

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Articulação Modular para Braços Robóticos

Marco António Mendonça Rocha

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Paulo José Cerqueira Gomes da Costa

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

28 de julho de 2021

Resumo

Os manipuladores robóticos têm como principal objectivo aumentar a repetibilidade e produtividade em vários processos. No entanto, por vezes as soluções disponíveis comercialmente não apresentam flexibilidade arquitectónica suficiente para serem colocadas directamente em alguns desses processos. Consequentemente, as articulações modulares pretendem compensar essa mesma falta de flexibilidade, permitindo que se configure braços robóticos com uma configuração apropriada para a sua aplicação. Isto é, utilizam-se tantas articulações modulares quanto o número de graus de liberdade que se pretende que o manipulador tenha, e interligam-se fisicamente as articulações de forma a favorecer a tarefa que desempenhará.

Desta forma, esta dissertação tem como objectivo desenvolver uma Articulação Modular para Braços Robóticos, que seja compacta e facilmente reconfigurável, que tenha controladores de velocidade e posição incorporados, que possua um meio de comunicação com o exterior, e que tenha um custo final seja reduzido. Adicionalmente, esta articulação terá também um conjunto de sensores que permitem assegurar o seu controlo e devido funcionamento, tais como: *encoder* magnético absoluto, sensor de Hall para detectar a "*Home Position*", sensor de tensão, sensor de corrente e sensor de temperatura.

Como tal, ao longo deste documento são focados todos os aspectos associados ao desenvolvimento da articulação. Desde o conceito ao produto final, passando pelo desenho da caixa redutora, pela integração dos sensores e dos actuadores, pelo protocolo de comunicação, e pelos controladores e respectiva afinação.

Abstract

Robotic manipulators have as main goal to increase repeatability and productivity in various processes. However, sometimes commercially available solutions don't have enough architectural flexibility to be placed directly in some of those processes. Consequently, modular joints are intended to compensate for this exact lack of flexibility, allowing to configure robotic arms with an appropriate configuration for a specific application. That is, as many joints can be used as the number of degrees of freedom that favors the most a given task, and the joints can be physically interconnected in a way that also favors the same task.

Thus, this dissertation aims to develop a Modular Joint for Robotic Arms, which is compact and easily reconfigurable, and which also has built-in speed and position controllers, means of communication with the outside, and a low final cost. Additionally, this joint also has a set of sensors that ensure its control and proper functioning, such as: an absolute magnetic encoder, an Hall sensor to detect the joint's "Home Position", a voltage sensor, a current sensor and a temperature sensor.

This document focuses on the main aspects associated with the joint's development. From the concept to the final product, passing through: the gearbox design, the sensors and actuators integration, the communication protocol, and the controllers and respective tuning.

Agradecimentos

Em primeiro lugar, quero começar por agradecer ao Professor Doutor Paulo Costa e ao Professor Doutor José Lima, por toda a disponibilidade, paciência e apoio ao longo deste processo. Nos momentos de maior complexidade aprontaram-se sempre a auxiliar-me.

Em segundo lugar quero agradecer ao Professor Vítor Pinto, especialmente pelo empréstimo da Impressora 3D, o que permitiu acelerar o processo.

Adicionalmente, quero também agradecer a todos os restantes professores que tive ao longo do curso. Visto que estes sempre se dispuseram a ajudar, tanto em trabalhos curriculares como extra-curriculares, sempre em prol do nosso conhecimento.

De seguida, quero agradecer aos meus colegas e amigos de curso, que, mesmo além do ambiente académico, contribuíram directa e indirectamente para o bom término deste percurso.

Quero agradecer também à minha namorada, pelo acompanhamento contínuo, e por ser uma verdadeira companheira para a vida.

Por fim, um agradecimento muito especial aos meus pais, à minha irmã, e a toda a minha família, por todo o carinho, apoio, e ajuda incondicional. Certamente foram o pilar principal ao longo destes 5 anos. Vocês são a prova viva de que força de vontade e empenho vence sempre.

Marco António Mendonça Rocha

*“Discovery consists of looking at the same thing as everyone else
and thinking something different.”*

Albert Szent-Gyorgyi

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	1
1.3	Objectivos	2
1.4	Estrutura da Dissertação	2
2	Estado da Arte	3
2.1	Análise de Casos: Articulações Modulares	3
2.1.1	Proposta de Solução para uma Articulação Modular	3
2.1.2	Articulação Modular para Robô <i>dual-arm</i>	4
2.1.3	Articulação Modular com Redução de Velocidade Variável	5
2.2	Mercado	6
2.2.1	HEBI Robotics	6
2.2.2	Rozum Robotics	7
3	Fundamentação Teórica	9
3.1	Sistemas de Actuação para Articulações Rotativas	9
3.1.1	Caixa Redutora de Parafuso Sem-fim	9
3.1.2	Caixa Redutora Cicloidal	10
3.1.3	Caixa Redutora Harmónica	11
3.1.4	Caixa Redutora Planetária	12
3.1.5	Caixa Redutora Planetária Composta	12
3.2	Tipos de Motores	13
3.2.1	Motor DC com Escovas de Ímanes Permanentes	13
3.2.2	Motor DC sem Escovas	14
3.2.3	Motor de Passo	15
3.3	Sensores	17
3.3.1	Sensor Potenciométrico	17
3.3.2	Sensor Ótico	18
3.3.3	Sensor Magnético	21
3.4	Metodologias de Controlo de Articulações	23
3.4.1	Controlador PID	23
3.4.2	Controlador PIV	26
3.4.3	Arquitecturas em Cascata e Paralelo	27
3.4.4	Realimentação de Estado e Observadores de Estado	29
3.5	Estimação dos Parâmetros do Motor	32
3.6	Microcontroladores	34
3.7	Ferramentas e Tecnologias	35

3.7.1	Ferramentas de Projecto CAD	35
3.7.2	Maquinagem CNC	36
3.7.3	Impressão 3D	36
4	Implementação	39
4.1	Hardware	40
4.1.1	Caixa Redutora de Velocidade	40
4.1.2	Motor	43
4.1.3	<i>Driver</i> de Motor	44
4.1.4	<i>Encoder</i> Absoluto Magnético	45
4.1.5	Sensor de <i>Home Position</i>	46
4.1.6	Sensor de Tensão	47
4.1.7	Sensor de Corrente	48
4.1.8	Sensor de Temperatura	49
4.1.9	Interface de Comunicação RS-485	49
4.2	Software	50
4.2.1	<i>Firmware</i> do Microcontrolador	50
4.2.2	Protocolo de Comunicação	52
4.2.3	Software do <i>Master</i>	54
4.3	Algoritmos de Controlo	55
4.3.1	Controlador PID em Tempo Discreto	55
4.3.2	Controlador PI de Velocidade com Módulo <i>Feedforward</i>	56
4.3.3	Controlador PD de Posição	59
4.3.4	Controlador em Cascata com Compensação de Zona Morta	61
4.4	Estimação dos Parâmetros da Articulação	63
5	Resultados	69
5.1	Resultados Construtivos	69
5.2	Resultados de Desempenho	71
6	Conclusão e Trabalho Futuro	75
6.1	Conclusão	75
6.2	Trabalho Futuro	76
A	Figuras Adicionais	79
B	Artigo Publicado	87
	Referências	101

Lista de Figuras

2.1	Modelo da articulação à esquerda e articulação concluída à direita [1].	4
2.2	Articulação modular à direita e respectivo modelo à esquerda [2].	4
2.3	Articulação com relação de redução variável [3].	5
2.4	Esquema construtivo da articulação com relação de redução variável [3].	6
2.5	Articulação modular Hebi Robotics R8-16 [4].	7
2.6	Articulação modular Rozum RDrive 50 [5].	8
3.1	Caixa redutora de parafuso sem-fim [6].	10
3.2	Vista 1D e seccionada de uma caixa redutora cicloidal [7].	10
3.3	Caixa redutora harmónica: (a) componentes principais; (b) interação entre os componentes [8].	11
3.4	Composição de uma caixa redutora planetária [9].	12
3.5	Caixa redutora planetária composta com planetas interligados [9].	13
3.6	Construção física do motor DC: à esquerda com escovas; à direita <i>brushless</i> [10].	14
3.7	Construção de motor de passo de ímanes permanentes [11].	16
3.8	Construção de motor de passo de relutância variável [11].	16
3.9	Princípio de funcionamento de um sensor potenciométrico [12].	17
3.10	Princípio de funcionamento de um sensor ótico [12].	18
3.11	Modelo de um <i>encoder</i> ótico incremental e respetivos sinais gerados [12].	19
3.12	Modelo de um <i>encoder</i> ótico absoluto de 4 bits [13].	20
3.13	Sensor de Hall e pólos magnéticos alternados [12].	21
3.14	Múltiplos sensores de Hall e um íman magnetizado diametralmente [12].	22
3.15	Arquitectura típica de um controlador PID.	24
3.16	Esquema de controlo com componente de <i>feedforward</i>	25
3.17	Arquitetura típica de um controlador PIV.	26
3.18	Arquitectura típica de um controlador em cascata.	27
3.19	Arquitectura típica de controlo em paralelo.	28
3.20	Circuito equivalente de um motor DC.	33
4.1	Conceito de sistema da Articulação Modular para Braços Robóticos.	39
4.2	Vista seccionada da Articulação Modular para Braços Robóticos.	40
4.3	Engrenagem cilíndrica com dentes helicoidais simétricos.	41
4.4	Pontos de fixação e veio de saída da caixa redutora.	42
4.5	Caixa redutora de velocidade completa.	43
4.6	Motor DC escovado de ímanes permanentes: Surpass Hoby 550-35T.	44
4.7	<i>Driver</i> de motor BTS7960 [14].	44
4.8	Configuração mecânica do <i>encoder</i> AS5600 [15].	45
4.9	Vista seccionada da colocação do <i>encoder</i> AS5600.	46

4.10	Esquema do circuito do sensor de Efeito de Hall A3144.	46
4.11	Vista seccionada da colocação do sensor de Efeito de Hall A3144.	47
4.12	Esquema do divisor resistivo para medição da tensão de alimentação.	47
4.13	Esquema do circuito do sensor de corrente ACS712-30A.	48
4.14	Esquema do circuito do sensor de temperatura DS18B20.	49
4.15	Esquema do circuito do conversor MAX485CSA.	50
4.16	Fluxograma do funcionamento do <i>firmware</i> do microcontrolador.	52
4.17	Formato das tramas enviadas pelo <i>Master</i>	53
4.18	Formato das tramas enviadas pelos <i>Slaves</i>	53
4.19	Diagrama de blocos do Controlador PI de Velocidade com <i>Feedforward</i>	57
4.20	Resposta da articulação a um degrau de velocidade: PWM com <i>duty-cycle</i> inicial de 23,5% e final de 47%.	58
4.21	Velocidade angular (rad/s) da articulação em função do <i>duty-cycle</i> do PWM.	59
4.22	Diagrama de blocos do Controlador PD de Posição.	60
4.23	Diagrama de blocos do Controlador PD em malha fechada.	61
4.24	Diagrama de blocos do Controlador em Cascata com Compensação de Zona Morta (DZC).	62
4.25	Velocidade angular da articulação em função da tensão.	63
4.26	Corrente consumida pela articulação em função da tensão.	64
4.27	Razão V/ω em função da razão i/ω	64
4.28	Binário $K_t i_a$ gerado pela articulação em função da velocidade angular ω	65
4.29	Comparação dos dados experimentais aos gerados pelas Equações 4.23 e 4.24.	66
4.30	Comparação das respostas ao degrau dadas pelas funções de transferência estimadas.	67
5.1	Articulação Modular com braço de testes de 200 mm.	70
5.2	Resposta do Controlador PI de Velocidade com FF a diferentes comandos de velocidade.	71
5.3	Resposta do Controlador PD de Posição a diferentes comandos de posição.	72
5.4	Resposta do Controlador em Cascata com DZC a diferentes comandos de posição.	72
A.1	Vista lateral seccionada da Articulação Modular para Braços Robóticos.	79
A.2	Componentes impressos da Articulação Modular para Braços Robóticos.	80
A.3	Articulação Modular para Braços Robóticos - vista lateral frontal.	81
A.4	Articulação Modular para Braços Robóticos - vista lateral traseira.	82
A.5	Articulação Modular para Braços Robóticos - vista lateral.	83
A.6	Articulação Modular para Braços Robóticos - vista frontal.	84
A.7	Sistema completo: Articulação Modular para Braços Robóticos e respectivos componentes electrónicos.	85
A.8	Exemplo de configuração: Manipulador com dois graus de liberdade/articulações.	86

Lista de Tabelas

4.1	Número de dentes de cada componente da caixa redutora.	41
5.1	Lista detalhada dos custos de produção da articulação.	70
5.2	Características construtivas da Articulação Modular.	71
5.3	Erros transitórios dos controladores.	73
5.4	Erros dos controladores em regime estacionário.	74

Abreviaturas

AC	<i>Alternating Current</i> / Corrente Alternada
ADC	<i>Analogue-to-digital Converter</i> / Conversor Analógico-Digital
BLDC	<i>Brushless DC (motor)</i> / Motor DC sem escovas
BJ	<i>Binder Jetting</i>
BW	<i>Bandwidth</i> / Largura de Banda
CAD	<i>Computer-aided Design</i> / Desenho assistido por computador
CAN	<i>Controller Area Network</i>
CNC	<i>Computer Numeric Control</i> / Controlo Numérico Computorizado
DC	<i>Direct Current</i> / Corrente contínua
DE	<i>Driver Enable</i>
DI	<i>Driver Input</i>
DSP	<i>Digital Signal Processor</i>
DZC	<i>Dead Zone Compensation</i> / Compensação de Zona Morta
FDM	<i>Fused Deposition Modeling</i> / Fabricação por Extrusão de Filamento Plástico
IMC	<i>Internal Model Control</i>
IP	<i>Ingress Protection</i> / Grau de Proteção IP
LED	<i>Light-emitting Diode</i> / Díodo Emissor de Luz
PD	<i>Proportional and Derivative</i> / Proporcional e Derivativo
PI	<i>Proportional and Integral</i> / Proporcional e Integral
PID	<i>Proportional Integral and Derivative</i> / Proporcional Integral e Derivativo
PIV	<i>Proportional Integral and Velocity</i> / Proporcional Integral e Velocidade
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random-access Memory</i>
RE	<i>Receiver Enable</i>
RO	<i>Receiver Output</i>
ROM	<i>Read-only Memory</i>
RPM	Rotações por Minuto
SLA	<i>Stereolithography</i> / Estereolitografia
SLS	<i>Selective Laser Sintering</i> / Sinterização a Laser Seletiva
SPI	<i>Serial Peripheral Interface</i>

Capítulo 1

Introdução

Neste primeiro capítulo introduz-se a dissertação "Articulação Modular para Braços Robóticos". Esta introdução consiste na apresentação do contexto, da motivação para a realização desta dissertação, dos objectivos que se pretende atingir, e da estrutura seguida no desenvolvimento deste documento.

1.1 Contexto

A indústria pretende cada vez mais automatizar o maior número de processos possíveis, e para que tal seja possível recorrem frequentemente ao uso de braços robóticos.

Desta forma, como a variedade de processos a automatizar representa um vasto leque de opções, os braços robóticos têm de apresentar a maior flexibilidade possível relativamente à sua configuração, isto é, relativamente aos graus de liberdade e ao comprimento dos seus segmentos.

Consequentemente, as articulações modulares permitem que facilmente se configure um braço robótico com as características pretendidas. Ou seja, usa-se tantas articulações quanto o número de graus de liberdade necessários a uma dada aplicação, e coloca-se os segmentos de ligação entre as articulações de forma a favorecer a tarefa a desempenhar.

1.2 Motivação

As articulações modulares, como se pretende que tenham a maior flexibilidade possível na realização de tarefas, têm de reunir um conjunto de características que lhes permita ter um desempenho ao nível dos manipuladores robóticos tradicionalmente usados na indústria. No entanto, estas articulações têm normalmente associadas características como baixa relação binário/peso, elevado *backlash* e custo, que representam as características contrárias às pretendidas.

Isto é, caso se pretenda que a articulação produza mais binário, pode-se substituir o motor por um de maior potência, ou modificar a caixa redutora para se obter uma maior redução. No entanto, um motor de maior potência implica um aumento do custo e peso, e a caixa redutora com uma redução maior implica um aumento do *backlash*.

Caso se pretenda a redução do *backlash* opta-se normalmente por uma de duas abordagens: projeto e produção das engrenagens segundo técnicas mais precisas; ou opta-se por uma arquitectura para a caixa redutora que implique um menor *backlash*. No entanto, ambas as opções implicam normalmente custos de projeto e produção mais elevados.

Como tal, o projeto de uma articulação modular consiste num *trade-off* entre a relação binário/peso, *backlash* e custo.

1.3 Objectivos

Este trabalho tem como objectivo desenvolver uma proposta de solução para uma articulação modular para braços robóticos. Esta solução tem de ser controlável em velocidade e posição, tem de conseguir comunicar com outras articulações ou microcontroladores, e deve apresentar um custo final reduzido.

Adicionalmente, tem de ser o mais compacta possível, deve permitir uma conexão física com outras articulações, e tem de apresentar uma arquitetura facilmente reconfigurável.

1.4 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 5 capítulos. No próximo capítulo, que é o Capítulo 2, é desenvolvido o estado da arte, ou seja, são apresentados trabalhos relacionados e soluções semelhantes já existentes no mercado. No Capítulo 3 é realizada uma fundamentação teórica, isto é, abordam-se os assuntos a considerar no desenvolvimento da solução. No Capítulo 4 é descrito o processo de desenvolvimento da articulação modular, incluindo tanto o *hardware* como o *software*, e os algoritmos de controlo. No Capítulo 5 são apresentados os resultados obtidos com a solução desenvolvida. E no último capítulo, que é o Capítulo 6, são explanadas as conclusões tiradas no final deste trabalho, e apresentadas as possíveis melhorias a realizar em trabalhos futuros.

Capítulo 2

Estado da Arte

Neste capítulo serão abordados projectos existentes com teor semelhante ao desenvolvido nesta dissertação.

Inicialmente serão apresentados os projectos de conteúdo académico e de investigação. E de seguida serão apresentados produtos disponíveis comercialmente.

2.1 Análise de Casos: Articulações Modulares

Na Secção 2.1 deste documento é realizada uma análise de casos semelhantes ao que será desenvolvido nesta dissertação. Mais concretamente, serão apresentadas diferentes abordagens para se obter uma articulação modular para braços robóticos.

2.1.1 Proposta de Solução para uma Articulação Modular

A primeira solução aqui apresentada, representada na Figura 2.1, representa um método proposto por um conjunto de autores da Universidade de Beijing. Neste caso, a articulação modular que estes autores apresentam tem como principais objectivos: ter uma construção conveniente, ser de fácil montagem, ter boa agilidade e ser de fácil manutenção.

Relativamente ao mecanismo proposto, segundo os autores, este é composto em: caixa redutora, motor, *encoder*, interruptor de fim de curso e componentes acessórios.

A caixa redutora escolhida neste exemplo é uma harmónica, visto que permite obter razões de redução elevadas de forma compacta, e sem complicações do ponto de vista de montagem. O motor, que se encontra directamente ligado à caixa redutora, é do tipo DC (*Direct Current*) *brushless*. E juntamente com o motor e com a caixa redutora encontra-se um *encoder* ótico incremental.

Nesta arquitectura existe um computador central que pode comunicar e enviar comandos a um conjunto de articulações modulares, e cada uma delas possui um DSP (*Digital Signal Processor*) que realiza o controlo em velocidade e posição. Consequentemente, os autores seleccionaram o protocolo CAN (*Controller Area Network*) como meio de comunicação a implementar entre as articulações e o computador que envia os comandos [1].

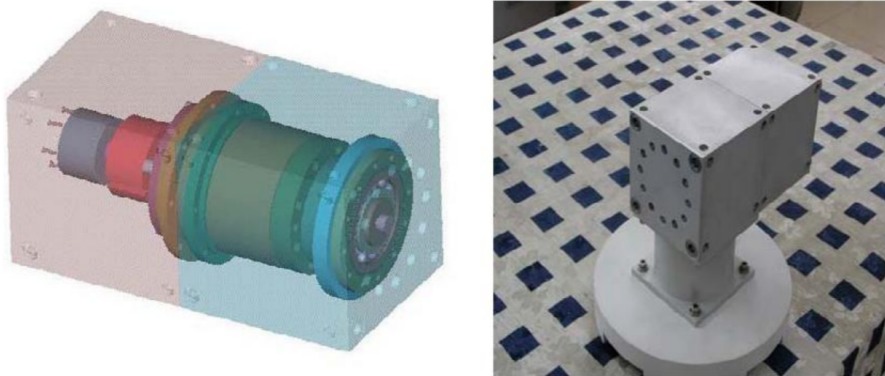


Figura 2.1: Modelo da articulação à esquerda e articulação concluída à direita [1].

2.1.2 Articulação Modular para Robô *dual-arm*

A segunda proposta de solução apresentada originou do objectivo de construir um robô com 2 braços. Sendo assim, como ponto de partida inicial, foi desenvolvida uma articulação modular que permitiu, conjugando um conjunto de articulações, obter o robô completo. Esta articulação e o respectivo modelo encontram-se demonstrados na Figura 2.2. Durante o desenvolvimento, os autores pretendiam que a articulação tivesse as seguintes características: baixo peso, elevada precisão e baixo volume.

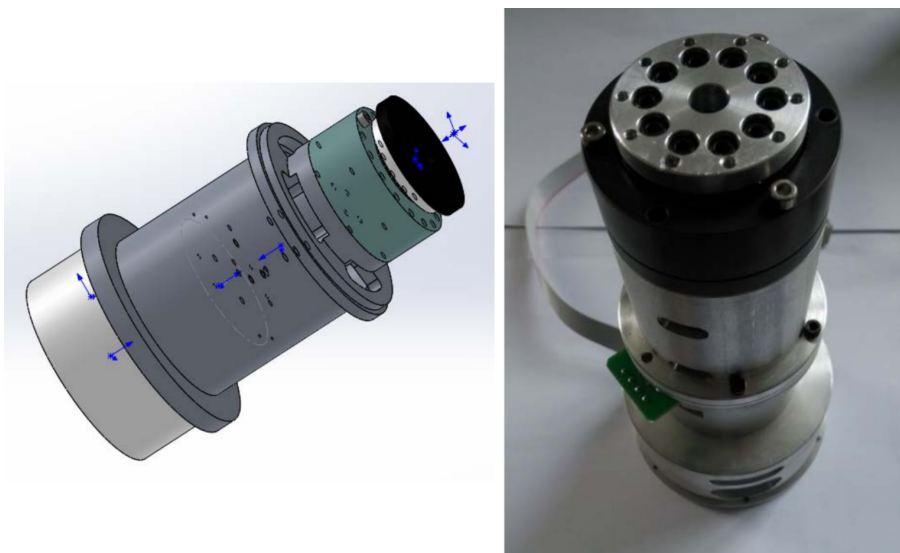


Figura 2.2: Articulação modular à direita e respectivo modelo à esquerda [2].

Ao contrário da proposta apresentada anteriormente, esta solução inclui dois *encoders*, um absoluto e um incremental, que, juntamente com o motor, formam a malha de controlo completa. O uso de duplo *encoder* tem como objectivo aumentar a precisão dos movimentos da articulação.

A componente mecânica desta articulação consiste numa caixa redutora do tipo harmónica. Esta possui também um travão que auxilia a permanência da articulação numa posição fixa.

O motor usado é um *brushless* DC, e teve os seguintes critérios de selecção: ter um tamanho que permita estar em conformidade com a estrutura da articulação, e ter um binário máximo igual a 1,5 vezes o binário máximo que os rolamentos usados suportam.

Como é pretendido que o robô final tenha uma arquitectura distribuída, as articulações possuem comunicação com o protocolo CAN [2].

2.1.3 Articulação Modular com Redução de Velocidade Variável

Por último, esta solução, que se encontra representada na Figura 2.3, consiste na construção de uma articulação modular para braços robóticos com relação de redução de velocidade variável. Ou seja, é possível mudar manualmente a relação da caixa redutora, de forma a ajustar a velocidade e binário conforme o uso do manipulador.

O projecto deste sistema tem os seguintes requisitos: o utilizador deve conseguir mudar a relação de redução facilmente; durante essa mudança a articulação deve continuar alimentada com energia e a transmitir os sinais dos sensores; a articulação deve ter ligações físicas que permitam que seja ligada verticalmente e horizontalmente a outras articulações; e o controlador deve reconhecer automaticamente a redução colocada.

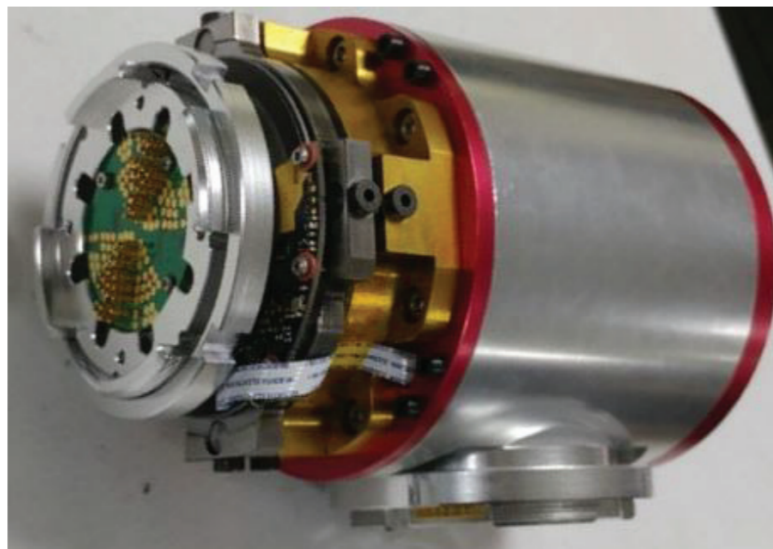


Figura 2.3: Articulação com relação de redução variável [3].

A parte mecânica desta solução é dividida em 2 partes principais: uma caixa redutora de relação fixa e uma de relação variável. A fixa é planetária não *back-drivable*, ou seja, é fisicamente impossível o veio de saída induzir rotação no veio de entrada, o que leva a que seja desnecessário o uso de travões. A de relação variável encontra-se ligada directamente ao veio de saída da planetária, e é composta por um conjunto de rodas dentadas que permite obter a relação pretendida.

Na Figura 2.4, a caixa redutora planetária está identificada como *Main Reducer*, a caixa redutora de relação variável como *Configurable Reducer*, e a interação entre estas está representada a amarelo.

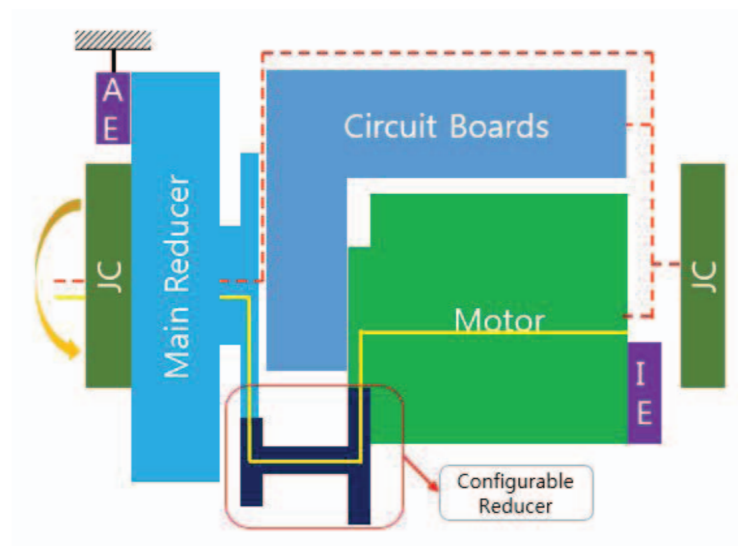


Figura 2.4: Esquema construtivo da articulação com relação de redução variável [3].

O motor, tal como nas propostas anteriores, é um *brushless* DC, ao qual se encontra acoplado um *encoder* incremental. Adicionalmente, no veio de saída da articulação existe um *encoder* absoluto. Na Figura 2.4, IE representa o *encoder* incremental e AE o *encoder* absoluto.

Neste caso, o protocolo de comunicação seleccionado pelos autores é EtherCAT [3].

2.2 Mercado

A modularidade na robótica, além de objecto de investigação tal como visto na secção anterior, também se encontra cada vez mais disponível no mercado. Como tal, nesta presente secção, serão exploradas algumas soluções já disponíveis comercialmente.

2.2.1 HEBI Robotics

A HEBI Robotics dispõe comercialmente de articulações robóticas modulares para braços robóticos articulados.

Esta empresa disponibiliza duas séries: a série R e a série X.

A série R destaca-se principalmente da X devido ao grau de protecção IP67 (*Ingress Protection*). Os actuadores desta série são compostos por uma caixa redutora, um motor *brushless*, sensores de força, *encoders* absolutos magnéticos, controladores que permitem alimentar com uma tensão de 24 V a 48 V, e comunicação por Ethernet *standard*. Adicionalmente, também inclui sensores de temperatura, tensão, corrente, acelerómetro e giroscópio.

Esta série contém uma gama com três opções com diferentes especificações. A de menor potência apresenta um binário de pico de 7 Nm, um binário nominal de 3 Nm, uma velocidade máxima igual a 84 RPM (Rotações Por Minuto), e um peso igual a 670 g. A opção de maior potência, que representa também a mais dispendiosa e se encontra representada na Figura 2.5, tem um binário de pico igual a 38 Nm, um nominal igual a 16 Nm, uma velocidade máxima de 15 RPM, e um peso igual a 715 g [16].



Figura 2.5: Articulação modular Hebi Robotics R8-16 [4].

A série X tem a mesma composição que a R, no entanto, além de não ter grau de protecção IP67, disponibiliza opções mais acessíveis e com menores binários, visto que se destina principalmente ao meio académico e de investigação [17].

2.2.2 Rozum Robotics

A Rozum Robotics também disponibiliza articulações compactas denominadas RDrive, que incorporam num único elemento os seguintes componentes: dois *encoders* absolutos magnéticos, um controlador que é alimentado a 48 V DC, uma caixa redutora harmónica e um motor *brushless* AC.

