

**METODOLOGIA PARA ANÁLISE E MITIGAÇÃO DE  
ERROS EM SISTEMAS DE MEDIÇÃO INERCIAL  
UTILIZANDO REDES NEURAS ARTIFICIAIS E  
MOVIMENTO GUIADO PARA CALIBRAÇÃO DE ROBÔS**

**BENEDITO ALOÍSIO NUNES CAMPOS**

**TESE DE DOUTORADO EM SISTEMAS MECATRÔNICOS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**METHODOLOGY FOR ANALYSIS AND MITIGATION OF  
ERRORS IN INERTIAL MEASUREMENT SYSTEMS USING  
ARTIFICIAL NEURAL NETWORKS AND GUIDED  
MOVEMENT FOR CALIBRATING ROBOTS**

**BENEDITO ALOÍSIO NUNES CAMPOS**

**ORIENTADOR: JOSÉ MAURÍCIO S. T. MOTTA**

**TESE DE DOUTORADO EM SISTEMAS  
MECATRÔNICOS**

**BRASÍLIA/DF: MAIO - 2021**

UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

METODOLOGIA PARA ANÁLISE E MITIGAÇÃO DE  
ERROS EM SISTEMAS DE MEDIÇÃO INERCIAL  
UTILIZANDO REDES NEURAS ARTIFICIAIS E  
MOVIMENTO GUIADO PARA CALIBRAÇÃO DE ROBÔS

BENEDITO ALOÍSIO NUNES CAMPOS

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA  
MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA  
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
DOUTOR EM SISTEMAS MECATRÔNICOS.

APROVADA POR:

---

Prof. Dr. José Maurício S. T. Motta –(ENM/UnB)  
Orientador

---

Prof. Dr. Flávio de Barros Vidal – (CIC/UnB))  
Membro interno

---

Prof. Dr. Guilherme Caribé de Carvalho – (ENM/UnB))  
Membro interno

---

Prof. Dr. Carlos Humberto Llanos Quintero – (ENM/UnB)  
Membro interno

---

Prof. Dr. Tales Cleber Pimenta – (IESTI/UNIFEI)  
Membro externo

BRASÍLIA, 27 DE MAIO DE 2021.

## FICHA CATALOGRÁFICA

CAMPOS, B.

METODOLOGIA PARA ANÁLISE E MITIGAÇÃO DE ERROS EM SISTEMAS DE MEDIÇÃO INERCIAL UTILIZANDO REDES NEURAIAS ARTIFICIAIS E MOVIMENTO GUIADO PARA CALIBRAÇÃO DE ROBÔS [Distrito Federal] 2021.

xviii, 108p., 210 x 297 mm (ENM/FT/UnB, Doutor, Sistemas Mecatrônicos , 2021).

Tese de doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Mecânica

1. Calibração

2. Robôs

3. Redes Neurais

4. IMU

I. ENM/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

(2021). METODOLOGIA PARA ANÁLISE E MITIGAÇÃO DE ERROS EM SISTEMAS DE MEDIÇÃO INERCIAL UTILIZANDO REDES NEURAIAS ARTIFICIAIS E MOVIMENTO GUIADO PARA CALIBRAÇÃO DE ROBÔS . Tese de doutorado em Sistemas Mecatrônicos , Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 108p.

## CESSÃO DE DIREITOS

AUTOR: Benedito Aloísio Nunes Campos

TÍTULO: METODOLOGIA PARA ANÁLISE E MITIGAÇÃO DE ERROS EM SISTEMAS DE MEDIÇÃO INERCIAL UTILIZANDO REDES NEURAIAS ARTIFICIAIS E MOVIMENTO GUIADO PARA CALIBRAÇÃO DE ROBÔS .

GRAU: Doutor

ANO: 2021

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa tese de doutorado pode ser reproduzida sem autorização por escrito do autor.

---

Benedito Aloísio Nunes Campos

Departamento de Engenharia Mecânica (ENM) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*Dedico esse trabalho ao poderoso  
e desconhecido Engenheiro Aloísio  
Campos \*1941 - †2015 .*

## AGRADECIMENTOS

*Agradeço primeiramente a Deus. Agradeço também ao meu orientador, que mais que orientador sempre foi um amigo. Agradeço a minha família e a certas pessoas especiais que tem estado próximas de minha vida me inspirando e incentivando.*

## RESUMO

Esta pesquisa tem como objetivo desenvolver uma metodologia para prever a posição de um robô durante seu movimento. A abordagem introduzida pode ser usada para criar um modelo de calibração de robô mais robusto do que os métodos disponíveis atualmente. A ideia é utilizar Sistemas de Medição Inercial (IMUs) e Redes Neurais Artificiais (RNAs) para realizar um tipo de navegação estimada (*Dead Reckoning*) do Elemento Terminal de um Robô ou *Tool Center Point* (TCP). Os sistemas IMUs são influenciados por ruídos em suas medições, e esses ruídos aumentam com o tempo, dessa forma, os valores de erro devem ser conhecidos e mitigados. Este estudo utilizou RNAs para reconhecer alvos característicos em um espaço vetorial durante o movimento guiado previamente programado do robô. O objetivo é mitigar o ruído do sistema de medição para que os valores se aproximem da precisão necessária para calibrar o robô. A contribuição da investigação será fornecer um método de navegação estimado capaz de alimentar o software do controlador e, assim, corrigir os desvios encontrados e calibrar a posição do TCP. A RNA foi capaz de prever a posição do elemento terminal com um erro máximo de 0.9 mm. Os resultados mostram que o método desenvolvido é satisfatório e pode ser adotado para retroalimentar o software que controla o robô sempre que ele perceber uma variação entre os valores nominais e os previstos pelo RNA. Além disso, o trabalho aplicou o conceito a outras redes neurais que representam o estado da arte, usando um conjunto de dados menor. A metodologia apresentada neste trabalho também pode ser extrapolada para outros problemas de posicionamento em tempo real.

**Palavras-chave:** Calibração, Fusão de Sensores, IMU, Online, Redes Neurais, Robôs.

## ABSTRACT

This research aims to develop a methodology to predict the position of a robot during its movement. The approach introduced can be used to create a more robust robot calibration model than currently available methods. The idea is to use Inertial Measurement Systems (IMUs) and Artificial Neural Networks (ANNs) to perform a type of estimated navigation (*Dead Reckoning*) of the Terminal Element of a Robot or *Tool Center Point* (TCP). IMU systems are influenced by noise in their measurements, and these noises increase over time, so error values must be known and mitigated. This study used ANNs to recognize characteristic targets in a vector space during the robot's previously programmed guided movement. The objective is to mitigate the noise of the measurement system so that the values approach the precision needed to calibrate the robot. The contribution of the investigation will be to provide an estimated navigation method capable of feeding the controller software and, thus, correcting the deviations found and calibrating the TCP position. ANN was able to predict the position of the terminal element with a maximum error of 0.9 mm. The results show that the developed method is satisfactory and can be adopted to provide feedback to the software that controls the robot whenever it perceives a variation between the nominal values and those predicted by the ANN. Furthermore, the work applied the concept to other neural networks that represent the state-of-the-art, using a smaller dataset. The methodology presented in this work can also be extrapolated to other real-time positioning problems.

**Keywords:** Calibration, Sensor Fusion, IMU, Online, Neural Networks, Robots..



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO	1
1.2	PROPOSTA DE TESE E SUA JUSTIFICATIVA	3
1.3	QUESTÃO A RESPONDER	4
1.4	HIPÓTESE A COMPROVAR	4
1.5	OBJETIVO GERAL E SUA CONTRIBUIÇÃO	4
1.6	OBJETIVOS ESPECÍFICOS	5
1.7	ESTRUTURA DO DOCUMENTO	5
<b>2</b>	<b>REFERENCIAL TEÓRICO E REVISÃO BIBLIOGRÁFICA</b>	<b>6</b>
2.1	TRABALHOS RELACIONADOS	6
2.2	CALIBRAÇÃO DE ROBÔS	10
2.2.1	MODELO DENAVIT-HARTENBERG	13
2.2.2	MODELO S	13
2.2.3	MODELO CPC	13
2.2.4	MODELO REFERÊNCIA ZERO	14
2.2.5	MODELO POE	14
2.2.6	CLASSIFICAÇÃO DO MODELO	14
2.2.7	FONTES DE ERROS	16
2.2.8	MODELO PARAMÉTRICO	16
2.2.9	SISTEMAS DE MEDIÇÃO	19
2.2.10	CALIBRAÇÃO ONLINE E OFFLINE	20
2.2.11	UNIDADES DE MEDIÇÃO INERCIAL (IMUs)	25
2.2.12	IMUs: SENSORES	25
2.2.13	IMUs: GIROSCÓPIO	26
2.2.14	IMUs: ACELERÔMETROS	36
2.2.15	IMUs: MEMs	41
2.2.16	IMUs: ERROS	42
2.2.17	IMUs: CONCLUSÃO E ESCOLHA DE UM MODELO	46
2.3	REDES NEURAIS ARTIFICIAIS	48

2.3.1	O SISTEMA NERVOSO ANIMAL .....	48
2.3.2	MODELO DE UMA REDE NEURAL ARTIFICIAL .....	51
2.3.3	RNAS: TIPOS .....	52
2.3.4	RNAS: FORMULAÇÃO MATEMÁTICA NO CASO DA IMU .....	61
<b>3</b>	<b>METODOLOGIA .....</b>	<b>68</b>
3.1	PREPARAÇÃO DOS DADOS .....	72
3.1.1	PROCESSO DE TREINAMENTO .....	74
3.1.2	VALIDAÇÃO CRUZADA .....	75
3.2	COMPARAÇÃO COM OUTRAS TOPOLOGIAS .....	76
3.3	MATERIAIS .....	78
<b>4</b>	<b>RESULTADOS .....</b>	<b>79</b>
4.1	REDE NEURAL ARTIFICIAL PROPOSTA .....	80
4.2	ALGORITMO PERCEPTRON MULTICAMADAS .....	82
4.3	REDE NEURAL CONVOLUCIONAL .....	84
4.4	ALGORITMO XGBOOST REGRESSOR .....	85
<b>5</b>	<b>DISCUSSÃO .....</b>	<b>88</b>
5.1	DISCUSSÃO GERAL .....	88
5.2	COMPARAÇÃO COM OUTRAS REDES .....	90
5.3	TRABALHOS FUTUROS .....	92
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>94</b>

# Lista de Figuras

1.1	Projeção de crescimento por setor. Imagem desenvolvida pelo autor com base em [Consultants 2014].	2
2.1	Pesquisa feita pelo Autor na base Google Acadêmico e Capes Indexado com os termos <i>robot calibration</i> .	7
2.2	Robô IRB 360 presente no laboratório. Fonte: Compilação do Autor e [POWER-MIG 2020]	12
2.3	Modelo paramétrico de dois graus de liberdade, considerando o eixo x e y do plano [Zollo et al. 2015].	16
2.4	Matriz transformação para um grau de liberdade e seu Jacobiano. Desenvolvido com base em [CAMPOS 2006]	17
2.5	Extrapolação para mais graus de liberdade, considerando a possibilidade de rotação e translação (A). Singularidade na notação DH para eixos paralelos (B). Fonte: modificado de [Djuric, Saidi e ElMaraghy 2012]	18
2.6	Configuração interna de um giroscópio. Desenvolvida com base em figura obtida de <i>Creative Commons Attribution-Share Alike 3.0</i> .	27
2.7	Exemplo de giroscópio do tipo HRG. Fonte: Compilação através do arquivo pessoal do autor (software 3D).	29
2.8	Desenho esquemático em corte do projeto de um sistema Ring Laser em um triângulo equilátero. (1) O caminho do laser triangular é fechado usando espelhos altamente refletivos. (2) a descarga de gás é sustentada indutivamente com uma antena de rádio no tubo de ganho estreito. (3) cavidade ressonante evacuada preenchida com uma mistura de Hélio-Néon. (C) combinação dos dois feixes usando dois espelhos de deflexão [Brotzer et al. 2021]	30
2.9	Esquema construtivo do <i>Fiber Optics Gyroscope</i> (FOG). Fonte: O Autor.	31
2.10	Exemplo de um <i>design</i> de um giroscópio do tipo MEMs capacitivo, suas respectivas partes fixas e móveis. Fonte: [Shah et al. 2016].	31
2.11	Esquema de funcionamento de um giroscópio tipo MEMs. Fonte: O Autor.	32
2.12	Esquema de detecção da capacitância diferencial no giroscópio MEMs. Fonte: O Autor.	33
2.13	Esquemático de um acelerômetro em repouso. Fonte: O Autor.	36

2.14	Esquema de um acelerômetro em movimento. Fonte: O Autor.....	37
2.15	Esquema do acelerômetro desalinhado. Fonte: O Autor.....	38
2.16	Esquema de acelerômetro em posição inclinada. Fonte: O Autor.....	38
2.17	Esquema do processo de fabricação em lote. Fonte: O Autor.....	41
2.18	Etapa inicial de orientação da IMU. Fonte: O Autor.....	44
2.19	Pontos possíveis para reinicialização de uma IMU. A aceleração máxima muda a cada passo. Fonte: [Zhang et al. 2017] .....	45
2.20	IA e algumas de suas respectivas áreas. Fonte: O Autor.....	48
2.21	Exemplo de um Neurônio animal. Fonte: [Oliveira 2020] .....	49
2.22	Analogia entre um neurônio artificial e o neurônio animal. Imagem feita a partir de [Silva e Schimidt 2016, Nimbus 2017] .....	50
2.23	Esquema de uma topologia de rede neural artificial. Imagem feita a partir de [Abiodun et al. 2018] .....	51
2.24	O modelo tenta diminuir o erro entre os valores esperados e obtidos, a evolução nem sempre é homogênea e varia entre as diferentes fases do processo. Figura criada a partir de [Li et al. 2018].....	52
2.25	Grafo de uma Rede Neural Artificial supervisionada. Ilustração desenvolvida com base em [Haykin 2009]. .....	54
2.26	Exemplos de funções de ativação e seus respectivos centróides, usadas para realizar o processo de aprendizado da Rede Neural Artificial. Fonte: Creative Commons. ....	55
2.27	Passos de funcionamento dos agrupadores lógicos. Fonte: Creative Commons. .	56
2.28	Exemplos das camadas empregadas em uma arquitetura típica de uma CNN. Fonte: O Autor.....	57
2.29	Operação de convolução de um determinado Kernel. Fonte [Shen et al. 2019]...	58
2.30	Extração de características de uma CNN. A partir da imagem fornecida, a rede mapeia os <i>features</i> automaticamente, possibilitando assim, o reconhecimento de padrões. Fonte: [Zhou et al. 2016]. .....	60
2.31	Topologia de uma rede neural MLP. Fonte: [Haykin 2009]. .....	63
2.32	Grafo de uma rede neural MLP e convergência do treinamento. Imagem feita a partir de [Haykin 2009]. .....	65
2.33	Esquema de experimento de um Trabalho Fatorial. Diversos parâmetros são testados em um esquema em <i>grid</i> que varre diversas possibilidades. Fonte: O Autor.....	67
3.1	RNA recorrente com <i>backpropagation</i> . Fonte: [Chen 2017] .....	69
3.2	IMU acoplada ao robô (A) e produzindo medições em movimento guiado (B)...	71
3.3	Distribuição dos dados de acordo com a normalização. Fonte: [Santos 2018].....	73
3.4	Exemplo de Normalização.....	74
3.5	Exemplo de sobre-treinamento (A) e sub-treinamento (B) .....	75

3.6	Esquema de divisão de dados, validação cruzada e teste. [Campos e Motta 2021]	76
3.7	Validação cruzada adota pela metodologia no processo de treinamento [Campos e Motta 2021].....	76
3.8	Esquema geral da abordagem introduzida por este trabalho. ....	77
3.9	Estação de trabalho e placas GPU com mais de 23000 núcleos CUDA. ....	78
4.1	Trajetórias geradas, programadas no robô. ....	80
4.2	Valores da IMU (vermelho) X valores do Controlador (azul).....	80
4.3	Comparação entre valores do controlador do robô (A) e IMU (B).....	81
4.4	Comparação entre valores do controlador do robô (a) e IMU (b). ....	81
4.5	Distribuição dos valores de erros medidos. ....	82
4.6	Resultado obtido por uma das redes MLP que foi testada.....	83
4.7	Resultados obtidos pelo algoritmo CNN. ....	85
4.8	Resultados obtidos pelo algoritmo XGBoost.....	87

# Lista de Tabelas

2.1	Classificação do Métodos .....	15
2.2	Subdivisões das abordagens paramétricas para calibração de robôs.....	15
2.3	Classificação dos giroscópios quanto ao <i>Bias Instability</i> [KVH 2016].....	35
2.4	Exemplos de Marcas, características e preços de IMUs comerciais. ....	46
2.5	Configurações de fusão do BNO055 - baseada no manual.....	47
2.6	Configurações de interrupção BNO055. - baseado no manual. ....	47
2.7	Exemplos de algumas possibilidades de funcionamento de uma RNA. ....	53
3.1	Exemplo da Organização dos Dados.....	71
4.1	Esquemático das Camadas e Parâmetros.....	84
4.2	Resultado geral.....	87

# Lista de Símbolos

## Símbolos Gregos

$\alpha$	Angulo de revolução no eixo x	[°]
$\alpha_x$	Aceleração angular em $x$	[rad/s <sup>2</sup> ]
$\alpha_y$	Aceleração angular em $y$	[rad/s <sup>2</sup> ]
$\alpha_z$	Aceleração angular em $z$	[rad/s <sup>2</sup> ]
$\beta$	Angulo de revolução no eixo y	[°]
$\delta_j(n)$	Gradiente local ou fator de sensibilidade	
$\eta$	Taxa de aprendizagem	
$\mu$	Coefficiente do método Levenberg-Marquardt	
$\theta$	Variação dos deslocamento angular	

## Símbolos Latino

$M_{(y)}$	Campo magnético da terra em y	[T]
$DCM$	Matriz de cossenos diretores	
$E(n)$	Energia média	
$E_{(W^*)}$	Função de energia aplicada ao vetor ótimo de pesos	
$E_{(W_v)}$	Função de energia aplicada ao vetor de pesos	
$E_{instantânea}$	Energia instantânea do erro do neurônio $j$ e na iteração $n$	
$E_{med}$	Energia total	
$F^b$	Força especifica no referencial do corpo $b$	[N]
$F_R$	Força Resultante	[N]
$F_c$	Força de Coriolis	[N]
$H$	Matriz Hessiana	
$J$	Matriz Jacobiana de uma função vetorial	
$K$	Constante da equação	
$M_{(x)}$	Campo magnético da terra em x	[T]
$M_{(z)}$	Campo magnético da terra em z	[T]
$R^{cn}$	Matriz de rotação entre os referenciais $c$ e $n$	
$R^{ic}$	Matriz de rotação entre os referenciais $i$ e $c$	
$R^{nb}$	Matriz de rotação entre os referenciais $n$ e $b$	
$R^{uv}$	Matriz de rotação entre os referenciais $u$ e $v$	

$T$	Temperatura	$[^{\circ}C]$
$W^*$	Vetor de pesos ótimo	
$W_v$	Vetor de pesos	
$\ddot{x}$	Aceleração no eixo $x$	$[m/s^2]$
$\omega_{ib}^b$	Velocidade angular da estrutura do corpo da IMU em relação ao referencial inercial	$[rad/s]$
$\omega_{en}^n$	Velocidade angular do referencial da terra	$[rad/s]$
$\omega_{ie}^n$	Velocidade angular do referencial inercial $i$ que gira com a terra	$[rad/s]$
$\overrightarrow{G(z)}$	Gravidade em $z$	$[m/s^2]$
$\overrightarrow{M}$	Vetor compo magnético da terra	$[T]$
$\overrightarrow{\omega_X}$	Vetor da velocidade angular no eixo $x$	$[rad/s]$
$\overrightarrow{\omega_Y}$	Vetor da velocidade angular no eixo $y$	$[rad/s]$
$\overrightarrow{\omega}$	Vetor da velocidade angular	$[rad/s]$
$a$	Aceleração	$[m/s^2]$
$a(t)$	Aceleração de um corpo no tempo	$[m/s^2]$
$a_i^i$	Aceleração em $i$	
$a_{ii}^n$	Aceleração linear do sensor no referencial de navegação	$[m/s^2]$
$a_{ij}^n$	Aceleração do referencial $i$ em relação $j$ expressa no referencial $n$	$[m/s^2]$
$a_{nn}^n$	Aceleração linear do sensor no referencial $n$	$[m/s^2]$
$a_x$	Aceleração linear em $x$	$[m/s^2]$
$a_y$	Aceleração linear em $y$	$[m/s^2]$
$a_z$	Aceleração linear em $z$	$[m/s^2]$
$b$	<i>Bias</i> associado ao peso	
$d_j(n)$	Valor desejado na iteração $n$ do neurônio $j$	
$d_{(k)}$	Valor calculado na saída da rede	
$e_j(n)$	Sinal de erro $e$ no neurônio $j$ durante a iteração $n$	
$e_{(k)}$	Erro entre o valor calculado e desejado	
$f(x_n)$	Função objetivo para minimização	
$g$	Aceleração da gravidade	$[m/s^2]$
$g(u_k)$	Função que ajusta a saída de acordo com os pesos e a entrada da rede	
$g_x$	Gravidade em $x$	$[m/s^2]$
$g_y$	Gravidade em $y$	$[m/s^2]$
$g_z$	Gravidade em $z$	$[m/s^2]$
$k$	Número de dobras na validação cruzada	
$m$	Massa	$[kg]$
$n_{nc}^c$	Distância entre o referencial da Terra e o referencial de navegação $c$	
$o_x$	Orientação em $x$	
$o_y$	Orientação em $y$	
$o_z$	Orientação em $z$	



$p^c$	Referencial de navegação em $c$	
$p^i$	Referencial de navegação em $i$	
$p^n$	Posição do sensor no referencial de navegação $n$	
$s(t)$	Posição de um corpo no tempo	[m]
$s_j(n)$	Soma ponderada em cada neurônio	
$t$	Tempo	[s]
$v$	Velocidade	[m/s]
$v(t)$	Velocidade de um corpo no tempo	[m/s]
$v_i^i$	Velocidade em $i$	[m/s]
$v_n^n$	Velocidade do sensor no referencial de navegação $n$	[m/]
$w$	Velocidade Angular	[rad/s]
$w(n)$	Valor do peso na iteração $n$	
$w_{ic}^i$	Velocidade angular do referencial $i$ em relação ao referencial $c$	[rad/s]
$w_{ic}^n$	Velocidade angular do referencial $i$ em relação ao referencial $c$ , expresso no referencial $n$ .	[rad/s]
$w_{uv}^u$	Velocidade angular do referencial $v$ em relação ao referencial $u$	[rad/s]
$w_j$	Peso do neurônio $j$	
$wa_x$	Velocidade angular em $x$	[rad/s]
$wa_y$	Velocidade angular em $y$	[rad/s]
$wa_z$	Velocidade angular em $z$	[rad/s]
$x^*$	Aproximação da solução pelo método Levenberg-Marquardt	
$x^b$	Referencial em relação ao referencial local de navegação	
$x^n$	Referencial geográfico local de navegação	
$x^u$	Vetor em um referencial de coordenadas $u$	
$x_j$	Entrada do neurônio $j$	
$y_j(n)$	Saída obtida na iteração $n$ do neurônio $j$	
$y(k)$	Saída da rede neural	
$y_k$	Saída $k$ de um neurônio	
$x_k, x_{k+1}$	Sequência de aproximações da solução pelo método Levenberg-Marquardt	[m]

# Lista de Acrônimos

<b>ARW</b>	Caminho Aleatório Angular. <a href="#">1</a>
<b>CNN</b>	Rede Neural Convolutacional. <a href="#">1</a>
<b>CVG</b>	Giroscópios Vibratórios Coriolis. <a href="#">1</a>
<b>DCM</b>	Matriz de Cossenos Diretores. <a href="#">1</a>
<b>DH</b>	Método Denavit-Hartenberg. <a href="#">1</a>
<b>DL</b>	Aprendizado Profundo ou Deep Learning. <a href="#">1</a>
<b>FOG</b>	Giroscópio de Fibra Óptica. <a href="#">1</a>
<b>GNSS</b>	Sistemas de Posicionamento por Satélites com Cobertura Mundial. <a href="#">1</a>
<b>GPS</b>	Sistema de Posicionamento Global. <a href="#">1</a>
<b>GPU</b>	Unidade de Processamento Gráfico. <a href="#">1</a>
<b>HRG</b>	Giroscópio de Ressonador Hemisférico. <a href="#">1</a>
<b>IA</b>	Inteligência Artificial. <a href="#">1</a>
<b>IC</b>	Circuitos Integrados Semicondutores. <a href="#">1</a>
<b>IMU</b>	Unidades de Medida Inercial. <a href="#">1</a>
<b>ITAR</b>	Regulamento Internacional de Tráfico de Armas. <a href="#">1</a>
<b>KF</b>	Filtro de Kalman. <a href="#">1</a>
<b>LM</b>	Algoritmo de Levenberg-Marquardt. <a href="#">1</a>
<b>LSB</b>	Less Significant Bit. <a href="#">1</a>
<b>MEM</b>	Sistemas Micro Eletromecânicos. <a href="#">1</a>
<b>MLP</b>	Perceptron Multicamada. <a href="#">1</a>
<b>MSE</b>	Erro Quadrático Médio. <a href="#">1</a>

<b>RBF</b>	Função de Base Radial. <a href="#">1</a>
<b>ReLU</b>	Unidade Linear Retificada. <a href="#">1</a>
<b>RMSE</b>	Raiz do Erro Quadrático Médio. <a href="#">1</a>
<b>RNA</b>	Rede Neural Artificial. <a href="#">1</a>
<b>TCP</b>	Elemento Terminal de um Robô ou Tool Center Point. <a href="#">1</a>

# Capítulo 1

## Introdução

Nós somos os robôs, prisioneiros da tecnologia que dominamos.

---

Pottier

### 1.1 Motivação

O número de robôs existentes no planeta vem crescendo ininterruptamente nas últimas décadas [Tobe 2017]. As previsões de aumento estão sendo atualizadas para acomodar um novo nicho: o setor de consumo geral. A tendência de expansão deixou de ser exclusividade militar, industrial e comercial. Hoje o departamento de consumo é responsável pelas maiores estimativas do mercado, além disso, a robótica já está inserida em outras atividades humanas como em cirurgias [Jeelani et al. 2015], resgate em acidentes, navegação e demais setores [Pagliarini e Lund 2017], onde o trabalho humano tende a ser perigoso, tedioso, ou economicamente menos vantajoso. A Figura 1.1 apresenta a relação de crescimento de acordo com determinada área. A área comercial e de consumidores representados pelo público geral tiveram suas estimativas atualizadas e podem impactar de maneira expressiva nos próximos anos. O BCG (*Boston Consulting Group*) projeta de forma conservadora que o mercado chegará a US\$87 bilhões em 2025.

A importância da indústria de robôs tem aumentado, não só em função do ganho econômico com as máquinas demandadas, mas também com o ganho de toda a infraestrutura que a envolve: compras, transporte, montagem, manutenção, pesquisa, etc. Neste mercado dinâmico e de alto desempenho o que define a sobrevivência ou o fim das empresas é a qualidade de suas máquinas. Além disso, os robôs desempenham um papel central na indústria manufatureira atualmente [Bahrin et al. 2016]. A indústria 4.0 (quarta revolução industrial)

praticamente dobrou o número de robôs industriais na Europa desde 2004 [Consultants 2014].

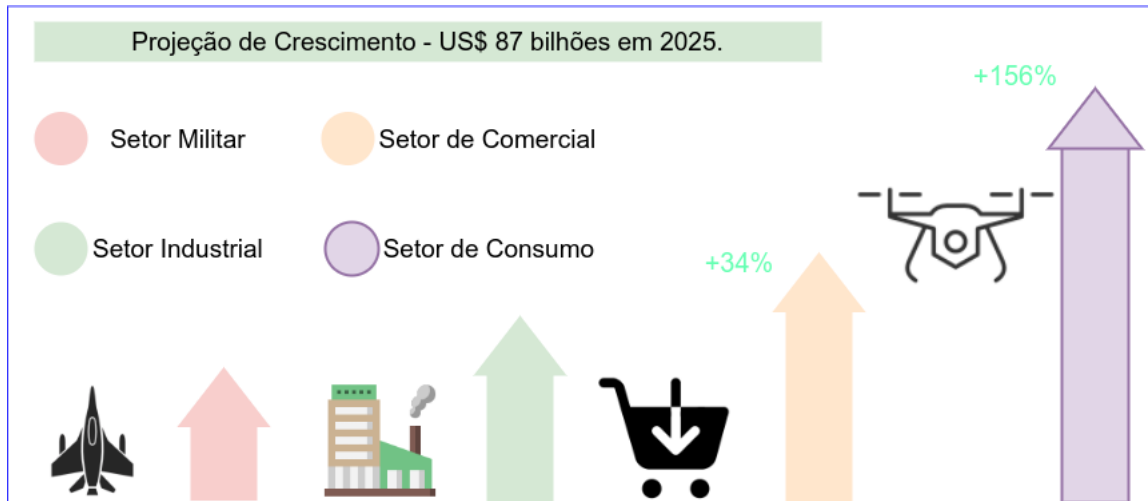


Figura 1.1 – Projeção de crescimento por setor. Imagem desenvolvida pelo autor com base em [Consultants 2014].

Quando se fala em qualidade vêm em mente algumas das características mais importantes de um robô, que são a sua exatidão e a sua repetibilidade. No caso dos robôs industriais, a exatidão é a proximidade ou mínimo desvio que o conjunto posição-orientação real do elemento terminal (efetuador) está do seu valor nominal calculado pelo controlador do robô [INMETRO 2012] ou a capacidade do robô de atingir o conjunto posição-orientação dentro de seu espaço de trabalho (*workspace*) conforme programado. Repetibilidade é o máximo desvio da média, ou desvio-padrão, da ação de repetidamente o robô retornar a um conjunto posição-orientação programado [INMETRO 2012]. Ou seja, a repetibilidade é a medida da capacidade do robô de atingir de forma consistente um ponto especificado. A exatidão é a medida do erro de distância associado ao ponto desejado e ao ponto obtido [Şirinterlikçi et al. 2009].

A calibração do robô é um processo sistêmico de modelagem, medição, identificação numérica das características físicas do mesmo e implementação de um novo modelo [Ginani e Motta 2011, Motta 2006]. Uma segunda definição diz que a calibração é a operação que estabelece sob condições especificadas, numa primeira etapa, uma relação entre os valores e as incertezas de medição fornecidos por padrões e as indicações correspondentes com as incertezas associadas; numa segunda etapa, utiliza esta informação para estabelecer uma relação visando a obtenção de um resultado de medição a partir de uma indicação [INMETRO 2012]. No caso dos robôs industriais pode-se pensá-la como que um método de se minimizar os efeitos das diversas fontes de erros que afetem a exatidão de posicionamento e/ou orientação do elemento terminal do robô, por conta de desvios geométricos e outros aspectos de sua operação [Wang et al. 2019].

No processo de calibração de robôs industriais os erros de posicionamento e/ou de orientação do efetuador são medidos, de forma a registrar os desvios/erros entre as posições desejadas no programa do robô e aquelas efetivamente alcançadas em sua operação. Com estes pontos medidos, utiliza-se um método numérico que minimize uma função desvio ou erro a partir de um modelo cinemático, baseado nos parâmetros de erros [Elatta et al. 2004]. Em seguida, estes parâmetros são identificados através de um método de solução específico, de modo a minimizar esses desvios ou erros entre a posição comandada no controlador do robô e posição atingida pelo elemento terminal ou ferramenta na extremidade do robô. Há também métodos não-paramétricos, com uso de equações de regressão e redes neurais [Sabsch et al. 2017]. Todos os processos tradicionais e paramétricos exigem instrumentos de medição com alta precisão e, portanto, de grande custo para a indústria.

No entanto, não só robôs industriais são submetidos à calibração de seus modelos cinemáticos e/ou dinâmicos. Outros tipos de robôs com muito menos exigências de exatidão, como robôs de limpeza, domésticos, veículos a motor com direção autônoma, *drones*, etc. também são passíveis de serem calibrados, porém com resultados finais contendo tolerâncias muito maiores em termos de sua exatidão posicional, com desvios na faixa de centímetros ou até metros [Jung et al. 2018], bem maiores do que os esperados erros sub milimétricos das suas contrapartes industriais e conseqüentemente mais baratos e rápidos. Isso se dá essencialmente pela ausência de equipamentos de medição, com alta precisão, aplicáveis às condições de operação desses robôs e com custo viável.

## 1.2 Proposta de Tese e sua Justificativa

Métodos de calibração que possuam requisitos metrológicos mais simples podem se tornar mais rápidos, baratos e passíveis de serem aplicados de forma *online* (no decorrer do movimento do robô). Dessa forma, tanto robôs mais simples (*drones*, robôs de limpeza, etc.) como robôs industriais irão se beneficiar de tal incremento. Eficiência de calibração, tempo de operação e redução dos custos são algumas das potencialidades que impactarão no desenvolvimento de tais soluções.

Técnicas com estas características poderiam ser aperfeiçoados para alcançar maior exatidão, o que os tornaria interessantes para aplicação em robôs industriais. Essa possibilidade viria a baratear e popularizar processos até então não utilizados na indústria, por conta de seu alto custo, como no caso da programação *offline* de robôs industriais em operação em linhas de montagem, quando todo o processo de montagem da linha tem de ser interrompido por longo tempo para realizar a calibração dos robôs. Isso criaria um ciclo virtuoso, em que a economia gerada neste processo atrairia investimento em outros que permeiam seu entorno.

Em resumo, pode-se dizer que o aumento da precisão pelo processo de calibração de robôs industriais, com aumento significativo da exatidão posicional dos mesmos a níveis sub

milimétricos, exige um longo período de parada na operação dos robôs, o que gera altos custos. Em contrapartida, robôs com menores exigências de exatidão podem ser calibrados com métodos que são mais rápidos, porém com piores resultados em termos de tolerâncias posicionais. O desenvolvimento de métodos rápidos de calibração de robôs industriais, sem as interrupções de operação em linhas de montagem ou de processos industriais robotizados, tornaria viável a calibração de robôs com redução de custos, pela sua rapidez, e com aumento da qualidade dessa calibração pela exatidão obtida, o que beneficiaria a indústria e melhoraria sua competitividade e produtividade. A Tese propõe a investigação da possibilidade de melhorar essas formas de calibração menos precisas e baratas, usadas em robôs genéricos, como os *drones*. A ideia é estender o conceito a robôs industriais que precisam de mais precisão, permitindo um ajuste eficiente. Essa motivação justifica o trabalho e visa preencher essa lacuna de conhecimento.

### 1.3 Questão a responder

Muitas das formas menos precisas de calibração, referidas anteriormente, costumam fazer uso de Unidades de Medições Inerciais (IMUs) de baixo custo. Seria possível utilizar estes sistemas inerciais para se calibrar robôs industriais de forma rápida, barata e eficiente?

### 1.4 Hipótese a comprovar

A hipótese dessa tese é que os IMUs poderiam ser utilizados em conjunto com técnicas de Machine-Learning (ML) para fazer a navegação estimada do elemento terminal de um robô no curso de sua operação e fornecer subsídios para atualizar o software de controle e posicionamento do dispositivo. Considerando que as Redes Neurais Artificiais podem ser utilizadas em problemas de regressão para identificar e mitigar os erros do sistema de medida e, assim, prever com exatidão suficiente a posição do efetuador do robô, pode-se adotar RNAs para prever a trajetória e contrapor os valores obtidos pela abordagem com os dados nominais do controlador.

### 1.5 Objetivo Geral e sua contribuição

Para desenvolver essa nova metodologia de calibração de robôs industriais, usando micro medidores inerciais, propõe-se o uso de redes neurais artificiais (RNA), como ferramenta de software, para mapear as posições do elemento terminal do robô em um movimento guiado dentro de um espaço de trabalho. O objetivo da pesquisa é disponibilizar um novo método de calibração de IMUs utilizando movimento controlado de um robô industrial e permitir a sua utilização como sensor de medição de posição para a calibração de robôs industriais de

forma simples, rápida, acurada e barata. Além disso, outro objetivo derivado desta pesquisa é a utilização de uma heurística para aproximar a melhor arquitetura possível da rede através do Planejamento Fatorial.

## 1.6 Objetivos específicos

É importante destacar que, até o momento desta pesquisa, sistemas inerciais não foram usados para mapear posição de ferramentas em robôs, sendo seu uso restrito ao mapeamento apenas da orientação da mesma [Du e Zhang 2013, Vieler et al. 2016], fato este que será considerado e detalhado mais à frente neste relatório, no capítulo que trata sobre calibração. Para entender o porquê do não uso, até agora, de IMUs em calibração da posição de robôs, torna-se necessário não só conhecimento sobre calibração de robôs, mas também o entendimento desses dois novos aspectos da metodologia proposta: o funcionamento das IMUs e a construção de Redes Neurais Artificiais. Assim, o desenvolvimento deste relatório inicia-se com a apresentação de um referencial teórico e tem como objetivos específicos o entendimento aprofundado e testes de funcionamento de cada subtema correlato ao mesmo:

1. A calibração de robôs;
2. As Unidades de Medição Inercial – IMUs;
3. As Redes Neurais Artificiais – RNAs.

## 1.7 Estrutura do documento

A revisão bibliográfica de cada um dos temas é apresentada em seu respectivo capítulo e são mescladas quando conveniente. Como os tópicos são cronologicamente e teoricamente independentes, seria confuso se descrito em um único capítulo de revisão bibliográfica.

No Capítulo 2 são apresentadas as metodologias de calibração, mencionando trabalhos, procurando aspectos práticos, econômicos, evolução ao longo dos anos e a vantagem que poderia ser obtida com uma metodologia de calibração usando IMUs de baixo custo. Em seguida, as IMUs são estudadas e individualizados diversos tipos de IMUs e suas características, com referência a trabalhos correlatos. Finalmente, é mostrado um referencial teórico na forma de uma introdução às Redes Neurais Artificiais, seu equacionamento e, também, um esboço dos tipos mais comuns dessas redes, suas aplicações e a forma de uso na presente pesquisa. No Capítulo 3, é mostrado em detalhe a junção das IMUs e RNAs que esse trabalho adotou. Os métodos necessários para criação da abordagem são descritos para permitir a reprodutibilidade do experimento. Os resultados são apresentados no Capítulo 4 e discutidos no Capítulo 5. A Conclusão no Capítulo 6 apresenta ao leitor os principais pontos dessa pesquisa.



## Capítulo 2

# Referencial Teórico e Revisão Bibliográfica

Reúno em mim mesmo a teoria e a prática.

---

Machado de Assis

Este capítulo será dividido em três temas: a Calibração de Robôs, os Sistemas de Medição Inerciais – IMUs e as Redes Neurais Artificiais – RNAs.

Na seção sobre calibração de robôs discorre-se basicamente sobre as metodologias de calibração de robôs, não só os robôs industriais, mas também a calibração de posição de robôs móveis como *drones*, robôs de limpeza e veículos autoguiados mencionando trabalhos na área, buscando aspectos práticos, econômicos e evolução ao longo do tempo.

Na seção sobre IMUs são estudados e individualizados diversos tipos de IMUs, suas características e trabalhos correlatos.

Na seção RNAs, é mostrado o referencial teórico na forma de uma introdução às Redes Neurais Artificiais – RNAs, suas aplicações e a forma de uso na presente pesquisa.

### 2.1 Trabalhos Relacionados

Para ilustrar a relevância da calibração de robôs na indústria e na academia é possível fazer uma pesquisa em sites com bibliotecas de artigos, como Elsevier, IEEE e Springer. Com essa pesquisa é possível obter o número de artigos por ano com relação ao termo buscado. Um dos repositórios que centraliza e indexa diversas coleções de periódicos é o Google Acadêmico, nele é possível encontrar publicações de diversos editoriais, revistas e livros. Uma busca rápida com os termos *robot calibration* no Google Acadêmico resulta em mais de 1700 publicações

somente no ano de 2021. Estendendo a busca para textos publicados desde 2020 esse número chega a 17.700.

Esta tese fez um inventário sobre o termo *robot calibration* e, para isso, utilizou duas ferramentas de buscas acadêmicas com o objetivo de mensurar a relevância do tema e taxa de publicação associada. As pesquisas foram feitas considerando intervalos de 5 anos, desde 2000 até 2021 (2000-2005, 2005-2010, 2010-2015, 2015-atual). A Figura 2.1 resume o número de publicações relacionadas ao tema no intervalo especificado e mostra que o assunto apresenta uma tendência de crescimento considerando a base da CAPES (<[www-periodicos-capes-gov-br.ezl.periodicos.capes.gov.br](http://www-periodicos-capes-gov-br.ezl.periodicos.capes.gov.br)>). Com relação ao Google Acadêmico observa-se a mesma tendência de interesse e pesquisa com relação ao tema.

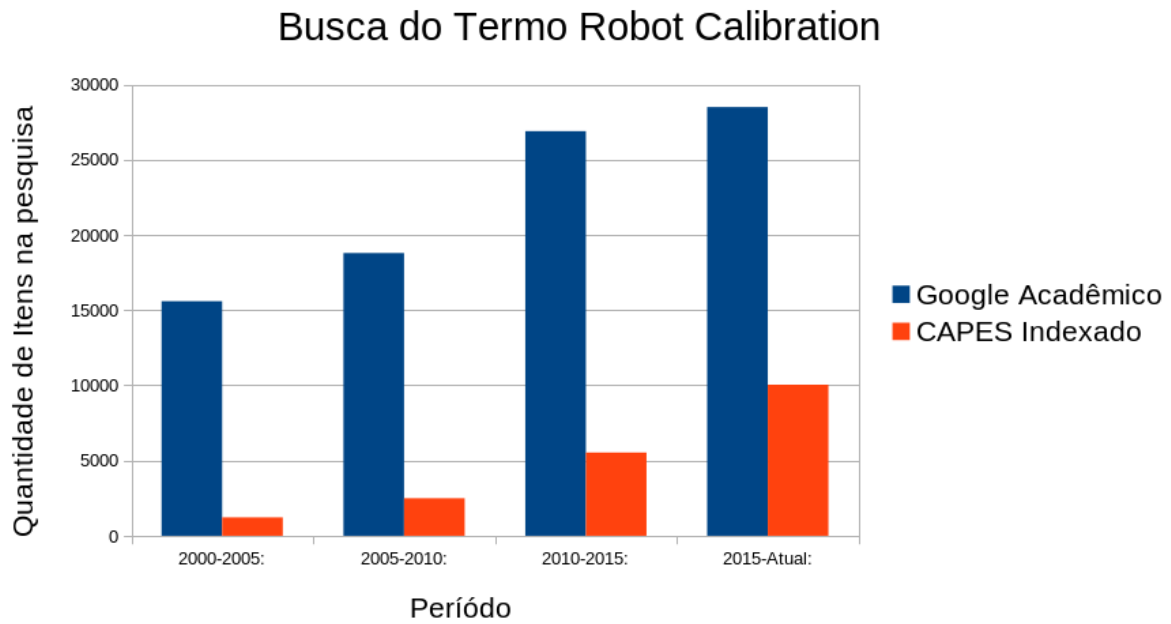


Figura 2.1 – Pesquisa feita pelo Autor na base Google Acadêmico e Capes Indexado com os termos *robot calibration*.

