FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de Controlo para o Desenvolvimento de Dispositivos Médicos

Anabela Ribeiro

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Armando Araújo

24 de Outubro de 2020

© Anabela Ribeiro, 2020

Resumo

O envelhecimento populacional é uma realidade em muitos dos países europeus e é o resultado de um conjunto de fatores, como a melhoria das condições de vida e o melhor acesso à saúde. Esta nova realidade é responsável por um acelerado desenvolvimento de produtos e serviços geriátricos e pelo consequente crescimento nas áreas de negócio da sua comercialização. Camas articuladas e elevadores de transferência são exemplos práticos de produtos que oferecem ao paciente e aos seu cuidadores maior conforto e suporte na realização de diversas atividades no quotidiano.

O objeto de estudo desta dissertação foca-se em dispositivos médicos que executam movimentos lineares através da ação de atuadores lineares. A este tipo de dispositivos médicos pode estar associado a execução de um conjunto de movimentos que visam a maximização do bem-estar e do conforto do paciente. A avaliação e os testes destas sequências de movimentos apresentam alguns desafios às equipas de *R&D*, uma vez que, atualmente, o processo de definição e avaliação é manual e inadaptável entre dispositivos médico distintos. Esta avaliação manual limita as diferentes fases de desenvolvimento de um produto, tanto num quadro temporal, como num quadro criativo. É, portanto, objetivo desta dissertação apresentar um sistema configurável e dinâmico com a capacidade de auxiliar as equipas de *R&D* durante o processo de afinação e avaliação de sequência de movimentos.

O sistema de controlo para o desenvolvimento de dispositivos médicos apresenta dois principais focos: o controlo de posição de um atuador linear e o desenvolvimento de uma *interface* gráfica configurável que permite a definição de sequências de movimentos lineares e a execução de testes.

O atuador linear é o elemento do dispositivo médico capaz produzir movimento através da atuação de um motor de corrente contínua (*permanent magnet bushed DC motor*) que aciona um parafuso sem fim. O controlo de posição e de velocidade do atuador linear está associado a dois sensores de efeito *Hall* integrados no motor CC e a um sensor de corrente integrado na unidade de comutação. Por outro lado, a *interface* gráfica desenvolvida permite acionar individualmente cada atuador do protótipo em desenvolvimento, bem como pares de atuadores que realizam movimentos síncronos nesse protótipo. A *interface* permite ainda à equipa de *R&D* configurar a velocidade de cada atuador, ligar/desligar um mecanismo de *soft start stop* e monitorizar cada movimento executado pelo dispositivo médico.

O sistema de controlo e a interface desenvolvida foi integrado num protótipo de uma cama articulada de forma a avaliar o desempenho e as funcionalidades implementadas de suporte à equipa de *R&D*.

Abstract

Population aging is a reality in many European countries and is the result of several factors, such as improved living conditions and better access to healthcare. This new reality became responsible for fast development in geriatric produts for the consequent growth in the business areas of their commercialization. The beds are examples are practical examples of products that offer patients and their caregivers greater comfort and support in carrying out various activities in their daily lives. This manual evaluation limits the different stages of product development, both at a time level and at a creative level. Therefore, it is the objective of this dissertation to present a configurable and dynamic system with the ability to help the *R&D* team during the process of tuning and evaluating the sequence of movements.

The control system for the development of medical devices has two main focuses: the position control of a linear actuator and the development of a configurable interface that allows the definition of sequences of linear movements and the execution of tests.

The linear actuator is the element of the medical device capable create movement the actuation of a direct current motor (permanent magnet brushed DC motor) that drives an endless screw. The position and speed control of the linear actuator is associated with Hall effect sensors integrated into the DC motor and a current sensor integrated into the switching unit. On the other hand, the developed interface allows us to individually activate each driver of the prototype under development, as well as pairs of actuators that perform synchronous movements in this prototype. The interface also allows the *R&D* team to configure the speed of each actuator, to turn on / off a soft start stope mechanism to monitor each movement performed by the medical device.

The control system and the developed interface were integrated in a prototype of an articulated bed in order to evaluate the performance and the implemented functionalities to support the *R&D* team.

Agradecimentos

Em primeiro lugar queria agradecer ao meu orientador, professor Armando Araújo, por todo acompanhamento e colaboração nesta etapa.

À Invacare, na pessoa do doutor Henrique Gonçalves, pela oportunidade e confiança.

À Faculdade de Engenharia da Universidade do Porto e a todos aqueles com quem me cruzei durante os últimos cinco anos que contribuíram para o meu enriquecimento académico e crescimento pessoal.

À minha família, ao meu namorado e aos meus amigos por toda a motivação e apoio.

Anabela Ribeiro

"As always in life, people want a simple answer . . . and it's always wrong."

Susan Greenfield

Conteúdo

1 Introdução **1**
- 1.1 Contexto . . . 1
- 1.2 Motivação . . . 1
- 1.3 Objetivos . . . 2
- 1.4 Estrutura da Dissertação . . . 2

2 Revisão Bibliográfica **5**
- 2.1 Motores Elétricos . . . 5
  - 2.1.1 *Stepper Motor* . . . 6
  - 2.1.2 *Permanent magnet synchronous machine* . . . 8
  - 2.1.3 *Permanent magnet brushed DC motor* . . . 10
  - 2.1.4 *Permanet magnet brushless DC motor* . . . 14
  - 2.1.5 Resumos e Conclusões . . . 16
- 2.2 Dispositivos de Processamento . . . 17
  - 2.2.1 *Field programmable gate arrays* . . . 17
  - 2.2.2 *Microcontroller unit* . . . 17
  - 2.2.3 *Single board computers* . . . 18

3 Caraterização do Problema **21**
- 3.1 Definição do problema . . . 21
- 3.2 Definição dos requisitos do sistema . . . 22
- 3.3 Arquitetura global da solução . . . 23

4 Atuador Linear Elétrico **25**
- 4.1 Princípio de funcionamento . . . 25
- 4.2 Componentes principais . . . 26
  - 4.2.1 Motor elétrico . . . 26
  - 4.2.2 Sensores de fim de curso . . . 27
  - 4.2.3 Caixa de velocidades . . . 27
  - 4.2.4 Parafuso sem fim . . . 27
- 4.3 Princípio de controlo . . . 28
- 4.4 Curvas de caraterização . . . 28
  - 4.4.1 Modelo TA31 . . . 30
  - 4.4.2 Modelo TA23 . . . 31
  - 4.4.3 Modelo LA27 . . . 32
  - 4.4.4 Modelo LA40 . . . 33
  - 4.4.5 Síntese dos atuadores estudados . . . 34

5 Unidade de Controlo **35**
5.1 Estrutura e componentes . . . 36
5.1.1 Unidade de comutação - *H-Bridge* e *Gate Driver* . . . 38
5.1.2 *Microcontroller unit* . . . 40
5.2 Parametrização e configuração inicial . . . 41
5.3 Estratégia para controlo de posição . . . 42
5.4 Estratégia para controlo de velocidade . . . 43
5.5 Modos de operação . . . 44
5.5.1 Considerações gerais da implementação . . . 45
5.5.2 *Inital Setup* . . . 47
5.5.3 *Safe Reset* . . . 48
5.5.4 *Basic Extended / Basic Refracted* . . . 48
5.5.5 *Advanced Extended / Advanced Refracted* . . . 48

6 Comunicação SPI **51**
6.1 Introdução - Princípio de funcionamento . . . 51
6.2 Implementação - Vetores de Comandos . . . 52

7 Unidade Central **55**
7.1 Aspetos Estruturais . . . 55
7.1.1 Unidade de processamento - *Single board computer* . . . 55
7.1.2 Tecnologias . . . 55
7.2 Aspetos Funcionais . . . 57
7.2.1 Controlo individual . . . 59
7.2.2 Controlo global . . . 60
7.2.3 Definição de sequências e testes de repetibilidade . . . 61

8 Resultados - Aplicação Prática **63**
8.1 Apresentação do protótipo da cama articulada . . . 63
8.2 Testes e Resultados . . . 64
8.2.1 Unidade de Comutação - Avaliação Térmica . . . 64
8.2.2 Avaliação do Desempenho . . . 67
8.3 Conclusões e Trabalho Futuro . . . 68

A Estudo do Mercado **71**
A.1 *Texas Instruments* . . . 71
A.2 *Infineon Technologies* . . . 72
A.3 *STMicroeletronics* . . . 73

B *VNH5019A-E: H-Bridge Driver* **75**
B.1 Características principais . . . 75
B.2 Dissipação de calor . . . 75

C Comunicação *SPI* **87**
C.1 Códigos e *masks* . . . 87
C.1.1 Modos . . . 87
C.1.2 *Masks* . . . 87
C.1.3 Comandos . . . 88
C.2 Estrtura geral dos vetores . . . 89

C.3 Algoritmos 90
C.3.1 Unidade de Controlo 90
C.3.2 Unidade Central 91

D Esquemáticos **95**
D.1 Unidade de Controlo 95
D.2 Unidade Central 96

E Estrutura dos Ficheiros **97**
E.1 Estrutura dos Ficheiros *JSON* 97

F Repositórios **99**
F.1 Unidade de Controlo 99
F.2 Unidade Central 99

Referências **101**

Lista de Figuras

2.1 Construções básicas do motor passo-a-passo[1] . . . 7
2.2 Circuito básico de controlo para um motor passo-a-passo de relutância variável[2] 7
2.3 Arranjo unipolar e bipolar dos enrolamentos estatóricos[3] . . . 8
2.4 Construção básica do motor CC de ímanes permanentes com escovas[4] . . . 11
2.5 Configurações para a ligação dos enrolamentos do estator[5] . . . 11
2.6 Circuito básico para o controlo unidirecional[6] . . . 12
2.7 Circuito básico para o controlo bidirecional[6] . . . 12
2.8 Sensores para medição da velocidade rotórica[6] . . . 14
2.9 Medição da tensão *back-emf*[6] . . . 14
2.10 Circuitos Básicos de Comutação[7] . . . 15

3.1 Arquitetura global do sistema proposto . . . 23

4.1 Estrutura típica de um actuador linear elétrico[8] . . . 26
4.2 Individualização de sub-sistemas que compõem um atuador linear elétrico . . . 29

5.1 Unidade de Controlo - Arquitetura . . . 35
5.2 Soluções existentes no mercado . . . 37
5.3 Configuração inicial . . . 41
5.4 Funções básicas implementadas na estratégia para o controlo de posição . . . 43
5.5 Controlo de velocidade . . . 43
5.6 Mecanismo de *soft start and stop* . . . 44
5.7 Modos de operação da unidade de controlo . . . 45
5.8 *Initial Setup* . . . 47
5.9 *Safe Reset* . . . 48
5.10 *Basic Extended/Refracted* . . . 49
5.11 *Advanced Extended/Refracted* . . . 49

6.1 Arquitetura *master-slave*[9] . . . 51

7.1 *Cross-compiler*[10] . . . 56
7.2 Abordagens para o controlo por acesso remoto . . . 57
7.3 Interacções principais da *interface* gráfica . . . 57
7.4 Interacão entre a unidade central e as unidades de controlo . . . 58
7.5 Separador de controlo individual . . . 59
7.6 Separador de controlo global . . . 60
7.7 Separador de definição de sequências . . . 61

8.1 Protótipo da cama articulada . . . 63

8.2 Situação I - Carga 15Ω 65
8.3 Situação II - Carga 15Ω//15Ω 66
8.4 Situação III - Carga 15Ω//15Ω//15Ω 66
8.5 Situação IV - Carga 15Ω//15Ω//15Ω//15Ω 66

B.1 Disposição dos *chips* no integrado[11] 75
B.2 Temperatura *chip* 1 - Situação *I* 77
B.3 Temperatura *chip* 2 - Situação *I* 78
B.4 Temperatura *chip* 3 - Situação *I* 79
B.5 Temperatura *chip* 1 - Situação *II* 80
B.6 Temperatura *chip* 2 - Situação *II* 81
B.7 Temperatura *chip* 3 - Situação *II* 82
B.8 Temperatura *chip* 1 - Situação *III* 83
B.9 Temperatura *chip* 2 - Situação *III* 83
B.10 Temperatura *chip* 3 - Situação *III* 84
B.11 Temperatura *chip* 1 - Situação *IV* 84
B.12 Temperatura *chip* 2 - Situação *IV* 85
B.13 Temperatura *chip* 3 - Situação *IV* 85

C.1 Estrutura dos vetores *TxCommands* 89
C.2 Estrutura dos vetores *RxCommands* 89

D.1 Esquemático Unidade de Controlo 95
D.2 Esquemático Unidade Central 96

Lista de algoritmos

C.1 Mecanismo de interpretação 90
C.2 Mecanismo de construção - Vetor *Communication()* 91
C.3 Mecanismo de construção - Vetor *Update()* 91
C.4 Mecanismo de construção - Vetor *Basic()* 92
C.5 Mecanismo de construção - Vetor *Advanced()* 92
C.6 Mecanismo de construção - Vetor *Setup()* 93

Lista de Tabelas

2.1 Modos de Operação[6] . . . 13
2.2 Analise comparativa de vários motores elétricos[3][12][13] . . . 16
2.3 Tabela comparativa de microcontroladores . . . 18
2.4 Tabela comparativa para dispositivos *SBC* . . . 19

4.1 Curvas de caracterização - Modelo TA31[14] . . . 30
4.2 Curvas de caracterização - Modelo TA23[15] . . . 31
4.3 Curvas de caracterização - Modelo LA27[16][17] . . . 32
4.4 Curvas de caracterização - Modelo LA40[18] . . . 33
4.5 Resumo das características dos quatro modelos em estudo[14][15][16][18] . . . 34

5.1 Conceitos básicos - unidade de controlo . . . 36
5.2 Funcões básicas desempenhadas pelo *IC VNH5019A-E*[11] . . . 38
5.3 Dinâmica térmica - Conclusões . . . 40
5.4 Requisitos para a seleção do microcontrolador . . . 40
5.5 Requisitos da implementação . . . 47

6.1 Descrição dos modos de comunicação . . . 52

8.1 Resumo das caraterísticas do prototipo . . . 64
8.2 Especificações termómetro infravermelho . . . 64
8.3 Resumo das condições de teste . . . 65

A.1 Texas Instruments - Arquitetura da Solução 3 . . . 71
A.2 *Infineon Technologies* - Arquitetura da Solução 1 . . . 72
A.3 *Infineon Technologies* - Arquitetura da Solução 2 . . . 72
A.4 *STMicroeletrocnics* - Arquitetura da Solução 1 . . . 73

B.1 Resumo das características do *IC VNH5019A-E*[11] . . . 75
B.2 Dissipação de calor em regime estacionário [11] . . . 76

C.1 Modos *SPI* . . . 87
C.2 *Masks SPI* . . . 87
C.3 Comandos *SPI* . . . 88

E.1 Estrutura ficheiros *JSON* - *RecordsOfChannels.js* . . . 97
E.2 Estrutura ficheiros *JSON* - *RecordsMoves_x.js* . . . 97

Abreviaturas e Símbolos

R&D	Research and Development
PI	Proportional Integral Control
PID	Proportional Integral Derivative Control
PWM	Pulse Width Modulation
CC	Corrente Contínua
DC	Direct Current
FPGA	Field Programmable Gate Array
MCU	Microcontroller Unit
SBC	Single board Computers
CPU	Central Process Unit
ADC	Analog-to-Digital Converter
FOC	Field Oriented Control
EMF	Electromotive Force
BEMF	Back Electromotive Force
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter
HDMI	High-Definition Multimedia Interface
USB	Universal Serial Bus
CAN	Controller Area Network
RAM	Random Access Memory
OS	Operating System
SRAM	Static Random Access Memory
IC	Integrated Circuit
MOSI	Master Output Slave Input
MISO	Master Input Slave Output
SS	Slave Select
JSON	JavaScript Object Notation
PC	Personal Computer
SSH	Secure Shell
VNC	Virtual Network Computing
PCB	Printed Circuit Board

Capítulo 1

Introdução

1.1 Contexto

Os países desenvolvidos enfrentam hoje em dia uma novo paradigma social: o envelhecimento populacional. Fruto de diversos fatores como o aumento da esperança média de vida, a redução da taxa de mortalidade infantil e o acesso gratuito aos serviços de saúde, a estrutura geracional das sociedades atuais está centrada num novo elemento: o idoso. Desta forma, observamos um crescimento em áreas de negócio de produtos e serviços geriátricos, que procuram dar resposta às necessidades do mercado. Estas necessidades têm-se, à medida do tempo, revelado cada vez mais amplas e exigentes tentando garantir maior conforto e qualidade de vida à população idosa.

Neste contexto, a comercialização de alguns produtos, como é exemplo as camas articuladas, tornou-se cada vez mais complexa englobando um maior número de funcionalidades extras de forma a elevar a qualidade global dos produtos. Assim, o aumento da complexidade traz novos desafios como a necessidade de um maior nível de controlo nos movimentos executados e um novo paradigma de encomendas: singulares e com especificidades adequadas a cada cliente.

1.2 Motivação

As camas articuladas, como outros dispositivos médicos, são desenhadas de forma a auxiliar pessoas com mobilidade reduzida sendo uma importante ferramenta no seu quotidiano proporcionado um maior nível de segurança, conforto e descanso. De forma a assegurar que o bem-estar do paciente seja maximizado, estes produtos realizam uma sequência complexa de movimentos que devem respeitar todas as normas de segurança. Estes movimentos podem ser funções de elevação, articulação e inclinação, por exemplo, sendo realizadas com o auxílio de atuadores lineares, tipicamente constituídos por motores de corrente contínua que accionam um parafuso sem fim.

O desenvolvimento de um dispositivo específico atravessa diferentes estágios de desenvolvimento até atingir o mercado final. Durante o desenvolvimento de cada protótipo de uma cama articulada são definidas e testadas sequências de movimentos de forma a avaliar o conforto e segurança que proporcionam. A fase de definição e avaliação destas sequências é, neste momento, um

processo manual e moroso uma vez que os engenheiros recolhem medições manuais sucessivas até atingirem um nível de conforto e segurança satisfatório. Assim, deste processo manual resulta uma limitação no processo dinâmico de desenvolvimento quer num quadro temporal, quer num nível criativo.

Dada as limitações do processo atual de avaliação e afinação dos movimentos de um dispositivo médico é necessário melhorar este processo, desenvolvendo um sistema de controlo de suporte à equipa de *R&D* que permita acionar e controlar a posição dos diversos atuadores que constituem os protótipo em desenvolvimento de forma a corresponder às maiores exigências do mercado e a desenvolver um produto com maior qualidade e fiabilidade.

1.3 Objetivos

A dissertação pretende atingir dois objetivos principais:

- Controlo de posição e proteção elétrica de atuadores lineares;
- Implementação de uma plataforma que possibilita a configuração do sistema do protótipo em desenvolvimento, a definição de sequências de movimentos e a configuração de testes de repetibilidade.

O primeiro objetivo foca-se no desenvolvimento de um sistema de controlo de posição flexível e expansível aos modelos de atuadores lineares estudados respeitando as normas de segurança. O sistema de controlo deve ser ainda dotado de um mecanismo de controlo de velocidade através da implementação de um mecanismo de *soft start and stop*. O sistema de controlo implementado deve permitir calibrações iniciais para a adaptação das estratégias de controlo aos diferentes modelos e tolerar situações de falhas inesperadas acionando mecanismos de recuperação.

O segundo objectivo direciona-se ao desenvolvimento de uma plataforma com interface gráfica que permita construir uma ponte entre o sistema de controlo e a equipa de *R&D*. Neste sentido, a equipa de *R&D* consegue posicionar cada atuador do dispositivo com pequenos comandos facilitando o processo de definição de movimentos, bem como o processo final de testes de repetibilidade, e por consequência toda a dinâmica de desenvolvimento do produto.

1.4 Estrutura da Dissertação

A estrutura do documento segue uma linha semelhante ao desenvolvimento temporal da solução implementada. Assim, sucintamente,

- No capítulo 2 é apresentada, na primeira secção, uma revisão bibliográfica relativa a aplicação de diferentes motores elétricos em dispositivos médicos. Nesta revisão é apresentada quer o princípio de funcionamento, quer as principais estratégias de controlo adequada a cada tipo de motor estudado. Na segunda secção estão apresentadas algumas unidades de processamento e expostas as principais vantagens e desvantagens de cada uma delas.

- No capítulo 3 é, em primeiro lugar, exposto com mais detalhe o funcionamento atual do processo de definição das sequências de movimentos realizadas pelo dispositivo médico e, seguidamente, é apresentada a arquitetura global da solução implementada que permite corresponder aos requisitos e objetivos desta dissertação.

- No capítulo 4 é descrito o princípio de funcionamento, as principais curvas de caraterísticas e o mecanismo típico de controlo de um atuador elétrico linear. Numa secção final são apresentados os quatros modelos de atuadores lineares estudados.

- No capítulo 5 encontra-se descrito o funcionamento e a arquitetura da unidade de controlo que permite acionar e controlar a posição e a velocidade de um atuador linear, bem como a implementação do mecanismo de *soft start and stop*.

- No capítulo 6 está apresentada a estrutura e a implementação da comunicação *SPI* estabelecida entre o sistema de controlo e a *interface* gráfica de suporte à equipa de *R&D*.

- No capítulo 7 é exposto o desenvolvimento da plataforma com interface gráfica apresentado as suas principais funcionalidade e modos de funcionamento.

- No capítulo 8 é apresentado, primeiro, o modelo do protótipo da cama articulada que serviu de base para a integração e avaliação do sistema desenvolvido, em segundo, os testes realizados à solução implementada e, por último, apresenta-se uma análise crítica aos resultados obtidos.

Capítulo 2

Revisão Bibliográfica

Neste capítulo serão expostas temáticas no âmbito da atuação linear em aplicações médicas. Desta forma, são apresentadas máquinas elétricas recorrentes em projetos e equipamentos de auxílio médico sendo efetuado primeiro um estudo focado em aspetos funcionais e estruturais de cada motor e, de seguida, uma análise de mecanismos de controlo adequados a cada motor apresentado. Numa fase posterior, são apresentados distintos equipamentos electrónicos de processamento tendo em vista o desenvolvimento quer do sistema controlo de posição, quer o desenvolvimento da plataforma para a definição e a avaliação das sequências de movimentos.

2.1 Motores Elétricos

Em aplicações médicas onde a atuação linear é um fator critico, espera-se que o motor elétrico seja capaz de satisfazer níveis de robustez elevados, bem como a capacidade de obedecer a *standards* de segurança e fiabilidade devido às interações diretas e indiretas com humanos e aos diferentes ambientes hostis que é exposto, como a exposição a radiações e altas temperaturas.[19]

Uma máquina elétrica, sucintamente, transforma energia elétrica em energia mecânica quando funciona como motor ou o contrário, funcionamento neste caso como gerador. Tipicamente, uma máquina elétrica é constituída por dois elementos principais: o indutor e o induzido. O indutor é responsável por gerar um campo eletromagnético e o induzido por experimentar fenómenos eletromagnéticos. Um dos elementos é fixo, o estator e o outro móvel, o rotor. A produção de binário devido às forças magnéticas que atuam sobre o rotor de uma máquina elétrica está associada a duas forças diferentes: a força *Lorentz* causada pela atuação de um campo magnético num condutor de corrente quando imerso nesse campo e a força de relutância causada pela alteração da resistência magnética, a chamada relutância, com o objetivo de minimizar a relutância total de um circuito magnético.[3]

Seguidamente serão apresentados os motores tipicamente usados pela indústria médica seguindo as linhas orientadores de [3] para efetuar uma análise comparativa entre eles.

2.1.1 *Stepper Motor*

Os motores passo-a-passo são amplamente utilizados em aplicações de medições e controlo de movimento.[2] Em [20] está descrito a implementação de uma plataforma de monitorização tridimensional com scanners para avaliar o rosto humano no âmbito de tratamentos ortodônticos e moldagens de próteses dentárias em que o movimento do laser está associado a um motor de passo-a-passo. Por outro lado, o desenvolvimento de um modelo *syringe pump* com integração de um motor de passo-a-passo para um desempenho preciso está apresentada em [21].

O motor passo-a-passo pode ser considerado uma versão especial de uma máquina elétrica síncrona.[22] Estruturalmente este tipo de motor é composto por um íman permanente no rotor enquanto que o estator é constituído por um conjunto de enrolamentos excitados por uma determinada sequência de pulsos, funcionamento assim o motor como um atuador digital que converte impulsos elétricos em movimentos mecânicos discretos.[3] Existem três classificações distintas para este tipo de motor[23], a primeira relativa à sua construção: motor passo-a-passo de relutância variável (*variable reluctance stepper motor*), motor passo-a-passo de ímanes permanentes (*permanent magnet stepper motor*) e motor passo-a-passo híbrido (*hybrid synchronous stepper motor*)[3]; a segunda relativa à topologia do circuito de comutação, modo unipolar e modo bipolar[23] e a terceira relativa ao padrão de passo, *hall-step*, *full-step* e *micro-step* [22].

2.1.1.1 Construções Básicos

O motor passo-a-passo de relutância variável apresenta um funcionamento semelhante ao motor de relutância comutada (*switched reluctance machine*) sendo constituído por um rotor laminado com múltiplos dentes e um estator com enrolamentos excitados por corrente contínua. A rotação rotórica ocorre quando os dentes do rotor são atraídos para os polos excitados do estator de forma a que a posição do rotor seja aquela em que o fluxo magnético é mínimo. O estator necessita de pelo menos dois enrolamentos para garantir a mudança de direção.[2][3]

O motor passa-a-passo de ímanes permanentes é constituído por um rotor de íman permanente com uma polarização uniforme ao longo de numa circunferência e enrolamentos controláveis individualmente no estator. O motor produz um binário eletromagnético atraindo os polos do rotor permanentemente magnetizados para os polos do estator.[2][3]

Por fim, o motor passo-a-passo híbrido combina os dois princípios de funcionamento apresentados anteriormente sendo constituído por um rotor formado por um íman permanente cilíndrico em forma de dois discos representativos do polo sul e do polo norte, respetivamente. Por outro lado, os enrolamentos do estator encontram-se enroladas nos polos estatóricos de forma semelhante ao que que acontece num motor de relutância comutada.[2][3] Os polos do rotor alinham-se com os polos do estator dependendo de qual enrolamento está excitado.[22]

De forma sucinta, o motor passo-a-passo de relutância variável é o que apresenta, de entre os três expostos, menor densidade de binário, menor nível de complexidade de funcionamento, menor custo monetário e maior nível de ruído. O motor passa-a-passo de ímanes permanentes,

por sua vez, apresenta uma melhor performance em termos de ruído e vibrações contudo, é considerado o motor de maior custo. Por fim, motor passo-a-passo híbrido mostra-se mais eficiente na capacidade produzir binário e funciona com ângulos de passo inferiores em comparação com os restantes.[2][22][3] Na figura 2.1 podemos avaliar uma comparação relativa à construção dos três tipos de motores, à esquerda, a representação de uma secção transversal de um motor passo-a-passo de relutância variável, ao centro, a estrutura relativa ao motor passa-a-passo de ímanes permanentes e à direita, a representação de um corte transversal do motor passo-a-passo híbrido.

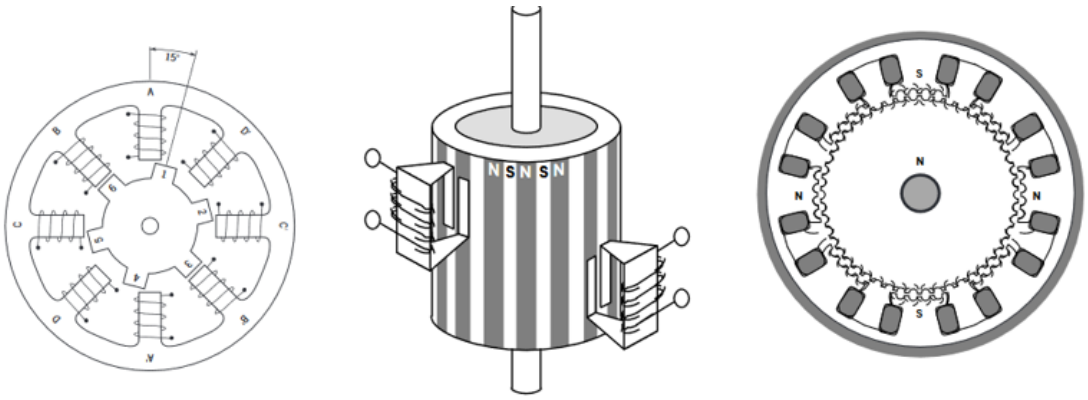


Figura 2.1: Construções básicas do motor passo-a-passo[1]

2.1.1.2 Circuitos Básicos de Controlo

Para motores passo-a-passo de relutância variável não é necessária nenhuma alteração na polaridade da corrente para uma mudança na direção de rotação pelo que o circuito de controlo deste tipo de motor torna-se mais simples e distinto das outras duas construções.[22] Na figura 2.2 está presente o circuito básico comum para este tipo de motor. Os díodos apresentam uma função de proteção, uma vez que no momento da comutação surge um pico de tensão que poderá danificar o transístor.[2]

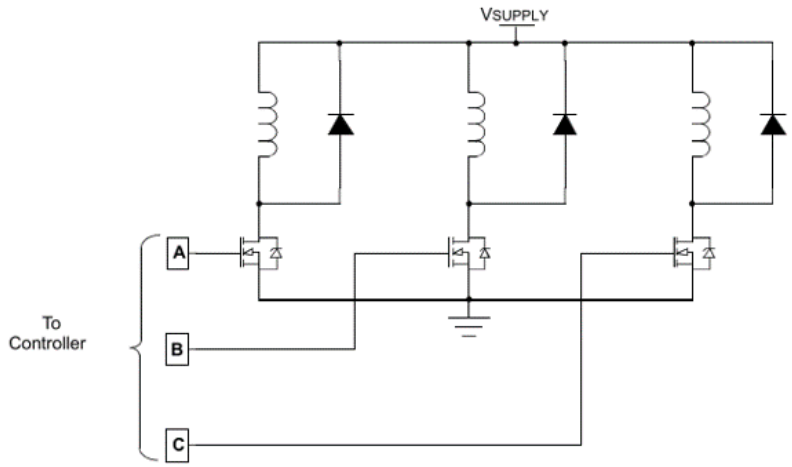


Figura 2.2: Circuito básico de controlo para um motor passo-a-passo de relutância variável[2]

Relativamente às duas outras duas construções do motor existe a necessidade da inversão da polaridade da corrente da alimentação para inversão do movimento. Desta forma, existem dois típicos modos de controlo condicionados pelo arranjo dos enrolamentos, unipolar e bipolar. No modo bipolar é possível obter uma maior eficiência, embora o circuito de comutação do modo unipolar seja mais simples e económico.[22] No modo bipolar existe apenas um enrolamento por fase, sendo este alimentado, tipicamente, por um topologia *H-Bridge* que permite transportar correntes em ambos sentidos de forma a inverter o polo magnético. Pelo contrário, no modo unipolar cada fase possui um par de enrolamentos com uma alimentação central, permitindo que a corrente flua apenas num sentido em cada um dos enrolamento da fase. Desta forma, elimina-se a necessidade de inverter a polaridade da corrente para inverter a polaridade do polo magnético simplificando o circuito de comutação.[2][3] Na figura 2.3 podemos comparar o arranjo dos enrolamentos e o circuito de comutação para cada modo de comutação.