A gama desta empresa tem 5 opções. A de menor potência, que está representada na Figura 2.6, apresenta um binário de pico de 28 Nm, um nominal de 11 Nm, e uma velocidade máxima igual a 55 RPM. Enquanto a opção de maior potência tem um binário de pico igual a 333 Nm, um nominal igual a 216 Nm, e uma velocidade máxima igual a 20 RPM.

Ao contrário das opções apresentadas na Sub-secção 2.2.1, estas apresentam uma grande discrepância entre pesos: 0,9 kg e 3,9 kg, para a de menor e maior potência respectivamente.



Figura 2.6: Articulação modular Rozum RDrive 50 [5].

Estas articulações têm todas grau de protecção IP21 e suportam comunicação CANopen [5].

Capítulo 3

Fundamentação Teórica

No capítulo que se segue é realizada uma fundamentação teórica relativamente às articulações modulares. Inicialmente são analisadas diferentes configurações para as caixas redutoras, são apresentados os diferentes motores e sensores usados nestas aplicações, e são também estudadas as diferentes abordagens para controlo de articulações rotativas. Depois é analisado o método de parametrização de motores, e são apresentados os critérios de selecção de microcontroladores para estas aplicações. Por último, são mencionadas as diferentes tecnologias associadas à produção de caixas redutoras.

3.1 Sistemas de Actuação para Articulações Rotativas

Os sistemas de actuação para articulações rotativas consistem em caixas redutoras que, para a mesma potência, permitem reduzir a velocidade de rotação e aumentar o binário produzido por um motor. Mais concretamente, têm um veio de entrada e um veio de saída que estão relacionados mecanicamente por uma dada razão numérica. Esta razão, segundo a qual se reduz a velocidade e se aumenta o binário, é dada pela relação entre o número de rotações do veio de entrada e o número de rotações do veio de saída.

Tipicamente estas caixas redutoras apresentam uma das seguintes configurações: de parafuso, cicloidal, harmónica, planetária ou planetária composta.

3.1.1 Caixa Redutora de Parafuso Sem-fim

A caixa redutora de parafuso sem-fim consiste numa arquitetura composta por 2 engrenagens, sendo uma delas um parafuso e a outra uma roda dentada. Os veios destas 2 engrenagens encontram-se perpendiculares, e os dentes do parafuso engrenam nos dentes da roda dentada. O veio de entrada destas caixas redutoras encontra-se directamente ligado fisicamente ao parafuso, e a roda dentada tem o seu veio ligado com o veio de saída.

Na Figura 3.1, o parafuso está identificado como elemento 1, a roda dentada como elemento 2 e a caixa externa que suporta as engrenagens como elemento 3.

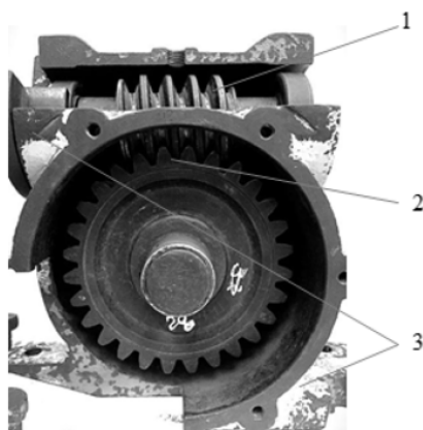


Figura 3.1: Caixa redutora de parafuso sem-fim [6].

Estas caixas redutoras apresentam as vantagens de serem silenciosas, suaves e absorverem mais facilmente vibrações. Também não são reversíveis, isto é, o veio de saída não consegue induzir rotação no veio de entrada devido ao elevado atrito entre os dentes das engrenagens, o que pode ser vantajoso para aguentar cargas numa dada posição sem esforço do motor.

No entanto também apresentam a desvantagem de terem perdas energéticas muito elevadas quando comparadas às restantes configurações, daí serem usadas em aplicações que tenham pouca relevância do ponto de vista da eficiência energética [6].

3.1.2 Caixa Redutora Cicloidal

A caixa redutora cicloidal é composta por 5 partes principais: a engrenagem cicloidal, o veio de entrada com excêntrico, a caixa, os pinos e o veio de saída. Sendo estes os elementos 1, 2, 3, 4 e 5 na Figura 3.2, respectivamente.

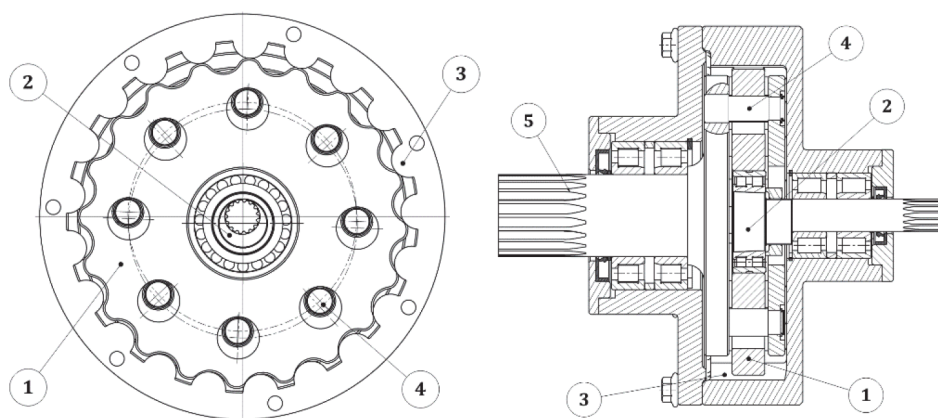


Figura 3.2: Vista 1D e seccionada de uma caixa redutora cicloidal [7].

A rotação do veio de entrada induz, através do seu excêntrico, rotação na engrenagem cicloidal. Consequentemente, a engrenagem roda e os seus dentes engrenam gradualmente nos dentes da caixa. Com isto, a engrenagem movimenta-se relativamente à caixa, e os pinos transferem o seu deslocamento angular para o veio de saída.

Estas caixas redutoras têm a vantagem de permitirem obter grandes reduções mantendo dimensões compactas. Também conseguem ter *backlash* praticamente nulo e aguentam elevados binários. O *backlash* é caracterizado pela folga existente no veio de saída com o veio de entrada completamente estático.

No entanto, dada a elevada complexidade da cinemática das engrenagens, apresentam a desvantagem de induzir *ripple* na velocidade e no binário de saída. Isto é, mantendo a velocidade de rotação do motor constante, o binário e a velocidade de saída oscilam durante uma rotação completa do veio de saída [7].

3.1.3 Caixa Redutora Harmónica

A caixa redutora harmónica tem como componentes principais: uma engrenagem dentada interna fixa, uma engrenagem flexível e um gerador de ondas que deforma radialmente a engrenagem flexível. Estes componentes são exemplificados na Figura 3.3, e a respectiva montagem também.

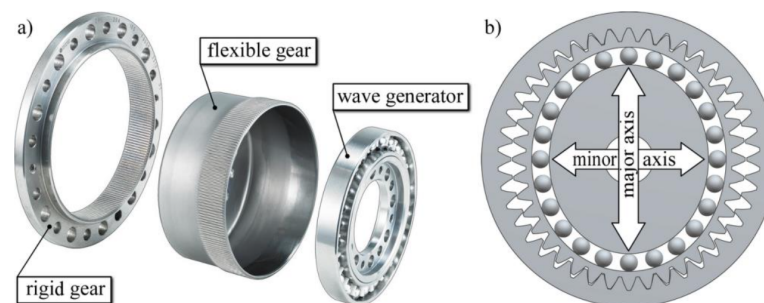


Figura 3.3: Caixa redutora harmónica: (a) componentes principais; (b) interação entre os componentes [8].

O motor liga-se ao gerador de ondas, e a cada volta desse gerador a engrenagem flexível tem uma rotação equivalente à diferença entre o número de dentes da engrenagem flexível e o número de dentes da engrenagem interna fixa.

As caixas redutoras harmónicas destacam-se pelo *backlash* praticamente nulo que têm, pela sua eficiência elevada a baixas velocidades de rotação e pela elevada redução de velocidade que permitem.

No entanto, estas também estão limitadas a velocidades máximas muito baixas quando comparadas com outras configurações. E mesmo tendo eficiência elevada, esta reduz drasticamente conforme o aumento da velocidade [18].

3.1.4 Caixa Redutora Planetária

A caixa redutora planetária é composta principalmente pelos seguintes componentes: engrenagem sol, engrenagens planeta, engrenagem anel e suporte de planetas.

Esta arquitectura permite obter 12 relações de redução diferentes para um único conjunto de engrenagens, simplesmente alternando o componente no qual se aplica a velocidade de rotação [9].

Na Figura 3.4 é observável um conjunto de engrenagens que compõem uma caixa redutora planetária: sendo a engrenagem ao centro o sol, a engrenagem mais externa o anel, e todas as engrenagens entre o sol e o anel são os planetas.

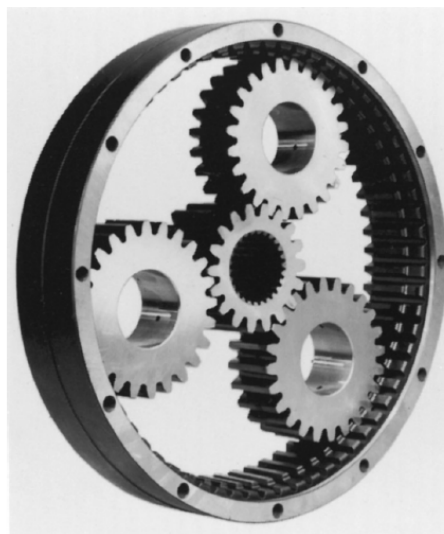


Figura 3.4: Composição de uma caixa redutora planetária [9].

Estas caixas redutoras são vantajosas visto que permitem obter reduções elevadas numa configuração bastante compacta e, ao contrário das restantes configurações, são as que possibilitam maiores velocidades de rotação, tanto de entrada como de saída [19]. Também aguentam elevados binários e mantêm elevada eficiência energética.

Em contrapartida, apresentam a desvantagem de ser a arquitectura que dificulta mais a obtenção de *backlash* totalmente nulo [18].

3.1.5 Caixa Redutora Planetária Composta

As caixas redutoras planetárias apresentam uma variância, que consiste em conjugar várias compondo-as de tal forma que exista apenas um veio de entrada e um de saída, e se multiplique as razões de redução de cada uma delas.

Esta configuração é a mais usada quando se pretende multiplicar o binário segundo uma razão elevada enquanto se mantém uma caixa redutora compacta. E graças à versatilidade que as engrenagens planetárias apresentam relativamente à sua configuração, pode-se obter razões até 1000:1 [18].

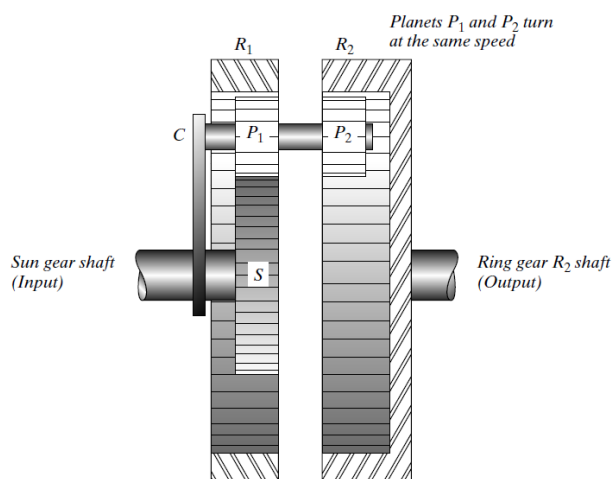


Figura 3.5: Caixa redutora planetária composta com planetas interligados [9].

Na configuração apresentada na Figura 3.5, os planetas de ambas as reduções planetárias rodam sincronamente; só existe um sol ao qual é ligado o motor; e a velocidade de rotação de saída é obtida a partir da engrenagem anel da segunda caixa redutora.

3.2 Tipos de Motores

Os sistemas de actuação para articulações rotativas recorrem a motores eléctricos para transformarem a energia eléctrica em energia mecânica/cinética.

Tendo em conta a variedade de motores eléctricos existentes, serão de seguida apresentados os que são mais frequentemente usados em robótica. Mais concretamente: motores de passo, motores sem escovas e motores de ímanes permanentes com escovas, todos rotativos e com alimentação DC.

Durante o estudo de motores serão tidos em conta fatores como: constituição do motor, vantagens e desvantagens e características de binário e velocidade.

3.2.1 Motor DC com Escovas de Ímanes Permanentes

Os motores DC com escovas de ímanes permanentes, tal como o nome sugere, apresentam esta designação devido ao facto de possuírem ímanes permanentes e de recorrerem a escovas para excitar o rotor.

Estes são os mais usados em produtos comercialmente disponíveis, nos quais se pretende um bom desempenho e um baixo custo, como por exemplo, aparelhos domésticos.

Relativamente à sua constituição, estes são compostos por: um estator com ímanes permanentes; um rotor com pólos, e cada um dos pólos tem enrolamentos em torno de si; um conjunto de escovas; e um comutador, que se encontra ligado aos pólos do rotor, e onde as escovas estão em contacto para excitar o movimento.

Na Figura 3.6 é observável a sua constituição típica, e a respectiva comparação à arquitectura sem escovas.

Relativamente ao seu funcionamento, estes são alimentados através do comutador e das escovas. As escovas estando em contacto com o comutador estabelecem um dado campo magnético no rotor que, conforme a sua direcção, faz com que o rotor rode. E, consequentemente, a rotação do rotor faz com que se varie continuamente a direcção do campo, visto que esta é estabelecida pelo contacto entre as escovas e as diferentes secções do comutador.

Nestes motores, a tensão é comutada mecanicamente, o que pode revelar um potencial problema devido a desgaste acrescido das escovas e do comutador [20].

3.2.2 Motor DC sem Escovas

Os motores de alimentação DC sem escovas, ou *brushless*, têm esta denominação pelo facto de não recorrerem ao uso de escovas para estimular a rotação do motor.

Os motores *brushless* são extremamente usados em aplicações onde não é possível recorrer a elementos de transmissão, também conhecidas como aplicações *direct-drive*, devido à sua capacidade de produzir elevado binário.

Estes motores são constituídos por um rotor de ímanes permanentes e por um estator com pólos, tendo cada pólo os enrolamentos correspondentes [21], tal como é verificável no elemento à direita na Figura 3.6.

Como os ímanes estão no rotor, ao contrário do motor anterior, é desnecessário o uso de escovas, o que leva a que este motor tenha mais durabilidade visto que não há desgaste mecânico. Os motores *brushless* também têm as vantagens de dissipar melhor temperatura e operar mais facilmente a elevadas velocidades que os escovados.

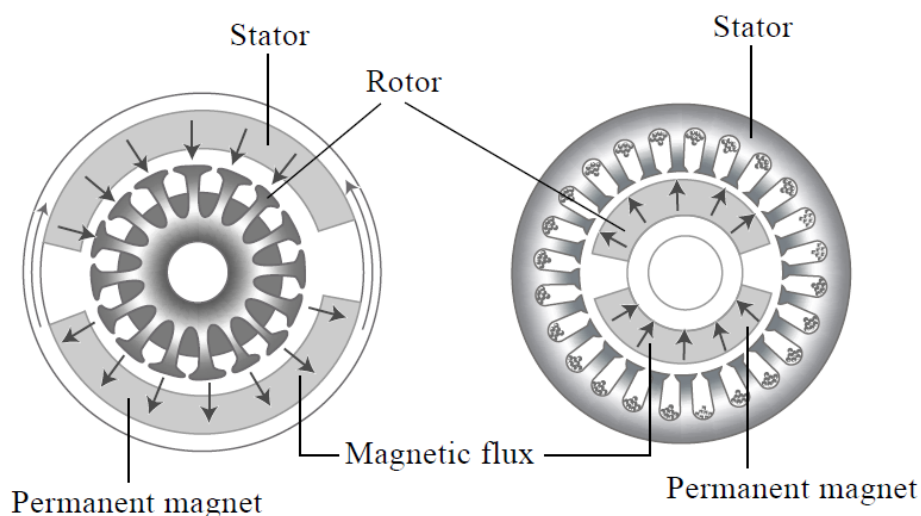


Figura 3.6: Construção física do motor DC: à esquerda com escovas; à direita *brushless* [10].

No entanto, também apresentam a desvantagem de serem mais complicados de modelar e de controlar quando comparados com outros motores [22].

A alimentação do motor é feita através dos enrolamentos presentes no estator através de um controlador. O controlador excita os enrolamentos de tal forma que se crie um campo magnético rotativo no estator. Consequentemente, o rotor roda de forma a alinhar o seu campo magnético com o do estator.

3.2.3 Motor de Passo

Um motor de passo, também comumente chamado *stepper*, é um motor de corrente contínua sem escovas que secciona uma rotação completa num dado número de passos iguais. Com estes passos consegue-se controlar a posição e velocidade do motor [22].

Estes são extremamente usados em aplicações que envolvam medições precisas, como impressoras 3D, máquinas CNC (Controlo Numérico Computorizado) e braços robóticos de baixa potência.

Os *steppers* têm a vantagem de conseguirem ser controlados em malha aberta e malha fechada, e conseguirem manter a posição do veio constante, o que leva a que o controlo destes seja acessível e se obtenha um bom desempenho em controlo de velocidade e posição [20].

Apesar de se conseguir controlar a velocidade facilmente, estes apresentam a desvantagem do seu binário decrescer com o aumento da velocidade. À velocidade máxima, o binário é praticamente nulo [23].

Tal como mencionado, o veio do motor é controlado em passos. E quanto maior for o número de passos que o motor precisa para dar uma rotação completa, maior é a sua resolução.

Os *steppers* podem ser divididos em 3 tipos de motores diferentes conforme a construção do rotor: ímanes permanentes, relutância variável e híbridos.

Os de ímanes permanentes têm um rotor composto por ímanes permanentes e um estator composto por pólos magnéticos salientes com enrolamentos em torno de si, a partir dos quais se alimenta o motor. Na Figura 3.7, podemos observar uma construção típica de um motor de passo de ímanes permanentes com o rotor magnético e os pares de pólos (A, B, C) à sua volta. A alimentação é feita a partir de pulsos que geram um campo magnético no estator. Consequentemente, o princípio de funcionamento consiste na rotação do rotor de forma a alinhar o seu campo magnético com o do estator. Por último, como o rotor é composto por um íman permanente, o motor consegue manter um dado binário mesmo quando os enrolamentos do estator deixam de estar excitados, este binário é denominado de binário remanescente [24].

Os de relutância variável têm um rotor metálico não magnético indentedo, e têm um estator igual à arquitectura anterior, tal como pode ser observado na Figura 3.8. Consequentemente, o princípio de funcionamento consiste no alinhamento dos dentes do rotor com os pólos do estator, de forma a minimizar o percurso percorrido pelo fluxo magnético, e alinhar os campos magnéticos do estator e do rotor [23]. É chamado de relutância variável porque a relutância do circuito magnético formado pelo rotor e pelo estator varia com a posição angular do veio.

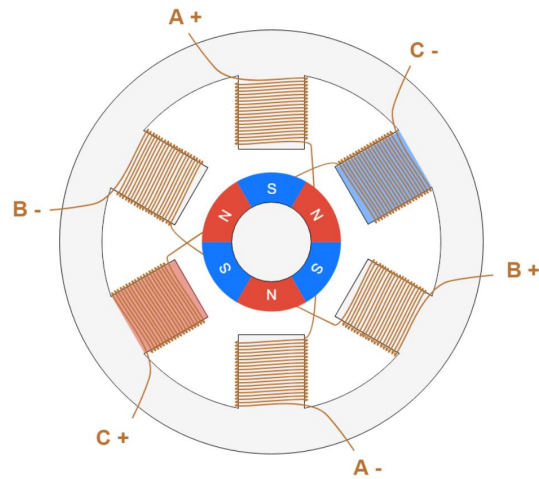


Figura 3.7: Construção de motor de passo de ímãs permanentes [11].

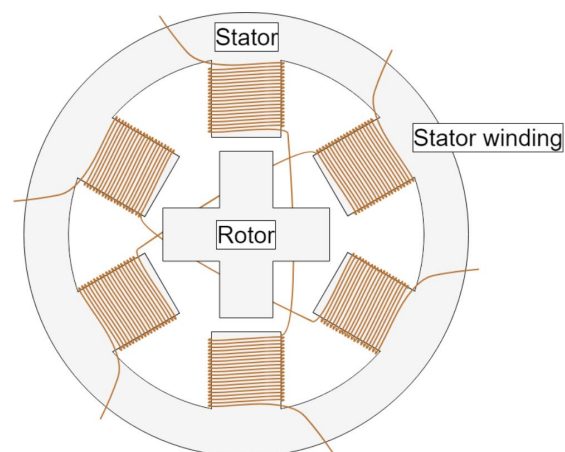


Figura 3.8: Construção de motor de passo de relutância variável [11].

Os motores de passo híbridos, tal como o nome sugere, têm características provenientes tanto dos de ímanes permanentes como dos de relutância variável. Mais concretamente, o seu rotor é constituído por um íman permanente, tal como o motor de passo de ímanes permanentes, e é também indentado, como o de relutância variável [22].

3.3 Sensores

Os sensores são dispositivos que permitem obter uma percepção do estado interno e/ou externo de um sistema. Tendo em conta a aplicação, são sensores que nos permitem determinar a posição e/ou velocidade da articulação. Estes dispositivos são comumente denominados *encoders*.

Estes podem ter uma arquitectura linear ou rotativa. No entanto como neste caso a articulação a desenvolver é do tipo rotativa, ou seja não é linear, abordar-se-ão apenas os sensores rotativos.

Os *encoders* podem ser distinguidos entre sensores absolutos, que produzem um sinal dependente da posição actual do sensor, e sensores incrementais, que produzem um sinal dependente da variação da posição.

Como tal, ao longo desta secção serão abordados os sensores rotativos mais frequentes nas articulações robóticas rotativas.

3.3.1 Sensor Potenciométrico

Um sensor potenciométrico consiste no uso de um potenciómetro para determinação da posição angular. O potenciómetro é uma resistência variável com três terminais, em que entre os 2 terminais mais extremos se encontra uma resistência fixa e o terminal do meio está ligado a um contacto móvel. Na Figura 3.9 os terminais extremos são o A e o B, e o terminal central é representado pela letra C.

A variação da posição do contacto móvel, que ocorre num dado eixo de rotação, faz com que se varie a resistência entre esse terminal e um dos outros terminais. Consequentemente, obtém-se a posição angular do potenciómetro partindo do princípio do divisor de tensão resistivo que ocorre com a variação da posição. Como tal, medindo a tensão no terminal central obtém-se a posição do contacto móvel.

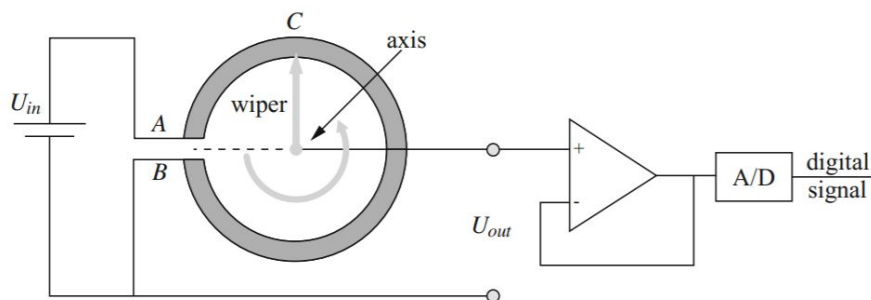


Figura 3.9: Princípio de funcionamento de um sensor potenciométrico [12].

As principais vantagens de um sensor potenciométrico são: baixo custo, tamanho compacto e versatilidade no uso. No entanto, também apresentam desvantagens: requerem o uso de *hardware* adicional, como um conversor analógico-digital; e estão sujeitos a desgaste mecânico e a ruído [13].

Por último, como o sensor potenciométrico permite obter a posição absoluta, calculando o tempo que ocorre entre amostras sucessivas é também possível calcular a velocidade.

3.3.2 Sensor Ótico

Um dos tipos de *encoders* mais usados na indústria são os óticos. Estes sensores transformam o movimento da articulação num conjunto de impulsos eléctricos.

Como tal, usa-se uma fonte de luz para transformar a energia eléctrica em luz, e um fototransístor ou um fotodíodo para transformar a luz num sinal eléctrico.

Desta forma, o veio do sensor encontra-se fisicamente ligado a um disco com ranhuras que, conforme a sua rotação, permite intermitentemente que a luz o atravesse ou que seja bloqueada.

O princípio de funcionamento de um *encoder* ótico pode ser observado na Figura 3.10. No disco apresentado nesta imagem podemos ver representadas as devidas ranhuras, o fototransístor e um LED (*Light-emitting Diode*) como fonte emissora de luz.

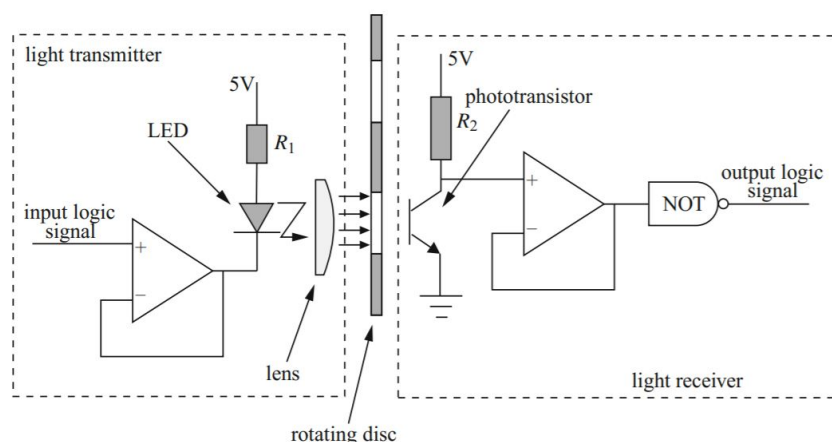


Figura 3.10: Princípio de funcionamento de um sensor ótico [12].

Desta forma, a saída lógica do sensor tem o valor máximo quando a luz atravessa o disco e atinge o fototransístor. Quando o disco bloqueia a passagem de luz, a saída lógica apresenta o valor mínimo [12].

Estes sensores apresentam a vantagem de terem elevada resolução e não sofrerem desgaste mecânico. No entanto a resolução também está limitada às características construtivas do mesmo, isto é, a quantidade de ranhuras no disco [13].

Os sensores óticos podem ser divididos em absolutos e incrementais.

3.3.2.1 Sensor Ótico Incremental

Os sensores óticos incrementais usam as características mencionadas na secção anterior para conseguirem fornecer informação relativamente à variação da posição angular. Ou seja, a rotação do disco gera um conjunto de impulsos elétricos a partir dos quais podemos calcular o deslocamento angular do *encoder*, sendo o número de impulsos proporcionais ao deslocamento.

Um sensor ótico incremental com apenas uma única fonte emissora de luz e um único fototransístor consegue descrever o deslocamento angular mas não consegue descrever em que sentido ocorreu. Como tal, para determinar o sentido de rotação adiciona-se mais um LED e mais um fototransístor tangencialmente ao par já existente, como podemos observar na Figura 3.11. Nesta figura, cada par de elementos LED-fototransístor estão representados pelas letras A e B.

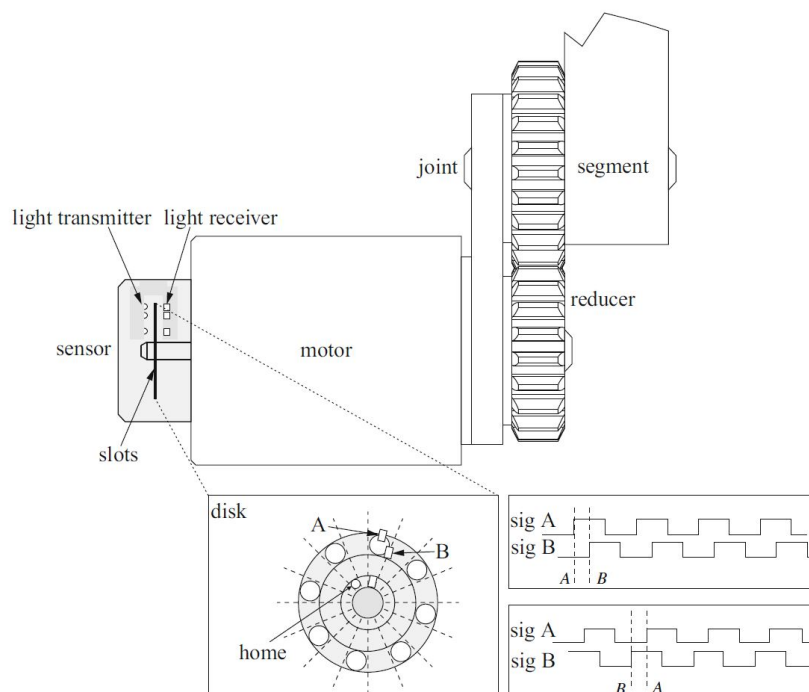


Figura 3.11: Modelo de um *encoder* ótico incremental e respetivos sinais gerados [12].

Desta forma, observando na Figura 3.11 os sinais gerados pelos pares LED-fototransístor A e B, podemos identificar o sentido de rotação detectado pelo sensor. Mais concretamente, ocorre um desfaseamento entre os sinais A e B sendo que, quando A se encontra em avanço relativamente ao B a rotação ocorre no sentido positivo, e vice-versa quando a rotação ocorre no sentido negativo.

Tendo em conta o período dos impulsos gerados, o desfaseamento entre os sinais A e B tipicamente é de $\pi/2$ [12].

Adicionalmente, a resolução deste tipo de sensores é igual à calculada na Equação 3.1. Nesta equação n representa o número de ranhuras existentes no disco do *encoder*:

$$\Delta\alpha = \frac{2\pi}{n} \quad (3.1)$$

Por último, os incrementais apresentam a vantagem de serem simples, compactos e de terem um custo baixo quando comparados aos absolutos. No entanto, como estes apenas contabilizam a variação angular, é necessário obter-se um ponto de referência inicial e contabilizar constantemente a variação de forma a ter conhecimento da sua posição [13]. Para se obter a velocidade, basta ter em conta o intervalo de tempo associado a cada amostra de variação angular.

