



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

DARIELSON ARAÚJO DE SOUZA

**NOVA ABORDAGEM DE APROXIMAÇÃO DE MODELOS BASEADA EM
APRENDIZADO DE MÁQUINA APLICADA À UM MANIPULADOR ROBÓTICO**

FORTALEZA

2022

DARIELSON ARAÚJO DE SOUZA

NOVA ABORDAGEM DE APROXIMAÇÃO DE MODELOS BASEADA EM
APRENDIZADO DE MÁQUINA APLICADA À UM MANIPULADOR ROBÓTICO

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

Orientadora: Prof^a. Dr^a. Laurinda Lúcia Nogueira dos Reis

FORTALEZA

2022

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

S237n Souza, Darielson Araújo de.

Nova abordagem de aproximação de modelos baseada em aprendizado de máquina aplicada à um manipulador robótico / Darielson Araújo de Souza. – 2022.

136 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2022.

Orientação: Profa. Dra. Laurinda Lúcia Nogueira dos Reis.

1. Manipulador robótico cilíndrico. 2. Aprendizagem de Máquina. 3. Modelos cinemáticos e dinâmicos.
I. Título.

CDD 621.3

DARIELSON ARAÚJO DE SOUZA

NOVA ABORDAGEM DE APROXIMAÇÃO DE MODELOS BASEADA EM
APRENDIZADO DE MÁQUINA APLICADA À UM MANIPULADOR ROBÓTICO

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

Aprovada em: 26 de Maio de 2022

BANCA EXAMINADORA

Prof^ª. Dr^ª. Laurinda Lúcia Nogueira dos
Reis (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Arthur Plínio de Souza Braga
Universidade Federal do Ceará (UFC)

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

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

Prof. Dr. Antonio Barbosa de Souza Júnior
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará (IFCE)

Aos meus pais, irmãos e todos os amigos que
acompanharam o meu trabalho.

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Antônio de Pádua e Antônia Aparecida pelo amor, carinho e determinação.

Aos meus queridos irmãos Denis, Diego e Danielle, pelos seus exemplos.

A minha noiva Larissa, pelo companheirismo e carinho.

A minha orientadora Prof^a. Dr^a. Laurinda dos Reis, pela paciência e excelente orientação.

Ao professor Antônio Barbosa, pelas dúvidas e sempre disposto a ajudar.

Aos professores participantes da banca examinadora pelo tempo, pelas valiosas colaborações e sugestões.

Aos colegas do GPAR, pelas reflexões, críticas e sugestões recebidas em especial: Josias Batista, Renê Olímpio e José Raimundo.

A todos os professores que fui aluno durante o doutorado e a Coordenação do programa.

À CAPES, pelo apoio financeiro com a manutenção da bolsa de auxílio.

“Atitude é uma coisa pequena que faz uma grande
diferença.”

(Winston Churchil)

RESUMO

Algoritmos baseados em Inteligência Computacional estão sendo aplicados na área de robótica para solucionar problemas referentes à modelagem. Além disso, algoritmos de aprendizado de máquina também estão sendo indicados em uma grande variedade de aplicações de sistemas. Nesta tese, com o uso de algoritmos de aprendizado de máquina, propõe-se um novo algoritmo de aproximação de modelos para um manipulador robótico do tipo cilíndrico. Inicialmente, é descrita a modelagem matemática de manipuladores, levando em consideração os modelos cinemáticos e dinâmicos. Também alguns algoritmos de regressão presentes na literatura são abordados e aplicados ao sistema, bem como o algoritmo proposto baseado em máquinas de comitê também foi aplicado. Este algoritmo usa como base o método de regressão de vetor suporte (SVR), onde um comitê escolhe um entre alguns dos três tipos de estruturas de SVRs com os kernels: gaussiano, polinomial e sigmoideal. A seleção é feita por um critério baseado no erro. Para tanto, descreve-se a modelagem matemática de manipuladores levando-se em consideração os modelos cinemáticos e dinâmicos os quais também são obtidos para o manipulador cilíndrico acionado por motores de indução trifásicos montado. Uma comparação entre os vários métodos apresentados é realizada utilizando-se dados de simulação e experimental obtidos do manipulador cilíndrico, e em seguida é feita a validação cruzada para todos os métodos. Esta comparação é realizada por meio de tarefas executadas pelo manipulador com cargas constante e variável. As tarefas com carga variável têm o objetivo de simular possíveis problemas em produtos durante a sua realização. Finalmente, os resultados são comparados por métricas de avaliação como raiz do erro quadrático médio (RMSE, do inglês: Root Mean Squared Error) e coeficiente de determinação ajustável (R_a^2), que confirmam que o método proposto apresentou os melhores resultados na maioria dos cenários.

Palavras-chave: manipulador robótico cilíndrico; modelos cinemáticos e dinâmicos; modelagem de trajetórias; aprendizagem de máquina; comitê; carga variável.

ABSTRACT

Algorithms based on Computational Intelligence are being applied in robotics to solve modeling problems. In addition, machine learning algorithms are also being indicated in a wide variety of systems applications. In this thesis, using machine learning algorithms, we propose a new model approximation algorithm for a cylindrical robotic manipulator. Initially, the mathematical modeling of manipulators is described, taking into account the kinematic and dynamic models. Also some regression algorithms present in the literature are approached and applied to the system, as well as the proposed algorithm based on committee machines was also applied. This algorithm is based on the support vector regression (SVR) method, where a committee chooses one of three types of SVR structures with kernels: gaussian, polynomial and sigmoidal. The selection is made by an error-based criterion. Therefore, the mathematical modeling of manipulators is described, taking into account the kinematic and dynamic models which are also obtained for the cylindrical manipulator driven by three-phase induction motors mounted on the GPAR. A comparison between the various methods presented is performed using simulation and experimental data obtained from the GPAR cylindrical manipulator, and then cross-validation is performed for all methods. This comparison is performed through tasks performed by the manipulator with constant and variable loads. Tasks with variable load have the objective of simulating possible problems in products during their execution. Finally, the results are compared by evaluation metrics such as root mean square error (RMSE) and adjustable coefficient of determination (R_a^2), which confirm that the proposed method presented the best results in most scenarios.

Keywords: cylindrical robotic manipulator; kinematic and dynamic models; trajectory modeling; machine learning; committee; variable load.

LISTA DE FIGURAS

Figura 1 – Previsão do número anual de robôs fornecidos para fábricas no mundo.	24
Figura 2 – Sistema de controle de posição no espaço cartesiano.	29
Figura 3 – Coordenadas de estruturas ligadas ao manipulador.	30
Figura 4 – Desacoplamento cinemático.	35
Figura 5 – Corpo rígido geral.	41
Figura 6 – Sistema de referências geométricas de um manipulador cilíndrico.	48
Figura 7 – Estrutura do manipulador cilíndrico - <i>Software Solid Edge®</i>	57
Figura 8 – Propriedades físicas da junta 2 do manipulador - <i>Software Solid Edge®</i> . . .	57
Figura 9 – Propriedades físicas da junta 3 do manipulador - <i>Software Solid Edge®</i> . . .	58
Figura 10 – Projeção da primeira junta $x_0 - y_0$	58
Figura 11 – Configuração do acionamento de campo orientado indireto para uma máquina de indução.	60
Figura 12 – Estrutura da RNA MLP de duas camadas ocultas.	64
Figura 13 – Arquitetura de uma RNA RBF.	66
Figura 14 – Arquitetura da Rede ELM.	69
Figura 15 – Arquitetura da rede ANFIS.	72
Figura 16 – Fluxograma do comitê proposto.	74
Figura 17 – Estrutura da máquina de comitê proposta.	75
Figura 18 – SVM	79
Figura 19 – Regressão da cinemática e dinâmica inversa a partir dos dados gerados por simulação.	84
Figura 20 – Trajetória helicoidal no espaço cartesiano.	84
Figura 21 – Posições no espaço das juntas durante a trajetória helicoidal.	85
Figura 22 – Velocidades no espaço das juntas durante a trajetória helicoidal	85
Figura 23 – Aceleração no espaço das juntas durante a trajetória helicoidal	86
Figura 24 – Torque no espaço das juntas durante a trajetória helicoidal	86
Figura 25 – Trajetória elíptica no espaço cartesiano.	87
Figura 26 – Posições no espaço das juntas durante a trajetória elíptica.	87
Figura 27 – Velocidades no espaço das juntas durante a trajetória elíptica.	88
Figura 28 – Acelerações no espaço das juntas durante a trajetória elíptica.	88
Figura 29 – Torques no espaço das juntas durante a trajetória elíptica.	89

Figura 30 – Manipulador com carga variável	89
Figura 31 – Predição da rede MLP das posições no espaço das juntas durante a trajetória helicoidal.	90
Figura 32 – Predição da rede MLP dos torques da trajetória helicoidal.	91
Figura 33 – Predição da rede MLP das posições no espaço das juntas na trajetória elíptica.	92
Figura 34 – Predição da rede MLP dos torques da trajetória elíptica.	93
Figura 35 – Predição da rede MLP dos torques da trajetória helicoidal com carga variável.	93
Figura 36 – Predição da rede MLP dos torques da trajetória elíptica com carga variável.	94
Figura 37 – Predição da rede RBF das posições no espaço das juntas na trajetória helicoidal.	95
Figura 38 – Predição da rede RBF dos torques da trajetória helicoidal.	95
Figura 39 – Predição da rede RBF das posições no espaço das juntas na trajetória elíptica.	96
Figura 40 – Predição da rede RBF dos torques da trajetória elíptica.	97
Figura 41 – Predição da rede RBF dos torques da trajetória Helicoidal com carga variável.	97
Figura 42 – Predição da rede RBF dos torques da trajetória elíptica com carga variável.	98
Figura 43 – Predição da rede ELM das posições no espaço das juntas na trajetória helicoidal.	99
Figura 44 – Predição da rede ELM dos torques da trajetória helicoidal.	99
Figura 45 – Predição da rede ELM das posições no espaço das juntas na trajetória elíptica.	100
Figura 46 – Predição da rede ELM dos torques da trajetória elíptica.	101
Figura 47 – Predição da rede ELM dos torques da trajetória helicoidal com carga variável.	101
Figura 48 – Predição da rede ELM dos torques da trajetória elíptica com carga variável.	102
Figura 49 – Predição do ANFIS das posições no espaço das juntas na trajetória helicoidal.	103
Figura 50 – Predição do ANFIS dos torques da trajetória helicoidal.	103
Figura 51 – Predição do ANFIS das posições no espaço das juntas na trajetória elíptica.	104
Figura 52 – Predição do ANFIS dos torques da trajetória elíptica.	105
Figura 53 – Predição do ANFIS dos torques da trajetória helicoidal com carga variável.	105
Figura 54 – Predição do ANFIS dos torques da trajetória elíptica com carga variável.	106
Figura 55 – Predição do comitê nas posições no espaço das juntas na trajetória helicoidal.	107
Figura 56 – Predição do comitê nos torques da trajetória helicoidal.	107
Figura 57 – Predição do comitê nas posições no espaço das juntas na trajetória elíptica com carga constante.	108
Figura 58 – Predição do comitê nos torques da trajetória elíptica com carga constante.	108
Figura 59 – Predição do comitê nos torques da trajetória helicoidal com carga variável.	109

Figura 60 – Predição do ANFIS dos torques da trajetória elíptica com carga variável. . .	109
Figura 61 – Predição dos torques usando os modelos na trajetória helicoidal com carga constante.	110
Figura 62 – Predição dos torques usando os modelos na trajetória elíptica com carga constante.	111
Figura 63 – Predição dos torques usando os modelos na trajetória helicoidal com carga variável.	112
Figura 64 – Predição dos torques usando os modelos na trajetória helicoidal	113
Figura 65 – Manipulador robótico cilíndrico do GPAR.	114
Figura 66 – Circuitos para controle e acionamento do manipulador	115
Figura 67 – Regressão para a obtenção do mapeamento da cinemática inversa a partir dos dados experimentais.	116
Figura 68 – Espaço cartesiano com o movimento helicoidal dos dados reais.	117
Figura 69 – Trajetória no espaço das juntas da posição durante o movimento helicoidal com dados reais.	118
Figura 70 – Espaço cartesiano com o movimento elíptico com dados reais.	118
Figura 71 – Trajetória no espaço das juntas da posição durante o movimento elíptico com dados reais.	119
Figura 72 – Predição da rede MLP das posições reais no espaço das juntas na trajetória espiral	120
Figura 73 – Predição da rede MLP das posições reais no espaço das juntas na trajetória elíptica	120
Figura 74 – Predição da rede RBF das posições reais no espaço das juntas na trajetória espiral	121
Figura 75 – Predição da rede RBF das posições reais no espaço das juntas na trajetória elíptica	122
Figura 76 – Predição da rede ELM das posições reais no espaço das juntas na trajetória espiral	122
Figura 77 – Predição da rede ELM das posições reais no espaço das juntas na trajetória elíptica	123
Figura 78 – Predição da rede ANFIS das posições reais no espaço das juntas na trajetória espiral	123

Figura 79 – Predição da rede ANFIS das posições reais no espaço das juntas na trajetória elíptica	124
Figura 80 – Predição do Comitê das posições reais no espaço das juntas na trajetória espiral	124
Figura 81 – Predição do Comitê das posições reais no espaço das juntas na trajetória elíptica	125

LISTA DE TABELAS

Tabela 1 – Parâmetros de ligação do manipulador cilíndrico.	49
Tabela 2 – Parâmetros D-H do manipulador do GPAR.	55
Tabela 3 – Valores de massas (m) e comprimentos (l) das juntas.	58
Tabela 4 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque de todos os métodos para trajetória helicoidal com carga constante.	110
Tabela 5 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico do torque de todos os métodos com carga constante.	111
Tabela 6 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento helicoidal com carga variável.	112
Tabela 7 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento elíptico com carga variável.	113
Tabela 8 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento helicoidal das trajetórias com dados reais.	125
Tabela 9 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico das trajetórias com dados reais.	126

LISTA DE ALGORITMOS

Algoritmo 1 – Treinamento dos pesos da rede MLP.	64
Algoritmo 2 – Treinamento dos centros com <i>k-means</i>	68
Algoritmo 3 – Treinamento da ELM.	70

LISTA DE ABREVIATURAS E SIGLAS

R_a^2	coeficiente de determinação ajustável
ABC	Colônia de abelhas baseada em conhecimento (<i>knowledge-based artificial bee colony</i>)
ANFIS	<i>Adaptive Network based Fuzzy Inference System</i>
DH	<i>Denavit Hartenberg</i>
DSP	<i>Digital Signal Processors</i>
ELM	<i>Extreme Learning Machine</i>
FOC	<i>Field Oriented Control</i>
GDL	graus de liberdade
GPAR	Grupo de Pesquisa em Automação Controle e Robótica
IFR	<i>International Federation of Robotics</i>
MLP	<i>Multilayer Perceptron</i>
RBF	<i>Radial Basis Function</i>
RMSE	Raiz do erro quadrático médio
SVPWM	<i>Spatial Vector Pulse Width Modulation</i>
SVR	<i>Support Vector Regression</i>
UFC	Universidade Federal do Ceará
VSI	<i>Voltage Source Inverter</i>

LISTA DE SÍMBOLOS

A	Ampere
A_i	Matriz de transformação
B	Coefficiente de atrito viscoso
C_v	Cavalo vapor
d	Vetor de respostas desejadas
d_i	Junta i linear
d_q^0	Frequência calculada nas
$e(t)$	Perturbação
$f(x_i)$	Saída da rede neural para um vetor de entrada x_i
G_c	Bloco de controle de velocidade
G_l^s	Sistema em malha fechada
Hz	Frequência do sinal
i	Junção i associada a um elo
i_a^*, i_b, i_c^*	correntes elétricas
i_q^*	Corrente de quadratura
J	Constante momento de inércia
$K(x, x')$	Função de covariância ou kernel
n	Número de articulações
$o_i x_i y_i z_i$	Quadro de coordenadas associado a um elo i
O_n^0	Vetor de coordenadas de referencia do <i>end-effector</i> no que diz respeito ao quadro base
q_i	Articulação i
r	Tamanho da escala
T_a, T_b, T_c	Período para chaveamento dos circuitos de potência
T_e	Torque eletromagnético
T_j^i	Matriz de transformação
U_{out}	Tensão de referência

$u(t)$	Entrada do sistema
$y(t)$	Saída do sistema
V	Volts
v_o	Variância do sinal
v_1	Variância da bias
v_2	Variância do ruído
X	Eixo do plano cartesiano
x_i	Dados de entrada do GP
$X(0 : k)^i$	Partículas
Y	Eixo do plano cartesiano
y_i	Dados de saída do GP
w	Vetor de pesos das RNAs
w_{ik}	Pesos associados as partículas
Z	Eixo do plano cartesiano
z^{-d}	Atraso do sistema
β	Pesos de conexão da camada oculta
ε	Erro de previsão
ε_R	Função custo rede neural RBF
θ	Rotação em torno do eixo x
θ_i	Junta i rotativa
θ_r^*	Comando de posição do rotor
θ^T	Vetor de parâmetros
λ_j	Parâmetro de regularização
λ_q^r	Fluxo de dispersão
$\mu(x)$	Função média de parametrização para processo gaussiano
ρ	Rugosidade/largura da função radial
σ	Desvio padrão
ϕ	Matriz de regressores

ϕ^T	Vetor de medidas
Φ	Matriz de interpolação
φ	Rotação em torno do eixo y
ψ	Rotação em torno do eixo z
ω_r^*	Comando de velocidade do rotor
ω_s^l	Frequência de escorregamento

SUMÁRIO

1	INTRODUÇÃO	22
1.1	Estado da Arte	22
1.2	Justificativa	24
1.3	Objetivos	25
<i>1.3.1</i>	<i>Objetivo Geral</i>	25
<i>1.3.2</i>	<i>Objetivos Específicos</i>	25
<i>1.3.3</i>	<i>Contribuições do Trabalho</i>	26
1.4	Organização do Texto	27
2	MODELAGEM MATEMÁTICA DO MANIPULADOR ROBÓTICO	29
2.1	Descrição de um manipulador	29
2.2	Modelos Cinemáticos	29
<i>2.2.1</i>	<i>Cinemática Direta</i>	30
<i>2.2.1.1</i>	<i>Convenção de Denavit-Hartenberg</i>	31
<i>2.2.2</i>	<i>Cinemática Inversa</i>	32
<i>2.2.2.1</i>	<i>Desacoplamento cinemático</i>	33
<i>2.2.2.2</i>	<i>Abordagem Geométrica</i>	35
2.3	Jacobiano de Manipuladores	36
<i>2.3.1</i>	<i>Derivação do Jacobiano</i>	36
<i>2.3.2</i>	<i>Velocidade angular</i>	37
<i>2.3.3</i>	<i>Velocidade linear</i>	38
<i>2.3.4</i>	<i>Combinando as jacobianas angular e linear</i>	38
2.4	Dinâmica de manipuladores	39
<i>2.4.1</i>	<i>Equações de Euler-Lagrange</i>	39
<i>2.4.2</i>	<i>Expressões gerais para energia cinética e potencial</i>	40
<i>2.4.2.1</i>	<i>Energia cinética para um robô de n elos</i>	41
<i>2.4.2.2</i>	<i>Energia potencial para um robô de n elos</i>	42
<i>2.4.3</i>	<i>Equações de Movimento</i>	42
2.5	Dinâmica Inversa	43
<i>2.5.1</i>	<i>Dinâmica Inversa do Espaço de Tarefas</i>	46
2.6	Manipulador Cilíndrico	48

2.6.1	<i>Jacobiana do Manipulador Cilíndrico</i>	49
2.6.2	<i>Dinâmica do Manipulador Cilíndrico</i>	51
2.6.2.1	<i>Energia Cinética</i>	51
2.6.2.2	<i>Energia potencial</i>	51
2.6.2.3	<i>Equação de Lagrange</i>	52
2.6.2.4	<i>Dinâmica do manipulador cilíndrico na forma matricial</i>	54
2.6.2.5	<i>Equações dinâmicas na forma de espaço de estados</i>	54
2.6.3	<i>Manipulador cilíndrico do GPAR</i>	55
2.6.3.1	<i>Cinemática Inversa do manipulador do GPAR</i>	57
2.6.3.2	<i>Modelagem do motor de indução trifásico como servo-posicionador</i>	59
2.6.3.3	<i>Modelagem dinâmica do campo orientado indireto para uma máquina de indução</i>	59
2.6.3.4	<i>Modulação por largura de pulso usando vetores espaciais</i>	61
2.7	Considerações finais	62
3	ALGORITMOS DE REGRESSÃO	63
3.1	Considerações iniciais	63
3.2	Rede Neural <i>Multilayer Perceptron</i> (MLP)	63
3.3	Rede neural de função de base radial	65
3.4	Rede Neural ELM	68
3.5	ANFIS	71
3.6	Método de Regressão Proposto	73
3.6.1	<i>Máquinas de Comitê</i>	74
3.6.2	<i>Regressão de Vetor Suporte</i>	75
3.6.2.1	<i>Mapeamento via Kernel</i>	79
3.7	Considerações finais	80
4	RESULTADOS	81
4.1	Considerações iniciais	81
4.2	Validação-cruzada	81
4.3	Métricas de avaliação	82
4.4	Resultados de simulações	83
4.4.1	<i>Rede MLP</i>	90
4.4.2	<i>Rede RBF</i>	94

4.4.3	<i>Rede ELM</i>	98
4.4.4	<i>ANFIS</i>	102
4.4.5	<i>Método proposto</i>	106
4.4.6	<i>Comparação entre os métodos com carga constante</i>	110
4.4.7	<i>Comparação entre os métodos com carga variável</i>	111
4.5	Resultados com dados experimentais	114
4.5.1	<i>Rede MLP com dados reais</i>	119
4.5.2	<i>Rede RBF com dados reais</i>	121
4.5.3	<i>Rede ELM</i>	122
4.5.4	<i>ANFIS</i>	123
4.5.5	<i>Comitê de SVRs</i>	124
4.5.6	<i>Comparação entre os métodos com dados reais</i>	125
5	CONCLUSÕES	127
5.1	Trabalhos futuros	128
	REFERÊNCIAS	129
	APÊNDICES	134
	APÊNDICE A – Resultados de simulação com os comitês dos métodos com a carga constante	134
	APÊNDICE B – Resultados de simulação com os comitês dos métodos com a carga variável	135
	APÊNDICE C – Resultados com dados experimentais com os comitês dos métodos	136

1 INTRODUÇÃO

A automação industrial está cada vez mais ganhando destaque em cenários e situações em que há perigo para o ser humano. Manipuladores robóticos, robôs móveis e VANTs (Veículos Aéreos não tripulados) são agentes que podem substituir o homem na perspectiva de risco de vida, porém eles precisam ser projetados para realizar operações de forma eficaz em situações adversas, tanto quanto a sua estrutura física, quanto a seus algoritmos.

Portanto, este capítulo trata de uma contextualização do estado da arte do problema abordado, justificativa, objetivos e contribuições da pesquisa. Também apresentam-se as publicações aceitas e submetidas oriundas deste trabalho, assim como a organização do texto.

1.1 Estado da Arte

Na primeira fase da evolução da robótica, máquinas elétricas foram programadas para executar tarefas específicas, mas não interagem com o mundo real. Com o tempo, robôs industriais tornaram-se progressivamente mais sofisticados. Entretanto, o potencial para uma aceitação industrial e de consumidores está ligado ao desenvolvimento e aos avanços atuais ocorridos na segunda fase da evolução da robótica. Nesta fase, robôs não são mais simplesmente programados para executar tarefas repetitivas. Eles também podem fazer aquisição de dados, reconhecer objetos e responder a informações e objetos em seu ambiente com maior precisão. Atualmente, muitas empresas já utilizam robôs industriais em muitas áreas, tais como fabricação de automóveis, centros de distribuição, fundições, embalagens farmacêuticas, dentre outros (ZUNT, 2013).

Em contrapartida ainda existem muitos problemas na robótica que os pesquisadores tentam solucionar. Um deles é a modelagem de acordo com o aumento do número de graus de liberdade, tornando-a mais complexa. Nos últimos anos, o problema de modelagem de configuração de manipuladores vem sendo objeto de pesquisa e desta forma, o surgimento de novos métodos de solução. Por exemplo, ZHAO *et al.* (2018) apresentam uma solução do problema de ajuste de trajetórias de um manipulador de 6 graus de liberdade usando métodos analíticos, geométricos e algébricos combinados com o subproblema de Paden-Kahan. O trabalho de JIANGPING *et al.* (2019) propõe um algoritmo que combina o método de planejamento de trajetórias baseado em amostragem com o cálculo analítico da cinemática inversa. Além disso, o algoritmo aleatoriamente obtém configurações satisfatórias de restrições usando o solucionador

analítico da modelagem e conecta incrementalmente aos caminhos de movimento no espaço das juntas. Outro trabalho de destaque é o de GUPTA *et al.* (2018), em que foi desenvolvida e apresentada uma abordagem geométrica para calcular as configurações de um braço robótico de 3 graus de liberdade. A proposta de WEI e RAHMANI (2018) foi o desenvolvimento de uma estratégia de rastreamento e trajetórias de movimentos semi-descentralizados baseados na matriz jacobiana relacionada a manipulação de objetos.

Algoritmos oriundos de Inteligência Computacional, como redes neurais, sistemas *fuzzy*, metaheurísticas, estão também sendo aplicados na área de robótica, tentando solucionar a mesma problemática dos trabalhos citados anteriormente. O trabalho apresentado em Zhou *et al.* (2018) aborda uma solução para obtenção das configurações de um manipulador onde utiliza uma rede *Extreme Learning Machine* (ELM) otimizada com algoritmo genético e apresentando uma espécie de mutação sequencial. O trabalho apresentado por SHERBINY e ELHOSSEINI M. A. and HIAKAL (2018) usa um algoritmo baseado no algoritmo de colônias de abelhas artificial (Colônia de abelhas baseada em conhecimento (*knowledge-based artificial bee colony*) (ABC)) para encontrar as melhores configurações de um manipulador de 5 graus de liberdade e depois faz uma comparação com outras técnicas clássicas, tais como GA, PSO, etc. Já CARLOS *et al.* (2018) usam uma metaheurística de evolução diferencial para a modelagem de um manipulador móvel em que a abordagem é capaz de evitar configurações de singularidades, uma vez que não requer a inversão de matriz jacobiana. O trabalho de ONDER *et al.* (2017) propõe um estudo de aprendizagem por reforço para o desenvolvimento de um sistema de equilíbrio do corpo inferior de um robô NAO que possui 12 graus de liberdade. Em VU *et al.* (2018) é apresentado um método baseado em sistemas de Inferência *Neuro-Fuzzy* (*Adaptive Network based Fuzzy Inference System* (ANFIS)) para gerar trajetória para o braço de uma escavadeira em que as trajetórias são predeterminadas com alguns pontos específicos no espaço de trabalho.

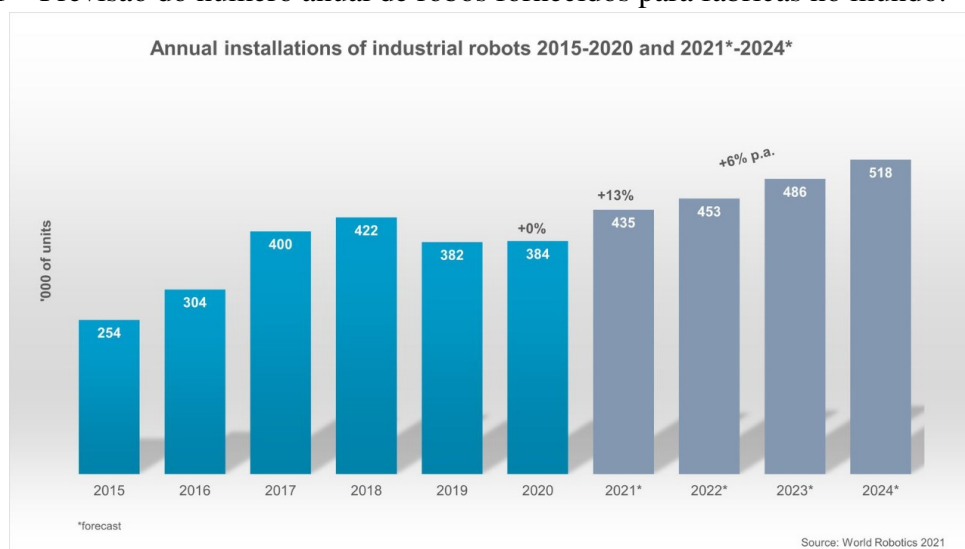
Diante do exposto acima, a realização de tarefas usando manipuladores é importante para a área da robótica. No entanto, o ajuste de trajetórias para a realização destas tarefas ainda é uma questão que apresenta vários desafios teóricos e práticos para a área. Isso se deve principalmente à dinâmica complexa desses robôs e à necessidade de se obter uma trajetória precisa, na perspectiva de cargas variáveis, como apresentado em (SHARMA *et al.*, 2016).

1.2 Justificativa

Existe uma contínua necessidade de melhorar os modelos matemáticos e métodos de controle, a fim de atender aos requisitos conflitantes, como o aumento do desempenho de um robô com redução de peso, com menor rigidez mecânica e modos de vibração mais complexos. Consequentemente, tem-se a redução de custos e outros benefícios, como menor impacto ambiental e consumo de energia, além de maior habilidade e segurança. Portanto, várias propostas de métodos estão sendo testadas constantemente para realizar tarefas de forma eficaz, tornando-se um problema de difícil solução à longo prazo, pois além de resolver a problemática é preciso monitorar as trajetórias das juntas do robô durante o seu uso. Também são desafiadoras situações em que um manipulador precisa realizar tarefas com cargas de diferentes tamanhos e dimensões, além de desviar de obstáculos.

Conforme descrito acima observa-se que a utilização de robôs em diversas áreas de aplicação vem crescendo de forma exponencial. A partir da Figura 1 apresentada no site da *International Federation of Robotics* (IFR), alguns pesquisadores fazem uma previsão da quantidade de robôs até 2024 (IFR, 2021).

Figura 1 – Previsão do número anual de robôs fornecidos para fábricas no mundo.



Fonte: IFR (2021).

Segundo Junji Tsuda, presidente da IFR, os robôs industriais são parte do progresso da indústria manufatureira. Podendo também executar tarefas de reconhecimento de imagens, aprendizado de habilidades, previsão de falhas, realização de tarefas com cargas variáveis, entre outras. Assim, originando a solução de problemas relacionados à área de robótica.

1.3 Objetivos

Este trabalho propõe um método de regressão baseado em máquinas de comitê (*Committe*), fazendo a combinação de algumas estruturas de máquina de vetor de suporte para regressão (*Support Vector Regression (SVR)*) e uma rede Extreme Learning Machine (ELM). A proposta resolve a problemática do rastreamento de trajetória de um manipulador de modo que este faça correções de forma autônoma. Assim, com um método que visa a aproximação do modelo real de um manipulador, o algoritmo pode aprender através de dados coletados de modo a garantir que variações nas condições ambientais serão percebidas e que o manipulador se adaptará às novas condições de operações. Pode-se verificar ainda que os trabalhos citados na Seção 1.1 fazem a aproximação do modelo de robôs que utilizam diferentes tipos de motores, motor CC e motor de passo. Assim, neste trabalho utilizam-se motores de indução trifásicos, sendo uma das novidades da pesquisa na área da robótica, e encontra-se em desenvolvimento no Laboratório do Grupo de Pesquisa em Automação Controle e Robótica (GPAR) do Departamento de Engenharia Elétrica - Universidade Federal do Ceará (UFC).

Assim, neste trabalho utilizam-se motores de indução trifásicos, sendo uma das novidades de pesquisa na área de robótica em que é utilizado no manipulador usado neste trabalho encontra-se em desenvolvimento no Laboratório do GPAR do Departamento de Engenharia Elétrica - UFC.

1.3.1 *Objetivo Geral*

Aplicar uma nova abordagem de um algoritmo de regressão baseado em aprendizado de máquina para solucionar o problema de rastreamento de trajetórias de um manipulador robótico com carga variável é o objetivo de pesquisa deste trabalho.

1.3.2 *Objetivos Específicos*

- Utilizar um manipulador robótico de dois graus de liberdade acionados com motores de indução trifásicos e também um manipulador simulado, baseado no modelo real, mas com 3-graus de liberdade (GDL);
- Implementar um método de regressão baseado em máquinas de comitê com SVR para realizar a aproximação dos modelos cinemático e dinâmico para o rastreamento das trajetórias do manipulador robótico;

- Propor a execução de trajetórias comumente usadas em ambientes industriais para que o método proposto possa ser avaliado;
- Adicionar cargas variáveis nos manipuladores, para tornar mais complexa a modelagem, para que dessa forma seja simulado um objeto com massa variável durante o transporte;
- Realizar uma análise comparativa com outros métodos clássicos da literatura em regressão com aprendizado de máquina.

1.3.3 Contribuições do Trabalho

As principais publicações relacionadas a este trabalho foram:

1. Souza, D. A., Reis, L. L., Batista, J. G., Costa, J. R., Junior, A. B., Araújo, J. P., & Braga, A. P. Nonlinear Identification of a Robotic Arm Using Machine Learning Techniques. *Advances in Intelligent Systems and Computing*. Springer, v. 2, p. 492-501, 2019. (Publicado);
2. Souza, D., Batista, J., dos Reis, L., Santana, K., Júnior, J. R., Júnior, J. N., ... & Araújo, R. (2019, September). Short Paper: Identification by Recursive Least Squares with RMO Applied to a Robotic Manipulator. In *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)* (pp. 95-96). IEEE. (Publicado);
3. Batista, J. G., Souza, D. A., dos Reis, L. L. N., de Souza Júnior, A. B., & Santana, K. A. Modelo Cinemático e Dinâmico de um Manipulador 3-GDL. *DINCON - Conferência Brasileira de Dinâmica, Controle e Aplicações*. São Carlos, 2019. (Publicado);
4. Batista, J. G., Souza, D. A., dos Reis, L. L., Filgueiras, L. V., Ramos, K. M., Junior, A. B., & Correia, W. B. (2019, October). Performance comparison between the PID and LQR controllers applied to a robotic manipulator joint. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society* (Vol. 1, pp. 479-484). IEEE. (Publicado);
5. Souza, D., Batista, J., Reis, L., & de Souza Junior, A. Model approach of a robotic manipulator 3DOF using Machine Learning. *Journal on Advances in Theoretical and Applied Informatics* 5.1, 2019. (Publicado);
6. Batista J. G., Costa J. R., Souza, D. A., Filgueiras, L. V. O., Júnior, J. N. N., Júnior, A. B. S. e dos Reis, L. L. N. Modelagem Dinâmica e Simulação de um Controlador PID e LQR para um Manipulador Cilíndrico. *14º Simpósio Brasileiro de Automação Inteligente - SBAI, 2019, Ouro Preto. Anais eletrônicos, Campinas, GALOÁ, 2020*. (Publicado);
7. Souza, D.A., Batista J. G., Reis, L. L. N., Júnior, J. N. N., Junior, R. J. O., Ramos, K.

- M. e Júnior, A. B. S. Identificação com uma Variação de Mínimos Quadrados Recursivo aplicado a um Manipulador Robótico. 14º Simpósio Brasileiro de Automação Inteligente - SBAI, 2019, Ouro Preto. Anais eletrônicos, Campinas, GALOÁ, 2020. (Publicado);
8. Batista, J., Souza, D., Dos Reis, L., Barbosa, A., & Araújo, R. Dynamic Model and Inverse Kinematic Identification of a 3-DOF Manipulator Using RLSPSO. *Sensors* 20.2:416, 2020. (Publicado);
 9. Batista, J., Souza, D., Silva, J., Ramos, K., Costa, J., dos Reis, L., & Braga, A. Trajectory Planning Using Artificial Potential Fields with Metaheuristics. *IEEE Latin America Transactions*, 18(05), 914-922, 2020. (Publicado);
 10. Souza, D. A., de Mesquita, V. A., Reis, L. L., Silva, W. A., & Batista, J. G. Optimal LQI and PID Synthesis for Speed Control of Switched Reluctance Motor Using Metaheuristic Techniques. *International Journal of Control, Automation and Systems*, 1-9, 2020. (Publicado);
 11. Souza, D. A., Batista, J. G., Vasconcelos, F. J. S.; Reis, L. L. N.; Machado, G. F.; Costa, J. R.; N., J. N.; Silva, J. L. N.; Rios, C. S. N.; Júnior, A. B. S. Identification by Recursive Least Squares with Kalman filter (RLS-KF) applied to a robotic manipulator. *IEEE Access*, p. 1–12, 2021. (Publicado);
 12. Souza, Darielson A.; Batista, Josias G. ; Dos Reis, Laurinda L. N. ; Júnior, Antônio B. S. . PID controller with novel PSO applied to a joint of a robotic manipulator. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2021. (Publicado).

1.4 Organização do Texto

Este trabalho está estruturado em 5 capítulos, os quais apresentam uma sequência do desenvolvimento do mesmo.

O Capítulo 2 aborda a modelagem matemática dos manipuladores utilizados, apresentando suas cinemáticas direta e inversa, bem como a modelagem dos motores (de indução trifásico e CC) usados como servo-posicionadores.

No Capítulo 3 são apresentados alguns conceitos de aprendizado de máquina, o desenvolvimento da proposta, e a base de dados para os métodos utilizados para se realizar uma análise comparativa.

O Capítulo 4 apresenta os resultados do método proposto sendo comparado com outros comumente usados na literatura, com uma discussão de cada resultado.

O Capítulo 5 apresenta as considerações finais que norteiam o trabalho, abordando os métodos, o sistema robótico e análise de resultados.

2 MODELAGEM MATEMÁTICA DO MANIPULADOR ROBÓTICO

Neste capítulo serão abordados os conceitos e a modelagem de elementos relevantes para este trabalho. Serão apresentados os modelos cinemáticos e dinâmicos do manipulador cilíndrico utilizado neste trabalho montado no GPAR. Também será apresentada a modelagem do motor de indução trifásico usado como servoposicionador do manipulador cilíndrico.

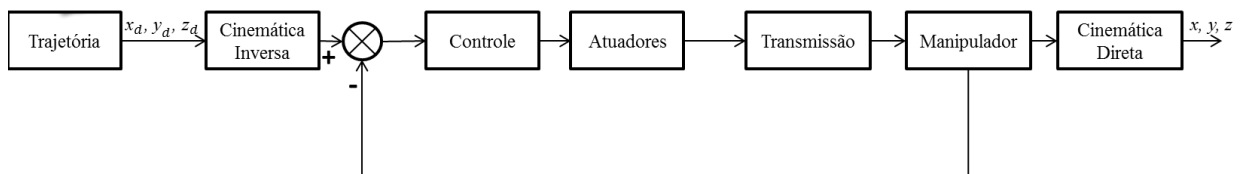
2.1 Descrição de um manipulador

Um manipulador robótico é formado por uma série de segmentos deslizantes ou articulados sendo capaz de mover objetos automaticamente dentro de um determinado número de graus de liberdade. Todos os manipuladores comerciais incluem um controlador e um braço manipulado. O desempenho do manipulador depende de sua velocidade, peso e precisão. No entanto, o alcance de seus *end-effectors* no espaço total de trabalho e a sua orientação são determinados pela estrutura do manipulador (KURFESS, 2004).

Manipuladores são classificados em vários tipos com base na combinação das juntas: braço de geometria cartesiana, braço de geometria cilíndrica, braço de geometria polar/esférica, braço de geometria articulada e braço robótico automático de conformidade seletiva.

