

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE MECÂNICA  
CURSO DE ENGENHARIA MECÂNICA

LAWRENCE MIERS SAMBURGO

**PROJETO DE UM MANIPULADOR ROBÓTICO DE BAIXO CUSTO**

TRABALHO DE CONCLUSÃO DE CURSO  
(Tcc2 - Nº 30)

CURITIBA

2017

LAWRENCE MIERS SAMBURGO

## **PROJETO DE UM MANIPULADOR ROBÓTICO DE BAIXO CUSTO**

Monografia do Projeto de Pesquisa apresentada à disciplina de Trabalho de Conclusão de Curso - Tcc2 do curso de Engenharia Mecânica da Universidade Tecnológica Federal do Paraná, como requisito parcial para aprovação na disciplina.

Orientador: Prof. Msc. Sidney Carlos Gasoto

CURITIBA

2017

## TERMO DE APROVAÇÃO

Por meio deste termo, aprovamos a monografia do Projeto de Pesquisa "PROJETO DE UM MANIPULADOR ROBÓTICO DE BAIXO CUSTO", realizado pelo aluno Lawrence Miers Samburgo, como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso - Tcc2, do curso de Engenharia Mecânica da Universidade Tecnológica Federal do Paraná.

Prof. Msc. Sidney Carlos Gasoto

Damec, UTFPR

Orientador

Prof. Dr. Luiz Carlos de Abreu Rodrigues

Damec, UTFPR

Avaliador

Prof. Dr. Luís Paulo Laus

Damec, UTFPR

Avaliador

Curitiba, 10 de novembro de 2017

## RESUMO

SAMBURGO, Lawrence. Compartilhamento da informação e do conhecimento em bibliotecas especializadas. 2017. (79) f. Monografia (Projeto de um Manipulador Robótico de Baixo Custo) – Programa de Graduação em Engenharia Mecânica, Univesidade Tecnológica Federal do Paraná. Curitiba, 2017.

Mesmo que o uso de manipuladores robóticos seja interessante em diversos meios, como o doméstico ou o comercial, por exemplo, por possuir, dentre outras, a capacidade de movimentar objetos de um ponto a outro, ele ainda não está sendo observado, e manipuladores de baixo custo são atualmente limitados apenas a robôs recreacionais ou educativos, por não possuírem confiabilidade e capacidade suficientes para realizarem tarefas de maneira mais satisfatória. Tomando isso em consideração, este trabalho realizou o projeto cinemático, estrutural e dinâmico de um manipulador robótico semelhante aos atuais de baixo custo utilizando a mesma teoria e os mesmos conceitos que são utilizados para o projeto de manipuladores industriais, onde foram desenvolvidas tanto equações algébricas quanto implementações em Matlab, assim como também foram realizados desenhos através do SolidWorks para todos os seus componentes e foi construído um protótipo funcional do manipulador desenvolvido. Analisou-se a partir do resultado obtido as capacidades e limitações do robô resultante e também se existem possíveis medidas que podem ser tomadas para refinar o funcionamento de manipuladores deste tipo no futuro. Com isso, o trabalho realizado permitiu a verificação de que o uso da teoria da robótica para projetos de manipuladores simples de baixo custo é uma poderosa ferramenta para evidenciar as capacidades e limitações deste tipo de robô e a partir disso identificar possíveis pontos de melhoria. Ele também proporcionou o desenvolvimento de ferramentas em Matlab e criação de procedimentos de projeto característicos que poderão ser úteis no desenvolvimento de outros robôs do mesmo tipo.

**Palavras-chave:** Manipulador robótico. Robô. Baixo custo. Projeto. Protótipo.

## ABSTRACT

SAMBURGO, Lawrence. Compartilhamento da informação e do conhecimento em bibliotecas especializadas. 2017. (79) f. Monografia (Projeto de um Manipulador Robótico de Baixo Custo) – Programa de Graduação em Engenharia Mecânica, Univesidade Tecnológica Federal do Paraná. Curitiba, 2017.

Although the use of robotic manipulators is interesting in all sorts of areas, such as inside houses or commercial stores, mainly because it possesses the capacity of moving objects from one point to another, this is not being seen, and all of the low cost manipulators nowadays are mainly being used only for recreational and educative purposes, because of their low reliability and capability to do satisfactory tasks. Taking all of this into consideration, this paper developed a kinematic, structural and dynamic design of a robotic manipulator with a configuration similar to low cost ones, using the same theory and the same concepts that are used for the project of industrial manipulators, where algebraic equations were developed and Matlab functions were implemented. Also, all the components were designed using SolidWorks and a functional prototype was built. From the result obtained from this, an analysis of the designed manipulator was made to identify capabilities, limitations and possible future improvement points. With this, this paper made it clear that the use of robotics theory in the design of even the simplest of low cost manipulators is a powerful tool to identify capabilities and limitations of this kind of robot, and from that find out possible points that could upgrade the whole design to a new level. It also created characteristic project procedures that can be useful in the development of similar robots.

**Keywords:** Robotic manipulator. Robot. Low Cost. Design. Prototype.

## LISTA DE ILUSTRAÇÕES

Figura 1 – OWI Robotic Arm .....	9
Figura 2 – DAGU Robotic Arm .....	10
Figura 3 – Parâmetros de Denavit-Hartenberg para um segmento $i-1$ .....	13
Figura 4 – Associação de sistemas de coordenadas em séries de segmentos .....	14
Figura 5 – Ângulos Z-Y-X de Euler.....	16
Figura 6 – Sistemas de coordenadas usualmente utilizados em manipuladores robóticos.....	18
Figura 7 – Manipulador planar com dois graus de liberdade.....	19
Figura 8 – Solução cinemática inversa que gera colisão com obstáculo .....	20
Figura 9 – Balanço de forças e momentos entre dois segmentos vizinhos .....	27
Figura 10 – Representação esquemática do manipulador com todos os sistemas de coordenadas, juntas rotativas numeradas de 1 a 6.....	29
Figura 11 – Gráfico da configuração do manipulador para as variáveis de junta .....	34
Figura 12 – Área de trabalho do manipulador .....	40
Figura 13 – Vista 3D do manipulador robótico projetado .....	44
Figura 14 – Servomotor TowerPro MG946R .....	46
Figura 15 – Segmento 0 do manipulador .....	48
Figura 16 – Segmento 1 do manipulador .....	49
Figura 17 – Segmento 2 do manipulador .....	50
Figura 18 – Segmento 3 do manipulador .....	51
Figura 19 – Segmento 4 do manipulador .....	52
Figura 20 – Segmento 5 do manipulador .....	53
Figura 21 – Segmento 6 do manipulador .....	54
Figura 22 – Duas vistas do manipulador robótico em configuração extrema .....	56
Figura 22 – Manipulador robótico construído .....	59
Figura 23 – Cabeamento do manipulador robótico .....	60

## LISTA DE TABELAS

Tabela 1 – Parâmetros D-H do manipulador .....	30
Tabela 2 – Benchmarking do alcance de manipuladores de baixo custo .....	45
Tabela 3 – Dimensões dos segmentos do manipulador.....	45
Tabela 4 – Servomotores utilizados no manipulador.....	46
Tabela 5 – Parâmetros dinâmicos do segmento 1 .....	50
Tabela 6 – Parâmetros dinâmicos do segmento 2 .....	51
Tabela 7 – Parâmetros dinâmicos do segmento 3 .....	52
Tabela 8 – Parâmetros dinâmicos do segmento 4 .....	52
Tabela 9 – Parâmetros dinâmicos do segmento 5 .....	53
Tabela 10 – Parâmetros dinâmicos do segmento 6 .....	55
Tabela 11 – Custo do manipulador .....	55
Tabela 12 – Torque nas juntas X Massa na garra e aceleração das juntas.....	57
Tabela 12 – Torque nas juntas X Massa na garra e aceleração das juntas.....	58
Tabela 13 – Relação juntas X pinos.....	60
Tabela 14 – Características do manipulador robótico .....	62

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	Contexto do Tema	9
1.2	Caracterização do Problema	10
1.3	Objetivos	10
1.3.1	Objetivos Gerais	10
1.3.2	Objetivos Específicos	11
1.4	Justificativa	11
1.5	Conteúdo ou Etapas do Trabalho	11
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>12</b>
2.1	Análise Cinemática	12
2.1.1	Análise Cinemática Direta	12
2.1.2	Análise Cinemática Inversa	18
2.1.3	Propagação de velocidades	20
2.2	Análise Dinâmica	23
2.2.1	Formulação Lagrangiana	24
2.2.2	Formulação Iterativa de Newton-Euler	25
<b>3</b>	<b>PROCEDIMENTOS METODOLÓGICOS</b>	<b>28</b>
<b>4</b>	<b>PROJETO DO MANIPULADOR</b>	<b>29</b>
4.1	Cinemática	29
4.1.1	Descrição do Manipulador Robótico	29
4.1.2	Cinemática Direta	31
4.1.3	Cinemática Inversa	34
4.1.4	Jacobiano	41
4.2	Dinâmica	43
4.2.1	Projeto estrutural do manipulador robótico	44
4.2.2	Custo dos componentes do manipulador	55
4.2.3	Solução dinâmica do manipulador	56
4.3	Construção de protótipo do manipulador robótico	59
<b>5</b>	<b>RESULTADOS</b>	<b>62</b>
<b>6</b>	<b>CONCLUSÕES</b>	<b>65</b>
	<b>REFERÊNCIAS</b>	<b>68</b>
	APÊNDICE A – Função Matlab cinemática direta 1	69
	APÊNDICE B – Função Matlab cinemática direta 2	70
	APÊNDICE C – Função Matlab cinemática inversa	71
	APÊNDICE D – Função Matlab Jacobiano	75
	APÊNDICE E – Função Matlab dinâmica	76
	APÊNDICE F – Rotina de controle em arduino	78

# 1 INTRODUÇÃO

## 1.1 Contexto do Tema

O rápido crescimento industrial da atualidade fez com que indústrias estejam cada vez mais interessadas e necessitadas de soluções que forneçam alta produtividade com precisão e baixo custo de operação. Dentre as soluções existentes a robótica é uma das principais, viabilizando processos eficientes, rápidos, seguros e custo efetivos que fornecem resultados satisfatórios (NABEEL, et al, 2013). Um exemplo disso é a utilização de manipuladores robóticos, que é muito frequente em indústrias de manufatura e automóveis para operações de corte, furação, solda, manipulação, montagem e pintura, e é efetuada tendo como objetivo a obtenção de alta precisão e desempenho nos processos NABEEL, et al, 2013).

Este aumento progressivo no uso de manipuladores no meio industrial gerou o surgimento de versões de baixo custo deste tipo de robô, que possuem, em geral, um propósito mais recreativo, como o “OWI Robotic Arm” e o “DAGU Robotic Arm”, mostrados nas figuras 1 e 2 abaixo:



**Figura 1 – OWI Robotic Arm**

**Fonte: (Site oficial do OWI Robotic Arm)**



**Figura 2 – DAGU Robotic Arm**

**Fonte: (Site Oficial do DAGU Robotic Arm)**

## **1.2 Caracterização do Problema**

O aumento do uso de manipuladores robóticos na indústria não foi acompanhado na mesma proporção pelo aumento fora dela, e mesmo que o uso deste tipo de robô seja interessante em diversos meios, como o doméstico ou o comercial, por exemplo, por possuir, dentre outras, a capacidade de movimentar objetos de um ponto a outro, ele ainda não está sendo observado, e um grande fator para isso é o custo proibitivo que um manipulador de grande qualidade possui, sendo os de baixo custo limitados apenas a robôs recreacionais ou educativos, por não possuírem uma confiabilidade e uma capacidade suficientes para realizarem tais tarefas.

## **1.3 Objetivos**

### **1.3.1 Objetivos Gerais**

Este trabalho tem como objetivo relatar as etapas do projeto detalhado de um manipulador robótico de baixo custo semelhante ao da Figura 2, utilizando a mesma teoria e os mesmos conceitos que são utilizados para o projeto de manipuladores industriais e analisar a partir do resultado obtido as capacidades e limitações do robô resultante e também se existem possíveis medidas que podem ser tomadas para refinar o funcionamento de manipuladores deste tipo no futuro.

### **1.3.2 Objetivos Específicos**

Os objetivos específicos do trabalho que permitirão o atingimento dos objetivos gerais são:

- Desenvolver matemática e numericamente um manipulador robótico de baixo custo, modelando todos os seus componentes através de softwares CAD;
- Realizar a construção do manipulador desenvolvido e implementar rotinas de controle simples para verificar seu funcionamento;
- Analisar os resultados obtidos.

### **1.4 Justificativa**

A automação na forma de manipuladores robóticos já se provou muito útil no meio industrial, principalmente em tarefas repetitivas como manipulação de objetos, porém este não é o único meio que possui tarefas desta natureza, diversas atividades tanto domésticas quanto comerciais são ainda extremamente repetitivas e seria interessante a viabilização financeira de robôs para que qualquer pessoa física possa usufruir das suas vantagens.

### **1.5 Conteúdo ou Etapas do Trabalho**

Para descrever o projeto, o presente trabalho tem seis capítulos. No Capítulo 2 é realizada toda uma revisão teórica dos conceitos de análise cinemática e dinâmica da robótica, no Capítulo 3 foram descritos todos os procedimentos e ferramentas que foram utilizados no Capítulo 4, o Capítulo de projeto. No Capítulo 5 todos os resultados provindos dos cálculos foram utilizados e uma análise de capacidades e limitações do manipulador foi realizada. O Capítulo 6 apresenta as conclusões sobre o trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para realizar o projeto de um manipulador robótico é necessário estudar como se dá o seu movimento. Todos os aspectos geométricos e dependentes do tempo deste movimento são considerados em uma análise cinemática, e a relação entre o movimento e as forças e torques que o geram é considerada no problema da análise dinâmica (CRAIG, 2005).

Neste Capítulo uma revisão teórica destes dois tipos de análise será feita, ajudando a esclarecer como serão feitos todos os cálculos nos Capítulos subsequentes de projeto.

### 2.1 Análise Cinemática

A análise cinemática de manipuladores robóticos pode ser dividida em três partes: cinemática direta, cinemática inversa e propagação de velocidades. O objetivo da cinemática direta é determinar qual será a posição final da garra do robô para determinada rotação das juntas, e o objetivo da cinemática inversa é determinar para quais rotações de cada junta é possível se atingir uma posição final pré-determinada da garra do robô, ou seja, uma é o inverso da outra, enquanto a cinemática direta fornece posições cartesianas com base em posições angulares dadas, a inversa fornece posições angulares para posições cartesianas dadas (CRAIG, 2005).

A determinação da velocidade de cada um dos segmentos é também uma tarefa da análise cinemática, para isso será utilizado um operador chamado de Jacobiano, que consegue fornecer tanto velocidades cartesianas a partir de velocidades angulares, quanto velocidades angulares a partir de velocidades cartesianas, é um operador que consegue fazer, para velocidades, tanto a tarefa da cinemática direta, quanto a da inversa (CRAIG, 2005).

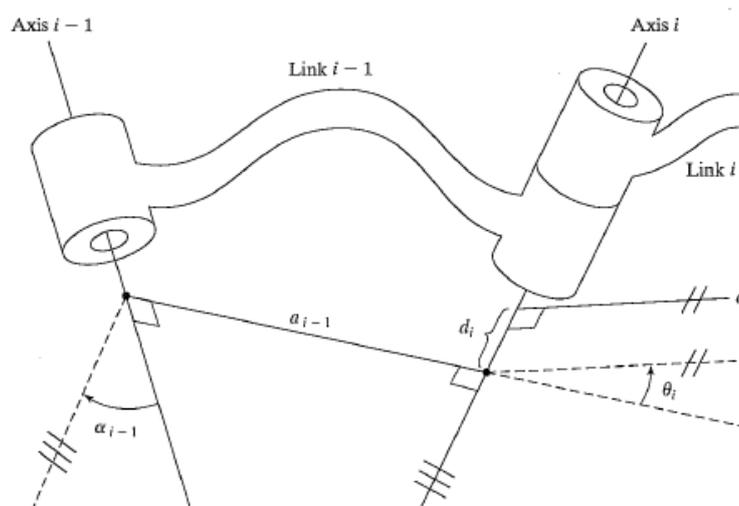
#### 2.1.1 Análise Cinemática Direta

Um manipulador robótico é composto por uma série de segmentos ligados por juntas ou de rotação ou de translação, e sempre pode ser descrito cinematicamente atribuindo-se quatro valores a cada um dos segmentos, dois destes valores

definindo o próprio segmento e outros dois definindo a relação do segmento com um dos segmentos vizinhos (CRAIG, 2005).

Para o caso deste trabalho, onde o manipulador possui apenas juntas de rotação, um dos valores que define o segmento é a variável de junta, representada por  $\theta_i$ , que representa a quantidade de rotação em relação a um eixo comum entre dois segmentos, e os outros três valores são os parâmetros fixos do segmento: sendo o comprimento do segmento representado por  $a_{i-1}$ ,  $d_i$  o deslocamento do segmento em relação a um segmento vizinho e  $\alpha_{i-1}$  a torção do segmento, considerando um segmento qualquer  $i-1$ . A descrição de mecanismos com estes parâmetros é chamada de notação de Denavit-Hartenberg (D-H) (CRAIG, 2005).

A Figura 3 abaixo ilustra os quatro parâmetros descritos.



**Figura 3 – Parâmetros de Denavit-Hartenberg para um segmento  $i-1$**

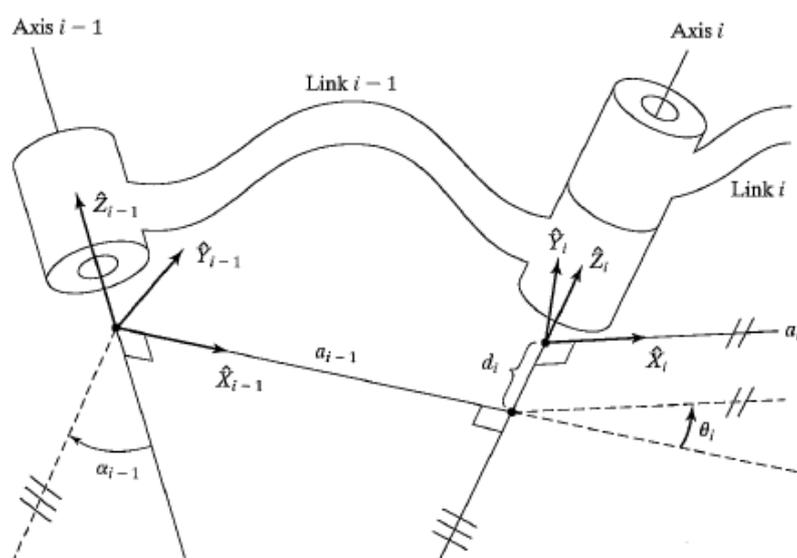
**Fonte: (CRAIG, 2005)**

Para descrever cada segmento em relação a seus vizinhos é necessário fixar um sistema de coordenadas a cada um deles, e para a determinação destes sistemas de coordenadas uma rotina padrão pode ser utilizada (CRAIG, 2005):

1. Identificar os eixos das juntas e imaginar que existam linhas infinitas ao longo deles, para os passos 2 a 5 abaixo considerar duas destas linhas (eixos  $i$  e  $i+1$ ).

2. Identificar a perpendicular comum entre estas linhas, ou o ponto de interseção. No ponto de interseção ou no ponto onde a perpendicular comum encontra o eixo  $i$ , associar a origem do sistema de coordenada do segmento. No caso de linhas paralelas, associar a perpendicular comum de maneira a zerar a maior quantidade possível de parâmetros.
3. Associar o eixo  $Z_i$  apontando ao longo do eixo  $i$ .
4. Associar o eixo  $X_i$  apontando ao longo da perpendicular em comum ou, no caso de interseção de eixos, associar para ser normal ao plano contendo os dois eixos.
5. Associar  $Y_i$  de acordo com um sistema de coordenadas de mão direita.
6. Associar o sistema de coordenadas  $\{0\}$  para coincidir com  $\{1\}$  quando  $\theta_1 = 0$ . Para o sistema de coordenadas  $\{N\}$  escolher uma origem e uma direção de  $X_n$  visando zerar a maior quantidade de parâmetros de segmento.

