



DISSERTAÇÃO DE MESTRADO

**DESENVOLVIMENTO DE GARRA ROBÓTICA
ANTROPOMÓRFICA COM PERCEPÇÃO TÁTIL**

ARTUR VITÓRIO ANDRADE SANTOS

Brasília, Dezembro de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

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

**DESENVOLVIMENTO DE GARRA ROBÓTICA ANTROPOMÓRFICA
COM PERCEPÇÃO TÁTIL**

ARTUR VITÓRIO ANDRADE SANTOS

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
SISTEMAS MECATRÔNICOS**

APROVADA POR:

Prof. Dr. Alberto J. Alvares, PPMEC/UnB
Orientador

Prof. Dr. Li Weigang
Membro Interno

Prof. Dr. Marcelino Monteiro de Andrade
Membro Externo

BRASÍLIA/DF, 7 de Dezembro

Vitório, Artur

Desenvolvimento de garra robótica antropomórfica com percepção tátil /
ARTUR VITÓRIO ANDRADE SANTOS. –Brasil, 2018. 163p.

Orientador: Alberto José Alvares

Dissertação (Mestrado) – Universidade de Brasília – UnB

Faculdade de Tecnologia – FT

Programa de Pós-Graduação em Sistemas Mecatrônicos – PPMEC, 2018.

1. Garra Robótica Antropomórfica. 2. Percepção Tátil. 3. Lógica Fuzzy. 4.
Desenvolvimento de Produto. 5. Robótica Industrial. I. Alberto José Alvares,
orientador. II. Universidade de Brasília. III. Faculdade de Tecnologia.

Dedicatória

Dedico esse trabalho ao meu pai, mãe e irmã.

ARTUR VITÓRIO ANDRADE SANTOS

Agradecimentos

Agradeço primeiramente a minha família, os quais sempre estiveram ao meu lado, nas minhas conquistas e derrotas. A todos os meus amigos de trabalho, especialmente ao Kariston Dias e Paulo Roberto, por compartilharem os estudos ao longo do mestrado e em especial ao último por compartilhar seu lar e orientações. Ao meu orientador Alberto José Alvares pela paciência e ensinamentos prestados a minha pessoa e por ter confiado em mim durante a realização desse trabalho. Agradeço imensamente aos meus colegas de laboratório Efrain, Christian, Sebastian, Luiz, Glauber, Renan e Sérgio por me auxiliarem no desenvolvimento do trabalho de forma direta ou indiretamente. E finalmente ao meu local de trabalho, Instituto Federal de Rondônia, que me deu a oportunidade de realizar o mestrado e onde pude estar realizando parte do trabalho aqui desenvolvido.

ARTUR VITÓRIO ANDRADE SANTOS

RESUMO

Esta dissertação tem por objetivo o desenvolvimento de uma garra robótica antropomórfica com percepção tátil, bem como, apresentar e documentar o projeto mecatrônico da garra robótica com base em uma metodologia de desenvolvimento de produto genérico, realizando o estudo cinemático e posterior manipulação de objetos com restrição de força evitando deformação e colapso. Para tanto, devido aos fins práticos do projeto proposto, foi utilizado da pesquisa exploratório com o objetivo de gerar conhecimento dirigido à solução do problema específico proposto. A partir da abordagem adotada, foi possível desenvolver a garra robótica antropomórfica com 5 dedos, sendo o maior desafio, a integração da percepção tátil, através do controle de conformidade ativo, utilizando a plataforma *Raspberry Pi 3 B+*. Através da solução proposta, o destaque fica a cargo da manipulação de objetos com diferentes geometrias, pesos e tamanhos. A importância de um sistema de controle para percepção tátil, permite a manipulação de objetos sem danificá-los. Devido à solução proposta ser de baixo custo, a maior limitação dá-se no sistema de controle de conformidade, uma vez que, para a implementação do controle para os 5 dedos da garra robótica, requer um custo computacional muito alto. Com o algoritmo desenvolvido, foi possível controlar os 5 dedos da garra robótica com tempo de resposta em cerca de 300 ms. Outrossim, a necessidade de interação do sensor com o objeto manipulado, tornou-se um fator limitante da dissertação, tendo em vista, que a superfície dos objetos nem sempre são planas e a base de contato do sensor de força FSR ser pequena. Ao fim, de 15 requisitos definidos para a garra robótica, apenas 5 não atenderam satisfatoriamente a solução, porém, eles não foram tratados como os principais requisitos de projeto.

ABSTRACT

This dissertation aims to develop an anthropomorphic robotic hand with tactile perception, as well as to present and document the gripper robotic mechanical model based on a generic product method, performing the kinematic data and later handling of the force avoiding deformations and collapses. For this, due to the objectives of the project, the exploratory research was used with the objective of generating knowledge directed to the solution of the problem. Based on the adopted approach, an anthropomorphic robotic package with 5 fingers was elaborated, being the biggest challenge, an integration of the tactile perception, through the control of connection, using a platform Raspberry Pi 3 B+. Through the proposed solution, the highlight is the handling of objects with different geometries, weights and sizes. The importance of a control system for tactile perception allows the manipulation of objects without damaging them. The biggest low cost service solution is most of the access control systems, because, for the implementation of the control of 5 fingers of the table, a very high computational cost is necessary. With the algorithm developed, it was possible to control the 5 fingers of the robotic task with the response time in about 300 ms. In addition, the need for interaction of the sensor with the manipulated object has become a limiting factor of the dissertation, considering that a surface of the objects is not always flat and a contact base of the FSR force sensor. Finally, of 15 requirements for robotic robotics, only 5 did not respond satisfactorily to the solution, however, they were not included as the main design requirements.

SUMÁRIO

RESUMO	i
ABSTRACT	ii
LISTA DE FIGURAS	v
LISTA DE TABELAS	ix
LISTA DE ABREVIATURAS	xi
1 Introdução	1
1.1 Contextualização	1
1.2 Definição do Problema.....	3
1.3 Justificativa	4
1.4 Objetivos da Dissertação	4
1.4.1 Objetivo Geral	4
1.4.2 Objetivos Específicos.....	4
1.5 Aspectos Metodológicos	4
1.6 Contribuições do Trabalho.....	5
1.7 Estrutura da Dissertação.....	5
2 Fundamentação Teórica	7
2.1 Dados Históricos e Características das Garras Robóticas Antropomórficas	7
2.2 Garras Robóticas	10
2.3 Garra Robótica Sub-Atuada	13
2.4 Mão Humana.....	15
2.5 Controle de Força em Garras Robóticas	18
2.6 Estratégia de Controle <i>Fuzzy</i>	20
2.6.1 Configuração Básica de um Controlador <i>Fuzzy</i>	22
2.6.2 Funções de Pertinência <i>Fuzzy</i>	23
2.7 Sensores de Força	24
2.8 Processo de Desenvolvimento de Produto	27

2.8.1	Fase 0 - Planejamento	28
2.8.2	Fase 1 - Desenvolvimento do Conceito	29
2.8.3	Fase 2 - Projeto ao Nível de Sistema	31
2.8.4	Fase 3 - Projeto detalhado.....	31
2.8.5	Fase 4 - Testes e Refinamento.....	31
2.8.6	Fase 5 - Lançamento à Produção	32
3	Trabalhos Correlatos	33
3.1	Garras Robóticas Antropomórfica	33
3.2	Controle de Força em Garras Robóticas	39
3.3	Sensores Táteis em Garras Robóticas Antropomórfica	42
4	Desenvolvimento da Garra Robótica Antropomórfica	48
4.1	Fase 0 - Planejamento	48
4.2	Fase 1 - Desenvolvimento do Conceito	48
4.2.1	Necessidades dos clientes	49
4.2.2	Priorização dos requisitos dos clientes.....	49
4.2.3	Requisitos de projeto	52
4.2.4	Relacionamento entre necessidades dos clientes e requisitos do produto	53
4.2.5	Correlação dos requisitos de projeto.....	53
4.2.6	Análise de competitividade	54
4.2.7	Definição das especificações-meta	55
4.3	Fase 2 - Projeto ao Nível de Sistema	55
4.4	Fase 3 - Projeto Detalhado	61
4.4.1	Projeto Mecânico.....	62
4.4.2	Análise de mobilidade e modelagem cinemática	63
4.4.3	Descrição dos sistemas do domínio mecânico	73
4.4.4	Projeto eletroeletrônico	77
4.4.5	Projeto software de controle	78
4.4.6	Controle de Conformidade.....	81
4.5	Custo de Produção	84
5	Testes Funcionais e Experimentos da Garra Robótica Antropomórfica.....	86
5.1	Materiais Empregados na Construção da Garra Robótica Antropomórfica	86
5.2	Análise da Garra Robótica baseado em Métodos Analíticos	87
5.3	Análise da Garra Robótica baseado em Métodos Experimentais	94
5.3.1	Manipulação de Objetos.....	97
5.3.2	Formas Canônicas de Agarre	100
6	Conclusões.....	104
6.1	Trabalhos futuros.....	107

REFERÊNCIAS BIBLIOGRÁFICAS	108
APÊNDICES.....	117
A Algoritmo de Controle de Conformidade por Lógica Fuzzy.....	118
B Lista Completa de Materiais Utilizados.....	139
C Algoritmo da Cinemática no Matlab	141
D Projeto Eletrônico.....	144
D.0.1 Layout Completo da Placa Eletrônica.....	145
D.0.2 Legenda de Componentes	146
D.0.3 Furos	147
E Material Compartilhado.....	148

LISTA DE FIGURAS

1.1	Robo Moley com garra robótica antropomórfica realizando atividades de cozinha ..	2
2.1	Garra robótica com características antropomórficas desenvolvida por Salisbury	8
2.2	Número de dedos utilizados nas garras robóticas	8
2.3	Comparativo dos diferentes mecanismos utilizados em garras robóticas	10
2.4	Linha do tempo dos eventos mais marcantes relacionados as garras robóticas.....	11
2.5	Classificação das garras robóticas	12
2.6	Crescimento dos robôs no setor industrial até 2020, conforme a federação internacional de robótica.....	13
2.7	Anatomia da mão humana. Os números indicam a quantidade de GDL da mão.....	14
2.8	Demonstração do mecanismos de uma garra robótica sub-atuada de dois dedos	15
2.9	Representação de barras da garra robótica sub-atuada	15
2.10	Mecanismos sub-atuado com tendões e polias	16
2.11	Estrutura da mão humana.....	17
2.12	Formas comuns de agarre de objetos baseadas na mão humana, sendo da esquerda para direita: agarre prismático, circular, esférico e cilíndrico e precisão planar	17
2.13	10 tipos mais comuns de agarres usados no dia a dia de uma empregada doméstica e dois maquinistas	18
2.14	Disposição das juntas e GDL referente ao modelo proposto.....	19
2.15	Modelo genérico de controle de conformidade	20
2.16	Modelo de representação de um sistema de controle tradicional	21
2.17	Modelo de representação de um sistema de controle <i>fuzzy</i> típico.....	22
2.18	Blocos funcionais de um controlador <i>fuzzy</i>	23
2.19	Representação das principais funções de pertinências.....	23
2.20	Mapa de regras para <i>fuzzificação</i>	24
2.21	BarretHand e Shadow Hand utilizando os dedos com sensores táteis (em azul) desenvolvidos por [71]	25
2.22	Vista explodida do sensor resistivo de força	26
2.23	Comportamento do sensor resistivo de força	27
2.24	Modelo genérico de Desenvolvimento de Produto Proposto.....	28
2.25	Processo de planejamento do produto	29

2.26	Modelo da Casa da Qualidade Resumido.....	30
2.27	Modelo desenvolvido pelo professor Gausmeier	32
3.1	Garra robótica antropomórfica desenvolvida por [99]	34
3.2	Garra robótica antropomórfica <i>SHU II</i> desenvolvida por [66].....	35
3.3	Garra robótica antropomórfica <i>ISR-SoftHand</i> desenvolvida por [61].....	35
3.4	Garra robótica antropomórfica <i>Hit Hand</i> desenvolvida por [100]	36
3.5	Garra robótica antropomórfica <i>DLR II</i> desenvolvida por [36].....	37
3.6	Garra robótica antropomórfica <i>KNU Hand</i> desenvolvida por [67]	37
3.7	Garra robótica antropomórfica desenvolvida por [102], com diferentes agarre de diferentes diâmetros	40
3.8	Garra robótica antropomórfica desenvolvida por [104]. Figura (a) vista frontal e Figura (b) vista lateral	41
3.9	sistema de controle de força proposto por [105].	41
3.10	Disposição dos FSR sobre a mão robótica de Yokoi [106].	42
3.11	Garra robótica desenvolvida por [107]. Na figura a) a estrutura da garra robótica e os sensores táteis em branco. Na figura b) a manipulação de um ovo cozido sem deformação.....	43
3.12	Garra robótica equipada com os sensores capacitivos [108]. Figura a) Garra robótica montada com os sensores capacitivo em azul claro e Figura b) Vista CAD do acoplamento dos sensores capacitivos em laranja	44
3.13	Robonaut desenvolvido pelo projeto NASA/DARP. Figura a) Robonaut desenvolvido para operações espaciais equipada com a garra robótica. Figura b) Garra robótica com percepção tátil para operações espaciais.....	45
3.14	Garra robótica antropomórfica desenvolvida por [110]. Figura a) Mostra a manipulação de uma garra de vidro e na Figura b) A disposição dos sensores sobre a garra robótica.....	46
3.15	Matriz de sensores táteis desenvolvido por [77]	46
4.1	Hierarquização dos requisitos dos clientes, conforme diagrama de Mudge.....	50
4.2	Matriz da Casa da qualidade	54
4.3	Representação da função global e desdobramento da função global.....	56
4.4	Visão geral da garra robótica antropomórfica proposta	62
4.5	Representação de barras para a garra robótica desenvolvida.....	63
4.6	Representação dos referenciais locais do dedo mínimo, anelar, médio e indicador ..	64
4.7	Representação dos referenciais locais para o polegar.....	66
4.8	Descrição dos componentes comuns aos dedos mínimo, anelar, médio e indicador .	74
4.9	Descrição dos componentes para o dedo polegar	75
4.10	Descrição da base da palma.....	75
4.11	Descrição do dorso da palma	76

4.12	Flange para acoplamento em manipuladores robóticos	76
4.13	Desenho CAD renderizado da garra robótica antropomórfica completa	77
4.14	Garra robótica antropomórfica desenvolvida pelo projeto <i>The Open Hand</i>	78
4.15	Caso de Uso para o projeto de Software da Garra Robótica	79
4.16	Interface Gráfica do Usuário	80
4.17	Comunicação através do Servidor MQTT Broker	81
4.18	Diagrama de controle de conformidade, baseado em lógica <i>fuzzy</i>	82
4.19	Funções de pertinência para a entrada de força	82
4.20	Funções de pertinência para a saída de controle de força	83
5.1	Impressoras 3D utilizadas para construção da garra robótica. Figura (a) impressora 3D Profissional uPrint e Figura (b) impressora 3D RepRap	87
5.2	Estrutura em MDF com a garra robótica antropomórfica montada. Figura (a) a garra montada no suporte simulando o acoplamento de um manipulador e Figura (b) a eletrônica fixada para possibilitar o controle da garra robótica	88
5.3	Modelo da CNC router 3018 utilizada para fabricação da placa eletrônica	88
5.4	Espaço de trabalho considerado da garra robótica antropomórfica, modelo simplificado para simulação	89
5.5	Deslocamento do Eixo X no espaço tridimensional	91
5.6	Deslocamento do eixo Y no espaço tridimensional	91
5.7	Deslocamento do Eixo X no espaço tridimensional para o polegar	92
5.8	Deslocamento do eixo Y no espaço tridimensional para o polegar	93
5.9	Deslocamento do eixo Z no espaço tridimensional para o polegar	93
5.10	Custo computacional para o sistema de controle de conformidade desenvolvido	94
5.11	Resposta do sistema de controle para um <i>setpoint</i> de 2N	95
5.12	Resposta do sistema de controle para um <i>setpoint</i> de 4N	95
5.13	Resposta do sistema de controle para um <i>setpoint</i> de 1N	96
5.14	Resposta do sistema de controle para um <i>setpoint</i> de 6N	96
5.15	Resposta do sistema de controle para um <i>setpoint</i> de 8N	97
5.16	Resposta do sistema de controle para um <i>setpoint</i> de 10N	97
5.17	Diferentes objetos considerados para manipulação. Sendo da esquerda para direita: fita isolante, alicate universal, caneta, multímetro, <i>protoboard</i> , chave combinada 15, tubo de cola bastão, objeto com formato elíptico 3D, chave <i>allen</i>	98
5.18	Objetos considerados para manipulação	99
5.19	Ajuste do controle de conformidade para 2N sem deformação do balão	100
5.20	Ajuste do controle de conformidade para 7.82N com deformação do balão	100
5.21	Comportamento do sistema mediante falha na interação entre o sensor de força e objeto	101
5.22	Programação das formas canônicas e controle de abertura e fechamento da garra robótica	101

5.23	Agarre da forma canônica “pegar garrafa”	102
5.24	Agarre da forma canônica “pegar tipo pinça”	103
5.25	Formas canônicas para a garra robótica antropomórfica. Figura (a) Agarre da forma canônica “pegar três dedos” e Figura (b) Agarre da forma canônica “pegar quatro dedos”	103

LISTA DE TABELAS

2.1	Comparativo em relação ao número de atuadores, número de dedos e graus de liberdade de diferentes garras robóticas encontradas na literatura.....	9
2.2	Comparação entre os diferentes sensores existentes	26
3.1	Comparação entre garras robóticas antropomórficas desenvolvidas	38
3.2	Comparativo dos principais tipos de sensores aplicados em garras robóticas	45
4.1	Necessidades para Garra Robótica Antropomórfica	50
4.2	Estabelecimento dos Requisitos de Projeto	52
4.3	Especificação Meta da Garra Robótica Antropomórfica	55
4.4	Matriz morfológica com os princípios de solução	57
4.5	Tabela com as alternativas de solução	59
4.6	Matriz de Pugh	60
4.7	Parâmetros DH para os dedos mínimo, anelar, médio e indicador	65
4.8	Parâmetros DH para o polegar	66
4.9	Matriz de comando para variável de controle de força à partir da variável de medição de força e <i>setpoint</i> de força.....	83
4.10	Custo dos componentes para confecção da garra robótica antropomórfica.....	84
5.1	Restrições de ângulo máximo para os elos dos dedos mínimo, anelar, médio e indicador.....	89
5.2	Restrições de ângulo máximo para os elos do polegar	89
5.3	Valores calculados através do <i>Matlab</i> para posição final dos dedos mínimo, anelar, médio e indicador	90
5.4	Valores obtidos através da simulação do <i>SolidWorks</i> para posição final dos dedos ..	90
5.5	Valores calculados através do <i>Matlab</i> para posição final do polegar.....	92
5.6	Valores obtidos através da simulação do <i>SolidWorks</i> para posição final do polegar..	92
5.7	Tabela com diferentes tipos de objetos considerados para manipulação.	98
6.1	Requisitos de projeto atingidos conforme a necessidades dos clientes	105

LISTA DE ABREVIATURAS

AB.	INGLÊS	PORTUGUÊS
ABS	Acrylonitrile Butadiene Styrene	Acrilonitrilo Butadieno Estireno
API	Application Programming Interface	Interface de Programação de Aplicativos
CAD	Computer Aided Design	Desenho Assistido por Computador
DH	Denavit-Hartenberg	Denavit-Hartenberg
DIP	Distal Interphalangeal Joint	Junta Interfalângica Distal
EMG	Electromyogram	Eletromiográfico
FPGA	Field Programmable Gate Array	Matriz de Portas Programáveis em Campo
FSR	Force Sensing Resistor	Sensor Resistivo de Força
MCP	Metacarpophalangeal Joints	Junta Metacarpofalângica
MEMS	Micro-Electro-Mechanical Systems	Sistemas Microeletromecânicos
MQTT	Message Queuing Telemetry Transport	Fila Mensageria de Transporte de Telemetria
PDP	Product Development Process	Processo de Desenvolvimento de Produto
PID	Proportional-Integral-Derivative	Proporcional-Integral-Derivativo
PIP	Proximal Interphalangeal Joints	Junta Interfalângica Proxima
PWM	Pulse Width Modulation	Modulação por Largura de Pulso
QFD	Quality Function Deployment	Desdobramento da Função de Qualidade
RC	Radio-Carpal Joint	Junta Radiocarpática
VDI	Verein Deutscher Ingenieure (Alemão)	Associação de Engenheiros Alemães
GDL	Degree of Freedom (DoF)	Graus de Liberdade
CNC	Computer Numeric Control	Comando Numérico Computadorizado

Capítulo 1

Introdução

1.1 Contextualização

Com o advento da nova revolução industrial em curso, denotada indústria 4.0, os robôs estão cada vez mais substituindo os humanos em tarefas repetitivas, inseguras ou de fabricação pesada. A maioria das indústrias de automóveis, eletrônicos, aeroespacial, entre outras, utilizam manipuladores robóticos com garras robóticas do tipo pinça em linhas de produção de grande escala. Porém, devido às atuais condições de fabricação e montagem, os produtos estão sendo produzidos em menor escala, contudo, mais personalizáveis e variáveis, necessitando de robôs com maior adaptabilidade, fácil reconfiguração em *software* e *hardware*, mais flexibilidade e mais capacidade de manipulação [1].

Conforme Groover [2], as garras robóticas mecânicas são efetadores finais utilizadas para agarrar e manipular objetos durante o ciclo de trabalho. Devido à variedade de formatos, tamanhos e pesos das peças, a maioria das garras devem ser personalizadas, existindo diferentes tipos de garras robóticas utilizadas em aplicações industriais, tais como: garra a vácuo, dispositivos magnetizados e garras mecânicas. As garras mecânicas de dois dedos são os tipos mais comuns de garras, apresentando grande variedade, sendo sua principal desvantagem a limitação da abertura dos dedos não assegurando uma compreensão segura do objeto, podendo deslizar facilmente caso possua irregularidades [3].

As garras robóticas antropomórficas, por outro lado, possuem a característica da mão humana, possibilitando a manipulação de diferentes categorias de objetos, com formas, peso e tamanho diferente, conforme as restrições do projeto. A maioria dos pesquisadores que trabalham com manipulação de robôs e manipulação de objetos tendem a usar robôs manipuladores disponíveis no mercado, equipados com vários efetadores para estudos experimentais. No entanto, garras robóticas comercialmente disponíveis são caras e não são fáceis de modificar para propósitos específicos [4]. O uso de garras robóticas antropomórficas não se limitam apenas ao uso industrial. Fernando et al. [5] relatam o uso nas áreas doméstica, médica, construção, limpeza

e entretenimento. Por exemplo, o protótipo desenvolvido pela empresa Moley Robotic [6], visto na Figura 1.1, consiste em uma cozinha robótica. O robô é equipado com uma garra robótica, que pode interagir com a maioria dos equipamentos da cozinha, como liquidificadores, batedeiras, facas e fogão [7]. Através de uma câmera 3D integrada e uma luva para captar os gestos de um cozinheiro profissional, todas as informações de trabalho de um chef humano são captadas e armazenadas em um banco de dados. Posteriormente, essas informações são decodificadas e enviadas para o robô reproduzir todos os movimentos.



Figura 1.1: Robo Moley com garra robótica antropomórfica realizando atividades de cozinha
Fonte: [6]

Um dos desafios relacionado a produção de garras robóticas, encontra-se na integração dos atuadores, sensores e controladores em um projeto leve e compacto. Pons et. al [8] relatam que para o desenvolvimento de garra robótica com características de mão humana, uma série de características devem ser levados em conta, como: ser antropomórfica, fácil manipulação, ter a capacidade de pressão de objetos, baixo custo, apresentar interface de comunicação com o usuário, sentir a presença do objeto, entre outros.

Destaca-se também a manipulação autônoma dos objetos, sendo essa, uma das habilidades-chave desejadas pelos robôs industriais e sociais [9]. Ao contrário dos ambientes industriais, os espaços domésticos normalmente não são estruturados, o que significa que a percepção tátil precisa ser adicionada às estratégias de controle dos robôs [1]. Estudos mostram que pessoas com a ponta dos dedos anestesiadas são incapazes de manter um aperto estável [10]. Os sensores táteis fornecem aos robôs informações sobre o contato físico, por meio dos quais as mãos de robôs autônomos podem operar em ambientes não estruturados e manipular objetos desconhecidos [11].

Considerando as dificuldades apresentadas, bem como a aplicação das garras robóticas antropomórficas em ambientes industriais e domésticos. Para ampliar a escolha de garras robóticas de código aberto disponíveis para pesquisadores e educadores, é apresentado o desenvolvimento de

uma garra robótica antropomórfica de baixo custo com código aberto. Para obter-se os requisitos necessários para o desenvolvimento, o trabalho é baseado na metodologia de desenvolvimento de produto proposta por [12].

1.2 Definição do Problema

As garras mecânicas de dois dedos são as categorias mais comuns de garras robóticas apresentando grande variedade. A principal desvantagem é a limitação da abertura dos dedos, não assegurando uma compreensão segura dos objetos, onde os objetos podem deslizar facilmente caso possuam irregularidades [3]. As garras robóticas antropomórficas por se assemelharem a mão humana, possuem grande capacidade de manipulação de diferentes tipos de objetos. O desenvolvimento de garras robóticas antropomórficas, podem ser realizados com muitos conceitos e *design* diferentes. O *design* do sistema de atuação e transmissão de movimentos são uns dos principais aspectos a serem considerados no desenvolvimento, devido ao espaço e as dimensões serem geralmente limitados em um projeto de garra robótica antropomórfica [13].

Outrossim, dentro dos sistemas robóticos, o controle da interação física entre um robô manipulador e o ambiente em que está inserido é crucial para a execução bem-sucedida de uma série de tarefas práticas onde o efetuador final do robô (garra robótica) tem que manipular um objeto ou executar alguma operação em uma superfície [14]. Conforme Kappassov et al. [1] nas aplicações industriais tradicionais, o controle do efetuador final do robô é obtido através do conhecimento prévio do objeto a ser manipulado e do ambiente os quais são inseridos no algoritmo de controle do robô. Diante de situações inesperadas e ambiente poucos conhecidos, os efetuadores finais do robô apresentam dificuldade de manipulação. Para superar essa limitação, a utilização de sensores de força, baseada em controle de conformidade ativo, pode ser empregado em situações onde não se conhece o ambiente e o objeto a ser manipulado. O controle de conformidade foi introduzido em pesquisas para replicar com segurança os movimentos das mãos durante a interação com os objetos [15].

Diante do exposto, percebe-se que o desenvolvimento de garras robóticas com múltiplos dedos (antropomórfica) e sistema de realimentação de força é um processo muito complexo de natureza interdisciplinar envolvendo vários campos de pesquisa [16]. Sendo assim, buscou-se reunir dados/informações com o propósito de responder ao seguinte problema de pesquisa: É possível desenvolver uma garra robótica antropomórfica de baixo custo com percepção tátil utilizando controle de conformidade ativo?

1.3 Justificativa

Com advento da indústria 4.0 e os atuais processos de fabricação trabalharem com objetos cada vez mais personalizados, as atuais garras robóticas por possuírem movimento tipo pinça, possuem movimentos limitados manipulando apenas um objeto por vez. Outrossim, uma garra com características antropomórficas pode ser utilizada tanto em ambiente industrial como residencial, em robôs domésticos e pessoas com deficiência de membro. As garras robóticas industriais possuem custos elevados que para serem aplicadas como recurso educacional, acaba tornando-se inviável.

1.4 Objetivos da Dissertação

1.4.1 Objetivo Geral

O objetivo geral do projeto é desenvolver uma garra robótica antropomórfica com percepção tátil, utilizando metodologia de desenvolvimento de produto.

1.4.2 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, foram estabelecidos os objetivos específicos a seguir:

- Apresentar e documentar o projeto mecatrônico da garra com base na metodologia de desenvolvimento de produto proposta por [12].
- Realizar o estudo da cinemática da garra robótica antropomórfica desenvolvida.
- Implementar um sistema de percepção tátil para a garra robótica antropomórfica desenvolvida.
- Manipular objetos com restrição de força evitando a deformação e colapso a fim de realizar testes funcionais, através de métodos analíticos e experimentais.

1.5 Aspectos Metodológicos

A fim de se obter os objetivos específicos dessa pesquisa, observou-se que ela é classificada como pesquisa exploratório, pois, utiliza-se de pesquisa bibliográfica, tais como desenvolvimento de garras robóticas de sucesso, bem como o estudo da metodologia de desenvolvimento de produto para possibilitar o desenvolvimento da garra robótica. Quanto a forma de abordagem da

pesquisa, a mesma pode ser tanto qualitativa, como quantitativa. A pesquisa qualitativa, utiliza da subjetividade não requerendo o uso de métodos e estatística. Enquanto a pesquisa quantitativa utiliza de parâmetros estatísticos para analisar os dados [17].

Devido às especificação-metas estabelecidas com base na metodologia de desenvolvimento de produto adotado para o desenvolvimento do trabalho, os parâmetros da garra robótica antropomórfica são mensuráveis, sendo assim, passíveis de medição. Sendo analisada de forma hipotético-dedutiva na base da tentativa e erro, testes e experimentações, por se tratar de um projeto de produto.

1.6 Contribuições do Trabalho

A contribuição do trabalho, consiste no desenvolvimento de uma garra robótica com baixo custo empregando sistema de controle de conformidade com recursos encontrados no mercado nacional a fim de ser replicado por estudantes e pesquisadores que estejam iniciando no mundo da robótica. A partir do desenvolvimento e as necessidades dos clientes propostas no trabalho, observou-se que o modelo desenvolvido pelo projeto *The Open Hand Project* [18], assemelhavam-se a algumas necessidades estabelecidas no projeto. O modelo foi remodelado, sendo melhorado o mecanismo de suporte dos motores, transmissão de movimento e estrutura dos dedos, a fim de comportar os sensores de força. Através das alterações na estrutura da garra, foi implementado o controle de conformidade com o intuito de atender as necessidades dos clientes, conforme metodologia proposta.

1.7 Estrutura da Dissertação

O trabalho está organizado em 6 capítulos. Além do capítulo introdução, aqui abordado, possui o Capítulo 2 onde é descrito a fundamentação teórica do trabalho. É descrito a evolução das garras robóticas antropomórficas, como funciona uma garra robótica sub-atuada, classificação dos sensores de força, estratégia de controle *fuzzy*, o que significa o controle de conformidade em garras robóticas, bem como a metodologia de desenvolvimento de produto, além de outros tópicos fundamentais para o entendimento do trabalho.

No capítulo 3 é apresentado os trabalhos correlatos, dando destaque para os trabalhos desenvolvidos com garras robóticas sub-atuadas e estudos onde empregam o controle de força. É apresentado os trabalhos mais recentes na área de relevância e que contribuíram para o desenvolvimento do trabalho aqui proposto.

No capítulo 4 é descrito a metodologia de desenvolvimento de produto empregado, elencando as necessidades dos clientes e requisitos do produto para possibilitar, através da casa da quali-

dade, elaborar os requisitos-meta da garra robótica. Ainda, através da abordagem em modelo V [19] é descrito o projeto detalhado, descrevendo os sistemas mecânicos, eletroeletrônico e computacional. No sistema mecânico é abordado a cinemática e construção do mecanismo. Para o sistema eletroeletrônico é descrito a interligação dos componentes elétricos e eletrônicos, enquanto o computacional, o sistema de controle de conformidade implementado e interface gráfica desenvolvida.

Já no capítulo 5 é realizado a validação e experimentação da garra robótica desenvolvida. Na validação é comparado os valores computacionais da cinemática realizada, com simulações de posição realizadas no software *SolidWorks* através do mecanismo desenvolvido. Para o sistema de controle de conformidade é analisado o comportamento dinâmico da resposta ao agarre a um objeto com características de maleabilidade.

Para o capítulo 6 é exposto à conclusão do trabalho, dando destaque para as principais observações ao longo do desenvolvimento do projeto e as previsões de trabalhos futuros a serem realizados na garra robótica.

Capítulo 2

Fundamentação Teórica

No presente capítulo é apresentado a fundamentação teórica associada ao desenvolvimento da garra robótica antropomórfica. É apresentado, uma visão geral das garras robóticas com dados históricos, principais atuadores e mecanismos de transmissão empregados, além das vantagens das garras robóticas de múltiplos dedos. Posteriormente é apresentado a importância do controle de conformidade para manipulação de objetos baseado no controle de conformidade ativo. Para melhor compreensão do desenvolvimento da garra robótica é demonstrado a estrutura da mão humana, à qual são baseadas, demonstrando as juntas existentes da mão humana. Por fim, é apresentado a metodologia de desenvolvimento de produto utilizada para desenvolvimento do trabalho.

2.1 Dados Históricos e Características das Garras Robóticas Antropomórficas

O efetivo desenvolvimento de garras robóticas como conhecemos atualmente, em formato de mão (mais conhecidas como antropomórficas), deu-se por volta de 1960 nos Estados Unidos e Japão onde os pesquisadores desenvolveram uma garra robótica constituída de três dedos incluindo o polegar [20]. Salisbury [21] desenvolveu a garra robótica em formato de mão denominada mão Stanford/JPL com o intuito de ser utilizada em laboratório com fins educacionais, conforme Figura 2.1. A mesma foi projetada de forma modular, onde cada dedo possui três graus de liberdade, sendo movimentados através de servo-motores e cabos.

Na década de 90, Okada [22], Ulrich e Kumar [23], Rakic [24] e Jacobsen et al. [25] desenvolveram garras robóticas com características antropomórficas. Porém, a garra robótica desenvolvida por [21] a mais conhecida, pois, foi a primeira a ser publicada com características de mão humana. Os estudos desenvolvidos por ele serviram de base para a concepção das mãos robóticas que conhecemos atualmente, possibilitando maior destreza relacionada as garras robóticas, po-

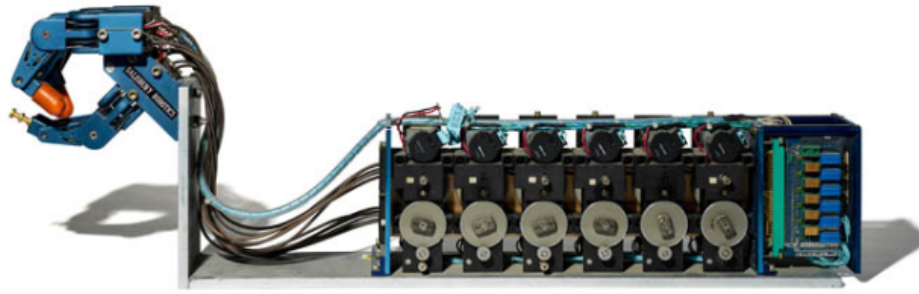


Figura 2.1: Garra robótica com características antropomórficas desenvolvida por Salisbury
Fonte: [21]

dendo ser utilizadas em manipuladores robóticos para manipulação de diferentes tipos de objetos ou servindo de prótese de mão humana para pessoas que possuem a restrição do membro [26].

Groover [2], destaca as garras robóticas antropomórficas como uma inovação e avanço tecnológico, pois, elas se diferenciam das garras robóticas tradicionais, tendo maior destreza. Enquanto as garras robóticas tradicionais têm um design simples e são projetadas para tarefas de pegar e largar, as garras robóticas antropomórficas podem manipular uma variedade de tipos de objeto.

O design das garras robóticas antropomórficas são muito particulares e depende, em muito dos casos, do campo de atuação que a garra encontra-se. Na literatura é encontrado garras robóticas desde dois dedos até 5 dedos. Alba e Ponticelli [27] demonstram a preferência pelo número de dedo onde as garras com 5 dedos, são as que possuem maior aceitação, conforme observado na Figura 2.2. Para Matsuoka [28] enquanto as garras robóticas de 2 dedos manipulam cerca de 40% dos objetos, uma garra com 4 dedos manipula em torno de 99% dos objetos, desde que, tenha o dedão como oposição do movimento. Entretanto, quanto maior o número de dedos, mais complexo torna-se o sistema de controle da garra.

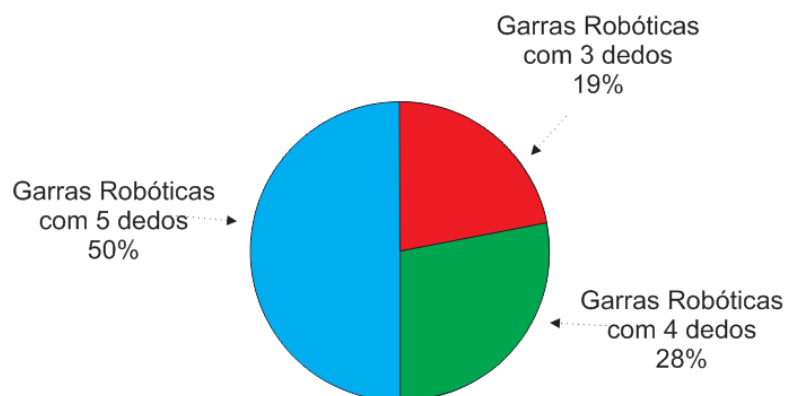


Figura 2.2: Número de dedos utilizados nas garras robóticas
Fonte: Adaptado [27]

A Tabela 2.1 mostra um comparativo das diversas garras robóticas já desenvolvidas, com destaque na literatura, em relação ao número de atuadores, número de dedos, Graus de Liberdade (GDL) e tipo de atuador.

Os atuadores são os mecanismos responsáveis por fazerem os transmissores dos dedos movimentarem para possibilitar a abertura e o fechamento. Os atuadores podem ser elétricos, hidráulicos, pneumáticos, entre outros. Escrivà [29] descreve as vantagens e desvantagens de diferentes atuadores, sendo o mais utilizado, os atuadores elétricos por darem menos problema, além de serem fáceis de controlar. Alba e Ponticelli [27] descreve que 75% dos atuadores utilizados em garras robóticas são do tipo elétrico, sendo o restante distribuído em pneumáticos e outros.

Tabela 2.1: Comparativo em relação ao número de atuadores, número de dedos e graus de liberdade de diferentes garras robóticas encontradas na literatura

Garra Robótica	GDL	Número de Atuadores	Número de Dedos (3, 4 ou 5)	Ano	Tipo do Atuador
Mão Stanford/JPL [30][31]	9	9	3	1983	Motor Elétrico (DC)
Mão Utah/MIT [32][33]	16	16	4	1985	Cilindro Pneumático
Mão Belgrade/USC [34]	15	4	5	1988	Motor Elétrico (DC)
Mão Barret [35]	8	4	3	1988	Motor Brushless
Mão DLR I [36]	16	12	4	1997	Elétrico
Mão Dist [37]	16	16	5	1998	Elétrico
Mão Robonaut [38]	19	14	5	1999	Elétrico
Mão DLR II [39]	17	13	5	2001	Elétrico
Mão Tuat/Karlsruhe [40]	24	1	5	2000	Elétrico
Mão Ultralight [41]	18	13	5	2000	Fluído Flexível
Mão Gifu [42]	20	16	5	1999	Micro motor DC
Mão Variable Force [43]	10	5	5	2000	-
Mão Blackfingers [44]	22	36	5	2000	Atuador Mckibben
Mão RTR I [45]	9	3	3	2001	Micro motor DC
Mão Dexterous Robot [46]	16	5	4	2001	-
Mão Shadow [47]	23	23	5	2002	Músculo Pneumático
Mão Thing [48]	14	6	4	2002	Elétrico
High Speed Mulfigerd [49]	10	10	3	2003	Elétrico

Os mecanismos transmissores, fazem a efetiva movimentação da garra robótica. Eles podem ser comparados com os tendões do nosso corpo [50], que são os elementos responsáveis por manter o equilíbrio dinâmico e estático do corpo. Para o desenvolvimento das garras robóticas antropomórfica, ele se torna o elemento crucial para possibilitar a movimentação dos dedos. Dentre os mecanismos, os tendões são os mais simples permitindo os atuadores ficarem fora da mão.

As engrenagens são mais precisas, porém, possuem mais fricção, enquanto a transmissão direta pode ser a melhor, porém, mais difícil de se projetar e menos usual. Na Figura 2.3 é comparado os diferentes mecanismos transmissores nas garras robóticas com destaque para os tendões, devido às características aqui citadas.

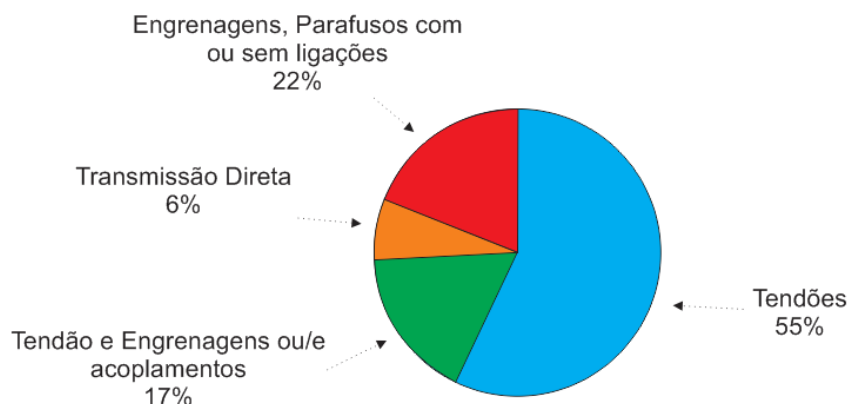


Figura 2.3: Comparativo dos diferentes mecanismos utilizados em garras robóticas

Fonte: Adaptado [27]

Devido o fato das garras robóticas antropomórficas crescerem substancialmente a partir dos anos 90, na linha do tempo da Figura 2.4 é apresentado o desenvolvimento cronológico relacionado aos eventos mais marcantes nas últimas décadas que mudaram a forma como vemos as garras robóticas atualmente. A partir da década de 1990, devido ao desenvolvimento e evolução da eletrônica e computação, as garras robóticas ganharam grande destaque no meio acadêmico e industrial. Atualmente, com o desenvolvimento de dispositivos eletrônicos e atuadores cada vez menores, com o desenvolvimento de sistemas embarcados cada vez mais eficientes à possibilidade de criação de garras robóticas menores, com maior versatilidade, menor custo e mais sensíveis que as garras robóticas de antigamente, crescem rapidamente.

2.2 Garras Robóticas

Conforme apresentado no capítulo anteriormente, as garras robóticas antropomórficas ganharam grande destaque. Por assemelhar-se a forma da mão humana, possibilitam o manuseio de diferentes objetos, diferente das garras robóticas de dois dedos que possibilitam apenas operações de pegar e largar. As garras robóticas, podem ser denominadas de órgão terminal ou elemento terminal de um robô. Rosário [51] compara as garras robóticas com a mão humana, conforme apresentado na Figura 2.5.

Nesta classificação, as garras de dois dedos são o tipo mais comum de garra, apresentando grande variedade, podendo ter movimentos paralelos ou movimentos rotativos. Devido à limita-

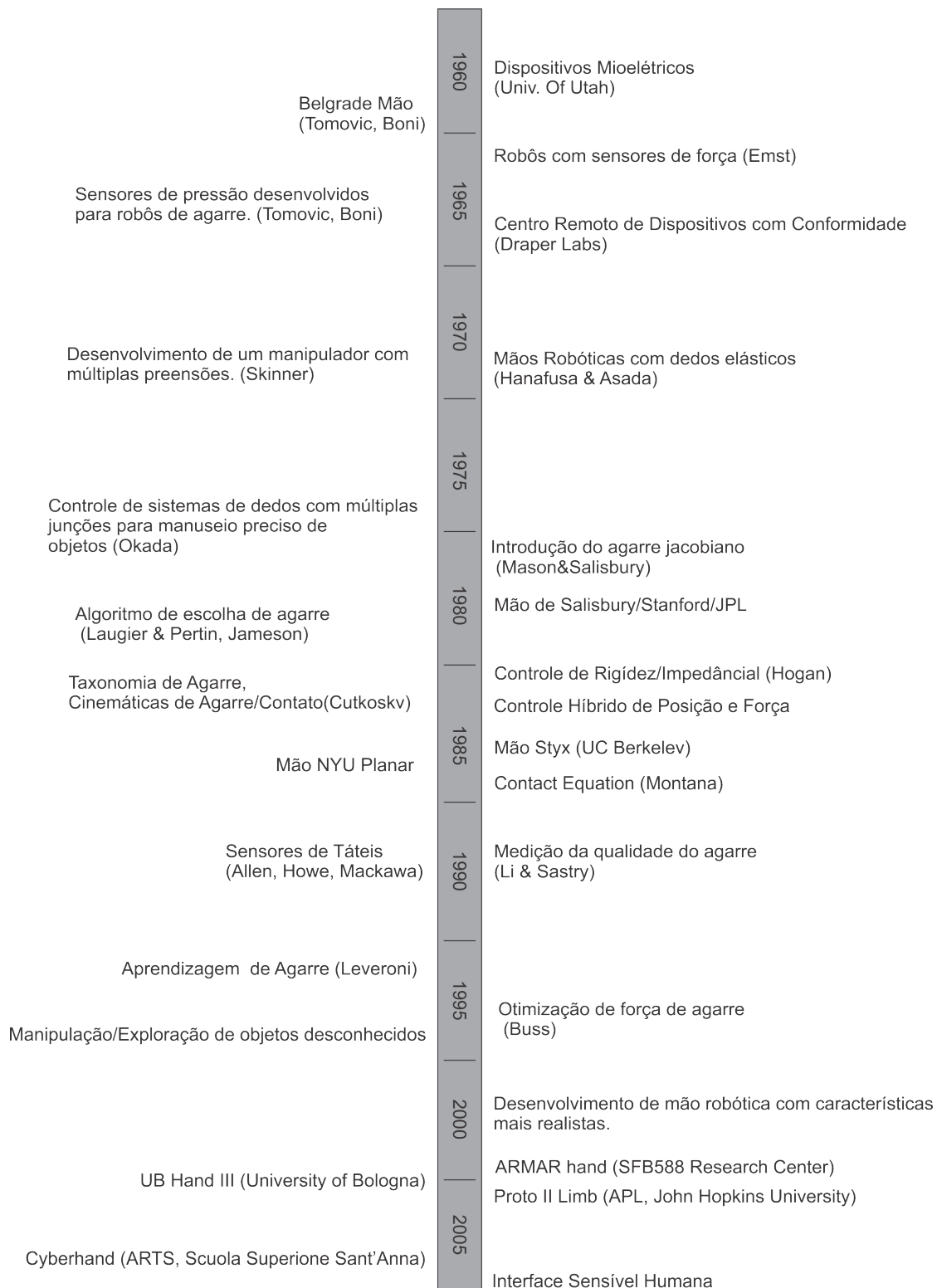


Figura 2.4: Linha do tempo dos eventos mais marcantes relacionados as garras robóticas

Fonte: Adaptado [27]

ção da abertura e fechamento dos dedos, acaba restringindo a movimentação, onde o objeto não pode exceder a abertura máxima da garra robótica. Já as garras de três dedos permitem a pre-

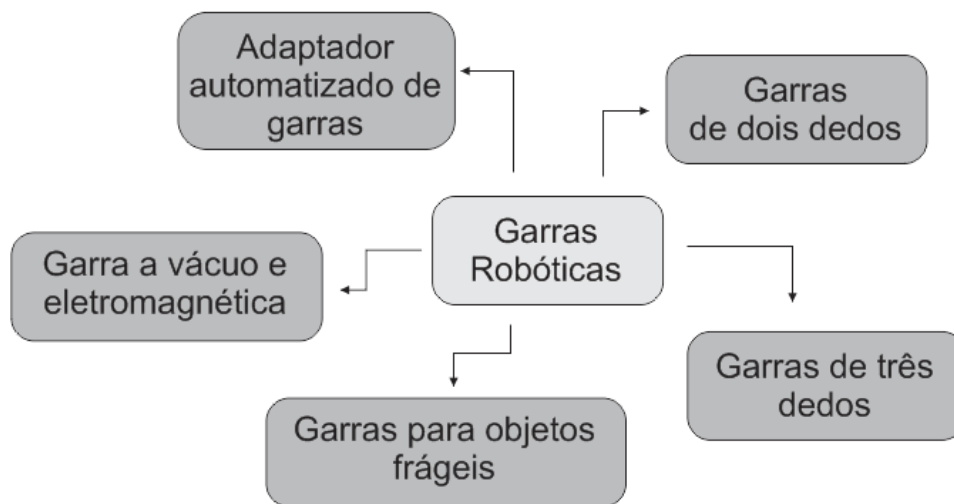


Figura 2.5: Classificação das garras robóticas

Fonte: Adaptado [51]

ensão mais firme de objetos com forma circular, triangular e irregular. Para as garras de objetos cilíndricos, como o próprio nome já está dizendo, só permitem segurar objetos cilíndricos, sendo sua principal desvantagem a sua movimentação limitada. As garras de objetos frágeis, exercem um certo grau de força para manipulação do objeto para não danificá-lo. As garras articuladas são utilizadas para manipulação de objetos de diferentes tamanhos e forma, enquanto as garras a vácuo e eletromagnética, funcionam com a sucção do ar e através da geração de um campo eletromagnético, respectivamente. Os adaptadores automatizados, surgiram devido à necessidade das garras robóticas realizarem as mais diferentes categorias de prensão, onde permite acoplar diferentes sistemas de prensão ligados ou removidos rapidamente do elemento terminal do robô.