Os componentes que compõem um manipulador, geralmente, podem ser amplamente denominados como: braço e punho, atuadores, sensores e controladores, os quais são definidos em (SICILIANO *et al.*, 2008). De um modo geral, a Figura 2 apresenta um exemplo de sistema de controle de posição aplicado a um manipulador no espaço cartesiano.

Figura 2 – Sistema de controle de posição no espaço cartesiano.



Fonte: Adaptada de Santos (2004).

2.2 Modelos Cinemáticos

Os modelos cinemáticos descrevem o movimento sem levar em conta as forças que o causam, ou seja, todas as propriedades geométricas e base de tempo do movimento. A

posição, velocidade, aceleração, são todas descritas pela cinemática. Uma descrição detalhada dos modelos cinemáticos pode ser encontrada em Craig (1989), SPONG e VIDYASAGAR (2008) e Siciliano *et al.* (2008).

2.2.1 Cinemática Direta

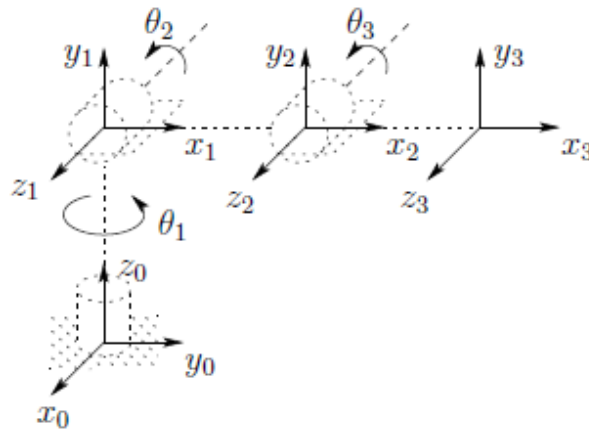
O problema da cinemática direta está relacionado com a relação entre as articulações individuais do robô manipulador e a posição e orientação da ferramenta, dados os valores das variáveis conjuntas do robô. Estas são os ângulos entre os elos no caso de juntas de revolução ou juntas rotacionais, e a extensão do elo no caso de juntas prismáticas ou deslizantes.

Considerando-se um robô manipulador com n articulações e a i -ésima junta denotada por q_i tem-se,

$$q_i = \begin{cases} \theta_i : \text{junta } i \text{ rotativa.} \\ d_i : \text{junta } i \text{ linear.} \end{cases} \quad (2.1)$$

A figura 3 ilustra a ideia de anexar as estruturas aos elos no caso de um manipulador de juntas rotacionais apresentada em (SPONG; VIDYASAGAR, 2008).

Figura 3 – Coordenadas de estruturas ligadas ao manipulador.



Fonte: Adaptada de SPONG e VIDYASAGAR (2008).

Considere que A_i é a matriz de transformação homogênea que expressa a posição e orientação do $o_i x_i y_i z_i$ em relação ao $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. A matriz A_i não é constante e varia conforme a configuração do robô seja alterada. Assim, supondo-se que todas as articulações são de revolução ou prismáticas significa que A_i é uma função de apenas uma única articulação variável, ou seja, q_i . Isto é,

$$A_i = A_i(q_i) \quad (2.2)$$

Por convenção considere uma matriz de transformação, e denotada por T_j^i ,

$$\begin{aligned} T_j^i &= A_{i+1}A_{i+2}\dots A_{j-1}A_j \text{ se } i < j \\ T_j^i &= I \text{ se } i = j \\ T_j^i &= (T_j^i)^{-1} \text{ se } j > i \end{aligned} \quad (2.3)$$

E ainda uma matriz de rotação 3×3 , R_n^0 , que define a matriz de transformação homogênea, ou seja:

$$H = \begin{bmatrix} R_n^0 & O_n^0 \\ 0 & 1 \end{bmatrix}, \quad (2.4)$$

com O_n^0 (que fornece as coordenadas da origem do referencial da ferramenta no que diz respeito ao quadro base) e cuja posição e orientação da ferramenta no quadro inercial são dadas por:

$$H = T_n^0 = A_i(q_i)\dots A_n(q_n) \quad (2.5)$$

2.2.1.1 Convenção de Denavit-Hartenberg

A partir da equação (2.2) utilizou-se a convenção Denavit Hartenberg *Denavit Hartenberg* (DH) (HARTENBERG; DANAVIT, 1964) que fornece a posição e a orientação do *end-effector* no espaço num sistema mecânico articulado com "i" graus de liberdade, atribuindo um sistema de coordenadas. Considerando-se as quatro grandezas θ_i , a_i , d_i , α_i são parâmetros associados com elo i e a junta i sendo comprimento do elo, torção do elo, desvio de ligação e ângulo de junta, respectivamente na equação (2.7).

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (2.6)$$

$$\begin{aligned} A_i &= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.7)$$

A matriz (2.6) é uma função de uma única variável com três dos quatro parâmetros constantes para um determinado elo, enquanto que o quarto parâmetro θ_i representa uma revolução da junta e d_i é a variável para uma junta prismática.

Na convenção DH o único ângulo variável é θ . Assim, simplifica-se a notação escrevendo C_i para $\cos\theta_i$, etc. Também denota-se $\theta_1 + \theta_2$ por θ_{12} e $\cos(\theta_1 + \theta_2)$ por C_{12} .

2.2.2 Cinemática Inversa

O problema da cinemática inversa pode ser resolvido através do princípio do desacoplamento cinemático. Essa ferramenta é usada para simplificar o sistema de posicionamento e orientação da maioria dos manipuladores modernos. Usando o desacoplamento cinemático pode-se considerar os problemas de posição e orientação de forma independente. Uma outra forma de resolver o problema é a abordagem geométrica, que resolve o problema de posicionamento, enquanto explora-se a parametrização dos ângulos de Euler para resolver o problema de orientação (SPONG; VIDYASAGAR, 2008).

O problema geral da cinemática inversa é obtido da seguinte maneira para uma transformação homogênea 4×4 ,

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix} \in SE(3), \quad (2.8)$$

com $R \in SE(3)$, encontram-se uma ou todas soluções da equação

$$T_n^0(q_1, q_2, \dots, q_n) = H, \quad (2.9)$$

onde $T_n^0(q_1, q_2, \dots, q_n) = A_1(q_1) \times A_2(q_2) \times \dots \times A_n(q_n)$, H representa posição e orientação desejadas do *end-effector* e $SE(3)$ é um conjunto de todos os deslocamentos no espaço tridimensional, conhecido como grupo euclidiano em três dimensões. A tarefa é então encontrar os valores para as variáveis das juntas q_1, q_2, \dots, q_n .

As equações obtidas a partir de (2.9), são difíceis de se obter diretamente na forma fechada, que é o caso da maioria dos braços robóticos. Portanto, precisa-se desenvolver técnicas eficientes e sistemáticas que explorem a estrutura cinemática específica do manipulador. Enquanto o problema da cinemática direta sempre tem uma solução única que pode ser obtida pela avaliação das equações, o problema da cinemática inversa pode ou não ter uma solução. Isto é, mesmo que exista uma solução, ela pode ou não ser única (LI *et al.*, 2018), (TAKAHASHI

et al., 2017), (EL-SHERBINY *et al.*, 2018b) e (EL-SHERBINY *et al.*, 2018a). Na resolução do problema da cinemática inversa, necessita-se encontrar uma solução das equações de forma explícita:

$$q_k = f_k(h_{11}, h_{11}, \dots, h_{34}), k = 1, 2, \dots, n. \quad (2.10)$$

Assume-se que a posição e orientação dadas são tais que pelo menos uma solução da equação (2.9) existe. A partir desta, deve-se verificar se esta satisfaz todas as restrições nos possíveis intervalos de movimento ou não.

Também assume-se que a dada matriz homogênea H da equação (2.9) corresponda a uma configuração dentro do espaço de trabalho do manipulador com uma orientação atingível, de modo a garantir que as soluções obtidas correspondem a configurações alcançáveis.

2.2.2.1 Desacoplamento cinemático

Pelo fato da cinemática inversa ser de difícil solução, verifica-se que para manipuladores de seis articulações, com as três últimas articulações se interceptando em um ponto, é possível desacoplar o problema da cinemática inversa em dois problemas mais simples. Estes problemas são conhecidos respectivamente, como cinemática da posição inversa e cinemática da orientação inversa.

Supondo-se que exista exatamente 6-GDL de liberdade e que os últimos três eixos articulados se cruzam em um ponto o_c . Pode-se expressar (2.9) como dois conjuntos de equações que representam as equações de rotação e posição,

$$R_6^0(q_1, \dots, q_6) = R, \quad (2.11)$$

$$o_6^0(q_1, \dots, q_6) = o, \quad (2.12)$$

onde o e R são a posição e orientação desejadas do quadro do *end-effector*, expressas em relação ao sistema de coordenadas gerais. Assim, tem-se o e R , e o problema da cinemática inversa implica em resolver (2.11) e (2.12) para q_1, \dots, q_6 .

Considere agora os eixos de um manipulador 6-GDL como z_i , onde $i = 1, \dots, 6$ e suas origens como o_n , onde $n = 1, \dots, 5$ e a origem do *end-effector* como o_c . A suposição de um pulso esférico significa que os eixos z_3, z_4 e z_5 se cruzam em o_c e, portanto, as origens o_4 e o_5

atribuídas pela convenção DH estarão sempre no centro do pulso o_c . Às vezes, o_3 também estará no o_c , mas isso não é necessário para o nosso desenvolvimento subsequente do manipulador do projeto desenvolvido. O ponto importante desta suposição para a cinemática inversa, é que o movimento dos últimos três elos sobre esses eixos não mudará a posição de o_c . Desse modo, a posição do centro do punho é, portanto, uma função apenas das três primeiras variáveis conjuntas.

A origem do quadro de ferramentas (cujas coordenadas desejadas são dadas por o) é obtida por uma translação da distância d_6 ao longo de z_5 , a partir de o_c . No nosso caso, z_5 e z_6 são o mesmo eixo e a terceira coluna de R expressa a direção de z_6 em relação ao quadro de base. Portanto, tem-se:

$$o = o_c^0 + d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.13)$$

Assim, para ter o *end-effector* do robô no ponto com coordenadas dadas por o e com a sua orientação dada por $R = (r_{ij})$, é necessário e suficiente que o centro do punho, o_c tenha coordenadas dadas por:

$$o_c^0 = o - d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.14)$$

e que a orientação do *frame* $o_6x_6y_6z_6$ em relação à base seja dada por R .

Se os componentes da posição do *end-effector* o forem denotados por o_x, o_y, o_z e os componentes do centro do pulso o_c^0 são denotados de x_c, y_c, z_c , então (2.14) dá a seguinte relação:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{bmatrix}. \quad (2.15)$$

Usando a equação (2.15), encontra-se os valores das três primeiras variáveis das juntas e determina-se a transformação de orientação R_3^0 . Pode-se então determinar a orientação do *end-effector* em relação ao quadro $o_3x_3y_3z_3$ por

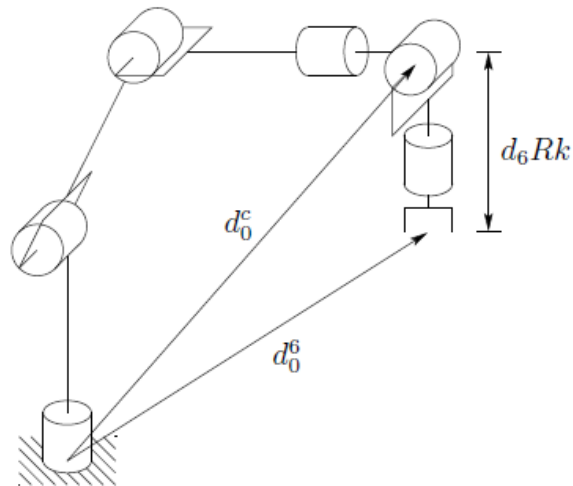
$$R = R_3^0 R_6^3, \quad (2.16)$$

onde

$$R_6^3 = (R_3^0)^{-1} R = (R_3^0)^T R. \quad (2.17)$$

Os três ângulos finais podem ser encontrados como um conjunto de ângulos de Euler correspondentes a R_6^3 . Observa-se que o lado direito da equação (2.17) é completamente conhecido, uma vez que R é dado e R_3^0 pode ser calculado, já que as três primeiras variáveis das juntas são conhecidas. A ideia do desacoplamento cinemático é ilustrada na Figura 4.

Figura 4 – Desacoplamento cinemático.



Fonte: Adaptada Spong, 2008.

2.2.2.2 Abordagem Geométrica

Pode-se utilizar uma abordagem geométrica para encontrar as variáveis q_1, q_2, q_3 , onde o_c^0 é dada por (2.14). Restringe-se o tratamento à abordagem geométrica por dois motivos. Primeiro, como já foi dito, os modelos de manipulador atuais são cinematicamente simples e, em geral, consistem de uma das cinco configurações básicas com um pulso esférico. Em segundo lugar, existem poucas técnicas que podem lidar com o problema da cinemática inversa geral para configurações arbitrárias. De fato, é em parte devido à dificuldade do problema da cinemática inversa geral que os projetos de manipulador evoluíram ao seu estado atual.

Em geral, a complexidade do problema da cinemática inversa aumenta com o número de parâmetros de ligação não nulos. Para a maioria dos manipuladores, muitos dos parâmetros a_i, d_i podem ser iguais a zero, e os parâmetros α_i podem ser iguais a zero ou $\pm\pi/2$, etc. Desse modo, para esses casos, uma abordagem geométrica é a mais adequada e natural.

2.3 Jacobiano de Manipuladores

O jacobiano fornece as relações entre as velocidades lineares e angulares do *end-effector* (ou qualquer outro ponto no manipulador) com as velocidades das juntas. Em particular, deriva-se a velocidade angular da estrutura do *end-effector* (que fornece a taxa de rotação da estrutura) e a velocidade linear da origem. Essas velocidades estão relacionadas com as velocidades das juntas, q_i . Matematicamente, as equações cinemáticas avançadas definem uma função entre o espaço das posições e orientações cartesianas e o espaço das posições das articulações. As relações de velocidade são então determinadas pelo jacobiano dessa função.

Esta matriz jacobiana é uma grandeza importante na análise e controle do movimento do robô, sendo usada no planejamento e execução de trajetórias suaves, na determinação de configurações singulares, na execução de movimento antropomórfico coordenado, na derivação das equações dinâmicas de movimento e na transformação de forças e torques do *end-effector* para as articulações do manipulador.

Com o jacobiano pode-se encontrar as configurações singulares nas quais o manipulador perde um ou mais graus de liberdade, as quais são determinadas geometricamente. Pode-se obter também os problemas inversos da determinação das velocidades e acelerações da junta para velocidades e acelerações de atuadores finais especificados (SPONG; VIDYASAGAR, 2008).

2.3.1 Derivação do Jacobiano

Considere um manipulador de n links com variáveis das juntas q_1, \dots, q_n ,

$$T_n^0(q) = \begin{bmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{bmatrix}, \quad (2.18)$$

que é a transformação do quadro do *end-effector* para o quadro base, onde $q = (q_1, \dots, q_n)^T$ é o vetor de variáveis das juntas. À medida que o robô se move, tanto as variáveis das juntas q_i como a posição do *end-effector* o_n^0 e a orientação R_n^0 serão funções do tempo. O objetivo agora é relacionar a velocidade linear e angular do *end-effector* ao vetor de velocidades de junção $\dot{q}(t)$.

Desta forma, tem-se:

$$S(\omega_n^0) = \dot{R}_n^0 (R_n^0)^T, \quad (2.19)$$

o vetor velocidade angular ω_n^0 , do *end-effector* é definido como

$$v_n^0 = \dot{o}_n^0, \quad (2.20)$$

é a velocidade linear do *end-effector*. Obteve-se expressões da forma:

$$v_n^0 = Jv\dot{q}, \quad (2.21)$$

$$\omega_n^0 = J\omega\dot{q}, \quad (2.22)$$

onde Jv e $J\omega$ resultam em matrizes $3 \times n$. Pode-se escrever (2.21) e (2.22) juntos como,

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = J_n^0 \dot{q}, \quad (2.23)$$

onde J_n^0 é dado por:

$$J_n^0 = \begin{bmatrix} Jv \\ J\omega \end{bmatrix}. \quad (2.24)$$

A matriz J_n^0 é chamada de jacobiana do manipulador ou jacobiana. Note que J_n^0 é uma matriz $6 \times n$ onde n é o número de *links*. Em seguida, deduz-se expressões simples para o jacobiana de qualquer manipulador (SPONG; VIDYASAGAR, 2008).

2.3.2 Velocidade angular

A velocidade angular contribuída por cada junta na orientação do referencial é obtida como,

$$\omega_i^{i-1} = \dot{q}_i z_{i-1}^{i-1} = \dot{q}_i k, \quad (2.25)$$

em que, ω_i^{i-1} que representa a velocidade angular da junta i que é transmitida pela rotação da junta i , expressa em relação ao quadro $o_{i-1}x_{i-1}y_{i-1}z_{i-1}$ e k é o vetor de coordenadas unitárias $(0, 0, 1)^T$.

Se a i -ésima articulação é prismática, então o movimento do quadro i em relação ao quadro $i - 1$ é uma translação e,

$$\omega_i^{i-1} = 0. \quad (2.26)$$

Assim, se a junta i é prismática, a velocidade angular do *end-effector* não depende de q_i , que agora é d_i .

Portanto, a velocidade angular total do *end-effector*, ω_n^0 , na estrutura da base é determinado por

$$\begin{aligned}\omega_n^0 &= \rho_1 \dot{q}_1 k + \rho_2 \dot{q}_2 R_1^0 k + \dots + \rho_n \dot{q}_n R_{n-1}^0 k \\ &= \sum_{i=1}^n \rho_i \dot{q}_i z_{i-1}^0\end{aligned}\quad (2.27)$$

em que ρ_i é igual a 1 se a articulação i é de revolução e 0 se a articulação i é prismática, uma vez que

$$z_{i-1}^0 = R_{i-1}^0 k. \quad (2.28)$$

Observa-se que $z_0^0 = k = (0, 0, 1)^T$.

A metade inferior do Jacobiano $J\omega$, em (2.24) é assim dada como,

$$J\omega = [\rho_1 z_0 \dots \rho_n z_{n-1}]. \quad (2.29)$$

Da equação 2.28, omitiu-se os sobrescritos para os vetores unitários ao longo dos eixos z , pois todos são referenciados ao quadro inercial.

2.3.3 Velocidade linear

A velocidade linear do *end-effector* é de apenas \dot{o}_n^0 . Pela regra da cadeia para diferenciação

$$\dot{o}_n^0 = \sum_{i=1}^n \frac{\partial o_n^0}{\partial q_i} \dot{q}_i. \quad (2.30)$$

Assim, a i -ésima coluna de J_v , foi chamada de J_{v_i} , é dada por

$$J_{v_i} = \frac{\partial o_n^0}{\partial q_i}. \quad (2.31)$$

A equação (2.31) é apenas a velocidade linear do *end-effector* que resultaria se \dot{q}_i fosse igual a um e o outro \dot{q}_j fosse zero.

2.3.4 Combinando as jacobianas angular e linear

A partir do desenvolvimento em (SPONG; VIDYASAGAR, 2008) e colocando-se a metade superior e inferior da matriz jacobiana juntos, obtém-se que a jacobiana para um manipulador de n links é da forma,

$$J = [J_1 J_2 \dots J_n], \quad (2.32)$$

onde a i -ésima coluna J_i é dada por

$$J_i = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}, \quad (2.33)$$

se a junta i é de revolução e

$$J_i = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix}, \quad (2.34)$$

se a junta i é prismática.

As equações (2.33) e (2.34) determinam a matriz jacobiana de qualquer manipulador, já que todas as quantidades necessárias estão disponíveis uma vez que a cinemática direta é calculada.

O procedimento acima funciona não apenas para calcular a velocidade do *end-effector*, mas também para calcular a velocidade de qualquer ponto no manipulador. Isto é importante quando necessita-se calcular a velocidade do centro de massa dos vários elos para derivar as equações dinâmicas do movimento.

2.4 Dinâmica de manipuladores

A dinâmica estuda o movimento de um sistema levando-se em consideração as forças que produzem o movimento. A dinâmica de um corpo rígido de n elos é descrita pela formulação Lagrangiana (MITTAL; NAGRATH, 2003). No enfoque Lagrangiano as variáveis das juntas q são um conjunto adequado de coordenadas generalizadas.

Para o propósito de projetar um controlador é necessário ter um modelo matemático que revele as propriedades dinâmicas do sistema. Desta forma, deduz-se as equações dinâmicas do movimento para o manipulador. Em seguida, calculam-se as equações da energia cinética e da potencial do manipulador e aplica-se a equação de Lagrange para o movimento (SANZ, 2009).

2.4.1 Equações de Euler-Lagrange

As equações de movimento de Euler-Lagrange são encontradas a partir de um conjunto geral de equações diferenciais que descrevem a evolução temporal dos sistemas mecânicos sujeitos a restrições holonômicas quando as forças de restrição satisfazem o princípio do trabalho virtual (SPONG; VIDYASAGAR, 2008).

Pela segunda lei de Newton, e considerando-se uma força f aplicada obtém-se a equação, (2.35):

$$\frac{d}{dt}(m\dot{y}) = \frac{d}{dt} \frac{\partial}{\partial \dot{y}} \left(\frac{1}{2} m \dot{y}^2 \right) = \frac{d}{dt} \frac{\partial K}{\partial \dot{y}}, \quad (2.35)$$

em que $K = \frac{1}{2} m \dot{y}^2$ é a energia cinética.

Utiliza-se a notação de derivada parcial na expressão acima para ser consistente com sistemas considerados mais tarde quando a energia cinética será uma função de diversas variáveis. Da mesma forma, pode-se expressar a força gravitacional como

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial P}{\partial y}, \quad (2.36)$$

em que $P = mgy$ é a energia potencial devido a gravidade.

Definindo-se

$$L = K - P = \frac{1}{2} m \dot{y}^2 - mgy, \quad (2.37)$$

nota-se que

$$\frac{\partial L}{\partial \dot{y}} = \frac{\partial K}{\partial \dot{y}} \text{ e } \frac{\partial L}{\partial y} = -\frac{\partial P}{\partial y}. \quad (2.38)$$

Desse modo, a equação pode ser expressa em:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f. \quad (2.39)$$

A função L , que é a diferença da energia cinética e potencial é chamada de **Lagrangiana** do sistema e a equação (2.39) é chamada de **Equação de Euler-Lagrange**.

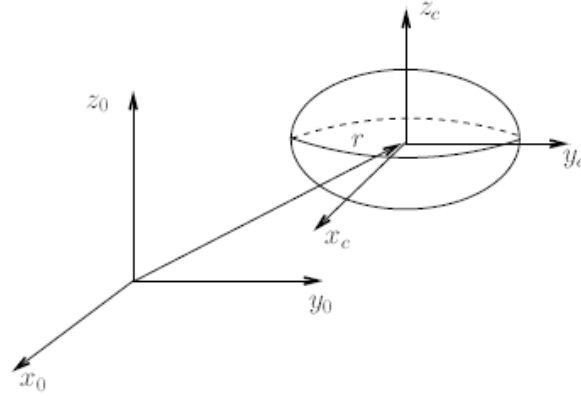
2.4.2 Expressões gerais para energia cinética e potencial

As equações de Euler-Lagrange podem ser usadas para derivar as equações dinâmicas de maneira direta, desde que seja possível expressar a energia cinética e potencial do sistema em termos de um conjunto de coordenadas generalizadas. Para que este resultado seja útil em um contexto prático, necessita-se computar esses termos para um manipulador robótico de n links. Para isso, é preciso deduzir-se fórmulas para a energia cinética e a energia potencial de um corpo rígido usando as variáveis das juntas encontradas através da conversão DH (SICILIANO *et al.*, 2010).

A energia cinética de um corpo rígido é a soma de dois termos: a energia translacional obtida pela concentração de toda a massa do objeto no centro de massa e a energia cinética

rotacional do corpo sobre o centro de massa. Referindo-se à Figura 5, anexa-se um quadro de coordenadas no centro de massa (chamado de quadro anexado ao corpo), como apresentado em (SPONG; VIDYASAGAR, 2008).

Figura 5 – Corpo rígido geral.



Fonte: SPONG e VIDYASAGAR (2008).

A energia cinética do corpo rígido é então dada como,

$$K = \frac{1}{2}mv^T v + \frac{1}{2}\omega^T I \omega, \quad (2.40)$$

em que m é a massa total do objeto, v e ω são os vetores de velocidade linear e angular, respectivamente, e I é uma matriz simétrica 3×3 chamada de Tensor de Inércia.

2.4.2.1 Energia cinética para um robô de n elos

Considere um manipulador que consiste de $i = 1, 2, \dots, n$ elos. Verifica-se que as velocidades lineares e angulares de qualquer ponto em qualquer elo podem ser expressas em termos da matriz Jacobiana e a derivada das variáveis das juntas. Como no nosso caso as variáveis das juntas são de fato as coordenadas generalizadas, portanto, para as matrizes Jacobianas apropriadas J_{v_i} e J_{ω_i} , tem-se que:

$$v_i = J_{v_i}(q)\dot{q}, \quad \omega_i = J_{\omega_i}(q)\dot{q}. \quad (2.41)$$

Seja m_i a massa de um elo i e I_i a matriz de inércia do elo i em torno de um quadro de coordenadas paralelo ao quadro i e cuja origem está no centro de massa. Então, de (2.40), segue-se que a energia cinética geral do manipulador é igual a

$$K = \frac{1}{2}\dot{q}^T \sum_{i=1}^n [m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q^T) J_{\omega_i}(q)] \dot{q}, \quad (2.42)$$

ou seja, a energia cinética do manipulador é da forma

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q}, \quad (2.43)$$

onde $D(q)$ é a matriz de inércia de dimensão $n \times n$ e é definida positiva simétrica para cada $q \in R^n$.

2.4.2.2 Energia potencial para um robô de n elos

Considerando-se agora o termo de energia potencial, no caso da dinâmica do corpo rígido, a única fonte de energia potencial é a gravidade. A energia potencial do i -ésimo elo pode ser calculada assumindo que a massa de todo o objeto está concentrada em seu centro de massa e é dada por:

$$P_i = g^T r_{ci} m_i, \quad (2.44)$$

onde g é o vetor dado na direção da gravidade no referencial inercial e o vetor r_{ci} são as coordenadas do centro de massa do elo i . A energia potencial total do robô com n elos é, portanto,

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n g^T r_{ci} m_i. \quad (2.45)$$

No caso de o robô possuir elasticidade, por exemplo, no caso de possuir juntas flexíveis, a energia potencial incluirá termos contendo a energia armazenada nos elementos elásticos. Observa-se que a energia potencial é uma função das coordenadas generalizadas e não de suas derivadas, ou seja, a energia potencial depende da configuração do robô, mas não da sua velocidade.

2.4.3 Equações de Movimento

Considerando-se as equações de Euler-Lagrange para o caso especial cujas condições são válidas: (i) a energia cinética é uma função quadrática do vetor \dot{q} da forma

$$K = \frac{1}{2} \sum_{ij} d_{ij}(q) \dot{q}_i \dot{q}_j := \frac{1}{2} \dot{q}^T D(q) \dot{q}, \quad (2.46)$$

e (ii) a energia potencial $P = P(q)$ é independente de \dot{q} . E portanto, os manipuladores robóticos, satisfazem essa condição.

A equação de Lagrange de um sistema mecânico é dada por,

$$L = K - P = \frac{1}{2} \sum_{ij} d_{ij}(q) \dot{q}_i \dot{q}_j - P(q), \quad (2.47)$$

cujo desenvolvimento para a obtenção das equações de Euler-Lagrange do sistema é amplamente descrito na literatura, como em (SPONG; VIDYASAGAR, 2008).

Portanto, a partir de (2.47), as equações de Euler-Lagrange podem ser obtidas como:

$$\sum_i d_{kj}(q) \ddot{q}_j + \sum_{ij} c_{ijk}(q) \dot{q}_i \dot{q}_j + \phi_k(q) = \tau_k, \quad k = 1, \dots, n, \quad (2.48)$$

onde $d_{kj}(q)$ são termos que dependem da configuração do manipulador, $\phi_k = \partial P / \partial q_k$, τ_k é o torque que atua sobre cada junta e os termos

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\}, \quad (2.49)$$

são conhecidos como símbolos Christoffel (do primeiro tipo). Note que para um k fixo, tem-se $c_{ijk} = c_{jik}$, o que reduz o esforço envolvido na computação desses símbolos por um fator de cerca da metade.

A equação (2.48) pode ser escrita na forma matricial como,

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau, \quad (2.50)$$

em que o k, j -ésimo elemento da matriz $C(q, \dot{q})$ é definido como:

$$\begin{aligned} c_{kj} &= \sum_{i=1}^n c_{ijk}(q) \dot{q}_i \\ &= \sum_{i=1}^n \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i. \end{aligned} \quad (2.51)$$

A partir do desenvolvimento das equações desta seção (2.4.3) pode-se verificar que em geral estas se aplicam a qualquer sistema mecânico cuja energia cinética esteja na forma da equação (2.46) e cuja energia potencial seja independente de \dot{q} .

2.5 Dinâmica Inversa

O cálculo da dinâmica inversa de um robô manipulador é essencial para o seu sistema de controle e são comumente usadas para formular os modelos dinâmicos que são obtidos pelos métodos de *Newton-Euler* e *Lagrange-Euler*. Estes são métodos recursivos numéricos

computacionalmente caros e não são adequados para serem usados diretamente em aplicações em tempo real (RAIN *et al.*, 2019).

Portanto, a abordagem do controle da dinâmica inversa faz uso da relação entre os torques das juntas, como as forças de contato das entradas e do atuador final e as variáveis de posição do *end-effector* ao longo das superfícies de restrição (KORKMAZ; IDER, 2014).

A ideia da dinâmica inversa é buscar uma lei de controle de realimentação não-linear do tipo

$$u = f(q, \dot{q}, t), \quad (2.52)$$

resultando em um sistema linear de malha fechada.

Considere novamente as equações dinâmicas de um robô de n elos em forma de matriz (SPONG; VIDYASAGAR, 2008)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (2.53)$$

onde $M(q) = D(q) + J$, onde J é uma matriz diagonal com elementos diagonais $r_k^2 J_{m_k}$ e o vetor de entrada u tem componentes

$$u_k = r_k \frac{K_{m_k}}{R_k} V_k. \quad (2.54)$$

No caso da equação dinâmica do manipulador (2.54), esta é obtida escolhendo-se o controle u de acordo com a equação

$$u = M(q)a_q + C(q, \dot{q})\dot{q} + g(q), \quad (2.55)$$

então, como a matriz de inércia M é invertível, o sistema combinado de (2.53) e (2.55) se reduz para

$$\ddot{q} = a_q, \quad (2.56)$$

onde a_q representa uma nova entrada para o sistema que ainda está para ser escolhido. A equação (2.56) é conhecida como sistema de integração dupla, pois representa dois integradores duplos desacoplados. A lei de controle não linear (2.55) é chamada de controle de dinâmica inversa em que tem-se um resultado linear e desacoplado (SPONG; VIDYASAGAR, 2008).

De acordo com a_{qk} pode-se projetá-lo para controlar um sistema linear de segunda ordem escolhendo-se:

$$a_q = -K_0 q - K_1 \dot{q} + r, \quad (2.57)$$

onde K_0 e K_1 são matrizes diagonais com elementos diagonais consistindo de ganhos de posição e velocidade, respectivamente. O sistema de malha fechada é então o sistema linear

$$\dot{q} + K_1\dot{q} + K_0q = r. \quad (2.58)$$

Considerando-se uma trajetória desejada

$$t \rightarrow (q^d(t), \dot{q}^d(t)), \quad (2.59)$$

pode-se escolher a entrada de referência $r(t)$ como

$$r(t) = \ddot{q}^d(t) + K_0q^d(t) + K_1\dot{q}^d(t), \quad (2.60)$$

então o erro de rastreamento $e(t) = q - q^d$ satisfaz

$$\ddot{e}(t) + K_1\dot{e}(t) + K_0e(t) = 0. \quad (2.61)$$

Deste modo, uma escolha para as matrizes de ganho K_0 e K_1 é

$$K_0 = \text{diag}\{\omega_1^2, \dots, \omega_n^2\}, \quad (2.62)$$

$$K_1 = \text{diag}\{2\omega_1, \dots, 2\omega_n\} \quad (2.63)$$

que resulta em um sistema de malha fechada que é dissociado globalmente, com cada resposta da junta igual à resposta de um sistema linear de segunda ordem criticamente amortecido com frequência natural ω_i , a qual determina a velocidade de resposta da articulação ou, equivalentemente, a taxa de decaimento do erro de rastreamento.

A abordagem de dinâmica inversa é importante para o controle de robôs manipuladores podendo-se analisar de forma diferente a partir da lei de controle (2.55). Ou seja, das equações dinâmicas do manipulador (2.53). Assim, como $M(q)$ é invertível para $q \in \mathbb{R}^n$ pode-se resolver a aceleração \ddot{q} do manipulador como,

$$\ddot{q} = M^{-1}\{u - C(q, \dot{q})\dot{q} - g(q)\}. \quad (2.64)$$

Especificando-se a aceleração como a entrada para o sistema e supondo-se que se tenha atuadores capazes de produzir diretamente uma aceleração comandada (em vez de

indiretamente, produzir uma força ou torque). Então a dinâmica do manipulador, que é afinal um dispositivo de controle de posição,

$$\ddot{q} = a_q(t), \quad (2.65)$$

onde $a_q(t)$ é o vetor de aceleração de entrada. O problema de controle do sistema (2.65) torna-se simples de modo que a entrada de aceleração a_q pode ser escolhida como antes, de acordo com (2.57).

Comparando as equações (2.64) e (2.65) tem-se que o torque u e a aceleração a_q do manipulador são relacionados por:

$$M^{-1}\{u(t) - C(q, \dot{q})\dot{q} - g(q)\} = a_q. \quad (2.66)$$

Pela invertibilidade da matriz de inércia, tem-se $u(t)$ como

$$u = M(q)a_q + C(q, \dot{q})\dot{q} + g(q), \quad (2.67)$$

que é igual à expressão dada na equação (2.55). Assim, a dinâmica inversa pode ser vista como uma transformação de entrada que se transforma em entrada de aceleração tornando-se simples de implementar.

2.5.1 Dinâmica Inversa do Espaço de Tarefas

O rastreamento no espaço de tarefas pode ser encontrado modificando a escolha do controle da malha externa, \ddot{q} em (2.56), mantendo o controle da malha interna inalterada. Seja $X \in \mathbb{R}^3$ a posição do *end-effector*, como X é uma função das variáveis das juntas $q \in C$, temos:

$$\dot{X} = J(q)\dot{q} \quad (2.68)$$

$$\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q}. \quad (2.69)$$

onde $J = J_a$ é a jacobiana analítica.

Dado o sistema de integrador duplo, (2.56), no espaço das juntas, tem-se a_q que é escolhido como

$$a_q = J^{-1}\{a_X - \dot{J}\dot{q}\}, \quad (2.70)$$

onde o resultado é um sistema integrador duplo nas coordenadas do espaço de tarefas:

$$\dot{X} = a_X. \quad (2.71)$$

Dada uma trajetória no espaço de tarefa $X^d(t)$, satisfazendo as mesmas suposições de suavidade e delimitação que a trajetória de espaço das juntas $q^d(t)$, pode-se escolher a_X como,

$$a_X = \ddot{X}^d + K_P(X^d - X) + K_D(\dot{X}^d - \dot{X}), \quad (2.72)$$

para que o erro de rastreamento do espaço cartesiano, $\tilde{X} = X - X_d$, satisfaça

$$\ddot{\tilde{X}} + K_D\dot{\tilde{X}} + K_P\tilde{X} = 0. \quad (2.73)$$

Portanto, uma modificação do controle da malha externa atinge um sistema linear e desacoplado diretamente nas coordenadas do espaço de tarefas, sem a necessidade de calcular uma trajetória no espaço das juntas e sem a necessidade de modificar o controle da malha interna não linear.

Usando-se uma representação mínima para a orientação do *end-effector*, pode-se especificar uma trajetória $X \in \mathbb{R}^3$. Em geral, se as coordenadas do *end-effector* são dadas em função da jacobiana J , a formulação acima será a jacobiana geométrica $J(q)$. Nesse caso, as velocidades do manipulador são dadas por

$$V = \begin{pmatrix} \dot{x} \\ \omega \end{pmatrix} = J(q)\dot{q} \quad (2.74)$$

e o controle da malha externa é

$$a_q = J^{-1}(q) \left\{ \begin{pmatrix} a_x \\ a_\omega \end{pmatrix} - J(q)\dot{q} \right\}. \quad (2.75)$$

Aplicado a equação acima em (2.56) resulta no sistema

$$\ddot{x} = a_x \in \mathbb{R}^3 \quad (2.76)$$

$$\dot{\omega} = a_\omega \in \mathbb{R}^3 \quad (2.77)$$

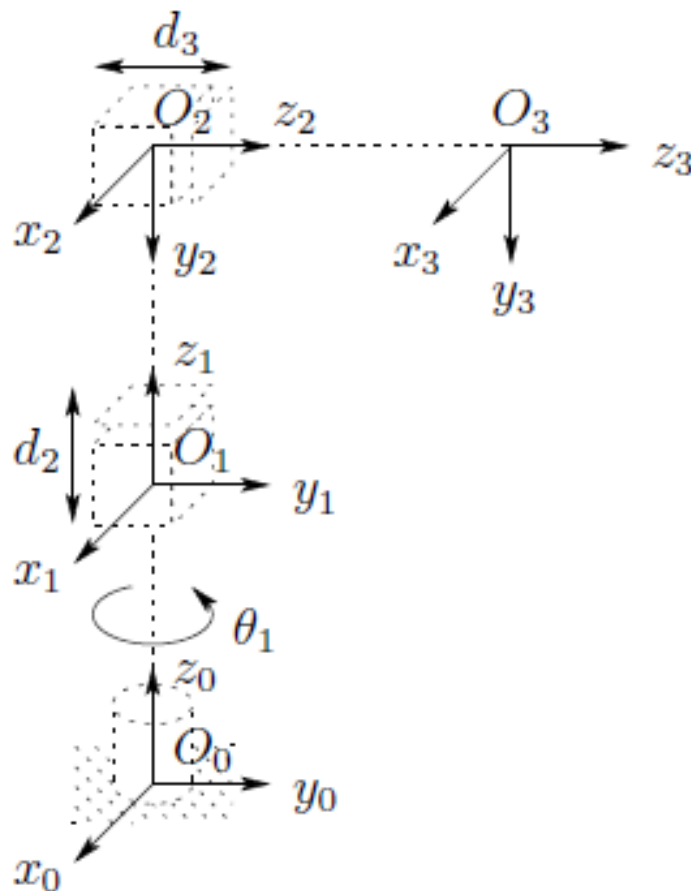
A partir dos dois casos, observa-se que a não singularidade da jacobiana é necessária para implementar o controle de malha externa. Se o robô possuir menos que seis articulações, os jacobianos não são quadrados. Nesse caso, a solução do problema pode ser obtida resolvendo-se a pseudo-inverso no lugar da inversa da jacobiana. A abordagem de controle de dinâmica inversa pode ser proposta em várias formas diferentes, como o controle de aceleração resolvido e o controle de espaço operacional (SPONG; VIDYASAGAR, 2008).

