



**PROJETO DE GRADUAÇÃO**

**MODELAGEM CINEMÁTICA E CONTROLE DE  
TRAJETÓRIA DE UM ROBÔ PARA  
APLICAÇÕES ASSISTIVAS**

Por,  
**Paulo Maia da Costa Filho**

**Brasília, 20 de Novembro de 2019**

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Departamento de Engenharia Mecânica

PROJETO DE GRADUAÇÃO

**MODELAGEM CINEMÁTICA E CONTROLE DE  
TRAJETÓRIA DE UM ROBÔ PARA  
APLICAÇÕES ASSISTIVAS**

POR,

**Paulo Maia da Costa Filho**

Relatório submetido como requisito parcial para obtenção do grau de Engenheiro Mecânico.

**Banca Examinadora**

Prof. Walter de Britto Vidal Filho, UnB/ENM  
(Orientador)

\_\_\_\_\_

Prof. Guilherme Caribé de Carvalho, UnB/ENM

\_\_\_\_\_

Prof. Adriano Todorovic Fabro, UnB/ENM

\_\_\_\_\_

Brasília, 20 de Novembro 2019

## **Agradecimentos**

Agradeço acima de tudo aos meus pais, Paulo Maia e Rívea Fernandes, que nunca mediram esforços para oferecer, a mim e às minhas irmãs, as melhores oportunidades, o melhor estudo, a melhor educação e, antes de mais nada, o maior carinho, dedicação e amor. Obrigado por me darem raízes para saber tomar as decisões, e asas para ir atrás do que eu quiser. Agradeço também às minhas irmãs, Clarinha e Bebel, pela parceria e cumplicidade.

Agradeço à UnB, pela oportunidade de estudar em uma das melhores Universidades do Brasil, onde pude não só estudar engenharia, mas ter vivências fantásticas e conhecer pessoas incríveis.

Agradeço à TECMEC e à Concentro, que por 3 anos e meio me mostraram que a melhoria e aprendizado são constantes, que estamos sim mudando o país e que de nada vale se não estivermos de corpo e alma naquilo que estamos fazendo.

Agradeço aos amigos do ensino médio que ainda mantenho contato; aos amigos do curso, que ajudam a seguir em frente na batalha diária; aos amigos do Movimento Empresa Júnior, que sempre sobem a régua e dizem que dá para fazer mais; aos amigos da correria, que de alguma forma cruzaram com meu caminho e, com certeza, me transformaram de alguma forma; e aos amigos para a vida, que espero que todos se tornem.

Agradeço em especial também à Bianca, por ter começado a caminhar junto comigo nessa estrada da “vida adulta”, sendo cúmplice, admiradora e confiança.

Professor Britto, muito obrigado por todos os questionamentos, preocupações, orientações, conselhos e puxões de orelha, que tenho certeza que me forjam enquanto engenheiro e profissional. Afinal, como você sempre diz: “Melhor te apertar aqui para não se enrolar na hora do vamos ver”.

Por fim, agradeço a todos que compartilham comigo essa conquista!

*Paulo Maia da Costa Filho*

---

## RESUMO

O presente projeto é continuação de um Projeto de Graduação, no qual se desenvolveu a parte mecânica de um manipulador para utilização em cadeira de rodas (WRMA - *Wheelchair mounted robotic arm*). Este manipulador vai auxiliar pessoas tetraplégicas adquirir liberdade de interação com o meio. No presente projeto, a cinemática inversa de posição e planejamento e geração de trajetória são os objetivos. A primeira foi desenvolvida geometricamente pela imposição de algumas restrições de movimentação, e a segunda por meio da utilização do jacobiano inverso de velocidades e da simulação de leitura de um joystick. A validade da solução foi verificada por meio de uma simulação para um robô planar.

Palavras-chave: WMRA, Cadeira de rodas, Tecnologia Assistiva, Lesão medular, Manipulador robótico, Controle de Trajetória, modelagem cinemática.

---

## ABSTRACT

The present study continues the job of a Graduation Project, in which the mechanical part of a wheelchair mounted robotic arm, WRMA, was developed. This manipulator is going to help quadriplegic people acquire independence interaction with the environment. In this project the inverse kinematic of position and the planning and generation of the trajectory are the objectives. The first one was geometrically solved imposing movement restrictions, and the second by inverting the jacobian matrix and simulating reading a joystick. The validity of the solution was verified by a planar robot simulation.

Keywords: WMRA, Wheelchair, Assistive Technology, Spinal Cord Injury, Robotic manipulator, trajectory control kinematic modeling.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS	2
1.3	METODOLOGIA E ORGANIZAÇÃO DO TRABALHO	3
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>6</b>
2.1	TRABALHOS ANTERIORES	6
2.1.1	GEOMETRIA DO MANIPULADOR	6
2.1.2	MODELAGEM CINEMÁTICA	8
2.1.3	ANÁLISE DINÂMICA	9
2.1.4	MOTORES E SISTEMA DE REDUÇÃO	10
2.2	LESÃO MEDULAR, TETRAPLEGIA E REABILITAÇÃO	11
2.3	ESTADO DA ARTE	12
2.4	SISTEMAS DE CONTROLE E PROGRAMAÇÃO ROBÓTICA	21
2.4.1	SISTEMA DE CONTROLE	22
2.4.2	PROGRAMAÇÃO ROBÓTICA	24
2.3	VOLUME DE CONTROLE	26
2.4	CINEMÁTICA DE MANIPULADORES	28
2.4.1	MÉTODO ALGÉBRICO	29
2.4.2	MÉTODO GEOMÉTRICO	29
2.4.3	MÉTODOS NUMÉRICOS	33
2.4.4	MÉTODOS MISTOS	33
2.5	PLANEJAMENTO DA TRAJETÓRIA	34
2.5.1	VELOCIDADES	36
2.6	DISPOSITIVOS DE COMANDO	41
<b>3</b>	<b>SOLUÇÃO CONCEITUAL</b>	<b>45</b>
3.1	PREMISSAS	45
3.1	SÍNTESE E CONCEPÇÃO	45
3.3	ANÁLISE	46
3.6	RESUMO DO CONCEITO DO PROJETO	50
<b>4</b>	<b>DESENVOLVIMENTO DA SOLUÇÃO</b>	<b>52</b>
4.1	VOLUME DE TRABALHO	52
4.2	ROBOTICS TOOLBOX FOR MATLAB (RTB)	55
4.3	CINEMÁTICA INVERSA	56
4.3.1	PROBLEMÁTICA	56
4.3.2	SOLUÇÃO	60
4.4	VOLUME DE TRABALHO ATUALIZADO	64
4.5	PLANEJAMENTO DA TRAJETÓRIA	67
<b>5</b>	<b>TESTES E VALIDAÇÃO</b>	<b>71</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>85</b>
6.1	TRABALHOS FUTUROS	85
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>86</b>
	<b>APÊNDICES</b>	<b>91</b>

## LISTA DE FIGURAS

Figura 1 – Sistemática do projeto de graduação .....	5
Figura 2 – Conceito da solução (Néto, 2019).....	6
Figura 3 – Modelo em SolidWorks da solução proposta, em construção (Néto, 2019).....	7
Figura 4 – Dimensões dos elos (Néto, 2019) .....	7
Figura 5 – Desenho esquemático do manipulador com os parâmetros D-H (Néto, 2019) .....	8
Figura 6 – Indicações dos motores de cada junta. Adaptado (Néto, 2019).....	10
Figura 7 – Perspectivas funcionais de acordo com nível de lesão medular. Adaptado (Ministério da Saúde, 2013).....	11
Figura 8 – Usuário do KARES (Song, Lee, Kim, Yoon, & Bien, 1998). .....	13
Figura 9 – Usuário do Manus Arm, com o controle original por joystick (Exact Dynamics, 2018).....	14
Figura 10 – Distribuição espacial das 12 posições pré-programadas do MANUS (Didi, Grandjean, Mokhtari, & Roby-Brami, 1998).....	14
Figura 11 – Partição sequencial da vista em 4 quadrantes (Tsui & Yanco).....	15
Figura 12 – Página inicial da interface gráfica com o usuário (Yanco, Hazel, Peacock, Smith, & Wintermute, 1995) .....	16
Figura 13 – Simulação virtual do movimento do manipulador (Wagner, Loos, Smaby, Chang, & Bugar, 1999) .....	17
Figura 14 – Modo de controle direto dos 3 eixos do efetuador do JACO (Rehab).....	18
Figura 15 – Modo de controle das juntas de orientação do JACO (Rehab).....	18
Figura 16 – Modo de controle das juntas de orientação com a opção “Drinking mode” (Rehab) .....	19
Figura 17 – Modo de controle dos dedos da garra do JACO (Rehab) .....	19
Figura 18 – Diagrama de blocos do funcionamento do dispositivo de apontamento com infravermelho (Hatakeyama, Kkoizumi, Umeda, Uchio, & Kasuga, 1998) .....	20
Figura 19 – Bandeja de alimentação e bebida do Handy 1 (Topping, 2002).....	20
Figura 20 – Tela da interface com o usuário (Hillman, Hagan, Hagan, Jepson, & Orpwood, 1999).....	21
Figura 21 – Representação simplificada de um sistema .....	22
Figura 22 – Representação dos subsistemas de um sistema.....	22
Figura 23 – Representação de um sistema de controle .....	22
Figura 24 – Representação dos elementos básicos de um sistema em malha aberta (Bolton, 1995).....	23
Figura 25 – Representação dos elementos básicos de um sistema em malha fechada (Bolton, 1995).....	24
Figura 26 – Teaching-box robô ABB TRB 6400 .....	25
Figura 27 - Singularidades de contorno (esquerda) e interna (direita) (Erthal, 1992) .....	26
Figura 28 – Volume de trabalho. Adaptado (Hongyu & Saramago, 2008) .....	27
Figura 29 – Poliedro definido pelos 8 pontos alcançáveis, em torno de um ponto desejado, de um manipulador com 3 graus de liberdade (Costa, 2003).....	27
Figura 30–Transformações entre variáveis de junta e cartesianas (Carrara, 2015) .....	28
Figura 31– Manipular RRR se movimentando em um plano vertical $xy$ (Carrara, 2015) .....	29
Figura 32– Algumas das infinitas possibilidades de configuração do manipulador para alcançar um mesmo ponto P (Carrara, 2015).....	30
Figura 33 – Braço TRR em movimento no espaço. Adaptado (Carrara, 2015).....	31
Figura 34– Vistas dos movimentos do manipulador TRR. Adaptado (Carrara, 2015).....	31

Figura 35 – Braço robótico do tipo pick-and-place aplicado à indústria alimentícia (Knight, 2018).....	34
Figura 36 – Robô OEM de Solda MIG HA1440 (Ali Express, s.d.) .....	35
Figura 37– Robô do GRACO, UnB .....	36
Figura 38 – Trajetória de um efetuador com velocidades máximas das juntas (esquerda) e velocidades sincronizadas (direita) (Carrara, 2015).....	36
Figura 39 – Trajetórias possíveis de um efetuador controlado por CP (Carrara, 2015). .....	37
Figura 40 - Interpolação e filtragem de pontos de passagem (Rosário, 2012).....	38
Figura 41– Posições das juntas para a descrição de uma trajetória retilínea por um robô planar (Carrara, 2015) .....	38
Figura 42– Taxa de variação dos ângulos das juntas em função do tempo (Carrara, 2015)....	39
Figura 43– Taxa de variação das velocidades das juntas em função do tempo (Carrara, 2015) .....	39
Figura 44 – Contatos normalmente aberto e fechado (Bluelux, 2018) .....	42
Figura 45 – Botões pulsadores/sem retenção (Botoeiras e Sinalizadores, s.d.) .....	42
Figura 46 – Chave Seletora (Botoeiras e Sinalizadores, s.d.) .....	43
Figura 47 – Joystick analógico (Addicore, 2016) .....	44
Figura 48 - Movimentos possíveis da cabeça/pescoço de um ser humano. Adaptado (Cabeça e Face, 2010).....	47
Figura 49 - Movimentos pertinentes da cabeça/pescoço de um ser humano. Adaptado (Cabeça e Face, 2010) .....	48
Figura 50 - Trajetória de trabalho das juntas 5 e 6.....	52
Figura 51 - Área de trabalho do subproblema 1 .....	53
Figura 52 - Linhas de construção volume de trabalho .....	53
Figura 53 - Área de trabalho das juntas 2 a 4.....	54
Figura 54 - Área lateral final, com os movimentos das juntas 2 a 6 .....	54
Figura 55 - Área de trabalho lateral do manipulador considerando movimento de todas as juntas .....	55
Figura 56 - Volume de trabalho do manipulador robótico assistivo .....	55
Figura 57 - Cotovelo para baixo.....	57
Figura 58 - Cotovelo para cima.....	57
Figura 59 - Modelagem de um robô planar.....	58
Figura 60 – Separação do manipulador em posicionamento e orientação. Adaptado (Néto, 2019).....	59
Figura 61 – Modelagem do manipulador com pulso esférico .....	59
Figura 62 - Vista isométrica da posição $qa$ .....	61
Figura 63 - Vista superior do manipulador na posição $qa$ .....	61
Figura 64 - Vista frontal do manipulador na posição $qa$ .....	62
Figura 65 - Variável auxiliar $r$ .....	63
Figura 66 - Ângulos auxiliares no plano frontal .....	64
Figura 67 - Desenho esquemático do manipulador em seus máximos e mínimos.....	65
Figura 68 - Trajetórias do efetuador, a partir da movimentação da junta $J3$ .....	66
Figura 69 - Contornos externos (amarelo e vermelho) da área de trabalho lateral do manipulador.....	66
Figura 70 - Volume de trabalho atualizado do manipulador.....	67
Figura 71 - Trigonometria de um joystick de 2 eixos (Last Minute Engineers, 2019) .....	69
Figura 72 - Esquema de como implementar o planejamento da trajetória.....	70
Figura 73 - Algoritmo de implementação do controle cartesiano do efetuador no manipulador real .....	73
Figura 74 - Manipulador planar para testes.....	74

Figura 75 - Algoritmo para implementação do controle cartesiano do efetuador no manipulador planar.....	76
Figura 76 - Diagrama de simulação da solução no MATLAB para o robô planar .....	77
Figura 77 - Trajetória Real do efetuador na primeira simulação (2 segundos).....	78
Figura 78 - Trajetória Real do efetuador na segunda simulação (1 segundo).....	78
Figura 79 - Trajetória Real versus Teórica do efetuador (1 segundo) .....	79
Figura 80 - Trajetória Real 2° Quadrante (1 segundo).....	80
Figura 81 - Trajetória Real versus Teórica 2° Quadrante (0,5 segundo) .....	80
Figura 82 - Algoritmo para correção de erros e simulação da solução no MATLAB em um robô planar.....	82
Figura 83 - Trajetória Real versus Teórica no 2° Quadrante (0,5 segundo), com o algoritmo de correção .....	83
Figura 84 - Trajetória Real versus Teórica no 2° Quadrante (0,25), com o algoritmo de correção .....	83



## LISTA DE TABELAS

Tabela 1– Dimensões dos elos (Néto, 2019).....	7
Tabela 2– Limites Angulares das Juntas (Néto, 2019).....	8
Tabela 3 – Parâmetros de Denavit-Hartenberg .....	8
Tabela 4 – Torques necessários em cada junta (Néto, 2019) .....	10
Tabela 5 – Matriz de funções .....	46
Tabela 6 - Comandos necessários para controle da posição escalar do efetuador terminal.....	50
Tabela 7 - Matriz de validação da funções fkine e ikcon da Toolbox.....	58
Tabela 8 – Novos Limites angulares das juntas .....	65
Tabela 9 - Leitura do joystick e velocidade do efetuador .....	71
Tabela 10 - Parâmetros D-H do robô planar .....	74
Tabela 11 – Erro (Teórico-Real) das coordenadas Y para o 2° Quadrante.....	81
Tabela 12 - Erro relativo entre os pontos reais para o 2° Quadrante .....	81
Tabela 13 - Erro (Teórico-Real) das coordenadas Y para o 2° Quadrante após correção .....	84

# LISTA DE SÍMBOLOS

## Símbolos Latinos

$a$	Distância em $x_i$
$A$	Matriz de transformação homogênea
$\{A\}$	Sistema de coordenadas A
$\{B\}$	Sistema de coordenadas B
$C$	Cosseno
$d$	Distância em $z_{i-1}$
$n$	Momento
$N$	Momento em relação ao centro de massa
$m$	Massa
$l$	Comprimento da viga
$J$	Jacobiano
$P$	Vetor de Translação
$R$	Matriz de Rotação
$x$	Vetor posição do Efetuador terminal
$\{S\}$	Sistema de coordenadas
$S$	Seno
$t$	Tempo
$T$	Matriz de Transformação
$v$	Velocidade linear
$V$	Velocidade
$Q$	Vetor Posição

## Símbolos Gregos

$\alpha$	Ângulo em torno de $x_i$
$\Delta$	Varição entre duas grandezas similares
$\theta$	Ângulo em torno de $z_{i-1}$
$\Omega$	Velocidade angular do sistema de coordenadas
$\omega$	Velocidade angular
$\tau$	Vetor do momento
$\rho$	Raio de curvatura

## Subscritos

$ORG$	Origem
$médio$	Média Aritmética
$JACO$	WMRA JACO
$Manus Arm$	WMRA iARM
$Kares$	WMRA Kares

## Sobrescritos

•	Varição temporal
---	------------------

## **Siglas**

ABRAFIN	Associação Brasileira de Fisioterapia Neurofuncional
D-H	Denavit-Hartenberg
GDL	Grau de Liberdade
IBGE	Instituto Brasileiro de Geografia e Estatística
iARM	<i>Intelligent Arm Robot Manipulator</i>
OMS	Organização Mundial da Saúde
ONU	Organização das Nações Unidas
TA	Tecnologia Assitiva
WMRA	Wheelchair mounted robotic arm

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

Mais de um bilhão de pessoas em todo o mundo convive com alguma forma de deficiência, das quais aproximadamente 200 milhões vivenciam dificuldades funcionais consideráveis. Isso se torna uma preocupação para os próximos anos, tendo em vista que a população tende a envelhecer com os avanços na qualidade de vida, ao passo de que sua suscetibilidade à ocorrência de acidentes que levem à deficiências é maior, além de haver também um aumento global de doenças crônicas (Organização Mundial da Saúde, 2011).

Em um recorte local, o mais recente levantamento sobre pessoas com deficiência no Brasil, realizado em 2010 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), revelou que 23,9% da população (aproximadamente 45,6 milhões de pessoas) apresenta algum tipo de deficiência, sendo que 7% sofre de alguma disfunção motora (IBGE, 2012).

Desde os anos 1970, as reações a um cenário de pessoas com deficiência têm mudado, principalmente por iniciativa de pessoas que possuem algum tipo de deficiência, mas também pela evolução moral do ser humano ao longo dos anos, que passa a encarar a deficiência como uma questão de direitos humanos. As pessoas com deficiência são historicamente atendidas com soluções segregacionistas, como escolas especiais e abrigo. Todavia, as políticas mudaram em prol das comunidades e da inclusão social, e as soluções propostas pela medicina passaram a focar no indivíduo e a incluí-lo no desenvolvimento da solução (Organização Mundial da Saúde, 2011).

Nesse contexto, a Tecnologia Assistiva (TA) tem sido uma aliada na reinserção dessas pessoas na sociedade e na realização de atividades cotidianas por elas. Tecnologia assistiva é definida como uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas encontrados pelos indivíduos com deficiências (Cook & Hussey, 2007). E muitos são os atores envolvidos em trabalhos relacionados com tecnologia, desde a iniciativa privada, até a própria Organização Mundial de Saúde (OMS). Esta reconhece que atualmente são necessários mais de um bilhão de dispositivos assistivos, para suprir a demanda. A OMS informa também que este número deve dobrar até 2050 (Nardini, 2016).

Porém, muitas pessoas com deficiência não têm acesso igualitário à assistência médica, educação, oportunidade de emprego, e não recebem serviços correspondentes à deficiência da qual são portadoras. Toda essa desigualdade submete essas pessoas a uma situação de exclusão de atividades sociais cotidianas. Há cada vez mais evidências de que pessoas com deficiências experimentam piores resultados socioeconômicos em comparação com pessoas não portadoras de quaisquer tipos de deficiência (Organização Mundial da Saúde, 2011).

Em virtude disso, atualmente, apenas 1 a cada 10 pessoas que necessitam de uma tecnologia assistiva possuem acesso a ela, em virtude dos altos custos, falta de conscientização, disponibilidade, falta de pessoas capacitadas e treinadas para utilização, além de falta de políticas e de financiamentos, públicos ou privados (Organização Mundial da Saúde, 2018).

Este trabalho foca no atendimento a pessoas com lesões medulares. Para este perfil de paciente, uma tecnologia ascendente é a de braços robóticos montados em cadeiras de rodas (WMRA – Wheelchair mounted robotic arm), que consiste em um manipulador robótico acoplado a uma cadeira de rodas elétrica, com controles, da cadeira e do braço, integrados.

No entanto, para o caso de pacientes com tetraplegia, público-alvo a ser contemplado pelo produto deste trabalho, o desafio de controlar a cadeira, e o braço, se torna ainda maior, pela falta de movimento do pescoço para baixo, descartando alternativas de controle comuns operadas com as mãos. Além do mais, é desejável que o controle seja feito por movimentos naturais do corpo, sem gerar desconforto ao usuário (EUA Patente Nº 4207959, 1980).

Todo esse contexto configura um grande desafio para as tecnologias assistivas pois, além de serem efetivas em fornecer autonomia e independência aos usuários, precisam ser acessíveis financeira e intelectualmente.

## 1.2 OBJETIVOS

Donos de algumas das maiores empresas do mundo, Jorge Paulo Lemann e seus sócios Beto Sucupira e Marcel Telles, têm como lema de que “Sonhar grande e sonhar pequeno dá o mesmo trabalho” (Exame, 2015). Com isso em mente, este trabalho tem o objetivo de oferecer aos usuários autonomia e independência a pacientes tetraplégicos, os possibilitando atingir seu máximo potencial como seres humanos.

“A deficiência não precisa ser um obstáculo para o sucesso. Durante praticamente toda a minha vida adulta sofri da doença do neurônio motor.

Mesmo assim, isso não me impediu de ter uma destacada carreira de astrofísico e uma vida familiar feliz.”.

(Stephen W Hawking, 2011, OMS, Relatório Mundial sobre Deficiência,p. ix)

Contudo, sonhar alto exige começar pequeno (Endeavor Brasil, 2015). Assim, para atingir este sonho grande, o presente trabalho deseja controlar um manipulador robótico montado em uma cadeira de rodas comercial, projetado e construído anteriormente (Néto, 2019). Este, ainda grande, objetivo tem como objetivos específicos:

- Cálculo da cinemática inversa do manipulador; e

- Planejamento e geração da trajetória do efetuator terminal.

### 1.3 METODOLOGIA E ORGANIZAÇÃO DO TRABALHO

A metodologia de projeto adotada neste trabalho não se restringiu a uma única fonte, nem mesmo se ateve apenas a metodologias tradicionais. Foram usadas algumas ferramentas do Design Thinking (MJV) e alguns conceitos e orientações do Guia PMBOK® (Project Management Institute, 2017), além de uma metodologia de um livro texto de Engenharia, que conta com 10 etapas (Norton, 2013).

Com bases nas fontes citadas, o projeto seguiu a seguinte metodologia:

- Contextualização e Definição do problema: esta etapa consiste em extrapolar o problema em si e entender qual o papel deste problema na sociedade, seus impactos, além de entender quem é afetado por ele, ou seja, o seu público-alvo. É importante entender quais as necessidades e dores dos usuários;
- Pesquisa de suporte: esta etapa é caracterizada por construir todo o arcabouço teórico necessário para compreender as análises e cálculos que serão feitos, além de fornecer embasamento para a proposição de uma solução. Nesta etapa ocorre também a exploração de que já foi feito neste campo de estudo;
- Metodologia de análise e Geração de soluções: esta etapa consiste na definição das diretrizes e premissas que irão guiar as análises que serão feitas em seguida, buscando compreender o que se entende por uma solução bem-sucedida, os pré-requisitos e até onde se deseja ir com o projeto;
- Síntese e Avaliação: esta etapa é responsável por compilar e organizar as soluções encontradas, avaliando seus mecanismos de funcionamento, a relevância e o grau de atendimento destas aos requisitos do projeto. A partir desta etapa se começa a vislumbrar como seria a solução mais adequada;
- Conceito do projeto: com base no extraído e percorrido nas etapas anteriores, propõe-se então o conceito de uma solução, que atenda às necessidades, premissas e limitações do projeto;
- Detalhamento da solução: esta etapa consiste no detalhamento e desenvolvimento, com auxílio da teoria absorvida pela pesquisa de suporte e das análises feitas, do conceito proposto anteriormente;
- Validação: por fim, esta etapa é responsável por ensaiar a solução desenvolvida, corrigindo e fazendo os ajustes necessários, até o alcance de um produto que atenda aos pré-requisitos e objetivos definidos para o projeto.

De posse da definição da metodologia a ser utilizada, este trabalho foi então organizado da seguinte maneira:

- Introdução: seção que contempla a etapa de Contextualização e Definição do problema;
- Revisão Bibliográfica: esta seção abarca a etapa de Pesquisa de Suporte para embasamento teórico do projeto. É importante citar que nesta etapa se apresenta os dois projetos diretamente relacionados, e usados como base, com o presente trabalho;
- Solução Conceitual: esta seção abrange as etapas de Metodologia de análise e Geração de soluções, Síntese e Avaliação e o próprio Conceito do projeto. Ao fim desta seção será possível entender o que será desenvolvido;
- Desenvolvimento da Solução: nesta seção se encontra a etapa de Detalhamento da solução;
- Testes: nesta seção deseja-se registrar como foi o desempenho do produto “em ação”, ou seja, a etapa de Validação;
- Conclusão: esta seção contempla as análises finais sobre o desempenho do projeto desenvolvido, identificando falhas e sugerindo melhorias e próximos passos de desenvolvimento do projeto;
- Referências Bibliográficas, Anexos e Apêndices: seção que organiza todos os materiais consultados para o desenvolvimento do projeto, além daqueles desenvolvidos ao longo dele e complementam o que foi registrado em seu corpo, nas seções anteriores.

O diagrama do projeto de graduação, com a metodologia dentro da organização do trabalho utilizada, é ilustrado pela Figura 1:

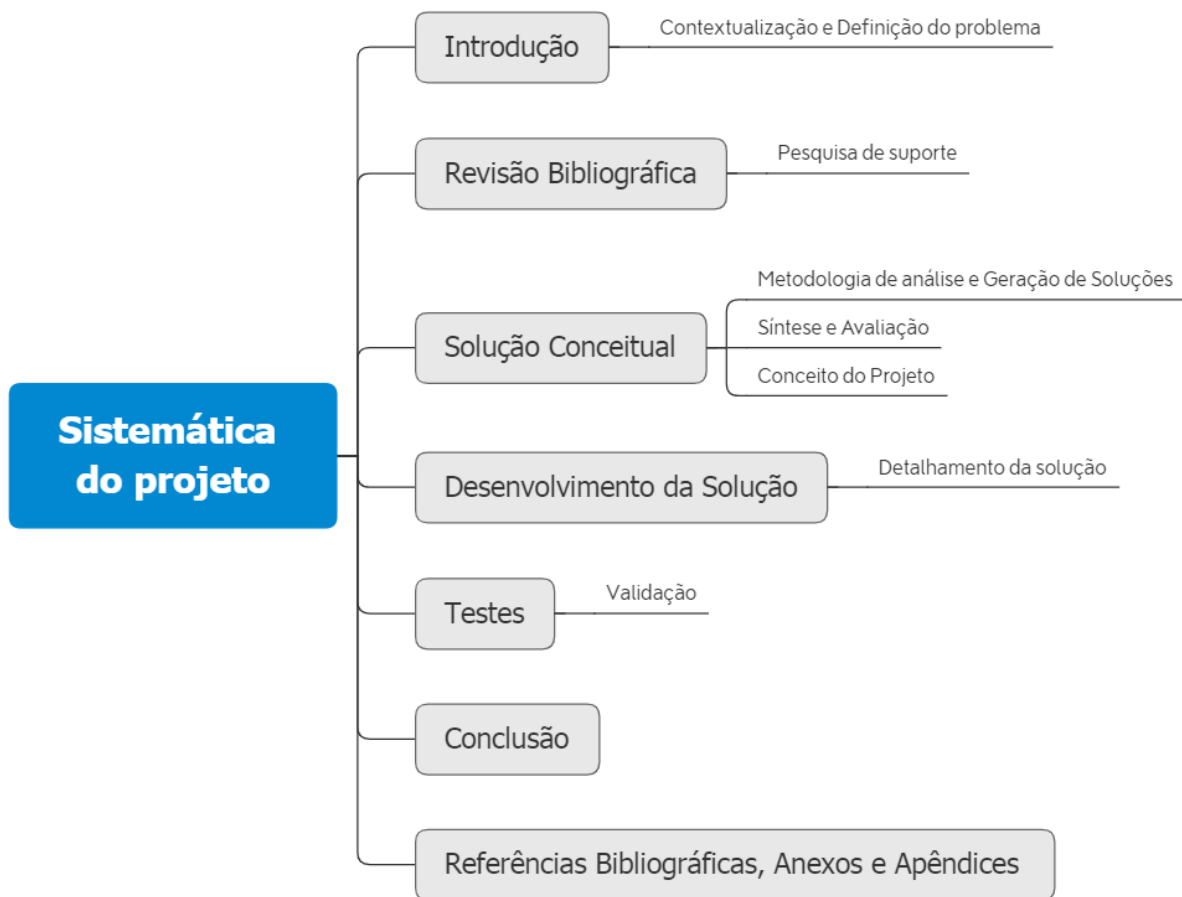


Figura 1 – Sistemática do projeto de graduação



## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 TRABALHOS ANTERIORES

Este projeto tem como objeto de estudo o manipulador robótico desenvolvido anteriormente em um Projeto de Graduação pela Universidade de Brasília (Néto, 2019). O projeto de base teve como objetivos o projeto e construção de um protótipo de baixo custo de um manipulador assistivo, a ser montado em uma cadeira de rodas elétrica, com o intuito de atender às necessidades motoras de pessoas com deficiências nos membros inferiores e superiores.

Como já mencionado, este trabalho almeja controlar os movimentos do manipulador projetado. Assim, é necessária a compreensão dos resultados alcançados previamente, os quais serão úteis para o atingimento dos objetivos postos.

#### 2.1.1 GEOMETRIA DO MANIPULADOR

Trata-se de um robô de configuração articulada e 3 graus de liberdade, além dos 3 do punho, para os movimentos de *roll*, *pitch* e *yaw*, representado nas Figuras 2 e 3. Uma das premissas do robô é de que apresente um custo de montagem e fabricação acessível, em comparação aos modelos existentes no mercado. Essa premissa pautou diversas decisões do projeto anterior, e guiará igualmente as do presente trabalho.

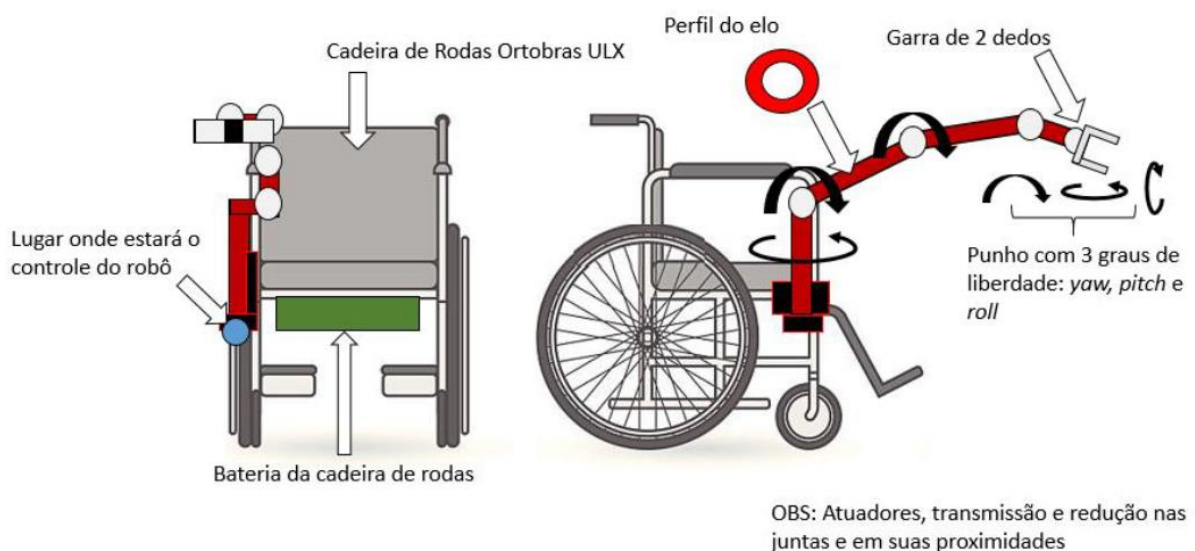


Figura 2 – Conceito da solução (Néto, 2019)

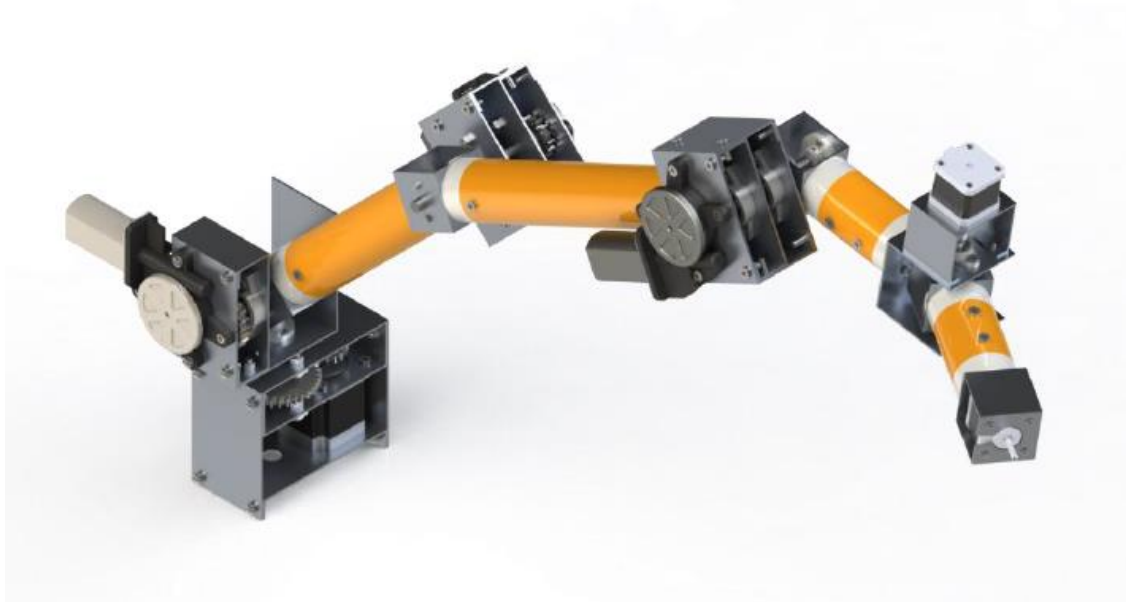


Figura 3 – Modelo em SolidWorks da solução proposta, em construção (Néto, 2019)

Os elos principais do manipulador (Figura 4) possuem as dimensões indicadas na Tabela 1, possibilitando a ele um alcance de 1,1 m. Por suas características construtivas, o braço robótico tem um *payload*, ou seja, capacidade de carga, de 1,2 kg.