Rosário [51], destaca a importância do desenvolvimento de garras robóticas, considerando os diferentes tipos de garras existentes no mercado: "...verifica-se que o desenvolvimento e a produção de garras fazem parte de um estágio importante no projeto de robôs para tarefas particulares. Em geral, os fabricantes vendem robôs desprovidos do atuador; garras e ferramentas são escolhidas e adaptadas pela equipe de engenharia que instala o robô no local de trabalho. Trata-se de uma etapa crítica da instalação, por requerer alto nível de conhecimento e prática".

Considerando que os robôs não são vendidos com garras robóticas, sendo as mesmas, específicas para cada tarefa. Conforme dados da federação internacional de robótica, mais de 3 milhões de robôs ainda serão usados nas fábricas ao redor do mundo até 2020, conforme demonstrado na Figura 2.6. Sendo assim, há necessidade pelo desenvolvimento de uma garra robótica que atenda ao maior número de aplicações, torna-se fundamental.

Diferente da classificação de [51], Lionel et al. [20] classificam as garras robóticas quanto ao número de atuador (n_{act}) em relação ao número de graus de liberdade da garra robótica (n_{gdl}).

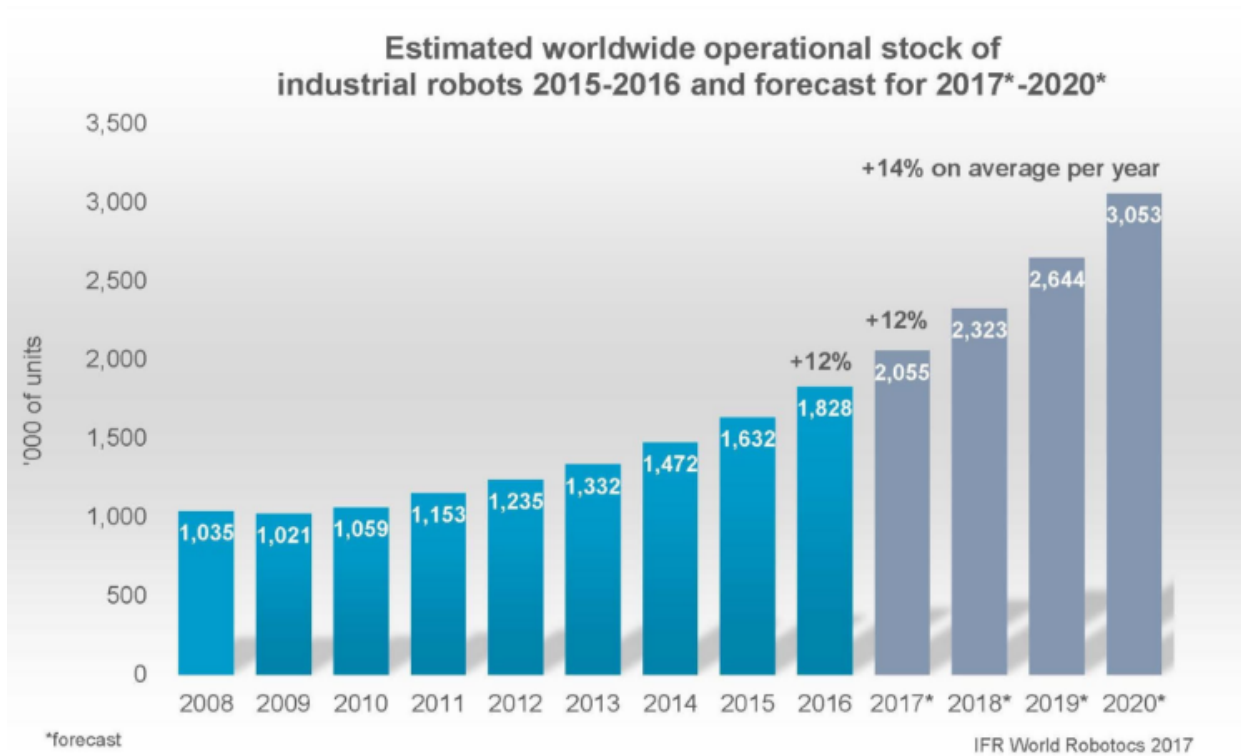


Figura 2.6: Crescimento dos robôs no setor industrial até 2020, conforme a federação internacional de robótica

Fonte: [52]

Caso $n_{gdl} = n_{act}$ a garra é totalmente atuada, caso $n_{gdl} > n_{act}$ ela é considerada sub-atuada. As garras totalmente atuadas, são muito complexas requerendo um custo computacional muito alto, que foge ao escopo desse trabalho. Enquanto as garras sub-atuadas podem ser mais simples se comparadas as totalmente atuada, conforme descrito na seção seguinte.

2.3 Garra Robótica Sub-Atuada

Devido à estrutura de composição da mão humana, que segundo estudos de Cunha et al. [53] possui 23 GDL, conforme Figura 2.7 adaptado por [54], tornando-se um órgão do corpo humano muito complexo. A fim de se obter as características de manipulação de objetos como a mão humana, as garras robóticas sub-atuadas possuem a característica de diminuir o número de atuadores necessários para movimentar os dedos, mantendo o mesmo número de graus de liberdade do mecanismo.

Os mecanismos sub-atuados são constituídos com elementos passivos que tem características de se moldarem a forma do objeto a ser manipulado. Conforme Lionel et al. [20], devido à automatização dos sistemas de manufatura são cada vez mais empregados sistemas de manipulação

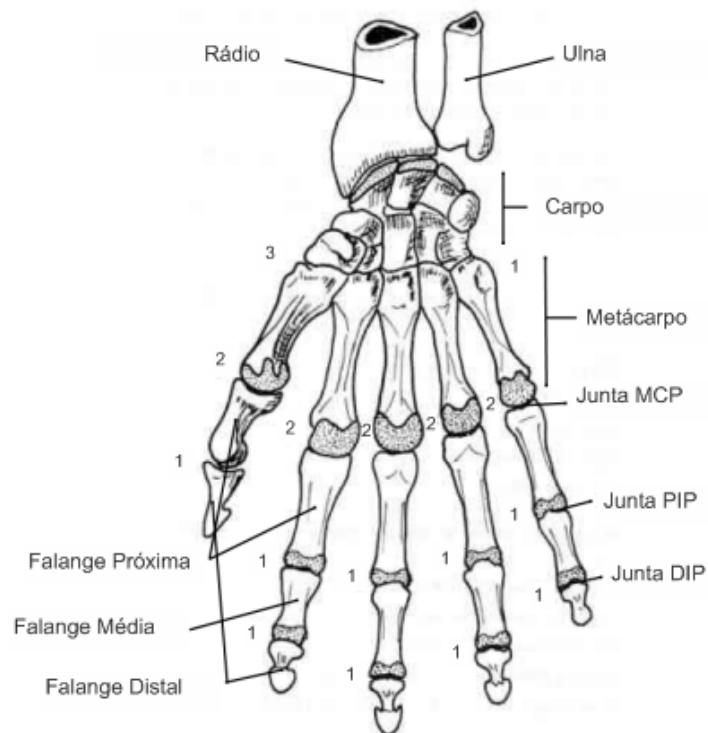


Figura 2.7: Anatomia da mão humana. Os números indicam a quantidade de GDL da mão.

Fonte: Adaptado [54]

para reduzir os custos e tempo de fabricação de peças nas células de trabalho.

Com o intuito de exemplificar o funcionamento de um mecanismo sub-atuado, temos como exemplo o mecanismo da Figura 2.8, desenvolvido por [55]. Ele consiste de uma garra robótica com dois dedos, onde cada dedo possui 3 falanges, movido por 1 atuador na base da garra robótica. O dedo é constituído de elementos elásticos, para possibilitar o retorno dos conjuntos de juntas e elos para posição normal, sendo melhor ilustrado na representação por barras da Figura 2.9.

Na Figura 2.9 é possível observar a representação por barras do mecanismo, onde através da movimentação do dedo é possível verificar a deformação dos conjuntos de juntas e elos para se adaptarem a forma do objeto, possibilitando assim, um agarre efetivo.

Outros mecanismos sub-atuados podem ser encontrados na literatura que foram utilizados em garras robóticas clássicas e que podem ser encontrados até os dias atuais. Por exemplo, o mecanismo desenvolvido por [56], representado na Figura 2.10, que consiste em tendões e polias. É importante observar, que para o mecanismo ser considerado sub-atuado, as polias e as articulações devem estar no mesmo eixo e não rigidamente acoplados sendo livre de giro. Caso contrário, o mecanismo só terá um grau de liberdade [20].

Os mecanismos sub-atuados possuem como vantagem uma estrutura elétrica simples, com baixo peso, tamanho, arquitetura de controle simples, baixo número de atuadores e um custo



Figura 2.8: Demonstração do mecanismos de uma garra robótica sub-atuada de dois dedos

Fonte: [55]

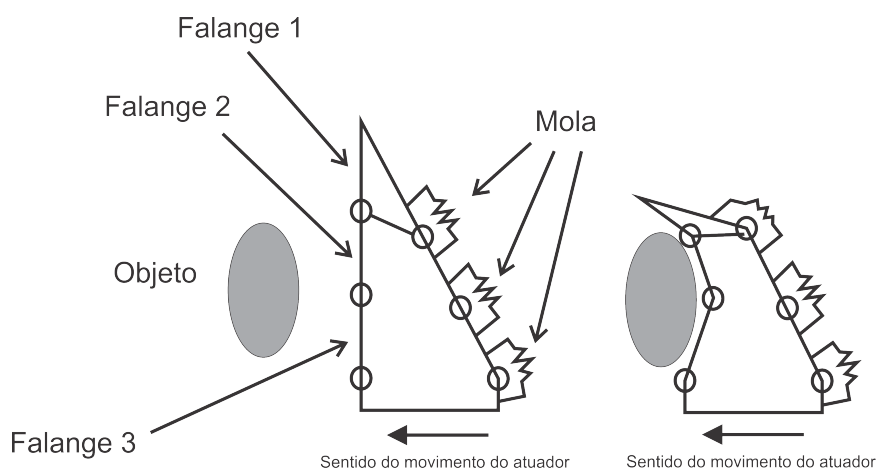


Figura 2.9: Representação de barras da garra robótica sub-atuada

Fonte: Adaptado [55]

benefício bem maior do que os mecanismos totalmente atuado, conforme [57]. Entretanto, para possibilitar o desenvolvimento de garras robóticas antropomórficas, antes precisamos conhecer a estrutura da mão humana, conforme descrito na seção seguinte.

2.4 Mão Humana

Para uma garra robótica ser considerada antropomórfica, essa deve possuir as características e propriedades relacionadas à mão humana, tais como: tamanho, forma, desempenho, GDL e peso. Pode ainda, possuir aspectos tácteis, como sentir o objeto a ser manipulado, sensação térmica, etc. A mão humana consiste em 4 dedos (mínimo, anelar, médio e indicador) e 1 polegar,

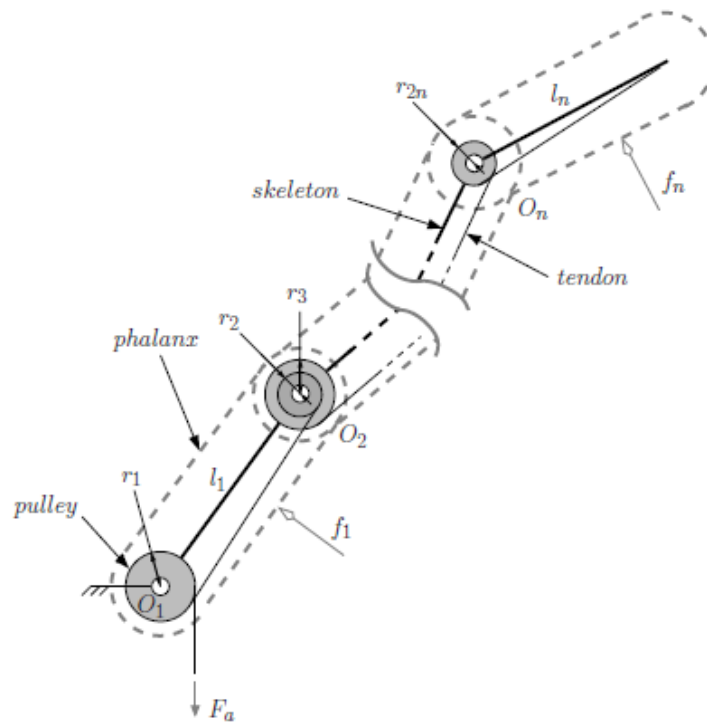


Figura 2.10: Mecanismos sub-atuado com tendões e polias

Fonte: [56]

sendo a posição dos dedos um ao lado do outro e o polegar na posição perpendicular. Como dito anteriormente a mão humana possui 23 GDL, permitindo mudar de forma e se adaptar a diferentes objetos. A palma é composta por 5 ossos metacarpianos conectados aos dedos e ao polegar pelo pulso. Cada dedo é composto por 3 ossos longos chamados falanges, cujos nomes são dados de acordo com sua distância da palma da mão: proximal, médio e distal, representado na Figura 2.11.

Dentre as articulações da mão humana, podemos destacar as articulações para possibilitar o desenvolvimento da garra robótica antropomórfica. Sendo elas: Metacarpofalângica (*Metacarpophalangeal - MCP*), relacionada ao metacarpo, a articulação Interfalângica Distal (*Distal Interphalangeal - DIP*), a articulação Interfalângica Proximal (*Proximal Interphalangeal - PIP*) e a articulação Radiocarpática (*Radio-Carpal - RC*).

No trabalho desenvolvido por Feix [59] é demonstrado o estudo taxonômico de agarres para garras robóticas antropomórficas. Ele demonstra o estudo taxonômico de agarres baseado em diferentes autores. O autor classifica os agarres em dois grandes grupos, sendo de precisão e força, baseado em objetos com forma prismática e circular. Na Figura 2.12, pode-se observar algumas formas de agarre baseado na mão humana.

Em estudos mais recentes desenvolvido por Bullock et al. [60] são apresentados 10 formas de agarres mais comum utilizados no dia a dia por empregadas domésticas e dois maquinistas. Os estudos tiveram como objetivo, auxiliar no desenvolvimento de garras robóticas e protéticas mais eficientes. Os 10 tipos mais comuns são mostrados na Figura 2.13, podendo ser classificados em:

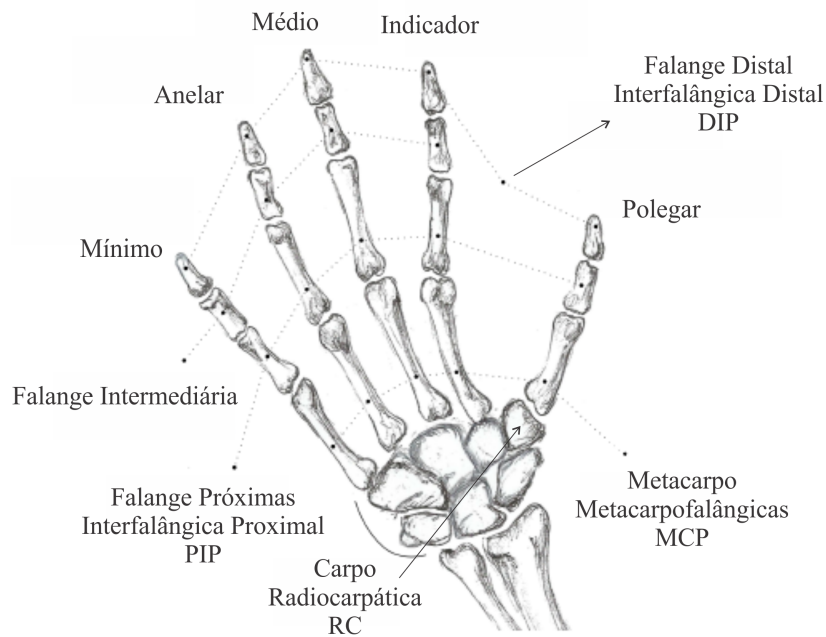


Figura 2.11: Estrutura da mão humana
 Fonte: Adaptado [58]



Figura 2.12: Formas comuns de agarre de objetos baseadas na mão humana, sendo da esquerda para direita: agarre prismático, circular, esférico e cilíndrico e precisão planar
 Fonte: [59]

medium wrap, precision disk, lateral pinch, tripod, lateral tripod, power shere, thumb-2 finger, index finger extension, light tool, thumb-3 finger. Nos trabalhos de Mahmoud e Aníbal [61] a garra robótica antropomórfica desenvolvida manipula as 10 formas satisfatoriamente, sendo que para outros movimentos, o mecanismo apresenta restrição de movimento do polegar.

O modelo da garra antropomórfica desenvolvida conforme a metodologia proposta, possui 15 GDL, sendo 3 por cada dedo, onde os GDL referente ao dedo polegar são para possibilitar os movimentos de adução e abdução, conforme apresentado na Figura 2.14. O estudo da mobilidade da garra robótica antropomórfica bem como a análise cinemática, fundamentais para movimentação da garra robótica, serão descritos posteriormente no projeto detalhado.



Figura 2.13: 10 tipos mais comuns de agarres usados no dia a dia de uma empregada doméstica e dois maquinistas

Fonte: [60]

2.5 Controle de Força em Garras Robóticas

Conforme descrito por Groover [2], a realimentação sensória nos dedos, possibilita a garra sentir a peça que será trabalhada e aplicar uma força limitada específica, evitando assim, a deformação da mesma. Diante de situações inesperadas e ambientes poucos conhecidos os efetadores finais, nesse caso as garras robóticas, apresentam dificuldade de manipulação. Para superar essa limitação, a utilização de sensores de força, baseado em controle automático, pode ser empregado

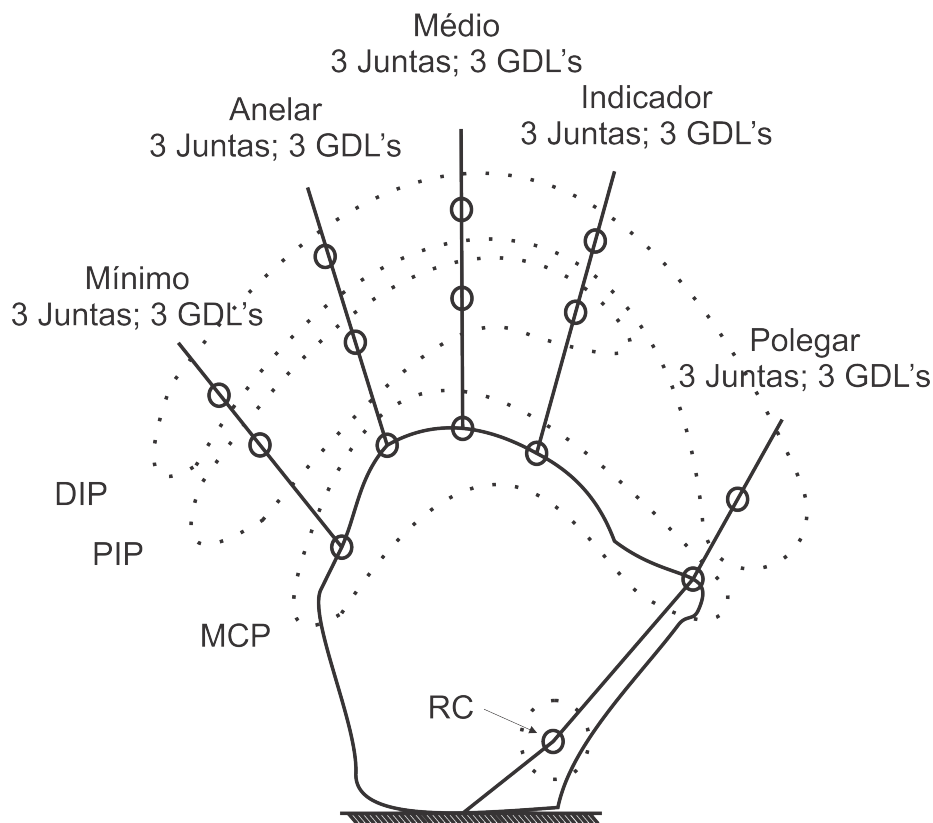


Figura 2.14: Disposição das juntas e GDL referente ao modelo proposto

em situações onde não se conhece o ambiente e o objeto a ser manipulado. Segundo Liu et al. [62] o controle de força tornou-se um problema importante nas operações de manipulação robótica que envolva precisão. A solução desse problema emprega técnicas de controle, assim como, sensores de força e/ou posição para realizar a medição.

Dentre as técnicas que são empregadas, o controle de conformidade, tem como vantagem a replicação com segurança dos movimentos das mãos durante a interação com o objeto, onde haverá a interação dos dedos da garra com um objeto sem danificá-lo. Sendo assim, o controle de conformidade pode ser definido como a tolerância de desvio de sua posição de equilíbrio, dependendo das forças externas aplicadas [63].

Jalani [63] descreve um modelo genérico massa-mola de controle de conformidade para uma garra robótica, conforme ilustrado na Figura 2.15, onde a massa m_r representa o movimento efetivo da massa do atuador. A mola b_r para dar o modo rígido apropriado ao robô não conectado. O sensor tem rigidez k_s e amortecimento b_s , sendo a peça de trabalho mostrada como um estado fundamental. O atuador da garra robótica é representado pela força de entrada F e a variável de estado x_r mede a posição da massa do robô.

A equação do sistema dinâmico proposto por Jalani [63] pode ser descrita conforme a função de transferência da Equação 2.1. Esse modelo pode ser usado para ajustar a conformidade do

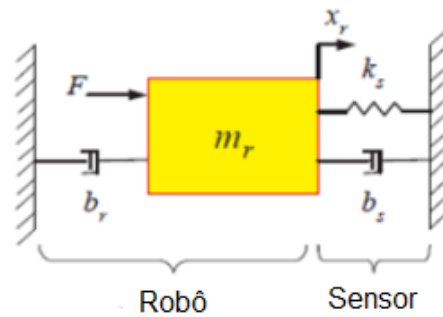


Figura 2.15: Modelo genérico de controle de conformidade
 Fonte: Adaptado [63]

sistema, alterando a rigidez do sistema e também do amortecimento.

$$\frac{X_r(s)}{F(s)} = \frac{1}{[m_r s^2 + (b_r + b_s)s + k_s]} \quad (2.1)$$

Dentro do controle de conformidade, existem dois métodos propostos na literatura, que são o controle de conformidade passivo e ativo. O controle de conformidade passivo, não usa dados dos sensores de força e dependem do design mecânico da garra robótica. Já o segundo, controle de conformidade ativo, usa o atuador F para modificar o modelo do sistema, dependendo apenas da estratégia de controle empregada. As estratégias de controle empregadas, podem ser proporcional-derivativo, proporcional-integral, controle *fuzzy*, neuro *fuzzy*, ou também incorporando dados do sensor. O sensor é utilizado para medir força, possibilitando moldar a resposta do sistema.

Segundo Lefebvre et al. [64], o controle de conformidade passivo aplica o elemento de elasticidade da estrutura mecânica para gerar tarefas de conformidade na garra robótica. Enquanto o controle de conformidade ativo, usa o método de *feedback* de força. Na seção seguinte é demonstrado a estratégia de controle *fuzzy* como alternativa para implementação do controle de conformidade ativo.

2.6 Estratégia de Controle *Fuzzy*

Tratando-se de teorias de controle clássico e moderna, o primeiro passo para implementar o controle de um processo, é derivar o modelo matemático que descreve o processo. Para isso é necessário conhecer detalhadamente o processo a ser controlado, onde muitas das vezes nem sempre é possível. Um dos exemplos dessas teorias de controle é o PID (Proporcional, Integral e Derivativo), diante de sistemas não-lineares e imprecisos, necessita de ajustes periódicos dos parâmetros do controlador. Mesmo que haja diversos métodos de ajustes de PID como Ziegler-Nichols e Cohen-Coon, diante de processos altamente não-lineares ou quando a modelagem ma-

temática da planta for muito complexa, os controladores PID não tem desempenhos satisfatórios [65].

Nas garras robóticas sub-atuadas, por causa da fricção, dos elementos elásticos, das folgas e outras incertezas relacionadas, torna-se extremamente difícil modelar com precisão as garras robóticas sub-atuadas [66]. Sendo assim, muitos trabalhos utilizam o controle de posição integrado com o controle de força, precisando de mais sensores nos dedos da mão e um sistema de controle complexo para prover a retroalimentação do sistema, conforme os trabalhos de [36] e [67]. O controle de posição faz-se necessário, porque, enquanto a mão do robô e o objeto não entram em contato, o algoritmo de controle de posição controla o dedo ao longo da trajetória desejada para alcançar a posição estipulada.

O controle *fuzzy* é uma técnica descrita por Lee [68], em que considera a falta de exatidão e a incerteza do sistema. Devido às técnicas descritas atualmente não exigirem cálculos matemáticos complexos, torna-se viável a utilização em computadores, microcontroladores e processos em tempo real. A teoria dos conjuntos *fuzzy* relacionam as entradas e saídas, agregando vários parâmetros de processo e controle, permitindo um resultado com desempenho estável e robusto para processos complexos [69].

Na Figura 2.16 observa-se a representação dos sistemas de controle convencionais. Para os sistemas de controle convencionais, a modelagem matemática é a planta ou processo que está sendo controlado. A modelagem consiste na identificação dos sistemas, sendo caracterizado por um conjunto de equações diferenciais. Através dessas equações são obtidos os parâmetros a serem ajustados para um determinado comportamento do sistema. Enquanto na Figura 2.17, por outro lado, é demonstrado uma típica representação dos sistemas onde não são empregados controle automático. Observa-se que os operados humanos são responsáveis por efetuarem o controle do sistema, mediante as circunstâncias do sistema, diferente dos controles convencionais. Essa diferença que torna os controladores *fuzzy* tão efetivos [65].

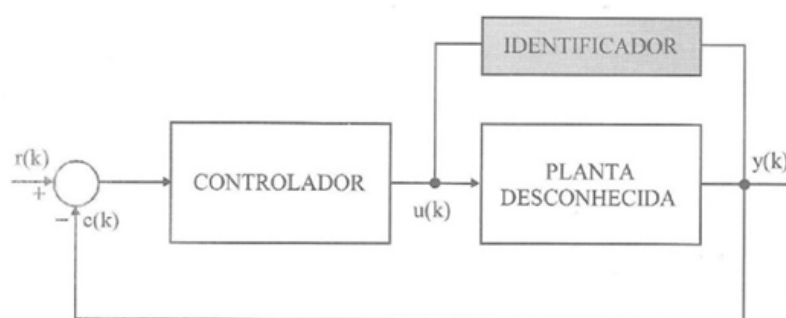


Figura 2.16: Modelo de representação de um sistema de controle tradicional

Fonte: [65]

A utilização da lógica *fuzzy* é baseada na experiência e experimentos do especialista da planta, em vez de modelos matemáticos bem definidos. Dentre os controladores *fuzzy* utilizados, o con-

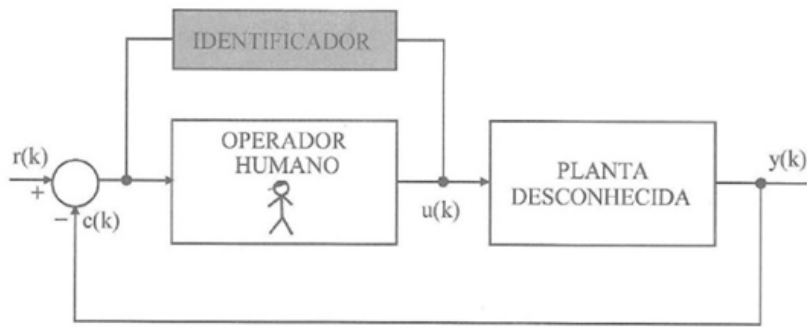


Figura 2.17: Modelo de representação de um sistema de controle *fuzzy* típico

Fonte: [65]

trolador *fuzzy* baseado em regras é um dos mais populares usados em softwares de sistemas de desenvolvimento, devido serem de fácil compreensão pelo pessoal da manutenção; podem ser testados de forma individual combinando-se para formar uma estrutura de controle complexo e tanto funções de controle lineares quanto não-lineares podem ser implementados por um sistema baseado em regras, usando apenas o conhecimento de um especialista [65].

2.6.1 Configuração Básica de um Controlador *Fuzzy*

Um controlador *fuzzy* é composto por 4 blocos funcionais: Interface de *fuzzificação*, Base de conhecimento, Motor de inferência e Interface de *defuzzificação*. Os blocos podem ser descritos de forma resumida da seguinte forma, conforme apresentado na Figura 2.18:

Interface de *fuzzificação*: Nesse bloco, os valores que são obtidos através de sensores das grandezas físicas a serem medidas são convertidos em um intervalo de 0 a 1 que são associados a rótulos linguísticos, por exemplo, *pressão_alta*, *pressão_baixa*, *pressão_média*, etc.

Base de conhecimento: A base de conhecimento, representa o “cérebro” do controle *fuzzy* baseado em regras, onde são armazenados as regras *fuzzy* linguísticas denominadas de funções de pertinência, sendo essas regras, o conhecimento do especialista na área de controle do processo que são armazenados em forma linguística.

Motor de inferência: Nessa parte, são gerados as ações de controle a partir da entrada do sistema. As ações de controle podem ser do tipo *Se X então Y* para simular as tomadas de decisão humanas.

Interface de *defuzzificação*: A *defuzzificação* consiste em traduzir as ações de controle de forma linguística para controle concreto do mundo real.

De maneira geral, um controlador *fuzzy* baseia-se em regras *Se X então Y*. Sendo que no controlador *fuzzy*, há várias destas regras *fuzzy*, e são todas ativadas em paralelo. Essa operação em paralelo é o que garante aos controladores *fuzzy* sua alta velocidade de processamento. Conforme

retratado por [65] em um controlador *fuzzy* industrial típico, de 3 entradas e 1 saída, com 80 regras, o tempo de ciclo para varrer a estrutura de regras pode levar menos de um milissegundo.

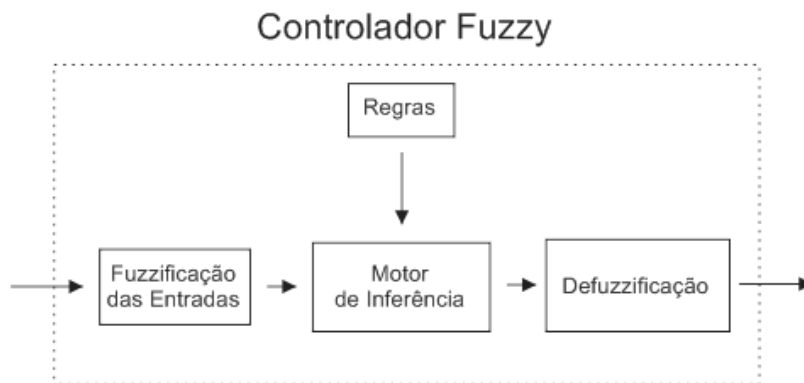


Figura 2.18: Blocos funcionais de um controlador *fuzzy*

2.6.2 Funções de Pertinência *Fuzzy*

As funções de pertinência *fuzzy* são responsáveis por atribuir valores discretos de uma variável. A Figura 2.19 mostra três funções de pertinência, onde o eixo vertical representa o intervalo de $[0,1]$ e o eixo horizontal o universo de discurso. Às três funções demonstradas são: triangulares, trapezoidal e *fuzzy-tom*. Na literatura existem outros modelos de funções de pertinência para aplicações específicas. A quantidade de funções em um universo de discurso e seu formato são escolhidos com base na experiência, na natureza do processo a ser controlado, ou numa entrevista com um operador humano especializado, que realize as funções de controle manualmente.

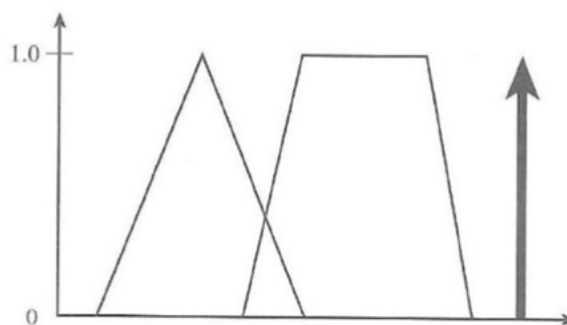


Figura 2.19: Representação das principais funções de pertinências

Fonte: [65]

Simões [65] dá algumas dicas práticas para encontrar os melhores resultados (ou quase) conforme as funções de pertinências utilizadas. Primeira: um número prático de funções de pertinência é algo entre 2 e 7. Segunda: quanto maior o número de conjuntos, maior a precisão, mas a demanda computacional também é mais significativa. Terceira: os formatos mais frequentes

encontrados são triângulos e trapezoides, pois, são gerados com facilidade. Em casos onde um desempenho suave é de importância crítica, funções do tipo, gaussiana, sigmoide e *spline* cúbico podem ser usadas.

O mapa de regras *fuzzy* são funções de pertinência definidos como, por exemplo: PA (Positivo Alto), PM (Positivo Médio), PB (Positivo Baixo), MA (Médio Alto), (MM) Médio-Médio, etc. Como já foi dito anteriormente, o mapa de regras *fuzzy* é a base de conhecimento, o depósito da inteligência. Na Figura 2.20 é demonstrado um mapa de regras para um dado exemplo.

	NM	NS	ZE	PS	PM
NM					
NS					
ZE					
PS					NS
PM					

(Variável 1 = linhas, Variável 2 = colunas)

Figura 2.20: Mapa de regras para *fuzzificação*

Entretanto, para que os sistemas de controle possam ser efetivo, seja com estratégia de controle *fuzzy*, PID, Redes Neurais, entre outros, antes é necessário conhecer os tipos de sensores envolvidos em aplicações industriais, sejam eles, sensores de força, temperatura, pressão, nível, entre outros. Eles têm como função mandar as informações para os sistemas de controle, realimentando a malha de controle. No caso das garras robóticas, os sensores de força é uma das formas de prover as informações táteis para evitar que a garra robótica danifique o objeto. Sendo assim, eles são descritos com maiores detalhes na seção seguinte.

2.7 Sensores de Força

Para sentir a força aplicada no objeto, os sensores táteis são empregados para possibilitar os movimentos de agarrar, segurar e manipular com segurança. Exemplos de sensores táteis disponíveis comercialmente podem ser observados nos trabalhos desenvolvidos por Townsend [70] e Shadow [47], onde os mesmos empregam os sensores táteis desenvolvidos pela empresa Syntouch [71]. Em um mesmo sensor tátil é integrado as informações de temperatura, força e vibração, para isso, são usados termistores, eletrodos sensíveis de impedância e hidrofones para ler as informações, possibilitando que o robô manipule objetos com extrema destreza [72]. As garras robóticas com os sensores táteis desenvolvidos, podem ser observadas na Figura 2.21.

Devido ao acelerado desenvolvimento industrial e a passagem do desenvolvimento de robôs de ambientes estruturados para não-estruturados, tais como robôs com características autônomas, a necessidade de desenvolvimento de sensores para permitir os robôs entenderem o ambiente em

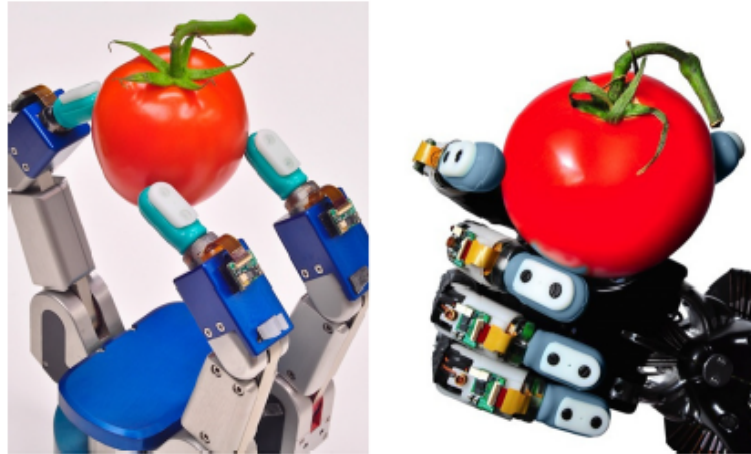


Figura 2.21: BarretHand e Shadow Hand utilizando os dedos com sensores táteis (em azul) desenvolvidos por [71]

Fonte: [72]

que estão inseridos torna-se crucial. A utilização de sensores táteis engloba atividades agrícolas, médicas, processamento de comida e outros serviços industriais [73]. Ao longo dos anos, sensores foram desenvolvidos, baseados em princípios de funcionamento diferentes, como: resistivo, capacitivo, óptico, ultrassônico, piezoelétrico, entre outros. Amed et al. [74] relacionam as vantagens e desvantagens de cada sensor de 1992 a 2013, resumidos na Tabela 2.2.

No trabalho desenvolvido por Sadun et al. [90] é utilizado um Sensor Resistivo de Força (*Force Sensitive Resistor - FSR*) em uma garra robótica para prover a medição de força. Entretanto, o mesmo tem como limitação a sua pequena área de contato, que mediante pequenos objetos apresentam problemas. O sensor resistivo de força é constituído de dois filmes de polímero e um espaçador, conforme demonstrado na Figura 2.22. Eletrodos interdigitantes são depositados em um polímero, enquanto o outro possui material semicondutor. Ao pressionar o sensor há um aumento da superfície de contato entre elas e conseqüentemente uma diminuição da resistividade do material [91]. Sendo assim, a resistência passa a variar inversamente com a força a ser aplicada. Quanto menor for a resistência, maior a força que está sendo aplicada ao sensor. A resistência no sensor resistivo de força é dado pela Equação 2.2.

$$R_{FSR} = \frac{k}{F} \quad (2.2)$$

Onde R_{FSR} representa a resistência apresentada pelo sensor, k é a constante de proporcionalidade e F é a força aplicada sobre o sensor.

Para possibilitar a energização dos sensores resistivos de força em microcontroladores é necessário a utilização de um divisor de tensão. Sendo que a saída de tensão no divisor de tensão é dado pela Equação 2.3.

Tabela 2.2: Comparação entre os diferentes sensores existentes

Tipo	Vantagens	Desvantagens
Resistivo [75, 76, 77]	Sensibilidade Baixo Custo	Alto Consumo de Energia Detecta apenas um ponto de contato
Piezoresistivos [78, 79, 80]	Baixo Custo Boa sensibilidade Baixo Ruído Eletrônica Simples	Frágil Resposta não linear Histereses Sensível a temperatura
Capacitivo [81, 82, 83]	Sensibilidade Baixo Custo	Histere Eletrônica Complexa
Óptico [84, 85, 86]	Fisicamente flexível Sensibilidade Rápido Sem conexões	Implementação Computacional Consumo de Energia
Magnético [87, 88, 89]	Alta sensibilidade Bom range dinâmico Fisicamente Robusto	Sofre interferência magnética Implementação Computacional Volumoso Alto consumo de energia

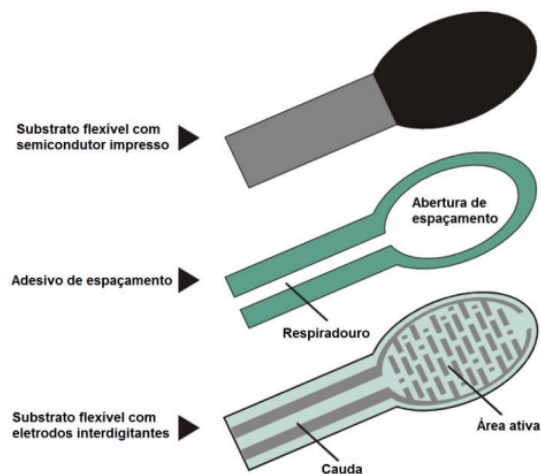


Figura 2.22: Vista explodida do sensor resistivo de força

Fonte: [92]

$$V_{out} = R_m \times \frac{V}{R_m + R_{FSR}} \quad (2.3)$$

Onde R_m é o divisor de tensão, V a tensão fornecida pelo microcontrolador e R_{FSR} a resistência variável do sensor. Se não existe força sobre o sensor, a resistência deste será alta, e ele

absorverá a maior parte da tensão do conjunto, de modo que a leitura, baseada na tensão sobre o resistor em série com o FSR, será baixa. Se existe força aplicada, a resistência do sensor diminui, e a tensão sobre o resistor R_m aumenta, assim como o valor lido pelo microcontrolador. O comportamento de sensor é mostrado na Figura 2.23.

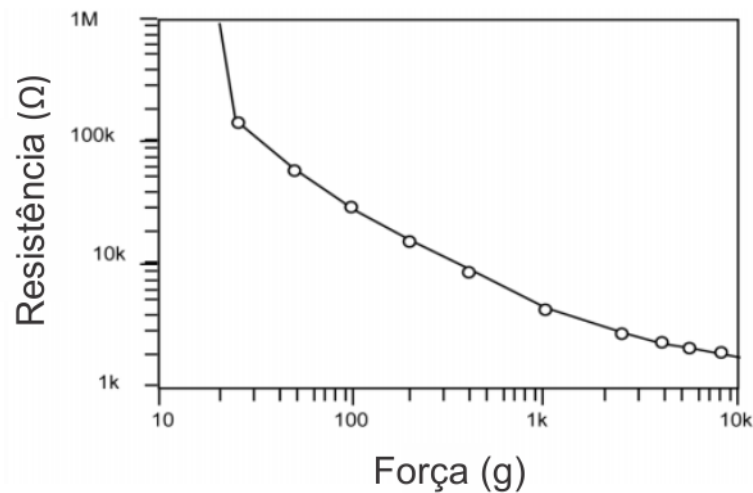


Figura 2.23: Comportamento do sensor resistivo de força
Fonte: Adaptado [93]

2.8 Processo de Desenvolvimento de Produto

O Processo de Desenvolvimento de Produto (*Product Development Process - PDP*) consiste em uma forma sistemática de concepção de um produto, para possibilitar a redução de tempo de lançamento, bem como a resolução rápida de problemas relacionado ao mesmo. Para Ulrich e Eppinger [12], o PDP consiste em um conjunto de etapas que uma empresa emprega para conceber, projetar e comercializar um produto. Para Kechinski [94], o processo de desenvolvimento de produto garante a qualidade e redução de custos do produto.

Conforme Ulrich e Eppinger [12] o PDP é uma atividade complexa, pois, na produção de um projeto completo, é agregado diversas áreas tais como: marketing, design e manufatura no ciclo de vida do produto desde a produção até o descarte. O processo de desenvolvimento de produto é importante além de outras razões, por possibilitar a garantia de qualidade, gerenciamento e aperfeiçoamento do produto. A garantia da qualidade é possibilitada pelas fases do projeto e os pontos de verificação ao longo do caminho. O gerenciamento possibilita ao gerente identificar possíveis áreas com problemas. Já o aperfeiçoamento do produto, da-se devido à documentação que é gerada ao longo do PDP, possibilitando identificar oportunidades de melhoria.

Os mesmos autores propuseram uma metodologia de desenvolvimento de produto genérico em que consiste em seis fases, conforme demonstrado na Figura 2.24, organizados em: Plane-

jamento, Desenvolvimento do Conceito, Projeto dos Sistemas, Projeto Detalhado, Teste/Refinamento e Produção *Ramp Up*. Cabe destacar, que em cada fase é passível de utilização de uma metodologia estruturada, para que possa dar subsídios no desenvolvimento do projeto.

Essas metodologias estruturadas têm como base auxiliar o levantamento de informações, estabelecer requisitos, propor soluções, combinar princípios de solução, avaliar concepções, entre outros. Como exemplo e que será descrito nas suas respectivas fases, temos a casa da qualidade, síntese funcional, método morfológico, métodos de seleção e uma metodologia para estruturar o projeto detalhado do desenvolvimento do produto, chamada modelo V.

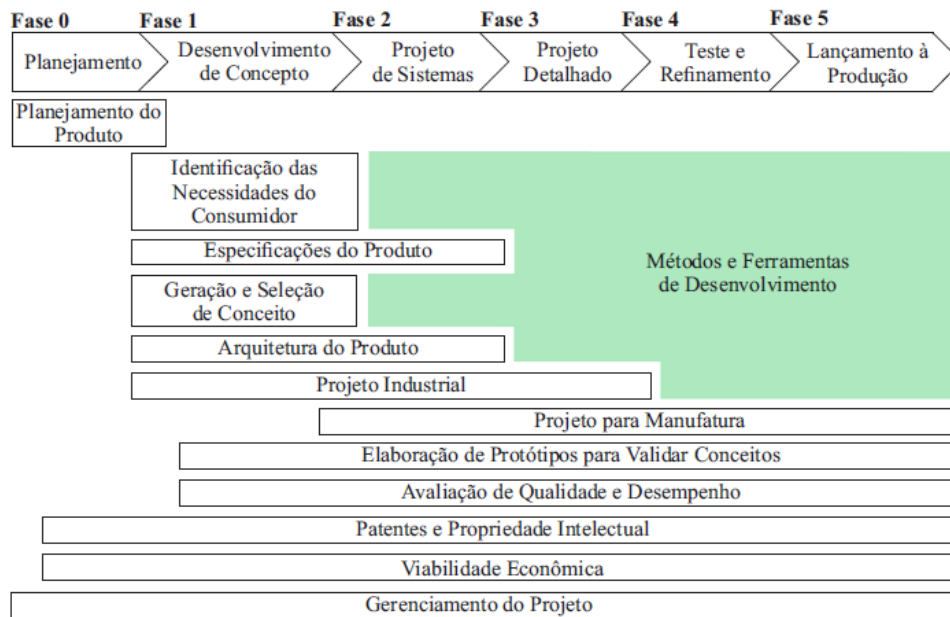


Figura 2.24: Modelo genérico de Desenvolvimento de Produto Proposto

Fonte: [12]

2.8.1 Fase 0 - Planejamento

A fase de planejamento procede a aprovação do projeto e o lançamento do processo de desenvolvimento, é nessa fase que irá se identificar e avaliar as tecnologias disponíveis, objetivos de marketing e restrições do projeto. A partir dos levantamentos feitos são documentados as oportunidades de projetos e projetos em potencial, que às vezes, são chamados de gerenciamento de portfólio ou plano de projeto que servirá como base para desenvolvimento do produto.

O plano de projeto, embora tenha uma estrutura linear, devido à realidade orçamentária para produção do produto, muitas das vezes é reavaliado, sendo atualizado com base nas informações mais recentes das equipes de desenvolvimento, laboratórios de pesquisa, organizações de produção, serviços e marketing. Onde o sucesso do desenvolvimento do produto, vai depender da capacidade da equipe em reavaliar o plano de projeto constantemente [12].

O planejamento do produto, como pode ser observado na Figura 2.24, envolve até o desenvolvimento do conceito do produto. O processo de planejamento de produto pode ser ilustrado na Figura 2.25, onde é dividido em 5 fases: identificação da oportunidade, avaliação e priorização dos projetos, alocação do recurso e time de desenvolvimento, completar o pré-projeto de planejamento e refletir o resultado do processo. Essas fases de maneira geral, coleta possibilidade para novas plataformas de produtos. Sendo os projetos em potenciais, avaliados com base na estratégia competitiva da empresa, documentados, avaliados os mercados-alvo, as partes interessadas do produto e os problemas críticos.

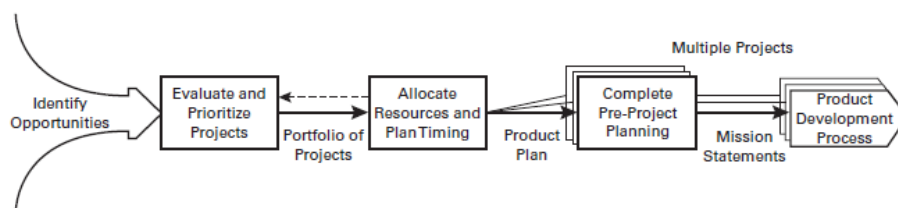


Figura 2.25: Processo de planejamento do produto

Fonte: [12]

2.8.2 Fase 1 - Desenvolvimento do Conceito

O desenvolvimento do conceito, consiste na identificação das necessidades dos clientes e à transformação em requisitos de produtos a fim de se obter as especificação meta do produto. A partir das especificação meta do produto são gerados os conceitos alternativos do produto e um ou mais conceitos são selecionados para desenvolver e testar.

Nessa fase, a imersão dos engenheiros e projetista industriais no problema do cliente é de fundamental importância, interagindo e experimentando o ambiente de uso do produto. Caso não haja essa imersão, o risco de desenvolvimento de um produto sem inovação tecnológica e soluções inovadoras para necessidades dos clientes, podem nunca serem descobertas [12]. Nessa etapa à geração de conceitos, seleção de conceitos, *benchmarking* competitivo e estabelecimento das especificações são parte integrantes da necessidade dos clientes. Uma ferramenta que auxilia o desenvolvimento dessa etapa é a utilização do Desdobramento da Função da Qualidade (*Quality Function Deployment - QFD*).

O QFD tem por objetivo o desenvolvimento de um novo produto ou melhora de um produto já existente, considerando as necessidades dos clientes. Na abordagem do QFD, a voz do cliente (necessidade) é traduzida em requisitos adequados para gerar a especificação meta do produto. O objetivo da utilização do QFD conforme Cheng [95] são duas: a 1^o está na busca, tradução e transmissão das necessidades do cliente em requisitos do produto e a 2^o é de garantir as necessidades do cliente ao longo do processo de desenvolvimento do produto. Na implementação, o QFD utiliza uma matriz conhecida como “casa da qualidade” para traduzir a necessidade dos

clientes em especificação meta do produto.

