



Beatriz
Marques

**Desenvolvimento e Operacionalização de uma
Mão Robótica Antropomórfica**

**Development and Operationalization of an
Anthropomorphic Robotic Hand**



Beatriz
Marques

**Desenvolvimento e Operacionalização de uma
Mão Robótica Antropomórfica**

**Development and Operationalization of an
Anthropomorphic Robotic Hand**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Robótica e Sistemas Inteligentes, realizada sob a orientação científica do Doutor Vítor Manuel Ferreira dos Santos, Professor associado c/ agregação do Departamento de Engenharia Mecânica da Universidade de Aveiro, do Doutor Filipe Miguel Teixeira Pereira da Silva, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

palavras-chave

mão robótica antropomórfica,sensores de contacto, ROS2

resumo

A robótica tem vindo a assumir um papel cada vez mais relevante no quotidiano humano, sendo aplicada em domínios como a indústria, medicina, e assistência pessoal. No entanto, muitos sistemas robóticos continuam a ser projetados para executar tarefas específicas, apresentando dificuldades na adaptação a contextos dinâmicos. Neste contexto, a percepção sensorial e a capacidade de adaptação tornam-se fundamentais para aumentar a robustez e flexibilidade dos sistemas robóticos. Este trabalho apresenta o desenvolvimento de uma mão robótica antropomórfica equipada com sensores táteis para deteção de contacto. O projeto incluiu a montagem da estrutura mecânica, a programação e controlo dos motores, a integração da eletrónica sensorial e o desenvolvimento de um sistema de aquisição de dados. Através dos dados recolhidos em diferentes cenários de interação, foi treinado um modelo de *machine learning* capaz de detetar e classificar colisões entre os dedos, contribuindo para a percepção do sistema e abrindo caminho a futuros mecanismos de controlo mais inteligentes.

keywords

abstract

Robotics has been playing an increasingly important role in everyday human life, with applications in areas such as industry, healthcare, and personal assistance. However, many robotic systems are still designed to perform specific tasks, showing limited adaptability to dynamic and unstructured environments. In this context, sensory perception and adaptability become key to enhancing the robustness and flexibility of robotic systems. This work presents the development of an anthropomorphic robotic hand equipped with tactile sensors for contact detection. The project involved the assembly of the mechanical structure, programming and control of the motors, integration of the sensory electronics, and development of a data acquisition system. Based on the data collected in different interaction scenarios, a machine learning model was trained to detect and classify collisions between fingers, contributing to the system's perception and paving the way for more intelligent control mechanisms in the future.

**acknowledgement of use of
AI tools**

**Recognition of the use of generative Artificial Intelligence
technologies and tools, software and other support tools.**

Reconheço a utilização do ChatGPT 4.0 (Open AI, <https://chat.openai.com>) para melhoria e auxílio na revisão da escrita da dissertação.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Siglas e Acrónimos	vi
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	1
1.3 Estrutura do documento	1
2 Trabalhos relacionados	2
2.1 Introdução	2
2.2 Mãos Robóticas Antropomórficas	2
2.2.1 Soluções comerciais	3
2.2.2 Soluções <i>open-source</i>	4
2.2.3 Seleção da mão a construir	8
2.3 Sensores de Contacto	8
2.3.1 Sensores baseados em Tecnologia Magnética	9
2.3.2 Sensores baseados em Polímeros Condutoivos	10
2.3.3 Sensores baseados em Materiais Piezoresistivos	11
2.3.4 Escolha dos sensores a incorporar no projeto	11
2.4 Conclusão	12
3 Ferramentas de suporte	14
3.1 Introdução	14
3.2 Hardware	14
3.2.1 Motores Dynamixel	14
3.2.2 Controlador e Fonte de Alimentação	15

3.3	Software	16
3.3.1	Dynamixel Wizard 2.0	16
3.3.2	Biblioteca Dynamixel SDK	17
3.3.3	ROS2	18
3.3.4	Documentação da LEAP Hand	18
3.4	Conclusão	19
4	Projeto e Implementação da Solução	20
4.1	Introdução	20
4.2	Desenvolvimento da Mão Robótica	20
4.2.1	Fabrico e Montagem da estrutura	20
4.2.2	Configuração dos Motores	22
4.2.3	Programação e Controlo	24
4.2.4	Arquitetura de Software Desenvolvida	30
4.2.5	Funcionalidades Avançadas	30
4.3	Implementação dos Sensores de Contacto	30
4.3.1	Seleção do Microcontrolador	30
4.3.2	Implementação Física do Sistema sensorial	30
4.3.3	Testes com sensores	30
4.4	Conclusão	30
Referências		31
A	Apêndice	33
A.1	Montagem dos Dedos	33
A.1.1	Montagem do Polegar	33
A.1.2	Montagem dos restantes dedos	34
A.1.3	Hiperligações	35
A.2	Testes realizados com um dedo	35
A.2.1	Teste com duas velocidades diferentes	35
A.2.2	Teste com obstáculo em posições diferentes	36

Lista de Figuras

2.1	OceanTrix Hand desenvolvida pela OceanTrix Robotics[1].	4
2.2	Agile Hand desenvolvida pela Agile Robots SE [2].	4
2.3	Mão HRI e seu mecanismo	5
2.4	Robot Nano Hand desenvolvida por The Robot Studio[4].	6
2.5	Alaris Hand desenvolvida por Nurpeissova, et al.[5]	7
2.6	Leap Hand desenvolvida por Shaw, et al.[6]	7
2.7	Sensor Biotac desenvolvido pela SynTouch [7].	9
2.8	Sensor com filme magnético flexível desenvolvido por Jamone, et al. [8].	9
2.9	Composição do sensor desenvolvido por Kawasetsu et al. [10].	10
2.10	Princípio de operação do sensor baseado em materiais magnetoresistivos [11].	10
3.1	Motor Dynamixel XC330-M288-T utilizado na construção da LEAP Hand[6]	15
3.2	Controlador Dynamixel U2D2 utilizado para controlar os motores do projeto	16
3.3	Fonte de Alimentação industrial com saída a 5V e capacidade de 30A utilizada neste projeto	16
3.4	Interface Gráfica da ferramenta Dynamixel Wizard 2.0 utilizada para alterar o ID de cada motor	17
4.1	Polegar construído de acordo com a Documentação da Leap Hand	21
4.2	Dedo construído de acordo com a documentação da Leap Hand correspondente à estrutura do dedo indicador, médio e anelar	21
4.3	Mão robótica completamente montada, seguindo a estrutura proposta na documentação da LEAP Hand.	22
4.4	Posições dos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução	27
4.5	Velocidades dos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução	27
4.6	Correntes consumidas pelos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução	27
4.7	Posições dos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo	28
4.8	Velocidades dos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo	28

4.9	Correntes consumidas pelos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo	29
A.1	Fotografias sequenciais da montagem do polegar	33
A.2	Fotografias sequenciais da montagem do dedo referente ao dedo indicador, médio ou anelar	34
A.3	Gráficos da posição, velocidade e corrente consumida pelos motores de um dedo. A coluna da esquerda corresponde a uma velocidade máxima configurada de 1.2 rad/s e a coluna da direita a 23.98 rad/s.	35
A.4	Gráficos da posição, velocidade e corrente consumida pelos motores de um dedo, em diferentes cenários de obstrução. A coluna da esquerda representa a presença de um obstáculo junto ao motor mais próximo da base do dedo, a coluna central refere-se à obstrução a meio do dedo, e a coluna da direita corresponde à interferência na ponta do dedo.	36

Lista de Tabelas

2.1	Comparação entre diferentes mãos robóticas open-source	8
2.2	Comparação entre diferentes tipos de sensores táteis	12
4.1	Resumo dos modos de operação dos motores Dynamixel	24

Listas de Siglas e Acrónimos

ROS	Robot Operating System	FSR	Force Sensing Resistor
PIP	Proximal Interphalangeal Joint	PVDF	Polyvinylidene fluoride
DIP	Distal Interphalangeal Joint		
MCP	Metacarpophalangeal Joint	DDS	Data Distribution Service

CAPÍTULO

1

Introdução

1.1 ENQUADRAMENTO

-

1.2 OBJETIVOS

1.3 ESTRUTURA DO DOCUMENTO

CAPÍTULO 2

Trabalhos relacionados

2.1 INTRODUÇÃO

A área da manipulação robótica apresenta inúmeros desafios, particularmente em tarefas que envolvem preensão precisa e troca física de objetos, tanto entre robôs como entre robôs e seres humanos. Nestes contextos, a configuração e natureza da extremidade manipuladora desempenham um papel crucial na eficácia da interação. Para alcançar níveis mais elevados de destreza, flexibilidade e adaptabilidade, o uso de mãos robóticas antropomórficas tem-se revelado uma abordagem promissora. No entanto, estas soluções implicam desafios significativos, desde a conceção e fabrico até à complexidade do controlo, devido ao elevado número de graus de liberdade e à redundância cinemática associada.

Paralelamente, a integração de sensores de contacto é essencial para dotar as mãos robóticas de percepção tátil, permitindo-lhes detetar interações físicas com o ambiente ou entre os próprios dedos. Esta capacidade é fundamental para melhorar o controlo em tempo real, garantir uma manipulação mais segura e possibilitar a implementação de estratégias adaptativas baseadas na percepção sensorial.

Este capítulo apresenta uma análise de diferentes mãos robóticas antropomórficas, incluindo soluções comerciais e plataformas *open-source*, com o objetivo de fundamentar a seleção do modelo adotado neste projeto. Adicionalmente, é feita uma revisão das principais tecnologias de sensores de contacto, culminando na escolha dos mais adequados para integração na mão desenvolvida.

2.2 MÃOS ROBÓTICAS ANTROPOMÓRFICAS

O desenvolvimento de mãos robóticas antropomórficas tem sido alvo de grande interesse na comunidade científica e industrial, devido ao seu potencial para executar tarefas complexas de manipulação com maior destreza e versatilidade. Estas mãos procuram reproduzir, de forma funcional, algumas das capacidades da mão humana, beneficiando de um elevado número de graus de liberdade e de uma estrutura mecânica inspirada na anatomia biológica.

Esta secção apresenta e analisa diferentes soluções de mãos robóticas antropomórficas existentes, tanto comerciais como *open-source*, com o objetivo de identificar características relevantes para o projeto em desenvolvimento.

2.2.1 Soluções comerciais

Existem várias mãos robóticas antropomórficas disponíveis comercialmente, desenvolvidas com foco na destreza, robustez e integração em ambientes industriais ou de investigação. Estas soluções oferecem alto desempenho, mas geralmente implicam custos elevados, com valores que podem ultrapassar os 10 000€, o que as torna menos acessíveis para projetos académicos ou de prototipagem. Nesta subsecção são apresentadas algumas das opções disponíveis no mercado, como a Oceantrix Hand e a Agile Hand.

OceanTrix Hand

A OceanTrix Dexterous Hand (Figura 2.1) é uma mão robótica de sétima geração que apresenta uma arquitetura mecânica com 6 graus de liberdade (DOF), permitindo movimentos coordenados e versáteis dos dedos, adequados para tarefas complexas de manipulação.

O sistema de atuação é totalmente elétrico, utilizando motores de alto desempenho encapsulados em componentes metálicos de grau aeronáutico (alumínio AL6061), o que assegura robustez estrutural e resistência a ambientes exigentes.

A mão incorpora 13 sensores táteis distribuídos estratégicamente nos dedos, proporcionando *feedback* sensorial detalhado que permite um controlo em malha fechada e uma interação sensível com objetos de diferentes formas e texturas. No entanto, esta característica é opcional.

Em termos de integração, a OceanTrix Dexterous Hand é compatível com interfaces de comunicação RS485 e CAN, suportando protocolos como Modbus-RTU. A programação e controlo podem ser realizados através de linguagens como Python e C, oferecendo flexibilidade para integração em diversos sistemas robóticos.

A compatibilidade mecânica com manipuladores específicos não é diretamente indicada nas fontes disponíveis. No entanto, a utilização de interfaces mecânicas padrão e a flexibilidade de comunicação sugerem que a integração com manipuladores existentes é possível, desde que sejam realizadas as adaptações necessárias[1].



