, 22 e 23 são submetidos a RNA_{20} para as redes MLP, TLFN focada e TLFN distribuída.

Na Fig. 30, Fig. 31 e 32 podem ser visualizadas as curvas da RNA_{20} quando aplicada aos dados de entrada da trajetória 21, 22 e 23 para os três tipos de RNA.

Para a trajetória 21 todas as RNAs fizeram boas previsões se sobressaindo a MPL. A previsão da trajetória 22 e 23 não foram boas, as redes que obtiveram os melhores resultados nessa ocasião foram a MPL e TLFN distribuída.

Figura 29 – RNA_{21} aplicada à trajetória 23 para os três tipos de RNAs

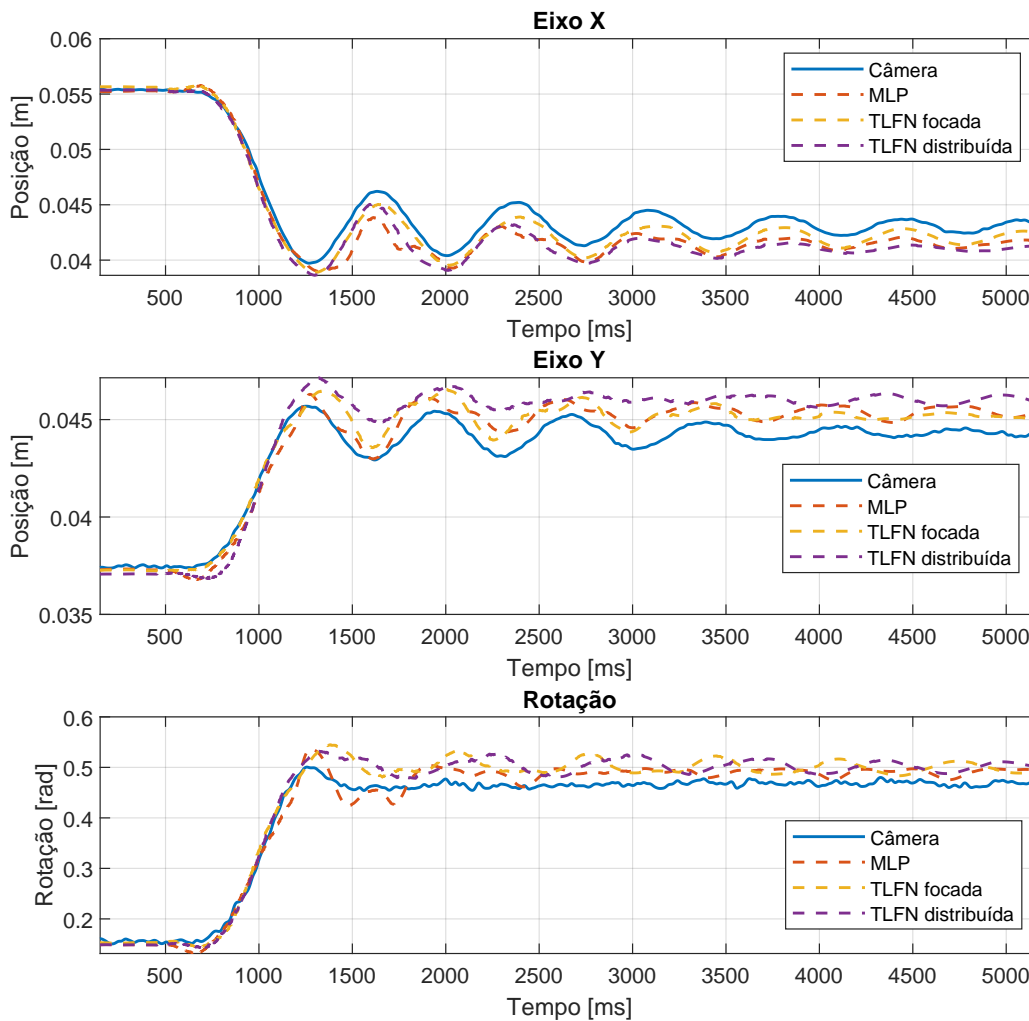
Fonte: Próprio Autor

Por fim, analisando os resultados apresentados da RNA_{20} , RNA_{21} e RNA_{22} . O tipo de RNA que aparenta apresentar uma maior consistência em suas previsões é a TLFN distribuída, no entanto não é possível chegar a uma conclusão, uma vez que os resultados indicam que um número maior de trajetórias deveria ser coletado.

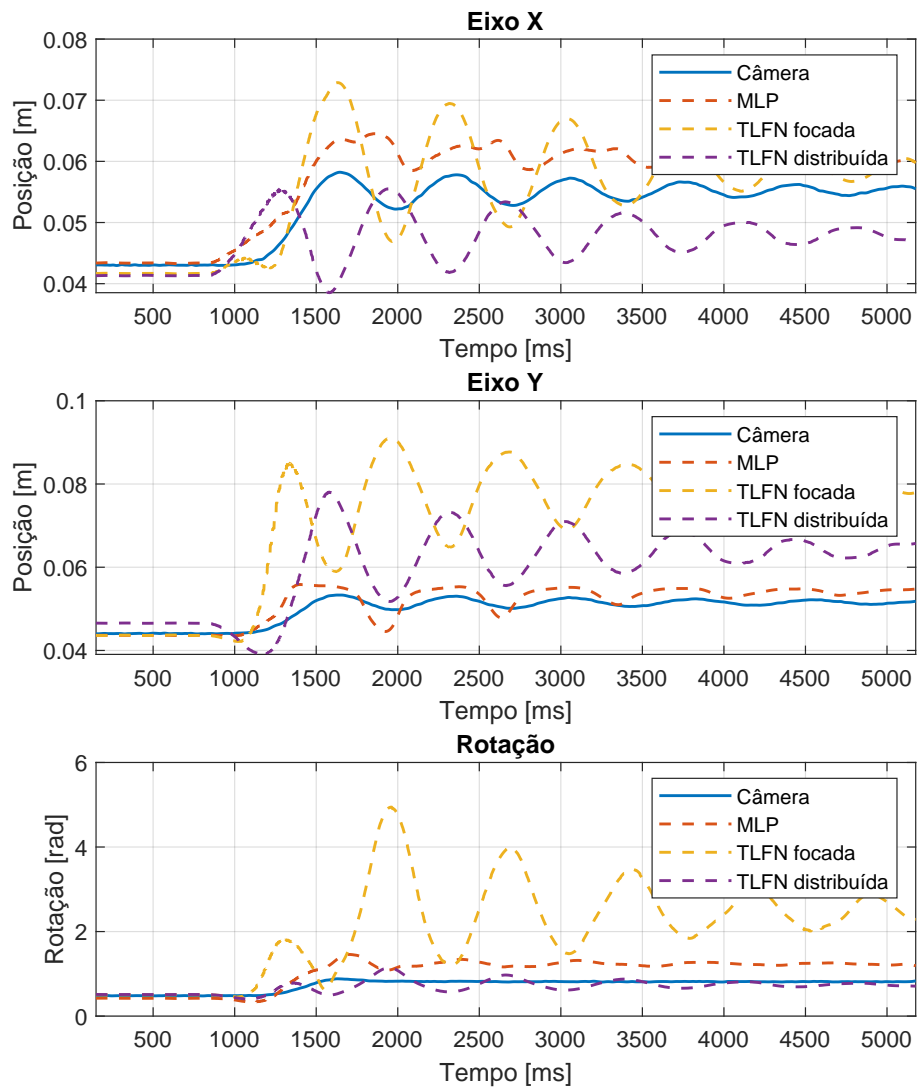
Tabela 9 – Comparação da previsão entre os tipos de RNA para a RNA_{20} aplicada à trajetória 21, 22 e 23.

Traj.	Tipo de RNA	Média do MSE [m^2]	Média do MAPE [%]
21 ^a	MLP	$8,7 * 10^{-5}$	3,18
	TLFN focada	$6,1 * 10^{-5}$	3,48
	TLFN distribuída	$2,4 * 10^{-4}$	4,98
22 ^a	MLP	$3,4 * 10^{-3}$	15,24
	TLFN focada	$6,8 * 10^{-2}$	36,67
	TLFN distribuída	$4,9 * 10^{-3}$	15,08
23 ^a	MLP	$2,6 * 10^{-3}$	15,48
	TLFN focada	$1,1 * 10^{-1}$	37,87
	TLFN distribuída	$4,1 * 10^{-3}$	16,45

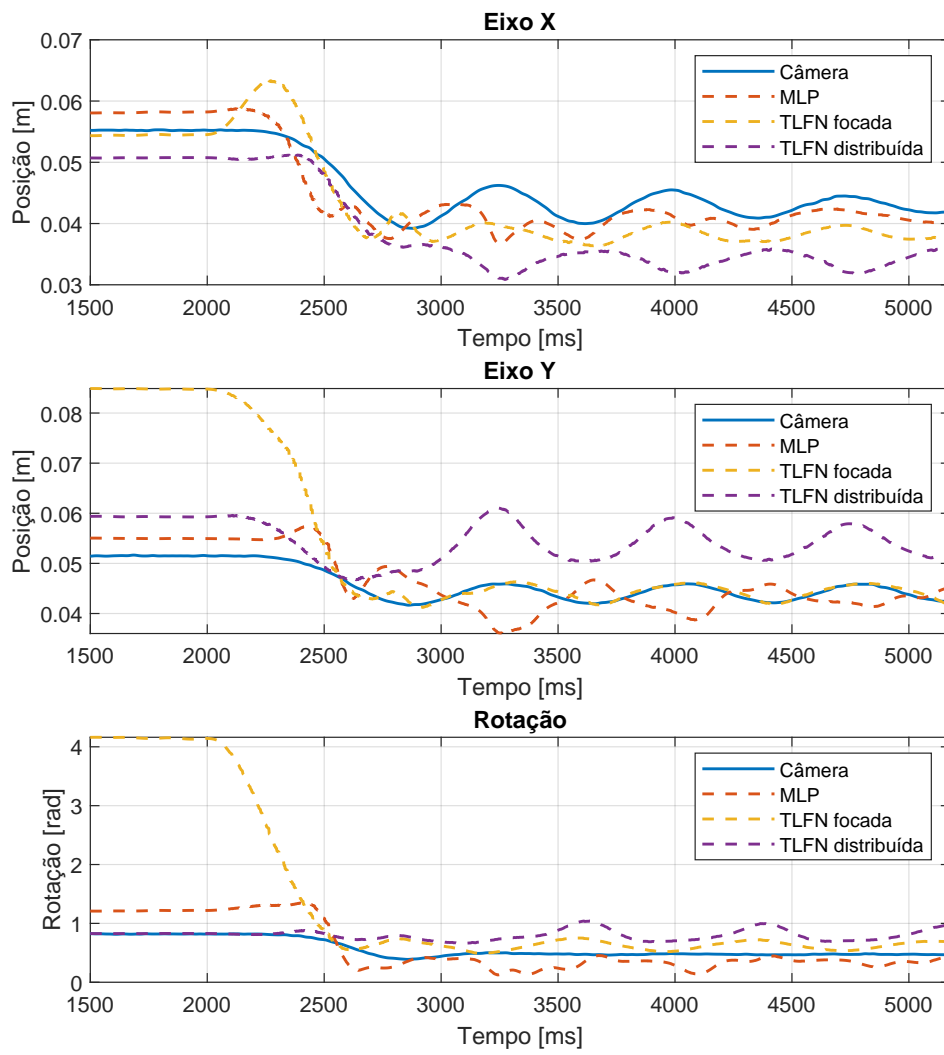
Figura 30 – RNA_{20} aplicada à trajetória 21 para os três tipos de RNAs