Com a casa da qualidade é possível estabelecer relações entre as necessidades dos clientes e os requisitos de projeto, sendo esse último, possível de ser hierarquizado com base nas especificações de projeto [96]. Para Fonseca [97], as necessidades dos clientes, são as necessidades brutas obtidas dos usuários que irão ou desejam utilizar o produto, essas por sua vez, serão levadas até o projetista que irá transformá-las em requisitos de projeto, tornando as especificações mensuráveis para que o produto possa satisfazer as necessidades dos clientes. Um modelo resumido do QFD é mostrado na Figura 2.26 adaptado por [98].

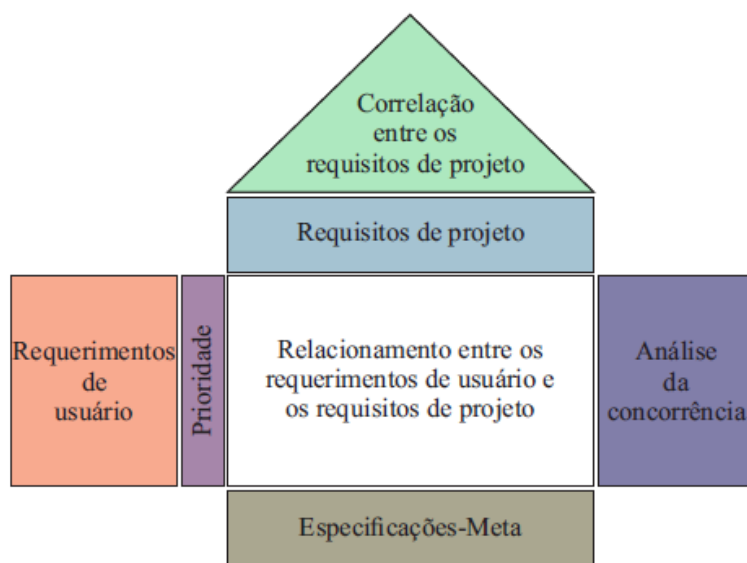


Figura 2.26: Modelo da Casa da Qualidade Resumido

Fonte: [98]

Os *requerimentos de usuários* (tratados nesse trabalho como necessidades dos clientes) na casa da qualidade são tratados em relação ao “O QUÊ” os clientes desejam ou esperam do produto para atender suas necessidades, sendo classificados conforme o ciclo de vida do produto. O campo *prioridade*, corresponde a avaliação quantitativa de cada cliente, onde pode-se empregar a ferramenta denominada de diagrama de Mudge e gráfico de Pareto para auxiliar a equipe de projeto. Os *requisitos de projeto*, são tratados com a relação “COMO” para atender aos requisitos do cliente e atributos específicos do produto. O *relacionamento entre os requisitos de usuário e os requisitos de projeto* são feitos na parte central da tabela, sendo projetados em uma escala quantitativa de (1-3-5). A *análise da concorrência*, serve para verificação dos produtos correlatos, a fim de verificar as vantagens e desvantagens em relação aos produtos já desenvolvidos. Na *correlação entre os requisitos de projeto* são feitas a comparação entre cada requisito para verificar se há relações fortes ou fracas entre cada requisito. Enquanto a *especificação meta*, destina-se a análise final da casa da qualidade, possuindo o resultado e hierarquia entre os requisitos propostos.

2.8.3 Fase 2 - Projeto ao Nível de Sistema

Na fase de projeto ao nível de sistema, faz-se a definição da arquitetura do produto e a decomposição do produto em subsistemas e em componentes. As soluções são realizadas através de métodos tanto intuitivos como sistemáticos. Os métodos intuitivos são baseados nas ideias e experiências do projetista, enquanto o método sistemático utiliza de ferramentas de projeto como a matriz de Pugh e matriz morfológica, que serão descritos no desenvolvimento da garra.

2.8.4 Fase 3 - Projeto detalhado

Na fase do projeto detalhado, é incluso toda parte de detalhamento do produto, incluindo parâmetros de geometria, das tolerâncias e dos materiais. Nessa fase, é gerado a documentação detalhada do projeto, bem como as partes que serão adquiridas, assim como, o custo total do projeto. Por se tratar de um produto mecatrônico, existe na literatura uma metodologia estrutura para auxiliar no desenvolvimento de produto, proposto pelo professor Gausmeier [19], através da Associação de Engenheiros Alemães (*Verein Deutscher Ingenieure - VDI*).

O modelo ficou conhecido como modelo V, onde o principal objetivo, era integração com métodos de projetos já concebidos, a fim de auxiliá-los no desenvolvimento de produtos mecânicos complexos. Uma das principais contribuições do modelo, consiste na denominação do modelo V, conforme apresentado na Figura 2.27. A análise do modelo V inicia-se com os *requerimentos de produtos*, realizando o projeto do sistema, através da *definição dos sistemas, subsistemas e definição dos componentes*, os quais serão definidos através da metodologia de desenvolvimento de produto utilizada. Simultaneamente, em cada etapa é realizado a *validação do sistema, integração dos subsistemas e integração dos componentes*, culminando nos domínios específicos referente ao produto desenvolvido, sendo eles: mecânicos, eletroeletrônicos e computacionais.

Mais especificamente, a parte central do modelo é desenvolvido o conceito de solução para cada um dos domínios específicos envolvidos. Nessa etapa, a elaboração de cálculos e modelos de análise são importantes para garantir o desempenho funcional das funções críticas do produto. Dentro do domínio mecânico são realizados cálculos cinemáticos, Desenho Assistido por Computador (*Computer Aided Design - CAD*), especificação de tolerâncias, etc. Quanto ao domínio eletroeletrônico são definidos e elaborados os esquemas elétricos, circuitos eletrônicos de potência, interfaces de comunicação, fonte de alimentação, etc. Por último, no domínio computacional ocorre a programação de interfaces de usuário, sistema de controle, etc.

2.8.5 Fase 4 - Testes e Refinamento

Na fase de testes e refinamento, são desenvolvidos os protótipos para teste. A partir dos resultados, são ajustados os parâmetros do produto e refinado as especificações para que ele atenda

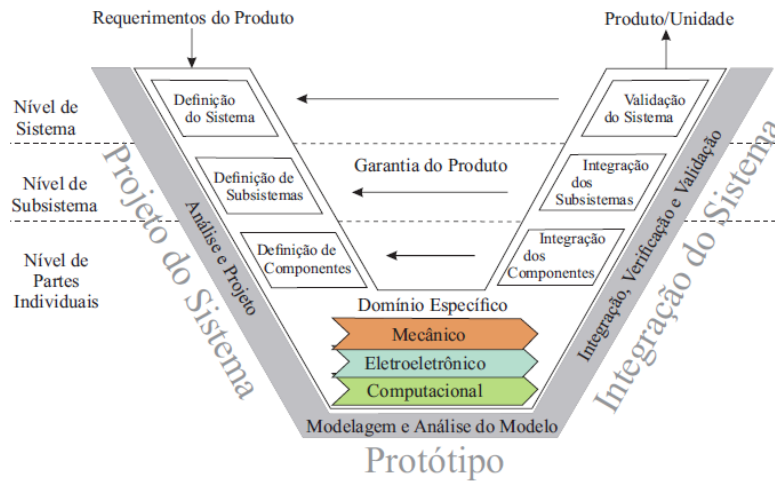


Figura 2.27: Modelo desenvolvido pelo professor Gausmeier
 Fonte: Adaptado [98]

da melhor forma os requisitos do cliente especificado na fase anterior. Atualmente com o desenvolvimento das impressoras 3D, essa fase tornou-se muito mais fácil de ser desenvolvida, pois, possibilita a criação de um protótipo funcional e de baixo custo.

2.8.6 Fase 5 - Lançamento à Produção

A última fase, lançamento à produção, tem a ver com a fabricação do produto final através de um determinado sistema de produção e sua colocação no mercado. A fase contempla todas aquelas atividades da cadeia de suprimentos totais, como as estratégias de venda e distribuição, formas de atendimento ao cliente, assistência técnica, publicidade e marketing. Dentro dessa fase, também existe um pós-lançamento, o qual permite obter uma perspectiva tanto comercial como técnica do impacto gerado pelo produto no mercado, a fim de identificar possíveis melhorias no futuro.

A metodologia de desenvolvimento de produto genérico desenvolvido por Ulrich e Eppinger [12] é possível retornar as fases anteriores para refazer as atividades. Além do que, é possível desenvolver as fases posteriores, sem necessariamente finalizar a atual. Sendo assim, conforme descrito e considerando a metodologia proposta, o desenvolvimento da garra robótica antropomórfica será abordado considerando essa metodologia com o intuito de estruturar o processo do projeto e desenvolvimento da garra robótica.

Capítulo 3

Trabalhos Correlatos

Neste capítulo, são apresentados os principais trabalhos correlatos encontrados na literatura para o projeto e desenvolvimento da garra robótica antropomórfica, demonstrando suas características tais como: atuadores, graus de liberdade, mecanismos empregados, etc. Devido existirem uma grande variedade de garras robóticas antropomórfica, no trabalho é focado apenas nas garras robóticas sub-atuadas, controle de força e sensores táteis aplicados em garras robóticas, conforme relacionado pelas necessidades dos clientes no Capítulo 3.

3.1 Garras Robóticas Antropomórfica

No trabalho desenvolvido por Pertuz [99], é proposto uma garra robótica sub-atuada, visualizado na Figura 3.1, com sistema embarcado baseado em arquiteturas reconfiguráveis de controle dinâmico sintonizado com algoritmo bioinspirado aplicado ao problema de resolver agarres. A garra desenvolvida possui 7 atuadores e uma cadeia cinemática de 16 graus de liberdade. Os mecanismos de transmissão de movimento para os dedos são feitos com barras dispostas em 4 elos. No entanto, esse mecanismo não permite a adaptação das falanges para se adaptarem ao formato do objeto. Através dos estudos desenvolvidos pelo autor, conseguiu melhorar o projeto da garra robótica antropomórfica aproximando-se das configurações da mão humana. É comprovado a oponibilidade do polegar através do teste de Kapandji e possui controle de impedância para possibilitar o controle da força de agarre. A arquitetura de controle por sua vez, é controlada através de Arduino e Arranjo de Portas Programáveis em Campo (*Field Programmable Gate Array - FPGA*).

Conforme mencionado pelo autor, o processo de otimização atingiu um erro quadrático médio 0.00266 entre o dedo projetado e o dedo da mão humana, onde as características da garra robótica, aproximou-se da mão humana. Sendo a garra robótica desenvolvida pelo autor de 210x84x38mm com 413.13gr vs 175 x 89x43 mm com 409.5gr da mão humana. Como proposta de inovação,

o autor utiliza algoritmo bioinspirado para sintonização do controle de impedância, onde obtém resposta sub-amortecido com tempo de subida do controlador de 355ms e erro em estado estável de 1.2%. O mesmo não utiliza uma proposta de desenvolvimento de produto para confecção da garra robótica. Utilizando como mecanismo de movimento, micro-motores dc 12V. O protótipo é confeccionado em impressão 3D com desenvolvimento da placa de controle dos motores que ficam integrados na mão, assim como, o conjunto de transmissão de movimento.



Figura 3.1: Garra robótica antropomórfica desenvolvida por [99]

No trabalho de Yulin et al. [66], propõem uma garra robótica, ilustrado na Figura 3.2, com controle de força através de controle proporcional, integral e derivativo. Como sensor de força, utiliza *Force Measure Resistive* na ponta dos dedos. O mecanismo é composto por 6 atuadores, sendo 5 motores, 1 servo motor e *encoders* possuindo uma cadeia cinemática de 16 graus de liberdade. A arquitetura de controle é baseado no chip MSP430F5529 embutido na palma da mão e integrado com o computador através de comunicação serial, sendo apresentado a cinemática e dinâmica da garra robótica.

Com a utilização do controle por PID, a fricção, a elasticidade dos elementos e outras incertezas influenciam no controle da garra robótica. A fim de possibilitar o controle efetivo da garra, implementam controle de posição nos dedos. Diferente dos outros estudos, só demonstram a manipulação de apenas duas formas canônica de agarre, que é o movimento tipo pinça e força. Não demonstram uma metodologia estrutura para desenvolvimento da garra.

Nos estudos desenvolvidos por Mahmoud e Aníbal [61] é desenvolvido uma garra robótica antropomórfica com características sub-atuadas, possuindo 3 atuadores com 10 graus de liberdade na cadeia cinemática dos dedos, conforme Figura 3.3. A mesma possui a capacidade de realizar até 10 movimentos realizados pela mão humana nas tarefas do dia a dia, conforme os estudos desenvolvidos por [60]. Dessas 10, o controlador possui a capacidade de manipular 7 movimentos

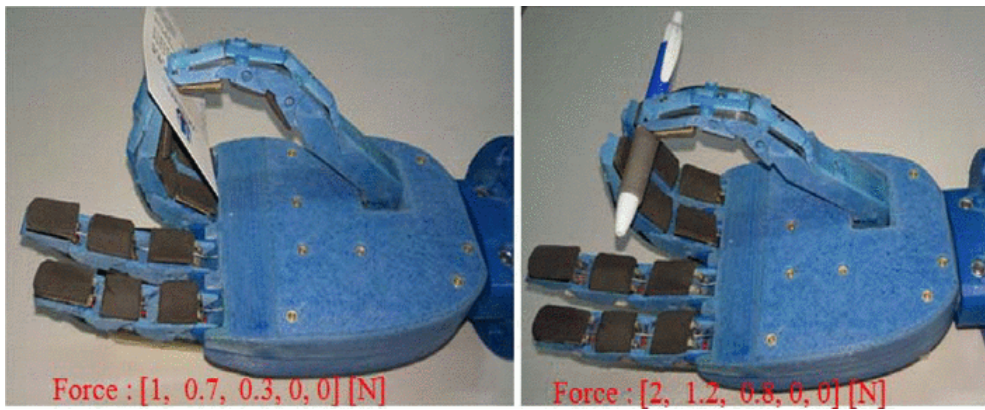


Figura 3.2: Garra robótica antropomórfica *SHU II* desenvolvida por [66]

diferentes, sem mudar a entrada de controle do sistema. O controle de força é realizado através do controle proporcional-derivativo existente no controlador dos motores Dynamixel AX-12, porém, esses motores possuem peso e tamanho considerável, atrapalhando a dimensão da garra. É desenvolvida com base em impressão 3D, tendo um custo aproximado entre 400 à 800 euros. Os atuadores são ajustados para o torque preciso dos dedos, sendo sob-dimensionados para o polegar e subdimensionado para os outros dedos. Em alguns casos para as formas de agarres considerados, a garra robótica não consegue imitar o movimento das mãos, devido o dedo não possibilitar a atuação do movimento de abdução.

Conforme mencionado pelos autores, apesar de desenvolverem uma garra robótica antropomórfica sub-atuada, não utilizaram uma metodologia estruturada de desenvolvimento de produto. Os estudos conduziram em uma garra com atuadores sob e subdimensionados para as características de agarre propostas. O controle de força é desenvolvido através do controlador interno dos motores Dynamixels Ax-12, sem a utilização de sensores táteis para sentir o objeto a ser manipulado.

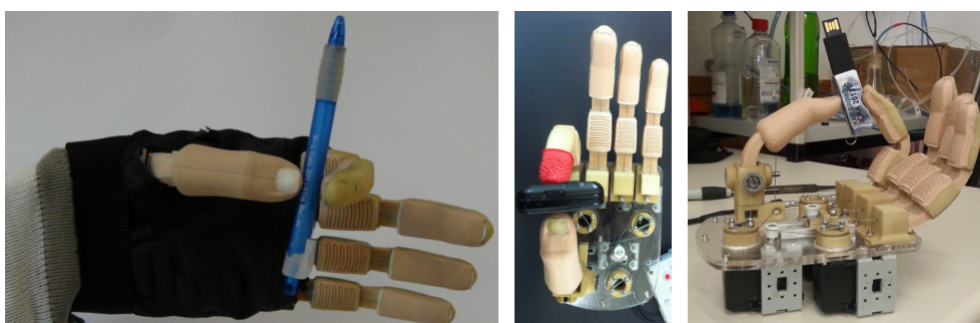


Figura 3.3: Garra robótica antropomórfica *ISR-SoftHand* desenvolvida por [61]

Ting et al. [100] apresentam uma garra robótica antropomórfica com 5 dedos e com multi-sensores possuindo uma cadeia cinemática de 11 graus de liberdade, visualizado na Figura 3.4. Ela possui uma arquitetura de controle integrada utilizando FPGA para controle local de processamento e comunicação e um sistema embarcado de processador digital de sinal para controle

dos motores na palma da mão. A garra possui em seus dedos 9 sensores táteis, 6 de posição, 5 de força e 6 sensores de corrente. A arquitetura de controle é composto por duas partes principais: sendo uma de baixo nível e outra de alto nível. A de baixo nível controla a estabilidade do agarre e a de alto nível para interpretações subjetivas da interface do usuário com a prótese.

A garra robótica é controlada através de sinais de eletrodo EMG (Eletromiográfico). No trabalho é demonstrado o agarre satisfatório de agarres que são realizados no dia a dia, sendo que, a garra possui a capacidade de 18 movimentos diferentes através de 6 eletrodos distribuídos para controle. A capacidade de manipulação da garra é determinada pelos sensores de posição e torque integrado nos dedos que possui um sistema de controle através da impedância dos motores.

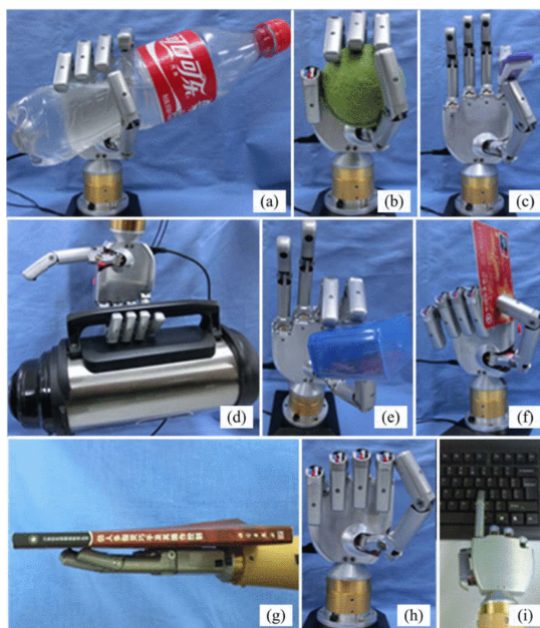


Figura 3.4: Garra robótica antropomórfica *Hit Hand* desenvolvida por [100]

Liu et al. [36] desenvolveram uma garra que possui 15 atuadores e uma cadeia cinemática de 20 graus de liberdade, conforme Figura 3.5. Essa garra sub-atuada é uma das mais desenvolvidas já projetada, onde tem integrado em cada dedo 3 sensores de posição, 3 sensores de torque, 3 *encoders* absolutos de efeito hall, 1 sensor de força/torque e 2 sensores de temperatura. Possui controle de posição e impedância para controlar a força de agarre. Ela é produzida com um sistema eletrônico embarcado baseado em FPGA e seus atuadores são baseados em mini motores DC BLDC e *driver* harmônicos.

Um das desvantagens dessa garra robótica é que torna-se inviável para reprodução em laboratório para fins de pesquisa e educação, pois, possui muitos sistemas embarcados. Cada dedo é composto por um sistema embarcado de processamento digital de sinais e uma arquitetura baseada em FPGA. Existe o FPGA na base do dedo para realizar a calibração e monitoramento dos sensores (além dos controles dos motores) e uma arquitetura em FPGA na palma da mão que controla todos os outros FPGA da Altera com chip NIOS.



Figura 3.5: Garra robótica antropomórfica *DLR II* desenvolvida por [36]

Em contrapartida, a mão desenvolvida por Jun-Uk [67] é uma garra robótica antropomórfica com apenas 2 atuadores, possuindo uma cadeia cinemática de 16 graus de liberdade, visualizado na Figura 3.6. Tem como mecanismo de transmissão de movimentos engrenagens planetárias que movimenta os quatro dedos principais e uma engrenagem helicoidal que movimenta o dedo. Possui estrutura toda em metal, podendo realizar o agarre de quatro formas canônicas de movimento: cilíndrico, ponta, gancho e agarre lateral.

O trabalho desenvolvido pelo autor, diferente do trabalho abordado [61], possui uma capacidade de manipulação inferior. O controle da garra robótica é feito através de um processador digital de sinais. O controle de força, por sua vez, é feito por dois sensores planos conectados aos eixos. Conforme testes realizados pelos autores, a força de agarre chega a 15N. Na Tabela 3.1 é resumido as principais características descritas dos autores aqui apresentados.

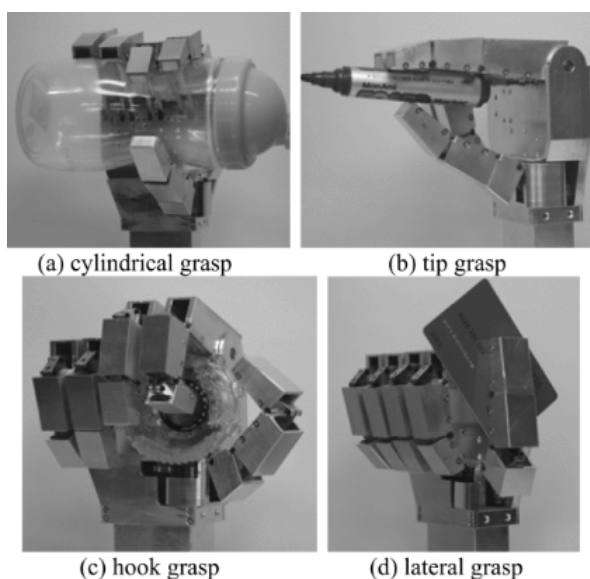


Figura 3.6: Garra robótica antropomórfica *KNU Hand* desenvolvida por [67]

Tabela 3.1: Comparação entre garras robóticas antropomórficas desenvolvidas

Garra Robótica	Ano	Número de Atuadores	GDL	Transmissão dos Dedos	Atuadores	Sensores	Tipo de Controle	Plataforma	Peso
Mão Sérgio [99]	2017	7	16	Barras e Parafusos sem fim	Micro Motores DC	Sensores de Torque	Controle de Impedância	Arduino Mega e VHDL-FPGA	413.13g
SHU-II [66]	2016	6	16	Transmissão por fios	Motores DC e Servo Motor	FSR	PID	PC; Microcontrolador	600g
ISR-Soft hand [61]	2014	3	10	Transmissão por fios	Motores DC Dynamixel AX-12	Ver Manual AX-12	Controle PD de Posição	PC	-
Hit Hand [100]	2014	5	11	Transmissão por fios	Motores DC	Sensor de Posição; Sensor de Força/-Torque; Sensor de Corrente; Sensor de Tato	Controle de Impedância	FPGA	440g
DLR Prosthetic hand II [36]	2009	15	20	Transmissão por fios Metálicos	Motores Brushless DC	15 SP; 15 ST; 5 SF; 15 Enconders absolutos de efeito hall; 10 ST; Sensores Táteis	CI; Controle PD de Posição	6 DSP e 6 FPGA	-
KNU Hand [67]	2009	2	16	Transmissão por Cabos	Motores de DC	2 Sensores de Posição Resistivo	Controle de Força	DSP	800g

3.2 Controle de Força em Garras Robóticas

Considerandos os trabalhos que atuam exclusivamente com o controle de força em garras robóticas, os estudos desenvolvidos Jalani et al. [75], mostram a utilização de um modelo de controle de conformidade baseado em impedância, com base em um sistema de PID a fim de controlar a garra robótica RED Hand. Através de duas simulações, demonstram os efeitos do nível de conformidade, através da variação de dois parâmetros K_i (Coeficiente de Rigidez) e K_s (Coeficiente de Amortecimento) para o dedo polegar e os outros dedos da garra robótica. Uma vez fixando o valor de K_i e variando o valor de K_s e vice-versa, observou-se que os níveis de controle de conformidade basicamente se mantiveram para o dedão e o restante dos dedos da garra robótica. Através da variação dos valores de K_i e K_s , observou-se que a garra robótica RED Hand, obteve melhores níveis de controle de conformidade entre 2 a 6,5 segundos. Os autores apenas realizaram a simulação do controle de conformidade para a garra robótica, sendo a mesma, ainda em construção.

Em um segundo estudo desenvolvido Jamil et al. [101], de relevância para o desenvolvimento desse trabalho, mostram a implementação de um controle de conformidade baseado em controle de conformidade ativo com controle através de lógica *fuzzy* a fim de reduzir os impactos e as forças durante a interação física com objetos e os robôs, utilizando um modelo de impedância e sensores de força e posição para *feedback*. O universo de discurso considerado para geração das funções de pertinência variam de -0.025 a 0.025 rad para erro, -1 a 1 rad/s para variação do erro e -1 para 1 para voltagem. As regras para o controle *fuzzy* foram baseados na experiência e conhecimento dos autores sobre o problema, consistindo em 25 regras para posição/agarre da RED HAND. Através da variação dos dois parâmetros K_i e K_s e utilizando o controle *fuzzy*, observou-se que garra robótica obteve melhores resultados, permanecendo com a estabilidade no agarre sem danificar o objeto por mais tempo. O controle de conformidade manteve-se estável entre 2 a 10,5 segundos, ante os 2 a 6,5 segundos.

Nos estudos desenvolvidos por Faraj et. al [102], é desenvolvido um controle PID em cascata utilizando uma garra robótica pneumática sub-atuada de 3 dedos, conforme ilustrado na Figura 3.7. A saída do controlador vai para um regulador pneumático PCI-6024E enquanto a entrada do *feedback* dos sensores analógicos de pressão são obtidos através de um bloco de entrada PCI-6024E. Há validação da garra robótica da-se com três objetos com diâmetros de 3.5cm, 5cm e 7cm. Os resultados mostraram que o controlador é capaz de mover os dedos simultaneamente e agarrar objetos com diferentes formas perfeitamente. Quanto aos erros do controlador, possui um erro de posição entre o *setpoint* e a referência medida de 0.05° à 0.11° .

Sono [103][104] apresenta o desenvolvimento de uma garra robótica com 3 dedos e 3 graus de liberdade, visualizado na Figura 3.8. O sistema de controle de preensão é realizado em duas partes. A primeira parte relacionada ao movimento de abertura e fechamento do dedo através de sinal eletromiográfico e a segunda em relação ao controle de força já agarrado o objeto. O

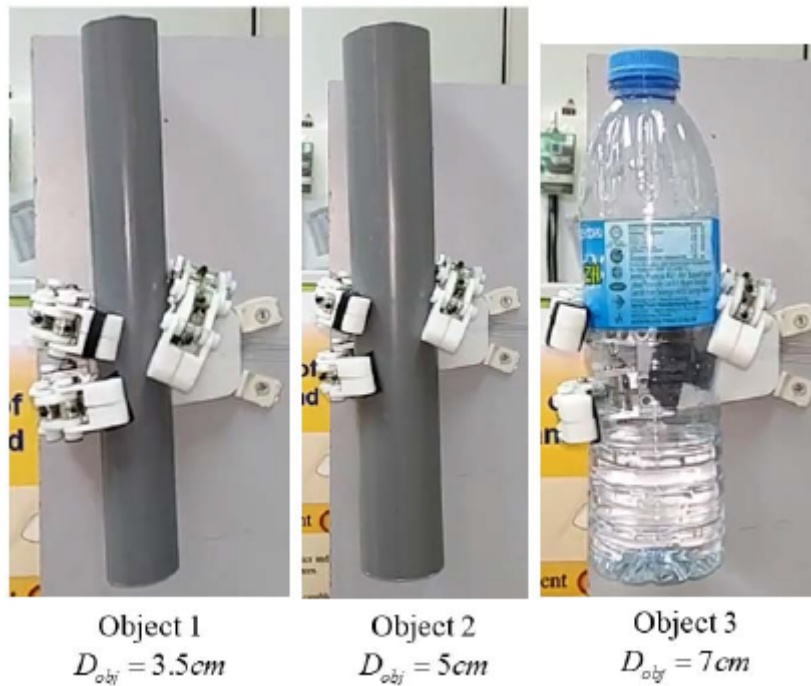


Figura 3.7: Garra robótica antropomórfica desenvolvida por [102], com diferentes agarre de diferentes diâmetros

controle de força por sua vez é realizado de forma indireta através do controle de impedância do motor. Através da simulação cinemática, é verificadas as restrições para cada grau de liberdade do mecanismo, demonstrando o comportamento do dedo desenvolvido. O controlador proposto é baseado em PID, sendo os dados do controlador obtidos de forma empírica, comparando as respostas obtidas do modelo real com o simulado, obtendo $K_p = 11,5$, $K_d = 0,06$ e $K_i = 0,6$. Devido K_d e K_i serem pequenos, o autor despreza para realização dos testes no modelo, bem como as não-linearidades do sistema. Para o sistema de controle de posição o modelo apresenta grande erro em regime permanente, devido a corrente do motor não apresentar uma relação linear com a sua posição angular. Já para o sistema de controle de força, apresentou bons resultados. Entretanto, para adquirir corretamente o sinal de corrente do motor, precisou ser utilizado um sistema de aquisição de alto desempenho para possibilitar as medições de frequência do sinal em 10kHz. O trabalho não implementa o sistema de controle em um *hardware* microcontrolado, sendo utilizado através de um PC e Labview.

Scheurenbrand [105], propõe uma garra robótica com 3 dedos com 3 GDL em cada dedo, apresentado na Figura 3.9. Desenvolve o *hardware* e *software* de controle de força e posição para a garra. A força é medida através de um sensor resistivo de força FSR. O sistema de controle é proposto através do software Matlab/Simulink e hardware dSPACE. O hardware disponibiliza 16 canais analógico/digital, 4 saída PWM, 2 entradas para medição de *encoder* e 20 entradas e saídas digitais. Modelos baseados no Simulink podem ser descarregados para a placa a fim de implementar o controle do sistema. A garra utiliza motores DC com caixa de redução, sendo a

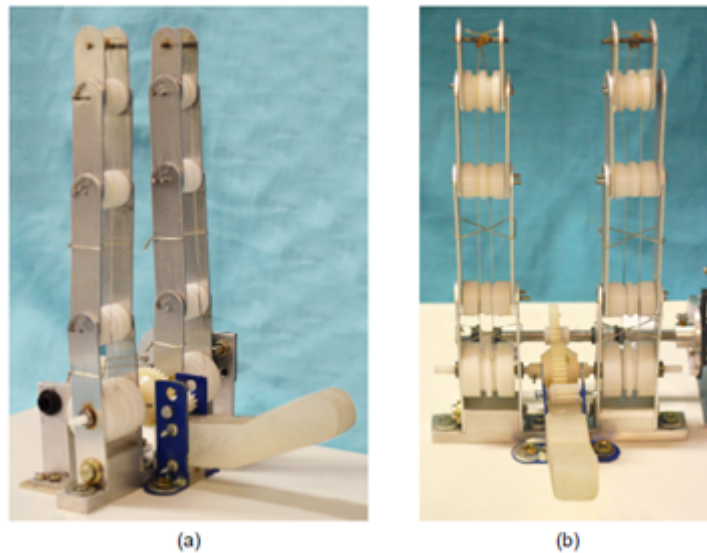


Figura 3.8: Garra robótica antropomórfica desenvolvida por [104]. Figura (a) vista frontal e Figura (b) vista lateral

força máxima dos dedos de 20N. O autor propõe um sistema de controle baseado em PI, sendo que não realiza a modelagem do sistema. Obtém um sistema com melhor estabilidade com o controle Proporcional = 3.0 e Integral = 5.0 apenas para um tipo de agarre. Com a implementação do sistema e realização de testes, o autor chega a conclusão que a área de contato entre os objetos e o sensor é de apenas 21%, influenciando no comportamento do sistema de controle, dependendo da geometria do objeto a ser agarrado.

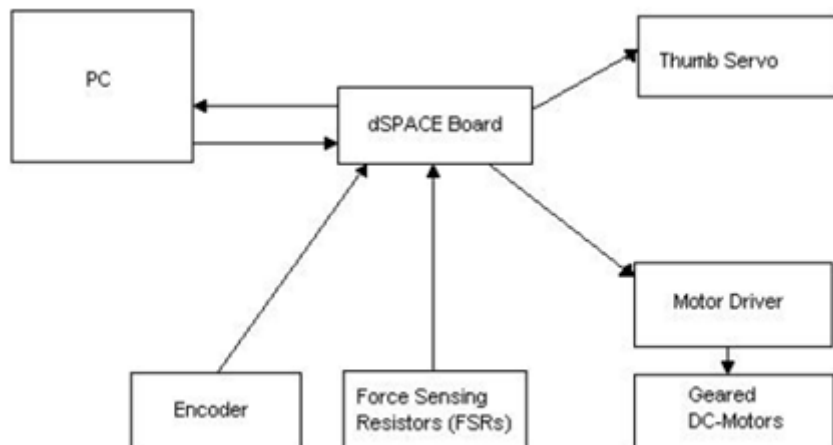


Figura 3.9: sistema de controle de força proposto por [105].

3.3 Sensores Táteis em Garras Robóticas Antropomórfica

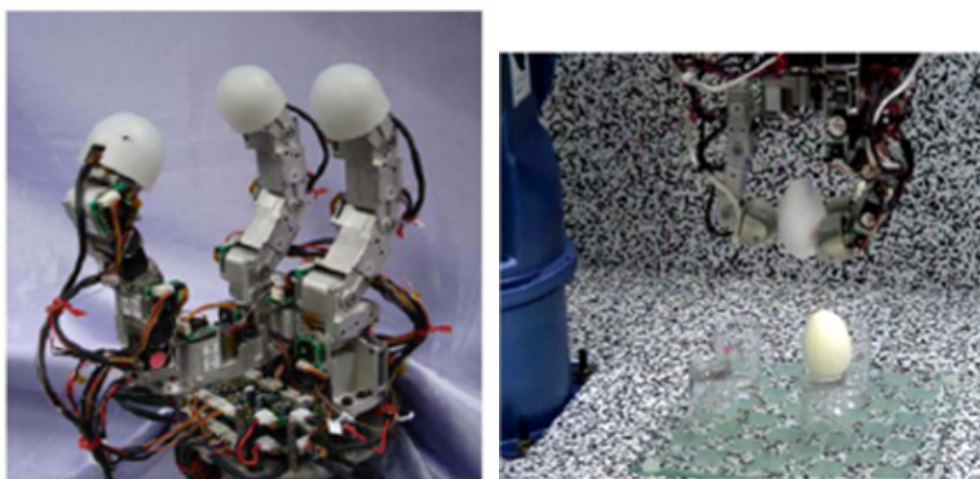
Pfeifer [106] desenvolveu a garra robótica denominada de mão de Yokoi. A mão constitui 13 GDL, desenvolvida com material elástico, flexíveis e deformáveis. Os GDL são acionados por servomotores. A mão é equipada com um conjunto de sensores FSR, cobrindo diferentes locais de contato da mão, conforme ilustrado na Figura 3.10. Incluindo áreas de contato nas pontas dos dedos, na parte de trás de cada dedo, além da palma da mão. A garra robótica é utilizada no laboratório de inteligência artificial da Universidade de Tóquio, sendo utilizada para estudos de dinâmica, morfologia e aprendizagem. Devido a morfologia da mão, dos tendões elásticos e das pontas dos dedos serem deformáveis, a mão se adapta automaticamente ao objeto que será agarrado. Conforme o autor, não há necessidade de saber de antemão qual será a forma do objeto a ser agarrado. A medida que a mão interage com o objeto, ela assume sua forma.



Figura 3.10: Disposição dos FSR sobre a mão robótica de Yokoi [106].

Takahashi et al. [107], propuseram um método robusto de controle de força e posição para objetos de propriedade desconhecida durante o agarre. A técnica de controle proposta é capaz de selecionar o controle de força ou posição, comutando suavemente de acordo com a quantidade de força que é aplicada sobre o objeto. O método foi aplicado ao agarre adaptativo através de uma mão robótica que possui 12 GDL. A garra é apresentada na Figura 3.11, pegando um ovo cozido. Conforme testes realizados, a mão consegue realizar o agarre estável mesmo que a posição precisa da superfície, a massa e a rigidez sejam desconhecidas.

Ulmen e Cutkosky [108] desenvolveram uma matriz de sensores capacitivos que podem ser utilizados para captar informações táteis para ser utilizados em garras robóticas antropomórfica. O projeto apresenta baixo ruído e é mecanicamente robusto, porém confeccionado com material macio para permitir absorção de energia em caso de impactos e não romper o mecanismo. Os sensores capacitivos foram utilizados pela Universidade de Stanford em uma garra robótica de



(a)

(b)

Figura 3.11: Garra robótica desenvolvida por [107]. Na figura a) a estrutura da garra robótica e os sensores táteis em branco. Na figura b) a manipulação de um ovo cozido sem deformação.

três dedos com 3 GDL cada, a fim agarrar objetos sem conhecê-los previamente, usando apenas o feedback tátil dos sensores. Cada dedo é constituído de um único motor para possibilitar abertura e fechamento e transmissão dos movimentos através de um mecanismo por barras e molas. A garra robótica foi equipada com a matriz de sensores, conforme Figura 3.12, para possibilitar a percepção tátil. Ela utiliza Sistemas Microeletromecânicos (Micro-Electro-Mechanical System - MEMS) de 3 eixos para fornecer dados dinâmicos na ponta dos dedos com uma taxa de informação de 800Hz. Os sensores táteis podem medir uma força de 20mN (aproximadamente 2g) e atingir uma saturação sob forças na faixa de 100N. Os sensores táteis nas mãos tem uma taxa de amostragem de 100 Hz.

Dilfeter et al. [109], através do projeto NASA/DARPA Robonaut efetuaram o controle de um robô antropomórfico controlado através de tele operação. O Robonaut, visto na Figura 3.13, foi projetado para permitir a destreza de um astronauta no espaço. Através de uma luva tátil personalizada, permite a identificação e apreensão reflexiva de objetos no espaço em tempo real. A luva é constituída de 19 sensores de força, sendo que cada articulação do dedo é equipada com um sensor, e o polegar tem vários sensores para distinguir entre os diferentes tipos de agarre. Três sensores estão localizados estrategicamente ao longo da palma da mão para determinar o contato com ferramentas. A luva é projetada para ser robusta e proteger os sensores. Um sistema de controle localizado no cérebro do Robonaut, comanda os dedos para fecharem automaticamente quando a estimulação tátil é detectada nos sensores da palma da luva. No Robonaut, o tele operador move o braço do robô até que a palma da mão entre em contato com o objeto. Neste ponto, a garra reflexiva é ativada automaticamente. O tele operador avalia a compreensão resultante e decide se um novo agarre é necessário.

Kyberd [110] propôs um sistema de percepção tátil que utiliza sensores FSR montados na

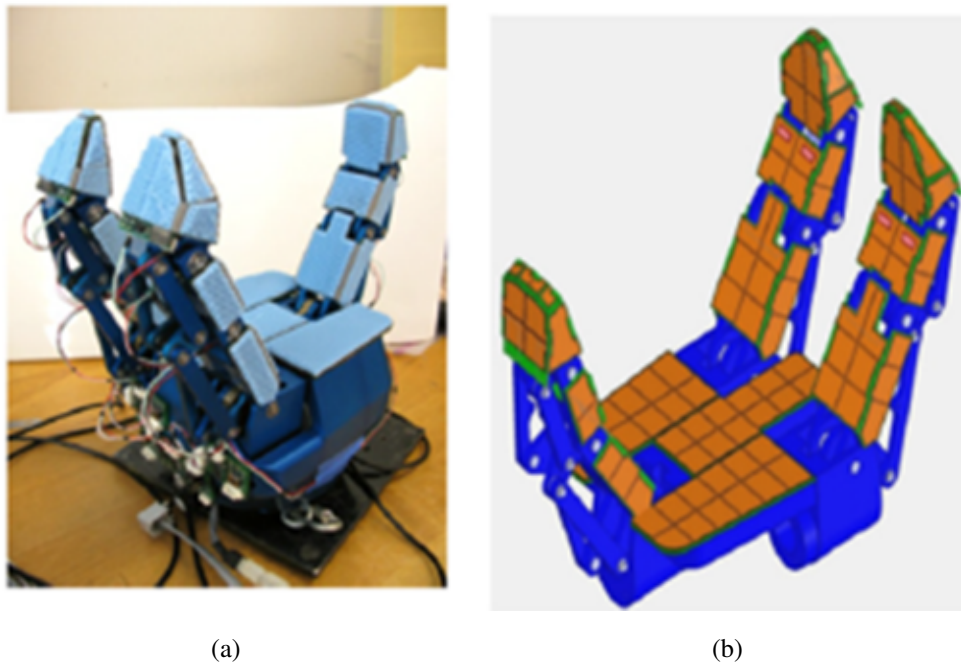


Figura 3.12: Garra robótica equipada com os sensores capacitivos [108]. Figura a) Garra robótica montada com os sensores capacitivo em azul claro e Figura b) Vista CAD do acoplamento dos sensores capacitivos em laranja

superfície da palma da mão de uma mão robótica antropomórfica, conforme Figura 3.14. É combinado aos sensores resistivos, sensores ópticos/acústicos no polegar para possibilitar as informações de deslizamento no polegar. Um microcontrolador Intel 8096 é utilizado para coletar os dados, calcular o vetor de força e coordenar as ações da mão. As tensões analógicas lidas pelos sensores são armazenadas e convertidos em uma matriz de bits. Os sensores são utilizados para derivar um vetor de força na direção do deslizamento do objeto com base nas alterações dos valores relativos a sua saída.

No trabalho realizado por Romano [77], utilizando uma garra robótica de dois dedos apresentado na Figura 3.15, propõe sensoriamento tátil robusto a fim de ser aplicado ao maior número de objetos. Cada uma das pontas dos dedos da garra robótica é equipada com um conjunto de sensores de preensão que consiste em 22 células individuais. As 22 células são divididas em uma matriz de 5x3, dois elementos sensores no final da ponta do dedo, dois elementos sensores em cada lado do dedo e um na parte de trás. Esses sensores são capacitivos e medem a força de compressão perpendicular que é aplicada em cada região detectada e têm uma resolução nominal de 6,25mN. Toda a superfície dos sensores é coberta por uma camada protetora de borracha de silicone que fornece a aderência e a fricção necessárias para o sucesso do agarre. Na Tabela 3.2 é resumido as principais características apresentadas pelos autores referente a implementação de sensores táteis em garras robóticas.

Dentre os trabalhos realizados, cabe salientar que nenhum adotou uma metodologia específica

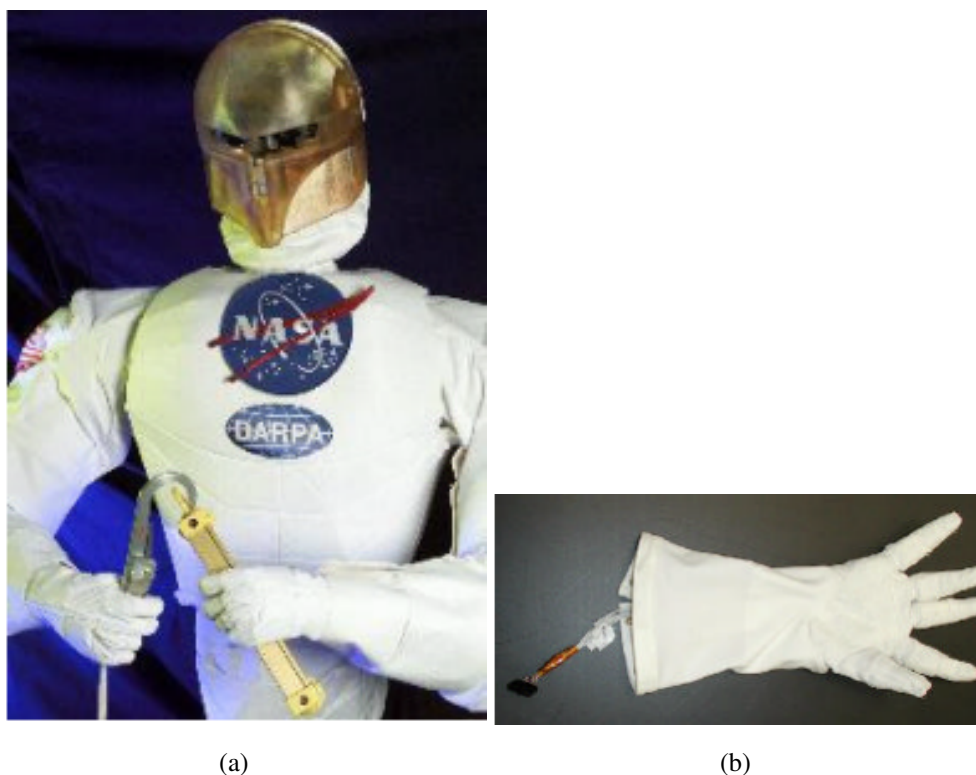


Figura 3.13: Robonaut desenvolvido pelo projeto NASA/DARP. Figura a) Robonaut desenvolvido para operações espaciais equipada com a garra robótica. Figura b) Garra robótica com percepção tátil para operações espaciais

Tabela 3.2: Comparativo dos principais tipos de sensores aplicados em garras robóticas

Sensor Tátil	Mão Robótica	N° de Ponto de Contatos	Range	Taxa de Amostragem
FSR [76]	Luva Robonaut	19	20N	1kHz
FSR [110]	Mão Southampton	15	-	-
Sensor Capacitivo [77]	PR2 Garra Robótica	22	7kPA	24.4Hz
Sensor Capacitivo [108]	Garra Robotiq	132	12N	300Hz
Matriz de Sensores [107]	Takashi hand	86 por dedo	2N	38.9Hz
FSR [106]	Yokoi hand	-	-	-

de desenvolvimento de produto. Sendo assim, pode-se observar que muitas das garras desenvolvidas possuem um sistema de controle muito complexo com multi-sensores e estrutura em FPGA. A reprodução das garras desenvolvidas, tornam-se inviáveis a fim de pesquisa e estudos robóticos em universidade e institutos federais em desenvolvimento, devido aos altos custos de implementação. Com exceção das garras desenvolvidas por [99] e [66], porém, a primeira é dotada de mecanismos de transmissão de movimentos em barras, não tendo as características de adaptação aos objetos. A segunda é dotada de um sistema de controle que não tem tanta efetividade para

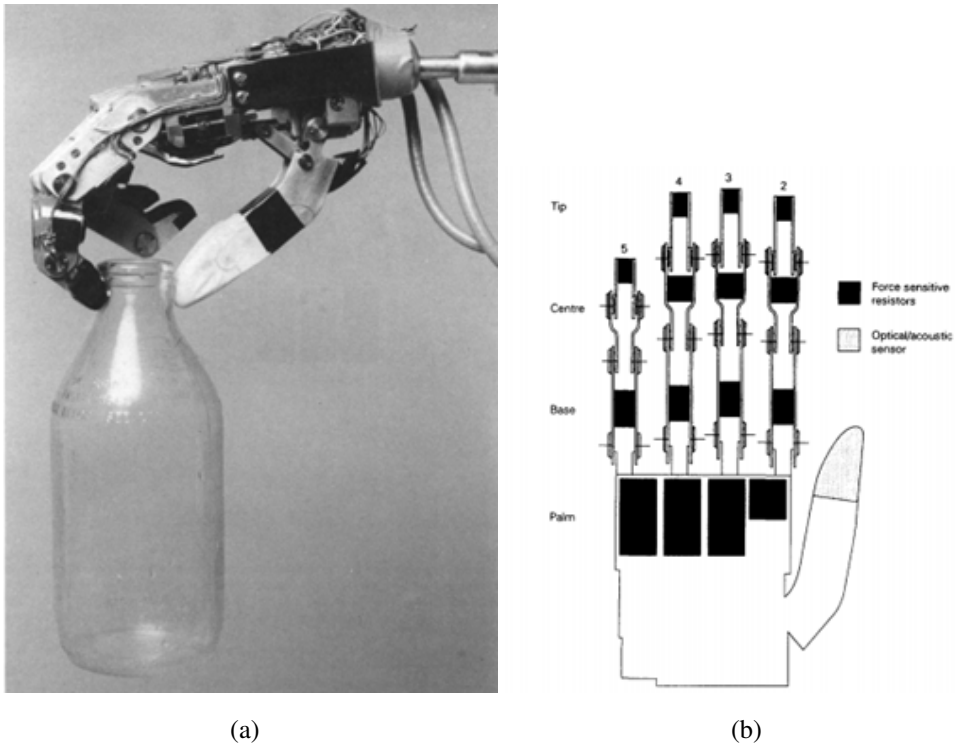


Figura 3.14: Garra robótica antropomórfica desenvolvida por [110]. Figura a) Mostra a manipulação de uma garra de vidro e na Figura b) A disposição dos sensores sobre a garra robótica

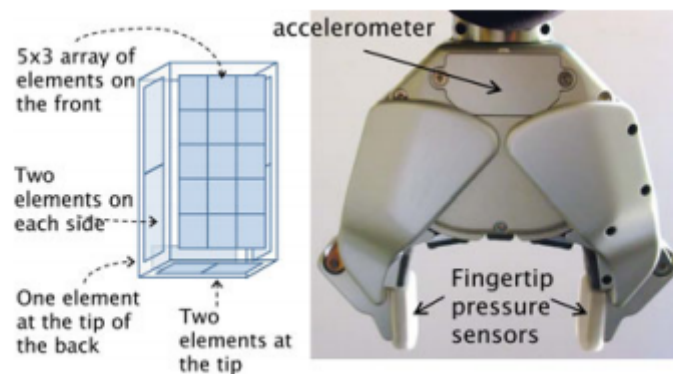


Figura 3.15: Matriz de sensores táteis desenvolvido por [77]

manipulação dos objetos, devido à alta característica de não-linearidade das garras robóticas.