Quando aplicada a rotina na Figura 3, obtêm-se os sistemas de coordenadas mostrados na Figura 4 a seguir:



**Figura 4 – Associação de sistemas de coordenadas em séries de segmentos**

Fonte: (CRAIG, 2005)

Observando a figura, uma definição mais geral para os parâmetros D-H pode ser obtida, com base nos eixos dos sistemas de coordenadas, como segue:

$a_{i-1}$  = a distância de  $Z_{i-1}$  a  $Z_i$  medida ao longo de  $X_{i-1}$ ;

$\alpha_{i-1}$  = o ângulo de  $Z_{i-1}$  a  $Z_i$  medida sobre  $X_{i-1}$ ;

$d_i$  = a distância de  $X_{i-1}$  a  $X_i$  medida ao longo de  $Z_i$ ;

$\theta_i$  = o ângulo de  $X_{i-1}$  a  $X_i$  medida sobre  $Z_i$ .

### 2.1.1.1 Transformação de coordenadas

Após todos os sistemas de coordenadas definidos e os parâmetros D-H determinados, a equação (1) pode ser utilizada para cada par de sistemas de coordenadas vizinhos (CRAIG, 2005).

$${}^{i-1}T_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

onde as funções trigonométricas seno e cosseno foram representadas pelas letras “s” e “c”, respectivamente. Esta nomenclatura é usada diversas vezes ao longo deste trabalho.

Esta equação calcula um operador de transformação de coordenadas entre dois segmentos com base apenas nos parâmetros D-H, ou seja,  ${}^{i-1}T_i$  dá a posição e a orientação do sistema de coordenadas  $i$  em relação ao sistema de coordenadas  $i-1$ . A posição é dada pela última coluna da matriz e a orientação pelo conjunto das três primeiras linhas e colunas, que também são chamadas de submatriz de rotação de coordenadas, como mostra a equação (2) a seguir (CRAIG, 2005).

$${}^{i-1}R_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (2)$$

Utilizando a equação (1) com  $i=1 \rightarrow n$ , para um manipulador com  $n$  juntas, pode se multiplicar todos os operadores de transformação resultantes na forma da equação (3) abaixo

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot (\dots) \cdot {}^{n-1}T_n, \quad (3)$$

na qual o resultado será um operador de transformação que fornece as coordenadas cartesianas de posição e orientação do último sistema de coordenadas do robô escritas no sistema de coordenadas do sistema de coordenadas base, em função apenas das coordenadas angulares de cada junta, ou seja, com isso tem-se a cinemática direta (CRAIG, 2005).

### 2.1.1.2 Ângulos Z-Y-X de Euler

Embora o operador de transformação resultante da equação (3) forneça a orientação final do robô através da matriz de rotação de 9 variáveis, este resultado não promove uma fácil visualização de como o último sistema de coordenadas rotacionou para atingir a nova orientação, e existe uma representação de orientação chamada de ângulos Z-Y-X de Euler que transforma todas estas 9 variáveis em apenas 3 rotações, seguindo a seguinte lógica (CRAIG, 2005):

*Inicie com o sistema de coordenadas {B} coincidente com um outro sistema já conhecido {A}. Primeiro rotacione {B} sobre  $\hat{Z}_B$  por um ângulo  $\alpha$ , em seguida sobre o eixo  $\hat{Y}_B$  por um ângulo  $\beta$  e finalmente sobre o eixo  $\hat{X}_B$  por um ângulo  $\gamma$ .*

A Figura 5 abaixo representa este procedimento descrito.

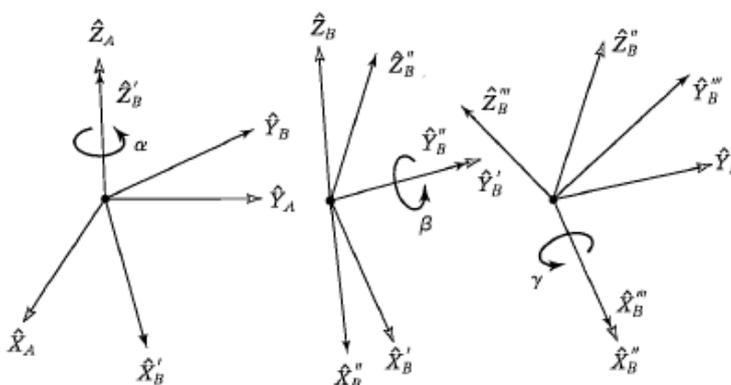


Figura 5 – Ângulos Z-Y-X de Euler

Fonte: (CRAIG, 2005)

A matriz de rotação pode então ser transformada nos ângulos de Euler de acordo com a equação (4) a seguir (CRAIG, 2005)

$$\begin{aligned}\beta &= \text{atan } 2\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \\ \alpha &= \text{atan } 2\left(r_{21} / c\beta, r_{11} / c\beta\right) . \\ \gamma &= \text{atan } 2\left(r_{32} / c\beta, r_{33} / c\beta\right)\end{aligned}\quad (4)$$

O inverso deste procedimento é também muito importante, principalmente na cinemática inversa, pois nela é necessário dar como parâmetros de entrada a posição e orientação final do manipulador em coordenadas cartesianas, e é inviável visualizar quais nove parâmetros da matriz de rotação são necessários para uma determinada orientação que se queira, então ao invés disso são selecionados os ângulos de Euler necessários para uma dada orientação, e então, a partir deles, os nove parâmetros de rotação são encontrados através da equação (5) (CRAIG, 2005).

$${}^i R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \quad (5)$$

### 2.1.1.3 Inclusão da Garra do Robô na Cinemática

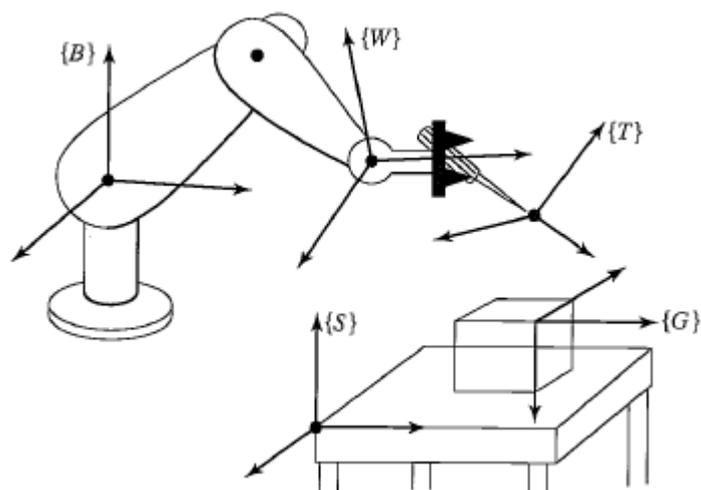
Uma observação válida é a de que estes cálculos da cinemática do manipulador são realizados apenas até a última junta, ou seja, eles não incluem a garra do robô (ou outra ferramenta que possa estar na sua extremidade) em seus cálculos, pois assim o processo fica mais simples. A garra do robô pode ser inserida nos cálculos posteriormente realizando-se uma multiplicação simples de operadores de transformação na forma da equação (6) (CRAIG, 2005)

$${}^0 T = {}^0 T \cdot {}^n T, \quad (6)$$

onde  $\{T\}$  é o sistema de coordenadas associado à garra do robô.

Da mesma forma, vários outros sistemas de coordenadas usualmente utilizados podem ser relacionados ao operador de transformação resultante da equação (3), como o sistema de coordenadas  $\{S\}$  e o  $\{G\}$ , que são comumente

associados a estações de trabalho e pontos de destino, respectivamente, como mostra a Figura 6 a seguir, onde  $\{0\} = \{B\}$  e  $\{n\} = \{W\}$ , representando a base e o punho no robô (CRAIG, 2005).



**Figura 6 – Sistemas de coordenadas usualmente utilizados em manipuladores robóticos**

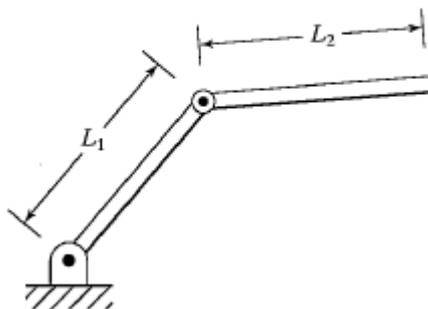
Fonte: (CRAIG, 2005)

## 2.1.2 Análise Cinemática Inversa

O problema da cinemática inversa é mais complexo do que o da direta por ser um problema não linear, após resolvê-lo é necessário se saber se, para a configuração do manipulador analisado, sequer existe uma solução inversa possível (CRAIG, 2005).

### 2.1.2.1 Existência de Soluções

Na cinemática direta, os limites dos deslocamentos angulares das juntas do manipulador são conhecidos, logo, para qualquer valor alocado a eles uma solução em coordenadas cartesianas pode ser obtida. Mas no caso da cinemática inversa esta situação é mais delicada. Como os parâmetros de entrada são as próprias coordenadas cartesianas é necessário primeiro saber se o manipulador consegue ou não chegar até elas, ou seja, é necessário conhecer primeiro qual é o espaço de trabalho do manipulador, pois se, por exemplo, uma coordenada tridimensional cartesiana for dada a um manipulador planar de apenas dois graus de liberdade, como o da Figura 7 abaixo, é visível que nenhuma solução possível será obtida (CRAIG, 2005).



**Figura 7 – Manipulador planar com dois graus de liberdade**

Fonte: (CRAIG, 2005)

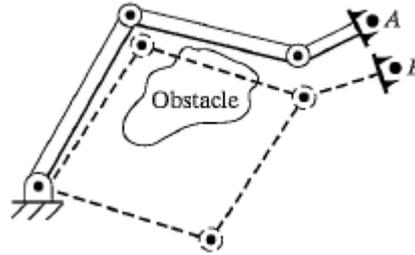
Embora estudos recentes em cinemática mostraram que a solução da cinemática inversa pode ser obtida para qualquer manipulador com juntas rotativas e/ou prismáticas que tenha até seis graus de liberdade, ela em casos especiais não pode ser obtida na forma fechada, e apenas numericamente através de processos iterativos que são muito mais demorados. Uma condição suficiente para a existência de solução na forma fechada para manipuladores com seis graus de liberdade é que eles tenham pelo menos 3 eixos em interseção, característica que a vasta maioria dos manipuladores da atualidade possuem (CRAIG, 2005), assim como o projetado neste trabalho, como será explicado em mais detalhes no Capítulo 4.

#### **2.1.2.2 Múltiplas Soluções**

Outro ponto importante na cinemática inversa é que muitas vezes mais de uma solução possível é encontrada, e esse número de soluções depende tanto do número de juntas do manipulador quanto dos seus parâmetros D-H. Muitos manipuladores com seis graus de liberdade, como o famoso PUMA 560, possuem até oito soluções possíveis para a cinemática inversa, ou seja, para dada coordenada cartesiana de posição e orientação, existem oito combinações de suas seis variáveis de junta  $\theta_i$  que fazem o manipulador chegar até o destino (CRAIG, 2005).

Qual das soluções utilizar em cada caso é algo que depende muito da situação, as soluções podem ser limitadas, por exemplo, por juntas que não podem realizar uma revolução completa, caso em que faz que sejam eliminadas todas as soluções que tenham variáveis de junta maiores que determinado ângulo. Outro exemplo de

seleção de soluções é o de obstáculos no caminho do manipulador, pode ser que uma das soluções, mesmo que viável na teoria, na prática faria o robô bater em alguma superfície, como mostra a Figura 8 a seguir (CRAIG, 2005):



**Figura 8 – Solução cinemática inversa que gera colisão com obstáculo**

Fonte: (CRAIG, 2005)

As soluções de manipuladores na forma fechada podem ser obtidas de duas maneiras, algebricamente e geometricamente. No método de solução geométrica a geometria espacial do robô é decomposta em vários problemas de geometria planar, e relações de trigonometria são utilizadas para a obtenção de uma solução. No método algébrico são utilizadas as equações matriciais provindas da formulação da cinemática inversa, e através de artifícios e transformações a solução inversa é obtida (CRAIG, 2005).

Este trabalho realizará uma solução na forma algébrica, que será descrita em detalhes no Capítulo 4, quando for obtida para o manipulador do projeto.

### 2.1.3 Propagação de velocidades

As duas análises cinemáticas feitas até agora descrevem o manipulador apenas estaticamente, não mostrando se ele vai chegar até certo ponto com uma velocidade alta ou baixa, o que é uma informação de suma importância para, por exemplo, processos industriais que tenham como requisito uma produtividade alta, ou processos de soldagem, onde o manipulador precisa se mover com uma velocidade constante.

O problema da propagação de velocidades possui os mesmos fundamentos do problema da cinemática direta e inversa: quer se saber qual é a quantidade cartesiana de certa grandeza no final do robô para determinadas variáveis de junta, ou vice-versa. No caso já visto anteriormente de cinemática direta a grandeza que se

desejava saber era a posição e orientação do final do robô  ${}^0_nT$ , em relação a determinadas variáveis de junta de deslocamento angular  $\theta_i$ , no caso das velocidades a ideia é semelhante, deseja-se saber a velocidade cartesiana linear e rotacional final do robô,  $v$  e  $\omega$  respectivamente, para determinadas variáveis de junta de velocidade angular  $\dot{\theta}_i$ , ou no caso inverso, quais velocidades angulares são necessárias para atender a determinadas velocidades rotacionais e lineares na ponta do manipulador (CRAIG, 2005).

Felizmente para o caso das velocidades, existe um operador matricial que consegue realizar a transformação tanto direta quanto inversa de velocidades angulares para velocidades cartesianas, e ele é chamado de Jacobiano, sendo representado pela letra  $J$ , uma matriz dependente das variáveis de junta  $\theta_i$ . As equações (7) e (8) abaixo mostram como a Jacobiano pode ser utilizada em conjunto com os vetores de velocidade cartesiana  $v = \begin{bmatrix} v \\ \omega \end{bmatrix}$  e velocidade angular das juntas,  $\dot{\Theta}$  (CRAIG, 2005):

$$v = J(\Theta) \cdot \dot{\Theta}; \quad (7)$$

$$\dot{\Theta} = J^{-1}(\Theta) \cdot v. \quad (8)$$

O Jacobiano de um manipulador pode ser encontrado por diversos métodos existentes, o descrito a seguir está de acordo com o estilo usado em (KHATIB, 2008) e é muito interessante por utilizar apenas as informações que já foram obtidas na cinemática direta.

### 2.1.3.1 Obtenção do Jacobiano

Considerando operador de transformação da equação (1) no formato descrito na equação (9) a seguir

$${}^n_rT = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

o Jacobiano de um manipulador pode ser obtido pela equação (10) abaixo

$${}^n J = \begin{bmatrix} \frac{\partial {}^n \mathbf{p}}{\partial \theta_1} & \frac{\partial {}^n \mathbf{p}}{\partial \theta_2} & \dots & \frac{\partial {}^n \mathbf{p}}{\partial \theta_i} \\ {}^n \mathbf{a}_1 & {}^n \mathbf{a}_2 & \dots & {}^n \mathbf{a}_i \end{bmatrix}, \quad (10)$$

onde  $n$  é o sistemas de coordenadas em que se deseja representar a Jacobiano,  ${}^n \mathbf{p}$  é o vetor posição do último sistema de coordenadas do robô escrito no sistema  $n$  e  $\mathbf{a}$  é dado pela respectiva coluna na matriz do operador de transformação, já mostrado na equação (9).

Para uma representação das velocidades na garra do robô escritas no sistema de coordenadas  $\{0\}$  (de base), não só é necessário escrever a equação (10) com  $n=0$ , como também precisa-se adicionar a influência daquele deslocamento final da garra em relação ao último sistema de coordenadas contemplado pelas equações de análise cinemática  ${}^n T$ , explicado na equação (6).

A formula para transformar o Jacobiano da última junta do robô escrita no sistema de coordenadas  $\{0\}$   ${}^0 J_n$  em um Jacobiano da garra do robô, escrito também no mesmo sistema de coordenadas  ${}^0 J_T$ , é dada pela equação (11) abaixo

$${}^0 J_T = \begin{bmatrix} {}^0 R & -{}^0 P_{ne} \\ 0 & {}^0 R \end{bmatrix} \cdot {}^0 J_n. \quad (11)$$

Nesta equação,  ${}^0 R$  é a parcela de rotação do operador de transformação e  ${}^0 P_{ne}$  é um vetor de deslocamento, representando a distância entre a última junta e a garra, escrito no sistema de coordenadas de base.

Após a determinação do Jacobiano, a sua inversa pode ser computada e assim as equações (7) e (8) podem ser utilizadas.

### 2.1.3.2 Singularidades

Um ponto a ser considerado na determinação do Jacobiano inversa  $J^{-1}(\Theta)$  é se ela realmente é inverssível para determinados  $\Theta$ . Todos os manipuladores, em determinadas regiões de sua área de trabalho possuem variáveis de junta  $\theta_i$  que geram Jacobianos não inverssíveis, estes pontos são chamados de singularidades e neles o manipulador fica impossibilitado de seguir determinada direção no espaço

cartesiano, para toda e qualquer velocidade angular possível, ou seja, ele perde um ou mais de seus graus de liberdade (KHATIB, 2008).

A ocorrência de singularidades varia conforme o número de juntas e os parâmetros D-H de cada manipulador, mas uma singularidade facilmente visualizada em qualquer robô é a no limite da área de trabalho. Como as singularidades ocorrem quando a matriz Jacobiana não pode ser invertida, elas podem ser encontradas por meio da equação (12) abaixo (CRAIG, 2005)

$$DET[J(\Theta)] = 0. \quad (12)$$

Conhecer os pontos de singularidade de um manipulador é de suma importância, pois assim estratégias de programação de sua trajetória podem ser adotadas para evitarem-se não só regiões de singularidade, como também regiões próximas a singularidades, que geram picos de velocidade angular (KHATIB, 2008).

## 2.2 Análise Dinâmica

A análise dinâmica visa estudar as forças necessárias para gerar movimento em manipuladores. Existem dois problemas relacionados à dinâmica destes robôs que são solucionados pela análise, o primeiro é: dado um ponto de trajetória com posição, velocidade e aceleração angulares ( $\Theta$ ,  $\dot{\Theta}$  e  $\ddot{\Theta}$ , respectivamente), qual o torque nas juntas  $\tau$  necessário. O segundo problema consiste do inverso disso, qual ponto de trajetória será gerado com a aplicação de determinado torque. A solução do primeiro problema é utilizada de muitas maneiras, como no controle dos manipuladores e também na seleção de motores para as juntas, já a solução do segundo problema tem outras aplicações como auxiliar na simulação computacional (CRAIG, 2005).

Duas das principais formas de realizar a análise dinâmica são a formulação Lagrangiana e a formulação iterativa de Newton-Euler, a primeira é baseada em princípios de conservação de energia, enquanto a segunda baseia-se em princípios de balanços de força. Ambas funcionam bem, mas com focos diferentes. A Lagrangiana facilita a visualização de uma equação na forma fechada da dinâmica do manipulador, enquanto a formulação de Newton-Euler utiliza equações de propagação de velocidades, acelerações e força que são facilmente implementadas

computacionalmente para uma solução numérica da dinâmica. A implementação das equações na forma Lagrangiana por outro lado, não é nada trivial, porém obter a forma fechada da solução dinâmica é muito mais intuitivo com esta formulação (KHATIB, 2008).

