

TRABALHO DE GRADUAÇÃO

**VERIFICAÇÃO E REMODELAGEM DE UM SISTEMA  
DE PRÓTESE ROBÓTICA PARA AMPUTADOS  
DE MEMBRO INFERIOR**

Juliana Lemos Bento

Brasília, Julho de 2019



**ENGENHARIA  
MECATRÔNICA**  
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**VERIFICAÇÃO E REMODELAGEM DE UM SISTEMA  
DE PRÓTESE ROBÓTICA PARA AMPUTADOS  
DE MEMBRO INFERIOR**

**Juliana Lemos Bento**

*Relatório submetido como requisito parcial de obtenção  
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB

*Orientador*

\_\_\_\_\_

Prof. Roberto de Souza Baptista, FGA/Unb

*Banca examinadora*

\_\_\_\_\_

Profa. Cláudia Patricia Ochoa Diaz, FGA/Unb

*Banca examinadora*

\_\_\_\_\_

**Brasília, Julho de 2019**

## FICHA CATALOGRÁFICA

BENTO, JULIANA LEMOS

Verificação e remodelagem de um sistema de prótese robótica para amputados de membro inferior.

[Distrito Federal] 2019.

x, 101p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2019). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Marcha de amputado

2. Prótese transfemoral

3. Controle de marcha

I. Mecatrônica/FT/UnB

II. Título (Série)

## REFERÊNCIA BIBLIOGRÁFICA

BENTO, JULIANA LEMOS. (2019). Verificação e remodelagem de um sistema de prótese robótica para amputados de membro inferior.

Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°014, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 101p.

## CESSÃO DE DIREITOS

AUTOR: Juliana Lemos Bento

TÍTULO DO TRABALHO DE GRADUAÇÃO: Verificação e remodelagem de um sistema de prótese robótica para amputados de membro inferior.

GRAU: Engenheiro

ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

---

Juliana Lemos Bento

Endereço de e-mail: lemos.julianab@gmail.com

Campus Darcy Ribeiro, FT, Universidade de Brasília

Brasília – DF – Brasil.

## **Dedicatória**

*Dedico este trabalho a meu avô, Manoel Bento, que um dia decidiu mudar de vida para que seus filhos pudessem estudar.*

*Juliana Lemos Bento*

## Agradecimentos

*Gostaria de agradecer, primeiramente, à minha mãe, Cláudia, por sempre me apoiar em minhas escolhas, mesmo que esta em particular tenha sido motivo de imensa saudade diária. Em segundo lugar, ao meu irmão, Vinícius e ao meu pai, Joel, por me trazerem ensinamentos a uma maneira calada, por meio de gestos e do exercício da paciência. Aos meus amigos Alexandre, Karla, Túlio, Pablo e Artur, que se constituíram em família da qual eu me orgulho muito e por toda a vida extra-acadêmica que me proporcionaram. Brasília teria sido uma experiência muito diferente sem vocês. A Gabriel Monteiro, por me inspirar de diversas formas em uma graduação que pede muito de todos nós, mas que sempre resultou em alegria compartilhada em meio às dificuldades e em almoços de comida chinesa. Gostaria também de agradecer e celebrar a chance que me foi dada de realização de um intercâmbio, onde tive o prazer de conhecer colegas de engenharia que se converteram em melhores amigos e conselheiros. Somado a isso, aos professores das mais diversas nacionalidades que tive a chance de conhecer quando no exterior e que me inspiram como seres humanos no exercício de suas profissões como engenheiros e educadores, estas que tomo como exemplo e que desejo seguir. Ainda, dentro de um curso extremamente dominado por uma visão e ambientes masculinos tal qual é a engenharia, dos quais a Engenharia Mecatrônica não se exclui, gostaria de agradecer às colegas de curso e amigas Mariana e Débora. Vocês compartilharam de experiências e pressão impostas por gênero, assim como eu, e me deram apoio, amizade e inspiração durante diversas fases da graduação. Por fim, gostaria de agradecer ao meu orientador Geovany Borges pela imensa paciência e atenção dados a mim, e a Caio Eduardo, por ser companhia de jantares de RU e noites de trabalho.*

*Juliana Lemos Bento*

---

## RESUMO

O presente trabalho visa a recuperação de funcionalidades e construção de uma documentação técnica do projeto de prótese robótica automatizada RLEG, concebida no ano de 2011 dentro do Laboratório de Automação e Robótica da Universidade de Brasília. Após verificação do funcionamento do sistema existente, mudanças no *design de hardware* foram implementadas, com adição de novos componentes eletrônicos e substituição do microcontrolador utilizado. O funcionamento do circuito elétrico embarcado à prótese se baseia na atuação de um pistão amortecedor de fluido magnetizável. O coeficiente de amortecimento do dispositivo é proporcional à corrente de entrada aplicada, portanto, sendo este o sinal de controle que deve ser definido por meio de leituras de variáveis de movimento do indivíduo que utiliza a prótese. O projeto implementa uma verificação básica da correção de atuação da nova placa controladora implementada. Rotinas mais apropriadas de controle e identificação de intenção de movimento devem ser construídas sobre o circuito obtido, em trabalhos futuros.

Palavras Chave: Design de circuito embarcado, Marcha de amputado, Prótese transfemoral, Controle de marcha

---

## ABSTRACT

The present work recovers the system functionalities of the RLEG project and builds a technical documentation of the system features. The RLEG project consists of an automated robotic prosthesis, conceived in 2015 by an interdisciplinary team of students in the Robotics and Automation Laboratory at the University of Brasilia. After a system operational check, changes in its hardware layout were performed. In that sense, a new embedded system was designed with the addition of new electronic components and the replacement of the previous microcontroller. The main idea of the system mode of operation is the actuation of a magnetorheological piston. The fluid damping coefficient can be changed by means of the applied current, and therefore, this consists on the control variable of interest. The level of damping is determined by sensor readings concerning the user's motion. This work implements a basic check on the proper behaviour of the new embedded circuit board. More sophisticated control and intention of motion identification routines should be built as a continuation of the current work.

Keywords: Embedded Circuit Design, Amputee Gait, Transfemoral Prosthesis, Gait Control System

# SUMÁRIO

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>1</b>  |
| 1.1      | CONTEXTUALIZAÇÃO   | 1         |
| 1.2      | DEFINIÇÃO DO PROBLEMA  | 2         |
| 1.3      | OBJETIVOS DO PROJETO   | 2         |
| <b>2</b> | <b>Fundamentos e Revisão</b>   | <b>3</b>  |
| 2.1      | FUNDAMENTOS TEÓRICOS   | 3         |
| 2.1.1    | O CICLO DE MARCHA  | 3         |
| 2.1.2    | DEFICIÊNCIAS DA MARCHA DE AMPUTADO   | 6         |
| 2.1.3    | PRÓTESES E ÓRTESES COMERCIAIS  | 6         |
| 2.2      | REVISÃO DE ALGORITMOS DE ESTIMAÇÃO DE FASE DE MARCHA   | 8         |
| 2.2.1    | CONTROLE POR MÁQUINAS DE ESTADO FINITO   | 8         |
| 2.2.2    | CONTROLE POR PERIODICIDADE DE MOVIMENTO  | 10        |
| 2.2.3    | ALGORITMOS DE CLASSIFICAÇÃO DE INTENÇÃO DE MOVIMENTO   | 11        |
| 2.2.4    | DETECÇÃO DE INTENÇÃO DE MOVIMENTO POR ELETROMIOGRAFIA  | 12        |
| <b>3</b> | <b>Verificação do Sistema de Prótese Automatizada RLEG</b>   | <b>13</b> |
| 3.1      | VISÃO GERAL SOBRE O FUNCIONAMENTO DA PRÓTESE RLEG  | 13        |
| 3.2      | COMPONENTES ELETRÔNICOS PERIFÉRICOS  | 15        |
| 3.2.1    | UNIDADE DE MEDIÇÃO INERCIAL - IMU  | 15        |
| 3.2.2    | ENCODER ABSOLUTO AMT20   | 16        |
| 3.2.3    | PISTÃO MAGNETO-REOLÓGICO   | 17        |
| 3.2.4    | <i>Driver</i> DE CORRENTE/PONTE H  | 20        |
| 3.2.5    | AQUISIÇÃO DE DADOS - CARTÃO SD E MÓDULO XBEE   | 21        |
| 3.2.6    | MÓDULO SENSOR DE CORRENTE ACS712   | 23        |
| 3.2.7    | PROTOCOLOS DE COMUNICAÇÃO DE MÓDULOS UTILIZADOS  | 24        |
| 3.3      | INTERFACE DE CONTROLE: MICROCONTROLADOR TEENSY 3.0 E CÓDIGOS<br>IMPLEMENTADOS PARA TESTES EM PACIENTES | 26        |
| 3.3.1    | TEENSY 3.0   | 26        |
| <b>4</b> | <b>Adaptação do sistema RLEG a um microcontrolador de baixo custo</b>                                  | <b>30</b> |
| 4.0.1    | O MICROCONTROLADOR: DESCRIÇÃO  | 30        |
| 4.0.2    | AMBIENTE DE DESENVOLVIMENTO  | 32        |

|            |  |           |
|------------|--|-----------|
| 4.0.3      | CAMADA DE IMPLEMENTAÇÃO HAL PARA PROTOCOLOS DE COMUNICAÇÃO ...                                       | 33        |
| 4.0.4      | FREERTOS .....   | 34        |
| 4.0.5      | DESCRIÇÃO DA GERAÇÃO DE SINAL PWM .....  | 35        |
| <b>5</b>   | <b>Redesign do circuito embarcado do sistema RLEG .....</b>  | <b>37</b> |
| 5.0.1      | CONCEITOS FUNDAMENTAIS DO NOVO <i>design</i> DE <i>hardware</i> .....                                | 37        |
| 5.0.2      | TECNOLOGIAS UTILIZADAS PARA MODELAGEM DE CIRCUITO EMBARCADO.....                                     | 38        |
| 5.0.3      | METODOLOGIA DE FABRICAÇÃO DO CIRCUITO EMBARCADO .....  | 41        |
| 5.0.4      | CALIBRAGEM DE MÁQUINA CNC .....  | 42        |
| <b>6</b>   | <b>Avaliação do funcionamento do circuito e caracterização do sistema RLEG ..</b>                    | <b>44</b> |
| 6.0.1      | AQUISIÇÃO DE DADOS DOS SENSORES DE MEDIÇÃO INERCIAL E ENCODER.....                                   | 44        |
| 6.0.2      | ENSAIOS COM O SENSOR DE CORRENTE.....  | 51        |
| 6.0.3      | ENSAIO ATRAVÉS DA UTILIZAÇÃO DE UM CIRCUITO COM ADIÇÃO DE RESIS-<br>TÊNCIA EM SÉRIE AO ATUADOR ..... | 53        |
| 6.0.4      | TESTES COM USO DE CÓDIGO DE CONTROLE DE AMORTECIMENTO DO PISTÃO<br>POR VELOCIDADES-GATILHO. ....     | 55        |
| <b>7</b>   | <b>Considerações Finais .....</b>  | <b>58</b> |
| 7.1        | CONCLUSÕES .....   | 58        |
| 7.2        | TRABALHOS FUTUROS.....   | 58        |
|            | <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>  | <b>59</b> |
|            | <b>Apêndice .....</b>  | <b>62</b> |
| <b>I</b>   | <b>Esquemáticos da PCB RLEG .....</b>  | <b>63</b> |
| <b>II</b>  | <b>Esquemático do circuito RLEG .....</b>  | <b>64</b> |
| <b>III</b> | <b>Imagens em osciloscópio .....</b>   | <b>66</b> |
| III.1      | ENSAIOS PARA FREQUÊNCIA PWM DE 72 KHz .....  | 66        |
| III.2      | ENSAIOS PARA FREQUÊNCIA PWM DE 20 KHz .....  | 67        |
| III.3      | RESPOSTA DO ATUADOR COM USO DE RESISTORES EM SÉRIE.....  | 68        |
|            | <b>Anexos.....</b>   | <b>69</b> |
| <b>I</b>   | <b>Esquemático do circuito RLEG anterior.....</b>  | <b>70</b> |



# LISTA DE FIGURAS

|      |  |    |
|------|--|----|
| 1.1  | Prótese RLEG desenvolvida no Laboratório de Robótica e Automação, UnB. Fonte: [Ochoa Diaz, 2015].                        | 2  |
| 2.1  | Representação do ciclo de marcha na caminhada. Fonte: Adaptado de [Neumann, 2002].                                       | 4  |
| 2.2  | Planos anatômicos do corpo humano. Fonte: Wikimedia Commons.   | 4  |
| 2.3  | C-Leg (Ottobock) à esquerda; Rheo Knee (Össur), à direita. Fonte: [Ochoa Diaz, 2015].                                    | 7  |
| 2.4  | Pé SACH 1K30 (Ottobock) à esquerda; Proprio Foot (Össur) à direita. Fonte: [Rocha, 2015].                                | 8  |
| 2.5  | Órtese ativada de tornozelo (AAFO) desenvolvida por Blaya, MIT. Fonte: Adaptado de [Blaya, 2000].                        | 9  |
| 2.6  | Máquina de Estados Finitos para controle da AAFO. Fonte: Adaptado de [Blaya, 2000].                                      | 10 |
| 2.7  | Protótipo transfemoral. Fonte: Adaptado de [Varol, 2008].  | 11 |
| 3.1  | Circuito embarcado da prótese RLEG. Fonte: Adaptado de [Rocha, 2015].  | 14 |
| 3.2  | Sistema RLEG com circuito embarcado acoplado. Fonte: Adaptado de [Rocha, 2015].  | 15 |
| 3.3  | Módulo de Medição Inercial Sparkfun com 9 graus de liberdade. Fonte: Sparkfun.   | 16 |
| 3.4  | Encoder Absoluto capacitivo AMT20. Fonte: CUI INC.   | 17 |
| 3.5  | Diagrama com interface de conexão do Encoder Absoluto capacitivo AMT20. Informações do fabricante, CUI INC.              | 17 |
| 3.6  | Pistão amortecedor magneto-reológico produzido pela LORD Coporation. Fonte: LORD.  | 18 |
| 3.7  | Estrutura interna do pistão MR LORD. Imagem adaptada do fabricante.  | 19 |
| 3.8  | Gráfico Força x Velocidade do sistema LORD RD-8040-1 para diferentes correntes aplicadas. Fonte: Adaptado do fabricante. | 20 |
| 3.9  | Módulo Driver de Motor TB6612FNG, da fabricante Pololu. Fonte: Pololu.   | 21 |
| 3.10 | Módulo <i>Sparkfun Openlog Data Logger</i> para comunicação com cartão SD. Fonte: Sparkfun.                              | 23 |
| 3.11 | Módulo de comunicação wireless XBEE. Fonte: Digi Internacional.  | 23 |
| 3.12 | Teensy 3.0, visão superior. Fonte: PJRC.   | 27 |
| 3.13 | Extensão de Arduino para suporte ao microcontrolador Teensy.   | 27 |
| 3.14 | Carregador de programas Teensy Loader.   | 28 |

|       |   |    |
|-------|---|----|
| 4.1   | Interface STM32CubeMX para configuração visual de periféricos internos de dispositivos da família STM32. ....   | 33 |
| 5.1   | Projeto de placa de circuito impresso desenvolvido com uso da ferramenta CAD Eagle. ....  | 40 |
| 5.2   | Componentes utilizados no novo circuito RLEG e seus posicionamentos. Legenda:<br>1. Terminais do botão liga/desliga; 2. Regulador de tensão de 5 V; 3. Conector da bateria; 4. Módulo regulador de tensão de 3,3 V; 5. IMU; 6. <i>Blue Pill</i> ; 7. <i>Driver</i> de corrente; 8. Módulo <i>bluetooth</i> ; 9. Conector do pistão; 10. Divisor de tensão e filtro RC; 11. Módulo ACS712; 12. LEDs indicadores de modo de operação; 13. Conector LCD; 14. Módulo de gravação SD; 15. Conector do encoder..... | 41 |
| 6.1   | Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo Z negativo. ....   | 48 |
| 6.2   | Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo X positivo e negativo. ....  | 48 |
| 6.3   | Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo Y positivo e negativo. ....  | 49 |
| 6.4   | Leituras de velocidade angular através da rotação da placa PCB em torno do eixo X do giroscópio ITG3200. ....   | 49 |
| 6.5   | Leituras de velocidade angular através da rotação da placa PCB em torno do eixo Y do giroscópio ITG3200. ....   | 50 |
| 6.6   | Leituras de velocidade angular através da rotação da placa PCB em torno do eixo Z do giroscópio ITG3200.....  | 50 |
| 6.7   | Leituras de ângulo de rotação do joelho protético, obtidas por meio do encoder AMT20. ....  | 51 |
| 6.8   | Montagem realizada para medição do sinal de corrente $I_{damper}$ . ....  | 53 |
| 6.9   | Tensão nos terminais da resistência utilizada de $1 \Omega$ para um 5 V médio de entrada e frequência 72 kHz. ....  | 54 |
| 6.10  | Borda de subida do sinal PWM gerado pela <i>Blue pill</i> , com resposta do sinal de corrente.....  | 54 |
| 6.11  | Borda de descida do sinal PWM gerado pela <i>Blue pill</i> , com resposta do sinal de corrente.....   | 55 |
| 6.12  | FSM para controle de amortecimento por velocidades-gatilho. ....  | 56 |
| 6.13  | Resultados obtidos para código de atuação por tensão com adição de região de histerese. ....  | 57 |
| II.1  | Modelo da PCB RLEG com contorno de componentes. ....  | 63 |
| II.2  | Modelo da PCB RLEG com destaque de trilhas. ....  | 63 |
| III.1 | $V_{ACS}$ para $V_{AVG} = 2V$ , 72 kHz. ....  | 66 |
| III.2 | $V_{ACS}$ para $V_{AVG} = 4V$ , 72 kHz. ....  | 66 |
| III.3 | $V_{ACS}$ para $V_{AVG} = 6V$ , 72 kHz. ....  | 66 |

|   |    |
|---|----|
| III.4 $V_{ACS}$ para $V_{AVG} = 8V$ , 72 kHz. ....                              | 66 |
| III.5 $V_{ACS}$ para $V_{AVG} = 2V$ , 20 kHz. ....                              | 67 |
| III.6 $V_{ACS}$ para $V_{AVG} = 4V$ , 20 kHz. ....                              | 67 |
| III.7 $V_{ACS}$ para $V_{AVG} = 6V$ , 20 kHz. ....                              | 67 |
| III.8 $V_{ACS}$ para $V_{AVG} = 8V$ , 20 kHz. ....                              | 67 |
| III.9 Resposta para $V_{AVG} = 5V$ , 72 kHz. Divisão de tempo 10 $\mu s$ . .... | 68 |
| III.10 Resposta para $V_{AVG} = 5V$ , 72 kHz. Divisão de tempo 5 $\mu s$ . .... | 68 |
| III.11 Resposta para $V_{AVG} = 5V$ , 72 kHz, borda de subida. ....             | 68 |
| III.12 Resposta para $V_{AVG} = 5V$ , 72 kHz, borda de descida. ....            | 68 |
| III.13 Resposta transiente em destaque. $V_{AVG} = 5V$ , 72 kHz. ....           | 68 |

# LISTA DE TABELAS

|     |  |    |
|-----|--|----|
| 2.1 | Eventos relacionados em inglês que ocorrem em cada subfase do ciclo de apoio. ....   | 5  |
| 2.2 | Eventos relacionados em inglês que ocorrem em cada subfase do ciclo de balanço. ...  | 5  |
| 2.3 | Eventos relacionados ao disparo de cada uma das subfases do ciclo de marcha e suas durações. ....  | 6  |
| 3.1 | Características elétricas típicas do pistão magneto-reológico LORD RD-8040-1. ....   | 20 |
| 3.2 | Características elétricas e mecânicas típicas do <i>driver</i> de motor TB6612FNG. ....  | 21 |
| 3.3 | Operações básicas de manipulação de arquivos no modo de Comando do <i>Sparkfun Data-Logger</i> . ....  | 22 |
| 3.4 | Características elétricas típicas do sensor de corrente ACS712. ....   | 24 |
| 3.5 | Modos de transmissão e captura de dados SPI. Fonte: Adaptado da documentação SPI - Arduino. ....   | 25 |
| 3.6 | Características Eletrônicas - Teensy 3.0. Fonte: Adaptado da documentação PJRC, <i>Teensy USB Development Board</i> . ....   | 27 |
| 4.1 | Especificações de <i>hardware</i> do microcontrolador STM32F103C8. ....  | 31 |
| 4.2 | Características de <i>hardware</i> dos microcontroladores Arduino Nano e Teensy 3.0. ....  | 32 |
| 4.3 | Principais funções de comunicação - camada HAL. ....   | 34 |
| 4.4 | Parâmetros definidos para geração de sinal em modulação de largura de pulso. ....  | 36 |
| 5.1 | Modificações aplicadas ao circuito embarcado RLEG. ....  | 39 |
| 6.1 | Registradores de configuração e leitura de dados do acelerômetro ADXL345. ....   | 45 |
| 6.2 | Registradores de configuração e leitura de dados do giroscópio ITG3200. ....   | 46 |
| 6.3 | Tensão média de sinal PWM, tensão média de saída do sensor de corrente ACS712 e corrente $I_{damper}$ calculada, para uma frequência PWM constante de 72 kHz. .... | 52 |
| 6.4 | Tensão média de sinal PWM, tensão média de saída do sensor de corrente ACS712 e corrente $I_{damper}$ calculada, para uma frequência PWM constante de 20 kHz. .... | 53 |
| 6.5 | Parâmetros aplicados ao sistema de atuação do pistão LORD por meio de tensão, com região de histerese. ....  | 56 |

# LISTA DE SÍMBOLOS

## Símbolos Latinos

|              |   |        |
|--------------|---|--------|
| $I_{damper}$ | Corrente de entrada ao pistão amortecedor                                       | A      |
| $a_p$        | Profundidade de corte   | mm     |
| $v_f$        | Velocidade de avanço  | mm/min |
| $V_{ACS}$    | Tensão de saída do sensor de corrente ACS712                                    | V      |
| $V_{offset}$ | Tensão de <i>offset</i> do sensor de corrente ACS712                            | V      |
| $V'_{ACS}$   | Tensão aplicada ao microcontrolador para leitura do sinal analógico de corrente | V      |
| $V_{avg}$    | Valor médio de tensão aplicada ao pistão  | V      |
| $V_{min}$    | Valor mínimo de tensão de sinal PWM   | V      |
| $V_{max}$    | Valor máximo de tensão de sinal PWM   | V      |
| $d$          | Ciclo de trabalho   | -      |
| $t_{sample}$ | Tempo de amostragem   | s      |

## Símbolos Gregos

|                  |  |     |
|------------------|--|-----|
| $\omega$         | Velocidade angular                           | °/s |
| $\omega_{upper}$ | Valor-gatilho superior de velocidade angular | °/s |
| $\omega_{lower}$ | Valor-gatilho inferior de velocidade angular | °/s |
| $\theta$         | Ângulo do joelho                             | °   |

## Siglas

|       |  |
|-------|--|
| HS    | <i>Heel Strike</i>                                       |
| FF    | <i>Foot Flat</i>   |
| SL    | <i>Single Led Support</i>                                |
| HO    | <i>Heel Off</i>  |
| TO    | <i>Toe Off</i>   |
| SACH  | <i>Solid Ankle Cushioned Heel</i>                        |
| FSM   | <i>Máquina de Estados Finitos - Finite State Machine</i> |
| AAFO  | <i>Ankle Active Foot Orthosis</i>                        |
| SEA   | <i>Series Elastic Actuator</i>                           |
| GRF   | <i>Ground Reaction Forces</i>                            |
| GMM   | <i>Gaussian Mixture Model</i>                            |
| LHMM  | <i>Layered Hidden Markov Model</i>                       |
| EMG   | <i>Eletromiografia</i>                                   |
| PCB   | <i>Printed Circuit Board</i>                             |
| IMU   | <i>Inertial Measurement Unit</i>                         |
| PWM   | <i>Pulse-Width Modulation</i>                            |
| I2C   | <i>Inter-Integrated Circuit</i>                          |
| SDA   | <i>Serial Data</i>                                       |
| SCL   | <i>Serial Clock</i>                                      |
| SPI   | <i>Serial Peripheral Interface</i>                       |
| MISO  | <i>Master In Slave Out</i>                               |
| MOSI  | <i>Master Out Slave In</i>                               |
| SCK   | <i>Serial Clock</i>                                      |
| CS/SS | <i>Chip Select/Slave Select</i>                          |
| MR    | <i>Magneto-reológico</i>                                 |
| DC    | <i>Direct Current</i>                                    |
| VMOT  | <i>Tensão de alimentação do motor</i>                    |
| UART  | <i>Universal Asynchronous Receiver/Transmitter</i>       |
| RX    | <i>Receiver line</i>                                     |
| TX    | <i>Transmitter line</i>                                  |
| CPOL  | <i>Clock Polarity</i>                                    |
| CPHA  | <i>Clock Phase</i>                                       |
| MSB   | <i>Most Significant Bit/Most Significant Byte</i>        |
| RAM   | <i>Random Access Memory</i>                              |
| DMA   | <i>Direct Memory Access</i>                              |
| MCU   | <i>Microcontroller Unit</i>                              |
| HAL   | <i>Hardware Abstraction Layer</i>                        |

|      |                                     |
|------|-------------------------------------|
| ADC  | <i>Analog-to-Digital Converter</i>  |
| RTOS | <i>Real Time Operating System</i>   |
| ARR  | <i>Auto-Reload Register</i>         |
| CCR  | <i>Capture-Compare Register</i>     |
| CAD  | <i>Computer-Aided Design</i>        |
| CAM  | <i>Computer-Aided Manufacturing</i> |
| CNC  | <i>Computer Numerical Control</i>   |

# Capítulo 1

## Introdução

Este capítulo apresenta as seções introdutórias que motivaram a implementação deste Trabalho de Graduação. Dessa forma, contextualiza o problema, assim como define a abordagem adotada para sua solução e dados numéricos que o justifiquem.

