

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Projeto, modelo e construção de um manipulador com elevado grau de redundância

Joaquim Manuel Costa Alves Duarte Ribeiro

PARA APRECIÇÃO POR JÚRI

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Paulo José Cerqueira Gomes da Costa

Co-orientador: Professor Doutor José Luis Sousa de Magalhães Lima

27 de Outubro de 2016

Resumo

O desenvolvimento científico e tecnológico associado ao mundo da robótica aponta para um manipulador capaz de se adaptar a diversas aplicações e cenários, permitindo executar tarefas em situações em que um manipulador clássico não é capaz. Tipicamente denominado redundante ou hiper-redundante, um manipulador com maior número de graus de mobilidades do que graus necessários para cumprir uma determinada tarefa, vem aumentar a performance em ambientes de trabalho complexos e com múltiplos obstáculos, como, por exemplo, na manipulação de um elevado número de peças, onde existam zonas que deixam de ser atingíveis por partes de um manipulador não redundante, pois embateria contra peças já posicionadas. O objetivo principal do projeto de dissertação será construir um manipulador robótico de baixo custo capaz de se movimentar num espaço reduzido com obstáculos e atingir um vasto número de posições. Inicialmente, foi desenhada em OpenSCAD, impressa e validada a estrutura mecânica do manipulador que serviu como base do protótipo. Foi projetado e validado um sistema de atuação das articulações, tendo como referência a utilização de servomotores e o seu controlo em posição.

Abstract

The scientific and technological development together with the world of robotics points to a manipulator able to adapt to various applications and scenarios, allowing to perform tasks in certain context that an usual manipulator is not capable. Typically called redundant or hyper-redundant, a robotic manipulator with more degrees of mobility than required necessary to manage a certain task, enhances the performance in complex working environments with multiple obstacles, such as in handling a large number of objects, where there are areas which are no longer are attainable by parts of a non-redundant manipulator, as a result, it would collide against objects already positioned. The main objective of this dissertation project will be building a low cost robotic manipulator capable of moving in a small space with obstacles and still reach a large number of positions. Initially, it was designed in OpenSCAD, printed and validated the mechanical structure of the manipulator that was the basis to the prototype. It was developed and verified a joint motion system, based on the use of servomotors and their position control.

Agradecimentos

Gostaria de expressar os meus agradecimentos ao meu orientador, Professor Doutor Paulo Costa, pelos seus conselhos, orientação e encorajamento no desenvolvimento desta dissertação.

Um agradecimento ao meu co-orientador, Professor Doutor José Lima pela sua disponibilidade e esclarecimentos.

À Alexandra, um agradecimento único, pela paciência, compreensão, apoio, dedicação e estabilidade que me tem dado ao longo destes anos.

Aos meus pais, em especial, e ao resto da minha família pelo apoio e incentivo incondicional.

Ao Alexandre, pelo esforço, contribuição, companheirismo e apoio.

Ao Ricardo, pela ironia, churrascos e motivação.

Ao Cisco, ao Sebe, ao Pires e ao Joni, pela lufada de ar fresco.

Ao Bruno, ao Tiago e ao Gabriel, pelos fins de semana e churrascos.

A todos os meus colegas da FEUP, pelo apoio e motivação.

Ao Núcleo do IEEE da Universidade do Porto, em especial ao Diogo Eiras, e ao Núcleo de Estudantes de Engenharia Electrotécnica da FEUP, pelo extra-faculdade.

Ao Dr. Isidro, ao Sr. Fernando e ao Jorge, pela disponibilidade.

À UT, pela reintegração e motivação.

Ao Ruben, ao Gil, ao Rui, ao Paulino, ao Sousa e ao Edgar, por toda a motivação na área.

Joaquim Duarte Ribeiro

*“If my doctor told me I had only six minutes to live, I wouldn’t brood.
I’d type a little faster.”*

Isaac Asimov

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	2
1.3	Objetivos	3
1.4	Estrutura da Dissertação	3
2	Revisão de Literatura	5
2.1	Introdução	5
2.1.1	<i>Design</i> de Manipuladores	6
2.2	Tipos de Manipuladores	7
2.2.1	Classificação por Atuação	7
2.2.2	Classificação por DOF	7
2.2.3	Classificação por Volume de Trabalho	7
2.3	Sistemas de Atuação	9
2.3.1	Motores Elétricos	10
2.3.2	Controlo	12
2.4	Controlador	12
2.5	Protocolos de Comunicação	12
2.5.1	I ² C	13
2.6	<i>Software</i> relevante	13
2.6.1	CAD	13
2.6.2	Simuladores	15
2.7	Resumo	17
3	Metodologia	19
3.1	Parâmetros a considerar	20
3.1.1	Configuração 1	21
3.1.2	Configuração 2	22
3.2	Componentes do Manipulador	24
3.2.1	Servomotores	24
3.2.2	Estrutura do Manipulador	25
3.2.3	Controlador	25
3.2.4	<i>Driver</i> do Motor	25
3.2.5	Fonte de Alimentação	26
3.3	Arquitetura de Controlo	27

4	Sistema de Atuação	29
4.1	Introdução	29
4.2	Modelo	29
4.2.1	Limitações	30
4.3	Testes e parametrizações	30
4.3.1	Forças de atrito	31
4.4	Modelo de Simulação	33
4.4.1	Servomotor	33
4.4.2	Manipulador	34
4.5	Controlador	35
4.5.1	Soluções comuns	35
4.6	Controlo Implementado	37
4.6.1	Controlo em Posição	37
4.6.2	Controlo em Posição e em Velocidade	39
4.7	Resumo	40
5	Construção da Estrutura Mecânica	43
5.1	Introdução	43
5.2	Impressão 3D	43
5.3	Prototipagem	44
5.4	Resultados	49
6	Validação Experimental	55
6.1	Introdução	55
6.2	Validação da Estrutura Mecânica	56
6.3	Validação da Movimentação	57
6.4	Teste de Posição	60
6.5	Validação das Trajetórias	61
6.6	Resultados	61
7	Conclusões e Trabalho Futuro	63
7.1	Trabalho Futuro	63
	Referências	65

Lista de Figuras

2.1	Manipulador Antropomórfico (RRR), retirado de [1].	8
2.2	Manipulador Esférico (RRP), retirado de [2].	8
2.3	Manipulador SCARA (RRP), retirado de [1].	8
2.4	Manipulador Cilíndrico (RPP), retirado de [1].	9
2.5	Manipulador Cartesiano (PPP), retirado de [1].	9
2.6	Controlo típico usado em servomotores.	12
2.7	Esquema de ligações com um mestre e três escravos - I^2C	13
2.8	OpenSCAD logo, retirado de OpenSCAD website	14
2.9	Interface e estrutura exemplo de um programa em OpenSCAD.	15
2.10	Típica simulação no SimTwo, retirada do website	17
3.1	Metodologia da dissertação.	19
3.2	Primeira configuração.	20
3.3	Segunda configuração.	21
3.4	Cálculo do binário.	22
3.5	Arduino Mega, retirado de Arduino website	25
3.6	MotorShield para Arduino, retirado de Adrafruit website	26
3.7	16 PWM <i>shield</i> para Arduino, retirado de Adrafruit website	26
3.8	Níveis de controlo.	27
4.1	Esquema elétrico do modelo do motor DC.	29
4.2	Cálculo da constante K do servo, retirado de [3].	31
4.3	Modelo da força de atrito viscoso, retirado de [3].	32
4.4	Modelo da força de atrito estático, retirado de [3].	32
4.5	Modelo da força de atrito final, retirado de [3].	33
4.6	Arquitetura básica do simulador.	33
4.7	Modelo do manipulador no SimTwo. Posição <i>home</i>	34
4.8	Modelo do manipulador no SimTwo após rotações.	35
4.9	Etapas de inicialização.	38
4.10	Esquema de controlo utilizado para movimentar as articulações.	38
4.11	Sistema a controlar.	39
4.12	Esquema de controlo utilizado para movimentar as articulações.	40
5.1	<i>Hinge</i> e <i>Sleeve</i> , peças básicas para interligar os diversos servomotores.	45
5.2	<i>Hinge</i> e <i>Sleeve</i> unidas por um retângulo.	45
5.3	Ligação.	46
5.4	Ligação entre duas peças da estrutura.	46
5.5	Dimensões do servomotor exemplo SG90.	47
5.6	Base projetada.	47

5.7	Interligação entre a base e a estrutura.	48
5.8	Estrutura em OpenScad.	49
5.9	<i>Hinge</i> e a base após impressão.	50
5.10	<i>Sleeve</i> após impressão.	50
5.11	Peça de suporte na base do servomotor que a interliga com a dobradiça.	50
5.12	<i>Hinge</i> e <i>sleeve</i> com ligação retangular entre as duas após impressão.	51
5.13	Base e primeiros servomotores.	51
5.14	Ligação após impressão.	52
5.15	Interligação entre a estrutura e a garra.	52
5.16	Estrutura após impressão.	53
6.1	Análise de requisitos dos servomotores utilizados.	56
6.2	Rotação de 90° da <i>hinge</i>	56
6.3	Rotação de 90° da <i>sleeve</i>	57
6.4	Movimentação para -90°.	58
6.5	Movimentação para 90°.	58
6.6	Movimentação para -90°.	59
6.7	Movimentação para 90°.	59
6.8	Movimentação complexa.	60

Lista de Tabelas

2.1	Configurações e o seu Volume de Trabalho.	9
3.1	Dimensões para cálculo do binário.	21
3.2	Binário necessário para cada articulação.	22
3.3	Dimensões para cálculo do binário.	23
3.4	Binário necessário para cada articulação.	23
3.5	Binário necessário para cada articulação.	24
3.6	Servomotores escolhidos.	24
3.7	Corrente de <i>Stall</i>	27
4.1	Parâmetros do servomotor MG995.	31
6.1	Desvios de cada articulação.	61

Abreviaturas e Símbolos

Lista de Abreviaturas:

AC	<i>Alternating Current</i>
AGV	<i>Automated Guided Vehicle</i>
CAD	<i>Computer-Aided Design</i>
DC	<i>Direct Current</i>
DOF	<i>Degrees of Freedom</i>
I ² C	<i>Inter-Integrated Circuit</i>
IFR	<i>International Federation of Robotics</i>
MIEEC	<i>Master in Electrical and Computers Engineering</i>
ODE	<i>Open Dynamics Engine</i>
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i>
P	<i>Prismatic</i>
PWM	<i>Pulse-Width Modulation</i>
R	<i>Revolution</i>
RI/SME	<i>The Robotics International Division of the Society of Manufacturing Engineers</i>
ROS	<i>Robotic Operating System</i>
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
XML	<i>eXtensible Markup Language</i>

Lista de Símbolos:

ω	Frequência Angular
α	Ângulo
A	Ampere
μ	Micro (10^{-6})
V	Volt

Capítulo 1

Introdução

O presente capítulo apresenta um breve contexto e a motivação para o desenvolvimento do tema de dissertação, assim como os objetivos propostos e termina com a organização estrutural dos restantes capítulos.

1.1 Contexto

O desenvolvimento científico e tecnológico associado ao mundo da robótica é recente e encontra-se em constante evolução motivada pela necessidade e ambição do ser humano em encontrar soluções alternativas que o beneficiem. Sendo transversal a diferentes áreas, permite uma correlação entre Engenharia Electrotécnica, Engenharia Mecânica, Engenharia Informática, entre outras.

A RI/SME define *Robot* como manipuladores que possuem um sistema de controlo integrado e capaz de operar de forma independente.

Um manipulador robótico surge ainda definido como: “um manipulador programável controlado automaticamente, reprogramável e multifuncional, em 3 ou mais eixos, que tanto pode estar fixo num lugar como móvel para aplicações de automação industrial”[4].

Tipicamente, um manipulador é constituído por uma estrutura mecânica, um sistema de atuação e um sistema de controlo. A estrutura mecânica pode ser dividida em três categorias: a) discreta - manipulador com ligações rígidas -; b) contínua - de flexibilidade contínua -; c) em cascata - normalmente associada a estruturas geométricas fechadas e ativadas em paralelo [5].

Para esta dissertação só serão relevantes os corpos ou segmentos rígidos que devido à sua multifuncionalidade, elevados níveis de produtividade e de precisão, têm suscitado um elevado interesse por parte da indústria de produção. Alguns exemplos de tarefas na indústria de produção:

- Pintura;
- Montagem de peças;
- Inspeção;
- Perfuração.

Atualmente, o desenvolvimento aponta para uma solução capaz de se adaptar a diversas aplicações e cenários, permitindo executar tarefas em situações em que um manipulador clássico não é capaz.

Os manipuladores robóticos industriais dispõem, normalmente, até seis graus de liberdade, que são os necessários para, dentro do seu volume de trabalho, posicionar a ferramenta num ponto requerido no espaço e também orientá-la segundo a direção desejada. Isso requer, pelo menos, seis atuadores, geralmente motores trifásicos AC ou DC, de elevado custo. Porém as possíveis configurações em que o manipulador tem de se posicionar para atingir esse objetivo muitas vezes não são viáveis caso haja a presença de restrições no ambiente de trabalho.

Para ultrapassar este problema, existem diversos projetos de investigação em robôs redundantes [6] [7] [8] [9], isto é, um manipulador robótico cujo número de graus de liberdade de atuação excede o número mínimo necessário para executar uma determinada tarefa [10] [11]. Esta redundância vem aumentar a performance em ambientes de trabalho complexos e com múltiplos obstáculos, como por exemplo na manipulação de um elevado número de peças onde existem zonas que deixam de ser atingíveis por partes de um manipulador não redundante pois embateria contra peças já posicionadas.

O estudo de robôs hiper-redundantes pode ser dividido em dois níveis. O primeiro será encarregue da estrutura mecânica e da coordenação dos seus atuadores para que o manipulador se consiga movimentar. Este inclui movimentação em superfícies planas e no espaço, as trajetórias genéricas e o controlo dos atuadores para cumprir as movimentações.

O segundo, num nível de abstração superior, estará o planeamento de trajetórias, desviar de obstáculos e outras tarefas.

Esta dissertação concentra-se no primeiro nível.

1.2 Motivação

Normalmente, um manipulador hiper-redundante pode ser identificado pela sua estrutura mecânica, locomoção, movimentação no espaço a três dimensões, pelos seus atuadores, engrenagens e pelos métodos de resolução da cinemática e da dinâmica do sistema.

A literatura providencia bastantes exemplos de robôs adaptados de animais com morfologia redundante quer terrestres quer aquáticos. Aliás, os robôs hiper-redundantes começaram por ser designados como *snake like*, *tentacles* ou *elephants's trunks* devido à sua forma [5].

Teoricamente, um manipulador redundante pode ter um número infinito de configurações para a mesma posição da ferramenta. Os graus extra permitem evitar obstáculos e singularidades cinemáticas, isto é, restrições de uma só solução, permitem a otimização do movimento em função do custo e permitem aumentar o volume de trabalho [12].

Em suma, quanto mais elevado for esse grau de redundância mais adaptável será o posicionamento do manipulador e, conseqüentemente, haverá uma menor degradação da área de trabalho à medida que vão sendo posicionadas peças.

No entanto, existem sempre restrições em diversos factores como o tamanho, o peso e a potência do manipulador que irão afetar, principalmente, as articulações.

Este sistema pode ser utilizado, por exemplo, para aplicações no âmbito militar, aeroespacial [13], médicas [14], de salvamento [15], entre outras.

De forma a garantir um desempenho dentro das expectativas de um manipulador robótico a movimentar-se no espaço a três dimensões serão usados servomotores especialmente projetados para robótica ou melhorando servos de modelismo, possibilitando um controlo dos motores mais adequado e uma realimentação da posição. Com estes atuadores é possível construir um protótipo à escala que possibilite analisar diferentes configurações, identificar o seu modelo dinâmico e testar os algoritmos de movimentação e posicionamento.

1.3 Objetivos

O objetivo principal do projeto de dissertação será construir um manipulador robótico capaz de se movimentar num espaço reduzido com obstáculos e atingir um vasto número de posições.

O desenvolvimento deste projeto terá o seguinte conjunto de objetivos secundários:

- Estudo de Arte em Manipuladores hiper-redundantes, isto é, com um número elevado de graus de liberdade;
- Testes e parametrização dos Servomotores;
- Desenvolvimento do sistema de controlo dos motores;
- Implementação e simulação do protótipo do manipulador em software;
- Construção e integração dos vários módulos para validações finais.

Adicionalmente, será um requisito garantir que a solução seja de baixo custo comparativamente com os manipuladores existentes.

1.4 Estrutura da Dissertação

Os restantes capítulos desta dissertação são divididos em três grandes áreas: sistema de atuação, desenvolvimento do modelo do manipulador e a construção da estrutura mecânica do protótipo.

O capítulo 2 providencia uma revisão à literatura e ao trabalho similar em robôs hiper-redundantes e em sistemas de controlo de motor para uma melhor compreensão do documento.

O capítulo 3 apresenta a metodologia de trabalho ao longo da dissertação, parâmetros relevantes para a escolha dos diversos constituintes do manipulador, a própria escolha e a arquitetura de controlo.

O capítulo 4 apresenta o sistema de atuação, o seu modelo e controlo.

O capítulo 5 detalha a projeção das peças em CAD e o seu resultado após impressão.

No capítulo 6 é feita a validação experimental do sistema desenvolvido.

Por fim, o capítulo 7 apresenta as conclusões da dissertação e sugestões para trabalho futuro.

Capítulo 2

Revisão de Literatura

Neste capítulo é apresentada a revisão de literatura no âmbito do tema desta dissertação.

A secção 2.1 apresenta uma breve introdução à projeção da construção, desenvolvimento e implementação de manipuladores robóticos e os desafios inerentes.

Na secção 2.2 são abordados os tipos de manipuladores existentes, nomeadamente o robô hiper-redundante.

Na secção 2.3 é feita uma introdução a atuadores, incluindo os passos necessários para controlar e estimar um motor DC a partir do seu modelo.”, deveria estar “Na secção 2.3 é feita uma introdução a atuadores.

Na secção 2.4 é apresentada a plataforma que permite o controlo de movimentação.

Em seguida, secção 2.5 são apresentados os protocolos de comunicação necessários para a comunicação entre o controlador e os atuadores, assim como entre a interface e o controlador.