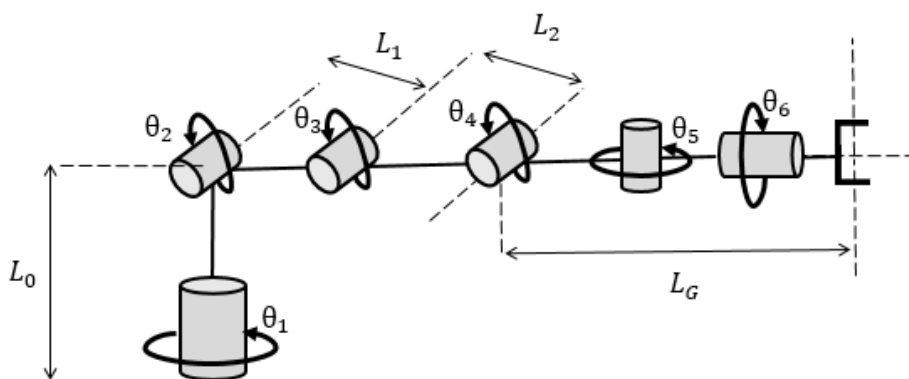


Figura 4 – Dimensões dos elos (Néto, 2019)

Tabela 1 – Dimensões dos elos (Néto, 2019)

	$L_1$	$L_2$	$L_3$	$L_G$
Comprimento	0,17 m	0,35 m	0,35 m	0,35 m

Para ter o alcance de 1,1 m mencionado acima e respeitar os limites construtivos do manipulador, também foram calculados dos limites angulares das juntas, sintetizados na Tabela 2:

Tabela 2– Limites Angulares das Juntas (Néto, 2019)

	$\theta_1$ (Base)	$\theta_2$ (Ombro)	$\theta_3$ (Cotovelo)	$\theta_4$ (Pitch)	$\theta_5$ (Yaw)	$\theta_6$ (Roll)
Limite Angular	$\pm 90^\circ$	$-45^\circ$ a $+135^\circ$	$-135^\circ$ a $+160^\circ$	$\pm 160^\circ$	$\pm 90^\circ$	$\pm 360^\circ$

### 2.1.2 MODELAGEM CINEMÁTICA

A modelagem cinemática proposta (Néto, 2019), embasada no desenho esquemático da Figura 5, forneceu os seguintes parâmetros de Denavit e Hartenberg (Tabela 3):

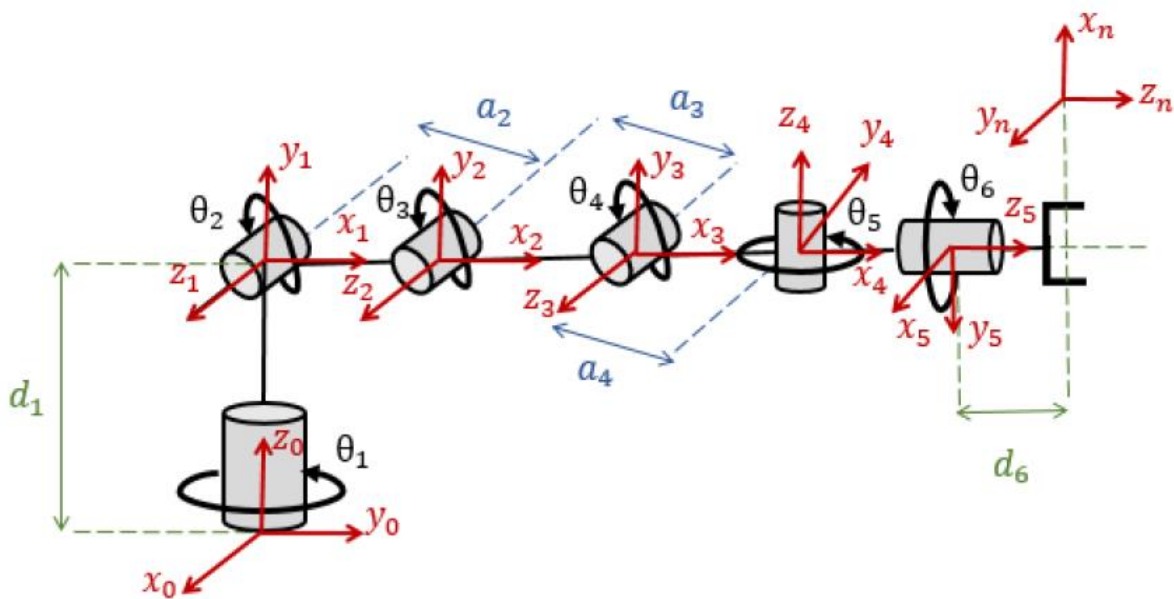


Figura 5 – Desenho esquemático do manipulador com os parâmetros D-H (Néto, 2019)

Tabela 3 – Parâmetros de Denavit-Hartenberg

Junta	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1^*$	$d_1$	0	$+90^\circ$
2	$\theta_2^*$	0	$a_2$	0
3	$\theta_3^*$	0	$a_3$	0
4	$\theta_4^*$	0	$a_4$	$-90^\circ$
5	$\theta_5^*$	0	0	$-90^\circ$
6	$\theta_6^*$	$d_6$	0	0

De posse desses parâmetros, as matrizes de transformações para cada uma das 6 juntas (Eq. 1 a 6) foram construídas:

$${}^1_0\mathbf{A} = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^2_1\mathbf{A} = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^3_2\mathbf{A} = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_3C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & a_3S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^4_3\mathbf{A} = \begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & a_4C\theta_4 \\ S\theta_4 & 0 & C\theta_4 & a_4S\theta_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^5_4\mathbf{A} = \begin{bmatrix} C\theta_5 & 0 & -S\theta_5 & 0 \\ S\theta_5 & 0 & C\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^6_5\mathbf{A} = \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Assim, a matriz de transformação homogênea da junta 1 para a junta 6 é dada por:

$${}^6_0\mathbf{A} = {}^1_0\mathbf{A}{}^2_1\mathbf{A}{}^3_2\mathbf{A}{}^4_3\mathbf{A}{}^5_4\mathbf{A}{}^6_5\mathbf{A} \quad (7)$$

Como a Equação 7 fornece a posição e a orientação do efetuador terminal em relação ao sistema de coordenadas da base, pode ser reescrita como:

$${}^6_0\mathbf{A} = \begin{bmatrix} {}^6_0R & {}^6_0x \\ 0 & 1 \end{bmatrix} \quad (8)$$

### 2.1.3 ANÁLISE DINÂMICA

A análise dinâmica do manipulador foi desenvolvida (Néto, 2019) por diversas abordagens, e retornou os valores de torque nas juntas, registrados na Tabela 4, já considerando o fator de segurança global do sistema igual a 1,36:

Tabela 4 – Torques necessários em cada junta (Néto, 2019)

Juntas	Torque calculado (N.m)
1	1,07
2	23,62
3	12,33
4	2,87
5	0,15
6	0,02

#### 2.1.4 MOTORES E SISTEMA DE REDUÇÃO

O manipulador utiliza motores de passo, circulosados na Figura 6, e motores DC, enquadrados na Figura 6.

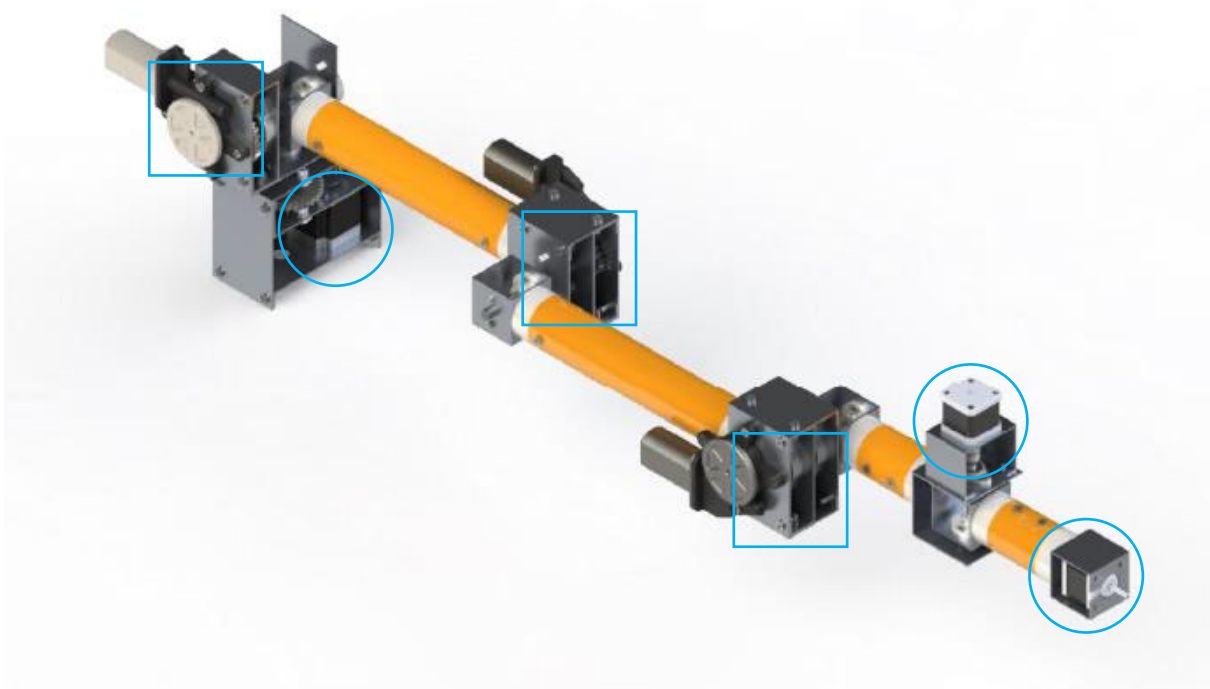


Figura 6 – Indicações dos motores de cada junta. Adaptado (Néto, 2019)

No que tange ao controle desses motores, o autor sugeriu a utilização de Arduino, tanto pela facilidade de acesso financeiro, quanto a intelectual, por ser uma linguagem *OpenSource*. Essa decisão foi mantida no presente trabalho.

## 2.2 LESÃO MEDULAR, TETRAPLEGIA E REABILITAÇÃO

A lesão da medula espinhal é um dos mais graves acontecimentos que pode afetar o ser humano e com enorme repercussão física, psíquica e social. Lesão medular é toda injúria às estruturas contidas no canal medular (medula, cone medular e cauda equina), podendo levar a alterações motoras, sensitivas, autonômicas e psicoafetivas (Ministério da Saúde, 2013).

As classificações de nível e tamanho de uma lesão medular (ABRAFIN, s.d.) permitem compreender quais membros, movimentos e sensações foram prejudicados por ela, e se esse prejuízo é total ou parcial. Uma vez que o foco deste trabalho, e dos que o antecedem, é nas lesões nas vértebras 3 a 5, a Figura 7 apresenta um recorte da tabela de perspectivas funcionais de indivíduos com lesão medular completa, definida como tetraplegia, por nível de lesão (Ministério da Saúde, 2013).

Último nível inervado/ principais músculos acrescentados	Metas funcionais
<b>C1-C3</b> Esternocleidomastóideo, possível parte do trapézio e do diafragma	Locomoção e alívio de pressão com cadeira de rodas motorizada com inclinação eletrônica (controle cervical ou mentoniano), uso do computador com <i>softwares</i> de controle de <i>mouse</i> /emuladores de <i>mouse</i> , uso de unidades de controle ambiental, dependente de respirador portátil.
<b>C4</b> Inervação plena do diafragma e trapézio	Locomoção e alívio de pressão com cadeira de rodas motorizada com inclinação eletrônica (controle cervical, mentoniano ou sugar/soprar), uso do computador com <i>softwares</i> de controle de <i>mouse</i> /emuladores de <i>mouse</i> ou <i>clip</i> (bucal, mentoniano ou de cabeça), uso de unidades de controle ambiental.
<b>C5</b> Parte do deltoide, parte do bíceps, romboides, parte do manguito rotador	Capaz de auxiliar no banho, alimentação, escrita e auto-cuidado com adaptações utilizando estabilizador de punho. Auxilia no vestuário de membros superiores. Locomoção com cadeira de rodas convencional com pinos nos sobrelaros e mancal adiantado + rodas anti-tombo. Transferências com apoio de cinta de transferências. Uso do computador com suporte de punho e <i>clip</i> para digitação e/ou <i>softwares</i> de controle do <i>mouse</i> . Capaz de realizar alívios de pressão por deslocamento mínimo do corpo.

Figura 7 – Perspectivas funcionais de acordo com nível de lesão medular. Adaptado (Ministério da Saúde, 2013)

O programa de reabilitação de uma pessoa com lesão medular tem início no atendimento prestado ao paciente ainda na urgência ou quando da definição do diagnóstico nos casos não traumáticos, e segue até reinserção do indivíduo no convívio social. Para a conclusão dessa reinserção deve-se recorrer não apenas a soluções químicas, físicas e biológicas, mas também a soluções de acompanhamento psicológico (Ministério da Saúde, 2013).

Um paciente com lesão na medula costuma recorrer a uma cadeira de rodas em busca de conforto, segurança e posicionamento adequados, além de boa funcionalidade e independência (Ministério da Saúde, 2013). Especificamente para pacientes com lesões nas vértebras 3 a 5, braços robóticos são mais apropriados para o processo de reabilitação (M. Alqasemi, J. McCaffrey, D. Edwards, & V. Dubey, 2005).

Com essa nova interface do paciente com o mundo, todo o seu processo de reabilitação e adaptação ao dispositivo deve ser abastecido de informações e treinamentos sobre a nova condição, tanto para ele, quanto seu cuidador, o que insere mais um desafio nesse processo, qual seja, o de haver uma linguagem simples de comunicação das vontades do paciente para o dispositivo robótico, a fim deste realmente proporcionar ao usuário autonomia e independência.

## 2.3 ESTADO DA ARTE

As pesquisas em torno de braços robóticos montados em cadeiras de rodas (WMRA – Wheelchair Mounted Robotic Arms) não são de hoje, e mais do que o design e habilidades do braço em si, a interface homem-máquina é vasta fonte de alternativas e fértil campo para mais pesquisas e mais desenvolvimento. Com isso em vista, é imperativo apurar e avaliar as alternativas já existentes (estado da arte) no mercado e na academia, com um olhar voltado para o controle e interface com o usuário do braço, e não mais para a parte mecânica, como desenvolvida anteriormente. Como este trabalho foca na interface de comando e na geração de trajetórias de um dispositivo por pacientes tetraplégicos, a pesquisa foi extrapolada à WMRA, trazendo outros mecanismos a serem controlados. A seguir, algumas das soluções estudadas.

### a. KARES

KARES é um acrônimo para KAIST Rehabilitation Engineering System, um sistema inteligente de reabilitação de um manipulador montado em cadeira de rodas com 6 graus de liberdade, mostrado na Figura 8 (Song, Lee, Kim, Yoon, & Bien, 1998). Como melhoria em relação à versão anterior, a versão estudada utiliza uma câmera com leitura colorida e um sensor de força acoplado ao efetuator terminal. Esses dois acréscimos foram feitos para facilitar que o manipulador efetue as tarefas de maneira autônoma. Para o controle autônomo das 6 juntas e da garra, foi desenvolvido um controlador “multi-motion”. Com relação ao controle manual, foi utilizado um dispositivo com 6 graus de liberdade e 10 chaves. Além disso, o manipulador aceita entradas de voz simples, como “start” e “stop”.

O KARES é programado para realizar 4 atividades básicas: pegar um copo em cima de uma mesa, pegar uma caneta no chão, mover objetos pequenos em direção ao rosto do usuário e operar um interruptor na parede. Por se desejar utilizar o robô em ambientes desestruturados, a utilização da câmera de cores fez-se útil para identificação dos objetos/alvos no ambiente. Em alguns testes, foi necessário fazer intervenções (marcações coloridas) para facilitar essa identificação.

Na parte de programação e movimentação do robô, a estimativa da posição do objeto desejado é feita, no plano XY, analisando o plano da imagem da câmera. Em relação à profundidade (eixo Z), o sistema utiliza uma câmera montada na garra, e o princípio de aumento da imagem do objeto conforme diminui a distância do efetuador até ele. Por fim, o sensor de força é usado para conferir o sucesso, ou fracasso, no agarramento do objeto desejado.

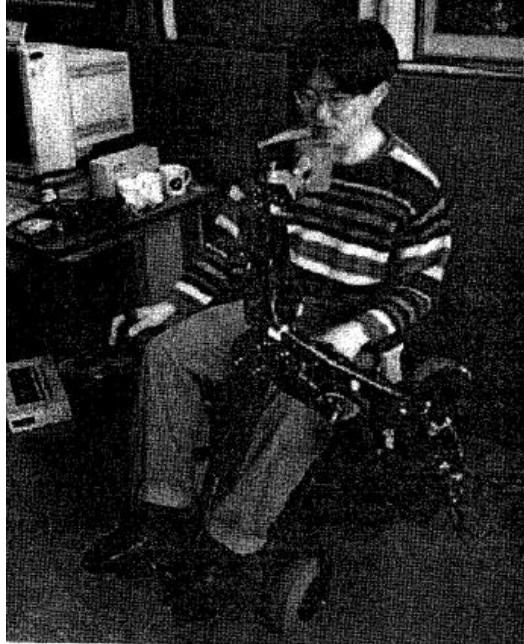


Figura 8 – Usuário do KARES (Song, Lee, Kim, Yoon, & Bien, 1998).

#### b. MANUS Arm Robot

O MANUS é um famoso WMRA (Figura 9), produzido pela *Exact Dynamics*, uma empresa europeia que faz parte do grupo Assistive Innovations (Exact Dynamics, 2018). O MANUS possui uma garra de dois dedos como efetuador terminal e possui 6+2 graus de liberdade, com encoders em cada junta. Em seu modelo original, ele é controlado por meio do acesso a menus que seleciona entre controle cartesiano do efetuador, ou controle junta a junta, através de um joystick, teclado ou interruptor simples. Há muitas pesquisas em torno do MANUS, e por isso há várias soluções de controle para ele.





Figura 9 – Usuário do Manus Arm, com o controle original por joystick (Exact Dynamics, 2018).

Uma delas, desenvolvida na Universidade Pierre & Marie Curie, na França, utiliza 3 modos de trabalho: RM (Record Mode), PPCM (Point-to-point Control Mode) e RCM (Replay Control Mode). O primeiro é usado para aproximação de um objeto desejado e para gravar posições dos elos do manipulador, normalmente posições e atividades recorrentes. Já o segundo utiliza determinações de posição ( $xyz$ ) e orientação ( $yaw$ ,  $pitch$  e  $roll$ ). Por fim, o último acessa e repete movimentos pré-programados. Além disso, no modo PPCM, 12 posições/gestos foram pré-programados (Figura 10), os quais podem ser acessados por um teclado. Há também a possibilidade de controlar o manipulador por um joystick de dois graus de liberdade, normalmente usados para ajustes mais finos, podendo controlar a posição do efetuador, ou controlar separadamente cada junta. Os dois dispositivos de controle executam o comando desejado enquanto o botão é pressionado/joystick movimentado, e cessam quando o toque é suspenso.

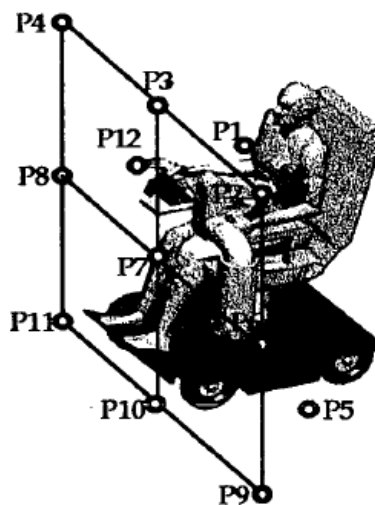
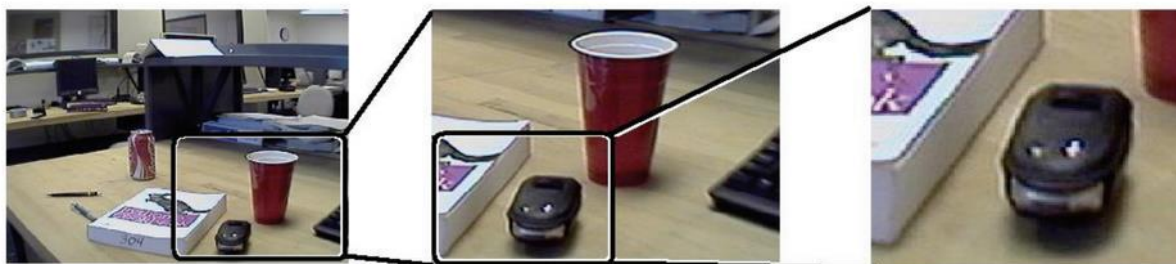


Figura 10 – Distribuição espacial das 12 posições pré-programadas do MANUS (Didi, Grandjean, Mokhtari, & Roby-Brami, 1998)

Outra pesquisa se debruçou em simplificar o controle já existente do MANUS e diminuir o esforço do usuário utilizando uma interface visual (Tsui & Yanco). Essa pesquisa foca no alcance aproximado e grosseiro da localização desejada, e não no ajuste fino da posição da garra. Para isso, a interface visual divide em 4 quadrantes a imagem ao vivo gerada por uma câmera no ombro do usuário, e ele escolhe um desses quadrantes. Logo em seguida, o quadrante selecionado é novamente dividido em 4 outros, e o usuário escolhe novamente. Ou seja, a resolução do sistema é de 1/16 (Figura 13). O efetuador se posiciona no centro desse quadrante, no plano xy, e resta ao usuário controlar a profundidade (eixo z) do efetuador.



*Figura 11 – Partição sequencial da vista em 4 quadrantes (Tsui & Yanco)*

### c. Wheellesley

O Departamento de Ciência da Computação do Wellesley College (Yanco, Hazel, Peacock, Smith, & Wintermute, 1995), desenvolveu um projeto com enfoque no controle da cadeira de rodas em si, e não de um manipulador acoplado a ela. Os desenvolvedores propuseram uma solução para ambientes desestruturados, ou seja, ambientes diferentes de um totalmente mapeado e com condições controladas, como em um laboratório de testes. Isso foi feito com o intuito de justamente fornecer maior liberdade e independência ao usuário. A única demanda é a do ambiente cumprir com as normas de construção e dimensões para o trânsito de cadeiras de rodas.

O controle proposto para a cadeira é feito por uma interface gráfica, do tipo point-and-click, na qual o usuário escolhe em uma tela as funções que deseja executar. Além de teclas padrões de movimentação da cadeira e controle de sua velocidade, há algumas teclas de ações/obstáculos pré-programados, como “empurrar” e “puxar”, para portas, “mesa” e “rampa”. Mesmo com a intenção de independência do usuário em qualquer ambiente, alguns ambientes recorrentes foram mapeados, e ações pré-carregas foram definidas. Por fim, a cadeira de roda conta também com uma série de sensores ao seu redor, auxiliando o usuário com pontos cegos e evitando obstáculos. A Figura 14 mostra a página inicial da interface gráfica do sistema Wheellesley da cadeira de rodas, com os botões de movimentação, funções padrões, ambientes mapeados e um mapa dos sensores:

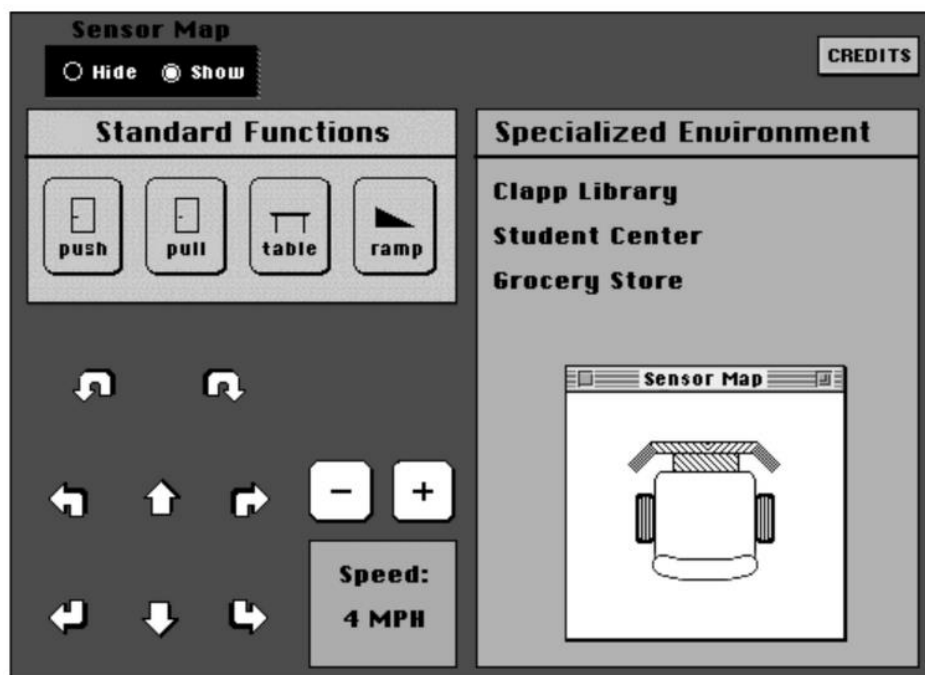


Figura 12 – Página inicial da interface gráfica com o usuário (Yanco, Hazel, Peacock, Smith, & Wintermute, 1995)

#### d. DeVAR & ProVAR

O DeVAR & ProVAR é uma solução diferentes das apresentadas até então, pois é um manipulador montado em um ambiente, e não em uma cadeira de rodas. Ele é constituído por duas arquiteturas complementares e que se comunicam entre si: Pinocchio e Jiminey. O primeiro é um controlador para um robô Puma 260, capaz de capturar diversos parâmetros, como velocidade da junta, limites de torque e força, posição do efetuador, ângulo das juntas e leituras dos sensores de proximidade presentes no braço. Com isso, ele recebe e realiza ações simples. Em outras palavras, ele é um controlador em tempo real do robô. O Jiminey, por sua vez, é responsável pela interface com o usuário (UI) e por monitorar, planejar e realizar atividades mais complexas. Ele pode se comunicar com o usuário de diversas maneiras: voz, movimentos da cabeça, interruptores, operado pelo assoprar e sugar (mouthstick), além de teclado e mouse/trackball (Wagner, Van der Loss, & Leifer, Two-Agent User Interface for an Assistive Robot, 1998).

Contudo, a UI do ProVAR é baseada em Java e utiliza realidade virtual (VRML – Virtual Reality Modeling Language). Assim, há um modelo 3D virtual do ambiente (Figura 13) e dos objetos “interessantes” nele presentes, e cada atividade pré-definida tem seus passos/etapas mapeados.



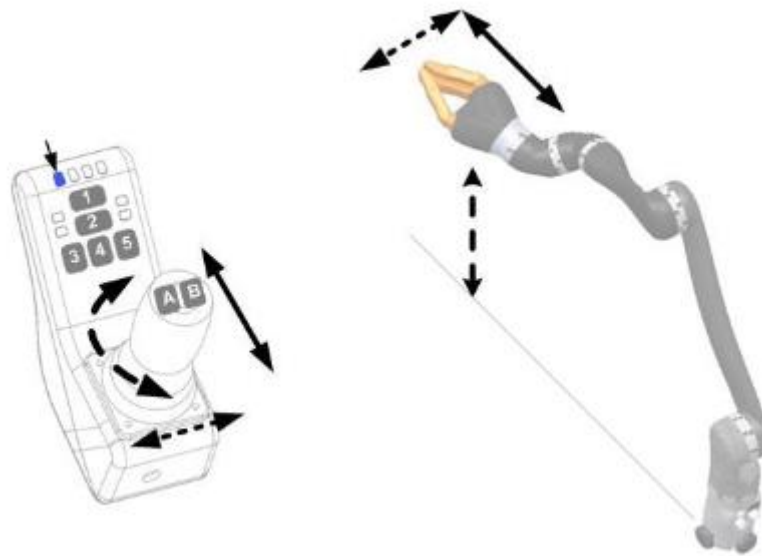
Figura 13 – Simulação virtual do movimento do manipulador (Wagner, Loos, Smaby, Chang, & Burgar, 1999)

#### e. JACO

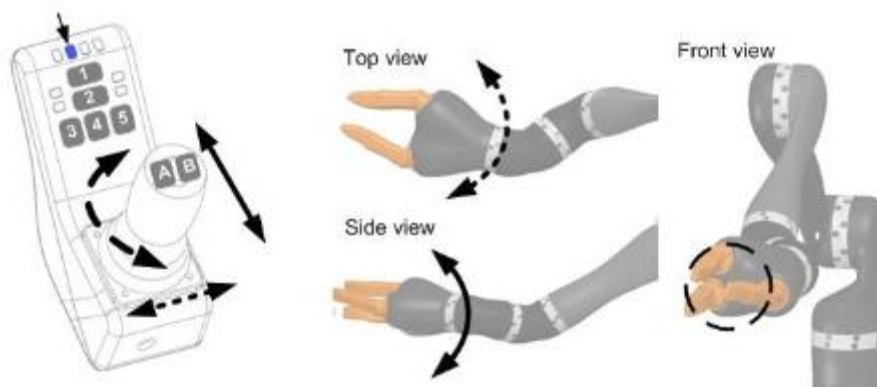
Esse é um braço robótico com 6 graus de liberdade, desenvolvido e comercializado pela Kinova, uma empresa canadense de soluções em tecnologia assistiva (Kinova Technology, s.d.). A empresa disponibiliza uma aplicação, para ser instalada em um computador, para gerenciar o braço robótico. Nessa aplicação é possível definir perfis de usuários, sensibilidade do joystick, assim como pré-definir movimentos, trajetórias, velocidades, posições e comandos, além de definir padrões para as atividades e objetos, como o tamanho comum de um copo, no “drinking mode”. Essa aplicação permite monitorar desempenho, temperatura e outros parâmetros do robô. Todas as configurações definidas são carregadas no braço via USB.

No que tange à interface com o usuário, a fabricante visa promover um controle intuitivo e minimizar a quantidade de movimentos necessários, realizados pelo usuário, a fim de atingir seu alvo. O JACO utiliza um joystick com 3 graus de liberdade e botões de seleção de movimentação/control: translação (em dois ou três eixos, depende do modelo), rotação, dedos e “drinking mode”. Ele ainda conta com leds que indicam e qual tipo de movimento/control está sendo usado. O JACO tem 3 modos de operação: direta, onde o usuário controla a garra e o JACO ajusta automaticamente as juntas para atender à solicitação; “translation mode”, onde o operador controla a posição da garra e o JACO garante que a garra ficará paralela ao solo durante todo o movimento; e “rotation mode”, no qual o usuário controla a

posição do braço em torno do ponto fixo central da garra, e o JACO assegura que esse ponto não irá se mover (Rehab). As Figuras 14 a 19 mostram como são esses modos de controle do manipulador.



*Figura 14 – Modo de controle direto dos 3 eixos do efetuator do JACO (Rehab)*



*Figura 15 – Modo de controle das juntas de orientação do JACO (Rehab)*

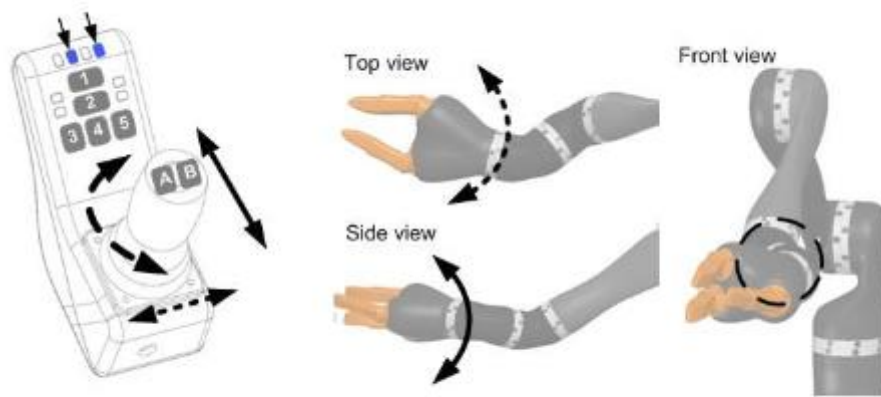


Figura 16 – Modo de controle das juntas de orientação com a opção “Drinking mode” (Rehab)

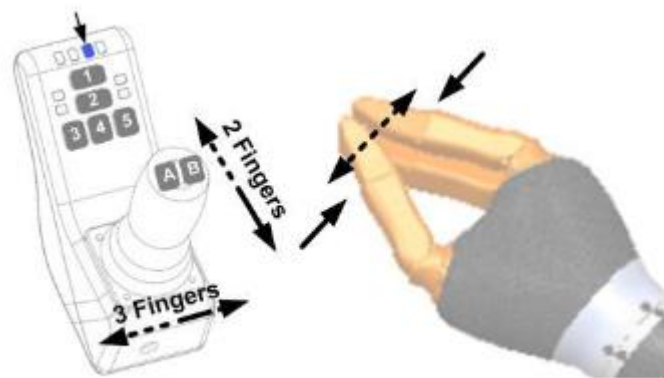


Figura 17 – Modo de controle dos dedos da garra do JACO (Rehab)

*f. Head Pointing Device*

Pesquisadores do Centro de Reabilitação de Yokohama, Japão (Hatakeyama, Kkoizumi, Umeda, Uchio, & Kasuga, 1998), desenvolveram um dispositivo apontador com luz infravermelha para pessoas com severas limitações físicas poderem usar computadores pessoais. O sistema é composto por um apontador, controlador e um computador pessoal. O dispositivo basicamente usa o ângulo de movimento da cabeça/pescoço do usuário e, por meio de cálculos e ajustes, define para qual a posição do cursor, em um display, o usuário estaria com os olhos voltados (Figura 20). O dispositivo utiliza dos movimentos de sugar e soprar para selecionar objetos na tela, como os botões de um mouse.

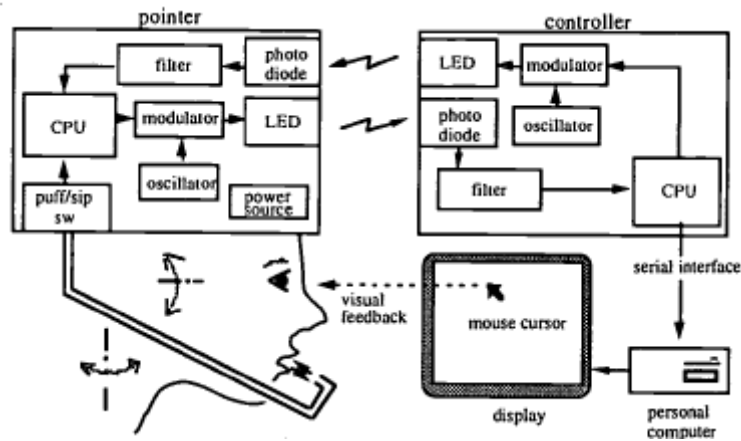


Figura 18 – Diagrama de blocos do funcionamento do dispositivo de apontamento com infravermelho (Hatakeyama, Kkoizumi, Umeda, Uchio, & Kasuga, 1998)

g. Handy 1

O Handy 1 (Topping, 2002) é outra solução não-WMRA: é uma estação de trabalho que tem sua finalidade ditada pela bandeja/mesa montada nele, podendo esta ser para: alimentação, higiene pessoal, maquiagem e lazer (pintura). Ou seja, ele é pré-carregado com um tipo de programa para cada finalidade. Pela limitação de possibilidades de uso, o Handy 1 tem uma interface de controle simples, com apenas um interruptor. A exemplo da plataforma de alimentação, ele utiliza uma série de sensores para escanear cada nicho de sua bandeja e executa as tarefas de “encher a colher e levar até a boca” apenas acionando o interruptor quando seus leds e sensores passarem pela divisão que contém o alimento desejado (Figura 19). Para repetir a “colherada do alimento” basta ativar o interruptor novamente. Há também a opção de selecionar um compartimento com bebida. O Handy 1 funciona para o usuário apenas quando ele está devidamente montado e organizado, ou seja, em um ambiente extremamente controlado e estruturado.