A partir das bases mencionadas o estudo fez um levantamento de artigos da atualidade que englobam alguns dos temas dessa pesquisa. Termos como *robot calibration*, *online robot calibration*, *robotics and AI*, *parametric calibration* e outros que permeiam o assunto, foram buscados em variadas bases. A seguir, pode-se listar alguns dos estudos recentes que adotam os conceitos trabalhados por essa tese.

O trabalho [Du et al. 2020] utiliza uma unidade de medição inercial e um marcador de posição fixados no TCO do robô para medir a orientação e a posição. Para melhorar ainda mais a precisão os autores adotam um filtro híbrido, que integra o filtro Kalman e um filtro de partículas iterativas. O filtro de partículas iterativas é usado para calcular os erros dos

parâmetros cinemáticos robóticos. O texto reforça que com o advento da Indústria 4.0, robôs não serão mais fixos na linha de produção e irão fazer tarefas repetitivas de forma adaptável e versátil. Para isso eles devem corrigir seus próprios parâmetros quase instantaneamente, assim, os métodos tradicionais de calibração offline não podem atender tais requisitos [Du et al. 2020].

No artigo [Gaudreault et al. 2018] os autores introduzem um método de auto calibração usando um dispositivo de medição sem fio de baixo custo montado no flange do robô. A ideia do método é que o robô traga automaticamente o TCP para três posições conhecidas e com diferentes orientações do efetuador. O método de auto calibração foi testado em um robô industrial de seis eixos, o ABB IRB 120. Os parâmetros foram identificados através de regressão linear utilizando mínimos quadrados [Gaudreault et al. 2018]. Os autores mencionam que os métodos atuais são inflexíveis e demorados, além disso, exigem dispositivos de medição caros, como rastreadores a laser, para calibrar o robô e aumentar a sua precisão.

A pesquisa [Nguyen, Le e Kang 2019] propõe um novo método para aumentar a precisão da posição do robô. A abordagem modela e identifica fontes de erros determináveis como erros geométricos e erros de deflexão da junta e, em seguida, usa uma rede neural artificial para compensar os erros de posição do robô, que são causados por fontes não geométricas que são difíceis de modelar como a folga de umas das engrenagens ou a conformidade do *link*. O método é adotado para calibração de um robô industrial Hyundai HH800, voltado para cargas pesadas. Após a correção dos erros geométricos ainda há erros residuais causadas por fontes não geométricas que não possuem linearidade. Para aumentar a precisão do robô, uma RNA é empregada para compensar os erros de posição residual [Nguyen, Le e Kang 2019]. O estudo utilizou uma RNA de três camadas, com 6 neurônios de entrada (representado a junta do robô). A camada de saída é formada por três neurônios que possuem função de ativação “linear”. As saídas são equivalentes a três elementos do vetor de erro de posição do robô [Nguyen, Le e Kang 2019]. A abordagem utiliza do método de retro-propagação para realizar o treinamento.

Os autores da investigação [Wang et al. 2019] introduz um novo método de calibração que trabalha com erros dependentes da deflexão da junta para aumentar a precisão do posicionamento do robô em seu espaço de trabalho. Os autores dividem a área de trabalho do ângulo da junta em diversas regiões, de acordo com a distribuição de massa do robô. Em seguida, os parâmetros geométricos são modelados e identificados usando o modelo Denavit – Hartenberg. Para mitigar as fontes de erros não geométricas (difíceis de mensurar) o trabalho aplica uma RNA como fator de compensação. Para os testes a metodologia empregou um robô com 8 graus de liberdade.

O estudo [Le e Kang 2020] propôs um algoritmo de calibração para melhorar a precisão da posição do robô. Primeiro, os autores identificaram os parâmetros cinemáticos e os parâmetros de conformidade do robô para melhorar sua precisão usando o modelo de deflexão da

junta e a técnica de calibração do modelo cinemático convencional. Em seguida, adotaram uma rede neural artificial para compensar os erros não modelados, para otimizar os parâmetros da rede neural utilizaram uma técnica conhecida como erva daninha invasiva, que se caracteriza pela boa taxa de conversão. Para demonstrar a técnica sugerida, um robô HH800 foi utilizado durante o estudo experimental do algoritmo proposto.

O artigo [Vieler et al. 2016] sugere um método para calibração online de robôs de média precisão, capaz de fazer a verificação online de alta precisão usando sensores inerciais. A metodologia anexa ao TCP do robô uma IMU e um marcador de posição para medir as posições do robô em tempo real, a abordagem utiliza o método Denavit-Hartenberg para fazer a modelagem cinemática padrão. Com as posições medidas utiliza-se um filtro de Kalman para estimar as posições precisas e um filtro estendido de Kalman para estimar erros de parâmetro usando as poses estimadas, esses erros são usados para corrigir os parâmetros cinemáticos. O método não requer etapas complexas.

O estudo [Xiao et al. 2019] introduz um método online de auto-calibração de uma IMU para sistemas visuais inerciais, equipado com um sensor inercial de baixo custo. As IMUs baseadas em sistemas Microeletromecânicos de Baixo Custo (MEMS) são afetadas por fatores de escala imprecisos, desalinhamentos de eixo e erros de sensibilidade [Xiao et al. 2019]. Esses erros podem influenciar significativamente o desempenho dos métodos visual-inerciais. O objetivo do método é realizar a estimativa de pose 3D e calibração IMU online de modo paralelo, sem qualquer equipamento externo. Os autores desenvolveram um método de pré-integração para lidar com a propagação de erros dos parâmetros intrínsecos da IMU. Os resultados demonstram que a abordagem introduzida é capaz de calibrar com precisão todos os parâmetros considerados e em tempo real.

Na trabalho [Du e Zhang 2013] apresenta uma abordagem de auto-calibração de robô baseado em unidade de medição inercial. O método incorpora o algoritmo de quatérnio fatorado e o filtro de Kalman para estimar a orientação do IMU. A metodologia usou um filtro de Kalman estendido para estimar os erros dos parâmetros cinemáticos e os autores argumentam que o método proposto resultará em maior confiabilidade e precisão. Em comparação com os métodos baseado em visão, a vantagem desta abordagem é que não há etapas complexas, como calibração de câmera, captura de imagens e detecção de canto [Du e Zhang 2013]. Os experimentos contaram com um robô GOOGOL GRB3016 mostram que este método é preferível a outros métodos que se baseiam em visão ou medição.

Durante a pesquisa percebeu-se que existem trabalhos sobre calibração online e offline, estudos sobre a calibração de IMUs, trabalhos sobre calibração utilizando filtros de Kalman e estudos que realizam a calibração de modo offline e necessitam de instrumentos sofisticados de medição. Porém, no entendimento dessa pesquisa, sistemas de calibração online utilizando IMUs de baixo custo e RNAs destinadas a modelos de regressão, ainda não foram investigados, exceto pelo trabalho [Campos e Motta 2021] que é derivado dessa tese.

## 2.2 Calibração de Robôs

O termo calibração, como já definido anteriormente, pode ser explicado como a operação que estabelece, sob condições especificadas, numa primeira etapa, uma relação entre os valores e as incertezas de medição fornecidos por padrões e as indicações correspondentes com as incertezas associadas; numa segunda etapa, utiliza esta informação para estabelecer uma relação visando a obtenção de um resultado de medição a partir de uma indicação [INMETRO 2012]. Também pode ser visto como o conjunto de procedimentos projetados para estabelecer uma correspondência entre uma quantidade física conhecida ou padronizada e as leituras de um instrumento no qual essa quantidade é medida [Hofmann-Wellenhof et al. 2003]. A calibração de robô é um meio efetivo de melhorar a exatidão do dispositivo e garantir uma faixa de operação ideal [Du e Zhang 2013].

O termo calibração do robô é descrito como a calibração do modelo cinemático desse robô, e as equações de calibração buscam compensar desvios geométricos e outros efeitos que não foram incluídos no modelo nominal pelo fabricante do robô ou devido à variação destes ao longo da vida útil do robô ou devido ao desgaste natural e desalinhamento de suas partes mecânicas. Esses desvios e efeitos podem ser causados por:

- a Variação de dimensões por efeito da temperatura;
- b Desvios geométricos em juntas;
- c Erros nos parâmetros cinemáticos do modelo (nos comprimentos dos segmentos e ângulos de torção);
- d Elasticidade dos segmentos e das juntas (transmissão);
- e Deformação da estrutura causada por forças externas;
- f Movimento da base do robô;
- g Folgas transversais e axiais de eixos, mancais e na transmissão (*gear backlash*);
- h Desvios resultantes de vibrações provocadas pelos dentes das engrenagens da transmissão;
- i Erros no controlador ou erro nos servomecanismos;
- j Desalinhamento e excentricidade de elementos da transmissão (inclusive sensores).

Estes desvios ou erros foram estudados em diversas circunstâncias e ocasiões [Shang et al. 2019, Wang et al. 2018, Schillreff et al. 2017, Zheng e Zhang 2016, Meggiolaro et al. 2004, Mavroidis et al. 1997, Schroer et al. 1997, Tchou 1992, Judd e Knasinski 1990]. De

forma geral, há três tipos de erros que devem ser minimizados no processo de calibração. Tais erros são associados a uma determinada parte e dinâmica do sistema e podem ser classificados em:

1. **Dinâmicos** – servo-sistemas, fricção e inércia;
2. **Estruturais** – fricção, carga e temperatura;
3. **Cinemáticos** – espaço de junta, atuador e comprimento dos elos.

No que concerne à calibração destas diversas fontes de erros, existem a priori duas abordagens. A primeira é chamada de paramétrica [Kolyubin et al. 2015, Marie et al. 2013], em que o erro está relacionado com os parâmetros que refletem a geometria e outras características mecânicas como elasticidade, excentricidade, folga e outros, como já citado anteriormente. A segunda abordagem é chamada de não-paramétrica, em que o modelo cinemático e/ou dinâmico do robô é substituído por equações de regressão não-linear e outras formas mais abstratas de tratamento de equações, como o caso de redes neurais [Sabsch et al. 2017, Cicco et al. 2015].

A abordagem paramétrica pode ser dividida em dois grupos: modelo cinemático ou modelo de erro. Na primeira, pode-se modelar a posição e a orientação, ou apenas a posição, em função do comprimento dos elos e orientação relativa entre os elos. Na segunda, pode-se modelar o erro de posição-orientação ou apenas de posição.

No caso específico desta pesquisa, o objetivo é corrigir apenas a posição, uma vez que o robô objeto da calibração será um robô paralelo IRB-360, que mantém sua orientação durante seu movimento. Outro motivo para isto é que a calibração para correção da orientação exige instrumentos de medição com alta precisão para medir os ângulos de orientação do efetuador do robô, em todos os pontos de medição, o que faz com que esse tipo de correção seja predominantemente ignorado na indústria, sendo que normalmente apenas a posição seja medida e corrigida. A Figura 2.2 ilustra o robô disponível. Conhecido como Flex Picker, ele opera com capacidade de carga entre 1 e 8 kg e apresenta um excelente desempenho, com precisão e rapidez.

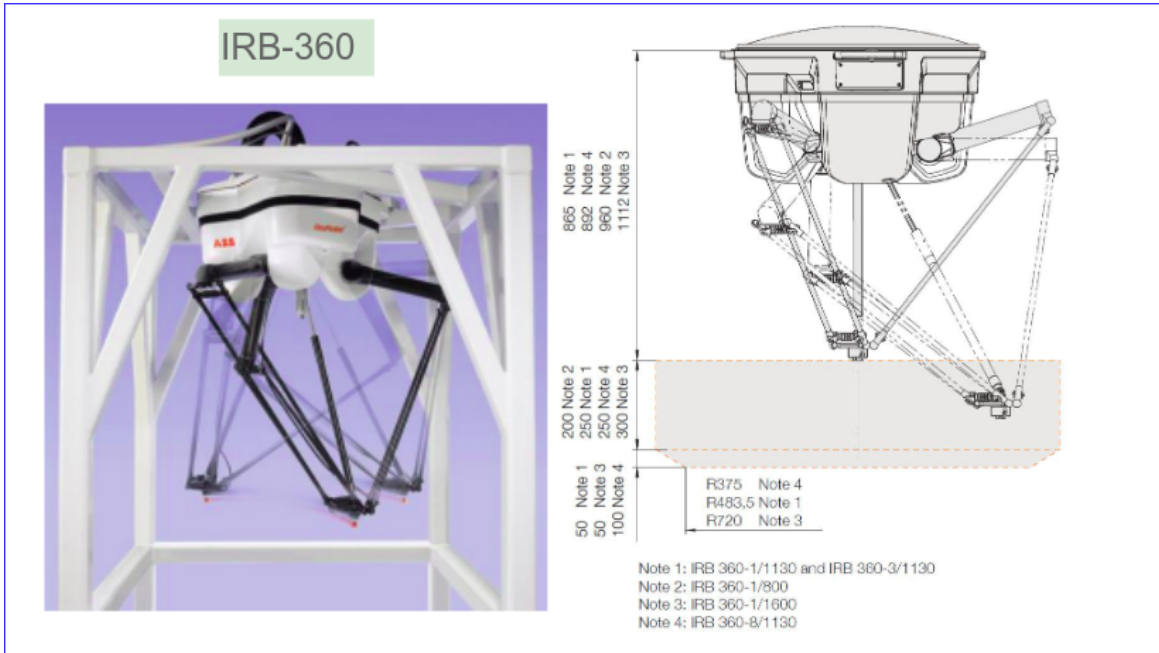


Figura 2.2 – Robô IRB 360 presente no laboratório. Fonte: Compilação do Autor e [POWER-MIG 2020]

A abordagem paramétrica por modelo de erro tem a vantagem de eliminar os parâmetros redundantes [Tong et al. 2003, Zhuang et al. 1990], enquanto a abordagem por modelo cinemático tem a desvantagem de necessitar que os parâmetros redundantes sejam eliminados antes do processo de estimação [Meggiolaro 2000, Motta e McMaster 1999, Chen e Chao 1986]. A abordagem paramétrica ainda pode ser dividida em:

1. **Geométrica:** onde parâmetros de uma junta são identificados de modo serial [Wu et al. 2015];
2. **Sistêmica:** onde os parâmetros são identificados simultaneamente.

Segundo [Duelen e Schröer 1991] a abordagem sistêmica tem as vantagens de requerer menos pontos de medições, não necessitar de movimentos que dependam de relações geométricas e permitir a identificação de parâmetros não-geométricos (elasticidade e folga da transmissão, flexibilidade dos segmentos, etc). Ainda dentro da abordagem sistêmica é possível eliminar a necessidade de um meio externo de localização do elemento terminal, fechando a cadeia cinemática. Assim, ainda pode-se dividir a abordagem paramétrica também em cadeia aberta e fechada.

A abordagem paramétrica de modelos cinemáticos geométricos aplicada a cadeias abertas pode ser dividida em métodos de ponto e métodos de pose. No método de ponto medem-se pontos em cada elo móvel do manipulador e no método de pose (posição e orientação) o ponto de medição está restrito ao elemento terminal.

Os métodos de calibração também podem ser divididos em métodos iterativos que minimizam o quadrado do erro medido ou, no caso da abordagem geométrica, os parâmetros podem ser calculados diretamente usando fórmulas fechadas ou métodos diretos. Vários métodos para modelagem cinemática de robôs foram propostos na literatura, alguns deles, apesar de datados dos anos 80/90, ainda são relevantes e estudados.

### 2.2.1 Modelo Denavit-Hartenberg

A notação Denavit-Hartenberg é amplamente aplicada e utilizada na modelagem cinemática de robôs seriais. O sistema de coordenadas é definido explicitamente, o que torna a abordagem consistente. Considerando um robô com  $n$  graus de liberdade (n-DOF) o modelo irá necessitar de  $4r + 2p + 6$  parâmetros (onde  $r$  é número de juntas rotativas e  $p$  representa o número de juntas prismáticas). Tais parâmetros devem ser independentes e não redundantes. Todavia, o modelo DH devido a uma série de restrições limita o sistema de coordenadas de acordo com a configuração do robô. Além do mais, eixos articulares paralelos estão sujeitos a mudanças drásticas devido a pequenas alterações de estado, isso pode levar a uma singularidade e conseqüentemente à descontinuidade do robô [Chen-Gang et al. 2014, Veitschegger e Wu 1988]. Para mitigar o problema da completude na etapa de identificação de parâmetros, alguns autores sugerem um sistema de mudança do sistema de coordenadas [Veitschegger e Wu 1988], o modelo melhorado também mitiga os problemas de continuidade.

### 2.2.2 Modelo S

O modelo S foi construído por Stone [Stone 1987], que adicionou dois parâmetros ao modelo DH, assim é possível descrever a transformação do sistema de coordenadas de forma completa. Isso torna o modelo mais flexível. No entanto, são necessárias 6 ligações de parâmetros para desenvolver o modelo completo. Além disso, os parâmetros adicionais tornam a calibração redundante, o que leva a uma perda de precisão na identificação dos parâmetros cinemáticos [Chen-Gang et al. 2014].

### 2.2.3 Modelo CPC

Semelhante ao método S, o modelo CPC também adiciona dois parâmetros ao modelo DH [Chen-Gang et al. 2014, Zhuang et al. 1993, Zhuang e Roth 1993], para compensar a incompletude e a não continuidade do modelo DH. Três parâmetros são empregados para descrever a direção do eixo da junta e três parâmetros são aplicados para explicar a relação de posição entre as origens do sistema de coordenadas adjacente. Os parâmetros redundantes podem ser eliminados no processo de calibração [Chen-Gang et al. 2014]. Para eliminar o problema da singularidade o trabalho [Zhuang et al. 1993] propôs uma modificação no



modelo CPC usando quatro parâmetros  $(\alpha, \beta, x, y)$  para descrever a transformação entre os sistemas de coordenadas adjacentes, tal sistema pode ser chamado de MCPC. Dessa forma consegue manter as características de completude e continuidade do modelo CPC. O ângulo de revolução dos eixos  $x$  e  $y$  são representados por  $\alpha$  e  $\beta$ , a partir da  $i$ -ésima coordenada até  $i+1$ -ésima, enquanto  $x$  e  $y$  representam a translação a partir da origem. Os outros dois parâmetros ( $\theta$  e  $z$ ) representam o ângulo de revolução e de translação na direção do eixo  $z$  [Chen-Gang et al. 2014]. Assim, o sistema MCPC é capaz de satisfazer qualquer arranjo de coordenadas.

#### 2.2.4 Modelo Referência Zero

Tal modelo adota um sistema coordenadas cujo referencial é fixo (chamado referência zero) [Zhong et al. 1996]. O sistema é implementado no robô e, portanto, é capaz de descrever cada eixo e junta de acordo com sua configuração. Seu modo de ajuste é flexível, no entanto, há alguns problemas para validar a eliminação de parâmetros redundantes. A falta de meios sistemáticos para validar o modelo limita a calibração cinemática. É completamente diferente do método DH e CPC [Chen-Gang et al. 2014].

#### 2.2.5 Modelo POE

O modelo POE foi descrito baseando-se no sistema de referência zero. Utiliza em sua formulação seis parâmetros  $(6r + 3p + \theta)$  [Ruibo He et al. 2010], onde, a variável  $r$  é responsável por endereçar as juntas de revolução e  $p$  as prismáticas. Tal abordagem direciona muita atenção ao erro final e considera cada tipo de erro de forma separada. Assim tal modelo consegue uma representação abrangente do erro e mais elaborada do que os modelos anteriores. O erro zero da junta pode ser capturado e dessa forma a precisão final da pose do efetuador é mais satisfatória. Seu modelo cinemático utiliza um sistema de coordenadas base e outro como suporte, isso simplifica sua aplicação e torna o sistema universal. Outra variação do modelo é proposta usando um método que descreve a cinemática com base em um sistema de coordenadas local. Dessa forma, todos os eixos das juntas são representados com o mesmo sistema de coordenadas.

#### 2.2.6 Classificação do Modelo

Um modelo é classificado de acordo com sua capacidade de completude, continuidade e parâmetros mínimos (parâmetros não redundantes) [Schröer et al. 1997]. Quando as três métricas são atingidas a calibração é viável.

- Completude: O modelo completo deve possuir parâmetros suficientes para descrever

qualquer desvio relativo aos parâmetros cinemáticos em comparação com os valores nominais [Zhuang e Roth 2018, Ruibo He et al. 2010];

- Continuidade: Pequenas mudanças na estrutura geométrica do robô deve corresponder a pequenas alterações nos parâmetros cinemáticos;
- Minimalidade ou não redundância: O modelo cinemático deve incluir apenas um número mínimo de parâmetros. O modelo de erro não deve apresentar redundância no processo de calibração [Schröer et al. 1997].

A Tabela 2.1 resume tais características dos sistemas apresentados.

Tabela 2.1 – Classificação do Métodos

Modelo	Não Redundante	Completo	Continuidade	Viabilidade de Calibração
DH	Sim	Não	Não	Não
DH Melhorado	Sim	Sim	Sim	Sim
S	Não	Sim	Não	Não
CPC	Sim	Sim	Sim	Sim
MCPC	Sim	Sim	Sim	Sim
Referência Zero	Não	Sim	Sim	Não
POE	Sim	Sim	Sim	Sim

A Tabela 2.2 apresenta as características de alguns trabalhos e sua evolução cronológica. Apesar da literatura de alguns métodos não ser recente, os mesmos ainda são utilizados.

Tabela 2.2 – Subdivisões das abordagens paramétricas para calibração de robôs

Autor	Paramétrico	Modelo	Parâmetro	Método de Modelagem	Algoritmo	Tipo de Medição
[Hayati e Mirmirani 1985]	Sim	Erro	Geométrico	DH + Correção dos eixos paralelos	Método Iterativo próprio	Aberta
[Chen e Chao 1986]	Sim	Cinemático	Geométrico/ Não geométrico	Translação xyz + Euler	Sistema Linearizado	Aberta
[Judd e Knasinski 1990]	Sim	Cinemático	Geométrico/ Não geométrico	DH + Correção dos eixos paralelos	Método Próprio	Aberta
[Bennett e Hollerbach 1991]	Sim	Cinemático	Geométrico	DH	Levenberg–Marquardt	Fechada
[Duelen e Schröer 1991]	Sim	Cinemático	Geométrico/ Não geométrico	DH	Newton	Aberta
[Zhuang et al. 1992]	Sim	Erro	Geométrico	DH + Correção dos eixos paralelos	Levenberg–Marquardt	Aberta
[Goswami et al. 1993]	Sim	Cinemático	Geométrico/ Não geométrico	S Model	Levenberg–Marquardt	Aberta
[Hollerbach e Lokhorst 1995]	Sim	Cinemático	Geométrico/ Não geométrico	Especial	Mínimos Quadrados	Aberta e Fechada
[Zhuang e Roth 1995]	Sim	Cinemático	Geométrico	CPC	Sistema Linear	Aberta
[Zhuang e Roth 1995]	Sim	Cinemático	Geométrico	MCPE	Levenberg–Marquardt	Aberta
[Ruibo He et al. 2010]	Sim	Erro	Geométrico	POE	Mínimos Quadrados	Aberta

### 2.2.7 Fontes de Erros

Sabe-se que as principais fontes de erro (*offsets* de junta e comprimentos de elos) são responsáveis por cerca de 90% do erro médio quadrático de posição. As variações nos parâmetros que definem as relações geométricas entre os segmentos são responsáveis por cerca de 5% e as transmissões por 1% [Judd e Knasinski 1990]. Assim, uma divisão dos tipos de parâmetros aparece de acordo com o que se deseja modelar: parâmetros geométricos e parâmetros não geométricos [Bai et al. 2019]. Os primeiros são o comprimento do elo, ângulos de torção entre eixos e desvios ou deslocamentos da junta. Os seguintes são responsáveis por deformações constantes (flexões e torções), torções e deslocamentos de juntas, dependendo da configuração e carga, atrito, folga, excentricidade da transmissão, etc.

### 2.2.8 Modelo Paramétrico

Um modelo cinemático completo é o que possui parâmetros suficientes para expressar qualquer variação da estrutura do robô em relação ao projeto original (nominal) [Zhuang e Roth 2018]. A Figura 2.3 exemplifica um modelo paramétrico com 2-DOF, os anglos relativos são obtidos através da lei dos cossenos.

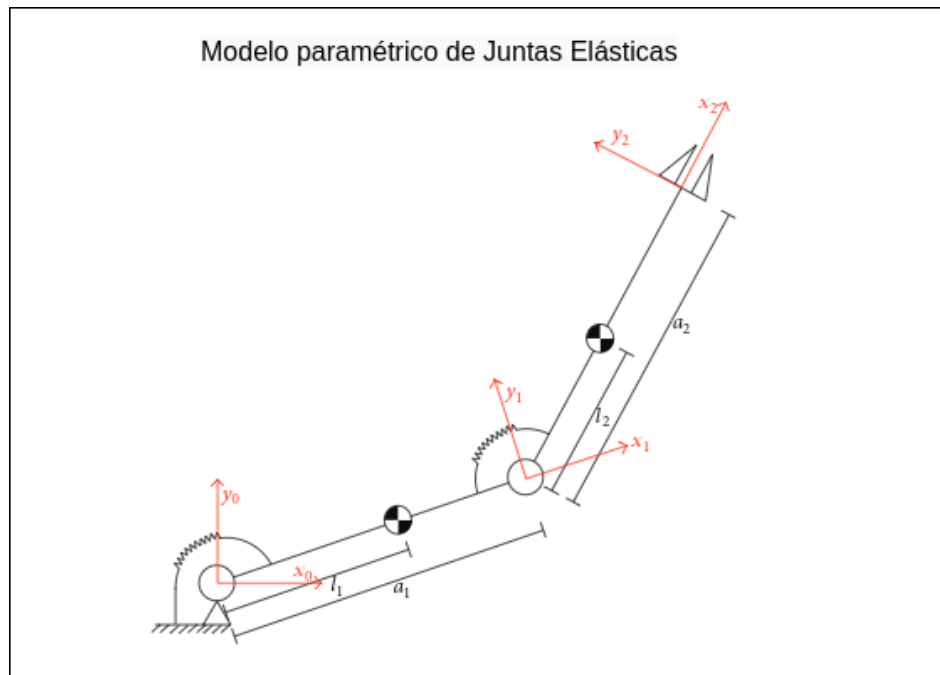


Figura 2.3 – Modelo paramétrico de dois graus de liberdade, considerando o eixo x e y do plano [Zollo et al. 2015].

O número máximo de parâmetros que podem ser determinados é igual ao máximo posto do Jacobiano da matriz, que representa o total de transformações da base do robô ao seu

elemento terminal que, por sua vez, é igual ao número de equações necessárias para especificar completamente a coordenada e a orientação do elemento terminal e das juntas [Everett e Hsu 1988]. Bennet e Hollerbach (1991) provaram que um vetor de parâmetros  $\Phi$  é uma solução única da calibração em vizinhança arbitrária se, e somente se, o posto do Jacobiano for cheio.

Como o modelo a ser descrito se resume em um problema vetorial o uso da matriz Jacobiana torna-se propício, pois a mesma é formada pelas derivadas parciais de primeira ordem de uma função vetorial. Assim, é possível descrever o sistema em termos matriciais. Uma matriz  $4 \times 4$  é suficiente para descrever o movimento de rotação, translação ou ambos. A Figura 2.4 exemplifica esse conceito considerando um modelo com apenas 1-DOF. Parte da matriz ( $H_{rot}$ ) descreve rotação, que pode ser em torno dos eixos  $X$ ,  $Y$ ,  $Z$ , ou de uma combinação dos mesmos.  $H$  é uma matriz  $4 \times 4$  que pode descrever a rotação, a translação ou ambas em uma única matriz.

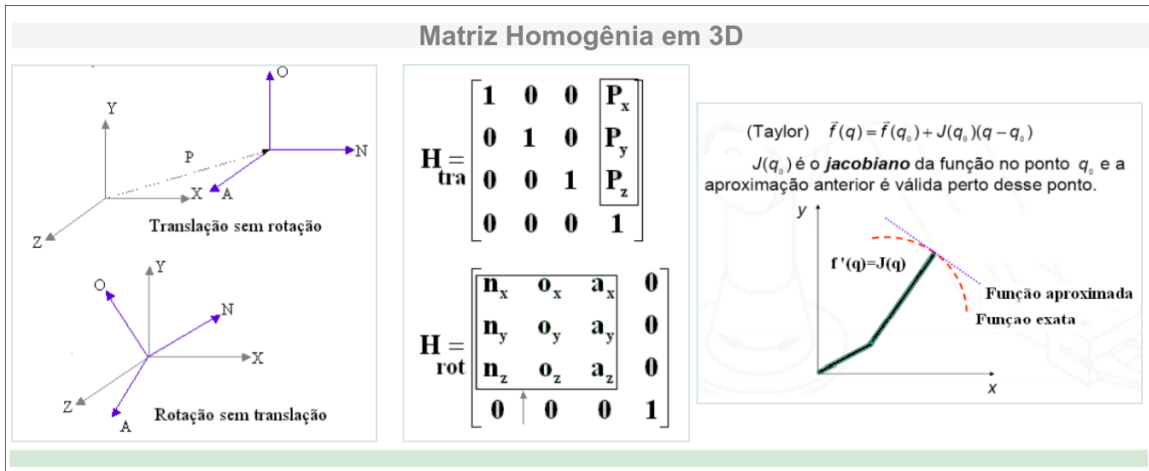


Figura 2.4 – Matriz transformação para um grau de liberdade e seu Jacobiano. Desenvolvido com base em [CAMPOS 2006]

O conceito matricial pode ser extrapolado para N-DOF, o mapeamento é feito de forma similar para cada grau e o resultado final é o produto matricial de cada movimento mapeado (rotação ou translação). A Figura 2.5 (A) contextualiza tal afirmação. O grau de liberdade de um sistema pode estar relacionado com a posição ou a velocidade, relativa a direção do movimento.

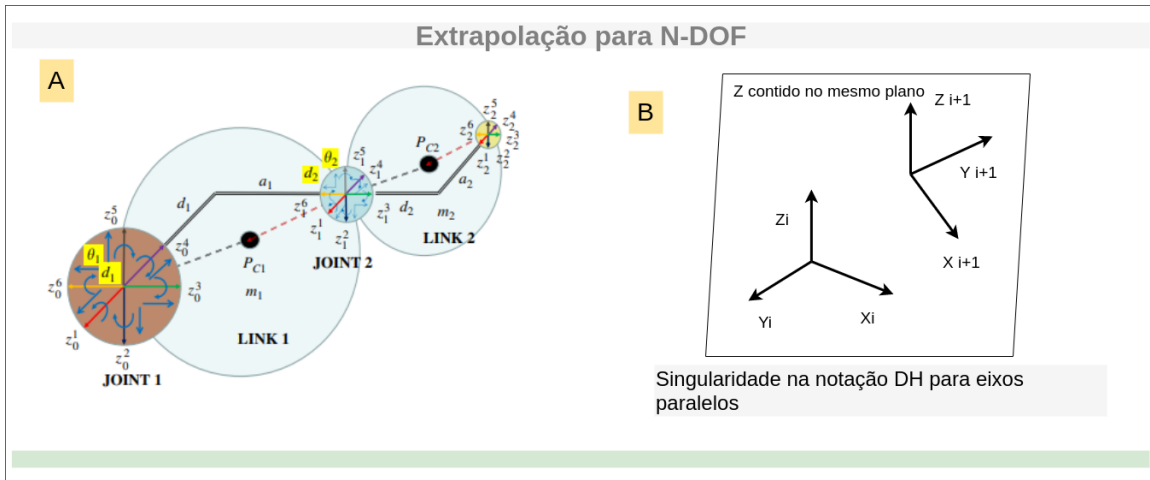


Figura 2.5 – Extrapolação para mais graus de liberdade, considerando a possibilidade de rotação e translação (A). Singularidade na notação DH para eixos paralelos (B). Fonte: modificado de [Djuric, Saidi e ElMaraghy 2012]

Caso o Jacobiano venha a perder seu posto (número de linhas não-nulas da matriz), antes ou mesmo durante a simulação que leva à solução, o número de parâmetros identificáveis diminui. Três razões principais podem ocasionar perda de posto: tipo de modelagem, excitação insuficiente e singularidade transiente.

Para exemplificar a queda do posto matricial vamos tomar como exemplo o caso do modelo DH. A notação DH apresenta uma falha quando descreve a mudança de referência onde os eixos das juntas consecutivas são paralelos e a perpendicular mútua dos eixos (da direção Z) não é única, implicando em um problema de modelagem. No segundo caso, quando o mecanismo traz um número insuficiente de posições medidas, o número de equações necessárias torna-se insuficiente, isso faz com que o posto caia. Se as configurações não envolvem todo o vetor de parâmetros, podem ocorrer dependências entre os parâmetros fazendo também com que o posto do Jacobiano caia.

No último caso o que se tem é uma falha numérica que tende a levar o vetor solução a posições singulares locais. Isto traz singularidade uma vez que uma pequena variação na posição espacial das linhas pode gerar uma grande distorção na coordenada descrita por meio da notação DH. Este tipo problema está associado ao algoritmo de busca usado. Pode-se evitá-lo empregando um algoritmo que minimize a variação dos parâmetros como o Algoritmo de Levenberg-Marquardt (LM). Ainda assim não funciona bem e haverá problemas de indeterminação. A indeterminação é intrínseca ao modelo, não depende do método de solução.

### 2.2.9 Sistemas de Medição

A posição do efetuador final irá determinar a precisão da calibração cinemática. Para tal, dispositivos de medição externos são aplicados. As medições variam entre os dispositivos e a complexidade da operação também é relativa, podem-se listar os seguintes meios de medição [Gaudreault et al. 2018, Gan e Dai 2011, Ha 2008, Driels e Pathre 1991]:

- Máquina de medição por coordenadas;
- Cubo com furos localizados onde o efetuador é inserido para calibração;
- Mesa com furos localizados no efetuador.

Também há outros meios de medição, como por exemplo:

- Laser;
- Ultra-som;
- Infravermelho;
- Teodolitos;
- Tempo de propagação de uma onda acústica emitida do efetuador até microfones.

A pesquisa bibliográfica estudada mostra que os problemas matemáticos clássicos de calibração paramétrica foram resolvidos através de algoritmos desenvolvidos principalmente entre as décadas de 1980 a 2010, e a maioria dos artigos atuais vistos que tratam de calibração robótica dispõe basicamente sobre métodos de medição da posição e orientação do elemento terminal do robô [Gaudreault et al. 2018].

Nos demais casos, como os já citados robôs de limpeza, automotores de guiagem autônoma, *drones*, e outros, o que se procura compensar, não são mais os desvios e efeitos que não foram levados em conta no modelo nominal ou na variação deste ao longo da vida útil do robô, mas principalmente seu posicionamento em um ambiente, muitas vezes aberto e variável, contendo fatores inesperados como o encontro com obstáculos, mudança de trajetória, etc.

No estudo da calibração robótica a correção do posicionamento dos robôs móveis pode ser tomada como o sensoriamento para correção imediata de posição. No termo utilizado no estudo de robôs móveis se define a palavra calibração como conjunto de procedimentos destinados a estabelecer uma correspondência entre a grandeza física real de posição e a posição mostrada pelos sensores de posicionamento do robô [Hofmann-Wellenhof et al. 2003], principalmente quando se pensa em termos de sua navegação. Assim, outros métodos de

calibração para a posição de robôs móveis menos precisos [Ryoichi et al. 2018] poderiam ser empregados para calibração de robôs industriais. Esses métodos podem ser exemplificados por:

- Restrição de posicionamento;
- Visão com câmeras estéreo/monoculares;
- Radiofrequência (RFID);
- GPS;
- Laserfinder (LIDAR);
- Ultrassom
- IMUs para orientação.

É justamente a lacuna entre esses citados métodos e os métodos de calibração paramétricos que este estudo pretende preencher, para proporcionar um processo de calibração da posição de robôs industriais de forma mais rápida e barata usando IMUs com Redes Neurais em movimento guiado.

### 2.2.10 Calibração online e offline

No que tange ao aspecto prático vem em mente duas propostas distintas: o processo de calibração *offline* e o *online*, sendo que esse último também pode ser chamado de auto-calibração. De forma geral, no processo de calibração *offline* o robô permanece parado entre intervalos de trabalho e um programa para atingir certas posições dentro de seu volume de serviço é empregado. Movimentando seu ponto extremo ou ferramenta até essas posições programadas, comparam-se as posições medidas por algum sistema de medição com precisão suficiente, com as programadas no controlador do robô, calculando-se os erros. A partir destes erros os parâmetros geométricos são identificados, a fim de refletir melhor o modelo cinemático de controle.

Nesse caso, o procedimento de calibração clássico envolve antes o desenvolvimento de um modelo cinemático cujos parâmetros representam com precisão o robô real. Em seguida, características do robô especificamente selecionadas são medidas utilizando instrumentos com precisão conhecida. Este procedimento é usado para calcular o conjunto de valores de parâmetros que, quando introduzido no modelo nominal do robô, representa com precisão o comportamento do robô avaliado [Tong et al. 2003, Zhuang et al. 1990, Motta e McMaster 1999]

De posse dos novos valores dos parâmetros, o modelo no software de controle de posição é corrigido. A calibração *offline* permite a “programação *offline*” dos manipuladores, que se torna possível após a calibração, uma vez que os erros remanescentes não corrigíveis, que podem ser constatados com ensaios padrão de repetibilidade de posição nos manipuladores industriais, têm valores pequenos [Goswami et al. 1993, Zhong et al. 1996], na faixa de 0.1 mm. Além disso, para que a programação *offline* tenha sucesso, é necessária a medição e calibração da posição com alta precisão [Chen-Gang et al. 2014].

Já no processo de calibração *online*, também chamado auto-calibração, o modelo cinemático no controlador ou as coordenadas de posição do robô são ajustados enquanto o robô realiza seu serviço [Maye et al. 2016], não sendo necessário interromper sua operação para efetuar a calibração, tendo ainda como conveniência a possibilidade do monitoramento e a adaptação do seu modelo ao seu desgaste natural futuro.

Do ponto de vista econômico, o benefício imediato que se vê no método *online* em comparação com o primeiro método, é principalmente o fato de que no método online poderia não ser necessário que se interrompesse o serviço do robô na linha de manufatura para calibrá-lo, caso fosse viável a colocação de uma IMU na ponta da ferramenta capaz de medir sua posição enquanto este trabalha, além da vantagem de se evitar a operação de complexas máquinas de medição, sistemas de interferometria, rastreamento a laser, etc.

Dentre esses raros estudos de calibração *online* de robôs apresentadas na literatura, alguns têm buscado métodos mais baratos de fazê-lo, como por exemplo, no caso onde são usados medidores *wireless* montados na flange do robô para detecção de erros de medição [Gaudreault et al. 2018]. Em que pese sua precisão se encontre na faixa de dezenas de microns, o equipamento de medição ainda é grande, caro (podendo custar dezenas de milhares de dólares) e necessita de uma pequena parada no trabalho do mesmo para que seja efetuada, de fato, a auto-calibração.

Não obstante o pouco uso destes métodos *online*, o desenvolvimento de MEMS (*micro eletromechanical systems*), que despertou interesse de pesquisadores em várias áreas ao longo do tempo [Judy 2001], têm agora subsidiado pesquisas em calibração de robôs, usando sistemas de unidades medidoras inerciais (IMU’s) com giroscópios e acelerômetros acoplados, para efetuar a medição da orientação nos pontos programados [Du e Zhang 2013].

Esses medidores inerciais, que costumavam ser grandes e caros e eram usados apenas na indústria aeroespacial, já se encontram disponíveis a preços abaixo de cem dólares [Sparkfun 2018] e com peso próximo de vinte gramas.

Assim, os métodos baseados em IMUs poderiam retirar a complexidade de operação quanto ao hardware, como braços medidores, sistemas de interferometria, rastreamento a laser, etc., que estaria presente no método tradicional de calibração, e embarcá-la no software de controle do sistema da IMU. Isso pode ser considerado bom, na medida em que,



diferentemente do hardware, o software pode ser replicado praticamente sem custo por quem o construiu.

Mesmo contendo ótimas características de peso e a possibilidade de efetuar a calibração *online* os sistemas de IMUs baseados em MEMS sofrem com a rápida acumulação de erros na medição de posição e foram usados, até agora, apenas para calibrar a orientação da ponta da ferramenta em robôs seriais [Du e Zhang 2013, Vieler et al. 2016], e não sua posição.

Esses erros de medição poderiam, como já foi comentado antes, ser compensados por software, para que pudessem ser usados na etapa de medição para a calibração de robôs, o que é o objetivo dessa pesquisa.

No que se refere ao software e seus custos computacionais, os métodos numéricos usados para minimização da função de erro que compara os valores desejados na programação e os efetivamente obtidos na operação do robô, ou seja, na identificação dos parâmetros de erro, têm peso computacional importante na calibração. Sendo assim, algoritmos mais leves, velozes e com maior garantia de convergência tendem a ser o objetivo primário de quem se propõe a realizar um processo de calibração em robôs.

Como um exemplo de método numérico usado na calibração de robôs, o algoritmo de Levenberg-Marquardt (LMA) usa a técnica de aproximação de mínimos quadrados não lineares para ajustar a curva da função de erro dos pontos programados, onde se atualiza um valor de erro  $e$  de forma iterativa (equação 1). Considere o problema de minimização da função  $f(x_n)$  (função objetivo), onde  $J$  denota o jacobiano e  $H$  é a matriz Hessiana de  $f$  calculada na aproximação  $x_k$ . O método iterativo consiste em gerar uma sequência de  $x_k$  de aproximações da solução  $x^*$ , onde a cada iteração o método considera uma nova aproximação. O algoritmo se baseia na combinação do método de Quase-Newton (em que a matriz hessiana  $H$  é substituída por uma aproximação de  $J^T(Transposto) * J$  onde  $J$  é o jacobiano da matriz da função) e do método do gradiente. As equações (1) e (2) descrevem essa relação. Em (2)  $I$  é a matriz identidade,  $\mu$  é um coeficiente e onde  $k$  denota a iteração.

$$x_{k+1} = x_k - [Hf(x_n)]^{-1} \nabla f(x_n) n > 0 \quad (1)$$

$$x_{k+1} = x_k - [J^T J + \mu * I]^{-1} J^T e \quad (2)$$

Um ponto importante a ser mencionado é que o LMA é um método de solução de sistemas não lineares, utilizado quando a calibração é a do modelo cinemático do robô, ou seja, é a calibração do robô via seu modelo cinemático. Seu resultado seria uma espécie de “equação de ajuste” ou correlação entre os pontos programados no controlador do robô e os pontos efetivamente atingidos por seu elemento terminal.