3.3.2.2 Sensor Ótico Absoluto

Os sensores óticos absolutos trabalham segundo o mesmo princípio dos incrementais, isto é, também usam a interrupção e/ou ininterrupção da passagem de luz através de um disco de forma a codificar a sua posição. No entanto, enquanto os discos dos incrementais têm apenas um canal com ranhuras espaçadas igualmente entre si, os discos dos absolutos têm vários canais com ranhuras espaçadas de forma a formar um código binário que descreve a sua posição. Consequentemente, o valor de saída do sensor ótico absoluto é um valor que descreve directamente a posição absoluta do disco.

Desta forma, na Figura 3.12 podemos observar um exemplo de um disco usado num sensor ótico absoluto com 4 canais de ranhuras, ou seja de 4 bits. Tendo 4 bits consegue descrever $2^4 = 16$ posições diferentes.

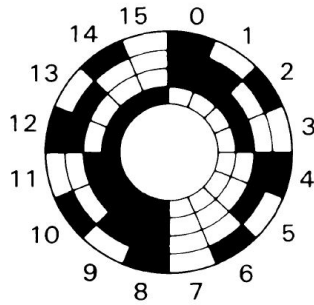


Figura 3.12: Modelo de um *encoder* ótico absoluto de 4 bits [13].

A resolução deste tipo de sensores é igual à calculada na Equação 3.2. Sendo que nesta equação, c representa o número de canais diferentes existentes no disco do *encoder*, que é igual ao número de bits que descreve a respetiva posição angular [13].

$$\Delta\alpha = \frac{2\pi}{2^c} \quad (3.2)$$

Os sensores óticos absolutos são ideais quando se pretende saber automaticamente a posição sem ter que deslocar a articulação até uma posição de referência, visto que este tipo de sensores

permite saber a posição a qualquer momento. Também permitem determinar o sentido de rotação e calcular a velocidade respectiva, tendo em conta o intervalo de tempo e o deslocamento angular que ocorre entre amostras. No entanto têm a desvantagem de terem um custo bastante elevado quando comparados a outros sensores [12].

3.3.3 Sensor Magnético

Os sensores magnéticos servem-se da medida da intensidade do campo magnético para medir posição ou variação da posição (velocidade). Esta variância também apresenta tanto a arquitectura incremental como a arquitectura absoluta. Ou seja, na primeira apenas se detecta variação da posição, e na segunda detecta-se variação e posição absoluta.

Os sensores magnéticos têm a vantagem de permitirem obter boa precisão e resistência a ambientes com condições agrestes a nível de pressão e temperatura. Os baseados em efeito de Hall mais especificamente, são característicos por conseguirem detectar mudanças mínimas no campo magnético, terem um baixo consumo energético e terem uma geometria relativamente pequena [25].

No entanto também apresentam a desvantagem de introduzir elevados erros caso a distância entre o sensor e o íman varie durante a rotação, o que pode ocorrer devido à existência de rolamentos defeituosos/desgastados ou montagens erróneas do sistema [26].

3.3.3.1 Sensor Magnético Incremental

Enquanto nos sensores óticos o disco do sensor era definido pelo bloqueio intermitente da passagem de luz, nos magnéticos o disco é composto por vários pólos magnéticos alternados em torno do disco, tal como se pode verificar no exemplo da Figura 3.13. Nesta figura, podemos verificar a existência de um íman em que os pólos alternam de forma uniforme ao longo da sua circunferência.

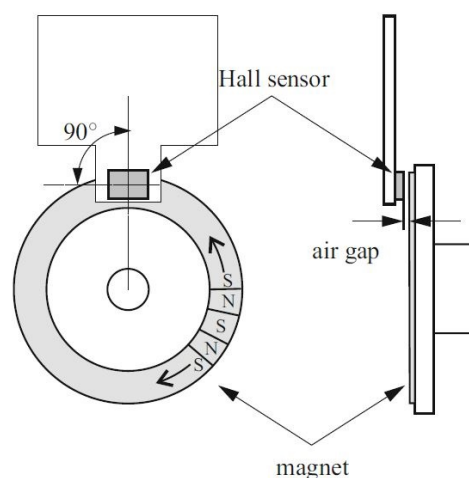


Figura 3.13: Sensor de Hall e pólos magnéticos alternados [12].

O tipo de sensor usado nesta arquitectura pode ser um sensor de Hall ou Magneto-resistivo. Os de Hall geram uma tensão proporcional ao campo magnético e os Magneto-resistivos variam a sua resistência conforme a intensidade [12].

Tal como os incrementais óticos, os incrementais magnéticos também contabilizam apenas a variação angular, o que implica que seja necessário ter um ponto de referência, e observar constantemente a variação de forma a manter o conhecimento da posição.

3.3.3.2 Sensor Magnético Absoluto

Apesar de ser recorrente o uso de sensores de Hall ou Magneto-resistivos na variante incremental, na variante absoluta tradicionalmente recorre-se apenas aos de Hall.

Nos sensores magnéticos absolutos usa-se um íman magnetizado diametralmente fixado ao eixo do qual pretendemos saber a posição angular, e um mínimo de 4 sensores de Hall para que se consiga ter conhecimento absoluto da posição do íman e respectivo sentido de rotação. Caso se usem quatro sensores, estes devem estar desfasados 90° uns dos outros; e caso se pretenda usar mais, deve-se usar um múltiplo de quatro visto que cada quadrante deve ter o mesmo número de sensores [26].

Na Figura 3.14 podemos observar um exemplo com 4 sensores de Hall igualmente espaçados entre si relativamente ao centro do íman. Desta forma, produzem sinais sinusoidais também desfasados 90° entre si e proporcionais ao respectivo campo magnético detectado [12].

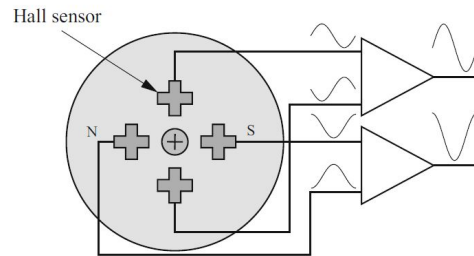


Figura 3.14: Múltiplos sensores de Hall e um íman magnetizado diametralmente [12].

Tendo quatro sensores, cada um deles gera um dado sinal sinusoidal S1, S2, S3 e S4 respectivamente. Desta forma, a posição angular do íman α relativamente ao centro dos sensores é calculado recorrendo às Equações 3.3, 3.4 e 3.5 [25].

$$S_{sine} = S1 + S2 - S3 - S4 \quad (3.3)$$

$$S_{cos} = S1 - S2 - S3 + S4 \quad (3.4)$$

$$\alpha = \arctang\left(\frac{S_{sine}}{S_{cos}}\right) \quad (3.5)$$

Tal como os absolutos óticos, os absolutos magnéticos permitem saber directamente a posição do sensor a qualquer momento, o que permite calcular também a velocidade tendo em conta o deslocamento angular e o período entre amostras.

3.4 Metodologias de Controlo de Articulações

O controlo de articulações pode ser feito em posição e/ou velocidade. Para tal recorre-se a estratégias de controlo em malha fechada, visto que estas permitem obter um erro em regime permanente praticamente nulo, uma baixa sensibilidade a variações de carga, e tempos de resposta rápidos.

O controlo servo pode ser dividido em dois tipos de problemas: seguimento da referência e rejeição de perturbações. Desta forma, os controladores Proporcionais, Integrais e Derivativos (PID), e Proporcionais, Integrais e de Velocidade (PIV), são as variantes mais comuns que conseguem solucionar os dois tipos de problemas [27].

Como tal, nas Sub-secções 3.4.1 e 3.4.2, serão abordados os Controladores PID e PIV respectivamente.

Adicionalmente, na Sub-secção 3.4.3 serão abordadas diferentes arquitecturas possíveis para os controladores, e na Sub-secção 3.4.4 serão abordados os observadores de estado e respectiva realimentação.

3.4.1 Controlador PID

O controlador PID, tal como as siglas indicam, possui um componente proporcional, um componente integral e um componente derivativo, tal como podemos observar na Figura 3.15. Esta figura representa uma arquitectura típica de uma malha de controlo com um PID realimentado pela posição do motor.

Como o binário produzido por um motor de corrente contínua é praticamente proporcional à corrente consumida (Equação 3.6), na arquitetura da Figura 3.15 o controlador PID calcula a corrente que o *driver* deve aplicar no motor.

$$T \approx K_t I \quad (3.6)$$

Assim, o *driver* aplica uma dada corrente no motor que induz o movimento e consequentemente varia a posição. A posição e/ou velocidade podem ser determinadas pelo uso de *encoders*.

No exemplo da Figura 3.15 o controlador actua conforme o erro em posição dado pela equação 3.7, sendo θ_r a posição angular de referência e θ a posição angular actual do motor.

$$e(t) = \theta_r(t) - \theta(t) \quad (3.7)$$

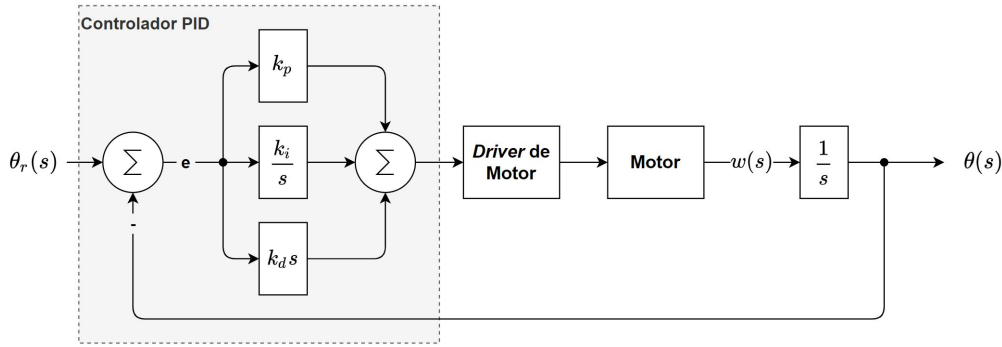


Figura 3.15: Arquitectura típica de um controlador PID.

O valor de saída do controlador PID, dado pela Equação 3.8 no domínio temporal, depende do erro e dos ganhos associados K_p , K_i e K_d .

$$PID_{output} = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t) \quad (3.8)$$

A determinação dos ganhos K_p , K_i e K_d , pode ser realizada por tentativa-erro (método pouco eficiente) ou por vias analíticas [27]. Quando se opta por vias analíticas pode-se recorrer a métodos como o de *Ziegler-Nichols*.

Quando se recorre ao método de *Ziegler-Nichols*, caso se consiga obter um modelo matemático do sistema, é possível determinar os parâmetros do controlador que permitirão que se obtenha o comportamento desejado. No entanto, caso não se consiga obter, não é possível também aplicar um modelo analítico para projectar o controlador. Nesse caso pode-se recorrer a abordagens experimentais [28].

Por último, é possível gerar também controladores Proporcionais e Derivativos (PD), que derivam do PID mas sem a componente integradora. Ou controladores Proporcionais e Integrais (PI), que derivam do PID mas sem a componente derivadora.

3.4.1.1 Módulo *Anti-windup*

Os sistemas, mesmo aqueles que podem ser linearizados, podem conter não linearidades associadas a limitações, como por exemplo, velocidade máxima do motor, tensão máxima que um *driver* suporta, entre outros.

Desta forma, um controlador durante o seu funcionamento pode atingir o valor máximo da variável de controlo. E caso o valor da variável permaneça no valor máximo e o erro continue não nulo, a componente integradora do PID continuará a integrar o erro [29].

O *Anti-windup* consiste em não realizar a integração do erro enquanto a variável de controlo se encontra no seu valor máximo e, consequentemente, evitar a ocorrência de sobre-elongações elevadas e tempos de estabelecimento longos [30].

A componente de *Anti-windup* num sistema de controlo tem três técnicas principais de implementação [31]:

- **Integração Condicional** — Neste método, a componente integradora do controlador apenas ocorre segundo dadas condições. Isto é, a integração apenas ocorre enquanto a variável de controlo se encontra dentro dos limites operacionais, caso os iguale ou ultrapasse a integração do erro é interrompida.
- **Integrador Limitado** — Com a técnica do integrador limitado, a componente integradora do controlador é realimentada com a soma da própria saída associada a um ganho. Este ganho é nulo no intervalo de operação em que o sistema é linear, e é muito elevado quando a variável de controlo ultrapassa os limites da linearidade, desta forma reduz-se o efeito da componente integradora na saída do controlador.
- **Tracking Anti-windup** — Esta técnica é a mais comum, e consiste em subtrair à entrada da componente integradora a diferença entre o valor de saída do sistema saturado e o valor de saída do sistema não saturado. Esta subtracção ocorre apenas quando a variável de controlo ultrapassa os limites máximos, superiores ou inferiores.

3.4.1.2 Módulo *Feedforward*

Os módulos *Feedforward* são frequentemente associados aos controladores PID, tendo como objectivo melhorar a velocidade de resposta do sistema sem afectar a estabilidade do mesmo, através da redução da sobre-elongação e do tempo de subida [27].

Numa malha de controlo, como a da Figura 3.16, em que $G(s)$ é a função de transferência do sistema a controlar e $H(s)$ do controlador, adiciona-se a componente de *Feedforward* representada pela função de transferência $F(s)$.

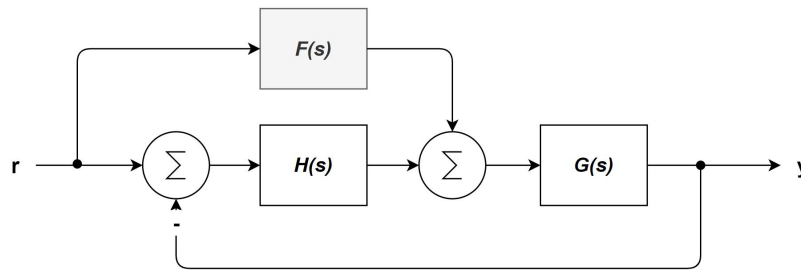


Figura 3.16: Esquema de controlo com componente de *feedforward*.

As funções de transferência de $G(s)$, $H(s)$ e $F(s)$, podem ser individualmente descritas pelas Equações 3.9, 3.10 e 3.11, respetivamente.

$$G(s) = \frac{q(s)}{p(s)} \quad (3.9)$$

$$H(s) = \frac{c(s)}{d(s)} \quad (3.10)$$

$$F(s) = \frac{a(s)}{b(s)} \quad (3.11)$$

O que permite obter uma função de transferência do sistema em malha fechada igual a $T(s) = \frac{Y(s)}{R(s)}$, e que pode ser mais detalhadamente descrita pela Equação 3.12.

$$T(s) = \frac{q(s)(c(s)b(s) + a(s)d(s))}{b(s)(p(s)d(s) + q(s)c(s))} \quad (3.12)$$

Desta forma, definindo a função de transferência da componente de *Feedforward* igual a $F(s) = \frac{1}{G(s)}$ implica-se que $a(s) = p(s)$ e $b(s) = q(s)$, o que leva consequentemente a que $T(s) = 1$. Ou seja, a saída do sistema é igual à entrada, o que é sinónimo de obtenção de erro nulo [32].

3.4.2 Controlador PIV

O controlador PIV é uma alternativa ao PID mais simples de afinar e que permite ter uma melhor previsão da resposta do sistema.

O controlador PIV, que se encontra representado na Figura 3.17, resulta da combinação de uma malha de controlo de velocidade com uma de posição.

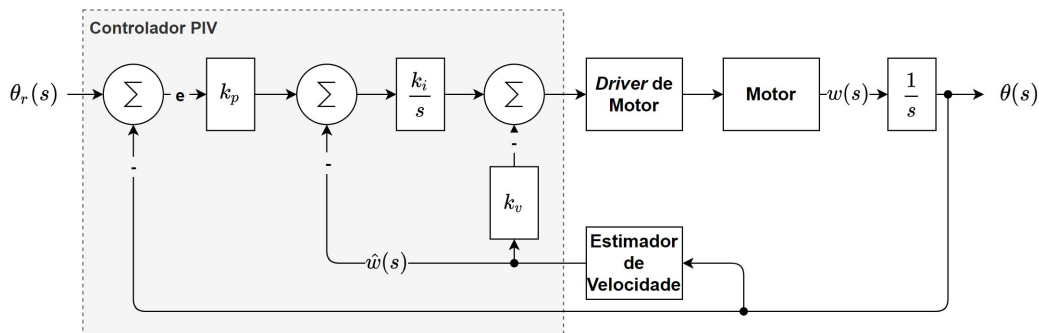


Figura 3.17: Arquitetura típica de um controlador PIV.

Nesta arquitectura o erro em posição é multiplicado pela constante k_p , dando resultado a um comando de correcção da velocidade. Desta forma, a componente integral do controlador $\frac{k_i}{s}$ actua directamente no erro em velocidade, em vez de actuar sobre o erro em posição como na topologia PID da Secção 3.4.1. O ganho derivativo k_d de um PID é substituído pelo ganho k_v , que atua directamente sobre a velocidade estimada.

O controlador PIV requer o conhecimento correto da velocidade do motor/articulação. De forma a estimar a velocidade do motor podem ser usados observadores de estado (Seccção 3.4.4).

Para afinar o controlador PIV é necessário definir apenas 2 parâmetros: a largura de banda BW (*Bandwidth*) e o factor de amortecimento ζ . No entanto também é necessário ter uma estimativa do momento de inércia $\hat{J}$ e da constante de amortecimento $\hat{b}$ da carga e do motor.

Quanto maior for a largura de banda BW , mais rápidos são os tempos de subida e de estabelecimento. Por outro lado, para uma largura de banda fixa, quanto menor for o factor de amortecimento ζ maior é a sobre-elongação.

Os ganhos do controlador PIV são calculados tal como descrito nas Equações 3.13, 3.14 e 3.15 [27].

$$k_p = \frac{2\pi BW}{2\zeta + 1} \quad (3.13)$$

$$k_i = (2\pi BW)^2(1 + 2\zeta)\hat{J} \quad (3.14)$$

$$k_v = (2\pi BW)((1 + 2\zeta)\hat{J}) - \hat{b} \quad (3.15)$$

3.4.3 Arquitecturas em Cascata e Paralelo

O desenho de controladores passa muitas vezes por arquitecturas diferentes das convencionais, como a demonstrada na Figura 3.15. Como tal, existem outras duas arquitecturas extremamente frequentes denominadas cascata e paralelo.

O controlador em cascata envolve o uso de dois controladores, em que a saída do primeiro é directamente a entrada do segundo. No caso do controlo de actuadores, o primeiro controlador é o responsável pela posição e o segundo pela velocidade, tal como o exemplo da Figura 3.18 em que temos um PID para controlo da posição e um PI para a velocidade.

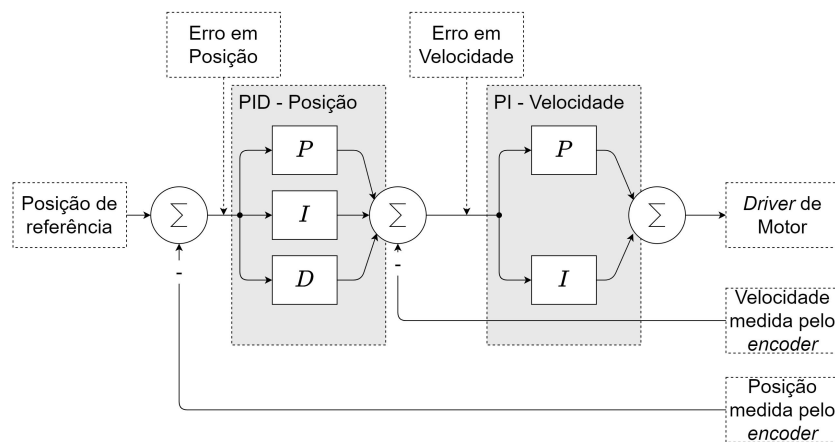


Figura 3.18: Arquitectura típica de um controlador em cascata.

Neste exemplo a entrada do controlador é uma referência de posição que permite calcular o erro tendo em conta a posição medida pelo *encoder*. Este erro serve de entrada no controlador PID que gera um comando de saída relativamente à velocidade a impôr no actuador. Este comando serve de igual forma para calcular o erro em velocidade tendo em conta a velocidade medida pelo *encoder*, permitindo assim que o controlador de velocidade PI calcule o sinal a transmitir para o controlador dos motores.

O controlador em cascata tem como principal vantagem conseguir garantir o controlo quando ocorrem variações de cargas. Mais concretamente, o primeiro controlador garante que se atinge a posição de referência rapidamente e o segundo compensa as variações de carga através de um ganho proporcional elevado.

Relativamente à afinação dos controladores, tipicamente o segundo controlador é afinado primeiro, e de seguida afina-se o primeiro controlador já com o segundo afinado [33].

O controlador com arquitectura em paralelo, tal como se pode verificar na Figura 3.19, é um controlador que tenta seguir uma referência de posição descrita por uma função polinomial, como uma rampa ou uma parábola, através de um conjunto de controladores em paralelo.

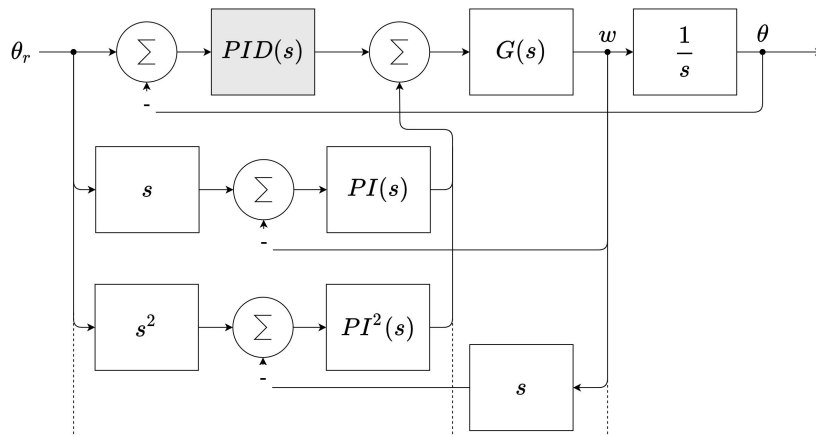


Figura 3.19: Arquitectura típica de controlo em paralelo.

Esta variante, além do controlador PID usado para controlar a posição (que se encontra sombreado na Figura 3.19) tem adicionalmente um controlador PI, que compara a derivada do sinal de referência de posição com a velocidade actual da articulação, de forma a antecipar a correção da mesma. Desta forma, realiza-se um controlo simultâneo da velocidade e da posição.

Adicionalmente, fazendo a segunda derivada do sinal de referência consegue-se corrigir antecipadamente a aceleração que se pretende obter, comparando esta derivada à aceleração da articulação. Como tal, integra-se também um controlador de aceleração. No entanto, este controlador é do tipo PI^2 , que consiste num PI com um integrador adicional, tal como podemos verificar na Equação 3.19. Este integrador adicional permite manter a ordem da função de transferência do sistema constante, devido ao facto de ser necessário derivar a velocidade para se obter a aceleração da articulação.

Desta forma, consegue-se expandir facilmente a ordem do sinal de referência polinomial sem aumentar a ordem da função de transferência do sistema. Isto é, para uma referência polinomial de ordem n , a função de transferência do sistema será sempre igual à descrita na Equação 3.16:

$$\frac{\theta(s)}{\theta_r(s)} = \frac{PID(s) G(s) (1/s) + n PI(s) G(s)}{1 + PID(s) G(s) (1/s) + n PI(s) G(s)} \quad (3.16)$$

Sendo:

$$PID(s) = K_c \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (3.17)$$

$$PI(s) = K_c \left(1 + \frac{1}{T_i s}\right) \quad (3.18)$$

$$PI^2(s) = K_c \left(1 + \frac{1}{T_i s}\right) \frac{1}{s} \quad (3.19)$$

Caso o sistema seja estável, este controlador, além de permitir expandir facilmente a ordem do sinal de referência, permite também obter um erro em regime permanente nulo para uma referência polinomial de ordem n .

No entanto, tem a desvantagem de ser sensível ao ruído existente no sensor de velocidade, visto que este ruído é amplificado através da derivada da velocidade.

Do ponto de vista prático, caso o modelo do processo seja preciso, consegue-se atingir óptimos resultados com um controlador de posição em paralelo com um de velocidade e com um bloco de *Feed-forward*. Se o modelo não for exacto, a arquitectura em cascata representa a melhor solução [34].

3.4.4 Realimentação de Estado e Observadores de Estado

O controlo de sistemas também é viável recorrendo à realimentação de estado. Esta abordagem permite estabilizar um sistema através da modificação da lei de controlo $u(t)$, ou seja, da entrada.

No entanto este tipo de controlo implica que todos os estados do sistema sejam permanentemente conhecidos. Havendo alguns estados que não estão directamente acessíveis, recorre-se aos observadores de estado para os estimar.

A dinâmica de uma articulação pode ser descrita em espaço de estados na forma descrita pelas Equações 3.20, 3.21, 3.22 e 3.23, sendo J_l e J_m o momento de inércia da carga e do motor respectivamente, e B_l e B_m as constantes de amortecimento da carga e do motor respectivamente.

$$\dot{x}_1 = x_2 \quad (3.20)$$

$$\dot{x}_2 = -\frac{k}{J_l}x_1 - \frac{B_l}{J_l}x_2 + \frac{k}{J_l}x_3 \quad (3.21)$$

$$\dot{x}_3 = x_4 \quad (3.22)$$

$$\dot{x}_4 = \frac{k}{J_m}x_1 - \frac{B_l}{J_m}x_4 - \frac{k}{J_m}x_3 + \frac{1}{J_m}u \quad (3.23)$$

A dinâmica da articulação na forma matricial é descrita pela Equação 3.24, com as constantes **A** e **B** definidas em 3.25.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.24)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_l} & -\frac{B_l}{J_l} & \frac{k}{J_l} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & -\frac{k}{J_m} & \frac{B_m}{J_m} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix} \quad (3.25)$$

A saída do sistema é a posição angular da carga descrita pela Equação 3.26, e a constante **C** está definida em 3.28.

$$y = x_1 \quad (3.26)$$

$$= \mathbf{C}^T \mathbf{x} \quad (3.27)$$

$$\mathbf{C}^T = [1, 0, 0, 0] \quad (3.28)$$

A realimentação de estado neste sistema é feita através da lei de controlo $u(t)$, como descrito na Equação 3.29.

$$u(t) = -\mathbf{k}^T \mathbf{x} + r \quad (3.29)$$

$$= \sum_{i=1}^4 k_i x_i + r \quad (3.30)$$

Em que k_i são constantes a serem determinadas e r é uma entrada de referência. Ou seja, o controlo por realimentação de estado é obtido através de uma combinação linear dos estados do sistema, sendo neste caso a velocidade e posição do motor e da carga.

Substituindo a lei de controlo descrita pela Equação 3.29 em 3.24, obtemos uma nova equação do sistema em malha fechada descrita pela Equação 3.31.

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{k}^T)\mathbf{x} + \mathbf{B}r \quad (3.31)$$

Desta forma, a realimentação de estados consegue mudar os pólos do sistema determinados pela matriz $\mathbf{A}$ para os pólos determinados pela matriz $\mathbf{A} - \mathbf{B}\mathbf{k}^T$. No entanto isto implica que o sistema seja controlável.

Um sistema é controlável se e só se o determinante da matriz de controlabilidade for não nulo. Num sistema de ordem n definido pelo par $(\mathbf{A}, \mathbf{B})$, a matriz de controlabilidade tem uma dimensão igual a $n \times n$ e é dada por $[\mathbf{B}, \mathbf{A}\mathbf{B}, \dots, \mathbf{A}^{n-1}\mathbf{B}]$. No caso da articulação, o determinante da matriz de controlabilidade é igual ao dado pela Equação 3.32, que nunca é nulo visto que $k > 0$. Desta forma, pode-se seleccionar quaisquer pólos de malha fechada visto que o sistema é sempre controlável.

$$\det([\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \mathbf{A}^3\mathbf{B}]) = \frac{k^2}{J_m^4 J_l^2} \quad (3.32)$$

Existem vários métodos que permitem determinar os ganhos k_i de forma a obter um dado conjunto de pólos desejados. No entanto existe dificuldade associada à atribuição dos pólos tendo em conta o desempenho pretendido e o binário máximo disponível, isto é, idealmente o sistema tem uma resposta rápida sem requerer muito binário do motor. Desta forma, normalmente a selecção dos ganhos k_i baseia-se na minimização de critérios de desempenho.

A realimentação de estado implica o conhecimento completo do estado do sistema. Como tal, de forma a fazer a realimentação tendo conhecimento apenas da saída do sistema é necessário um observador de estado. Um observador estima o estado do sistema tendo como base o modelo (Equações 3.24 e 3.26) e a saída do sistema $y(t)$.

Desta forma, tendo conhecimento do modelo do sistema (Equação 3.24) e dos parâmetros $(\mathbf{A}, \mathbf{B})$ que o define, é possível obter um observador de estado, como o da Equação 3.33.

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{I}(y - \mathbf{C}^T\hat{\mathbf{x}}) \quad (3.33)$$

O observador de estado representa um modelo do sistema com um termo adicional $\mathbf{I}(y - \mathbf{C}^T\hat{\mathbf{x}})$. Este termo adicional é uma medida do erro entre o valor de saída do sistema $y(t)$ e o valor de saída estimado $\mathbf{C}^T\hat{\mathbf{x}}(t)$. Visto que as matrizes $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ da Equação 3.33 são conhecidas e que a saída do sistema é adquirível directamente, é possível determinar $\hat{\mathbf{x}}(t)$ partindo de qualquer condição inicial, e usar este valor em vez do valor real $\mathbf{x}$ na lei de controlo do sistema, definida pela Equação 3.29.

O termo adicional $\mathbf{I}$ na Equação 3.33 é projectado de forma a permitir que o valor estimado $\hat{\mathbf{x}}$ converja para o valor verdadeiro $\mathbf{x}$ quando $t \rightarrow \infty$, independentemente do estado inicial do sistema.

Definindo o erro de estimação como $\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$, partindo da Equação 3.24 e 3.33, é possível determinar que a dinâmica do erro é igual à descrita pela Equação 3.34.