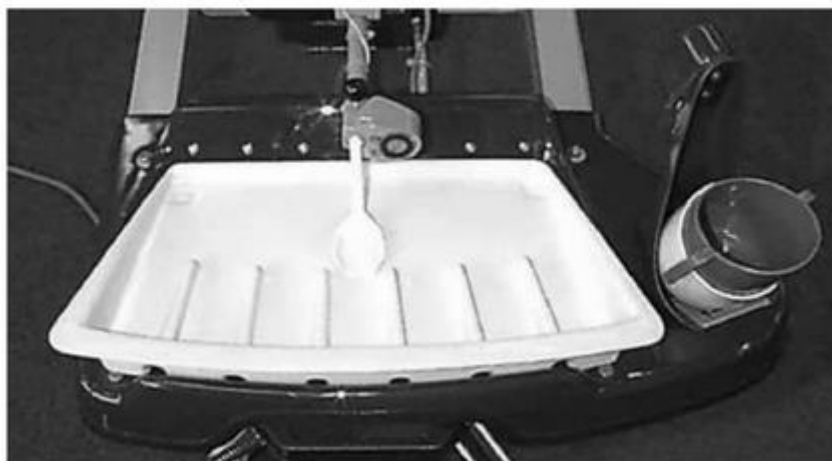


Figura 19 – Bandeja de alimentação e bebida do Handy 1 (Topping, 2002)

#### h. Designability

Um grupo de pesquisadores do Designability (Bath Institute of Medical Engineering, Reino Unido) apresentou na ICORR '99 (International Conference on Rehabilitation Robotics) um projeto desenvolvido por eles de um robô montado em uma cadeira de rodas. A solução é focada na independência do usuário, sem necessidade de cuidador ou intermediário. Com esse intuito, tanto a cadeira, quanto o braço robótico, são controlados pelo usuário. Foram estabelecidos requisitos indispensáveis para o projeto: alcançar o chão e o nível da cabeça, não pode comprometer a “manobrabilidade”, acesso, conforto ou controle da cadeira de rodas, nem obstruir a visão do usuário, ou prejudicar esteticamente o dispositivo. Além disso, as tarefas a serem realizadas foram classificadas em 5 categorias: comer e beber, lazer, higiene pessoal, trabalho e mobilidade (Hillman, Hagan, Hagan, Jepson, & Orpwood, 1999).

A interface com o usuário é feita por controle direto, que permite a atuação em qualquer ambiente, embora possa ser trabalhoso para o usuário. Os pesquisadores pré-definiram algumas atividades e comandos, para facilitar para o usuário. O manipulador é controlado por um *joystick*, com alternância de modos, para atender ao controle dos 6 graus de liberdade do braço robótico. Essa alternância de modos é feita pela navegação por opções em uma tela LCD, como mostrado na Figura 20.

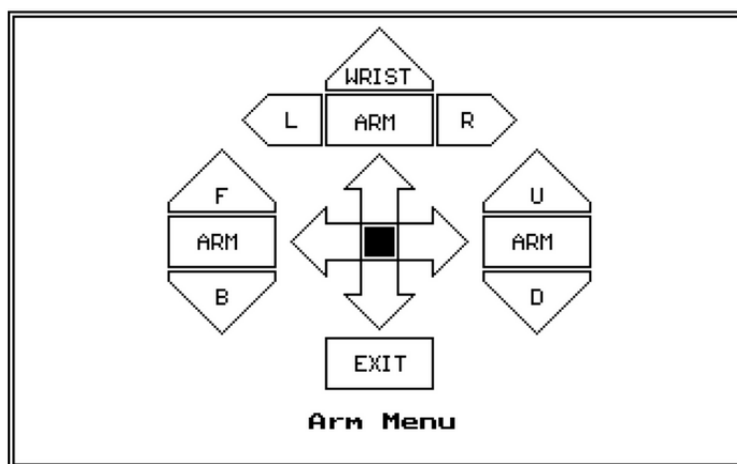


Figura 20 – Tela da interface com o usuário (Hillman, Hagan, Hagan, Jepson, & Orpwood, 1999)

## 2.4 SISTEMAS DE CONTROLE E PROGRAMAÇÃO ROBÓTICA

O controle de um robô é feito por meio da combinação de dois sistemas: o software, que é a parte computacional propriamente dita, e o hardware, que é a parte física do robô, como motores, cabos, sensores e amplificadores de potência. Em conjunto, esses dois sistemas processam sinais de entrada (*input*) e os converte em uma ação programada (*output*) (Carrara, 2015).



## 2.4.1 SISTEMA DE CONTROLE

O termo sistema é utilizado para descrever uma série de componentes que interagem em torno de uma condição limite imaginária, e somente a entrada e a saída são de interesse. Ou seja, um sistema pode ser visto como uma caixa preta com uma entrada e uma saída, na qual não se sabe o que há em seu interior, apenas a relação entre a entrada e a saída. Mesmo podendo ser complexos os elementos que compõem o sistema e as relações entre eles, pode-se apenas considerar que todos esses componentes fazem parte de um bloco, a tal caixa preta (Bolton, 1995). A Figura 21 mostra simplificada a representação de um sistema (bloco) e sua entrada e saída (setas):

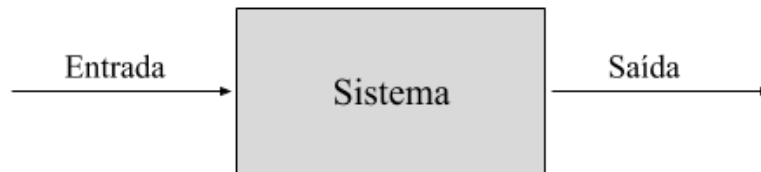


Figura 21 – Representação simplificada de um sistema

Quando conveniente e necessário, pode-se dividir o sistema em vários subsistemas acoplados (Bolton, 1995). Neste caso, a representação pode se manter como a da Figura 22, ou como na Figura 23:

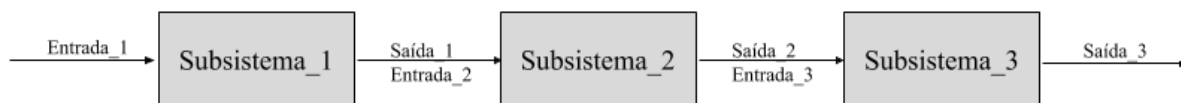


Figura 22 – Representação dos subsistemas de um sistema

Em um sistema de controle, por sua vez, a saída é controlada para assumir um valor particular ou variar de uma determinada forma pela entrada do sistema. Um sistema de controle recebe uma informação do ambiente, a temperatura atual ou a desejada por exemplo, e atua sobre a entrada a fim de gerar uma saída da forma esperada. Usando o mesmo tipo de diagrama anterior, um sistema de controle pode ser representado por:

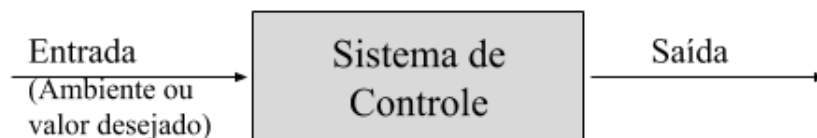


Figura 23 – Representação de um sistema de controle

O sistema de controle do robô pode funcionar de duas maneiras: em malha aberta ou malha fechada. O primeiro modelo não faz nenhum tipo de validação do output adquirido, a entrada é escolhida com base na experiência, de tal forma que o sistema dê o valor de saída desejado. No entanto, essa saída não é modificada para compensar ou acompanhar alterações nas condições de operação. Por exemplo, um

ar-condicionado onde apenas se seleciona a potência de trabalho, mas que não trabalhará “mais”, ou “menos”, caso uma janela ou porta do ambiente esteja aberta. Esse exemplo representa um sistema de controle em malha aberta sem realimentação deste para manutenção ou ajuste da variável de saída (temperatura) constante (Bolton, 1995).

Já no modelo de controle em malha fechada, um sinal da saída realimenta a entrada do sistema e é usado para modificá-la, mantendo a saída constante, mesmo com alterações nas condições de operação. No caso do ar-condicionado citado anteriormente, ele deveria verificar a temperatura ambiente, comparar com a desejada, e chavear a potência do equipamento a fim de manter constante a temperatura do ambiente (Bolton, 1995). Ou seja, o controle do sistema verifica se o resultado adquirido é compatível com o desejado (valor de referência), realimentando o sistema com a diferença encontrada (desejado v.s. real) e reajustando o output de uma maneira iterativa, até a diferença estar dentro de um valor aceitável.

Do ponto de vista constitutivo, um sistema em malha aberta é composto por subsistemas, tal quais: elemento de controle, que determina a ação que deve ser tomada, tendo em vista a entrada do sistema de controle; elemento de correção, que responde ao sinal de saída do elemento de controle e age de forma a levar a variável controlada ao valor desejado; e processo, também chamado de planta, que é o sistema no qual uma variável é controlada (Figura 24). Os dois primeiros fazem parte do que se entende por “controlador” (Bolton, 1995).

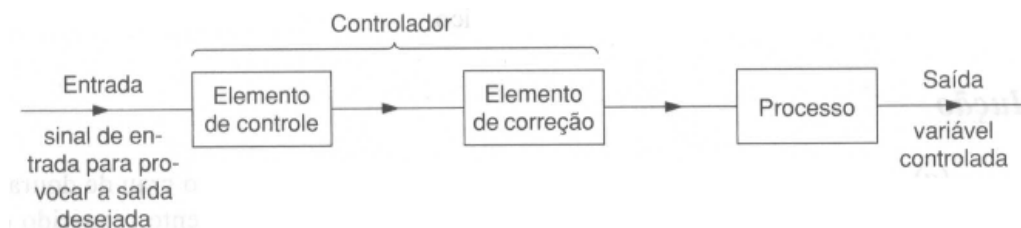


Figura 24 – Representação dos elementos básicos de um sistema em malha aberta (Bolton, 1995)

Sistemas de controle em malha aberta utilizam, geralmente, um elemento de controle que envia um sinal, ou sequência de sinais, para realizar uma ação, ou sequência delas, após um ou mais intervalos de tempo. Este tipo de controle é encontrado em máquinas de lavar, por exemplo.

Um sistema em malha fechada, por sua vez, é composto por: um elemento de comparação, que confronta o valor real (medido) com o desejado e determina o sinal de erro; elemento de controle, que decide qual ação tomar com base no sinal de erro recebido; elemento de correção, também chamado de atuador, que provoca mudança no processo a fim de corrigir o erro; um processo, que é o sistema no qual a variável é controlada; e um elemento de medida, que gera um sinal relacionado com a condição da variável que está sendo controlada, fornecendo o sinal realimentado para o elemento de comparação (Bolton, 1995). Esses itens que compõem um sistema de malha fechada são representados na Figura 25.

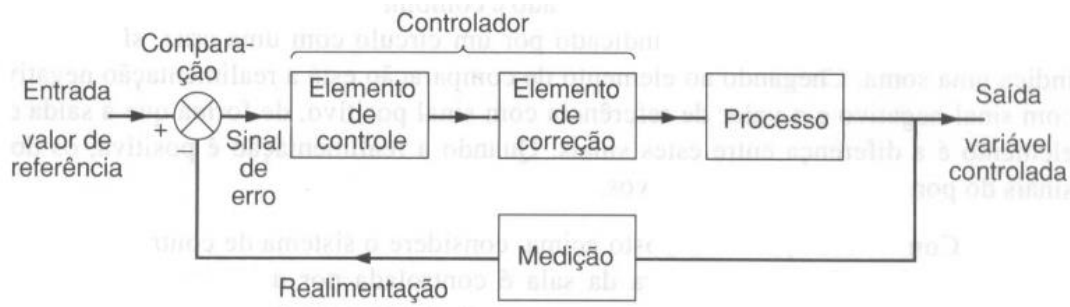


Figura 25 – Representação dos elementos básicos de um sistema em malha fechada (Bolton, 1995)

O que diferencia um sistema de malha fechada para um de malha aberta é que aquele possui uma etapa de realimentação e de comparação. Essa realimentação é característica necessária para um sistema de malha fechada, podendo ser negativa, quando o sinal realimentado é subtraído do sinal de referência, ou seja, o realimentado é maior que o desejado; ou positiva, quando o sinal realimentado é somado ao de referência, no caso dele ser menor que o de referência (Bolton, 1995).

Embora os sistemas de malha fechada sejam relativamente mais simples de se fazer e possuam baixo custo, apresentam uma frequente imprecisão, em decorrência da ausência de correção para os erros e alterações de operação. Por outro lado, os sistemas de malha fechada possuem maior precisão e menor sensibilidade a distúrbios e variações nas características dos componentes, mas são mais complexos, mais caros e mais propícios a danos, por possuírem muitos componentes (ESSEL Engenharia, 2019).

No cotidiano, objeto de estudo também deste trabalho, há vários sistemas de controle. Por exemplo, o “agarrar um objeto em cima de uma mesa” envolve diversas etapas de decisão, análise e até de realimentações, como saber se já tocou o objeto, se está agarrando-o muito forte, se ele demanda mais força que o previsto, se ele está escorregando, dentre outros. Ou seja, mais de uma variável deve ser observada e realimentar o sistema. E essas são etapas a serem analisadas neste trabalho, a fim de controlar o manipulador proposto anteriormente (Néto, 2019).

## 2.4.2 PROGRAMAÇÃO ROBÓTICA

O fato de um manipulador robótico ser programável é de fundamental relevância para este trabalho, que visa atender a necessidades cotidianas de pessoas com dificuldades motoras. Cabe então entender as nuances da programação, a começar pela distinção entre programação do proprietário, sistema operacional, feito pelo fabricante; e programação usuário, que é o executado para cumprimento da tarefa desejada por ele. O presente trabalho focará na segunda.

Com a noção adquirida de sistemas de controle, é imperativo entender como comunicar à máquina as variáveis e valores desejados para seu funcionamento adequado. De maneira geral, há duas maneiras de se comunicar com o robô: *on-line* e *off-line* (Programação de Robôs, s.d.).

A programação online é um processo interativo homem-máquina, que ocorre no ambiente de atividade do manipulador. Nela o operador posiciona o efetuador nas posições desejadas e envia as

informações de coordenadas do efetuador e das juntas para seu sistema, a fim de reproduzir a trajetória futuramente. Contudo, ela apresenta a desvantagem, nos casos industriais, de necessitar a parada da linha de produção para “escrever ou reescrever o programa”.

A programação *off-line* não necessita, por sua vez, parar a linha de produção, empregando linguagens textuais de programação, que serão enviadas para o robô e, assim, executá-las. Pela ausência de interface ao vivo com o ambiente de trabalho, esse tipo de programação demanda um levantamento topográfico e modelagem virtual do meio, criando um simulador 3D do espaço, usado para simular, validar, testar e escrever o programa de controle do manipulador.

Há também outras maneiras de se classificar o método de programação de um robô:

- Programação por aprendizagem (guiada): Esse método é o mais comum e utilizado na programação de robôs. Semelhante à programação *online* citada acima, essa técnica consiste em instruir (ensinar) o braço robótico, o guiando através dos pontos da trajetória desejada. O posicionamento do efetuador, por sua vez, pode ser feito de maneira mecanizada, onde o operador ou usuário utiliza uma eletrônica dotada de joystick e botões, chamada de *teaching-box* ou *teach-in-box* (Figura 26), que pode controlar apenas o efetuador, ou as juntas separadamente, alternando esses modos por meio de chaves ou botões. Outra forma de posicionar o efetuador é manualmente pelo usuário, movendo fisicamente o robô, restrita a manipuladores com estrutura leve e acionamentos reversíveis. Em ambas as formas de posicionamento, o controlador armazena na memória os ângulos e deslocamentos das juntas durante o processo (Groover, 1989).



Figura 26 – Teaching-box robô ABB TRB 6400

- Linguagem Textual: Todos os sistemas de controle de manipuladores robóticos contam com recursos de programação em linguagem textual (Carrara, 2015). As linguagens de programação utilizadas variam de fabricante para fabricante, mas, via de regra, é do tipo

sequencial. Nesse método os pontos da trajetória são representados por símbolos, para os quais o manipulador deve se mover. Embora a maioria dos fabricantes não permite acesso à sua programação textual, diversos deles utilizam a VAL – *Variable Assembly Language*, ou *Linguagem de Montagem Variável* (Programação de Robôs Industriais, 2003).

- Programação de célula de trabalho: A mais complexa das citadas até então, é uma programação de controle inteligente, capaz de tomar decisões com base nas informações coletadas pelos sensores externos. Dessa forma, pode haver a interação de programas distintos e de dados provenientes de fontes diversas para essa tomada de decisões (Carrara, 2015).

### 2.3 VOLUME DE CONTROLE

Antes de se definir a trajetória e cinemáticas de um manipulador, é preponderante entender dois aspectos: o volume de trabalho do manipulador, e, complementarmente, quais os pontos inalcançáveis durante a sua movimentação. Volume de trabalho é o conjunto de todos os pontos que o efetuador pode ocupar quando as variáveis de junta são variadas em todos os seus intervalos de definição (Gupta & Roth, 1982). Já os pontos inalcançáveis e os movimentos não executáveis de um manipulador são considerados singularidades. Essas singularidades podem ser de contorno ou internas (Figura 27), a depender de qual região do volume de trabalho se encontram, e devem ser previstas, pois interferem tanto na solução da cinemática inversa, quanto no planejamento da trajetória do efetuador (Erthal, 1992).

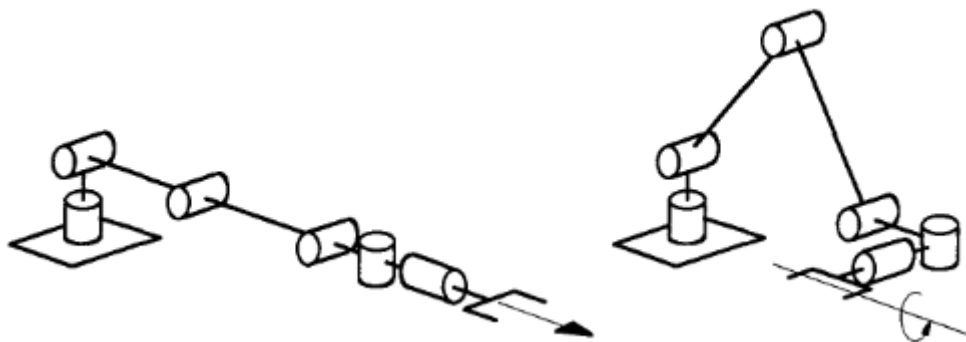
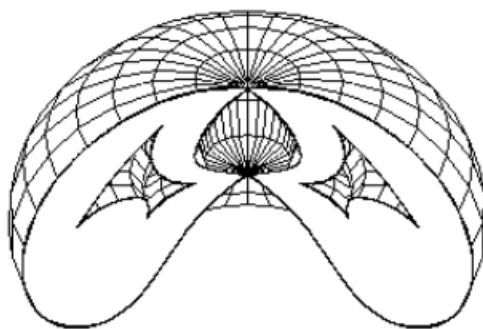


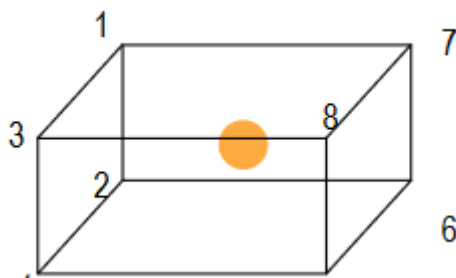
Figura 27 - Singularidades de contorno (esquerda) e interna (direita) (Erthal, 1992)

A Figura 28 representa o volume, ou espaço, de trabalho de um robô hipotético com 3 juntas rotacionais:



*Figura 28 – Volume de trabalho. Adaptado (Hongyu & Saramago, 2008)*

O procedimento mais imediato para se encontrar o volume de trabalho de um manipulador é variando os valores de suas juntas (prismáticas e rotacionais) sobre seus intervalos de definição e estimar as coordenadas da ponta do efetuador, em relação à base do manipulador (Hongyu & Saramago, 2008). Metodologias analíticas para calcular o volume de trabalho de trabalho já foram desenvolvidas, mas em sua maioria para manipuladores com apenas 3 juntas rotacionais (Oliveira, Saramago, & Nogueira, 2010). No entanto, é possível prever (Costa, 2003) o número de pontos alcançáveis por um manipulador que rodeiam um ponto qualquer que se deseja atingir, sendo este no máximo  $2^N$ , onde N é o número total de juntas o manipulador (Figura 29). Para um robô com 3 graus de liberdade, o ponto que se deseja atingir pode estar contido dentro de um polígono de  $2^3 = 8$  vértices.



*Figura 29 – Poliedro definido pelos 8 pontos alcançáveis, em torno de um ponto desejado, de um manipulador com 3 graus de liberdade (Costa, 2003)*

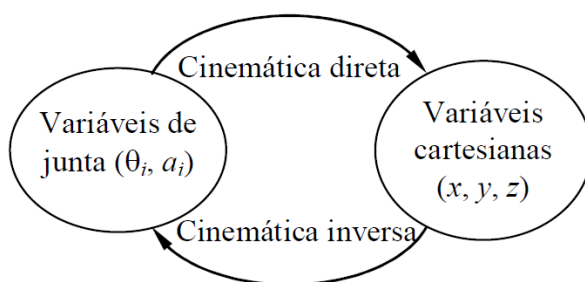
No manipulador hipotético da Figura 28, é possível visualizar um volume interno não pertencendo ao volume de trabalho do robô. Todos os pontos que compõem tal vazio fazem parte de uma região que o manipulador não alcança. No caso de manipuladores robóticos montados em cadeiras de rodas, é indispensável pontuar que, além do conjunto de pontos não acessíveis, pelo volume de trabalho do efetuador, há também pontos inalcançáveis por conta do espaço ocupado pela cadeira de rodas e pelo próprio usuário.

Para ir de um ponto a outro do espaço, é requerido do efetuador que ele percorra uma determinada trajetória dentro do volume de trabalho. Em alguns casos, pode-se solicitar que o manipulador percorra o caminho mais rápido, ou o mais curto, ou que mantenha a garra com alguma orientação específica,

por exemplo. Computacionalmente, o percurso que o efetuador irá percorrer pode ser descrito como uma série de pontos intermediários entre o inicial e o final (Costa, 2003).

## 2.4 CINEMÁTICA DE MANIPULADORES

Há dois tipos de cinemáticas de robôs manipuladores: direta e inversa. Esta diz respeito de encontrar os valores das variáveis de juntas em função da posição e velocidade do efetuador terminal, enquanto aquela trata de encontrar parâmetros de posição e velocidade do efetuador em função dos valores das juntas do robô (Figura 30). Para o problema de controle de um robô é necessário encontrar a cinemática inversa do manipulador (Carrara, 2015).



*Figura 30–Transformações entre variáveis de junta e cartesianas (Carrara, 2015)*

A cinemática direta sempre possui uma solução e esta é única. A cinemática inversa, por sua vez, lida com sistemas de equações não-lineares, tornando sua solução mais complexa. Pode haver casos onde não há uma solução para a cinemática inversa, como em caso de singularidades do volume de trabalho, seja por restrições mecânicas ou geométricas. Assim como também é possível que haja infinitas soluções, o que ocorre quando o robô é redundante e possui “mais graus de liberdade que o necessário” para realizar determinado movimento ou atingir determinada posição (Erthal, 1992).

O problema de cinemática inversa pode ser resolvido então de diversas formas. As duas mais utilizadas são a analítica, separadas em algébrica, que manipula as equações de cinemática inversa isolando as variáveis de junta; e a geométrica que, como o próprio nome já diz, utiliza a geometria do manipulador para obter as equações para as variáveis de junta. Contudo, em alguns casos, a cinemática inversa do manipulador não pode ser resolvida por nenhum desses métodos, e se recorre a métodos numéricos de solução.

Embora haja metodologias diferentes para obtenção da cinemática inversa de um manipulador, a escolha da alternativa mais adequada para determinado problema deve levar em consideração alguns fatores (Erthal, 1992), como:

- Velocidade de obtenção das soluções;
- Recursos disponíveis para cálculo das soluções;

- Generalidade;
- Estabilidade; e
- Obtenção das soluções.

É importante levar em consideração também a usabilidade e contexto do manipulador, por quem, para quê, e se é em tempo real ou não (telecontrole). Uma vez que o problema abordado por este trabalho possui várias peculiaridades, premissas e restrições características de sua funcionalidade, estas devem ser consideradas na escolha da melhor (mais adequada) metodologia de solução da cinemática inversa do manipulador.

### 2.4.1 MÉTODO ALGÉBRICO

Embora este método possa ser desenvolvido de diversas formas, o proposto em (Paul, 1979) é bastante conhecido e consiste em multiplicar a equação da matriz de transformações homogênea (Equação 7) à esquerda pelas inversas de cada uma das matrizes de transformação intermediárias. Com isso, busca-se obter uma matriz que, a partir de um vetor posição, retorne o valor das variáveis de junta do robô (Erthal, 1992):

$${}^5A_6 {}^4A_5 {}^3A_4 {}^2A_3 {}^1A_2 {}^0A_1 AT_6 = {}^5A_6 {}^4A_5 {}^3A_4 {}^2A_3 {}^1A_2 {}^0A_1 A_1^0 A_1^1 A_2^2 A_3^3 A_4^4 A_5^5 A_6^6 A \approx {}^0A_6 \quad (9)$$

### 2.4.2 MÉTODO GEOMÉTRICO

No caso de um manipulador RRR (Figura 31), se movimentando em um plano vertical com dimensões dos elos conhecidas ( $a_1$ ,  $a_2$  e  $a_3$ ), o problema da cinemática inversa só é possível de ser resolvido com a pré-definição de uma orientação do último elo com relação à horizontal por um ângulo  $\varphi$  conhecido (Carrara, 2015).

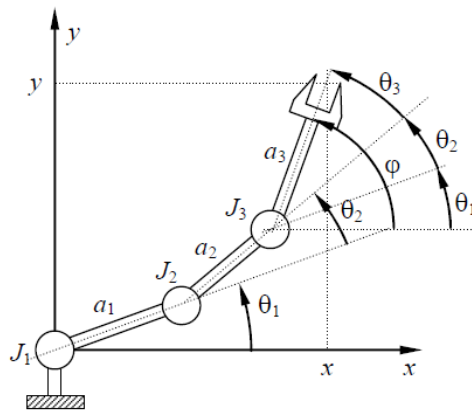


Figura 31– Manipular RRR se movimentando em um plano vertical xy (Carrara, 2015)

A cinemática direta desse manipulador é obtida projetando as juntas nos eixos cartesianos, fornecendo:

$$x = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (10)$$



$$y = a_1 \text{sen} \theta_1 + a_2 \text{sen}(\theta_1 + \theta_2) + a_3 \text{sen}(\theta_1 + \theta_2 + \theta_3) \quad (11)$$

Algebricamente, a cinemática direta não pode ser calculada, pois há mais incógnitas (os ângulos das juntas) que equações. Essa impossibilidade também é visualizada pela diversas possibilidades (Figura 32) que o manipulador pode ser posicionado para atingir com seu efetuador um mesmo ponto. Assim, como mencionado anteriormente, é necessário inserir uma condição conhecida ao problema, o ângulo  $\varphi$  (Carrara, 2015).

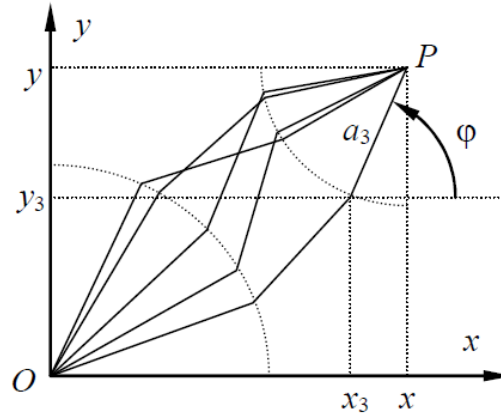


Figura 32– Algumas das infinitas possibilidades de configuração do manipulador para alcançar um mesmo ponto P (Carrara, 2015)

Essa nova condição fornece um novo ponto conhecido  $(x_3; y_3)$ . Em função das dimensões dos elos  $a_1$  e  $a_2$ , do ângulo  $\varphi$  e das coordenadas do ponto P original, as coordenadas do novo ponto são dadas por:

$$x_3 = x - a_3 \cos \varphi \quad (12)$$

$$y_3 = y - a_3 \text{sen} \varphi \quad (13)$$

A determinação de  $\varphi$  reduzirá também o problema de cinemática inversa a um sistema com duas equações e duas incógnitas, onde se busca, inicialmente, esta cinemática para o ponto  $(x_3; y_3)$ . Os ângulos das juntas  $J_1$  e  $J_2$  são obtidos, então, por:

$$\theta_2 = \pm \arccos \left[ \frac{(x - a_3 \cos \varphi)^2 + (y - a_3 \text{sen} \varphi)^2 - a_1^2 - a_2^2}{2a_1 a_2} \right] \quad (14)$$

$$\theta_1 = \arctan \left[ \frac{(y - a_3 \text{sen} \varphi)(a_1 + a_2 \cos \theta_2) - (x - a_3 \cos \varphi)a_2 \text{sen} \theta_2}{(x - a_3 \cos \varphi)(a_1 + a_2 \cos \theta_2) + (y - a_3 \text{sen} \varphi)a_2 \text{sen} \theta_2} \right] \quad (15)$$

Para finalizar, como se conhece o valor de  $\varphi$  e se calculou os de  $\theta_1$  e  $\theta_2$ , o último ângulo desconhecido é obtido por:

$$\theta_3 = \varphi - \theta_1 - \theta_2 \quad (16)$$

No caso de um manipulador que se movimenta no espaço, por exemplo com configuração TRR articulada, o problema da cinemática inversa deve ser resolvido considerando que o movimento do robô por ser decomposto em um movimento plano e um movimento rotativo ao redor de um eixo contido no

plano, fazendo com que o primeiro plano gire no espaço. A Figura 33 representa um robô com três graus de liberdade, com a junta  $J_1$  gira sob um eixo vertical, enquanto as  $J_2$  e  $J_3$  possuem eixos horizontais (Carrara, 2015).

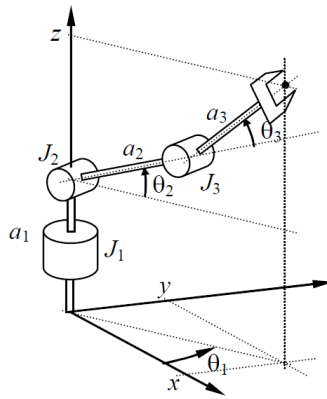


Figura 33 – Braço TRR em movimento no espaço. Adaptado (Carrara, 2015)

Para visualização e compreensão dos movimentos do robô e posições das juntas, duas vistas são utilizadas (Figura 34): superior e lateral, a primeira representando o movimento horizontal o manipulador, e a segunda o movimento vertical, no plano.

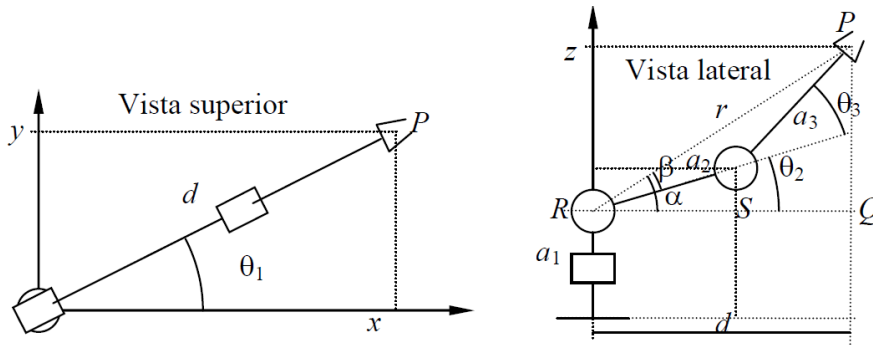


Figura 34– Vistas dos movimentos do manipulador TRR. Adaptado (Carrara, 2015)

Sobre o plano superior  $xy$ , a projeção do ponto  $P$  (central do efetuador terminal) é fornecida pela distância  $d$ , obtida pelas projeções dos elos  $a_2$  e  $a_3$  no plano, dadas por:

$$d = a_2 \cos \theta_2 + a_3 \cos (\theta_2 + \theta_3) \quad (17)$$

Da vista superior, a distância  $d$  é dada pela hipotenusa de um triângulo retângulo formado pelas componentes  $xy$  do ponto  $P$ . A coordenada  $z$  é calculada de maneira semelhante. Ou seja, é necessário calcular a cinemática direta antes:

$$x = d \cos \theta_1 = [a_2 \cos \theta_2 + a_3 \cos (\theta_2 + \theta_3)] \cos \theta_1 \quad (18)$$

$$y = d \sin \theta_1 = [a_2 \cos \theta_2 + a_3 \cos (\theta_2 + \theta_3)] \sin \theta_1 \quad (19)$$

$$z = a_1 + a_2 \sin \theta_2 + a_3 \sin (\theta_2 + \theta_3) \quad (20)$$

Assim, a distância  $d$ , da vista superior, é dado por:

$$d = \sqrt{x^2 + y^2} \quad (21)$$

Na vista lateral, a distância  $r$  corresponde à distância entre o centro da junta  $J_2$  ao ponto  $P$ , e pode ser calculada pela hipotenusa do triângulo retângulo  $PQR$ :

$$r^2 = d^2 + (z - a_1)^2 = x^2 + y^2 + (z - a_1)^2 \quad (22)$$

Aplicando a Lei dos Cossenos ao triângulo  $PRS$ , sendo  $S$  o centro da junta  $J_2$ , tem-se que:

$$\theta_3 = \pm \arccos \left[ \frac{x^2 + y^2 + (z - a_1)^2 - a_2^2 - a_3^2}{2a_2a_3} \right] \quad (23)$$

A equação acima indica que a cinemática inversa do manipulador TRR possui mais de uma solução para o ângulo da junta  $J_2$ . Isso é observado pois o cotovelo, evidente na vista lateral pelo ponto  $S$ , pode assumir duas posições: cotovelo para cima e para baixo, o que fornece dois valores para cada um dos ângulos  $\theta_2$  e  $\theta_3$ , sendo estes dependentes, também, um do outro (Carrara, 2015).

Já a variável de junta  $\theta_2$  pode ser obtida pela diferença entre os ângulos  $\alpha$  e  $\beta$ , a partir das tangentes destes:

$$\tan \alpha = \frac{z - a_1}{d} = \frac{z - a_1}{\sqrt{x^2 + y^2}} \quad (24)$$

$$\tan \beta = \frac{a_3 \sin \theta_3}{a_2 + a_3 \cos \theta_3} \quad (25)$$

Assim,

$$\theta_2 = \arctan \left[ \frac{(z - a_1)(a_2 + a_3 \cos \theta_3) - \sqrt{x^2 + y^2} a_3 \sin \theta_3}{\sqrt{x^2 + y^2}(a_2 + a_3 \cos \theta_3) + (z - a_1)a_3 \sin \theta_3} \right] \quad (26)$$

Por fim, da vista superior, é possível calcular de maneira simples o ângulo  $\theta_1$  por uma equação de arco-tangente simples, a partir das coordenadas  $xy$  do efetuador:

$$\theta_1 = \arctan \frac{x}{y} \quad (27)$$

A função do arco-tangente é computacionalmente limitada, retornando apenas valores de ângulos entre  $-\pi/2$  e  $\pi/2$ . No contexto de um manipulador robótico isso é um fator preocupante, pois as juntas devem assumir valores que extrapolam esses limites para alcançar determinados pontos e executar certos comandos. Para corrigir isso, é necessário estudar o denominador da função, o seno:

$$\theta = \arctan \frac{\text{seno}}{\text{cosseno}} \quad (28)$$

Quando o seno for positivo, o computador retornará um valor de  $\theta$  que estará corretamente dentro do limite de  $-\pi/2$  a  $\pi/2$ . Quando o seno for negativo, será necessário somar  $-\pi$  ou  $\pi$  ao resultado.