A secção 2.6 apresenta o *software* relevante para a dissertação.

O capítulo termina com um 2.7 resumo do capítulo.

2.1 Introdução

Este capítulo visa a introduzir todos os conceitos fundamentais para a projeção e construção de um manipulador robótico de arquitectura série, do seu sistema de atuação e controlo.

- o número de graus de liberdade;
- o volume de trabalho;
- a capacidade de carga;
- a velocidade;
- a repetibilidade e a precisão;
- a cinemática;
- a dinâmica;

- entre outros.

A capacidade de carga depende da sua estrutura, sistema de alimentação e dos atuadores. A carga a ser colocada nos atuadores e nos seus sistemas de transmissão depende da configuração do manipulador, da percentagem de tempo a suportar a mesma e do binário fornecido pelo atuador [16].

A velocidade do manipulador tende a ser visivelmente lenta pois depende do ciclo do processo e da tarefa a desempenhar. No entanto, é necessário procurar diminuir o tempo necessário.

Repetibilidade e precisão são das características que mais determinam o preço de um manipulador [16]. A precisão depende do objetivo da tarefa e não tanto do *design* do robô pois pode ser obtida através de afinações.

A dinâmica de um robot pode ser separada em termos das propriedades dos movimentos que dependem da geometria da sua estrutura mecânica, designada cinemática, e das que dependem das forças que atuam no sistema, conhecida como cinética [17].

Serão então analisadas algumas ferramentas e critérios usados no *design* mecânico e na avaliação de performance de manipuladores.

2.1.1 *Design* de Manipuladores

A construção de manipuladores robóticos é, de forma geral, controlada por empresas como **KUKA**, **ABB**, **OMRON** e **YASKAWA**. No entanto, para diversos projetos académicos, têm vindo a ser construídos alguns manipuladores que se centram em quatro categorias: 1) a estrutura física do manipulador e sensores internos; 2) a ferramenta; 3) sensores externos e sistemas de visão; 4) o controlador [16]. Esta dissertação inclui o tópico 1) e 4) nos seus objetivos.

A estrutura física contém barras, ligações, veios, entre outros, que permitem construir o seu esqueleto. Os motores e outros elementos que permitem a movimentação, são designados por atuadores[17].

É possível então identificar um conjunto de passos sequenciais para a "Projeção, construção e modulação de um manipulador com elevado grau de redundância":

1. Definição e análise da estrutura cinemática;
2. Definição do sistema de atuação;
3. Modelação e Estimação do Sistema de Atuação;
4. Controlo de Movimentação;
5. Construção da Estrutura Mecânica;
6. Testes e Afinação.

2.2 Tipos de Manipuladores

Presentemente estes manipuladores robóticos são classificados e distinguidos pela sua geometria construtiva, pelo que existem dois grandes tipos de manipuladores: em série e em paralelo. A arquitetura em série distingue-se pela sucessão de corpos rígidos desde da base do robot até ao ponto de ferramenta (entenda-se por este ponto de ferramenta, o atuador que permita a manipulação dos objetos). A arquitetura em paralelo consiste num robot composto por uma ferramenta com vários graus de liberdade e uma base fixa, ligado entre si por cadeias cinemáticas independentes [18].

Atendendo ao custo inferior dos robots em série comparado com os robots em paralelo, este projeto utilizará robots com configuração em série.

No entanto, ainda podem ser classificados por graus de liberdade, atuadores, volume de trabalho, entre outros.

2.2.1 Classificação por Atuação

A mobilidade é assegurada pela presença de articulações que se encontram entre dois *links* rígidos.

Estas podem ser de revolução, prismáticas, de parafuso, cilíndricas, esféricas, entre outras. As mais usais são a articulação prismática que garante um movimento linear enquanto que a de revolução garante um movimento rotativo. Cada uma garante um grau de mobilidade que deve ser devidamente distribuído ao longo da estrutura mecânica para garantir graus de liberdade necessários para uma determinada tarefa [19].

Os três atuadores mais comuns são os hidráulicos, os pneumáticos e os electromagnéticos [19].

2.2.2 Classificação por DOF

Graus de Liberdade ou *Degrees of Freedom* (DOF) são o número de parâmetros independentes pertencentes a uma configuração, entenda-se configuração como a localização de cada articulação do manipulador e espaço de configuração como o conjunto de todas as configurações [2].

Um objeto rígido num espaço tridimensional necessita assim de seis DOF, 3 para a orientação e 3 para a posição. Um manipulador planar é representado por pelo menos três DOF.

Um manipulador com mais graus de mobilidade que graus de liberdade é vulgarmente designado por manipulador cinemáticamente redundante.

Se o número de graus de liberdade que o robô possui ultrapassa largamente os necessários para uma tarefa, então ele é hiper-redundante.

2.2.3 Classificação por Volume de Trabalho

Abaixo apresentam-se as principais características dos robots com arquitetura em série. Exemplos deste tipo de arquitetura são os manipuladores antropomórficos (RRR) (figura 2.1), esféricos

(RRP) (figura 2.2), SCARA (RRP) (figura 2.3), cilíndricos (RPP) (figura 2.4) ou cartesianos (PPP) (figura 2.5) [2].

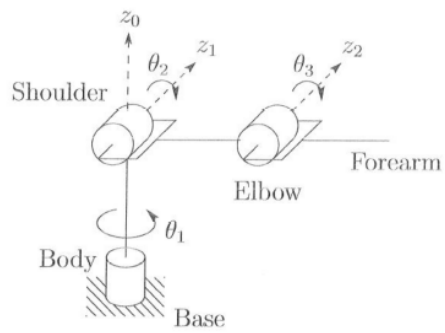


Figura 2.1: Manipulador Antropomórfico (RRR), retirado de [1].

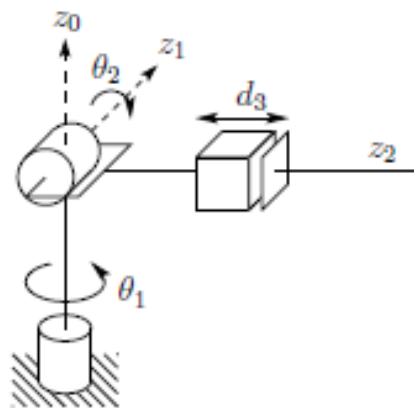


Figura 2.2: Manipulador Esférico (RRP), retirado de [2].

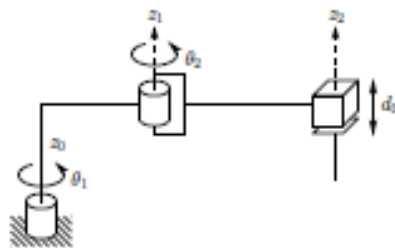


Figura 2.3: Manipulador SCARA (RRP), retirado de [1].

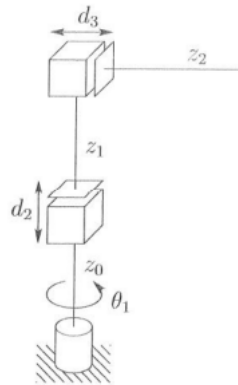


Figura 2.4: Manipulador Cilíndrico (RPP), retirado de [1].

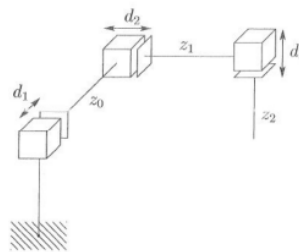


Figura 2.5: Manipulador Cartesiano (PPP), retirado de [1].

O volume em que a ferramenta pode interagir com o ambiente é denominado *workspace* ou volume de trabalho [2]. Este é limitado pelas articulações e pela geometria do manipulador. Assim, as configurações acima correspondem ao seguinte volume de trabalho 2.1:

Tabela 2.1: Configurações e o seu Volume de Trabalho.

Antropomórfico	$V = 33,41 * L * L * L$
Esférico	$V = 29,32 * L * L * L$
SCARA	$V = 12,56 * L * L * L$
Cilíndrico	$V = 9,42 * L * L * L$
Cartesiano	$V = L * L * L$

2.3 Sistemas de Atuação

Tal como num ser humano, são as articulações que provocam o movimento num robô e são os atuadores fornecem a força motriz necessária para a movimentação. Tipicamente, um sistema de atuação é caracterizado pelo motor, redução, transmissão, fonte de alimentação e controlador. No caso específico da dissertação, parâmetros como redução e transmissão não serão analisados.

Os tipos de atuadores mais comuns são os hidráulicos, os pneumáticos e os elétricos, tal como referido na subsecção 2.2.1.

Os acionamentos hidráulicos possuem alto binário, precisão de operação e podem manter um binário elevado para um longo período de tempo. No entanto, necessitam de uma fonte de energia cara, assim como as válvulas de precisão, a manutenção é cara e tem fugas de óleo[1].

Os acionamentos pneumáticos, por sua vez, podem funcionar a velocidades altas, têm um custo relativamente baixo, fácil manutenção, podem manter um binário constante numa grande faixa de velocidades e quando parado, pode manter o binário por longos períodos de tempo. No entanto, não possui alta precisão e está sujeito a vibrações quando o motor ou cilindro é parado[1].

Por último, o elétrico é o mais eficiente e possui a maior precisão, tem uma estrutura simples e de fácil manutenção, custo relativamente baixo. Não consegue manter um binário constante nas mudanças de velocidade de rotação, é frágil e tem uma baixa razão de potência de saída do motor em relação ao seu peso, necessitando de motores de elevado porte[1].

2.3.1 Motores Elétricos

Os motores elétricos têm sido bastante utilizados em sistemas mecânicos devido à sua estrutura simples, controlo acessível e baixo custo. Podem ser atuadores rotativos ou lineares. Para o caso desta dissertação apenas interessam os atuadores rotativos que estão divididos em motores de corrente alternada, motores de corrente contínua, motores de passo e servomotores.

Um motor de corrente contínua (CC) ou no inglês *direct current* (DC), é constituído por dois fios de alimentação (power e ground). Quando é fornecida alimentação, o motor entra em rotação contínua.

Usando uma modulação de largura de impulso para controlar a alimentação fornecida ao motor, é possível variar a sua velocidade do motor. No entanto, apenas se está a ligar e a desligar o motor a elevada frequência e esta pode ser limitada.

Os motores de íman permanente são normalmente utilizados para *Robot Toys* e para *Hobby Robots*.

2.3.1.1 Servomecanismos

Um mecanismo que incorpora "servo" na sua descrição, implica malha fechada ou controlo por *feedback* do binário, velocidade ou posição [20]. Este controlo corrige o erro gerado pela diferença entre o sinal de referência e o sinal obtido através da medição do parâmetro de saída. O servomecanismo monitoriza de forma continuada a performance do sistema, o que torna o sistema controlado muito estável, preciso e com *performances* consistentes [].

2.3.1.2 Servomotor

Um servomotor é um atuador rotativo, alimentado por corrente que recebe um sinal elétrico e produz um espaçamento angular ou roda a uma certa velocidade. Pode conter sensores embebidos

para feedback de posição ou velocidade, o que permite controlo de malha fechada da posição angular.

Servomotores de modelismo, são constituídos por um motor DC, um conjunto de engrenagens, um circuito de controlo e um sensor de posição, normalmente um potenciómetro.

O circuito de controlo é responsável por garantir que o motor roda até à posição desejada com precisão. Recebe três fios como entradas: dois deles de alimentação e, consequentemente, sempre alimentados, e o terceiro é o de controlo de posição, através da modulação de pulsos.

O ângulo de rotação é limitado, dependendo do modelo. Para rodar, o motor recebe um comando de PWM. No entanto, devido ao sistema de controlo embebido, a duração do pulso positivo enviado determina a posição, tornando-o mais preciso, sendo o seu sistema a controlar diretamente a rotação do motor dc.

O pulso de controlo do servo é à volta dos 20 milissegundos e quando um servo recebe esse comando, irá continuar em rotação ou manter a mesma posição nem que sejam aplicadas forças externas. Quanto maior forem as forças, maior a corrente que o servomotor irá consumir.

Servomotores simples apenas possuem feedback interno de posição enquanto que servomotores mais sofisticados usados em aplicações industriais utilizam feedback de posição e velocidade.

O feedback interno é obtido a partir do encoder. Apesar de existirem diversos tipos de encoders como óticos e magnéticos, os mais comuns em servomotores de modelismo são os potenciómetros. A variação da resistência está diretamente relacionada com o movimento angular produzido pela *shaft* do servomotor, sendo esta informação a obtida pelo malha fechada do sistema de *feedback*.

A caixa de redução presente no interior do servomotor e que interliga o motor dc e o eixo exterior, é, tal como o nome indica, uma redução da velocidade do servomotor em si mesmo relativamente ao motor elétrico no seu interior. No entanto, permite aumentar o binário. Este rácio da velocidade/binário é proporcional à relação entre o número de dentes das engrenagens.

Atualmente, o mercado dos servomotores pratica um preço bastante competitivo e dispõe de uma enorme variedade, facilitando a escolha do que se adequa melhor à construção de um manipulador sem comprometer as expectativas. As características para a escolha de um servomotor são: precisão de posição, binário necessário, gamas de velocidade, tensão de funcionamento e o peso do equipamento.

2.3.1.3 Motor de Passo

Um motor de passo, em vez do potenciómetro do servomotor, utiliza atuadores eletromagnéticos que definem a resolução do passo do motor [21].

Tal como é perceptível, os passos são um conjunto de intervalos angulares discretos, o que permite uma maior precisão para controlo de posição em malha aberta [20]. No entanto, são também mais caros e têm pior performance dinâmica.

O *ratio* da potência em função do peso é mais baixo que os restantes motores.

São utilizados em robôs simples e pequenos, de controlo em malha aberta quer em posição quer em velocidade.

2.3.2 Controlo

Tal como mencionado anteriormente, um servomotor é constituído por três fios, dois de alimentação e um de sinal. Quando não é modificado, é geralmente usado para posicionamento, sendo que o seu ângulo de rotação é baseado num sinal PWM.

Ao contrário do controlo em motores DC que utilizam o sinal de PWM para diminuir a tensão média fornecida ao motor e, conseqüentemente, a velocidade de rotação, o servomotor identifica a posição pretendida através da comparação da largura do seu pulso com a largura do pulso fornecida pelo *encoder*.

A diferença entre os dois sinais origina um erro que irá atuar o motor. Um controlo típico pode ser visualizado na figura 2.6

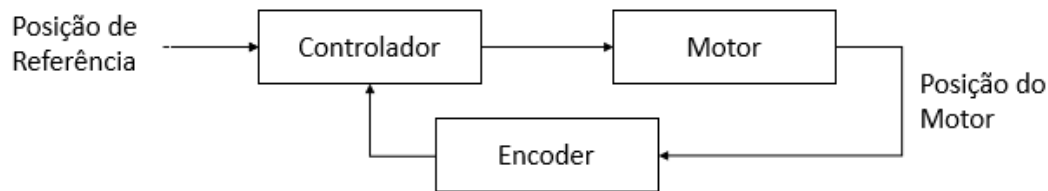


Figura 2.6: Controlo típico usado em servomotores.

O potenciômetro utilizado como sensor de posição que fornece *feedback* relativamente ao ângulo real de rotação, roda em função do eixo da *shaft* do servomotor e a sua diferença de potencial é alterada de acordo com a posição a cada momento do eixo de rotação. O valor mínimo e máximo da diferença de potencial corresponde ao valor mínimo e máximo da rotação do eixo. A tensão é aos terminais do pino do meio e do pino terra é utilizado para comparar com o sinal de referência.

É interessante realçar que o controlo de realimentação é feito internamente ao servomotor, apenas sendo possível ao utilizador a escolha da posição desejada.

2.4 Controlador

O robô é operado através de uma plataforma programável de controlo que faz a interligação entre as tarefas programáveis e a movimentação da estrutura.

É possível utilizar ou uma carta de rede ou um microcontrolador. Um microcontrolador é um sistema de controlo dedicado a uma determinada tarefa com periféricos de entrada e saída, memória e com um processador.

2.5 Protocolos de Comunicação

Nesta secção será feita uma abordagem aos tipos de comunicação a serem utilizados.

2.5.1 I²C

Inter-Integrated Circuit (I²C) é um protocolo síncrono projetado pela Phillips em 1982 para controlar equipamentos eletrônicos com vários *chips* Phillips.

Serial data (SDA) e *serial clock* (SCL), são as duas linhas de sinal, *pulled up* tipicamente por 5V, que transportam a informação entre dispositivos conectados ao mesmo barramento. SDA é a linha de transmissão de dados bidireccional e SCL é responsável pela sincronização entre dispositivos [22].

Cada dispositivo tem o seu endereço específico pré-atribuído e pode operar como transmissor ou receptor. O *master* é quem inicia a transmissão de informação e gera os sinais de *clock* que permitem a transferência de dados. Nessa altura, todos os dispositivos a quem se dirige são os *slaves* [22].

É gratuito desde 2006.

A figura 2.7 apresenta uma típica ligação entre mestre e escravos.

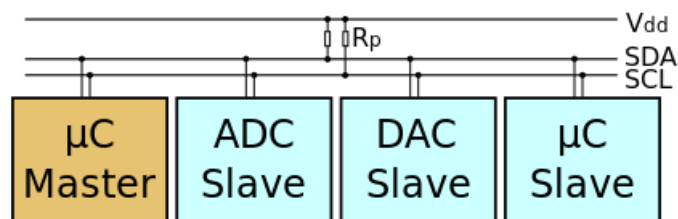


Figura 2.7: Esquema de ligações com um mestre e três escravos - I²C.

2.6 Software relevante

2.6.1 CAD

Um programa de Desenho Assistido por Computador (DAC) ou CAD (do inglês: *computer aided design*), usado em projeção e desenvolvimento de desenho técnico, traz vantagens face ao desenho manual através de geração automática de conteúdo, assistência à forma do objeto, auto preenchimento, entre outras vantagens, tipicamente através da seleção ou modificação de parâmetros no desenho 3D com o rato ou com uma linha de comandos bastante simples como [AutoCAD](#) ou [Inventor](#).