Já em métodos de calibração utilizando Redes Neurais Artificiais – RNA, não se envolve o modelo cinemático, pois a função obtida com o aprendizado da rede relaciona as posições de junta diretamente às posições do elemento terminal, sem necessidade de conhecimento do modelo. Isso ocorre através dos pesos  $w$ , existentes nos neurônios da rede que correlacionam de forma não linear as entradas do sistema com a saída desejada. A entrada do neurônio  $j$  é representada por  $x_j$  e pode ser qualquer informação, como as características de um produto. A saída  $k$  é descrita por  $y_k$  e representa uma das respostas, que pode ser passada a diante ou formar a saída geral. O peso do neurônio  $j$  é representado por  $w_j$ , a função  $g(u_k)$  descreve a relação matemática necessária para o ajuste e  $b$  é um valor de ajuste conhecido como *bias*. A equação (3) apresenta essa relação.

$$y_k = g(u_k) = g(\sum_{j=1}^n w_j x_j - b) \quad (3)$$

A diferença prática entre os dois métodos de calibração no sentido de sua viabilidade de aplicação é que as técnicas que envolvem o modelo cinemático exigem uma pequena quantidade de pontos de medição, enquanto que o procedimento baseado em Redes Neurais Artificiais exige uma imensa quantidade desses pontos para que haja convergência. Nesse último evento, se esses pontos tiverem de ser medidos por um sistema externo e lento, inclui-se aí mais um complicador que pode inviabilizar o uso da RNA.

Sabendo-se que nos métodos de calibração utilizando Redes Neurais Artificiais, a função obtida com o aprendizado da rede relaciona as posições de junta diretamente às posições do elemento terminal, torna-se necessário que se conheçam as posições desses pontos e que eles sejam lidos de forma rápida e precisa para que a rede neural convirja satisfatoriamente.

Importante lembrar que os métodos que utilizam redes neurais artificiais têm tido seu custo computacional diminuído com o passar do tempo, isso devido a melhoria do software e com a queda no preço do hardware. Sobretudo na medida em que o valor do processa-

mento computacional paralelo, força motriz das RNAs, tem caído devido ao barateamento da microeletrônica baseada em GPU – *Graphics Processing Unit*.

Assim seria de grande valia se um equipamento com uma IMU pudesse ser usado para medir a posição desses vários pontos de forma rápida e precisa como em uma navegação estimada (*Dead Reckoning*), o que não ocorreu até o momento por conta de uma característica intrínseca das IMUs: perda de sua precisão poucos segundos após a inicialização, inviabilizando a medição de posição no grau necessário à calibração de robôs [Vieler et al. 2016, Du e Zhang 2013]. Ocorre que, atualmente, os sistemas IMU's, quando usados para realizar a navegação estimada (*Dead Reckoning*) de veículos e outros dispositivos, funcionam apenas quando seu sinal atua em conjunto com outras tecnologias, como GPS (*Global Positioning System*), câmeras etc., que, mesmo assim, trabalham de forma subsidiária. Isso se deve ao fato de que as IMU's propagam ruídos que se acumulam, levando a uma perda progressiva de sua exatidão ao longo do processo, o que pode ser corrigido com a metodologia a ser proposta nesta pesquisa. Sabe-se que as causas dessa perda de exatidão se dão em decorrência de vários fatores que serão aprofundadas no capítulo dedicado exclusivamente às IMUs, mas pode-se mencionar que os principais problemas advêm dos seguintes fatos:

- a Os giroscópios do sistema IMU, cuja plataforma contém seus acelerômetros, sofrem variações intrínsecas em seus sinais que tendem a se traduzir por oscilações irreais em sua posição angular;
- b As medidas dos acelerômetros, que são muito sensíveis a ruídos mecânicos, captam vibrações em seu entorno que alteram a estatística dos valores de aceleração na IMU, mesmo quando permanecendo em repouso;
- c A mecanização, que é a integração matemática necessária para se calcular a velocidade e posição da IMU a partir da aceleração, adiciona ruído devido à propagação de erro das constantes de integração a cada operação matemática para obter posições e velocidades;

Por conta desses ruídos, os pesquisadores que usaram IMUs para calibração de robôs industriais restringiram sua pesquisa apenas no sentido de calibrar a orientação do robô, já que os sistemas de giroscópios, quando auxiliados por acelerômetros, resolvem apenas seus erros de inclinação de forma precisa [Vieler et al. 2016, Du e Zhang 2013], evitando medirem a posição, ou seja, de se aprofundarem no problema de como se usar a IMU para calcular a posição do elemento terminal do robô; A calibração da posição de robôs usando IMUs é justamente a contribuição original que esta pesquisa objetiva oferecer.

Como o que se deseja é calibrar a posição do robô ao invés da sua orientação, antes mesmo de calibrar o robô, dever-se-ia ou criar uma metodologia ou algoritmo que mitigasse o comportamento ruidoso da IMU, a fim de que a mesma fosse capaz de produzir medições mais exatas de posição ao longo do tempo. Assim, antes de calibrar o robô, a metodologia

aqui proposta tem como objetivo primeiro calibrar a IMU e só após usa-la para calibrar o robô (trabalhos futuros).

### 2.2.11 Unidades de Medição Inercial (IMUs)

IMUs desempenham um papel importante na navegação, estabilização e posicionamento de muitos sistemas mecânicos e veículos, abrangendo aplicações industriais, comerciais e militares [Ribeiro e Santos 2017]. Os IMUs foram inicialmente compostos por conjuntos de sensores inerciais muito caros, contendo giroscópios e acelerômetros, que fornecem medições brutas compensadas, tais como aceleração linear e orientação ao longo de três direções da estrutura de um corpo. Com o avanço tecnológico em Sistemas Micro-Eleto-Mecânicos - MEMS existe uma ampla gama de IMUs baratas no mercado [Talukdar et al. 2011, Dong et al. 2020], variando em preço e qualidade e voltadas para um número amplo de aplicações.

Esses sistemas MEMS também usam um algoritmo de mecanização inercial que é basicamente uma maneira de integrar matematicamente as medidas de aceleração do sensor para calcular sua velocidade e integrar novamente para calcular sua posição. No entanto, a mecanização leva a taxas de erro que precisam ser calibradas para prevenir ou atenuar possíveis falhas no sistema, além de outros erros possíveis.

O objetivo dessa seção é caracterizar diversos tipos de IMUs e seus devidos erros para auxiliar na escolha de um equipamento que garanta desempenho e preço necessários para subsidiar este trabalho.

### 2.2.12 IMUs: Sensores

Os IMUs são compostos por conjuntos de sensores inerciais, basicamente giroscópios e acelerômetros [Xu et al. 2016], que fornecem medições tais como orientação ao longo de três direções da estrutura de um corpo e aceleração linear. Estas medições depois de mecanizadas (matematicamente integradas) poderiam fornecer a velocidade e posição do corpo.

Seja  $s$  a posição de um corpo em um referencial,  $v$  sua velocidade linear e  $a$  sua aceleração, temos a descrição completa nas equações (4) e (5):

$$s(t) = \int v(t)dt \tag{4}$$

$$v(t) = \int a(t)dt \tag{5}$$

Assim, de posse dos valores de aceleração do corpo, fornecidos pela IMU, poder-se-ia integrá-los ao longo do tempo para obter sua velocidade atual e, da mesma forma, integrar

os valores desta velocidade para estimar a posição instantânea do corpo.

Importante ressaltar que as posições, velocidades e acelerações medidas pelo IMU tem como base o sistema de coordenadas de seu próprio corpo [Xu et al. 2016] que será chamado de  $b$ . A origem desse sistema está localizada no centro da tríade do acelerômetro e está alinhada ao invólucro que contém a IMU. Todas as medidas inerciais são resolvidas neste referencial.

O referencial de navegação  $n$  é um referencial geográfico local no qual queremos navegar. Em outras palavras, estamos interessados na posição e orientação do referencial  $b$  em relação a esse referencial  $n$ . Para a maioria das aplicações, esse referencial  $n$  é definido como estacionário em relação à terra. No entanto, nos casos em que se espera que o sensor se mova em grandes distâncias, é costume mover e girar o referencial  $n$  ao longo da superfície da Terra. Esses diferentes referenciais se relacionam de acordo com a matriz de rotação  $R$  entre os sistemas de coordenadas, tal relação é resumida pela equação (6):

$$x^n = R^{nb}x^b, x^b = (R^{nb})^T x^n = R^{bn}x^n \quad (6)$$

onde  $x^n$  é o referencial geográfico local,  $x^b$  é o referencial de interesse,  $R^{nb}$  é a matriz de rotação entre os sistemas de coordenadas desses referenciais e  $T$  é a transposta.

Lembrando que o erro do sistema é fator importante na qualidade da medição, é relevante ressaltar que dos primeiros IMUs aos modelos atuais percebe-se uma evolução em direção da miniaturização de seus componentes, assim como a fusão de outros sensores não inerciais em microprocessadores, que incorporados ao conjunto aumentam a capacidade e precisão das unidades de gerações anteriores. Além disso, os novos sistemas podem filtrar e calcular os valores desejados de maneira mais rápida e econômica.

### 2.2.13 IMUs: Giroscópio

Componente básico da IMU, o giroscópio é um equipamento usado para calcular orientação de um corpo em relação a um referencial e serve de plataforma para inserção dos acelerômetros contidos no sistema.

Como já mencionado anteriormente, as posições, velocidades e acelerações medidas pelo IMU tem como base o sistema de coordenadas de seu próprio corpo que será chamado de  $b$  e o referencial de navegação  $n$  é um referencial geográfico local no qual queremos navegar. O referencial inercial, chamado de  $i$ , é um referencial estacionário. A IMU mede aceleração linear e velocidade angular em relação a este referencial  $i$ . Sua origem está localizada no centro da Terra e seus eixos estão alinhados em relação às estrelas.

O referencial da Terra, chamado de  $e$ , coincide com seu eixo que tem origem no referencial

inercial  $i$ , mas gira com a terra. Isto é, tem sua origem no centro da Terra e eixos que são fixos em relação à Terra. Como o giroscópio mede a velocidade angular da estrutura do corpo da IMU em relação ao referencial inercial, expresso na estrutura do corpo, denotada por  $\omega_{ib}^b$ , e esses diferentes referenciais se relacionam de acordo com a matriz de rotação  $R$  entre os sistemas de coordenadas, esta velocidade angular pode ser expressa como na equação (7).

$$\omega_{ib}^b = R^{bn}(\omega_{ie}^n + \omega_{en}^n) + \omega_{nb}^b \quad (7)$$

onde a velocidade angular do referencial inercial do corpo é  $\omega_{ib}^b$ ,  $\omega_{ie}^n$  é a velocidade angular do referencial inercial  $i$  que gira com a terra,  $\omega_{en}^n$  é a velocidade angular do referencial da Terra e  $R^{bn}$  representa a matriz de rotação entre os sistemas de coordenadas.

O disco do giroscópio pode ser inserido em uma montagem que possibilite variação de sua orientação, como em um sistema de três eixos cardans (Gimbal - Inc.). Ao se girar o disco do giroscópio, a orientação do seu primeiro eixo (de rotação do disco) não é afetada pela inclinação ou rotação do restante da montagem, o que torna os giroscópios úteis para medir inclinação ou manter determinada orientação de uma plataforma. A Figura 2.6 apresenta um giroscópio e o conjunto de suas partes.

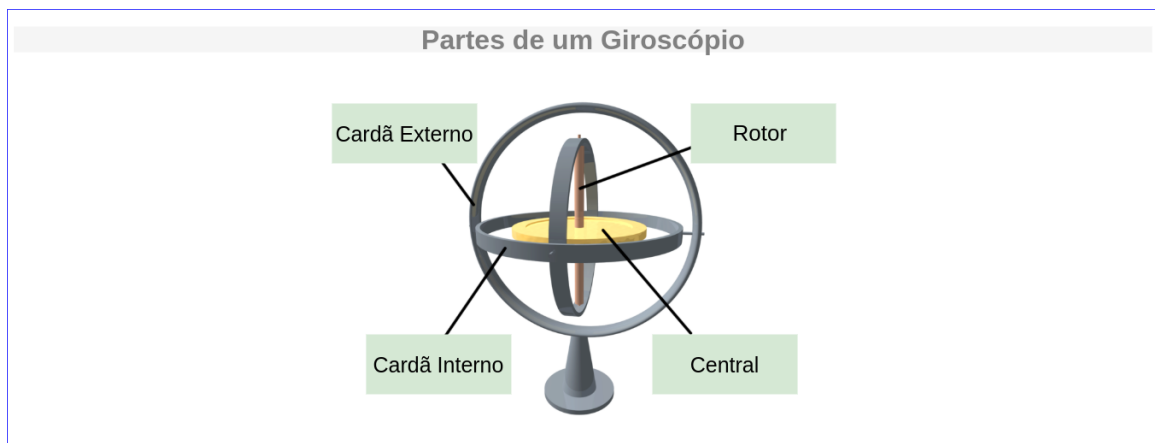


Figura 2.6 – Configuração interna de um giroscópio. Desenvolvida com base em figura obtida de *Creative Commons Attribution-Share Alike 3.0*.

O giroscópio mantém sempre a mesma direção na inexistência de forças que possam perturbá-lo. Dessa forma a massa giratória isolada tende a manter sua posição angular em relação a um referencial inercial [Passaro et al. 2017]. Quando ocorrem torques externos ou rotações em torno de um determinado eixo da montagem, sua orientação pode ser mantida e a medição da velocidade angular pode ser obtida devido ao fenômeno da precessão [Passaro et al. 2017]. Em um sistema rotacional, havendo torques externos líquidos e ortogonais ao vetor momento angular, o vetor do momento angular se moverá na direção do vetor de torque aplicado. Como resultado do torque aplicado, o eixo de rotação gira em torno de um eixo

perpendicular a si mesmo chamado eixo de saída. Nos sistemas de giroscópios automatizados, esta rotação em torno do eixo de saída é então detectada e alimentada de volta ao eixo de entrada, onde um motor ou dispositivo similar aplica torque na direção oposta, cancelando a precessão do giroscópio e mantendo sua orientação.

Os giroscópios mecânicos foram e ainda são amplamente usados em barcos e aviões, mas devido à existência de partes móveis, têm problemas de desgaste ao longo do tempo, precisão afetada pela variação de temperatura e não resistem bem a choques [Fraden 2004].

A geração seguinte de giroscópios se utilizou do efeito de Coriolis [North 2015]. Em um sistema rotativo, cada ponto gira com a mesma velocidade angular. À medida que um corpo, livre para se mover dentro desse sistema, se aproxima do eixo de rotação do sistema, sua velocidade angular permanece a mesma, mas a sua velocidade tangencial, que é perpendicular ao eixo de rotação, diminui. Em um referencial sujeito a velocidade angular uniforme, os corpos em movimento na direção do eixo de rotação, vistos por um observador nesse referencial, experimentam uma força inercial perpendicular à direção de seu movimento: a força de Coriolis. A força de Coriolis depende da velocidade angular do objeto rotativo, de sua massa e velocidade em relação ao eixo de rotação. Um corpo sujeito a força de Coriolis,  $F_c$ , tende a desacelerar ou acelerar dependendo do sentido em sua trajetória, segundo a equação (8), onde  $m$  denota a massa,  $V$  a velocidade linear e  $W$  a velocidade angular.

$$F_c = 2mvw \quad (8)$$

A partir desse princípio surgiram os *Coriolis Vibratory Gyroscopes* (CVG) que empregam um conjunto de extensômetros (*strain gauges* - transdutor apto a medir deformações mecânicas em um corpo de prova) e circuitos eletrônicos acoplados a um corpo [Lavrik e Datskos 2019]. Este conjunto é capaz de analisar as características vibracionais desse corpo, com geometria determinada e sujeito a uma frequência de oscilação previamente induzida. Sendo capaz de traduzir essas características de vibração em um valor de tensão elétrica inferindo sua orientação.

Um exemplo dessa tecnologia é o HRG (*Hemispherical Resonator Gyroscope*) [Taheri et al. 2014]. Consiste em um hemisfério oco feito do quartzo ( $\text{SiO}_2$ ) com propriedades piezoelétricas apoiado em uma haste do mesmo material induzido eletricamente a vibrar. Os seus pontos nodais (onde as deformações oriundas da vibração são máximas) são alocados como referencial e na medida em que o hemisfério sofre uma rotação em seu eixo os pontos nodais também giram, porém com um atraso em relação a seu ponto inicial, o que é proporcional à rotação aplicada [Verplaetse 1996]. A Figura 2.7 exemplifica o dispositivo. Um impacto no ressonador hemisférico causa vibração nos seus pontos nodais.

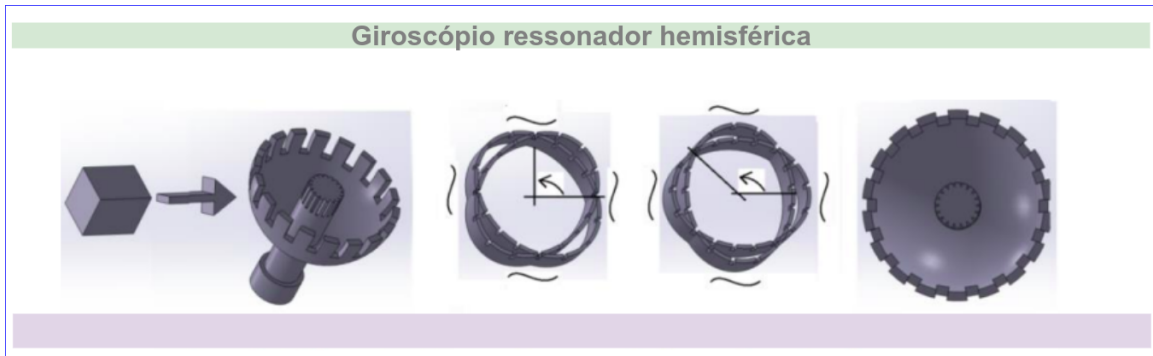


Figura 2.7 – Exemplo de giroscópio do tipo HRG. Fonte: Compilação através do arquivo pessoal do autor (software 3D).

Já outra geração de giroscópios o *Ring Laser* se baseia no princípio de Sagnac [Virgilio et al. 2019], tal aparato é um dos mais sensíveis de sua família. Uma fonte de luz coerente (Laser) é dividida em duas por um espelho que a reflete parcialmente. Os dois feixes Laser oriundos da divisão são forçados a percorrer um circuito (anel) em sentidos opostos.

Se o sistema permanecer estacionário, sem rotação no eixo do anel, ambos os feixes de luz irão gastar a mesma quantidade de tempo para percorrer o circuito e chegar de volta à fonte. No entanto, se o sistema girar ao longo do eixo do anel, é preciso um menor tempo para o feixe em sentido contrário à velocidade tangencial de rotação da estrutura completar seu caminho.

Quando o sistema está em rotação e os caminhos percorridos pelos dois feixes têm comprimentos diferentes, origina-se uma pequena disparidade em seu tempo de percurso que é proporcional a essa rotação. Este tempo pode ser medido por um interferômetro. Um exemplo desse conceito é demonstrado na Figura 2.8. Os espelhos são empregados para focalizar e redirecionar os raios laser nos cantos. Ao viajar entre espelhos, os feixes passam por tubos de gás.



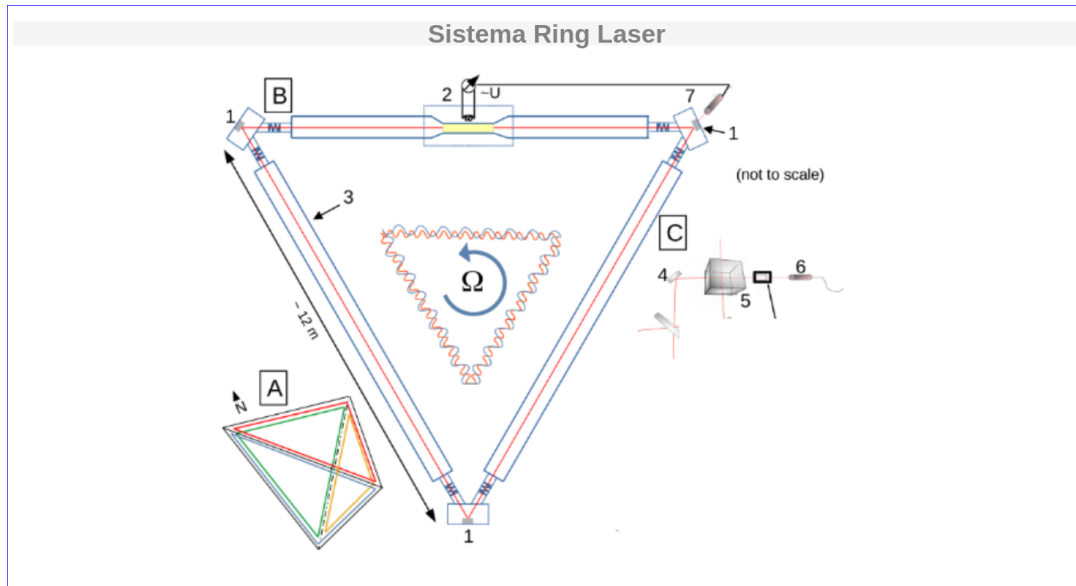


Figura 2.8 – Desenho esquemático em corte do projeto de um sistema Ring Laser em um triângulo equilátero. (1) O caminho do laser triangular é fechado usando espelhos altamente refletivos. (2) a descarga de gás é sustentada indutivamente com uma antena de rádio no tubo de ganho estreito. (3) cavidade ressonante evacuada preenchida com uma mistura de Hélio-Néon. (C) combinação dos dois feixes usando dois espelhos de deflexão [Brotzer et al. 2021]

Vale observar que este princípio pode ser aplicado para qualquer tipo de percurso fechado (anel) independente de sua forma ser circular, triangular, quadrada, etc.

Assim, outro exemplo utilizando desse mesmo princípio é o FOG (*Fiber Optics Gyroscope*) [Kurbatov, Kurbatov e Goryachkin 2019]. A diferença em relação ao *Ring Laser* é que ao invés de um único anel de *loop*, esta tecnologia utiliza uma bobina de fibra óptica (vários anéis), o que aumenta em muito o caminho percorrido pela luz, permitindo uma maior precisão na medição de rotação. A Figura 2.9 exemplifica esse tipo de giroscópio. Seu princípio de funcionamento é baseado na interferência da luz que foi passada através de uma bobina de fibra óptica a qual pode chegar a 5 km. Dois feixes de laser são injetados na mesma fibra óptica, porém em direções opostas.

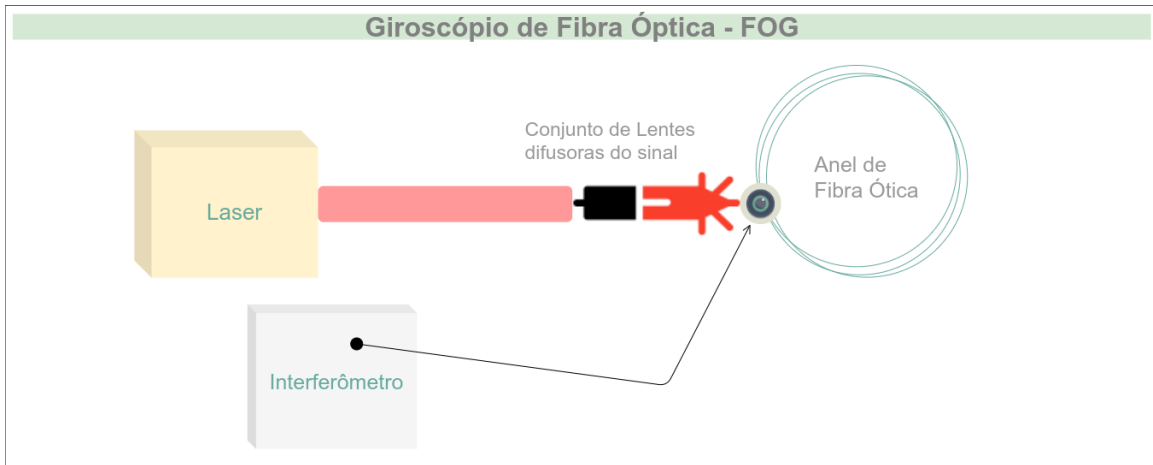


Figura 2.9 – Esquema construtivo do *Fiber Optics Gyroscope* (FOG). Fonte: O Autor.

Esta tecnologia oferece, além da alta precisão, ótima estabilidade térmica, resistência ao choque, pequeno consumo de energia, peso e volume. Também não possui partes móveis, o que leva a uma vida útil quase infinita. Sua desvantagem é que além de alto preço, os equipamentos com maior precisão são classificados como artigos militares e sujeitos a restrições de compra (ITAR – *International Traffic in Arms Regulations*).

Por fim, os giroscópios do tipo MEMs tem como princípio a vibração de um elemento piezoelétrico sujeito aos efeitos de Coriolis, porém com grande grau de miniaturização, já que são construídos na escala nanométrica dos circuitos integrados, oferecendo grande faixa de precisão e preços variados em função da tecnologia empregada. Estão disponíveis comercialmente, são acessíveis e de tamanho muito pequeno [Shah et al. 2016]. A Figura 2.10 descreve o conceito.

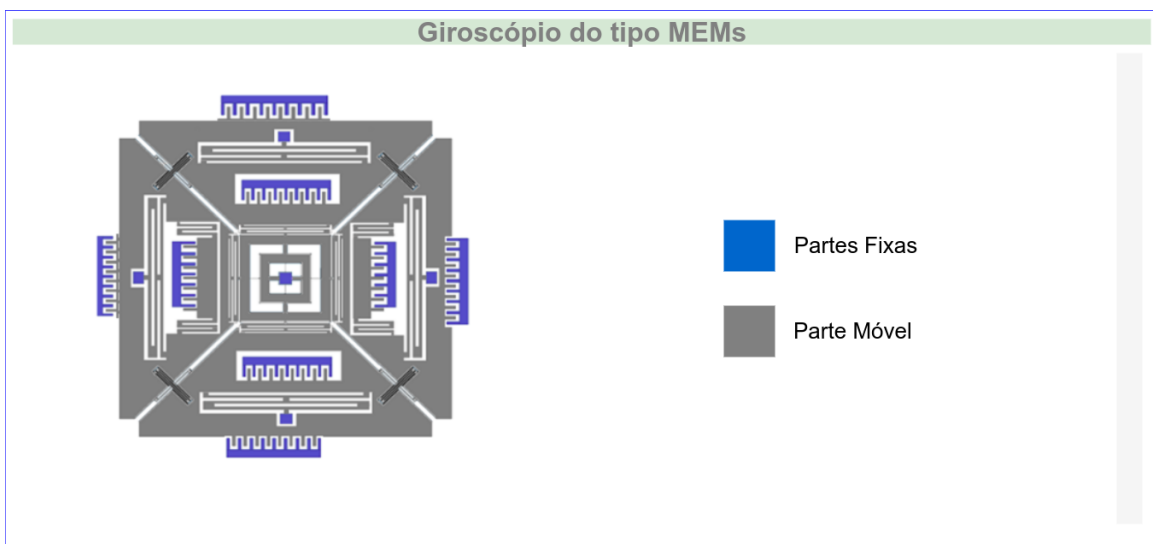


Figura 2.10 – Exemplo de um *design* de um giroscópio do tipo MEMs capacitivo, suas respectivas partes fixas e móveis. Fonte: [Shah et al. 2016].



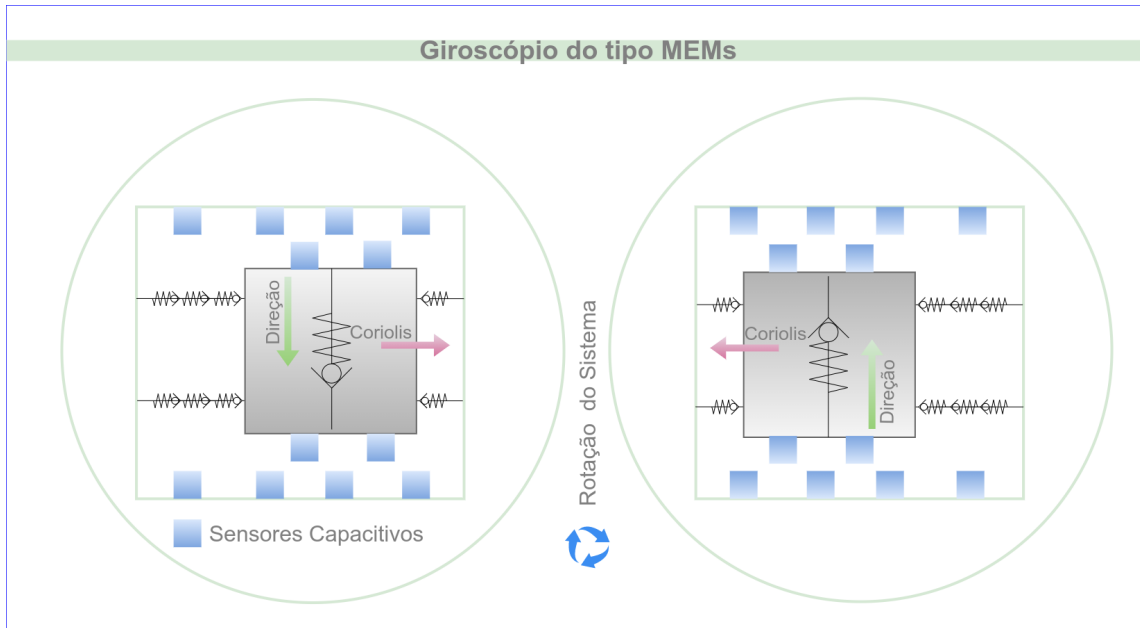


Figura 2.12 – Esquema de detecção da capacitância diferencial no giroscópio MEMS. Fonte: O Autor.

A conclusão que se chega em relação aos giroscópios é que independente do princípio físico, eles compartilham características básicas que os definem e que devem ser especificadas para cada caso de uso:

- a **Número de eixos de detecção ou Número de eixos sensores** – Existem giroscópios que medem a rotação angular em um, dois ou três eixos. Os giroscópios de detecção de múltiplos eixos têm múltiplos giroscópios de eixo único orientados ortogonalmente um para o outro [Baranov et al. 2019].
- b **Escala ou faixa de medição** – Este parâmetro determina a máxima velocidade angular com a qual o sensor pode medir [Baranov et al. 2019] e geralmente é dada em graus por segundo( $^{\circ}/\text{seg.}$ );
- c **Temperatura de trabalho** – As temperaturas de operação dos giroscópios variam de aproximadamente  $-40^{\circ}\text{C}$  a valores entre  $70$  e  $200^{\circ}\text{C}$  [Patel et al. 2009]. Muitos giroscópios detêm sensores de temperatura embutidos, o que mitiga a necessidade de se preocupar com questões de calibração relacionadas à temperatura;
- d **Não linearidade** – Os giroscópios produzem uma tensão proporcional à taxa angular detectada. A não linearidade é uma medida de quão próxima a linear essa tensão é proporcional à taxa angular real. A não linearidade é medida como um erro percentual de um ajuste linear em relação ao intervalo de escala total, ou um erro em partes por milhão (ppm) [BACURAU 2017];

- e **Largura de Banda (*bandwidth*)** – A largura de banda de um giroscópio mede o número de medidas possíveis por segundo. Assim, a largura de banda do giroscópio é geralmente dada em Hz [Si et al. 2013];
- f **Caminho Aleatório Angular (Angular Random Walk-ARW)** – Esta é uma medida de ruído do giroscópio com unidades de ( $^{\circ}/h$ ) ou ( $^{\circ}/seg.$ ) [Lam et al. 2003]. Pode-se pensar no ARW como a variação (ou desvio padrão), devido ao ruído, do resultado da integração da saída de um giro estacionário ao longo do tempo;
- g **Capacidades de sobrevivência de choque** - Em sistemas onde a aceleração linear e a taxa de rotação angular são medidas, é importante saber quanta força o giroscópio pode suportar antes de falhar [Yoon 2009]. Isso geralmente é medido em g (1g = aceleração devido à gravidade);
- h **Bias** – Este erro também conhecido como viés ou erro de polarização, é a saída de sinal do giroscópio quando este não está experimentando qualquer rotação [Kirkko-Jaakkola, Collin e Takala 2012]. A polarização pode ser expressa como uma tensão ou uma porcentagem da saída da escala total, mas essencialmente ela representa uma velocidade rotacional (em graus por segundo). O erro de polarização tende a variar, tanto com a temperatura quanto com o tempo. O erro de polarização de um giroscópio se deve a uma série de componentes : erros de calibração, desligar e ligar, efeitos de choque (nível g);
- i **Bias Drift (deriva)** - Isso se refere especificamente à variação do viés ao longo do tempo, assumindo que todos os outros fatores permanecem constantes. Basicamente, este é um efeito de aquecimento, causado pelo auto aquecimento do giroscópio e seus componentes mecânicos e elétricos associados [Wang e Han 2020]. Espera-se que esse efeito seja mais prevalente nos primeiros segundos após a ativação e seja quase inexistente após cerca de cinco minutos;
- j **Instabilidades de polarização (*Bias Instability*)**- A instabilidade é uma medida fundamental da qualidade de um giroscópio. É definido como o ponto mínimo na curva Allan Variance [Hiller et al. 2019], geralmente medido em ( $^{\circ}/h$ ). Ele representa a melhor estabilidade de polarização que poderia ser alcançada para um determinado giroscópio, assumindo que a média de polarização ocorre no intervalo mínimo;
- k **Ponto Flutuante da medida**- Este fator (que em geral não é comentado nos *papers* sobre o assunto, mas vem explicitado no *datasheet* dos IMUs) é de suma importância, na medida em que hoje os giroscópios que vêm embutidos nos sistemas inerciais passam seu sinal por um filtro digital que envia seus dados em forma de bits para a leitura. Dependendo da taxa de envio (*baud rate*) e do tipo de interface digital que o sistema disponibiliza (4, 8, 16 bits etc.) os erros de ponto flutuante se tornam maiores ou me-

nores por conta de arredondamentos ou truncamentos dos valores menos significativos (**Less Significant Bit** – LSB) dos registradores.

Os giroscópios podem ser classificados com base na estabilidade de polarização conforme a Tabela 2.3 :

Tabela 2.3 – Classificação dos giroscópios quanto ao *Bias Instability* [KVH 2016].

Grau de Desempenho	Estabilidade de polarização
Consumidor	30-1000° / h
Industrial	1-30° / h
Tático	0,1-30° / h
Tático - High End	0,1° / h
Navegação	0,01-0,1° / h
Estratégico	0,0001-0,01° / h

Retornando ao sistema de equações, é possível agora entrar em mais detalhes. Analogamente às equações (4) e (5), onde as acelerações são integradas para que sejam obtidas as posições e velocidades lineares, tem-se posições e velocidades angulares, onde  $\theta$  é a variação do deslocamento angular,  $\omega$  é a velocidade angular e  $a$  é a aceleração:

$$\theta = \frac{\partial \theta}{\partial t} = \omega \Leftrightarrow \int \omega dt = \theta \quad (9)$$

$$\omega = \frac{\partial \omega}{\partial t} = a \Leftrightarrow \int a dt = \omega \quad (10)$$

Importante mostrar que:

$$\int \omega dt = \theta + K\theta \quad (11)$$

$$\int a dt = \omega + K\omega \quad (12)$$

então:

$$\omega = \left( \int a dt \right) - K\omega \quad (13)$$

$$\theta = \int \left( \left( \int a dt \right) - K\omega \right) dt - Ks \quad (14)$$

onde os valores de  $K$  são as constantes que desapareceram na derivada das funções e agora reaparecem em forma de erro acumulado. Por exemplo, considerando um giroscópio com um ARW de ( $1^\circ/\text{seg.}$ ) sendo integrado muitas vezes para obter uma medida de posição angular, se o giroscópio estiver estacionário, o resultado ideal – e também o resultado médio – seria zero. Porém, quanto maior o tempo de integração, maior será a dispersão do resultado ideal para mais distante do zero, sendo essa distância proporcional à raiz quadrada do tempo de integração.

#### 2.2.14 IMUs: Acelerômetros

Os acelerômetros são parte integrante dos sistemas de navegação inercial e como o nome diz medem a aceleração de um corpo, de forma indireta, medindo a força específica  $f$  no referencial do corpo  $b$  da IMU (descrito anteriormente).

Da mesma forma que os giroscópios, evoluíram em termos de preço, consumo de energia, resistência, etc. devido às características de miniaturização obtidas usando MEMs, também os acelerômetros tiveram resultados semelhantes [Gonseth et al. 2015].

Em seu funcionamento, várias configurações são possíveis, mas todas se baseiam na força que uma massa de prova exerce contra a estrutura interna do conjunto mecânico que compõe o acelerômetro. Este sistema pode ser simplificado fazendo-se analogia a uma caixa em forma de cubo com uma bola dentro como na Figura 2.13.

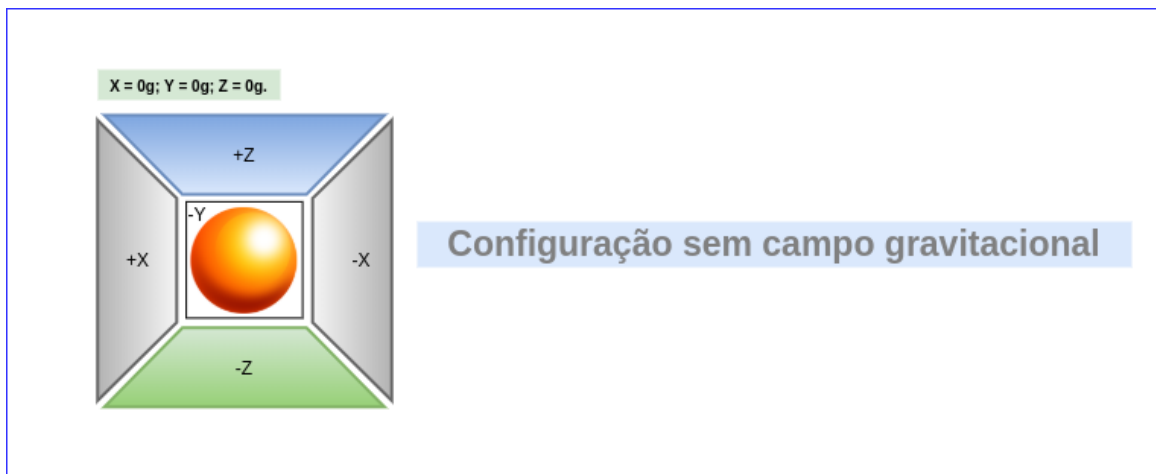


Figura 2.13 – Esquemático de um acelerômetro em repouso. Fonte: O Autor.

Se esta caixa estiver em um local sem campos gravitacionais ou outros campos quaisquer que possam afetar a posição da bola, ela simplesmente flutua no meio da caixa. Na Figura 2.13, a cada eixo é atribuído um par de paredes em que cada parede é sensível à pressão.

Se de repente a caixa for movida para a esquerda (acelerando com a aceleração constante

de  $1g = 9.8m/s^2$ ), a bola atingirá a parede  $-X$ . Medindo a força de pressão (força resultante) que a bola aplica à parede produz-se um valor de  $-1g$  no eixo  $X$ , como na Figura 2.14. A esfera tende a ir na direção da inércia representada pelo vetor na imagem. Dessa forma o acelerômetro percebe a força na parede  $-X$ . A equação (15) descreve essa relação, onde  $\sum F_R$  é a força resultante do sistema de forças,  $m$  é a massa e  $\ddot{x}$  é a aceleração no eixo  $x$ .

$$\sum F_R = m * \ddot{x} \quad \text{onde} \quad \ddot{x} = a \quad (15)$$

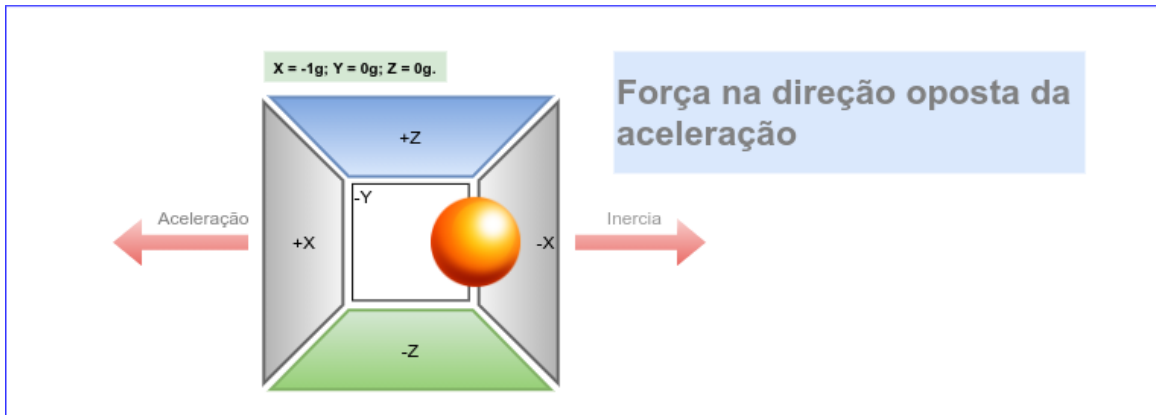


Figura 2.14 – Esquema de um acelerômetro em movimento. Fonte: O Autor.

Neste caso o acelerômetro detecta realmente uma força na direção oposta do vetor de aceleração. Esta força é uma Força Inercial que possibilita ao acelerômetro medir a aceleração indiretamente através da pressão aplicada a uma das suas paredes. Se o modelo estiver sujeito ao campo gravitacional da Terra, a bola cairá na parede  $Z$  e aplicará uma força resultante de uma aceleração de  $1g$  na parede inferior, conforme mostrado na Figura 2.15 e especificado na equação (16):

$$\sum F_R = m\ddot{x} \quad \text{onde} \quad \ddot{x} = g \quad (16)$$



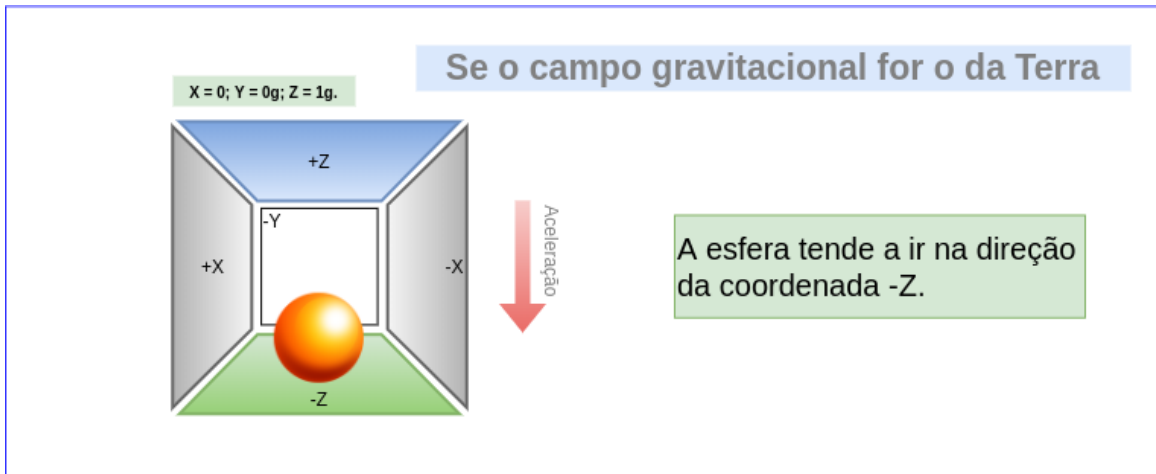


Figura 2.15 – Esquema do acelerômetro desalinhado. Fonte: O Autor.