Fonte: Próprio Autor

Figura 31 – RNA_{20} aplicada à trajetória 22 para os três tipos de RNAs

Fonte: Próprio Autor

Figura 32 – RNA_{20} aplicada à trajetória 23 para os três tipos de RNAs

Fonte: Próprio Autor

5 CONCLUSÃO

Neste trabalho buscou-se a criação de um modelo utilizando RNAs para prever a postura do manipulador paralelo flexível em questão, o 3RRR. Para isso foram feitos vários acionamentos no manipulador e os dados referentes as trajetórias, geradas com esses acionamentos, foram armazenados e posteriormente tratados. A partir desses dados foram treinadas RNAs do tipo: MLPs, TLFNs focadas e TLFNs distribuídas. Os resultados encontrados indicam que a tentativa de criação de um modelo com RNA que possa prever a postura do manipulador paralelo flexível não foi alcançado, uma vez que os modelos encontrados conseguem fazer previsões que giram em torno dos 3 % da média do MAPE e média do MSE da ordem de 10^{-5} , no entanto essas previsões não se dão com consistência, algumas vezes oscilam para a casa dos 30 % do MAPE e média do MSE para valores da ordem de 10^{-2} . Foi verificado que essa oscilação diminuí tanto para as MLPs, TLFNs focadas e TLFNs distribuídas quando o número de exemplos de trajetórias que é submetido a rede é aumentado e que as RNAs que apresentam atrasos de tempo tendem a apresentar melhores resultados. Com isso, propõe-se que para trabalhos futuros aumente-se o conjunto de trajetórias coletadas para que se possa submeter as RNAs a um maior número de exemplos durante o treinamento e que as RNAs utilizadas tenham atraso de tempo.

REFERÊNCIAS

- BIDOKHTI, H. S.; ENFERADI, J. Direct kinematics solution of 3-rrr robot by using two different artificial neural networks. *In: IEEE. 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. [S.l.: s.n.], 2015. p. 606–611.
- COLOMBO, F. T.; FONTES, J. V. de C.; SILVA, M. M. da. A visual servoing strategy under limited frame rates for planar parallel kinematic machines. **Journal of Intelligent & Robotic Systems**, Springer, v. 96, n. 1, p. 95–107, 2019.
- CSISZAR, A.; EILERS, J.; VERL, A. On solving the inverse kinematics problem using neural networks. *In: IEEE. 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. [S.l.: s.n.], 2017. p. 1–6.
- FACELI, K. **Inteligência Artificial Uma Abordagem de Aprendizado de Máquina**. [S.l.: s.n.]: genio, 2021. v. 2.
- FONTES, J. V. d. C. **Aplicação de redundância para atingir altas acelerações com manipuladores robóticos planares**. 2015. Tese (Doutorado) — Universidade de São Paulo, 2015.
- FONTES, J. V. d. C. **Controle de um manipulador planar paralelo com redundância cinemática**. 2019. Tese (Doutorado) — Universidade de São Paulo, 2019.
- FÉLIX, L. Instrumentation of a parallel manipulator with flexible links: a neural network application. **COBEM 2021, 26^a international Congress of Mechanical Engineering**, p. 1–9, 2021.
- HAYKIN, S. **Neural Networks and Learning Machines**. [S.l.: s.n.]: genio, 2009. v. 2.
- LI, Y.; BONE, G. M. Are parallel manipulators more energy efficient? *In: IEEE. Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515)*. [S.l.: s.n.], 2001. p. 41–46.
- MERLET, J.-P. **Parallel robots**. [S.l.: s.n.]: Springer Science & Business Media, 2005. v. 128.
- MOORI, A.; KHORAMDEL, J.; MOOSAVIAN, S. A. A. Deep learning approach for object tracking of roboeye. *In: IEEE. 2019 7th International Conference on Robotics and Mechatronics (ICRoM)*. [S.l.: s.n.], 2019. p. 386–391.
- NABAT, V. *et al.* Par4: very high speed parallel robot for pick-and-place. *In: IEEE. 2005 IEEE/RSJ International Conference on intelligent robots and systems*. [S.l.: s.n.], 2005. p. 553–558.
- PACCOT, F.; ANDREFF, N.; MARTINET, P. A review on the dynamic control of parallel kinematic machines: Theory and experiments. **The International Journal of Robotics Research**, SAGE Publications Sage UK: London, England, v. 28, n. 3, p. 395–416, 2009.

PARKER, D. Optimal algorithms for adaptive networks: Second order back propagation, second order direct propagation, and second order hebbian learning. **IEEE 1st International Conference on Neural Networks**, v. 2, p. 593–600, 1987.

RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation function for artificial neural network. *In*: IEEE. **2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)**. [*S.l.: s.n.*], 2020. p. 281–286.

SILVA, M. M. D. *et al.* Integrating structural and input design of a 2-dof high-speed parallel manipulator: A flexible model-based approach. **Mechanism and Machine Theory**, Elsevier, v. 45, n. 11, p. 1509–1519, 2010.

WANG, J. *et al.* Dynamic feed-forward control of a parallel kinematic machine. **Mechatronics**, Elsevier, v. 19, n. 3, p. 313–324, 2009.

YU, Y.-Q. *et al.* An experimental study on the dynamics of a 3-rrr flexible parallel robot. **IEEE Transactions on Robotics**, IEEE, v. 27, n. 5, p. 992–997, 2011.

ZHANG, Q. *et al.* Dynamic model and input shaping control of a flexible link parallel manipulator considering the exact boundary conditions. **Robotica**, Cambridge University Press, v. 33, n. 6, p. 1201–1230, 2015.

APÊNDICES

APÊNDICE A – MATRIZ DE ERRO (MSE) PARA TREINAMENTO COM UMA TRAJETÓRIA, RNA DE MÚLTIPLAS CAMADAS

Figura 33 – Parte I da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: Matriz_nn 29/01/2023					Page 1 11:40:31 PM
	1	2	3	4	5
1	4.7033e-08	0.0652	0.9396	0.8421	0.1704
2	0.0371	5.3394e-08	3.8692	0.4215	0.0289
3	0.1283	0.0592	5.3813e-05	0.4879	0.0472
4	0.2069	0.0356	0.7839	1.1695e-06	0.0603
5	0.1490	0.0691	1.1849	0.0847	6.1028e-08
6	0.1272	0.0646	0.3262	0.0665	0.0950
7	0.1107	0.1097	0.7572	0.7741	0.1582
8	0.0425	0.0697	1.4191	0.3450	0.0721
9	0.0894	0.0356	1.3538	0.2690	0.0424
10	0.0730	0.0757	0.2740	0.3697	0.0599
11	0.1113	0.0881	4.2859	0.0931	0.0841
12	0.0421	0.0921	0.7568	0.3542	0.0311
13	0.0750	0.0675	1.1547	0.2925	0.0658
14	0.1387	0.0789	2.2410	0.1765	0.2321
15	0.1238	0.0722	1.0277	0.4222	0.2071
16	0.0746	0.0351	0.9209	0.2798	0.2062
17	0.1042	0.0642	1.1925	0.4043	0.2272
18	0.0813	0.0518	1.2230	0.3936	0.2836
19	0.0941	0.0519	1.0842	0.4093	0.2464
20	0.1302	0.0797	0.9005	0.2656	0.3120
21	0.0711	0.0524	1.1290	0.4112	0.2811
22	0.0763	0.0622	4.2045	0.4053	0.4426
23	0.0700	0.0702	2.9549	0.4833	0.3484

Figura 34 – Parte II da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: Matriz_nn 29/01/2023						Page 2 11:40:31 PM
	6	7	8	9	10	
1	0.0244	0.8478	0.0659	0.1948	0.0312	
2	0.0773	0.3006	0.0690	0.0589	0.0183	
3	0.0344	0.1937	0.0550	0.0617	0.0282	
4	0.0411	0.1385	0.0363	0.0678	0.0522	
5	0.0836	0.4098	0.0497	0.0548	0.0095	
6	6.1598e-08	0.2190	0.0571	0.0635	0.0210	
7	0.0559	4.3416e-07	0.0777	0.0899	0.0279	
8	0.0693	0.2067	8.5751e-08	0.0525	0.0350	
9	0.0259	0.1924	0.0771	6.2767e-08	0.0076	
10	0.0365	0.1704	0.0695	0.0877	8.7958e-08	
11	0.0287	0.4032	0.0694	0.0614	0.0077	
12	0.0388	0.3870	0.1086	0.1142	0.0222	
13	0.0742	0.2420	0.1152	0.0846	0.0679	
14	0.2791	0.2174	0.1760	0.2024	0.1624	
15	0.1612	0.7280	0.2142	0.1711	0.1011	
16	0.1269	0.8475	0.1282	0.1103	0.0805	
17	0.2108	1.1450	0.2108	0.1878	0.1077	
18	0.2123	1.0933	0.2098	0.1289	0.1025	
19	0.2309	1.1554	0.2105	0.1498	0.1081	
20	0.1880	0.7016	0.2149	0.1561	0.1377	
21	0.2133	1.0187	0.2078	0.1315	0.0842	
22	0.1751	0.7974	0.1720	0.2115	0.0682	
23	0.1866	0.9037	0.1792	0.2525	0.0704	

Fonte: Próprio Autor

Figura 35 – Parte III da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: Matriz_nn 29/01/2023					Page 3 11:40:31 PM
	11	12	13	14	15
1	0.1552	0.0485	0.3677	0.0959	0.1407
2	0.0632	0.0859	0.1746	0.1217	0.1583
3	0.0526	0.0725	0.1537	0.1577	0.2068
4	0.0530	0.0310	0.2389	0.1243	0.1768
5	0.2621	0.0227	0.3491	0.0844	0.2240
6	0.2502	0.0418	0.4137	0.1164	0.4109
7	0.1374	0.0907	0.3881	0.1744	0.3125
8	0.1370	0.0603	0.2813	0.1391	0.3609
9	0.0611	0.0305	0.2515	0.0831	0.1983
10	0.2231	0.0480	0.3056	0.0773	0.2095
11	7.7041e-08	0.0223	0.3211	0.0584	0.1607
12	0.1649	4.7606e-08	0.0878	0.0789	0.0664
13	0.0640	0.0341	7.3185e-08	0.1225	0.0678
14	0.0843	0.0771	0.0750	1.9424e-08	0.1130
15	0.1292	0.0690	0.3164	0.0375	1.0890e-08
16	0.2375	0.0680	0.6011	0.0883	0.0473
17	0.0941	0.0757	0.3382	0.0395	0.0466
18	0.0691	0.0717	0.4204	0.0506	0.0568
19	0.1086	0.0735	0.3686	0.0423	0.0473
20	0.0913	0.1123	0.3223	0.0450	0.0391
21	0.0792	0.0669	0.4250	0.0505	0.0616
22	0.7937	0.0358	0.3892	0.0316	0.0345
23	0.4220	0.0397	0.3924	0.0387	0.0500