Figura 2.1: OceanTrix Hand desenvolvida pela OceanTrix Robotics[1].

Agile Hand

A *Agile Hand* (Figura 2.2), desenvolvida pela Agile Robots SE, é uma mão robótica antropomórfica com 15 a 16 graus de liberdade, distribuídos por 20 articulações, permitindo movimentos complexos e precisos. Cada dedo é modular e atuado por motores elétricos integrados, capazes de gerar até 10 N de força na ponta dos dedos e velocidades de até 360°/s. A mão incorpora sensores de torque e posição em todas as articulações, permitindo controlo em malha fechada com conformidade ativa. É compatível com manipuladores que seguem a norma ISO 9409-1-50-4-M6 e oferece suporte a C++, Python e ROS, com comunicação a 1 kHz por protocolo proprietário[2].



Figura 2.2: Agile Hand desenvolvida pela Agile Robots SE [2].

2.2.2 Soluções *open-source*

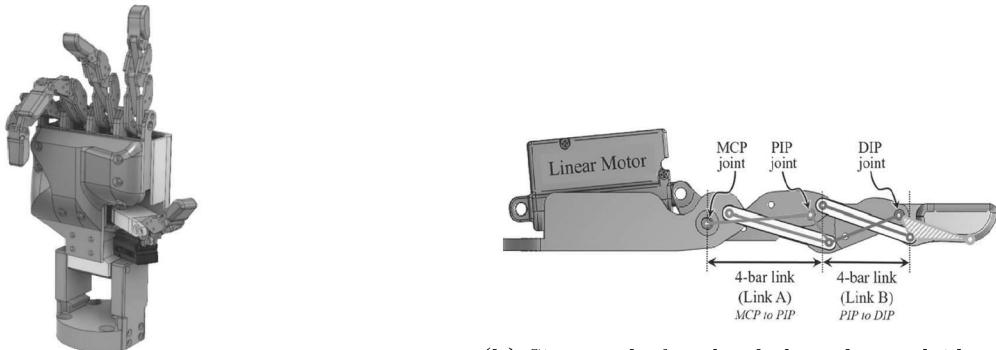
As mãos robóticas *open-source* surgem como alternativas acessíveis e personalizáveis às soluções comerciais, especialmente em contextos de investigação e desenvolvimento. Entre as suas principais vantagens destacam-se o baixo custo, a flexibilidade de modificação e a comunidade ativa de utilizadores e desenvolvedores, que facilita a partilha de conhecimento e melhorias contínuas. No entanto, estas soluções apresentam também desvantagens, como a

menor robustez mecânica, limitações em precisão e desempenho, e falta de suporte técnico especializado, o que pode representar um desafio em aplicações industriais exigentes.

Para este projeto, a escolha de construir uma mão robótica *open-source* revela-se mais vantajosa, uma vez que será utilizada em contexto de investigação e desenvolvimento, onde a possibilidade de realizar modificações estruturais e funcionais é essencial. Para além disso, a natureza aberta do projeto permite uma maior adaptabilidade aos requisitos específicos da aplicação.

HRI Hand

A HRI Hand [3] (Figura 2.3a) é uma mão robótica antropomórfica *open-source* desenvolvida para atuar como *end-effector* em manipuladores robóticos. Esta mão possui 15 graus de liberdade (sendo apenas 6 juntas ativas), distribuídos entre os 5 dedos, com cada dedo controlado por um motor linear dedicado, enquanto o polegar utiliza dois motores adicionais para realizar movimentos de abdução e adução. A mão utiliza um mecanismo subatuado baseado num sistema de duas barras articuladas (*four-bar linkage*, ilustrada na figura 2.3b), permitindo que as articulações Proximal Interphalangeal Joint (PIP) e Distal Interphalangeal Joint (DIP) sejam movidas de forma dependente a partir do motor que controla a articulação Metacarpophalangeal Joint (MCP).



(a) HRI Hand desenvolvida por Park, et al. [3].

(b) Sistema de *four-bar linkage* desenvolvida na HRI Hand [3].

Figura 2.3: Mão HRI e seu mecanismo

Em termos de compatibilidade de software, a HRI Hand é totalmente integrada com o Robot Operating System (ROS) e segue a norma ISO 9409-1-50-4-M6, garantindo compatibilidade com diversos manipuladores robóticos, como o UR3.

Robot Nano Hand

A Robot Nano Hand [4] (Figura 2.4) é uma mão robótica antropomórfica *open source*, concebida para replicar o tamanho e os movimentos de uma mão humana adulta, oferecendo um total de 11 graus de liberdade.

A sua estrutura inclui 4 dedos independentes e 1 polegar parcialmente opositor, com cada dedo capaz de realizar movimentos de flexão e laterais, garantindo uma ampla gama de gestos. O sistema de atuação é baseado no uso de tendões, que percorrem os dedos e são controlados por servomotores.

A mão conta também com uma câmara Raspberry Pi V2 integrada na palma, o que a torna adequada para aplicações em visão computacional, incluindo reconhecimento de gestos e análise de objetos. O controlo é gerido por um NVIDIA Jetson Nano Development Kit, um dispositivo robusto que suporta algoritmos de inteligência artificial, como redes neurais para classificação de imagens e segmentação de objetos. Embora projetada como uma solução autónoma, com os servos incorporados diretamente na estrutura e um servo adicional que fornece rotação no eixo de inclinação para o conjunto da mão, o design modular da Robot Nano Hand permite adaptações que podem facilitar a sua integração com manipuladores robóticos ou outros sistemas robóticos.

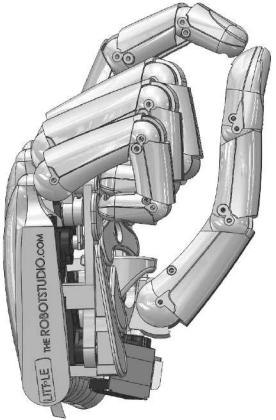


Figura 2.4: Robot Nano Hand desenvolvida por The Robot Studio[4].

Alaris Hand

A Alaris Hand [5] (Figura 2.5) é uma mão robótica antropomórfica *open source*, concebida para reproduzir as funcionalidades e movimentos de uma mão humana. Este dispositivo apresenta um total de 6 graus de liberdade, permitindo movimentos versáteis dos dedos, que incluem flexão, extensão e manipulação de objetos. O sistema de atuação baseia-se em atuadores lineares miniaturizados, acoplados a mecanismos de quatro barras acionados por transmissões de parafuso sem-fim e cremalheira, dispensando o uso de cabos que simulem tendões artificiais. Este método garante um controlo eficaz das articulações e proporciona uma força de preensão robusta. Além disso, a Alaris Hand está equipada com sensores de posição que monitorizam os movimentos articulares em tempo real.

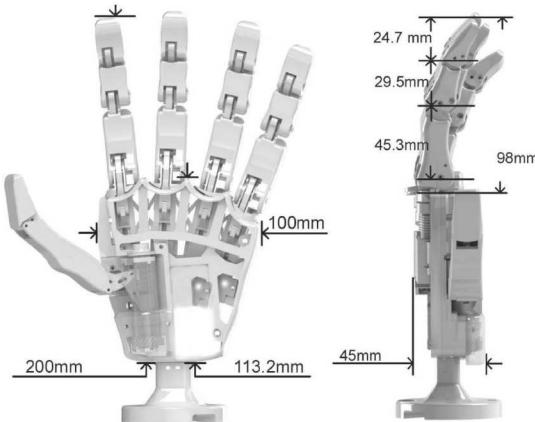


Figura 2.5: Alaris Hand desenvolvida por Nurpeissova, et al.[5]

LEAP Hand

A LEAP Hand [6] (Figura 2.6) é uma mão robótica antropomórfica de baixo custo, comparado com as soluções comerciais, projetada para aplicações em *machine learning*. Esta mão possui 4 dedos e um total de 16 graus de liberdade, com um mecanismo inovador de abdução-adução que mantém todos os graus de liberdade disponíveis em diversas posições das articulações, o que contribui para a sua alta agilidade. O método de atuação utiliza motores diretos Dynamixel, o que proporciona resistência a torques elevados e aumenta a durabilidade em tarefas de manipulação prolongada.

No que diz respeito aos sensores, a LEAP Hand é capaz de integrar sensores externos para teleoperação ou aprendizagem, mas não possui sensores embutidos nativamente. Em termos de software, ela é compatível com diversas plataformas, incluindo ROS, Python e C++, além de oferecer um simulador baseado no Isaac Gym e Pybullet, o que permite a realização de experiências de simulação para o mundo real (*sim2real*).

A LEAP Hand é projetada para ser modular e facilmente reparável, utilizando peças impressas em 3D e componentes disponíveis no mercado. A sua compatibilidade com manipuladores robóticos inclui braços como o UR5, o que amplia as suas aplicações em investigação e desenvolvimento.

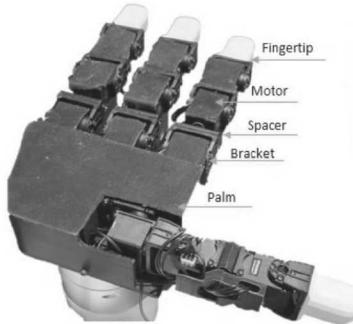


Figura 2.6: Leap Hand desenvolvida por Shaw, et al.[6]

2.2.3 Seleção da mão a construir

Após a análise comparativa das diferentes mãos robóticas open-source disponíveis, summarizada na Tabela 2.1, optou-se pela LEAP Hand [6] como a solução mais adequada para replicação e adaptação no âmbito deste projeto. Esta decisão fundamenta-se principalmente no seu design de construção simples, baseado em servomotores em vez de sistemas de tendões, o que reduz a complexidade mecânica, facilita a montagem e minimiza potenciais pontos de falha.

Modelo	DoF	Atuação	Sensores	Compatibilidade com manipulador	Software
HRI Hand	6	Motores Lineares	Nenhum	UR3	ROS
Robot Nano Hand	11	Tendões	Câmera	Não mencionado	Não mencionado
Alaris Hand	6	Tendões	Posição	Não	Não mencionado
LEAP Hand	16	Servomotores	Posição	UR5	ROS / Python / C++ / simulação

Tabela 2.1: Comparação entre diferentes mãos robóticas open-source

Apesar da sua simplicidade, a LEAP Hand oferece um elevado número de graus de liberdade (16 DoF), superior à maioria das alternativas analisadas, permitindo uma manipulação mais versátil. A presença de sensores de posição integrados garante um controlo eficaz dos movimentos articulares, fator essencial para aplicações em tarefas complexas de manipulação.

Adicionalmente, destaca-se a sua compatibilidade com o sistema ROS, bem como com linguagens de programação como Python e C++, o que proporciona uma plataforma de desenvolvimento acessível e extensível. A possibilidade de integração com manipuladores como o UR5 reforça ainda mais a sua aplicabilidade prática. Por fim, o seu design modular e totalmente imprimível em 3D permite uma manutenção económica e favorece alterações e melhorias iterativas durante o desenvolvimento.

2.3 SENSORES DE CONTACTO

A percepção tátil desempenha um papel vital ao permitir que mãos robóticas interajam com o ambiente de forma adaptativa, imitando as capacidades da mão humana. Esta tecnologia é essencial em áreas como a robótica industrial, onde é necessária uma manipulação precisa, e a aprendizagem por reforço, que se beneficia do *feedback* do mundo real para melhorar o desempenho robótico.

Ao longo dos anos foram propostas diversas tecnologias para sensores táteis, sendo as mais relevantes, no contexto deste trabalho, os sensores baseados em tecnologia magnética, polímeros condutivos e materiais piezoresistivos. Esta secção abordará as principais características e aplicações destas tecnologias em mãos robóticas, destacando as suas vantagens e limitações.

Apesar dos avanços, apenas um número limitado dessas tecnologias tem sido implementado com sucesso em plataformas robóticas reais. Algumas empresas já produzem sensores táteis para aplicações robóticas, como o sensor de ponta de dedo BioTac da SynTouch (ilustrado na figura 2.7) [7]. No entanto, estes sensores continuam a apresentar custos elevados e, frequentemente, requerem a integração com outras tecnologias para otimização do desempenho.