Considerando os sistemas de controle desenvolvidos, a maioria das aplicações utilizam uma das configurações do controlador PID, adotando estratégia de controle híbrido força/posição. Esse tipo de controle possibilita o controle de interação com o objeto sem o dedo estar em contato e estando em contato com o objeto. Outrossim, as estratégias de controle adotadas para os trabalhos desenvolvidos de [63], [101], [75] foram apenas simulações, sem a implementação física. Enquanto os outros trabalhos fizeram a implementação, porém, utilizando estruturas em FPGA ou Computador com alto desempenho de processamento. Quanto a implementação de sensores táteis

nas mãos, os FSR destacam-se por serem de baixo custo, ao passo que são efetivos para aplicação proposta. Entretanto, deve-se observar a área de contato entre o objeto e sensor, conforme retrato pelos trabalhos de

Capítulo 4

Desenvolvimento da Garra Robótica Antropomórfica

Este capítulo apresenta o processo de desenvolvimento da garra robótica antropomórfica. O processo é abordado na visão de desenvolvimento de produto obedecendo à estrutura do modelo de referência genérico descrito na Seção 2.8. São descritos quatro fases do modelo aplicado ao desenvolvimento da garra robótica antropomórfica: a fase de planejamento, desenvolvimento do conceito, projeto ao nível de sistema e projeto detalhado. A fase de testes e refinamento e lançamento à produção, não abordado nesse trabalho. Ao longo de cada uma das fases, são empregadas ferramentas de projeto para auxiliar a tomada de decisões durante a definição de especificações técnicas, a geração de princípios de solução para o projeto e a consolidação da arquitetura do produto.

4.1 Fase 0 - Planejamento

Conforme modelo de referência genérico desenvolvido por Ulrich e Eppinger [12], essa fase de maneira geral, coleta possibilidade para novas plataformas de produtos. Sendo os projetos em potenciais, avaliados com base na estratégia competitiva da empresa, documentados e avaliados os mercados-alvo, as partes interessadas do produto e os problemas críticos. Sendo o trabalho aqui desenvolvido, motivado conforme as informações descritas no Capítulo 1.

4.2 Fase 1 - Desenvolvimento do Conceito

Essa fase tem por objetivo a identificação das necessidades dos clientes e à transformação em requisitos de produtos a fim de obter as especificações metas do produto. A definição dos clientes é fundamental para levantar as necessidades, que devem ser definidas antes de tomar decisões

importantes. Os clientes não são somente os usuários finais do produto, mas todas aquelas pessoas envolvidas na fase do ciclo de vida do produto, aqui representados por: professores, alunos, projetistas, técnicos, etc. O ciclo de vida do produto, por sua vez, é a descrição da vida do produto, ao longo das diversas fases pelas quais o produto passa, desde a concepção e produção, até o descarte do produto.

Conforme descrito na Subseção 2.8.2, a casa da qualidade é uma ferramenta que auxilia o desenvolvimento de produto, pois, segue uma metodologia bem estruturada. Através dela é possível transformar a voz do cliente em requisitos adequados para gerar as especificações meta do produto, garantindo as necessidades do cliente ao longo do processo de desenvolvimento do produto. Sendo assim, é empregada nesse trabalho a fim de proporcionar as especificações metas da garra robótica antropomórfica a ser desenvolvida, conforme a demanda dos clientes, considerando as limitações do projeto tanto em recursos capitais como conhecimento do projetista que é o autor desse trabalho.

4.2.1 Necessidades dos clientes

Os clientes (usuários das garras robóticas) incluem estudantes e professores tanto da Universidade de Brasília e do Instituto Federal de Rondônia onde o autor trabalha. Mediante consulta dos envolvidos, foram levantados os requisitos necessários para o desenvolvimento da garra robótica.

As necessidades dos clientes, deu-se através de fontes de informações como a experiência dos usuários contatados nas áreas mencionadas, o apoio em pesquisas bibliográficas realizadas pelo autor e revisão de projetos já existentes. Em tal levantamento levou-se em consideração a ideia de um produto com a possibilidade de manipulação do maior número possível de agarres, percepção tátil, interface gráfica e um produto com alta versatilidade e baixo custo. Desta forma, foram consolidados 15 requisitos de clientes devidamente agrupados em cinco categorias principais (funcionalidade, produção, usabilidade, consumo e reciclagem) visando facilitar sua análise e avaliação. Os requisitos dos clientes e suas respectivas categorias são listados na Tabela 4.1.

4.2.2 Priorização dos requisitos dos clientes

A fim de desdobrar as necessidades dos clientes em necessidades do produto, com o intuito de descobrir, em linguagem de engenharia o que o cliente quer, antes é necessário realizar a priorização dos requisitos dos clientes para verificar quais requisitos possuem maior importância dentro do projeto do produto, sendo necessário atribuir uma importância e peso relativo a cada requisito. Sendo assim, para determinar a importância dos requisitos dos clientes, outra ferramenta de auxílio ao desenvolvimento de produto é utilizada: o diagrama de Mudge. O diagrama de Mudge, segundo Rocco e Silveira [111], permite realizar a hierarquização dos requisitos dos clientes em pares, exceto os iguais, obtendo-se a importância relativa de cada requisito. Cada requisito as-

Tabela 4.1: Necessidades para Garra Robótica Antropomórfica

Etapas do Ciclo de Vida	Lista de Necessidades
Funcionalidade	Manipular objetos de diferentes formas e tamanhos Possuir sistema de <i>feedback</i> de força e com medição direta Manipular objetos de pesos diferentes Controle manual e automático dos dedos Ser possível interagir com o usuário Ser compatível com manipuladores robóticos Ter estabilidade de movimentação dos dedos
Produção	Baixo custo de montagem Produtos para montagem serem de fácil acesso Ter aparência da mão humana Modular e fácil de montar
Usabilidade	Executar movimentos de força Executar movimentos de precisão
Consumo	Baixo consumo de energia
Reciclagem	Usar materiais de fácil reutilização

sume um nível de importância: muito mais importante (valor 5 - letra A), mais importante (valor 3 - letra B) e pouco mais importante (valor 1 - letra C). Sendo o valor final de cada requisito, o somatório dos valores observados em todo diagrama, conforme Figura 4.1.

Diagrama de Mudge																	
Número dos Requisitos do Cliente																	
-	2	3	4	5	6	7	8	9	10	11	12	13	14	15	SOMA	%	
1	2A	3A	1C	5A	6A	1B	8A	9C	10A	11B	12A	12A	14B	15C	9	3%	
	2	2A	2B	2B	6A	2B	2B	2C	10A	2C	12B	13B	2B	2B	30	10%	
		3	4B	5B	6A	7B	8A	3B	10A	11B	12A	13A	3C	15C	4	1%	
			4	5B	6A	4C	4A	4B	10A	4B	12C	13C	4C	15C	16	5%	
				5	6A	7B	5A	5B	5B	11B	5B	5B	5B	5C	29	9%	
					6	7B	8B	9C	10C	11C	12C	13C	6A	6A	35	11%	
						7	7A	7C	10C	7C	7C	7C	7B	7C	22	7%	
							8	9A	10C	11C	12C	13C	8B	15B	16	5%	
								9	10C	11C	12C	13C	14A	9C	8	3%	
									10	10A	10C	10C	10A	10A	41	13%	
										11	11A	11A	14B	11A	27	9%	
											12	12B	12A	12C	26	8%	
												13	13A	13C	23	7%	
													14	14A	16	5%	
														15	6	2%	
															Total	308	100%

Figura 4.1: Hierarquização dos requisitos dos clientes, conforme diagrama de Mudge

Durante a aplicação do diagrama de Mudge, a decisão sobre qual requisito do par é o mais importante, teve-se como base a necessidade de aplicação da garra robótica antropomórfica para fins de pesquisa e educação, bem como o levantamento da literatura relacionada a aplicação de garras robóticas em manipuladores robóticos, tomando como base as limitações do projeto.

Mediante o uso do diagrama e priorização das necessidades dos clientes, foi possível determinar as necessidades mais importantes para o projeto da garra, sendo eles: 1 - Ter aparência da mão humana, 2 - Ser compatível com manipuladores robóticos, 3 - Possuir sistema de *feedback* de força com medição direta, 4 - Ser possível interagir com o usuário, 5 - Modular e fácil de montar, 6 - Executar movimentos de força e 7 - Executar movimentos de precisão. Sendo os requisitos de menor importância para o projeto: 13 - Produtos para montagem serem de fácil acesso, 14 - Usar materiais de fácil utilização e 15 - Manipular objetos de pesos diferentes.

A análise da importância relativa atribuída aos requisitos dos clientes, através da comparação par a par, mostra que a obtenção de uma garra robótica com a aparência da mão humana é o principal objetivo. Uma vez que, obtendo a geometria parecida com a da mão, a possibilidade de manipulação de diferentes objetos, torna-se uma característica fundamental a ser atingida no projeto. Caso a garra robótica tenha a aparência das já tradicionais pinças robóticas, iremos nos deparar com os mesmos problemas dos quais foram descritos na introdução do trabalho.

O segundo item mais importante é ser compatível com manipuladores robóticos, sendo peça chave para possibilitar o acoplamento da garra robótica desenvolvida à robôs manipuladores em trabalhos futuros. A necessidade do desenvolvimento de uma flange, torna-se fundamental para possibilitar o acoplamento rápido e seguro aos manipuladores robóticos.

O terceiro item mais importante para o desenvolvimento da garra robótica é possuir um sistema de *feedback* de força com medição direta. Esse item torna-se de extrema importância, pois, é ele quem irá dar a percepção tátil a garra robótica antropomórfica para possibilitar manipular os objetos com destreza. Sendo essa característica, possível de se obter das mais diferentes formas através de diversas técnicas de controle, seja através de controle proporcional, integral e derivativo ou de redes neurais artificiais. Sendo que atualmente, o controle de força em garras robóticas é tratado como controle de conformidade, conforme descrito na Seção 2.5.

O quarto e não menos importante, trata-se da interface gráfica com o usuário, o qual permite ajustar para o controle manual ou automático, enviar comandos para a garra robótica, bem como visualizar graficamente a força que está sendo aplicada para manipular o objeto através de gráficos.

Para os últimos requisitos, não se trata de não serem importantes, porém, eles não são tratados como objetivos principais e necessidade fundamental de obtenção desse projeto. Muitas das vezes os materiais a serem utilizados nos projetos são de difícil acesso ou muito caros, inviabilizando o desenvolvimento do projeto. Dependendo das características e robustez do projeto a utilização de materiais reciclados nem sempre é possível. Por ser tratar de um projeto que requer a manipulação do maior número de objetos, ao mesmo tempo, ter que se assemelhar a mão humana, requer componentes eletrônicos/elétricos compactos. A força empregada pelos atuadores, por serem compactos, acabam limitando a manipulação de objetos com pesos diferentes, sendo restringido ao peso estipulado pelo projeto.

4.2.3 Requisitos de projeto

Essa etapa visa a mensuração dos requisitos dos clientes a fim de se tornarem características físicas com linguagem técnica de engenharia para desenvolvimento da garra robótica antropomórfica. Nessa etapa, os requisitos de projeto devem ser descritos a fim de possibilitar a mensuração por algum sensor. Sendo a geração e seleção dos requisitos de projeto baseados na experiência do projetista (autor) e na revisão bibliográfica. A partir das necessidades dos clientes foi expresso uma possível função do produto, sendo convertida em um total de 15 funções. Cabe salientar, que todas as necessidades dos clientes foram convertidas em necessidades de produto, não sendo necessário utilizar outro critério de priorização de requisitos de projeto, tal como o diagrama de Kano. Não há muitos requisitos definidos, sendo assim, os requisitos dos clientes foram convertidos em requisitos de produto e listados na Tabela 4.2. Os requisitos de projeto ainda são classificados quanto ao seu comportamento de necessidade do requisito, como crescente ou decrescente. Ou seja, quanto maior melhor ou quanto menor melhor.

Tabela 4.2: Estabelecimento dos Requisitos de Projeto

Requisitos de Projeto	Tendência	Unidade
Quantidade de dedos nas mãos	Crescente	Un.
Medição de Força	Crescente	-
Capacidade Máxima de Manipulação	Crescente	Kg
Controle de Conformidade	Crescente	-
Interface Gráfica	-	-
Sistema de Fixação	-	-
Precisão do Sistema de Controle	Crescente	-
Custo Total de Produção	Decrescente	R\$
Produtos Encontrados no Mercado Brasileiro	Crescente	
Graus de Liberdade	Crescente	Un.
Possuir Módulos de Montagem	Decrescente	Un.
Formas Canônicas de Movimento de Força	Crescente	Un.
Formas Canônicas de Movimento de Precisão	Crescente	Un.
Potência Consumida	Decrescente	kW
Componentes Reprogramáveis	-	Un.

4.2.4 Relacionamento entre necessidades dos clientes e requisitos do produto

O processo de identificar os relacionamentos entre as necessidades dos clientes e os requisitos de produto são realizados de maneira subjetiva, utilizando a experiência do projetista e as informações de revisão da literatura. Os níveis de relacionamento entre as necessidades dos clientes e os requisitos do produto, foram estipulados como forte, moderado ou fraco. O processo é realizado sistematicamente sobre a parte central da matriz QFD apresentada na Figura 4.2. Como resultado deste relacionamento, tem-se na parte direita da matriz as pontuações para as necessidades dos clientes de acordo com a quantidade e grau de relacionamento com os requisitos do produto.

4.2.5 Correlação dos requisitos de projeto

A matriz de correlação (teto da casa da qualidade) identifica qual a interferência de um requisito de projeto em relação a outro no atendimento de cada uma das exigências. É comparado a fim de identificar os efeitos negativos ou positivos, sobre os outros. Com essa comparação é possível não adotar ante mão solução que antes pareciam às ideais, e vice-versa, priorizando os requisitos que terão resultados satisfatório para o projeto.

Diante do exposto, o requisito de projeto que mais tem correlações negativas é o custo total de produção, pois, para se obter uma quantidade maior de dedos na mão, inevitavelmente, terá que ser empregados mais atuadores, mecanismos de transmissão, componentes elétricos, assim como, para capacidade de manipulação e controle de conformidade. Para o primeiro item quanto maior for os motores, maior será a capacidade de manipulação, entretanto, aumenta os custos com motores maiores e mais potentes, que por sua vez terá que aumentar o tamanho da mão, e assim por diante. Enquanto o controle de conformidade, dependendo da técnica que for empregada requer um custo computacional muito alto, sendo assim, a necessidade de utilizar uma técnica que exija baixa capacidade de processamento e que seja funcional é de extrema importância. As correlações negativas são as que devem ser tratadas com atenção especial durante a seleção e definição de soluções para o projeto.

Além disso, na parte inferior da matriz QFD são disponibilizadas informações referente à importância e peso relativos dos requisitos de produto, as quais são resultantes da análise de correlação do relacionamento com as necessidades dos clientes. Sendo assim, os requisitos de produto com maior importância relativa são: custo total do projeto, quantidade de dedos nas mãos, sensor de medição de força, precisão do sistema de controle e graus de liberdade.

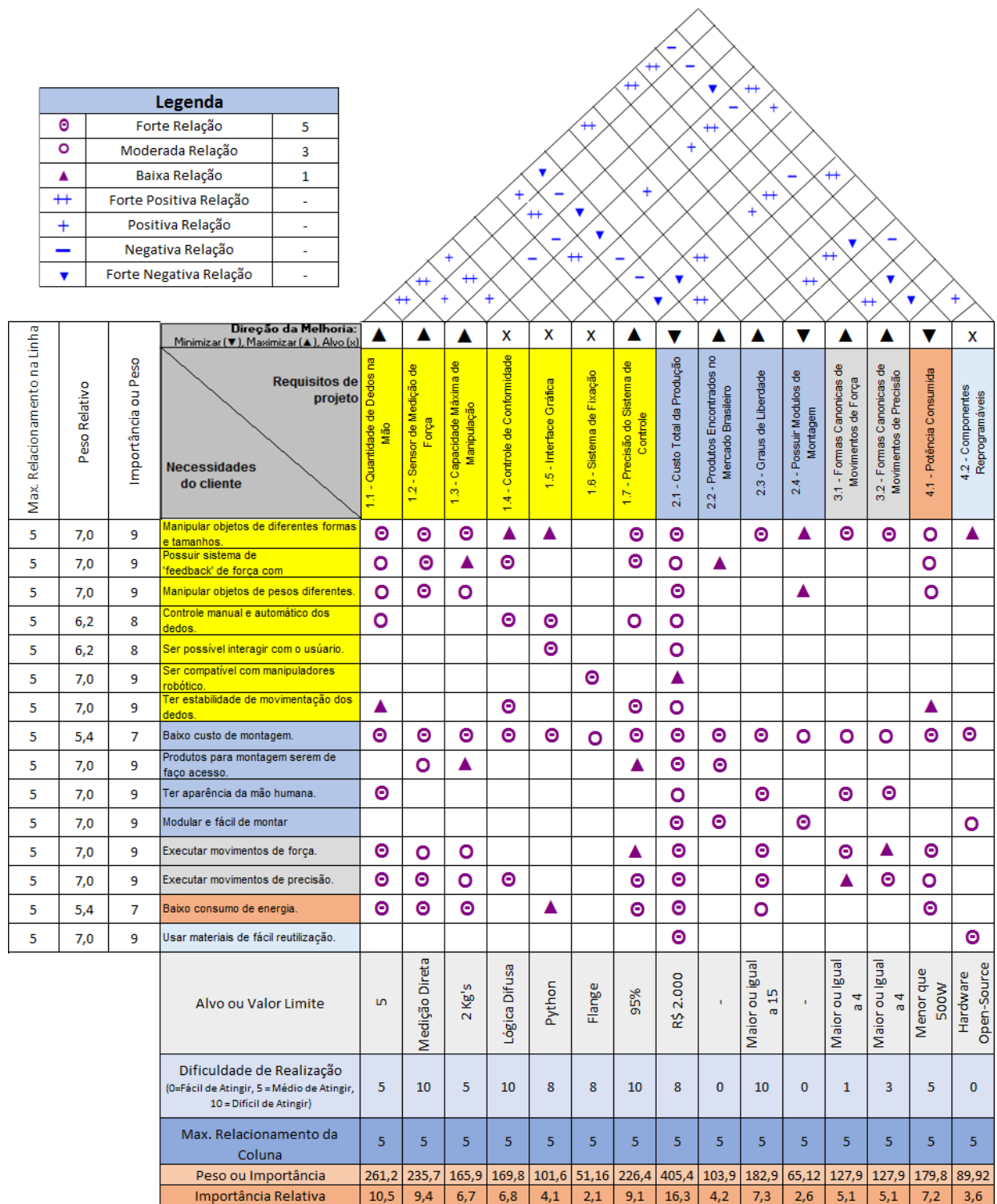


Figura 4.2: Matriz da Casa da qualidade

4.2.6 Análise de competitividade

A análise de competitividade permite identificar vantagens e desvantagens do produto frente aos possíveis concorrentes, que nesse caso são tratados como os trabalhos correlatos. A análise

dentro da metodologia de desenvolvimento de produto não contribuiu para o desenvolvimento do projeto, não sendo considerada tão relevante para o trabalho aqui desenvolvido.

4.2.7 Definição das especificações-meta

Como resultado desta fase, foram obtidas as especificações metas da garra robótica antropomórfica a ser desenvolvida. As especificações metas ou *targets*, são apresentadas sobre a parte inferior da matriz QFD. Cada um dos requisitos de projeto possui uma especificação-meta associada, sendo eles, parâmetros quantitativos e mensuráveis que o produto deverá ter. Diante do exposto, os principais parâmetros da garra robótica controlada pode ser verificado na Tabela 4.3.

Tabela 4.3: Especificação Meta da Garra Robótica Antropomórfica

Requisitos do Produto	Unidade	Valor-Meta
1.1 - Quantidade de dedos nas mãos	Unidade	5
1.2 - Sensor de Medição de Força	-	Medição Direta
1.3 - Capacidade Máxima de Manipulação	Kg	2
1.4 - Controle de Conformidade	-	Lógica Difusa
1.5 - Interface Gráfica	-	<i>Python</i>
1.6 - Sistema de Fixação	Unidade	Flange
1.7 - Precisão do Sistema de Controle	-	95%
2.1 - Custo Total de Produção	R\$	2000
2.2 - Produtos Encontrados no Mercado Brasileiro	-	
2.3 - Graus de Liberdade	Unidade	15
2.4 - Possuir Módulos de Montagem	Unidade	-
3.1 - Formas Canônicas de Movimento de Força	Unidade	4
3.2 - Formas Canônicas de Movimento de Precisão	Unidade	4
4.1 - Potência Consumida	kW/h	menor que 500W
4.2 - Componentes Reprogramáveis	-	<i>Hardware Open-source</i>

4.3 Fase 2 - Projeto ao Nível de Sistema

O projeto em nível de sistemas da-se a partir da estrutura funcional do produto que é representada de forma abstrata, através das funções que o produto deve realizar, independente de qualquer solução particular. Descreve as capacidades desejadas ou necessárias a fim de atingir as especificações-meta do produto, elencadas na seção anterior.

Mediante requisitos do projeto e o problema a ser solucionado, o problema em questão trata-se de *Agarrar Objetos*. A partir da estrutura funcional representada no diagrama de fluxo da Figura 4.3. A função global do produto deve relacionar os sistemas técnicos periféricos, que são classificados em energia, material e sinal, tanto das entradas, como saídas. Aqui foram definidos as entradas e saídas como: *energia elétrica, objetos e comando*; *energia elétrica, objeto agarrado e força de agarre*, respectivamente. Definido a função global, o desdobramento da função global corresponde a um nível inferior, onde as funções do produto são retratadas mais especificamente.

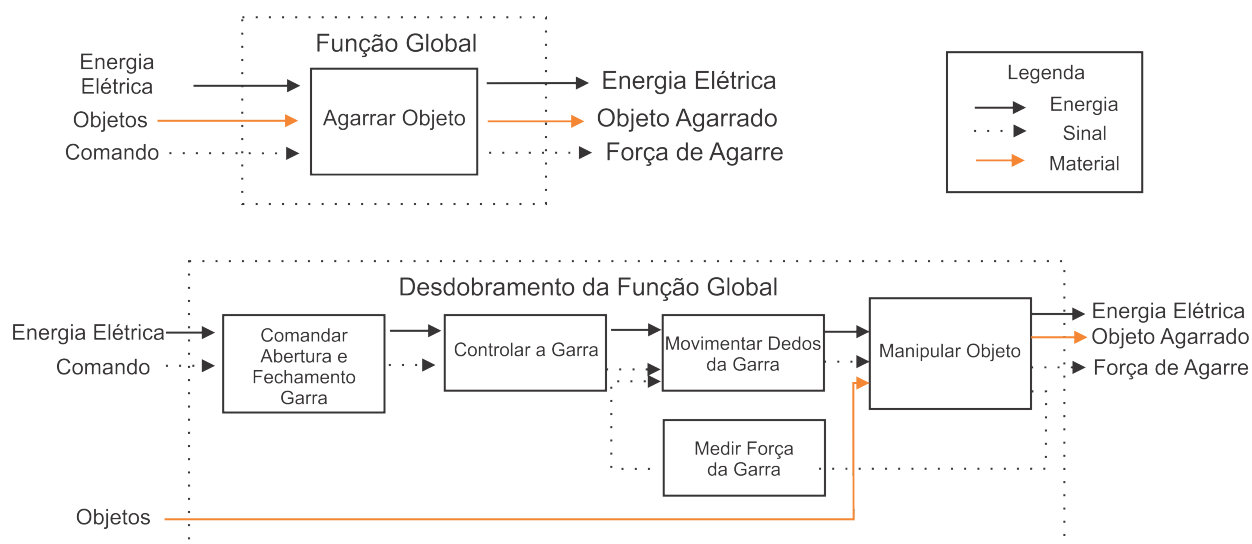


Figura 4.3: Representação da função global e desdobramento da função global

Uma vez definido as funções abstratas do produto, para cada uma das subfunção da estrutura funcional realizada a partir do desdobramento da função global, é atribuído um princípio de solução. Nessa etapa, a ferramenta de apoio ao desenvolvimento de produto que pode ser empregada é a matriz morfológica. Essa ferramenta tem como vantagem a disposição das funções e subfunções estabelecidas anteriormente do produto e as diversas possibilidades de solução para elas.

Primeiramente, preencheu-se a primeira coluna da matriz morfológica com as funções auxiliares definidas na estrutura funcional. Essas funções são chamadas funções de primeira ordem dentro da matriz morfológica apresentada na Tabela 4.4. Posteriormente, as funções de primeira ordem foram decompostas em funções de segunda ordem a fim de aprofundar o nível de detalhamento dos dispositivos, mecanismos e/ou elementos que podem ser usados como solução. As funções de segunda ordem são colocadas sobre a segunda coluna da tabela.

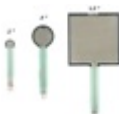
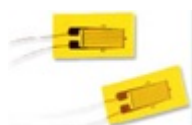









No total foram definidos 10 funções de segunda ordem: 4 para função de movimentar dedos da garra, 1 para manipular objeto, 1 para controlar força, 3 para controlar a garra e 1 para comandar abertura e fechamento da garra robótica antropomórfica. Logo, para cada uma das funções de segunda ordem foram formulados um máximo de três princípios de solução, concebidas com base no processo de *brainstorming* e apoiado sobre pesquisas bibliográficas relacionadas no Capítulo 2. A matriz morfológica completa é apresentada na Tabela 4.4, com as funções sobre a

parte esquerda e os princípios de solução formulados em três colunas (A, B e C) sobre a parte direita.

Tabela 4.4: Matriz morfológica com os princípios de solução

Funções de 1° Ordem	Funções de 2° Ordem	Princípios de solução		
		A	B	C
Movimentar Dedos da Garra	Gerar Movimento	 Micro Motor DC Caixa de Redução	 Mini Servo Motor	 Motor Brushless
	Transmitir Movimento	 Mecanismo de Barras	 Cabo + Polia	 Memória de Liga de Forma
	Tipo de Material	 Impressão 3D	 Alumínio	
	Junta Rotacional	 Rolamento	Material Elástico	
Manipular Objetos	Quantidade de Dedos	 Três Dedos	 Quatro Dedos	 Cinco Dedos
Continua na página seguinte				

Tabela 4.4 – Continuação da página anterior

Funções de 1° Ordem	Funções de 2° Ordem	Princípios de solução		
		A	B	C
Medir Força	Tipo de Sensor	 FSR Sensor	 <i>Strain Gauge</i>	 Sensor Capacitivo
Controlar Garra	Fonte de Alimentação	 Fonte Atx	 Fonte Chaveada	
	Hardware	 Arduino	 Raspberry	 FPGA
	<i>Driver de Potência</i>	 L298N	 ULM2003	 A4988
Comandar Abertura e Fechamento da Garra	Tipo de Software	Proprietário	<i>Open-Source</i>	

Uma vez formulados os princípios de solução, iniciou-se o processo de combinação desses princípios a fim de selecionar aqueles que tenham melhor afinidade técnica. Para isso, foram levados em conta aspectos importantes como a disponibilidade e viabilidade tecnológica e econômica, avaliando a possibilidade de produção com os processos de manufatura disponíveis. Ao

tudo foram geradas três alternativas de solução S1, S2 e S3 apresentadas na Tabela 4.5. Cada alternativa relaciona um princípio de solução à respectiva função de segunda ordem usando as letras A, B ou C designadas na matriz morfológica.

Tabela 4.5: Tabela com as alternativas de solução

Funções de 1° Ordem	Funções de 2° Ordem	Alternativas de Solução		
		S1	S2	S3
Movimentar Dedos da Garra	Gerar Movimento	A	B	A
	Transmitir Movimento	A	C	B
	Tipo de Material	A	A	A
	Junta	B	B	A
Manipular Objetos	Quantidade de Dedos	C	A	C
Medir a Força	Tipo de Sensor	B	A	A
Controlar a Garra	Fonte de Alimentação	B	B	B
	Hardware	A	C	B
	Driver de Potência	C	A	A
Comandar Abertura e Fechamento da Garra	Tipo de Software	A	B	B

Já com o intuito de selecionar uma das alternativas de solução, foi aplicado a matriz de Pugh. Ela permite avaliar cada um dos princípios de solução em relação aos requisitos dos clientes levantados na fase de desenvolvimento do conceito. Para isso, é definida uma escala de desempenho de 1 a 5, onde 1 representa o pior desempenho e 5 o melhor. Dita avaliação é subjetiva, realizada pelo engenheiro projetista (autor). A matriz de Pugh com as avaliações das alternativas de solução para garra robótica antropomórfica é apresentada na Tabela 4.6. Como resultado dessa análise tem-se no final da tabela, as somas ponderadas das pontuações para cada alternativa de solução, sendo a alternativa S3 a que maior pontuação obteve. Isso quer dizer que os princípios de solução correspondente à alternativa S3 serão aplicados ao projeto.

A partir da seleção realizada através da aplicação da matriz de *Pugh*, os princípios de solução são justificados a seguir:

A utilização de micro motor dc com caixa de redução, justifica-se devido ao fato do baixo custo de aquisição, ao mesmo tempo, permite um controle preciso, através de comando por Modulação por Largura de Pulso (*Pulse Width Modulation - PWM*). Possui ainda, redução que amplia sua força, permitindo o agarre de objetos com até 1 Kg. Outrossim, o mesmo por ter pequenas dimensões, possibilita o desenvolvimento de uma garra robótica que se aproxime as dimensões da mão humana.

Já para transmitir os movimentos para os dedos, optou-se por um mecanismo sub-atuado movido por polia + cabo, pois, possibilita manter o grau de liberdade do mecanismo, ao passo, em

Tabela 4.6: Matriz de Pugh

Critérios de Avaliação: (0) Nulo; (1) Insatisfatório; (2) Regular; (3) Bom; (4) Muito Bom; (5) Ótimo				
Necessidades do Usuário	Importância para o Usuário	Alternativas de Solução		
		S1	S2	S3
Ter aparência da mão humana	41	5	5	5
Ser compatível com manipuladores robóticos	35	3	4	5
Possuir sistema de feedback de força com medição direta	30	5	5	5
Ser possível interagir com o usuário	29	3	4	4
Modular e fácil de montar	27	3	5	3
Executar movimentos de força	26	3	4	3
Executar movimentos de precisão	23	3	3	5
Ter estabilidade de movimentação dos dedos	22	3	3	3
Controle manual e automático dos dedos	16	3	3	3
Baixo custo de montagem	16	3	4	5
Baixo consumo de energia	16	5	3	4
Manipular objetos de diferentes formas e tamanhos	9	3	3	5
Produtos para montagem serem de fácil acesso	8	5	5	5
Usar material de fácil reutilização	6	5	5	5
Manipular objetos de pesos diferentes	4	4	4	5
	Total	1.130	1.258	1.313

que o uso do conjunto polia + cabo, permite ao dedo a adaptar-se a forma do objeto a ser agarrado, dando destreza para mão. Diferente de um mecanismo com elos fixos e as características de sub-atuação descritas no Capítulo 2.

Quanto ao material para confecção da garra robótica, a utilização de impressão 3D possibilita a confecção do protótipo em baixo custo, preservando as características de projetos. Permite ainda, testar a solução empregada, bem como reconstruir a peça em caso de erros de projetos, para atender as necessidades especificadas pelos clientes.

O tipo de junta para movimentação dos dedos com rolamento, permite uma movimentação mais suave dos dedos, dando estabilidade no movimento e precisão no agarre. Os enrolamentos são facilmente encontrados no mercado.

Uma garra com 5 dedos teoricamente possibilita a manipulação de diferentes objetos, conforme demonstrado no Capítulo 2. Ela tem como resultado a aparência da mão humana, dando as características de antropomorfismo desejada pelos clientes do produto. Cabe ressaltar que, quanto

maior o número de dedos na garra robótica, mais complexo torna-se o projeto.

Para possibilitar o fechamento da malha do controle de conformidade, o sensor resistivo FSR é empregado a fim de se atender a uma solução simples e de baixo custo, porém, funcional. Caso utiliza-se outros sensores, teria que haver a necessidade do emprego de conversores de sinais, bem como uma eletrônica mais complexa.

Para suprir a alimentação da garra robótica, é utilizada uma fonte chaveada, por ser de fácil aquisição, ao tempo que, supri a alimentação do sistema, atendendo tanto os motores elétricos, como o sistema embarcado empregado para controle efetivo da garra robótica.

Quanto ao controlador, a utilização do microcomputador *Raspberry Pi 3* permite o emprego de soluções customizadas, permitindo integração com *Ethernet*, sendo possível de confeccionar interface gráfica para o usuário e adotar estratégias de controle como PID ou *Fuzzy*. Na atual versão, tem uma boa taxa de processamento de dados e recursos de auxílio ao desenvolvimento de soluções integradas.

Para controle dos motores DC, *drivers* de potência são utilizados, uma vez que, os controladores não conseguem fornecer energia diretamente para os motores, precisando de uma interface de potência. Os drives L298N são de fácil aquisição, possibilitando o controle de motores com até 2A e controle PWM.

Quanto ao sistema de comando de abertura e fechamento da garra, devido à utilização do *Raspberry Pi 3* a integração torna-se mais fácil, permitindo a confecção de uma interface gráfica de baixo custo e código aberto. Uma solução proprietária elevaria os custos do projeto exponencialmente.

Mediante a justificativa dos itens a serem utilizados no desenvolvimento do projeto e considerando as especificações de produto a fim de atender aos requisitos do cliente. Com base na literatura, observou-se que o projeto denominado *The Open Hand Project* [18], visto na Figura 4.14, possui algumas características aqui abordadas, como possuir 5 dedos, mecanismo sub-atuado e motores DC para transmissão dos movimentos. Entretanto, o projeto não possui um sistema de controle de conformidade ativo, sensores de força, análises cinemáticas e eletrônica de controle. Algumas dessas características foram melhoradas e serviram como base para o desenvolvimento da garra robótica, conforme os requisitos dos clientes aqui abordado. Sendo assim, na Figura 4.4 é mostrado a visão geral da garra robótica antropomórfica, com base nos requisitos projetados.

4.4 Fase 3 - Projeto Detalhado

Através da concepção gerada da garra robótica antropomórfica, essa fase tem por objetivo, entrar em detalhe através do modelo V descrito na Subseção 2.8.4, para especificar a garra ao nível mecânico, eletroeletrônico e computacional. No domínio mecânico são descritos os sistemas da

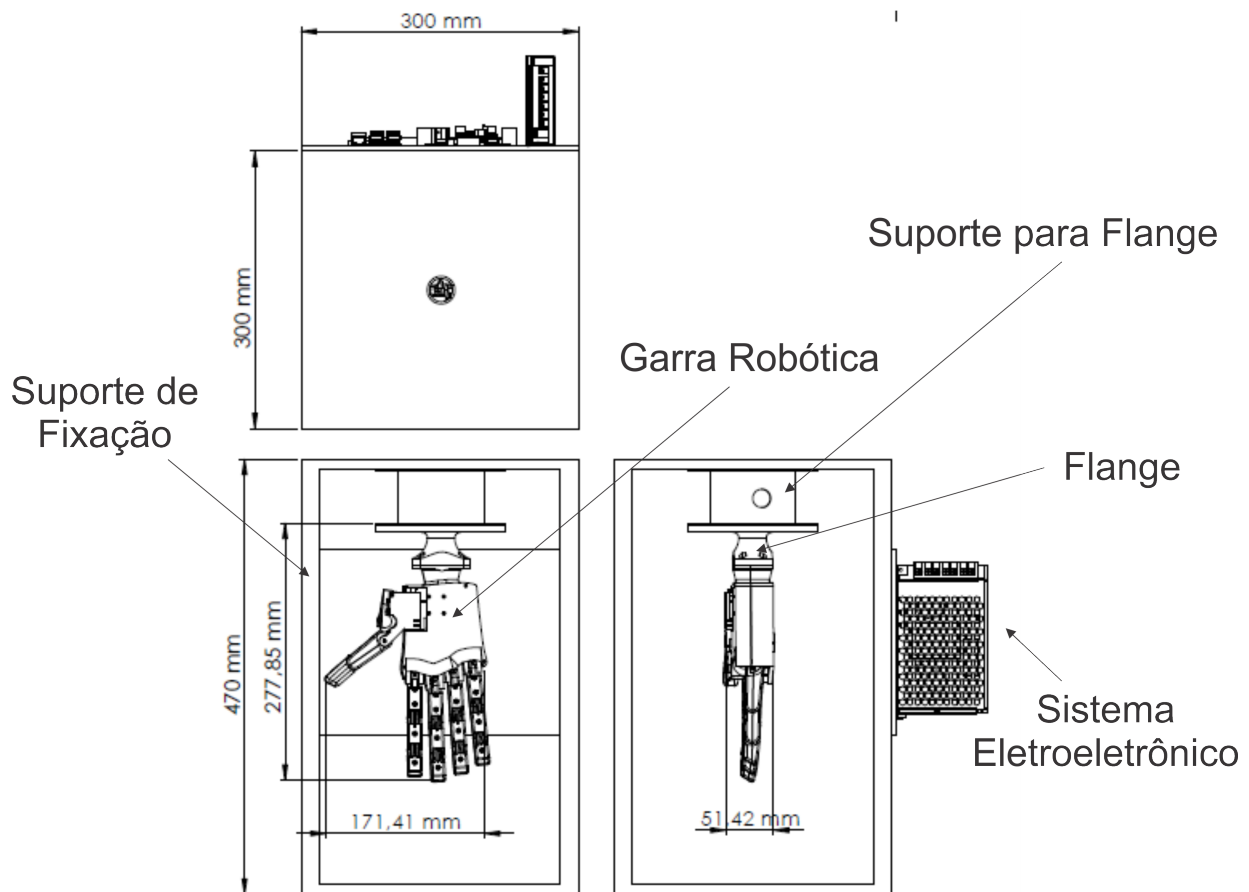


Figura 4.4: Visão geral da garra robótica antropomórfica proposta

estrutura da garra robótica antropomórfica, cálculos cinemáticos e mecanismos de transmissão de movimento. No domínio eletroeletrônico são detalhados diagramas de conexões e componentes e dispositivos eletrônicos utilizados na garra robótica. Já no domínio de projeto computacional são descritos o algoritmo e o software de controle. O processo inicia com o domínio do projeto mecânico a seguir:

4.4.1 Projeto Mecânico

No projeto mecânico é realizado os cálculos de mobilidade e cinemática da garra robótica, posteriormente é descrito a estrutura, a qual foi realizado com modelagem CAD. Entretanto, cabe ressaltar que para os domínios aqui descritos, foi considerado as restrições do projetista, uma vez que, o desenvolvimento de garra robótica possui caráter interdisciplinar conforme descrito na descrição do problema.

4.4.2 Análise de mobilidade e modelagem cinemática

O número de graus de liberdade de um mecanismo pode ser calculado através da equação de Gruebler, onde M é o grau de liberdade ou mobilidade do mecanismo, L é o número de juntas e G o número de elos fixados, conforme descreve Norton [112], representado pela Equação 4.1.

$$M = 3L - 2J - 3G \quad (4.1)$$

Para o problema proposto, podemos representá-lo como um sistema de 5 barras paralelas, onde ao todo temos 20 elos e 15 juntas rotacionais, sendo 5 desses elos fixados à base, conforme representação da Figura 4.5. Uma vez substituindo os valores na Equação 4.1, é obtida uma cadeia cinemática de 15 graus de liberdade para a garra robótica proposta.

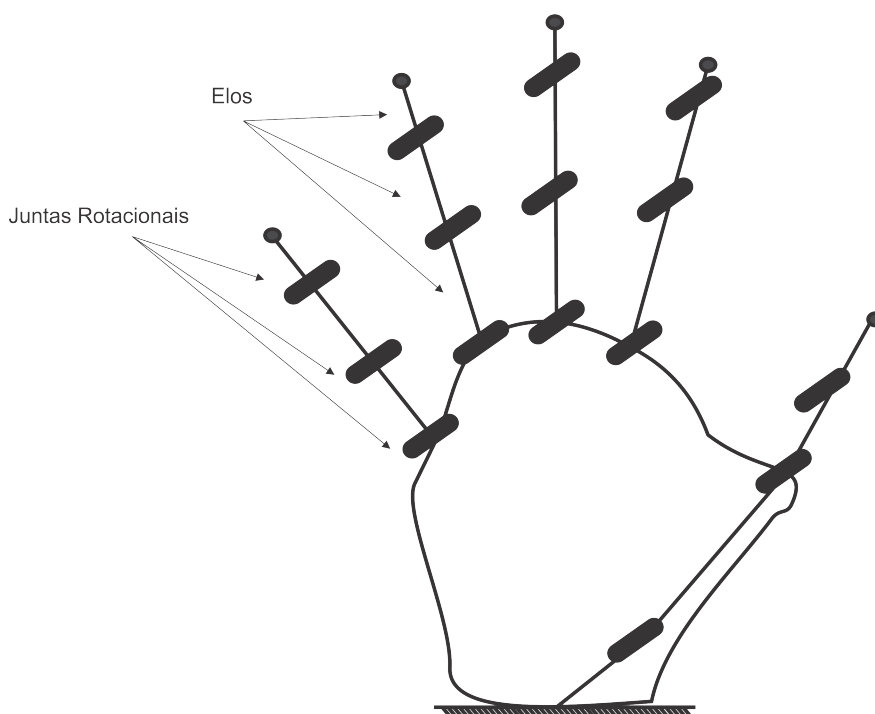


Figura 4.5: Representação de barras para a garra robótica desenvolvida

4.4.2.1 Cinemática Direta

A cinemática direta nos permite calcular onde o robô se encontra em qualquer instante, sendo para o trabalho aqui desenvolvido, onde se encontra posicionado os dedos da garra robótica antropomórfica, considerando os ângulos das juntas rotacionais. A técnica de Denavit-Hartenberg (DH) foi proposta em torno de 1955 e tornou-se padrão para representar e modelar robôs e derivar suas equações de movimento atualmente. A técnica de DH tem como benefícios a análise

de movimentos diferenciais e Jacobianos, análise dinâmica, análise de forças [113] e outras que tomam como base os resultados obtidos com a técnica, que serão objetos de trabalhos futuros.

De forma resumida, a modelagem por cinemática direta a partir da notação de DH, entre dois referências, pode ser representado pela transformação das coordenadas presentes em relação as coordenadas anteriores representados por ${}^nT_{n-1}$, representado pela Equação 4.2.

$${}^nT_{n-1} = Rot(z, \theta_{n+1}).Trans(0, 0, d_{n+1}).Trans(a_{n+1}, 0, 0).Rot(x, \theta_{n+1}) \quad (4.2)$$

Sendo que na forma matricial é representado conforme a Equação 4.3, onde $c = \cos$ e $s = \sin$.

$${}^nT_{n-1} = \begin{bmatrix} c\theta_{n+1} & -s\theta_{n+1}c\alpha_{n+1} & s\theta_{n+1}s\alpha_{n+1} & a_{n+1}c\theta_{n+1} \\ s\theta_{n+1} & c\theta_{n+1}c\alpha_{n+1} & -c\theta_{n+1}s\alpha_{n+1} & a_{n+1}s\theta_{n+1} \\ 0 & s\alpha_{n+1} & c\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

4.4.2.2 Cinemática direta para o dedo mínimo, anelar, médio e indicador

Para a garra robótica antropomórfica desenvolvida nesse trabalho, os referencias locais para o dedo mínimo, anelar, médio e indicador, podem ser visualizados conforme Figura 4.6. Através da figura é possível constatar que os sistemas de referência têm eixos mutuamente paralelos.

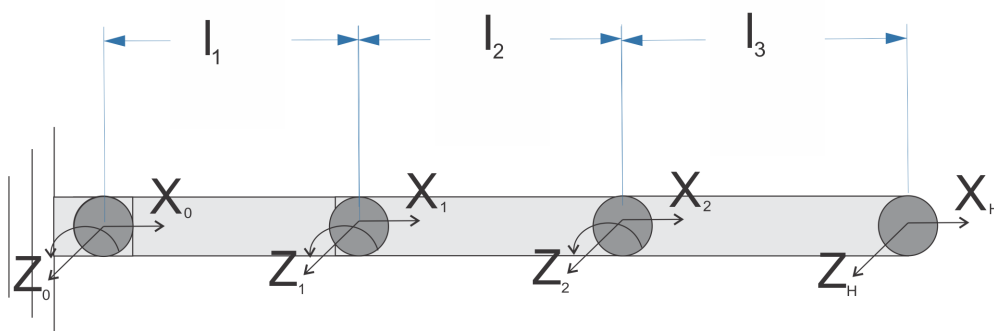


Figura 4.6: Representação dos referenciais locais do dedo mínimo, anelar, médio e indicador

A cinemática direta para o dedo mínimo, anelar, médio e indicador, conforme demonstrado na Figura 4.6 é definida conforme os parâmetros da Tabela 4.7 e da Equação 4.4, que representam as matrizes que contém a posição e orientação das pontas dos dedos em relação à base.

$$T_3^0 = T_1^0.T_2^1.T_3^2 \quad (4.4)$$

Tabela 4.7: Parâmetros DH para os dedos mínimo, anelar, médio e indicador

Junta	θ	α	l	d
MCP	θ_1	0	l_1	0
PIP	θ_2	0	l_2	0
DIP	θ_3	0	l_3	0

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$T_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

$$T_3^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Substituindo a Equação 4.5, 4.6, 4.7 na Equação 4.4 e efetuando o produto, obtêm-se as posições das extremidades dos dedos representados pela quarta coluna da Equação 4.8.

$$T_0^3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

4.4.2.3 Cinemática direta para o polegar

A cinemática direta para o dedo polegar foi definida analogamente ao procedimento empregado anteriormente. Sendo os referencias locais para o polegar de acordo com a notação de Denavit-Hartenberg, demonstrado na Figura 4.7 e os parâmetros explícitos na Tabela 4.8.

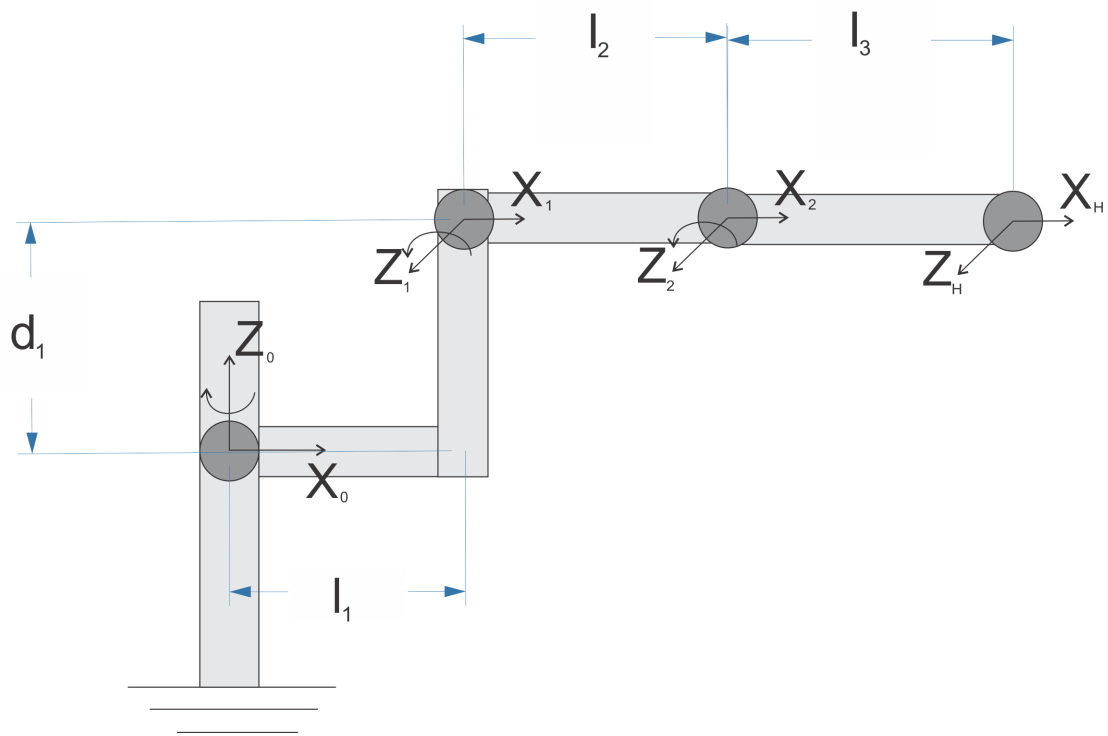


Figura 4.7: Representação dos referenciais locais para o polegar

Tabela 4.8: Parâmetros DH para o polegar

Junta	θ	α	l	d
RC	θ_1	90	l_1	d_1
MCP	θ_2	0	l_2	0
DIP	θ_3	0	l_3	0

A substituição dos parâmetros da Tabela 4.8 nas suas respectivas matrizes de transformação homogêneas resulta nas Equações 4.9, 4.10 e 4.11.