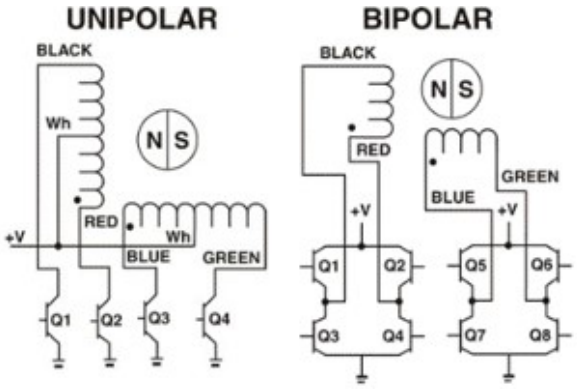


Figura 2.3: Arranjo unipolar e bipolar dos enrolamentos estatóricos[3]

O circuito de comutação pode ainda ser classificado em relação ao padrão de passo: *hall step/wave drive* e *full step*. No primeiro apenas uma fase é excitada de cada vez, ao invés que no segundo duas fases são excitados no mesmo instante originado valores de binário mais elevados neste caso.[2][3]

Existe ainda um outro método para gerar os sinais *PWM* que controlam a comutação dos transístores, é conhecido por *microstepping*.[2] Neste método, ao contrário do acontece com o *half-step* e *full-step*, os enrolamentos são alimentados por correntes sinusoidais desfasadas permitindo a transferência de energia gradual de um enrolamento para outro. Com este método o binário produzido está relacionado com a intensidade da correte de alimentação e permite alcançar resoluções de passo superior e transições de comutação mais suaves que resulta em menos vibrações e fenómenos de ressonância.[2][24]

2.1.2 *Permanent magnet synchronous machine*

Em situações onde o peso e a eficiência do motor são fatores cruciais na escolha do motor, o motor síncrono de ímanes permanentes revela-se a melhor opção.[3] Assim, em [25] está apresentado o desenvolvimento de um novo atuador para uma articulação do joelho para aplicação num

exoesqueleto do membro inferior com recurso a este tipo de motor elétrico. Por outro lado, em [26] está exposto um projeto para desenvolver uma cadeira que produz oscilações e movimentos com o objetivo simular quer o andamento de um automóvel, quer de um avião para estudar o fenómeno de aparecimentos de enjoos nos viajantes. Por fim, uma última aplicação deste tipo de motor é no desenvolvimento de um elevador de transferência de pacientes como visto em [27].

2.1.2.1 Princípio de Funcionamento

O motor síncrono de ímanes permanentes é constituído estruturalmente por um íman permanente no rotor e por um conjunto de enrolamentos no estator. No motor síncrono de ímanes permanentes a velocidade angular do rotor está em sincronismo com a velocidade de rotação do campo magnético indutor do estator. O motor requer correntes sinusoidais no estator para produzir um binário constante.[3]

Existem dois aspetos estruturais que distinguem os motores síncronos entre si: a geometria do rotor (máquina de polos salientes e máquina de polos não salientes) e a dispersão dos enrolamentos no estator (enrolamentos concentrados e enrolamentos distribuídos). Numa máquina de polos não salientes, a indutância do estator não depende da posição rotórica, ao invés que nas máquinas de polos salientes a indutância do estator depende da posição do rotor. Por outro lado, quando o estator é constituído por enrolamentos concentrados, o rácio entre o número de ranhuras do rotor e o número polos é unitário enquanto que quando é constituído por enrolamentos distribuídos, este rácio é superior a um. Uma avaliação comparativa entre as duas disposições dos enrolamentos, concluí que apenas existe sobreposição de enrolamentos na segunda situação e que a primeira requer uma menor quantidade de cobre pelo que torna esta construção mais simples e económica. Contudo, a disposição distribuída é a mais recorrente devido à sua superior *performance* conseguindo produzir um campo no estator quase constante.[3]

2.1.2.2 Mecanismos de Controlo

Para uma máquina síncrona operar em diferentes pontos de binário/velocidade, é necessário ajustar a tensão (ou corrente) e a frequência de alimentação. Seguidamente são apresentados alguns estratégias de controlo que enfatizam aspetos como a fator de potência e o maximização do binário produzido.[3]

O primeiro mecanismo, apresentado em [28], baseia-se na operação da máquina síncrona com um fator de potência unitário permitindo maximizar a utilização da potência aparente do inversor.[3] Este mecanismo revela-se adequado a aplicações que requerem uma velocidade de rotação superior à nominal, contudo em termos de eficiência apresenta uma *performance* deficiente devido às perdas no ferro.[28] Dois outros mecanismos estão apresentados em [29], o primeiro sugere que para uma determinada excitação é produzido o maior binário para a menor quantidade de corrente;[3] o segundo baseia-se na zona de enfraquecimento de campo, onde o controlo é feito pela injeção de correntes negativas de forma a contrariar o fluxo de excitação provocado pelos ímanes permanentes. Destes dois últimos métodos, o primeiro método mostra-se mais eficiente

na maioria das situações, contudo apresenta um pior desempenho em aplicações requerem altas rotações. A principal desvantagem do primeiro método é o reduzido fator de potência devido ao elevado fluxo presente no estator. Por outro lado, a principal desvantagem do segundo é a desmagnetização do íman permanente ao longo do tempo de vida do motor.[3]

Podemos, também, distinguir os métodos de controlo relativamente ao formato das ondas das correntes de alimentação, existe assim o controlo trapezoidal e o controlo sinusoidal. No controlo trapezoidal cada par de enrolamentos é excitado alternadamente para garantir o movimento contínuo do rotor sendo a comutação auxiliada por sensores, por exemplo, os sensores de efeito *Hall*.[30] Embora seja considerado um método simples apresenta como desvantagem: um nível elevado de *ripple* associado ao binário produzido que desencadeia problemas de vibração e ruído.[3] Quanto ao controlo sinusoidal necessita igualmente de informação da posição do rotor, contudo em vez de produzir ondas quadradas, o resultado são ondas sinusoidais desfasadas entre si.[31]. Este método permite uma redução do *ripple*, bem como rotações mais suaves melhorando os níveis de ruído.[3]

O método apresentado em [30] conhecido como controlo vetorial, apresenta-se em comparação com o controlo trapezoidal e sinusoidal bastante vantajoso. Este método oferece níveis superiores de eficiência em todas as gamas de velocidades, bem como melhores *performances* nas respostas transitórias resultantes de mudanças dinâmicas na carga, reduzindo também o *ripple* ao nível do binário produzido.[3] A principal característica do controlo vetorial é a transformação do sistema de correntes trifásico do estator num sistema bifásico ortogonal através das transformadas *Clarle* e *Park*, em que cada componente do sistema bifásico representa, respetivamente, o fluxo e o binário. Para o controlo vetorial existe também a necessidade de conhecimento preciso da posição do rotor que pode ser consiga por *encoders*[3] ou por algoritmos de estimação[30].

2.1.3 *Permanent magnet brushed DC motor*

O motor CC de ímanes permanentes com escovas encontra-se largamente difundidos em diversos formatos e tamanho.[4] Devido à sua simplicidade de operação e controlo, a sua utilização em pequenos *robots* cirúrgicos começa a desenvolver-se.[3] Em [32] está apresentado um estudo sobre monitorização da operação de um braço robótico usado numa cirurgia assistida que permite ao cirurgião avaliar a força aplicada pelo braço robótico no tecido durante uma saturaração.

2.1.3.1 Princípio de Funcionamento

Este tipo de motor é considerado por alguns autores como um dos modelos mais antigos e mais simples de uma máquina elétrica.[5] Estruturalmente é composto por ímanes permanentes no estator e por múltiplos enrolamentos alimentados por tensões contínuas através do comutador e das escovas no rotor. O estator é responsável por gerar um campo magnético constante, ao invés que a excitação dos enrolamentos rotóricos provoca o aparecimento de um campo magnético oposto. Desta forma os polos magnéticos do campo rotórico são atraídos pelos polos opostos do campo gerado no estator originando o movimento de rotação do rotor. A comutação dos enrolamentos,

isto é, a inversão do sentido de rotação, é feita mecanicamente invertendo a tensão de alimentação. Por outro lado, a velocidade de rotação é proporcional à tensão aplicada, da mesma forma que o binário é proporcional à corrente.[3][5][6]

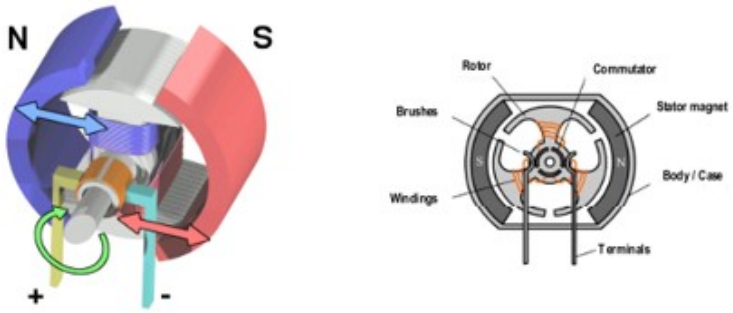


Figura 2.4: Construção básica do motor CC de ímanes permanentes com escovas[4]

Existem três classificações distintas relativa à forma como os enrolamentos estatórios estão conectados à fonte de alimentação, excitação paralela, excitação em série e uma combinação das duas. Na primeira, figura 2.5 à esquerda, os enrolamentos são colocados paralelamente com o rotor através das escovas, desta forma conseguimos uma independência da corrente que flui nos enrolamentos face à corrente rotórica. Esta combinação permite que o motor opere com uma velocidade independente das mudanças da carga permitindo um excelente controlo através da corrente. Relativamente, à excitação em série (figura 2.5 ao centro) os enrolamentos estatóricos encontram-se em série com o rotor, esta configuração permite que a corrente aumente proporcionalmente com a carga sendo bastante útil em aplicações que requerem operações com altos binários. A combinação destas duas configurações (figura 2.5 à direita) permite alcançar melhor *performances* em motores de alto binário com controlo através da velocidade.[3][6]

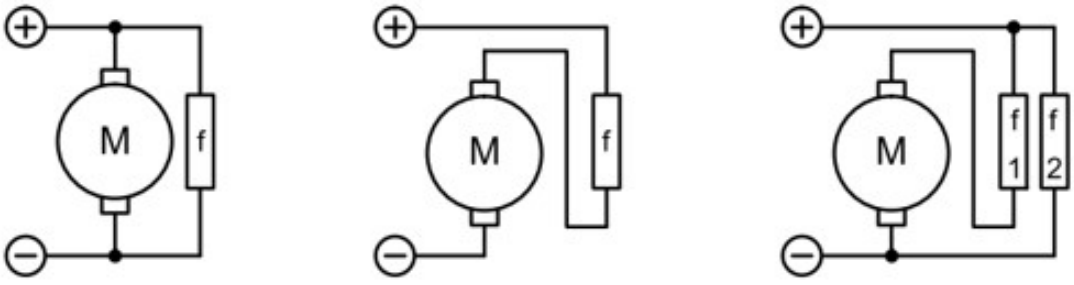


Figura 2.5: Configurações para a ligação dos enrolamentos do estator[5]

2.1.3.2 Circuitos Básicos de Comutação

Existem aplicações que requerem apenas que o motor opere apenas num sentido de rotação, enquanto que outras requerem que o movimento do motor se inverta.

Para a primeira situação estão expostos na figura 2.6 dois circuitos básicos de comutação, à esquerda *low-side driver* e à direita *hight-side driver*, que permitem o controlo unidirecional do motor. Em cada um dos circuitos está presente um díodo com uma função de proteção do *transístor* face ao pico de tensão inversa que assistimos no momento da comutação.[6]

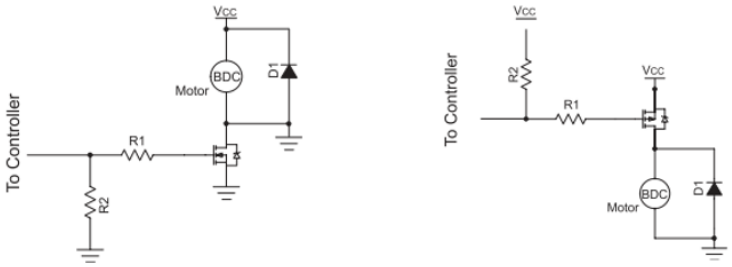


Figura 2.6: Circuito básico para o controlo unidirecional[6]

Por outro lado, o controlo bidirecional carateriza-se pela inversão do sentido da corrente com recurso a uma configuração *H-Bridge*, como na figura 2.7. Cada meia ponte é capaz de comutar um lado do motor ou para a tensão de alimentação ou para o valor de *ground*. Na tabela 2.1 estão avaliados o estado dos transístores para os diferentes modos de operação que o circuito consegue impor no funcionamento do motor. No modo *Coast* as extremidades dos motores são deixadas a flutuar o que levará ao fim do movimento de forma gradual. Pelo contrário, no modo *Brake* todos os enrolamentos são conectados ao *ground* obrigando o motor a parar rapidamente.[3][5][6] Neste circuitos os díodos apresentam a mesma função de proteção enquanto que os condensadores pretendem reduzir a radiação electromagnética produzida pela comutação.[6]

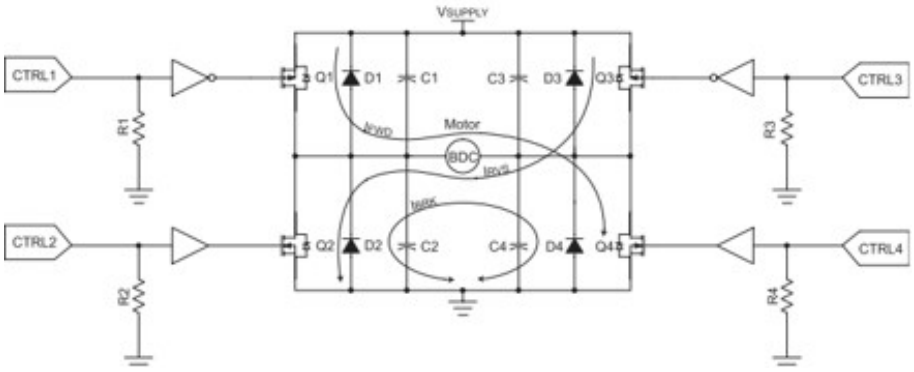


Figura 2.7: Circuito básico para o controlo bidirecional[6]

2.1.3.3 Mecanismos de Controlo

Em pequenas aplicações em que o motor opera sempre à mesma velocidade, a alimentação pode ser conseguida apenas com uma fonte de alimentação associada a um potenciómetro que regula a velocidade de rotação. Contudo, em aplicações em que o motor é um elemento integrante de um sistema global outras técnicas mais sofisticadas de controlo são necessárias.[5]

	Q1 (CTRL1)	*Q2 (CTRL2)*	*Q3 (CTRL3)*	*Q4 (CTRL4)*
Forward	on	off	off	on
Reverse	off	on	on	off
Coast	off	off	off	off
Brake	off	on	off	on

Tabela 2.1: Modos de Operação[6]

Como vimos, neste tipo de motores a velocidade de rotação é proporcional à tensão aplicada, desta forma, utilizando uma técnica de controlo digital, *PWM*, é possível gerar um sinal com um valor de tensão média. O enrolamento funciona com filtro passa-baixo para a onda *PWM* gerada com uma frequência capaz produzir uma corrente estável no enrolamento.[6] Esta técnica garante que a velocidade de rotação é proporcional ao *duty-cycle* da forma de onde gerada. A definição da frequência de operação é um parâmetro limitador da eficiência deste método, na medida em que a operação em baixas frequências produz um maior nível de ruído e uma resposta transitória mais lenta. Por outro lado, a operação em altas frequências resulta num aumento de perdas de comutação fazendo diminuir a eficiência de todo o sistema.[4][3][6]. Os autores de [6] defendem que a frequência deve manter-se numa gama de intervalos entre os 4kHz e os 20kHz.

Embora a velocidade deste tipo de motores seja geralmente proporcional ao *duty-cycle* da onda *PWM*, sabemos que nenhum motor é ideal, isto é, o calor, o desgaste do comutador e a própria carga afetam o desempenho do motor. Em sistemas onde o controle preciso da velocidade é necessário, é uma prática incluir algum tipo de mecanismo de realimentação no sistema de controlo.[6] Os mecanismos de realimentação podem ser divididos em dois grandes grupos, embora em ambos a realimentação seja conseguida através da velocidade rotórica. O primeiro grupo envolve a medição da velocidade através de algum tipo de sensores, enquanto que o segundo utiliza envolve a medição da tensão *BEMF*.[4][6] Relativamente ao primeiro grupo, os sensores de efeito *Hall* e os *encoders* óticos são os sensores mais comuns para medir a velocidade de rotação do motor. Os sensores de efeito *Hall* são formados por um elemento rotativo que contém um íman à superfície no motor e por um elemento estático que deteta quando o elemento rotativo passa por ele (figura 2.8 à esquerda). Relativamente aos *encoders* óticos (figura 2.8 à direita) são compostos igualmente por dois elementos, um elemento rotativo contém uma fonte de luz e outro elemento estático que deteta a passagem do feixe.

A tensão *back-EMF* é uma força eletromotriz que ocorre em máquinas elétricas onde existe um movimento relativo entre o íman do rotor e outro campo externo, ou seja, existe um momento em que a máquina funciona como gerador produzindo uma tensão proporcional à velocidade de rotação do rotor. Na imagem 2.9 podemos observar os dois locais no circuito de comutação onde se deve proceder a essa medição.[4]

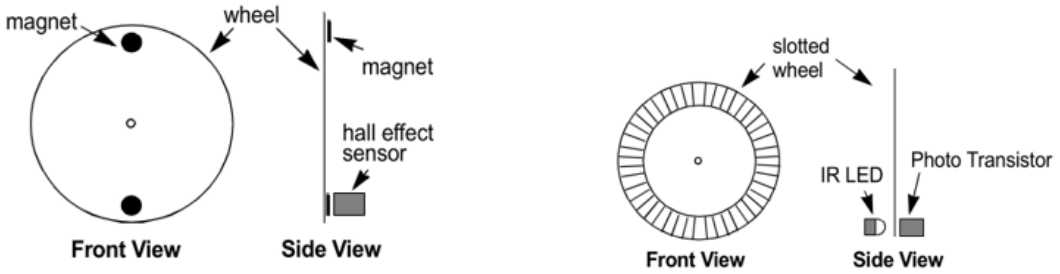


Figura 2.8: Sensores para medição da velocidade rotórica[6]

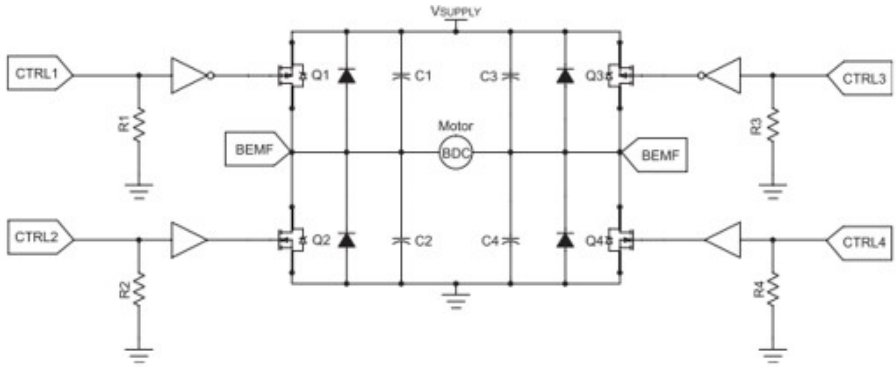


Figura 2.9: Medição da tensão *back-emf* [6]

2.1.4 *Permanet magnet brushless DC motor*

O motor CC de ímanes permanentes sem escovas devido à relação entre o binário produzido e a dimensão e peso deste motor apresenta-se vantajoso em aplicações onde o peso e o espaço são fatores críticos.[28] Em [33] está exposta o controlo deste tipo de motor numa aplicação numa cirurgia minimamente invasiva *(Minimally Invasive Surgery)*. O braço robótico foi desenhado para realizar pequenas incisões em pequenas cirurgias tendo a sua atuação do braço associada a este tipo de motor. Por outro lado, em [34] está descrito e avaliado a *performance* de uma endoscopia por cápsula com um sistema de locomoção atuado por este tipo de motores.

2.1.4.1 Princípio de Funcionamento

Neste tipo de máquinas o rotor é constituído por um íman permanente com um número de par de polos que pode variar entre dois e oito enquanto o estator é constituído por diversos enrolamentos.[35] Os ímanes permanentes criam um fluxo magnético do rotor enquanto que os enrolamentos estatóricos excitados criam polos eletromagnéticos. O rotor é atraído pela fase de estator excitada e, portanto, usando uma sequência apropriada para excitar as fases do estator é criado um campo rotativo no estator.[36] O fluxo então gerado no estator interage com o fluxo do rotor produzindo binário, ou seja, produzindo o movimento rotativo do motor.[37]

O número de enrolamentos no estator origina uma classificação para distinguir grupos deste tipo de motores: motores monofásicos, com apenas um enrolamento estatórico; motores bifásicos

com dois enrolamentos e motores trifásicos com três enrolamentos no estator.[7] A configuração trifásica é mais comum devido ao bom compromisso entre custo, binário e volume.[36] Os enrolamentos numa configuração trifásica podem estar conectados em forma de estrela ou triângulo.[38]

2.1.4.2 Circuitos Básicos de Comutação

Os motores CC de ímanes permanentes sem escovas utilizam interruptores eletrónicos para realizarem a comutação. Assim, na figura 2.10 à esquerda está representado o circuito tipicamente usado para acionar um motor com uma confirmação monofásica e à direita uma configuração trifásica. No primeiro caso, este circuito básico é conhecido como *H-Bridge* e no segundo *Three-Phase Bridge*.[36]

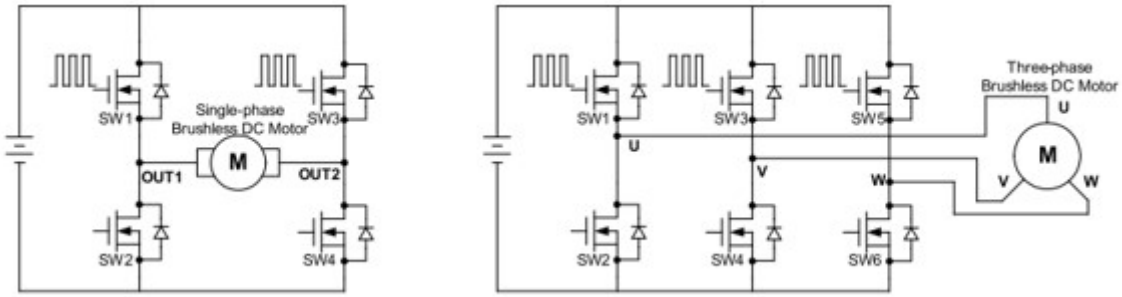


Figura 2.10: Circuitos Básicos de Comutação[7]

O controlo deste tipo de motores está dividido em duas operações independentes: sincronização do fluxo do rotor e do estator (trata-se de uma máquina síncrona) e controlo do valor da corrente de alimentação. A primeira operação requer informação sobre a posição do rotor uma vez que é assegurada pela rotação rotórica, enquanto que a segunda recorre a técnicas de *PWM* para controlar a velocidade de rotação.[36][37]

Dada a necessidade de conhecer a posição do rotor, existem duas abordagens distintas para o fazer. A forma mais precisa de detetar a posição é através de sensores, sendo os sensores de efeito *Hall* os mais comuns. Em [7] está apresentada esta técnica utilizando sensores de efeito *Hall* quer para configurações monofásicas, quer trifásicas. Contudo existem aplicações onde a inclusão de sensores de posição no rotor não é possível recorrendo-se a outra técnica através da medição da tensão *back-EMF*. Em [39] está descrito como esta técnica pode ser implementada.

2.1.4.3 Estratégias de Comutação

O controlo trapezoidal é um algoritmo mais simples quando comparado ao controlo sinusoidal e ao controlo vetorial. É constituído por 6 passos de comutação no caso da configuração trifásica, permitindo que, a cada instante, ocorra um fluxo de corrente entre um par de enrolamentos enquanto que o terceiro enrolamento não se encontra alimentado.[3]

O controlo sinusoidal permite fornecer correntes sinusoidais ao três enrolamento reduzindo o *ripple* e melhorando a *performance* durante a comutação quando comparado com o controlo

trapezoidal. Contudo este método requer uma resolução maior para determinar a posição rotórica sendo conseguida por *encoders* óticos e magnéticos que acarretam um custo superior no sistema de controlo.[3]

Field Oriented Control, conhecido também como controlo vetorial permite alcançar melhores níveis de eficiência em altas velocidades quando comparado com os dois métodos anteriores. Este método controla as correntes estatóricas assumindo uma representação vetorial destas e permitindo um controlo independente do fluxo e do binário. Em [22] esta descrito uma implementação deste método.

2.1.5 Resumos e Conclusões

Cada um dos quatros motores apresentados nos pontos anteriores contém características vantajosos em determinadas aplicações bem como inconvenientes que podem funcionar como factores eliminatórios para uma possível seleção.

	Motor passo-a-passo	**Motor de Ímanes Permanentes**		
		síncrono	**com escovas**	**sem escovas**
Velocidade nominal	-2	2	-1	1
Velocidade Máxima	-1	1	0	2
Binário nominal	2	1	-1	0
Densidade de binário	1	2	0	0
***Ripple* do binário**	-1	2	0	1
Densidade de potência	0	2	-1	1
Eficiência	-1	2	0	1
Capacidade sobrecarga	-1	0	1	0
Preço	2	-1	1	0
Tempo de vida	1	2	-2	2
Robustez	0	1	0	2
Baixa manutenção	1	2	-1	1
Facilidade de controlo	1	-1	2	0
Baixo ruído	-2	2	0	-1

Tabela 2.2: Analise comparativa de vários motores elétricos[3][12][13]

De forma sucinta, os motores passo-a-passo (secção 2.1.1) apresentam como principal vantagem a simplicidade de controlo e a elevada relação entre o binário produzido e o volume do motor (densidade de binário) mas de entre todos os apresentados é o que apresentam o pior nível de ruído e vibrações e uma pobre eficiência. Por outra lado, os motores síncronos de ímanes permanentes (secção 2.1.2) embora apresentem uma melhor *performance* em geral com reduzido *ripple* e alta eficiência apresentam um elevado custo monetário devido à elevada complexidade de construção fazendo deste motor pouco popular para pequenas aplicações na indústria médica. Relativamente ao motor de íman permanente com escovas (secção 2.1.3) apresenta como principal vantagem a facilidade de controlo devido a proporcionalidade da velocidade e binário face à tensão e correntes aplicadas. Por outro lado, devido à comutação mecânica entre os comutadores e as escovas

este motor apresenta grande necessidade de manutenção devido ao desgaste das escovas. Por sua vez, os motores de ímanes permanentes sem escovas (secção 2.1.4) apresentam um bom desempenho em operações contínuas que requerem rotações altas, sendo a sua grande desvantagem a necessidade de determição da posição rotórica para alcançar uma ótima comutação.[13][40][19]

De forma a sintetizar as características de cada motor, a tabela 2.2 apresenta a *performance* de cada motor em diferentes aspetos associada a uma escala de numérica de -2 a 2, em que -2 representa o pior desempenho e 2 o melhor desempenho.

2.2 Dispositivos de Processamento

Atualmente, na grande maioria das aplicações, o motor, é mais um elemento do sistema que deve colaborar no sentido concretizar o objetivo final do sistema. Assim, o motor e todo sistema de controlo devem contribuir para a eficiência final do sistema global. Desta forma, e em conjunto com o aumento do poder computacional encontramos várias soluções para a implementação de uma plataforma de controlo que passam pela integração de FPGAs, microcontroladores, microprocessadores ou pequenos circuitos integrados.[41]

2.2.1 *Field programmable gate arrays*

FPGAs são dispositivos semicondutores que se baseiam numa matriz de blocos lógicos configuráveis conectados por canais programáveis permitindo que a grande maioria das funcionalidades elétricas possam ser alteradas. Estes dispositivos caracterizam-se por proporcionar ao sistema um nível superior de flexibilidade e integração.[42] Desta forma, esta solução atende a alguns requisitos críticos como o controlo mais preciso, a escalabilidade dos sistemas e segurança de operação em ambientes hostis.[43]

Os autores de [44] apresentam um sistema de controlo baseado no algoritmo *FOC* para um motor síncrono e ímanes permanentes com a integração de um dispositivo *FPGA*. Por outro lado, está disponível no mercado um *kit* mais genérico, *SmartFusion2 Dual-Axis Motor Control Starter Kit*[43], que permite o controlo de três diferentes tipos de motores através de duas técnicas distintas (*FOC* e *microstepping*). Este *kit* integra uma *FPGA* em conjunto com interfaces de comunicação e mecanismos de segurança para a deteção de sobrecarga, por exemplo.

2.2.2 *Microcontroller unit*

O mercado de microcontroladores é um mercado vasto e diversificado. Assim, para garantir o melhor compromisso entre custo, funcionalidades e fiabilidade, para a escolha do microcontrolador devem ser considerados alguns critérios como o desempenho do CPU, a memória, o número de pinos de I/O, os canais para conversão *ADC* e os mecanismos de comunicação.[41]

Em [43] está sintetizada uma lista de um grande fabricante mundial neste campo, *Microchip Technology* que sugere para cada tipo de motor e para cada método de controlo uma família de microcontroladores que melhor satisfaz essas necessidades. Seguidamente, vão ser apresentados

alguns modelos de diversos fabricantes de forma exemplificar o diversificado mercado que está disponível. Em [1] é apresentado uma solução baseada no microcontrolador *dsPIC30F3011*, um microcontrolador de 16 bits, enquanto que em [23] a solução baseiam-se no microcontrolador *C8051F300* (microcontrolador de 8 bits comercializado pelo fabricante *Silicon Laboratories*) para implementar, em ambos, um mecanismo de controlo para um motor passo-a-passo. Por sua vez, uma solução para controlo de um motor de ímanes permanentes sem escovas utilizando um microcontrolador *EFM32* (um microcontrolador de 32 bits do fabricante *Silicon Laboratories*) está presente em [38], enquanto que o controlo baseado no dispositivo *PIC16F877* (microcontrolador de 8 bits da família *PIC*) está presente em [39]. Relativamente aos motores de ímanes permanentes com escovas, a título de exemplo está apresentado em [4] uma solução baseado no microcontrolador *LPC2101* (microcontrolador de 32 bits comercializado pela *Philips*) e em [6] uma solução genérica para microcontroladores da família *PIC* (comercializada pelo fabricante *Microchip Technology*).

Na tabela 2.3 estão avaliados alguns modelos de microcontroladores sugeridos pelos fabricantes para integração num sistema de controlo de motores. A tabela compara, além de características de processamento do microcontrolador (memória e velocidade), a capacidade gerar ondas *PWM*, as interfaces de comunicação que suporta (*I2C/SPI/UART/CAN*) e a resolução dos canais do conversor *ADC*.

	ATMEGA328P	**PIC16F877**	**C8051F300**	**PIC30F3011**	**PIC16F684**	**LPC2101**	**ATtiny417**	**ATtiny817**	**STM8S207**
Size (bits)	8	8	8	16	8	16/32	8	82	8
Pins I/O	23/32	40	8	40/44	14	31	24	24	32
Flash (KB)	32	8	8	24	3.5	8/16/32	4	8	128
SRAM(Bytes)	2	368	0.25	1024	128	2/4/8	256	512	6
Max Clock (MHz)	20	20	25	20	32	70	20	20	24
ADC	6x10-bit	8x10-bit	8x8-bit	9x10-bit	4x10-bit	8x10-bit	12x10-bit	12x10-bit	16x10-bit
SPI/I2C/UART	S/S/S	S/S/S	N/S/S	S/S/S	N/N/N	S/S/S	S/S/S	S/S/S	S/S/S
CAN	N	N	N	N	N	N	N	N	S
Timers	2x8-bit 1x16-bit	2x8-bit 1x16-bit	3x16-bit	5x16-bit	2x8-bit 1x16-bit	2x16-bit 2x32-bit	6x16-bit	1x8-bit 2x16-bit	2x16-bit
PWM Channels	6	2	3	6	8	4	6	6	3
Preço(EUR/Unid)	1,7	7,44	2,96	4,41	2,14	3,9	0,667	8,27	2,34

Tabela 2.3: Tabela comparativa de microcontroladores

2.2.3 *Single board computers*

Este tipo de dispositivos surgiram neste âmbito como uma forma resolver alguns problemas que os microcontroladores apresentam em situações mais exigentes. O espaço restrito de memória, a programação em linguagem *assembly* e um tempo de resposta em algumas situações lento para aplicações de *feedback* são alguns problemas podem ser apontados aos microcontroladores. Os *single board computers* caracterizam-se por serem pequenos dispositivos de computação que aglomeram em uma única placa um ou mais microprocessadores, memória e interfaces *I/O*. Esta solução quando comparado aos grandes computadores proporciona um compromisso de computação com baixo consumo e uma arquitetura mais simples.[45]

Em [46] está apresentado uma aplicação que requer a recolha em tempo real de informação para a integração no sistema de controlo não linear de um motor CC com escovas baseado

no equipamento comercializado pela *Texas Instruments*, o dispositivo *Beaglebone Black*. Este equipamento é baseado num processador *ARM Cortex-A8* e suporta sistemas operativos como o *Ubuntu* e *Android*. Por outro lado, em [45] está apresentado uma plataforma de controlo baseado no dispositivo *Raspberry Pi 3* para aplicação em um motor que se caracteriza por operar a altas velocidades. Este dispositivo baseia-se no microprocessador *Cortex-A53* e é combatível com o sistema operativo *Linux*, embora exista um sistema operativo baseado em *Linux* desenhado especialmente para desenvolvimento de ferramentas através deste equipamento.