### 2.4.3 MÉTODOS NUMÉRICOS

A maioria dos métodos numéricos para solução do problema de cinemática inversa de um manipulador utilizam valores numéricos para as variáveis, ou seja, calculam iterativamente as funções relevantes para cada novo ponto onde se encontra o efetuador.

Kazem Kazerounian, professor do departamento de Engenharia Mecânica da Universidade de Connecticut, propõe o método da análise da posição-zero (Erthal, 1992), baseado um processo de modificação, individual e consecutiva, das variáveis de juntas, buscando aproximar a nova posição do efetuador daquela desejada. É um método que utiliza modelagem vetorial de posições e, mesmo sendo uma solução mais lenta em relação a outras propostas, é uma metodologia estável, mesmo próximo de regiões singulares do volume de trabalho.

O método de Newton-Raphson, se baseia na expansão linear da função erro, buscando minimizá-lo. Pela alta demanda computacional e limitação ferramental e construtiva dos manipuladores, não se assume que a função erro seja nula. Então, define-se um valor aceitável para o erro de posicionamento do manipulador e busca-se aproximar a função erro deste valor. A partir do controle desse erro há a concordância e convergência das soluções calculadas para um valor aceitável de cinemática inversa. A função erro é definida pela diferença entre o vetor da posição desejada e a posição fornecida pela cinemática direta do robô (Erthal, 1992):

$$F(q) = X_e - f(q) \quad (29)$$

É feita então uma expansão em série de Taylor da Equação 29 e se busca controlar a expressão entre colchetes:

$$q^{K+1} = q^K - J^{-1}(q^K) \cdot [X_e - f(q^K)] \quad (30)$$

Outro método comumente utilizado é o da integração das velocidades, que soluciona a cinemática inversa a partir das velocidades especificadas para o efetuador (Erthal, 1992). Como será visto mais adiante, utiliza o conceito do jacobiano inverso de velocidades para retornar as velocidades das juntas e, assim, estimar a posição inicial das juntas antes da movimentação. Dentro deste método podem ser utilizados outros para integração das velocidades.

### 2.4.4 MÉTODOS MISTOS

Há ainda os métodos que mesclam soluções analíticas (algébrica ou geométrica) e soluções numéricas.

Uma proposta desse gênero que pode ser relevante para a solução do problema proposta para este trabalho é a sugerida por Vladimir Lumelsky, pois trata de uma alternativa para manipuladores não-esféricos, ou seja, os 3 GDL de orientação não se interceptam em um mesmo ponto (Erthal, 1992). Nesta metodologia se resolve analiticamente a cinemática inversa para os graus de liberdade de posicionamento e orientação separadamente e, com o auxílio numérico, itera o ponto final dos GDL de

posicionamento com o ponto inicial (“base”) dos GDL de orientação, até uma aproximação aceitável. No entanto, o controle de erros deste método não é muito afinado em regiões próximas a singularidades do manipulador.

## 2.5 PLANEJAMENTO DA TRAJETÓRIA

O controle do movimento de um robô pode ser dividido em duas etapas: o planejamento do movimento e o acompanhamento do sinal. A primeira corresponde à geração do trajeto e sua lei temporal, e a segunda, como o próprio nome diz, é o acompanhamento do sinal de referência (Fanti, Ferreira, & Carvalho, 2013).

O planejamento e controle da trajetória de um manipulador é uma ferramenta importante tanto na programação, quanto na operação de robôs. Em muitos casos não basta atender aos requisitos de posição final e inicial do efetuador ou das juntas do manipulador, mas também garantir uma rota específica, a manutenção da orientação do efetuador terminal, ou até mesmo respeitar restrições de aceleração, posição e velocidade dos elos e atuadores de um robô. Em uma ótica econômica, a atenção com esses fatores resulta em menos desperdício, menos tempo, menos desgaste, menos acidentes, mais segurança, mais conforto e mais lucro.

Computacionalmente todo esse planejamento e acompanhamento é realizado pela programação do controle. Em um manipulador, essa programação pode ser realizada basicamente de três maneiras: por sequência fixa, por repetição, e com realimentação externa (Carrara, 2015).

- Por sequência fixa: é empregada geralmente em mecanismos projetados para realizar uma única tarefa, ou tarefas semelhantes entre si, e, por isso, apresentam o mais baixo nível de controle, entre as possibilidades citadas. Por exemplo, separadores automáticos de produtos em esteiras, mecanismos do tipo *pick-and-place* (Figura 35) e dispositivos de alimentação de material para corte ou prensagem.



Figura 35 – Braço robótico do tipo *pick-and-place* aplicado à indústria alimentícia (Knight, 2018)

- Por repetição: este tipo de programação é empregado em manipuladores programáveis, com um nível maior de controle, e com foco em atividades cíclicas. No entanto, esse tipo de programação é possível de ser feita por trajetória ponto-a-ponto (PTP) ou por trajetória contínua (CP). Tais metodologias de controle da trajetória serão explicadas adiante. Esse tipo de programação é usado em robôs de solda (Figura 36), por exemplo.



*Figura 36 – Robô OEM de Solda MIG HA1440 (Ali Express, s.d.)*

- Com realimentação externa: é empregado em linhas de trabalho/processos produtivos que envolvem várias máquinas, onde a troca de informação entre os processos e dispositivos é necessária e importante para o desempenho com excelência. Essa troca possibilita a adaptação e tomada de decisões quanto a, por exemplo, finalização de um processo, posição de uma peça, mudança de ciclo de trabalho, dentre outros. Esse tipo de controle é encontrado em operações de usinagem, onde o braço se comunica com a máquina CNC para informar o posicionamento da peça a ser usinada. No laboratório do GRACO (Grupo de Automação e Controle da Universidade de Brasília), há um robô que possui esse tipo de controle, no qual o braço deve se comunicar com a mesa que apoia a peça (Figura 37).



Figura 37– Robô do GRACO, UnB

### 2.5.1 VELOCIDADES

A coordenação das velocidades das juntas de um manipulador define tanto a velocidade de seu efetuador, quanto a trajetória que ele irá percorrer. O controle dessas velocidades pode ser tanto feito de maneira “grosseira”, como acionando cada junta ao mesmo tempo, uma após a outra ou até mesmo de forma que comecem e encerrem seu acionamento simultaneamente; quanto de maneira mais refinada, sincronizando as duas velocidades a fim de se percorrer uma trajetória definida por uma curva específica (Carrara, 2015).

Um jeito comum de planejamento de uma trajetória é através de sua definição ponto-a-ponto (*PTP – Point to Point*), na qual todo o caminho que se deseja trilhar com o efetuador é interpolado, definindo diversos pontos intermediários que, quando percorridos, descrevem um caminho próximo aquele desejado. Entre esses pontos, todos os atuadores podem ser acionados simultaneamente com máxima velocidade, ou com velocidades ajustáveis para sincronismo (Carrara, 2015). Essas duas possibilidades resultam em duas trajetórias diferentes para um manipulador RR planar, por exemplo, como representado na Figura 38:

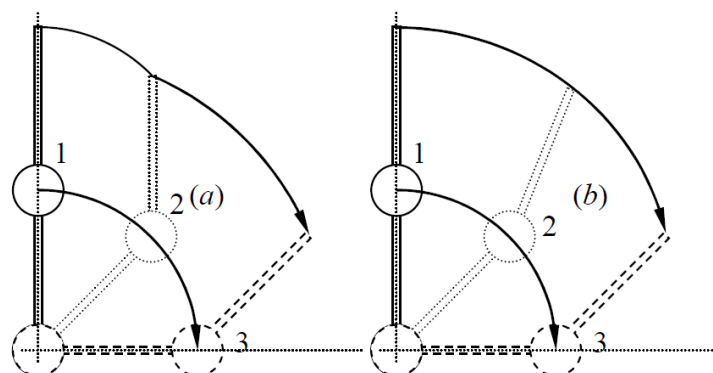


Figura 38 – Trajetória de um efetuador com velocidades máximas das juntas (esquerda) e velocidades sincronizadas (direita) (Carrara, 2015)

O planejamento ponto-a-ponto utiliza um princípio comparativo entre a posição inicial das juntas e a posição desejada destas, a fim de se percorrer todos os pontos intermediários entre o inicial e o final. No âmbito computacional, utilizar essa metodologia de planejamento demanda um processo iterativo para comparar as posições e, ainda, um gatilho para determinar quando esse processo iterativo deve cessar. Tais gatilhos podem ser por um erro máximo permitido (*posição desejada - posição atual*), quando se parte do pressuposto de que não será possível alcançar exatamente a posição desejada, por limitações de instrumentais; um número de iterações, uma solução simples, mas que demanda a obtenção de um “número ótimo” de iterações que atenda adequadamente às diversas possibilidades de trajetória; o limite físico/mecânico da junta; ou até que o rank da matriz jacobiana (explicada a frente) seja menor que a quantidade de linhas desta matriz, conhecido como teste do rank da matriz (Rosário, 2012).

No entanto, a velocidade de processamento e cálculos de velocidade e pontos intermediários demandam grande capacidade de processamento dos softwares e controle utilizados.

Há também o planejamento por trajetória contínua (CP), onde se define o tipo de trajetória entre dois pontos é especificado pelo operador, como retilínea, curva suave ou arco de circunferência, por exemplo. Essas possibilidades de trajetória são representadas na Figura 39. Contudo, diferentemente do moto PTP, o próprio software de controle do robô calcula os pontos intermediários para percorrer o caminho definido (Carrara, 2015).

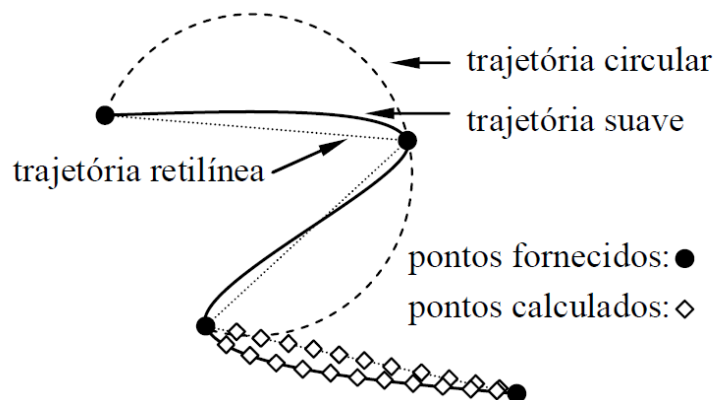


Figura 39 – Trajetórias possíveis de um efetuador controlado por CP (Carrara, 2015).

Essas trajetórias, inclusive, podem ser realizadas mediante interpolação. Contudo, a fim de se evitar que a passagem por todos esses pontos acarrete em acelerações e desacelerações elevadas dos atuadores (Snyder, 1985), comumente utiliza-se pontos de passagem, filtrando a interpolação realizada. A filtragem é feita incluindo pontos na parte inicial e final dos intervalos/trechos da trajetória desejada (Figura 40). Essa prática evita desestabilização do manipulador e, claro, mais conforto e segurança na movimentação e controle dos manipuladores (Rosário, 2012).



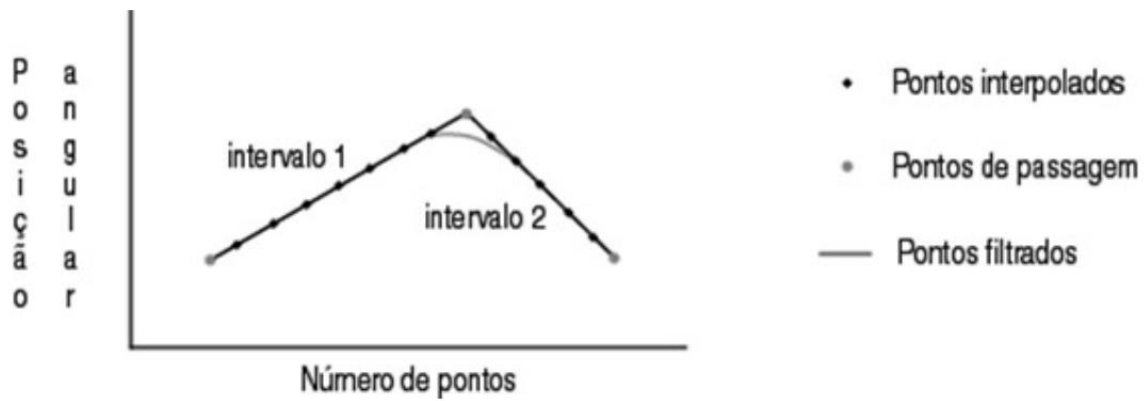


Figura 40 - Interpolação e filtragem de pontos de passagem (Rosário, 2012)

No entanto, assim como a cinemática inversa relaciona a posição do efetuador em termos das posições das variáveis de juntas, é importante compreender como as velocidades dessas entidades se relacionam para o controle do robô e de sua trajetória.

No caso de um manipulador planar RR com dois graus de liberdade, para que seu efetuador descreva uma trajetória retilínea (Figura 41) é necessário que as velocidades e acelerações das juntas sejam observadas e controladas. A Figura 42 mostra como o ângulo das duas juntas varia com o tempo, e a Figura 43 mostra como estas velocidades variam ao longo do tempo (Carrara, 2015).

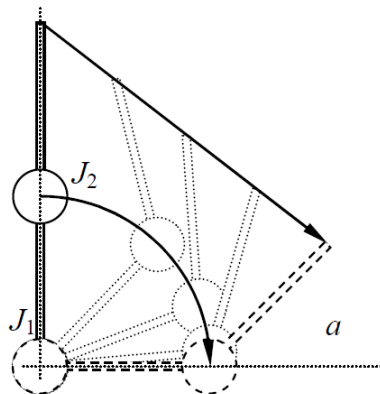


Figura 41– Posições das juntas para a descrição de uma trajetória retilínea por um robô planar (Carrara, 2015)

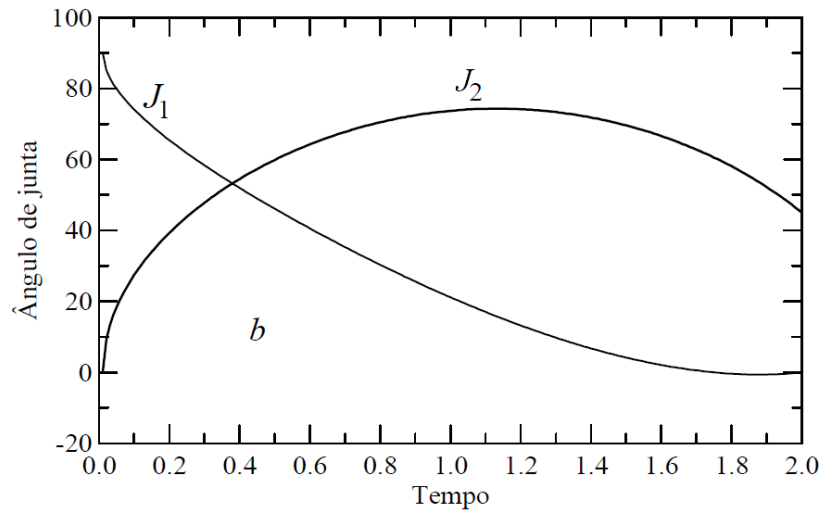


Figura 42– Taxa de variação dos ângulos das juntas em função do tempo (Carrara, 2015)

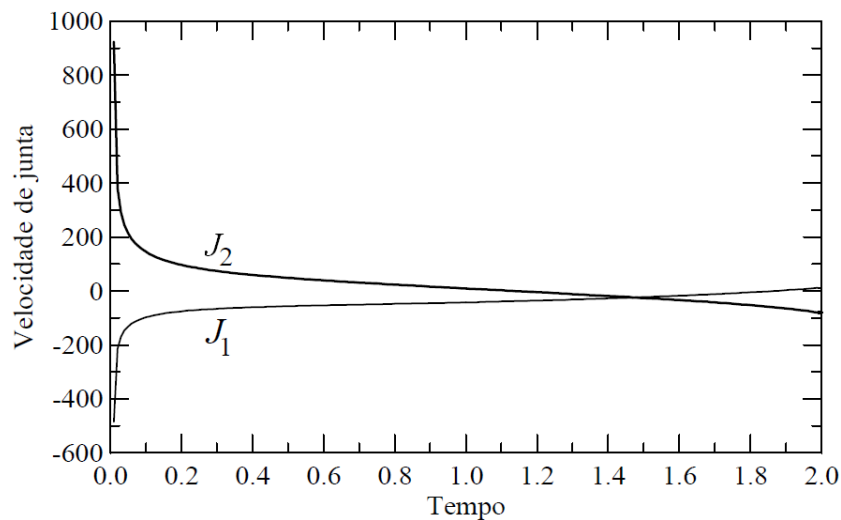


Figura 43– Taxa de variação das velocidades das juntas em função do tempo (Carrara, 2015)

Para realizar esses movimentos, o sistema de controle do robô deve, então, iniciar o movimento acelerando a junta, seguido de um período de manutenção das velocidades e, em seguida, desacelerar as juntas, a fim de estabilizar o efetuador na posição desejada. Essa noção é importante pois a precisão de posicionamento do efetuador depende desse controle minucioso das velocidades das juntas (Carrara, 2015).

A ferramenta matemática que relaciona as velocidades (taxa de variação da posição) das juntas e do efetuador é conhecido como matriz jacobiana de velocidades. Analiticamente, sendo  $q_i$  a variável de junta da junta  $J_i$  de um manipulador e  ${}^i_0x$ ,  ${}^i_0y$  e  ${}^i_0z$  as coordenadas  $x$ ,  $y$  e  $z$  de cada junta  $J_i$ , é possível construir tal matriz (Equação 31), obtida pelas derivadas de cada componente do vetor posição de cada junta, e do efetuador terminal, em relação a cada uma das próprias variáveis de junta (Cabral, 2017):

$$J_v(\mathbf{q}) = \begin{bmatrix} \frac{\partial \dot{x}}{\partial q_1} & \dots & \frac{\partial \dot{x}}{\partial q_i} \\ \frac{\partial \dot{y}}{\partial q_1} & \dots & \frac{\partial \dot{y}}{\partial q_i} \\ \frac{\partial \dot{z}}{\partial q_1} & \dots & \frac{\partial \dot{z}}{\partial q_i} \end{bmatrix} \quad (31)$$

Com essa matriz é possível relacionar as velocidades cartesianas finais de cada junta, ou do efetuador, em função das velocidades das variáveis de junta. Em outras palavras, a matriz jacobiana fornece o quanto cada velocidade de junta contribui para a velocidade do efetuador terminal, ou de uma junta qualquer. Ou seja:

$$\mathbf{v}_i = J_v(\mathbf{q})\dot{\mathbf{q}} \quad (32)$$

Sendo  $\mathbf{v}_i$  o vetor velocidade linear do efetuador, ou junta  $J_i$ , e  $\dot{\mathbf{q}}$  o da velocidade de cada variável de junta. Podendo ser escrito também da seguinte forma, considerando as velocidades angulares do efetuador, ou junta  $J_i$ :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = J_v(\mathbf{q}) \cdot \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_i \end{bmatrix} \quad (33)$$

Para o problema inverso, ou seja, obter as velocidades das variáveis de junta em função de uma velocidade linear do manipulador, é necessário encontrar o jacobiano inverso das velocidades. Nem sempre é uma tarefa simples, pois um jacobiano não quadrado ( $n^\circ$  de linhas ( $m$ )  $\neq$   $n^\circ$  de colunas ( $n$ )) não pode ser invertido pelo método “comum”. Para isso, pode-se recorrer a alguns métodos (Klug, Cinemática Diferencial de Manipuladores Seriais, 2019)

O primeiro deles é a simples inversão da matriz jacobiana, sendo possível apenas quando  $m = n$ :

$$J^{-1} = \frac{Adj(J_v(q))}{|J_v(q)|} \quad (34)$$

Outro método é a diferenciação da cinemática inversa do manipulado (Klug, Cinemática Diferencial de Manipuladores Seriais, 2019), ou seja, derivar as variáveis de junta em relação às variáveis cartesianas de posição. Dessa forma, o jacobiano inverso é obtido por:

$$J^{-1} = \begin{bmatrix} \frac{\partial q_1}{\partial x} & \frac{\partial q_1}{\partial y} & \frac{\partial q_1}{\partial z} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_i}{\partial x} & \frac{\partial q_i}{\partial y} & \frac{\partial q_i}{\partial z} \end{bmatrix} \quad (35)$$

Por fim, há também a possibilidade de se utilizar a definição de pseudo-inversa ( $A^+$ ). Assim como para a cinemática inversa em si, essa metodologia pode ser utilizada de maneira numérica, recalculando o valor desse jacobiano para cada novo ponto ocupado pelo efetuador terminal (Erthal, 1992). Essa

metodologia de solução relaciona matriz original ( $A$ ) e transposta ( $A^T$ ), a depender da comparação entre  $m$  e  $n$  (números de linhas e colunas, respectivamente, da matriz jacobiana):

- Para  $m \leq n$ :  $A^+ = A^T(AA^T)^{-1}$ ;
- Para  $m = n$ :  $A^+ = A^{-1}$ ; e
- Para  $m \geq n$ :  $A^+ = (A^T A)^{-1} A^T$ .

Independentemente do método adotado, a velocidade das juntas, em função da velocidade do efetuador, é dada multiplicando a Equação 32, à direita, pela matriz do jacobiano inverso, resultando em:

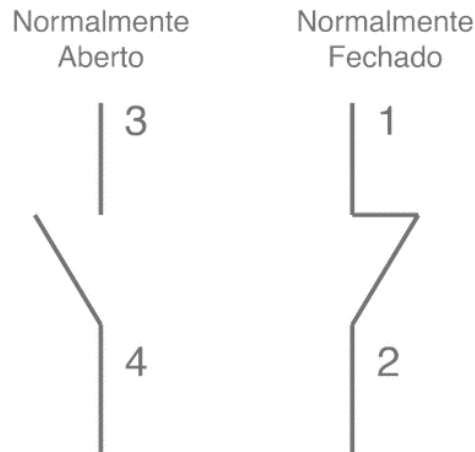
$$\dot{q}_i = J^{-1} v_i \quad (36)$$

No âmbito do controle e planejamento de trajetórias, o jacobiano inverso de velocidades é uma importante ferramenta matemática pois, a partir do vetor de velocidades lineares é possível obter as respectivas velocidades das variáveis de juntas. No que tange à interface com o usuário, o jacobiano facilita o uso do dispositivo de comando pelo paciente, uma vez que este deve apenas se preocupar em como deslocar o efetuador e, a programação por trás do software de controle, faz as contas e combinações necessárias para realizar o comando desejado.

## 2.6 DISPOSITIVOS DE COMANDO

Dispositivos de comando são fundamentais para se fazer a interface do usuário com uma máquina. Essencialmente são responsáveis por acionar (ligar e desligar) um outro dispositivo do sistema, como motores. São exemplos botoeiras ou botões, chaves e interruptores, além de *joysticks*.

As botoeiras são comumente conhecidas como botões de comando, e funcionam como um acionamento manual de motores. Seu princípio de funcionamento é pela comutação de seus contatos: NA (normalmente aberto) e NF (normalmente fechado), o que alterna o estado de “ligado” e “desligado” do motor (Figura 44). Chaves e interruptores têm o mesmo princípio de operação. (Comandos Elétricos: o que são e onde são usados, 2019) Embora seu funcionamento seja bem simplificado, há uma variedade de configurações, formato, cor, tipo de proteção do acionador, além de quantidade e tipos de contatos.



*Figura 44 – Contatos normalmente aberto e fechado (Bluelux, 2018)*

As botoeiras que possuem uma mola, que as faz retornar para a posição original após serem acionadas, são conhecidas como botão pulsador ou botão em retenção (Figura 45). São muito utilizadas em função de sua segurança de operação, pois retomam a posição original em caso de queda de energia elétrica. Este tipo de botoeira aciona o motor apenas enquanto o botão é pressionado (Botoeiras e Sinalizadores, s.d.).



*Figura 45 – Botões pulsadores/sem retenção (Botoeiras e Sinalizadores, s.d.)*

Há botoeiras que mantêm a posição de comutação dos contatos e só retornam à posição anterior quando são pressionadas novamente. São chamadas de botoeiras com intertravamento ou retenção. Sua principal vantagem é a de possuir em um mesmo dispositivo duas ou três funções. Contudo, diferentemente do tipo anterior, este tipo de dispositivo de comando permanecerá com seu contato comutado em caso de queda de energia e, quando esta retornar, o mecanismo será retomado automaticamente, gerando possíveis riscos aos usuários e trabalhadores envolvidos. As chaves seletoras (Figura 46) e alguns tipos de interruptores funcionam dessa maneira também (Botoeiras e Sinalizadores, s.d.).



Figura 46 – Chave Seletora (Botoeiras e Sinalizadores, s.d.)

Outro dispositivo de comando muito utilizado, inclusive no controle de cadeiras de rodas elétricas, é o *joystick* (Figura 47). Um *joystick* consiste em uma haste que pivota em uma base e reporta seu ângulo, ou direção, para o dispositivo que ele está controlando. Normalmente são de dois eixos, mas existe também com controle nos três eixos cartesianos (X, Y e Z). Sua operação, na maioria dos casos, é semelhante à de um botão sem retenção, retornando à posição neutra quando a força aplicada a sua haste é interrompida (Pence, 2013). Em um modelo padrão de *joystick*, a haste é alongada em direção à base e se encaixa em dois eixos ranhurados e giratórios. Inclinar a haste para frente e para trás gira o eixo relacionado a Y, e inclinar para os lados gira o relacionado a X. Para não gerar confusão no usuário, o controlador organiza as informações de rotação das hastes com base no que para ele (usuário) é eixo X e Y. Para determinar a posição da haste, o sistema de controle do *joystick* monitora a posição de cada um desses eixos ranhurados. Esse monitoramento, em *joysticks* analógicos, é feito por potenciômetros, os quais interpretam a posição da haste como leituras analógicas, pela mudança de suas resistências (Harris, 2002).

É importante pontuar também que o “ponto de equilíbrio” desse tipo de dispositivo de comando não necessariamente é representado por uma leitura “zero-zero” das resistências nos eixos X e Y. Esse “zero-zero”, no entanto, pode ser obtido com uma leitura simples das resistências por um sensor ou dispositivo de leitura, como o Arduino. Esse cuidado é importante para calibração do mesmo e dos acionamentos que se pretende fazer, evitando erros e sua propagação ao implementar a solução.

Além disso, pela leitura dessas resistências e conhecendo o ponto de equilíbrio, é possível identificar em qual direção e sentido a haste do *joystick* foi movimentada. Futuramente, essa noção de sentido será importante no controle do manipulador.



*Figura 47 – Joystick analógico (Addicore, 2016)*

Por fim, esses mesmos dispositivos podem executar mais de um comando, a depender do tipo de acionamento feito neles. Pela interface com uma plataforma de prototipagem eletrônica de *hardware*, como o Arduino, é possível então fazer leituras diferentes utilizando contadores de toques em um botão, ou utilizando um sensor de toque capacitivo, o que permite diferenciar toques curtos (pulsos no botão) de toques prolongados. Essas possibilidades permitem reduzir o número de dispositivos de comando da interface com o usuário.

## 3 SOLUÇÃO CONCEITUAL

### 3.1 PREMISSAS

Há várias metodologias que organizam a execução de um projeto, e muitas delas convergem para que, antes de se mergulhar no desenvolvimento da solução propriamente dita, se defina o problema. Além do objetivo e motivações para o projeto, “o quê” e “porquês”, é importante definir algumas premissas, as quais norteiam todo o projeto, definindo requisitos mínimos de qualidade, limites que não podem ser ultrapassados e até onde se deseja ir com a pesquisa, além de uma noção de sucesso para a solução proposta. As premissas, por sua vez, focam no “como” da definição do problema.

Espera-se que o paciente seja capaz de comunicar ao manipulador como deseja que este se movimente, respeitando uma trajetória específica. Com esse intuito, faz-se necessário desenvolver os cálculos relacionados à cinemática inversa do manipulador e planejamento da trajetória do seu efetuador.

Embora este trabalho não tenha um apelo tão forte para o baixo custo, como o trabalho de base (Néto, 2019), este é um fator que estará em mente na tomada de decisões, mas que não será o principal norte do projeto. Essa preocupação inicial com um baixo custo leva à premissa de ser uma solução mais acessível em relação às do mercado. Essa acessibilidade transborda o campo financeiro e se apresenta também como uma acessibilidade intelectual. Em outras palavras, espera-se que a solução seja de fácil compreensão de seu funcionamento e que também seja de fácil utilização, não necessitando de grande conhecimento técnico por parte do usuário. Assim, como Steve Jobs nos projetos da *Apple* (Isaacson, 2011), este projeto busca solucionar a problemática de uma maneira simples e intuitiva. Pragmaticamente, o sucesso da solução proposta será mensurado, além da efetividade na execução dos comandos do usuário, pelo baixo esforço demandado do usuário para controlar o braço robótico.

### 3.1 SÍNTESE E CONCEPÇÃO

A fim de se visualizar as alternativas comerciais, acadêmicas e outras existentes na literatura, todas as informações relevantes para este projeto foram organizadas e resumidas na Tabela 5. Para a construção desta tabela, o desafio de controle do braço robótico foi dividido em alguns desafios intermediários: interface com o usuário, tipos de programação e modo de acionamento das juntas do manipulador.



Tabela 5 – Matriz de funções

Interface com o usuário	Tipo de programação	Modo de acionamento das juntas
Joystick	Online	Junta a junta
Teclado	Off-line	Diretamente pelo movimento do efetuador
Canudo/tubos para assoprar e sugar	Online, com algumas ações pré-carregadas	Controle apenas das juntas, mantendo o efetuador fixo
Processamento de imagens		Completamente autônomo pelo robô, com base em pré-definições
Mouse		
Movimento da cabeça com dispositivo infravermelho		
Mapeamento 3D do ambiente		
Workstation		
Mouthstick		

Pela variedade e interseção das soluções apresentadas nos capítulos anteriores do relatório, não se vinculou nenhum item da Tabela 5 com qualquer solução apresentada.

### 3.3 ANÁLISE

Como apresentado na Tabela 5, há muitas alternativas para controle de um braço robótico que atendem às necessidades de pessoas com lesões medulares no grupo cervical da vértebra 3 a 5. Contudo, o público-alvo do manipulador proposto possui incapacidade de realização de certas tarefas e suas consequências, no que tange o rendimento funcional do corpo, o que aumenta o desafio de controle e interface com o usuário. Assim, antes de propor uma solução, é necessário entender quantos, e quais, movimentos, ou comandos, um usuário tetraplégico é capaz de realizar.

Analisando a Figura 7, nota-se que um usuário com lesão na vértebra 3 tem grande parte de seus movimentos comprometidos, inclusive alguns controlados de maneira autônoma pelo sistema nervoso humano, gerando dificuldade na respiração, por exemplo. Assim, as possibilidades mecânicas de controle e interface do usuário se restringem demasiadamente, assim como a população usuária do

mecanismo. Dessa maneira, optou-se neste projeto por focar a proposta de uma solução para um paciente com lesões nas vértebras C4 ou C5 apenas, que possuem os movimentos do pescoço para cima mantidos.

Isto posto, a pesquisa voltou-se para a compreensão dos movimentos que um ser humano pode fazer do pescoço para cima. Obteve-se a Figura 48, sendo as setas azuis indicando os movimentos de translação da cabeça/pescoço, e em verde os de rotação:

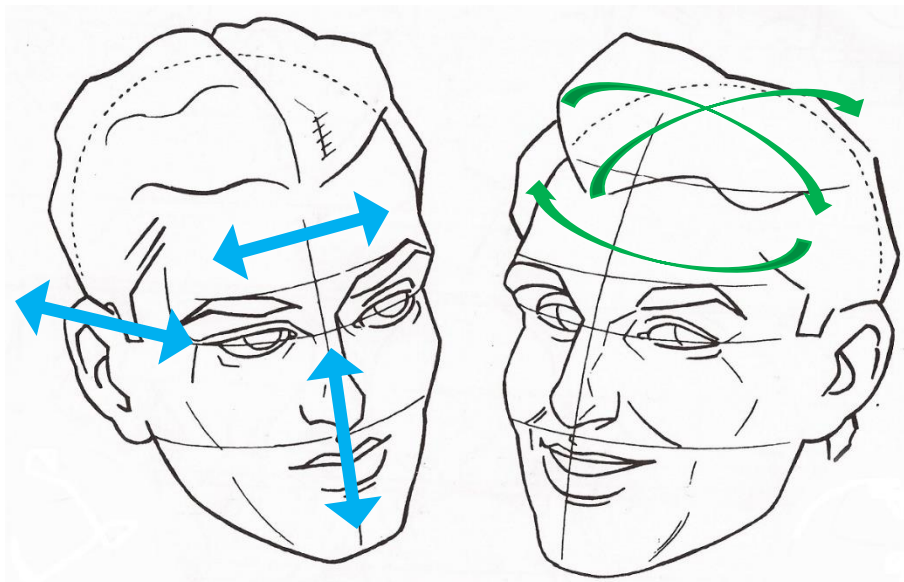


Figura 48 - Movimentos possíveis da cabeça/pescoço de um ser humano. Adaptado (*Cabeça e Face*, 2010)

Observa-se que não foram apresentados os movimentos de lábios, pálpebras, nariz e demais músculos faciais, uma vez que são movimentos muito sutis e sensíveis. Essas características são agravadas pelo fato de que mesmo indivíduos sem deficiência motora têm dificuldades em controlar os músculos faciais de maneira individual e com destreza. Focou-se, assim, nos movimentos principais do pescoço.