### 1.1 Contextualização

A amputação de um membro causa limitações motoras que afetam a rotina dos pacientes a que ela são submetidos. Uma amputação é decorrente de necrose tecidual que pode ser causada por diversos fatores, dentre eles: complicações de diabetes e doença vascular periférica (75% dos casos) e trauma (20% dos casos), de acordo com o verificado em [1]. Neste estudo, 181 prontuários foram avaliados, correspondendo a pacientes do Centro Regional de Reabilitação da cidade de Araraquara, São Paulo. Um tempo médio de três anos foi observado para implantação das próteses necessárias, com um alto custo associado. Próteses do tipo transfemoral, em especial, apresentaram valor elevado em relação a outras modalidades. Pode-se destacar ainda a ocorrência de acidentes de trânsito como fator originário de aproximadamente metade das amputações por trauma.

No estudo epidemiológico citado em [1] para amputações de membro inferior, observou-se uma predominância de pacientes do sexo masculino (74%). Em um outro estudo [2], 474 prontuários foram analisados, dos quais 72% correspondiam a homens, confirmando uma correlação associada ao gênero. Amputações transtibiais corresponderam a 44% dos casos de amputação de membro inferior, ao passo que amputações transfemorais corresponderam a 43%. Adicionalmente, tem-se que a causa das amputações encontradas foi predominantemente associada a quadros de origem vascular (72%), enquanto as traumáticas corresponderam a uma menor taxa (18%), porém, ainda significativa. De acordo com o mesmo estudo, uma média de idade de 56 anos foi verificada e cerca de 56% dos pacientes foram protetizados.

Dadas as condições a que um paciente que sofre amputação de membro inferior é submetido, com necessidade de readaptação de marcha e inúmeras dificuldades associadas, o presente estudo aborda aspectos da marcha humana padrão e suas discrepâncias em relação a indivíduos inseridos



no grupo abordado. Um estudo como este visa, assim, uma maior acessibilidade de próteses funcionais a indivíduos que dela necessitam, melhorando a qualidade de vida destes, diminuindo o período de adaptação, e provendo uma ferramenta eficaz e pouco custosa para suas necessidades motoras.

## 1.2 Definição do problema

Este trabalho visa atuação sobre uma prótese de membro inferior para indivíduos que sofreram amputação transfemoral. A prótese, denominada RLEG, foi desenvolvida no Laboratório de Robótica e Automação (LARA), Universidade de Brasília, e constitui-se em uma prótese automatizada, visto que é capaz de modificar seu fator de amortecimento através de um pistão magneto-reológico acoplado, atuando como o joelho de um indivíduo saudável. O amortecimento variável é controlado por meio da corrente elétrica aplicada ao atuador. Um controle eficaz da corrente aplicada é, portanto, indispensável para a qualidade do movimento realizado, assim como para uma economia de energia metabolizada empregada a esta tarefa. A figura 1.1 ilustra a prótese desenvolvida nos trabalhos de [3] e [4].



Figura 1.1: Prótese RLEG desenvolvida no Laboratório de Robótica e Automação, UnB. Fonte: [Ochoa Diaz, 2015].

## 1.3 Objetivos do projeto

Este projeto visa um levantamento das funcionalidades implementadas sobre a prótese RLEG, visto que o projeto realizado que culminou a obtenção do sistema encontrado teve fim no ano de 2015, deixando o sistema intocado deste então. Além disso, visa a adição de novas características ao sistema e um maior estudo sobre seu elemento atuador. Portanto, para a recuperação do projeto e implementação de possíveis melhorias, realizou-se um estudo sobre seu funcionamento, sendo este descrito com detalhamento neste trabalho. Mudanças de *design* foram realizadas para correção de falhas, e melhorias de projetos em *software* também foram aplicadas. Destaca-se, aqui, a falta de uma documentação existente sobre o design de circuito e software aplicados à prótese, existindo trabalhos descritivos apenas no âmbito de estudos sobre a marcha de amputados, de acordo com o trabalho presente em [3], e da realização mecânica da prótese, que pode ser encontrada em [4].

## Capítulo 2

# Fundamentos e Revisão

O presente capítulo aborda os conceitos necessários ao entendimento da marcha humana, assim como expõe situações presentes na marcha de indivíduo amputado. Por fim, realiza uma investigação da literatura no que concerne ao uso de algoritmos de estimação de fase de marcha e rotinas de controle de movimento.

### 2.1 Fundamentos Teóricos

Aqui, os aspectos biomecânicos da marcha humana durante o processo de caminhada são explicitados, assim como de que forma a dinâmica de marcha de um indivíduo amputado é afetada. Posteriormente, uma breve análise das opções de próteses comerciais existentes atualmente é realizada.

#### 2.1.1 O Ciclo de Marcha

O ciclo de marcha humana pode ser definido como um esforço periódico realizado para promover o avanço do corpo, sem perda de estabilidade [5]. Cada fase do ciclo e suas subdivisões estão associadas a movimentos realizados por juntas de interesse: tornozelo, joelho e quadril. Um ciclo é caracterizado adotando-se um lado do corpo do indivíduo, ou membro, como sendo o de referência. Com relação a ele, duas fases são definidas: **apoio** e **balanço**. A imagem 2.1 explicita as etapas do ciclo de marcha, abordadas a seguir. A descrição da marcha como apresentada a seguir se baseia nos trabalhos de [3], [4] e [6].

A fase de apoio inicia-se quando a perna de referência entra em contato com o solo e continua até que o membro perca tal contato. A fase de balanço então é iniciada e se dá com a perna sendo impulsionada, no ar, para a frente, permanecendo até que um novo contato com o chão ocorra.

As duas fases descritas são ainda subdivididas para uma melhor classificação e tratamento de patologias. No apoio, o **contato inicial** corresponde ao evento de primeiro contato do calcanhar com o chão; a **resposta à carga**, à fase em que o pé se aplaina com a superfície; o **apoio médio** corresponde ao momento em que o peso do corpo é posto totalmente sobre a perna de referência;

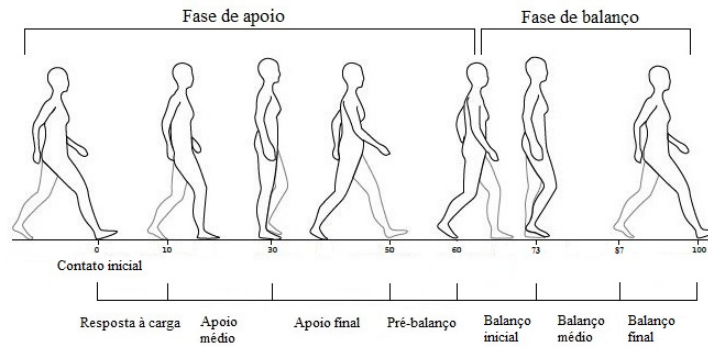


Figura 2.1: Representação do ciclo de marcha na caminhada. Fonte: Adaptado de [Neumann, 2002].

o **apoio terminal** é definido como o instante em que o calcanhar perde contato com o solo. A última fase do apoio consiste no **pré-balanço**, em que é iniciada a transferência de peso da perna de referência à perna opositora.

É importante observar ainda que o ciclo de apoio pode ser categorizado em fases de **apoio duplo** e fases de **apoio simples**. Um apoio duplo ocorre quando a carga corporal é distribuída entre as duas pernas do indivíduo, já o apoio simples consiste na carga atuando apenas sobre o lado de referência. Durante a subfase de resposta à carga, período que se inicia no contato inicial e corresponde a 10% do ciclo de marcha, ocorre o apoio duplo. Esta configuração também é presente na transição da fase de apoio para o balanço, desde a subfase de apoio final até o início da subfase de pré-balanço. Para todos os instantes remanescentes da fase de apoio, o suporte é do tipo simples.

Durante a etapa de apoio, que corresponde a 60% do ciclo total de marcha, um indivíduo deve ser capaz de manter seu próprio peso corporal, além de controlar a estabilidade de seu movimento por meio do controle dos ângulos das articulações envolvidas. A figura 2.2 explicita o corpo humano e seus planos de estudo. O plano sagital é o mais relevante para o estudo do ciclo de marcha, por nele ocorrerem movimentações de maior amplitude.

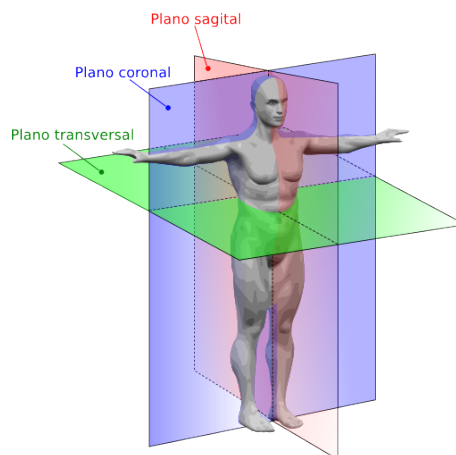


Figura 2.2: Planos anatômicos do corpo humano. Fonte: Wikimedia Commons.

A importância da fase de balanço está associada à manutenção da estabilidade corporal. A subfase de **balanço inicial** é definida como o momento em que os dedos deixam o solo, enquanto que a perna de referência avança; e o **balanço médio**, o momento em que a perna de referência ultrapassa a de suporte. O **balanço terminal** consiste na etapa final do ciclo de marcha, com o joelho totalmente estendido e o calcanhar retornando à sua configuração inicial. Cada uma das fases e subfases descritas apresentam termos correspondentes no inglês, com siglas bem específicas de eventos que ocorrem durante cada uma delas. A correspondência entre tais terminologias é realizada por acreditar-se na simplicidade dos eventos definidos para o inglês, tornando a modelagem mais intuitiva. As tabelas 2.1 e 2.2 explicitam tais relações.

| Fase de Apoio                |                                    |                         |
|------------------------------|------------------------------------|-------------------------|
| Subfase do ciclo             | Característica                     | Evento                  |
| Contato inicial <sup>1</sup> | Contato do calcanhar               | Heel Strike (HS)        |
| Resposta à carga             | Pé se torna plano                  | Foot Flat (FF)          |
| Apoio médio                  | Carga total na perna de referência | Single Leg Support (SL) |
| Apoio terminal               | Calcanhar deixa o solo             | Heel Off (HO)           |
| Pré-balanço                  | Início do apoio da perna oposta    | -                       |

Tabela 2.1: Eventos relacionados em inglês que ocorrem em cada subfase do ciclo de apoio.

| Fase de Balanço  |   |              |
|------------------|---|--------------|
| Subfase do ciclo | Característica                            | Evento       |
| Balanço inicial  | Dedos deixam o solo                       | Toe Off (TO) |
| Balanço médio    | Membro de referência ultrapassa o outro   | -            |
| Balanço terminal | Calcanhar retorna à posição de referência | -            |

Tabela 2.2: Eventos relacionados em inglês que ocorrem em cada subfase do ciclo de balanço.

As subfases do ciclo de marcha podem ainda ser melhor delimitadas por eventos que acarretam o disparo de cada uma delas, de acordo com o descrito em [6]. Dessa maneira, tem-se que a fase de **resposta à carga** é iniciada pelo contato inicial e tem duração até que os dedos da perna opositora deixem o solo, em aproximadamente 10% do ciclo de marcha. Logo após, a fase de **balanço médio** é iniciada, terminando com o ato de elevação do calcanhar da perna de referência, após o suporte de peso ocorrer sobre ela. O **balanço terminal** então começa, tendo fim à ocorrência do evento de contato inicial da perna opositora do indivíduo. A fase de **pré-balanço** é iniciada logo após, sendo finalizada com o evento de *Toe Off* do lado de referência.

No que corresponde às subfases de balanço, o **balanço inicial** tem início após o evento de TO e término no momento em que as pernas de referência e opositora se encontram lado-a-lado. A partir de então, inicia-se a subfase de **balanço médio**, até que a tíbia da perna de referência se oriente em posição vertical, o que ocorre em aproximadamente 87% do ciclo de marcha. Por fim,

<sup>1</sup>O contato inicial corresponde, em termos mais rigorosos, a um **evento** do ciclo de marcha humana.

inicia-se a subfase de **balanço terminal**, que encerra o ciclo total quando da ocorrência de um novo contato inicial, como definido anteriormente. A tabela 2.3 explicita os eventos de disparo de cada subfase de marcha, assim como indica as porcentagens do ciclo total que correspondem à duração de cada uma destas etapas.

| Ciclo total de marcha |  |           |
|-----------------------|--|-----------|
| Subfase do ciclo      | Iniciado por                                 | Duração   |
| Resposta à carga      | Contato inicial                              | 0 a 10%   |
| Apoio médio           | Dedos da perna opositora deixam o solo       | 10 a 30%  |
| Apoio terminal        | Elevação do calcanhar                        | 30 a 50%  |
| Pré-balanço           | Início do apoio da perna oposta              | 50 a 60%  |
| Balanço inicial       | Dedos deixam o solo                          | 60 a 73%  |
| Balanço médio         | Pernas de referência e opositora lado-a-lado | 73 a 87%  |
| Balanço terminal      | Alinhamento vertical de tíbia                | 87 a 100% |

Tabela 2.3: Eventos relacionados ao disparo de cada uma das subfases do ciclo de marcha e suas durações.

### 2.1.2 Deficiências da Marcha de Amputado

Uma marcha em condições normais existe sob uma condição de simetria em relação ao plano sagital corporal. Ao sofrer uma amputação, um indivíduo deixa de possuir um corpo com simetria, e, portanto, discrepâncias em relação à marcha padrão são observadas. Devido a limitações das próteses utilizadas, é observado que a maioria dos pacientes que sofreram amputações utilizam o membro sadio para compensar as limitações da nova condição imposta. Com isso, muitos sofrem risco de desenvolver patologias decorrentes do esforço aplicado. Dentre as condições encontradas, pode-se citar a ocorrência de osteoartrite, no membro saudável, devido ao esforço adicional empregado, e de osteoporose no membro amputado [3].

Em relação a aspectos dinâmicos da marcha, um maior tempo de apoio sobre membro intacto é observado em relação ao tempo sobre a perna de prótese do sujeito. Tal condição seria explicada pela falta de confiança no equipamento utilizado. Ainda, por limitações de prótese, uma configuração frequente de joelho protético totalmente estendido é capaz de acarretar desbalanceamentos de distribuição de carga, com maiores forças de reação aplicadas ao membro saudável. Aliada a essa condição, a realização de inclinação do tronco pode se fazer presente para contornar a falta de angulação do joelho da prótese, e obter espaço suficiente para a realização da etapa de balanço do ciclo de marcha.

### 2.1.3 Próteses e Órteses Comerciais

Uma prótese é utilizada para assistir o indivíduo no novo trabalho de movimentação pós-amputação. Para isso, deve atender requisitos de segurança, peso e conforto. Ainda, deve ser

esteticamente agradável, característica indispensável ao usuário. Cada prótese é ajustada para a pessoa que a utiliza; um ajuste mal-feito causará patologias ao paciente.

Uma prótese transfemoral completa, como a desenvolvida em [4], conta com quatro componentes: soquete, joelho, extensor tibial e conjunto tornozelo-pé. O soquete é responsável por fazer a conexão entre o coto e a prótese. Dessa forma, o seu preparo deve ser correto para evitar pontos de tensão de forças e torques.

Dentre as opções de próteses de joelho presentes atualmente no mercado, diversas apresentam mecanismos de microprocessamento, caracterizando-se em próteses ativas; porém, esta configuração não representa a maioria das ofertas. A primeira opção introduzida deste tipo foi a C-Leg da fabricante alemã Ottobock, capaz de controlar a marcha através do amortecimento provido por um atuador hidráulico. Outras opções encontradas de próteses de joelho ativas são a The Rheo Knee (Össur), utilizando um sistema de freio magneto-reológico. A imagem 2.3 demonstra as próteses de joelho mencionadas.

A prótese denominada BIOM, também se utilizando do mesmo mecanismo de freio, e comercializada pela BIOM Personal Bionics, é resultado de trabalhos acadêmicos conduzidos no MIT. A prótese constitui-se em uma alternativa avançada capaz de atuar em terrenos irregulares e diversos tipos de marcha, de acordo com o verificado em [3]. Quanto a próteses do conjunto tornozelo-pé, os modelos mais comuns são rígidos, feitos de materiais flexíveis, sendo estes SACH (*Solid Ankle Cushioned Heel*) ou carbono. Modelos encontrados para esta categoria também são de fabricantes Ottobock e Össur, como descrito em [4]. A imagem 2.4 explicita dois exemplos, Pé 1K30, Ottocock, e Proprio Foot, Össur.



Figura 2.3: C-Leg (Ottobock) à esquerda; Rheo Knee (Össur), à direita. Fonte: [Ochoa Diaz, 2015].



Figura 2.4: Pé SACH 1K30 (Ottobock) à esquerda; Proprio Foot (Össur) à direita. Fonte: [Rocha, 2015].

## 2.2 Revisão de Algoritmos de Estimação de Fase de Marcha

Diversos trabalhos acadêmicos desenvolvidos nas áreas de algoritmos de estimação de fase de marcha e tipos de controle empregado foram verificados, devido à necessidade de um sistema mais robusto que o já constituído pelo projeto RLEG. O estudo desenvolvido motivou o trabalho em relação à verificação das funcionalidades do sistema RLEG, além de fornecer uma maior contextualização das vantagens e desvantagens dos sistemas de controle existentes na literatura. Ainda, os modelos aqui expostos visam inspirar trabalhos futuros de continuação deste projeto.

Esta seção explicita uma diversidade de sistemas encontrados, em sua maioria tratando o ciclo de marcha como uma Máquina de Estados Finitos (*Finite State Machine – FSM*). Neste modelo, transições entre cada um dos estados são disparadas por meio da ultrapassagem de valores-limite de sinais proveniente de sensores acoplados ao sistema. Porém, outras abordagens encontradas também são discutidas. O uso de Eletromiografia é uma ferramenta que acrescenta maior confiabilidade à previsão da intenção de movimento, porém, aumenta a complexidade do sistema implementado. Grande parte dos artigos revisados são referentes ao estudo produzido em [7].

### 2.2.1 Controle por Máquinas de Estado Finito

Devido à natureza sequencial da marcha humana, com eventos de disparo de cada uma de suas fases e subfases, uma implementação em Máquina de Estados Finito é intuitiva de ser realizada. Assim, pode-se definir estados de marcha conforme desejado, com transições desencadeadas por eventos percebidos pelo sistema de identificação conforme as ações do indivíduo que realiza o movimento de avanço corporal.

Em [8], uma órtese ativa de tornozelo (*Ankle Active Foot Orthosis – AAFO*) é criada para assistência de um paciente que apresenta uma condição conhecida como arrasto de pé (*drop foot*). Nela, o sujeito apresenta dificuldade para levantar a parte frontal do pé, em movimento de dorsiflexão. Com isso, visa-se diminuir a ocorrência de **batidas** de pé, ou tapas, durante a fase de apoio, assim como eliminação de **arrasto de pé** na fase de balanço.

O sistema utiliza um atuador do tipo SEA (*Series Elastic Actuator*) para controle de ângulo do tornozelo. Este tipo de sensor é amplamente utilizado em sistemas biomecatrônicos por prover baixa impedância e movimentos suaves, visto que o motor não está diretamente em contato com

a carga aplicada. Ainda, um sensor de ângulo é utilizado, assim como sensores capacitivos de força para medição dos parâmetros de interesse: o ângulo do tornozelo (entre a tíbia e o pé), e a constante elástica de torção da mola presente no SEA. Por fim, um botão é posicionado para detecção do toque do calcanhar com o solo, em um evento de *Heel Strike*, conforme definido no capítulo anterior. A imagem 2.5 explicita a órtese desenvolvida.

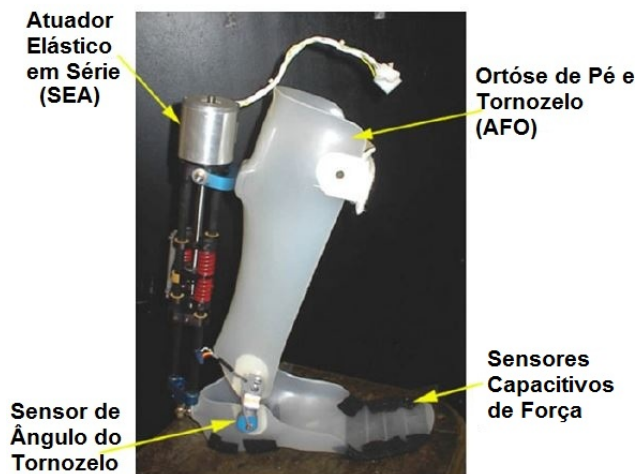


Figura 2.5: Órtese ativa de tornozelo (AAFO) desenvolvida por Blaya, MIT. Fonte: Adaptado de [Blaya, 2000].

Uma FSM de quatro estados é criada para realizar-se o controle de assistência ao usuário. Três destes estados implementam um objetivo de controle bem definido durante o ciclo de marcha, enquanto o estado adicional é acionado ao atingir-se valores medidos fora do escopo desejado (estado de segurança). Para atingir-se a finalidade desejada, os estados, denominados **Contato 1**, **Contato 2** e **Balanço**, ocorrem durante os seguintes instantes da marcha:

1. Contato 1: ocorre do contato inicial (HS) até o balanço-médio, quando o pé e a tíbia se tornam perpendiculares. O controle visa diminuir a ocorrência de *batidas de pé*.
2. Contato 2: do balanço-médio ao pré-balanço. Nesta etapa não há interferência no movimento padrão da marcha do usuário.
3. Balanço: etapa quando o pé não está em contato com o chão. O controle deve levantar o membro do assistido, a fim de evitar o *arrasto de pé*.

A máquina de estados presente na figura 2.6 ilustra o comportamento da órtese. Partindo-se do Contato 1 (contato inicial), pode-se atingir o estado 2 caso a força de reação no solo (*Ground Reaction Force – GRF*) seja maior que o limiar especificado para ela e o ângulo do tornozelo seja maior que a referência (o valor de referência ocorre quando tíbia e tornozelo são perpendiculares).

Do estado de Contato 2, pode-se transicionar para o Balanço através da medição da GRF. Assim, se esta variável é menor que o mínimo para contato com o solo, sabe-se que o membro do



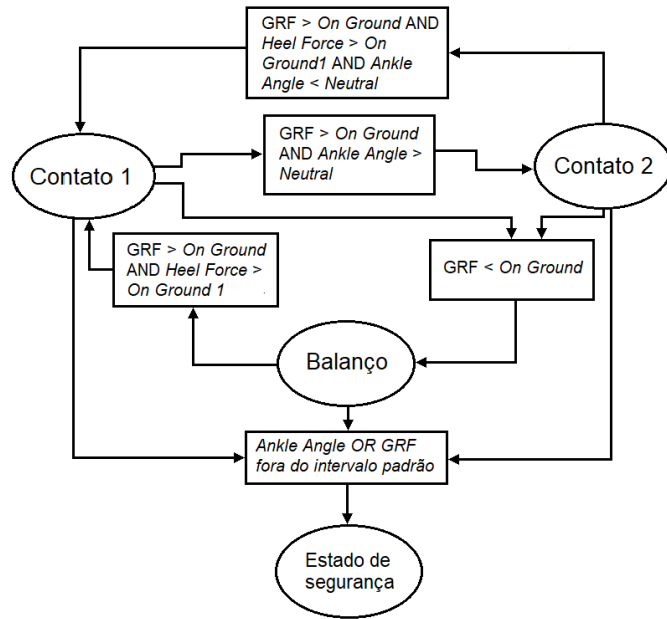


Figura 2.6: Máquina de Estados Finitos para controle da AAFO. Fonte: Adaptado de [Blaya, 2000].

indivíduo está suspenso. O ciclo reinicia-se após o retorno ao estado 1, disparado quando detecta-se novamente o contato do pé com o solo (força GRF maior que um limiar) e um evento *Heel Strike*. Caso as medições tenham valores fora do esperado, o estado de segurança é ativado desde cada um dos outros três estados. É importante notar que os estados de Contato 1 e 2 apresentam transições para todos os outros estados e que o estado de Balanço apresenta transição possível apenas para Contato 1 e o estado de segurança.

Com relação aos resultados obtidos com o sistema AAFO, conseguiu-se eliminar a ocorrência do arrasto para uma velocidade de marcha de até 1,2 m/s. Para maiores velocidades, ocorrências foram reduzidas se comparadas aos resultados sem o uso da órtese. Ainda, com relação ao ângulo de dorsiflexão do tornozelo, na fase de balanço, foram obtidos resultados próximos aos ideais para uma impedância variável aplicada ao atuador. Um paciente reportou que a órtese fez com que seu movimento fosse extremamente próximo ao de marcha natural.