A estrutura final de uma solução dinâmica na forma fechada é mostrada na equação (13) a seguir (KHATIB, 2008).

$$\tau = M(\Theta) \cdot \ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta), \quad (13)$$

onde  $M(\Theta)$  é a matriz de massa do manipulador, sendo constituída por termos dependentes da variável de junta e multiplicados ela aceleração angular,  $V(\Theta, \dot{\Theta})$  é o vetor de termos centrífugos e Coriolis e  $G(\Theta)$  é o vetor de termos de gravidade (KHATIB, 2008).

O atrito e outros efeitos são também usualmente considerados na análise dinâmica, e sua inclusão pode ser feita somando-se diretamente à equação (13) um termo adicional que descreva o efeito destes fenômenos.

As formulações Lagrangiana e de Newton-Euler serão brevemente descritas nas seções seguintes.

### 2.2.1 Formulação Lagrangiana

A formulação Lagrangiana fornece métodos para o cálculo direto de cada um dos termos da equação (13) através de uma análise baseada em conceitos de conservação de energia (KHATIB, 2008).

A matriz de massa pode ser obtida pela equação (14)

$$M = \sum_{i=1}^n \left( m_i \cdot J_{v_i}^T \cdot J_{v_i} + J_{w_i}^T \cdot I_{C_i} \cdot J_{w_i} \right), \quad (14)$$

onde  $n$  é o número de juntas do manipulador,  $m_i$  é a massa do segmento  $i$  e  $I_{C_i}$  é o tensor de inércia no centro de massa do mesmo segmento. Os termos  $J_{v_i}$  e  $J_{w_i}$  são calculados de maneira semelhante ao cálculo do Jacobiano demonstrado na equação (10), como pode ser visto nas equações (15) e (16) abaixo (KHATIB, 2008):

$$J_{v_i} = \left[ \frac{\partial p_{c_1}}{\partial \theta_1}, \frac{\partial p_{c_2}}{\partial \theta_2}, \dots, \frac{\partial p_{c_i}}{\partial \theta_i}, 0, \dots, 0 \right]; \quad (15)$$

$$J_{w_i} = [a_1, a_2, \dots, a_i, 0, \dots, 0]. \quad (16)$$

Na equação (15),  $p_{c_i}$  é o vetor posição que relaciona as coordenadas do centro de massa do segmento à origem do seu sistema de coordenadas principal. Para as duas equações as matrizes devem ser preenchidas até o valor presente de  $i$  e as demais colunas devem ser preenchidas com zeros (KHATIB, 2008).

O termo  $V(\Theta, \dot{\Theta})$  pode ser determinado pela equação (17) a seguir

$$\begin{aligned} V &= C(\Theta)[\dot{\Theta}^2] + B(\Theta)[\dot{\Theta} \dot{\Theta}] \\ C(\Theta)[\dot{\Theta}^2] &= \begin{bmatrix} b_{1,11} & \dots & b_{1,nn} \\ \vdots & \dots & \vdots \\ b_{n,11} & \dots & b_{n,nn} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1^2 \\ \vdots \\ \dot{\theta}_n^2 \end{bmatrix} \\ B(\Theta)[\dot{\Theta} \dot{\Theta}] &= \begin{bmatrix} 2b_{1,12} & \dots & 2b_{1,(n-1)n} \\ \vdots & \dots & \vdots \\ 2b_{n,12} & \dots & 2b_{n,(n-1)n} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_{n-1} \dot{\theta}_n \end{bmatrix}, \\ b_{i,jk} &= \frac{1}{2}(m_{ijk} + m_{ikj} - m_{jki}) \end{aligned} \quad (17)$$

onde  $b_{ijk}$  são chamados de símbolos de Christoffel e  $m_{ijk} = \frac{\partial m_{ij}}{\partial \theta_k}$  (KHATIB, 2008).

Por último, o vetor de termos de gravidade pode ser obtido com a equação (18) (KHATIB, 2008)

$$G = -\left[ J_{v_1}^T (m_1 \cdot g) + \dots + J_{v_n}^T (m_n \cdot g) \right]. \quad (18)$$

### 2.2.2 Formulação Iterativa de Newton-Euler

Alternativamente ao método lagrangiano que, apesar de ser interessante por já dar resultados na forma da equação (13), tem uma formulação algébrica relativamente pesada, o método iterativo de Newton-Euler fornece a solução da dinâmica do manipulador de forma muito mais simples e direta. Por meio da análise das forças e momentos agindo entre os segmentos do manipulador, a formulação de Newton-Euler fornece equações de propagação de velocidade, aceleração, forças e

momentos nos centros de massa dos segmentos, do segmento {0} ao segmento {n-1}, e posteriormente de equações de propagação de forças e momentos totais nos segmentos, fazendo o caminho inverso, do segmento {n} ao {1} (CRAIG, 2005).

As iterações de  $i:0 \rightarrow n$  são feitas por meio das equações (19) a (24) a seguir:

$${}^{i+1}\omega_{i+1} = {}^iR \cdot {}^i\omega_i + \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}; \quad (19)$$

$${}^{i+1}\dot{\omega}_{i+1} = {}^iR \cdot {}^i\dot{\omega}_i + {}^iR \cdot {}^i\omega_i \times \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}; \quad (20)$$

$${}^{i+1}\dot{v}_{i+1} = {}^iR \left( {}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot{v}_i \right); \quad (21)$$

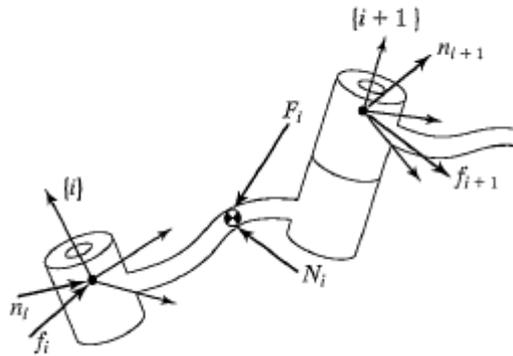
$${}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1}; \quad (22)$$

$${}^{i+1}F_{i+1} = m_{i+1} \cdot {}^{i+1}\dot{v}_{C_{i+1}}; \quad (23)$$

$${}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1} \cdot {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1} \cdot {}^{i+1}\omega_{i+1}. \quad (24)$$

A equação (19) calcula as velocidades rotacionais das juntas 1 a  $n$ , que são utilizadas na equação (20) para o cálculo das acelerações rotacionais das mesmas juntas, que por sua vez são utilizadas, em conjunto com as velocidades rotacionais, na equação (21) para a obtenção das acelerações lineares. Este processo iterativo repete-se nas equações seguintes para a determinação da aceleração, força e momento no centro de massa dos segmentos,  $\dot{v}_C$ ,  $F$  e  $N$ , respectivamente (CRAIG, 2005).

Os valores de força e momento encontrados nas equações (23) e (24) são então propagados de  $i:n-1 \rightarrow 1$ , por meio de análise de balanço de forças, como o da Figura 9 abaixo.



**Figura 9 – Balanço de forças e momentos entre dois segmentos vizinhos**

Fonte: (CRAIG, 2005)

A análise resulta nas equações (25) e (26) abaixo:

$${}^i f_i = {}^i R \cdot {}^{i+1} f_{i+1} + {}^i F_i; \quad (25)$$

$${}^i n_i = {}^i N_i + {}^i R \cdot {}^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}^i R \cdot {}^{i+1} f_{i+1}. \quad (26)$$

O torque necessário em cada junta pode ser então obtido, extraindo-se a componente Z do momento encontrado pela equação (26), como é mostrado na equação (27) (CRAIG, 2005)

$$\tau_i = {}^i n_i^T \cdot {}^i \hat{Z}_i. \quad (27)$$

### 3 PROCEDIMENTOS METODOLÓGICOS

O projeto do manipulador robótico foi feito seguindo as etapas descritas abaixo. O item 8 refere-se a uma etapa que será feita no Capítulo de resultados.

1. Escolha do tipo de manipulador robótico que será projetado, justificando a escolha de tipos de junta, número de graus de liberdade e parâmetros D-H;
2. Desenvolvimento da cinemática direta do manipulador com o auxílio do software Matlab;
3. Desenvolvimento da cinemática inversa com o auxílio do software Matlab e descrição da sua área de trabalho, a qual a cinemática inversa deve se limitar;
4. Obtenção da Jacobiano do manipulador e análise das singularidades;
5. Realização do projeto estrutural com o auxílio do software SolidWorks e explicações sobre algumas características do mecanismo desenvolvido;
6. Obtenção dos parâmetros dinâmicos dos segmentos do manipulador por meio do software SolidWorks;
7. Desenvolvimento da dinâmica utilizando os parâmetros resultantes do projeto estrutural e implementando equações iterativas no software Matlab;
8. Construção de protótipo manipulador robótico desenvolvido e implementação de rotinas de controle simples que possibilitem o seu funcionamento;
9. Análise do manipulador final resultante: características, custo, limitações e oportunidades de melhoria futuras.

## 4 PROJETO DO MANIPULADOR

### 4.1 Cinemática

#### 4.1.1 Descrição do Manipulador Robótico

A Figura 10 a seguir mostra o esquema do manipulador robótico idealizado, quando todas as suas variáveis de junta  $\theta_i$  são zero e já com todos os sistemas de coordenadas associados de acordo com o método descrito no Capítulo 2.

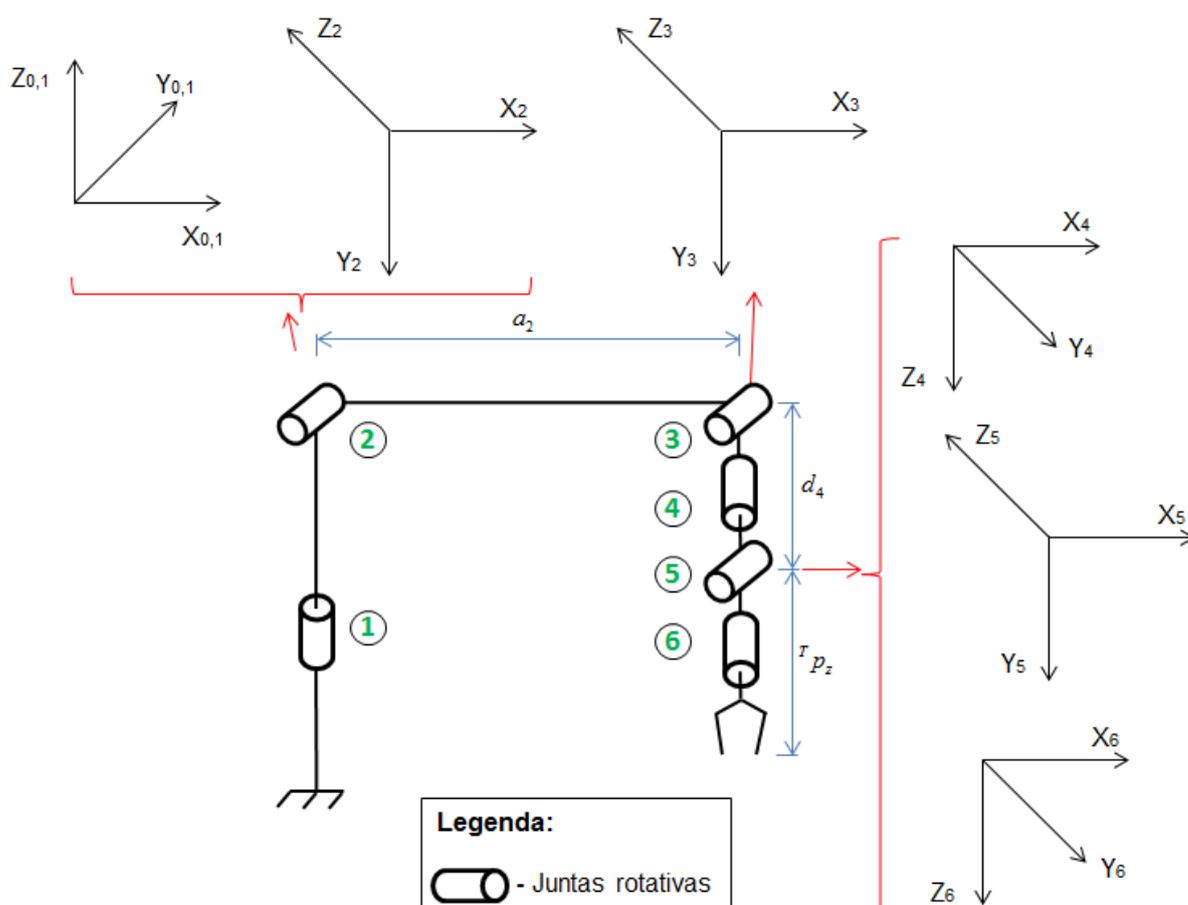


Figura 10 – Representação esquemática do manipulador com todos os sistemas de coordenadas, juntas rotativas numeradas de 1 a 6

Fonte: Autoria Própria<sup>1</sup>

Tendo em vista que o robô não possui uma tarefa específica pré-definida, uma configuração versátil foi utilizada, a de um manipulador articulado com seis graus de

<sup>1</sup> As demais figuras neste trabalho que não possuírem citação de fonte são de autoria própria.

liberdade. A estrutura articulada com todas as suas seis juntas rotativas foi escolhida, pois é a que proporciona a menor intrusão do manipulador em sua área de trabalho dentre todos os outros tipos de estrutura (CRAIG, 2005), o que faz com que este robô possa alcançar espaços confinados e ocupar a menor quantidade possível de espaço no ambiente, característica muito necessária para manipuladores em regiões menores do que fábricas, como meios domésticos. Os seis graus de liberdade foram escolhidos por possibilitarem que tarefas de manipulação de objetos possam ser executadas com maior abrangência possível, possibilitando o posicionamento de objetos em todas as três posições e orientações cartesianas.

Através da Figura 10, os parâmetros D-H são extraídos, eles são mostrados na Tabela 1 a seguir:

**Tabela 1 – Parâmetros D-H do manipulador**

$i$	$a_{i-1}$	$\alpha_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	0	$-90^\circ$	0	$\theta_2$
3	$a_2$	0	0	$\theta_3$
4	0	$-90^\circ$	$d_4$	$\theta_4$
5	0	$90^\circ$	0	$\theta_5$
6	0	$-90^\circ$	0	$\theta_6$

Fonte: Autoria Própria<sup>2</sup>

Os parâmetros D-H do manipulador foram escolhidos para serem similares aos de robôs industriais vastamente utilizados, como o PUMA 560, e alguns deles foram simplificados no projeto. Isso foi feito por meio da criação de uma configuração onde todas as origens dos sistemas de coordenadas das juntas estão contidas em um mesmo plano, então fora os parâmetros de variável de junta  $\theta_i$  e de rotação de junta  $\alpha_{i-1}$ , têm-se apenas parâmetros que descrevem os comprimentos dos segmentos

<sup>2</sup> As demais tabelas neste trabalho que não possuírem citação de fonte são de autoria própria.

entre as juntas 2 e 3, e entre as juntas 3 e 5, representados por  $a_2$  e  $d_4$ , respectivamente. O parâmetro  ${}^T p_z$  refere-se ao comprimento da garra e será visto mais adiante.

#### 4.1.2 Cinemática Direta

Tendo-se a configuração básica do manipulador com todos os parâmetros D-H encontrados, a cinemática direta pode ser solucionada por meio da aplicação sucessiva da equação (1) em cada par de sistemas de coordenada, os resultados podem então ser inseridos na equação (3) e a descrição do sistema de coordenadas final do robô escrita em termos do sistema base é encontrada.

O operador de transformação em cada par de sistemas de coordenada no manipulador é mostrado nas equações (28) a (33) abaixo:

$${}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (28)$$

$${}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (29)$$

$${}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (30)$$

$${}^3T_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (31)$$

$${}^4T_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (32)$$

$${}^5_6T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (33)$$

A aplicação de todas estas equações na equação (3) resulta na equação (34) a seguir

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} r_{11} &= c_6 [c_5 (s_1 s_4 + c_1 c_{23} c_4) - c_1 s_{23} s_5] + s_6 (s_1 c_4 - c_1 c_{23} s_4) \\ r_{12} &= c_6 (s_1 c_4 - c_1 c_{23} s_4) - s_6 [c_5 (s_1 s_4 + c_1 c_{23} c_4) - c_1 s_{23} s_5] \\ r_{13} &= -s_5 (s_1 s_4 + c_1 c_{23} c_4) - c_1 s_{23} c_5 \\ r_{21} &= -c_6 [c_5 (c_1 s_4 - s_1 c_{23} c_4) + s_1 s_{23} s_5] - s_6 (c_1 c_4 + s_1 c_{23} s_4) \\ r_{22} &= -c_6 (c_1 c_4 + s_1 c_{23} s_4) + s_6 [c_5 (c_1 s_4 - s_1 c_{23} c_4) + s_1 s_{23} s_5] \\ r_{23} &= s_5 (c_1 s_4 - s_1 c_{23} c_4) - s_1 s_{23} c_5 \\ r_{31} &= s_{23} s_4 s_6 - c_6 (c_{23} s_5 + s_{23} c_4 c_5) \\ r_{32} &= s_{23} s_4 c_6 + s_6 (c_{23} s_5 + s_{23} c_4 c_5) \\ r_{33} &= s_{23} c_4 s_5 - c_{23} c_5 \\ p_x &= a_2 c_1 c_2 - c_1 s_{23} d_4 \\ p_y &= a_2 s_1 c_2 - s_1 s_{23} d_4 \\ p_z &= -a_2 s_2 - c_{23} d_4 \end{aligned} \quad (34)$$

onde  $s_{23}$  e  $c_{23}$  são  $\sin(\theta_2 + \theta_3)$  e  $\cos(\theta_2 + \theta_3)$ , respectivamente.

Lembrando que a orientação do manipulador pode também ser escrita em relação aos ângulos de Euler com a equação (4).

#### 4.1.2.1 Cinemática na garra do manipulador

Conforme explicado no Capítulo 2, a solução da cinemática direta pode ser estendida para o sistema de coordenadas da garra do manipulador através da equação (6), e para o robô deste trabalho a matriz de transformação da garra em relação ao sistema de coordenadas {6} tem a forma descrita na equação (35) abaixo

$${}^6T = \begin{bmatrix} 1 & 0 & 0 & {}^T p_x \\ 0 & 1 & 0 & {}^T p_y \\ 0 & 0 & 1 & {}^T p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (35)$$

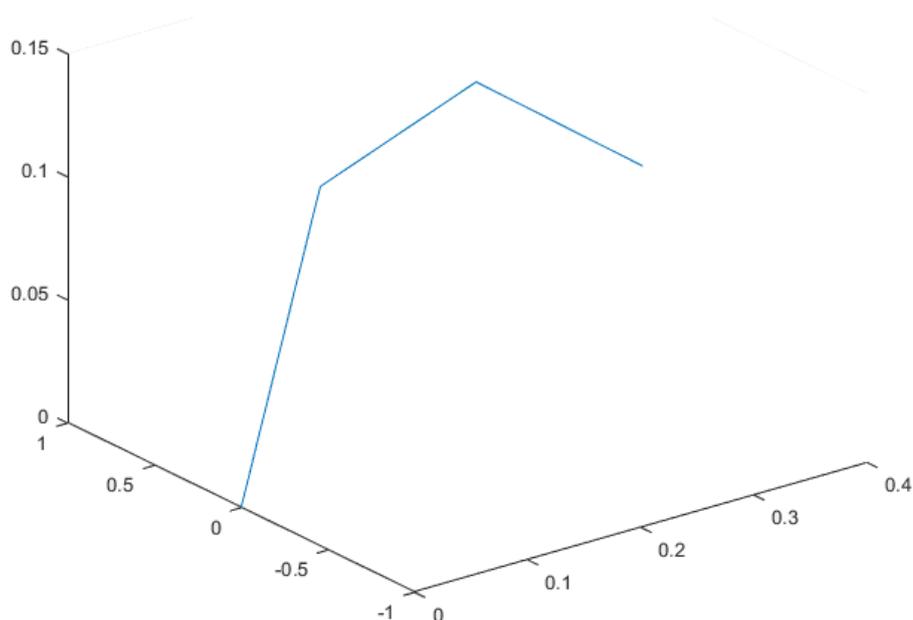
onde apenas as coordenadas da distância entre os sistemas de coordenadas são necessárias, pois a garra terá sempre a mesma orientação do sistema {6}.