2.6.1.1 OpenSCAD

O [OpenSCAD](#) é um programa gratuito para criar objetos sólidos 3D que não se foca na componente artística mas nos aspetos de CAD.

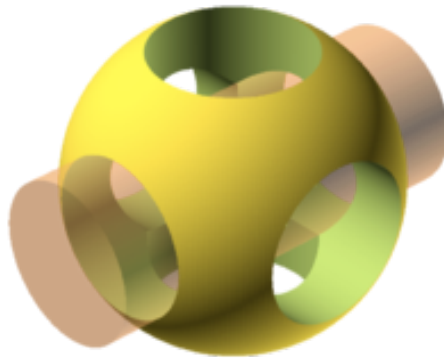


Figura 2.8: OpenSCAD logo, retirado de [OpenSCAD website](#).

Apesar de não interagir diretamente com a figura 3D e não ter um motor de física associado para realizar testes de *stress*, temperatura ou desgaste como outros programas, permite ao utilizador uma maior interação com os passos de construção ou a modificação de algum parâmetro configurável pois é similar a um compilador 3D de um *script*.

O OpenSCAD utiliza uma linguagem de programação funcional usada para criar modelos a partir de dois tipos principais de técnicas de modelação: primitivas geométricas ou através da extrusão de objetos 2D. A estrutura de um modelo é baseado na diferença entre duas uniões de objetos.

Um ficheiro especifica os modelos e os parâmetros a utilizar para renderização tal como é possível observar na figura 2.9.

É *open-source*.

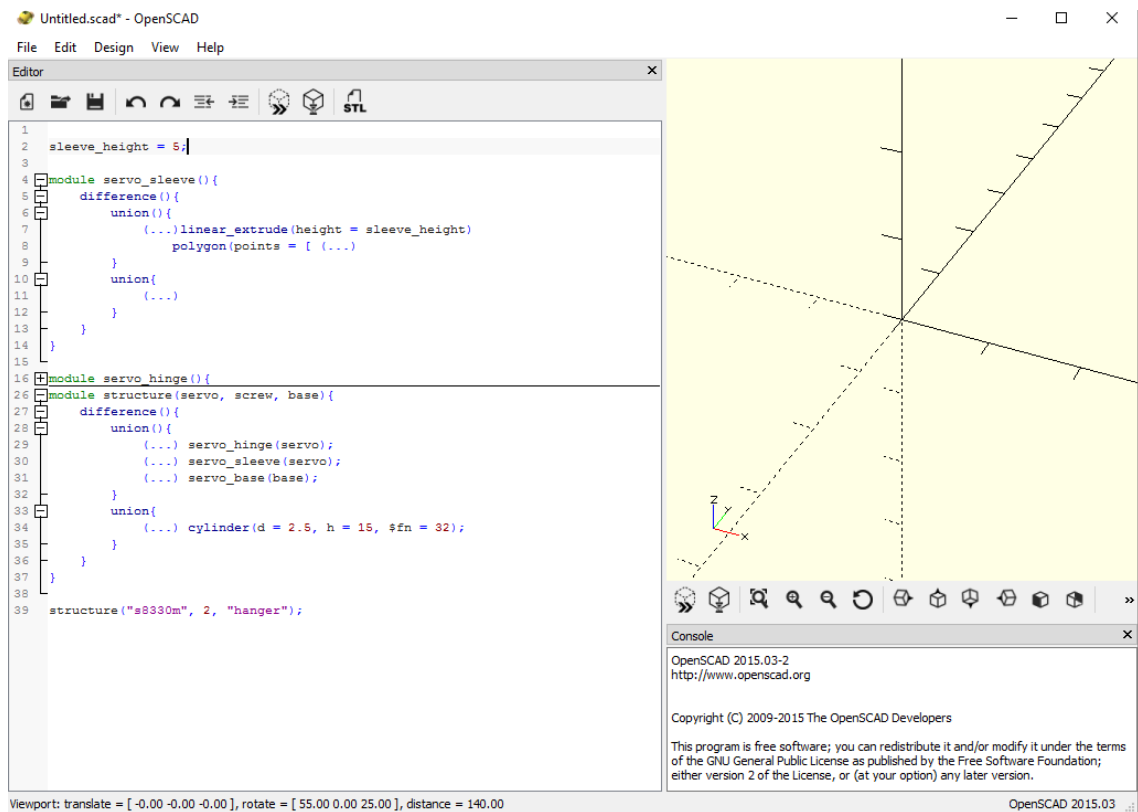


Figura 2.9: Interface e estrutura exemplo de um programa em OpenSCAD.

2.6.2 Simuladores

Obter resultados reais sem *hardware* é hoje em dia possível recorrendo a simuladores com motores de física implementados. Estes são o ponto chave para obter modelos de controlo mais próximos dos reais com possibilidade de afinações.

Existe um largo número de simuladores de qualidade, desde os comerciais como o [v-rep](#), [The Construct](#) e o [Webots](#), passando pelos gratuitos como o [Unreal Engine](#) aos *open-source* como o [SimTwo](#), [MORSE](#), [RoboDK](#) ou o [Gazebo](#).

Para uma simulação em tempo-real, deve ser usado um motor de física rápido e preciso. *Open Dynamics Engine (ODE)* é uma biblioteca gratuita utilizada como uma ferramenta de investigação para simular a dinâmica de corpos rígidos.

A opção passou então pela utilização do SimTwo desenvolvido por Paulo Costa [23].

2.6.2.1 SimTwo

O SimTWO é um sistema de simulação realista que tal como descrito no seu [manual](#) permite ainda criar todas as entidades que povoam o mundo a simular, como robôs, obstáculos, corpos livres e pistas ou marcadores.

Nomeadamente na criação do robô, é possível efetuar uma descrição dos corpos e das suas ligações em XML, apresentando ainda o modelo de atrito, do motor e do controlador.

O modelo de atrito poderá ser resumido a dois coeficientes de atrito: estático (f_c) e viscosa (b_v). A força de atrito estática é uma função sinal com uma amplitude f_c . A força de atrito viscosa depende linearmente (com declive b_v) da velocidade de rotação.

O controlador poderá ser de três tipos:

- *pidposition* (Implementa um PID que tenta controlar a posição e onde os ganhos k_p , k_i e k_d devem ser definidos. Para além destes um ganho de feed-forward k_f pode ser adicionado e o controlador permite definir uma saturação *ysat*);
- *pidspeed* (Implementa um PID que tenta controlar a velocidade e onde os ganhos k_p , k_i e k_d devem ser definidos. Para além destes um ganho de feed-forward k_f pode ser adicionado e o controlador permite definir uma saturação *ysat*);
- *state* (Implementa uma realimentação de estado composto por (velocidade, posição e integral da posição) com ganhos k_p , k_i , k_d e k_f para o termo proporcional à referência).

O modelo do motor é baseado num modelo de um motor CC. Para isso são necessários os parâmetros:

- *ri* (resistência);
- *ki* (constante do motor relaciona linearmente o binário com a corrente e a f.c.e. com a velocidade de rotação);
- *vmax* (tensão em Volt de máxima alimentação);
- *imax* (corrente em Ampère de máxima alimentação).

Para além de permitir o controlo do movimento do robô através das referências dos estados de todos os seus servos, serve também de sistema de aquisição de todas as variáveis das articulações. Esta ferramenta torna-se útil para efetuar a análise da resposta dinâmica das articulações para posteriormente serem comparadas com a simulação. A aplicação possibilita também o armazenamento em ficheiro dos estados pré-definidos de forma a gerar sequências de movimentos e partilhar com o simulador. Existe ainda um modo de desactivação dos binários dos servos. Neste modo, o robô permite que o movimento enquanto a aplicação adquire os valores instantâneos dos ângulos de cada articulação [3].

O simulador é então composto por quatro janelas: 1) Simulação, como é possível ver na figura 2.10; 2) Controlo; 3) Editor de Script; 4) Gráfico.

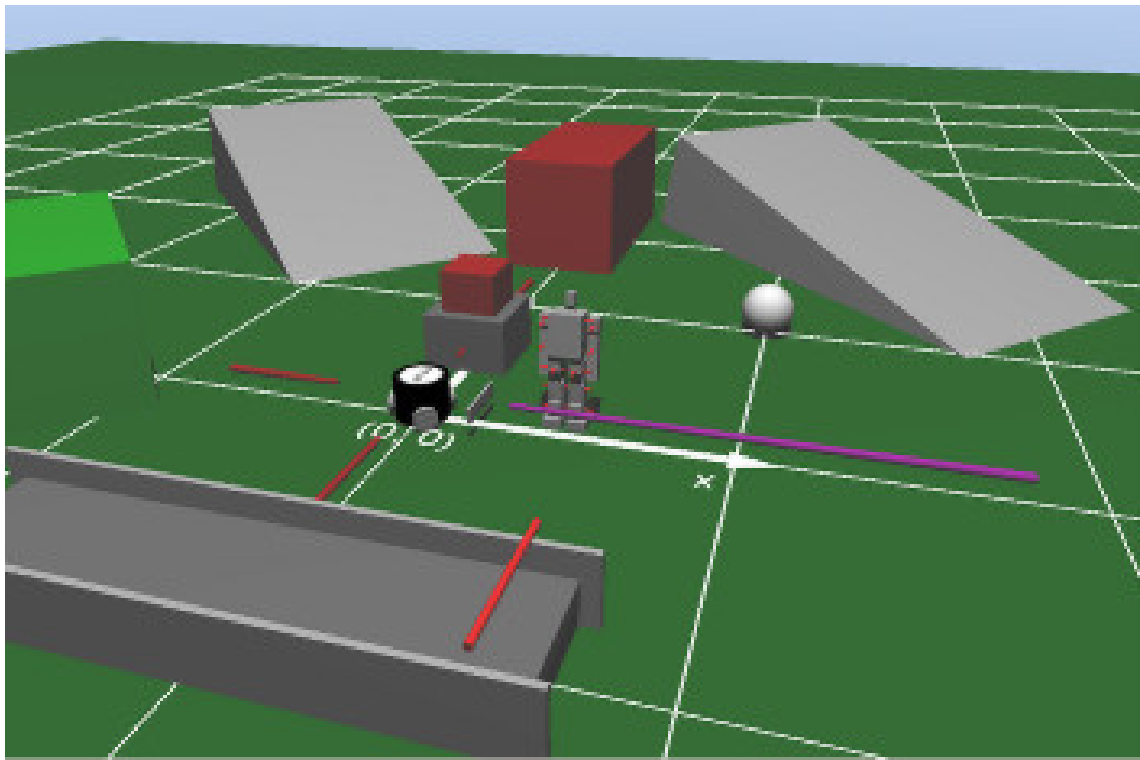


Figura 2.10: Típica simulação no SimTwo, retirada do [website](#).

2.7 Resumo

O protótipo a desenvolver irá seguir uma arquitetura série, com ligações discretas, isto é, ligações de corpos rígidos, com servomotores responsáveis pelos graus de mobilidade com *feedback* de posição e controlado por um Arduino.

A sua construção partiu da necessidade de validar algoritmos de planeamento de trajetória e da cinemática do manipulador que foram desenvolvidos paralelamente a esta dissertação, assim como validar algoritmos de estimação para controlo de posição das articulações de um manipulador com redundância. Também foi um fator relevante, a necessidade de validar a utilização de ligações de baixo custo para manipuladores fiáveis.

Capítulo 3

Metodologia

Este capítulo irá abordar a metodologia seguida ao longo da dissertação no *flowchart* 3.1, de seguida irão ser apresentados os parâmetros mais relevantes a considerar no projeto da dissertação em 3.1, as escolhas feitas ao nível de equipamento em 3.2 e a arquitetura de controlo em 3.3.

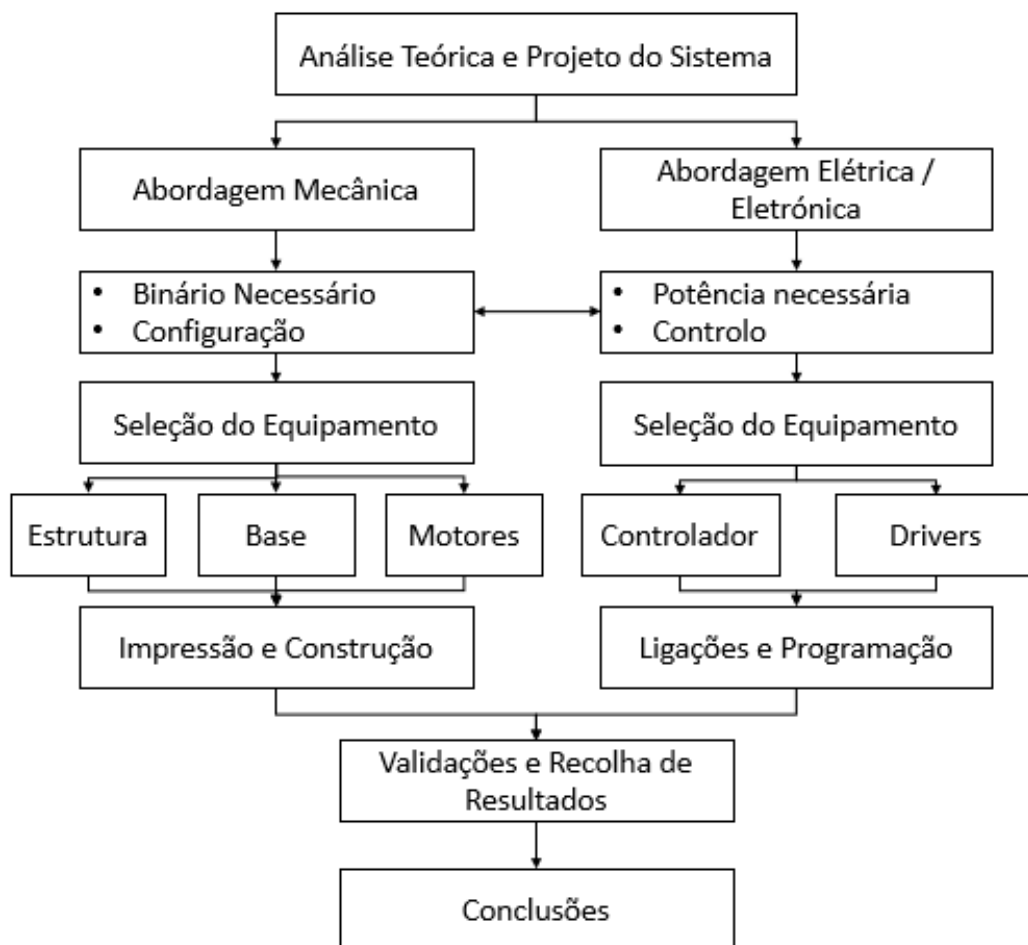


Figura 3.1: Metodologia da dissertação.

3.1 Parâmetros a considerar

O manipulador robótico considerado para construção, correspondia ao requisito de doze graus de movimentação, correspondendo a doze servomotores nas articulações. Estas serão responsáveis por suportar o peso do restante braço e a carga. A carga é assim um parâmetro que irá influenciar diretamente a escolha dos servomotores, pois o binário que estes serão capazes de fornecer terá de ser igual ou superior ao binário mínimo calculado para cada articulação.

Para o cálculo do binário, expresso na equação 3.1, assume-se que o centro de massa das ligações se encontra no centro da ligação. Tendo em conta a configuração pretendida, existem dois tipos de movimentação principais: a rotação segundo o próprio eixo e a rotação que efetua movimentos de translação ao restante manipulador.

$$\tau = \Delta x * F \sin(\theta) \quad (3.1)$$

Foram definidas duas configurações cinemáticas, uma em que todos os servomotores estão localizados de forma a possibilitar apenas um deslocamento no espaço a 2 dimensões, isto é, se o manipulador estiver segundo o eixo Z, as rotações serão feitas segundo o eixo X, figura 3.2. Na segunda configuração, os servomotores encontram-se posicionados ortogonalmente ao anterior e permite uma rotação no eixo quer no eixo dos Z quer no eixo dos X, sendo assim uma movimentação no espaço a 3 dimensões, figura 3.3.

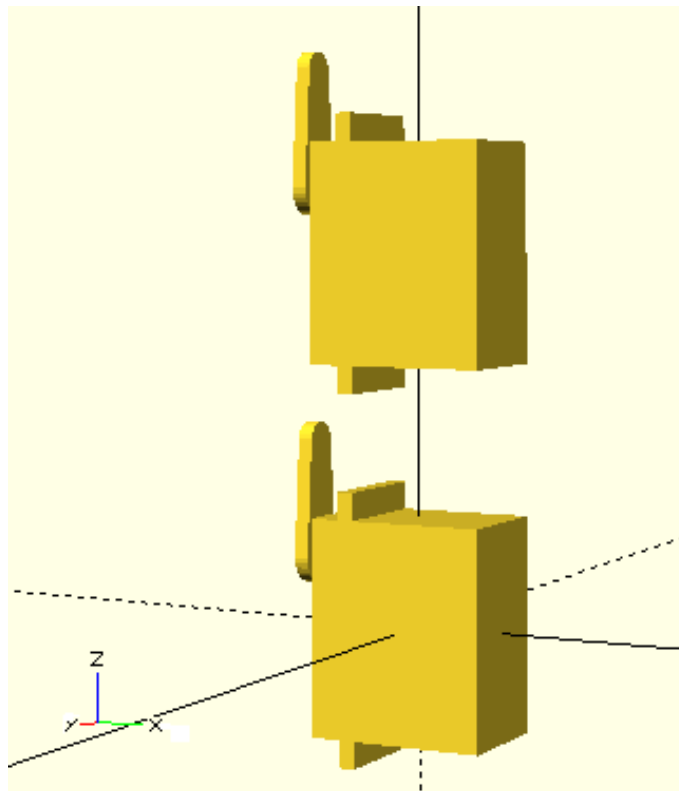


Figura 3.2: Primeira configuração.

