



Trabalho de Conclusão de Curso

# **Modelagem Dinâmica e Estática do Braço Robótico UR5**

Valdir de Souza Junior  
vsj@ic.ufal.br

Orientador:  
Prof. Dr. Erick de Andrade Barboza

Maceió, Janeiro de 2021

Valdir de Souza Junior

# **Modelagem Dinâmica e Estática do Braço Robótico UR5**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Erick de Andrade Barboza

Maceió, Janeiro de 2021

**Catálogo na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

S729m Souza Júnior, Valdir de.

Modelagem dinâmica e estática do braço robótico UR5 / Valdir de Souza Júnior. – 2021.

45 f. : il., figs. e tabs. color.

Orientador: Erick de Andrade Barboza.

Monografia (Trabalho de Conclusão de Curso em Engenharia da Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2021..

Bibliografia: f. 44-45.

1. Modelagem matemática. 2. Robótica. 3. UR5 (Manipulador robótico).  
I. Título.

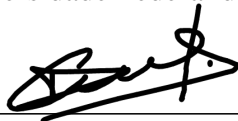
CDU: 004.896

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Erick de A. Barboza

---

Prof. Dr. Erick de Andrade Barboza - Orientador  
Instituto de Computação  
Universidade Federal de Alagoas



---

Prof. Dr. Thiago Damasceno Cordeiro - Examinador  
Instituto de Computação  
Universidade Federal de Alagoas

João Raphael Souza Martins

---

Prof. Dr. João Raphael Souza Martins - Examinador  
Instituto de Computação  
Universidade Federal de Alagoas

# Agradecimentos

Agradeço primeiro a Deus por ter me mantido na trilha certa durante minha graduação com saúde e forças para chegar até o final.

Agradeço aos meus pais por todo o esforço investido na minha educação.

Agradeço à minha namorada por sempre estar ao meu lado durante o meu percurso acadêmico e por me fazer ter confiança nas minhas decisões.

Agradeço a meus colegas do curso, pelas trocas de ideias e ajuda mútua. Juntos conseguimos avançar e ultrapassar todos os obstáculos.

Agradeço a meus amigos, que sempre estiveram ao meu lado, pelo apoio demonstrado ao longo de todo o período em que me dediquei a este trabalho.

Também quero agradecer à Universidade Federal de Alagoas e a todos os professores do meu curso pela elevada qualidade do ensino oferecido.

Deixo um agradecimento especial ao meu orientador, Erick Barboza, pelo incentivo e pela dedicação do seu tempo ao meu trabalho de conclusão de curso.

*“Se eu vi mais longe, foi por estar sobre ombros de gigantes.”*

– Newton, *Isaac*

# Resumo

Existe um grande impedimento no trabalho conjunto entre academia e indústria a respeito de produções envolvendo manipuladores robóticos, pois os detalhes físicos de um robô não são facilmente disponibilizados, inviabilizando a aplicação de novas tecnologias na indústria. Esse trabalho produz um modelo matemático de um robô para viabilizar uma maior aproximação entre academia e indústria. É descrito o procedimento para realizar uma análise cinemática e dinâmica de um braço robótico de 6 graus de liberdade. Foi escolhido como estudo de caso o Manipulador Robótico UR5, desenvolvido pela *Universal Robots*, que é um manipulador muito presente na indústria e que possui recursos que o permite trabalhar colaborativamente com seres humanos. Durante a análise, foram escolhidos os parâmetros de Denavit-Hartenberg para representar a construção do Manipulador Robótico, e a equação de Euler-Lagrange é utilizada para derivar a equação de dinâmica. Utilizando técnicas de otimização não lineares, é definido um controlador proporcional e derivativo para controlar o robô. A fim de validar a modelagem é elaborada uma trajetória, e são elaborados testes sendo executados em malha aberta e em malha fechada utilizando um controlador.

**Palavras-chave:** Robótica; UR5; Manipulador Robótico; Euler-Lagrange.

# Abstract

There is a great impediment in the joint work between academy and industry regarding works involving robotic manipulators, because the physical details of the robot are not easily available, making it difficult to apply new technologies in the industry. This work produces a mathematical model for a robot to enable a closer relationship between the academy and industry. It is described the procedure to perform kinematic and dynamic analysis of a robotic arm with 6 degrees-of-freedom. The UR5 Robotic Manipulator, developed by *Universal Robots*, was chosen as a case study, which is a manipulator very present in the industry and has resources that allow it to work collaboratively with human beings. During the analysis, the Denavit-Hartenberg parameters were chosen to represent the construction of the Robotic Manipulator, and the Euler-Lagrange equation is used to derive the dynamics equation. Using nonlinear optimization techniques, a proportional and derivative controller is defined to control the robot. In order to validate the modeling, a trajectory was elaborated, and tests were elaborated and executed in open loop and closed loop using a controller.

**Key-words:** Robotics; UR5; Robot Manipulator; Euler-Lagrange.

# Lista de Figuras

1	Manipulador Robótico UR5. . . . .	12
2	Representação simbólica das juntas de revolução. . . . .	15
3	Demonstração dos parâmetros de DH. . . . .	20
4	Cilindro com altura $h$ e raio $r$ . . . . .	23
5	Controle no Espaço das Juntas. . . . .	27
6	Demonstração da função <i>ikine</i> , disponibilizada na <i>toolbox</i> de Robótica. . . . .	29
7	Fluxograma de Trabalho. . . . .	32
8	Pontos-chave da trajetória aplicados no robô . . . . .	35
9	Posições da trajetória - Juntas 1 a 6. . . . .	36
10	Exemplo de singularidade interna. . . . .	36
11	Diagrama de Blocos do cálculo dos Torques. . . . .	38
12	Erro de posição em malha aberta . . . . .	39
13	Comparação Trajetória Desejada e Real, em Malha Aberta. . . . .	39
14	Erro de posição em malha fechada. . . . .	40
15	Comparação Trajetória Desejada e Real, em Malha Fechada. . . . .	41



# Lista de Tabelas

1	Pontos-chave da Trajetória. . . . .	28
2	Parâmetros de Denavit-Hartenberg . . . . .	33
3	Pontos-chave no espaço cartesiano e das juntas . . . . .	34

# Lista de Algoritmos

1	Cálculo de Matriz de Inércia - Matriz D . . . . .	30
2	Cálculo da Matriz de Força Centrípeta e de Coriolis - Matriz C . . . . .	30
3	Cálculo de Matriz de Energia Potencial . . . . .	31

# Lista de Símbolos

$z_i$	Eixo de rotação de uma junta de revolução
$\theta_i$	Ângulo de rotação de uma junta de revolução
$\alpha_i$	Ângulo entre dois planos de coordenada ao redor de $x$
$J$	Jacobiano Geométrico
$I$	Matriz Identidade
$\mathbf{I}$	Tensor de Inércia
$\rho$	Densidade de massa
$q$	Vetor de posições
$R$	Matriz de rotação
$d$	Matriz de deslocamento
$A$	Matriz de transformação homogênea entre par de elos.
$H$	Matriz de transformação homogênea da base ao efetuador final
$T_i^j$	Matriz de transformação homogênea de um elo "i" a um elo "j".
$\dot{q}$	Vetor de velocidades
$\ddot{q}$	Vetor de acelerações
$o_i x_i y_i z_i$	Plano cartesiano ligado ao elo $i$

# Conteúdo

<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>Lista de Símbolos</b>	<b>vii</b>
<b>1 Introdução</b>	<b>11</b>
1.1 Justificativa . . . . .	13
1.2 Objetivos . . . . .	13
1.2.1 Objetivo Geral . . . . .	13
1.2.2 Objetivos Específicos . . . . .	14
1.3 Resultados esperados . . . . .	14
<b>2 Fundamentação Teórica</b>	<b>15</b>
2.1 Modelagem de Manipuladores Industriais . . . . .	16
2.1.1 Transformações Homogêneas e Translações . . . . .	16
2.1.2 Cinemática Direta . . . . .	17
2.1.3 Velocidade na Cinemática . . . . .	20
2.1.4 Tensores de Inércia . . . . .	22
2.1.5 Dinâmica . . . . .	23
2.2 Controle de Manipuladores Industriais . . . . .	25
2.2.1 Controle com múltiplas variáveis . . . . .	25
2.2.2 Arquitetura de Malha Fechada . . . . .	27
<b>3 Materiais e Métodos</b>	<b>28</b>
<b>4 Resultados e Discussões</b>	<b>33</b>
4.1 Modelagem do Manipulador Robótico . . . . .	33
4.2 Teste da Modelagem em Malha Aberta . . . . .	38
4.3 Teste da Modelagem em Malha Fechada . . . . .	40
<b>5 Conclusão</b>	<b>42</b>
5.1 Trabalhos Futuros . . . . .	42
<b>Referências bibliográficas</b>	<b>44</b>

# 1

## Introdução

De acordo com o RIA (*Robot Institute of America*), um robô é um "Manipulador reprogramável e multifuncional desenvolvido para mover materiais, partes, ferramentas ou dispositivos especializados através de trajetórias programadas variáveis para o desempenho de uma variedade de tarefas". (Spong et al., 2006, p.10)

Nos últimos anos, sistemas industriais e comerciais com alta eficiência e ótimo desempenho tem crescido utilizando da tecnologia de robôs. Grande número de pesquisas de controle e inúmeras aplicações de controle foram apresentadas durante os últimos anos, concentrado no controle de sistemas robóticos. O campo de robôs manipuladores é um dos campos com mais aplicações industriais, educacionais e médicas, pois funciona em circunstâncias imprevisíveis, perigosas e inóspitas que o ser humano não pode alcançar. Trabalhar em reatores químicos ou nucleares, por exemplo, é muito perigoso, e utilizar um robô em vez de um humano pode minimizar acidentes fatais. Portanto, pesquisas em modelagem e análise do manipuladores robóticos, assim como a aplicação de técnicas de controle, são muito importantes.

Para manipuladores industriais com peso e custo otimizados, os requisitos de desempenho só podem ser alcançados utilizando técnicas de inteligência computacional, pois é necessário um controle de movimento aprimorado que as técnicas tradicionais de controle não conseguem prover. O controle de movimento de robôs manipuladores industriais é uma tarefa que tem sido estudada por pesquisadores acadêmicos e industriais. Até certo ponto, o desenvolvimento de algoritmos de controle de movimento seguiram duas rotas distintas, uma por pesquisadores acadêmicos e outra por fabricantes de robôs.

A situação pode ser parcialmente explicada pelo fato de que os algoritmos de movimento usados na indústria às vezes são considerados segredos comerciais. Devido a difícil situação competitiva entre os fabricantes de robôs, os algoritmos são raramente publicados.