#### 4.1.2.2 Implementação em Matlab

A solução da cinemática direta foi implementada no software Matlab, onde uma função foi criada para retornar valores de coordenadas cartesianas quando ângulos de junta são inseridos, o código da função é mostrado no Apêndice A. Explicando mais detalhadamente, ele recebe os seis valores das variáveis de junta e retorna uma matriz com as posições  $p_x$ ,  $p_y$ ,  $p_z$  da garra do manipulador e a sua orientação por ângulos de Euler na ordem  $\alpha, \beta, \gamma$ . Ele serve para verificar como a estrutura simplificada do robô se comporta quando inseridos valores diversos de variáveis de junta. Observa-se que neste código a cinemática direta é resolvida diretamente pela equação (1).

Para facilitar ainda mais a visualização da configuração do manipulador a cada conjunto de variáveis de junta inseridos, uma segunda função em Matlab foi criada, ela resolve a cinemática direta, e embora não transforme a orientação da garra para ângulos de Euler, ela plota em um gráfico tridimensional do manipulador. A função encontra-se no Apêndice B deste trabalho.

Como um exemplo, segue um gráfico, mostrado na Figura 11, gerado pela função para as variáveis de junta  $\theta = [0; -60; -40; 0; 30; 0]$ :



**Figura 11 – Gráfico da configuração do manipulador para as variáveis de junta**

#### **4.1.3 Cinemática Inversa**

A solução da cinemática inversa será feita no estilo de (PAUL, 1981), e se baseia em manipulações e análises das matrizes de transformação provindas da própria cinemática direta.

Algumas matrizes de transformação específicas serão necessárias para a solução da cinemática inversa, elas são rapidamente calculadas nas equações (36) a (41):

$${}^1_6T = {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot {}^4_5T \cdot {}^5_6T = \begin{bmatrix} {}^1r_{11} & {}^1r_{12} & {}^1r_{13} & {}^1p_x \\ {}^1r_{21} & {}^1r_{22} & {}^1r_{23} & {}^1p_y \\ {}^1r_{31} & {}^1r_{32} & {}^1r_{33} & {}^1p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} {}^1r_{11} &= c_6(c_{23}c_4c_5 - s_{23}s_5) - c_{23}s_4s_6 \\ {}^1r_{12} &= s_6(s_{23}s_5 - c_{23}c_4c_5) - c_{23}s_4c_6 \\ {}^1r_{13} &= -s_{23}c_5 - c_{23}c_4s_5 \\ {}^1r_{21} &= -c_4s_6 - s_4c_5c_6 \\ {}^1r_{22} &= s_4c_5s_6 - c_4c_6 \\ {}^1r_{23} &= s_4s_5 \\ {}^1r_{31} &= s_{23}s_4s_6 - c_6(c_{23}s_5 + s_{23}c_4c_5) \\ {}^1r_{32} &= s_{23}s_4c_6 + s_6(c_{23}s_5 + s_{23}c_4c_5) \\ {}^1r_{33} &= s_{23}c_4s_5 - c_{23}c_5 \\ {}^1p_x &= a_2c_2 - d_4s_{23} \\ {}^1p_y &= 0 \\ {}^1p_z &= -d_4c_{23} - a_2s_2 \end{aligned} \quad ; \quad (36)$$

$${}^3_6T = {}^3_4T \cdot {}^4_5T \cdot {}^5_6T = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (37)$$

$${}^0_2T = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & 0 \\ s_1c_2 & -s_1s_2 & c_1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (38)$$

$${}^0_3T = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & a_2c_1c_2 \\ s_1c_{23} & -s_1s_{23} & c_1 & a_2s_1c_2 \\ -s_{23} & -c_{23} & 0 & -a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (39)$$

$${}^0_4T = \begin{bmatrix} s_1s_4 + c_1c_{23}c_4 & s_1c_4 - c_1c_{23}s_4 & -c_1s_{23} & a_2c_1c_2 - c_1s_{23}d_4 \\ -c_1s_4 + s_1c_{23}c_4 & -c_1c_4 - s_1c_{23}s_4 & -s_1s_{23} & a_2s_1c_2 - s_1s_{23}d_4 \\ -s_{23}c_4 & s_{23}s_4 & -c_{23} & -a_2s_2 - c_{23}d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (40)$$

$${}^0_5T = \begin{bmatrix} c_5(s_1s_4 + c_1c_{23}c_4) - c_1s_{23}s_5 & -s_5(s_1s_4 + c_1c_{23}c_4) - c_1s_{23}c_5 & c_1c_{23}s_4 - s_1c_4 & a_2c_1c_2 - c_1s_{23}d_4 \\ -c_5(c_1s_4 - s_1c_{23}c_4) - s_1s_{23}s_5 & s_5(c_1s_4 - s_1c_{23}c_4) - s_1s_{23}c_5 & s_1c_{23}s_4 + c_1c_4 & a_2s_1c_2 - s_1s_{23}d_4 \\ -c_{23}s_5 - s_{23}c_4c_5 & -c_{23}c_5 + s_{23}c_4s_5 & -s_{23}s_4 & -a_2s_2 - c_{23}d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (41)$$

Para encontrar a primeira variável de junta, a equação (3) é reorganizada, como mostra a equação (42)

$${}^0_1T^{-1} \cdot {}^0_6T = {}^1_6T, \quad (42)$$

e os termos da direita para a esquerda nesta equação são obtidos a partir das equações (28), (34) e (36), respectivamente.

Analisando os elementos de (2,4) em ambos os lados da equação (42), tem-se a seguinte relação:

$$-s_1p_x + c_1p_y = 0. \quad (43)$$

Resolvendo a equação (43) em função de  $\theta_1$ , obtém-se a solução para esta primeira variável, como mostra a equação (44) abaixo

$$\begin{aligned} \theta_1 &= \text{atan } 2(p_y, p_x) \\ \theta_1 &= \begin{cases} \theta_1 + \pi & \text{se } \theta_1 < 0 \\ \theta_1 - \pi & \text{se } \theta_1 > 0 \end{cases} \end{aligned} \quad (44)$$

Esta equação possui duas soluções possíveis, representadas por  $\theta_1$  e  $\theta_1'$ .

Retornando à equação (42) e equacionando os elementos de (1,4) e (3,4) em ambos os seus lados, obtêm-se

$$\begin{aligned} c_1p_x + s_1p_y &= a_2c_2 - d_4s_{23} \\ p_z &= -a_2s_2 - d_4c_{23} \end{aligned} \quad (45)$$

Elevando-se ao quadrado as equações (43) e (45) e somando-as, uma relação para a terceira variável de junta pode ser obtida, como observa-se na equação (46)

$$\theta_3 = \text{asen} \left( -\frac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - d_4^2}{2a_2d_4} \right), \quad (46)$$

$$\theta_3' = -\theta_3$$

que fornece duas soluções possíveis,  $\theta_3$  e  $\theta_3'$ .

Considerando a equação (3) novamente, podemos agora organizá-la no formato a seguir

$${}^0_3T^{-1} \cdot {}^0_6T = {}^3_6T, \quad (47)$$

onde os termos, da direita para a esquerda, podem ser obtidos por meio das equações (39), (34) e (37).

Analisando os elementos (1,4) e (2,4) em ambos os lados da equação (47), obtêm-se a equação (48)

$$\begin{aligned} c_1c_{23}p_x + s_1c_{23}p_y - s_{23}p_z - a_2c_3 &= 0 \\ -c_1s_{23}p_x - s_1s_{23}p_y - c_{23}p_z + a_2s_3 &= d_4 \end{aligned} \quad (48)$$

E resolvendo esta relação simultaneamente para  $s_{23}$  e  $c_{23}$ , a seguinte relação é obtida

$$\theta_{23} = \text{atan} 2 \left[ -a_2c_3p_z + (a_2s_3 - d_4)(c_1p_x + s_1p_y), (a_2s_3 - d_4)p_z + (a_2c_3)(c_1p_x + s_1p_y) \right], \quad (49)$$

e esta equação (49) fornece a solução para a segunda variável de junta, pois sabe-se que existe a relação angular mostrada na equação (50) abaixo

$$\theta_2 = \theta_{23} - \theta_3. \quad (50)$$

Para a quarta variável de junta, são tomados os elementos (1,3) e (3,3) de ambos os lados da equação (47), que resultam na equação (51)

$$\begin{aligned} c_1c_{23}r_{13} + s_1c_{23}r_{23} - s_{23}r_{33} &= -c_4s_5 \\ -s_1r_{13} + c_1r_{23} &= s_4s_5 \end{aligned} \quad (51)$$

Esta equação pode ser solucionada para  $\theta_4$  contanto que  $s_5 \neq 0$ , resultando na equação (52) abaixo

$$\theta_4 = \text{atan } 2(-s_1 r_{13} + c_1 r_{23}, -c_1 c_{23} r_{13} - s_1 c_{23} r_{23} + s_{23} r_{33}). \quad (52)$$

Quando  $\theta_5 = 0$ , a equação não pode ser obtida, pois o manipulador encontra-se em uma posição singular, e, caso isto ocorra, o que importa é apenas a soma ou a diferença entre  $\theta_4$  e  $\theta_6$ , sendo geralmente escolhido um valor arbitrário para  $\theta_4$  com o objetivo de que  $\theta_6$  possa então ser calculado.

A quinta variável de junta é obtida de maneira semelhante às anteriores, toma-se a equação (3) novamente e reorganizam-se seus termos na seguinte forma:

$${}^0T^{-1} \cdot {}_6^0T = {}_5^4T \cdot {}_6^5T. \quad (53)$$

Os termos da equação (53), da direita para a esquerda, podem ser obtidos por meio das equações (40), (34), (32) e (33), respectivamente.

Analisando os elementos de (1,3) e (3,3) em ambos os lados da equação, têm-se a relação

$$\begin{aligned} -(c_1 c_{23} c_4 + s_1 s_4) r_{13} - (s_1 c_{23} c_4 - c_1 s_4) r_{23} + (s_{23} c_4) r_{33} &= s_5, \\ (-c_1 s_{23}) r_{13} - (s_1 s_{23}) r_{23} - c_{23} r_{33} &= c_5 \end{aligned}, \quad (54)$$

esta equação (54) pode ser resolvida facilmente em função de  $\theta_5$ , como é mostrado na equação abaixo

$$\theta_5 = \text{atan } 2(s_5, c_5). \quad (55)$$

Para a variável final de junta, novamente a equação (3) é utilizada e modificada para ficar na forma a seguir

$${}^5T^{-1} \cdot {}_0^6T = {}_6^5T, \quad (56)$$

onde os termos, da direita para a esquerda, podem ser obtidos por meio das equações (41), (34) e (33).

Utilizando os elementos (3,1) e (1,1) em ambos os lados da equação (56), o seguinte resultado é obtido:

$$\begin{aligned} s_6 &= -(c_1 c_{23} s_4 - s_1 c_4) r_{11} - (s_1 c_{23} s_4 + c_1 c_4) r_{21} + (s_{23} s_4) r_{31} \\ c_6 &= r_{11} [(c_1 c_{23} c_4 + s_1 s_4) c_5 - c_1 s_{23} s_5] + r_{21} [(s_1 c_{23} c_4 - c_1 s_4) c_5 - s_1 s_{23} s_5] - r_{31} (s_{23} c_4 c_5 + c_{23} s_5). \end{aligned} \quad (57)$$

A equação (58), para a variável de junta  $\theta_6$ , é então facilmente obtida:

$$\theta_6 = \text{atan2}(s_6, c_6). \quad (58)$$

Por conta das duas soluções da equação (44) e das duas soluções da equação (46), tem-se até agora quatro soluções distintas de conjuntos de ângulos para a cinemática inversa, porém mais quatro soluções ainda podem ser obtidas girando-se o punho do manipulador, como segue:

$$\begin{aligned} \theta_4' &= \theta_4 + 180^\circ \\ \theta_5' &= -\theta_5 \\ \theta_6' &= \theta_6 + 180^\circ \end{aligned} \quad (59)$$

Considerando a equação (59), conclui-se que o manipulador possui oito soluções para a cinemática inversa, algumas delas podem ser inviáveis devido a limitações angulares de juntas.

#### 4.1.3.1 Ângulos Z-Y-X de Euler na Cinemática Inversa

Como pôde ser observado nas equações desenvolvidas, elas são escritas em função diretamente dos parâmetros de orientação cartesianos do manipulador ( $r_{11}$ ,  $r_{12}$ , etc), e como foi comentado no Capítulo 2, estes parâmetros não são práticos de serem inseridos na geração de determinada orientação, então para descrevê-la serão utilizados os ângulos de Euler, e os parâmetros de orientação cartesianos serão derivados com base nestes ângulos a partir da equação a seguir, que é uma adaptação da equação (5) descrita no Capítulo 2 para o caso em que se desejam os parâmetros cartesianos no sistema de coordenadas de base

$${}^0_6R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = {}^0_3R \cdot \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \quad (60)$$

Lembrando que o termo  ${}^0_3R$  pode ser obtido diretamente pela equação (61) após a solução inversa de  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  já tiver sido calculada.

#### 4.1.3.2 Implementação em Matlab

A solução inversa da cinemática obtida foi também implementada em Matlab, ela é uma ferramenta importante de controle do manipulador por conseguir transformar um ponto no espaço cartesiano que se queira alcançar em variáveis de junta. É possível visualizá-la no Apêndice C.

O seu código permite a obtenção das oito soluções da cinemática inversa com base na inserção de coordenadas cartesianas e ângulos de Euler da última junta do manipulador.

Para o manipulador desenvolvido neste trabalho os motores só permitem uma variação de  $\pm 90^\circ$  em seu eixo, então a maior parte das soluções é eliminada quando esta restrição é incluída, e das soluções possíveis restantes pode ser escolhida a mais próxima da configuração atual através de algumas lógicas de programação.

#### 4.1.3.3 Área de trabalho

Como comentado previamente no Capítulo 2, saber a delimitação da área de trabalho do manipulador é algo muito importante, pois apenas dentro dela a cinemática inversa possui soluções, (exceto nas regiões de singularidades). Com base nos parâmetros D-H escolhidos e considerando que as juntas só podem rotacionar  $180^\circ$ , como será visto no Capítulo 4, a área de trabalho do manipulador tem o formato representado na Figura 12 abaixo. Observa-se que mesmo que a variável de junta  $\theta_1$  só possa rotacionar  $180^\circ$ , pelo fato de a segunda variável de junta,  $\theta_2$ , também poder rotacionar nesta mesma quantidade, uma região semiesférica de  $360^\circ$  pode ser alcançada.

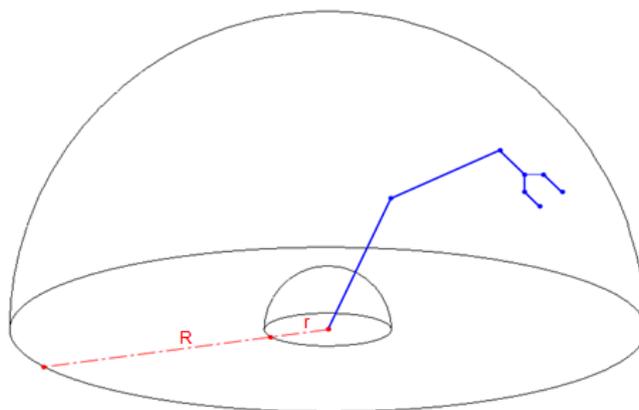


Figura 12 – Área de trabalho do manipulador

Como pode ser observada na figura, a área de trabalho do manipulador é a região semiesférica compreendida entre uma semiesfera de raio  $R$  e uma semiesfera de raio  $r$ , cujos valores são dependentes dos parâmetros D-H e também das coordenadas do final da garra que foram descritas na equação (35). Eles podem ser determinados pelas equações (62) e (63):

$$R = a_2 + d_4 + {}^T p_z; \quad (62)$$

$$r = a_2 - (d_4 + {}^T p_z). \quad (63)$$

Para um manipulador onde  $d_4 + {}^T p_z \geq a_2$  a semiesfera menor é eliminada e a área de trabalho passa a ser apenas a semiesfera maior, que é o caso do manipulador projetado neste trabalho, como será explicado mais adiante neste Capítulo. O volume da semiesfera pode ser obtido pela equação (64)

$$V_{Trabalho} = \frac{2}{3} \pi (a_2 + d_4 + {}^T p_z)^3 \quad (64)$$

#### 4.1.4 Jacobiano

Para conseguir determinar as velocidades cartesianas do manipulador em função das velocidades angulares, e vice-versa, encontrou-se o seu Jacobiano. O método de obtenção foi o da equação (10) com  $n=0$  e as soluções parciais são mostradas nas equações (65) a (69) abaixo:

$$p = \begin{bmatrix} c_1 (a_2 c_2 - s_{23} d_4) \\ s_1 (a_2 c_2 - s_{23} d_4) \\ -a_2 s_2 - c_{23} d_4 \end{bmatrix}; \quad (65)$$

$$\frac{\partial p}{\partial \theta_1} = \begin{bmatrix} -s_1 (a_2 c_2 - s_{23} d_4) \\ c_1 (a_2 c_2 - s_{23} d_4) \\ 0 \end{bmatrix}; \quad (66)$$

$$\frac{\partial p}{\partial \theta_2} = \begin{bmatrix} c_1 (-a_2 s_2 - d_4 c_{23}) \\ s_1 (-a_2 s_2 - d_4 c_{23}) \\ -a_2 c_2 + d_4 s_{23} \end{bmatrix}; \quad (67)$$

$$\frac{\partial p}{\partial \theta_3} = \begin{bmatrix} -c_1(d_4 c_{23}) \\ -s_1(d_4 c_{23}) \\ d_4 s_{23} \end{bmatrix}; \quad (68)$$

$$\frac{\partial p}{\partial \theta_4} = \frac{\partial p}{\partial \theta_5} = \frac{\partial p}{\partial \theta_6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad (69)$$

Lembrando que o valor do vetor  $\mathbf{p}$  foi obtido por meio da quarta coluna de  ${}^0_6T$  na equação (34) e os vetores  $\mathbf{a}$  podem ser obtidos, todos em relação ao sistema de coordenadas de base, a partir da terceira coluna das matrizes de transformação que foram obtidas nas equações (28) e (38) a (41).

O Jacobiano da última junta do manipulador, escrita em função do sistema de coordenadas de base foi então obtida e é mostrado na equação (70)

$${}^0J = \begin{bmatrix} -s_1(a_2 c_2 - s_{23} d_4) & -c_1(a_2 s_2 + c_{23} d_4) & -c_1 c_{23} d_4 & 0 & 0 & 0 \\ c_1(a_2 c_2 - s_{23} d_4) & -s_1(a_2 s_2 + c_{23} d_4) & -s_1 c_{23} d_4 & 0 & 0 & 0 \\ 0 & -a_2 c_2 + s_{23} d_4 & s_{23} d_4 & 0 & 0 & 0 \\ 0 & -s_1 & -s_1 & -c_1 s_{23} & c_1 c_{23} s_4 - s_1 c_4 & -s_5(s_1 s_4 + c_1 c_{23} c_4) - c_1 s_{23} c_5 \\ 0 & c_1 & c_1 & -s_1 s_{23} & s_1 c_{23} s_4 + c_1 c_4 & s_5(c_1 s_4 - s_1 c_{23} c_4) - s_1 s_{23} c_5 \\ 1 & 0 & 0 & -c_{23} & -s_{23} s_4 & s_{23} c_4 s_5 - c_{23} c_5 \end{bmatrix}. \quad (70)$$

#### 4.1.4.1 Implementação em matlab

O Jacobiano do manipulador, obtida na equação (70), foi implementado em uma função bem simples no Matlab, que é apresentada no Apêndice D.