2.6 Manipulador Cilíndrico

Nesta seção, são aplicados os conceitos apresentados anteriormente a um manipulador cilíndrico, que será utilizado para validar o método de aproximação de modelos proposto.

Considere o sistema de referências geométricas de um robô cilíndrico representado na Figura 6. A junta 1 tem como origem o_0 . O posicionamento desta origem ao longo de z_0 , assim como a direção do eixo x_0 , são arbitrários. O eixo x_0 é escolhido como normal na página. Em seguida, como z_0 e z_1 coincidem, a origem o_1 é escolhida ao longo do elo 1. O eixo x_1 é normal para a página quando $\theta_1 = 0$, mas como θ_1 é variável, sua direção também é variável. Como z_2 e z_1 se cruzam, a origem o_2 é colocada nessa interseção. A direção de x_2 é escolhida paralela a x_1 , de modo que θ_2 é igual a zero. Logo, a terceira estrutura é escolhida no final do elo 3. Os parâmetros dos elos obtidos a partir das equações 2.6 e 2.7 são mostrados na Tabela 1.

Figura 6 – Sistema de referências geométricas de um manipulador cilíndrico.



Fonte: Adaptada Spong, 2008.

Desse modo, as matrizes A e T foram obtidas utilizando-se as equações (2.6) e (2.7)

Tabela 1 – Parâmetros de ligação do manipulador cilíndrico.

elo	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90°	d_2^*	0
3	0	0	d_3^*	0

Fonte: O autor (2022).

da convenção de D-H são dadas então por:

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.78)$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.79)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.80)$$

$$T_3^0 = A_1 A_2 A_3 = \begin{bmatrix} C_1 & 0 & -S_1 & -S_1 d_3 \\ S_1 & 0 & C_1 & C_1 d_3 \\ 0 & -1 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.81)$$

2.6.1 Jacobiana do Manipulador Cilíndrico

Da Figura 6 observa-se que para um manipulador cilíndrico tem as seguintes variáveis das juntas, $q = (\theta_1, d_2, d_3)$.

Como o manipulador cilíndrico possui uma junta de revolução e duas prismáticas, ou seja, três juntas, a matriz jacobiana resulta 6×3 , sendo da forma:

$$J(q) = \begin{bmatrix} z_0 \times (o_3 - o_0) & z_1 & z_2 \\ z_0 & o_0 & o_0 \end{bmatrix}, \quad (2.82)$$

onde $z_o = [0 \ 0 \ 1]^T = z_1$, $o_0 = [0 \ 0 \ 0]^T$,

$$z_2 = \begin{bmatrix} \text{sen}(\theta_1) \\ \text{cos}(\theta_1) \\ 0 \end{bmatrix}, \quad (2.83)$$

e

$$o_3 = \begin{bmatrix} -\text{cos}(\theta_1)(l_3 + l_3^0) \\ -\text{sen}(\theta_1)(l_3 + l_3^0) \\ 0 \end{bmatrix}, \quad (2.84)$$

onde l_3^0 é o deslocamento entre os eixos z_1 e z_3 ao longo do eixo x_2 .

Substituindo-se cada matriz, realizando as operações necessárias e considerando $d_3 = l_3 + l_3^0$, obtém-se a matriz jacobiana 6×3 :

$$J = \begin{bmatrix} -\text{cos}(\theta_1)d_3 & 0 & \text{sen}(\theta_1) \\ -\text{sen}(\theta_1)d_3 & 0 & \text{cos}(\theta_1) \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \quad (2.85)$$

A jacobiana também pode ser representada na forma

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -\text{cos}(\theta_1)d_3 & 0 & \text{sen}(\theta_1) \\ -\text{sen}(\theta_1)d_3 & 0 & \text{cos}(\theta_1) \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}. \quad (2.86)$$

A partir da equação (2.86), nota-se que não é possível executar uma rotação em torno dos eixos x_0 e y_0 . A jacobiana em relação à velocidade linear do *end-effector* é obtido considerando-se apenas as três primeiras linhas, ou seja,

$$J = \begin{bmatrix} -\text{cos}(\theta_1)(d_3 + l_3^0) & 0 & \text{sen}(\theta_1) \\ -\text{sen}(\theta_1)(d_3 + l_3^0) & 0 & \text{cos}(\theta_1) \\ 0 & 1 & 0 \end{bmatrix}. \quad (2.87)$$

2.6.2 Dinâmica do Manipulador Cilíndrico

A formulação baseada no lagrangiano do sistema mecânico é definida como

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q). \quad (2.88)$$

Em seguida, são calculadas a energia cinética e potencial para um manipulador cilíndrico e aplicadas à formulação baseada no lagrangiano.

2.6.2.1 Energia Cinética

A energia cinética total do manipulador com acionamento nas três juntas é dada por (SANZ, 2009):

$$K = K_1 + K_2 + K_3. \quad (2.89)$$

A jacobiana do manipulador foi apresentado na equação (2.86). A partir dele, pode-se determinar as equações da energia cinética em função das velocidades, como a seguir:

$$K_1 = \frac{1}{2} [m_1 (-\cos\theta_1 d_3 \dot{\theta}_1 + \sin\theta_1 \dot{d}_3)^2], \quad (2.90)$$

$$K_2 = \frac{1}{2} [m_2 (-\sin\theta_1 d_3 \dot{\theta}_1 + \cos\theta_1 \dot{d}_3)^2], \quad (2.91)$$

$$K_3 = \frac{1}{2} [m_3 \dot{d}_2^2 + \dot{\theta}_1^2 I_3], \quad (2.92)$$

onde I_3 é o momento de inércia do elo 3.

Após a realização de algumas operações matemáticas e transformações trigonométricas, obtém-se a equação que representa a energia cinética total

$$K = \frac{1}{2} [(m_1 + m_2) (-2\cos\theta_1 \sin\theta_1) (d_3 \dot{d}_3 \dot{\theta}_1) + (m_1 \cos^2\theta_1 + m_2 \sin^2\theta_1) (d_3^2 \dot{\theta}_1^2) + (m_1 \sin^2\theta_1 + m_2 \cos^2\theta_1) (\dot{d}_3^2) + m_3 \dot{d}_2^2 + \dot{\theta}_1^2 I_3]. \quad (2.93)$$

2.6.2.2 Energia potencial

Das definições da mecânica clássica de ponto de referência (zero de energia potencial), a energia potencial para cada junta do manipulador é (SCIAVICCO; SICILIANO, 1996):

$$P_1 = m_1 g l_1 \sin \theta_1 = 0 \quad (2.94)$$

pois $l_1 = a_1 = 0$,

$$P_2 = m_2 g d_2, \quad (2.95)$$

e

$$P_3 = m_3 g d_2. \quad (2.96)$$

Como $P = P_1 + P_2 + P_3$, tem-se:

$$P = g d_2 (m_2 + m_3). \quad (2.97)$$

2.6.2.3 Equação de Lagrange

As equações de movimento do sistema são dadas por:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = \tau \quad (2.98)$$

onde $\tau \in \mathbb{R}^n$ são os torques aplicados às juntas. Assim, considerando-se a energia cinética do manipulador, a equação dinâmica do manipulador pode ser escrita de forma simplificada como:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (2.99)$$

onde $M(q)$ é a matriz de massa de dimensão $n \times n$, $C \in \mathbb{R}^n$ é a matriz que descreve as forças centrípetas e de Coriolis e $G = \frac{\partial g}{\partial q} \in \mathbb{R}^n$ é o vetor de gravidade.

Os efeitos de atrito nas juntas e de forças externas no *end-effector* podem ser incluídos no modelo dinâmico do manipulador da seguinte forma:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(q)\dot{q} + G(q) = \tau - f_{ext}, \quad (2.100)$$

onde f_{ext} é a força externa aplicada no *end-effector* e $F(q) \in \mathbb{R}^n$ representa os efeitos das forças de atrito dinâmico e estático nas juntas. Esse vetor também representa os distúrbios e dinâmicas não modeladas como folgas nos acoplamentos e transmissões mecânicas.

Aplicando-se a formulação de Lagrange, equação (2.88), o lagrangiano para o sistema será:

$$L = \frac{1}{2} [(m_1 + m_2)(-2 \cos \theta_1 \sin \theta_1)(d_3 \dot{d}_3 \dot{\theta}_1) + (m_1 \cos^2 \theta_1 + m_2 \sin^2 \theta_1)(d_3^2 \dot{\theta}_1^2) + (m_1 \sin^2 \theta_1 + m_2 \cos^2 \theta_1)(d_3^2) + m_3 d_2^2 + \dot{\theta}_1^2 I_3] - g d_2 (m_2 + m_3). \quad (2.101)$$

As equações do movimento do manipulador a partir da formulação de Lagrange são obtida pelas derivadas parciais do lagrangiano (2.101). Estas equações representam o torque em cada junta:

$$\begin{aligned}\tau_1 = & -[(4m_1 \text{sen}\theta_1 - 4m_2 \text{cos}\theta_1)d_3 + I_3]\ddot{\theta}_1 \\ & + [(m_1 + m_2)(\text{sen}\theta_1 \text{cos}\theta_1)d_3]\ddot{d}_3 \\ & + [(m_1 \text{sen}\theta_1 - m_2 \text{cos}\theta_1)d_3]\dot{\theta}_1^2 \\ & - [m_1 \text{cos}\theta_1 + m_2 \text{sen}\theta_1]d_3^2 \\ & - [(m_1 + m_2)(\text{sen}\theta_1 \text{cos}\theta_1)d_3]\dot{\theta}_1 \dot{d}_3,\end{aligned}\tag{2.102}$$

$$\tau_2 = m_3 \ddot{d}_2 + g(m_2 + m_3),\tag{2.103}$$

$$\begin{aligned}\tau_3 = & [m_1 \text{sen}\theta_1 \text{cos}\theta_1]\ddot{\theta}_1 \\ & - [2(m_1 \text{sen}\theta_1 + m_2 \text{cos}\theta_1)]\ddot{d}_3 \\ & + [2d_3(m_1 \text{sen}\theta_1 - m_2 \text{cos}\theta_1)]\dot{\theta}_1^2 \\ & - [(m_1 + m_2)(\text{sen}\theta_1 \text{cos}\theta_1)]\dot{\theta}_1 \dot{d}_3.\end{aligned}\tag{2.104}$$

Para o manipulador em estudo as equações de torque de cada junta, (2.102), (2.103) e (2.104), foram linearizadas em torno de um ponto de equilíbrio dado por, $\theta_1 = 0^\circ$, que podem ser expressas como:

$$\tau_1 = (4m_2 d_3 + I_3)\ddot{\theta}_1 - m_2 d_3 \dot{\theta}_1 - m_1 \dot{d}_3,\tag{2.105}$$

$$\tau_2 = m_3 \ddot{d}_2 + g(m_2 + m_3),\tag{2.106}$$

$$\tau_3 = -2m_2 \ddot{d}_3 - 2m_2 \dot{\theta}_1.\tag{2.107}$$

Nesse caso, as equações (2.105), (2.106) e (2.107) correspondem a um conjunto de equações diferenciais autônomas lineares.

2.6.2.4 Dinâmica do manipulador cilíndrico na forma matricial

Escrevendo as equações (2.102), (2.103) e (2.104) na forma padrão matricial como apresentado em (2.99) tem-se:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} &= \begin{bmatrix} (4m_1 \text{sen} \theta_1 - 4m_2 \text{cos} \theta_1) d_3 + I_3 & 0 & (m_1 + m_2) (\text{sen} \theta_1 \text{cos} \theta_1) d_3 \\ 0 & m_3 & 0 \\ m_1 \text{sen} \theta_1 \text{cos} \theta_1 & 0 & 2(m_1 \text{sen} \theta_1 + m_2 \text{cos} \theta_1) \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} \\
 &+ \begin{bmatrix} (m_1 \text{sen} \theta_1 - m_2 \text{cos} \theta_1) d_3 & 0 & -m_1 \text{cos} \theta_1 + m_2 \text{sen} \theta_1 \\ 0 & 0 & 0 \\ 2d_3 (m_1 \text{sen} \theta_1 - m_2 \text{cos} \theta_1) & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{d}_2^2 \\ \dot{d}_3^2 \end{bmatrix} \\
 &+ \begin{bmatrix} 0 & -(m_1 + m_2) (\text{sen} \theta_1 \text{cos} \theta_1) d_3 & 0 \\ 0 & 0 & 0 \\ 0 & -(m_1 + m_2) (\text{sen} \theta_1 \text{cos} \theta_1) & 0 \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_1 \dot{d}_2 \\ \dot{\theta}_1 \dot{d}_3 \\ \dot{d}_2 \dot{d}_3 \end{bmatrix} \\
 &+ \begin{bmatrix} 0 \\ g(m_2 + m_3) \\ 0 \end{bmatrix}.
 \end{aligned} \tag{2.108}$$

onde $\ddot{\theta}_1$, \ddot{d}_2 , \ddot{d}_3 são as acelerações das juntas, $\dot{\theta}_1^2$, \dot{d}_2^2 , \dot{d}_3^2 são acelerações centrípetas e $\dot{\theta}_1 \dot{d}_2$, $\dot{\theta}_1 \dot{d}_3$, $\dot{d}_2 \dot{d}_3$ são as acelerações de Coriolis.

O modelo dinâmico na forma matricial também pode ser linearizado em torno de um ponto, neste caso $\theta_1 = 0^\circ$. Portanto, temos:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} &= \begin{bmatrix} 4m_2 d_3 + I_3 & 0 & 0 \\ 0 & m_3 & 0 \\ 0 & 0 & -2m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} -m_2 d_3 \dot{\theta}_1 & 0 & -m_1 \dot{d}_3 \\ 0 & 0 & 0 \\ -2d_3 m_2 \dot{\theta}_1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ g(m_2 + m_3) \\ 0 \end{bmatrix}.
 \end{aligned} \tag{2.109}$$

2.6.2.5 Equações dinâmicas na forma de espaço de estados

As equações do modelo dinâmico do manipulador podem ser descritas em termos de espaço de estados (KELLY *et al.*, 2006), que facilita a aplicação do controlador ao modelo dinâmico do manipulador.

Em termos dos vetores de estados $[\theta_1 \ d_2 \ d_3 \ \dot{\theta}_1 \ \dot{d}_2 \ \dot{d}_3]$, as equações (2.105), (2.106) e

(2.107) podem ser expressas como,

$$\frac{d}{dt} \begin{bmatrix} \theta_1 \\ d_2 \\ d_3 \\ \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{d}_3 \\ \frac{1}{4m_1 d_3 I_3} [\tau_1 - (m_1 d_3 \dot{\theta}_1^2 + m_2 \dot{d}_3^2)] \\ \frac{1}{m_3} [\tau_2 - (m_2 + m_3)g] \\ \frac{1}{2m_1} [\tau_3 - (2d_3 m_1 \dot{\theta}_1^2)] \end{bmatrix}. \quad (2.110)$$

A condição necessária e suficiente para a existência de equilíbrios é dada quando tem-se $\tau_1 = m_1 d_3 \dot{\theta}_1^2 + m_2 \dot{d}_3^2$, $\tau_2 = (m_2 + m_3)g$ e $\tau_3 = 2d_3 m_1 \dot{\theta}_1^2$. Então tem-se,

$$[\theta_1 \ d_2 \ d_3 \ \dot{\theta}_1 \ \dot{d}_2 \ \dot{d}_3]^T = [\theta_1^* \ d_2^* \ d_3^* \ 0 \ 0 \ 0]^T, \quad (2.111)$$

com θ_1^* , d_2^* , $d_3^* \in \mathbb{R}$.

2.6.3 Manipulador cilíndrico do GPAR

Considerando-se as equações (2.6) e (2.7) apresentadas na Seção 2.2.1.1 pode-se obter as equações para o manipulador utilizado neste trabalho cuja geometria é do tipo cilíndrica e acionado por motores de indução trifásico. Para este tipo de manipulador encontrou-se os parâmetros D-H de acordo com as equações (2.78), (2.79), (2.80) e (2.81) apresentados na Tabela 2.

Tabela 2 – Parâmetros D-H do manipulador do GPAR.

elo	a_i	α_i	d_i	θ_i
1	0	0	0,245	θ_1^*
2	0,11	-90°	d_2^*	0
3	0	0	d_3^*	0

Fonte: O autor (2022).

Aplicando-se a convenção D-H a partir dos parâmetros apresentados na Tabela 2, encontram-se as matrizes A e T correspondentes, ou seja:

$$A_1 = \begin{bmatrix} C_1 & -S_2 & 0 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0,245 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.112)$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.113)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.114)$$

$$T_3^0 = A_1 A_2 A_3 = \begin{bmatrix} C_1 & 0 & -S_1 & -S_1(d_3 + 0,35) \\ S_1 & 0 & C_1 & C_1(d_3 + 0,35) \\ 0 & -1 & 0 & 0,245 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.115)$$

Assim, qualquer posição do elemento final do robô pode ser encontrada na sua área de trabalho a partir das coordenadas no espaço das juntas, como mostrado a seguir:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} -\sin(\theta_1)(d_3 + 0,35) \\ \cos(\theta_1)(d_3 + 0,35) \\ 0,245 + d_2 \end{bmatrix} \quad (2.116)$$

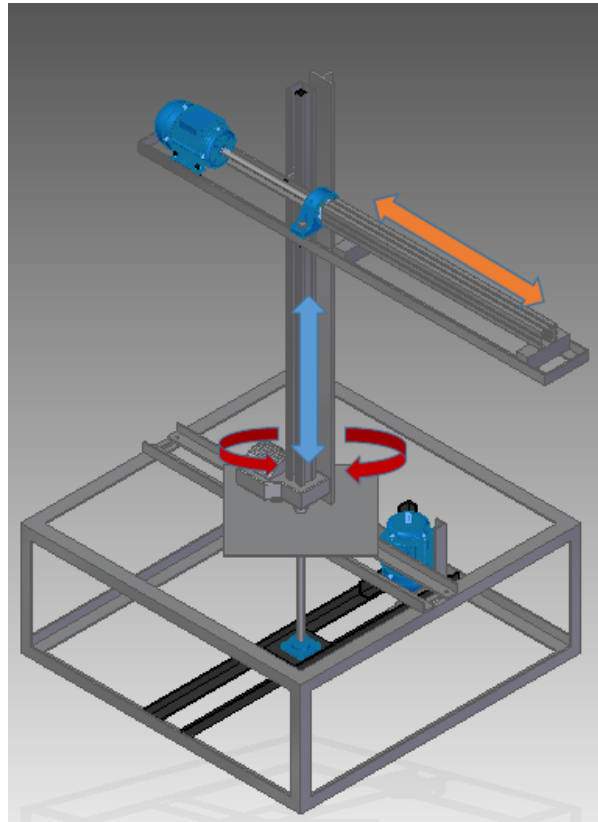
Para calcular os torques de cada junta foi necessário encontrar as massas das juntas do manipulador, que foram obtidas através do *software* de modelagem *Solid Edge*®. A Figura 7 apresenta a modelagem computacional do manipulador.

Através do *software* de modelagem computacional, foram encontradas as principais propriedades físicas do manipulador, como as dimensões, massas e momentos de inércia de cada elo. As Figuras 8 e 9 apresentam a tela do *software* de modelagem com as propriedades físicas do manipulador para as junta 2 e 3, respectivamente.

Os valores das massas (m) e comprimentos (l) de cada elo do manipulador são mostrados na Tabela 3.

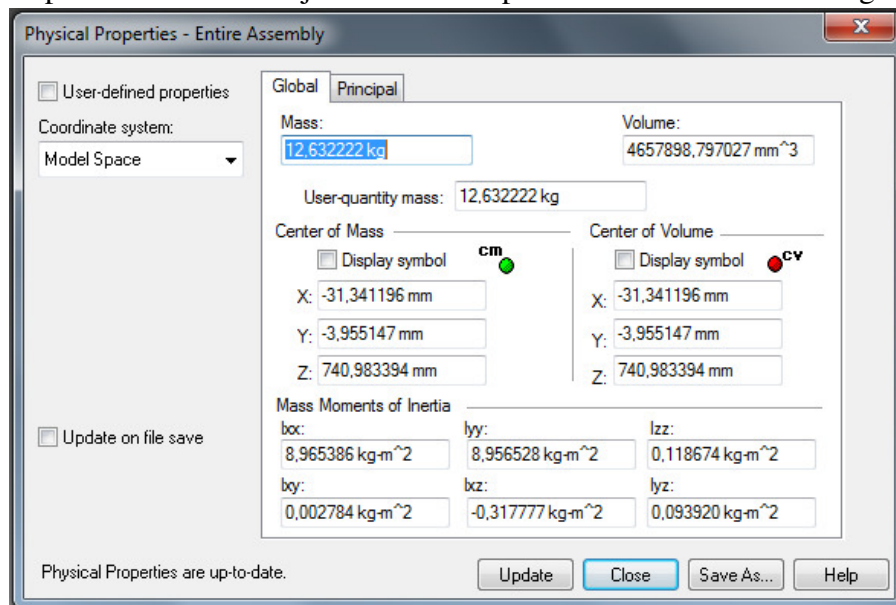
As informações apresentadas na Tabela 3 serão utilizadas para o cálculo dos torques das juntas 1, 2 e 3. Observe que a massa correspondente à junta 1 é a soma das massas das juntas 2 e 3.

Figura 7 – Estrutura do manipulador cilíndrico - *Software Solid Edge®*.



Fonte: O autor (2022).

Figura 8 – Propriedades físicas da junta 2 do manipulador - *Software Solid Edge®*.

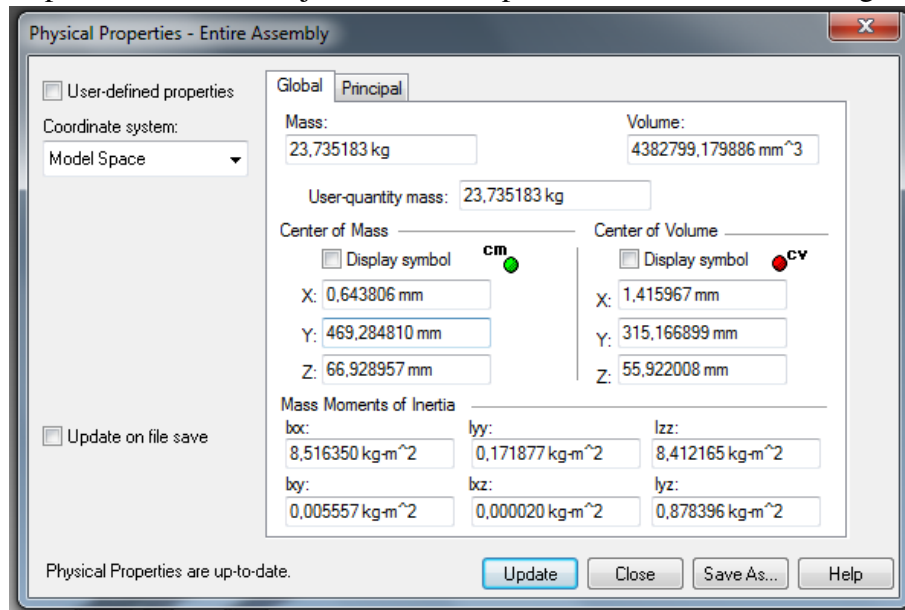


Fonte: O autor (2022).

2.6.3.1 Cinemática Inversa do manipulador do GPAR

A partir do desenvolvimento matemático da cinemática inversa apresentado anteriormente, aplicou-se o método geométrico ao robô manipulador do GPAR.

Figura 9 – Propriedades físicas da junta 3 do manipulador - Software Solid Edge®.



Fonte: O autor (2022).

Tabela 3 – Valores de massas (m) e comprimentos (l) das juntas.

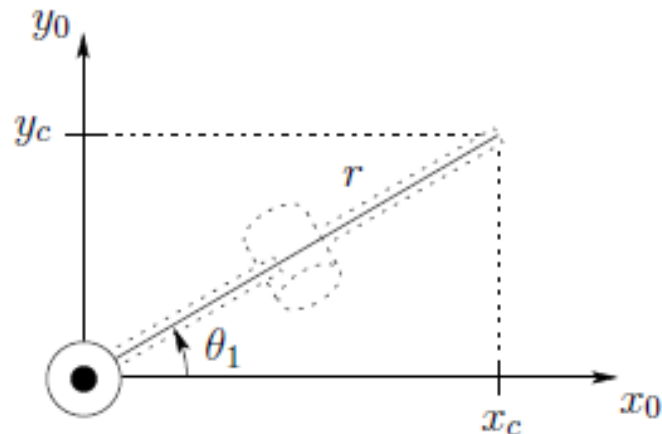
Junta	m (kg)	l (m)
1	36,367405	0,050
2	12,632222	0,790
3	23,735183	0,900

Fonte: O autor (2022).

A partir da Figura 10 pode-se encontrar θ_1 , que é definido como

$$\theta_1 = \text{Atan2}(P_x, P_y) \quad (2.117)$$

Figura 10 – Projeção da primeira junta $x_0 - y_0$.



Fonte: Adaptada Spong, 2008.

O parâmetro do elo 2 é prismático, portanto, d_2 está no mesmo eixo z_1 :

$$d_2 = P_z - 0,245. \quad (2.118)$$

No caso do terceiro parâmetro d_3 , irá se deslocar no plano formado por x e y , e pode ser determinado por

$$d_3 = (\sqrt{P_x^2 + P_y^2}) - 0,35 \quad (2.119)$$

As equações (2.117), (2.118) e (2.119) são as soluções do problema da cinemática inversa do manipulador cilíndrico e serão usadas para realizar a geração da trajetória do manipulador.

2.6.3.2 Modelagem do motor de indução trifásico como servo-posicionador

Como dito anteriormente as juntas do manipulador são acionadas por motores de indução trifásicos (MIT) utilizados como servo-posicionadores. Para se obter a modelagem dos MITs, foi aplicada a teoria de controle de campo orientado indireto, através da modelagem dinâmica do campo orientado apresentada em (RABELO L, 2017).

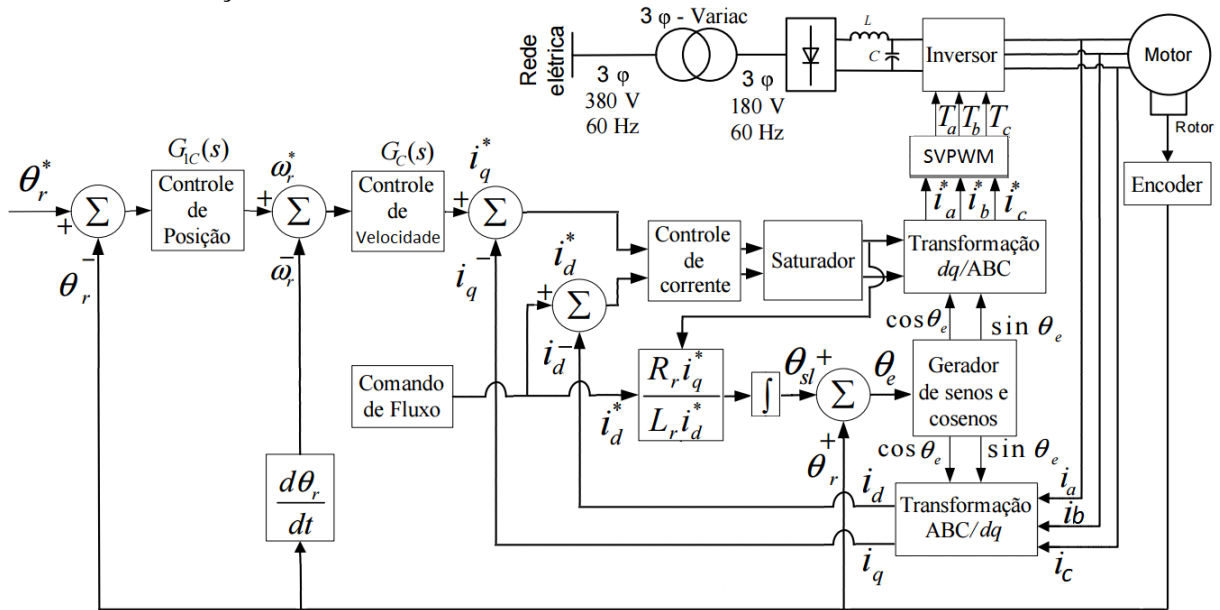
2.6.3.3 Modelagem dinâmica do campo orientado indireto para uma máquina de indução

O diagrama de blocos de controle de campo orientado (*Field Oriented Control* (FOC)) é mostrado na Figura 11. Este diagrama ilustra os blocos do inversor, fonte de tensão (*Voltage Source Inverter* (VSI)), o modulador de espaço vetorial baseado na técnica de modulação de largura de pulso (*Spatial Vector Pulse Width Modulation* (SVPWM)) e o bloco de controle com a posição de referência, que compõem a configuração do sistema. A descrição de cada um dos blocos do sistema da Figura 11 está apresentada em (RABELO L, 2017).

O modelo em espaço de estados do motor de indução no referencial dq rotativo é dado como segue: (BOSE, 1986):

$$\frac{d}{dt}[A] = [B][C] + \frac{1}{\sigma L_s}[D], \quad (2.120)$$

Figura 11 – Configuração do acionamento de campo orientado indireto para uma máquina de indução.



Fonte: RABELO L (2017).

onde:

$$[A] = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix}; \quad [C] = \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix}; \quad [D] = \begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix}; \quad (2.121)$$

e

$$B = \begin{bmatrix} -\frac{R_s}{\sigma L_s} - \frac{R_r(1-\sigma)}{\sigma L_r} & \omega_e & \frac{L_m R_r}{\sigma L_s L_r^2} & \frac{P \omega_r L_m}{2 \sigma L_s L_r^2} \\ \omega_2 & -\frac{R_s}{\sigma L_s} - \frac{R_r(1-\sigma)}{\sigma L_r} & -\frac{P \omega_r L_m}{2 \sigma L_s L_r^2} & \frac{L_m R_r}{\sigma L_s L_r^2} \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & \omega_e - \frac{P}{2} \omega_r \\ 0 & \frac{L_m R_r}{L_r} & -(\omega_e - \frac{P}{2} \omega_r) & -\frac{R_r}{L_r} \end{bmatrix} \quad (2.122)$$

onde

$$\sigma = 1 - \frac{L_m^2}{L_s L_r}; \quad \lambda_{qr} = L_m i_{qs} + L_r i_{dr}; \quad \lambda_{dr} = L_m i_{ds} + L_r i_{qr} \quad (2.123)$$

e o torque eletromagnético T_e é dado por:

$$T_e = \frac{3P}{4} \frac{L_m}{L_r} \left(i_{qs} \lambda_{dr} - i_{ds} \lambda_{qr} \right). \quad (2.124)$$

Em um campo orientado ideal de um motor de indução, o desacoplamento ocorre entre o motor e os eixos de quadratura. A partir desta metodologia, todas as equações foram

apresentadas em (SOUZA JÚNIOR, 2010). Desta forma, considerando-se as observações dadas a equação (2.124) pode ser reescrita pela equação (2.125) por:

$$T_e^* = \frac{3P L_m^2}{4 L_r} i_{qs}^* i_{ds}^* \quad (2.125)$$

A corrente i_{qs}^* indica que o controle de torque é controlado pela corrente do estator do eixo de quadratura, que é controlado pelo controlador $G_c(s)$ apresentado no diagrama da Figura 11. No método de campo orientado indireto, a frequência deve ser calculada nas coordenadas d_q0 . Usando-se a quarta linha da equação (2.121), a frequência de escorregamento pode ser calculada por:

$$\omega_{sl} = \frac{L_m R_r i_{qs}^*}{L_r \lambda_{dr}} = \frac{R_r i_{qs}^*}{L_r i_{ds}^*} \quad (2.126)$$

O torque T_e , a velocidade do rotor ω_r e a posição angular θ_r são relacionados por:

$$\omega_r = s\theta_r = \frac{1}{s + \frac{B}{J}} \left[T_e(s) - T_L(s) \right], \quad (2.127)$$

onde B é o coeficiente de atrito viscoso e J o momento de inércia. A equação (2.125) comprova o controle de campo orientado indireto, ou seja, é possível obter a posição da ligação do fluxo do rotor indiretamente através da posição do eixo da máquina e do ângulo de escorregamento entre os campos magnéticos do estator e do rotor.

2.6.3.4 Modulação por largura de pulso usando vetores espaciais

O chaveamento do inversor é executado pelo bloco da modulação de largura de pulso usando o espaço vetorial, SVPWM (*Space Vector Pulse Width Modulation*) apresentado na Figura 11. Neste utiliza-se o processador de sinais digital (DSP) utilizado no *hardware* de acionamento do manipulador cilíndrico do GPAR, modelo TMS320F228335[®] da *Texas Instrument*[®], possui suporte às funções de modulação em espaço vetorial, facilitando, assim, o acionamento do inversor trifásico utilizado no manipulador robótico e cujas vantagens foram apresentadas em (SOUZA JÚNIOR, 2010).

A técnica de chaveamento baseada no SVPWM baseada em (YU, 2001) foi apresentada conforme pode ser visto em (RABELO L, 2017), sendo dada como,

$$U_{out}(nT) = \frac{1}{T} (T_1 U_x + T_2 U_{x+60}) \quad (2.128)$$

onde os períodos T_1 e T_2 correspondem respectivamente à duração da comutação no estado correspondente aos vetores de base. Como $T_1 + T_2 \leq T_{pwm}$, o inversor deve permanecer no

período restante nos estados O_{000} ou O_{111} . A escolha destes vetores deve satisfazer o menor esforço de comutação do inversor (YU, 2001). Desse modo:

$$T_{pwm}U_{out} = T_1U_x + T_2U_{x+60} + T_0(O_{000,111}), \quad (2.129)$$

onde $T_0 = T_{pwm} - T_1 - T_2$.

Portanto, resolvendo a equação (2.129) para T_1 e T_2 os temporizadores no DSP são configurados com os valores obtidos, atribuindo-os aos respectivos registradores do processador. Desta forma, a partir da Figura 11 apresenta-se a implementação dos controles de posição, velocidade, corrente, bem como o chaveamento a partir do SVPWM ao motor de indução e conseqüentemente ao manipulador.

2.7 Considerações finais

Este capítulo, apresentou uma visão holística da modelagem do manipulador mostrando a matemática da cinemática inversa e direta e, por último a modelagem do motor de indução trifásico que é utilizado como servo-posicionador. Os conceitos apresentados são importantes para o entendimento de itens relevantes do manipulador robótico utilizado neste trabalho.

3 ALGORITMOS DE REGRESSÃO

3.1 Considerações iniciais

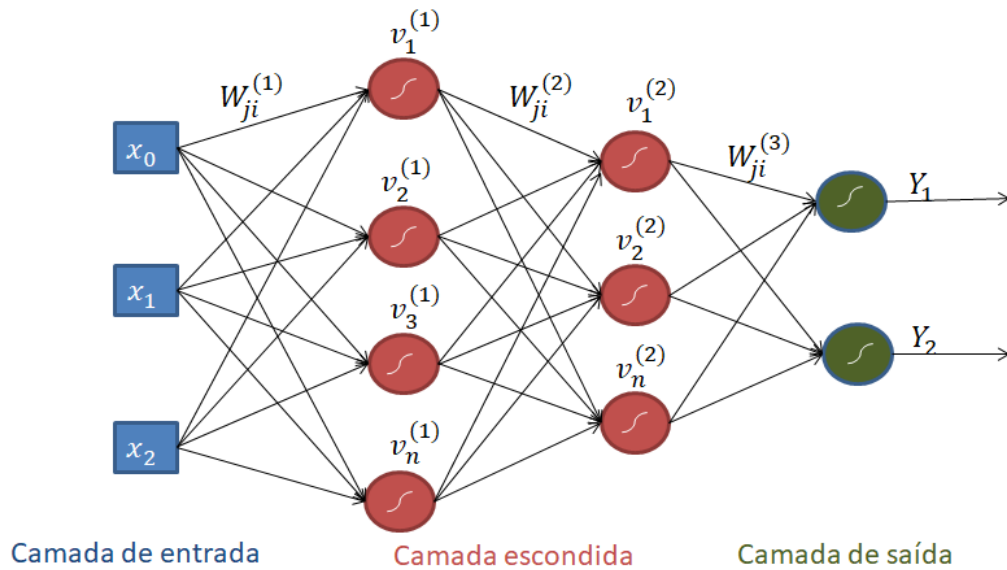
Neste capítulo, serão apresentados os algoritmos utilizados para uma análise comparativa de regressão de sistemas, bem como a proposta baseada em aprendizado de máquina.

Um modelo pode ser representado por teóricos e experimentais, Isermann e Münchhof (2011). Apesar de a aplicação destes modelos exigir a definição de uma estrutura, estes tornam-se capazes de modelar processos a partir de pouco ou nenhum conhecimento *a priori* sobre a física que rege o processo. De forma geral, os modelos entrada-saída são não-lineares e serão descritos neste capítulo para fins de comparação com o método proposto.

3.2 Rede Neural MLP

Um dos benefícios da utilização de Redes Neurais Artificiais (RNA) como método para resolver o problema apresentado é a possibilidade de obtenção de modelos lineares e não-lineares. A Rede Neural Multicamada Perceptron (MLP) é o tipo mais comumente usado para resolver problemas lineares ou não-lineares. Este tipo de RNAs é uma rede de aprendizado supervisionado, pois requer não apenas dados de entrada, mas também dados de saída para realizar o aprendizado de um mapeamento entre variáveis de entrada e saída (HAYKIN, 2001). A rede MLP tem como principal função criar um modelo que correlaciona entradas e saídas de um sistema. A Figura 12 apresenta uma RNA MLP.

Figura 12 – Estrutura da RNA MLP de duas camadas ocultas.



Fonte: O autor (2022).

A rede MLP pode aprender com um algoritmo chamado *backpropagation*. Portanto, é feita uma iteração dos dados de entrada na rede neural, onde em cada iteração a saída real da rede é comparada com a desejada, obtendo-se assim um erro. O objetivo é minimizar esse erro para ajustar os pesos até que a convergência desejada seja alcançada.

Algoritmo 1: Treinamento dos pesos da rede MLP.

- Etapa 1. Gerar aleatoriamente os pesos W_{ji} e Bias B_i , onde $i = 1, \dots, N$ dos N neurônios;
 - Etapa 2. Cálculo dos pesos da camada escondida;
 - Etapa 3. Cálculo dos pesos da camada de saída;
 - Etapa 4. Cálculo do erro;
 - Etapa 5. Treinamento por *backpropagation*.
-

O algoritmo de *backpropagation* é usado no treinamento de redes neurais multicamadas com uma ou mais camadas ocultas. Também o treinamento da RNA MLP por *backpropagation* é apresentado. A resposta de erro do neurônio j na iteração n é dado por:

$$e_j(n) = s_j(n) - y_i(n), \quad (3.1)$$

onde $s_j(n)$ é a resposta desejada para o neurônio j da camada de saída.

O valor instantâneo do erro quadrático para o neurônio j é definido por $\frac{1}{2}e_j^2(n)$. A soma dos erros quadráticos instantâneos da rede é então dada por:

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n), \quad (3.2)$$

onde C é o conjunto que contém todos os neurônios da camada de saída da rede.