A caixa não está em movimento, mas ainda existe uma leitura de  $-1g$  no eixo  $Z$ . A pressão que a bola aplicou na parede foi causada por uma força gravitacional.

Assim, as medidas do acelerômetro não implicam necessariamente em aceleração, mas sim na força de inércia gravitacional capturada pelo mecanismo de detecção de força do acelerômetro. Desse modo, quando um acelerômetro está desalinhado em relação ao vetor gravidade, cujo referencial é o da Terra, ele pode mostrar falsas leituras de aceleração, como apresentado na Figura 2.16. Considerando que a esfera esteja sujeita a forças somente no eixo  $X$  e  $Z$  temos como produto o resultado da soma vetorial.

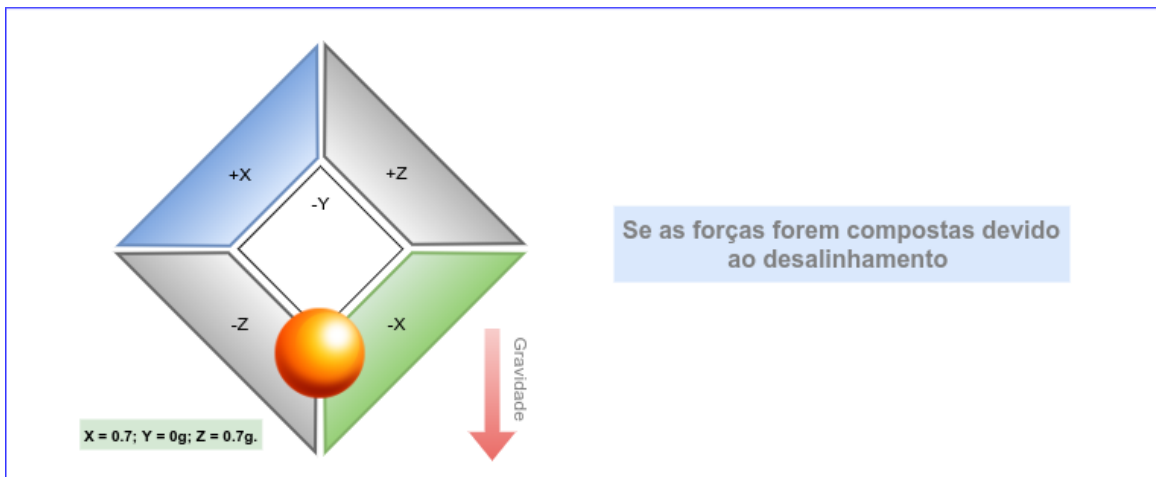


Figura 2.16 – Esquema de acelerômetro em posição inclinada. Fonte: O Autor.

Em termos do referencial de medida, sabendo que os acelerômetros medem a força específica  $F^b$  no referencial do corpo  $b$  da IMU. Ou seja:

$$F^b = R^{nb}(a_{ii}^n - g) \quad (17)$$

onde  $g$  denota o vetor gravidade e  $a_{ii}^n$  denota a aceleração linear do sensor no referencial de navegação que se dá por:

$$a_{ii}^n = R^{nc}R^{ci}a_{ii}^c \quad (18)$$

onde  $R^{nc}$  é a matriz de rotação entre os referenciais  $n$  e  $c$ ,  $R^{ci}$  é a matriz de rotação entre os referenciais  $c$  e  $i$ . Os subscritos na aceleração linear  $a$  são usados para indicar em qual referencial a diferenciação é executada. Onde  $a_{nn}^n$  é a aceleração no referencial  $n$ .

Para fins de navegação, estamos interessados na posição do sensor no referencial de navegação  $p^n$  e suas derivadas conforme executado no referencial de navegação conforme segue a fórmula (20) (usando a notação de Leibniz - derivada no ponto  $n$ ):

$$\frac{d}{dt}p^n|n = v_n^n, \quad \frac{d}{dt}v^n|n = a_{nn}^n \quad (19)$$

Uma relação entre  $a_{ii}$  e  $a_{nn}$  pode ser derivada usando a relação entre dois referenciais coordenados rotativos. Dado um vetor  $x$  em um referencial de coordenadas  $u$ , conforme a igualdade (20):

$$\frac{d}{dt}x^u|u = \frac{d}{dt}R^{uv}x^v|v = R^{uv}\frac{d}{dt}x^v|v + w_{uv}^u x^u \quad (20)$$

onde  $w_{uv}^u$  é a velocidade angular do referencial  $v$  em relação ao referencial  $u$ , expresso no referencial  $u$ . O referencial de navegação  $p^i$  pode ser descrito como segue em (21):

$$p^i = R^{ic}p^c \quad (21)$$

Dessa forma a velocidade em  $i$  é expressa como em (22):

$$v_i^i = \frac{d}{dt}p^i|i = \frac{d}{dt}R^{ic}p^c|i = R^{ic}\frac{d}{dt}p^c|c + w_{ic}^i p^i = v_c^i + w_{ic}^i p^i \quad (22)$$

E de maneira semelhante pode-se definir a aceleração em  $i$  conforme a equação (23):

$$a_i^i = \frac{d}{dt}v_i^i|i = \frac{d}{dt}v_c^i|i + \frac{d}{dt}w_{ic}^i p^i|i = a_{cc}^i + 2w_{ic}^i v_c^i + w_{ic}^i w_{ic}^i p^i \quad (23)$$

Sabendo que a velocidade angular da Terra é constante, denotando  $n_{nc}^c$  como a distância entre o referencial da Terra e o referencial de navegação e usando a relação entre o referencial

da Terra e o referencial de navegação, temos a relação (24):

$$p^c = R^{cn} p^n + n_{nc}^c \quad (24)$$

Assumindo que o eixo de navegação é fixo no eixo da Terra onde  $R^{cn}$  e  $n_{nc}^c$  são constantes e como o sensor não vai viajar distâncias significativas comparadas com o tamanho da Terra, já que estará preso no robô chega-se a fórmula (25), onde os subscritos são usados para indicar em qual referencial a grandeza é calculada no momento da diferenciação, expresso no referencial  $n$ :

$$a_{ij}^n = a_{nn}^n + 2w_{ic}^n v_n^n + w_{ic}^n w_{ic}^n p^n \quad (25)$$

onde  $a_{ij}^n$  é a aceleração do referencial  $i$  em relação  $j$ , expressa no referencial  $n$  e  $w_{ic}^n$  é a velocidade angular do referencial  $i$  em relação ao referencial  $c$ , expresso no referencial  $n$ .

Assim, para que os referenciais de medição não venham a inserir erros no processo de coleta de dados, o primeiro passo no uso de um dispositivo IMU que combina um acelerômetro e um giroscópio é alinhar seus sistemas de coordenadas. A maneira mais fácil de fazê-lo é escolher o sistema de coordenadas do acelerômetro como seu sistema de coordenadas de referência.

A maioria dos manuais de instrução de acelerômetros exibe a direção dos eixos  $X$ ,  $Y$ ,  $Z$  em relação à imagem do microcircuito ou dispositivo físico. Desta forma, o dispositivo deve ser alinhado na posição horizontal e as saídas  $X$  e  $Y$  do acelerômetro emitiriam a tensão zero-g. Em seguida, o dispositivo deve ser rotacionado em torno do eixo  $Y$ , ou seja, girado no plano  $XZ$ , de modo que as saídas do acelerômetro  $X$  e  $Z$  mudam e a saída  $Y$  permanece constante. O passo final é garantir que o sentido de rotação corresponda ao modelo.

É necessário mencionar que mesmo que o acelerômetro esteja em um estado relativamente estável, ainda é muito sensível à vibração e ao ruído mecânico em geral [Quadri e Sidek 2014]. Esta é a principal razão pela qual os sistemas IMU usam um giroscópio para suavizar os erros do acelerômetro. O giroscópio não está livre de ruído, no entanto, porque mede a rotação, é menos sensível aos movimentos mecânicos lineares (o tipo de ruído que o acelerômetro sofre), mesmo assim, os giroscópios têm outros tipos de problemas como, por exemplo, a já citada deriva (não voltando ao valor da taxa zero, quando a rotação para) [Lee et al. 2016, Li et al. 2019].

Portanto, é necessário que se use a média dos dados que vêm do acelerômetro e do giroscópio, no sentido de se obter uma estimativa relativamente melhor da inclinação, o que ocorre de fato, desde que estas configurações iniciais sejam feitas de forma correta. Nos sistemas modernos estas configurações básicas já vêm prontas de fábrica e possuem também outros sensores [Patonis et al. 2018] auxiliares como magnetômetros, barômetros, etc., além de ter um microprocessador matemático que faz os cálculos necessários à fusão dos sensores,

bastando ao usuário calibrar o sistema em relação ao referencial que vai usar.

### 2.2.15 IMUs: MEMs

A tecnologia MEMs traz ao uso cotidiano equipamentos menores, mais rápidos e mais baratos. Essa tecnologia se origina dos processos de fabricação de circuitos integrados. Pode detectar ou atuar em microescala, funcionar individualmente ou em combinação com outros dispositivos para gerar efeitos de conjunto macro [Judy 2001].

Além das vantagens mencionadas, a fabricação destes dispositivos oferece redução de custos devido a técnicas de processamento em lote (similares à fabricação de circuitos integrados semicondutores – IC). De uma forma geral o processo de fabricação das MEMs para construção de IMUs pode ser esquematizado de acordo com a Figura 2.17:

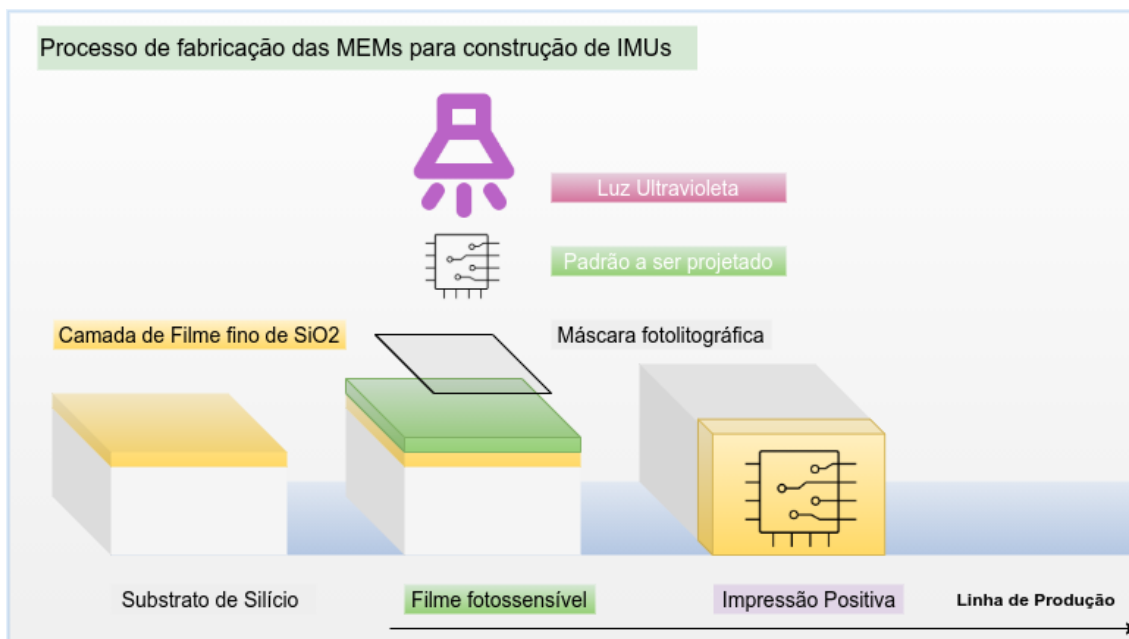


Figura 2.17 – Esquema do processo de fabricação em lote. Fonte: O Autor.

O processo passa por várias sequências de deposição, queima por raios ultravioleta e ataques químicos que podem se repetir várias vezes, para cada camada a ser fabricada [Ho 2014]. O método básico se inicia com um substrato de silício (como nos circuitos integrados) sendo a ele aplicada uma camada de filme fino em sua superfície. Note que existem vários meios possíveis para criar esta camada como dopagem, oxidação, deposição de plasma, etc. e vários materiais para formar o filme como  $SiO_2$ ,  $Si_3N_4$ ,  $Cu$ , etc.

Em seguida é assentada sobre a superfície de filme fino uma camada de material fotossensível, que reage à luz ultravioleta dependendo de sua composição. De forma simplificada, a reação pode ou fortalecer a camada exposta, deixando-a mais resistente ao ataque químico

ou enfraquecê-la, tornando-a suscetível ao ataque que a remove e expõem parte do filme fino. Sobre a camada de material fotossensível é inserida uma máscara translúcida, cuja geometria deixa passar luz apenas em certos locais e bloqueia sua passagem em outros (fotolitografia). Dessa forma a luz ultravioleta que for projetada através da passagem da máscara transferirá a imagem do desenho (*pattern*) feito na máscara para o elemento fotossensível.

Com a máscara inserida, o sistema recebe aplicação de luz ultravioleta [Ho 2014], cuja imagem projetada através da mesma queima o material fotossensível na forma desejada, expondo a parte do filme fino inicialmente contido no substrato de silício para posterior ataque químico. Há dois tipos possíveis de material fotossensível: o que vai ser fortalecido com a luz ultravioleta formando uma camada de proteção ao filme fino ou o que vai ser enfraquecido pela luz ultravioleta e removido quimicamente. O material fotossensível extraído expõe a parte do filme fino, que fica desprotegido. A estrutura é submetida ao ataque químico de um agente que corrói o material ao qual entra em contato (*etching*).

Como a parte exposta do filme fino se depara agora com o agente químico, ela acaba corroída e desaparece, formando uma geometria derivada da imagem projetada na máscara. Esta imagem forma o padrão que inicialmente estava disposto na máscara fotolitográfica.

### 2.2.16 IMUs: Erros

Erros em IMUs podem ser causados por muitos fatores, incluindo a instabilidade no viés de giroscópio, a incerteza do fator de escala do acelerômetro, mecanização no cálculo das integrais, não linearidades e outros parâmetros [Du et al. 2016]. Estudos mostram que devido às instabilidades, quanto mais tempo um giroscópio opera, maior é a taxa acumulada ou o erro de cálculo da posição Angular Random Walk – ARW, e uma vez que estes ruídos têm componente de alta frequência (curto prazo) e componente de baixa frequência (em longo prazo), se não forem minimizados podem diminuir a precisão do sistema [Zhou et al. 2017, Du et al. 2016].

Pesquisadores e fabricantes tem concluído que agregar outros dispositivos ao sinal da IMU melhora seu desempenho. Ou seja, conjuntos de magnetômetros, termômetros, barômetros e outros dispositivos estão sendo integrados na mesma placa de circuito que contém as IMUs baseadas em MEMS, para dar precisão adicional a todo o sistema [Du et al. 2016].

As medidas independentes de cada dispositivo são enviadas para um microprocessador que calcula, integra e faz a fusão dos dados de cada sensor para compensar seus deslocamentos específicos antes da mecanização. Além disso, o processo para a identificação e modelagem do erro estocástico de deriva está por vezes disponível e incorporado no filtro de hardware do sistema.

Apesar deste avanço, as IMUs que dependem de MEMS, quando usado para fins de

navegação ao ar livre, empregam um sistema auxiliar como GPS ou GNSS (sistemas de posicionamento por satélites com cobertura mundial) para rastrear sua posição real [Godha e Cannon 2007]. No campo de calibração robótica os MEMS são usados basicamente para calibração da orientação do robô e não o seu posicionamento e velocidade [Vieler et al. 2016].

De forma geral, antes de se iniciar a operação da IMU é sempre necessário que a orientação da mesma seja calibrada segundo as orientações de seu fabricante. Deve-se orientar um dos eixos da IMU, o eixo  $z$ , alinhado com o vetor gravidade da Terra e em certos casos pode-se alinhar um dos eixos perpendiculares ao da gravidade com o campo magnético da mesma. Dessa forma os sinais obtidos a partir do acelerômetro, giroscópio e magnetômetro são fundidos para obter a orientação do IMU, representada por uma Matriz de Cossenos Diretores DCM – também chamada de matriz de rotação, ou de um vetor de Quaternion. As equações (26), (27), (28), (29) e (30) exemplificam esse processo, onde  $\vec{G}_{(z)}$  é a gravidade em  $z$ ,  $\vec{M}$  é o campo magnético da terra, o cálculo do DCM é descrito com os seguintes passos:

$$\vec{G}_{(z)} \approx [0 \ 0 \ 9.8] \quad (26)$$

$$3DMag = \vec{M} = [M_{(x)} \ M_{(y)} \ M_{(z)}] \quad (27)$$

$$\vec{U}_{(x)} = |\vec{M} \times \vec{G}_{(z)}| \quad (28)$$

$$\vec{V}_{(y)} = |\vec{G}_{(z)} \times \vec{U}_{(x)}| \quad (29)$$

$$DCM = \begin{bmatrix} \vec{U}_{(x)} & \vec{V}_{(y)} & \vec{G}_{(z)} \end{bmatrix} = \begin{bmatrix} \hat{u}_x & \hat{v}_x & \hat{g}_x \\ \hat{u}_y & \hat{v}_y & \hat{g}_y \\ \hat{u}_z & \hat{v}_z & \hat{g}_z \end{bmatrix} \quad (30)$$

onde  $\vec{U}_{(x)}$  é o vetor resultante no eixo  $x$  do modulo do produto vetorial entre  $\vec{M}$  e  $\vec{G}_{(z)}$ ,  $\vec{V}_{(y)}$  é o vetor resultante no eixo  $y$  do modulo do produto vetorial entre  $\vec{G}_{(z)}$  e  $\vec{U}_{(x)}$  e, finalmente, a matriz do DCM é montada com base em  $\vec{U}_{(x)}$ ,  $\vec{V}_{(y)}$ ,  $\vec{G}_{(z)}$ .

O DCM expressa a orientação do IMU em relação a um sistema de coordenadas fixo, no qual a direção do eixo  $x$  aponta para o norte magnético e o eixo  $z$  apontando na mesma direção que a força da gravidade da Terra. O processo pelo qual a orientação da IMU pode ser estimada pode ser dividido em duas etapas principais:

1. Estimativa da orientação inicial através das medições do acelerômetro 3D e do magnetômetro 3D. Esse processo é feito apenas uma vez, quando o IMU está parado e antes

que o usuário possa iniciar qualquer movimento. Nas etapas de orientação inicial da IMU, alinha-se o eixo  $z$  com o vetor gravidade  $[0 \ 0 \ 9.8]$  e o magnetômetro com o campo magnético da terra  $[M_{(x)} \ M_{(y)} \ M_{(z)}]$ . A Figura 2.18 apresenta o diagrama de eventos da primeira etapa;

2. O segundo passo é estimar a orientação da IMU quando o movimento começa. O sensor de giroscópio 3D está envolvido neste processo. Através da integração de sinais de velocidades angulares obtidos a partir do giro 3D, é possível conhecer o ângulo de rotação de cada eixo do IMU. O DCM anterior é atualizado adicionando a nova rotação calculada. Na segunda etapa de orientação da IMU, gira-se todo o sistema IMU em dois eixos até que o sistema estabilize o giroscópio. O processo é resumido nas equações (31) e (32), onde  $\vec{\omega}$  é o vetor da velocidade angular que pode ser decomposto nos três eixos.

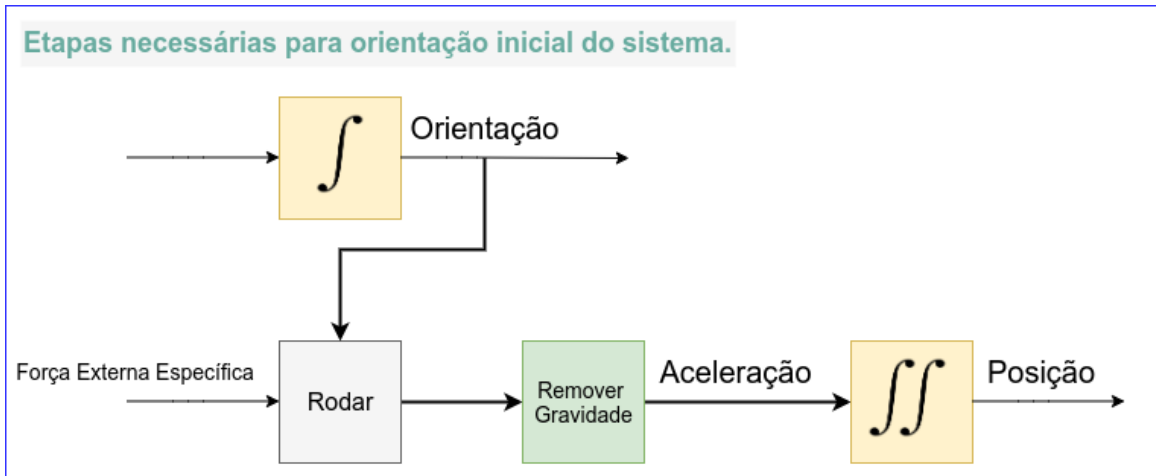


Figura 2.18 – Etapa inicial de orientação da IMU. Fonte: O Autor.

$$\vec{\omega} = [\vec{\omega}_X \ \vec{\omega}_Y \ \vec{\omega}_Z] \quad (31)$$

$$DCM_{(t)} = DCM_{(t-1)} + \int w(t)dt \quad (32)$$

Essas duas etapas são usadas em um processo de fusão dos sensores do giroscópio e acelerômetro para estimar a orientação a partir dos dados inerciais. Em alguns modelos de IMU um Filtro de Kalman (KF) [Hosseinyalamdary 2018], calculado dentro do processador no IMU, executa esse processo e envia um sinal mostrando que o sistema está pronto para operação.

É preciso salientar que a calibração de IMUs sugerida pelo fabricante se restringe apenas a sua orientação no espaço, pois à medida que seu sinal se deteriora rapidamente não há muito

que se falar em posição em relação a um referencial. Mesmo assim, alguns estudos foram feitos ao longo dos anos, no sentido de se obter valores de posição mais precisos a partir da medição de IMUs, sendo que na totalidade dos casos os estudos se baseavam ou na fusão com sinais de GPS e outros sensores ou no reinício de seu referencial [Zhang et al. 2017].

Quando se busca a fusão de sensores para melhorar a precisão de posicionamento o que se faz é acoplar sinais de baixa frequência como GPS, rádios, etc. aos sinais da IMU (alta frequência) onde o GPS corrige a posição da IMU fazendo com que a mesma se comporte como um indicador de direção e altitude, mostrando apenas a orientação em que o referencial navega. Isso ocorre em *drones*, veículos terrestres não tripulados, etc. e não será foco de uma análise mais detalhada, pois não há como ser praticado em locais fechados, caso de uma linha de robôs industriais pretendida neste trabalho.

Já no caso em que se usa o reinício de referencial, um algoritmo de parada é implementado e o referencial é reiniciado a partir de condições previamente determinadas, ou seja, sabendo-se que o sinal do sistema IMU se deteriora com valores crescentes em função do tempo, contado do momento do início da medição. Uma possibilidade interessante para mitigar essa deterioração seria reiniciar a medição em um intervalo que fosse suficientemente pequeno que impedisse seu incremento a valores acima do admitido.

Uma ideia nesse sentido, porém com pouca precisão, foi testada por pesquisadores que tinham como objetivo traçar a rota de pedestres ou tropas em movimento dentro de edifícios ou lugares sem sinais auxiliares de rádio, GPS, etc. usando apenas IMUs para calcular sua posição [Zhang et al. 2017]. A ideia foi colocar o sistema nos sapatos de pedestres voluntários, e empregar um algoritmo que zerava a posição  $Z$  – altura do sapato do pedestre em relação ao piso conforme a Figura 2.19, toda vez que o acelerômetro detectava um impacto do pé contra o solo.

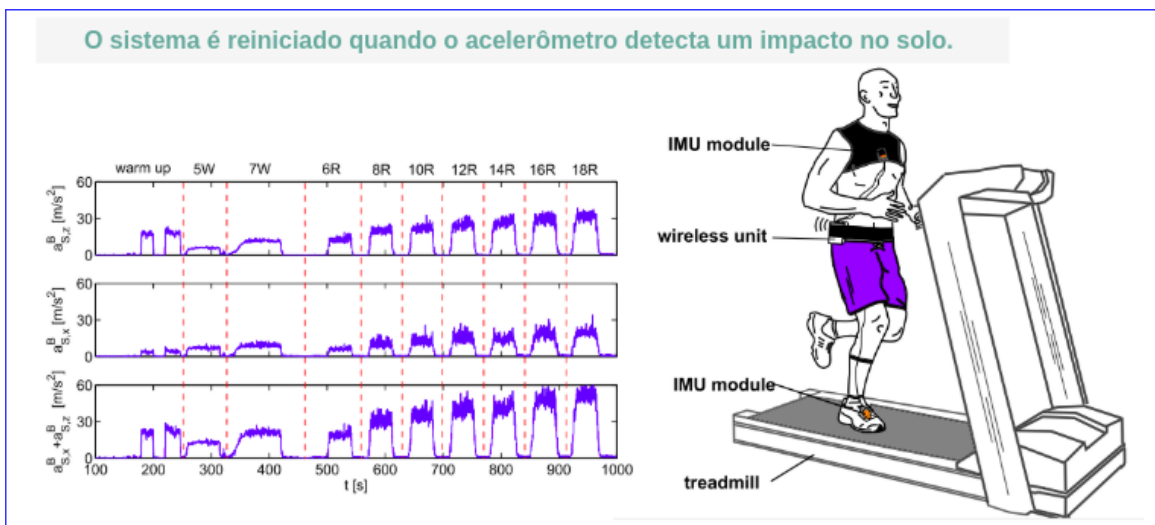


Figura 2.19 – Pontos possíveis para reinicialização de uma IMU. A aceleração máxima muda a cada passo. Fonte: [Zhang et al. 2017]



Desta forma, a cada toque no pavimento, poder-se-ia reiniciar a medição zerando o eixo  $z$ , e estimar o comprimento médio do passo do voluntário no experimento, desde que este se mantivesse em regime permanente, inferindo a próxima posição do pé contendo a IMU [Zhang et al. 2017]. Esse experimento, ao calcular o comprimento médio do passo do voluntário, consegue suavizar o progressivo aumento do erro, mas ainda carece em muito de exatidão, já que a precisão se encontra na faixa de metros ou dezenas de metros e não se avaliam os valores nos outros eixos, preenchendo-os com as estimativas médias de comprimento dos passos.

Foi a partir da evolução dessa ideia que se pensou na possibilidade de criar um método que pudesse ser usado na calibração de robôs usando IMUs, contando, porém com um artifício capaz de compensar falta de precisão aqui encontrada ao mesmo tempo em que mantém o baixo custo de operação de calibração: o uso de Redes Neurais Artificiais para obter os erros estatísticos e os pontos ideais de reinício do sistema.

### 2.2.17 IMUs: Conclusão e escolha de um modelo

Em que pese o assunto IMU ser vasto e apresentar enormes variações em termos de tecnologia empregada, características possíveis e preço, este bem sucinto resumo já é mais que suficiente para embasar e indicar a escolha de um modelo de IMU a ser usado na pesquisa sobre calibração. A Tabela 2.4 resume uma pesquisa de mercado feita pelo autor.

Tabela 2.4 – Exemplos de Marcas, características e preços de IMUs comerciais.

Marca	Acc.FS. g	Acc.Noise mg/ $\sqrt{Hz}$	Gyr.FS. /s	Gyr.Noise /s/ $\sqrt{Hz}$	Mag.FS. G	Mag.Noise mG/ $\sqrt{Hz}$	Preço US\$
Xsens Mti-10	5	0.080	450	0.030	0.8	0.2	800
Xsens Mti-G 710	5	0.080	450	0.010	0.8	0.2	3980
Invensense MPU-9150	16	0.4	2000	0.05	12	-	5
VectorNav VN-100	16	0.140	2000	0.003	2.5	0.14	800
Bosch	16	0.150	2000	0.014	13	0.3	70
KVH DSP3000 (FOG)	-	-	375	0.001	-	-	4200
FS= sensibilidade Ex: Gyr.FS=400 consegue ler entre 0 e 400 graus por segundo							

Dos modelos existentes no mercado (que são muitos e não estão presentes na Tabela 2.4) optou-se por utilizar o BNO055 da Bosh Sensortec GmbH, por ser acessível (ITAR Free), barato e ter um algoritmo de fusão de sensores que permite configurar quais sensores fundir individualmente, conforme Tabela 2.5 feita a partir do *datasheet* [BOSH 2021].

Tabela 2.5 – Configurações de fusão do BNO055 - baseada no manual.

Modo de Operação	Sensor Disponível			Fusão de Dados	
	Acelerômetro	Magnetômetro	Giroscópio	Orientação Relativa	Orientação Absoluta
CONFIGMOD	-	-	-	-	-
ACCOLY	X	-	-	-	-
MAGONLY	-	X	-	-	-
GYROONLY	-	-	X	-	-
ACCMAG	X	X	-	-	-
ACCGYRO	X	-	X	-	-
MAGGYRO	-	X	X	-	-
AMG	X	X	X	-	-
IMU	X	-	X	X	-
COMPASS	X	X	-	-	X
M4G	X	X	X	X	-
NDOF_FMC_OFF	X	X	X	-	X
NDOF	X	X	X	-	X

O dispositivo possui um módulo de interrupção definida por limite (*threshold*) conforme apresentado na Tabela 2.6 [BOSH 2021], o que significa que pode ser configurado para não “sentir” frequências de vibrações mais altas e conseqüentemente indesejadas, o que seria útil no caso da calibração.

Tabela 2.6 – Configurações de interrupção BNO055. - baseado no manual.

Parâmetros	Valor	Valor no Registrador
Tipo de Detecção	Sem Movimento	[ACC_NM_SET]: xxxxxxx1b
	Movimento Lento	[ACC_NM_SET]: xxxxxxx0b
Parâmetros de Interrupção	Limite	[ACC_NM_THRE]: bit7: bit()
	Duração	[ACC_NM_SET]: bit6:bit1
Seleção de Eixos	Eixo X	[ACC_INT_Settings]: xxxxx1xxb
	Eixo Y	[ACC_INT_Settings]: xxxx1xxxb
	Eixo Z	[ACC_INT_Settings]: xxx1xxxxb

O BNO055 será conectado a uma plataforma móvel de três graus de liberdade, já calibrada, e um conjunto de trajetórias pré definidas será seguido para coletar dados do sistema, a uma taxa inicial de 100 Hertz (padrão do sistema). Estes dados coletados serão inseridos no algoritmo de funcionamento da rede neural artificial, para que esta “aprenda” a correlação entre a classe de movimento e a leitura do sensor, em cada trajetória da plataforma.

Após a maturação do aprendizado da rede neural, com nova topologia de pesos já definida, novos dados serão alimentados *online* para se comparar com os dados de calibração do robô. O esquema proposto seria capaz de inferir como os sensores inerciais se comportam em função

de suas trajetórias e permitir a atribuição de estratégias adequadas para minimizar erros em um cronograma de medição.

## 2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais podem ser enquadradas em um ramo ou subconjunto de uma área maior do conhecimento humano: a Inteligência Artificial – IA [Ali et al. 2016]. A Figura 2.20 ilustra alguns dos ramos que fazem parte da IA.

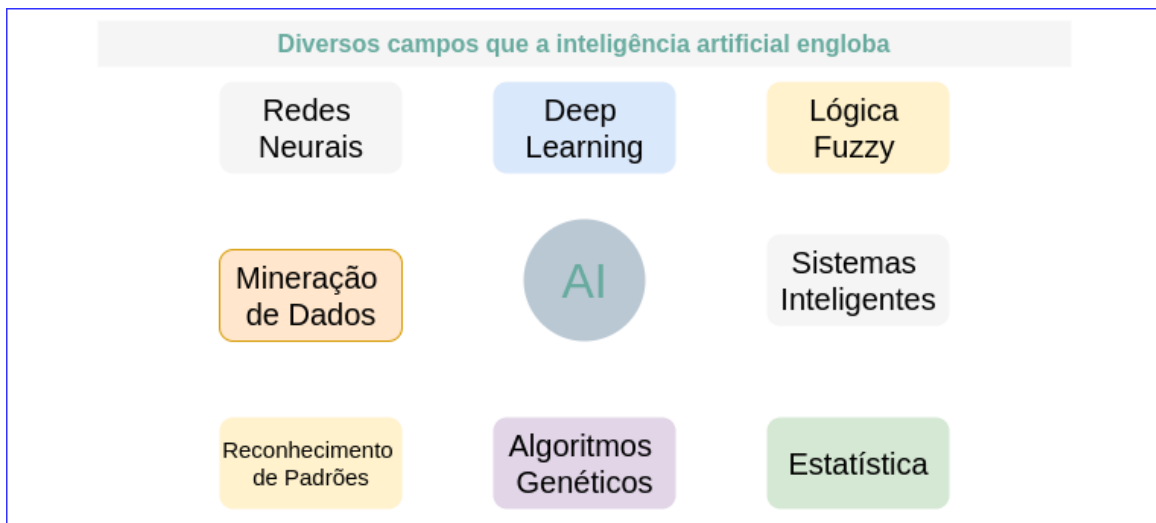


Figura 2.20 – IA e algumas de suas respectivas áreas. Fonte: O Autor.

Um dos objetivos da pesquisa nessa área é fazer com que os computadores "aprendam" a resolver problemas que são considerados simples do ponto de vista humano. Embora simples, esses problemas são significativos porque estão relacionados a trabalhos perigosos e repetitivos, e sua devida execução está ligada à economia e segurança. Além disso, a IA já está beneficiando vários setores, cujas tarefas podem ser automatizadas [Vinuesa et al. 2020]. Neste capítulo, o objetivo é explicar simplesmente a criação de redes neurais, bem como o detalhamento e operação de alguns exemplos típicos de funções desempenhadas por elas e como seu uso se encaixa neste trabalho.

### 2.3.1 O Sistema Nervoso Animal

As RNAs, de forma geral, se baseiam no funcionamento do sistema nervoso animal [Hassabis et al. 2017]. Mesmo os animais mais simples dispõem de um sistema nervoso que controla suas funções orgânicas como digestão, excreção, defesa, etc. Este sistema é capaz de prover o organismo de percepções sensoriais sobre o ambiente em que vive, controlar seus movimentos e armazenar informações úteis para sua sobrevivência e reprodução.

O sistema nervoso tem como elemento central o neurônio (Figura 2.21). Ele se constitui de uma célula, onde receptores eletroquímicos chamados sinapses transmitem o sinal recebido para seu núcleo, que o processa e envia através de sinapses de saída, de seu axônio para outro neurônio, posteriormente. A informação viaja através da sinapse entre os neurônios. Ela chega através dos dendritos e flui através dos axônios. O núcleo é responsável por coordenar toda atividade de uma dada unidade.

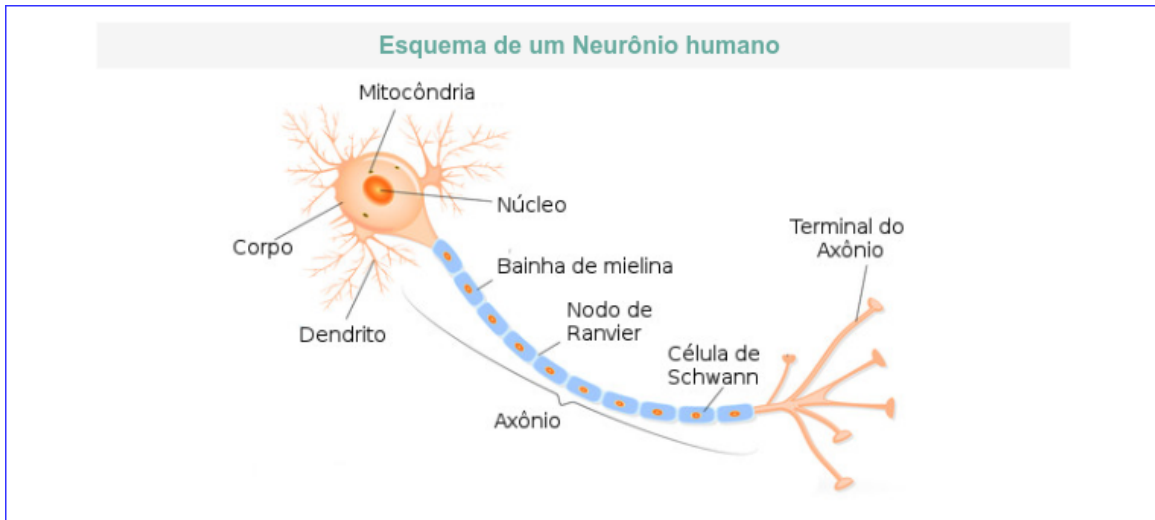


Figura 2.21 – Exemplo de um Neurônio animal. Fonte: [Oliveira 2020]

As sinapses alteram o sinal que recebem ao longo do tempo em função de necessidades orgânicas ou sensibilidade a novas entradas de sinal. Seus “pesos”, que podem ser imaginados como moduladores eletroquímicos aptos a transformar o sinal elétrico que por eles passam, definem o grau de alteração desse sinal e são passíveis de serem modificados.

A possibilidade de mudança física destes moduladores eletroquímicos (“pesos”) em função dos sinais elétricos que recebem é o que permite o “aprendizado” de um sistema nervoso, o reconhecimento de padrões, a memória e outros. Essa possibilidade de mudança se deve a plasticidade do neurônio, onde uma sinapse pode transformar suas características eletroquímicas em função dos inputs que recebe, levando a modificação no seu grau de atuação no sinal elétrico que transmite, refletindo assim o reconhecimento uma característica importante para o organismo [Haykin 1999].

A Figura 2.22 faz uma analogia entre os dois tipos de neurônios, os dendritos podem ser comparados às diversas entradas, as quais irão receber as excitações do sistema  $(x_1, x_2, \dots, x_n)$ , o núcleo pode ser comparado com uma função que irá minimizar o erro entre a entrada e saída e finalmente o axônio que pode representar a saída que irá passar a informação mais relevante para o próximo neurônio.

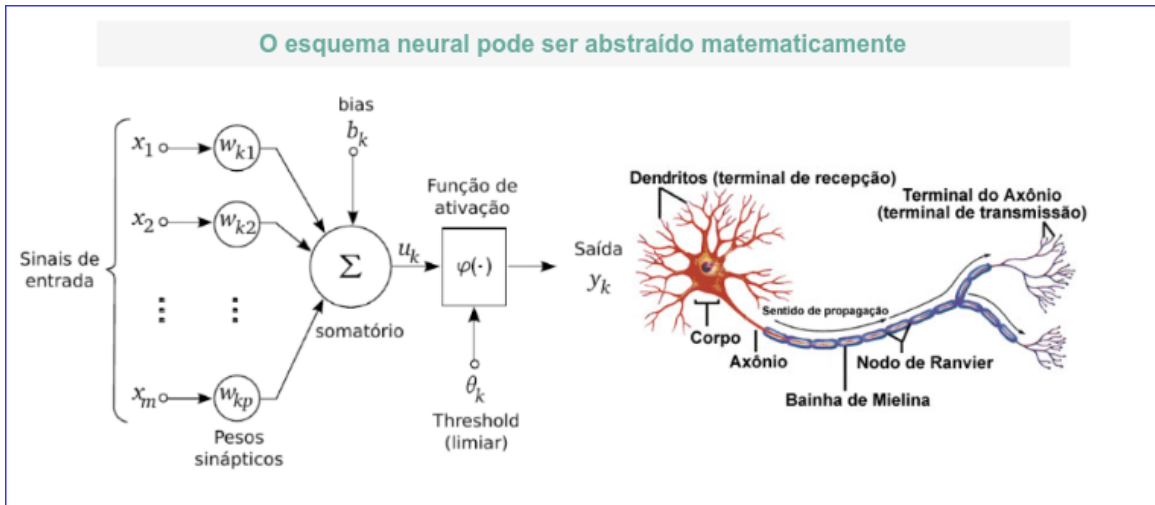


Figura 2.22 – Analogia entre um neurônio artificial e o neurônio animal. Imagem feita a partir de [Silva e Schmidt 2016, Nimbus 2017]

Uma RNA é um circuito composto por uma grande quantidade de unidades simples de processamento (neurônios), inspiradas em uma rede neural natural [Abiodun et al. 2018] e é um sistema massivamente paralelo e distribuído, onde estas unidades simples possuem a capacidade de armazenar e utilizar o conhecimento [Abiodun et al. 2018, Haykin 2009]. A Figura 2.23 apresenta uma topologia geral de uma rede neural.

O “conhecimento” da rede está armazenado nos “pesos  $w$ ” das sinapses de seus neurônios e neste caso o peso nada mais é do que um simples multiplicador matemático que será aplicado ao valor do sinal de entrada desta sinapse. Assim, pode-se entender o processo de aprendizado como a variação dos pesos sinápticos ao longo do tempo, em função dos *inputs* (entradas ou excitações) que recebe, até que os mesmos se estabilizam e reflitam uma relação correta entre um conjunto de entrada (domínio) e sua imagem, representando algo significativo como o reconhecimento de um padrão, a aproximação de uma função ou um tipo de memória.

A topologia de uma rede neural pode ser entendida como um conjunto de camadas paralelas, as quais são ligadas através dos axônios (sinapse) entre suas unidades (neurônios), que realizam operações matemáticas a fim de se obter uma solução que minimize o erro.