Figura 2.7: Sensor Biotac desenvolvido pela SynTouch [7].

2.3.1 Sensores baseados em Tecnologia Magnética

Os sensores baseados em tecnologia magnética têm ganho bastante destaque na área da robótica tátil devido à sua elevada precisão, ausência de desgaste e boa durabilidade. Estes sensores destacam-se também por permitirem a deteção de forças sem contacto físico direto com o sensor, o que os torna especialmente interessantes para aplicações em ambientes adversos.

Filmes magnéticos flexíveis, como os apresentados por Jamone, et al. [8] (esquema de funcionamento do sensor presente na figura 2.8) e Yan, et al.[9], consistem numa camada de elastómero incorporando partículas magnéticas e sensores de efeito Hall colocados abaixo dessa camada. Esta configuração permite medir forças normais e de cisalhamento com elevada sensibilidade, sendo particularmente útil para aplicações em superfícies curvas. No entanto, este tipo de sensor tende a saturar em forças superiores a 4 N, o que limita a sua utilização em tarefas que envolvem forças mais intensas.

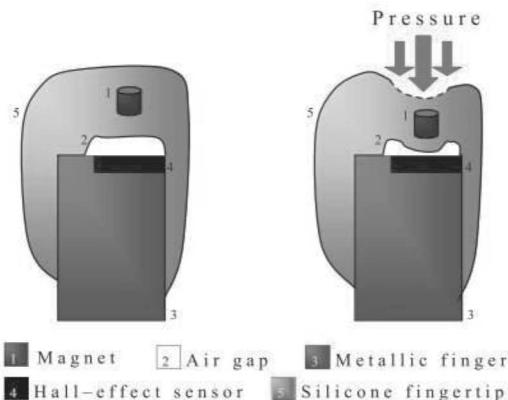


Figura 2.8: Sensor com filme magnético flexível desenvolvido por Jamone, et al. [8].

Outra abordagem promissora é o uso de *magneto-elastomer composites*, que combinam materiais elásticos com partículas magnéticas dispersas de forma controlada. Estes sensores,

como os utilizados nos trabalhos de Kawasetsu et al. [10] (Figura 2.9), mostram boa sensibilidade mesmo em contactos rápidos e com elevada aceleração, para além de apresentarem resposta estável a forças contínuas. Contudo, o processo de fabrico pode ser complexo, o que pode implicar maiores custos e limitações em termos de flexibilidade geométrica.

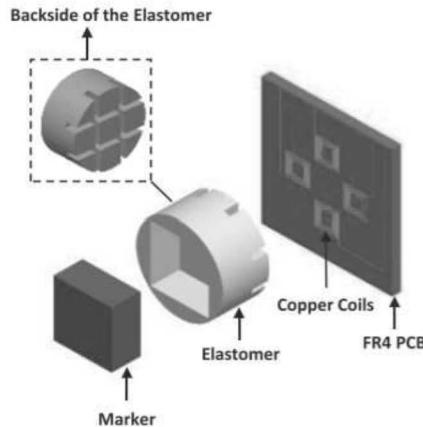


Figura 2.9: Composição do sensor desenvolvido por Kawasetsu et al. [10].

Os sensores magnetoresistivos representam uma alternativa particularmente robusta, sendo eficazes mesmo em condições adversas como altas temperaturas, humidade ou ambientes com poeiras. O sensor desenvolvido por Alfadhel et al. [11] utiliza uma estrutura em espiral com materiais magnetoresistivos sensíveis ao campo magnético gerado por uma fonte próxima. Embora a sua robustez seja uma vantagem, estes sensores exigem elevada precisão na montagem e estabilidade dos campos magnéticos externos, o que pode complicar a sua integração em sistemas compactos e móveis [12].

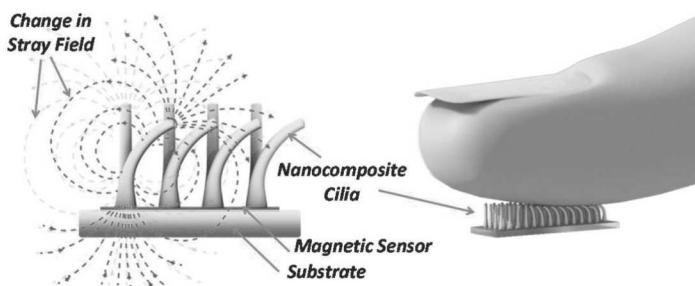


Figura 2.10: Princípio de operação do sensor baseado em materiais magnetoresistivos [11].

Apesar dos desafios, os sensores magnéticos continuam a evoluir e a mostrar grande potencial, sobretudo em aplicações onde a precisão e a resistência a condições ambientais extremas são fundamentais.

2.3.2 Sensores baseados em Polímeros Condutivos

Os sensores baseados em polímeros condutivos têm-se destacado como soluções promissoras para aplicações táteis em sistemas robóticos, sobretudo pela sua leveza, flexibilidade e facilidade de adaptação a superfícies irregulares. Estes dispositivos exploram a variação das propriedades

elétricas de materiais condutivos em resposta a deformações mecânicas, permitindo a detecção eficiente de toque e pressão.

Entre as principais abordagens, destacam-se os sensores piezoresistivos, que se baseiam na alteração da resistência elétrica com a aplicação de pressão. São dispositivos de baixo custo e integração simples, adequados para superfícies complexas, como demonstrado em [13], [14]. Contudo, apresentam limitações em termos de precisão, resposta não linear e durabilidade.

Os sensores piezoelétricos, por sua vez, utilizam materiais como o Polyvinylidene fluoride (PVDF), capazes de gerar sinais elétricos sob estímulos dinâmicos. São valorizados pela elevada sensibilidade, leveza e flexibilidade [15], [16], sendo eficazes na deteção de vibrações e forças variáveis. No entanto, não são apropriados para medições de forças estáticas e requerem circuitos de condicionamento de sinal mais complexos.

Adicionalmente, os sensores triboelétricos operam com base na eletrificação por contato, sendo capazes de gerar sinais elétricos de forma autossuficiente, sem necessidade de alimentação externa. Trabalhos recentes [17], [18] demonstram o seu bom desempenho em ambientes adversos. Ainda assim, enfrentam desafios associados à fragilidade mecânica, complexidade de fabrico e sensibilidade a fatores ambientais.

2.3.3 Sensores baseados em Materiais Piezoresistivos

Os sensores baseados em materiais piezoresistivos têm sido amplamente utilizados em aplicações de sensoriamento tátil, especialmente no domínio da robótica, devido à sua elevada sensibilidade à aplicação de forças e pressões. O seu funcionamento baseia-se na variação da resistência eléctrica resultante de deformações mecânicas, permitindo medições precisas e localizadas.

Um exemplo representativo desta tecnologia são os sensores de resistência sensível à força (Force Sensing Resistors (FSRs)), que combinam um design compacto e robusto com uma elevada adaptabilidade a superfícies curvas e complexas, como as encontradas em mãos robóticas [19], [20]. Estas características tornam-nos particularmente adequados para sistemas de deteção tátil de baixo custo e fácil integração.

Entre as principais vantagens dos sensores piezoresistivos destacam-se o custo reduzido de produção, a leveza, a flexibilidade e a possibilidade de personalização da sensibilidade em função dos requisitos da aplicação [14]. Para além disso, a simplicidade do seu princípio de funcionamento facilita a integração com circuitos eletrónicos de leitura e controlo.

No entanto, estes sensores apresentam também algumas limitações. A sua precisão é, por vezes, inferior à de outras tecnologias sensoriais, e a resposta não linear pode dificultar a calibração. Adicionalmente, a durabilidade e a estabilidade da resposta ao longo do tempo podem ser comprometidas, especialmente em aplicações sujeitas a esforços repetidos ou prolongados [19].

2.3.4 Escolha dos sensores a incorporar no projeto

A seleção dos sensores táteis para o presente projeto foi fundamentada numa análise comparativa entre as diferentes tecnologias disponíveis, com base em critérios como princípio

de funcionamento, vantagens técnicas e limitações práticas. A Tabela 2.2 apresenta uma síntese dos principais tipos de sensores considerados.

Tipo de Sensor	Princípio de Funcionamento	Vantagens	Desvantagens
Magnético	Deteta variações em campos magnéticos sem contacto físico direto	Elevada precisão, ausência de desgaste mecânico, boa durabilidade	Custo mais elevado, suscetibilidade a interferência magnética, maior complexidade
Polímero Condutivo	Condutividade elétrica varia com a deformação mecânica do material	Elevada sensibilidade, flexibilidade, leveza	Menor durabilidade, sensibilidade a fatores ambientais, resposta lenta
Piezoresistivo (FSR)	Resistência elétrica diminui com o aumento da pressão aplicada	Baixo custo, fácil integração, formato compacto e flexível	Resposta não linear, menor precisão, vida útil limitada

Tabela 2.2: Comparação entre diferentes tipos de sensores táteis

Com base na análise comparativa realizada, os sensores piezoresistivos do tipo FSR foram selecionados como a solução mais adequada para o projeto em desenvolvimento. Esta decisão fundamenta-se num conjunto de fatores que favorecem a sua aplicação em contextos de robótica tátil, nomeadamente em mãos robóticas.

Em primeiro lugar, destaca-se a sua estrutura fina, leve e flexível, que permite uma integração eficiente em superfícies curvas, como as pontas dos dedos e a palma da mão robótica. Para além disso, os FSRs oferecem um equilíbrio vantajoso entre desempenho e acessibilidade económica.

Adicionalmente, o seu design compacto e a simplicidade de integração com os sistemas eletrónicos permitem reduzir significativamente a complexidade de implementação, fator particularmente relevante em projetos com restrições de espaço físico, como é o caso da LEAP Hand [6].

2.4 CONCLUSÃO

Ao longo deste capítulo foram analisadas várias opções tecnológicas com vista à definição da arquitetura sensorial do projeto. Numa primeira fase, avaliou-se um conjunto de mãos robóticas antropomórficas, tanto comerciais como open-source, com o objetivo de selecionar uma solução adequada aos requisitos de integração tátil. Desta análise resultou a escolha da LEAP Hand[6], cuja estrutura modular, dimensões compactas e compatibilidade com sensores externos a tornam particularmente apropriada para o presente projeto.

De seguida, procedeu-se ao estudo comparativo de diferentes tecnologias sensoriais, nomeadamente sensores baseados em princípios magnéticos, polímeros condutivos e materiais piezoresistivos. A avaliação teve em consideração fatores como sensibilidade, facilidade de integração e viabilidade económica.

Com base nesta análise, foram selecionados sensores piezoresistivos do tipo FSR, devido ao seu formato compacto, baixo custo, facilidade de aplicação em superfícies curvas e capacidade de resposta adequada aos requisitos de deteção tátil em mãos robóticas.

Em suma, as decisões tomadas relativamente à mão robótica e à tecnologia sensorial asseguram uma base sólida para o desenvolvimento do sistema de percepção tátil, alinhando-se com os objetivos técnicos e funcionais definidos para o projeto.

CAPÍTULO 3

Ferramentas de suporte

3.1 INTRODUÇÃO

A adoção de uma mão robótica *open-source* apresenta diversas vantagens, entre as quais se destacam a disponibilização de código de apoio e a existência de decisões previamente fundamentadas relativamente à seleção de componentes mecânicos e eletrónicos. Estas características permitem reduzir significativamente o tempo necessário para o desenvolvimento de um sistema funcional, evitando o esforço associado ao desenho e especificação de soluções desde a parte inicial do projeto.

Neste capítulo, descrevem-se os principais elementos já existentes e utilizados como base no presente trabalho, com particular destaque na LEAP Hand, desenvolvida por Shaw, et al. [6]. Serão abordadas as escolhas efetuadas por estes autores ao nível dos motores, do controlador de motores, da fonte de alimentação e das ferramentas de software disponibilizadas, que serviram de ponto de partida para o desenvolvimento realizado neste projeto.