Considerando-se N o número total de padrões de treinamento contidos no conjunto de treinamento. O erro quadrático médio é então definido por:

$$\varepsilon_{av} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n), \quad (3.3)$$

O Algoritmo 1 mostra o processo de treinamento onde o neurônio de saída ($y = j$) é alimentado pelas ativações de todos os neurônios da camada imediatamente anterior. A função de ativação neste trabalho foi a sigmóide logística. O nível de ativação interna do neurônio j é dado por:

$$v_j(n) = \sum_{i=0}^P w_{ji}y_i(n), \quad (3.4)$$

onde P representa o número de entradas sem o bias.

Portanto, a ativação $y_j(n)$ do neurônio j é dada por:

$$y_j(n) = f_j(v_j(n)), \quad (3.5)$$

Para minimizar o erro quadrático médio, primeiro é necessário determinar o gradiente instantâneo $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$ que é determinado aplicando-se a regra da cadeia, este gradiente é expressado como (HAYKIN, 2001):

$$\frac{\partial \varepsilon(n)}{\partial w_{ij}(n)} = \frac{\partial \varepsilon(n)}{\partial e_i(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (3.6)$$

A regra de ajuste é dada por (HAYKIN, 2001):

$$w_{ji}(n+1) = w_{ji}(n) - \eta(n) \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}, \quad (3.7)$$

onde $\eta(n)$ é a taxa de aprendizado na iteração n .

O algoritmo de treinamento por *backpropagation* é formulado como: a derivada parcial da soma dos erros instantâneos em relação ao peso $w_{ji}(n)$ que conecta o neurônio i ao neurônio j . O cálculo do gradiente local $\eta_j(n)$ depende se o neurônio é um neurônio de saída ou um neurônio oculto.

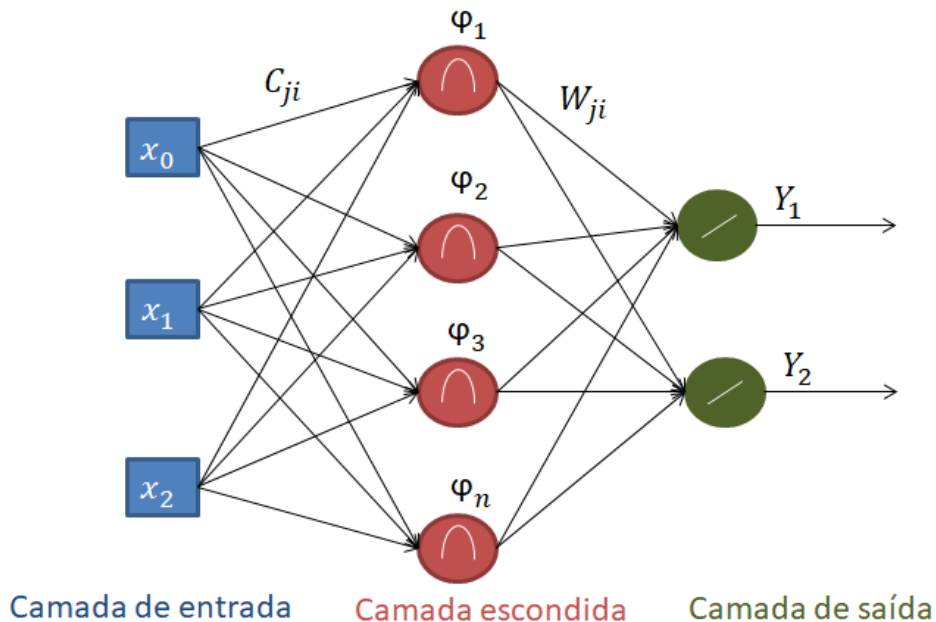
3.3 Rede neural de função de base radial

Uma rede de função de base radial *Radial Basis Function* (RBF) é a que utiliza funções de base radial como funções de ativação. A saída da rede é uma combinação linear

de funções de base radial dos parâmetros de entrada e neurônios. Estas têm muitas aplicações incluindo aproximação de função, previsão de séries temporais, classificação e controle do sistema, conforme apresentada em (WRIGHT, 2003).

A Figura 13 mostra a estrutura básica da rede neural RBF que consiste em três camadas. A camada de entrada interage entre a entrada e a camada oculta. A camada oculta executa uma transformação não-linear do vetor de entrada para um espaço vetorial interno tipicamente com uma dimensão maior, e a camada de saída realiza a transformação do espaço vetorial interno em uma saída usando um processo linear (WANG; LIU, 2002).

Figura 13 – Arquitetura de uma RNA RBF.



Fonte: O autor (2022).

As redes RBF constroem aproximações locais para mapeamentos de entrada-saída resultando em um aprendizado mais rápido e sensibilidade reduzida à ordem de apresentação do conjunto de dados durante o treinamento, onde se tenha um inteiro $P < N$ do conjunto, onde N é o número de exemplos de treinamento e P é o número de funções de base radial. A equação (3.8) fornece a saída do padrão neural da Figura 13:

$$\sum_{k=0}^P w_k \varphi(x; t_k) + w_o, \quad (3.8)$$

onde t_k representa os centros das funções de base radial.

Usando a regularização no treinamento dessa rede neural, a função ε_R a ser minimi-

zada é descrita por:

$$\varepsilon_R = \sum_{i=1}^N (d_i - f(x_i))^2 + \sum_{j=1}^P \lambda_j w_j^2, \quad (3.9)$$

onde $d = [d_1, d_2, \dots, d_N]^T$ é o vetor de respostas desejadas, $w = [w_0, w_1, w_2, \dots, w_p]^T$ é o vetor de pesos da rede neural RBF, λ_j é o parâmetro de regularização e $f(x_i)$ é a saída da rede neural para um vetor de entrada x_i .

No treinamento da rede neural RBF, encontram-se as localizações dos centros t_k das funções de base radial. A função de base radial escolhida para este trabalho foi a gaussiana, conforme definido na equação (3.10):

$$\varphi(x, t_k) = \exp\left(-\frac{1}{\sigma_k^2} \|x - t_k\|^2\right), \quad k = 1, 2, 3, \dots, P, \quad (3.10)$$

onde σ representa a largura da função radial.

Portanto, a saída da RNA RBF é dada por:

$$y = \sum_{k=0}^P w_k \exp\left(-\frac{1}{\sigma_k^2} \|x - t_k\|^2\right) + w_o. \quad (3.11)$$

A matriz de interpolação Φ cujo tamanho é $N \times (P + 1)$, é definida como segue

$$\Phi = \begin{bmatrix} 1 & \varphi(x_1, t_1) & \varphi(x_1, t_2) & \dots & \varphi(x_1, t_p) \\ 1 & \varphi(x_2, t_1) & \varphi(x_2, t_2) & \dots & \varphi(x_2, t_p) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \varphi(x_N, t_1) & \varphi(x_N, t_2) & \dots & \varphi(x_N, t_p) \end{bmatrix}, \quad (3.12)$$

Em termos matriciais, o vetor de pesos que minimiza a função de custo ε_R é dado por:

$$w = (\Phi^T \Phi + Q)^{-1} \Phi \cdot d \quad (3.13)$$

onde d é a saída desejada e Q é:

$$Q = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \lambda_r \end{bmatrix}. \quad (3.14)$$

Para o treinamento dos centros da rede RBF utilizou-se o método de aprendizagem não supervisionada *k-means*. O algoritmo destas equações foi proposto em (ZHANG; Li, 2006)

e é usado em aplicações de *clusterização*. Seja $c(t)_{j=1}^m$ o centro das funções de base radial na iteração t . O algoritmo *k-means* é descrito da forma:

Algoritmo 2: Treinamento dos centros com *k-means*.

Etapa 1. Escolha de valores aleatórios distintos para os centros $c_j(t)$;

Etapa 2. Pegue um vetor aleatório x_i do conjunto de padrões de entrada;

Etapa 3. Determine o índice k do centro mais próximo ao padrão de entrada como:

$$k(x_i) = \arg \min_j \|x_j(t) - c_j(t)\|, j = 1, 2, \dots, m;$$

Etapa 4. Ajuste os centros usando a regra de (3.15), em que $\gamma \in (0,1)$ é a taxa de ajuste;

Etapa 5. Repita as etapas 2 a 5 para todos os N padrões de entrada e até que os centros não mostrem alterações significativas após cada apresentação completa destes padrões.

$$\begin{cases} c_j(t+1) = c_j(t) + \gamma[x_i(t) - c_j(t)], k = k(x_i) \\ c_j(t), \end{cases} \quad (3.15)$$

A rede neural RBF foi utilizada com os dados do braço robótico que resultando uma configuração de parâmetros como sendo, 1 camada oculta, 10 funções de base radial e 1000 iterações. Outros detalhes dos parâmetros e ajustes serão vistos no próximo capítulo.

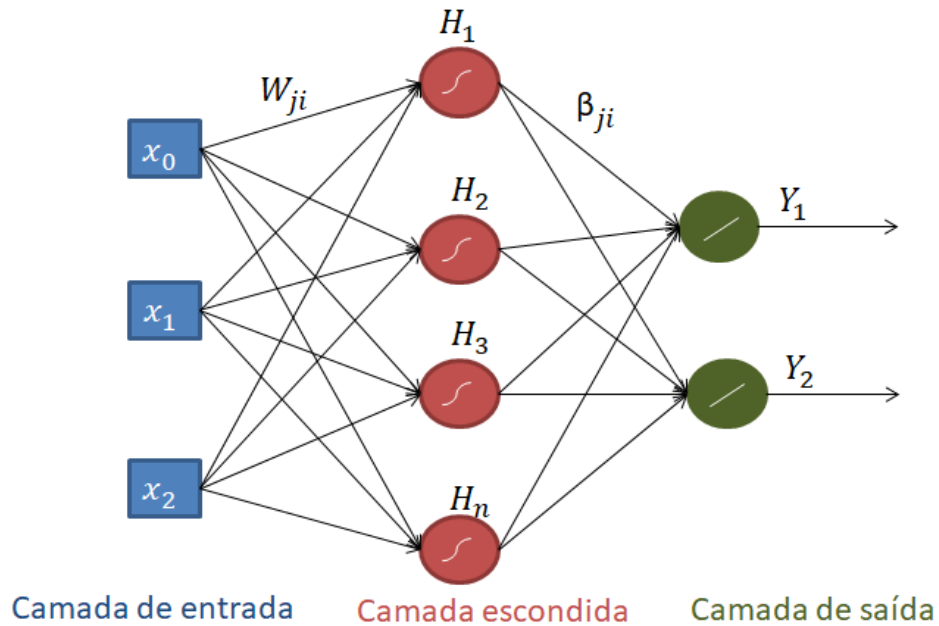
3.4 Rede Neural ELM

As máquinas de aprendizado extremo são redes neurais *feedforward* para classificação, regressão, clusterização, compactação e aprendizado de recursos com uma única camada ou várias camadas de neurônios ocultos. Esses neurônios ocultos podem ser atribuídos aleatoriamente e nunca atualizados. Na maioria dos casos os pesos de saída dos neurônios ocultos são geralmente aprendidos em uma única etapa, o que é basicamente equivalente a aprender um modelo linear (HUANG *et al.*, 2006).

A partir da Figura 14, o vetor X é a entrada da rede. Os pesos de conexão da camada de entrada são alocados em uma matriz chamada W e os da camada oculta em uma matriz chamada B . Para facilitar e agilizar os cálculos, os *bias* (-1) dos neurônios da camada oculta também são alocados na última linha de W e o *bias* (-1) da camada de saída não é usado na ELM. A modelagem da matriz ELM é descrita abaixo (ABDULLAH *et al.*, 2015):

$$X = [x_1, \dots, x_m, -1] \quad (3.16)$$

Figura 14 – Arquitetura da Rede ELM.



Fonte: O autor (2022).

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \cdots & \vdots \\ w_{m1} & \cdots & w_{md} \\ b_1 & \cdots & b_d \end{bmatrix}, \quad (3.17)$$

$$\beta = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1s} \\ \cdots & \vdots & \cdots \\ \beta_{d1} & \cdots & \beta_{ds} \end{bmatrix}, \quad (3.18)$$

$$Y = [y_1, \dots, y_s] \quad (3.19)$$

onde m é o número de neurônios de entrada, d é o número de neurônios na camada oculta e s é o número de neurônios de saída na rede.

Alguns trabalhos da rede ELM foram usados para resolver problemas de cinemática inversa em outros tipos de robôs (ZHOU *et al.*, 2018), (YUAN *et al.*, 2018) e (LI *et al.*, 2022). A rede ELM apresenta vantagens sobre as outras, tais como: os pesos de entrada e a polarização da camada oculta são escolhidos aleatoriamente, enquanto que os pesos da camada de saída são determinados analiticamente (HUANG *et al.*, 2006).

A matriz W é gerada aleatoriamente e não é alterada até o final do algoritmo. Esta pode ser gerada a partir de uma distribuição uniforme no intervalo $[-1,1]$. Portanto, o objetivo do treinamento ELM é encontrar a matriz de pesos B baseada na matriz de saída Y e na matriz de pesos aleatórios W , por meio da solução de uma equação linear.

Depois de inicializar a matriz de pesos aleatoriamente, o próximo passo é determinar a matriz H da seguinte maneira:

$$H^i = [x_1^i, \dots, x_m^i, -1] \times \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \cdots & \vdots & \cdots \\ w_{m1} & \cdots & w_{md} \\ b_1 & \cdots & b_d \end{bmatrix} \Rightarrow H = \begin{bmatrix} f(H^1) \\ f(H^2) \\ \vdots \\ f(H^N) \end{bmatrix}_{N \times d}, \quad (3.20)$$

onde $f(\cdot)$ é a função de ativação da camada e $i = 1, \dots, N$, N é o número de amostras do conjunto de treinamento. Assim, a matriz H armazena o resultado de todos os neurônios da camada oculta obtidos a partir da multiplicação entre X e W . A seguir obtém-se os pesos da matriz H para se encontrar os de B resolvendo-se o sistema linear:

$$H\beta = T \rightarrow \beta = H^\dagger Y \quad (3.21)$$

A matriz H^\dagger é a inversa generalizada de *Moore-Penrose* (HUANG *et al.*, 2006) e da matriz H . Se a inversa fosse usado, os algoritmos seriam limitados a problemas que esta inversa existisse. A matriz β é conhecida como a matriz de recursos, porque armazena as informações extraídas pela rede.

No algoritmo a seguir, é mostrada uma especificação passo-a-passo do procedimento de treinamento da rede.

Algoritmo 3: Treinamento da ELM.

Etapa 1. Selecione valores aleatórios para os pesos w_{ji} e bias b_{ji} , $i = 1, \dots, N$;

Etapa 2. Calcule H da camada oculta;

Etapa 3. Calcule os pesos β ;

Etapa 4. Calcule da camada de saída.

Os seguintes parâmetros da ELM foram selecionados por tentativa e erro a fim de obter-se o melhor resultado para o braço robótico, tais como, o número de neurônios na camada oculta igual a 15.

3.5 ANFIS

ANFIS são sistemas híbridos que unem uma rede neural e um sistema *fuzzy*. A maioria das aplicações iniciais estão no campo de controle de processos e, seguindo-se em todas as áreas do conhecimento como, análise de dados, classificação de dados, detecção de imperfeições e suporte à tomada de decisão (JOELIANTO *et al.*, 2013).

Redes neurais e sistemas *fuzzy* são combinados para unir suas vantagens e suprir seus pontos negativos individuais. As possuem suas características computacionais de aprendizagem nos sistemas difusos e recebem deles a interpretação e clareza da representação dos sistemas. Assim, as desvantagens dos sistemas difusos são compensadas pelas capacidades das redes neurais. Essas técnicas são complementares o que justifica seu uso em conjunto (HOU *et al.*, 2003).

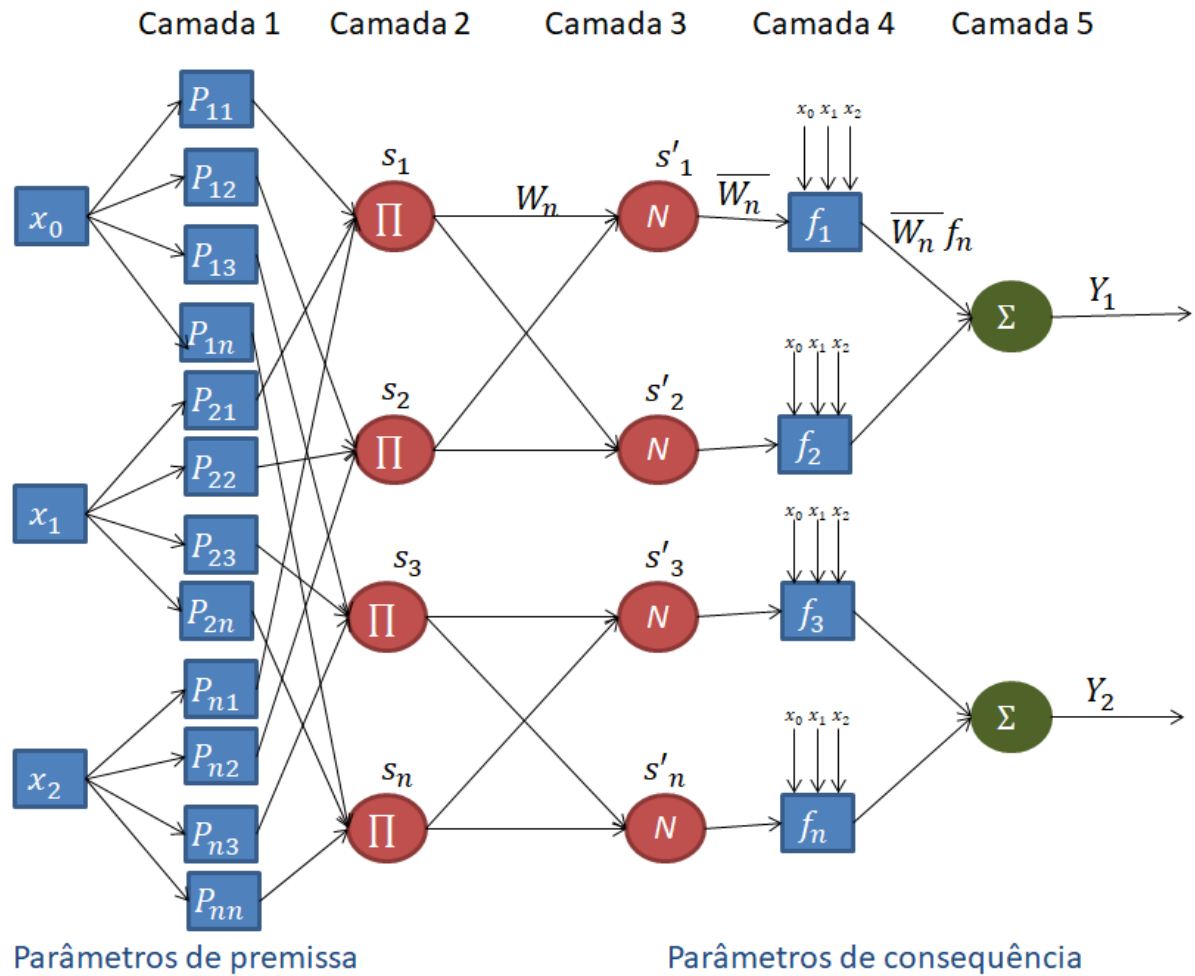
O sistema de inferência *Fuzzy* adaptável baseado em rede (*Adaptive Network based Fuzzy Inference System* - ANFIS) implementa um sistema de inferência *Fuzzy Takagi Sugeno* que tem cinco camadas (HOU *et al.*, 2003). A primeira camada oculta é responsável pelo mapeamento da variável de entrada relativamente a cada função de associação. O operador *t-norma* é aplicado na segunda camada oculta para calcular os antecedentes das regras. A terceira camada oculta normaliza as forças das regras seguidas pela quarta camada oculta onde os consequentes das regras são determinados. A camada de saída calcula a saída global como a soma de todos os sinais que chegam a essa camada.

O ANFIS usa o aprendizado *backpropagation* para determinar os parâmetros de funções de associação de entrada e o método de menor média quadrada para determinar os parâmetros consequentes. Cada etapa do algoritmo de aprendizado iterativo tem duas partes. Na primeira parte, os padrões de entrada são propagados e os parâmetros dos consequentes são calculados usando o algoritmo do método iterativo mínimo, enquanto os parâmetros das premissas são considerados fixos. Na segunda parte, os padrões de entrada são propagados novamente e em cada iteração o algoritmo de aprendizagem *backpropagation* é usado para modificar os parâmetros das premissas, enquanto os consequentes permanecem fixos. A Figura 15 apresenta a arquitetura usada para o sistema ANFIS.

Com base na Figura 15, tem-se:

- A camada 1 com a função de associação gaussiana que foi usada, onde os parâmetros podem ser tratados como premissa.
- Na camada 2 é calculado o grau de pertinência para cada regra. Os neurônios desta camada

Figura 15 – Arquitetura da rede ANFIS.



Fonte: O autor (2022).

usam a norma t – norma, conforme as regras a seguir.

$$S_1 = P_{11}(x_0) * P_{21}(x_1) * P_{n1}(x_2)$$

$$S_2 = P_{12}(x_0) * P_{22}(x_1) * P_{n2}(x_2)$$

$$S_3 = P_{13}(x_0) * P_{23}(x_1) * P_{n3}(x_2)$$

$$S_n = P_{14}(x_0) * P_{24}(x_1) * P_{n4}(x_2)$$

(3.22)

onde '*' representa t – norma.

- Na camada 3 é feita a normalização da regra, onde ocorre a ativação. Os neurônios dessa camada os quais recebem um pré-processamento resultam em uma desfuzzificação, dada

por,

$$\begin{aligned}
 S'_1 &= S_1 / (S_1 + S_2 + S_3 + S_n) \\
 S'_2 &= S_2 / (S_1 + S_2 + S_3 + S_n) \\
 S'_3 &= S_3 / (S_1 + S_2 + S_3 + S_n) \\
 S'_n &= S_n / (S_1 + S_2 + S_3 + S_n)
 \end{aligned}
 \tag{3.23}$$

- Na camada 4, tem-se o cálculo do produto das normalizações e o grau de ativação consequente, que são dados por:

$$\begin{aligned}
 h_1 &= S_1 / (S'_1 + C_1) \\
 h_2 &= S_2 / (S'_1 + C_2) \\
 h_3 &= S_3 / (S'_1 + C_3) \\
 h_n &= S_n / (S'_1 + C_n)
 \end{aligned}
 \tag{3.24}$$

onde C_i são os parâmetros atribuídos aos consequentes.

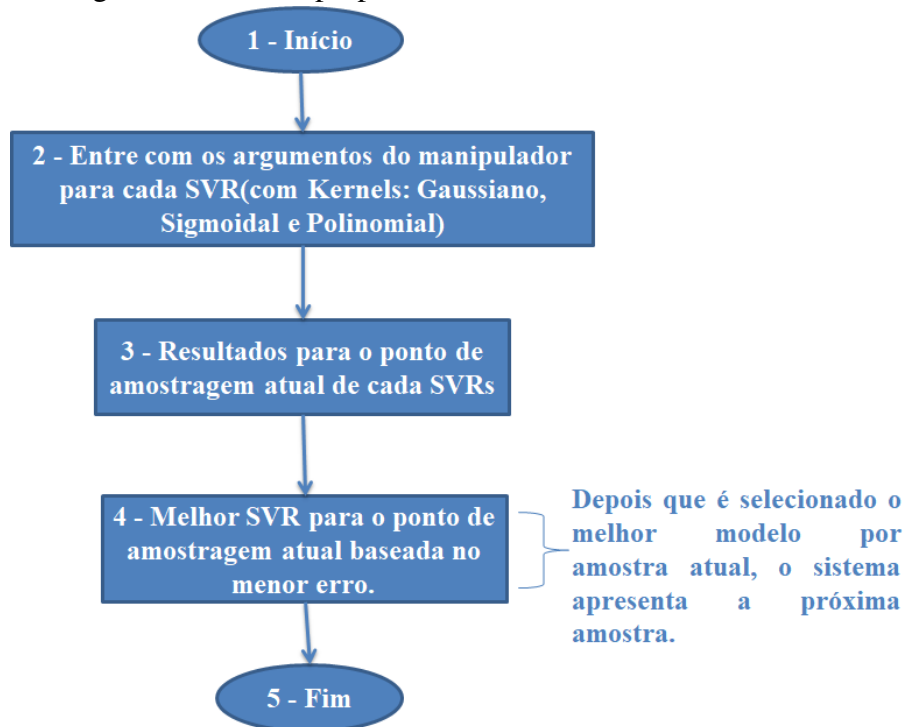
- Por último, a camada 5 é a saída que gera o resultado final do sistema, sendo dada por:

$$\begin{aligned}
 y_n &= \frac{\sum S_i \cdot f_i}{\sum S_i} = \sum S'_i \cdot f_i \\
 y_1 &= h_1 + h_2 \\
 y_2 &= h_3 + h_n
 \end{aligned}
 \tag{3.25}$$

3.6 Método de Regressão Proposto

A proposta deste trabalho é baseada na teoria de regressão utilizando máquinas de comitê onde foi feita uma implementação de um comitê de 3 regressores de vetor suporte (*support vector regression* - SVR) com diferentes tipos de *kernels* que são combinados com a resposta de múltiplas saídas, cujo objetivo é obter-se uma única resposta que será baseada no menor erro. A Figura 16 apresenta o fluxograma do método proposto.

Figura 16 – Fluxograma do comitê proposto.



Fonte: O autor (2022).

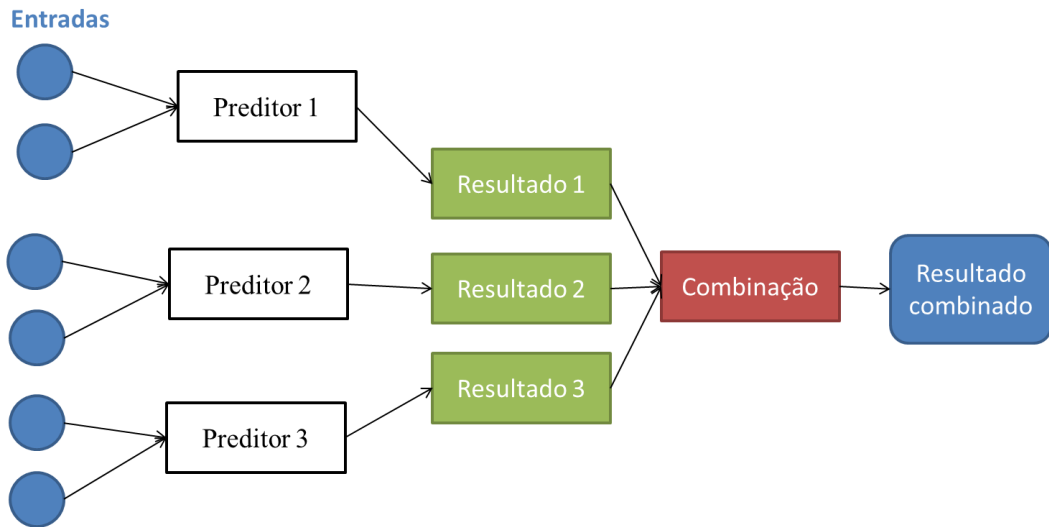
3.6.1 Máquinas de Comitê

Uma máquina de comitê é um tipo de método que usa uma estratégia de divisão e conquista na qual as respostas de múltiplas especialistas são combinadas em uma única resposta (NADIRI *et al.*, 2013). A resposta combinada deve ser superior àquela de seus especialistas constituintes. A Figura 17 apresenta a estrutura de máquina de comitê proposta.

Existem duas classes de máquinas de comitê: estática e dinâmica. Na classe estática as respostas de vários preditores (*experts*) são combinadas por meio de um mecanismo que não envolve o sinal de entrada, daí a designação estática. Esta categoria inclui os seguintes métodos: *Ensemble* médio e *Boosting*. O *Ensemble* médio tem as saídas de diferentes preditores linearmente combinadas para produzir uma saída global. Neste método utilizam-se saídas diferentes de preditores linearmente combinados, cuja diferença está na técnica de conjunto que tenta criar um preditor forte a partir de vários especialistas fracos (CIRESAN *et al.*, 2011).

Na classe de máquinas de comitê dinâmica o sinal de entrada está diretamente envolvido no acionamento do mecanismo que integra as saídas dos especialistas individuais em uma saída global, daí a dinâmica de designação. Existem dois tipos de estruturas dinâmicas: mistura de especialistas e mistura hierárquica de especialistas. Na mistura de especialistas as

Figura 17 – Estrutura da máquina de comitê proposta.



Fonte: O autor (2022).

respostas individuais das técnicas são combinadas de forma não-linear por meio de uma única rede de bloqueio (NADIRI *et al.*, 2017). Já a técnica de mistura hierárquica de especialistas as respostas individuais dos especialistas são combinadas de forma não-linear por meio de várias redes de conexão organizadas de maneira hierárquica.

3.6.2 Regressão de Vetor Suporte

As máquinas de vetores de suporte (*support vector machines* - SVMs) são métodos de aprendizagem supervisionados com algoritmos de aprendizagem associados que analisam os dados utilizados para problemas de classificação e análise de regressão (CORTES; VAPNIK, 1995). A regressão SVM é considerada uma técnica não paramétrica porque depende das funções do *kernel*.

Assim, supõem-se que em um conjunto de dados para treinamento $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset X \times R$, onde X denota o espaço dos padrões de entradas (por exemplo, $X = R^d$). Na regressão ε -SV, tem o objetivo de descobrir a função $f(x)$ que tem uma aproximação das saídas desejadas. Esta função pode ser descrita na forma de (CORTES; VAPNIK, 1995):

$$f(x) = \langle w, x \rangle + b \text{ com } w \in X, b \in R \quad (3.26)$$

onde $\langle \dots \rangle$ denota o produto de ponto em X . A construção da equação 3.26 significa que se procura um w bem baixo. Uma das maneiras de garantir isso é minimizar $\|w\|^2 = \langle w, w \rangle$, isso

pode ser escrito como um problema de otimização convexa:

$$\min \frac{1}{2} \|w\|^2, \quad \text{sujeito a} \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \quad (3.27)$$

A suposição na equação (3.27) é tal que a função f existente aproxima todos os pares (x_i, y_i) com precisão ε , ou seja, o problema de otimização convexa é viável. Portanto, a função de perda chamada de *softmargin*, proposta por (BENNETT; MANGASARIAN, 1992) e que foi adaptada para máquinas de SV por (CORTES; VAPNIK, 1995) faz uso de variáveis de relaxamento $\varepsilon_i, \varepsilon_i^*$ para lidar com outras restrições do problema de otimização dadas na equação 3.27. Logo pode-se usar a formulação da equação 3.28 citada por (CORTES; VAPNIK, 1995).

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad \text{sujeito a} \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (3.28)$$

A constante $C > 0$ determina o *trade-off* entre o estado plano de f e a quantidade até que desvios maiores do que ε são tolerados. Isso é correspondente com a chamada função de perda ε -insensível de $|\xi|_\varepsilon$ descrito por:

$$|\varepsilon|_\varepsilon := \begin{cases} 0 & \text{se } |\xi| \leq \varepsilon \\ |\varepsilon| - \varepsilon & \text{de outra forma} \end{cases} \quad (3.29)$$

A formulação dupla, equação (3.29), fornece a chave para estender a SVM a funções não-lineares. Desta forma, será usado um método de dualização padrão utilizando multiplicadores de Lagrange, como descrito em (FLETCHER, 1991).

A ideia chave é construir uma função de Lagrange a partir da função objetiva e as restrições correspondentes introduzindo-se um conjunto dual de variáveis. Mostra-se que esta função tem um ponto em relação às variáveis primárias e duplas na solução. Desse modo, procede-se da seguinte forma:

$$\begin{aligned} L := & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=1}^l \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle - b) \\ & - \sum_{i=1}^l \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle + b) \end{aligned} \quad (3.30)$$

Na equação (3.30), L é o Lagrangiano e $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ são multiplicadores de Lagrange. Portanto, essas variáveis devem satisfazer as restrições de positividade, ou seja,

$$\alpha_i^{(*)}, \eta_i^{(*)} \geq 0. \quad (3.31)$$

Segue-se da condição do ponto de relaxamento, equação (3.31), que as derivadas parciais de L em relação às variáveis primárias (w, b, ξ_i, ξ_i^*) têm que desaparecer para ser ótimo.

$$\partial_b L = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0 \quad (3.32)$$

$$\partial_w L = w - \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i = 0 \quad (3.33)$$

$$\partial_{\xi_i} L = C - \alpha_i^* - \eta_i^* = 0 \quad (3.34)$$

Substituindo-se (3.32), (3.33) e (3.34) em (3.30) encontra-se o problema de otimização dupla dado por:

$$\max \begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \end{cases} \text{ sujeito a } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ e } \alpha_i, \alpha_i^* \in [0, C] \quad (3.35)$$

Na derivação da equação (3.35) eliminou-se as variáveis duplas η_i, η_i^* através da condição da equação (3.34) que pode ser reformulada como $\eta_i^{(*)} = C - \alpha_i^{(*)}$. A equação 3.33 pode ser reescrita da seguinte forma:

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i, \quad \text{portanto } f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b. \quad (3.36)$$

A equação (3.36) corresponde à expansão vetor de suporte w que pode ser completamente descrita como uma combinação linear dos padrões de treinamento x_i . De certo modo, a complexidade da representação de uma função por SVs é independente da dimensionalidade do espaço de entrada X e depende apenas do número de SVs. Além disso, observa-se que o algoritmo completo pode ser descrito em termos de produtos de pontos entre os dados. Mesmo quando avalia-se $f(x)$, não há necessidade de se computar w explicitamente. Isto é útil para a formulação de uma extensão não-linear.

Para se calcular b exploram-se as chamadas condições de Karush-Kuhn-Tucker (KKT) (KUHN; TUCKER, 1951). Estas condições afirmam que no ponto da solução o produto entre variáveis duplas e restrições deve desaparecer.

$$\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) = 0 \quad \alpha_i(\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle + b) = 0 \quad (3.37)$$

$$(C - \alpha_i)\xi_i = 0 \quad e \quad (C - \alpha_i^*)\xi_i^* = 0 \quad (3.38)$$

Com isso é possível que apenas as amostras (x_i, y_i) com $\alpha_i^* = C$ correspondentes situem-se fora da área ε -insensível. Uma outra conclusão é que: $\alpha_i \alpha_i^* = 0$, isto é, nunca pode existir um conjunto de variáveis duplas α_i, α_i^* que são simultaneamente não-zero, com isso tem-se:

$$\varepsilon - y_i + \langle w, x_i \rangle + b \geq 0 \quad e \quad \xi_i = 0 \quad \text{Se } \alpha_i < C \quad (3.39)$$

$$\varepsilon - y_i + \langle w, x_i \rangle + b \leq 0 \quad \text{Se } \alpha_i > 0 \quad (3.40)$$

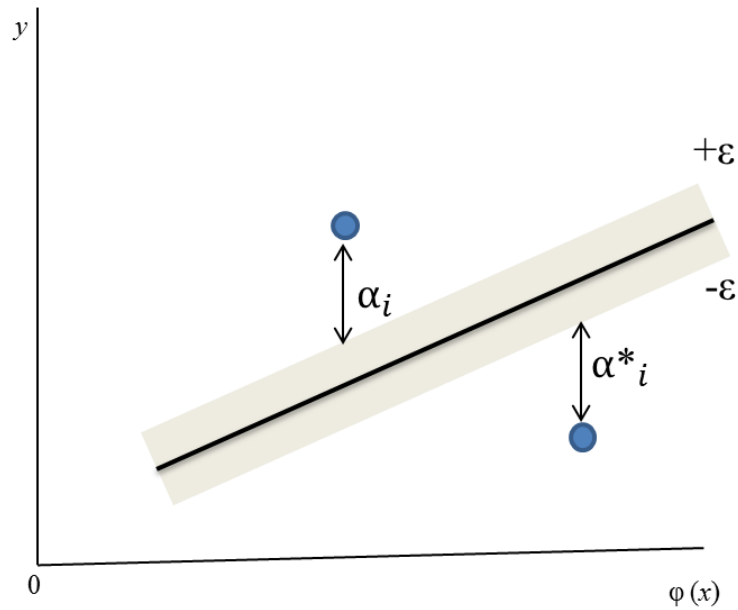
Em conjunto com uma análise análoga em α_i^* , tem-se:

$$\max \left\{ -\varepsilon + y_i - \langle w, x_i \rangle \mid \alpha_i < C \text{ ou } \alpha_i^* > 0 \right\} \leq b \leq \quad (3.41)$$

$$\min \left\{ -\varepsilon + y_i - \langle w, x_i \rangle \mid \alpha_i > 0 \text{ ou } \alpha_i^* < C \right\} \quad (3.42)$$

Se algum $\alpha_i^{(*)} \in (0, C)$ as desigualdades se tornam iguais, conforme (KEERTHI *et al.*, 2006), para outros meios de escolha de b . De acordo com a equação (3.37), somente se $|f(x_i) - y_i| \geq \varepsilon$ os multiplicadores de Lagrange podem ser diferentes de zero ou, ou seja, para todas as amostras dentro da região sombreada na Figura 18, o α_i, α_i^* . Para $|f(x_i) - y_i| < \varepsilon$, o segundo fator na equação 3.37 é diferente de zero. Portanto, α_i, α_i^* é zero, na forma que as condições de KKT são satisfeitas.

Figura 18 – SVM



Fonte: O autor (2022).

3.6.2.1 Mapeamento via Kernel

O próximo passo é tornar o algoritmo SV não linear. Para isso será abordado o mapeamento via *kernel*. Tem o objetivo de projetar os vetores de características de entrada em um espaço de características de alta dimensão para problemas que se encontram em espaços não linear. Esses produtos escalares do *kernel* são também chamados de truque do *kernel*. A equação (3.43) define a função do *kernel* K , que recebe dois pontos x_i e x_j do espaço de entradas:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (3.43)$$

O algoritmo SV depende somente dos produtos de pontos entre os padrões x_i , quando se pretende fazer o mapeamento $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ com $\Phi(x_1, x_2)$. Portanto, é suficiente conhecer $K(x, x')$: $= \langle \Phi(x), \Phi(x') \rangle$ ao invés de conhecer Φ explicitamente, o que permite reafirmar o problema de otimização de SV:

$$\begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) \\ -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \end{cases} \text{sujeito a } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ e } \alpha_i, \alpha_i^* \in [0, C] \quad (3.44)$$

Da mesma forma, a expansão de $f(x)$ na equação (3.36) pode ser escrita como:

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi x_i, \quad e \quad f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x) + b. \quad (3.45)$$

A diferença para o caso linear é que $f(x)$ já não é dada explicitamente. Na configuração não-linear, o problema de otimização corresponde a encontrar a função mais plana no espaço de características, não no espaço de entrada.

Existem várias funções *kernels*, porém no presente trabalho serão usadas apenas 3: polinomial, gaussiana e sigmoideal.

A função polinomial é definida por

$$K(x_i, x_j) = (\delta(x_i \cdot x_j) + k)^d, \quad (3.46)$$

onde d é o polinômio e δ, k são parâmetros de ajuste.

A função gaussiana é dada como

$$K(x_i, x_j) = \exp(-\sigma \|x_i - x_j\|^2), \quad (3.47)$$

sendo σ o parâmetro de ajuste.

Finalmente, a função sigmoideal é expressa por

$$K(x_i, x_j) = \tanh(\delta(x_i \cdot x_j) + k), \quad (3.48)$$

onde δ e k são parâmetros de ajuste.

O tipo de comitê que a proposta baseia-se no tipo de estático, pois cada especialista(SVR) terá sua saída individual, logo o critério de seleção é acionado para obter uma única reposta.