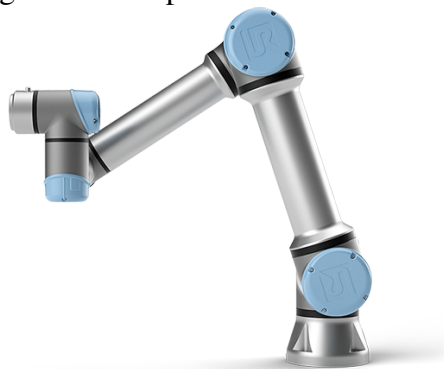
Outra explicação é que os pesquisadores de controle de robôs acadêmicos não possuem modelos físicos disponíveis, tendo que aplicar matemática avançada em alguns aspectos selecionados de sistemas relativamente pequenos, deixando parte de seus resultados num nível teórico

que não é de fácil adaptação para a indústria. Por outro lado, os pesquisadores e desenvolvedores de robôs industriais lidam com todos os aspectos significativos de um sistema complexo, contudo, a proposta matemática avançada frequentemente não pode ser aplicada. Pesquisas de robôs industriais e seu desenvolvimento devem equilibrar atividades de curto prazo e atividades de longo prazo. Colaborações de pesquisa a longo prazo entre a indústria e a academia devem ser possíveis, desde que a situação de propriedade intelectual seja tratada.

Como esse ramo é emergente, um vasto número de pesquisas vem sendo propostas nas últimas décadas no campo de manipuladores robóticos. Nos trabalhos de [Elgazzar \(1985\)](#) e [Anand \(1993\)](#), é feita uma análise cinemática do manipulador industrial PUMA 560, que é um robô de seis graus de liberdade. Nos trabalhos de [Jia et al. \(2012\)](#) e [Rashidifar et al. \(2013\)](#), são propostas modelagens e simulações utilizando um robô SCARA, que é um manipulador desenvolvido para fins educacionais. [Anand \(1993\)](#) deriva a cinemática do PUMA 560, e calcula as equações de movimento utilizando as equações de Euler-Lagrange. [Elgazzar \(1985\)](#) apresentou soluções eficientes para cinemática do PUMA 560. Sua solução é baseada num método que explora a geometria especial desse robô para otimizar as equações, demonstrando que é possível economizar uma quantidade de tempo computacional significativa para o controle em tempo real.

Este trabalho foca no processo de modelagem e design de um controlador de movimento para um manipulador robótico que atenda a requisitos impostos pela trajetória sugerida. O Manipulador Industrial UR5, fabricado pela *Universal Robots*, foi escolhido como estudo de caso nessa tese por ser um robô extremamente presente no mercado, possuir modernas tecnologias que o permitem atuar num ambiente colaborativo com humanos e uma característica importante desse manipulador é o fato da arquitetura de controle ser fechada, isto é, só é fornecido acesso ao usuário através de programas de alto nível como o *design* de trajetórias, sendo impossível criar aplicações utilizando informações de baixo nível. Arquiteturas de controle fechadas são um problema comum ao analisar manipuladores industriais disponíveis comercialmente.

Figura 1: Manipulador Robótico UR5.



A estrutura do presente trabalho inicia-se por uma breve explicação sobre o campo da robótica e a familiarização com conceitos necessários para o entendimento do estudo. O capítulo

dois trata da fundamentação teórica, abordando as principais teorias e equações necessárias para sustentar os métodos utilizados no decorrer do trabalho. O capítulo três, apresenta os materiais e ferramentas necessários para a reprodução do trabalho assim como os métodos utilizados para alcançar os resultados. O capítulo 4 demonstra os resultados encontrados ao produzir o modelo proposto no trabalho e discute as suas implicações. Por fim, no capítulo 5 são feitas as considerações finais sobre o trabalho e apresentadas algumas limitações e propostas de trabalhos futuros.

O controle de movimentação de robôs é uma competência chave para os fabricantes, e o desenvolvimento atual está focado em aumentar o desempenho do robô, reduzindo custos, melhorando a segurança e introduzindo novas funcionalidades conforme descrito em Brogaardh (2007), há necessidade de melhorar continuamente os modelos e métodos de controle a fim de cumprir todos os requisitos conflitantes, por exemplo, maior desempenho para um robô de peso menor, portanto menor rigidez mecânica.

## 1.1 Justificativa

Um robô industrial é uma máquina de uso geral para automação industrial, e apesar de que os requisitos de uma determinada aplicação possam ser formulados com precisão, não há limites para o que os usuários do robô desejam com relação ao desempenho e à funcionalidade desejáveis do controle de movimento de um robô. O controle de movimentação depende da aplicação. Quanto melhor o desempenho, mais aplicações estarão sujeitas a serem automatizadas por um modelo de robô específico.

Este trabalho foca no processo de modelagem e design de um controlador de movimento para um manipulador robótico que atenda a requisitos impostos pela trajetória sugerida. O Manipulador Industrial UR5, fabricado pela *Universal Robots*, é utilizado como estudo de caso nessa tese. Uma característica importante desse manipulador é o fato da arquitetura de controle ser fechada, isto é, só é fornecido acesso ao usuário através de programas de alto nível como o *design* de trajetórias, sendo impossível criar aplicações utilizando informações de baixo nível. Arquiteturas de controle fechadas são um problema comum ao analisar manipuladores industriais disponíveis comercialmente.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Este trabalho tem como objetivo criar um Modelo Matemático para o manipulador robótico UR5, apresentando todo equacionamento de cinemática e dinâmica, de maneira a facilitar a implementação de tecnologias avançadas e que pesquisadores e desenvolvedores possam criar

suas próprias funcionalidades com robô UR5 sem estar atreladas às limitações do Software do fabricante.

### **1.2.2 Objetivos Específicos**

Para alcançar o objetivo principal de produzir um Modelo Matemático para o Manipulador, será preciso alcançar outros itens que irão servir como base para a modelagem.

Será construído uma representação matemática da arquitetura do robô que será fundamental o processo de criação de trajetórias e também para fazer conversões entre o plano cartesiano e o plano das juntas, que é o plano onde todos os cálculos são feitos.

Um Controlador Proporcional e Derivativo será construído a fim de estabilizar o sistema robótico que é instável, também é almejado construir dois ambientes de teste para o sistema robótico, um em malha aberta que irá representar o robô sem nenhum acréscimo ao sistema e um em malha fechada contendo o controlador calculado a fim de estabilizar o sistema. Serão produzidas simulações numa trajetória pré-estabelecida em ambos ambientes de testes, que irão servir para testar a modelagem e produzir dados para a validação da solução através da comparação dos resultados.

## **1.3 Resultados esperados**

Nesse trabalho é esperado reduzir a distância entre trabalhos produzidos academicamente e sua contra-parte no meio industrial, através da construção do modelo matemático do manipulador Robótico UR5. Nesse modelo serão considerados todos os processos envolvendo desde a formação da rota do robô, até o desenvolvimento final do controlador para executar uma tarefa de exemplo em respeito a critérios de desempenho pré-estabelecidos.

É esperado que essa iniciativa de redução da distância entre indústria e academia possibilite a implementação de tarefas cada vez mais complexas com a tecnologia já disponível no mercado, através da disponibilização de modelos matemáticos que podem ser utilizados para teoricamente qualquer tarefa aplicando um método de controle adequado.

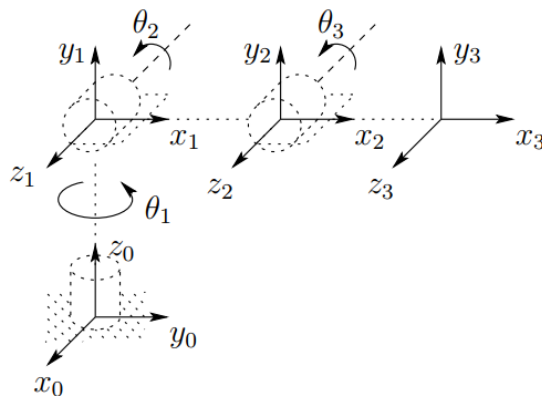


# 2

## Fundamentação Teórica

Manipuladores industriais são robôs com um braço mecânico operando junto de um controlador computadorizado. São compostos de elos que são conectados por juntas formando uma cadeia cinética. O Manipulador nessa tese é puramente rotacional, isto é, é formado apenas por juntas rotacionais. Cada junta representa a interconexão entre dois elos. O eixo de rotação de uma junta rotacional, denominado  $z_i$ , é interconexão entre os elos  $z_i$  e  $z_{i-1}$ . As variáveis de juntas, denominadas  $q_i$ , representam o deslocamento relativo entre elos adjacentes. Nesse estudo de caso as juntas são de rotação, portanto, é válido que  $q_i = \theta_i, \forall i \in \{1, \dots, n\}$ , onde  $\theta_i$  denota uma rotação relativa. Ambas notações são utilizadas durante a tese, uma representação sobre rotação relativa é demonstrada na Figura 2, onde cada junta permite um grau de liberdade de movimento entre elos adjacentes do manipulador. A junta de revolução produz uma rotação relativa entre elos adjacentes.

Figura 2: Representação simbólica das juntas de revolução.



Fonte: [Spong et al. \(2006\)](#)

A especificação da localização de ponto no manipulador é denominada configuração do robô. O conjunto de todas as configurações possíveis é chamado de espaço de configurações.

A base da cinemática dos manipuladores assume que são fixos e que seus elos são rígidos, portanto, é suficiente definir os valores de configuração de um robô apenas utilizando suas variáveis de junta, podendo ser agrupadas num vetor  $q = [q_1, \dots, q_n]$ . A velocidade das juntas pode ser representadas através de  $\dot{q} = [\dot{q}_1, \dots, \dot{q}_n]$ .

## 2.1 Modelagem de Manipuladores Industriais

Essa seção aborda os principais conceitos de modelagem física de um manipulador robótico, assim como apresenta formulações, técnicas e tecnologias utilizadas para embasar a formalização do modelo cinético e dinâmico.

A derivação das equações de cinemática direta, de velocidade e dinâmica são compostos de processos típicos em manipuladores industriais, e sua teoria é bem definida e estabelecida. A modelagem de manipuladores é geralmente simplificada em procedimentos passo-a-passo, através de movimentações rígidas (translações) e transformações homogêneas que são definidas para representar posições e orientações de objetos rígidos assim como entre sistemas de coordenadas relacionados. Um método chamado de convenção de Denavit-Hartenberg (DH) foi desenvolvido para padronizar o processo de atribuição dos sistemas de coordenadas aos elos e juntas do manipulador, assim como para criar matrizes de transformação homogêneas. Através dessas matrizes, se torna fácil derivar a cinemática direta. Para obter a cinemática de velocidade, um Jacobiano é definido especialmente para manipuladores robóticos, também chamado de manipulador Jacobiano. Utilizando essa definição de Jacobiano junto das equações de Euler-Lagrange, é possível derivar as equações de dinâmica do manipulador.

Apesar de todas essas simplificações, é uma tarefa desafiadora modelar um manipulador industrial visto que é requerido conhecimento específico sobre os parâmetros do manipulador. Esses parâmetros incluem dimensões do robô, peso e tensores de inércia das juntas e elos. Dependendo do fabricante esses parâmetros podem ser disponíveis abertamente, parcialmente ou até mantidos em segredo, sendo também difíceis de estimar. Conseqüentemente, as equações de dinâmica resultantes consistem de grandes matrizes que não podem ser verificadas analiticamente, apenas através de simulações e experimentos. A possibilidade de verificar o modelo de um manipulador depende do número de sensores e a quantia de grandezas disponíveis para consulta, o que varia entre diferentes fabricantes e manipuladores.