Fonte: Próprio Autor

Figura 36 – Parte IV da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: Matriz_nn 29/01/2023					Page 4 11:40:31 PM
	16	17	18	19	20
1	0.0316	0.0526	0.0418	0.0186	0.0685
2	0.1932	0.0409	0.0584	0.0312	0.0676
3	0.2360	0.0774	0.0938	0.1106	0.0883
4	0.2092	0.0505	0.0812	0.0391	0.0774
5	0.2590	0.0881	0.0963	0.0462	0.1251
6	0.3416	0.1359	0.1194	0.0570	0.1510
7	0.3448	0.1933	0.1675	0.1612	0.2112
8	0.3361	0.1469	0.1504	0.1001	0.1929
9	0.2331	0.0838	0.0793	0.0453	0.1062
10	0.2441	0.0811	0.0882	0.0405	0.1162
11	0.1211	0.0654	0.0631	0.0252	0.0890
12	0.0239	0.0590	0.0577	0.0230	0.0717
13	0.0184	0.0598	0.0668	0.0337	0.0862
14	0.1594	0.0679	0.1041	0.0710	0.0968
15	0.0141	0.0099	0.0149	0.0071	0.0023
16	9.6600e-09	0.0473	0.0078	0.0129	0.0113
17	0.0326	2.2424e-08	0.0073	0.0151	0.0014
18	0.2549	0.0252	2.7505e-08	0.0085	0.0046
19	0.3116	0.0058	0.0117	2.3406e-08	0.0020
20	0.0943	0.0029	0.0160	0.0135	6.0815e-08
21	0.2804	0.0255	0.0018	0.0116	0.0049
22	0.0616	0.0374	0.0109	0.0129	0.0197
23	0.1400	0.0356	0.0110	0.0098	0.0167

Fonte: Próprio Autor

Figura 37 – Parte V da matriz de erro (MSE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: Matriz_nn 29/01/2023		Page 5 11:40:31 PM	
	21	22	23
1	0.0128	0.0126	0.0737
2	0.0659	0.0611	0.0785
3	0.1079	0.0988	0.0888
4	0.0751	0.0731	0.0738
5	0.0672	0.1000	0.1514
6	0.0939	0.1242	0.1969
7	0.0816	0.1587	0.2398
8	0.1064	0.1538	0.2152
9	0.0548	0.0833	0.1309
10	0.0593	0.0902	0.1424
11	0.0359	0.0594	0.1113
12	0.0146	0.0202	0.0608
13	0.0213	0.0253	0.0361
14	0.1604	0.1280	0.2244
15	0.0446	0.0159	0.0277
16	0.0225	0.0214	0.0339
17	0.0100	0.0079	0.0215
18	0.0048	0.0046	0.0101
19	0.0190	0.0060	0.0154
20	0.0137	0.0111	0.0208
21	1.5241e-08	0.0050	0.0144
22	0.0107	2.1896e-08	0.0418
23	0.0308	0.0174	2.5966e-08

Fonte: Próprio Autor

APÊNDICE B – MATRIZ DE ERRO (MAPE) PARA TREINAMENTO COM UMA TRAJETÓRIA, RNA DE MÚLTIPLAS CAMADAS

Figura 38 – Parte I da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 29/01/2023					Page 1 11:36:07 PM
	1	2	3	4	5
1	0.0374	27.4039	63.6614	59.4190	44.8538
2	32.6257	0.0502	93.5309	57.1340	27.3957
3	47.1030	41.1297	1.4841	62.8115	45.6281
4	34.7170	33.2497	124.8886	0.0651	40.5542
5	41.4847	42.3496	105.0310	50.0170	0.0415
6	36.1559	37.3438	78.7815	22.9469	105.6982
7	47.2766	52.7649	105.4657	77.1904	67.7594
8	37.5772	48.7036	95.3849	48.4346	97.1447
9	36.5584	30.9946	95.7819	69.9401	42.2328
10	37.4647	40.6507	68.7557	76.6378	90.4572
11	34.5060	43.4316	97.7403	56.8869	35.2002
12	28.7999	49.2081	117.7910	70.5286	41.3077
13	37.0414	56.7587	88.2041	75.3103	51.1263
14	47.3725	57.4347	63.8552	44.8353	173.3299
15	181.8978	68.8322	78.7805	166.0122	176.6995
16	151.7140	106.5034	92.0345	98.0095	186.9922
17	165.9669	67.6693	91.3999	107.2602	85.9247
18	142.8085	76.6400	129.1599	121.0312	165.5743
19	156.0730	71.3830	104.4851	145.6913	174.1985
20	159.6172	67.4845	94.3746	94.4623	181.7681
21	134.2521	77.4610	115.1011	126.6398	175.6884
22	165.2784	100.7998	77.2579	93.9397	67.4315
23	151.2062	98.8875	77.9433	92.3415	70.0767

Figura 39 – Parte II da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 29/01/2023					Page 2 11:36:07 PM
	6	7	8	9	10
1	25.7999	107.9135	73.6072	58.2761	27.8494
2	49.8374	57.4363	59.0008	33.9426	27.7622
3	40.5497	55.4972	55.6063	69.5823	44.9769
4	37.0343	50.1909	114.6039	123.8088	33.3658
5	30.8609	48.3090	39.2914	30.5515	13.8528
6	0.0334	38.3196	42.6891	87.9148	19.4159
7	29.3917	0.0740	60.4812	63.3222	21.6179
8	34.2705	43.9794	0.0427	139.2211	29.0240
9	23.9909	42.2790	35.5594	0.0421	16.3416
10	28.1838	37.2071	32.0599	26.7522	0.0474
11	30.9452	48.5249	32.0517	50.8401	21.5534
12	32.2811	74.7421	53.7228	38.0412	35.7020
13	45.0455	74.8319	60.0740	43.0273	60.5646
14	103.7279	53.1210	776.2370	336.5429	85.2397
15	64.6096	78.5130	132.9946	98.6379	41.5208
16	103.4316	78.9611	96.4569	89.0153	50.8379
17	139.3780	81.2509	128.8346	61.1259	45.2311
18	158.7572	77.4729	145.6883	62.9656	52.6972
19	147.7591	79.6650	138.5027	75.8315	47.9379
20	65.2789	77.9748	130.2221	59.5634	73.4891
21	176.2021	74.3645	150.2822	55.8360	55.8749
22	270.0284	64.8423	101.7395	62.9808	56.2869
23	262.2584	67.0598	96.4002	63.2501	56.0340

Fonte: Próprio Autor

Figura 40 – Parte III da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 29/01/2023					Page 3 11:36:07 PM
	11	12	13	14	15
1	96.7368	35.8733	62.7064	192.4664	285.0924
2	33.5436	39.6430	43.3555	574.2944	576.5110
3	37.7600	42.9240	48.4266	599.4376	610.9440
4	26.3080	30.5924	46.9009	212.7300	605.0914
5	40.1067	28.9063	54.1598	118.2275	726.2895
6	92.3125	31.5046	50.6543	206.7111	300.9136
7	60.8208	52.1553	59.5229	1.1744e+03	594.1893
8	70.4619	42.7874	61.3010	144.2244	464.2136
9	40.3442	21.9702	48.2897	111.4813	679.4009
10	72.1488	42.4428	49.5484	113.1110	703.9565
11	0.0430	20.2375	45.0275	92.5613	619.4435
12	88.0847	0.0385	37.0811	92.5504	145.2144
13	51.1392	32.9503	0.0605	100.7892	135.1357
14	48.9827	42.7915	40.8992	0.0391	144.4497
15	129.2104	46.6706	63.9243	50.5643	0.0335
16	81.0379	48.8032	70.9363	78.5598	41.1399
17	45.9676	48.4090	63.2282	50.1256	46.6499
18	42.1228	48.8292	65.4684	40.4146	41.2712
19	149.9704	49.3033	62.2849	49.0903	42.5385
20	48.8408	57.3229	69.5321	55.6319	46.6105
21	61.1883	47.5090	62.9282	36.9280	39.3843
22	77.0150	32.3251	51.5729	30.5480	38.7591
23	77.6239	34.8366	53.4698	32.3067	41.5000

Fonte: Próprio Autor

Figura 41 – Parte IV da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p						Page 4
29/01/2023						11:36:07 PM
	16	17	18	19	20	
1	34.8560	154.2314	113.9408	43.4641	1.9846e+03	
2	118.0910	102.3555	201.0403	58.7322	934.1913	
3	131.8708	179.6318	336.4631	178.3770	970.3113	
4	124.1608	117.3112	194.7535	65.8069	999.6277	
5	137.7977	167.7940	163.9108	59.0744	2.6359e+03	
6	109.6708	412.2289	176.0964	65.0810	4.8394e+03	
7	144.3151	821.3087	222.1135	166.8973	2.6821e+04	
8	158.1334	221.0902	207.6869	101.1718	2.4806e+04	
9	130.0974	240.7917	149.1066	68.6252	2.4409e+03	
10	132.9077	170.3912	156.1715	54.9256	2.5495e+03	
11	151.9808	207.6626	137.0831	45.0454	2.2382e+03	
12	13.3189	153.3823	206.8173	45.2672	3.8743e+03	
13	24.1734	64.2444	417.6604	55.7813	1.0411e+03	
14	73.0479	181.7010	71.8914	66.6352	5.2133e+03	
15	38.2638	75.9383	46.7301	34.3681	43.0750	
16	0.0318	110.4112	39.3433	26.0262	219.2181	
17	39.1779	0.1082	37.6303	36.9043	74.4322	
18	48.3645	93.6596	0.0488	36.4413	51.6281	
19	53.5779	74.1584	42.9981	0.1017	14.3648	
20	45.2842	45.2765	51.1032	47.5939	0.4070	
21	55.7451	97.0114	12.1532	41.3440	59.7922	
22	32.5730	115.6908	42.6596	44.3933	174.3429	
23	38.8949	101.8580	27.8025	26.7468	190.1499	

Fonte: Próprio Autor

Figura 42 – Parte V da matriz de erro (MAPE) para treinamento com uma trajetória, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 29/01/2023				Page 5 11:36:07 PM
	21	22	23	
1	29.0082	30.8682	183.1177	
2	96.8686	97.2031	135.3942	
3	128.0905	114.9562	128.5998	
4	101.1266	105.8578	113.1493	
5	76.3725	127.3961	250.9101	
6	86.4480	134.6929	290.6549	
7	67.7884	153.0382	324.9814	
8	94.0932	157.2925	297.0443	
9	67.5648	116.1866	232.6173	
10	71.9272	120.6695	244.9666	
11	53.5691	95.7155	222.4357	
12	27.7734	37.9850	161.4496	
13	39.5835	46.5292	110.5049	
14	103.5449	77.9857	251.1527	
15	72.0836	42.3043	64.6374	
16	29.3363	31.8632	61.5338	
17	40.6118	37.3362	52.4409	
18	26.2780	33.9273	38.8143	
19	57.3852	34.4503	49.1117	
20	51.1879	41.9240	60.0973	
21	0.0478	37.4611	59.5661	
22	35.2173	0.0348	85.1495	
23	61.7916	26.4802	0.0454	