3.2 HARDWARE

3.2.1 Motores Dynamixel

No desenvolvimento da LEAP Hand, Shaw, et al. [6] optaram por utilizar os motores Dynamixel XC330-M288-T, amplamente utilizados em aplicações robóticas devido à sua fiabilidade, eficiência energética e dimensões compactas. Estes atuadores oferecem controlo detalhado sobre posição, velocidade e corrente, permitindo gerar movimentos suaves, reproduzíveis e bem definidos — uma característica fundamental em sistemas de manipulação, como mãos robóticas antropomórficas.

Os motores Dynamixel XC330-M288-T (Figura 3.1) operam a 5V e apresentam uma corrente nominal de 1.8A, o que os torna compatíveis com fontes de alimentação de baixa potência, mantendo um desempenho consistente. Adicionalmente, a comunicação digital via protocolo TTL permite a ligação e controlo eficiente de múltiplos motores em rede, facilitando a sua integração com microcontroladores e promovendo uma arquitetura modular e escalável.



Figura 3.1: Motor Dynamixel XC330-M288-T utilizado na construção da LEAP Hand[6]

O controlo destes motores pode ser realizado por diferentes meios, nomeadamente através da ferramenta gráfica Dynamixel Wizard 2.0, útil para configuração e diagnóstico, e da biblioteca Dynamixel SDK, que permite o controlo programático e a integração com sistemas desenvolvidos em diversas linguagens de programação. Estas opções tornam o sistema versátil e facilmente adaptável a diferentes contextos de desenvolvimento, incluindo ambientes compatíveis com ROS.

Os motores Dynamixel armazenam e gerem os seus parâmetros internos através de uma tabela de controlo de memória, onde cada parâmetro — como posição, velocidade, corrente ou temperatura — está associado a um endereço específico. Essa tabela encontra-se dividida em duas secções principais: a **EEPROM**, que contém parâmetros permanentes (como o ID do motor, taxa de transmissão, limites de corrente ou posição) e só pode ser modificada após a reinicialização do motor; e a **RAM**, onde se encontram os parâmetros dinâmicos, como a posição atual, velocidade, temperatura e estado do motor, permitindo atualizações em tempo real durante a operação. Esta estrutura torna o controlo dos motores altamente flexível e eficiente, uma vez que permite o acesso direto e segmentado aos dados necessários para a gestão do comportamento do sistema robótico.

3.2.2 Controlador e Fonte de Alimentação

Para além da seleção dos motores, a escolha do controlador de comunicação e da fonte de alimentação tem um impacto significativo no desempenho e na fiabilidade do sistema. No que respeita à interface de comunicação com os motores, os autores da LEAP Hand [6] optaram por utilizar o **Dynamixel U2D2** (Figura 3.2), um conversor USB para TTL *half-duplex*. Este dispositivo permite estabelecer comunicação entre um computador e múltiplos motores Dynamixel através de uma única porta serial, utilizando um barramento em cadeia (*daisy chain*). Este método de ligação, possível graças à arquitetura *half-duplex* dos motores, permite que vários atuadores sejam ligados em série, reduzindo significativamente a complexidade do sistema e facilitando a sincronização entre motores. O U2D2 suporta taxas de transmissão elevadas (até 4,5 Mbps), possui um conector JST de 3 pinos para comunicação TTL, e inclui

um terminal de alimentação auxiliar (VDD), permitindo a alimentação dos motores a partir de uma fonte externa apropriada.

Relativamente à fonte de alimentação, foi necessário considerar que cada motor Dynamixel XC330-M288-T pode atingir um consumo máximo de 1.8A em condições de carga. Tendo em conta a utilização de 16 motores, a corrente total exigida pode atingir os 28.8A. Assim, foi selecionada uma fonte de alimentação industrial (Figura 3.3) com saída estabilizada de 5V — a mesma tensão nominal dos motores — e capacidade de fornecimento de até 30A, assegurando uma margem de segurança adequada para o funcionamento simultâneo de todos os atuadores.



Figura 3.2: Controlador Dynamixel U2D2 utilizado para controlar os motores do projeto



Figura 3.3: Fonte de Alimentação industrial com saída a 5V e capacidade de 30A utilizada neste projeto

3.3 SOFTWARE

3.3.1 Dynamixel Wizard 2.0

A Dynamixel disponibiliza diversas ferramentas para configuração, monitorização e controlo dos seus motores, entre as quais se destaca o Dynamixel Wizard 2.0, uma interface gráfica intuitiva (Figura 3.4) que permite ao utilizador interagir diretamente com cada motor. Antes de iniciar o controlo coordenado de um sistema com múltiplos atuadores, é essencial garantir que cada motor possui um identificador único. Por defeito, todos os motores Dynamixel são fornecidos com o mesmo identificador ($ID = 1$), o que inviabiliza a sua utilização simultânea num mesmo barramento de comunicação. Assim, uma das primeiras tarefas consiste na atribuição de um ID exclusivo a cada motor, procedimento que pode ser facilmente realizado através do Dynamixel Wizard 2.0, modificando o parâmetro correspondente no firmware do atuador.

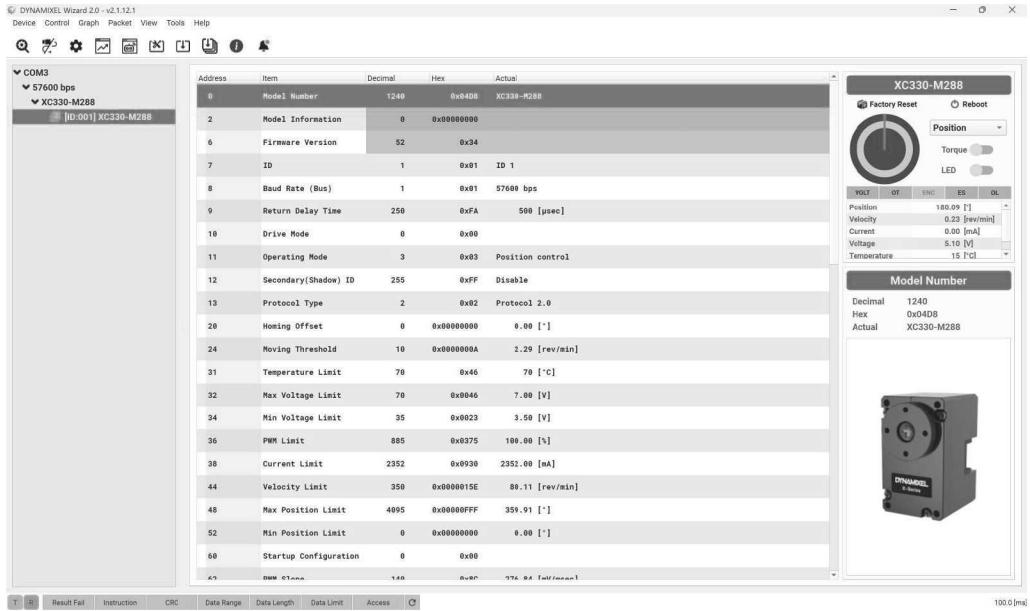


Figura 3.4: Interface Gráfica da ferramenta Dynamixel Wizard 2.0 utilizada para alterar o ID de cada motor

Para além da atribuição de IDs individuais, esta ferramenta permite ainda a definição de grupos de motores com IDs secundários, úteis para cenários em que se pretende acionar vários motores em simultâneo e de forma idêntica. No entanto, esta funcionalidade não será explorada neste projeto, dado que o controlo será efetuado de forma independente para cada atuador.

Adicionalmente, o Dynamixel Wizard 2.0 oferece funcionalidades de diagnóstico e monitorização em tempo real, possibilitando a visualização de parâmetros como a posição atual, velocidade, corrente consumida, entre outros. Esta capacidade é particularmente útil durante as fases de teste, permitindo validar o comportamento dos motores e detetar eventuais anomalias no sistema.

3.3.2 Biblioteca Dynamixel SDK

O Dynamixel SDK é uma biblioteca de software multiplataforma que permite a comunicação e o controlo eficiente dos motores Dynamixel em diversas linguagens de programação, tais como Python, C, C++ e Java. Concebido para proporcionar uma interface unificada e de alto nível, o SDK abstrai as complexidades da comunicação serial, simplificando as operações de leitura e escrita de dados nos motores.

Esta biblioteca suporta os principais protocolos de comunicação Dynamixel, incluindo o Protocolo 2.0, utilizado no presente projeto para os motores XC330-M288-T, o que assegura compatibilidade com as funcionalidades mais recentes dos atuadores, como o controlo de corrente, modos operacionais configuráveis e diagnósticos internos. Através do SDK, é possível configurar parâmetros como posição, velocidade, corrente máxima, limites de movimento e modo de operação, permitindo adaptar o comportamento dos motores às exigências específicas de cada aplicação.

3.3.3 ROS2

O ROS é uma *framework* amplamente utilizada no desenvolvimento de sistemas robóticos, proporcionando um ambiente modular que facilita o desenvolvimento, gestão e interligação de múltiplos nós que executam tarefas específicas em paralelo. Esta arquitetura distribuída permite uma organização estruturada do código, promovendo a escalabilidade e a reutilização de componentes, o que é particularmente vantajoso em projetos complexos, como o desenvolvimento de mãos robóticas antropomórficas.

No âmbito da LEAP Hand, Shaw et al. [6] disponibilizaram documentação e exemplos compatíveis com Python, ROS 1 e ROS 2, conferindo flexibilidade na escolha da infraestrutura de desenvolvimento. Neste projeto, optou-se pela utilização do ROS 2, em detrimento do ROS 1, com base nas vantagens técnicas e estruturais que a nova versão oferece.

Entre os principais fatores que justificam esta escolha destacam-se a utilização do *middleware Data Distribution Service (DDS)*, que melhora substancialmente a comunicação entre nós, tornando-a mais eficiente e robusta — um requisito fundamental em sistemas com múltiplos atuadores e sensores a operar em tempo real. Para além disso, o ROS 2 elimina a necessidade de um nó central como o *rosmaster*, permitindo uma arquitetura verdadeiramente distribuída e mais resiliente a falhas. Esta mudança traduz-se numa maior robustez e flexibilidade na gestão da rede de nós, especialmente em contextos com comunicações complexas ou heterogéneas.

Outro fator relevante é o fim gradual do suporte ao ROS 1, cuja manutenção oficial está em fase de descontinuação, sendo que muitas das novas bibliotecas e ferramentas estão a ser exclusivamente desenvolvidas para o ROS 2. A adoção do ROS 2 assegura, portanto, maior longevidade, compatibilidade com futuras atualizações e acesso a funcionalidades mais recentes da comunidade de robótica.

Por fim, o suporte nativo a sistemas real-time, a maior portabilidade entre plataformas e o foco na segurança e escalabilidade reforçam a decisão de utilizar o ROS 2 neste projeto. Esta escolha visa garantir um desenvolvimento mais moderno e sustentável, mantendo a compatibilidade com sistemas robóticos avançados e preparados para integração em ambientes distribuídos.

3.3.4 Documentação da LEAP Hand

Os autores da LEAP Hand [6] disponibilizam documentação detalhada e exemplos de código para facilitar a reprodução e o desenvolvimento com base na sua plataforma. Embora essa documentação suporte a utilização em Python, ROS 1 e ROS 2, o código fornecido foi desenvolvido com um foco específico: a execução de políticas previamente treinadas em simulação (utilizando o simulador incluído no projeto) e a realização de movimentos predefinidos da mão robótica.

No entanto, esta abordagem não contempla a possibilidade de controlo direto e individualizado dos motores, nem permite uma personalização completa do comportamento da mão, o que limita a sua aplicabilidade em contextos que exigem maior flexibilidade no desenvolvimento de novos controladores ou algoritmos de manipulação.

Tendo em conta estas limitações, e considerando os objetivos deste projeto, optou-se por não utilizar diretamente o código disponibilizado pelos autores da LEAP Hand, desenvolvendo-se, em alternativa, uma nova infraestrutura de controlo que permite acesso completo e individualizado a cada motor, facilitando o desenvolvimento de estratégias de controlo personalizadas e adaptadas às necessidades específicas deste trabalho.

