



**Daniel de Sousa Leite**

**Dispositivo Robótico para Assistência à Locomoção  
de Pessoas Idosas em Ambientes Urbanos**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio.

Orientadora: Profa. Marley Maria Bernardes Rebuzzi Vellasco  
Co-orientadora: Profa. Karla Tereza Figueiredo Leite

Rio de Janeiro  
Setembro de 2016



**Daniel de Sousa Leite**

**Dispositivo Robótico para Assistência à Locomoção  
de Pessoas Idosas em Ambientes Urbanos**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Profa. Marley Maria Bernardes Rebuzzi Vellasco**  
Orientadora  
Departamento de Engenharia Elétrica – PUC-Rio

**Profa. Karla Tereza Figueiredo Leite**  
Co-orientadora  
UEZO

**Prof. Paulo Fernando Ferreira Rosa**  
IME

**Prof. José Franco Machado do Amaral**  
UERJ

**Prof. Carlos Roberto Hall Barbosa**  
Programa de Pós-Graduação em Metrologia – PUC-Rio

**Prof. Márcio da Silveira Carvalho**  
Coordenador Setorial do Centro  
Técnico Científico

Rio de Janeiro, 14 de setembro de 2016

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

**Daniel de Sousa Leite**

Graduou-se em Engenharia de Controle e Automação pela Pontifícia Universidade Católica do Rio de Janeiro em 2013.

Ficha Catalográfica

Leite, Daniel de Sousa

Dispositivo robótico para assistência à locomoção de pessoas idosas em ambientes urbanos / Daniel de Sousa Leite ; orientadora: Marley Maria Bernardes Rebuzzi Vellasco ; co-orientadora: Karla Tereza Figueiredo Leite. – 2016.

230 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2016.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Robô móvel. 3. Bengala robótica. 4. Lógica Fuzzy. 5. PlayerStage. 6. Navegação autônoma. I. Vellasco, Marley Maria Bernardes Rebuzzi. II. Leite, Karla Tereza Figueiredo. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

## Agradecimentos

A Deus, pelas oportunidades concedidas e a força para enfrentar todos os desafios.

Aos meus pais, Marina Vieira de Sousa Leite e Luiz Carlos de Souza Leite, pelo amor, dedicação e valores que me foram passados.

Ao meu irmão, Luiz Gustavo de Sousa Leite, pelo apoio e amizade incondicional.

As minhas tias, Maurinha Vieira e Marlene Vieira Pereira, pelo apoio e amor.

A minha namorada, Walquiria Amaro da Paz, pela amizade, apoio e incentivo.

As minhas orientadoras, Marley Vellasco e Karla Figueiredo, por todo o apoio na elaboração dessa dissertação e no decorrer de minha vida acadêmica.

A FAPERJ e a PUC-Rio pelos auxílios concedidos.

Aos professores que participaram da banca examinadora.



## Resumo

Leite, Daniel de Sousa; Vellasco, Marley Maria Bernardes Rebuzzi; Leite, Karla Tereza Figueiredo. **Dispositivo Robótico para Assistência à Locomoção de Pessoas Idosas em Ambientes Urbanos**. Rio de Janeiro, 2016. 230p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Com o aumento da expectativa, de vida o envelhecimento da população vem se tornando uma realidade cada vez mais presente no Brasil e no mundo. Esse novo panorama demográfico já é vivenciado por países ricos, que vêm cada vez mais investindo para se enquadrar nessa nova realidade, seja por meio da adaptação de suas cidades ou pelo desenvolvimento de novas tecnologias para melhora da qualidade de vida. Na área da robótica, diversas pesquisas vêm sendo desenvolvidas com o intuito de reabilitação e melhora da qualidade de vida da população idosa. Nesses trabalhos são desenvolvidos, por exemplo, dispositivos que buscam auxiliar o idoso na realização de suas atividades diárias, provendo, principalmente, suporte e prevenção de quedas. Essa dissertação de mestrado apresenta o desenvolvimento do protótipo de um dispositivo para assistência a locomoção de pessoas idosas que possuam alguma deficiência visual, motora e/ou cognitiva. O dispositivo tem como objetivo guiar o usuário em ambientes urbanos de maneira autônoma. O protótipo deve ser capaz de desviar de qualquer obstáculo que possa levar o idoso à queda, além de ter uma estrutura que ofereça apoio para o seu deslocamento. O dispositivo proposto possui uma estrutura semelhante a um andador, cuja base é um robô móvel diferencial. Para que possa obter informações do ambiente, o dispositivo está equipado com sensores de distância, uma central inercial e *encoders* nas rodas. Todo o processamento ocorre em uma CPU de baixo custo, Raspberry Pi 1 versão 2, embarcada no próprio dispositivo e o controle de navegação ocorre por meio de um algoritmo baseado em lógica Fuzzy. Os acessos ao hardware e software de controle do dispositivo são gerenciados pelo framework de robótica Player (Gerkey e contribuidores, 2010). Para que o dispositivo receba a rota de navegação ele está conectado a um celular, com sistema operacional Android, via protocolo TCP/IP.

Esse celular está executando uma API (*Application Programming Interface*) do Google Maps que fornece direção e distância ao objetivo a cada passo da interação, além da localização global do dispositivo, por meio do sensor GPS do celular. O objetivo deve ser inicialmente estabelecido pelo usuário por meio da API desenvolvida, para que a navegação autônoma ocorra. Além da navegação autônoma, o dispositivo permite que usuário envie comandos diretamente para os motores por meio de sensores de força instalados próximos aos pontos de apoio do usuário.

### **Palavras-Chave**

Robô móvel; bengala robótica; Lógica Fuzzy; PlayerStage; sistema embarcado; navegação autônoma; ambiente externo; Raspberry Pi; idoso

## Abstract

Leite, Daniel de Sousa; Vellasco, Marley Maria Bernardes Rebuzzi (Advisor); Leite, Karla Tereza Figueiredo (Co-Advisor). **Robotic Device for Mobility Assistance to Elderly People in Urban Environments**. Rio de Janeiro, 2016. 230p. MSc. Dissertation - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro

With the increase in life expectation, the ageing population has become more present in Brazil and the world. This new demographic scenery has been already framed by rich countries, which are increasingly investing to fit this new reality, either through the adaptation of their cities or the development of new technologies to improve the quality of life. In the area of robotics, several researches have been developed with the aim of rehabilitation and improvement of the quality of life of the elderly population. These researches are developing, for example, devices to assist the elderly in carrying out their daily activities, providing support and prevention of falls. This work presents the development of the prototype of a device to assist elderly person with any visual, cognitive and/or motor impairment to locomotion by itself. The device aims to guide the user autonomously in urban environments. The prototype should be able to avoid any obstacle that can cause the elderly to fall, besides having a structure that offers support for his balance. The proposed device has a structure similar to a walker whose base is a differential mobile robot. For the device be able to get information from the environment, it is embedded with range sensors, a measurement central unit and encoders at the wheels. All processing occurs in a low-cost CPU, Raspberry Pi 1 B version 2, which is embedded in the mobile device, and the navigation control algorithm is based on fuzzy logic. The robotic framework Player (Gerkey and contributors, 2010) provides the access to the hardware and software of the device. For the device to receive the navigation route, it is connected to an Android operating system phone, by TCP/IP protocol. This phone runs an API (*Application Programming Interface*) from Google Maps that provides the direction and the distance to the goal in every step of its interaction, besides the global location of the robot, provided by the GPS

sensor of the phone. The user should firstly set the goal with the API developed, so that the autonomous navigation will occur. In addition to the autonomous navigation, the device allows the user to send commands directly to the motors by means of the force sensors installed at the robot cane.

## **Keywords**

Mobile robot; cane robotics; Fuzzy Logic; PlayerStage; embeded system; autonomous navigation; outdoor environment; Raspberry Pi; elder

# Sumário

1 Introdução	31
1.1 Motivação	31
1.2 Trabalhos relacionados	33
1.3 Objetivos	38
1.4 Descrição do Trabalho	39
1.5 Organização da Dissertação	40
2 Fundamentos teóricos	42
2.1 Projeto Mecânico	42
2.1.1 Design do protótipo	42
2.1.2 Modelo Cinemático	43
2.2 Projeto eletrônico	46
2.2.1 Atuadores	46
2.2.1.1 Motor CC	47
2.2.1.2 Caixa de Redução	53
2.2.1.3 Driver de Potência	54
2.2.2 Sensores	55
2.2.2.1 Sensores de detecção de obstáculo	56
2.2.2.1.1 Sensor de Ultrassom	56
2.2.2.1.2 Sensores Laser	59
2.2.2.1.3 Sensor de visão	62
2.2.2.2 Encoder Rotativo	64
2.2.2.3 Acelerômetro	68
2.2.2.4 Giroscópio	71

2.2.2.5	Magnetômetro	74
2.2.2.6	Central Inercial	81
2.2.2.7	Sensor de força e torque	82
2.2.3	Hardware processador	83
2.3	Software	86
2.3.1	Player	86
2.3.2	Stage	89
2.3.3	API Google Maps	91
2.4	Controle	92
2.4.1	Controle PID	93
2.4.2	Lógica Fuzzy	98
3	Desenvolvimento do dispositivo robótico – Implementação em ambiente virtual	102
3.1	Condições de contorno	102
3.2	Controle de navegação	105
3.2.1	Ambiente de Simulação	106
3.2.1.1	Robô virtual	108
3.2.1.2	Ambiente de navegação	111
3.2.2	Sistema de Inferência Fuzzy	113
3.2.2.1	Fuzzyficação	114
3.2.2.2	Base de regras	117
3.2.2.3	Defuzzyficação	119
4	Desenvolvimento do dispositivo robótico – Aplicação real	120
4.1	Projeto Mecânico	120
4.2	Projeto eletrônico	125
4.2.1	Atuadores	126

4.2.1.1	Motor CC	126
4.2.1.2	Driver de potência	134
4.2.1.3	Baterias	136
4.2.2	Sensores	137
4.2.2.1	Sensor de ultrassom	137
4.2.2.2	Encoder Rotativo	141
4.2.2.3	Central Inercial	143
4.2.2.4	Sensor de força	146
4.2.3	Hardware processador	148
4.2.4	Montagem	150
4.2.5	Player	157
4.2.5.1	AdcDriver	161
4.2.5.2	DioDriver	161
4.2.5.3	RTimuDriver	162
4.2.5.4	UltrassomDriver	162
4.2.5.5	OdometroDriver	163
4.3	Controle	163
4.3.1	Controle PID	164
4.3.2	Lógica Fuzzy	168
5	Testes e Resultados	172
5.1.	Modelo Simulado	172
5.1.1.	Trajeto 1	172
5.1.2.	Trajeto 2	179
5.1.3.	Trajeto 3	183
5.2.	Modelo Real	186
5.2.1.	Percurso 1	187

5.2.2. Percurso 2	192
5.2.3. Percurso 3	197
6 Conclusão e trabalhos futuros	203
6.1 Conclusão	203
6.2 Trabalhos futuros	205
7 Referências bibliográficas	208
8 Apêndices	218
8.1 Tabela de inferência do modelo virtual	218
8.2 Trecho da tabela de inferência do modelo real	230



## Lista de figuras

Figura 1 – Protótipo do “Intelligent Cane Robot” (Di et al., 2012)	33
Figura 2 - Dispositivo “SmartCane PAMM System” (Dubowsky et al., 2000)	34
Figura 3 - Dispositivo “Assistive Robotic Walker” (Wasson et al., 2001)	34
Figura 4 - Protótipo do “ZJU Walker”. (Zhou et al., 2010)	34
Figura 5 - Protótipo “RT-Walker” com sensor Lidar (Hirata et al., 2008)	35
Figura 6 - Protótipo “RT-Walker” com sensor Kinect (Taghvaei et al., 2011)	35
Figura 7 - Protótipo JAIST Active Robotic Walker (Ohnuma et al., 2011)	35
Figura 8 - Protótipo “GuideCane” (Borenstein e Ulrich, 1997)	37
Figura 9 - Bengala robótica interativa “RoJi” (Shim et al., 2004)	37
Figura 10 - Diagrama da base de um robô diferencial (Dhaouadi e Hatab, 2013)	43
Figura 11 - Circuito equivalente do Motor CC com escovas de excitação independente	48
Figura 12 - Circuito equivalente do Motor CC com escovas de excitação série	48

Figura 13 - Circuito equivalente do Motor CC com escovas de excitação paralela	49
Figura 14 - Circuito equivalente do Motor CC com escovas de excitação composta	49
Figura 15 - Circuito equivalente do Motor CC com escovas de excitação por ímãs permanentes	50
Figura 16 - Circuito equivalente de motor Brushless de três bobinas (Trindade, 2009)	50
Figura 17 - Diagrama em blocos de um motor CC com escovas e excitação por ímãs permanentes	52
Figura 18 - Representação de um circuito ponte-H	55
Figura 19 - Diagrama de tempo de um sensor de ultrassom (Siegwart e Nourbakhsh, 2004)	57
Figura 20 - Ilustração de dois tipos de superfícies que causam reflexão de canto (Wielenberg, 2015)	58
Figura 21 - Representação de sensor de distância por mudança de fase (Siegwart e Nourbakhsh, 2004)	59
Figura 22 - Ilustração de mecanismo de rotação de sensor LIDAR (Romero et al., 2014)	60
Figura 23 - Esquema de funcionamento de sensor de distância laser por triangulação (Siegwart e Nourbakhsh, 2004)	61
Figura 24 - Exemplo de disco de sensor rotativo absoluto de 16 posições codificada em Gray Code	65
Figura 25 - Exemplo das saídas por onda quadrada de um sensor rotativo incremental	66

Figura 26 - Exemplo de <i>encoder</i> rotativo incremental mecânico	66
Figura 27 - Ilustrações de <i>encoders</i> magnéticos por efeito Hall, material magneto resistivo e material indutivo (Eitel, 2014; Fraden, 2010).	67
Figura 28 – Ilustração de <i>encoder</i> óptico incremental e absoluto (Eitel, 2014)	67
Figura 29 – Ilustração de acelerômetro capacitivo MEMS (Nedelkovski, 2015)	69
Figura 30 - Ilustração de giroscópio mecânico	71
Figura 31 - Ilustração da estrutura de um giroscópio MEMS (SensorWiki, 2013a)	72
Figura 32 - Ilustração do efeito do deslocamento do corpo de prova com a rotação do sensor (SensorWiki, 2013a)	72
Figura 33 - Ilustração do percurso de teste do efeito Sagnac	73
Figura 34 - Ilustração da influência da rotação do circuito de teste de Sagnac na convergência dos feixes	73
Figura 35 - Ilustração do efeito Hall (Fraden, 2010)	75
Figura 36 - Representação de um magnetômetro <i>fluxgate</i>	76
Figura 37 - Exemplo de dados do magnetômetro sem distorção (Konvalin, 2009)	77
Figura 38 - Eixos em notação NED com indicação dos ângulos de Roll, Pitch e Yaw.	77
Figura 39 - Exemplo de erro de orientação devido a um ângulo de <i>pitch</i> de 40 graus, sem realizar a compensação de <i>tilt</i> (Kionix, 2015)	78
Figura 40 - Componente e orientação resultante do magnetômetro com e sem compensação de <i>tilt</i> (Konvalin, 2009)	78

Figura 41 - Exemplo de dados do magnetômetro com distorção hard-iron (Konvalin, 2009)	80
Figura 42 - Exemplo do efeito da distorção soft-iron (Phidgets, 2014; Konvalin, 2009)	81
Figura 43 - Diagrama básico de operação de uma central inercial (Siciliano e Khatib, 2008)	81
Figura 44 - Top 10 tipos SBCs de desenvolvimento aberto (Lehrbaum e Brown, 2014)	84
Figura 45 - Exemplo de arquivo de configuração do Player	88
Figura 46 - Estrutura macro do software Player	89
Figura 47 - Ilustração de simulação com o Stage (Vaughan e contribuidores, 2011)	89
Figura 48 - Exemplo de arquivo de configuração STAGE e diagrama dos módulos do STAGE	90
Figura 49 - Diagrama em blocos de um controlador PID paralelo aplicado a uma planta	94
Figura 50 - Resposta de um sistema de segunda ordem com polos complexos conjugados a uma entrada degrau unitário (Pavani, 2011)	95
Figura 51 - Resposta à entrada degrau de um sistema de 2ª ordem com controlador P em série	95
Figura 52 - Respostas à entrada degrau de sistemas de 2ª ordem com controlador P e PI em série	96
Figura 53 - Respostas a entrada degrau de sistemas de 2ª ordem com controlador P, PI e PD em série	97
Figura 54 - Respostas à entrada degrau de sistemas de 2ª ordem com controlador P, PI, PD e PID em série	97

Figura 55 - Exemplo de conjuntos de valores de uma variável Fuzzy	99
Figura 56 - Diagrama de um sistema de inferência fuzzy (Tanscheit, n.d.)	100
Figura 57 - Exemplo de fuzzyficação	100
Figura 58 - Diagrama macro das entradas (azul) e saídas (vermelho) do SIF do modelo virtual	106
Figura 59 - Representação das dimensões do robô virtual (vista superior)	108
Figura 60 - Representação do robô virtual com os sensores de ultrassom	110
Figura 61 - Ilustração da área de atuação dos sensores de ultrassom do robô virtual	111
Figura 62 - Mapa de navegação completo	112
Figura 63 - Mapa de navegação com zoom na região do robô	112
Figura 64 - Diagrama macro do SIF do modelo virtual com a separação dos dados dos sensores de distância	113
Figura 65 - Funções de pertinência das variáveis do tipo Sensor	114
Figura 66 - Funções de pertinência das variáveis do tipo Sensor Altura	115
Figura 67 - Funções de pertinência da variável Ângulo	115
Figura 68 - Indicação dos valores precisos de ângulos positivos e negativos de orientação ao objetivo	116
Figura 69 - Funções de pertinência da variável Distância	116
Figura 70 - Funções de pertinência das variáveis MotorE(D)	117

Figura 71 - Exemplo da base de regras com indicação dos operadores	119
Figura 72 - Vista frontal modelo CAD	121
Figura 73 - Vista superior modelo CAD	121
Figura 74 - Vista lateral esquerda modelo CAD	122
Figura 75 - Vista em perspectiva modelo CAD	122
Figura 76 - Vista em perspectiva com destaque em amarelo para peças impressas	123
Figura 77 - Vista frontal	124
Figura 78 - Vista superior	124
Figura 79 - Vista lateral esquerda	125
Figura 80 - Diagrama de corpo livre do protótipo subindo uma rampa	127
Figura 81 - Especificações motor CEP F 006 WM0 310	128
Figura 82 - Especificações motor GPB F 006 KM0 60N	128
Figura 83 - Especificações redutor MKS MR4 1:10 63 B14	129
Figura 84 - Tensão de entrada (azul), velocidade executada (preto) e corrente consumida (vermelho) do conjunto motor-redutor do protótipo	131
Figura 85 - Diagrama em blocos do modelo simulado para o conjunto motor-redutor	131
Figura 86 - Gráficos da tensão de entrada (azul), velocidade de saída (preto) e corrente consumida (vermelho) do modelo motor-redutor com os parâmetros da Tabela 2	132

Figura 87 - Velocidade angular e corrente consumida do conjunto motor-redutor simulado (azul), com os parâmetros da Tabela 2 e Tabela 3, e do conjunto real (preto) aplicando uma mesma entrada	133
Figura 88 - Velocidade angular e corrente consumida do conjunto motor-redutor simulado (azul) com os parâmetros $K_t$ e $K_e$ da Tabela 2 e demais parâmetros da Tabela 4 e do conjunto real (preto) aplicando uma mesma entrada	134
Figura 89 - Driver de potência Pololu Dual VNH5019 (Pololu, 2011)	135
Figura 90 - Esquema de conexões do driver de potência Pololu Dual VNH5019 (Pololu, 2011)	136
Figura 91 - Bateria LiPo Fullmax 6s 7,7 Ah	136
Figura 92 - Bateria LiPo Fullmax 3s 2,2 Ah	137
Figura 93 - Sensor de ultrassom LV-MaxSonar EZ1 (MaxBotix Inc., 2015)	138
Figura 94 - Campo de atuação LV-MaxSonar-EZ1 (MaxBotix Inc., 2015)	139
Figura 95 - Configuração de acionamento sequencial (MaxBotix Inc., 2015)	139
Figura 96 - Distribuição dos sensores de ultrassom no dispositivo robótico	140
Figura 97 - <i>Encoder</i> rotativo óptico o ENA1J-B28-L00100L (Bourns, 2015)	141
Figura 98 - Forma de onda das saídas do <i>encoder</i> ENA1J-B28-L00100L (Bourns, 2015)	142
Figura 99 - <i>Encoders</i> conectados aos motores	142
Figura 100 - IMU Pololu MinIMU-9 v2 (Pololu, 2014)	143

Figura 101 - Posição da IMU no protótipo	144
Figura 102 - Dados dos magnetômetros da IMU com correção (azul) e sem correção (vermelho)	146
Figura 103 - FSR 400 e curva característica (Interlink Electronics, 2015)	147
Figura 104 - Distribuição dos sensores de força sobre as pegas esquerda e direita do dispositivo	148
Figura 105 - Raspberry Pi B ver 2 com destaque aos pinos e entrada e saída	149
Figura 106 - Ilustração na pinagem do chip MCP3008 (Microchip Technology Inc., 2008)	150
Figura 107 - Ilustração da pinagem do chip MCP23017 (Microchip Technology Inc., 2007)	150
Figura 108 - Diagrama circuito regulador de tensão	151
Figura 109 - Diagrama do circuito de atuação e controle dos motores	152
Figura 110 - Diagrama de interligação dos sensores de ultrassom	154
Figura 111 - Diagrama da interface AIO e dos sensores de força	155
Figura 112 - Diagrama da interface IMU, DIO e dos sensores de contato	156
Figura 113 - Posicionamento das chaves do contato NF para identificar colisões	157
Figura 114 - Diagrama genérico da rotina de um driver para o servidor Player	158
Figura 115 - Fluxograma programa cliente	159



Figura 116 - Diagrama macro do SIF do modelo real com aplicação do controle PID	164
Figura 117 - Circuito de malha fechada do regulador PID discreto do conjunto motor-redutor	165
Figura 118 - Resposta ao degrau do diagrama da figura 113	165
Figura 119 - Resposta do modelo motor-redutor (preto) utilizando o regulador PID com o primeiro conjunto de ganhos (Tabela 5) e saída desejada (azul)	166
Figura 120 - Resposta do motor-redutor real (preto) com o controlador PID utilizando o primeiro conjunto (Tabela 5) de ganhos e saída desejada (azul)	167
Figura 121 - Saída do motor-redutor real com o regulador PID utilizando conjunto de ganhos da Tabela 5 (preto), o conjunto de ganhos da Tabela 6(vermelho) e a saída desejada (azul)	168
Figura 122 - Posicionamento dos sensores de altura na plataforma real com destaque para o <i>Sensor Altura Esquerda</i> (S6) e o <i>Sensor Altura Frente</i> (S7)	169
Figura 123 - Funções de pertinência dos sensores de altura do modelo real	169
Figura 124 - Funções de pertinência das chaves de contato	171
Figura 125 - Percurso realizado na primeira parte do trajeto 1	173
Figura 126 - Dados de orientação e distância ao objetivo durante a realização da primeira etapa do trajeto 1	174
Figura 127 - Dados dos sensores de distâncias frontais (S2, S3 e S4) durante a realização da primeira etapa do trajeto 1	174
Figura 128 - Dados dos sensores de distâncias laterais (S1 e S5) durante a realização da primeira etapa do trajeto 1	175

Figura 129 - Dados dos sensores de distâncias ao meio-fio durante (S6, S7 e S8) a realização da primeira etapa do trajeto 1	175
Figura 130 - Velocidades aplicadas durante a realização da primeira etapa do trajeto 1	176
Figura 131 - Percurso realizado na etapa final do trajeto 1	177
Figura 132 - Dados de orientação e distância ao objetivo durante a realização da etapa final do trajeto 1	177
Figura 133 - Dados dos sensores de distâncias frontais durante a realização da etapa final do trajeto 1	178
Figura 134 - Dados dos sensores de distâncias laterais durante a realização da etapa final do trajeto 1	178
Figura 135 - Dados dos sensores de distâncias ao meio-fio durante a realização da etapa final do trajeto 1	179
Figura 136 - Velocidades aplicadas durante a realização da etapa final do trajeto 1	179
Figura 137 - Percurso realizado no trajeto 2	180
Figura 138 - Dados de orientação e distância ao objetivo durante a realização do trajeto 2	180
Figura 139 - Dados dos sensores de distâncias frontais durante a realização do trajeto 2	181
Figura 140 - Dados dos sensores de distâncias laterais durante a realização do trajeto 2	181
Figura 141 - Dados dos sensores de distância ao meio-fio durante a realização do trajeto 2	182
Figura 142 - Velocidades aplicadas durante a realização do trajeto 2	182
Figura 143 - Percurso realizado no trajeto 3	184

Figura 144 - Velocidades aplicadas durante a realização do trajeto 3	184
Figura 145 - Dados de orientação e distância ao objetivo durante a realização do trajeto 3	185
Figura 146 - Dados dos sensores de distâncias frontais durante a realização do trajeto 3	185
Figura 147 - Dados dos sensores de distâncias laterais durante a realização do trajeto 3	186
Figura 148 - Dados dos sensores de distância ao meio-fio durante a realização do trajeto 3	186
Figura 149 - Posição inicial para a realização do percurso 1	188
Figura 150 - Velocidades aplicadas durante a realização do percurso 1	189
Figura 151 - Dados de orientação ao objetivo durante a realização do percurso	189
Figura 152 - Dados dos sensores de distâncias frontais durante a realização do percurso 1	190
Figura 153 - Dados dos sensores de distâncias laterais durante a realização do percurso 1	190
Figura 154 - Dados dos sensores de altura durante a realização do percurso 1	191
Figura 155 - Posição de parada devido a ruído do sensor de altura frente	192
Figura 156 - Posição final do percurso 1	192
Figura 157 - Posição inicial do percurso 2	193
Figura 158 - Condição de parada do percurso 2 devido a configuração de risco	194

Figura 159 - Condição de parada do percurso 2 devido a configuração de risco, em detalhes	194
Figura 160 - Velocidades aplicadas durante a realização do percurso 2	195
Figura 161 - Dados de orientação ao objetivo durante a realização do percurso 2	195
Figura 162 - Dados dos sensores de distâncias frontais durante a realização do percurso 2	196
Figura 163 - Dados dos sensores de distâncias laterais durante a realização do percurso 2	196
Figura 164 - Dados dos sensores de altura durante a realização do percurso 2	197
Figura 165 - Início do percurso 3	197
Figura 166 - Velocidades aplicadas durante a realização do percurso 3	198
Figura 167 - Dados de orientação ao objetivo durante a realização do percurso 3	198
Figura 168 - Dados dos sensores de distâncias frontais durante a realização do percurso 3	199
Figura 169 - Dados dos sensores de distâncias laterais durante a realização do percurso 3	199
Figura 170 - Dados dos sensores de altura durante a realização do percurso 3	200
Figura 171 - Instante $t = 34s$ do percurso 3	201
Figura 172 - Posição final do percurso 3	201

## Lista de Tabelas

Tabela 1 - Características dos 5 SBCs de desenvolvimento aberto mais populares	85
Tabela 2 – Lista de parâmetros do motor GPB F 006 KM0 60N	129
Tabela 3 - Primeiro conjunto de parâmetros estimados	132
Tabela 4 - Segundo conjunto de parâmetros estimados	133
Tabela 5 - Primeiro conjunto de ganhos para o controlador PID	165
Tabela 6 – Segundo conjunto de ganhos do regulador PID	168

## Lista de Abreviações

CPU - *Central Processing Unit*

TCP/IP - *Transmission Control Protocol / Internet Protocol*

API - *Application Programming Interface*

GPS - *Global Positioning System*

AVD - Atividade de Vida Diária

SUS - Sistema Único de Saúde

FAPERJ - Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro

PUC-Rio - Pontifícia Universidade Católica do Rio de Janeiro

CEPE - Centro de Estudo e Pesquisa do Envelhecimento

QbD - *Quality by Design*

PWM - *Pulse Width Modulation*

LIDAR - *Light Detection and Ranging*

PSD - *Position-Sensitive Device*

CCD - *Charge Coupled Device*

CMOS - *Complementary Metal Oxide Semiconductor*

ADC - *Analog to Digital Converter*

PPR - *Pulses Per Revolution*

MEMS - *Micro-Electro-Mechanical Systems*

ARW - *Angular Random Walk*

NED – *North-East-Down*

IMU - *Inertial Measurement Unit*

SBC - *Single Board Computer*

SPI - *Serial Peripheral Interface*

I2C - *Inter-Integrated Circuit*

SISO - *Single-Input Single-Output*

PID - *Proporcional Integral Derivativo*

MIMO - *Multiple-Input Multiple-Output*

SVD - *Singular Value Decomposition*

PI - *Proporcional Integral*

PD - *Proporcional Derivativo*

SIF - *Sistema de Inferência Fuzzy*

CQA - *Critical Quality Attributes*

CPP - *Critical Process Parameters*

CAD - *Computer Aided Design*

MDF - *Medium Density Fiberboard*

ABS - *Acrylonitrile Butadiene Styrene*

LiPo - *Lítio Polímero*

FRS - *Force Resistor Sensor*

UART - *Universal Asynchronous Receiver / Transmitter*

GPIO - *General Purpose Input/Output*

DIO - *Digital Input / Output*

AIO - *Analog Input / Output*



## Lista de Símbolos

MP - Muito Perto

P - Perto

L - Longe

BX - Baixa

AL - Alta

NG - Negativo Grande

NM - Negativo Médio

Z - Zero

PM - Positivo Médio

PG - Positivo Grande

NA - Negativo Alto

NM - Negativo Médio

NB - Negativo Baixo

Z - Zero

PB - Positivo Baixo

PM - Positivo Médio

PA - Positivo Alto

NF - Normalmente Fechado

# 1

## Introdução

### 1.1

#### Motivação

Com o aumento da expectativa de vida, o perfil etário da população brasileira vem sofrendo uma mudança já vivenciada por países ricos: o seu envelhecimento. Segundo dados do IBGE (2008, 2013; Costa, 2010), projeções indicam que em 2050 a população idosa, ou seja, pessoas com 65 anos ou mais, estará em torno de 25% da população do país. Nessa projeção, a pirâmide etária do Brasil será semelhante à vivenciada por países desenvolvidos, que enriqueceram antes do envelhecimento de sua população. Em entrevista à FAPESP (2008), Alexandre Kalache, médico e pesquisador de saúde pública, alerta sobre a necessidade de políticas públicas voltadas para essa nova realidade, e cita o caso da França, que levou 115 anos para dobrar a proporção de idosos de 7% para 14%, enquanto o Brasil vai dobrar de 9% para 18% em apenas 17 anos.

No guia das Cidades Amigas dos Idosos (OMS, 2007) são descritas várias características que uma cidade deve ter para atender às necessidades de uma população idosa, dentre as quais, pode-se destacar que a cidade deve ser adaptada para as necessidades de mobilidade, tais como: calçadas largas, degraus baixos, boa sinalização, poucos desníveis, entre outros. Contudo, mesmo em uma cidade adaptada, parte da população idosa faz uso de dispositivos de assistência à locomoção, como bengalas e andadores, por possuir alguma deficiência motora, cognitiva, visual, ou mesmo para ter maior conforto e segurança ao se locomover.

O uso de dispositivos de assistência à locomoção proporciona, à parcela da população idosa mais frágil, o aumento de sua capacidade funcional, gerando maior autonomia para locomoção, tendo como consequências diretas o fortalecimento muscular, no caso de dispositivos como andadores e bengalas, e a melhora da qualidade de vida. Esses dispositivos também auxiliam na prevenção de quedas, cujas consequências causam grande impacto para a qualidade de vida da população idosa, dentre as mais impactantes pode-se destacar a hospitalização, a deterioração

funcional, a institucionalização e as restrições às atividades de vida diária (AVDs), geradas de forma direta, devido a fraturas, ou indireta, pelas dores, incapacidade e receios de sofrerem novas quedas (Ribeiro et al., 2008). Dos tipos de fraturas mais graves que podem ocorrer na população idosa, devido a quedas, a fratura de fêmur destaca-se pela gravidade e incidência. Ela é responsável por cerca de 25% das internações no Sistema Único de Saúde (SUS), dos casos de fraturas pós queda, e suas consequências podem reduzir drasticamente a capacidade funcional do idoso lesionado (Louvison e Rosa, 2010).

Pensando na parcela da população idosa que faz uso de algum tipo de dispositivo para auxílio de locomoção, como bengalas e andadores, ou necessita de um acompanhante para auxiliar seus trajetos, foi proposto ao edital 06/2013 da FAPERJ “Apoio ao estudo de temas relacionados à saúde e cidadania de pessoas idosas”, o projeto “Bengala Robótica”, “que visa o desenvolvimento de um robô capaz de guiar idosos ou pessoas com alguma deficiência visual, motora ou cognitiva, em ambientes urbanos. O robô tem como principal objetivo evitar a queda do usuário, ajudando-o a manter-se equilibrado e guiando-o através de obstáculos até seu destino. O projeto recebeu apoio da FAPERJ e está sendo desenvolvido através de uma parceria entre a ThinkTank, a PUC-Rio e o Centro de Pesquisas e Estudos do Envelhecimento (CEPE), e visa atender centenas de milhares de idosos, proporcionando a estes uma maior autonomia”.

Para que o dispositivo realize sua navegação autônoma, ele está equipado com sensores para detectar a presença de obstáculos que ofereçam risco de queda ou desequilíbrio do usuário, além de sensores para orientação e localização, como bússola e GPS. Todo o processamento está embarcado no próprio dispositivo, contudo, para que robô obtenha a rota de navegação, a partir do objetivo estabelecido pelo usuário, foi utilizada uma API do Google Maps para sistema operacional Android. O desenvolvimento desse aplicativo não faz parte do escopo dessa dissertação, mas ele é necessário para que o dispositivo receba as informações de orientação e distância ao objetivo. O aplicativo foi desenvolvido pela equipe da empresa ThinkTank, participante do edital da FAPERJ, e ele se conecta com o dispositivo por meio de uma rede Wi-Fi, criada pelo próprio celular onde o aplicativo está instalado.

## 1.2 Trabalhos relacionados

O estudo e desenvolvimento de tecnologias assistivas para pessoas com dificuldades motores, cognitivas e/ou visuais é uma constante fonte de pesquisa na área de robótica, sendo a população idosa um dos públicos alvos que essa tecnologia busca atender.

Em Di et al. (2012), Figura 1, foi desenvolvida uma bengala robótica composta por um robô móvel omnidirecional, que seria a base da estrutura, conectado a uma haste de alumínio por meio de uma junta universal. O dispositivo está equipado com sensores de torque/força e infravermelho e seus principais objetivos são: prover auxílio na manutenção de uma postura adequada, a prevenção de queda e o auxílio à locomoção em ambientes internos.

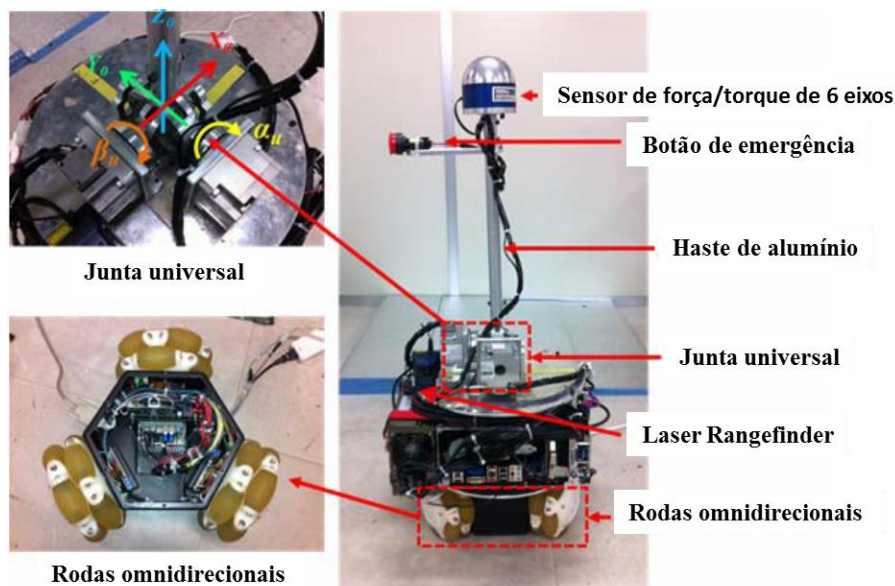


Figura 1 – Protótipo do “Intelligent Cane Robot” (Di et al., 2012)

Essa estrutura é bem semelhante à proposta por Dubowsky et al. (2000), Figura 2. Contudo, a base da bengala da Figura 2 é diferencial e a haste está acoplada à última de forma rígida, tendo na junção haste/base um sensor de força/torque. Dentre os objetivos funcionais dos projetos, as principais diferenças são que, no projeto proposto por Di et al. (2012), a manutenção da postura e equilíbrio do usuário, que são seus principais objetivos, são feitas de forma ativa por intermédio dos atuadores conectados na junção haste/base. Em relação ao projeto de Dubowsky et al. (2000), a postura e equilíbrio do usuário são providos

pelo design da bengala, ou seja, de forma passiva, sendo seus principais objetivos orientar, localizar e monitorar o usuário ao se deslocar em ambientes fechados, como clínicas de repouso e casas de saúde, evitando que o idoso confronte obstáculos que possam levar à queda.

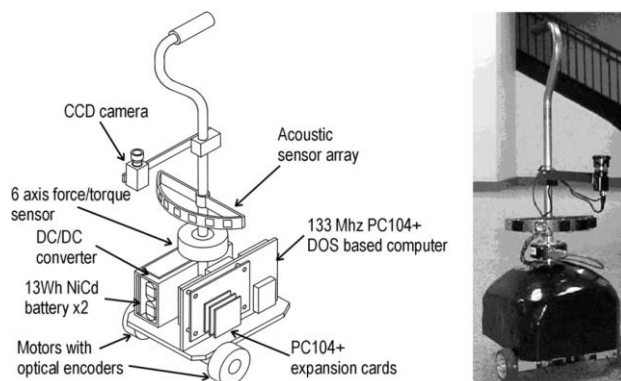


Figura 2 - Dispositivo “SmartCane PAMM System” (Dubowsky et al., 2000)

Outro tipo de estrutura bastante utilizada é a baseada em andadores. Em Wasson et al. (2001), Figura 3, por exemplo, utilizou-se como estrutura um andador comercial. Nos projetos de Zhou et al. (2010), Hirata et al. (2008), Taghvaei et al. (2011) e Ohnuma et al. (2011), Figura 4 a Figura 7, também utilizaram estruturas baseadas em andadores. Apesar de suas estruturas semelhantes, esses projetos apresentam metodologias e abordagens distintas para atuar na assistência à locomoção e prevenção de quedas.



Figura 3 - Dispositivo “Assistive Robotic Walker” (Wasson et al., 2001)

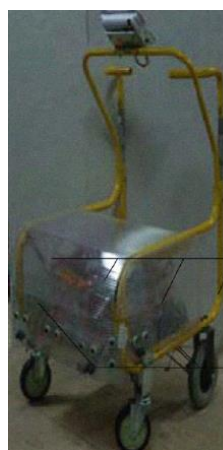


Figura 4 - Protótipo do “ZJU Walker”. (Zhou et al., 2010)

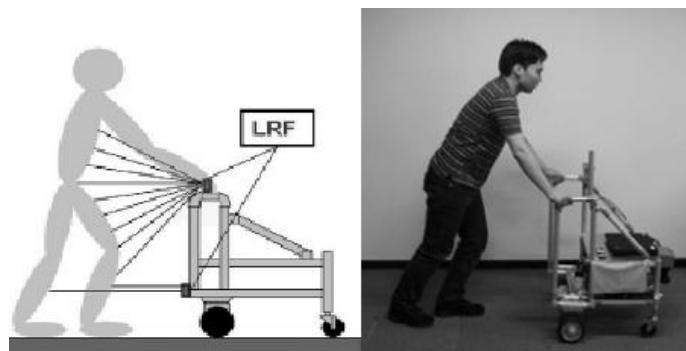


Figura 5 - Protótipo “RT-Walker” com sensor Lidar (Hirata et al., 2008)



Figura 6 - Protótipo “RT-Walker” com sensor Kinect (Taghvaei et al., 2011)

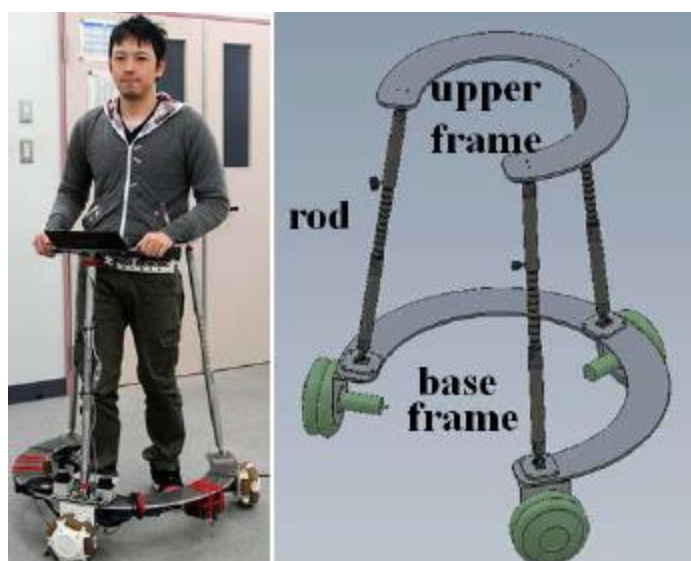


Figura 7 - Protótipo JAIST Active Robotic Walker (Ohnuma et al., 2011)

O trabalho de Wasson et al., (2001), Figura 3, propõe, entre outros objetivos, dosar a autonomia de navegação do dispositivo, baseado na intenção de navegação do usuário, que é estimada a partir dos dados dos sensores de força/torque, presentes nos manetes do andador, dos obstáculos captados pelos sensores de distância e do histórico de atuações, fazendo com que o andador ofereça uma resistência à

locomoção para caminhos que divirjam da navegação autônoma do dispositivo, ou seja, que não forem considerados seguros. O projeto de Zhou et al. (2010), Figura 4, tem uma abordagem bem semelhante ao de Wasson et al. (2001), diferindo-se do último por estar equipado com sensor GPS para locomoção em ambientes externos. Em Hirata et al. (2008) e Taghvaei et al. (2011), Figura 5 e Figura 6, o foco se deu sobre a prevenção de quedas a partir da identificação e classificação das condições de queda e caminhada. Nesses trabalhos foram utilizados, respectivamente, sensores infravermelho e câmeras, além de sensores de força/torque, para classificar a postura do usuário como uma postura de queda, com seus subtipos, ou caminhada, a partir da qual o dispositivo atua para prover o apoio necessário à condição classificada.

Diferente dos projetos apresentados anteriormente, a proposta de Ohnuma et al. (2011), Figura 7, propõe uma estrutura que envolva o usuário, com um design que fornece maior suporte. O objetivo desse projeto foi a implementação de um controle de navegação autônomo que reaja de maneira natural e suave ao caminhar da pessoa. Para tal, foram utilizados sensores infravermelho para observar a movimentação dos pés e quadril do usuário, e aplicado um filtro de partículas (Thrun et al., 2005) para estimar o deslocamento que o usuário aplicará, com base nos dados dos sensores. Portanto, a partir da predição do deslocamento do usuário, o dispositivo irá se movimentar.

Há ainda na literatura dispositivos para assistência à locomoção de pessoas com deficiência visual como Borenstein e Ulrich (1997) e Shim et al. (2004), Figura 8 e Figura 9. Esses dispositivos, apesar de possuírem um público alvo diferente dos apresentados anteriormente, se aproximam muito das bengalas robóticas destinadas à população idosa, devido à aplicação de algoritmos para navegação e desvio de obstáculos.





Figura 8 - Protótipo “GuideCane” (Borenstein e Ulrich, 1997)



Figura 9 - Bengala robótica interativa “RoJi” (Shim et al., 2004)

Com base nos trabalhos apresentados, pode-se observar que, independentemente do tipo de estrutura proposta para o dispositivo ou o tipo de suporte que ele ofereça para o auxílio da postura do usuário, é importante que sua navegação autônoma ocorra de maneira suave e que não “concorra” diretamente com a intenção de deslocamento do usuário. Caso contrário, o próprio dispositivo será o causador do desequilíbrio, ou no pior dos casos, da queda do idoso.

Ainda com relação aos trabalhos apresentados, pode-se observar que a maioria se destina a aplicações em ambientes internos. Ambientes internos, no geral, representam um cenário mais amigável para a navegação autônoma, quando comparados a ambientes externos. Dentre as vantagens que os ambientes internos apresentam pode-se destacar: são normalmente estruturados ou semiestruturados e as condições físicas como luz, temperatura e umidade podem ser controladas. Essas características, assim como outras não citadas, auxiliam no processo de navegação,

pois permitem, por exemplo, estabelecer mapas para a navegação e facilitam o controle de ruídos externos que prejudicariam a leitura dos sensores.

A seguir serão apresentados os objetivos dessa dissertação de mestrado, assim como sua estrutura e organização.

### **1.3 Objetivos**

Essa dissertação de mestrado tem como objetivo o desenvolvimento e construção do protótipo de um dispositivo robótico para auxílio à locomoção de pessoas idosas em ambientes urbanos. O dispositivo em questão, cuja estrutura é semelhante à de um andador, deverá ser capaz de se locomover de maneira autônoma em ambientes desconhecidos e não estruturados, tendo ainda a possibilidade de receber atuações diretas do usuário.

A construção desse protótipo inclui a especificação de sensores, atuadores, hardware processador e a aplicação de algoritmos de controle e processamento de sinais, com o intuito de obter uma plataforma móvel que ofereça segurança, robustez e a dinâmica necessária para a tarefa de guiar idosos ou pessoas com alguma deficiência visual, motora ou cognitiva, em ambientes urbanos para um determinado destino. Dessa forma, o ambiente de utilização da bengala robótica é externo e não estruturado havendo, portanto, obstáculos dinâmicos e desconhecidos no ambiente.

Os objetivos secundários são:

- Implementação dos algoritmos de controle e navegação baseados em Lógica Fuzzy, com o intuito de se obter uma rápida resposta e uma generalização para diferentes ambientes, necessária para a atuação em ambientes urbanos, que além de suas irregularidades são não estruturados.
- Aplicação dos algoritmos nos moldes da arquitetura do Framework de robótica Player (Gerkey e contribuidores, 2010). Foram escritos drivers (programas), para a adequação dos dados dos sensores e atuadores para o encapsulamento da ferramenta Player. Essa arquitetura foi utilizada porque oferece uma abstração de hardware que irá permitir a realização dos testes em ambiente físicos e virtuais

sem que haja necessidade de grandes mudanças no software. Além de permitir futuras migrações de hardware processador sem que haja impacto no software de controle escrito, a interface Player também será utilizada para gerenciar a comunicação do aplicativo da API do Google Maps com a Bengala Robótica.

- Especificação dos processadores, sensores e atuadores, além de embarcar todo o software em uma plataforma processadora móvel capaz de executar todas as etapas do gerenciamento do dispositivo em tempo hábil para uma execução segura para o usuário.

## 1.4

### Descrição do Trabalho

Este trabalho foi elaborado seguindo as seguintes etapas:

- Estudo de trabalhos relacionados;
- Especificação do processador, sensores e atuadores;
- Desenvolvimento da lógica de controle, aplicação e ajustes dos algoritmos na plataforma simulada pelo software de simulação de robôs Stage (Vaughan e contribuidores, 2011). Na plataforma de simulação, foi instanciado um robô móvel com características semelhantes ao protótipo planejado e foram feitas simulações de navegação autônoma em um ambiente que simule ruas e calçadas;
- Implementação dos drivers da plataforma Player para os sensores e atuadores especificados; e
- Montagem, aplicação e testes em uma plataforma real.

Na primeira etapa do trabalho, foi feito um levantamento bibliográfico referente a diferentes dispositivos robóticos para auxílio à locomoção de pessoas idosas e deficientes visuais, pois em muitos pontos há interseção entre os dispositivos.

A etapa de especificação da plataforma para processamento, sensores e atuadores foi feita com base no estudo dos trabalhos relacionados e dos requisitos obtidos pela pesquisa realizada em parceria com a equipe do CEPE, que avaliou os idosos.

Na etapa de desenvolvimento da lógica de controle foram consideradas as características dos sensores especificados para a plataforma real. Essas características foram utilizadas na implementação do dispositivo robótico virtual que foi simulado na plataforma Stage. Esta etapa foi realizada em ambiente virtual para ajustes e testes dos algoritmos de controle e navegação, permitindo que ele evoluísse de maneira mais rápida e segura. Nesse processo foi avaliada a capacidade de manobra e a navegação do algoritmo de controle

Na etapa de programação dos drivers foram escritos os programas que acessam os dados dos sensores e atuadores reais e os encapsulam para estrutura do framework de robótica Player. Com a conclusão dessa etapa, a abstração do hardware, proporcionada pelo software Player, fica completa. Portanto, os algoritmos de navegação utilizados na simulação podem ser aplicados diretamente na plataforma real.

Na última etapa foi construída uma plataforma real para testes, onde foram embarcados todos os sensores e atuadores para que se pudesse realizar os testes e ajustes do código. Nessa etapa foram realizados testes em diferentes configurações para aferir a capacidade do algoritmo de controle.

## **1.5**

### **Organização da Dissertação**

Esta dissertação está organizada em mais cinco capítulos, descritos a seguir.

No Capítulo 2 é apresentada a fundamentação teórica dessa dissertação. Essa fundamentação está subdividida em: conceitos de mecânica, com a parte estrutural e equações cinemáticas do dispositivo; conceitos de eletrônica, onde aborda toda a parte de sensores, atuadores e hardware processador; software, onde é feita uma introdução ao Player, Stage e o aplicativo para plataforma Android; e controle, onde é realizada uma introdução a teoria de controle com ênfase nas técnicas empregadas nessa dissertação.

O Capítulo 3 apresenta a implementação virtual do dispositivo, mostrando o desenvolvimento do controle de navegação e a configuração da ferramenta de simulação para os testes a serem apresentados no Capítulo 5.

O Capítulo 4 apresenta a implementação real do dispositivo. Nesse capítulo são apresentados todos os sensores, atuadores e o hardware processador utilizados, suas configurações e os drivers da ferramenta Player que foram desenvolvidos.

No Capítulo 5 são apresentados os testes realizados na plataforma virtual e real para diferentes condições.

Finalmente, no Capítulo 6 estão descritas as conclusões e os trabalhos futuros.

## 2 Fundamentos teóricos

Nesse capítulo serão apresentados os fundamentos teóricos utilizados no desenvolvimento do dispositivo proposto nessa dissertação. Esses fundamentos foram separados em conceitos sobre mecânica, eletrônica, software e controle.

### 2.1 Projeto Mecânico

Nessa seção serão apresentadas algumas das características físicas da estrutura do protótipo e o seu modelo cinemático.

#### 2.1.1 Design do protótipo

O design de um produto é de grande importância para a sua aceitação no mercado. Mesmo nas etapas iniciais de um protótipo, as características do design devem ser consideradas para que não sejam necessárias grandes correções nas etapas finais do projeto. Quando se trata de um protótipo como para o projeto “Bengala Robótica”, cujo objetivo é oferecer suporte e segurança para um público de saúde frágil ou idade avançada (idosos), o estudo da ergonomia, funcionalidade e aspectos de segurança se torna mais crítico.

Devido à complexidade, abrangência e multidisciplinaridade desse tema, ele foi objeto de estudo de Rivero (2015; Rivero et al., 2015a, 2015b), cujo objetivo foi a adaptação e implementação da técnica de *Quality by Design* (QbD) (Lionberger et al., 2008; Nadpara et al., 2012; DPT Laboratories, 2013) no desenvolvimento do protótipo do projeto “Bengala Robótica”.

Com base nos resultados desse estudo, foram estabelecidas algumas funcionalidades e condições de contorno, como peso, velocidade e tamanho, que o protótipo deve ter e, a partir desses dados, foram especificados os componentes mecânicos e eletrônicos utilizados.

Para o desenvolvimento da “Bengala Robótica” foi escolhida uma estrutura baseada em um andador, onde a base é um robô móvel diferencial com uma roda castor para apoio. O design semelhante a um andador foi adotado por oferecer maior estabilidade ao usuário do que uma estrutura tipo bengala. Já a arquitetura de robô diferencial foi adotada por permitir uma construção e atuação mais simples, quando comparado a uma estrutura omnidirecional, por exemplo, que teria como vantagem proporcionar maior manobrabilidade ao dispositivo. Outra vantagem da adoção de uma estrutura diferencial, em relação à omnidirecional, é que suas rodas paralelas facilitam a transposição de pequenos obstáculos, como desníveis entre calçadas.

### 2.1.2 Modelo Cinemático

Conforme descrito anteriormente, para construção da base do dispositivo robótico foi utilizado um robô móvel com acionamento diferencial e uma roda tipo castor para apoio, ilustrado na Figura 10.

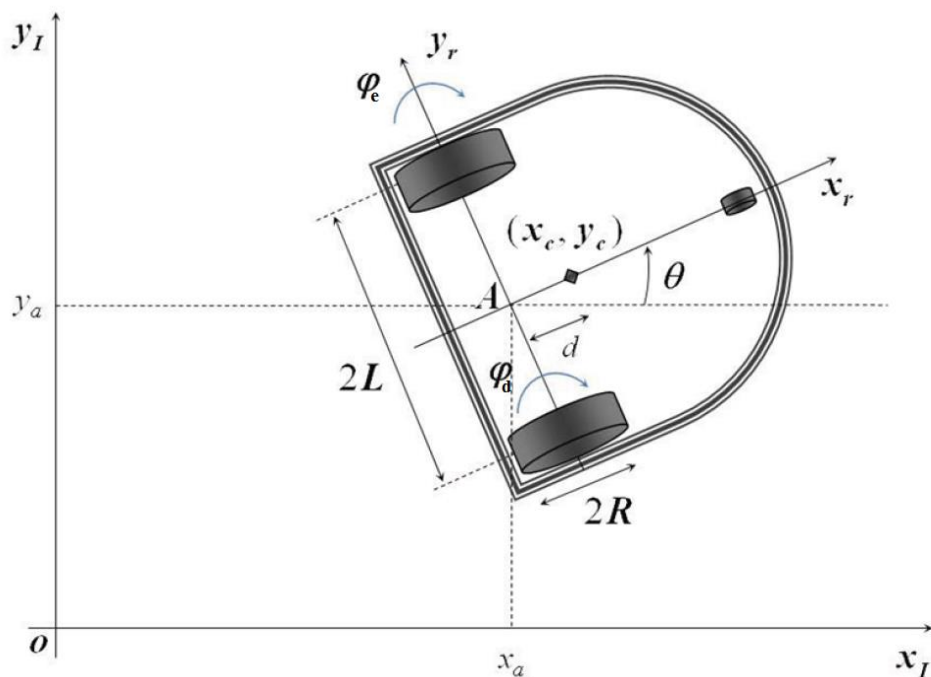


Figura 10 - Diagrama da base de um robô diferencial (Dhaouadi e Hatab, 2013)

Na Figura 10, os eixos  $(x_I, y_I)$  representam o sistema de coordenadas inercial, ou seja, o sistema fixo global no qual o robô se movimenta, os eixos  $(x_r, y_r)$  representam o sistema de coordenadas do robô, ou seja, o sistema local que é atrelado ao robô (eixo das rodas ativas), A é o centro de rotação do robô,  $\theta$  é o

deslocamento angular do robô,  $\varphi_e$ ,  $\varphi_d$  são as posições angulares das rodas esquerda e direita do robô,  $(x_c, y_c)$  é a posição do centro de massa do robô em relação ao sistema de coordenadas do robô,  $d$  é a distância do eixo de rotação do robô ao centro de massa,  $R$  é o raio das rodas e  $L$  representa a distância das rodas ao eixo de rotação.