### 2.2.2 Controle por Periodicidade de Movimento

Devido à natureza periódica do ciclo de marcha, pode-se obter uma função ciclo que relaciona o tempo percentual decorrido do ciclo de marcha de um usuário desde o evento que determina o início deste. Assim, trabalhos como [9] e [10] adotam esta abordagem. No primeiro, uma marcha padrão é definida e ajustada conforme o tempo de passada do indivíduo, onde evento de *Heel Strike* determina seu gatilho. No segundo, o ângulo da tíbia e sua velocidade angular são utilizados para ajuste de uma função que relaciona o percentual do ciclo, ângulo do tornozelo e tamanho de passada.

### 2.2.3 Algoritmos de Classificação de Intenção de Movimento

Uma outra abordagem verificada para determinação de fase de marcha ou intenção de movimento são os algoritmos de classificação. Tais algoritmos implementam um esquema de probabilidades de ocorrência de um certo evento, no caso, uma determinada fase ou movimento esperado. Dois estudos foram encontrados: um que realiza a implementação de um Modelo de Misturas de Gaussianas, *Gaussian Mixture Model* – GMM, e outro, um modelo de Markov oculto.

Em [11], tem-se o reconhecimento de intenção em tempo real para determinação do modo de marcha de uma prótese transfemoral. Os sinais adquiridos para controle da prótese são de posições e velocidades das juntas do joelho e tornozelo, torque no soquete da prótese, e forças atuando no calcanhar e planta do pé. A figura 2.7 demonstra o protótipo implementado.



Figura 2.7: Protótipo transfemoral. Fonte: Adaptado de [Varol, 2008].

Um conjunto de dados de treinamento é utilizado para determinar os parâmetros de classificação do GMM. As classificações de modos de marcha se encaixam nas categorias de: **caminhada** (devagar, normal e rápido) e **parado**. Ainda, o resultado proveniente do GMM é associado a um **esquema de votação**, em que um controlador modifica sua condição de caminhada para parado apenas se as últimas  $n$  decisões de classificação concordarem em 90% dos casos. Após classificação de modo de marcha, o sistema realiza o chaveamento para o controlador do modo indicado. Os resultados confirmam a detecção da intenção de movimento para as categorias definidas, porém com testes realizados em um paciente sem amputações.

Um outro estudo encontrado em [12] utiliza um modelo de Markov oculto em camadas, *Layered Hidden Markov Model* – LHMM, para estimação de detecção de movimento voltado à teleoperação cirúrgica. A modelagem conta com duas camadas de LHMM: a primeira define **gestos** primários, ao passo que a segunda camada, mais superior, utiliza um **conjunto de gestos** identificados para classificação da tarefa completa que se deseja realizar. Uma das dificuldades presentes neste

modelo consiste em uma implementação em tempo real, visto que não se saberia, ao certo, quando um gesto termina e outro começa. Porém, para o escopo adotado, sem *real-time*, o sistema se demonstrou eficaz.

#### 2.2.4 Detecção de Intenção de Movimento por Eletromiografia

Dentre os algoritmos existentes para detecção de intenção de movimento, grande parte utiliza sinais de eletromiografia (EMG) provenientes do membro residual do paciente. Dados de EMG medidos em um indivíduo podem assim ser relacionados a dados-padrão para cada movimento realizado pelo corpo humano, atuando como um mapa da intenção deste.

Devido às características não-lineares e descontínuas dos sinais de eletromiografia, estes são muitas vezes associados a esquemas de redes neurais. O trabalho de *Au et al.*, [13], implementa dois tipos de controlador de ângulo de tornozelo em uma prótese de pé: um modelo associado a rede neural e outro em que se construiu um modelo muscular dinâmico para a junta tornozelo-pé.

Os resultados observados para o modelo muscular, ou biomimético, foram superiores aos encontrados para a rede neural. Isto pode ser explicado pela característica mais intuitiva de uma modelagem muscular em relação a um modelo de rede, visto que este último não modela a dinâmica real do comportamento do membro de estudo. Portanto, para o usuário, é mais fácil lidar com modelos que se aproximam à dinâmica natural de marcha, o que ocorre no primeiro caso.

## Capítulo 3

# Verificação do Sistema de Prótese Automatizada RLEG

Este capítulo apresenta uma descrição detalhada acerca do funcionamento do protótipo RLEG desenvolvido no Laboratório de Controle e Automação (LARA), tal como encontrado. Devido à falta de uma documentação explícita sobre o funcionamento de seu circuito embarcado, uma investigação sobre as funcionalidades da prótese se fez extremamente necessária. Portanto, cada componente presente no circuito eletrônico da placa PCB encontrada é explicitado e discutido, assim como os programas que visavam a aplicação de rotinas iniciais de controle sobre o atuador magneto-reológico. Por fim, uma breve análise sobre o conjunto total do sistema é feita, justificando as modificações realizadas posteriormente sobre este trabalho.

### 3.1 Visão geral sobre o funcionamento da prótese RLEG

O sistema denominado RLEG constitui-se em uma prótese automatizada de amputados de membro inferior. Por meio da atuação de um pistão de fluido magnetizável, de impedância mecânica variável, é gerada uma resposta ao movimento e forças que são impressos a tal mecanismo. Dessa forma, o sistema foi construído sob a forma de circuito impresso com componentes eletrônicos acoplados, e ainda *scripts* para testes desenvolvidos em linguagem C++ e aplicados ao ambiente Arduino de desenvolvimento.

O funcionamento da prótese ocorre, de uma forma mais simplificada, da seguinte maneira: através da leitura de sensores de medição inercial (IMU) ou mesmo do ângulo de flexão do joelho, por meio de um encoder capacitivo, é possível que se identifique parâmetros de interesse que definem um sinal de controle de tensão aplicada a um pistão magneto-reológico, alterando-se assim sua impedância. Os dados adquiridos podem ser verificados através do *logging* destes em um cartão SD, ou, ainda, comunicados em tempo real através do módulo de transmissão wireless XBEE. No circuito impresso desenvolvido para acoplamento dos componentes, existe ainda um sensor de corrente, sendo, dessa forma, possível que se monitore a corrente que é efetivamente aplicada ao pistão, cuja impedância é resultado direto da corrente aplicada, e não do nível de tensão percebido.

Adicionalmente, um *driver* de corrente é utilizado, uma vez que o chip responsável por implementar o controle de tensão não é capaz de fornecer a corrente necessária para o funcionamento do atuador.

A prótese conta com dois módulos de circuito impresso: um para ajuste de tensão de alimentação, e outro contendo o microcontrolador empregado, um Teensy 3.0, juntamente com os demais componentes necessários à aplicação do sinal de controle ao atuador. O acoplamento entre ambos se dá por meio de conectores padrão fêmea. O módulo de energização conta com duas entradas de baterias: uma 9V e outra de 12V, acopladas cada uma a reguladores de tensão de 5V e 8V, respectivamente. O nível de tensão de 5 V é utilizado para alimentação da placa Teensy e do encoder óptico, já o nível de 8 V alimenta o *driver* de corrente citado previamente. Os demais componentes são alimentados pelo sinal de tensão de 3,3 V fornecidos pelo próprio microcontrolador através de uma de suas portas.

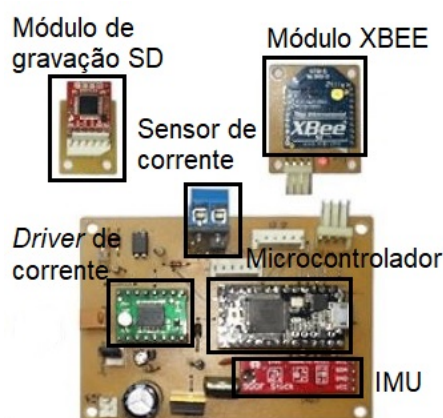


Figura 3.1: Circuito embarcado da prótese RLEG. Fonte: Adaptado de [Rocha, 2015].

A figura 3.1 apresenta o circuito embarcado da prótese contendo a unidade de controle Teensy 3.0, o *driver* de corrente responsável por fornecer a potência necessária à atuação magneto-reológica, a unidade de medição inercial (IMU) e um sensor de medição de corrente, assim como demais componentes para integração entre os módulos citados. Os módulos de gravação de memória SD e de comunicação wireless XBEE estão explicitados acima, em separado. Tais componentes conectam-se à parte mais superior de uma estrutura retangular construída, encontrando-se fora da placa principal de controle apresentada na imagem. O módulo regulador de tensão não se encontra explicitado na figura. O esquemático em PCB do circuito final desenvolvido até então encontra-se na seção de anexos. Neste, pode-se verificar o módulo de regulagem de tensão e também as conexões existentes no circuito total de controle RLEG em mais detalhes.

A Figura 3.2 explicita o conjunto prótese e circuito-embarcado. Nesta figura está presente o encoder capacitivo, posicionado na altura da junta do joelho. O encoder conecta-se ao microcontrolador para leitura de dados e o pistão, disposto na parte interna da prótese, dentro de uma estrutura de ligação de quatro barras, conecta-se à saída do *driver* de potência da placa PCB.



Figura 3.2: Sistema RLEG com circuito embarcado acoplado. Fonte: Adaptado de [Rocha, 2015].

## 3.2 Componentes eletrônicos periféricos

Nesta seção serão descritos o funcionamento e especificações de cada componente individual presente no sistema RLEG. São estes uma Unidade de Medição Inercial (IMU) de 9 graus de liberdade de fabricante Sparkfun, o Encoder AMT20, o pistão magneto-reológico LORD RD-8040, os módulos utilizados para visualização de dados adquiridos Openlog Data Logger e XBEE, assim como o *driver* PWM de corrente TB6612FNG e um sensor de corrente ACS712.

### 3.2.1 Unidade de Medição Inercial - IMU

O módulo adquirido para medição de unidades inerciais consiste no *Sparkfun 9 Degrees of Freedom - Sensor Stick*, sendo este composto por um acelerômetro ADXL345, um magnetômetro HMC5883L, e um giroscópio ITG-3200, de acordo com a figura 3.3. Os 9 graus de liberdade se devem às leituras em 3 eixos, X, Y e Z, de cada um dos 3 sensores. Cada um dos sensores individuais utilizados neste projeto tem suas próprias especificações que serão detalhadas. A interface de comunicação do dispositivo é a *Inter-Integrated Circuit*, mais conhecida como I2C.

Da folha de especificação do sensor de aceleração ADXL345, tem-se que este opera em um nível de alimentação de 2 a 3,6 V, com nível lógico alto que varia desde 1,7 V até o limite superior de tensão de alimentação provida. Seu consumo energético é extremamente baixo, com corrente de  $40\mu A$  quando operando em modo de medição de dados. Se configurado para medições entre  $-2g$  e  $2g$ , tal qual utilizado no sistema RLEG, possui uma sensibilidade de 256 LSB/g, ou seja, é capaz de medir variações de  $4mg$ , correspondendo a menos de  $1^\circ$  de inclinação. O sensor opera com até 13 bits de resolução, sendo a resolução padrão, de 10 bits, a adotada para o intervalo de leitura no sistema RLEG. A calibração do sensor se encontra implementada a nível de software nas rotinas utilizadas para a RLEG; portanto, sua unidade de medida no sistema já corresponde

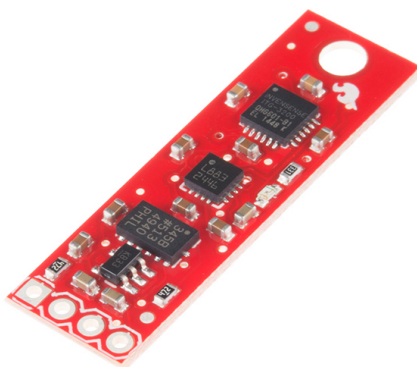


Figura 3.3: Módulo de Medição Inercial Sparkfun com 9 graus de liberdade. Fonte: Sparkfun.

à de gravidades, ou  $g$ .

O giroscópio, responsável por informar a velocidade angular do dispositivo, pode ser alimentado com tensões que vão de 2,1 a 3,6 V. O pino para definição de nível lógico é separado da tensão de alimentação, e pode ir de 1,71 V à tensão que alimenta este subcircuito. Adicionalmente, conta com conversores digitais-analógico de 16 bits, e possui seleção de filtros passa-baixa, eliminando a necessidade de implementação de uma filtragem por *software*. A sensibilidade do sensor é de 14,375 LSB por  $^{\circ}/s$ , operando nas faixas de  $-2000^{\circ}/s$  a  $+2000^{\circ}/s$ . Ainda, a corrente consumida quando em modo standby é de  $5\mu A$  e durante medições é da ordem de 6mA. A folha de especificação detalha ainda a configuração de registradores utilizados para correto estabelecimento da comunicação I2C com o sensor, de forma semelhante ao verificado para o acelerômetro. A calibração deste sensor é implementada por meio das rotinas de inicialização da IMU com códigos C++. Um zeramento desde sensor se faz necessário, já que deve indicar leitura igual a zero para todos os eixos quando não há movimentação angular.

O sensor magnetômetro não é utilizado neste trabalho, portanto, não será descrito em maior detalhes. Devido ao acoplamento do acelerômetro ADXL345 e giroscópio ITG-3200 em uma mesma interface, ambos são alimentados com a tensão de 3,3 V proveniente do chip Teensy 3.0.

### 3.2.2 Encoder Absoluto AMT20

O encoder AMT20 é essencial para a modelagem do sistema RLEG, uma vez que é utilizado para medição do ângulo da junta do joelho. Através desta medição, pode-se estimar subfases do ciclo de marcha executado. Para a implementação mecânica da prótese, a amplitude da medição desta grandeza é de aproximadamente 90 graus, que vai desde a perna totalmente estendida, correspondendo a uma medida nula nesta etapa, até que se dobre o joelho à máxima posição mecânica permitida. O encoder absoluto e capacitivo possui até 12 bits de resolução, portanto, capaz de identificar até 4096 posições diferentes possíveis. O módulo se comunica com o chip de controle por meio do protocolo de comunicação SPI. Opera sob uma grande amplitude de temperaturas ( $-40^{\circ}$  a  $125^{\circ}$ ), e com tensão de alimentação na faixa de 4,5 a 5,5 V. Devido ao nível

de alimentação requerido, utiliza-se uma bateria de 9 V regulada à tensão necessária de 5 V. A figura 6.7 explicita o dispositivo empregado.



Figura 3.4: Encoder Absoluto capacitivo AMT20. Fonte: *CUI INC*.

O zeramento do sensor ocorre por meio da interface SPI, que é implementado via programação C++ nas rotinas de leitura de dados, ou ainda por meio do software AMT ViewPoint (informação do fabricante). O consumo de energia é baixo, da ordem de 10 mA. Por ser um encoder capacitivo, este é imune a partículas de poeira, óleo, dentre outros contaminantes. O encoder pode ser acoplado ao circuito através de um conector Samtec ISDF-07-D de 14 pinos, sendo apenas 9 deles utilizados. Neste trabalho, os sinais de 6 pinos são transmitidos ao controlador, sendo estes: CSB/CS, MISO, MOSI, SCK, +5V e GND. A figura 3.5 explicita as conexões encontradas para o AMT20. Os pinos utilizados correspondem à interface SPI de comunicação.

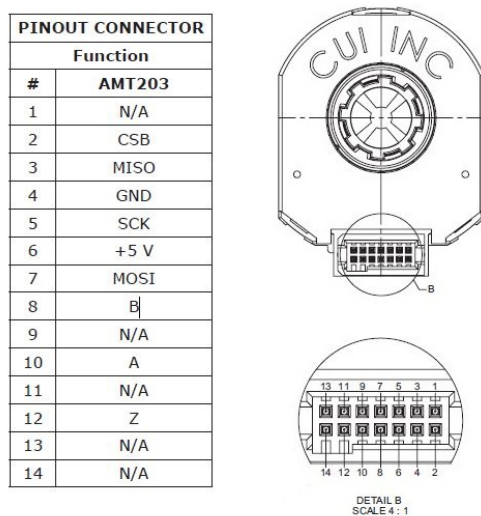


Figura 3.5: Diagrama com interface de conexão do Encoder Absoluto capacitivo AMT20. Informações do fabricante, *CUI INC*.

### 3.2.3 Pistão Magneto-Reológico

Visto que o objetivo principal da construção de uma prótese automática é proporcionar um movimento mais suave e confortável ao paciente que a utiliza, o dispositivo utilizado que atribui tais características ao sistema desenvolvido consiste em um pistão de fluido magneto-reológico. Um fluido deste tipo – *MR fluid* – consiste em meio fluido no qual estão inseridas partículas



metálicas magnetizáveis. Portanto, quando ocorre a geração de um campo elétrico sobre tal meio, suas partículas se alinham com o campo gerado, sendo capaz de modificar a força necessária para o escoamento do fluido, e alterando, conseqüentemente, seu coeficiente de amortecimento. O pistão LORD RD-8040-1 escolhido é capaz de alterar suas capacidades de amortecimento por meio deste mecanismo, quando da aplicação de corrente aos seus terminais de conexão. O componente em questão pode ser utilizado para aplicações industriais, biomédicas ou automobilísticas como um elemento dissipador de energia com uma rápida resposta de controle, de acordo com informações do fabricante. A figura 3.6 demonstra o dispositivo utilizado.



Figura 3.6: Pistão amortecedor magneto-reológico produzido pela LORD Coporation. Fonte: LORD.

De acordo com dados do fabricante, o pistão magneto-reológico LORD possui design mecânico semelhante a um pistão comum hidráulico, com exceção do tipo de fluido presente no interior de suas câmaras e de um circuito elétrico com enrolamentos de bobina responsáveis por gerar a ação de campo magnético desejada. Uma representação interna do dispositivo está explicitada na Figura 3.7. Pode-se notar a presença de duas regiões de acomodação de fluido, separadas pela estrutura que implementa a geração de campo magnético, contendo, em seu centro, um orifício para a passagem de material.

Ao aplicar-se uma corrente elétrica ao pistão, através dos seus terminais de alimentação, o campo magnético resultante é capaz de modificar a resistência de escoamento do fluido em questão (*yield strength*), e, portanto, a força aplicada necessária para promoção do movimento deste entre as câmaras do dispositivo empregado. A complexidade deste tipo de material e seu escoamento pode ser apontada pela sua não-homogeneidade, além de características mecânicas que se modificam com as condições impressas ao fluido. Portanto, equações matemáticas que relacionam diretamente a corrente aplicada à sua resistência ao escoamento não puderam ser encontradas em simples termos, ou em fontes do fabricante.

Um dos materiais magneto-reológicos fabricados pela companhia *LORD Corporation* é o fluido MRF-132DG [14]. Este fluido é composto por partículas magnetizáveis, de tamanho na ordem de micrômetros, inseridas em um meio fluido formados por estruturas de hidro-carbonetos para sua acomodação. O tempo de resposta apresentado para todos os fluidos produzidos pela companhia é alegado como sendo menor que 5 ms [14]. Além da presença do fluido descrito, o dispositivo amortecedor utilizado neste projeto conta com uma câmara denominada **acumulador**, contendo nitrogênio sob alta pressão (300 psi). Essa estrutura é separada do resto do meio por uma membrana flexível e adiciona uma compressibilidade ao sistema, visto que o fluido MR é incompressível. Portanto, pode-se afirmar que o componente utilizado apresenta ambas características de mola, com acúmulo de energia elástica promovida pelo deslocamento da haste do pistão, quanto de amortecedor, através da absorção e dissipação de energia pelo material magneto-reológico que se move entre as câmaras do dispositivo. O esquemático apresentado é de acordo com [15].

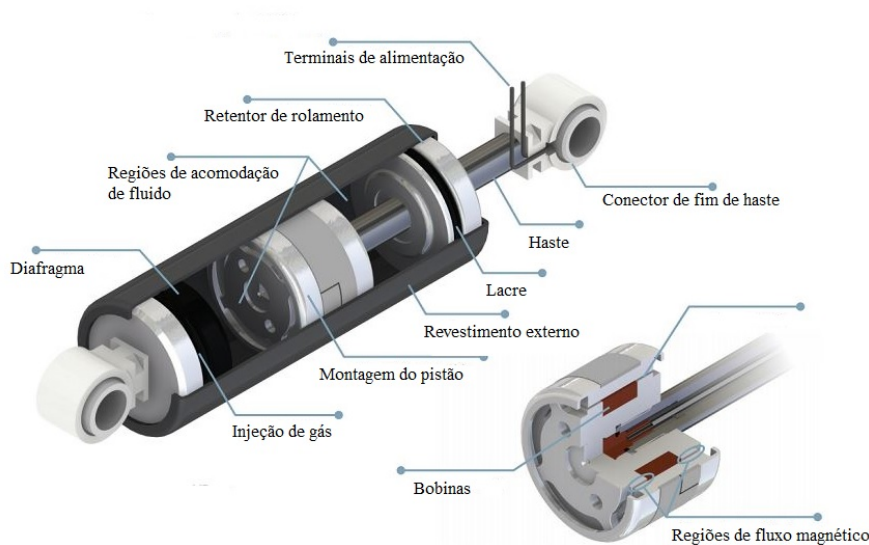


Figura 3.7: Estrutura interna do pistão MR LORD. Imagem adaptada do fabricante.

Algumas propriedades elétricas do amortecedor utilizado que podem ser destacadas são as correntes máximas permitidas. Para uma corrente contínua atuando por até 30 s, a especificação máxima é de 1 A, já para correntes aplicadas sob regime intermitentes, o valor máximo possível é de 2 A. A tensão de alimentação permitida pelo mecanismo é de até 12 V. Ainda, pode-se afirmar que sob temperatura ambiente o pistão comporta-se como uma carga resistiva de 5  $\Omega$ . Uma resposta do sistema a uma variação da corrente aplicada deve ocorrer em até 15 ms, segundo especificação. O aumento do tempo de resposta, em comparação ao tempo de resposta do fluido MR apenas, pode ser interpretado como o tempo adicional em decorrência da resposta do circuito elétrico embutido ao pistão. A tabela 3.1 descreve as principais características técnicas do componente LORD RD-8040-1.

Pode-se ainda avaliar a constante de amortecimento do pistão através do seu gráfico de **Força x Velocidade** para diferentes correntes aplicadas ao mecanismo. O gráfico da figura 3.8 explicita essa relação para valores de velocidade de até 20 cm/s, em movimentos de expansão e compressão de fluido, ou seja, velocidades positivas e negativas, respectivamente. A constante de

| Características elétricas típicas - LORD RD-8040-1 |            |
|--|------------|
| Corrente máxima de entrada - contínua              | 1 A (max)  |
| Corrente máxima de entrada - intermitente          | 2 A (max)  |
| Tensão de alimentação                              | 12 V (max) |
| Resistência elétrica - temperatura ambiente        | 5 $\Omega$ |

Tabela 3.1: Características elétricas típicas do pistão magneto-reológico LORD RD-8040-1.

amortecimento é dada pela inclinação das retas apresentadas.

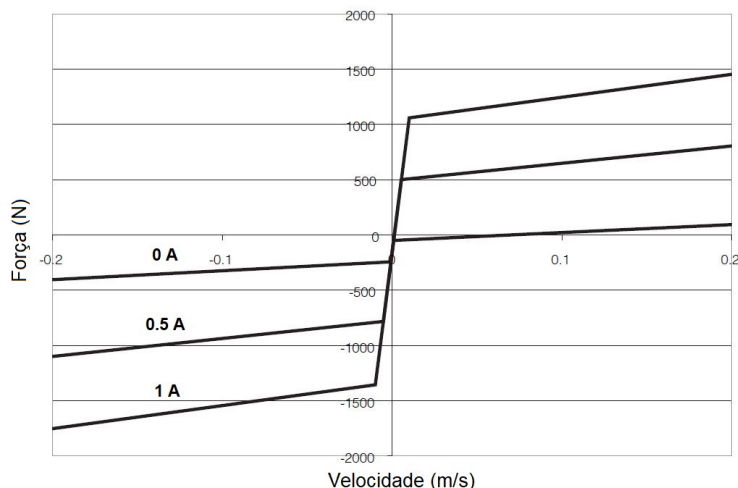


Figura 3.8: Gráfico Força x Velocidade do sistema LORD RD-8040-1 para diferentes correntes aplicadas. Fonte: Adaptado do fabricante.

### 3.2.4 *Driver* de Corrente/Ponte H

Para o fornecimento da corrente necessária à atuação do pistão magneto-reológico, utiliza-se o *driver* de corrente TB6612FNG. Este componente é comumente utilizado para realizar o controle de rotação de motores DC através de uma ponte H, além de fornecer a potência necessária ao movimento do motor empregado.

No escopo deste projeto, o circuito de ponte H é utilizado para converter um sinal PWM fornecido pelo microcontrolador em um sinal de mesma frequência, porém com amplitude igual à amplitude de alimentação  $VMOT$ , ou seja, à tensão de alimentação do motor/atuador. A figura 3.9 explicita o componente e suas pinagens correspondentes. Um maior detalhamento sobre as conexões realizadas pode ser encontrado na seção de Anexos, no arquivo de representação em esquemático do circuito total de controle e atuação.