Sucintamente a grande vantagem da *Raspberry Pi 3* deve-se ao diversificado suporte e documentação existente e a facilidade de instalação e desenvolvimento, ao invés que o *Beaglebone Black* permite alcançar um sistema com maior grau de flexibilidade na medida que possui um maior número de compatibilidades com diversos protocolos de comunicação e sistemas operativos. A tabela 2.4 avalia sucintamente alguns aspectos construtivos e funcionalidades dos dois equipamentos.

	Raspberry Pi 3 Model B+	**Beaglebone Black**
Size	64bit	32bits
Clock Speed	1.4GHz	1GHz
RAM	1GB	512MB
PWM	1 channel	3 channels
ADC	-	8x12bits
Interfaces	UART/SPI/I2C/CAN/USB Ehternet/HDMI/Bluetooth	UART/SPI/I2C/CAN/USB Ehternet/HDMI
SO	Raspbiann	Linux/Android/Windows Embedded
Preço (EUR/Unid)	29,33	53,16

Tabela 2.4: Tabela comparativa para dispositivos *SBC*

Capítulo 3

Caraterização do Problema

No presente capítulo pretende-se apresentar o processo atual de definição e afinação dos movimentos executados pelos dispositivos médicos (secção 3.1), definir os requisitos (secção 3.2) e a arquitetura global (secção 3.3) da solução a implementar.

3.1 Definição do problema

As mudanças demográficas que assistimos no presente obrigam a melhorias de conforto e segurança em diversos produtos largamente comercializados no passado.

O processo de desenvolvimento de um dispositivo médico integra diversas fases que permitem a definição e a avaliação de sequências de movimentos executados pelos diferentes atuadores que integram o protótipo. O processo atual de interação entre a equipa de *R&D* e os diferentes atuadores lineares que compõem o protótipo recorre a um comando básico desenvolvido por uma entidade externa. O comando de controlo é desenvolvido para cada protótipo em desenvolvimento e permite estender/retrair/parar cada um dos atuadores de forma independente e executar sequências previamente definidas. Desta forma, o processo atual é limitativo e desvantajoso, pois a adaptação do comando de protótipo para protótipo não é fiável e a definição de uma nova sequência ou a redefinição de sequências existentes implica um novo fluxo de informação entre a entidade externa e a equipa de *R&D* resultando num compasso de espera que contribui negativamente para a dinâmica do processo global do desenvolvimento de um novo dispositivo médico.

Dado o exposto, é possível identificar algumas necessidades com que a equipa de *R&D* se depara durante o desenvolvimento de dispositivo médicos. A necessidade de uma interação com os atuadores lineares mais intuitiva, o desenvolvimento de um sistema de controlo modular capaz se adaptar aos diferentes atuadores, a dificuldade na definição ou redefinição de sequências e uma plataforma capaz definir testes de repetibilidade são exemplos práticos destas necessidades.

3.2 Definição dos requisitos do sistema

De forma a tornar o processo de definição e avaliação das sequências de movimentos durante o desenvolvimento de dispositivos médicos mais dinâmico e simplificado pretende-se a construção de um sistema que permita à equipa de *R&D* posicionar cada atuador com maior fiabilidade e segurança.

Assim, o sistema desenvolvido deve cumprir os objetivos e requisitos listados em seguida:

1. A plataforma com *interface* gráfica deve poder integrar até ao máximo de cinco atuadores lineares;
2. O sistema deve ser modular permitindo a integração de atuadores com diferentes caraterísticas;
3. O sistema deve permitir o acionamento individual de cada atuador linear do dispositivo médico;
4. O sistema deve permitir o acionamento síncrono de dois atuadores lineares que integram o dispositivo médico em desenvolvimento;
5. O sistema deve permitir definir o deslocamento do atuador em cada movimento;
6. O sistema deve permitir um mecanismo de *reset* do posicionamento do atuador;
7. O sistema deve ser dotado de um mecanismo de *soft start and stop*;
8. O sistema deve permitir a configuração do mecanismo de *soft start and stop*;
9. O sistema deve permitir definir a velocidade de operação de cada atuador;
10. O sistema deve permitir a definição de sequência de movimentos resultantes do acionamento individual ou síncrono dos atuadores que compõem o protótipo em desenvolvimento;
11. O sistema deve permitir a definição da quantidade de repetições de uma sequência de movimentos;
12. O sistema deve permitir o acionamento de cada atuador a diferentes cargas;
13. O sistema deve ser capaz de executar uma calibração inicial relativa a cada atuador;
14. O sistema deve monitorizar o estado de cada atuador;
15. O sistema deve monitorizar a corrente consumida por cada atuador durante cada movimento executado;
16. O sistema deve possuir um mecanismo de *backup* com informações relativas ao estado de cada atuador do dispositivo médico.
17. O sistema deve conseguir armazenar uma sequência de movimentos para uma reutilização futura.

3.3 Arquitetura global da solução

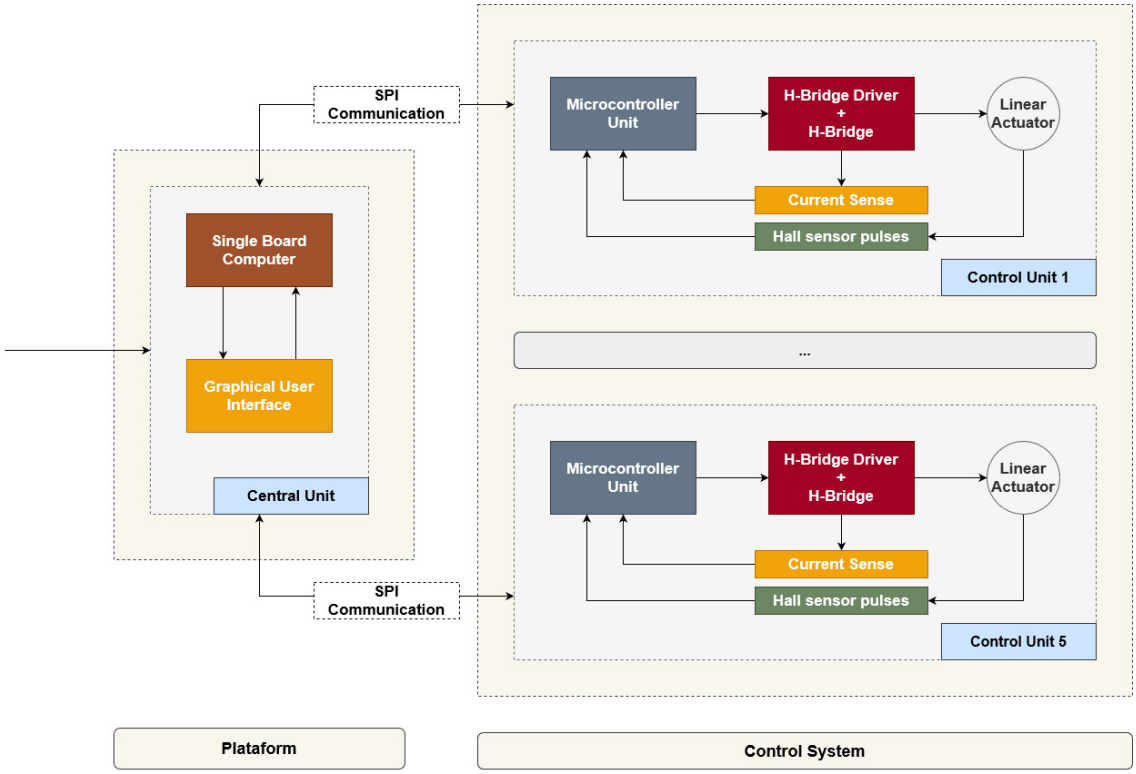


Figura 3.1: Arquitetura global do sistema proposto

De forma a cumprir os principais objetivos desta dissertação foi desenhada uma solução baseada em diferentes blocos com funções e características distintas. Na arquitetura global proposta, figura 3.1, podemos distinguir dois blocos principais: o sistema de controlo, constituído por cinco unidades de controlo individuais, que permitem acionar os vários atuadores lineares que constituem o dispositivo médico em desenvolvimento e a unidade central que engloba a plataforma com interface gráfica funcionando como uma ponte entre a equipa de *R&D* e os sistemas de controlo individuais permitindo o acionamento dos atuadores através de comandos básicos e a monitorização do estado de cada atuador. O fluxo de informação entre os dois blocos principais é feito com base no protocolo de comunicação *SPI* seguindo uma arquitetura *master-slave* em que a unidade de central adopta o comportamento do *master* e coordena todo o fluxo de informação e cada unidade de controlo assume o papel de *slave*. Nos capítulos 5 e 7 irá ter lugar uma análise mais detalhada de cada um dos blocos que compõem a arquitetura.

Capítulo 4

Atuador Linear Elétrico

O atuador linear elétrico permite transformar o movimento rotacional proveniente de um motor elétrico num movimento mecânico de puxar ou empurrar. Em dispositivos médicos são amplamente utilizados devido ao seu movimento simples e seguro, à sua fácil instalação e utilização e, também, por permitirem um controlo preciso e suave do movimento que executam. Este tipo de atuadores são normalmente integrados na indústria médica e geriátrica em diversos produtos, como por exemplo, cadeiras elevatórias, camas articuladas e elevadores transferência. Estes produtos permitem facilitar os diversos movimentos de levantar, inclinar e puxar do quotidiano do paciente e das equipas cuidadoras oferecendo, assim, aos pacientes melhor conforto e às suas equipas melhores condições.[47]

O presente capítulo tem como principais objetivos apresentar o princípio de funcionamento de um atuador linear (secção 4.1 e secção 4.2), analisar o princípio de controlo (secção 4.3) e apresentar uma análise dos modelos de atuadores lineares com que a equipa de *R&D* mais interage: TA23[15] e TA31[14] do fabricante TiMOTION e os modelos LA27[16][17] e LA40[18] do fabricante LINAK, secção 4.4.

4.1 Princípio de funcionamento

De forma generalizada, um atuador linear é um mecanismo responsável por originar um movimento mecânico linear capaz de mover um objeto utilizando uma fonte de energia externa. Existem no mercado diversos tipos de atuadores lineares que se distinguem principalmente quanto à fonte de energia externa: atuadores pneumáticos, atuadores hidráulicos e atuadores elétricos são exemplos. Em dispositivos médicos os atuadores elétricos são os mais comuns graças à fácil utilização e programação, ao *design* compacto e à precisão de controlo nos momentos de aceleração e desaceleração. [47]

Num atuador linear elétrico podemos distinguir dois sistemas principais, o sistema elétrico responsável por transformar energia elétrica em energia mecânica através de um motor elétrico e um sistema mecânico capaz transformar um movimento rotacional num movimento linear através de dois principais componentes: uma caixa de engrenagens e um parafuso sem fim. A caixa de

velocidades é responsável por alterar a relação entre a velocidade de rotação do eixo do motor e o parafuso, fazendo girar o parafuso de forma a acionar uma porca que permite a extensão ou a retração do atuador.

4.2 Componentes principais

O design de um atuador linear elétrico pode variar de fabricante para fabricante e de modelo para modelo. É um mercado vasto existindo diferentes tipologias com desempenhos e aplicações distintas. Nesta secção são analisados os componentes/mecanismos comuns à grande maioria dos modelos e que têm uma participação ativa no funcionamento do atuador.

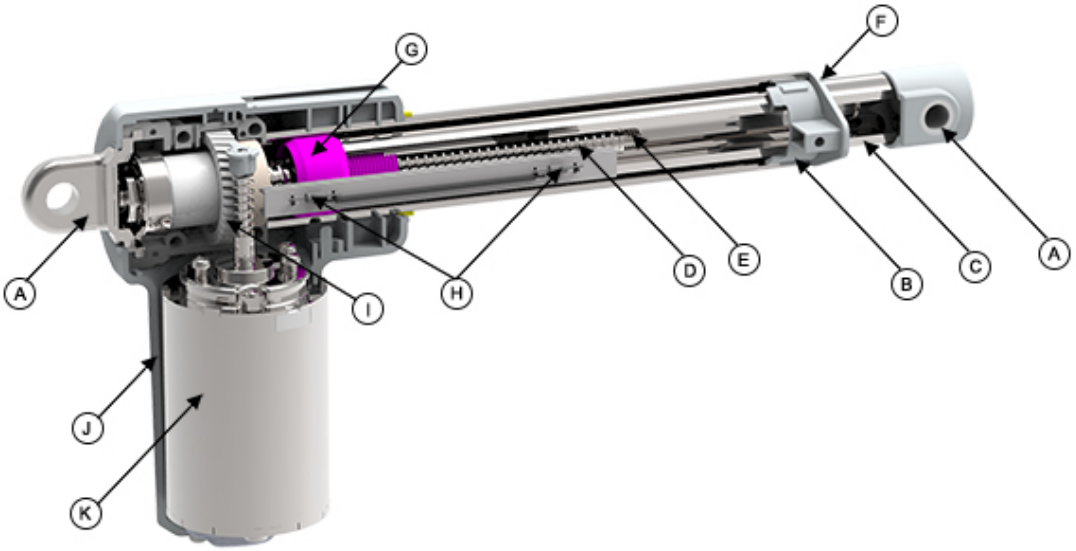


Figura 4.1: Estrutura típica de um actuador linear elétrico[8]

D - Parafuso sem fim;

G - Porca de acionamento;

H - Sensores de fim de curso;

I - Caixa de velocidades;

K - Motor elétrico

4.2.1 Motor elétrico

O motor elétrico integrado num atuador linear tem como principal objetivo a conversão de energia elétrica fornecida num movimento mecânico. Os modelos de atuadores lineares em análise integram um motor CC de ímanes permanentes com escovas que se encontram estudado na secção 2.1.3.

De forma a auxiliar o controlo de posição de cada atuador cada motor está equipado com dois sensores digitais de efeito *Hall*. O sinal de saída de cada sensor é ativo quando, na redondeza do sensor, a densidade do campo magnético excede um certo limite máximo e desativo quando a densidade do campo atinge o limite mínimo. Os sensores de efeito *Hall* são bastantes utilizados nestas aplicações devido à sua boa relação custo/eficiência e por ser considerados sensores robustos com uma esperança média de operação longa.[8][48]

4.2.2 Sensores de fim de curso

Os sensores de fim de curso têm um função de proteção uma vez que impedem que o atuador estenda ou retraia mais do que o seu *design* permita. Os dois sensores são responsáveis por detetar as duas posições extremas: quando o atuador se encontra totalmente estendido e quando o atuador se encontra totalmente retraído. Ambos são colocados em série com a alimentação interrompendo-a quando um das posições é atingida e impedindo, assim, que atuador continue o movimento.[8]

4.2.3 Caixa de velocidades

A caixa de velocidades é um mecanismo mecânico constituído por um conjunto de engrenagens que permite alterar a relação de velocidade entre a velocidade do mecanismo de acionamento e a velocidade do mecanismo acionado. A relação entre as duas velocidades é conhecida como relação de transmissão. Este mecanismo mecânico introduz, também, uma relação entre o binário dos dois mecanismos com um valor inverso à relação de transmissão. A dualidade entre as duas relações é justificada pela manutenção praticamente total da potência ao longo do mecanismo.

Neste contexto de aplicação, o motor elétrico é considerado o mecanismo de acionamento e o parafuso sem fim o mecanismo acionado. A utilização de uma caixa neste contexto é vantajosa na medida em que permite que motores com densidades de binário mais baixos que operem em altas rotações possam ser reduzidos e, assim, consigam acionar cargas mais pesadas do que seriam capazes de acionar diretamente.[49]

As engrenagens que constituem a caixa são normalmente fabricadas em plástico ou aço. As fabricadas em plástico apresentam maior sensibilidade em operações em altas temperaturas e menores níveis de eficiência quando comparadas com as fabricadas em aço.[48]

4.2.4 Parafuso sem fim

O parafuso sem fim é um mecanismo que permite converter o movimento rotativo proveniente do eixo do motor num movimento linear que permite a extensão e a retração do atuador linear. O parafuso gira, acionado pela caixa de velocidades, de forma a estender ou retrair a porca criando um movimento linear.

O *design* do parafuso influencia a rapidez com que atuador se estende ou retrai bem como a quantidade de carga que consegue movimentar. Existem duais principais especificações que permitem avaliar a *performance* do parafuso: são conhecidas como o valor de passo e o valor de avanço. O passo representa a distância entre duas ranhuras consecutivas do parafuso, ao invés

que o avanço representa a distância linear que o parafuso percorre por revolução. Valores de passo mais altos implicam níveis de eficiência mais alta devido ao menor atrito entre o parafuso e a porca, contudo a capacidade de manter uma carga estável numa determinada posição diminui.[49][48]

4.3 Princípio de controlo

O controlo de um atuador linear elétrico é muito semelhante ao controlo do motor elétrico que este integra, uma vez que é o motor elétrico a entidade capaz de gerar o movimento primário que permite a extensão/retração do atuador.

Os modelos analisados integram um motor CC de ímanes permanentes com escovas e, como visto na secção 2.1.3, estes motores caracterizam-se por duas principais relações de proporcionalidade: uma das relações é verificada entre a corrente consumida e o binário gerado e a outra relação entre a velocidade de rotação e a tensão de alimentação.

A implementação do mecanismo *soft start and stop* e, por consequência, do controlo de velocidade do atuador, baseia-se na dualidade existente entre a velocidade de rotação do rotor e a tensão de alimentação. Utilizando a técnica digital de modulação *PWM* é possível acionar o atuador com uma tensão média de alimentação proporcional ao *duty-cycle* que caracteriza a onda *PWM*. Se definirmos velocidade linear nominal como a velocidade linear típica do atuador a mover uma dada carga quando alimentando à tensão nominal, a modulação *PWM* permite-nos mover essa mesma carga com um velocidade proporcional à velocidade linear nominal numa razão representada pelo *duty-cycle* da onda *PWM*.

Por outro lado, em cada motor estão integrados dois sensores de efeito *Hall* com o objetivo de auxiliar o controlo de posição. Este controlo é baseado numa relação calculada *a priori* entre o comprimento do deslocamento linear máximo do atuador e o número total de rotações do rotor durante um movimento de extensão/retração máxima. De forma simplificada, esta relação representa o número de rotações do rotor por deslocamento unitário (1 *mm*) do atuador linear. Assim, uma contagem global dos pulsos digitais provenientes dos sensores permite calcular o número de rotações e, por consequência, o deslocamento total efetuado pelo atuador durante cada movimento. A utilização de dois sensores é vantajosa, na medida em que é possível a identificação da direção do deslocamento (extensão ou retração) pelo desfasamento dos pulsos provenientes dos dois sensores.

4.4 Curvas de caraterização

De forma a avaliar a *performance* de cada atuador em diferentes situações de carga serão analisadas duas curvas que servem para analisar o desempenho em duas variáveis: a velocidade linear de extensão/retração do atuador e a corrente consumida pelo motor.

Se fizermos uma análise da figura 4.2 facilmente entendemos que podemos individualizar subsistemas com características e especificações bem definidos que cooperam no sentido permitir

mover uma carga a uma dada velocidade. Cada subsistema influencia o desempenho e a eficiência global do sistema.

Cada modelo de atuador linear analisado possui variantes no seu *design* que permitem alcançar desempenhos distintos. As variantes podem estar relacionadas com o comprimento máximo de extensão do atuador, com o valor de passo e avanço do parafuso sem fim e com a relação de transmissão da caixa de velocidades.

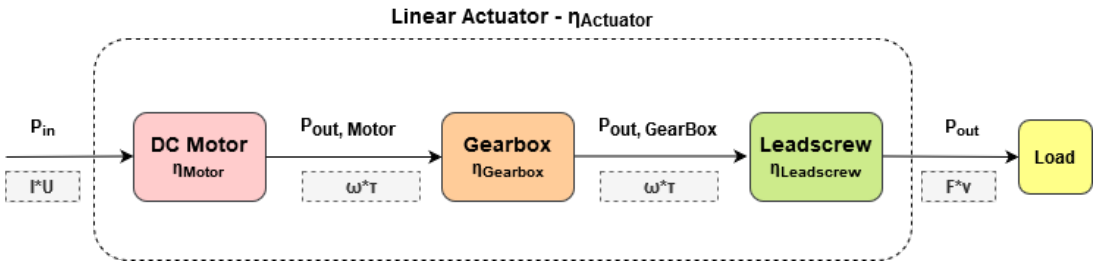


Figura 4.2: Individualização de sub-sistemas que compõem um atuador linear elétrico

As curvas apresentadas seguidamente resultam de análises feitas pelo fabricante com o atuador linear a operar à tensão nominal (24V) e a acionar cagas sucessivamente mais elevadas. As informações constam quer em tabelas, quer em gráficos nas *datasheets* dos modelos em estudo e foram compactadas em apenas dois gráficos por modelo de forma a facilitar a perceção das relações existente entre a corrente consumida pelo motor e a velocidade linear com o valor de carga acionada.

Da análise destas curvas é possível concluir que existe uma relação direta entre a corrente do motor e a carga acionada, bem como uma relação inversa entre a velocidade linear e a carga. Ou seja, o acionamento de cargas sucessivamente mais pesadas, em condições semelhantes, resulta em valores de correntes consumidas pelo motor sucessivamente superiores e a valores de velocidades de extensão/retração sucessivamente menores.

4.4.1 Modelo TA31

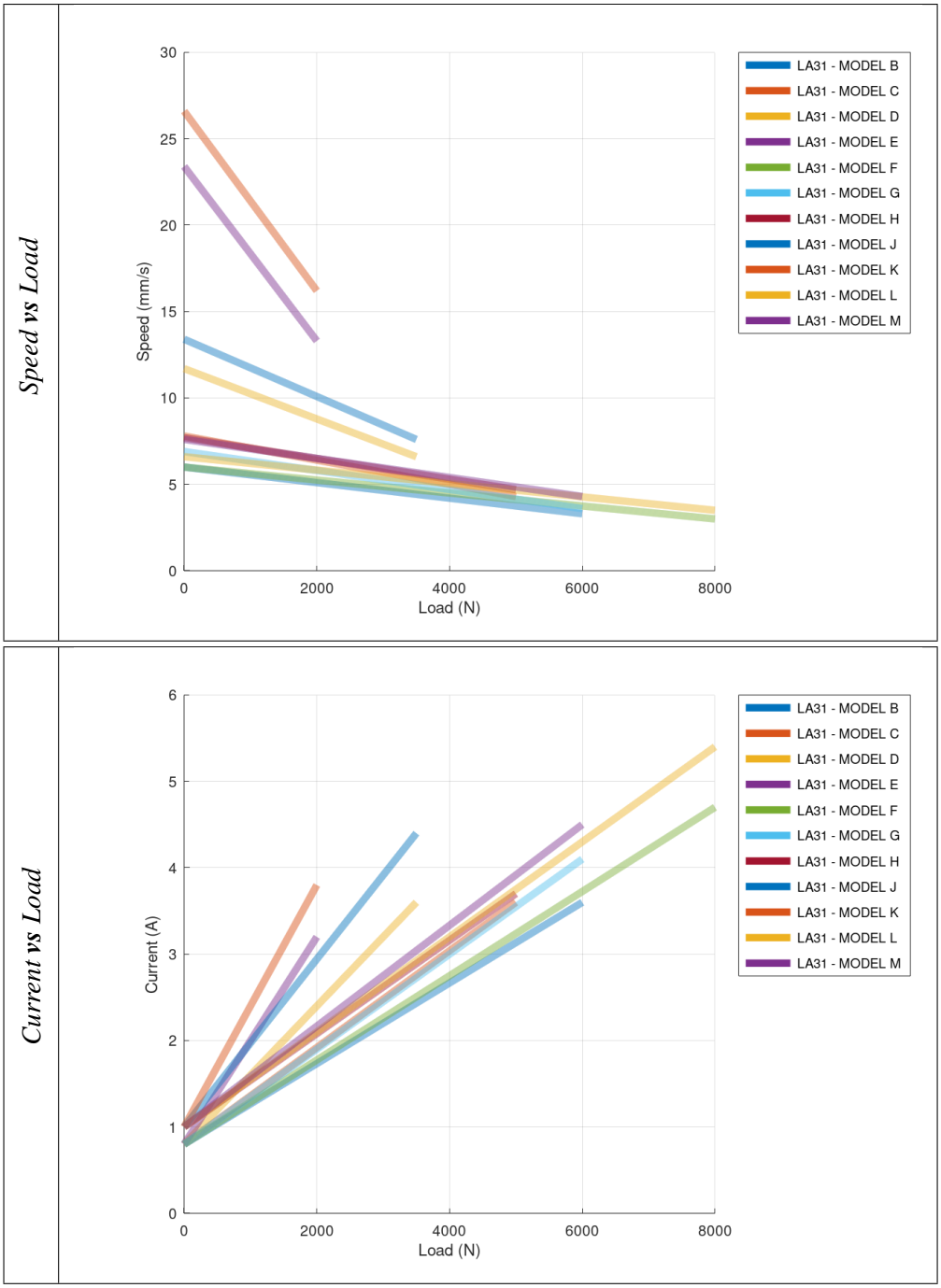


Tabela 4.1: Curvas de caracterização - Modelo TA31[14]

4.4.2 Modelo TA23

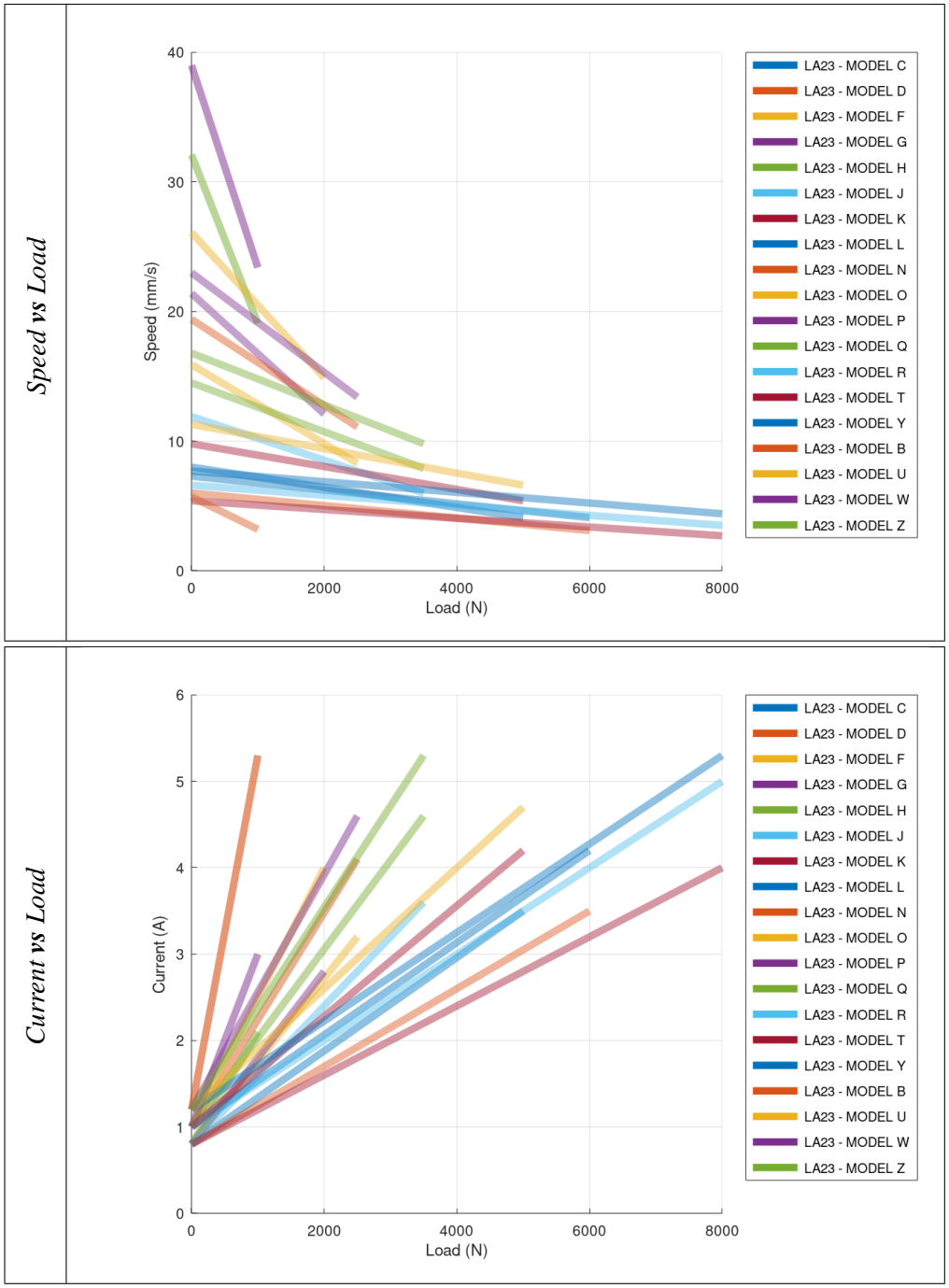


Tabela 4.2: Curvas de caracterização - Modelo TA23[15]