### 2.1.1 Transformações Homogêneas e Translações

Movimentações Rígidas e Transformações Homogêneas são utilizadas para descrever posições relativas e orientações entre sistemas de coordenadas atribuídos para cada junta e seu respectivo elo.

Transformações homogêneas combinam operações de rotação e translação numa única multiplicação de matrizes que é tipicamente utilizada para derivar a cinemática direta de um mani-

pulador robótico e calcular transformações de coordenadas.

Movimentações rígidas são definidas como um par ordenado  $(d, R)$  onde  $d \in \mathbb{R}^3$  é um vetor de translação e  $R \in \mathbb{R}^{3 \times 3}$  é uma matriz rotacional de ordem três.

Matrizes de rotação podem ser utilizadas para representar um plano de coordenadas em relação a outro assim como transformar um ponto de um plano de coordenada para outro plano diferente. Rotações sucessivas como de um plano de coordenadas  $o_i x_i y_i z_i$  para um plano  $o_j x_j y_j z_j$  e por fim a um  $o_k x_k y_k z_k$  podem ser obtidas através de:

$$R_k^i = R_j^i \cdot R_k^j \quad (1)$$

Um vetor apontando para um ponto  $p$  num plano de coordenadas  $o_i x_i y_i z_i$  é denominado  $p^i$ .

Transformações homogêneas simplificam o processo de lidar com grandes sequências de deslocamentos, visto que a composição é reduzida para uma multiplicação de matrizes.

Uma matriz de transformação homogênea  $H \in \mathbb{R}^{4 \times 4}$  possui a forma:

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, R \in \mathbb{R}^{3 \times 3}, d \in \mathbb{R}^3 \quad (2)$$

Para calcular as próximas transformações, as matrizes de transformação homogênea devem ser multiplicadas de acordo com:

$$H_j^i = H_{i+1}^i \cdot \dots \cdot H_j^{j-1} \quad (3)$$

A componente rotacional pode ser obtida através de:

$$R_j^i = R_{i+1}^i \cdot \dots \cdot R_j^{j-1} \quad (4)$$

E as componentes de deslocamento através de:

$$d_j^i = d_{j-1}^i + R_{j-1}^i \cdot d_j^{j-1} \quad (5)$$

### 2.1.2 Cinemática Direta

Cinemática descreve a movimentação do manipulador sem considerar as forças e torques que causam a movimento. A cinemática direta é usada para determinar a posição e orientação do efetuador final através dos estados das variáveis de junta.

Para derivar a cinemática direta é necessário atribuir um sistema de coordenadas para cada junta e elo que está associado a ela. A convenção de DH junto de outras definições auxiliam esse processo.

Um manipulador robótico com  $n$  juntas tem  $n + 1$  elos, onde a numeração para as juntas é dada por  $j_i \in \{1, \dots, n\}$  e para os elos  $l_i \in \{0, \dots, n\}$ , iniciando da base onde o manipulador está

fixo. Nessa convenção, a junta  $j_i$  conecta o elo  $l_{i-1}$  ao elo  $l_i$ .

A posição da junta  $j_i$  é considerada fixada com respeito ao elo  $l_{i-1}$ . O plano de coordenadas  $o_i x_i y_i z_i$  está fixo ao elo  $l_{i-1}$ . Quando a junta  $j_i$  é movimentada, o elo  $l_i$  e seu plano  $o_i x_i y_i z_i$  associado se movem em conjunto.

O plano de coordenadas  $o_0 x_0 y_0 z_0$ , que é afixado a base do robô é referido como plano de coordenadas referencial ou plano de coordenadas inercial.

Transformações Homogêneas são cruciais para descrever a cinemática.  $A_i$  é a matriz homogênea que fornece a posição e orientação de  $o_i x_i y_i z_i$  com respeito a  $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$ . Utilizando as equações baseadas em [Spong et al. \(2006\)](#):

$$A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (6)$$

$T_j^i$  é a matriz de transformação homogênea que expressa a posição e orientação de  $o_j x_j y_j z_j$  com respeito a  $o_i x_i y_i z_i$ .  $T_j^i$  é derivado da multiplicação das matrizes de transformação  $A_i$ :

$$T_j^i = \begin{cases} A_{i+1} A_{i+2} \dots A_{j-1} A_j, & \text{if } i < j \\ I, & \text{Se } i = j \\ (T_j^i)^{-1}, & \text{Se } i > j \end{cases} \quad (7)$$

A posição e orientação do orientador final em relação ao plano de coordenadas inercial é dado por:

$$T_n^0 = A_1(q_1) \dots A_n(q_n) = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} \quad (8)$$

### Convenção de Denavit-Hartenberg

A Transformação homogênea é utilizada para descrever a relação entre o plano de coordenadas  $o_i x_i y_i z_i$ , para  $i \in \{0, \dots, n\}$  que são atribuídos aos elos e juntas do robô. Para definir uma movimentação rígida, seis parâmetros são necessários: três parâmetros para definir a rotação e três parâmetros para definir a translação. A convenção de DH reduz o número de parâmetros que são necessários para definir uma transformação homogênea de seis para quatro, se aproveitando da geometria em comum dos manipuladores e fazendo escolhas inteligentes para a origem e os eixos de coordenada. A convenção de DH é o procedimento padrão usado na derivação da cinemática direta de manipuladores industriais.

Na convenção de DH cada transformação homogênea  $A_i$  é um produto de quatro transformações básicas. De acordo com [Spong et al. \(2006\)](#):

$$A_i = \text{Rot}_{z,\theta_i} \cdot \text{Trans}_{z,d_i} \cdot \text{Trans}_{x,a_i} \cdot \text{Rot}_{x,\alpha_i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$\text{Rot}_{z,\theta_i} = \begin{bmatrix} c\alpha_i & c\alpha_i & 0 & 0 \\ c\alpha_i & c\alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$\text{Trans}_{z_{n-1}}(d_n) = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (11)$$

$$\text{Trans}_{x_n}(r_n) = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & r_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (12)$$

$$\text{Rot}_{x_n}(\alpha_n) = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_n & -\sin \alpha_n & 0 \\ 0 & \sin \alpha_n & \cos \alpha_n & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (13)$$

Onde os quatro parâmetros  $\theta_i$ ,  $a_i$ ,  $d_i$ ,  $\alpha_i$  são ângulo da junta, comprimento do elo, distância do elo e torção da junta  $j_i$  e elo  $l_i$ . Como  $A_i$  é uma função de  $\theta_i$ , os outros três parâmetros são constantes. De acordo com a convenção DH, os planos de coordenadas  $o_i x_i y_i z_i$  atribuídos a junta  $j_i$  e ao elo  $l_i$  devem possuir os seguintes atributos:

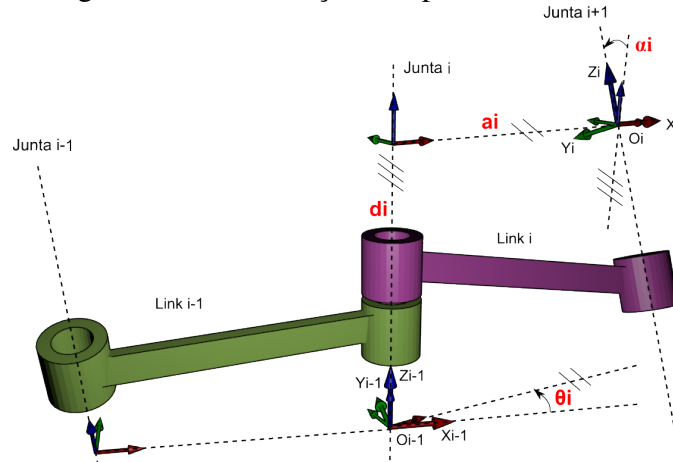
1. O Eixo  $z_i$  deve ser o eixo de revolução da junta  $j_{i+1}$ .
2. O Eixo  $x_i$  deve ser perpendicular aos eixos  $z_{i-1}$  e  $z_i$ .
3. O eixo  $x_i$  deve interceptar o eixo  $z_{i-1}$ .
4. Todos os planos de coordenadas devem seguir a regra da mão direita.

Após definir o plano de coordenadas, uma tabela com os parâmetros  $a_i$ ,  $d_i$ ,  $\alpha_i$  e  $\theta_i \forall i \in \{1, \dots, n\}$ , deve ser estabelecida de acordo com as seguintes regras:

1.  $a_i$  = Distância ao longo do eixo  $x_i$ , partindo da interseção entre  $x_i$  e  $z_{i-1}$  até  $o_i$ .
2.  $d_i$  = Distância ao longo do eixo  $z_{i-1}$ , partindo de  $o_{i-1}$  até a interseção entre  $x_i$  e  $z_{i-1}$ .

3.  $\alpha_i = \hat{\text{Ângulo de } z_{i-1} \text{ até } z_i \text{ medido ao redor de } x_i}$ .
4.  $\theta_i = \hat{\text{Ângulo de } x_{i-1} \text{ até } x_i \text{ medido ao redor de } z_{i-1}}$ . Como todas as juntas do robô UR5 são revolução,  $\theta_i$  é a única variável.

Figura 3: Demonstração dos parâmetros de DH.



Fonte: Adaptado de [Wikipédia \(2020\)](#)

A Figura 3 demonstra o uso dos parâmetros de DH com dois eixos de coordenadas de exemplo.

Um total de  $n$  matrizes de transformação homogêneas, nomeadas  $A_1, \dots, A_n$  são obtidas através da substituição dos parâmetros na Equação 9, que serão utilizadas para obter a cinemática direta.

### Cinemática Direta

Para obter a cinemática direta, as matrizes de transformação homogênea  $A_1, \dots, A_n$  devem ser inseridas na Equação 8. A matriz de transformação resultante descreve a transformação a partir da base até o efetuador final do manipulador, como uma função de variáveis  $q_i$ , concluindo o problema da cinemática direta.

### 2.1.3 Velocidade na Cinemática

A velocidade compreendida na cinemática se refere a relação entre as velocidades angulares e lineares do efetuador final com as velocidades das juntas. Para obtê-las, as equações de cinemática direta são usadas como base. A cinemática direta define uma função entre a posição e orientação do efetuador final com a posição das juntas. As relações de velocidade são determinadas através do Jacobiano dessa função.

O Manipulador Jacobiano  $J \in \mathbb{R}^{6 \times n}$  não é usado exclusivamente para derivação das equações de velocidade, mas também em outras tarefas de manipulação robótica, por exemplo, planejamento e execução de trajetórias suaves, determinação de posições singulares e também é utilizado nas equações de dinâmica de movimentação.

Representa a transformação instantânea entre velocidades das juntas  $\dot{q} \in \mathbb{R}^n$  e as velocidades linear e angular do efetuador final.  $J$  é também identificado como Jacobiano Geométrico.

O Jacobiano Geométrico consiste nas contribuições de velocidade de cada junta na velocidade linear e angular do efetuador final. É utilizado para derivar a dinâmica do manipulador, relacionando as forças do efetuador final  $F \in \mathbb{R}^6$  com os torques das juntas  $\tau \in \mathbb{R}^6$ .