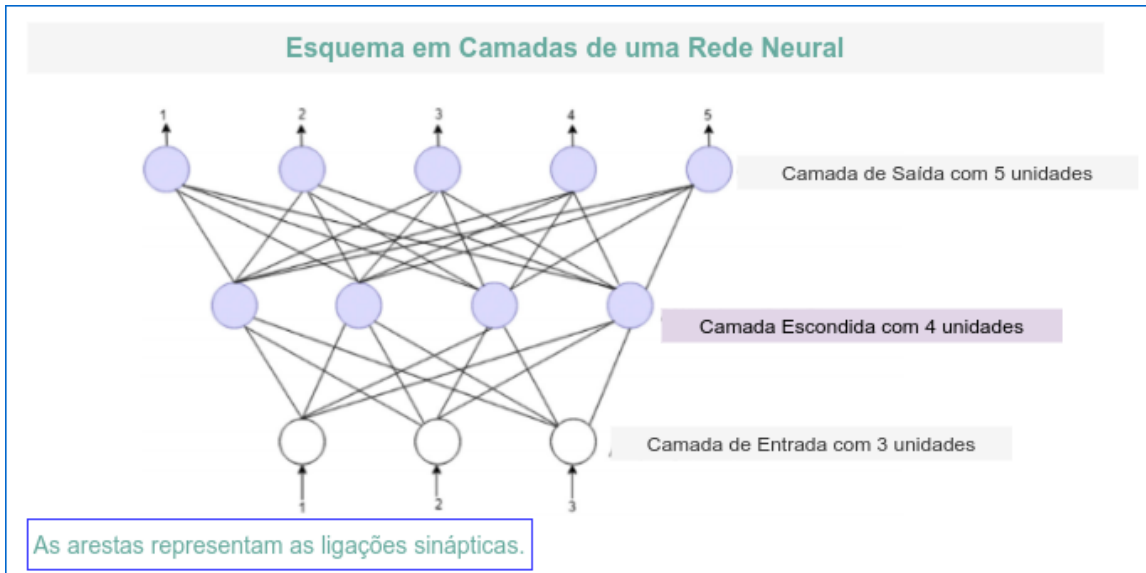


Figura 2.23 – Esquema de uma topologia de rede neural artificial. Imagem feita a partir de [Abiodun et al. 2018]

### 2.3.2 Modelo de uma Rede Neural Artificial

O modelo básico de funcionamento de um neurônio artificial dentro da rede ocorre da seguinte forma:

- As sinapses recebem sinais de entrada,  $x_1, x_2, x_n$ ;
- Os sinais são multiplicados por pesos já existentes (aleatórios no início),  $w_1, w_2, w_n$ ;
- Algum tipo de função é aplicada ao sinal: *threshold*, sigmóide, linear, etc., dando origem ao sinal de saída  $y$  deste neurônio;
- O sinal de saída  $y$  é enviado adiante;
- Após muitas passagens do sinal, os pesos  $w_i$  são alterados e acabam por representar algo relevante contendo um pequeno erro associado.

A saída  $y_k$  de um neurônio  $k$ , pode ser representada matematicamente pela fórmula (33), onde a  $g(\cdot)$  pode ser uma entre várias funções possíveis: *threshold*, sigmóide, linear ou outras. Onde  $w_{jk}$  é o neurônio  $k$  na iteração  $j$ .

$$y_{(k)} = g(u_k) = g\left(\sum_{k=1}^n w_{jk}x_{jk}\right) \quad (33)$$

Métodos de treinamento diversos, dependendo do objetivo, são aplicados até que o erro de análise dos pesos da rede seja mínimo. Dessa maneira a evolução do treinamento de uma rede neural segue o padrão da Figura 2.24. Depois de treinados, os pesos da rede se transformam de algo aleatório e sem sentido em algum tipo objetivo de conhecimento, função ou memória. Quando os sinais de entrada forem novamente inseridos no neurônio, sua saída refletirá uma relação correta entre um conjunto de entrada (domínio) e sua imagem.

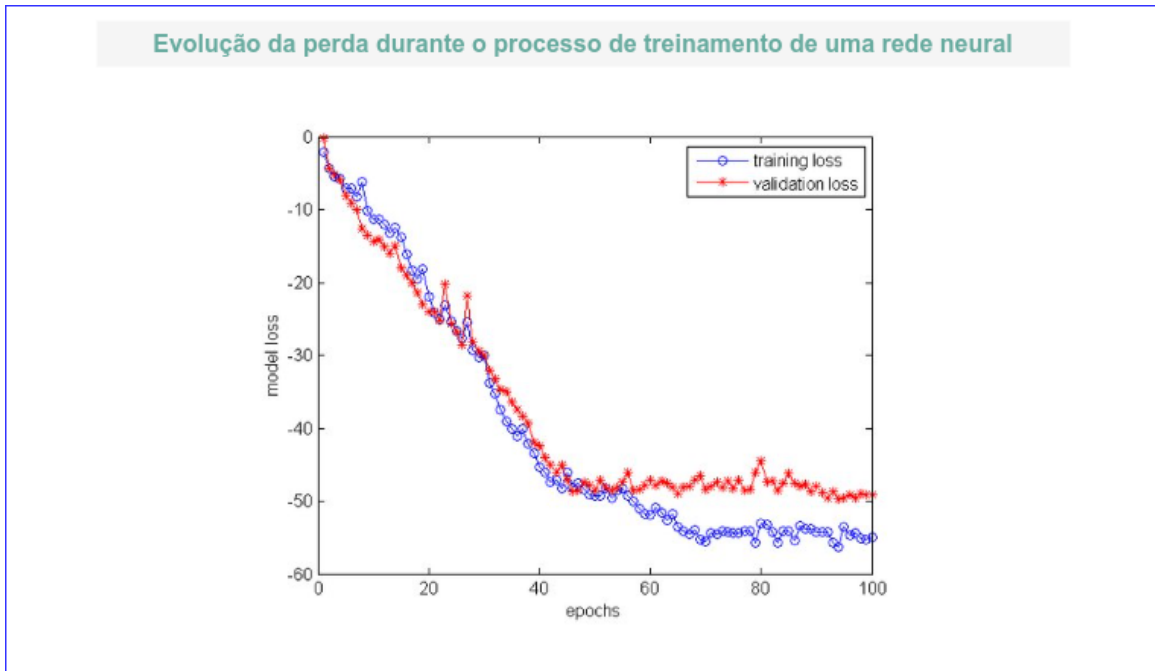


Figura 2.24 – O modelo tenta diminuir o erro entre os valores esperados e obtidos, a evolução nem sempre é homogênea e varia entre as diferentes fases do processo. Figura criada a partir de [Li et al. 2018].

É natural utilizar a maioria dos dados durante o processo de treino. A acurácia apresentada pelo conjunto de treino também é melhor, porém não reflete o comportamento do modelo em um sistema *online*. Os dados usados para validação representam uma parcela menor (20% ou 30%) do total. A validação obtém resultados inferiores em comparação com os dados usados para o treinamento, já que a rede é treinada sobre estes valores.

### 2.3.3 RNAs: Tipos

De uma forma geral, as RNAs podem ter seu aprendizado supervisionado ou não, ter diferentes topologias e contar com diversos métodos de funcionamento (Tabela 2.7).

Tabela 2.7 – Exemplos de algumas possibilidades de funcionamento de uma RNA.

Método de Aprendizado	Supervisionadas	Não-supervisionadas
Algoritmos	Algoritmo Backpropagation Algoritmos Convolucionais Algoritmos Mistos, etc.	Algoritmos de cluster, K-means, clustering hierárquico.
Topologias	Totalmente Conectadas	Centróides que minimizem a inércia

As RNA's supervisionadas, como o nome diz, recebem a supervisão de um parâmetro de controle que direciona seu aprendizado. As entradas são mapeadas para as saídas, dessa forma a rede aprende baseando-se no padrão apresentado. Pode-se fazer uma analogia com um aluno, que através de exemplos é capaz de descrever uma função que mapeia um vetor de entrada em uma respectiva classe [Nasteski 2017].

Em seu conjunto de treinamento o vetor de entrada é dividido em dados de entrada  $x_i$  e valores desejados de saída  $y$ , onde o objetivo a ser atingido é a transformação dos dados de entrada  $x$  em valores desejados  $y_d$  da seguinte forma:

1. Os valores dos dados de entradas são multiplicado pelos pesos  $W$  e somados (soma ponderada);
2. Ao resultado do somatório é aplicada uma função de ativação  $g(.)$  gerando um vetor de saída  $y$  em cada neurônio, que será propagado ao longo da rede;
3. O cálculo do último valor  $y$  (de saída) é produzido em resposta ao estímulo do vetor  $x_i$  de entradas, processado ao longo dos nós da rede;
4. - O  $y$  de saída é então comparado ao  $y_d$  desejado: fator de supervisão do algoritmo;
5. Ocorre o cálculo do erro, que corresponde à diferença entre o valor  $y$  calculado e o valor  $y_d$  esperado;
6. Acontece o processo de adaptação, que no caso seria a mudança dos pesos sinápticos dos neurônios da RNA em função do sinal de erro recebido e das derivadas em cada sinapse da rede;

A Figura 2.25 exemplifica a topologia de uma rede neural, onde o erro é retro propagado de volta as unidades para que possa ocorrer o ajuste dos pesos e, assim, minimizar o erro entre a saída e o valor desejado.



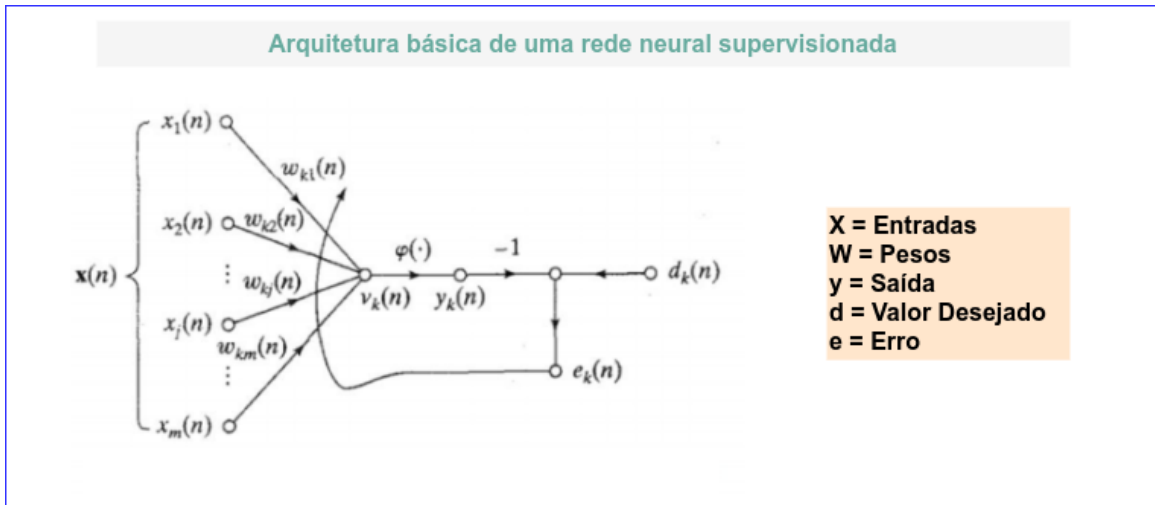


Figura 2.25 – Grafo de uma Rede Neural Artificial supervisionada. Ilustração desenvolvida com base em [Haykin 2009].

Nesse tipo de configuração, a saída esperada é apresentada a rede, que pode usar o erro para ajustar os pesos durante o processo de aprendizagem.

Já nas RNAs não-supervisionadas, não há a supervisão de um parâmetro de controle que direciona seu aprendizado. Ao contrário, o aprendizado sobrevém praticamente sem intervenção. O principal objetivo do aprendizado não supervisionado é criar automaticamente rótulos de classificações. O algoritmo pesquisa a semelhança entre os dados para em seguida poder agrupá-los [Nasteski 2017].

Um exemplo deste caso seria o posicionamento de centróides das Redes conhecidas como *Radial Basis Fuction* ou Função de Base Radial – RBF [Jessie e Albert 2018], usando agrupadores lógicos de segmentação para encontrar similaridade entre dados da amostra. Nas redes RBFs onde se queira, por exemplo, aproximar uma função, são usadas funções de ativação gaussianas cujos centros  $\mu$  inicialmente aleatórios convergem para uma distribuição que represente a função a ser aproximada [Wu et al. 2012]. A Figura 2.26 exemplifica as funções de ativação gaussianas.

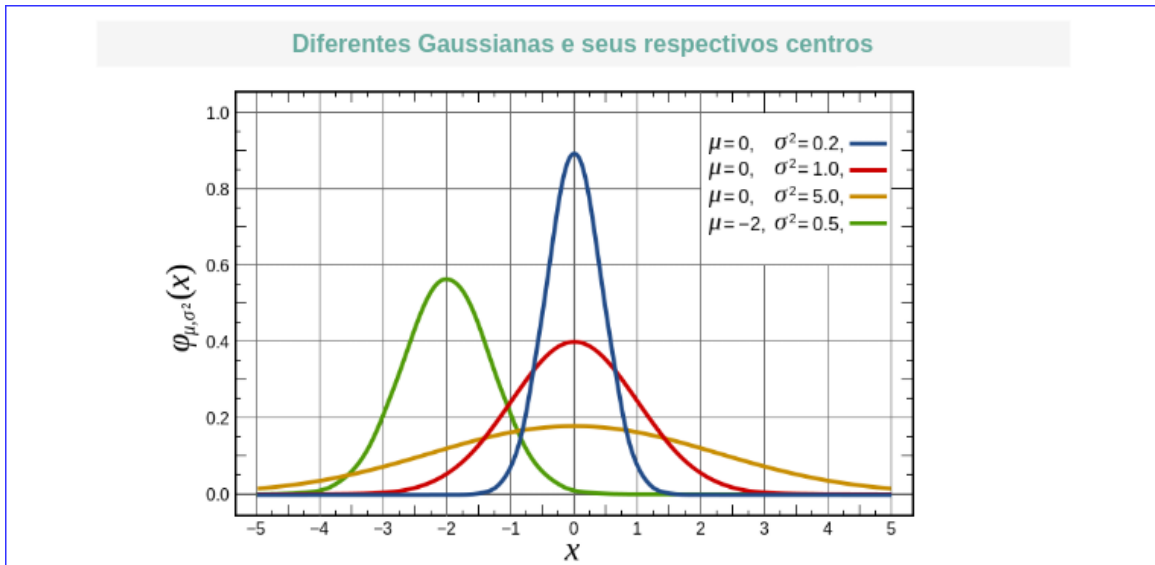


Figura 2.26 – Exemplos de funções de ativação e seus respectivos centróides, usadas para realizar o processo de aprendizado da Rede Neural Artificial. Fonte: Creative Commons.

As funções gaussianas correspondentes a cada neurônio são posicionadas aleatoriamente ou distribuídas de forma uniforme ao longo do conjunto de entradas em  $R^n$ . Um algoritmo agrupador move seus centros para onde se encontram as maiores densidades de valores de amostra. Ao resultado do agrupamento é aplicada a atualização de pesos (coeficientes das funções gaussianas) gerando um vetor de saída  $Y$  em cada neurônio, que corresponde à função aproximada; Como exemplo de funcionamento dos agrupadores lógicos (Figura 2.27) pode-se citar o seguinte caso:

1. A um conjunto de dados de entrada  $x$ , com três *clusters* (agrupamentos de pequenos quadrados cinza), são posicionados aleatoriamente três círculos (vermelho, verde, azul) chamados de centros;
2. As medições de distância de cada entidade dos *clusters* de entrada  $x$  são realizadas em relação a cada centro;
3. Cada ente do conjunto de entrada  $x$  é agora conectado ao centro mais próximo de si, delimitando assim uma fronteira de domínio de cada centro;
4. A partir deste domínio é calculado o centróide do *cluster* (com base na distância destes pontos entre si) e o centro é movido para o centróide;
5. Os passos 2 a 4 são repetidos até que a variação de posição dos centros seja menor que um valor determinado  $\xi$ .

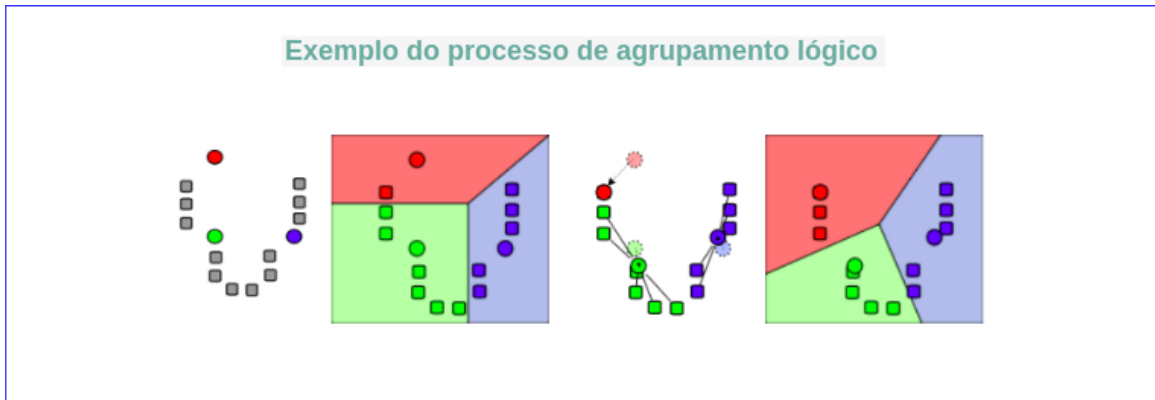


Figura 2.27 – Passos de funcionamento dos agrupadores lógicos. Fonte: Creative Commons.

Os grupos são separados de acordo com a semelhança de suas *features* (características). Dessa forma o algoritmo não precisa comparar os dados com uma saída esperada. Tal processo é dito não supervisionado. Isso mostra como dados semelhantes, no caso agrupados em pequenos *clusters*, podem ser detectados e separados sem nenhuma intervenção de supervisão.

As redes neurais também podem variar em termos de sua topologia [Stier et al. 2018], ou seja, no número de camadas que apresenta, número de neurônios em cada camada e ainda no tipo de conexão entre seus neurônios. Podem, inclusive, variar em função do algoritmo que usam para o aprendizado. As redes RBF, citadas anteriormente têm apenas três camadas [Jessie e Albert 2018], sendo elas a camada de entrada, a camada escondida e a camada de saída. Por apresentarem um pequeno número de camadas, as redes deste tipo são conhecidas como *Shallow Learnings* e as que têm muitas camadas são conhecidas como *Deep Learning*.

Um exemplo de arquitetura baseada em *Deep Learning* ou Aprendizado Profundo (DL) são as Redes Neurais Convolucionais (CNN). Elas costumam ser usadas para identificar e classificar imagens complexas mesmo que estejam embaralhadas, em posições diversas e em condições de iluminação variadas. Tal algoritmo também foi pensado com base em um sistema biológico. Inspirada no sistema visual dos gatos, os quais possuem um elaborado esquema de filtragem das imagens que são captadas através dos olhos. A rede utiliza a operação matemática da convolução para realizar a sobreposição das imagens e simular o mecanismo visual.

As CNNs são semelhantes às RNA comuns, descritas anteriormente: são constituídas por neurônios que possuem pesos e tendências. Cada neurônio recebe algumas entradas, realiza uma soma ponderada e, opcionalmente, prossegue com a aplicação de uma função cujo resultado é a não linearidade. A diferença é que as arquiteturas convolucionais assumem explicitamente que as entradas são imagens, o que permite a codificação de certas propriedades no projeto (*kernels*). Além disso, os neurônios em uma camada serão conectados apenas a uma pequena região da camada anterior e não estarão totalmente conectados [Emmert-Streib et al. 2020]. Isso torna a função direta mais eficiente para programar e reduzir o número de

parâmetros na rede.

CNNs, em geral, são compostas por um grupo de camadas convolucionais, *pooling* e camadas totalmente conectadas. As mesmas são descritas como segue:

1. **Camada Convolutacional:** Similar a uma camada escondida de uma rede neural qualquer, cujo a função é mapear a entrada de forma mais abstrata, porém utilizando uma conectividade apenas local. A rede usa essa camada para realizar a operação da convolução, onde no mínimo um *kernel* é deslizado sobre a entrada. O resultado da operação é armazenado em um mapa de ativação, contendo os *features* extraídos [Emmert-Streib et al. 2020];
2. **Pooling:** Visa reduzir o tamanho da entrada, tentando conservar a informação. É capaz de introduzir in-variança espacial, permitindo à rede melhorar a generalização do modelo. Ela examina toda entrada com o tamanho de uma janela específica chamada *pool* [Dominik et al. 2010];
3. **ReLu (*The Rectified Linear Unit*):** É usada para retificar os valores, ou seja, atribuir limites. Valores menores que zero recebem zero, por exemplo;
4. **Camada totalmente conectada:** É uma simples unidade escondida, colocada entre a penúltima camada e a saída. Ela modela a relação não linear entre os *features* [Szegedy et al. 2015]. Ultimamente seu uso tem sido questionado [Emmert-Streib et al. 2020].

A Figura 2.28 apresenta uma CNN genérica e suas partes.

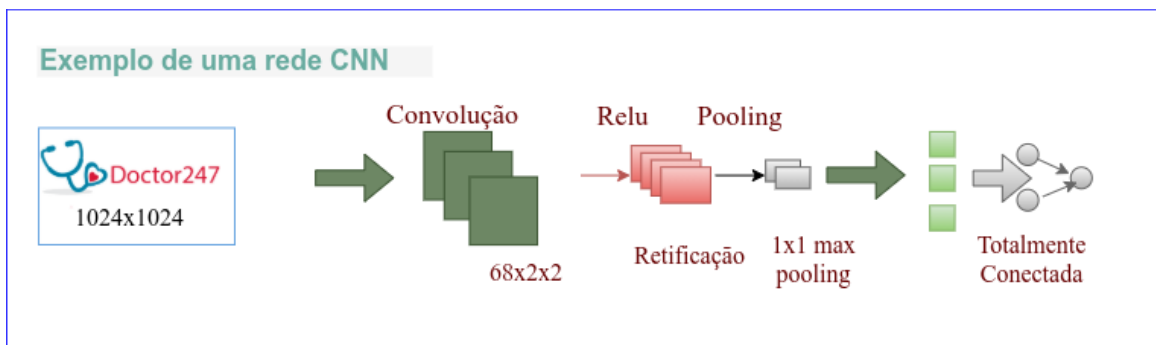


Figura 2.28 – Exemplos das camadas empregadas em uma arquitetura típica de uma CNN. Fonte: O Autor.

O córtex cerebral dos gatos possui uma estrutura sequencial de filtragem que sobrepõem as partes do campo de visão. Isso inspirou a construção da rede CNN utilizando um esquema sucessivo de filtragem entre as camadas. Através de um elaborado conceito de filtragem sucessivas, tal algoritmo é capaz de realizar a engenharia de *features* a priori. Ou seja,

a extração das características mais relevantes ficam a cargo da rede, eliminando uma das etapas do processo.

Supondo que as entradas são imagens e que as imagens são armazenadas no computador em forma de matrizes contendo zeros e uns, pode-se fazer uma analogia entre a operação de soma ponderada e o produto escalar de dois vetores, conforme descrito pelas seguintes equações:

$$A = \{a_1, a_2, a_3 \dots a_n\} \quad B = \{b_1, b_2, b_3 \dots b_n\} \quad (34)$$

$$A \bullet B = \{a_1 b_1 + a_2 b_2 + a_3 b_3 \dots + a_n b_n\} \leftrightarrow |A| |B| \cos \theta \quad (35)$$

Da mesma forma, na rede neural temos  $X$  e  $W$ , vetores representando conjunto e entrada e pesos respectivamente:

$$A \bullet B = \{x_1 w_1 + x_2 w_2 + x_3 w_3 \dots + x_n w_n\} \leftrightarrow |X| |W| \cos \theta \quad (36)$$

O que leva a inferir que um determinado arranjo de pesos pode ser considerado uma forma padrão (*kernel*) e a operação de convolução, no caso a soma ponderada, seria a projeção deste *kernel* na imagem (vide Figura 2.29) representada pelas entradas.

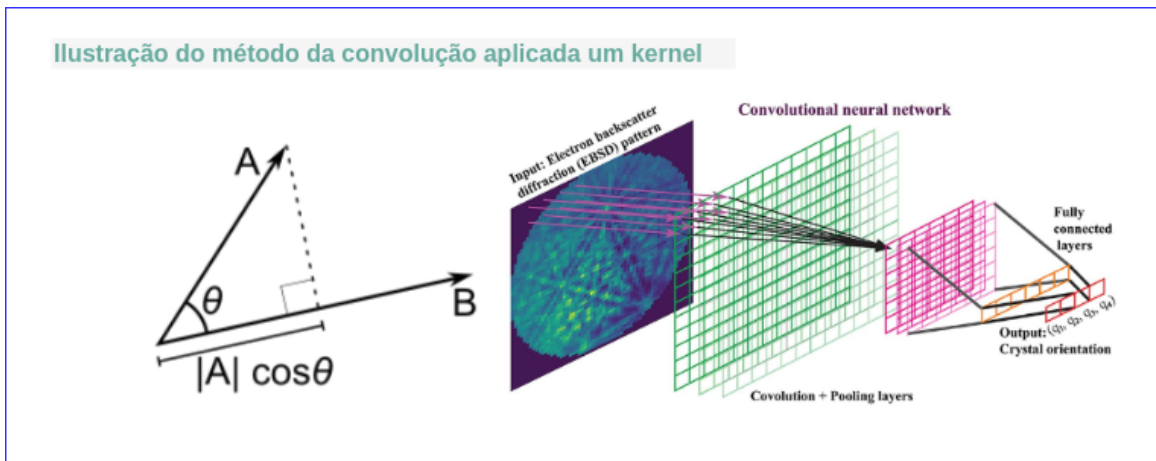


Figura 2.29 – Operação de convolução de um determinado Kernel. Fonte [Shen et al. 2019].

Assim um *kernel* representando uma característica de imagem pode extrair ou não esta característica, na medida em que o neurônio que representa este *kernel* está ou não ativado da seguinte forma:

a - Toma-se uma imagem, por exemplo, 100x100 *pixels* em tons de cinza, onde o nú-

mero 255 representa o preto, o número 0 representa o branco e números intermediários representam tons de cinza;

- b - Esta imagem é decomposta em 10000 *pixels* (100 x100) variando de 0 a 255 e seus valores são inseridos na rede neural como conjunto de entrada;
- c - Se um dos neurônios da primeira camada representar, por exemplo, uma borda do tipo (0,255,0; 0,255,0; 0,255,0), quando ocorrer a convolução da imagem (conjunto de entrada) com este neurônio, se ele estiver ativado vai propagar bordas horizontais coincidentes com a posição destas na imagem convolucionada;
- d - Assim vários tipos de *kernels* como retas horizontais, retas verticais, curvas, etc. são programados na primeira camada, fazendo com que a rede neural os reconheça na imagem e os propague para a próxima camada;
- e - Nas camadas seguintes aplicam-se novos *kernels* cada vez mais complexos como círculos, quadrados, etc. seguidos por texturas e depois por imagens como as de olhos, bocas, rodas até chegar às formas finais como rostos, objetos, etc;
- f - Estes últimos *kernels*, quando ativados, funcionam como reconhecedores e classificadores das imagens mais complexas, a partir de informações simples advindas dos *kernels* das camadas anteriores.

Esse tipo de RNA seria um candidato a ser usado na calibração usando IMU por ser capaz de descobrir os pontos de reinicialização do sistema como se percebesse uma imagem. A Figura 2.30 mostra o processo de mapeamento de *features* de uma CNN. É possível inspecionar os filtros bidimensionais que o modelo cria e dessa forma descobrir os tipos de características que a rede detecta. Com um tipo de engenharia reversa é viável descobrir a natureza das características de cada imagem.

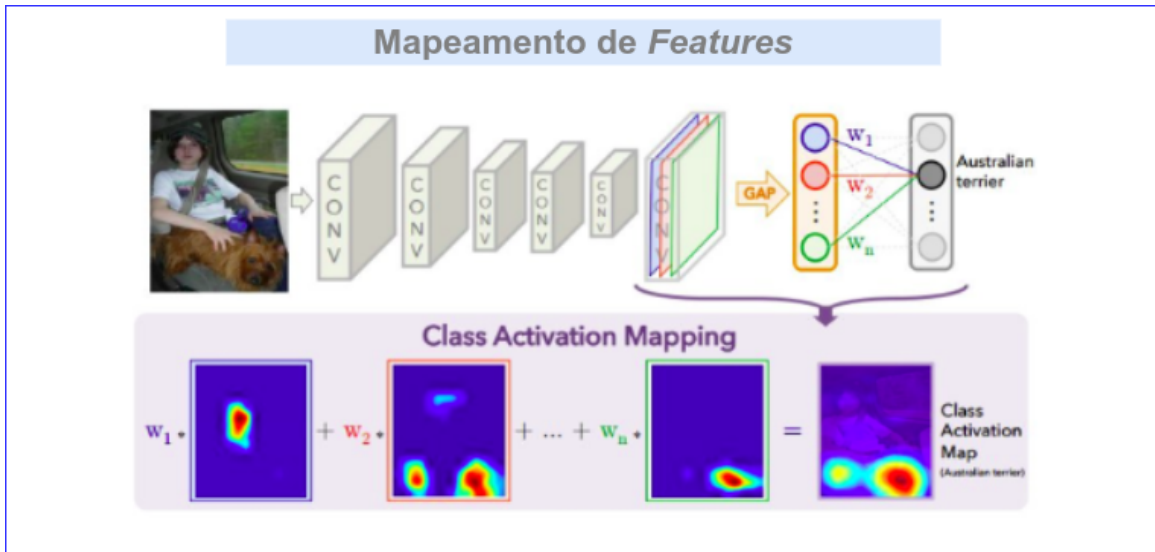


Figura 2.30 – Extração de características de uma CNN. A partir da imagem fornecida, a rede mapeia os *features* automaticamente, possibilitando assim, o reconhecimento de padrões. Fonte: [Zhou et al. 2016].

Outra forma de redes neurais, em especial e que também pode ser usada no caso da calibração dos IMU's, é a que mimetiza um filtro adaptativo [Haykin 1999]. Se considerarmos um sistema dinâmico onde não se conhece seu exato modelo matemático, mas se sabe que para cada vetor de entradas  $x$  no sistema existe um vetor de saídas  $y$  produzidas em função do estímulo de entrada, pode-se inferir o modelo matemático através do uso do filtro adaptativo. Tomando como hipótese que o sinal tem um ruído, quando o filtro elimina o ruído, sobra apenas a função que governa a relação de entrada e saída do sistema dinâmico. O método de filtragem envolve três passos fundamentais:

1. O cálculo do valor  $y$  (de saída) produzido em resposta ao estímulo do vetor  $x$  de entradas processado ao longo dos nós da rede;
2. O cálculo do erro que corresponde a diferença entre o valor  $y$  calculado e o valor  $y$  esperado;
3. O processo de adaptação, que no caso seria a adaptação dos pesos sinápticos dos neurônios da RNA em função do sinal de erro recebido;
4. - Estes passos se repetem em um *looping* de propagação e retropropagação que ocorre até que valores mínimos aceitáveis de ruído sejam percebidos pelo sistema.

A descrição acima se encaixa no caso concreto do sinal que um IMU envia para cálculo da posição do sensor, onde os ruídos inserem erros que contaminam a medição e, portanto, pode ser usada para calibrá-lo. Uma vez filtrados estes ruídos, o sinal limpo pode apresentar um valor confiável para cálculo da posição, ou seja, o IMU se torna calibrado.

Assim, no caso em questão aplicado ao trabalho de calibração de IMU's, a finalidade do filtro seria construir uma função que aproxima a relação entre um conjunto de partida contendo os valores oferecidos pelo IMU em tempo real e um conjunto de chegada que seriam os valores  $x$ ,  $y$ ,  $z$  da posição do mesmo. A RNA seria usada como aproximador de uma função não-linear, onde os pesos de suas sinapses representa os coeficientes desta função.

### 2.3.4 RNAs: Formulação matemática no caso da IMU

Para descrever seu funcionamento pode-se pensar no seguinte cenário:

- a - Sinais de entrada são enviados pelo IMU a uma taxa definida, por exemplo, 100 hertz;
- b - Estes valores, a priori, deveriam corresponder a certo conjunto  $x$ ,  $y$ ,  $z$  de posições que a IMU percorre ao tempo que envia os citados sinais;
- c - Na realidade estes valores de sinal não correspondem corretamente às posições percorridas, pois são contaminados de ruídos;
- d - A RNA é montada em alguma topologia previamente determinada usando o algoritmo *Backpropagation* (descrito adiante) com pesos sinápticos inicialmente aleatórios para receber os sinais de entrada do IMU;
- e - Os sinais são processados de acordo com a regra matemática da RNA proposta e uma saída é gerada contendo valores calculados  $x$ ,  $y$ ,  $z$ ;
- f - Estes valores calculados  $x$ ,  $y$ ,  $z$  são comparados com valores esperados medidos anteriormente e um sinal de erro é gerado a partir da diferença;
- g - O erro é retropropagado para o início da rede modificando seus valores de peso sináptico que ficarão à espera de novo sinal.
- h - Novos sinais de entrada são enviados pelo IMU e o processo se repete até que os valores de erro sejam menores que um valor definido  $\xi$ .

De forma matemática o valor  $y_k$  calculado na saída de cada neurônio é dependente da entrada  $x_j(k)$ , dos pesos  $w_j(k)$  das sinapses do neurônio e de uma função de ativação  $g(\cdot)$  aplicada à soma ponderada, conforme a equação (37).

$$y_{(k)} = g\left(\sum_{k=1}^n (w_j(k)x_j(k))\right) \quad (37)$$

O erro  $e_{(k)}$  é a diferença entre este valor calculado  $d_{(k)}$  e o valor desejado  $y_{(k)}$ , conhecido na saída do modelo, descrito pela fórmula (38).



$$e_{(k)} = d_{(k)} - y_{(k)} \quad (38)$$

No processo de adaptação deve-se utilizar uma regra que redimensione o valor dos pesos  $w_j(k)$  das sinapses dos neurônios para que seu efeito ao multiplicar o vetor de entradas  $x_j(k)$  aproxime o resultado do  $d_{(k)}$  (desejado). A literatura mostra que, considerando o objetivo de minimizar o erro, é interessante aplicar uma função de energia ou custo aplicada ao erro, que ao minimiza-lo aproxime os valores  $w_j(k)$  de seu ponto ótimo. Sendo  $E_{(W_v)}$  uma função de energia ou custo aplicada ao vetor de pesos  $W_v$ , continuamente diferenciável, que mapeia o vetor para números reais, então existe uma  $E_{(W^*)}$  onde os valores  $w^*$  são ótimos no sentido de aproximar a função alvo  $y_{(k)}$ . A relação entre elas é dada pela equação (39).

$$E(W^*) \leq E(W_v) \quad (39)$$

A condição necessária para minimizar a função  $E(W^*)$  onde  $w^*$  é seu valor ótimo é:

$$\nabla E(W^*) = 0 \quad (40)$$

onde  $\nabla$  é o operador gradiente:

$$\nabla = \left[ \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \frac{\partial}{\partial w_3} \dots \frac{\partial}{\partial w_n} \right]^T \quad (41)$$

e o gradiente da função de custo é:

$$\nabla E(W_v) = \left[ \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3} \dots \frac{\partial E}{\partial w_n} \right]^T \quad (42)$$

Sabendo disso, pode-se basear o algoritmo em uma iteração descendente, onde a cada passo  $n$  a função de custo diminui conforme segue, onde  $w(n)$  representa o peso na iteração  $n$ :

$$E(w(n+1)) < E(w(n)) \quad (43)$$

Neste ponto várias possibilidades de adaptação estão disponíveis como os algoritmos de descida íngreme (*steepest descent*), mínimos quadrados (*least mean squares*), *GaussNewton*, etc. Cada uma com suas vantagens e desvantagens. A que melhor se qualifica para este caso concreto é a *multilayer-perceptron* (MLP) com *Backpropagation* (retropropagação) pelos motivos a seguir:

- a Tem sua convergência garantida desde que respeitados seus requisitos;

- b) Pode ser implementada usando computação paralela o que a torna veloz quando usando grande frequência de amostragem;
- c) O algoritmo é compatível com o uso de acelerador CUDA para GPU's Nvidia (placas que foram adquiridas para uso neste trabalho).

Na MLP a topologia padrão se assemelha à Figura 2.31, onde  $x$  é o valor de entrada de cada camada e  $y$  seu valor de saída que deverá ser conectado na entrada da camada seguinte. A última camada da rede tem seu valor comparado formando o sinal de erro e seus sinais de erro geram sensibilidades que são retropropagadas no sentido de modificar os pesos. A Figura 2.31 exemplifica uma rede do tipo MLP e seu esquemática da retropropagação. A entrada aplicada a rede é propagada através das camadas e seus pesos permanecem fixos, tal evento é chamado de processamento direto. Já no processamento reverso o erro calculado na saída é propagado no sentido reverso, isso possibilita corrigir os pesos de acordo com uma regra de correção [Haykin 1999].

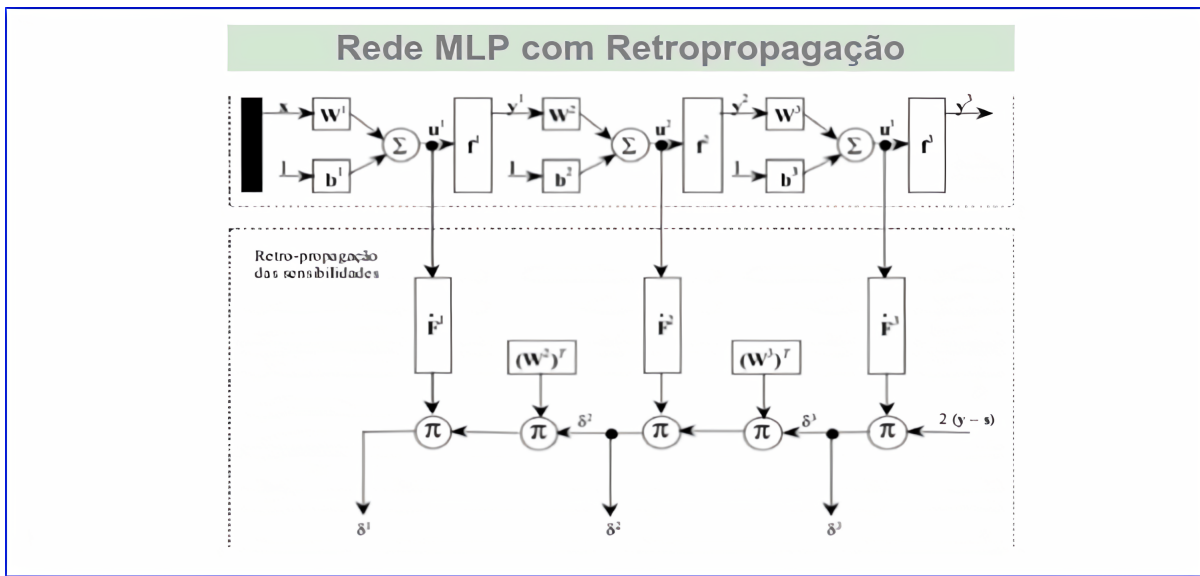


Figura 2.31 – Topologia de uma rede neural MLP. Fonte: [Haykin 2009].

Na MLP podemos definir o sinal de erro  $e$  no neurônio  $j$  durante a iteração  $n$  conforme (44):

$$e_j(n) = d_j(n) - y_j(n) \tag{44}$$

onde  $d_j(n)$  é o valor desejado na iteração  $n$  do neurônio  $j$  e  $y_j(n)$  é a saída obtida na iteração  $n$  do neurônio  $j$ . Assim a energia instantânea do erro do neurônio  $j$  e na iteração  $n$  é:

$$E_{instant\grave{a}nia} = \frac{1}{2} e_j^2(n) \quad (45)$$

Já a energia total obtida somando-se a de todos os neurônios da camada de saída (*output*) na iteração  $n$  seria definida pela equação (46):

$$E(n) = \frac{1}{2} \sum_{j \in c} e_j^2(n) \quad (46)$$

Sendo  $N$  a soma de todas as iterações  $n$ , a energia média total será:

$$E_{med} = \frac{1}{N} \left( \sum_{K=1}^N E(n) \right) \quad (47)$$

onde a energia média assim como a instantânea são funções de todos os parâmetros livres como os pesos sinápticos e os bias. Definindo  $s_j(n)$  como a soma ponderada em cada neurônio, conforme equação (48):

$$s_j(n) = \sum_{i=0}^m (w_{ij}(n) y_{ki}(n)) \quad (48)$$

onde  $n$  é o número total de entradas, excluindo o bias temos:

$$y_j(n) = g(s_j(n)) \quad (49)$$

O algoritmo escolhido, quando usa o Backpropagation, aplica a modificação dos pesos  $\nabla w$  proporcional à derivada parcial  $\frac{\partial E(n)}{\partial w_{ij}(n)}$ . Assim esta derivada também chamada de fator de sensibilidade pode ser expressa da forma:

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial s_j(n)} \frac{\partial s_j(n)}{\partial w_{ij}(n)} \quad (50)$$

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (51)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (52)$$

$$\frac{\partial y_j(n)}{\partial s_j(n)} = \varphi'(s_j(n)) \quad (53)$$

$$\frac{\partial s_j(n)}{\partial w_{ij}(n)} = y_j(n) \quad (54)$$

Utilizando-se (50), (51), (52), (53) e (54), chega-se a:

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -e_j(n)\varphi'(s_j(n))y_j(n) \quad (55)$$

e a correção aplicada ao peso sináptico  $\nabla w$  será:

$$\nabla w_{yj}(n) = -\eta \frac{\partial E(n)}{\partial w_{ij}(n)} \quad (56)$$

onde  $\eta$  representa a taxa de aprendizado aplicada ao algoritmo. Assim, o gradiente local também chamado de fator de sensibilidade será igual a equação (57):

$$\delta_j(n) = \frac{\partial E(n)}{\partial s_j(n)} \quad (57)$$

Este fator de sensibilidade  $\delta_j(n)$  será retropropagado até o início da topologia, de modo a subsidiar a atualização dos pesos sinápticos (Figura 2.32). Ao final de cada época os pesos são ajustados para minimizar o erro. A rede literalmente aprende através de exemplos e faz um ajuste fino nas etapas finais do treinamento. A taxa de convergência da função de treinamento é maior nas etapas iniciais.

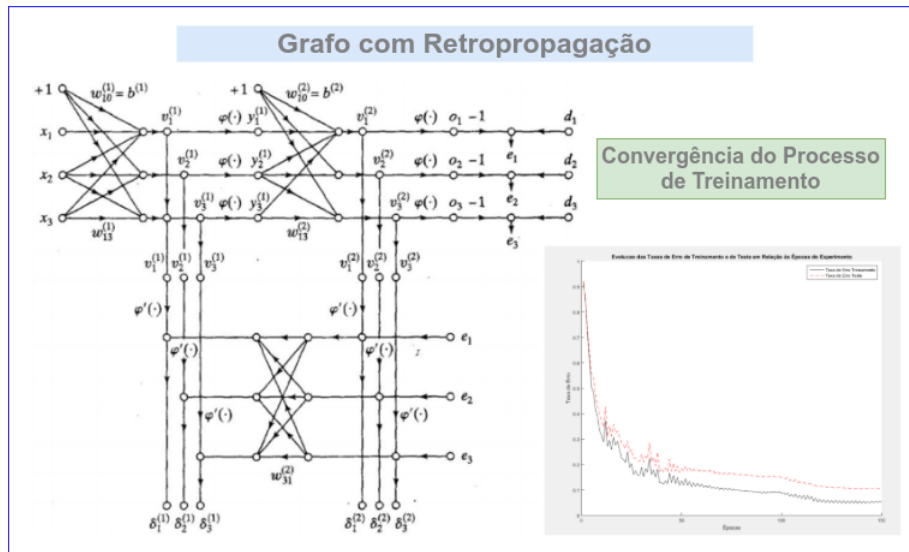


Figura 2.32 – Grafo de uma rede neural MLP e convergência do treinamento. Imagem feita a partir de [Haykin 2009].

O erro proveniente da diferença entre o valor de saída da última camada e o valor desejado

é multiplicado por  $\varphi'$  e o novo peso  $w$  será descrito segundo a equação (58):

$$w_{yj(n)} = w_{yj(n-1)} + \eta \frac{\partial E(n)}{\partial w_{ij(n)}} \quad (58)$$

Assim, a cada ciclo ou época rodada no algoritmo os pesos  $w$  são atualizados no sentido de aprender a regra da função que leva os valores do conjunto de entrada ao conjunto de saída.