4.4.3 Modelo LA27

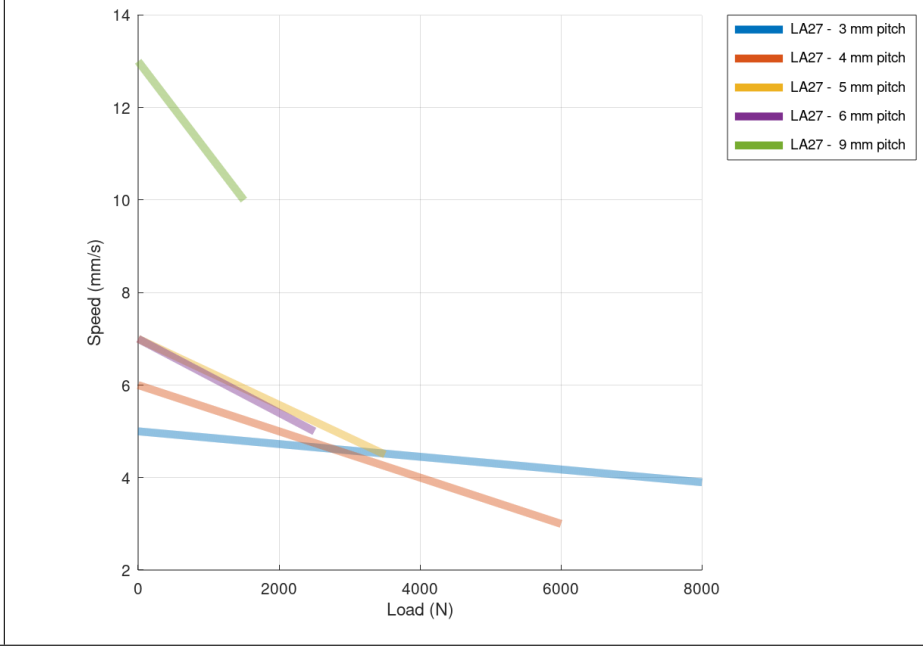
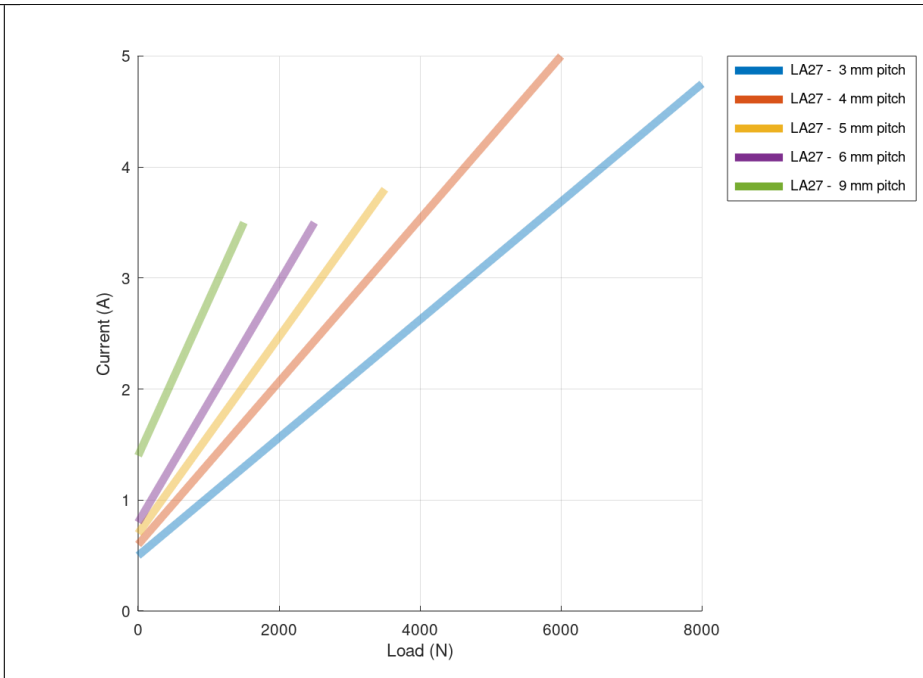
Speed **vs** *Load*	
Current **vs** *Load*	

Tabela 4.3: Curvas de caracterização - Modelo LA27[16][17]

4.4.4 Modelo LA40

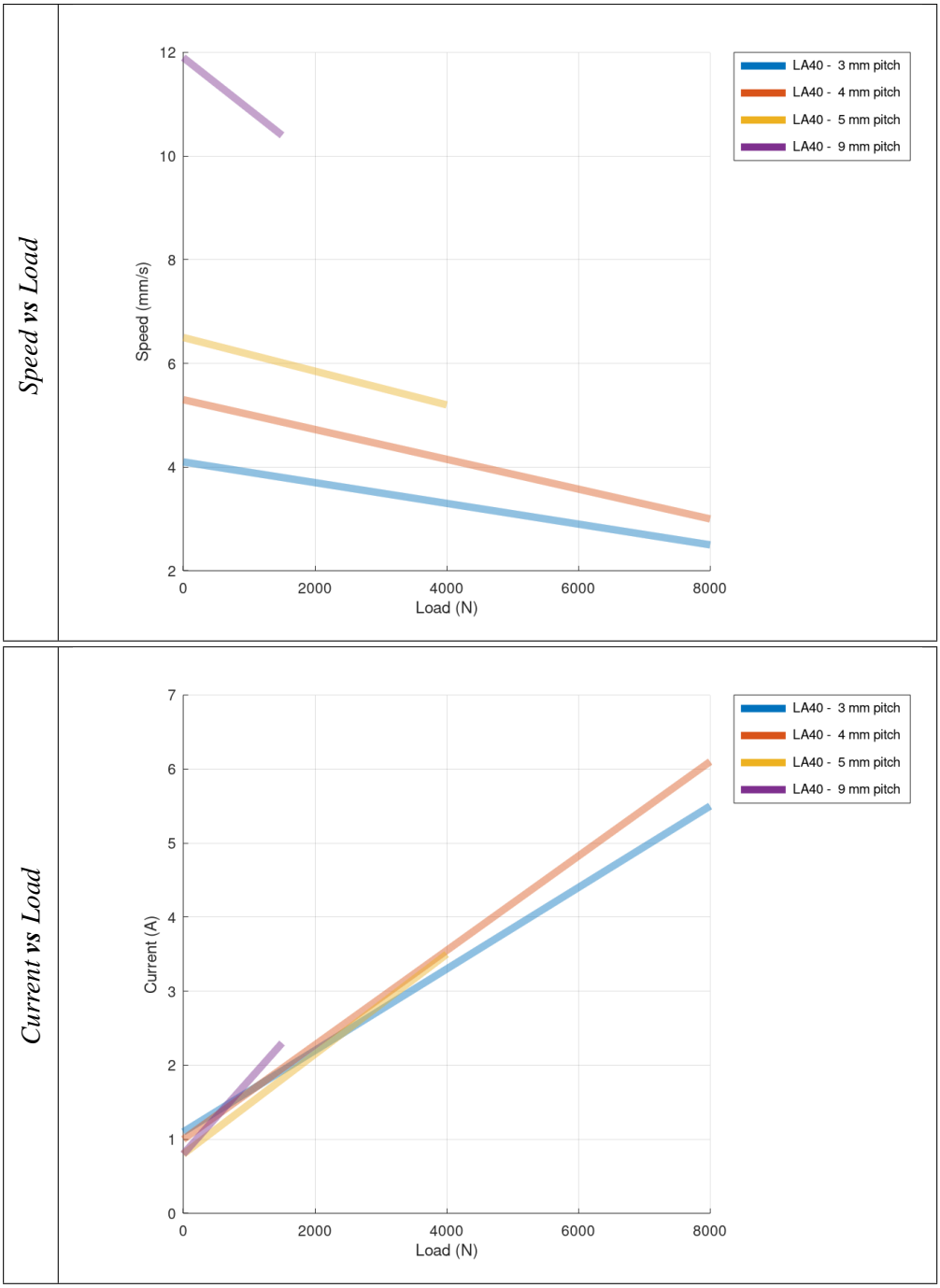


Tabela 4.4: Curvas de caracterização - Modelo LA40[18]

4.4.5 Síntese dos atuadores estudados

Modelo	***Duty-cycle***	**Tensão nominal**	**Máxima carga**	**Máximo comprimento**	**Corrente máxima**		**Velocidade mínima**	
					Máxima carga	**Sem carga**	**Máxima carga**	**Sem carga**
TA31	10%	24V	8 000N	450mm	5.4A	1.0A	3.3mm/s	6.0mm/s
TA23	10%	24V	10 000N	1 000mm	5.3A	1.2A	2.7mm/s	5.4mm/s
LA27	10%	24V	6 000N	405mm	5.0A	1.4A	5.0mm/s	3.0mm/s
LA40	10%	24V	8 000N	405mm	6.1A	1.1A	4.1mm/s	2.5mm/s

Tabela 4.5: Resumo das características dos quatro modelos em estudo[14][15][16][18]

Capítulo 5

Unidade de Controlo

A unidade de controlo é o subsistema implementado com o objetivo de controlar a posição e a velocidade de atuação de cada atuador linear. Este sub-sistema caracteriza-se pela sua função modular que através de mecanismos de configuração inicial consegue adaptar as estratégias de controlo aos modelos de atuadores lineares estudados na secção 4.4. Esta unidade é, portanto, parametrizada com as características individuais de cada atuador, como a corrente máxima consumida e o comprimento da extensão máxima do atuador. A unidade de controlo permite executar as diversas sequências de movimentos definidas na unidade central, bem como monitorizar o estado do atuador (posição, velocidade e corrente). Além da execução das diferentes sequências e dos mecanismos de configuração inicial, a unidade de controlo permite posicionar cada atuador numa posição de *reset* através de um mecanismo seguro em situações de recuperação de falhas, como a interrupção do fornecimento de energia ou a interrupção inesperada da comunicação com a unidade central, por exemplo.

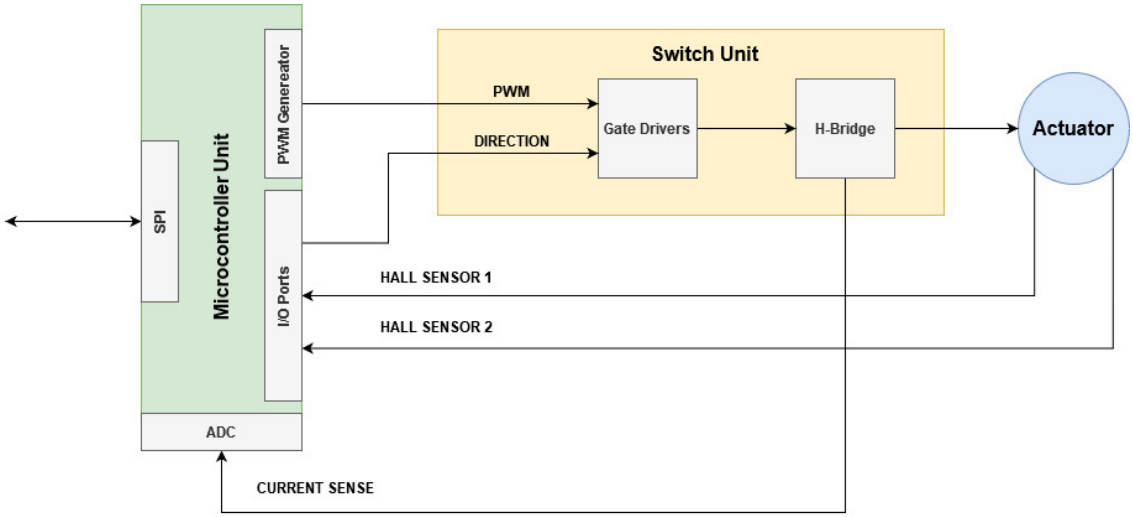


Figura 5.1: Unidade de Controlo - Arquitetura

Na figura 5.1 podem ser avaliados os principais elementos que põem a unidade de controlo, bem como os principais fluxos de informação entre eles. A unidade de comutação tem como

principal objetivo a inversão do sentido do movimento do atuador pela inversão da polaridade da alimentação do atuador. O microcontrolador, por sua vez, é a unidade de processamento responsável pela implementação das estratégias de controlo e pela mediação da comunicação com a unidade central, recebendo comandos que resultam na execução de num movimento e enviando comandos com atualizações do estado atual do atuador.

As secções seguintes estão dedicados a diferentes aspetos, quer estruturais, quer funcionais, da operação de toda a unidade de controlo. Na secção 5.1 é discutido a escolha e o desempenho de cada componente que integra a arquitetura desta unidade; na secção 5.2 está exposto o funcionamento da configuração inicial desta unidade; na secção 5.3 é apresentada a estratégia que permite controlar a posição de cada atuador; na secção 5.4 é avaliada a estratégia que permite implementar o mecanismo de *soft start and stop* e, por fim, na secção 5.5 são apresentados os diferentes algoritmos que permitem implementar os vários modos de operação desta unidade.

A tabela 5.1 permite reunir e definir alguns conceitos e parâmetros utilizados na apresentação e implementação da unidade de controlo.

Símbolo	**Definição**
stroke lenght	Comprimento do curso máximo do atuador.
refracted position	O atuador encontra-se totalmente retraído.
extended position	O atuador encontra-se totalmente estendido.
reset position	Posição do atuador depois da execução do mecanismo de *reset*, o atuador retrai totalmente.
absolute position	Corresponde ao comprimento da extensão atual do atuador.
initial position	Posição do atuador quando inicia um movimento.
final position	Posição do atuador quando termina um movimento.
shift	Valor absoluto da diferença entre a posição final e inicial do atuador durante um movimento.
pulse rate	Número de rotações do rotor por cada milímetro de deslocamento do atuador.
current sense	Valor proporcional à corrente consumida pelo motor.
nominal voltage	Tensão nominal do atuador
nominal speed	Velocidade linear esperado do atuador quando alimentado à tensão nominal
operation speed	Velocidade linear de operação do atuador
speed ratio	Relação entre a a velocidade de operação e velocidade nominal
accelaration time	Duração do mecanismo *soft start and stop*

Tabela 5.1: Conceitos básicos - unidade de controlo

5.1 Estrutura e componentes

O mercado atual de componentes eletrónicos, cada vez mais vasto e diversificado, permite-nos implementar a unidade de controlo com recursos a diferentes integrados que desempenham funções e tipologias distintas. De uma avaliação do mercado resultou uma seleção de componentes dispostos em três soluções (figura 5.2) que permitem a implementação global da arquitetura desta

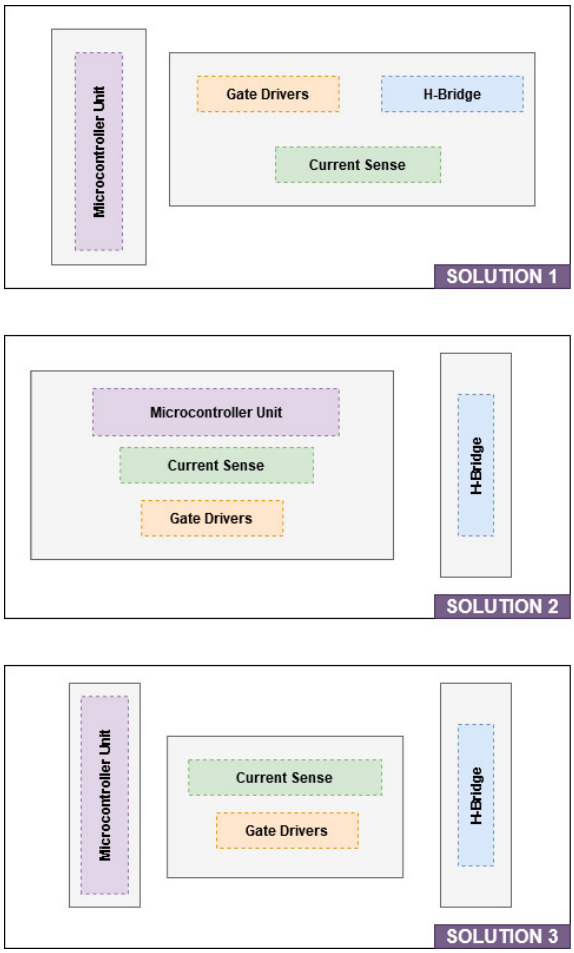


Figura 5.2: Soluções existentes no mercado

unidade. Na solução 1, o design da unidade de controlo seria constituído por dois componentes: um microcontrolador independente e um integrado que inclui, simultaneamente, os transístores dispostos numa tipologia *H-Bridge* e os *drivers* que permitem a comutação de cada transístor. Esta primeira solução apresenta-se vantajosa na medida em que todo a unidade de comutação é compactada num único componente permitindo economizar espaço e alcançar uma melhor relação custo/funcionalidades. A solução 2 apresentada é constituída também por dois elementos, um integrado constituído pelo microcontrolador e pelos *drivers* que permitem comutar os transístores externos sendo. O seu uso é privilegiado em aplicações requerem maior potência. Por fim, a solução mais modular das três, é constituída pelos três elementos individualizados: o microcontrolador, os *drivers* e os transístores. Esta solução embora seja a menos económica e eficiente das apresentadas apresenta como principal vantagem, a fácil adaptação futura da unidade a níveis de potência diferentes. No anexo A está presente o estudo de mercado das três soluções apresentadas.

A implementação da unidade de controlo baseou-se na arquitetura da solução 1 de forma a facilitar o desenvolvimento e o *design* de toda a unidade de comutação permitindo aumentar o número de funcionalidades (proteções e diagnósticos de operação ou falhas) oferecidas por esta unidade. Embora, a flexibilidade de adaptação fique mais comprometida com este solução, salvaguarda-se

uma adaptação futura de todo o sistema desenvolvido a outros atuadores com características elétricas diferentes pela substituição de toda a unidade de comutação mas mantendo-se toda a lógica que o microcontrolador executa.

5.1.1 Unidade de comutação - *H-Bridge* e *Gate Driver*

A tipologia *H-Bridge* foi escolhida uma vez que se pretende um controlo bidirecional do movimento e que, como visto na secção 2.1.3, a inversão do movimento é conseguida pela inversão do sentido de corrente, sendo cada meia ponte responsável por comutar um ponto de alimentação do motor.

Dado o estudo do mercado feito no anexo A e as características dos atuadores apresentadas na tabela 4.5 optou-se pelo circuito integrado *VNH5019A-E* do fabricante *STMicroeletronics*. O estudo mais detalhado do desempenho deste integrado é apresentado seguidamente e pretende avaliar diferentes parâmetros elétricos e térmicos.

O *IC* escolhido foi desenhado pela *STMicroeletronics* com o objetivo acionar motores de corrente contínua em diversas aplicações de automação. Este integrado além da sua função principal possui diversas funções de proteção que interrompem o funcionamento normal do integrado com o objetivo de preservar a integridade do circuito. Além das funções de proteção, o *IC* possui mecanismos que permitem diagnosticar diferentes situações de operação. As principais funções e mecanismos estão descritas na tabela 5.2.

Função	Descrição
Undervoltage and overvoltage shutdown	Desliga a operação de todo o integrado quando a tensão de alimentação está fora da gama de operação normal.
Overtemperature protection	Desliga a operação de todo o integrado numa situação de sobreaquecimento do integrado.
Low-side overload detector	Permite detetar quando a corrente nos transístores inferiores ultrapassa um certo valor, acionando os mecanismos de proteção contra sobrecorrente e sobreaquecimento
Reverse battery protection	Desliga a operação do integrado numa situação de inversão de polaridade da tensão de alimentação
Analogue current sense	Fornece um corrente proporcional à corrente do motor de acordo com uma razão conhecida

Tabela 5.2: Funcões básicas desempenhadas pelo *IC VNH5019A-E*[11]

O dimensionamento dos componentes que compõem a unidade de comutação baseou-se na avaliação do comportamento elétrico de cada atuador linear na situação de maior esforço. A situação de maior esforço coincide com o acionamento contínuo à carga máxima durante um período temporal máximo que respeite o *duty-cyle* do atuador. Desta forma, da análise da tabela 4.5 concluímos que o pior caso acontece no controlo do atuador linear *LA40* durante o acionamento de uma carga de 8000N à tensão nominal (24V) ao longo de 120 segundos, ou seja, consumindo continuamente 6.1A. Outro parâmetro a considerar para esta avaliação é a frequência da modulação *PWM* escolhida, uma vez que esta técnica influencia a potência dissipada devido às comutações dos transístores. Como se verá na secção 5.4 a frequência poderá ser redefinida sempre que necessário dentro de uma certa gama de valores que está dependente do nível de potência do atuador.

Desta forma, a avaliação do desempenho terá em conta uma gama de valores compreendidos entre os 4kHz e os 20kHz de forma a tirar o maior partido das características do integrado.

A dissipação de energia no integrado é a principal fonte de calor. Esta dissipação deve-se principalmente às perdas geradas pela comutação dos transístores que compõem o integrado. De forma a evitar a degradação do circuito foi necessário avaliar a quantidade de energia dissipada e a necessidade de incluir um dissipador de calor. O dispositivo *VNH5019A-E* desenhado sob o *package* $MultiPowerSO-30^{TM}$ e incorpora três *chips* diferentes com um mecanismo de dissipação de calor através das pistas de cobre do circuito impresso (anexo B). Assim, o desempenho deste mecanismo de dissipação de calor pode ser melhorado pelo aumento da área de cobre de algumas pistas. De forma a avaliar a potência dissipada por cada *chip* é necessário conhecer os elementos que cada um integra e o seu funcionamento: o *chip 1* é composto pelos dois lados *high* da tipologia *H-Bridge*, enquanto que o *chip 2* e o *chip 3* são compostos por cada um dos lados *low* da tipologia *H-Bridge*. Quando o controlo do estado dos transístores é feito com recurso à técnica *PWM* apenas os transístores do lado *low* comutam contribuindo, assim, para diminuir a potência total dissipada pois elimina-se as perdas por comutação no lado *high* da tipologia *H-Bridge*. Podemos destacar dois principais tipo de perdas durante o funcionamento do transístor, as perdas por condução e as perdas por comutação. O membro $P_{conduction}$ na equação 5.1 representa a energia dissipada nas resistências $R_{ds(On)}$ durante a condução de cada transístor, enquanto que o membro $P_{switching}$ da equação 5.2 representa as perdas por comutação de cada transístor.

$$P_{conduction} = R_{On} I_{Out}^2 \tag{5.1}$$

$$P_{switching} = P_{fall} + P_{rise} \tag{5.2}$$

$$P_{fall} = \frac{1}{2} V_{cc} I_{Out} f_{Pwm} t_{Fall} \tag{5.3}$$

$$P_{rise} = \frac{1}{2} V_{cc} I_{Out} f_{Pwm} t_{Rise} \tag{5.4}$$

$$\begin{aligned} P_{chip1} &= 2P_{conduction(HS)} \\ P_{chip1} &= 2R_{On(HS)} I_{Out}^2 \end{aligned} \tag{5.5}$$

$$\begin{aligned} P_{chip2} &= P_{conduction(LS)} + P_{switching(LS)} \\ P_{chip2} &= R_{On(LS)} I_{Out}^2 + \frac{1}{2} V_{cc} I_{Out} f_{Pwm} (t_{Rise} + t_{Fall}) \end{aligned} \tag{5.6}$$

$$\begin{aligned} P_{chip3} &= P_{conduction(LS)} + P_{switching(LS)} \\ P_{chip3} &= R_{On(LS)} I_{Out}^2 + \frac{1}{2} V_{cc} I_{Out} f_{Pwm} (t_{Rise} + t_{Fall}) \end{aligned} \tag{5.7}$$

O desempenho térmico foi avaliado utilizando as informações provenientes da *datasheet* e a análise detalha está presente no anexo B. A tabela 5.3 resume as principais conclusões retiradas da avaliação térmica realizada permitindo estabelecer a frequência máxima para a modulação *PWM* nas diferentes situações e condições de operação.

Área (cm^2)	**Corrente@Frequência**
Footprint	4A@8kHz / 5A@4kHz / 6A / 7A
4	4A@12kHz / 5A@8kHz / 6A@4kHz / 7A
8	4A@16kHz / 5A@8kHz / 6A@8kHz / 7A@4kHz
16	4A@16kHz / 5A@12kHz / 6A@8kHz / 7A@4kHz

Tabela 5.3: Dinâmica térmica - Conclusões

5.1.2 *Microcontroller unit*

O microcontrolador é o centro de processamento da unidade de controlo e deve ser dotado de algumas caraterísticas e funções que permitem a implementação de algumas funcionalidades. Assim, na tabela abaixo, tabela 5.4, estão reunidas os principais requisitos que o microcontrolador deve obedecer para pode ser integrado no sistema.

Comunicação SPI	
Tipologia de rede	*Full-Duplex*, Comunicação biderecional
Arquitetura	*Master-slave*
Data bits	8
Frequência do relógio máxima	125 MHz
Frequência do relógio mínima	7629 Hz
Ordenação	Bit mais significativo primeiro
Polaridade	Estado *idle* do relógio representado pelo estado lógico "0"
Fase	Captura de dados no flanco descendente
Canais *ADC*	
Tensão de referência	5V
Mínimo intervalo entre leituras	$70\mu s$
PWM	
Frequência mínima	4kHz
Frequência máxima	20kHz

Tabela 5.4: Requisitos para a seleção do microcontrolador

A implementação deste sistema baseou-se no microcontrolador *ATMEGA328-P*, utilizando a placa de desenvolvimento *Arduino UNO R3*. As principais caraterísticas e configurações estão listadas abaixo.

- **Frequência *CPU***: 16MHz
- **Memória *flash***: 32KB
- **Comunicação *SPI***: Permite frequências de operação até 4MHz no modo *slave*. Configuração no modo *slave* com todas as caratrísticas apresentadas na tabela 5.4.

- **Canais *ADC***: Permite uma taxa de amostragem máxima de 15kSPS, contudo foi configurado apenas com taxa de amostragem aproximadamente de 8,6kSPS, permitindo assim leituras a cada 116μs.

- ***PWM***: Configuração do *timer1* de 16 *bits* para gerar ondas *PWM* com frequências entre 4kHz e 20kHz e *duty-cycles* variáveis.

- **Interrupções externas**: As interrupções externas foram configuradas para ser desencadeadas durante um flanco descendente.

5.2 Parametrização e configuração inicial

A configuração inicial da unidade de controlo permite que as estratégias de controlo sejam executadas com o mesmo desempenho independente do atuador em operação. Este mecanismo deve ser executado sempre que um novo modelo de atuador linear seja incluído no protótipo desenvolvimento. É responsável por calcular a relação *pulse rate* do novo atuador de forma a adaptar as estratégias de controlo de posição e adaptar o controlo de velocidade ao nível de potência do atuador pela definição do limite máximo da frequência da modulação *PWM*.

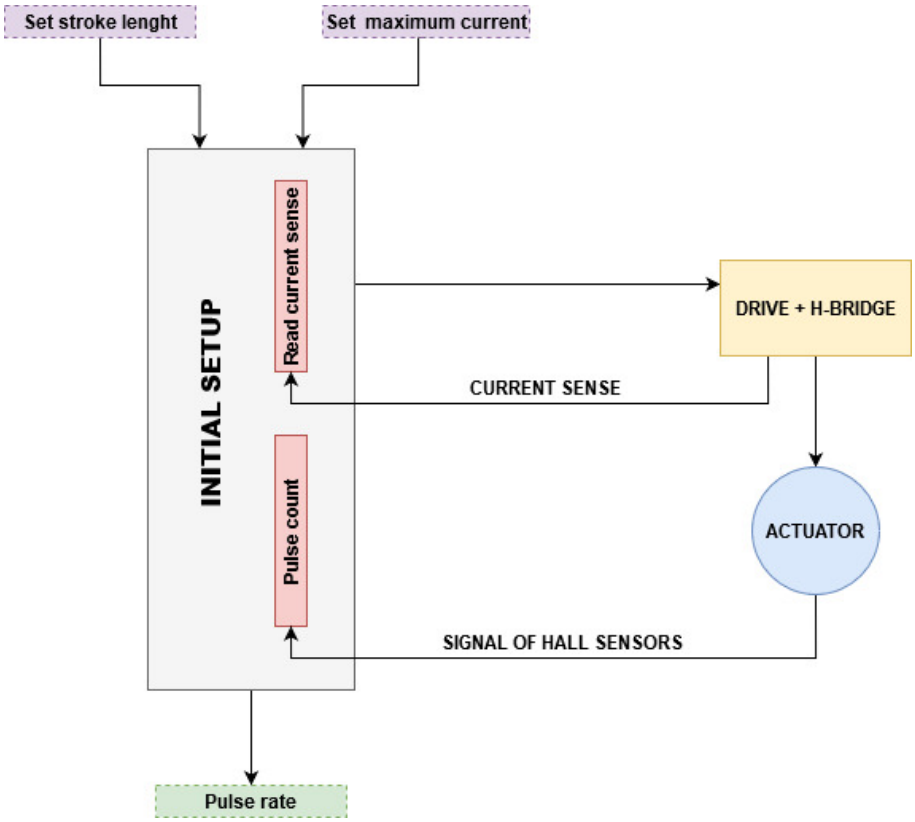


Figura 5.3: Configuração inicial

O cálculo da relação *pulse rate* precisa de dois parâmetros de entrada: o comprimento da extensão máxima do atuador (*stroke lenght*) e a corrente máxima consumida pelo atuador. A corrente máxima permite ajustar a interpretação da leitura do mecanismo *current sense* aos níveis de potência do novo atuador e definir a frequência máxima da onda *PWM* gerada pelo microcontrolador. Por outro lado, o comprimento está diretamente relacionada com a relação *pulse rate* que se pretende calcular (tabela 5.1). A figura 5.3 representa a dinâmica de todo o mecanismo de configuração: a roxo estão representados os dois parâmetros de entrada, a verde o parâmetro que resulta da sua execução e a vermelho as duas principais funções executadas pelo microcontrolador.

O mecanismo inicial de configuração executa dois movimentos independentes. O primeiro, estende totalmente o atuador; e, o segundo, de forma contrária, retrai totalmente o atuador. Durante o primeiro movimento, o mecanismo apenas executa leituras contínuas da corrente consumida pelo atuador através da função *current sense* até que a leitura indique uma interrupção na alimentação do motor devido à ativação de um dos sensores de fim de curso. Por outro lado, durante o segundo movimento, o mecanismo monitoriza a corrente consumida até que o outro sensor de fim de curso seja ativado e, paralelamente, executa uma contagem dos pulsos digitais provenientes de cada um dos dois sensores de efeito de *Hall* que o motor integra.

Os algoritmos que permitiram a implementação deste mecanismo de configuração serão descritos de forma mais pormenorizada numa secção seguinte, 5.5.2.

5.3 Estratégia para controlo de posição

O controlo de posição do atuador implementado pretende atingir dois principais objetivos: a monitorização da posição atual do atuador e o posicionamento do atuador através do controlo da direção e do deslocamento de cada movimento. A monitorização da posição atual é utilizada em todos os modos de operação, ao invés que o posicionamento é apenas utilizados nos dois modos avançados (*advanced extended* e *advanced refracted*).

A posição absoluta do atuador é calculada pelo produto entre a razão *pulse rate* e o valor ponderado de dois contadores globais que monitorizam os pulsos digitais dos sensores de efeito *Hall*. A extensão do atuador permite que cada novo pulso resulte num novo incremento do respetivo contador e, pelo contrário, durante a retração do atuador, cada pulso resulta num decremento do contador respetivo. No final da execução do mecanismo de configuração inicial e do mecanismo de *reset* os dois contadores são reiniciados com o valor zero de forma a sincronizar novamente a posição absoluta do atuador e o valor dos dois contadores globais.

O posicionamento do atuador numa dada posição é uma função da unidade de controlo que permite definir a direção e o deslocamento de cada movimento do atuador. A definição da direção do movimento é conseguida pelo controlo do estado dos transístores que integram a unidade de comutação enquanto que a definição do deslocamento é realizada pela estimação do valor esperado para cada um dos contadores, em primeiro lugar e, seguidamente pela monitorização dos contadores até que o seu valor atinja o valor estimado.

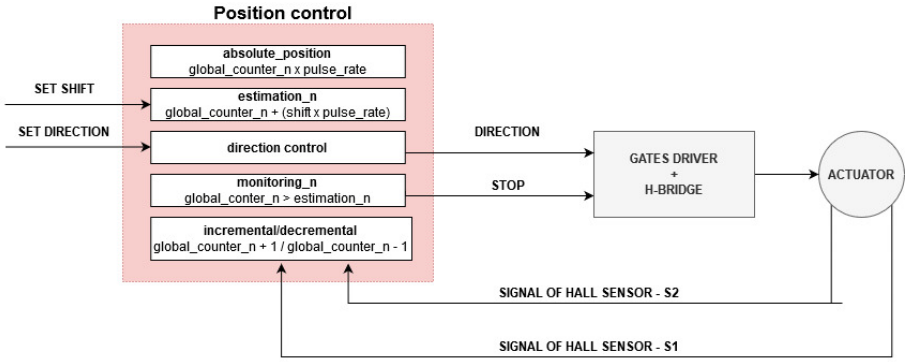


Figura 5.4: Funções básicas implementadas na estratégia para o controlo de posição

5.4 Estratégia para controlo de velocidade

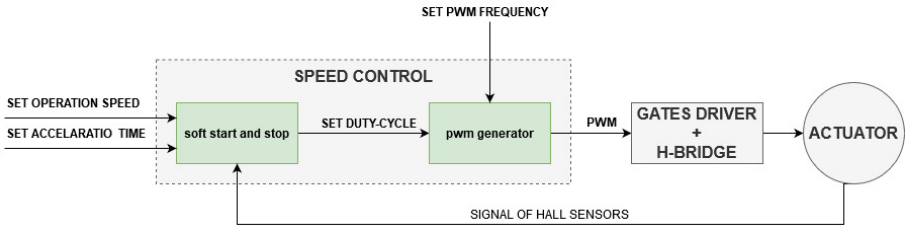


Figura 5.5: Controlo de velocidade

O controlo de velocidade resume-se a implementação de dois objetivos: a definição da velocidade de operação do atuador e a implementação doo mecanismo de *soft start and stop* que pode ser ativo e desativo durante cada movimento executado pelo atuador. A implementação de toda a estratégia teve em consideração que a tensão nominal do atuador e as velocidade lineares nominais a diferentes cargas são valores conhecidos que constam nas *datasheets* de cada atuador linear.