A relação entre o sistema inercial e o sistema de coordenadas do robô (Dhaouadi e Hatab, 2013) é obtida por

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}^I = R(\theta) * \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}^r, \quad (1)$$

onde  $R(\theta)$  é uma matriz de rotação ortogonal, definida por

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Para simplificar o equacionamento do robô, foi negligenciada a rotação do suporte da roda castor, sendo ela considerada apenas como um simples ponto de apoio, que não oferece nenhum atrito ou restrição ao movimento da base. Também foram assumidas duas restrições não-holonômicas<sup>1</sup>: a de rolamento puro das rodas, ou seja, cada roda mantém apenas um ponto de contato com o solo, não havendo deslizamento no eixo longitudinal,  $x_r$ , e, portanto, as velocidades lineares das rodas direita e esquerda podem ser obtidas conforme as equações (3) e (4), respectivamente; e a condição de não haver derrapagem no eixo ortogonal  $y_r$ , ou seja, a componente  $y$  da velocidade linear do centro de rotação,  $A$ , no referencial do robô,  $r$ , é nula (equação (5)).

$$v_d = \dot{\varphi}_d * R \quad (3)$$

$$v_e = \dot{\varphi}_e * R \quad (4)$$

---

<sup>1</sup> Sistema com dimensão finita onde algum tipo de restrição é imposta a um ou mais estados do sistema (Figueiredo e Jota, 2004).



$$\dot{y}_A^r = 0 \quad (5)$$

A partir das equações (3) a (5), as velocidades linear e angular no eixo de rotação do robô, em relação às coordenadas do robô, podem ser obtidas por

$$\dot{x}_A^r = v_A = R * \frac{(\dot{\phi}_d + \dot{\phi}_e)}{2} = \frac{(v_d + v_e)}{2}, \quad (6)$$

$$\dot{y}_A^r = 0 \quad (7)$$

e

$$\dot{\theta} = \omega = R * \frac{(\dot{\phi}_d - \dot{\phi}_e)}{2 * L} = \frac{(v_d - v_e)}{2 * L}, \quad (8)$$

onde  $\dot{x}_A^r$  e  $\dot{y}_A^r$  são as componentes  $x$  e  $y$  da velocidade linear do centro de rotação do robô no referencial do mesmo,  $v_d$  e  $v_e$  são as velocidades linear das rodas direita e esquerda e  $L$  é a distância das rodas ao centro de rotação.

As equações de (6) a (8) podem ser reescritas em sua forma vetorial conforme a equação (9), cuja representação em relação ao sistema de coordenadas inercial é obtida pela equação (10).

$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \theta \end{bmatrix}^r = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} * \begin{bmatrix} \dot{\phi}_d \\ \dot{\phi}_e \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \theta \end{bmatrix}^r = \begin{bmatrix} \frac{R}{2} \cos(\theta) & \frac{R}{2} \cos(\theta) \\ \frac{R}{2} \sin(\theta) & \frac{R}{2} \sin(\theta) \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} * \begin{bmatrix} \dot{\phi}_d \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (10)$$

Por último, a relação entre as velocidades do centro de rotação, ponto A, e o centro de massa, ponto C, pode ser obtida por

$$\dot{x}_C = v * \cos(\theta) - d * \sin(\theta) * \omega, \quad (11)$$

$$\dot{y}_c = v * \sin(\theta) + d * \cos(\theta) * \omega \quad (12)$$

e

$$\dot{\theta} = \omega. \quad (13)$$

## 2.2

### Projeto eletrônico

Nessa seção serão apresentados os conceitos básicos dos atuadores e sensores normalmente utilizados em tarefas de navegação e em projetos de dispositivos semelhantes ao desenvolvido nessa dissertação. Esses conceitos foram a base para escolha dos sensores e atuadores aplicados no protótipo. Nos capítulos seguintes, 3 e 4, serão feitas as especificações dos tipos e modelos dos sensores e atuadores utilizados, baseados nos conceitos que serão apresentados a seguir.

#### 2.2.1

##### Atuadores

A escolha do atuador é um fator importante para o dimensionamento de um projeto de robótica, pois irá influenciar na carga que o dispositivo poderá suportar, além de seu peso e autonomia. O tipo do atuador depende da tarefa que se deseja executar e pode ser classificado em dois grupos principais: atuadores de manipulação e atuadores de locomoção (Romero et al., 2014).

Os atuadores de manipulação são, por exemplo, pistões, braços e garras robóticas. Esses atuadores podem ser construídos com: motores de passo e servomotores, para aplicações que necessitam de maior precisão; motores/pistões pneumáticos, para aplicações de força e velocidade; e motores/pistões hidráulicos, para aplicações de alta carga/torque.

Os atuadores de locomoção são comuns a todo robô móvel sendo, para robôs móveis baseados em rodas ou esteiras, o tipo mais utilizado o motor CC (Romero et al., 2014). A arquitetura de motor CC empregado na tarefa de locomoção está relacionada com o ambiente e a forma com que o robô irá se deslocar. Por exemplo, robôs terrestres, movidos por rodas ou esteiras, tipicamente utilizam motores CC com escovas, enquanto robôs aéreos, como quadrimotores, utilizam-se de motores CC sem escovas como fonte de propulsão.

A seguir, será apresentada uma breve descrição dos tipos e características dos motores CC com e sem escovas.

### **2.2.1.1 Motor CC**

Motores CC são largamente utilizados em aplicações de robôs móveis devido à mobilidade proporcionada por sua fonte de energia, a bateria, e a melhor relação custo-benefício, quando comparados a motores a combustão, por exemplo. Os motores CC são classificados em dois tipos principais, com e sem escovas, e sua modelagem é um tema já bastante abordado na literatura (Mohan et al., 2003; Hanselman, 2006).

Os motores CC com escovas são assim chamados, pois fazem uso de escovas para fazer o contato da fonte de energia com as bobinas da armadura. Esse contato ocorre por meio do disco de comutação que alterna qual bobina da armadura é energizada à medida que o rotor gira. Existem cinco tipos principais de motores CC com escovas, que caracterizam como o campo magnético estacionário do estator, parte fixa do motor, é gerado. As configurações são: excitação independente, série, paralela, mista e ímã permanente (Electrical4u, n.d.). Cada uma das configurações acima apresenta diferentes características de torque, tensão e controle de velocidade (Mohan et al., 2003; Marques et al., 2013; Siemens, s.d.; WEG Indústrias S.A., s.d.).

A configuração de excitação independente tem como circuito equivalente a Figura 11. Nessa configuração o campo do estator é gerado por bobinas alimentadas por uma fonte de tensão independente da utilizada na armadura. O controle de velocidade pode ser feito variando-se o campo do estator, por meio da variação da corrente de suas bobinas, ou pela variação da tensão da armadura. Mantendo-se o fluxo magnético do estator constante, e variando-se a tensão da armadura, tem-se um controle de velocidade com o torque máximo desenvolvido. Já mantendo a tensão da armadura constante e variando-se o campo de estator, obtêm-se velocidades acima da nominal do motor ao custo da redução do torque máximo gerado.

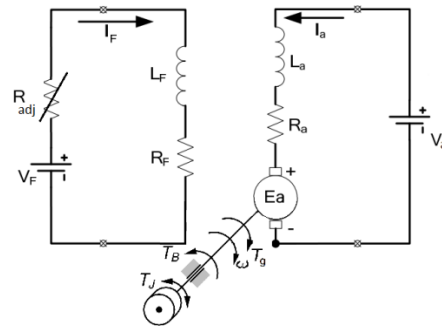


Figura 11 - Circuito equivalente do Motor CC com escovas de excitação independente

Na configuração de excitação série, a bobina do estator é ligada em série à fonte de tensão da armadura (Figura 12), fazendo com que a corrente que flui através da armadura seja a mesma que gera o campo do estator. Essa configuração tem como característica fornecer torques de saída elevados a um custo moderado de corrente. Contudo, operações com baixa carga podem gerar velocidades excessivamente altas, tornando o controle de velocidade mais complexo.

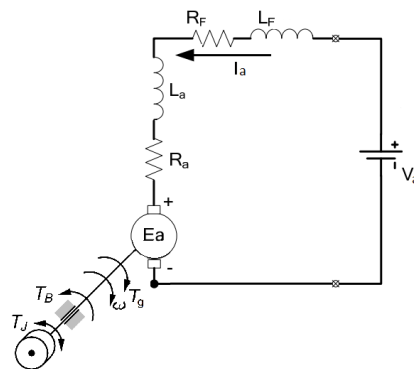


Figura 12 - Circuito equivalente do Motor CC com escovas de excitação série

A configuração paralela é uma das mais utilizadas em motores CC com escovas (Figura 13). Essa configuração, assim como a série, tem o controle de velocidade a partir da variação da tensão da armadura. Contudo, diferente da configuração série, a paralela apresenta uma baixa variação de velocidade devido à variação do torque da carga.

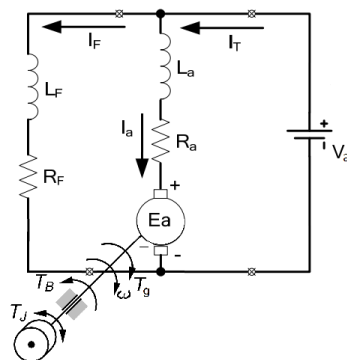


Figura 13 - Circuito equivalente do Motor CC com escovas de excitação paralela

A configuração composta, ilustrada na Figura 14, é uma combinação da configuração série com a paralela. Essa configuração apresenta um comportamento intermediário entre as duas configurações bases. Portanto, apresenta um bom torque de saída e consegue manter uma velocidade constante para aplicações que exijam baixo torque de saída.

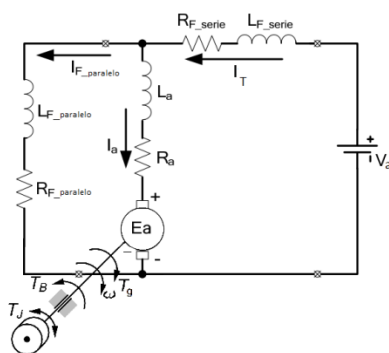


Figura 14 - Circuito equivalente do Motor CC com escovas de excitação composta

Na configuração com ímã permanente (Figura 15) o campo magnético do estator é gerado por ímãs permanentes presos ao seu interior sendo, portanto, constante. Essa configuração é muito utilizada, devido ao baixo custo, consumo e tamanho compacto. Outra característica desse tipo de construção é a relação linear entre o torque e a velocidade. Nessa relação, o torque máximo é obtido com rotação zero e a máxima rotação é alcançada quando o motor gira sem carga. Uma desvantagem dessa arquitetura é não alcançar valores elevados de torque de saída, quando comparado às outras arquiteturas apresentadas.

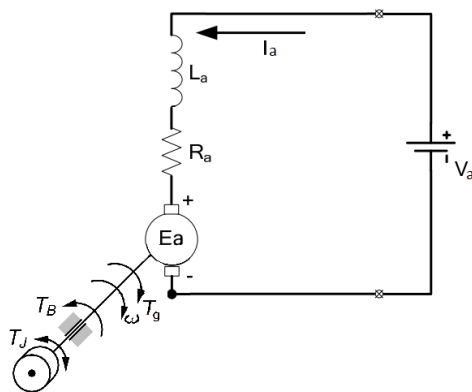


Figura 15 - Circuito equivalente do Motor CC com escovas de excitação por ímãs permanentes

Os motores sem escovas, *brushless*, têm uma armadura fixa alimentada diretamente por um circuito eletrônico, não fazendo uso de escovas. Esse tipo de motor, tipicamente, possui três bobinas em sua armadura (Figura 16). Essas bobinas estão posicionadas na parte fixa do motor e são cercadas por ímãs permanentes que estão presos ao rotor. O controle de velocidade dos motores *brushless* ocorre por meio da comutação das bobinas da armadura. Essa comutação é realizada por um circuito eletrônico que identifica, por meio de sensores de campo, a posição do rotor para acionar as bobinas de maneira que gerem forças eletromagnéticas de repulsão e/ou atração dos polos do rotor, de acordo com o sentido e velocidade de rotação desejados (Hanselman, 2006; ON Semiconductor, 2014; Trindade, 2009).

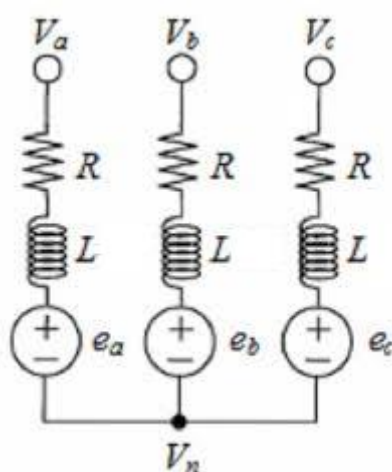


Figura 16 - Circuito equivalente de motor Brushless de três bobinas (Trindade, 2009)

Motores *brushless* oferecem grandes vantagens em relação aos motores com escova. Entre as vantagens destacam-se: maior relação torque/peso, torque/potência, maior eficiência energética, maior vida útil, eliminação de possíveis faíscas que ocorrem nos motores com escova devido ao atrito da escova com o disco de comutação, além de serem mais silenciosos. A principal desvantagem de motores *brushless*, em relação a motores com escova, é o custo mais elevado, principalmente para motores de alto torque de saída.

Nesse trabalho, foi utilizado um motor com escovas, devido ao menor custo, cujo campo do estator é gerado por ímãs prementes. O circuito equivalente dessa configuração pode ser observado na Figura 15 e sua modelagem pode ser descrita conforme as equações (14) a (22) (Mohan et al., 2003; Movellan, 2010).

$$v_{in}(t) = R_a * i_a(t) + L_a * \frac{di_a(t)}{dt} + E_a(t) \quad (14)$$

$$E_a(t) = K_e \omega(t) \quad (15)$$

$$T_g(t) = K_t * i_a(t) \quad (16)$$

$$P_g(t) = T_g(t) * \omega(t) \quad (17)$$

$$T_g(t) = T_L(t) + T_J(t) + T_B(t) \quad (18)$$

$$T_J(t) = J * \frac{d\omega(t)}{dt} \quad (19)$$

$$T_B(t) = B * \omega(t) \quad (20)$$

$$\tau_e = L_a / R_a \quad (21)$$

$$\tau_m = J / B \quad (22)$$

Nas equações acima,  $v_{in}$  é a tensão de entrada,  $R_a$ ,  $i_a$  e  $L_a$  são a resistência, a corrente e a indutância do circuito da armadura do motor,  $E_a$  é a força contra eletromotriz gerada pela armadura,  $K_e$  é a constante de força contra eletromotriz,  $\omega$  é a velocidade angular do motor,  $T_g$  é o torque gerado pelo motor,  $K_t$  é a constante de torque do motor,  $P_g$  é a potência mecânica gerada,  $T_L$  é o torque exigido pela carga,  $T_f$  é o torque devido à inércia do motor,  $J$  é a constante de inércia do motor,  $T_B$  é o torque devido ao atrito viscoso,  $B$  é a constante de amortecimento viscoso do motor,  $\tau_e$  é a constante de tempo elétrica do motor e  $\tau_m$  é a constante de tempo mecânica do motor.

Com o intuito de obter as funções de transferência do sistema, aplicando-se a transformada de Laplace nos dois lados das equações (14) e (18), obtêm-se as equações (23) e (24), por meio das quais é possível representar o modelo do motor CC através do diagrama de blocos da Figura 17.

$$V_{in}(s) = R_a * I_a(s) + L_a * I_a(s) * s + K_e \Omega(s) \quad (23)$$

$$K_t * I_a(s) = T_L(s) + J * \Omega(s) * s + B * \Omega(s) \quad (24)$$

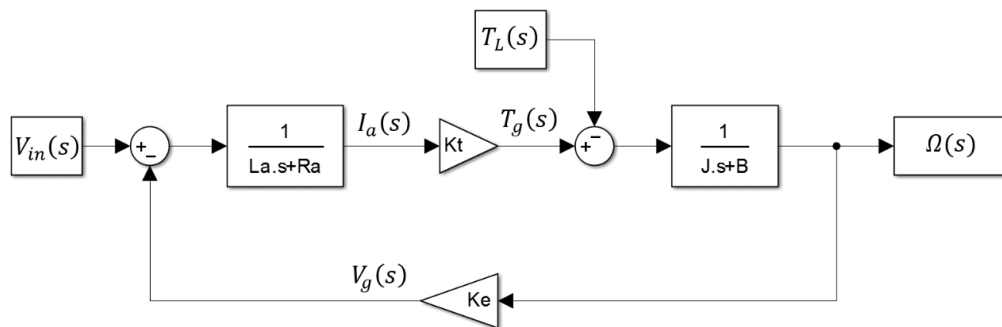


Figura 17 - Diagrama em blocos de um motor CC com escovas e excitação por ímãs permanentes

O diagrama acima representa um sistema com duas entradas, tensão da armadura e torque da carga, e uma saída, velocidade angular do rotor. Esse diagrama pode ser representado a partir de duas funções de transferência,  $H_1(s)$  e  $H_2(s)$ , conforme as equações (25) a (27).



$$\Omega(s) = H_1(s) * V_{in}(s) + H_2(s) * T_L(s) \quad (25)$$

$$H_1(s) = \left. \frac{\Omega(s)}{V_{in}(s)} \right|_{T_L(s)=0} = \frac{K_t}{(L_a * s + R_a)(J * s + B) + K_t K_e} \quad (26)$$

$$H_2(s) = \left. \frac{\Omega(s)}{T_L(s)} \right|_{V_{in}(s)=0} = - \frac{(L_a * s + R_a)}{(L_a * s + R_a)(J * s + B) + K_t K_e} \quad (27)$$

$H_1(s)$  é a função de transferência da velocidade angular em função da tensão de entrada quando o torque da carga é nulo e  $H_2(s)$  é a função de transferência da velocidade angular em função do torque da carga quando a tensão de entrada é nula.

### 2.2.1.2 Caixa de Redução

Redutores ou caixas de redução são normalmente utilizados para se obter uma maior relação de torque ou para adequar a velocidade do atuador à tarefa a ser realizada. Um redutor consiste em um conjunto de eixos com engrenagens cilíndricas de dentes, retos, cônicos e/ou helicoidais ou apenas uma coroa com parafuso sem fim (Andrade, n.d.). Dependendo dos tipos de engrenagens utilizadas na construção do redutor ele pode ser classificado como, por exemplo, transmissão de engrenagens helicoidais, transmissão de eixos paralelos ou transmissão de rosca sem fim (Siemens, 2008).

Independentemente dos tipos de engrenagens utilizadas na construção de um redutor, a relação entre os torques de entrada e saída está diretamente ligada à redução final do dispositivo, ou seja, à relação entre a rotação de entrada e saída, e pode ser obtido por

$$i = \frac{n_1}{n_2} = \frac{M_2}{M_1}, \quad (28)$$

onde  $n_1$  e  $M_1$  são a rotação e o torque de entrada,  $n_2$  e  $M_2$  são a rotação e o torque de saída.

Por meio da relação de transmissão também é possível obter a inércia do eixo de saída do motor devido às cargas suportadas pelo eixo de saída do redutor (equação (29)), considerando uma eficiência de 100% da redução.

$$J_{eq} = J_M + J_L * \left(\frac{n_2}{n_1}\right)^2, \quad (29)$$

onde  $J_M$  é a inércia de saída do motor desconectado da redução,  $J_{eq}$  é a inércia total enxergada pelo eixo de saída do motor e  $J_L$  é a soma dos momentos de inércia sentidos pelo eixo de saída do redutor.

Por último, outra característica importante que deve ser definida de um redutor é sua eficiência, dada por

$$\eta_r = \frac{P_{redutor}}{P_{motor}} = \frac{\tau_{redutor}}{\tau_{motor} * i}, \quad (30)$$

onde  $P_{redutor}$  e  $\tau_{redutor}$  são a potência mecânica e o torque de saída do redutor,  $P_{motor}$  e  $\tau_{motor}$  são a potência mecânica e o torque de saída do motor e  $i$  é a relação de redução.

### 2.2.1.3 Driver de Potência

Para que se possa realizar o controle de velocidade de um motor CC com ímãs permanentes é necessário realizar uma variação na tensão da armadura. Existem diversos circuitos comerciais que realizam essa tarefa, cuja essência é conjugar dois circuitos básicos: um regulador de tensão de saída ajustável e uma ponte H.

O controle de velocidade de motores CC normalmente é feito pelo chaveamento da tensão da armadura. Esse chaveamento é feito por um sinal PWM (*Pulse Width Modulation*) que habilita a condução de circuito regulador de acordo com o seu ciclo de trabalho, ou seja, a razão entre a largura de pulso e a frequência do sinal. O valor de tensão média aplicada aos terminais da armadura é obtido pelo produto do ciclo de trabalho com a tensão da bateria, assumindo que não há perda do circuito comutador. Esse tipo de regulador por chaveamento, conhecidos como CC-CC, normalmente utiliza filtros para remover as ondulações provenientes da

comutação do sinal. Contudo, no caso dos motores CC, tais componentes podem ser desprezados, pois a indutância da armadura atua como um filtro de baixas frequências, mantendo o sinal da corrente da armadura estável (Mohan et al., 2003; Microchip Technology Inc., 2004).

O circuito ponte H, ilustrado na Figura 18, é responsável por alterar a polaridade da tensão que alimenta o motor. Essa comutação é feita alterando os estados das chaves s1 a s4. O circuito ponte-H normalmente é construído replicando as chaves s1 a s4 por transistores, sejam eles na forma de componente discretos, como TBJs e MOSFET, ou com toda a estrutura encapsulada em um circuito integrado (Mohan et al., 2003).

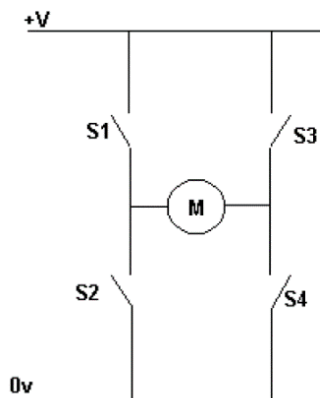


Figura 18 - Representação de um circuito ponte-H

Para especificar um driver para motor CC, deve-se conhecer as características de operação do motor, como o torque máximo exigido e as velocidades máximas com e sem a carga. Esses fatores, como podem ser observados nas equações (14) a (22), irão influenciar na tensão, corrente e potência que o dispositivo deverá ser capaz de fornecer para o motor.

### 2.2.2 Sensores

É fundamental, em todo dispositivo robótico, a existência de sensores com os quais o robô irá receber informações do ambiente no qual está inserido, para então interagir com o mesmo, por intermédio de seus atuadores. A natureza dos sensores utilizados em projetos de robótica está diretamente relacionada com a tarefa e/ou ambiente no qual o robô está inserido (Borenstein et al., 1996).

Em uma tarefa de detecção de obstáculos, por exemplo, sensores de diferentes naturezas podem ser usados, tais como: sensores de luz, som, contato e visão. Cada um dos tipos de sensores apresenta vantagens e desvantagens que serão acentuadas dependendo do ambiente e tipo de tarefa a ser executada. Uma prática comum na utilização de sensores em dispositivos robóticos é a combinação de seus dados para obter uma estimativa melhor da informação desejada. Essa prática é conhecida como fusão sensorial (Thrun et al., 2005) e, normalmente, utiliza dados de sensores de diferentes naturezas.

Conforme mostrado na revisão bibliográfica apresentada no capítulo 1, dispositivos robóticos para assistência à locomoção utilizam diversos tipos de sensores para extrair diferentes informações do ambiente e do usuário. Alguns dos sensores mais comuns são: sensor de distância, *encoder* rotativo, giroscópio, acelerômetro, bússola, sensor de força e sensor de torque. Além dos sensores, é importante que o dispositivo robótico tenha capacidade e velocidade de processamento suficientes para que possa realizar suas tarefas de leitura, tratamento e execução em tempo hábil.

A seguir, nas seções 2.2.2.1 a 2.2.3, será apresentada uma breve descrição dos tipos de sensores mais utilizados e alguns hardwares processadores que se enquadram na proposta do protótipo do dispositivo desenvolvido nessa dissertação.

#### **2.2.2.1**

##### **Sensores de detecção de obstáculo**

A seguir, nas subseções 2.2.2.1.1 a 2.2.2.1.3, será apresentada uma breve descrição dos principais tipos de sensores utilizados para detecção de obstáculos.

#### **2.2.2.1.1**

##### **Sensor de Ultrassom**

Sensores do tipo ultrassom são largamente utilizados em aplicações robóticas, principalmente devido a seu baixo custo e fácil leitura de seus dados. Esse tipo de sensor tem como princípio básico de funcionamento a medição do tempo de voo, ou seja, o tempo que uma onda sonora leva para refletir e retornar ao receptor do sensor (Siegwart e Nourbakhsh, 2004). A distância,  $d$ , do objeto que gerou a

reflexão pode ser calculada a partir da velocidade de propagação do som no meio,  $c$ , e do tempo de voo,  $t$ , conforme a equação (31).

$$d = \frac{c \cdot t}{2}, \quad (31)$$

onde  $c \cong 343m/s$  para o ar a  $20^{\circ}C$ .

Para realizar a medição de distância, o sensor de ultrassom emite uma sequência de pulsos sonoros, formando um pacote de transmissão. Com o envio do pacote concluído, o temporizador para medir o tempo de voo é iniciado e, simultaneamente, é estipulado um limiar para detectar ecos válidos. Esse limiar é decrementado à medida que o tempo passa, para compensar o decaimento da amplitude do eco esperado, conforme ilustrado na Figura 19.

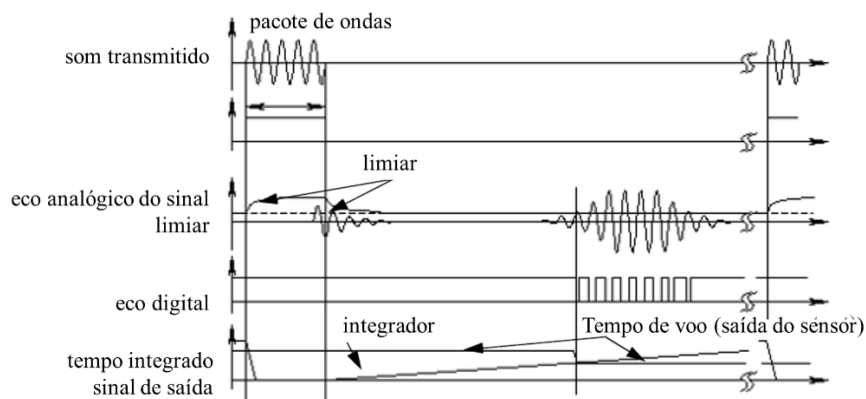


Figura 19 - Diagrama de tempo de um sensor de ultrassom (Siegwart e Nourbakhsh, 2004)

Sensores de ultrassom tipicamente operam com uma emissão de ondas sonoras em frequências entre 40 kHz – 180 kHz e um ângulo de abertura de 20 a 40 graus. A qualidade da informação de distância do objeto detectado depende de alguns fatores como: ângulo de abertura do sensor, características físicas do objeto detectado, temperatura do ambiente de operação, efeito *cross-talk* e reflexão de canto (Romero et al., 2014; Siegwart e Nourbakhsh, 2004; Wielenberg, 2015).

- *Ângulo de abertura do sensor*: sensores de ultrassom não conseguem, diretamente, determinar a posição do objeto dentro de seu campo de atuação, apenas a distância do objeto refletido. Portanto, ao detectar um objeto, sabe-se apenas sua distância dentro de uma região que

depende do campo de atuação do sensor. Quanto maior o campo de visão, mais fácil será detectar um objeto, contudo, menor será a certeza da posição em que ele se encontra nesse campo.

- *Características do objeto*: objetos que possuem baixa reflexão a ondas sonoras podem levar o sensor a realizar medições de distância com baixa exatidão. Outros fatores que influenciam são a geometria do objeto e o ângulo do eixo do sensor em relação ao objeto. Objetos finos exigem um ângulo mais fechado do sensor para que possam ser detectadas, caso contrário, as ondas sonoras não são refletidas para o receptor do sensor. O mesmo pode ocorrer com objetos que se encontram em um ângulo muito aberto em relação à fonte emissora do sensor.
- *Temperatura*: influencia na velocidade de propagação do som no ar. Com isso, caso o sensor não possua uma calibração para essa grandeza, valores de temperatura distantes da referência do sensor irão gerar erros de leitura.
- *Cross-talk*: esse efeito ocorre quando a onda sonora emitida por um sensor é captada por outro que esteja próximo, gerando leituras erradas.
- *Reflexão de canto*: esse efeito é gerado quando a onda sonora atinge duas superfícies lisas que formam um ângulo próximo a 90 graus ou uma superfície que forma um semicírculo (Figura 20), o que gera reflexões em regiões fora do padrão de captação do sensor. Esse efeito pode levar a problemas como *cross-talk*.

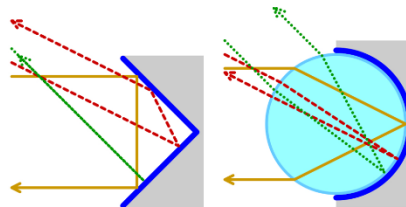


Figura 20 - Ilustração de dois tipos de superfícies que causam reflexão de canto  
(Wielenberg, 2015)

### 2.2.2.1.2 Sensores Laser

Os sensores de distância a laser, quanto ao princípio de funcionamento, podem ser separados em dois tipos principais: tempo de voo e triangulação.

Sensores de distância a laser baseados em tempo de voo têm o mesmo princípio de funcionamento apresentado nos sensores de ultrassom. No entanto, no lugar da propagação de uma onda sonora é realizada a propagação de uma onda eletromagnética de comprimento de onda na faixa do infravermelho. Para realizar a medição do tempo de voo nos sensores laser os métodos normalmente utilizados são: sinal pulsante e diferença de fase (Siegwart e Nourbakhsh, 2004).

A medição do tempo de voo a partir de um sinal pulsante é o mesmo método aplicado a sensores de ultrassom. Contudo, a aplicação desse método para sensores baseados em laser exige um processamento muito maior, pois a velocidade de propagação da luz é centenas de milhares de vezes maior que a do som. Com isso, o custo final desse tipo de sensor torna-se elevado.

O método de medição por diferença de fase (Figura 21) é o mais utilizado, pois é de mais simples aplicação e melhor custo/benefício. Nesse método, a distância do objeto é medida conforme a equação (32).

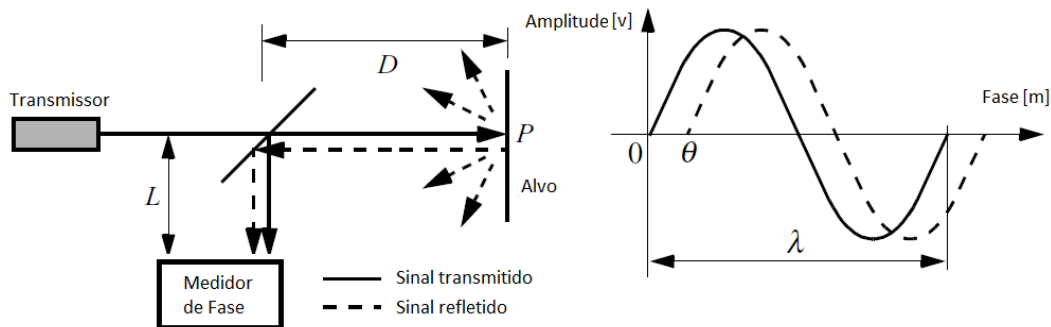


Figura 21 - Representação de sensor de distância por mudança de fase (Siegwart e Nourbakhsh, 2004)

$$D_{total} = L + 2D = L + \frac{\theta}{2\pi} \lambda, \quad (32)$$

onde  $D$  é a distância do ponto  $P$  de incidência do laser ao espelho que direciona o sinal refletido para o medidor de fase,  $L$  é a distância do espelho ao medidor de fase,  $\theta$  é a fase medida e  $\lambda$  é o comprimento de onda do laser.

O tipo de sensor laser baseado em tempo de voo mais utilizado é o LIDAR (*Light Detection and Ranging*), composto por um laser apontado para um espelho preso a um motor. Conforme o motor movimentar o espelho o sinal do laser é emitido em diferentes ângulos, formando assim o “campo de visão” do sensor, como ilustrado na Figura 22. Sensores do tipo LIDAR possuem valores típicos de “campo de visão”, resolução e alcance altos, respectivamente, maiores que  $180^\circ$ , menores que  $1^\circ$  e dezenas de metros (Romero et al., 2014). Essas características de varredura, resolução e alcance são muito superiores às apresentadas pelo sensor de ultrassom, cujo campo de visão varia entre  $20^\circ$  e  $40^\circ$ , a resolução é unitária, ou seja, do tamanho do campo de visão, e o alcance é de poucos metros. Outra vantagem do LIDAR em relação ao sensor de ultrassom é o tempo de resposta, pois é possível realizar medições de uma área bem maior em um tempo muito menor. A principal desvantagem dos sensores LIDAR, em relação aos sensores de ultrassom, é o preço, pois normalmente é centenas de vezes mais maior do que o um sensor de ultrassom.

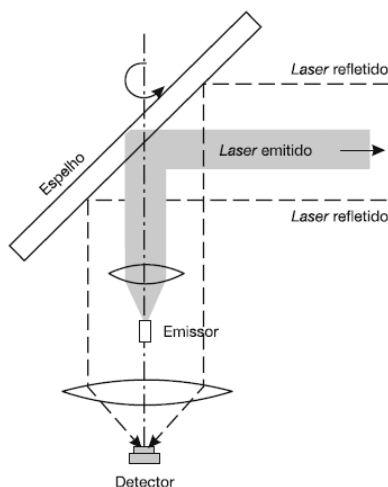


Figura 22 - Ilustração de mecanismo de rotação de sensor LIDAR (Romero et al., 2014)

Os sensores laser por triangulação são muito mais simples, quando comparados ao LIDAR. Esse tipo de sensor possui uma lente que direciona o sinal refletido para um PSD (*Position-Sensitive Device*) (Fraden, 2010) ou câmera linear, conforme ilustrado na Figura 23.



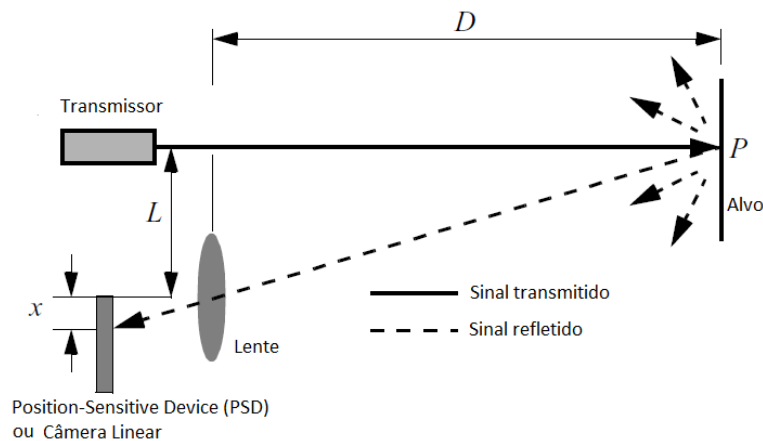


Figura 23 - Esquema de funcionamento de sensor de distância laser por triangulação (Siegwart e Nourbakhsh, 2004)

Na Figura 23,  $f$  é a distância focal da lente do receptor,  $L$  é a distância do transmissor ao receptor e  $x$  é a posição de incidência do sinal do receptor, podendo-se obter a distância  $D$ , conforme

$$D = f \frac{L}{x}. \quad (33)$$

Os sensores por triangulação, como pode ser observado na equação (33), têm o valor de distância proporcional a  $\frac{1}{x}$  e, conseqüentemente, uma melhor resolução para distâncias curtas. Esse tipo de sensor, diferente do LIDAR, tem o “campo de visão” pontual e apresenta um custo bem menor, semelhante ao de sensores de ultrassom. A principal vantagem dos sensores laser por triangulação, em relação ao ultrassom, é a maior resolução da distância. Uma das desvantagens é o alcance do sensor, pois ele é bem menor que o dos sensores de ultrassom e, principalmente, dos sensores tipo LIDAR.

Independente do princípio de funcionamento do sensor de distância a laser, tempo de voo ou triangulação, as principais vantagens em relação aos sensores de ultrassom são:

- *Maior frequência de leitura:* devido à velocidade de propagação da luz ser muito superior à do som, sensores laser possuem uma taxa de atualização bem maior que sensores de ultrassom.
- *Maior resolução:* sensores laser conseguem estimar melhor a distância ao obstáculo.

- *Cross-talk*: Sensores laser são menos suscetíveis a esse efeito.

Dentre as desvantagens de sensores que independem do princípio de funcionamento destacam-se:

- Não detecção de superfícies transparentes, como vidros e acrílicos.
- Superfícies altamente lisas, como espelhos, tendem a refletir o sinal luminoso para fora do receptor. Portanto, não são detectadas ou geram grandes erros de leitura.
- Tendem a apresentar erros de leituras em superfícies escuras, principalmente distantes.
- Podem sofrer interferência de outras fontes de luz de comprimento de onda na faixa do infravermelho, como lâmpadas incandescentes e o sol.

#### **2.2.2.1.3** **Sensor de visão**

Sensor de visão é o tipo de sensor que permite extrair a maior quantidade de informação dos objetos presentes no ambiente. Por meio desse tipo de sensor, além das informações de distância, é possível extrair informações como: tamanho, forma e cor. Eles são largamente utilizados com técnicas de visão e inteligência computacional para execução de tarefas complexas, como a identificação de padrões estáticos e/ou dinâmicos.

O elemento básico de um sensor de visão é o dispositivo sensor que converte a informação de luz em carga elétrica. Atualmente existem dois tipos principais de tecnologia que são empregados na construção desse dispositivo: o CCD (*Charge Coupled Device*) e o CMOS (*Complementary Metal Oxide Semiconductor*) (Karim Nice, n.d.). Ambas as tecnologias têm em sua composição básica uma matriz de pixels, elemento básico da imagem, que é construído a partir de foto sensores que liberam elétrons proporcionalmente à incidência de fótons. A quantidade de pixels presentes no sensor de visão é sua resolução, que normalmente está na ordem de milhões.

Passado o período de exposição do sensor à luz refletida pelo ambiente, a carga acumulada por cada elemento de imagem é lida para que a imagem seja

formada. A maneira como a carga é lida está relacionada à tecnologia empregada, CCD ou CMOS (Siegwart e Nourbakhsh, 2004). No CCD, a carga acumulada por cada pixel é transportada através da matriz de pixels até uma de suas laterais. Conforme a carga atinge a lateral ela é convertida de um valor analógico para um sinal digital correspondente à quantidade de carga acumulada. Já no circuito CMOS, cada pixel está ligado a um conjunto de transistores que condicionam seu sinal para seja transportado diretamente aos conversores analógico-digital (ADC).

Por ser uma tecnologia mais explorada e madura, o CCD apresenta vantagens em termos de qualidade, resolução e ruído das imagens captadas, em relação ao CMOS. Contudo, os dispositivos CMOS têm como principal vantagem, em relação ao CCD, o baixo consumo de energia.

Os sensores de imagem mais comuns e populares são as câmeras digitais que, além do elemento básico descrito anteriormente, possuem outros componentes como lente, obturador e o circuito processador, por exemplo, que possuem funções importantes no processo de criação da imagem. A lente é utilizada para direcionar a luz para a matriz de pixels do CCD ou CMOS, o obturador é um componente mecânico que regula o tempo de exposição dos pixels à luz e o circuito processador é responsável por controlar o processo de captação, tratamento e transmissão da informação da imagem captada.

Apesar da grande quantidade de dados do ambiente que podem ser extraídos por meio de uma imagem, sensores de visão apresentam limitações quanto às informações de dimensão e distância dos objetos captados. Para que se possa realizar a medição direta do tamanho de um objeto na imagem é necessária a existência de algum elemento cujas dimensões sejam conhecidas. Esse elemento será usado como referência para que se encontre a relação  $tamanho\ real / pixel$  da imagem. Contudo, o conhecimento dessa relação não é garantia para medição exata de tamanho e distância. Quando não há um conhecimento prévio da geometria e tamanho dos objetos que compõem o ambiente não é possível compensar o efeito de perda de profundidade (“achatamento”) que ocorre ao fazer uma representação 2D de um objeto 3D. Existem diversas técnicas que podem ser empregadas para compensar esse efeito de “achatamento”, como: luz estruturada, visão estéreo e variações de foco, zoom, posição ou ângulo entre fotos consecutivas (Siegwart e Nourbakhsh, 2004).

A técnica de luz estruturada consiste em projetar um padrão conhecido, por exemplo, uma grade, sobre o ambiente no qual a imagem será captada. Ao ser projetado, os quadrados conhecidos do padrão em grade serão distorcidos de acordo com o ângulo e distância de projeção e a geometria do objeto no qual o padrão foi projetado. A partir das medidas das distorções do padrão, a geometria dos objetos pode ser recuperada.

A técnica da visão estéreo consiste em recuperar a informação de profundidade por meio da comparação de duas imagens capturadas simultaneamente de um mesmo objeto através de duas câmeras que foram posicionadas de maneira paralela em relação aos seus eixos ópticos. Essas imagens captadas são processadas e as informações de distância são obtidas a partir das distâncias dos sensores às lentes e entre as lentes.

Apesar da grande quantidade de informação que pode ser extraída, sensores de visão têm como desvantagem demandar muito processamento para que a informação captada possua a representatividade desejada. Com isso, para aplicações em plataformas cuja capacidade de processamento seja baixa, o uso de sensores de visão torna-se pouco frequente, pois irá diminuir significativamente a frequência de operação do dispositivo.

#### **2.2.2.2 Encoder Rotativo**

O *encoder* rotativo é um dispositivo eletromecânico que traduz o movimento rotativo em um código analógico ou digital. Esse tipo de sensor é utilizado para medir a posição, deslocamento, e/ou velocidade angular de um atuador rotativo, sendo a última obtida de maneira indireta. Existem dois tipos principais de *encoders* rotativos: o absoluto e o incremental (Eitel, 2014; Danaher Industrial Controls, 2003).

O *encoder* rotativo absoluto é aquele cuja posição angular do sensor é codificada. Portanto, é possível identificá-la por uma leitura direta do sensor. Ele fornece a informação de posição enquanto estiver alimentado e sua posição pode ser representada, por exemplo, por um código binário no padrão *Gray Code* (Gray, 1953), Figura 24.

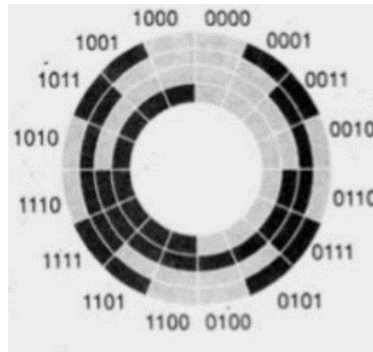


Figura 24 - Exemplo de disco de sensor rotativo absoluto de 16 posições codificada em Gray Code

*Encoders* rotativos incrementais se diferem dos absolutos por não representarem diretamente sua posição angular, mas sim a ocorrência de um deslocamento. A ocorrência do deslocamento é registrada toda vez que o sensor passa por um dos pontos de referência, que se encontram igualmente distribuídos em seu raio de atuação. O número de pontos de referência, mas conhecida como PPR (*Pulses Per Revolution*) é inversamente proporcional à resolução do sensor.

Para calcular a posição de um sensor incremental, após um dado deslocamento, é necessário acumular as ocorrências de deslocamentos e conhecer a posição inicial do sensor (equações (34) e (35)). Já sua velocidade é obtida pela variação desse deslocamento no tempo (equação (36)).

$$P_{final}(rad) = P_{inicial} + \left( \sum_{t=inicial}^{t=final} Deslocamento \right) * \left( \frac{2\pi}{Resolução} \right), \quad (34)$$

$$P_{final}(rad) = \frac{P_{final}}{2\pi} \text{ Se } (P_{final} > 2\pi) \quad (35)$$

e

$$\omega(rad/s) = \frac{\Delta P}{\Delta t} * \left( \frac{2\pi}{Resolução} \right), \quad (36)$$

onde  $P_{final}(rad)$  é a posição angular do dispositivo após realizar a rotação e  $\omega(rad/s)$  é a velocidade angular média exercida durante a rotação.

Para sinalizar a ocorrência do deslocamento, os sensores incrementais, normalmente, utilizam uma representação por ondas quadradas. Nessa representação o sensor possui duas saídas de dados, *A* e *B*, que geram ondas quadradas toda vez que o sensor passa por um de seus pontos de leitura. Essas ondas

são defasadas em  $90^\circ$  para que se identifique o sentido de rotação do sensor (Figura 25). Além das saídas A e B, alguns sensores incrementais possuem uma terceira saída, Z, que gera um pulso sempre que se passa por um ponto específico. Essa saída é normalmente utilizada para estabelecer uma posição de referência para o sensor.

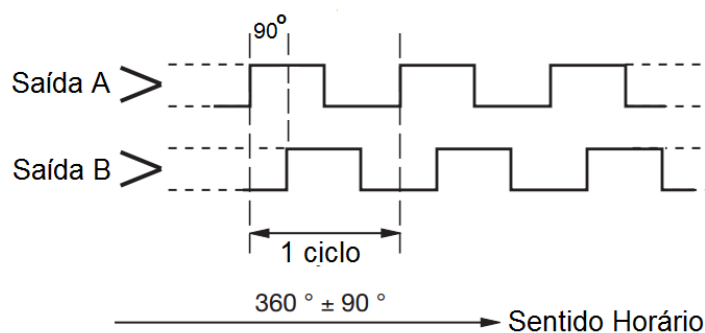


Figura 25 - Exemplo das saídas por onda quadrada de um sensor rotativo incremental

Os *encoders* rotativos têm três tipos principais de construção, mecânico, magnético e óptico (Eitel, 2014; Danaher Industrial Controls, 2003). Os sensores do tipo mecânicos são tipicamente formados por um disco com regiões de condução que, ao girar, fecha e abre um circuito elétrico, funcionando como uma chave. Ele pode ter uma saída independente para cada posição da chave, em uma construção de *encoder* absoluto, ou apenas duas saídas, para uma construção de *encoder* incremental (Figura 26). Esse tipo de sensor é normalmente utilizado em aplicações que requerem baixa velocidade e está suscetível a ruídos (*bounce*). A principal vantagem desse tipo de sensor é o baixo preço.

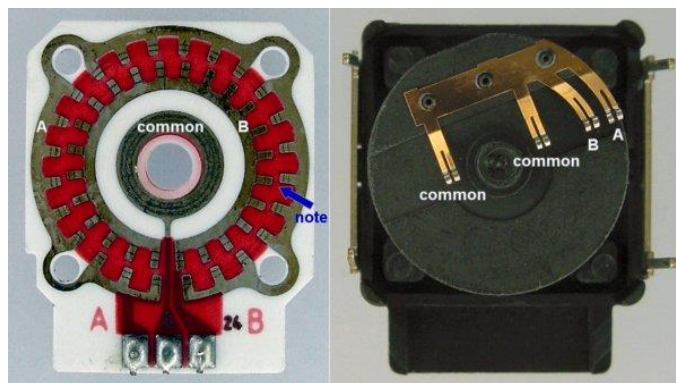


Figura 26 - Exemplo de *encoder* rotativo incremental mecânico

Os *encoders* do tipo magnéticos apresentam grande resistência a variações de temperatura, vibrações, impactos, poeira, graxa e outros contaminantes normalmente presentes em ambiente industriais. Ele tem como principal desvantagem ser sensível a ruídos magnéticos e o preço mais elevado. Existem três tipos principais de *encoders* magnéticos: o de efeito Hall, o de material magneto resistivo e o indutivo (Figura 27) (Eitel, 2014; Fraden, 2010).

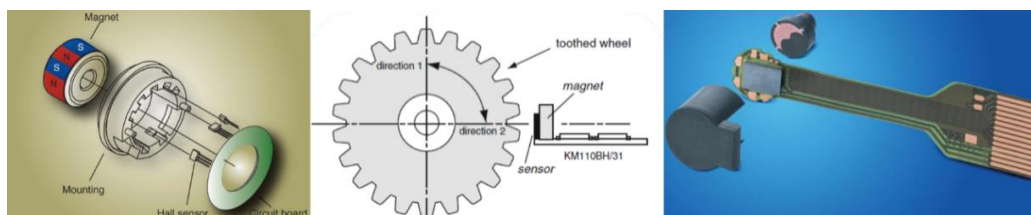


Figura 27 - Ilustrações de *encoders* magnéticos por efeito Hall, material magneto resistivo e material indutivo (Eitel, 2014; Fraden, 2010).

Os sensores do tipo óptico são os mais utilizados, pois costumam apresentar maior resolução, precisão e velocidade de operação que os demais tipos. Ao contrário dos sensores magnéticos, os sensores ópticos são mais sensíveis a variações de temperatura, vibrações, impactos, poeira e graxa, mas são pouco sensíveis a ruídos magnéticos. Eles são, tipicamente, compostos por um disco com perfurações ao redor de seu perímetro, um dispositivo emissor de luz e um sensor de luz. O disco perfurado é posicionado entre a fonte de luz e o sensor de luz (Figura 28). Ao girar o disco são realizadas interrupções na leitura dos sensores de luz e essas interrupções são utilizadas para identificar a rotação ou posição do sensor.

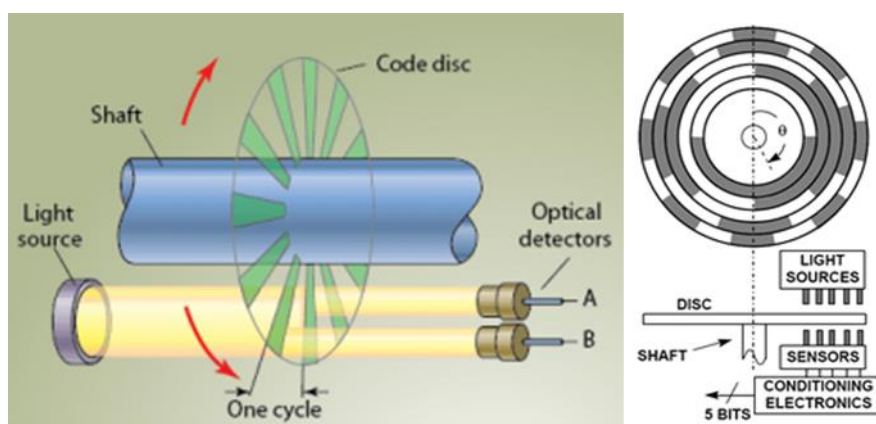


Figura 28 – Ilustração de *encoder* óptico incremental e absoluto (Eitel, 2014)

Em resumo, ao especificar um *encoder* rotativo, devem-se observar os seguintes pontos:

- Características do ambiente no qual o sensor irá trabalhar.
- Velocidade máxima na qual o sensor deverá realizar uma leitura.
- Resolução mínima requerida para a tarefa.
- Precisão e exatidão do sensor.

Essas características irão determinar o tipo de sensor rotativo ideal para a aplicação desejada.

### 2.2.2.3 Acelerômetro

O acelerômetro é um dos tipos de sensores mais comuns em aplicações de navegação inercial. Além da navegação inercial, esse tipo de sensor também é aplicado como um sensor de inclinação (tendo a gravidade como referência), vibração e impacto. Atualmente a maioria dos acelerômetros é construída com a tecnologia MEMS (*Micro-Electro-Mechanical Systems*) que é uma miniaturização de dispositivos e elementos mecânicos e eletromecânicos utilizando técnicas de micro fabricação (A MEMSnet, n.d.).

O princípio de funcionamento de um acelerômetro que utiliza a tecnologia MEMS se baseia na segunda lei de Newton ( $f = ma$ ). Para tal, a construção do acelerômetro é feita com um corpo de prova que tem liberdade se deslocar em um determinado eixo. Esse corpo de prova está preso ao encapsulamento do sensor por uma estrutura que simula o efeito de uma mola, e dentro deste encapsulamento há um fluido, normalmente o ar, que simula o efeito de um amortecedor (Figura 29). Ao exercer uma aceleração, o corpo de prova apresenta uma força resultante que é proporcional ao seu deslocamento (Fraden, 2010), conforme as equações (37) a (39).



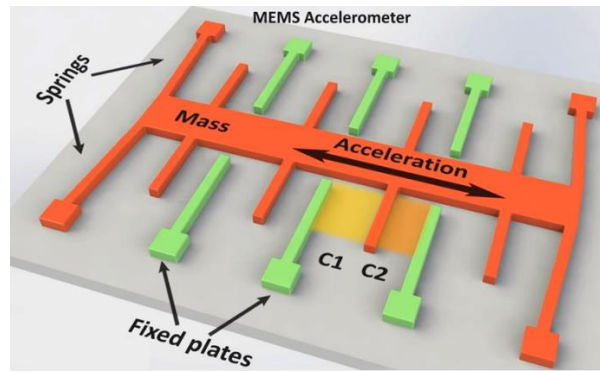


Figura 29 – Ilustração de acelerômetro capacitivo MEMS (Nedelkovski, 2015)

$$f(t) = m\ddot{x} + b\dot{x} + kx = -ma, \quad (37)$$

$$L\{f(t)\} = ms^2X(s) + bsX(s) + kX(s) = mA(s) \quad (38)$$

e

$$X(s) = \frac{mA(s)}{ms^2 + bs + k}, \quad (39)$$

onde  $\ddot{x}$ ,  $\dot{x}$ ,  $x$  e  $m$  são, respectivamente, a aceleração, velocidade, posição e massa do corpo de prova,  $k$  e  $b$  são, respectivamente, as constantes de elasticidade e amortecimento do suporte do corpo de prova.

A maneira como é medido o deslocamento do corpo de prova é o que difere os tipos de acelerômetros. Estes por sua vez podem ser capacitivos, piezoelétricos, piezo resistivo e térmicos (Fraden, 2010). A técnica empregada para medir o deslocamento irá influenciar nas características de sensibilidade a variações de temperatura, ruídos magnéticos e vibrações. Independente do tipo de acelerômetro, algumas características básicas devem ser consideradas para sua especificação, tais como:

- *Sensibilidade*: o quanto a saída do sensor varia, por unidade de aceleração, nas condições nominais de tensão e temperatura.
- *Temperatura de operação*: faixa de temperatura na qual o dispositivo opera com o desempenho de suas especificações.
- *Faixa de medição*: valores máximo e mínimo de aceleração que o dispositivo pode registrar.

- *Sensibilidade a variação e temperatura*: variação de *offset*, da incerteza e linearidade com a variação do valor de temperatura dentro da faixa de operação.
- Máximo impacto suportado pelo sensor sem perder as especificações nominais.
- Resolução.
- Exatidão.
- Número de eixos ortogonais que o sensor consegue medir.
- Frequência máxima de leitura.
- *Não-Linearidade*: maior distância que um valor medido apresenta ao aproximar todos os valores medidos dentro da faixa de operação do sensor por uma reta.