A Figura 2.32 mostra a evolução da taxa de erro de um exemplo criado para descrever o uso desta técnica. O programa resolve o problema clássico de classificação de dígitos escritos à mão (*handwritten classification*) e foi escrito em Matlab. Lembrando que a convergência rápida e correta da rede depende de vários fatores relacionados com sua topologia. É necessário lembrar que mesmo sendo uma versão simples de teste desse tipo de programa, em apenas 150 épocas a taxa de acerto sobe rápido para valores próximos de 90%, ou seja, com valores de erro próximos de 10% como vistos no gráfico. Fatores que influenciam o comportamento do erro como o número de camadas, número de neurônios em cada camada, taxa de aprendizado, etc. são chamados de hiperparâmetros da topologia e estes valores em geral são obtidos não a partir de um método rígido, mas sim por heurística, que depende em muito da experiência de quem constrói a rede neural [Jia et al. 2019].

Como não é possível de antemão saber qual a melhor topologia da RNA, pode-se partir de padrões conhecidos e modificá-los, medindo o desempenho de resposta da rede a cada mudança de seus hiperparâmetros. Neste presente trabalho, o que se deseja em termos de topologia é partir de uma rede neural recorrente com *backpropagation*, contendo os valores da IMU como entradas  $x_k$ , os valores do controlador do robô como saídas desejadas  $y_k$  e se utilizando dos valores de posição  $x$ ,  $y$ ,  $z$  como realimentação na entrada da RNA, onde seus hiperparâmetros seriam modificados por métodos heurísticos até que se aproximem de um resultado de convergência.

Um método heurístico que se acredita útil na aproximação de topologias aceitáveis é o chamado Planejamento Fatorial. A estratégia consiste em montar uma série de experimentos cujas variáveis de projeto consistem em dois ou mais fatores, cada um com valores ou "níveis" discretos e cujas unidades experimentais assumem todas as combinações possíveis desses níveis em todos esses fatores [Balestrassi et al. 2009]. Esse método não é comumente visto em redes neurais, porém foi bastante usado em outras áreas de conhecimento para aperfeiçoar processos industriais. É um método que parece promissor e será usado neste trabalho, na construção da rede que aprenderá a calibração do IMU. A Figura 2.33 ilustra esse método.

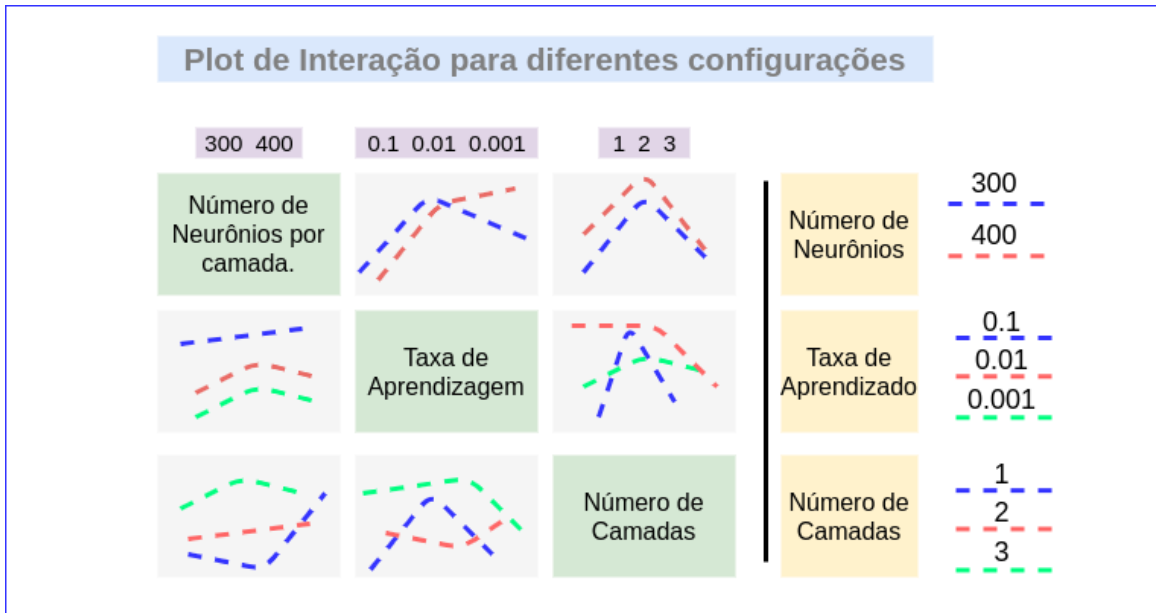


Figura 2.33 – Esquema de experimento de um Trabalho Fatorial. Diversos parâmetros são testados em um esquema em *grid* que varre diversas possibilidades. Fonte: O Autor.

Essa forma de heurística é importante não só para saber qual o fator que aprimora o funcionamento da rede, mas também permite entender os efeitos das interações entre eles, ou seja, qual a relação que este fator tem com outros fatores coadjuvantes. Dependendo do caso, o aumento de um deles leva a diminuição dos outros e conseqüentemente piora o rendimento da rede neural ou vice-versa. Existem também outras formas de melhorar a convergência da rede neural como parada antecipada, variação da taxa de aprendizado, etc. e isso costuma ser usado dependendo do problema e do custo computacional. Essas formas serão aqui exploradas.

## Capítulo 3

# Metodologia

Não pense por muito tempo, faça. Mas não faça por muito tempo, pense.

---

Confúcio

Neste trabalho, em termos de metodologia, o que se pretende é realizar um estudo experimental de medição comparativa das posições obtidas pelo robô usando IMUs, em um movimento guiado de *Dead Reckoning* programado anteriormente, que colete grande quantidade de pontos no decorrer de sua trajetória e, de posse destes dados, utilizar um software de RNA para efetuar os devidos processos numéricos, no sentido de calibrar primeiramente a IMU e posteriormente (em um trabalho futuro), usando a IMU calibrada, calibrar o robô em outras posições de seu espaço de trabalho. Assim, seria gerada uma função que correlaciona as posições da IMU diretamente às posições do elemento terminal desse robô, mapeando posições programadas com as efetivamente ocorridas.

Quando a IMU se torna capaz de calibrar o robô com precisão, ela pode ser reutilizada para conhecer as posições reais do elemento terminal do robô novamente, indefinidamente e de forma *online*, tantas vezes quanto se queira, compensando assim eventuais desgastes futuros, desvios, variações de temperatura, etc.

Como o que se deseja aqui é calibrar a posição do robô ao invés da sua orientação, antes mesmo de calibrar o robô, deve-se ou criar uma metodologia ou algoritmo que mitigue o comportamento ruidoso da IMU, a fim de que a mesma seja capaz de produzir medições mais precisas de posição ao longo do tempo. Levando-se em conta que existem vários métodos e algoritmos na literatura no sentido de diminuir erros de medida de posição calculados por IMUs, o que se propõe aqui é chegar próximo de valores sub milimétricos de precisão entre o valor real e o valor previsto pela rede, o que traria precisão necessária para possibilitar a calibração do robô.

Sendo assim, para ser eficaz do ponto de vista prático, a metodologia aqui empregada não usa, *a priori*, o desenvolvimento de um modelo cinemático cujos parâmetros representam com precisão o robô real, mas sim o uso dos valores medidos numa IMU como entrada  $x_k$  da rede neural RNA e os valores medidos no controlador do robô como saída esperada  $y_k$  dessa mesma rede, para primeiro calibrar a IMU.

Apenas após a calibração da IMU, esta poderia ser usada junto ao modelo cinemático do robô para calibrá-lo e comparar seus resultados. Usa-se aqui uma RNA tipo recorrente com *backpropagation*, onde os valores de saída são reinjetados na entrada, que foi descrita em maior detalhe no seu capítulo próprio: RNAs. A Figura 3.1 exemplifica tal algoritmo, quando utilizando-se de retro-propagação.

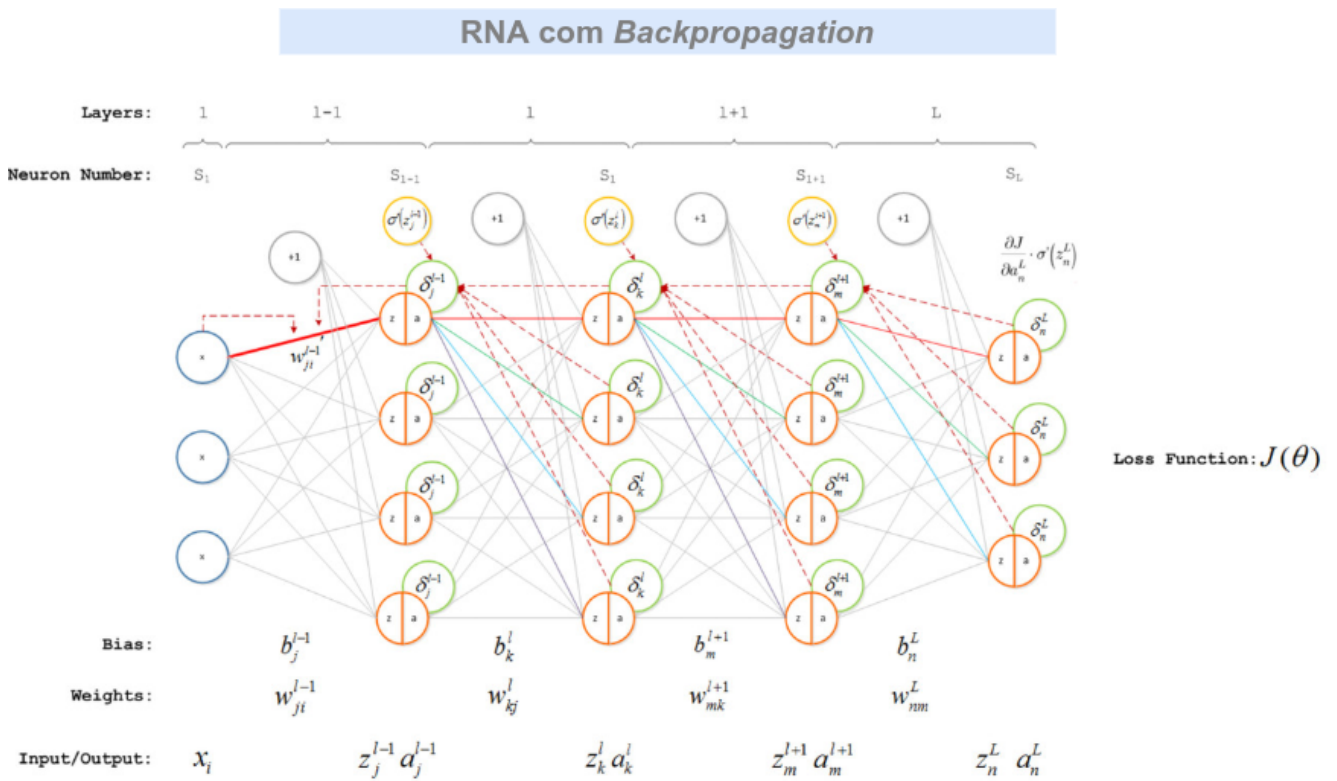


Figura 3.1 – RNA recorrente com *backpropagation*. Fonte: [Chen 2017]

Com base na convergência dos valores dos pesos  $w$  dos neurônios existentes na RNA, os comportamentos de variáveis ainda desconhecidas existentes na IMU, que implicam em seus desvios, são mapeados e os alvos para reinicialização do referencial são detectados. A fim de que a mitigação de erro do IMU ocorra e a mesma seja capaz de calibrar posições futuras do robô em questão, os pesos  $w$  dos neurônios da RNA devem fazer o papel equivalente ao dos coeficientes de uma equação de ajuste de erro, utilizada na calibração do modelo cinemático do robô.

Se essa equação (regressão) equivalente fosse usada para calibrar o modelo cinemático de



um robô, dificilmente haveria mais parâmetros cinemáticos do que aqueles existentes no seu modelo, uma vez que os pesos  $w$  de uma RNA, que representam essa correlação, podem ser maiores que o número de parâmetros. Por exemplo, enquanto os coeficientes de uma equação linear permaneceriam na faixa de 101, os pesos  $w$  da rede neural (não linear) poderiam estar na faixa de 106.

Sabendo-se que o sinal do sistema IMU se deteriora com valores crescentes em função do tempo (detalhado no capítulo IMUs), uma possibilidade testada por pesquisadores para mitigá-la, foi reiniciar a medição em um intervalo pequeno que impedisse seu incremento [Zarita e Ong 2009], colocando as IMU's nos sapatos de pedestres voluntários, e empregar um algoritmo que zerava a posição  $Z$  – altura do sapato do pedestre em relação ao piso, toda vez que o acelerômetro detectava um impacto do pé contra o solo. Esse experimento, ao calcular o comprimento médio do passo do voluntário, conseguiu suavizar o progressivo aumento do erro, mas ainda sofria de baixa exatidão, já que a mesma se encontrava na faixa de metros ou dezenas de metros e não se avaliavam valores nos outros eixos, preenchendo-os com as estimativas médias de comprimento dos passos.

No caso da presente pesquisa utilizou-se também uma reinicialização de referencial, porém contando com o auxílio de uma Rede Neural Artificial que pudesse “aprender” os pontos corretos de reinício a fim de subsidiar o aumento de exatidão necessária à calibração do robô.

As formas da RNA para aprimorar a exatidão seriam:

1. Aprender e mapear variáveis ocultas de funcionamento na IMU que caracterizam sua medição de movimento e reiniciá-las em condições discretas pré-determinadas. Essas variáveis estariam implícitas nos pesos  $w$  da RNA e conseqüentemente não seriam passíveis de visualização;
2. Aprender e mapear as próprias variáveis da IMU, tornando algumas dessas variáveis discretas na topologia da rede tratando-as com entradas  $xk$  realimentadas na RNA recorrente como:
  - a Inversões de sua derivada (caso do sapato dos pedestres) em todos os eixos;
  - b Degraus positivos ou negativos;
  - c Rampas positivas ou negativas;
  - d Paradas (fim de curso);
  - e Saídas;
  - f Pontos de máximo e mínimo;
  - g Tempos de existência;
  - h Correlações entre suas posições no movimento guiado, etc.

Após a obtenção dos valores medidos, a RNA pode comparar o comportamento desejado com o comportamento real do sistema a cada iteração e, em sequência, estimar o funcionamento de suas variáveis implementando a detecção dos pontos alvo e “aprendendo” a reiniciar o sistema na posição de mínima energia do espaço de ruído. Chamamos a MLP introduzida de ANN-RB.

A proposta consiste em colocar a IMU na ponta da ferramenta de um robô industrial, gerando um referencial estável e “reiniciável” nos pontos alvo, onde os valores oferecidos pelo sistema possam ser comparados com os desejados, tirados do referencial (Figura 3.2).

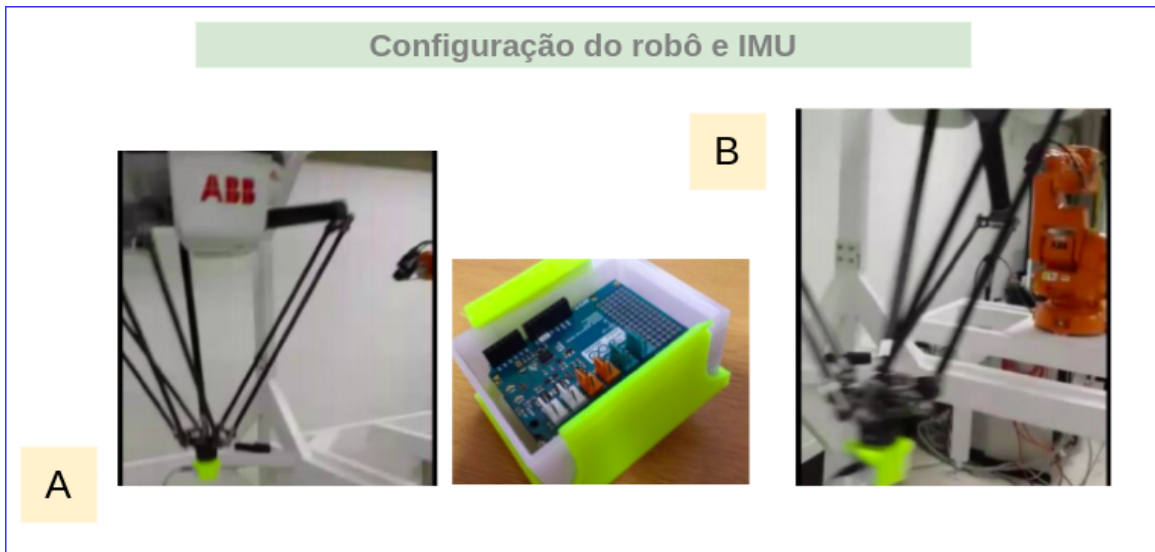


Figura 3.2 – IMU acoplada ao robô (A) e produzindo medições em movimento guiado (B).

Os pontos alvo não são necessariamente pontos de parada como no caso do impacto com o solo dos IMUs para pedestres, mas sim eventuais pontos em que a RNA “aprenda” como relevantes para minimizar sua equação de erro. Poderia ser um ponto de curva, um ponto de inversão de aceleração ou inclinação, de aceleração ou inclinação máxima ou mínima, um fim de curso, etc. Não há como saber, pois a RNA não mostra e nem depende destes pontos, mas sim de um conjunto de coeficientes vetoriais que se farão presentes nos seus pesos e que só são válidos quando considerados em conjunto. A Tabela 3.1 ilustra parte do conjunto de dados que foram obtidos no experimento.

Tabela 3.1 – Exemplo da Organização dos Dados

ALX	ALY	ALZ	MX	MY	MZ	GX	GY	GZ	QW	QX	QY	QZ	AX	AY	AZ	AGX	AGY	AGZ	X	Y	Z
0,07	0,15	-0,12	0,30	-0,25	0,50	-0,10	-0,04	-0,16	-0,44	0,98	1,00	-0,05	0,04	0,01	-0,09	-0,86	-0,99	1,00	-0,98	-0,75	1,00
0,05	0,15	-0,12	0,45	0,03	0,93	-0,10	-0,03	-0,17	-0,44	0,98	1,00	-0,05	0,03	0,00	-0,09	-0,86	-0,99	1,00	-0,34	-0,26	1,00
0,11	0,05	-0,16	0,37	-0,10	0,40	0,07	-0,01	-0,23	-0,43	0,97	0,97	-0,05	0,10	0,02	-0,05	-0,89	-0,98	1,00	0,00	0,00	1,00
0,45	-0,25	-0,06	0,37	-0,27	0,33	0,07	0,02	-0,21	-0,43	0,96	0,95	-0,05	0,43	-0,36	-0,03	-0,89	-0,95	0,99	-0,45	-0,45	1,00
0,00	0,04	-0,17	0,63	-0,44	-0,03	0,10	-0,17	-0,19	-0,43	0,96	0,96	-0,06	-0,03	-0,09	-0,14	-0,89	-0,96	1,00	-1,00	-1,00	1,00
-0,01	0,18	-0,15	0,63	0,03	0,65	-0,18	-0,05	-0,14	-0,43	0,97	0,97	-0,06	-0,02	0,04	-0,13	-0,88	-0,96	1,00	-0,53	-1,00	1,00

A IMU forneceu 19 saídas para a rede, ou seja, recursos foram obtidos a partir da fusão dos sinais dos sensores que o hardware disponibilizou e são os seguintes [Campos e Motta 2021]:

- Três entradas relacionadas à orientação  $(o_x, o_y, o_z)$  baseada em uma esfera de 360 graus;
- Vetor de velocidade angular com três eixos  $(wa_x, wa_y, wa_z)$  representado por três características;
- Três *features* referentes ao vetor de aceleração angular nos eixos  $(\alpha_x, \alpha_y, \alpha_z)$ ;
- Campo magnético nos três eixos  $(M_{(x)}, M_{(y)}, M_{(z)})$  representado por três entradas;
- Três características estão relacionadas à aceleração linear nos eixos  $(a_x, a_y, a_z)$ .
- Gravidade nos três eixos  $(g_x, g_y, g_z)$ ;
- A temperatura  $T$  em graus Celsius.

A rede neural artificial dá como saída uma matriz de pesos  $w$  representando seus neurônios, que, nesse caso, nada mais são do que uma correlação não linear dos valores de entrada com os da saída.

A identificação das variáveis que governam o sistema analisado – no caso a IMU, fica escondida de forma distribuída nos pesos  $w$  dos neurônios e não há como rastreá-las. Entretanto, o que se quer de fato não é identificar o comportamento individual de cada variável do sistema, mas sim conhecer a matriz de compensação geral que venha mitigar a progressiva perda de precisão do mesmo.

Finalmente, com essa matriz já treinada seria possível prever a posição do robô a partir dos valores lidos na IMU, sem observar os ruídos crescentes oriundos da mesma. Dessa forma, a IMU poderia ser usada para calibrar o modelo cinemático do robô no laboratório e, obtendo sucesso, poderia ser usado tantas vezes quanto se queira.

### 3.1 Preparação dos dados

Como discutido, as medidas efetuadas pela IMU não são precisas, pois o erro associado tende a crescer com o tempo. Uma das maneiras de mitigar esse problema é através da preparação adequada dos dados, que através de uma técnica de software chamada *Feature Scaling* normaliza as variáveis independentes em uma determinada faixa de valores. Como os dados são ruidosos, isso implica que o intervalo de valores bruto varia muito e assim não permite o funcionamento adequado da função objetiva de muitas redes. Geralmente a distância entre dois pontos é medida pela distância euclidiana, assim qualquer *feature* que apresentar uma

faixa de valores elevada terá seu peso elevado no processo de treinamento. Normalizar os dados permite que cada entrada apresente um grau de contribuição aproximado, assim, apesar da variação, o sistema pode relacionar melhor os dados para descrever o modelo. Além disso, o método do gradiente converge de maneira muito mais rápida utilizando-se da normalização [Ioffe e Szegedy 2015]. A Figura 3.3 mostra diferentes formas de normalizar as entradas que este estudo testou.

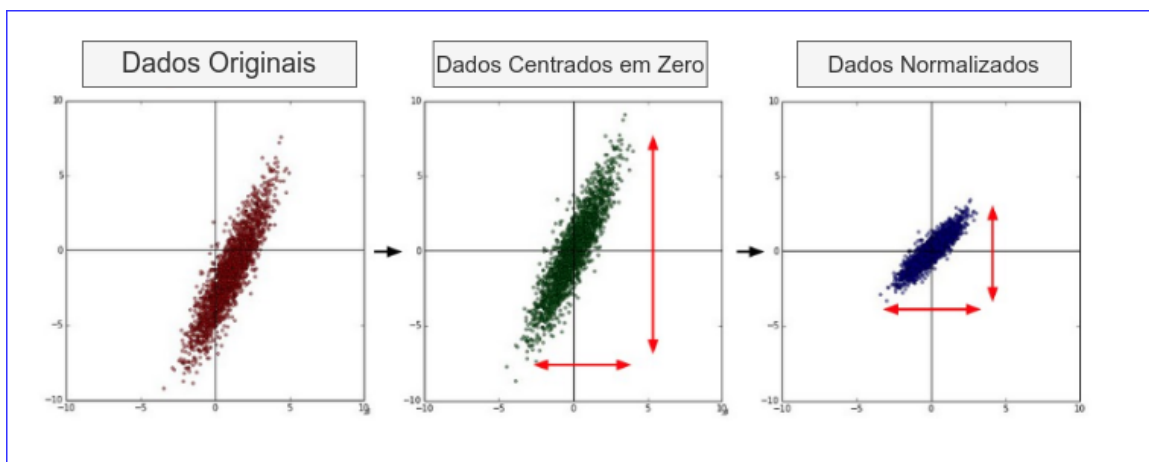


Figura 3.3 – Distribuição dos dados de acordo com a normalização. Fonte: [Santos 2018]

Existem muitos métodos de normalizar a entrada. Pode-se citar a normalização padrão que subtraia a média da amostra de cada valor e divide pelo desvio padrão, assim é possível estabelecer uma faixa específica cujos os intervalos devem ser sempre respeitados. As técnicas são variadas e sendo assim é possível incluí-las no planejamento fatorial. Por exemplo, a Figura 3.4 mostra três recursos chamados ALX, ALY e ALZ que foram devidamente normalizados no intervalo entre 1 e -1. Tais características representam os dados obtidos pela IMU referente ao acelerômetro nos três eixos. O presente estudo também considerou diferentes formas de normalização em sua metodologia.

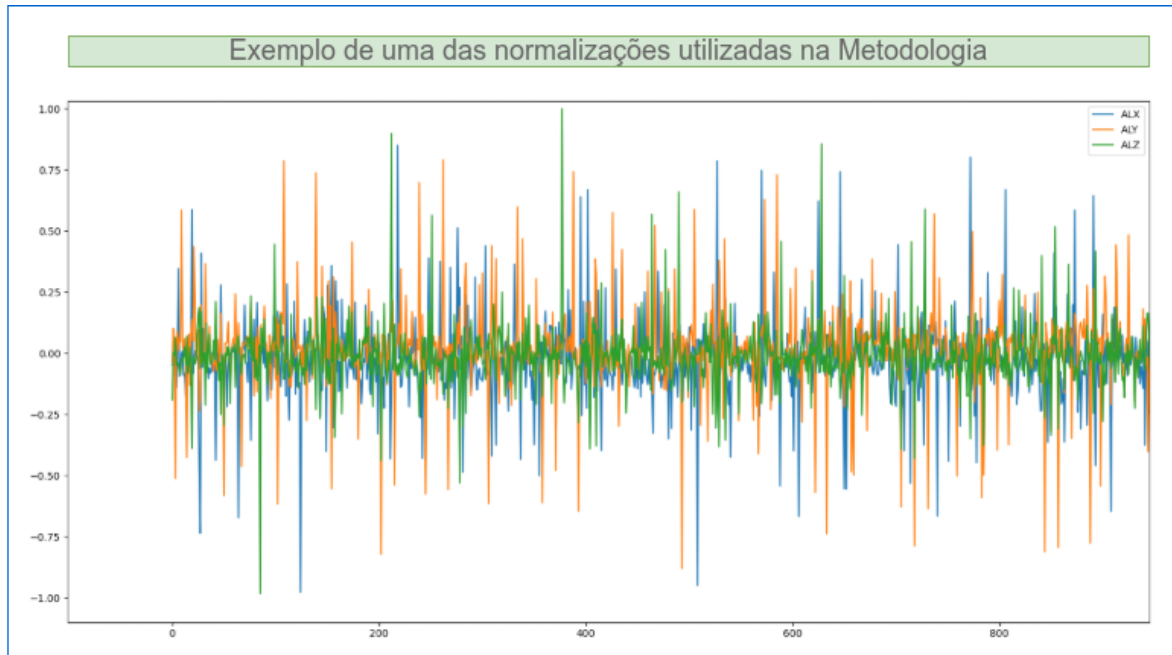


Figura 3.4 – Exemplo de Normalização

### 3.1.1 Processo de Treinamento

O sobre-treinamento (*overtraining*) [Ying 2019, Allamy 2014] ou treinamento excessivo ocorre quando a rede obtém uma solução que é muito específica. Pode-se fazer uma analogia com um aluno que resolve qualquer questão, desde que seja previamente conhecida, de forma que o sistema de regressão não é capaz de prever valores fora de uma faixa conhecida. No segundo caso, a rede é treinada de maneira insuficiente, o que leva a uma resolução pobre, mesmo utilizando valores conhecidos. Para que o sistema não incorra nessas duas adversidades é necessário que o mesmo seja passível de verificação, ou seja, é necessário apresentar os dados à rede de forma que seja possível verificar a consistência de operação em uma faixa desconhecida. Para alcançar tal objetivo é suficiente particionar a entrada e em três grupos distintos de conjuntos, respeitando-se uma porcentagem previamente definida. Tal porcentagem não possui uma faixa específica, porém é de costume deixar a maior parte para a bateria de treinamento e cada sistema possui sua própria faixa de valores (geralmente definida pelo projetista e baseada na sua experiência). Dessa forma os dados serão divididos como segue:

- **Treinamento:** o experimento utilizou 70% dos dados no processo de treinamento;
- **Validação:** a validação acontece durante a fase de treinamento e é usada pela rede para possibilitar um verificação da qualidade dos resultados obtidos e assim ajustar o modelo (15%).
- **Teste:** esses dados são aparentados à rede após o treinamento e são fundamentais para

definir se o modelo é robusto. A pesquisa considerou 15% dos valores medidos na etapa de teste.

Considerando a Figura 3.5, onde apenas a dimensão  $x$  é plotada, pode-se dizer que o primeiro modelo está sobre-treinado, pois é incapaz de fazer previsões fora da faixa de valores conhecidos. Já o segundo teste mostra que a rede está sub-treinada, apesar de suas previsões serem genéricas a qualidade é ruim, ou seja, possui uma acurácia baixa.

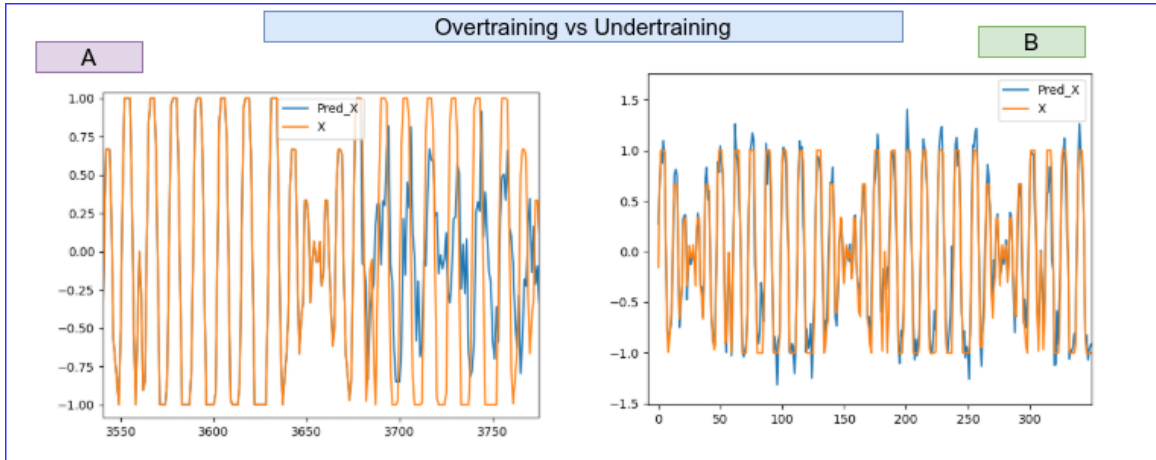


Figura 3.5 – Exemplo de sobre-treinamento (A) e sub-treinamento (B)

### 3.1.2 Validação Cruzada

O protocolo de validação cruzada (adequado para séries temporais) adota o conjunto de validação para verificar a qualidade dos resultados e selecionar o melhor modelo possível. Finalmente, o desenvolvedor pode usar os dados de teste para visualizar o resultado e compará-lo com a saída esperada.

Como há uma relação temporal entre as amostras, não é possível randomizar os dados e depois dividi-los em um conjunto de treinamento e teste (a divisão aleatória ou K-Fold pode não fornecer bons resultados). Os dados são tratados como uma série temporal e, portanto, a metodologia empregou uma técnica de validação conhecida como validação cruzada contínua [Campos e Motta 2021], que retorna as primeiras  $k$  dobras como um conjunto de treinamento e a  $(k + 1)$  enésima dobra como um conjunto de testes [Campos e Motta 2021]. O método é uma variação do K-Fold adequado para conjuntos de dados que possuem um relacionamento temporal. A metodologia treinou e validou a rede ANN-RB em um conjunto de dados com diferentes trajetórias e direções variadas e testou o modelo obtido no conjunto de teste. Os outros algoritmos foram treinados e avaliados em uma série com 3825 amostras e testados com 4500 entradas. A Figura 3.6 mostra o desenho geral usado na fase de preparação de dados. A Figura 3.7 ilustra como a metodologia organizou as divisões do conjunto de treinamento e validação [Campos e Motta 2021].

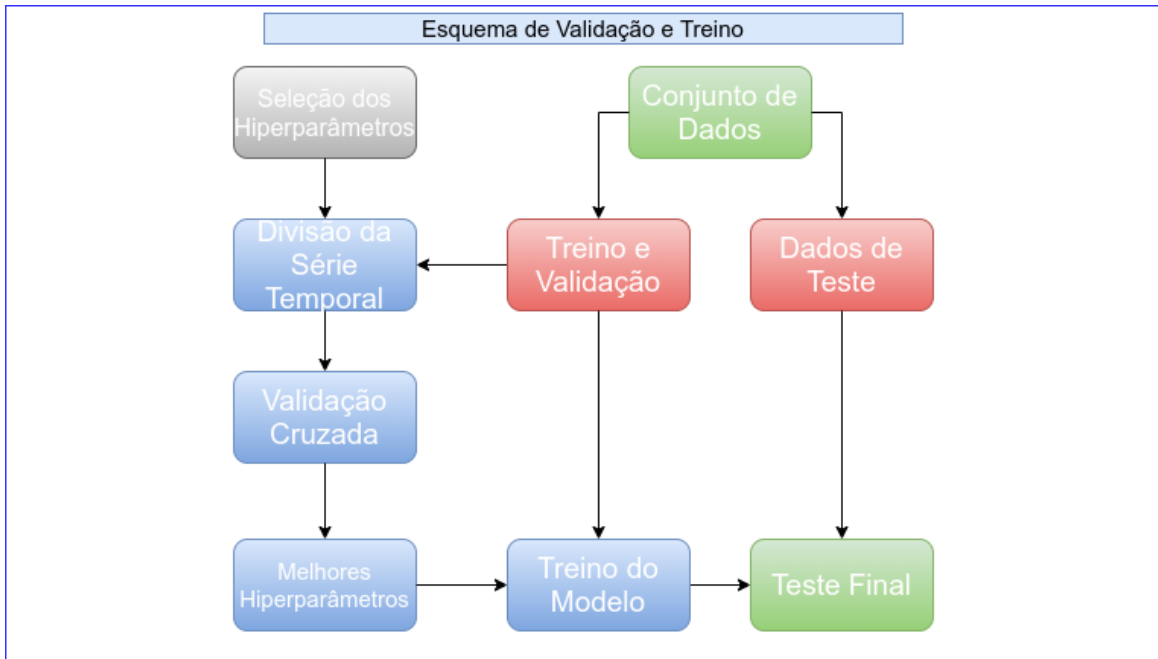


Figura 3.6 – Esquema de divisão de dados, validação cruzada e teste. [Campos e Motta 2021]



Figura 3.7 – Validação cruzada adota pela metodologia no processo de treinamento [Campos e Motta 2021].

### 3.2 Comparação com outras Topologias

Nesta pesquisa foram utilizados três outros algoritmos como critério de comparação, porém nesse caso não será permitido à rede o acesso a todos os dados. Em vez disso uma parcela bem menor será utilizada, para que os treinos sejam mais rápidos. Como os testes levaram

quase um mês seria inviável repetir em diferentes configurações e algoritmos. O objetivo aqui é verificar a qualidade dos resultados e validar a faixa de operação que permitiu alcançá-la. Dessa maneira, o trabalho não visa desqualificar os outros métodos em detrimento do que está sendo introduzido [Campos e Motta 2021] e sim usá-los para fortalecer a ideia da utilização conjunta de IMU's e redes neurais para a calibração de robôs. É importante lembrar que apesar dos tempos de treinos elevados, uma vez que o modelo foi criado a sua replicação e seu tempo de predição são imediatos.

Com um processo de treinamento mais rápido é possível descobrir os limites da rede e qual seu ajuste ótimo para esse tipo de problema. Ou seja, pode ser que não seja necessário esperar tanto tempo para obtenção do modelo funcional. Assim, se os algoritmos usados durante esses testes apresentarem resultados satisfatórios com um número reduzido de pontos, significa que é possível reduzir a dimensão do problema e assim desenvolver uma maneira mais rápida e objetiva de treino. Ao invés de fornecer 6 milhões de pontos a cada algoritmo de regressão serão utilizadas apenas 4500 amostras. Para isso os seguintes métodos serão empregados:

- Perceptron Multicamadas: Mesma arquitetura de classificação proposta por esse trabalho, porém mantida por terceiros [Scikit-Learn 2019] e treinada em um conjunto menor de dados;
- Rede neural Convolutacional: Utilizada em problemas de classificação e regressão, esse tipo de rede está sendo usada em diversas aplicações e apresentou ótimos resultados em diferentes domínios. Por isso esse trabalho optou por testar a sua aplicabilidade no caso desta pesquisa;
- XGBoost Regressor: Muito utilizados em sistemas de regressão o método é eficiente em problemas similares.

A implementação de tais métodos foi feita na Linguagem Python, pela facilidade de programação e quantidade de referência disponível.

A Figura 3.8 resume graficamente o método desenvolvido [Campos e Motta 2021].

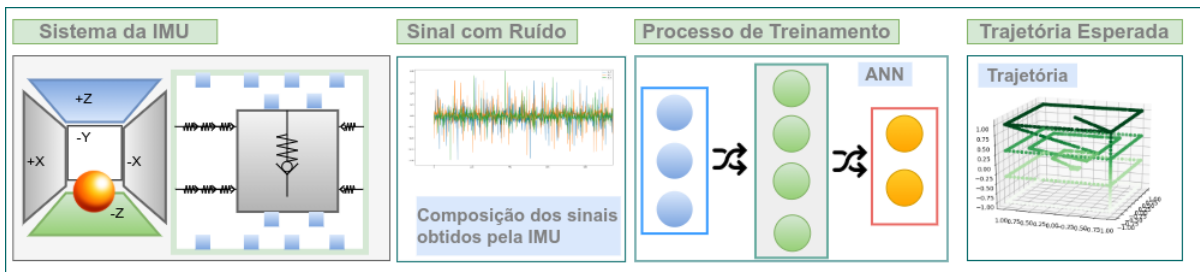


Figura 3.8 – Esquema geral da abordagem introduzida por este trabalho.



### 3.3 Materiais

Pela experiência acumulada no estudo das RNAs, sabe-se que, onde se usa *backpropagation steepest descent* ocorre alta probabilidade de convergência, porém o sucesso dessa convergência é proporcional ao número de pontos de análise colhidos no experimento e ao poder computacional associado aos cálculos matriciais recorrentes. Para permitir a suficiência computacional necessária ao experimento o seguinte hardware foi utilizado:

1. Doze placas Nvidia GTX 1070 com 1920 núcleos CUDA (núcleos para cálculo numérico de matrizes), totalizando 23040 núcleos;
2. Dois processadores XEON de 24 núcleos de CPU, com 64 gigas de memória e refrigeração líquida;
3. Estação de trabalho com HD SSD de 1 terabyte.

O objetivo foi montar um *cluster* de processamento paralelo em GPU (graphic processing unit) que permitisse alta computação. O *cluster* foi montado de acordo com a Figura 3.9.



Figura 3.9 – Estação de trabalho e placas GPU com mais de 23000 núcleos CUDA.

Além do poder computacional do hardware adquirido, também foi esperado que os acelerômetros tivessem a possibilidade real de capturar o movimento necessário ao cálculo, sem muitos desvios, pois já se sabe que isto depende de seu número e posicionamento [Vieler et al. 2016] além da resolução do próprio equipamento. Isso foi levado em conta na escolha do tipo de IMU para que o dispositivo ofereça a resolução necessária. Nesse sentido pareceu razoável que ocorresse a convergência da RNA levando a uma boa correlação entre o valor medido na IMU e o valor descrito no controlador do robô, o que pode ser comprovado a partir do sucesso da calibração inicial do mesmo.

## Capítulo 4

# Resultados

Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados.

---

Mahatma Gandhi

Este capítulo apresenta os resultados das três baterias de testes, que simulam trajetórias diferentes e com processo de treinamento distinto, considerando-se a rede implementada pelo autor. Em seguida o texto estende o trabalho e faz um estudo comparativo com outras redes neurais de uso geral, porém considerando uma amostra que representa apenas 0.075% do conjunto original. A análise visa verificar até que ponto a quantidade de informação é essencial para o modelo introduzido, estudar qual rede é capaz de entregar a técnica de maneira mais acurada e rápida, e comparar as características dos algoritmos. Ressalta-se que um dos métodos aplicados seria a calibração da rede, visando a mais alta acurácia possível. Ou seja, deseja-se minimizar os erros entre a distância real e a prevista após o processo de treinamento.

A metodologia para a conexão e programação do movimento guiado no robô, através do software de simulação e controle fornecido pelo fabricante do mesmo, encontra-se no Apêndice, assim como seu código fonte. O programa gerou vários conjuntos de trajetórias para o robô e os valores de medição da IMU, acoplada em seu elemento terminal foram utilizados como entradas na rede neural. A Figura 4.1 exemplifica a geração das trajetórias.

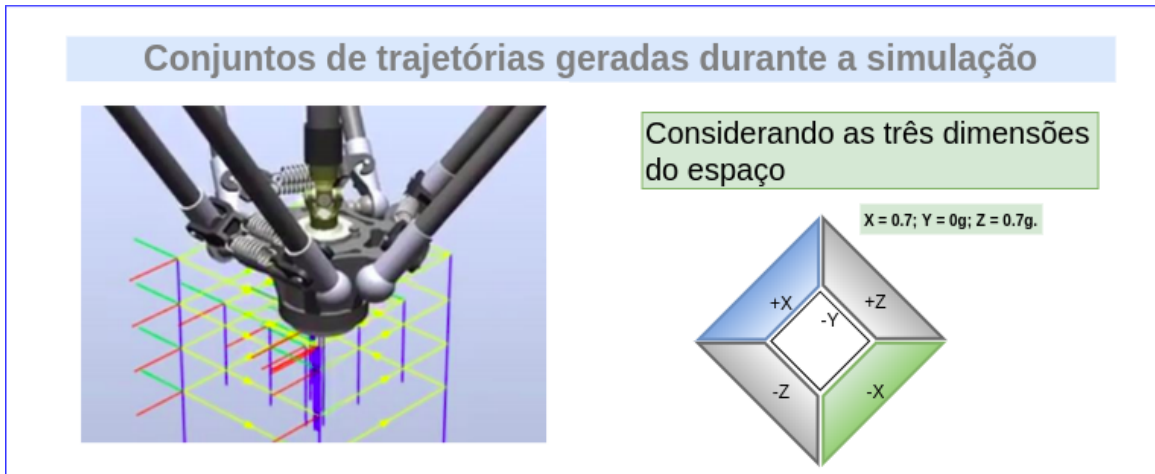


Figura 4.1 – Trajetórias geradas, programadas no robô.