3.7 Considerações finais

Neste capítulo, apresentou-se uma descrição de técnicas de aprendizado de máquinas, bem como a técnica proposta para aplicação no manipulador. O estudo dessas técnicas consiste em formular uma base teórica, facilitando assim a adaptação ou uma análise mais apurada com a intenção de contribuir para a área de aprendizado de máquinas.

4 RESULTADOS

4.1 Considerações iniciais

Este capítulo apresenta os resultados da análise comparativa entre o método proposto e os outros 4 métodos apresentados no capítulo 3. O primeiro passo desta análise é a modelagem de todas as características do manipulador em estudo com o objetivo de gerar modelos que possam auxiliar os modelos cinemáticos e dinâmicos. Em seguida, os métodos são utilizados para realizar o rastreamento de algumas trajetórias que serão dadas como forma de validação. Além do mais, os métodos são testados com trajetórias onde a carga no *end-effector* é variável durante a realização da tarefa, sendo assim um problema mais desafiador. Finalmente, os modelos também são avaliados com dados obtidos experimentalmente. Além disso, para este estudo foram utilizados índices de desempenho baseados na validação-cruzada do tipo *hold-out*, e as métricas baseadas no coeficiente de determinação ajustável (R_a^2) e na Raiz do erro quadrático médio (RMSE).

4.2 Validação-cruzada

A etapa de validação é importante para o ajuste de modelos para que eles possam ser testados com novos dados. O processo de ajuste otimiza os parâmetros do modelo para que este se ajuste aos dados de treinamento da melhor forma possível. Se amostras de dados de validação forem retiradas da mesma população que os dados de treinamento, geralmente o modelo não se ajusta tão bem quanto quando se tem dados de validação independentes dos de treinamento. O tamanho dessa diferença pode ser grande especialmente quando o tamanho do conjunto de dados de treinamento for pequeno ou quando o número de parâmetros no modelo for grande. A validação cruzada é uma forma de estimar o tamanho desse efeito.

Existem alguns tipos de validações cruzadas como: *leave one out*, *k-fold* e *hold-out*. A *hold-out* é o tipo mais simples de validação cruzada. O conjunto de dados é separado em dois conjuntos, de treinamento e de teste. O aproximador de função ajusta uma função usando apenas o conjunto de treinamento. Em seguida, o aproximador de função é solicitado a prever os valores de saída para os dados no conjunto de teste, que são diferentes dos valores de saída dados anteriormente. Os erros que ele comete são acumulados, como antes, para fornecer o erro médio absoluto do conjunto de teste, que é usado para avaliar o modelo (KOHAVI, 1995).

A validação cruzada *k-fold* é uma maneira de melhorar o *hold-out*. O conjunto de dados é dividido em k subconjuntos e o mesmo é repetido k vezes. A cada vez, um dos k subconjuntos é usado como conjunto de teste e os outros $k-1$ subconjuntos são colocados juntos para formar um conjunto de treinamento. Em seguida, o erro médio em todas as k tentativas é calculado. A vantagem desse método é que não importa como os dados são divididos. Cada ponto de dados fica em um conjunto de teste exatamente uma vez e fica em um conjunto de treinamento $k-1$ vezes. A variância da estimativa resultante é reduzida como k é aumentado. A desvantagem desse método é que o algoritmo de treinamento precisa ser executado novamente, desde o início, k vezes, o que significa que são necessárias k vezes mais computações para fazer uma avaliação. Uma variante desse método é dividir aleatoriamente os dados em um conjunto de teste e treinamento em k tempos diferentes. A vantagem de se fazer isso é que se pode escolher independentemente o tamanho de cada conjunto de teste e quantas tentativas, em média, se calculam.

A validação cruzada *leave-one-out* é a validação cruzada *k-fold* levada ao seu extremo, com k igual a N , onde N é o número de dados no conjunto. Isso significa que N vezes separadas, o aproximador da função é treinado em todos os dados, exceto em um ponto, e uma previsão é feita para esse ponto. Como antes, o erro médio é calculado e usado para avaliar o modelo. A avaliação dada pelo erro de validação cruzada *leave-one-out* é boa, mas na primeira passagem parece muito pesado para se calcular. No entanto, os modelos com peso local podem fazer previsões com a mesma facilidade com que fazem previsões regulares.

O tipo de validação cruzada usada no trabalho foi a *hold-out*, por ter algumas vantagens em relação às outras.

4.3 Métricas de avaliação

Uma análise quantitativa dos métodos é dada utilizando-se as métricas RMSE (EN-TEKHABI *et al.*, 2010) e o coeficiente de determinação ajustável, R_a^2 (LIAO; MCGEE, 2003), que são definidas por:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_{(i)} - y_{(i)})^2}{N}} \quad (4.1)$$

e

$$R_a^2 = 1 - \left(\frac{N-1}{N-(P+1)} \right) (1 - R^2), \quad (4.2)$$

onde

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_{(i)} - \hat{y}_{(i)})^2}{\sum_{i=1}^n (y_{(i)} - \bar{y}_{(i)})^2}, \quad (4.3)$$

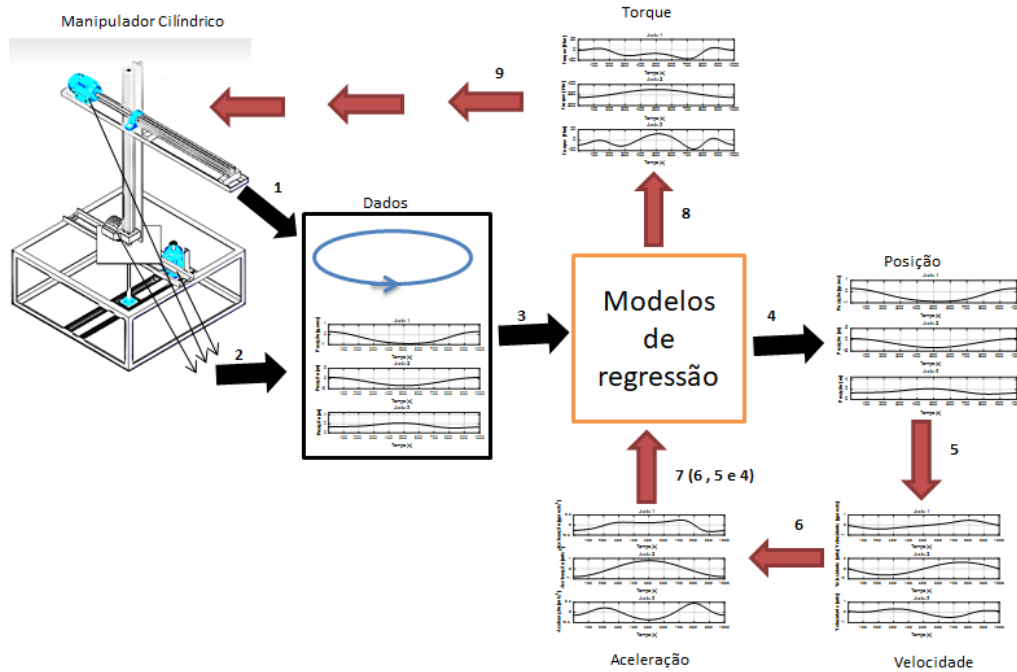
é o coeficiente de determinação, $y_{(i)}$ é a observação, $\hat{y}_{(i)}$ é previsão, $\bar{y}_{(i)}$ é a média das observações, N é o número de observações e P é o número de parâmetros.

4.4 Resultados de simulações

Na presente seção será abordada a simulação da cinemática inversa e dinâmica inversa com a utilização de algumas técnicas de regressão baseadas em aprendizado de máquina. A geração do conjunto de dados foi feita a partir das equações apresentadas no capítulo 2 da cinemática e da dinâmica do manipulador cilíndrico do GPAR. Assim, considerou-se duas etapas, em que a primeira é a geração de dados para o cálculo da cinemática inversa com as coordenadas cartesianas no espaço de trabalho e entradas x , y e z , representando a posição do *end-effector*, e como saídas as trajetórias das posições de cada junta como: θ_1 e d_3 . Os resultados da junta d_2 não serão mostrados, visto que, para a maioria destes resultados, ela apresenta valores constantes ou que não variam consideravelmente. A segunda etapa é a aproximação do modelo de dinâmica inversa que, a partir das trajetórias das posições geradas na primeira etapa, utiliza como entradas as velocidades e acelerações de cada junta e, como saída, o torque τ de cada junta.

A Figura 19 representa a metodologia de desenvolvimento do sistema em que tem-se os métodos de regressão aplicados. Para tanto, nesta etapa de simulação, tem-se a coleta de todos os dados através de um modelo cinemático e dinâmico conforme as tarefas a serem realizadas. Após a aquisição de dados de 1001 amostras para cada trajetória, os modelos foram treinados e testados com a técnica de validação cruzada baseada em *hold-out*, e usou-se o software matlab para fazer as simulações.

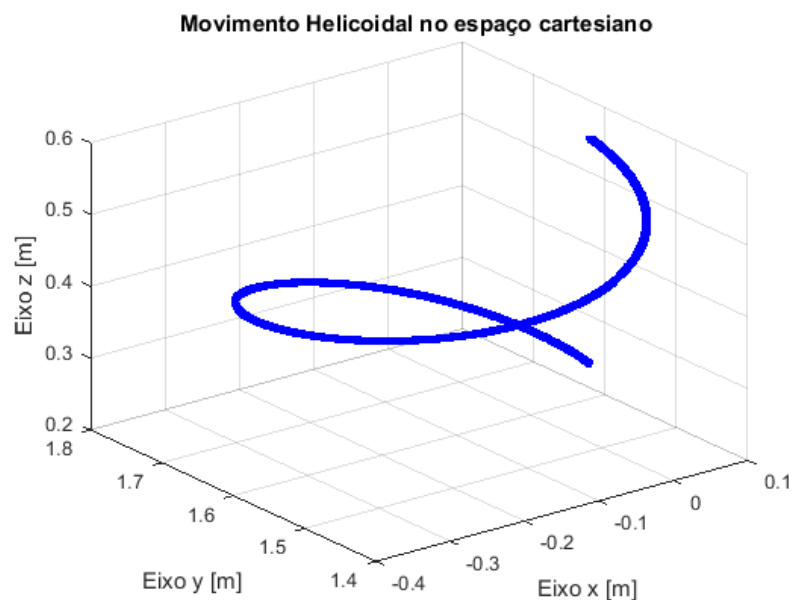
Figura 19 – Regressão da cinemática e dinâmica inversa a partir dos dados gerados por simulação.



Fonte: O autor (2022).

A Figura 20 demonstra a trajetória helicoidal do *end-effector*, no espaço cartesiano, que será utilizada na etapa de treinamento e teste dos modelos de regressão.

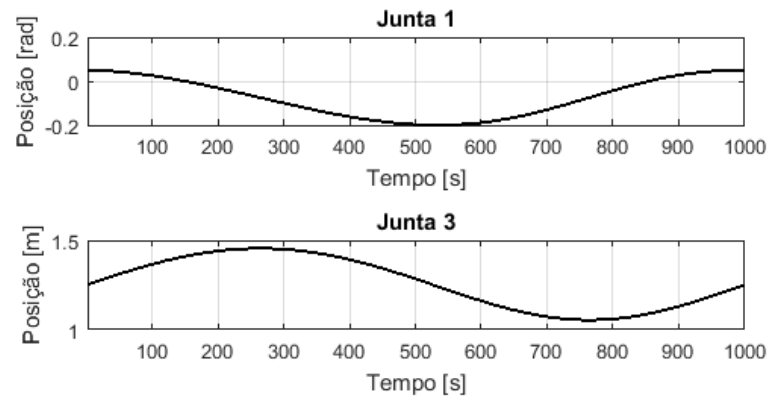
Figura 20 – Trajetória helicoidal no espaço cartesiano.



Fonte: O autor (2022).

A Figura 21 apresenta as posições das juntas durante a trajetória helicoidal mostrada na Figura 20.

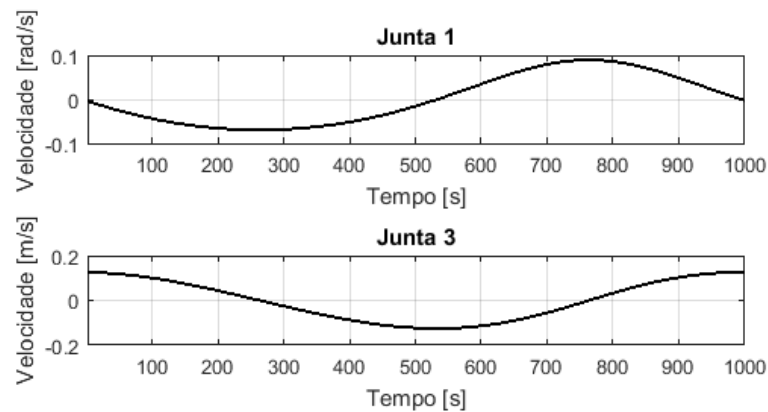
Figura 21 – Posições no espaço das juntas durante a trajetória helicoidal.



Fonte: O autor (2022).

A Figura 22 apresenta as velocidades nos espaços das juntas durante a trajetória helicoidal do *end-effector*.

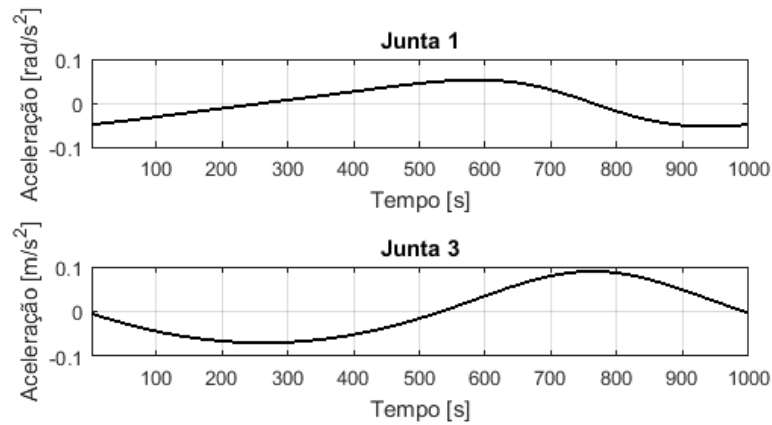
Figura 22 – Velocidades no espaço das juntas durante a trajetória helicoidal



Fonte: O autor (2022).

A Figura 23 mostra as acelerações nos espaços das juntas durante a trajetória helicoidal do *end-effector*.

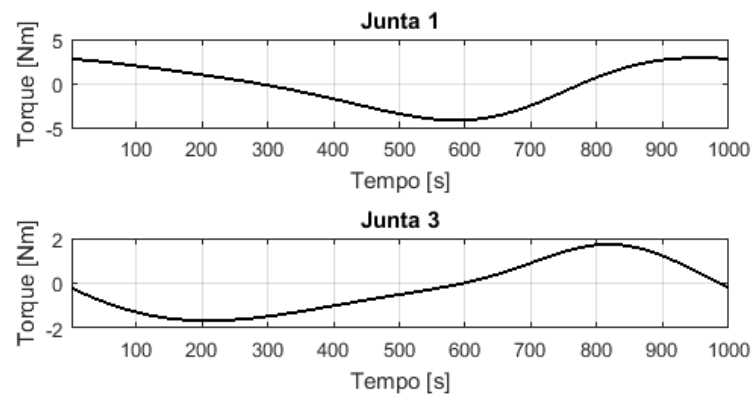
Figura 23 – Aceleração no espaço das juntas durante a trajetória helicoidal



Fonte: O autor (2022).

Os torques são apresentados na Figura 24 nos espaços das juntas durante a trajetória helicoidal do *end-effector*.

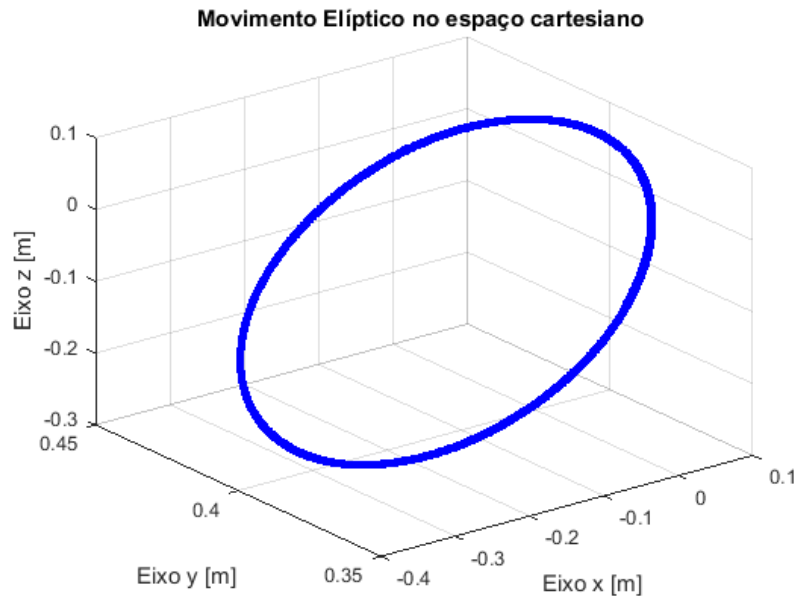
Figura 24 – Torque no espaço das juntas durante a trajetória helicoidal



Fonte: O autor (2022).

A Figura 25 apresenta a trajetória elíptica, que será usada como validação após a etapa de treinamento e testes.

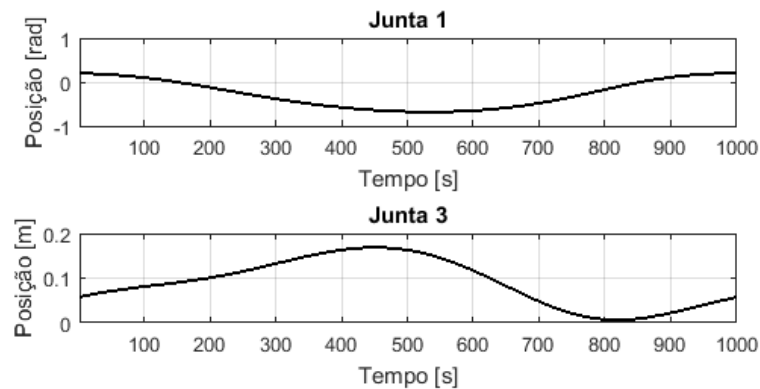
Figura 25 – Trajetória elíptica no espaço cartesiano.



Fonte: O autor (2022).

A Figura 26 mostra as posições no espaço das juntas durante a trajetória elíptica do *end-effector*.

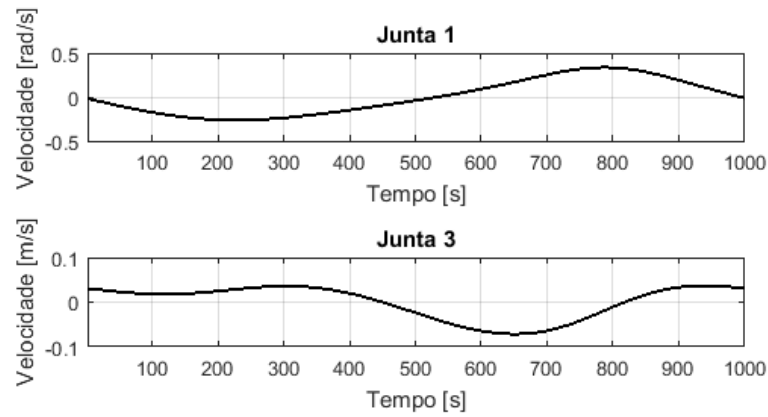
Figura 26 – Posições no espaço das juntas durante a trajetória elíptica.



Fonte: O autor (2022).

A Figura 27 apresenta as velocidades no espaço das juntas durante a trajetória elíptica do *end-effector*.

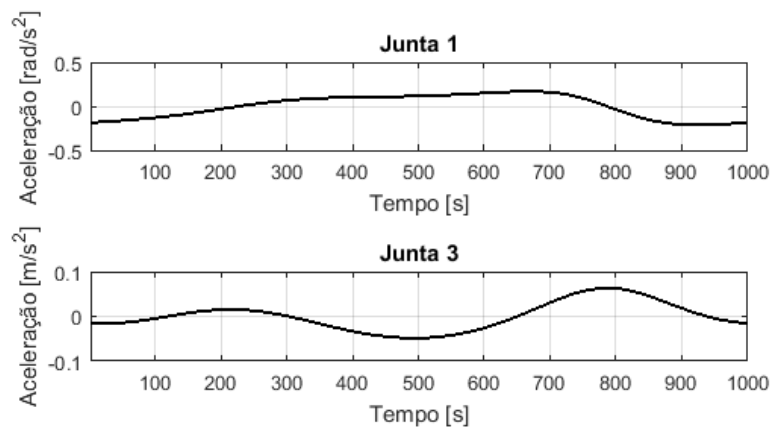
Figura 27 – Velocidades no espaço das juntas durante a trajetória elíptica.



Fonte: O autor (2022).

A Figura 28 mostra as acelerações no espaço das juntas durante a trajetória elíptica do *end-effector*.

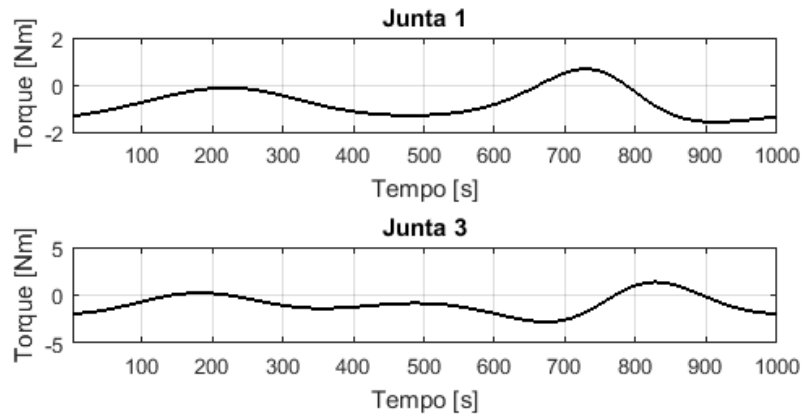
Figura 28 – Acelerações no espaço das juntas durante a trajetória elíptica.



Fonte: O autor (2022).

Os torques no espaço das juntas são apresentados na Figura 29 durante a trajetória elíptica do *end-effector*.

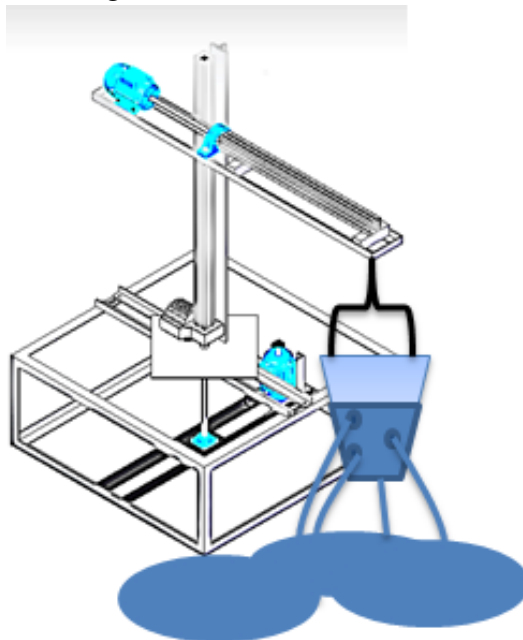
Figura 29 – Torques no espaço das juntas durante a trajetória elíptica.



Fonte: O autor (2022).

Após a etapa de validação das trajetórias impostas aos modelos de carga constante, serão aplicadas a um cenário com carga variável as mesmas atividades de trajetórias apresentadas anteriormente. A Figura 30 ilustra uma imagem baseada na atividade que será aplicada aos modelos após treinados e testados. Será utilizada uma carga de massa inicial de 10,0 Kg, variando de forma decrescente, 2,5 kg a cada 2,5 segundos.

Figura 30 – Manipulador com carga variável



Fonte: O autor (2022).

A seguir, serão apresentados os resultados dos modelos de regressão utilizando os métodos apresentados no capítulo 3. Eles foram aplicados aos modelos do braço robótico considerando, para todos os cenários, 1000 iterações, sendo selecionados os melhores resultados

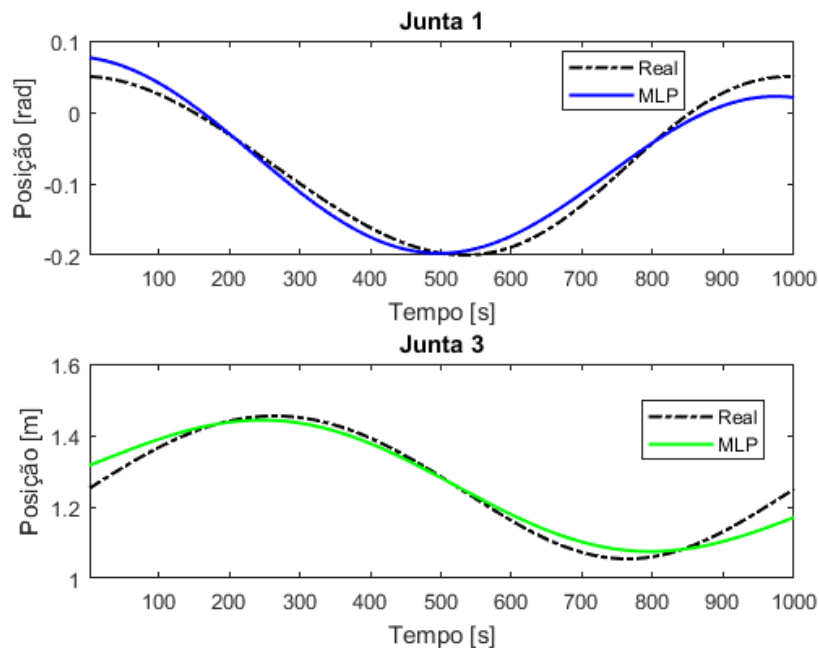
e a média dos resultados, com carga constante e variável.

4.4.1 Rede MLP

O primeiro método a ser aplicado é a rede MLP, apresentada na seção 3.2, onde foram testadas várias arquiteturas de redes, sempre verificando o desempenho. O melhor resultado foi obtido com 30 neurônios na camada oculta e 0,2 como passo de aprendizado. Também foram escolhidos os resultados aceitáveis para o índice RMSE e o coeficiente R_a^2 .

Após as etapas de treinamento e teste com as instâncias embaralhadas e aplicou-se a validação *hold-out* com o mesmo conjunto de forma padrão, conforme a Figura 31. Obteve-se resultados para o índice RMSE de 0,0146 para a junta 1, e 0,0270 para junta 3. E o coeficiente R_a^2 da junta 1 foi de 0,9715, e da junta 3 foi de 0,9617.

Figura 31 – Predição da rede MLP das posições no espaço das juntas durante a trajetória helicoidal.

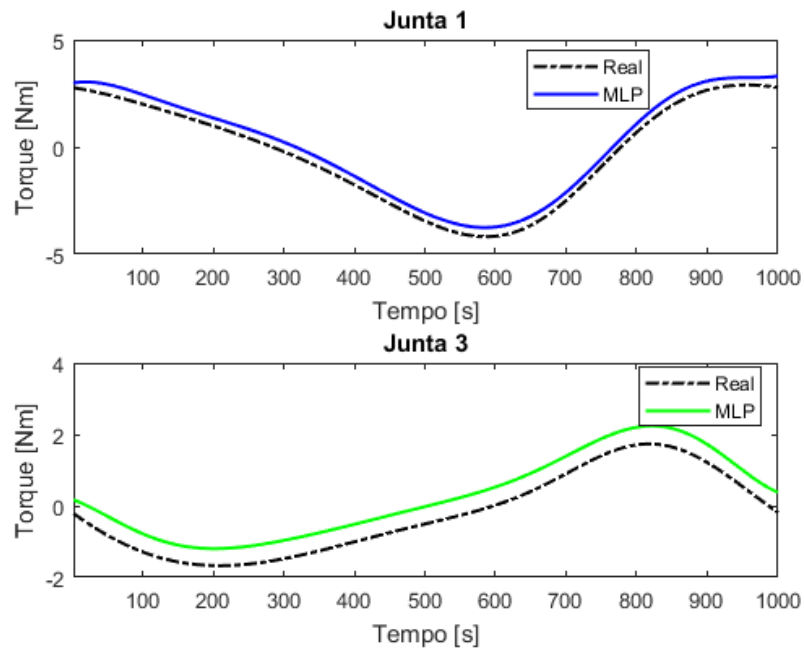


Fonte: O autor (2022).

Com as posições do modelo inteligente da cinemática inversa, calculou-se as derivadas primeira e segunda das posições, obtendo-se as velocidades e acelerações, respectivamente. Essas 3 grandezas serão entradas para uma nova modelagem inteligente da dinâmica inversa que obtém os torques, conforme aborda a Figura 32. Os resultados para o índice RMSE da junta 1 foi de 0,4077, e da junta 3 foi de 0,4165. O coeficiente R_a^2 da junta 1 foi de 0,9705 e da junta 3

foi de 0,8349.

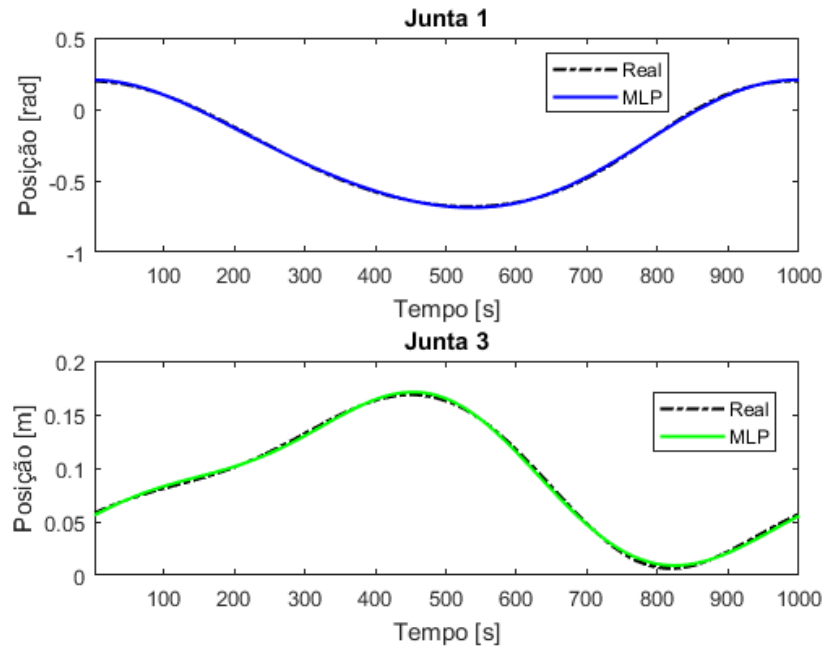
Figura 32 – Predição da rede MLP dos torques da trajetória helicoidal.



Fonte: O autor (2022).

Aplicou-se uma outra validação com o mesmo modelo utilizando a trajetória elíptica como entrada da rede e tendo como saída as posições, conforme mostra a Figura 33. Obteve-se resultados para o RMSE de 0,2430 para a junta 1, e 0,2431 para a junta 3. O coeficiente R_a^2 da junta 1 foi de 0,9996 e da junta 3 foi de 0,9987.

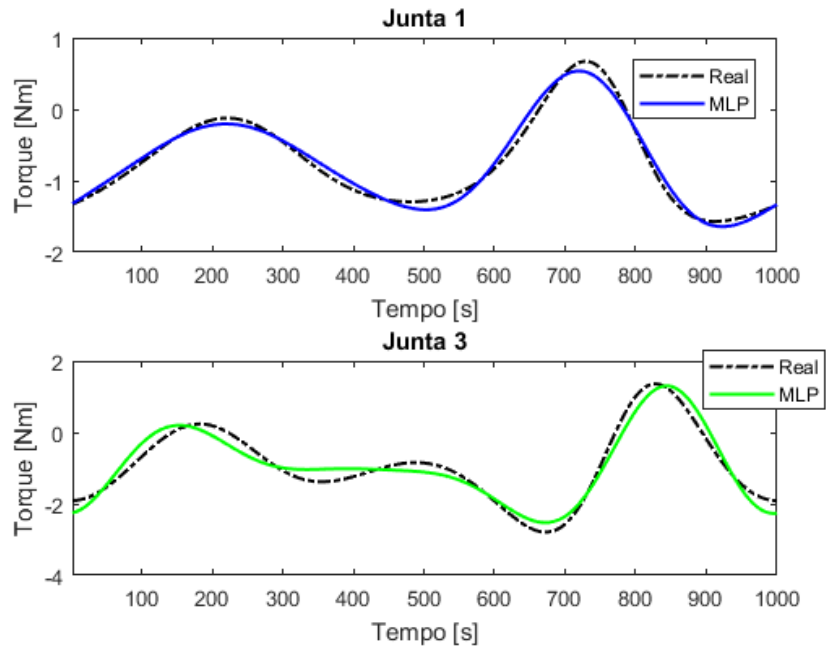
Figura 33 – Predição da rede MLP das posições no espaço das juntas na trajetória elíptica.



Fonte: O autor (2022).

Com os resultados obtidos para as posições do modelo inteligente da cinemática inversa, calculou-se as derivadas primeira e segunda destas, obtendo-se as velocidades e as acelerações, respectivamente. Essas três grandezas foram utilizadas como entradas para a aproximação da dinâmica inversa, obtendo-se os torques, conforme mostra a Figura 34. O RMSE da junta 1 foi de 0,0837 e da junta 3 foi de 0,2431. O coeficiente R_a^2 da junta 1 foi de 0,9804 e da junta 3 de 0,9353.

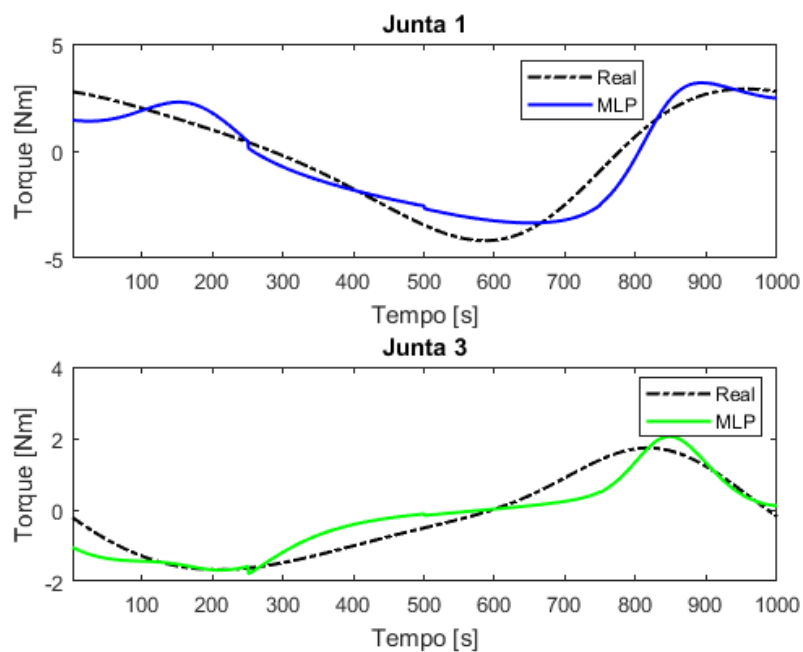
Figura 34 – Predição da rede MLP dos torques da trajetória elíptica.



Fonte: O autor (2022).

A fim de se verificar a eficácia dos modelos após a etapa de treinamento e testes, considerou-se a carga variável e a trajetória helicoidal. A Figura 35 apresenta a melhor situação da dinâmica inversa usando a MLP. Os resultados para o RMSE foram de 0,7373 para a junta 1 e 0,4955 para junta 3. Para o R_a^2 , o da junta 1 foi de 0,8937 e da junta 3 foi de 0,8002.

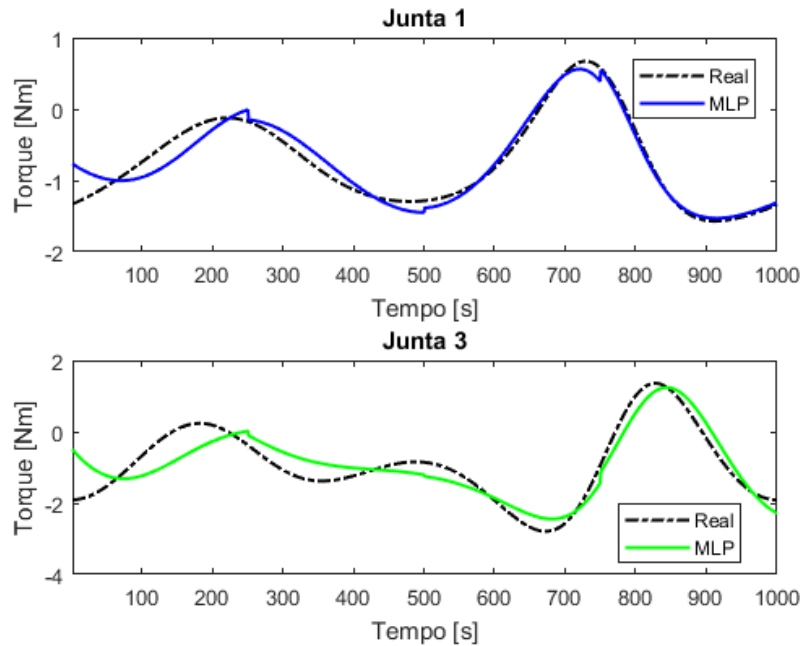
Figura 35 – Predição da rede MLP dos torques da trajetória helicoidal com carga variável.



Fonte: O autor (2022).

A Figura 36 mostra a predição superior dos torques com a MLP para carga variável. O RMSE da junta 1 foi de 0,1415 e da junta 3 foi de 0,4276. O R_a^2 da junta 1 foi de 0,9422, e da junta 3 de 0,7665.

Figura 36 – Predição da rede MLP dos torques da trajetória elíptica com carga variável.

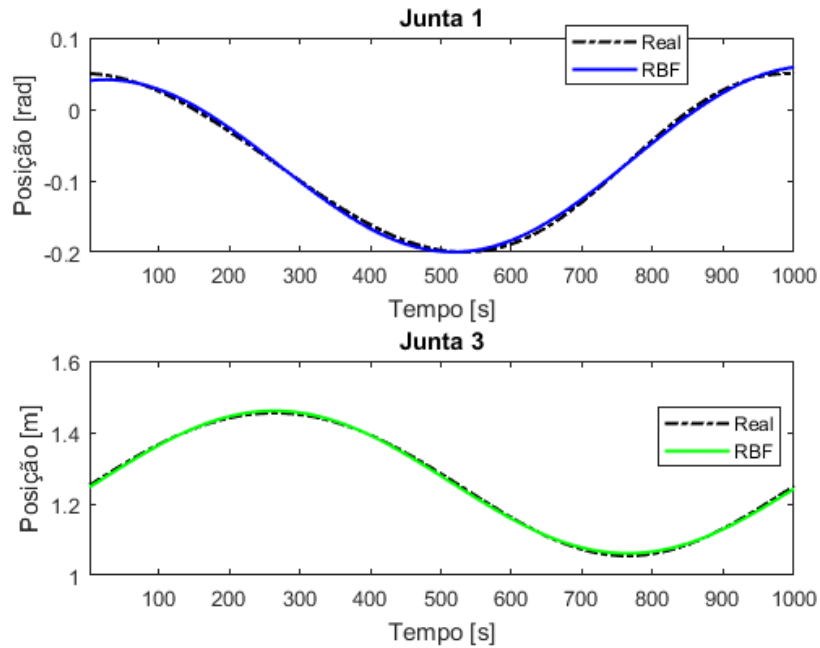


Fonte: O autor (2022).