Um fato importante a ser observado é que um acelerômetro não consegue distinguir uma aceleração dinâmica da gravidade. Caso se deseje obter a aceleração real do dispositivo, deve-se, primeiramente, identificar a orientação do sensor para que se possa remover a componente da gravidade de cada eixo. A identificação da orientação do sensor pode ser feita, por exemplo, a partir de outro sensor, como o giroscópio. Outro ponto a ser observado é a necessidade de calibração do sensor (Kionix, 2015). A calibração mais simples de um acelerômetro visa corrigir o erro do valor de *offset*, também conhecido como *bias*. O procedimento de calibração consiste nos seguintes passos:

- Girar lentamente o sensor em direção perpendicular ao eixo de medição, de modo a obter valores do vetor gravidade alinhado a esse eixo e em ambos os sentidos;
- Realizar uma média aritmética com os valores máximo e mínimo obtidos;
- Subtrair o valor médio obtido de cada nova leitura do sensor; e
- Repetir esse procedimento para cada eixo.

#### 2.2.2.4 Giroscópio

O giroscópio, juntamente com o acelerômetro, é um dos sensores mais empregados na navegação inercial. Eles são comumente utilizados para se obter uma referência geométrica ou para medir a velocidade angular de um dispositivo. Existem três tipos básicos de giroscópio: de rotação, de vibração e óptico (Fraden, 2010).

Os giroscópios de rotação foram os primeiros giroscópios criados. Eles são dispositivos mecânicos compostos por um disco rígido livre para girar ao redor de um eixo. Esse eixo, por sua vez, está preso a uma estrutura que permite a rotação sobre um ou dois eixos ortogonais (Figura 30). O princípio de funcionamento desses giroscópios é baseado na lei de conservação do momento angular, que afirma que o momento angular de um sistema é constante em direção e magnitude se o torque externo resultante que atua sobre o sistema é nulo, e na lei de Newton de movimento para rotação, que afirma que a taxa de variação com o tempo do momento angular em relação a um dado eixo é igual ao torque aplicado sobre o eixo.

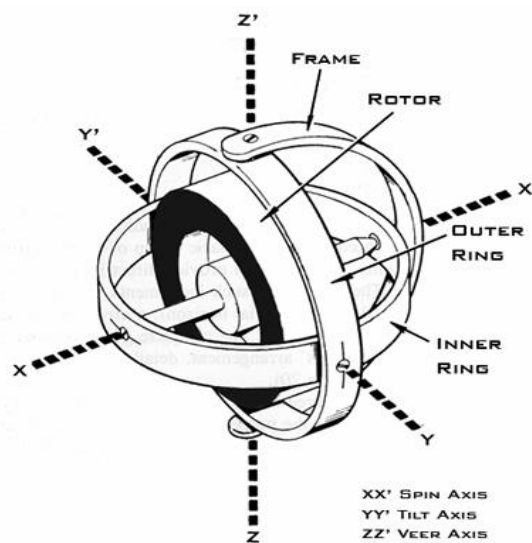


Figura 30 - Ilustração de giroscópio mecânico

Os giroscópios de vibração são atualmente os mais utilizados, devido ao menor tamanho e custo. Eles são construídos com a tecnologia MEMS, assim como os acelerômetros. Sua estrutura consiste em um corpo de prova que oscila em um dado eixo (Figura 31). Ao sofrer uma rotação externa ao eixo de oscilação é gerada uma aceleração perpendicular devido ao efeito de Coriolis. Essa aceleração,

proporcional à velocidade linear do corpo de prova,  $v_{osc}$ , e à velocidade angular do eixo,  $\Omega$ , equação (40), gera um deslocamento no corpo de prova perpendicular à sua oscilação, conforme Figura 32. Esse deslocamento é medido para então se obter o valor da velocidade angular do eixo.

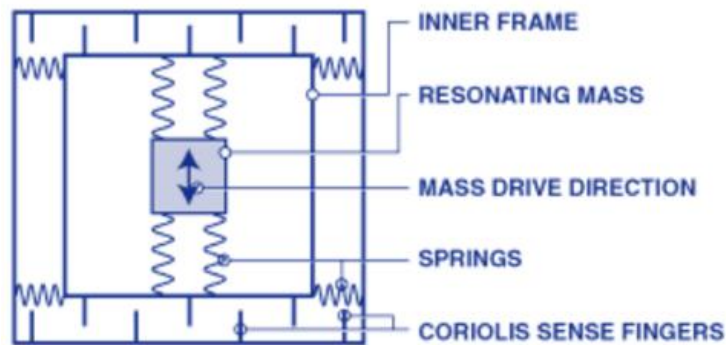


Figura 31 - Ilustração da estrutura de um giroscópio MEMS (SensorWiki, 2013a)

$$a_{coriolis} = 2v_{osc}X\Omega \quad (40)$$

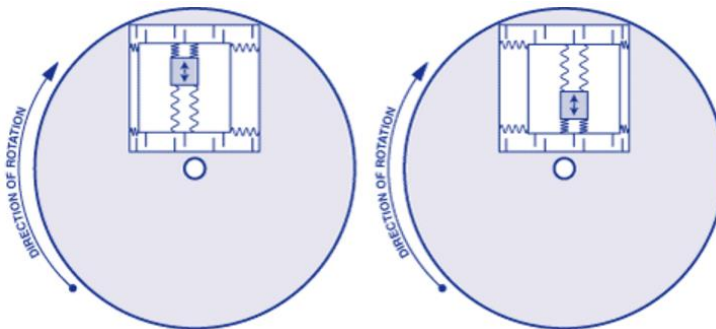


Figura 32 - Ilustração do efeito do deslocamento do corpo de prova com a rotação do sensor (SensorWiki, 2013a)

O giroscópio óptico, ao contrário dos modelos apresentados anteriormente, não possui partes móveis. Essa característica reduz efeitos indesejados como o *drift*, que é comum aos outros tipos de giroscópios. O princípio de funcionamento do giroscópio óptico é baseado no efeito *Sagnac* (Sagnac, 1913a, 1913b; Brown, 2015).

A construção dos primeiros giroscópios ópticos, assim como o experimento realizado por *Sagnac*, foi feita com um emissor de luz cuja onda, dividida por um espelho, percorria uma trajetória fechada em sentidos opostos (Figura 33). O efeito de *Sagnac* mostra que, ao atingirem o receptor, a soma das ondas irá gerar um

padrão correspondente a sua defasagem e que essa defasagem ocorrerá sempre que houver uma rotação na estrutura de seu percurso, pois a rotação fará com que a onda que se propaga no sentido oposto à rotação chegue antes ao receptor do que a onda que se propaga no sentido da rotação (Figura 34). Essa defasagem é utilizada para se medir a velocidade angular dos giroscópios ópticos. Atualmente, essa estrutura é replicada com fibras ópticas.

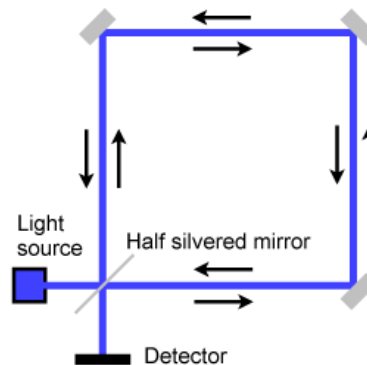


Figura 33 - Ilustração do percurso de teste do efeito Sagnac

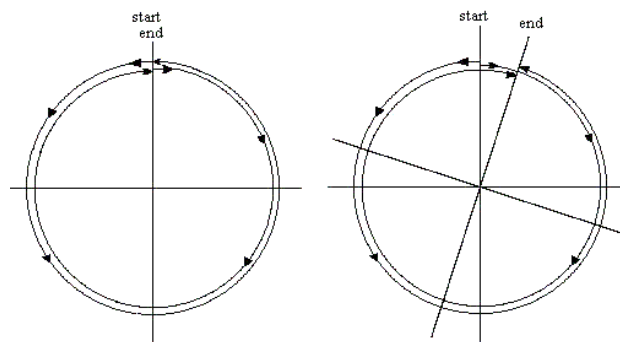


Figura 34 - Ilustração da influência da rotação do circuito de teste de Sagnac na convergência dos feixes

Independente do tipo de giroscópio utilizado, as características básicas, não comuns às apresentadas para o acelerômetro, que devem ser consideradas para sua especificação, são:

- *Angular Random Walk (ARW)*: é uma medida de ruído do giroscópio, que normalmente possui unidades em  $graus/\sqrt{s}$  ou  $graus/\sqrt{h}$ . Ela representa a variação devido ao ruído resultante da integração da saída de um giroscópio estacionário ao longo do tempo.

- *Bias*: erro de *offset* giroscópio quando não está sofrendo uma rotação. Ele pode variar com o tempo, temperatura e pode ser causado devido a erros de calibração, ao desligar e ligar o giroscópio ou devido a algum impacto sofrido.
- *Bias Drift*: o máximo desvio padrão obtido no valor do *bias* com o tempo, assumindo que todos os outros fatores externos permaneçam constantes. Ocorre principalmente durante os primeiros minutos em que o giroscópio é ligado.

Assim como o acelerômetro, o giroscópio necessita ser calibrado para corrigir o erro de *offset*. Contudo, no giroscópio, o valor de referência é obtido com a média de sucessivas medições com o sensor parado. Devido ao erro de *ARW* e *bias drift*, a calibração e referência de orientação do sensor devem ser periodicamente corrigidas.

#### 2.2.2.5 Magnetômetro

O magnetômetro é um instrumento utilizado para medir a direção e intensidade de campos magnéticos. Em uma tarefa de navegação ele realiza o papel de uma bússola, ou seja, a partir da medição do campo magnético da terra, obtém a orientação global do dispositivo. Atualmente, os sensores eletrônicos mais comuns para medir a direção de um campo magnético são os sensores de efeito *Hall*, magneto-resistivo e *fluxgate* (Siegwart e Nourbakhsh, 2004; Fraden, 2010; Konvalin, 2009; Phidgets, 2014).

O sensor de efeito Hall descreve o comportamento de um semicondutor mediante um campo magnético. Quando uma corrente constante é aplicada através de uma dada direção surgirá uma diferença de potencial, perpendicular à direção da corrente, de acordo com a orientação do semicondutor em relação ao fluxo magnético que o atravessa (Figura 35). Os magnetômetros de efeito Hall têm como vantagens o baixo custo e tamanho reduzido. Contudo, apresentam baixa resolução e sensibilidade.

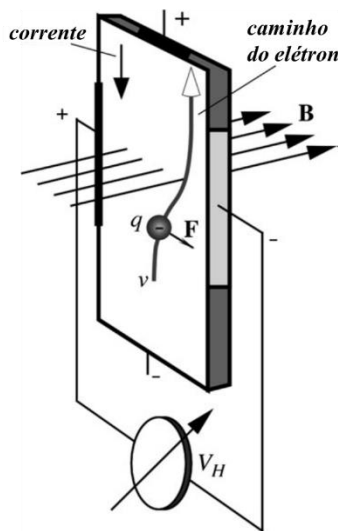


Figura 35 - Ilustração do efeito Hall (Fraden, 2010)

Os magnetômetros magneto-resistivos possuem um comportamento semelhante aos de efeito Hall. Entretanto, ao serem expostos a um campo magnético eles não geram uma diferença de potencial, mas sim uma força que se opõe ao fluxo de corrente. Portanto, seu comportamento é de um elemento resistivo cuja resistência varia com a intensidade do campo magnético. Esse tipo de sensor é largamente empregado em aplicações de robótica devido ao baixo preço e tamanho reduzido. Eles possuem as vantagens de terem maior sensibilidade, resolução e menor influência a ruídos que os magnetômetros de efeito Hall.

Os magnetômetros de *fluxgate* utilizam duas bobinas enroladas em dois núcleos de ferrite que estão posicionados paralelamente (Figura 36). As bobinas são enroladas de modo que a corrente elétrica circula em direções opostas entre elas. Ao circular uma corrente elétrica no circuito primário dois campos eletromagnéticos são induzidos, um para cada bobina. Esses campos terão mesma intensidade, porém fases opostas, fazendo com que a tensão induzida no circuito secundário seja nula. Ao ser exposto a um fluxo magnético externo, a fase entre os campos eletromagnéticos será alterada de acordo com a direção e intensidade que atinge cada bobina. Com isso, a tensão induzida no circuito secundário será proporcional ao campo externo. Esse tipo de magnetômetro apresenta melhor resolução e sensibilidade, em relação ao de efeito Hall. Contudo sua estrutura não permite construções tão pequenas e baratas.

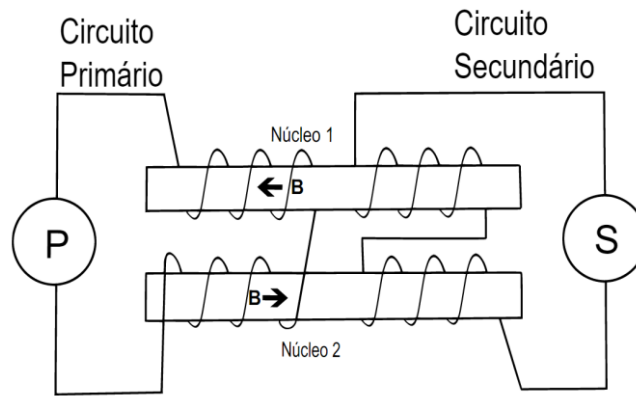


Figura 36 - Representação de um magnetômetro *fluxgate*

Independentemente do tipo de magnetômetro utilizado, algumas considerações devem ser feitas. Os sensores magnetômetros medem campos magnéticos presentes no ambiente e, portanto, são sujeitos a várias fontes de interferências eletromagnéticas. Para que um magnetômetro seja usado como uma bússola digital essas interferências, que geram distorções no campo magnético local da terra, devem ser compensados.

A Figura 37 representa o gráfico das saídas dos eixos  $Y$  e  $X$  de um magnetômetro em notação NED (North-East-Down), Figura 38, que sofreu uma rotação de 360 graus ao redor do eixo  $Z$ . O resultado, sendo um círculo centrado na origem, representa a saída de um magnetômetro calibrado ou que não está sofrendo nenhum tipo de interferência, seja por alinhamento com o campo ou por ruídos magnéticos (Kionix, 2015).



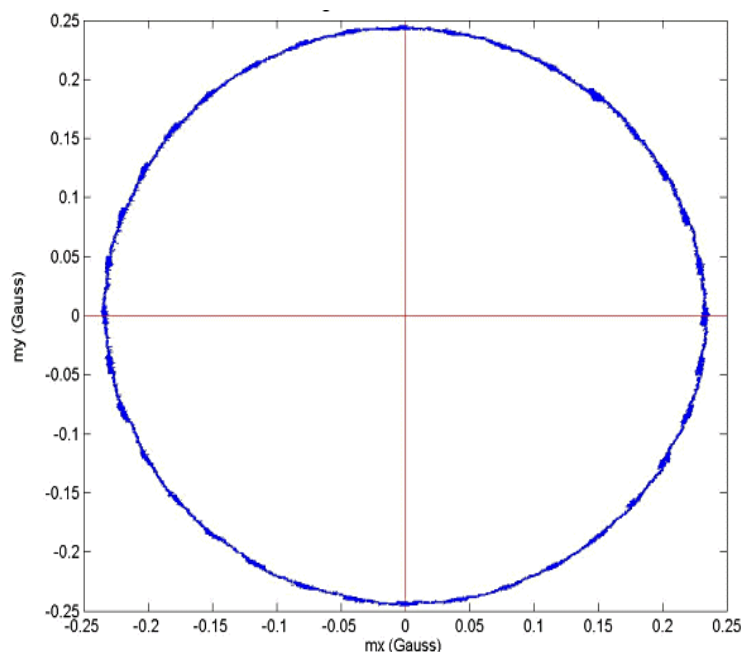


Figura 37 - Exemplo de dados do magnetômetro sem distorção (Konvalin, 2009)

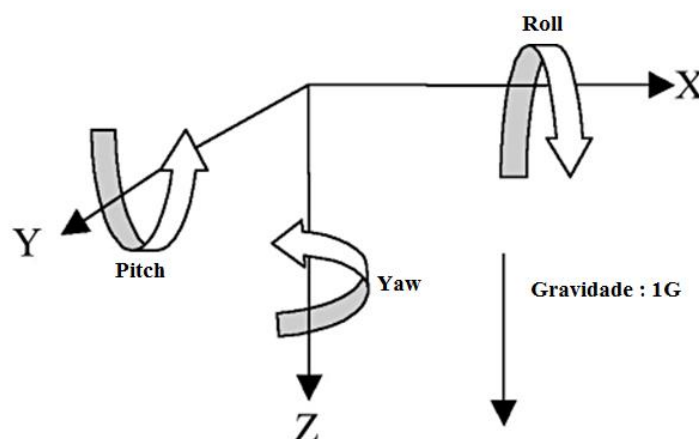


Figura 38 - Eixos em notação NED com indicação dos ângulos de Roll, Pitch e Yaw.

O primeiro passo para a calibração de um magnetômetro é compensar seu desalinhamento com o campo magnético da terra. Essa medida é necessária, pois a sensibilidade do magnetômetro varia com os ângulos de *pitch* e *roll* do dispositivo (Figura 39 e Figura 40), o que resultará em erros da orientação final. Esse efeito é conhecido como compensação de *tilt* (Kionix, 2015).

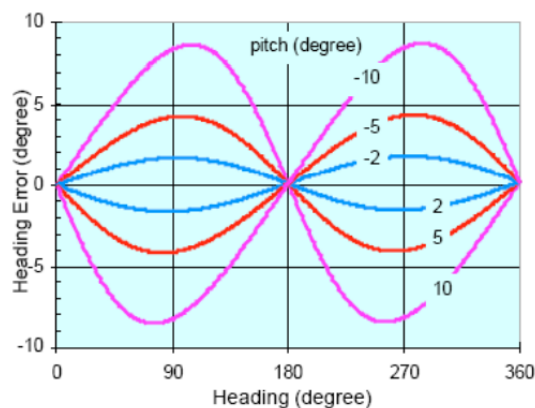


Figura 39 - Exemplo de erro de orientação devido a um ângulo de *pitch* de 40 graus, sem realizar a compensação de *tilt* (Kionix, 2015)

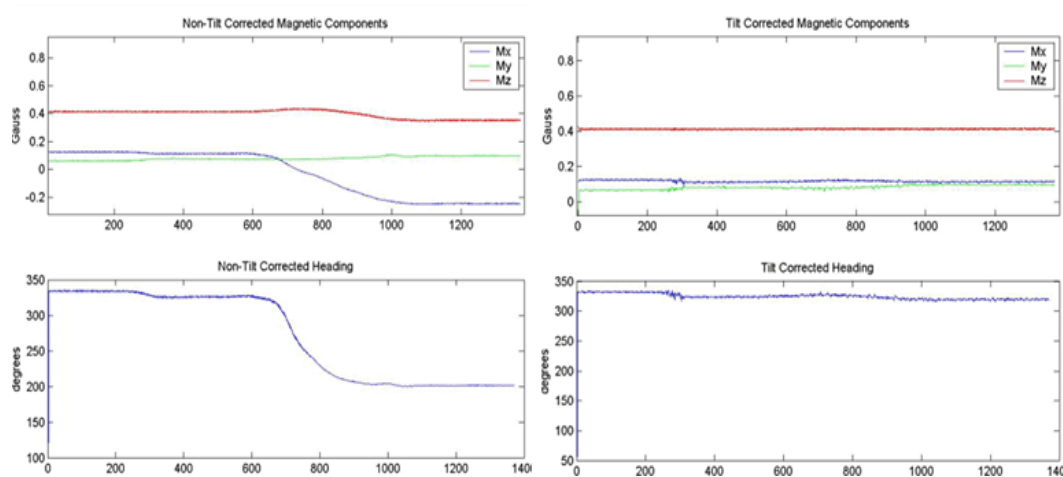


Figura 40 - Componente e orientação resultante do magnetômetro com e sem compensação de *tilt* (Konvalin, 2009)

Para compensar esse desalinhamento normalmente se faz uso de um sensor externo que indique a inclinação do magnetômetro. Um dos sensores mais utilizados nesse processo é o acelerômetro, no qual a gravidade é usada como referência para o alinhamento do sensor no plano. A partir dos dados fornecidos pelo acelerômetro, considerando que a única aceleração sofrida é a da gravidade, os ângulos de *pitch* e *roll* podem ser obtidos por (Phidgets, 2014)

$$roll = \arctan\left(\frac{g_y}{g_z}\right) \quad (41)$$

e

$$pitch = \arctan\left(\frac{-g_x}{g_y \sin(roll) + g_z \cos(roll)}\right), \quad (42)$$

onde  $g_x$ ,  $g_y$  e  $g_z$  são as saídas dos eixos  $x$ ,  $y$  e  $z$  do acelerômetro.

Com os valores de *pitch* e *roll*, a compensação de *tilt* é feita conforme as equações (43) e (44), e a orientação em relação ao Norte magnético,  $yaw_{NORTE}$ , pode ser calculada conforme a equação (45) (Kionix, 2015).

$$M_x = m_x * \cos(pitch) + m_y * \sin(pitch) * \sin(roll) + m_z * \sin(pitch) * \cos(roll), \quad (43)$$

$$M_y = m_z * \sin(roll) - m_y * \cos(roll) \quad (44)$$

e

$$yaw_{NORTE} = \arctan\left(\frac{M_y}{M_x}\right), \quad (45)$$

onde  $m_x$ ,  $m_y$  e  $m_z$  são as saídas não calibradas dos eixos  $x$ ,  $y$  e  $z$  do magnetômetro,  $M_x$  e  $M_y$  são as saídas dos eixos  $x$  e  $y$  do magnetômetro após a compensação de *tilt*.

Uma vez compensado o alinhamento do magnetômetro, a próxima etapa é a correção da distorção do campo magnético, devido a interferências externas. Esse efeito de distorção é classificado em dois tipos: *hard-iron* e o *soft-iron* (Konvalin, 2009).

O *hard-iron* é uma distorção constante que é adicionada ao campo magnético da terra. Essa distorção tem como efeito a adição de um valor constante às saídas do magnetômetro, gerando um erro de *offset*, conforme ilustrado na Figura 41.

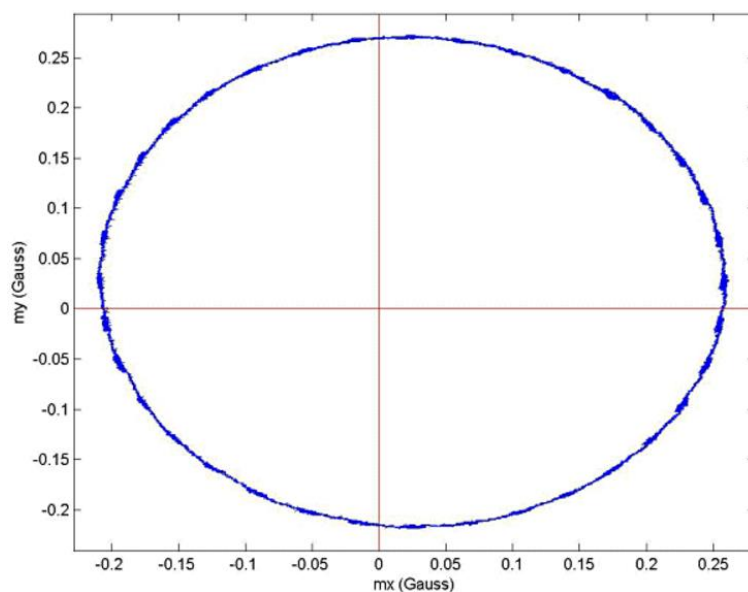


Figura 41 - Exemplo de dados do magnetômetro com distorção hard-iron (Konvalin, 2009)

A correção dessa distorção é feita determinando-se a distância à origem por meio da média entre os valores máximo e mínimo de cada eixo do magnetômetro. Esses valores máximos e mínimos são obtidos após uma amostragem realizando rotações ao redor de cada eixo.

O *soft-iron*, diferente do *hard-iron*, não é causado por uma fonte magnética constante, mas por algum material que gera distorção no campo magnético. Esse efeito pode ser visualizado na distorção do círculo plotado a partir das amostras dos eixos  $x$  e  $y$  de um magnetômetro após realizar uma rotação de 360 graus ao redor de seu eixo  $z$  (Figura 42). Uma das maneiras de corrigir a distorção *soft-iron* é identificar o maior eixo da elipse formada pelos dados e escaloná-la para aproximá-la a um círculo (Konvalin, 2009).

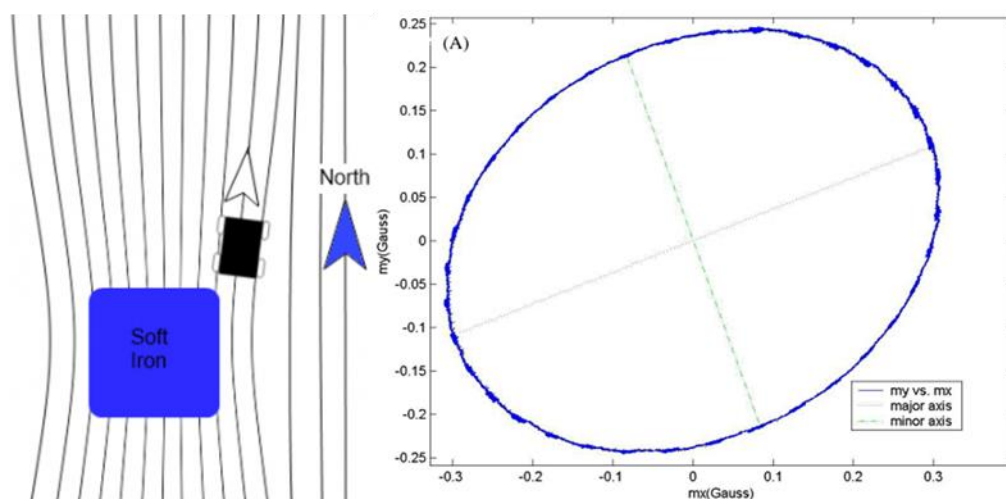


Figura 42 - Exemplo do efeito da distorção soft-iron (Phidgets, 2014; Konvalin, 2009)

Um fato importante a ser destacado é que toda a calibração deve ser feita a partir dos dados coletados na configuração na qual o magnetômetro irá operar. Portanto, ele deve estar no local que será usado e todos os equipamentos eletrônicos que irão funcionar durante a operação do magnetômetro devem estar ligados.

### 2.2.2.6 Central Inercial

A central inercial ou unidade de medição inercial (IMU) é a combinação do acelerômetro e giroscópio em um mesmo dispositivo sensor. Esse dispositivo é utilizado para estimar a posição relativa, velocidade e aceleração do dispositivo móvel. O diagrama básico de operação de uma central inercial pode ser observado na Figura 43, onde as informações do giroscópio e acelerômetros são combinadas para estimar as velocidades, posição e aceleração do dispositivo.

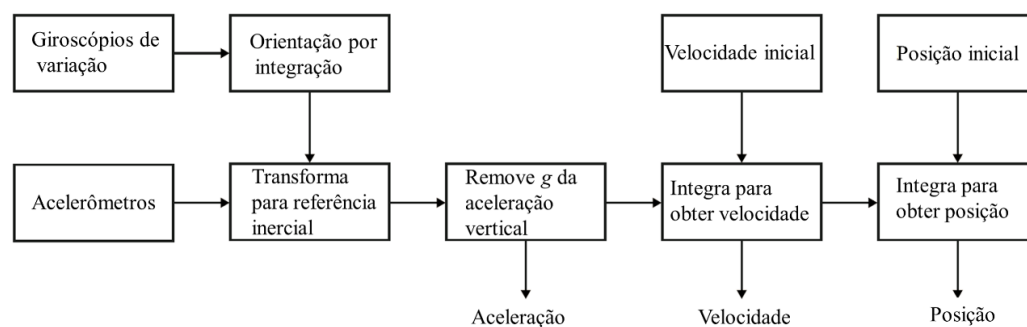


Figura 43 - Diagrama básico de operação de uma central inercial (Siciliano e Khatib, 2008)

Além desses dois dispositivos básicos, atualmente, muitas centrais inerciais também embarcam magnetômetros. Com a adição desse novo sensor torna-se possível estimar a orientação do dispositivo em relação a um referencial global, por exemplo, o norte magnético da terra, conforme mostrado na equação (43).

Centrais inerciais são extremamente sensíveis a erros de medição devido aos ruídos inerentes aos giroscópios e acelerômetros. Por exemplo, o *drift* do giroscópio leva a erros na estimativa na orientação relativa do dispositivo. O erro de orientação terá como consequência um erro no cancelamento da aceleração da gravidade. Ao realizar uma integração dupla na aceleração para estimar o deslocamento, os resíduos do vetor gravidade irão gerar erros quadráticos na estimativa da posição (Siciliano e Khatib, 2008). Uma prática bastante comum para minimizar esses erros é realizar a combinação das informações dos sensores por meio de algum filtro probabilístico para compensar os ruídos e, conseqüentemente, obter uma melhor estimativa. Essa técnica é conhecida como fusão sensorial e um dos filtros probabilísticos mais utilizados é o filtro de *Kalman* (Kalman, 1960), que normalmente é aplicado nas variações estendido ou *unscented* (Thrun et al., 2005). Em Madgwick (2010), Sabatini (2006, 2011) e Bancroft e Lachapelle (2011) podem-se observar aplicações da técnica de fusão sensorial com e sem filtro de *Kalman* para melhora da estimativa dos dados de uma central inercial. Além da fusão dos dados dos sensores embarcados na central inercial, também é comum utilizar dados de outros sensores para auxiliar na correção dos erros de estimativa. Em Zhang et al. (2005); Caron et al. (2004), Magnusson e Odenman (2012) e Rios e White (2002) é realizada a fusão dos dados da IMU com um GPS para obter uma melhor estimativa da posição e correção dos erros.

#### 2.2.2.7

##### **Sensor de força e torque**

Sensores de força e torque são muito utilizados em manipuladores robóticos. A partir desses sensores é possível identificar a força e/ou torque exercido pelo manipulador sobre o objeto a ser atuado. Essa informação, normalmente, é acrescida na malha de controle do robô para que ele gere uma força suficiente para manipular o objeto sem que haja algum tipo de deformação do mesmo.

Em robótica assistiva, esses sensores são de grande valia para identificar e/ou classificar a interação do usuário para com o dispositivo. Ao perceber a força exercida pelo usuário sobre o dispositivo, juntamente com informações de sensores, como a central inercial, é possível classificar/identificar uma intenção de deslocamento para uma dada direção, a de parada ou mesmo uma possível queda do usuário (Huang et al., 2008). Por meio desses sensores, a interação do usuário com o dispositivo assistivo torna-se mais natural, pois eles excluem a necessidade de botões ou alavancas.

Atualmente existem diversos tipos de sensores capazes de medir força e/ou torque. Eles são normalmente feitos por sensores que medem a deformação de um dado material devido à força e/ou torque sofrido, *strain gauges*, ou por sensores táteis, feitos com material piezoelétrico, piezo resistivo, capacitivo, MEMS ou óptico (Fraden, 2010). Esses sensores podem ser comercializados em configurações que meçam a força e/ou torque em um, dois ou três eixos. O número de graus de liberdade desses sensores está relacionado à quantidade de eixos e elementos de força e torque que ele é capaz de medir. Por exemplo, um sensor de força e torque de seis graus de liberdade mede forças e torques aplicados aos três eixos. Apesar de serem sensores que fornecem muita informação, sensores de seis graus de liberdade tendem a serem muito caros, principalmente se comparados aos sensores táteis de força.

### **2.2.3 Hardware processador**

A escolha do hardware processador é uma etapa bastante importante para o projeto, pois ele irá influenciar no tempo de resposta e os tipos de periféricos que o dispositivo poderá suportar. Para o desenvolvimento desse projeto, uma das restrições impostas é que o hardware escolhido permita a instalação de um sistema operacional Linux. Essa restrição é devida à proposta de executar todas as rotinas do robô por meio do framework de robótica Player, que necessita de uma plataforma Linux para sua execução. O objetivo de utilizar o Player está descrito na seção 2.3.

Além de executar o sistema operacional, idealmente, o hardware processador deve ser portátil, leve e de baixo consumo, pois o objetivo é o controle de uma plataforma móvel operada por uma pessoa idosa.

Com base nessas restrições gerais, optou-se por uma arquitetura mini PC de desenvolvimento de código aberto disponível como uma Single Board Computer (SBC). Essas arquiteturas vêm se tornando cada vez mais populares e possuem excelente relação custo-benefício. Dentre os modelos existentes no mercado destacam-se: *Raspberry Pi B Rev2* (RASPBERRY PI Foundation, n.d.), *BeagleBone Black* (Beagleboard, n.d.), *Odroid-U3* (Hardkernel, n.d.), *CubieTruck* (CubieBoard, n.d.) e *Banana Pi M1* (Banana Pi, 2014). Na Figura 44, pode-se observar o gráfico referente aos modelos mais populares, de acordo com a votação realizada com usuários do site LinuxGizmos (Lehrbaum e Brown, 2014).

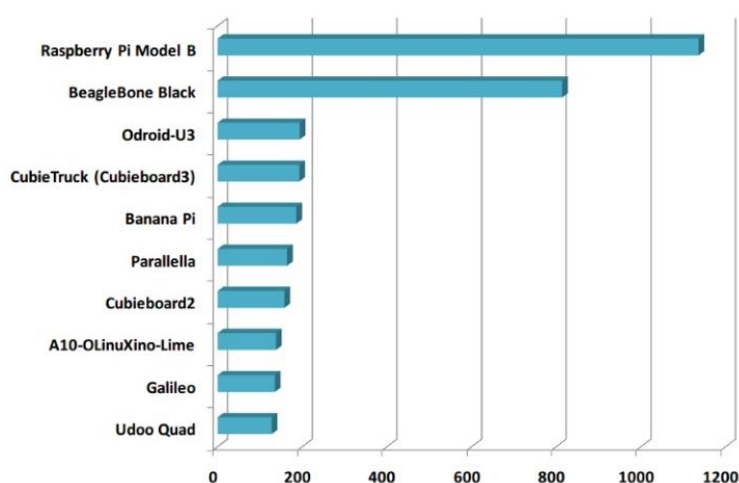


Figura 44 - Top 10 tipos SBCs de desenvolvimento aberto (Lehrbaum e Brown, 2014)

Uma comparação entre as características dos cinco melhores classificados, segundo a pesquisa de (Lehrbaum e Brown, 2014), pode ser observada na Tabela 1. Um ponto a ser observado é que muitos dos modelos listados possuem poucas ou nenhuma entrada analógica. Essa limitação pode ser facilmente contornada adicionando periféricos para a expansão das portas como o chip MCP3008 (Microchip Technology Inc., 2008). Esse chip possui 8 ADCs de 10 bits cada e se comunica via interface SPI (*Serial Peripheral Interface*). Já para expansão das I/Os digitais tem-se como alternativa o chip MCP23017 (Microchip Technology Inc., 2007). Ele acrescenta 16 bits de entradas/saídas por uma interface de comunicação I2C (*Inter-Integrated Circuit*).



Tabela 1 - Características dos 5 SBCs de desenvolvimento aberto mais populares

	<b>RASPBERRY PI B \$35</b>	<b>BeagleBone Black Rev C \$55</b>	<b>Odroid-U3 \$59</b>	<b>CubieTruck (Cubieboard3) \$89</b>	<b>Banana Pi M1 \$35</b>
<b>CPU</b>	1x ARM11 @ 700MHz + Broadcom VideoCore IV GPU	1x Cortex-A8 @ 1GHz	4x Cortex-A9 @ 1.7GHz + Mali-400 GPU	2x Cortex-A7 @ 1GHz + Mali-400 GPU	2x ARM Cortex-A7 1GHz + Mali400MP2 GPU
<b>MEMÓRIA</b>	512 MB RAM	512MB RAM + 4GB FLASH	2GB LP-DDR2 SDRAM	2GB DDR 3	1GB RAM DDR 3
<b>USB</b>	2	1	3	2	2
<b>VIDEO OUT</b>	HDMI + CVBS	HDMI	HDMI	HDMI + VGA + CVBS	HDMI
<b>WIFI</b>	X	X	X	Wifi+BT	X
<b>SATA</b>	X	X	X	1	SIM
<b>REDE</b>	10/100 wired Ethernet RJ45	10/100 wired Ethernet RJ45	10/100 wired Ethernet RJ45	10M/100M/1G Ethernet	10/100/1000 wired Ethernet RJ45
<b>DISPLAY</b>	SIM	SIM	X	X	SIM
<b>CAM</b>	SIM	SIM	X	X	SIM
<b>GPIOs</b>	34 Pins // 12 GPIOs	2x 46 Pins // 30 GPIOs	8 Pins // 8 GPIOs	54 Pins // 12 GPIOs	36 Pins // 9 GPIOs
<b>I2C</b>	1	2	1	1	1
<b>SPI</b>	2	2	X	1	2
<b>UART</b>	1	4	1	1	2
<b>PWM</b>	1	8	X	2	X
<b>ADC</b>	0	5	X	2	X
<b>TENSÃO</b>	5V	5V	5V	5V	5V
<b>CORRENTE</b>	700mA	460mA	2A	2.5A	---
<b>LINUX</b>	Debian GNU/Linux, Fedora, Arch Linux, Raspbian, RISC OS	Ubuntu, Debian GNU/Linux	Xubuntu	Lubuntu, Fedora, Cubieez	Bananian, Lubuntu, Raspbian, Debian GNU/Linux, Fedora, Arch Linux ARM, Gentoo, openSUSE
<b>ANDROID</b>	X	SIM	Android 4.x	SIM	Android 4.x
<b>TAMANHO</b>	85,60 x 56,5 mm	86,40 x 53,3 mm	83 x 48 mm	110 x 80 mm	92 x 60 mm

## 2.3 Software

Conforme apresentado no capítulo de introdução, uma das propostas desse projeto é realizar todo o gerenciamento de software e hardware por meio do framework de robótica Player. O objetivo da utilização dessa ferramenta é gerar um código de controle que independa do modelo do hardware escolhido, mas sim de suas funcionalidades. Além da abstração do hardware, o Player fornece uma interface de comunicação e acesso aos dispositivos embarcados no robô, o que facilita a leitura dos sensores por diferentes usuários. Essa característica permite, por exemplo, que um familiar ou médico do idoso monitore os dados do dispositivo em tempo real.

A seguir, será apresentada uma breve descrição do software Player, assim como do simulador de robôs Stage, que foi utilizado na validação inicial do código de controle, e das funcionalidades do aplicativo para comunicação do dispositivo com o usuário.

### 2.3.1 Player

O Player (Gerkey et al., 2003, 2010) é um framework de robótica que foi inicialmente desenvolvido por pesquisadores do Laboratório de Pesquisas Robóticas da *University of Southern California* no ano de 1999. O projeto surgiu da necessidade de uma interface para controle e simulação de sistemas de múltiplos robôs móveis que fosse eficiente, flexível e de grande compatibilidade. Ele foi criado como uma ferramenta de código aberto e de livre distribuição. Desde sua criação vem sendo modificado e expandido por diversos pesquisadores e colaboradores ao redor do mundo.

A ferramenta Player tem um funcionamento semelhante à de um sistema operacional. Ela proporciona um conjunto de interfaces padronizadas para o acesso de dados e envio de comandos que permitem uma manipulação em alto nível de diferentes tipos de hardware e/ou software. Por exemplo, por meio da interface *position2d* é possível realizar o controle de deslocamento e velocidade de diferentes modelos de robôs móveis, não importando as características do hardware ou mesmo se o robô está sendo executado em um simulador. Assim como em um sistema

operacional, para adequar os dados do hardware a ser manipulado a uma das interfaces do Player, é necessária a criação de um *Driver*. O *Driver* é um software que converte as instruções específicas do dispositivo em questão para os moldes do encapsulamento de uma dada interface.

A estrutura de operação do Player é baseada no modelo servidor/cliente. O servidor é o programa que gerencia as interfaces e *Drivers* dos sensores e atuadores. Ele cria portas de comunicação, via protocolo TCP/IP, para cada dispositivo que foi instanciado no momento da inicialização do servidor. Para instanciar um dispositivo, ele deve ser declarado no arquivo de configuração do Player (Figura 45). Nesse arquivo são declarados, para cada dispositivo, seja ele sensor, atuador, robô e/ou simulador, o *Driver*, as interfaces de entrada e/ou saída suportadas pelo *Driver* e as portas de acesso a cada interface declarada. Já o cliente é o programa, código de controle, que fará o acesso aos dados fornecidos pelo servidor e retornará os comandos de atuação. Como a interface externa do servidor Player é uma porta TCP/IP, programas clientes podem ser escritos em qualquer linguagem que forneça suporte a esse protocolo de comunicação e podem acessar o servidor por qualquer dispositivo que esteja na mesma rede. Atualmente, bibliotecas clientes, que encapsulam os detalhes do protocolo de comunicação do Player, estão disponíveis em diversas linguagens como: C, C++, Python, Java.

```

# Instantiate the MCP3008 driver
driver
(
    name "AdcDriver"
    plugin "../Adc/libAdc"
    N_ADC 8 # 8
    Com_ADC "/dev/spidev0.0" # "/dev/spidev0.0"
    Ref_ADC 4.82 # 5
    provides ["aio:0"]
)

# Instantiate the LV-MAXSONAR with MCP3008 driver
driver
(
    name "UltrassomDriver"
    plugin "../Ultrassom/libUltrassom"
    rangers [0 6.45] # [0 6.45]
    resolution 7
    ind_read 3
    average_read 3
    Com_ADC_Ranger "/dev/spidev0.1" # "/dev/spidev0.1"
    pin_control 7 # GPIO 7 // pin-7
    chip_select 15 # GPIO 15 // pin-8
    provides ["ranger:0"]
)

# Instantiate the Dual VNH5019 e ENA1J-B28-L00100L driver
driver
(
    name "OdometroDriver"
    plugin "../Odom/libOdometro"

```

Figura 45 - Exemplo de arquivo de configuração do Player

A estrutura servidor/cliente permite grande versatilidade no acesso aos dispositivos fornecidos pelo servidor. Uma das grandes vantagens dessa estrutura é que o cliente pode acessar informações de outros servidores assim com um servidor pode estar conectado a diferentes clientes. Essa característica permite que diferentes clientes tenham acesso simultaneamente a um mesmo dispositivo, bastando estar na mesma rede do servidor Player. Na Figura 46 é apresentado um diagrama resumindo as estruturas básicas do software Player. No destaque em verde está o servidor Player, fornecendo acesso e/ou recebendo comandos de um programa cliente, via porta TCP/IP, e acessando, via Drivers, o hardware real e/ou o dispositivo simulado.

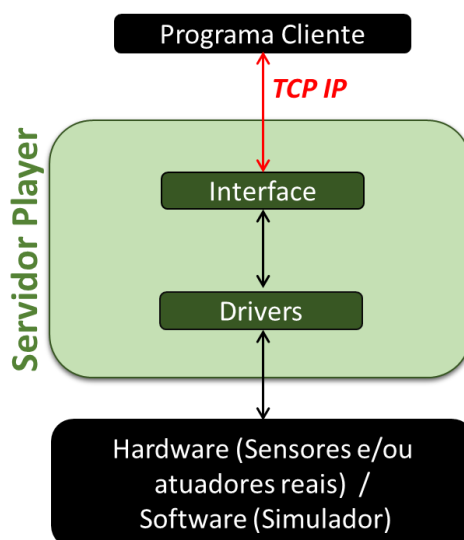


Figura 46 - Estrutura macro do software Player

### 2.3.2 Stage

O Stage (Vaughan, 2008; Vaughan e contribuidores, 2011; Gerkey et al., 2003) é um simulador 2D cinemático para robôs móveis que foi inicialmente desenvolvido como parte de projeto Player. Ele foi criado objetivando aplicações com sistemas multiagente e, portanto, é capaz de produzir um ambiente virtual com diversos robôs e sensores, além de objetos com os quais os robôs poderão interagir, a um baixo custo computacional, como ilustrado na Figura 47.

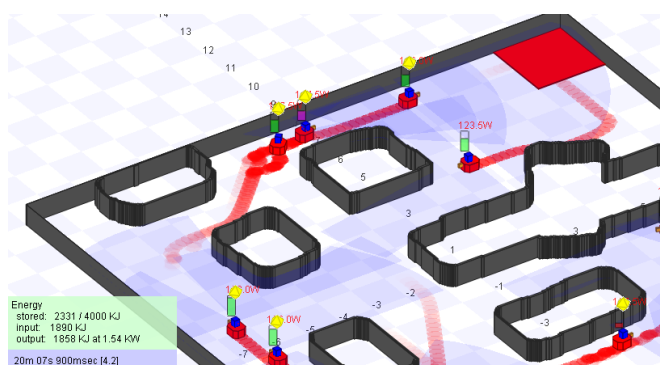


Figura 47 - Ilustração de simulação com o Stage (Vaughan e contribuidores, 2011)

O Stage possui uma estrutura modular, na qual, cada módulo (sensor, atuador ou objeto de manipulação) possui parâmetros editáveis para que suas características possam se assemelhar a diferentes dispositivos reais. Cada um desses módulos é formado a partir de um módulo básico de simulação, o “*model*”, no qual são

especificadas propriedades gerais, como posição, tamanho, cor, visibilidade a tipos de sensores, etc. Ao adicionar um módulo sensor/atuador a um dado módulo *model*, é criado um objeto virtual sensor/atuador (Figura 48).

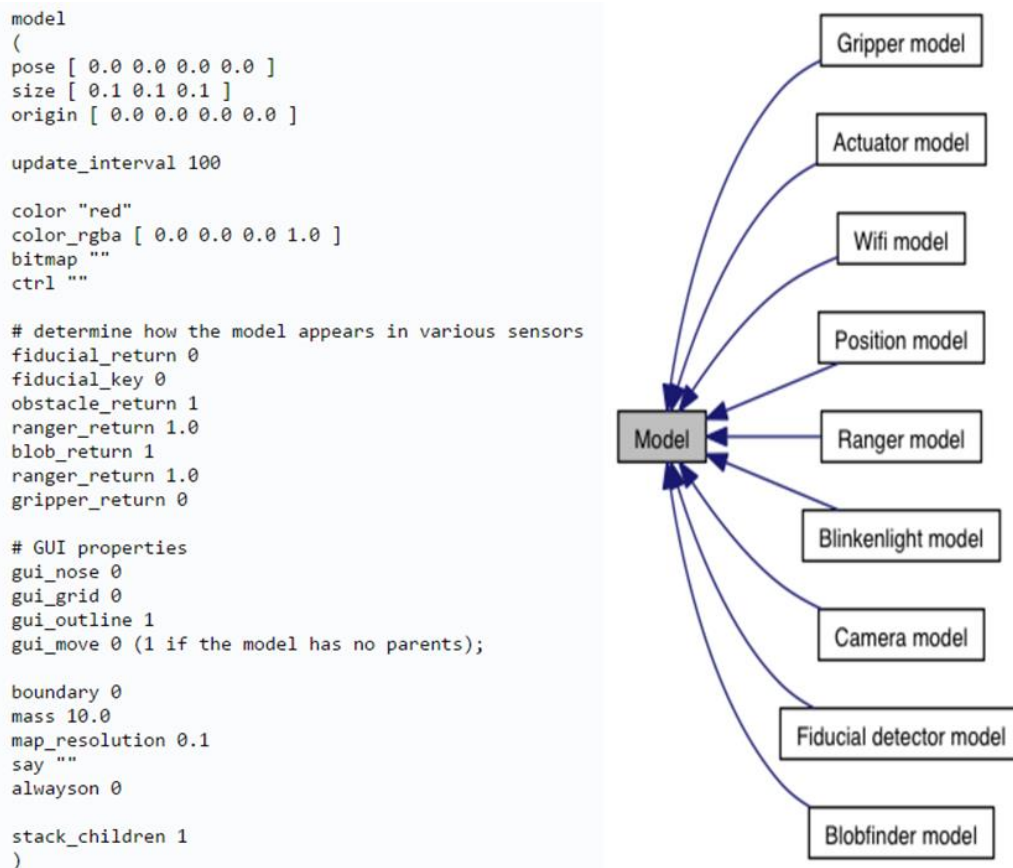


Figura 48 - Exemplo de arquivo de configuração STAGE e diagrama dos módulos do STAGE

O simulador Stage pode operar de três diferentes maneiras:

- *Como um simulador independente:* nessa configuração o próprio Stage executa o programa de controle do robô a partir de uma biblioteca fornecida pelo usuário.
- *Como um plugin para o Player:* nessa configuração o Stage fornece portas de acesso TCP/IP para os sensores e atuadores dos robôs que foram declarados no ambiente de simulação. Cada um dos módulos, sensor ou atuador, estará encapsulado segundo o padrão das interfaces do Player. Para que o Player acesse os dados do simulador Stage nessa configuração, deve-se adicionar o *Driver libstageplugin* na lista de

configuração do servidor Player, assim como os dispositivos que serão controlados no ambiente virtual.

- *Como uma biblioteca*: o Stage fornece uma biblioteca desenvolvida em C++, “*libstage*”, com o intuito de se desenvolver novos simuladores baseados no Stage.

### 2.3.3 API Google Maps

Para que o robô receba as rotas de navegação, foi desenvolvido pela equipe da empresa ThinkTank um aplicativo baseado na API do Google Maps para plataforma Android. Esse aplicativo será a principal interface de comunicação do usuário com o dispositivo. Por meio dele, o usuário irá receber informações do dispositivo como: nível das baterias, sinalização de obstáculos e os movimentos autônomos a serem executados; e enviará para o dispositivo comandos do usuário e a rota de navegação.

Para que o dispositivo receba a rota de navegação, o usuário deverá indicar no mapa do aplicativo o seu objetivo. Uma vez indicado o objetivo, o aplicativo, por meio da API do Google Maps, irá traçar a rota a pé de navegação. Essa rota está dividida em segmentos de retas e ângulos de navegação, que são enviados para o dispositivo com um período de 1,5 s. A cada interação do aplicativo ele atualiza a localização global do usuário, por meio do sensor GPS do celular, e envia uma atualização da distância a ser percorrida e do ângulo de navegação do trecho atual. Esse processo se repete até que o usuário atinja o objetivo desejado, sendo então a navegação autônoma desabilitada.

A conexão do dispositivo com o celular é feita via Wi-Fi e a comunicação via protocolo TCP/IP. Para tal, o aplicativo possui um programa cliente do servidor Player que foi desenvolvido por meio da biblioteca cliente *JavaClient3* (Javaclient, 2010). Por meio desse programa cliente, o aplicativo pode acessar qualquer informação do servidor, além de enviar comandos de atuação dos motores.

## 2.4 Controle

A teoria de controle é uma importante ferramenta para execução de tarefas e processos. Por meio da teoria de controle é possível regular a saída de um sistema para que alcance um valor desejado, garantindo estabilidade e robustez, além de regular a curva de convergência do mesmo. A teoria de controle vem evoluindo com o passar dos anos, devido à necessidade de controlar sistemas mais complexos e que exijam um comportamento cada vez mais preciso e robusto. Atualmente ela pode ser dividida em quatro grupos: controle clássico, controle moderno, controle robusto e controle inteligente.

O controle clássico foi onde surgiram as primeiras teorias de controle. Nele eram feitas análises de sistemas SISO (*single-input single-output*) e essas análises eram em sua maioria no domínio da frequência. Dentre as ferramentas desenvolvidas nesse período destacam-se: diagramas de Bode, critério de Nyquist, critério de Routh-Hurwitz, lugar das raízes e margens de ganho e fase. A síntese desses métodos de análise é aplicada nos controladores PID e de compensação Lead-Lag, que eram os utilizados nesse período (Ogata, 2010; Nise, 2012).

O controle moderno surgiu da necessidade de ferramentas para controle de sistemas MIMO (*multiple-input multiple-output*) e na análise no domínio do tempo de sistemas diferenciais. Com o controle moderno, surgiu a representação de sistemas por espaço de estado, assim como a análise por matrizes de controlabilidade e estabilidade (Ogata, 2010; Nise, 2012).

O controle robusto consiste no conjunto de ferramentas que vêm sendo criadas, desenvolvidas e/ou aplicadas para atender a uma das principais limitações encontradas no controle moderno: a sensibilidade gerada na estabilidade devido ao erro entre o sistema real e seu modelo (Ogata, 2010). A teoria de controle robusto realiza abordagens sobre os sistemas tanto no domínio do tempo quanto no da frequência. Matematicamente, a teoria de controle robusto é a mais complexa. Dentre as ferramentas utilizadas para análises dos sistemas destacam-se: decomposição em valores singulares (SVD), análise  $\mu$  e inequações de matrizes. A aplicação dessas análises é feita através de métodos para síntese de controladores robustos como:  $H_2/H_\infty$  e  $\mu$  (Dullerud e Paganini, 2005)



Os controles inteligentes surgiram com o desenvolvimento dos algoritmos de inteligência computacional. Eles possuem a vantagem de conseguir modelar o comportamento da planta sem a necessidade de aplicar as ferramentas matemáticas normalmente usadas nos outros métodos de controle. Nos métodos de controle inteligente, o controlador normalmente é obtido a partir da interação do algoritmo de inteligência computacional com uma base de dados extraída do sistema, da interação em tempo real com o sistema ou de um conhecimento prévio extraído de um especialista. Dentre as técnicas de inteligência computacional mais aplicadas em controle destacam-se: Redes Neurais, Aprendizado por Reforço e Lógica Fuzzy (Araújo, 2004). Essas técnicas ainda podem ser combinadas entre si ou com outras técnicas para a criação de controles híbridos.

A teoria de controle é muito extensa e envolve diversas áreas do conhecimento. Um mesmo problema pode ser abordado por diferentes técnicas de controle e a escolha da técnica ideal vai depender de diversos fatores, como complexidade do sistema, velocidade de resposta, precisão, robustez, custo computacional, entre outros. Nesse trabalho serão empregados controles PID e controles inteligentes baseados em Lógica Fuzzy, portanto, nas seções seguintes será apresentada uma breve introdução a essas técnicas.

#### 2.4.1 Controle PID

O controlador PID é o controlador mais utilizados na indústria, devido a sua fácil implementação e grande aplicabilidade. Ele é composto por três elementos básicos: um termo proporcional, um integral e um derivativo (equações (46) a (48)). Cada termo gera um diferente efeito no sistema e as combinações desses termos formam diferentes controladores. A equação (49) representa a equação de saída do controlador PID na configuração paralelo, dado o erro de entrada, e a equação (50) sua função de transferência.

$$P = k_p e(t) \quad (46)$$

$$I = k_I \int_0^t e(\tau) d\tau \quad (47)$$

$$D = k_D \frac{de(t)}{dt} \quad (48)$$

$$PID(t)_{out} = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \frac{de(t)}{dt} \quad (49)$$

$$\frac{PID(s)_{out}}{E(s)} = K_P + \frac{K_I}{s} + K_D s \quad (50)$$

Nas equações acima,  $k_P$ ,  $k_I$  e  $k_D$  são os pesos dos termos proporcional, integral e derivativo do controle PID e na Figura 49 é ilustrado o diagrama em blocos de um controlador PID paralelo aplicado a uma dada planta.

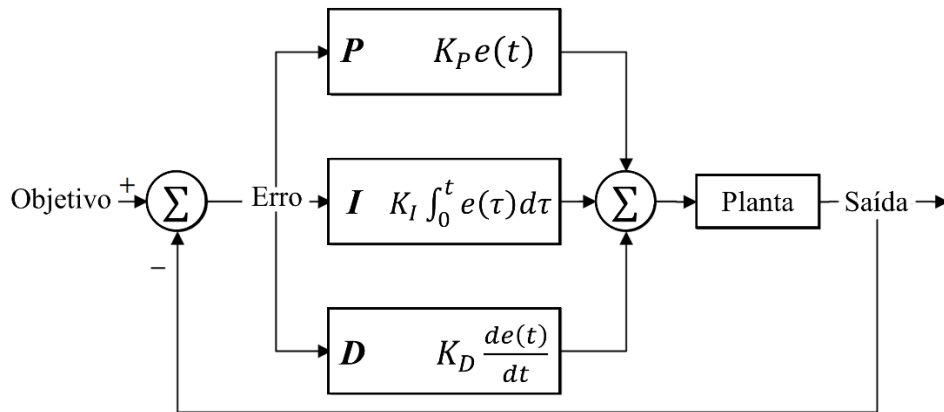


Figura 49 - Diagrama em blocos de um controlador PID paralelo aplicado a uma planta

Com o intuito de melhor ilustrar a nomenclatura das regiões da saída de um sistema que serão influenciadas por cada termo de um controlador PID, a Figura 50 destaca cada uma delas.