Algumas características elétricas a se destacar sobre o dispositivo TB6612FNG são sua tensão de alimentação de canais, tensão de nível lógico, máximas correntes de saída e máxima frequência PWM permitida. Ainda, pode-se adicionar ao elemento um dissipador de calor para melhoria

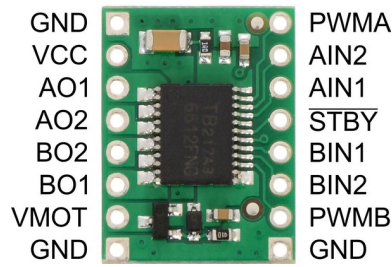


Figura 3.9: Módulo Driver de Motor TB6612FNG, da fabricante Pololu. Fonte: Pololu.

de performance sob a condição de altas correntes de saída. De acordo com especificação do microcontrolador Teensy 3.0 [16], o sinal PWM padrão fornecido é de 488,28 Hz, possibilitando a utilização do *driver* de corrente com segurança para esta aplicação. A tabela 3.2 ilustra os dados de interesse do componente.

| Características elétricas típicas - TB6612FNG       |              |
|---|--------------|
| Voltagem de alimentação do motor - contínua         | 4,5 a 13,5 V |
| Voltagem lógica                                     | 2,7 a 5,5 V  |
| Máxima corrente de saída (por canal) - intermitente | 3 A          |
| Máxima corrente de saída (por canal) - contínua     | 1 A          |
| Máxima frequência PWM                               | 100 kHz      |

Tabela 3.2: Características elétricas e mecânicas típicas do *driver* de motor TB6612FNG.

### 3.2.5 Aquisição de Dados - Cartão SD e módulo XBEE

Para que se possa realizar o monitoramento de dados de forma remota, ou seja, sem uma conexão direta do microprocessador programável com o computador, foram selecionados dois módulos que permitem o *logging* dos dados provenientes dos sensores existentes no circuito RLEG: o *Sparkfun Openlog Data Logger* e XBEE.

#### 3.2.5.1 *Sparkfun Openlog Data Logger*

Para o gravamento de dados em um cartão micro-SD, utiliza-se o módulo *Sparkfun Openlog Data Logger*, que se comunica com o microprocessador Teensy por meio de portas Serial UART. O módulo possui um microcontrolador ATmega328 pré-programado de 16 Hz, e provê uma interface de simples uso para troca de dados, se comparado a outros dispositivos semelhantes. O cartão SD de escolha deve ter formatação FAT16 ou FAT32 de até 64 gigabytes, e recomenda-se alimentação na faixa de 3,3 a 5 V. As portas RX e TX presentes no módulo trabalham com nível lógico de 3,3 V. A taxa de comunicação *baud rate* pode ser de até 115200 bps. A empresa *Sparkfun*, idealizadora do dispositivo, oferece uma ampla documentação sobre seu funcionamento, com códigos de *benchmarking* disponibilizados, tutoriais de conexão do dispositivo com outros módulos, além de modos de operação.

O dispositivo pode operar no modo automático, quando dados transferidos são gravados no cartão SD, ou em modo de comando, que permite a configuração de certos parâmetros de escrita, leitura e criação de arquivos. Alguns comandos para a manipulação dos arquivos encontram-se na tabela 3.3.

| Comando                               | Descrição   |
|---------------------------------------|---|
| new FileName                          | Cria um novo arquivo de nome <i>FileName</i> em um formato 8.3 <sup>1</sup> .   |
| append FileName                       | Acrescenta dados ao fim de um arquivo de nome <i>FileName</i> . Caso seja inexistente, um novo arquivo é criado.  |
| write FileName<br>OFFSET              | Escreve no arquivo de nome <i>FileName</i> na localização definida por OFFSET.  |
| rm FileName                           | Deleta o arquivo de nome <i>FileName</i> do diretório atual.  |
| read FileName<br>START<br>LENGTH TYPE | Lê do arquivo <i>FileName</i> a partir da posição START por um tamanho LENGTH. o parâmetro TYPE define o tipo de output, que pode ser ASCII (1), hexadecimal(2), ou <i>raw</i> (3). |

Tabela 3.3: Operações básicas de manipulação de arquivos no modo de Comando do *Sparkfun Data-Logger*.

Além da manipulação de arquivos, o modo de comando também permite operações sobre diretórios contidos no cartão de memória, de forma semelhante a comandos de *prompt* de sistemas operacionais conhecidos. As operações previstas abrangem navegação pela árvore, criação, listagem e deleção de caminhos.

Para correto funcionamento do módulo de gravação de dados, um arquivo de configuração deve estar presente no diretório raiz, com seus devidos parâmetros. Através deste arquivo pode-se definir a taxa de transmissão de dados, caractere de escape e o número necessário destes para sair do modo de comando, configurações sobre mensagens de erro, dentre outros. Caso um arquivo de configuração seja inexistente, o módulo gravará um novo arquivo com as configurações previamente definidas.

Ainda, é possível que se verifique o funcionamento do cartão por meio de luzes LED indicadoras. O LED de status **azul** traz informações acerca do funcionamento da comunicação Serial. Caso o dispositivo falhe em sua inicialização, este piscará três vezes consecutivas. Em uma situação de inicialização correta, piscará a cada novo caractere recebido durante a transmissão de dados. O segundo LED, **verde**, é responsável por indicar o status da comunicação SPI que o microprocessador utiliza para gravar informações. Dessa forma, o LED piscará para cada 512 bytes de dados transmitidos por interface SPI. As informações pré-citadas são de acordo com a documentação do dispositivo, encontrada em [17].

<sup>1</sup>Padrão em que o nome do arquivo é formado por 8 caracteres, seguidos de um “.” e de 3 caracteres de extensão do arquivo.

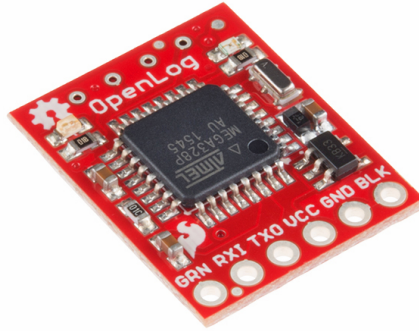


Figura 3.10: Módulo *Sparkfun Openlog Data Logger* para comunicação com cartão SD. Fonte: Sparkfun.

### 3.2.5.2 Módulo de comunicação wireless XBEE

Para uma comunicação sem fio e em tempo real, o módulo XBEE, de fabricante DIGI, foi utilizado no sistema da prótese, quando da sua concepção inicial. Para estabelecimento de uma comunicação, deve-se possuir dois dispositivos XBEE, sendo um deles configurado como Coordenador (*Coordinator*) da rede de comunicação, enquanto o segundo dispositivo deve ser acionado em modo Roteador (*Router*). Para ajuste dos modos de operação, utiliza-se o software XCTU, de propriedade da empresa criadora dos módulos. No escopo deste trabalho de verificação do funcionamento da prótese RLEG como encontrada, não foi possível uma avaliação sobre este tipo de comunicação wireless. Isso se deve à necessidade de um segundo XBEE, para conexão com o computador, o qual não foi possível de se obter no período de tempo da realização deste estudo. A forma de transmissão de dados ao XBEE acoplado ao circuito RLEG se dá por meio de uma linha serial padrão na programação empregada C++/Arduino. A Figura 3.11 ilustra o dispositivo.

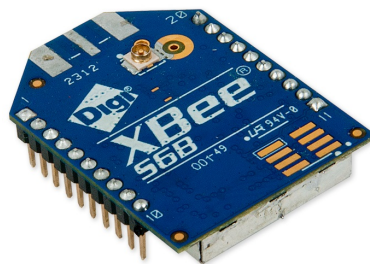


Figura 3.11: Módulo de comunicação wireless XBEE. Fonte: Digi Internacional.

### 3.2.6 Módulo sensor de corrente ACS712

Um módulo ACS712 é utilizado para medição da corrente de entrada ao pistão atuador. Este componente, em sua versão de medição em um intervalo de -5 a 5 A de corrente, possui um ganho de 185 mV/A em sua saída analógica. Dessa forma, sua tensão de saída altera-se conforme a

intensidade e sentido da corrente aplicada em relação à tensão de *offset* do dispositivo. Pode-se destacar que sua alimentação padrão é de 5 V, com uma frequência de corte do seu filtro passa-baixas interno igual a 80 KHz. A tabela 3.4 demonstra as principais características do dispositivo.

| <b>Características elétricas típicas - ACS712</b> |          |
|---|----------|
| Intervalo de medição                              | -5 a 5 A |
| Sensibilidade                                     | 185 mV/A |
| Tensão de alimentação                             | 5 V      |
| Frequência de corte                               | 80 kHz   |
| Máximo erro de medição                            | 1,5%     |

Tabela 3.4: Características elétricas típicas do sensor de corrente ACS712.

### 3.2.7 Protocolos de comunicação de módulos utilizados

Os diferentes dispositivos empregados na arquitetura de *hardware* do protótipo RLEG utilizam-se de formas distintas de comunicação com o módulo principal, o microprocessador Teensy. Cada um destes contém especificações e normas, quantidade de trilhas/barramentos necessários à comunicação, assim como bibliotecas Arduino próprias, que implementam os parâmetros necessários à sua operação. Dessa forma, serão explicitados aqui os três protocolos de comunicação identificados no sistema, para os dispositivos previamente apresentados: Serial UART, SPI e I2C. Uma breve documentação sobre cada um destes mecanismos de transmissão de dados pode ser encontrada no site oficial Arduino – [18], [19] e [20].

#### 3.2.7.1 Protocolo Serial UART

O protocolo de comunicação conhecido como UART, uma abreviação para *Universal Asynchronous Receiver/Transmitter*, é um dos mais simples e utilizados meios de comunicação serial. Por ser um protocolo assíncrono, este conta com apenas duas linhas de transmissão para envio e recebimento de dados, não havendo a necessidade de um canal dedicado a uma sincronização externa de relógio.

A comunicação Serial UART é utilizada pelos dispositivos conectados a portas seriais RX e TX. Dessa forma, são necessários dois pinos para troca de informações, sendo o RX o canal pelo qual as informações chegam, enquanto o pino TX configura-se como o canal de envio de dados.

O protocolo utiliza-se do módulo UART, um serializador de dados, para transmitir informações que são originalmente paralelas [21]. Alguns dos mais importantes parâmetros de configuração de comunicação Serial constituem-se na taxa de transmissão de dados (*baud-rate*), extremidade (*endianness*) e tamanho da janela de dados, juntamente com o tamanho do pacote após adição de bits de sinalização (início e fim) e de um bit opcional de paridade [22]. A comunicação Serial permite, neste projeto, a comunicação com o gravador de cartão SD e o módulo de transmissão *wireless* XBEE.

### 3.2.7.2 *Serial Peripheral Interface - SPI*

Dentre os protocolos síncronos de comunicação utilizados, estão o SPI e I2C.

O *Serial Peripheral Interface*, SPI, utiliza-se de 4 linhas para o estabelecimento da comunicação em arquitetura Mestre-Escravo, sendo estas: MISO, MOSI, SCK e SS/CS. O barramento conhecido por MISO é uma abreviação para *Master In Slave Out*, portanto, sendo este o canal por onde uma informação deixa o dispositivo sendo controlado, conhecido como o escravo, e chega ao mestre (neste caso, o microprocessador Teensy). A linha de dados MOSI, *Master Out Slave In*, representa, simetricamente, o caminho inverso. Em realidade, em uma comunicação SPI, ocorre o *shifting* de dados em ambas as direções a cada nova transmissão, não importando se o dado de interesse será lido pelo mestre ou pelo escravo.

A linha para sincronização de dados com o relógio externo é a *Serial Clock* – SCK, enquanto o pino SS/CS, *Slave Select ou Chip Select*, funciona como um chaveador de transmissão para cada dispositivo escravo. Quando em nível lógico baixo, um dispositivo periférico pode realizar a transmissão de dados ao mestre.

Alguns dos parâmetros que devem ser configurados quando do uso da biblioteca SPI são: máxima velocidade SPI, extremidade (*endianness*), polaridade (CPOL), este determinando o estado lógico de *clock* ocioso, e fase (CPHA) do CLOCK. O conjunto CPOL e CPHA determinam um modo SPI de operação. A tabela 3.5 explicita tais modos de acordo com as definições de polaridade e fase.

| Modo de Operação | Polaridade | Fase | Borda de saída | Borda de captura |
|------------------|------------|------|----------------|------------------|
| SPI Modo 0       | 0          | 0    | Descida        | Subida           |
| SPI Modo 1       | 0          | 1    | Subida         | Descida          |
| SPI Modo 2       | 1          | 0    | Subida         | Descida          |
| SPI Modo 3       | 1          | 1    | Descida        | Subida           |

Tabela 3.5: Modos de transmissão e captura de dados SPI. Fonte: Adaptado da documentação SPI - Arduino.

De acordo com a especificação do encoder AMT-20 utilizado, e sendo este o único dispositivo que comunica-se por SPI, pode-se verificar que este opera em SPI modo 0. Portanto, dados são expelidos pela borda de descida do relógio, enquanto que novos dados são adquiridos na borda de subida deste. Para cada quadro de dados, múltiplos de 8 bits são transferidos. A taxa de transmissão de dados pertimida é de até 1 Mhz, e dados são tranferidos a partir do *Most Significant Bit* – MSB.

Como o protocolo SPI permite implementações diversas, pode-se verificar nas especificações do encoder AMT20 as particularidades de uso da comunicação com este dispositivo. Em software, utiliza-se uma biblioteca própria *SPI.h* para realização da interface necessária ao acesso de dados.



### 3.2.7.3 *Inter-Integrated Circuit - I2C*

O protocolo *Inter-Integrated Circuit*, mais conhecido como I2C, é utilizado para comunicação entre dispositivos em um modelo Mestre-Escravo. Nesta composição, um ou mais elementos em modo Escravo dividem o mesmo barramento de transmissão de dados (*Serial Data - SDA*), assim como a mesma sincronização de relógio (*Serial Clock - SCL*). Portanto, o protocolo proporciona uma forma simples de comunicação de múltiplos dispositivos, sendo cada mensagem enviada com a informação de endereçamento correspondente. No projeto RLEG, esta arquitetura é utilizada para comunicação com a unidade inercial de medição, portanto, com o acelerômetro e o giroscópio presentes no sensor IMU.

A implementação em software da comunicação I2C no microcontrolador utilizado é desempenhada pela biblioteca *Wire.h*. Dessa forma, são construídos os comandos necessários ao endereçamento do dispositivo com o qual deseja-se comunicar, além da configuração do chip de controle Teensy como Mestre, seguindo a definição do protocolo. Adicionalmente, no circuito PCB construído, são presentes resistores *pull-up* acoplados às linhas de SDA e SCL, responsáveis por recuperar o nível de tensão alto quando não há transmissão por nenhum dispositivo.

## 3.3 Interface de controle: microcontrolador Teensy 3.0 e códigos implementados para testes em pacientes

Esta seção descreve o microcontrolador empregado na arquitetura da prótese RLEG, além das rotinas de controle encontradas e implementadas em linguagem C++ (Arduino). Quanto ao microcontrolador, são explicitadas algumas características do chip utilizado e os procedimentos adotados para correto carregamento de programas através do software Arduino. Por fim, a ideia principal dos programas de controle encontrados é apresentada, porém, sem conhecimentos numéricos claros dos resultados obtidos quando da execução de testes com a participação de um sujeito em condição de amputação transfemoral. Dados acerca da performance das rotinas construídas podem, conseqüentemente, ser estudadas mais a fundo em trabalhos futuros.

### 3.3.1 Teensy 3.0

O sistema RLEG construído foi implementado sobre o microcontrolador Teensy, em sua versão 3.0, e contando com um processador MK20DX128 de 32 bits e 48 MHz. O dispositivo possui 34 pinos de *input/output*, operando com 3,3 V. A tabela 3.6 explicita os dados de interesse deste componente [23]. Pode-se notar que portas de comunicação UART, I2C e SPI são presentes. A figura 3.12 demonstra o dispositivo.

O microcontrolador pode ser programado por meio de conexão USB mediante a utilização do próprio software Arduino, combinado a uma interface denominada *Teensyduino*, que adiciona as funcionalidades necessárias à utilização do Teensy. Alternativamente, pode-se utilizar a ferramenta denominada *Teensy Loader* para verificação do correto funcionamento da plataforma de forma

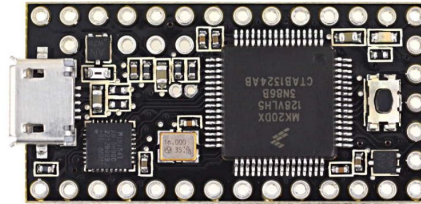


Figura 3.12: Teensy 3.0, visão superior. Fonte: PJRC

| TEMicrocontrolador Teensy 3.0 - Características |                              |
|---|------------------------------|
| Processador                                     | MK20DX128 de 32 bits, 48 MHz |
| Memória Flash                                   | 131072 <i>bytes</i>          |
| Memória RAM                                     | 16384 <i>bytes</i>           |
| EEPROM  | 2048 <i>bytes</i>            |
| Número de IOs                                   | 34                           |
| Entradas analógicas                             | 14                           |
| Portas PWM                                      | 10                           |
| Portas UART, I2C e SPI                          | 3,1,1                        |

Tabela 3.6: Características Eletrônicas - Teensy 3.0. Fonte: Adaptado da documentação PJRC, *Teensy USB Development Board*.

simplificada e realização do upload de códigos. As figuras 3.13 e 3.14 explicitam as ferramentas citadas.

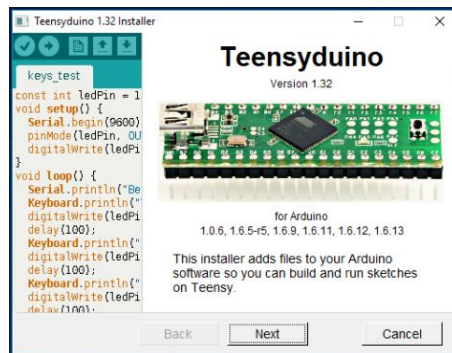


Figura 3.13: Extensão de Arduino para suporte ao microcontrolador Teensy.

Os códigos em C++ desenvolvidos para o sistema RLEG, portanto, necessitam do suporte ao microcontrolador utilizado. Por meio da interface Arduino pode-se realizar a compilação dos códigos e o *upload* ao Teensy. Os códigos que se encontraram desenvolvidos contam com uma organização de *scripts* de controle e verificação de sensores e cartão SD, um diretório para bibliotecas utilitárias, que contêm a implementação de interface para comunicação e calibração de sensores, e um diretório contendo arquivos salvos e *scripts* de tratamentos de dados (Matlab). Os códigos contendo rotinas de controle são descritos brevemente, alguns contendo maior importância que outros em avaliações experimentais.

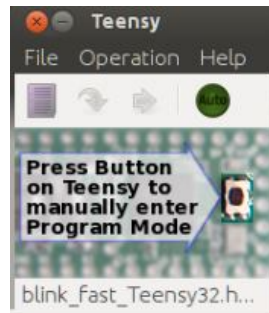


Figura 3.14: Carregador de programas Teensy Loader.

Primeiramente, para verificação do funcionamento da prótese como encontrada, realizou-se a leitura dos sensores embarcados ao sistema. Portanto, através da execução do *script* de leitura denominado *sensor\_print.ino*, pode-se observar a saída de dados da IMU, assim como do encoder absoluto AMT20. O programa não realiza o zeramento dos sensores (giroscópio e encoder) de forma automática, sendo sempre necessário que se envie um comando de zeramento por uma linha de comunicação Serial (XBEE, USB).

O programa permite ainda uma seleção de modos de operação, os quais são: modo de acelerômetro, modo de giroscópio, modo de comunicação Serial USB, modo de comunicação Serial SD e modo de comunicação Serial XBEE. Os dois primeiros modos definem a configuração aplicada à IMU, podendo-se realizar a inicialização e leitura de dados de ambos acelerômetro e giroscópio de forma simultânea, ou de apenas um dos dois. Os três últimos modos definem a forma de obtenção dos dados medidos. Os três modos podem também ser utilizados de forma simultânea, visto que portas seriais distintas são utilizadas para cada componente. Para a rotina de verificação realizada, não foi utilizado o modo de comunicação XBEE.

Ainda, tem-se que para a checagem da leitura correta de dados da IMU e do encoder, pode-se utilizar o programa *zero\_and\_start.ino*. Este define um período de amostragem de 10 ms para leitura de dados, realizando o zeramento dos sensores ao ser inicializado, e aguardando um comando de início de execução da rotina. A coleta de dados com intervalos de tempo regulares é fundamental para realização de uma rotina de controle efetivo, com uma discretização constante. Por isso, este *script* implementa um avanço na constituição de uma arquitetura confiável de leitura de sensores. A interrupção de software necessária à chamada programada à função desejada é implementada por meio da biblioteca *IntervalTimer* [24], prevista como uma ferramenta do microcontrolador Teensy.

Para a realização de testes em pacientes, utiliza-se a rotina *thres\_control.ino*. A sua execução visa o controle do pistão magnetoreológico por meio da aplicação de uma tensão efetiva proporcional a um sinal de controle enviado a um pino de saída PWM. O controle é acionado caso a velocidade angular, calculada por meio de leituras de ângulo provenientes do AMT20, atinja um valor acima do limiar pré-determinado. Caso a velocidade seja menor que tal limiar, o paciente caminha sobre a perna em modo passivo, com nenhuma corrente fluindo sobre o atuador MR.

De acordo com o reportado sobre a utilização deste algoritmo no sistema RLEG com um

paciente voluntário, a tensão injetada sobre o pistão, responsável por aumentar a resistência ao movimento realizado, causava um estado de chaveamento entre uma atuação presente e o desligamento desta. Portanto, testes realizados sobre o uso deste procedimento não foram bem sucedidos.

Outras rotinas de controle que podem ser encontradas consistem em máquinas de estado finito. Uma destas trata de dois estados: *BEFORE\_SWING*, que define toda a fase de **apoio**, e uma segunda, denominada *ON\_SWING*, determinando a fase de **balanço**. A segunda rotina realiza uma estimação do período de um ciclo completo de marcha, e, dessa forma, imprime um nível de atuação distinto de acordo com a fase atual, sendo estas iguais aos eventos definidos como *load-response* e *toe-off*.

## Capítulo 4

# Adaptação do sistema RLEG a um microcontrolador de baixo custo

Este capítulo apresenta o microcontrolador de baixo custo adotado para utilização em um novo circuito embarcado do sistema automático existente. A justificativa para o uso de tal consiste na necessidade de elaboração de uma nova placa de controle da perna de prótese em meio à dani-ficação do microcontrolador anterior, aliado ao alto custo de importação deste. Adicionalmente, modificações realizadas no funcionamento do *hardware* puderam ser realizadas, de forma a elimi-nar possíveis falhas e tornar seu modo de atuação mais eficiente e transparente. Nesse sentido, o presente capítulo descreve as principais características e forma de utilização deste dispositivo, enquanto o capítulo seguinte realiza o detalhamento do novo circuito implementado.

### 4.0.1 O microcontrolador: descrição

Partindo-se da idealização até a implementação de um sistema mecatrônico, isto é, um sistema que envolva desde a instrumentação de sinais analógico-digitais até a atuação de alguma estrutura que realize uma trajetória ou implemente uma rotina de controle, torna-se fundamental a sele-ção de dispositivos compatíveis com a aplicação e resultados desejados. Portanto, foi utilizado, neste projeto, um microcontrolador ARM de 32 bits correspondente à placa de desenvolvimento STM32F103C8, ou, simplesmente, *Blue pill*, como é conhecida popularmente, em substituição ao antigo dispositivo empregado.

De fabricante STMicroelectronics, a placa de desenvolvimento *Blue Pill* de 32 bits conta com um processador de frequência 72 Mhz, 64 kbytes de memória flash e 20 kbytes de memória RAM. Sua faixa de operação é de 2 a 3,6 V, apresentando tolerância de 5 V em muitas de suas portas de entrada/saída digitais. Essa versatilidade contribui para a simplicidade de design do circuito empregado, dispensando a necessidade de conversores de nível lógico ou divisores de tensão nas entradas que contam com esta tolerância. O dispositivo oferece conexões com interface para comu-nicação *USART*, *SPI* e *I2C*, além de implementações de transferência de dados via padrão *USB*. Ainda, oferece uma implementação de *hardware* para comunicação direta com a memória periférica

(*Direct Memory Access, DMA*) da unidade, portanto, sendo capaz de transferir dados de memória para registradores de forma independente do gerenciamento deste recurso pelo processador ARM. Pode-se destacar sua arquitetura de multiplicação e divisão de ciclo único como uma vantagem de performance encontrada no microcontrolador conhecido como *Blue pill*. Por fim, pode-se ressaltar a existência de uma aplicação para execução de tarefas *Real Time* sobre o dispositivo, denominado FreeRTOS. A tabela 4.1 apresenta as principais características de *hardware* do MCU utilizado.

| Especificações de <i>hardware</i> - <i>Blue pill</i> |                            |
|--|----------------------------|
| Processador  | ARM M3 de 32 bits e 72 Mhz |
| Memória <i>Flash</i>                                 | 64 kbytes                  |
| Memória RAM  | 20 kbytes                  |
| Portas GPIO  | 32                         |
| Portas analógicas (ADC)                              | 10                         |
| Portas PWM   | 15                         |
| UART, I2C e SPI                                      | 3, 2, 2                    |

Tabela 4.1: Especificações de *hardware* do microcontrolador STM32F103C8.