Fonte: Próprio Autor

APÊNDICE C – MATRIZ DE ERRO (MSE) PARA TREINAMENTO COM MÚLTIPLAS TRAJETÓRIAS, RNA DE MÚLTIPLAS CAMADAS

Figura 43 – Parte I da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas

MATLAB Variable: Matriz_nn 30/01/2023					Page 1 3:12:39 AM
	1	2	3	4	5
1	1.1771e-06	8.3904e-06	8.9484e-05	7.3840e-05	9.5621e-05
2	0.0501	1.2534e-05	5.1347e-05	4.7809e-05	7.4296e-05
3	0.0625	0.0944	4.9643e-04	9.2727e-05	4.8629e-04
4	0.2870	0.0772	1.7261	4.3332e-05	7.3242e-05
5	0.2918	0.1240	1.4862	0.1155	9.3783e-05
6	0.1932	0.0520	0.1988	6.5518	0.0132
7	0.3074	0.1606	1.0079	2.3060	0.1024
8	0.4606	0.1122	0.2010	5.2802	0.0514
9	0.5119	0.0760	1.4833	0.2288	0.0344
10	0.4512	0.0767	1.3355	0.1670	0.2057
11	0.2027	0.0942	1.3109	0.2146	0.0275
12	0.2263	0.2843	1.1655	98.8294	2.3635
13	0.2097	0.1932	1.3884	5.2344	1.1116
14	0.0662	0.1066	0.1534	5.8685	0.4998
15	0.1742	0.3341	1.9421	175.0147	0.3825
16	0.1453	0.6359	1.7426	42.8841	0.2843
17	0.3535	0.2219	1.9698	175.6254	0.3947
18	0.2888	0.2636	1.9234	185.3501	0.6177
19	0.3401	0.2118	1.9237	184.6722	0.4283
20	0.3958	0.3427	2.3888	114.1526	0.5185
21	0.2560	0.2106	1.8694	185.5249	0.6472
22	0.0439	0.6830	2.3632	40.8424	0.3900
23	0.0389	0.6283	2.0994	94.8289	0.4553

Figura 44 – Parte II da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas

MATLAB Variable: Matriz_nn 30/01/2023						Page 2 3:12:39 AM
	6	7	8	9	10	
1	6.4492e-05	9.0751e-05	1.0344e-04	9.9177e-05	9.2754e-05	
2	6.2623e-05	7.2893e-05	7.3853e-05	7.0566e-05	8.0026e-05	
3	1.8836e-04	2.5531e-04	2.8672e-04	3.2679e-04	3.5784e-04	
4	7.1591e-05	7.7794e-05	7.0025e-05	7.3125e-05	8.8267e-05	
5	9.8780e-05	1.0902e-04	9.5091e-05	1.0932e-04	1.0965e-04	
6	5.7432e-05	6.0662e-05	6.6330e-05	6.6072e-05	7.2296e-05	
7	0.9524	7.2796e-05	8.5527e-05	9.0011e-05	8.8160e-05	
8	0.0729	0.0235	7.6215e-05	8.4350e-05	6.9829e-05	
9	0.0030	5.8244e-04	0.0125	7.9290e-05	8.3818e-05	
10	0.0017	0.0039	0.0138	0.0039	8.8090e-05	
11	0.0049	0.0233	0.4725	0.0480	0.0902	
12	0.1508	2.6639	0.4574	0.0436	1.8386	
13	0.4386	1.1264	0.7830	0.0154	4.0021	
14	0.1765	0.4973	0.0714	0.1264	0.1979	
15	0.1954	90.7457	3.4520	0.1613	0.2308	
16	0.0972	85.2699	2.7557	0.0703	0.0845	
17	0.1845	108.9494	16.0365	0.1240	0.2352	
18	0.1928	110.8403	8.6725	0.2322	0.0906	
19	0.1836	110.9995	20.4760	0.1209	0.1038	
20	0.1931	98.1680	2.6406	0.1436	0.1446	
21	0.1973	112.4852	13.2051	0.3253	0.0683	
22	0.1487	103.7239	18.5913	0.1121	2.4179	
23	0.1839	108.1852	16.0648	0.2289	1.3959	

Fonte: Próprio Autor

Figura 45 – Parte III da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas

MATLAB Variable: Matriz_nn 30/01/2023					Page 3 3:12:39 AM
	11	12	13	14	15
1	9.6325e-05	1.0007e-04	8.9982e-05	9.4657e-05	1.0367e-04
2	9.2536e-05	8.4013e-05	1.0619e-04	7.9416e-05	8.3950e-05
3	3.6234e-04	3.3421e-04	5.4359e-04	3.2415e-04	4.8478e-04
4	9.3121e-05	8.8380e-05	9.2295e-05	8.3462e-05	9.7028e-05
5	1.0338e-04	1.1388e-04	1.3088e-04	1.1959e-04	1.2247e-04
6	7.0172e-05	7.6814e-05	9.2977e-05	8.2049e-05	8.0406e-05
7	7.6860e-05	8.3893e-05	1.1596e-04	1.0315e-04	9.8273e-05
8	7.9387e-05	8.7496e-05	9.3614e-05	8.7709e-05	9.5344e-05
9	7.9340e-05	8.5583e-05	9.3754e-05	8.4903e-05	9.1345e-05
10	9.0679e-05	9.9270e-05	1.0279e-04	9.6671e-05	9.8523e-05
11	7.7719e-05	8.0913e-05	9.3738e-05	8.7149e-05	8.7588e-05
12	3.8975	6.6250e-05	7.5163e-05	1.0088e-04	8.0382e-05
13	4.4929	0.2456	7.8820e-05	8.0604e-05	7.4132e-05
14	0.1570	0.2061	0.4580	7.8397e-06	6.5467e-06
15	2.6801	0.0935	0.1576	1.0034	3.9195e-06
16	1.0621	0.1084	0.0316	1.1666	0.0144
17	1.3942	0.1178	0.0953	0.6349	0.2126
18	2.4069	0.2377	0.0717	0.3506	0.0107
19	1.4150	0.1565	0.0901	0.3918	0.0082
20	1.6785	0.4930	0.3342	0.3266	1.8094
21	3.1584	0.1418	0.0896	0.3439	0.0058
22	1.9859	0.2566	0.1075	0.3826	0.2937
23	3.4151	0.2058	0.1044	0.4295	0.0333

Fonte: Próprio Autor

Figura 46 – Parte IV da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas

MATLAB Variable: Matriz_nn 30/01/2023					Page 4 3:12:39 AM
	16	17	18	19	20
1	9.9733e-05	1.0971e-04	1.1807e-04	1.1097e-04	1.1603e-04
2	1.0167e-04	8.3920e-05	8.6135e-05	1.1739e-04	1.1554e-04
3	3.6695e-04	3.2687e-04	5.4471e-04	5.7558e-04	5.9877e-04
4	1.0261e-04	9.4742e-05	9.9685e-05	1.0176e-04	1.0996e-04
5	1.2682e-04	1.2300e-04	1.2915e-04	1.2758e-04	1.2654e-04
6	8.2256e-05	8.4737e-05	8.7291e-05	9.3196e-05	9.1678e-05
7	1.0773e-04	1.0221e-04	1.1813e-04	1.4006e-04	1.0343e-04
8	9.0143e-05	9.5752e-05	9.5907e-05	1.0214e-04	1.0096e-04
9	9.5740e-05	9.2887e-05	9.9468e-05	1.0251e-04	9.4307e-05
10	1.0235e-04	1.0278e-04	1.0686e-04	1.0553e-04	1.0504e-04
11	8.9282e-05	8.7734e-05	9.5708e-05	1.0765e-04	9.9347e-05
12	9.7690e-05	9.3588e-05	9.5324e-05	1.0199e-04	1.0578e-04
13	9.4981e-05	8.4617e-05	9.6791e-05	1.0952e-04	9.7166e-05
14	1.2864e-05	1.0026e-05	1.3297e-05	1.1280e-05	1.4814e-05
15	1.8207e-06	3.1729e-06	5.1772e-06	5.2365e-06	6.2170e-06
16	1.7709e-06	3.2594e-06	6.2625e-06	7.6670e-06	5.0268e-06
17	0.0146	1.0139e-05	1.1769e-05	1.1810e-05	9.5087e-06
18	0.0024	0.0015	1.4324e-05	1.1486e-05	9.9562e-06
19	3.3873e-04	2.4785e-04	2.3001e-04	1.0378e-05	8.0664e-06
20	0.0151	5.0013e-04	0.0056	0.0183	5.5982e-06
21	0.0028	0.0014	1.0801e-04	1.3619e-04	8.7444e-05
22	0.0301	0.0070	0.0027	0.0038	0.0034
23	0.0208	0.0050	0.0030	0.0042	0.0026

Fonte: Próprio Autor

Figura 47 – Parte V da matriz de erro (MSE) para treinamento com múltiplas trajetórias, RNA de Múltipla Camadas

MATLAB Variable: Matriz_nn 30/01/2023			Page 5 3:12:39 AM
	21	22	23
1	1.1370e-04	1.1863e-04	1.1829e-04
2	1.1489e-04	9.8876e-05	1.0260e-04
3	6.2134e-04	3.8040e-04	3.9638e-04
4	1.0285e-04	1.0465e-04	1.0676e-04
5	1.2628e-04	1.3489e-04	1.3035e-04
6	9.1514e-05	8.3163e-05	8.2154e-05
7	1.2183e-04	1.1016e-04	9.9617e-05
8	1.0572e-04	9.4486e-05	8.6820e-05
9	9.5043e-05	9.7145e-05	1.0120e-04
10	1.0136e-04	9.7101e-05	1.0401e-04
11	9.5007e-05	9.2119e-05	9.1878e-05
12	9.8826e-05	9.6037e-05	9.6487e-05
13	9.4532e-05	8.9695e-05	1.0770e-04
14	1.4828e-05	9.6486e-06	1.2604e-05
15	5.2686e-06	4.9447e-06	5.7317e-06
16	3.6247e-06	8.1185e-06	9.6594e-06
17	1.6741e-05	1.4504e-05	5.1975e-06
18	1.1121e-05	1.7976e-05	1.7058e-05
19	1.5337e-05	5.0280e-06	8.1206e-06
20	5.3301e-06	3.8920e-06	7.2221e-06
21	5.0545e-06	6.5611e-06	6.1760e-06
22	0.0054	6.4866e-06	5.7226e-06
23	0.0147	0.0010	6.8035e-06