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{I}\mathbf{C}^T)\mathbf{e} \quad (3.34)$$

Ou seja, a dinâmica do erro é determinada pelos valores próprios de $\mathbf{A} - \mathbf{I}\mathbf{C}^T$. E visto que pretendemos obter um erro nulo quando $t \rightarrow \infty$, o valor de $\mathbf{I}$ tem de ser definido de forma a que os valores próprios de $\mathbf{A} - \mathbf{I}\mathbf{C}^T$ estejam no semi-plano esquerdo. Estes podem ser arbitrariamente definidos desde que o par $(\mathbf{A}, \mathbf{C})$ satisfaça a condição de observabilidade.

Um sistema é observável se e só se o determinante da matriz de observabilidade for não nulo. Num sistema de ordem n definido pelo par $(\mathbf{A}, \mathbf{C})$, a matriz de observabilidade tem uma dimensão igual a $n \times n$ e é dada por $[\mathbf{C}, \mathbf{A}^T \mathbf{C}, \dots, \mathbf{A}^{T^{n-1}} \mathbf{C}]$. No caso da articulação, o determinante da matriz de observabilidade é igual ao dado pela Equação 3.35, que nunca é nulo visto que $k > 0$.

$$\det([\mathbf{C}, \mathbf{A}^T \mathbf{C}, \mathbf{A}^{T^2} \mathbf{C}, \mathbf{A}^{T^3} \mathbf{C}]) = \frac{k^2}{J_l^2} \quad (3.35)$$

O Princípio da Separação permite separar o projeto do observador do projeto da realimentação de estado. No entanto existe um procedimento típico que consiste em colocar os pólos do observador à esquerda dos pólos desejados para $\mathbf{A} - \mathbf{B}\mathbf{k}^T$, de forma a que o valor estimado do estado converja rapidamente para o valor real [32].

Como produto final, os pólos do sistema em malha fechada consistem numa união dos valores próprios de $\mathbf{A} - \mathbf{I}\mathbf{C}^T$ e de $\mathbf{A} - \mathbf{B}\mathbf{k}^T$. E a equação em malha fechada do sistema com o observador de estado é a descrita pela Equação 3.36 [34].

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{k}^T & \mathbf{B}\mathbf{k}^T \\ 0 & \mathbf{A} - \mathbf{I}\mathbf{C}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} \quad (3.36)$$

3.5 Estimação dos Parâmetros do Motor

O projeto de um controlador, tal como referenciado na Secção 3.4.1, pode ser feito de forma mais rápida caso se opte por vias analíticas. No entanto as vias analíticas implicam a existência de um modelo do sistema. Este pode ser obtido a partir da resposta transitória a um degrau, e a partir da resposta em regime estacionário a diferentes tensões. Desta forma, nesta secção tratar-se-á o problema da modelação de um motor a partir da resposta em regime estacionário.

A existência de um modelo de um sistema implica a determinação dos parâmetros que o define. Na Figura 3.20 podemos observar o modelo, e a equação que o define com os respectivos parâmetros em 3.37.

$$V = R i_a + L \frac{di_a}{dt} + e \quad (3.37)$$

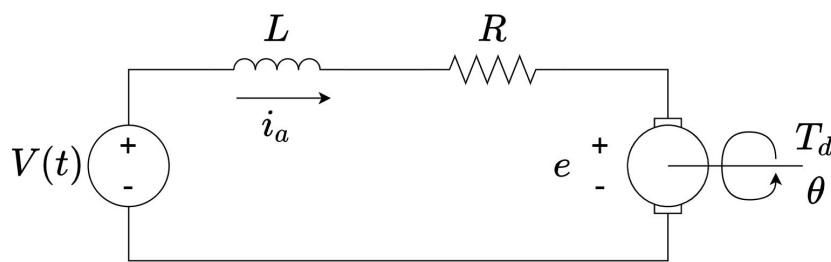


Figura 3.20: Circuito equivalente de um motor DC.

Na Figura 3.20, V representa a tensão aplicada no motor, R e L a resistência e indutância equivalente respectivamente, e e a força contraeletromotriz.

O binário T_L que o motor entrega à carga é igual ao binário produzido T_d subtraído da força do atrito viscoso $B_v \omega$ e do atrito constante ou de *Coulomb* T_c , tal como descrito pela Equação 3.38.

$$T_L = T_d - T_c - B_v \omega \quad (3.38)$$

Sendo ω a velocidade angular que é igual à derivada da posição angular, ou seja, $\dot{\theta}$.

Consequentemente, podemos relacionar o binário produzido T_d com a corrente consumida i_a , a força contraeletromotriz e com a velocidade angular ω , e o binário entregue à carga T_L com o momento de inércia e a aceleração angular. Estas relações encontram-se descritas nas Equações 3.39, 3.40 e 3.41.

$$T_d = K_t i_a \quad (3.39)$$

$$e = K_t \omega \quad (3.40)$$

$$T_L = J \dot{\omega} \quad (3.41)$$

A partir destas equações, mais concretamente, da 3.38, 3.39 e 3.41 é possível determinar a aceleração angular $\dot{\omega}$ como demonstrado na Equação 3.42. A sua discretização, considerando ΔT o tempo de amostragem, resulta na Equação 3.43.

$$\dot{\omega} = \frac{K_t i_a - T_c - B_v \omega}{J} \quad (3.42)$$

$$\omega[k] = \omega[k-1] + \Delta T \frac{K_t i_a[k-1] - T_c - B_v \omega[k-1]}{J} \quad (3.43)$$

Desta forma, minimizando a soma do erro absoluto entre a resposta transitória estimada (Equação 3.43) e a real obtida experimentalmente, podemos estimar os valores de B_v e J . Assumindo também que os valores de K_t e T_c são conhecidos e que a queda de tensão em L é praticamente nula, obtém-se a Equação 3.44.

$$J\dot{\omega} = \frac{K_t}{R}(V - K_t\omega) - B_v\omega - T_c \quad (3.44)$$

Resolvendo a equação diferencial de primeira ordem, obtém-se 3.45

$$\omega(t) = \frac{a}{b}(1 - e^{-bt}) \quad (3.45)$$

Em que:

$$a = \frac{K_t V - RT_c}{RJ} \quad (3.46)$$

$$b = \frac{K_t^2 + RB_v}{RJ} \quad (3.47)$$

Em regime estacionário, em que $\omega = \frac{a}{b}$, a velocidade angular é igual a:

$$\omega = \frac{K_t}{K_t^2 + RB_v}V - \frac{RT_c}{K_t^2 + RB_v} \quad (3.48)$$

Com isto, a minimização do erro absoluto entre o comportamento estimado e o comportamento obtido experimentalmente em regime estacionário, permite estimar os parâmetros K_t e T_c , assumindo um valor inicial para R . De seguida, recorrendo à Equação 3.45, minimiza-se o erro entre os dados estimados para a resposta transitória e os dados reais, o que permite estimar R . Este processo é repetido ciclicamente até que os parâmetros convirjam com os reais. Os parâmetros que são assumidos inicialmente, como T_c , R e K_t são substituídos pelos estimados [35].

3.6 Microcontroladores

Um microcontrolador consiste basicamente num circuito integrado que pode ser programado para realizar uma dada tarefa. Estes sempre representaram a opção mais vantajosa para realizar o controlo de sistemas digitalmente, dado o baixo custo e compactibilidade dos mesmos.

Os periféricos de um microcontrolador estão integrados num único componente de forma a reduzir o peso, tamanho e custo. Têm incorporado também um processador, conversores analógico-digitais, e em alguns casos conversores digitais-analógicos. Estes possuem ainda vários tipos de memória como RAM (*Random-access Memory*), ROM (*Read-only Memory*) e memória *flash*, dependendo do microcontrolador [36].

Estes são capazes de realizar sequências de instruções que se encontram guardadas na ROM, ou na memória *flash* em dispositivos mais recentes.

Os microcontroladores mais frequentes podem operar a 8, 16, 32 ou 64 bits. Tipicamente funcionam a frequências na ordem das centenas de MHz e apresentam um poder computacional elevado.

Em praticamente todos os casos, os microcontroladores têm temporizadores integrados, possuem entradas e saídas adicionais, e têm também comunicação série [37].

Para uma dada aplicação, a escolha do microcontrolador deve ser criteriosa e justificada. Consequentemente, tipicamente a sua escolha baseia-se nos seguintes aspectos [38]:

- **Exigências físicas:** as dimensões do microcontrolador e as condições em que este permanecerá, como por exemplo, temperatura, humidade e pressão.
- **Exigências de processamento:** precisão no cálculo matemático, tempo máximo de execução de ciclos de controlo, tempos máximos de resposta a interrupções, entre outros.
- **Entradas e saídas:** quantos canais digitais tem e qual a sua frequência máxima, quantos canais analógicos tem e qual a sua resolução, entre outros factores.
- **Disponibilidade:** a selecção do microcontrolador deve ser baseada também na disponibilidade no mercado e na existência de ferramentas de desenvolvimento.

3.7 Ferramentas e Tecnologias

A realização desta dissertação envolve também a construção do actuador, o que implica tanto o projeto e a produção da caixa redutora, como a respectiva integração do restante *hardware*.

Desta forma, nesta secção são apresentadas algumas ferramentas e tecnologias que contribuirão no cumprimento dos objectivos.

3.7.1 Ferramentas de Projecto CAD

As ferramentas de projecto CAD (*Computer-aided Design*) consistem em diferentes *softwares* que permitem modelar componentes bi e tridimensionais. Além disto, permitem também realizar montagens compostas por vários componentes e simular a respectiva dinâmica e interacção entre eles.

Algumas das alternativas disponíveis são:

- Autodesk AutoCAD
- Autodesk Inventor
- Autodesk Fusion360
- SolidWorks
- IronCAD

- CadKey

Estas ferramentas permitem exportar os modelos para ficheiros em formatos processáveis para vários métodos de fabricação, como por exemplo, maquinagem CNC (Secção 3.7.2) e impressão 3D (Secção 3.7.3).

3.7.2 Maquinagem CNC

Um dos métodos de fabricação mais utilizados na indústria, que também representa uma potencial alternativa para o fabrico de actuadores, é a maquinagem, que consiste num processo mecânico de modificação de materiais onde há desbaste. Estes são processos precisos e rápidos, mas dispendiosos.

As principais técnicas associadas à maquinagem são:

- Torneamento
- Aplainamento
- Furação
- Serragem
- Rectificação
- Fresagem

Como as diferentes técnicas têm diferentes procedimentos, existem máquinas específicas para cada uma delas. Desta forma, foram integrados nelas módulos de Comando Numérico Computorizado (CNC), que consistem no controlo dos sistemas de segurança, dos meios de comunicação e dos respectivos actuadores, tanto em posição e velocidade. Desta forma, o principal objectivo da tecnologia CNC é permitir controlar os eixos de movimento de uma máquina de forma automatizada.

O uso de máquinas que realizam maquinagem CNC implica a geração de programas com as instruções corretas a serem realizadas. Sendo assim, é necessário usar pós-processadores que geram a partir de ficheiros CAD os programas interpretáveis pelas máquinas. Outra alternativa, são os *softwares* de projecto CAD com pós-processadores integrados, que permitem gerar directamente os ficheiros para a maquinagem CNC, como por o Powermill e Mastercam [39].

3.7.3 Impressão 3D

A produção de actuadores também se torna viável através da impressão 3D. Este método representa uma variante da fabricação aditiva, e apresenta a vantagem de permitir realizar prototipagem rápida, isto é, tendo custos baixos associados, permite integrar e testar mais facilmente um produto na fase de protótipo.

A fabricação aditiva engloba processos como a sinterização por *laser* (SLS - *Selective Laser Sintering*), a estereolitografia (SLA - *Stereolithography*), a modelação por extrusão de plástico (FDM - *Fused Deposition Modeling*), o *Binder Jetting* (BJ), entre outros.

A impressão 3D através do processo de FDM é a opção mais recorrente devido a factores como: o aumento de *softwares open-source* disponíveis, o baixo custo associado, e um mercado vasto de opções.

Para se conseguir realizar uma impressão 3D, é necessário partir do modelo CAD e gerar o *G-Code*, que é o código interpretado pelas Impressoras 3D. Este processo pode ser feito em *softwares* como Cura e Slic3r.

O *G-Code* contém a informação necessária para que a impressora realize a impressão, sendo esta informação dependente da impressora e do filamento plástico seleccionado. Este código consiste em coordenar as operações da máquina através de código CNC, contendo informações como a velocidade de impressão, a altura da camada, o formato da peça, a percentagem de enchimento, entre outros [40].

Capítulo 4

Implementação

O desenvolvimento da Articulação Modular para Braços Robóticos inclui o tratamento do seu *hardware*, *software* e algoritmos de controlo.

Ao longo do desenvolvimento da articulação seguiu-se um conceito de sistema inicial para o funcionamento da mesma. Desta forma, a arquitectura geral pode ser vista de forma simplificada na Figura 4.1.

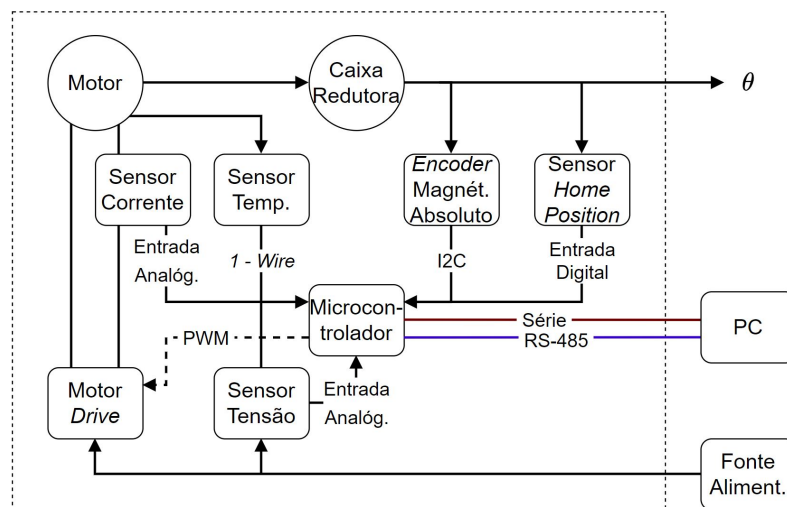


Figura 4.1: Conceito de sistema da Articulação Modular para Braços Robóticos.

Este sistema tem um microcontrolador que recebe, através de comunicação série, comandos que identificam a variável a controlar: velocidade ou posição. Consequentemente, sendo no microcontrolador que se realiza todo o processamento, este recebe os dados provenientes do *encoder* absoluto magnético, o que permite que tenha conhecimento da velocidade e da posição da articulação. Desta forma executa os algoritmos de controlo, e comunica com o *driver* do motor induzindo o seu movimento.

Adicionalmente, de forma a garantir o funcionamento correto do sistema, adicionou-se: um sensor de temperatura, que permite garantir que o motor opera sempre dentro dos seus limites de

segurança; um sensor de tensão de alimentação, que possibilita calcular a tensão no motor; e um sensor de corrente, caso se pretenda controlar a articulação tendo em conta a corrente consumida.

Neste capítulo serão descritos os diferentes componentes relativos ao *hardware*, ao *software* e ao controlo, nas Secções 4.1, 4.2 e 4.3 respectivamente.

4.1 Hardware

O *hardware* da Articulação Modular para Braços Robóticos inclui: a caixa redutora de velocidade, o motor, o *driver* de motor, os sensores, e a *interface* de comunicação.

Desta forma, estes componentes serão tratados detalhadamente ao longo desta Secção.

4.1.1 Caixa Redutora de Velocidade

A caixa redutora de velocidade permite, para a mesma potência, reduzir a velocidade de rotação do motor e aumentar o binário produzido.

Tendo em conta os diferentes tipos de caixa redutora enunciados na Secção 3.1, a arquitectura escolhida para esta aplicação foi a Caixa Redutora Planetária Composta, tal como detalhada na Sub-secção 3.1.5. Esta selecção deve-se ao facto desta permitir obter razões de redução elevadas, sem prejudicar significativamente a existência de *backlash*, enquanto se mantém a caixa compacta.

Nesta configuração, a caixa é composta por dois conjuntos de planetas, dois sóis e dois anéis, cada um deles com um número de dentes característicos. Cada planeta do primeiro conjunto está fisicamente ligado a um planeta do segundo conjunto. Desta forma, o sol do primeiro conjunto funciona como entrada da caixa redutora e o anel do segundo conjunto como saída.

Nas Figuras 4.2, A.1 e A.2 podemos visualizar a composição da caixa redutora desenvolvida.

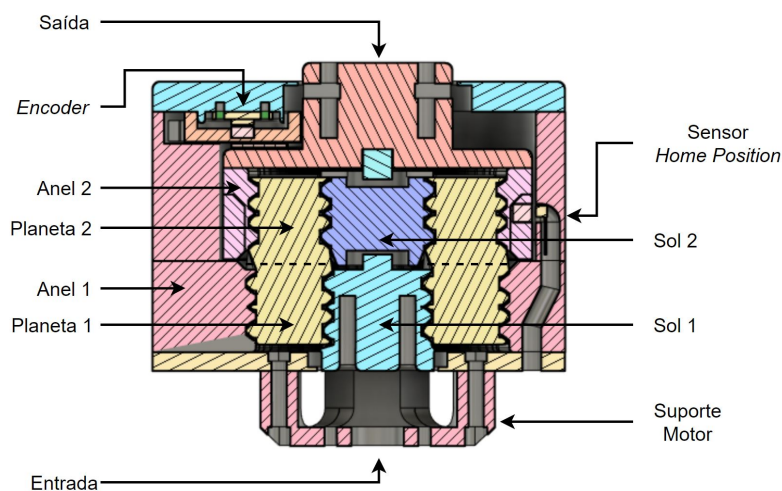


Figura 4.2: Vista seccionada da Articulação Modular para Braços Robóticos.

Tabela 4.1: Número de dentes de cada componente da caixa redutora.

Componente	Número de Dentes
A1	49
P1	15
S1	19
A2	48
P2	14
S2	20

Adicionalmente, na Figura 4.2 é possível verificar a localização do sensor de *Home Position* e do *encoder*. O *encoder* tem um *offset* axial relativamente ao veio de saída, no entanto ambos rodam através de uma relação 1:1.

A razão de redução da Caixa Redutora Planetária Composta é calculada tal como na Equação 4.1.

$$N_g = \frac{\frac{P1}{A1} - \frac{P2}{A2}}{\frac{P1}{S1} + \frac{P1}{A2}} \quad (4.1)$$

Na Equação 4.1, A1 refere-se ao número de dentes do Anel do conjunto 1, P2 refere-se ao dos Planetas do conjunto 2, e assim sucessivamente. Neste caso, os diferentes componentes da caixa redutora têm tantos dentes quanto os explicitados na Tabela 4.1. Tendo em conta estes parâmetros e a Equação 4.1, a caixa redutora tem uma razão de redução de 75,8:1. Esta foi a máxima razão de redução alcançável mantendo a articulação compacta.

As engrenagens da caixa redutora de velocidade são do tipo cilíndricas com dentes helicoidais simétricos. Estes dentes não se encontram alinhados com o eixo da engrenagem, tal como é observável na Figura 4.3.

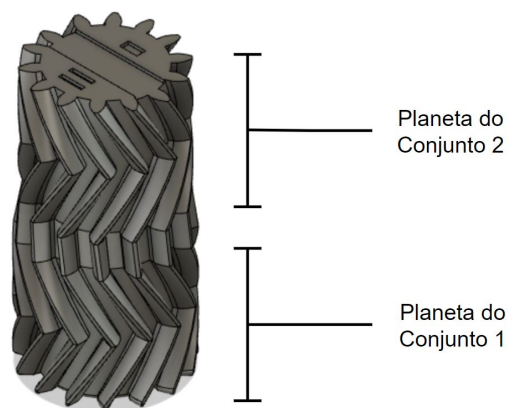


Figura 4.3: Engrenagem cilíndrica com dentes helicoidais simétricos.

Este formato de engrenagens permite que o engrenamento entre pares de engrenagens seja gradual e suave, o que permite reduzir o ruído e aumentar a eficiência do conjunto. Além disto, o facto de serem simétricos confere às engrenagens a capacidade de se auto-centrarem e permanecerem alinhadas.

De forma a maximizar a modularidade da articulação, os encaixes para fixação foram desenvolvidos para que não restrinja qualquer tipo de posicionamento e fixação. Isto é, tal como pode ser visto na Figura 4.4, o local de fixação consiste num padrão rectangular de 40×43 mm ao qual se pode conectar qualquer tipo de adaptador com o mesmo padrão recorrendo a parafusos M3.

O veio de saída, que tem um diâmetro de 40 mm, tem um padrão quadrangular de 18×18 mm ao qual pode ser conectado qualquer tipo de ligação física, ou até directamente outra articulação (tal como na Figura A.8). Tal como a fixação, o veio de saída também usa parafusos M3.

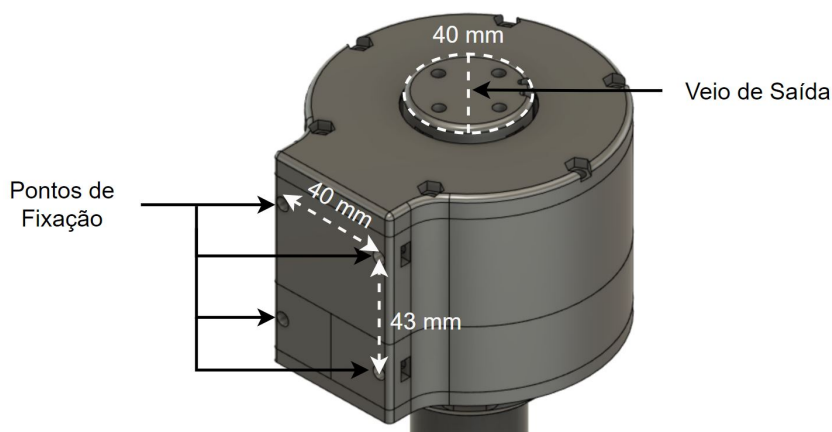


Figura 4.4: Pontos de fixação e veio de saída da caixa redutora.

Um dos objectivos iniciais e principais para esta articulação é o custo final reduzido. Desta forma, como o processo de produção pode trazer custos acrescidos elevados, este factor tornou-se relevante para o processo de desenvolvimento. Consequentemente, escolheu-se a Impressão 3D como método de produção.

Este método, além de ter custos extremamente reduzidos quando comparado com outros, como por exemplo a maquinagem de metais, permite também que esteja rapidamente acessível a qualquer pessoa que tenha acesso a uma Impressora 3D.

Adicionalmente, como o processo de prototipagem torna-se rápido e financeiramente acessível desta forma, também permite que a articulação seja extremamente reconfigurável. Isto é, caso seja necessário adaptá-la a uma nova aplicação, rapidamente se realiza as modificações necessárias e imprimem-se os novos componentes.

No entanto, durante o processo de desenvolvimento, a selecção deste método de fabrico implicou que o projecto fosse modificado de forma a contabilizar as tolerâncias provenientes da Impressão 3D. Estas não podem ser demasiado justas porque após a impressão os componentes

não encaixam correctamente, e não podem ser demasiado largas porque implicam um aumento significativo do *backlash*.

Como produto final deste processo, foi possível obter a caixa redutora de velocidade representada na Figura 4.5 (e no Anexo A, nas Figuras A.3, A.4, A.5 e A.6). Nesta Figura encontra-se já com o motor acoplado ao seu veio de entrada.



Figura 4.5: Caixa redutora de velocidade completa.

4.1.2 Motor

Durante a selecção do motor para a articulação consideraram-se apenas os DC, visto que estes são os mais frequentes na robótica de baixa potência, e são os que mais facilmente se encontram disponíveis no mercado.

Desta forma, considerando os motores estudados ao longo da Secção 3.2: não se optou pelos motores de passo visto que apresentam a desvantagem do seu binário decrescer com o aumento da velocidade, sendo praticamente nulo à velocidade máxima; e não se escolheu o motor sem escovas dada a dificuldade em encontrar controladores/*drivers* no mercado com desempenho satisfatório. Consequentemente, optou-se pelo motor com escovas de ímanes permanentes.

Mais concretamente, o motor DC usado, que se encontra representado na Figura 4.6, é um Surpass Hoby 550-35T, que consiste num motor escovado que consome uma corrente máxima de 20 A e suporta até 7,2 V.

O motor tem o tamanho padrão 550, isto é, tem 36 mm de diâmetro e 58 mm de comprimento, e cada pólo do motor tem enrolamentos com 35 voltas em torno de si mesmo.



Figura 4.6: Motor DC escovado de ímãs permanentes: Surpass Hoby 550-35T.

4.1.3 Driver de Motor

O *driver* de motor seleccionado, que se encontra na Figura 4.7, é composto por duas meias Pontes-H Infineon Technologies BTS7960 [41], o que permite formar uma Ponte-H completa e consequentemente controlar o motor nos dois sentidos de rotação.

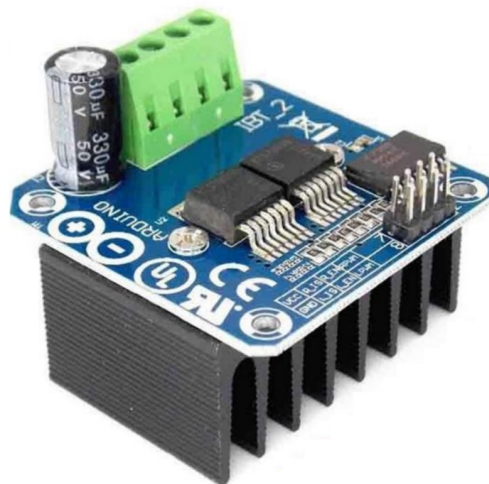


Figura 4.7: *Driver* de motor BTS7960 [14].

A Ponte-H completa suporta uma corrente máxima de 43 A e uma tensão de alimentação máxima de 27 V DC. A nível lógico suporta uma tensão máxima de 5,3 V.

Desta forma, o microcontrolador comunica com o *driver* através de um sinal PWM (*Pulse Width Modulation*) que permite controlar a tensão aplicada no motor. Usa também dois sinais adicionais, sendo cada um deles para uma das meias Ponte-H, que, estando *HIGH* ou *LOW*, controlam o sentido de rotação do motor.

Os motivos que levaram à selecção deste *driver* são: satisfação dos requisitos de tensão e corrente do motor; controlo simplificado recorrendo simplesmente a PWM; e protecção contra consumos de corrente acima do limite máximo.

4.1.4 Encoder Absoluto Magnético

O sensor de posição e velocidade seleccionado é do tipo magnético absoluto. Optou-se pelo magnético visto que, tal como enunciado na Sub-secção 3.3.3, apresenta uma geometria tipicamente pequena e consegue detectar mudanças mínimas no campo magnético. Adicionalmente, optou-se por um absoluto dado que permite obter tanto a posição como a velocidade simultaneamente, sem dados adicionais.

Mais concretamente, o *encoder* utilizado é um AMS AS5600 [42]. É um sensor que se baseia no Efeito de Hall, isto é, usa sensores planares que geram uma tensão proporcional à componente do campo magnético perpendicular ao sensor. Este apresenta uma resolução de 12 bits, é alimentado a 5 V, e mede o ângulo absoluto de um íman magnetizado diametralmente relativamente ao seu eixo de rotação. O ângulo medido cobre uma rotação completa, isto é, de 0° a 360°, tendo sido calibrado com um *encoder* ótico incremental.

Na Figura 4.8 é possível verificar a configuração mecânica associada ao uso do AS5600, isto é, é observável o posicionamento do íman relativamente ao sensor e o uso de um íman magnetizado diametralmente.

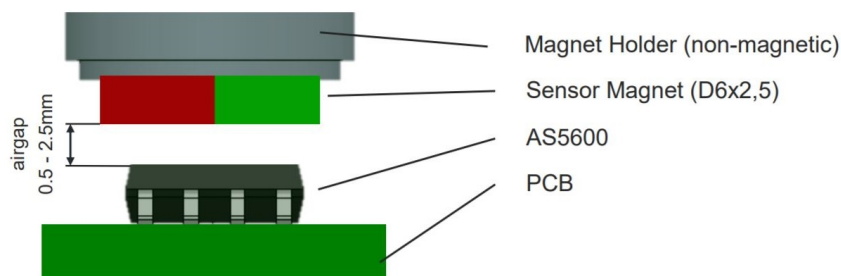


Figura 4.8: Configuração mecânica do *encoder* AS5600 [15].

O *encoder* transmite o ângulo absoluto do íman para o microcontrolador através do protocolo de comunicação I^2C .

Sendo este um *encoder* absoluto, a partir do qual se obtém em graus a posição, pretende-se que este esteja colocado directamente no veio de saída. Para tal, dada a arquitectura da caixa redutora, optou-se por colocar o *encoder* na parte de baixo da tampa superior e fixar o íman a uma engrenagem que roda sincronamente com o veio de saída (Figura 4.9). Ou seja, o íman encontra-se numa engrenagem que tem uma relação unitária (1:1) com outra engrenagem que se encontra no veio de saída.

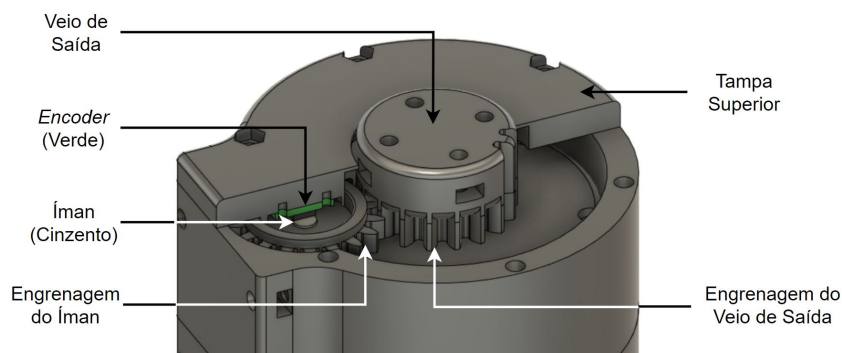


Figura 4.9: Vista seccionada da colocação do *encoder* AS5600.

4.1.5 Sensor de *Home Position*

O *encoder* fornece a posição angular do veio de saída num intervalo de 0° a 360° . No entanto, é praticamente impossível colocar o íman exactamente com a mesma orientação em todas as réplicas da articulação modular. Consequentemente, usa-se um sensor que serve de referência posicional, e que permite obter sempre o mesmo comportamento em todas as caixas redutoras. Isto é, uma dada posição angular é sempre igual em todas as articulações.