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & l_1 \cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & l_1 \sin(\theta_1) \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

$$T_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

$$T_3^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

Substituindo as Equações 4.9, 4.10 e 4.11 na Equação 4.4, obtém-se, como resultado da multiplicação, as posições das extremidades do dedo polegar, contidas na quarta coluna da matriz indicada na Equação 4.12.

$$T_0^3 = \begin{bmatrix} c_{23} \cdot c_1 & -s_{23} \cdot c_1 & s_1 & c_1(l_1 + l_3 c_{23} + l_2 c_2) \\ c_{23} \cdot s_1 & -s_{23} \cdot s_1 & -c_1 & s_1 \cdot (l_1 + l_3 \cdot c_{23} + l_2 c_2) \\ s_{23} & c_{23} & 0 & d_1 + l_3 \cdot s_{23} + l_2 \cdot s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

4.4.2.4 Cinemática Inversa

Já a cinemática inversa objetiva encontrar o valor de cada articulação, possibilitando colocar o robô na posição e orientação desejada. Enquanto as equações da cinemática direta tem uma infinidade de ângulos acoplados como c_{123} impossibilitando encontrar elementos suficientes na matriz para resolver os senos e cossenos individuais para calcular os ângulos, a cinemática inversa possibilita multiplicar a matriz ${}^R T_H$ por matrizes individuais A_n^{-1} resultando encontrar os senos e cossenos do ângulo e, por fim, o ângulo em si [113].

4.4.2.5 Cinemática inversa para os dedos mínimos, anelar, médio e indicador

Analisando a configuração cinemática dos dedos mínimo, anelar, médio e indicador, observa-se que todos os eixos das juntas rotacionais são paralelos, o que se assemelha a um manipulador planar simples de três elos.

Pode ser visto a partir da Equação 4.8 da cinemática direta para estes dedos que a posição e a orientação do elemento terminal estão relacionadas às variáveis de juntas por meio de equações transcendentais não-lineares. A solução da cinemática inversa implica em resolver essas equações não-lineares. Idealmente, o objetivo consiste em obter expressões de forma fechada e analítica para as variáveis de juntas em termos da posição e da orientação do último elo.

Para esta análise cinemática, é necessário assumir que as transformações necessárias são realizadas de forma que o ponto alvo é uma especificação do sistema de referência da ponta do dedo em relação ao sistema de referência da base, ou seja, T_W^B . Pelo fato da cinemática destes dedos ser semelhante ao de um manipulador planar, a especificação desses pontos alvos pode ser

obtida com mais facilidade especificando-se três números: x , y e ϕ sendo a orientação do elo 3 no plano (em relação ao eixo $+X$). Dessa forma, ao invés de fornecer um T_W^B genérico como especificação de alvo [114], é necessário considerar uma transformação com a estrutura mostrada pela Equação 4.13.

$$T_W^B = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

Todos os alvos alcançáveis devem estar no subespaço indicado pela estrutura da Equação 4.13. Igualando a Equação 4.8 com a Equação 4.13, chega-se a um conjunto de quatro equações não lineares representadas pelas Equações 4.14, 4.15, 4.16 e 4.17 que precisam ser resolvidas para θ_1 , θ_2 e θ_3 .

$$c_\phi = c_{123} \quad (4.14)$$

$$s_\phi = s_{123} \quad (4.15)$$

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123} \quad (4.16)$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123} \quad (4.17)$$

Onde o ângulo ϕ é dado pela Equação 4.18.

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (4.18)$$

Substituindo as Equações 4.14 e 4.15, nas Equações 4.16 e 4.17 respectivamente, obtêm-se as Equações 4.19 e 4.20.

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_\phi \quad (4.19)$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_\phi \quad (4.20)$$

Definindo X e Y pelas Equações 4.21 e 4.22, têm-se:

$$X = x - l_3 c_\phi \quad (4.21)$$

$$Y = y - l_3 s_\phi \quad (4.22)$$

Nas Equações 4.21 e 4.22, X e Y podem ser substituídos por suas definições expressas pelas Equações 4.16 e 4.17, o que resulta nas expressões matemáticas representadas pelas Equações 4.23 e 4.24.

$$X = l_1 c_1 + l_2 c_{12} \quad (4.23)$$

$$Y = l_1 s_1 + l_2 s_{12} \quad (4.24)$$

Elevando as Equações 4.23 e 4.24 ao quadrado e somando as duas, obtém-se a Equação 4.25.

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2 \quad (4.25)$$

Resolvendo a Equação 4.25 para c_2 , têm-se como resultado a Equação 4.26.

$$c_2 = \frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (4.26)$$

Portanto, o valor de θ_2 é dado pela Equação 4.27.

$$\theta_2 = \pm \cos^{-1} \left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \quad (4.27)$$

Sabendo que o *coseno* e o *seno* da soma de dois ângulos são dados, respectivamente, pelas Equações 4.28 e 4.29, e substituindo nas Equações 4.23 e 4.24, têm-se como resultado, as Equações 4.30 e 4.31.

$$c_{12} = c_1 c_2 - s_1 s_2 \quad (4.28)$$

$$s_{12} = c_1 s_2 - s_1 c_2 \quad (4.29)$$

$$X = k_1 c_1 - k_2 s_1 \quad (4.30)$$

$$Y = k_1 s_1 + k_2 c_1 \quad (4.31)$$

Nas Equações 4.30 e 4.31, k_1 e k_2 são representados, respectivamente, pelas Equações 4.32 e 4.33.

$$k_1 = l_1 + l_2 c_2 \quad (4.32)$$

$$k_2 = l_2 s_2 \quad (4.33)$$

Com o intuito de resolver as equações acima, é necessário fazer uma troca de variáveis, o que, na prática significa mudar a forma de escrever as constantes k_1 e k_2 .

Se r for dado pela Equação 4.34, e γ pela Equação 4.35, têm-se k_1 e k_2 representados pelas Equações 4.36 e 4.37.

$$r = +\sqrt{k_1^2 + k_2^2} \quad (4.34)$$

$$\gamma = \arctan 2(k_2, k_1) \quad (4.35)$$

$$k_1 = r \cos(\gamma) \quad (4.36)$$

$$k_2 = r \sin(\gamma) \quad (4.37)$$

Substituindo as Equações 4.36 e 4.37 nas Equações 4.30 e 4.31, obtêm-se as Equações 4.38 e 4.39.

$$\frac{X}{r} = \cos(\gamma) \cos(\theta_1) - \sin(\gamma) \sin(\theta_1) \quad (4.38)$$

$$\frac{Y}{r} = \cos(\gamma) \sin(\theta_1) + \sin(\gamma) \cos(\theta_1) \quad (4.39)$$

Analisando as Equações 4.38 e 4.39, observa-se que o lado direito destas expressões trata-se de uma soma de ângulos, portanto, substituindo pelas devidas identidades trigonométricas, obtêm-se as Equação 4.40 e 4.41.

$$\cos(\gamma + \theta_1) = \frac{X}{r} \quad (4.40)$$

$$\sin(\gamma + \theta_1) = \frac{Y}{r} \quad (4.41)$$

Usando a função arco tangente de dois argumentos, obtêm-se a Equação 4.42.

$$\gamma + \theta_1 = \arctan 2 \left(\frac{Y}{r}, \frac{X}{r} \right) = \arctan 2(Y, X) \quad (4.42)$$

Substituindo a Equação 4.35 na Equação 4.42, e resolvendo para θ_1 , tem-se a Equação 4.43.

$$\theta_1 = \arctan 2(Y, X) - \arctan 2(k_2, k_1) \quad (4.43)$$

É importante frisar que a escolha de sinal na solução de θ_2 provoca uma mudança de sinal em k_2 , afetando em consequência k_1 . Também é importante notar que se $x = y = 0$, a Equação 4.43 torna-se indefinida, e θ_1 torna-se arbitrário.

Por último, com base na Equação 4.18, é possível determinar o valor de θ_3 , expresso pela Equação 4.44.

$$\theta_3 = \phi - \theta_1 - \theta_2 \quad (4.44)$$

4.4.2.6 Cinemática inversa para o dedo polegar

O objetivo da cinemática inversa do dedo polegar é encontrar θ_i quando T_3^0 é dado em valores numéricos, onde T_3^0 é representado pela Equação 4.4. Colocando a dependência em θ_1 da Equação 4.4 do lado esquerdo dessa, obtêm-se a Equação 4.45.

$$(T_1^0)^{-1} \cdot T_3^0 = T_2^1 \cdot T_3^2 \quad (4.45)$$

Essa técnica simples de multiplicar cada lado de uma equação de transformação por uma inversa é muito usada com vantagens para a separação de variáveis em busca de uma equação solucionável.

Invertendo a matriz T_1^0 , é possível escrever a Equação 4.45 na forma indicada pela Equação 4.46:

$$\begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & -l_1 \\ 0 & 0 & 1 & -d_1 \\ \sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & l_3 \cdot \cos(\theta_2 + \theta_3) + l_2 \cdot \cos(\theta_2) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & l_3 \cdot \sin(\theta_2 + \theta_3) + l_2 \cdot \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.46)$$

Igualando os elementos (1,3) dos dois lados da Equação 4.46, permite obter a expressão indicada pela Equação 4.47.

$$\cos(\theta_1) \cdot r_{13} + \sin(\theta_1) \cdot r_{23} = 0 \quad (4.47)$$

Reorganizando os termos da Equação 4.47, chega-se à solução de θ_1 , indicado pela Equação 4.48.

$$\theta_1 = \text{atan2}(-r_{13}, r_{23}) \quad (4.48)$$

Considerando ainda a Equação 4.46, e igualando os elementos (1,1), obtém-se a Equação 4.49 expressa como:

$$\cos(\theta_1) \cdot r_{11} + \sin(\theta_1) \cdot r_{21} = \cos(\theta_2 + \theta_3) \quad (4.49)$$

Aplicando o arco cosseno, encontra-se a expressão matemática para a soma dos ângulos θ_2 e θ_3 , indicada pela Equação 4.50.

$$\theta_{23} = \cos^{-1}(\cos(\theta_1).r_{11} + \sin(\theta_1).r_{21}) \quad (4.50)$$

De maneira semelhante, a partir da Equação 4.46 é possível determinar o valor de θ_2 . Para tanto, deve-se igualar os elementos (1,4), conforme indicado pela Equação 4.51.

$$\cos(\theta_1).p_x + \sin(\theta_1).p_y - l_1 = l_3 \cdot \cos(\theta_2 + \theta_3) + l_2 \cdot \cos(\theta_2) \quad (4.51)$$

Isolando θ_2 na Equação 4.51, obtém-se o modelo matemático expresso pela Equação 4.52.

$$\theta_2 = \cos^{-1} \cdot \frac{\cos(\theta_1).p_x + \sin(\theta_1).p_y - l_1 - l_3 \cdot \cos(\theta_2 + \theta_3)}{l_2} \quad (4.52)$$

O termo θ_{23} também pode ser calculado pela Equação 4.53.

$$\theta_{23} = \theta_2 + \theta_3 \quad (4.53)$$

Reorganizando a Equação 4.53 para determinar θ_3 , obtém-se a Equação 4.54.

$$\theta_3 = \theta_{23} - \theta_2 \quad (4.54)$$

4.4.3 Descrição dos sistemas do domínio mecânico

A descrição da garra robótica antropomórfica é realizada a partir do domínio mecânico através de modelagem CAD. A garra robótica será descrita da seguinte forma para melhor entendimento: dedos principais (mínimo, anelar, médio e indicador), dedo polegar, base da palma, dorso da palma e flange. Para visualização da montagem do projeto completo, observar a Figura 4.13. Os dedos principais, possuem uma estrutura única entre todos os dedos que são replicados para os demais, exceto o polegar. Os dedos principais são descritos como a Figura 4.8.

Os dedos (mínimo, anelar, médio e indicador), representados na Figura 4.8 possuem motores DC (12) que através da polia em espiral (13) transmite os movimentos para um fio de nylon que está conectado à falanges MCP, PIP e DIP. A polia possui uma estrutura com fixação do fio de nylon para possibilitar enrolar em torno de si para possibilitar flexionar ou esticar o dedo. A base dos motores DC (10) possibilita a fixação dos motores, enquanto os extensores (16) junto com os enrolamentos com sulco em V (18) possibilitam o tensionamento do fio de nylon através

da mola (19). A estrutura do dedo, sendo as falanges MCP (4 e 1), PIP (7 e 6) e DIP (9 e 8) são conectados através do eixo (11) a base do motor para possibilitar transmitir os movimentos. As falanges, possuem rolamentos (5) para possibilitar uma movimentação suave, enquanto as falanges são fixados pelos parafusos e porca (2)(3). Os sensores de força FSR ficam na ponta dos dedos (20) para possibilitar o controle de conformidade ativo. A montagem dos dedos pode ser visualizado na Figura 4.13.

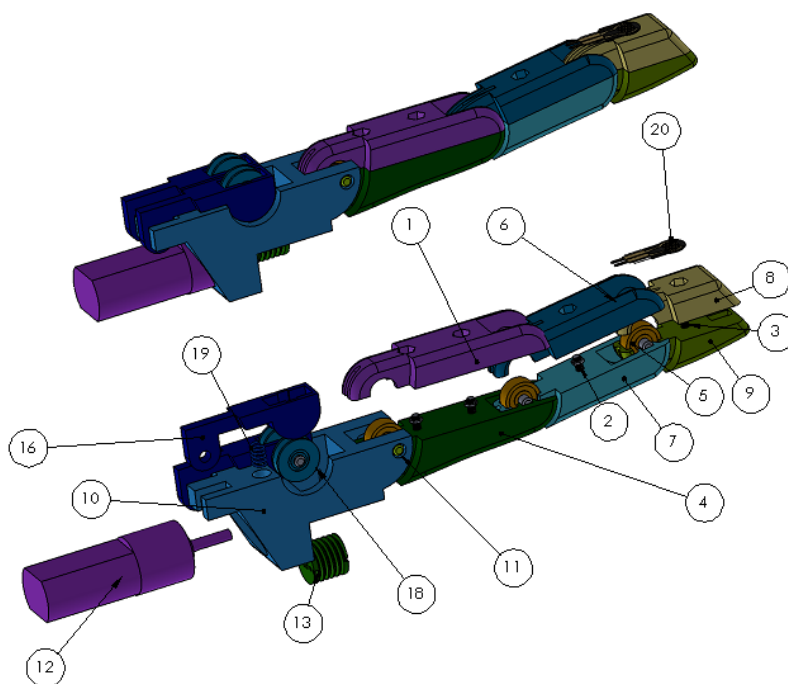


Figura 4.8: Descrição dos componentes comuns aos dedos mínimo, anelar, médio e indicador

Já o polegar, representado na Figura 4.9, possui uma montagem diferente, pois, o mesmo além de realizar os movimentos de extensão e flexão, faz os movimentos de abdução e adução. Para possibilitar os movimentos de extensão e flexão, possui motores DC fixado no dorso da palma da mão, representado na Figura 4.11. Através dos conjuntos de extensores do fio de nylon (13), mola (15), rolamentos (9) e eixo (10), possibilitam a movimentação das falanges MCP (5 e 6) e DIP (7 e 8). A movimentação da falange RC que é constituída pela base do suporte (1) do servo motor (2) possibilita os movimentos de abdução e adução. A base da falange RC é fixada no dorso da palma, podendo ser visualizado na Figura 4.13. A haste do servo motor (3) possibilita transmitir os movimentos do servo para o conjunto de movimentação do polegar, enquanto o conjunto de rolamento (4) possibilitam a movimentação suave das falanges. As falanges são fixadas pelos parafusos (11)(12). O sensor FSR (16) possibilita o controle de conformidade do polegar.

A base da mão (palma), representado na Figura 4.10 possui sulcos para fixação dos dedos, onde os mesmos são encaixados através da base dos motores (10) principais. Os sulcos possuem uma angulação de 3° entre eles, para possibilitar as características de antropomorfismo da mão humana, caso contrário a mesma possuirá aspectos rudimentares. A base também possibilita a

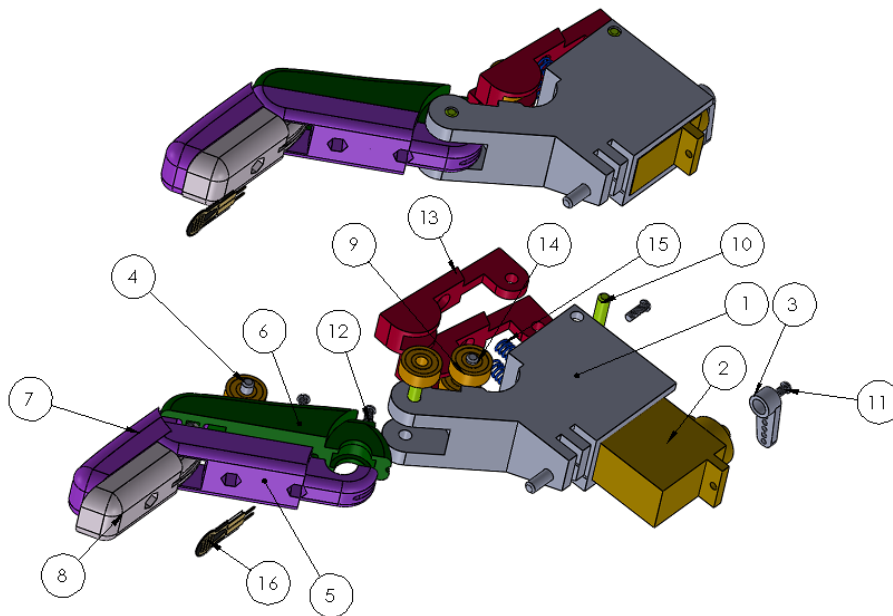


Figura 4.9: Descrição dos componentes para o dedo polegar

passagem dos cabos dos motores para o microcomputador. O suporte de fixação da flange é integrado a palma da mão, sendo uma peça única para possibilitar a montagem em manipuladores robóticos.

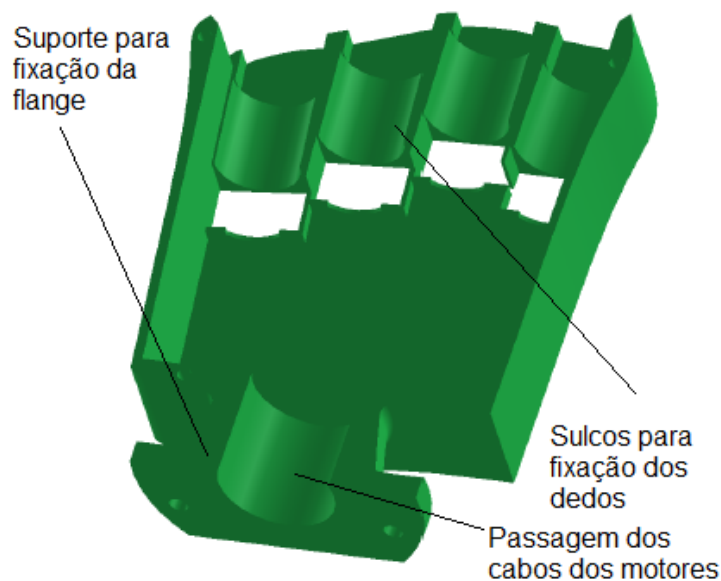


Figura 4.10: Descrição da base da palma

O dorso da palma, representado na Figura 4.11, possibilita o alojamento do conjunto de movimentação do polegar, assim como serve de suporte para manter a base de suporte dos motores (10) dos dedos principais fixados. A base tem uma parte para suporte do motor DC para possibi-

litar movimentação do polegar, que junto com a base de fixação da flange completam a estrutura do dorso da palma. Tanto a base da palma da mão, como o dorso, podem ser visualizados na Figura 4.13.

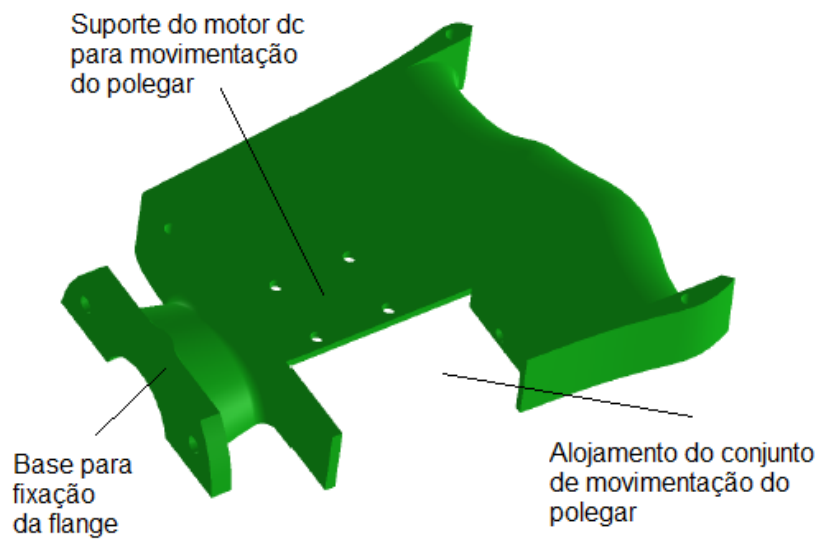


Figura 4.11: Descrição do dorso da palma

A Figura 4.12 constitui a flange para possibilitar a integração da garra robótica antropomórfica com manipuladores robóticos industriais, conforme as necessidades dos clientes definido na Tabela 4.1.

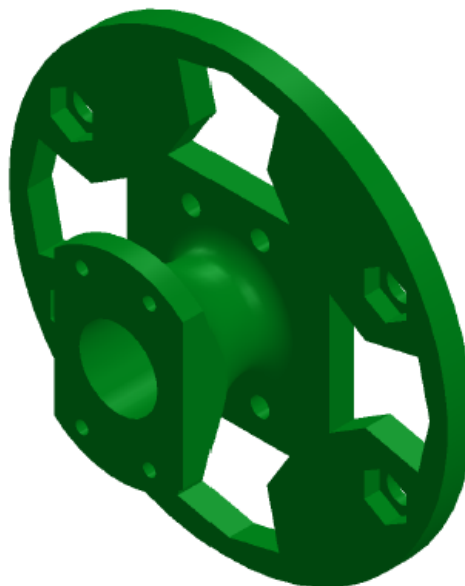


Figura 4.12: Flange para acoplamento em manipuladores robóticos

A montagem da mão completa com todas as partes descritas anteriormente e representadas

por cores, podem ser visualizadas na Figura 4.13 para melhor entendimento e compreensão. A montagem completa da garra robótica antropomórfica possui 47 peças.

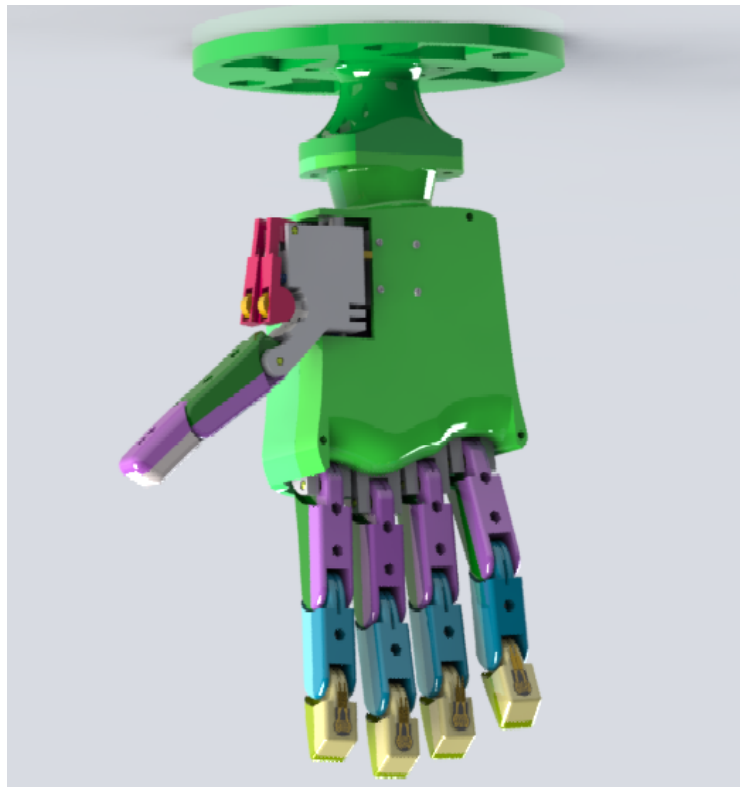


Figura 4.13: Desenho CAD renderizado da garra robótica antropomórfica completa

A título de comparação, a garra robótica desenvolvida pelo projeto *The Open Hand* [18] pode ser visualizada na Figura 4.14. Ela possui anatomia geral diferente da garra aqui desenvolvida, pois, não foi projetada com o intuito de receber sensores para medição direta de força. O dorso e a palma da mão possuem estrutura diferente, assim como o suporte para integração com manipuladores robóticos. Ao todo a garra possui 15 GDL, sendo que, não há um sistema de controle de conformidade previsto junto com a eletrônica de controle. Outrossim, conforme relatado anteriormente, não possui estudos cinemáticos disponíveis.

4.4.4 Projeto eletroeletrônico

O diagrama do sistema eletroeletrônico da garra robótica antropomórfica é apresentado no Apêndice D. O sistema utiliza duas placas *RaspberryPi 3 B+* para atender aos requisitos dos clientes possuindo entre elas comunicação através do protocolo de Fila Mensageria de Transporte de Telemetria (*Message Queuing Telemetry Transport - MQTT*) . Em uma placa é executado o sistema de controle de conformidade com lógica *fuzzy* e a outra é responsável pela geração da interface gráfica para o usuário, bem como o controle de acionamento das formas canônicas e visualização gráfica da força de atuação sobre o objeto. As vantagens de usar a *Raspberry* incluem



Figura 4.14: Garra robótica antropomórfica desenvolvida pelo projeto *The Open Hand*

uma tecnologia aberta, de baixo custo, fácil de usar, alto processamento e possui compatibilidade com grande quantidade de módulos eletrônicos para distintas aplicações.

Os drives utilizados para controle dos micro motores DC são da referência L298N, os quais são amplamente utilizados para controle de motores, permitindo o controle de motores DC com até 2A, além de permitir a inversão de giro e controle PWM. No total, são utilizados 3 *drivers* L298N, sendo que cada *driver* possibilita o controle de até 2 motores simultaneamente. A garra robótica utiliza 5 motores DC com caixa de redução, com força de até 1 kg/cm e velocidade de 100 rpm. Os motores são de fácil aquisição através do mercado nacional. Quanto aos sensores, são utilizados sensores de força FSR que tem sensibilidade ao toque, onde uma vez variando a resistência ao toque, permite o controle da tensão que é lido por uma entrada digital através de um multiplexador M3008 para a placa de controle.

Devido à utilização do servomotor para possibilitar as movimentações de abdução e adução do dedo polegar, e pelo mesmo ser alimentado por 6Vdc e como a fonte de alimentação utilizada disponibiliza 12Vdc até 20A, foi necessário utilizar um conversor de tensão DC-DC 35-4V LM2596. O projeto completo a fim de ser reproduzido com os códigos G para usinagem da placa eletrônica, diagramas e layouts, podem ser encontrados no site www.arturvitorio.weebly.com.

4.4.5 Projeto software de controle

Neste último domínio de projeto, são descritos os módulos de software utilizado para o controle do sistema da garra robótica antropomórfica, além do controle de conformidade os quais são baseados em código aberto. De forma geral, o sistema computacional da garra robótica é

composto por um algoritmo principal que realiza a parte de controle da garra robótica em uma *Raspberry*. Em outra *Raspberry* há um segundo algoritmo que realiza a geração da interface gráfica para possibilitar ao usuário o acionamento dos dedos e as formas canônicas.

O projeto pode ser melhor representado pelo caso de uso descrito na Figura 4.15. Ele é composto por 7 atores, sendo o primeiro ator representado pelo *Usuário* que irá controlar a garra robótica. Ele tem por função ativar os modos de controle, selecionar a forma canônica de agarre e solicitar o gráfico de força.

O ator *Controlador Manual* comanda os dedos individualmente, sem a especificação de força, além de controlar o ângulo de posição do polegar para possibilitar rotacionar em torno da palma da mão. A movimentação dos dedos da garra robótica e o posicionamento do polegar são controlados através dos atores *Motor DC* e *Servo Motor*, respectivamente. O ator *Controlador Automático* é responsável, através da forma canônica de agarre selecionado, controlar a força que será aplicada ao objeto dependendo da força que é medida pelo ator *Sensor de Força*. O caso de uso controlar força é gerenciado pelo ator *Controle Fuzzy*.

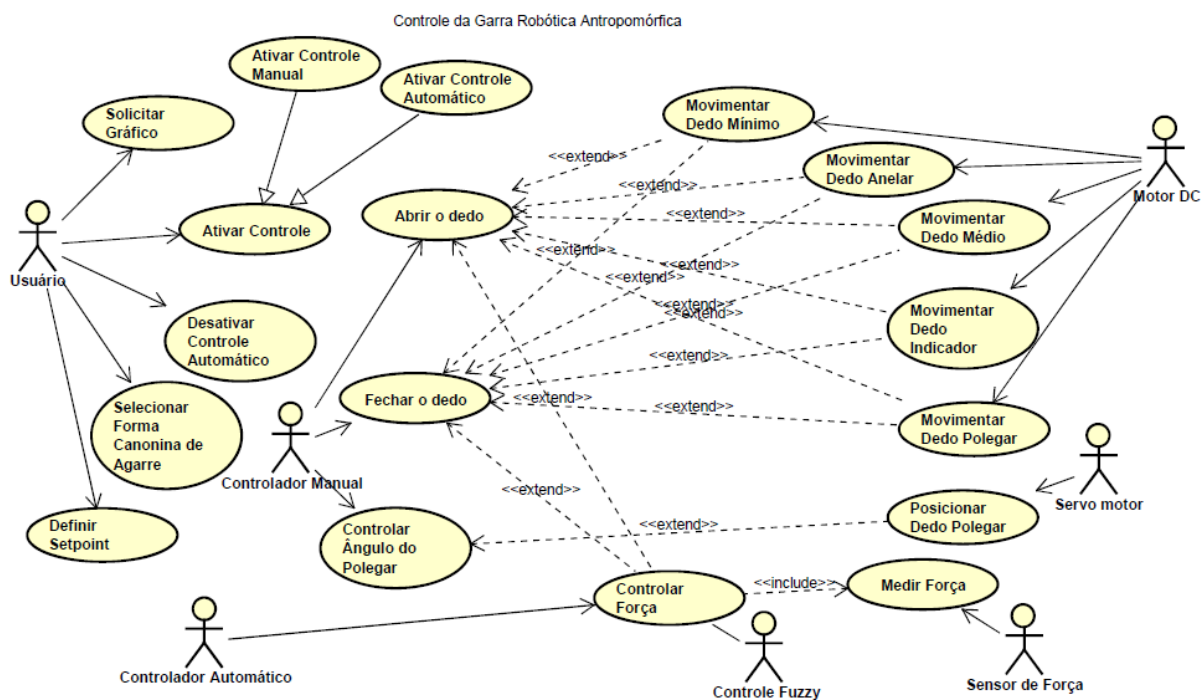


Figura 4.15: Caso de Uso para o projeto de Software da Garra Robótica

A interface gráfica é desenvolvida baseada no software *Node-RED*. O mesmo permite através de fluxos a programação visual para conectar dispositivos, hardware, Interface de Programação de Aplicativos (*Application Programming Interface - API*), entre outros para a internet das coisas. A interface gráfica é constituída pelo controle manual e automático. No modo manual é possível o controle independente de cada dedo. O polegar é constituído de um *slicer* de controle do ângulo do setpoint, possibilitando os movimentos de abdução e adução. O controle automático, possibilita o agarre prévio de determinada forma canônica de agarre. Sendo que para o controle

aqui desenvolvido, possui quatro: pegar garrafa, pegar tipo pinça, pegar três dedos e pegar quatro dedos. Ainda conta com um gráfico de tendência do dedo médio para verificação da aplicação da força. A interface gráfica desenvolvida pode ser observado na Figura 4.16.

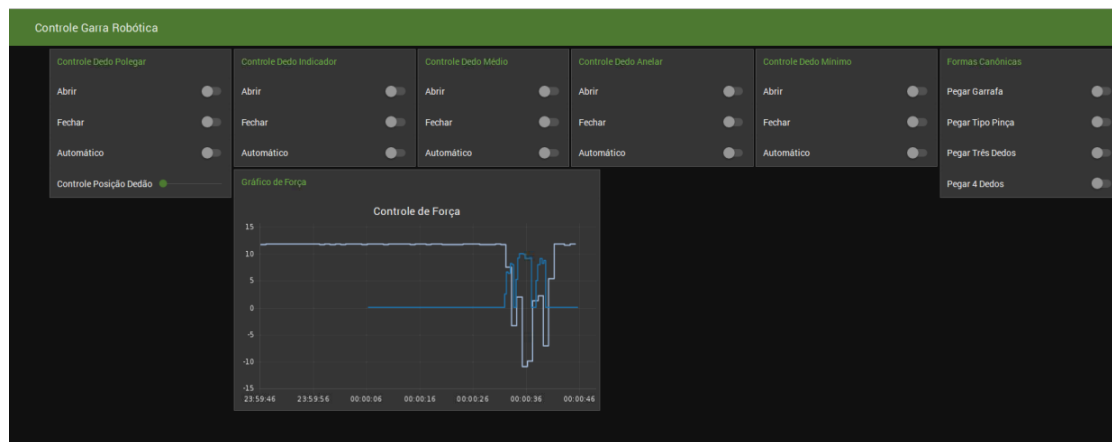


Figura 4.16: Interface Gráfica do Usuário

4.4.5.1 Comunicação MQTT

A comunicação entre os *Raspberry Pi 3 B+* é realizada através de um servidor *broker* que gerencia as informações que serão disponibilizadas no canal de comunicação através de uma topologia de publicação e assinante, descritos no protocolo de comunicação MQTT. O *broker* recebe todas as informações dos clientes e, em seguida, roteia essas mensagens para os clientes com destino especificado. O cliente, por sua vez, pode ser desde um sensor, outra *Raspberry* ou um aplicativo em um data center que processa dados de Internet das Coisas (IoT).

O cliente conecta-se ao *broker*. Ele pode assinar qualquer “tópico” de mensagem no *broker*. Posteriormente, o cliente publica as mensagens em um tópico, enviando a mensagem e o tópico ao *broker*. Em seguida, o *broker* encaminha a mensagem a todos os clientes que assinam esse tópico. No projeto desenvolvido, existem dois clientes, conforme visualizado na Figura 4.17. O primeiro cliente é responsável por realizar o controle da garra robótica antropomórfica, enquanto o segundo cliente é responsável pela interface gráfica e envio dos comandos para o primeiro cliente. Por exemplo, no cliente 1 o comando referente a fechar o dedo indicador é assinado no tópico “*controle/aciona_dedo/automatico/di*” e sua mensagem pode ser um valor “*true*” ou “*false*”. Enquanto as informações do sensor de força referente ao dedo indicador são publicados no tópico “*leitura_sensor_di*”, que corresponde ao valor analógico da medição do sensor FSR. Por sua vez, o cliente 2, assina o tópico “*leitura_sensor_di*” e publica no tópico “*controle/aciona_dedo/automatico/di*”.

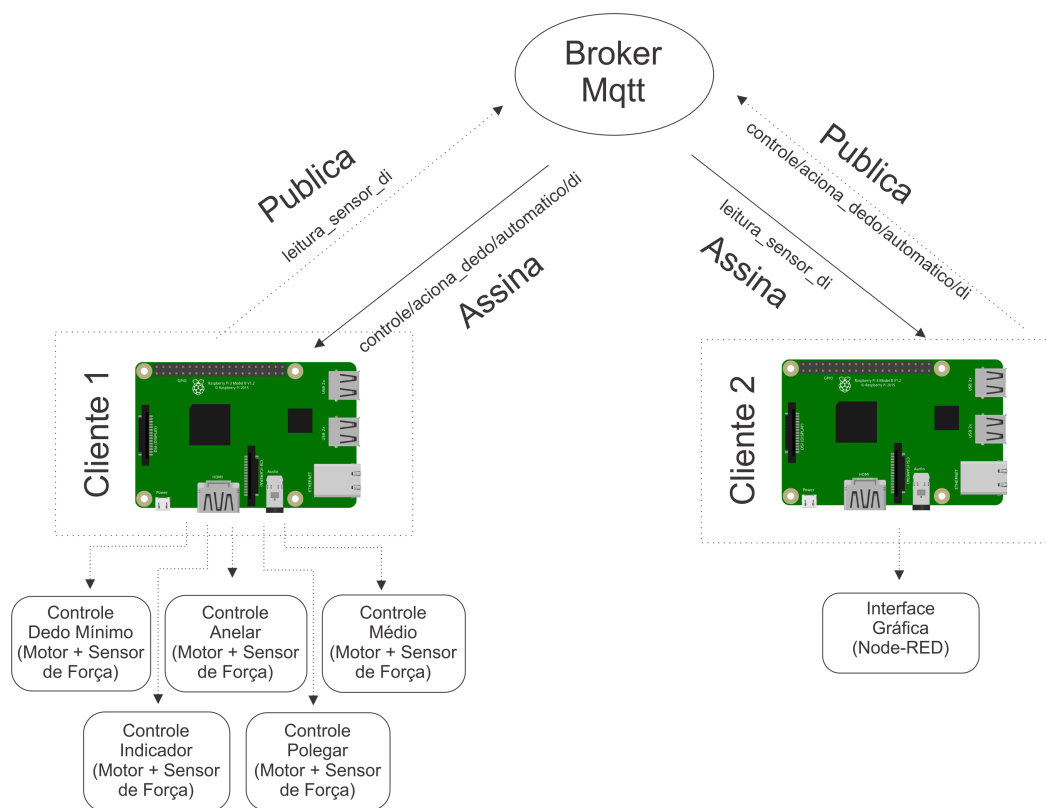


Figura 4.17: Comunicação através do Servidor MQTT Broker

4.4.6 Controle de Conformidade

O controle de conformidade foi desenvolvido para os 5 dedos, em uma única *Raspberry*. Sendo assim, conforme descrito no Capítulo 2, é utilizado controle por lógica *fuzzy* em alternativas ao controle clássico PID, devido à complexidade do sistema abordado. A fim de se obter o controle de conformidade desejado, é representado na Figura 4.18, através do sistema em malha fechada o processo a ser controlado. De forma geral, consiste em um *setpoint* de força, que entra no bloco de controle da lógica *fuzzy*, é convertido em variáveis linguísticas e a saída *defuzzificada* vai para o motor fazendo o dedo movimentar proporcionalmente a força aplicada. Como retroalimentação para possibilitar o controle em malha fechada é utilizado o sensor de força FSR. Essa mesma malha de controle se repete para os cinco dedos da garra robótica antropomórfica desenvolvida.

A variável do sistema a ser controlada será a força aplicada sobre o dedo para possibilitar a percepção tátil, que será estimada através do sensor FSR. Senso assim, iremos definir (f) conforme o sistema *fuzzy* para representar a força que pode ser *baixa*, *média* ou *alta*. Transformando os valores de entradas para termos linguísticos relacionado a lógica *fuzzy*, temos que:

$$F(f) = \text{baixa, média, alta}$$

Sendo assim, $F(f)$ representa uma função de decomposição para a variável de entrada f . Cada

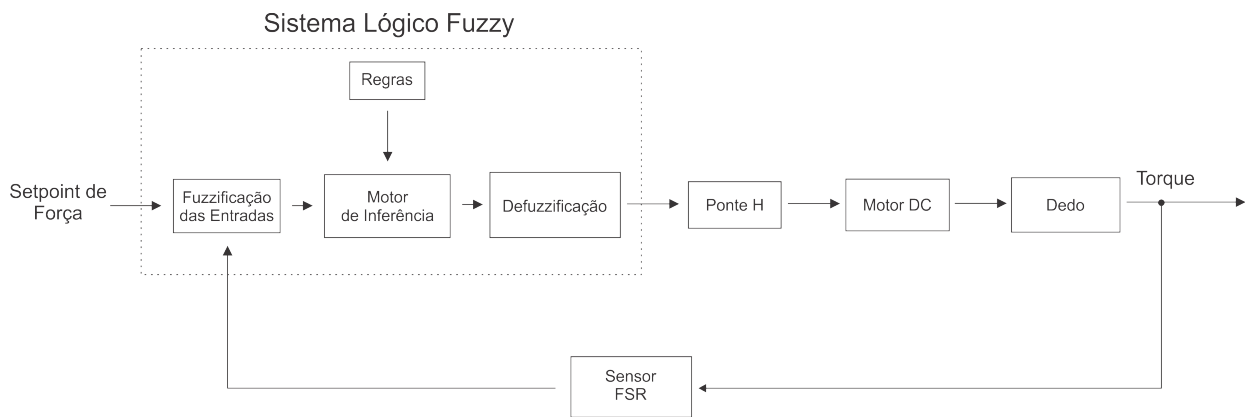


Figura 4.18: Diagrama de controle de conformidade, baseado em lógica *fuzzy*

membro desse conjunto de decomposição linguístico representa ou está associado a um intervalo de força numérico. Há outra variável *setpoint* de força será definido pelo usuário, sendo que nesse caso, será definido como um valor fixo de força de agarre para determinadas formas canônicas, sendo trabalhos futuros, trabalhado em cima de valores dinâmicos para otimizar o sistema de controle.

A implementação do controle de conformidade através da lógica *fuzzy* foi gerado através do *Fuzzy Logic Toolbox* do software *Matlab* e implementado através da biblioteca *scikit-fuzzy* na plataforma *Python*. Sendo assim, na Figura 4.19 é mostrado os valores associados as funções de pertinência do conjunto *fuzzy* para a entrada de força. Os valores relacionados a entrada do *setpoint* de força são os mesmos relacionados ao anterior. Já na Figura 4.20 é mostrado as funções de pertinências associadas ao conjunto *fuzzy* para saída do controle de força.

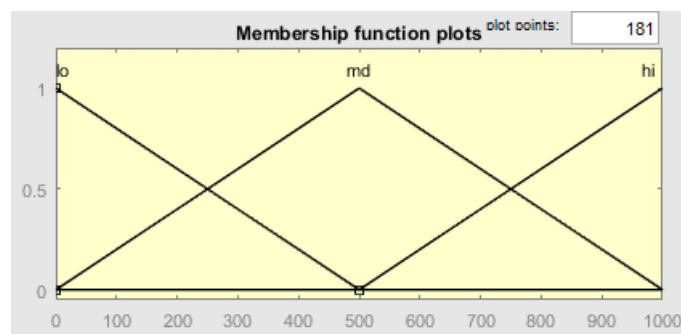


Figura 4.19: Funções de pertinência para a entrada de força

Ambas entradas e saída estão em função da leitura analógica do controlador indo de 0 a 1023. Tanto para as entradas e saídas, foram utilizados funções triangulares. O controle de força é dado em função da pressão lida pelo sensor FSR e o *setpoint* de pressão definido pelo usuário. As regras foram definidas com base em testes e referências bibliográficas a fim de obter o melhor controle de força, com um baixo custo computacional. Na Tabela 4.9 é mostrado as regras *fuzzy* definidas para o controle de força.

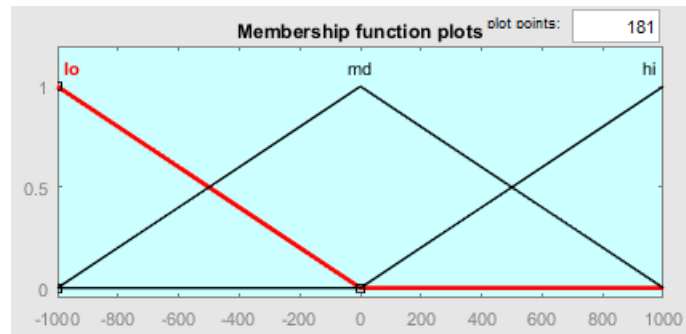


Figura 4.20: Funções de pertinência para a saída de controle de força

Tabela 4.9: Matriz de comando para variável de controle de força à partir da variável de medição de força e *setpoint* de força

Força	<i>Setpoint</i> de Força		
	Baixa	Média	Alta
Baixa	Não Muda	Alta	Alta
Média	Baixa	Não Muda	Alta
Alta	Baixa	Baixa	Não Muda

Ao todo existem 6 regras para acomodar todas as combinações de interseção da força e *setpoint* de força para poder controlar em termos linguísticos a força aplicada sobre o objeto. Sendo que a regra para não mudar, corresponde ao erro do sistema igual a 0, sendo a força do atuador igual à força requerida pelo sistema, obtendo-se assim o controle de conformidade. As regras definidas foram:

- Se a força é baixa e o *setpoint* de força é médio, então o controle de força é alto.
- Se a força é baixa e o *setpoint* de força é alto, então o controle de força é alto.
- Se a força é média e o *setpoint* de força é baixo, então o controle de força é baixo.
- Se a força é média e o *setpoint* de força é alto, então o controle de força é alto.
- Se a força é alta e o *setpoint* de força é baixo, o controle de força é baixo.
- Se a força é alta e o *setpoint* de força é médio, então o controle de força é baixo.

Desse modo é descrito o projeto eletroeletrônico da garra robótica antropomórfica desenvolvida. Sendo que o principal desafio do sistema eletroeletrônico foi a integração do controle de conformidade com sistema *fuzzy* na *Raspberry* para possibilitar a garra robótica a sentir os objetos manipulados a fim de atender aos requisitos dos clientes.

4.5 Custo de Produção

A garra robótica antropomórfica desenvolvida, por possuir sistemas mecanismos, elétricos e eletrônicos, os componentes foram pensados em serem de baixo custo e de fácil aquisição para possibilitar replicação, conforme os requisitos dos clientes. Sendo assim, todos os materiais necessários para desenvolvido foram adquiridos no mercado nacional. O custo de cada componente utilizado pode ser observado na Tabela 4.10, onde, também pode ser visualizado a quantidade de componentes utilizados para confecção da garra robótica antropomórfica. Para maiores informações em relação a garra robótica antropomórfica, como, componentes utilizados, peças em formato de impressão 3D e algoritmo de controle pode ser encontrado no site www.arturvitorio.weebly.com.

Tabela 4.10: Custo dos componentes para confecção da garra robótica antropomórfica

Item	Descrição	R\$	Quant	Total
1	Fonte de Alimentação 10A	R\$ 34,90	1	R\$ 34,90
2	Kit-Raspberry Pi	R\$ 349,99	2	R\$ 699,98
3	Driver L298n	R\$ 10,50	3	R\$ 31,50
4	Conversor Analógico-Digal MCP3008	R\$ 19,90	1	R\$ 19,90
5	Conversor Digital DC-DC 35-5V	R\$ 19,90	1	R\$ 19,90
6	Sensor de força FSR	R\$ 92,00	5	R\$ 460,00
7	Micro Motores DC	R\$ 95,00	4	R\$ 380,00
8	Micro Motor DC	R\$ 35,00	1	R\$ 35,00
9	Servo Motor	R\$ 7,99	1	R\$ 7,99
10	Kit 10 Rolamentos 623 ZZ 3x10x4mm	R\$ 23,90	1	R\$ 23,90
11	Kit 10 Rolamentos V623zz com Sulco	R\$ 89,90	1	R\$ 89,90
12	Base de Apoio MDF	R\$ 40,00	1	R\$ 40,00
13	Kit 100 Parafusos 4x2mm	R\$ 57,00	1	R\$ 57,00
14	Kit 100 Parafusos 6x3mm	R\$ 60,00	1	R\$ 60,00
15	Resistores 320Ohm	R\$ 1,00	5	R\$5,00
16	Kit conectores M-M e F-F	R\$ 100,00	1	R\$ 100,00
17	Rolo de Cabo 1mm	R\$ 32,00	1	R\$ 32,00
18	Protoboard	R\$ 15,00	1	R\$ 15,00
		Total		R\$ 2.111,97

Desta forma o projeto de cada um dos domínios que formão a garra robótica, desde a abordagem de desenvolvimento de produto para confecção da garra robótica e modelo V para descrever os sistemas mecânicos, eletroeletrônico e software do projeto são descritos. Como resultado do domínio do projeto mecânico, obteve-se a resolução da cinemática direta e inversa fundamental para o controle da garra robótica. Na modelagem CAD onde foi gerada uma visão preliminar

de cada sistema junto com a especificação dos componentes e planos de projeto para as partes fabricadas. O projeto eletroeletrônico deixou como resultados a especificação dos dispositivos eletrônicos necessários e o diagrama de conexão geral. Por último, no domínio do projeto computacional foram descritos os módulos de software utilizados, a criação da interface gráfica e sistema de controle de conformidade. O algoritmo de controle desenvolvido pode ser verificado no Apêndice A e também disponível no site www.arturvitorio.weebly.com.br. Já no Apêndice B é listado todo o conjunto de materiais/peças utilizados no projeto.