4.4.2 Rede RBF

A RBF foi apresentada conforme a seção 3.3 do capítulo 3. A rede com desempenho superior teve 10 funções de base radial na camada oculta e largura da função de base radial igual a 0,6480. Também aplicou-se a validação cruzada com as instâncias da base de dados embaralhadas da trajetória helicoidal. Após as etapas de treino/teste, foi apresentada a trajetória com a ordem temporal. O modelo da RBF para estimar a cinemática é apresentado na Figura 37. O RMSE foi de 0,0043 para a junta 1 e 0,0044 para a junta 3. Já quanto ao R_a^2 , o da junta 1 foi de 0,9975 e da junta 3 foi de 0,9990. Também foi escolhido o modelo mais adequado que apresentou resultados aceitáveis.

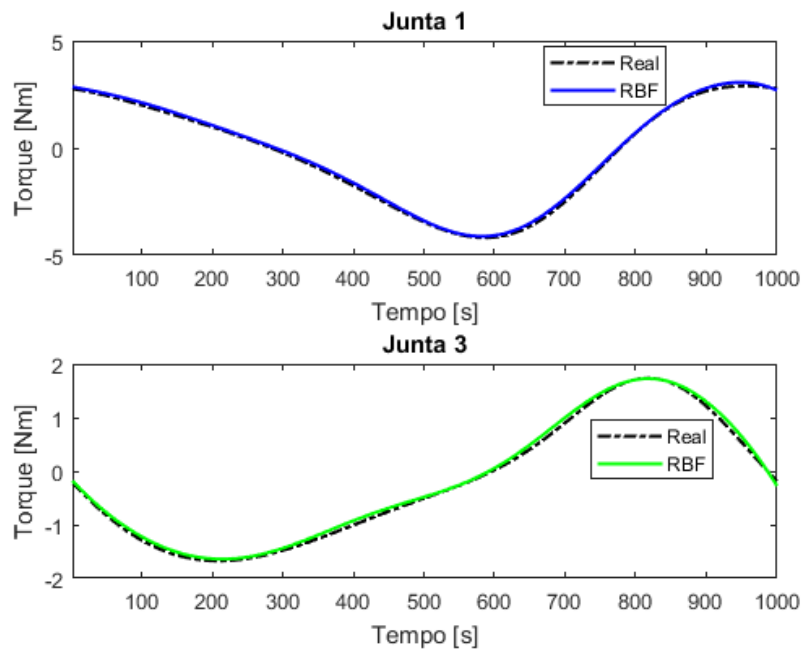
Figura 37 – Predição da rede RBF das posições no espaço das juntas na trajetória helicoidal.



Fonte: O autor (2022).

Aplicando-se o modelo mais adequado na dinâmica inversa com a RBF, conforme a Figura 38, o RMSE da junta 1 foi de 0,1042 e da junta 3 foi de 0,0591. O R_a^2 da junta 1 foi de 0,9980 e da junta 3 de 0,9971.

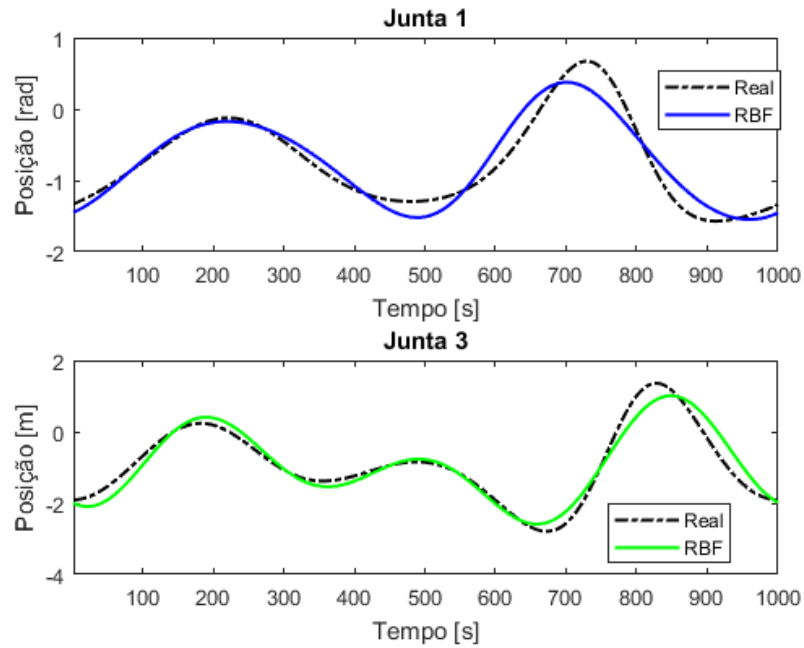
Figura 38 – Predição da rede RBF dos torques da trajetória helicoidal.



Fonte: O autor (2022).

Considerando-se a validação com o movimento elíptico, de acordo com a Figura 39, obteve-se o RMSE de 0,0057 para a junta 1 e 0,0015 da junta 3. Já o R_a^2 da junta 1 foi de 0,9997 e da junta 3 foi de 0,99992.

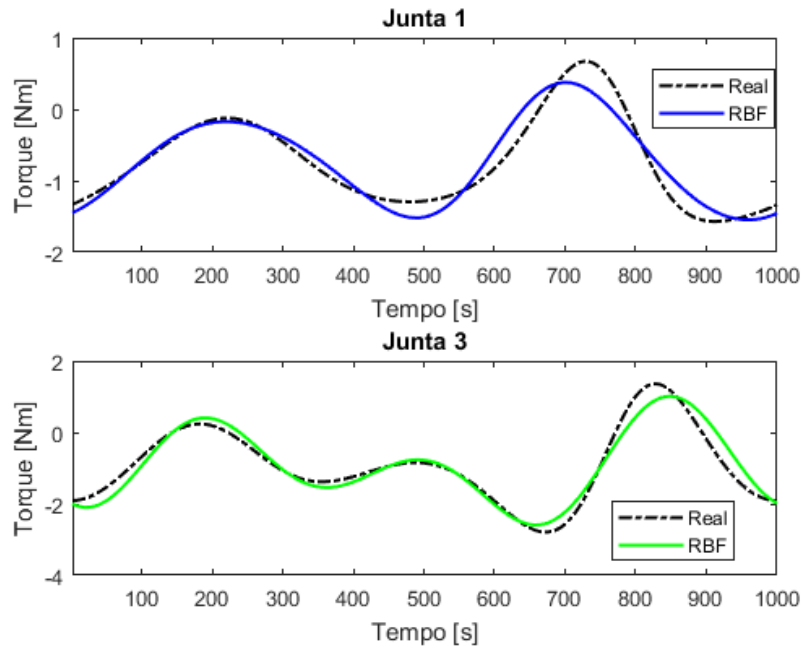
Figura 39 – Predição da rede RBF das posições no espaço das juntas na trajetória elíptica.



Fonte: O autor (2022).

A Figura 40 apresenta o modelo superior para calcular os torques da trajetória elíptica. O RMSE da junta 1 foi de 0,1759, e da junta 3 foi de 0,0015. O Ra^2 da junta 1 foi de 0,9071 e da junta 3 de 0,9264.

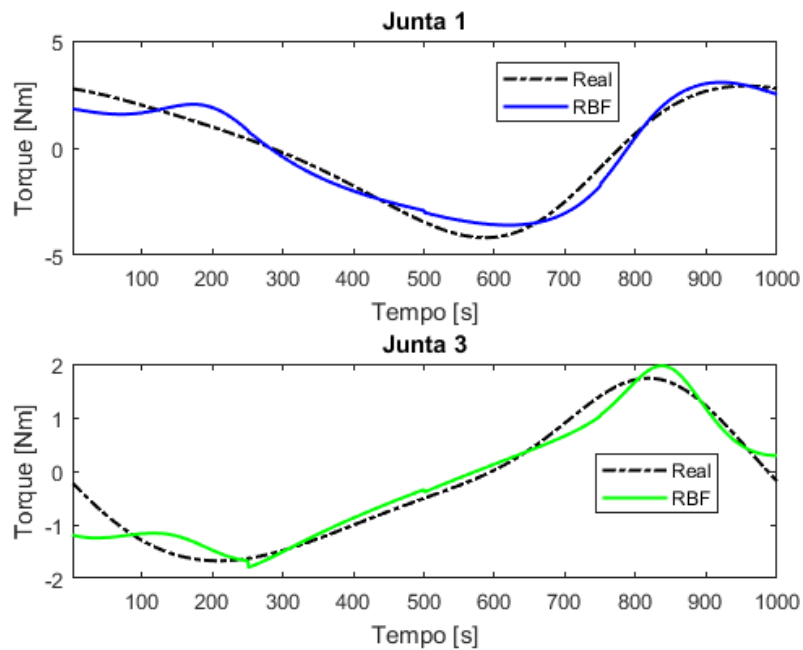
Figura 40 – Predição da rede RBF dos torques da trajetória elíptica.



Fonte: O autor (2022).

A Figura 41 apresenta a predição superior dos torques com a RBF para carga variável com a trajetória elíptica. O RMSE da junta 1 foi de 0,4996 e da junta 3 foi de 0,2428. O R_a^2 da junta 1 foi de 0,9533 e da junta 3 de 0,9492.

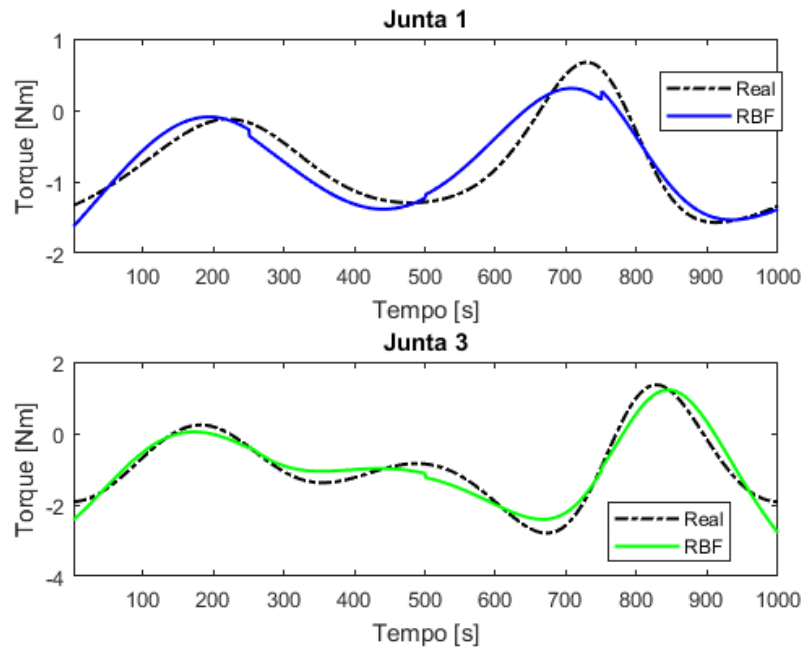
Figura 41 – Predição da rede RBF dos torques da trajetória Helicoidal com carga variável.



Fonte: O autor (2022).

A Figura 42 apresenta a predição mais favorável dos torques com a RBF para carga variável com a trajetória helicoidal. O RMSE da junta 1 foi de 0,2106 e da junta 3 foi de 0,2686. O R_a^2 da junta 1 foi de 0,8653 e da junta 3 de 0,9195.

Figura 42 – Predição da rede RBF dos torques da trajetória elíptica com carga variável.

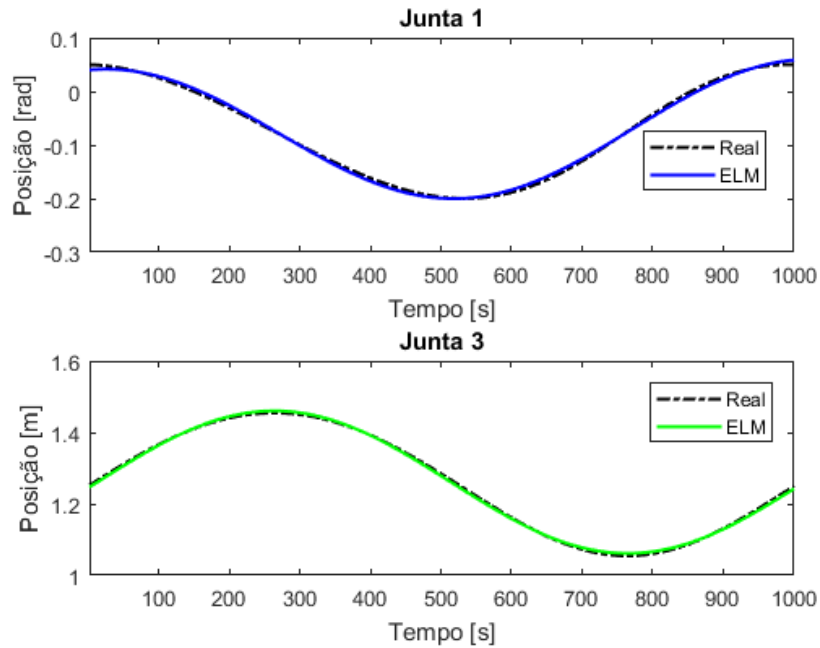


Fonte: O autor (2022).

4.4.3 Rede ELM

O mesmo procedimento para as redes anteriores foi utilizado para a ELM, porém considerando a quantidade de neurônios igual a 10. Desta forma, obteve-se o melhor modelo da ELM para estimar a cinemática, como apresentado na Figura 43, com a validação da trajetória helicoidal. O RMSE foi de 0,0043 para a junta 1 e 0,0044 para a junta 3. Já o R_a^2 da junta 1 foi de 0,9975 e da junta 3 foi de 0,9990.

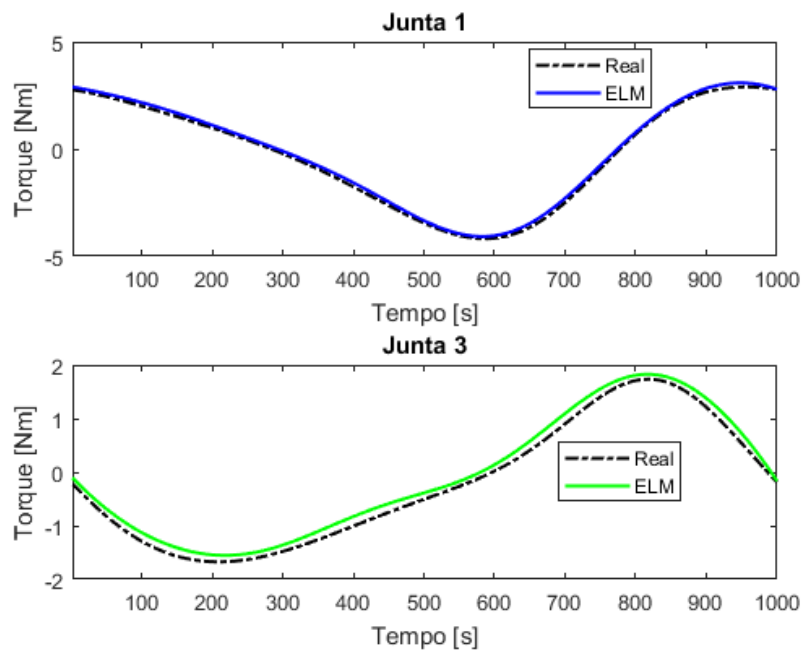
Figura 43 – Predição da rede ELM das posições no espaço das juntas na trajetória helicoidal.



Fonte: O autor (2022).

A Figura 44 mostra os resultados dos torques da ELM da trajetória helicoidal. O RMSE da junta 1 foi de 0,1474 e da junta 3 foi de 0,1435. O R_a^2 da junta 1 foi de 0,9961 e junta 3 de 0,9833.

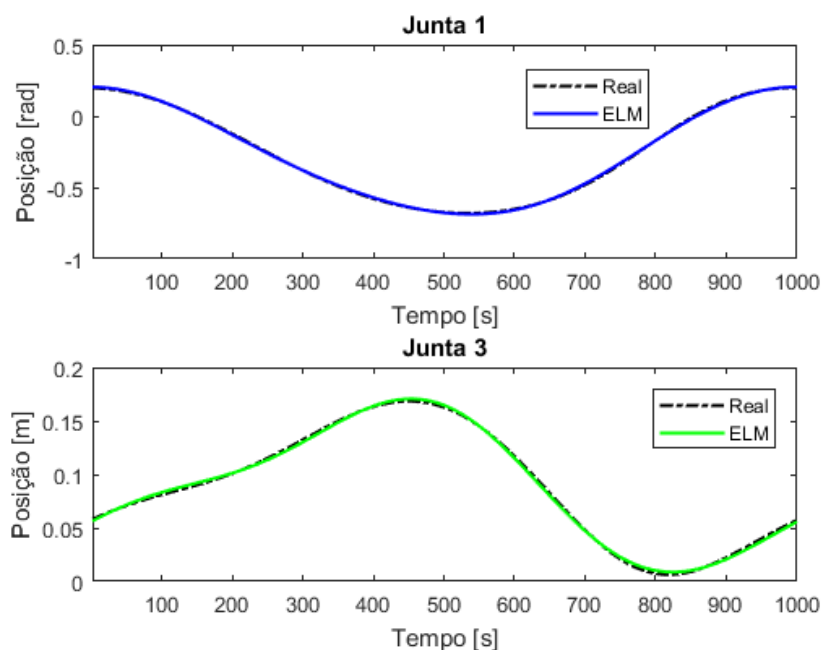
Figura 44 – Predição da rede ELM dos torques da trajetória helicoidal.



Fonte: O autor (2022).

Na Figura 45, mostra-se a validação com a trajetória elíptica em que se obteve o RMSE de 0,0056 para a junta 1 e 0,0017 para a junta 3. Já o R_a^2 da junta 1 foi de 0,9997 e da junta 3 foi de 0,9989.

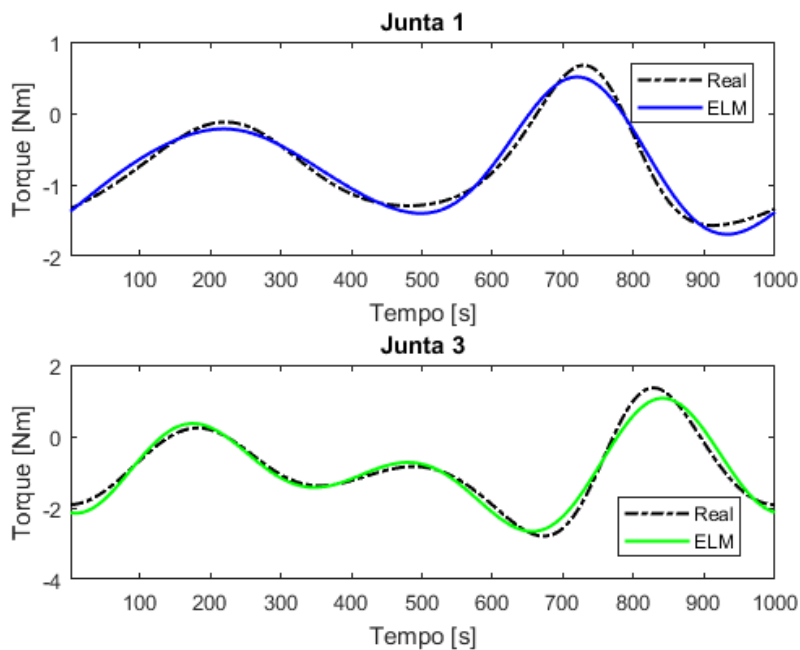
Figura 45 – Predição da rede ELM das posições no espaço das juntas na trajetória elíptica.



Fonte: O autor (2022).

A Figura 46 apresenta o modelo mais adequado para calcular os torques da trajetória elíptica com a ELM. O RMSE da junta 1 foi de 0,1001 e da junta 3 foi de 0,1845. O Ra^2 da junta 1 foi de 0,9722 e da junta 3 de 0,9639.

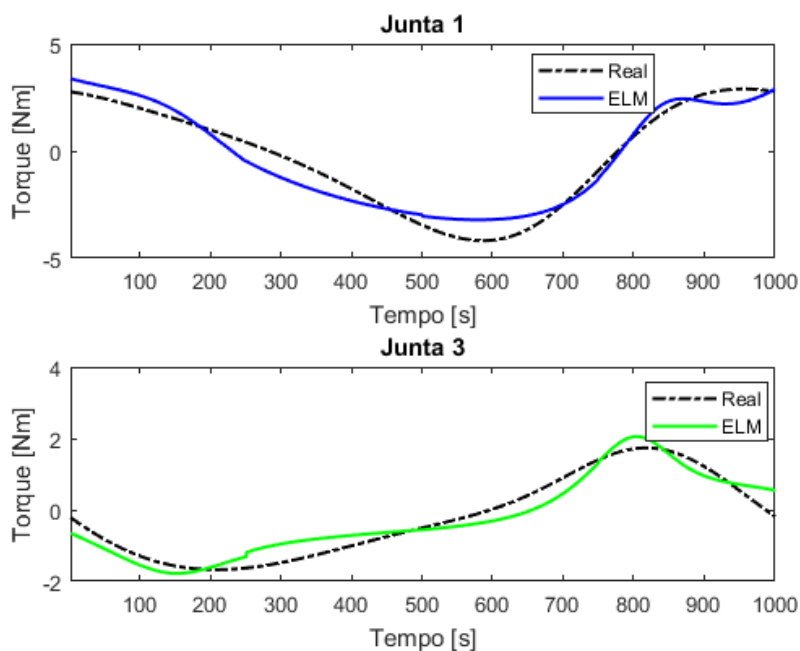
Figura 46 – Predição da rede ELM dos torques da trajetória elíptica.



Fonte: O autor (2022).

A Figura 47 apresenta a melhor predição dos torques com a rede ELM para carga variável com a trajetória helicoidal. O RMSE da junta 1 foi de 0,5915 e da junta 3 foi de 0,3135. O Ra^2 a da junta 1 foi de 0,9332 e da junta 3 de 0,9133.

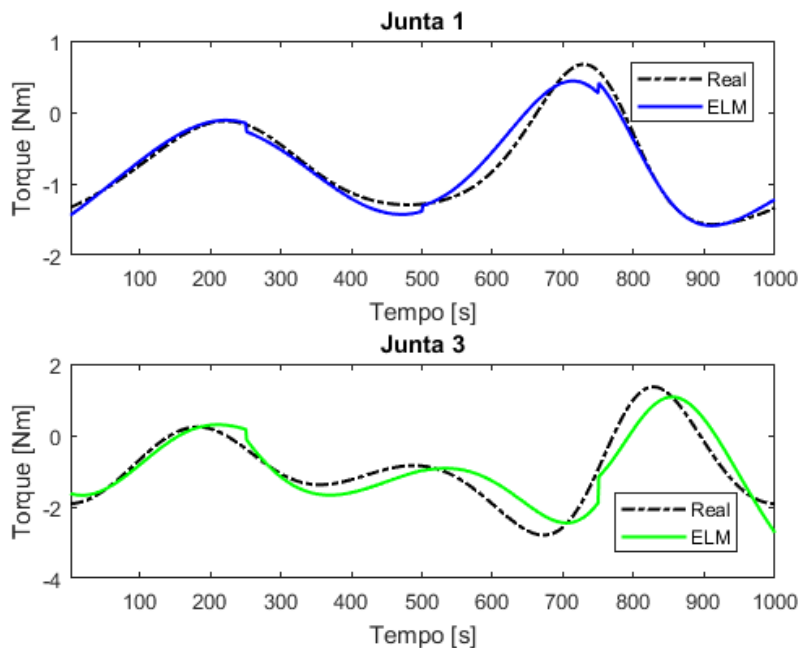
Figura 47 – Predição da rede ELM dos torques da trajetória helicoidal com carga variável.



Fonte: O autor (2022).

A Figura 48 mostra a predição superior dos torques com a rede ELM para carga variável com a trajetória elíptica. O RMSE da junta 1 foi de 0,1178 e da junta 3 foi de 0,4056. O R_a^2 da junta 1 foi de 0,9602 e junta 3 de 0,8099.

Figura 48 – Predição da rede ELM dos torques da trajetória elíptica com carga variável.

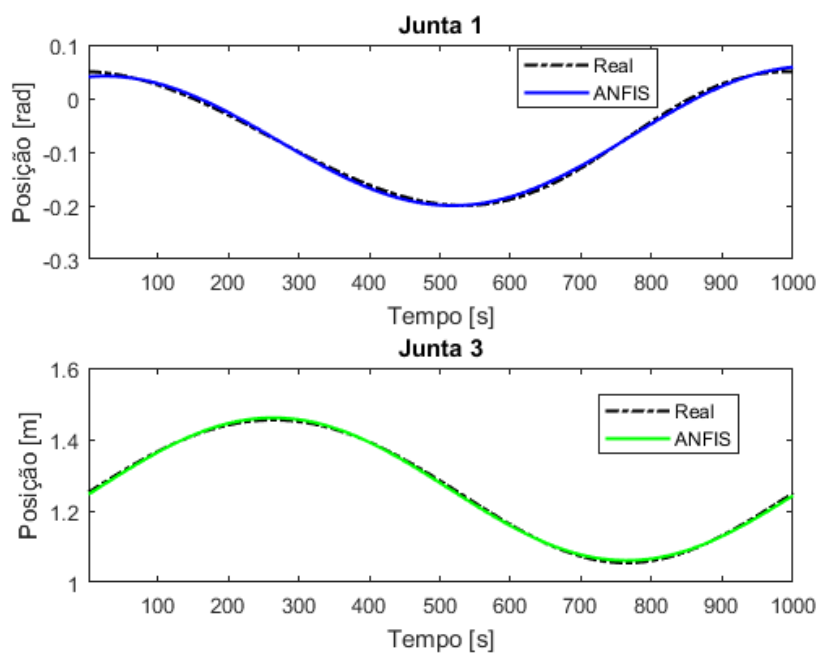


Fonte: O autor (2022).

4.4.4 ANFIS

No caso da rede ANFIS utilizou-se uma configuração com 15 neurônios na camada escondida. O modelo da ANFIS que foi mais adequado para estimar a cinemática é apresentado na Figura 49 com a validação da trajetória helicoidal. O RMSE foi de 0,0072 para a junta 1 e 0,0079 para a junta 3. Já o R_a^2 da junta 1 foi de 0,9891 e da junta 3 foi de 0,9920.

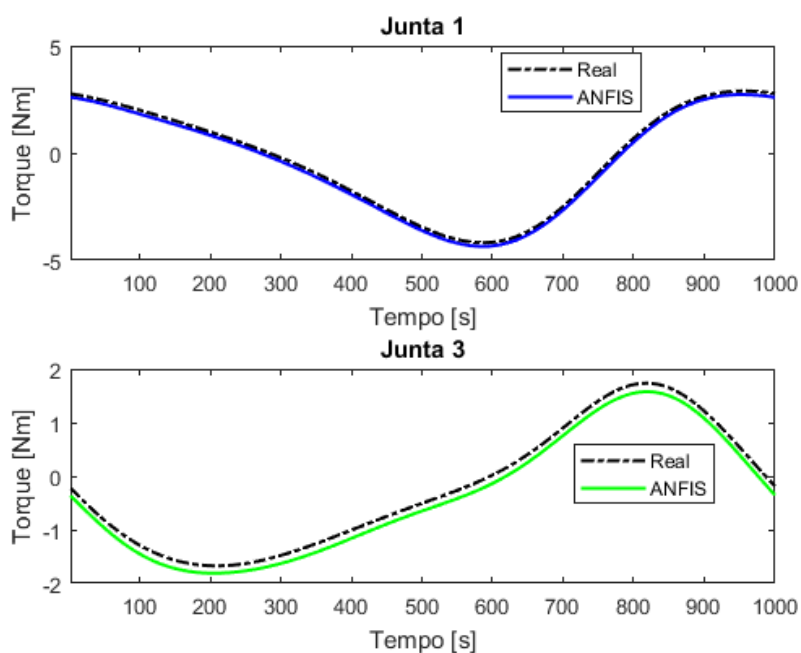
Figura 49 – Predição do ANFIS das posições no espaço das juntas na trajetória helicoidal.



Fonte: O autor (2022).

A Figura 50 mostra os resultados dos torques da ANFIS do movimento helicoidal. O RMSE da junta 1 foi de 0,1693 e da junta 3 foi de 0,1495. O R_a^2 da junta 1 foi de 0,9949 e da junta 3 de 0,9818.

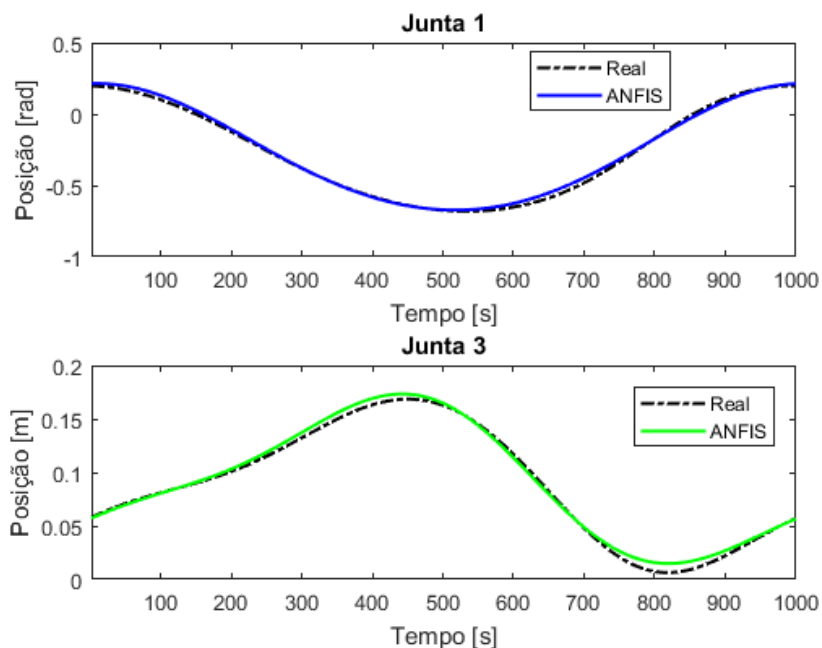
Figura 50 – Predição do ANFIS dos torques da trajetória helicoidal.



Fonte: O autor (2022).

A Figura 51 apresenta a validação do movimento elíptico da rede ANFIS. O RMSE foi de 0,0194 para a junta 1 e 0,0041 para a junta 3. Já o R_a^2 da junta 1 foi de 0,9961 e da junta 3 foi de 0,9935.

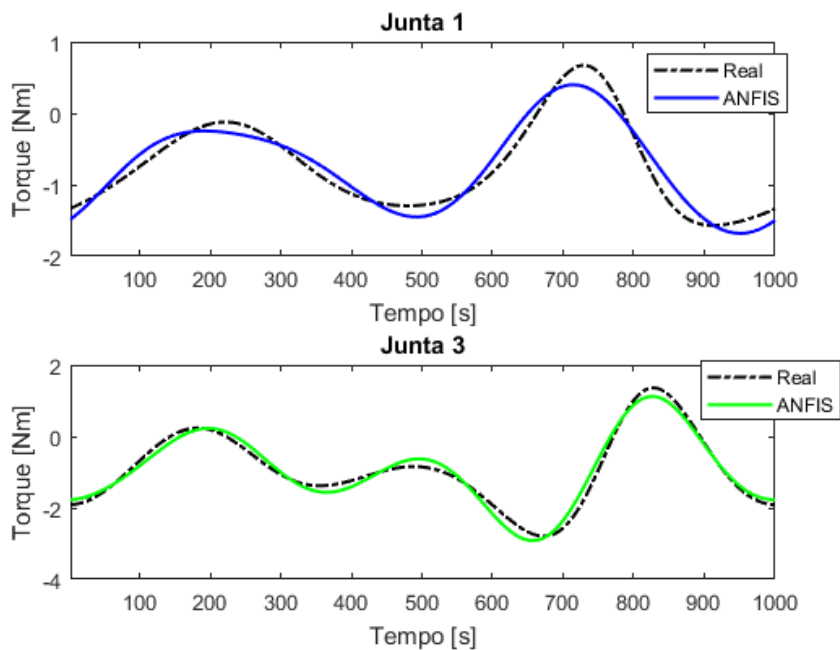
Figura 51 – Predição do ANFIS das posições no espaço das juntas na trajetória elíptica.



Fonte: O autor (2022).

A Figura 52 mostra os resultados dos torques da ANFIS da trajetória elíptica. O RMSE da junta 1 foi de 0,1512 e da junta 3 foi de 0,1641. O R_a^2 da junta 1 foi de 0,9333 e da junta 3 de 0,9713.

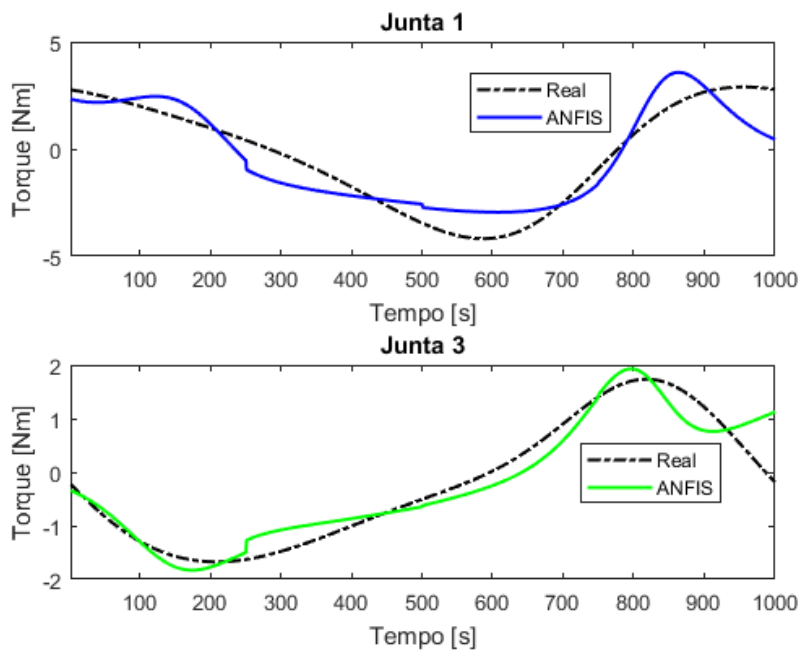
Figura 52 – Predição do ANFIS dos torques da trajetória elíptica.



Fonte: O autor (2022).

A Figura 53 apresenta a predição superior dos torques com a ANFIS para carga variável com a trajetória helicoidal. O RMSE da junta 1 foi de 0,9215 e da junta 3 foi de 0,3071. O R_a^2 da junta 1 foi de 0,8203 e da junta 3 de 0,9163.

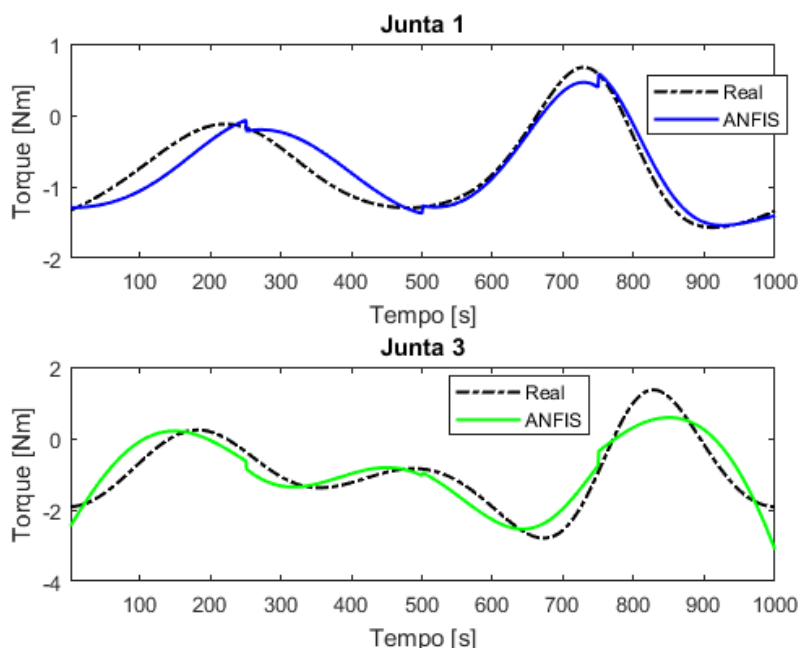
Figura 53 – Predição do ANFIS dos torques da trajetória helicoidal com carga variável.



Fonte: O autor (2022).

A Figura 54 demonstra a predição superior dos torques com a ANFIS para carga variável com a trajetória elíptica. O RMSE da junta 1 foi de 0,1907 e da junta 3 foi de 0,4024. O R_a^2 da junta 1 foi de 0,8909 e da junta 3 de 0,8001.

Figura 54 – Predição do ANFIS dos torques da trajetória elíptica com carga variável.

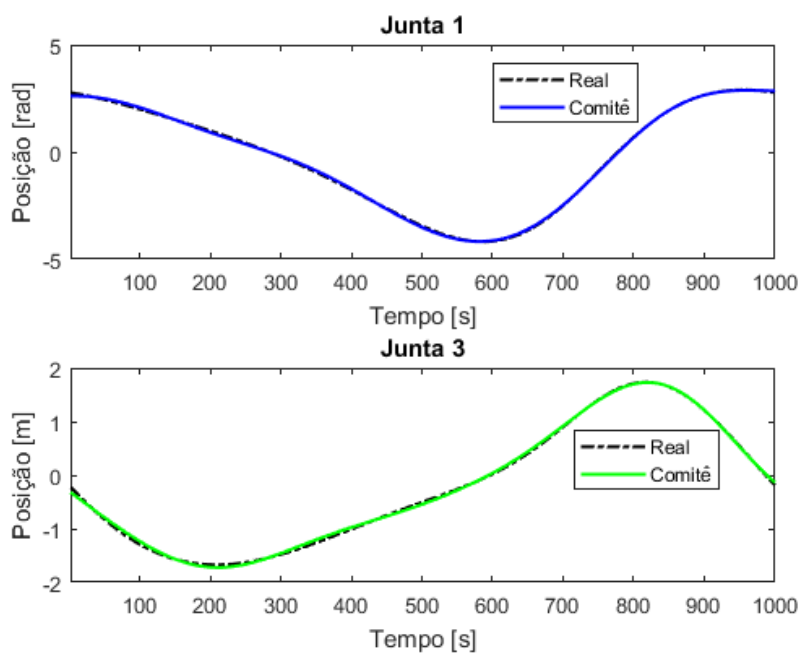


Fonte: O autor (2022).

4.4.5 Método proposto

Para o caso do comitê de SVRs, utilizou-se 3 SVRs com configurações diferentes, em especial os *kernels*. Considerando a configuração mais adequada do comitê, utilizou-se para a primeira SVR ϵ de 0,1, tolerância 0,2 e *kernel* gaussiano. Para a segunda SVR ϵ de 0,2, tolerância 0,1 e *kernel* sigmoidal. Para a terceira ϵ de 0,5, tolerância 0,1 e *kernel* polinomial de 7º ordem. A Figura 55 apresenta a estimativa da trajetória helicoidal com o comitê de SVRs. O RMSE da junta 1 foi de 0,0029 e da junta 3 foi de 0,0041. O R_a^2 da junta 1 foi de 0,9989 e da junta 3 foi de 0,9992.