Contudo, preza-se pela ergonomia e conforto do usuário ao usar um mecanismo que envolva esses movimentos. Desta maneira, tem-se que os movimentos de translação na vertical são incomuns e desconfortáveis, e os de rotação na horizontal (para os lados, em relação a um eixo horizontal do pescoço) se mostram desconfortáveis e cansativos quando executados repetitivamente. Os demais movimentos foram mantidos ou pela similaridade com os movimentos demandados por um *joystick* de queixo, ou por serem naturais do corpo no dia-a-dia. Assim, as possibilidades de movimento se reduzem aos representados na Figura 49:

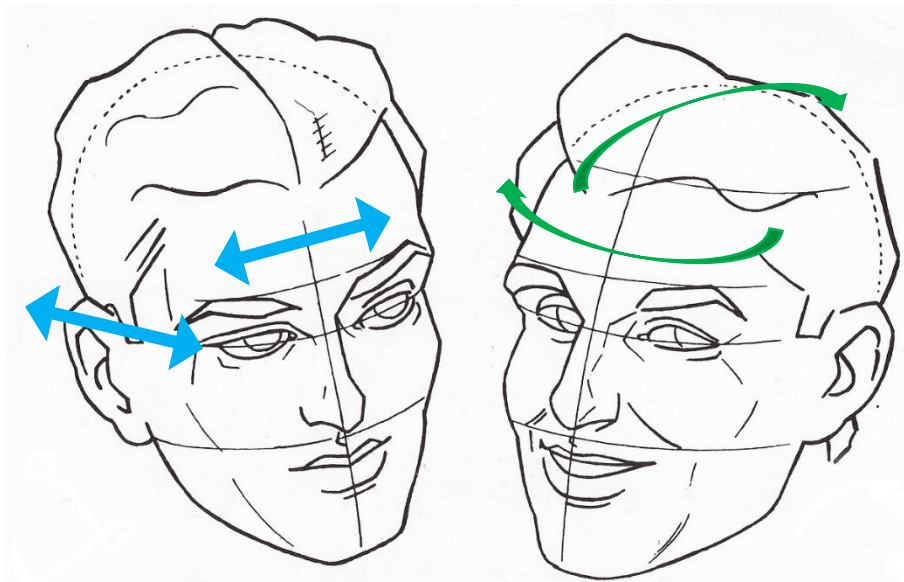


Figura 49 - Movimentos pertinentes da cabeça/pescoço de um ser humano. Adaptado (Cabeça e Face, 2010)

É relevante citar que cada um desses movimentos possui dois sentidos de comando, ou seja, a Figura 49 indica que é possível controlar até 8 variáveis de comando diferentes.

Diante das condições apresentadas, percebe-se que as soluções compiladas na Tabela 5 não atendem por completo às restrições do paciente. Analisando as interfaces com o usuário, teclados, mouses e joysticks, por exemplo, usualmente demandam o uso das mãos ou pés do paciente, que são disfuncionais em tetraplégicos. Ou seja, não são soluções adequadas para a problemática posta.

Há ainda os controladores que utilizam tubos para assoprar e sugar (Ferreira, 2008). Essa alternativa se mostra atrativa no controle de uma cadeira de rodas ou como interface com uma tela, usando os tubos como um mouse. No entanto, a necessidade de muitos tubos e a ocupação da boca do usuário, que no caso deste manipulador deseja se alimentar e hidratar, torna essa alternativa inadequada. A mesma análise pode ser feita para o uso de *mouthstick*.

Como não se deseja restringir os ambientes de utilização do manipulador, as alternativas de mapeamento 3D do ambiente e a utilização de estações de trabalho, ou bandejas no caso do Handy 1, são soluções que não atendem às premissas definidas e, portanto, não estão no horizonte de desenvolvimento de um projeto conceitual.

O predomínio da autonomia do controle pelo usuário em relação ao controle autônomo pelo robô, por sua vez, atribui às soluções que utilizam processamento de imagem incoerência em proporção às premissas estabelecidas. Além disso, dispositivos de análise e processamento de imagem são susceptíveis a erros e descalibração. Dessa forma, esse tipo de interface com o usuário foi descartado.

Outros métodos já foram propostos exclusivamente para controle de cadeiras de rodas por tetraplégicos, como pelo movimento dos olhos, por exemplo. Contudo, esse tipo de controle é muito susceptível à intensidade da luz solar, incomodando os olhos e o tornando uma solução limitada (EUA

Patente Nº 4207959, 1980), sem mencionar a alta complexidade de rastreamento e acompanhamento do movimento deles.

Assim, acredita-se que, dentre as soluções analisadas para interface com usuário, a melhor delas, com algumas adaptações, é a que utiliza o movimento do pescoço e sua posição angular para controlar um dispositivo. No entanto, pela alta complexidade deste tipo de interface, buscou-se sua adaptação. Extrapolando o que foi apresentado na seção de Estado da Arte, há *joysticks* que são apoiados no queixo de pacientes, utilizando o movimento da cabeça/pescoço para controlá-lo. Essa alternativa mescla tanto o uso dos movimentos do pescoço possíveis, quanto um dispositivo de controle já conhecido e utilizado. Assim, essa adaptação configura o uso de *joystick* como uma solução mais adequada.

Analisando também o tipo de programação utilizada nas soluções encontradas (*online* e *off-line*), as que mais apresentaram equilíbrio entre efetividade da solução e manutenção da autonomia do usuário em relação ao movimento do braço, ou da cadeira de rodas, foram as que mesclaram o controle do dispositivo por meio de uma interface com o pré-carregamento algumas posições e movimentos para o robô. Ou seja, as programações puramente *online* e *off-line* foram descartadas. No contexto deste projeto, algumas posições e ações são desejáveis de se pré-definir, quais sejam: trazer um objeto até a boca do usuário (para alimentos e bebidas), colo do usuário (para que este possa pegar objetos e transportá-los consigo, sem uso do braço), devolução (tem o intuito de colocar o objeto agarrado de volta na posição em que foi agarrado) e casa (posição onde o braço é recolhido sobre si mesmo).

No que concerne o controle de um manipulador, este pode ser feito, do ponto de vista de acionamento, de quatro maneiras: controle junta a junta; controle diretamente da posição do efetuador terminal; controle apenas das juntas, mas mantendo o efetuador fixo; e controle completamente autônomo pelo robô, com base em pré-definições. Neste ponto da análise é importante ponderar duas coisas: as pessoas possuem noções e percepções espaciais diferentes, e deseja-se que elas possuam autonomia no controle do braço robótico. Com isso, os dois últimos modos de acionamento das juntas apresentados não atendem a essas premissas, ou por tirarem autonomia do usuário (controle completamente autônomo do robô), ou por demandarem noção espacial mais avançada (controle apenas das juntas com o efetuador fixo). Ainda, a solução junta a junta se apresenta morosa para o usuário, além de demandar uma compreensão espacial e de comportamento de mecanismos “avançada”. Dessa maneira, essa alternativa também foi descartada.

Assim, o controle da posição do efetuador se mostra a melhor opção para controle pelo usuário. Embora várias variáveis possam ser controladas e vários comandos possam ser realizado com os movimentos selecionados do pescoço (Figura 49), foram levantadas apenas as variáveis mínimas, e seus respectivos valores possíveis, para a realização segura de uma atividade como pegar um objeto. As variáveis consideradas foram: eixo de translação (no caso do controle direto do efetuador terminal), acionamento dos motores (sentido positivo e negativo de rotação), garra (agarrar e soltar), além das variáveis de orientação do pulso (Yaw, Pitch e Roll). Tendo em vista ser um protótipo e proposta inicial,

as ações pré-programadas não entraram nessa análise. Com essas definições, foi construída a Tabela 6, para controle junta a junta e controle diretamente do efetuador, respectivamente.

*Tabela 6 - Comandos necessários para controle da posição escalar do efetuador terminal*

<b>Para controle diretamente do efetuador terminal</b>					
<i>Variáveis</i>	<i>Valores possíveis</i>				<i>Nº de possibilidades</i>
Eixo de movimento do efetuador	Eixo X	Eixo Y	Eixo Z	-	3
Acionamento motores	Positivo		Negativo		2
Garra	Agarrar		Soltar		2
Orientação do efetuador	Yaw ( $J_4$ )	Pitch ( $J_5$ )	Roll ( $J_6$ )	-	3
<i>Total de possibilidades</i>					10

Da Tabela 6, nota-se que há 10 diferentes comandos a serem executados em cada uma delas, sendo 4 variáveis a serem controladas, em ambos os modos de controle dos motores. Pela Figura 49 há pelo menos um movimento para cada variável controlável. Além disso, como já abordado, dispositivos de controle como botões, *joysticks* chaves e alavancas podem executar mais de um comando, a partir de diferentes tipos de acionamentos (pressionamento contínuo e com contador de toques) ou posições das chaves e alavancas. Isso viabiliza diversas possibilidades a partir de um mesmo dispositivo de controle.

Diante do exposto, propõe-se então a utilização dos movimentos confortáveis do pescoço (Figura 49) para controle do braço robótico. Para este controle, sugere-se utilizar um *joystick*, com o paciente teleguiando os movimentos do manipulador.

Com essas definições, deve-se partir para as próximas etapas do projeto: detalhamento da solução proposta, realizando os cálculos de cinemática inversa, volume de trabalho e razões de velocidades necessárias para planejamento da trajetória; além de verificação da efetividade dessa solução, por meio de testes e/ou simulações.

### **3.6 RESUMO DO CONCEITO DO PROJETO**

Diante das análises desenvolvidas, o conceito proposto da solução se resume em um *joystick* como dispositivo básico de controle, acionado pelos movimentos da cabeça/pescoço do usuário, utilizando uma programação majoritariamente online. Essa solução permitirá ao usuário controlar o efetuador

terminal cartesianamente, por meio da navegação pelas 4 variáveis de controle abordadas. Como sugerido (Néto, 2019) e mantido, o controle dos atuadores será feito por Arduino.

## 4 DESENVOLVIMENTO DA SOLUÇÃO

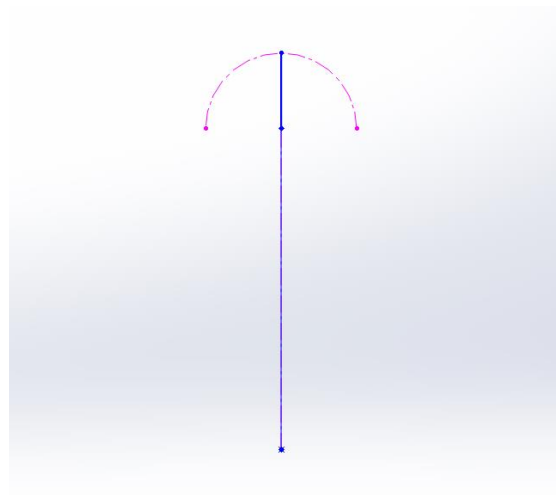
### 4.1 VOLUME DE TRABALHO

Antes de investir esforços na solução da cinemática inversa do manipulador e no planejamento da trajetória do seu efetuador, é importante entender onde este manipulador pode trabalhar e onde ele consegue chegar com seu efetuador, análise que não fora elaborada até então.

Duas metodologias de resolução dessa problemática foram analisadas: uma numérica, encontrando todos os vetores-posição possíveis para o efetuador; e outra geométrica, que envolve variar as juntas “manualmente” dentre seus pontos de máximo e mínimo e encontrar as superfícies que o efetuador percorrem. Optou-se pelo segundo método em virtude da facilidade didática e de visualização do volume de trabalho do manipulador.

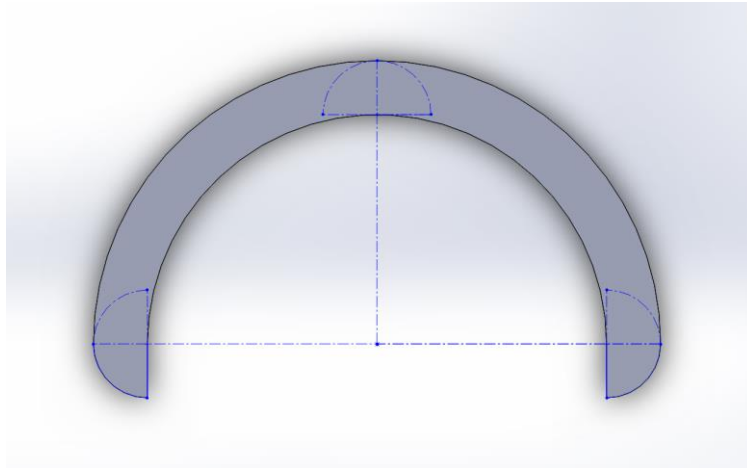
Utilizou-se os limites de junta definidos geometricamente (Néto, 2019) e, com o auxílio do SolidWorks, foi possível entender e visualizar como os graus de liberdade do manipulador se comportam. Para isso, o problema do volume de trabalho do manipulador foi dividido em dois subproblemas: o primeiro envolvendo apenas os dois últimos graus de liberdade (juntas 5 e 6) do manipulador, e um segundo problema envolvendo os demais graus de liberdade anteriores (juntas 1 a 4).

O primeiro problema foi simples de ser analisado, uma vez que a variação das juntas 5 e 6 pelos seus limites de movimentação gerava, com as demais juntas imóveis, um semicírculo, como representado em rosa na Figura 50. Em azul é a estrutura mecânica do manipulador.



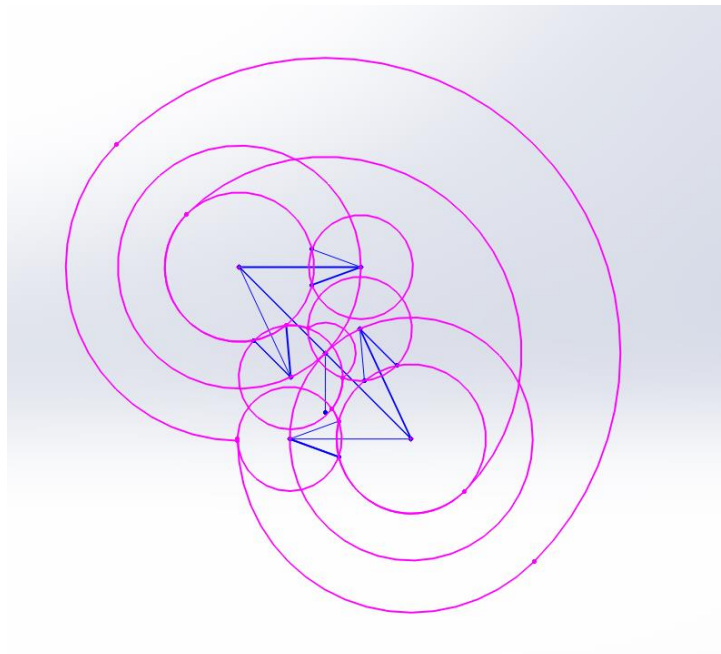
*Figura 50 - Trajetória de trabalho das juntas 5 e 6*

Movimentando agora a junta 1 ( $\theta_1$ ) continuamente entre seus valores máximo e mínimo ( $+90^\circ$  e  $-90^\circ$ ), obtém-se a área de trabalho ilustrada na Figura 51 para o subproblema 1:



*Figura 51 - Área de trabalho do subproblema 1*

Uma metodologia semelhante foi usada para o segundo subproblema, mas com algumas adaptações. As variações constantes dos ângulos máximo e mínimo das juntas 1 a 4 foi feita de maneira gradual, ou seja, para cada extremo de cada junta variou-se as demais juntas entre seus respectivos máximo e mínimo. Tais posições extremas foram representadas pelas linhas em azul escuro nas figuras a seguir. Inicialmente o ângulo  $\theta_1$  não foi variado. Essa sequência de passos possibilitou a construção de uma série de arcos de circunferência que se interceptam em alguns pontos, representados na Figura 52.



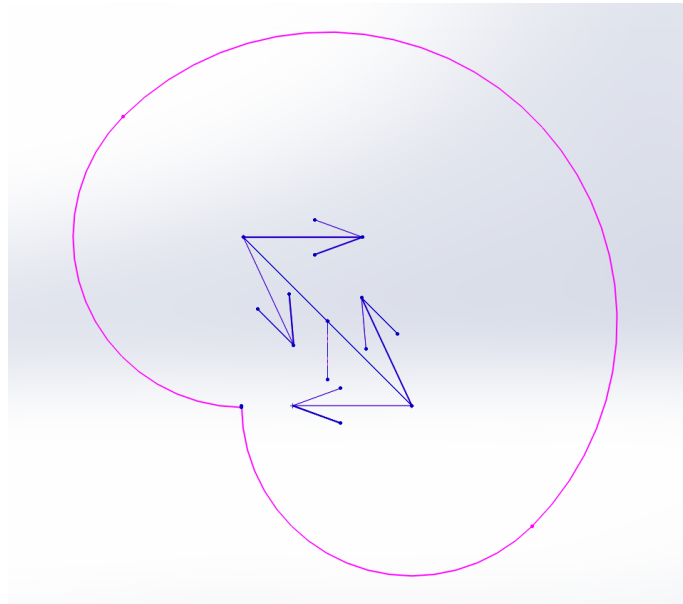
*Figura 52 - Linhas de construção volume de trabalho*

Tais arcos representam tanto a trajetória percorrida por cada junta, quanto a trajetória do efetuador, quando  $\theta_5$  e  $\theta_6$  não variam.

Em seguida, foi feita uma limpeza do desenho traçado anteriormente, definindo as áreas que seriam alcançáveis dentro desses arcos de circunferência, a fim de se obter a área de trabalho do efetuador, no

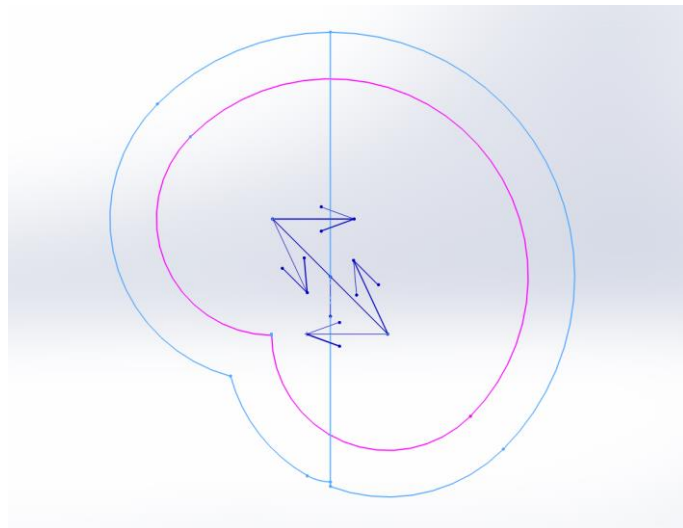


plano XZ. Essa “limpeza” gerou a área representada na Figura 53, delimitada pelos contornos cor-de-rosa:



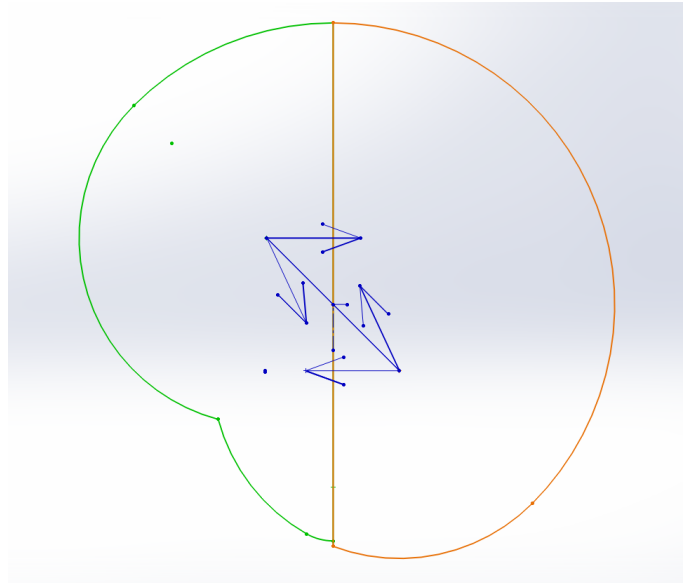
*Figura 53 - Área de trabalho das juntas 2 a 4*

O comportamento de  $\theta_5$  e  $\theta_6$ , estudados no subproblema 1, expandiu a área acima, representada em azul claro na Figura 54, frente à área original e as possíveis configurações limites do manipulador:



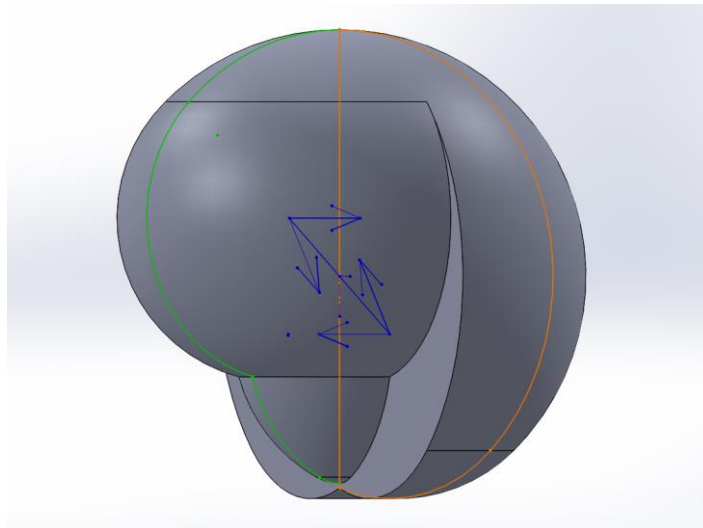
*Figura 54 - Área lateral final, com os movimentos das juntas 2 a 6*

Por fim, para completar o desenho com a variação de todas as juntas, as áreas encontradas foram rotacionadas em torno da base (junta 1), respeitando os limites estipulados. Foram feitas duas rotações: uma da área lateral-frontal (delimitada por contornos em laranja), e outra da área lateral-traseira (delimitada por contornos verdes), indicadas na Figura 55:



*Figura 55 - Área de trabalho lateral do manipulador considerando movimento de todas as juntas*

Por fim, o volume de trabalho final pode ser compreendido por:



*Figura 56 - Volume de trabalho do manipulador robótico assistivo*

É importante pontuar que este volume de trabalho (Figura 56) não leva em consideração o volume ocupado pela cadeira e pelo usuário, nem o volume ocupado pela própria estrutura e componentes do robô.

## **4.2 ROBOTICS TOOLBOX FOR MATLAB (RTB)**

A utilização da toolbox de robótica do MATLAB foi útil não apenas para auxiliar na visualização da modelagem esquemática do manipulador a partir de valores para as variáveis de junta (Néto, 2019), mas também para dar mais agilidade a algumas contas e validar se as soluções calculadas a frente são geometricamente viáveis e podem ser esquematizadas.

A Toolbox possui funções desde cálculo da cinemática direta, inversa, visualização de trajetórias e cálculo do jacobiano de manipuladores robóticos (Corke, 2001). Ao longo deste capítulo será possível ver a importância dessa ferramenta inclusive na análise do comportamento de algumas alternativas propostas a frente para a solução do problema de cinemática inversa.

Os programas utilizados para estes cálculos e demais modelagens utilizadas neste trabalho se encontram no capítulo de Apêndices.

## 4.3 CINEMÁTICA INVERSA

A partir da modelagem do robô construído e da sua cinemática direta (Néto, 2019), é possível desenvolver a cinemática inversa. Analisando de maneira mais cuidadosa, algumas características do manipulador tornam a tarefa de cinemática inversa mais complexa.

### 4.3.1 PROBLEMÁTICA

Antes de tudo, é notável que o fato da construção das juntas de orientação,  $J_4$  a  $J_6$ , não ter eixos que se cruzam em um mesmo ponto descarta uma solução simples, e já desenvolvida algebricamente na literatura. Além disso, o espaço de trabalho no manipulador tem apenas 3 variáveis para navegação (eixos X, Y e Z), enquanto o manipulador em si possui 6 GDL, o que indica mais de uma solução para uma mesma posição do efetuador.

Do ponto de vista do jacobiano de velocidades, essa configuração implica em uma matriz retangular ( $m \neq n$ ) do jacobiano, se desconsiderar os ângulos de orientação. Esta situação indica que, pelas definições de determinante e matriz inversa, a matriz jacobiana não pode ser invertida analiticamente.

A última junta ( $J_6$ ) também pode adotar infinitos valores sem alterar o ponto alcançado pelo efetuador terminal. Essa característica por si só já torna o manipulador redundante. Cabe então entender se é desejável que o efetuador alcance objetos com valores diferentes de  $\theta_6$ , se este pode ser pré-definido ou, dada as tarefas que o robô é destinado a realizar, uma posição é mais utilizada ou preferida.

Analisando mais precisamente a junta  $J_5$ , percebe-se que não basta que os elos anteriores alcance o ponto por ela ocupado, mas que chegue sempre em uma mesma orientação ( $\theta_4$ ), pois qualquer variação neste, um ponto  $P$  previamente alcançável deixa de sê-lo. Nas Figuras 57 e 58 a junta  $J_5$  ocupa a mesma posição, e há mesmos valores para  $\theta_5$  e  $\theta_6$ , porém a orientação final do efetuador muda, em função de diferentes valores para as variáveis de junta anteriores. Ou seja, não basta que o ponto ocupado pela junta 5 seja alcançado, mas **como** é alcançado também é relevante:

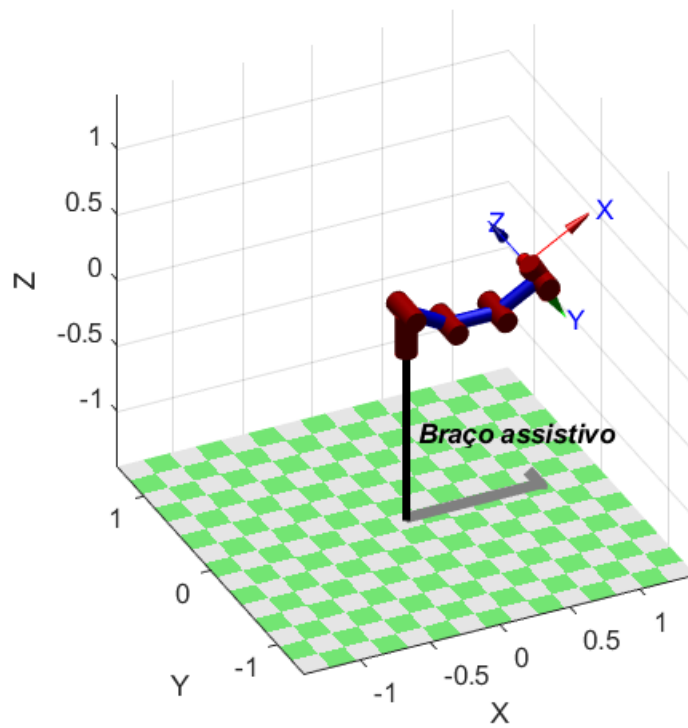


Figura 57 - Cotovelo para baixo

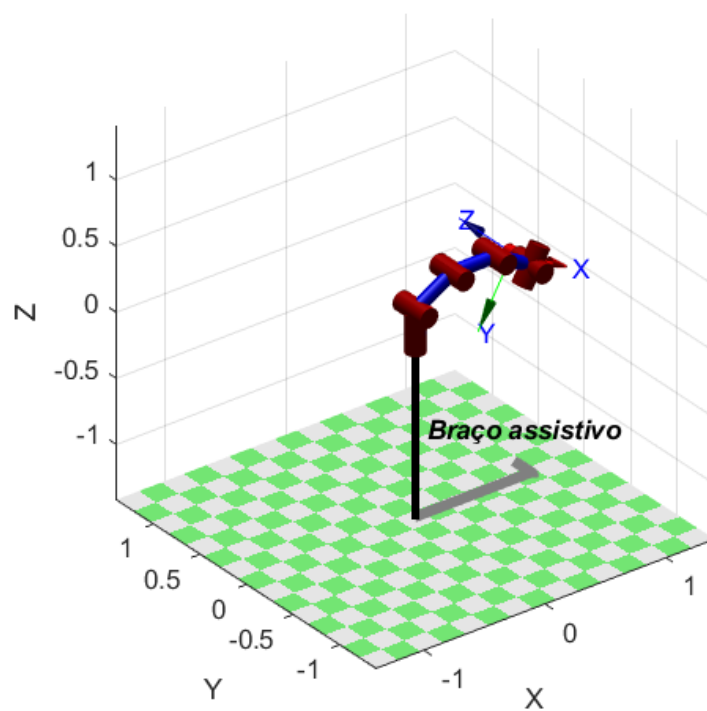


Figura 58 - Cotovelo para cima

Ainda, é notável uma redundância entre os movimentos das juntas  $J_1$  e  $J_5$ . Isso quer dizer que ajustes diferentes nas demais juntas deverão ser realizados caso, de uma posição para outra do efetuador, se opte por movimentar a junta  $J_1$  em vez da  $J_5$ , ou vice-versa. Essa decisão atribui mais complexidade não apenas na solução, mas no programa a ser implementado no sistema de controle do manipulador.

Tais problemáticas descartam uma solução analítica simplificada, seja algébrica ou geométrica, e despertam o olhar para outras alternativas.

Como abordado anteriormente, a toolbox do MATLAB possui uma função para cálculo da cinemática inversa (Corke, 2001). Para se verificar a eficácia dessa ferramenta, foi feita uma programação para um problema mais simples, robô planar, que tem o intuito de comparar se as funções de cinemática direta (*fkine*) e inversa (*ikcon*) retornam os respectivos resultados esperado. Para um robô planar (Figura 59), a função de cinemática direta funcionou bem, mas a de cinemática inversa apresentou algumas divergências, e foi então descartada.

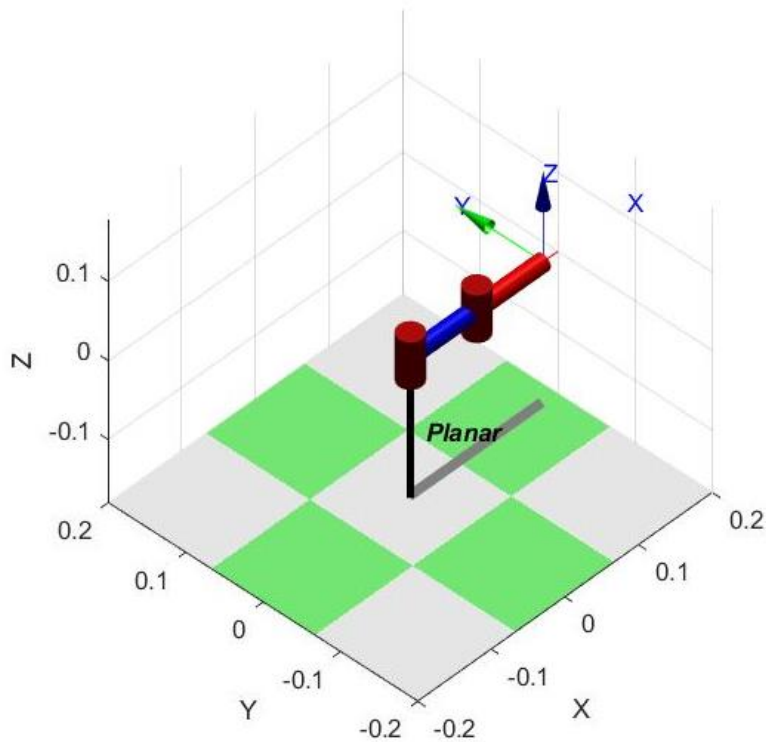


Figura 59 - Modelagem de um robô planar

Tabela 7 - Matriz de validação da funções *fkine* e *ikcon* da Toolbox

Inputs iniciais para $q_1$ e $q_2$	$x, y$ pela cinemática direta	$q_1$ e $q_2$ pela cinemática inversa
0; 0	0.18; 0	0; 0
$\pi/2$ ; $\pi/2$	-0.09; 0.09	$\pi$ ; $\pi^*$
0; $-\pi/4$	0.1536; -0.06364	0; $-\pi/4$
$\pi/4$ ; 0	0.1273; 0.	$\pi/4$ ; 0

Como assinalado na Tabela 7, a função de cinemática inversa retornou um valor diferente do esperado para o ponto  $x, y = (-0.09; 0.09)$ . As variáveis de junta calculadas posicionariam o efetuador terminal na base do manipulador, ponto  $x, y = (0; 0)$ , de acordo com sua cinemática direta, o que

inclusive é um ponto inalcançável para o manipulador. Por esses motivos, essa alternativa foi desconsiderada para solucionar o problema de cinemática inversa.

Como o pulso do manipulador não é esférico, ou seja, os eixos de orientação não se cruzam em um mesmo ponto, a solução mista proposta por Vladimir Lumelsky (Erthal, 1992) foi levada em consideração, dividindo o problema em posicionamento e orientação (Figura 60):

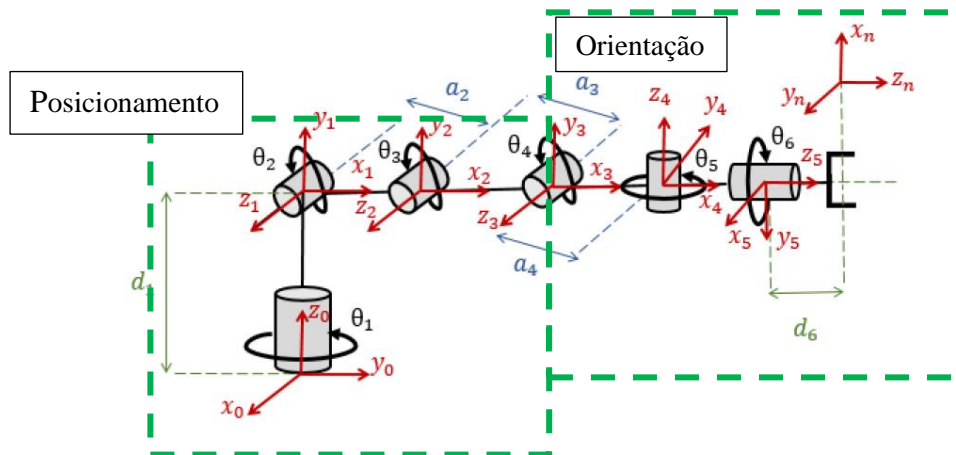


Figura 60 – Separação do manipulador em posicionamento e orientação. Adaptado (Néto, 2019)

Mas essa solução demonstrou demandar muitos recursos computacionais (software de processamento), além de agilidade para se fazer isso em tempo real, como almejado. Além disso, esta solução não resolve a problemática de que  $\theta_6$  pode assumir infinitos valores. Portanto, não foi adotada.

Uma solução discutida que foge às alternativas apresentadas até então é a de modificar construtivamente o manipulador, propondo um mecanismo que permita os três eixos de orientação se interceptando em um mesmo ponto. Na literatura inclusive já há soluções geométricas para este desafio (Craig, 2012). Esta alternativa definiria uma nova configuração e modelagem para o manipulador, agora com o pulso esférico, representada na Figura 61:

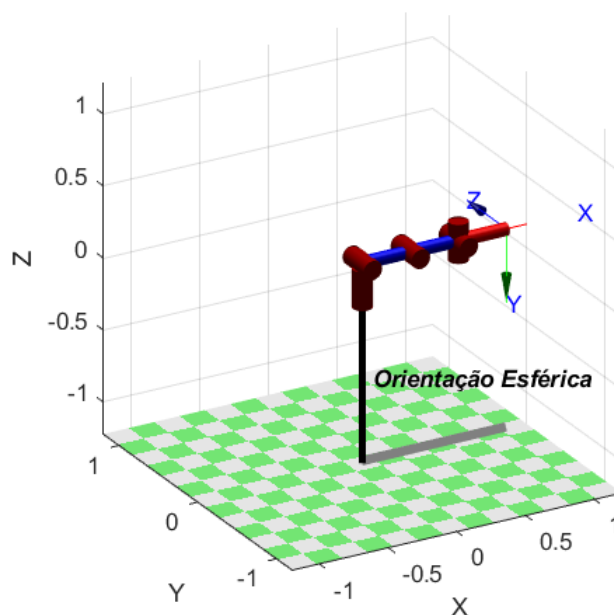


Figura 61 – Modelagem do manipulador com pulso esférico

No entanto, verificou-se que é uma solução construtivamente complicada e, durante o desenvolvimento e estudo da alternativa, se mostrou um mecanismo complexo de compreender, reproduzir e fabricar. Por isso, esta alternativa também foi descartada.

Por fim, pensou-se nas tarefas que foram mapeadas para o manipulador realizar (Filho, 2019) e (Néto, 2019), e se havia algum padrão ou predominância de modos de se utilizar o robô. Primeiramente percebeu-se que o paciente só desejaria pegar objetos que estejam contemplados pelo seu campo de visão, limitado pelas restrições de movimentação do corpo. Ou seja, objetos “escondidos”, atrás, debaixo ou dentro de outros dificilmente serão localizados pelo paciente. Essa condição possibilita criar restrições de movimento para o manipulador, pré-definindo alguns valores para as variáveis de junta.