Sua inversa pode também ser facilmente obtida por meio do mesmo software (quando o manipulador não está em um ponto de singularidade). O Jacobiano obtido por este código é de grande utilidade para a análise da relação entre velocidades angulares e cartesianas no manipulador em qualquer configuração de variáveis de junta.

#### 4.1.4.2 Singularidades do manipulador

Para encontrar os pontos de singularidade do manipulador a equação (12) foi utilizada em conjunto com o Jacobiano da equação (70), o resultado foi:

$$-a_2 s_5 d_4 (a_2 c_2 - s_{23} d_4) (c_2 c_{23} + s_2 s_{23}) = 0. \quad (71)$$

A equação (71) é zerada quando qualquer um dos três termos sendo multiplicados torna-se zero, então tem-se as seguintes equações que representam pontos de singularidade:

$$a_2 d_4 s_5 = 0 \quad (72)$$

$$a_2 c_2 - s_{23} d_4 = 0 \quad (73)$$

$$c_2 c_{23} + s_2 s_{23} = 0 \quad (74)$$

A singularidade quando  $\theta_5 = 0$  é facilmente visível agora, por meio da equação (72), a qual só pode ser zerada desta maneira, pois os seus outros dois termos são comprimentos dos segmentos, que devem ser sempre reais e positivos.

As equações (73) e (74) mostram que para determinados valores específicos de variáveis de junta, quando combinados em determinada proporção, geram uma singularidade, e este tipo de singularidade é o menos intuitivo de se visualizar, diferentemente das singularidades na superfície da área de trabalho.

O conjunto de equações desenvolvido pode ser usado para prevenir que se atinjam pontos de singularidade, ou até mesmo pontos próximos a singularidades, isso pode ser feito da seguinte maneira: para todos os novos valores de  $\theta_i$  em determinada trajetória, o conjunto de equações é calculado, e observa-se se ele está ou não muito próximo de zero. Caso esteja, as chances são grandes de a posição angular do manipulador estar muito próxima de singularidades, o que pode gerar picos de velocidade.

## 4.2 Dinâmica

A análise dinâmica do manipulador foi realizada com o objetivo de se verificar quais os torques necessários em cada uma de suas juntas para gerar certa quantidade de movimento e aceleração e então serem selecionados os motores. Porém, ao mesmo tempo em que a seleção de motores e adequação da estrutura do robô de acordo com ela dependem da análise dinâmica, a análise dinâmica depende dos parâmetros de massa e inércia, que são resultados diretos de como a estrutura do robô foi projetada, ou seja, este é um processo iterativo. Então, embora o projeto

estrutural do manipulador esteja sendo descrito na sequência com motores já selecionados e uma estrutura definida, é importante se ter em mente que ele foi o resultado de algumas iterações com a análise dinâmica do manipulador.

#### 4.2.1 Projeto estrutural do manipulador robótico

Com o objetivo de obterem-se os parâmetros dinâmicos de massa e inércia do manipulador, ele foi desenhado utilizando-se o software SolidWorks. A estrutura final desenvolvida é mostrada na Figura 13 abaixo, onde se procurou manter a simplicidade mecânica com a ausência de, por exemplo, relações de engrenamento nas juntas, para facilitar os cálculos do projeto neste primeiro momento e tentar manter a estrutura do manipulador o mais próxima possível das dos manipuladores de baixo custo existentes no mercado. Na sequência algumas justificativas de escolha de dimensões, materiais, componentes de fixação, tipo de motores e de microcontrolador também serão dadas.

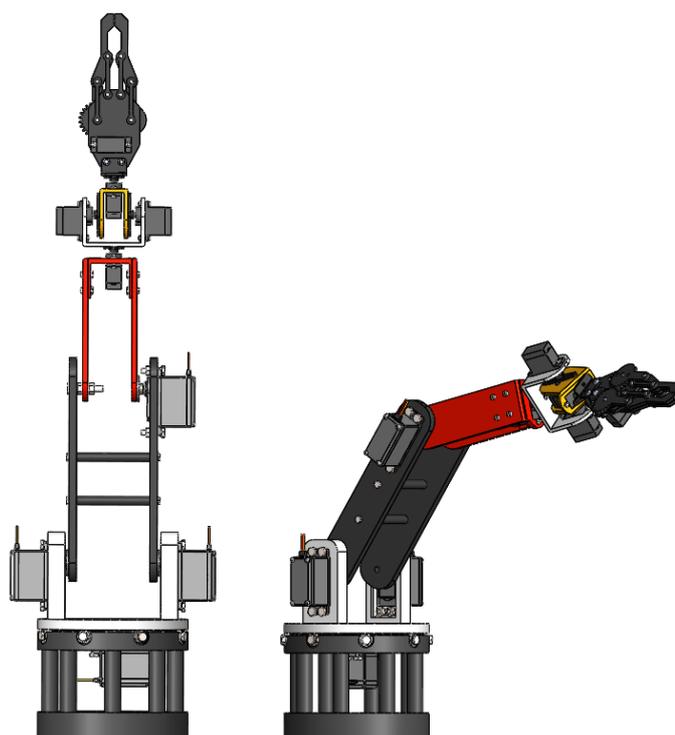


Figura 13 – Vista 3D do manipulador robótico projetado

##### 4.2.1.1 Determinação do comprimento dos segmentos

Os parâmetros D-H referentes ao comprimento dos segmentos ( $a_2$  e  $d_4$ ), e o tipo e comprimento da garra ( ${}^T p_z$ ) foram selecionados através de uma análise de

benchmarking com vários manipuladores robóticos de baixo custo, que está mostrada na Tabela 2 abaixo:

**Tabela 2 – Benchmarking do alcance de manipuladores de baixo custo**

<i>Manipulador</i>	<i>Alcance Vertical (mm)</i>	<i>Alcance Horizontal (mm)</i>	<i>Nº de graus de liberdade</i>	<i>Fonte</i>
Robotic arm with 6 DOF	430	340	6	<a href="https://www.thingiverse.com/thing:30163">https://www.thingiverse.com/thing:30163</a>
OWI Robotic Arm	380	310	5	<a href="http://www.owirobot.com/robotic-arm-edge-1/">http://www.owirobot.com/robotic-arm-edge-1/</a>
DAGU Robotic Arm	350	310	5	<a href="http://www.dagurobot.com/goods.php?id=68">http://www.dagurobot.com/goods.php?id=68</a>
Phantom X Pincher	390	-	5	<a href="http://www.trossenrobotics.com/p/PhantomX-Pincher-Robot-Arm.aspx">http://www.trossenrobotics.com/p/PhantomX-Pincher-Robot-Arm.aspx</a>
uArm Swift	-	320	4	<a href="http://www.ufactory.cc/#/en/uarmswift">http://www.ufactory.cc/#/en/uarmswift</a>
AL5D Robotic Arm	480	260	4	<a href="http://www.lynxmotion.com/c-130-al5d.aspx">http://www.lynxmotion.com/c-130-al5d.aspx</a>

Pode se dizer que alcance médio destes manipuladores está entre 300 e 400mm junto com a garra, então considerando que esta é uma dimensão boa para a manipulação de objetos em um meio doméstico, comercial ou educacional (ambiente para o qual eles foram projetados), os parâmetros foram selecionados da seguinte maneira:

**Tabela 3 – Dimensões dos segmentos do manipulador**

<i>Parâmetro</i>	<i>Dimensão (mm)</i>
Comprimento do primeiro segmento ( $a_2$ )	140
Comprimento do segundo segmento ( $d_4$ )	140
Comprimento da garra ( ${}^T p_z$ )	157

O comprimento dos segmentos foi selecionado de forma que  $a_2 = d_4$  e foi escolhido um tipo de garra semelhante a do DAGU Robotic Arm da Tabela 2, a intenção foi a de se ter um manipulador resultante com alcance um pouco maior do que o dos mostrados na tabela, de modo a se ter um maior volume de área de trabalho.

#### 4.2.1.2 Seleção dos motores e do microcontrolador

De modo a se ter um manipulador semelhante aos de baixo custo existentes, servomotores de pequeno porte, como o da Figura 14 foram utilizados.



Figura 14 – Servomotor TowerPro MG946R

Fonte: (Site Oficial da TowerPro)

A Tabela 4 abaixo mostra quais motores foram utilizados em cada uma das juntas do robô.

Tabela 4 – Servomotores utilizados no manipulador

<i>Junta (i)</i>	<i>Modelo</i>	<i>Massa</i>	<i>Quantidade</i>	<i>Torque total</i>
1	MG946R	55g	1	13 kg/cm
2	MG946R	55g	2	26 kg/cm
3	MG946R	55g	1	13 kg/cm
4	MG90S	13,4g	1	2,2 kg/cm
5	MG90S	13,4g	2	4,4 kg/cm
6	SG90	9g	1	1,8 kg/cm
Garra	SG90	9g	1	1,8 kg/cm

A lógica usada na seleção foi a de que quanto mais perto da base do robô, mais torque é necessário e menos influência a massa do motor possui na estrutura geral do manipulador.

A escolha de servomotores deste tipo, embora seja econômica por terem um custo e um peso baixos em relação ao torque gerado, dificulta muito a implementação de qualquer tipo de controle de velocidade e aceleração, pois eles já vêm com um sistema de controle interno que converge para o ângulo que for inserido na sua programação com uma velocidade proporcional à diferença entre o ângulo atual e o ângulo requerido, por meio de um potenciômetro que gera feedback de rotação ao sistema, que deve então gerar pulsos PWM (Pulse Width Modulation) para acionar o motor (SIANG).

Apesar disso, este controle pré-determinado torna a configuração eletrônica do manipulador muito mais simples, não sendo necessário o uso de encoders de posição angular e de uma implementação de algum tipo de controle mais complexo na programação. Para este trabalho, que foca mais em tentar se assimilar manipuladores de baixo custo, isso é conveniente e também foi um dos motivos para a sua escolha.

Como será comentado no Capítulo 5, um dos aspectos do manipulador que deve ser melhorado em trabalhos futuros é esta situação dos motores não possibilitarem um controle livre de velocidades e acelerações.

Por este tipo de motor ser controlado por sinais PWM ele tem uma grande compatibilidade com o microcontrolador Arduino que pode gerar este tipo de sinal, então este foi o escolhido para controlar o robô. Porém ele possui algumas limitações, como será comentado também no Capítulo 5.

#### **4.2.1.3 Materiais e componentes de fixação**

Todos os componentes estruturais do manipulador foram projetados como sendo de plástico ABS, e a razão para isso foi a facilidade de fabricação e o preço mais em conta do que materiais metálicos como alumínio, além da menor densidade e, conseqüentemente, peso. Enquanto este plástico possui uma densidade média de  $1,07 \text{ g/cm}^3$ , o alumínio, por exemplo, tem uma densidade média de  $2,7 \text{ g/cm}^3$ . A

fabricação fica muito mais simples porque se o material for ABS todos os componentes podem ser impressos em impressoras 3D.

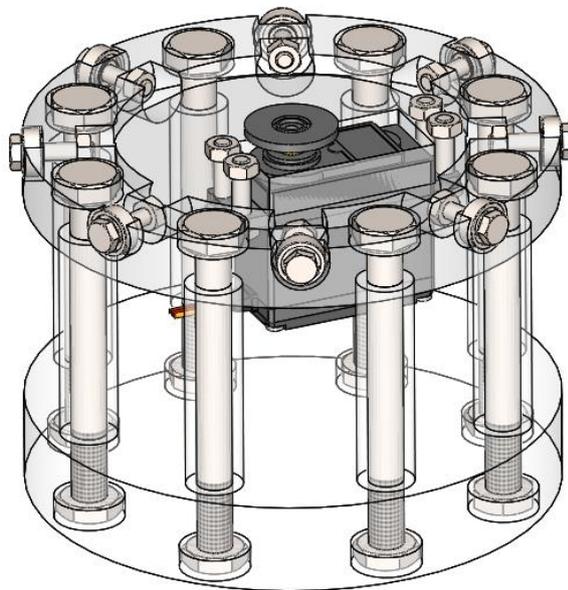
Os componentes de fixação utilizados foram: parafusos, porcas, pinos e rolamentos, no material padrão que é aço carbono. Na próxima seção estes componentes serão vistos em mais detalhes para cada um dos segmentos.

#### **4.2.1.4 Análise dos segmentos do manipulador**

A seguir cada um dos segmentos do manipulador será analisado separadamente e serão observados os aspectos mecânicos de seu funcionamento. Como já comentado, para a utilização das equações de dinâmica demonstradas no Capítulo 2 são necessários parâmetros dinâmicos (massa e tensor de inércia) para cada um dos segmentos, então eles também serão obtidos.

##### **4.2.1.4.1 Segmento 0**

A Figura 15 mostra a base do manipulador, e embora os seus parâmetros dinâmicos não sejam necessários por ser um segmento fixo, ele possui um aspecto interessante de funcionamento mecânico.



**Figura 15 – Segmento 0 do manipulador**

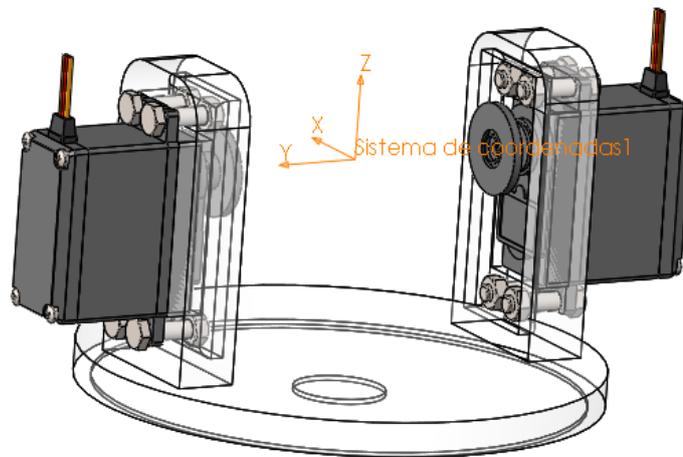
Como pode ser observado na figura, existem oito rolamentos na parte superior do segmento. Isso foi feito para dar mais estabilidade ao manipulador, pois todo o seu peso será colocado sobre esta base e caso ela fosse ligada no próximo

segmento apenas pelo acoplamento do servo motor, os diversos momentos resultantes do movimento do robô causariam uma grande instabilidade em toda a estrutura.

Tanto os rolamentos quanto o servomotor e os componentes estruturais de ABS foram fixados por meio de porcas e parafusos.

#### 4.2.1.4.2 Segmento 1

O segmento 1 é mostrado em detalhes na Figura 16 a seguir.



**Figura 16 – Segmento 1 do manipulador**

O sistema de coordenadas mostrado na figura está representando os sistemas  $\{0\}$  e  $\{1\}$ , que são coincidentes quando as suas variáveis de junta são zero. A massa desse segmento não gera influência no torque necessário pelo motor do segmento 0, que só sofre a influência dos momentos de inércia causados pela distribuição de massa.

Os parâmetros dinâmicos do segmento foram obtidos pela ferramenta de propriedades de massa do SolidWorks e são mostrados na Tabela 5 abaixo. Vale observar que a localização do centro de massa dos segmentos está escrita no sistema de coordenadas da respectiva junta, como o da Figura 15 para o caso do Segmento 1, por exemplo, e da mesma maneira o tensor de inercias foi rotacionado para ter a mesma orientação deste sistema de coordenadas.

Tabela 5 – Parâmetros dinâmicos do segmento 1

<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>				
<b>Localização do centro de massa (mm)</b>	$p_x = 0$	$I_{xx} = 742895,03$	$I_{xy} = 0$	$I_{xz} = 0$
	$p_y = 0$	$I_{yx} = 0$	$I_{yy} = 242804,64$	$I_{yz} = 2,58$
	$p_z = -26,03$	$I_{zx} = 0$	$I_{zy} = 2,58$	$I_{zz} = 707185,62$
<b>Massa (g)</b>	$m = 258,01$			

## 4.2.1.4.3 Segmento 2

A Figura 17 abaixo retrata a configuração mecânica do segmento 2 do manipulador.

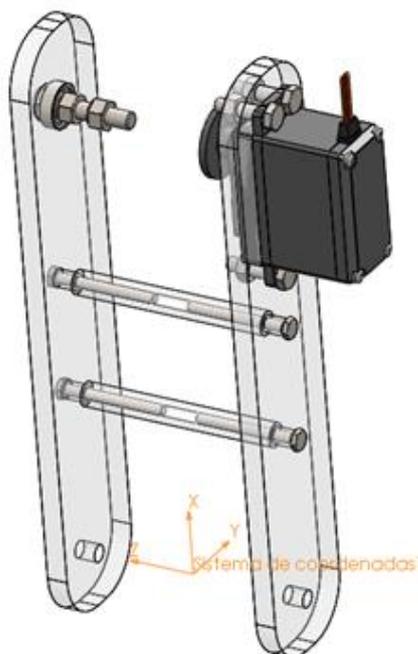


Figura 17 – Segmento 2 do manipulador

O sistema de coordenadas mostrado é o sistema da segunda junta, {2}, e a sua origem é coincidente à origem dos sistemas {0} e {1}.

Observa-se que foram inseridos dois cilindros entre as hastes do segmento, para que elas sempre se movimentem em sincronia. Do lado oposto ao motor na parte superior do segmento existe um parafuso que, através de duas porcas, fará a fixação do próximo segmento neste lado da haste. Um rolamento radial foi colocado

entre a haste do segmento e o parafuso, para dar uma maior estabilidade e minimizar o atrito.

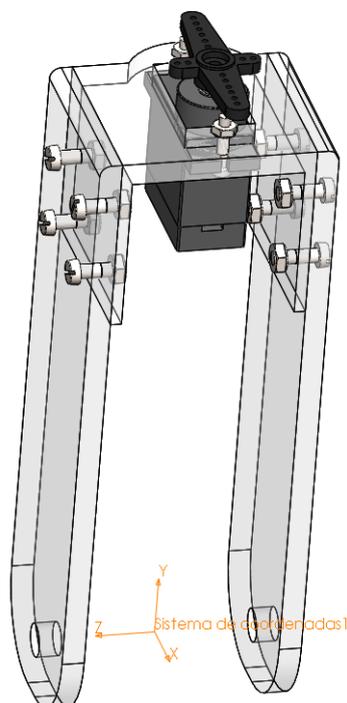
Os parâmetros dinâmicos do segmento são mostrados na Tabela 6.

**Tabela 6 – Parâmetros dinâmicos do segmento 2**

		<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>			
<b>Localização do centro de massa (mm)</b>	$p_x = 96,59$	$I_{xx} = 163347,71$	$I_{xy} = -0,31$	$I_{xz} = -77560,91$	
	$p_y = 0$	$I_{yx} = -0,31$	$I_{yy} = 463549,5$	$I_{yz} = 0,29$	
	$p_z = -16,93$	$I_{zx} = -77560,91$	$I_{zy} = 0,29$	$I_{zz} = 318631,48$	
<b>Massa (g)</b>	$m = 134,46$				

#### 4.2.1.4.4 Segmento 3

No segmento 3, as duas hastes são interligadas a um componente estrutural que acomodará o motor, a Figura 18 exibe a sua configuração.



**Figura 18 – Segmento 3 do manipulador**

O sistema de coordenadas representado na figura é o sistema {3}, os parâmetros dinâmicos do segmento podem ser visualizados na Tabela 7 abaixo.

Tabela 7 – Parâmetros dinâmicos do segmento 3

<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>				
<b>Localização do centro de massa (mm)</b>	$p_x = -1,08$	$I_{xx} = 79553,17$	$I_{xy} = -1498,82$	$I_{xz} = 141,45$
	$p_y = 67,53$	$I_{yx} = -1498,82$	$I_{yy} = 20165,80$	$I_{yz} = -1010,31$
	$p_z = -0,68$	$I_{zx} = -141,45$	$I_{zy} = -1010,31$	$I_{zz} = 69956,53$
<b>Massa (g)</b>	$m = 55,98$			

## 4.2.1.4.5 Segmento 4

O segmento 4 é mostrado em detalhes na Figura 19 a seguir.