O sensor usado para este efeito é um Sensor de Efeito de Hall A3144 da Allegro MicroSystems [43]. Este sensor não é do tipo linear mas sim unipolar, ou seja, o seu valor de saída apresenta o estado *LOW* caso o campo magnético detectado ultrapasse um dado *threshold*, e apresenta o estado *HIGH* caso contrário.

O circuito usado para o sensor é observável na Figura 4.10, onde a sua saída se encontra ligada a uma entrada do microcontrolador.

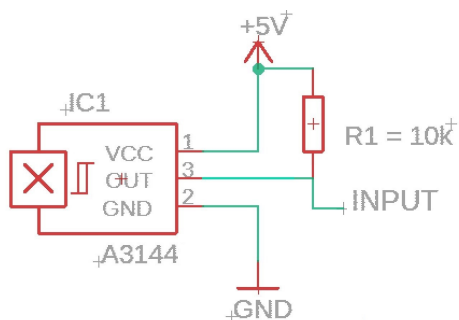


Figura 4.10: Esquema do circuito do sensor de Efeito de Hall A3144.

Relativamente à sua integração na articulação: tal como dito na Sub-secção 4.1.1, a saída da caixa redutora nesta configuração é no segundo anel, que se encontra representado como Anel 2 na Figura 4.2 da página 40. Como tal, o íman a detectar pelo sensor é colocado na lateral desse mesmo anel, que é a saída da articulação, e o sensor é colocado na caixa da articulação. Esta solução encontra-se representada na Figura 4.11.

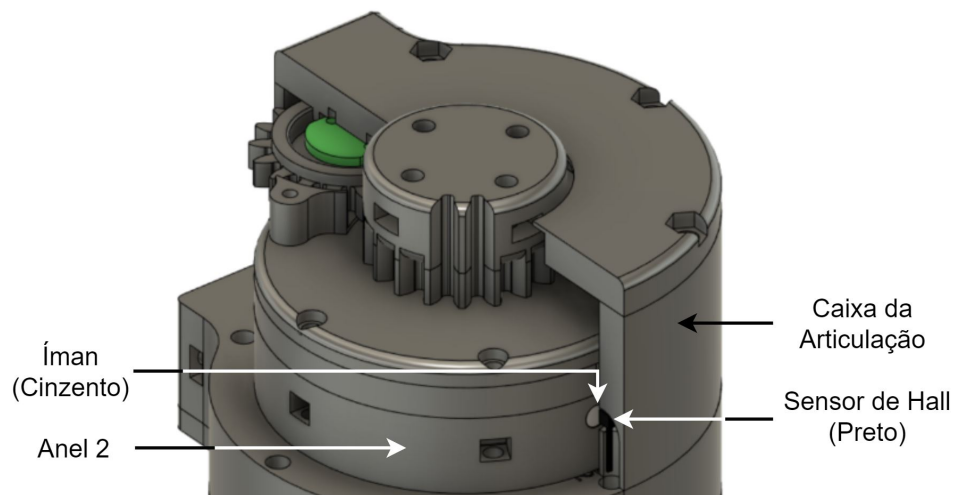


Figura 4.11: Vista seccionada da colocação do sensor de Efeito de Hall A3144.

4.1.6 Sensor de Tensão

A tensão de alimentação é monitorizada com o objectivo final de calcular a tensão no motor. Isto é, sabendo o *duty-cycle* no *driver* do motor, que pode variar entre 0 e 100%, e a tensão de alimentação, obtém-se a tensão no motor tal como a Equação 4.2 demonstra.

$$V_{motor} = \frac{duty}{100} V_{alimentacao} \quad (4.2)$$

A tensão de alimentação tem um valor típico de 12 V, e o microcontrolador tem um conversor analógico-digital que suporta uma tensão máxima de 5 V [44]. Como tal, recorre-se a um divisor resistivo para converter a tensão de alimentação numa tensão dentro do intervalo aceitável do microcontrolador. O circuito usado encontra-se representado na Figura 4.12, onde VCC representa a tensão de alimentação e ADC_INPUT a entrada do conversor analógico-digital.

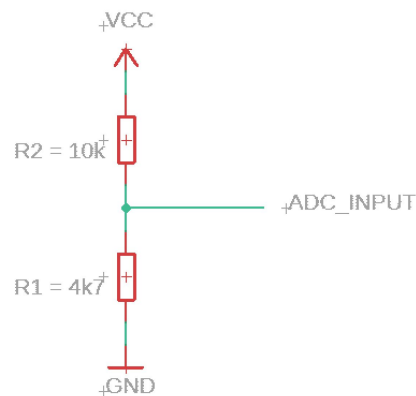


Figura 4.12: Esquema do divisor resistivo para medição da tensão de alimentação.

Desta forma, a tensão à entrada do conversor é dada pela Equação 4.3. E o seu inverso permite obter a tensão de alimentação a partir da tensão lida pelo conversor.

$$V_{ADC_INPUT} = \frac{R1}{R1 + R2} V_{CC} \quad (4.3)$$

4.1.7 Sensor de Corrente

A corrente consumida pelo motor pode ser monitorizada com vários objectivos: estimação dos parâmetros do modelo do motor (Secção 3.5), estimação e controlo do binário produzido, detecção de anomalias, entre outros.

Como tal, tendo em conta que o fabricante do motor explicita que a corrente máxima consumida é 20 A, é necessário escolher um sensor de corrente que suporte esta corrente máxima. Adicionalmente, tem de ser um sensor que suporte tanto corrente positiva como negativa dado os dois sentidos de rotação do motor.

Consequentemente, o sensor escolhido é um ACS712-30A [45] da Allegro MicroSystems. Este sensor suporta correntes no intervalo de -30 A a +30 A, é alimentado a 5 V DC e tem a mesma tensão lógica. É um sensor baseado no Efeito de Hall, e tem uma saída que gera um sinal analógico com uma tensão proporcional à corrente, segundo uma sensibilidade de 66 mV/A.

O circuito usado encontra-se representado na Figura 4.13, onde ADC_INPUT representa a entrada do conversor analógico-digital do microcontrolador, DRIVER a tensão à saída do *driver* do motor, e MOTOR é a saída que aplica a tensão no motor. Com este circuito é possível detectar a corrente consumida apenas pelo motor.

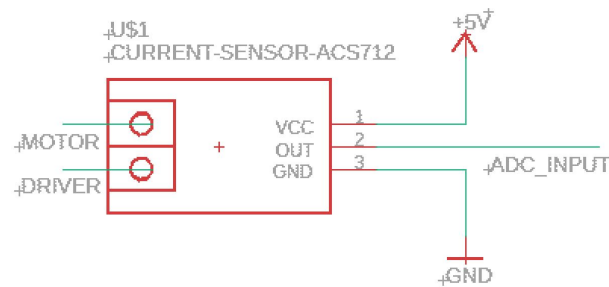


Figura 4.13: Esquema do circuito do sensor de corrente ACS712-30A.

Desta forma, a corrente consumida em função da tensão em ADC_INPUT é dada pela Equação 4.4. Nesta equação inicialmente multiplica-se a tensão lida por 1000, para passar de V para mV, de seguida subtrai-se o *offset* correspondente a 0 A de consumo, e por último divide-se pela sensibilidade do sensor de 66 mV/A.

$$I_{MOTOR} = \frac{V_{ADC_INPUT} \times 1000 - 2500}{66} \quad (4.4)$$

Por último, este método de sensorização foi afinado recorrendo a um multímetro calibrado.

4.1.8 Sensor de Temperatura

A monitorização da temperatura do motor é fulcral para assegurar o seu correto funcionamento, permitindo garantir que este trabalha sempre dentro dos limites de segurança. Para isso, utilizou-se um sensor de temperatura, mais especificamente um Maxim Integrated DS18B20 [46].

Este sensor tem uma resolução configurável que pode variar entre 9 e 12 bits. Neste caso está configurado para 9 bits, permitindo obter uma resolução de 0,5 °C e tempos de conversão mais rápidos (12 ms). Tem também funções de alarme pré-definidas, o que simplifica o processo visto que basta definir as temperaturas nas quais se pretende que o alarme dispare, não sendo necessário que o microcontrolador verifique a temperatura periodicamente. No entanto, como uma máquina eléctrica não é um corpo homogéneo e a temperatura não é medida no ponto mais quente, é necessário assegurar uma margem de segurança.

O sensor comunica com o microcontrolador através do protocolo 1-Wire, sendo desta forma necessário simplesmente uma linha de comunicação e a sua alimentação. O circuito usado para o sensor de temperatura é o da Figura 4.14.

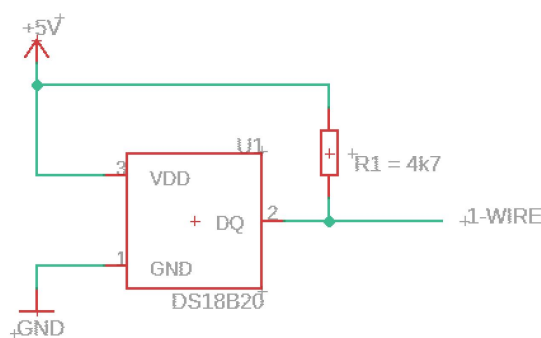


Figura 4.14: Esquema do circuito do sensor de temperatura DS18B20.

O circuito da Figura 4.14 tem uma resistência de *pull-up* tal como a *datasheet* [46] do fabricante recomenda. Deve-se ao facto de que o estado normal da linha de comunicação no protocolo 1-Wire é *HIGH*.

4.1.9 Interface de Comunicação RS-485

A articulação recebe instruções de controlo do exterior. E para que as consiga receber, o protocolo RS-485 foi seleccionado como camada física de comunicação, devido a vantagens como: capacidade de comunicar a elevadas distâncias; boa resistência a interferências; alta velocidade de comunicação; e possibilidade de integrar vários conversores no mesmo par de linhas de comunicação.

O protocolo RS-485 usa dois fios para comunicar, estando tipicamente trançados. Nestes fios (“A” e “B”) são transmitidos sinais diferenciados, sendo que em conjunto definem o nível lógico do sinal, isto é, se a tensão em “A” for maior que em “B” o sinal recebido é 1, e se a tensão em “B” for maior que em “A” o sinal recebido é 0. No entanto, esta lógica também pode ser invertida.

Como tal, para que o microcontrolador tenha esta capacidade de comunicação, usou-se um conversor RS-485 da Maxim Integrated, um MAX485CSA [47]. Este conversor é *half-duplex*, ou seja, não consegue transmitir e receber simultaneamente, tem uma taxa máxima de transmissão de 2,5 Mbps, e suporta um máximo de 32 conversores no mesmo par de linhas de comunicação.

No circuito associado ao conversor, que se encontra representado na Figura 4.15, encontra-se: a entrada DI (*Driver Input*) na qual entra o sinal a ser transmitido pelo microcontrolador; a saída RO (*Receiver Output*) a partir da qual se obtém os dados recebidos; a entrada DE (*Driver Enable*) que é colocada no estado *HIGH* durante a emissão de dados e *LOW* caso não esteja a emitir; e a entrada RE (*Receiver Enable*) que, sendo a negação de DE, é colocada no estado *LOW* durante a emissão de dados e *HIGH* caso não esteja a emitir.

As saídas A e B da Figura 4.15, que é onde o sinal é recebido ou transmitido diferencialmente, têm resistências de *pull-up* e *pull-down* respectivamente, para assegurar que o seu estado permanece constante durante os intervalos de tempo em que não há transmissão de dados.

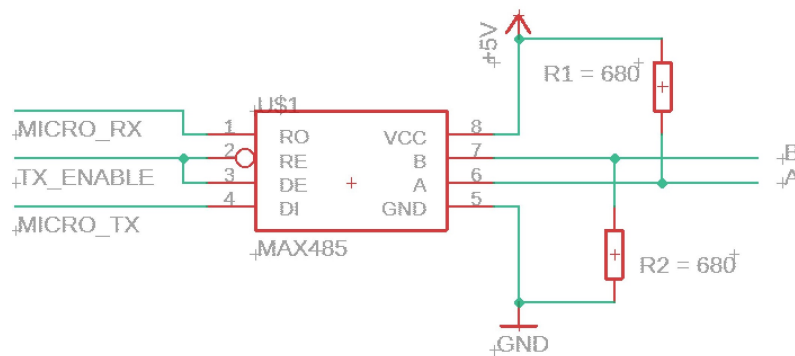


Figura 4.15: Esquema do circuito do conversor MAX485CSA.

4.2 Software

O *software* necessário para o funcionamento da Articulação Modular para Braços Robóticos é composto por 3 partes principais: o protocolo de comunicação entre articulações, e entre articulações e máquinas externas; o *firmware* do microcontrolador, que trata do controlo da articulação; e o *software* que permite controlar um conjunto de articulações simultaneamente.

Nas Sub-seções seguintes, estas diferentes partes serão detalhadas.

4.2.1 Firmware do Microcontrolador

O *firmware* do microcontrolador está composto pelos seguintes elementos: controlo, sensorização, actuação e comunicação.

O elemento controlo é relativo aos algoritmos de controlo da articulação, mais concretamente: o Controlador PI de velocidade, o Controlador PD de posição e o Controlador em Cascata. O algoritmo de controlo em vigência num dado instante é definido pelos comandos de controlo

vindos do Master no protocolo de comunicação (Sub-secção 4.2.2). Todos estes controladores são responsáveis pela atuação do motor, têm um período de execução de 25 ms, e serão abordados com mais detalhe na Secção 4.3. Como o *encoder* utilizado comunica por I^2C , e este não é um protocolo muito rápido, foi determinado que a frequência de controlo mais rápida que permite manter o período constante é 40 Hz.

A sensorização trata, tal como o nome indica, de obter os dados provenientes dos sensores: tensão, corrente, temperatura, *encoder*, e sensor de referência de posição (*Home Position*). Adicionalmente, esta componente também é responsável pelo cálculo da velocidade: tal como os algoritmos de controlo, os valores dos sensores são actualizados com um período de 25 ms (à excepção da temperatura e do sensor de *Home Position*), e como tal a velocidade da articulação é actualizada com a mesma taxa. O sensor de *Home Position* não é actualizado periodicamente com esta frequência, visto que apenas é necessário durante o processo de calibração da posição da articulação, que ocorre no arranque do sistema. E o sensor de temperatura não é actualizado periodicamente, visto que este apenas é necessário quando o *Master* na comunicação realiza uma *query* ao *Slave* (que é a articulação) para saber a temperatura do motor.

A componente de actuação consiste nos procedimentos necessários para o movimento do motor. Como tal, esta componente inclui: os algoritmos de controlo que, tendo em conta o erro da variável a controlar, calculam o *duty-cycle* a comunicar ao *driver* de motor; e as funções necessárias à comunicação com o *driver*, que colocam o PWM na saída do microcontrolador com o *duty-cycle* calculado, e garantem que se induz a rotação no sentido correto.

O protocolo de comunicação usado é o definido na Sub-secção 4.2.2. No entanto, a comunicação é coordenada da seguinte forma: no início de cada ciclo o algoritmo de controlo calcula o *duty-cycle* a aplicar, actualiza-o no *driver*, e logo de seguida o microcontrolador passa ao estado de recetor na comunicação até que se inicie um novo ciclo de controlo. Caso o microcontrolador receba um comando que reconhece, responde de volta a informar que foi recebido com sucesso (emissor). Para garantir a inexistência de *deadlocks*: o início e o término das mensagens são definidos por caracteres específicos e, caso ocorra uma falha durante a recepção, esta é terminada pelo *timeout*.

Para além das componentes acima definidas, o *firmware* do microcontrolador tem também outras partes fundamentais, tais como a configuração de entradas e saídas e a inicialização de variáveis. Consequentemente, a sequência de operações do *firmware* é a representada pelo fluxograma da Figura 4.16.

De forma a cumprir com os objetivos propostos, nos quais se definiram que a articulação tem de ser controlável em velocidade e posição, criaram-se 5 estados para a máquina de estados do microcontrolador: Parado, Controlo PI de Velocidade, Controlo PD de Posição, Controlo Cascata e Calibração da Posição. A alternância entre estados ocorre conforme os comandos que são recebidos. Ou seja, a máquina de estados passa para o estado Controlo PI de Velocidade caso a articulação receba um comando de controlo para uma dada velocidade, e assim sucessivamente para os restantes estados.

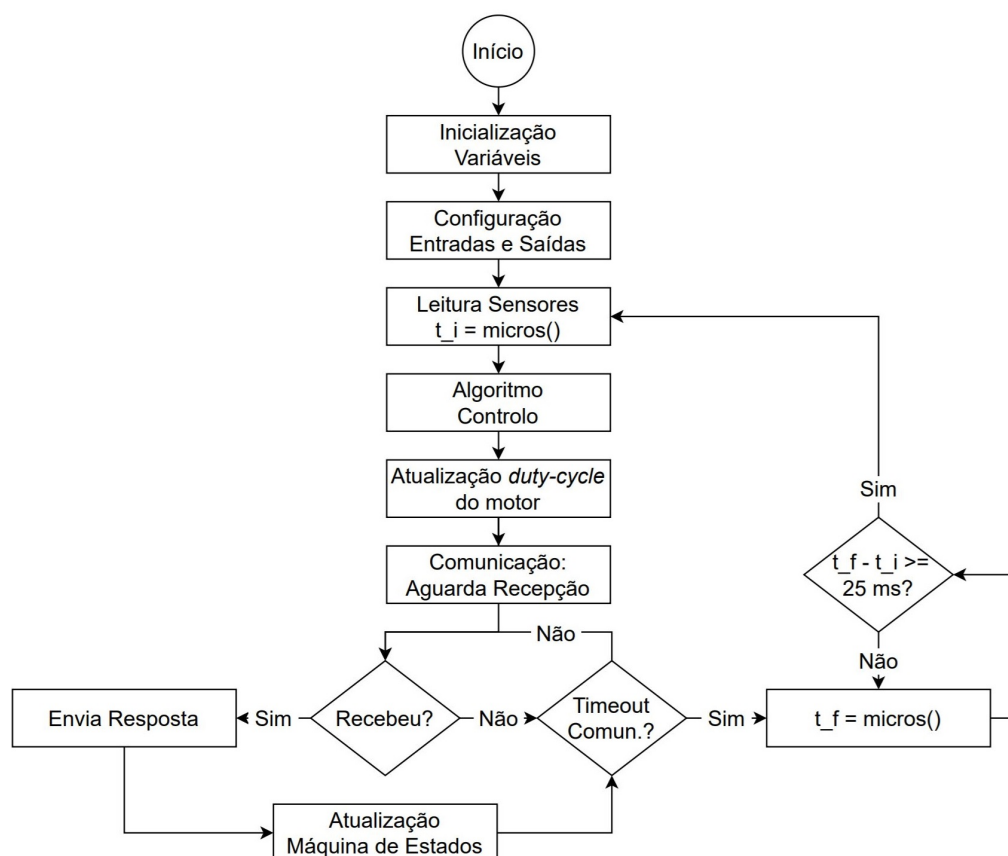


Figura 4.16: Fluxograma do funcionamento do *firmware* do microcontrolador.

4.2.2 Protocolo de Comunicação

A Articulação Modular para Braços Robóticos recebe instruções do exterior, que indicam o tipo de controlo a realizar ou que pedem dados acerca do seu estado actual, tal como temperatura, posição e velocidade.

Como meio de comunicação usou-se RS-485, que define a camada física, recorrendo ao *hardware* definido na Sub-secção 4.1.9. A estrutura desenvolvida suporta um único *Master* e múltiplos *Slaves*, em que cada articulação representa um *Slave*, e o *Master* é um computador que está responsável por controlar um conjunto de articulações. Segundo esta estrutura, apenas o *Master* inicia a comunicação e os *Slaves* simplesmente confirmam a recepção do comando, ou respondem com a informação pedida.

Desta forma, o Protocolo de Comunicação pode ser decomposto em 2 partes: transmissão e formato das tramas das mensagens.

Para a transmissão foi usada uma biblioteca [48] específica para o *hardware* utilizado, que permite o envio de tramas com um tamanho máximo de 255 *bytes*. Esta biblioteca adiciona à trama a transmitir dois *bytes*, um que indica o seu início e outro o seu fim, sendo 0x02 e 0x03 respectivamente. De forma a tornar o protocolo mais resistente à ocorrência de ruído, cada *byte*, excepto o de início e o de fim, é enviado num formato duplicado e invertido. Isto é, cada *nibble*

(4 bits) é enviado duas vezes, sendo uma delas normal e a outra invertida, o que leva a que cada *nibble* passe a ter o tamanho de um *byte* (8 bits). Como por exemplo, visto que o inverso de 0 é F e o inverso de 1 é E, o *byte* 0x01 passa a ser representado pelos bytes 0x0F e 0x1E, sendo o primeiro correspondente ao *nibble* 0, e o segundo ao *nibble* 1. Este método protege contra a ocorrência de ruído, visto que caso ocorra e corrompa um *byte*, é improvável que esta relação normal/inversa se mantenha, detectando-se desta forma o erro. Como este método não se aplica ao *byte* de início e de fim, consegue-se assim distingui-los dos restantes *bytes* da trama.

Para a construção e interpretação das tramas enviadas e recebidas, foi desenvolvida uma nova biblioteca. Nesta são construídas tramas com a informação devidamente colocada, e logo após são transmitidas. Da mesma forma, após a recepção, esta nova biblioteca também interpreta e retira os dados das tramas.

Para simplificar o processo, tendo em conta os requisitos do projecto, estabeleceu-se um tamanho fixo de 6 *bytes* para as tramas. A estrutura utilizada é a da Figura 4.17 e 4.18, sendo a que o *Master* e o *Slave* enviam respectivamente.

Endereço 1 Byte	Comando 1 Byte	Referência - <i>Float</i> 4 Bytes			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5

Figura 4.17: Formato das tramas enviadas pelo *Master*.

Endereço 1 Byte	Confirm. 1 Byte	Resposta - <i>Float</i> 4 Bytes			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5

Figura 4.18: Formato das tramas enviadas pelos *Slaves*.

Os diferentes componentes dos formatos definidos nas Figuras 4.17 e 4.18 são:

- **Endereço** — O Endereço é um número entre 0 e 255 que identifica o receptor da trama. O *Master* tem por defeito o endereço 0, e os restantes são para os *Slaves*;
- **Comando** — O Comando representa o tipo de controlo ou pedido especificado pelo *Master*. Isto é, especifica se o controlador que o *Slave* deve usar é o PD de Posição, o PI de Velocidade ou o Controlador em Cascata. Ou também pode identificar a informação pedida pelo *Master* ao *Slave*, tal como temperatura, posição e velocidade actual;
- **Referência** — A Referência é a variável de referência para o controlador especificado na componente Comando. A referência é nula caso o Comando enviado pelo *Master* seja um pedido de informação. Como a referência é um *float*, usa 4 *bytes*;

- **Confirmação** — A Confirmação serve para o *Slave* informar o *Master* se recebeu o pedido com sucesso. Desta forma, caso não tenha sido com sucesso, o *Master* pode reenviar;
- **Resposta** — Caso o *Master* faça um pedido de informação ao *Slave* acerca de temperatura, posição ou velocidade atual, a componente Resposta possui essa variável quando responde ao Master. Como é um *float*, usa 4 *bytes*.

Por último, a comunicação é sempre iniciada pelo *Master* através das funções especificadas na Sub-secção 4.2.3. O *Slave* está continuamente a ouvir o meio, e responde quando capta uma trama que tenha sido recebida com sucesso e possua o seu endereço como destino.

4.2.3 Software do *Master*

Tal como descrito na Sub-secção 4.2.2, no protocolo de comunicação as articulações são *Slaves*, e existe apenas um *Master* que comunica com um conjunto de articulações. Este *Master* pode enviar comandos de controlo ou realizar *queries* aos *Slaves* para saber dados acerca dos mesmos.

Para tal, desenvolveu-se em cima da biblioteca [48] uma nova biblioteca em *Python*, que permite que um computador comunique com articulações através de RS-485. Esta biblioteca dispõe ao utilizador dos seguintes comandos:

- **setSpeed(address, refSpeed)** — envia um comando de controlo em velocidade para a articulação com o endereço *address* e com referência *refSpeed*;
- **setPos(address, refPos)** — envia um comando de controlo em posição para a articulação com o endereço *address* e com referência *refPos*;
- **setCascade(address, refPos)** — envia um comando para que a articulação com o endereço *address* utilize o controlador em cascata com a referência *refPos*;
- **calibHomePosSensor(address)** — envia um comando de calibração da posição para a articulação com o endereço *address*;
- **stopControl(address)** — envia um comando de paragem para a articulação com o endereço *address*;
- **getSpeedPos(address)** — envia uma *query* de posição e velocidade para a articulação com o endereço *address*;
- **getMotorTemp(address)** — envia uma *query* da temperatura do motor para a articulação com o endereço *address*.

No *script* abaixo apresentado encontra-se um exemplo no qual se envia a uma articulação um comando de posição. Adicionalmente, especifica-se que o controlo deve ser realizado com o controlador em Cascata.

```

1 import actuatorComs as act
2
3 act.init("COM3")
4
5 address = 1
6 refPos = 45.0
7
8 act.setCascade(address, refPos)
9
10 act.close()

```

Neste *script*, inicialmente importa-se a biblioteca desenvolvida para comunicar com as articulações, de seguida inicia-se a comunicação especificando qual a porta usada, seguidamente corre-se o comando tendo o endereço e a referência definidos, e por último fecha-se a comunicação.

4.3 Algoritmos de Controlo

Os Algoritmos de Controlo são responsáveis por calcular o *duty-cycle* do PWM a aplicar no motor pelo *driver*, tendo como objectivo controlar a articulação para uma dada posição ou velocidade.

Na Sub-secção 3.4.1 foi abordado o Controlador PID e algumas das suas variantes, tais como o Controlador PD e PI. Adicionalmente, também foram abordadas técnicas adicionais como por exemplo o *Anti-windup* e o *Feedforward*. No entanto, foram todos demonstrados num domínio temporal contínuo e neste caso serão implementados num domínio temporal discreto. Como tal, visto que todos os controladores usados derivam do Controlador PID, nesta sub-secção inicialmente transpor-se-á o PID para o domínio discreto, e de seguida serão apresentados os implementados na articulação.

Tendo em conta o desempenho obtido com estes controladores, não foi necessário recorrer a controladores mais complexos, como por exemplo a realimentação de estado.

4.3.1 Controlador PID em Tempo Discreto

O Controlador PID em tempo contínuo é representado pela Equação 3.8 da página 24. Esta pode ser discretizada resultando na Equação 4.5.

$$u(k) = K_p e(k) + K_i \sum_0^k \frac{e(k)}{f} + K_d(e(k) - e(k-1))f \quad (4.5)$$

Na Equação 4.5, $e(k)$ simboliza o erro que é igual a $e(k) = r(k) - y(k)$, sendo $r(k)$ a referência e $y(k)$ a variável a controlar. Na mesma equação, a variável f é a frequência do algoritmo de controlo, que, sendo o inverso do período, tem um valor igual a 40 Hz nesta aplicação.

No entanto, considerando T_i o tempo de integração e T_d o tempo de derivação, é possível substituir K_i por $\frac{K_c}{T_i}$ e K_d por $K_c T_d$, o que leva à representação de um PID discreto tal como na Equação 4.6. Nesta Equação, K_c tem o mesmo valor que K_p na Equação 4.5.

$$u(k) = K_c \left(e(k) + \frac{1}{T_i} \sum_0^k \frac{e(k)}{f} + T_d (e(k) - e(k-1))f \right) \quad (4.6)$$

Consequentemente, a implementação discreta do PID segundo a Equação 4.6 é feita tal como no *script* abaixo apresentado, considerando T_s o período do controlador.

```

1  int discretePID(float ref, float controlVar) {
2      float output = 0.0;
3      // Error Calc
4      error = ref - controlVar;
5      // Error Integration
6      intError = intError + error * Ts;
7      // PID Loop
8      output = kc * (error + Td * (error - oldError) / Ts + (1 / Ti) * intError * Ts);
9      // UpdateVars
10     oldError = error;
11     // Returns Integer
12     return round(output);
13 }

```

4.3.2 Controlador PI de Velocidade com Módulo *Feedforward*

O Controlador de Velocidade PI, tal como o nome indica, é constituído por uma componente proporcional e por uma integradora. Desta forma, a sua representação matemática consiste na Equação 4.6 sem a componente derivativa.

Ao Controlador de Velocidade PI foi adicionado um Módulo de *Feedforward* (Sub-sub-secção 3.4.1.2). Este componente é o inverso da função de transferência da articulação, e actua directamente sobre a saída do controlador, tendo como sua entrada o sinal de referência.

Um Módulo *Anti-windup* (Sub-sub-secção 3.4.1.1) foi também adicionado ao controlador PI. Este consiste em limitar a integração de forma a que a variável de saída do controlador não ultrapasse os limites da sua capacidade.

O controlador pode ser representado em diagrama de blocos tal como na Figura 4.19.

Desta forma, a implementação no microcontrolador do Controlador PI de Velocidade com Módulo *Feedforward* consiste no *script* abaixo apresentado. Neste *script*, inicialmente calcula-se e integra-se o erro em velocidade. De seguida, caso a componente integradora do controlador ultrapasse o valor máximo da tensão que o motor suporta, a integração anteriormente feita é revertida. E por último, aplica-se a equação do controlador PI e a componente de *Feedforward*.