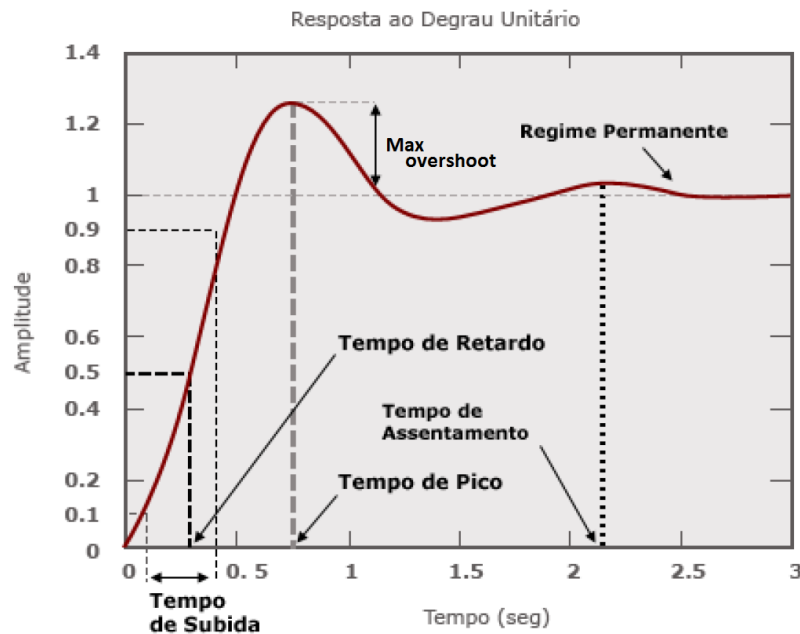


Figura 50 - Resposta de um sistema de segunda ordem com polos complexos conjugados a uma entrada degrau unitário (Pavani, 2011)

O termo proporcional tem efeito nos valores atuais de erro. Ele introduz um fator de escala no erro da planta e tem como efeitos positivos a redução do tempo de subida e do erro em regime permanente, mas não consegue eliminá-lo completamente. O efeito negativo é o aumento do *overshoot* e, para valores muito grandes de ganho, pode gerar instabilidade no sistema (Ogata, 2010; Nise, 2012), conforme Figura 51.

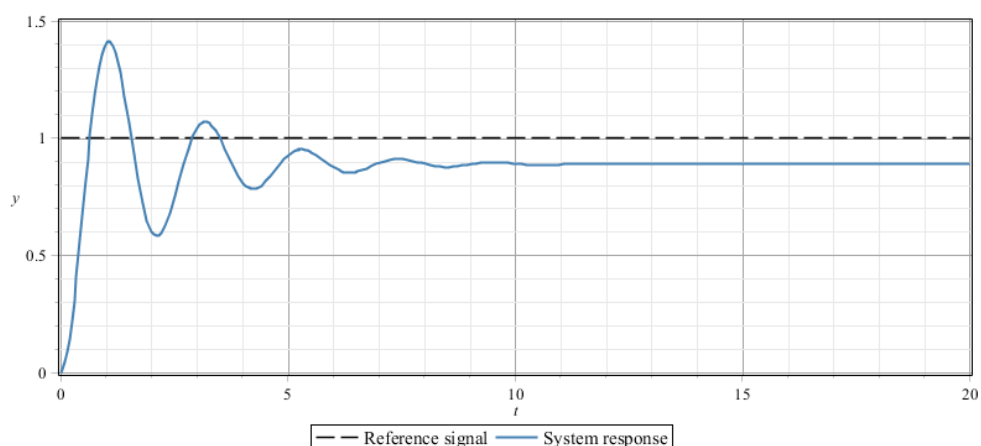


Figura 51 - Resposta à entrada degrau de um sistema de 2ª ordem com controlador P em série

O termo integral tem efeito sobre os valores passados de erro. A atuação do controlador integral é proporcional tanto à magnitude quanto à duração do erro. Sua principal vantagem é a eliminação do erro em regime permanente. Dentre as desvantagens está o aumento do tempo de assentamento e do *overshoot* (Figura 52). O controlador integral normalmente é aplicado em conjunto com o controlador proporcional, gerando o controlador PI. Esse controlador também é conhecido como controlador de atraso de fase (*Phase-lag*). Uma prática bem comum ao se desenvolver um controlador integral, seja ele PI ou PID, é estipular limitadores para que o ganho integral não atinja valores muito acima ou abaixo de um dado limiar desejado, evitando a saturação e *overshoots* muito altos. (Ogata, 2010; Nise, 2012).

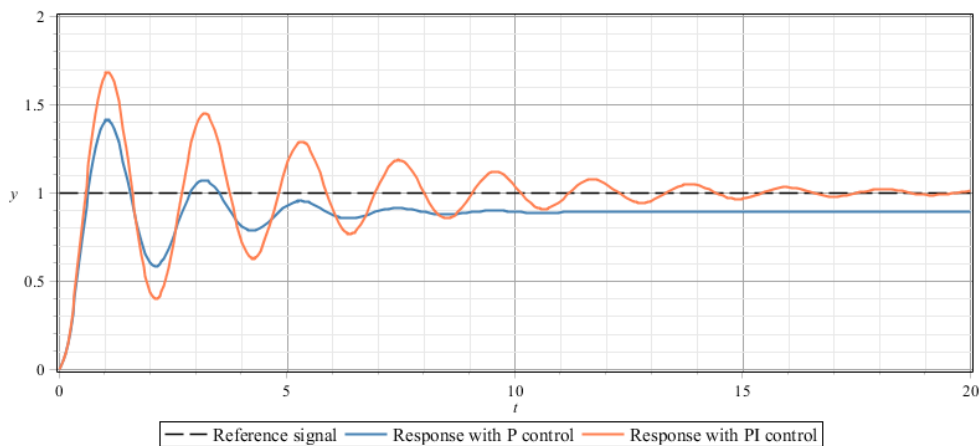


Figura 52 - Respostas à entrada degrau de sistemas de 2ª ordem com controlador P e PI em série

O termo derivativo tem um efeito sobre os valores de erros “futuros”, pois ao atuar sobre a derivada do erro ele atua sobre a tendência do mesmo, acelerando a resposta do sistema. Ele tem como característica aumentar a estabilidade do sistema, reduzindo o *overshoot* e melhorando a resposta transitória (Figura 53). A desvantagem desse termo é que ele introduz uma maior sensibilidade a ruídos. Assim como o controlador integral, ele normalmente é usado em combinação com o controlador proporcional. A combinação desses dois controladores gera o controlador PD, também conhecido como controlador de avanço de fase (*phase-lead*) (Ogata, 2010; Nise, 2012).

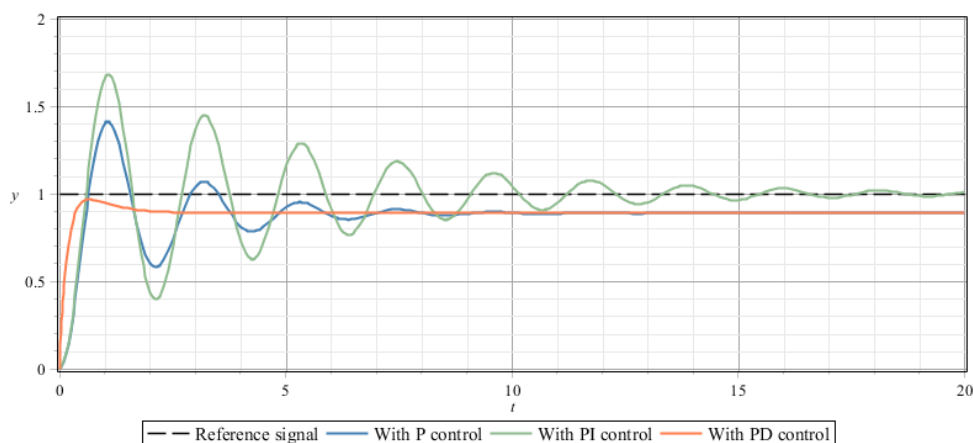


Figura 53 - Respostas a entrada degrau de sistemas de 2ª ordem com controlador P, PI e PD em série

A união dos três termos forma o controlador PID. O controlador PID combina as características dos três termos e a atuação de cada um será balanceada de acordo com o valor dado aos seus pesos:  $k_P$ ,  $k_I$  e  $k_D$ . Ao balancear os valores dos pesos é possível regular a resposta do sistema para que atinja o comportamento mais próximo possível do desejado, conforme ilustrado na Figura 54.

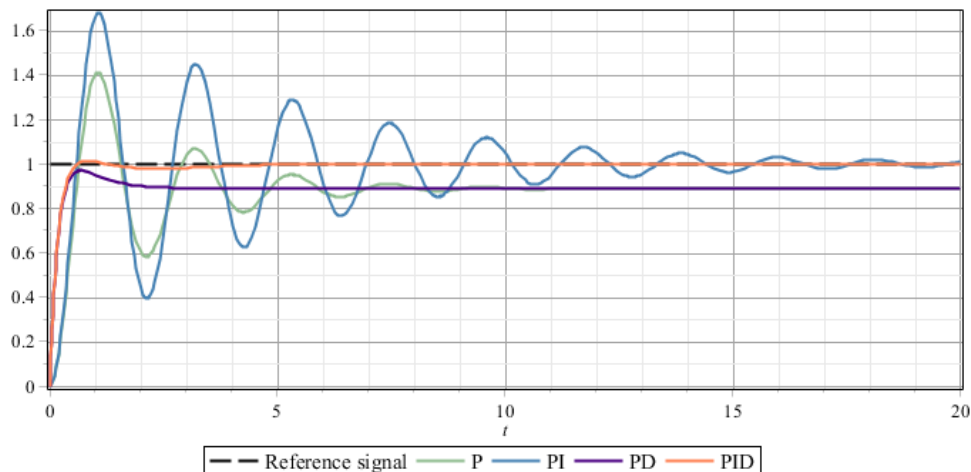


Figura 54 - Respostas à entrada degrau de sistemas de 2ª ordem com controlador P, PI, PD e PID em série

O processo de regular os pesos de um dado controlador, seja ele PI, PD ou PID, é conhecido com sintonia do controlador. Para realizar a sintonia de um controlador existem diversas técnicas que podem ser aplicadas, como as regras de Ziegler-Nichols, método do lugar das raízes, método de resposta de frequência, otimização computacional ou mesmo uma sintonia manual (Ogata, 2010). Algumas

das técnicas para sintonia do PID, como a técnica de Ziegler-Nichols, são aplicadas considerando que o controlador PID está escrito na forma padrão e não na forma paralela, conforme apresentado na equação (49), ou seja,

$$PID(t)_{out} = k_P \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right), \quad (51)$$

onde  $T_i$  é o intervalo de integração e  $T_D$  é o intervalo de derivação.

A relação do ganho da configuração padrão com os da configuração paralela é dada por

$$k_I = \frac{k_P}{T_i} \quad (52)$$

e

$$k_D = k_P T_D. \quad (53)$$

## 2.4.2 Lógica Fuzzy

A teoria dos conjuntos fuzzy foi criada por Zadeh (1965), com o objetivo de desenvolver ferramentas matemáticas para o tratamento de valores de caráter impreciso ou vago. A Lógica Fuzzy, baseada na teoria de conjuntos fuzzy, surgiu inspirada na lógica tradicional, contudo, diferentemente da característica binária da lógica clássica, na qual um valor pertence ou não a um conjunto, na Lógica Fuzzy, o valor possui um grau de compatibilidade que varia de 0 a 1. Essa característica permite que uma variável pertença a vários conjuntos com diferentes graus de compatibilidade. O grau de compatibilidade de uma variável a um dado conjunto fuzzy é conhecido como grau de pertinência.

Na Lógica Fuzzy, as variáveis cujos valores são nomes de conjuntos fuzzy são conhecidas como variáveis linguísticas. Por exemplo, a variável linguística velocidade, que, hipoteticamente, representa a rotação de um motor em um determinado processo, pode ser uma variável linguística cujos valores possíveis são: *baixo*, *médio* e *alto*. Esses valores representam conjuntos fuzzy que, por sua vez, são descritos por suas respectivas funções de pertinência, ilustrados na Figura 55.

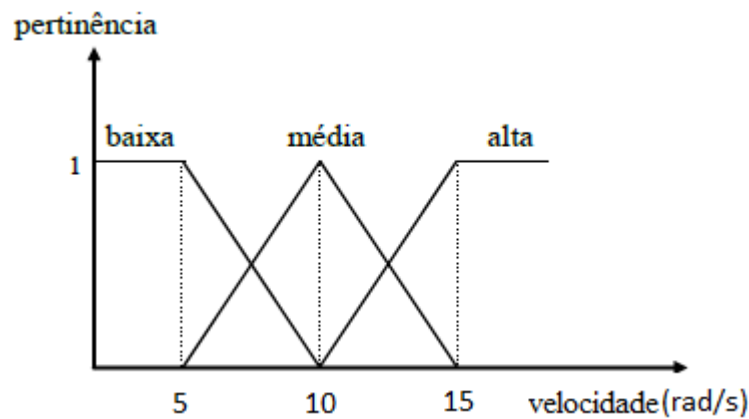


Figura 55 - Exemplo de conjuntos de valores de uma variável Fuzzy

A função de pertinência é aquela que descreve a distribuição dos valores pertencentes a um dado conjunto fuzzy. Ela pode ter diferentes formas, dependendo do conceito e das características do problema que se deseja retratar. Algumas das formas mais utilizadas para a área de controle são: triangular, trapezoidal e gaussiana.

Ao tratar um problema por meio de variáveis linguísticas, obtém-se uma forma sistemática para descrever fenômenos complexos ou mal definidos. Essa abordagem permite modelar de maneira simples problemas complexos de serem modelados por meio de técnicas matemáticas convencionais. A Lógica Fuzzy se mostra extremamente útil em aplicações que buscam emular a tomada de decisão de operadores humanos, que são capazes de tomar decisões complexas baseados em informações de caráter impreciso e vago, pois ela consegue tratar os modos imprecisos do raciocínio (Tanscheit, n.d.).

A Lógica Fuzzy possui aplicações em diversas áreas do conhecimento, como previsão de séries, classificação e controle. O conjunto de ações para o tratamento de um problema pela técnica de Lógica Fuzzy é conhecido como Sistema de Inferência Fuzzy (SIF) (Figura 56), cujas etapas são: fuzzyficação, regras, inferência das regras e defuzzyficação (Mendel, 1995; Tanscheit, s.d.).

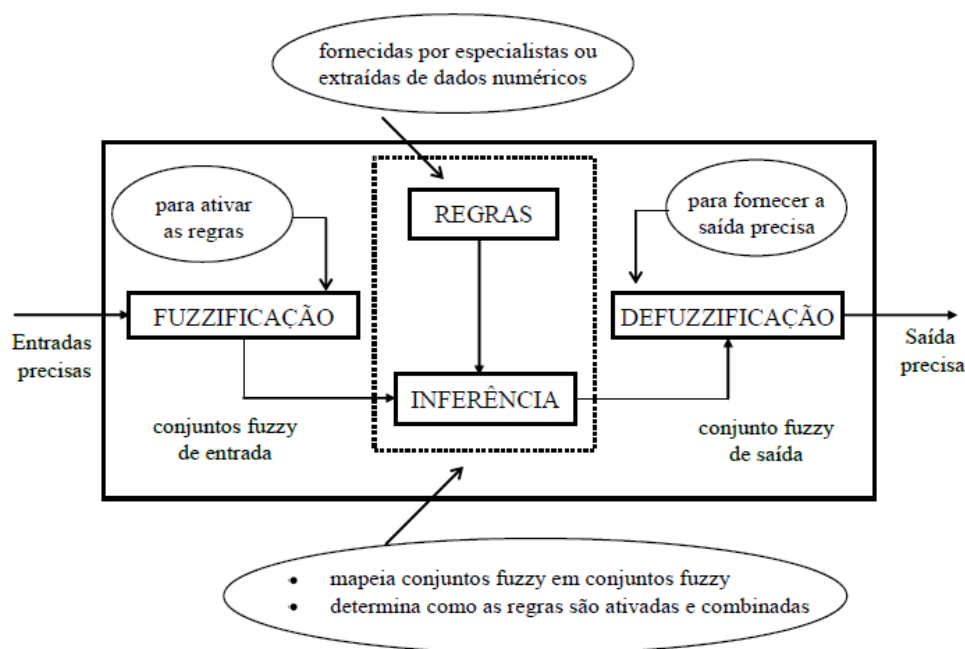


Figura 56 - Diagrama de um sistema de inferência fuzzy (Tanscheit, n.d.)

A fuzzificação consiste em transformar uma variável precisa em uma variável fuzzy. Isso ocorre verificando-se o grau de pertinência do valor preciso aos conjuntos fuzzy que representam as variáveis de entrada, conforme Figura 57.

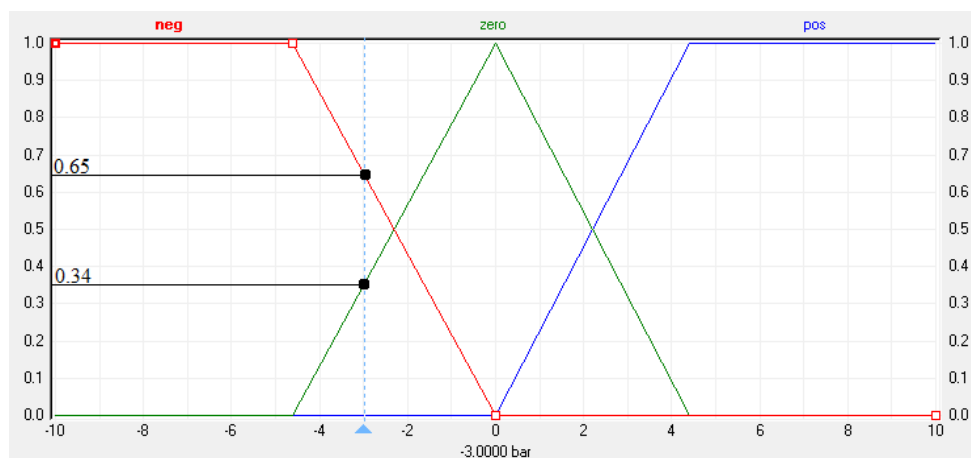


Figura 57 - Exemplo de fuzzyficação

No exemplo da Figura 57, o valor preciso de pressão -3 bar pertence ao conjunto fuzzy negativo, com pertinência de 0,65, e ao conjunto zero, com pertinência de 0,34. Esses conjuntos poderiam compor a variável linguística pressão, por exemplo.



As regras do sistema de inferência fuzzy são fornecidas por um especialista ou extraídas de um conjunto de dados e são do tipo “se ... então”. Assim como na lógica clássica, a Lógica Fuzzy possui operadores para união, interseção e negação dos conjuntos que compõem as regras. Os operadores de conjunção utilizados na Lógica Fuzzy são aqueles que se enquadram na norma triangular, sendo os mais utilizados o mínimo e o produto. Para representar a disjunção, utilizam-se operadores de co-norma triangular, sendo o operador máximo o mais utilizado. A negação é obtida pelo complemento da função de pertinência que representa o valor do conjunto fuzzy. Por exemplo, se  $\mu(x)$  é a função de pertinência de um dado conjunto, a negação desse conjunto será  $1 - \mu(x)$  (Tanscheit, n.d.).

A inferência das regras fuzzy, assim como na lógica clássica, é feita por um operador de implicação do antecedente com o consequente da regra. Contudo, em fuzzy, esse operador, em geral, é um operador de norma triangular, sendo os mais utilizados o mínimo e o produto. Com isso, o resultado de um processo de inferência fuzzy é uma saída fuzzy, cujos conjuntos que compõem seus valores serão ativados de acordo com a ativação dos antecedentes de cada regra.

O processo de defuzzificação consiste em transformar a saída fuzzy do conjunto resultante do processo de inferência em uma saída precisa. Para tal, diversas técnicas podem ser aplicadas como: centroide, primeiro dos máximos, média dos máximos, média ponderada dos máximos, entre outros (Passino e Yurkovich, 1997). Cada uma das técnicas poderá apresentar diferentes valores precisos para um mesmo conjunto fuzzy. Contudo, cabe ao especialista avaliar qual método representa melhor a natureza do problema que está sendo avaliado.

A seguir, no capítulo 3, será apresentado desenvolvimento do modelo virtual. Nesse capítulo, serão listadas as condições de contorno que o protótipo deverá atender e alguns dos sensores que serão utilizados. Também serão apresentados o agente virtual, o ambiente de testes e as características do controle de navegação Fuzzy aplicado.

### 3

## **Desenvolvimento do dispositivo robótico – Implementação em ambiente virtual**

Com a conclusão da fundamentação teórica e do estudo dos trabalhos relacionados, o desenvolvimento do dispositivo robótico proposto foi dividido em duas etapas: a primeira etapa consiste no desenvolvimento e aplicação do modelo em ambiente virtual e a segunda na aplicação e refinamento do modelo em uma plataforma real. Nesse capítulo, será abordada a etapa de desenvolvimento e aplicação do modelo em ambiente virtual. Essa etapa é de grande importância para acelerar o processo de desenvolvimento do modelo, pois permite o estudo em diversas configurações de ambiente de navegação, do dispositivo robótico e do algoritmo de controle, de maneira rápida, segura e sem custo financeiro. O resultado dessa etapa será aplicado como condição inicial do modelo em uma plataforma real, que será abordado no capítulo 4.

Para ter maior entendimento dos requisitos que o dispositivo deverá atender para se adequar ao usuário idoso, primeiramente, será apresentado nesse capítulo o resultado dos estudos das funcionalidades e condições de contorno. Também será feita uma introdução de alguns periféricos e recursos que serão utilizados na plataforma real para que se possa fazer a devida analogia com o que foi utilizado no modelo virtual.

### **3.1**

#### **Condições de contorno**

Antes de iniciar a apresentação das etapas de desenvolvimento do protótipo robótico, é fundamental destacar-se que o desenvolvimento foi baseado em levantamento e estudo dos requisitos considerando os futuros usuários. Assim, conforme apresentado no capítulo anterior, as funcionalidades e condições de contorno do protótipo foram objeto de estudo de Rivero (2015; Rivero et al, 2015a, 2015b), em sua adaptação e implementação da técnica de *Quality by Design* (QbD)

(Lionberger et al., 2008; Nadpara et al., 2012; DPT Laboratories, 2013) no desenvolvimento do protótipo do projeto “Bengala Robótica”.

Conforme os resultados obtidos por Rivero (2015), as principais funcionalidades que o dispositivo proposto deverá atender, e em ordem de prioridade, são:

- Prevenção de quedas;
- Assistência à locomoção; e
- Desvio de obstáculos e posicionamento.

Apesar da prevenção de queda ter sido indicada como a funcionalidade mais relevante, o protótipo desenvolvido nessa dissertação teve como foco a navegação e locomoção. Nesse primeiro momento o equilíbrio do usuário foi tratado de maneira passiva, ou seja, proporcionado apenas pelo apoio do usuário no dispositivo, sem que haja algum controle ativo que identifique tendência de queda e atue de modo a tentar impedi-la.

Em relação às condições de contorno, Rivero (2015) elaborou um Sistema de Inferência Fuzzy (SIF) para, a partir dos *Critical Quality Attributes* (CQA) e *Critical Process Parameters* (CPP), componentes da metodologia QbD, inferir a qualidade do protótipo em atender suas principais funcionalidades. Pelo fato da pesquisa de Rivero ter ocorrido em paralelo com o desenvolvimento dessa dissertação, para essa etapa do protótipo, as condições de contorno foram, de fato, baseadas em recomendações de diversos artigos científicos de projetos relacionados. Esse levantamento foi realizado por Rivero, como uma das etapas de seu trabalho, e pode ser observado no quadro 1 (Rivero, 2015).

Quadro 1 - Compilação de recomendações de artigos científicos para as condições de contorno de trabalhos relacionados realizada por Rivero (2015).

Aspecto relacionado	Recomendação relacionada.
Velocidade	Velocidade: 0,62 m/s (Andreoni et al., 2014) a 1.2 m/s (Rentschler et al., 2003). Cadência: 76,12 passos/min (em pacientes com síndrome post-pólio). Capacidade de se adaptar à velocidade máxima, mínima e média do andar cada usuário (Almeida, Zhang e Liu, 2007).
Força	O esforço feito pelo paciente deve ser menor que 10 N (Rentschler et al., 2003)

	Medir o esforço aplicado pelo usuário e transformar esse sinal em um valor de velocidade e direção (Martins et al., 2011)
Massa	Inferior a 41 kg (Rentschler et al., 2003)
Projeto	Não obstrutivo para o andar do paciente. Desligamento do modo assistivo para permitir que o usuário faça correções na navegação (Rentschler et al., 2003) Motores nas rodas ou seus eixos para melhor controle dos freios para compensar a gravidade nos chãos inclinados (Martins et al., 2011)
Assistência cognitiva	Capacidade de navegação e auto localização estruturado em ambientes conhecidos e não conhecidos (Rentschler et al., 2003)
Prevenção de quedas	O dispositivo deve suportar um ângulo de subida de 34,5 graus e um ângulo de descida de 21 graus (Rentschler et al., 2003) Sensores para detectar os obstáculos de degraus com suficiente antecedência (Lacey e Dawson-Howe, 1996)
Ubiquação	Controle compartilhado entre o dispositivo e usuário a fim de corrigir possíveis erros do dispositivo ao “navegar” por lugares ainda não mapeados ou conhecidos (Lacey e Dawson-Howe, 1996).
Segurança com respeito a deficiências Locomotivas.	Corrente de fuga inferior a 5 mA. Engrenagens e partes propulsivas não podem ser tocadas (Rentschler et al., 2003). Reconhecer escada e graus, mandando aviso com antecipação (Lacey e Dawson-Howe, 1996). Suporte parcial do peso do usuário, assim como suporte parcial dos braços do usuário, para portadores de doenças referidas a doenças e lesões na espinha dorsal (Martins et al., 2011) As rodas devem ter um elevado coeficiente de fricção para diminuir a probabilidade de deslizamento devido a um declive do chão (Martins et al., 2011)
Desvio de obstáculos	Deve ter a capacidade de anunciar e evitar obstáculos com ações evasivas (Rentschler et al., 2003)
Aviso de obstáculos	Sensores para anunciar obstáculos (Rentschler et al., 2003; Yu, 2008; Martins et al., 2011).
Segurança com respeito às deficiências visuais.	Caso os dispositivos sensores tenham mau funcionamento o usuário ou paciente deve ser avisado mediante sinal sonoro ou tátil (Rentschler et al., 2003)
Prevenção de quedas	Capacidade de detectar a queda da pessoa em poucos segundos dado que a queda é um processo de 1 ou 2 segundos e abrange vários movimentos (Noury et al., 2003; Yu, 2008)
Segurança quanto a deficiências de equilíbrio.	Valores máximos, mínimos e mais comuns das velocidades angulares de giro para evitar emissão de sinais falsos ou falsos positivos (Almeida, Zhang e Liu, 2007). Detectar a presença de duas mãos do usuário, de forma a comprovar se efetivamente o usuário sofreu uma queda ou se foi o dispositivo que caiu (Martins et al., 2011)

### 3.2

#### Controle de navegação

Conforme exposto no capítulo 1, um dos objetivos dessa dissertação é o desenvolvimento de um algoritmo de navegação baseado em Lógica Fuzzy. A Lógica Fuzzy foi escolhida principalmente devido a sua capacidade de modelar os modos imprecisos do raciocínio. Essa característica a torna uma ferramenta muito poderosa para a realização de tarefas de navegação, pois a navegação é uma tarefa simples e natural de ser executada por um ser humano, mas complexa de ser modelada por ferramentas matemáticas convencionais, principalmente em ambientes desconhecidos e não estruturados, como é o caso de ambientes urbanos. Além da capacidade de exprimir um conhecimento a partir de um conjunto de regras, a Lógica Fuzzy é uma ferramenta de fácil aplicação e baixo custo computacional, o que a torna ideal para a tarefa proposta.

Para que o robô se localize no ambiente e receba a informação de seu objetivo, conforme mencionado no Capítulo 2, ele estará conectado a um aplicativo para plataforma Android, que executará uma API do Google Maps. Essa API irá fornecer duas informações essenciais para a navegação autônoma do dispositivo: a distância euclidiana ao objetivo e a direção do objetivo. O objetivo fornecido pelo aplicativo é sempre um segmento de reta referente a um trecho da rota de navegação completa, para o objetivo estabelecido pelo usuário. Para que o robô consiga corrigir sua orientação com o objetivo ele estará embarcado com uma bússola. Já a correção da distância ao objetivo será feita pelo próprio aplicativo, que irá utilizar as informações do sensor GPS do celular para localizar o robô globalmente.

Para que o robô perceba os obstáculos presentes na sua rota de navegação, serão utilizados sensores do tipo ultrassom. Esse tipo de sensor de distância foi o escolhido, pois o mesmo não é suscetível a ruídos devido à intensidade luminosa do ambiente, principalmente gerada de ondas de comprimento infravermelho (o sol), ao contrário dos sensores de visão (câmeras) e infravermelho (PSD e LIDAR). Em relação aos sensores LIDAR, que são excelentes sensores para detecção de obstáculos, os sensores de ultrassom ainda têm como vantagem o custo, que chega a ser centenas de vezes menor. Em relação aos sensores de visão, o ultrassom também tem como vantagem o baixo custo computacional exigido para obter informações de distância.

As informações provenientes dos sensores de distância, da bússola, de orientação e de distância ao objetivo, serão entradas do sistema de inferência fuzzy que terá como saída as velocidades angulares de cada motor. Na Figura 58 é apresentado o diagrama macro das entradas e saídas utilizadas no SIF do modelo virtual, se será detalhado nas seções seguintes.

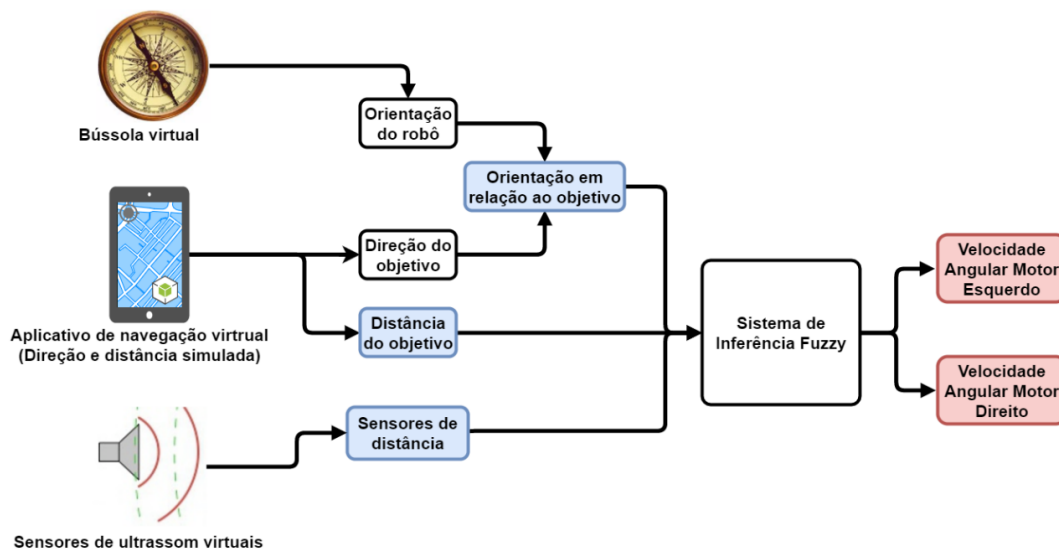


Figura 58 - Diagrama macro das entradas (azul) e saídas (vermelho) do SIF do modelo virtual

Na seção seguinte serão apresentadas as características do simulador Stage que foram editadas para a composição do ambiente de navegação e do robô móvel.

### 3.2.1 Ambiente de Simulação

Conforme apresentado no capítulo anterior, o Stage é um simulador 2D para robôs móveis. Ele é composto por módulos nos quais são adicionadas as propriedades e parâmetros para formar os sensores e atuadores. Todos os sensores, atuadores e objetos do ambiente são formados a partir do módulo básico “*model*”, cujas principais propriedades editáveis são:

- *pose* [ *x*:<float> *y*:<float> *z*:<float> *heading*:<float> ]: especifica a posição do modelo em relação a sua origem.
- *size* [ *x*:<float> *y*:<float> *z*:<float> ]: especifica as dimensões do objeto.

- *origin* [ x:<float> y:<float> z:<float> *heading*:<float> ]: especifica a posição da origem do objeto.
- *bitmap* <string>: desenha um modelo interpretando as linhas da imagem (bmp, jpeg, gif ou png).
- *update\_interval* <int> (valor padrão 100): taxa de atualização em milissegundos.
- *ctrl* <string>: especifica um módulo de controle. Por exemplo, o controle “*lasernoise*” que insere um ruído gaussiano na leitura do módulo “*ranger model*”.
- *obstacle\_return* <int>: especifica se o módulo pode colidir com outros módulos
- *ranger\_return* <float>: determina o grau de reflexibilidade do módulo a um sensor feito a partir de um “*ranger model*”.

Existem subclasses do módulo “*model*” que irão editar as propriedades específicas dos sensores e/ou atuadores a que elas correspondem. Na configuração do ambiente de simulação, para os testes de validação e ajuste do código de navegação fuzzy, foram utilizadas as subclasses “*ranger model*” e “*position model*”.

O “*ranger model*” é utilizado para criar um sensor de distância, seja ele sonar ou laser. Os parâmetros editáveis que permitem aproximar o modelo simulado a um sensor real são:

- *fov* <float>: especifica o campo de visão do sensor;
- *ranger* [min:<float> max:<float>]: especifica alcance mínimo e máximo do sensor;
- *samples* <float>: o número de amostras presente no campo de visão do sensor, ou seja, a resolução angular.

O módulo “*position model*” simula a base de um robô móvel. Por meio desse módulo são configuradas as propriedades de deslocamento e localização do robô móvel. Os principais parâmetros editáveis são:

- *drive* <string>: especifica o tipo de tração da base do robô móvel. A tração pode ser diferencial (“*diff*”), omnidirecional (“*omni*”) ou similar a um carro (“*car*”), ou seja, com roda que esterça.
- *localization* <string>: especifica como ocorrerá a localização do robô no ambiente de simulação, por GPS (“*gps*”) ou odometria (“*odom*”). Caso seja especificado GPS a localização será exata, ou seja, o simulador, por padrão, não possui uma ferramenta para gerar incerteza na informação de localização por GPS.
- *odom\_error* [x:<float> y:<float> heading:<float>]: especifica a quantidade de erro inserido a cada interação do odometro.

A seguir, nas seções 3.2.1.1 e 3.2.1.2, serão apresentados o resultado das configurações dos módulos “*model*”, “*ranger model*” e “*position model*”, para a composição do robô móvel e do ambiente de navegação.

### 3.2.1.1 Robô virtual

Para realizar os ajustes e validação do algoritmo de navegação Fuzzy, conforme já mencionado, foi criado um robô móvel no ambiente do simulador Stage. Esse robô móvel possui uma base diferencial e suas dimensões podem ser observadas na Figura 59. Essas dimensões foram escolhidas como sendo próximas ao do modelo real pretendido.

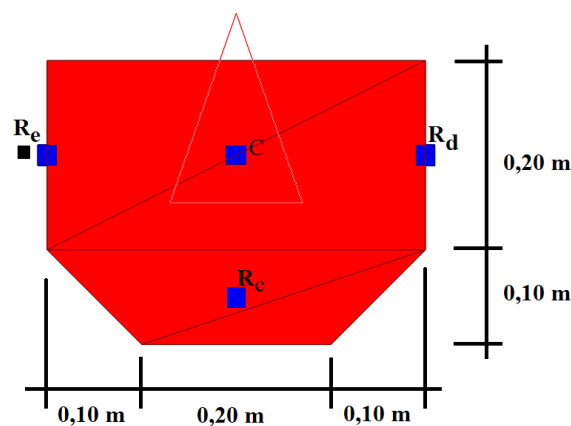


Figura 59 - Representação das dimensões do robô virtual (vista superior)



Na Figura 59, o ponto  $C$  representa o centro de rotação do robô,  $R_e$  a localização da roda esquerda,  $R_D$  a localização da roda direita,  $R_C$  a localização da roda castor e o triângulo central a direção da frente do robô.

O módulo “*position model*”, utilizado na construção da base diferencial do robô virtual, simula as atuações do motor, a localização por GPS e a bússola embarcada, que indica a orientação global do robô. Apesar de ser uma base diferencial, o simulador só permite inserir comandos de velocidade linear e angular para o deslocamento da origem do dispositivo. Portanto, para que a atuação corresponda verdadeiramente a um modelo diferencial, foram aplicadas as equações de cinemática (6) e (8), tendo como entradas as velocidades angulares desejadas para cada roda, e a origem do módulo foi posicionada no centro de rotação do robô, ponto  $C$  da Figura 59.

O robô está embarcado com oito sensores de distância. Cinco desses sensores, S1 a S5, foram utilizados para detectar obstáculos e seu posicionamento pode ser observado na Figura 60. Esses sensores foram configurados, dentro das limitações do simulador, para representar os sensores de ultrassom que serão utilizados na plataforma real. As características do sensor e distância virtual são: campo de visão 40 graus, distância de leitura 6,45 m e resolução angular 2 graus. Os três sensores restantes, S6 a S8, foram posicionados em paralelo aos sensores S1, S3 e S5, respectivamente, porém em uma altura próxima ao chão. Com isso, ele detecta objetos em um nível mais baixo que os sensores S1 a S5.

Para que haja uma diferenciação entre a calçada e os demais objetos do mapa, objetos que não representam calçadas foram posicionados acima da região de atuação dos sensores S6 a S8, mas dentro da região de atuação dos sensores S1 a S5. Os sensores S6 a S8 foram criados com um campo de atuação de 10 graus, distância de leitura 1 m e resolução angular de 2 graus. O procedimento de diferenciação da altura dos objetos que representam o meio-fio foi adotado devido à limitação do simulador de não permitir medir a distância do sensor ao chão.

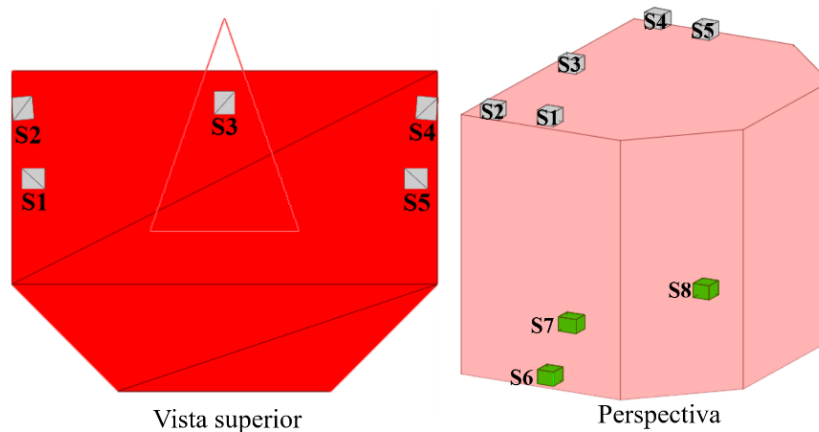


Figura 60 - Representação do robô virtual com os sensores de ultrassom

Por representar um sensor de ultrassom, a resolução angular deve ser o próprio campo de visão, contudo, foi identificada uma falha no software do simulador, no qual ele detecta objetos apenas no eixo de sua resolução. Para que o sensor virtual tenha a representatividade correta em relação à resolução de um sensor de ultrassom real, as saídas das leituras de cada passo da resolução foram combinadas para que o valor mínimo entre elas represente a leitura do sensor. A cada valor de leitura dos sensores de distância foi inserido um ruído gaussiano de desvio padrão de duas vezes a resolução linear do sensor real pretendido, ou seja, 5,08 cm. Na Figura 61 é possível observar o campo de visão dos sensores S1 a S8, onde os sensores que detectam as calçadas, S6 a S8, são aqueles cujos campos de visão estão em verde. Como referência para o tamanho do robô e o alcance do sensor, cada quadrado da grade do fundo da figura possui área de  $1 m^2$ .

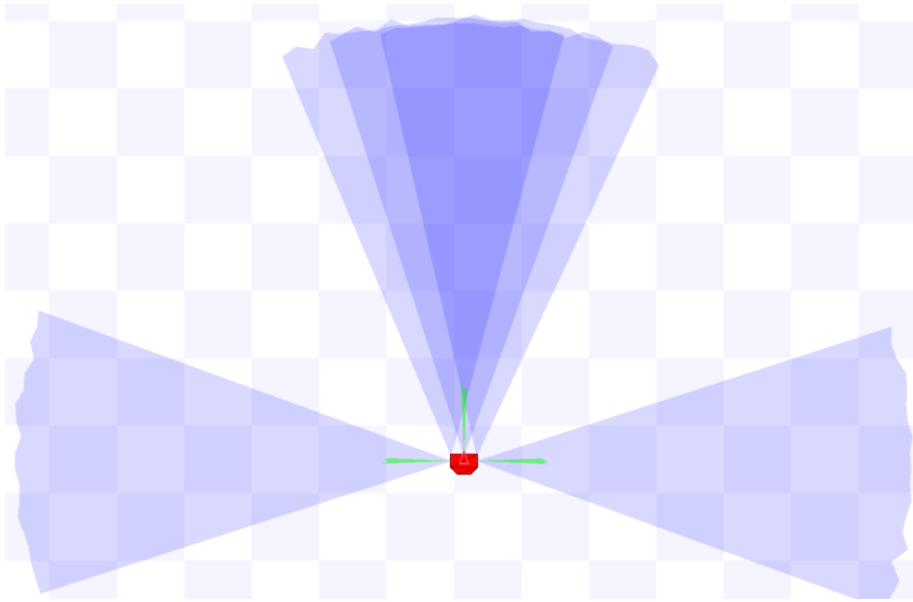


Figura 61 - Ilustração da área de atuação dos sensores de ultrassom do robô virtual

### 3.2.1.2 Ambiente de navegação

Para realizar os testes do algoritmo de navegação foi criado um mapa representando uma região com vários quarteirões. Esse mapa possui área total de 50x50 m, e as calçadas possuem espaçamento de 2 m, conforme Figura 62 e Figura 63. As dimensões do ambiente criado para navegação, bem como os objetos nele colocados, foram escolhidas de maneira a melhor representar um ambiente urbano, que é o objetivo final de navegação do modelo real. Como referência para as dimensões do mapa, assim como na Figura 61, cada quadrado da grade do fundo da figura possui área de  $1\text{ m}^2$ .

Nesse ambiente de navegação foram espalhados diversos objetos para criar regiões de navegação com diferentes graus de complexidade. Por meio da interação do robô móvel com esse ambiente, foi feito o refinamento das regras do controle de navegação.

Na seção seguinte será apresentado o sistema de inferência fuzzy utilizado como algoritmo de navegação do robô.

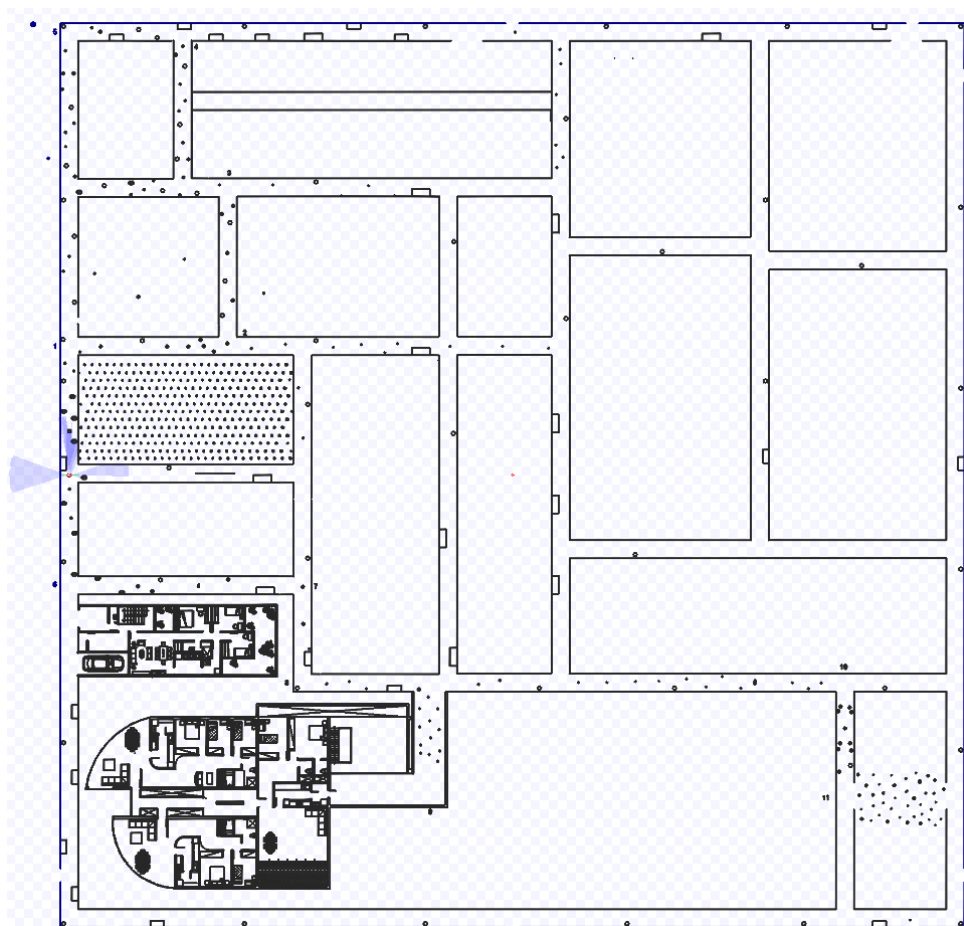


Figura 62 - Mapa de navegação completo

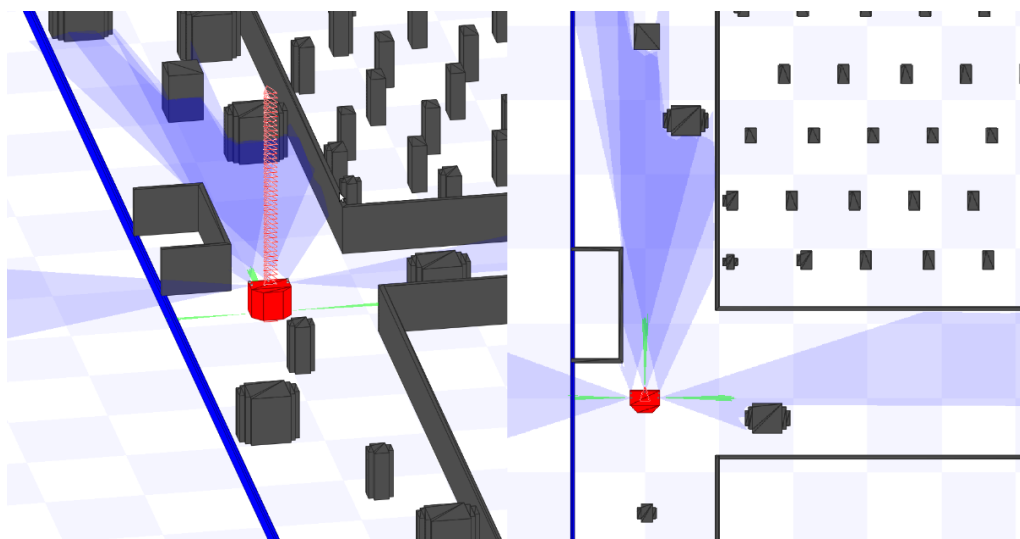


Figura 63 - Mapa de navegação com zoom na região do robô

### 3.2.2 Sistema de Inferência Fuzzy

Para o desenvolvimento do Sistema de Inferência Fuzzy (SIF) foram utilizadas funções de pertinências de formas triangulares e trapezoidais. Essas formas foram as adotadas por facilitarem a escrita do código. O conjunto de regras utilizado foi extraído do conhecimento de um especialista e refinado com a observação da interação do robô com o ambiente. As variáveis fuzzy utilizadas representam os sensores de distância, a orientação em relação ao objetivo, a distância ao objetivo, a velocidade angular desejada no motor esquerdo e a velocidade angular desejada no motor direito. Assim como na Figura 58, é na Figura 64 também é apresentado o diagrama macro do SIF do modelo virtual, contudo, nessa nova representação as informações dos sensores de distância são diferenciadas conforme as aplicações apresentadas na seção 3.2.1.1.

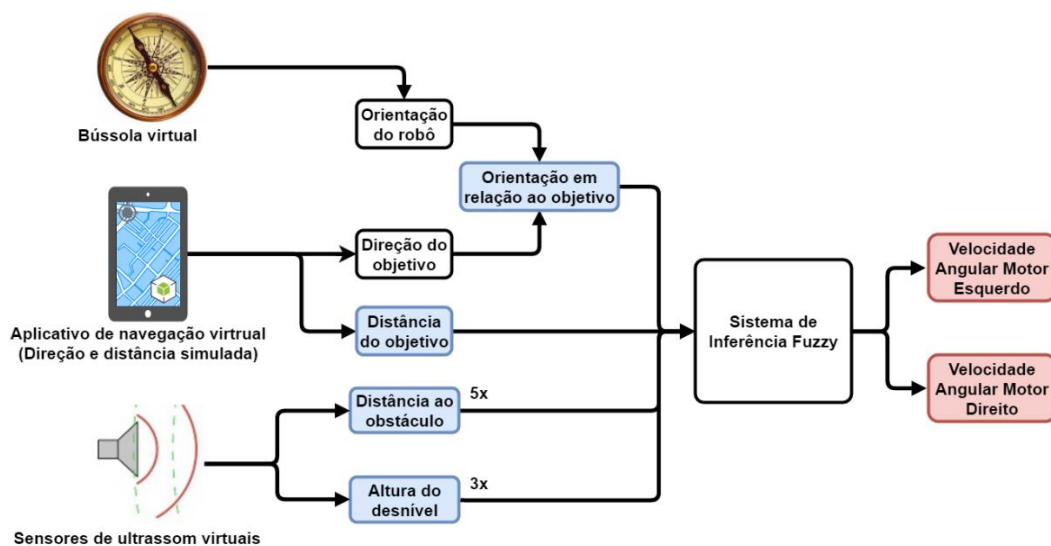


Figura 64 - Diagrama macro do SIF do modelo virtual com a separação dos dados dos sensores de distância

A seguir, nas seções 3.2.2.1 a 3.2.2.3 serão apresentadas as características de cada uma das etapas da criação do SIF.

### 3.2.2.1 Fuzzyficação

As variáveis de entrada utilizadas para o SIF são os dados dos sensores de distância, a orientação em relação ao objetivo e a distância ao objetivo. Já as saídas são os níveis de atuação desejados para cada motor. Os sensores de ultrassom S1 a S5, mostrados na Figura 60, geraram as seguintes variáveis linguísticas:

- S1: Sensor Lateral Esquerda.
- S2: Sensor Frente Esquerda.
- S3: Sensor Frente
- S4: Sensor Frente Direita
- S5: Sensor Lateral Direita.

Cada uma dessas variáveis linguísticas possui três possíveis valores fuzzy: *Muito Perto* (MP), *Perto* (P) e *Longe* (L); e seu universo de discurso vai de 0 a 1,5 m. Como o alcance do sensor de distância é superior ao universo de discurso da variável que o representa, caso o sensor detecte um objeto a uma distância maior que 1,5 m o valor lido será trocado pela distância de 1,5 m. Na Figura 65 estão representadas as funções de pertinência de cada um dos valores das variáveis linguísticas dos sensores de distância.

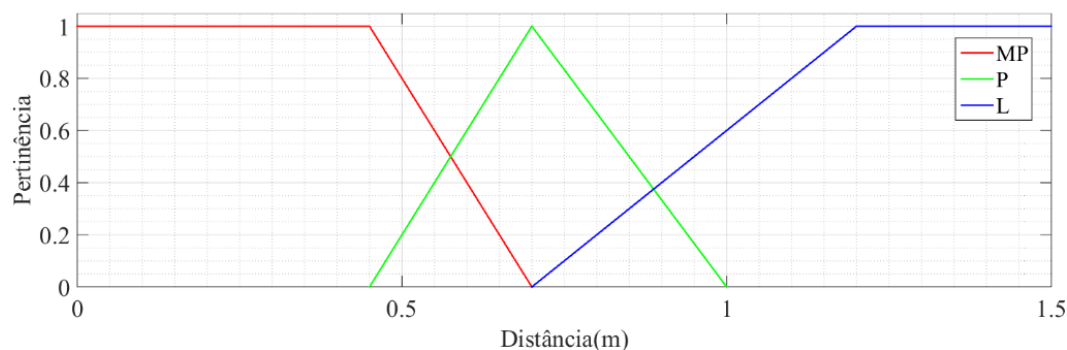


Figura 65 - Funções de pertinência das variáveis do tipo Sensor

Os sensores de distância S6 a S8, utilizados para detectar a divisão entre a calçada e a rua, geraram as seguintes variáveis linguísticas:

- S6: Sensor Altura Esquerda.
- S7: Sensor Altura Frente.
- S8: Sensor Altura Direita.

Cada uma das variáveis do tipo *Sensor Altura* possui dois valores fuzzy possíveis, *Baixa* (BX) e *Alta* (AL), cujas funções de pertinência podem ser observadas na Figura 66. No desenvolvimento das regras foi considerada como uma zona segura os valores de distância *Baixa*. Portanto, o robô tenderá a manter uma distância acima de 50 cm do meio-fio, para o caso do modelo virtual.

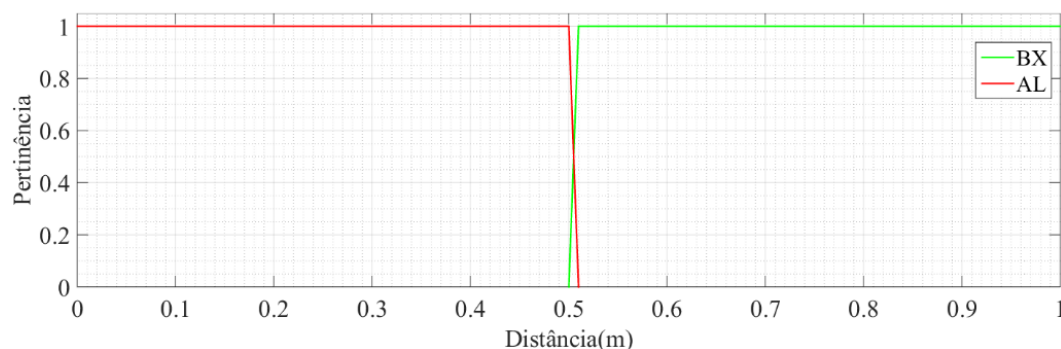


Figura 66 - Funções de pertinência das variáveis do tipo Sensor Altura

A orientação do dispositivo em relação ao objetivo é representada pela variável linguística *Ângulo*. A variável *Ângulo* possui cinco valores fuzzy possíveis: *Negativo Grande* (NG), *Negativo Médio* (NM), *Zero* (Z), *Positivo Médio* (PM) e *Positivo Grande* (PG); e seu universo de discurso vai de  $-180^{\circ}$  a  $180^{\circ}$ . As funções de pertinência de cada um dos valores da variável *Ângulo* estão representadas na Figura 67; e na Figura 68 está a indicação das regiões de valores precisos de ângulos positivos e negativos de orientação ao objetivo. O valor preciso de orientação ao objetivo é obtido pela diferença entre a orientação do robô e a direção global do objetivo, ou seja, em relação ao norte.