## 4.1 Rede Neural Artificial Proposta

Os primeiros resultados (antes do aprendizado da ANN-RB), obviamente foram insatisfatórios, com imensos valores de erro entre a posição aferida pela IMU e os valores no controlador do robô, como visto na Figura 4.2, onde os pontos em vermelho, no centro da Figura, mostram o que a IMU reconheceu como posição do elemento terminal do robô e a linha em azul representa a trajetória lida no controlador do robô, realizadas pelo seu elemento terminal. Isso demonstra que a RNA introduzida ainda não havia “aprendido” a corrigir os valores provenientes dos erros da IMU e desse modo seria necessário que fossem feitas mudanças nos hiper-parâmetros da rede neural e em sua topologia.

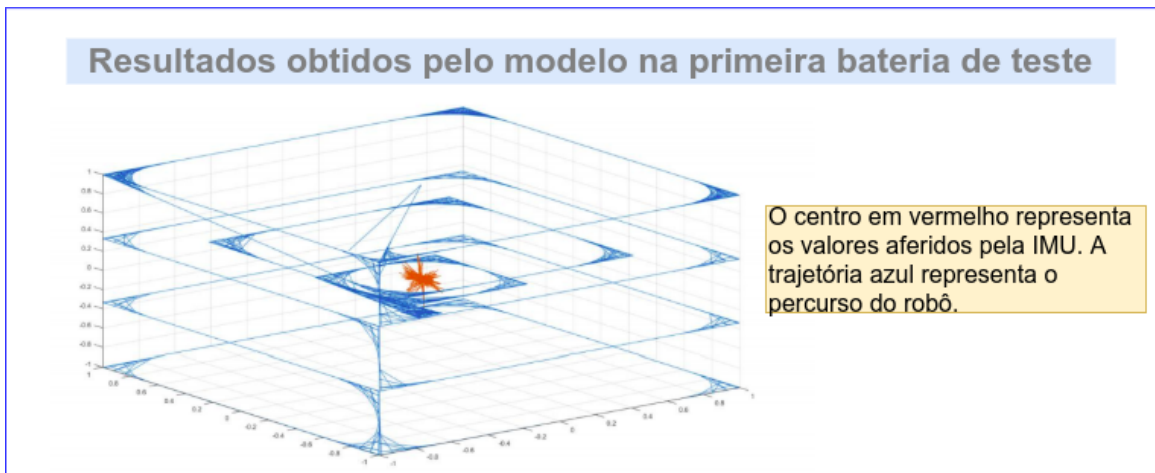


Figura 4.2 – Valores da IMU (vermelho) X valores do Controlador (azul).

Após as modificações na topologia da RNA e seus hiper-parâmetros sugeridos pelo algo-

ritmo de planejamento fatorial (descrito no capítulo de redes neurais) e inúmeras épocas de aprendizado [Campos e Motta 2021], seus valores de saída começam a convergir para valores próximos dos pontos do controlador do robô. Nas Figuras 4.3 e 4.4 os pontos previstos pela RNA através dos 19 sinais entregues pela IMU já são próximos dos valores do controlador do robô e praticamente se confundem com os mesmos. A trajetória do robô (azul) foi praticamente idêntica ao resultado obtido pela IMU. A acurácia diminui quando se considera uma trajetória circular. Mesmo assim, a diferença ainda é pequena

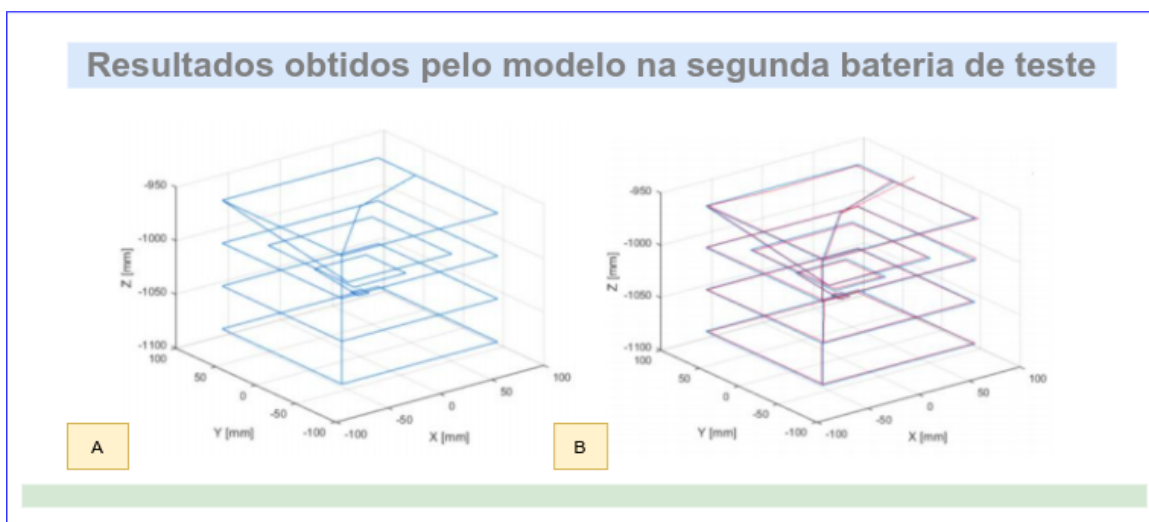


Figura 4.3 – Comparação entre valores do controlador do robô (A) e IMU (B)

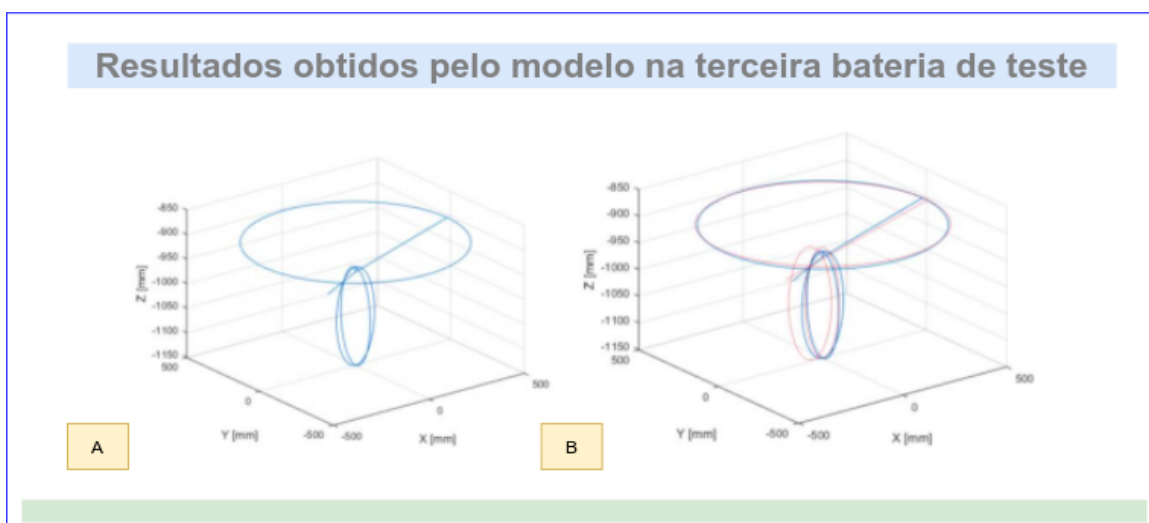


Figura 4.4 – Comparação entre valores do controlador do robô (a) e IMU (b).

O método apresentou erro máximo de 0.9 mm. De acordo com o manual do IRB-360, a precisão do TCP é de 0,04 mm, então o erro máximo da abordagem é  $(0,9 \pm 0,04)$ . O Erro Quadrático Médio (MSE) foi  $(0,68 \pm 0,21)$  mm e a Raiz do Erro Quadrático Médio foi  $(0,82$

$\pm 0,46$ ) mm. O erro quadrático médio (MSE) de um estimador mede a média dos quadrados dos erros - ou seja, a diferença quadrática média entre a estimativa e o valor esperado.

O experimento avaliou abordagens em diferentes trajetórias, como subidas, rampas e círculos nas três coordenadas. Assim, o sistema inteligente pode realizar a navegação estimada do TCP do robô a partir dos dados fornecidos pela fusão dos sinais dos sensores do IMU.

Assim o sistema inercial mantém sua calibração e está apto para ser usado em novas posições do espaço de trabalho do robô paralelo IRB-360, operando em outros volumes de teste para conferir sua calibração. Nos volumes finais analisados os erros mostraram uma distribuição gaussiana em que os valores de erro, num intervalo de 6,2 milhões de medições, ficaram limitados a 0,9 mm, conforme a Figura 4.5:

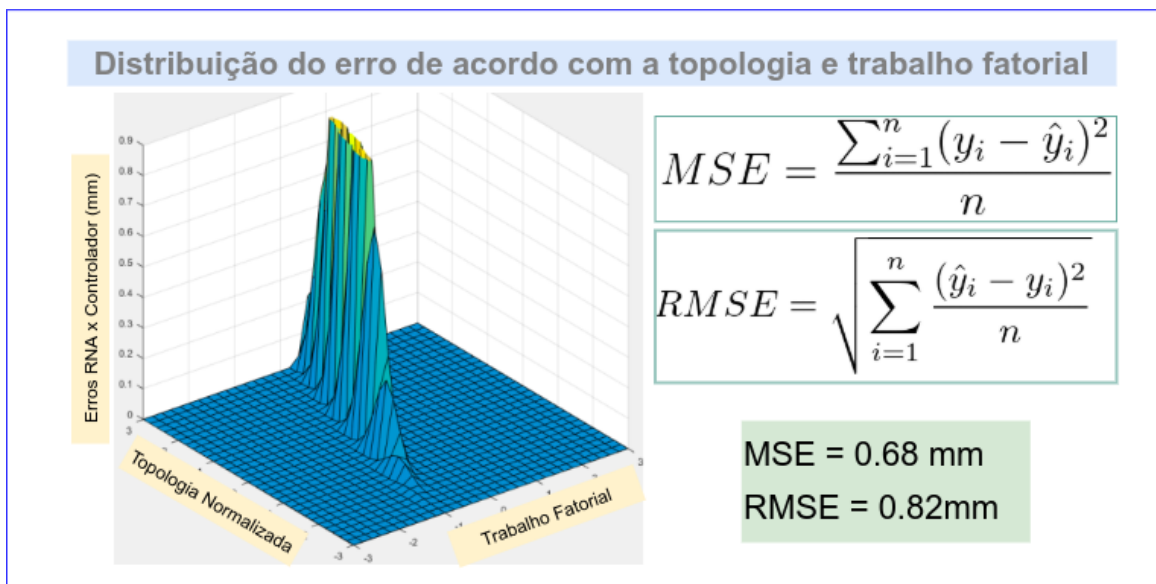


Figura 4.5 – Distribuição dos valores de erros medidos.

A abordagem foi capaz de prever a posição do robô usando os dados obtidos da IMU junto à rede neural proposta, com alta acurácia e desvio submilimétrico, tendo como única desvantagem aparente do modelo a necessidade de um grande volume de dados, o que resultou em um treinamento de 20 dias [Campos e Motta 2021]. Por isso, as próximas seções irão investigar a possibilidade de generalizar a pesquisa para um espaço de resolução menor e utilizando-se de algoritmos conceituados no meio acadêmico e comercial.

## 4.2 Algoritmo Perceptron Multicamadas

A rede MLP do [Scikit-Learn 2019] foi ajustada em diferentes configurações, obedecendo-se a mesma regra do trabalho fatorial empregada no desenvolvimento da metodologia do autor.

O algoritmo MLP apresentou resultados interessantes com Erro Quadrático Médio de 2,81 mm e Erro Máximo de 135,2 mm. O processo de validação conhecido como validação cruzada melhorou os resultados, e o caminho descrito na Figura 4.6 se aproximou da trajetória real feita pelo software do robô. O processo de treinamento durou aproximadamente 5 minutos [Campos e Motta 2021]. Diferentes configurações não alteraram muito o resultado, o que indica que o principal fator limitante seja o volume de dados.

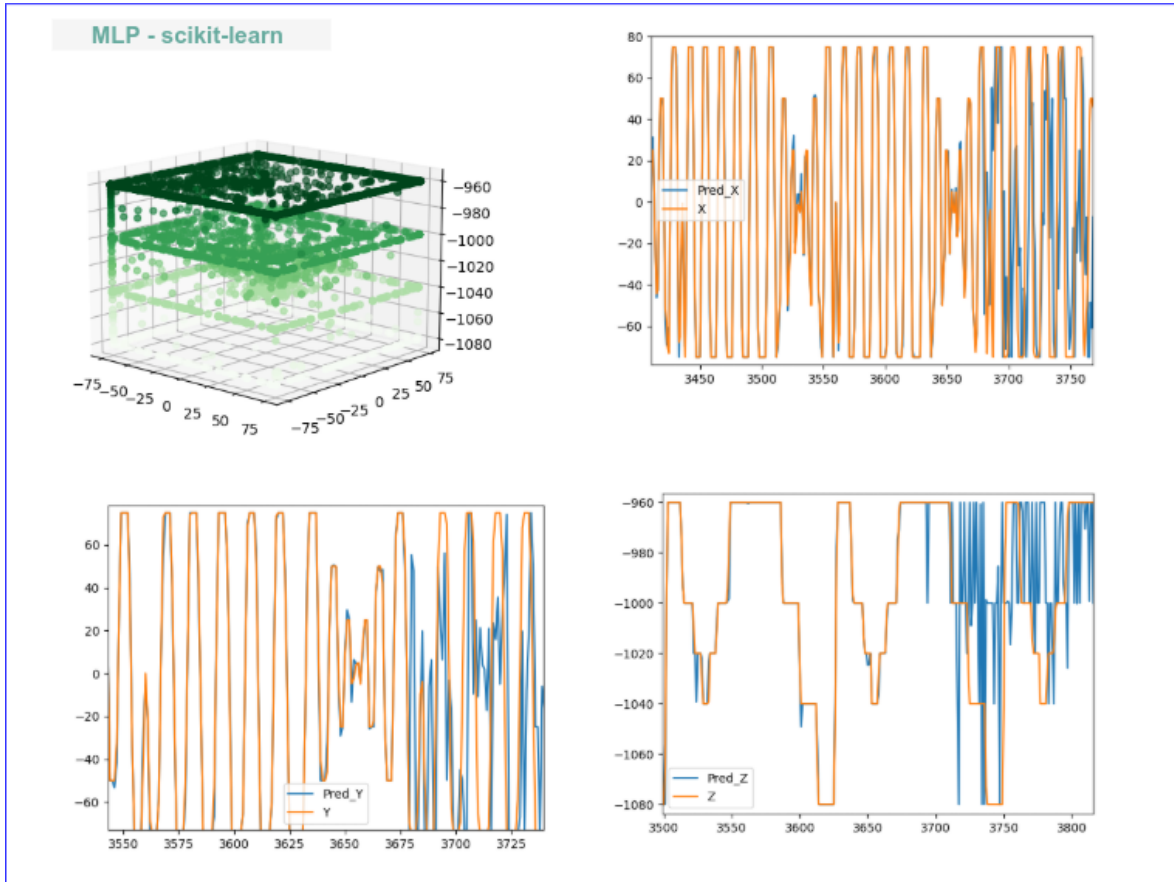


Figura 4.6 – Resultado obtido por uma das redes MLP que foi testada.

Os resultados obtidos foram parcialmente satisfatórios, o tempo de treino ficou na faixa dos minutos, o que sugere que é possível encontrar uma configuração que permita treinos mais rápidos, porém a qualidade dos resultados regrediu. Hipoteticamente, um treino mais demorado levaria a soluções mais complexas e satisfatórias (como no caso da abordagem apresentada). Além disso, os resultados alcançados na porção de testes são inferiores comparados com aqueles obtidos na fase de treino e validação. Considerando-se a configuração com três camadas escondidas, cada uma com 100 neurônios, taxa de aprendizado igual a 0.001 e com um processo de treino de 5000 épocas, o estudo obteve os resultados da Figura 4.6 [Campos e Motta 2021].

Tabela 4.1 – Esquemático das Camadas e Parâmetros

Camada	Saída	Parâmetros
Conv 1D	(None, 16, 256)	1280
Conv 1D	(None, 13, 128)	131200
Conv 1D	(None, 10, 64)	32832
Flatten	(None, 640)	0
Dense	(None, 64)	41024
Dense	(None, 1)	65

### 4.3 Rede Neural Convolutacional

A rede CNN usada contou com a implementação através do Keras e de modo análogo, diversas configurações de seus hiperparâmetros foram testadas, sem muita variação entre elas. O experimento utilizou uma faixa de épocas entre 5 mil e 15 mil, o que não afetou os resultados. A Tabela 4.1 apresenta uma das configurações que foi avaliada no experimento, no início dos testes muitas arquiteturas desse algoritmo foram avaliadas e os resultados não variaram muito. A escolha desse arranjo aconteceu pois, entre todos os teste, foi a que obteve a melhor precisão.

O algoritmo não foi acurado em nenhuma das três dimensões. A rede CNN além de apresentar resultados inferiores à MLP, também gastou mais tempo no processo de treinamento e não foi capaz de fazer previsões fora do espaço de trabalho conhecido. Isso sugere que apesar da complexidade do algoritmo o mesmo não é o mais viável para esse problema. Os resultados estão na Figura 4.7

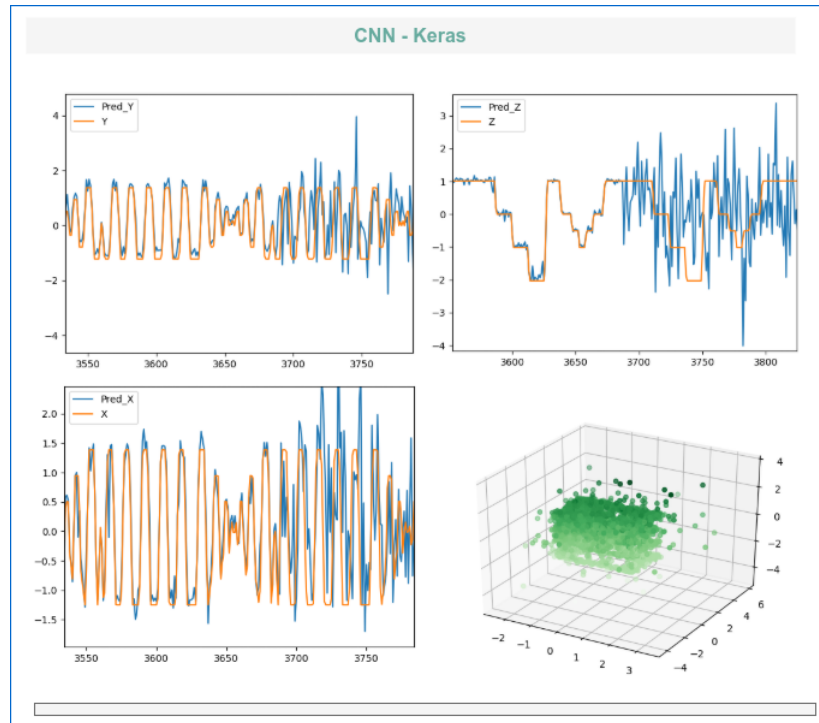


Figura 4.7 – Resultados obtidos pelo algoritmo CNN.

#### 4.4 Algoritmo XGBoost Regressor

O último algoritmo analisado foi o que obteve os melhores resultados entre os três, porém como tal rede se baseia em árvores de decisão, não é possível afirmar que os dados obtidos na fase de teste representem os melhores resultados, já que tal rede é muito suscetível ao excesso de treinamento. Com relação ao tempo obtido pode-se afirmar que não passaram de alguns segundos, o que era de se esperar, pois a referida rede foi projetada justamente para alcançar os menores tempos de execução. Como no caso da CNN que possui diversas maneiras de ajustar seus hiperparâmetros, o XGBoost também necessita de uma varredura. O experimento considerou diversas configurações e os resultados novamente não variaram muito. A configuração dos hiperparâmetros da rede que apresentou a melhor faixa de valor possível foi a seguinte:

- Base score de 0.5;
- Booster *gbtree*;
- Gamma igual a 0;
- Tipo de Importância baseada no ganho;
- Taxa de Aprendizagem de 0.05;



- Profundidade máxima igual a 10;
- Número de estimadores igual a 3000;
- Função objetiva baseada no erro quadrático;
- Sub-amostragem de 0.9.

A Figura 4.8 mostra os resultados obtidos pelo XGBoost. É possível notar que com dados conhecidos o sistema cumpre fielmente a trajetória nominal do robô, porém quando os valores são novos e estão contaminados com o ruído progressivo da IMU, a rede não consegue manter a mesma acurácia e com isso é possível concluir que o modelo se tornou muito específico ou sensível demais a qualquer variação nas medições. Como a variação dos resultados foi pequena entre os diferentes arranjos desse modelo é possível inferir que a falta de dados foi o fator limitante.

Como as três redes foram incapazes de prever de maneira acurada utilizando-se de um conjunto de informação reduzido é razoável pensar em duas abordagens que possam melhorar o modelo. Assim, seria necessário procurar um balanço entre o tamanho da amostra e o tempo de treinamento. Dois caminhos podem ser seguidos nesse sentido:

- Particionamento sucessivo do conjunto de medições: a ideia é realizar os teste de modo incremental, adicionando-se dados de maneira progressiva até que o erro se encontre dentro de uma faixa esperada e tempo de treino seja o menor possível;
- Reduzir o ruído da IMU: para isso seria necessário utilizar outras técnicas, como filtragem e reinício do referencial em um dado intervalo de tempo;

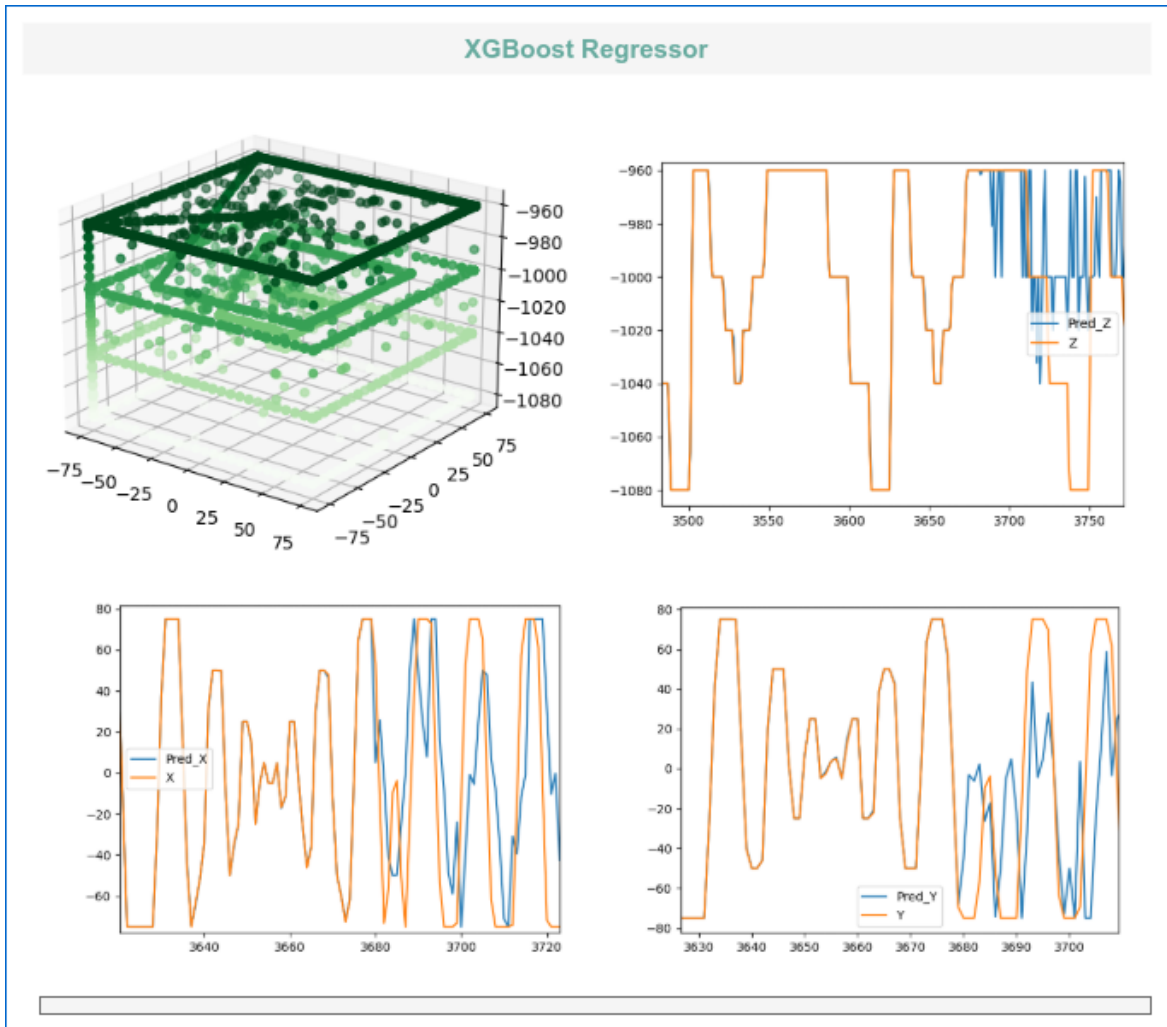


Figura 4.8 – Resultados obtidos pelo algoritmo XGBoost.

A tabela 4.2 mostra o resultado geral do experimento, excluindo-se a rede CNN, já que os resultados da mesma foram pobres. As CNN's são empregadas tipicamente em problemas que envolvem imagens como entrada e talvez por isso o algoritmo não é apropriado para esse tipo de problema. Graficamente, é possível ver que todas as redes passaram a descrever a trajetória. Quantitativamente, os algoritmos que foram treinados em um conjunto de dados limitado apresentaram erros maiores [Campos e Motta 2021].

Tabela 4.2 – Resultado geral

Algoritmo	Erro Máximo (mm)	Raiz Quadrada do Erro (mm)
XGBoost	120.3	2.34
MLP	135.2	2.81
ANN-RB	<b>0.9</b>	<b>0.68</b>

## Capítulo 5

# Discussão

A meta de uma discussão ou debate não deveria ser a vitória, mas o progresso.

---

Joseph Joubert

### 5.1 Discussão Geral

Os resultados alcançados nesta pesquisa demonstram que é viável realizar a navegação estimada do TCP do robô utilizando um método de regressão não linear por meio da fusão de sinais de microssensores. A rede ANN-RB foi capaz de prever com exatidão o caminho na bateria de teste (com um erro máximo de 0,9 mm). O estudo alimentou o algoritmo de modo serial, e o ANN-RB fez a previsão instantaneamente (online). Portanto, a metodologia não considerou o tempo de previsão (por ser insignificante) [Campos e Motta 2021].

A pesquisa atendeu de maneira satisfatória seus requisitos iniciais, que pretendiam fornecer um método de calibração da IMU adotando uma abordagem não paramétrica. Aplicando-se uma solução de software barata que adquire medidas ruidosas é possível relacionar as entradas de maneira não linear em um sistema caixa preta, que leva o conjunto solução para um estado ótimo e sem se preocupar com o relacionamento entre as variáveis. Com os valores aferidos corretamente é possível ajustar o modelo cinemático do robô de maneira recursiva e no seu curso de operação. A proposta de utilizar dados de uma IMU para alimentar a rede neural desenvolvida pelo autor se apresentou viável e passível de melhorias. A viabilidade se configura no sentido de que, a rede conseguiu prever a localização do robô em seu espaço de trabalho tomando como referência seu modelo nominal. A melhoria pode ser entendida como a capacidade de generalização e otimização do tempo de treino do conceito introduzido.

Com o caminho previsto pela rede ao longo do tempo é possível comparar com os dados originais no controlador e calcular o erro. Com esse erro pode-se ajustar o posicionamento

do efetuator e mitigar *online* essa diferença, fazendo com que o sistema sempre fique em sua configuração ótima. A passividade de melhoria acontece com relação a arquitetura geral do sistema e sua modelagem. No início, como o objetivo era um treinamento amplo e considerando diversas possibilidades de planejamento fatorial, forneceu-se à rede um volume bruto de informação. Essa decisão foi tomada com base na premissa de que a quantidade e a qualidade da informação são fatores limitantes para o aprendizado de máquina e por isso optou-se por um treinamento vasto. Da mesma maneira que essa decisão ajudou a cumprir o desafio inicial e a confirmar a premissa principal, ela também apresentou sua desvantagem, com tempo de treino próximo de 20 dias. Apesar disso, não é possível dizer que o modelo é ineficiente, por duas razões:

- É possível otimizar o modelo e descobrir qual sua faixa de operação ideal, evitando assim treinos desnecessários. Além disso, pode-se testar a rede com uma quantidade menor de dados e de maneira incremental descobrir o ponto de equilíbrio entre volume de dados e acurácia;
- Uma vez que o modelo está treinado ele pode ser amplamente replicado. Ou seja, considere que no pior dos casos e após todos os ajustes a rede necessite de 2 dias para realizar seu treino, com isso ao final desse período iríamos ter um produto funcional e barato;
- Outras redes de propósito geral, treinadas de modo mais enxuto, começaram a descrever a trajetória do robô, com tempo de treino na casa dos segundos ou minutos.

Com relação aos erros obtidos, pode-se dizer que foram satisfatórios e sub milimétricos (ANN-RB), o que abre uma gama muito grande de possibilidades. Primeiro, como já mencionado, o modelo pode ser melhorado através de um ajuste fino de parâmetros e meios que anulem o ruído progressivo da IMU. Segundo e mais importante, mesmo que isso não seja possível, diversas aplicações reais que não exigem de tanta precisão podem se beneficiar do conceito apresentado aqui, ou seja, qualquer produto que necessite de calibração espacial ao longo do tempo e cujo ajuste deva ser milimétrico pode empregar a proposta desta pesquisa.

A consequência inicial do trabalho é permitir que uma linha de produção possa ser mais eficiente e evitar paradas desnecessárias. A auto-calibração não paramétrica introduzida por essa pesquisa possibilita de maneira barata esse avanço. Assim, esta tese pode fornecer subsídios para elementos gerais e outros mais sofisticados, que demandam um sistema de navegação guiada de baixo custo.

Considerando um produto geral como um robô de limpeza, que utiliza tal abordagem para mapear sua posição no espaço de trabalho, é possível que o mesmo empregue um sistema adaptativo que evite colisões e assim proteja melhor suas partes internas do impacto e

da vibração desnecessária. Uma vez que o robô mapeasse o ambiente de trabalho nas coordenadas  $x$  e  $y$  de acordo com os impactos captados, seria possível evitar tais pontos sempre que a rede previsse aquela localização.

Um elemento sofisticado que iria se beneficiar destes resultados seria um sistema de navegação e, nesse caso, a precisão poderia ser ainda maior. Primeiro que sistemas de navegação são caros e seria possível utilizar uma IMU mais precisa e menos ruidosa, segundo que o seu posicionamento poderia ter uma margem de erro maior (já que o posicionamento de um submarino é dado em metros e desvios milimétricos não são considerados). Considere um submarino que não possui sinais eletromagnéticos, GPS ou qualquer onda que possa atingir o fundo dos oceanos. A proposta desta tese se aplicaria muito bem nesse caso e ainda poderia ser estendida para um método balístico capaz de rastrear a trajetória de um míssil.

## 5.2 Comparação com outras Redes

Na sequência do estudo criou-se um protocolo comparativo com outros algoritmos de regressão para analisar a viabilidade geral do modelo e sua sensibilidade ao conjunto de informação. A decisão de usar um conjunto reduzido de dados surgiu para permitir testar uma quantidade enorme de coisas em um intervalo curto e assim observar a dinâmica de diversos modelos e suas implicações técnicas. Os pontos principais apresentados aqui são: o método é robusto, cumpre seu objetivo e é sensível a informação.

O conceito é robusto pois as redes usadas para validar o sistema foram capazes de obter resultados preliminares interessantes, mesmo com uma quantidade ínfima de dados, em comparação com a rede criada pelo autor. Todos os algoritmos começaram a dar forma para a trajetória, isso significa que a abstração entre a entrada e saída é passível de ser resolvida por outras RNAs e isso facilita a generalização do método introduzido. Além disso, houve uma queda drástica no tempo de treino.

A sensibilidade à informação já era algo esperado nesse protótipo e se confirmou no estudo comparativo. Assim, é possível sugerir outros cenários de teste e considerar técnicas que permitam um tratamento mais elegante dos dados. Isso pode ser feito através da Engenharia de *Feature*, que entre outras coisas visa:

- Eliminar *Features* redundantes;
- Criar novos *Features* que possam ser úteis no processo de convergência;
- Obter uma espaço de resolução mais objetivo com técnicas como análise principal de componente;
- Extrair a informação de forma mais objetiva e eliminar *outliers*;

- Utilizar um sistema de filtragem por *software* de amortize o ruído;

Dessa forma, em uma etapa futura a esse trabalho, seria interessante ajustar a entrada de maneira mais objetiva e representativa, com isso seria possível reduzir o tempo de treino enquanto a acurácia é mantida ou aumentada.

Os algoritmos avaliados foram testados em diferentes configurações, como no caso do planejamento fatorial. É importante mencionar que a rede CNN apesar ter vários hiperparâmetros modificados, possui uma possibilidade de configurações que vai além do escopo desse texto. Tais redes podem ser ajustadas de diversas maneiras, assim a pesquisa cobriu apenas uma pequena parte, mas se baseou em conhecimentos sólidos sobre as melhores práticas. Por exemplo, sabe-se que o tempo de treino cresce de forma exponencial com a adição de novas camadas convolucionais e neurônios, mas isso não implica que os resultados irão progredir na mesma proporção. A pesquisa constatou exatamente isso, pois não houve uma melhora significativa entre uma rede simples e outra mais complexa, apenas o tempo teve um incremento significativo. O mesmo padrão foi observado nos demais algoritmos, cuja configuração impactou o tempo de treino e teve pouca influência nos resultados.

Isso não significa que o ajuste dos parâmetros não seja relevante para esse problema; pelo contrário, o ponto é que, antes de fazer esse ajuste fino, é necessário trabalhar com um conjunto mínimo de dados, mas que represente fielmente a dinâmica do modelo. Ou seja, o conjunto de informações deve primeiro ser tratado adequadamente e depois deve-se escolher um modelo que seja rápido o suficiente para permitir o teste de várias possibilidades. Sobre as redes testadas pode-se inferir o seguinte:

- **MLP:** os treinos ficaram na faixa dos minutos e pouca melhoria foi observada de uma rede para outra. Com esse embasamento é possível afirmar que em um trabalho futuro, antes mesmo de fazer o planejamento fatorial é necessário criar uma abstração mais relevante e objetiva dos dados;
- **CNN:** entre os três algoritmos esse método apresentou os piores resultados, o tempo de treino foi demasiado e a qualidade obtida foi a mais baixa. Essa rede é notável por ser usada em problemas de classificação e recentemente tem sido aplicada em sistemas de regressão [Lathuili et al. 2019], por isso não é possível dizer até onde os resultados foram condizentes com sua capacidade de operação. Além disso, a rede foi concebida para trabalhar com imagens, o que sugere que seria necessário desenvolver uma abordagem alternativa para este algoritmo. Ou seja, é um campo fértil para ser explorado, mas necessita ser estudado e modelado de acordo com os critérios práticos da arquitetura;
- **XBoost Regressor:** este algoritmo apresentou os melhores resultados entre as redes comparativas. A acurácia é visivelmente melhor e o tempo de treino se limitou apenas a alguns segundos, porém esses dados devem ser analisados com cautela, pois uma das

tendências negativas desse método é que o mesmo pode incorrer facilmente em sobre-treinamento. Uma verificação mais rigorosa do método incluiria testá-lo com mais dados e sobre diferentes configurações.

### 5.3 Trabalhos Futuros

Como descrito anteriormente pretende-se avaliar a rede com relação a quantidade mínima de informação que deve ser usada para alcançar um balanço entre tempo e acurácia. A princípio, essa etapa deve ser feita de modo incremental, por exemplo, vamos considerar o XBoost Regressor, pois o tempo de treino foi mais reduzido. Escolhida a rede pode-se primeiro tentar melhorar os dados. Com o estudo de técnicas que versam sobre os *Features* é possível formatar a entrada de forma que a mesma apresente uma variabilidade e um ruído menor. Com essa entrada homogênea e representativa o treino será realizado novamente e os resultados serão reavaliados. A partir do momento que não for mais possível obter melhoria, os dados apresentados à rede serão incrementados, ou seja, ao invés de 4 mil amostras podemos utilizar 8 mil e assim refazer o processo sucessivamente.

Isso pode ser feito de modo paralelo com as outras redes, pois uma vez que os dados foram tratados e as redes estão configuradas, basta replicar o experimento e fazer a devida comparação. Outras possibilidades que podem ser testadas são relativas a frequência de amostragem da IMU, sua reinicialização em um intervalo de tempo ou até mesmo sua substituição por um modelo que apresente um ruído menor, mas possibilite a fusão de seus sensores.

Sabendo-se que a previsão da rede é confiável é possível hipotetizar que esses dados podem ser utilizados para realimentar o software controlador e atualizar o seu modelo funcional de modo incremental e eficaz através de uma matriz de compensação, sempre comparando as medidas obtidas pela junção entre a IMU e a RNA com os dados nominais que são esperados, e havendo uma divergência entre as partes medidas é possível adicionar um valor ao software que compense esse desvio, aumentando a vida útil do robô e seu tempo de trabalho, sem a necessidade uma intervenção mecânica.

Outra etapa da pesquisa é desenvolver um método que possibilite a calibração *online* do robô em termos práticos, ou seja, visa-se desenvolver uma matriz de compensação que atenuie os desvios quando os mesmos forem detectados. Por exemplo, considerando que as medidas obtidas pelo ANN-RB comecem a desviar dos valores nominais, assim, é necessário realimentar o software do robô com essa diferença. Com esses dados o software poderia fazer os ajustes de maneira recursiva. Vamos supor que o erro no eixo  $x$  fosse de 1 milímetro para a direita e que o modelo cinemático dessa coordenada é denotado por  $X$ , o que pode ser feito é adicionar incrementos ao modelo e comparar com a nova medida, assim teríamos:

$$X_{ajustado} = X + 0.1n$$

onde  $n$  seria o número da iteração e esta iria ocorrer quantas vezes fosse necessário, até que o trajeto medido pela Rede Neural Artificial seja igual ao modelo nominal  $X$  ou apresente um erro  $e$  mínimo, definido explicitamente. Assim, o sistema implementado no software do robô seria o ajustado e o modelo nominal só iria existir como referência para comparar aquilo que é medido pela RNA com o trajeto esperado. A técnica seria então aplicada aos demais eixos, considerando-se sempre duas possibilidades de alinhamento em cada coordenada, ou seja, alinhamento positivo ou negativo.

Obviamente, esse exemplo é trivial e incipiente, e devemos considerar métodos numéricos mais sofisticados para ajustar a curva da função do erro de acordo com os pontos programados no controlador. Ou seja, o que se deseja é criar uma espécie de “equação de ajuste” ou correlação entre os pontos programados no controlador do robô e os pontos efetivamente atingidos por seu elemento terminal.

Com o sistema ajustado, possuindo tempo de treino aceitável e acurácia satisfatória, pode-se pensar em um modelo mais ambicioso. Ao invés de prever uma posição dentro do espaço de trabalho, seria desejável prever uma posição qualquer, independente do caminho ser conhecido ou não. Isto é, o sistema seria treinado dentro de uma malha no plano  $x, y, z$  conhecida e depois seria testada em trajetórias completamente diferentes e aleatórias, incluindo paradas súbitas, trajeto reverso e outras adversidades que um sistema prático de navegação está sujeito. Isso permitiria ao modelo a capacidade de extrapolação, sem a limitação de uma faixa pré definida.



## Capítulo 6

# Conclusão

Eu não chuto. Como cientista eu chego a conclusões baseadas em observação e experimentação.

---

Sheldon Cooper

A primeira conclusão que se obtém desta pesquisa versa diretamente sobre a hipótese inicial e confirma que a abordagem pode ser implementada na indústria, já que o experimento mostrou-se viável e reproduzível, o que permitiu realizar a navegação guiada utilizando uma IMU e Redes Neurais Artificiais. Os pontos previstos pela rede são obtidos através de uma solução de software barata e dispensa o uso de equipamentos de medição sofisticados. Ou seja, conclui-se que a técnica introduzida pode ser empregada para calibração da posição de uma IMU colocada na flange do efetuador de um robô. As RNAs foram hábeis em realizar a correlação não linear entre as entradas e saídas em um sistema caixa preta, onde as relações entre as variáveis não são conhecidas, e isso permitiu descrever a trajetória do IRB-360 em seu espaço de trabalho.

O modelo foi capaz de mitigar os erros oriundos da IMU, tornando o sistema de predição altamente preciso e com erros sub milimétricos. A posição do efetuador do robô em movimento guiado foi comparada com os valores reais do software e comprovou-se que a diferença máxima não passou de 0.9 milímetros. Outra derivação dos resultados é a possibilidade de otimizar e baratear um sistema de calibração e, como a IMU é um elemento simples que pode ser facilmente acoplado ao robô sem interferir no seu funcionamento e a um custo baixo, é plausível substituir os métodos atuais de calibração que necessitam de equipamentos sofisticados, são caros e acabam parando toda linha de produção, o que inevitavelmente incorre em perdas.

Verificou-se com este trabalho que os sinais gerados pela IMU de fato eram ruidosos e não representam a dinâmica real do efetuador no seu curso de operação, porém escondem infor-

mações importantes, como uma rampa, fim de curso ou mudança de direção. Foi exatamente essa relação subjetiva e não óbvia que a rede proposta conseguiu descrever com precisão.

Uma conclusão que pode ser vista como negativa é com relação ao tempo de treino, porém é preciso cuidado antes de afirmar que isso seja uma desvantagem do sistema. Primeiro, que uma vez que o sistema foi treinado ele pode ser replicado e distribuído de modo trivial. Segundo, é preciso analisar a metodologia e perceber que a mesma foi feita de maneira extremamente ampla, seja pelo critério de utilização do planejamento fatorial ou pelo volume de dados que foram inseridos no sistema sem um tratamento estatístico *a priori*, ou seja, o autor preferiu atacar o problema de modo amplo e obter uma resolução ótima, para futuramente pensar em uma maneira de otimização. Assim, pelos dois motivos citados, o tempo de treino não deve ser considerado um fator limitante, muito menos uma impossibilidade. Nesse caso, o tempo deve ser tratado como um elemento motivador que direciona a pesquisa para outros campos, como por exemplo, a otimização do método e o tratamento e extração da informação.

Como o autor desenvolveu sua própria rede MLP (ANN-RB) com retro-propagação, fez-se necessário comparar a abordagem com outras já estabelecidas na literatura e amplamente divulgadas no meio comercial e acadêmico. Nessa etapa utilizou-se um conjunto de dados com apenas 0.075% das amostras totais, isso foi feito por dois motivos: acomodar o tempo de treino de diversas configurações e verificar se o modelo era mais sensível à informação ou a arquitetura da rede. Nesse ponto, pode-se concluir que a rede é mais dependente dos dados, pois os classificadores não conseguiram uma representação fidedigna do trajeto (embora os resultados tenham sido satisfatórios), quando testados com diferentes arranjos e variando-se seus respectivos hiper-parâmetros. Isso não quer dizer que o ajuste seja dispensável e sim que ele deve ser feito sobre um conjunto mínimo de dados, capazes de maximizar o resultado e em um tempo aceitável. O segundo motivo dessa escolha diz respeito a viabilidade de realizar tantos testes em um conjunto de amostras elevado, como o tempo de treino possui uma relação direta com a quantidade de dados, é inviável realizar centenas de operações onde cada uma delas poderia demandar dias. Ao invés disso, a pesquisa vai além do proposto e sugere uma metodologia para um trabalho futuro que possa ser feita de maneira incremental, utilizando um conjunto de dados que seja representativo, objetivo e capaz de descrever melhor o modelo cinemático do efetor, e assim atender os requisitos de acurácia e tempo de treino de modo conjunto. A cerca disso, pode-se concluir que a pesquisa ainda possui desdobramentos futuros.