Pode-se ainda apresentar as vantagens do uso desse microcontrolador com relação a um dispositivo da família Arduino de dimensões semelhantes, a saber, o seu modelo *Nano*. Comparando-se cada uma das características avaliadas na tabela anterior, tem-se que o chip da família STM32 ultrapassa o concorrente da Arduino em todos os quesitos explicitados. Isso evidencia o aspecto iniciante dos chips Arduino em comparação a produtos de preço semelhante no mercado: enquanto um Arduino *Nano* original custa cerca de 22 dólares no site do fabricante oficial, um microcontrolador *Teensy* apresenta valor muito semelhante. A *Blue pill*, por sua vez, tem valor cerca de 10 vezes menor que os dispositivos pré-citados, enquanto é capaz de atender a pré-requisitos de projeto de circuitos embarcados com eficiência.

Em questões de desempenho, em relação ao microcontrolador Teensy 3.0, a *Blue pill* apresenta apenas uma desvantagem real, relacionada à memória do tipo *Flash*. Porém, o novo dispositivo empregado é capaz de balancear tal diferença por possuir um clock de maior frequência de processamento – 72 Mhz em comparação a 48 Mhz do chip Teensy 3.0 –, além de contar com um maior número de canais de comunicação de padrão I2C e SPI. Portanto, avalia-se que ambos os microcontroladores são de eficiência semelhante, porém, com significativa diferença de custo de aquisição. A tabela 4.2 apresenta as características de *hardware* dos microcontroladores *Teensy* 3.0 e Arduino Nano, respectivamente, para fins de comparação.

Relativamente à sua programação, o dispositivo conhecido como *Blue pill* possui um conjunto de IDEs próprias para uso: SW4STM32, EWARM, TrueSTUDIO, dentre outras. A complexidade de seu uso torna-se um pouco maior em comparação aos demais dispositivos citados, que são programados por meio da interface Arduino, com ampla oferta de bibliotecas compatíveis às diversas funções necessárias ao funcionamento de periféricos. Em contrapartida, a utilização de um ambiente próprio de desenvolvimento, dentre os pré-citados, aliada a um uso mais transparente

de funcionalidades C/C++ promovem maior confiabilidade ao sistema, apesar de maior tempo empregado à construção de uma estrutura básica de configuração do STM.

| Parâmetro               | Arduino Nano                    | Teensy 3.0                      |
|-------------------------|---------------------------------|---------------------------------|
| Processador             | ATmega328P (8 bits, max 20 Mhz) | ARM Cortex-M4 (32 bits, 48 Mhz) |
| Memória <i>Flash</i>    | 32 kbytes                       | 128 Kbytes                      |
| Memória RAM             | 2 kbytes                        | 16 kbytes                       |
| Portas GPIO             | 22                              | 34                              |
| Portas analógicas (ADC) | 8                               | 14                              |
| Portas PWM*             | 6                               | 10                              |
| UART, I2C e SPI         | 1, 1, 1                         | 3, 1, 1                         |

Tabela 4.2: Características de *hardware* dos microcontroladores Arduino Nano e Teensy 3.0

#### 4.0.2 Ambiente de desenvolvimento

O ambiente de desenvolvimento adotado para correta configuração do dispositivo da família STM32, foi, neste trabalho, o *System Workbench for STM32 – SW4STM32*, aliado a outras duas aplicações instaladas sobre sua IDE: a biblioteca STM32CubeF1 e a ferramenta gráfica para configuração de periféricos STM32CubeMX. As próximas seções buscam explicitar o uso destas tecnologias em maiores detalhes.

O *System Workbench* consiste em uma IDE baseada em Eclipse para programação de dispositivos da família STM32. A ferramenta conta com compilador GCC e *OpenOCD* para a execução de código em modo de *debug*, permitindo, assim, a depuração de código por meio do acesso de endereços de memória da CPU do dispositivo durante a sua execução. O MCU deve ser utilizado, portanto, em conjunto a uma gravadora/depuradora STLink. Esta forma de transferência de dados se deve ao *bootloader* padrão da placa, que faz uso dos pinos de SWIO e *clock* para transferência de programas conforme padrão definido na gravadora em questão. O uso da IDE selecionada para programação do STM32 permite uma melhor compreensão e controle sobre as rotinas executadas, assim como otimização de recursos, contrapondo uma possível programação no ambiente Arduino.

Conjuntamente à utilização da IDE SW4STM32, faz-se uso da biblioteca STM32CubeF1, contendo ferramentas que contribuem para a redução dos esforços de desenvolvimento sobre o *chip* utilizado. Esta aplicação conta com uma camada denominada *Hardware Abstraction Layer*, HAL, para fácil configuração de periféricos e utilização de interfaces de comunicação sobre demais dispositivos. Esta implementação é amplamente utilizada no projeto RLEG em sua nova versão. Componentes de *middleware* também são existentes, dentre eles, o mais relevante sendo o Sistema Operacional de Tempo Real FreeRTOS. Por fim, pode-se destacar o uso de camada HAL em diversos exemplos do fabricante, embutidos na instalação da biblioteca STM32CubeF1, proporcionando, assim, maior facilidade à sua compreensão prática.

Como última ferramenta utilizada, tem-se o STM32CubeMX: uma interface de configuração visual dos periféricos internos ao *chip* STM32. Essa camada é aliada à HAL para correto funcionamento do sistema existente. Dessa forma, pode-se definir pinos de atuação UART, I2C, SPI, analógicos e entrada/saída digital. O uso de *clocks* e *timers*, velocidade e tamanhos de endereços para transferência de dados e demais características, assim como habilitação de uso da porta USB ou modo debug via gravadora STLINK ocorrem, portanto, por meio desta interface. A imagem 4.1 demonstra a aplicação de forma visual. Após devida configuração, gera-se o código de inicialização do dispositivo, através da exportação do projeto ao ambiente SW4STM32.

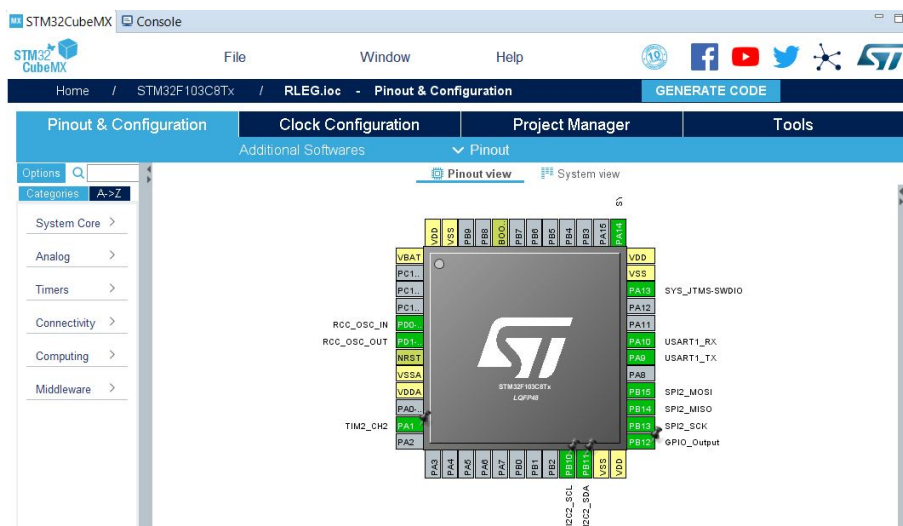


Figura 4.1: Interface STM32CubeMX para configuração visual de periféricos internos de dispositivos da família STM32.

### 4.0.3 Camada de implementação HAL para protocolos de comunicação

A camada de abstração de *hardware*, HAL, é utilizada em conjunto ao ambiente *System Workbench* para simplicidade de programação do dispositivo *Blue pill*. A API conta com diversos *drivers*, cada um destes contendo um conjunto de funções de configuração de periféricos. Exemplos de cada conjunto de funções podem ser dados com relação a configurações de *clock*, GPIO, I2C, SPI, TIM (*timers*), UART, dentre outros. As principais tarefas de comunicação necessárias à execução do projeto RLEG são implementadas por meio do uso das funções de camada HAL explicitadas na tabela 4.3. Configurações de relógio são implementadas por meio da API CubeMX de forma automática.

A aplicação da camada de abstração de *hardware* como *plug-in* para o ambiente de desenvolvimento SW4STM32 fornece utilização em três modos distintos de programação: *polling*, *interrupt* e *Direct Access Memory, DMA*. O primeiro destes métodos representa um algoritmo bloqueante, ou seja, que bloqueia o uso de outros recursos até o termino da execução da tarefa. Os demais métodos são denominados não-blocantes. Sendo assim, mesmo enquanto as tarefas não são finalizadas, os componentes de *hardware* podem ser utilizados para processamento de outras funções. Os *drivers* correspondentes são utilizados, até a execução deste trabalho, em modo de *polling*, por



facilidade de uso.

| Função de camada HAL             | Ação   |
|----------------------------------|--|
| <b>HAL_Delay</b>                 | Aguarda a passagem de tempo especificado como parâmetro.   |
| <b>HAL_GPIO_ReadPin</b>          | Leitura de pino digital.   |
| <b>HAL_GPIO_WritePin</b>         | Escrita de pino digital.   |
| <b>HAL_I2C_IsDeviceReady</b>     | Verifica se dispositivo I2C está pronto para comunicação, de acordo com o endereço especificado.         |
| <b>HAL_I2C_Master_Transmit</b>   | Realiza a transmissão de dados por meio do dispositivo mestre.   |
| <b>HAL_I2C_Mem_Read</b>          | Realiza uma leitura em um endereço especificado de memória do dispositivo escravo.                       |
| <b>HAL_SPI_TransmitReceive</b>   | Realiza uma transmissão por comunicação SPI, ao mesmo tempo em que recebe dados do buffer do periférico. |
| <b>HAL_ADC_Start</b>             | Habilita o uso do canal ADC configurado e leitura por meio de <i>polling</i> .                           |
| <b>HAL_ADC_PollForConversion</b> | Função que realiza a conversão analógico-digital de sinal.   |
| <b>HAL_ADC_GetValue</b>          | Retorna o valor de leitura ADC.  |
| <b>HAL_TIM_PWM_Start</b>         | Inicializa a geração de sinal PWM.   |

Tabela 4.3: Principais funções de comunicação - camada HAL

#### 4.0.4 FreeRTOS

Dentro da necessidade de uma aplicação com características de tempo real, faz-se indispensável o uso de algum tipo de sistema em *software* que possibilite o gerenciamento de tarefas concorrentes, em contraposição à abordagem padrão de execução de algoritmos de lógica sequencial. Em aplicações industriais e robóticas, esse paradigma de programação é essencial para a correta operação dos sistemas desejados. No caso específico do projeto RLEG, as rotinas de leituras de parâmetros de marcha devem ser executadas a intervalos regulares, além da leitura do sinal de controle e sua atuação, para que o sistema seja operacionalmente confiável. Sem o uso de rotinas que garantam uma execução programada das tarefas existentes, os tempos de leitura da variável controlada e atualização do sinal de controle se tornam estocásticos, a depender do caminho tomado ao longo do código de programação. Tal configuração pode gerar aquisições de dados e ações de controle defasadas, e, ao extremo, ocasionar em uma ação de controle que gera comportamentos instáveis.

Para corrigir isso, pode-se utilizar um Sistema Operacional de Tempo Real, RTOS, a fim

de gerenciar a execução de tarefas em intervalos regulares. As tarefas, ou rotinas, existentes atuam de forma independente, porém, sendo capazes de se comunicar através de variáveis de controle de recursos (*semaphores*) e recursos de notificação de conclusão de tarefas. Uma versão de RTOS presente no *firmware* dos chips da família STM32 consiste na aplicação denominada FreeRTOS. Essa implementação de um Sistema Operacional de Tempo Real visa aliar um conjunto de ferramentas para correto agendamento de tarefas a uma simplicidade de arquitetura, otimizando a relação desempenho por uso de memória do dispositivo.

#### 4.0.5 Descrição da geração de sinal PWM

A programação do microcontrolador utilizado por meio do ambiente SW4STM32 permite um maior controle da execução de algoritmos a nível de *hardware* do dispositivo. Essa característica é capaz de tornar o sistema mais otimizado e confiável; por outro lado, traz uma maior complexidade na configuração dos periféricos e rotinas utilizadas.

Aqui, explicitam-se os passos necessários para a geração de um sinal modulado em largura de pulso – *Pulse-Width Modulation*, PWM – com o microcontrolador da família STM32. A geração do sinal citado é essencial para a correta execução do projeto, uma vez que é sinal de entrada ao pistão atuador. Deve-se, ainda, observar que o chaveamento, ou frequência, do sinal PWM gerado deve ser rápido o suficiente para gerar uma resposta estacionária da atuação do pistão. Dessa forma, deve-se ser capaz de manter seu coeficiente de amortecimento constante para um sinal com largura de pulso fixa.

Para a geração de um sinal PWM, duas variáveis principais devem ser definidas: frequência e ciclo de trabalho. Essas duas variáveis dependem diretamente da configuração de dois registradores: ARR e CCRx, em que x representa os números de 1 a 4 (quatro canais). O registrador ARR (*Auto-Reload Register*) é responsável por armazenar a quantidade de divisões presentes na contagem de um ciclo completo de onda PWM. Dessa forma, ele controla a resolução de ajuste de sinal PWM gerado, em que a quantidade de passos de PWM se iguala ao valor de *auto-reload* + 1. Por exemplo, para um valor de 99 de ARR, um ciclo completo de onda contará com 99 + 1 divisões de contagem, ou passos de PWM. Portanto, para este valor, pode-se ajustar o valor de ciclo de trabalho de um sinal com resolução igual a uma variação unitária.

O registrador de comparação, CCRx, por sua vez, é utilizado para armazenar o valor máximo de contagem para o qual o pulso de saída ainda é ativo. De tal forma que:

$$pwm_{out} = \begin{cases} 1, & \text{se } CNT < CCRx \\ 0, & \text{caso contrário} \end{cases}, \quad (4.1)$$

sendo CNT o registrador de contagem de passo e para um *setup* de PWM em modo padrão (modo 1). Portanto, através da definição do valor do registrador de comparação, define-se o ciclo de trabalho da onda desejada. Ressalta-se, ainda, o significado do valor 1 na equação definida em 4.1: um valor de nível lógico alto de PWM, resultando, portanto, em uma saída de sinal de 3,3 V.

Um último parâmetro a ser definido neste escopo é o valor de divisão do *clock* utilizado no circuito STM32. Uma variável de *Prescaler* é então utilizada para tal finalidade, sendo responsável por dividir o clock de contagem para obtenção da frequência necessária à realização desta tarefa. O dimensionamento desta variável é feito de acordo com a equação:

$$Prescaler = (freq_{timer}/freq_{count}) - 1, \quad (4.2)$$

em que  $freq_{timer}$  representa a frequência do timer escolhido, podendo ser igual à frequência do *clock* principal, e  $freq_{count}$  sendo igual à frequência de entrada ao registrador de contagem CNT.

A frequência de contagem necessária, por sua vez, é dada por:

$$freq_{count} = freq_{pwm} \cdot (pwm_{steps}), \quad (4.3)$$

em que  $freq_{pwm}$  e  $pwm_{steps}$  correspondem à frequência desejada de saída do sinal PWM gerado e à quantidade de passos, ou resolução, de PWM, respectivamente.

Dessa forma, a partir da definição de duas variáveis desejadas à geração de sinal modulado – frequência e resolução de ciclo de trabalho –, pode-se configurar as variáveis correspondentes à frequência do contador, valor de *auto-reload* – registrador ARR – e valor de *prescaler* no ambiente STM32CubeMX. Para a execução deste trabalho, um valor de 72 kHz de frequência foi definido como padrão, além de uma resolução de ciclo de trabalho igual a 100. A tabela 4.4 explicita os parâmetros utilizados. O valor do ciclo de trabalho desejado, definido por meio do registrador de comparação CCRx, é primariamente definido como sendo igual a 20%, podendo ser modificado de acordo com a aplicação desejada.

| Parâmetro                   | Valor   |
|-----------------------------|---------|
| <i>freq_pwm</i>             | 72 kHz  |
| <i>pwm_steps</i>            | 100     |
| <i>freq_count</i>           | 7,2 Mhz |
| <i>Prescaler</i>            | 9       |
| <i>Auto-Reload Register</i> | 99      |
| <i>CCR2</i>                 | 20      |

Tabela 4.4: Parâmetros definidos para geração de sinal em modulação de largura de pulso.

## Capítulo 5

# *Redesign* do circuito embarcado do sistema RLEG

O presente capítulo apresenta as modificações implementadas sobre o sistema da prótese RLEG para adequação ao novo microcontrolador adotado. Alterações tanto a nível de *hardware* quanto de códigos de programação foram realizadas, com justificativas para a realização de tais. Uma primeira mudança diz respeito ao *redesign* de circuito para adaptação ao uso do novo microcontrolador *Blue pill*, seguindo as motivações descritas no capítulo anterior. Uma segunda melhoria está relacionada a um dos programas de atuação sobre o amortecedor LORD. Assim, foi aplicada uma condição de histerese a fim de eliminar deficiências do antigo modelo de controle implementado.

### 5.0.1 Conceitos fundamentais do novo *design* de *hardware*

Devido a mudanças no paradigma de controle por *hardware* do antigo projeto RLEG, houve necessidade de implementação e fabricação de um novo circuito lógico para atuação do pistão magneto-reológico. Dessa forma, além de substituição do microcontrolador empregado, demais modificações foram realizadas para: 1. Eliminação de possíveis falhas de circuito e 2. Adição de funcionalidades desejadas ao sistema. Tais modificações são descritas em detalhe a seguir.

A correta operação do *hardware* ocorre da seguinte forma: a leitura de sinais de marcha é realizada periodicamente através dos componentes de unidade de medição inercial e encoder. Os dados de aceleração e velocidade angular são extraídos da IMU e o ângulo de rotação do joelho, do encoder empregado. Após processamento de tais dados, um sinal de tensão é aplicado à entrada do componente magnetizável, controlando, portanto, seu coeficiente de amortecimento de forma indireta. A geração do sinal se dá por meio do microcontrolador STM32, via modulação em largura de pulso, sendo este aplicado como entrada ao módulo *driver* de corrente TB6612FNG.

O *driver* de corrente é responsável por suprir a corrente necessária à atuação do pistão do joelho, pois uma alimentação de tensão direta por meio do microcontrolador não seria capaz de suprir tal demanda. A termo de exemplificação, para o MCU utilizado, uma corrente de 150 mA é a máxima permitida. O *driver*, portanto, se torna fundamentalmente necessário nesta aplicação.

Em termos mais explícitos, o dispositivo realiza a conversão de um sinal PWM de amplitude igual a 3,3 V em um sinal modulado de maior amplitude, conforme alimentação em seu terminal VMOT (alimentação do motor/atuador). No caso do circuito empregado, esta alimentação consiste em uma bateria, sendo, portanto, o sinal de saída igual à tensão presente na bateria, nominalmente, 12 V. O dispositivo, funciona, assim, como um amplificador de sinal PWM. Ademais, na configuração existente, o componente possui dois canais de saída de onda, sendo, cada um, capaz de suprir até 1 A de corrente. Para ampliação do seu potencial de utilização, os dois canais do dispositivo foram conectados em uma ligação paralela, resultando em uma saída de sinal PWM com capacidade de suprimento de até 2 A.

A saída do módulo TB6612FNG, por sua vez, é sinal de entrada ao terminais de passagem de corrente do sensor correspondente. Um sensor de corrente ACS712, de fabricante Allegro, e com range de leitura de +/- 5 A é utilizado para avaliação do sinal de corrente que passa pelo sensor e vai de entrada ao atuador. Uma mudança realizada nesta versão da placa de controle consiste na passagem obrigatória da corrente de entrada  $I_{damper}$  por este sensor. Um circuito divisor de tensão foi adicionado à saída de sinal analógico do sensor de corrente ACS para que valores acima do esperado não causem risco ao funcionamento do microcontrolador, de tolerância 3,3 V para pinos de leitura analógica. Assim, dois resistores de valor equivalente são utilizados no circuito, dividindo, portanto, o valor analógico de saída pela metade. A utilização do sensor de corrente para identificação do parâmetro  $I_{damper}$  visa a possibilidade de implementação de uma rotina de controle desta grandeza. Ainda, inclui-se no projeto espaço para adição de um capacitor em paralelo a uma das resistências do circuito divisor de tensão empregado, configurando um filtro passa-baixas do sinal lido de corrente. O valor de capacitância desejada seria definido conforme a resposta de corrente do atuador, para que se observe apenas o sinal médio desta grandeza.

Dentre as modificações realizadas, tem-se que, no circuito de alimentação, eliminou-se o uso da bateria de 9 V para acionamento de dispositivos de baixa potência. Tal mudança se justifica pela utilização de uma única bateria de tensão nominal igual a 12 V, servindo ao propósito de alimentação do circuito como um todo, e tendo seu sinal regulado em 5 e 3,3 V para cada necessidade. O sinal de 3,3 V era previamente proveniente do microcontrolador, porém, devido à possibilidade de sobrecorrente, este *design* de circuito poderia gerar falhas, acarretando em malfuncionamento dos dispositivos. O circuito de regulação de tensões de alimentação foi, ainda, posicionado na mesma placa de controle lógico.

Além das modificações pré-citadas, pode-se apresentar a adição de um módulo *bluetooth* HC-05, para transferência de dados de interesse, e de um *display* LCD 16x2 para disposição do sinal adquirido de corrente. O circuito conta ainda com três LEDs para sinalização do modo de operação da perna, a saber: sem atuação, com atuação fixa e com atuação variável. A tabela 5.1 apresenta as principais modificações aplicadas ao novo circuito embarcado obtido.

## 5.0.2 Tecnologias utilizadas para modelagem de circuito embarcado

A fim de se obter um produto na forma de circuito impresso para a placa controladora da perna de prótese RLEG, fez-uso do *software* de desenho assistido por computador Eagle. Este visa a

| <b>Modificação</b>               | <b>Circuito antigo</b>  | <b>Novo circuito RLEG</b>                           |
|----------------------------------|---|---|
| Placa controladora               | Teensy 3.0  | <i>Blue pill</i>                                    |
| Alimentação                      | Baterias de 9 e 12 V  | Única bateria de 12 V                               |
| Regulação de tensão              | 5 e 8 V   | 3,3 e 5 V   |
| <i>Driver</i> de corrente        | Utilização de 1 canal   | Utilização de 2 canais                              |
| Saída analógica do sensor ACS712 | Entrada direta ao microcontrolador                                    | Implementação de um divisor de tensão e filtro RC   |
| Módulos adicionais               | XBEE  | <i>Bluetooth</i> e <i>display</i> LCD               |
| PCB                              | Existência de módulos de alimentação, controle, SD e XBEE em separado | Implementação do circuito PCB sobre uma única placa |

Tabela 5.1: Modificações aplicadas ao circuito embarcado RLEG.

facilidade de execução de *design* de circuitos eletrônicos, integrando os ambientes de desenvolvimento de esquemáticos do circuito e sua representação na forma de placa de circuito impresso. Ainda, permite a criação de bibliotecas personalizadas pelo autor para adição de componentes que não são disponibilizados em suas bibliotecas padrão, além de simulações de circuito mediante a importação de modelos para os dispositivos utilizados.

Dessa forma, para a execução do projeto em questão, criou-se primeiramente uma biblioteca específica para reunião dos componentes necessários ao seu correto funcionamento. O uso de uma biblioteca própria ao projeto facilita a recuperação dos passos utilizados para a produção da PCB, visando, assim, sua organização para continuidade do uso destas tecnologias, caso necessário. Através da manipulação de bibliotecas, pode-se criar representações de dispositivos na forma de esquemático e na visão de componente sobre PCB. Após criação de uma biblioteca específica, pode-se organizar o esquemático do circuito visado, sendo as conexões existentes posteriormente importadas ao projeto de placa impressa. Ambas as interfaces são conectadas de forma que modificações realizadas em esquemático são transmitidas à placa, e vice-versa. Esta característica previne a existência de erros de conexão e auxilia a revisão técnica do projeto, em uma etapa prévia à sua fabricação.

A implementação do circuito em sua forma impressa requereu a tomada de decisões de projeto conforme requisitos do sistema. Dentre os parâmetros definidos ao *layout* executado, pode-se citar a espessura das trilhas de cobre, tamanhos de *pads* para soldagem dos dispositivos e furos para encaixe, além da quantidade de camadas utilizadas. A espessura padrão das trilhas foi mantida em 1 mm, exceto por trilhas de alimentação em 12 V e trilhas que vão de encontro aos terminais do atuador, estas, definidas como espessura de 2 mm. Diversas ferramentas *online* podem ser encontradas para cálculo de espessura mínima de trilhas de fabricação PCB. Na realização do projeto, optou-se pela utilização de uma maior margem de segurança à esse parâmetro pela existência de possíveis erros de manufatura. A diferença entre os raios externo e interno de *pads* foi definida como sendo de 20 mil, aproximadamente 0,5 mm. Um tamanho bem dimensionado dessa grandeza permite uma boa aderência de solda conforme a área disponibilizada. Os furos utilizados

foram de 0,9 mm, para dispositivos padrão, e 1,0 mm para dispositivos de maiores dimensões de pinos, como os conectores utilizados para bateria e atuador.