A primeira restrição a ser imposta é a de que o efetuador, para alcançar os objetos do campo de visão do usuário, chegue então paralelo ao solo. Em outras palavras, o manipulador se aproximará dos objetos sempre “de frente”. Para que isso ocorra, a variável de junta  $\theta_4$  deve ser o oposto da soma das variáveis  $\theta_2$  e  $\theta_3$ :

$$\theta_4 = -(\theta_2 + \theta_3) \quad (37)$$

Além disso, para solucionar também o problema de redundância entre os movimentos das juntas  $J_1$  e  $J_5$ , optou-se por pré-definir o valor de  $\theta_5$ . Tendo em vista a premissa de alcance do campo de visão, o valor mais adequado para esta variável de junta foi de  $\theta_5 = -\pi/2$  *radianos*. É importante pontuar que, na Toolbox do MATLAB, a posição de origem (posição-zero) da junta  $J_5$  é dada por esse valor para  $\theta_5$ .

Por fim, em relação ao valor de  $\theta_6$ , entende-se que grande parte dos objetos podem ser abordados pela garra com duas orientações diferentes: vertical e horizontal, e que quaisquer orientações diferentes seriam em função da rotação de uma maçaneta ou um controle mais fino do objeto. Com isso em mente, optou-se por utilizar um sistema de chaveamento para alternar entre essas duas orientações: horizontal ( $0$  *radiano*) e vertical ( $\pm \pi/2$  *radianos*). Em ambos os casos, portanto, o valor de  $\theta_6$  permaneceria constante, e conhecido.

As restrições impostas permitiram solucionar a cinemática inversa de maneira geométrica, combinando as duas situações apresentadas em 2.4.1.

### 4.3.2 SOLUÇÃO

Conforme discussão acima, o valor de  $\theta_4$  será dado pela Equação 37, assim como  $\theta_5$  e  $\theta_6$  terão valores fixos e pré-determinados.

Para esta análise, o manipulador então foi modelado em uma posição qualquer,  $q_a$ :

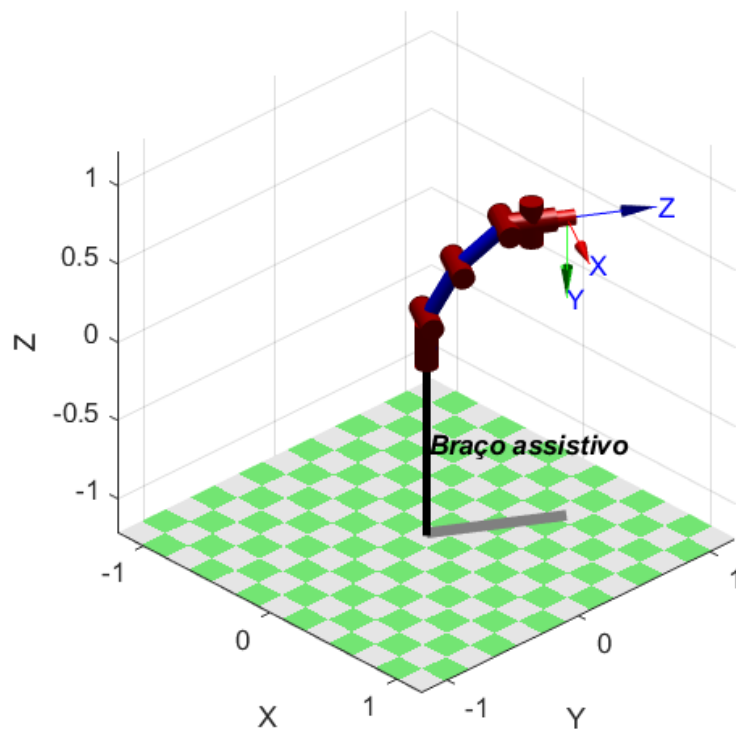


Figura 62 - Vista isométrica da posição  $q_a$

Como prevê a metodologia (Craig, 2012), o problema foi observado de cima (Figura 63) e de frente (Figura 64).

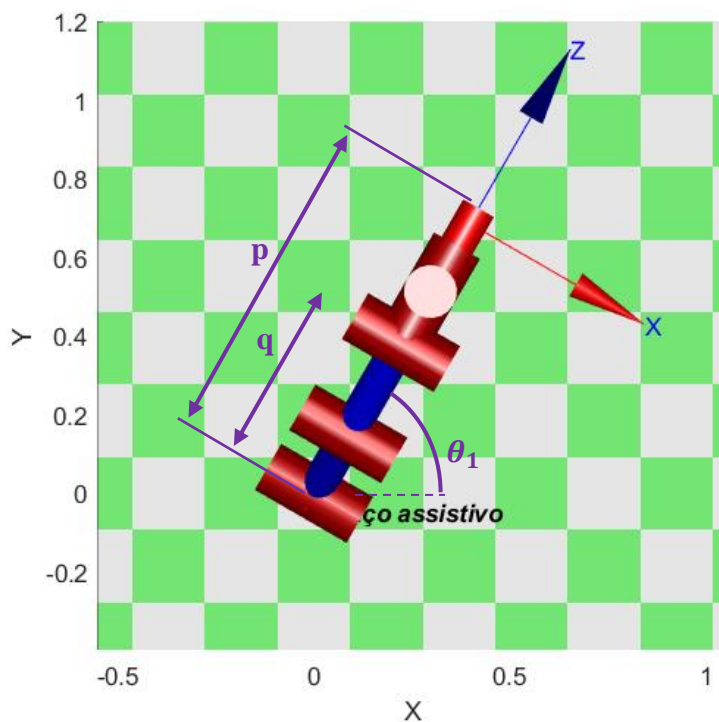


Figura 63 - Vista superior do manipulador na posição  $q_a$



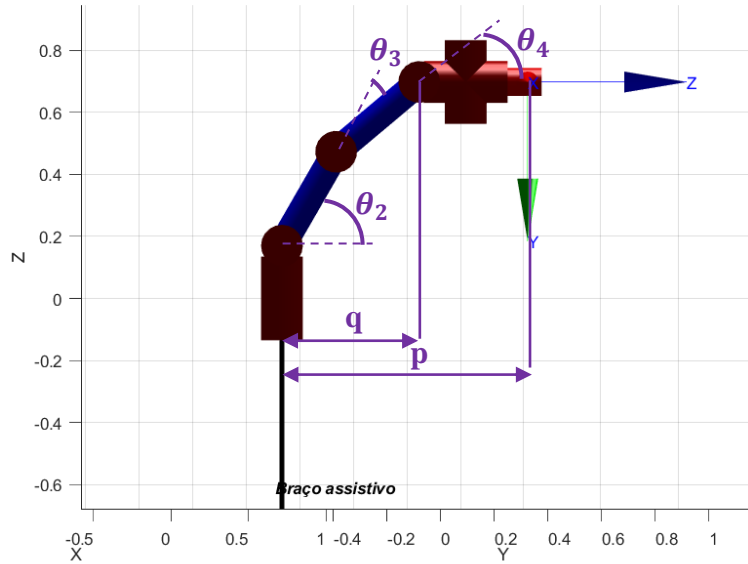


Figura 64 - Vista frontal do manipulador na posição  $q_a$

Os cálculos foram realizados com base nas variáveis simbólicas de elos e ângulos já definidas (Néto, 2019), e as componentes da posição do efetuador terminal, pelo sistema de coordenadas da base do manipulador são  $x$ ,  $y$  e  $z$ . Tais valores devem ser obtidos para cálculo de uma solução numérica para o problema.

As variáveis  $p$  e  $q$  representam, respectivamente, a distância do centro da junta  $J_2$  até o centro do efetuador, e do centro da mesma junta  $J_2$  ao centro da junta  $J_4$ . São variáveis auxiliares e se relacionam da seguinte forma:

$$p = \sqrt{x^2 + y^2} \quad (38)$$

$$q = p - a_4 - d_6 \quad (39)$$

Como  $\theta_5$  não irá variar, a despeito de cálculo da cinemática inversa, a vista superior do manipulador fornece a equação para a variável de junta da base:

$$\theta_1 = \text{tg}^{-1} \left( \frac{y}{x} \right) \quad (40)$$

O que implica nas seguintes relações:

$$x = p \cos(\theta_1) \quad (41)$$

$$y = p \sin(\theta_1) \quad (42)$$

Já da vista frontal, obtém-se a equação geométrica de  $z$  e de  $x$ , levando em consideração para esta a condição apresentada na Equação 37:

$$z = d_1 + a_2 \sin(\theta_2) + a_3 \sin(\theta_2 + \theta_3) \quad (43)$$

$$x = [a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3) + (a_4 + d_6) \cos(\theta_2 + \theta_3 + \theta_4)] \cos(\theta_1) \quad (44)$$

Para encontrar a equação de  $\theta_3$ , será necessário inserir mais duas variáveis auxiliares,  $r$  e  $d$ , na vista frontal:

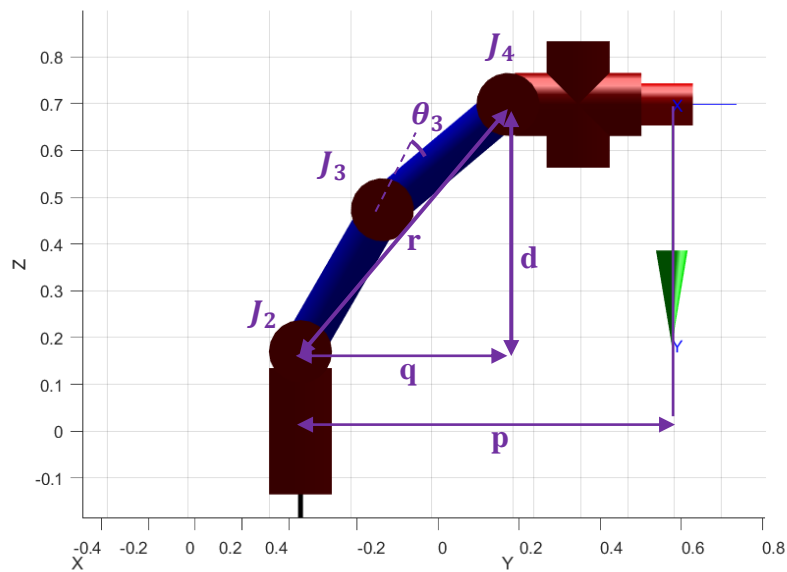


Figura 65 - Variável auxiliar  $r$

A variável  $d$  se relaciona com o tamanho do primeiro elo do robô e com a posição, em  $z$ , do efetuador terminal:

$$d = |z - d_1| \quad (45)$$

Já  $r$ , pode ser solucionado pelo Teorema de Pitágoras, considerando o triângulo formado pelos segmentos  $q$ ,  $r$  e  $d$ :

$$r = \sqrt{d^2 + q^2} = \sqrt{(z - d_1)^2 + (p - a_4 - d_6)^2} \quad (46)$$

Com todas as variáveis auxiliares definidas, aplica-se a Lei dos Cossenos no triângulo composto pelas juntas  $J_2, J_3$  e  $J_4$ :

$$r^2 = (a_2)^2 + (a_3)^2 - 2a_2a_3 \cos(\pi - \theta_3) \quad (47)$$

Sendo cosseno uma função par e  $\cos(\theta) = \cos(\pi - \theta)$  e isolando a variável de interesse, tem-se:

$$\theta_3 = \pm \cos^{-1} \left[ \frac{r^2 - (a_2)^2 - (a_3)^2}{2a_2a_3} \right] \quad (48)$$

A decisão por “cotovelo para cima ou para baixo” implicará em uma das duas soluções dadas pela equação acima. Na etapa de implementação, esta decisão será tomada com base na menor diferença entre valor atual e novo valor para  $\theta_3$ .

Por fim, para se obter a equação para  $\theta_2$ , recorreu-se a mais algumas variáveis auxiliares no plano frontal. Para isso, o desenho esquemático a seguir foi construído:

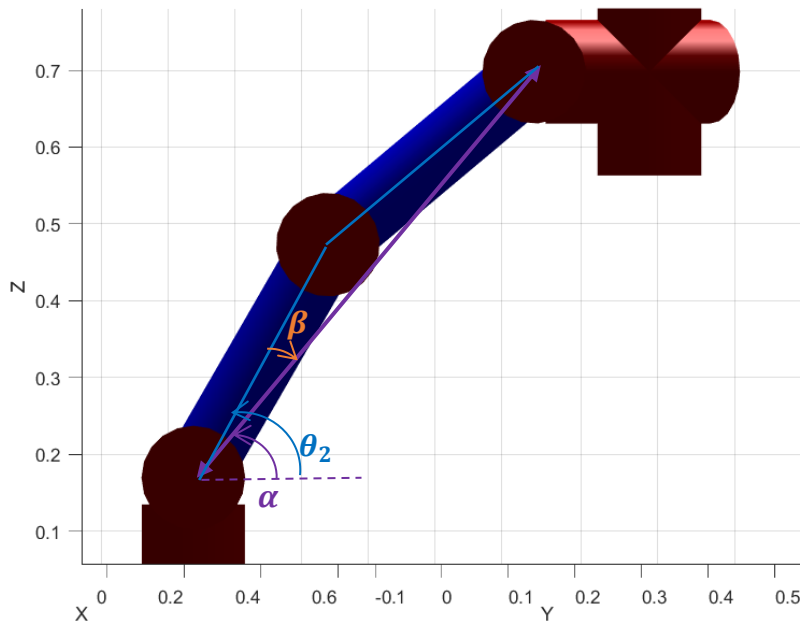


Figura 66 - Ângulos auxiliares no plano frontal

Pela Figura 66, tem-se a seguinte relação para os ângulos auxiliares apresentados:

$$\theta_2 = \alpha - \beta \quad (49)$$

Assim, a propriedade trigonométrica da tangente para diferença de arcos foi usada:

$$\operatorname{tg}(\alpha - \beta) = \frac{\operatorname{tg}(\alpha) - \operatorname{tg}(\beta)}{1 + \operatorname{tg}(\alpha) \cdot \operatorname{tg}(\beta)} \quad (50)$$

Sendo

$$\operatorname{tg}(\alpha) = \frac{d}{r} = \frac{|z - d_1|}{r} \quad (51)$$

$$\operatorname{tg}(\beta) = \frac{a_3 \sin(\theta_3)}{a_2 + a_3 \cos(\theta_3)} \quad (52)$$

Assim, após as devidas substituições e simplificações, a última equação foi encontrada:

$$\theta_2 = \operatorname{tg}^{-1} \left[ \frac{(z - d_1)(a_2 + a_3 \cos(\theta_3)) - (a_3 \sin(\theta_3)r)}{r(a_2 + a_3 \cos(\theta_3)) + (z - d_1)(a_3 \sin(\theta_3))} \right] \quad (53)$$

Com isso, o problema de cinemática inversa foi resolvido, dadas as restrições impostas.

#### 4.4 VOLUME DE TRABALHO ATUALIZADO

Com as novas condições de trabalho estabelecidas, os limites das variáveis de junta e seus valores possíveis foram atualizados (Tabela 8), assim como o volume de trabalho do manipulador, que deve ser recalculado.

Tabela 8 – Novos Limites angulares das juntas

Variáveis de junta	Novos limites
$\theta_1$	$+\pi/2$ a $-\pi/2$ radianos
$\theta_2$	$-\pi/4$ a $3\pi/4$ radianos
$\theta_3$	$-3\pi/4$ a $8\pi/9$ radianos
$\theta_4$	$-59\pi/36$ a $\pi$ radianos
$\theta_5$	0 radianos (ou $-\pi/2$ para a Toolbox)
$\theta_6$	0 ou $\pm \pi/2$ radianos

O volume de trabalho atualizado foi calculado utilizando a mesma metodologia apresentada na sessão 4.1, e o passo a passo desse cálculo é apresentado a seguir.

A Figura 67 representa o esqueleto do manipulador ocupando todas as posições extremas, ou seja, todas as combinações possíveis com as juntas em seus valores máximo e mínimo, e respeitando as restrições impostas. A base no manipulador é indicada pela letra *B*.

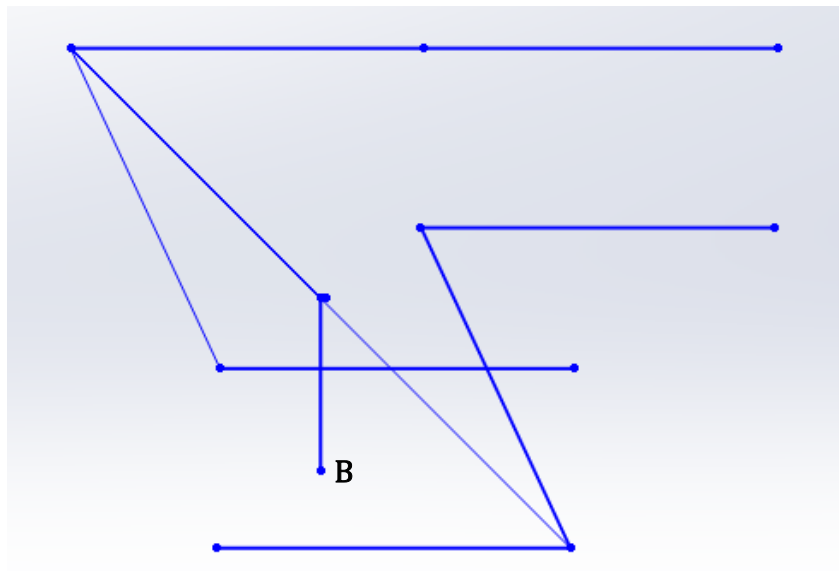
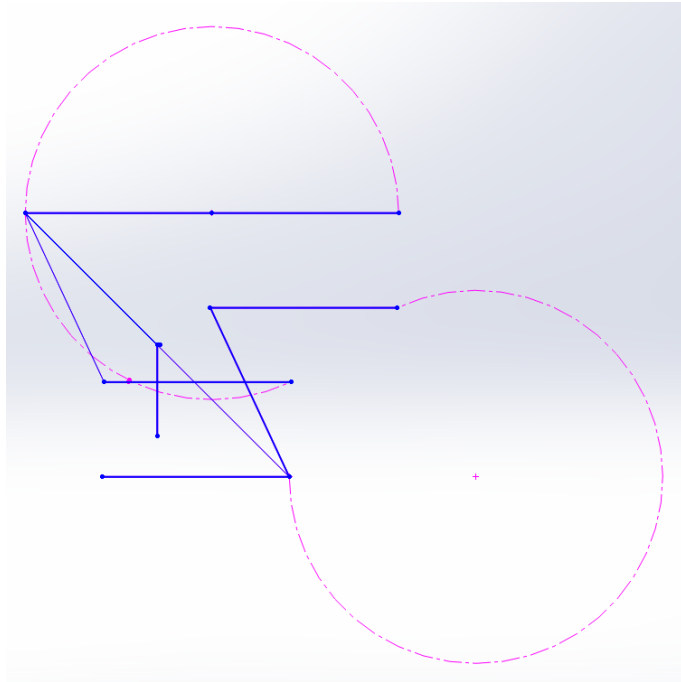


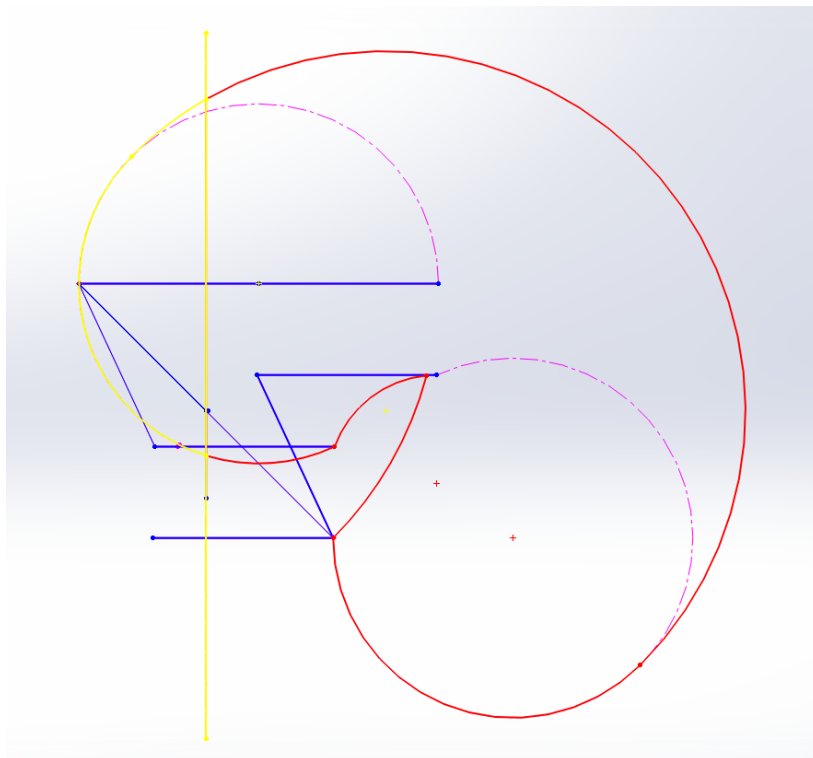
Figura 67 - Desenho esquemático do manipulador em seus máximos e mínimos

Em seguida, na Figura 68 foi desenhada a trajetória que o efetuador desempenha quando apenas  $\theta_3$  varia entre seu máximo e mínimo.



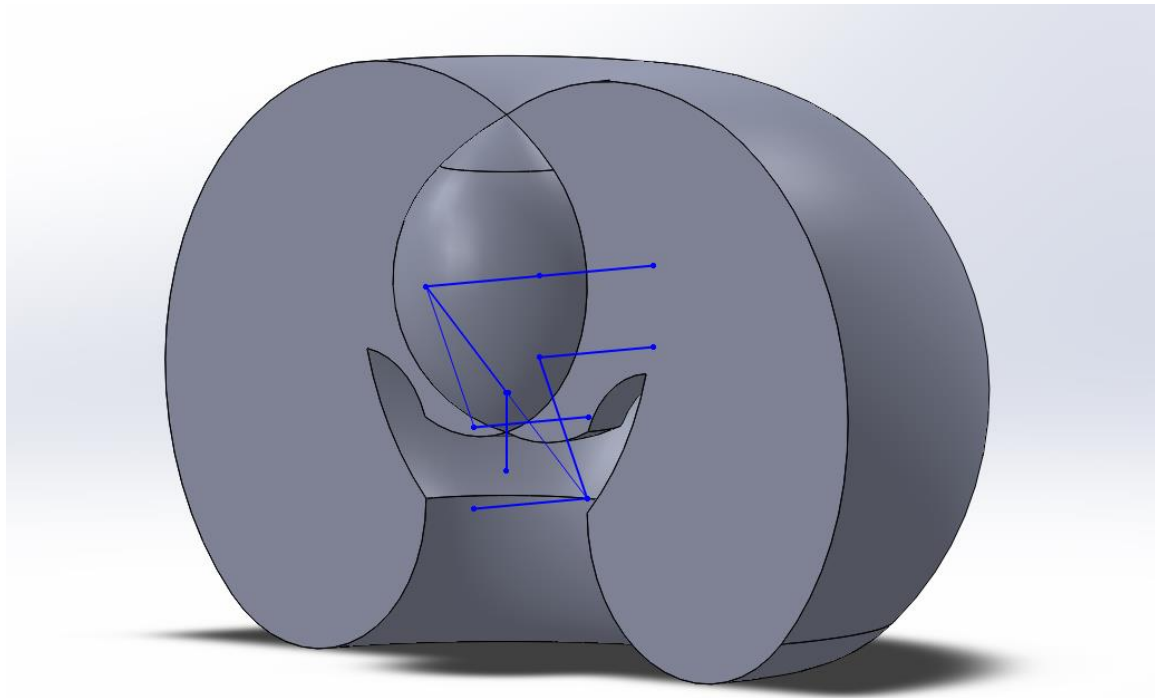
*Figura 68 - Trajetórias do efetuador, a partir da movimentação da junta  $J_3$*

Variando as demais juntas e limpando o desenho para representar apenas os contornos externo, foi obtida a Figura 69:



*Figura 69 - Contornos externos (amarelo e vermelho) da área de trabalho lateral do manipulador*

Por fim, rotacionando a junta  $J_1$  entre seus valores máximo e mínimo, foi obtido o volume de trabalho apresentado na Figura 70



*Figura 70 - Volume de trabalho atualizado do manipulador*

Assim como o volume de trabalho calculado originalmente (Figura 56), o ilustrado pela Figura 70 desconsidera os espaços ocupados tanto pela cadeira de rodas e pelo paciente, quanto pela própria estrutura do manipulador.

#### **4.5 PLANEJAMENTO DA TRAJETÓRIA**

Para o planejamento da trajetória, foram analisadas as alternativas de solução apresentadas na sessão 2.5, e optou-se por aquela que mais fazia sentido da ótica do usuário. A interface de controle escolhida foi o joystick, que é capaz de converter um sinal analógico (contínuo) em digital (discreto), por meio do seu potenciômetro. Do ponto de vista de controle eletrônico, essa peculiaridade se traduz no controle do vetor velocidade com a qual o efetuador deve ser movimentar, ou controlar individualmente cada motor, escolhendo sentido e velocidade de rotação.

Porém, como foi apresentado anteriormente, independentemente do método de planejamento escolhido, é necessário calcular a matriz jacobiana de velocidades, e sua inversa. No entanto, essa tarefa se mostrou complexa, dado que há mais variáveis de junta do que coordenadas escalares do espaço de trabalho tridimensional.

Observando construtivamente o manipulador e as restrições impostas para a cinemática inversa e controle, percebeu-se que a velocidade do ponto ocupado pela junta  $J_4$  (ponto Q), velocidade cartesiana, será a mesma do efetuador terminal (ponto P), pois há apenas uma translação entre ambos os pontos. Dessa maneira, é possível calcular o jacobiano e controlar a velocidade do ponto Q.

Para calcular o jacobiano é necessário ter as coordenadas do ponto Q, fornecida pelo vetor  $a_{14}: a_{34}$  da matriz de transformação homogênea  ${}^3_0A$ , obtida pela multiplicação a direita das Equações (1) a (3).

$${}^3_0A = {}^1_0A_1^2 A_2^3 A \quad (54)$$

$$Q = \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} = \begin{pmatrix} a_2 \cos(\theta_1) \cos(\theta_2) + a_3 \cos(\theta_1) [\cos(\theta_2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)] \\ a_2 \cos(\theta_2) \sin(\theta_1) + a_3 \sin(\theta_1) [\cos(\theta_2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)] \\ d_1 + a_2 \sin(\theta_2) + a_3 [\cos(\theta_2) \sin(\theta_3) + \cos(\theta_3) \sin(\theta_2)] \end{pmatrix} \quad (55)$$

Da Equação 31, tem-se:

$$J_v(\mathbf{q}) = \begin{bmatrix} \frac{\partial Q_x}{\partial \theta_1} & \frac{\partial Q_x}{\partial \theta_2} & \frac{\partial Q_x}{\partial \theta_3} \\ \frac{\partial Q_y}{\partial \theta_1} & \frac{\partial Q_y}{\partial \theta_2} & \frac{\partial Q_y}{\partial \theta_3} \\ \frac{\partial Q_z}{\partial \theta_1} & \frac{\partial Q_z}{\partial \theta_2} & \frac{\partial Q_z}{\partial \theta_3} \end{bmatrix} \quad (56)$$

Cada elemento da Equação 56 foi calculado, e obteve-se:

$$\frac{\partial Q_x}{\partial \theta_1} = -a_2 \sin(\theta_1) \cos(\theta_2) - a_3 \sin(\theta_1) [\cos(\theta_2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)] \quad (57)$$

$$\frac{\partial Q_x}{\partial \theta_2} = -a_2 \sin(\theta_2) \cos(\theta_1) - a_3 \cos(\theta_1) [\sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3)] \quad (58)$$

$$\frac{\partial Q_x}{\partial \theta_3} = -a_3 \cos(\theta_1) [\sin(\theta_3) \cos(\theta_2) + \cos(\theta_3) \sin(\theta_2)] \quad (59)$$

$$\frac{\partial Q_y}{\partial \theta_1} = a_2 \cos(\theta_2) \cos(\theta_1) + a_3 \cos(\theta_1) [\cos(\theta_2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)] \quad (60)$$

$$\frac{\partial Q_y}{\partial \theta_2} = -a_2 \sin(\theta_2) \sin(\theta_1) - a_3 \sin(\theta_1) [\sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3)] \quad (61)$$

$$\frac{\partial Q_y}{\partial \theta_3} = -a_3 \sin(\theta_1) [\sin(\theta_3) \cos(\theta_2) + \cos(\theta_3) \sin(\theta_2)] \quad (62)$$

$$\frac{\partial Q_z}{\partial \theta_1} = 0 \quad (63)$$

$$\frac{\partial Q_z}{\partial \theta_2} = a_2 \cos(\theta_2) + a_3 [-\sin(\theta_2) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3)] \quad (64)$$

$$\frac{\partial Q_z}{\partial \theta_3} = a_3 [\cos(\theta_2) \cos(\theta_3) - \sin(\theta_2) \sin(\theta_3)] \quad (65)$$

Para obter o jacobiano inverso de velocidades, utilizou-se o próprio MATLAB, que inverteu a matriz desejada de maneira simbólica. O programa utilizado encontra-se no Apêndice I. Adiante, nas simulações, essa matriz adquire valores numéricos.

Com o jacobiano inverso de velocidades, é possível, a partir de velocidades escalares do efetuador, no caso do ponto Q, obter a velocidade das juntas para realizar o movimento desejado. Essas velocidades escalares, como apresentado anteriormente, podem ser fornecidas pela leitura da posição do joystick.

Assim, com o intuito de dar mais fluidez e agilidade ao movimento do robô, além de permitir uma operação centrada no usuário e permitindo o trabalho em ambientes desestruturados, as alternativas de interpolação da trajetória, controle ponto a ponto e trajetória contínua foram descartadas, e optou-se por controlar o vetor velocidade do ponto Q através do joystick. Dessa forma, o manipulador será teleguiado.

Por fim, para finalizar a etapa algébrica de planejamento da trajetória, a matriz encontrada deve ser inserida na Equação 36, sendo que cada elemento do vetor  $v_i$  são as velocidades fornecidas pelo joystick, respectivamente  $v_x$ ,  $v_y$  e  $v_z$ :

$$v_i = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad (66)$$

O resultado obtido irá retornar as respectivas equações para os componentes do vetor  $\dot{q}$ . O programa utilizado encontra-se no Apêndice I.

Há, então, o desafio de implementar essa solução, utilizando o Arduino e o MATLAB. A primeira atividade deve ser identificar qual a posição atual ( $P_0$ ) das juntas, ou seja, cada valor de  $\theta$ . Essa compreensão é importante pois o valor numérico das respectivas velocidades de juntas é obtido aplicando essa posição no jacobiano inverso de velocidades.

Logo em seguida, deve-se captar e armazenar as componentes da velocidade escalar, que são obtidas pela leitura dos potenciômetros do joystick. No plano XY do joystick, por exemplo, a movimentação deste cria uma diagonal (hipotenusa), e os potenciômetros de cada eixo assumem valores para os catetos, correspondendo à hipotenusa gerada. Na Figura 71, os valores de X e Y são os do potenciômetro de cada eixo.

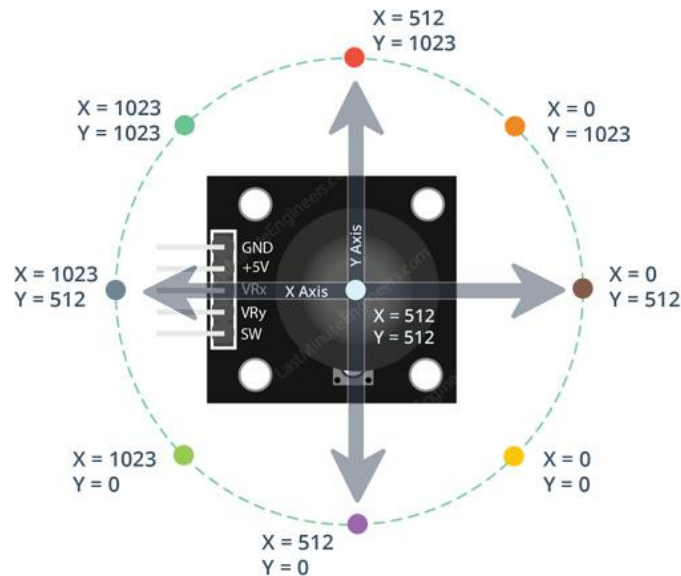


Figura 71 - Trigonometria de um joystick de 2 eixos (Last Minute Engineers, 2019)



Já no eixo Z de um joystick simples (Figura 47), a leitura é feita de maneira binária, pelo *push botton* existente neste eixo. Há também joystick que usa movimento rotativo para movimentação ao longo do eixo Z.

Concomitantemente a leitura dos potenciômetros, é necessário calcular a matriz jacobiana e a jacobiana inversa para a posição  $P_0$ . Com esta em mãos, deve-se aplicar os dados obtidos na Equação 36, a fim de encontrar as velocidades das juntas.

Com isso, deve-se enviar esses valores para as placas de acionamento dos motores, durante o tempo determinado e garantir que a condição imposta para  $\theta_4$  está sendo respeitada.

Paralelo a todas essas etapas, é interessante ter o vetor posição cartesiano, dado pelo vetor  $a_{13}$ :  $a_{34}$  extraído da matriz de transformações homogêneas  ${}^4_0A$  de cinemática direta de posição.

Por fim, para garantir a continuidade do processo, é preciso atualizar o valor de  $P_0$  com os novos valores das variáveis de junta. Assim, o processo iterativo se repete até que o acionamento do joystick seja interrompido.

Esquemáticamente, o fluxo de tarefas apresentado acima é representado na Figura 72:

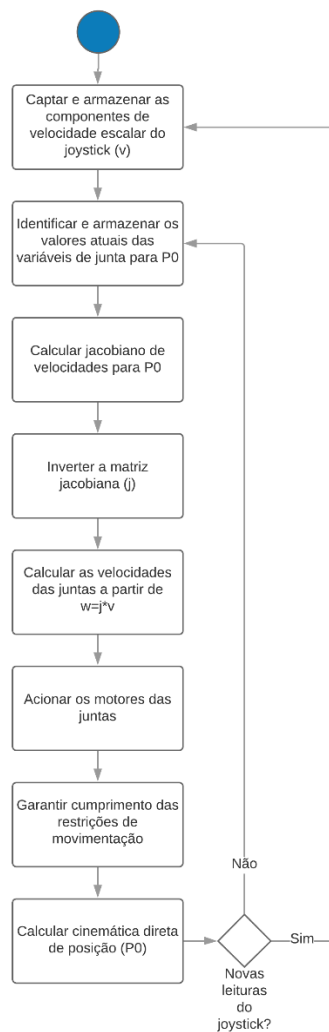


Figura 72 - Esquema de como implementar o planejamento da trajetória

## 5 TESTES E VALIDAÇÃO

As formulações desenvolvidas no capítulo anterior indicam a viabilidade teórica da solução proposta. Porém, diversos fatores podem interferir na utilização dessa solução no manipulador, como inércia, folgas, arredondamento de contas, acúmulo de erros, etc. Por conta disso, é necessário testar e validar a proposta de solução.

Dadas as propostas feitas na solução conceitual, a implementação da solução deve ser capaz de controlar, movimentando um joystick, a posição e a trajetória percorrida pelo efetuador terminal do manipulador (movimento cartesiano), além de manipular objetos e realizar tarefas cotidianas.