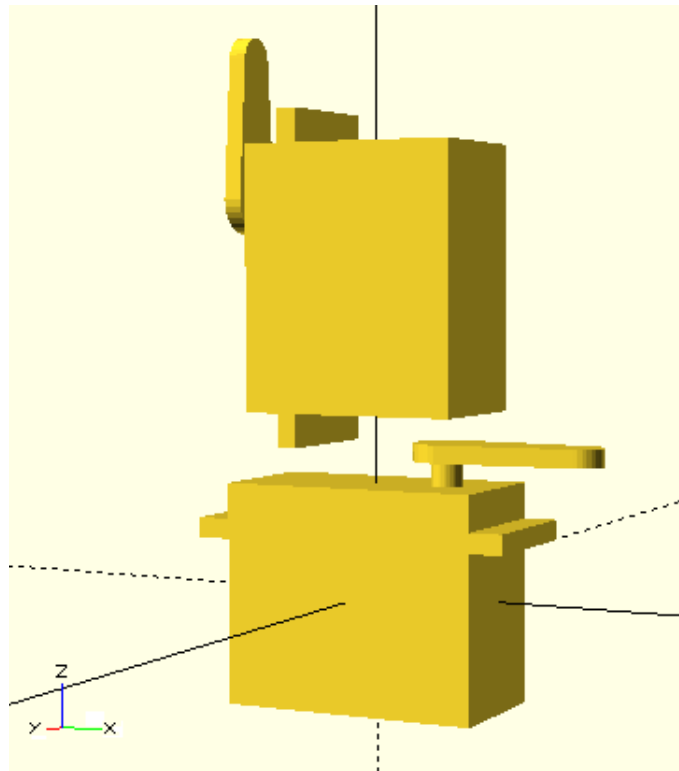


Figura 3.3: Segunda configuração.

3.1.1 Configuração 1

O cálculo do binário para um motor, de forma simplificada para duas articulações e três motores, é apresentado na equação 3.2, onde $M1$ representa a massa do motor 1, $M2$ a massa do motor 2, $m1$ a massa da ligação 1, $m2$ a massa da ligação 2 e d_i representa a ligação entre articulações.

$$\tau_3 = (d2 + \frac{d1}{2}) * m1 + d2 * M2 + \frac{d2}{2} * m2 \quad (3.2)$$

Na tabela 3.1 são apresentadas as dimensões definidas para cálculo do binário.

Tabela 3.1: Dimensões para cálculo do binário.

Parte do Manipulador	Distância (cm)	Massa (g)
Carga + Ferramenta	8	40
Ligação	2,5	15
Servomotor na base x 2	6	140
Servomotor central x 4	5	60
Microservo na ponta x 6	3	12

Os resultados obtidos são apresentados na tabela 3.2.

Tabela 3.2: Binário necessário para cada articulação.

Parte do Manipulador	Binário (kg.cm)
Articulação 11	0.398
Articulação 10	0.638
Articulação 9	1.031
Articulação 8	1.518
Articulação 7	2.098
Articulação 6	2.772
Articulação 5	3.813
Articulação 4	5.347
Articulação 3	7.368
Articulação 2	9.877
Articulação 1	13.344
Articulação 0	17.972

3.1.2 Configuração 2

Sabendo que o binário é igual ao somatório da multiplicação entre a massa da carga e a distância entre o centro de massa e o ponto pivô, para a primeira movimentação, tal como é possível observar na figura que representa a parte de cima da articulação 3.4:

- a distância é igual ao comprimento do *horn* e metade da largura do suporte;
- a massa da carga é igual ao somatório do restante manipulador.

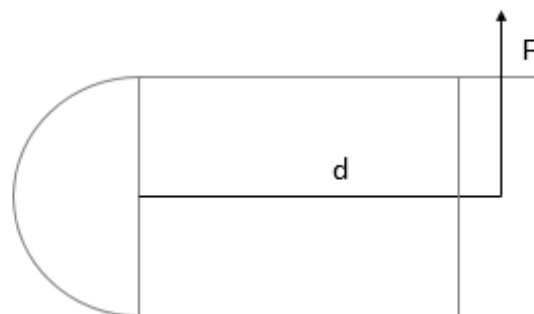


Figura 3.4: Cálculo do binário.

Para a segunda movimentação:

- os pontos pivôs são o centro de cada servomotor e de cada ligação;
- os pontos pivôs possuem a carga que o antecede.

Na tabela 3.3 são apresentadas as dimensões definidas para cálculo do binário.

Tabela 3.3: Dimensões para cálculo do binário.

Parte do Manipulador	Altura (cm)	Largura (cm)	Massa (g)
Carga + Ferramenta	8	-	40
Ligação	2,5	-	15
Servomotor de base x 2	6	4.5	140
Servomotor central x 4	5	3.5	60
Microservo na ponta x 6	3	3	12

Os resultados obtidos são apresentados na tabela 3.4. As articulações pares correspondem ao movimento 1 e as ímpares correspondem ao movimento 2.

Tabela 3.4: Binário necessário para cada articulação.

Parte do Manipulador	Binário (kg.cm)
Articulação 11	0.398
Articulação 10	0.192
Articulação 9	1.094
Articulação 8	0.264
Articulação 7	2.258
Articulação 6	0.336
Articulação 5	4.104
Articulação 4	0.812
Articulação 3	7.659
Articulação 2	1.232
Articulação 1	13.635
Articulação 0	2.844

O cálculo do binário para as articulações pares que efetuam movimentos complexos é diferente do apresentado anteriormente. Os movimentos complexos resultam do facto da articulação imediatamente a seguir se encontrar com uma rotação de 90° . Nessa situação, o binário resultante é igual ao binário da articulação seguinte pois a distância dos centros de massa que o sucedem ao eixo de rotação é a mesma que para o movimento da articulação seguinte, tal como é possível observar na tabela 3.5.

Tabela 3.5: Binário necessário para cada articulação.

Parte do Manipulador	Binário (kg.cm)
Articulação 11	0.398
Articulação 10	0.398
Articulação 9	1.094
Articulação 8	1.094
Articulação 7	2.258
Articulação 6	2.258
Articulação 5	4.104
Articulação 4	4.104
Articulação 3	7.659
Articulação 2	7.659
Articulação 1	13.635
Articulação 0	13.635

3.2 Componentes do Manipulador

3.2.1 Servomotores

A escolha dos servomotores respeitou as seguintes considerações:

- Acessibilidade de custo - O preço por unidade é reduzido comparativamente com outros de elevada performance?
- Disponibilidade no mercado - A compra não está limitada a poucas unidades?
- Binário - Corresponde ao binário necessário para a articulação?

Tendo em conta as restrições de binário da primeira configuração, os motores ficaram reduzidos a três tipos: servomotores *High Torque*, microserves e servomotores *standard*.

Devido à necessidade de uma performance similar nas diversas articulações e de forma a facilitar o próprio *design* das ligações, ficou definido que seria utilizado um modelo de servomotor por cada tipo. De seguida, foi feita uma análise em função da massa de cada servo em função do binário fornecido e do próprio custo, chegando aos exemplares na tabela .

Tabela 3.6: Servomotores escolhidos.

Servomotor	Massa	Binário
S8330M	137 g	30 kg.cm
TGY-S901D	58 g	13 kg.cm
11HMB	11 g	3 kg.cm

Estes servos apresentam diversas vantagens, nomeadamente o baixo custo, a realimentação interna da posição, a facilidade de serem encontrados no mercado e o facto de possuírem dimensões *standard*, isto é, dimensões comuns à maior parte dos *hobby* servos existentes no mercado.

Comparando com os valores da análise ao binário a fornecer feita anteriormente, é possível observar que cumprem os requisitos.

3.2.2 Estrutura do Manipulador

O material constituinte da estrutura do manipulador deveria ser suficientemente leve de forma a poder movimentar uma carga com maior massa, assim como suficientemente robusto para aguentar determinadas tensões em configurações mais complexas.

Visto que existia o acesso a uma impressora 3D e que o material impresso seria em ABS, um plástico robusto, optou-se por esta solução.

3.2.3 Controlador

Tendo em conta as soluções existentes no mercado, o enquadramento do projeto, ser multi-plataforma, o preço e a curva de aprendizagem bastante suave, foi escolhida uma plataforma de processamento *open-source* baseada em *easy-to-use hardware e software*, o Arduino.

3.2.3.1 Arduino Mega

O *Arduino Mega* é uma *board* baseada no *ATmega2560*, na figura 3.5, possui 54 *digital input/output pins*, 15 dos quais podem ser usados como *PWM outputs*, 16 *analog inputs*, 4 *UARTs*, um cristal de 16MHz, uma conexão *USB*, entre outros recursos, com o custo de 35 euros. É uma plataforma com mais *features* que o *Arduino Uno* demonstrado anteriormente.



Figura 3.5: Arduino Mega, retirado de [Arduino website](#).

3.2.4 Driver do Motor

Tipicamente, um motor DC é controlado por uma Ponte-H, permitindo assim ser bidireccional. No entanto, tal como foi referido na secção anterior 2.3 é necessário um controlo por PWM para ser possível o controlo da sua velocidade.

Assim, é possível utilizar componentes como *Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit*, na figura 3.6, que é um *shield* para Arduino que permite controlar até 4 motores DC bidireccionais, 0.5% de resolução. Utiliza 4 MOSFET *TB6612* que permite 1.2A a 25°C e 100% de *duty cycle* por canal. Possui uma *PWM driver* totalmente dedicada. A comunicação é feita por I2C com o Arduino.

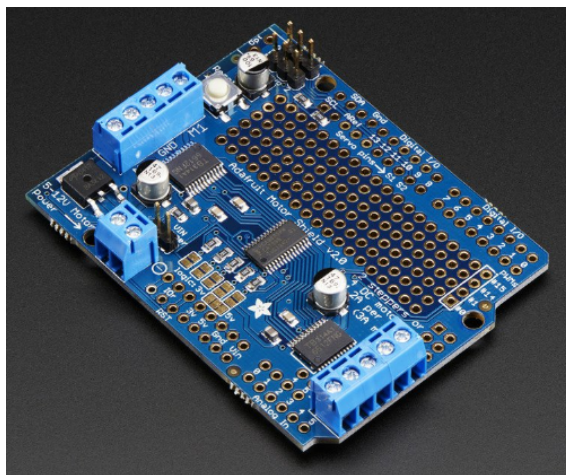


Figura 3.6: MotorShield para Arduino, retirado de [Adrafruit website](#).

O controlo de um conjunto de servomotores também pode ser efetuado a partir de um *shield* como *Adafruit 16-Channel 12-bit PWM/Servo Shield - I2C interface*, na figura 3.7 que permite controlar até 16 servomotores com 4 μ s de resolução. A comunicação é feita por I2C.

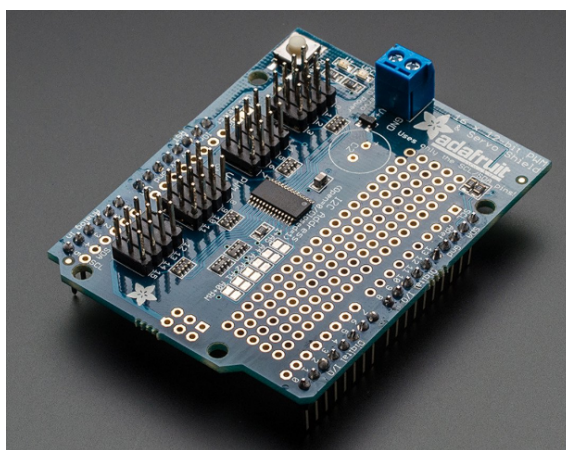


Figura 3.7: 16 PWM *shield* para Arduino, retirado de [Adrafruit website](#).

3.2.5 Fonte de Alimentação

Os servomotores devem ser alimentados por uma fonte de alimentação DC. Esta foi dimensionada para distribuir 5V DC e 20A aos servomotores, tendo em consideração a corrente de *stall*,

tabela 3.7. Os valores foram obtidos experimentalmente.

Tabela 3.7: Corrente de *Stall*.

Servomotor	Corrente	Total
S8330M	2,7A	2 * 2,7 = 5,4A
TGY-S901D	1,7A	4 * 1,7 = 6,8A
1HMB	1,3A	6 * 1,3 = 7,8A
Total	-	20A

A alimentação do controlador será realizada à parte.

3.3 Arquitetura de Controlo

Tendo em conta a figura 3.8 e tal como referido na secção 1.1, é possível distinguir dois níveis.

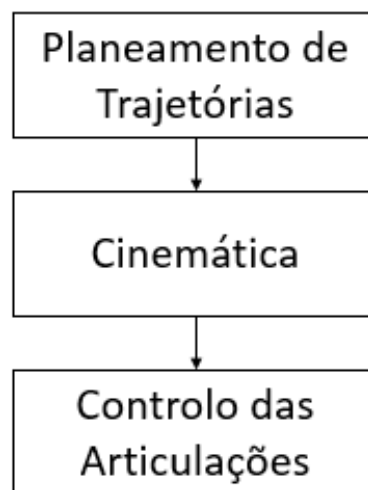


Figura 3.8: Níveis de controlo.

Os níveis referentes à computação do Planeamento de Trajetória e da Cinemática relevante exigem uma elevada capacidade de processamento. Normalmente é usada uma plataforma baseado num sistema operativo.

Enquanto que para o Controlo das Articulações, é utilizada uma plataforma que permita controlar os periféricos e comunicar com a do nível superior.

O controlo foi feito pela plataforma de desenvolvimento Arduino, neste caso o [Arduino Mega](#). Tal como descrito na secção anterior 3.2.3.1, este possui um microcontrolador programável e *software user-friendly* e *open source* para o programar. É também um equipamento barato.

Um sistema de controlo genérico é constituído pela entrada, neste caso a referência da posição, por um compensador da função de transferência, neste caso o controlador, e pela planta do sistema,

isto é, o sistema físico real que terá como saída a posição atual do motor, tal como está ilustrado na figura 2.6.

Capítulo 4

Sistema de Atuação

4.1 Introdução

O braço robô hiper-redundante proposto é então composto por doze servomotores, um sistema de controlo e diversas peças de plástico. A ligação entre as peças é obtida a partir de alguns parafusos que apertam os plásticos em locais pré-definidos, assegurando a ligação dos servos com os restantes componentes.

O sistema de atuação do manipulador será constituído pelos servomotores divididos em três categorias: 2 como base - [S8330M](#), 4 como intermédios - [TGY-S901D](#) - e 6 finais - [11HMB](#).

Este capítulo irá abordar o modelo [4.2](#) do servo e os testes e parametrizações feitas aos servomotores [4.3](#) para validar o modelo.

Será depois apresentado o modelo simulado em [4.4](#), o controlo implementado [4.6](#), e serão retiradas conclusões [4.7](#)

4.2 Modelo

O modelo é essencial para se poder simular o sistema dinâmico do motor.

A figura [4.1](#) ilustra o esquema elétrico do modelo (baseado no motor DC) de uma articulação com uma carga genérica [\[3\]](#).

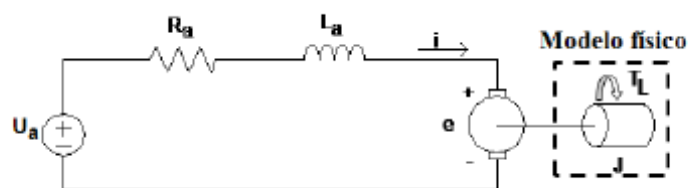


Figura 4.1: Esquema elétrico do modelo do motor DC.

A equação diferencial para a corrente na armadura é:

$$R_a i_a + L_a \frac{\partial i_a}{\partial t} = u_a - e \quad (4.1)$$

Como o fluxo ϕ é constante, o binário desenvolvido pelo motor é dado por:

$$\tau_m = K_1 * \phi * i_a = K_m * i_a \quad (4.2)$$

Onde K_m é a constante do binário. Segundo a lei de Faraday, a força electromotriz é dada por:

$$e = K_2 * \phi * \omega_m = K_b * \omega_m = K_b * \frac{\partial \theta_m}{\partial t} \quad (4.3)$$

Onde K_b é a constante da força electromotriz.

Num motor DC, a constante do binário é igual à constante da força electromotriz, isto é $K_m = K_b = K$.

Quanto às características mecânicas do modelo, a equação de equilíbrio está apresentada na equação 4.4 onde o binário desenvolvido é igual à soma dos binários necessários para vencer o momento de inércia, a força de atrito resistente e o binário resistente da carga:

$$\tau_m = \tau_J + \tau_a + \tau_L \quad (4.4)$$

O binário necessário para vencer o momento de inércia é dado por:

$$\tau_J(t) = J \frac{\partial \omega_m(t)}{\partial t} \quad (4.5)$$

O modelo final é obtido em 4.6. Para tal, é desprezada a indutância L_a .

$$K * i(t) = J \frac{\partial \omega_m(t)}{\partial t} + \tau_a(t) + \tau_L(t) \quad (4.6)$$

4.2.1 Limitações

Devido aos componentes utilizados é necessário acrescentar restrições de tensão e corrente.

A partir da equação 4.2 é possível observar que uma limitação da corrente irá originar uma limitação do binário disponível ao motor.

A partir da equação 4.1 e da equação 4.3 é possível chegar à equação seguinte, 4.7, e concluir que uma restrição na tensão fornecida ao motor irá alterar a sua velocidade de rotação.

$$u_a = R_a i_a + L_a \frac{\partial i_a}{\partial t} + K \omega_m \quad (4.7)$$

4.3 Testes e parametrizações

Os valores R_a , L_a e K podem ser obtidos através de algumas medições diretas ou indiretas no motor [24]. Quanto à constante K , pode-se obter o valor através de alguns ensaios experimentais

tal como demonstrado em [3]. Uma vez que a força electromotriz induzida é proporcional à rotação com ganho K , podem ser medidos vários pontos de funcionamento para diferentes velocidades.

Foram usados os valores dos testes realizados na tese de doutoramento de José Lima [3] como referência.

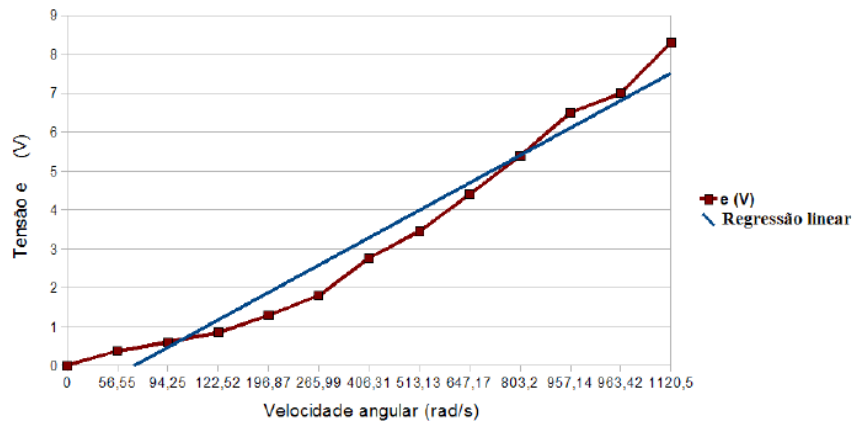


Figura 4.2: Cálculo da constante K do servo, retirado de [3].