A disposição dos componentes utilizados foi realizada de forma a otimizar o uso do espaço existente, para utilização de uma única camada de cobre (*bottom layer*). Para isso, uma conexão na forma de *jumper* foi utilizada, eliminando-se a necessidade de adição de uma segunda camada de usinagem. O aterramento da placa foi realizado por meio da execução de um plano de terra, sendo este mais confiável que a utilização de trilhas, à medida que elimina flutuações localizadas em tensão. Pode-se citar, por fim, o posicionamento estratégico dos dispositivos de alta potência – ponte H, conector para pistão magneto-reológico, sensor de corrente e elementos de alimentação – de forma a se distanciar dos demais, de menor potência empregada. Esse *layout* provê uma maior segurança ao projeto implementado. O circuito total obtido é conforme a Figura 5.1. A Figura 5.2 explicita os componentes empregados no novo circuito RLEG e seus posicionamentos em maiores detalhes.

Após a realização do circuito em formato de PCB por meio da ferramenta de CAD, geram-se os arquivos de manufatura assistida por computador. Tais arquivos podem ser gerados por meio do *software* utilizado e são de formato *gerber*, para trilhas, perfil de dispositivos (*silkscreens*) e *pads*; e *excellon*, para perfilamento de furos. A metodologia para fabricação do circuito explicitado é detalhada na seção seguinte e ocorre por meio do uso das ferramentas FlatCAM e bCNC.

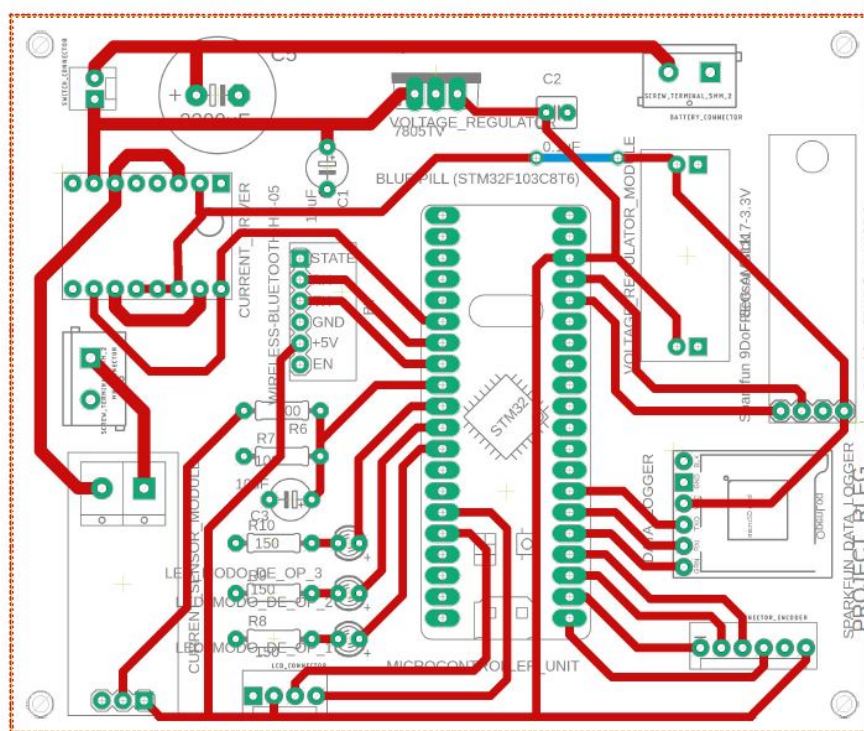


Figura 5.1: Projeto de placa de circuito impresso desenvolvido com uso da ferramenta CAD Eagle.

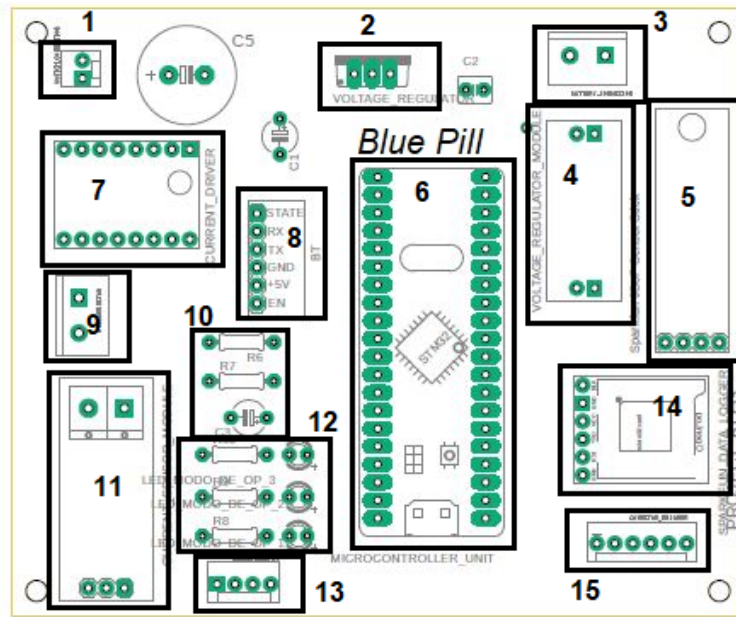


Figura 5.2: Componentes utilizados no novo circuito RLEG e seus posicionamentos. Legenda: 1. Terminais do botão liga/desliga; 2. Regulador de tensão de 5 V; 3. Conector da bateria; 4. Módulo regulador de tensão de 3,3 V; 5. IMU; 6. *Blue Pill*; 7. *Driver* de corrente; 8. Módulo *bluetooth*; 9. Conector do pistão; 10. Divisor de tensão e filtro RC; 11. Módulo ACS712; 12. LEDs indicadores de modo de operação; 13. Conector LCD; 14. Módulo de gravação SD; 15. Conector do encoder.

### 5.0.3 Metodologia de fabricação do circuito embarcado

O processo de fabricação de uma placa de circuito impresso, no espaço do laboratório LARA, foi realizado por meio do uso de ferramentas adicionais de *software* após a geração de arquivos em formato *gerber* e *excellon*, ou seja, arquivos que representam o perfilamento do plano a ser obtido como produto final. São empregadas duas ferramentas em *software* de manufatura assistida por computador - CAM, do inglês *Computer-Aided Manufacturing*. A primeira destas, de nome FlatCAM, consiste em um programa que realiza a interpretação do *design* obtido, por meio da ferramenta CAD Eagle, e o converte em código G, ou seja, código que representa a trajetória a ser executada pela máquina de usinagem. A segunda destas ferramentas, bCNC, é responsável por interpretar o código G gerado previamente e realizar o controle da máquina de usinagem utilizada, sendo esta uma fresadora e furadeira CNC.

No ambiente disponibilizado pela ferramenta FlatCAM, os arquivos de perfilamento gerados são importados, e alguns parâmetros de interesse necessários à criação do código G são definidos. Dentre tais parâmetros, tem-se o diâmetro da ferramenta utilizada, quantidade de passes e sobreposição de passe. Uma fresa de topo de 0,3 mm de diâmetro foi utilizada, com a realização de apenas um passe sobre a peça. Estas especificações geram um objeto de geometria, passo intermediário para a criação da trajetória e parâmetros finais de movimento.

Após a geração de um objeto de geometria, deve-se definir variáveis de profundidade de usi-



nagem ( $a_p$  ou *Cut Z*), plano de movimentação da ferramenta (*travel Z*), velocidade de avanço ( $v_f$  ou *feed rate*) e, novamente, diâmetro da ferramenta. Estas configurações definem a criação de um objeto de tarefa CNC, podendo-se, a partir deste, obter o código G. Devido à utilização do sistema imperial em *software*, todas as unidades foram convertidas a esse sistema. A profundidade de usinagem,  $a_p$ , foi determinada experimentalmente, devido a dificuldades encontradas na metodologia de calibragem da máquina CNC associada, e teve valor definido em 0,3 mm. O plano de movimentação da ferramenta pode ser definido em qualquer intervalo que seja seguro para a execução das trajetórias necessárias, sendo assim, utilizado em 1 polegada. A variável de velocidade de avanço foi utilizada conforme definido em projetos anteriores, por segurança.

Um processo semelhante é realizado para a geometria de furos, com a alteração de ferramenta para uma broca de 0,9 mm de diâmetro. Furos de maiores dimensões foram executados manualmente, com auxílio de uma retificadora *DREMEL*, após marcação com a furadeira CNC. Uma observação a ser feita consiste na forma de importação do arquivo *excellon* a ser utilizado. Para a não-ocorrência de erros de visualização do perfil de furos, arquivos importados em formato métrico devem ser editados para ajuste do tamanho das ferramentas definidas em projeto. Do contrário, o que se observa são furos com marcação correta de centro, porém de diâmetros incompatíveis. Isto ocorre como consequência de uma falha encontrada em *software*. A correção pode ser realizada por meio da interface de *software* utilizada ou edição do código G, por meio da modificação da dimensão da ferramenta utilizada – comando T.

Após sua obtenção, o código G é importado pelo *software* de controle numérico bCNC. Sua interface provê uma forma facilitada de conexão com o maquinário utilizado e movimentação da ponta de prova para calibragem. Os procedimentos utilizados são explicitados a seguir.

#### 5.0.4 Calibragem de máquina CNC

Para a obtenção da placa PCB como desejada, deve-se realizar a calibragem da máquina CNC, sempre após a troca de ferramenta empregada. Em um primeiro passo de calibragem, realiza-se o ajuste de zero peça com zero máquina. Dessa forma, executa-se o alinhamento da ferramenta de usinagem e o recorte de chapa metálica de cobre que dará origem ao circuito impresso em uma mesma origem. Em uma segunda etapa, deve-se ser capaz de obter o nível do plano XY do objeto utilizado, já que este pode apresentar diferentes valores em Z com relação ao eixo da ferramenta. Variações deste tipo são causadas por imperfeições no material, diferenças de nível na mesa de trabalho ou ainda variações adicionadas por adição de material para aderência do objeto à mesa de trabalho.

O *software* open-source bCNC conta com uma ferramenta própria para calibração de nível, denominada *autolevel*. Esta ação ocorre por meio da verificação do nível de diversos pontos ao longo de um retângulo que passa pelas extremidades do objeto utilizado. A rotina de checagem dos pontos pré-definidos é então executada, sendo o valor no eixo Z de cada ponto encontrado por meio da utilização da ferramenta como ponta de prova. Assim, ao encontrar-se o exato local em que a ferramenta toca o corpo de cobre, em cada ponto, o teste de continuidade tem resultado positivo, portanto, encontra-se o nível do ponto especificado. Um mesmo teste por ponta de prova pode

ser realizado na origem da mesa de trabalho – coordenadas  $(0, 0)$  do plano  $XY$  – para realizar-se o zeramento do eixo  $Z$  da máquina.

Devido a falhas observadas na rotina de auto-nivelamento da fresadora CNC, realizou-se um zeramento em  $Z$  na origem  $XY$ , apenas. A profundidade de corte aplicada à operação de fresamento foi então determinada experimentalmente, pois, por vezes, a ferramenta não adentrava o objeto em medida suficiente para realização do corte necessário. Portanto, uma forma de ajuste de tentativa e erro foi adotada, sendo o valor de 0,3 mm definido como parâmetro  $a_p$ . Observa-se, porém, que este valor é muito superior ao necessário em condições ideais, em que a espessura de cobre da placa utilizada não ultrapassa o valor de 0,1 mm. Após a realização dos passos descritos para calibragem de máquina, as operações de fresamento, furação e lixação – para a retirada de rebarbas –, os componentes foram soldados ao circuito impresso obtido, obtendo-se o produto final desejado.

## Capítulo 6

# Avaliação do funcionamento do circuito e caracterização do sistema RLEG

O capítulo em questão visa explicitar os resultados obtidos por meio de testes realizados com uso da placa controladora construída para este fim, conforme descrito no capítulo anterior. Para a verificação de seu funcionamento, foram concebidas rotinas individuais das funcionalidades do microcontrolador. Assim, realizaram-se testes para a leitura de dados analógicos, provenientes do sensor de corrente utilizado, leitura dos sensores embarcados e atuação de sinal PWM ao pistão. Portanto, são expostos os dados adquiridos por meio da leitura dos sensores de marcha embarcados, assim como os resultados experimentais obtidos no processo de caracterização da resposta do atuador magneto-reológico adotado. Adicionalmente, avalia-se a utilização de um código de controle de marcha com adição de região de histerese para atuação do pistão de fluido magnetizável.

### 6.0.1 Aquisição de dados dos sensores de medição inercial e encoder

A aquisição de dados relativos à unidade de medição inercial foi implementada com uso da camada de abstração de *hardware* HAL, por meio de sua interface de comunicação I2C. Nesse sentido, foram construídas bibliotecas em linguagem C para configuração dos dispositivos de acelerômetro ADXL345 e giroscópio ITG3200 conforme informações disponibilizadas em suas folhas de especificação. A premissa básica para a troca de informações com os sensores empregados consiste na escrita de dados em registradores, no caso de inicialização dos dispositivos, e leitura de dados provenientes de registradores reservados para tal função.

A rotina de inicialização do dispositivo ADXL345 consiste na leitura do registrador que guarda o valor de identificação do dispositivo acessado. Este valor difere do endereço do dispositivo e deve retornar um valor conforme a tabela 6.1. Cada mensagem transmitida é endereçada, na linha de comunicação I2C, ao endereço de 7 bits do acelerômetro, de valor 0x53 (hexadecimal). Este

endereço é então definido e deslocado 1 bit à esquerda, para envio de um endereço de 8 bits. Após a leitura do identificador, realiza-se a configuração do modo de operação do dispositivo, resultando na escrita de dados de registradores de controle de energia e escala de operação, no caso deste trabalho, obtendo-se valores de aceleração que variam no intervalo  $[-2g, 2g]$ . Em caso de retorno diferente do esperado, sabe-se que houve um erro de inicialização do dispositivo. Um passo anterior à leitura de dados dos instrumentos utilizados consiste na utilização de uma função em camada HAL que sinaliza a disponibilidade do equipamento para conexão.

A leitura dos dados de aceleração ocorre mediante acesso aos registradores reservados para tal função [25]. Cada eixo de medição possui duas posições de memória de 8 bits cada, dessa forma, resultando em um dado de tamanho de 16 bits. Dessa maneira, obtém-se uma leitura para a aceleração de cada eixo por meio do deslocamento do *byte* mais significativo de dados e sua concatenação com o *byte* menos significativo, e multiplicando-se o valor lido pelo inverso da sensibilidade do dispositivo, dada em LSB/g. Os dados resultam em leituras de aceleração em termos da gravidade percebida.

| Registrador | Posição ( <i>hex</i> ) | Valor ( <i>hex</i> ) | Operação |
|-------------|------------------------|----------------------|----------|
| DEVICE_ID   | 0x00                   | 0xE5                 | Leitura  |
| POWER_CTL   | 0x2D                   | 0x08                 | Escrita  |
| DATA_FORMAT | 0x31                   | 0X00                 | Escrita  |
| DATA_X0     | 0X32                   | -                    | Leitura  |
| DATA_X1     | 0X33                   | -                    | Leitura  |
| DATAY0      | 0X34                   | -                    | Leitura  |
| DATAY1      | 0X35                   | -                    | Leitura  |
| DATA_Z0     | 0X36                   | -                    | Leitura  |
| DATA_Z1     | 0X37                   | -                    | Leitura  |

Tabela 6.1: Registradores de configuração e leitura de dados do acelerômetro ADXL345.

Da tabela 6.1, tem-se que o registrador denominado *DEVICE\_ID* é responsável por guardar o valor de identificação do dispositivo, sinalizando um correto funcionamento deste. Os registradores de *POWER\_CTL* e *DATA\_FORMAT* são escritos de forma a configurar o dispositivo para início de medição (controle de energia) e intervalo de medição. No intervalo desejado, de  $[-2g, 2g]$ , a resolução do dispositivo é igual a 10 bits. Os demais registradores apresentados guardam os valores de medição de acelerações, correspondendo, cada um, a um valor sem sinal de 8 bits, e sendo correspondentes ao *byte* mais ou menos significativo do conjunto de dados.

Os processos de inicialização e leitura de dados referentes ao giroscópio ITG3200 se dá de forma semelhante ao descrito previamente para uso do acelerômetro. Posteriormente ao estabelecimento de uma conexão com o sensor, dados de configuração são escritos à memória deste. A tabela 6.2 explicita os parâmetros utilizados para leitura de dados em forma contínua, sem o uso de interrupções. No *setup* utilizado, uma nova amostra de dados é disponível a cada 1 ms. Cada mensagem enviada é destinada ao valor hexadecimal de 0x68 deslocado em 1 bit para a esquerda, este correspondendo ao endereço de 8 bits do dispositivo.

Ainda, implementa-se uma rotina de zeramento do sensor de giro, realizando-se a leitura de 100 amostras de dados e adotando-se a média destas como valor de *offset* para o ajuste de leituras realizadas. Os tamanhos dos dados referentes à velocidade angular de cada um dos três eixos, além do procedimento de deslocamento de bits, é análogo à configuração do acelerômetro descrito. A sensibilidade atribuída ao sensor em código é conforme definido em sua ficha técnica [26].

| Registrador | Posição ( <i>hex</i> ) | Valor ( <i>hex</i> ) | Operação |
|-------------|------------------------|----------------------|----------|
| PWR_MGM     | 0x3E                   | 0x00                 | Escrita  |
| SMPLRT_DIV  | 0x15                   | 0X00                 | Escrita  |
| DLPF_FS     | 0x16                   | 0x1E                 | Escrita  |
| INT_CFG     | 0x17                   | 0x00                 | Escrita  |
| GYRO_XOUT_H | 0x1D                   | -                    | Leitura  |
| GYRO_XOUT_L | 0x1E                   | -                    | Leitura  |
| GYRO_YOUT_H | 0x1F                   | -                    | Leitura  |
| GYRO_YOUT_L | 0x20                   | -                    | Leitura  |
| GYRO_ZOUT_H | 0x21                   | -                    | Leitura  |
| GYRO_ZOUT_L | 0x22                   | -                    | Leitura  |

Tabela 6.2: Registradores de configuração e leitura de dados do giroscópio ITG3200.

A tabela 6.2 explicita os registradores utilizados para comunicação com o sensor de velocidade angular. Os quatro primeiros registradores são utilizados para a configuração do dispositivo, podendo ser utilizado modo com ou sem interrupção. No modo empregado, sem o uso de interrupção, o registrador *PWR\_MGM*, de controle de energia, é utilizado em modo padrão: o oscilador interno do dispositivo é habilitado como sendo o relógio. O registrador de controle de filtro, *DLPF\_FS* é utilizado para configuração do filtro passa-baixas interno ao componente, além da frequência de amostragem. De acordo com o utilizado, uma frequência de amostragem de 1 kHz e uma frequência de corte de 5 hz são definidas para uso do dispositivo. Uma baixa frequência de corte é utilizada devido à natureza do movimento de marcha humana, sendo este suave e lento, com ausência de movimentos bruscos – de alta frequência. Assim, o uso de uma frequência de corte mais baixa elimina a presença de vibrações no sistema, que não compõem o movimento original de marcha. O registrador de divisão de amostragem, *SMPLRT\_DIV*, é inicializado com valor nulo, portanto, o sensor gera amostras a cada 1 ms. O uso de interrupção foi desabilitado por meio do registrador *INT\_CFG*.

A obtenção de dados de posição angular por meio do encoder absoluto AMT20 ocorre por meio de uma interface SPI, com taxa de transmissão de dados em até 1 MHz. O dispositivo utilizado possui comunicação mais simples que os sensores anteriores, possuindo três tipos de comandos definidos. Assim, tem-se comandos de leitura de posição, zeramento do sensor e comando de "não-operação", este último utilizado em caso de espera de novos dados em *buffer*.

A configuração do dispositivo no ambiente CubeMX consistiu na definição de pacotes de dados de 8 bits, sendo o primeiro byte de dados correspondente ao mais significativo (MSB). O parâmetro de utilização do relógio é igual a um *prescaler* de valor 64, resultando, assim, em uma transmissão

de dados de 562,5 Kbits/s. Visto que o encoder é utilizado em modo 0 de especificação SPI, como descrito em capítulos prévios, valores de polaridade e fase iguais a *low* e *1 edge* são utilizados. Todos estes parâmetros são de acordo com as especificações do componente.

Pode-se citar aqui, ainda, a configuração do pino de *Slave Select* do microcontrolador como uma porta GPIO na interface de inicialização CubeMX. A mudança de nível lógico deste deve ser realizada, portanto, na forma de uma entrada/saída padrão de dados. Caso contrário, se configurado como saída SS do canal de comunicação SPI correspondente, o dispositivo não funcionará corretamente, devido a uma falha de arquitetura que acarreta um nível lógico sempre baixo implementado na camada de dados *HAL*.

Os resultados obtidos para o sensor de aceleração consistiram em testes de rotação da placa até que estivesse disposta em uma orientação ortogonal à inicial. Nesta configuração inicial, verifica-se uma aceleração de aproximadamente 1 g no eixo Z do dispositivo. Dessa forma, três ensaios foram realizados, rotacionando-se o objeto até uma posição de aceleração percebida igual a 1 g em cada um de seus eixos X, Y e Z. As leituras obtidas correspondem a dados de orientação aproximada ao ideal.

O primeiro dos gráficos apresentados, de acordo com a Figura 6.1, representa uma situação de rotação do objeto até uma posição de leitura de 1 g em seu eixo Z negativo, retornando, ao fim, a uma posição igual à inicial. Pode-se notar que, no processo de movimentação da placa, esta passa por uma posição de aceleração de gravidade igual a -1 em seu eixo Y. Isso confirma a correção dos dados obtidos, pois tal valor deve aparecer para algum de seus eixos de leitura ao passar por esta posição intermediária. Observa-se, ainda, a presença de um valor de *offset* no eixo X quando o objeto encontra-se em posição inicial. Isto se dá por deformidades na base da placa PCB, pois a altura dos pinos soldados diferem entre si. Ao atingir-se a posição desejada no ensaio, percebe-se que os valores de *offset* de X e Y aproximam-se entre si.

O segundo gráfico é reflexo de um ensaio no qual o sensor de aceleração linear é rotacionado, partindo-se da posição inicial, até uma disposição no plano YZ. Logo, em uma configuração onde uma aceleração da gravidade de 1 g é percebida em seu eixo X. Simetricamente, leva-se o objeto até uma disposição em que uma aceleração de -1 g é percebida, passando-se por sua posição inicial. Tais regiões podem ser percebidas na Figura 6.2. Valores maiores de aceleração no eixo X positivo podem ser percebidas, sugerindo a presença de um *offset* ou *bias* dos dados obtidos. Oscilações de leitura no eixo Y ocorrem, naturalmente, por vibrações presentes no movimento efetuado.

O terceiro gráfico de aceleração apresentado, de acordo com a Figura 6.3, é resultante de um ensaio que realiza a rotação da placa até uma posição de aceleração 1 g em Y positivo e negativo, respectivamente. Os valores obtidos são, novamente, próximos aos desejados idealmente. Valores um pouco superiores a 1 g sugerem *offset* de leitura em Y. Valores de gravidade em Z que diferem do valor nulo esperado entre os tempos de 1 e 1,4 s podem ser decorrentes de uma orientação incorreta ou *offset*.

A partir da análise dos dados de aceleração linear por meio do sensor apresentado, sugere-se a implementação de uma rotina de calibração do componente por meio da adição de um valor de *bias*. Para isso, entretanto, espera-se que seja possível simular valores corretos de orientação como

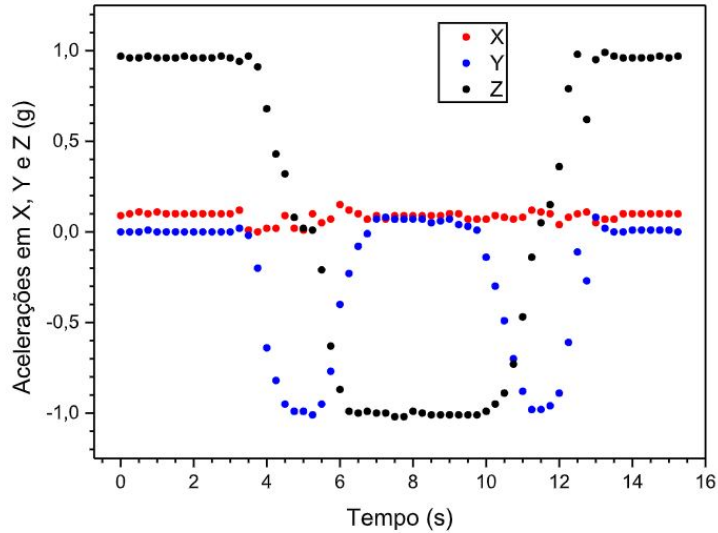


Figura 6.1: Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo Z negativo.

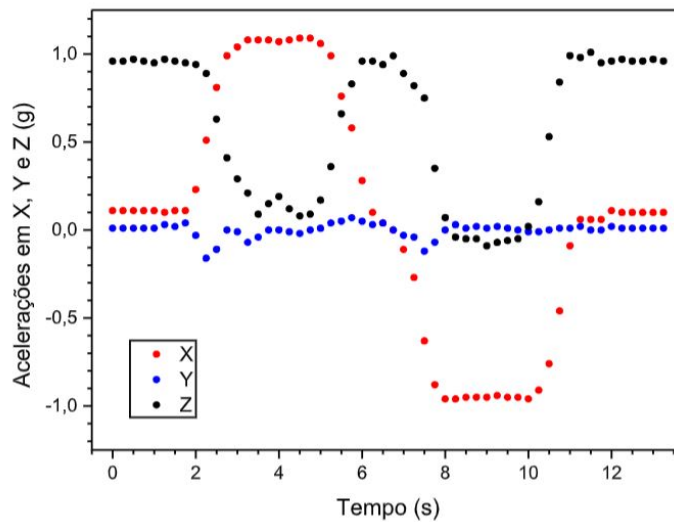


Figura 6.2: Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo X positivo e negativo.

esperado, além da impressão de valores de aceleração linear conhecidos, para comparação.