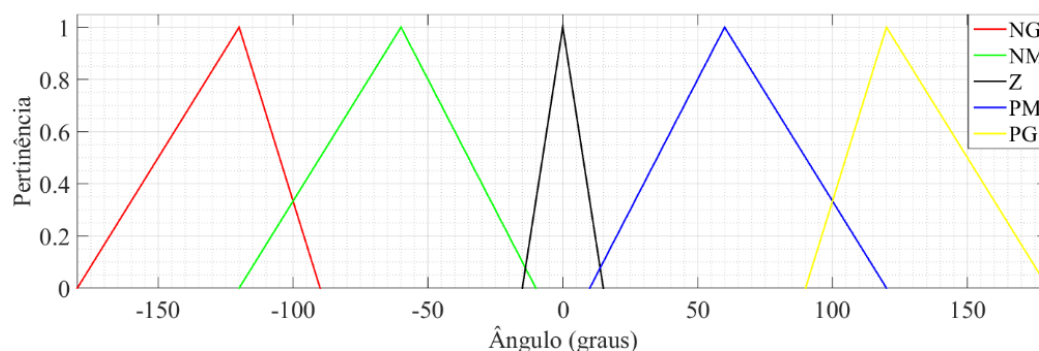


Figura 67 - Funções de pertinência da variável Ângulo

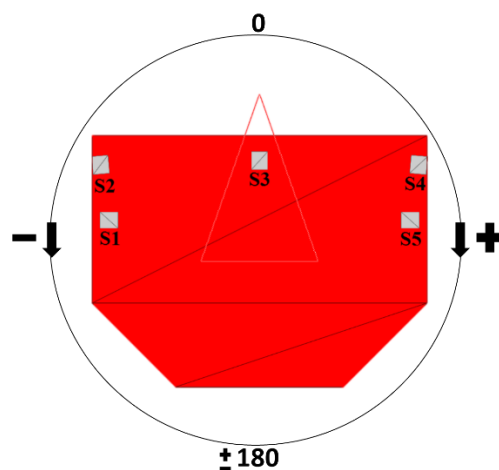


Figura 68 - Indicação dos valores precisos de ângulos positivos e negativos de orientação ao objetivo

A distância do robô ao objetivo é representada pela variável linguística *Distância*. Essa variável possui dois valores fuzzy possíveis: *Perto* (P) e *Longe* (L); e seu universo de discurso vai de 0 a 3 m. Assim como no caso dos sensores de ultrassom, ao se obter um valor de distância acima do universo de discurso da variável linguística *Distância*, esse valor é substituído pelo limite do universo de discurso, ou seja, 3 m. Na Figura 69 estão representadas as funções de pertinência dos valores fuzzy da variável *Distância*.

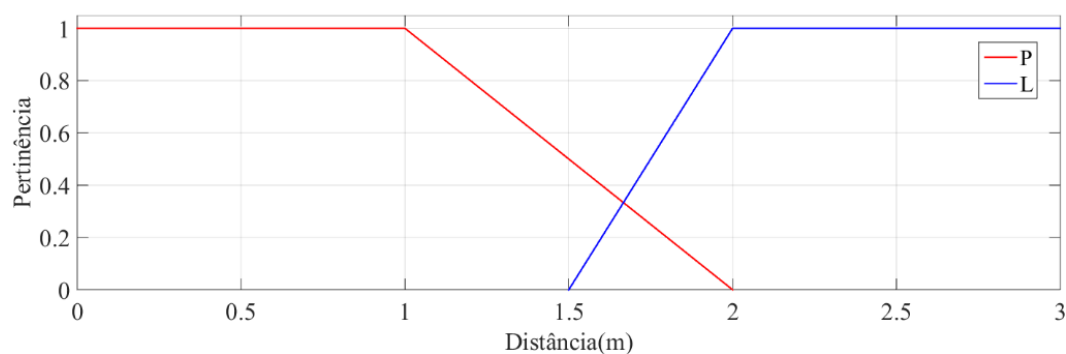


Figura 69 - Funções de pertinência da variável Distância

Para representar a atuação desejada nos motores esquerdo e direito foram, respectivamente, criadas as variáveis linguísticas *MotorE* e *MotorD*. As variáveis linguísticas dos motores possuem sete valores fuzzy possíveis cada: *Negativo Alto* (NA), *Negativo Médio* (NM), *Negativo Baixo* (NB), *Zero* (Z), *Positivo Baixo* (PB), *Positivo Médio* (PM) e *Positivo Alto* (PA); e seu universo de discurso vai de -1



$rad/s$  a  $1 rad/s$ . Essa maior quantidade de valores fuzzy, em relação às outras variáveis do SIF, foi adotada para tentar gerar maior suavidade na transição de atuação, devido às mudanças de regras. Na Figura 70 estão representadas as funções de pertinência de cada um dos valores das variáveis linguísticas dos motores.

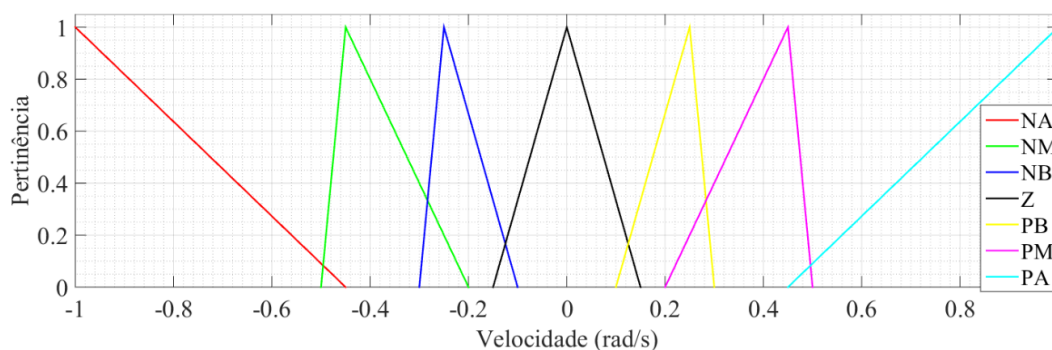


Figura 70 - Funções de pertinência das variáveis MotorE(D)

### 3.2.2.2 Base de regras

Conforme exposto anteriormente, a base de regras foi criada a partir do conhecimento de um “especialista”. Ela foi escrita visando uma navegação com conversões suaves, com pouca oscilação no valor de velocidade angular, uma velocidade linear média próxima de  $0,7 m/s$  e que não executasse movimentos de ré. Essas características de navegação foram consideradas para que o dispositivo não gere desequilíbrio no usuário.

Para combinar os termos dos antecedentes das regras, foram utilizados os conectores “E” e “OU”. O operador de norma triangular aplicado para representar o conector “E” foi o mínimo e o de co-norma triangular para o conector “OU” foi o máximo. Para representar a implicação da regra utilizou-se o operador mínimo e para combinar as ativações dos consequentes das regras, o operador máximo. Cada regra possui dois conjuntos em seu consequente, que representam os valores desejados para cada motor.

Na composição das regras, o conector “OU” foi aplicado apenas para combinar os valores fuzzy da variável *Ângulo*. O resultado dessa operação, assim como os demais termos na regra, é combinado pelo conector “E”. Por exemplo, a regra linguística representada pela equação (54) é reescrita com os operadores mínimo e máximo, conforme a equação (55).

$$Se[(\hat{Angulo}_{NG} \text{ OU } \hat{Angulo}_{NM}) \text{ E } Sensor \text{ Frente}_L] \rightarrow \begin{cases} MotorE_{PM} \\ MotorD_{PB} \end{cases} \quad (54)$$

$$Mín \rightarrow \begin{cases} Mín_E[Máx_{OU}(\hat{Angulo}_{NG}, \hat{Angulo}_{NM}), Sensor \text{ Frente}_L], \\ MotorE_{PM} \\ MotorD_{PB} \end{cases} \quad (55)$$

Na Figura 71, é exibido um trecho do conjunto de regras com a indicação de seus operadores. No total foram escritas 518 regras de 19440 possíveis. A tabela com todas as regras escritas pode ser observada no apêndice 8.1.

O raciocínio empregado para a criação das regras foi sempre tentar realizar desvios de obstáculos direcionados ao objetivo. Por essa razão, as regras de navegação e desvio não foram separadas. No exemplo da Figura 71, as regras 114 e 115 exemplificam esse raciocínio. Em ambas as regras, o robô possui um obstáculo localizado no seu lado direito, tanto à frente quanto na lateral, e seu movimento de desvio o obriga a virar para a esquerda. Contudo, quando mais à esquerda estiver o seu objetivo, ou seja, quanto mais negativa for a variável  $\hat{Angulo}$ , maior será o seu desvio. Vale observar que em ambos os casos o desvio empregado é considerado suficiente para evitar a colisão, mas se torna mais acentuado quando coincide com o objetivo. Outro ponto a destacar na metodologia de criação das regras é que procurou-se evitar realizar comandos que resultassem em um movimento de rotação puro e que a atuação negativa de um motor fosse mais acentuada que a atuação positiva do outro. A primeira medida foi adotada para evitar configurações nas quais o robô ficasse preso em um movimento oscilatório, já a segunda, foi para que a soma das regras nunca resultasse em uma velocidade linear negativa, ou seja, que o dispositivo realizasse um movimento de ré.

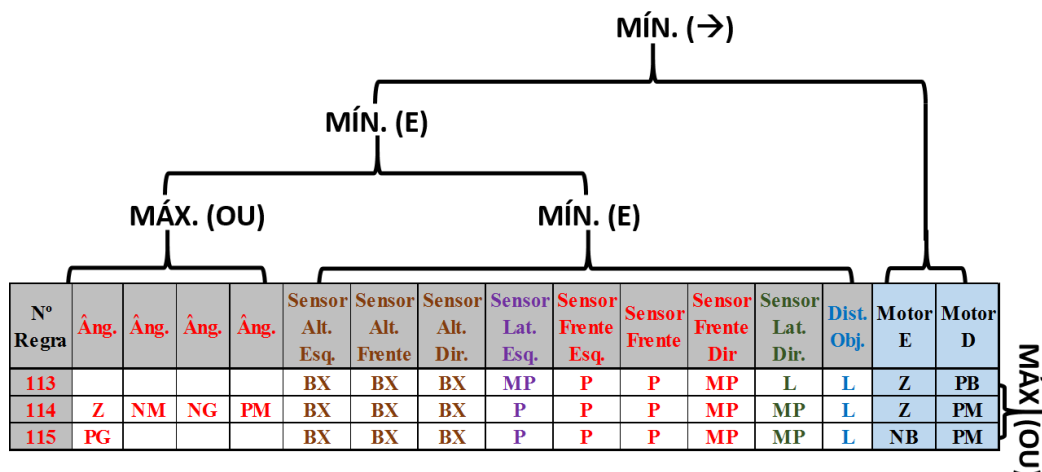


Figura 71 - Exemplo da base de regras com indicação dos operadores

### 3.2.2.3 Defuzzyficação

Para realizar a operação de defuzzyficação dos conjuntos resultantes das regras ativadas, utilizou-se a média ponderada da média dos máximos, ou seja,

$$Defuzzy_{MotorE(D)} = \frac{\sum_{i=NA}^{PA} \mu_i * MédiaMax_i}{\sum_{i=NA}^{PA} \mu_i} \quad (56)$$

onde  $i$  representa os valores fuzzy da variável MotorE(D),  $\mu_i$  a pertinência resultante do valor fuzzy  $i$  e  $MédiaMax_i$  a média dos máximos da função de pertinência do valor fuzzy  $i$ .

Esse método foi utilizado devido a sua simples implementação, quando comparado ao método do Centroide, e por ter maior representatividade em relação as ativações dos conjuntos que o método da Média dos Máximos.

No próximo capítulo será apresentada a aplicação e desenvolvimento do modelo em uma plataforma real. Serão apresentados os sensores, atuadores e hardware processador utilizados, bem como as configurações aplicadas a cada um dos dispositivos. Também serão apresentados os ajustes realizados no SIF apresentado nesse capítulo.

## 4

## Desenvolvimento do dispositivo robótico – Aplicação real

Nesse capítulo serão apresentadas todas as partes da modelagem do protótipo desenvolvido. Serão abordados os sensores, atuadores e o hardware processador utilizado, assim como as configurações e procedimentos para calibração aplicadas a cada um deles. Também serão apresentadas as configurações do servidor Player e os drivers criados para permitir o acesso aos diferentes tipos de hardware utilizados. Outro ponto que será abordado são as características da estrutura do dispositivo que foi montada para os testes de navegação e locomoção.

### 4.1

### Projeto Mecânico

O projeto da estrutura do protótipo, nesta fase de desenvolvimento, foi elaborado objetivando as funcionalidades de navegação descritas no capítulo 3. Portanto, a estrutura não suporta os esforços realizados por um usuário que deseja manter ou reestabelecer seu equilíbrio ou as condições não ideais de um ambiente externo como, por exemplo: umidade, sujeira e caminhos muito acidentados.

Para elaborar o projeto da estrutura do protótipo foram estabelecidos os requisitos visando o deslocamento do mesmo em ambiente urbano. Segundo a norma NBR 9050:2015 da Associação Brasileira de Normas Técnicas (Associação Brasileira de Normas Técnicas, 2015), a altura de um degrau isolado deve estar entre 16 cm e 18 cm e o espaçamento mínimo entre obstáculos deve ser de 80 cm ou 90 cm, dependendo do comprimento do mesmo. Assim, considerando esse requisito, as rodas foram dimensionadas com raio de 15 cm e espaçamento entre elas de 45 cm. Outra condição estabelecida no projeto foi que a estrutura permitisse o reposicionamento dos sensores de distância, com o intuito de facilitar a transição do sistema virtual para a plataforma real.

Baseado nessas premissas, foi criado, pela equipe de design do projeto, o modelo do protótipo em ambiente CAD (*Computer Aided Design*). Na Figura 72 a Figura 75 estão representadas as vistas frontal, superior, lateral esquerda e em

perspectiva do protótipo modelado. Todas as cotas representam as dimensões em milímetros.

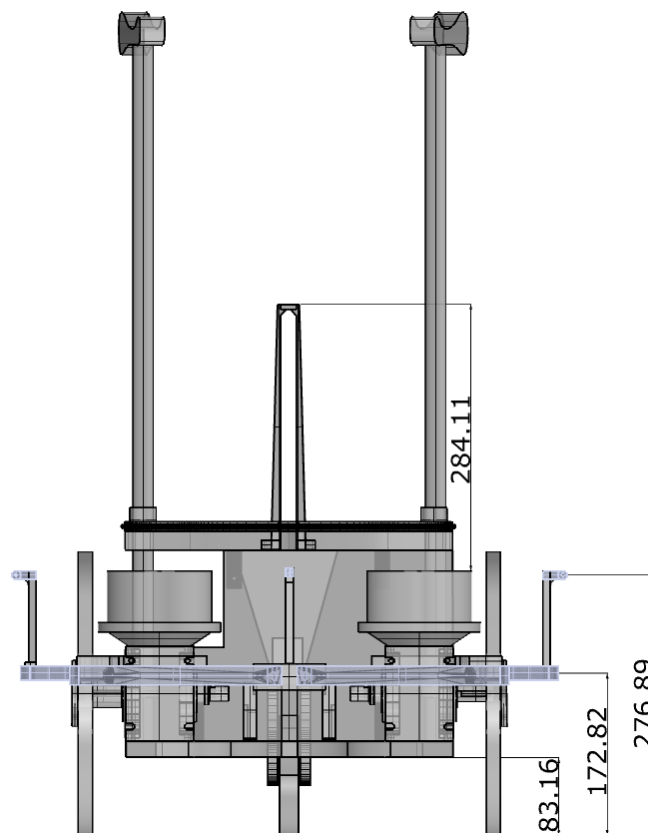


Figura 72 - Vista frontal modelo CAD

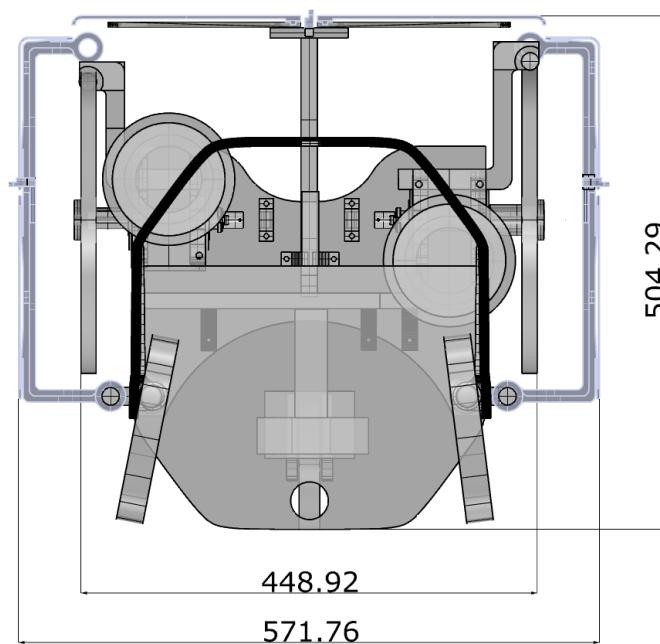


Figura 73 - Vista superior modelo CAD

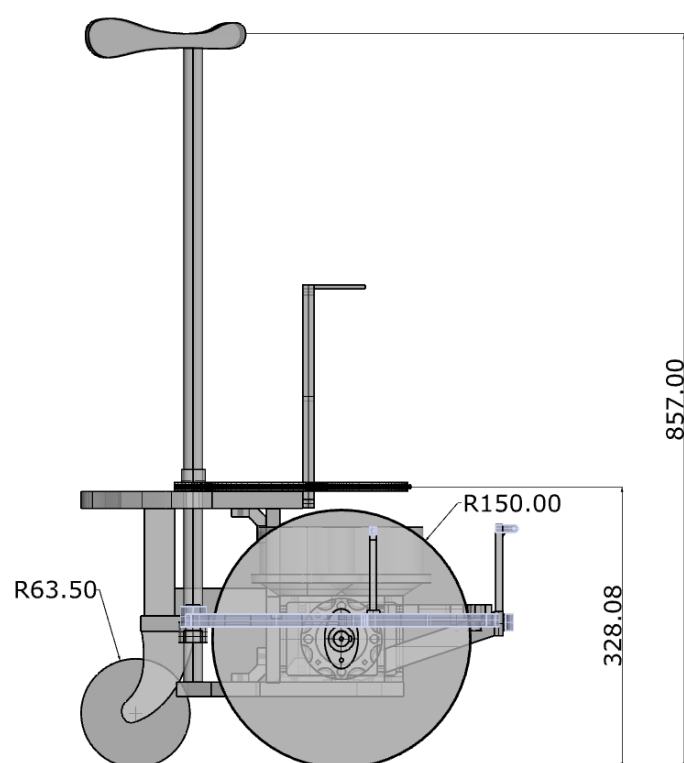


Figura 74 - Vista lateral esquerda modelo CAD

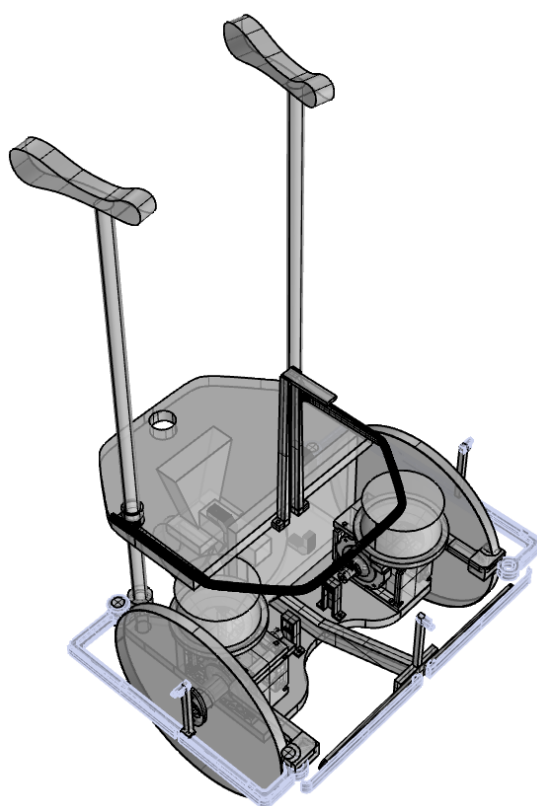


Figura 75 - Vista em perspectiva modelo CAD

O material utilizado para a construção do protótipo foi em grande parte madeira, do tipo compensado e MDF (*Medium Density Fiberboard*). As partes que exigiam maior exatidão no design, como os encaixes dos sensores, foram produzidas em plástico ABS (*Acrylonitrile Butadiene Styrene*) por meio de uma impressora 3D. Na Figura 76 encontram-se destacadas em amarelo as peças que foram impressas. A escolha dos materiais foi baseada apenas na praticidade e facilidade de manipulação para a construção. Assim, nesse primeiro momento, o protótipo não possui a resistência necessária para operar de maneira segura com uma pessoa idosa, mas é suficientemente resistente para a realização dos testes de navegação e locomoção que irão validar o modelo.

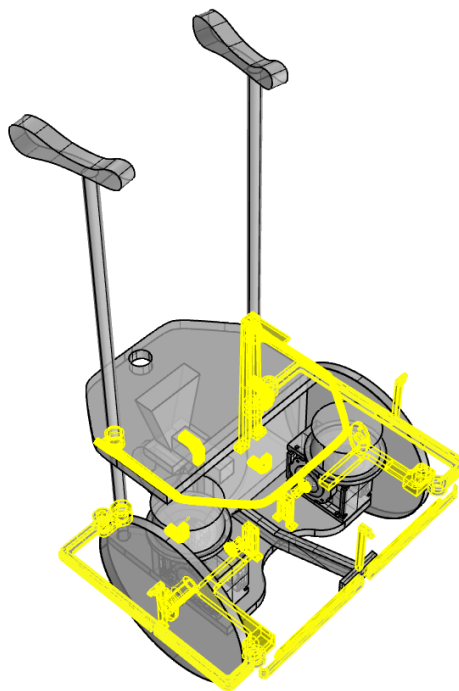


Figura 76 - Vista em perspectiva com destaque em amarelo para peças impressas

O resultado da construção dessa estrutura pode ser observado na Figura 77 a Figura 79, que apresentam as vistas frontal, superior e lateral esquerda do protótipo, respectivamente. Vale destacar que para a área denominada “pega do usuário”, ou seja, o local onde o usuário deverá usar como apoio, foram utilizadas duas bengalas comerciais modelo Dilepe DB-108 (Americans, 2016). A massa final do protótipo, somando a estrutura e os componentes, é de 18 kg. Essa massa está dentro das recomendações apresentadas no quadro 1, contudo pode ser facilmente reduzida com a utilização de materiais mais apropriados para a estrutura, como alumínio e fibra de carbono, e motores mais adequados, como será visto mais adiante.

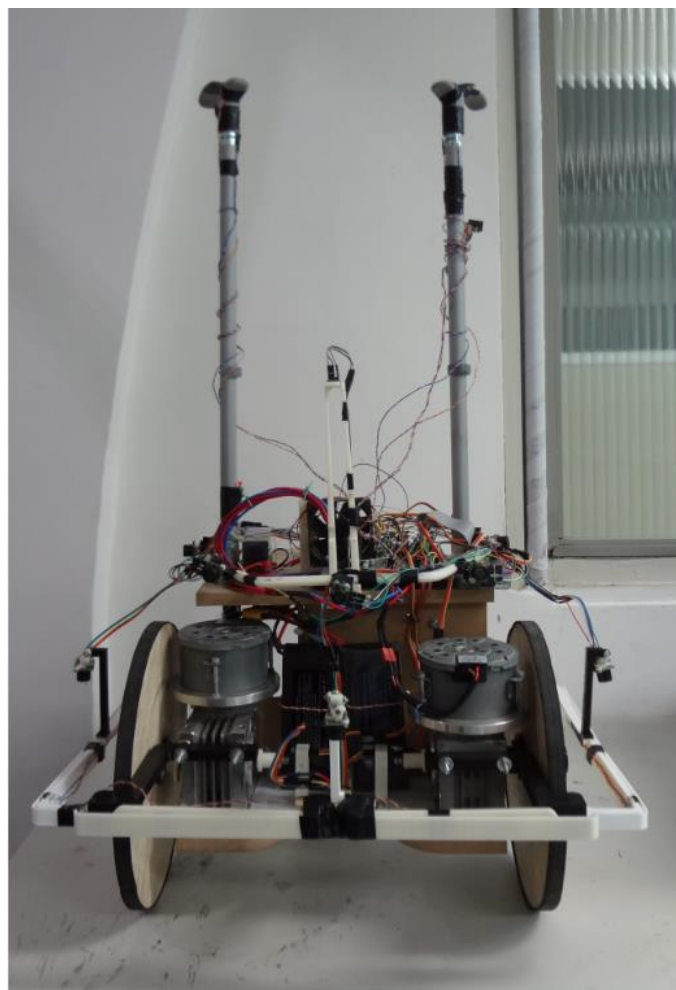


Figura 77 - Vista frontal

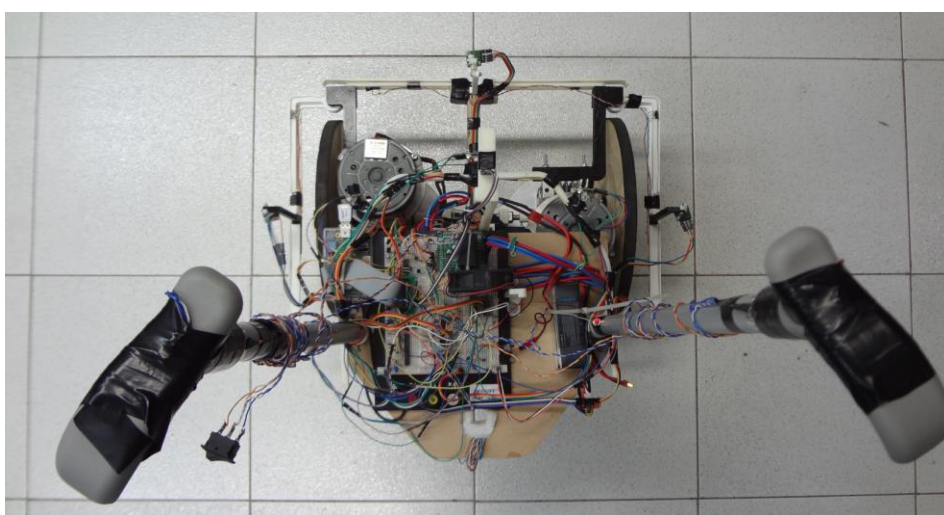


Figura 78 - Vista superior



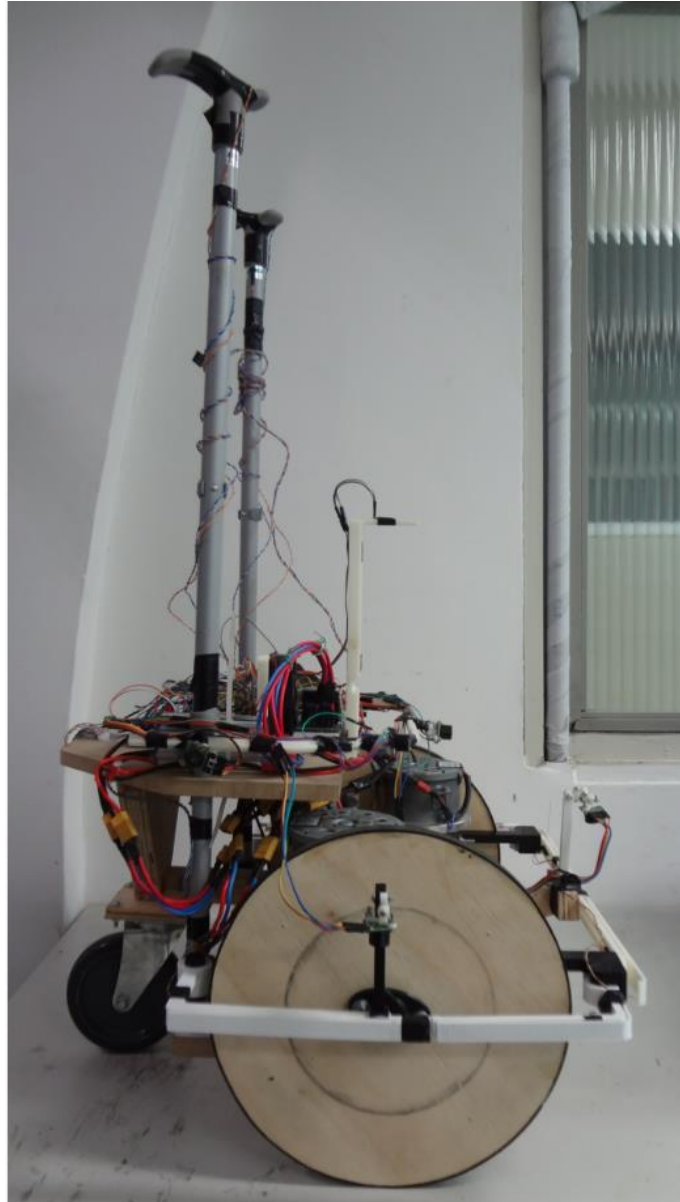


Figura 79 - Vista lateral esquerda

## 4.2 Projeto eletrônico

Nessa seção serão abordados todos os aspectos referentes ao hardware utilizado no desenvolvimento do protótipo. Serão especificados os componentes utilizados, os procedimentos para calibração adotados, os drivers criados para acessá-los e as conexões realizadas no processo de construção da eletrônica do protótipo.

## 4.2.1 Atuadores

Nessa primeira subseção serão abordadas as informações referentes aos atuadores do protótipo. Serão especificados os atuadores empregados no protótipo e identificados os parâmetros necessários para o desenvolvimento do controle de velocidade, que será apresentado na subseção 4.3.1.

### 4.2.1.1 Motor CC

Conforme exposto no capítulo 2, o tipo de motor CC empregado na atuação das rodas diferenciais foi o motor CC com escovas de excitação por ímãs permanentes. O modelo com escovas foi escolhido por apresentar menor custo que o sem escovas, e a excitação por ímãs permanentes, por ter menor custo e ser mais compacta, quando comparada às demais arquiteturas com escova.

Com base nas condições de contorno para o deslocamento apresentadas no quadro 1, o conjunto de motores deverá ser tal que o dispositivo consiga executar uma velocidade linear entre  $0,62 \text{ m/s}$  e  $1,2 \text{ m/s}$  e suportar ângulos de subida de até  $34,5^\circ$  e descida de até  $21^\circ$ . A partir da equação (6) e das dimensões do protótipo apresentadas na seção 4.1, conclui-se que o motor deve ser capaz de realizar uma rotação entre  $4,13 \text{ rad/s}$  e  $8 \text{ rad/s}$  ou entre 40 RPM e 76 RPM.

Para o cálculo do torque necessário para a subida e descida da rampa, foi feita a hipótese simplificada de que a força exercida pelo usuário sobre o protótipo represente apenas um acréscimo a sua força peso e que não há deformação e nem deslizamento da roda em relação ao solo. Com isso, a força exercida pelo usuário não gera nenhum termo de momento, as perdas devido à resistência ao rolamento podem ser desprezadas e o atrito presente entre a roda e o solo é apenas o estático. Considerando esse modelo simplificado, supondo que um usuário de massa 70 kg consiga transferir 40% de sua força peso, através do esforço exercido sobre o protótipo, o dispositivo terá um acréscimo de 274,4 N ao seu peso. A partir dessas características, tem-se o diagrama de corpo livre conforme a Figura 80. Nessa hipótese simplificada, a soma da força exercida pelo usuário e da força peso do protótipo, que possui massa de 18 kg, é de 450,8 N, equação (57), e a componente da força peso contrária à subida 255,3 N, equação (58). Portanto, o torque mínimo

necessário para cancelar a componente contrária à subida, para cada motor, considerando a roda de raio 0,15 m, é de 19,15 Nm, equação (59).

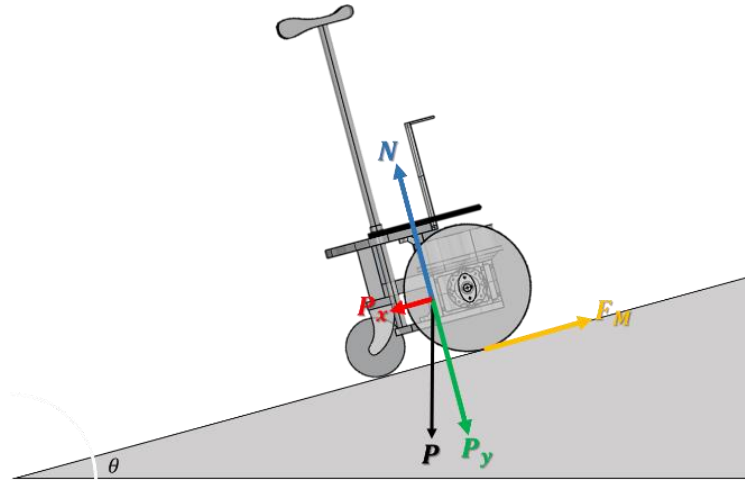


Figura 80 - Diagrama de corpo livre do protótipo subindo uma rampa

Na Figura 80,  $N$  é a força normal,  $P$  é a força resultante da combinação da força peso do dispositivo com a força exercida pelo usuário,  $P_y$  é a componente da força  $P$  no sentido oposto à força normal,  $P_x$  é a componente da força  $P$  que se opõe à subida da rampa e  $F_M$  é a força de reação do solo sobre a roda, gerada pela presença do atrito estático, devido à força resultante que o torque dos dois motores exercem sobre o solo.

$$P = (0.4m_{us} + m_{disp}) * g, \quad (57)$$

$$P_x = P \sin \theta \quad (58)$$

e

$$F_M = -(\tau_{Me} + \tau_{Md}) * R, \quad (59)$$

onde  $m_{us}$  é a massa total do usuário,  $m_{disp}$  a massa total do dispositivo,  $g$  a aceleração da gravidade,  $\tau_{Me}$  é o torque exercido pelo motor esquerdo,  $\tau_{Md}$  o torque exercido pelo motor direito e  $R$  o raio das rodas.

Para atender às condições listadas acima, uma opção é o conjunto motor-redutor modelo CEP F 006 WM0 310 da fabricante Bosch, cujas especificações se encontram na Figura 81. Esse conjunto motor-redutor, além de atender as condições de contorno, possui um baixo consumo de corrente e um peso relativamente baixo,

o que o torna ideal para a tarefa proposta. Contudo, apesar das características favoráveis, não foi encontrada disponibilidade de venda desse produto.

### CEP

**24 V 46 W**



$U_N$	24 V
$P_N$	46 W
$n_N$	45 rpm
$I_N$	5,0 A
$I_{MAX}$	18,6 A
$M_N$	10 N.m
$M_A$	48 N.m
$i$	63:1
Rot.	L / R
S	S1
IP	IP 44
kg	1,100 kg
Ⓢ	<b>F 006 WM0 310</b>

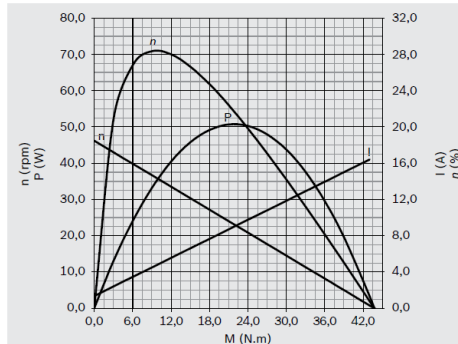


Figura 81 - Especificações motor CEP F 006 WM0 310

Devido à falta de disponibilidade para compra do modelo sugerido e de modelos com características semelhantes, foi utilizado o motor de Bosch modelo GPB F 006 KM0 60N (Figura 82) juntamente com uma caixa de redução MR4 1:10 63 B14 (Figura 83) da fabricante MKS. Esse conjunto está superdimensionado para o problema em questão, sendo capaz de suportar, na condição hipotética anteriormente apresentada, toda a massa do idoso.

### GPB

**24 V 175 W**



$U_N$	24 V
$P_N$	175 W
$n_N$	2.600 rpm
$I_N$	12,5 A
$M_A$	506 N.cm
Rot.	R
S	S1
IP	IP 03
kg	1,350 kg
Ⓢ	<b>F 006 KM0 60N</b>

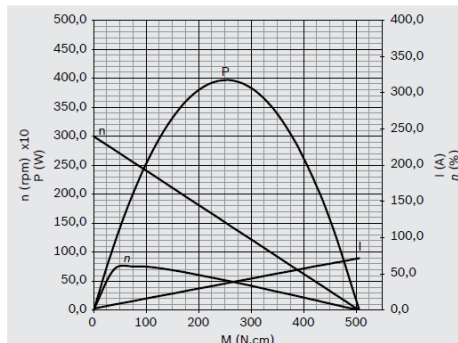


Figura 82 - Especificações motor GPB F 006 KM0 60N


	<b>Especificações Técnicas</b>	
	<b>Redução</b>	1/10
	<b>Potência de Entrada</b>	Até 3/4 cv
	<b>Torque de Saída</b>	Até 39Nm
	<b>Rotação de Saída</b>	Até 250 rpm

Figura 83 - Especificações redutor MKS MR4 1:10 63 B14

Com o conjunto motor-redutor escolhido, a próxima etapa foi obter seus parâmetros para poder estabelecer seu modelo. Esse modelo será utilizado para criação e sintonia do controle de velocidade que, conforme já mencionado, será descrito na subseção 4.3.1.

Com base nos dados fornecidos pelos gráficos das curvas de resposta do motor (Figura 82) e das equações (14) a (20), foi possível especificar grande parte dos parâmetros que descrevem o comportamento do motor. Esses parâmetros são apresentados na Tabela 2, que também especifica como foram obtidos, ou seja, da leitura direta do gráfico das curvas características do motor ou a partir das equações (14) a (20), assumindo a condição de parada, regime permanente ou rotação sem carga do conjunto motor-redutor.

Tabela 2 – Lista de parâmetros do motor GPB F 006 KM0 60N

Parâmetro	Valor	Condição
$I_{stall}$	71,053 A	Leitura direta gráfico da Figura 82
$I_{no\ load}$	1,974 A	Leitura direta gráfico da Figura 82
$\omega_{no\ load}$	314,159 rad/s	Leitura direta gráfico da Figura 82
$K_t$	0,0712 Nm/A	Condição de Parada
$K_e$	0,0712 Vs/rad	Regime Permanente
$R_a$	0,82445 $\Omega$	Sem carga
$B_M$	$4,747 * 10^{-4}$ Nm <sup>2</sup> /s	Sem carga

Na Tabela 2,  $I_{stall}$  é a corrente no motor travado,  $I_{no\ load}$  a corrente do motor rodando sem carga e  $\omega_{no\ load}$  a velocidade do motor rodando sem carga.

Os parâmetros de inércia do motor,  $J_M$ , e da indutância da armadura,  $L_a$ , foram obtidos de maneira experimental. Contudo, como o motor foi comprado

acoplado à caixa de redução, optou-se em estimar a inércia equivalente,  $J_{eq}$  (equação (60)), para não ser necessário desmontar o conjunto motor-redutor.

$$J_{eq} = J_M + \left(\frac{n_2}{n_1}\right)^2 J_{red} \quad (60)$$

onde  $\left(\frac{n_2}{n_1}\right)$  é a relação entre a rotação de saída e a de entrada do redutor e  $J_{red}$  é a inércia enxergada pelo redutor, ou seja, a soma da própria inércia com a da carga acoplada.

O mesmo raciocínio foi adotado para a constante de amortecimento devido ao atrito viscoso do conjunto motor-redutor. O valor de  $B_M$ , anteriormente obtido, representa as perdas devido ao atrito apenas do motor. O valor equivalente,  $B_{eq}$ , também será estimado a partir dos dados do conjunto motor-redutor.

Para estimar os parâmetros de constante de amortecimento e inércia equivalentes do conjunto motor-redutor, foram armazenados os valores da velocidade angular e corrente do motor para diferentes níveis de tensão de entrada. Esses valores foram obtidos por meio do circuito criado para o acionamento e controle de velocidade dos motores, descrito na seção 4.2.4. Esse circuito possui *encoders*, para obter a velocidade angular dos motores, e sensores de corrente, para obter a corrente consumida por cada motor. Maiores detalhes sobre esses e os demais componentes dos circuitos de controle e aquisição de dados do motor serão apresentados nas seções seguintes.

N Figura 84 é possível observar os dados da tensão de entrada, velocidade angular executada e corrente consumida, todas em função do tempo, do conjunto motor-redutor sem carga. Os dados desses gráficos foram aplicados no modelo do motor-redutor desenvolvido no software *Simulink*<sup>®</sup> (MathWorks, Inc., 2016a) para, a partir da ferramenta de otimização de parâmetros fornecida pelo software MATLAB (MathWorks, Inc, 2016b), estimar os valores da inércia equivalente, constante de amortecimento equivalente e indutância da armadura.

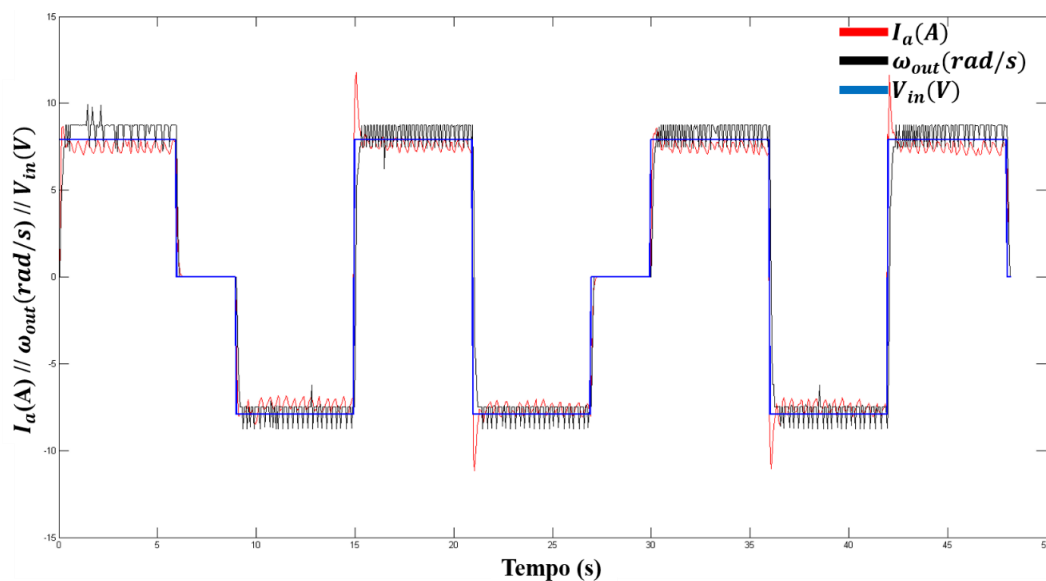


Figura 84 - Tensão de entrada (azul), velocidade executada (preto) e corrente consumida (vermelho) do conjunto motor-redutor do protótipo

O modelo utilizado para simular o sistema motor-redutor é semelhante ao apresentado na Figura 17 da seção 2.2.1.1 e pode ser observado na Figura 85. As principais diferenças entre eles são a inclusão do fator de redução no torque exigido pela carga e na velocidade angular de saída do sistema, além da substituição da inércia e constante de amortecimento do motor pelos valores equivalentes do conjunto motor-redutor.

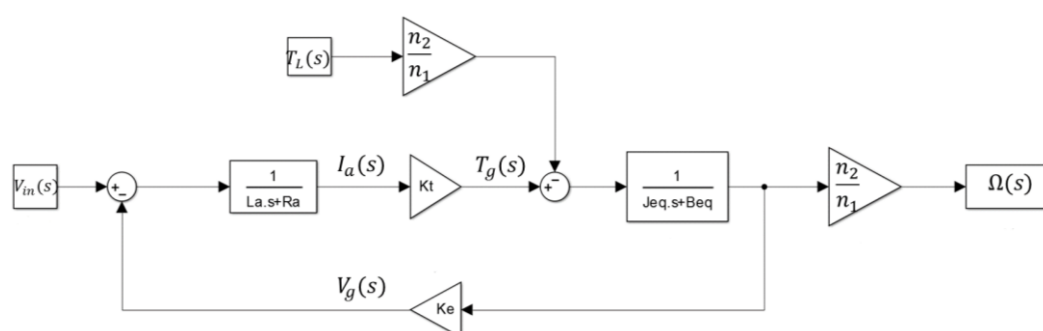


Figura 85 - Diagrama em blocos do modelo simulado para o conjunto motor-redutor

Para o valor inicial da estimativa de  $B_{eq}$  foi utilizado o valor calculado para  $B_M$ . Já para os valores de  $L_a$  e  $J_{eq}$  foram utilizados, como palpite inicial, os valores  $0,1\text{ H}$  e  $0,001\text{ kgm}^2$ , respectivamente. Ao executar a simulação do modelo motor-redutor, com os valores dos demais parâmetros conforme a Tabela 2, e tensão de entrada conforme a apresentada na Figura 84, obteve-se como saída os gráficos

apresentados na Figura 86. Como pode ser observado na Figura 86, os valores de velocidade angular e, principalmente, corrente consumida pelo motor, apresentam erros em relação aos valores medidos do motor real.

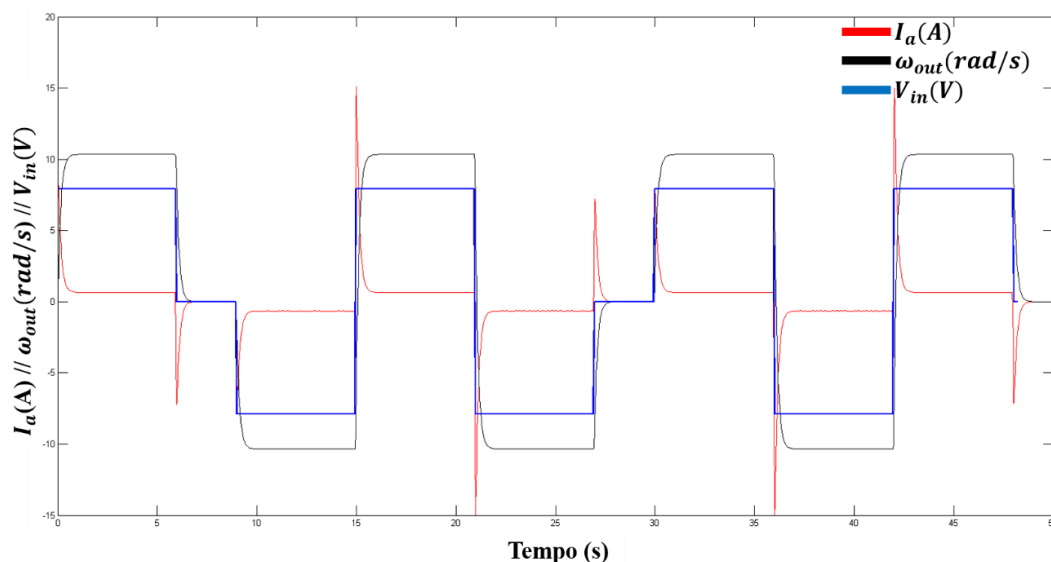


Figura 86 - Gráficos da tensão de entrada (azul), velocidade de saída (preto) e corrente consumida (vermelho) do modelo motor-reductor com os parâmetros da Tabela 2

Ao executar a ferramenta de otimização de parâmetros do software *Simulink*®, para obter novas estimativas de  $J_{eq}$ ,  $B_{eq}$  e  $L_a$ , obteve-se como resultado os valores da Tabela 3. Esses valores ainda geram erros entre o modelo real e o simulado. A comparação entre eles pode ser observada na Figura 87.

Tabela 3 - Primeiro conjunto de parâmetros estimados

Variável	Valor Original	Valor Estimado	Faixa para otimização
$B_{eq}(\text{Nm}^2/\text{s})$	0,000447	0,0056556	0 - 1
$J_{eq}(\text{kgm}^2)$	0,001	0,0011023	0 - 1
$L_a(\text{H})$	0,01	0,044153	0 - 1



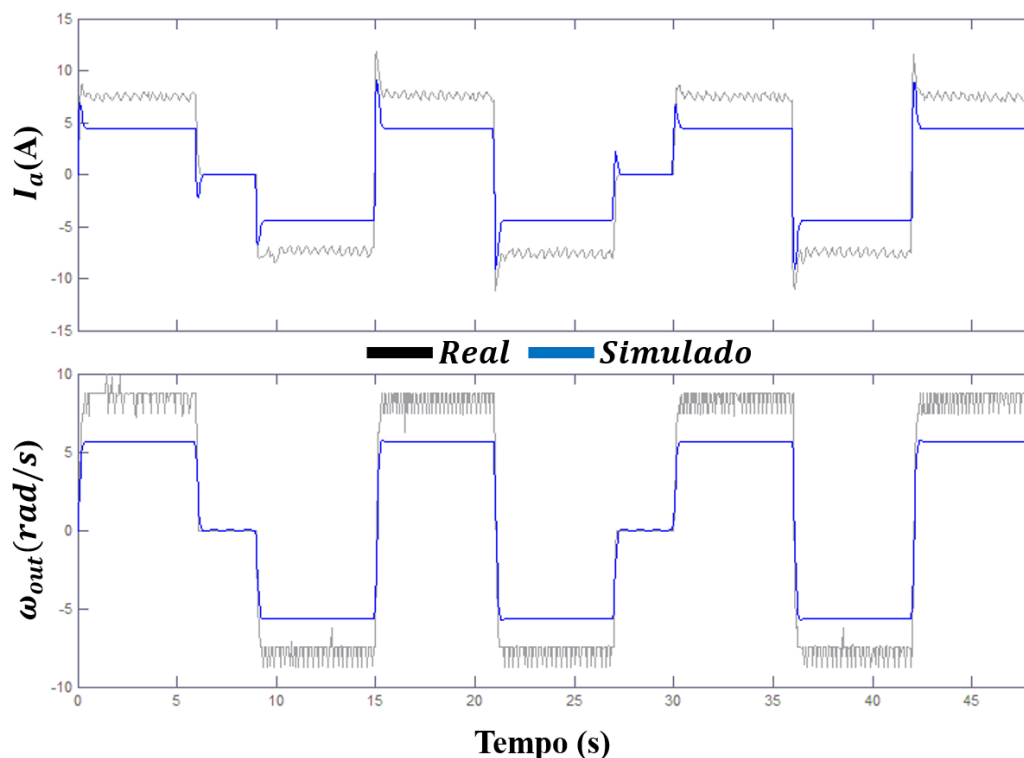


Figura 87 - Velocidade angular e corrente consumida do conjunto motor-redutor simulado (azul), com os parâmetros da Tabela 2 e Tabela 3, e do conjunto real (preto) aplicando uma mesma entrada

Para minimizar os erros da simulação foi realizada uma nova estimativa, agora incluindo o fator de redução e a resistência do motor. Na Tabela 4 são apresentados os novos valores estimados e na Figura 88 a comparação entre os dados do motor-redutor real e da simulação. Como pode ser observado, a partir do conjunto de parâmetros apresentados na Tabela 4 e os valores de  $K_t$  e  $K_e$ , apresentados na Tabela 2, foi possível reproduzir, no modelo simulado, o comportamento da rotação e corrente do motor-redutor real sem carga, para um mesmo conjunto de valores de tensão de entrada, exceto pelo erro de quantização presente na velocidade de rotação dos motores devido à baixa resolução dos *encoders*, em relação a velocidade aplicada para o teste. Com isso, conclui-se a etapa de obtenção dos parâmetros do conjunto motor-redutor.

Tabela 4 - Segundo conjunto de parâmetros estimados

Variável	Valor Original	Valor Estimado	Faixa para otimização
$B_{eq}(\text{Nm}^2/\text{s})$	0,000447	0,0073791	0 - 1
$J_{eq}(\text{kgm}^2)$	0,001	0,0005801	0 - 1
$L_a(\text{H})$	0,01	0,053542	0 - 1

$n_2/n_1$	10	9,0002	9 - 11
$Ra(\Omega)$	0,82445	0,34037	0,3 – 1,3

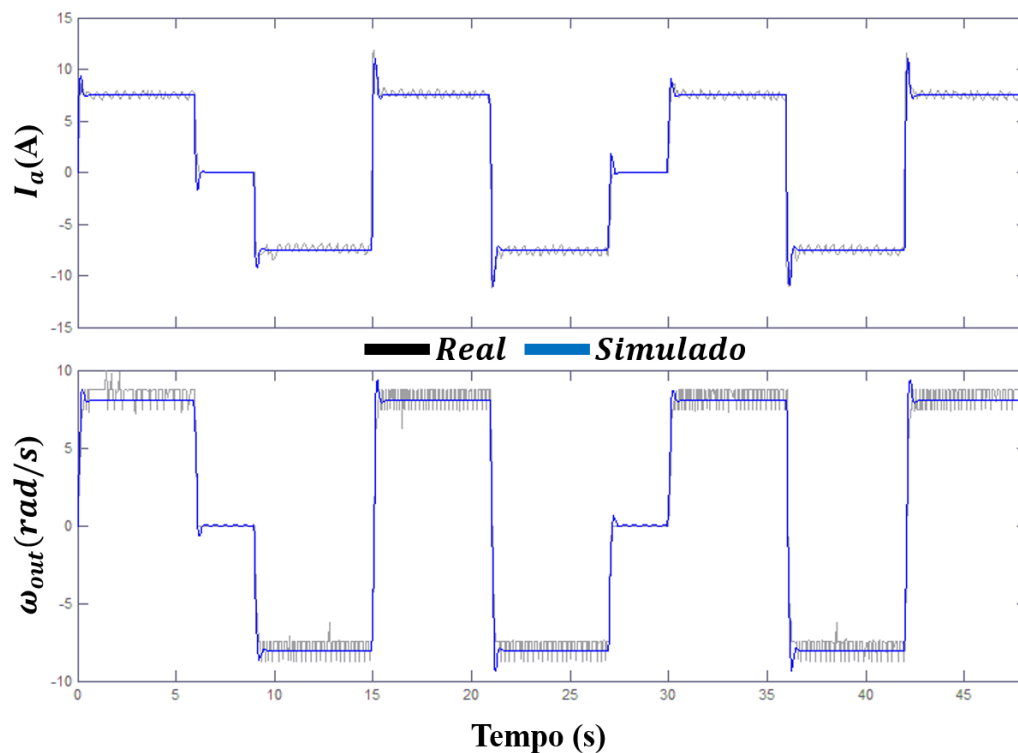


Figura 88 - Velocidade angular e corrente consumida do conjunto motor-reductor simulado (azul) com os parâmetros  $K_t$  e  $K_e$  da Tabela 2 e demais parâmetros da Tabela 4 e do conjunto real (preto) aplicando uma mesma entrada

A seguir, na subseção 4.2.1.2, será apresentado o driver de potência utilizado para controlar a tensão de entrada fornecida para os motores.

#### 4.2.1.2 Driver de potência

Para regular a potência de acionamento dos motores foi utilizado o driver de potência Pololu Dual VNH5019 ash02a (Pololu, 2011), Figura 89.