O declive da reta da figura 4.2 indica assim o valor de K que é igual a $0.006705V.s/rad$. Nesta figura, a velocidade angular é a do motor dc.

Na tabela 4.1 apresentam-se os parâmetros relevantes do servomotor MG995. O parâmetro do binário é o valor retirado do *datasheet* do ensaio com o rotor bloqueado e o parâmetro da velocidade angular é o valor sem carga e referente à velocidade da *shaft*.

Tabela 4.1: Parâmetros do servomotor MG995.

Grandeza	Valor
τ_s	9.400 kg.cm
ω_0	5.236 rad/s
R_a	25 Ω
K_i	0.006 N/A
K	1.3 V.s/rad

O parâmetro K_i foi retirado após o ensaio com o rotor bloqueado e é um parâmetro do motor DC, enquanto que o parâmetro K é obtido em função da velocidade angular do eixo do servomotor. Apesar de não ser algo descrito no *datasheet*, é possível especular o rácio da caixa redutora como 220:1.

4.3.1 Forças de atrito

O binário de atrito resistente é igual à soma do atrito viscoso com o atrito estático, tal como descrito na equação 4.8. A força de atrito viscoso é dada pela equação 4.9, onde B é um coeficiente

de viscosidade e pode ser representado pela função linear 4.3. O atrito estático é descrito pela equação 4.10 e pode ser representado pela função ilustrada em 4.4.

$$\tau_a = \tau_{av} + \tau_{ae} \quad (4.8)$$

$$\tau_{av}(t) = B\omega(t) \quad (4.9)$$

$$\tau_{ae}(t) = K_e \text{sign}(\omega(t)) \quad (4.10)$$

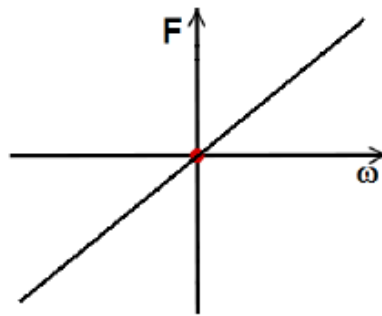


Figura 4.3: Modelo da força de atrito viscoso, retirado de [3].

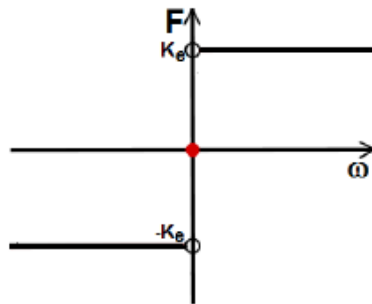


Figura 4.4: Modelo da força de atrito estático, retirado de [3].

A força de atrito resultante é ilustrada pela figura 4.5.

O binário útil τ_s exercido pelo servomotor na carga é igual à diferença entre o binário total e a soma dos atritos, equação 4.11.

$$\tau_s = \tau_d - (\tau_{av} + \tau_{ae}) \quad (4.11)$$

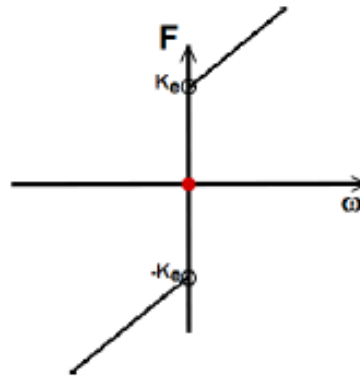


Figura 4.5: Modelo da força de atrito final, retirado de [3].

4.4 Modelo de Simulação

Tal como descrito na secção 2.6.2.1, o simulador escolhido foi o SimTwo. A sua arquitectura básica define dois níveis de controlo: planeamento de trajetória e cinemática como alto nível e controlo dos servomotores como baixo nível, como ilustrado na figura 4.6.

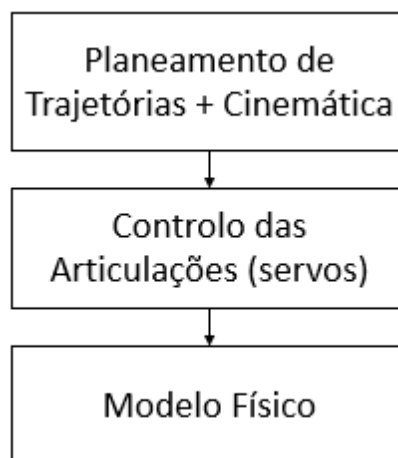


Figura 4.6: Arquitetura básica do simulador.

O bloco modelo físico efetua os cálculos da dinâmica (momento de inércia e binário de carga) do sistema e fornece informação de diversas variáveis das articulações. Esta informação permitirá controlo por realimentação.

4.4.1 Servomotor

Uma vez que a descrição do servomotor é feita a partir de código XML, é possível exemplificar com um excerto do código apresentado de seguida.

```

<articulations >
  <!-- small motor -->
  <default >
    <draw radius='0.02' height='0.0427' rgb24='FFFFFF' />
    <motor ri='1.9' li='0.001' ki='1.3'
      vmax='5' imax='2' active='1' />
    <rotor J='1e-4' bv='1e-3' fc='0' />
    <gear ratio='500' bv='1e-5' ke='50' />
    <friction bv='0.2' fc='0.1' />
    <encoder ppr='1000' mean='0' stdev='0' />
    <controller mode='pidposition' kp='75' ki='0.05'
      kd='5' kf='0.0' active='1' period='10' />
    <spring k='0' zeropos='0' />
  </default >

```

4.4.2 Manipulador

O modelo do braço robótico hiper-redundante implementado encontra-se na figura 4.7, na posição *home*, e na figura 4.8 está representado uma posição onde existiu uma série de atuações nas articulações.

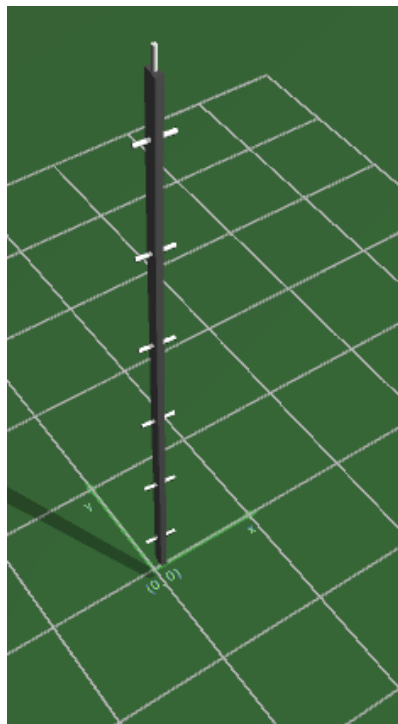


Figura 4.7: Modelo do manipulador no SimTwo. Posição *home*.

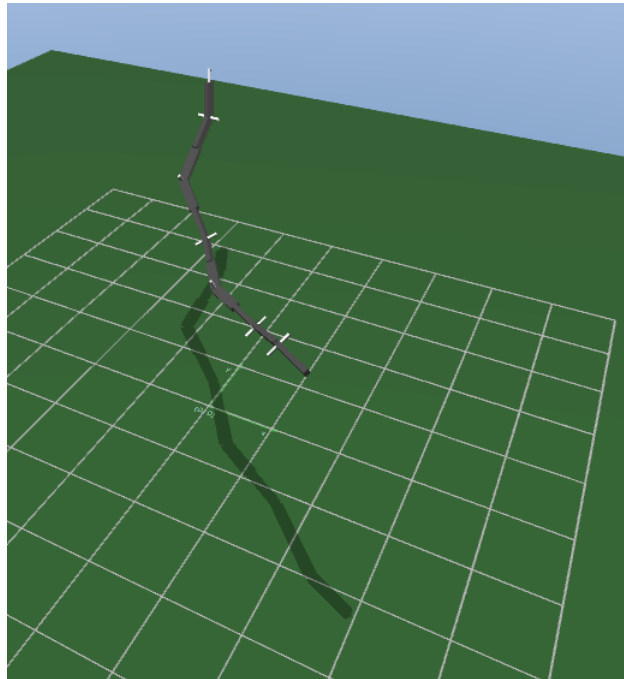


Figura 4.8: Modelo do manipulador no SimTwo após rotações.

4.5 Controlador

O controlo dos servomotores efetuado pelo Arduino pode ser feito de diferentes formas, sendo que todas partem da geração de uma onda quadrada com *duty cycle*.

O objetivo da simulação passa pela análise, *design* e controlo de um sistema antes da implementação prática, permitindo otimizações e segurança.

4.5.1 Soluções comuns

4.5.1.1 AnalogWrite

A forma mais simples de controlar um servomotor é através da função *analogWrite* (*pin, value*) para enviar um sinal de *PWM* a partir do valor definido, entre 0 (sempre *off*) e 255 (sempre *on*).

Esta função poderá ter tempos de ciclo superiores aos definidos devido às interações entre as funções de *delay()* e *millis()* que utilizam o mesmo temporizador interno que geram o *PWM*.

4.5.1.2 Servo Library

Esta biblioteca permite o controlo de servomotores, suporta até 12 motores no caso do Arduino Uno e 48 no Arduino Mega.

É criada uma classe *servo* que possibilita:

- *attach* e *detach* de um motor ao pino de controlo;

- *write*: colocar o ângulo pretendido em graus (entre 0 e 180 graus);
- *writeMicroseconds*: escrever os microssegundos pretendidos para o *duty cycle*;
- *read*: ler o último ângulo obtido entre 0 e 180 graus;
- *readMicroseconds*: ler o último pulso em microssegundos.

Apesar da vantagem inerente a esta biblioteca, apresenta *jitter*.

4.5.1.3 Arduino *shield* - Controlo Servomotor

Contrariamente às funções anteriores, esta PCI que acopla diretamente ao Arduino, possui um gerador de PWM totalmente dedicado.

Para o controlo de 12 servomotores, é apenas necessário um *Adafruit 16-channel PWM/Servo Shield*.

Recorrendo à biblioteca *Adafruit_PWMServoDriver*, é preciso atribuir um controlador para cada servomotor com o seu endereço específico, definir a frequência de PWM, isto é, quantos impulsos por segundo são gerados, e atribuir o valor a que o impulso fica *high* e a que valor fica *low*.

4.5.1.4 Arduino *shield* - Controlo Motor DC

Para o controlo de 12 motores dc, os *Motor/Stepper/Servo Shields for Arduino v2 Kit - v2.3* serão acoplados uns aos outros, o que exige um endereço I2C único para cada *shield*. O endereço por definição é 0x60 (96 em decimal) e é possível ajustar até 0x80 (128 em decimal), totalizando 32 endereços únicos.

Para programar o *offset*, foram seguidas as indicações do [website](#), sendo adicionado um pouco de solda a unir os *jumpers* de endereço respetivos para o bit lógico pretendido.

Recorrendo à biblioteca *Adafruit_MotorShield* é possível controlar múltiplos *shields*. Inicialmente é preciso atribuir um controlador para cada um com o seu endereço específico.

Este depois é configurado quanto aos seus constituintes (motores DC e motores passo-a-passo) e é iniciado.

Pode ser visto o código exemplo a seguir:

```
Adafruit_MotorShield bot(0x61); // Rightmost jumper closed
Adafruit_MotorShield top(0x60); // Default address, no jumpers

// On the top shield, connect two steppers, each with 200 steps
Adafruit_StepperMotor *myStepper1 = top.getStepper(200, 1);
Adafruit_StepperMotor *myStepper2 = top.getStepper(200, 2);

// On the bottom shield connect four DC Motor
Adafruit_DCMotor *myMotor4 = bot.getMotor(4);
```

```

Adafruit_DCMotor *myMotor3 = bot.getMotor(3);
Adafruit_DCMotor *myMotor2 = bot.getMotor(2);
Adafruit_DCMotor *myMotor1 = bot.getMotor(1);

void setup() {
  Serial.begin(9600);
  bot.begin(); // Start the bottom shield
  top.begin(); // Start the top shield

  // turn on the DC motors
  myMotor1->setSpeed(200);
  myMotor1->run(RELEASE);
  (...)
}

```

4.6 Controlo Implementado

4.6.1 Controlo em Posição

O sistema foi desenvolvido de forma a que a entrada possa ser em PWM ou em graus, assim é possível guiar o servomotor até um determinado ângulo ou esperar que ele atinja um determinado ângulo em função de alguns pré-definidos. É também possível enviar o sinal de entrada a todos os servomotores ou a apenas um em específico e de forma manual, isto é, recorrendo a um potenciómetro ou enviando uma string pela porta série.

Tal como referido anteriormente, os servomotores possuem um sistema de controlo interno sem feedback externo de posição e com uma entrada de controlo de posição PWM. Assim, é necessário identificar o valor máximo e mínimo do PWM de controlo e os correspondentes graus como calibração manual.

A calibração foi realizada com um auxílio de um potenciómetro que permitia variar o momento a que o impulso de PWM ficava *low*. Como pontos máximos e mínimos foram definidos os ângulos 90 e -90°. Foi ainda também registado o ponto zero.

Foram desenvolvidas três movimentações padrão:

1. Após receber a *string* com a entrada de referência, as articulações deslocam-se uma a uma, da base até à ferramenta;
2. Após receber a *string* com a entrada de referência, as articulações deslocam-se uma a uma, mas da ferramenta até à base;
3. Após receber a *string* com a entrada de referência, as articulações deslocam-se em simultâneo.

O programa desenvolvido obriga ao cumprimento de umas etapas sequenciais iniciais, na figura 4.9, deslocando os servos para uma posição de referência obtendo uma estimação da posição

atual. Esta estimativa permite, no caso desta dissertação, enviar a configuração atual ao *High-Level* e permite limitar a velocidade de deslocamento, atribuindo deslocamentos de forma iterativa e com restrição temporal. Caso não se fizessem estas etapas, não existiria um conhecimento sobre a posição atual, o que iria fazer com o manipulador se deslocasse à velocidade máxima que o servo de cada articulação permitisse.

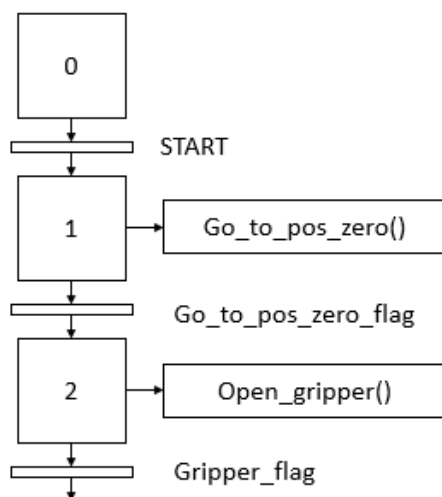


Figura 4.9: Etapas de inicialização.

A partir do momento que a inicialização está concluída, o programa espera que lhe seja enviado por porta série uma das instruções previamente definidas, sendo a mais relevante a *string* de ângulos para movimentação das articulações. A partir desse instante, o programa segue o esquema 4.10.

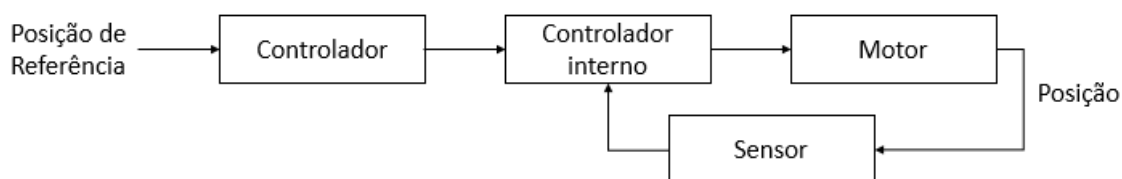


Figura 4.10: Esquema de controle utilizado para movimentar as articulações.

A alimentação do servomotor é fornecida através de um *shield* para o Arduino. Estes *shields* podem ser colocados em cima da placa do Arduino, estendendo assim as suas capacidades. O escolhido para o efeito foi o [Adafruit 16-Channel 12-bit PWM/Servo Shield - I2C interface](#). Devido ao seu *design* é possível o empilhamento de sucessivos *shields* permitindo 992 saídas em PWM.

4.6.2 Controlo em Posição e em Velocidade

De forma a poder ter um certo controlo sobre a velocidade a que a articulação se desloca, seria necessário modificar o servomotor. A limitação relativa à posição atual do motor é possível obter através do potenciómetro presente no sistema de engrenagens do motor.

A alimentação do servomotor é fornecida através de um *shield* para o Arduino. Estes *shields* podem ser colocados em cima da placa do Arduino, extendendo assim as suas capacidades. O escolhido para o efeito foi o *Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit*. Devido ao seu *design* é possível o empilhamento de sucessivos *shields* permitindo controlar até 128 motores.

Tal como mencionado anteriormente, a velocidade é diretamente influenciada pela tensão 4.7. A limitação da tensão a partir do *shield* é obtida através do envio de uma onda quadrada com um certo *on to off ratio* como alimentação do motor em vez de este estar sempre no estado *on*. Assim, a tensão média será igual a $V_{mean} = V * D$, onde D é o *duty cycle*, isto é, a divisão entre o tempo que o sinal está ligado e o período do sinal.

A corrente máxima permitida é, tal como mencionada em 3.2.4, 1.2A a 25°C e com um *duty cycle* de 100%. No entanto, segundo o datasheet, a 20% de *duty cycle* poderá passar 2A. Sabendo o K a partir dos testes efetuados aos motores em 4.3 é possível saber a limitação de binário de cada motor. Esta limitação irá ser usada no cálculo do melhor trajeto do manipulador para uma determinada posição.

O sistema fica então como em 4.11 para um controlo da velocidade das articulações enquanto se deslocam para a posição desejada.