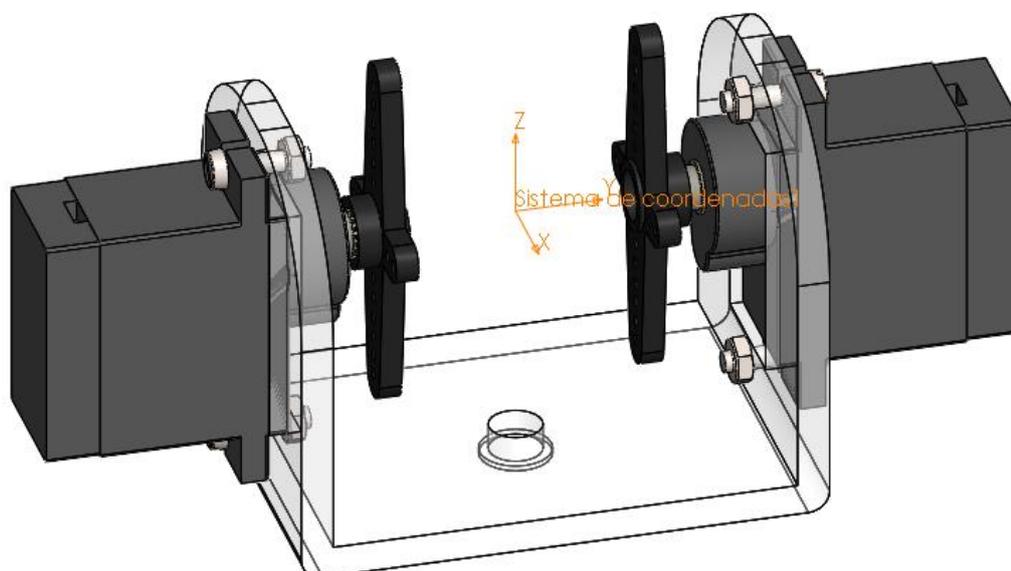


Figura 19 – Segmento 4 do manipulador

O sistema de coordenadas mostrado é o da quarta junta {4}, e os parâmetros dinâmicos do segmento estão exibidos na Tabela 8.

Tabela 8 – Parâmetros dinâmicos do segmento 4

<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>				
<b>Localização do centro de massa (mm)</b>	$p_x = 0$	$I_{xx} = 36165,48$	$I_{xy} = 0$	$I_{xz} = 0$
	$p_y = 0$	$I_{yx} = 0$	$I_{yy} = 5314,19$	$I_{yz} = 0$
	$p_z = -7,38$	$I_{zx} = 0$	$I_{zy} = 0$	$I_{zz} = 33973,34$
<b>Massa (g)</b>	$m = 38,07$			

## 4.2.1.4.6 Segmento 5

A Figura 20 abaixo retrata a configuração mecânica do segmento 2 do manipulador.

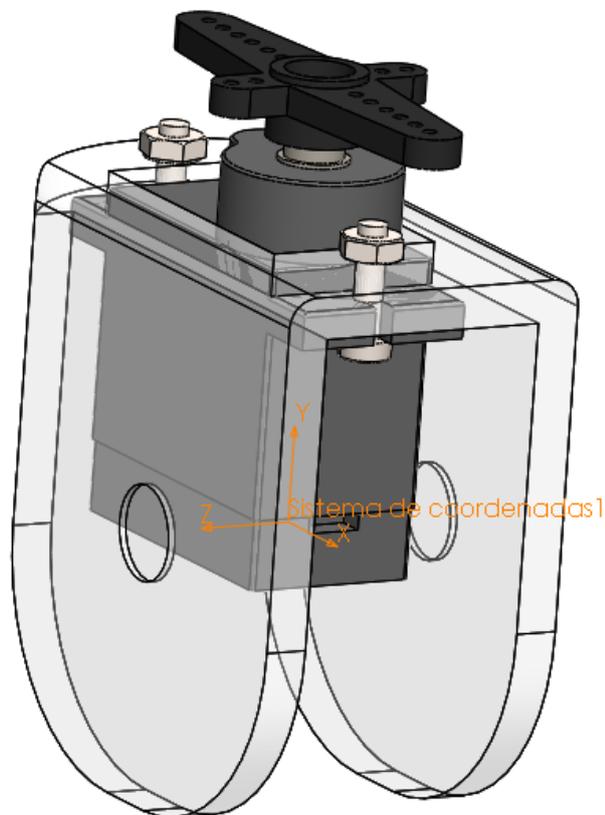


Figura 20 – Segmento 5 do manipulador

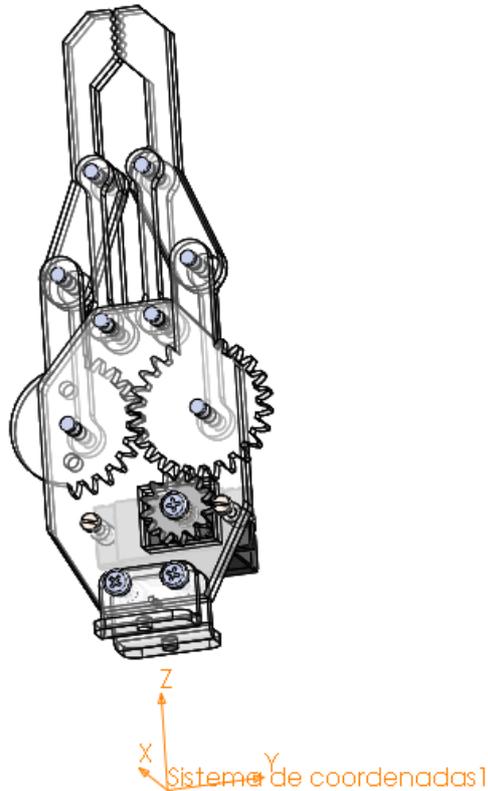
O sistema de coordenada {5} está sendo exibido na figura. Sua origem é coincidente à origem do sistema {4}. Os parâmetros dinâmicos do segmento são mostrados na Tabela 9.

Tabela 9 – Parâmetros dinâmicos do segmento 5

		<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>			
<b>Localização do centro de massa (mm)</b>	$p_x = -2,61$	$I_{xx} = 3643,16$	$I_{xy} = -41,04$	$I_{xz} = 1,48$	
	$p_y = 8,65$	$I_{yx} = -41,04$	$I_{yy} = 2792,90$	$I_{yz} = 0,01$	
	$p_z = 0$	$I_{zx} = 1,48$	$I_{zy} = 0,01$	$I_{zz} = 4455,03$	
<b>Massa (g)</b>	$m = 22,04$				

#### 4.2.1.4.7 Segmento 6

O sexto segmento do manipulador é a garra, que é mostrada na Figura 21 abaixo.



**Figura 21 – Segmento 6 do manipulador**

O sistema de coordenadas representado na figura é o {6}, e sua origem encontra-se fora do segmento 6, sendo coincidente com a origem dos sistemas de coordenada {4} e {5}.

Como foi comentado, a garra selecionada é muito semelhante à garra do DAGU Robotic Arm, o seu funcionamento se dá por meio de engrenamentos. Um servomotor acoplado à garra rotacional uma engrenagem, que por sua vez rotacional outros dois engrenamentos, cada um fixado a uma das pinças da garra.

Os parâmetros dinâmicos do último segmento estão exibidos na Tabela 10 a seguir.

Tabela 10 – Parâmetros dinâmicos do segmento 6

<i>Tensor de Inércia (g*mm<sup>2</sup>)</i>				
<b>Localização do centro de massa (mm)</b>	$p_x = -3,99$	$I_{xx} = 41071,01$	$I_{xy} = -246,59$	$I_{xz} = 3932,00$
	$p_y = 1$	$I_{yx} = -246,59$	$I_{yy} = 40370,90$	$I_{yz} = -808,27$
	$p_z = 76,74$	$I_{zx} = 3932,00$	$I_{zy} = -808,27$	$I_{zz} = 7678,28$
<b>Massa (g)</b>	$m = 42,83$			

#### 4.2.2 Custo dos componentes do manipulador

O custo do manipulador projetado está exibido na Tabela 11 abaixo.

Tabela 11 – Custo do manipulador

<b>Componentes</b>	<b>Detalhes</b>	<b>Custo total (R\$)</b>
Motores	8 un	303,00
Peças estruturais (ABS)	651 cm <sup>3</sup>	111,50
Rolamentos	9 un	70,00
Parafusos	58 un	43,50
Porcas	55 un	8,50
Pinos	8 un	0,50
Microcontrolador	Arduino	130,00
	<b>Total</b>	<b>667,00</b>

Observa-se que maior custo é o dos motores, aproximadamente 3 vezes maior do que o custo de todos os outros tipos de componente (peças estruturais  $\approx$  R\$110,00, componentes de fixação  $\approx$  R\$100,00 e microcontrolador = R\$130,00).

O preço das peças estruturais foi obtido considerando-se o volume de cada uma delas, gerado pelo SolidWorks, considerando-se a densidade do material e sabendo-se que rolos de 500g de material ABS custam em média R\$ 80,00.

### 4.2.3 Solução dinâmica do manipulador

Com os parâmetros dinâmicos de cada segmento determinados, a dinâmica do manipulador é realizada através da formulação iterativa de Newton-Euler, dada pelas equações (19) a (27). Ela foi escolhida porque a solução dinâmica no caso deste trabalho tem como objetivo validar a escolha de componentes e motores e não precisa ser utilizada em nenhum tipo de sistema de controle, pelas limitações dos motores que foram comentadas na seção onde foram selecionados.

Como a formulação de Newton-Euler foi feita para ser calculada facilmente de forma numérica, as suas equações foram implementadas em uma função no software Matlab, que é mostrada no Apêndice E.

A função criada recebe os parâmetros  $\theta_i$ ,  $\dot{\theta}_i$ ,  $\ddot{\theta}_i$  e também um parâmetro de massa, em gramas, referente à massa de um objeto que se deseja inserir na ponta da garra.

Para validar a escolha dos motores, a função foi executada em um caso extremo, quando o peso dos componentes exerce o maior efeito sobre as juntas, que ocorre com  $\theta_3 = 0$  e o restante das variáveis de junta zeradas, a configuração do manipulador nesta situação é retratada na Figura 22.

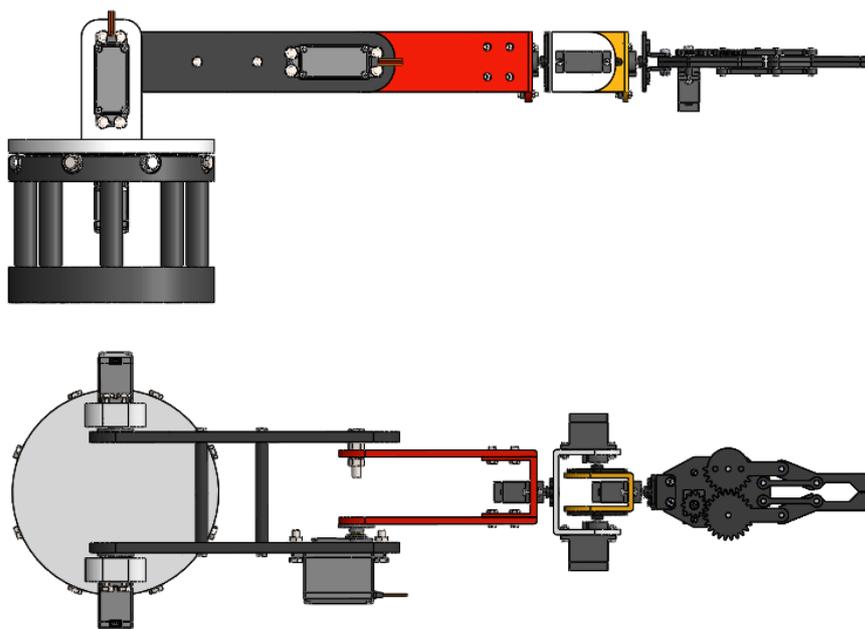


Figura 22 – Duas vistas do manipulador robótico em configuração extrema

Nesta configuração, foram inseridas massas de diferentes pesos na ponta da garra e, além disso também foram consideradas acelerações angulares nas três juntas que “suportam” o peso do mecanismo, e por isso são as mais críticas, as juntas 2, 3 e 5.

A Tabela 12 a seguir mostra os resultados de torque necessário em cada uma das juntas para uma série de massas e acelerações distintas.

**Tabela 12 – Torque nas juntas X Massa na garra e aceleração das juntas**

(continua)

	<i>Junta</i> (i)	$\ddot{\theta}_{2,3,5} = 0$	$\ddot{\theta}_{2,3,5} = -\pi / 4$	$\ddot{\theta}_{2,3,5} = -\pi / 2$	$\ddot{\theta}_{2,3,5} = -\pi$	<i>rad/s<sup>2</sup></i>
<i>m = 150g</i>	1	0	-0.0000	-0.0000	-0.0000	<b>Torque necessário (kg/cm)</b>
	2	-12.1657	-12.8074	-13.4491	-14.7325	
	3	-6.5663	-6.9698	-7.3732	-8.1801	
	4	-0.0043	-0.0045	-0.0048	-0.0053	
	5	-2.6987	-2.8851	-3.0715	-3.4443	
	6	-0.0043	-0.0045	-0.0048	-0.0053	
<i>m = 200g</i>	1	0	-0.0000	-0.0000	-0.0000	<b>Torque necessário (kg/cm)</b>
	2	-14.3498	-15.1471	-15.9444	-17.5391	
	3	-8.0502	-8.5594	-9.0686	-10.0869	
	4	-0.0043	-0.0045	-0.0048	-0.0053	
	5	-3.4823	-3.7246	-3.9668	(-4.4513)	
	6	-0.0043	-0.0045	-0.0048	-0.0053	

Tabela 12 – Torque nas juntas X Massa na garra e aceleração das juntas

(conclusão)

	<i>Junta</i> ( <i>i</i> )	$\ddot{\theta}_{2,3,5} = 0$	$\ddot{\theta}_{2,3,5} = -\pi/4$	$\ddot{\theta}_{2,3,5} = -\pi/2$	$\ddot{\theta}_{2,3,5} = -\pi$	$rad/s^2$
<i>m</i> = 250g	1	0	-0.0000	-0.0000	-0.0000	
	2	-16.5339	-17.4868	-18.4398	-20.3457	
	3	-9.5340	-10.1490	-10.7639	-11.9937	
	4	-0.0043	-0.0045	-0.0048	-0.0053	
	5	-4.2659	(-4.5640)	(-4.8621)	(-5.4583)	
	6	-0.0043	-0.0045	-0.0048	-0.0053	
<i>m</i> = 300g	1	0	-0.0000	-0.0000	-0.0000	
	2	-18.7180	-19.8266	-20.9351	-23.1523	
	3	-11.0179	-11.7386	-12.4592	(-13.9005)	
	4	-0.0043	-0.0045	-0.0048	-0.0053	
	5	(-5.0496)	(-5.4035)	(-5.7574)	(-6.4652)	
	6	-0.0043	-0.0045	-0.0048	-0.0053	

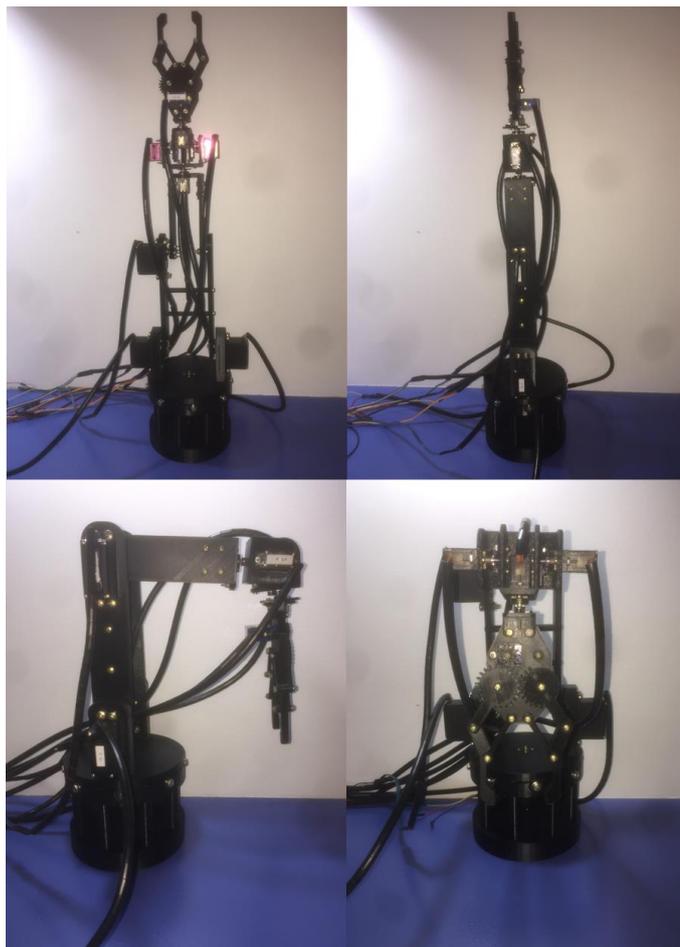
Torque necessário (kg/cm)

Os números entre parênteses na tabela representam os torques que excederam os torques máximos dos motores selecionados, que foram descritos na Tabela 4, e observando-se os resultados obtidos fica claro que o ponto mais crítico em todo o robô é a junta 5, que torna-se incapaz de sustentar a massa de 200g na ponta do manipulador a partir de uma aceleração de  $-\pi$  rad/s<sup>2</sup> e a massa de 250g a partir de  $-\pi/2$  rad/s<sup>2</sup>, sendo que para a massa de 300g até mesmo no caso estático ela não possui torque o suficiente.

De maneira geral, o manipulador consegue movimentar massas de até 200g com boas acelerações, e massas maiores do que isso até 250g com acelerações baixas. Como a aceleração de  $-\pi$  rad/s<sup>2</sup> está no limite da junta 5 para os 200g, pode se dizer que como uso recomendado deve se trabalhar com massas de no máximo 200g e acelerações medianas para massas próximas disso.

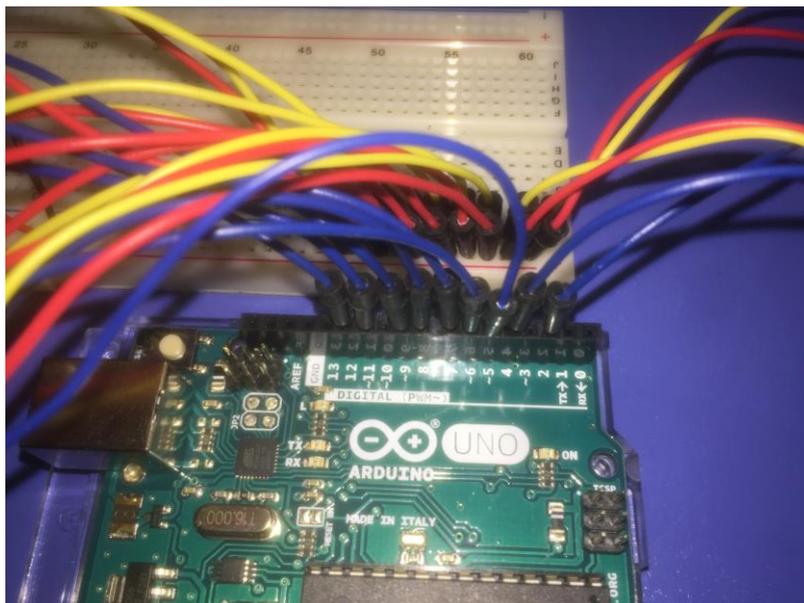
### 4.3 Construção de protótipo do manipulador robótico

Após o desenvolvimento da cinemática, projeto estrutural e dinâmica do manipulador, ele foi construído de acordo com a Figura 15. O resultado final obtido está mostrado na Figura 23. A construção do manipulador foi feita com o objetivo de se obterem ainda mais dados de análise das capacidades e limitações deste tipo de robô.