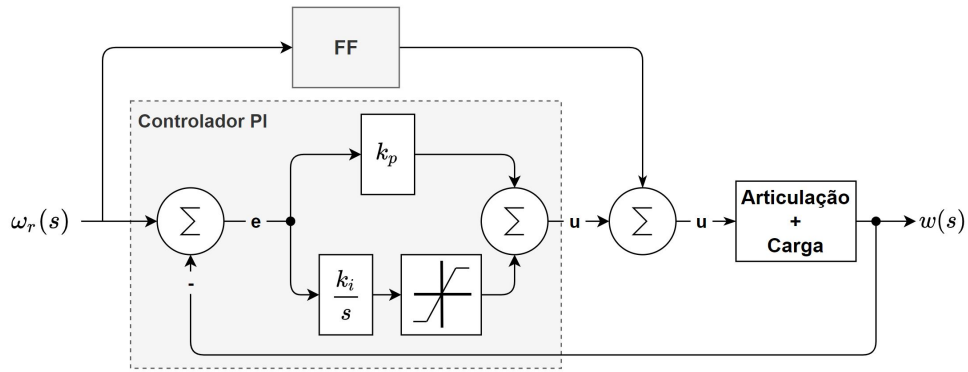


Figura 4.19: Diagrama de blocos do Controlador PI de Velocidade com *Feedforward*.

```

1  int SpeedControlPI(float refSpeed, float actualSpeed){
2      float output = 0.0;
3      // Error Calc
4      speedError = refSpeed - actualSpeed;
5      // Error Integration
6      intSpeedError = intSpeedError + speedError * Ts;
7      // Anti wind up
8      if(intSpeedError * (1/Ti) * kc >= windupLimit){
9          intSpeedError = intSpeedError - speedError * Ts;
10     }
11     else if(intSpeedError * (1/Ti) * kc <= -windupLimit){
12         intSpeedError = intSpeedError - speedError * Ts;
13     }
14     // PI Control Loop
15     output = kc * (speedError + (1 / Ti) * intSpeedError * Ts);
16     // FeedForward
17     output = output + kf * refSpeed;
18     // Output = PWM Value
19     return round(output);
20 }

```

A afinação deste controlador foi feita recorrendo ao Método IMC (*Internal Model Control*) [49], o que implicou a determinação do modelo aproximado de 1ª ordem da articulação, tal como na Equação 4.7.

$$G(s) = \frac{k_p}{\tau s + 1} \quad (4.7)$$

A obtenção do modelo de 1ª ordem iniciou-se com a análise da resposta da articulação a um degrau de velocidade, tal como na Figura 4.20.

Na resposta ao degrau da Figura 4.20, o *duty-cycle* inicial do PWM é 23,5%, e aplica-se um igual a 47% aos 1290 ms. O que leva a que a velocidade inicial seja de 3,16 rad/s e a final

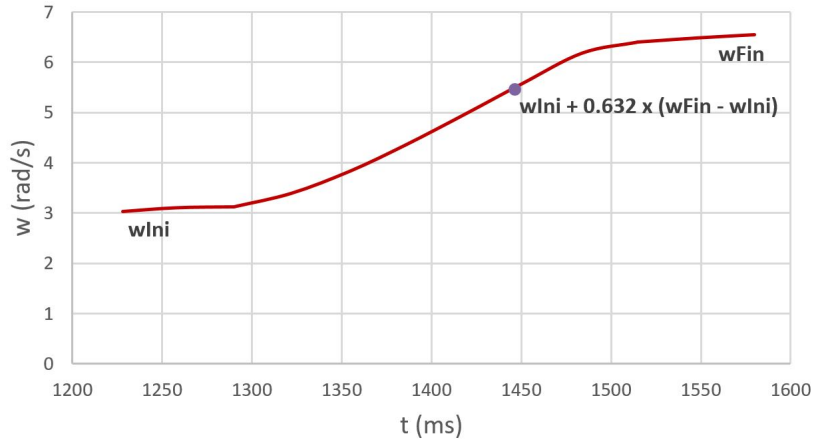


Figura 4.20: Resposta da articulação a um degrau de velocidade: PWM com *duty-cycle* inicial de 23,5% e final de 47%.

de 6,80 rad/s. Adicionalmente, atingiu-se 63,2% da elevação, que corresponde a 5,46 rad/s, aos 1446 ms. Consequentemente, com estes dados conseguimos determinar os parâmetros k_p e τ que caracterizam o modelo de 1ª ordem da articulação, tal como nas Equações 4.8 e 4.9

$$k_p = \frac{wFin - wIni}{dutyFin - dutyIni} = \frac{6,80 - 3,16}{47 - 23,5} = 0,155 \quad (4.8)$$

$$\tau = \frac{1446 - 1290}{1000} = 0,156 \text{ s} \quad (4.9)$$

Desta forma, o modelo de 1ª ordem da articulação é igual à Equação 4.10.

$$G(s) = \frac{0,155}{0,156 s + 1} \quad (4.10)$$

Tendo o modelo de 1ª ordem definido, é possível afinar o controlador recorrendo ao Método IMC, isto é, determinar os valores de k_c e T_i . Como tal, segundo este método, estas variáveis são calculadas tal como nas Equações 4.11 e 4.12.

$$T_i = \tau \quad (4.11)$$

$$k_c = \frac{\tau}{\tau_{cl} k_p} \quad (4.12)$$

Tomando como referência as Equações 4.11 e 4.12, T_i é igual a τ (Equação 4.9), e k_c depende do valor da constante de tempo de malha fechada τ_{cl} seleccionada. Para a seleccionar, foram realizados testes nos quais se verificou a resposta ao degrau, variando o valor de τ_{cl} desde 0,1 s até 1 s, em incrementos de 0,05. Nestes testes compararam-se os seguintes parâmetros: o tempo de subida, que é o tempo necessário para a resposta passar de 10% a 90%, a sobre-elongação, e o erro médio em regime estacionário. O melhor desempenho foi obtido com um valor de τ_{cl} igual a 0,15 s, isto é, com este valor o sistema apresentou o menor tempo de subida, a menor sobre-elongação, e o menor erro médio em regime estacionário, simultaneamente.

O último parâmetro a definir neste controlador é o k_f do *Feedforward*, que é dado pelo inverso do declive da recta que define a velocidade em função do *duty-cycle* do PWM, tal como a da Figura 4.21. Como tal, o valor de k_f é igual a $\frac{1}{0,1516} = 6,596$.

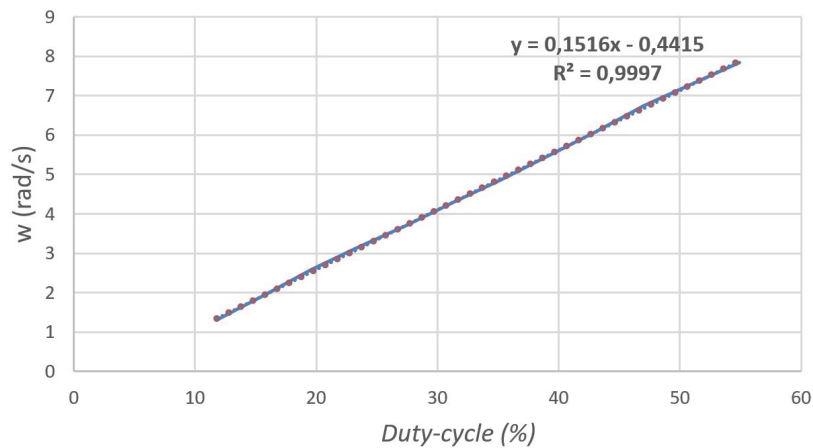


Figura 4.21: Velocidade angular (rad/s) da articulação em função do *duty-cycle* do PWM.

4.3.3 Controlador PD de Posição

O Controlador PD de Posição, é composto por uma componente proporcional e por uma derivativa. Ou seja, a sua representação matemática consiste na Equação 4.6 sem a componente integradora.

Dada a inexistência da componente integradora, este controlador não possui um Módulo *Anti-windup*.

Visto que pretendemos controlar a posição, temos de considerar que esta é resultado da integração da velocidade. Consequentemente, o controlador pode ser representado em diagrama de blocos como na Figura 4.22.

Desta forma, a implementação do Controlador PD de Posição no microcontrolador consiste no *script* abaixo apresentado. Este *script* inicia-se com o cálculo do erro em posição tendo em conta as variáveis de entrada. De seguida aplica-se a equação do Controlador PD, na qual se calcula também a derivada do erro recorrendo ao erro actual e ao erro do ciclo de controlo anterior. E por último, retorna-se o valor do *duty-cycle* do PWM.

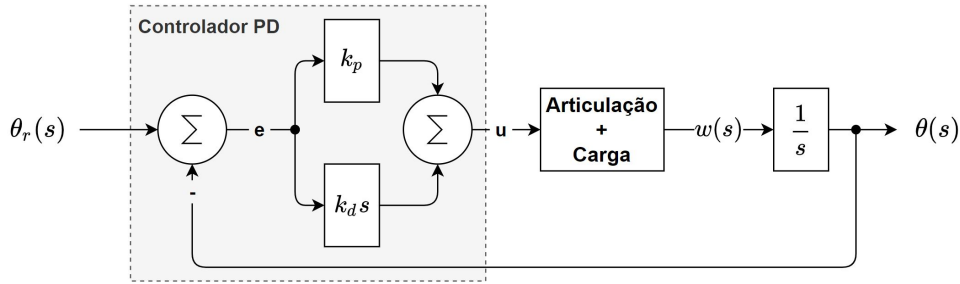


Figura 4.22: Diagrama de blocos do Controlador PD de Posição.

```

1  int PosControlPD(float refPos, float actualPos) {
2      float output = 0.0;
3      // Error Calc
4      posError = refPos - actualPos;
5      // PD Loop
6      output = kc * (posError + Td * (posError - oldPosError) / Ts);
7      // Update Vars
8      oldPosError = posError;
9      // Output = PWM Value
10     return round(output);
11 }

```

A afinação do controlador PD foi feita recorrendo a um Protótipo de Bessel de 2ª Ordem [34]. Para tal, é necessário converter o modelo do sistema para a resposta em velocidade num modelo para a resposta em posição: que consiste em multiplicar $G(s)$ da Equação 4.7 por $\frac{1}{s}$, visto que a posição é obtida a partir da integração da velocidade. Consequentemente, obtém-se a função de transferência dada pela Equação 4.13.

$$G(s) = \frac{k_p}{\tau s^2 + s} \quad (4.13)$$

No entanto, para recorrer ao Protótipo de Bessel para afinar o controlador, é necessária a função de transferência do sistema em malha fechada, tal como na Figura 4.23. Nesta Figura, $G(s)$ representa a Equação 4.13, e $k_c (1 + t_d s)$ representa o Controlador $C(s)$, onde $k_c t_d = k_d$ é o ganho da componente derivativa.

Consequentemente, a função de transferência em malha fechada é dada pela Equação 4.14

$$\frac{\theta(s)}{\theta_r(s)} = \frac{G(s) C(s)}{1 + G(s) C(s)} = \frac{k_c k_p (1 + t_d s)}{(\tau s + 1) s + k_c k_p (1 + t_d s)} \quad (4.14)$$

Como a afinação do controlador consiste no reposicionamento dos pólos da função de transferência, de forma a determinar k_c e t_d , apenas é necessário considerar o denominador $D(s)$ da função

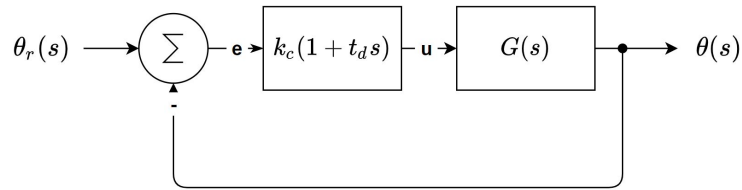


Figura 4.23: Diagrama de blocos do Controlador PD em malha fechada.

de transferência dada pela Equação 4.14. O denominador da função de transferência simplificado na forma canónica dos polinómios $x^2 + a_1x + a_2$ pode ser representado tal como na Equação 4.15.

$$D(s) = s^2 + \left(\frac{1 + k_p k_c t_d}{\tau} \right) s + \frac{k_p k_c}{\tau} \quad (4.15)$$

O Polinómio de Bessel de ordem 2 é descrito pela Equação 4.16, onde r_p e i_p são a componente real e imaginária do pólo p , e t_{sett} é o tempo de estabelecimento desejado para o sistema. Os pólos p do Polinómio de Bessel de ordem 2 correspondentes a um tempo de estabelecimento de 1 s são $-4,0530 \pm j 2,3400$.

$$\left(s - \frac{p}{t_{sett}} \right) \left(s - \frac{\bar{p}}{t_{sett}} \right) = s^2 - \left(2 \frac{r_p}{t_{sett}} \right) s + \frac{r_p^2 + i_p^2}{t_{sett}} \quad (4.16)$$

Consequentemente, é possível estabelecer uma relação entre as Equações 4.16 e 4.15, que permite deduzir as Equações 4.17 e 4.18 que definem o valor de k_c e t_d .

$$k_c = \frac{r_p^2 + i_p^2}{t_{sett}} \frac{\tau}{k_p} \quad (4.17)$$

$$t_d = -2 \frac{r_p}{t_{sett}} \frac{\tau}{k_p k_c} \quad (4.18)$$

Para seleccionar o valor de t_{sett} que melhor satisfaz os requisitos de controlo, realizaram-se testes nos quais se verificou a resposta ao degrau, variando o valor de t_{sett} desde 0,1 s até 1 s, em incrementos de 0,05 s. Tal como no Controlador PI, nestes testes comparou-se o tempo de subida, a sobre-elongação, e o erro médio em regime estacionário. O melhor desempenho nestes 3 parâmetros foi obtido com um valor de t_{sett} igual a 0,5 s.

4.3.4 Controlador em Cascata com Compensação de Zona Morta

O Controlador em Cascata difere dos anteriores principalmente por ser composto por dois controladores individuais. Ou seja, o primeiro controlador, que neste caso é do tipo PD, actua

directamente sobre a referência externa e gera um novo sinal de referência para o controlador seguinte, que neste caso é um PI. O Controlador em Cascata tem como principal vantagem conseguir garantir o controlo estável quando ocorrem variações de cargas.

A zona morta do motor representa uma não linearidade para o sistema, então para minimizar o efeito provocado por esta não linearidade foi adicionado ao Controlador em Cascata uma componente de Compensação de Zona Morta do motor (DZC - *Dead Zone Compensation*). Mais concretamente, tendo conhecimento de qual a tensão a partir da qual o motor sai da zona morta, é possível compensar a saída do controlador de forma a que este opere na zona linear, enquanto não atinge a referência.

A integração da componente de Compensação de Zona Morta do motor com o restante sistema encontra-se representado no diagrama de blocos da Figura 4.24.

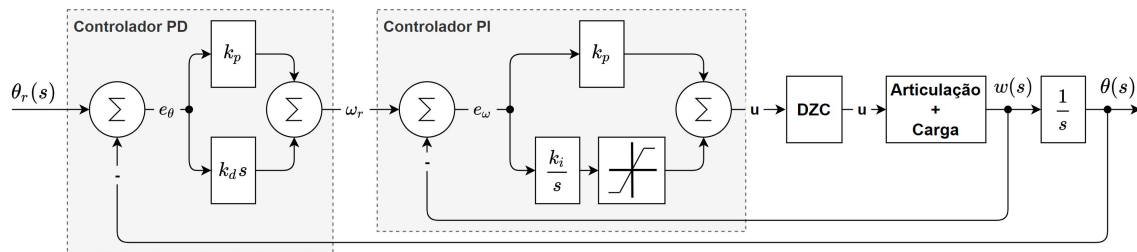


Figura 4.24: Diagrama de blocos do Controlador em Cascata com Compensação de Zona Morta (DZC).

A implementação deste controlador é composta por duas partes principais: o Controlador PD e o Controlador PI. Desta forma, o PD tem uma implementação igual à da Sub-secção 4.3.3 e o PI igual à da 4.3.2. Já a implementação do Compensador de Zona Morta é feita tal como no excerto do *script* do controlador abaixo apresentado. Nesta implementação, *output* é o valor de saída do Controlador em Cascata, *posError* é o erro actual em posição, *minPosError* é o erro mínimo admitido para a posição, e *offset* é o *duty-cycle* mínimo que coloca o motor fora da zona morta.

```

1  int CascadeControl(float refPos, float actualPos, float actualSpeed) {
2      ...
3      // Dead Zone Compensation
4      if((abs(posError) > minPosError) and (abs(output) < offset)){
5          output = (output / abs(output)) * offset;
6      }
7      ...
8  }

```

A Compensação de Zona Morta tem o seguinte procedimento: quando o erro em posição é maior que o valor mínimo admitido e o valor de saída do controlador estiver dentro da zona morta do motor, o valor de saída é corrigido para o valor mínimo que induz o movimento do motor. No final, o valor de *output* é dividido pelo próprio valor absoluto de forma a conservar o sentido de rotação do motor.

Por último, como o Controlador em Cascata é composto por um Controlador PD e um PI, o seu processo de afinação decompõe-se em duas fases: inicialmente afina-se o Controlador PI tal como na Sub-secção 4.3.2, e de seguida afina-se o Controlador PD tal como na Sub-secção 4.3.3, considerando como sistema a controlar a articulação juntamente com o Controlador PI.

4.4 Estimação dos Parâmetros da Articulação

Na Secção 3.5 foi abordada a estimação dos parâmetros que definem o modelo de um motor. Da mesma forma, é possível estimar os da articulação, que é composta pela caixa redutora de velocidade e pelo motor.

Desta forma, os parâmetros da articulação a obter são: a resistência equivalente R , o coeficiente de atrito viscoso B_v , o atrito constante ou de *Coulomb* T_c , a constante de binário do motor K_t , o momento de Inércia J , e a indutância L .

Neste caso, os primeiros 4 parâmetros foram obtidos a partir da resposta da articulação em regime estacionário a diferentes tensões. Para tal, aplicou-se tensões fora da zona morta do motor sucessivamente maiores, e registou-se para cada uma delas a tensão, a corrente consumida e a velocidade da articulação em regime estacionário. Os resultados obtidos da velocidade e da corrente em função da tensão, e a respectiva aproximação linear, encontram-se representados nas Figuras 4.25 e 4.26.

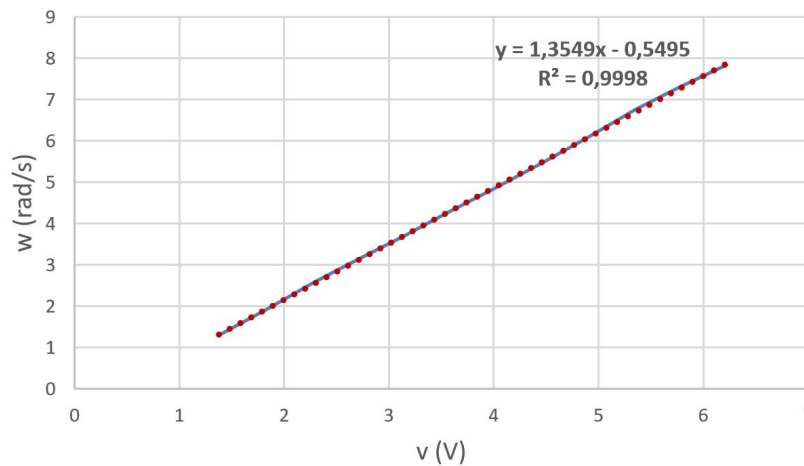


Figura 4.25: Velocidade angular da articulação em função da tensão.

Em regime estacionário considera-se que a corrente consumida é praticamente constante, o que leva a que a sua derivada seja nula. Desta forma, partindo da Equação 3.37 ($V = R i_a + L \frac{di_a}{dt} + e$), e da Equação 3.40 ($e = K_t \omega$), é possível deduzir a Equação 4.19.

$$V = R i_a + K_t \omega \quad (4.19)$$

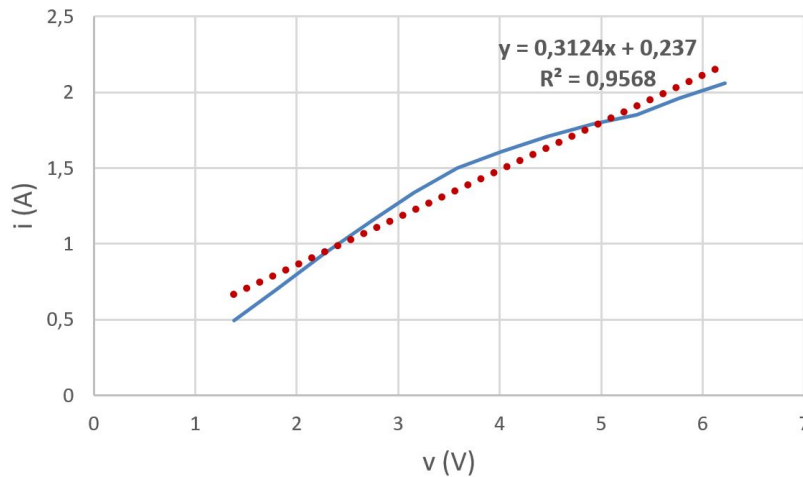


Figura 4.26: Corrente consumida pela articulação em função da tensão.

No entanto, para que seja possível obter o valor da resistência R e da constante de binário do motor K_t , transformou-se a Equação 4.19 na Equação 4.20.

$$\frac{V}{\omega} = R \frac{i_a}{\omega} + K_t \quad (4.20)$$

Partindo da Equação 4.20 obtém-se o gráfico da Figura 4.27, permitindo assim obter o valor de R a partir do declive da recta de tendência, e o valor de K_t a partir da ordenada na origem. Consequentemente, a melhor aproximação possível que se pode obter para R é $1,3979 \Omega$, e para K_t é $0,3974 \text{ Nm/A}$.

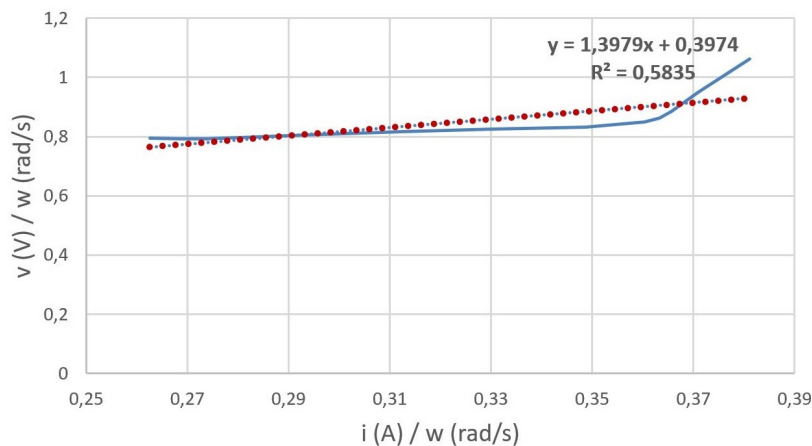


Figura 4.27: Razão V/ω em função da razão i/ω .

Para determinar os parâmetros B_v e T_c é necessário partir da Equação 3.42 da página 33 e considerar um ensaio estático. E como um ensaio estático implica que a derivada da velocidade seja nula, pode-se obter as Equações 4.21 e 4.22 a partir da 3.42.

$$0 = K_t i_a - T_c - B_v \omega \quad (4.21)$$

$$K_t i_a = B_v \omega + T_c \quad (4.22)$$

Tendo em conta que K_t já está definido, e que os valores da velocidade angular ω e da corrente i_a foram obtidos experimentalmente, é possível definir o gráfico da Figura 4.28 que define a Equação 4.22. Desta forma, analisando o gráfico pode-se concluir que o valor aproximado para B_v é dado pelo declive da aproximação linear, e que o dado para T_c é definido pela respectiva ordenada na origem. Sendo $T_c \approx 0,1447$ Nm e $B_v \approx 0,0916$ Nm.s/rad.

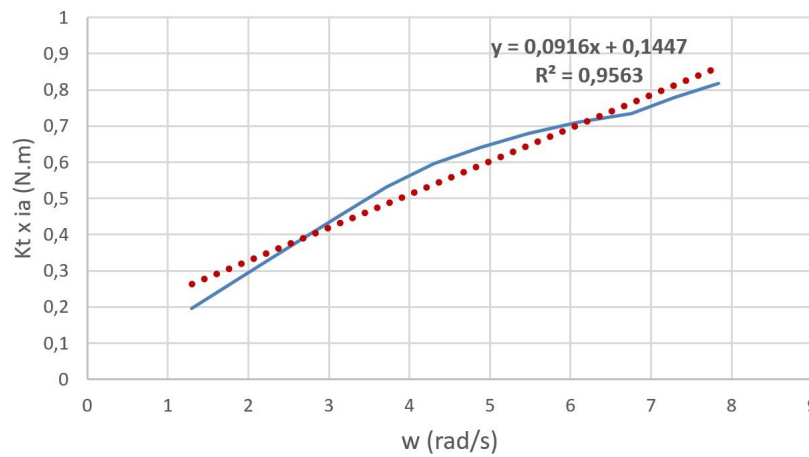


Figura 4.28: Binário $K_t i_a$ gerado pela articulação em função da velocidade angular ω .

É possível comprovar os resultados obtidos, comparando os dados experimentais representados no gráfico da Figura 4.25 com as curvas geradas pelas Equações 4.23 e 4.24, visto que as equações contêm os parâmetros estimados.

$$V_1 = \left(\frac{R B_v}{K_t} + K_t \right) \omega + \frac{R T_c}{K_t} \quad (4.23)$$

$$V_2 = R T_c + K_t \omega \quad (4.24)$$

Consequentemente, na Figura 4.29 é possível observar a comparação entre os resultados obtidos experimentalmente (V_{exp}) e os resultados obtidos a partir das Equações 4.23 (V_1) e 4.24 (V_2) usando os parâmetros estimados. E tal como é verificável na figura, as rectas geradas com os parâmetros obtidos estão bastante próximas dos dados originais, o que permite concluir que o modelo desenvolvido da articulação está aproximadamente preciso.

Por último, a determinação do momento de inércia J e da indutância L da articulação é feita a partir de um ensaio dinâmico, sendo neste caso a resposta a um degrau. Para tal, coloca-se as Equações 3.37 e 3.42 no domínio de Laplace, originando as Equações 4.25 e 4.26.

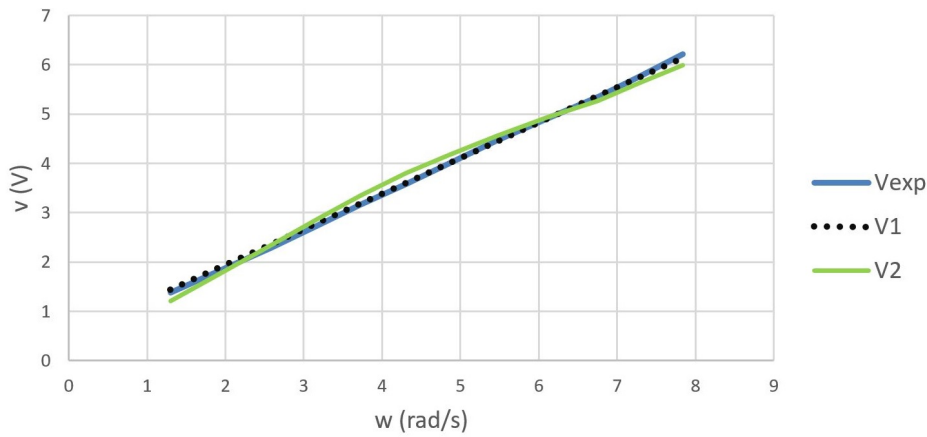


Figura 4.29: Comparação dos dados experimentais aos gerados pelas Equações 4.23 e 4.24.

$$V(s) = R i_a(s) + s L i_a(s) + K_t \omega(s) \quad (4.25)$$

$$J s \omega(s) = K_t i_a(s) - B_v \omega(s) \quad (4.26)$$

Desta forma, conjugando as Equações 4.25 e 4.26, é possível deduzir uma função de transferência de 2ª ordem no domínio de *Laplace*, dada pela Equação 4.27.

$$\frac{\omega(s)}{V(s)} = \frac{\frac{k_t}{JL}}{s^2 + s \left(\frac{R}{L} + \frac{B_v}{J} \right) + \left(\frac{R B_v + K_t^2}{JL} \right)} \quad (4.27)$$

Na função de transferência definida pela Equação 4.27 todas as variáveis estão definidas excepto L e J , que são as que pretendemos determinar. Como tal, é necessário definir a função de transferência de 2ª ordem para que as incógnitas possam ser extraídas [50].

Para tal, recorrendo aos dados obtidos na resposta ao degrau representada na Figura 4.20, definiu-se a função de transferência de 1ª ordem representada na Equação 4.28.

$$\frac{\omega(s)}{V(s)} = \frac{1,486}{0,156s + 1} \quad (4.28)$$

De seguida, recorrendo à aplicação de Identificação de Sistemas [51] da *Toolbox* do *Matlab*, e usando os dados obtidos na resposta ao degrau, obteve-se uma função de transferência de 2ª ordem para o mesmo sistema. A função de transferência de 2ª ordem é a representada na Equação 4.29.

$$\frac{\omega(s)}{V(s)} = \frac{4534}{s^2 + 482,6 s + 3051} \quad (4.29)$$

Ao longo deste processo, pretendia-se minimizar a diferença entre a resposta ao degrau da função de transferência de 1ª ordem e a resposta da de 2ª ordem. Na Figura 4.30 é possível comparar as respostas ao degrau dadas por ambas funções de transferência.

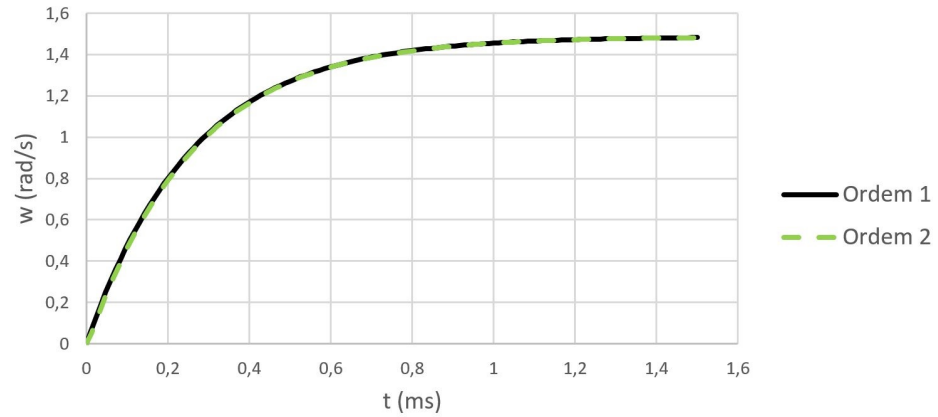


Figura 4.30: Comparação das respostas ao degrau dadas pelas funções de transferência estimadas.

Consequentemente, igualando a função de transferência definida na Equação 4.29 à Equação 4.27, é possível obter a igualdade que define as incógnitas. Desta forma, obteve-se os seguintes valores: $L = 0,4909$ H e $J = 1,909 \times 10^{-4}$ kg.m².