### Jacobiano Geométrico

O Jacobiano Geométrico é calculado utilizando as transformações homogêneas  $A_1, \dots, A_n$ . Para o caso de um manipulador de  $n$  elos, a parcela angular  $J_\omega$  das velocidades é calculada a partir de:

$$J_\omega = [z_0^0, \dots, z_{n-1}^0] \quad (14)$$

onde:

$$z_{i-1}^0 = R_{i-1}^0 \cdot k \quad (15)$$

g sendo  $k = [0, 0, 1]^T$ . A parcela linear do Jacobiano  $J_v$  é dada por:

$$J_{v_i} = \frac{n!}{k!(n-k)!} = z_0^0 \times (o_n^0 - o_{i-1}^0) \quad (16)$$

O Jacobiano Geométrico é obtido a partir de:

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (17)$$

Onde as Equações 14 e 16 são aplicadas em 17.

### Singularidades

Uma singularidade ocorre quando a velocidade da junta no espaço da junta torna-se infinita para manter a velocidade no espaço cartesiano. A singularidade mostra onde a continuidade no espaço comum se rompe em relação ao espaço cartesiano. Uma singularidade ocorre sempre que  $\det(J) = 0$ . Quando o Jacobiano perde um *rank*, ou seja, uma linha se torna nula, um ponto de singularidade ocorre. Há dois tipos de singularidades, são as singularidades internas e de limites.

As singularidades de limites surgem tipicamente quando o braço robótico é esticado para seu tamanho máximo perdendo assim liberdade de se mover para qualquer uma direção. As

singularidades internas não costumam ser simples, seus pontos são obtidos pela resolução do determinante.

### 2.1.4 Tensores de Inércia

Para derivar as equações da dinâmica do manipulador é necessário conhecer o tensor de inércia de cada elo. Normalmente não são disponibilizados pelo fabricante, portanto, devem ser calculadas. O tensor de inércia tem a formulação geral dada por:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (18)$$

onde  $I_{xx}$ ,  $I_{yy}$  e  $I_{zz}$  são os principais momentos de inércia, e  $I_{xy} = I_{yx}$ ,  $I_{xz} = I_{zx}$  e  $I_{yz} = I_{zy}$  são os produtos de inércia. São calculados a partir de:

$$\begin{aligned} I_{xx} &= \int \int \int (y^2 + z^2) \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \\ I_{yy} &= \int \int \int (x^2 + z^2) \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \\ I_{zz} &= \int \int \int (x^2 + y^2) \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \end{aligned} \quad (19)$$

e

$$\begin{aligned} I_{xy} = I_{yx} &= - \int \int \int xy \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \\ I_{xz} = I_{zx} &= - \int \int \int xz \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \\ I_{yz} = I_{zy} &= - \int \int \int yz \cdot \rho(x, y, z) \cdot dx \cdot dy \cdot dz \end{aligned} \quad (20)$$

onde  $\rho(x, y, z)$  é a densidade de massa do objeto, representada por uma função de posição.

### Tensor de Inércia aproximado com cilindros de densidade constante

Para o seguinte caso, é assumido que a massa  $M$  de cada elo é conhecida. Os elos podem ser aproximados por cilindros com raio  $r$  e altura  $h$ . A densidade de massa é assumida ser uma constante, por exemplo,  $\rho(x, y, z) = \rho = \frac{M}{\pi r^2 h} = const.$ , então o tensor de inércia resultante ao redor do corpo anexo ao plano de coordenadas é dado por:

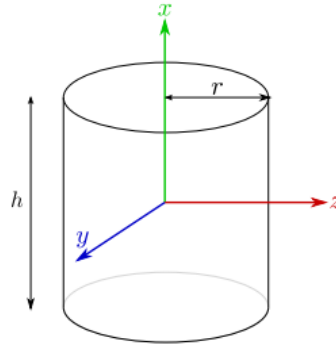
$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (21)$$



Fazendo com que o eixo de simetria do cilindro seja alinhado com o eixo  $x$ , conforme visualizado na Figura 4, os momentos de inércia são dados por:

$$\begin{aligned} I_{xy} &= \frac{1}{2}Mr^2 \\ I_{yy} = I_{zz} &= \frac{1}{12}M(3r^2 + h^2) \end{aligned} \quad (22)$$

Figura 4: Cilindro com altura  $h$  e raio  $r$ .



Fonte: (Spong et al., 2006, p. 63).

## 2.1.5 Dinâmica

A dinâmica de um robô descreve a relação entre forças, torques e movimento. Dois métodos são comumente usados para a derivação das equações de dinâmica: as equações de Euler-Lagrange e a formulação de Newton-Euler. O método de Euler-Lagrange é derivado do princípio de D'Alembert e do princípio do trabalho virtual, enquanto a formulação de Newton-Euler é a formação recursiva das equações de dinâmica geralmente utilizada para cálculos numéricos.

O método de Euler-Lagrange foi escolhido neste trabalho. Enquanto a derivação das equações dinâmicas para um manipulador n-link por meio das equações de Euler-Lagrange podem ser encontradas em Spong et al. (2006), apenas as equações cinéticas e de energia potencial substanciais e, eventualmente, as equações dinâmicas são apresentadas na seção seguinte. As equações são baseadas nas apresentadas por Ragazzon (2012).

### Energia Cinética para um Manipulador de n-Elos

Dado um corpo rígido com um plano de coordenadas anexado ao corpo, a energia cinética desse corpo é a soma da energia cinética translacional e rotacional. A energia translacional é obtida pela concentração de toda a massa do objeto no centro de massa, enquanto a energia rotacional é calculada sobre o centro de massa. A energia cinética total é então:

$$K = \frac{1}{2}mv^T v + \frac{1}{2}\omega^T J \omega \quad (23)$$

onde  $m \in \mathbb{R}$  é a massa total do objeto,  $v \in \mathbb{R}^3$  e  $\omega \in \mathbb{R}^3$  são os vetores de velocidade linear e angular, respectivamente, e  $\mathcal{J} \in \mathbb{R}^{3 \times 3}$  é o tensor de inércia expresso no referencial inercial. O tensor de inércia  $\mathcal{J}$  pode ser escrito como:

$$\mathcal{J} = RIR^T \quad (24)$$

onde  $R \in \mathbb{R}^{3 \times 3}$  é a matriz de rotação desde o referencial inercial até o manipulador, e  $I \in \mathbb{R}^{3 \times 3}$  é o tensor de inércia do elo. Dada a suposição de que cada elo  $l_i$ ,  $i \in \{1, \dots, n\}$  é um corpo rígido, a energia cinética do manipulador é igual a:

$$K = \frac{1}{2} \dot{q}^T \underbrace{\left[ \sum_{i=1}^n \{m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)\} \right]}_{D(q)} \dot{q} \quad (25)$$

onde  $m_i \in \mathbb{R}$  é a massa do elo  $l_i$  e onde  $J_{v_i} \in \mathbb{R}^{3 \times n}$  e  $J_{\omega_i} \in \mathbb{R}^{3 \times n}$  são as partes lineares e angulares da matriz Jacobiana da base ao centro de massa do elo  $l_i$  conforme descrito pela Equação 17.  $R_i \in \mathbb{R}^{3 \times 3}$  é a matriz de rotação base até o elo  $l_i$  e  $I_i \in \mathbb{R}^{3 \times 3}$  é o tensor de inércia expresso até o elo  $l_i$ .  $D(q) \in \mathbb{R}^{n \times n}$  é a matriz de inércia simétrica e positiva do manipulador.

### Energia Potencial para um manipulador de n-Elos

No caso de dinâmica rígida, a única fonte de energia potencial é a gravidade. A energia potencial do enésimo link pode ser calculada assumindo que a massa de todo o objeto é concentrado em seu centro de massa e é dada por:

$$P_i = m_i g_0^T r_i \quad (26)$$

onde  $m_i$  é a massa do elo  $l_i$ ,  $g_0$  é o vetor que dá a direção da gravidade no referencial inercial e o vetor  $r_i$  dá as coordenadas do centro de massa do elo  $i$ . A energia potencial total do robô de n-Elos é, portanto:

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i g_0^T r_i \quad (27)$$

### Equações de Movimento

As equações de movimento são comumente escritas em sua forma matricial como:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (28)$$

onde  $D \in \mathbb{R}^{n \times n}$  é a matriz de inércia,  $C \in \mathbb{R}^{n \times n}$  é a matriz forças de Coriolis e centrífuga,  $g \in \mathbb{R}^n$  é o vetor de gravidade,  $\tau \in \mathbb{R}^n$  é o vetor de torque,  $q \in \mathbb{R}^n$ ,  $\dot{q} \in \mathbb{R}^n$  e  $\ddot{q} \in \mathbb{R}^n$  são os vetores de posição, velocidade e aceleração, respectivamente. A matriz de inércia  $D(q)$  é

definida na Equação 25. O elemento  $(k, j)$  da matriz  $C(q, \dot{q})$  é definido como:

$$c_{kj} = \sum_{i=1}^n c_{ijk}(q) \dot{q}_i \quad (29)$$

onde  $c_{ijk}$  são os símbolos de Christoffel, dados por:

$$c_{ijk} := \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \quad (30)$$

e onde  $d_{ij}$  são valores da matriz de inércia. Finalmente, os valores do vetor de gravidade são definidos por:

$$g_k = \frac{\partial P}{\partial q_k} \quad (31)$$

O controle de manipuladores industriais e a problemática do controle fechado são discutidas na seção seguinte.

## 2.2 Controle de Manipuladores Industriais

Essa seção introduz o embasamento teórico necessário para a implementação de um controlador PD no espaço das juntas, utilizando a dinâmica inversa para representar o sistema. Demonstrando como derivar um controlador PD e como representar o sistema através de diagrama de blocos.

O controle de juntas independentes é usado para controlar a saída de torque com base na entrada de tensão. Ele usa um modelo do motor nas articulações e não leva a dinâmica do manipulador em consideração. Um método de controle comumente escolhido para o controle conjunto independente é o controlador PID. O controle multivariável controla a posição das articulações de todas as articulações  $q_i \in \{1, \dots, n\}$ , ou a posição do efector final e orientação  $X(q)$ , com base no torque  $\tau$  como uma entrada. O modelo dinâmico do manipulador é levado em consideração e a dinâmica inversa é usada para tratar a não linearidade. Um controlador PD é então usado para controlar o sistema linearizado.

### 2.2.1 Controle com múltiplas variáveis

Uma forma de controle multivariável é utilizada nessa tese, através do controle de trajetória através da realimentação linearizada.