Capítulo 5

Testes Funcionais e Experimentos da Garra Robótica Antropomórfica

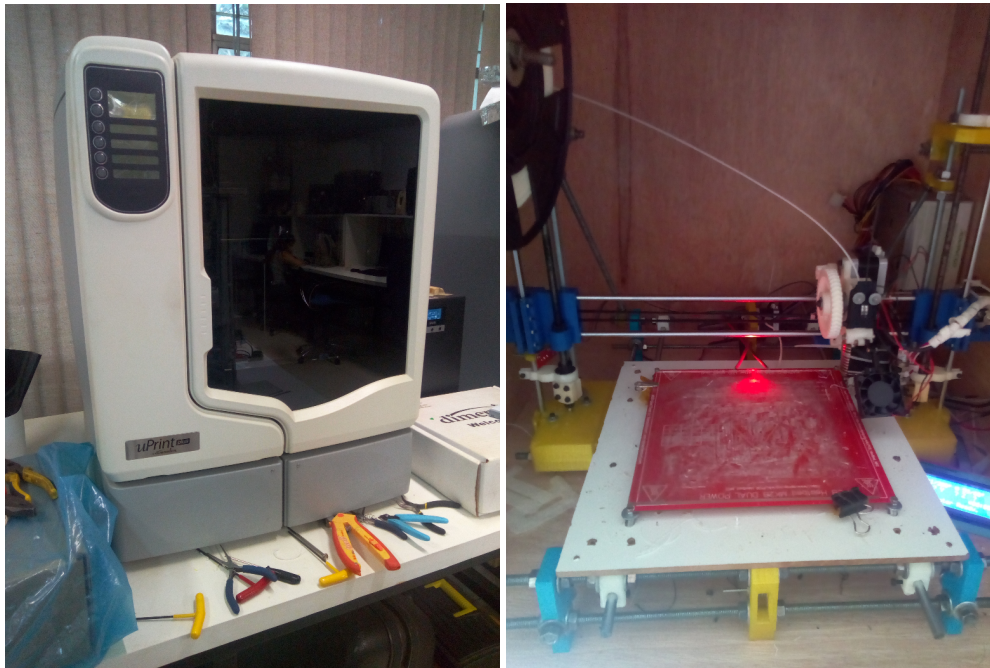
A fim de verificar o efetivo desenvolvimento da garra robótica, será demonstrado os materiais que são utilizados para desenvolver a garra, tais como as impressoras 3D utilizadas para prototipação do projeto. Outrossim, é demonstrado os testes funcionais e experimentos da garra baseado em métodos analíticos e experimentais com o intuito de comprovar o desenvolvimento da garra robótica antropomórfica.

5.1 Materiais Empregados na Construção da Garra Robótica Antropomórfica

Devido ao tempo limitado e restrições do projeto, tais como materiais e experiência do projetista (autor), foi desenvolvido apenas um modelo de protótipo utilizando impressoras 3D com material plástico ABS (*Acrilonitrilo Butadieno Estireno*). A vantagem do plástico ABS são suas propriedades mecânicas de maior resistência comparada com outros materiais plásticos da mesma gama. Foram utilizadas duas impressoras 3D, uma modelo profissional uPrint [115], conforme Figura 5.1a, para desenvolver as peças que necessitam de um esforço mecânico maior, por exemplo, a base de apoio dos motores DC e extensores de nylon. E uma segunda impressora 3D, modelo RepRap desenvolvida pelo autor, para prototipação das peças que exigiam menores esforços mecânicos, por exemplo, os elos de movimentação, palma e dorso da garra robótica, conforme Figura 5.1b.

A garra robótica é desenvolvida baseada nos materiais especificados através dos princípios de solução estabelecidos na Tabela 4.4 e selecionado através da Tabela 4.6. A justificativa para utilização dos materiais empregados encontram-se na Seção 4.3.

Para possibilitar acomodar a garra robótica desenvolvida e a eletroeletrônica do projeto, foi



(a)

(b)

Figura 5.1: Impressoras 3D utilizadas para construção da garra robótica. Figura (a) impressora 3D Profissional uPrint e Figura (b) impressora 3D RepRap

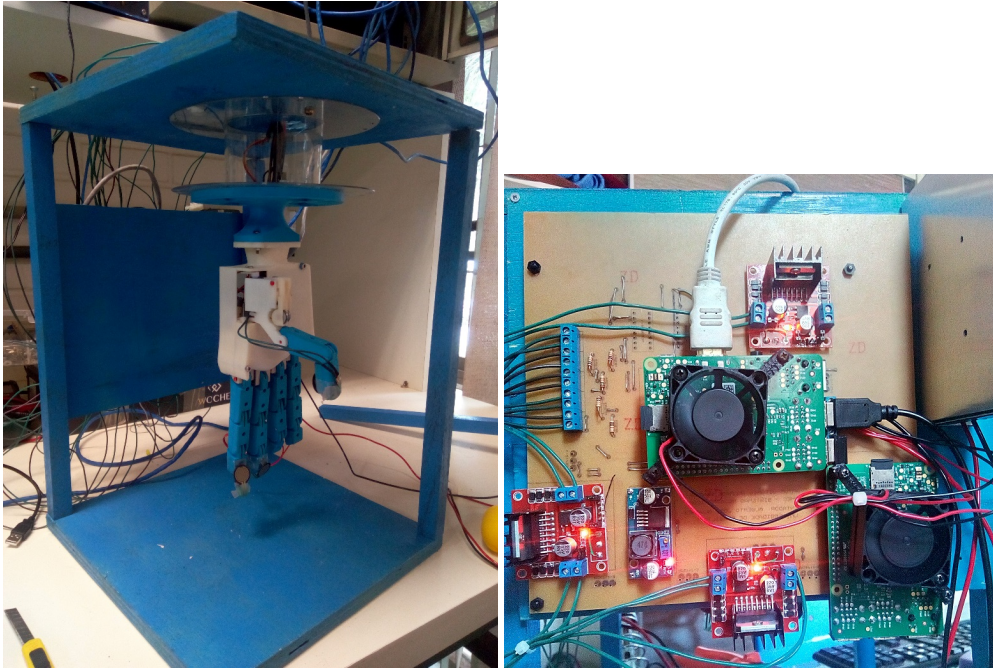
desenvolvido uma estrutura de suporte em madeira, conforme apresentado na Figura 5.2. Além de dar o suporte para a garra robótica, “simula” o elo de acoplamento de um manipulador robótico.

A placa eletrônica do projeto apresentado na Subseção 4.4.4, foi desenvolvida a partir de uma máquina CNC router 3018 para fabricação de placas de circuito aberto, modelo faça-você-mesmo, apresentado na Figura 5.3. Todo o projeto eletrônico da placa eletrônica, como, código G, para usinagem e furação da placa estão disponíveis no site www.arturvitorio.weebly.com.

5.2 Análise da Garra Robótica baseado em Métodos Analíticos

O mecanismo tem como restrição de movimentos para os elos considerados os seguintes valores de ângulos para os dedos mínimo, anelar, médio, indicador e polegar conforme Tabelas 5.1 e 5.2. Caso esses ângulos não sejam respeitados o mecanismo corre risco de rompimento conforme as simulações realizadas. Todas as simulações e peças desenvolvidas a serem impressas em 3D, podem ser encontradas no site www.arturvitorio.weebly.com.

A partir das restrições máximas dos ângulos, o espaço de trabalho da garra robótica pode ser



(a)

(b)

Figura 5.2: Estrutura em MDF com a garra robótica antropomórfica montada. Figura (a) a garra montada no suporte simulando o acoplamento de um manipulador e Figura (b) a eletrônica fixada para possibilitar o controle da garra robótica

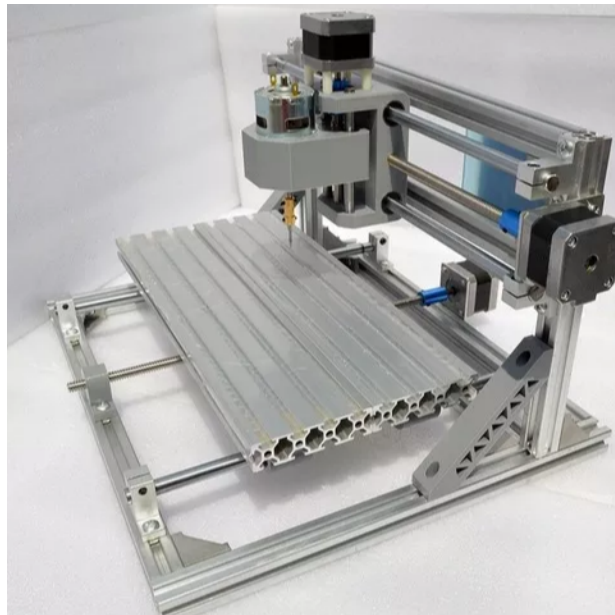


Figura 5.3: Modelo da CNC router 3018 utilizada para fabricação da placa eletrônica

determinada, conforme demonstrado na Figura 5.4. Observa-se que a movimentação do dedo polegar é movimentado sem a restrição de oposição com os outros dedos, uma vez que, o objetivo é verificar o espaço de trabalho total do mecanismo. Porém, caso o dedo polegar movimente junto

Tabela 5.1: Restrições de ângulo máximo para os elos dos dedos mínimo, anelar, médio e indicador

Junta	Comprimento	Ângulo Máximo
MCP	39.37 mm	0 à 89°
PIP	36.20 mm	0 à 84°
DIP	26.71 mm	0 à 70°

Tabela 5.2: Restrições de ângulo máximo para os elos do polegar

Junta	Comprimento	Ângulo Máximo
RC	36.61 mm	0 à 114°
PIP	39.19 mm	0 à 52°
DIP	35.68 mm	77°

com o dedo indicador, por exemplo, deve-se observar que não pode ser ao mesmo tempo, pois, os mesmos irão se encontrar em um ângulo menor que o máximo a ser movimentado, igual ocorre na mão humana.

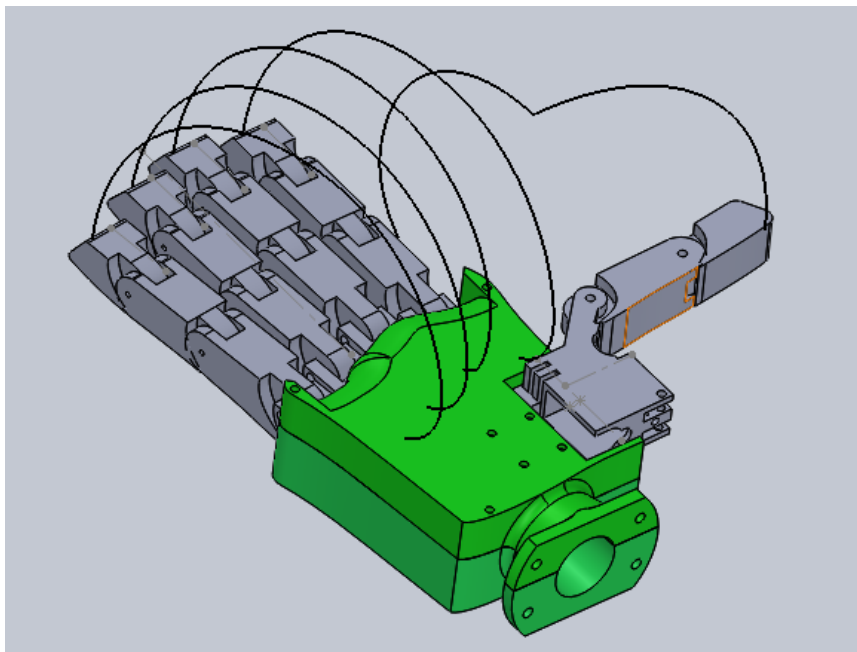


Figura 5.4: Espaço de trabalho considerado da garra robótica antropomórfica, modelo simplificado para simulação

A simulação para comprovação dos valores calculados e parâmetros geométricos foi realizado com o software *SolidWorks*, através do *Toolbox SolidWorks Motion*. A partir da representação de

barras do mecanismo, foi feita a comprovação da cinemática, bem como verificado a área de trabalho. A simplificação por barras do mecanismo serve para possibilitar a determinação dos parâmetros de engenharia, uma vez que, o modelo 3D original possui muitos componentes.

A partir dos cálculos desenvolvidos na Equação 4.8, e utilizando os valores de projeto, através do *Matlab* e algoritmo desenvolvido no Apêndice C, foi encontrado os valores da posição final dos dedos para x, y e z no plano cartesiano para os dedos mínimo, anelar, médio e indicador, conforme a Tabela 5.3

Tabela 5.3: Valores calculados através do *Matlab* para posição final dos dedos mínimo, anelar, médio e indicador

MCP, DIP, PIP	X	Y	Z
25, 30, 60	46.31 mm	71.58 mm	0 mm
40, 50, 70	5.68 mm	71.86 mm	0 mm
80, 80, 75	-42.78 mm	30.31 mm	0 mm

Através do *SoldiWorks* foi possível realizar a simulação e medição dos valores calculados, conforme apresentado na Tabela 5.4. Apenas será mostrado as simulações para os ângulos de 25, 30 e 60°. As outras simulações correspondem ao mesmo padrão demonstrado, assim como os outros dedos. Vale lembrar que a componente do eixo Z é 0 pois o mecanismo não se move nessa direção.

Tabela 5.4: Valores obtidos através da simulação do *SolidWorks* para posição final dos dedos

MCP, DIP, PIP	X	Y	Z
25, 30, 60	48.15 mm	71.07 mm	0 mm
40, 50, 70	6.50 mm	72.45 mm	0 mm
80, 80, 70	-45.42 mm	32.08 mm	0 mm

Na Figura 5.5 é mostrado o deslocamento do eixo X, através de uma simulação de 15s. Ele sai do ponto inicial considerado, sendo o comprimento total do dedo e chega até o ponto final para os ângulos movimentados de 25, 30 e 60°. Pode-se observar que entre o valor calculado na simulação e a posição da garra robótica para o dedo considerado possui um erro de 1,83 mm no eixo X e 0,51 mm no eixo Y na posição das falanges consideradas. Esse erro deve-se ao fato da posição inicial do mecanismo, uma vez que a posição inicial não é exata. Há necessidade de travar o mecanismo em uma posição de referência para fazer a medição. Considerando as folgas e desgaste do mecanismo, nem sempre é possível obter a referência 0.

Na Figura 5.6 é mostrado o deslocamento do eixo Y, através de uma simulação de 15s. Ele sai do ponto inicial considerado, sendo o comprimento total do dedo e chega até o ponto final para

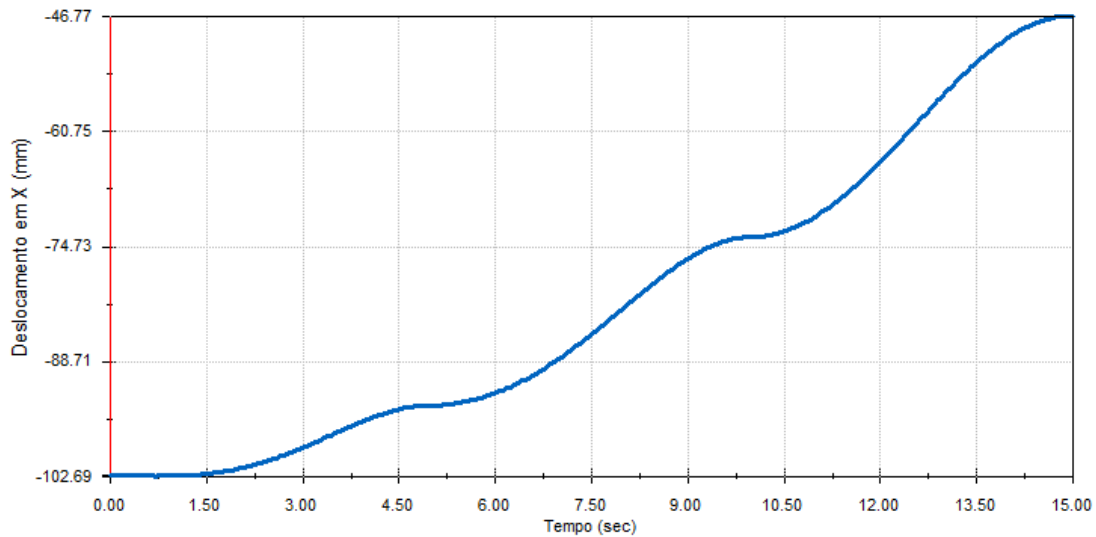


Figura 5.5: Deslocamento do Eixo X no espaço tridimensional

os ângulos movimentados de 25, 30 e 60°. Vale observar que o mecanismo sai da posição -2.54 mm, pois, é complicado posicionar exatamente na referência 0 do mecanismo para possibilitar a movimentação, conforme descrito anteriormente.

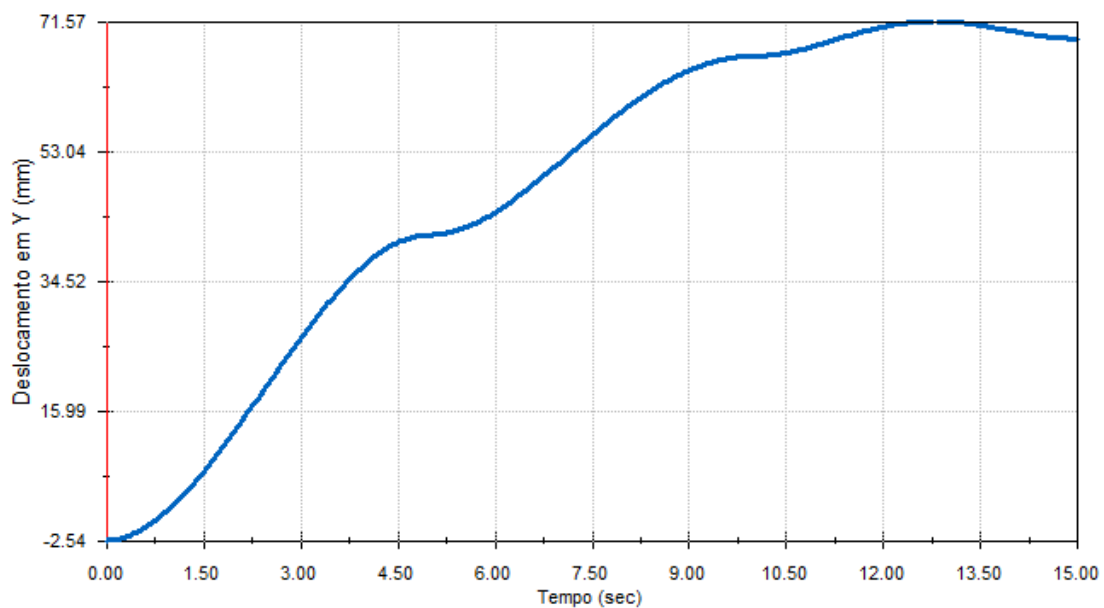


Figura 5.6: Deslocamento do eixo Y no espaço tridimensional

Já para o polegar, a partir dos cálculos desenvolvidos na Equação 4.12, e utilizando os valores de projeto, através do *Matlab* foi encontrado os valores da posição final do polegar para x, y e z no plano cartesiano, através da Tabela 5.5.

Através do *SoldiWorks* foi possível realizar a simulação e medição dos valores calculados,

Tabela 5.5: Valores calculados através do *Matlab* para posição final do polegar

MCP, DIP, RC	X	Y	Z
25, 30, 60	61.14 mm	28.51 mm	76.07 mm
40, 50, 70	34.59 mm	29.03 mm	82.96 mm
80, 80, 75	2.72 mm	15.43 mm	79.64 mm

conforme apresentado na Tabela 5.6. Apenas será mostrado as simulações para os ângulos de 25, 30 e 60°. As outras simulações correspondem ao mesmo padrão demonstrado. Vale lembrar que a componente do eixo Z, diferente dos outros dedos, agora possui uma dimensão associada, pois, a posição dos elos e juntas do polegar são diferentes.

Tabela 5.6: Valores obtidos através da simulação do *SolidWorks* para posição final do polegar

MCP, DIP, RC	X	Y	Z
25, 30, 60	60.69 mm	28.80 mm	76.36 mm
40, 50, 70	34.20 mm	29.09 mm	83.12 mm
80, 80, 70	2.45 mm	14.44 mm	77.50 mm

Na Figura 5.7 é mostrado o deslocamento do eixo X, através de uma simulação de 15s. Ele sai do ponto inicial considerado, sendo o comprimento total do dedo e chega até o ponto final para os ângulos movimentados de 25, 30 e 60°, observa-se que a componente X possui um erro de 0,45 mm entre o valor calculado e o valor medido.

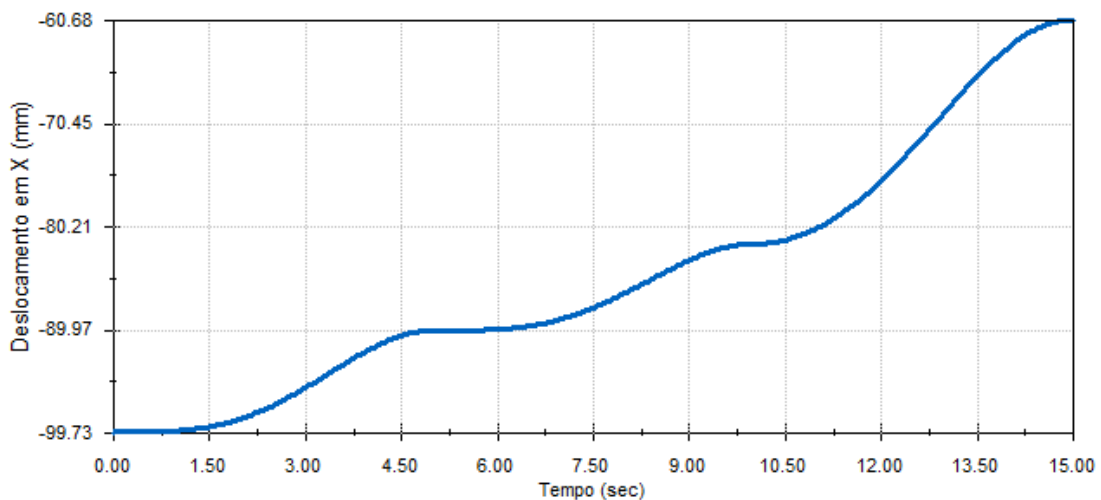


Figura 5.7: Deslocamento do Eixo X no espaço tridimensional para o polegar

Na Figura 5.8 é mostrado o deslocamento do eixo Y, através de uma simulação de 15s, o

mesmo sai do ponto inicial considerado, sendo o comprimento total do dedo e chega até o ponto final para os ângulos movimentados de 25, 30 e 60°, observa-se que o erro para essa componente é de 0,28 mm.

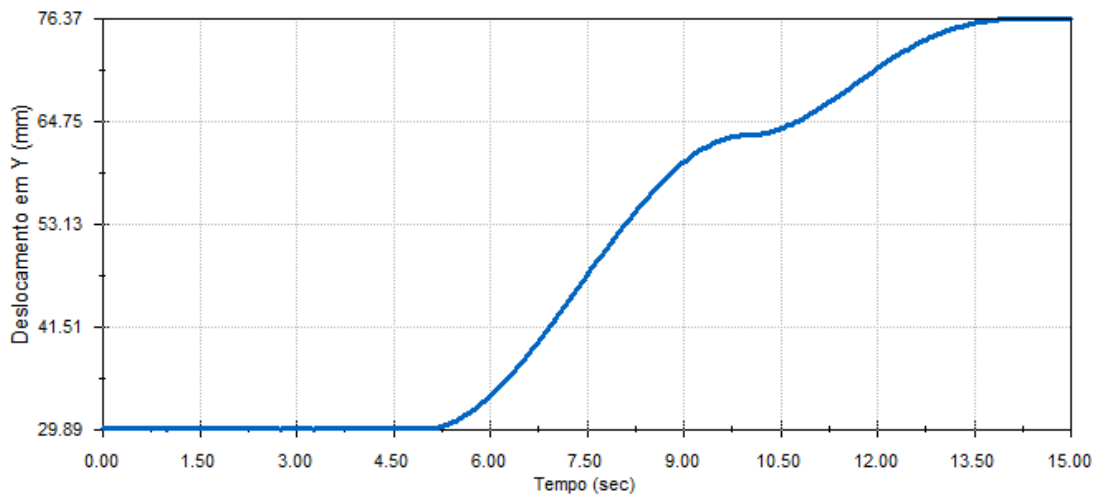


Figura 5.8: Deslocamento do eixo Y no espaço tridimensional para o polegar

Já na Figura 5.9 é mostrado o deslocamento do eixo Z, através de uma simulação de 15s. O polegar sai do ponto inicial considerado, sendo o comprimento total do dedo e chega até o ponto final para os ângulos movimentados de 25, 30 e 60°, possuindo um erro de 0.29 mm. Os erros aqui considerados para o polegar deve-se ao mesmo motivo já descrito para os outros dedos.

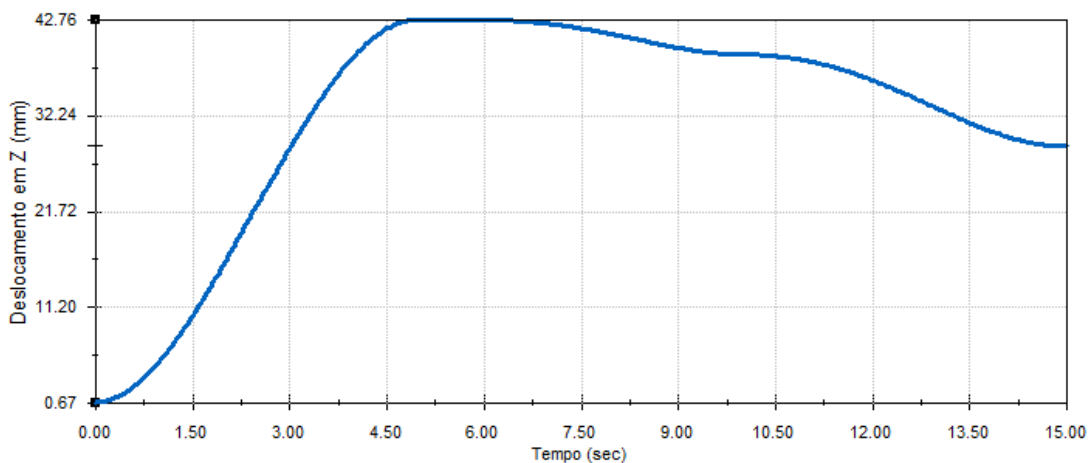
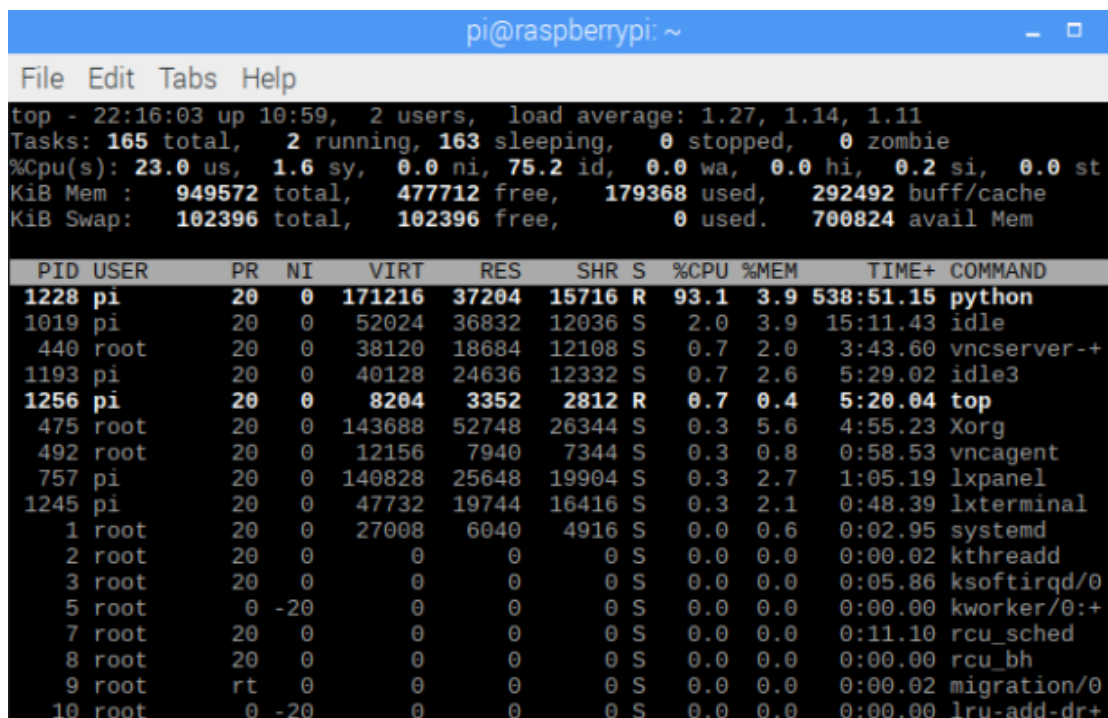


Figura 5.9: Deslocamento do eixo Z no espaço tridimensional para o polegar

5.3 Análise da Garra Robótica baseado em Métodos Experimentais

Com o controle de percepção tátil desenvolvido, foi possível executar em uma *Raspberry* para os cinco dedos da garra robótica desenvolvida, devido à implementação do controle *fuzzy* que possui baixo custo computacional, ao mesmo tempo, boa resposta, conforme descrito no Capítulo 2. O tempo de resposta para os cinco dedos ficou em cerca de 300 ms, sendo que o consumo da CPU chegou a 95%, conforme Figura 5.10.



```
pi@raspberrypi: ~
File Edit Tabs Help
top - 22:16:03 up 10:59, 2 users, load average: 1.27, 1.14, 1.11
Tasks: 165 total, 2 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 23.0 us, 1.6 sy, 0.0 ni, 75.2 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 949572 total, 477712 free, 179368 used, 292492 buff/cache
KiB Swap: 102396 total, 102396 free, 0 used, 700824 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 1228 pi        20   0  171216  37204 15716 R  93.1   3.9  538:51.15 python
 1019 pi        20   0  52024  36832 12036 S   2.0   3.9  15:11.43 idle
   440 root      20   0   38120  18684 12108 S   0.7   2.0   3:43.60 vncserver-+
  1193 pi        20   0   40128  24636 12332 S   0.7   2.6   5:29.02 idle3
  1256 pi        20   0   8204   3352   2812 R   0.7   0.4   5:20.04 top
   475 root      20   0  143688  52748 26344 S   0.3   5.6   4:55.23 Xorg
   492 root      20   0   12156   7940  7344 S   0.3   0.8   0:58.53 ksoftirqd/0
   757 pi        20   0  140828  25648 19904 S   0.3   2.7   1:05.19 lxpanel
  1245 pi        20   0   47732  19744 16416 S   0.3   2.1   0:48.39 lxterminal
     1 root      20   0   27008   6040  4916 S   0.0   0.6   0:02.95 systemd
     2 root      20   0         0         0     0 S   0.0   0.0   0:00.02 kthreadd
     3 root      20   0         0         0     0 S   0.0   0.0   0:05.86 ksoftirqd/0
     5 root      0 -20         0         0     0 S   0.0   0.0   0:00.00 kworker/0:+
     7 root      20   0         0         0     0 S   0.0   0.0   0:11.10 rcu_sched
     8 root      20   0         0         0     0 S   0.0   0.0   0:00.00 rcu_bh
     9 root      rt   0         0         0     0 S   0.0   0.0   0:00.02 migration/0
    10 root      0 -20         0         0     0 S   0.0   0.0   0:00.00 lru-add-dr+
```

Figura 5.10: Custo computacional para o sistema de controle de conformidade desenvolvido.

Através da implementação computacional, foi realizado testes de controle a fim de validar o sistema desenvolvido. Será demonstrado os testes para um dos dedos da garra robótica (médio), sendo que os outros dedos possuem o mesmo comportamento. Foi utilizado um objeto maleável (balão), para testar a conformidade do sistema, a fim de verificar a atuação do controle de força. Os testes realizados e uma melhor visualização da descrição podem ser encontrados no site www.arturvitorio.weebly.com.br. Foi realizada 5 tentativas para cada faixa de atuação, sendo a que apresentou melhor resultado foi o *setpoint* de 2N, 4N e 6N apresentados nas Figuras 5.11, 5.12, 5.14, respectivamente. Devido à dinâmica do mecanismo, os atritos, posição do objeto de agarre, interação dedo-objeto, entre outros, observa-se uma variação de resposta para o mecanismo.

Para o *setpoint* de 1N apresentado na Figura 5.13 o sistema tem uma instabilidade de controle, pois, devido às variáveis de pertinências utilizadas, o sistema torna-se robusto, sendo que, para

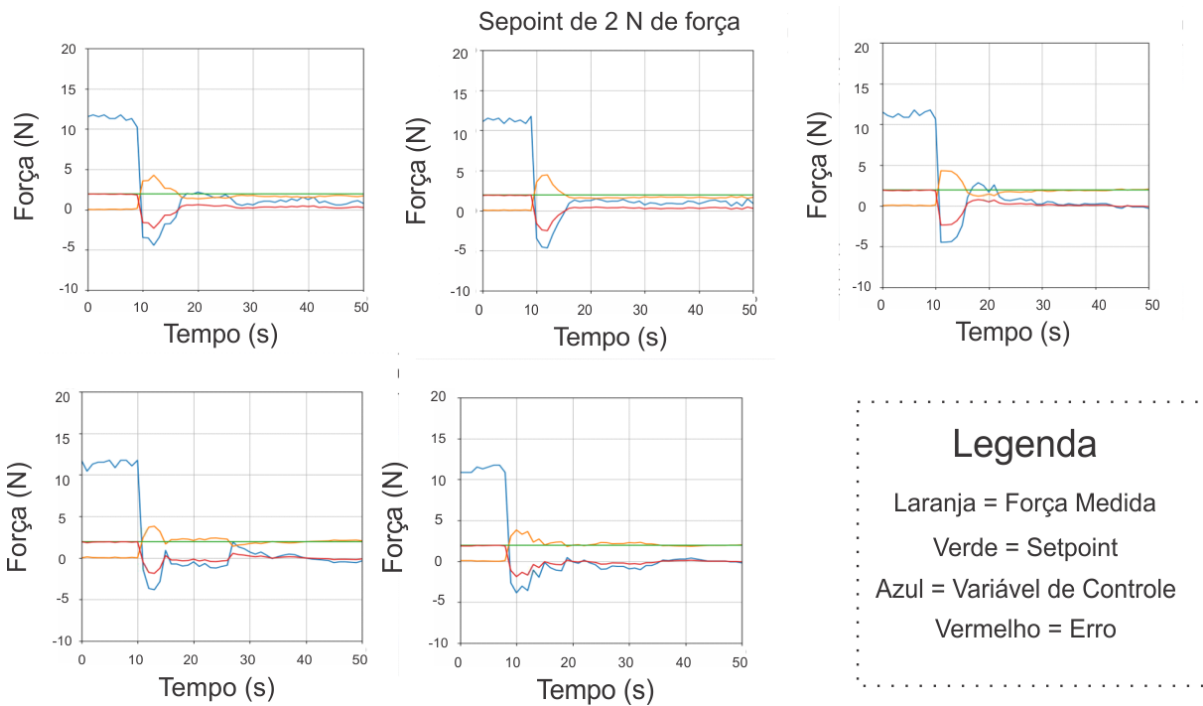


Figura 5.11: Resposta do sistema de controle para um *setpoint* de 2N

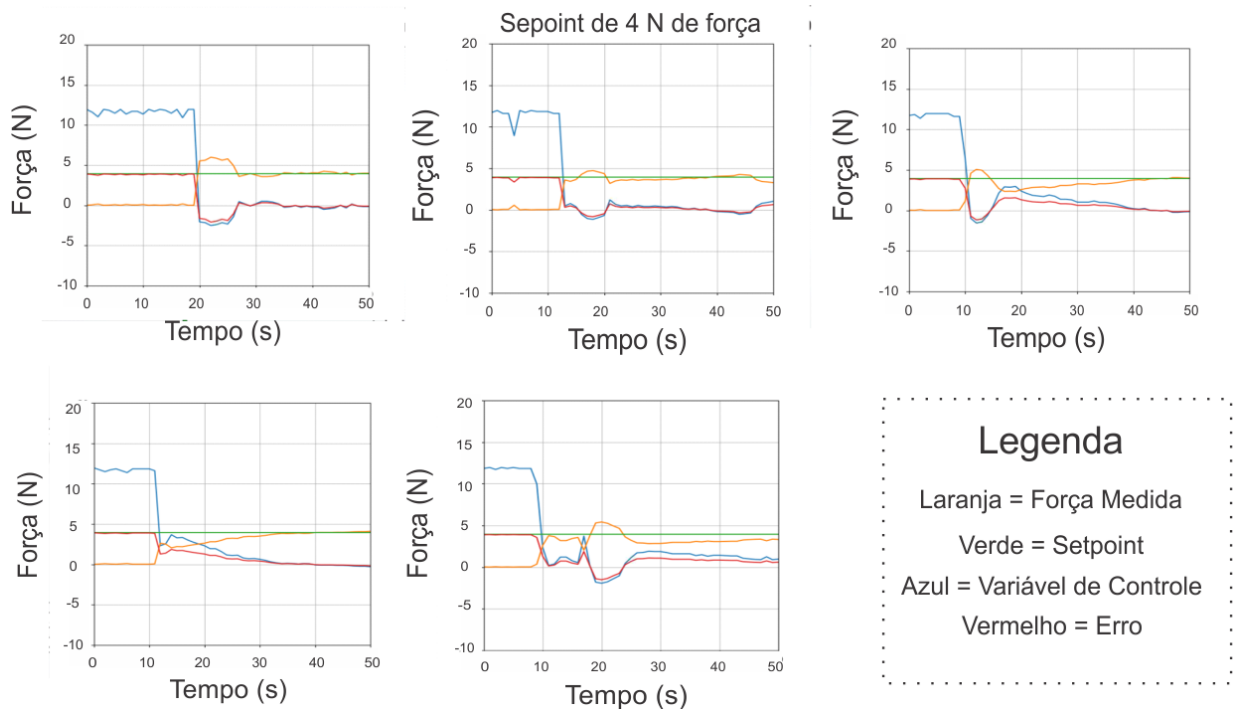


Figura 5.12: Resposta do sistema de controle para um *setpoint* de 4N

uma maior sensibilidade e melhor controle em baixos *setpoints* deve-se incluir mais variáveis de pertinências ao sistema. Observa-se, porém, que para a quarta e quinta tentativa, o sistema comportou-se como esperado. Comprovando a eficiência do mesmo.

Enquanto para os *setpoints* de 8 e 10N+, o sistema não conseguiu chegar no *setpoint* definido,

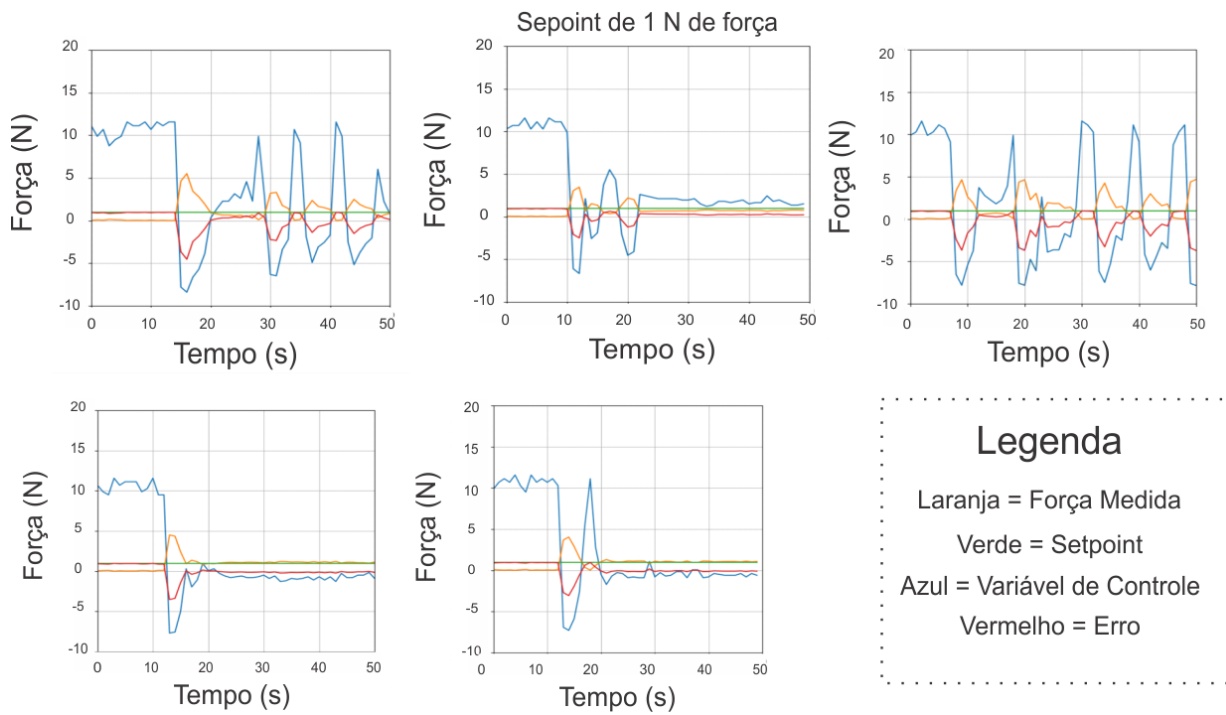


Figura 5.13: Resposta do sistema de controle para um *setpoint* de 1N

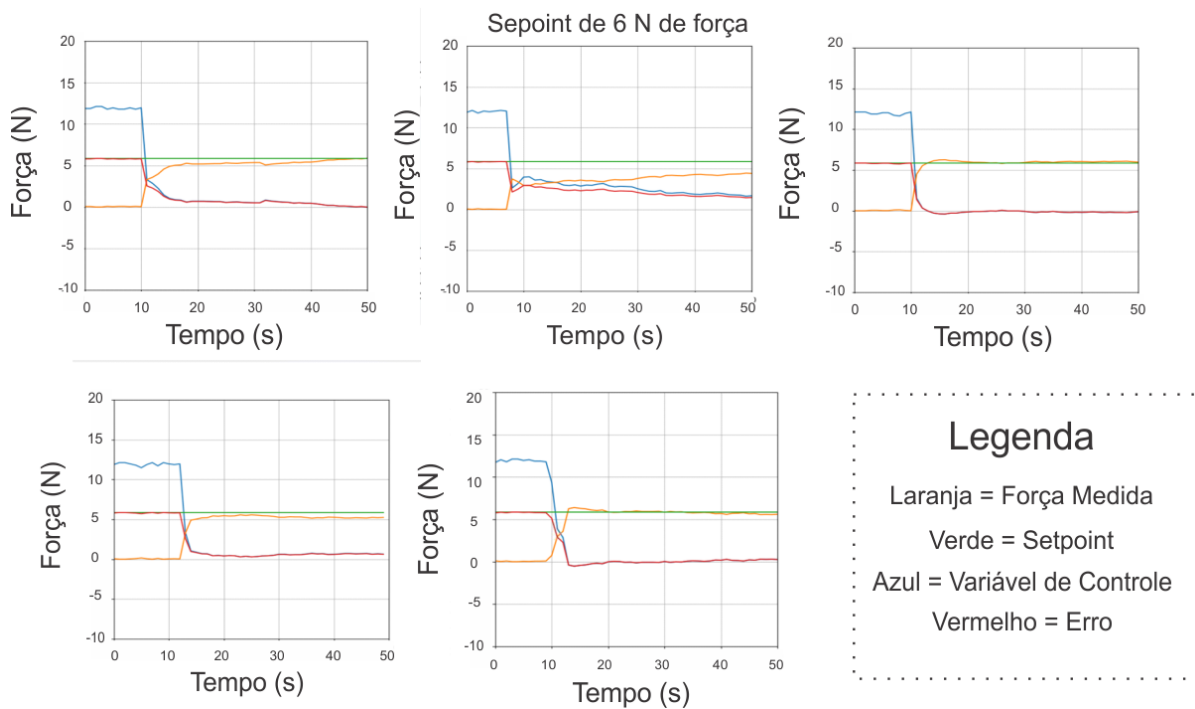


Figura 5.14: Resposta do sistema de controle para um *setpoint* de 6N

pois, devido às funções de pertinências utilizados, a saída do controlador não tem boas repostas, impedindo o motor de executar força máxima.

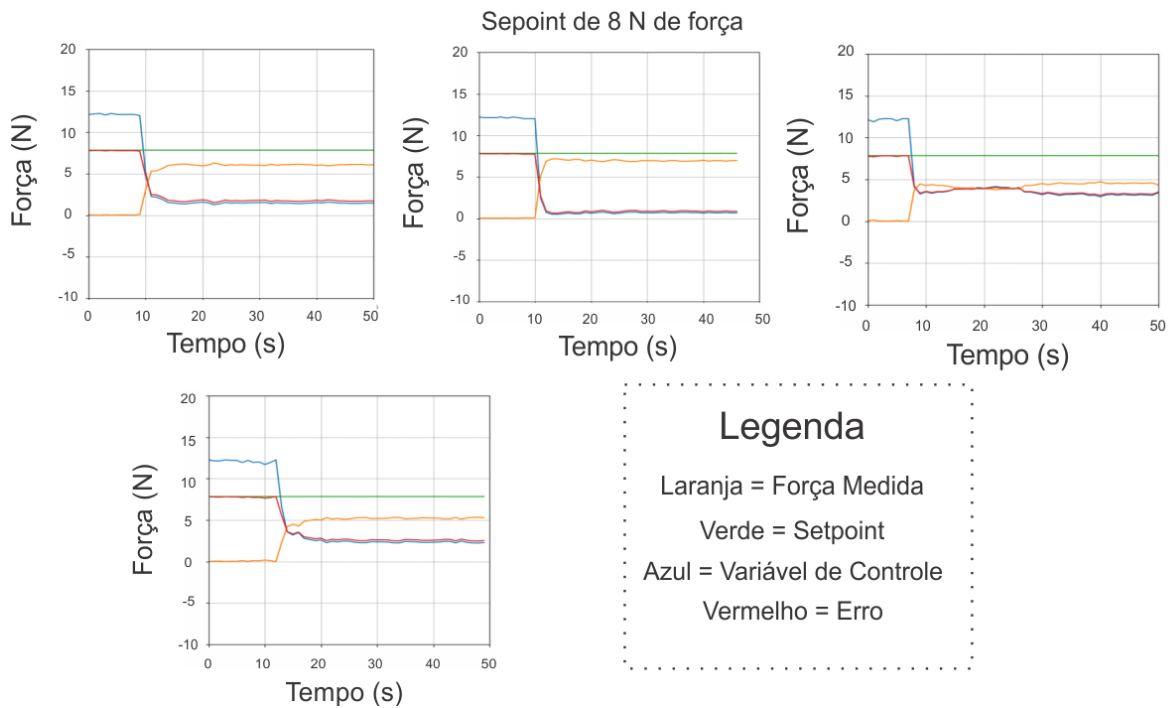


Figura 5.15: Resposta do sistema de controle para um *setpoint* de 8N

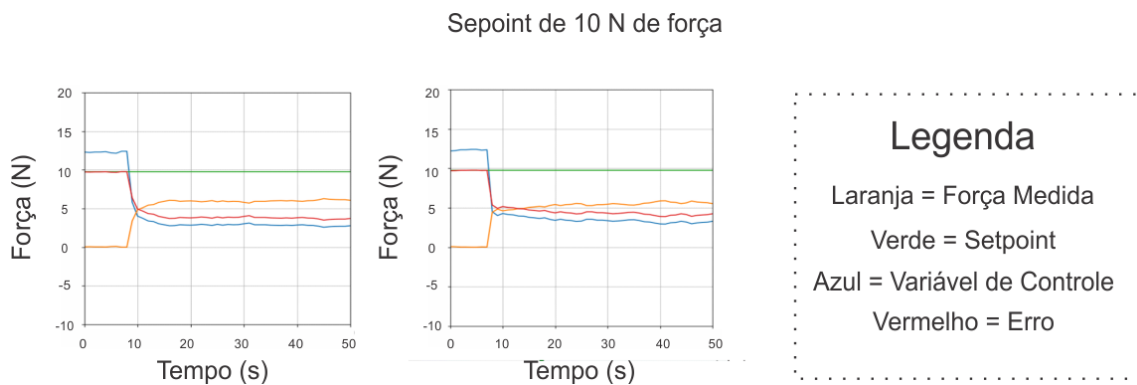


Figura 5.16: Resposta do sistema de controle para um *setpoint* de 10N

5.3.1 Manipulação de Objetos

Os testes deram-se tanto na forma manual como automática. Na forma manual o sistema ignora o controle de conformidade, possibilitando operar o sistema abrindo e fechando os dedos individualmente. Enquanto na forma automática o sistema habilita o sistema de controle de conformidade conforme descrito no diagrama de caso de uso da Figura 4.15.

Os objetos considerados para teste em controle manual foram: uma fita isolante, um alicate universal, uma caneta, um multímetro, uma *protoboard*, uma chave de boca combinada n° 15, um tubo de cola bastão, um objeto com formato elíptico impresso em 3D e chave *allen*. Esses objetos foram considerados a fim de testar as possibilidades de agarres de formas canônicas e a destreza da garra robótica, conforme visualizado na Figura 5.17.



Figura 5.17: Diferentes objetos considerados para manipulação. Sendo da esquerda para direita: fita isolante, alicate universal, caneta, multímetro, *protoboard*, chave combinada 15, tubo de cola bastão, objeto com formato elíptico 3D, chave *allen*

Os objetos considerados possuem pesos diferentes a fim de comprovar a estabilidade de agarre. Cabe ressaltar que devido aos componentes empregados no desenvolvimento do projeto, os objetos considerados possuem pesos limitados a fim de preservar o mecanismo desenvolvido. Os pesos dos objetos considerados podem ser visualizados na Tabela 5.7, onde os mesmos vão desde 9.95g até 437g.