3.4 CONCLUSÃO

Ao longo deste capítulo, foram apresentadas as principais ferramentas de suporte utilizadas no desenvolvimento do projeto, com especial destaque nos componentes de hardware e software que viabilizam a implementação da mão robótica.

Inicialmente, identificaram-se os motores Dynamixel XC330-M288-T e o controlador U2D2 como soluções adequadas para assegurar movimentos compatíveis com a estrutura da LEAP Hand. A fonte de alimentação foi selecionada com base nas exigências energéticas do sistema, garantindo estabilidade e segurança durante a operação.

Paralelamente, foram descritas as ferramentas de software de apoio à configuração e controlo dos motores, nomeadamente o Dynamixel Wizard 2.0 e a biblioteca Dynamixel SDK, bem como o ambiente de desenvolvimento baseado em ROS 2, que oferece escalabilidade e integração com outros módulos do sistema.

Adicionalmente, a documentação da LEAP Hand foi analisada como referência complementar, optando-se por não utilizar diretamente o código disponibilizado de forma a permitir o controlo individualizado de cada motor.

Em suma, a seleção e integração das ferramentas de suporte descritas neste capítulo estabeleceram uma infraestrutura adaptada às necessidades do sistema, contribuindo para a robustez, modularidade e evolução futura do desenvolvimento da mão robótica.

CAPÍTULO 4

Projeto e Implementação da Solução

4.1 INTRODUÇÃO

Este capítulo apresenta o processo de desenvolvimento da solução proposta, desde a construção física da mão robótica até à implementação do sistema sensorial. Inicia-se com a apresentação das etapas associadas ao fabrico e montagem da estrutura da mão robótica, seguindo-se a configuração dos motores e o desenvolvimento da lógica de controlo, desde o acionamento de motores individuais até ao controlo coordenado da mão completa.

É igualmente descrita a arquitetura de software desenvolvida, com especial destaque para as funcionalidades avançadas implementadas, como o ajuste dinâmico de corrente durante o contacto com objetos, a definição de velocidades relativas entre dedos e falanges, e a introdução de atrasos temporais entre movimentos.

Adicionalmente, aborda-se a implementação do sistema sensorial de deteção de contacto, incluindo a seleção do microcontrolador, o desenvolvimento físico da unidade de aquisição e os testes realizados com diferentes sensores.

4.2 DESENVOLVIMENTO DA MÃO ROBÓTICA

4.2.1 Fabrico e Montagem da estrutura

A construção da estrutura da mão robótica seguiu integralmente as indicações fornecidas na documentação da LEAP Hand, a qual disponibiliza de forma aberta e acessível todos os ficheiros CAD necessários, bem como instruções detalhadas de montagem. Esta abordagem permitiu realizar uma primeira montagem da estrutura de forma fiel ao projeto original, facilitando a replicação inicial da arquitetura mecânica.

As peças foram produzidas por impressão 3D com recurso a uma Creality K1, uma impressora de alto desempenho que assegura uma boa qualidade dimensional e tempos de fabrico reduzidos. Para a estrutura principal foi utilizado o filamento Hyper PLA da Creality, selecionado pela sua rigidez e facilidade de impressão. Já para as pontas dos dedos, onde se

pretende maior flexibilidade e capacidade de aderência ao contacto com objetos, foi utilizado o Python Flex TPU da FormFutura.

A construção dos dedos revelou-se a fase mais demorada do processo de montagem, devido ao elevado número de componentes envolvidos e à complexidade dos passos necessários. Todos os dedos seguem a mesma configuração estrutural, com exceção do polegar, que apresenta um desenho distinto adaptado à sua função anatómica e posição na mão.

Na Figura 4.1 é apresentado o polegar já montado, enquanto na Figura 4.2 observa-se um dos restantes dedos, com estrutura idêntica entre si. A montagem de cada dedo envolveu operações detalhadas e sequenciais que exigiram precisão e atenção redobrada, especialmente na organização dos cabos e no alinhamento das juntas móveis.



Figura 4.1: Polegar construído de acordo com a Documentação da Leap Hand

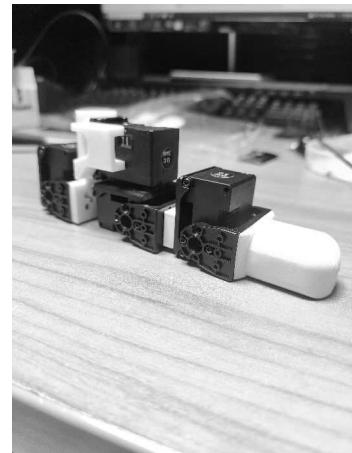


Figura 4.2: Dedo construído de acordo com a documentação da Leap Hand correspondente à estrutura do dedo indicador, médio e anelar

No final do processo, obteve-se a mão totalmente montada, de acordo com a estrutura proposta na LEAP Hand, conforme ilustrado na Figura 4.3.



Figura 4.3: Mão robótica completamente montada, seguindo a estrutura proposta na documentação da LEAP Hand.

Para complementar a descrição deste processo, no Apêndice A.1.1 e A.1.2 e são apresentadas imagens sequenciais representativas da construção de cada dedo, bem como hiperligações A.1.3 para vídeos que documentam o procedimento completo de montagem, constituindo assim um apoio visual adicional à documentação escrita.

Embora a montagem inicial tenha seguido rigorosamente as orientações da documentação da LEAP Hand, durante o desenvolvimento do projeto surgiram necessidades específicas que motivaram adaptações à estrutura original. Estas modificações serão detalhadas nas secções seguintes deste capítulo.

4.2.2 Configuração dos Motores

Antes de iniciar o desenvolvimento do código responsável pelo controlo dos movimentos da mão robótica, foi necessário proceder à configuração inicial dos motores Dynamixel. Esta fase incluiu a atribuição de identificadores únicos (IDs) a cada motor, uma vez que todos os motores, por defeito, vêm configurados com o mesmo ID. Para isso, recorreu-se à ferramenta gráfica Dynamixel Wizard 2.0, que permite, de forma intuitiva, alterar os parâmetros fundamentais de cada atuador.

Nesta configuração inicial, os IDs atribuídos seguiram a mesma lógica utilizada na documentação da LEAP Hand, garantindo assim compatibilidade com a estrutura de referência proposta pelos autores.

Modos de Operação do Motores Dynamixel

Após a atribuição dos IDs a cada motor, foi necessário configurar o modo de operação correspondente. Os motores Dynamixel XC330-M288-T suportam diversos modos de funcionamento, permitindo adaptá-los a diferentes tipos de movimentos e estratégias de controlo.

Na presente subsecção, são descritos os principais modos de operação disponibilizados por estes atuadores, destacando as suas características e o comportamento em situações com ou sem contacto com obstáculos.

1. Position Control Mode

Este modo permite definir uma posição angular específica que o motor deve atingir, dentro dos limites físicos configurados. Caso surja um obstáculo durante o movimento, o motor continuará a aplicar torque máximo na tentativa de alcançar a posição desejada, o que pode levar ao sobreaquecimento dos motores.

2. Extended Position Control Mode

Uma extensão do modo anterior, permite rotações superiores a 360°, tornando-se útil para aplicações que requerem movimento contínuo em rotação. No entanto, não é relevante no contexto deste projeto, dado que as juntas dos dedos não realizam rotações completas.

3. Velocity Control Mode

Este modo permite definir uma velocidade angular constante, sem especificar uma posição final. O motor acelera até atingir a velocidade pretendida e mantém-se em rotação. Na presença de um obstáculo, o motor continua a tentar girar, aplicando o torque máximo, o que pode resultar em sobreaquecimento.

4. PWM Control Mode

Permite controlar diretamente a potência fornecida ao motor através de modulação por largura de pulso (PWM). Este modo oferece um controlo mais direto e flexível, mas não garante precisão de movimento, nem proteção contra obstáculos, podendo causar sobreaquecimento em situações de bloqueio mecânico.

5. Current Control Mode

Neste modo, controla-se diretamente o torque aplicado ao definir a corrente desejada. Se não houver resistência, o motor continua a girar. Na presença de um obstáculo, o motor não tenta forçar o movimento, e mantém um torque constante correspondente ao valor de corrente definido, oferecendo uma resposta segura e previsível.

6. Current-Based Position Control Mode

Combina as vantagens do controlo de posição com limitação de corrente. O motor tenta alcançar uma determinada posição, mas limita o torque máximo aplicado, de acordo com o valor de corrente permitido. Se o obstáculo exigir mais torque do que o configurado, o motor interrompe o movimento, mantendo a força aplicada constante.

A Tabela 4.1 resume os diferentes modos de operação, destacando os parâmetros controlados, as aplicações ideais e o comportamento em situações de contacto com obstáculos:

Modo	Controla	Reação a Obstáculos
Position Control	Posição	Aplica torque máximo até atingir posição
Extended Position	Posição ($>360^\circ$)	Idêntico ao Position Control
Velocity Control	Velocidade	Aplica torque máximo para manter velocidade
Current Control	Corrente (torque)	Mantém torque constante, sem tentar forçar
Current-Based Position	Posição + Corrente	Tenta atingir posição, mas respeita limite de torque
PWM Control	Potência (PWM)	Não protege contra sobrecarga ou obstáculos

Tabela 4.1: Resumo dos modos de operação dos motores Dynamixel

A escolha do modo de operação adequado é essencial para garantir o desempenho desejado e a segurança do sistema. No contexto deste projeto, onde o contacto com objetos e a necessidade de limitar a força aplicada pelos dedos são aspetos fundamentais, foi escolhido o *Current-Based Position Control Mode*, uma vez que este modo permite controlar a posição de cada articulação, ao mesmo tempo que impõe um limite ao torque aplicado, o que evita danos em situações de bloqueio ou contacto inesperado.

4.2.3 Programação e Controlo

A implementação do controlo dos motores Dynamixel exige, numa fase inicial, a correta definição dos parâmetros de comunicação, de forma a garantir a fiabilidade da ligação entre o sistema computacional e os atuadores.

Entre os elementos essenciais a configurar, destaca-se a especificação da porta serial utilizada, tipicamente `/dev/ttyUSB0` em sistemas operativos baseados em Linux, a taxa de transmissão de dados (*baud rate*), o protocolo de comunicação (sendo utilizada a versão 2.0), a série dos motores (neste caso, a série X, correspondente aos modelos Dynamixel XC330-M288-T) e o identificador único (*ID*) atribuído previamente a cada motor.

A correta parametrização destes elementos é fundamental para que seja possível estabelecer comunicação com os motores, enviar comandos de controlo e ler os parâmetros relevantes de operação. Nos pontos seguintes, descreve-se de forma progressiva a programação de um motor individual, o controlo coordenado de um dedo completo e, por fim, a implementação da lógica de controlo da mão robótica na sua totalidade.

Programação de um motor individual

A programação de um motor individual constitui o primeiro passo no desenvolvimento da lógica de controlo da mão robótica. O processo tem início com a importação da biblioteca e a configuração da interface de comunicação serial. Tal configuração é efetuada através da criação dos objetos `portHandler` e `packetHandler`, com as instruções `portHandler = PortHandler(DEVICENAME)` e `packetHandler = PacketHandler(PROTOCOL_VERSION)`. O primeiro estabelece a ligação física à porta serial especificada (por exemplo, `/dev/ttyUSB0`),

enquanto o segundo gere a codificação e descodificação dos pacotes de dados, de acordo com o protocolo de comunicação adotado (neste projeto, a versão 2.0).

Após a abertura da porta e a verificação do sucesso da ligação, é necessário ativar o torque do motor para que este possa responder aos comandos enviados. A partir deste momento, torna-se possível controlar diversos parâmetros do motor, como a posição, a velocidade e a corrente máxima, dependendo do modo de operação previamente selecionado.

A leitura e escrita desses parâmetros nos registos internos do motor são realizadas através das funções disponibilizadas pelo SDK. A leitura pode abranger dados como a posição atual, velocidade de rotação, corrente consumida ou valor de PWM. A escolha da função adequada depende da dimensão dos dados a transferir, sendo comum o uso de `read2ByteTxRx()` e `read4ByteTxRx()` para valores de 2 e 4 bytes, respetivamente.