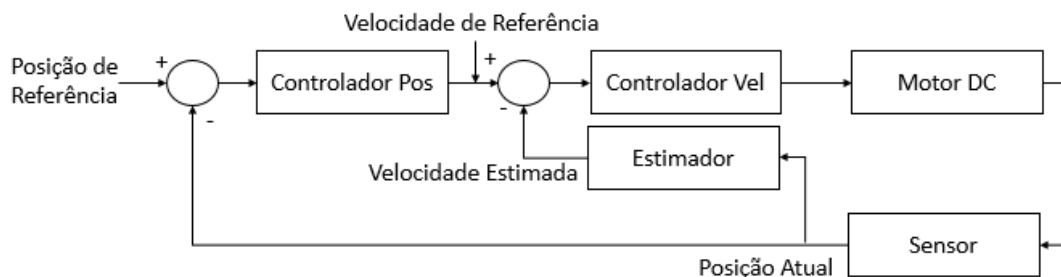


Figura 4.11: Sistema a controlar.

O estimador da velocidade não foi projetado, sendo então a velocidade a que o servo se desloca controlada por uma série de regras presentes na figura 4.12. Na listagem que se segue, são apresentados alguns dos parâmetros:

- erro de posição = posição de referência - posição atual;
- erro de movimentação -> valor afinado manualmente;
- max velocidade -> valor afinado manualmente (tipicamente 90% do valor máximo possível);

- erro de desaceleração- > valor afinado manualmente;
- vel -> valor afinado manualmente (70% do valor máximo possível);
- erro de ajuste -> valor afinado manualmente;
- min velocidade -> valor afinado manualmente (60% do valor máximo possível);
- erro de fim -> valor afinado manualmente.

O programa espera então por uma das instruções, sendo a mais relevante a *string* com a posição desejada para as articulações. O programa segue então o esquema 4.12.

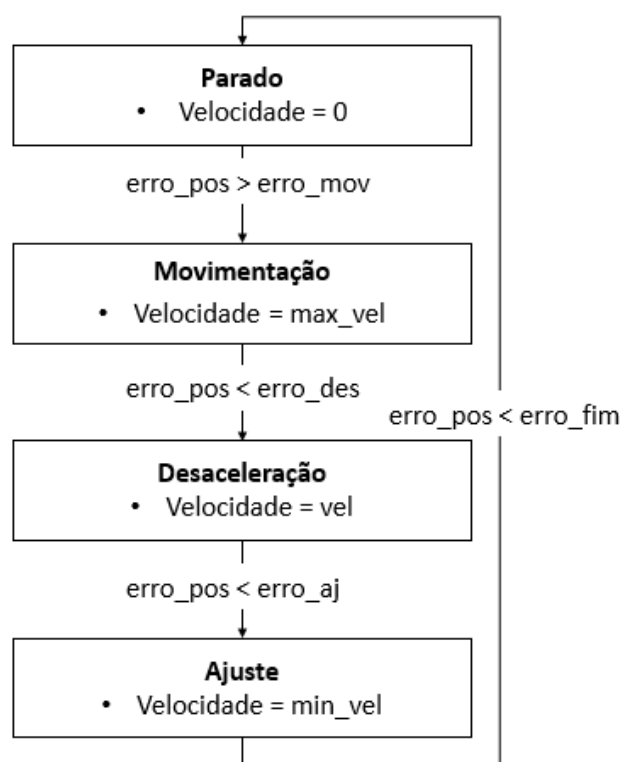


Figura 4.12: Esquema de controlo utilizado para movimentar as articulações.

4.7 Resumo

O controlador utilizado foi o Arduino Mega e foram projetados dois sistemas de controlo. Um utilizando os servomotores originais, através de um shield com saída em PWM para controlo em posição e outro com a alimentação ao motor fornecida pelo *Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit* tendo sido removida a eletrónica de controlo do servomotor.

Foi obtido um modelo do servomotor para utilizar na sua simulação em SimTwo.

Foi desenvolvido um sistema de controlo para o motor em função das limitações e objetivos. O modelo final pode ser visualizado em [4.11](#).

Capítulo 5

Construção da Estrutura Mecânica

A construção da estrutura mecânica do manipulador de 12-DOF será descrita ao longo deste capítulo. Esta estará presa a uma base fixa, também construída de raiz.

Inicialmente será abordada uma introdução à impressão 3D em 5.2.

De seguida serão apresentados os passos para a prototipagem do manipulador em 5.3 e as principais estruturas constituintes.

O capítulo termina com as peças desenvolvidas em 5.4.

5.1 Introdução

A estrutura mecânica do manipulador hiper-redundante deverá suportar 12 servomotores que irão estar dispostos ortogonalmente uns aos outros.

As peças terão de ter em conta a diferente dimensão entre servomotores, assim como o acesso à cablagem de alimentação e controlo.

Estas foram impressas por uma Impressora 3D.

5.2 Impressão 3D

As peças a dimensionar irão ser impressas por uma Impressora 3D. Trata-se de um processo no qual se geram objetos sólidos tridimensionais a partir de modelos digitais, recorrendo frequentemente à fusão a laser, uma tecnologia de fabrico que utiliza energia laser concentrada para fundir pós metálicos em objetos 3D, com presença nos setores de engenharia de alta tecnologia, eletrónica, indústria médica (ortopedia) e aeroespacial, por exemplo.

As principais vantagens passam pelo custo reduzido do material e a facilidade de correção das peças.

Caso as peças não fossem de dimensões reduzidas e de um *design* simples, teria de ser repensado o método de fabricação pois as impressoras 3D no mercado ainda apresentam uma série de

limitações nessas situações como a dimensão da superfície de impressão e o tempo para impressão. As peças são também menos robustas comparativamente com as peças criadas pelo método de injeção de molde.

Nos últimos anos, o preço do equipamento tornou-se financeiramente acessível, as mais baratas rondam os 200 euros e as mais caras chegam aos 20000 euros.

Para que ocorra a impressão de algum objeto em três dimensões é necessário antes de mais realizar a modelação. Os modelos para impressões 3D podem ser criados através de um software de modelação em 3D ou a partir de uma digitalização em três dimensões. Um software de modelação tridimensional é utilizado para desenhar o que será impresso. Como alternativa, pode-se fazer *download* de um modelo já pronto. O computador envia as instruções para a impressora, que aquece a matéria-prima e começa a desenvolver o modelo numa pilha muito fina de camadas.

Os modelos para impressões 3D podem ser criados através de um pacote CAD (do inglês: *Computer Aided Design*) ou através de scanner 3D. A digitalização 3D é um processo de análise e recolha de dados de objetos reais, nomeadamente da sua forma e aparência, e construção de modelos tridimensionais digitais.

Para executar uma impressão, o dispositivo lê o projeto a partir de um arquivo *.STL e processa camadas sucessivas de líquido, pó, papel ou folha de material para construir o modelo através de uma série de secções transversais. Estas camadas, que correspondem às secções transversais virtuais a partir do modelo de CAD, são unidas automaticamente ou fundidas para criar a forma final. A principal vantagem desta técnica é a sua capacidade para criar praticamente qualquer forma ou característica geométrica.

Em geral, o material é derretido (no caso do plástico, por exemplo) e aplicado por bicos de extrusão até formar o objeto tridimensional.

5.3 Prototipagem

Visto que as peças deveriam ser o mais robustas possíveis mas mantendo um preço reduzido, foram projetados dois tipos principais de peças: *hinge* 5.1, uma dobradiça que permite o encaixe no *horn* do servomotor, permitindo o movimento, e *sleeve* 5.1, uma peça de suporte na base do servomotor que o interliga com a dobradiça anterior.

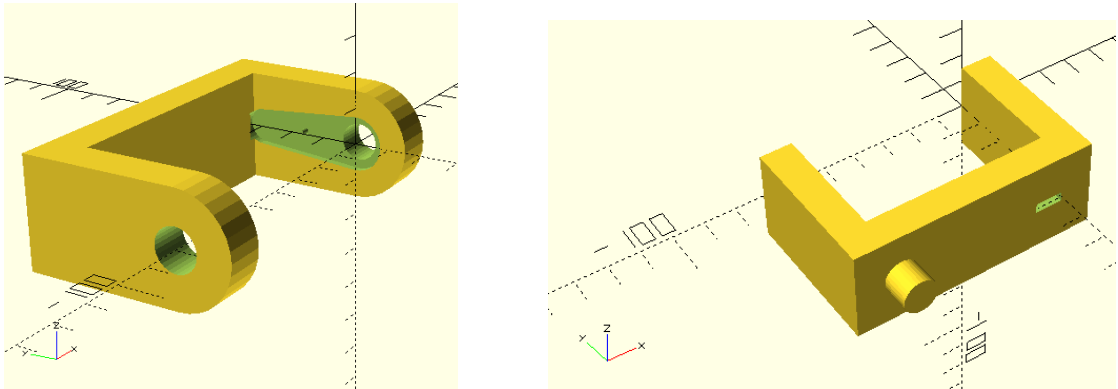


Figura 5.1: *Hinge* e *Sleeve*, peças básicas para interligar os diversos servomotores.

Para a interligação da estrutura, foi desenvolvida uma terceira peça a partir das duas anteriores, 5.2, e ainda uma ligação retangular, na figura 5.3 e a azul na figura 5.4.

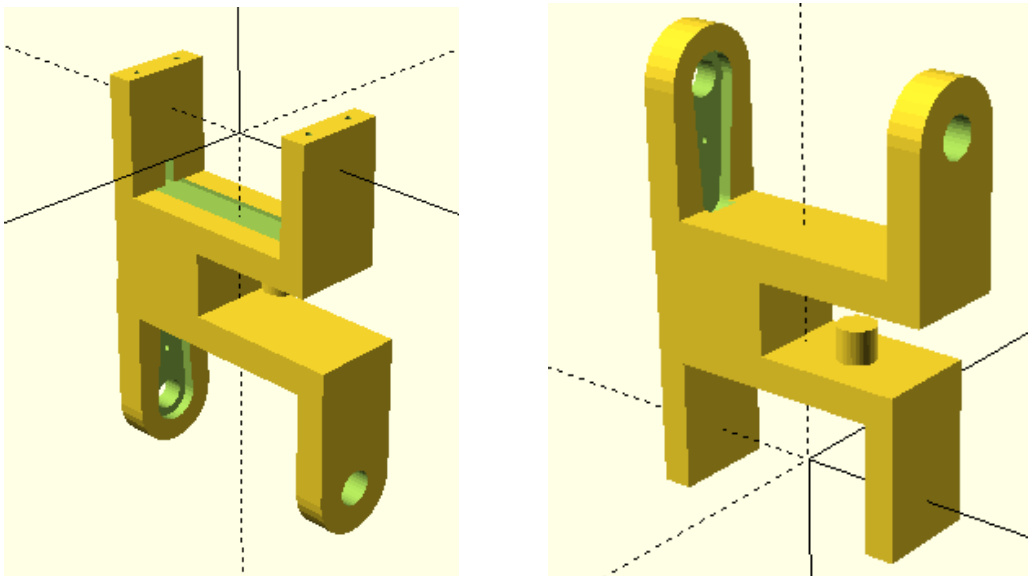


Figura 5.2: *Hinge* e *Sleeve* unidas por um retângulo.

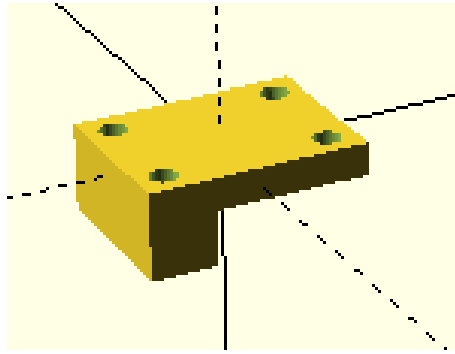


Figura 5.3: Ligação.

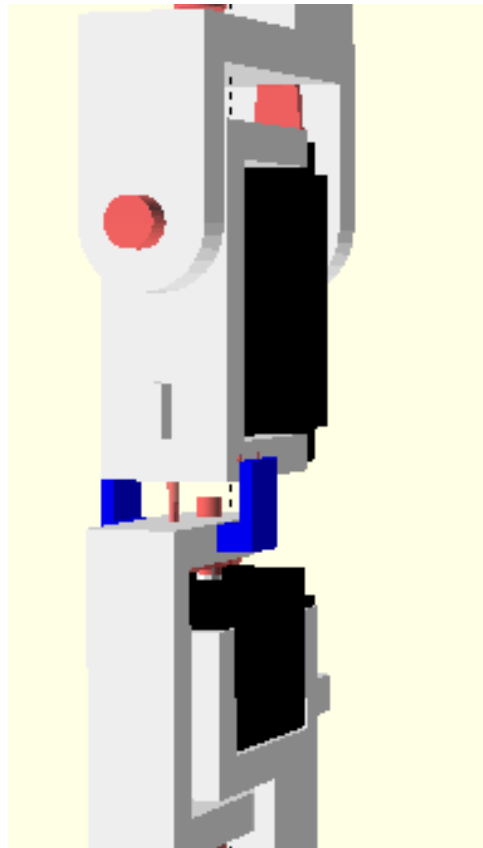
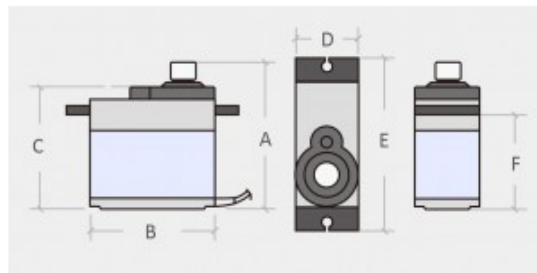


Figura 5.4: Ligação entre duas peças da estrutura.

Tal como está descrito em 2.6.1.1, o *software* utilizado permite a definição de variáveis que irão ser as responsáveis pelos formatos e tamanhos das peças. As variáveis, por sua vez, estão intimamente relacionadas com as medidas de um servomotor, tipicamente as representadas na figura 5.5 pois o seu binário e tamanho podem afetar as dimensões da estrutura.

Product Description



PRODUCT CONFIGURE TABLE

Weight(g)	9
Torque(kg)(4.8v)	1.8
Speed(sec/60deg)	0.1
A(mm)	34.5
B(mm)	22.8
C(mm)	26.7
D(mm)	12.6
E(mm)	32.5
F(mm)	16

Figura 5.5: Dimensões do servomotor exemplo SG90.

A base na figura 5.6 apresenta uma borda interior que permite o encaixe da primeira articulação. Para garantir maior fixação foram adicionados furos para parafusos com 2.5 mm de diâmetro.

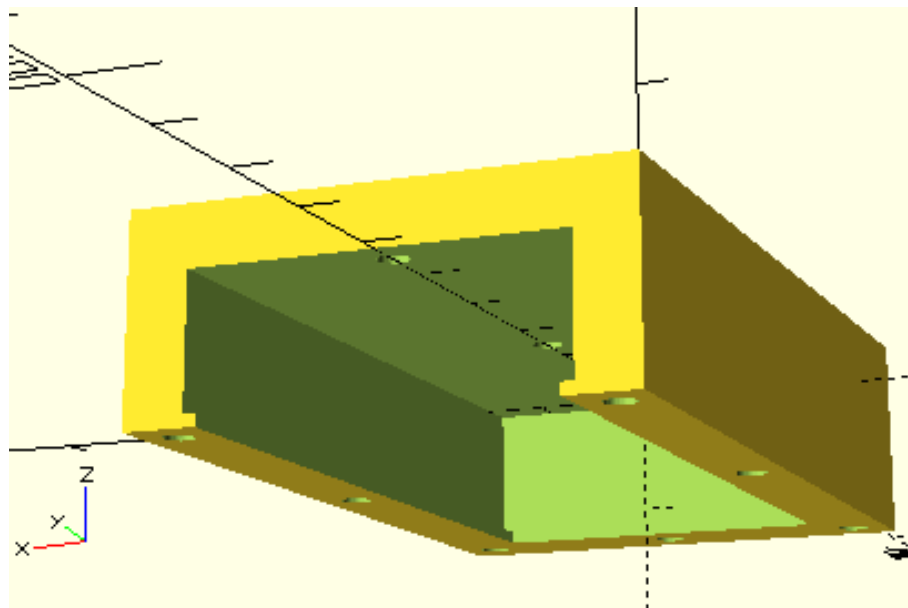


Figura 5.6: Base projetada.

O resultado esperado da interligação da base com a estrutura é possível observar na figura 5.7.

É possível ainda observar a estrutura completa na figura do lado esquerdo da 5.8 e a figura sem os servomotores na figura do lado direito da 5.8

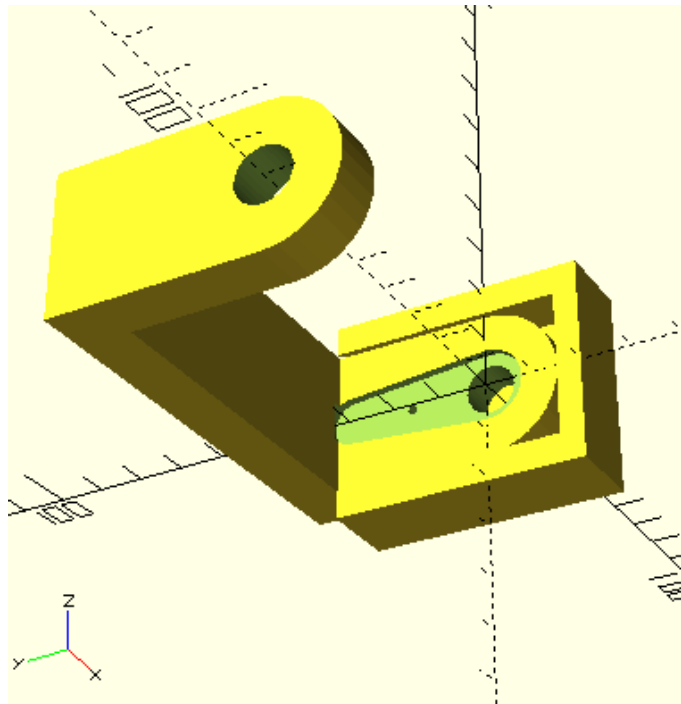


Figura 5.7: Interligação entre a base e a estrutura.