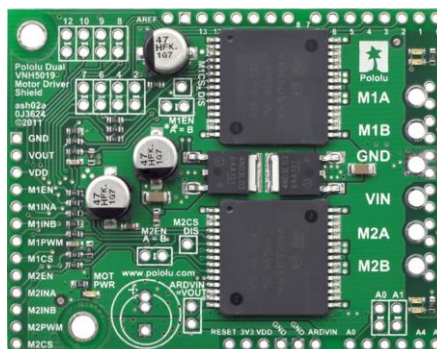


Figura 89 - Driver de potência Pololu Dual VN5019 (Pololu, 2011)

Esse driver para motores CC possui as seguintes características:

- Dois canais para controle independente de motores.
- Tensão de operação de 5,5 V a 24 V, com circuito de proteção de sobre tensão de 27V a 41V. Portanto, o driver consegue operar um pouco acima de sua tensão nominal máxima, ou seja, 24V.
- Corrente contínua máxima 12 A e picos de até 30 A, por canal.
- Acionamento por PWM de até 20 kHz
- Indicação da corrente fornecida para cada motor. Fornece um valor de tensão proporcional à corrente consumida pelo motor (140mA/V).
- Proteção para curto, superaquecimento e sobre tensão.

Uma observação a ser feita em relação ao driver de potência escolhido é que a corrente máxima de operação em regime permanente está um pouco abaixo da corrente nominal do motor. Essa característica não é um problema para o modelo proposto, pois a operação do motor é feita abaixo de seus valores nominais. Os pequenos intervalos em que a corrente exigida pelo motor supera o seu valor nominal estão dentro das condições de pico de corrente do driver de potência, ou seja, 30 A.

Na Figura 90 está representado o esquema de ligação do driver de potência juntamente com suas entradas e saídas, onde MXINA e MXINB são as entradas para a comutação da direção de condução da ponte H do motor X, MXPWM é a entrada PWM para o motor X, MXCS é a saída do sensor de corrente do motor X, MXEN/DIAG é a entrada que habilita a saída de tensão para o motor X, MXA e MXB são as saídas para alimentar o motor X e GND e VIN as entradas para a fonte de alimentação dos motores.

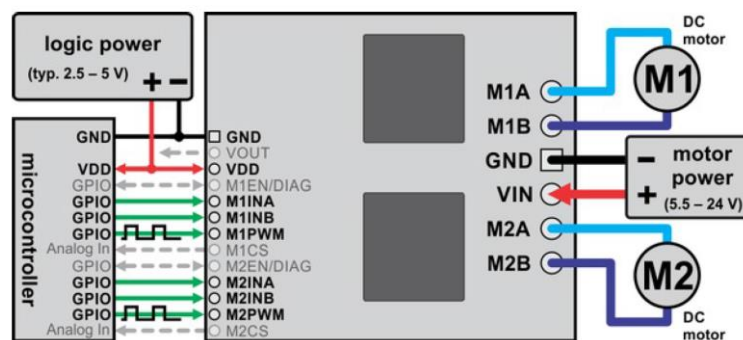


Figura 90 - Esquema de conexões do driver de potência Pololu Dual VNH5019 (Pololu, 2011)

Na próxima subseção, serão especificadas as baterias utilizadas para fornecer potência para os motores e o restante dos dispositivos eletrônicos embarcados no protótipo.

#### 4.2.1.3 Baterias

Para fornecer a tensão e correntes necessárias para o acionamento dos motores, foram utilizadas baterias de Polímero de Lítio (LiPo). Esse tipo de tecnologia de bateria foi escolhido devido sua grande densidade de carga. As baterias de LiPo também têm como vantagem a capacidade de fornecer altos valores de corrente com pequenas variações no seu valor de tensão. Para acionar os motores, que trabalham com 24 V, foram utilizadas duas baterias de LiPo de seis células e 7,7 Ah (Figura 91). Essas baterias têm massa de 948 g cada e foram colocadas em paralelo com o intuito de obter uma maior autonomia. Portanto, a carga máxima final é 15,4 Ah, a tensão nominal é 22,7 V e a tensão máxima é 25,2 V, que está dentro da tolerância do driver de potência.



Figura 91 - Bateria LiPo Fullmax 6s 7,7 Ah

A energia dos demais componentes do projeto, como o hardware controlador e os sensores, é fornecida por duas baterias de LiPo de 3 células e carga de 2,2 Ah (Figura 92). Essas baterias possuem 176 g cada e, assim como as dos motores, foram posicionadas em paralelo. Portanto, ao final tem-se uma carga máxima de 4,4 Ah, uma tensão nominal de 11,1 V e uma tensão máxima de 12,6 V.



Figura 92 - Bateria LiPo Fullmax 3s 2,2 Ah

Com esses conjuntos de baterias, estima-se uma autonomia de operação dos motores em torno de 1 h e do restante do circuito de cerca de 4 h. A autonomia dos motores é relativamente baixa, pois os motores não são ideais e evitou-se o acréscimo de mais baterias com o intuito de redução da massa final do dispositivo.

A seguir serão apresentadas as características dos sensores utilizados no projeto, suas configurações e procedimentos aplicados para tratamento de seus dados.

#### **4.2.2 Sensores**

Nessa seção serão apresentados os modelos dos sensores e as configurações aplicadas na composição desse projeto. Os sensores foram escolhidos com base na fundamentação teórica apresentada no capítulo 2, assim como os procedimentos aplicados para tratamentos de seus dados.

##### **4.2.2.1 Sensor de ultrassom**

Conforme mencionado no capítulo 3, a detecção de obstáculos foi feita por meio de sensores do tipo ultrassom. O modelo de sensor escolhido para o projeto

foi o LV-MaxSonar EZ1 (MaxBotix Inc., 2015), Figura 93. Esse sensor foi escolhido por possuir excelente desempenho e suas características de medição são: faixa de operação de 0 a 6,45 m, sendo que distâncias menores que 15,24 cm são identificadas como 15,24 cm, resolução linear do sensor de 2,54 cm e frequência de leitura de 20 Hz.

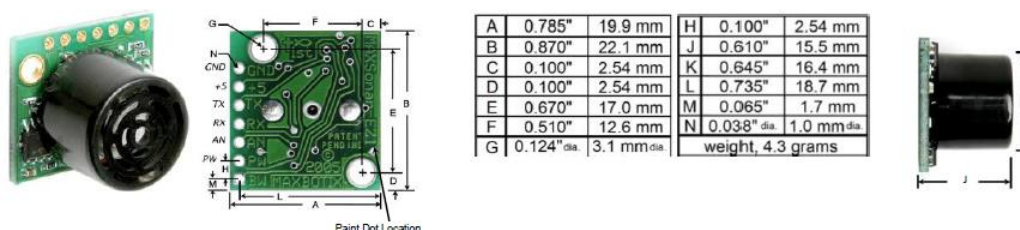


Figura 93 - Sensor de ultrassom LV-MaxSonar EZ1 (MaxBotix Inc., 2015)

Além das características anteriormente apresentadas, o fabricante também disponibiliza a representação do campo de atuação do sensor para diferentes corpos de prova (Figura 94). Na Figura 94, cada quadrado da grade possui largura de 30 cm. A linha contínua representa a detecção do sensor quando alimentado a uma tensão de 5 V e a linha pontilhada quando alimentado a 3,3 V. Nesse trabalho, o valor de tensão utilizado para alimentar os sensores foi 5 V. Como a Figura 94 e as demais figuras dessa subseção foram retiradas do manual do fabricante, (MaxBotix Inc., 2015), manteve-se o idioma original.

# MB1010

## LV-MaxSonar®-EZ1™ Beam Pattern

Sample results for measured beam pattern are shown on a 30-cm grid. The detection pattern is shown for dowels of varying diameters that are placed in front of the sensor

**A** 6.1-mm (0.25-inch) diameter dowel

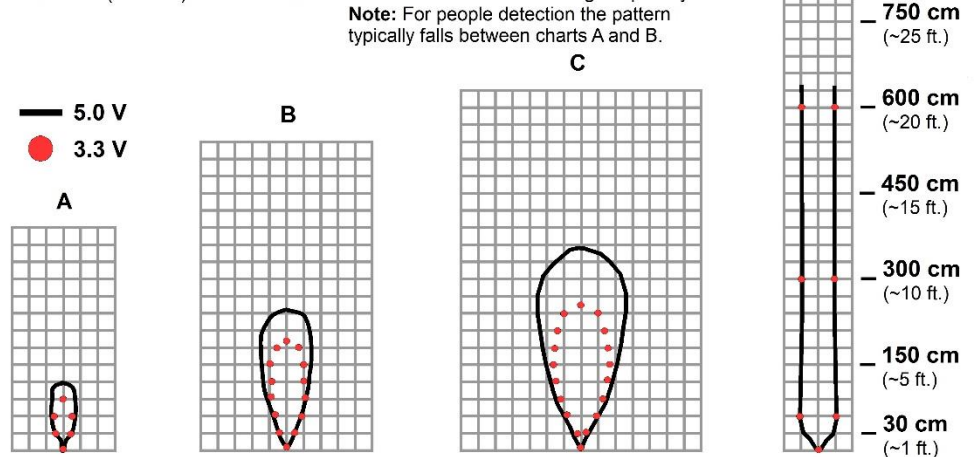
**B** 2.54-cm (1-inch) diameter dowel

**C** 8.89-cm (3.5-inch) diameter dowel

**D** 11-inch wide board moved left to right with the board parallel to the front sensor face.

This shows the sensor's range capability.

**Note:** For people detection the pattern typically falls between charts A and B.



**Beam Characteristics are Approximate**

Beam Pattern drawn to a 1:95 scale for easy comparison to our other products.

MaxBotix® Inc.

For more information or latest product datasheets visit: [www.maxbotix.com](http://www.maxbotix.com)

The names MaxBotix, MaxSonar, EZ0, EZ1, EZ2, EZ3, EZ4, AE0, AE1, AE2, AE3, AE4 and WR1 are trademarks of MaxBotix Inc.

Figura 94 - Campo de atuação LV-MaxSonar-EZ1 (MaxBotix Inc., 2015)

Em relação à leitura dos dados, o sensor oferece três tipos de saída e dois modos de acionamento. As interfaces de saída são: RS232, analógica e largura de pulso, sendo a última a utilizada nesse trabalho, devido a ter maior exatidão que a saída analógica e por utilizar menos portas que a saída RS232. Já os modos de acionamento são: leitura contínua e leitura a partir de um comando de início. No modo de leitura a partir de um comando de início, é possível realizar arranjos de múltiplos sensores com disparos intercalados, para evitar interferência devido efeito de *cross-talk*. Nessa arquitetura de montagem, ao concluir sua leitura, o sensor envia um comando de início de leitura àquele sensor cujo RX está conectado ao seu TX (Figura 95).

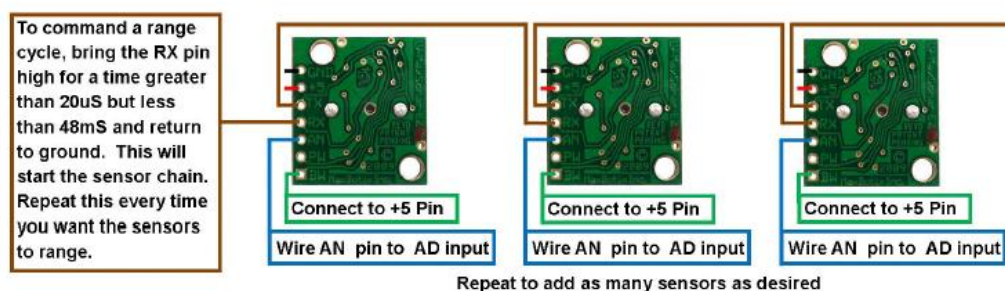


Figura 95 - Configuração de acionamento sequencial (MaxBotix Inc., 2015)



Ao todo, foram utilizados 8 sensores de distância, cinco deles posicionados conforme o modelo simulado apresentado no capítulo 3, e os outros três restantes foram posicionados para captar buracos e degraus nas laterais e a frente do protótipo. O posicionamento final de cada sensor no protótipo real pode ser observado na Figura 96, onde os sensores S1 a S8 são: *Sensor Lateral Esquerda* (S1), *Sensor Frente Esquerda* (S2), *Sensor Frente* (S3), *Sensor Frente Direita* (S4), *Sensor Lateral Direita* (S5), *Sensor Altura Esquerda* (S6), *Sensor Altura Frente* (S7) e *Sensor Altura Direita* (S8).

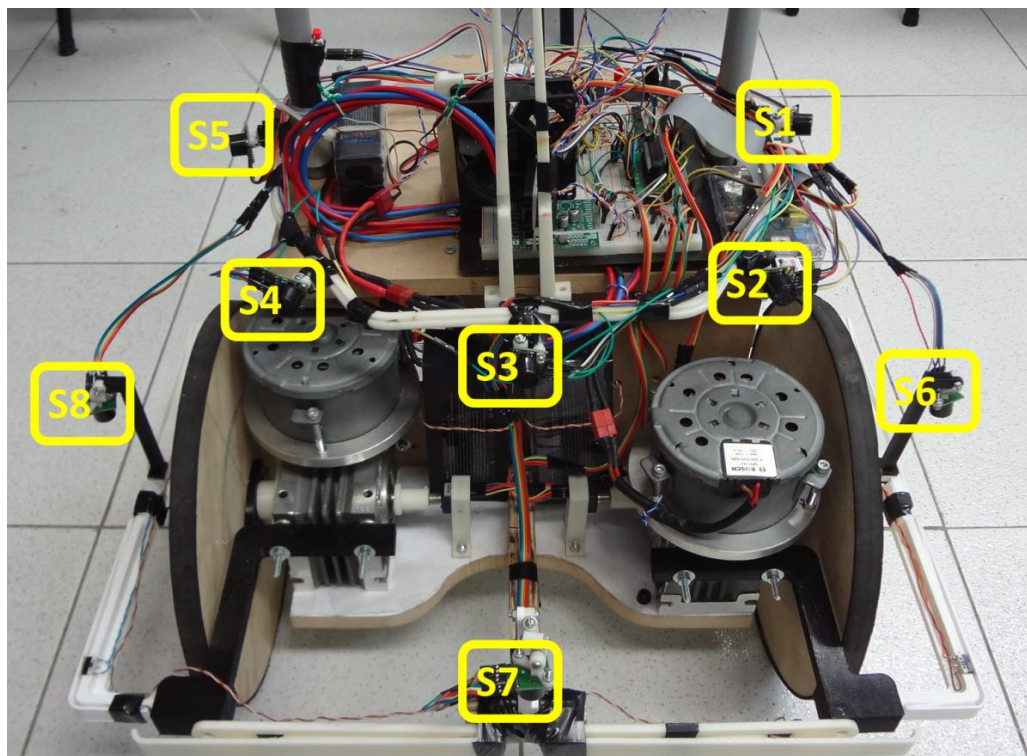


Figura 96 - Distribuição dos sensores de ultrassom no dispositivo robótico

O acionamento desses 8 sensores foi feito de maneira intercalada, para evitar interferências devido a *cross-talk*. Na sequência de acionamento adotada, os sensores S2 e S8 são acionados simultaneamente. Após concluir a leitura, o sensor S2 dispara um comando para o início da leitura dos sensores S4 e S6. Ao concluir a leitura, o sensor S4 envia um comando de leitura para os sensores S1, S5 e S7. Com a conclusão da leitura do sensor S7, o mesmo envia um comando de leitura para o sensor S3, finalizando um ciclo de leitura. Essa sequência de acionamento



foi adotada para que sensores que apontam na mesma direção não realizem leituras simultâneas, evitando assim a ocorrência do efeito do *cross-talk*.

Outro procedimento adotado para melhorar a exatidão da medida de distância obtida pelos sensores foi aplicar um filtro a cada medição realizada. O filtro adotado exerce um efeito semelhante a uma média móvel, contudo seu acionamento ocorre somente quando o valor medido por um dado sensor tiver uma variação de 50 % em relação a medida anterior. Nas equações (61) e (62) é apresentada a implementação do filtro.

Se  $\Delta_{Distância}$

$$> 50 \% \begin{cases} Distância_{computada} = \left( \frac{Distância_{lida} + 2 * Distância_{velha}}{3} \right) \\ Distância_{velha} = \left( \frac{Distância_{lida} + Distância_{velha}}{2} \right) \end{cases} \quad (61)$$

$$Se \text{ não } \begin{cases} Distância_{computada} = Distância_{lida} \\ Distância_{velha} = Distância_{lida} \end{cases} \quad (62)$$

#### 4.2.2.2 Encoder Rotativo

O *encoder* rotativo utilizado foi o ENA1J-B28-L00100L (Bourns, 2015), ilustrado na Figura 97. Ele é um modelo óptico incremental e foi aplicado para o retorno do controle de velocidade das rodas.

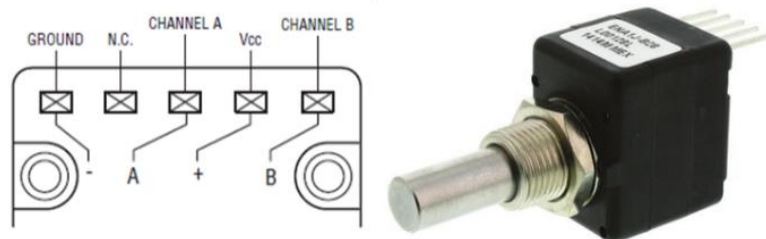


Figura 97 - *Encoder* rotativo óptico o ENA1J-B28-L00100L (Bourns, 2015)

O *encoder* possui um PPR de 100, ou seja, resolução de  $3,6^\circ$ , opera com velocidade máxima de 300 RPM, suporta impactos de até 50 vezes a aceleração da

gravidade, vibrações de até 5 vezes a gravidade e temperaturas de operação de  $-40^{\circ}\text{C}$  a  $70^{\circ}\text{C}$ .

As saídas do *encoder* são duas ondas quadradas defasadas em  $90^{\circ}$ , para indicar o sentido de rotação (Figura 98), e não requerem circuito *debounce*. A desvantagem que esse modelo apresenta para aplicação desse projeto é o baixo PPR, pois em aplicações de baixa velocidade a resolução de  $3,6^{\circ}$  não permite grande exatidão do controle de velocidade.

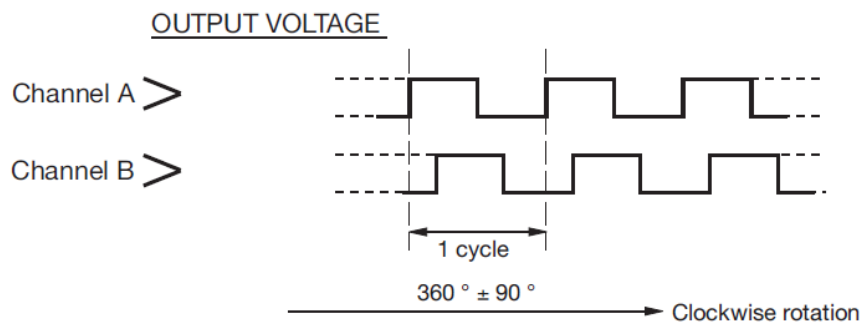


Figura 98 - Forma de onda das saídas do *encoder* ENA1J-B28-L00100L (Bourns, 2015)

Os *encoders* foram posicionados junto às saídas dos motores e sua fixação foi feita por meio de peças de plástico ABS impressas por uma impressora 3D. Na Figura 99 é possível visualizar os *encoders* conectados aos motores.

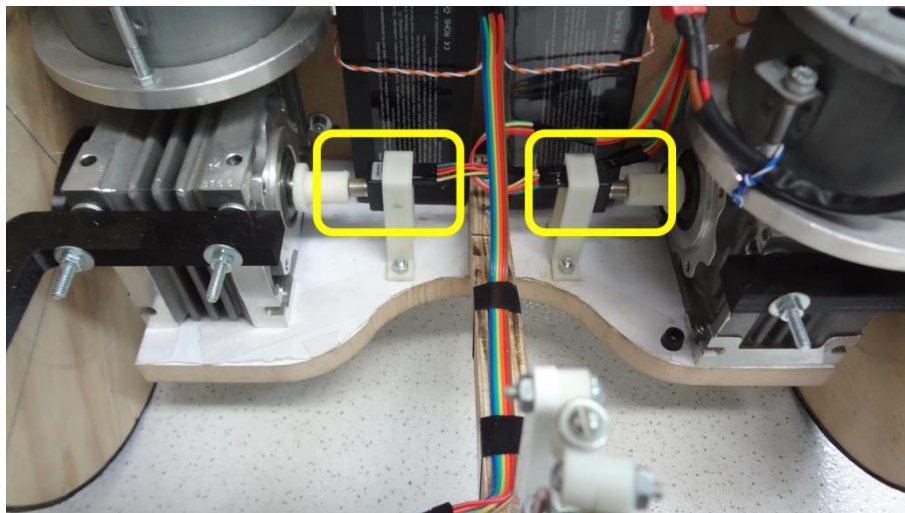


Figura 99 - *Encoders* conectados aos motores

### 4.2.2.3 Central Inercial

Para identificar a orientação do dispositivo foi utilizada uma central inercial de baixo custo. A central inercial escolhida foi a Pololu MinIMU-9 v2 (Pololu, 2014), apresentada na Figura 100.

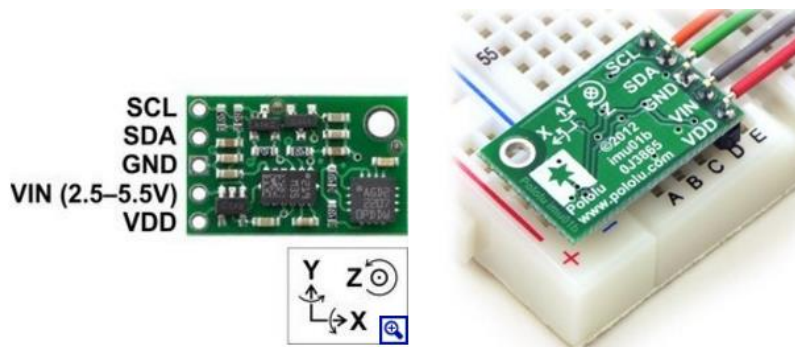


Figura 100 - IMU Pololu MinIMU-9 v2 (Pololu, 2014)

A MinIMU-9 é uma central inercial composta por um giroscópio, um acelerômetro e um magnetômetro, sendo que cada um dos sensores possui três eixos independentes de medição, ou seja, a central inercial possui nove graus de liberdade. O protocolo de comunicação utilizado para acesso aos dados e envio de comandos é o I2C (NXP Semiconductors, 2014) e as demais características de operação são:

- Tensão de operação: 2,5 V a 5,5 V
- Corrente de operação: 10 mA
- Giroscópio: leitura de 16 bits por eixo
- Acelerômetro: leitura de 12 bits por eixo
- Magnetômetro: leitura de 12 bits por eixo
- Faixa de sensibilidade (configurável):
  - Giroscópio:  $\pm 245$ ,  $\pm 500$ ,  $\pm 2000$  °/s
  - Acelerômetro:  $\pm 2$ ,  $\pm 4$ ,  $\pm 6$ ,  $\pm 8$  ou  $\pm 16$  g
  - Magnetômetro:  $\pm 1,3$ ,  $\pm 1,9$ ,  $\pm 2,5$ ,  $\pm 4,7$ ,  $\pm 5,6$  ou  $\pm 8,1$  gauss

Esse modelo foi escolhido devido à relação custo-benefício, disponibilidade para compra e, principalmente, por possuir uma biblioteca de código aberto para

calibração e fusão de dados. A biblioteca utilizada foi a RTIMULib (Richards-Tech, 2014a).

A RTIMULib é uma biblioteca escrita em C++ para tratamento e aquisição de dados de diversos modelos de central inercial, além do modelo utilizado nesse trabalho. Dentre as funcionalidades que a biblioteca apresenta, destacam-se:

- *Representação de rotação por quatérnion* (Kuipers, 2000): a notação por quatérnion tem como vantagem, em relação à notação por ângulos de Euler, de permitir representar uma rotação independente da escolha do sistema de coordenadas, não sofrendo o efeito de *gimbal lock* (Biasi e Gattass, 2002).
- *Calibração dos sensores*: ela apresenta uma interface gráfica que auxilia na calibragem dos erros de *offset* do acelerômetro, giroscópio e magnetômetro e corrige os erros de *tilt*, *soft-iron* e *hard-iron* do magnetômetro.
- *Fusão de dados*: a biblioteca possui um filtro de Kalman para fusão dos dados dos sensores da IMU, com o intuito de obter uma melhor estimativa dos ângulos de *roll*, *pitch* e *yaw*.

O posicionamento do sensor (Figura 101) foi realizado de modo que seu eixo Z ficasse alinhado com o eixo de rotação do robô e que o sensor não ficasse tão próximo dos motores, que são uma fonte de ruído magnético.

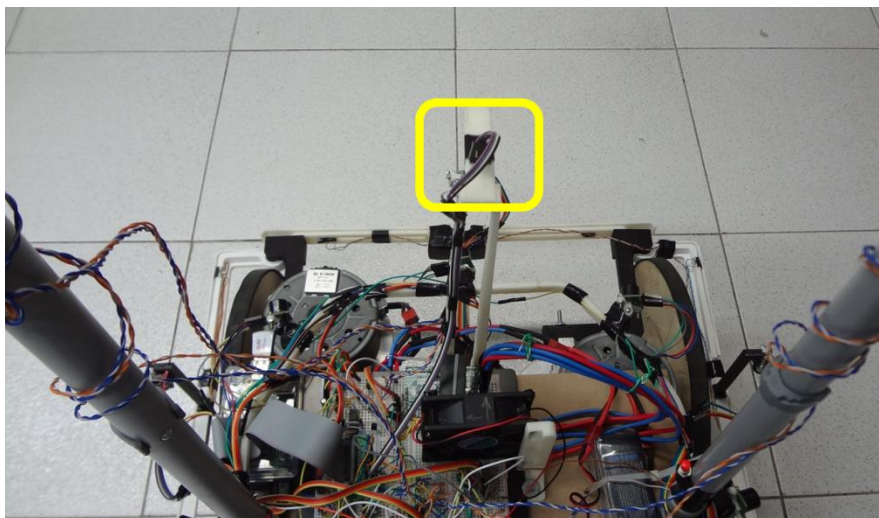


Figura 101 - Posição da IMU no protótipo

Para realizar a calibração dos dados da IMU, a biblioteca RTIMULib oferece dois aplicativos: o RTIMULibDemo e o RTIMULibCal. Ambos realizam as mesmas etapas de calibração, contudo o primeiro apresenta uma interface gráfica para operação do usuário. Os procedimentos e etapas de calibração utilizando RTIMULibDemo são detalhadamente descritos em (Richards-Tech, 2014b) e em resumo são: cálculo do erro de *bias* do giroscópio, cálculo do erro de *bias* do acelerômetro e correção dos erros de *soft* e *hard-iron* do magnetômetro.

O erro de *bias* do giroscópio é calculado assim que o aplicativo de calibração é iniciado. Ao abrir o aplicativo, com a IMU estática, são armazenadas amostras de leituras dos três eixos do giroscópio. A média aritmética desses valores será a correção do erro de *bias* para cada eixo do giroscópio.

Para obter o erro de *bias* do acelerômetro são realizadas medições da aceleração da gravidade alinhada com cada um dos eixos do acelerômetro. Portanto, obtém-se dois valores de leituras para cada eixo,  $\pm g$ , e a média desses valores será a correção do erro de *bias*. Durante esse procedimento deve-se ter o cuidado de movimentar o acelerômetro lentamente, de modo que a aceleração obtida por ele seja apenas a gravidade.

A correção dos ruídos do magnetômetro é dividida em duas etapas. A primeira etapa é obter os valores máximo e mínimo do campo captado por cada eixo. O procedimento dessa etapa é girar a central inercial com o intuito de obter medidas do campo em todas as direções. Os valores máximo e mínimo do campo obtidos, para cada eixo, são automaticamente armazenados.

Na segunda etapa da calibração são obtidos os parâmetros para correção da distorção e *offset* do campo. Para realizar essa correção, o aplicativo utiliza o algoritmo *REllipsoidFit*. Esse algoritmo aproxima um conjunto de pontos à forma de uma esfera, tendo como retorno um vetor  $3 \times 1$ , que representa do centro da esfera, e uma matriz  $3 \times 3$ , que representa a distorção da esfera. Esse algoritmo foi feito baseado no algoritmo *ellipsoidfit.m* (Petrov, 2015).

Para que essa etapa seja realizável, é necessário ter instalado o software Octave (Eaton, 2016), que é um software de código aberto para aplicações numéricas. Esse software é necessário para executar o arquivo *ellipsoidfit.m* que terá como retorno os parâmetros para correção da distorção do campo.

Realizando-se o mínimo de 200 amostras para cada octante da esfera, o programa habilita o cálculo do vetor e da matriz para correção da distorção. Ao

término do processo é possível plotar os dados amostrados antes e depois da calibração ser executada, como comparativo da correção feita. Na Figura 102 é apresentado o gráfico das amostras não corrigidas e das amostras corrigidas e nas equações (63) e (64) a matriz e o vetor de correção.

$$Cor_{Matriz} = \begin{bmatrix} 0.999332 & -0.009251 & -0.001187 \\ -0.009251 & 0.870729 & -0.000269 \\ -0.001187 & -0.000269 & 0.779796 \end{bmatrix} \quad (63)$$

$$Offset_{Vet} = \begin{bmatrix} -0.037945 \\ -1.719840 \\ 6.231551 \end{bmatrix} \quad (64)$$

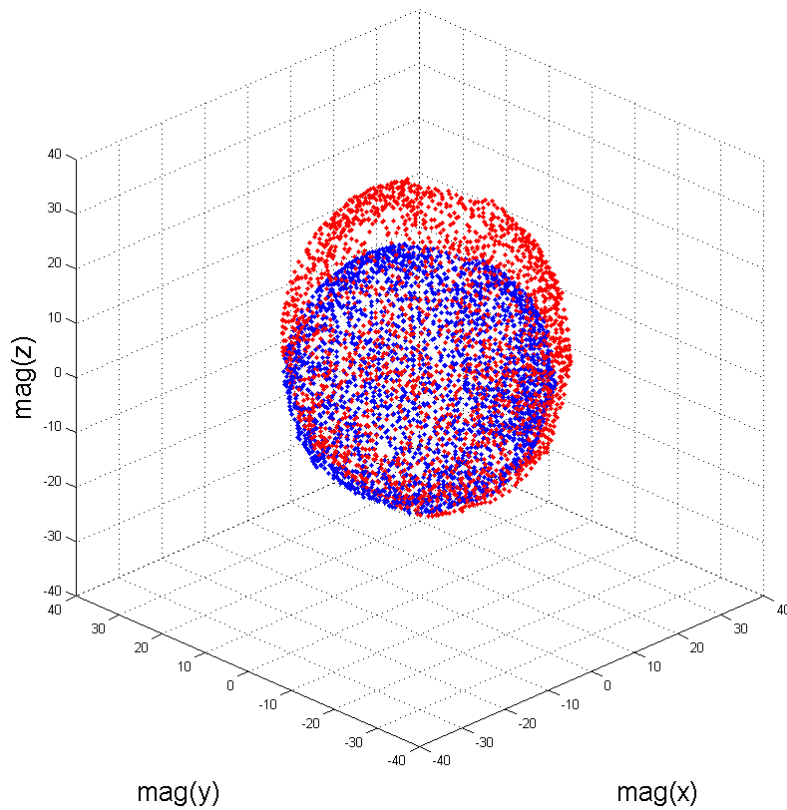


Figura 102 - Dados dos magnetômetros da IMU com correção (azul) e sem correção (vermelho)

#### 4.2.2.4 Sensor de força

Para detectar a atuação do usuário sobre o dispositivo foram utilizados sensores de força FRS 400 (Interlink Electronics, 2015), ilustrado na Figura 103.

Esse sensor tem como característica apresentar um valor de resistência em seus terminais de saída, proporcional à força exercida sobre seu elemento sensor. Esse tipo de sensor foi escolhido devido ao seu baixo custo, principalmente quando comparado a sensores de torque.

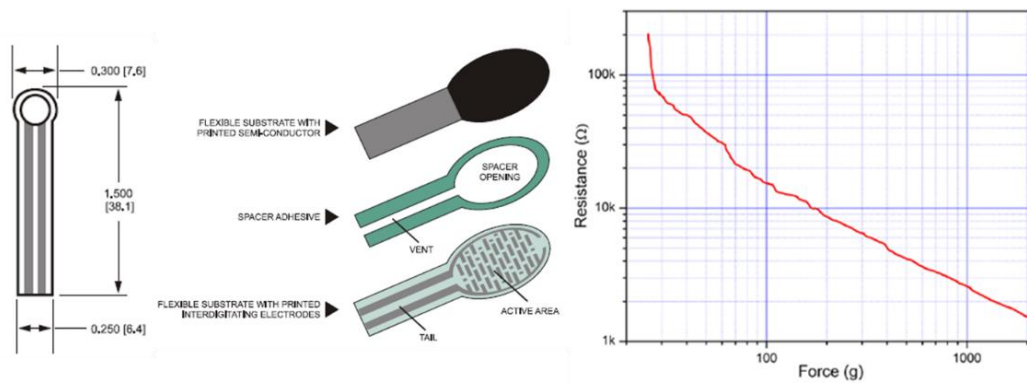


Figura 103 - FSR 400 e curva característica (Interlink Electronics, 2015)

Para identificar a atuação do usuário sobre o dispositivo foram utilizados 6 sensores FSR 400. Esses sensores foram posicionados na pega do dispositivo conforme a Figura 104. Os sensores identificados como *Sensor Pega Esquerda* e *Sensor Pega Direita* são responsáveis por sinalizar que o usuário está segurando o dispositivo com as duas mãos. Caso um dos sensores não capture uma força mínima do usuário, a atuação dos motores é interrompida e bloqueada. Esse valor da força mínima necessária foi obtido de forma empírica, ao registrar a leitura do sensor quando o usuário segura a pega do protótipo com firmeza.

Os sensores identificados como *Sensor Atua Esquerda* e *Sensor Atua Direita* são a interface para atuação direta do usuário sobre os motores. Portanto, caso o usuário exerça uma força acima de um limiar sobre um dos sensores de atuação, será enviado um valor de velocidade proporcional à força percebida acima desse limiar ao motor correspondente. O mesmo ocorre com os sensores *Atua Esquerda Ré* e *Atua Direita Ré*, contudo, sua atuação faz com que os motores girem no sentido oposto à atuação dos sensores *Atua Esquerda* e *Atua Direita*. O valor desse limiar, assim como nos sensores de força da *Pega Esquerda* e *Direita*, foi obtido de maneira empírica e pode gerar uma velocidade máxima de 2,5 *rad/s* para o motor correspondente à atuação.

A atuação do usuário tem prioridade sobre o SIF e, portanto, enquanto o usuário estiver atuando no motor, ou seja, aplicando uma força sobre os sensores



de atuação esquerda ou direita, os valores fornecidos pelo SIF são ignorados. Assim como no caso dos sensores da pega, o limiar que habilita a atuação manual dos sensores *Atua Esquerda* e *Atua Direita* foi obtido de maneira empírica.

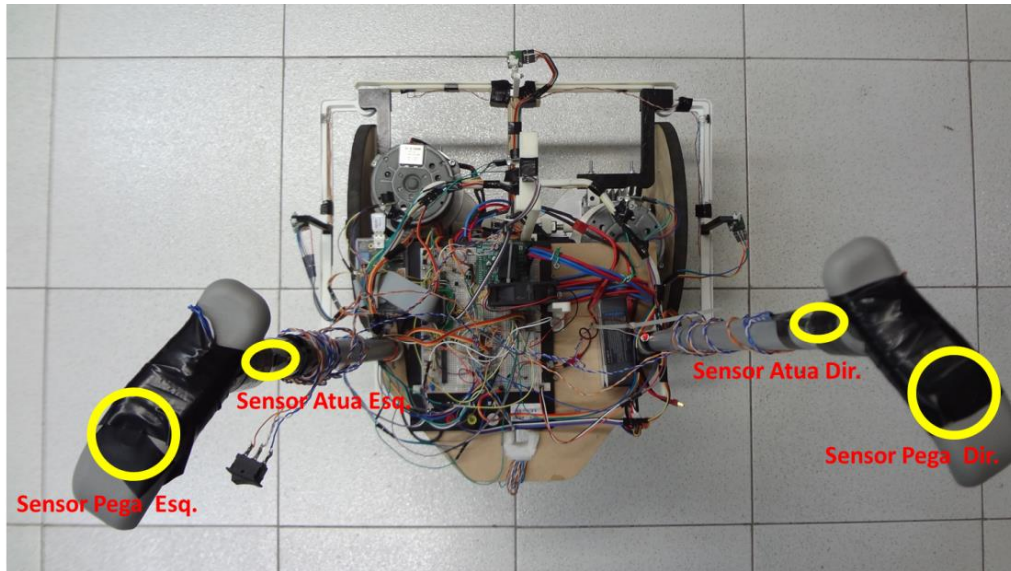


Figura 104 - Distribuição dos sensores de força sobre as pegas esquerda e direita do dispositivo

A seguir, na subseção 4.2.3, será apresentado o hardware processador utilizado e suas interfaces de entrada e saída. Na sequência serão apresentadas na subseção 4.2.4 as interligações dos sensores, apresentados nas subseções anteriores, com o hardware processador.

### 4.2.3 Hardware processador

O hardware processador utilizado foi a plataforma Raspberry Pi, modelo B Rev2 (Upton e Halfacree, 2012), ilustrado na Figura 105. Esse SBC foi escolhido por apresentar melhor relação custo-benefício que os demais listados no capítulo 2, além de possuir uma grande comunidade de desenvolvimento.



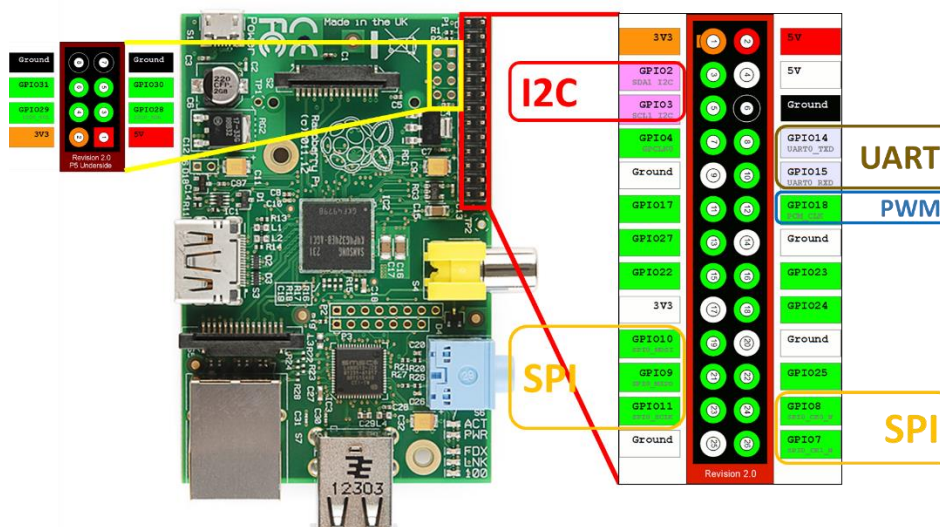


Figura 105 - Raspberry Pi B ver 2 com destaque aos pinos e entrada e saída

Ele possui uma interface de comunicação SPI (Microchip Technology Inc., n.d.) com duas saídas *chip select*, uma interface I2C, uma interface UART, 12 portas de entrada ou saída (GPIO) para uso geral, sendo uma delas configurável como saída PWM. Cada um dos pinos das interfaces listadas anteriormente pode ser configurado como um GPIO de uso geral, caso não se deseje usar a interface padrão. Para acessar as interfaces e portas de entrada e saída do Raspberry Pi, foi utilizada a biblioteca WiringPi (Hendersons, 2015).

A WiringPi é uma biblioteca escrita em C para acesso aos GPIOs do CPU ARM11 BCM2835 (Broadcom Corporation, 2012), CPU do Raspberry PI 1. Por meio dessa biblioteca é possível, por exemplo, configurar o tipo da porta (entrada ou saída), gerenciar os dados das interfaces, tratar interrupções e criar saídas PWM via *threads*. A biblioteca também oferece suporte a alguns periféricos comumente utilizados com o Raspberry Pi, como chips e blocos de expansão de portas.

Para suprir a falta de entradas analógicas, foi utilizado o chip de expansão de entradas analógicas MCP3008 (Microchip Technology Inc., 2008), apresentado na Figura 106. Esse chip possui oito ADCs de 10 bits, tensão de referência de até 5 V e uma interface de comunicação SPI. Ao todo foram utilizados 3 chips MCP3008, dois para a leitura dos sensores de força e corrente e um para uma possível utilização com os sensores de ultrassom.

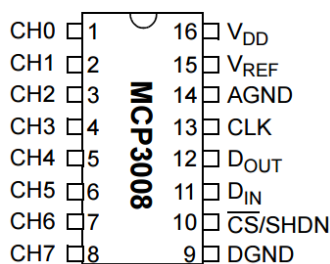


Figura 106 - Ilustração na pinagem do chip MCP3008 (Microchip Technology Inc., 2008)

Além de circuitos integrados para expansão das entradas analógicas, também foi utilizado um chip para expansão das entradas e saídas digitais. Para aumentar o número de portas digitais, foi utilizado o chip MCP23017 (Microchip Technology Inc., 2007), Figura 107. Esse chip possui 16 portas digitais que podem ser configuradas como entrada ou saída, e seu protocolo de comunicação é I2C.

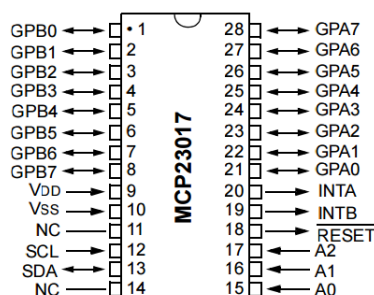


Figura 107 - Ilustração da pinagem do chip MCP23017 (Microchip Technology Inc., 2007)

Por fim, o último periférico adicionado ao hardware processador foi um adaptador USB/Wi-Fi. Esse adaptador foi utilizado para que o robô possa se comunicar por uma rede sem fio ao celular e a qualquer outro dispositivo da rede que esteja executando um programa cliente para o servidor Player.

A seguir, na subseção 4.2.4, será apresentada a maneira como foi realizada a montagem dos sensores e atuadores anteriormente apresentados, para que então, na seção 4.2.5, sejam apresentadas as características dos drivers que irão adaptar todo o hardware com as interfaces do servidor Player.

#### 4.2.4 Montagem

O primeiro diagrama a ser apresentado é do regulador de tensão que foi montado para reduzir a tensão fornecida pelas baterias de LiPo de 11,1 V para a

tensão do Raspberry Pi, ou seja, 5 V. Esse circuito utiliza o chip regulador de tensão LM7805 em uma das aplicações apresentadas em seu datasheet (Fairchild Semiconductor, 2014). Essa implementação, apresentada na Figura 108, permite fornecer correntes acima de 3 A, com queda de menos de 5% no valor nominal de tensão, e possui proteção contra curto circuito, devido a fonte de corrente implementada com os transistores de potência TIP42.

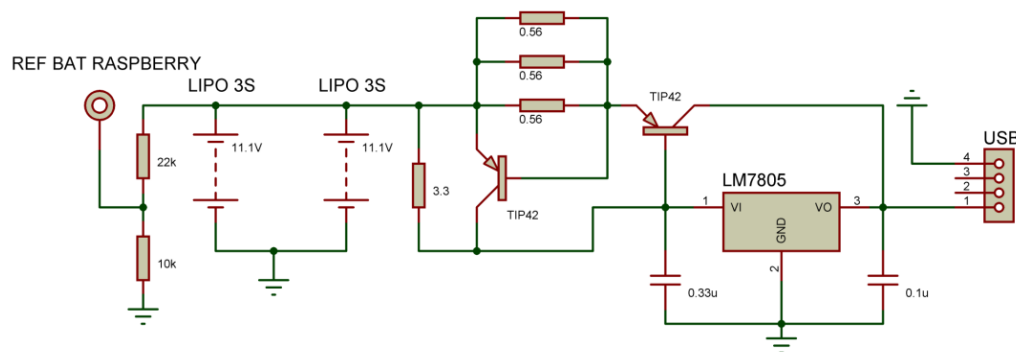


Figura 108 - Diagrama circuito regulador de tensão

O próximo diagrama a ser apresentado (Figura 109) é o diagrama referente à atuação e controle de velocidade dos motores. No diagrama é possível observar, além das portas utilizadas para o controle do driver de potência e leitura dos *encoders*, os chips de expansão utilizados para as interfaces AIO e DIO do servidor Player (Gerkey et al., 2010), que serão descritas nas subseções 4.2.5.1 e 4.2.5.2, e o botão de parada de emergência. Outro ponto a ser destacado são os diodos de sinal, modelo 1N4148 (NXP Semiconductors, 2004), que foram utilizados para reduzir o nível de tensão das saídas dos *encoders*. Essa redução foi necessária porque o nível lógico alto do Raspberry Pi é 3.3 V, e o *encoder* opera a uma tensão de 5 V. As entradas do Raspberry Pi que foram utilizadas para a leitura dos *encoder*, foram colocadas, internamente, na condição de PULL DOWN, ou seja, ligadas ao terra por um resistor.

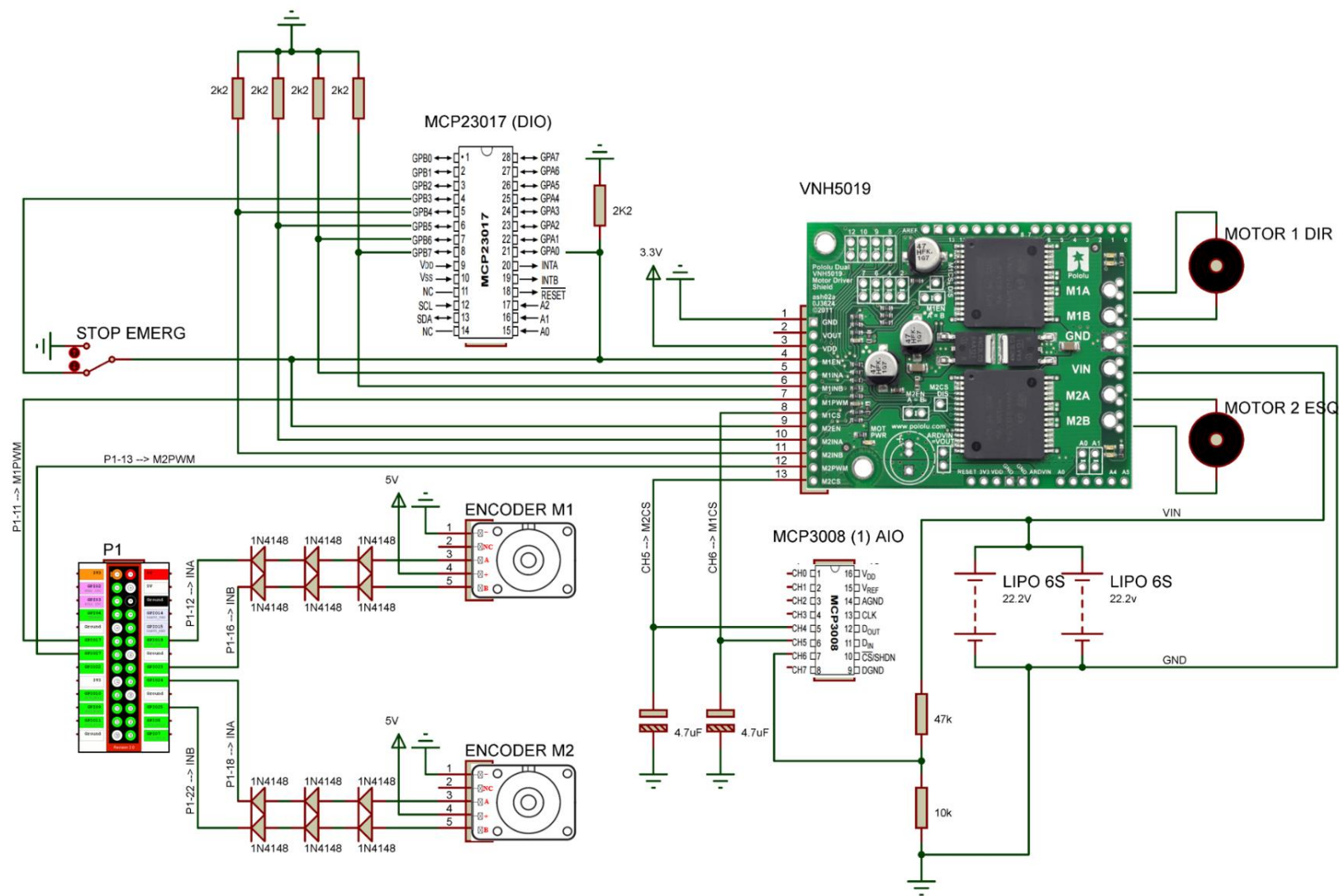


Figura 109 - Diagrama do circuito de atuação e controle dos motores

O terceiro diagrama, apresentado na Figura 110, refere-se aos sensores de distância utilizados para detecção de obstáculos e desníveis. Nesse diagrama é possível observar os oito sensores de distância, S1 a S8 que, conforme apresentado no capítulo anterior, foram utilizados para detectar a distância ao obstáculo, S1 a S5, e os desníveis do chão, S6 a S8. Também é possível observar a sequência de acionamento dos sensores, descrita na subseção 4.2.2.1, onde a indicação START mostra onde ocorre o primeiro comando de leitura e END o último. Outra observação a ser feita é que foi inserido um divisor de tensão nas saídas de largura de pulso dos sensores com o intuito de adequar a tensão de saída, 5 V, à tensão de entrada, 3,3 V, do Raspberry Pi.

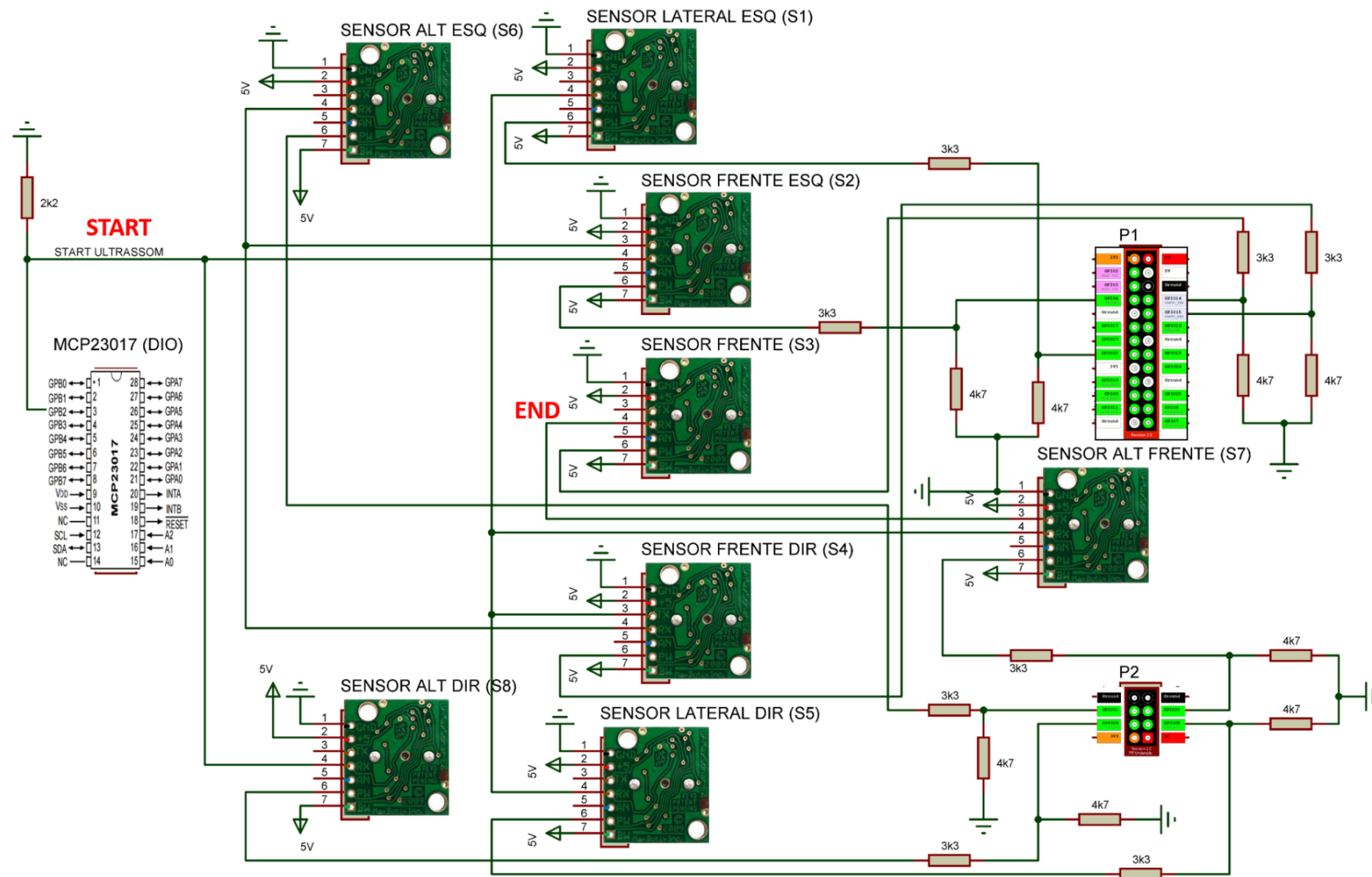


Figura 110 - Diagrama de interligação dos sensores de ultrassom

O quarto diagrama (Figura 111) refere-se aos sensores de força, instalados nas pegas das bengalas esquerda e direita, e as portas analógicas utilizadas na interface AIO do servidor Player. Os sensores de força, conforme descrito na subseção 4.2.2.4, são usados para identificar se o usuário está segurando o dispositivo e sua intenção de locomoção. Outra observação a ser feita é que foi inserido um circuito lógico, destacado em azul, para aumentar a possibilidade de inclusão de novos sensores.

Nesse diagrama também é possível observar que, além dos sensores de força, os AIO do dispositivo também estão destinados aos sensores de corrente dos motores e para medir os níveis das baterias do motor e do hardware processador.