O controlo da velocidade linear do movimento utiliza a técnica de modulação digital *PWM* e baseia-se na dualidade tensão/velocidade característica deste tipo de motores. A relação *ratio speed* representa o rácio entre a velocidade pretendida para o acionamento de uma dada carga e a velocidade nominal. Desta forma, o atuador é alimentado com uma tensão média que se relaciona com a tensão nominal na proporção *ratio speed* através da técnica *PWM* em que o *duty-cycle* da onda corresponde à relação *ratio speed*.

O mecanismo de *soft start and stop* (figura 5.6) permite uma arranque e uma paragem suave do movimento através da definição do tempo de aceleração/desaceleração. Quando este mecanismo está ativo e o atuador se encontra parado, o movimento é iniciado com uma aceleração constante até que seja atingida a velocidade de operação definida. Quando, por outro lado, é processada uma ordem de paragem, o atuador desacelera de forma constante até parar totalmente o movimento. A suavidade do mecanismos está relacionada diretamente com o tempo definido para a aceleração. Assim, quanto maior o tempo disponível para acelerar/desacelerar, maior será o tempo aceleração

contínua do movimento e, por consequência, maior a suavidade do arranque/paragem do movimento.

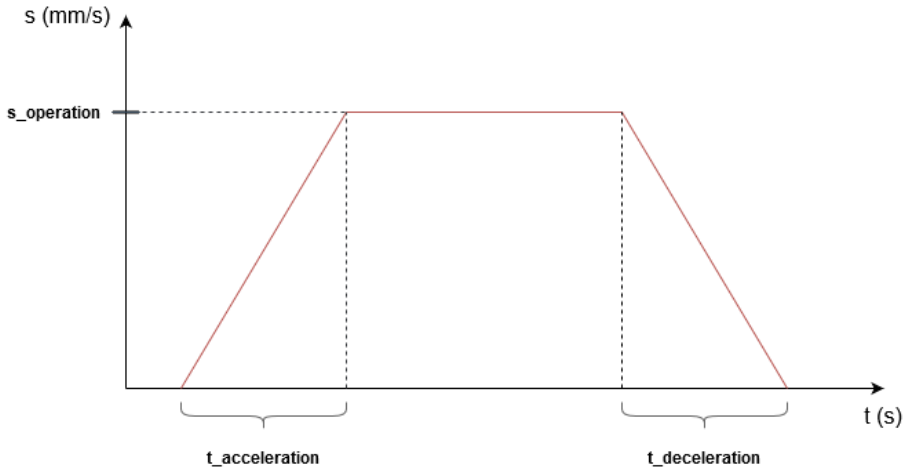


Figura 5.6: Mecanismo de *soft start and stop*

A técnica *PWM* requer a definição da frequência e o *duty-cycle* da onda. A frequência da onda é um parâmetro definido pelo nível de corrente consumida pelo atuador, ao invés que o *duty-cycle* é um parâmetro variável que permite aumentar/diminuir a tensão média de alimentação. A gama de valores para a frequência está definida e justificada na secção 5.1.1

5.5 Modos de operação

A unidade de controlo é responsável por executar diferentes movimentos e mecanismos de controlo sendo, por isso, modelada de forma a funcionar em diferentes modos de operação. A definição do modo de operação da unidade é desencadeada por comandos recebidos através da comunicação *SPI* estabelecida com a unidade central. Os comandos recebidos são também responsáveis por definir parâmetros de configuração do funcionamento da unidade de controlo e por monitorizar o estado do atuador em tempo real.

A implementação da unidade de controlo baseia-se em oito modos de operação agrupados consoante a sua função principal. Relativamente à figura 5.7, os modos de operação representados com a cor verde são responsáveis por produzir diferentes movimentos individualizados; com a cor vermelha são responsáveis por desencadear mecanismos de configuração e *reset* e com a cor azul são responsáveis por realizar as trocas de informação com a unidade central através da comunicação *SPI* bidirecional. A máquina de estados, figura 5.7 representa o encadeamento dos modos de operação condicionado pelos comandos *SPI* recebidos e por *flags* que sinalizam a conclusão de movimentos e mecanismos. Nas secções seguintes serão apresentadas detalhadamente o funcionamento de cada modo de operação.

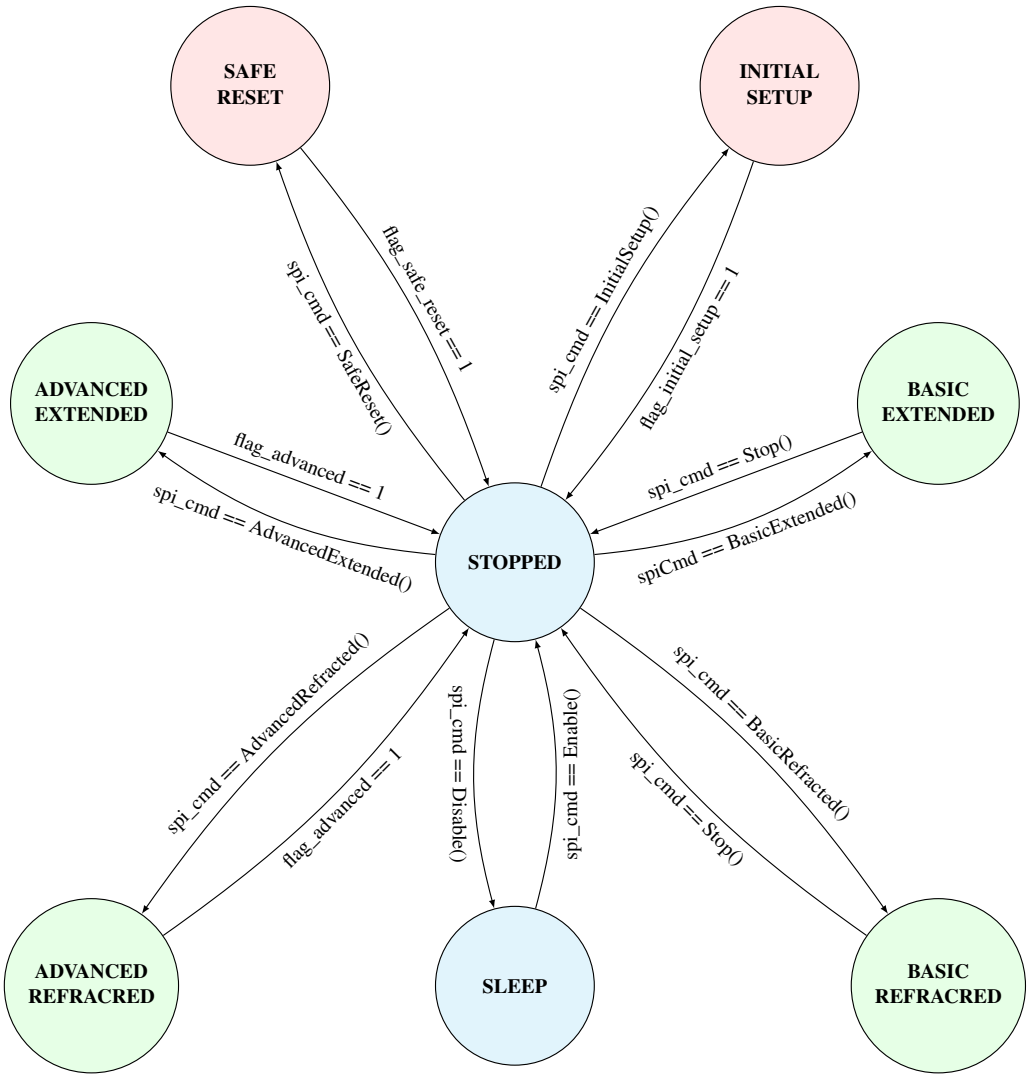


Figura 5.7: Modos de operação da unidade de controlo

5.5.1 Considerações gerais da implementação

O *firmware* implementado para a correta operação da unidade de controlo foi desenvolvido em linguagem *C* utilizando uma plataforma *open-source*, *PlatfomrIO IDE for VSCode*, que permite compilar e descarregar o *firmware* diretamente para o microcontrolador através da porta série. Esta plataforma permite configurar e gerir facilmente a comunicação série de forma a monitorizar o funcionamento da unidade durante a fase de desenvolvimento e a fase da integração no sistema global.

5.5.1.1 *Current Sense*

O mecanismo *current sense* da unidade de comutação produz uma corrente proporcional à corrente consumida pelo atuador durante os momentos de operação normal. Esta corrente é regulada por um multiplexador que permite selecionar o amplificador consoante o nível de potência

da unidade. A corrente produzida por este mecanismo, *I_sense*, pode ser convertida numa tensão (*V_sense*) através de uma resistência externa, *R_sense*. Através desta conversão e dos canais *ADC* do microcontrolador é possível monitorizar a corrente consumida pelo atuador. A resistência *R_sense* deve ser dimensionada de forma a garantir a linearidade para toda a faixa de valores da corrente consumida pelos atuadores. Por outro lado, de forma a proteger a integridade do atuador, a tensão *V_sense* não deve exceder os 5V (equação 5.5.1.1). É necessário, também, dimensionar uma resistência em série com a entrada analógica do microcontrolador para impedir que flua uma corrente superior a 40mA. O integrado *VNH5019* possui um mecanismo que limita a queda de tensão na resistência *R_sense* em 7.5V (5.5.1.1).

$$I_{sense} = I_{out}/k \tag{5.8}$$

$$k_1@3A = 4670(min) \rightarrow I_{sense} = 642\mu A \tag{5.9}$$

$$k_2@8A = 6060(min) \rightarrow I_{sense} = 1320\mu A \tag{5.10}$$

$$k_3@15A = 6000(min) \rightarrow I_{sense} = 2500\mu A \tag{5.11}$$

$$V_{sense} = R_{sense}I_{sense} \tag{5.12}$$

$$\begin{aligned} V_{sense} < 5V \rightarrow R_{sense}I_{sense} < 5 \\ R_{sense}@3A < 7,8k\Omega \\ R_{sense}@8A < 3,8k\Omega \\ R_{sense}@15A < 2k\Omega \\ R_{sense} < 2k\Omega \end{aligned} \tag{5.13}$$

$$\begin{aligned} V_{sense,max} = 7.5V \\ I_{sense,max} < 40mA \\ I_{sense} < V_{sense,max}/R_{prot} \\ R_{prot} > 187,5\Omega \end{aligned} \tag{5.14}$$

5.5.1.2 *Tipo e tamanho de variáveis*

Para a implementação dos diferentes modo de operação foi necessário ponderar questões de *software* relacionadas com o tipo e o tamanho de algumas variáveis. Assim, na tabela 5.5 estão reunidas os requisitos para principais variáveis utilizadas na implementação com referencia ao parâmetro do sistema que representam.

Parâmetro	Requisito máximo	Tipo	Valor mínimo	Valor máximo
stroke_lenght	1 000 *mm*	*uint16_t*	0	65 535
pulse_rate	23 *pulses/mm*	*uint8_t*	0	127
position_absolute	1 000 *mm*	*uint16_t*	0	65 535
counter_pulse_s1/counter_pulse_s2	23 000 *pulses*	*uint16_t*	0	65 535
time_accelaration	6 000 *ms*	*uint16_t*	0	65 535
ratio_speed	100 %	*uint8_t*	0	127

Tabela 5.5: Requisitos da implementação

5.5.2 *Inital Setup*

O modo *Inital Setup* tem como principal objetivo executar a parametrização e a configuração inicial pata adaptar as estratégias de controlo ao modelo de atuador linear em causa. Deste mecanismo resulta a execução de dois movimentos independentes, o primeiro de extensão e o segundo de retração, com o objetivo de calcular a razão *pulse_rate* e sincronizar os contadores globais dos sensores de efeito *Hall* e a posição absoluta do atuador. No final deste mecanismo, o atuador encontra-se totalmente retraído e, portanto, a sua posição absoluta é iniciada com o valor nulo bem como os contadores globais (figura 5.8).

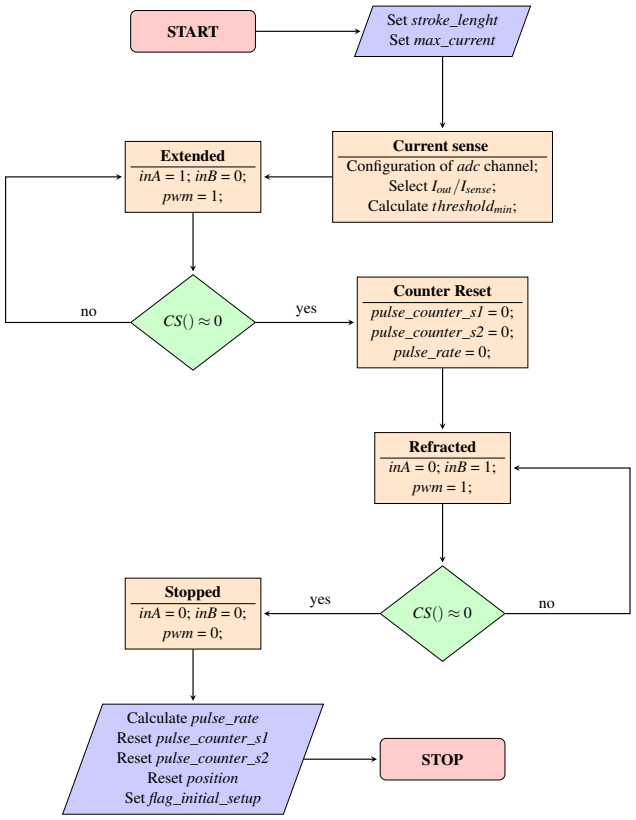


Figura 5.8: *Initial Setup*

5.5.3 *Safe Reset*

O mecanismo de *safe reset* é um mecanismo de segurança que permite retrair totalmente o atuador e sincronizar a posição do atuador com a unidade central em situações de recuperação após uma falha inesperada do sistema. Este mecanismo executa um mecanismo de retração total do atuador. Depois da execução do movimento, a unidade central sincroniza o seu estado com a unidade central através da transmissão da posição atual do atuador (figura 5.9).

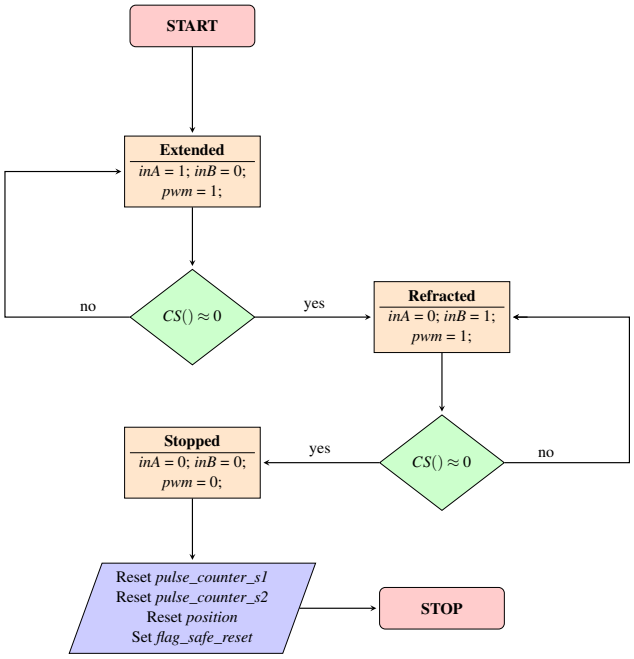


Figura 5.9: *Safe Reset*

5.5.4 *Basic Extended / Basic Refracted*

Os modos de operação *Basic Extended* e *Basic Refracted* permitem estender/retrair o atuador continuamente. Este modo é despoletado pela recepção de uma ordem de extensão/retração através da comunicação *SPI* estabelecida com a unidade central, *spi_cmd == BasicExtended() / BasicRefracted()*. A extensão/retração do atuador é terminada com a recepção de uma ordem de paragem, *spi_cmd == Stop()*, ou, no momento em que o atuador atinge a extensão/retração máxima (figura 5.10).

5.5.5 *Advanced Extended / Advanced Refracted*

Os modos de operação *Advanced Extended* e *Advanced Refracted* permite definir o deslocamento do atuador durante um movimento de extensão/retração. Antes iniciar o movimento, o mecanismo através da razão *pulse_rate* e do deslocamento pretendido estima o número de pulsos dos sensores de efeito *Hall* que a unidade de controlo deve contar durante o movimento do

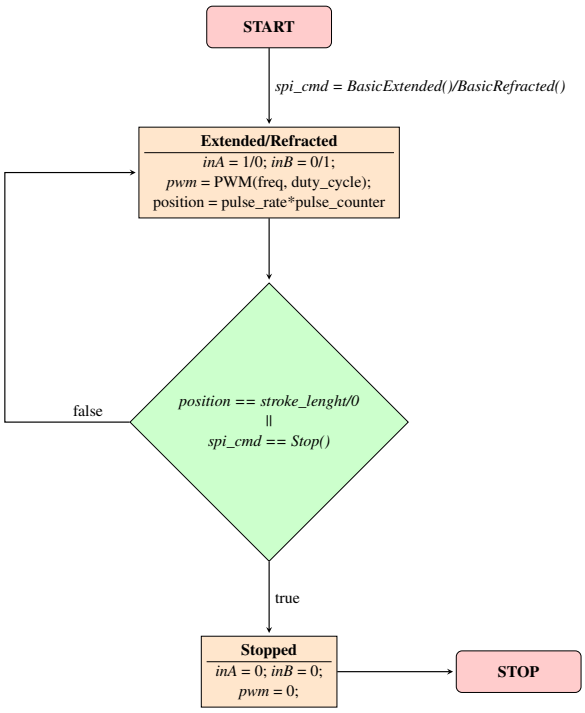


Figura 5.10: *Basic Extended/Refracted*

atuador e a posição final do atuador. Se a posição estimada coincidir com a posição totalmente estendido/retraído, o mecanismo redefine o deslocamento a efetuar pelo atuador considerando a sua posição atual. No final desta estimativa, o atuador é estendido/retraído até que a posição esperada para o final do movimento seja atingida (figura 5.11).

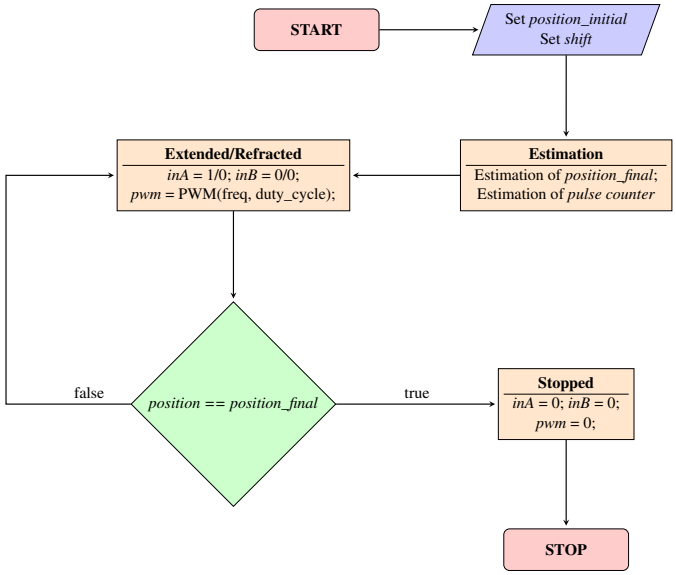


Figura 5.11: *Advanced Extended/Refracted*

Capítulo 6

Comunicação SPI

O presente capítulo pretende apresentar o princípio de funcionamento do protocolo de comunicação *serial peripheral interface* (secção 6.1), a arquitetura *master-slave* implementada (secção 6.1) e, por fim, apresentar a estrutura geral dos vetores de comandos, bem como os mecanismos responsáveis pela sua concepção e interpretação (secção 6.2).

6.1 Introdução - Princípio de funcionamento

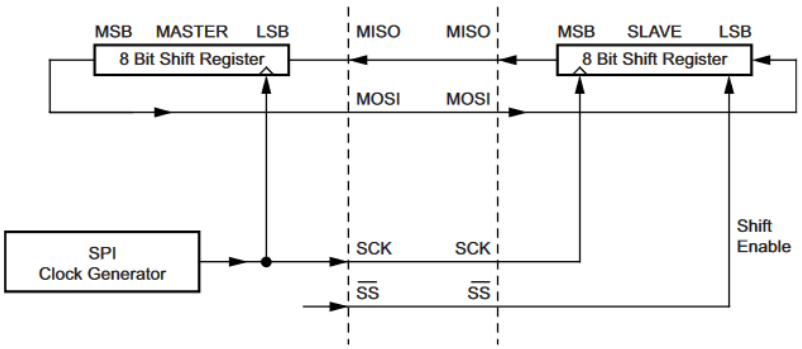


Figura 6.1: Arquitetura *master-slave*[9]

O protocolo *SPI* é um protocolo de comunicação síncrona pois o seu funcionamento resulta do deslocamento sucessivo de *bits* ao ritmo de um sinal de relógio partilhado pelos dispositivos intervenientes. O dispositivo responsável por gerar o sinal de relógio desempenha o papel de *master* e, todos os restantes o papel de *slave*. Cada dispositivo interveniente nesta comunicação possui um registo de deslocamento interno que se encontram ligados entre si através de duas linhas de dados, *MISO* e *MOSI*, figura 6.1. Embora este protocolo permita uma comunicação bidirecional, a comunicação é sempre desencadeada pelo dispositivo *master*. A velocidade de transmissão é definida pela velocidade do sinal de relógio enquanto que o momento de captura (recepção) e amostragem (transmissão) de cada *bit* é definido pela polaridade e pela fase do sinal de relógio. O protocolo permite a integração de múltiplos dispositivos *slaves*, contudo apenas permite comunicar com um

slave de cada vez através da ativação da linha de seleção (*SS*) correspondente. Quando uma linha *SS* é ativa (pelo regresso ao estado *low*), o dispositivo *slave* coloca-se num modo de escuta e, portanto, os dados da linha *MOSI* podem ser capturados e/ou novos dados podem ser amostrados para a linha *MISO*. Teoricamente, o número de dispositivos *slaves* é ilimitado, contudo, a capacidade de carga pode influenciar a precisão com que os níveis lógicos nas linhas de dados são definidos pelo dispositivo *master*.

6.2 Implementação - Vetores de Comandos

As informações trocadas entre a unidade central e as unidades de controlo têm como principais objetivos: a configuração individual de cada unidade de controlo, a definição do movimento a executar por cada unidade de controlo e a monitorização dos estados dos atuadores do protótipo em desenvolvimento. Desta forma, a comunicação foi agrupado em cinco modos distintos consoante a informação que transmitem (tabela 6.1).

Modo	Descrição
Setup()	· Transmitir à unidade de controlo o último estado conhecido do atuador: posição e *pulse_rate* · Ligar/desligar o mecanismo de *soft start and stop* · Transmitir à unidade de controlo o tempo de aceleração · Transmitir à unidade de controlo a velocidade de operação
Update()	· Transmitir à unidade central a posição atual do atuador · Transmitir à unidade central a conclusão de um movimento avançado · Transmitir à unidade central a conclusão do mecanismo de *safe reset/inital setup* · Transmitir à unidade central o valor estimado da razão *pulse rate*
Basic()	· Iniciar o mecanismo de *safe reset* · Iniciar o mecanismo de *initial setup* · Iniciar o movimento de extensão do atuador · Iniciar o movimento de retração do atuador · Parar o movimento atual do atuador
Advanced()	· Iniciar o movimento de extensão do atuador com um deslocamento definido · Iniciar o movimento de retração do atuador com um deslocamento definido
Communcation()	· Controlo da arquitetura *master-slave* · Despoletar a comunicação com uma unidade de controlo de cada vez · Colocar a unidade de controlo no estado *sleep*

Tabela 6.1: Descrição dos modos de comunicação

No sistema global implementado a comunicação *SPI* estabelecida é o elemento responsável

pela manutenção do fluxo de informação entre as unidades que compõem o sistema. Neste contexto, o papel de *master* é desempenhado pelo *SBC* incluído na unidade central, enquanto que o papel de *slave* é desempenhado por todos os microcontroladores integrados nas unidades de controlo. Desta forma, a unidade central consegue garantir uma comunicação independente com cada unidade de controlo. Cada momento de comunicação entre a unidade central e uma das unidades de controlo é caracterizada pelo deslocamento de vários *bytes* nas linhas *MOSI* e *MISO*. Ao conjunto de *bytes* trocados chamamos vetores de comandos. O comprimento do vetor de comandos é definido pelo modo da comunicação (tabela 6.1).

É importante que a estrutura dos vetores de comandos esteja bem definida, de forma a permitir a correta interpretação das informações trocadas entre as duas unidades do sistema. De forma a facilitar a interpretação dos vetores da comunicação, os vetores transmitidos pela unidade central (e, portanto, recebidos pela unidade de controlo) foram denominados de *TxCommands*; por outro lado, os vetores recebidos pela unidade central (e, portanto, transmitidos pela unidade de controlo) foram denominados de *RxCommands*.

A estrutura do vetor *TxCommands* é, de forma generalizada, composta por um *byte* que assinala o início do vetor, um *byte* que indica o modo da comunicação, um *byte* com o código hexadecimal do comando, um *byte* opcional com informação extra e um *byte* a assinalar o fim do vetor. A estrutura geral dos cinco modos está definida detalhadamente no anexo C, figura C.1.

Os vetores *RxCommands* permitem, principalmente, à unidade central verificar a recepção correta dos vetores *TxCommands* pela unidade de controlo e monitorizar o estado atual do atuador. Assim, a unidade de controlo transmite um *byte* de sucesso sempre que recebe um novo *byte* de dados, excepto no modo *Update()* em que transmite uma informação sobre o estado do atuador quando recebe o código hexadecimal do comando *SPI*. A figura C.2, anexo C, reúne a estrutura esperada dos vetores *RxCommands* em resposta à recepção dos diferentes tipos de vetores.

A unidade de controlo está dotada de um mecanismo que permite a interpretação dos vetores recebidos através da recuperação dos *bytes* de dados armazenados num *buffer* auxiliar. Este mecanismo é, também, responsável por gerar os vetores*RxCommands* que constituem a resposta da unidade de controlo aos vetores recebidos *TxCommands*. Cada *byte* deslocado nas linhas *MOSI* e *MISO* desencadeia uma interrupção no microcontrolador que despoleta o mecanismo de interpretação. Este mecanismo permite a ativação de uma *flag* para assinalar a recepção completa de um vetor e, desta forma, permitir à unidade de controlo executar o novo movimento e/ou alterar as definições de funcionamento. O algoritmo C.1, anexo C, descreve a implementação deste mecanismo.

A unidade central, por sua vez, está dotada de um mecanismo que permite a construção dos vetores *TxCommands* e a verificação dos vetores *RxCommands*. Este mecanismo foi implementado à custa de funções parametrizadas que constroem os vários tipos de vetores e avaliam a resposta da unidade de controlo. Os algoritmos C.2, C.3, C.4, C.5 e C.6 no anexo C descrevem o funcionamento das funções de construção.

Capítulo 7

Unidade Central

A unidade central é o elemento do sistema global que permite centralizar a informação relativa aos vários atuadores que compõem o dispositivo médico em desenvolvimento e realizar o papel de *interface* com a equipa de *R&D*. O presente capítulo pretende expor os aspetos estruturais (secção 7.1) e os aspetos funcionais (secção 7.2) da unidade central.

7.1 Aspetos Estruturais

A unidade central é composta por um *single board computer* responsável por estabelecer a comunicação *SPI* e alojar a plataforma de interface com a equipa de *R&D*.

7.1.1 Unidade de processamento - *Single board computer*

O *SBC* escolhido para a implementação da unidade central foi o modelo *Raspberry 3 Model B+*, uma vez que este modelo permite implementar o protocolo de comunicação série *SPI* e o acesso à plataforma de desenvolvimento através de acesso remoto. A escolha deste *SBC* foi reforçado por todo suporte e documentação existente deste equipamento em diversos projetos e soluções.

7.1.2 Tecnologias

7.1.2.1 *QT*

A *interface* gráfica da aplicação de apoio à equipa de *R&D* foi desenvolvida na *framework QT*, uma *framework* que permite o desenvolvimento de interfaces gráficas em *C++*. A principal vantagem do *QT* é a possibilidade configurar um *cross-compile* facilitando o desenvolvimento da interface gráfica. *Cross-compiler* é um processo utilizado no desenvolvimento de sistemas embarcados que permite gerar executáveis numa plataforma e executá-los numa plataforma diferente. Nesta implementação, o executável é gerado por uma máquina *Ubuntu* alocada num *PC* e executado numa máquina *Raspbian* alocada na *Raspberry Pi* que integra a unidade central, figura 7.1.

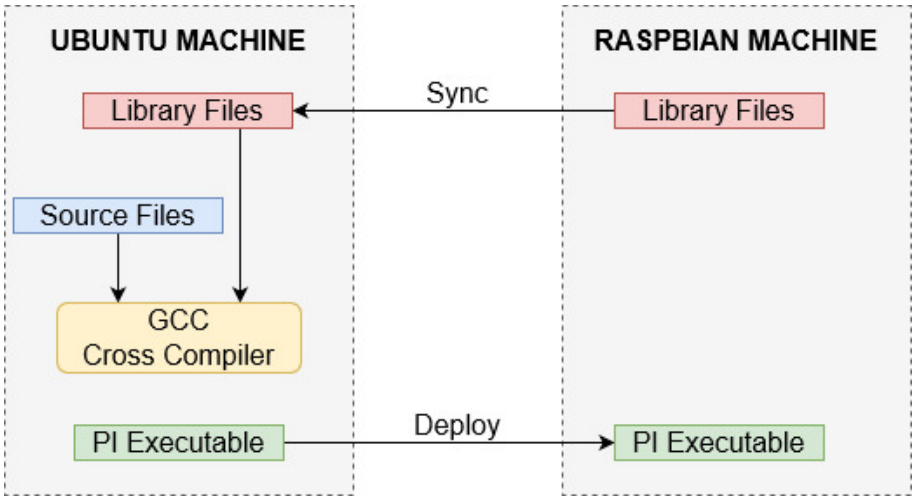


Figura 7.1: *Cross-compiler*[10]

7.1.2.2 Formato *JSON*

JSON é um formato *standard* de ficheiros de texto que permite a partilha e armazenamento de dados. Este formato permite um leitura mais legível por humanos quando comparado com o formato *XML*. A semântica deste formato está associada a um par de dados, o nome e o respetivo valor de cada parâmetro. O nome é sempre uma variável do tipo *string*, enquanto que o valor pode representar variáveis de vários tipos (*int*, *double*, *bool* e *string*) ou objetos de diferentes classes do sistema.

O sistema implementado integra vários ficheiros *JSON*, um deles com o objetivo de armazenar as caraterísticas e o estado de cada atuador que compõem o dispositivo médico (*RecordsOfChannels.js*) e os restantes com o objetivo guardar todas as sequências de movimentos definidas para o dispositivo médico (*RecordsMoves_x.js*, em que o valor de *"x"* representa o nome dado à sequência). A estrutura de cada um dos ficheiros está apresentada no anexo E.

7.1.2.3 Acesso Remoto

- ***SSH***: Permite apenas o acesso à linha de comandos da *Raspberry PI* através de outro dispositivo com quem partilha a rede. Utilizado em momentos de gestão e configurações mais simples do ambiente e propriedades da *Raspberry Pi*.