Tabela 5.7: Tabela com diferentes tipos de objetos considerados para manipulação.

Objetos		
Item	Item	Peso
1	Caneta Esferográfica	9.95g
2	Peça Elíptica 3D	12.31g
3	<i>Protoboard</i>	37.97g
4	Fita Isolante	60.34g
5	Chave <i>Allen</i>	72.40g
6	Tubo de Cola Bastão	73.51g
7	Chave Combinada 15	75.34g
8	Multímetro	120.138g
9	Alicate Universal	437g

Na Figura 5.18 pode-se observar os objetos considerados para manipulação. A manipulação da caneta, por possuir uma espessura mínima, além de possuir formato cilindro, devido à forma de manipulação da garra robótica e a área de trabalho entre os dedos polegar e indicador, torna-se um complicador de realizar o agarre da mesma, assim como a chave *allen*. Porém, caso a manipulação da caneta e a chave *allen* seja realizada com os dedos mínimo, anelar, médio e indicador a manipulação torna-se satisfatória. Pode-se observar na Figura 5.18 que são realizados diferentes

formas de manipulação a fim de validar os agarres para a garra robótica. Cabe ressaltar, que para trabalhos futuros, é necessário realizar uma forma de validação utilizados métodos estatísticos a fim de validar os agarres desenvolvidos.



Figura 5.18: Objetos considerados para manipulação

Já para o teste do controle automático, foi utilizado um balão para simulação de objetos com diferentes maleabilidades a fim de comprovar o sistema de controle de conformidade aplicado. Os testes em vídeo demonstrando a movimentação do dedo considerado pode ser encontrado no site www.arturvitorio.weebly.com. Nas Figura 5.19 e 5.20 é demonstrado a diferença no agarre do balão para o ajuste de conformidade adotado. Para a primeira figura é ajustado o *setpoint* para 2N de força, enquanto para a segunda figura é ajusta o *setpoint* para 7.82N, onde pode-se observar a deformação do balão, verificando-se assim um maior aperto. Nas figuras é demonstrado também o comportamento do ajuste de força para o agarre considerado.

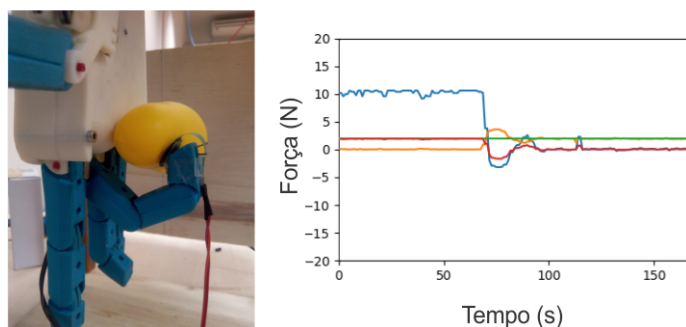


Figura 5.19: Ajuste do controle de conformidade para 2N sem deformação do balão

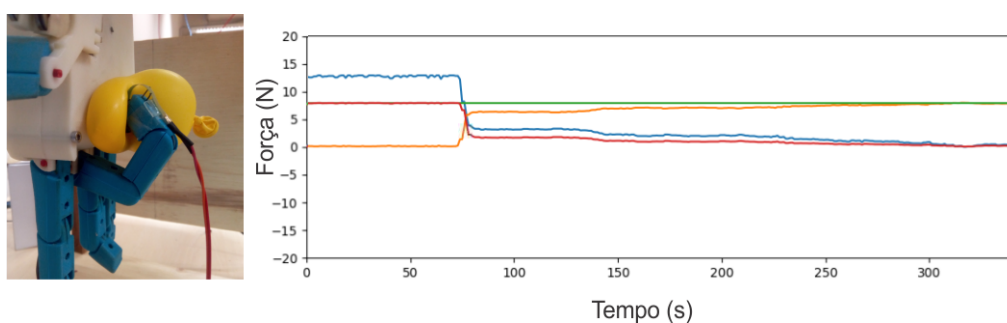


Figura 5.20: Ajuste do controle de conformidade para 7.82N com deformação do balão

No site também é possível encontrar um conjunto de agarres aumentando gradativamente o *setpoint* de força a fim de realizar testes em diferentes faixas de operação. Cabe ressaltar que para o controle de força adotado, caso o sensor não entre totalmente em contato com o objeto a ser manipulado, ocasiona erros na medição e consequentemente erros no agarre. A título de exemplo, é possível observar na Figura 5.21, o comportamento do agarre até o ajuste do objeto para estabilidade do agarre.

5.3.2 Formas Canônicas de Agarre

As formas canônicas de agarre deu-se de forma a imitar a manipulação de objetos conhecidos, por exemplo, uma garra *pet* de água mineral de 500g vazia. Para possibilitar o agarre, a garra deve fechar além do polegar envolvendo a garrafa, os outros 4 dedos da mão para que o agarre

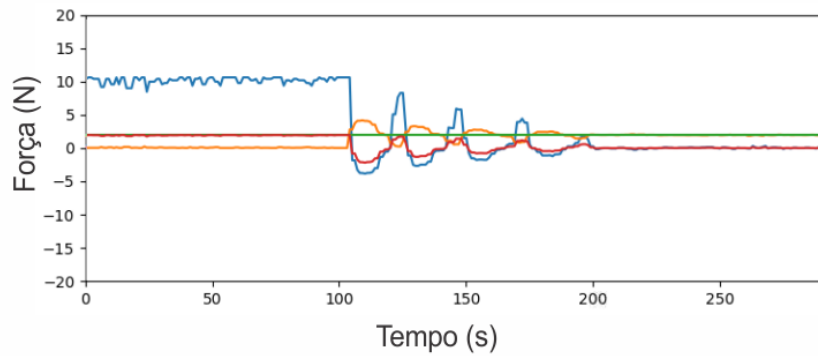


Figura 5.21: Comportamento do sistema mediante falha na interação entre o sensor de força e objeto

mantenha-se estabilizado. Sendo assim, através do sistema desenvolvido, as formas canônicas foram ajustas no software a fim de movimentar os dedos correspondentes. Para que isso fosse possível, através de uma segunda *Raspberry*, foi descentralizado o controle de conformidade da garra robótica, para possibilitar tanto a parametrização das formas canônicas, como a geração da interface gráfica. A comunicação entre as *Raspberry* deu-se através do protocolo de comunicação Paho e Mosquitto MQTT. As formas canônicas, bem como o controle de abertura e fechamento dos dedos foram realizados através do software Node-RED, conforme descrito anteriormente. Ao todo foram parametrizados quatro formas canônicas de agarre, sendo elas: pegar garrafa, pegar tipo pinça, pegar com três dedos e pegar com quatro dedos. Na Figura 5.22 é possível observar a estrutura configurada para manipulação da garra.

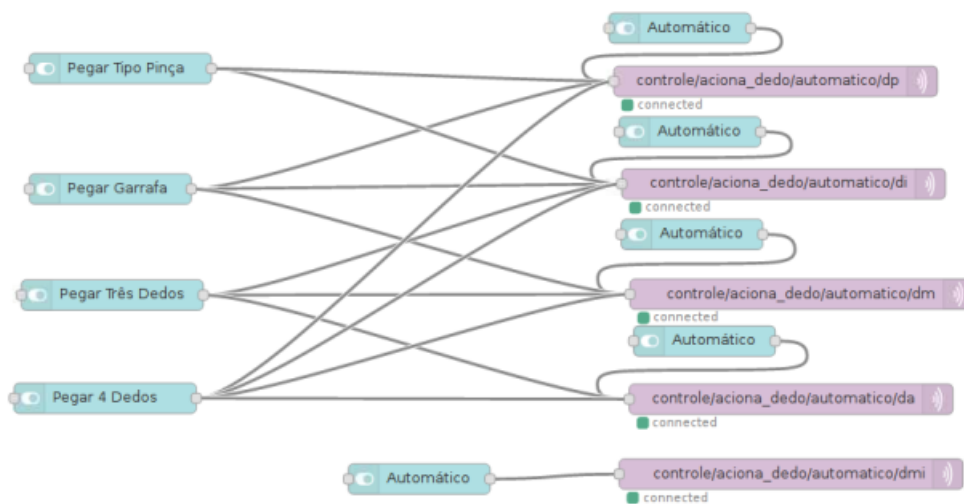


Figura 5.22: Programação das formas canônicas e controle de abertura e fechamento da garra robótica

Para o agarre “pegar garrafa” o cliente 2 publica a informação referente ao fechamento do dedo polegar, dedo indicador e dedo médio no servidor *broker* conforme descrito na Figura 4.17.

O cliente 1, por sua vez, assina através do servidor a informação nos seus respectivos tópicos e através do algoritmo desenvolvido, envia a informação para os motores correspondente mandando fechar os dedos, possibilitando realizar o agarre. A forma canônica “pegar garrafa” pode ser visualizado na Figura 5.23.



Figura 5.23: Agarre da forma canônica “pegar garrafa”

Já para a forma canônica “pegar tipo pinça” o cliente 2 publica a informação referente ao fechamento do dedo polegar e dedo indicador no servidor *broker*. O cliente 1, por sua vez, assina através do servidor a informação nos seus respectivos tópicos e através do algoritmo desenvolvido, envia a informação para os motores correspondente mandando fechar os dedos, possibilitando realizar o agarre. A forma canônica “pegar tipo pinça” pode ser visualizado na Figura 5.24.

Já para a forma canônica “pegar com três dedos” e “pegar com quatro dedos” são realizadas da mesma forma, podendo ser visualizadas na Figura 5.25.

Com os testes realizados a validação e experimentação da garra robótica antropomórfica, com base na metodologia de desenvolvimento de produto, é comprovada. Cabe, entretanto, por parte do projetista, o estudo em temas relacionados a mecânica da garra, como exemplo, simulação da dinâmica e cálculo do jacobiano a fim de verificar as acelerações e comportamento da garra robótica a nível de engenharia. Além de conceitos relacionados ao sistema de controle para verificação de tempo de resposta, *overshoot*, entre outros.

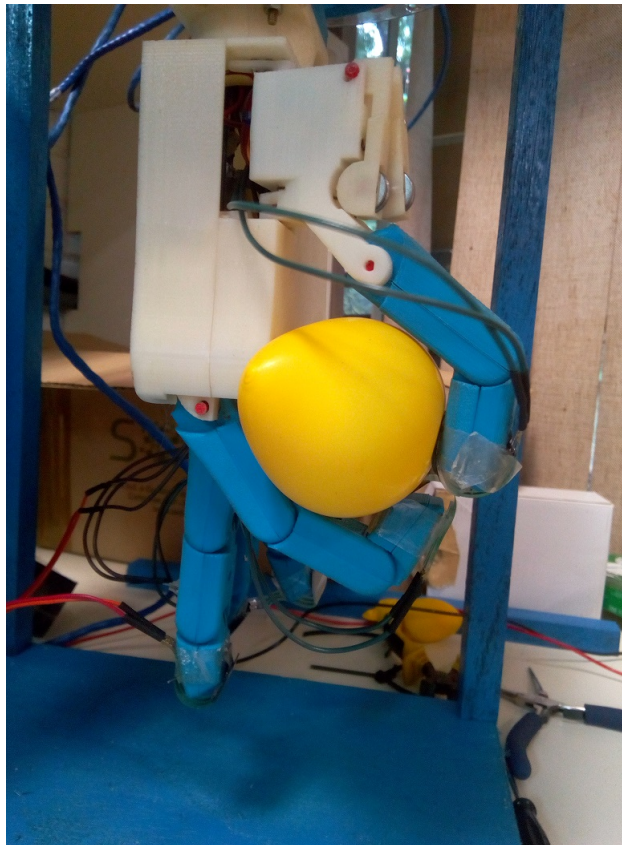
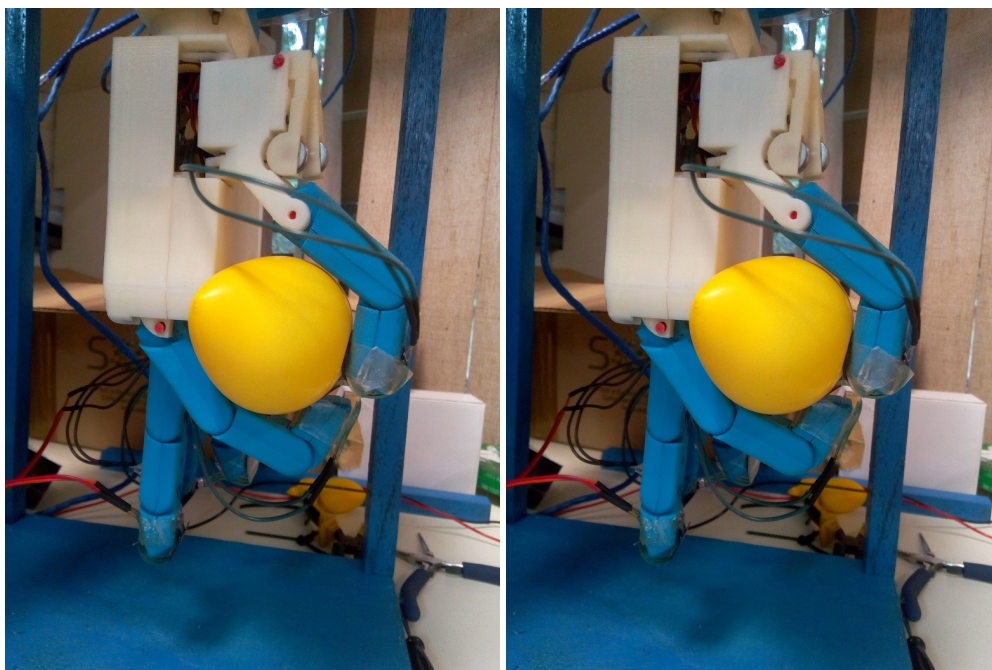


Figura 5.24: Agarre da forma canônica “pegar tipo pinça”



(a)

(b)

Figura 5.25: Formas canônicas para a garra robótica antropomórfica. Figura (a) Agarre da forma canônica “pegar três dedos” e Figura (b) Agarre da forma canônica “pegar quatro dedos”

Capítulo 6

Conclusões

O desenvolvimento da dissertação possibilitou a montagem da garra robótica antropomórfica mediante metodologia de desenvolvimento de produto conforme os requisitos dos clientes. O trabalho baseou-se em uma proposta de baixo custo, utilizando materiais de fácil aquisição no mercado nacional. Através do desenvolvimento da garra, além da construção do mecanismo, foi possível a análise cinemática e o emprego do controle de conformidade através da lógica *fuzzy*.

De modo geral, o desenvolvimento da garra robótica antropomórfica mostrou-se efetivo, onde a garra possui a forma anatômica da mão humana, possuindo 5 dedos, sendo um deles o polegar, atendendo a uns dos requisitos principais dos clientes. O emprego da garra robótica antropomórfica possibilita a manipulação de objetos com diferentes pesos e tamanhos, possibilitando o agarre com estabilidade. Enquanto as garras robóticas do tipo pinça manipulam apenas cerca de 40% dos objetos das tarefas realizadas no dia a dia, a garra com 5 dedos possibilita a manipulação de quase todos os objetos, conforme demonstrado no Capítulo 5.

Mediante as especificações metas requeridas na Tabela 4.3 para a garra robótica desenvolvida, os itens 1.3, 1.5, 1.7, 3.1 e 3.2 não foram atingidos satisfatórios ou não estimados. Para o item 1.3, devido aos motores serem pequenos para poder caber internamente na palma da mão, não possuem uma capacidade máxima de 2kg/cm. Conforme o *datasheet* do motor utilizado, possui 1kg/cm. Para o item 1.5, a geração da interface gráfica teve que ser realizada através do *software* Node-RED para possibilitar a descentralização do sistema de controle do *cliente1* para manter a eficiência do sistema. Quanto ao item 1.7, conforme as limitações do autor, não foi possível estimar a precisão do sistema de controle desenvolvido. Para o item 3.1 e 3.2, os movimentos empregados na seção 5.3.1, tanto de forma manual e automático, ocorreu-se de maneira geral, pois, há necessidade de realizar melhorias na garra para possibilitar a classificação do agarre. Entretanto, esses itens, conforme hierarquização dos requisitos dos clientes não eram de suma importância, como por exemplo, o controle de conformidade e a aparência da garra robótica, os quais foram atingidos. Na Tabela 6.1 é representados os itens que foram atingidos ou não.

Quanto ao sistema de controle de conformidade, há necessidade de realizar o ajuste da lógica

Tabela 6.1: Requisitos de projeto atingidos conforme a necessidades dos clientes

Requisitos do Projeto	Valor Meta	Atingido?
1.1 - Quantidade de dedos nas mãos	5	Sim
1.2 - Sensor de medição de força	Medição Direta	Sim. Através do sensor FSR
1.3 - Capacidade máxima de manipulação	2 Kgf	Não. Força máxima 0,8kgf
1.4 - Controle de conformidade	Lógica Difusa	Sim. Fuzzy-Logic Toolbox
1.5 - Interface gráfica	Python	Não. Node-Red
1.6 - Sistema de fixação	Flange	Sim
1.7 - Precisão sistema de controle	95%	Não estimado
2.1 - Custo total de produção	R\$2000,00	R\$2.111,97
2.2 - Produtos encontrados no mercado brasileiro	-	Sim
2.3 - Graus de liberdade	15	15
2.4 - Possuir módulos de montagem	-	47
3.1 - Formas canônicas de movimento de força	4	Não estimado
3.2 - Formas canônicas de movimento de precisão	4	Não estimado
4.1 - Potência consumida	500W	Sim
4.2 - Componentes reprogramáveis	Código Aberto	Sim

fuzzy para os *setpoints* inferiores a 0.5N e superiores a 8N. Devido a utilização do mínimo de conjunto *fuzzy*, para os *ranges* de operação considerado, o sistema torna-se ineficiente. Cabe ressaltar que para o sistema de controle de conformidade desenvolvido o tempo de resposta ficou em cerca de 300ms para os 5 dedos, conforme testes realizados na Seção 5.3.

Já o mecanismo de transmissão de movimento, pode em versões futuras, ser substituído por um mecanismo com retorno das falanges dos dedos através de mola. Com esse mecanismo, possibilitara a substituição dos extensores de nylon reduzindo a base de acoplamento dos motores, possibilitando aumentar a área de trabalho da garra robótica. O emprego de um mecanismo através de polia e fio de nylon, onde um abre e o outro fecha, para o fechamento há o controle de conformidade, porém, para a abertura à necessidade do emprego de um fim de curso em cada dedo para que o sistema de controle saiba onde é a posição de totalmente aberto, mandando desligar os motores. Outrossim, teria a possibilidade do emprego de um sistema de controle de posição independente em uma terceira *Raspberry* a fim de controlar a posição do dedo, evitando assim, problemas com o mecanismo, conforme revisões na literatura. Entretanto, o controle desenvolvido atende ao requisito do cliente estabelecido no item 1.4 da Tabela 4.3.

Conforme as análises cinemáticas desenvolvidas, o mecanismo apresentou erros entre 0,28

mm à 1,83 mm devido a erros de posicionamento inicial. A medição deu-se através do software *SolidWorks* e comparados com os valores calculados através do algoritmo de DH. Há necessidade de travar o mecanismo em uma posição de referência para realizar todas as medições. Considerando as folgas e desgaste do mecanismo, nem sempre é possível obter a referência 0. Porém, os resultados são satisfatórios, onde o modelo cinemático desenvolvido, corresponde ao modelo cinemático real da garra robótica. Entretanto, em relação à dinâmica do mecanismo, devido ao conhecimento mínimo do autor na área da dinâmica, não foi possível realizar os cálculos e simulações, contudo, não é um dos itens fundamentais dos requisitos dos clientes.

Comparando a garra robótica desenvolvida as relacionadas nos trabalhos correlatos, pode-se observar que tanto a implementação, como os recursos empregados foram diferentes. A garra robótica desenvolvida por Pertuz [99], possui controle de impedância passivo e mecanismo de transmissão de movimento por barras. Enquanto o trabalho desenvolvido possui controle de conformidade baseado em lógica *fuzzy* com medição direta e o mecanismo de transmissão por fio de nylon e polias. A vantagem do controle de conformidade baseado em lógica *fuzzy* em alternativa ao PID foram descritas na seção Seção 2.6 possibilitando a implementação do controle dos cinco dedos da garra robótica na *Raspberry Pi 3 B+*.

No trabalho desenvolvido por Yulin et al. [66], se depararam com as fricções e elasticidades do mecanismo gerando incertezas no modelo de controle desenvolvido. No trabalho aqui desenvolvido, devido à utilização da lógica fuzzy, essas incertezas são desprezadas para o modelo, considerando a experiência do projetista. Já no trabalho realizado por Mahmoud e Aníbal [61] obtiveram ótimos resultados com uma garra sub-atuada com 10 GDL, onde a mesma possibilita o agarre de 10 formas canônicas realizadas pela mão humana. Porém, utilizam motores *Dynamixel AX-12* de alto custo, não sendo encontrado no mercado brasileiro, além do que, para atender aos requisitos definidos pelos clientes o controle deve ser efetivo e de baixo custo.

A maior contribuição do trabalho, consiste no desenvolvimento da garra robótica com baixo custo empregando o sistema de controle de conformidade com recursos encontrados no mercado nacional a fim de ser replicado por estudantes e pesquisadores que estejam iniciando no mundo da robótica. O modelo aqui proposto, teve como inspiração, o modelo desenvolvido pelo projeto *The Open Hand Project* [18]. O modelo foi remodelado, sendo melhorado o mecanismo de suporte dos motores, transmissão de movimento e estrutura dos dedos, a fim de comportar os sensores de força e implementação do controle de conformidade com o intuito de atender as necessidades dos clientes, conforme metodologia proposta. Todo o projeto desenvolvido pode ser encontrado no site www.arturvitorio.weebly.com a fim de ser replicado.

6.1 Trabalhos futuros

Com o intuito de melhorar o sistema de controle de conformidade, será desenvolvido uma rede de *Raspberry pi 3+*, descentralizando o sistema de controle de conformidade de um único *Rapsberry*. Com isso será possível a implementação de mais conjuntos *fuzzys*, possibilitando assim, um controle mais preciso da garra robótica antropomórfica, além de otimizar o tempo de resposta do mecanismo.

Há ainda a necessidade de se realizar um estudo sobre a aplicação de força para manipulação dos objetos, possibilitando um agarre estável. Da forma que o trabalho aqui foi desenvolvido, o *setpoint* de força foi configurado para uma referência de força e assim, manipulado os diferentes objetos. O emprego dessa força de manipulação da-se por não existir um estudo prévio dizendo quanto de força deve ser aplicado em determinado objeto, sendo necessário visualizar a deformação visualmente. Ainda para o emprego dessa forma de manipulação, o sistema precisa saber o peso do objeto e as características consideradas do objeto para poder ajustar os parâmetros de referência.

Outrossim, com o intuito de melhorar a interação objeto-dedo, será desenvolvido outro mecanismo para os dedos, comportando internamente, sensores *strain-gauge* pegando toda área de atuação dos dedos. Com isso, as formas canônicas de agarre poderão ser ampliadas, manipulando um *range* maior de objetos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Z. Kappassov, J.-A. Corrales, and V. Perdereau, “Tactile sensing in dexterous robot hands review,” *Robotics and Autonomous Systems*, vol. 74, pp. 195 – 220, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889015001621>
- [2] M. P. Groover, *Automação industrial e sistemas de manufatura*. Pearson; Engenharia edition, 2011.
- [3] A. Singh, D. Singh, and S. Dwivedy, “Design and fabrication of a gripper for grasping irregular objects,” 2016.
- [4] K. Telegenov, Y. Tlegenov, and A. Shintemirov, “A low-cost open-source 3-d-printed three-finger gripper platform for research and educational purposes,” vol. 3, 2015, pp. 638–647.
- [5] Y. Fernando, A. Mathath, and M. A. Murshid, “Improving productivity: A review of robotic applications in food industry,” 2016.
- [6] “Robotics innovation: a kitchen that cooks for itself,” <http://http://www.moley.com/>, accessed: 2017-03-08.
- [7] “Robotics innovation: a kitchen that cooks for itself,” <http://www.gov.uk/government/news/robotics-innovation-a-kitchen-that-cooks-for-itself>, accessed: 2017-03-08.
- [8] R. R. E. R. D. S. B. L. S. PONS, J. L.; CERES and W. VAN MOORLEGHEM, “Objectives and technological approach to the development of the multifunctional manus upper limb prosthesis,” 2016.
- [9] A. Bicchi, “Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, Dec 2000.
- [10] R. S. Johansson and J. R. Flanagan, “Coding and use of tactile signals from the fingertips in object manipulation tasks,” *Nature Reviews Neuroscience*, 2009.
- [11] M. Prats, Á. Pobil, and P. Sanz, *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback: Applications to Assistive Robotics*, ser. Springer

- Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2012. [Online]. Available: <https://books.google.com.br/books?id=3r25BQAAQBAJ>
- [12] K. Ulrich and S. Eppinger, *Product Design and Development*. McGraw-Hill Education; 5 edition, 2012.
- [13] P. K. Parida, “Kinematic analysis of multi-fingered, anthropomorphic robotic hands,” Ph.D. dissertation, 2013.
- [14] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2008.
- [15] A. S. Sadun, J. Jalani, and J. Abdul Sukor, “An overview of active compliance control for a robotic hand,” *ARPJ Journal of Engineering and Applied Sciences*, vol. 11, pp. 11 872–11 876, 2016.
- [16] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, K. J., R. C. Voorhies, G. S. Sukhatme, and S. Schaal, “An autonomous manipulation system based on force control and optimization,” *Autonomous Robots*, vol. 36, pp. 11–30, 2014.
- [17] D. I. de Souza, D. M. Müller, M. A. T. Fracassi, and S. B. B. Romeiro, *Manual de orientação para projetos de pesquisa*. Liberato, 2013.
- [18] “The open hand project,” <http://www.openhandproject.org/>, accessed: 2017-07-19.
- [19] G. Pahl, W. Beitz, H. Schulz, and U. Jarecki, *Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. Springer Berlin Heidelberg, 2013. [Online]. Available: <https://books.google.com.br/books?id=qJkiBgAAQBAJ>
- [20] L. Birglen, C. Gosselin, and T. Laliberté, *Underactuated Robotic Hands*. Springer, 2008.
- [21] J. K. Salisbury and J. Craig, “Articulated hands, force control and kinematic issues,” *Int. J. of Robotics Research*, p. Vol. 1. N1, 1982.
- [22] T. Okada, “Computer control of multijointed finger system for precise object-handling,” *IEEE Transactions on Systems, Man and Cybernetics*, pp. 289–299, 1982.
- [23] N. Ulrich and V. Kumar, “Grasping using fingers with coupled joints.” *ASME Trends and Developments in Mechanisms*, pp. 201–207, 1988.
- [24] M. Rakic, “Multifingered robot hand with selfadaptability,” *Robotics and Computer-integrated Manufacturing*, pp. 269–276, 1989.
- [25] S. C. Jacobse, E. K. Iversen, R. T. J. D. F. Knutti, and K. B. Biggers, “Design of the utah/m.i.t. dextrous hand,” *IEEE International Conference on Robotics and Automation*, pp. 1520–1532, 1986.

- [26] P. Chappell, *Mechatronic Hands: Prosthetic and Robotic Design*, ser. Control, Robotics and Sensors. Institution of Engineering and Technology, 2016. [Online]. Available: <https://books.google.com.br/books?id=f-1IDAAAQBAJ>
- [27] M. A. D. Alba and R. Ponticelli, “An introductory revision to humanoid robot hands, instituto de automática industrial,” *Instituto de Automática Industrial, Carretera de Campo Real Km.0220; La Poveda, Arganda del Ray, 28500 Madrid Apartado 56; ESPAÑA*, 2005.
- [28] Y. Matsuoka, “Embodiment and manipulation learning process for a humanoid hand,” 1995.
- [29] A. A. I. Escrivá, “Design of a smart gripper for industrial applications.” 2016.
- [30] J. K. Salisbury and J. J. Craig, “Articulated hands: Force control and kinematic issues,” *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982. [Online]. Available: <https://doi.org/10.1177/027836498200100102>
- [31] ———, “Kinematic and force analysis of articulated mechanical hands,” *ASME. J. Mech., Trans., and Automation*, vol. 105, no. 1, pp. 35–34, 1983.
- [32] I. D. McCammon and S. C. Jacobsen, *Tactile Sensing and Control for the Utah/MIT Hand*. New York, NY: Springer New York, 1990, pp. 239–266. [Online]. Available: https://doi.org/10.1007/978-1-4613-8974-3_11
- [33] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers, “Design of the utah/m.i.t. dextrous hand,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, April 1986, pp. 1520–1532.
- [34] G. A. Bekey, R. Tomovic, and I. Zeljkovic, *Control Architecture for the Belgrade/USC Hand*. New York, NY: Springer New York, 1990, pp. 136–149. [Online]. Available: https://doi.org/10.1007/978-1-4613-8974-3_7
- [35] W. Townsend, “The barretthand grasper programmably flexible part handling and assembly,” *Industrial Robot: An International Journal*, vol. 27, no. 3, pp. 181–188, 2000. [Online]. Available: <https://doi.org/10.1108/01439910010371597>
- [36] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. H. Jin, Y. W. Liu, S. W. Fan, T. Lan, and Z. P. Chen, “Multisensory five-finger dextrous hand: The dlr/hit hand ii,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3692–3697.
- [37] A. Caffaz and G. Cannata, “The design and development of the dist-hand dextrous gripper,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 3, May 1998, pp. 2075–2080 vol.3.

- [38] C. S. Lovchik and M. A. Diftler, “The robonaut hand: a dexterous robot hand for space,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, May 1999, pp. 907–912 vol.2.
- [39] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, “Dlr-hand ii: next generation of a dextrous robot hand,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, May 2001, pp. 109–114 vol.1.
- [40] N. Fukaya, S. Toyama, T. Asfour, and R. Dillmann, “Design of the tuat/karlsruhe humanoid hand,” in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 3, Oct 2000, pp. 1754–1759 vol.3.
- [41] S. Schulz, C. Pylatiuk, and G. Bretthauer, “A new ultralight anthropomorphic hand,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, May 2001, pp. 2437–2441 vol.3.
- [42] H. Kawasaki, T. Komatsu, K. Uchiyama, and T. Kurimoto, “Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii,” in *IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, vol. 2, Oct 1999, pp. 782–787 vol.2.
- [43] L. Resnik, S. L. Klinger, and K. Etter, “The deka arm: Its features, functionality, and evolution during the veterans affairs study to optimize the deka arm,” *Prosthetics and Orthotics International*, vol. 38, no. 6, pp. 492–504, 2014, pMID: 24150930. [Online]. Available: <https://doi.org/10.1177/0309364613506913>
- [44] F. M and G. G, “Blackfingers: Artificial hand that copies human hand in structure, size and functions,” In: *Proceedings of Humanoids*, 2000.
- [45] C. MC, M. B, Z. M. Micera S, and D. P, “A wearable artificial hand for prosthetics and humanoid robotics applications,” In: *Proceedings of the IEEE RAS International Conference on Humanoid Robots*, 2001.
- [46] D. L. Akin, C. R. Carignan, and A. W. Foster, “Development of a four-fingered dexterous robot end effector for space operations,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, May 2002, pp. 2302–2308 vol.3.
- [47] P. Tuffield and H. Elias, “The shadow robot mimics human actions,” *Industrial Robot: the international journal of robotics research and application*, vol. 30, no. 1, pp. 56–60, 2003. [Online]. Available: <https://doi.org/10.1108/01439910310457715>
- [48] G. M, A. A, and N. M, “Thing: A robotic hand with realistic thumb pronations,” *Proceedings of the FCRAR*, 2002.

- [49] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, “Development of a high-speed multifingered hand system and its application to catching,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, Oct 2003, pp. 2666–2671 vol.3.
- [50] A. F. C. Silva, “Desenvolvimento e caracterização de uma mão robótica acionada por atuadores de liga com memória de forma,” 2015.
- [51] J. M. Rosário, *Princípios de Mecatrônica*. Pearson, 2004.
- [52] “International federation of robotics,” <https://ifr.org/>, accessed: 2018-07-19.
- [53] F. L. Cunha, H. J. A. Schneebeli, and V. I. Dynnikov, “Development of anthropomorphic upper limb prosthesis with human-like interphalangeal and interdigital couplings.” *Artificial Organs*, pp. Vol. 24, No. 3, pp. 193–197, 2000.
- [54] M. Tits, “Development of an optical motion capture setup for feature extraction and statistical analysis of the pianist’s expert gestures,” *Completed at CIRMMT, McGill University, Montreal (QC), Canada*, 2014.
- [55] B. Gao, S. Yang, H. Jin, Y. Hu, X. Yang, and J. Zhang, “Design and analysis of underactuated robotic gripper with adaptive fingers for objects grasping tasks,” *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 987–992, 2016.
- [56] B. Massa, S. Roccella, M. C. Carrozza, and P. Dario, “Design and development of an underactuated prosthetic hand,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, May 2002, pp. 3374–3379 vol.4.
- [57] M. Tavakoli and A. T. de Almeida, “Adaptive under-actuated anthropomorphic hand: Isrsofthand,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 1629–1634.
- [58] G. Cerruti, D. Chablat, D. Gouaillier, and S. Sakka, “Design method for an anthropomorphic hand able to gesture and grasp,” *CoRR*, vol. abs/1504.01151, 2015.
- [59] T. Feix, “Anthropomorphic hand optimization based on a latent space analysis,” 2011.
- [60] I. M. Bullock, J. Z. Zheng, S. de La Rosa, C. Guertler, and A. M. Dollar, “Grasp frequency and usage in daily household and machine shop tasks,” 2013.
- [61] M. Tavakoli and A. T. de Almeida, “Adaptive under-actuated anthropomorphic hand: Isrsofthand,” 2014.
- [62] Y. Liu, Y. Zhang, and Q. Xu, “Design and control of a novel compliant constant-force gripper based on buckled fixed-guided beams,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 476–486, Feb 2017.

- [63] J. Jalani, “Robust active compliance controle for robot hand,” 2013.
- [64] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. Gersem, “Active compliant motion: a survey.” *Advanced Robotics*, vol. 19, pp. 479–499, 2005.
- [65] M. G. S. Simões and I. S. S. Shaw, *Controle e Modelagem Fuzzy*. Blucher, 2007.
- [66] X. Yulin, J. Caijun, and Y. Jie, “Compliance control for grasping with a bionic robot hand,” 2016.
- [67] J.-U. Chu, D.-H. Jung, and Y.-J. Lee, “Design and control of a multifunction myoelectric hand with new adaptive grasping and self-locking mechanisms,” 2008.
- [68] C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller. i,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404–418, Mar 1990.
- [69] F. A. C. Gomide and R. R. Gudwin, “Modelagem, controle, sistemas e lógica fuzzy,” *SBA controle & Automação*, vol. 4, no. 3, pp. 97–115, 1994.
- [70] W. T. Townsend, “The barrett hand grasper-programmably flexible part handling and assembly,” *Industrial Robot: An International Journal*, pp. 181–188, 2000.
- [71] “Syntouch,” <https://www.syntouchinc.com/en/prosthetics/>, accessed: 2018-07-19.
- [72] P. S. Girão, P. M. P. Ramos, O. Postolache, and J. M. D. Pereira, “Tactile sensor for robotic applications,” *Measurement* 46, pp. 1257–1271, 2013.
- [73] R. D. D?Souza, A. Joshy, A. Mini, and N. Panicker, “The art of tactile sensing: A state of art survey,” *International Journal of Sciences: Basic and Applied Research*, pp. Volume 26, No 3, pp 252–266, 2016.
- [74] A. M. Almassri, W. Z. W. Hasan, S. Ahmad, A. M. A. J. Ishak, D. T. Ghazali, and C. Wada, “Pressure sensor: State of the art, design, and application for robotic hand,” *Journal of Sensor*, pp. 1257–1271, 2015.
- [75] M. F. A. Jamil, J. Jalani, A. Ahmad, and E. I. S. Saadon, “Active compliance control of the RED hand: A PID control approach,” in *Towards Autonomous Robotic Systems - 17th Annual Conference, TAROS 2016, Sheffield, UK, June 26 - July 1, 2016, Proceedings*, 2016, pp. 1–7. [Online]. Available: https://doi.org/10.1007/978-3-319-40379-3_1
- [76] L. B. Bridgwater, C. A. Ihrke, M. A. Diftler, M. E. Abdallah, N. A. Radford, J. M. Rogers, S. Yayathi, R. S. Askew, and D. M. Linn, “The robonaut 2 hand - designed to do work with tools,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3425–3430.

- [77] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, Dec 2011.
- [78] C.-C. Wen and W. Fang, "Tuning the sensing range and sensitivity of three axes tactile sensors using the polymer composite membrane," *Sensors and Actuators A: Physical*, vol. 145-146, pp. 14 – 22, 2008, special Issue: Transducers/07 Eurosensors XXI, The 14th International Conference on Solid State Sensors, Actuators and Microsystems and the 21st European Conference on Solid-State Transducers. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924424707007236>
- [79] Y. Hasegawa, M. Shikida, T. Shimizu, T. Miyaji, H. Sasaki, K. Sato, and K. Itoigawa, "Amicromachined active tactile sensor for hardness detection," *Sensors and Actuators A: Physical*, vol. 114, no. 2, pp. 141 – 146, 2004, selected papers from Transducers 03. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924424703007118>
- [80] H. Zhang and E. So, "Hybrid resistive tactile sensing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 1, pp. 57–65, Feb 2002.
- [81] A. Shashank, M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "Design, simulation and fabrication of a low cost capacitive tactile shear sensor for a robotic hand," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sept 2009, pp. 4132–4135.
- [82] H. Lee, J. Chung, S. Chang, and E. Yoon, "Normal and shear force measurement using a flexible polymer tactile sensor with embedded multiple capacitors," *Journal of Microelectromechanical Systems*, vol. 17, no. 4, pp. 934–942, Aug 2008.
- [83] C. Ko, S. Tseng, and M. S. . Lu, "A cmos micromachined capacitive tactile sensor with high-frequency output," *Journal of Microelectromechanical Systems*, vol. 15, no. 6, pp. 1708–1714, Dec 2006.
- [84] M. OHKA, H. KOBAYASHI, J. TAKATA, and Y. MITSUYA, "An experimental optical three-axis tactile sensor featured with hemispherical surface," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 2, no. 5, pp. 860–873, 2008.
- [85] M. Ohka, N. Morisawa, H. Suzuki, J. Takata, H. Koboyashi, and H. B. Yussof, "A robotic finger equipped with an optical three-axis tactile sensor," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 3425–3430.
- [86] H. Yussof, M. Ohka, H. Kobayashi, J. Takata, M. Yamano, and Y. Nasu, *Development of an Optical Three-Axis Tactile Sensor for Object Handing Tasks in Humanoid Robot*

- Navigation System*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 43–51. [Online]. Available: https://doi.org/10.1007/978-3-540-73424-6_6
- [87] X. Chen, J. Sakai, S. Yang, and S. Motojima, “Biomimetic tactile sensors with fingerprint-type surface made of carbon microcoils/polysilicone,” *Japanese Journal of Applied Physics*, vol. 45, no. 10L, p. L1019, 2006. [Online]. Available: <http://stacks.iop.org/1347-4065/45/i=10L/a=L1019>
- [88] N. Futai, N. Futai, K. Matsumoto, and I. Shimoyama, “A flexible micromachined planar spiral inductor for use as an artificial tactile mechanoreceptor,” *Sensors and Actuators, A: Physical*, vol. 111, no. 2-3, pp. 293–303, 3 2004.
- [89] S. Takenawa, “A soft three-axis tactile sensor based on electromagnetic induction,” in *2009 IEEE International Conference on Mechatronics*, April 2009, pp. 1–6.
- [90] A. S. Sadun, J. Jalani, and J. Sukor, “Force sensing resistor (fsr): A brief overview and the low cost sensor for active compliance control,” 2016.
- [91] S. I. Yaniger, “Force sensing resistor: A review of the technology,” *Electro International*, pp. 666–668, 1991.
- [92] F. B. Hernandez, “Sensor de força utilizando fiber taper,” 2016.
- [93] “Force sensing resistor integration guide and evaluation,” <https://www.interlinkelectronics.com/fsr-402>, accessed: 2018-07-19.
- [94] C. P. F. Kechinski, K. Rosa, L. R. da Echeveste, and M. E. Soares, “Análise do modelo de desenvolvimento de produto de uma empresa fabricante de produtos e soluções para o setor automotivo,” *Exacta São Paulo*, pp. 81–88, 2010.
- [95] L. C. Cheng, *QFD planejamento da qualidade*, 1195.
- [96] M. L. Giacomini, “Apoio à decisão para a implantação de ferramentas de projeto no processo de desenvolvimento de produtos,” 2007.
- [97] A. J. H. Fonseca, “Sistematização do processo de obtenção das especificações de projeto de produtos industriais e sua implementação computacional,” 2000.
- [98] E. Rodrigues, “Desenvolvimento de um robô com cinemática paralela delta linear para manufatura aditiva aderente a step-nc,” 2018.
- [99] S. Pertuz, “Sistema embarcado baseado em arquiteturas reconfiguráveis do controle dinâmico de uma mão robótica sintonizada com algoritmos bioinspirados,” 2017.
- [100] T. Zhang, S. Fan, J. Zhao, L. Jiang, and H. Liu, “Design and control of a multisensory five-finger prosthetic hand,” 2014.

- [101] A. A. M. F. A. Jamil, J. Jalani, “A new approach of active compliance control via fuzzy logic control for multifingered robot hand,” pp. 10011 – 10011 – 9, 2016. [Online]. Available: <https://doi.org/10.1117/12.2242807>
- [102] M. Faraj, N. Zainul Azlan, and S. Ahmad, “Cascade control of robotic fingers with anthropomorphic inspiration,” pp. 168–173, 10 2016.
- [103] T. S. P. Sono, “Projeto de um sistema de controle sub-atuado para uma prótese de mão,” 2008.
- [104] ———, “Projeto de uma prótese de mão sub-atuada,” 2013.
- [105] H. Scheurenbrand, “Force measurement and position and force control for a robot hand,” 2007.
- [106] R. Pfeifer, F. Iida, and G. Gomez, “Designing intelligent robots,” *Journal of the Robotics Society of Japan*, vol. 24, no. 7, pp. 783–790, 2006.
- [107] T. Takahashi, T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, M. Iribe, T. Fukushima, and M. Fujita, “Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control,” in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 264–271.
- [108] J. Ulmen and M. Cutkosky, “A robust, low-cost and low-noise artificial skin for human-friendly robots,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 4836–4841.
- [109] M. A. Diftler, C. J. Culbert, R. O. Ambrose, R. Platt, and W. J. Bluethmann, “Evolution of the nasa/darpa robonaut control system,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, Sept 2003, pp. 2543–2548 vol.2.
- [110] P. J. Kyberd and P. H. Chappell, “Object-slip detection during manipulation using a derived force vector,” *Mechatronics*, vol. 2, no. 1, pp. 1 – 13, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/095741589290034L>
- [111] A. M. Rocco and A. D. Silveira, “Ferramental para eficiência em vendas,” *Congresso de Administração e Gerência, Cascavel*, 2008.
- [112] R. L. Norton, *Cinemática e dinâmica dos mecanismos*. AMGH Editora, 2010.
- [113] S. B. Niku, *Introdução a Robótica - Análise, Controle e Aplicações*. LTC, 2013.
- [114] J. J. Craig, *Robótica*. Pearson, São Paulo, 3. ed, 2013.
- [115] “Uprint se plus,” <https://www.stratasys.com/br/3d-printers/uprint-se-plus>, accessed: 2018-07-20.