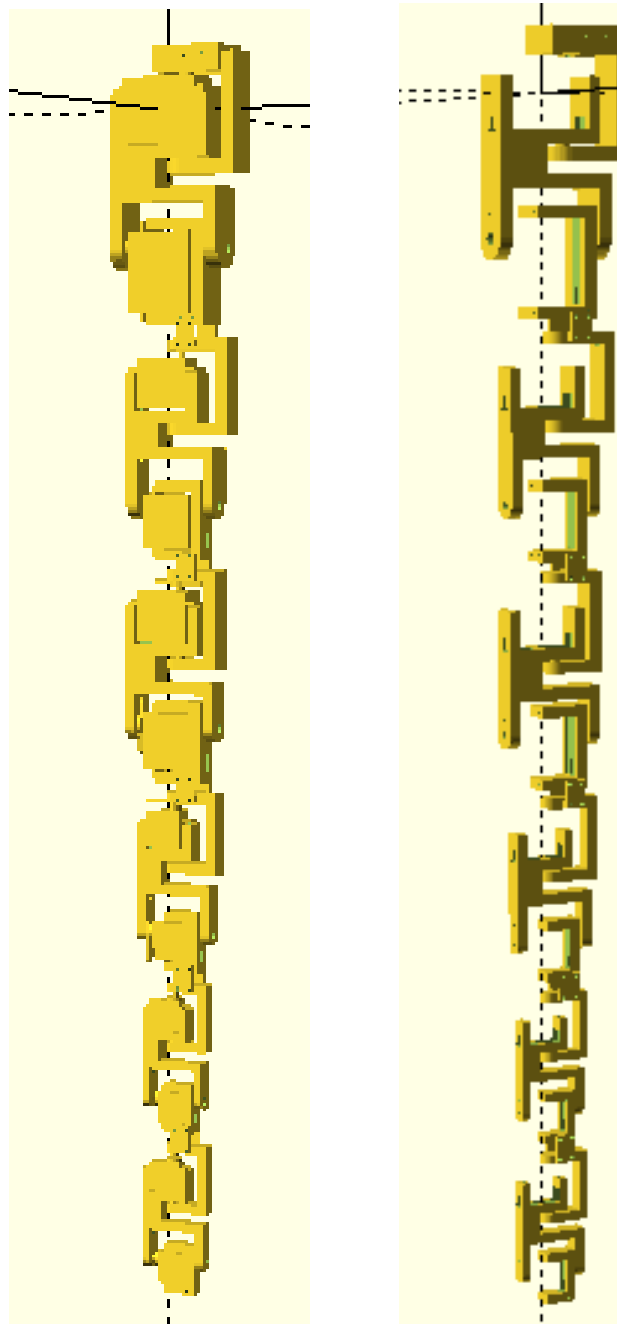


Figura 5.8: Estrutura em OpenScad.

5.4 Resultados

Após a impressão é possível observar a *hinge* e a base na figura 5.9, a *sleeve* na figura 5.10 e a peça de suporte na base do servomotor que a interliga com a dobradiça, 5.11. A conjugação das duas peças bases pode ser visualizada em 5.12.

A figura 5.13 apresenta a base da estrutura e os primeiros servomotores.

A figura 5.14 apresenta a ligação entre as diversas peças da estrutura.

A figura 5.15 apresenta a interligação entre a estrutura e a garra após impressão.

Por fim, é possível observar a estrutura completa na figura 5.16.

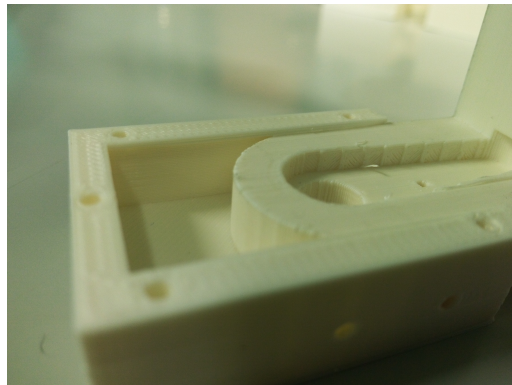


Figura 5.9: *Hinge* e a base após impressão.

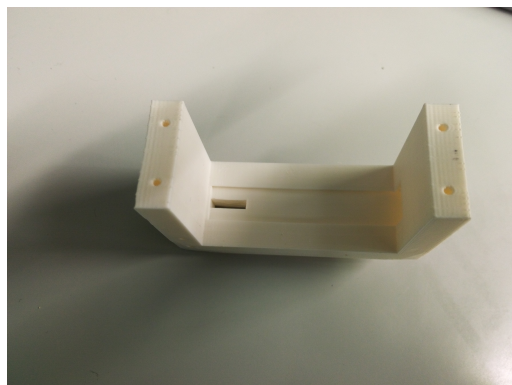


Figura 5.10: *Sleeve* após impressão.

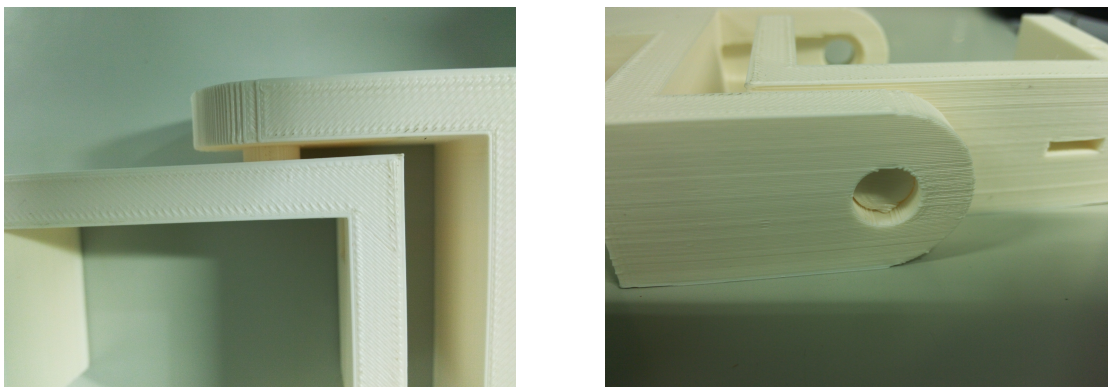


Figura 5.11: Peça de suporte na base do servomotor que a interliga com a dobradiça.

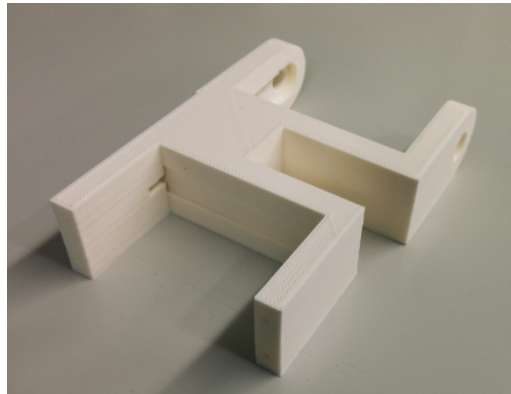


Figura 5.12: *Hinge* e *sleeve* com ligação retangular entre as duas após impressão.

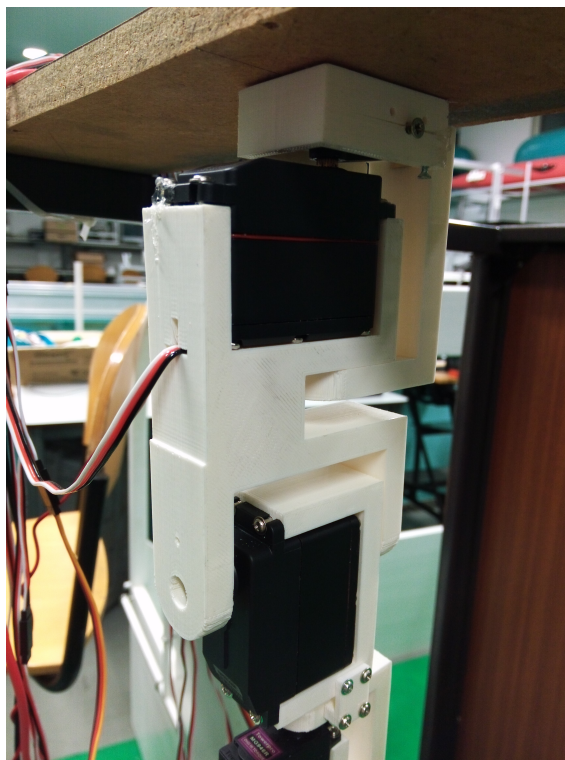


Figura 5.13: Base e primeiros servomotores.

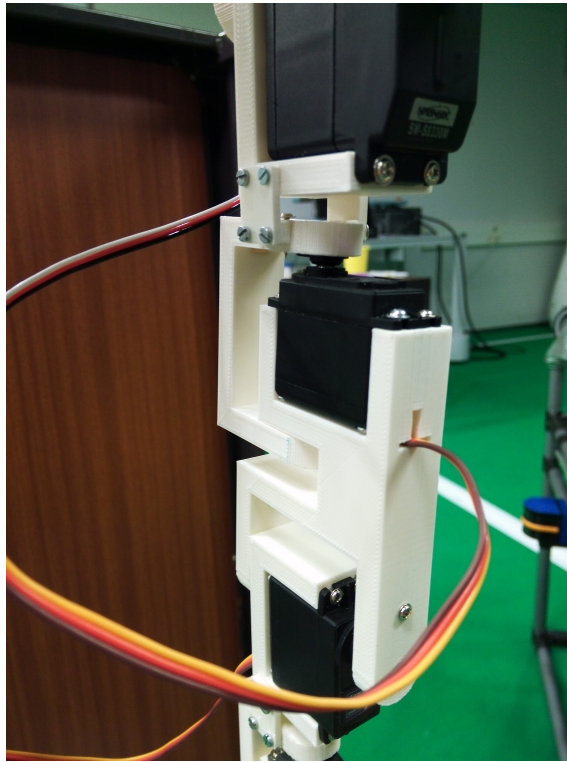


Figura 5.14: Ligação após impressão.

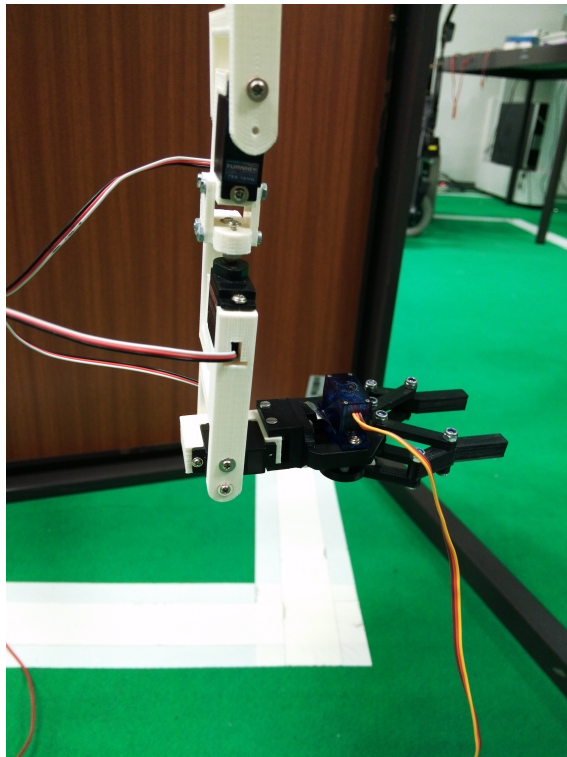


Figura 5.15: Interligação entre a estrutura e a garra.



Figura 5.16: Estrutura após impressão.

Capítulo 6

Validação Experimental

A validação experimental terá em consideração a estrutura mecânica [6.2](#), a movimentação de todos os servomotores [6.3](#) e o cumprimento de uma trajetória [6.5](#).

Os testes para validação serão descritos nas secções seguintes.

6.1 Introdução

A estrutura mecânica para validação não pode ser a dimensionada nos capítulos anteriores devido a atrasos nos servomotores encomendados. Após alguma deliberação, a dissertação avançou para outro conjunto de servomotores:

- 2 x *high torque* servos - [S8330M](#);
- 2 x servo - [MG946R](#);
- 2 x servo - [HK15328](#);
- 2 x *micro* servos - [HD1160A](#);
- 2 x *micro* servos - [TSS10MG](#);
- 2 x *micro* servos - [MAX3002](#).

Estes servomotores cumprem os requisitos mencionados anteriormente para a alimentação, sendo que um dos servomotores não cumpre para o binário máximo a exercer, como é possível observar na figura [6.1](#).

Servomotor	Binário	Potência	Requisito Binário Máximo	Requisito Potência
S8330M	23.8 kg.cm	5V – 2.7 A <u>stall</u>	1.972 kg.cm	5V – 5.4 A
MG946R	12 kg.cm	5V – 2 A <u>stall</u>	9.877 kg.cm	5V – 4 A
HK15328	10 kg.cm	5V – 1.86 A <u>stall</u>	5.347 kg.cm	5V – 3.72 A
HD1160A	2 kg.cm	5V – 0.84 A <u>stall</u>	2.772 kg cm	5V – 1.68 A
TSS10MG	2.2 kg.cm	5V – 1.02 A <u>stall</u>	1.518 kg.cm	5V – 2.04 A
MAXPRO3002	1.9 kg.cm	5V – 0.81 A <u>stall</u>	0.638 kg.cm	5V – 1.62 A
Total	-	-	-	5V - 18.3/20 A

Figura 6.1: Análise de requisitos dos servomotores utilizados.

6.2 Validação da Estrutura Mecânica

A estrutura mecânica será validada experimentalmente com a interligação de toda a estrutura e verificação da mobilidade e da dimensão correspondente.

A mobilidade da *hinge* está comprovada pela figura 6.2.

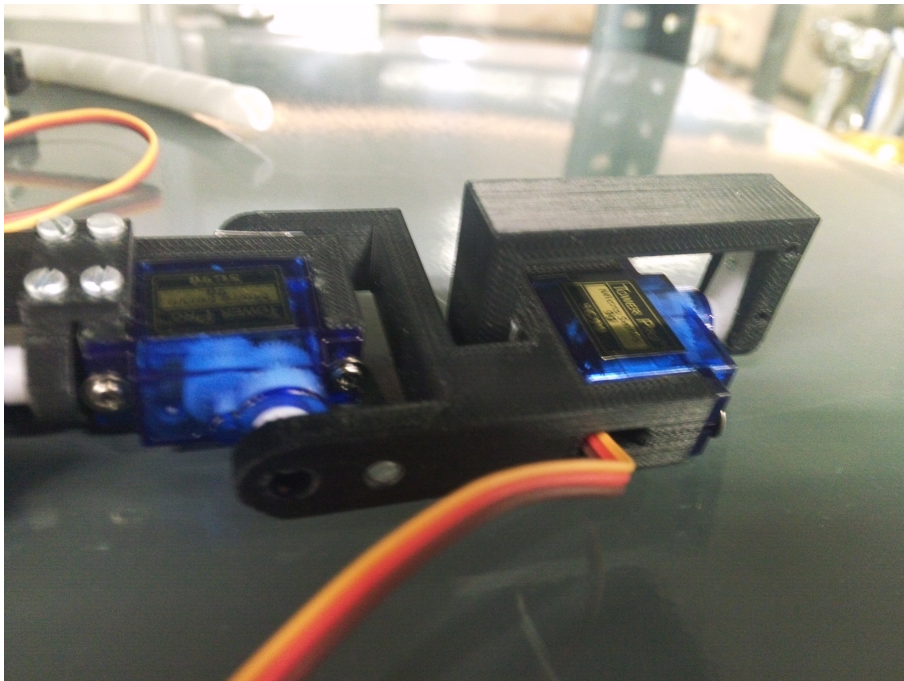


Figura 6.2: Rotação de 90° da *hinge*.

A mobilidade da *sleeve* está comprovada pela figura 6.3.

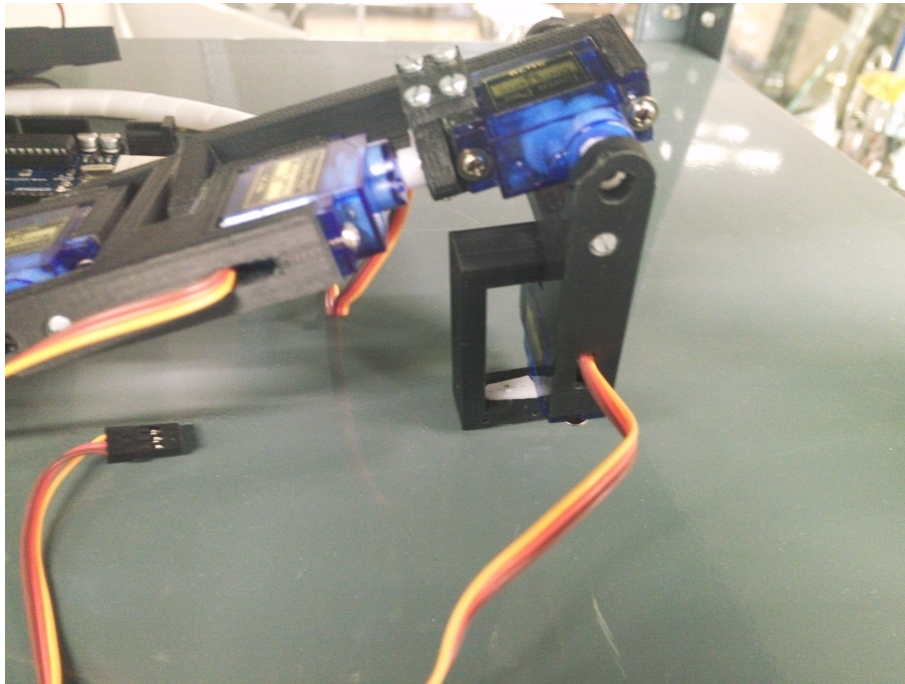


Figura 6.3: Rotação de 90° da *sleeve*.

6.3 Validação da Movimentação

Após a validação da estrutura estar concluída, é possível validar a movimentação do servomotor de acordo com a posição fornecida. Este teste será realizado sem qualquer trajetória para ponto específico.