Fonte: Próprio Autor

APÊNDICE D – MATRIZ DE ERRO (MAPE) PARA TREINAMENTO COM MÚLTIPLAS TRAJETÓRIAS, RNA DE MÚLTIPLAS CAMADAS

Figura 48 – Parte I da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 30/01/2023					Page 1 3:10:59 AM
	1	2	3	4	5
1	0.1899	0.4888	1.4249	1.3618	1.4540
2	39.2765	0.7738	1.6753	1.5581	2.0569
3	43.5902	44.3218	3.9737	2.8711	4.2350
4	48.7973	36.1658	72.9742	1.5430	2.1209
5	58.9122	37.3872	73.8999	34.3394	1.6897
6	43.1713	33.7906	41.8029	65.2916	22.4857
7	67.6743	57.9049	58.0459	61.6787	32.4267
8	58.2380	45.3555	55.9199	64.7550	28.7590
9	44.8601	32.6790	88.4239	39.0081	31.0675
10	43.8547	30.0300	69.7898	36.1125	59.7857
11	47.1543	39.9426	42.5457	33.8781	52.4579
12	38.0083	67.2180	38.9140	67.2433	74.9509
13	43.6676	88.3901	59.8731	99.3792	75.8133
14	45.5022	122.2403	188.6309	90.6680	100.6189
15	64.9522	228.9426	68.7388	79.4340	119.8634
16	53.6906	150.1792	72.3527	68.0697	93.3868
17	60.6174	340.8394	57.7761	91.3694	109.8154
18	56.9988	275.3444	70.4067	70.8669	93.7448
19	57.2385	288.3869	60.0580	73.9596	102.6371
20	64.0332	285.8941	85.0527	76.8623	110.5797
21	55.9789	316.5781	71.5006	63.8891	87.4630
22	46.1112	194.7940	68.7284	80.2238	103.4244
23	39.7479	198.3786	57.9000	65.9514	95.8858

Figura 49 – Parte II da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 30/01/2023						Page 2 3:10:59 AM
	6	7	8	9	10	
1	1.2225	1.4522	1.5991	1.4926	1.4928	
2	1.9017	2.0564	2.0988	2.0395	2.1284	
3	4.2610	4.1015	3.5670	4.8008	3.6168	
4	2.0950	2.1912	2.1136	2.1313	2.4150	
5	1.7299	1.8653	1.7716	1.8414	1.8864	
6	1.1532	1.2317	1.2800	1.2922	1.3579	
7	65.8040	1.0465	1.1959	1.1872	1.1759	
8	34.2418	23.6910	1.4906	1.4909	1.3654	
9	12.0689	4.9670	28.0221	1.6463	1.7242	
10	7.8661	17.9360	35.0450	17.6177	1.6116	
11	13.8089	23.1512	53.1299	25.6961	26.0279	
12	38.9990	71.7524	59.0744	37.4295	55.2574	
13	64.1427	88.6381	66.8172	29.8407	66.2216	
14	82.3422	581.2821	38.5389	41.9275	173.9285	
15	109.5156	128.7874	84.7187	77.1482	59.7253	
16	59.7483	76.4564	90.4960	76.9787	48.8093	
17	119.0228	111.3816	87.6543	90.1934	56.9310	
18	123.8895	113.8890	85.6772	68.0571	49.7863	
19	125.4135	110.1595	87.7590	63.0545	49.5489	
20	127.0725	218.1148	85.1035	121.1728	54.6426	
21	126.3801	115.4692	85.1586	69.7999	46.9143	
22	118.9415	131.1161	85.3882	56.2297	61.2896	
23	106.9217	131.4931	83.3015	64.1000	60.0061	

Fonte: Próprio Autor

Figura 50 – Parte III da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 30/01/2023					Page 3 3:10:59 AM
	11	12	13	14	15
1	1.5020	1.5287	1.4834	1.5435	1.5749
2	2.3575	2.2066	2.5742	2.2026	2.3261
3	4.8774	4.7484	4.2927	3.5451	3.8054
4	2.4166	2.4190	2.4273	2.3565	2.5337
5	1.8162	1.9467	2.0530	1.9625	2.0118
6	1.3467	1.3818	1.5615	1.4606	1.4269
7	1.0985	1.1849	1.3894	1.3304	1.3013
8	1.5214	1.5912	1.6157	1.5930	1.6586
9	1.6559	1.7587	1.8304	1.7247	1.7769
10	1.6732	1.7507	1.7698	1.7376	1.7536
11	1.4635	1.5096	1.5847	1.5404	1.5631
12	73.1053	1.5283	1.6102	1.8297	1.6246
13	78.5627	57.8928	2.1734	2.1618	2.0671
14	77.5921	56.8108	131.4827	0.6768	0.5406
15	130.1312	49.0904	64.7105	78.8316	0.8230
16	121.7291	43.1793	39.4229	108.4955	47.5039
17	135.9077	54.0527	54.9965	78.5685	61.5239
18	120.2081	56.1860	51.2612	74.2610	15.8012
19	133.1662	55.0281	53.0613	80.6489	37.3616
20	130.0232	62.8649	75.3427	73.1880	74.6485
21	118.3759	54.1749	51.0945	75.1339	11.2112
22	114.9305	55.1216	59.2617	76.8996	49.3299
23	117.8576	50.9815	53.5303	73.9028	28.7863

Fonte: Próprio Autor

Figura 51 – Parte IV da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 30/01/2023					Page 4 3:10:59 AM
	16	17	18	19	20
1	1.5121	1.6399	1.6873	1.6601	1.6555
2	2.4592	2.3193	2.2540	2.7304	2.6946
3	4.6056	5.6402	5.7126	5.3712	6.5093
4	2.5605	2.4741	2.5541	2.6071	2.7072
5	2.0360	1.9783	1.9952	2.0640	2.0238
6	1.4716	1.4906	1.5418	1.5782	1.5271
7	1.3292	1.2836	1.4329	1.5544	1.3315
8	1.5835	1.6885	1.7138	1.7259	1.7327
9	1.8519	1.7869	1.9246	1.9526	1.8174
10	1.7720	1.7907	1.8098	1.8288	1.8155
11	1.5749	1.5683	1.6651	1.7276	1.6775
12	1.7920	1.7727	1.7800	1.8624	1.9207
13	2.4199	2.3219	2.4148	2.5913	2.4290
14	0.7710	0.7343	0.8480	0.8801	0.8458
15	0.7512	0.6690	1.0894	1.3033	1.1195
16	0.3870	0.5208	0.5918	0.6961	0.6587
17	24.7101	1.9919	2.0790	1.9348	1.8386
18	55.9402	12.6882	1.3028	1.4351	1.1464
19	16.7416	10.3346	7.0668	2.1571	1.4124
20	48.9492	19.2471	36.6366	43.7534	2.0547
21	56.8348	9.6338	4.4357	3.2864	3.1839
22	93.2596	19.6385	17.2267	28.9175	15.2431
23	85.4243	14.4751	19.2376	24.3109	15.4826

Fonte: Próprio Autor

Figura 52 – Parte V da matriz de erro (MAPE) para treinamento com múltiplas trajetórias, RNA de Múltiplas Camadas

MATLAB Variable: DATASET_test_nn_p 30/01/2023		Page 5 3:10:59 AM		
	21	22	23	
1	1.6384	1.7141	1.6692	
2	2.5305	2.5186	2.5231	
3	5.7502	5.2409	5.1116	
4	2.6173	2.7013	2.6457	
5	2.0587	2.0452	2.0539	
6	1.5409	1.4813	1.4883	
7	1.4630	1.3633	1.2908	
8	1.7527	1.6527	1.5580	
9	1.8563	1.8301	1.8779	
10	1.7851	1.7602	1.7787	
11	1.6563	1.5947	1.6407	
12	1.8055	1.7625	1.7790	
13	2.4174	2.3332	2.5943	
14	0.7413	0.5847	0.7209	
15	0.7852	0.9427	1.1266	
16	0.5200	0.7248	0.7106	
17	2.0887	2.5750	1.3710	
18	1.1535	1.9299	1.8729	
19	2.2505	1.1417	1.9579	
20	1.9718	1.4406	2.0209	
21	0.7980	0.8887	0.9150	
22	29.4334	0.6305	0.6306	
23	31.7338	6.5300	0.5352	

Fonte: Próprio Autor

APÊNDICE E – MATRIZ DE ERRO (MSE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA FOCADA

Figura 53 – Parte I da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: Matriz_nn 30/01/2023					Page 1 5:49:05 AM
	1	2	3	4	5
1	9.5904e-07	1.1571e-05	8.9122e-05	5.3254e-05	7.8844e-05
2	0.1464	7.9166e-06	6.2637e-05	3.7747e-05	5.2792e-05
3	0.1598	0.1590	5.2414e-04	1.1449e-04	2.4683e-04
4	0.1523	0.0953	2.1104	4.4934e-05	5.3416e-05
5	0.1855	0.0878	1.6277	0.4368	8.1135e-05
6	0.2531	0.1471	0.3074	0.0916	0.1824
7	0.2943	0.0705	1.2134	0.1927	0.0812
8	0.1864	0.0735	0.1832	0.2336	0.5421
9	0.0919	0.0275	1.7620	0.7284	0.0128
10	0.2004	0.0495	1.6384	0.0967	0.0136
11	0.1270	0.2090	1.6042	0.5912	4.4245
12	0.2475	0.4581	1.4383	0.8774	5.4940
13	0.3046	1.0376	1.0559	0.0708	7.4974
14	0.1307	0.0522	0.1588	0.1414	6.5683
15	0.1689	0.1934	0.9020	0.1592	2.6384
16	0.1606	0.1088	0.7650	0.6411	1.4548
17	0.2071	0.0866	0.9086	0.3632	0.7177
18	0.3189	0.0679	0.8507	0.6542	0.3785
19	0.2596	0.0623	0.8741	0.2626	0.3142
20	0.2969	0.0998	0.9080	0.4707	0.3035
21	0.3226	0.0741	0.8320	0.4643	0.2586
22	0.2052	0.1668	1.2391	0.1684	1.3781
23	0.2571	0.1207	1.0736	0.2605	0.8929

Figura 54 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: Matriz_nn 30/01/2023						Page 2 5:49:05 AM
	6	7	8	9	10	
1	5.5340e-05	9.2521e-05	1.0371e-04	9.2049e-05	1.0181e-04	
2	5.4921e-05	8.0486e-05	1.0828e-04	7.1059e-05	9.4864e-05	
3	1.0610e-04	5.7305e-04	6.4811e-04	4.6388e-04	5.3157e-04	
4	6.8804e-05	8.3628e-05	9.7876e-05	7.9126e-05	9.3808e-05	
5	8.4647e-05	1.1087e-04	1.2349e-04	1.1163e-04	1.0585e-04	
6	5.5224e-05	7.1823e-05	8.6942e-05	6.8993e-05	7.5647e-05	
7	2.4092	9.2957e-05	1.1841e-04	1.0086e-04	8.9482e-05	
8	0.1175	0.0296	9.4377e-05	8.5959e-05	7.9709e-05	
9	0.0013	0.0092	3.9561e-04	7.8655e-05	8.6677e-05	
10	6.2907e-04	0.0020	3.1602e-04	9.8776e-04	1.0238e-04	
11	2.9976	0.0115	0.0054	0.2975	0.1920	
12	8.0733	0.0460	0.0952	0.9626	0.1658	
13	81.5222	0.1809	0.1108	0.4092	0.2331	
14	2.4088	0.1712	0.1142	0.2471	0.1434	
15	4.3195	0.2229	0.0858	0.6849	0.2280	
16	46.1948	0.1924	0.0832	0.3464	0.2259	
17	1.1318	0.1899	0.1205	1.3733	0.1793	
18	0.8130	0.2665	0.0907	1.4686	0.1586	
19	0.8545	0.2278	0.0934	1.4201	0.1814	
20	11.0722	0.7467	0.1052	0.6271	0.2039	
21	1.2971	0.2777	0.0911	1.5821	0.1545	
22	28.6520	0.1365	0.0937	1.5063	0.1281	
23	36.0126	0.1853	0.0952	1.7662	0.1414	