Capítulo 5

Resultados

Este capítulo apresenta os resultados finais obtidos com a articulação modular totalmente construída e funcional.

Inicialmente serão apresentados os resultados relativos aos aspectos construtivos da articulação, como por exemplo, *backlash*, peso e custo. E de seguida serão apresentados e discutidos os resultados associados ao desempenho dos controladores.

5.1 Resultados Construtivos

Após o desenho e projeto da articulação, procedeu-se à impressão da mesma, e adicionou-se todo o *hardware* necessário. Como tal, o produto final, que inclui caixa redutora, motor, rolamentos, parafusos, fêmeas, sensores e cabos, ficou com uma massa final igual a 890 g e um *footprint* igual a $100 \times 110 \times 168$ mm.

Relativamente ao custo da articulação, a soma de todos os componentes implica um valor total de 95,13 €. E na Tabela 5.1 é possível observar a discriminação detalhada do valor final.

O *backlash* da articulação foi um factor extremamente considerado durante o projeto da mesma, e como tal tornou-se fulcral determinar o seu valor final. Para tal: fixou-se o veio de entrada da caixa redutora, que é onde o motor acopla; ligou-se um braço com 200 mm de comprimento ao veio de saída, tal como na Figura 5.1; fixou-se a articulação a uma base sólida; e tirou-se amostras do deslocamento do braço com a entrada da articulação fixa. Como resultado deste processo, o deslocamento médio da articulação a 200 mm foi de 1,811 mm que, traduzindo para ângulos de rotação, corresponde a um *backlash* de 0,00453 rad, que é equivalente a $0,26^\circ$.

Como o objectivo final da articulação modular é permitir construir manipuladores, é essencial determinar o binário máximo que consegue produzir. A determinação do binário máximo procedeu-se da seguinte forma: ligou-se um braço com 200 mm de comprimento ao veio de saída, tal como na Figura 5.1; fixou-se a articulação a uma base sólida; colocou-se a extremidade do braço na superfície de uma balança; aplicou-se a tensão máxima no motor no sentido de rotação que induzia pressão sobre a balança; e tirou-se amostras da força gerada pela articulação na balança. Consequentemente, a força média gerada pela articulação a 200 mm foi de 2,313 kg

Tabela 5.1: Lista detalhada dos custos de produção da articulação.

Componente	Custo (€)
Filamento de Impressão 3D	6,00
Rolamentos de Esferas	15,80
Rolamentos Axiais	1,00
Parafusos e Fêmeas	0,85
Microcontrolador ATmega328P	12,90
Sensor de Temperatura DS18B20	4,46
Sensor de Corrente ACS712-30A	4,70
Sensor de Hall A3144	0,99
Encoder AS5600	6,99
Motor Surpass Hoby 550-35T	17,09
Driver de Motor BTS7960	14,40
Conversor RS-485 MAX485CSA	5,50
Adaptador USB-RS-485	4,45
Total	95,13

que, traduzindo para binário, corresponde a 46,26 kg.cm ou 4,54 Nm. Isto implica uma relação binário/peso igual a $\frac{4,54 \text{ Nm}}{0,890 \text{ kg}} = 5,10 \text{ Nm/kg}$.



Figura 5.1: Articulação Modular com braço de testes de 200 mm.

Com o binário máximo da articulação determinado, é possível calcular a eficiência η_{Caixa} da caixa redutora. Sendo N_g a razão de redução da caixa, a eficiência é calculada pela Equação 5.1.

$$\eta_{Caixa} = \frac{T_{MáxArticulação}}{N_g T_{MáxMotor}} \times 100\% \quad (5.1)$$

Adicionalmente, para calcular a eficiência, também é necessário saber o binário máximo produzido pelo motor sem caixa redutora. Então realizou-se apenas com o motor exactamente o

Tabela 5.2: Características construtivas da Articulação Modular.

Peso	Footprint	Custo	Backlash	T_{Max}	η_{Caixa}	ω_{Max}
890 g	$100 \times 110 \times 168$ mm	95,13 €	$0,26^\circ$	4,54 Nm	30,52 %	$449,1^\circ/s$

mesmo procedimento que se realizou para obter o binário máximo da articulação completa, tendo-se obtido um valor final para o binário máximo do motor de 0,196 Nm. Consequentemente, recorrendo à Equação 5.1 e sabendo que $N_g = 75,8 : 1$, obteve-se uma eficiência da caixa redutora de 30,52%.

A velocidade máxima de rotação da articulação é de 7,84 rad/s, que é equivalente a $449,1^\circ/s$. Por último, a Tabela 5.2 apresenta um resumo dos resultados obtidos nesta Sub-secção.

5.2 Resultados de Desempenho

Os resultados de desempenho consistem na análise do comportamento dos três controladores desenvolvidos: o Controlador PI de Velocidade com Módulo *Feedforward* (Sub-secção 4.3.2), o Controlador PD de Posição (Sub-secção 4.3.3), e o Controlador em Cascata com Compensação de Zona Morta (Sub-secção 4.3.4).

Ao longo dos testes realizados para obter os resultados de desempenho, foram usadas Splines Cúbicas de Hermite [52] para gerar os sinais de referência para os controladores. Isto é, quando os controladores recebem um novo valor como referência, automaticamente realizam uma interpolação entre o valor actual da variável a controlar e a nova referência. Em todas as Figuras demonstrativas de resultados serão apresentadas as respectivas Splines de referência geradas.

Na Figura 5.2 é possível verificar a resposta à referência do Controlador PI de Velocidade com Módulo *Feedforward*.

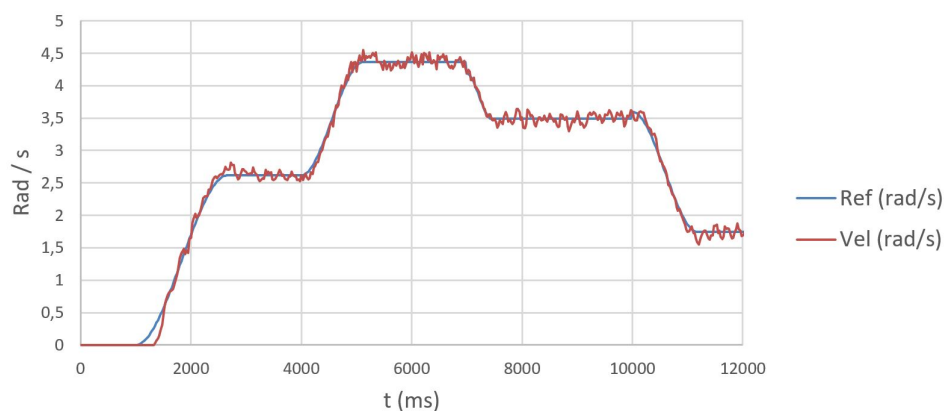


Figura 5.2: Resposta do Controlador PI de Velocidade com FF a diferentes comandos de velocidade.

Na Figura 5.3 é verificável a resposta do Controlador PD de Posição, e na Figura 5.4 a do Controlador em Cascata com Compensação de Zona Morta (DZC). Para ambos foi usado o mesmo sinal de referência.

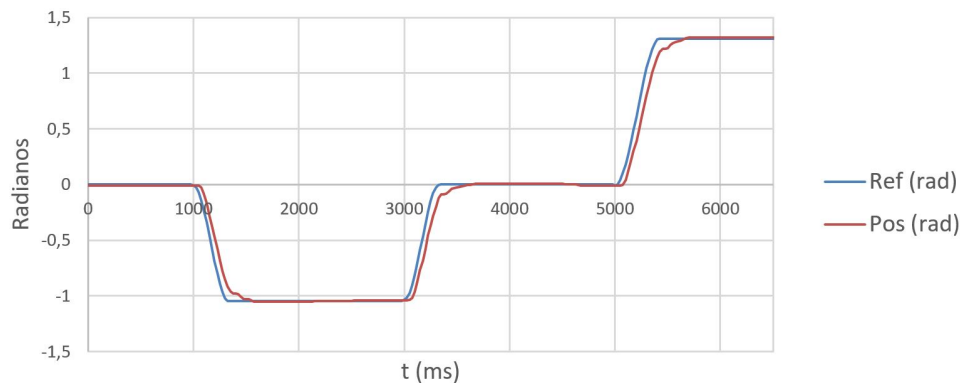


Figura 5.3: Resposta do Controlador PD de Posição a diferentes comandos de posição.

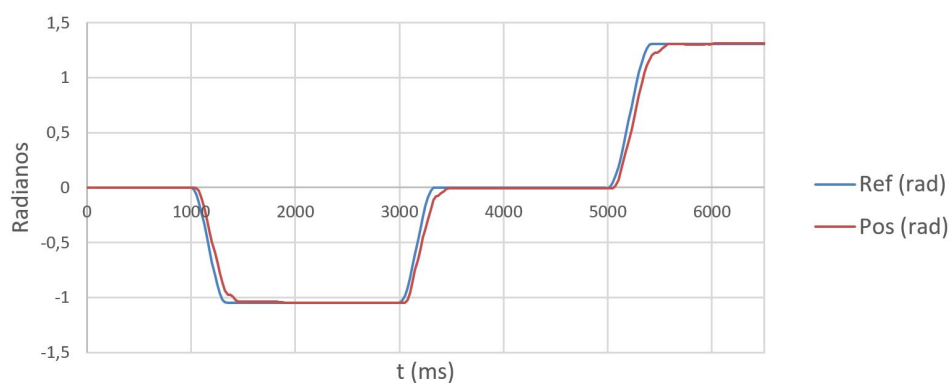


Figura 5.4: Resposta do Controlador em Cascata com DZC a diferentes comandos de posição.

A avaliação do desempenho dos controladores foi feita medindo o erro máximo absoluto e o erro absoluto médio para cada um dos controladores. Esta análise foi feita tanto para a resposta transitória, como para regime estacionário. As Tabelas 5.3 e 5.4 descrevem os erros para cada controlador, sendo a primeira tabela correspondente à resposta transitória, e a segunda ao regime estacionário.

O Controlador PI de Velocidade, apesar de seguir facilmente a referência, apresenta erros significativos tanto transitórios como estacionários, tal como é verificável nas Tabelas 5.3 e 5.4. Desta forma, também se pode concluir que estes erros estão associados ao ruído proveniente do cálculo da velocidade, visto que como é feito a partir da derivada da posição, todos os ruídos associados ao *encoder* também são derivados e consequentemente amplificados.

Relativamente ao Controlador PD de Posição e ao Controlador em Cascata com Compensação de Zona Morta (DZC), tal como é evidente nas Tabelas 5.3 e 5.4, a configuração em Cascata apresenta os menores erros tanto em regime transitório como estacionário, sendo desta forma o

Tabela 5.3: Erros transitórios dos controladores.

Controlador	Erro Máximo Absoluto	Erro Absoluto Médio
Controlador PI de Velocidade com FF	0,2820 rad/s	0,0728 rad/s
Controlador PD de Posição	0,2480 rad	0,1226 rad
Controlador em Cascata com DZC	0,2180 rad	0,0913 rad

que apresenta o melhor desempenho entre os dois. Adicionalmente, observando cuidadosamente os gráficos das Figuras 5.3 e 5.4, é possível concluir que o Controlador em Cascata atinge o regime estacionário mais rapidamente que o Controlador PD, sendo em média 75 ms mais rápido.

Tabela 5.4: Erros dos controladores em regime estacionário.

Controlador	Erro Máximo Absoluto	Erro Absoluto Médio
Controlador PI de Velocidade com FF	0,1910 rad/s	0,0524 rad/s
Controlador PD de Posição	0,0160 rad	0,0064 rad
Controlador em Cascata com DZC	0,0110 rad	0,0041 rad

Capítulo 6

Conclusão e Trabalho Futuro

Neste último capítulo será feita uma reflexão no processo de desenvolvimento e na satisfação dos objectivos do projecto. E será feita também uma análise dos trabalhos futuros necessários para o melhoramento dos resultados obtidos.

6.1 Conclusão

O objectivo desta dissertação consistia em desenvolver uma Articulação Modular para Braços Robóticos. Esta teria de ser de baixo custo, facilmente reconfigurável e com controlo em posição e velocidade incorporado.

Inicialmente, realizou-se uma análise do estado da arte onde se detalhou soluções semelhantes. Dentro das soluções apresentadas, umas foram desenvolvidas a nível académico e de investigação, e outras representam opções já disponíveis no mercado.

De seguida, foi feito um estudo acerca da fundamentação teórica necessária ao desenvolvimento deste projecto. Desta forma, foi possível ponderar as opções disponíveis associadas a *hardware* e algoritmos, e fazer selecções justificadas. Como por exemplo, o estudo dos sistemas de actuação para articulações rotativas permitiu seleccionar a caixa redutora planetária composta como a arquitectura mais adequada ao efeito.

A implementação arrancou com o desenho 3D da articulação no Autodesk Fusion360. Inicialmente desenharam-se as engrenagens que compõem a caixa redutora, e de seguida desenvolveu-se a restante articulação em volta das engrenagens, de forma a integrar o *encoder*, o sensor de *Home Position*, o motor, os rolamentos e os pontos de fixação. Aquando o término do desenho, todos os componentes foram fabricados através de Impressão 3D, o que permitiu agilizar e acelerar o processo de prototipagem. Os primeiros testes realizados com a articulação permitiram tomar conhecimento de potenciais melhorias, o que levou a que se retocasse o desenho e se repetisse o processo. Consequentemente, a 3ª versão totalmente impressa tornou-se a versão final.

Com a articulação produzida, instalou-se-lhe o *encoder*, o sensor de Hall, e acoplou-se-lhe o motor. Adicionalmente, também se tratou de todo o *hardware* externo necessário ao seu correto funcionamento. Como tal, começou-se pelo sensor de corrente e pelo sensor de tensão, que foram

testados e calibrados individualmente, e integrou-se o sensor de temperatura, o *driver* de motor e o conversor RS-485. Logo após, incorporou-se todo o *hardware*, e correspondente *software*, num só microcontrolador.

Na fase seguinte do processo de desenvolvimento tratou-se do *software*, tendo sido decomposto em 3 partes principais: *Firmware* do Microcontrolador, Protocolo de Comunicação e *Software* do *Master*. O *Firmware* do Microcontrolador está responsável por coordenar a sensorização, os algoritmos de controlo, a actuação do motor e a comunicação. Esta coordenação garante que a articulação consegue comunicar com o exterior, sem romper a periodicidade do algoritmo de controlo. O Protocolo de Comunicação estabelece uma estrutura fixa para as tramas usadas na comunicação, estando responsável por construí-las antes do envio e interpretá-las após a recepção. E por último, o *Software* do *Master* é composto por um conjunto de funções que simplifica o envio de comandos às articulações (que são os *Slaves*).

A última fase deste processo consistiu em desenvolver os algoritmos de controlo. Os controladores implementados na articulação foram: um Controlador PI de Velocidade com Módulo *Feedforward*, um Controlador PD de Posição, e um Controlador em Cascata com Compensação de Zona Morta. Para o primeiro recorreu-se ao Método IMC para realizar a afinação, para o segundo usou-se um Protótipo de Bessel de 2ª Ordem, e para o terceiro usou-se uma combinação dos dois métodos anteriores. Ao longo do processo de afinação seguiu-se um critério de minimização do tempo de subida, da sobre-elongação, e do erro médio em regime estacionário na resposta ao degrau.

Em suma, a partir dos resultados obtidos, é possível concluir que se conseguiu obter uma articulação facilmente reconfigurável, compacta, de baixo custo, com *backlash* reduzido, e com uma relação binário/peso satisfatória. Adicionalmente, também se cumpriu com os requisitos estabelecidos para os controladores de velocidade e posição. Consequentemente, é possível constatar que os requisitos iniciais estabelecidos para este projecto foram cumpridos.

Por último, este trabalho permitiu escrever um artigo para a Conferência [CLAWAR 2021](#) denominado "Design and Modelling of a Modular Robotic Joint". Este artigo, que pode ser consultado no Anexo [B](#), foi submetido e aceite para apresentação oral. Adicionalmente, o artigo [53], que já se encontra publicado na Revista [Applied Sciences](#), também representa uma contribuição feita com esta dissertação, e um possível caminho para a continuação do trabalho desenvolvido. Este artigo permitiu compreender a integração de uma articulação não-rígida.

6.2 Trabalho Futuro

Apesar deste projecto se encontrar concluído, existem ainda potenciais melhorias. Como tal, nesta Secção serão explanados os possíveis trabalhos futuros.

- **Motor** — Numa versão futura o motor poderia ser substituído por uma de duas alternativas. A primeira alternativa seria continuar com um motor escovado de ímanes permanentes, no entanto substituído por outro com uma zona morta menor e com uma relação RPM/V

menor, o que permitiria obter mais sensibilidade no controlo da velocidade. A segunda alternativa seria substituir o motor por um sem escovas, visto que têm tipicamente uma densidade de potência maior, o que levaria à produção de mais binário sem prejuízo do peso da articulação;

- **Encoder Magnético** — Relativamente ao *encoder*, a versão actual que comunica por *I²C* seria trocada pela versão que comunica por *SPI (Serial Peripheral Interface)*. Desta forma a comunicação seria mais rápida e permitiria aumentar a frequência do ciclo de controlo;
- **Caixa Redutora** — O próximo passo para a caixa redutora seria melhorar a sua eficiência, de forma a tirar o melhor proveito possível da sua razão de redução. Para tal conduziriam-se os seguintes passos: realizaria-se um estudo que indicasse qual o melhor lubrificante para engrenagens plásticas; e analisava-se comparativamente qual o tipo de engrenagem que tinha maior eficiência, entre helicoidais, helicoidais duplas e rectas;
- **Redução do Ruído na Velocidade** — Outra melhoria relevante para o projecto seria a redução do ruído associado ao cálculo da velocidade. Para tal, inicialmente modificaria-se a engrenagem que suporta o íman, de forma a minimizar o seu desvio axial relativamente ao *encoder*. Caso esta primeira alternativa não reduzisse o ruído, ou apenas reduzisse de forma pouco significativa, optaria-se também pela implementação de um filtro.

Anexo A

Figuras Adicionais

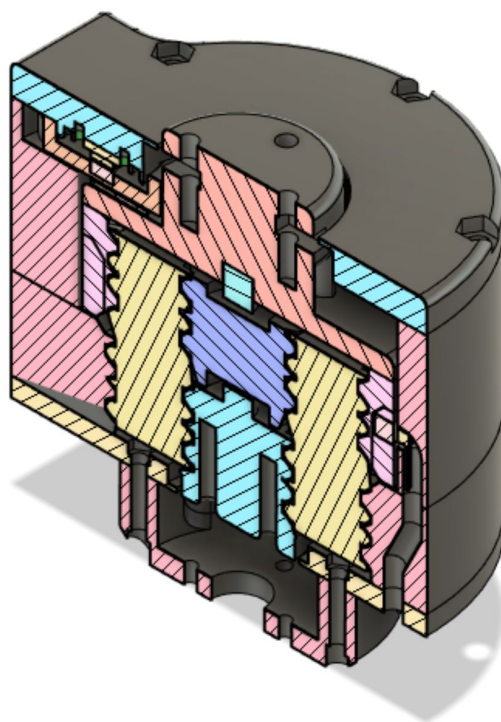


Figura A.1: Vista lateral seccionada da Articulação Modular para Braços Robóticos.



Figura A.2: Componentes impressos da Articulação Modular para Braços Robóticos.



Figura A.3: Articulação Modular para Braços Robóticos - vista lateral frontal.



Figura A.4: Articulação Modular para Braços Robóticos - vista lateral traseira.



Figura A.5: Articulação Modular para Braços Robóticos - vista lateral.



Figura A.6: Articulação Modular para Braços Robóticos - vista frontal.

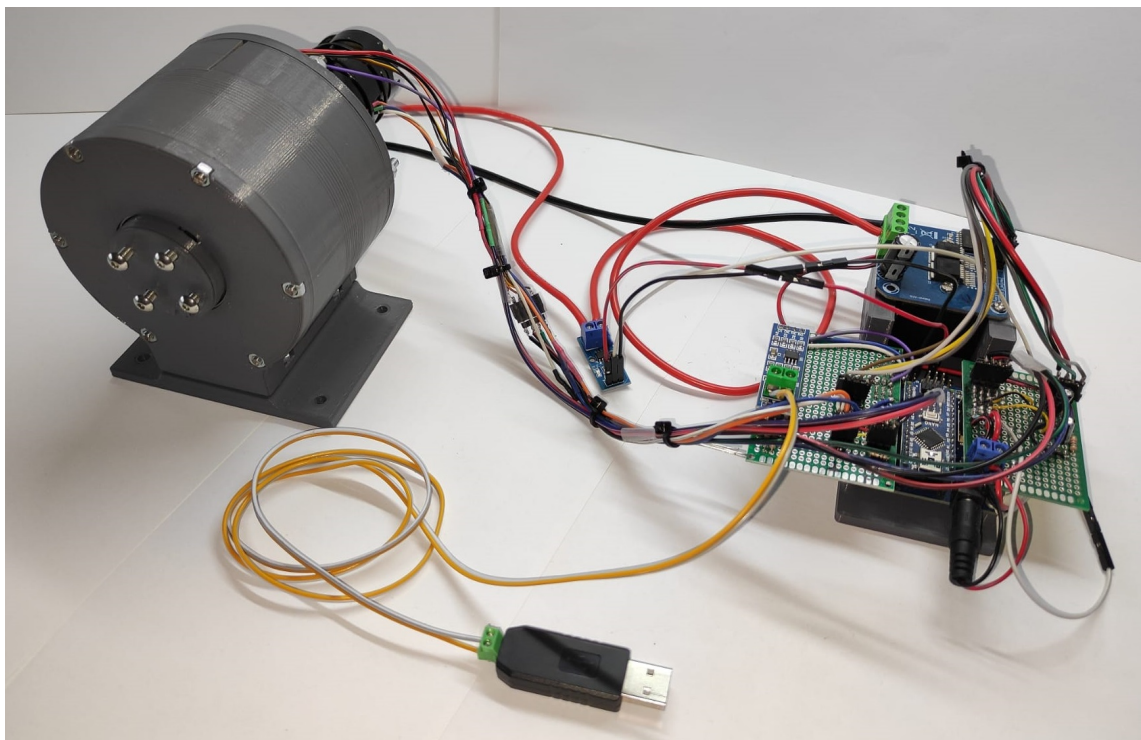


Figura A.7: Sistema completo: Articulação Modular para Braços Robóticos e respectivos componentes electrónicos.

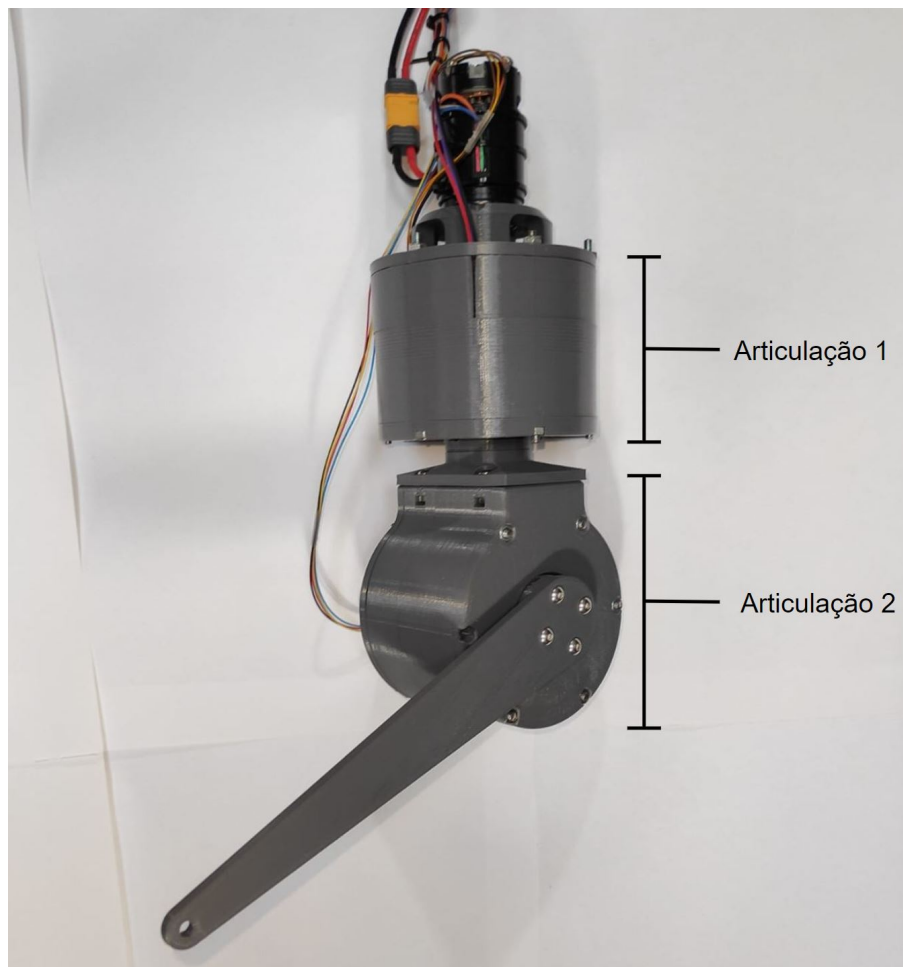


Figura A.8: Exemplo de configuração: Manipulador com dois graus de liberdade/articulações.

Anexo B

Artigo Publicado

O artigo escrito, submetido e aceite na Conferência [CLAWAR 2021](#) encontra-se abaixo anexado.

Design and Modelling of a Modular Robotic Joint

Marco Rocha¹, Vitor H. Pinto^{1,2}, José Lima^{2,3}, and Paulo Costa^{1,2}

¹ Faculty of Engineering, University of Porto

`up201604163@edu.fe.up.pt`

`{vitorpinto, paco}@fe.up.pt`

² INESC TEC—Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal

³ Research Centre in Digitalization and Intelligent Robotics, CeDRI, Polytechnic Institute of Bragança

`jllima@ipb.pt`

Abstract. The industry tends to increasingly automate as many processes as possible, and to make this possible, they often resort to the use of robotic arms. This paper presents the development of a proposal for a modular joint for robotic arms that allows: to obtain the best possible torque/weight ratio; to be controlled in speed and/or position; to communicate with other joints and external microcontrollers; to keep the cost as low as possible; and to be easily reconfigurable. The proposed prototype was validated with real results.

Keywords: Manipulator, Modular Joint, Gearbox, Error Control

1 Introduction

As the variety of processes to be automated in the industry represents a wide range of options, robotic arms must be as flexible as possible in terms of their configuration. Consequently, the modular joints allow to easily configure a robotic arm with the desired characteristics for a given process. In other words, as many joints as the number of degrees of freedom desired can be used, and links between the joints can be placed in order to constitute the manipulator.

Modular joints, as they are intended to have the most flexibility performing tasks, they have to bring together a set of characteristics that allow them to perform at the level of traditional robotic manipulators. However, these joints usually have associated characteristics such as low torque/weight ratio, high backlash and cost, which represent the opposite characteristics to the intended ones. That is, if the joint has to produce more torque, the motor can be replaced by one with more power, or the gearbox can be modified for a higher speed reduction. However, a more powerful motor implies an increase in cost and weight, and a gearbox with a bigger reduction implies an increase in backlash.

If backlash reduction is desired, one of two approaches is usually chosen: design and production of gears according to more precise techniques; or it is

opted for an architecture for the gearbox that implies a lower backlash. However, both options usually involve higher production and projection costs. As such, the projection of a modular joint consists of a trade-off between the torque/weight ratio, backlash and cost. Consequently, for the modular joint proposed, that can be seen in Figure 1, it was opted to use a compound planetary gearbox, since it allows to obtain a high reduction ratio without dramatically increasing backlash.

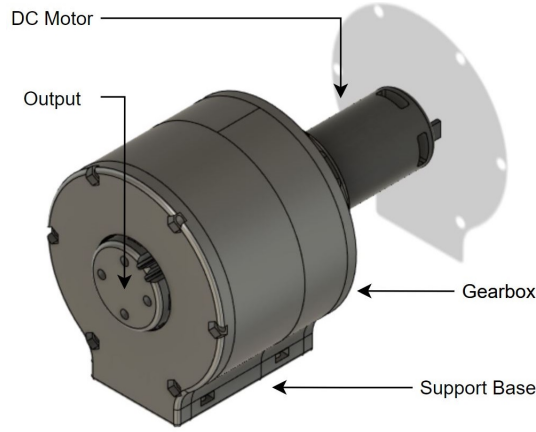


Fig. 1. The Developed Modular Robotic Joint.

Regarding the motor, a brushed DC one was used, which contributes to the low final cost. An 8-bit microcontroller was also incorporated, where the joint control algorithms are executed. In order to make possible to carry out the control safely, the joint has the following sensors: voltage, current, temperature, absolute magnetic encoder, and an Hall sensor to detect the Home Position.

The rest of the paper is organized as follows. After the introduction, section 2 presents the related work and section 3 addresses the system architecture. Then, in section 4 the modular joint design is presented while in section 5 different control methods are addressed. Section 6 presents the results and its discussion whereas last section concludes the paper and points out the future work.

2 Related Work

The study of modular joints is a subject of interest that has been extensively explored both at the academic and industrial levels. In this way, there are already other proposals developed, some of which are simply the result of a study and others that are already commercially available.

Regarding the solutions developed academically, [1] presents an approach composed by: gearbox, motor, encoder, limit switch and accessory components.

In this case, a gearbox of the harmonic type, to which a brushless DC motor and an incremental optical encoder are connected, was used. The controller used is a DSP that receives commands via the CAN protocol. The biggest disadvantage of this joint is its high weight of 7.2 kg and its large footprint.

As mentioned earlier, there are models already commercially available such as [2]. This version has two absolute magnetic encoders, a brushless AC motor and an harmonic gearbox. There are 5 options within this model, the most powerful option, which is also the heaviest, produces 333Nm of peak torque and weighs 3.9 kg. However, the big disadvantage of this option is the high cost: 3370.60 €.

Consequently, the proposal made in this article intends to stand out from those mentioned in the following aspects: low cost, low weight, more compact, and easily reconfigurable.

3 System Architecture

The joint's development started from an initial concept for its operation, according to which the entire system architecture was developed. So, the general architecture can be seen explicitly in Figure 2.