- ***VNC***: Permite controlar remotamente o ambiente gráfico da *Raspberry Pi* através de outro dispositivo. Utilizado, principalmente, para executar a plataforma gráfica.

Numa primeira abordagem (figura 7.2 em cima), o acesso remoto foi conseguido pela conexão da *Raspberry Pi* e do dispositivo de acesso (um computador básico) à mesma rede *Wireless*. Contudo, devido às condições de acesso a algumas redes *Wireless* optou-se por uma ligação *Ethernet* entre a *Raspberry Pi* e um dispositivo que permite acesso por *Wireless*.

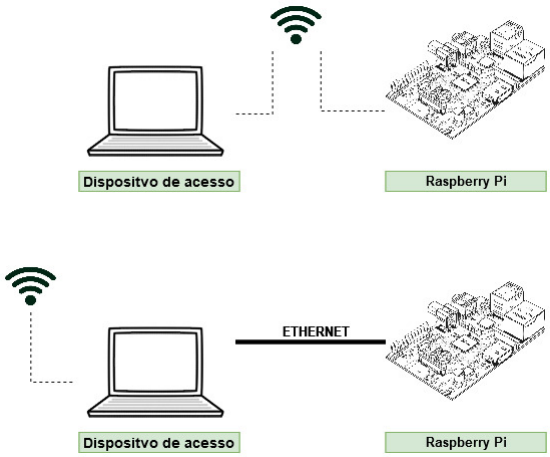


Figura 7.2: Abordagens para o controlo por acesso remoto

7.2 Aspetos Funcionais

A *interface* gráfica desenvolvida permite controlar e configurar diferentes mecanismos e movimentos que a solução implementada integra. A figura 7.3 representa as principais interacções que a equipa de *R&D* pode tomar com a unidade central.

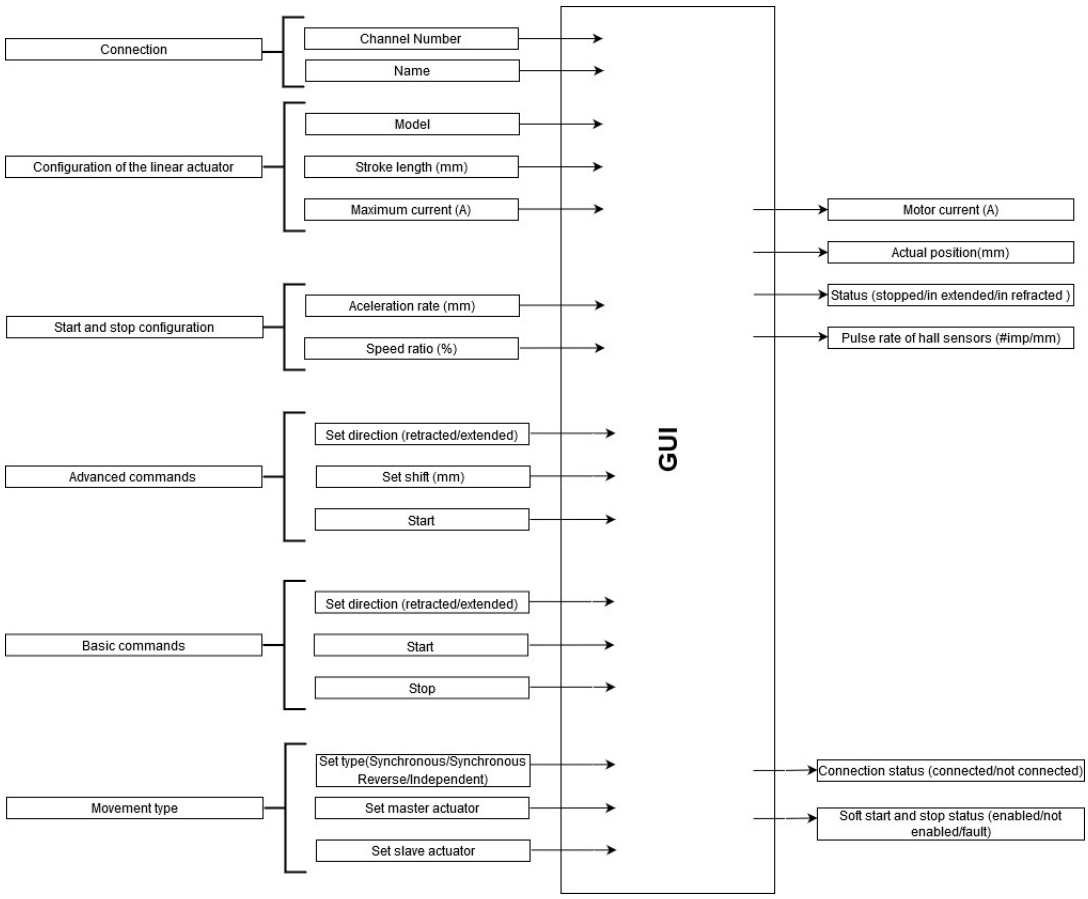


Figura 7.3: Interacções principais da *interface* gráfica

De forma a tornar a interação entre a unidade central e a equipa de *R&D* intuitiva e de fácil aprendizagem foram implementados separadores com *design* e objetivos distintos na *interface* gráfica. Os separadores de controlo individual, um por atuador que o dispositivo médico integra, permitem controlar, configurar e monitorizar de forma individual cada unidade de controlo e o respetivo atuador linear (secção 7.2.1). O separador de controlo global com um *desgin* inspirado na operação de um comando de controlo permite definir cada movimento a ser executado por qualquer atuador ou par de atuadores que constituem o dispositivo médico (secção 7.2.2). Por fim, foi implementado um separador que tem como objetivo a definição de sequências de movimentos a serem executadas pelo protótipo em desenvolvimento (secção 7.2.3). As sequências definidas neste separador podem ser guardadas para uma reutilização futura ou para a realização de testes de repetição.

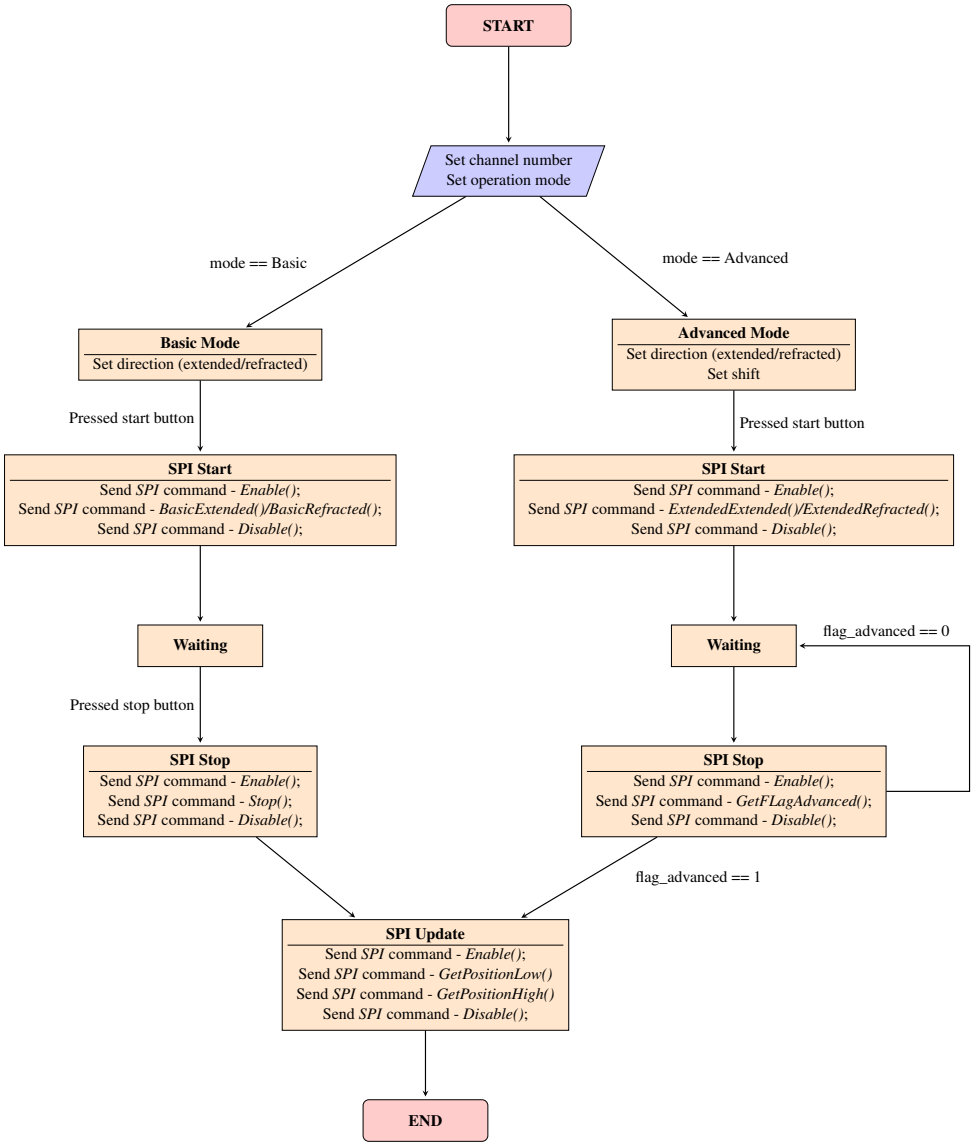


Figura 7.4: Interacão entre a unidade central e as unidades de controlo

De forma a distinguir o acionamento simultâneo dos atuadores do dispositivo foi necessário classificar os movimentos relativamente à operação independente ou paralela dos atuadores. Assim, o movimento é do tipo independente quando apenas um atuador linear é deslocado e do tipo síncrono quando dois atuadores lineares executam um movimento em paralelo.

De forma a clarificar a interação entre a unidade central e as unidades de controlo é importar recordar os modos de operação das unidades de controlo. Assim, nos separadores responsáveis pelo controlo individual é possível desencadear qualquer modo de operação do atuador: *Initial Setup*, *Safe Reset*, *Basic Extended*, *Basic Refracted*, *Adavanced Extended* e *Advanced Refracted*. Por sua vez, no separador responsável pelo controlo global apenas é possível despoletar os modos básicos em cada unidade de controlo: *Basic Extended* e *Basic Refracted*. E, por fim, no separador responsável pela definição de sequências apenas os modos avançados serão despoletados: *Adavanced Extended* e *Advanced Refracted*. Nos modos básicos é necessário selecionar a direção do movimento e definir qual dos atuadores do dispositivo médico será acionado. Por sua vez, nos modos avançados além da seleção da direção e do atuador, é necessário definir o deslocamento efetuado durante o movimento pelo atuador. As interações entre a unidade central e as unidade de controlo baseiam-se na nos comandos *SPI* implementados (figura 7.4).

7.2.1 Controlo individual

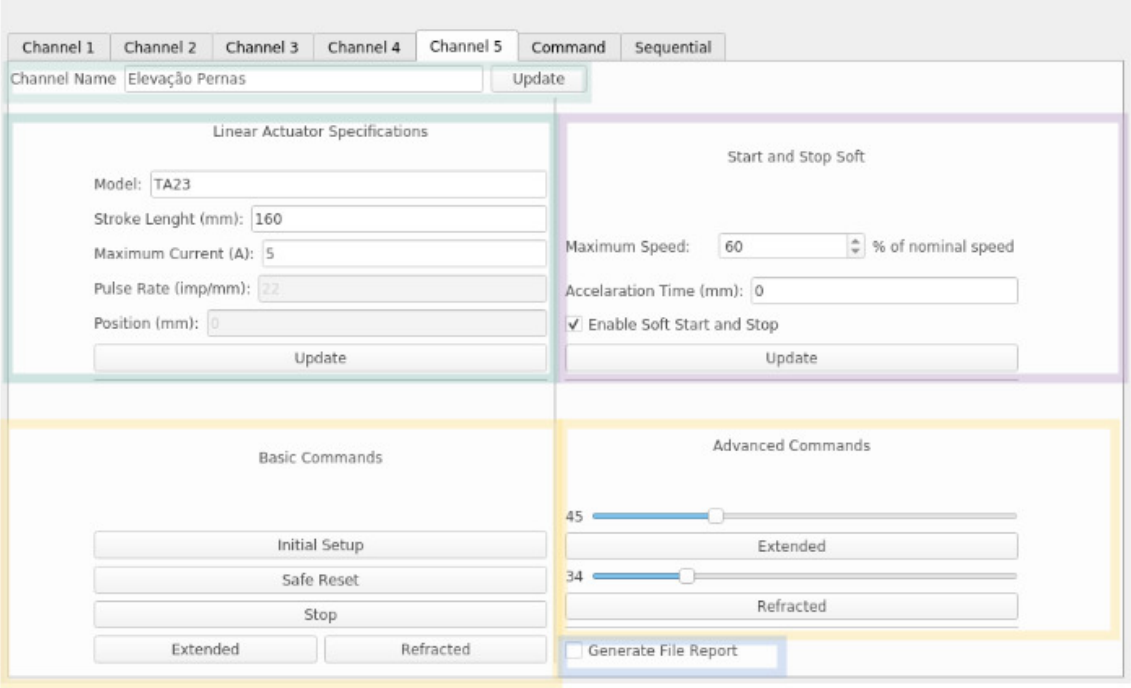


Figura 7.5: Separador de controlo individual

Os separados de controlo individual (figura 7.5) foram desenhados de forma a cumprir alguns objetivos. O grupo representado com a cor verde, figura 7.5, permite introduzir e consultar parâmetro caraterísticos do atuador e a monitorizar o estado atual do atuador. O grupo sinalizado com

a cor roxa, por sua vez, permite configurar o funcionamento do mecanismo de *soft start and stop* e ativar/desativar o mecanismo sempre que necessário. O grupo representado pela cor amarela permite despoletar um novo movimento do atuador do tipo básico ou avançado e, também, um mecanismo de configuração (*InitialSetup()*) ou de recuperação (*SafeReset()*). Foram implementados cinco separadores deste tipo, uma vez que a plataforma deve suportar dispositivos médicos que integram até um máximo de cinco atuadores. Quando um canal se encontra desconetado da unidade de controlo respetiva, o seu separador encontra-se bloqueado e nenhum operação pode ser efetuada nele.

7.2.2 Controlo global

O separador de controlo global permite definir movimentos a serem executados por qualquer atuador ou pares de atuadores que integram o protótipo do dispositivo em desenvolvimento.

O *design* do funcionamento deste separador baseou-se na operação de comando de controlo remoto. Assim, este separador permite a configuração dos dois tipos de movimentos implementados (movimentos independentes e movimentos síncronos) no modo de operação básica (figura 7.6).

Assim, nos movimentos independentes sempre que um dos botões de extensão ou retração é pressionado o atuador correspondente executa um movimento de extensão/retração até que o botão seja largado. Por sua vez, nos movimentos síncronos existem duas situações distintas. Se o sincronismo inverso estiver ativo, sempre que um dos botões de extensão ou retração é pressionado, o atuador seleccionado para o papel de *master* executa o movimento correspondente ao botão pressionado, enquanto que o atuador com o papel *slave* executa um movimento contrário ao do indicado pelo botão pressionado. Nas situações em que o sincronismo inverso está desativo, ambos os atuadores executam o movimento correspondente ao botão pressionado. Em ambas as situações, o movimento é interrompido quando o botão que se encontra pressionado é largado.

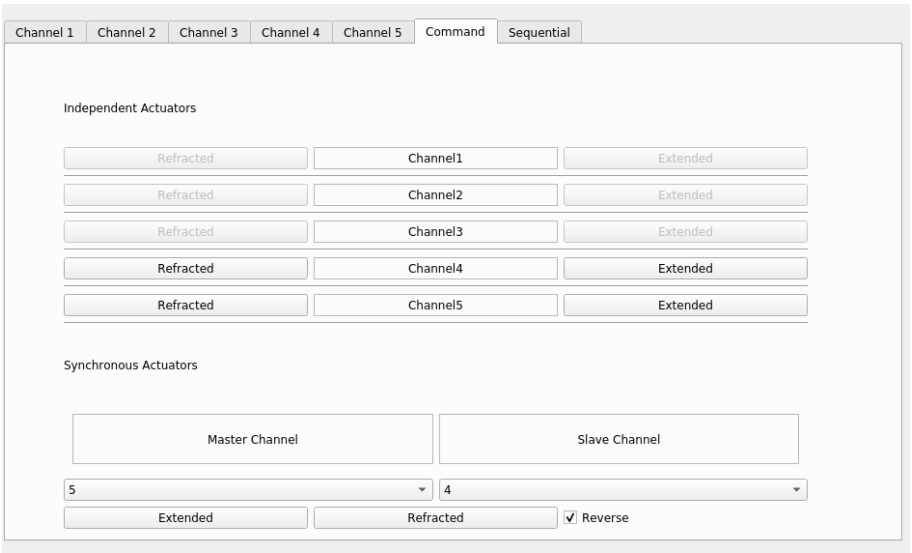


Figura 7.6: Separador de controlo global

7.2.3 Definição de sequências e testes de repetibilidade

O separador responsável pela definição de sequências apresenta um *design* modular que permite encadear movimentos sucessivos até a um máximo de 10 movimentos por sequência. As sequências definidas podem ser guardados de forma a serem reutilizadas ou modificadas no futuro. Este separador apenas permite implementar os movimentos no modo de operação avançado.

O *design* inicial deste separador permite definir uma nova sequência (figura 7.7 - cor vermelha) ou, por outro lado, reutilizar uma anteriormente definida (figura 7.7 - cor azul). Sempre que for necessário acrescentar um novo movimento à sequência é necessário definir o tipo de movimento (independente, sincronismo, sincronismo inverso), selecionar o atuador ou o par de atuadores que irão executar o movimento, a direção do movimento do atuador *master* (extensão ou retração) e o deslocamento pretendido do atuador *master* (figura 7.7 - cor verde). Os testes de repetibilidade da sequência são definidos pela sequência a executar e o número de repetições sucessivas a realizar (figura 7.7 - cor vermelha roxa).



Figura 7.7: Separador de definição de sequências

Capítulo 8

Resultados - Aplicação Prática

O presente capítulo tem como principais objetivos a apresentar o ambiente de avaliação do sistema global implementado (secção 8.1), os resultados obtidos (secção 8.2) e uma reflexão do trabalho desenvolvido e futuro (secção 8.3).

8.1 Apresentação do protótipo da cama articulada

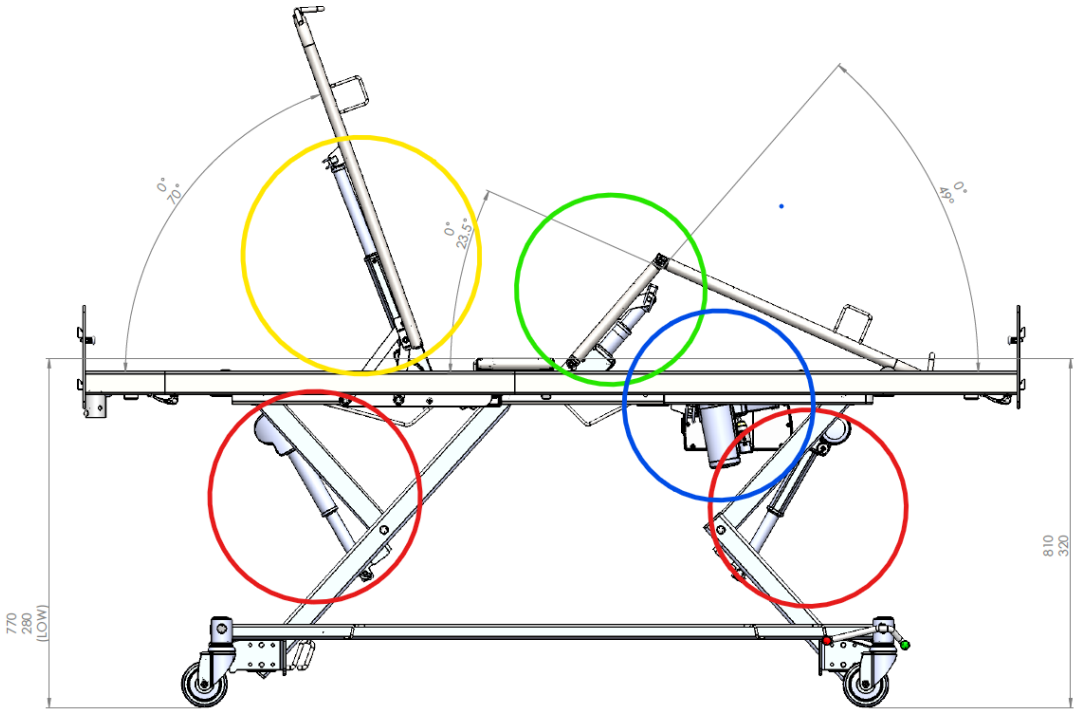


Figura 8.1: Protótipo da cama articulada

O sistema de controlo para o desenvolvimento de dispositivos médico foi implementado e testado num protótipo de uma cama articulada que engloba cinco atuadores lineares responsáveis por diferentes funções e movimentos. A figura 8.1 e a tabela 8.1 resume as funções e as característica de cada atuador que compõem o protótipo em desenvolvimento.

Função	Cor	Modelo	Máxima Extensão	Máxima Carga
HiLo - Elevação e inclinação da cama	Vermelho	TA23	160 cm	8000 N
Backrest - Inclinação das costas	Amarelo	TA31	190 cm	5000 N
Thigh - Inclinação das coxas	Verde	TA31	45 cm	5000 N
Lower Leg - Inclinação pernas	Azul	TA31	50 cm	3500 N

Tabela 8.1: Resumo das caraterísticas do prototipo

8.2 Testes e Resultados

8.2.1 Unidade de Comutação - Avaliação Térmica

Numa primeira abordagem, a unidade de controlo foi avaliada de forma isolada para garantir o acionamento contínuo durante dois minutos de diferentes cargas com diferentes frequências das ondas *PWM*.

Os testes térmicos realizados utilizando cargas resistivas diferentes permitiram avaliar a evolução temporal da temperatura do *chip*. As cargas utilizadas foram escolhidas tendo em conta a tensão de alimentação nominal dos atuadores e as correntes máximas consumidas por estes em regime final. Assim, as cargas resistivas utilizadas têm um valor nominal de $15\Omega, 100W$ e foram incrementados sucessivos paralelos de forma a avaliar diferentes níveis de potência. As medições de temperatura foram auxiliadas por um termómetro infravermelho com as especificações listadas na tabela 8.2.

Especificação	Valor
Alcance	-40^{o}C até 500^{o}C
Precisão (Abaixo 0^{o}C)	$\pm$ 5^{o}C
Precisão (Acima 0^{o}C)	$\pm$ 2^{o}C
Tempo de Resposta	$\leq$ 0.8 segundos

Tabela 8.2: Especificações termómetro infravermelho

As condições de carga para cada de teste estão apresentadas na tabela 8.3 e o procedimento do teste encontra-se listado em baixo:

1. Colocar a unidade de comutação no modo *Brake to GND*;
2. Aguardar até que a temperatura do integrado estabilize;
3. Registar a temperatura inicial do integrado;

4. Colocar a unidade de comutação no modo *clockwise*;

5. Efectuar medições sucessivas a cada 30 segundo durante um intervalo de 120 segundos da temperatura do integrado;

Carga	**Corrente nominal**	**Potência dissipada total**	**Potência dissipada** (15Ω)
15Ω	$1,6A$	$38,4W$	$38,4W$
15Ω//15Ω	$3,2A$	$76,8W$	$38,4W$
15Ω//15Ω//15Ω	$4,8A$	$115,2W$	$38,4W$
15Ω//15Ω//15Ω//15Ω	$6,4A$	$153,6W$	$38,4W$

Tabela 8.3: Resumo das condições de teste

Para uma melhor análises das medições efectuados, os dados estão apresentados em quatro gráficos distintos, figuras 8.2, 8.3, 8.4 e 8.5. Da análise cuidada do conjunto de gráficos retiram-se as principais conclusões:

- Em nenhum teste a temperatura do integrado excedeu os 150ºC;
- A unidade de comutação permite acionar continuamente durante dois minutos qualquer um dos atuadores em estudo sem ativar nenhum dos mecanismos de segurança;
- A unidade de comutação consegue operar correctamente em toda a gama de frequências, de 4kHz até 20kHz;
- O integrado atinge temperaturas cada vez mais elevadas quando, acionando a mesma carga, opera com frequências sucessivamente maiores.

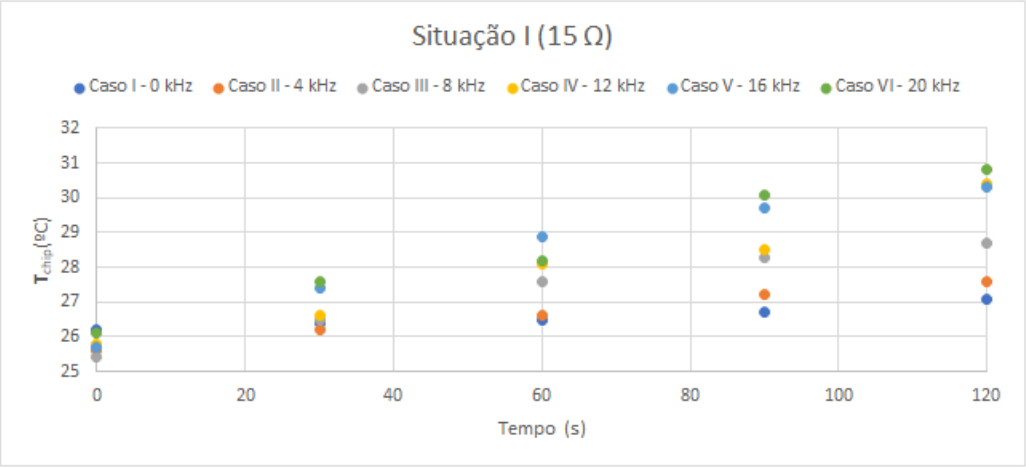


Figura 8.2: Situação I - Carga 15Ω

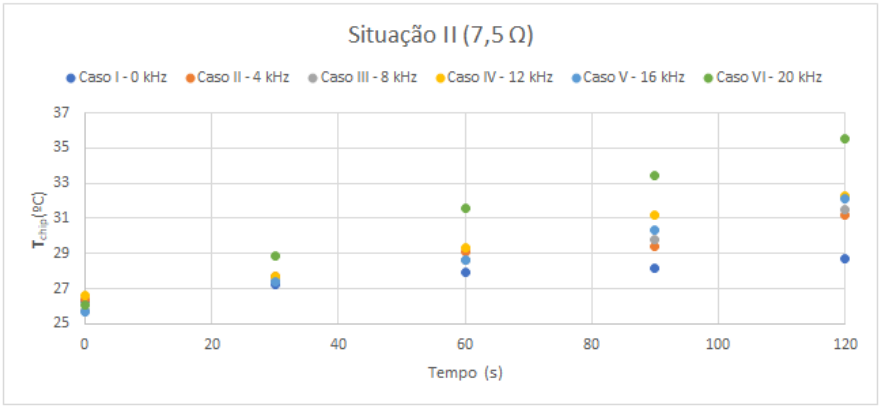


Figura 8.3: Situação II - Carga 15Ω//15Ω

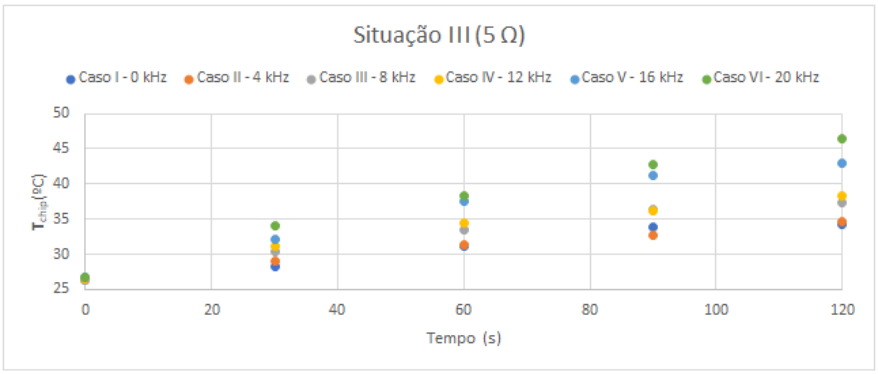


Figura 8.4: Situação III - Carga 15Ω//15Ω//15Ω

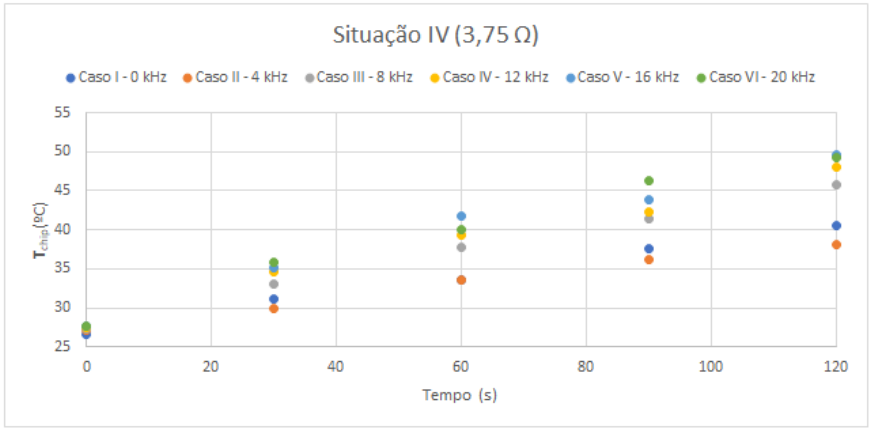


Figura 8.5: Situação IV - Carga 15Ω//15Ω//15Ω//15Ω

8.2.2 Avaliação do Desempenho

A avaliação do desempenho do sistema implementado foi realizada em três fases distintas. A primeira realizada apenas com dois atuadores isolados sem integrarem nenhum protótipo, teve como principal objetivo a avaliação da comunicação entre as unidades de controlo e unidade central. Numa segunda fase, testou-se o controlo de três atuadores integrados no protótipo apresentado na secção 8.1 sem carga, de forma a avaliar o desempenho das funções implementadas na unidade central. Por fim, na terceira fase, repetiu-se a avaliação feita na segunda fase mas desta vez o protótipo acciona uma carga máxima de 140Kg. A sequência de tarefas avaliadas estão descritas em baixo.