Fonte: Próprio Autor

Figura 55 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: Matriz_nn 30/01/2023					Page 3 5:49:05 AM
	11	12	13	14	15
1	1.0700e-04	9.9339e-05	1.0489e-04	1.1008e-04	1.1883e-04
2	9.5288e-05	6.5239e-05	8.8987e-05	1.0214e-04	1.2304e-04
3	3.8901e-04	4.8711e-04	4.3050e-04	5.3056e-04	6.8736e-04
4	1.0213e-04	8.7512e-05	7.9612e-05	1.0052e-04	1.0382e-04
5	1.2599e-04	1.1767e-04	1.1469e-04	1.2285e-04	1.3987e-04
6	7.8510e-05	6.5975e-05	8.2000e-05	8.9872e-05	8.9589e-05
7	1.1514e-04	9.4735e-05	8.6247e-05	9.1700e-05	1.0400e-04
8	8.9354e-05	8.5752e-05	8.0251e-05	9.7774e-05	1.0557e-04
9	9.7289e-05	8.1178e-05	8.4800e-05	9.0800e-05	9.6850e-05
10	1.0033e-04	8.4366e-05	9.8650e-05	1.0155e-04	1.1357e-04
11	8.8190e-05	8.9734e-05	9.2573e-05	8.4464e-05	9.5012e-05
12	0.5954	7.2978e-05	9.6323e-05	8.7188e-05	9.8330e-05
13	0.0287	0.0371	8.7002e-05	7.9916e-05	1.0372e-04
14	0.2558	0.1317	0.7321	1.1308e-05	9.7656e-06
15	0.7063	0.5508	3.7275	0.4725	4.7064e-06
16	4.8142	0.1760	1.2452	0.2432	0.0097
17	0.4462	0.1226	0.7695	0.4139	0.0035
18	0.0912	0.1569	0.6407	0.7054	0.0063
19	0.0891	0.1220	0.5773	0.7236	9.8101e-04
20	0.1655	0.3292	1.3285	0.9437	0.0063
21	0.0740	0.2142	0.6664	0.3412	0.0043
22	5.5473	0.3016	2.7808	0.1179	0.0173
23	4.9415	0.2280	2.0819	0.1594	0.0107

Fonte: Próprio Autor

Figura 56 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: Matriz_nn 30/01/2023					Page 4 5:49:05 AM
	16	17	18	19	20
1	1.2207e-04	1.1555e-04	1.1888e-04	1.1683e-04	1.1305e-04
2	9.2225e-05	1.1276e-04	1.0544e-04	9.9276e-05	1.1361e-04
3	5.1395e-04	5.9392e-04	4.5725e-04	6.0997e-04	6.3351e-04
4	9.7283e-05	1.0277e-04	9.9075e-05	1.0925e-04	1.0717e-04
5	1.2631e-04	1.3008e-04	1.2179e-04	1.3026e-04	1.3702e-04
6	7.9485e-05	9.2643e-05	8.2905e-05	9.3048e-05	9.0368e-05
7	9.0202e-05	1.3115e-04	1.2007e-04	1.1314e-04	1.2151e-04
8	9.4177e-05	1.0293e-04	9.9917e-05	1.0131e-04	1.0739e-04
9	9.5471e-05	9.9818e-05	9.2532e-05	8.8860e-05	9.6608e-05
10	1.0013e-04	1.0819e-04	1.0297e-04	9.9000e-05	1.0846e-04
11	8.3629e-05	1.0338e-04	8.3571e-05	1.1707e-04	1.0649e-04
12	9.1930e-05	9.8741e-05	9.6546e-05	9.8280e-05	9.8840e-05
13	7.3967e-05	9.9284e-05	8.7587e-05	9.0430e-05	9.6823e-05
14	1.0478e-05	9.8917e-06	8.0805e-06	1.2301e-05	1.0946e-05
15	3.8421e-06	4.4585e-06	6.0782e-06	8.7931e-06	3.5025e-06
16	2.7027e-06	2.8547e-06	2.9264e-06	4.1335e-06	5.2404e-06
17	0.0023	7.6552e-06	6.9351e-06	9.3336e-06	9.6308e-06
18	0.0110	0.0040	1.0656e-05	1.2735e-05	1.1086e-05
19	0.0011	0.0021	9.0294e-05	1.2700e-05	5.7411e-06
20	0.0034	8.5097e-04	0.0058	0.0101	5.1515e-06
21	0.0119	0.0062	7.8291e-05	6.5663e-05	6.1149e-05
22	0.1276	0.0095	0.0055	0.0263	0.0680
23	0.1059	0.0041	0.0020	0.0286	0.1067

Fonte: Próprio Autor

Figura 57 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: Matriz_nn 30/01/2023		Page 5 5:49:05 AM		
	21	22	23	
1	1.1547e-04	1.1838e-04	1.2052e-04	
2	1.1197e-04	1.1256e-04	1.5145e-04	
3	5.6986e-04	0.0015	9.0852e-04	
4	9.7755e-05	1.0625e-04	1.2011e-04	
5	1.2653e-04	1.4139e-04	1.4111e-04	
6	8.3226e-05	8.7996e-05	1.1455e-04	
7	1.0494e-04	1.1420e-04	1.4057e-04	
8	9.4294e-05	9.8428e-05	1.1767e-04	
9	9.0758e-05	1.0304e-04	1.1019e-04	
10	1.0462e-04	1.1040e-04	1.1745e-04	
11	9.6829e-05	9.6655e-05	1.2137e-04	
12	1.0123e-04	9.6923e-05	1.0317e-04	
13	9.1842e-05	9.9568e-05	1.1044e-04	
14	8.5339e-06	1.2398e-05	1.8674e-05	
15	4.2867e-06	1.5059e-05	1.0625e-05	
16	7.3529e-06	7.5668e-06	1.0559e-05	
17	5.8748e-06	1.7595e-05	1.2949e-05	
18	7.9080e-06	2.2059e-05	2.0863e-05	
19	5.4613e-06	1.1223e-05	9.1663e-06	
20	5.2951e-06	7.6011e-06	5.9468e-06	
21	2.9183e-06	1.0380e-05	7.1979e-06	
22	0.0173	7.7415e-06	9.6856e-06	
23	0.0029	1.6277e-04	8.2579e-06	

Fonte: Próprio Autor

APÊNDICE F – MATRIZ DE ERRO (MAPE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA FOCADA

Figura 58 – Parte I da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: DATASET_test_nn_p 01/02/2023					Page 1 7:16:24 PM
	1	2	3	4	5
1	0.1729	0.5370	1.4172	1.1252	1.3929
2	35.8739	0.6762	1.8752	1.4180	1.7748
3	47.2429	44.2375	4.1976	3.0109	3.9326
4	42.3287	40.9750	56.9985	1.6100	1.7830
5	55.3816	50.9461	49.7563	54.3215	1.5813
6	48.4347	41.4847	70.2291	36.9788	40.0899
7	56.8259	42.4430	46.2249	45.6058	25.7403
8	67.8017	61.3125	61.6681	51.0129	56.5120
9	40.8047	28.3507	50.4796	56.9868	19.8092
10	63.1106	62.3915	48.8562	80.0410	18.5925
11	40.6657	39.4910	49.5872	74.4104	61.0651
12	43.0238	45.1220	52.6289	61.0943	90.2457
13	48.5623	57.0533	66.4238	51.3462	158.7489
14	159.3128	39.0225	241.3142	128.2633	82.6431
15	594.0593	63.3947	44.7036	91.9898	105.3384
16	311.6546	63.4934	56.5516	73.9786	55.5316
17	71.1301	51.3020	50.3616	78.9897	67.7961
18	66.5236	47.3725	51.1726	73.8784	57.2928
19	79.6012	42.4355	46.2102	72.3973	57.2714
20	72.6723	56.3797	50.8288	71.4320	61.8214
21	63.6831	60.3290	53.8793	64.5399	57.5531
22	58.3854	74.2596	63.9971	52.7764	93.9445
23	55.5300	62.7570	64.7633	50.4902	89.2454

Figura 59 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: DATASET_test_nn_p 01/02/2023						Page 2 7:16:24 PM
	6	7	8	9	10	
1	1.1427	1.4651	1.5334	1.4206	1.5360	
2	1.7503	2.1365	2.5778	2.0042	2.3727	
3	3.2434	4.9411	6.4116	4.0766	4.9452	
4	2.0683	2.3265	2.5350	2.2209	2.4797	
5	1.6381	1.9070	1.9712	1.8455	1.8367	
6	1.1751	1.3482	1.4942	1.3205	1.3791	
7	75.3425	1.1977	1.4041	1.3051	1.1783	
8	35.2383	26.8967	1.6411	1.5620	1.5003	
9	7.9035	25.0140	4.7606	1.6423	1.7400	
10	5.7365	7.0150	3.2571	6.0639	1.7909	
11	81.8565	19.2952	12.0275	46.8887	39.6309	
12	88.9962	33.1456	36.7094	76.2156	31.0771	
13	91.2955	59.9939	43.8258	75.9778	47.6551	
14	310.8487	254.8357	87.8812	146.5718	51.9075	
15	97.7066	64.8399	211.4115	67.1128	66.9963	
16	101.0963	61.6649	47.7909	87.6787	51.9110	
17	83.7868	63.1266	54.4023	74.3633	67.1101	
18	84.4317	67.6351	60.9815	75.1715	82.6169	
19	89.9921	66.1795	64.8359	74.0100	107.3518	
20	96.5051	76.8059	53.9994	75.8419	68.3528	
21	80.7763	67.5387	75.5630	73.3222	67.3001	
22	93.8840	52.5449	198.1189	90.7857	41.7539	
23	95.7779	58.8382	132.2588	75.5411	40.5305	