A rede CNN avaliada obteve os resultados mais pobres do experimento, porém não é possível afirmar que sua utilização é inviável, tal algoritmo é dotado de grandes qualidades, mas é sensível ao conjunto de entrada, assim sendo, é necessário avaliar tal método em configurações que contem com mais informações ou utilizem um esquema de pré processamento para a entrada. O algoritmo XBoost Regressor apresentou resultados promissores em todos os aspectos avaliados, uma comparação com a rede CNN evidencia esse fato. Enquanto o

XBoost Regressor levou aproximadamente apenas alguns segundos para realizar uma etapa de treino, a rede CNN não conseguiu obter os mesmos resultados de acurácia, mesmo com um processo de treino na casa dos minutos. Mas isso também não é suficiente para cravar a afirmação que o XBoost Regressor é o melhor método, pois um de seus problemas é justamente o treino excessivo, assim seria fundamental realizar outras rotinas avaliativas para esclarecer qual abordagem é capaz de endereçar de maneira efetiva essa lacuna. Conclui-se assim, que o conceito pode ser estendido e averiguado além dos pontos expostos nesse documento.

No decorrer da pesquisa, foi possível vislumbrar outros campos que podem se beneficiar dessa tecnologia. Com o avanço dos dispositivos microeletrônicos e sua consequente redução de preços, é razoável concluir que outros setores irão se beneficiar dos conceitos apresentados nesta tese. Qualquer mecanismo que necessite de um modo de localização preciso e imediato poderá embarcar na solução introduzida nessa pesquisa. Um *drone* que realize entregas e um sistema de navegação são duas possibilidades preliminares que instigam a pesquisa futura. Portanto, conclui-se que o artefato gerado é passível de beneficiar variados setores e domínios do conhecimento.

Finalmente, sobre todos os aspectos avaliados, pode-se concluir que a pesquisa é significativa, pois introduziu um método autônomo capaz de realizar a predição exata do efator do robô, permitindo assim, a sua futura calibração *online* (utilizando elementos relativamente baratos e ferramentas que estão ao alcance de qualquer empresa). Além disso, os resultados não colocam um ponto final no tema, pelo contrário, o que se observa é a abertura de outros campos férteis e a possibilidade da aplicação da pesquisa em diferentes domínios, seja pelo aperfeiçoamento do método ou através de outras propostas que visem resolver problemas semelhantes. Assim sendo, conclui-se que a Tese endereçou satisfatoriamente o problema o qual se propôs, introduzindo assim uma nova metodologia de navegação guiada, capaz de fornecer os subsídios necessários à calibração cinemática *online* do TCP de um robô.

# REFERÊNCIAS BIBLIOGRÁFICAS

Abiodun et al. 2018 ABIODUN, O. I. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, v. 4, n. 11, p. e00938, nov. 2018. ISSN 24058440. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S2405844018332067>>.

Ali et al. 2016 ALI, Y. H. et al. Acoustic emission and artificial intelligent methods in condition monitoring of rotating machine – a review. *The National Conference for Postgraduate Research 2016, Universiti Malaysia Pahang*, 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7847379/>>.

Allamy 2014 ALLAMY, H. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). 12 2014.

BACURAU 2017 BACURAU, R. M. *Projeto e implementação de um sistema eletrônico para giroscópios interferométricos a fibra óptica de malha fechada*. 2017. 179 p. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/332999>>.

Bahrin et al. 2016 BAHRIN, M. A. K. et al. Industry 4. 0: a review on industrial automation 422 and robotic. *Jurnal Teknologi*, v. 78, n. 6-13, jun. 2016. ISSN 2180-3722, 0127-9696. Disponível em: <<https://jurnalteknologi.utm.my/index.php/jurnalteknologi/article/view/9285>>.

Bai et al. 2019 BAI, J. et al. The parameter identification model considering both geometric parameters and joint stiffness. *Industrial Robot: the international journal of robotics research and application*, v. 47, n. 1, p. 76–81, set. 2019. ISSN 0143-991X, 0143-991X. Disponível em: <<https://www.emerald.com/insight/content/doi/10.1108/IR-11-2018-0223/full/html>>.

Balestrassi et al. 2009 BALESTRASSI, P. et al. Design of experiments on neural network’s training for nonlinear time series forecasting. *Neurocomputing*, v. 72, n. 4-6, p. 1160–1178, jan. 2009. ISSN 09252312. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0925231208001513>>.

Baranov et al. 2019 BARANOV, P. et al. A novel multiple-axis MEMS gyroscope-accelerometer with decoupling frames. *Sensor Review*, v. 39, n. 5, p. 670–681, set. 2019. ISSN 0260-2288, 0260-2288. Disponível em: <<https://www.emerald.com/insight/content/doi/10.1108/SR-05-2018-0133/full/html>>.

Bennett e Hollerbach 1991 BENNETT, D.; HOLLERBACH, J. Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint

- constraints. *IEEE Transactions on Robotics and Automation*, v. 7, n. 5, p. 597–606, out. 1991. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/97871/>>.
- BOSH 2021 BOSH. *BNO055 Intelligent 9-axis absolute orientation sensor*. 2021. 118 p. Disponível em: <<https://datasheetspdf.com/datasheet/BNO055.html>>.
- Brotzer et al. 2021 BROTZER, A. et al. Automated Quality Assessment of Interferometric Ring Laser Data. *Sensors*, v. 21, n. 10, p. 3425, maio 2021. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/21/10/3425>>.
- CAMPOS 2006 CAMPOS, B. A. N. *Simulador para Modelagem e Calibração de Robôs Industriais*. 2006. 153 p. Disponível em: <<https://repositorio.unb.br/handle/10482/2448>>.
- Campos e Motta 2021 Campos, B. A. N.; Motta, J. M. S. T. Online measuring of robot positions using inertial measurement units, sensor fusion and artificial intelligence. *IEEE Access*, v. 9, p. 5678–5689, 2021. Disponível em: <<https://ieeexplore.ieee.org/document/9312193>>.
- Chen-Gang et al. 2014 Chen-Gang et al. Review on kinematics calibration technology of serial robots. *International Journal of Precision Engineering and Manufacturing*, v. 15, n. 8, p. 1759–1774, ago. 2014. ISSN 2234-7593, 2005-4602. Disponível em: <<http://link.springer.com/10.1007/s12541-014-0528-1>>.
- Chen e Chao 1986 CHEN, J.; CHAO, L. Positioning error analysis for robot manipulators with all rotary joints. In: *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. San Francisco, CA, USA: Institute of Electrical and Electronics Engineers, 1986. v. 3, p. 1011–1016. Disponível em: <<http://ieeexplore.ieee.org/document/1087544/>>.
- Chen 2017 CHEN, K. *Gradient Descent and Backpropagation*. 2017. Disponível em: <<https://www.linkedin.com/pulse/gradient-descent-backpropagation-ken-chen/>>.
- Cicco et al. 2015 CICCIO, M. D. et al. Non-parametric calibration for depth sensors. *Robotics and Autonomous Systems*, v. 74, p. 309–317, dez. 2015. ISSN 09218890. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0921889015001724>>.
- Consultants 2014 CONSULTANTS, R. B. S. *The New Industrial Revolution: How Europe Will Succeed*. 2014. Disponível em: <[http://www.iberglobal.com/files/Roland\\_Berger\\_Industry.pdf](http://www.iberglobal.com/files/Roland_Berger_Industry.pdf)>.
- Djuric, Saidi e ElMaraghy 2012 DJURIC, A.; SAIDI, R. A.; ELMARAGHY, W. Dynamics solution of n-DOF global machinery model. *Robotics and Computer-Integrated Manufacturing*, v. 28, n. 5, p. 621–630, out. 2012. ISSN 07365845. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0736584512000233>>.
- Dominik et al. 2010 DOMINIK, S. et al. Evaluation of pooling operations in convolutional architectures for object recognition. In: *Artificial Neural Networks – ICANN 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. v. 6354, p. 92–101. ISBN 9783642158247 9783642158254. Disponível em: <[http://link.springer.com/10.1007/978-3-642-15825-4\\_10](http://link.springer.com/10.1007/978-3-642-15825-4_10)>.
- Dong et al. 2020 DONG, M. et al. Calibration of Low Cost IMU's Inertial Sensors for Improved Attitude Estimation. *Journal of Intelligent & Robotic Systems*, v. 100,

- n. 3-4, p. 1015–1029, dez. 2020. ISSN 0921-0296, 1573-0409. Disponível em: <<http://link.springer.com/10.1007/s10846-020-01259-0>>.
- Driels e Pathre 1991 DRIELS, M.; PATHRE, U. Vision-based automatic theodolite for robot calibration. *IEEE Transactions on Robotics and Automation*, v. 7, n. 3, p. 351–360, jun. 1991. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/88144/>>.
- Du et al. 2020 DU, G. et al. Online robot kinematic calibration using hybrid filter with multiple sensors. *IEEE Transactions on Instrumentation and Measurement*, v. 69, n. 9, p. 7092–7107, 2020.
- Du e Zhang 2013 DU, G.; ZHANG, P. Imu-based online kinematic calibration of robot manipulator. *The Scientific World Journal*, v. 2013, p. 1–10, 2013. ISSN 1537-744X. Disponível em: <<http://www.hindawi.com/journals/tswj/2013/139738/>>.
- Du et al. 2016 DU, S. et al. Mems imu error mitigation using rotation modulation technique. *Sensors*, v. 16, n. 12, p. 2017, nov. 2016. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/16/12/2017>>.
- Duelen e Schröer 1991 DUELEN, G.; SCHRÖER, K. Robot calibration—Method and results. *Robotics and Computer-Integrated Manufacturing*, v. 8, n. 4, p. 223–231, 1991. ISSN 07365845. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/073658459190034P>>.
- Elatta et al. 2004 ELATTA, A. et al. An overview of robot calibration. *Information Technology Journal*, v. 3, n. 1, p. 74–78, jan. 2004. ISSN 18125638, 18125646. Disponível em: <<http://www.scialert.net/abstract/?doi=itj.2004.74.78>>.
- Emmert-Streib et al. 2020 EMMERT-STREIB, F. et al. An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, v. 3, p. 4, fev. 2020. ISSN 2624-8212. Disponível em: <<https://www.frontiersin.org/article/10.3389/frai.2020.00004/full>>.
- Everett e Hsu 1988 EVERETT, L. J.; HSU, T.-W. The theory of kinematic parameter identification for industrial robots. *Journal of Dynamic Systems, Measurement, and Control*, v. 110, n. 1, p. 96–100, mar. 1988. ISSN 0022-0434, 1528-9028. Disponível em: <<https://asmedigitalcollection.asme.org/dynamicsystems/article/110/1/96/400218/The-Theory-of-Kinematic-Parameter-Identification>>.
- Fraden 2004 FRADEN, J. *Handbook of modern sensors: physics, designs, and applications*. 3rd ed. ed. New York: Springer, 2004. ISBN 9780387007502.
- Gan e Dai 2011 GAN, Y.; DAI, X. Base frame calibration for coordinated industrial robots. *Robotics and Autonomous Systems*, v. 59, n. 7-8, p. 563–570, jul. 2011. ISSN 09218890. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0921889011000728>>.
- Gaudreault et al. 2018 GAUDREAUULT, M. et al. Self-calibration of an industrial robot using a novel affordable 3d measuring device. *Sensors*, v. 18, n. 10, p. 3380, out. 2018. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/18/10/3380>>.
- Ginani e Motta 2011 GINANI, L. S.; MOTTA, J. M. S. T. Theoretical and practical aspects of robot calibration with experimental verification. *Journal of the Brazilian*

- Society of Mechanical Sciences and Engineering*, v. 33, n. 1, p. 15–21, mar. 2011. ISSN 1678-5878. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_abstract&pid=S1678-58782011000100003&lng=en&nrm=iso&tlng=en](http://www.scielo.br/scielo.php?script=sci_abstract&pid=S1678-58782011000100003&lng=en&nrm=iso&tlng=en)>.
- Godha e Cannon 2007 GODHA, S.; CANNON, M. E. GPS/MEMS INS integrated system for navigation in urban areas. *GPS Solutions*, v. 11, n. 3, p. 193–203, maio 2007. ISSN 1080-5370, 1521-1886. Disponível em: <<http://link.springer.com/10.1007/s10291-006-0050-8>>.
- Gonseth et al. 2015 GONSETH, S. et al. Miniaturized high-performance MEMS accelerometer detector. *CEAS Space Journal*, v. 7, n. 2, p. 263–270, jun. 2015. ISSN 1868-2502, 1868-2510. Disponível em: <<http://link.springer.com/10.1007/s12567-015-0093-1>>.
- Goswami et al. 1993 GOSWAMI, A. et al. Complete parameter identification of a robot from partial pose information. In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*. Atlanta, GA, USA: IEEE Comput. Soc. Press, 1993. p. 168–173. ISBN 9780818634505. Disponível em: <<http://ieeexplore.ieee.org/document/291978/>>.
- Ha 2008 HA, I.-C. Kinematic parameter calibration method for industrial robot manipulator using the relative position. *Journal of Mechanical Science and Technology*, v. 22, n. 6, p. 1084–1090, jun. 2008. ISSN 1738-494X, 1976-3824. Disponível em: <<http://link.springer.com/10.1007/s12206-008-0305-0>>.
- Hassabis et al. 2017 HASSABIS, D. et al. Neuroscience-inspired artificial intelligence. *Neuron*, v. 95, n. 2, p. 245–258, jul. 2017. ISSN 08966273. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0896627317305093>>.
- Hayati e Mirmirani 1985 HAYATI, S.; MIRMIRANI, M. Improving the absolute positioning accuracy of robot manipulators. *Journal of Robotic Systems*, v. 2, n. 4, p. 397–413, 1985. ISSN 07412223, 10974563. Disponível em: <<http://doi.wiley.com/10.1002/rob.4620020406>>.
- Haykin 1999 HAYKIN, S. S. *Neural networks: a comprehensive foundation*. 2nd ed. ed. Upper Saddle River, N.J: Prentice Hall, 1999. ISBN 9780132733502.
- Haykin 2009 HAYKIN, S. S. *Neural networks and learning machines*. 3rd ed. ed. New York: Prentice Hall, 2009. OCLC: ocn237325326. ISBN 9780131471399.
- Hiller et al. 2019 HILLER, T. et al. Origins and mechanisms of bias instability noise in a three-axis mode-matched mems gyroscope. *Journal of Microelectromechanical Systems*, v. 28, n. 4, p. 586–596, 2019.
- Ho 2014 HO, J. *An Introduction to Semiconductor Physics, Technology, and Industry*. 2014. Disponível em: <<https://www.anandtech.com/show/8223/an-introduction-to-semiconductor-physics-technology-and-industry/3>>.
- Hofmann-Wellenhof et al. 2003 HOFMANN-WELLENHOF, B. et al. *Navigation: principles of positioning and guidance*. Wien ; New York: Springer, 2003. OCLC: ocm52058434. ISBN 9783211008287.
- Hollerbach e Lokhorst 1995 HOLLERBACH, J.; LOKHORST, D. Closed-loop kinematic calibration of the RSI 6-DOF hand controller. *IEEE Transactions on Robotics and Automation*, v. 11, n. 3, p. 352–359, jun. 1995. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/388777/>>.

Hosseinyalamdary 2018 HOSSEINYALAMDARY, S. Deep Kalman Filter: Simultaneous Multi-Sensor Integration and Modelling; A GNSS/IMU Case Study. *Sensors*, v. 18, n. 5, p. 1316, abr. 2018. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/18/5/1316>>.

INMETRO 2012 INMETRO. *Vocabulário Internacional de Metrologia*. Instituto Nacional de Metrologia Qualidade e Tecnologia, 2012. Disponível em: <[http://www.inmetro.gov.br/inovacao/publicacoes/vim\\_2012.pdf](http://www.inmetro.gov.br/inovacao/publicacoes/vim_2012.pdf)>.

Ioffe e Szegedy 2015 IOFFE, S.; SZEGEDY, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. Disponível em: <<https://arxiv.org/abs/1502.03167>>.

Jeelani et al. 2015 JEELANI, S. et al. Robotics and medicine: A scientific rainbow in hospital. *Journal of Pharmacy and Bioallied Sciences*, v. 7, n. 6, p. 381, 2015. ISSN 0975-7406. Disponível em: <<http://www.jpbonline.org/text.asp?2015/7/6/381/163460>>.

Jessie e Albert 2018 JESSIE, R. R. H.; ALBERT, V. T. A. Training radial basis function networks for wind speed prediction using PSO enhanced differential search optimizer. *PLOS ONE*, v. 13, n. 5, p. e0196871, maio 2018. ISSN 1932-6203. Disponível em: <<https://dx.plos.org/10.1371/journal.pone.0196871>>.

Jia et al. 2019 JIA, W. et al. Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 2019.

Judd e Knasinski 1990 JUDD, R.; KNASINSKI, A. A technique to calibrate industrial robots with experimental verification. *IEEE Transactions on Robotics and Automation*, v. 6, n. 1, p. 20–30, fev. 1990. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/88114/>>.

Judd e Knasinski 1990 JUDD, R.; KNASINSKI, A. A technique to calibrate industrial robots with experimental verification. *IEEE Transactions on Robotics and Automation*, v. 6, n. 1, p. 20–30, fev. 1990. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/88114/>>.

Judy 2001 JUDY, J. W. Microelectromechanical systems (Mems): fabrication, design and applications. *Smart Materials and Structures*, v. 10, n. 6, p. 1115–1134, dez. 2001. ISSN 0964-1726, 1361-665X. Disponível em: <<https://iopscience.iop.org/article/10.1088/0964-1726/10/6/301>>.

Jung et al. 2018 JUNG, S. et al. A direct visual servoing-based framework for the 2016 IROS Autonomous Drone Racing Challenge. *Journal of Field Robotics*, v. 35, n. 1, p. 146–166, jan. 2018. ISSN 15564959. Disponível em: <<http://doi.wiley.com/10.1002/rob.21743>>.

Kirkko-Jaakkola, Collin e Takala 2012 KIRKKO-JAANKOLA, M.; COLLIN, J.; TAKALA, J. Bias prediction for mems gyroscopes. *Sensors Journal, IEEE*, v. 12, p. 2157–2163, 06 2012.

Kolyubin et al. 2015 KOLYUBIN, S. et al. Robot kinematics identification: kuka lwr4+ redundant manipulator example. *Journal of Physics: Conference Series*, v. 659, p. 012011, nov. 2015. ISSN 1742-6588, 1742-6596. Disponível em: <<https://iopscience.iop.org/article/10.1088/1742-6596/659/1/012011>>.



- Kurbatov, Kurbatov e Goryachkin 2019 KURBATOV, A. M.; KURBATOV, R. A.; GORYACHKIN, A. M. Fiber-optic gyroscope accuracy improvement by suppressing the parasitic effects in integrated optic phase modulators. *Gyroscopy and Navigation*, v. 10, n. 4, p. 256–267, out. 2019. ISSN 2075-1087, 2075-1109. Disponível em: <<http://link.springer.com/10.1134/S2075108719040114>>.
- KVH 2016 KVH. *Guide to Comparing Gyro and IMU Technologies – Micro-Electro-Mechanical Systems and Fiber Optic Gyros*. 2016. Disponível em: <<https://caclase.co.uk/wp-content/uploads/2016/11/Guide-to-Comparing-Gyros-0914.pdf>>.
- Lam et al. 2003 LAM, Q. et al. Gyro Modeling and Estimation of Its Random Noise Sources. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Austin, Texas: American Institute of Aeronautics and Astronautics, 2003. ISBN 9781624100901. Disponível em: <<https://arc.aiaa.org/doi/10.2514/6.2003-5562>>.
- Lathuili et al. 2019 LATHUILI, S. et al. *A Comprehensive Analysis of Deep Regression*. 2019. Disponível em: <<https://arxiv.org/pdf/1803.08450.pdf>>.
- Lavrik e Datskos 2019 LAVRIK, N. V.; DATSKOS, P. G. Optically read Coriolis vibratory gyroscope based on a silicon tuning fork. *Microsystems & Nanoengineering*, v. 5, n. 1, p. 47, dez. 2019. ISSN 2055-7434. Disponível em: <<http://www.nature.com/articles/s41378-019-0087-9>>.
- Le e Kang 2020 LE, P.-N.; KANG, H.-J. A Robotic Calibration Method Using a Model-Based Identification Technique and an Invasive Weed Optimization Neural Network Compensator. *Applied Sciences*, v. 10, n. 20, p. 7320, out. 2020. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/10/20/7320>>.
- Lee et al. 2016 LEE, C.-G. et al. Gyro Drift Correction for An Indirect Kalman Filter Based Sensor Fusion Driver. *Sensors*, v. 16, n. 6, p. 864, jun. 2016. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/16/6/864>>.
- Li et al. 2018 LI, H. et al. Fully convolutional network ensembles for white matter hyperintensities segmentation in MR images. *NeuroImage*, v. 183, p. 650–665, dez. 2018. ISSN 10538119. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1053811918305974>>.
- Li et al. 2019 LI, S. et al. Accelerometer-Based Gyroscope Drift Compensation Approach in a Dual-Axial Stabilization Platform. *Electronics*, v. 8, n. 5, p. 594, maio 2019. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/8/5/594>>.
- Marie et al. 2013 MARIE, S. et al. Elasto-geometrical modeling and calibration of robot manipulators: Application to machining and forming applications. *Mechanism and Machine Theory*, v. 69, p. 13–43, nov. 2013. ISSN 0094114X. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0094114X13000967>>.
- Mavroidis et al. 1997 MAVROIDIS, C. et al. A systematic error analysis of robotic manipulators: application to a high performance medical robot. In: *Proceedings of International Conference on Robotics and Automation*. Albuquerque, NM, USA: IEEE, 1997. v. 2, p. 980–985. ISBN 9780780336124. Disponível em: <<http://ieeexplore.ieee.org/document/614262/>>.

- Maye et al. 2016 MAYE, J. et al. Online self-calibration for robotic systems. *The International Journal of Robotics Research*, v. 35, n. 4, p. 357–380, abr. 2016. ISSN 0278-3649. Disponível em: <<https://doi.org/10.1177/0278364915596232>>.
- Meggiolaro 2000 MEGGIOLARO, M. *Achieving fine absolute positioning accuracy in large powerful manipulators*. Tese (2000) — Massachusetts Institute of Technology, Massachusetts, USA, 2000. Disponível em: <[http://meggi.usuarios.rdc.puc-rio.br/docs/Thesis\\_PhD.pdf](http://meggi.usuarios.rdc.puc-rio.br/docs/Thesis_PhD.pdf)>.
- Meggiolaro et al. 2004 MEGGIOLARO, M. et al. Error identification and compensation in large manipulators with application in cancer proton therapy. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, v. 15, n. 1, p. 71–77, mar. 2004. ISSN 0103-1759. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-17592004000100009&lng=en&tlng=en](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592004000100009&lng=en&tlng=en)>.
- Motta 2006 MOTTA, J. M. S. T. Robot calibration: modeling measurement and applications. In: KIN, L. (Ed.). *Industrial Robotics: Programming, Simulation and Applications*. Pro Literatur Verlag, Germany / ARS, Austria, 2006. ISBN 9783866112865. Disponível em: <[http://www.intechopen.com/books/industrial\\_robotics\\_programming\\_simulation\\_and\\_applications/robot\\_calibration\\_\\_modeling\\_measurement\\_and\\_applications](http://www.intechopen.com/books/industrial_robotics_programming_simulation_and_applications/robot_calibration__modeling_measurement_and_applications)>.
- Motta e McMaster 1999 MOTTA, J. M. S. T.; MCMMASTER, R. S. Modeling, optimizing and simulating robot calibration with accuracy improvement. *Journal of the Brazilian Society of Mechanical Sciences*, v. 21, n. 3, p. 384–401, set. 1999. ISSN 0100-7386. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-73861999000300002&lng=en&tlng=en](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-73861999000300002&lng=en&tlng=en)>.
- Nasteski 2017 NASTESKI, V. An overview of the supervised machine learning methods. *HORIZONS.B*, v. 4, p. 51–62, dez. 2017. ISSN 18578578, 18579892. Disponível em: <<http://uklo.edu.mk/filemanager/HORIZONTALI%202017/Serija%20B%20br.%204/6.An%20overview%20of%20the%20supervised.pdf>>.
- Nguyen, Le e Kang 2019 NGUYEN, H.-N.; LE, P.-N.; KANG, H.-J. A new calibration method for enhancing robot position accuracy by combining a robot model-based identification approach and an artificial neural network-based error compensation technique. *Advances in Mechanical Engineering*, v. 11, n. 1, p. 168781401882293, jan. 2019. ISSN 1687-8140, 1687-8140. Disponível em: <<http://journals.sagepub.com/doi/10.1177/1687814018822935>>.
- Nimbus 2017 NIMBUS, V. M. *PERCEPTRON – REDES NEURAIS*. 2017. Disponível em: <<https://www.monolitonimbus.com.br/perceptron-redes-neurais/>>.
- North 2015 NORTH, G. R. *Encyclopedia of atmospheric sciences*. 2. ed. ed. Amsterdam: Elsevier, Acad. Press, 2015. ISBN 9780123822253.
- Oliveira 2020 OLIVEIRA, M. *Neurônios*. 2020. Disponível em: <<https://www.infoescola.com/sistema-nervoso/neuronios>>.
- Pagliarini e Lund 2017 PAGLIARINI, L.; LUND, H. H. The future of robotics technology. *Journal of Robotics, Networking and Artificial Life*, v. 3, n. 4, p. 270, 2017. ISSN 2352-6386. Disponível em: <<http://www.atlantis-press.com/php/paper-details.php?id=25872654>>.

- Passaro et al. 2017 PASSARO, V. M. N. et al. Gyroscope technology and applications: a review in the industrial perspective. *Sensors*, v. 17, n. 10, p. 2284, out. 2017. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/17/10/2284>>.
- Patel et al. 2009 PATEL, C. et al. Temperature Effects on the Performance and Reliability of MEMS Gyroscope Sensors. In: *ASME 2009 InterPACK Conference, Volume 1*. San Francisco, California, USA: ASMEDC, 2009. p. 507–512. ISBN 9780791843598 9780791838518. Disponível em: <<https://asmedigitalcollection.asme.org/InterPACK/proceedings/InterPACK2009/43598/507/341291>>.
- Patonis et al. 2018 PATONIS, P. et al. A Fusion Method for Combining Low-Cost IMU/Magnetometer Outputs for Use in Applications on Mobile Devices. *Sensors*, v. 18, n. 8, p. 2616, ago. 2018. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/18/8/2616>>.
- POWERMIG 2020 POWERMIG. *ROBÔ DE MANIPULAÇÃO IRB 360*. 2020. Disponível em: <<https://www.powermig.com.br/pt-br/tecnologia/robo-de-manipulacao-irb-360-191>>.
- Quadri e Sidek 2014 QUADRI, S. A.; SIDEK, O. Error and Noise Analysis in an IMU using Kalman Filter. *International Journal of Hybrid Information Technology*, v. 7, n. 3, p. 39–48, maio 2014. ISSN 17389968. Disponível em: <[http://gvpress.com/journals/IJHIT/vol7\\_no3/6.pdf](http://gvpress.com/journals/IJHIT/vol7_no3/6.pdf)>.
- Ribeiro e Santos 2017 RIBEIRO, N. F.; SANTOS, C. P. Inertial measurement units: A brief state of the art on gait analysis. In: *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*. Coimbra: IEEE, 2017. p. 1–4. ISBN 9781509048014. Disponível em: <<https://ieeexplore.ieee.org/document/7889458/>>.
- Ruibo He et al. 2010 Ruibo He et al. Kinematic-parameter identification for serial-robot calibration based on poe formula. *IEEE Transactions on Robotics*, v. 26, n. 3, p. 411–423, jun. 2010. ISSN 1552-3098, 1941-0468. Disponível em: <<http://ieeexplore.ieee.org/document/5457990/>>.
- Ryoichi et al. 2018 RYOICHI, I. et al. Offline and online calibration of mobile robot and slam device for navigation. *arXiv:1804.04817*, 2018. Disponível em: <[https://www.researchgate.net/publication/324536087\\_Offline\\_and\\_Online\\_calibration\\_of\\_Mobile\\_Robot\\_and\\_SLAM\\_Device\\_for\\_Navigation](https://www.researchgate.net/publication/324536087_Offline_and_Online_calibration_of_Mobile_Robot_and_SLAM_Device_for_Navigation)>.
- Sabsch et al. 2017 SABSCH, J. et al. Towards improving the absolute accuracy of lightweight robots by nonparametric calibration. In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Limassol: IEEE, 2017. p. 1–4. ISBN 9781509065059. Disponível em: <<http://ieeexplore.ieee.org/document/8247764/>>.
- Santos 2018 SANTOS, L. *Artificial Inteligence*. 2018. Disponível em: <[https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/feature\\_scaling.html](https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/feature_scaling.html)>.
- Schillreff et al. 2017 SCHILLREFF, N. et al. Towards high accuracy robot-assisted surgery \* \*this work was partly funded by the federal ministry of education and research in germany within the research campus stimulate under grant no. 13gw0095a. *IFAC-PapersOnLine*, v. 50, n. 1, p. 5666–5671, jul. 2017. ISSN 24058963. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S2405896317316063>>.

Schroer et al. 1997 SCHROER, K. et al. Modeling closed-loop mechanisms in robots for purposes of calibration. *IEEE Transactions on Robotics and Automation*, v. 13, n. 2, p. 218–229, abr. 1997. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/563644/>>.

Schröer et al. 1997 SCHRÖER, K. et al. Complete, minimal and model-continuous kinematic models for robot calibration. *Robotics and Computer-Integrated Manufacturing*, v. 13, n. 1, p. 73–85, mar. 1997. ISSN 07365845. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0736584596000257>>.

Scikit-Learn 2019 SCIKIT-LEARN. *Multi-layer Perceptron*. 2019. Disponível em: <[https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html#regression](https://scikit-learn.org/stable/modules/neural_networks_supervised.html#regression)>.

Shah et al. 2016 SHAH, M. A. et al. Modal analysis of a single-structure multi-axis MEMS gyroscope. *Journal of Sensors*, v. 2016, p. 1–8, 2016. ISSN 1687-725X, 1687-7268. Disponível em: <<https://www.hindawi.com/journals/js/2016/4615389/>>.

Shang et al. 2019 SHANG, D. et al. Research on the motion error analysis and compensation strategy of the delta robot. *Mathematics*, v. 7, n. 5, p. 411, maio 2019. ISSN 2227-7390. Disponível em: <<https://www.mdpi.com/2227-7390/7/5/411>>.

Shen et al. 2019 SHEN, Y.-F. et al. Convolutional neural network-based method for real-time orientation indexing of measured electron backscatter diffraction patterns. *Acta Materialia*, v. 170, p. 118–131, maio 2019. ISSN 13596454. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1359645419301697>>.

Si et al. 2013 SI, C. et al. Bandwidth Optimization Design of a Multi Degree of Freedom MEMS Gyroscope. *Sensors*, v. 13, n. 8, p. 10550–10560, ago. 2013. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/13/8/10550>>.

Silva e Schmidt 2016 SILVA, S. R. e.; SCHIMIDT, F. Reduction of input variables in artificial neural networks as from principal components analysis data in the modeling of dissolved oxygen. *Química Nova*, 2016. ISSN 0100-4042. Disponível em: <<http://www.gnresearch.org/doi/10.5935/0100-4042.20160024>>.

Sparkfun 2018 SPARKFUN. *Accelerometer, Gyro and IMU Buying Guide*. 2018. Disponível em: <[https://www.sparkfun.com/pages/accel\\_gyro\\_guide](https://www.sparkfun.com/pages/accel_gyro_guide)>.

Stier et al. 2018 STIER, J. et al. Analysing neural network topologies: a game theoretic approach. *Procedia Computer Science*, v. 126, p. 234–243, 2018. ISSN 18770509. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S187705091831233X>>.

Stone 1987 STONE, H. W. *Kinematic modeling, identification, and control of robotic manipulators*. Boston, MA: Springer US, 1987. v. 29. (The Kluwer International Series in Engineering and Computer Science, v. 29). ISBN 9781461291930 9781461319993. Disponível em: <<http://link.springer.com/10.1007/978-1-4613-1999-3>>.

Szegedy et al. 2015 SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 1–9. ISBN 9781467369640. Disponível em: <<http://ieeexplore.ieee.org/document/7298594/>>.

- Taheri et al. 2014 TAHERI, P. T. et al. Micro-scale diamond hemispherical resonator gyroscope. In: *2014 Solid-State, Actuators, and Microsystems Workshop Technical Digest*. Hilton Head, South Carolina, USA: Transducer Research Foundation, 2014. p. 289–292. ISBN 9781940470016. Disponível em: <[https://transducer-research-foundation.org/technical\\_digests/HiltonHead\\_2014/hh2014\\_0289.pdf](https://transducer-research-foundation.org/technical_digests/HiltonHead_2014/hh2014_0289.pdf)>.
- Talukdar et al. 2011 TALUKDAR, S. et al. Implementation of low cost inertial measurement unit (Imu) integrated with a global positioning system (Gps) receiver- a study. In: . [s.n.], 2011. p. 2011–01–1040. Disponível em: <<https://www.sae.org/content/2011-01-1040/>>.
- Tchon 1992 TCHON, K. Calibration of manipulator kinematics: a singularity theory approach. *IEEE Transactions on Robotics and Automation*, v. 8, n. 5, p. 671–678, out. 1992. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/163792/>>.
- Tobe 2017 TOBE, F. *Robotics industry growing faster than expected*. 2017. Disponível em: <<https://www.therobotreport.com/robotics-industry-growing-faster-than-expected/>>.
- Tong et al. 2003 TONG, K. et al. Parametric error modeling and software error compensation for rapid prototyping. *Rapid Prototyping Journal*, v. 9, n. 5, p. 301–313, dez. 2003. ISSN 1355-2546. Disponível em: <<https://www.emerald.com/insight/content/doi/10.1108/13552540310502202/full/html>>.
- Veitschegger e Wu 1988 VEITSCHEGGER, W.; WU, C.-H. Robot calibration and compensation. *IEEE Journal on Robotics and Automation*, v. 4, n. 6, p. 643–656, dez. 1988. ISSN 08824967. Disponível em: <<http://ieeexplore.ieee.org/document/9302/>>.
- Verplaetse 1996 VERPLAETSE, C. Inertial proprioceptive devices: Self-motion-sensing toys and tools. *IBM Systems Journal*, v. 35, n. 3.4, p. 639–650, 1996. ISSN 0018-8670. Disponível em: <<http://ieeexplore.ieee.org/document/5387210/>>.
- Vieler et al. 2016 VIELER, H. et al. Online calibration of industrial robots using inertial sensors. *Proceedings of ISR 2016: 47th International Symposium on Robotics*, 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7559120/authors>>.
- Vinuesa et al. 2020 VINUESA, R. et al. The role of artificial intelligence in achieving the Sustainable Development Goals. *Nature Communications*, v. 11, n. 1, p. 233, dez. 2020. ISSN 2041-1723. Disponível em: <<http://www.nature.com/articles/s41467-019-14108-y>>.
- Virgilio et al. 2019 VIRGILIO, A. D. V. D. et al. Analysis of ring laser gyroscopes including laser dynamics. *The European Physical Journal C*, v. 79, n. 7, p. 573, jul. 2019. ISSN 1434-6052. Disponível em: <<https://doi.org/10.1140/epjc/s10052-019-7089-5>>.
- Wang e Han 2020 WANG, S.; HAN, F. Analysis and Compensation of Bias Drift for a Micromachined Spinning-Rotor Gyroscope with Electrostatic Suspension. *Sensors*, v. 20, n. 6, p. 1799, mar. 2020. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/20/6/1799>>.
- Wang et al. 2019 WANG, Z. et al. A robot calibration method based on joint angle division and an artificial neural network. *Mathematical Problems in Engineering*, v. 2019, p. 1–12, mar. 2019. ISSN 1024-123X, 1563-5147. Disponível em: <<https://www.hindawi.com/journals/mpe/2019/9293484/>>.

- Wang et al. 2019 WANG, Z. et al. A Robot Calibration Method Based on Joint Angle Division and an Artificial Neural Network. *Mathematical Problems in Engineering*, v. 2019, p. 1–12, mar. 2019. ISSN 1024-123X, 1563-5147. Disponível em: <<https://www.hindawi.com/journals/mpe/2019/9293484/>>.
- Wang et al. 2018 WANG, Z. et al. Industrial robot trajectory accuracy evaluation maps for hybrid manufacturing process based on joint angle error analysis. *Advances in Robotics & Automation*, v. 07, n. 01, 2018. ISSN 21689695. Disponível em: <[shorturl.at/syVZ6](http://shorturl.at/syVZ6)>.
- Wu et al. 2015 WU, Y. et al. Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments. *Robotics and Computer-Integrated Manufacturing*, v. 35, p. 151–168, out. 2015. ISSN 07365845. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0736584515000411>>.
- Wu et al. 2012 WU, Y. et al. Using radial basis function networks for function approximation and classification. *ISRN Applied Mathematics*, v. 2012, p. 1–34, 2012. ISSN 2090-5572. Disponível em: <<https://www.hindawi.com/archive/2012/324194/>>.
- Xiao et al. 2019 XIAO, Y. et al. Online IMU Self-Calibration for Visual-Inertial Systems. *Sensors*, v. 19, n. 7, p. 1624, abr. 2019. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/19/7/1624>>.
- Xu et al. 2016 XU, Y. et al. Research on the calibration method of micro inertial measurement unit for engineering application. *Journal of Sensors*, v. 2016, p. 1–11, 2016. ISSN 1687-725X, 1687-7268. Disponível em: <<http://www.hindawi.com/journals/js/2016/9108197/>>.
- Ying 2019 YING, X. An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, v. 1168, p. 022022, fev. 2019. ISSN 1742-6588, 1742-6596. Disponível em: <<https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022>>.
- Yoon 2009 YOON, S. Vibration isolation and shock protection for mems. In: . [S.l.: s.n.], 2009.
- Zarita e Ong 2009 ZARITA, Z.; ONG, P. Function approximation using artificial neural networks. *WSEAS Transactions on Mathematics*, p. 1160–1178, jan. 2009. Disponível em: <<https://dl.acm.org/doi/10.5555/1466915.1466916>>.
- Zhang et al. 2017 ZHANG, R. et al. Adaptive zero velocity update based on velocity classification for pedestrian tracking. *IEEE Sensors Journal*, v. 17, n. 7, p. 2137–2145, abr. 2017. ISSN 1530-437X, 1558-1748, 2379-9153. Disponível em: <<http://ieeexplore.ieee.org/document/7847379/>>.
- Zheng e Zhang 2016 ZHENG, K.; ZHANG, Q. Comprehensive analysis of the position error and vibration characteristics of Delta robot. *Advanced Robotics*, v. 30, n. 20, p. 1322–1340, out. 2016. ISSN 0169-1864, 1568-5535. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/01691864.2016.1205512>>.
- Zhong et al. 1996 ZHONG, X.-L. et al. Autonomous robot calibration using a trigger probe. *Robotics and Autonomous Systems*, v. 18, n. 4, p. 395–410, out. 1996. ISSN 09218890. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/0921889096000115>>.

Zhou et al. 2016 ZHOU, B. et al. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

Zhou et al. 2017 ZHOU, Z. et al. Integrated navigation system for a low-cost quadrotor aerial vehicle in the presence of rotor influences. *Journal of Surveying Engineering*, v. 143, n. 1, p. 05016006, fev. 2017. ISSN 0733-9453, 1943-5428. Disponível em: <<http://ascelibrary.org/doi/10.1061/%28ASCE%29SU.1943-5428.0000194>>.

Zhuang e Roth 1993 ZHUANG, H.; ROTH, Z. A linear solution to the kinematic parameter identification of robot manipulators. *IEEE Transactions on Robotics and Automation*, v. 9, n. 2, p. 174–185, abr. 1993. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/238281/>>.

Zhuang et al. 1990 ZHUANG, H. et al. A complete and parametrically continuous kinematic model for robot manipulators. In: *Proceedings., IEEE International Conference on Robotics and Automation*. Cincinnati, OH, USA: IEEE Comput. Soc. Press, 1990. p. 92–97. ISBN 9780818690617. Disponível em: <<http://ieeexplore.ieee.org/document/125952/>>.

Zhuang et al. 1992 ZHUANG, H. et al. A complete and parametrically continuous kinematic model for robot manipulators. *IEEE Transactions on Robotics and Automation*, v. 8, n. 4, p. 451–463, ago. 1992. ISSN 1042296X. Disponível em: <<http://ieeexplore.ieee.org/document/149944/>>.

Zhuang e Roth 1995 ZHUANG, H.; ROTH, Z. S. A tutorial on robot calibration with hand-mounted cameras. *Intelligent Automation & Soft Computing*, v. 1, n. 3, p. 325–343, jan. 1995. ISSN 1079-8587, 2326-005X. Disponível em: <<http://autosoftjournal.net/paperShow.php?paper=10750638>>.

Zhuang e Roth 2018 ZHUANG, H.; ROTH, Z. S. *Camera-aided robot calibration*. [S.l.: s.n.], 2018. OCLC: 1103970560. ISBN 9781315138725.

Zhuang et al. 1993 ZHUANG, H. et al. Error-model-based robot calibration using a modified CPC model. *Robotics and Computer-Integrated Manufacturing*, v. 10, n. 4, p. 287–299, ago. 1993. ISSN 07365845. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/073658459390042I>>.

Zollo et al. 2015 ZOLLO, L. et al. Identification of Dynamic Parameters for Robots with Elastic Joints. *Advances in Mechanical Engineering*, v. 7, n. 2, p. 843186, jan. 2015. ISSN 1687-8140, 1687-8140. Disponível em: <<http://journals.sagepub.com/doi/10.1155/2014/843186>>.

Şirinterlikçi et al. 2009 ŞİRINTERLIKÇİ, A. et al. Repeatability and accuracy of an industrial robot: Laboratory experience for a design of experiments course. *Technology Interface Journal/Spring 2009*, v. 78, 2009. Disponível em: <[https://www.researchgate.net/publication/260336817\\_Repeatability\\_and\\_Accuracy\\_of\\_an\\_Industrial\\_Robot\\_Laboratory\\_Experience\\_for\\_a\\_Design\\_of\\_Experiments\\_Course](https://www.researchgate.net/publication/260336817_Repeatability_and_Accuracy_of_an_Industrial_Robot_Laboratory_Experience_for_a_Design_of_Experiments_Course)>.