8.2.2.1 Primeira Fase

1. Desligar o mecanismo *soft start and stop* nos dois canais em utilização
2. Configuração e parametrização inicial do atuador linear do canal 1
3. Configuração e parametrização inicial do atuador linear do canal 2
4. Extensão do atuador linear do canal 1
5. Extensão do atuador linear do canal 2 durante 40cm
6. Retração do atuador linear do canal 2 durante 20cm
7. Colocar o atuadores lineares dos dois canais em utilização na posição inicial de *reset*
8. Estender os dois atuadores de forma síncrona durante 40cm
9. Estender o atuador linear do canal 1 e retrair o atuador linear do canal 2 durante 20cm simultaneamente
10. Colocar os atuadores lineares dos dois canais em utilização na posição inicial de *reset*
11. Ligar e configurar o mecanismo de *soft start and stop* no canal 1 com um tempo de aceleração de 3s e com uma velocidade de operação de 80%
12. Estender o atuador linear do canal 1 durante 40cm
13. Colocar o autador linear do canal 1 na posição inicial de *reset*
14. Desligar o mecanismo de *soft start and stop* do canal 1
15. Criar e guardar um sequência que permite, em primeiro, estender os atuadores lineares do canal 1 e canal 2 durante 30cm simultaneamente, em segundo estender o atuador linear do canal 1 durante 20cm e, em último retrair o atuador linear do canal 2 durante 10cm
16. Repetir a sequência anterior num total de 3 vezes

17. Colocar os atuadores lineares dos dois canais em utilização na posição inicial de *reset*
18. Estender o atuador linear do canal 1 durante 50cm
19. Desligar o acesso à plataforma da unidade central
20. Aceder novamente à plataforma e confirmar o armazenamento da informação relativo ao último movimento em cada canal

8.2.2.2 Segunda e Terceira Fase

1. Desligar o mecanismo *soft start and stop* nos dois canais em utilização
2. Configuração e parametrização inicial do atuador linear do canal 1 (*Hi*)
3. Configuração e parametrização inicial do atuador linear do canal 2 (*Lo*)
4. Configuração e parametrização inicial do atuador linear do canal 3 (*Backrest*)
5. Colocar o atuadores lineares dos três canais em utilização na posição inicial de *reset*
6. Estender os dois atuadores (*Hi-Lo*) de forma síncrona durante 40cm
7. Estender o atuador linear do canal 3 durante 40cm
8. Colocar os três atuadores lineares na posição inicial de *reset*
9. Criar e guardar um sequência que permite, em primeiro, estender os atuadores lineares *Hi-Lo* durante 30cm e, em segundo, estender o atuador *Backrest* durante 20cm

8.3 Conclusões e Trabalho Futuro

Os objetivos e requisitos principais apresentados na secção 3.2 foram, de uma forma global, cumpridos com sucesso. Salientamos apenas que relativamente ao requisito 15 que, embora, a unidade de controlo consiga monitorizar a corrente consumida pelo atuador não foi implementada a transmissão desta informação à unidade central. Dado o volume de dados que esta monitorização acrescenta e o protocolo de comunicação *SPI* implementada entre as duas entidades da solução optou-se que a avaliação da corrente consumida por cada atuador fosse feita de forma individual em cada unidade de controlo através da porta série associada ao microcontrolador. Assim, conseguimos salvaguardar, em caso de necessidade, a monitorização da corrente em tempo real por parte da equipa *R&D*.

Sucintamente, o sistema implementado permite controlar a posição e a velocidade dos atuadores lineares que integram os diversos dispositivos médicos em desenvolvimento, facilita a definição de sequências de movimentos e a realização de testes de repetibilidade e carateriza-se pela sua capacidade adaptativa de ajustar o controlo aos diferentes modelos de atuadores lineares (LA40, LA27, TA23 E TA31).

De uma avaliação crítica feita ao funcionamento do sistema implementado destacamos aspetos estruturais e funcionais que podem ser melhorados de forma a proporcionar à equipa de *R&D* um experiência mais intuitiva. Ao nível da unidade de controlo é interessante a integração de todos os seus componentes numa única placa *PCB* com conetores de forma a facilitar a conexão/desconexão à unidade central e ao atuador linear permitindo um transporte mais simplificado e uma maior flexibilidade na integração nos diferentes protótipos em desenvolvimento.

Relativamente à unidade central é necessário desenvolver um método que permite transmitir a corrente consumida pelos atuadores durante cada movimento à unidade central tornando a interação da equipa *R&D* com o sistema global implementado mais dinâmica e intuitiva.

Anexo A

Estudo do Mercado

A.1 *Texas Instruments*

Modelo	*DRV8704DCPR*
Configuração	2 x *H-Bridge Driver*
Descrição	Tensão de alimentação do motor até 60V, V_M Compatível com lógica digital 5V Frequência *PWM* até 500 kHz Monitorização da corrente *SPI* *PWM Interface* Sinalização de existência de falhas (*SPI* e *pin nFault*) Proteção contra sobrecorrente, sobreaquecimento e curto-circuito *HTSSOP-38 package*

+

Modelo	4 X *CSD18537NKCS*
Configuração	*N-channel MOSFET*
Descrição	V_{DS} até 60V V_{GS} até 20V I_D até 39A (Contínua, 25^{o}C) *TO-220 Package*

+

Modelo	*MSP-EXP430G2ET*
Configuração	*Microcontroller unit*
Descrição	Conversor *ADC* de 10 bits *Timers de 16 bits* F_{CPU} até 1MHz *SPI Interface*

Tabela A.1: Texas Instruments - Arquitetura da Solução 3

A.2 *Infineon Technologies*

Modelo	*BTM7710G*	2 x *BTN8962TA*
Configuração	*H-Bridge Driver* + *H-Bridge*	*Half-Bridge Driver* + *Half-Bridge*
Descrição	*Peak current 15A*, 25°C Tensão de alimentação até 42V *PG-TO263-15-1 Package* R_{ds} : $70m\Omega + 40m\Omega$, 25°C	*Peak current 30A*, 25°C Tensão de alimentação até 42V *PG-TO263-7-1 Package* R_{ds} : $13.4m\Omega + 16.9m\Omega$, 25°C

+

Modelo	*XC886*	*XMC1400*
Configuração	*MCU*	*MCU*
Descrição	*SPI* Conversor *ADC* de 10 bits *Timers de 16 bits* F_{CPU} até 96MHz	*CAN* e *SPI* Conversor *ADC* de 10 bits *Timers de 16 bits* F_{CPU} até 48MHz

Tabela A.2: *Infineon Technologies* - Arquitetura da Solução 1

Modelo	*TLE9853QX*
Configuração	*MCU* + *H-Bridge Driver*
Descrição	F_{CPU} até 40MHz Conversor *ADC* de 10 bits Frequência *PWM* até 25kHz *SPI Interface*

+

Modelo	2 x *IPG20N04S4-09A*
Configuração	*Half-Bridge*
Descrição	Corrente até 20A(Contínuo, 25°C) Tensão de Alimentação até 42V R_{ds} : $12.4m\Omega + 12.4m\Omega$, 25°C *PG-TDSON-8-10 Package*

Tabela A.3: *Infineon Technologies* - Arquitetura da Solução 2

A.3 *STMicroeletronics*

Modelo	*VNH7100AS*	*VNH7070AS*	*VNH5019A-E*
Configuração	*H-Bridge Driver* + *H-Bridge*	*H-Bridge Driver* + *H-Bridge*	*H-Bridge Driver* + *H-Bridge*
Descrição	*Peak current 12A*, 25ºC Tensão de alimentação até 41V *SO-16N Package* R_{ds} : 100$m\Omega$, 25ºC	*Peak current 15A*, 25ºC Tensão de alimentação até 41V *SO-16N Package* R_{ds} : 70$m\Omega$, 25ºC	*Peak current 30A*, 25ºC Tensão de alimentação até 41V *MultiPowerSO-30* R_{ds} : 18$M\Omega$, 25ºC

+

Modelo	*STM8S208MB*	*STM32F103*	*ATMEGA328P*
Configuração	*MCU*	*MCU*	*MCU*
Descrição	*CAN* e *SPI* Conversor *ADC* de 10 bits *Timers* de 16 *bits* e 8 *bits* F_{CPU} até 24MHz	*CAN* e *SPI* Conversor *ADC* de 10 bits *Timers* de 16 *bits* e 8 *bits* F_{CPU} até 72MHz	*SPI* Conversor *ADC* de 10 bits *Timers* de 16 *bits* e 8 *bits* F_{CPU} até 20MHz

Tabela A.4: *STMicroeletrocnics* - Arquitetura da Solução 1

Anexo B

VNH5019A-E: H-Bridge Driver

B.1 Características principais

Símbolo	Parâmetro	Valor	Unidade
V_{CC}	Tensão de alimentação máxima	41	V
I_{Out}	Corrente máxima	30	A
$T_{J(max)}$	Temperatura máxima da junção	150	^{o}C
$R_{On(HS)}$	Resistência *high-side*, T_j = -40^{o}C até 150^{o}C	26.5	$m\Omega$
$R_{On(LS)}$	Resistência *low-side*, T_j = -40^{o}C até 150^{o}C	11.5	$m\Omega$
t_r	*LSD Rise time*	2	μs
t_f	*LSD Fall time*	2	μs
$t_{d(on)}$	*HSD Rise time*	250	μs
$t_{d(off)}$	*HSD Fall time*	250	μs
f_{pwm}	Frequência *PWM* máxima	20	kHz

Tabela B.1: Resumo das características do *IC VNH5019A-E*[11]

B.2 Dissipação de calor

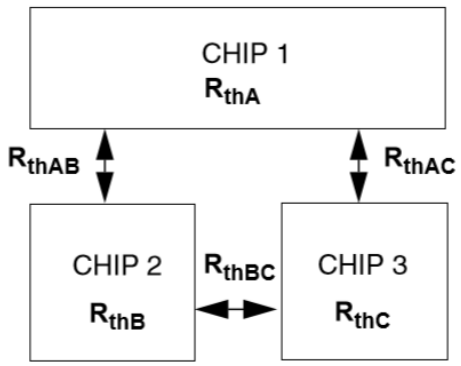


Figura B.1: Disposição dos *chips* no integrado[11]

Situação *I*					
Chip 1	**Chip 2**	**Chip 3**	**T_chip 1**	**T_chip 2**	**T_chip 3**
On	*Off*	*On*	$P_{chip1}R_{thA} + P_{chip3}R_{thAC} + T_{amb}$	$P_{chip1}R_{thAB} + P_{chip3}R_{thBC} + T_{amb}$	$P_{chip1}R_{thAC} + P_{chip3}R_{thC} + T_{amb}$
Situação *II*					
Chip 1	**Chip 2**	**Chip 3**	**T_chip 1**	**T_chip 2**	**T_chip 3**
On	*On*	*Off*	$P_{chip1}R_{thA} + P_{chip2}R_{thAB} + T_{amb}$	$P_{chip1}R_{thAB} + P_{chip2}R_{thB} + T_{amb}$	$P_{chip1}R_{thAC} + P_{chip2}R_{thBC} + T_{amb}$
Situação *III*					
On	*Off*	*Off*	$P_{chip1}R_{thA} + T_{amb}$	$P_{chip1}R_{thAB} + T_{amb}$	$P_{chip1}R_{thAC} + T_{amb}$
Situação *IV*					
Chip 1	**Chip 2**	**Chip 3**	**T_chip 1**	**T_chip 2**	**T_chip 3**
On	*On*	*On*	$P_{chip1}R_{thA} + (P_{chip2} + P_{chip3})R_{thAB} + T_{amb}$	$P_{chip2}R_{thB} + P_{chip1}R_{thAB} + P_{chip3}R_{thBC} + T_{amb}$	$P_{chip1}R_{thAB} + P_{chip2}R_{thBC} + P_{chip3}R_{thC} + T_{amb}$

Tabela B.2: Dissipação de calor em regime estacionário [11]