É importante salientar que os motores Dynamixel não fornecem diretamente a posição angular em graus ou radianos. Em vez disso, utilizam um sistema interno de representação baseado em *ticks*, no qual a posição é expressa por um valor inteiro. No caso do modelo XC330-M288-T, a resolução do encoder é de 12 bits, o que resulta em $2^{12} = 4096$ valores distintos ao longo de uma rotação completa de 360° . Deste modo, cada unidade (*tick*) corresponde a um incremento angular de aproximadamente $0,088^\circ$.

A conversão do valor lido, em *ticks*, para a correspondente posição angular em graus é efetuada através da equação (4.1):

$$\theta = \frac{360 \times \text{ticks}}{4096} \quad (4.1)$$

onde θ representa a posição angular em graus, e `ticks` é o valor devolvido diretamente pelo motor.

De forma análoga, a velocidade de rotação também é representada em *ticks*. Neste caso, a relação com as unidades físicas é definida por um fator de conversão fornecido pelo fabricante. Para o motor em questão, cada unidade corresponde a aproximadamente 0,229 rotações por minuto (RPM). Assim, a velocidade real pode ser obtida através da equação (4.2):

$$\text{Velocidade (RPM)} = \text{ticks} \times 0.229 \quad (4.2)$$

Adicionalmente, o sinal do valor atribuído à velocidade determina o sentido de rotação: valores positivos indicam rotação no sentido horário, enquanto valores negativos correspondem a uma rotação no sentido anti-horário.

No que diz respeito à corrente elétrica consumida pelo motor, esta é diretamente disponibilizada pelo SDK em miliamperes (mA), não sendo necessária qualquer conversão adicional. Este parâmetro está diretamente relacionado com o torque aplicado pelo motor, sendo, por isso, relevante para o diagnóstico do esforço mecânico exigido durante o funcionamento.

Programação e Controlo de um dedo

Após a compreensão do funcionamento individual dos motores Dynamixel e da configuração dos seus parâmetros, procedeu-se ao desenvolvimento do controlo coordenado de um dedo completo da mão robótica. Cada dedo integra quatro motores, tornando necessário garantir a

leitura sincronizada dos seus estados e o envio simultâneo de comandos de controlo, de forma eficiente e escalável.

Para tal, recorreu-se às funcionalidades de comunicação em bloco disponibilizadas pelo Dynamixel SDK, nomeadamente as operações `sync_read`, `sync_write`, `bulk_read` e `bulk_write`. No que respeita à leitura dos parâmetros dos motores — posição, velocidade e corrente — a função `sync_read` foi a selecionada, por permitir aceder a um mesmo conjunto de registo em múltiplos motores de forma síncrona. Esta abordagem revelou-se eficiente tanto em termos de desempenho como de utilização de memória, além de respeitar as limitações de tamanho de mensagem impostas pelo protocolo de comunicação, evitando os riscos associados à utilização do `bulk_read`.

Relativamente à escrita de comandos, optou-se pela função `bulk_write`, que permite enviar dados distintos para registo diferentes de múltiplos motores numa única operação. Apesar de o registo alvo ser frequentemente o mesmo (como a posição ou o limite de corrente), esta escolha confere maior versatilidade ao sistema, permitindo a sua adaptação a cenários futuros em que diferentes parâmetros possam ser configurados seletivamente em cada motor.

Assim, a combinação de `sync_read` para a leitura periódica do estado dos motores e `bulk_write` para o envio de comandos de controlo assegura simultaneidade, escalabilidade e robustez no controlo coordenado de um dedo da mão robótica, servindo como base sólida para a extensão posterior ao controlo integrado de múltiplos dedos. Com esta abordagem, foi possível atingir uma taxa de leitura estável de aproximadamente 50Hz, garantindo uma monitorização suficientemente rápida para aplicações em tempo quase real.

Após implementar com sucesso o envio de comandos e a leitura simultânea dos parâmetros dos motores, procedeu-se à aplicação do modo de operação selecionado: o Current-Based Position Control Mode. A adaptação do código previamente desenvolvido foi direta, bastando configurar o registo correspondente ao modo de operação e definir um limite máximo de corrente na zona de inicialização do sistema. Esta configuração é realizada através da escrita nos endereços apropriados de cada motor Dynamixel.

Nos primeiros testes, foi definido um limite de corrente de 100mA. Embora este modo de operação não permita o controlo direto da velocidade, é possível estabelecer uma velocidade máxima por meio da configuração do parâmetro `Profile Velocity`. Assim, foram enviadas posições de destino previamente definidas para provocar o movimento de fecho e posterior abertura do dedo, com o objetivo de avaliar o comportamento dos motores em condições ideais. Os gráficos resultantes desta experiência encontram-se nas Figuras 4.4, 4.5 e 4.6.

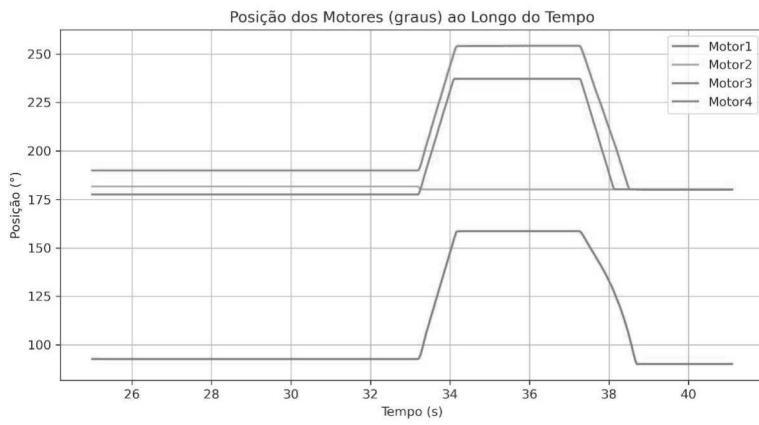


Figura 4.4: Posições dos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução

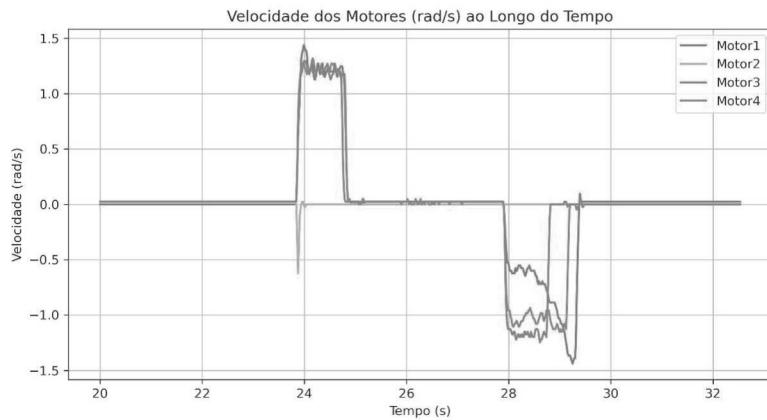


Figura 4.5: Velocidades dos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução

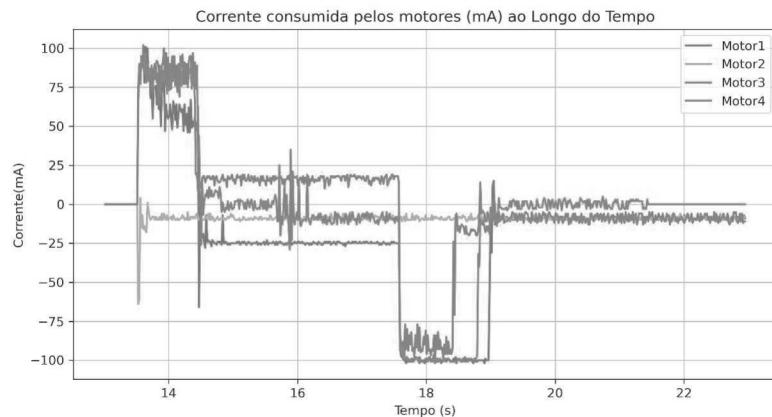


Figura 4.6: Correntes consumidas pelos motores ao longo do tempo durante o fecho e abertura do dedo sem nenhuma obstrução

Como se pode observar, durante todo o movimento, os valores de corrente e velocidade

mantiveram-se dentro dos limites especificados. A única exceção ocorreu no motor 1, onde se registou uma ligeira oscilação de velocidade, ultrapassando o valor definido em apenas 0.07 rad/s. Esta variação é considerada normal e esperada no funcionamento real dos motores, não comprometendo a estabilidade nem a segurança do sistema.

Na experiência seguinte, foram mantidas as mesmas posições de destino, mas introduziu-se um obstáculo físico que impedia a concretização do movimento pretendido. Tal como ilustrado nas Figuras 4.7, 4.8 e 4.9, a análise dos dados revela diferenças claras entre os dois cenários.

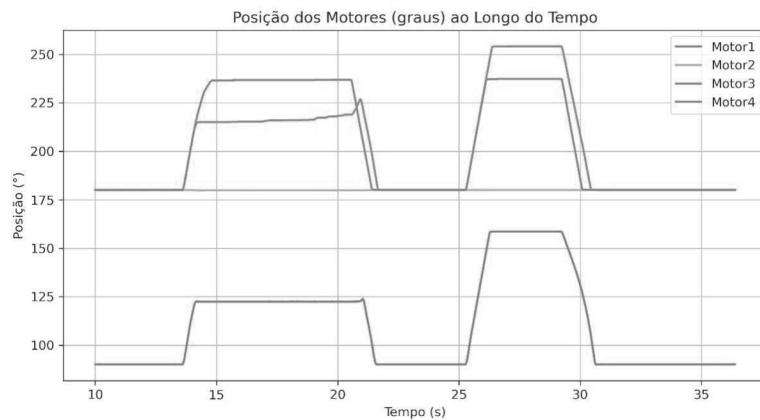


Figura 4.7: Posições dos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo

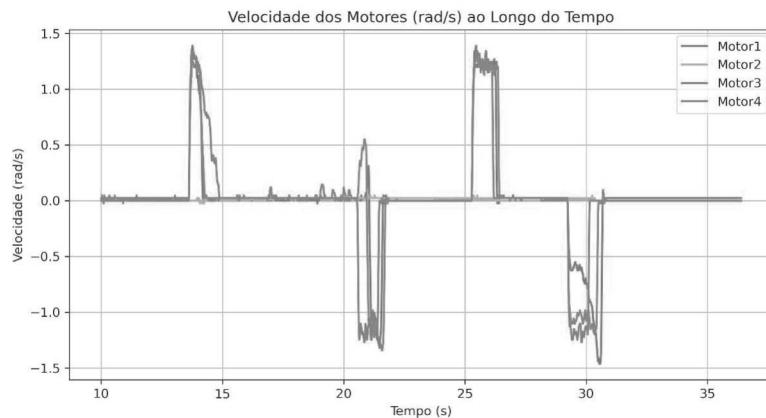


Figura 4.8: Velocidades dos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo

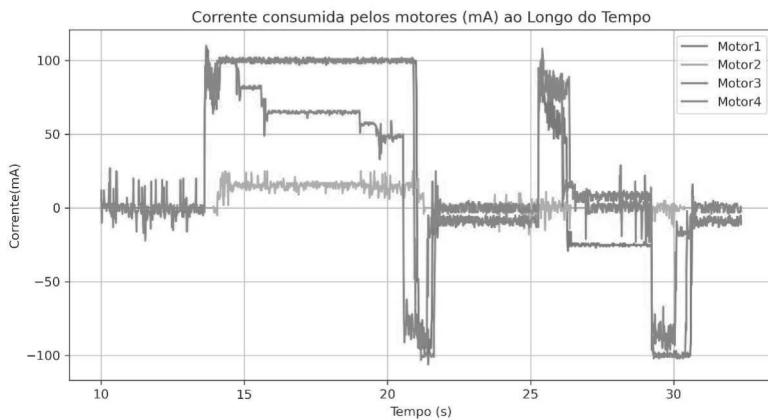


Figura 4.9: Correntes consumidas pelos motores ao longo do tempo durante o fecho e abertura do dedo com obstáculo

Na presença do obstáculo, o dedo não consegue atingir a posição alvo, resultando numa estabilização precoce da posição e no corte da velocidade. Simultaneamente, verifica-se um aumento significativo na corrente consumida, evidenciando o esforço do motor perante a resistência sem nunca ultrapassar o limite de corrente definido. Após a remoção do obstáculo, o sistema retoma o movimento normal, alcançando a posição desejada com maior estabilidade e menor consumo energético.