Fonte: Próprio Autor

Figura 60 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: DATASET_test_nn_p 01/02/2023					Page 3 7:16:24 PM
	11	12	13	14	15
1	1.5774	1.5663	1.6210	1.5837	1.6418
2	2.3521	1.9390	2.2814	2.5334	2.8058
3	4.7412	3.8881	4.0793	4.5246	6.8150
4	2.6491	2.3482	2.2581	2.5610	2.7239
5	1.9944	1.9518	1.9269	1.9777	2.1748
6	1.4252	1.3164	1.4541	1.5662	1.5378
7	1.3576	1.2590	1.2205	1.2372	1.3485
8	1.6069	1.5529	1.5210	1.6772	1.7591
9	1.8351	1.6743	1.6938	1.7759	1.8984
10	1.8116	1.6117	1.7359	1.8066	1.8664
11	1.5544	1.5668	1.5949	1.5000	1.6375
12	68.4377	1.6208	1.7958	1.6945	1.8277
13	40.0467	51.8729	2.3078	2.1739	2.5514
14	160.3240	38.6339	77.2518	0.6934	0.5828
15	244.9549	73.7756	74.1849	71.4133	1.2377
16	164.9607	63.9227	75.3356	58.6107	29.6610
17	142.1051	67.4259	71.3012	63.5313	30.1723
18	74.8154	58.9166	73.4351	71.4088	34.7763
19	151.3164	65.6495	55.4994	66.8425	13.1688
20	82.3568	67.8744	73.9553	81.7839	58.5780
21	70.0129	57.3292	100.3566	60.2173	20.8725
22	161.4902	77.0941	75.7657	84.5629	27.9758
23	157.2942	67.8231	74.6434	56.3285	28.8077

Fonte: Próprio Autor

Figura 61 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: DATASET_test_nn_p 01/02/2023					Page 4 7:16:24 PM
	16	17	18	19	20
1	1.6146	1.6569	1.6877	1.6667	1.6065
2	2.3923	2.6548	2.4665	2.4418	2.6364
3	5.4390	6.2841	4.7593	6.0450	7.2844
4	2.5180	2.6119	2.5021	2.6969	2.6883
5	2.0177	2.0509	1.9456	2.0438	2.0849
6	1.4085	1.5434	1.4591	1.5659	1.5225
7	1.1998	1.5033	1.3986	1.4113	1.4363
8	1.6618	1.7565	1.6895	1.7240	1.7705
9	1.8310	1.8626	1.7913	1.7512	1.8654
10	1.7641	1.8227	1.7918	1.7388	1.8613
11	1.4995	1.7110	1.5208	1.7945	1.6877
12	1.7409	1.8050	1.8028	1.8088	1.8327
13	2.1133	2.4055	2.2885	2.3564	2.4960
14	0.7118	0.6728	0.5753	0.9357	0.5706
15	1.1532	1.2601	1.6046	1.2132	0.7927
16	0.4862	0.4647	0.4549	0.6852	0.6303
17	33.1446	1.8305	2.1783	1.8130	1.6754
18	56.8966	21.4141	1.3435	1.4917	1.6149
19	10.3472	23.9885	4.9726	2.1102	1.5192
20	37.0565	25.8920	39.2086	39.3834	1.3096
21	53.4526	21.9566	2.8098	2.5195	3.4827
22	100.7824	34.4855	17.0788	28.1045	36.6683
23	80.4120	22.3647	12.3583	29.0157	37.8680

Fonte: Próprio Autor

Figura 62 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA Focada

MATLAB Variable: DATASET_test_nn_p 01/02/2023		Page 5 7:16:24 PM	
	21	22	23
1	1.6282	1.6923	1.7482
2	2.6317	2.6930	3.0882
3	6.2541	8.4769	8.3171
4	2.5255	2.6566	2.9069
5	2.0169	2.1415	2.1618
6	1.4803	1.5355	1.7434
7	1.3196	1.3838	1.5672
8	1.6691	1.7301	1.9457
9	1.8012	1.9847	2.0724
10	1.8261	1.8816	1.9163
11	1.6642	1.6926	1.8572
12	1.8070	1.7997	1.8899
13	2.3231	2.4448	2.6275
14	0.6223	0.6450	0.9692
15	0.8322	1.2461	1.4268
16	0.7172	0.6322	0.7951
17	1.6296	2.4497	2.7461
18	1.2599	2.1368	1.8172
19	1.4640	2.4228	1.9826
20	1.8829	2.7592	1.6378
21	0.6205	1.0252	0.9894
22	25.6101	0.6845	0.8728
23	14.9493	3.4343	0.7406

Fonte: Próprio Autor

APÊNDICE G – MATRIZ DE ERRO (MSE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA DISTRIBUÍDA

Figura 63 – Parte I da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: Matriz_nn 30/01/2023					Page 1 6:38:03 AM
	1	2	3	4	5
1	3.3568e-06	3.8838e-05	1.0396e-04	9.6952e-05	1.5572e-04
2	0.0075	4.6599e-05	9.8140e-05	9.9650e-05	9.9998e-05
3	0.0103	0.0893	5.1753e-04	5.7653e-04	4.9500e-04
4	0.0112	0.1361	2.0171	1.3691e-04	1.0032e-04
5	0.0589	0.1099	1.6220	0.0268	1.2686e-04
6	0.1305	0.0909	0.8830	0.2056	0.0902
7	0.0572	0.1037	1.2153	0.2128	0.0447
8	0.0510	0.1977	0.6342	0.6262	0.0897
9	0.0389	0.2828	1.8896	0.0178	0.0038
10	0.0523	0.6984	2.2768	0.0409	0.0099
11	0.0961	0.2659	2.4151	0.0175	0.0088
12	0.1246	0.1049	2.4060	0.0816	0.7231
13	0.1545	0.4575	3.1080	0.0285	1.1694
14	0.2139	0.1097	1.3140	0.0553	0.1561
15	0.1275	0.3378	3.7362	0.8409	0.2566
16	0.1000	0.1957	3.3294	0.8643	0.3887
17	0.1406	0.3107	3.6102	0.9260	0.2776
18	0.1641	0.2768	3.4495	0.3976	0.3190
19	0.1495	0.3088	3.5841	0.5169	0.3086
20	0.1373	0.2936	2.5876	0.2586	0.2675
21	0.1710	0.2604	3.4617	0.2779	0.3480
22	0.0986	0.1033	3.4563	0.2293	0.3538
23	0.1285	0.1600	3.4728	0.1861	0.3886

Figura 64 – Parte II da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: Matriz_nn 30/01/2023						Page 2 6:38:03 AM
	6	7	8	9	10	
1	1.4204e-04	9.8120e-05	1.1962e-04	6.5388e-04	3.0967e-04	
2	1.3487e-04	9.9295e-05	9.8092e-05	0.0032	8.3338e-04	
3	5.8220e-04	3.6746e-04	3.2915e-04	0.0028	4.6988e-04	
4	1.6557e-04	9.2682e-05	1.0616e-04	0.0094	3.4739e-04	
5	1.8719e-04	1.2013e-04	1.2862e-04	0.0095	3.9968e-04	
6	1.4687e-04	8.0914e-05	8.7343e-05	0.0087	4.4492e-04	
7	0.1456	1.1077e-04	1.1444e-04	0.0104	4.1616e-04	
8	0.0103	0.0168	1.0856e-04	0.0095	0.0012	
9	7.9680e-04	6.6436e-04	9.6797e-04	0.0085	3.6396e-04	
10	5.4415e-04	5.7527e-04	3.7270e-04	0.0141	0.0024	
11	0.0047	0.0149	0.0080	1.6592	0.1231	
12	0.4040	0.5585	0.1457	1.6172	0.3833	
13	0.7673	0.5079	0.4612	2.1944	0.0250	
14	0.0626	0.0592	0.0515	0.1006	0.9634	
15	0.7653	5.5531	0.3926	2.0538	0.3884	
16	0.1193	6.2691	0.3648	1.5408	0.1354	
17	0.2900	5.4188	0.4514	0.7681	0.3705	
18	0.2903	6.6252	0.6254	1.1886	0.1582	
19	0.2563	6.3018	0.5068	0.8814	0.1742	
20	0.4276	6.1377	0.8737	1.2588	0.4415	
21	0.2187	6.8096	1.5879	0.7102	0.1564	
22	0.1880	1.6824	0.5069	1.8682	2.6895	
23	0.6188	3.9672	0.5906	1.8860	1.7670	

Fonte: Próprio Autor

Figura 65 – Parte III da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: Matriz_nn 30/01/2023					Page 3 6:38:03 AM
	11	12	13	14	15
1	1.1357e-04	1.2097e-04	1.1698e-04	1.1583e-04	1.3818e-04
2	1.2291e-04	1.0515e-04	1.0530e-04	1.1588e-04	1.8363e-04
3	6.2995e-04	6.0319e-04	5.8597e-04	6.4520e-04	6.7258e-04
4	1.2184e-04	1.1781e-04	1.1382e-04	1.2399e-04	2.2163e-04
5	1.5356e-04	1.4144e-04	1.3969e-04	1.4984e-04	2.3042e-04
6	1.0341e-04	1.0902e-04	9.6421e-05	1.0379e-04	1.9809e-04
7	1.3350e-04	1.9401e-04	1.4537e-04	1.2632e-04	2.0780e-04
8	1.2157e-04	1.3443e-04	1.2797e-04	1.2647e-04	1.5469e-04
9	1.1582e-04	1.0947e-04	1.0539e-04	1.0868e-04	1.8720e-04
10	1.1839e-04	1.1516e-04	1.1237e-04	1.1343e-04	1.9822e-04
11	1.0569e-04	1.0650e-04	9.3824e-05	1.0435e-04	1.9358e-04
12	1.7147	9.3857e-05	9.3285e-05	1.0824e-04	1.1895e-04
13	0.7969	0.0246	1.1188e-04	1.0134e-04	1.9227e-04
14	0.0720	0.0395	0.2248	6.1199e-05	1.1825e-04
15	0.2040	0.2223	0.1236	0.0935	6.5076e-05
16	0.1464	0.1069	0.1774	0.1147	0.1088
17	0.1195	0.2872	0.0697	0.1986	0.0033
18	0.2204	1.0077	0.0899	0.2057	0.0024
19	0.1689	0.4134	0.1021	0.1969	9.5380e-04
20	0.1286	0.2272	0.1320	0.1269	0.0137
21	0.2127	1.7599	0.1086	0.1699	0.0024
22	0.2458	0.4381	0.2801	0.1009	0.0414
23	0.2611	1.0381	0.2308	0.1149	0.0187

Fonte: Próprio Autor

Figura 66 – Parte IV da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: Matriz_nn 30/01/2023					Page 4 6:38:03 AM
	16	17	18	19	20
1	1.1241e-04	1.4844e-04	1.2330e-04	2.3106e-04	1.8516e-04
2	1.4663e-04	1.6962e-04	1.6317e-04	2.6636e-04	1.6183e-04
3	0.0015	0.0015	6.9154e-04	0.0015	6.8442e-04
4	1.4232e-04	1.3710e-04	1.3928e-04	2.8497e-04	2.6206e-04
5	1.7592e-04	1.6913e-04	1.6844e-04	3.0536e-04	2.9046e-04
6	1.2897e-04	1.5180e-04	1.1134e-04	2.7783e-04	1.5292e-04
7	1.6631e-04	1.7619e-04	1.9740e-04	3.0356e-04	2.0639e-04
8	1.4483e-04	1.8813e-04	1.3865e-04	4.1547e-04	2.2366e-04
9	1.3595e-04	1.7183e-04	1.1702e-04	3.1051e-04	1.5546e-04
10	1.3565e-04	1.3400e-04	1.2776e-04	2.5378e-04	1.4774e-04
11	1.2621e-04	1.4565e-04	1.2626e-04	2.7713e-04	1.5266e-04
12	1.0913e-04	1.3528e-04	1.3130e-04	2.8128e-04	1.2982e-04
13	1.3530e-04	1.6378e-04	1.3665e-04	3.2474e-04	1.6428e-04
14	5.1591e-05	4.8426e-05	4.4197e-05	1.8533e-04	1.5641e-04
15	2.4636e-05	1.1760e-04	3.4482e-05	1.4895e-04	1.1893e-04
16	4.0044e-05	1.1494e-04	3.8932e-05	1.7913e-04	1.0121e-04
17	2.3293e-04	1.0272e-04	6.3528e-05	1.5143e-04	1.0202e-04
18	0.0021	9.9193e-04	4.0053e-05	1.9405e-04	1.0726e-04
19	9.7038e-04	1.8871e-04	1.9559e-04	1.6789e-04	1.1654e-04
20	0.0028	5.7605e-04	0.0116	7.9186e-04	9.8275e-05
21	0.0026	7.1723e-04	1.4564e-04	2.8836e-04	2.4130e-04
22	0.0078	0.0046	0.0042	0.0394	0.0049
23	0.0079	0.0030	0.0033	0.0238	0.0041