#### Dinâmica inversa no Espaço das Juntas

O método de controle para esse robô é baseado nas equações apresentadas por [Ragazzon \(2012\)](#), que produzem um controlador baseado na dinâmica inversa. Considerando as equações dinâmicas de um manipulador de n-elos conforme fornecido na Equação 28 e escolhendo o torque

como a entrada de controle  $u = \tau \in \mathbb{R}^n$ , então uma lei de controle realimentada não-linear no formato:

$$u = f(q, \dot{q}, t) \quad (32)$$

é procurada de modo que o sistema geral em malha fechada seja linear. No caso do dinâmica do manipulador dada pela Equação 28, o problema é simples e a solução é escolher o vetor de controle  $u$  de acordo com a equação:

$$u = D(q)a_q + C(q, \dot{q})\dot{q} + g(q) \quad (33)$$

Uma vez que a matriz de inércia  $D$  é inversível (Spong et al. (2006)), o sistema combinado dado pelas Equações 28, 32 e 33, se reduz a:

$$\ddot{q} = a_q \quad (34)$$

que representa um sistema integrador duplo e onde  $a_q$  é a nova entrada que deve ser escolhida. Uma escolha para  $a_q$  é um controlador PD com realimentação na aceleração, demonstrado por:

$$a_q = \ddot{q}_d(t) - K_0\tilde{q} - K_1\dot{\tilde{q}} \quad (35)$$

onde  $\tilde{q} = q - q_d$ ,  $\dot{\tilde{q}} = \dot{q} - \dot{q}_d$  e onde  $K_0, K_1$  são matrizes diagonais com elementos consistindo em ganhos de posição e velocidade, respectivamente. A trajetória de referência:

$$t \longrightarrow (q_d(t), \dot{q}_d(t), \ddot{q}_d(t)) \quad (36)$$

define o histórico de tempo desejado das posições, velocidades e acelerações das juntas. As matrizes de ganho  $K_0$  e  $K_1$  podem ser escolhidas como

$$K_0 = \begin{bmatrix} K_{0,1} & 0 & \dots & 0 \\ 0 & K_{0,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & K_{0,n} \end{bmatrix} \quad K_1 = \begin{bmatrix} K_{1,1} & 0 & \dots & 0 \\ 0 & K_{1,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & K_{1,n} \end{bmatrix} \quad (37)$$

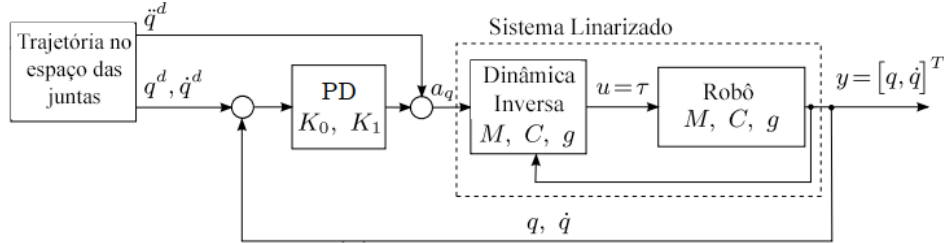
que resulta num sistema de segunda ordem desacoplado. Para estabilizar o sistema, basta escolher os parâmetros de forma que:

$$\begin{aligned} K_{0,i} &= \omega_{n,i}^2 \\ K_{1,i} &= 2\omega_{n,i} \end{aligned} \quad \forall i = 1, \dots, n \quad (38)$$

cada resposta das juntas será igual à resposta de um modelo linear de segunda ordem criticamente amortecido com frequência natural  $\omega_{n,i}$ . A Figura 5 visualiza o controle em malha

fechada.

Figura 5: Controle no Espaço das Juntas.



Fonte: Adaptado de (Spong et al., 2006, p. 259).

## 2.2.2 Arquitetura de Malha Fechada

### Estimativa de Parâmetros PD

Um primeiro palpite sobre o espaço das juntas e espaço de tarefa implementados é de que os controladores mais comuns como P, PD, PI e PID possam ser usados. Assumindo que os parâmetros PD são os únicos parâmetros desconhecidos na malha fechada, eles podem ser estimados usando o método dos mínimos quadrados não lineares.

### Mínimos quadrados não lineares

Considerando as equações de malha fechada:

$$\begin{aligned} D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) &= \tau \\ \tau &= D(q)a_q + C(q, \dot{q})\dot{q} + g(q) \\ a_q &= \ddot{q}_d - K_0(q - q_d) - K_1(\dot{q} - \dot{q}_d) \end{aligned} \quad (39)$$

e tendo  $K_0$  e  $K_1$  como matrizes desconhecidas.

Para estimar os parâmetros de  $K$ ,  $N$  medições  $y_m(t_j)$  são tomadas nos instantes de tempo  $t_j$ , para  $j = 1, \dots, N$ . É formado um problema de otimização onde deseja-se minimizar  $K$ .

$$\min(\mathcal{J}(K)) = \sum_{i=1}^n \sum_{j=1}^N (y_i(t_j, K) - y_{m,i}(t_j))^2 \quad (40)$$

Onde  $y(t_j, K)$  é a solução do sistema da Equação 39 e  $n$  é o número de medições. O vetor  $K$  que minimiza  $\mathcal{J}(K)$  é o estimador de Mínimos Quadrados não Lineares (NLSE) e é escrito como:

$$\hat{K} = \min(\mathcal{J}(K)) \quad (41)$$

# 3

## Materiais e Métodos

Inicialmente foi feita uma pesquisa no site do fabricante *Universal Robots* a respeito das informações sobre as dimensões do robô para a construção da modelagem de Denavit-Hartenberg, assim como informações físicas a respeito da massa e limitações do robô. Esses itens foram encontrados no Relatório Técnico 110105, disponível em [Robots \(2016\)](#).

A partir disso, foi calculada a matriz de Denavit-Hartenberg, seguindo as diretrizes apresentadas na Seção 2.1.2 por meio do *software* Matlab®, que oferece um ambiente de programação destinado a fazer cálculos com Matrizes.

Utilizando o Matlab® para facilitar os cálculos analíticos, foi possível obter as matrizes de transformação homogêneas para poder chegar a cinemática direta do robô com o auxílio da Matriz de Denavit-Hartenberg, conforme demonstrado na Equação 8.

Para a produção de uma simulação será construída uma trajetória para que o manipulador robótico possa seguir. A trajetória será composta dos pontos-chave que estão expostos na Tabela 1. O robô deve passar por todos os pontos-chaves para concluir a trajetória desejada.

Ponto-Chave	X(m)	Y(m)	Z(m)
A	0.642	0.088	0.175
B	0.344	0.473	0.175
C	0.344	0.473	0.030
D	0.344	0.473	0.175
E	0.344	-0.473	0.175
F	0.344	0.473	0.030

Tabela 1: Pontos-chave da Trajetória.

Utilizando a *toolbox* de Robótica ([Corke \(2011\)](#)) inclusa no Matlab®, foi possível a posição do conjunto de juntas do manipulador robótico para cada ponto-chave, com o auxílio da função *ikine*, que calcula uma solução para a inversa da cinemática iterativamente.

Figura 6: Demonstração da função *ikine*, disponibilizada na *toolbox* de Robótica.

```

32 - default_rotation = [1 0 0 ; 0 0 -1; 0 1 0];
33 - T_A = getTransformationMatrix(A, default_rotation);
34 - Q_A= Robo.ikine(T_A, 'verbose');
35 - T_B = getTransformationMatrix(B, default_rotation);
36 - Q_B= Robo.ikine(T_B, 'verbose');
37 - T_C = getTransformationMatrix(C, default_rotation);
38 - Q_C= Robo.ikine(T_C, 'verbose');
39 - T_D = getTransformationMatrix(D, default_rotation);
40 - Q_D= Robo.ikine(T_D, 'verbose');
41 - T_E = getTransformationMatrix(E, default_rotation);
42 - Q_E= Robo.ikine(T_E, 'verbose');
43 - T_F = getTransformationMatrix(F, default_rotation);
44 - Q_F= Robo.ikine(T_F, 'verbose');
45 - trajectory = [Q_A; Q_B; Q_C; Q_D; Q_E; Q_F];
<

```

Command Window

```

trajectory =

    0.4237    0.6177   -1.2833    0.6656    0.4237   -0.0000
    1.1836    0.8573   -1.2289    0.3717    1.1836   -0.0000
    1.1836    0.6841   -1.3255    0.6414    1.1836    0.0000
    1.1836    0.8573   -1.2289    0.3717    1.1836   -0.0000
   -0.6384    1.1428   -1.7010    0.5582   -0.6384   -0.0000
   -0.6384    0.9138   -1.7950    0.8811   -0.6384    0.0000

```

Fonte: Elaborada pelo autor.

Uma demonstração do uso da função é exibida na Figura 6. Após obter a configuração necessária das 6 juntas do manipulador robótico correspondentes aos pontos-chave foi feita regressão polinomial de quarto grau para obter a trajetória contendo os pontos-chave e pontos intermediários gerados pela regressão.

Com o uso das matrizes de transformação homogênea, as matrizes Jacobianas foram calculadas partindo da base até cada elo utilizando as Equações 14 a 17. As matrizes Jacobianas são utilizadas para encontrar os pontos de singularidade e também são necessárias durante o cálculo das equações de Dinâmica.

Resolvendo o determinante do Jacobiano, obtemos os pontos de singularidade do Manipulador. Esses pontos são utilizados para o planejamento da rota, de maneira a evitar passar por essas regiões de instabilidade.

Para poder dar início ao cálculo das equações de dinâmica, ainda resta ser determinado o tensor de inércia, a matriz de inércia  $D$ , a matriz  $C$  correspondendo ao efeito de Coriolis e a matriz  $G$ .

O Fabricante disponibiliza um modelo 3D na página de *Downloads* do produto, onde através de *softwares* de CAD é possível obter a matriz do tensor de inércia calculada automaticamente pelo *Software*.

Também foi feito a estimativa do tensor de inércia utilizando cilindros e os valores das massas e dimensões disponibilizados pelo fabricante usando as Equações 21 e 22.

Para o cálculo da Matriz de Inércia, os cálculos apresentados na Seção 2.1.4 foram consolidados no Algoritmo 1.

---

**Algoritmo 1** Cálculo de Matriz de Inércia - Matriz D

---

**Input:**  $m, T, J, I$ **Output:**  $D$  $gdl \leftarrow 6$  $D \leftarrow 0$  $Jw \leftarrow getAngularVelocity(J)$  $Jv \leftarrow getLinearVelocity(J)$  $R \leftarrow getRotations(T)$  $i \leftarrow 1$ **while**  $i \neq gdl$  **do** $D \leftarrow D + m(i) \cdot Jv(i)^T \cdot Jv(i) + Jw(i)^T \cdot R(i) \cdot I(i) \cdot R(i)^T \cdot Jw(i)$  $i \leftarrow i + 1$ **end while**

---

Dando continuidade aos parâmetros para as equações de dinâmica, para calcular a Matriz  $C$ , é necessário apenas a Matriz  $D$  obtida no algoritmo anterior. Aplicando a matriz  $D$  como entrada no Algoritmo 2, é possível obter a Matriz  $C$ .