A Figura 72 indica a utilização de duas plataformas, o Arduino e o MATLAB. Essas plataformas precisam dialogar, tanto na execução das contas necessárias, como na leitura e envio de sinais entre si e para os atuadores.

Além disso, é importante levar em consideração que o ponto de equilíbrio do joystick é em  $X = 512$  e  $Y = 512$  (Figura 71), o que define um outro problema: criar uma escala de intensidade (velocidade) e quantidade de passos (deslocamento) para relacionar a leitura do joystick com o acionamento dos motores pelo Arduino.

Para isso, as velocidades foram divididas em positivas (rotação dos motores no sentido anti-horário, leitura do *joystick*  $> 512$ ) e negativas (rotação dos motores no sentido horário e leitura do *joystick*  $< 512$ ). Além disso, para controle de velocidade de movimentação, reduzindo a sensibilidade do controlador, haverá apenas duas velocidades: velocidade normal e velocidade rápida. Essas duas velocidades foram definidas, para testes iniciais, com as respectivas intensidades:  $20\text{ mm/s}$  e  $50\text{ mm/s}$ . Mesmo com apenas duas velocidades, sua combinação com os sentidos de rotação fornece 16 possibilidades para movimentação do manipulador, no plano  $XY$  por exemplo.

Tabela 9 - Leitura do joystick e velocidade do efetuador

Leitura Joystick	Velocidade normal (256 a 768)	Velocidade rápida ( $< 256$ e/ou $> 768$ )
Sentido horário ( $< 512$ )	$20\text{ mm/s}$	$20\text{ mm/s}$
Sentido anti-horário ( $> 512$ )	$50\text{ mm/s}$	$50\text{ mm/s}$

No Arduino, diferentes velocidades são traduzidas por diferentes *delays* (atrasos, intervalos, em milissegundos) entre o chaveamento das bobinas dos motores de passo. É importante estabelecer uma relação entre a velocidade fornecida pela Equação 36, em radianos por segundo, e o *delay* dos motores de passo. Motores de passo do tipo 28BYJ-48, como os utilizados no teste, dão 64 passos por volta, sendo que uma volta equivale a  $2\pi$  radianos. Seja  $\omega$  a velocidade fornecida pela Equação 36, então:

$$\omega \left[ \frac{\text{passos}}{s} \right] = \frac{64 \times w \left[ \frac{\text{rad}}{s} \right]}{2\pi} \quad (67)$$

Como *delay* é uma fração de segundos (milissegundos), basta converteres o resultado acima da seguinte forma:

$$\text{delay}[ms] = \left( \frac{1}{\omega \left[ \frac{\text{passos}}{s} \right]} \right) \times 1000 \quad (68)$$

É necessário também entender por quanto tempo ou “quantos chaveamentos” deverão ser realizados. Para isso, foi determinado que esse processo deve acontecer até que se tenha novos valores lidos no *joystick*, ou que a leitura indique repouso ( $X = Y = 512$ ), cessando todos os movimentos. Na simulação, no entanto, foi definindo um intervalo durante o qual o programa iria rodar, ou seja, por quanto tempo o loop vai acontecer e uma nova leitura do *joystick* será realizada.

A parte eletrônica do manipulador estudado (Néto, 2019) não estava pronta, o que inviabilizou os testes nele. Todavia, a implementação deve seguir 3 processos independentes: definição da orientação da  $J_5$  e acionamento do motor associado; um grande loop de leitura do joystick, cálculo do jacobiano inverso de velocidades, acionamento dos motores e retroalimentação do processo; além de controle da garra (agarrar e soltar). Esse algoritmo é representado na Figura 73.

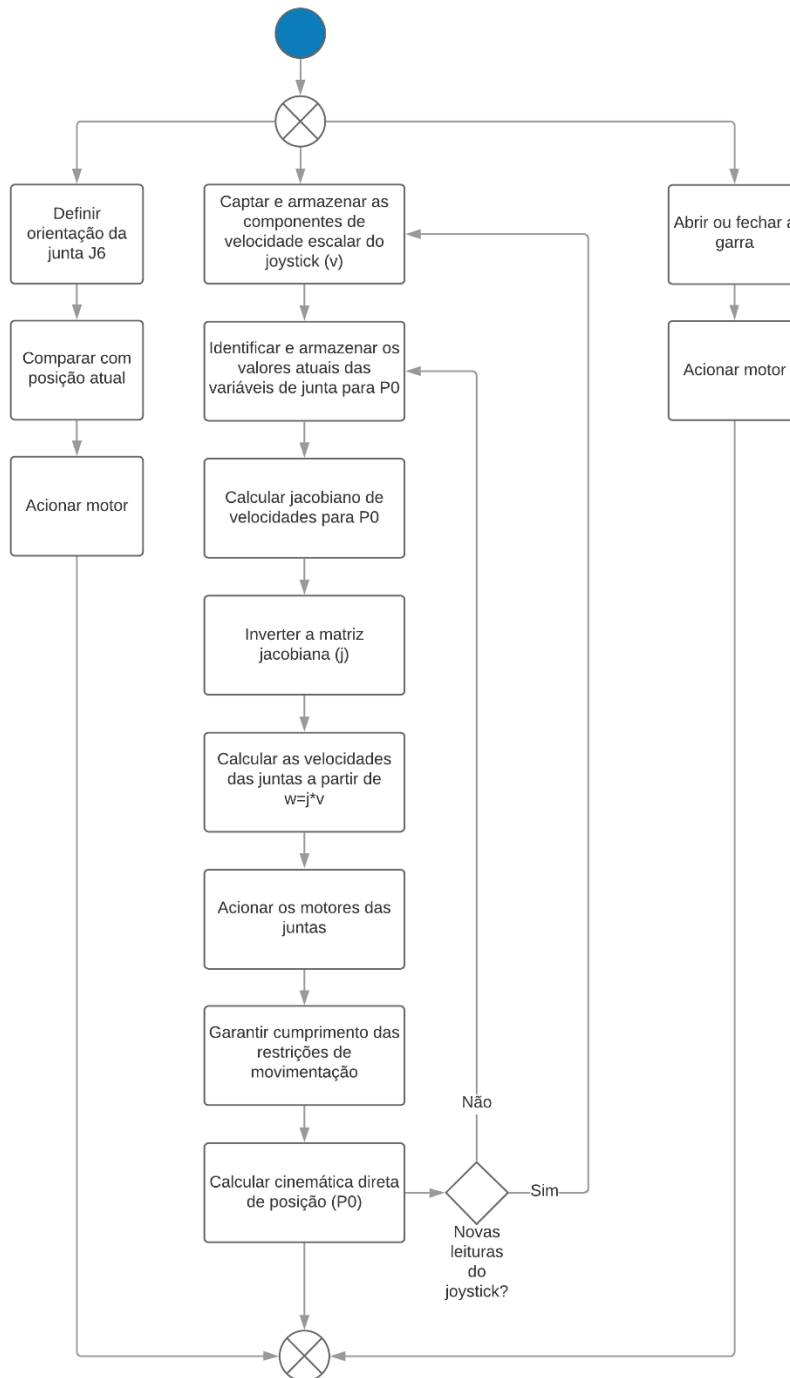


Figura 73 - Algoritmo de implementação do controle cartesiano do efetuador no manipulador real

Importante pontuar que, para essa leitura de posição atual dos atuadores, é necessário que a parte eletrônica tenha sensores capazes de identificar a informação desejada. No teste explicado a seguir, será usado um contador para contar quantos passos foram dados por cada motor e, assim, calcular suas novas posições.

A fim de validar o desenvolvimento realizado neste trabalho, a solução foi então testada para um manipulador planar de 2 GDL (Figura 74), operando no plano XY.

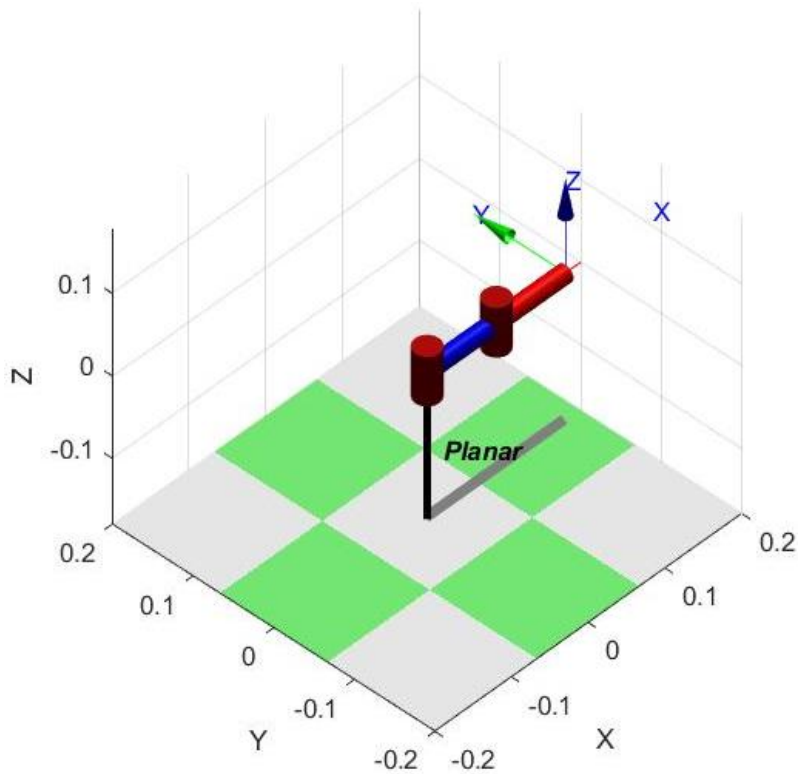


Figura 74 - Manipulador planar para testes

O robô apresentado é o mesmo da Figura 59 e possui os seguintes parâmetros D-H:

Tabela 10 - Parâmetros D-H do robô planar

Junta	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1 *$	0	$a_1 = 90 \text{ mm}$	0
2	$\theta_2 *$	0	$a_2 = 90 \text{ mm}$	0

Sua cinemática direta por ser obtida de maneira geométrica, por sua simplicidade:

$$\begin{cases} x = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) \\ y = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) \end{cases} \quad (69)$$

Mas também pode ser obtida utilizando os parâmetros da Tabela 10, fornecendo a matriz de transformações homogênea:

$${}^2_0A = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (70)$$

Sua cinemática inversa foi calculada de maneira geométrica também, resultando em:

$$\begin{cases} \theta_1 = \text{tg}^{-1} \left[ \frac{y[a_1+a_2 \cos(\theta_2)]-xa_2 \sin(\theta_2)}{x[a_1+a_2 \cos(\theta_2)]+ya_2 \sin(\theta_2)} \right] \\ \theta_2 = \pm \cos^{-1} \left[ \frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2} \right] \end{cases} \quad (71)$$

Dessa cinemática inversa, é importante ressaltar que, a depender do quadrante no qual o efetuador se encontra, as juntas podem assumir valores diferentes, assim como um mesmo ponto pode ser alcançado com posições diferentes das juntas (cotovelo para cima e/ou para baixo).

Em relação aos quadrantes, garantindo que o robô irá operar apenas nos quadrantes superiores do círculo trigonométrico (quadrantes 1 e 2), a tangente deverá ser corrigida da seguinte forma:

- Se  $x \geq 0$ , nenhuma correção deverá ser feita;
- Se  $x < 0$ , ao valor encontrado para  $\theta_1$  deverá ser acrescido  $\pi$ .

A matriz jacobiana e jacobiana inversa são, respectivamente:

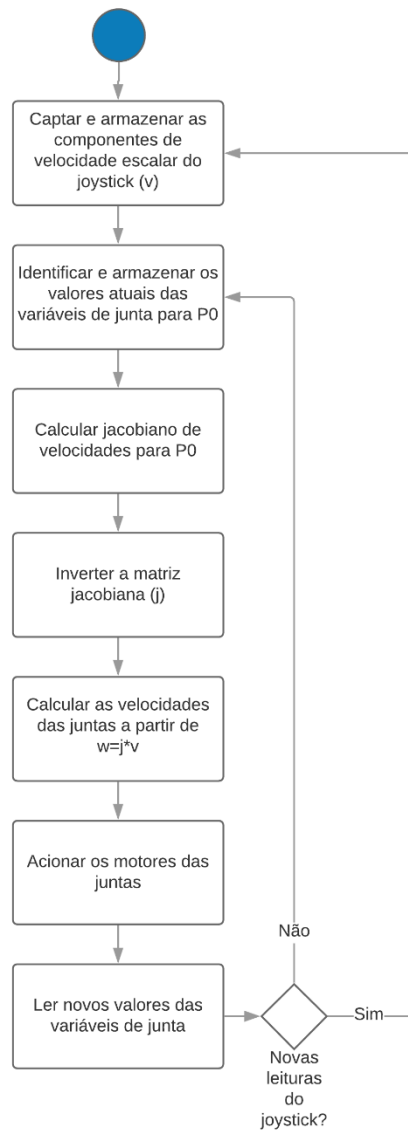
$$J = \begin{bmatrix} -a_1 \sin(\theta_1) - a_2 \sin(\theta_1 + \theta_2) & -a_2 \cos(\theta_1 + \theta_2) \\ a_1 \cos(\theta_1) + a_2 \sin(\theta_1 + \theta_2) & a_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (72)$$

$$J^{-1} = \frac{1}{a_2 a_1 [\cos(2\theta_1 + \theta_2)] + a_2^2 [\cos(\theta_1 + \theta_2) \sin(\theta_1 + \theta_2) - \sin^2(\theta_1 + \theta_2)]} \begin{bmatrix} a_2 \sin(\theta_1 + \theta_2) & a_2 \cos(\theta_1 + \theta_2) \\ -a_1 \cos(\theta_1) - a_2 \sin(\theta_1 + \theta_2) & -a_1 \sin(\theta_1) - a_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (73)$$

O software utilizado (MATLAB) não compreendeu a transformação algébrica da Equação 74 e, por isso, esta não foi feita na Equação 73.

$$\cos(\theta_1) \cos(\theta_1 + \theta_2) - \sin(\theta_1) \sin(\theta_1 + \theta_2) = \cos(2\theta_1 + \theta_2) \quad (74)$$

Uma vez que o manipulador de testes possui apenas 2 GDL, apenas os eixos  $X$  e  $Y$  do joystick serão utilizados. Ainda, o diagrama da Figura 73 teve de ser adaptado para calcular as velocidades de apenas duas juntas. O novo diagrama se reduz ao grande loop e é ilustrado na Figura 75.



*Figura 75 - Algoritmo para implementação do controle cartesiano do efetuador no manipulador planar*

Antes de implementar efetivamente, integrando Arduino e MATLAB, um programa de simulação foi desenvolvido nesta plataforma, explicado na Figura 76:

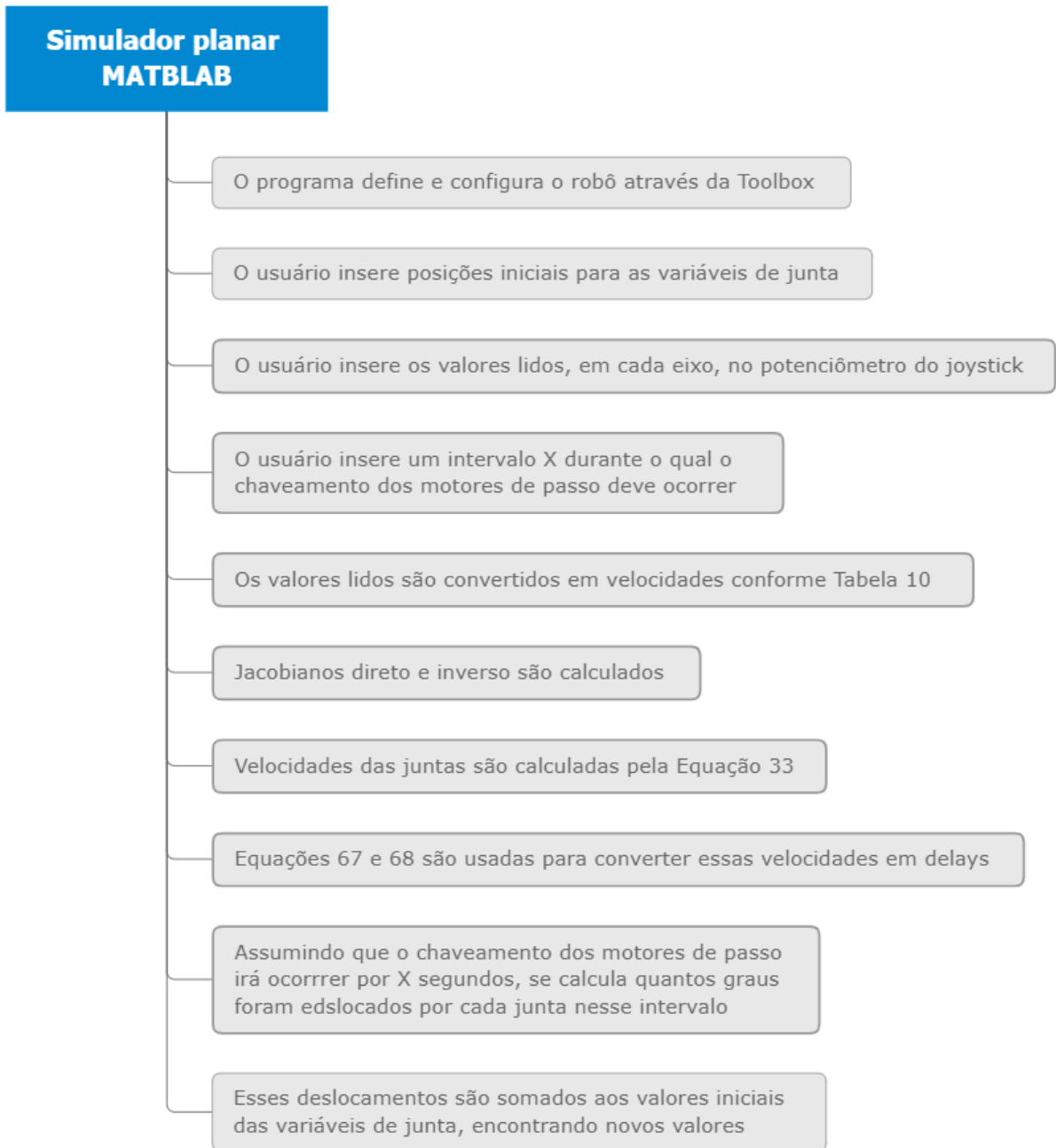
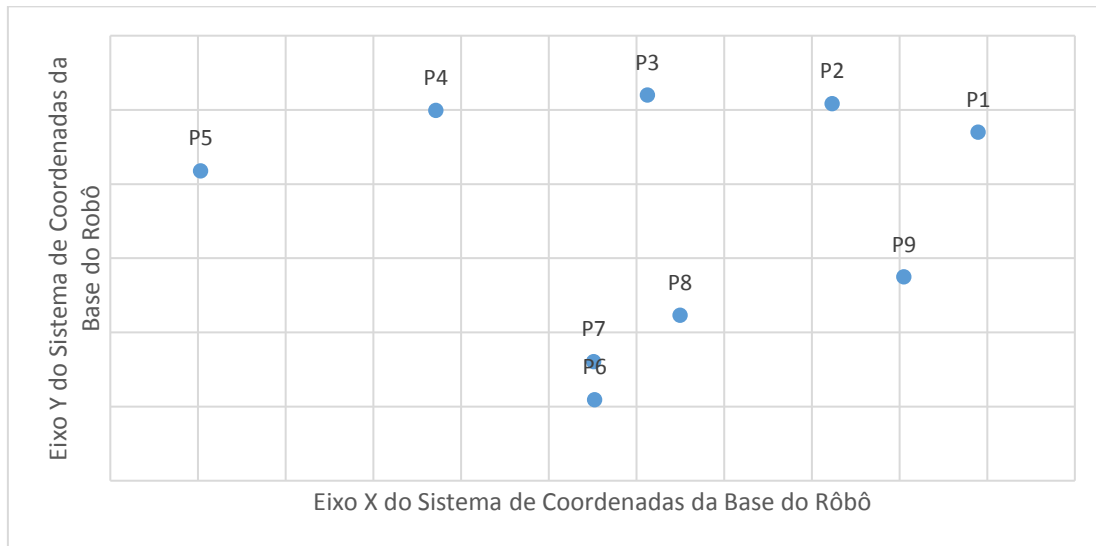


Figura 76 - Diagrama de simulação da solução no MATLAB para o robô planar

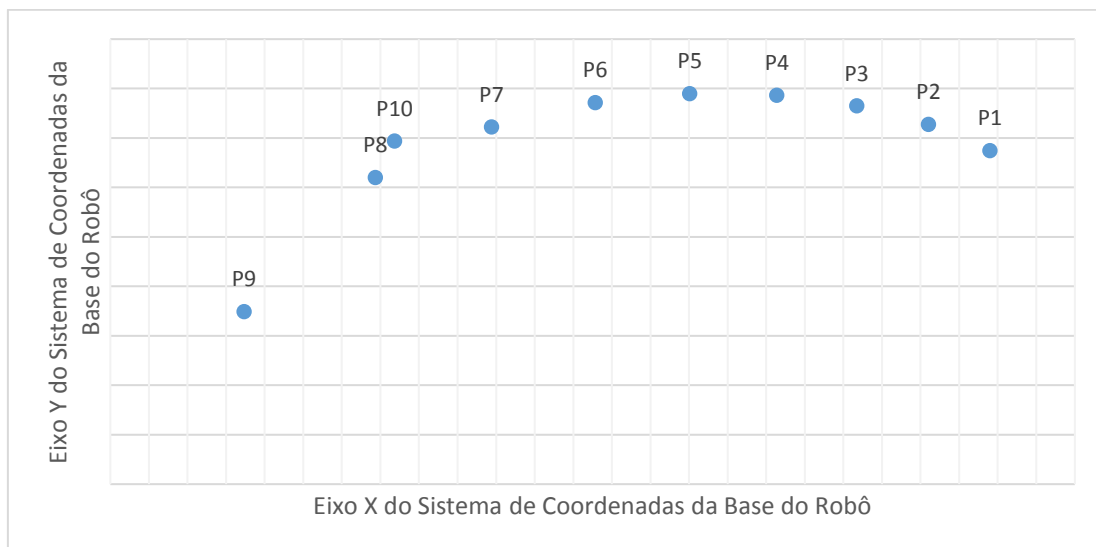
Inicialmente usou-se como posição inicial do manipulador  $\theta_1 = \frac{\pi}{6}$  e  $\theta_2 = \frac{\pi}{3}$ , a um deslocamento constante com o joystick na posição 300 e 520, para  $X$  e  $Y$ , respectivamente. O programa de simulação foi rodado nove vezes, os valores de  $\theta$  foram capturados e, pela Equação 64, a cinemática direta de posição foi calculada e o seguinte gráfico, de trajetória real, foi obtido, sendo  $P1$  o ponto inicial e  $P9$  a última posição do efetuador e um intervalo de 2 segundos entre cada ponto:





*Figura 77 - Trajetória Real do efetuador na primeira simulação (2 segundos)*

Tendo em vista o resultado inesperado acima, com um salto do ponto *P5* para o *P6*, além da curvatura da trajetória, a priori retilínea, o intervalo entre cada ponto foi reduzido para 1 segundo:



*Figura 78 - Trajetória Real do efetuador na segunda simulação (1 segundo)*

Embora o salto de descontinuidade só apareça, agora, a partir do ponto *P10*, nota-se que a trajetória começa a se curvar entre *P2* e *P5*.

Foi então verificado se os pontos se encontram fora do volume de trabalho, o que, para o manipulador planar, representa uma distância entre o ponto e a origem menor que  $0,18\text{ mm}$ . Dessa análise concluiu-se que todos os pontos, tanto na primeira quanto na segunda simulação, se encontram dentro do volume de trabalho. Foi então levantada a hipótese de que a descontinuidade ocorra devido a um acúmulo de erros, a iniciar pela curvatura da trajetória.

Com os valores das componentes da velocidade do efetuador são conhecidos (Tabela 9), assim como o ponto inicial e os demais valores de X, pôde-se construir a reta que define a trajetória teórica (em laranja). Esta foi então plotada junto à discretização real da Figura 78:

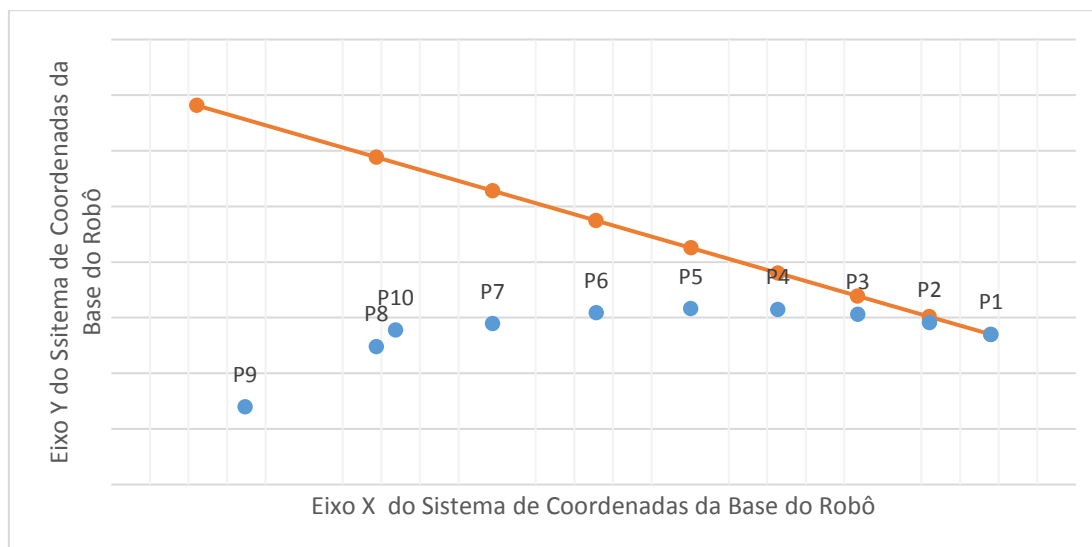


Figura 79 - Trajetória Real versus Teórica do efetuador (1 segundo)

A mesma verificação de volume de trabalho foi realizada para a trajetória teórica, e notou-se que no ponto *P3* já se aproxima dos limites (distante da origem  $17,5\text{ mm}$ ), que indica uma singularidade. Estudando a reta teórica, o limite do volume de trabalho será atingido, para as condições de trabalho definidas, em  $X; Y = (0,037; 0,176)\text{ m}$ . Com isso, foi inserido um aviso no programa de “Proximidades de uma singularidade”, a fim de alertar o usuário. Esse aviso, irá aparecer quando a distância entre o efetuador e a origem for maior ou igual a  $16\text{ mm}$  e/ou menor ou igual de  $3\text{ mm}$ .

Assim, com a solução deste problema, o algoritmo foi testado para posições iniciais em quadrantes diferentes do círculo trigonométrico.

Para o segundo quadrante, com uma posição inicial de  $\theta_1 = 150^\circ$  e  $\theta_2 = -90^\circ$  e  $X; Y = (-0,0329; 0,1229)\text{ m}$ , e leituras constantes no joystick de 800 e 512 em X e Y respectivamente, espera-se que o efetuador realize uma trajetória horizontal no sentido positivo do eixo X. Porém, notou-se o seguinte comportamento:

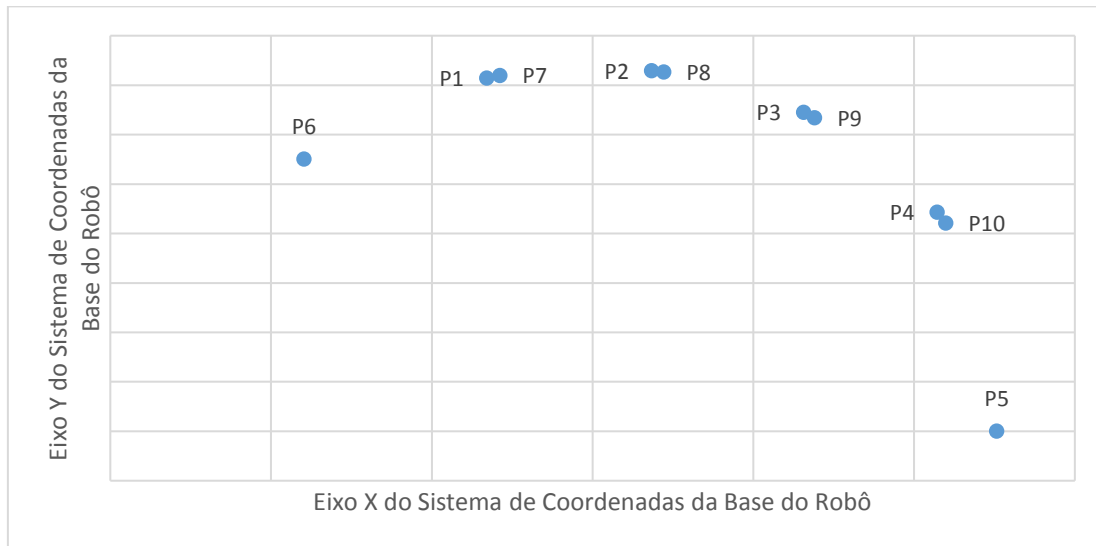


Figura 80 - Trajetória Real 2º Quadrante (1 segundo)

Novamente não é a trajetória esperada. Para tentar solucionar, optou-se por diminuir o intervalo de execução para 0,5 s. Dessa vez, o resultado foi melhor que o anterior:

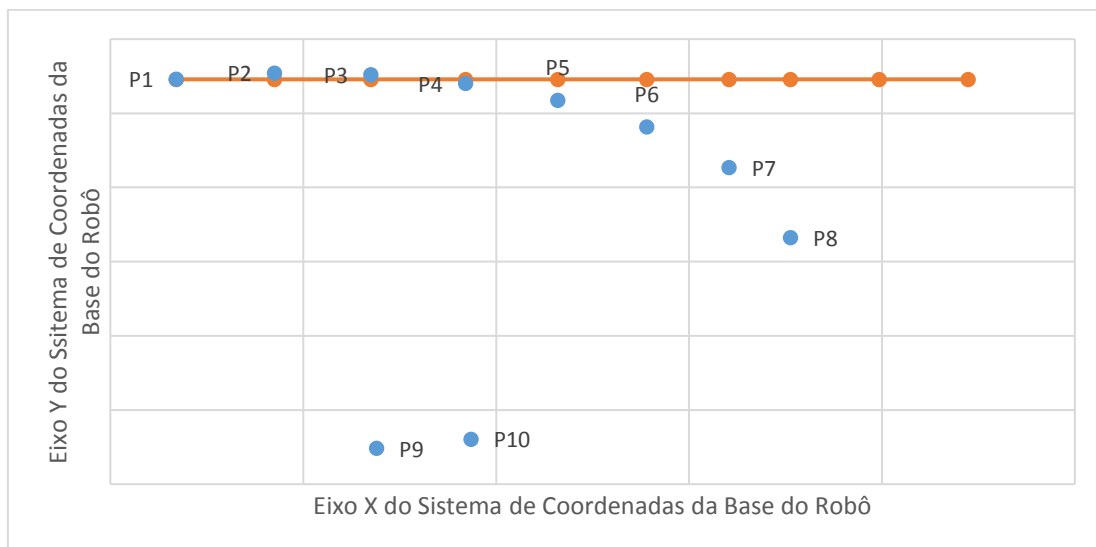


Figura 81 - Trajetória Real versus Teórica 2º Quadrante (0,5 segundo)

Contudo, o resultado acima ainda chamou atenção para o distanciamento progressivo da trajetória real e relação à teórica, e verificou-se que é necessária uma velocidade “corretiva”, a fim de minimizar o erro que é acumulado ao longo das iterações.

Analisando o erro que aparece no Eixo Y, obteve-se, para as mesmas condições da Figura 81, os seguintes resultados:

Tabela 11 – Erro (Teórico-Real) das coordenadas Y para o 2° Quadrante

Iteração	Cinemática Direta		Reta Teórica		Delta Y (Teórica – Real) (mm)
	X (mm)	Y (mm)	X (mm)	Y (mm)	
<i>P1</i>	-0,0329	0,1229	-0,0329	0,1229	0,0000
<i>P2</i>	-0,0074	0,1271	-0,0074	0,1229	-0,004,2
<i>P3</i>	0,0176	0,1261	0,0176	0,1229	-0,0032
<i>P4</i>	0,0420	0,1201	0,0420	0,1229	0,0028
<i>P5</i>	0,0660	0,1089	0,0660	0,1229	0,0140
<i>P6</i>	0,0890	0,0910	0,0890	0,1229	0,0319
<i>P7</i>	0,1104	0,0634	0,1104	0,1229	0,0595
<i>P8</i>	0,1263	0,0161	0,1263	0,1229	0,1068
<i>P9</i>	0,0190	-0,1258	0,1493	0,1229	0,2487
<i>P10</i>	0,0435	-0,1196	0,1724	0,1229	0,2425

Com esses dados, foi verificada também a de quanto é o erro entre um ponto e outro, ou seja, o erro relativo:

Tabela 12 - Erro relativo entre os pontos reais para o 2° Quadrante

Iteração	Delta Y <sub>(i)-(i-1)</sub> (mm)
<i>P1</i>	0,0000
<i>P2</i>	-0,0041
<i>P3</i>	0,0010
<i>P4</i>	0,0059
<i>P5</i>	0,0113
<i>P6</i>	0,0179
<i>P7</i>	0,0276
<i>P8</i>	0,0473
<i>P9</i>	0,1420
<i>P10</i>	-0,0062

Analisando cuidadosamente os dados da Tabela 12, nota-se que os erros relativos se aproximam da soma, em módulo, dos dois erros anteriores, ou seja, o erro vai se acumulando, como previsto anteriormente. Em outras palavras, o manipulador, a cada novo ponto, segue uma reta diferente da prevista.

Para correção desses erros, propõe-se então a adição, no algoritmo da simulação, de comandos que comparem o coeficiente angular original da reta original com a recém-executada, e corrija, modificando a velocidade e sentido real de deslocamento do efetuador, a fim deste constantemente buscar a reta teórica, até que o valor lido dos potenciômetros do joystick mude.