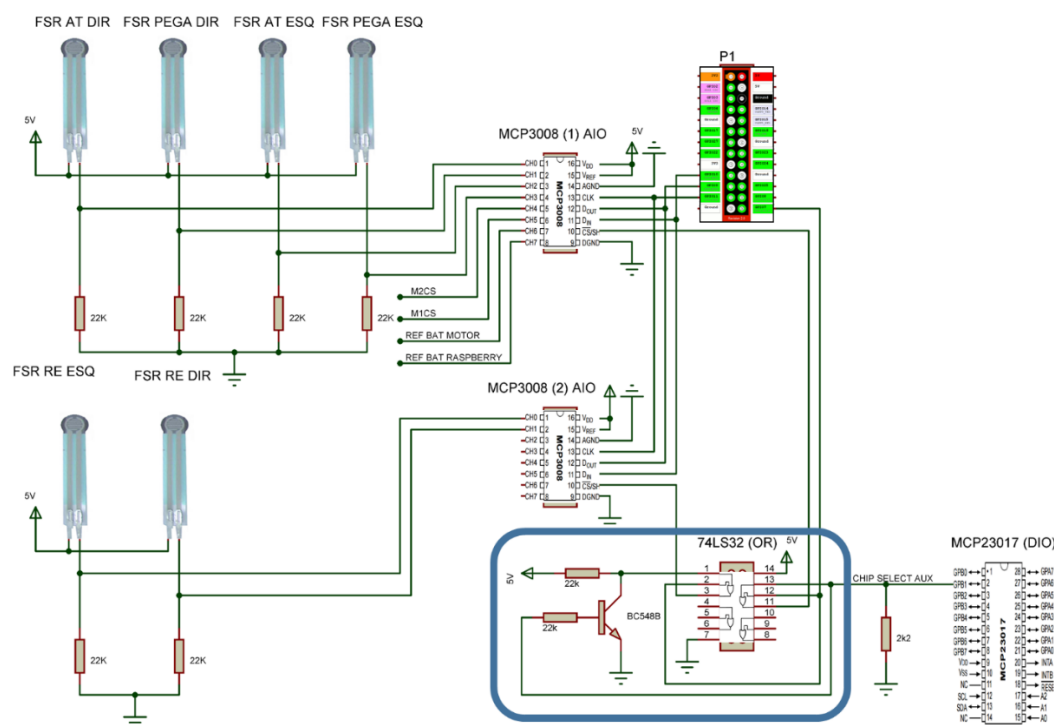


Figura 111 - Diagrama da interface AIO e dos sensores de força

O quinto e último diagrama (Figura 112) apresenta as conexões da central inercial (MINIMU-9 V2) e do chip de expansão de entradas e saídas digitais (MCP23017), ou seja, dos periféricos utilizados nas interfaces IMU e DIO do servidor Player (Gerkey et al., 2010). Em relação às conexões do chip de expansão, é importante destacar a conexão da porta *GPA7*, que representa a saída 0 da interface DIO, com a porta *GPB0*, que representa a entrada 7 da interface DIO, e o LED. Essas conexões foram utilizadas para indicar ao usuário, por meio da



sinalização do LED ligado, e a outros programas, por meio do nível lógico alto da porta *GPB0*, o funcionamento do programa cliente. Conforme será apresentado na seção 4.2.5, enquanto o programa cliente estiver em execução, o mesmo mantém a porta *GPA7*, ou seja, a saída 0 da interface DIO, em nível lógico alto. Além dessa sinalização, esse diagrama também apresenta a conexão do botão de *RESET*, utilizado pelo usuário para reiniciar o programa cliente e o servidor Player, e as chaves de contato normalmente fechadas (NF) que foram utilizadas para detectar a ocorrência de colisões do dispositivo, identificadas no diagrama como *BUMPER*. O posicionamento das chaves de contato no protótipo pode ser observado na Figura 113.

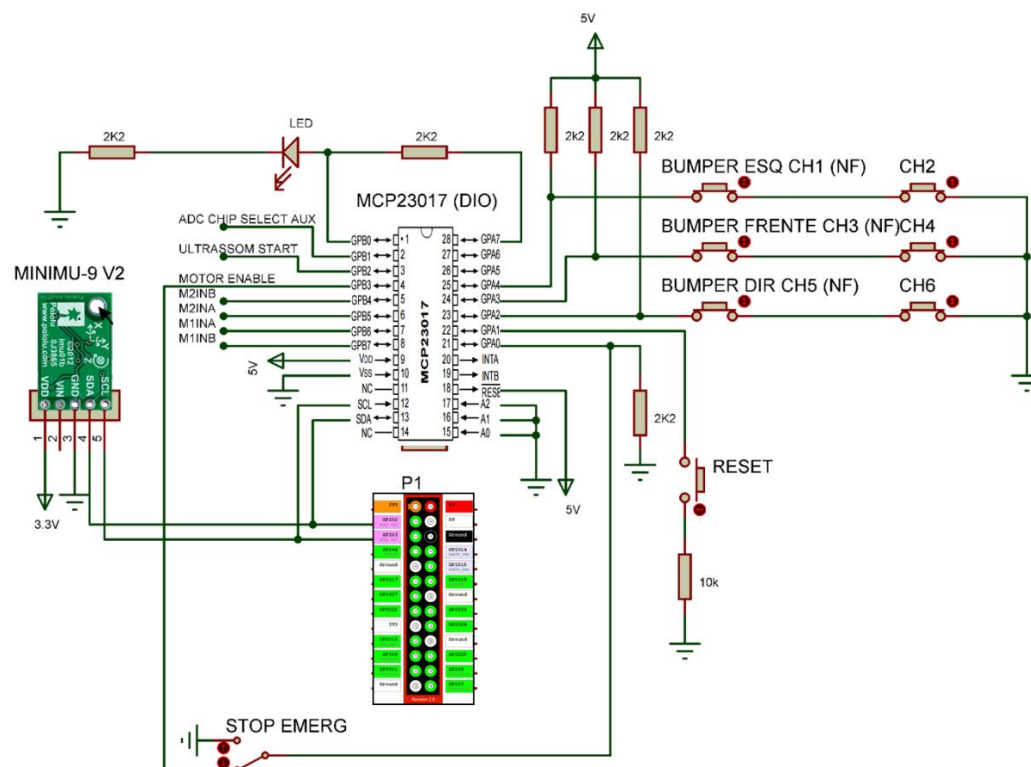


Figura 112 - Diagrama da interface IMU, DIO e dos sensores de contato



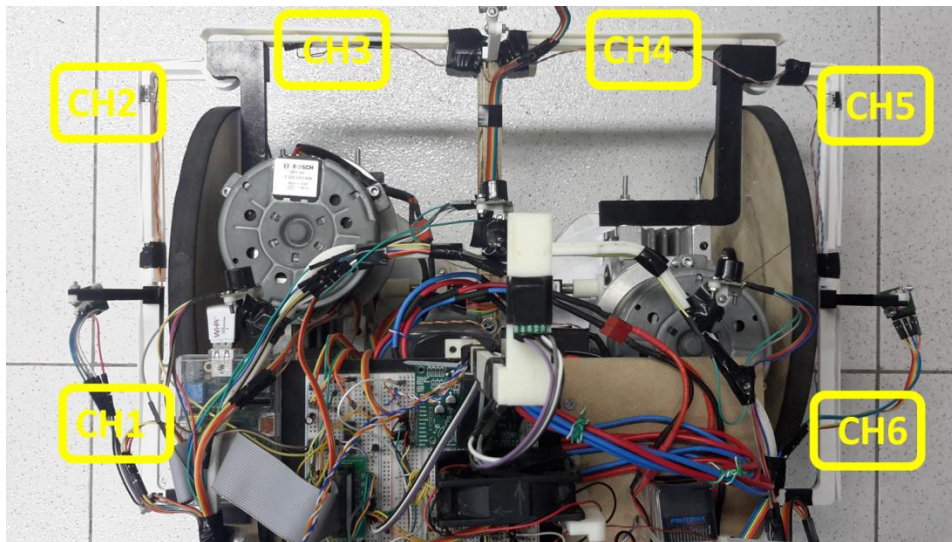


Figura 113 - Posicionamento das chaves do contato NF para identificar colisões

Com a conclusão da apresentação dos diagramas das interligações dos sensores, atuadores e hardware processador, na próxima subseção serão apresentados os drivers criados para que o servidor Player gerencie cada um dos hardwares apresentados.

#### 4.2.5 Player

Para utilizar o servidor Player para gerenciar os sensores e atuadores da plataforma real, foi necessário programar drivers específicos para cada tipo de sensor e atuador utilizado. A estrutura genérica de um driver para o servidor Player pode ser observada no fluxograma exposto na Figura 114.

A rotina de um driver começa com a criação de uma *thread* no sistema operacional, dentro da qual serão executadas e chamadas suas demais rotinas. Com a *thread* de execução do driver criada, são carregados os parâmetros armazenados no arquivo de configuração. Após carregar esses parâmetros, o driver aguarda a conexão externa de um cliente para continuar sua rotina de execução. Com uma conexão com um programa cliente estabelecida, é iniciada uma rotina de *setup*, onde são estabelecidas todas as condições iniciais que o driver deve executar. Após concluir o *setup*, o driver verifica se ainda há algum usuário conectado e, caso positivo, executa as mensagens enviadas pelo o usuário e, em seguida, publica os seus dados. Caso não haja nenhum usuário conectado, é executada a rotina de

encerramento. Nessa rotina, o driver finaliza todos os processos que foram iniciados em sua rotina de *setup* e fica aguardando a conexão de um usuário.

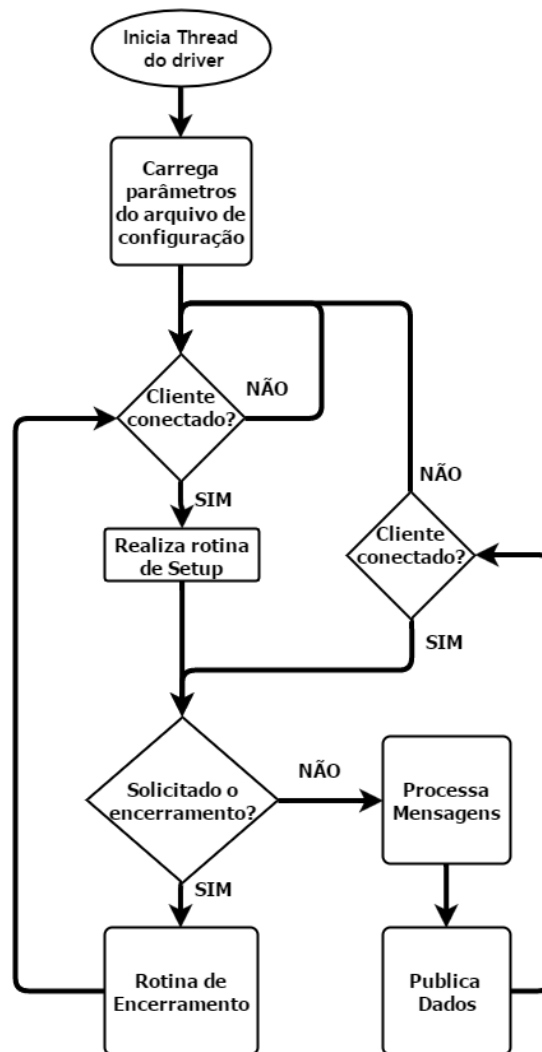


Figura 114 - Diagrama genérico da rotina de um driver para o servidor Player

Ao todo foram escritos seis drivers para acesso dos sensores e atuadores anteriormente apresentados. Os drivers são:

- *AdcDriver*, para acesso às entradas do chip ADC MCP3008,
- *DioDriver*, para acesso às entradas e saídas do chip de expansão de portas digitais MCP23017,
- *RTimuDriver*, para leitura e configuração da central inercial,
- *UltrassomDriver*, para leitura e acionamento dos sensores de ultrassom,
- *OdometroDriver*, para o controle de velocidade dos motores e

- *PlannarDriver*, para receber as informações de orientação e distância ao objetivo.

O fluxograma do funcionamento do programa cliente pode ser observado na Figura 115.

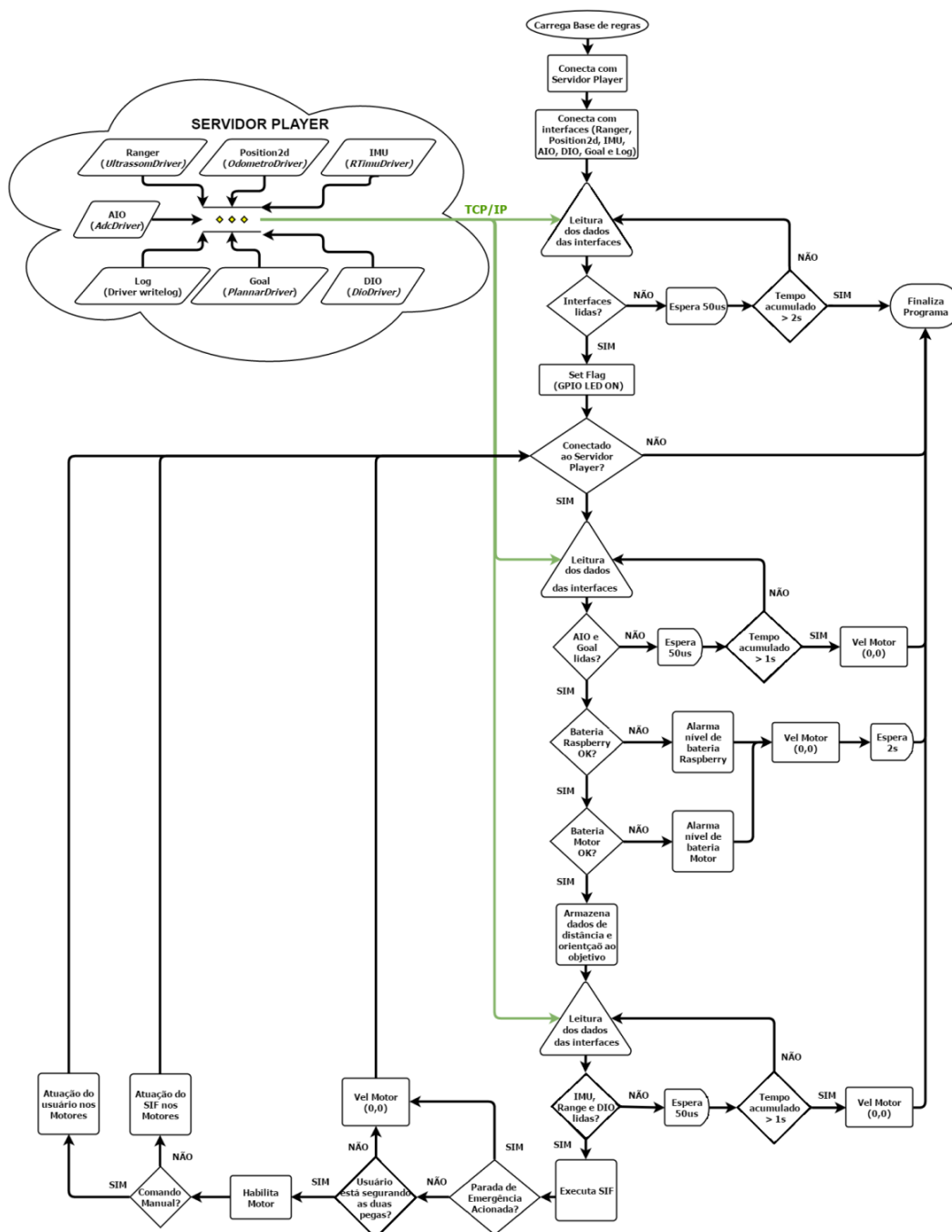


Figura 115 - Fluxograma programa cliente

O programa cliente, cujo fluxograma pode ser observado na Figura 115, começa com a iniciação das variáveis, entre elas a estrutura que armazena o conjunto de regras. Após realizar a conexão com o servidor Player e as interfaces

do robô, é realizada uma leitura inicial, para verificar os acessos aos dados das interfaces. Caso essa leitura não ocorra em um tempo limite, o programa é encerrado.

Um ponto a ser destacado na arquitetura do servidor Player é que, ao enviar um comando de leitura, os dados de todas as interfaces conectadas são lidos; contudo, como o processamento das interfaces ocorre em paralelo e de maneira independente, uma interface pode estar com o dado desatualizado. Devido a essa característica, ao realizar uma leitura, a mesma é repetida até que todas as interfaces de interesse tenham enviado o dado atualizado. Esse procedimento se repete todas as vezes que se deseja ler o dado de uma ou mais interfaces. Nas leituras subsequentes à leitura inicial, o tempo limite é menor, pois na primeira leitura algumas rotinas de *setup* dos drivers podem não estar concluídas, o que aumentaria a latência da primeira leitura.

Com a conclusão da primeira leitura, a porta de saída zero da interface DIO (porta *GPA7* da Figura 112) é colocada em nível lógico alto para identificar que o programa cliente está operacional e, em seguida, o programa entra em seu loop principal. A condição para execução do loop principal é que o programa cliente esteja conectado ao servidor Player, caso contrário, o programa é encerrado. Ao encerrar as conexões com o programa cliente, os drivers executam suas rotinas de encerramento, que serão descritas nas subseções seguintes. Além da perda de conexão, o nível crítico das baterias, ou seja, menor que 9 V para a bateria do Raspberry e menor que 18 V para a bateria do motor, também são condições para o encerramento do programa. Caso não ocorra nenhuma condição de encerramento, o programa continua seu loop de execução, lendo as demais interfaces, executando o SIF e atuando nos motores. A atuação nos motores, por sua vez, depende da condição de o usuário estar segurando a bengala com as duas mãos, lembrando que a atuação dos motores pode ocorrer devido a uma operação direta do usuário ou pela atuação do SIF.

Além do programa cliente, foi escrito outro programa com o intuito de monitorar a execução do programa cliente. Essa execução é monitorada por meio do nível lógico da porta de entrada 7 da interface DIO (porta *GPB0* da Figura 112) e pela lista de processos do sistema operacional. Portanto, caso seja identificada a não execução do programa cliente, o programa supervisor reinicia o servidor Player e, em seguida, o programa cliente. Além dessa atuação automática, o programa

supervisor pode receber uma atuação direta do usuário, por meio do botão de *RESET* apresentado no diagrama da Figura 112.

A seguir, nas subseções 4.2.5.1 a 4.2.5.5, será apresentada uma breve descrição das rotinas estabelecidas nas etapas de execução de cada um dos drivers criados.

#### **4.2.5.1 AdcDriver**

O *AdcDriver* foi criado para encapsular os dados fornecidos pelo chip MCP3008 dentro da interface *Analog I/O* (AIO) servidor Player. A interface AIO é responsável por gerenciar as entradas e saídas analógicas do servidor Player, fornecendo funções para leitura e escrita das mesmas. No caso *AdcDriver*, como o chip ADC utilizado possui somente entradas, o driver não opera as funções de escrita das saídas analógicas.

Em relação à rotina de execução do driver, os parâmetros passados pelo arquivo de configuração são: o valor da tensão de referência, a porta SPI e a quantidade de entradas que serão lidas do ADC. Na rotina de setup é criado o objeto para manipular os dados do MCP3008 e é estabelecida a comunicação SPI com o dispositivo. Já na rotina de encerramento esse objeto é destruído, encerrando a comunicação com o ADC. Esse driver não processa mensagens enviadas por programas clientes e a frequência de leitura das portas analógicas é de 20 Hz.

#### **4.2.5.2 DioDriver**

O *DioDriver* foi criado para gerenciar as portas de entrada e saída do chip MCP23017 e encapsulá-las na interface *Digital I/O* (DIO) do servidor Player. Os parâmetros do arquivo de configuração são: o número de portas configuradas como entrada e o endereço I2C do chip. Na rotina de *setup* do driver é estabelecida a comunicação I2C com o Raspberry Pi e configurada a quantidade de portas que serão utilizadas como entrada e saída. Já a rotina de encerramento a comunicação entre o chip e o Raspberry Pi é finalizada e as portas são colocadas em nível lógico zero. Esse driver processa as mensagens de programas clientes para alterar o estado

lógico das portas de saída e publica o estado lógico das portas de entrada a uma frequência de 20 Hz.

#### **4.2.5.3 RTimuDriver**

O RTimuDriver foi criado para gerenciar os dados da central inercial MinIMU-9 e encapsulá-los na interface *Inertial Measurement Unit* (IMU) do servidor Player. No arquivo de configuração é especificado o formato em que serão publicados os dados da central inercial, ou seja, ângulos de Euler, quatérnion, dados dos sensores sem fusão ou a informação de posição e orientação. Na rotina de *setup* é carregado o arquivo de configuração que contém os dados da calibração dos sensores, o endereço I2C de cada um dos sensores da central inercial e é estabelecida a comunicação com o Raspberry Pi. Na rotina de encerramento a comunicação da central inercial com Raspberry Pi é encerrada. A mensagem processada pelo driver é um comando do programa cliente para alterar o formato em que os dados da IMU serão publicados, ou seja, ângulos de Euler, quatérnion, dados sem fusão ou posição e orientação. A frequência com que os dados da IMU são publicados é de 20 Hz.

#### **4.2.5.4 UltrassomDriver**

O *UltrassomDriver* foi criado para obter os dados dos sensores de ultrassom e encapsulá-los na interface *Ranger* do servidor Player. No arquivo de configuração do driver são passados os seguintes parâmetros: distância mínima e máxima de detecção de obstáculos, número de sensores, quantidade de sensores lidos separadamente e portas utilizadas para a leitura do sinal de largura de pulso de cada sensor. Na rotina de *setup* as portas selecionadas para a leituras dos sensores são configuradas como entradas com interrupção e na rotina de encerramento é desabilitada a interrupção dessas entradas.

O driver não processa mensagens do programa cliente e a frequência com que os dados são publicados depende do número de leituras independentes e do valor desejado para a média das leituras. Essa relação é devido ao fato de que cada leitura independente adicionar uma espera de 50 ms, que é o tempo do ciclo de leitura de

um sensor. Na aplicação desse trabalho, é realizada a leitura de quatro conjuntos de sensores, ou seja, quatro leituras independentes, não sendo realizada a leitura média pelo driver. Portanto, a frequência com que os dados são publicados é de 5 Hz.

#### 4.2.5.5 OdometroDriver

O OdometroDriver foi criado para realizar o controle da velocidade angular de cada motor. Esse driver realiza a leitura da rotação de cada motor por meio dos *encoders* e envia os comandos de atuação para o driver de potência. Os dados dos *encoders* e driver de potência foram encapsulados na interface *position2d* do servidor Player. No arquivo de configuração são enviados como parâmetros: as portas que serão utilizadas para a atuação no driver de potência, para leitura dos dados dos *encoders*, os ganhos do controlador PID e o tempo do loop de controle, que também é utilizado para o cálculo dos termos integral e derivativo. Na rotina de *setup* as portas são configuradas conforme a função escolhida, ou seja, como entrada digital, saída digital, saída PWM ou entrada com interrupção. Também são criadas *threads* para os controladores PID de cada motor e todas as variáveis referentes aos erros do PID, atuação dos motores e posição dos *encoders* são colocados em zero. Na rotina de encerramento, as portas que controlam o driver de potência são colocadas em nível lógico zero, garantindo a não atuação dos motores, as *threads* dos PIDs são encerradas e, assim como na rotina de *setup*, as variáveis dos erros dos PIDs, atuação dos motores e posição dos *encoders* são zeradas. Esse driver processa as seguintes mensagens: velocidade angular desejada para cada motor, habilitar ou desabilitar a atuação dos motores e reset da posição dos *encoders*.

### 4.3 Controle

Nessa seção será apresentada a versão final do controle Fuzzy utilizado para a navegação do dispositivo e o controle PID aplicado na atuação de cada roda. Na Figura 116 é possível observar o diagrama macro da versão do SIF para o modelo real com a aplicação do controle PID e a representação dos sensores e atuadores utilizados.

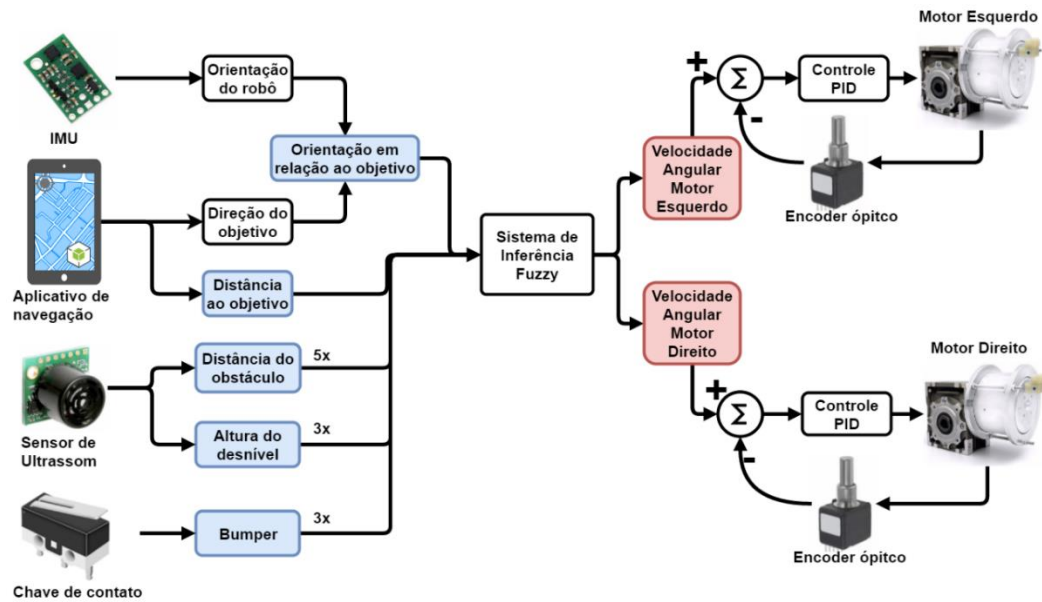


Figura 116 - Diagrama macro do SIF do modelo real com aplicação do controle PID

### 4.3.1 Controle PID

Para realizar o controle da velocidade individual das rodas foram utilizados dois reguladores PID, um para cada roda. O processo para obter os ganhos desses reguladores utilizou o modelo motor-redutor e os parâmetros apresentados na subseção 4.2.1.1. Com base nesses dados, foi criado o circuito de malha fechada com um regulador PID discreto, apresentado na Figura 117. A partir desse novo modelo, foi aplicada a ferramenta gráfica *PID TUNE* do *Simulink*<sup>®</sup> (MathWorks, Inc, 2016c) para a escolha dos ganhos do regulador PID. Essa ferramenta permite ajustar os ganhos do PID baseados no tipo de resposta desejado, por exemplo, uma resposta mais agressiva ou mais robusta.



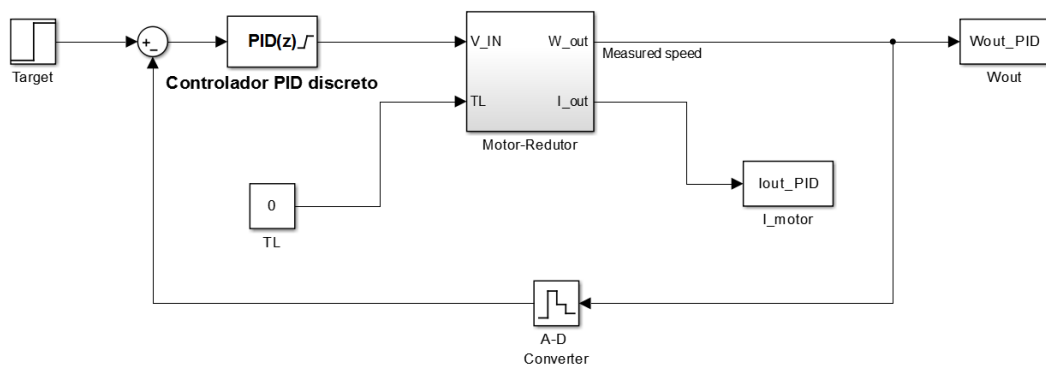


Figura 117 - Circuito de malha fechada do regulador PID discreto do conjunto motor-reductor

Os ganhos do controlador foram sintonizados a partir de uma entrada degrau unitário, tendo como resposta a Figura 118. Os valores obtidos para os ganhos são apresentados na Tabela 5, juntamente o período de operação do controlador PID.

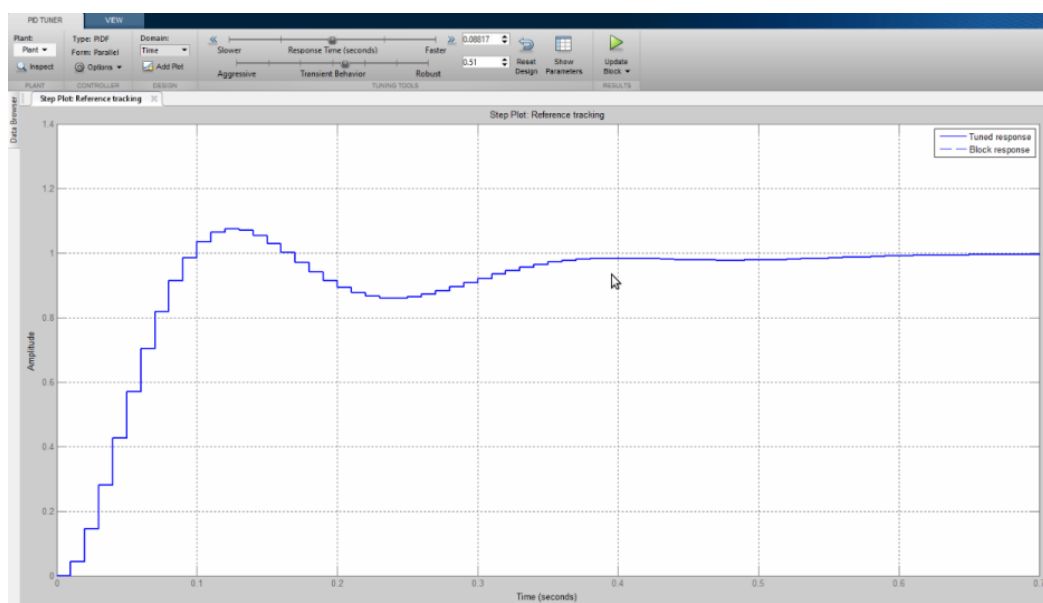


Figura 118 - Resposta ao degrau do diagrama da figura 113

Tabela 5 - Primeiro conjunto de ganhos para o controlador PID

$k_p$	2,0127
$k_i$	18,7412
$k_d$	0,03381
$T$	0,01s

A partir dos ganhos obtidos, foi executado o modelo da Figura 117 variando-se a velocidade angular desejada. O resultado dessa simulação pode ser observado na Figura 119. O mesmo procedimento foi repetido para o conjunto motor-redutor real com o controlador PID implementado no driver *OdometroDriver*, e o resultado pode ser observado na Figura 120. Assim como no resultado simulado, o resultado real obteve uma resposta bem próxima do objetivo.

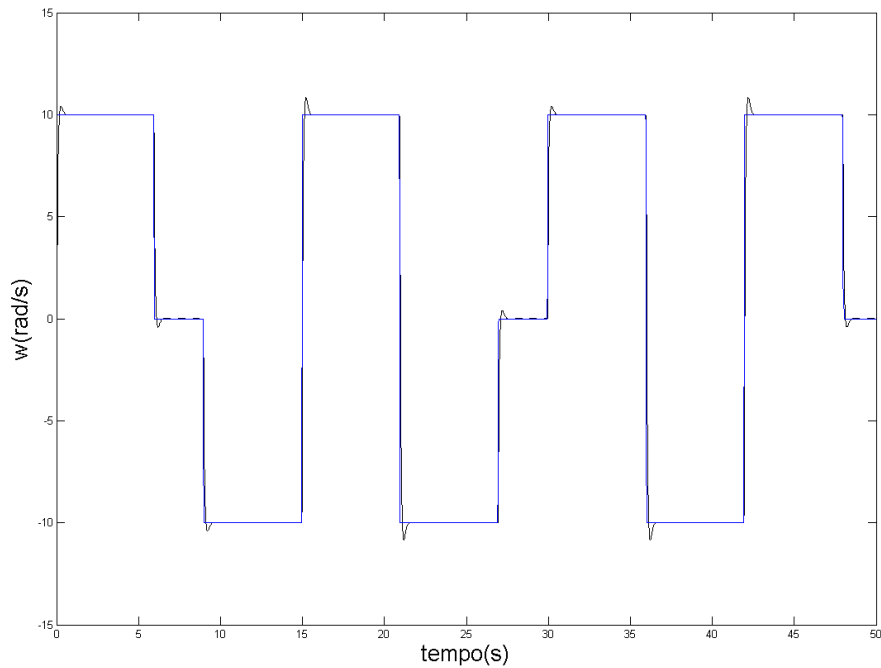


Figura 119 - Resposta do modelo motor-redutor (preto) utilizando o regulador PID com o primeiro conjunto de ganhos (Tabela 5) e saída desejada (azul)

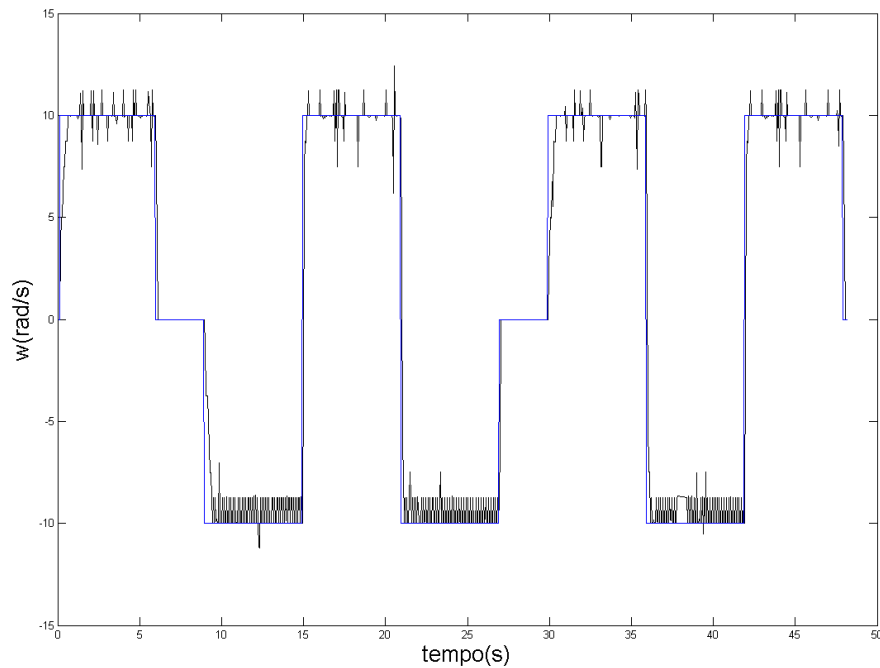


Figura 120 - Resposta do motor-reductor real (preto) com o controlador PID utilizando o primeiro conjunto (Tabela 5) de ganhos e saída desejada (azul)

Um aspecto importante do desenvolvimento do controlador PID do driver *OdometroDriver* a ser destacado é que, para casos que haja inversão do sinal de atuação do PID, não é realizada a inversão no sentido da atuação do motor, mas aplicada uma potência próxima de zero. Esse procedimento foi adotado para evitar picos de correntes excessivos devido a constantes inversões de direção do motor. Outra característica do controlador escrito é que foi estabelecido um limite para o ganho integral, a fim de evitar um grande *overshoot* devido a possíveis restrições no movimento das rodas.

Por fim, os valores dos ganhos do PID obtidos na simulação foram mais uma vez sintonizados, mas agora a partir da observação da resposta no conjunto motor-reductor real. Os ganhos finais obtidos encontram-se na Tabela 6 e a comparação entre as respostas dos dois conjuntos de ganhos pode ser observada na Figura 121. Apesar de apresentarem respostas similares, pelo fato do segundo conjunto ter um ganho derivativo menor, ele é menos sensível a ruídos. Outra característica observada na resposta dos motores, quando utilizado o segundo conjunto de ganhos, é que ele teve uma menor quantidade de *overshoot*. Esse fato teve como consequência uma resposta mais suave na atuação do motor, pois gerou um menor número de desligamentos.

Tabela 6 – Segundo conjunto de ganhos do regulador PID

$k_p$	2
$k_i$	11
$k_d$	0,008
$T$	0,01s

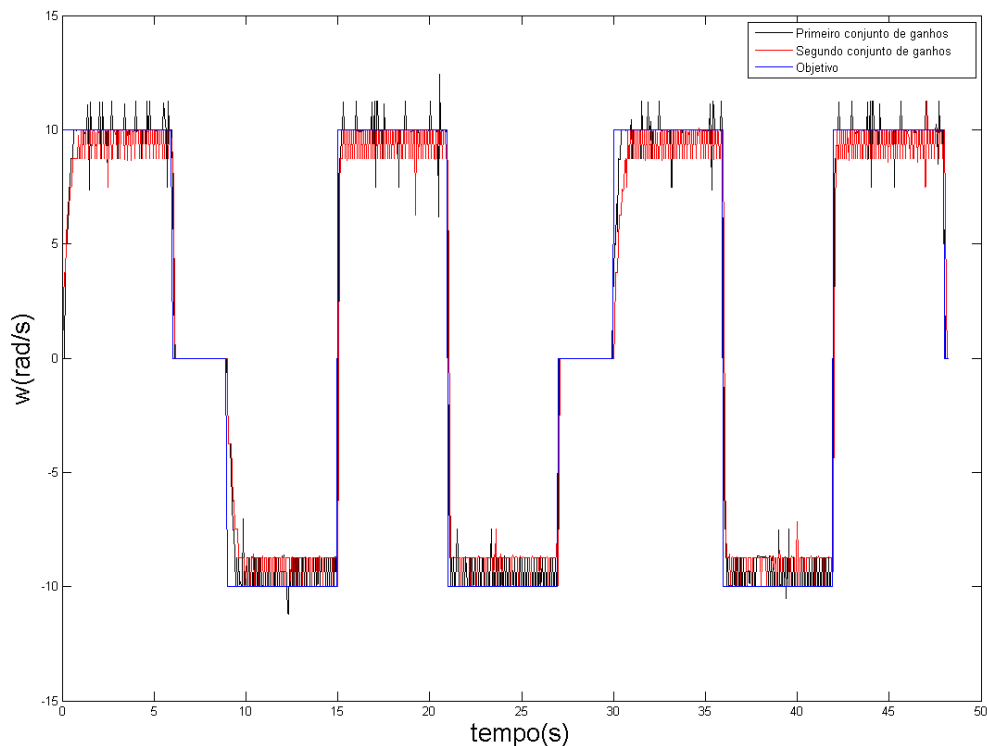


Figura 121 - Saída do motor-reductor real com o regulador PID utilizando conjunto de ganhos da Tabela 5 (preto), o conjunto de ganhos da Tabela 6(vermelho) e a saída desejada (azul)

#### 4.3.2 Lógica Fuzzy

Para aplicação no dispositivo real, o sistema de inferência fuzzy utilizado para a navegação em ambiente virtual teve quatro alterações.

A primeira alteração foi devido à mudança das funções de pertinência dos sensores de altura, S6 a S8. As variáveis fuzzy correspondentes a esses sensores, ou seja, as variáveis *Sensor Altura Esquerda*, *Sensor Altura Frente* e *Sensor Altura Direita*, tiveram as funções de pertinência alteradas para que seu valor fuzzy passasse a representar de fato a distância do sensor ao solo, e não mais a distância ao meio-fio, como havia sido adaptado para o modelo virtual. Para que os sensores

de distância possam medir a informação de altura, seu posicionamento foi direcionado para o solo. Com isso, a informação da distância do robô ao desnível não é mais fornecida, como no caso do modelo simulado, mas sim a presença de um desnível na região de leitura do sensor. Devido a essa característica, para que haja maior antecipação da presença do desnível, os sensores de distância foram posicionados o mais externamente possível do dispositivo e com uma inclinação que possibilite uma medição mais afastada. Na Figura 122 é possível ver com mais detalhe o posicionamento desses sensores, enquanto na Figura 123 são apresentadas as novas funções de pertinência para seus conjuntos fuzzy *Baixa* (BX) e *Alta* (AL).

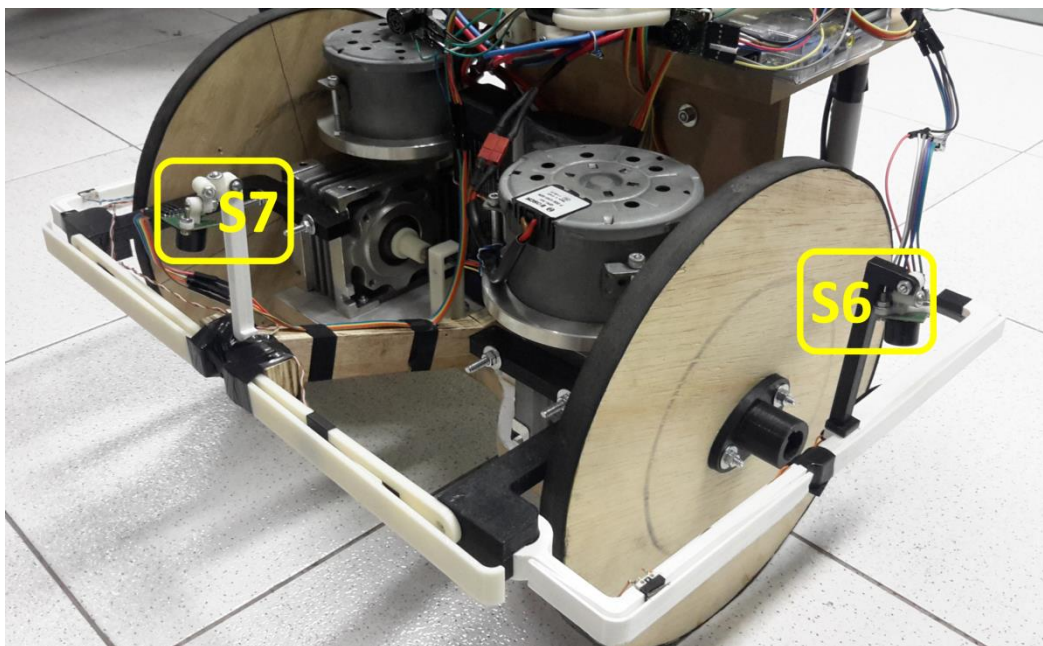


Figura 122 - Posicionamento dos sensores de altura na plataforma real com destaque para o *Sensor Altura Esquerda* (S6) e o *Sensor Altura Frente* (S7)

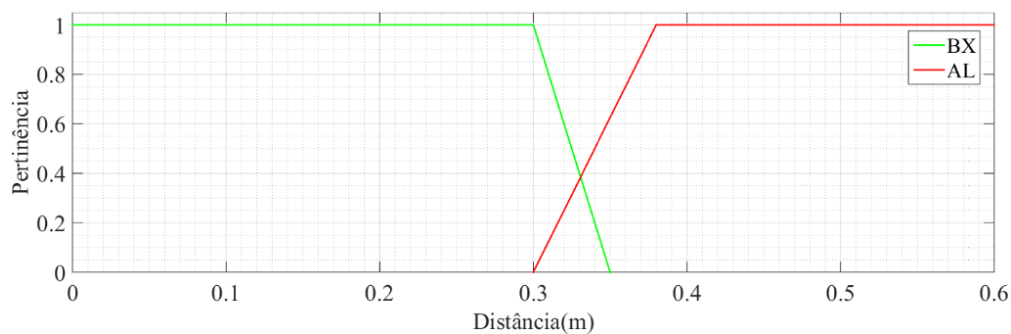


Figura 123 - Funções de pertinência dos sensores de altura do modelo real

A segunda mudança realizada foi no valor da constante que escalona a saída do SIF. No modelo virtual o valor adotado foi de 4,67 e, portanto, a velocidade linear máxima atingida pelo dispositivo era de 0,7 m/s. No modelo real, esse valor foi reduzido para 2,6, reduzindo a velocidade máxima para 0,39 m/s. Essa mudança foi realizada porque, durante os testes com o protótipo, observou-se que a velocidade de 0,7 m/s era alta para uma navegação confortável.

A terceira mudança realizada foi a aplicação de um regulador externo para o conjunto de regras responsável por tratar a configuração na qual o robô está alinhado com o objetivo e seus sensores de distância da parte frontal indicam objeto muito perto. Nessa configuração, caso os sensores laterais enviem a mesma informação, foi observado que o robô tende a oscilar entre a configuração de *Ângulo* negativo e positivo, quando o obstáculo detectado à frente do robô é muito grande. Esse problema ocorre, porque ao desviar para a direção favorável a correção da orientação, o robô passa a errar o valor de *Ângulo* para o sentido oposto, mas sem alterar o estado dos demais sensores de distância. O controle externo aplicado consiste em monitorar a ativação das regras referentes a essa condição para que, quando uma delas for acionada, a regra simétrica não possa atuar até que o robô entre em uma configuração em que ambas as regras não tenham sido ativadas. Durante esse período, apenas a primeira regra a ser ativada tem atuação. A atuação dessa regra será o valor da máxima ativação entre as regras simétricas. Com isso, mesmo que o SIF gere oscilações entre a ativação de regras simétricas, apenas uma das regras terá atuação no sistema.

A quarta e última alteração foi a adição de sensores de contato, ou seja, chaves utilizadas para identificar a ocorrência de colisão do dispositivo. A partir desses novos sensores foram criadas 3 novas variáveis Fuzzy: *Bumper Esquerdo*, *Bumper Frente* e *Bumper Direito*. Assim como nas variáveis linguísticas referentes à altura, as variáveis dos sensores de contato também possuem dois valores Fuzzy, ON e OFF. Contudo, diferente das demais variáveis utilizadas no SIF desse projeto, as funções de pertinência das variáveis *Bumper* são singletons, ou seja, possuem apenas um valor correspondente à pertinência não nula, que é o valor lógico 1, ilustrado na Figura 124

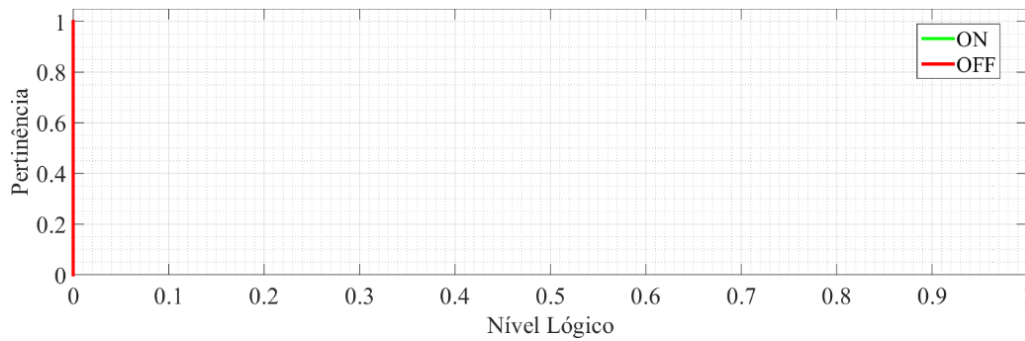


Figura 124 - Funções de pertinência das chaves de contato

A adição dessas novas variáveis linguísticas ao conjunto de regras foi feita por meio do conector “E”, representado pelo operador mínimo. Com a inclusão dessas novas variáveis, o conjunto de regras passou de 518 para 537 e o número de combinações possíveis passou de 19440 para 155520. As regras numeradas de 0 a 511 foram alteradas em relação ao conjunto de regras do modelo virtual, apêndice 8.1, pela adição das variáveis linguísticas *Bumper* (*Esquerdo*, *Frente* e *Direito*) com o valor Fuzzy *OFF*, já a nova composição das regras 512 a 536 pode ser observada no apêndice 8.2.

No próximo capítulo, serão apresentados os resultados dos experimentos realizados com o protótipo descrito nesse capítulo e com o protótipo virtual, descrito no capítulo anterior.

## 5

### Testes e Resultados

Com a conclusão das definições dos modelos real e simulado, que foram apresentados nos capítulos 3 e 4, neste capítulo são apresentados os testes e resultados desses dois modelos. Os primeiros testes foram realizados em ambiente virtual, para que a validação do modelo ocorresse de maneira rápida e que não compromettesse a estrutura do dispositivo real. A partir da melhor configuração obtida do modelo virtual, foram realizados os testes em ambiente real.

#### 5.1.

##### Modelo Simulado

Nessa seção são apresentados os resultados da navegação em ambiente simulado. Ao todo são apresentados três trajetos, cujas disposições dos obstáculos geram diferentes níveis de dificuldade para a navegação. Todos os trajetos foram realizados com a mesma configuração do SIF.

Com relação às dimensões do mapa, conforme mencionado no capítulo 3, as calçadas possuem largura de 2 m, o robô possui largura de 0,4 m e, como referência das dimensões dos objetos, cada quadrado da grade do plano de fundo do mapa possui área de  $1\text{ m}^2$ .

##### 5.1.1.

###### Trajetos 1

O primeiro trajeto inicia com o robô alinhado com o primeiro objetivo, destacado em amarelo na Figura 125. À esquerda do robô, linha sólida azul, está o meio-fio da calçada, detectável apenas pelos sensores de distância S6 a S8, enquanto os demais objetos do mapa, desenhados em preto, são detectáveis apenas pelos sensores S1 a S5. O centro do círculo amarelo indica o objetivo da navegação e o círculo amarelo, de raio 2 m a tolerância para alcançar esse objetivo



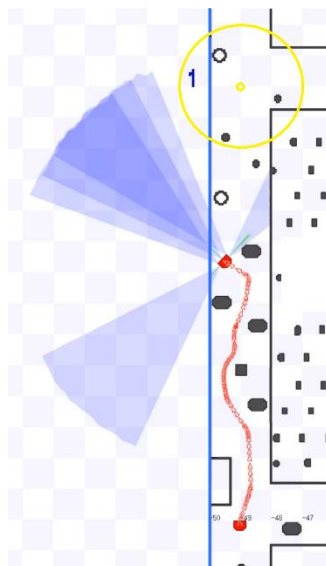


Figura 125 - Percurso realizado na primeira parte do trajeto 1

Os dados referentes aos sensores e ao SIF dessa primeira etapa do trajeto 1 podem ser observados na Figura 126 a Figura 130, onde: a Figura 126 apresenta as informações de distância e orientação em relação ao objetivo; a Figura 127 e Figura 128 as informações dos sensores de distância, S1 a S5, utilizados para detectar os obstáculos; a Figura 129 a informação dos sensores de distância, S6 a S8, utilizados para detectar a distância ao meio-fio da calçada; e, por último, na Figura 130 são apresentadas as saídas do SIF e as velocidades angular e linear executadas pelo robô. Para facilitar a visualização da transição dos valores, todos os pontos referentes aos instantes de leitura ou atuação foram interligados.

Como as saídas do SIF são normalizadas entre -1 e 1, as mesmas foram multiplicadas pela constante 4,67 para que a velocidade linear máxima exercida pelo dispositivo fosse de 0,7 m/s. Em relação aos gráficos dos sensores de distância e da orientação ao objetivo (Figura 126 a Figura 129), para melhor compreensão das influência dos valores obtidos, foram inseridas, à direita dos gráficos, representações dos conjuntos de pertinência das variáveis Fuzzy que os representam.

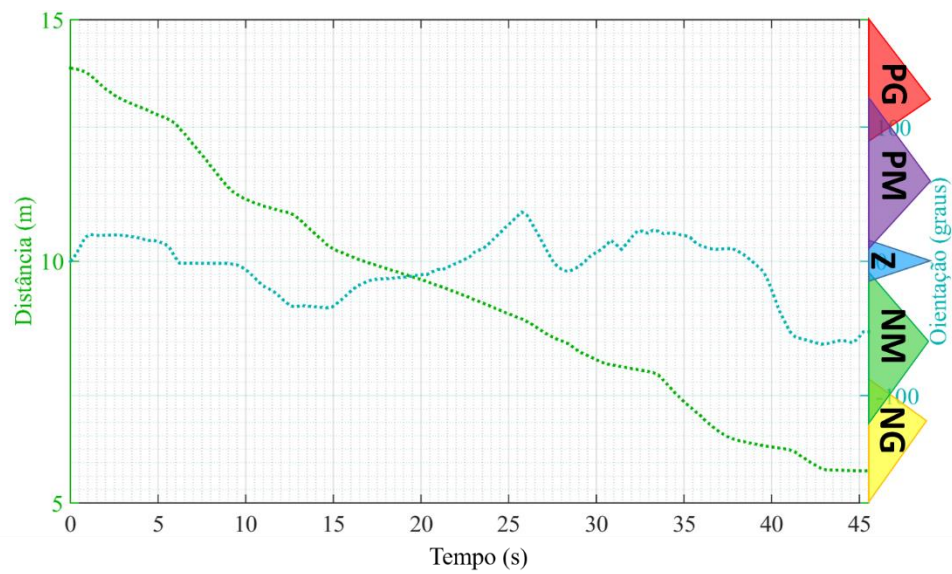


Figura 126 - Dados de orientação e distância ao objetivo durante a realização da primeira etapa do trajeto 1

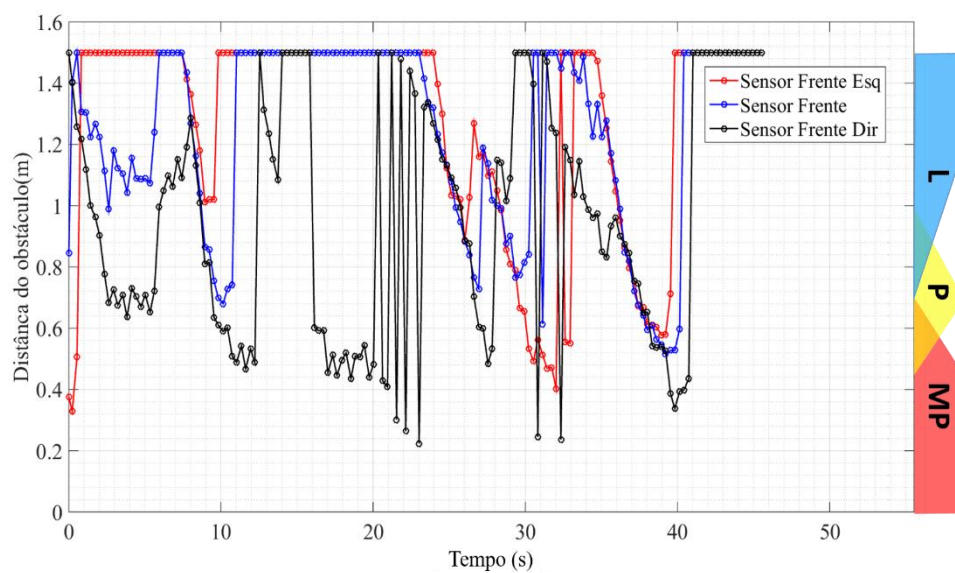


Figura 127 - Dados dos sensores de distâncias frontais (S2, S3 e S4) durante a realização da primeira etapa do trajeto 1

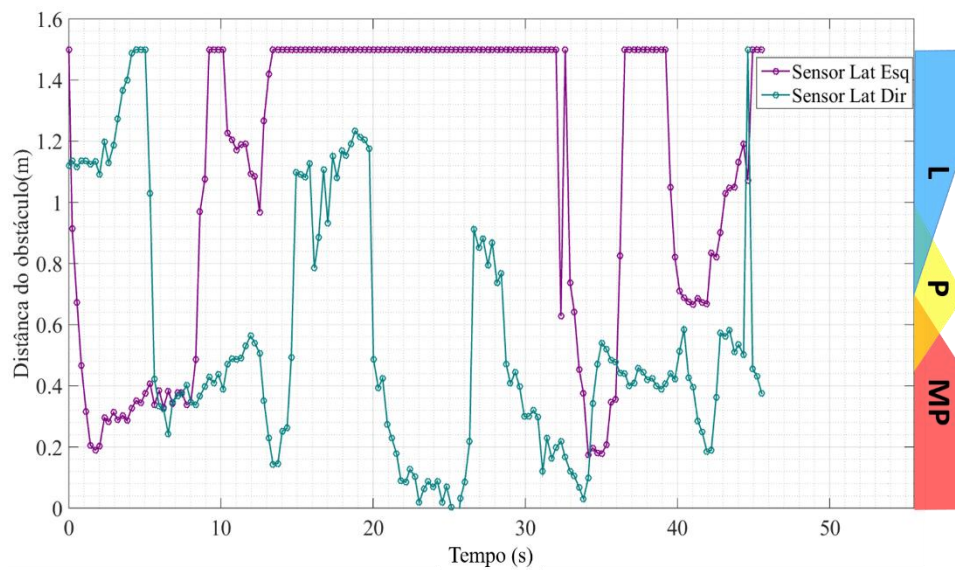


Figura 128 - Dados dos sensores de distâncias laterais (S1 e S5) durante a realização da primeira etapa do trajeto 1

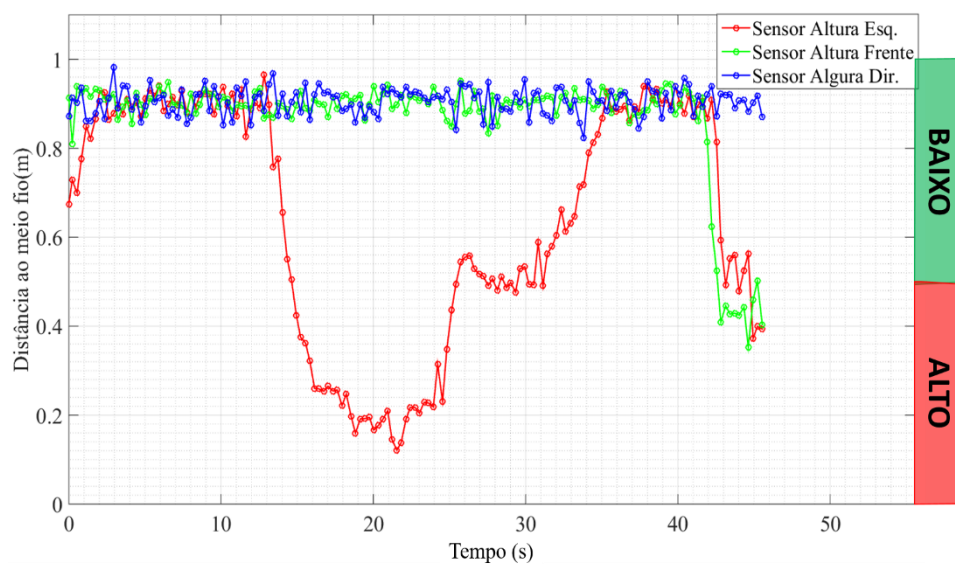


Figura 129 - Dados dos sensores de distâncias ao meio-fio durante (S6, S7 e S8) a realização da primeira etapa do trajeto 1

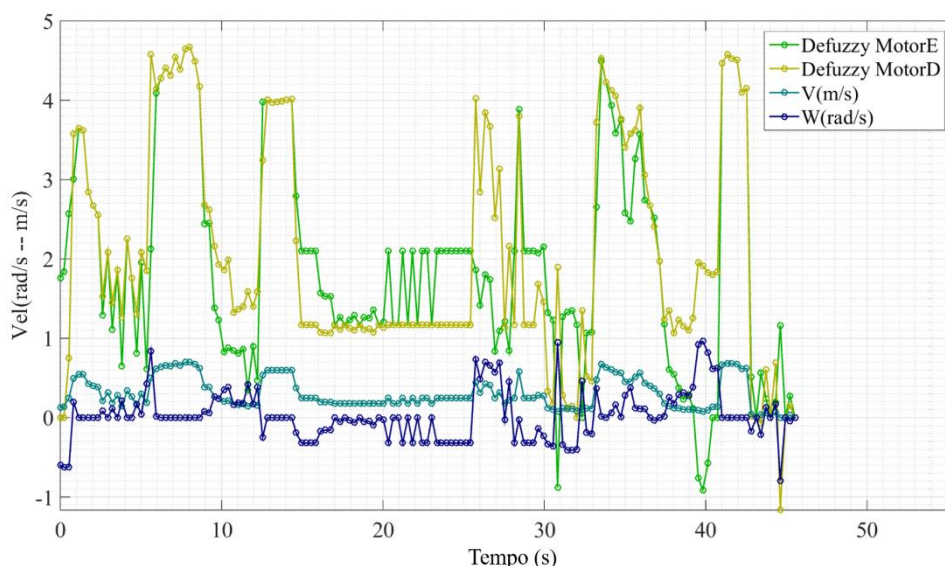


Figura 130 - Velocidades aplicadas durante a realização da primeira etapa do trajeto 1

Nesse primeiro trajeto o robô não consegue chegar ao objetivo, pois, ao desviar de um obstáculo, ele entra em uma configuração na qual detecta uma calçada a sua frente e lateral (Figura 129 instante  $t = 45$  s). Nessa configuração o SIF tem como resposta uma atuação nula (Figura 130 instante  $t = 45$  s), pois não possui informação suficiente para garantir a segurança do usuário ao manobrar. Portanto, nesse caso, o usuário deverá ativar o controle manual, ou seja, operar o robô por meio dos sensores de força posicionados nas pegadas da bengala, para alterar o estado do robô. Esse procedimento foi realizado no simulador, reposicionando o robô manualmente de modo que apenas o sensor da lateral esquerda detectasse a calçada (Figura 131). Com isso, o SIF passou a ter a possibilidade de se deslocar de maneira mais segura para o usuário. A sequência dos passos do robô até atingir o objetivo pode ser observada na Figura 131, onde o robô reposicionado segue até o objetivo final. Na Figura 132 a Figura 136 são apresentados os dados dos sensores e atuadores do instante após o reposicionamento até o final do percurso. Os dados referentes ao intervalo de tempo entre a atuação nula do SIF e o reposicionamento manual do robô foram omitidos para facilitar a visualização da atuação autônoma.