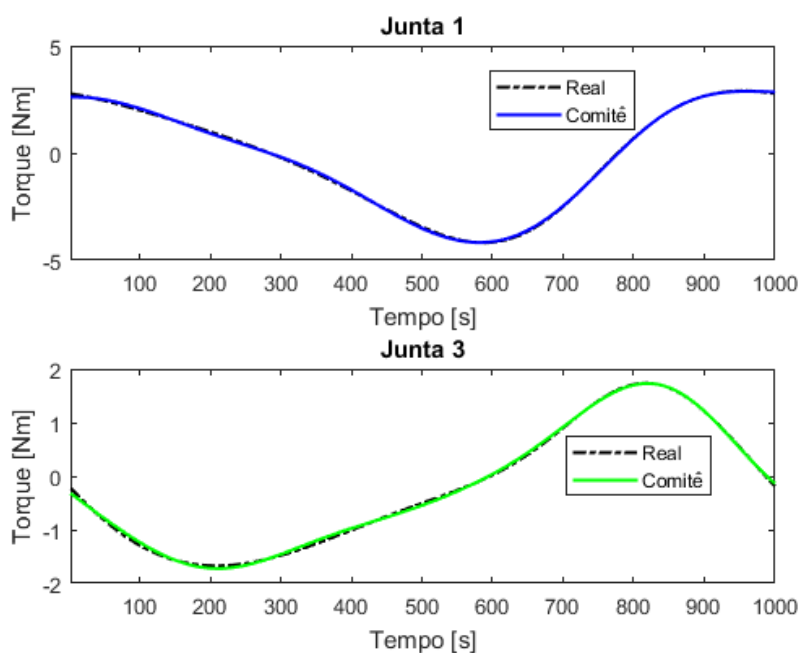
Figura 55 – Predição do comitê nas posições no espaço das juntas na trajetória helicoidal.



Fonte: O autor (2022).

A Figura 56 apresenta a estimativa dos torques para a trajetória helicoidal usando o comitê de SVRs. O RMSE da junta 1 foi de 0,0443 e da junta 3 foi de 0,0303. O R_a^2 da junta 1 foi de 0,9997 e da junta 3 foi de 0,9993.

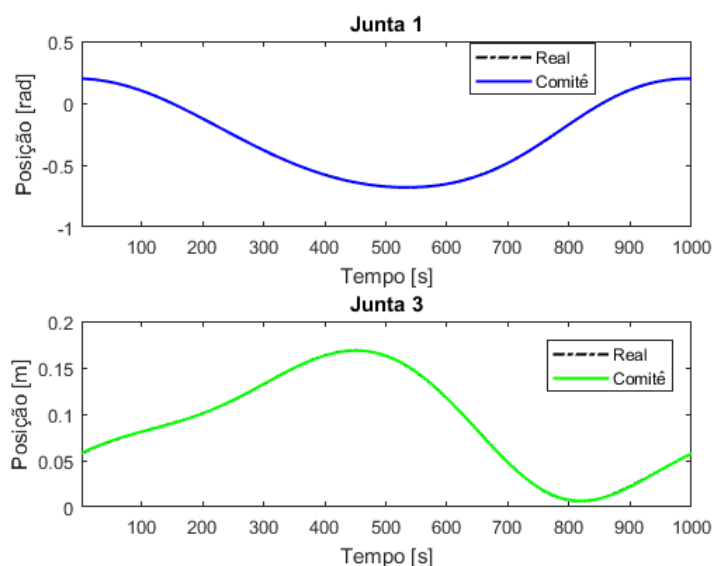
Figura 56 – Predição do comitê nos torques da trajetória helicoidal.



Fonte: O autor (2022).

A Figura 57 demonstra a estimativa da trajetória elíptica com o comitê de SVRs. O RMSE da junta 1 foi de 0,0006 e da junta 3 foi de 0,0001. O R_a^2 da junta 1 foi de 0,9999 e da junta 3 foi de 0,9999.

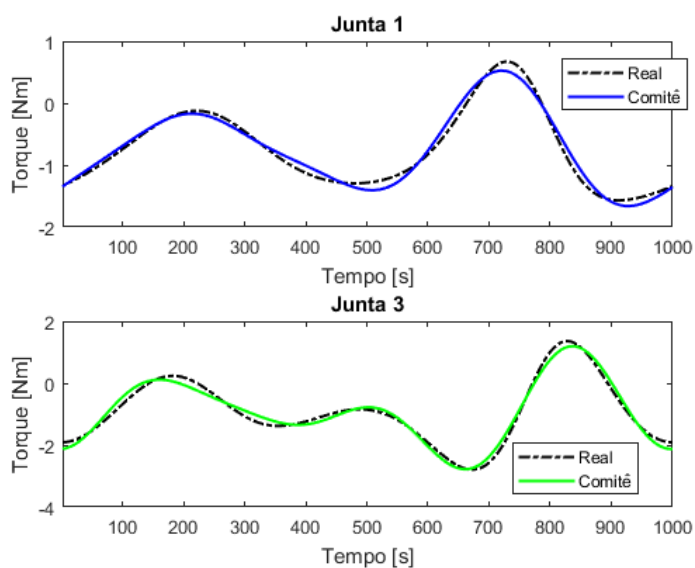
Figura 57 – Predição do comitê nas posições no espaço das juntas na trajetória elíptica com carga constante.



Fonte: O autor (2022).

A Figura 58 apresenta a estimativa dos torques com o movimento helicoidal usando o comitê de SVRs. O RMSE da junta 1 foi de 0,0883 e da junta 3 foi de 0,1462. O R_a^2 da junta 1 foi de 0,9782 e da junta 3 foi de 0,9772.

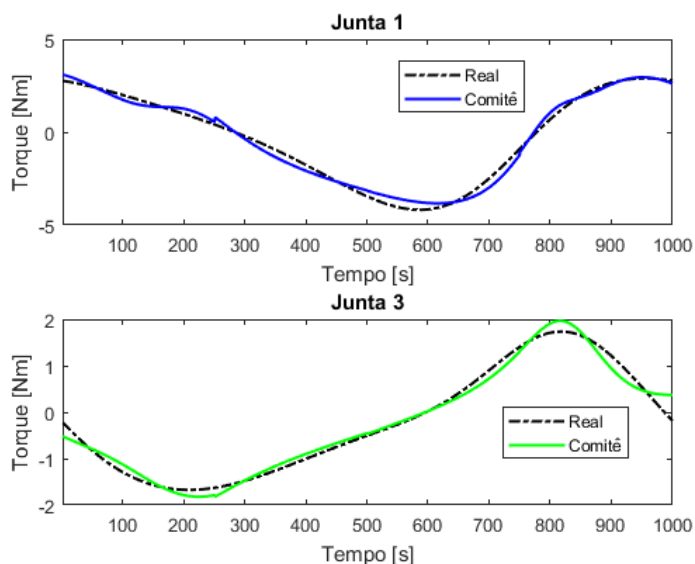
Figura 58 – Predição do comitê nos torques da trajetória elíptica com carga constante.



Fonte: O autor (2022).

A Figura 59 mostra a predição mais adequada dos torques com comitê de SVRs para carga variável e com a trajetória helicoidal. O RMSE da junta 1 foi de 0,2675 e da junta 3 foi de 0,1399. O R_2^2 da junta 1 foi de 0,9871 e da junta 3 de 0,9838.

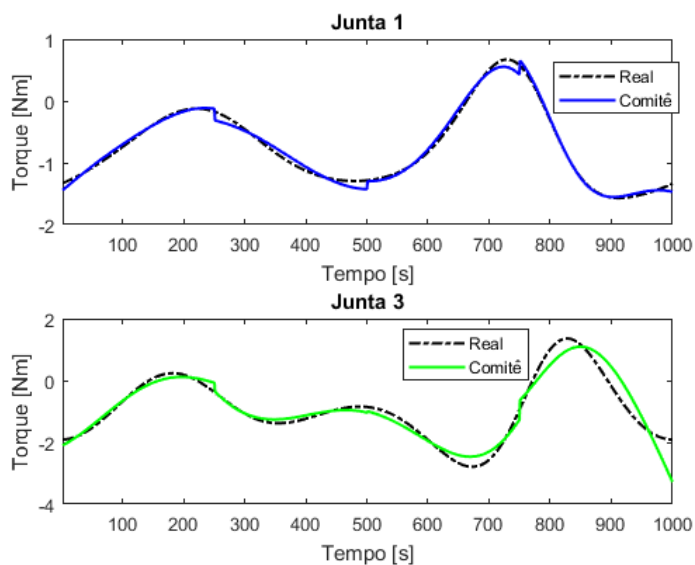
Figura 59 – Predição do comitê nos torques da trajetória helicoidal com carga variável.



Fonte: O autor (2022).

A Figura 60 apresenta a predição superior dos torques com comitê de SVRs para carga variável e com a trajetória elíptica. O RMSE da junta 1 foi de 0,0561 e da junta 3 foi de 0,2717. O R_2^2 da junta 1 foi de 0,9913 e da junta 3 foi de 0,9203.

Figura 60 – Predição do ANFIS dos torques da trajetória elíptica com carga variável.



Fonte: O autor (2022).

4.4.6 Comparação entre os métodos com carga constante

A Tabela 4 mostra os resultados médios do R_a^2 ajustado, RMSE e desvio padrão do erro dos modelos para cada junta durante a validação do manipulador robótico. Nesse cenário, usando-se uma trajetória helicoidal completa, a máquina de comitê obteve um desempenho mais adequado em relação aos demais métodos.

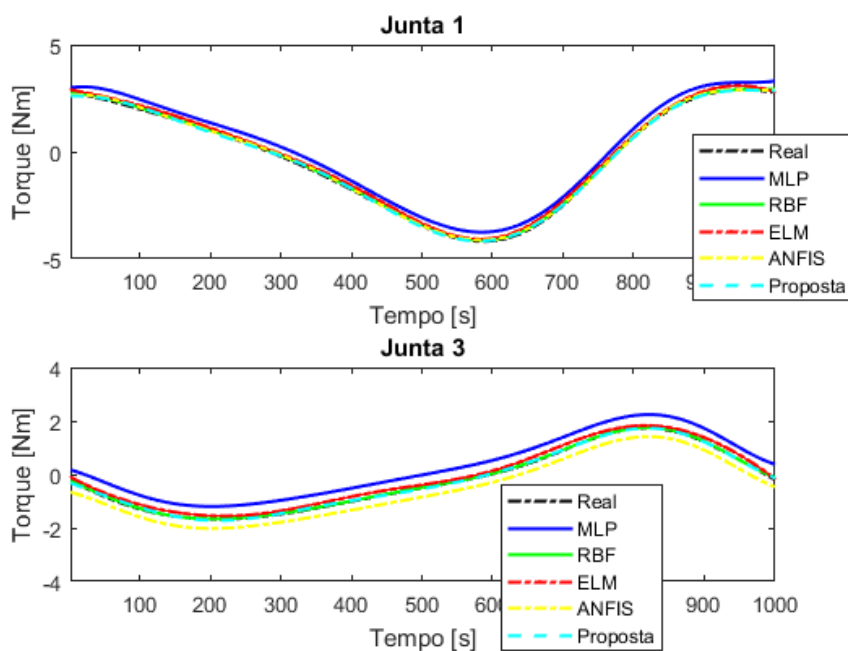
Tabela 4 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque de todos os métodos para trajetória helicoidal com carga constante.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,9215	0,8012	0,4321	0,4322	0,8188	0,8091
<i>RBF</i>	0,9408	0,9390	0,2071	0,0798	0,7992	0,7911
<i>ELM</i>	0,9511	0,9502	0,2087	0,1995	0,8090	0,8197
<i>ANFIS</i>	0,9320	0,9241	0,1928	0,1998	0,8182	0,8170
Comitê proposto	0,9721	0,9719	0,0638	0,0435	0,0687	0,0598

Fonte: O autor (2022).

A Figura 61 apresenta a predição com todos os métodos com carga constante baseado na trajetória helicoidal.

Figura 61 – Predição dos torques usando os modelos na trajetória helicoidal com carga constante.



Fonte: O autor (2022).

A Tabela 5 mostra os resultados médios dos coeficientes R_a^2 ajustados, RMSE e o

desvio padrão do erro dos modelos para cada junta durante a validação do manipulador robótico. Também considerou-se um cenário com trajetória elíptica completa. O método baseado em comitê obteve melhor desempenho quando comparado aos demais métodos.

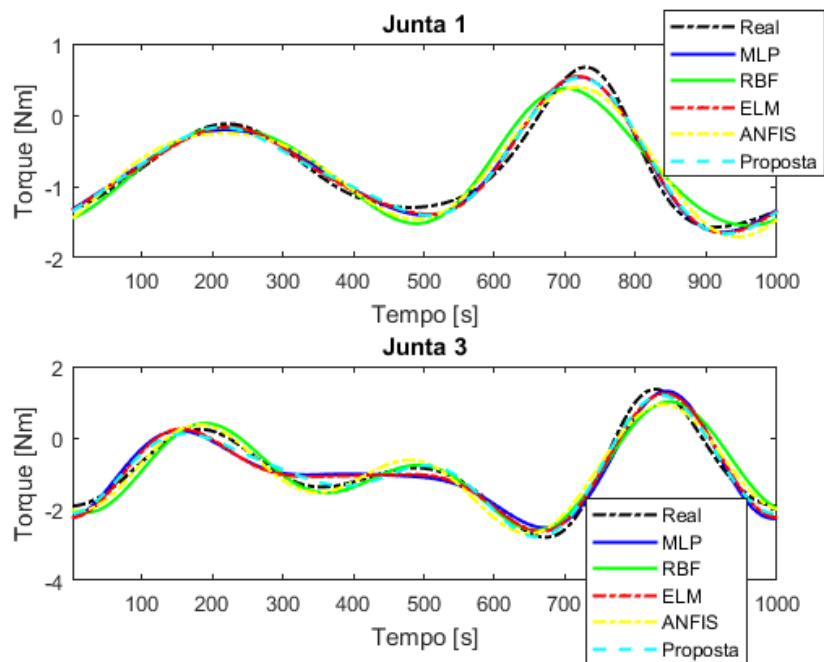
Tabela 5 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico do torque de todos os métodos com carga constante.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,9152	0,9120	0,2110	0,2925	0,7060	0,7045
<i>RBF</i>	0,8711	0,9083	0,2480	0,0027	0,6833	0,6857
<i>ELM</i>	0,9210	0,9180	0,2510	0,2629	0,7181	0,7125
<i>ANFIS</i>	0,9012	0,9210	0,2367	0,2398	0,6971	0,6892
Comitê proposto	0,9411	0,9403	0,0950	0,1824	0,0832	0,0829

Fonte: O autor (2022).

A Figura 62 demonstra a predição com todos os métodos com carga constante baseado na trajetória elíptica.

Figura 62 – Predição dos torques usando os modelos na trajetória elíptica com carga constante.



Fonte: O autor (2022).

4.4.7 Comparação entre os métodos com carga variável

A tabela 7 mostra os resultados médios dos coeficientes R_a^2 ajustados, RMSE e desvio padrão do erro dos modelos para cada junta durante a validação do manipulador robótico.

Nessa cenário com validação usando uma trajetória helicoidal completa com carga variável, a máquina de comitê obteve o desempenho superior em relação aos demais métodos.

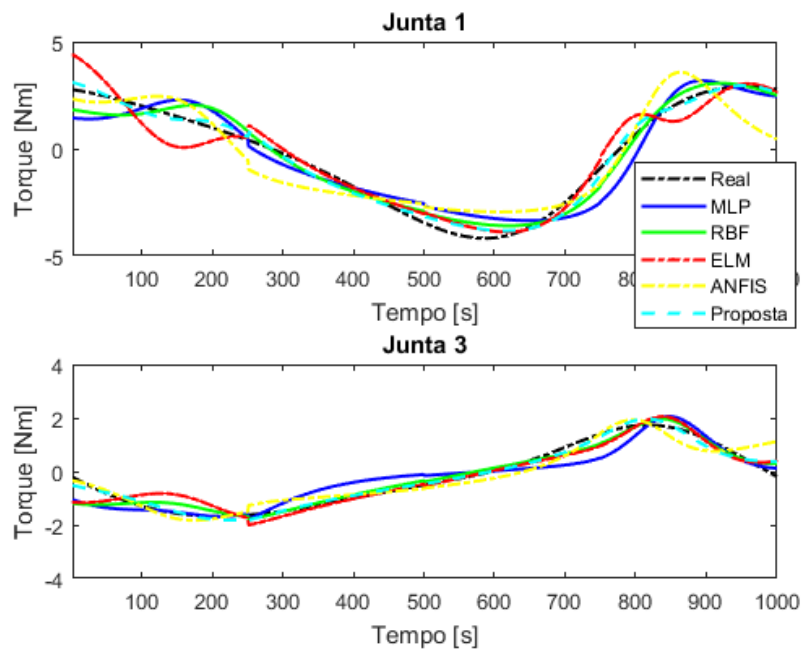
Tabela 6 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento helicoidal com carga variável.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,8612	0,7821	0,7612	0,5178	0,8290	0,8245
<i>RBF</i>	0,9256	0,9188	0,5328	0,2930	0,7891	0,7894
<i>ELM</i>	0,9088	0,8681	0,6428	0,3629	0,8398	0,8289
<i>ANFIS</i>	0,9078	0,9181	0,2045	0,1829	0,7587	0,7486
Comitê proposto	0,9475	0,9411	0,2032	0,1824	0,0857	0,0861

Fonte: O autor (2022).

A Figura 63 apresenta a predição com todos os métodos com carga variável baseado na trajetória elíptica.

Figura 63 – Predição dos torques usando os modelos na trajetória helicoidal com carga variável.



Fonte: O autor (2022).

A tabela 7 mostra os resultados médios dos coeficientes R_a^2 ajustados, RMSE e desvio padrão do erro dos modelos para cada junta durante a validação do manipulador robótico. Neste caso, o cenário de validação é uma trajetória elíptica completa com carga variável. O método baseado em comitê obteve um desempenho superior em relação aos demais métodos.

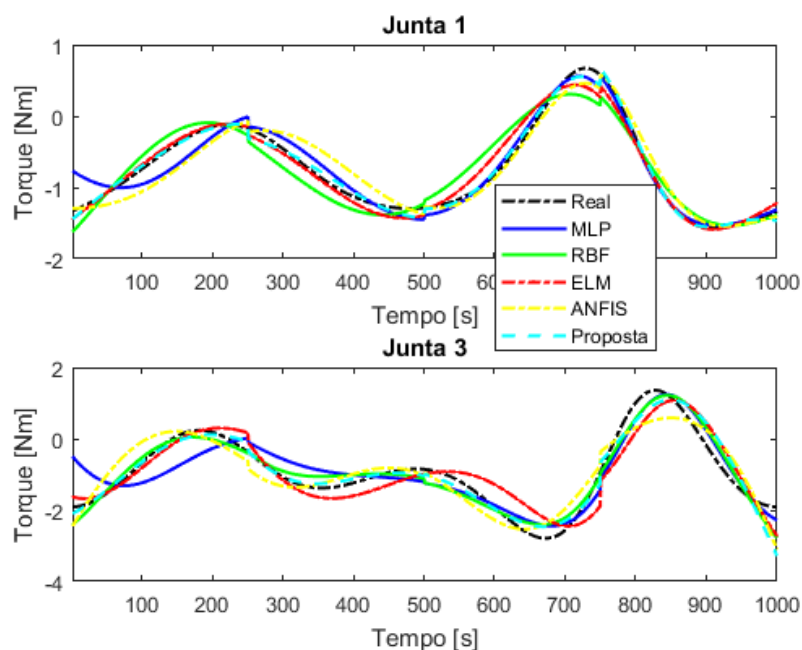
Tabela 7 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento elíptico com carga variável.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,9223	0,7276	0,2110	0,4721	0,8191	0,8201
<i>RBF</i>	0,8290	0,8789	0,2711	0,2986	0,7961	0,7956
<i>ELM</i>	0,8389	0,7887	0,1927	0,4528	0,8289	0,8288
<i>ANFIS</i>	0,8632	0,7790	0,2720	0,4398	0,7698	0,7611
Comitê proposto	0,9389	0,8856	0,0720	0,2902	0,0862	0,0865

Fonte: O autor (2022).

A Figura 64 apresenta a predição com todos os métodos com com variável baseado na trajetória elíptica.

Figura 64 – Predição dos torques usando os modelos na trajetória helicoidal .



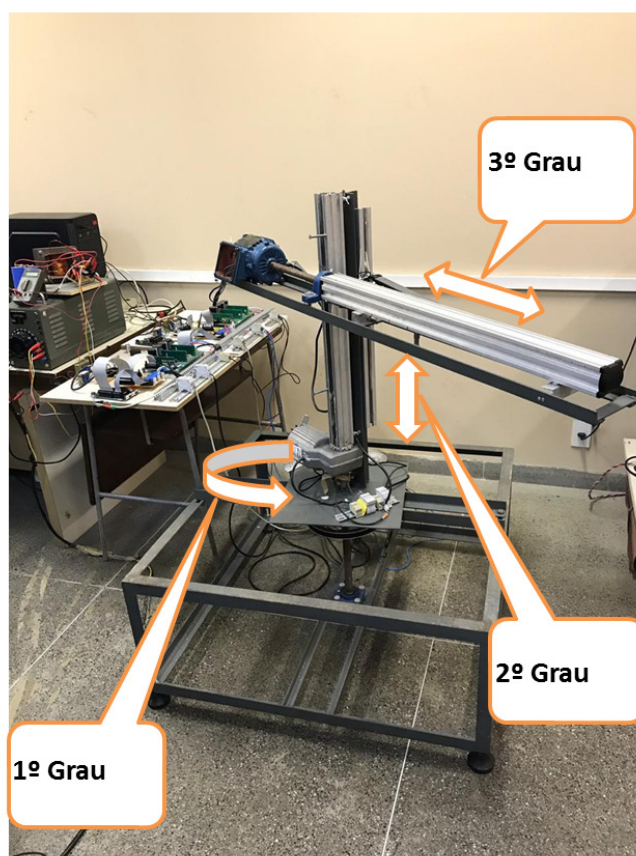
Fonte: O autor (2022).

É importante observar que o desvio padrão do comitê proposto tende a ser mais baixo que os demais métodos, isso quer dizer que o mesmo obteve resultados mais padronizados, e isso é muito relevante para essa aplicação. Alguns outros testes que não tiveram muito sucesso devido ao custo computacional foram desenvolvidos e encontra-se nos Apêndices A e B, estes testes foram os comitês com todos os métodos em comparação com o proposto tanto com carga constante quanto com carga variável.

4.5 Resultados com dados experimentais

Nesta seção, tem-se resultados com dados experimentais. Para a realização desta etapa foi necessário a implementação de uma estrutura de um manipulador robótico acionada com 3 motores de indução trifásicos do tipo gaiola de esquilo com 0,5 cv de potência nominal, tensão nominal de 380/220 V, 4 polos e 1,18 A de corrente nominal ligado em delta, que se encontra no GPAR. Este manipulador irá comportar 3 graus de liberdade, 2 graus do *end-effector* que serão acionados com motores CC e mais 3 graus que são: o da base, o do tronco e do anti-braço, todos acionados com motores de indução trifásicos. Como a bancada ainda está em desenvolvimento, apenas a primeira junta e a terceira estão em funcionamento. A Figura 65 apresenta o manipulador robótico do GPAR.

Figura 65 – Manipulador robótico cilíndrico do GPAR.



Fonte: O autor (2022).

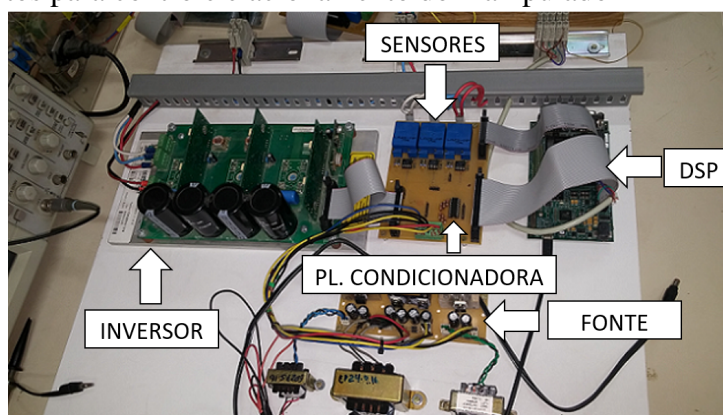
Para a leitura da posição do eixo do motor que é usado para fazer o controle de campo orientado indireto foi instalado um *encoder* do tipo incremental da *Avago*, modelo HEDS-5500, com resolução de 500 PPR (Pulsos por Revolução), na ponta traseira do eixo.

Para o condicionamento de sinais foi desenvolvida uma placa para comportar os sensores de corrente ligados à amplificadores operacionais de modo a ser possível calibrar os limites de tensão fornecidos pelos sensores ao *Digital Signal Processors* (DSP) e realizar a interface através de circuitos integrados com optoacopladores entre os sinais de controle e os sinais de potência.

Para a implementação do controle digital foi necessário a aquisição dos sinais de corrente e da posição do eixo do motor e implementação do controle por orientação de campo. Utilizou-se o processador digital de sinais, DSP (Digital Signal Processors), da *Texas Instruments*, modelo TMS320F28335. A principal vantagem da utilização deste modelo se deve ao fato de que, além do alto desempenho, sendo capaz de executar 150 MPIS (Milhões de Instruções por Segundo), o suporte de forma intrínseca à modulação em vetores espaciais, SVPWM, sem que seja necessária a implementação desta técnica no dispositivo, demandando um menor esforço de desenvolvimento do algoritmo.

Para o acionamento do motor utilizou-se um inversor trifásico de três níveis da *Semikron*, modelo SKS 18G B6111 V12, operando a uma frequência de chaveamento de 10kHz. Para a alimentação dos circuitos e placas foi desenvolvido uma fonte de tensão auxiliar, capaz de fornecer 4 níveis de tensão contínua. Sendo 18 V para chaveamento dos gatilhos do inversor trifásico, 15 vV para os sensores de corrente e 5 vV para alimentação da placa de condicionamento de sinais. As placas e circuitos utilizados são apresentados na Figura 66.

Figura 66 – Circuitos para controle e acionamento do manipulador



Fonte: (RABELO; L; R, 2017)

Para fazer o controle das juntas foi usado o modelo apresentado no diagrama de blocos da Figura 11 da seção 2.6.3.3 em que se faz primeiro o controle da malha interna de corrente, seguindo-se da malha de velocidade e por último a de posição. A Figura 11 apresenta o esquema do controle da malha de posição, em que foi projetado um controlador Proporcional-

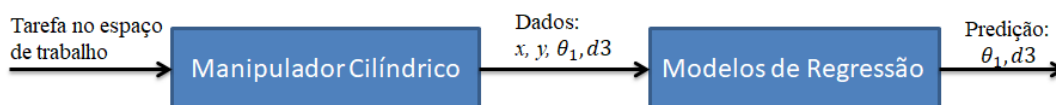
Integral (PI) em ambas as juntas.

De acordo com o diagrama de blocos da Figura 11 o θ_r^* é a referência da posição do controlador do que atua na malha através do sinal de controle ω_r^* . Através da leitura do *encoder* tem-se a posição atual do rotor θ_r^* . Maiores informações do projeto de todo o sistema de acionamento pode ser visto em (RABELO L, 2017).

Considerando-se as duas juntas que estão em funcionamento sendo a junta 1 que é base rotativa e a junta 3 que é linear. Desta forma, abordou-se também no manipulador duas dimensões no seu espaço de trabalho. Os mesmos métodos de regressões aplicados na simulação foram aplicados nos dados experimentais, porém aqui será usado apenas a cinemática.

A Figura 67 apresenta uma abordagem do problema, em que há uma modelagem da cinemática inversa a partir do conjunto de dados coletados do manipulador. Suas entradas em todos os algoritmos serão as coordenadas cartesianas (x, y) do *end-effector*, e as saídas são as trajetórias da juntas (θ_1, d_3).

Figura 67 – Regressão para a obtenção do mapeamento da cinemática inversa a partir dos dados experimentais.



Fonte: O autor (2022).

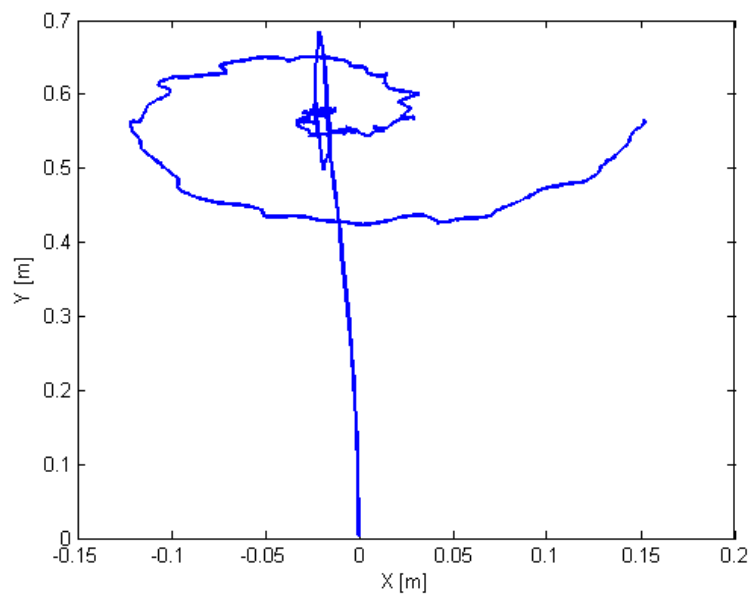
O mapeamento do manipulador foi feito através da aquisição de sensores (*encoders*) colocado nos motores e com a medição dos graus de liberdade para aquisição dos movimentos das duas juntas trabalhadas, conforme apresentado na Figura 11. Ainda foi usado uma cinemática direta das trajetórias das juntas para coletar os movimentos do *end-effector*. Assim, inicialmente fez-se o mapeamento da área de trabalho completa com o espaço cartesiano em duas dimensões de acordo com as limitações do manipulador a qual foi de 80 cm de altura e 160 cm de largura.

Para o mapeamento das trajetórias das juntas em relação a rotação dos motores, foi usado uma conversão de acordo com as dimensões de cada elo, distâncias e limitações. A posição da junta θ_1 será igual a posição da rotação do motor em graus/2,5 graus. Já a posição da junta d_3 é calculada considerando-se que a cada 1 grau de rotação do motor é igual a 0,0166 cm. Na preparação dos dados de treinamento / teste foi usado movimento espiral como entrada e as trajetórias das juntas 1 e 3 como saída do manipulador com as instâncias de 80% para treinamento e 20% para teste, após ser embaralhada (usando validação cruzada *hold-out*). Depois da geração de modelo foi aplicado as trajetórias elíptica e espiral de forma original, ou seja,

com a sequência correta. Foi aplicado em todos os métodos 1000 iterações e foi obtido como resultado a média dos coeficientes RMSE's e o R_d^2 , como também o melhor resultado de ambos.

A Figura 69 apresenta a movimento espiral no espaço cartesiano, ou seja, a trajetória real do *end-effector*. Ela será usada para o treinamento/teste dos métodos.

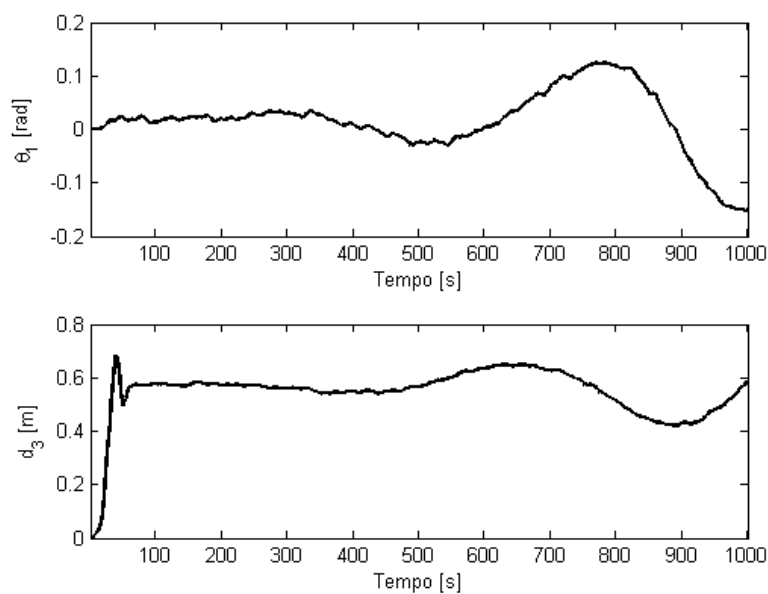
Figura 68 – Espaço cartesiano com o movimento helicoidal dos dados reais.



Fonte: O autor (2022).

A Figura 69 apresenta as trajetórias das juntas 1 e 3 do manipulador quando aplicado o movimento espiral no espaço cartesiano.

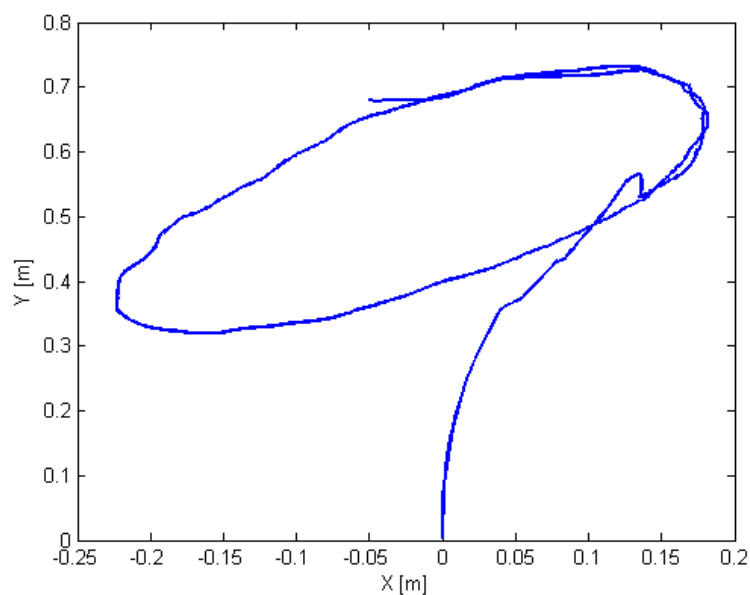
Figura 69 – Trajetória no espaço das juntas da posição durante o movimento helicoidal com dados reais.



Fonte: O autor (2022).

A Figura 70 mostra o movimento elíptico no espaço cartesiano. Os dados oriundos desse movimento serão aplicados após a etapa de treinamento/teste dos métodos.

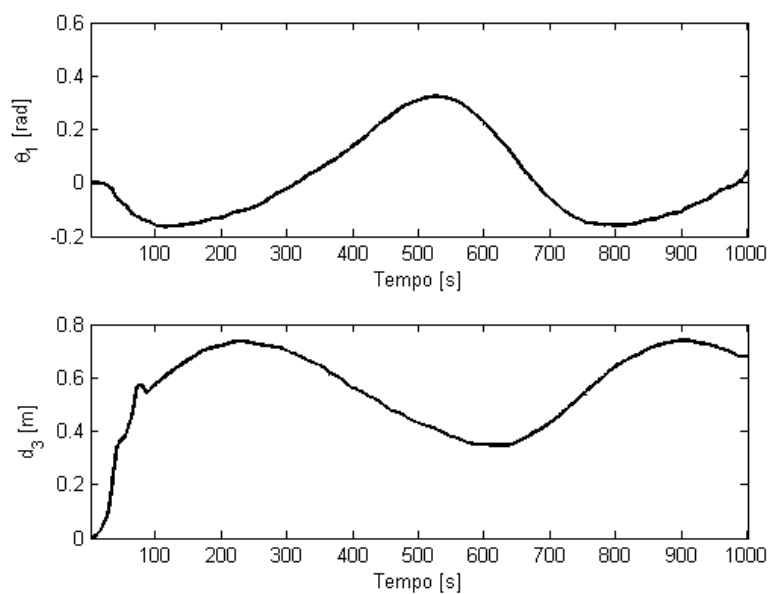
Figura 70 – Espaço cartesiano com o movimento elíptico com dados reais.



Fonte: O autor (2022).

A Figura 71 apresenta as trajetórias das juntas 1 e 3 do manipulador quando aplicado o movimento elíptico no espaço cartesiano.

Figura 71 – Trajetória no espaço das juntas da posição durante o movimento elíptico com dados reais.

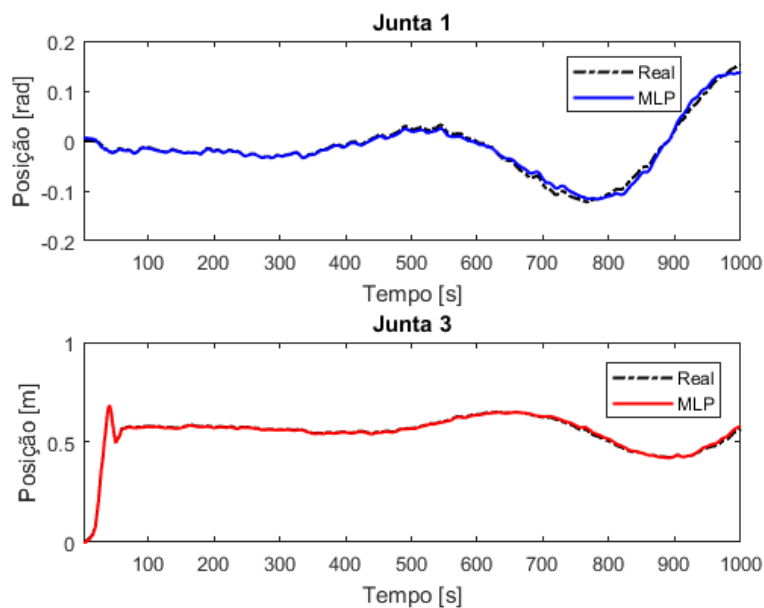


Fonte: O autor (2022).

4.5.1 Rede MLP com dados reais

A rede MLP foi usada da mesma forma que foi aplicada para a cinemática conforme seção 4.4.1, onde as entradas são as variáveis das coordenadas cartesianas e a saída desejada são as trajetórias reais das juntas. A Figura 72 apresentam os resultados da rede MLP nas duas juntas (1 e 3), na junta 1 o RMSE foi de 0,34, e o R_a^2 foi de 0,92. Já na junta 3 o RMSE foi de 0,31, e o R_a^2 foi de 0,95.

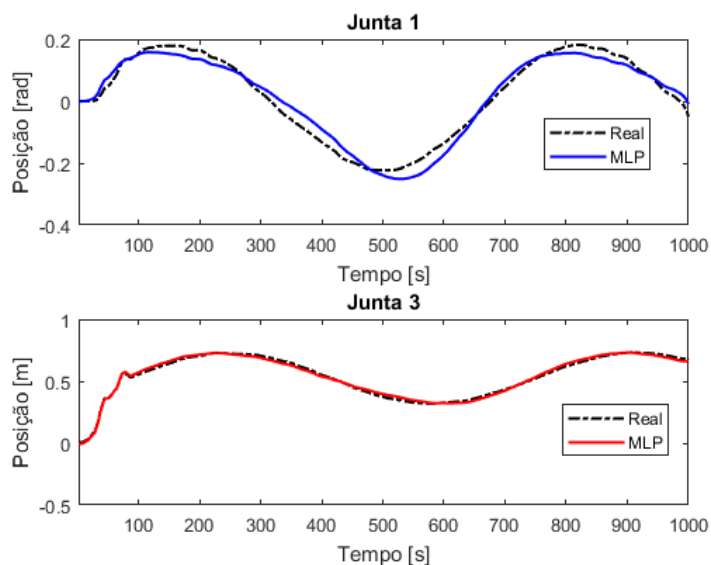
Figura 72 – Predição da rede MLP das posições reais no espaço das juntas na trajetória espiral



Fonte: O autor (2022).