Estes resultados comprovam a utilidade e a eficácia do *Current-Based Position Control Mode*, permitindo limitar a força exercida pelos motores durante a interação com o ambiente. Esta funcionalidade é fundamental para aplicações em que a segurança e a adaptabilidade ao contacto com objetos são essenciais, como é o caso da manipulação robótica.

Para além das duas experiências iniciais, foram conduzidos testes adicionais com o objetivo de aprofundar a análise do comportamento dos motores em diferentes condições operacionais. Numa dessas experiências, o dedo foi submetido a movimentos de abertura e fecho com diferentes velocidades máximas configuradas. Os resultados, apresentados no Apêndice A.2.1, demonstram que velocidades mais reduzidas conduzem a um menor consumo energético e a uma variação mais suave da posição ao longo do tempo. A inclinação menos acentuada da curva de posição indica um movimento mais progressivo e controlado, confirmando a influência direta da velocidade sobre a dinâmica do sistema.

Outra experiência relevante consistiu em introduzir obstáculos que interrompessem o movimento do dedo em diferentes posições, com o intuito de estudar o efeito da localização do obstáculo no comportamento individual dos motores. Verificou-se que, quando o obstáculo atua sobre a base do dedo (afetando os motores inferiores), apenas o motor bloqueado entra em esforço, enquanto os restantes continuam a tentar atingir as suas posições alvo. Em contraste, quando o obstáculo é colocado na extremidade do dedo, todos os motores são simultaneamente afetados, ficando em esforço e impedidos de completar o movimento. Esta experiência, cujos gráficos se encontram no Apêndice A.2.2, reforça ainda a eficácia do controlo baseado em corrente, uma vez que a corrente consumida em nenhuma circunstância ultrapassou o limite máximo previamente definido.

Os dados experimentais obtidos encontram-se acompanhados de gráficos e ligações para vídeos demonstrativos no Apêndice A.2.1 e A.2.2, permitindo uma análise visual clara dos diferentes comportamentos observados. Estas experiências complementares contribuem significativamente para a validação do sistema de controlo implementado, demonstrando não só a sua eficácia em cenários ideais, mas também a sua robustez perante interações físicas imprevistas.

Programação da Mão completa

4.2.4 Arquitetura de Software Desenvolvida

4.2.5 Funcionalidades Avançadas

Ajuste dinâmico de corrente no contacto com objetos

Implementação de velocidades relativas entre dedos e falanges

Implementação de atrasos temporais programáveis entre movimentos de dedos

4.3 IMPLEMENTAÇÃO DOS SENSORES DE CONTACTO

4.3.1 Seleção do Microcontrolador

4.3.2 Implementação Física do Sistema sensorial

Unidade de Aquisição dos sensores

Desenho da peça para fixação

Desenvolvimento do PCB

4.3.3 Testes com sensores

Testes iniciais com sensores FSR

Teste com sensor FSR 406

Desenho de Peça para propagar a força para o sensor

4.4 CONCLUSÃO

Referências

- [1] *Dexterous Robotic Hand - OceanTrix Robotics Since 1996.* URL: <https://oceantrix.com/robotic-hand/> (acedido em 02/05/2025).
- [2] *Agile Hand – the human-like robotic hand.* URL: <https://www.agile-robots.com/en/solutions/agile-hand>.
- [3] H. Park e D. Kim, «An open-source anthropomorphic robot hand system: HRI hand,» *HardwareX*, vol. 7, e00100, fev. de 2020, ISSN: 2468-0672. DOI: 10.1016/J.OHX.2020.E00100.
- [4] *Robot Nano Hand - open source robot hand project.* URL: <https://robotnanohand.com/>.
- [5] A. Nurpeissova, T. Tursynbekov e A. Shintemirov, «An Open-Source Mechanical Design of Alaris Hand: A 6-DOF Anthropomorphic Robotic Hand,» *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1177–1183, mai. de 2021, ISSN: 10504729. DOI: 10.1109/ICRA48506.2021.9561977.
- [6] K. Shaw, A. Agarwal e D. Pathak, «LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning,» *Robotics: Science and Systems (RSS)*, 2023. (acedido em 02/05/2025).
- [7] Syntouch, *Biotac*, <https://www.syntouchinc.com/wp-content/uploads/2018/08/BioTac-Manual-V.21.pdf>, 2018.
- [8] L. Jamone, L. Natale, M. Giorgio e G. Sandini, «Highly sensitive soft tactile sensors for an anthropomorphic robotic hand,» *IEEE Sensors Journal*, vol. 15, pp. 4226–4233, 8 mar. de 2015. DOI: 10.1109/JSEN.2015.2417759. URL: <https://ieeexplore.ieee.org/abstract/document/7070742/>.
- [9] Y. Yan, Z. Hu, Z. Yang et al., «Soft magnetic skin for super-resolution tactile sensing with force self-decoupling,» *Science Robotics*, vol. 6, 51 fev. de 2021, ISSN: 24709476. DOI: 10.1126/SCIROBOTICS.ABC8801.
- [10] T. Kawasetsu, T. Horii, H. Ishihara e M. Asada, «Mexican-Hat-Like Response in a Flexible Tactile Sensor Using a Magnetorheological Elastomer,» *Sensors*, vol. 18, p. 587, 2 fev. de 2018, ISSN: 1424-8220. DOI: 10.3390/S18020587. URL: <https://www.mdpi.com/1424-8220/18/2/587/htm%20https://www.mdpi.com/1424-8220/18/2/587>.
- [11] A. Alfadhel, M. A. Khan, S. Cardoso, D. Leitao e J. Kosel, «A Magnetoresistive Tactile Sensor for Harsh Environment Applications,» *Sensors*, vol. 16, p. 650, 5 mai. de 2016, ISSN: 1424-8220. DOI: 10.3390/S16050650. URL: <https://www.mdpi.com/1424-8220/16/5/650/htm%20https://www.mdpi.com/1424-8220/16/5/650>.
- [12] H. Yang, L. Weng, B. Wang e W. Huang, «Design and Characterization of High-Sensitivity Magnetostrictive Tactile Sensor Array,» *IEEE Sensors Journal*, vol. 22, pp. 4004–4013, 5 mar. de 2022, ISSN: 15581748. DOI: 10.1109/JSEN.2022.3145822.
- [13] G. Canavese, S. Stassi, C. Fallauto et al., «Piezoresistive flexible composite for robotic tactile applications,» *Sensors and Actuators, A: Physical*, vol. 208, pp. 1–9, fev. de 2014, ISSN: 09244247. DOI: 10.1016/j.sna.2013.11.018.
- [14] S. Stassi, V. Cauda, G. Canavese e C. F. Pirri, «Flexible Tactile Sensing Based on Piezoresistive Composites: A Review,» *Sensors*, vol. 14, pp. 5296–5332, 3 mar. de 2014, ISSN: 1424-8220. DOI: 10.3390/S140305296. URL: <https://www.mdpi.com/1424-8220/14/3/5296/htm%20https://www.mdpi.com/1424-8220/14/3/5296>.

- [15] F. Yin, H. Niu, E. S. Kim, Y. K. Shin, Y. Li e N. Y. Kim, «Advanced polymer materials-based electronic skins for tactile and non-contact sensing applications,» *InfoMat*, vol. 5, e12424, 7 jun. de 2023, ISSN: 2567-3165. DOI: 10.1002/INF2.12424. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/inf2.12424%20https://onlinelibrary.wiley.com/doi/abs/10.1002/inf2.12424%20https://onlinelibrary.wiley.com/doi/10.1002/inf2.12424>.
- [16] L. Seminara, L. Pinna, M. Capurro e M. Valle, «A Tactile Sensing System Based on Arrays of Piezoelectric Polymer Transducers,» em *Smart Actuation and Sensing Systems - Recent Advances and Future Challenges*. IntechOpen, out. de 2012, pp. 611–638, ISBN: 978-953-51-0798-9. DOI: 10.5772/50112. URL: <https://www.intechopen.com/chapters/39972%20undefined/chapters/39972>.
- [17] Y. Zhong, J. Wang, L. Han et al., «High-performance flexible self-powered triboelectric pressure sensor based on chemically modified micropatterned PDMS film,» *Sensors and Actuators A: Physical*, vol. 349, p. 114013, jan. de 2023, ISSN: 0924-4247. DOI: 10.1016/J.SNA.2022.114013.
- [18] K. Tao, Z. Chen, J. Yu et al., «Ultra-Sensitive, Deformable, and Transparent Triboelectric Tactile Sensor Based on Micro-Pyramid Patterned Ionic Hydrogel for Interactive Human–Machine Interfaces,» *Advanced Science*, vol. 9, p. 2104168, 10 jan. de 2022, ISSN: 2198-3844. DOI: 10.1002/ADVS.202104168. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/advs.202104168%20https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202104168%20https://onlinelibrary.wiley.com/doi/10.1002/advs.202104168>.
- [19] A. Ke, J. Huang, L. Chen et al., «Fingertip Tactile Sensor with Single Sensing Element Based on FSR and PVDF,» *IEEE Sensors Journal*, vol. 19, pp. 11100–11112, 23 dez. de 2019, ISSN: 15581748. DOI: 10.1109/JSEN.2019.2936304. URL: <https://ieeexplore.ieee.org/document/8805357>.
- [20] X. Lu, D. Sun, H. Yin et al., «3-D Tactile-Based Object Recognition for Robot Hands Using Force-Sensitive and Bend Sensor Arrays,» *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, pp. 1645–1655, 4 dez. de 2023, ISSN: 23798939. DOI: 10.1109/TCDS.2022.3215021.