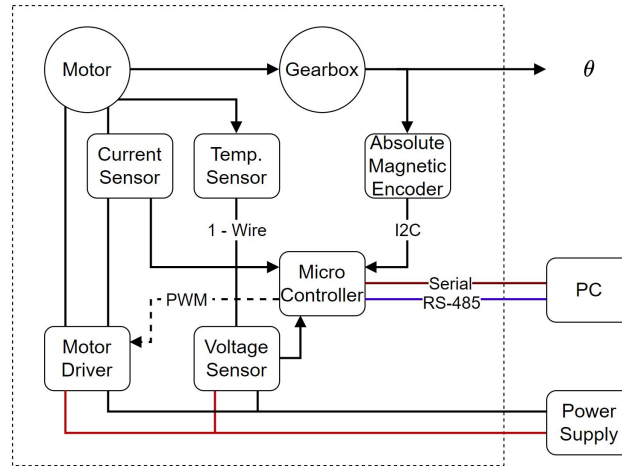


Fig. 2. Developed System Architecture.

This system has an 8-bit microcontroller as its operating brain. It receives through Serial Communication, using RS-485 as a physical layer, commands that represent the control (speed and/or position) that must be carried out. When these commands are received, they have information about the references used for control. Consequently, since the microcontroller carries out all the processing, it receives the data from the absolute magnetic encoder via I2C, which allows it

to be aware of the joint's speed and position. With that, it communicates with the motor driver through PWM inducing the proper motor movement.

In addition, to guarantee the motor's safe operation, a temperature sensor was added, which communicates the motor's temperature to the microcontroller through the 1-Wire protocol. The supply voltage is monitored using a resistive divider and the microcontroller's analog-to-digital converter (ADC). In order to monitor current, a current sensor was also added.

A 12V DC Power Supply is used to supply the system. This way, only four wires are needed to make the joint work: two for communication and two for power supply.

During the development of this joint, the desired specifications were: to have a build cost of less than 150€; to weigh less than 1 kg; to have a backlash less than 0.5°; and to be able to lift 0.5 kg with a 0.5 m extension, which is equivalent to produce 2.45 Nm of torque.

4 Modular Joint Design

The modular joint projection is based on the following principles: low cost, high physical connectivity, easily reconfigurable, low backlash and built-in control. In addition, it had to be able to house all the sensors already mentioned.

Thus, in the following subsections the different parts of the joint construction process and the reasoning that underlies on each one will be explained.

4.1 Modular Joint Mechanical Design

As it is intended to produce as much torque as possible while maintaining the joint compact, it was decided to select a reduction ratio that allowed to maximize the torque production while using a low-cost and low power DC motor. Thus, after investigating the different types of gearboxes, a compound planetary type was developed. This type of gearbox allows to achieve high reduction ratios without significantly hindering its compactness [3].

A compound planetary gearbox basically consists of two different planetary gearbox sets with their planets physically connected. This way, the sun of the first gearbox acts as the input and the ring of the second gearbox as the output. Each set is composed by one ring, one sun and four planets.

In Figure 3 it is possible to see the composition of the developed gearbox. As it can be seen, Ring 1, Planet 1 and Sun 1 are the components related to the first planetary gearbox. Doing the same reasoning, Ring 2, Planet 2 and Sun 2 are the components related to the second one.

Additionally, it is possible to check the Home Position sensor and Magnetic Encoder location. The encoder has an offset relative to the output axis, however it rotates synchronously through a 1:1 ratio.

The reduction ratio N_g of the compound planetary gearbox is calculated as:

$$N_g = \frac{\frac{P1}{R1} - \frac{P2}{R2}}{\frac{P1}{S1} + \frac{P1}{R2}} \quad (1)$$

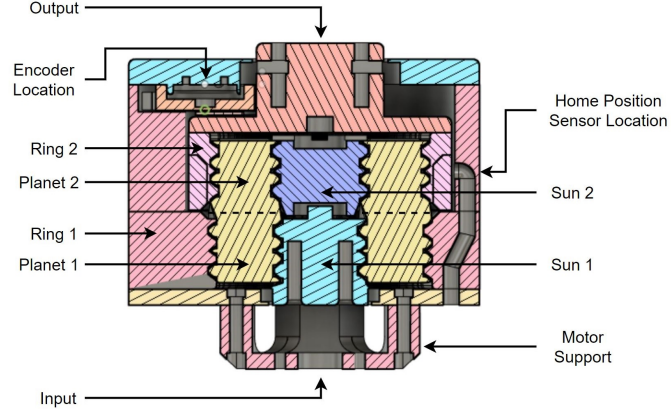


Fig. 3. Compound Planetary Gearbox Sectional View.

In Equation 1, R1 refers to the number of teeth in Ring 1, P2 refers to the number of teeth in Planet 2, and so on. In this case, the different components of the gearbox have as many teeth as those in Table 1. Taking into account these parameters and the Equation 1, the gearbox's reduction ratio is about 75.8:1.

Table 1. Number of Teeth per Gearbox Component.

Component Number of Teeth	
R1	49
P1	15
S1	19
R2	48
P2	14
S2	20

4.2 Modular Joint Production

One of the main goals for this joint was a low final cost. As the production process can bring high added costs, this factor has become relevant to the development process. Consequently, the method chosen to produce this joint, and similar ones, was the additive manufacturing (3D Printing). This method, in addition to being extremely low cost when compared to others, such as metal machining, also allows it to be easily accessible.

Additionally, as the prototyping process is fast and affordable using 3D Printing, it also contributes for the joint to be extremely reconfigurable. That is, if it is necessary to adapt it to a new application, the necessary modifications are quickly made and the new components are printed without adding high costs.

However, during the development process, the selection of this manufacturing method implied that the CAD design was modified in order to account for the lack of tolerances coming from 3D Printing. These could not be too tight because after printing the components do not fit properly, and they could not be too loose because they meant a significant increase in backlash.

After 3D printing the complete joint and attaching the motor and the electronics, the final result from this process can be seen in Figure 4.



Fig. 4. Complete 3D Printed Robotic Joint.

4.3 Modular Joint Connection

In order to maximize the joint's modularity, the fixation fittings have been developed so that it does not restrict any type of positioning. As can be seen in Figure 5, the fixation consists of a rectangular pattern (40 x 43 mm) to which any type of adapter with the same pattern can be built and connected using standard M3 bolts. Consequently, this joint can be placed either vertically, horizontally, or even in any other way that meets the requirements of a specific application.

The output shaft has a square bolt pattern (18 x 18 mm), as can be seen in Figures 1 and 4, to which any type of physical connection, or even another modular joint, can be connected directly. The output shaft also uses standard M3 bolts that allows an adapter to be connected to it according to a given application.

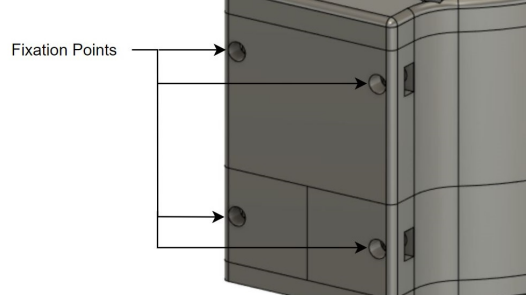


Fig. 5. Modular Joint Fixation Points (Bottom View).

4.4 Modular Joint Electronics

The electronic part of the robotic joint is divided into four different groups: controller, sensorization, communication and motor.

The controller is an 8-bit microcontroller, more specifically an ATmega328P. The choice of this controller made it possible to satisfy the control requirements while maintaining the final cost low. It is responsible for carrying out the speed and/or position control of the joint and for monitoring the system variables. That is, it captures the motor's temperature with a frequency of 1 Hz, and updates the voltage, current, position and speed with a 40 Hz frequency. Since the encoder communicates via I2C, and this is not a very fast protocol, it has been determined through tests that 40 Hz is the fastest possible control frequency that allows to maintain the control period constant.

Regarding sensorization, it has a Maxim Integrated DS18B20 temperature sensor that communicates through the 1-Wire protocol. This sensor has a variable resolution between 9 and 12 bits, and in this case it is programmed to work at 9 bits, which allows to have a resolution of 0.5°C and a fast conversion and reading time. To monitor the supply voltage, a resistive divider converts this voltage into a value within the safe parameters of the microcontroller's ADC. In order to obtain the current consumed by the motor, an ACS712-30A current sensor is used. It generates a voltage proportional to the consumed current that is read by the microcontroller's ADC. Finally, the joint's position reading, which also allows to calculate speed, is made using a AS5600 absolute magnetic encoder with 12-bit resolution.

For the joint to receive commands, the communication was established using RS-485 as a physical layer, due to allowing the use of several joints simultaneously on the same line of communication. The RS-485 transceiver used is a Maxim Integrated MAX485CSA. Using this method, a serial communication is established between the Master, which is a main computer/controller, and the Slaves, which are the joints. The transmitted packet has the Slave address on the first byte, the type of command ordered on the second byte, and the next 4 bytes identify the float-type variable that serves as a reference for joint control.

The commands can specify control in speed, position, or both. The DC motor used is a Surpass Hobby 550-35T, which consists of a brushed motor that holds up to a maximum current of 20 A and supports up to 7.2 V.

5 Joint Control

In addition to having the main goal of building a modular robotic joint, it also intends that this joint has built-in functional controllers. Thus, the implemented controllers are: PD Position Controller, PI Speed Controller with Inverse Dynamic Feedforward, and a Cascade PID Controller with Dead Zone Compensation (DZC). The following subsections will detail each one of the controllers.

The PI Speed Controller was tuned using the Internal Model Control (IMC) method [4] whereas the PD Position Controller was tuned using the second order Bessel prototype.

5.1 PD Position Controller

The PD Position Controller consists, as the name implies, of a controller with a proportional and a derivative component. With that, the notation used to represent the controller is as follows: $e(t)$ is the position error, K_c the proportional gain, T_d the derivation time and as a controller's result is $u(t)$. With this, the controller can mathematically be represented as:

$$u(t) = K_c \left(e(t) + T_d \frac{d}{dt} e(t) \right) \quad (2)$$

Since it is intended to control position, it must be considered that it is a result of the speed's integration. Consequently, the controller can be represented in a block diagram as in Figure 6.

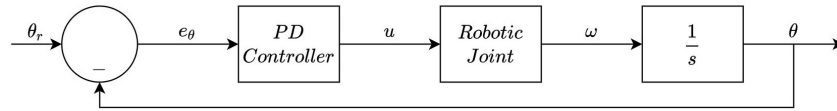


Fig. 6. PD Position Controller represented in a Block Diagram.

5.2 PI Speed Controller with Inverse Dynamic Feedforward

The PI Speed Controller differs from the PD by replacing the derivative component with an integrative one. This way, its mathematical representation uses T_i to represent the integration time, as:

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int e(t) dt \right) \quad (3)$$

An Inverse Dynamic Feedforward (FF) block was added to the PI Speed Controller. This component is the joint's inverse transfer function and acts directly on the controller output. The FF's main goal is to improve the system's response without affecting its stability, by reducing the response time and overshoot [5]. Its interaction with the PI Controller can be represented as in Figure 7.

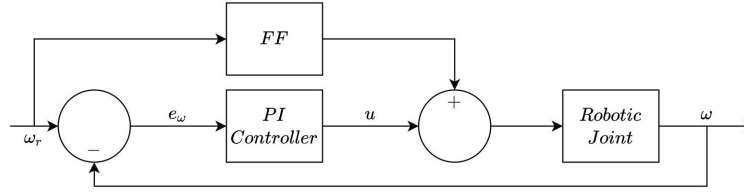


Fig. 7. PI Speed Controller with Feedforward Block represented in a Block Diagram.

5.3 Cascade PID Controller with Dead Zone Compensation

The Cascade Controller differs from the previous ones mainly in the fact that it consists of two controllers connected. More specifically, the first controller, which in this case is a PD, acts directly on the external reference and generates a new reference signal for the next controller, which in this case is a PI. The Cascade Controller has the main advantage of being able to guarantee stable control when load variations occur [6].

A motor's Dead Zone Compensation (DZC) component has been added to the Cascade Controller as can be seen in Figure 8. This component intends to compensate the motor's dead zone as it represents a non-linearity. The DZC component consists on the motor's inverse model. More specifically, knowing the voltage from which the motor leaves the dead zone, it is possible to compensate the output of the controller so that it operates in the linear zone while it has not reached the reference.

Therefore, its implementation is encoded below, where *output* is the real output value of the Cascade Controller, *posError* is the current position error, *minPosError* is the threshold value for the minimum position error and *offset* is the minimum voltage that puts the motor outside the dead zone.

```
if ((abs(posError) > minPosError) and (abs(output) < offset))
    output = (output / abs(output)) * offset;
```

The value of *output* is divided by its absolute value in order to preserve the motor's direction of rotation.

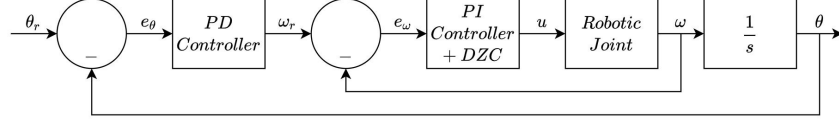


Fig. 8. Cascade Controller with Dead Zone Compensation Block represented in a Block Diagram.

6 Results

As it was initially intended to develop a modular, low-cost joint, with easy reconfiguration and built-in control, the following subsections will present the results about these aspects.

6.1 Architectural Results

After the joint was built and all the components were added to it, the characteristics represented in Table 2 were obtained. Additionally, the joint has a backlash of 0.00453 radians, which is equivalent to 0.26° .

Table 2. Comparison of characteristics.

Joint	Weight (kg)	Footprint (mm)	Cost (€)	Max. Torque (Nm)
Current Proposal	0.890	$100 \times 110 \times 168$	95.13	4.54
Proposal [1]	7.2	$110 \times 110 \times 220.5$	Unknown	80
Proposal [2]	3.9	$115 \times 115 \times 150$	3370.60	333

Despite the difference in torque being significant, the proposal presented is substantially lighter and more affordable than the other existing ones. Consequently, it fills this existing gap.

6.2 Performance Results

The performance results consists on the analysis of the three controller's behavior to a given reference: PD Position Controller, Cascade PID Controller with Dead Zone Compensation, and PI Speed Controller with Inverse Dynamic Feedforward. Thus, Figures 9, 10 and 11 demonstrate the controller's response.

Regarding the steady-state behavior, after several experiments, the PD Controller presented an average absolute error of 0.00416 radians and a maximum



Fig. 9. PD Position Controller response to reference.

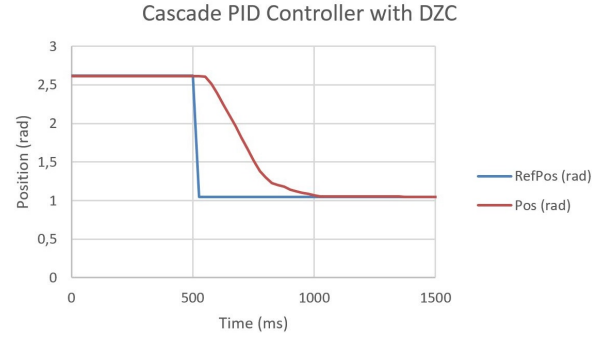


Fig. 10. Cascade PID Controller with Dead Zone Compensation(DZC) response to reference.

absolute error of 0.01 radians. In the same tests, the Cascade Controller had an average absolute error of 0.0035 radians and a maximum absolute error of 0.004 radians. In the transient response, the PD controller had a settling time of 650 ms and the cascade controller of 525 ms. Consequently, we can conclude that the performance of the Cascade Controller with DZC is the best one, given the lower errors and the faster response.

About the PI Speed Controller, after several experiments, it was obtained an average absolute error of 0.006124 rad/s and a maximum absolute error of 0.035 rad/s in steady-state. Regarding its transient response, despite presenting an overshoot of 5.2%, it allows to obtain an average settling-time of 425 ms.

7 Conclusion

In this paper, a new approach on how to build a modular robotic joint was proposed. The final result meets the initial requirements, more specifically, low

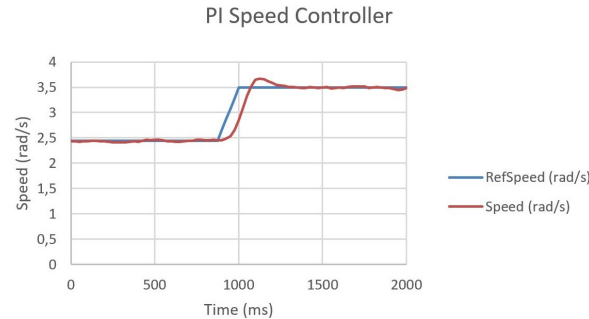


Fig. 11. PI Speed Controller response to reference.

cost, high modularity, easy reconfiguration and control in speed and position incorporated. To develop the prototype, 3D printing was used as a manufacturing method, and the sensors, the actuator and the microcontroller responsible for the control were installed. Results obtained from the dynamic tests validated the proposed prototype.

Future work would go through processes such as: weight reduction, increasing structural rigidity and the use of Hermite polynomials for the control references.

Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UIDB/50014/2020.

References

1. Jia, Q.-X. and Sun, H.-X. and Song, J. and Cheng, T. and Zheng, P.: Development of modular robot joint, 2006 IEEE International Conference on Industrial Informatics, INDIN'06, doi:10.1109/INDIN.2006.275669.
2. Robot Joints RDrive, <https://rozum.com/robotic-joints-by-rozum/>.
3. Crispel, Stein and Lopez Garcia, Pablo and Verstraten, Tom and Convens, Bryan and Saerens, Elias and Vanderborght, Bram and Lefeber, Dirk.: Introducing Compound Planetary Gears (C-PGTs): A Compact Way to Achieve High Gear Ratios for Wearable Robots, pp. 485-489. Springer, 2019.
4. Paul S. Fruehauf, I-Lung Chien, Mark D. Lauritsen: Simplified IMC-PID tuning rules, ISA Transactions, Volume 33, Issue 1, pp. 43-59, 1994, ISSN 0019-0578, [https://doi.org/10.1016/0019-0578\(94\)90035-3](https://doi.org/10.1016/0019-0578(94)90035-3).
5. Kaiser, D.: Fundamentals of servo motion control, Motion System Design, Volume 43, pp. 22-28, 2001, <https://www.scopus.com/inward/record.uri?eid=2-s2.0-17944399301&partnerID=40&md5=7afec604117ae23eac145a7eb3a02429>.
6. William Bolton.: Instrumentation and Control Systems (Second Edition), Chapter 13 - Control Systems, Second Edition, pp. 281-302, 2015, <https://doi.org/10.1016/B978-0-08-100613-9.00013-4>.

Referências

- [1] Q.-X. Jia, H.-X. Sun, J. Song, T. Cheng, e P. Zheng. Development of modular robot joint. *2006 IEEE International Conference on Industrial Informatics, INDIN'06*, páginas 827–832, 2006. doi:10.1109/INDIN.2006.275669.
- [2] L. Fang, M. Luo, J. Chen, e P. Wang. Dual-arm robot modular joint design and error analysis. *2014 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2014*, páginas 1300–1305, 2014. doi:10.1109/ICMA.2014.6885887.
- [3] K. Kim, W. Choi, W. Lee, W. Chung, e S. Kang. Design of joint module equipped with manually configurable reducer for gearing. *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2017*, páginas 597–601, 2017. doi:10.1109/URAI.2017.7992681.
- [4] Hebi Robotics. R Series Actuators. Disponível em <https://docs.hebi.us/hardware.html#r-series-actuators>, acessado a última vez em 28 de Dezembro de 2020.
- [5] Rozum Robotics LLC. Robot Joints RDrive. Drive Your Business Effectively. Disponível em <https://rozum.com/robotic-joints-by-rozum/>, acessado a última vez em 27 de Dezembro de 2020.
- [6] Pavel Melnikov e Svetlana Schegoleva. Methods of increasing the efficiency of the worm gear. *MATEC Web of Conferences*, 298:00001, 2019. doi:https://doi.org/10.1051/mateconf/201929800001.
- [7] A. Mihailidis, E. Athanasopoulos, e E. Okkas. Efficiency of a cycloid reducer. *International Gear Conference 2014: 26th–28th August 2014, Lyon*, páginas 794–803, 2014. doi:https://doi.org/10.1533/9781782421955.784.
- [8] S Kamenev. Building a conceptual model of the chain harmonic drive. *Journal of Physics: Conference Series*, 1333:052008, 10 2019. doi:10.1088/1742-6596/1333/5/052008.
- [9] Charles E. Wilson e J. Peter Sadler. *Kinematics and Dynamics of Machinery*. Pearson Education Limited, terceira edição, 2014.
- [10] H. Harry Asada. *Actuators and Drive Systems*, 2005. Massachusetts Institute of Technology, Department of Mechanical Engineering, Disponível em <https://ocw.mit.edu/courses/mechanical-engineering/2-12-introduction-to-robotics-fall-2005/>, acessado a última vez em 29 de Novembro de 2020.

- [11] Inc. Monolithic Power Systems. Stepper Motors Basics: Types, Uses, and Working Principles, 2020. Disponível em <https://www.monolithicpower.com/stepper-motors-basics-types-uses>, acessado a última vez em 24 de Janeiro de 2021.
- [12] Matjaz Mihelj, Tadej Bajd, Aleš Ude, Jadran Lenarcic, Aleš Stanovnik, Marko Munih, Jure Rejc, e Sebastjan Šlajpah. *Robot Sensors*, páginas 85–105. Springer International, 01 2019.
- [13] Sarah Ruocco. *Robot sensors and transducers*. Springer Netherlands, 01 1988.
- [14] Handson Technology. Bts7960 high current 43a h-bridge motor driver. Disponível em <https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%20Driver.pdf>, acessado a última vez em 3 de Junho de 2021.
- [15] AMS. As5600 12-bit programmable contactless potentiometer operation manual. Disponível em https://www.mouser.com/datasheet/2/588/AS5600_UG000254_2-00-1877354.pdf, acessado a última vez em 1 de Junho de 2021.
- [16] Hebi Robotics. R Series Actuators Datasheet. Disponível em <https://docs.hebi.us/resources/datasheets/R-SeriesDatasheet.pdf>, acessado a última vez em 28 de Dezembro de 2020.
- [17] Hebi Robotics. X Series Actuators. Disponível em <https://www.hebirobotics.com/x-series-smart-actuators>, acessado a última vez em 28 de Dezembro de 2020.
- [18] Stein Crispel, Pablo Lopez Garcia, Tom Verstraten, Bryan Convens, Elias Saerens, Bram Vanderborght, e Dirk Lefeber. *Introducing Compound Planetary Gears (C-PGTs): A Compact Way to Achieve High Gear Ratios for Wearable Robots*. Springer International, 01 2019.
- [19] Pablo Lopez Garcia, Stein Crispel, Elias Saerens, Tom Verstraten, e Dirk Lefeber. Compact Gearboxes for Modern Robotics: A Review. *Frontiers in Robotics and AI*, 7, 08 2020. doi:10.3389/frobt.2020.00103.
- [20] Fang Ki, Daniel Scharfenstein, e Claude Weiss. Motor Handbook, Março 2019. Infineon Technologies AG, disponível em <https://www.infineon.com/>.
- [21] A. Kusko e S. M. Peeran. Definition of the brushless DC motor. Em *Conference Record of the 1988 IEEE Industry Applications Society Annual Meeting*, páginas 20–22 vol.1, 1988. doi:10.1109/IAS.1988.25036.
- [22] F. Khorrami, P. Krishnamurthy, e H. Melkote. *Modeling and Adaptive Nonlinear Control of Electric Motors*. Springer-Verlag Berlin And Heidelberg GMBH & CO., primeira edição, 2003.
- [23] Reston Condit. Stepping Motors Fundamentals, Março 2004. Microchip Technology Inc, disponível em <http://ww1.microchip.com/downloads/en/Appnotes/00907a.pdf>.
- [24] Maria de Fátima de Castro Chouzal. Controlo Digital de Motores Passo-a-passo. Dissertação de mestrado, Faculdade de Engenharia da Universidade do Porto, 1991.
- [25] S. Nkosi, L. Patsika, e L. Masisi. Development of a low cost rotor position sensor. *2020 International SAUPEC/RobMech/PRASA Conference, SAUPEC/RobMech/PRASA 2020*, 2020. doi:10.1109/SAUPEC/RobMech/PRASA48453.2020.9041149.

- [26] Alex Ellin e Gregor Dolsak. The design and application of rotary encoders. *SENSOR REVIEW*, 28(2):150–158, 2008. doi:10.1108/02602280810856723.
- [27] D. Kaiser. Fundamentals of servo motion control. *Motion System Design*, 43(9):22–28, 2001. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-17944399301&partnerID=40&md5=7afec604117ae23eac145a7eb3a02429>.
- [28] Katsuhiko Ogata. *Modern Control Engineering*. Pearson, 01 2009.
- [29] K.J. Åström e Tore Hägglund. *PID controllers: Theory, Design and Tuning*. Instrument Society of America, 01 1995.
- [30] Lucian R. da Silva, Rodolfo C. C. Flesch, e Julio E. Normey-Rico. Analysis of Anti-windup Techniques in PID Control of Processes with Noise Measurement. *IFAC PAPERSONLINE*, 51(4):948–953, 2018. doi:10.1016/j.ifacol.2018.06.100.
- [31] C. Bohn. An Analysis Package Comparing PID Anti-Windup Strategies. *IEEE Control Systems*, 15(2):34–40, 1995. doi:10.1109/37.375281.
- [32] Mark Spong, Seth Hutchinson, e M. Vidyasagar. *Robot Modeling and Control*. Emerald Group Publishing Limited, 09 2006.
- [33] William Bolton. Chapter 13 - Control Systems. Em William Bolton, editor, *Instrumentation and Control Systems (Second Edition)*, páginas 281–302. Newnes, second edition edição, 2015. doi:<https://doi.org/10.1016/B978-0-08-100613-9.00013-4>.
- [34] A. Paulo Moreira. Apontamentos da unidade curricular RIND - Robótica Industrial do Mestrado Integrado em Engenharia Eletrotécnica da FEUP - Joint Control, Maio 2020.
- [35] José Gonçalves, José Lima, e Paulo Gomes Costa. DC Motors Modeling Resorting to a Simple Setup and Estimation Procedure. Em António Paulo Moreira, Aníbal Matos, e Germano Veiga, editores, *CONTROLO'2014 – Proceedings of the 11th Portuguese Conference on Automatic Control*, páginas 441–447. Springer International Publishing, 2015.
- [36] Mohammad Sadraey. *Design of Unmanned Aerial Systems*, páginas 355–398. John Wiley & Sons Ltd, primeira edição, 02 2020.
- [37] A. Malinowski e H. Yu. Comparison of Embedded System Design for Industrial Applications. *IEEE Transactions on Industrial Informatics*, 7(2):244–254, 2011. doi:10.1109/TII.2011.2124466.
- [38] Milan Ponjavic. Microcontroller Selection in Embedded Systems. *Electronics*, 12:21–25, 06 2008.
- [39] Nuno Ferreira Rodrigues. Célula robótica industrial : aplicação de ferramentas CAD/CAM na programação de robôs. Dissertação de mestrado, Faculdade de Engenharia da Universidade do Porto, 2011. URL: <https://hdl.handle.net/10216/61275>.
- [40] Joaquim José Linhares Moreira Regadas. Otimização de uma impressora 3D Delta e desenvolvimento da impressão simultânea de 3 cores. Dissertação de mestrado, Faculdade de Engenharia da Universidade do Porto, 2017. URL: <https://hdl.handle.net/10216/105218>.

- [41] Infineon Technologies AG. Bts7960 high current pn half bridge datasheet. Disponível em <https://www.digikey.com/htmldatasheets/production/70497/0/0/1/bts7960.html>, acessado a última vez em 1 de Junho de 2021.
- [42] AMS. As5600 12-bit programmable contactless potentiometer datasheet. Disponível em https://ams.com/documents/20143/36005/AS5600_DS000365_5-00.pdf, acessado a última vez em 1 de Junho de 2021.
- [43] Allegro Microsystems. Sensitive hall-effect switches for high-temperature operation 3141 thru 3144. Disponível em <https://www.mpja.com/download/a3144eul.pdf>, acessado a última vez em 1 de Junho de 2021.
- [44] Atmel. Atmel atmega328p datasheet. Disponível em https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf, acessado a última vez em 1 de Junho de 2021.
- [45] Allegro Microsystems. Acs712 fully integrated, hall effect-based linear current sensor. Disponível em <https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf>, acessado a última vez em 4 de Junho de 2021.
- [46] Maxim Integrated. Ds18b20 programmable resolution 1-wire digital thermometer. Disponível em <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>, acessado a última vez em 2 de Junho de 2021.
- [47] Maxim Integrated. Low-power, slew-rate-limited rs-485/rs-422 transceivers. Disponível em <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>, acessado a última vez em 2 de Junho de 2021.
- [48] Nick Gamon. Rs485 communications, 2015. Disponível em <https://www.gammon.com.au/forum/?id=11428>, acessado a última vez em 2 de Junho de 2021.
- [49] Mark D. Lauritsen Paul S. Fruehauf, I-Lung Chien. Simplified imc-pid tuning rules. *ISA Transactions*, 33(1):43–59, 1994. doi:[https://doi.org/10.1016/0019-0578\(94\)90035-3](https://doi.org/10.1016/0019-0578(94)90035-3).
- [50] Vítor Pinto, José Gonçalves, e Paulo Costa. *Model of a DC Motor with Worm Gearbox*, páginas 638–647. Springer International, 09 2020. doi:10.1007/978-3-030-58653-9_61.
- [51] The MathWorks. System identification toolbox. Disponível em <https://www.mathworks.com/products/sysid.html>, acessado a última vez em 11 de Junho de 2021.
- [52] M. Hermite e Académie des sciences (France). *Sur Un Nouveau Développement en Série Des Fonctions*. Comptes rendus des séances de l'Académie des sciences. Imprimerie de Gauthier-Villars, 1864.
- [53] Vítor H. Pinto, Inês N. Soares, Marco Rocha, José Lima, José Gonçalves, e Paulo Costa. Design, modeling, and control of an autonomous legged-wheeled hybrid robotic vehicle with non-rigid joints. *Applied Sciences*, 11(13), 2021. URL: <https://www.mdpi.com/2076-3417/11/13/6116>, doi:10.3390/app11136116.