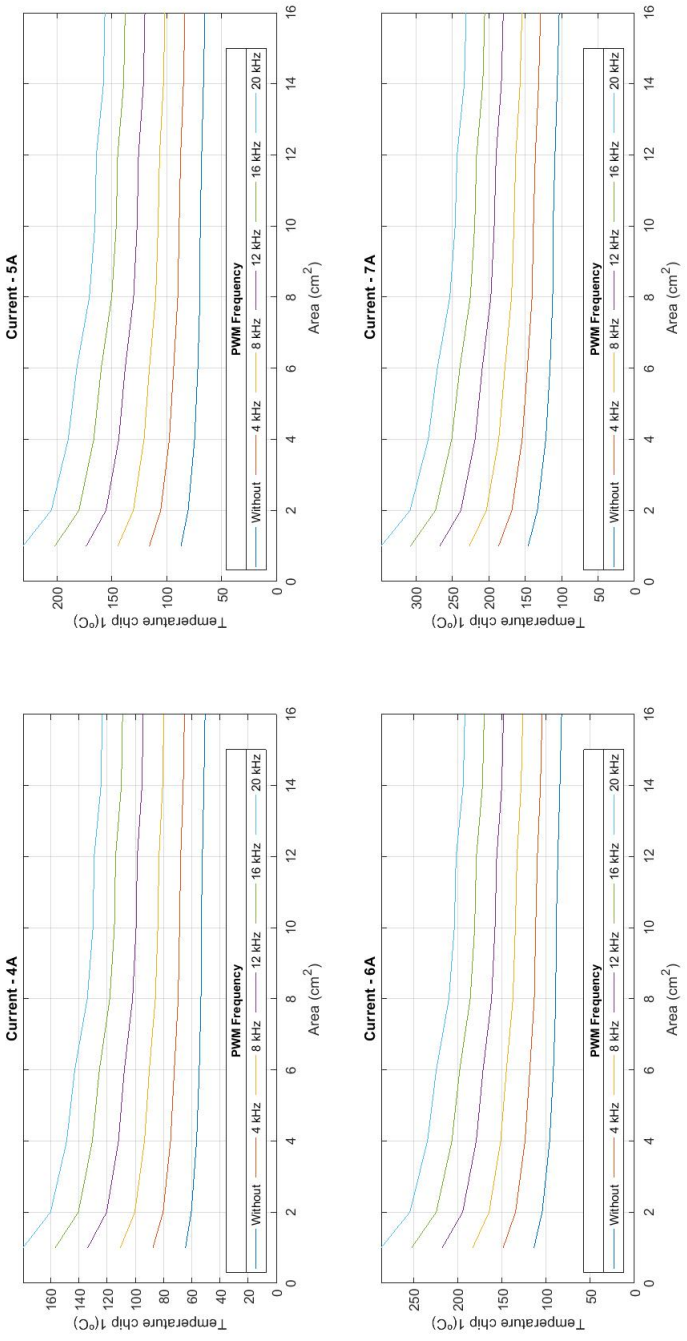


Figura B.2: Temperatura *chip* 1 - Situação *I*

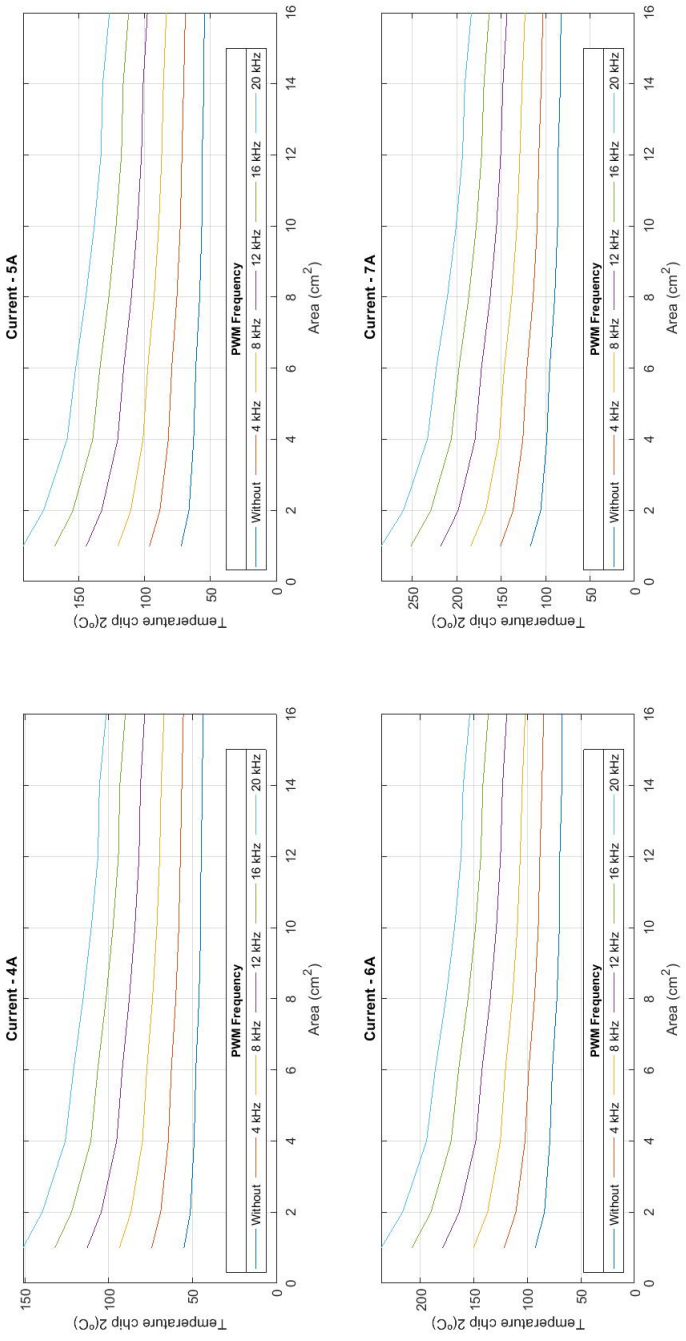


Figura B.3: Temperatura *chip* 2 - Situação *I*

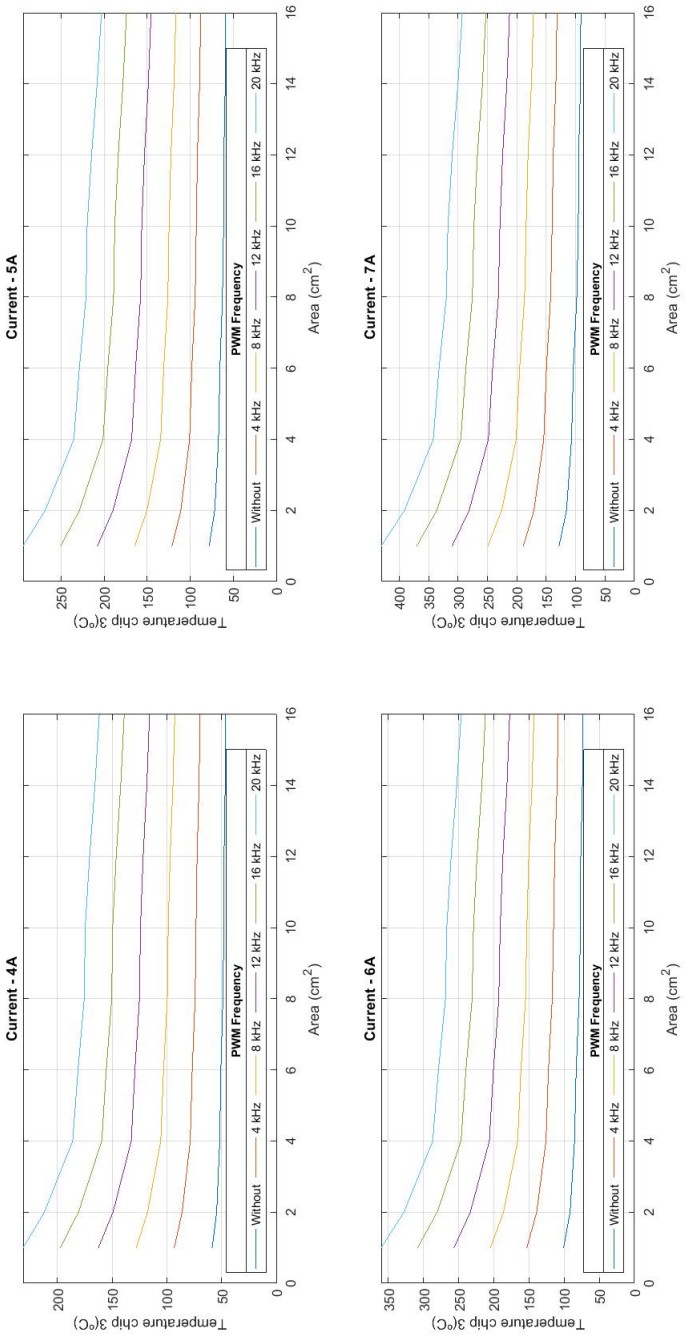


Figura B.4: Temperatura *chip* 3 - Situação *I*

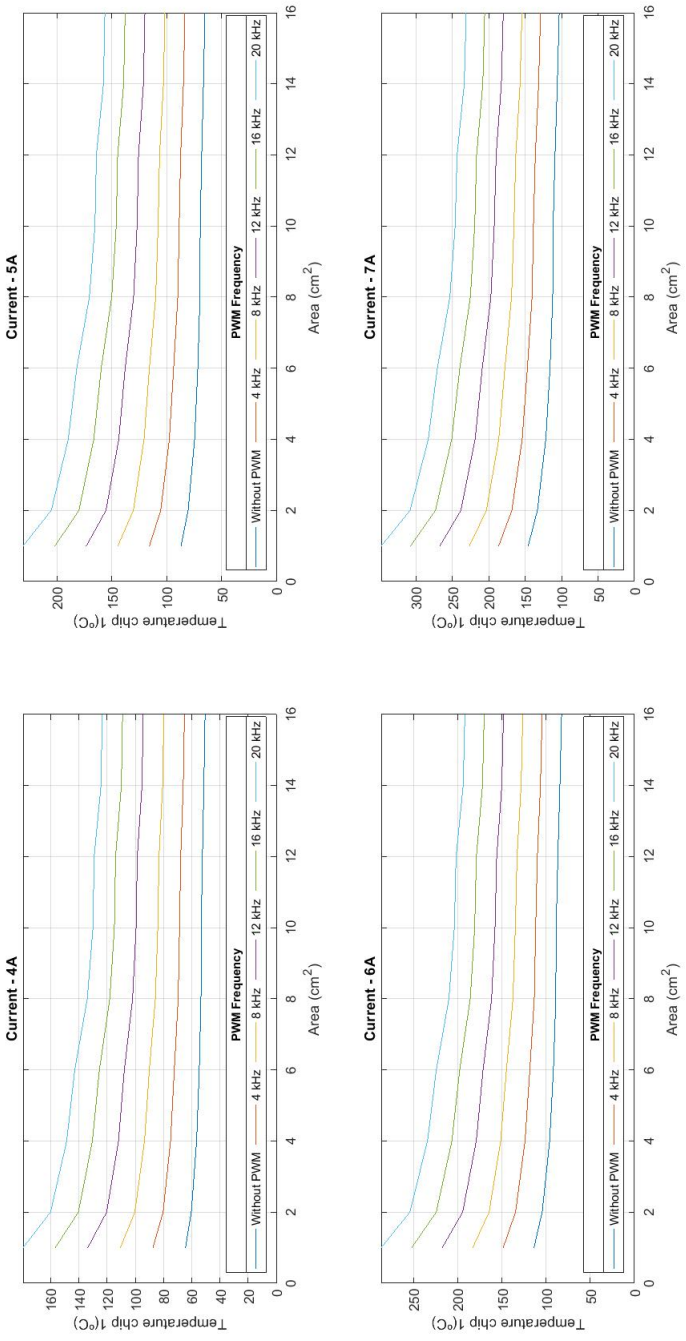


Figura B.5: Temperatura *chip* 1 - Situação *II*

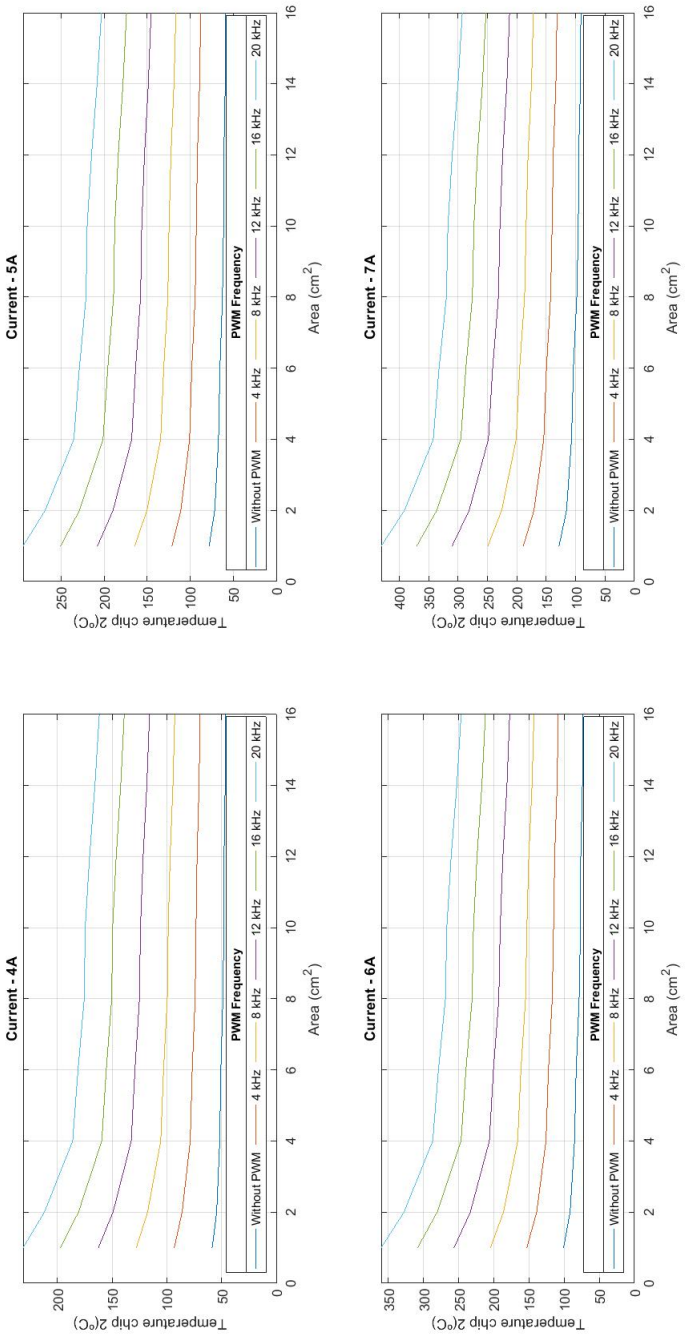


Figura B.6: Temperatura *chip* 2 - Situação *II*

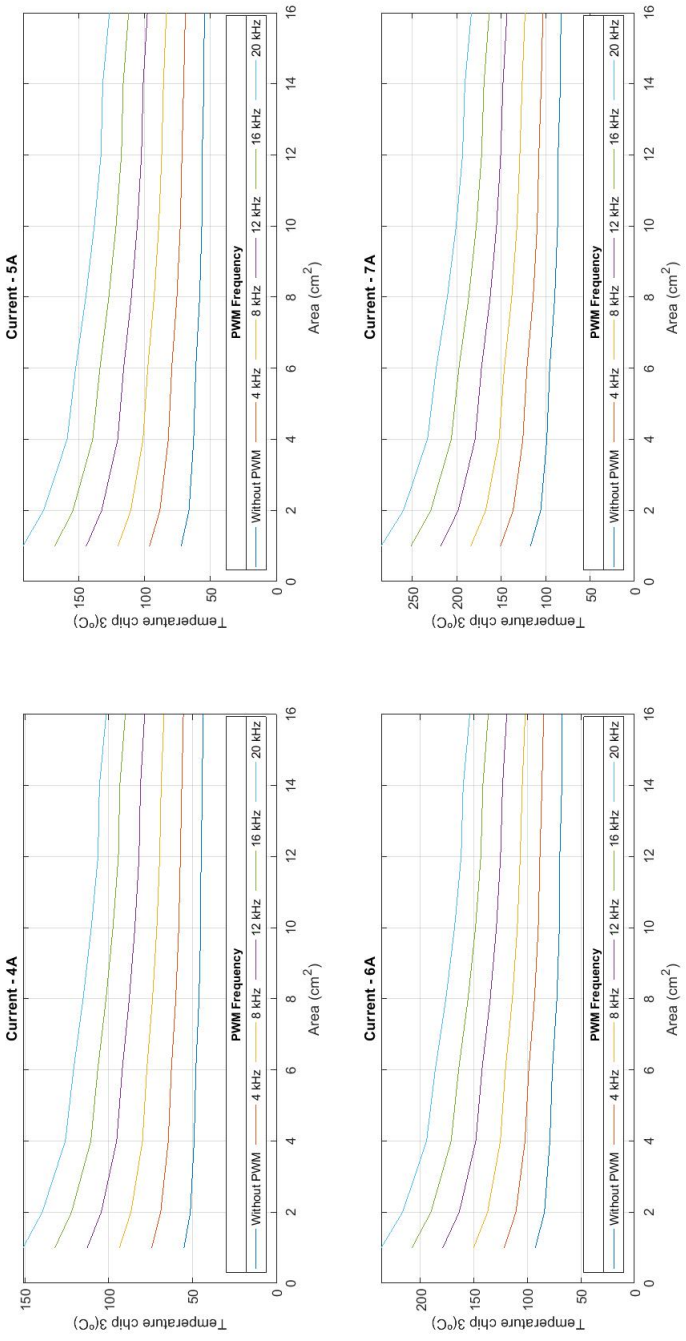


Figura B.7: Temperatura *chip* 3 - Situação *II*

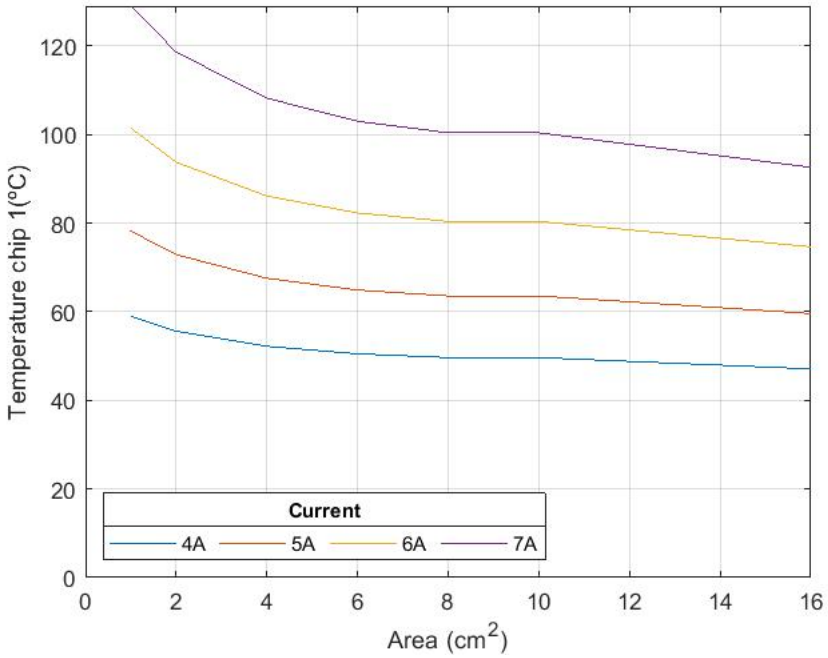


Figura B.8: Temperatura *chip* 1 - Situação *III*

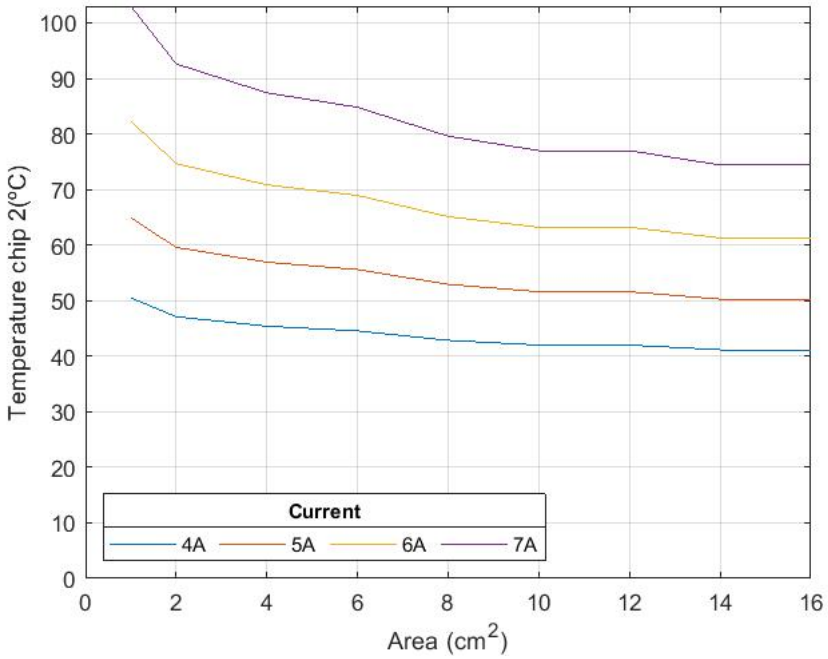


Figura B.9: Temperatura *chip* 2 - Situação *III*

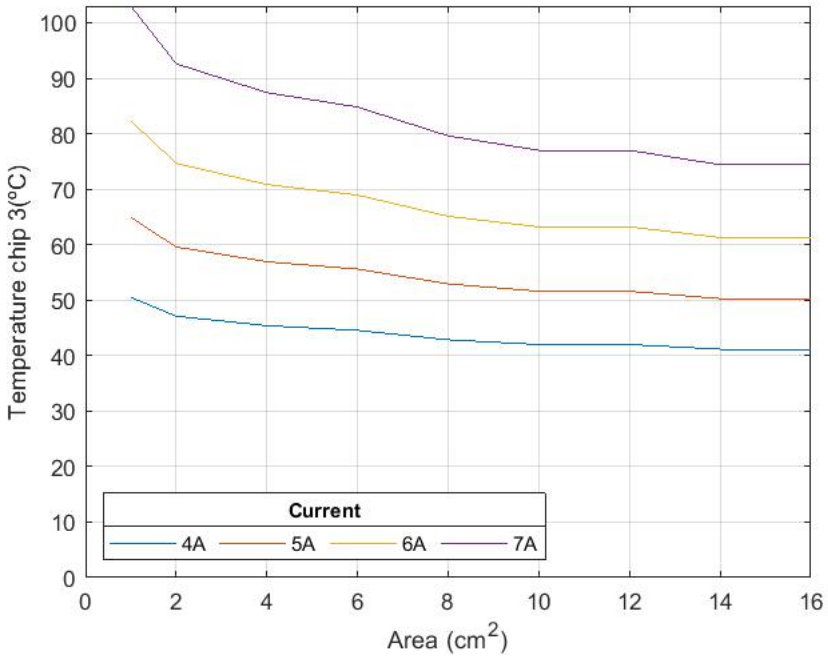


Figura B.10: Temperatura *chip* 3 - Situação *III*

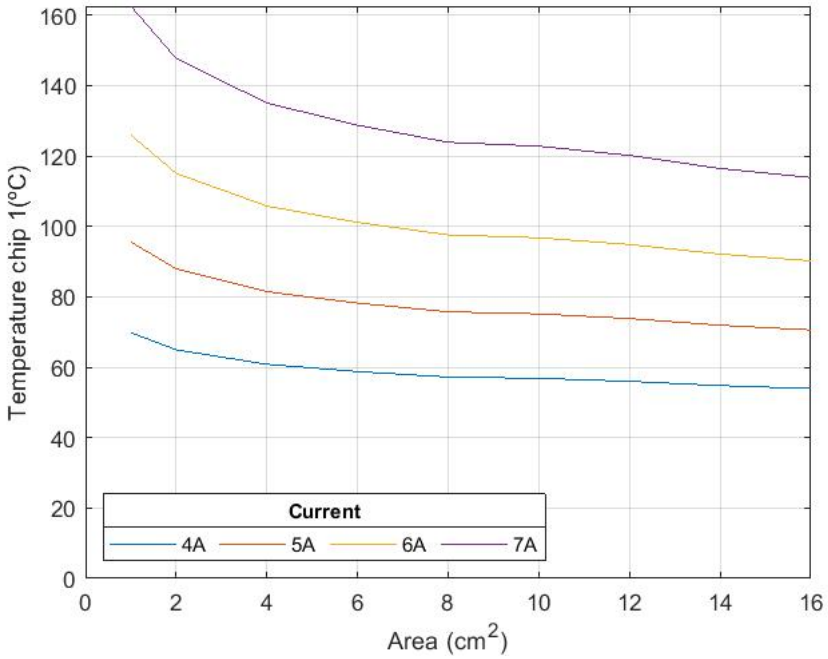


Figura B.11: Temperatura *chip* 1 - Situação *IV*

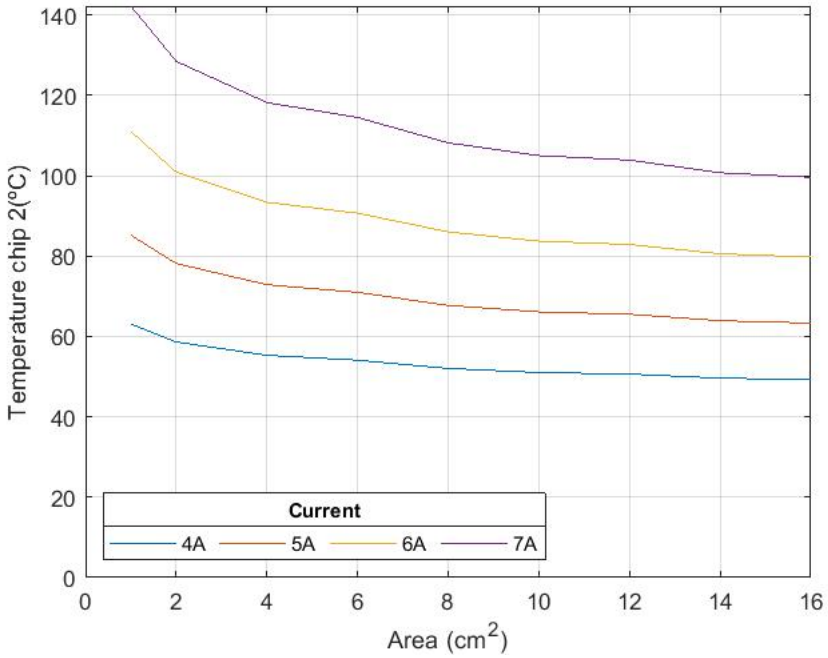


Figura B.12: Temperatura *chip* 2 - Situação *IV*

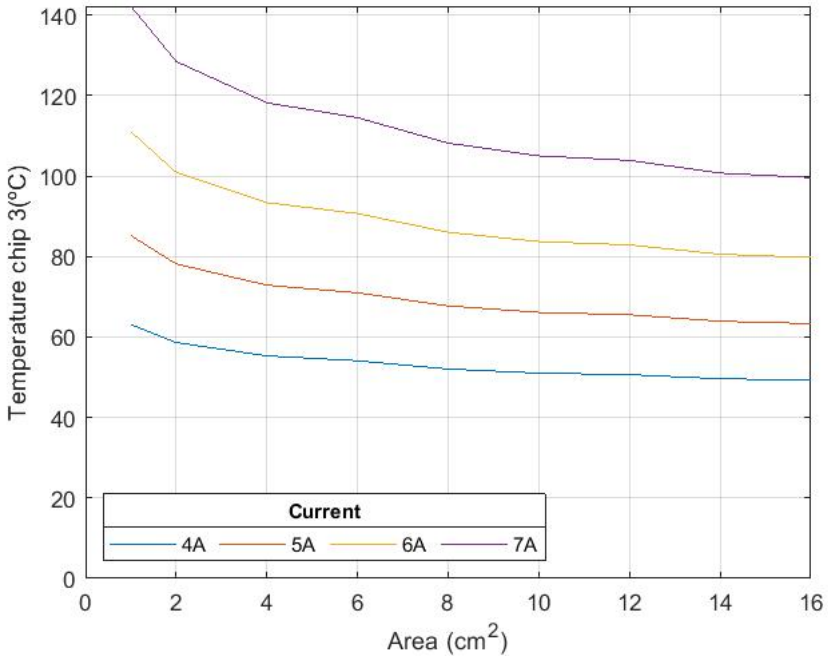


Figura B.13: Temperatura *chip* 3 - Situação *IV*

Anexo C

Comunicação *SPI*

C.1 Códigos e *masks*

C.1.1 Modos

Modo	Código
Setup()	0xF8
Update()	0xF9
Advanced()	0xFA
Basic()	0xFB
Communication()	0xFC

Tabela C.1: Modos *SPI*

C.1.2 *Masks*

Tipo	Mask	Valor
Comprimento	$size_buffer_communication_mode$	4
Comprimento	$size_buffer_setup_mode$	5
Comprimento	$size_buffer_update_mode$	4
Comprimento	$size_buffer_basic_mode$	4
Comprimento	$size_buffer_advanced_mode$	5
Posição	$mode$	1
Posição	$command$	2
Posição	$data$	3

Tabela C.2: *Masks SPI*

C.1.3 Comandos

Comando	Código	Modo
Enable()	0x01	*Communication()*
Disable()	0x02	*Communication()*
InitialSetup()	0x03	*Basic()*
SafeReset()	0x04	*Basic()*
BasicExtended()	0x05	*Basic()*
BasicRefracted()	0x06	*Basic()*
Stop()	0x07	*Basic()*
AdvancedExtended()	0x08	*Advanced()*
AdvancedRefracted()	0x09	*Advanced()*
GetPositionLow()	0x0A	*Update()*
GetPositionHigh()	0x0B	*Update()*
GetPulseRate()	0x0C	*Update()*
GetFlagAdvanced()	0x14	*Update()*
SendPositionLow()	0x0D	*Setup()*
SendPositionHigh()	0x0E	*Setup()*
SendPulseRate()	0x0F	*Setup()*
SendStrokeLenghtLow()	0x10	*Setup()*
SendStrokeLenghtHigh()	0x11	*Setup()*
SendSoftStart()	0x12	*Setup()*
SendAccelarationTimeLow()	0x13	*Setup()*
SendAccelarationTimeHigh()	0x15	*Setup()*
SendMaxSpeed()	0x16	*Setup()*
ReadWithSucess()	0xFD	–
End()	0xFE	–
Start()	0xFF	–

Tabela C.3: Comandos *SPI*

C.2 Estrtura geral dos vetores

Modo *Setup()*

Start()	*Setup()*	*command*	*data*	*End()*

Update()

Start()	*Update()*	*command*	*End()*

Modo *Basic()*

Start()	*Basic()*	*command*	*End()*

Modo *Advanced()*

Start()	*Advanced()*	*command*	*shift*	*End()*

Modo *Communication()*

Start()	*Communication()*	*command*	*End()*

Figura C.1: Estrutura dos vetores *TxCommands*

Setup()/Advanced()

$--$	*ReadWithSucess*()	*ReadWithSucess*()	*ReadWithSucess*()	*ReadWithSucess*()

Update()

$--$	*ReadWithSucess*()	*ReadWithSucess*()	*data*

Basic()/Communication()

$--$	*ReadWithSucess*()	*ReadWithSucess*()	*ReadWithSucess*()

Figura C.2: Estrutura dos vetores *RxCommands*

C.3 Algoritmos

C.3.1 Unidade de Controlo

Algorithm C.1 Mecanismo de interpretação

```
1:  spiCode ← spi_shift_data_register
2:  if spiCode == Start() then
3:      posBuffer ← 0
4:      flag_buffer_spi ← 0
5:      spi_shift_data_register ← ReadWithSucess()
6:  else if spiCode == End() then
7:      flag_buffer_spi ← 1
8:      spi_shift_data_register ← ReadWithSucess()
9:  else if posBuffer == command_mask then
10:     if spiCode == GetPositionLow() then
11:         spi_shift_data_register ← position & 0xFF
12:     else if spiCode == GetPositionHigh() then
13:         spi_shift_data_register ← (position >> 8) & 0xFF
14:     else if spiCode == GetPulseRate() then
15:         spi_shift_data_register ← pulse_rate
16:     else if spiCode == GetFinishedAdvanced() then
17:         spi_shift_data_register ← flag_finished_advanced
18:     else
19:         spi_shift_data_register ← ReadWithSucess()
20:     end if
21: else
22:     spi_shift_data_register ← ReadWithSucess()
23: end if
24: spiBuffer[posBuffer] ← spiCode
25: posBuffer ← posBuffer + 1
```

C.3.2 Unidade Central

Algorithm C.2 Mecanismo de construção - Vetor *Communication()*

```
1:  function CMDCOMMUNICATION(slave_channel, cmd)
2:      select_ss(slave_channel)
3:      Initialise buffer[size_buffer_communication_mode] ← {}
4:      buffer[0] ← Start()
5:      buffer[mode_mask] ← Communication()
6:      buffer[command_maks] ← cmd
7:      buffer[size_buffer_communication_mode − 1] ← End()
8:      spi_shift(buffer)
9:      disable_ss(slave_channel)
10:     iterator ← mode
11:     for iterator < size_buffer_communication_mode do
12:         if buffer[iterator] ≠ ReadWithSucess() then
13:             return false
14:         end if
15:         iterator ← iterator + 1
16:     end for
17:     return true
18: end function
```

Algorithm C.3 Mecanismo de construção - Vetor *Update()*

```
1:  function CMDUPDATE(slave_channel, cmd, ∗data)
2:      select_ss(slave_channel)
3:      Initialise buffer[size_buffer_update_mode] ← {}
4:      buffer[0] ← Start()
5:      buffer[mode_mask] ← Update()
6:      buffer[command_mask] ← cmd
7:      buffer[size_buffer_update_mode − 1] ← End()
8:      spi_shift(buffer)
9:      disable_ss(slave_channel)
10:     iterator ← mode
11:     for iterator < size_buffer_update_mode − 1 do
12:         if buffer[iterator] ≠ ReadWithSucess() then
13:             return false
14:         end if
15:         iterator ← iterator + 1
16:     end for
17:     ∗data ← buffer[size_buffer_update_mode − 1]
18:     return true
19: end function
```

Algorithm C.4 Mecanismo de construção - Vetor *Basic()*

```
1:  function CMDBASIC(slave_channel, cmd)
2:      select_ss(slave_channel)
3:      Initialise buffer[size_buffer_basic_mode] ← {}
4:      buffer[0] ← Start()
5:      buffer[mode_mask] ← Basic()
6:      buffer[command_mask] ← cmd
7:      buffer[size_buffer_basic_mode − 1] ← End()
8:      spi_shift(buffer)
9:      disable_ss(slave_channel)
10:     iterator ← mode
11:     for iterator < size_buffer_basic_mode do
12:         if buffer[iterator] ≠ ReadWithSucess() then
13:             return false
14:         end if
15:         iterator ← iterator + 1
16:     end for
17:     return true
18: end function
```

Algorithm C.5 Mecanismo de construção - Vetor *Advanced()*

```
1:  function CMDADVANCED(slave_channel, cmd, shift)
2:      select_ss(slave_channel)
3:      Initialise buffer[size_buffer_advanced_mode] ← {}
4:      buffer[0] ← Start()
5:      buffer[mode_mask] ← Advanced()
6:      buffer[command_mask] ← cmd
7:      buffer[shift_mask] ← shift
8:      buffer[size_buffer_advanced_mode − 1] ← End()
9:      spi_shift(buffer)
10:     disable_ss(slave_channel)
11:     iterator ← mode
12:     for iterator < size_buffer_advanced_mode do
13:         if buffer[iterator] ≠ ReadWithSucess() then
14:             return false
15:         end if
16:         iterator ← iterator + 1
17:     end for
18:     return true
19: end function
```

Algorithm C.6 Mecanismo de construção - Vetor *Setup()*

```
1:  function CMDADVANCED(slave_channel, cmd, data)
2:      select_ss(slave_channel)
3:      Initialise buffer[size_buffer_setup_mode] ← {}
4:      buffer[0] ← Start()
5:      buffer[mode_mask] ← Setup()
6:      buffer[command_mask] ← cmd
7:      buffer[data_mask] ← data
8:      buffer[size_buffer_setup_mode − 1] ← End()
9:      spi_shift(buffer)
10:     disable_ss(slave_channel)
11:     iterator ← mode
12:     for iterator < size_buffer_setup_mode do
13:         if buffer[iterator] ≠ ReadWithSucess() then
14:             return false
15:         end if
16:         iterator ← iterator + 1
17:     end for
18:     return true
19: end function
```

Anexo D

Esquemáticos

D.1 Unidade de Controlo

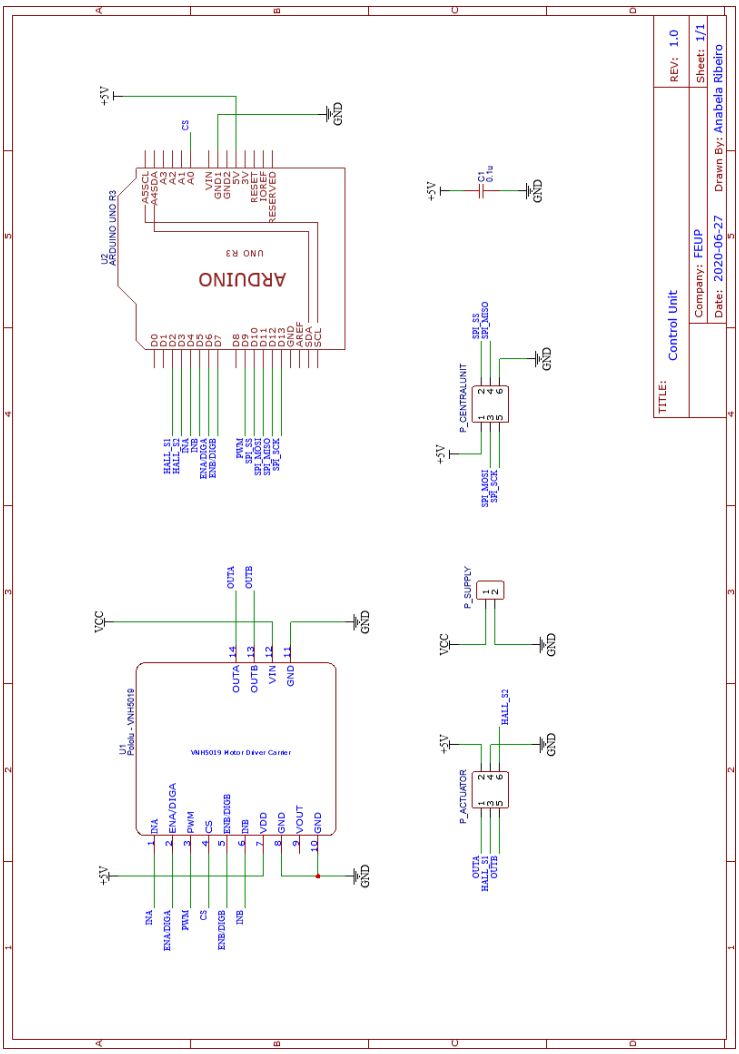


Figura D.1: Esquemático Unidade de Controlo

D.2 Unidade Central

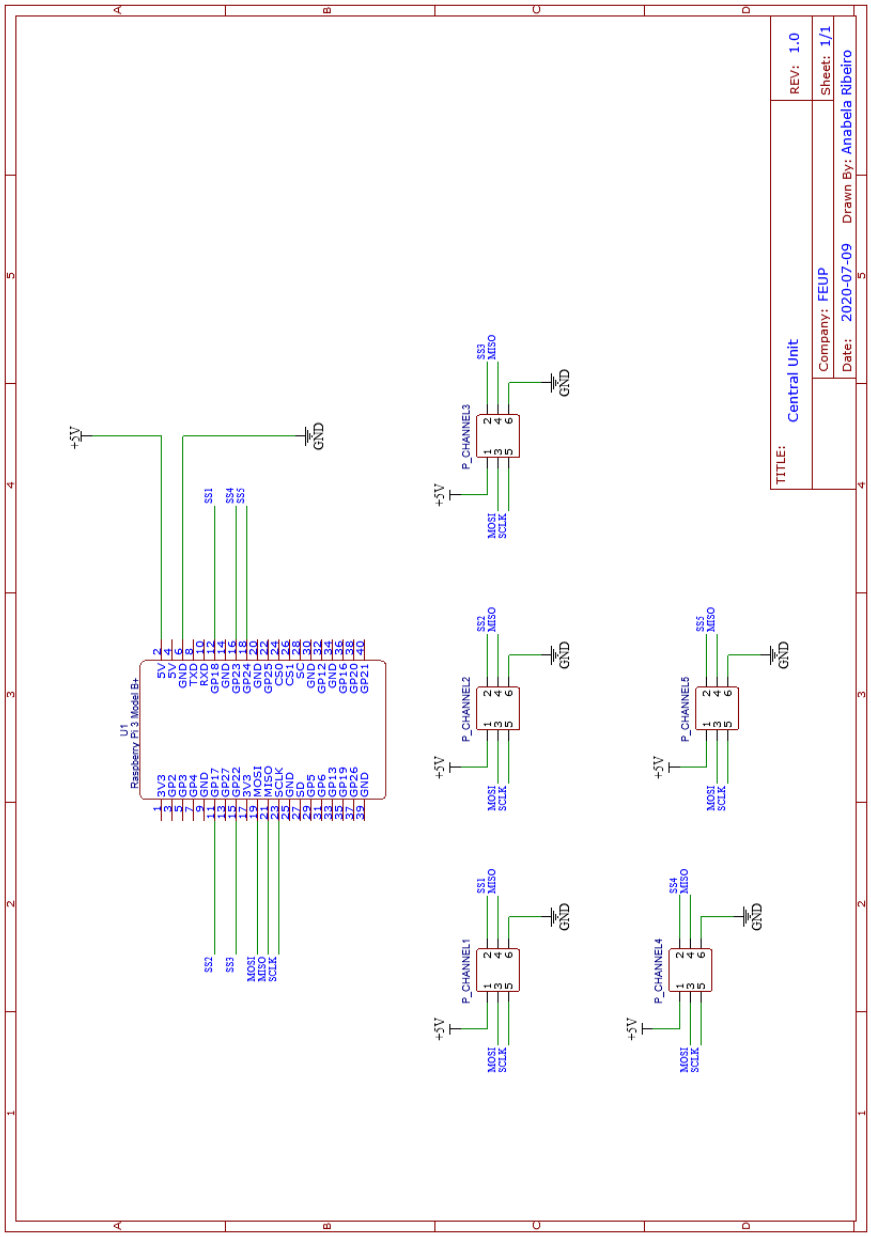


Figura D.2: Esquemático Unidade Central

Anexo E

Estrutura dos Ficheiros

E.1 Estrutura dos Ficheiros *JSON*

RecordsOfChannels.js		
Nome	**Valor**	**Comentário**
"Channel Number"	*int*	Número da unidade de controlo: valor numérico inteiro compreendido entre 1 e 5.
"Stroke Lenght"	*int*	Comprimento da extensão máxima do atuador: valor numérico inteiro compreendido entre 1 e 65535.
"Accelaration Rate"	*int*	Tempo de aceleração, em *ms*, quando o mecanismo *soft start and stop* está ligado: valor numérico inteiro compreendido entre 0 e 65535.
"Pulse Rate"	*int*	Número de rotações rotóricas por cada mílimetro de deslocamento linear do atuador: valor numérico inteiro compreendido entre 0 e 255.
"Pin"	*int*	Número do pin *slave select* que habilita a comunicação *SPI*: valor numérico inteiro compreendido entre 0 e 255.
"Position"	*int*	Última posição conhecida do atuador tendo como referência o atuador totalmente retraído *SPI*: valor numérico inteiro compreendido entre 0 e 65536.
"Speed"	*int*	Relação entre a velocidade pretendida e a velocidade nominal esperada: valor percentual compreendido entre 0 e 100.
"Maximum Current"	*double*	Máxima corrente consumida pelo atuador: valor decimal compreendido entre 0,00 e 7,00.
"Communication"	*string*	Estado atual da comunicação *SPI*: "Enable"ou "Disable".
"Modelo"	*string*	Modelo do atuador: "TA31", "TA23", "LA27", E "LA40".

Tabela E.1: Estrutura ficheiros *JSON* - *RecordsOfChannels.js*

RecordsOfMoves_x.js		
Nome	**Valor**	**Comentário**
"Name"	*string*	Nome definido para a sequência a que o movimento pertence.
"Order"	*int*	Representa a ordem do movimento na sequência.
"Type"	*string*	Tipo do movimento: "Independend", "Synchronous", "Reverse Synchronous".
"Master"	*int*	Número da unidade de controlo que desempenha o papel *master* se o movimento for do tipo "Synchronous"ou "Reverse Synchronous".
"Slave"	*int*	Número da unidade de controlo que desempenha o papel *slave* se o movimento for do tipo "Synchronous"ou "Reverse Synchronous".
"Shift"	*int*	Valor do deslocamento do atuador: valor numérico inteiro compreendido entre 1 e 65535.

Tabela E.2: Estrutura ficheiros *JSON* - *RecordsMoves_x.js*

Anexo F

Repositórios

F.1 Unidade de Controlo

https://github.com/up201505284/ControlUnit

F.2 Unidade Central

https://github.com/up201505284/CentralUnit

Referências

[1] Ajith Asok e Mathew Dominic. A Novel Method of Using Direct Torque Control in Bipolar Stepper Motor. *International Journal of Scientific & Engineering Research*, páginas 1171–1178, 2014.

[2] Microchip Technology Inc. Stepping Motors Fundamentals - AN907, 2004.

[3] Institute for Power Electronics and Electrical Drive of RWTH Aachen University. Motor Handbook, 2013.

[4] NXP Semiconductors. AN10513 Brushed DC motor control using the LPC2101, 2007.

[5] ON Semiconductor. DC Motor Driver Fundamentals, 2014.

[6] Condit, Reston e Microchip Technology Inc. Brushed DC Motor Fundamentals, 2004.

[7] Jian Zhao e Yu Yangwei. Brushless DC Motor Fundamentals, 2011.

[8] TiMOTION. Part 2: Components of an electric linear actuator, 2017. Acedido: 2020-06-01. URL: https://www.timotion.com/en/news/news_content/news-and-articles/general/part-2%3a-components-of-an-electric-linear-actuator?upcls=1481189409&guid=1496378846.

[9] Atmel. ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET, 2015.

[10] Anne Barela. Cross compiling - compile c programs on your windows machine for a raspberry pi - pi setup, 2016. Acedido: 2020-06-01. URL: http://21stdigitalhome.blogspot.com/2016/03/cross-compiling-compile-c-programs-on_46.html.

[11] STMicroelectronics. VNH5019A-E Automotive fully integrated H-bridge motor driver, 2017.

[12] Portescap. Miniature Motors for Medical Applications, 2017.

[13] S. Derammelaere, M. Haemers, J. De Viaene, F. Verbelen, e K. Stockman. A quantitative comparison between BLDC, PMSM, brushed DC and stepping motor technologies. Em *19th International Conference on Electrical Machines and Systems, ICEMS 2016*, 2017.

[14] TiMOTION. TA31 series.

[15] TiMOTION. TA23 series.

[16] Linak. Actuator LA27 Data sheet.

[17] Linak. Actuator LA27 HOMELINE Hall sensor Data sheet.

[18] Linak. Actuator LA40 Data sheet.

[19] Marius Andrei Nicolaescu. Rotary electrical motors used in medical industry relevant design considerations and actual advances. *EPE 2012 - Proceedings of the 2012 International Conference and Exposition on Electrical and Power Engineering*, páginas 664–667, 2012.

[20] Dan Brüllmann, Tobias Muench, e Ralf Schulze. Construction of a low-cost surface scanner for medical studies: a feasibility study. *Oral Radiology*, páginas 211–216, 2016.

[21] M. A. Khan, S. Tehami, e O. Mazhar. Designing of microcontroller based Syringe Pump with variable and low delivery rates for the administration of small volumes. *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging*, páginas 135–138, 2015.

[22] Beckhoff. Application Note DK9222-0410-0014. *Beckhoff Automation GmbH*, páginas 1–10, 2010.

[23] Silicon Laboratories. Stepper Motor Reference Design - AN155, 2008.

[24] Texas Instruments. High Resolution Microstepping Driver With the DRV88xx Series, 2010.

[25] J. Schick, M. Glaser, e S. Huber. Design, Realization and Verification of a Novel Knee Joint Actuator for Robotic Exoskeletons. Em *Proceedings - 2019 IEEE International Conference on Mechatronics, ICM 2019*, páginas 175–181, 2019.

[26] Jong Sun Ko, Tae Ho Lee, Kyu Gyeom Kim, e In Dong Kim. Precision speed control of PMSM for stimulation of the vestibular system using rotatory chair. *Mechatronics*, 13(5):465–475, 2003.

[27] Seungho Kim, Wonsung Lee, Hoebeom Yang, Soon Bae Chung, Yeoung geol Bae, e Jae Hee Kim. Development of a modular intelligent care capsule robot for the elderly in response to an aging society with a light actuator. *International Conference on Control, Automation and Systems*, páginas 1113–1116, 2018.

[28] H.A. Youssef, M.F. Moussa, A. Helal, e Y. Gaber. Unity Power Factor control of permanent magnet motor drive system. *12th International Middle-East Power System Conference*, 2008.

[29] Shoudao Huang, Ziqiang Chen, Keyuan Huang, e Jian Gao. Maximum torque per ampere and flux-weakening control for PMSM based on curve fitting. *2010 IEEE Vehicle Power and Propulsion Conference, VPPC 2010*, 4(3):1–5, 2010.

[30] Jorge Zambada e Debraj Deb. Sensorless Field Oriented Control (FOC) of a Permanent Magnet Synchronous Motor (PMSM), 2010.

[31] Jorge Zambada. Sinusoidal control of PMSM motors with dsPIC30F DSC, 2005.

[32] Daniel Jones, Andrew Lewis, e Gregory S. Fischer. Development of a standalone surgical haptic arm. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, páginas 2136–2139, 2011.

[33] Giuseppe Tortora, Tommaso Ranzani, Iris De Falco, Paolo Dario, e Arianna Menciassi. A miniature robot for retraction tasks under vision assistance in Minimally Invasive Surgery. *Robotics*, 3:70–82, 2014.

[34] Arianna Menciassi, Cesare Stefanini, Giovanni Orlandi, Marco Quirini, e Paolo Dario. Towards active capsular endoscopy: Preliminary results on a legged platform. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, páginas 2215–2218, 2006.

[35] P. Yedamale e Microchip Technology Inc. Brushless DC (BLDC) Motor Fundamentals, 2003.

[36] Bilal Akin, Manish Bhardwaj, e Warriner Jon. Trapezoidal Control of BLDC Motors Using Hall Effect Sensors, 2010.

[37] Milan Breijl, Michal Princ, e Pavel Sustek. BLDC Motor with Speed Closed Loop and DC-Bus Break Controller , Driven by eTPU on MPC5554, 2006.

[38] Silicon Laboratories Inc. AN0816: EFM32 Brushless DC Motor Control, 2017.

[39] Ward Brown. Brushless DC Motor Control Made Easy, 2002.

[40] Microchip Technology Inc. Motor Control and Drive Design Solutions Motor Control and Drive Solutions, 2012.

[41] Gaurav Pandey e Ankit Vora. Open electronics for medical devices: State-of-art and unique advantages, 2019.

[42] Intel Corporation. FPGAs Resource Center. Acedido: 2020-02-01. URL: "https://www.intel.com/content/www/us/en/products/programmable/fpga/new-to-fpgas/resource-center/overview.html".

[43] Microchip Technology Inc. Motor Control and Drive Design Solutions Motor Control and Drive Solutions, 2019.

[44] Tsuyoshi Hanamoto, Masashi Deriha, Hidehiro Ikeda, e Teruo Tsuji. Digital Hardware Circuit Using FPGA for Speed Control System of Permanent Magnet Synchronous Motor, 2008.

[45] S. Balaji e M. Venkatesan. BLDC Motor Control Based on Duty Cycle. *2017 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2017*, páginas 1–4, 2018.

[46] Eric N. Sihite, Daniel J. Yang, e Thomas R. Bewley. Derivation of a new drive/coast motor driver model for real-time brushed DC motor control, and validation on a MIP robot. *IEEE International Conference on Automation Science and Engineering*, páginas 1099–1105, 2019.

[47] TiMOTION. Advantages and drawbacks of pneumatic, hydraulic, and electric linear actuators, jun 2017. Acedido: 2020-06-01. URL: https://www.timotion.com/en/news/news_content/news-and-articles/general/advantages-and-drawbacks-of-pneumatic,-hydraulic,-and-electric-linear-actuators?upcls=1481189409&guid=1499762723.

[48] Linak. Actuator academy^{TM} - industrial actuators, 2020. Acedido: 2020-06-01. URL: https://www.linak.com/segments/techline/actuator-academyindustrial-actuators/.

[49] TiMOTION. Part 4: Load and speed characteristics of an actuator, 2017. Acedido: 2020-06-01. URL: https://www.timotion.com/en/news/news_content/news-and-articles/general/part-4%3a-load-and-speed-characteristics-of-an-actuator?upcls=1481189409&guid=1496648215.