---

**Algoritmo 2** Cálculo da Matriz de Força Centrípeta e de Coriolis - Matriz  $C$ 

---

**Input:**  $D$ **Output:**  $C$  $gdl \leftarrow 6$  $k \leftarrow 1$  $j \leftarrow 1$  $i \leftarrow 1$ **while**  $k \neq gdl$  **do****while**  $j \neq gdl$  **do****while**  $i \neq gdl$  **do** $C(j,k,i) \leftarrow diff(D(k,j),theta(i))/2 + diff(D(k,i),theta(j))/2 - diff(D(i,j),theta(k))/2$  $i \leftarrow i + 1$ **end while** $j \leftarrow j + 1$ **end while** $k \leftarrow k + 1$ **end while**

---

Calculando a última matriz necessária para equação da dinâmica, basta aplicar os conceitos de energia potencial das leis de newton apresentados na Seção 2.1.5. Aplicando as matrizes



de transformação homogênea para obter a altura de cada elo e também as massas, é possível conseguir a matriz  $G$  usando o Algoritmo 3.

---

**Algoritmo 3** Cálculo de Matriz de Energia Potencial

---

**Input:**  $m, T$

**Output:**  $G$

```
 $g \leftarrow 9.8$   
 $gdl \leftarrow 6$   
 $P \leftarrow 0$   
 $i \leftarrow 1$   
while  $i \neq gdl$  do  
   $P \leftarrow P + m(i) \cdot g \cdot Z(T(1))$   
   $i \leftarrow i + 1$   
end while  
 $i \leftarrow 1$   
while  $i \neq gdl$  do  
   $G(i) \leftarrow \text{diff}(P, \text{theta}(i))$   
   $i \leftarrow i + 1$   
end while
```

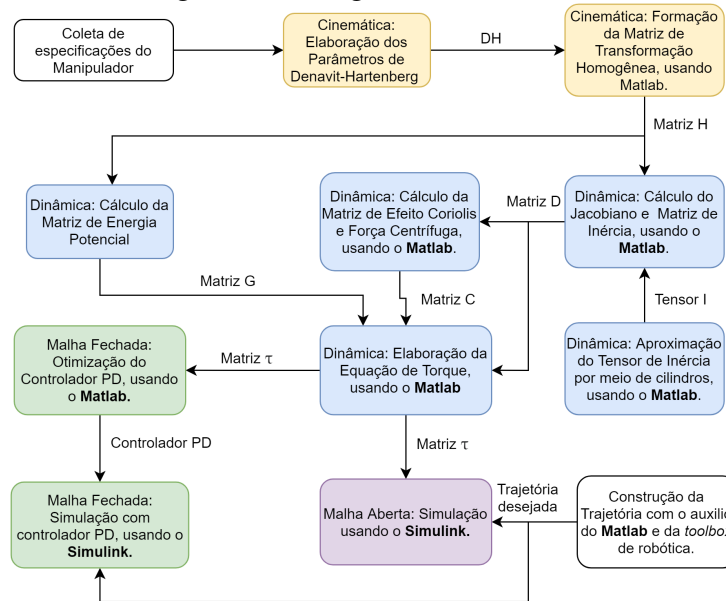
---

Ao consolidar os resultados na equação de dinâmica, já é possível criar o diagrama de blocos que será responsável por representar o sistema em malha fechada.

É possível obter valores de ganhos  $K_0$  e  $K_1$  aplicando o método dos Mínimos Quadrados não Lineares, conforme descrito em [Bosse et al. \(2016\)](#), na Equação 40. Por conta de ser um problema de otimização com um número indefinido de soluções, é possível utilizá-lo até obter ganhos que satisfaçam o desempenho pedido.

Utilizando a ferramenta Simulink®, embutida no pacote do Matlab®, é possível analisar o sistema em diferentes domínios e realizar simulações. Ao modelar o sistema seguindo o esquema da Figura 5, é possível realizar as simulações em malha fechada com um controlador PD com os parâmetros previamente calculados.

Figura 7: Fluxograma de Trabalho.



Fonte: Elaborada pelo autor.

A Figura 7 mostra uma visão geral da metodologia adotada nesse projeto para alcançar os resultados. Iniciando o processo através da coleta de especificações até a obtenção do modelo e ao estágio final que consiste nas simulações em malha aberta e malha fechada, obtidas utilizando o Matlab®, através da equação de torque calculada e pelo controlador otimizado para o robô.



## Resultados e Discussões

Neste capítulo serão apresentados os resultados e discussões a cerca do desenvolvimento deste trabalho, seguindo cada etapa apresentada no fluxograma da Figura 7, assim como uma discussão ao fim desse capítulo comparando o desempenho do modelo desenvolvido em malha aberta e malha fechada, para comprovar a eficácia do controlador. Os resultados estão divididos em três seções, a Seção 4.1 que apresenta os resultados obtidos durante a modelagem matemática do manipulador industrial, elaboração da Trajetória e otimização do controlador PD para a trajetória, a Seção 4.2 apresenta os resultados obtidos durante a execução da Simulação em Malha aberta e a Seção 4.3 apresenta os resultados obtidos da simulação em malha fechada, assim como um comparativo com a malha aberta.

### 4.1 Modelagem do Manipulador Robótico

Nesta seção serão apresentados os resultados obtidos durante o processo de modelagem do manipulador robótico, elaboração da trajetória que será usada para as simulações e a otimização do Manipulador PD.

	$q_i[rad]$	$d_i[m]$	$a_i[m]$	$\alpha_i[rad]$
Junta 1	$q_1$	0.089159	0	$\frac{\pi}{2}$
Junta 2	$q_2$	0	-0.425	0
Junta 3	$q_3$	0	-0.39225	0
Junta 4	$q_4$	0.10915	0	$\frac{\pi}{2}$
Junta 5	$q_5$	0.09465	0	$-\frac{\pi}{2}$
Junta 6	$q_6$	0.0823	0	0

Tabela 2: Parâmetros de Denavit-Hartenberg

A Tabela 2 apresenta os Parâmetros de Denavit-Hartenberg, que foram obtidos através da modelagem espacial do Robô para cada junta rotacional do manipulador.

Os Parâmetros de Denavit-Hartenberg foram projetados de maneira com que todo eixo  $x$  possua a mesma direção e sentido, garantindo que o máximo de campos  $a$  sejam nulos, obtendo assim uma expressão menos complexa. Os parâmetros serão importantes para estabelecer as matrizes de Transformação Homogênea para cada elo, e criar uma modelagem universal para representar o esquema do manipulador.

$$\begin{aligned}
 A_1 &= \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0.0892 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2 &= \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0.00000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_3 &= \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0.0000 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_4 &= \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0.10930 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_5 &= \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0.0947 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_6 &= \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0.08250 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{42}$$

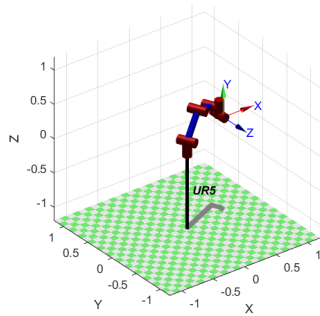
A Equação 42 apresenta as matrizes de transformação Homogêneas do robô. Cada Matriz  $A_i$  indica a operação para levar coordenadas do plano da junta  $i$  até o plano cartesiano relativo a essa junta. As Matrizes de Transformação Homogêneas foram obtidas a partir da Equação 9. Essas matrizes serão fundamentais para a definição da Matriz de Inércia, que é necessária para a produção do cálculo da equação da dinâmica do Robô.

	Espaço 3D	Espaço das Juntas
1	(0.642, 0.088, 0.175)	[0, -1.5708, 1.5708, 0, 0, 0]
2	(0.3924, -0.191, -0.4288)	[0, -1.5708, 1.5708, 0, 0, 0]
3	(0.3924, -0.191, -0.4288)	[0, -1.5708, 1.5708, 0, 0, 0]
4	(0.3924, -0.191, -0.4288)	[0, -1.5708, 1.5708, 0, 0, 0]
5	(0.3924, -0.191, -0.4288)	[0, -1.5708, 1.5708, 0, 0, 0]
6	(0.3924, -0.191, -0.4288)	[0, -1.5708, 1.5708, 0, 0, 0]

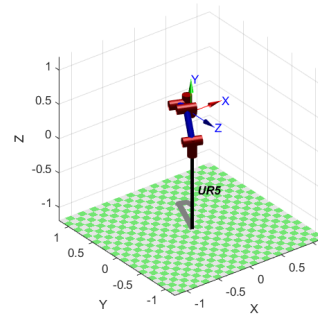
Tabela 3: Pontos-chave no espaço cartesiano e das juntas

A Tabela 3 apresenta a conversão dos pontos-chave da trajetória estabelecida na Tabela 1 no plano cartesiano traduzidos para o espaço das juntas, onde cada valor presente na 3-upla do Espaço 3D corresponde a coordenada  $x$ ,  $y$  e  $z$ , respectivamente. Cada linha possui em sua coluna Espaço das Juntas uma 6-upla representando a configuração no espaço das juntas, isto é, o valor de posição de cada junta necessário para alcançar a posição desejada no plano cartesiano 3D.

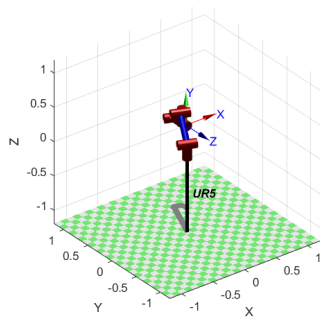
As configurações no espaço das juntas foram obtidas através da função *ikine* da toolbox de Robótica do Matlab®. Esses pontos-chave representam posições que o manipulador deve passar para concluir a trajetória, com configurações já no plano das juntas é possível estabelecer a trajetória interpolando as posições das juntas.



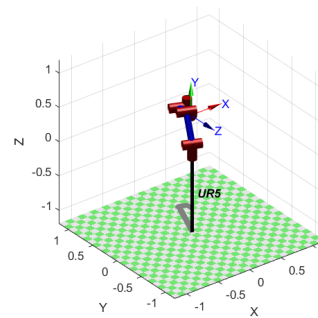
Instante Chave 1



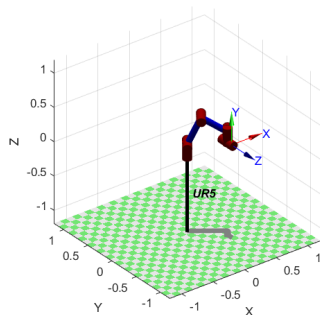
Instante Chave 2



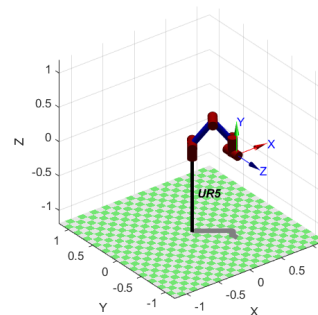
Instante Chave 3



Instante Chave 4



Instante Chave 5



Instante Chave 6

Figura 8: Pontos-chave da trajetória aplicados no robô

A Figura 8 apresenta seis gráficos correspondentes as configurações obtidas na Cinemática Inversa, expostas anteriormente na Tabela 3. Essa figura serve para ilustrar como funciona a conversão de pontos no plano cartesiano para uma configuração no manipulador robótico, isto é, como cada posição do efetuador final é traduzida nas posições em cada uma de suas juntas.