APÊNDICES

A. ALGORITMO DE CONTROLE DE CONFORMIDADE POR LÓGICA FUZZY

File - C:\Users\ARTUR\Desktop\mqtt6.py

```
1 import sys
2 import paho.mqtt.client as mqtt # importa o pacote mqtt
3 import time
4 import numpy as np
5 from gpiozero import MCP3008
6 import skfuzzy as fuzz
7 import RPi.GPIO as GPIO
8 from gpiozero import PWMOutputDevice
9 from time import sleep
10
11
12 forca_sensor_dm = MCP3008(channel=4, device=0) ## define
    leitura atraves do sensor de pressão dedo médio
13 forca_sensor_dp = MCP3008(channel=3, device=0) ## define
    leitura através do sensor de pressão dedo polegar
14 forca_sensor_di = MCP3008(channel=5, device=0)
15 forca_sensor_da = MCP3008(channel=6, device=0)
16 forca_sensor_dmi = MCP3008(channel=7, device=0)
17
18
19
20 aciona_dm = 0 ##definição variavel global dedo médio
21 aciona_dp = 0 ##definição variável global dedo polegar
22 aciona_di = 0 ##definição variável global dedo indicador
23 aciona_da = 0 ##definição variável global dedo indicador
24 aciona_dmi = 0 ##definição variável global dedo indicador
25 habilita_manual_dp = 0
26 habilita_manual_di = 0
27 habilita_manual_dm = 0
28 habilita_manual_da = 0
29 habilita_manual_dmi = 0
30
31
32 leitura_forca_dm_convertido= 0
33 forca_dm_convertido = 0
34
35 GPIO.setmode(GPIO.BCM)
36 GPIO.setup(17, GPIO.OUT) ## defini saída gpio dedo médio
37 GPIO.setup(18, GPIO.OUT) ## defini saída gpio dedo médio
38 GPIO.setup(12, GPIO.OUT) ## defini saída gpio dedo
    indicador
39 GPIO.setup(13, GPIO.OUT) ## defini saída gpio dedo
    indicador
40 GPIO.setup(19, GPIO.OUT) ## defini saída gpio dedo anelar
41 GPIO.setup(26, GPIO.OUT) ## defini saída gpio dedo anelar
```

Page 1 of 21

```
42 GPIO.setup(21, GPIO.OUT) ## defini saída gpio dedo mínimo
43 GPIO.setup(20, GPIO.OUT) ## defini saída gpio dedo mínimo
44 GPIO.setup(6, GPIO.OUT) ## defini saída gpio dedo polegar
45 GPIO.setup(5, GPIO.OUT) ## defini saída gpio dedo polegar
46 GPIO.setup(4, GPIO.OUT) ## defini saída pwm para controle
  de posição do polegar
47 pwm = GPIO.PWM(4, 100)
48 pwm.start(5)
49
50
51 ### Comando PWM DEDO MEDIO ###
52 PWM_FORWARD_LEFT_PIN_dm = 17
53 PWM_REVERSE_LEFT_PIN_dm = 18
54
55 forwardLeft_dm = PWMOutputDevice(PWM_FORWARD_LEFT_PIN_dm,
  True, 0, 1023)
56 reverseLeft_dm = PWMOutputDevice(PWM_REVERSE_LEFT_PIN_dm,
  True, 0, 1023)
57
58 ### Comando PWM DEDO POLEGAR ###
59
60 PWM_FORWARD_LEFT_PIN_dp = 5
61 PWM_REVERSE_LEFT_PIN_dp = 6
62
63 forwardLeft_dp = PWMOutputDevice(PWM_FORWARD_LEFT_PIN_dp,
  True, 0, 1023)
64 reverseLeft_dp = PWMOutputDevice(PWM_REVERSE_LEFT_PIN_dp,
  True, 0, 1023)
65
66
67 ### Comando PWM DEDO INDICADOR ###
68
69 PWM_FORWARD_LEFT_PIN_di = 12
70 PWM_REVERSE_LEFT_PIN_di = 13
71
72 forwardLeft_di = PWMOutputDevice(PWM_FORWARD_LEFT_PIN_di,
  True, 0, 1023)
73 reverseLeft_di = PWMOutputDevice(PWM_REVERSE_LEFT_PIN_di,
  True, 0, 1023)
74
75 ### Comando PWM DEDO ANELAR ###
76
77 PWM_FORWARD_LEFT_PIN_da = 19
78 PWM_REVERSE_LEFT_PIN_da = 26
79
```

```
80 forwardLeft_da = PWMOutputDevice(PWM_FORWARD_LEFT_PIN_da,
    True, 0, 1023)
81 reverseLeft_da = PWMOutputDevice(PWM_REVERSE_LEFT_PIN_da,
    True, 0, 1023)
82
83
84 ### Comando PWM DEDO MINÍMO ###
85
86 PWM_FORWARD_LEFT_PIN_dmi = 21
87 PWM_REVERSE_LEFT_PIN_dmi = 20
88
89 forwardLeft_dmi = PWMOutputDevice(
    PWM_FORWARD_LEFT_PIN_dmi, True, 0, 1023)
90 reverseLeft_dmi = PWMOutputDevice(
    PWM_REVERSE_LEFT_PIN_dmi, True, 0, 1023)
91
92 #### Conexão MQTT
93
94 broker = "164.41.45.179" # define o host do broker mqtt'
95 port = 1883 # define a porta do broker
96 keepAlive = 60 # define o keepAlive da conexao
97 topic = 'controle/#' # define o topico que este script
    assinara
98
99
100 # funcao on_connect sera atribuida e chamada quando a
    conexao for iniciada
101 # ela printara na tela caso tudo ocorra certo durante a
    tentativa de conexao
102 # tambem ira assina o topico que foi declarado acima
103 def on_connect(client, userdata, flags, rc):
104     print("[STATUS] Conectado ao Broker. Resultado de
    conexao: "+str(rc))
105
106     client.subscribe(topic)
107
108 # possui o mesmo cenario que o on_connect, porem, ela
    sera atrelada ao loop
109 # do script, pois toda vez que receber uma nova mensagem
    do topico assinado, ela sera invocada
110 def on_message(client, userdata, msg):
111     global aciona_dm
112     global aciona_dp
113     global aciona_di
114     global aciona_da
```



```
115     global aciona_dmi
116     global habilita_manual_dm
117     global habilita_manual_dp
118     global habilita_manual_di
119     global habilita_manual_da
120     global habilita_manual_dmi
121
122
123
124         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
125         # e, caso receber qualquer outra coisa, apagara o
led
126
127     message = str(msg.payload) # converte a mensagem
recebida
128     print("[MSG RECEBIDA] Topico: "+msg.topic+" /
Mensagem: "+ message) # imprime no console a mensagem
129
130     if msg.topic == 'controle/aciona_dedo/automatico/dm':
131
132         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
133         # e, caso receber qualquer outra coisa, apagara o
led
134         if(message == 'true'):
135             aciona_dm = 1
136         else:
137             aciona_dm = 0
138         print aciona_dm
139
140     if msg.topic == 'controle/aciona_dedo/manual/dm':
141
142         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
143         # e, caso receber qualquer outra coisa, apagara o
led
144         if(message == 'true'):
145             habilita_manual_dm = 1
146             reverseLeft_dm.value = 1
147         else:
148             habilita_manual_dm = 0
149             reverseLeft_dm.value = 0
150
151     if msg.topic == 'controle/aciona_dedo/automatico/dp':
```

```
152
153     # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
154     # e, caso receber qualquer outra coisa, apagara o
led
155     if(message == 'true'):
156         aciona_dp = 1
157     else:
158         aciona_dp = 0
159     #print aciona_dp
160
161     if msg.topic == 'controle/aciona_dedo/manual/dp':
162
163         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
164         # e, caso receber qualquer outra coisa, apagara o
led
165         if(message == 'true'):
166             habilita_manual_dp = 1
167             reverseLeft_dp.value = 1
168         else:
169             habilita_manual_dp = 0
170             reverseLeft_dp.value = 0
171
172     if msg.topic == 'controle/angulo':
173
174         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
175         # e, caso receber qualquer outra coisa, apagara o
led
176         angle = message
177         duty = float(angle)/10.0 + 1
178         print angle
179         print duty
180         pwm.ChangeDutyCycle(duty)
181
182     if msg.topic == 'controle/aciona_dedo/automatico/di':
183
184         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
185         # e, caso receber qualquer outra coisa, apagara o
led
186         if(message == 'true'):
187             aciona_di = 1
188         else:
```

```
189         aciona_di = 0
190
191     if msg.topic == 'controle/aciona_dedo/manual/di':
192
193         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
194         # e, caso receber qualquer outra coisa, apagara o
led
195         if(message == 'true):
196             habilita_manual_di = 1
197             reverseLeft_di.value = 1
198         else:
199             habilita_manual_di = 0
200             reverseLeft_di.value = 0
201
202     if msg.topic == 'controle/aciona_dedo/automatico/da':
203
204         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
205         # e, caso receber qualquer outra coisa, apagara o
led
206         if(message == 'true):
207             aciona_da = 1
208         else:
209             aciona_da = 0
210
211     if msg.topic == 'controle/aciona_dedo/manual/da':
212
213         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
214         # e, caso receber qualquer outra coisa, apagara o
led
215         if(message == 'true):
216             habilita_manual_da = 1
217             reverseLeft_da.value = 1
218         else:
219             habilita_manual_da = 0
220             reverseLeft_da.value = 0
221
222     if msg.topic == 'controle/aciona_dedo/automatico/dmi'
:
223
224         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
225         # e, caso receber qualquer outra coisa, apagara o
```

```
225 led
226     if(message == 'true'):
227         aciona_dmi = 1
228     else:
229         aciona_dmi = 0
230
231     if msg.topic == 'controle/aciona_dedo/manual/dmi':
232
233         # basicamente nessa condicional testara se o
valor recebido sera 1, sendo 1 acende o led
234         # e, caso receber qualquer outra coisa, apagara o
led
235         if(message == 'true'):
236             habilita_manual_dmi = 1
237             reverseLeft_dmi.value = 1
238         else:
239             habilita_manual_dmi = 0
240             reverseLeft_dmi.value = 0
241
242
243
244
245 try:
246     print("[STATUS] Inicializando MQTT...")
247
248     client = mqtt.Client() # instancia a conexao
249     client.on_connect = on_connect # define o callback do
evento on_connect
250     client.on_message = on_message # define o callback do
evento on_message
251
252     client.connect(broker, port, keepAlive) # inicia a
conexao
253     client.loop_start() # a conexao mqtt entrara em loop
ou seja, ficara escutando e processando todas mensagens
recebidas
254
255     while True:
256
257         ##Controle do Dedo Médio
258         if habilita_manual_dm == 0:
259             leitura_forca_dm = forca_sensor_dm.value *
1000
260
261             forca_dm = np.arange(0, 1000, 1)
```

```
262         setpoint_dm = np.arange(0, 1000, 1)
263         control_dm = np.arange(-1000, 1000, 1)
264
265         forca_dm_lo    = fuzz.trimf(forca_dm, [0, 0,
266     500])
267         forca_dm_md    = fuzz.trimf(forca_dm, [0, 500,
268     1000])
269         forca_dm_hi    = fuzz.trimf(forca_dm, [500,
270     1000, 1000])
271
272         setpoint_dm_lo = fuzz.trimf(setpoint_dm, [0,
273     0, 500])
274         setpoint_dm_md = fuzz.trimf(setpoint_dm, [0,
275     500, 1000])
276         setpoint_dm_hi = fuzz.trimf(setpoint_dm, [500
277     , 1000, 1000])
278
279         control_dm_lo = fuzz.trimf(control_dm, [-1000
280     , -1000, 0])
281         control_dm_md = fuzz.trimf(control_dm, [-1000
282     , 0, 1000])
283         control_dm_hi = fuzz.trimf(control_dm, [0,
284     1000, 1000])
285
286         forca_dm_medida = leitura_forca_dm
287         setpoint_dm_medido = 200
288
289         forca_dm_level_lo = fuzz.interp_membership(
290     forca_dm, forca_dm_lo, float(forca_dm_medida))
291         forca_dm_level_md = fuzz.interp_membership(
292     forca_dm, forca_dm_md, float(forca_dm_medida))
293         forca_dm_level_hi = fuzz.interp_membership(
294     forca_dm, forca_dm_hi, float(forca_dm_medida))
295
296         setpoint_dm_level_lo = fuzz.interp_membership
297     (setpoint_dm, setpoint_dm_lo, float(setpoint_dm_medido))
298         setpoint_dm_level_md = fuzz.interp_membership
299     (setpoint_dm, setpoint_dm_md, float(setpoint_dm_medido))
300         setpoint_dm_level_hi = fuzz.interp_membership
301     (setpoint_dm, setpoint_dm_hi, float(setpoint_dm_medido))
302
303         active_rule1_dm = np.fmin(forca_dm_level_lo,
304     setpoint_dm_level_md)
305         control_activation_1_dm = np.fmin(
306     active_rule1_dm, control_dm_hi)
```

```
290
291         active_rule2_dm = np.fmin(forca_dm_level_lo,
setpoint_dm_level_hi)
292         control_activation_2_dm = np.fmin(
active_rule2_dm, control_dm_hi)
293
294         active_rule3_dm = np.fmin(forca_dm_level_md,
setpoint_dm_level_lo)
295         control_activation_3_dm = np.fmin(
active_rule3_dm, control_dm_lo)
296
297         active_rule4_dm = np.fmin(forca_dm_level_md,
setpoint_dm_level_hi)
298         control_activation_4_dm = np.fmin(
active_rule4_dm, control_dm_hi)
299
300         active_rule5_dm = np.fmin(forca_dm_level_hi,
setpoint_dm_level_lo)
301         control_activation_5_dm = np.fmin(
active_rule5_dm, control_dm_lo)
302
303         active_rule6_dm = np.fmin(forca_dm_level_hi,
setpoint_dm_level_md)
304         control_activation_6_dm = np.fmin(
active_rule6_dm, control_dm_lo)
305
306         c1_dm = np.fmax(control_activation_1_dm,
control_activation_2_dm)
307         c2_dm = np.fmax(control_activation_3_dm,
control_activation_4_dm)
308         c3_dm = np.fmax(control_activation_5_dm,
control_activation_6_dm)
309         c4_dm = np.fmax(c2_dm, c3_dm)
310         aggregated_dm = np.fmax(c1_dm, c4_dm)
311
312         control_value_dm = fuzz.defuzz(control_dm,
aggregated_dm, 'centroid')
313
314         error_forca_dm = setpoint_dm_medido -
forca_dm_medida
315
316         if control_value_dm >= 0 and aciona_dm == 1:
317             forwardLeft_dm.value = (control_value_dm/
700)
318         else:
```

```

319         forwardLeft_dm.value = 0
320
321         if control_value_dm < 0 and aciona_dm == 1:
322             reverseLeft_dm.value = (abs(
control_value_dm/700))
323         else:
324             reverseLeft_dm.value = 0
325
326             control_dm = str(control_value_dm*0.01955) ##
converte para string para publicar no servidor
327             leitura_forca_dm_convertido = str(
leitura_forca_dm*0.01955)
328             error_forca_dm_convertido = str(
error_forca_dm*0.01955)
329             setoint_dm_medido= str(setpoint_dm_medido*0.
01955)
330
331
332             #print control_dm
333             client.publish("leitura_control_dm",
control_dm)
334             client.publish("leitura_sensor_dm",
leitura_forca_dm_convertido)
335             client.publish("leitura_error_dm",
error_forca_dm_convertido)
336             client.publish("leitura_setpoint_dm",
setpoint_dm_medido)
337
338
339             #####
340
341             ##Controle do Polegar
342             if habilita_manual_dp == 0:
343                 leitura_forca_dp = forca_sensor_dp.value *
1000
344
345                 forca_dp = np.arange(0, 1000, 1)
346                 setpoint_dp = np.arange(0, 1000, 1)
347                 control_dp = np.arange(-1000, 1000, 1)
348
349                 forca_dp_lo    = fuzz.trimf(forca_dp, [0, 0,
500])
350                 forca_dp_md    = fuzz.trimf(forca_dp, [0, 500,
1000])
351                 forca_dp_hi    = fuzz.trimf(forca_dp, [500,

```

```
351 1000, 1000])
352
353         setpoint_dp_lo = fuzz.trimf(setpoint_dp, [0,
354         0, 500])
355         setpoint_dp_md = fuzz.trimf(setpoint_dp, [0,
356         500, 1000])
357         setpoint_dp_hi = fuzz.trimf(setpoint_dp, [500
358         , 1000, 1000])
359
360         control_dp_lo = fuzz.trimf(control_dp, [-1000
361         , -1000, 0])
362         control_dp_md = fuzz.trimf(control_dp, [-1000
363         , 0, 1000])
364         control_dp_hi = fuzz.trimf(control_dp, [0,
365         1000, 1000])
366
367         forca_dp_medida = leitura_forca_dp
368         setpoint_dp_medido = 200
369
370         forca_dp_level_lo = fuzz.interp_membership(
371         forca_dp, forca_dp_lo, float(forca_dp_medida))
372         forca_dp_level_md = fuzz.interp_membership(
373         forca_dp, forca_dp_md, float(forca_dp_medida))
374         forca_dp_level_hi = fuzz.interp_membership(
375         forca_dp, forca_dp_hi, float(forca_dp_medida))
376
377         setpoint_dp_level_lo = fuzz.interp_membership
378         (setpoint_dp, setpoint_dp_lo, float(setpoint_dp_medido))
379         setpoint_dp_level_md = fuzz.interp_membership
380         (setpoint_dp, setpoint_dp_md, float(setpoint_dp_medido))
381         setpoint_dp_level_hi = fuzz.interp_membership
382         (setpoint_dp, setpoint_dp_hi, float(setpoint_dp_medido))
383
384         active_rule1_dp = np.fmin(forca_dp_level_lo,
385         setpoint_dp_level_md)
386         control_activation_1_dp = np.fmin(
387         active_rule1_dp, control_dp_hi)
388
389         active_rule2_dp = np.fmin(forca_dp_level_lo,
390         setpoint_dp_level_hi)
391         control_activation_2_dp = np.fmin(
392         active_rule2_dp, control_dp_hi)
393
394         active_rule3_dp = np.fmin(forca_dp_level_md,
395         setpoint_dp_level_lo)
```



```
379         control_activation_3_dp = np.fmin(
active_rule3_dp, control_dp_lo)
380
381         active_rule4_dp = np.fmin(forca_dp_level_md,
setpoint_dp_level_hi)
382         control_activation_4_dp = np.fmin(
active_rule4_dp, control_dp_hi)
383
384         active_rule5_dp = np.fmin(forca_dp_level_hi,
setpoint_dp_level_lo)
385         control_activation_5_dp = np.fmin(
active_rule5_dp, control_dp_lo)
386
387         active_rule6_dp = np.fmin(forca_dp_level_hi,
setpoint_dp_level_md)
388         control_activation_6_dp = np.fmin(
active_rule6_dp, control_dp_lo)
389
390         c1_dp = np.fmax(control_activation_1_dp,
control_activation_2_dp)
391         c2_dp = np.fmax(control_activation_3_dp,
control_activation_4_dp)
392         c3_dp = np.fmax(control_activation_5_dp,
control_activation_6_dp)
393         c4_dp = np.fmax(c2_dp, c3_dp)
394         aggregated_dp = np.fmax(c1_dp, c4_dp)
395
396         control_value_dp = fuzz.defuzz(control_dp,
aggregated_dp, 'centroid')
397
398 #         error_forca_dp = setpoint_dp_medido -
forca_dp_medida
399
400         if control_value_dp >= 0 and aciona_dp == 1:
401             forwardLeft_dp.value = (control_value_dp/
700)
402         else:
403             forwardLeft_dp.value = 0
404
405         if control_value_dp < 0 and aciona_dp == 1:
406             reverseLeft_dp.value = (abs(
control_value_dp/700))
407         else:
408             reverseLeft_dp.value = 0
409
```

```
410
411         control_dp = str(control_value_dp*0.01955) ##
           converte para string para publicar no servidor
412
413         ##print control_dp
414         client.publish("leitura_control_dp",
           control_dp)
415 #         client.publish("leitura_sensor_dp",
           leitura_forca_dp)
416 #         client.publish("leitura_error_dp",
           error_forca_dp)
417
418
419         ### Controle força dedo indicador##
420         if habilita_manual_di == 0:
421             leitura_forca_di = forca_sensor_di.value *
           1000
422             #print leitura_forca_di
423
424             forca_di = np.arange(0, 1000, 1)
425             setpoint_di = np.arange(0, 1000, 1)
426             control_di = np.arange(-1000, 1000, 1)
427
428             forca_di_lo   = fuzz.trimf(forca_di, [0, 0,
           500])
429             forca_di_md   = fuzz.trimf(forca_di, [0, 500,
           1000])
430             forca_di_hi   = fuzz.trimf(forca_di, [500,
           1000, 1000])
431
432             setpoint_di_lo = fuzz.trimf(setpoint_di, [0,
           0, 500])
433             setpoint_di_md = fuzz.trimf(setpoint_di, [0,
           500, 1000])
434             setpoint_di_hi = fuzz.trimf(setpoint_di, [500
           , 1000, 1000])
435
436             control_di_lo = fuzz.trimf(control_di, [-1000
           , -1000, 0])
437             control_di_md = fuzz.trimf(control_di, [-1000
           , 0, 1000])
438             control_di_hi = fuzz.trimf(control_di, [0,
           1000, 1000])
439
440             forca_di_medida = leitura_forca_di
```

```
441         setpoint_di_medido = 200
442
443         forca_di_level_lo = fuzz.interp_membership(
444             forca_di, forca_di_lo, float(forca_di_medida))
445         forca_di_level_md = fuzz.interp_membership(
446             forca_di, forca_di_md, float(forca_di_medida))
447         forca_di_level_hi = fuzz.interp_membership(
448             forca_di, forca_di_hi, float(forca_di_medida))
449
450         setpoint_di_level_lo = fuzz.interp_membership
451             (setpoint_di, setpoint_di_lo, float(setpoint_di_medido))
452         setpoint_di_level_md = fuzz.interp_membership
453             (setpoint_di, setpoint_di_md, float(setpoint_di_medido))
454         setpoint_di_level_hi = fuzz.interp_membership
455             (setpoint_di, setpoint_di_hi, float(setpoint_di_medido))
456
457         active_rule1_di = np.fmin(forca_di_level_lo,
458             setpoint_di_level_md)
459         control_activation_1_di = np.fmin(
460             active_rule1_di, control_di_hi)
461
462         active_rule2_di = np.fmin(forca_di_level_lo,
463             setpoint_di_level_hi)
464         control_activation_2_di = np.fmin(
465             active_rule2_di, control_di_hi)
466
467         active_rule3_di = np.fmin(forca_di_level_md,
468             setpoint_di_level_lo)
469         control_activation_3_di = np.fmin(
470             active_rule3_di, control_di_lo)
471
472         active_rule4_di = np.fmin(forca_di_level_md,
473             setpoint_di_level_hi)
474         control_activation_4_di = np.fmin(
475             active_rule4_di, control_di_hi)
476
477         active_rule5_di = np.fmin(forca_di_level_hi,
478             setpoint_di_level_lo)
479         control_activation_5_di = np.fmin(
480             active_rule5_di, control_di_lo)
481
482         active_rule6_di = np.fmin(forca_di_level_hi,
483             setpoint_di_level_md)
484         control_activation_6_di = np.fmin(
485             active_rule6_di, control_di_lo)
```

```
468
469         c1_di = np.fmax(control_activation_1_di,
control_activation_2_di)
470         c2_di = np.fmax(control_activation_3_di,
control_activation_4_di)
471         c3_di = np.fmax(control_activation_5_di,
control_activation_6_di)
472         c4_di = np.fmax(c2_di,c3_di)
473         aggregated_di = np.fmax(c1_di,c4_di)
474
475         control_value_di = fuzz.defuzz(control_di,
aggregated_di, 'centroid')
476
477 #         error_forca_di = setpoint_di_medido -
forca_di_medida
478
479         if control_value_di >= 0 and aciona_di == 1:
480             forwardLeft_di.value = (control_value_di/
700)
481         else:
482             forwardLeft_di.value = 0
483
484         if control_value_di < 0 and aciona_di == 1:
485             reverseLeft_di.value = (abs(
control_value_di/700))
486         else:
487             reverseLeft_di.value = 0
488
489         control_di = str(control_value_di*0.01955) ##
converte para string para publicar no servidor
490         #print control_di
491
492         client.publish("leitura_control_di",
control_di)
493 #         client.publish("leitura_sensor_di",
leitura_forca_di)
494 #         client.publish("leitura_error_di",
error_forca_di)
495
496
497         if habilita_manual_da == 0:
498             leitura_forca_da = forca_sensor_da.value *
1000
499             #print leitura_forca_di
500
```

```
501         forca_da = np.arange(0, 1000, 1)
502         setpoint_da = np.arange(0, 1000, 1)
503         control_da = np.arange(-1000, 1000, 1)
504
505         forca_da_lo    = fuzz.trimf(forca_da, [0, 0,
506         500])
507         forca_da_md    = fuzz.trimf(forca_da, [0, 500,
508         1000])
509         forca_da_hi    = fuzz.trimf(forca_da, [500,
510         1000, 1000])
511
512         setpoint_da_lo = fuzz.trimf(setpoint_da, [0,
513         0, 500])
514         setpoint_da_md = fuzz.trimf(setpoint_da, [0,
515         500, 1000])
516         setpoint_da_hi = fuzz.trimf(setpoint_da, [500
517         , 1000, 1000])
518
519         control_da_lo = fuzz.trimf(control_da, [-1000
520         , -1000, 0])
521         control_da_md = fuzz.trimf(control_da, [-1000
522         , 0, 1000])
523         control_da_hi = fuzz.trimf(control_da, [0,
524         1000, 1000])
525
526         forca_da_medida = leitura_forca_da
527         setpoint_da_medido = 200
528
529         forca_da_level_lo = fuzz.interp_membership(
530         forca_da, forca_da_lo, float(forca_da_medida))
531         forca_da_level_md = fuzz.interp_membership(
532         forca_da, forca_da_md, float(forca_da_medida))
533         forca_da_level_hi = fuzz.interp_membership(
534         forca_da, forca_da_hi, float(forca_da_medida))
535
536         setpoint_da_level_lo = fuzz.interp_membership
537         (setpoint_da, setpoint_da_lo, float(setpoint_da_medido))
538         setpoint_da_level_md = fuzz.interp_membership
539         (setpoint_da, setpoint_da_md, float(setpoint_da_medido))
540         setpoint_da_level_hi = fuzz.interp_membership
541         (setpoint_da, setpoint_da_hi, float(setpoint_da_medido))
542
543         active_rule1_da = np.fmin(forca_da_level_lo,
544         setpoint_da_level_md)
545         control_activation_1_da = np.fmin(
```

```
529 active_rule1_da, control_da_hi)
530
531         active_rule2_da = np.fmin(forca_da_level_lo,
532         setpoint_da_level_hi)
533         control_activation_2_da = np.fmin(
534         active_rule2_da, control_da_hi)
535
536         active_rule3_da = np.fmin(forca_da_level_md,
537         setpoint_da_level_lo)
538         control_activation_3_da = np.fmin(
539         active_rule3_da, control_da_lo)
540
541         active_rule4_da = np.fmin(forca_da_level_md,
542         setpoint_da_level_hi)
543         control_activation_4_da = np.fmin(
544         active_rule4_da, control_da_hi)
545
546         active_rule5_da = np.fmin(forca_da_level_hi,
547         setpoint_da_level_lo)
548         control_activation_5_da = np.fmin(
549         active_rule5_da, control_da_lo)
550
551         active_rule6_da = np.fmin(forca_da_level_hi,
552         setpoint_da_level_md)
553         control_activation_6_da = np.fmin(
554         active_rule6_da, control_da_lo)
555
556         c1_da = np.fmax(control_activation_1_da,
557         control_activation_2_da)
558         c2_da = np.fmax(control_activation_3_da,
559         control_activation_4_da)
560         c3_da = np.fmax(control_activation_5_da,
561         control_activation_6_da)
562         c4_da = np.fmax(c2_da, c3_da)
563         aggregated_da = np.fmax(c1_da, c4_da)
564
565         control_value_da = fuzz.defuzz(control_da,
566         aggregated_da, 'centroid')
567
568         #         error_forca_da = setpoint_da_medido -
569         forca_da_medida
570
571         if control_value_da >= 0 and aciona_da == 1:
572             forwardLeft_da.value = (control_value_da/
573             700)
```

```
558         else:
559             forwardLeft_da.value = 0
560
561         if control_value_da < 0 and aciona_da == 1:
562             reverseLeft_da.value = (abs(
control_value_da/700))
563         else:
564             reverseLeft_da.value = 0
565
566             control_da = str(control_value_da*0.01955) ##
converte para string para publicar no servidor
567             #print control_da
568
569             client.publish("leitura_control_da",
control_da)
570 #             client.publish("leitura_sensor_da",
leitura_forca_da)
571 #             client.publish("leitura_error_da",
error_forca_da)
572
573         if habilita_manual_dmi == 0:
574             leitura_forca_dmi = forca_sensor_dmi.value *
1000
575             #print leitura_forca_di
576
577             forca_dmi = np.arange(0, 1000, 1)
578             setpoint_dmi = np.arange(0, 1000, 1)
579             control_dmi = np.arange(-1000, 1000, 1)
580
581             forca_dmi_lo    = fuzz.trimf(forca_dmi, [0, 0,
500])
582             forca_dmi_md    = fuzz.trimf(forca_dmi, [0,
500, 1000])
583             forca_dmi_hi    = fuzz.trimf(forca_dmi, [500,
1000, 1000])
584
585             setpoint_dmi_lo = fuzz.trimf(setpoint_dmi, [0
, 0, 500])
586             setpoint_dmi_md = fuzz.trimf(setpoint_dmi, [0
, 500, 1000])
587             setpoint_dmi_hi = fuzz.trimf(setpoint_dmi, [
500, 1000, 1000])
588
589             control_dmi_lo = fuzz.trimf(control_dmi, [-
1000, -1000, 0])
```

```
590         control_dmi_md = fuzz.trimf(control_dmi, [-
1000, 0, 1000])
591         control_dmi_hi = fuzz.trimf(control_dmi, [0,
1000, 1000])
592
593         forca_dmi_medida = leitura_forca_dmi
594         setpoint_dmi_medido = 200
595
596         forca_dmi_level_lo = fuzz.interp_membership(
forca_dmi, forca_dmi_lo, float(forca_dmi_medida))
597         forca_dmi_level_md = fuzz.interp_membership(
forca_dmi, forca_dmi_md, float(forca_dmi_medida))
598         forca_dmi_level_hi = fuzz.interp_membership(
forca_dmi, forca_dmi_hi, float(forca_dmi_medida))
599
600         setpoint_dmi_level_lo = fuzz.
interp_membership(setpoint_dmi, setpoint_dmi_lo, float(
setpoint_dmi_medido))
601         setpoint_dmi_level_md = fuzz.
interp_membership(setpoint_dmi, setpoint_dmi_md, float(
setpoint_dmi_medido))
602         setpoint_dmi_level_hi = fuzz.
interp_membership(setpoint_dmi, setpoint_dmi_hi, float(
setpoint_dmi_medido))
603
604         active_rule1_dmi = np.fmin(forca_dmi_level_lo
, setpoint_dmi_level_md)
605         control_activation_1_dmi = np.fmin(
active_rule1_dmi, control_dmi_hi)
606
607         active_rule2_dmi = np.fmin(forca_dmi_level_lo
, setpoint_dmi_level_hi)
608         control_activation_2_dmi = np.fmin(
active_rule2_dmi, control_dmi_hi)
609
610         active_rule3_dmi = np.fmin(forca_dmi_level_md
, setpoint_dmi_level_lo)
611         control_activation_3_dmi = np.fmin(
active_rule3_dmi, control_dmi_lo)
612
613         active_rule4_dmi = np.fmin(forca_dmi_level_md
, setpoint_dmi_level_hi)
614         control_activation_4_dmi = np.fmin(
active_rule4_dmi, control_dmi_hi)
615
```



```
616         active_rule5_dmi = np.fmin(forca_dmi_level_hi
, setpoint_dmi_level_lo)
617         control_activation_5_dmi = np.fmin(
active_rule5_dmi, control_dmi_lo)
618
619         active_rule6_dmi = np.fmin(forca_dmi_level_hi
, setpoint_dmi_level_md)
620         control_activation_6_dmi = np.fmin(
active_rule6_dmi, control_dmi_lo)
621
622         c1_dmi = np.fmax(control_activation_1_dmi,
control_activation_2_dmi)
623         c2_dmi = np.fmax(control_activation_3_dmi,
control_activation_4_dmi)
624         c3_dmi = np.fmax(control_activation_5_dmi,
control_activation_6_dmi)
625         c4_dmi = np.fmax(c2_da, c3_dmi)
626         aggregated_dmi = np.fmax(c1_dmi, c4_dmi)
627
628         control_value_dmi = fuzz.defuzz(control_dmi,
aggregated_dmi, 'centroid')
629
630 #         error_forca_dmi = setpoint_dmi_medido -
forca_dmi_medida
631
632         if control_value_dmi >= 0 and aciona_dmi == 1
:
633             forwardLeft_dmi.value = (
control_value_dmi/700)
634         else:
635             forwardLeft_dmi.value = 0
636
637         if control_value_dmi < 0 and aciona_dmi == 1:
638             reverseLeft_dmi.value = (abs(
control_value_dmi/700))
639         else:
640             reverseLeft_dmi.value = 0
641
642         control_dmi = str(control_value_dmi*0.01955)
##converte para string para publicar no servidor
643         #print control_dmi
644
645         client.publish("leitura_control_dmi",
control_dmi)
646 #         client.publish("leitura_sensor_dmi",
```

```
646 leitura_forca_dmi)
647 #         client.publish("leitura_error_dmi",
        error_forca_dmi)
648
649
650
651
652         ### tempo de scann
653         time.sleep(0.05)
654
655 except KeyboardInterrupt:
656     print "\nScript finalizado."
657     sys.exit(0)
658
```

B. LISTA COMPLETA DE MATERIAIS UTILIZADOS

A lista completo com a descrição e materiais utilizados para confecção da garra robótica antropomórfica pode ser visualizado na tabela abaixo.

Item	Nome da Peça	Descrição	Qtd
1	suporte_de_madeira	peça em mdf	1
2	suporte_base		1
3	suporte_fixação_eletrônica		1
4	fonte_250w	peça em mdf	1
5	protoboard		1
6	driver l298N		3
7	raspberry pi3		1
8	conversor_dc_dc-lm2596		1
9	base_da_palma	impressão 3d	1
10	suporte_motores_dc	suporte para motores dc, impressão 3D	4
11	motor_dc	motores dc 12vdc com caixa de redução	4
12	carretel_extensor_nylon	carretel para extensionar tendão, impressão 3d	5
13	extensor_nylon	extensor para manter o tendão sempre extensionado, impresão 3d	8
14	eixo 3x16mm		4
15	rolamento v623zz		8
16	eixo 3x6.2mm		10
17	mola		10
18	dedo_junta_1_inferior	primeira junta dos dedos, sendo comum ao dedão, parte superior	5
19	dedo_junta_1_superior	primeira junta dos dedos, parte inferior	5
20	rolamento 623zz		16
21	eixo 3x10mm		10
22	eixo 3x19mm		6
23	dedo_junta_2_inferior	segunda junta dos dedos, parte inferior, impressão 3d	4
24	dedo_junta_2_superior	segunda junta dos dedos, parte inferior, impressão 3d	4

25	dedo_junta_3_inferior	impressão 3d	4
26	dedo_junta_3_superior	impressão 3d	4
27	parafuso m2x6mm		29
28	porca m2		19
29	dorso_da_palma	impressão 3d	1
30	suporte_motor_dc_dedão_1	impressão 3d	1
31	suporte_motor_dc_dedão_2	impressão 3d	1
32	micro motor SPG10Motor		1
33	suporte_do_dedão	impressão 3d	1
34	servo 9g		1
35	servo_haste		1
36	dedo_junta_2_dedao_inferior	impressão 3d	1
37	dedo_junta_2_dedão_superior	impressão 3d	1
38	extensor_nylon_dedao	impressão 3d	2
39	flange	impressão 3d	1

C. ALGORITMO DA CINEMÁTICA NO MATLAB

```
%% Função da matriz DH
%%
function [DH] = DH_function (alpha_i, l_i, d_i, theta_i)

DH = [cos(theta_i) -sin(theta_i)*cos(alpha_i) sin(theta_i)*sin(alpha_i)
      l_i*cos(theta_i);
      sin(theta_i) cos(theta_i)*cos(alpha_i) -cos(theta_i)*sin(alpha_i)
      l_i*sin(theta_i);
      0 sin(alpha_i) cos(alpha_i) d_i;
      0 0 0 1];

end

%% Cinemática Direta da Mão Robótica (Quatro dedos)
%% Universidade de Brasília
%%
close all
clear all
clc

%% Declaração dos parâmetros cinemáticos
%%

alpha_1 = 0;
alpha_2 = 0;
alpha_3 = 0;

l_1 = 40.60;
l_2 = 36.52;
l_3 = 27.05;

d_1 = 0;
d_2 = 0;
d_3 = 0;

%% Parâmetros de entrada
%%
theta_1 = input ('Entre com o valor de theta1 em graus = \n');
theta_2 = input ('Entre com o valor de theta2 em graus = \n');
theta_3 = input ('Entre com o valor de theta3 em graus = \n');

%% Cálculo das matrizes de transformação
%%
T01 = DH_function (alpha_1, l_1, d_1, theta_1*pi/180);
T12 = DH_function (alpha_2, l_2, d_2, theta_2*pi/180);
T23 = DH_function (alpha_3, l_3, d_3, theta_3*pi/180);

T02 = T01*T12;
```

```

T03 = T02*T23;

%% Posição do elemento terminal na junta 6
%%
x = T03(1,4);
y = T03(2,4);
z = T03(3,4);
P_xyz = [x; y; z]

%% Cinemática Direta da Mão Robótica (Polegar)
%% Universidade de Brasília
%%
close all
clear all
clc

%% Declaração dos parâmetros cinemáticos
%%

alpha_1 = pi/2;
alpha_2 = 0;
alpha_3 = 0;

l_1 = 33.26;
l_2 = 39.5;
l_3 = 26.97;

d_1 = 29.35;
d_2 = 0;
d_3 = 0;

%% Parâmetros de entrada
%%
theta_1 = input('Entre com o valor de theta1 em graus = \n');
theta_2 = input('Entre com o valor de theta2 em graus = \n');
theta_3 = input('Entre com o valor de theta3 em graus = \n');

%% Cálculo das matrizes de transformação
%%
T01 = DH_function(alpha_1, l_1, d_1, theta_1*pi/180);
T12 = DH_function(alpha_2, l_2, d_2, theta_2*pi/180);
T23 = DH_function(alpha_3, l_3, d_3, theta_3*pi/180);

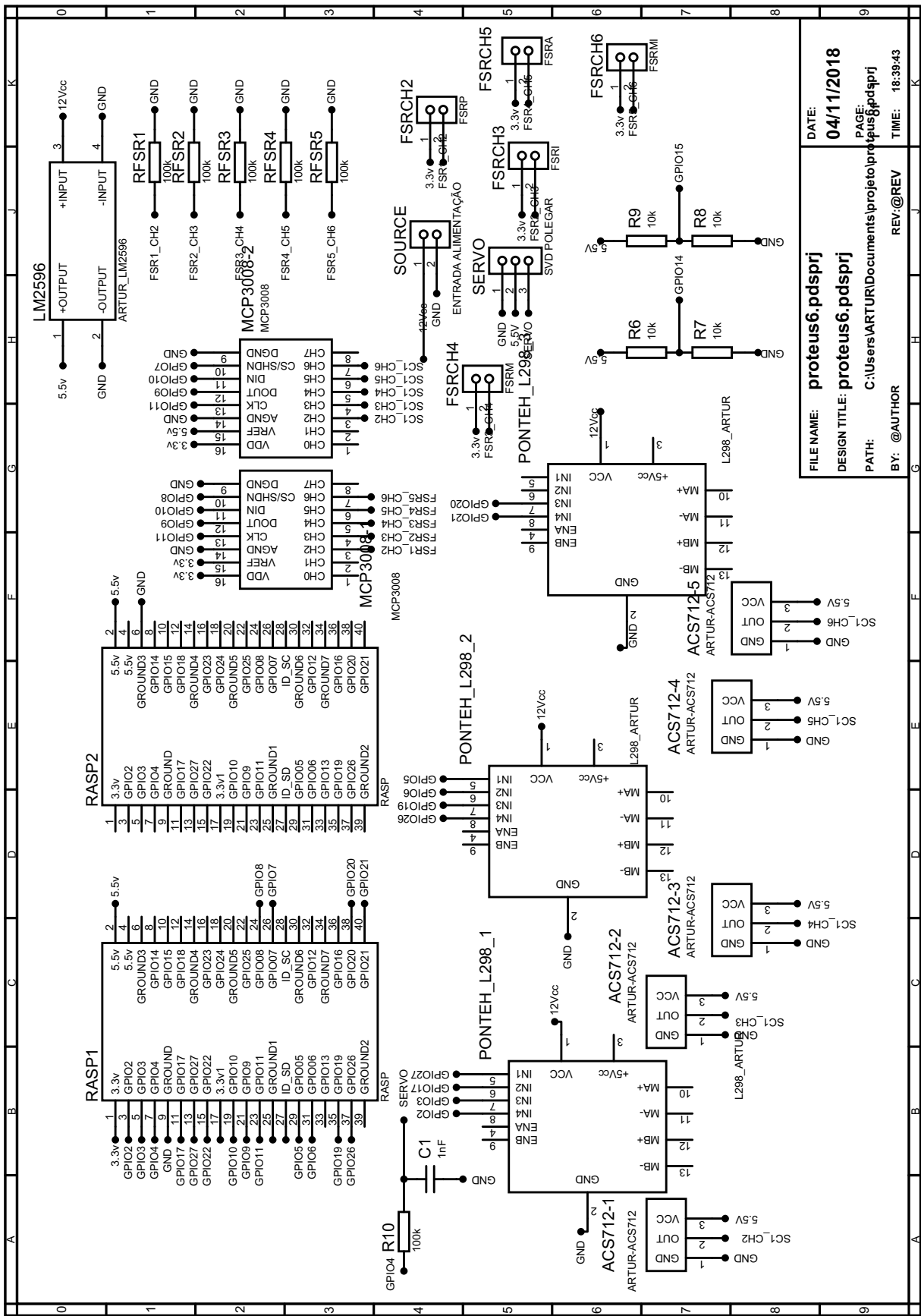
T02 = T01*T12;
T03 = T02*T23;

%% Posição do elemento terminal na junta 6
%%
x = T03(1,4);
y = T03(2,4);

```

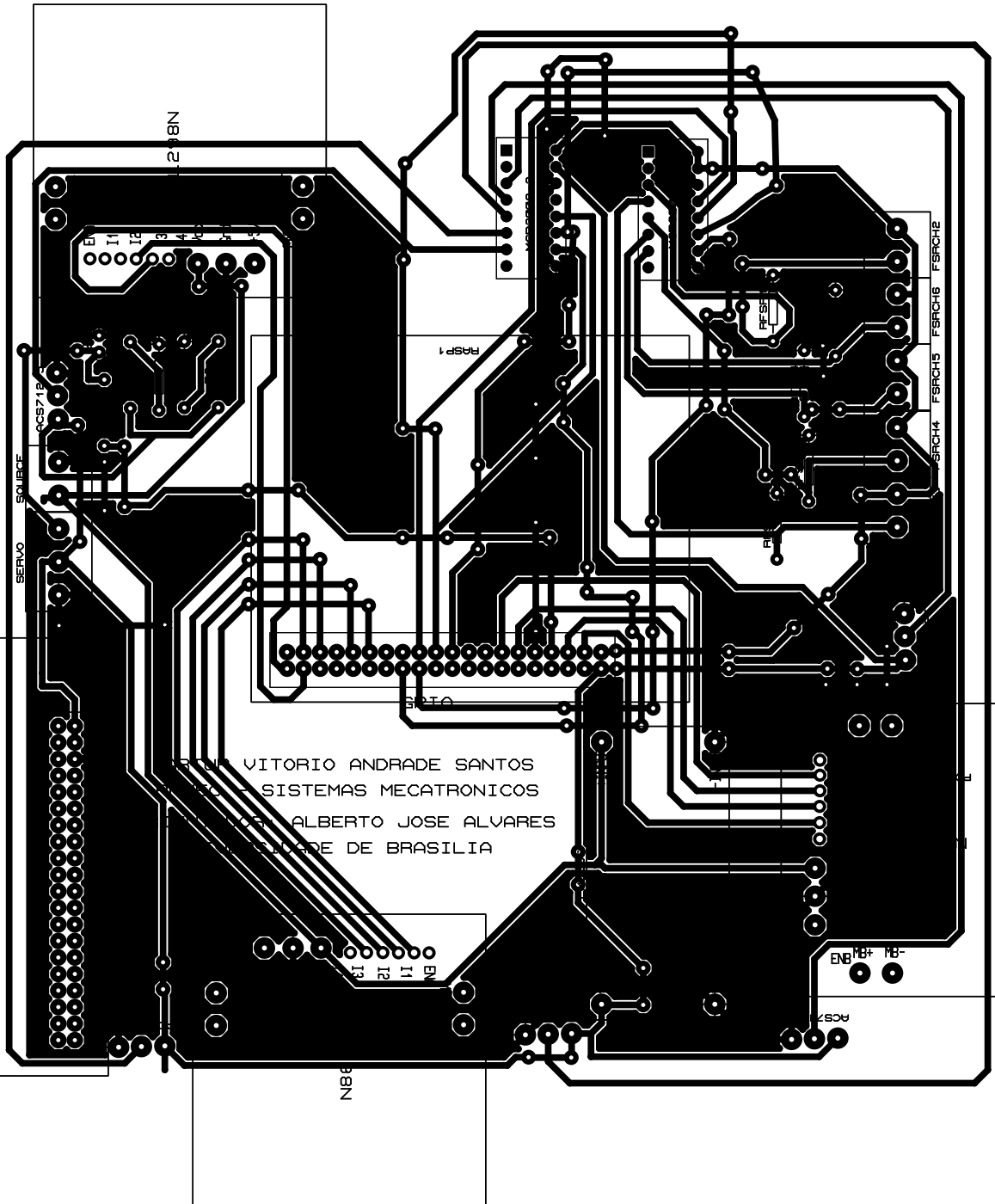
```
z = T03(3,4);  
P_xyz = [x; y; z]
```

D. PROJETO ELETRÔNICO

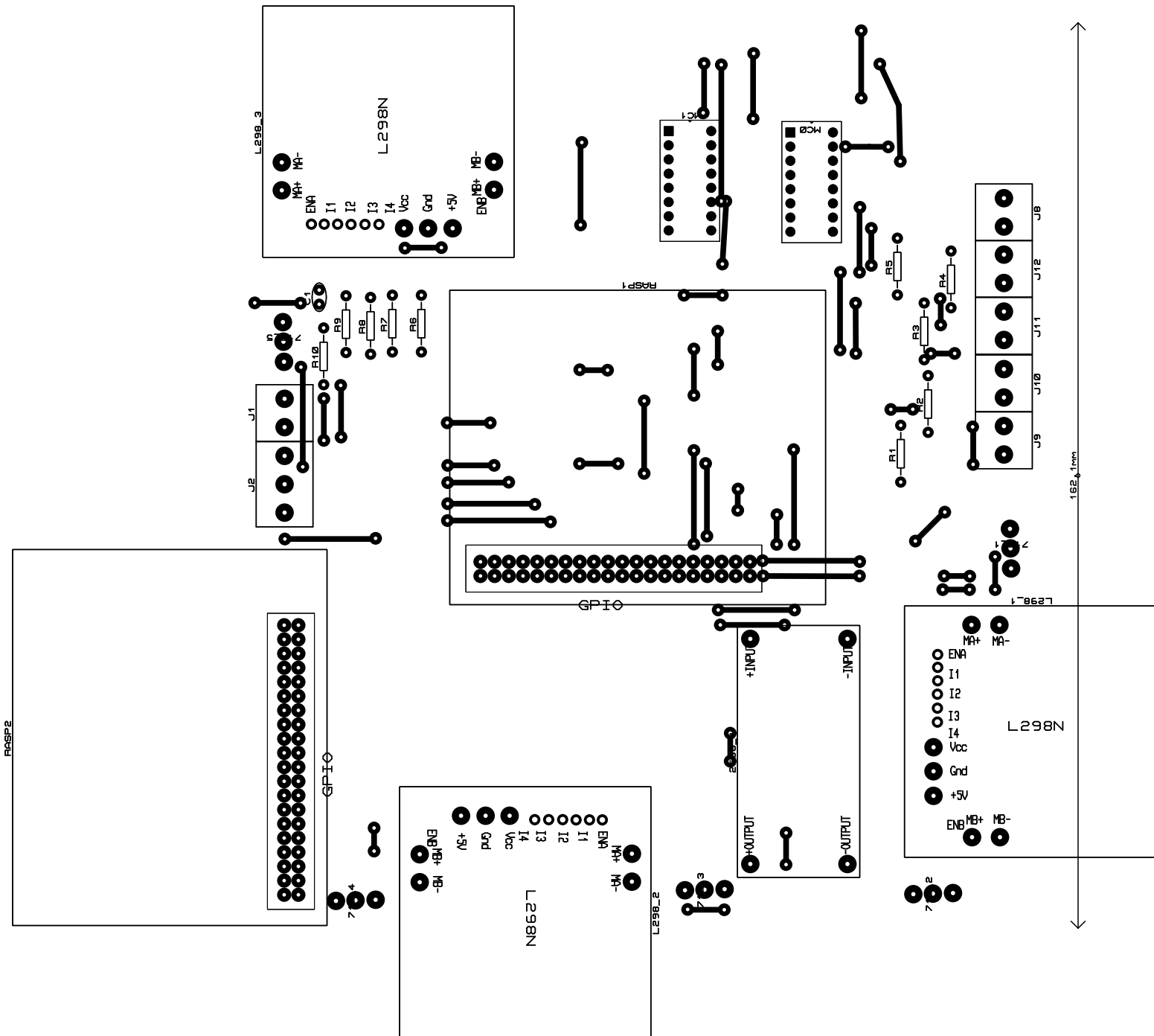


FILE NAME: **proteus6.pdsprj**
 DESIGN TITLE: **proteus6.pdsprj**
 PATH: C:\Users\ARTUR\Documents\projeto\proteus6.pdsprj
 BY: @AUTHOR
 REV: @REV
 DATE: **04/11/2018**
 PAGE: proteus6.pdsprj
 TIME: 18:39:43

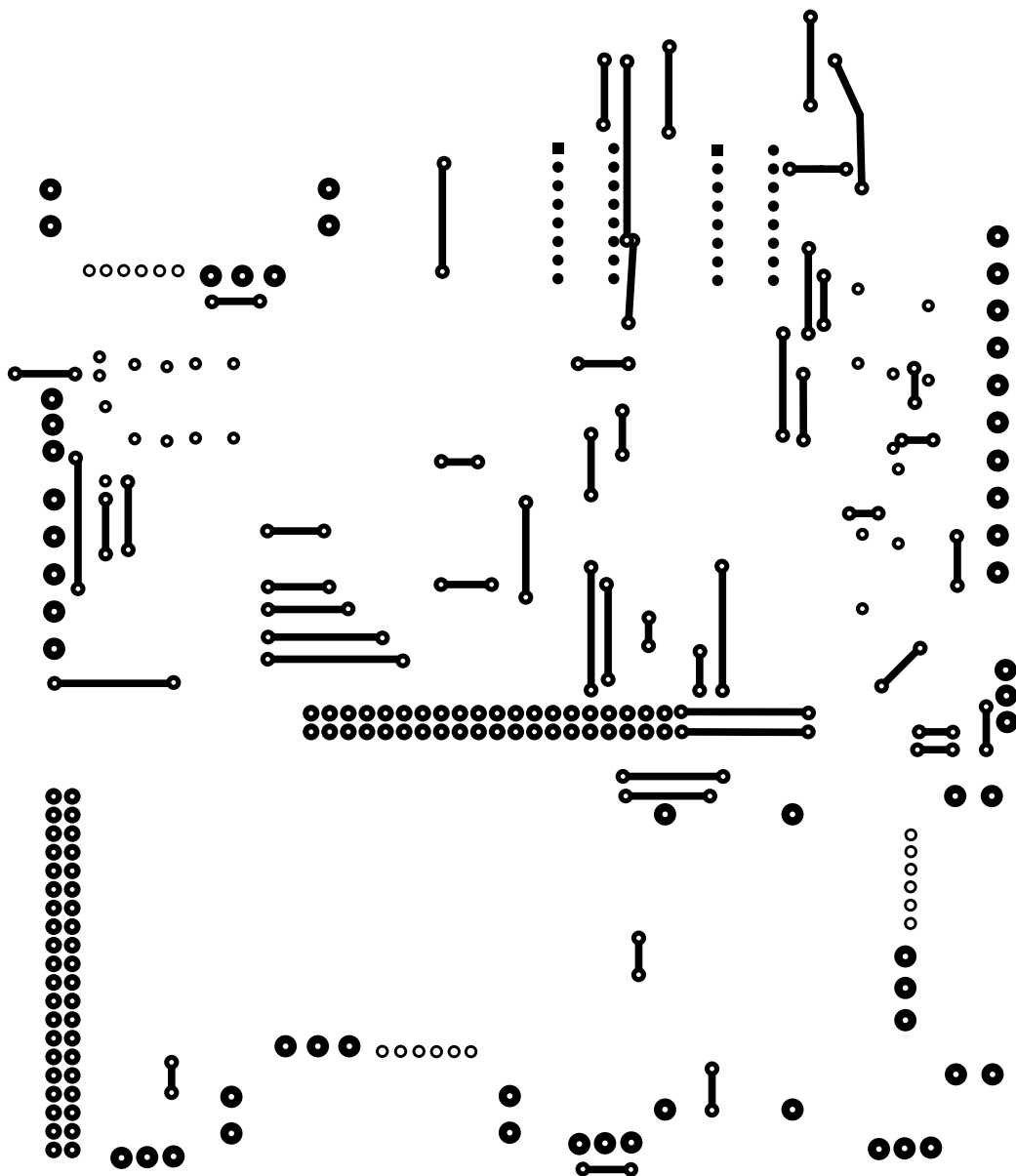
D.0.1 Layout Completo da Placa Eletrônica



D.0.2 Legenda de Componentes



D.0.3 Furos



E. MATERIAL COMPARTILHADO

Todo material compartilhado pode ser encontrado no site: www.arturvitorio.weebly.com