Ainda, pode-se apresentar dados provenientes de leituras de velocidade angular por meio do sensor ITG3200. Os ensaios consistiram na aplicação de oscilações à placa PCB obtida em torno de cada um de seus eixos de rotação. Os gráficos obtidos são apresentados de acordo com as Figuras 6.4, 6.5 e 6.6. Cada um destes apresenta rotação em torno dos eixos X, Y e Z, respectivamente.

A partir dos gráficos apresentados para a leitura do sensor de giro, pode-se observar que os dados são correspondentes ao esperado. Uma maior dispersão destes para os eixos X e Y

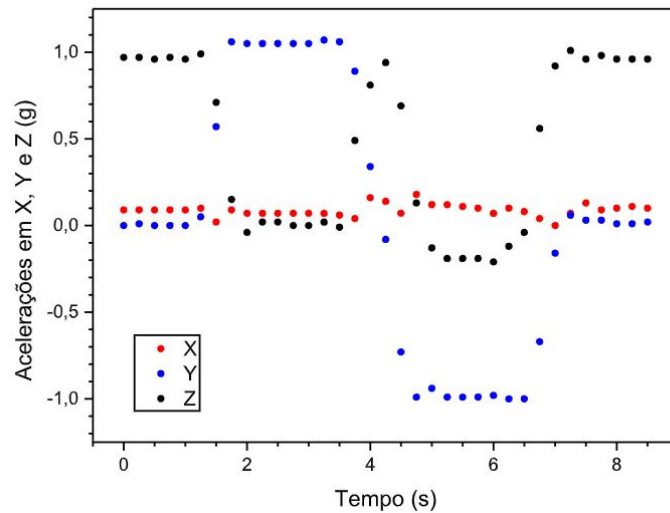


Figura 6.3: Leituras de aceleração percebida através da rotação da placa PCB para uma posição de 1 g no eixo Y positivo e negativo.

na Figura 6.6 podem ser explicadas por vibrações ocorridas durante o experimento realizado. Adicionalmente, percebe-se que os dados obtidos estão dentro de valores razoáveis de velocidade angular. O *offset* do sensor foi realizado com sucesso, uma vez que para os primeiros instantes de amostragem as leituras para todos os três eixos de rotação correspondem a zero.

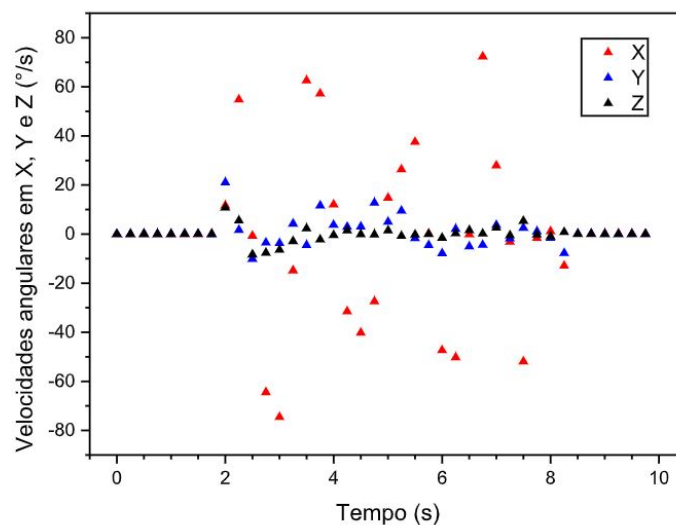


Figura 6.4: Leituras de velocidade angular através da rotação da placa PCB em torno do eixo X do giroscópio ITG3200.

Os dados de leitura de ângulo provenientes do encoder absoluto foram obtidos por meio de rotação do joelho até sua completa flexão, após rotina de zeramento deste sensor. Um ajuste dos dados lidos é realizado com relação ao sentido de rotação do sensor, definido em sentido contrário à rotação do joelho da perna de prótese. Os valores obtidos após a correção do sentido de rotação,



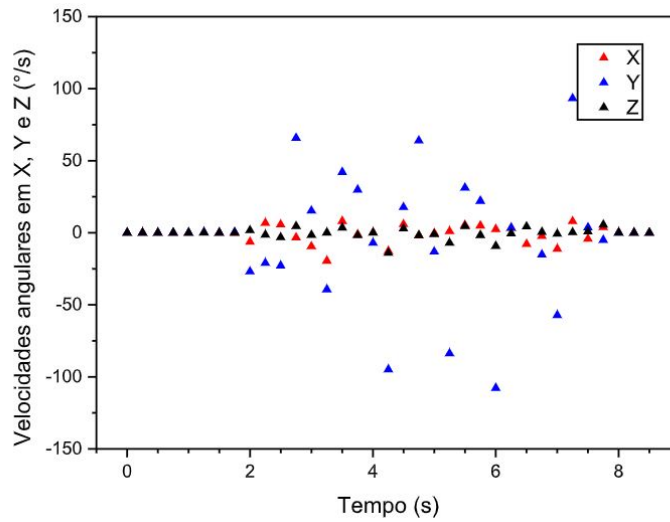


Figura 6.5: Leituras de velocidade angular através da rotação da placa PCB em torno do eixo Y do giroscópio ITG3200.

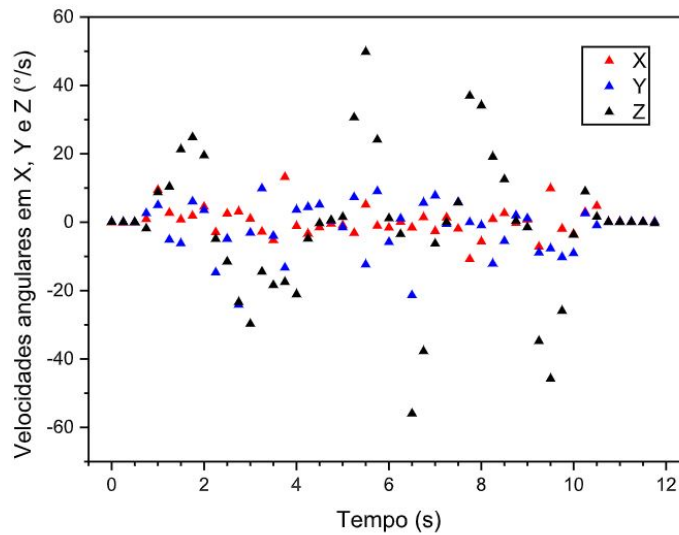


Figura 6.6: Leituras de velocidade angular através da rotação da placa PCB em torno do eixo Z do giroscópio ITG3200.

juntamente com o zeramento do sensor, são explicitados na Figura 6.7. O procedimento de teste consistiu na total flexão do joelho, em uma posição de 93 graus, e retorno à posição inicial, em sua total extensão. Uma maior velocidade de retorno pode ser percebida, no momento de extensão da perna, devido à força impressa pelo pistão de retorno à sua posição inicial.

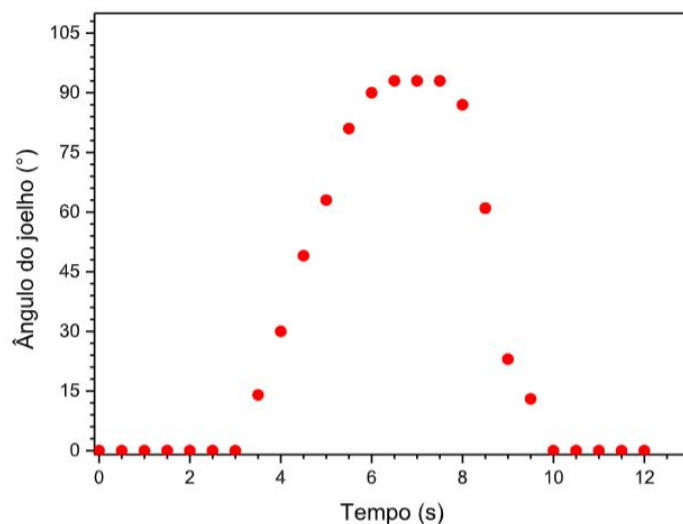


Figura 6.7: Leituras de ângulo de rotação do joelho protético, obtidas por meio do encoder AMT20.

### 6.0.2 Ensaios com o Sensor de Corrente

O sensor de corrente ACS712 foi utilizado em circuito com resistores acoplados ao seu sinal analógico de saída em um circuito divisor de tensão, evitando, assim, possíveis danos ao microcontrolador empregado. Em sua configuração original, resistores de  $100\ \Omega$  foram adicionados ao circuito. Com o uso destes valores, observou-se um sinal de saída do sensor de aproximadamente  $780\ \text{mV}$ , com pequena variação quando da aplicação de sinais de corrente através de seus terminais. Portanto, um mal-funcionamento do componente quando da utilização do circuito de divisão de tensão original foi verificado.

Para a realização de um novo ensaio da resposta do sensor ACS, modificou-se as resistências utilizadas para valores de  $1,6\ \text{k}\Omega$ . Dessa forma, um novo valor de *offset* foi observado, igual a  $2,497\ \text{V}$  na saída de sinal analógico e  $1,249\ \text{V}$  à entrada do microcontrolador. Verifica-se que ambos os valores são de acordo com o esperado, sendo o segundo metade do sinal original obtido, conforme desejado.

Após correção do circuito descrito anteriormente, realizaram-se testes com o sensor que consistiram na aplicação de valores variados de largura de pulso, para observância da corrente média aplicada ao atuador LORD. Assim, para frequências fixas de  $72\ \text{kHz}$  e de  $20\ \text{kHz}$ , obtiveram-se os valores médios de leitura do sensor de corrente ACS12. O valor de alimentação do circuito é igual a  $11,8\ \text{V}$ .

As tabelas 6.3 e 6.4 explicitam os dados de leitura obtidos através do emprego de diferentes valores médios de tensão,  $V_{avg}$ , definido conforme a relação:

$$V_{avg} = (1 - d)(V_{min}) + d(V_{max}), \quad (6.1)$$

na qual  $V_{avg}$  representa o valor médio de tensão,  $V_{min}$  representa o valor inferior de tensão aplicada,

$V_{max}$ , o valor máximo desta grandeza, e  $d$ , o ciclo de trabalho desejado. A equação é válida para qualquer sinal de onda retangular. Como o sinal inferior de tensão para a aplicação em questão é nulo, pode-se considerar apenas o segundo termo desta. A partir da relação citada anteriormente, define-se o ciclo de trabalho necessário, em programação, à obtenção do valor médio de tensão desejado.

A relação entre tensão de saída do sensor ACS712,  $V_{ACS}$ , e a corrente  $I_{damper}$  aplicada ao pistão é expressa como:

$$V_{ACS} = V_{offset} - 0,185(I_{damper}), \quad (6.2)$$

na qual  $V_{offset}$  corresponde à tensão de saída quando da ausência de passagem de corrente pelo sensor. Para a equação apresentada, o termo contendo a grandeza  $I_{damper}$  apresenta sinal negativo devido ao sentido da corrente aplicada ao dispositivo. Assim, uma diminuição do sinal medido em comparação ao valor de  $offset$  corresponde à uma maior corrente atravessando seus terminais.

O sinal de entrada ao microcontrolador para leitura do sinal de corrente, portanto,  $V'_{ACS}$ , é igual a:

$$V'_{ACS} = \frac{V_{offset} - 0,185(I_{damper})}{2}, \quad (6.3)$$

na qual a divisão do sinal pela metade se dá pela presença do divisor de tensão no circuito embarcado.

Os resultados encontrados são de acordo com o explicitado nas tabelas 6.3 e 6.4. Na seção de apêndices, pode-se verificar as imagens obtidas de leitura de valores médio de saída do sensor utilizado.

| $V_{avg}$ (V) | $V_{ACS}$ (V) | $I_{damper}$ (A) |
|---------------|---------------|------------------|
| 2             | 2,462         | 0,12             |
| 4             | 2,429         | 0,37             |
| 6             | 2,392         | 0,57             |
| 8             | 2,317         | 0,97             |

Tabela 6.3: Tensão média de sinal PWM, tensão média de saída do sensor de corrente ACS712 e corrente  $I_{damper}$  calculada, para uma frequência PWM constante de 72 kHz.

Os valores médios obtidos de acordo com a tabela 6.3 demonstram valores razoáveis de corrente aplicada ao pistão LORD. Ainda, pode-se destacar que os dados obtidos diferem do esperado para uma carga puramente resistiva de  $5 \Omega$ .

Da tabela 6.4, verifica-se que os valores de corrente a uma frequência de 20 kHz se aproximam dos obtidos à frequência de 72 kHz. Um valor negativo de corrente para um  $V_{avg}$  de 2V pode ser associado ao ruído obtido para o sensor quando da medição de correntes baixas.

De acordo com as imagens obtidas para as medições explicitadas, pode-se notar a presença de

| $V_{avg}$ (V) | $V_{ACS}$ (V) | $I_{damper}$ (A) |
|---------------|---------------|------------------|
| 2             | 2,501         | - 0,02           |
| 4             | 2,435         | 0,34             |
| 6             | 2,389         | 0,58             |
| 8             | 2,312         | 1                |

Tabela 6.4: Tensão média de sinal PWM, tensão média de saída do sensor de corrente ACS712 e corrente  $I_{damper}$  calculada, para uma frequência PWM constante de 20 kHz.

picos de corrente durante as fases de transiente do sinal modulado em largura de pulso. Portanto, a seção seguinte descreve um novo ensaio realizado para verificação deste sinal.

### 6.0.3 Ensaio através da utilização de um circuito com adição de resistência em série ao atuador

Para ser capaz de observar a forma de onda de corrente que passa pelo pistão LORD, sem a utilização do dispositivo anterior, que possui uma frequência de corte em 80 kHz, realizou-se a montagem de um circuito com resistores em série com o atuador. Dessa forma, pode-se visualizar a tensão presente nos terminais dos resistores a fim de se obter a corrente que atua no fluido, uma vez que estas são grandezas proporcionais e de mesma fase. Devido ao valor da resistência utilizada, o ganho entre tensão observada e corrente de atuação é igual a 1 V/A. A Figura 6.8 explicita a montagem realizada, com adição de dois resistores de valor total de 1  $\Omega$  em série com o atuador LORD.

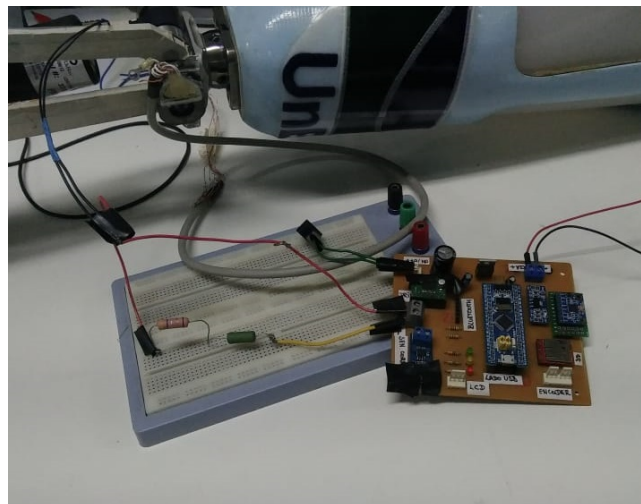


Figura 6.8: Montagem realizada para medição do sinal de corrente  $I_{damper}$ .

O sinal aplicado consiste em uma onda retangular PWM de valor médio igual a 5 V, portanto, correspondendo a um sinal de ciclo de trabalho de 42%, aproximadamente. Os resultados demonstrados nas Figuras 6.9, 6.10 e 6.11 indicam uma dinâmica com presença de sobressinal após a borda de subida do sinal PWM, em um tempo de resposta menor que 400 ns e valor máximo de

1,36 V, de acordo com 6.10.

Ainda, observa-se que a presença novamente de um sobressinal como resposta à borda de descida da onda PWM. Este sinal é de maior magnitude que o primeiro, com um menor tempo de resposta, tendo seu pico em 2,96 V, e seu mínimo em -3,72 V, de acordo com a Figura 6.11. Portanto, os valores de corrente presente no circuito durante tais instantes correspondem à 2,96 e -3,72 A. Durante os instantes remanescentes, ou seja, fora do período de resposta ao transiente, a corrente que atravessa o atuador é da ordem de 500 mA, conforme a Figura 6.10. O canal 2 de visualização nas Figuras 6.9, 6.10 e 6.11 demonstra o sinal de saída do microcontrolador.

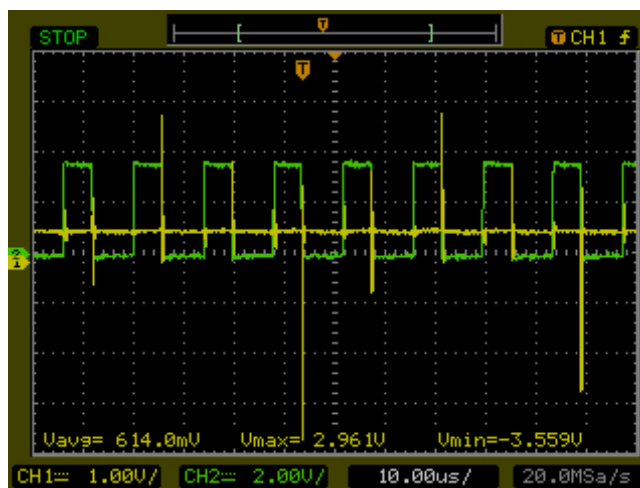


Figura 6.9: Tensão nos terminais da resistência utilizada de 1  $\Omega$  para um 5 V médio de entrada e frequência 72 kHz.

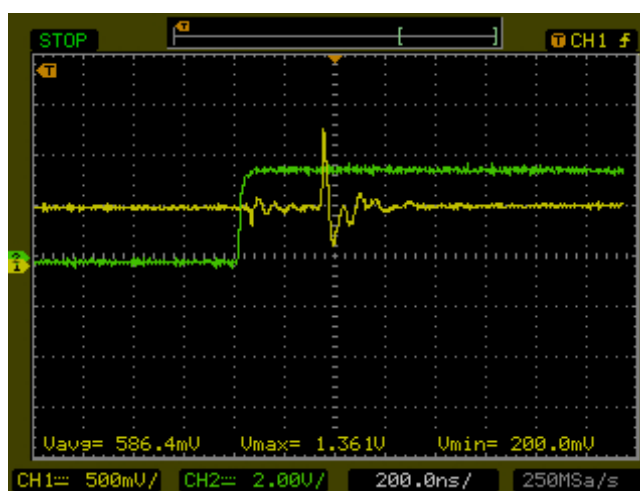


Figura 6.10: Borda de subida do sinal PWM gerado pela *Blue pill*, com resposta do sinal de corrente.

Pode-se acrescentar, ainda, uma possível interferência das resistências inseridas no circuito sobre o sinal medido. Idealmente, deve-se ser possível verificar o valor de corrente de entrada ao circuito sem a adição de tais componentes. Demais imagens obtidas durante o ensaio podem ser encontradas na seção de apêndices, e ressalta-se, por fim, a necessidade da realização de novos

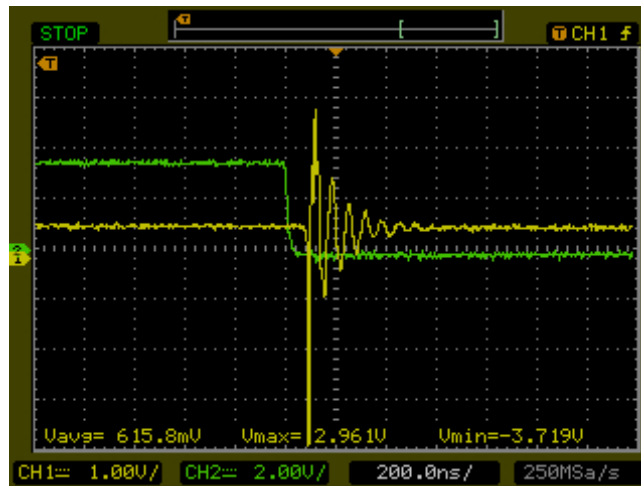


Figura 6.11: Borda de descida do sinal PWM gerado pela *Blue pill*, com resposta do sinal de corrente.

ensaios em diferentes frequências, para melhor caracterização da resposta do atuador empregado.

#### 6.0.4 Testes com uso de código de controle de amortecimento do pistão por velocidades-gatilho.

Durante a execução do projeto RLEG em sua versão anterior, foram sugeridas alterações em um dos códigos existentes de controle de marcha por meio da aplicação de um sinal em tensão fixo. O princípio de funcionamento deste é a execução de uma FSM com dois estados de atuação. Sendo assim, inicia-se o movimento de marcha em ausência de atuação. Através da identificação de um valor de velocidade angular maior que uma constante pré-definida, tem-se então a aplicação de um sinal PWM de ciclo de trabalho fixo. O sistema retorna ao estado inicial para uma velocidade menor que a definida como gatilho.

Um dos problemas encontrados na performance do algoritmo de controle descrito é o chavamento sucessivo de atuação do pistão magnetizável, uma vez que, durante o movimento de marcha, a velocidade percebida alterava-se acima e abaixo do valor definido como limite para atuação. Dessa maneira, inseriu-se uma componente de histerese ao sistema ao definir-se, em uma nova rotina de controle, limites inferior e superior de velocidades para controle do atuador. A Figura 6.12 demonstra a Máquina de Estados Finitos implementada, conforme parâmetros  $\omega_{lower}$  e  $\omega_{upper}$  definidos, e na qual  $S_0$  corresponde ao estado sem atuação, e  $S_1$ , ao estado com atuação por nível de tensão ao pistão.

O código de controle implementado, conta, portanto, com a definição de três tarefas concorrentes, gerenciadas sobre a arquitetura FreeRTOS. As tarefas consistem em: 1. Leitura de dados do encoder; 2. Escrita dos dados lidos por meio do cartão de memória; e 3. Atuação sobre o pistão. Na primeira destas, realiza-se o cálculo de velocidade angular por meio de duas amostragens de leitura de encoder, divididas pelo tempo de amostragem, como definido em 6.4:

$$\omega = \frac{\theta[k] - \theta[k - 1]}{t_{sample}}, \quad (6.4)$$

na qual  $\omega$  corresponde à velocidade angular calculada,  $\theta[k]$  ao ângulo amostrado na iteração atual,  $\theta[k - 1]$ , ao ângulo da iteração anterior, e  $t_{sample}$ , ao período de amostragem. A terceira das tarefas descritas tem como intenção verificar a alteração do estado de atuação, portanto, compara, em termos da velocidade absoluta, o valor calculado de velocidade, gerando um sinal PWM de ciclo de trabalho como definido, se necessária atuação. A tabela 6.5 explicita os parâmetros utilizados na execução da rotina.

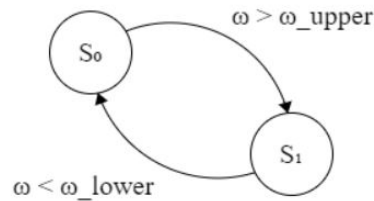


Figura 6.12: FSM para controle de amortecimento por velocidades-gatilho.

| Parâmetro        | Valor  |
|------------------|--------|
| $\omega_{lower}$ | 60°/s  |
| $\omega_{upper}$ | 80°/s  |
| $d$              | 20     |
| $t_{sample}$     | 100 ms |

Tabela 6.5: Parâmetros aplicados ao sistema de atuação do pistão LORD por meio de tensão, com região de histerese.

Através da realização de testes manuais na prótese, observou-se a correta execução das funcionalidades desejadas. O gráfico explicitado por meio da Figura 6.13 demonstra os resultados de atuação obtidos por meio desta abordagem. Nele estão presentes as variáveis de velocidade angular,  $\omega$ , calculada, e da variável de estado,  $s$ . Ainda, as barras horizontais em vermelho delimitam a região de histerese implementada.

Por meio da observação dos pontos do gráfico com relação ao estado de atuação, nota-se que este é modificado após uma iteração em relação à velocidade angular correspondente. A rotina de verificação e atuação de estado pode ser melhorada por meio da adição de notificações entre as tarefas, evitando-se um *delay* de atuação.