**Figura 22 – Manipulador robótico construído**

O cabeamento foi feito através de um protoboard em conjunto com o microcontrolador Arduino, e é mostrado na Figura 26.



**Figura 23 – Cabeamento do manipulador robótico**

Observa-se na figura que cabos vermelhos foram utilizados como fase, cabos amarelos como terra e cabos azuis como sinal, sendo que os servomotores foram ligados nos pinos 2,4,5,6,7,8,9,11 e 12 de saída de sinal PWM do Arduino. A fonte de energia utilizada foi a associação de 4 pilhas AA de 1,5V. A Tabela 13 a seguir mostra uma relação de quais motores de junta foram ligados e quais pinos PWM.

**Tabela 13 – Relação juntas X pinos**

<i>Junta (i)</i>	<i>Pino PWM</i>
1	2
2	4 e 5
3	6
4	7
5	8 e 9
6	11
Garra	12

Uma rotina de controle simples foi implementada em Arduino para verificar o funcionamento do manipulador, ela encontra-se no Apêndice F.

O código desenvolvido permite o controle simultâneo de todas as juntas do robô e também da abertura e fechamento da garra. Ele recebe como inputs o valor, em graus, dos ângulos de junta que se deseja atingir, e também a velocidade com que se deseja realizar os movimentos. Uma biblioteca chamada de “VarSpeedServo.h” foi utilizada pois ela permite que se controle a velocidade dos movimentos de maneira bem simples, baseando-se no uso de *delays* temporais, que é a única maneira de se ter algum controle sobre este tipo de motor, pois, como já foi comentado no capítulo de seleção dos motores, eles não possibilitam nenhum tipo mais sofisticado de controle de velocidade ou torque.

## 5 RESULTADOS

Os resultados do projeto do manipulador robótico foram os seguintes:

- Funções em Matlab da cinemática direta para estudo do comportamento do manipulador sob a aplicação de quaisquer variáveis de junta;
- Função em Matlab da cinemática inversa que permite a transformação de qualquer coordenada de posição cartesiana e de orientação de ângulos de Euler em um conjunto de variáveis de junta, para controle dos movimentos do manipulador;
- Equação para o cálculo do volume da área de trabalho do manipulador;
- Equação algébrica do Jacobiano do manipulador e implementação em Matlab de função para obtê-lo numericamente para análise da relação entre velocidades angulares e cartesianas;
- Projeto mecânico em SolidWorks dos componentes e da estrutura do manipulador, e obtenção de seus parâmetros dinâmicos;
- Função em Matlab da dinâmica do robô que permite o cálculo de torques necessários para quaisquer variáveis de junta de posição, velocidade e aceleração, para validação da escolha de motores e estudo dos limites de trabalho do manipulador;
- Protótipo funcional de um manipulador robótico de baixo custo, capaz de realizar movimentos simples a partir de uma rotina de controle em Arduino.

Diversas características finais do manipulador robótico obtido no projeto são exibidas na Tabela 14 a seguir, o volume de trabalho na tabela representa o volume da área de trabalho, obtido pela equação (64).

**Tabela 14 – Características do manipulador robótico**

<i>Graus de Liberdade</i>	<i>Alcance Vertical</i>	<i>Alcance Horizontal</i>	<i>Volume de trabalho</i>	<i>Capacidade máxima</i>	<i>Peso</i>	<i>Nível de controle</i>	<i>Custo (R\$)</i>
6	578 mm	437mm	175dm <sup>3</sup>	200g	1043g	Baixo	667,00

O volume de trabalho, alcance e capacidade máxima do manipulador teoricamente fazem possível o seu uso em atividades variadas em meios domésticos e comerciais, como, por exemplo, na manipulação de objetos leves de escritório como papéis, cadernos e canetas, e objetos em domicílios como louças leves e roupas. O volume de trabalho seria suficiente para manipular objetos ao longo de uma mesa de escritório padrão, porém seria interessante que fosse maior para abranger mais tarefas com facilidade, da mesma maneira, a capacidade máxima de manipulação de 200g, embora tenha certa utilidade, poderia sim ser maior.

De maneira geral, as capacidades mecânicas do manipulador e o seu custo parecem satisfatórios para torna-lo versátil a ponto de viabilizar o seu uso em diversas tarefas de pequeno porte, embora possam com certeza ser aprimoradas.

Por outro lado, ao longo do projeto percebeu-se um fator limitante muito significativo que prejudica toda esta versatilidade da parte mecânica do manipulador: o nível de controle dos movimentos, que foi considerado baixo, pois com o projeto atual apenas pode se controlar a posição das juntas, e não a sua velocidade e aceleração, e isso ocorre principalmente por causa dos motores.

A impossibilidade de um controle mais alto, além de tornar menos preciso todo o funcionamento do manipulador, impede a implementação de fatores importantes no projeto, como um sistema de geração de trajetórias, no qual a aceleração, velocidade e posição do robô são comandadas e podem ser geradas rotas como trajetórias angulares sem picos de velocidades e acelerações e até trajetórias nas quais a garra do manipulador se movimenta em linha reta no plano cartesiano através de movimentos complexos nas juntas. Ser capaz de realizar estas ações seria muito útil para aumentar as aplicações do manipulador. Além disso, muitas outras coisas podem ser feitas com o controle de acelerações e velocidades das juntas, como, por exemplo, um controle do manipulador por forças, onde através de sensores, é possível fazer o robô reconhecer quanto de força está sendo aplicado em sua garra, e através disso fazer o manipulador aplicar determinada força em um objeto ou prevenir que ele bata com força em algum obstáculo. Para mais informações sobre as teorias de geração de trajetórias e de controle de força, consultar (CRAIG, 2005).

Com a construção do protótipo outro fator limitante foi observado: a qualidade dos servomotores utilizados é muito variável e a sua precisão não é muito alta. Os motores foram incapazes de realizar rotações de pequenas dimensões (ângulos menores do que  $10^\circ$ ) e muitos deles apresentaram vibrações elevadas durante seu funcionamento, sem contar que mesmo para rotações em quantidades significativas tiveram dificuldade de posicionar o manipulador com uma boa repetitividade.

## 6 CONCLUSÕES

O presente trabalho relatou o projeto cinemático, estrutural e dinâmico detalhado de um manipulador robótico de baixo custo através do uso da teoria da robótica e de ferramentas computacionais como o SolidWorks e o Matlab. Nele foram relatados procedimentos onde foi possível verificar como cada uma destas ferramentas pode ser útil como auxílio na transformação da teoria em prática.

A análise das características do manipulador resultante obtido tornou possível verificar até qual ponto o seu uso é viável e também identificar quais são as suas limitações e oportunidades de aprimoramento.

Com isso, o trabalho realizado permitiu a verificação de que o uso da teoria da robótica para projetos de manipuladores simples de baixo custo é uma poderosa ferramenta para evidenciar as capacidades e limitações deste tipo de robô e a partir disso identificar possíveis pontos de melhoria. Ele também proporcionou o desenvolvimento de ferramentas em Matlab e criação de procedimentos de projeto característicos que poderão ser úteis no desenvolvimento de outros robôs do mesmo tipo, além de ter resultado na construção de um protótipo funcional para realizar ainda mais análises.

Como soluções para o aprimoramento das características físicas do manipulador, algumas medidas podem ser exploradas para próximos projetos, como:

- Reduzir o comprimento da garra e compensar aumentando o segundo segmento. Como foi visto na análise dinâmica, com a configuração atual a junta que estava limitando o projeto na manipulação de cargas é a junta 5, que possui dois motores MG90S, enquanto a junta 2 que possui 2 MG946R conseguiria atender até o caso mais extremo simulado, de 300g na maior aceleração angular. Com essa mudança nas dimensões o momento sobre a junta 5 diminuiria enquanto o na junta 2 aumentaria;
- Incluir reduções de algum tipo nas juntas e estudar se, embora adicionem complexidade e peso extras ao projeto, elas conseguem aumentar a capacidade máxima do manipulador, em que magnitude e por qual custo;

- Modificar os motores. Procurar opções de motores com maior torque e não muito mais pesados ou caros.

E para proporcionar uma evolução no funcionamento do manipulador no quesito controle de movimentos, os principais pontos que precisariam ser melhorados são:

- Troca dos motores. Embora os servomotores do tipo utilizado neste trabalho sejam vastamente utilizados em manipuladores de baixo custo, o fato deles terem um controle próprio e este controle ser muito simples é um fator muito limitante para o funcionamento mais completo do robô. É necessário estudar alternativas viáveis de motores e controlá-los com o uso de encoders ou algum outro sistema de medição de posição angular;
- Estudar a troca do microcontrolador. O Arduino funciona muito bem com os motores utilizados por ter a capacidade de gerar sinais PWM, porém com o aumento da complexidade do projeto com a troca deles e o surgimento da necessidade da implementação de um controle de algum tipo, uma alternativa ao uso de microcontroladores é o uso de microprocessadores, como o Raspberry Pi, pois por serem capazes de executar sistemas operacionais como o Linux, eles podem se beneficiar do uso de softwares computacionais de robótica que simplificam muito o trabalho por trás do controle do robô, como o ROS (Robot Operating System) e seus complementos.

Por fim, um ponto muito importante para a viabilização de qualquer uso de manipuladores em meios domésticos é o nível de dificuldade de uso. Os métodos de movimentação usuais destes robôs, independentemente do tipo de controle, são geralmente ou movimentação de juntas para atingir-se um ponto cartesiano desejado, ou a definição de um ponto cartesiano desejado e a movimentação das juntas para atingí-lo, problemas que já foram descritos na cinemática direta e inversa. Porém, em uma situação cotidiana, o ponto que o manipulador precisa atingir depende muito da situação e é extremamente variável, não podendo ser pré-configurado como em muitos processos industriais, e também não se pode exigir

que o usuário movimenta junta por junta do manipulador em cada situação, pois isso não seria nada prático.

Então existe um problema a ser resolvido, o de fazer o manipulador saber, em determinado instante, qual é o destino de sua cinemática inversa, destino este que pode variar dependendo da situação, e para isso pode ser usado, por exemplo, um sistema de visão, onde através de câmeras e sensores o manipulador consegue ter esta habilidade (KHATIB, 2008). Este é um ponto muito interessante de melhoria a ser estudado.

## REFERÊNCIAS

CRAIG, JOHN J. **Introduction to robotics: mechanics and control**. Upper Saddle River: Pearson Prentice Hall, 2005.

KHATIB O. **Lecture Collection | Introduction to Robotics**. Stanford Computer Science Department, Stanford University, 2008. 90 minutos. Disponível em <https://www.youtube.com/playlist?list=PL65CC0384A1798ADF>. Acesso em: outubro de 2016

NABEEL, H. M. et al. **Designing, fabrication and controlling of multipurpose 3-dof robotic arm**. Materials Science and Engineering, 2013. 12-23.

PAUL, R.P.; SIMANO, B.; MAYER G. **Kinematic Control Equations for Simple Manipulators**. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 6, 1981.

SIANG, S.Y. **RC Servo Motor: Theory of Operation**. Underwater Robotics Research Group. Disponível em: <http://urrg.eng.usm.my/index.php/ms/>. Acesso em: outubro de 2016.

**Site oficial do OWI Robotic Arm**. Disponível em: <http://www.owirobot.com/robotic-arm-edge-1/>. Acesso em: novembro de 2016.

**Site oficial do DAGU Robotic Arm**. Disponível em: <http://www.dagurobot.com/goods.php?id=68>. Acesso em: outubro 2017.

**Site oficial da TowerPro**. Disponível em: <http://www.towerpro.com.tw/product/mg946r/>. Acesso em: outubro 2017.

## APÊNDICE A – FUNÇÃO MATLAB CINEMÁTICA DIRETA 1

```

function [Final] = FK( theta )
%Parâmetros D-H
a=[0 0 0.14 0 0 0];
alpha=[0 -90 0 -90 90 -90];
d= [0 0 0 0.14 0 0];
%Cinemática direta
for i=1:6
T=[cosd(theta(i)) -sind(theta(i)) 0 a(i);
    sind(theta(i))*cosd(alpha(i)) cosd(theta(i))*cosd(alpha(i)) -
sind(alpha(i)) -sind(alpha(i))*d(i);
    sind(theta(i))*sind(alpha(i)) cosd(theta(i))*sind(alpha(i))
cosd(alpha(i)) cosd(alpha(i))*d(i);
    0 0 0 1];
if i==1 T10=T; end
if i==2 T21=T; end
if i==3 T32=T; end
if i==4 T43=T; end
if i==5 T54=T; end
if i==6 T65=T; end
end
T63=T43*T54*T65;
T76=[1 0 0 0; 0 1 0 0; 0 0 1 0.15667; 0 0 0 1]; %Transformação para garra
T60=T10*T21*T32*T43*T54*T65; %Solução para a última junta
T70=T60*T76; %Solução para a garra
pitch=atan2d(-T63(3,1),sqrt(T63(1,1)^2+T63(2,1)^2)); % Yaw, pitch e roll
são os ângulos alpha, beta e gama de Euler
if pitch==90%Checando para o caso de cos(beta)=0, quando a solução dentro
do "else" torna-se degenerada
pitch=90;
yaw=0;
roll=atan2d(T63(1,2),T63(2,2));
elseif pitch== -90
pitch=-90;
yaw=0;
roll=-atan2d(T63(1,2),T63(2,2));
else
roll=atan2d((T63(3,2)/cosd(pitch)),(T63(3,3)/cosd(pitch)));
yaw=atan2d((T63(2,1)/cosd(pitch)),(T63(1,1)/cosd(pitch)));
end
Final=[T70(1,4) T70(2,4) T70(3,4); yaw pitch roll]; %Solução para a garra,
com ângulos Z-Y-X de Euler
end

```

## APÊNDICE B – FUNÇÃO MATLAB CINEMÁTICA DIRETA 2

```

function [T70] = plot_robot( theta )
%Parâmetros D-H
a=[0 0 0.14 0 0 0];
alpha=[0 -90 0 -90 90 -90];
d= [0 0 0 0.14 0 0];
%Cinemática Direta
for i=1:6
T=[cosd(theta(i)) -sind(theta(i)) 0 a(i);
   sind(theta(i))*cosd(alpha(i)) cosd(theta(i))*cosd(alpha(i)) -
sind(alpha(i)) -sind(alpha(i))*d(i);
   sind(theta(i))*sind(alpha(i)) cosd(theta(i))*sind(alpha(i))
cosd(alpha(i)) cosd(alpha(i))*d(i);
   0 0 0 1];
if i==1 T10=T; end
if i==2 T21=T; end
if i==3 T32=T; end
if i==4 T43=T; end
if i==5 T54=T; end
if i==6 T65=T; end
end
T76=[1 0 0 0; 0 1 0 0; 0 0 1 0.15667; 0 0 0 1];
T60=T10*T21*T32*T43*T54*T65;
T70=T60*T76;
T50=T10*T21*T32*T43*T54;
T30=T10*T21*T32;
x=[T10(1,4) T30(1,4) T50(1,4) T70(1,4)];
y=[T10(2,4) T30(2,4) T50(2,4) T70(2,4)];
z=[T10(3,4) T30(3,4) T50(3,4) T70(3,4)];
plot3(x,y,z)
end

```

## APÊNDICE C – FUNÇÃO MATLAB CINEMÁTICA INVERSA

```

function [ final ] = IK( coord )
%Parâmetros D-H
a=[0 0 0.14 0 0 0];
alpha=[0 -90 0 -90 90 -90];
d= [0 0 0 0.14 0 0];
%Transformação de dados de entrada
px=coord(1,1);
py=coord(1,2);
pz=coord(1,3);
g=coord(2,3); %Ângulo gama de Euler
b=coord(2,2); %Ângulo beta de Euler
al=coord(2,1); %Ângulo alpha de Euler
r11=cosd(al)*cosd(b);
r12=cosd(al)*sind(b)*sind(g)-sind(al)*cosd(g);
r13=cosd(al)*sind(b)*cosd(g)+sind(al)*sind(g);
r21=sind(al)*cosd(b);
r22=sind(al)*sind(b)*sind(g)+cosd(al)*cosd(g);
r23=sind(al)*sind(b)*cosd(g)-cosd(al)*sind(g);
r31=-sind(b);
r32=cosd(b)*sind(g);
r33=cosd(b)*cosd(g);
%Theta1 - Duas soluções possíveis
thetala=atan2d(py,px);
thetalb=atan2d(-py,-px);
c1a=cosd(thetala);
c1b=cosd(thetalb);
s1a=sind(thetala);
s1b=sind(thetalb);
%Theta3 - Duas soluções possíveis
k=(px^2+py^2+pz^2-a(3)^2-d(4)^2)/(2*a(3)*d(4));
theta3a=atan2d(-k,+sqrt(1-k^2));
theta3b=atan2d(-k,-sqrt(1-k^2));
s3a=sind(theta3a);
s3b=sind(theta3b);
c3a=cosd(theta3a);
c3b=cosd(theta3b);
%Theta2 - Quatro soluções possíveis (aa, ab, ba, bb) com base em theta1 e
theta3
theta23aa=atan2d(-a(3)*c3a*pz+(a(3)*s3a-d(4))*(c1a*px+s1a*py),(a(3)*s3a-
d(4))*pz+(a(3)*c3a)*(c1a*px+s1a*py));
theta23ab=atan2d(-a(3)*c3b*pz+(a(3)*s3b-d(4))*(c1a*px+s1a*py),(a(3)*s3b-
d(4))*pz+(a(3)*c3b)*(c1a*px+s1a*py));
theta23ba=atan2d(-a(3)*c3a*pz+(a(3)*s3a-d(4))*(c1b*px+s1b*py),(a(3)*s3a-
d(4))*pz+(a(3)*c3a)*(c1b*px+s1b*py));
theta23bb=atan2d(-a(3)*c3b*pz+(a(3)*s3b-d(4))*(c1b*px+s1b*py),(a(3)*s3b-
d(4))*pz+(a(3)*c3b)*(c1b*px+s1b*py));
theta2aa=theta23aa-theta3a;
theta2ab=theta23ab-theta3b;
theta2ba=theta23ba-theta3a;
theta2bb=theta23bb-theta3b;
s2aa=sind(theta2aa);
s2ab=sind(theta2ab);
s2ba=sind(theta2ba);
s2bb=sind(theta2bb);
c2aa=cosd(theta2aa);
c2ab=cosd(theta2ab);