També foi aplicado uma outra trajetória pós-validação com o mesmo modelo. Dessa vez, usou-se o movimento elíptico como entrada na rede, e as saídas as posições. Obteve-se o RMSE de 0,34 para a junta 1 e 0,32 da junta 3. Já o R_a^2 da junta 1 foi de 0,91 e da junta 3 foi de 0,94.

Figura 73 – Predição da rede MLP das posições reais no espaço das juntas na trajetória elíptica

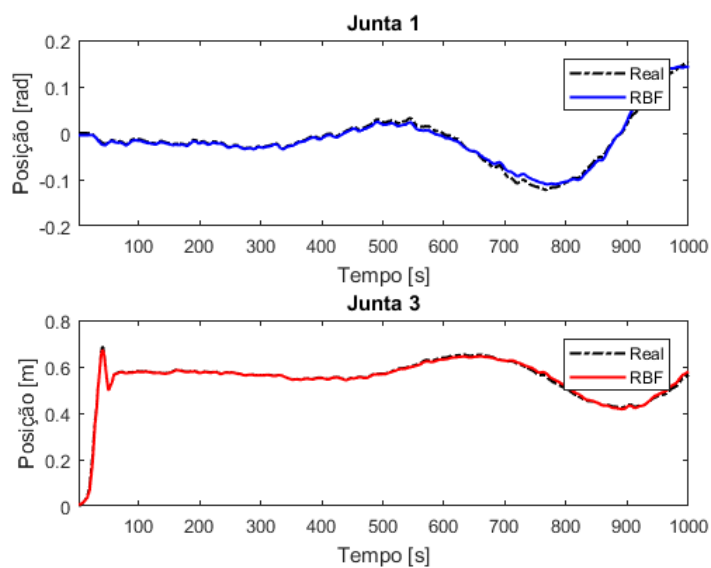


Fonte: O autor (2022).

4.5.2 Rede RBF com dados reais

O modelo mais adequado da RBF para estimar a cinemática é apresentado na Figura 74 para o movimento espiral. O RMSE foi de 0,27 para a junta 1 e 0,23 para a junta 3. Já o do R_a^2 da junta 1 foi de 0,95, e da junta 3 foi de 0,97.

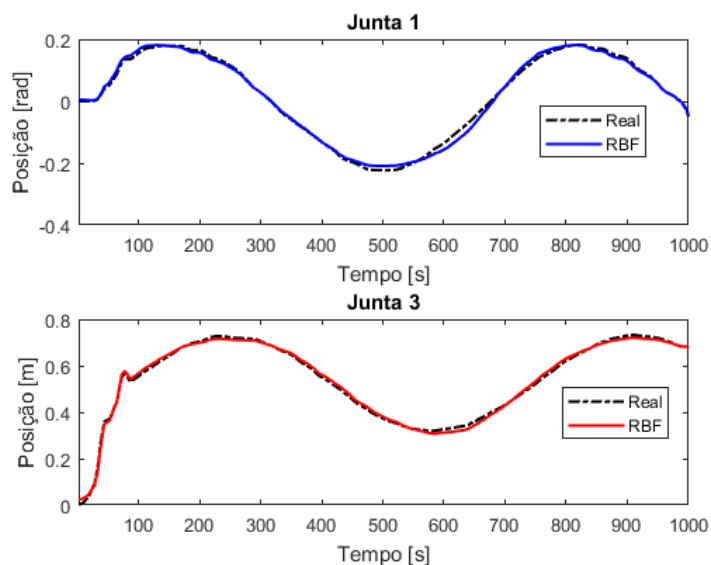
Figura 74 – Predição da rede RBF das posições reais no espaço das juntas na trajetória espiral



Fonte: O autor (2022).

O modelo superior da RBF para a trajetória elíptica é apresentado na Figura 75 . O RMSE foi de 0,27 para a junta 1 e 0,23 para a junta 3. Já o do R_a^2 da junta 1 foi de 0,95, e da junta 3 foi de 0,97.

Figura 75 – Predição da rede RBF das posições reais no espaço das juntas na trajetória elíptica

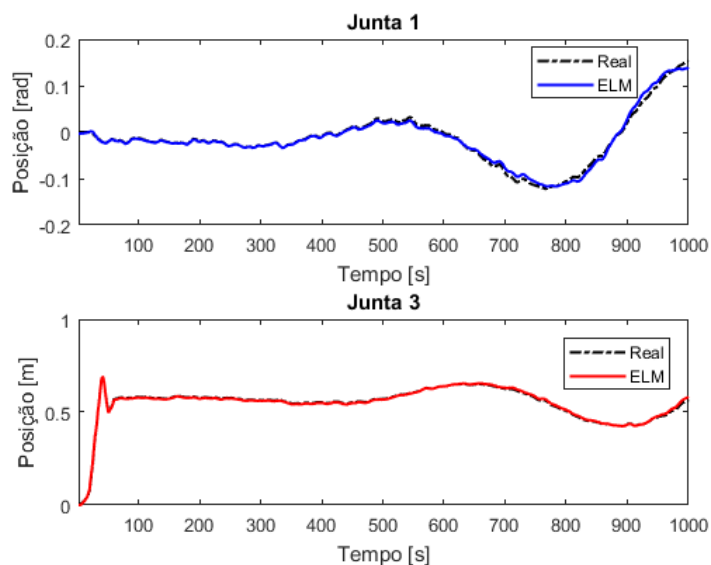


Fonte: O autor (2022).

4.5.3 Rede ELM

A ELM obteve um modelo para estimar a cinemática, que é apresentado o mais adequado na Figura 76 para o movimento espiral. O RMSE foi de 0,22 para a junta 1 e 0,19 para a junta 3. Já o do R_a^2 da junta 1 foi de 0,97, e da junta 3 foi de 0,98.

Figura 76 – Predição da rede ELM das posições reais no espaço das juntas na trajetória espiral

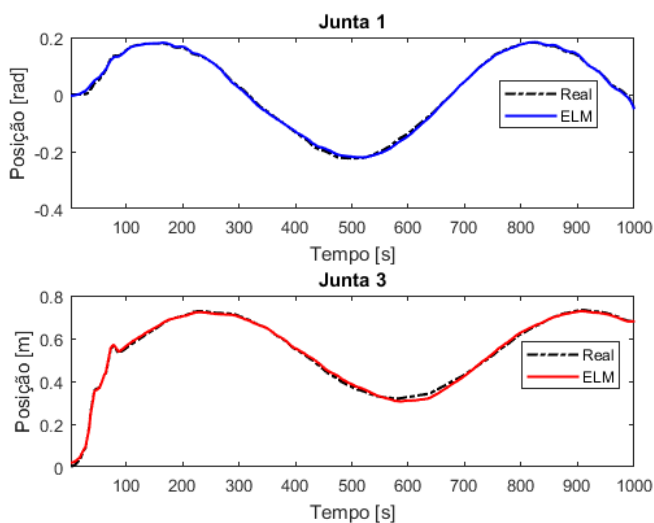


Fonte: O autor (2022).

Logo o modelo superior da ELM para a trajetória elíptica é apresentado na Figura 77. O RMSE foi de 0,20 para a junta 1 e 0,23 para a junta 3. Já o do R_a^2 da junta 1 foi de 0,97, e

da junta 3 foi de 0,95.

Figura 77 – Predição da rede ELM das posições reais no espaço das juntas na trajetória elíptica

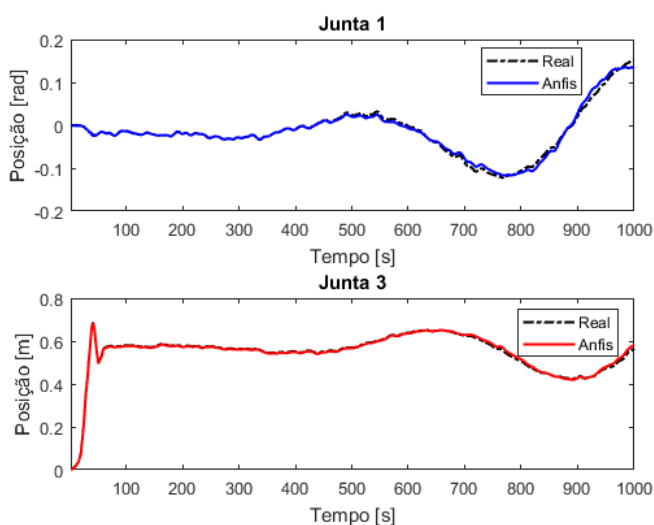


Fonte: O autor (2022).

4.5.4 ANFIS

O modelo da ANFIS mais adequado para estimar a cinemática, é apresentado na Figura 78 para o movimento espiral. O RMSE foi de 0,24 para a junta 1 e 0,19 para a junta 3. Já o do R_a^2 da junta 1 foi de 0,94, e da junta 3 foi de 0,98.

Figura 78 – Predição da rede ANFIS das posições reais no espaço das juntas na trajetória espiral

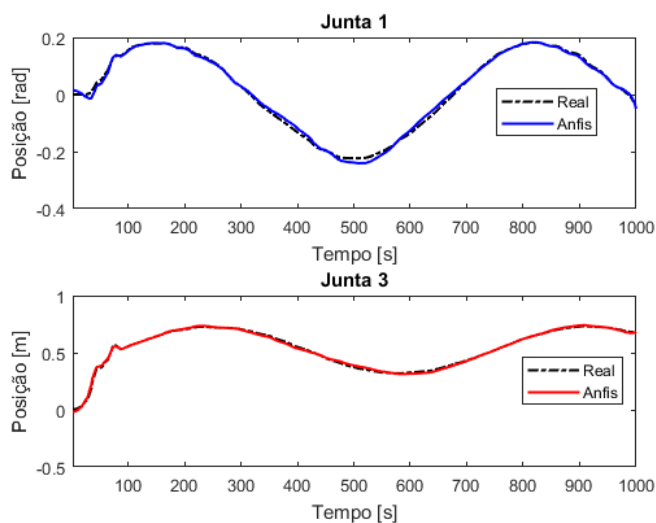


Fonte: O autor (2022).

Logo o modelo da ANFIS mais adequado para a trajetória elíptica é apresentado na Figura 77. O RMSE foi de 0,26 para a junta 1 e 0,23 para a junta 3. Já o do R_a^2 da junta 1 foi de

0,93, e da junta 3 foi de 0,94.

Figura 79 – Predição da rede ANFIS das posições reais no espaço das juntas na trajetória elíptica

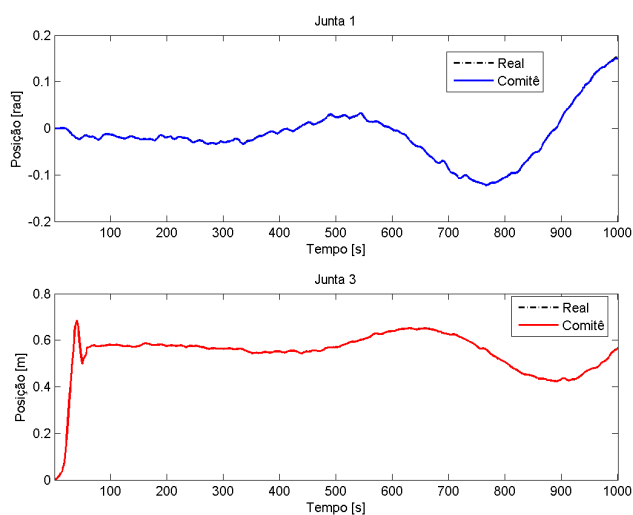


Fonte: O autor (2022).

4.5.5 Comitê de SVRs

O modelo proposto conforme os parâmetros ajustados e apresentado na seção 3.6, obteve para o movimento espiral para a trajetória real, conforme mostra a Figura 80. O RMSE de 0,12 para a junta 1 e 0,13 para a junta 3. Já o R_a^2 da junta 1 foi de 0,99 e da junta 3 foi de 0,98.

Figura 80 – Predição do Comitê das posições reais no espaço das juntas na trajetória espiral

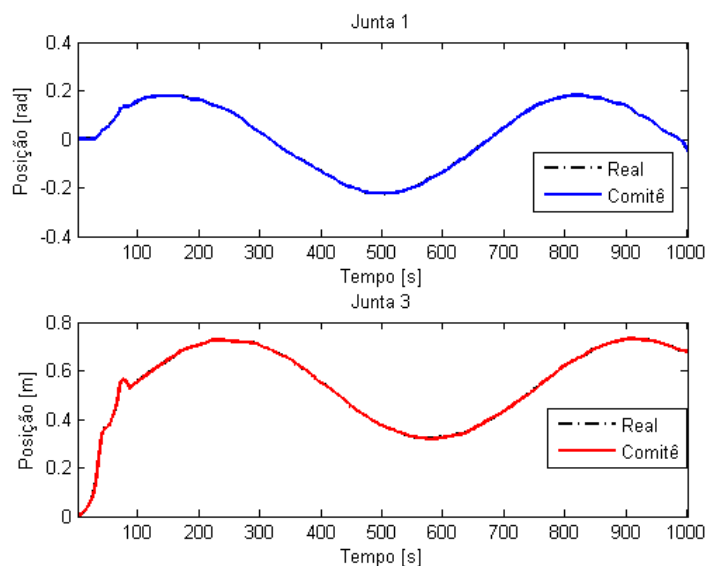


Fonte: O autor (2022).

A Figura 81 apresenta mais resultados do modelo proposto conforme para o movimento elíptico para a trajetória real. O RMSE foi de 0,11 para a junta 1 e 0,13 para a junta 3. Já

o R_a^2 da junta 1 foi de 0,99 e da junta 3 foi de 0,98.

Figura 81 – Predição do Comitê das posições reais no espaço das juntas na trajetória elíptica



Fonte: O autor (2022).

4.5.6 Comparação entre os métodos com dados reais

Os resultados apresentados na tabela 8 são as médias do coeficiente R_a^2 ajustado, RMSE e desvio padrão do erro dos modelos para cada junta durante a validação com dados reais do manipulador robótico. Nesse cenário com validação usando um movimento espiral completo, a máquina de comitê obteve um melhor desempenho em relação aos demais métodos.

Tabela 8 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento helicoidal das trajetórias com dados reais.

Métodos	R_{a1}^2	R_{a3}^2	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,8801	0,8910	0,2902	0,2811	0,7276	0,7189
<i>RBF</i>	0,9018	0,9201	0,2602	0,2504	0,7031	0,7012
<i>ELM</i>	0,9101	0,9204	0,2602	0,2401	0,7266	0,7201
<i>ANFIS</i>	0,9002	0,9003	0,2701	0,2501	0,6992	0,7002
Comitê proposto	0,9504	0,9701	0,2003	0,1904	0,0734	0,0761

Fonte: O autor (2022).

A Tabela 9 apresenta os resultados médios do R_a^2 ajustado, RMSE e desvio padrão do erro dos modelos para cada junta durante a validação com dados reais do manipulador robótico. Dessa vez o movimento usado é o elíptico completo. Mais uma vez a máquina de comitê obteve um desempenho superior em relação aos demais métodos.

Tabela 9 – Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico das trajetórias com dados reais.

Métodos	R_{a1}^2	R_{a3}^2	RMSE ₁	RMSE ₃	σ_1	σ_3
<i>MLP</i>	0,8701	0,8804	0,2910	0,2931	0,7387	0,7178
<i>RBF</i>	0,9110	0,9202	0,2703	0,2511	0,7215	0,7105
<i>ELM</i>	0,9212	0,9210	0,2613	0,2501	0,7324	0,7126
<i>ANFIS</i>	0,8914	0,9010	0,2802	0,2611	0,7017	0,7066
Comitê proposto	0,9521	0,9613	0,2010	0,2011	0,0778	0,0738

Fonte: O autor (2022).

Apesar da máquina de comitê mostrar um desempenho superior em relação aos demais métodos, o tempo de execução foi inferior aos métodos ELM e RBF. Porém, a comparação mostrada nos resultados entre o comitê e os demais métodos é válido nos cenários mostrados com dados experimentais. Também foram testados os comitês de todos os métodos analisados neste trabalho e comparados com o proposto, e encontra-se no Apêndice C

5 CONCLUSÕES

Na literatura atual de aproximação de modelos em robótica industrial, os conceitos básicos de inúmeras aplicações dentro desse campo vêm sendo reformulados ao longo dos anos visando a redução de custos, o aumento de disponibilidade e maior segurança. Dessa forma, estudos sobre previsibilidade de degradação de equipamentos e correção estão ganhando cada vez mais espaço. Novas técnicas para aquisição e tratamento de dados, bem como algoritmos e metodologias passaram a ser foco de pesquisas recentes a fim de otimizar o monitoramento de equipamentos e componentes de sistemas críticos ou de grande valor agregado.

Neste trabalho, foi proposto um método baseado em regressão de comitê aplicado a 3 SVRs com diferentes parâmetros e com um critério baseado no erro da saída de cada especialista. O método proposto foi comparado com 4 algoritmos muito utilizado na área aprendizado de máquina que são: a rede MLP, RBF, ELM e ANFIS. Foram aplicados aos modelos de regressão algumas trajetórias conhecidas dentro do espaço de trabalho do manipulador com dados de simulação e experimentais. A ideia geral é que através do espaço cartesiano do manipulador sendo a entrada dos métodos, faz-se o mapeamento de todos o espaço das juntas relacionadas as coordenadas cartesianas, com a validação cruzada. Após isso foi validado com 2 trajetórias do efetuador: com os dados de simulações no plano 3D, a trajetória elíptica e helicoidal, já com dados experimentais foi a trajetória elíptica e espiral, também as mesmas trajetórias foram aplicadas carga variável para dificultar os rastreamentos. Todos os métodos foram avaliados pelo RMSE, R_a^2 e o desvio padrão do erro com 1000 rodadas de execuções cada.

Foram apresentados após as 1000 rodadas de execuções em todos os cenários, o melhor e a média do RMSE, e do coeficiente R_a^2 . Essas análises foram úteis para verificar a eficácia dos algoritmos em situações usando a validação cruzada com o intuito de não enviesar os dados. Ao observar os resultados, verificou-se que a MLP obteve um pior desempenho em relação aos outros métodos no contexto geral, isso se deve a quantidade de hiper-parâmetros que a mesma carregou durante a sua execução. Logo, o R_a^2 avalia a qualidade do modelo juntamente com a quantidade de parâmetros.

O método proposto obteve um melhor desempenho em relação aos outros métodos, desta forma, apresentou-se o melhor resultado e a média de ambos os índices de desempenhos durante as iterações e após a geração do modelo pela validação cruzada.

5.1 Trabalhos futuros

Para as etapas futuras, pretende-se acionar o segundo grau de liberdade do manipulador real, pois possui a gravidade e muitas incertezas que possam dificultar o desenvolvimento. Além do acionamento completo com o processamento de imagem para coletar as trajetórias no espaço cartesiano em tempo real no sistema.

Com isso pode-se fazer novas modelagens com todos os graus de liberdade quando o manipulador estiver completamente em funcionamento. Assim será testado o método proposto e comparado com os demais apresentados nesse trabalho novamente e podendo incluir novas contribuições, além de auxiliar no controle adaptativo das juntas do manipulador.

Uma outra proposta para trabalho futuro seria a instalação de uma câmera por cima do manipulador. A partir disto, seria desenvolvido um sistema de rastreamento das trajetórias via processamento de imagem com o objetivo de fazer o controle com auxílio de técnicas de processamento de imagens.

Pode-se ainda apresentar uma outra proposta que seria o planejamento de trajetórias livres de colisão com algumas técnicas de inteligência computacional implementadas experimentalmente.

REFERÊNCIAS

- ABDULLAH, S. S.; MALEK, M. A.; ABDULLAH, N. S.; KISI, O.; YAP, K. S. Extreme learning machines: a new approach for prediction of reference evapotranspiration. **Journal of Hydrology**, Elsevier, v. 527, p. 184–195, 2015. Disponível em: <<https://doi.org/10.1016/j.jhydrol.2015.04.073>>. Acesso em: 17 ago. 2018.
- BENNETT, K. P.; MANGASARIAN, O. L. **Robust linear programming discrimination of two linearly inseparable sets**. University of Wisconsin-Madison: [s.n.], 1992.
- BOSE, B. K. **Power Electronics and AC drives**. University of Michigan: Englewood, NJ: Prentice-Hall, 1986. v. 2.
- CARLOS, L. F.; BARRAGÁN, J. H.; ALANIS, A. Y.; DANIEL A., N. Inverse kinematics of mobile manipulators based on differential evolution. **International Journal of Advanced Robotic Systems**, 2018. Disponível em: <<https://doi.org/10.1177/1729881417752738>>. Acesso em: 05 jan. 2019.
- CIRESAN, D. C.; MEIER, U.; MASCI, J.; SCHMIDHUBER, J. A committee of neural networks for traffic sign classification. *In: International Joint Conference on Neural Networks (IJCNN)*. [s.n.], 2011. p. 1918–1921. Disponível em: <10.1109/IJCNN.2011.6033458>. Acesso em: 27 out. 2018.
- CORTES, C.; VAPNIK, V. Support vector machine. **Machine learning**, v. 20, n. 3, p. 273–297, 1995. Disponível em: <<https://link.springer.com/article/10.1007/bf00994018>>. Acesso em: 25 ago. 2017.
- CRAIG, L. W. Control structure for batch reactor control. **Plant/Operations Progress**, v. 8, n. 1, p. 35–39, 1989. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/prsb.720080112>>. Acesso em: 28 mar. 2019.
- EL-SHERBINY, A.; ELHOSSEINI, M. A.; HAIKAL, A. Y. A comparative study of soft computing methods to solve inverse kinematics problem. **Ain Shams Engineering Journal**, v. 9, n. 4, p. 2535–2548, 2018. ISSN 2090-4479. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2090447917300965>>. Acesso em: 28 jan. 2019.
- EL-SHERBINY, A.; ELHOSSEINI, M. A.; HAIKAL, A. Y. A new abc variant for solving inverse kinematics problem in 5 dof robot arm. **Applied Soft Computing**, v. 73, p. 24–38, 2018. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494618304885>>. Acesso em: 18 jan. 2019.
- ENTEKHABI, D.; REICHLER, R. H.; KOSTER, R. D.; CROW, W. T. Performance metrics for soil moisture retrievals and application requirements. **Journal of Hydrometeorology**, v. 11, n. 3, p. 832–840, 2010. Disponível em: <<https://doi.org/10.1175/2010JHM1223.1>>. Acesso em: 20 fev. 2018.
- FLETCHER, R. A new variational result for quasi-newton formulae. **SIAM Journal on Optimization**, SIAM, v. 1, n. 1, p. 18–21, 1991. Disponível em: <<https://doi.org/10.1137/0801002>>. Acesso em: 15 jan. 2018.
- GUPTA, A.; BHARGAVA, P.; DESHMUKH, A.; AGRAWAL, S.; CHOURIKA, S. A geometric approach to inverse kinematics of a 3 dof robotic arm. **International Journal for Research**

- in **Applied Science & Engineering Technology (IJRASET)**, India, p. 3524–3530, 2018. Disponível em: <<https://www.ijraset.com/files/serve.php?FID=13030>>. Acesso em: 10 fev. 2019.
- HARTENBERG, R.; DANAVIT, J. **Kinematic synthesis of linkages**. New York, USA: McGraw-Hill, 1964.
- HAYKIN, S. **Neural networks: principles and practice**. McNaster University, Hamilton, Canada: [s.n.], 2001. v. 11. 900 p.
- HOU, Z.; SHEN, Q.; LI, H. Nonlinear system identification based on anfis. In: **International Conference on Neural Networks and Signal Processing**. United States: [s.n.], 2003. v. 1, p. 510–512. Disponível em: <10.1109/ICNNSP.2003.1279323>. Acesso em: 08 ago. 2017.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: theory and applications. **Neurocomputing**, Elsevier, v. 70, n. 1-3, p. 489–501, 2006. Disponível em: <<https://doi.org/10.1016/j.neucom.2005.12.126>>. Acesso em: 09 ago. 2017.
- IFR. **Foreword World Robotics Industrial Robots**. [s.n.], 2021. Disponível em: <<https://ifr.org/ifr-press-releases/news/robot-sales-rise-again>>. Acesso em: 28 fev. 2022.
- ISERMANN, R.; MÜNCHHOF, M. **Identification of Dynamic Systems: An Introduction with Applications**. Switzerland AG: [s.n.], 2011.
- JIANGPING, W.; SHIRONG, L.; BOTAO, Z.; CHANGBIN, Y. Inverse kinematics-based motion planning for dual-arm robot with orientation constraints. **International Journal of Advanced Robotic Systems**, USA, p. 1–14, 2019. Disponível em: <<https://doi.org/10.1177/1729881419836858>>. Acesso em: 19 fev. 2020.
- JOELIANTO, E.; ANURA, D. C.; PRIYANTO, M. Anfis–hybrid reference control for improving transient response of controlled systems using pid controller. **International Journal of Artificial Intelligence**, v. 10, n. S13, p. 88–111, 2013. Disponível em: <https://www.aut.upt.ro/~rprecup/IJAI_21.pdf>. Acesso em: 15 mar. 2018.
- KEERTHI, S. S.; CHAPELLE, O.; DECOSTE, D. Building support vector machines with reduced classifier complexity. **Journal of Machine Learning Research**, United States, v. 7, p. 1493–1515, 2006. Disponível em: <<https://www.jmlr.org/papers/volume7/keerthi06a/keerthi06a.pdf>>. Acesso em: 10 jan. 2018.
- KELLY, R.; DAVILA, V. S.; PEREZ, J. A. L. **Control of robot manipulators in joint space**. Leipzig, Germany: Springer Science & Business Media, 2006.
- KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **International Joint Conference on Artificial Intelligence**. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995. v. 2, p. 1137–1143. ISBN 1558603638. Disponível em: <<https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf>>. Acesso em: 10 set. 2017.
- KORKMAZ, O.; IDER, S. K. Hybrid force and motion control of flexible joint parallel manipulators using inverse dynamics approach. **Advanced Robotics**, London, England, v. 28, n. 18, p. 1221–1230, 2014. Disponível em: <<https://doi.org/10.1080/01691864.2014.920719>>. Acesso em: 12 out. 2017.

KUHN, H. W.; TUCKER, A. W. Nonlinear programming. *In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951. p. 481–492. Disponível em: <<https://projecteuclid.org/euclid.bsm/1200500249>>. Acesso em: 09 set. 2017.

KURFESS, T. R. **Robotics and Automation Handbook**. London - UK: [s.n.], 2004.

LI, G.; CUI, Y.; WANG, L.; MENG, L. Research on 6-dof robot inverse kinematics based on blended optimization algorithm of elm-ssa-sca. **Advances in Mechanical Engineering**, London, England, v. 14, n. 7, p. 1–8, 2022. Disponível em: <<https://journals.sagepub.com/doi/pdf/10.1177/16878132221113419>>. Acesso em: 2022.

LI, Z.; HUANG, B.; AJODANI, A.; YANG, C.; SU, C.-Y.; BICCHI, A. Asymmetric bimanual control of dual-arm exoskeletons for human-cooperative manipulations. **IEEE Transactions on Robotics**, v. 34, n. 1, p. 264–271, 2018. Disponível em: <<https://doi.org/10.1109/TRO.2017.2765334>>. Acesso em: 20 mar. 2019.

LIAO, J.; MCGEE, D. Adjusted coefficients of determination for logistic regression. **The American Statistician**, London, England, v. 57, n. 3, p. 161–165, 2003. Disponível em: <<https://doi.org/10.1198/0003130031964>>. Acesso em: 18 set. 2017.

MITTAL, R.; NAGRATH, I. **Robotics and control**. [S.l.]: Tata McGraw-Hill, 2003.

NADIRI, A. A.; FIJANI, E.; TSAI, F. T.-C.; MOGHADDAM, A. A. Supervised committee machine with artificial intelligence for prediction of fluoride concentration. **Journal of Hydroinformatics**, London, England, v. 15, n. 4, p. 1474–1490, 2013. Disponível em: <<https://doi.org/10.2166/hydro.2013.008>>. Acesso em: 11 nov. 2017.

NADIRI, A. A.; GHAREKHANI, M.; KHATIBI, R.; SADEGHFAM, S.; MOGHADDAM, A. A. Groundwater vulnerability indices conditioned by supervised intelligence committee machine (sicm). **Science of the Total Environment**, Amsterdam, Netherlands, v. 574, p. 691–706, 2017. Disponível em: <<https://doi.org/10.1016/j.scitotenv.2016.09.093>>. Acesso em: 03 mar. 2018.

ONDER, T.; BARKANA, E.; DUYGUN, C.; SULE. Learning to balance an nao robot using reinforcement learning with symbolic inverse kinematic. **Transactions of the Institute of Measurement and Control**, v. 39, n. 11, p. 1735–1748, 2017. Disponível em: <<https://doi.org/10.1177/0142331216645176>>. Acesso em: 03 abr. 2019.

RABELO L, R. **Controle de um motor de indução trifásico como servoposicionador aplicado a um manipulador robótico utilizando controle de campo orientado** 2017. 107 f. Dissertação (Mestrado em Engenharia Elétrica) — Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Ceará, Fortaleza, 2017.

RAIN, T.; DOVGAL, V. M.; SOE, Y. N. Efficient method for inverse dynamics of robot manipulators by using adaptive-network-based fuzzy inference system. *In: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. United States: [s.n.], 2019. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICIEAM.2019.8743014>>. Acesso em: 19 jan. 2020.

SANTOS, V. M. F. **Robótica Industrial**. [S.l.]: Departamento de Engenharia Mecânica, Universidade de Aveiro, 2004.

SANZ, P. Robotics: Modeling, planning, and control (siciliano, b. et al; 2009)[on the shelf]. **IEEE Robotics & Automation Magazine**, teste, v. 16, n. 4, p. 101–101, 2009.

SCIAVICCO, L.; SICILIANO, B. **Modeling and control of robot manipulators**. New York, USA: McGraw-Hill, 1996. v. 8.

SHARMA, R.; KUMAR, V.; GAUR, P.; MITTAL, A. An adaptive pid like controller using mix locally recurrent neural network for robotic manipulator with variable payload. **ISA Transactions**, v. 62, p. 258–267, 2016. ISSN 0019-0578. SI: Control of Renewable Energy Systems. Disponível em: <<https://doi.org/10.1016/j.isatra.2016.01.016>>. Acesso em: 19 ago. 2019.

SHERBINY, A.; ELHOSSEINI M. A.AND HIAKAL, A. Y. A new abc variant for solving inverse kinematics problem in 5 dof robot arm. **Applied Soft Computing**, Amsterdam, Netherlands, p. 24–38, 2018. ISSN 1568-4946. Disponível em: <<https://doi.org/10.1016/j.asoc.2018.08.028>>. Acesso em: 07 fev. 2019.

SICILIANO, B.; SCIAVICCO, L.; VILLANI, L.; ORIOLO, G. **Robotics: Modelling, Planning and Control**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 1846286417, 9781846286414.

SICILIANO, B.; SCIAVICCO, L.; VILLANI, L.; ORIOLO, G. **Robotics: modelling, planning and control**. [S.l.]: Springer Science & Business Media, 2010.

SOUZA JÚNIOR, A. B. D. **Estudo e implementação de um servoposicionador aplicando controle vetorial indireto a um motor de indução trifásico**. Dissertação (Mestrado em Engenharia Elétrica) — Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Ceará, Fortaleza, 2010.

SPONG, M. W.; VIDYASAGAR, M. **Robot dynamics and control**. [S.l.]: John Wiley & Sons, 2008.

TAKAHASHI, K.; OGATA, T.; NAKANISHI, J.; CHENG, G.; SUGANO, S. Dynamic motion learning for multi-dof flexible-joint robots using active–passive motor babbling through deep learning. **Advanced Robotics**, London, Englad, v. 31, n. 18, p. 1002–1015, 2017. Disponível em: <<https://doi.org/10.1080/01691864.2017.1383939>>. Acesso em: 17 set. 2018.

VU, N. T.-T.; TRAN, N. P.; NGUYEN, N. H. Adaptive neuro-fuzzy inference system based path planning for excavator arm. **Journal of Robotics**, London, UK, p. 1–8, 2018. Disponível em: <<https://doi.org/10.1155/2018/2571243>>. Acesso em: 06 fev. 2019.

WANG, J.; LIU, G. A point interpolation meshless method based on radial basis functions. **International Journal for Numerical Methods in Engineering**, USA, v. 54, n. 11, p. 1623–1648, 2002. Disponível em: <<https://doi.org/10.1002/nme.489>>. Acesso em: 02 out. 2017.

WEI, Y.; RAHMANI, A. Motion tracking and object manipulation of a hyper-redundant mdams under joint limits based on task-priority & relative jacobian. **Chinese Control And Decision Conference (CCDC)**, 2018. Disponível em: <<https://doi.org/10.1109/CCDC.2018.8407984>>. Acesso em: 03 jan. 2019.

WRIGHT, G. B. **Radial basis function interpolation: numerical and analytical developments**. Tese (Doutorado), University of Colorado at Boulder, 2003.

YU, Z. Space vector pwm witch tms320c24x using hardware and software determined switching patterns. **Application Report SPRA524, Texas Instruments**, Texas, USA, 2001. Disponível em: <<https://www.ti.com/>>. Acesso em: 03 ago. 2017.

YUAN, P.; CHEN, D.; WANG, T.; CAO, S.; CAI, Y.; XUE, L. A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot. **Advances in Mechanical Engineering**, London, England, v. 10, n. 3, p. 1–11, 2018. Disponível em: <<https://doi.org/10.1177/1687814018763411>>. Acesso em: 05 fev. 2019.

ZHANG, M.; Li, W. Single neuron pid model reference adaptive control based on rbf neural network. *In: International Conference on Machine Learning and Cybernetics*. United States: [s.n.], 2006. p. 3021–3025. ISSN 2160-133X. Disponível em: <<https://doi.org/10.1109/ICMLC.2006.258358>>. Acesso em: 27 jan. 2018.

ZHAO, R.; SHI, Z.; GUAN, Y.; SHAO, Z.; ZHANG, Q.; WANG, G. Inverse kinematic solution of 6r robot manipulators based on screw theory and the paden-kahan subproblem. **International Journal of Advanced Robotic Systems**, United States, p. 1–11, 2018. Disponível em: <<https://doi.org/10.1177/1729881418818297>>. Acesso em: 11 mar. 2019.

ZHOU, Z.; GUO, H.; WANG, Y.; ZHU, Z.; WU, J.; LIU, X. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm. **International Journal of Advanced Robotic Systems**, United States, v. 15, p. 1–15, 2018. Disponível em: <<https://doi.org/10.1177/1729881418792992>>. Acesso em: 08 fev. 2019.

ZUNT, D. **Who did actually invent the word "robot"and what does it mean?** Czech Republic: The Karel Čapek website, 2013. Disponível em: <<https://web.archive.org/web/20130123023343/http://capek.misto.cz/english/robot.html>>. Acesso em: 17 mar. 2018.

**APÊNDICE A – RESULTADOS DE SIMULAÇÃO COM OS COMITÊS DOS MÉTODOS
COM A CARGA CONSTANTE**

Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque de todos os métodos para trajetória helicoidal com carga constante.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,9217	0,8014	0,4319	0,4320	0,8186	0,8090
Comitê RBF	0,9410	0,9400	0,2068	0,0797	0,7990	0,7910
Comitê ELM	0,9521	0,9506	0,2086	0,1994	0,8089	0,8195
Comitê ANFIS	0,9340	0,9242	0,1926	0,1997	0,8181	0,8169
Comitê proposto	0,9721	0,9719	0,0638	0,0435	0,0687	0,0598

Fonte: O autor (2022).

Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico do torque de todos os métodos com carga constante.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,9160	0,9130	0,2090	0,2924	0,7058	0,7044
Comitê RBF	0,8712	0,9086	0,2460	0,0026	0,6832	0,6855
Comitê ELM	0,9250	0,9200	0,2480	0,2628	0,7180	0,7124
Comitê ANFIS	0,9014	0,9220	0,2366	0,2396	0,6970	0,6891
Comitê proposto	0,9411	0,9403	0,0950	0,1824	0,0832	0,0829

Fonte: O autor (2022).

**APÊNDICE B – RESULTADOS DE SIMULAÇÃO COM OS COMITÊS DOS MÉTODOS
COM A CARGA VARIÁVEL**

Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento helicoidal com carga variável.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,8612	0,7822	0,7611	0,5177	0,8289	0,8243
Comitê RBF	0,9256	0,9188	0,5326	0,2920	0,7890	0,7893
Comitê ELM	0,9089	0,8682	0,6427	0,3628	0,8395	0,8288
Comitê ANFIS	0,9079	0,9182	0,2044	0,1828	0,7586	0,7484
Comitê proposto	0,9475	0,9411	0,2032	0,1824	0,0857	0,0861

Fonte: O autor (2022).

Comparação da média do R_a^2 , RMSE e desvio padrão do erro do torque do movimento elíptico com carga variável.

Metodos	$R_{a_1}^2$	$R_{a_3}^2$	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,9223	0,7277	0,2090	0,4718	0,8190	0,8200
Comitê RBF	0,8291	0,8789	0,2708	0,2985	0,7960	0,7954
Comitê ELM	0,8390	0,7887	0,1928	0,4527	0,8287	0,8283
Comitê ANFIS	0,8633	0,7791	0,2719	0,4397	0,7697	0,7609
Comitê proposto	0,9389	0,8856	0,0720	0,2902	0,0862	0,0865

Fonte: O autor (2022).

APÊNDICE C – RESULTADOS COM DADOS EXPERIMENTAIS COM OS COMITÊS DOS MÉTODOS

Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento helicoidal das trajetórias com dados reais.

Métodos	R_{a1}^2	R_{a3}^2	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,8812	0,8921	0,2821	0,2703	0,7275	0,7188
Comitê RBF	0,9102	0,9310	0,2514	0,2402	0,7021	0,7002
Comitê ELM	0,9123	0,9221	0,2511	0,2400	0,7234	0,7123
Comitê ANFIS	0,9101	0,9233	0,2620	0,2431	0,6990	0,7001
Comitê proposto	0,9504	0,9701	0,2003	0,1904	0,0734	0,0761

Fonte: O autor (2022).

Comparação da média do R_a^2 , RMSE e desvio padrão do erro com movimento elíptico das trajetórias com dados reais.

Métodos	R_{a1}^2	R_{a3}^2	RMSE ₁	RMSE ₃	σ_1	σ_3
Comitê MLP	0,8875	0,8830	0,2820	0,2803	0,7386	0,7177
Comitê RBF	0,9120	0,9303	0,2606	0,2460	0,7213	0,7103
Comitê ELM	0,9222	0,9220	0,2560	0,2480	0,7323	0,7125
Comitê ANFIS	0,8924	0,9111	0,2780	0,2590	0,7016	0,7064
Comitê proposto	0,9521	0,9613	0,2010	0,2011	0,0778	0,0738

Fonte: O autor (2022).