Tal como referido anteriormente, o teste de validação da movimentação poderia ser realizado recorrendo a um potenciômetro ou ao envio de uma *string*, assim como validando todos os servomotores em simultâneo ou um a um.

Neste caso, os testes a realizar apenas confirmarão que os servomotores estão aptos a receber um determinado ângulo de referência e se deslocam até lá, como é possível observar nas imagens que se seguem e no vídeo disponível no seguinte [url](#).

Também é possível observar na figura 6.8 que consegue atingir posições mais complexas.

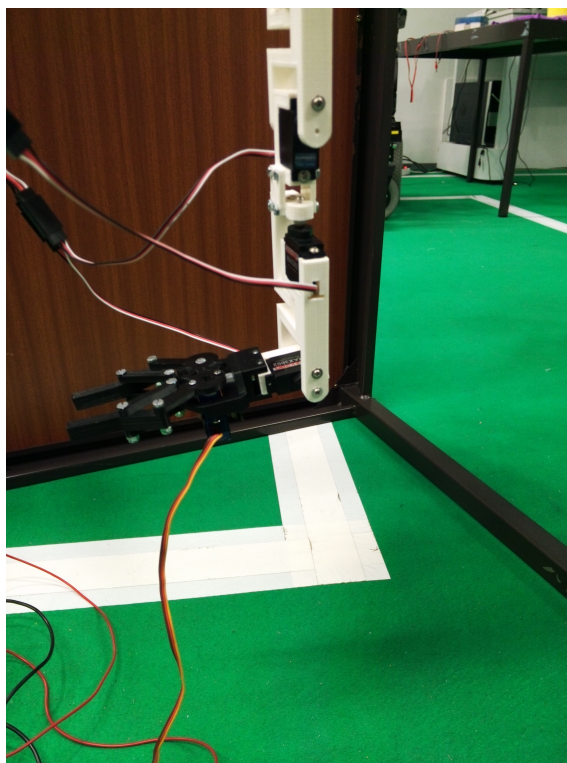


Figura 6.4: Movimentação para -90° .

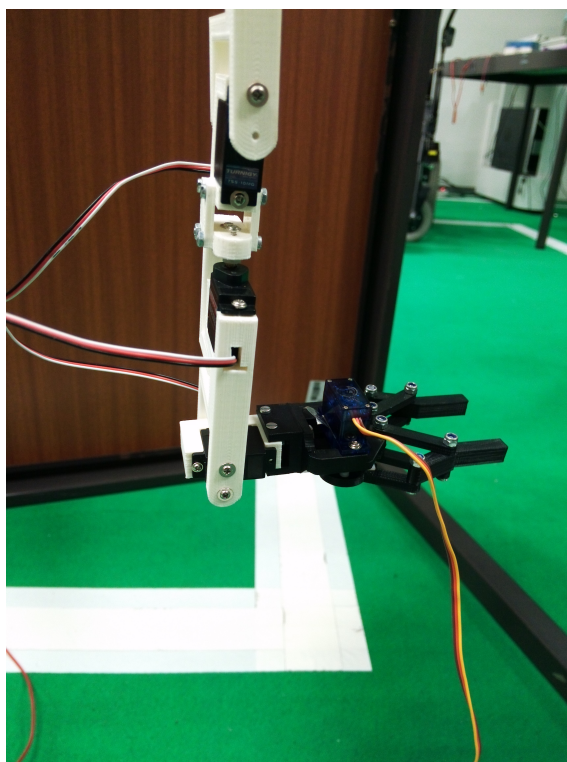


Figura 6.5: Movimentação para 90° .

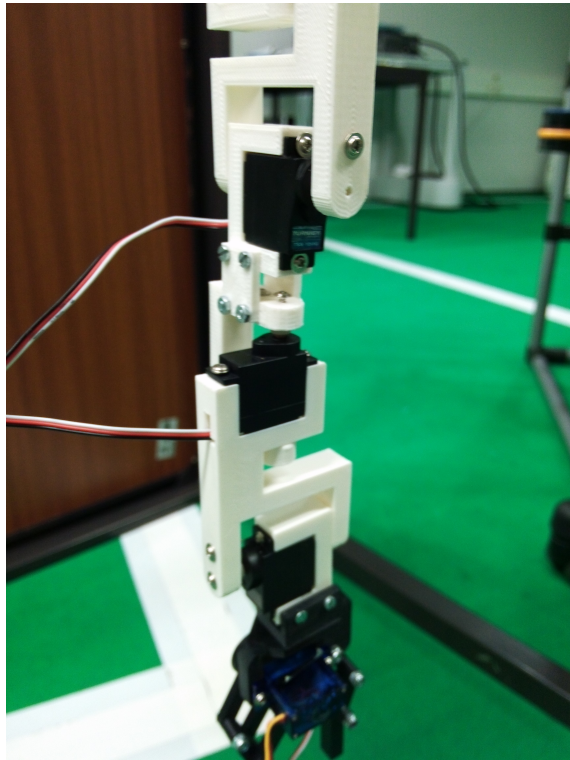


Figura 6.6: Movimentação para -90° .

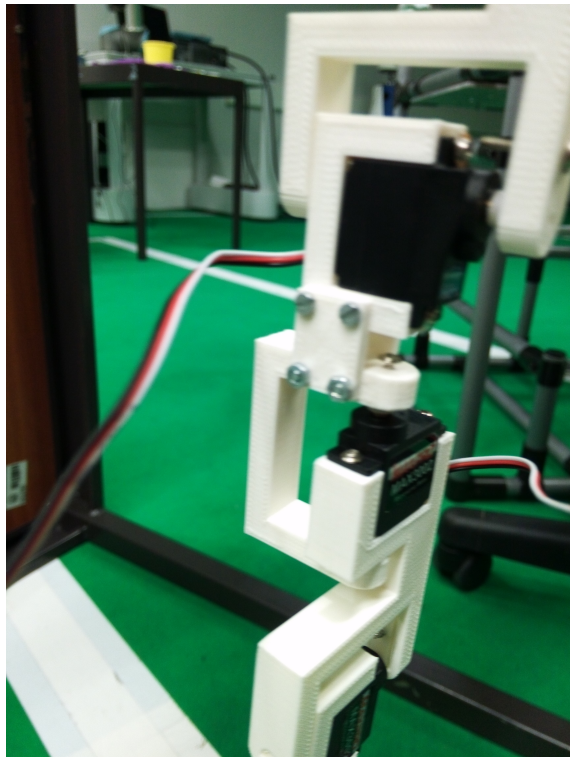


Figura 6.7: Movimentação para 90° .

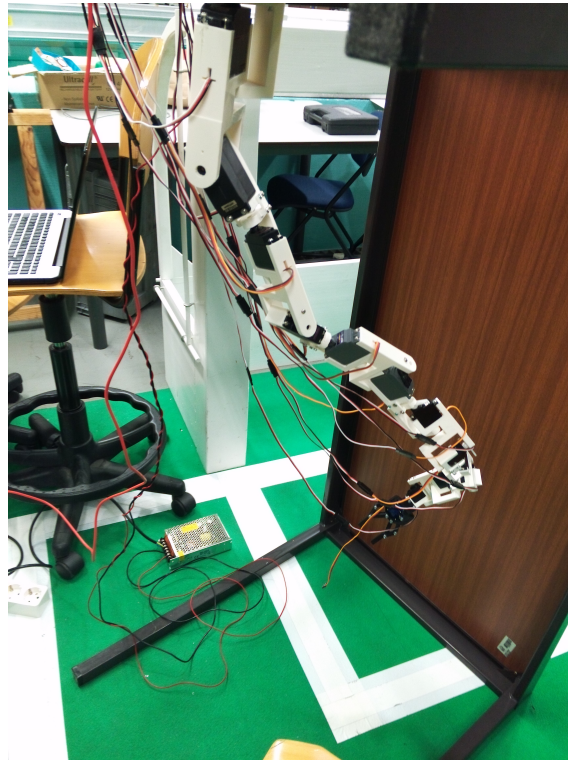


Figura 6.8: Movimentação complexa.

6.4 Teste de Posição

O teste de posição é consequente da validação acima demonstrada. Este teste determinará o grau de exatidão de cada articulação através da comparação entre a movimentação pretendida e a feita na realidade pelo manipulador.

Abaixo encontra-se a tabela 6.1 com o sumário dos desvios máximos obtidos após medições para diversas posições.

Tabela 6.1: Desvios de cada articulação.

Servomotor	Desvio (°)
Servomotor 0	5
Servomotor 1	5
Servomotor 2	1
Servomotor 3	5
Servomotor 4	7
Servomotor 5	6
Servomotor 6	1
Servomotor 7	5
Servomotor 8	5
Servomotor 9	1
Servomotor 10	1
Servomotor 11	1

6.5 Validação das Trajetórias

A validação das trajetórias consistirá na junção desta dissertação com o planeamento da trajetória feito em paralelo de forma a verificar que o manipulador se encontra capaz de atingir determinado ponto seguindo a trajetória definida.

O comprovativo da validação encontra-se no canal criado para o efeito no [youtube](#).

6.6 Resultados

É possível observar que a estrutura mecânica está validada.

Relativamente à movimentação, é possível observar que os testes foram realizados com sucesso.

O teste de posição permitiu verificar que apesar das análises teóricas efetuadas, existe um erro na maior parte das articulações. Apesar dos valores apresentados, deverão ser realizados testes com maior precisão.

É possível observar que o manipulador cumpre, dentro das suas limitações, o conjunto de pontos que define a trajetória.

Capítulo 7

Conclusões e Trabalho Futuro

O objetivo principal da dissertação foi construir um manipulador robótico capaz de se movimentar num espaço reduzido com obstáculos e atingir um vasto número de posições, tendo como requisito garantir que a solução fosse de baixo custo comparativamente com os manipuladores existentes no mercado.

Inicialmente, foi efetuada uma análise em função da carga a transportar, foi escolhido o equipamento a utilizar, a estrutura mecânica do manipulador foi projetada em OpenSCAD, impressa e validada, servindo como base do protótipo. Foi projetado e validado um sistema de atuação das articulações, tendo como referência a utilização de servomotores e o seu controlo em posição.

O controlo interno dos servomotores não permite regular a velocidade a que a *shaft* se desloca. Foi implementando um sistema iterativo com um atraso temporal para permitir um deslocamento fluído mas restringindo a velocidade a que se desloca. A modificação dos servomotores e o respetivo controlo de velocidade foi apenas realizado num motor.

Constituído por servomotores e peças de ligação e suporte, facilmente removíveis, fica acessível testar diferentes configurações e escolher quantos DOF se pretende. O seu controlo em posição permitiu uma boa fiabilidade para atingir posições complexas, um contorno de formas e desvio de obstáculos.

A sua construção validou ainda algoritmos de planeamento de trajetória e de cinemática que foram desenvolvidos paralelamente a esta dissertação [25]. Também validou algoritmos de estimação para controlo de posição das articulações e da ferramenta de um manipulador em operações *human like*.

7.1 Trabalho Futuro

O trabalho apresentado pode ser melhorado nalguns campos.

Apesar dos pontos positivos relativamente aos demais manipuladores industriais, existem também alguns a melhorar, nomeadamente o parâmetro de exatidão. Um sistema de *feedback* por visão, poderia ser útil para corrigir certas formas e diminuir o erro detetado no teste de exatidão.

A implementação do sistema de controlo de velocidade a todos os servomotores. A integração adicional de um estimador da velocidade atual, permitiria obter maior robustez no controlador.

Ainda no sistema de atuação, a *driver* de alimentação dos motores deve ser redimensionada de forma a garantir que não existam limitações de corrente, devem ser realizados testes aos servomotores para validar o modelo. A implementação de uma metodologia de controlo baseada em lógica difusa, PID ou Kalman com a realização de testes de comparação. A nível de *hardware*, poderão ser adicionados sensores de corrente aos servomotores de forma a garantir que caso exista um obstáculo não referenciado, o servomotor não fique a forçar a posição e, por consequência, a aumentar a corrente de forma não controlada. Introduzir um estimador dos parâmetros não lineares do motor.

A introdução de um elemento de *feedforward* permitira uma convergência mais rápida para o controlador. Tipicamente, este elemento poderia ser o inverso do modelo dinâmico do sistema ou um mapa com alguns dados conhecidos em função do parâmetro de entrada.

Ao nível da estrutura mecânica, deveriam ser realizados testes de stress às peças impressas.

Integrar esta dissertação com métodos de cinemática num microcomputador que comunique diretamente com o arduino de forma a ter um sistema de controlo dedicado. Após esta integração, seria interessante adicionar um *joystick* para movimentar a ferramenta, movimentando consequentemente as articulações de acordo com uma otimização escolhida, como uma otimização da potência consumida, por exemplo.

Adicionar o planeamento de trajetórias ao microcomputador mencionado anteriormente, permitiria adicionar obstáculos e o manipulador ficaria com uma solução robusta a nível industrial.

Por fim, implementar uma interface gráfica para *design* das peças e uma plataforma de testes padronizada para obtenção dos parâmetros dos motores.

Referências

- [1] António Moreira. Lecture notes in industrial robotics (eec0093), June 2015.
- [2] Mark W. Spong, Seth Hutchinson, e M. Vidyasagar. *Robot modeling and control*. Hoboken, NJ : John Wiley & Sons, c2006., 2006.
- [3] José Luís Sousa Magalhães Lima. Construção de um modelo realista e controlo de um robô humanóide , 2009. URL: <http://eos.fe.up.pt:1801/webclient/DeliveryManager?custom{ }att{ }2=simple{ }viewer{&}metadata{ }request=false{&}pid=53234>.
- [4] International Organization for Standardization. Technical Committee Automation systems, integration. Subcommittee Robots, e robotic devices. *ISO 8373: Robots and Robotic Devices - Vocabulary*. ISO, 2012.
- [5] Gregory S Chirikjian. *Theory and applications of hyper-redundant robotic manipulators*. CIT theses: 1992. Pasadena, Calif. : California Institute of Technology, c1992, 1992. URL: <https://clsproxy.library.caltech.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat04350a&AN=caltech.581997&site=eds-live&scope=site>.
- [6] Gregory S. Chirikjian e Joel W. Burdick. A hyper-redundant manipulator. *IEEE Robotics & Automation Magazine*, December 1994.
- [7] G. S. Chirikjian e J. W. Burdick. Design and experiments with a 30 dof robot. Em *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, páginas 113–119 vol.3, May 1993. doi:10.1109/ROBOT.1993.291862.
- [8] X. Xu, H. Ananthanarayanan, e R. Ordóñez. Design and construction of 9-dof hyper-redundant robotic arm. Em *NAECON 2014 - IEEE National Aerospace and Electronics Conference*, páginas 321–326, June 2014. doi:10.1109/NAECON.2014.7045828.
- [9] Mario Saenz Espinoza, Jose Goncalves, Paulo Leitao, Jose Luis Gonzalez Sanchez, e Alberto Herreros. Inverse kinematics of a 10 dof modular hyper-redundant robot resorting to exhaustive and error-optimization methods: A comparative study. Em *Proceedings of the 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, SBR-LARS '12*, páginas 125–130, Washington, DC, USA, 2012. IEEE Computer Society. URL: <http://dx.doi.org/10.1109/SBR-LARS.2012.28>, doi:10.1109/SBR-LARS.2012.28.
- [10] Philip Mckerrow. *Introduction to Robotics*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edição, 1991.
- [11] E. Sahin Conkur e Rob Buckingham. Clarifying the definition of redundancy as used in robotics. *Robotica*, 15:583–586, 9 1997. doi:null.

- [12] Richard M. Murray, S. Shankar Sastry, e Li Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edição, 1994.
- [13] T. W. Danko e P. Y. Oh. A hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles. Em *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, páginas 974–981, May 2013. doi:10.1109/ICUAS.2013.6564784.
- [14] Mário Saenz Espinoza, Ana I. Pereira, e José Gonçalves. Optimization methods for hyper-redundant robots' inverse kinematics in biomedical applications. Em *AIP Conference Proceedings*, volume 1479, páginas 818–821, 2012.
- [15] H. B. Brown, M. Schwerin, E. Shamma, e H. Choset. Design and control of a second-generation hyper-redundant mechanism. Em *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 2603–2608, Oct 2007. doi:10.1109/IROS.2007.4399504.
- [16] John J. Craig. *Introduction to robotics : mechanics and control*. Addison-Wesley series in electrical and computer engineering. Control engineering. Reading, Mass. : Addison-Wesley, c1989., 1989.
- [17] Bruno Siciliano e Oussama Khatib. *Springer handbook of robotics*. Berlin : Springer, c2008., Fourth edição, 2008.
- [18] T. Arai, K. Cleary, T. Nakamura, H. Adachi, e K. Homma. Design, analysis and construction of a prototype parallel link manipulator. Em *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, páginas 205–212 vol.1, Jul 1990. doi:10.1109/IROS.1990.262388.
- [19] L. Sciavicco e Bruno Siciliano. *Modelling and control of robot manipulators*. Advanced textbooks in control and signal processing. London ; New York : Springer, c2000., 2000.
- [20] Austin Hughes. *Electric motors and drives : fundamentals, types, and applications*. Amsterdam ; Elsevier/Newnes, 2013., 2013. URL: <http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=cat00916a&AN=mit.002354286&site=eds-live>.
- [21] A. J. Bianculli. Stepper motors: application and selection. *IEEE Spectrum*, 7(12):25–29, Dec 1970. doi:10.1109/MSPEC.1970.5213082.
- [22] NXP Semiconductors. *UM10204: I2C-bus specification and user manual*. NXP Semiconductors, 2014.
- [23] Paulo Costa A, José Gonçalves B, José Lima B, e Paulo Malheiros A. Simtwo realistic simulator: A tool for the development and validation of robot software.
- [24] Robert H. Bishop. *The mechatronics handbook*. The electrical engineering handbook series. Boca Raton, Fla. : CRC Press, c2002., 2002.
- [25] Alexandre Pires. *Model of an hyper reduntant manipulator and respective path planning*. Tese de mestrado integrado : Engenharia Electrotécnica e de Computadores , Faculdade de Engenharia. Universidade do Porto . 2016. Porto : FEUP, 2016.