```

```

c2ba=cosd(theta2ba);
c2bb=cosd(theta2bb);
s23aa=sind(theta2aa+theta3a);
s23ab=sind(theta2ab+theta3b);
s23ba=sind(theta2ba+theta3a);
s23bb=sind(theta2bb+theta3b);
c23aa=cosd(theta2aa+theta3a);
c23ab=cosd(theta2ab+theta3b);
c23ba=cosd(theta2ba+theta3a);
c23bb=cosd(theta2bb+theta3b);
%Transformação dos ângulos de Euler
R30aa=[ cosd(theta2aa + theta3a)*cosd(thetalb), -sind(theta2aa +
theta3a)*cosd(thetalb), -sind(thetalb);
        cosd(theta2aa + theta3a)*sind(thetalb), -sind(theta2aa +
theta3a)*sind(thetalb),  cosd(thetalb);
        -sind(theta2aa + theta3a),          -cosd(theta2aa + theta3a),
0];
R30ab=[ cosd(theta2ab + theta3a)*cosd(thetalb), -sind(theta2ab +
theta3a)*cosd(thetalb), -sind(thetalb);
        cosd(theta2ab + theta3a)*sind(thetalb), -sind(theta2ab +
theta3a)*sind(thetalb),  cosd(thetalb);
        -sind(theta2ab + theta3a),          -cosd(theta2ab + theta3a),
0];
R30ba=[ cosd(theta2ba + theta3b)*cosd(thetalb), -sind(theta2ba +
theta3b)*cosd(thetalb), -sind(thetalb);
        cosd(theta2ba + theta3b)*sind(thetalb), -sind(theta2ba +
theta3b)*sind(thetalb),  cosd(thetalb);
        -sind(theta2ba + theta3b),          -cosd(theta2ba + theta3b),
0];
R30bb=[ cosd(theta2bb + theta3b)*cosd(thetalb), -sind(theta2bb +
theta3b)*cosd(thetalb), -sind(thetalb);
        cosd(theta2bb + theta3b)*sind(thetalb), -sind(theta2bb +
theta3b)*sind(thetalb),  cosd(thetalb);
        -sind(theta2bb + theta3b),          -cosd(theta2bb + theta3b),
0];
R63=[r11 r12 r13;r21 r22 r23;r31 r32 r33];
R60aa=R30aa*R63;R60ab=R30ab*R63;R60ba=R30ba*R63;R60bb=R30bb*R63;
r11aa=R60aa(1,1);r11ab=R60ab(1,1);r11ba=R60ba(1,1);r11bb=R60bb(1,1);
r12aa=R60aa(1,2);r12ab=R60ab(1,2);r12ba=R60ba(1,2);r12bb=R60bb(1,2);
r13aa=R60aa(1,3);r13ab=R60ab(1,3);r13ba=R60ba(1,3);r13bb=R60bb(1,3);
r21aa=R60aa(2,1);r21ab=R60ab(2,1);r21ba=R60ba(2,1);r21bb=R60bb(2,1);
r22aa=R60aa(2,2);r22ab=R60ab(2,2);r22ba=R60ba(2,2);r22bb=R60bb(2,2);
r23aa=R60aa(2,3);r23ab=R60ab(2,3);r23ba=R60ba(2,3);r23bb=R60bb(2,3);
r31aa=R60aa(3,1);r31ab=R60ab(3,1);r31ba=R60ba(3,1);r31bb=R60bb(3,1);
r32aa=R60aa(3,2);r32ab=R60ab(3,2);r32ba=R60ba(3,2);r32bb=R60bb(3,2);
r33aa=R60aa(3,3);r33ab=R60ab(3,3);r33ba=R60ba(3,3);r33bb=R60bb(3,3);
%Theta4 - Quatro soluções possíveis (aa, ab, ba, bb) com base em theta 1 e
theta3
theta4aa=atan2d(-s1a*r13aa+c1a*r23aa,-c1a*c23aa*r13aa-
s1a*c23aa*r23aa+s23aa*r33aa);
theta4ab=atan2d(-s1a*r13ab+c1a*r23ab,-c1a*c23ab*r13ab-
s1a*c23ab*r23ab+s23ab*r33ab);
theta4ba=atan2d(-s1b*r13ba+c1b*r23ba,-c1b*c23ba*r13ba-
s1b*c23ba*r23ba+s23ba*r33ba);
theta4bb=atan2d(-s1b*r13bb+c1b*r23bb,-c1b*c23bb*r13bb-
s1b*c23bb*r23bb+s23bb*r33bb);
s4aa=sind(theta4aa);
s4ab=sind(theta4ab);
s4ba=sind(theta4ba);
s4bb=sind(theta4bb);

```

```

c4aa=cosd(theta4aa);
c4ab=cosd(theta4ab);
c4ba=cosd(theta4ba);
c4bb=cosd(theta4bb);
%Theta5 - Quatro soluções, singularidade quando theta5=0
v5aa=r33aa*s23aa*c4aa-r23aa*(s1a*c23aa*c4aa-c1a*s4aa)-
r13aa*(c1a*c23aa*c4aa+s1a*s4aa);
v5ab=r33ab*s23ab*c4ab-r23ab*(s1a*c23ab*c4ab-c1a*s4ab)-
r13ab*(c1a*c23ab*c4ab+s1a*s4ab);
v5ba=r33ba*s23ba*c4ba-r23ba*(s1b*c23ba*c4ba-c1b*s4ba)-
r13ba*(c1b*c23ba*c4ba+s1b*s4ba);
v5bb=r33bb*s23bb*c4bb-r23bb*(s1b*c23bb*c4bb-c1b*s4bb)-
r13bb*(c1b*c23bb*c4bb+s1b*s4bb);
w5aa=-r13aa*c1a*s23aa-r23aa*s1a*s23aa-r33aa*c23aa;
w5ab=-r13ab*c1a*s23ab-r23ab*s1a*s23ab-r33ab*c23ab;
w5ba=-r13ba*c1b*s23ba-r23ba*s1b*s23ba-r33ba*c23ba;
w5bb=-r13bb*c1b*s23bb-r23bb*s1b*s23bb-r33bb*c23bb;
theta5aa=atan2d(v5aa,w5aa);
theta5ab=atan2d(v5ab,w5ab);
theta5ba=atan2d(v5ba,w5ba);
theta5bb=atan2d(v5bb,w5bb);
s5aa=sind(theta5aa);
s5ab=sind(theta5ab);
s5ba=sind(theta5ba);
s5bb=sind(theta5bb);
c5aa=cosd(theta5aa);
c5ab=cosd(theta5ab);
c5ba=cosd(theta5ba);
c5bb=cosd(theta5bb);
%Theta 6
v6aa=-r11aa*(c1a*c23aa*s4aa-s1a*c4aa)-
r21aa*(s1a*c23aa*s4aa+c1a*c4aa)+r31aa*(s23aa*s4aa);
v6ab=-r11ab*(c1a*c23ab*s4ab-s1a*c4ab)-
r21ab*(s1a*c23ab*s4ab+c1a*c4ab)+r31ab*(s23ab*s4ab);
v6ba=-r11ba*(c1b*c23ba*s4ba-s1b*c4ba)-
r21ba*(s1b*c23ba*s4ba+c1b*c4ba)+r31ba*(s23ba*s4ba);
v6bb=-r11bb*(c1b*c23bb*s4bb-s1b*c4bb)-
r21bb*(s1b*c23bb*s4bb+c1b*c4bb)+r31bb*(s23bb*s4bb);
w6aa=r11aa*((c1a*c23aa*c4aa+s1a*s4aa)*c5aa-
c1a*s23aa*s5aa)+r21aa*((s1a*c23aa*c4aa-c1a*s4aa)*c5aa-s1a*s23aa*s5aa)-
r31aa*(s23aa*c4aa*c5aa+c23aa*s5aa);
w6ab=r11ab*((c1a*c23ab*c4ab+s1a*s4ab)*c5ab-
c1a*s23ab*s5ab)+r21ab*((s1a*c23ab*c4ab-c1a*s4ab)*c5ab-s1a*s23ab*s5ab)-
r31ab*(s23ab*c4ab*c5ab+c23ab*s5ab);
w6ba=r11ba*((c1b*c23ba*c4ba+s1b*s4ba)*c5ba-
c1b*s23ba*s5ba)+r21ba*((s1b*c23ba*c4ba-c1b*s4ba)*c5ba-s1b*s23ba*s5ba)-
r31ba*(s23ba*c4ba*c5ba+c23ba*s5ba);
w6bb=r11bb*((c1b*c23bb*c4bb+s1b*s4bb)*c5bb-
c1b*s23bb*s5bb)+r21bb*((s1b*c23bb*c4bb-c1b*s4bb)*c5bb-s1b*s23bb*s5bb)-
r31bb*(s23bb*c4bb*c5bb+c23bb*s5bb);
theta6aa=atan2d(v6aa,w6aa);
theta6ab=atan2d(v6ab,w6ab);
theta6ba=atan2d(v6ba,w6ba);
theta6bb=atan2d(v6bb,w6bb);
%Soluções
aa=[theta1a theta2aa theta3a theta4aa theta5aa theta6aa]';
ab=[theta1a theta2ab theta3b theta4ab theta5ab theta6ab]';
ba=[theta1b theta2ba theta3a theta4ba theta5ba theta6ba]';
bb=[theta1b theta2bb theta3b theta4bb theta5bb theta6bb]';
aaf=[theta1a theta2aa theta3a theta4aa+180 -theta5aa theta6aa+180]';

```

```
abf=[theta1a theta2ab theta3b theta4ab+180 -theta5ab theta6ab+180]';  
baf=[theta1b theta2ba theta3a theta4ba+180 -theta5ba theta6ba+180]';  
bbf=[theta1b theta2bb theta3b theta4bb+180 -theta5bb theta6bb+180]';  
final=[aa ab ba bb aaf abf baf bbf];  
end
```

## APÊNDICE D – FUNÇÃO MATLAB JACOBIANO

```

function [J] = jacobian( theta )
%Parâmetros D-H
a=[0 0 200 0 0 0];
d= [0 0 0 150 0 0];
a3=a(3);
d4=d(4);
%Simplificações de nomenclatura
s1=sind(theta(1)); s2=sind(theta(2)); s3=sind(theta(3)); s4=sind(theta(4));
s5=sind(theta(5)); s6=sind(theta(6));
s23=sind(theta(2)+theta(3)); c23=cosd(theta(2)+theta(3));
c1=cosd(theta(1)); c2=cosd(theta(2)); c3=cosd(theta(3)); c4=cosd(theta(4));
c5=cosd(theta(5)); c6=cosd(theta(6));
%Jacobiano
J=[-s1*(a3*c2-s23*d4) -c1*(a3*s2+d4*c23) -c1*d4*c23 0 0 0;
   c1*(a3*c2-s23*d4) -s1*(a3*s2+d4*c23) -s1*d4*c23 0 0 0;
   0 -a3*c2+d4*s23 d4*s23 0 0 0;
   0 -s1 -s1 -c1*s23 c1*c23*s4-s1*c4 -s5*(s1*s4+c1*c4*c23)-c1*c5*s23;
   0 c1 c1 -s1*s23 s1*c23*s4+c1*c4 s5*(c1*s4-s1*c4*c23)-s1*c5*s23;
   1 0 0 -c23 -s4*s23 c4*s5*s23-c5*c23];
end

```

## APÊNDICE E – FUNÇÃO MATLAB DINÂMICA

```

function [T] = dynamics( theta, thetap, thetapp, massa )
%Parâmetros D-H
a=[0 0 0.14 0 0 0];
alpha=[0 -90 0 -90 90 -90];
d=[0 0 0 0.14 0 0];
%Cinemática Direta
for i=1:6
T=[cosd(theta(i)) -sind(theta(i)) 0 a(i);sind(theta(i))*cosd(alpha(i))
cosd(theta(i))*cosd(alpha(i)) -sind(alpha(i)) -
sind(alpha(i))*d(i);sind(theta(i))*sind(alpha(i))
cosd(theta(i))*sind(alpha(i)) cosd(alpha(i)) cosd(alpha(i))*d(i);0 0 0 1];
if i==1 T10=T; end
if i==2 T21=T; end
if i==3 T32=T; end
if i==4 T43=T; end
if i==5 T54=T; end
if i==6 T65=T; end
end
T76=[1 0 0 0; 0 1 0 0; 0 0 1 0.15667; 0 0 0 1]; %Transformação para garra
%Parâmetros Dinâmicos
pc1=[0;0;-0.02603];
pc2=[0.09659;0;-0.01693];
pc3=[-0.00108;0.06753;-0.00068];
pc4=[0;0;-0.00738];
pc5=[-0.00261;0.00865;0];
pc6=[-0.00399;0.001;0.07674];
ic1=10^-9*[742895.03 0 0; 0 242804.64 2.58; 0 2.58 707185.62];
ic2=10^-9*[163347.71 -0.31 -77560.91; -0.31 463549.5 0.29; -77560.91 0.29
318631.48];
ic3=10^-9*[79553.17 -1498.82 141.45; -1498.82 20165.80 -1010.31; 141.45 -
1010.31 69956.53];
ic4=10^-9*[36165.48 0 0;0 5314.19 0;0 0 33973.34];
ic5=10^-9*[3643.16 -41.04 1.48;-41.04 2792.90 0.01;1.48 0.01 4455.03];
ic6=10^-9*[41071.01 -246.59 3932.00; -246.59 40370.90 -808.27; 3932.00 -
808.27 7678.28];
m=[0.25801;0.13446;0.05598;0.03620;0.02204;0.04283;massa/1000]; %massa/1000
é algum objeto inserido na ponta da garra
g=9.81; %Aceleração da gravidade "para cima"
%Equações Iterativas
w1=thetap(1)*[0;0;1]; %Velocidades rotacionais
w2=(T21(1:3,1:3)^-1)*w1+thetap(2)*[0;0;1];
w3=(T32(1:3,1:3)^-1)*w2+thetap(3)*[0;0;1];
w4=(T43(1:3,1:3)^-1)*w3+thetap(4)*[0;0;1];
w5=(T54(1:3,1:3)^-1)*w4+thetap(5)*[0;0;1];
w6=(T65(1:3,1:3)^-1)*w5+thetap(6)*[0;0;1];
w7=w6;
wp1=thetapp(1)*[0;0;1]; %Acelerações rotacionais
wp2=(T21(1:3,1:3)^-1)*wp1+cross((T21(1:3,1:3)^-
1)*w1,thetap(2)*[0;0;1])+thetapp(2)*[0;0;1];
wp3=(T32(1:3,1:3)^-1)*wp2+cross((T32(1:3,1:3)^-
1)*w2,thetap(3)*[0;0;1])+thetapp(3)*[0;0;1];
wp4=(T43(1:3,1:3)^-1)*wp3+cross((T43(1:3,1:3)^-
1)*w3,thetap(4)*[0;0;1])+thetapp(4)*[0;0;1];
wp5=(T54(1:3,1:3)^-1)*wp4+cross((T54(1:3,1:3)^-
1)*w4,thetap(5)*[0;0;1])+thetapp(5)*[0;0;1];

```

```

wp6=(T65(1:3,1:3)^-1)*wp5+cross((T65(1:3,1:3)^-
1)*w5,thetap(6)*[0;0;1])+thetapp(6)*[0;0;1];
wp7=wp6;
vp1=(T10(1:3,1:3)^-1)*[0;0;g]; %Acelerações lineares
vp2=(T21(1:3,1:3)^-
1)*(cross(wp1,T21(1:3,4))+cross(w1,cross(w1,T21(1:3,4))))+vp1);
vp3=(T32(1:3,1:3)^-
1)*(cross(wp2,T32(1:3,4))+cross(w2,cross(w2,T32(1:3,4))))+vp2);
vp4=(T43(1:3,1:3)^-
1)*(cross(wp3,T43(1:3,4))+cross(w3,cross(w3,T43(1:3,4))))+vp3);
vp5=(T54(1:3,1:3)^-
1)*(cross(wp4,T54(1:3,4))+cross(w4,cross(w4,T54(1:3,4))))+vp4);
vp6=(T65(1:3,1:3)^-
1)*(cross(wp5,T65(1:3,4))+cross(w5,cross(w5,T65(1:3,4))))+vp5);
vp7=(cross(wp6,T76(1:3,4))+cross(w6,cross(w6,T76(1:3,4))))+vp6);
vpc1=cross(wp1,pc1)+cross(w1,cross(w1,pc1))+vp1; %Acelerações nos centros
de massa
vpc2=cross(wp2,pc2)+cross(w2,cross(w2,pc2))+vp2;
vpc3=cross(wp3,pc3)+cross(w3,cross(w3,pc3))+vp3;
vpc4=cross(wp4,pc4)+cross(w4,cross(w4,pc4))+vp4;
vpc5=cross(wp5,pc5)+cross(w5,cross(w5,pc5))+vp5;
vpc6=cross(wp6,pc6)+cross(w6,cross(w6,pc6))+vp6;
vpc7=vp7;
fc1=m(1)*vpc1; %Forças nos centros de massa
fc2=m(2)*vpc2;
fc3=m(3)*vpc3;
fc4=m(4)*vpc4;
fc5=m(5)*vpc5;
fc6=m(6)*vpc6;
fc7=m(7)*vpc7;
nc1=ic1*wp1+cross(w1,ic1*w1); %Momentos nos centros de massa
nc2=ic2*wp2+cross(w2,ic2*w2);
nc3=ic3*wp3+cross(w3,ic3*w3);
nc4=ic4*wp4+cross(w4,ic4*w4);
nc5=ic5*wp5+cross(w5,ic5*w5);
nc6=ic6*wp6+cross(w6,ic6*w6);
nc7=[0;0;0];
f6=fc7+fc6; %Forças totais
f5=T65(1:3,1:3)*f6+fc5;
f4=T54(1:3,1:3)*f5+fc4;
f3=T43(1:3,1:3)*f4+fc3;
f2=T32(1:3,1:3)*f3+fc2;
f1=T21(1:3,1:3)*f2+fc1;
n6=nc6+cross(pc6,fc6)+cross(T76(1:3,4),fc7); %Momentos totais
n5=nc5+T65(1:3,1:3)*n6+cross(pc5,fc5)+cross(T65(1:3,4),T65(1:3,1:3)*f6);
n4=nc4+T54(1:3,1:3)*n5+cross(pc4,fc4)+cross(T54(1:3,4),T54(1:3,1:3)*f5);
n3=nc3+T43(1:3,1:3)*n4+cross(pc3,fc3)+cross(T43(1:3,4),T43(1:3,1:3)*f4);
n2=nc2+T32(1:3,1:3)*n3+cross(pc2,fc2)+cross(T32(1:3,4),T32(1:3,1:3)*f3);
n1=nc1+T21(1:3,1:3)*n2+cross(pc1,fc1)+cross(T21(1:3,4),T21(1:3,1:3)*f2);
t6=n6'*[0;0;1]; %Torques nas juntas
t5=n5'*[0;0;1];
t4=n4'*[0;0;1];
t3=n3'*[0;0;1];
t2=n2'*[0;0;1];
t1=n1'*[0;0;1];
T=10.197162129779*[t1;t2;t3;t4;t5;t6]; %Vetor de torques
end

```

## APÊNDICE F – ROTINA DE CONTROLE EM ARDUINO

```

String readString;

#include <VarSpeedServo.h>
VarSpeedServo myservoa, myservob, myservobb, myservoc, myservod, myservoe,
myservoee, myservof, myservog;
void setup() {

    Serial.begin(9600);
    Serial.println("Forward Kinematics");

}

void loop() {

    if (Serial.available()>0) {
        char c = Serial.read();
        if (c == ',') {
            if (readString.length() >1) {
                Serial.println(readString);
                int n = readString.toInt();
                if(readString.indexOf('a') >0) {
                    myservoa.attach(2);
                    Serial.print("###THETA 1:###\n");
                    Serial.println(n);
                    myservoa.write(n, 30, true);
                }
                if(readString.indexOf('b') >0) {
                    myservob.attach(3);
                    myservobb.attach(5);
                    Serial.print("###THETA 2:###\n");
                    Serial.println(n);
                    myservob.write(n, 30, true);
                    myservobb.write(180-n, 30, true);
                }
                if(readString.indexOf('c') >0) {
                    myservoc.attach(6);
                    Serial.print("###THETA 3:###\n");
                    Serial.println(n);
                    myservoc.write(n, 30, true);
                }
                if(readString.indexOf('d') >0) {
                    myservod.attach(7);
                    Serial.print("###THETA 4:###\n");
                    Serial.println(n);
                    myservod.write(n, 30, true);
                    delay(1000);
                    myservod.detach();
                }
                if(readString.indexOf('e') >0) {
                    myservoe.attach(8);
                    myservoee.attach(9);
                    Serial.print("###THETA 5:###\n");
                    Serial.println(n);
                    myservoe.write(n, 30, true);
                    myservoee.write(180-n, 30, true);
                    delay(1000);
                }
            }
        }
    }
}

```

```
        myservoee.detach();
        myservoee.detach();
    }
    if(readString.indexOf('f') >0) {
        myservof.attach(11);
        Serial.print("###THETA 6:###\n");
        Serial.println(n);
        myservof.write(n, 30, true);
        delay(1000);
        myservof.detach();
    }
    if(readString.indexOf('g') >0) {
        myservog.attach(12);
        Serial.print("###GRIPPER:###\n");
        Serial.println(n);
        myservog.write(n, 60, true);
    }
    }
    readString="";
}
else {
    readString += c;
}
}
}
```