Analisando cada caso, tanto para erros no eixo  $X$  quando no  $Y$ , tem-se que:

- Quando a posição real do efetuador se encontra adiantada em relação à teórica, a velocidade  $V_i$  deve ter seu módulo diminuído proporcionalmente;
- Quando a posição real do efetuador se encontra atrasada em relação à teórica, a velocidade  $V_i$  deve ter seu módulo aumentado proporcionalmente;

O algoritmo, então, foi reescrito:

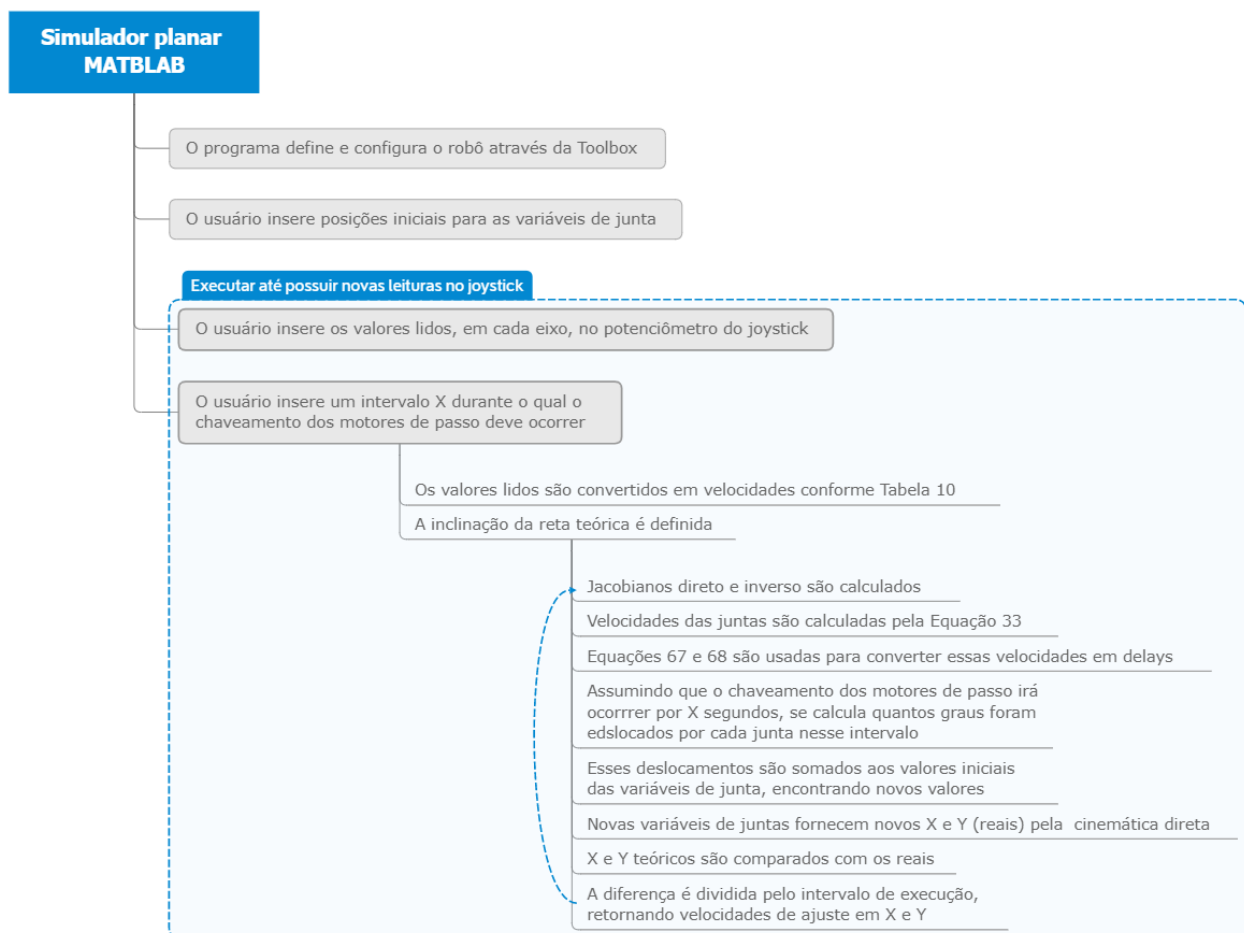


Figura 82 - Algoritmo para correção de erros e simulação da solução no MATLAB em um robô planar

Com esse novo algoritmo (Apêndice II), resultados interessantes foram obtidos, para o mesmo intervalo de 0,5 s, posição inicial no segundo quadrante ( $\theta_1 = 150^\circ$  e  $\theta_2 = -90^\circ$ ) e valores lidos nos potenciômetros do joystick para  $X$  e  $Y$  de, respectivamente, 800 e 600:

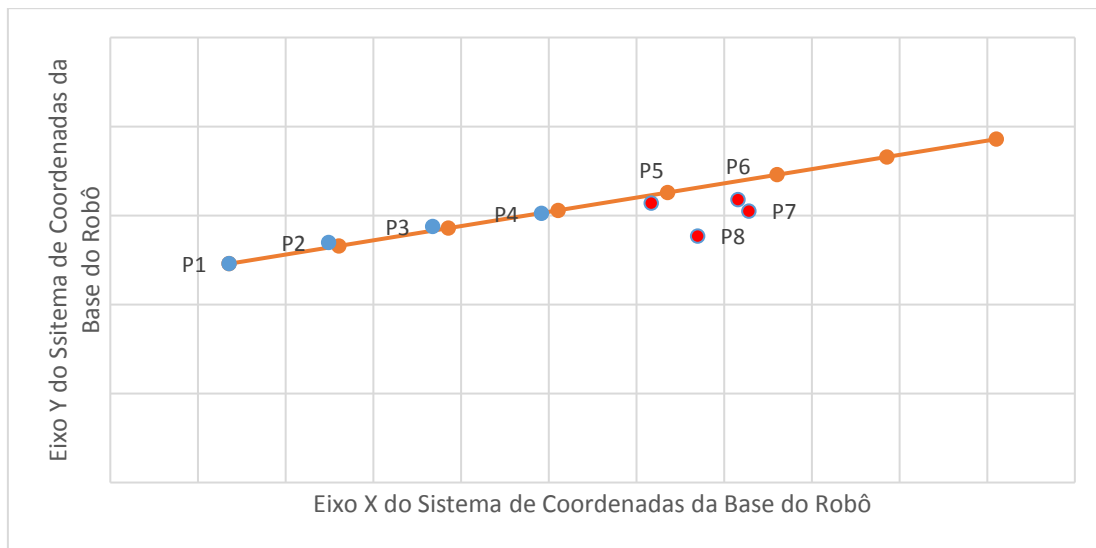


Figura 83 - Trajetória Real versus Teórica no 2º Quadrante (0,5 segundo), com o algoritmo de correção

A partir do ponto  $P5$  o programa acusou aproximação dos limites do volume de trabalho. Porém, entre os pontos  $P1$  e  $P5$  os resultados foram satisfatórios, considerando um deslocamento do efetuador no sentido ascendente da reta teórica (em laranja).

O intervalo de execução foi reduzido então para 0,25 segundo, a fim de possuir mais pontos calculados antes de alcançar uma singularidade, e a nova trajetória é representada pela Figura 84:

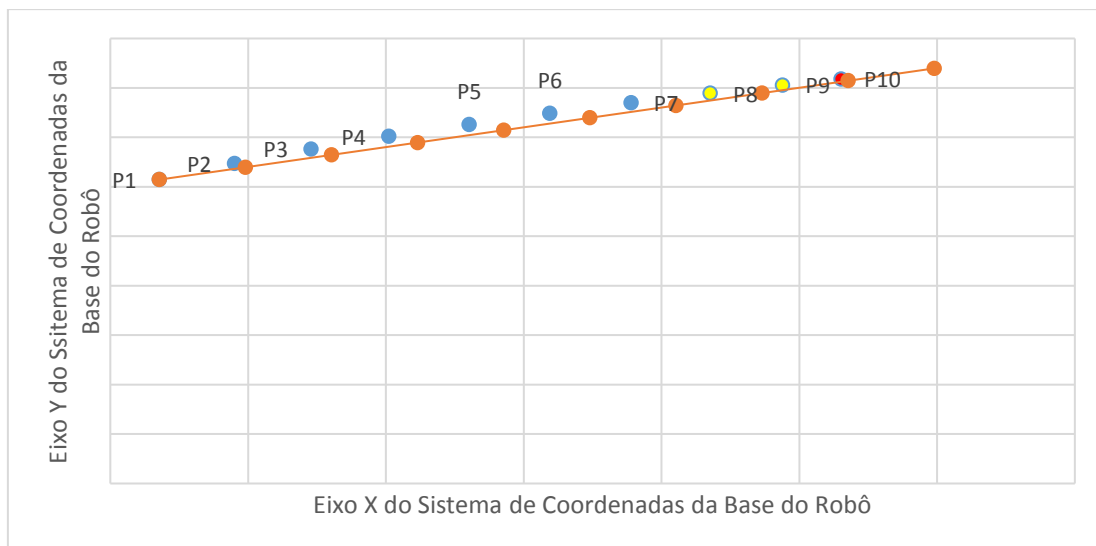


Figura 84 - Trajetória Real versus Teórica no 2º Quadrante (0,25), com o algoritmo de correção

Tabela 13 - Erro (Teórico-Real) das coordenadas Y para o 2º Quadrante após correção

Iteração	Cinemática Direta		Reta Teórica		Delta Y (Teórica – Real) (mm)
	X (mm)	Y (mm)	X (mm)	Y (mm)	
<i>P1</i>	-0,0329	0,1229	-0,0329	0,1229	0,0000
<i>P2</i>	-0,0220	0,1294	-0,0204	0,1279	-0,0015
<i>P3</i>	-0,0109	0,1352	-0,0079	0,1329	-0,0023
<i>P4</i>	0,0004	0,1404	0,0046	0,1379	-0,0025
<i>P5</i>	0,0121	0,1452	0,0171	0,1429	-0,0023
<i>P6</i>	0,0238	0,1497	0,0296	0,1479	-0,0018
<i>P7</i>	0,0356	0,1539	0,0421	0,1529	-0,0010
<i>P8</i>	0,0471	0,1578	0,0546	0,1579	0,0001
<i>P9</i>	0,0576	0,1611	0,0671	0,1629	0,0018
<i>P10</i>	0,0661	0,1636	0,0796	0,1679	0,0043

Pelos dados apresentados na Tabela 13, percebe-se que o algoritmo tem um tempo para reação, tendo em vista as correções proporcionais, e cumpre a tarefa de buscar a reta teórica e diminuir o erro (Delta Y).

Dessa maneira, os resultados apresentados para a simulação verificam e validam a sistemática de planejamento e geração da trajetória desenvolvida.

## 6 CONCLUSÃO

Este trabalho foi dedicado à construção do arcabouço teórico necessário para a realização do controle dos movimentos de um manipulador por usuários tetraplégicos, assim como o desenvolvimento de uma solução para controlá-lo a partir de um joystick como dispositivo de comando e interface com o usuário. Discorreu-se sobre as motivações para o projeto, sua importância, as áreas e conhecimentos envolvidos, além de todo o contexto em que se está inserido. No que tange aos conceitos e conhecimentos necessários, o presente trabalho buscou não apenas entender as soluções de controle e interface homem-máquina existentes no mercado e na academia, mas também as possibilidades de aprimoramento e o que será necessário entender para atingir o objetivo posto do projeto, sob o amparo das premissas definidas. Além disso, o trabalho buscou analisar as alternativas de controle com uma ótica respeitosa em relação ao paciente e às suas necessidades e limitações, e propor uma solução condizente, assim como alguns cálculos e definições preliminares, sendo utilizar de um *joystick* controlado por alguns movimentos do pescoço do paciente a melhor alternativa encontrada. Por fim, os capítulos finais se dedicaram a desenvolver a solução proposta pelo autor, calculando a cinemática inversa, volume de trabalho e o planejamento de trajetórias, assim como os algoritmos necessários para implementação da solução, a priori realizada em um robô planar. Nos testes realizados percebeu que o erro encontrado era baixo em comparação às dimensões do robô, tornando a solução de telecontrole proposta satisfatória. Com isso, os objetivos de calcular a cinemática inversa e planejar e gerar trajetórias foram atendidos, dentro das limitações e comparações impostas.

### 6.1 TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se implementar no manipulador real, assim que sua eletrônica estiver finalizada; utilizar sensores para facilitar a leitura da posição atual das juntas e do próprio efetuador terminal; adicionar comandos mais finos de controle, como de movimentação da junta  $J_6$ , para a manipulação de objetos e rotação de maçanetas, por exemplo; inserir um controle mais fino da garra, tanto da posição quanto torque dos dedos da garra; e desenvolver um *design* ergonômico para a interface *joystick* e usuário. Com a implementação da solução no manipulador real, outras necessidades podem surgir, como suavização de trajetórias e aumento da velocidade de processamento, iteração e movimentação. Por fim, é interessante pensar em como tornar o controle mais inteligente, em um horizonte maior de tempo, como pré-carregando posições e implementando *machine learning* no dispositivo.



# REFERÊNCIAS BIBLIOGRÁFICAS

- (s.d.). Acesso em 15 de Junho de 2019, disponível em Ali Express:  
<https://pt.aliexpress.com/item/32856031894.html>
- (2 de Novembro de 2018). Acesso em 27 de Maio de 2019, disponível em Bluelux:  
<https://keycdn.bluelux.com.br/wp-content/uploads/contatos-1-768x459.png>
- ABRAFIN. (s.d.). *Desordens da Função Neurológica e Fisioterapia*. Acesso em 4 de Abril de 2019, disponível em Associação Brasileira de Fisioterapia Neurofuncional:  
<http://abrafin.org.br/desordens-da-funcao-neurologica-e-fisioterapia/>
- Addicore. (2016). *Duas-Axis XY Joystick Modulo woth Push Botton*. Acesso em 3 de Junho de 2019, disponível em Addicore: <https://cdn3.volusion.com/btfzd.umflq/v/vspfiles/photos/139-2.jpg?1437257596>
- Bersch, R. (2017). *Introdução à Tecnologia Assistiva*. Acesso em 20 de Setembro de 2018, disponível em [www.assistiva.com.br](http://www.assistiva.com.br): [http://www.assistiva.com.br/Introducao\\_Tecnologia\\_Assistiva.pdf](http://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf)
- Bolton, W. (1995). *Engenharia de Controle* (1 ed.). MAKRON Books.
- Botoeiras e Sinalizadores*. (s.d.). Acesso em 29 de Maio de 2019, disponível em Mundo da Elétrica:  
<https://www.mundodaeletrica.com.br/i/2219/botao-cogumelo-1024x682-620.webp>
- Botoeiras e Sinalizadores*. (s.d.). Acesso em 30 de Maio de 2019, disponível em Mundo da Elétrica:  
<https://www.mundodaeletrica.com.br/botoeiras-e-sinalizadores/>
- Botoeiras e Sinalizadores*. (s.d.). Acesso em 30 de Maio de 2019, disponível em Mundo da Elétrica:  
<https://www.mundodaeletrica.com.br/i/2219/chave-seletora-1024x682-620.webp>
- Botoeiras e Sinalizadores*. (s.d.). Acesso em 30 de Maio de 2019, disponível em Mundo da Elétrica:  
<https://www.mundodaeletrica.com.br/i/2219/botao-sem-retencao-1024x682-620.webp>
- Brites, F. G., & Santos, V. P. (Julho de 2008). *Motor de Passo*. Grupo PET-Tele. Niterói, Rio de Janeiro: UFF.
- Brooks, R. A. (1985). *A Robust Layered Control System for a Mobile Robot*. EUA.
- Cabeça e Face*. (2010). Acesso em 15 de Junho de 2019, disponível em Desenhe Tudo:  
<https://desenhetudo.blogspot.com/p/cabeça-e-face.html>
- Cabral, E. L. (2017). *Cinemática Direta*. Acesso em 10 de Junho de 2019, disponível em Eduardo Lobo Lustosa Cabral: <http://sites.poli.usp.br/p/eduardo.cabral/>
- Carrara, V. (2015). *Introdução à Robótica Industrial*. São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE).
- Collins, J. A. (2006). *Projeto Mecânico de Elementos de Máquinas*. LTC.
- Comandos Elétricos: o que são e onde são usados*. (2019). Acesso em 2 de Junho de 2019, disponível em Mundo da Elétrica: <https://www.mundodaeletrica.com.br/comando-eletricos-o-que-sao-e-onde-sao-usados/>
- Cook, & Hussey. (2007). *Assisitive Technologies: Principles and Practives* (3 ed.). Mosby.

- Corke, P. (Abril de 2001). Fonte: petercorke.com:  
<http://petercorke.com/wordpress/toolboxes/robotics-toolbox>
- Costa, E. R. (Dezembro de 2003). Simulação de Movimentos e Planejamento de Trajetória para Robôs Manipuladores. *Dissertação para obtenção do grau de mestre em engenharia mecânica*. Recife, Pernambuco: UFPE.
- Craig, J. (2012). *Robótica* (3ª ed.). São Paulo: Pearson Education do Brasil.
- Didi, N., Grandjean, B., Mokhtari, M., & Roby-Brami, A. (1998). An Assistive Control System to the manipulation of the MANUS Arm Robot. *RESNA'98*.
- Driessen, B. J., Evers, H. G., & Woerden, J. A. (2001). MANUS - A Wheelchair-mounted rehabilitation robot. *Journal of Engineering in Medicine*.
- Dynapar. (Novembro de 2016). *Dynapar: o que é Encoder?* Acesso em 28 de Abril de 2019, disponível em Dynapar: <https://www.dynaparencoders.com.br/blog/como-funciona-encoder/>
- Endeavor Brasil. (2015). *Comece pequeno, mas sonhe grande*. Acesso em 09 de Julho de 2019, disponível em Endeavor Baasil: <https://endeavor.org.br/sobre-a-endeavor/comece-pequeno-mas-sonhe-grande/>
- Erthal, J. L. (Outubro de 1992). Dissertação submetida à Universidade Federal de Santa Catarina para obtenção do grau de mestre em Engenharia Mecânica. *Métodos para a Solução Cinemática Inversa de robôs industriais para implementação computacional*. Florianópolis, Santa Catarina, Brasil.
- ESSEL Engenharia. (2019). *Cursos Essel*. Acesso em 20 de Abril de 2019, disponível em ESSEL Engenharia: <https://essel.com.br/cursos/material/05/scap2.pdf>
- Exact Dynamics. (2018). *iARM*. Acesso em 26 de Setembro de 2019, disponível em Exact Dynamics.
- Exame. (2015). *6 Lições de Jorge Paulo Lemann para empreendedores*. Acesso em 29 de Julho de 2019, disponível em Exame: <https://exame.abril.com.br/pme/6-licoes-de-jorge-paulo-lemann-para-empreendedores/>
- Fanti, C. E., Ferreira, W. R., & Carvalho, J. C. (Setembro de 2013). Planejamento de Trajetórias Robóticas definidas por Segmentos de Reta Concordantes por Polinomiais. UFU.
- Ferreira, C. L. (2008). Interface de Sopro de SUCção para controle de cadeira de rodas. *Dissertação de mestrado em Engenharia Elétrica*. Londrina, Paraná: Universidade Estadual de Londrina.
- Filho, P. M. (2019). Desenvolvimento de um efetuador terminal para um manipulador robótico assistivo. *PIBITI*. Brasília, Distrito Federal, Brasil: UnB.
- Gonzalez, C. (26 de Outubro de 2017). *What's the Difference Between Absolute and Incremental Encoders?* Acesso em 8 de Maio de 2019, disponível em Machine Design: [https://www.machinedesign.com/sites/machinedesign.com/files/Absolute-vs-Incremental-Encoder-Discs\\_0.jpg](https://www.machinedesign.com/sites/machinedesign.com/files/Absolute-vs-Incremental-Encoder-Discs_0.jpg)
- Groover, M. P. (1989). *Robótica - Tecnologia e Programação*. Mcgraw Hill.
- Gupta, K. C., & Roth, B. (1982). Design Considerations for Manipulador Workspace. *ASME Jnl of Mechanical Design*.

- Harris, T. (25 de Junho de 2002). *How Joysticks Work*. Acesso em 28 de Abril de 2019, disponível em How Stuff Works: <http://trizgyrus.com/trizgyrus/wp-content/howstuff/howstuff/Howstuffworks%20How%20Joysticks%20Work.htm>
- Hatakeyama, T., Kkoizumi, Y., Umeda, Y., Uchio, M., & Kasuga, M. (1998). Development of infrared light head pointing device for people with severe physical disability on manipulating personal computer. *RESNA'98*, p. 274.
- Hillman, M., Hagan, K., Hagan, S., Jepson, J., & Orpwood, R. (1999). A Wheelchair Mounted Assistive Robot. *ICORR'99: Internacional Conference on Rehabilitation Robotics*.
- Hongyu, K., & Saramago, S. d. (2008). Modelo para Cálculo do volume do espaço de trabalho de manipuladores 3R. *XII Seminário de Iniciação Científica*. Uberlândia: UFU.
- IBGE. (2012). *Censo Demográfico 2010*. Rio de Janeiro: IBGE.
- Isaacson, W. (2011). *Steve Jobs - A Biografia* (1 ed.). Companhia das Letras.
- Kalil, E. (15 de Setembro de 2011). *Lesões na coluna*. Acesso em 30 de Outubro de 2018, disponível em Drauzio Varella: <https://drauziovarella.uol.com.br/entrevistas-2/lesoes-na-coluna/>
- Kinova Technology. (s.d.). *KINOVA JACO Assistive Robotic Arm*. Acesso em 29 de Março de 2019, disponível em KINOVA: <https://www.kinovarobotics.com/en/products/assistive-robotics/jaco-assistive-robotic-arm>
- Klug, M. (2019). *Cinemática Diferencial de Manipuladores Seriais*. Santa Catarina: IFSC.
- Klug, M. (s.d.). *Cinemática Diferencial de Manipuladores Seriais*. Fonte: [http://www.joinville.ifsc.edu.br/~michael.klug/ROB74/Aulas/aula5\\_cin\\_diferencial.pdf](http://www.joinville.ifsc.edu.br/~michael.klug/ROB74/Aulas/aula5_cin_diferencial.pdf)
- Knight, W. (25 de Junho de 2018). Acesso em 19 de Abril de 2019, disponível em MIT Technology Review: <https://cdn.technologyreview.com/i/images/osaro-200-background-v3.jpg?sw=3000>
- Last Minute Engineers. (2019). Fonte: Last Minute Engineers: <https://lastminuteengineers.com/wp-content/uploads/2018/07/PS2-Joystick-Module-Movement-Analog-Values-on-Arduino.jpg>
- Learchannel-TV. (s.d.). *Como funciona um sensor resolver*. Acesso em 26 de Abril de 2019, disponível em Learchannel-TV: <http://learchannel-tv.com/pb/drives/servomotor/resolver/>
- Michael Hillman, K. H. (1999). A Wheelchair Mounted Assistive Robot. *ICORR'99: Internacional Conference on Rehabilitation Robotics*.
- Ministério da Saúde. (2013). *Diretrizes de Atenção à Pessoa com Lesão medular*. Brasília. Acesso em 28 de Março de 2019, disponível em [http://bvsms.saude.gov.br/bvs/publicacoes/diretrizes\\_atencao\\_pessoa\\_lesao\\_medular.pdf](http://bvsms.saude.gov.br/bvs/publicacoes/diretrizes_atencao_pessoa_lesao_medular.pdf)
- MJV. (s.d.). *Toolkit - Ferramentas do Design Thinking*. Acesso em 10 de Março de 2019, disponível em Inova.gov: [http://inova.gov.br/wp-content/uploads/2017/10/Toolkit\\_DT\\_MJV.pdf](http://inova.gov.br/wp-content/uploads/2017/10/Toolkit_DT_MJV.pdf)
- Myron Youndin, F., Mario W. Clagnaz Jr., F. P., & Louie, H. (1980). *EUA Patente Nº 4207959*.
- Nardini, E. (2016). Novas tecnologias assistivas oferecem mais autonomia a pessoas com deficiência. *Com Ciência*. Acesso em 12 de Abril de 2019, disponível em <http://www.comciencia.br/comciencia/handler.php?section=8&edicao=120&id=1453>
- Néto, L. F. (2019). Desenvolvimento de um manipulador robótico para aplicações assistivas. *Projeto de Graduação*. Brasília, Distrito Federal, Brasil: UnB.

- Norton, R. (2013). *Projeto de Máquinas - Uma Abordagem Integrada* (4ª ed.). São Paulo: Bookman.
- Oliveira, G. T., Saramago, S. F., & Nogueira, A. C. (2010). Estudo das Singularidades de Robôs Manipuladores usando Base de Groebner. *XXXIII Congresso Nacional de Matemática Aplicada e Computacional*.
- Organização Mundial da Saúde. (2011). *Relatório Mundial sobre a Deficiência*.
- Organização Mundial da Saúde. (18 de Maio de 2018). *Assistive technology*. Acesso em 10 de Junho de 2019, disponível em World Health Organization: <https://www.who.int/en/news-room/fact-sheets/detail/assistive-technology>
- Paul, R. (Novembro de 1979). Manipulator Cartesian path control. *IEEE Transactions on Systems, Man, and Cybernetics*, 702-711.
- Pence, G. (2013). Design of a Joystick. *ECE 480*. Michigan, EUA. Acesso em 23 de Abril de 2019, disponível em <https://www.egr.msu.edu/classes/ece480/capstone/spring13/group05/downloads/Application%20Note%20Graham%20Pence.pdf>
- Programação de Robôs*. (s.d.). Acesso em 17 de Maio de 2019, disponível em TECMES: <https://www.tecmes.com.br/solucoes-plm/manufatura/programacao-de-robos/>
- Programação de Robôs Industriais*. (Setembro de 2003). Acesso em 10 de Março de 2019, disponível em e-Disciplinas USP: [https://edisciplinas.usp.br/pluginfile.php/4624520/mod\\_resource/content/0/Aula%20-%20Lab2.pdf](https://edisciplinas.usp.br/pluginfile.php/4624520/mod_resource/content/0/Aula%20-%20Lab2.pdf)
- Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge* (1 ed.). Project Management Institute.
- Rehab, K. (s.d.). *JACO Arm User Guide*. Acesso em 29 de Março de 2019, disponível em <https://www.robotshop.com/media/files/PDF/jaco-arm-user-guide-jaco-academique.pdf>
- RoboCore. (2018). *Controlando motores: motor de passo*. Acesso em 19 de Maio de 2019, disponível em RoboCore: <https://www.robocore.net/tutorials/97.html>
- Rosário, J. M. (2012). *ROBOTICA INDUSTRIAL I: MODELAGEM, UTILIZAÇÃO E PROGRAMAÇÃO*. Baraúna.
- Santos, W. (2015). *Robótica Industrial: fundamentos, tecnologias, programação e simulação*. São Paulo: Érica.
- Secretaria de Direitos Humanos da Presidência da República - SDH/PR. (2009). *Tecnologia Assistiva*. Brasília: Comitê de Ajudas Técnicas.
- Silveira, C. B. (s.d.). *Servo Motor: Veja como Funciona e Quais os Tipo*. Acesso em 13 de Maio de 2019, disponível em Citisystems: <https://www.citisystems.com.br/servo-motor/>
- Snyder. (1985).
- Song, W.-K., Lee, H.-Y., Kim, J.-S., Yoon, Y.-S., & Bien, Z. (1998). KARES - Intelligent Rehabilitation Robotic System for the Disabled and the Elderly. *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2682-2685.

- Topping, M. (2002). An Overview of the Development of Handy 1, a Rehabilitation Robot to Assist the Severely Disabled. *Journal of Intelligent and Robotic Systems*.
- Tsui, K. M., & Yanco, H. A. (s.d.). Interface, Simplifying Wheelchair Mounted Robotic Arm Control with a Visual.
- United Nations. (2016). *A ONU e as pessoas com deficiência*. Acesso em 20 de Setembro de 2018, disponível em ONU BR: <https://nacoesunidas.org/acao/pessoas-com-deficiencia/>
- United Nations. (2018). *United Nations - Disability*. Acesso em 20 de Setembro de 2018, disponível em United Nations: <https://www.un.org/development/desa/disabilities/disability-and-ageing.html>
- Wagner, J. J., Loos, H. F., Smaby, N., Chang, K., & Burgar, C. (1999). ProVAR Assistive Robot Interface. *ICORR'99: Internacional Conference on Rehabilitation Robotics*.
- Wagner, J. J., Van der Loss, H. M., & Leifer, L. J. (1998). Two-Agent User Interface for an Assistive Robot. *RESNA'98*, p. 297.
- Wang, W., Suga, Y., & Sugano, S. (2010). Task Execution Support of a Wheelchair Mounted Robotic Arm in Activity of Daily Livings. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 201-207.
- Yanco, H. A., Hazel, A., Peacock, A., Smith, S., & Wintermute, H. (1995). Initial Report on Wheellesley: A Robotic Wheelchair System.

# APÊNDICES

	Pág.	
Apêndice I	Programa em <i>MATLAB</i> para cálculo da cinemática inversa do manipulador simplificado (até junta $J_4$ )	90
Apêndice II	Programa em <i>MATLAB</i> para simulação com algoritmo de correção dos erros	91

## APÊNDICE I: Programa em *MATLAB* para cálculo da cinemática inversa do manipulador simplificado (até junta $J_4$ )

```
%Programa para cálculo do jacobiano e jacobiano inverso de velocidades do
%manipulador e ser controlado no PG 2
%considerando até a junta 4 para cálculo do jacobiano

clc

%Declarando as variáveis simbólicas
syms theta_1 theta_2 theta_3
syms d_1 a_2 a_3
syms x y z p q v_x v_y v_z omega_1 omega_2 omega_3

%Constantes dimensionais do robô
d_1=0.17;
a_2=0.35;
a_3=0.35;

%Leitura do potenciômetro do joystick para as velocidades
v_x=input('Leitura do valor de v_x pelo joystick: ');
v_y=input('Leitura do valor de v_y pelo joystick: ');
v_z=input('Leitura do valor de v_z pelo joystick: ');

V=[v_x; v_y; v_z];

%Descrevendo as matrizes homogêneas de transformação linear
A_01=[cos(theta_1) 0 sin(theta_1) 0;
      sin(theta_1) 0 -cos(theta_1) 0;
      0 1 0 d_1;
      0 0 0 1];
A_12=[cos(theta_2) -sin(theta_2) 0 a_2*cos(theta_2);
      sin(theta_2) cos(theta_2) 0 a_2*sin(theta_2);
      0 0 1 0;
      0 0 0 1];
A_23=[cos(theta_3) -sin(theta_3) 0 a_3*cos(theta_3);
      sin(theta_3) cos(theta_3) 0 a_3*sin(theta_3);
      0 0 1 0;
      0 0 0 1];

%Logo, a matriz de transformação homogênea da base para a junta 4:
A_04=A_01*A_12*A_23;

%Componentes do vetor posição
p_x=A_04(13);
p_y=A_04(14);
p_z=A_04(15);

% Calculando o jacobiano
Jacob=[diff(p_x,theta_1), diff(p_x,theta_2), diff(p_x,theta_3);
       diff(p_y,theta_1), diff(p_y,theta_2), diff(p_y,theta_3);
       diff(p_z,theta_1), diff(p_z,theta_2), diff(p_z,theta_3)];
% Calculando o jacobiano inverso
iJacob=inv(Jacob);

%Velocidade das juntas
Omega=[omega_1; omega_2; omega_3];

%Leitura do valor atual das variáveis de junta
theta_1=input('Leitura do valor de theta_1 pelo Arduino: ');
theta_2=input('Leitura do valor de theta_2 pelo Arduino: ');
theta_3=input('Leitura do valor de theta_3 pelo Arduino: ');

%Cálculo das velocidades das juntas
Omega=Jacob*V
```

## APÊNDICE II: Programa em *MATLAB* para simulação com algoritmo de correção dos erros

```
%Este programa é para controlar de maneira cartesiana o robo planar de teste
%Com base nas orientacoes de petercorke.com sobre a toolbox de robotica
%Autoria de Paulo Maia da Costa Filho 140170405
```

```
clc
clear all
close all

syms teta_1 teta_2 x y alpha k

%Descrevendo os elos do manipulador
L(1)=Link([teta_1, 0, 0.09, 0]);
L(2)=Link([teta_2, 0, 0.09, 0]);
L
planar=SerialLink(L, 'name','Planar');

%Posição cartesiana do efetador
teta_1=input('Insira um valor atual de teta_1: ');
teta_2=input('Insira um valor atual de teta_2: ');

%Posição do joystick
Jx=input('Insira um valor para Vx do joystick: ');
Jy=input('Insira um valor para Vy do joystick: ');

%Intervalo de tempo para o programa rotar o chaveamento
intervalo=input('Insira o intervalo de tempo de trabalho: ');

%Cinemática direta inicial
xi = 0.09*cos(teta_1) + 0.09*cos(teta_1+teta_2)
yi = 0.09*sin(teta_1) + 0.09*sin(teta_1+teta_2)

%%Análise para padronizar velocidades
%Para Jx
if Jx >768
    Vx=+0.05; %50 mm/s ou 0.05 m/s
elseif 512<Jx && Jx<=768
    Vx=+0.02; %20 mm/s ou 0.02 m/s
elseif Jx==512 && Jy>512
    Vy=0.0;
elseif 256<=Jx && Jx<512
    Vx=-0.02; %20 mm/s ou 0.02 m/s
else Jx<256
    Vx=-0.05; %50 mm/s ou 0.05 m/s
end

%Para Jy
if Jy >768
    Vy=+0.05; %50 mm/s ou 0.05 m/s
elseif 512<Jy && Jy<=768
    Vy=+0.02; %20 mm/s ou 0.02 m/s
elseif Jy==512
    Vy=0.0;
elseif 256<=Jy && Jy<512
    Vy=-0.02; %20 mm/s ou 0.02 m/s
else Jy<256
    Vy=-0.05; %50 mm/s ou 0.05 m/s
end

%Inclinação da reta teórica
alpha_i=tan(Vy/Vx)

%Definindo o contador
k=1;

%Valor inicial de corretor de velocidades
betax=0;
betay=0;
```



```

%Mudando variáveis
Vcx=Vx;
Vcy=Vy;

for k=1:10
    %Correção das velocidades
    %Vcx
    if betax >=0 && Vcx<=0 || betax <=0 && Vcx>=0
        Vcx=Vcx+abs(betax); %deve fazer o movimento oposto
    elseif betax<0 && Vcx<0 || betax>0 && Vcx>0
        Vcx=Vcx-abs(betax);
    end
    %Vcy
    if betay >=0 && Vcy<=0 || betay <=0 && Vcy>=0
        Vcy=Vcy+abs(betay); %deve fazer o movimento oposto
    elseif betay<0 && Vcy<0 || betay>0 && Vcy>0
        Vcy=Vcy-abs(betay);
    end

    %Compilando resultados
    V=[Vx; Vy];

    %Cinemática direta teórica
    xt=xi+((k-1)*Vx*intervalo)
    yt=yi+((k-1)*Vy*intervalo)

    %Jacobiano
    J= [-0.09*(sin(teta_1))-0.09*(sin(teta_1+teta_2)), -0.09*(cos(teta_1+teta_2));
        0.09*(cos(teta_1))+0.09*(sin(teta_1+teta_2)), 0.09*(sin(teta_1+teta_2))];

    %Jacobiano inverso
    j=(1/(0.09*0.09*((cos(teta_1)*cos(teta_1+teta_2))-
    (sin(teta_1)*sin(teta_1+teta_2)))+(0.09*0.09*((cos(teta_1+teta_2)*sin(teta_1+teta_2))-
    sin(teta_1+teta_2)^2))))).*([0.09*sin(teta_1+teta_2) 0.09*cos(teta_1+teta_2);
    -0.09*cos(teta_1)-0.09*sin(teta_1+teta_2), -0.09*sin(teta_1)-
    0.09*sin(teta_1+teta_2)]);

    %Encontrando velocidade das juntas
    w=j*V;

    %velocidades em delay
    delay1=1000*pi/(32*w(1));
    delay2=1000*pi/(32*w(2));

    %variação da variável de junta cada motor de passo por intervalo segundo
    q1=intervalo*w(1);
    q2=intervalo*w(2);

    %novas variáveis de junta (posição das juntas)
    teta_1=teta_1+q1
    teta_2=teta_2+q2

    %Cinemática direta real
    xr = 0.09*cos(teta_1) + 0.09*cos(teta_1+teta_2)
    yr = 0.09*sin(teta_1) + 0.09*sin(teta_1+teta_2)

    %Inclinação da reta real
    alpha=(yr-yt)/(xr-xt)

    %Ajustes nas velocidades
    betax=(xr-xt)/intervalo
    betay=(yr-yt)/intervalo

    %Verificando se o efetuador está próximo a uma singularidade
    if sqrt((xr)^2+(yr)^2)>=0.16 || sqrt((xr)^2+(yr)^2)<=0.03
        fprintf('Cuidado! Você está próximo(a) a uma singularidade!');
    end
    k=k+1;
end

```