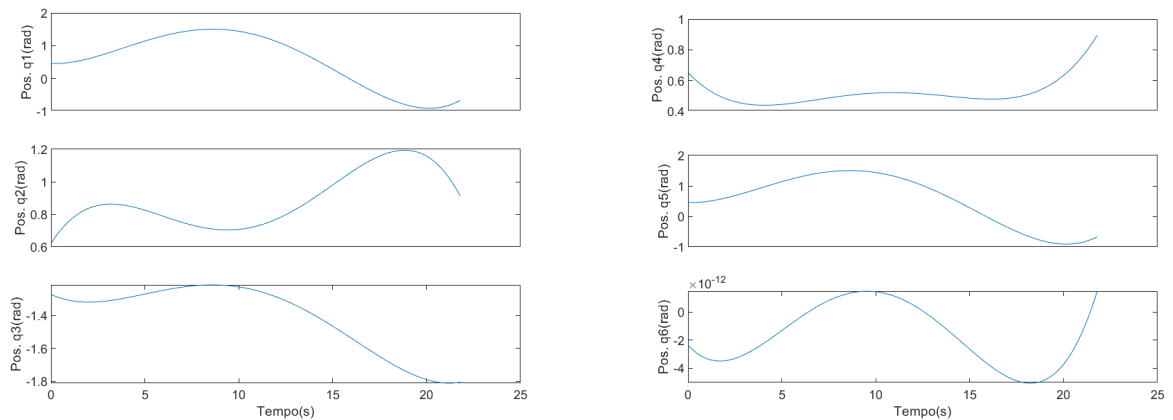


Figura 9: Posições da trajetória - Juntas 1 a 6.

A Figura 9 apresenta seis gráficos de posição em relação ao tempo, onde cada um representa o resultado da interpolação polinomial das configurações expostas na Tabela 3.

Esse conjunto compõe a trajetória do Robô, onde cada junta deve seguir sua trajetória esperada para alcançar os pontos-chave estabelecidos. Esses resultados serão utilizados para alimentar as simulações em Malha Aberta e Fechada no trabalho, assim como serão utilizados como referência para extrair o erro de posição com a trajetória obtida durante a simulação.

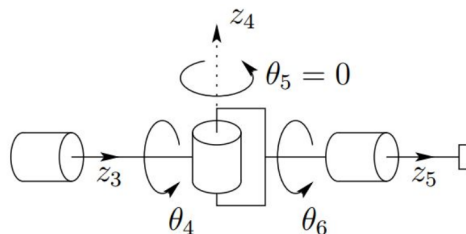


Figura 10: Exemplo de singularidade interna.

A Figura 10 apresenta uma das singularidades encontradas na resolução do determinante da equação 17, isto é, uma situação em que a ação de uma junta pode ser cancelada pela outra, deixando em aberto infinitas configurações que levam a uma mesma posição do manipulador final.

Conforme descrito na Seção 2.1.3, as singularidades representam pontos de indefinição para o projeto do Manipulador, isto é, são configurações que o Manipulador deve evitar em sua trajetória para que não encontre indefinições matemáticas durante suas soluções de posição. É de fundamental importância estabelecer as singularidades para que os usuários desse modelo matemático levem em conta durante o processo de elaboração da trajetória. Um desses pontos de singularidade é mostrado na Figura 10, que ocorre quando  $\theta_5 = 0$ , fazendo com que as juntas 4 e 6 se tornem paralelas.

$$\begin{aligned}
I_1 &= \begin{bmatrix} 0.0067 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} & I_2 &= \begin{bmatrix} 0.0149 & 0 & 0 \\ 0 & 0.3564 & 0 \\ 0 & 0 & 0.3553 \end{bmatrix} \\
I_3 &= \begin{bmatrix} 0.0025 & 0 & 0.0034 \\ 0 & 0.0551 & 0 \\ 0.0034 & 0 & 0.0546 \end{bmatrix} & I_4 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \\
I_5 &= \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} & I_6 &= \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}
\end{aligned} \tag{43}$$

A Equação 43 mostra os tensores de inércia obtidos através do modelo 3D fornecido pelo fabricante. Cada tensor corresponde a seu respectivo elo do manipulador robótico em forma matricial, e cada matriz é composta pelos momentos de inércia em relação a cada par de eixos do plano cartesiano, conforme demonstrado na Seção 2.1.4.

Os tensores de inércia aproximados na Equação 43 foram extraídos do Modelo 3D, são de alta fidelidade com o manipulador robótico real, contendo detalhes como massa não uniforme e também a presença de momentos de inércia fora da diagonal principal da matriz, indicando que os corpos não são simétricos. Os Tensores de Inércia são fundamentais para a determinação da Matriz de Inércia do Manipulador Robótico, utilizando o Algoritmo 1.

$$\begin{aligned}
I_1 &= \begin{bmatrix} 0.0084 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0084 \end{bmatrix} & I_2 &= \begin{bmatrix} 0.0078 & 0 & 0 \\ 0 & 0.0021 & 0 \\ 0 & 0 & 0.0021 \end{bmatrix} \\
I_3 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0462 & 0 \\ 0 & 0 & 0.0462 \end{bmatrix} & I_4 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0016 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \\
I_5 &= \begin{bmatrix} 0.0016 & 0 & 0 \\ 0 & 0.0016 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} & I_6 &= \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}
\end{aligned} \tag{44}$$

De maneira similar a Figura 43, a Figura 44 também mostra os tensores de inércia do robô apresentados da mesma maneira, porém essa versão dos tensores de inércia foi obtida através de uma aproximação utilizando cilindros que considera simetria e uniformidade de massa, conforme a Seção 2.1.4.

A versão aproximada é de grande utilidade para o modelo, pois incluindo suposições de simetria foi possível obter apenas a diagonal principal, garantindo equações mais simples para representar a Matriz de Inércia. Apesar da versão aproximada trazer benefícios de desempenho, é recomendado utilizar a aproximação 3D em aplicações onde maior precisão são necessárias, para a trajetória de teste dessa aplicação a versão aproximada é suficiente.

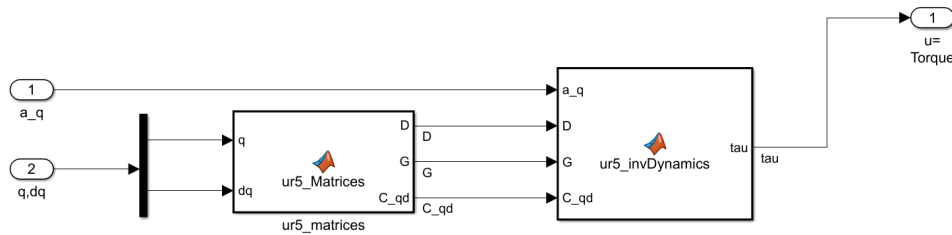


Figura 11: Diagrama de Blocos do cálculo dos Torques.

A Figura 11 mostra o diagrama de blocos utilizado para a obtenção dos torques de cada junta rotacional. O bloco *ur5\_matrices* tem como saída as Matrizes de Inércia, de Energia Potencial e de Efeito Coriolis, sendo identificadas por  $D$ ,  $G$ ,  $C_{qd}$ , respectivamente. O bloco *ur5\_invDynamics* faz o cálculo propriamente dito da Equação 28, unindo as matrizes calculadas e produzindo o torque do sistema.

A Matriz de Inércia representando as forças inerciais devido às acelerações das juntas foi computada utilizando o Algoritmo 1. A Matriz de Forças Centrífluga e de Coriolis são computadas pelo Algoritmo 2, representando rotações em relação ao plano de referência. A Matriz  $G$  simbolizando a energia potencial foi calculada utilizando o Algoritmo 3. O Diagrama de Blocos da Figura 11 recebe como entrada a trajetória desejada e computa o torque resultante em função do tempo. Será utilizado como sistema base para as simulações em malha aberta e também será utilizado para projetar um controlador e produzir simulações em malha fechada.

$$\begin{aligned} K_0 &= \begin{bmatrix} 379.6302 & 44.4089 & 104.6120 & 76.9813 & 76.9813 & 76.9813 \end{bmatrix} \\ K_1 &= \begin{bmatrix} 73.0732 & 45.1008 & 113.2602 & 77.8688 & 77.8688 & 77.8688 \end{bmatrix} \end{aligned} \quad (45)$$

A Equação 45 mostra os ganhos calculados para o controlador PD, conforme descrito na seção 2.2.1. Os ganhos  $K_0$  correspondem aos ganhos proporcionais e os ganhos  $K_1$  correspondem aos ganhos derivativos, conforme demonstrado na Figura 5.

Os ganhos representam os parâmetros necessários para a formação do controlador do sistema, que será responsável por estabilizar o sistema, tornando o erro o mais próximo de zero. O processo de obtenção dos parâmetros  $K_0$  e  $K_1$  foi feito através da execução do algoritmo de regressão de mínimos quadrados não lineares, conforme descrito em Bosse et al. (2016). O processo de otimização é iterativo, podendo encontrar soluções em tentativas até que, caso seja possível, encontrar uma que atenda aos pré-requisitos de desempenho.

## 4.2 Teste da Modelagem em Malha Aberta

Nesta seção serão apresentados os resultados obtidos durante o processo de teste da Modelagem em Malha Aberta, isto é, sem a presença de um controlador ou realimentação durante sua



execução.

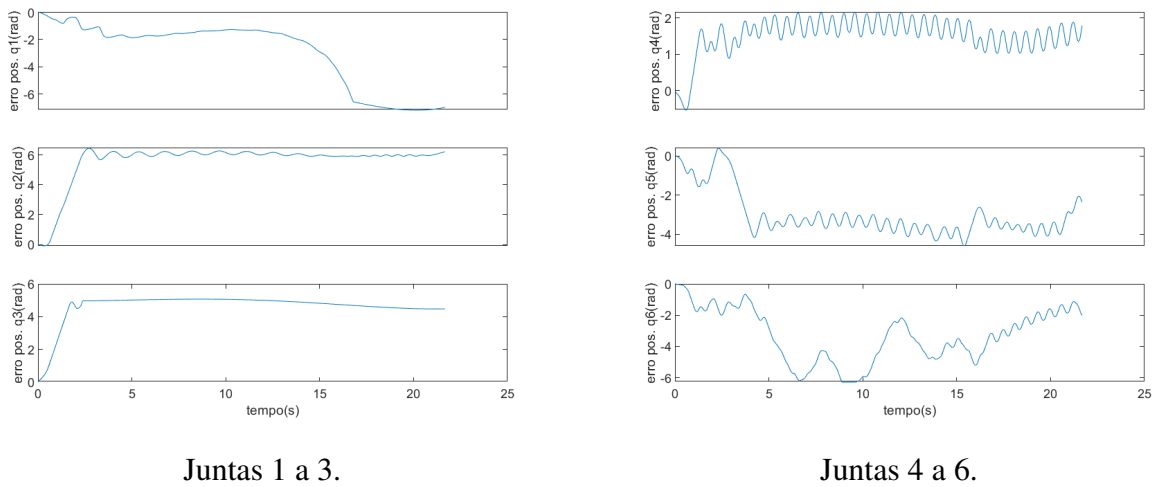


Figura 12: Erro de posição em malha aberta

A Figura 12 apresenta 6 gráficos contendo os erros absolutos de posição obtidos fazendo a diferença entre a trajetória esperada e a obtida, que é encontrada a partir do toque resultante no esquema em malha aberta exposto na Figura 11, aplicando a inversa da Equação 28, obtendo assim a aceleração, velocidade e posição através do torque.