O código desenvolvido para controle por velocidades angulares de gatilho é capaz de demonstrar uma correta execução das tarefas desejadas por *hardware*. Pode-se adicionar, porém, que para a realização de experimentos com um paciente, existe a necessidade de melhor definição dos valores de velocidade gatilho e valor médio de tensão aplicado ao atuador. Por fim, através da manipulação da perna de prótese para fins de verificação do funcionamento da rotina em questão, observa-se

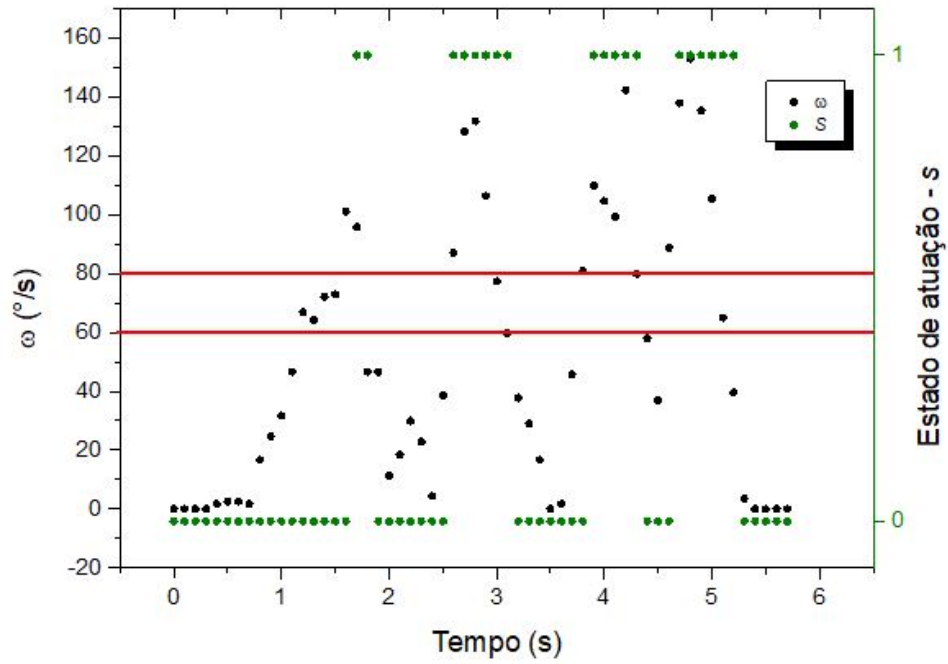


Figura 6.13: Resultados obtidos para código de atuação por tensão com adição de região de histerese.

que o emprego de um valor constante de ciclo de trabalho não é a melhor abordagem possível, pois resulta em uma resposta brusca do sistema. Assim, uma melhoria a se realizar consiste no aumento do ciclo de trabalho em passos, de forma a promover um controle mais suave ao usuário.



# Capítulo 7

## Considerações Finais

### 7.1 Conclusões

O trabalho de graduação realizado foi capaz de efetuar uma verificação do funcionamento do sistema RLEG de maneira geral, como proposto. Visto que o projeto encontrava-se estagnado por falta de uma documentação própria de seu uso, pode-se realizar um estudo sobre o circuito embarcado desenvolvido, além de uma remodelagem deste para correção de possíveis falhas e redução de custos do microcontrolador utilizado. Ainda, desenvolveram-se *scripts* de teste de cada sensor aplicado ao projeto por meio do ambiente SW4STM32 e do uso de funções de abstração de *hardware*, por meio da arquitetura HAL. Além disso, uma implementação de códigos sobre o sistema operacional de tempo real, FreeRTOS, do dispositivo, permite a garantia de execução de rotinas conforme as restrições de tempo existentes, essencial à aplicação de rotinas de controle de corrente discretizadas que podem vir a ser implementadas com uso do *hardware* obtido.

### 7.2 Trabalhos Futuros

Como trabalhos futuros a serem desenvolvidos, destaca-se um aprofundamento na caracterização da resposta do atuador LORD empregado, e em especial, um estudo acerca do comportamento do fluido magneto-reológico e de suas propriedades. Dessa forma, uma correta caracterização destes elementos permitiria a aplicação de um controle efetivo sobre a ação pistão em termos da corrente aplicada. Adicionalmente, pode-se desenvolver um sistema de identificação da intenção de movimento alidado à distinção de cada etapa do ciclo de marcha, possibilitando, conseqüentemente, a correta aplicação das características mecânicas que se deseja para cada instante da marcha. Considera-se ainda a retomada da execução de testes com um usuário real, para avaliação de códigos de controle simples e coleta de dados dos sensores embarcados aplicados à marcha, trazendo grandes benefícios à área de pesquisa em questão. Por fim, pode-se pensar em maneiras de trazer maior robustez ao sistema eletrônico da prótese, por meio de isolamento da placa PCB, ou da adição de elementos que confirmam maior segurança aos componentes utilizados, a longo prazo.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] GARLIPPE, L. A. *Estudo epidemiológico dos pacientes com amputação de membros inferiores atendidos no Centro Regional de Reabilitação de Araraquara, Estado de São Paulo, Brasil*. Dissertação (Mestrado) — Faculdade de Medicina de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2014.
- [2] CHAMLIAN, T. R. e. a. Perfil epidemiológico dos pacientes amputados de membros inferiores atendidos no lar escola são francisco entre 2006 e 2012. *Acta Fisiátrica, São Paulo*, v. 20, n. 4, p. 219–223, 2013. Acesso em: 01 de julho de 2018. Disponível em: <<http://www.revistas.usp.br/actafisiatrica/article/view/103814/102285>>.
- [3] DIAZ, C. P. O. Characterization of amputee gait using a biomechanical approach. 2015.
- [4] ROCHA, T. S. *Desenvolvimento de prótese transfemural robótica: projeto mecânico e de atuação*. Dissertação (Mestrado) — Universidade de Brasília, 2015.
- [5] PERRY, J.; DAVIDS, J. R. et al. Gait analysis: normal and pathological function. *Journal of Pediatric Orthopaedics*, LWW, v. 12, n. 6, p. 815, 1992.
- [6] NEUMANN, D. A. *Kinesiology of the Musculoskeletal System*. [S.l.]: Mosby, 2002.
- [7] JIMENEZ-FABIAN, R.; VERLINDEN, O. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Medical engineering & physics*, Elsevier, v. 34, n. 4, p. 397–408, 2012.
- [8] BLAYA, J. A. *Force-controllable ankle foot orthosis (AFO) to assist drop foot gait*. Tese (Doutorado) — Massachusetts Institute of Technology, 2002.
- [9] OYMAGIL, A. M. et al. Control of a regenerative braking powered ankle foot orthosis. In: IEEE. *Rehabilitation robotics, 2007. ICORR 2007. IEEE 10th international conference on*. [S.l.], 2007. p. 28–34.
- [10] HOLGATE, M. A.; BOHLER, A. W.; SUGA, T. G. Control algorithms for ankle robots: A reflection on the state-of-the-art and presentation of two novel algorithms. In: IEEE. *Biomedical Robotics and Biomechanics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*. [S.l.], 2008. p. 97–102.

- [11] VAROL, H. A.; GOLDFARB, M. Real-time intent recognition for a powered knee and ankle transfemoral prosthesis. In: IEEE. *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*. [S.l.], 2007. p. 16–23.
- [12] AARNO, D.; KRAGIC, D. Layered hmm for motion intention recognition. In: IEEE. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. [S.l.], 2006. p. 5130–5135.
- [13] AU, S. K.; BONATO, P.; HERR, H. An emg-position controlled system for an active ankle-foot prosthesis: an initial experimental study. In: IEEE. *Rehabilitation robotics, 2005. ICORR 2005. 9th international conference on*. [S.l.], 2005. p. 375–379.
- [14] LORD CORPORATION. *Magneto-Rheological (MR) Fluid*. Disponível em: <<https://www.lord.com/products-and-solutions/active-vibration-control/industrial-suspension-systems/magneto-rheological-mr-fluid>>. Acesso em: 28 de maio de 2019.
- [15] LORD CORPORATION. *How Does an MR Damper Work?* Disponível em: <<https://www.lord.com/products-and-solutions/active-vibration-control/industrial-suspension-systems/how-does-mr-damper-work>>. Acesso em: 27 de maio de 2019.
- [16] PJRC. *Pulsed Output: PWM Tone*. Disponível em: <[https://www.pjrc.com/teensy/td\\_pulse.html](https://www.pjrc.com/teensy/td_pulse.html)>. Acesso em: 15 de maio de 2019.
- [17] SPARKFUN. *OpenLog Hookup Guide*. Disponível em: <<https://learn.sparkfun.com/tutorials/openlog-hookup-guide/all>>. Acesso em: 30 de maio de 2019.
- [18] ARDUINO. *Serial*. Disponível em: <<https://www.arduino.cc/reference/pt/language/functions/communication/serial/>>. Acesso em: 01 de fevereiro de 2019.
- [19] ARDUINO. *Wire Library*. Disponível em: <<https://www.arduino.cc/en/Reference/Wire>>. Acesso em: 01 de fevereiro de 2019.
- [20] ARDUINO. *SPI library*. Disponível em: <<https://www.arduino.cc/en/Reference/SPI>>. Acesso em: 01 de fevereiro de 2019.
- [21] SPARKFUN. *Serial Communication*. Disponível em: <<https://learn.sparkfun.com/tutorials/serial-communication>>. Acesso em: 01 de fevereiro de 2018.
- [22] STANFORD. *Basics of UART Communication - Stanford University*. Disponível em: <<https://web.stanford.edu/class/cs140e/docs/uart-basics.pdf>>. Acesso em: 01 de novembro de 2018.
- [23] PJRC. *Teensy USB Development Board*. Disponível em: <<https://www.pjrc.com/teensy/>>. Acesso em: 03 de maio de 2019.
- [24] PJRC. *IntervalTimer*. Disponível em: <[https://www.pjrc.com/teensy/td\\_timing\\_IntervalTimer.html](https://www.pjrc.com/teensy/td_timing_IntervalTimer.html)>. Acesso em: 03 de maio de 2019.

- [25] SPARKFUN. *Digital Accelerometer ADXL345 Datasheet*. Disponível em: <<https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>>. Acesso em: 01 de junho de 2019.
- [26] SPARKFUN. *ITG-3200 Product Specification Revision 1.4*. Disponível em: <<https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>>. Acesso em: 01 de junho de 2019.

# APÊNDICE

# I. ESQUEMÁTICOS DA PCB RLEG

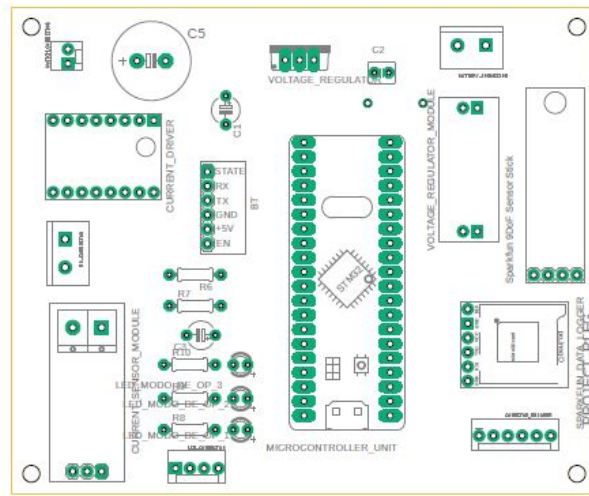


Figura II.1: Modelo da PCB RLEG com contorno de componentes.

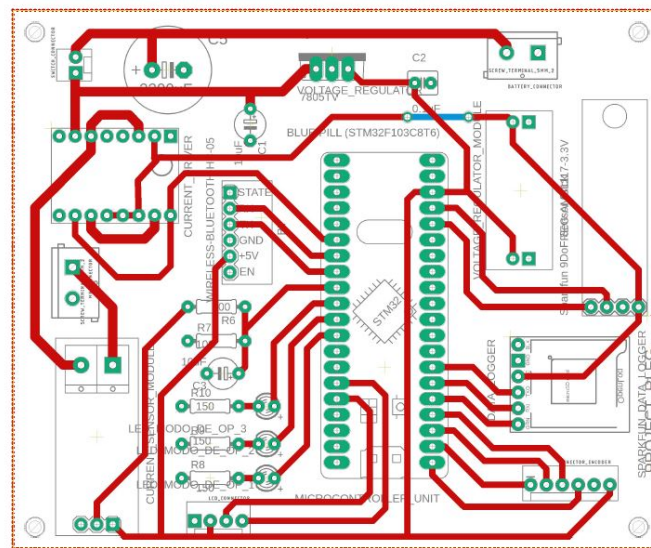
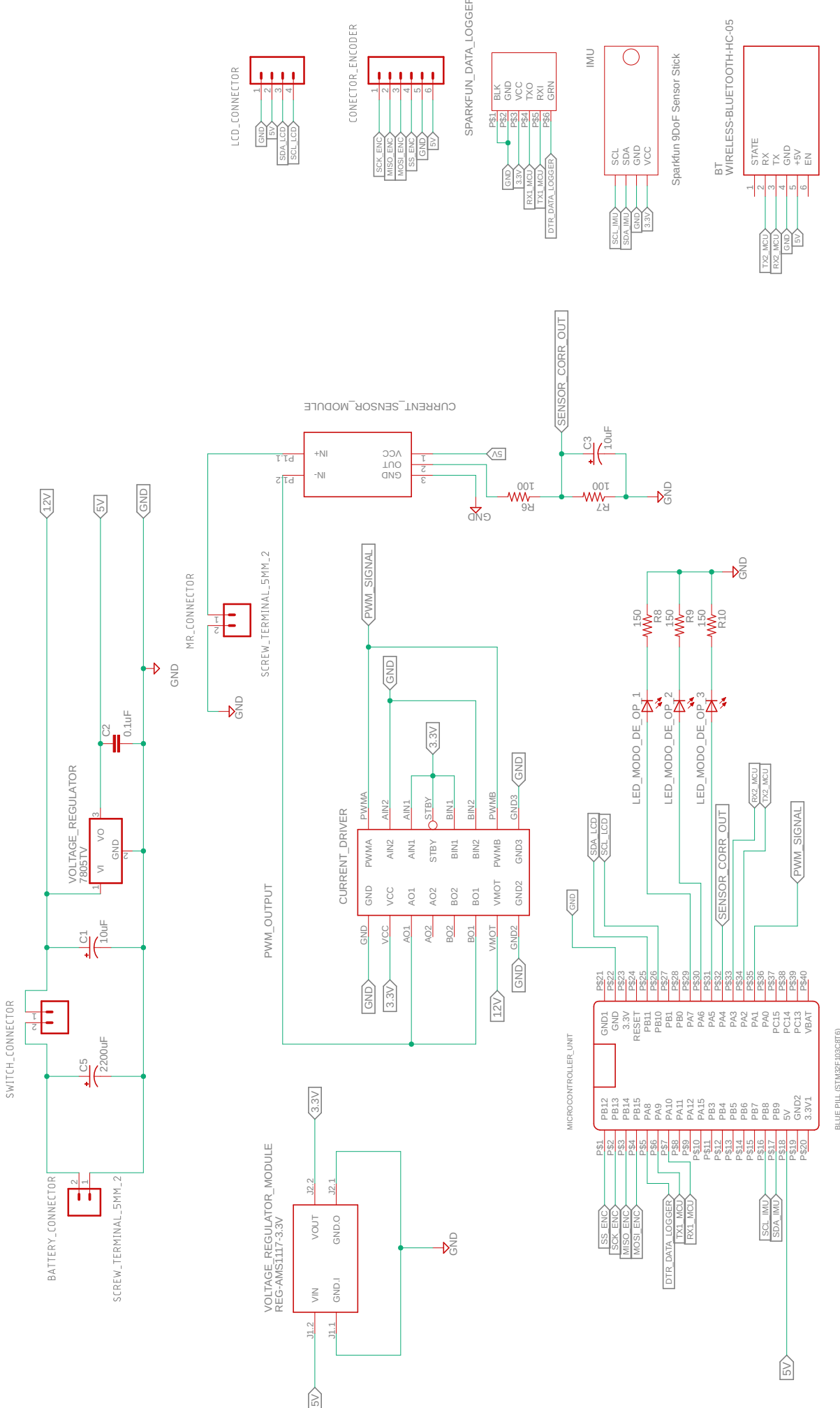


Figura II.2: Modelo da PCB RLEG com destaque de trilhas.

## II. ESQUEMÁTICO DO CIRCUITO RLEG





### III. IMAGENS EM OSCILOSCÓPIO

#### III.1 Ensaios para frequência PWM de 72 kHz

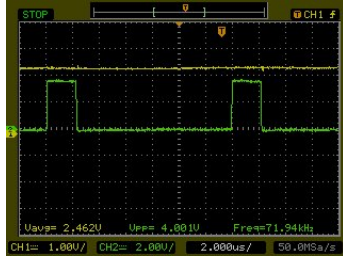


Figura III.1:  $V_{ACS}$  para  $V_{AVG} = 2V$ , 72 kHz.

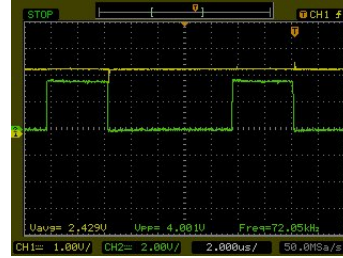


Figura III.2:  $V_{ACS}$  para  $V_{AVG} = 4V$ , 72 kHz.

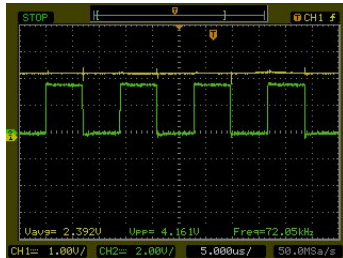


Figura III.3:  $V_{ACS}$  para  $V_{AVG} = 6V$ , 72 kHz.

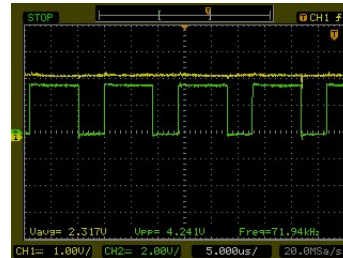


Figura III.4:  $V_{ACS}$  para  $V_{AVG} = 8V$ , 72 kHz.

### III.2 Ensaios para frequência PWM de 20 kHz

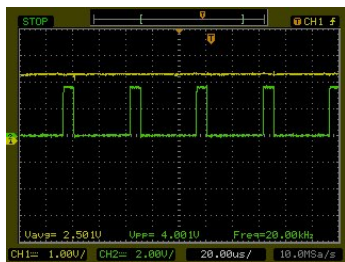


Figura III.5:  $V_{ACS}$  para  $V_{AVG} = 2V$ , 20 kHz.

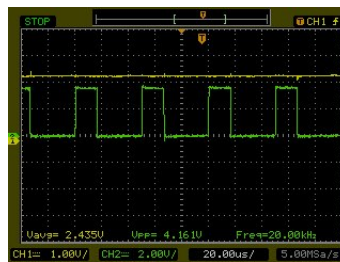


Figura III.6:  $V_{ACS}$  para  $V_{AVG} = 4V$ , 20 kHz.

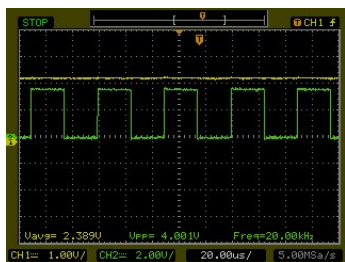


Figura III.7:  $V_{ACS}$  para  $V_{AVG} = 6V$ , 20 kHz.

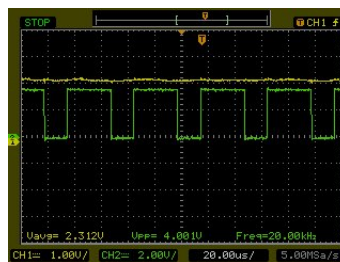


Figura III.8:  $V_{ACS}$  para  $V_{AVG} = 8V$ , 20 kHz.

### III.3 Resposta do atuador com uso de resistores em série

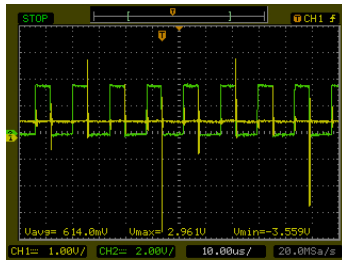


Figura III.9: Resposta para  $V_{AVG} = 5V$ , 72 kHz. Divisão de tempo 10  $\mu s$ .

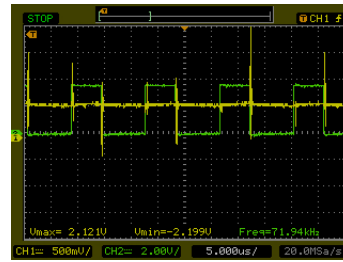


Figura III.10: Resposta para  $V_{AVG} = 5V$ , 72 kHz. Divisão de tempo 5  $\mu s$ .

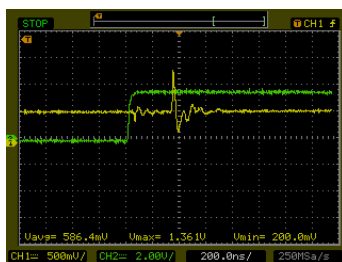


Figura III.11: Resposta para  $V_{AVG} = 5V$ , 72 kHz, borda de subida.

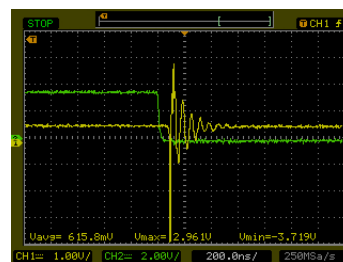


Figura III.12: Resposta para  $V_{AVG} = 5V$ , 72 kHz, borda de descida.

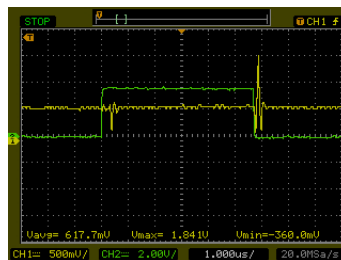
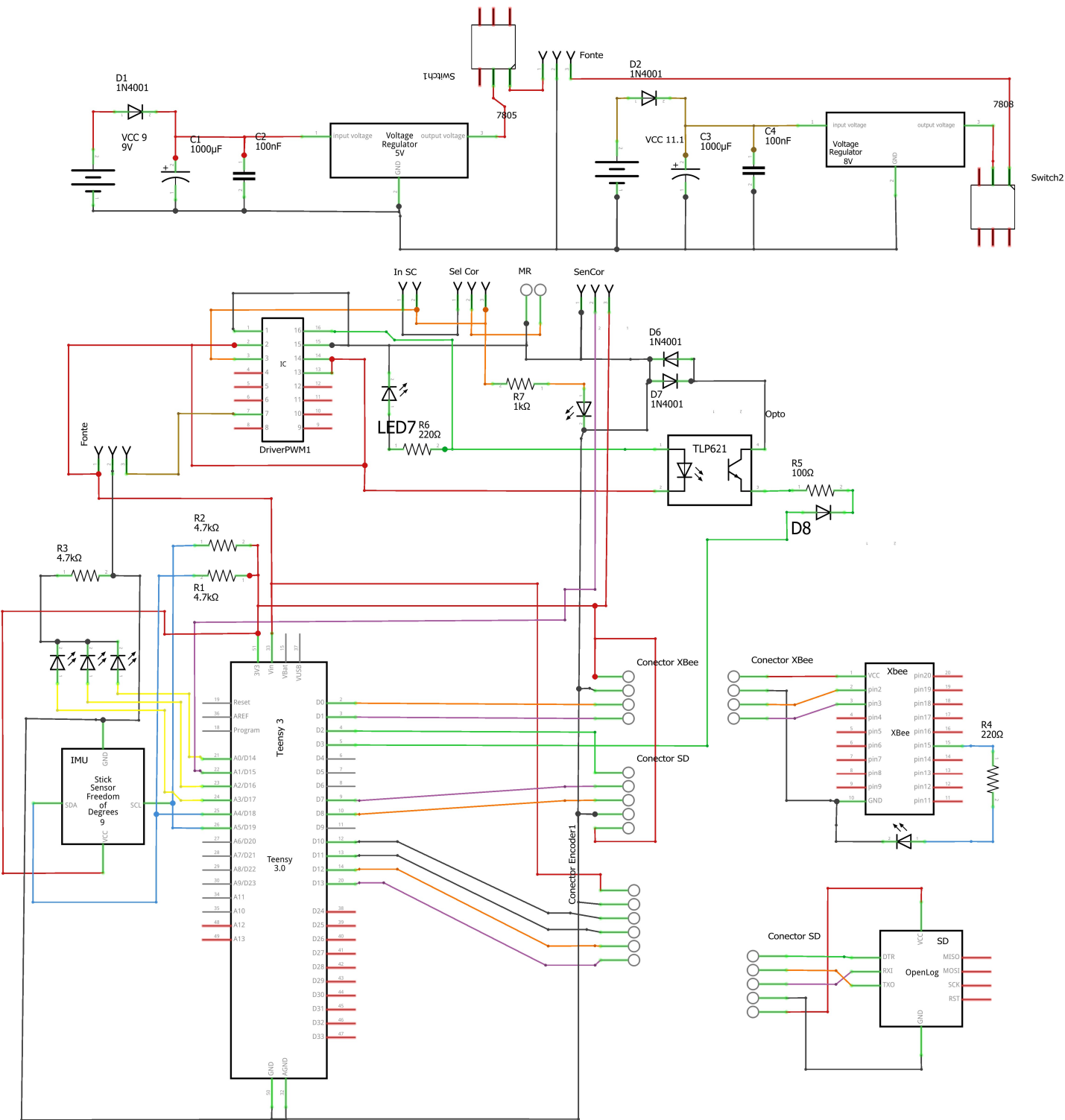


Figura III.13: Resposta transiente em destaque.  $V_{AVG} = 5V$ , 72 kHz.

# ANEXOS

# I. ESQUEMÁTICO DO CIRCUITO RLEG ANTERIOR



fritzing