APÊNDICE A

Apêndice

A.1 MONTAGEM DOS DEDOS

A.1.1 Montagem do Polegar

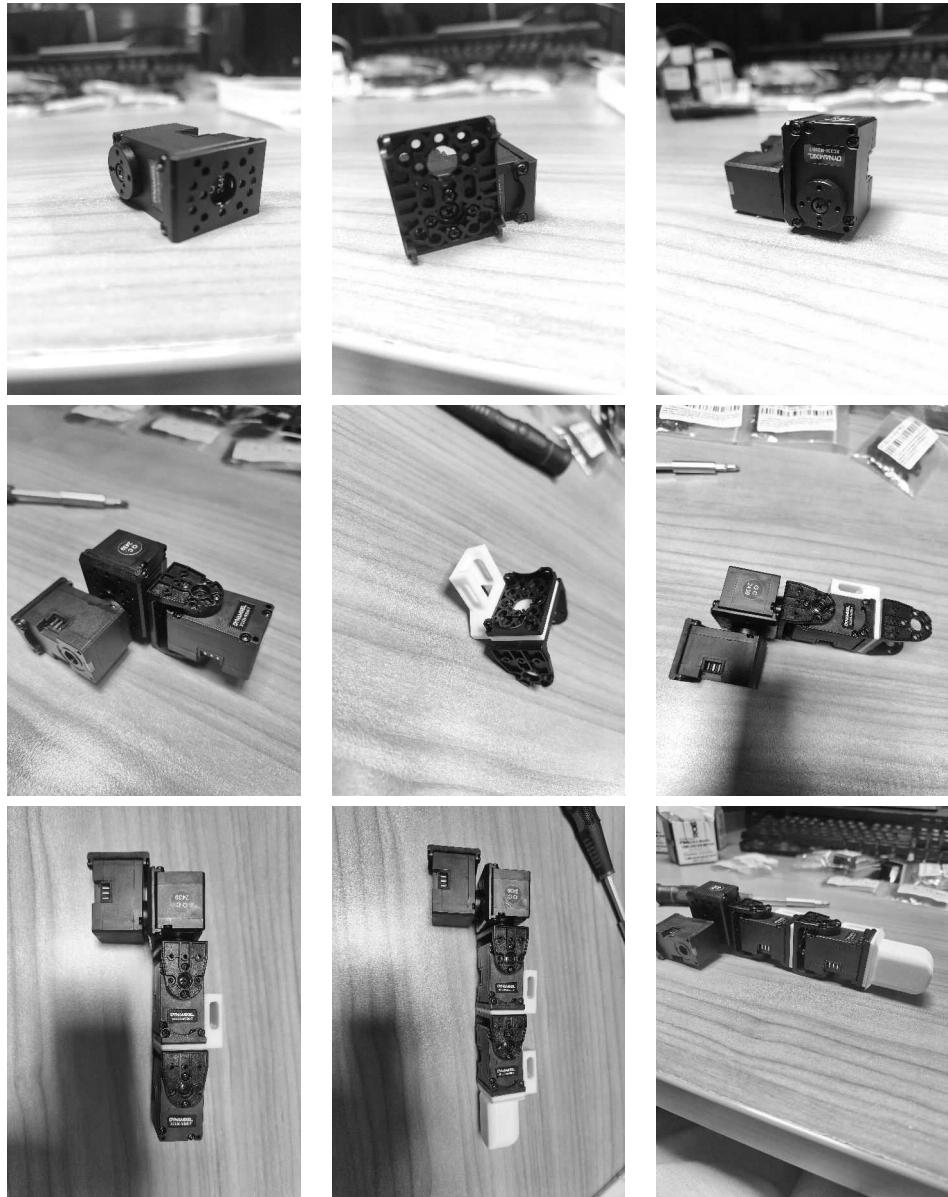


Figura A.1: Fotografias sequenciais da montagem do polegar

A.1.2 Montagem dos restantes dedos

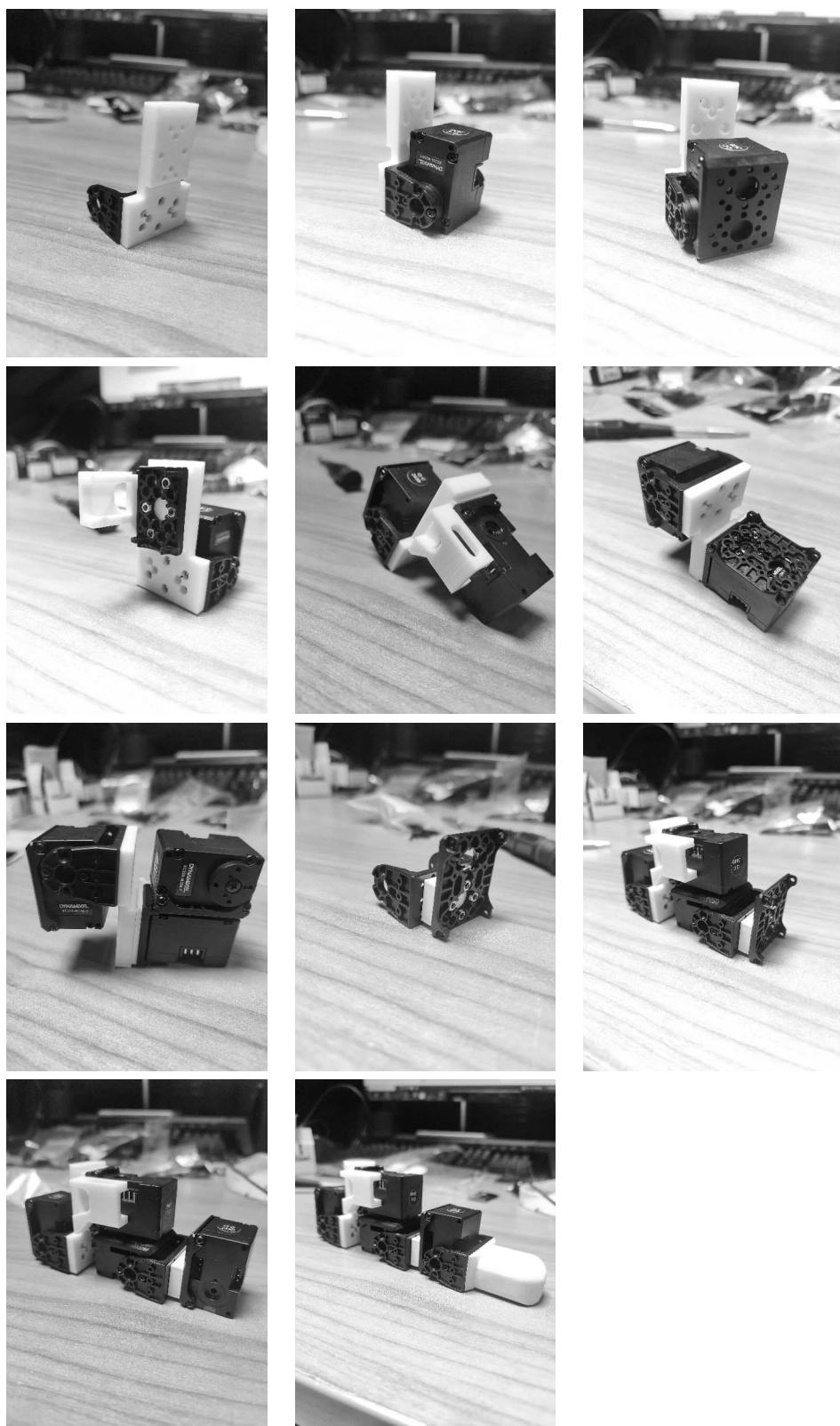


Figura A.2: Fotografias sequenciais da montagem do dedo referente ao dedo indicador, médio ou anelar

A.1.3 Hiperligações

A.2 TESTES REALIZADOS COM UM DEDO

A.2.1 Teste com duas velocidades diferentes

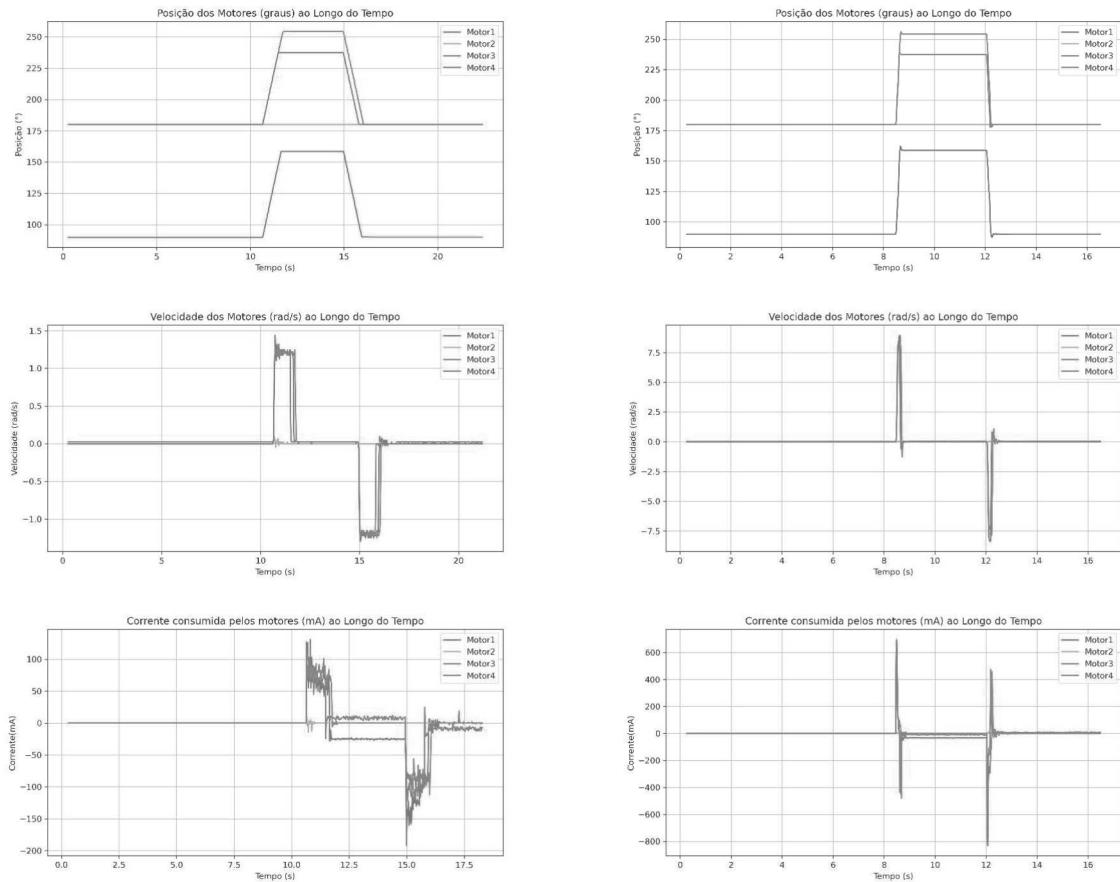


Figura A.3: Gráficos da posição, velocidade e corrente consumida pelos motores de um dedo. A coluna da esquerda corresponde a uma velocidade máxima configurada de 1.2 rad/s e a coluna da direita a 23.98 rad/s.

Hiperligação para vídeo: https://www.youtube.com/shorts/_319AzUPu9A

A.2.2 Teste com obstáculo em posições diferentes

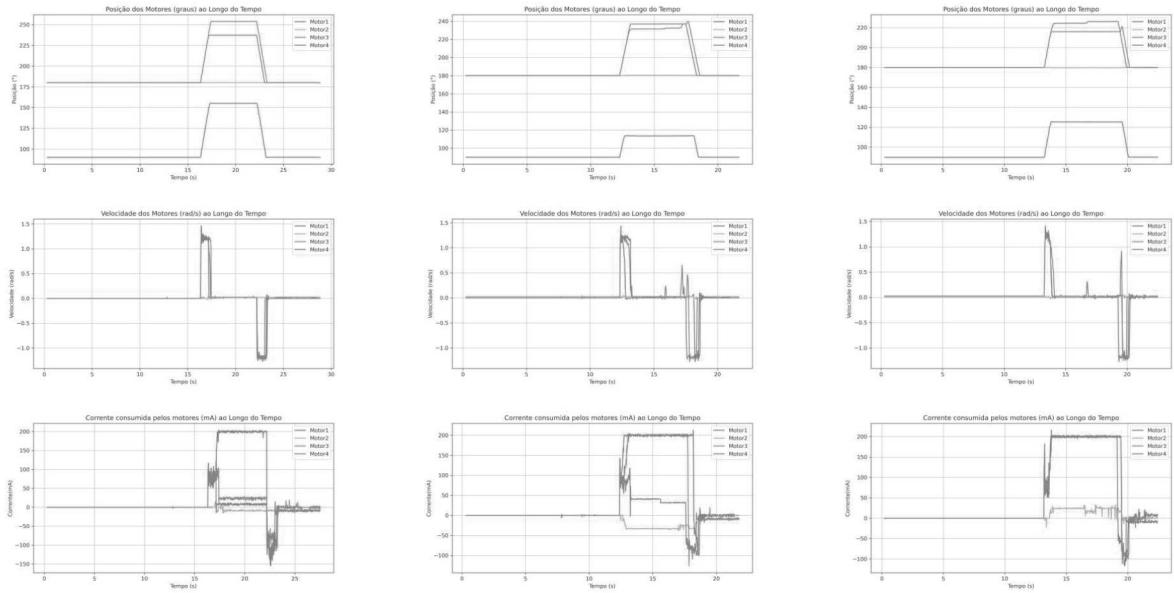


Figura A.4: Gráficos da posição, velocidade e corrente consumida pelos motores de um dedo, em diferentes cenários de obstrução. A coluna da esquerda representa a presença de um obstáculo junto ao motor mais próximo da base do dedo, a coluna central refere-se à obstrução a meio do dedo, e a coluna da direita corresponde à interferência na ponta do dedo.

Hiperligação para vídeo: <https://www.youtube.com/shorts/Qy4lXXRUP4Q>