Os erros de posição obtidos durante a malha aberta demonstram a natureza instável do manipulador, apresentando picos de desvio da trajetória esperada ultrapassando 6 radianos, conforme a Figura 12. Como cada junta possui erros altos, fica claro que o manipulador robótico não é capaz de obter um funcionamento satisfatório sem um sistema de controle.

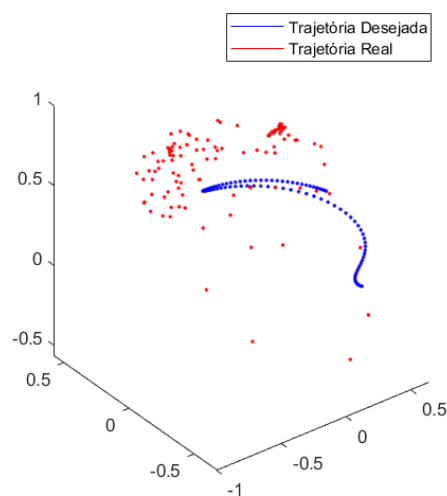


Figura 13: Comparação Trajetória Desejada e Real, em Malha Aberta.

A Figura 13 representa uma comparação da trajetória resultante da execução da Malha

Aberta com a trajetória desejada. Por conta dos erros de posição individuais em cada junta, a trajetória do efetuador final recebe um acúmulo de erros se tornando muito distante da trajetória prevista.

### 4.3 Teste da Modelagem em Malha Fechada

Nesta seção serão apresentados os resultados obtidos durante o processo de teste da Modelagem em Malha Fechada, isto é, utilizando realimentação e um controlador para tornar o sistema estável, com os ganhos expostos na Equação 45. Também será feito comparativos com os testes em Malha Aberta e discussões.

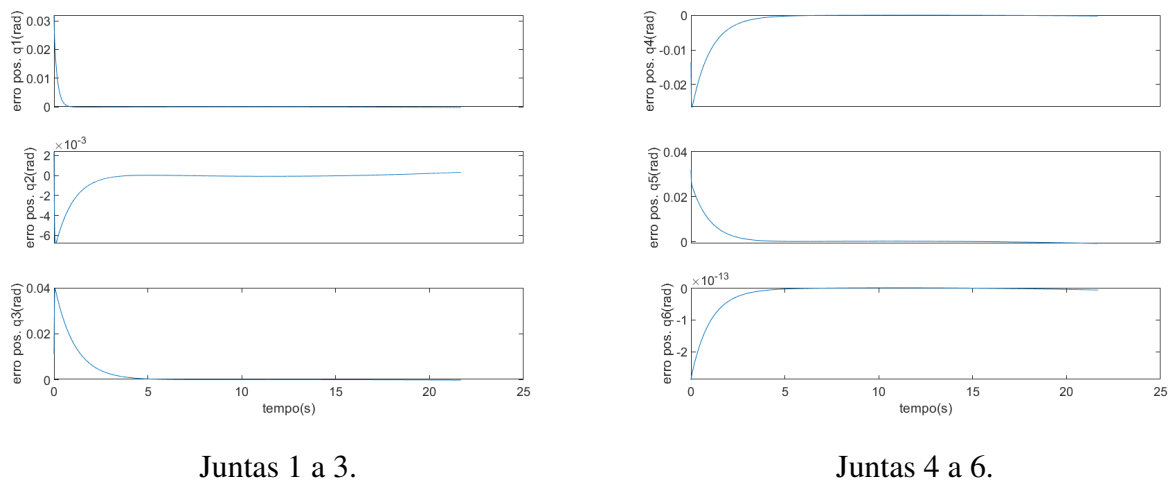


Figura 14: Erro de posição em malha fechada.

A Figura 12 apresenta 6 gráficos contendo os erros absolutos de posição obtidos fazendo a diferença entre a trajetória esperada e a obtida, que é encontrada a partir do toque resultante no esquema em malha aberta exposto na Figura 11, aplicando a inversa da Equação 28, obtendo assim a aceleração, velocidade e posição através do torque.

A Figura 14 apresenta 6 gráficos contendo os erros absolutos de posição obtidos fazendo a diferença entre a trajetória esperada e a obtida, que é encontrada a partir do toque resultante no esquema em malha fechada exposto na Figura 5, aplicando a inversa da Equação 28, obtendo assim a aceleração, velocidade e posição através do torque controlado.

Os erros de posição obtidos durante a malha fechada mostram que o controlador conseguiu efetuar seu papel estabilizando o sistema, é possível observar que todos os desvios de posição da trajetória esperada não ultrapassaram 0,03 rad e com o passar do tempo se aproximam mais do zero. Como os erros de cada junta individual são pequenos, é esperado obter resultados satisfatórios no seguimento de trajetória devido ao erro acumulado no manipulador final seja pequeno.

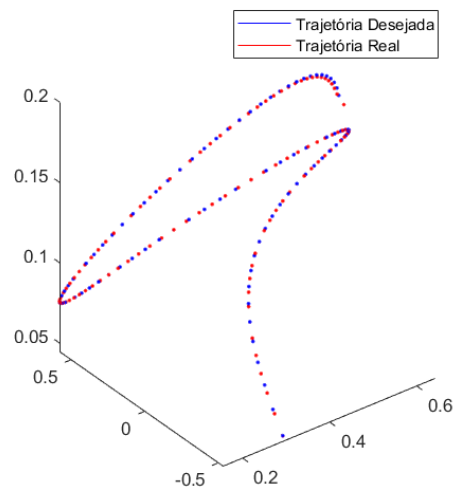


Figura 15: Comparação Trajetória Desejada e Real, em Malha Fechada.

A Figura 15 representa uma comparação da trajetória resultante da execução da Malha Fechada com a trajetória desejada. Por conta dos erros pequenos de posição individuais em cada junta, a trajetória do efetuador final recebe um acúmulo de erros pequeno, não apresentando grandes desvios da trajetória prevista. Sendo possível comprovar a eficácia do modelo apresentado, assim como do controlador.

Toda a produção deste trabalho está disponível de maneira digital num repositório <sup>1</sup>.

<sup>1</sup>Repositório disponível em: <https://github.com/junior1407/UR5>.

# 5

## Conclusão

Foram apresentados métodos para obter um modelo matemático do braço robótico UR5, e como forma de validação foi submetido a uma trajetória em malha fechada usando o erro de posição como métrica de desempenho.

O modelo foi construído usando notação de Denavit-Hartenberg e seu equacionamento com base nas equações de Euler-Lagrange. Foi proposto um controlador do tipo PD para lidar com a natureza instável do manipulador e poder seguir a trajetória prevista, que foi calculado a partir de um algoritmo de otimização. A solução foi simulada utilizando Matlab®. O experimento foi criado de maneira que o manipulador tenha que se mover de uma posição inicial, passando por pontos-chave até um ponto final. Com base nos resultados apresentados, foi possível constatar um erro condizente com o esperado no modelo em malha fechada, comprovando a eficácia do modelo ao lidar com a tarefa e também do controlador calculado.

Com base no modelo final obtido, é esperado que a distância entre pesquisas a nível industrial e nível acadêmico seja encurtada, isto é, através da disponibilização de um modelo matemático que já lida com a física do sistema é criada uma camada de abstração no desenvolvimento de aplicações para esse manipulador robótico, restando apenas ao desenvolvedor se concentrar em problemas de alto nível, podendo assim dedicar todo seu esforço em aplicar modernas técnicas e tecnologias.

### 5.1 Trabalhos Futuros

Como considerações finais, é importante elucidar possíveis melhorias que podem enriquecer esse trabalho. A solução proposta só foi implementada por meio de simulações. Uma extensão natural do trabalho é implementar o modelo e fazer testes num manipulador real, observando divergências e como o sistema se comporta com incertezas. Isso agregaria mais robustez ao modelo, com valores mais próximos da realidade e conseqüentemente gerando um controlador

também mais robusto. Como alternativa a obter o manipulador robótico que possui um custo muito elevado, é possível conseguir uma simulação com física de alta fidelidade utilizando o motor *Unity3D*, que consiste numa aplicação que fornece um ambiente 3D utilizando tecnologias de ponta para fornecer um ambiente com física implementada e recursos próprios para prototipagem de robôs.

Outra possível melhoria seria fazer mais estudos na modelagem do tensor de inércia considerando uma distribuição de massa não uniforme. Métodos de modelagem do tensor de inércia podem melhorar a acurácia do modelo, pois os tensores de inércia tem um impacto nas equações dinâmicas que cresce conforme a massa do elo.

# Referências bibliográficas

- S. Anand. Software for control and dynamic simulation of unimate puma 560 robot. 1993.
- Michael Bosse, Gabriel Agamennoni, and Igor Gilitschenski. Robust estimation and applications in robotics. *Foundations and Trends in Robotics*, 4:225–269, 01 2016. DOI [10.1561/23000000047](https://doi.org/10.1561/23000000047).
- Torgny Brogaardh. Present and future robot control development—an industrial perspective. *Annual Reviews in Control*, 31(1):69–79, 2007.
- Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011.
- S. Elgazzar. Efficient kinematic transformations for the puma 560 robot. *IEEE J. Robotics Autom.*, 1:142–151, 1985.
- Wenchuan Jia, Huayan Pu, Xin Luo, Xuedong Chen, Dandan Huang, and Ou Bai. Reliable control of a quadruped walking robot in uneven terrain environment based on noninvasive brain-computer interface. 08 2012.
- M.R.P Ragazzon. Robot manipulator collision handling in unknown environment without using external sensors. Master’s thesis, Norwegian University of Science and Technology, NTNU, 2012.
- Mohammad Rashidifar, Ali Rashidifar, and Hossein Ahmadi. Modeling and control of 5dof robot arm using fuzzy logic supervisory control. *IAES International Journal of Robotics and Automation (IJRA)*, 2, 01 2013. DOI [10.11591/ijra.v2i2.2974](https://doi.org/10.11591/ijra.v2i2.2974).
- Universal Robots. Ur5 technical specifications. Technical Report 110105, 2016. URL [https://www.universal-robots.com/media/50588/ur5\\_en.pdf](https://www.universal-robots.com/media/50588/ur5_en.pdf). [Online; Acessado em 21 de Dezembro de 2020].
- Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

Wikipédia. Parâmetros de Denavit-Hartenberg — Wikipedia, a enciclopédia livre.

[https://pt.wikipedia.org/wiki/Parâmetros\\_de\\_Denavit-Hartenberg](https://pt.wikipedia.org/wiki/Parâmetros_de_Denavit-Hartenberg), 2020.

[Online; Acessado em 21 de Dezembro de 2020].