Fonte: Próprio Autor

Figura 67 – Parte V da matriz de erro (MSE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: Matriz_nn 30/01/2023			Page 5 6:38:03 AM
	21	22	23
1	1.1848e-04	1.6464e-04	1.8738e-04
2	1.6560e-04	1.9614e-04	1.6198e-04
3	0.0017	0.0017	6.3451e-04
4	1.5003e-04	1.6673e-04	2.0138e-04
5	1.7850e-04	2.0473e-04	1.8269e-04
6	1.3408e-04	1.6954e-04	1.6545e-04
7	1.6199e-04	1.9419e-04	1.6904e-04
8	1.3990e-04	1.6469e-04	1.9421e-04
9	1.2813e-04	1.3911e-04	1.5318e-04
10	1.2569e-04	1.6087e-04	1.8149e-04
11	1.3026e-04	1.4958e-04	1.7778e-04
12	1.1421e-04	1.5073e-04	1.4095e-04
13	1.2133e-04	1.5106e-04	1.8185e-04
14	5.7875e-05	5.8383e-05	5.1132e-05
15	5.9395e-05	4.8438e-05	2.0877e-04
16	7.6777e-05	4.9846e-05	2.3851e-04
17	7.5202e-05	6.0545e-05	1.6314e-04
18	8.6779e-05	6.9884e-05	2.5408e-04
19	6.1420e-05	4.2378e-05	2.1042e-04
20	6.7219e-05	4.4495e-05	2.3084e-04
21	6.9440e-05	4.9297e-05	2.3745e-04
22	0.0027	4.4766e-05	2.0993e-04
23	0.0016	7.4535e-04	1.5499e-04

Fonte: Próprio Autor

APÊNDICE H – MATRIZ DE ERRO (MAPE) PARA ENTRADA COM MÚLTIPLAS TRAJETÓRIAS, RNA DISTRIBUÍDA

Figura 68 – Parte I da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: DATASET_test_nn_p 01/02/2023					Page 1 7:14:35 PM
	1	2	3	4	5
1	0.2842	0.8843	1.4605	1.4769	1.6165
2	16.5122	1.2258	1.9758	2.3237	2.1679
3	32.3633	36.5428	4.2557	5.0880	4.1506
4	25.9497	36.4073	63.5654	2.5637	2.2882
5	57.8029	64.7751	81.3450	26.8656	1.9053
6	52.1698	48.7624	81.0359	34.8021	90.1989
7	39.4125	62.5271	86.6612	46.1842	25.5649
8	43.6732	75.1296	71.8371	46.2624	96.6831
9	71.2141	58.5168	90.4835	20.1513	13.4233
10	53.6001	52.5291	86.6812	24.8214	36.8345
11	152.5650	58.2847	91.8416	23.3574	16.4453
12	174.7683	32.3999	70.9395	55.7349	55.4298
13	155.5239	43.1970	80.1239	50.7939	64.1758
14	161.5595	55.6245	227.2422	40.9561	357.2582
15	235.7014	67.6901	118.1518	82.6785	151.1546
16	139.2672	49.3942	114.2402	63.1185	112.6430
17	72.6416	60.9969	97.2260	116.7820	142.8319
18	306.9368	55.5047	91.0498	87.2826	135.0290
19	282.5698	57.8946	128.7386	94.7625	138.7572
20	257.1973	63.6767	76.9538	62.5825	128.0556
21	320.0619	53.2095	92.4271	81.6875	132.7665
22	48.9449	40.7016	70.8230	114.3767	114.3792
23	55.9347	45.4421	73.4049	95.9513	123.3836

Figura 69 – Parte II da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: DATASET_test_nn_p 01/02/2023						Page 2 7:14:35 PM
	6	7	8	9	10	
1	1.4146	1.5000	1.5975	1.8865	1.5857	
2	2.1996	2.2281	2.1861	3.6141	2.5471	
3	4.7928	4.1300	3.9583	5.7942	4.5837	
4	2.4952	2.3215	2.5141	5.8107	2.6330	
5	1.9213	1.9204	1.8938	4.4064	2.0249	
6	1.2522	1.3304	1.3216	3.3947	1.5785	
7	39.7687	1.1604	1.2405	3.0196	1.2464	
8	18.0890	22.3350	1.5519	3.4549	1.7419	
9	7.0352	6.0010	7.7766	4.2225	1.8175	
10	4.7340	4.3454	3.8917	10.8927	2.3644	
11	11.5257	17.5531	16.0480	64.9266	85.3129	
12	83.9875	117.5807	54.9787	69.0579	71.9275	
13	119.8082	119.7412	91.7480	74.2072	38.6377	
14	48.5568	32.1949	57.8413	59.5161	215.9473	
15	55.4871	66.0670	105.9051	83.2679	62.9842	
16	81.5848	70.1221	97.2266	81.1915	45.6665	
17	87.4323	68.6117	101.0708	73.7379	66.9259	
18	95.5762	75.2380	103.0836	80.2891	48.4656	
19	95.7524	71.4903	102.1709	74.8853	61.1493	
20	93.4597	67.6593	99.3552	82.7455	65.1215	
21	87.7228	77.3338	101.9131	75.5472	43.7554	
22	55.7954	73.6556	73.3050	74.9110	66.5609	
23	64.0065	76.3191	83.6297	79.1852	66.1555	

Fonte: Próprio Autor

Figura 70 – Parte III da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: DATASET_test_nn_p 01/02/2023					Page 3 7:14:35 PM
	11	12	13	14	15
1	1.6206	1.6550	1.6516	1.6504	1.6225
2	2.6115	2.3747	2.3591	2.6406	2.7557
3	6.1725	5.5002	5.5603	7.0890	6.7399
4	2.6787	2.6490	2.6184	2.6596	2.7506
5	2.1421	2.0379	2.0280	2.0793	2.0882
6	1.5071	1.5851	1.4576	1.5231	1.4560
7	1.3812	1.3863	1.3705	1.3622	1.4041
8	1.6220	1.6651	1.6253	1.7300	1.6749
9	1.9537	1.7617	1.8334	1.8998	1.9126
10	1.8291	1.8206	1.8151	1.8946	1.8922
11	1.6139	1.5700	1.5189	1.6618	1.7314
12	68.8329	1.7827	1.7470	1.9160	1.7742
13	73.6896	30.1514	2.2792	2.4358	2.5074
14	45.0820	33.3611	47.6339	0.8024	1.1104
15	71.1427	101.0111	61.3859	51.6995	1.8208
16	61.5623	75.3232	78.2477	78.3011	57.2637
17	66.3079	90.9680	47.8778	118.4438	37.8280
18	55.5199	88.7529	51.1510	118.5581	18.0360
19	58.6246	88.5266	55.1728	120.2876	18.4113
20	55.0118	89.3982	54.0879	66.9803	46.6749
21	57.3586	87.7814	50.1944	130.4460	18.0736
22	72.1823	76.1177	56.7700	215.9149	32.1900
23	76.2591	89.9442	55.4142	180.7783	28.8959

Fonte: Próprio Autor

Figura 71 – Parte IV da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: DATASET_test_nn_p 01/02/2023						Page 4 7:14:35 PM
	16	17	18	19	20	
1	1.6164	1.7246	1.6959	1.7070	1.7580	
2	2.9572	2.9365	2.7895	2.9209	2.7927	
3	8.1650	8.0829	7.7355	8.0866	7.0356	
4	2.6944	2.7167	2.5774	2.8637	2.7235	
5	2.1896	2.2058	2.0902	2.1899	2.1573	
6	1.6453	1.6943	1.5772	1.6418	1.6682	
7	1.5608	1.6300	1.5160	1.4434	1.3908	
8	1.8002	1.8628	1.7782	1.7734	1.7837	
9	1.9780	1.9451	1.8984	1.9351	1.9776	
10	1.9377	1.8969	1.8207	1.8664	1.8761	
11	1.7325	1.7707	1.7406	1.7082	1.7480	
12	1.8740	1.9783	1.9481	1.9405	1.8836	
13	2.5327	2.7251	2.6737	2.4885	2.4634	
14	0.8920	0.8106	0.8366	0.9688	1.2719	
15	2.0980	3.1641	2.0363	3.0608	3.0971	
16	0.5276	0.9793	0.8395	1.1965	0.7693	
17	12.8535	5.2680	4.0225	4.0652	5.0821	
18	17.8097	13.1126	1.3537	2.3006	2.2379	
19	19.1260	6.8975	7.1271	3.6586	4.0166	
20	34.4155	20.4550	36.5518	23.8633	11.4583	
21	17.3131	9.3976	4.3172	5.0273	4.9850	
22	24.7426	32.8641	17.1405	61.1473	15.0761	
23	19.6859	18.3470	15.3442	52.0804	16.4558	

Fonte: Próprio Autor

Figura 72 – Parte V da matriz de erro (MAPE) para entrada com múltiplas trajetórias, RNA DISTRIBUÍDA

MATLAB Variable: DATASET_test_nn_p 01/02/2023		Page 5 7:14:35 PM		
	21	22	23	
1	1.6888	1.6935	1.7231	
2	3.1849	3.1320	2.7356	
3	11.3339	9.4572	6.9151	
4	3.0465	2.6872	2.7361	
5	2.2710	2.2047	2.0900	
6	1.7375	1.9076	1.6355	
7	1.6122	1.6668	1.4635	
8	1.9101	1.8975	1.8197	
9	2.0565	2.0906	1.8568	
10	1.8622	1.9944	1.9385	
11	1.7829	1.8316	1.7867	
12	1.9385	2.0024	1.9311	
13	2.6517	2.7039	2.5577	
14	0.9076	0.9476	0.7554	
15	3.2229	2.1512	2.8012	
16	1.2622	0.9369	1.6052	
17	5.6897	4.1291	3.4375	
18	3.0193	3.3465	2.6030	
19	3.4605	3.0271	2.6439	
20	9.6348	8.5448	7.2184	
21	1.5948	1.4800	1.5122	
22	13.7339	0.9893	0.8239	
23	11.5057	6.5203	0.6710	

Fonte: Próprio Autor