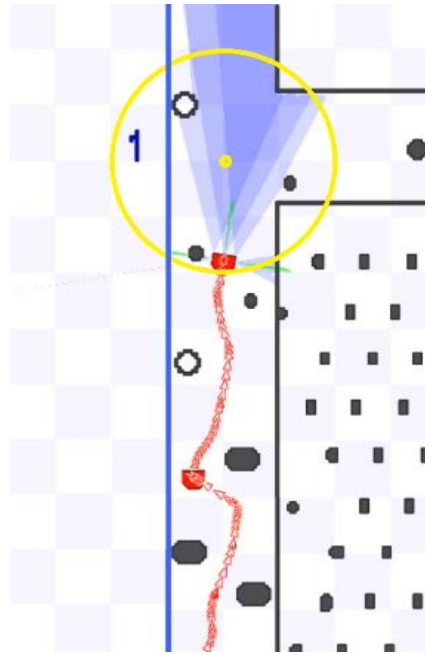


Figura 131 - Percurso realizado na etapa final do trajeto 1

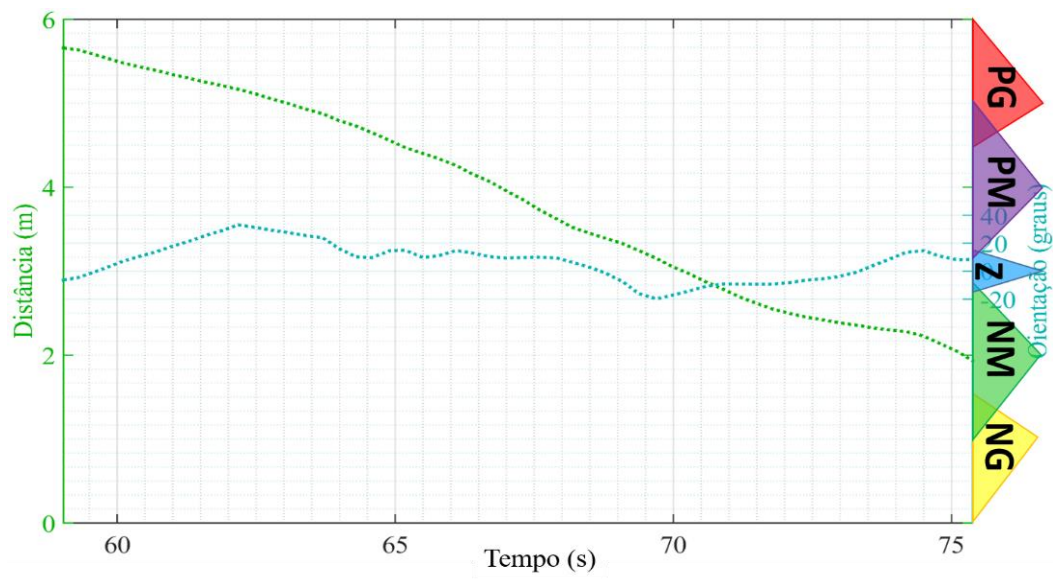


Figura 132 - Dados de orientação e distância ao objetivo durante a realização da etapa final do trajeto 1



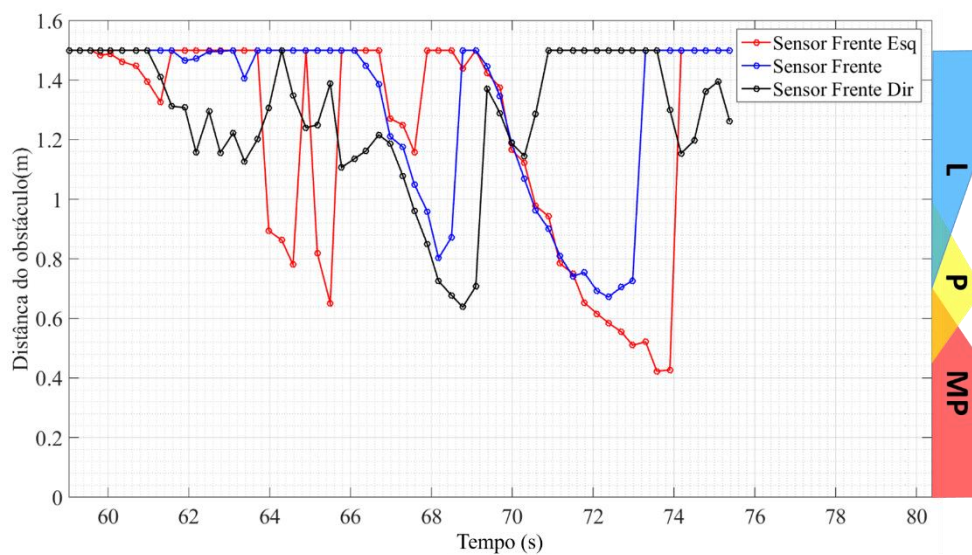


Figura 133 - Dados dos sensores de distâncias frontais durante a realização da etapa final do trajeto 1

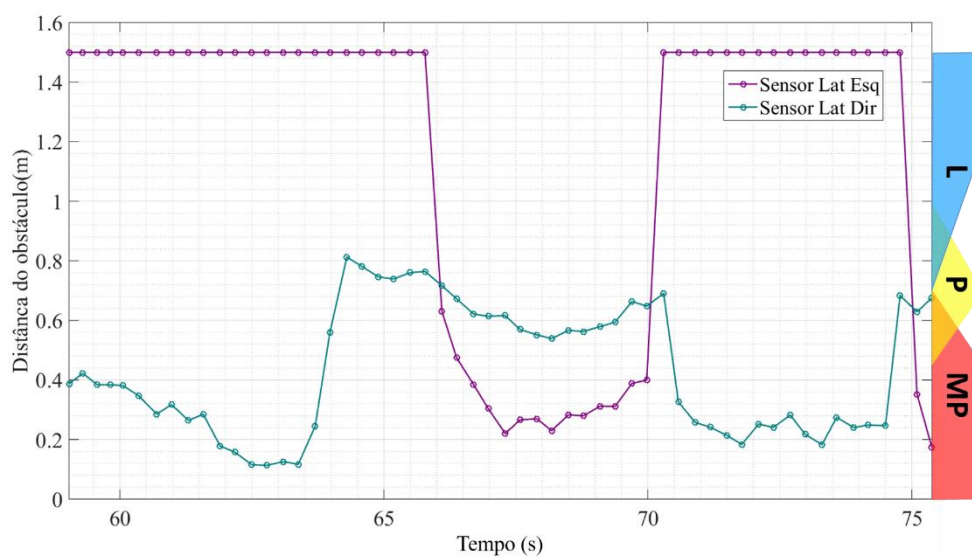


Figura 134 - Dados dos sensores de distâncias laterais durante a realização da etapa final do trajeto 1

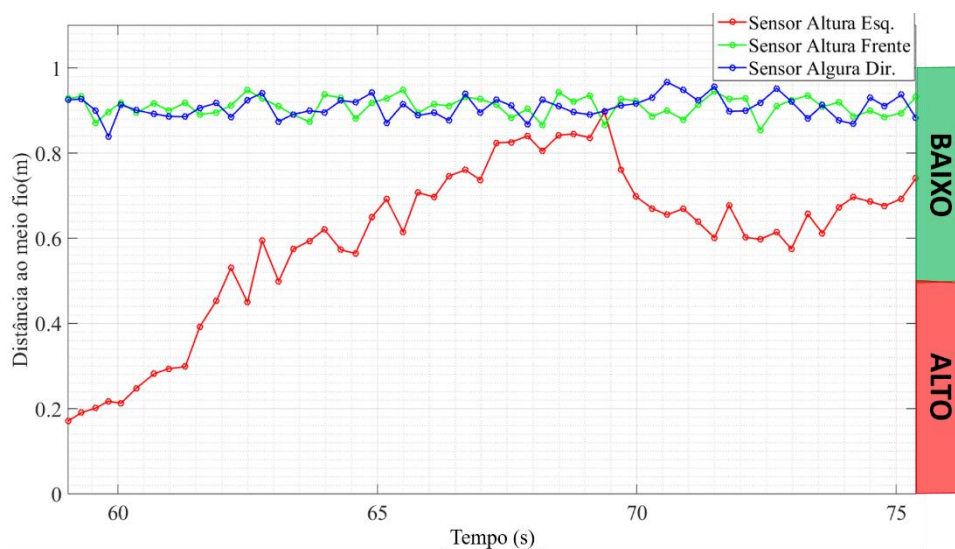


Figura 135 - Dados dos sensores de distâncias ao meio-fio durante a realização da etapa final do trajeto 1

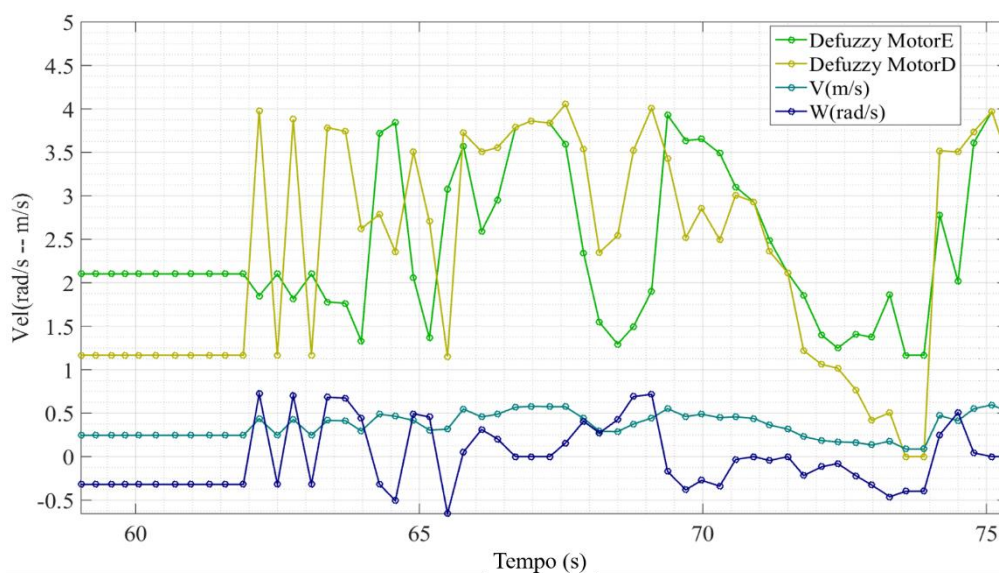


Figura 136 - Velocidades aplicadas durante a realização da etapa final do trajeto 1

### 5.1.2. Trajeto 2

Nesse trajeto, o robô inicia do ponto de parada ao alcançar o objetivo do trajeto 1. O trajeto 2, apresentado na Figura 137, simula uma condição de corredor, onde em ambos os lados do trajeto existem paredes. Entre o robô e o objetivo, existem objetos espalhados de forma mais centralizada no corredor. Assim como no trajeto 1, da Figura 137 a Figura 142 são apresentados o percurso executado pelo

robô, as leituras de seus sensores e as saídas do SIF, durante a execução do trajeto 2.

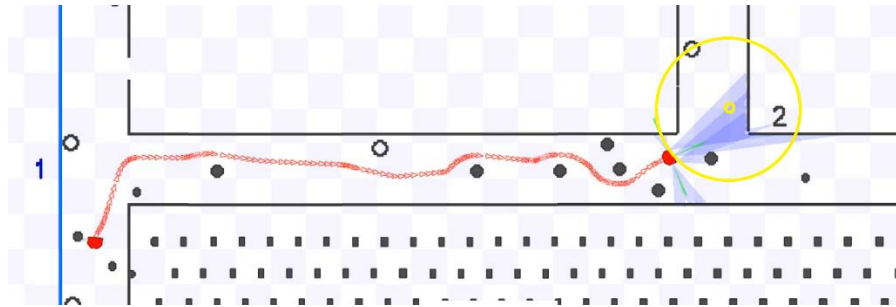


Figura 137 - Percurso realizado no trajeto 2

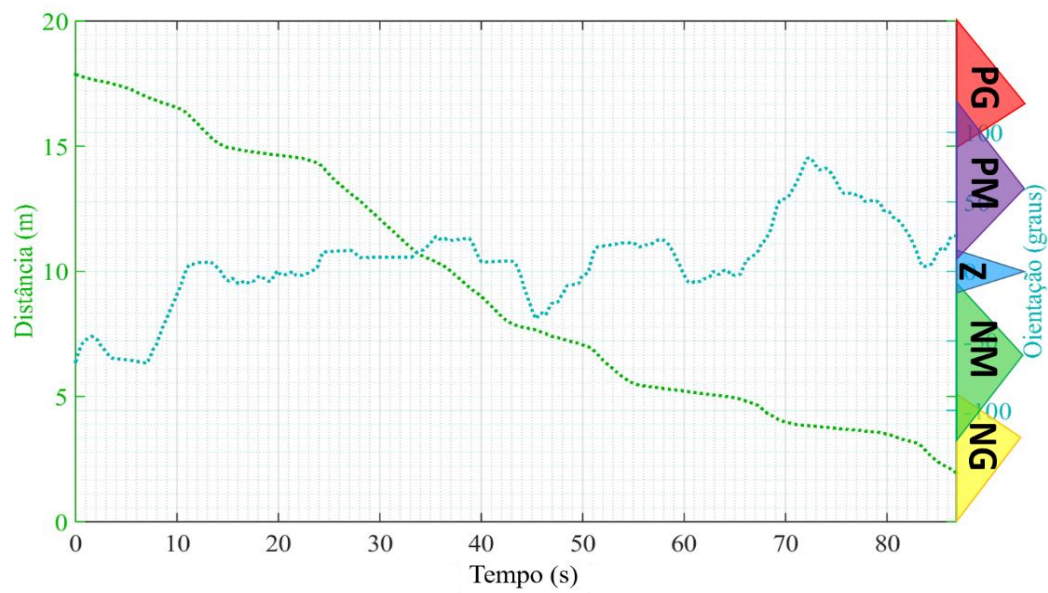


Figura 138 - Dados de orientação e distância ao objetivo durante a realização do trajeto 2



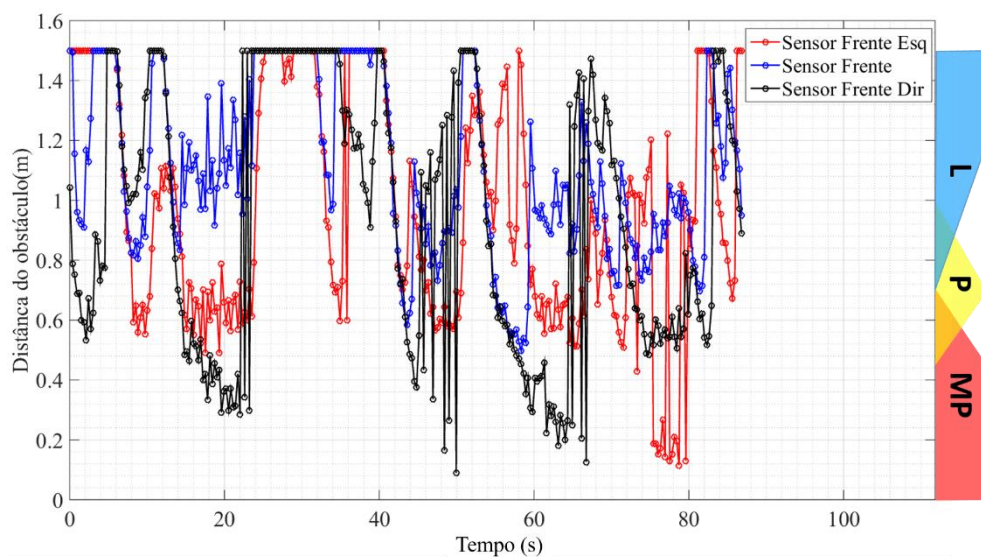


Figura 139 - Dados dos sensores de distâncias frontais durante a realização do trajeto 2

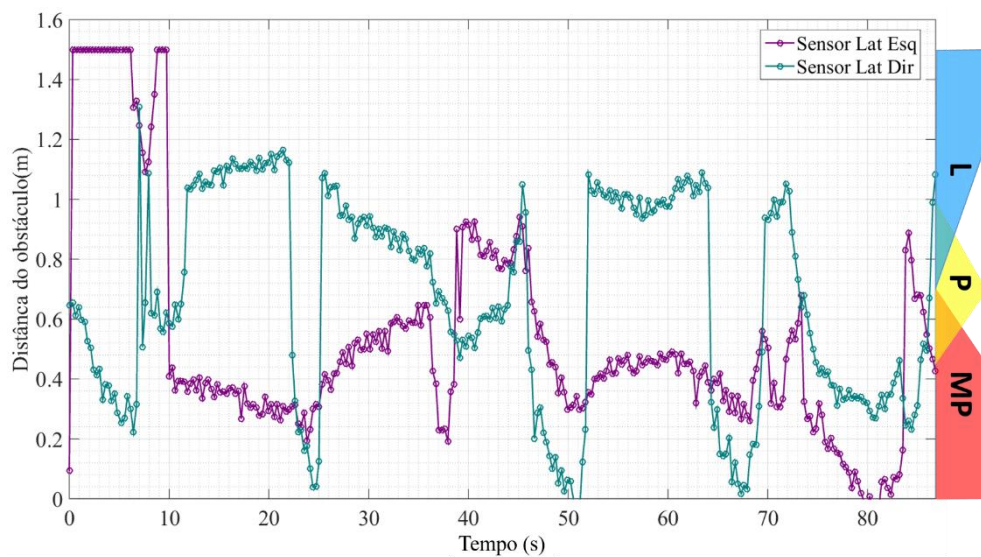


Figura 140 - Dados dos sensores de distâncias laterais durante a realização do trajeto 2

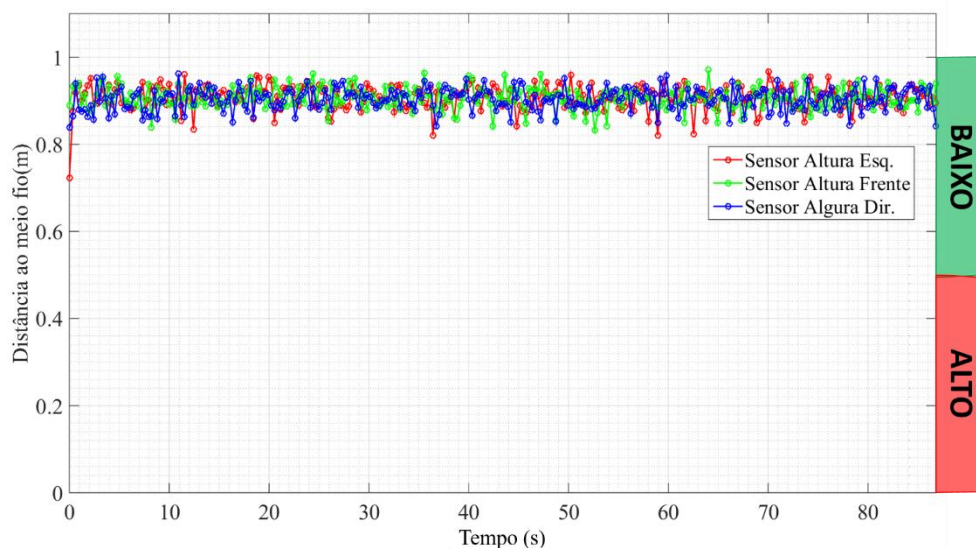


Figura 141 - Dados dos sensores de distância ao meio-fio durante a realização do trajeto 2

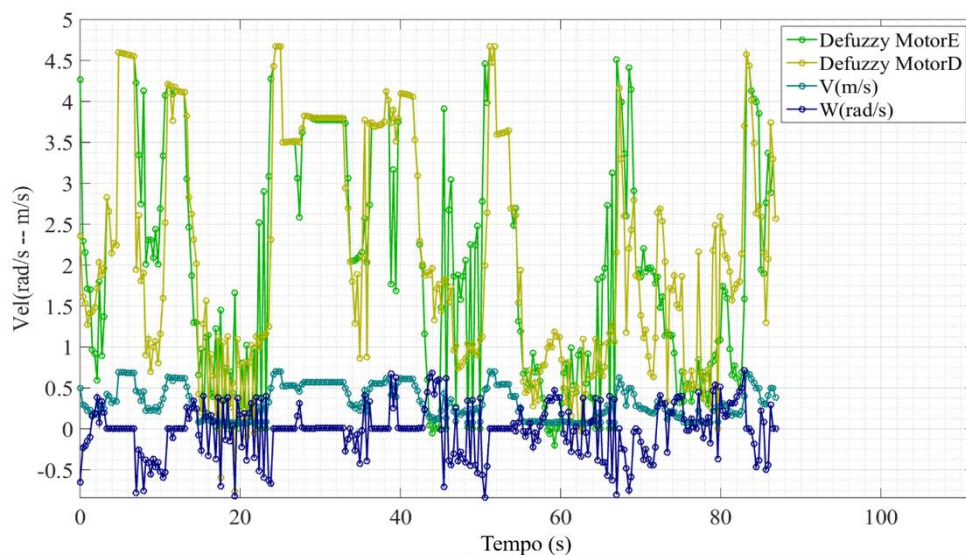


Figura 142 - Velocidades aplicadas durante a realização do trajeto 2

Na execução desse trajeto, o robô conseguiu desviar de todos os obstáculos, realizando conversões suaves e aplicando a máxima velocidade nas regiões em que possui obstáculos com distância *Longe* (L). Apesar de o robô apresentar oscilações ao passar por regiões com muito pouco espaço de manobra, como o primeiro obstáculo centralizado no corredor, entre os instantes  $t = 17$  s e  $t = 22$  s da Figura 139 e Figura 140, a amplitude das oscilações é baixa, cerca de 0,5 rad/s, e o robô não executa uma oscilação pura, mas sempre se desloca em relação ao objetivo (Figura 142 entre os instantes  $t = 17$  s e  $t = 22$  s.)

Nos instantes finais do percurso,  $t > 60$  s, o robô encontra-se alinhado com o objetivo (Figura 138 instante  $t = 60$  s) e próximo ao último obstáculo centralizado no corredor (Figura 137). A partir desse instante, o robô começa a se desviar, errando sua orientação com o objetivo, para ultrapassar o último obstáculo centralizado no corredor e, em seguida, contorna de maneira suave os últimos três obstáculos entre ele e o objetivo.

### **5.1.3. Trajeto 3**

Este último trajeto do ambiente virtual está dividido em duas partes. A primeira parte, objetivo 3 na Figura 143, possui obstáculos que forçam o robô a se alinhar com o centro do “corredor”, e em seguida o robô confronta um novo obstáculo alinhado entre ele e objetivo. Essa etapa tem como objetivo avaliar a capacidade de manobra do robô ao encontrar um obstáculo alinhado com o objetivo. Na segunda parte, objetivo 4 na Figura 143, não possui paredes, como nos casos anteriores, mas diversos obstáculos espalhados entre o robô e o objetivo. Essa segunda parte tem como objetivo avaliar a capacidade do robô em manter-se alinhado ao objetivo. Na Figura 143 é possível observar o percurso realizado pelo robô para alcançar os objetivos 3 e 4.

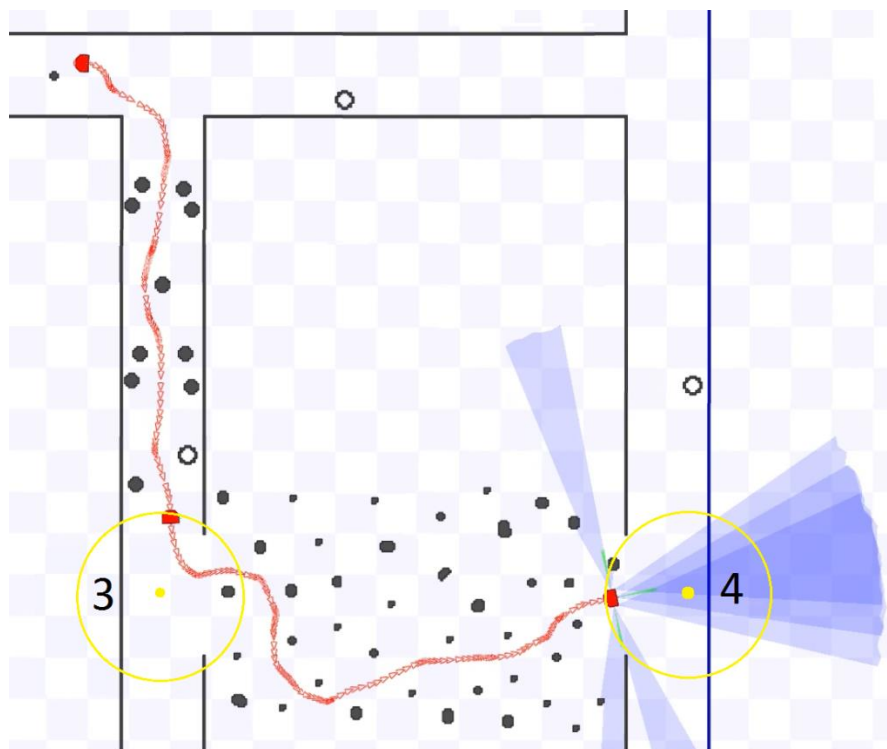


Figura 143 - Percurso realizado no trajeto 3

Assim como nos trajetos anteriores, o robô consegue realizar o trajeto 3 de maneira suave e sem colisões. Na Figura 144 é possível visualizar a saída do SIF e as velocidades linear e angular executadas pelo robô durante todo o trajeto 3, já da Figura 145 a Figura 148 são apresentados os dados dos sensores. No intervalo de tempo entre de  $t = 0$  s e  $t = 65$  s ocorreu a primeira parte do trajeto 3, já a segunda parte ocorreu entre os instantes  $t = 70$  s e  $t = 135$  s.

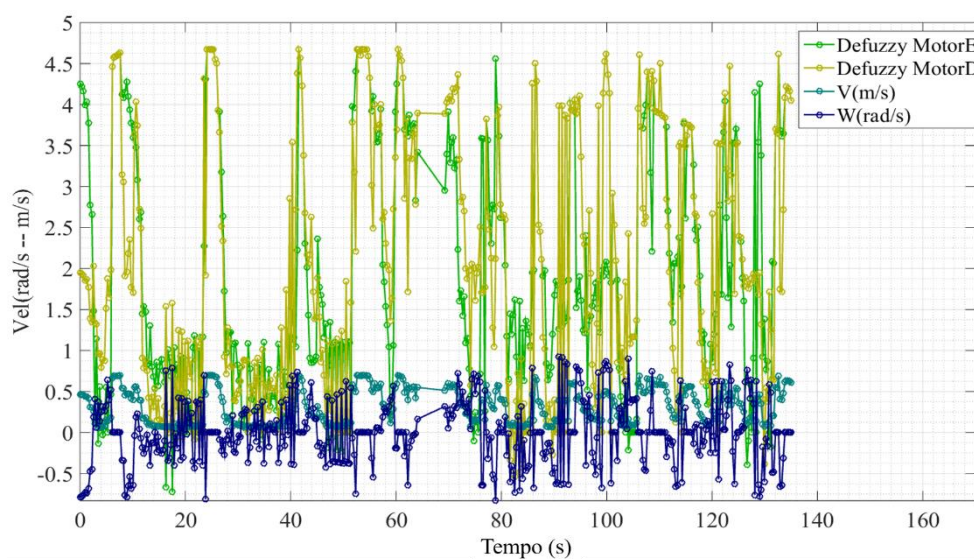


Figura 144 - Velocidades aplicadas durante a realização do trajeto 3



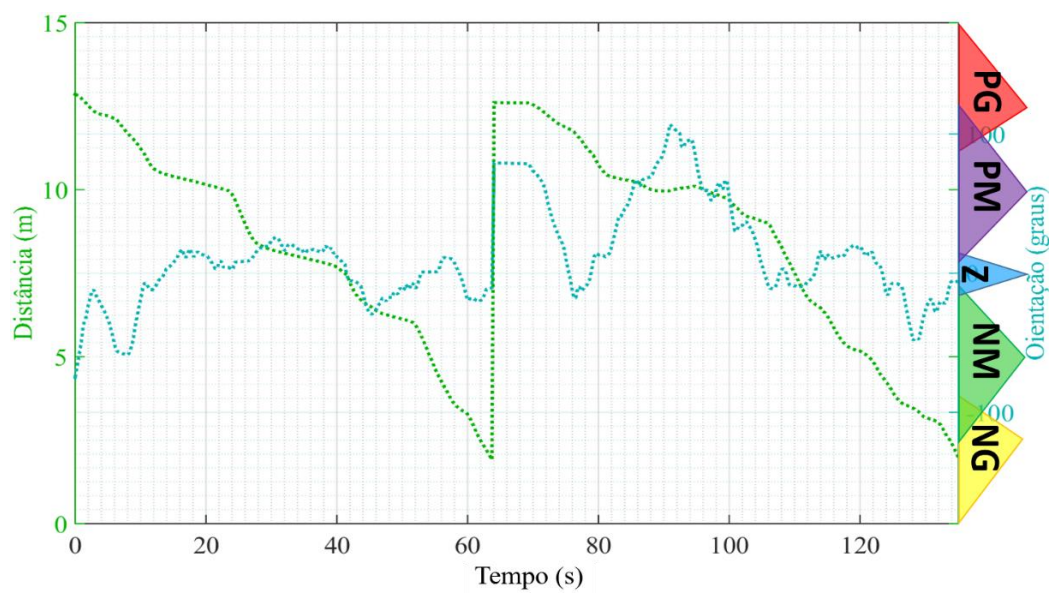


Figura 145 - Dados de orientação e distância ao objetivo durante a realização do trajeto 3

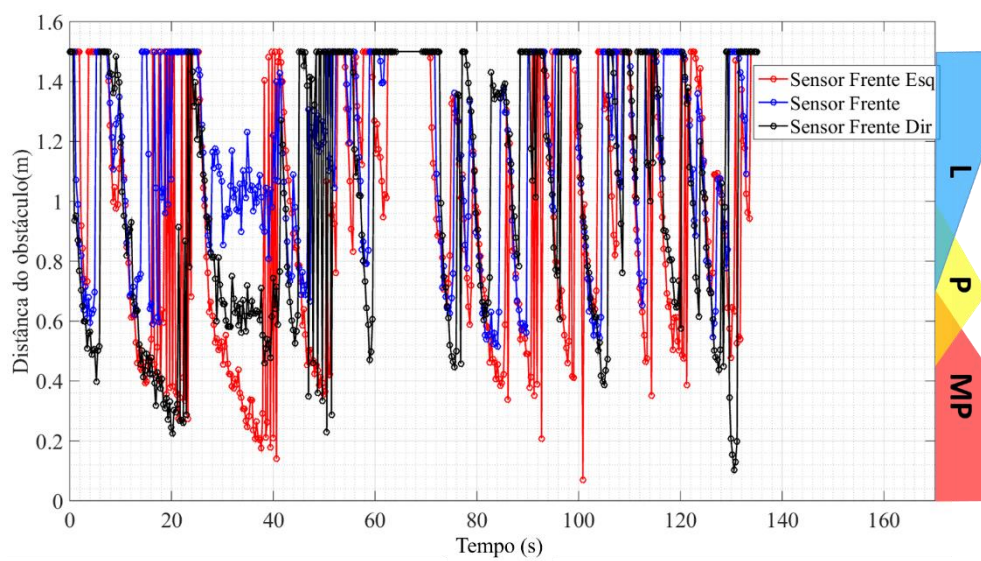


Figura 146 - Dados dos sensores de distâncias frontais durante a realização do trajeto 3

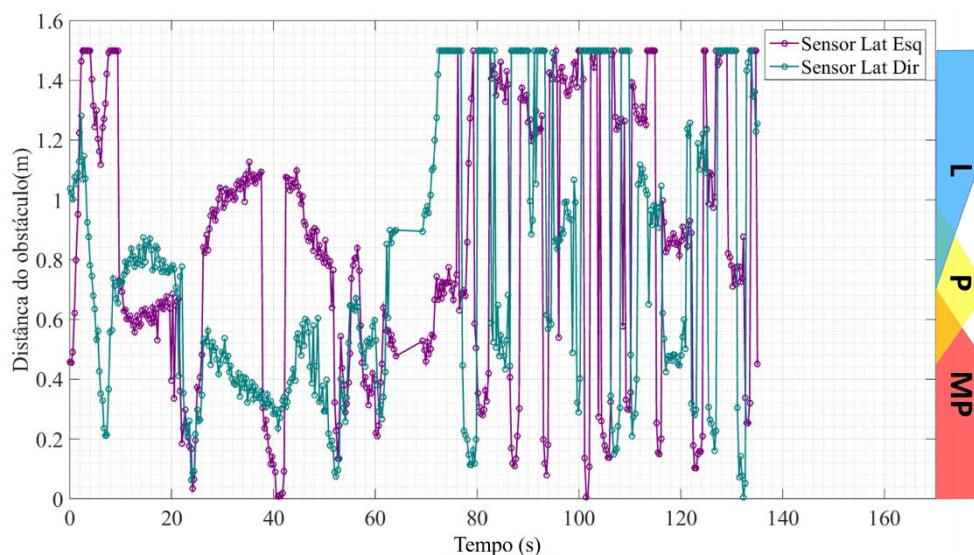


Figura 147 - Dados dos sensores de distâncias laterais durante a realização do trajeto 3

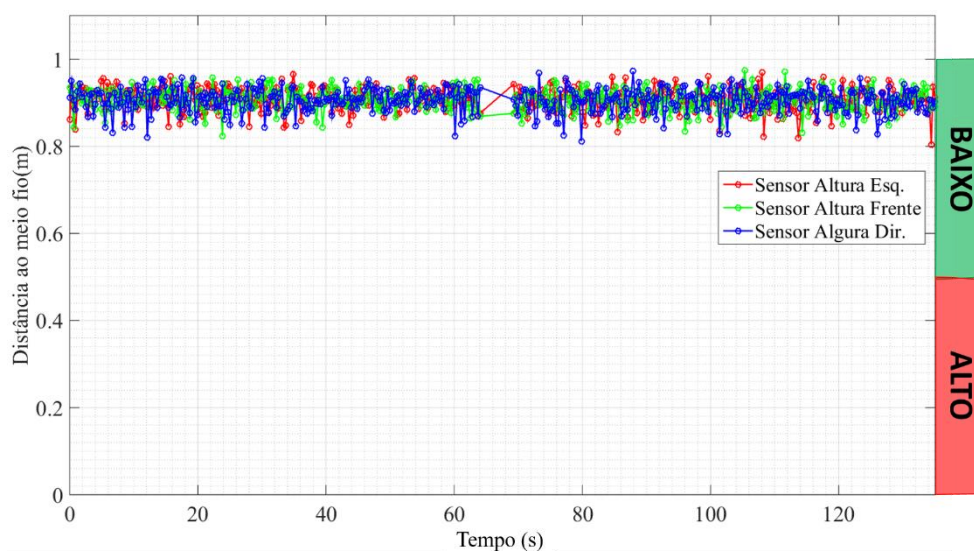


Figura 148 - Dados dos sensores de distância ao meio-fio durante a realização do trajeto 3

Com os testes realizados no trajeto 3, conclui-se a etapa de testes no ambiente virtual. A seguir, na seção 5.2, são apresentados os testes e resultados realizados com o robô real.

## 5.2. Modelo Real

Nessa seção são apresentados os testes realizados com a plataforma real. O ambiente utilizado para os testes foi uma calçada de largura de 3 m e altura do meio-fio de 16 cm. Para realizar os testes de navegação e desvio, foram inseridos 5 cones,

espaçados em torno de 1 m, no percurso de navegação da calçada. Além dos objetos inseridos, a calçada possui um banco, totalizando 6 obstáculos que o robô pode confrontar durante sua navegação, além do desnível gerado pelo meio-fio da calçada. Ao todo foram realizados três testes que se diferem pela posição e orientação do robô em relação ao objetivo, e pela direção do objetivo em si, que foi alternado entre os dois sentidos da calçada.

O teste foi realizado dentro do ambiente da PUC-Rio e, portanto, não há rota de navegação que possa ser extraída da API do Google Maps. Para que o robô receba a informação de distância e orientação, a mesma foi enviada manualmente por meio de um programa cliente do servidor Player. Esse programa altera a informação de distância e orientação ao objetivo por meio de um comando manual. Devido a essa característica, a informação de distância ao objetivo não é atualizada durante o percurso de teste, pois a mesma deve ser corrigida por meio do sensor GPS do celular instalado com a API do Google Maps, conforme exposto na subseção 2.3.3.

A seguir, nas subseções seguintes, serão apresentados em detalhes os percursos e resultados obtidos em cada um dos três testes.

### **5.2.1. Percurso 1**

O primeiro percurso inicia com o robô a -177 graus do objetivo, como consta na Figura 149. Conforme citado na seção anterior, a informação de distância ao objetivo não é atualizada durante o trajeto de teste. Portanto, o objetivo do robô é apenas a orientação, ou seja, no caso desse primeiro trajeto, a direção paralela à calçada no sentido do fundo da Figura 149. Ao ultrapassar todos os obstáculos, é considerado que o objetivo foi alcançado e o teste é encerrado. Na Figura 149, traçado em vermelho, é possível observar o trajeto aproximado que o robô realizou durante o percurso 1.

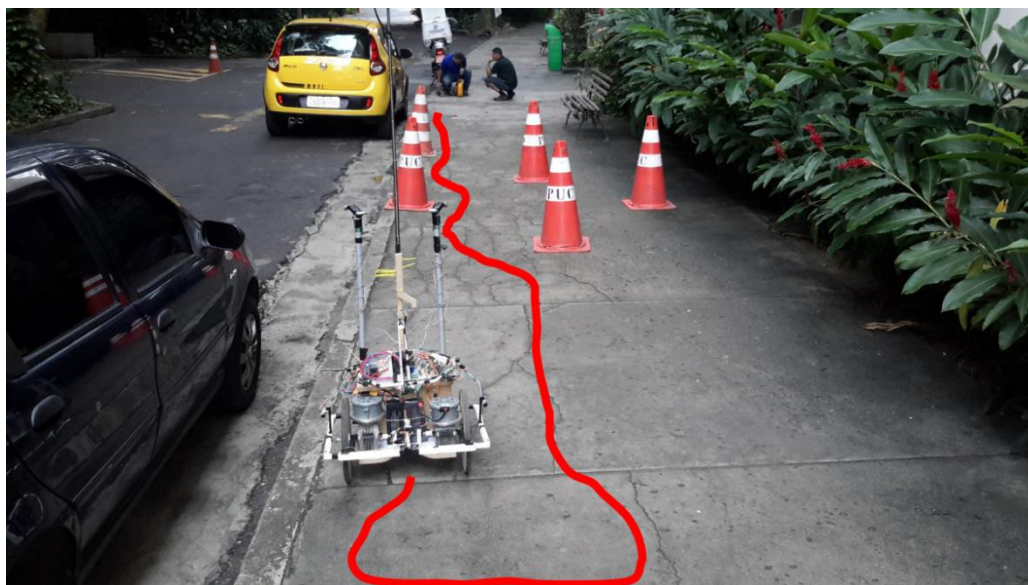


Figura 149 - Posição inicial para a realização do percurso 1

Os dados dos sensores e atuadores, durante a execução do trajeto de teste do modelo real, estão dispostos conforme a apresentação realizada durante os testes em ambiente virtual e podem ser observados na Figura 150 a Figura 154. Dentre as diferenças entre os dados apresentados no ambiente virtual e no ambiente real, está o gráfico dos sensores de altura (Figura 154), que agora medem a distância da posição do sensor ao chão, e não mais a distância do robô à lateral da calçada virtual. Esses valores estão conforme as funções de pertinência da variável dos sensores de altura, apresentadas na seção 4.3.2. A segunda diferença se dá pela saída do SIF, pois o valor da saída, normalizado entre -1 e 1, agora passa a ser multiplicado pelo valor de 2,6. Essa correção é necessária para alterar a velocidade linear máxima do robô para 0,39 m/s, e não mais os 0,7 m/s do modelo virtual. Essa mudança do valor de velocidade foi adotada por ter sido considerada, durante os testes de navegação, desconfortável a velocidade de 0,7 m/s para a operação do dispositivo.



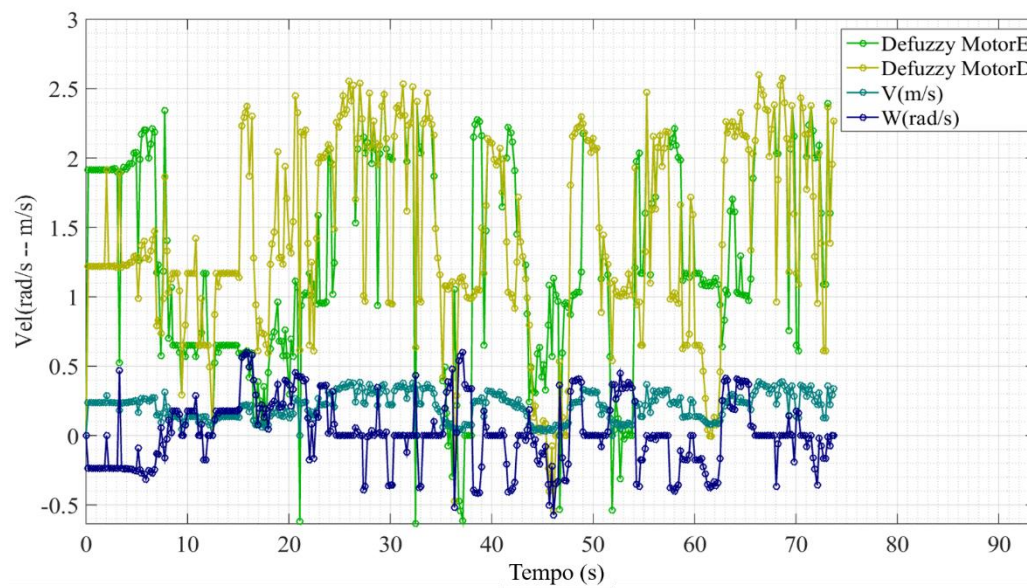


Figura 150 - Velocidades aplicadas durante a realização do percurso 1

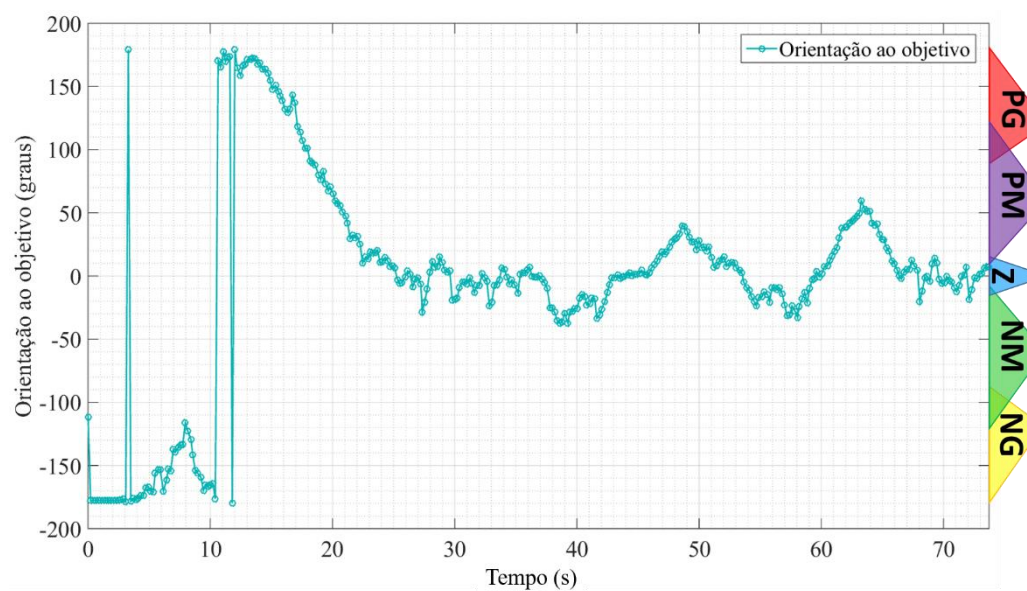


Figura 151 - Dados de orientação ao objetivo durante a realização do percurso

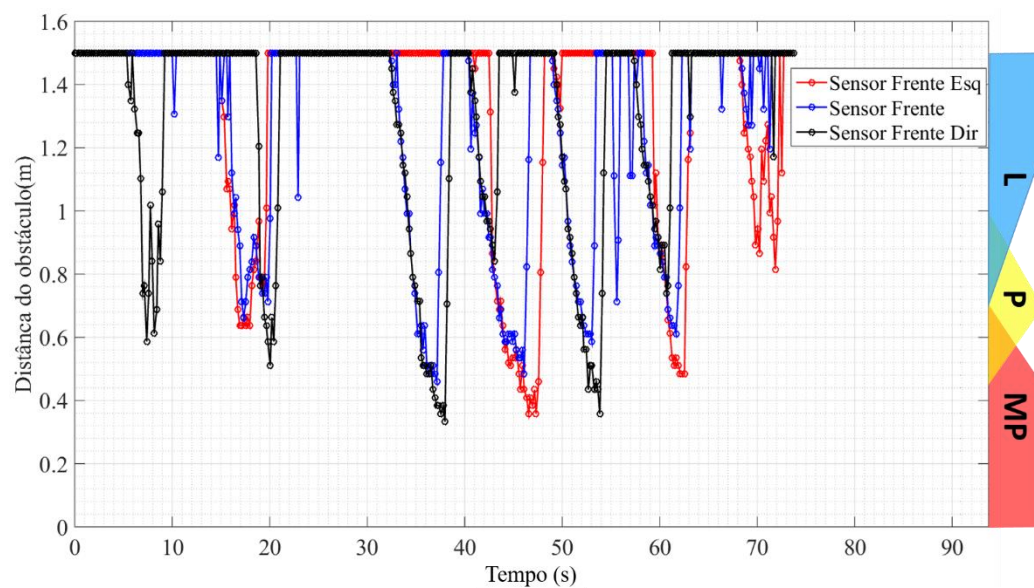


Figura 152 - Dados dos sensores de distâncias frontais durante a realização do percurso 1

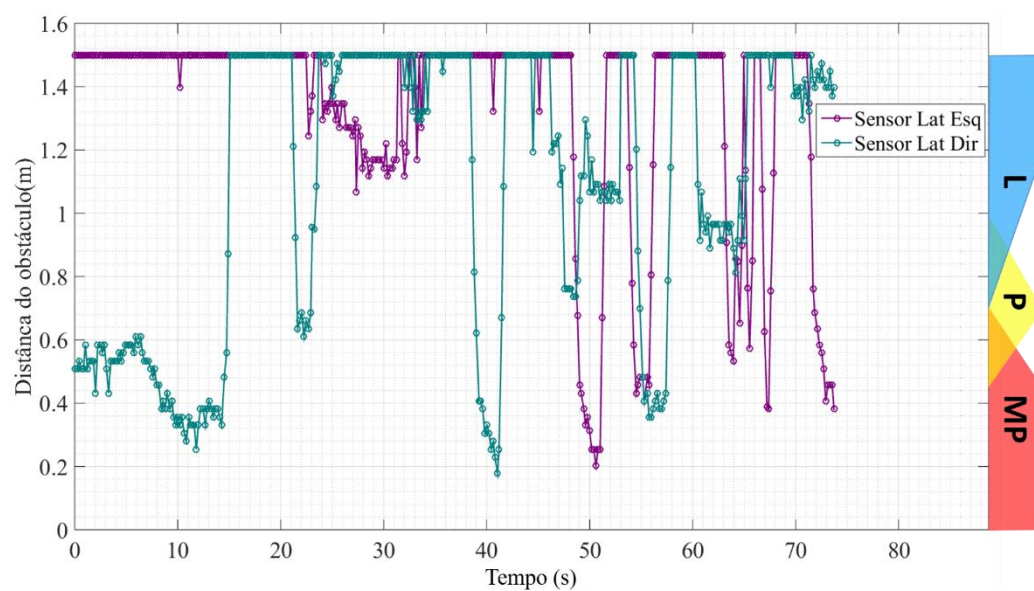


Figura 153 - Dados dos sensores de distâncias laterais durante a realização do percurso 1

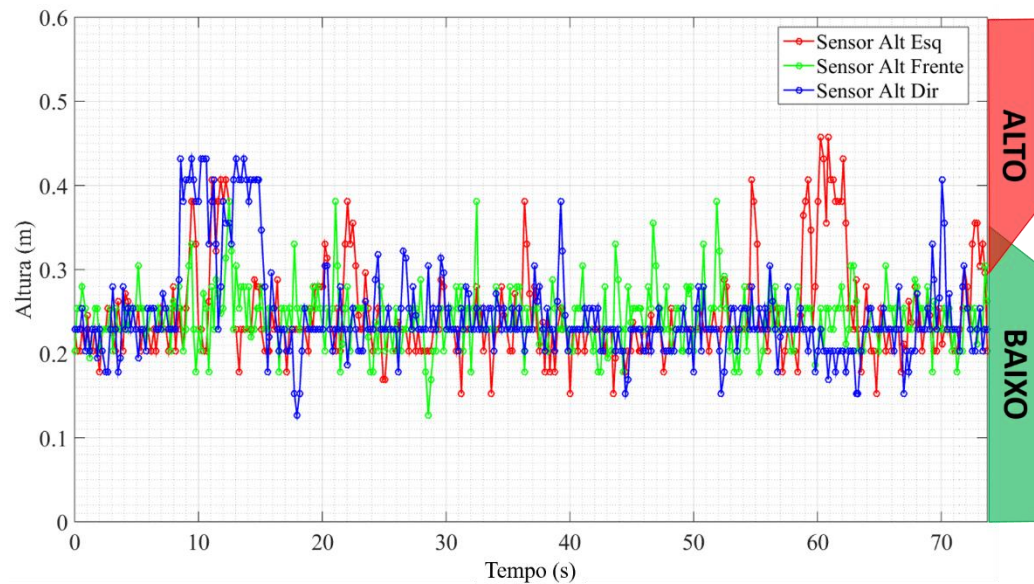


Figura 154 - Dados dos sensores de altura durante a realização do percurso 1

A execução desse primeiro percurso ocorreu de maneira suave e sem grandes oscilações ou desvios do objetivo, sendo o robô capaz de executar todo o trajeto sem apresentar colisões. Durante o trajeto o robô apresentou uma pequena parada de menos de um segundo, devido a um ruído na leitura do *Sensor Altura Frente* (instante  $t=10$  s da Figura 154), que o levou para a condição de desnível na frente e na lateral. Nessa condição (Figura 155), conforme ocorrido no trajeto 1 do ambiente virtual, o SIF apresenta saída nula (Figura 150 instante  $t = 10$  s), pois o sistema não pode oferecer a segurança necessária para a mudança de configuração. Contudo, por ser apenas um ruído isolado, o robô, após a breve parada, segue até o fim de seu trajeto, terminado conforme a Figura 156.



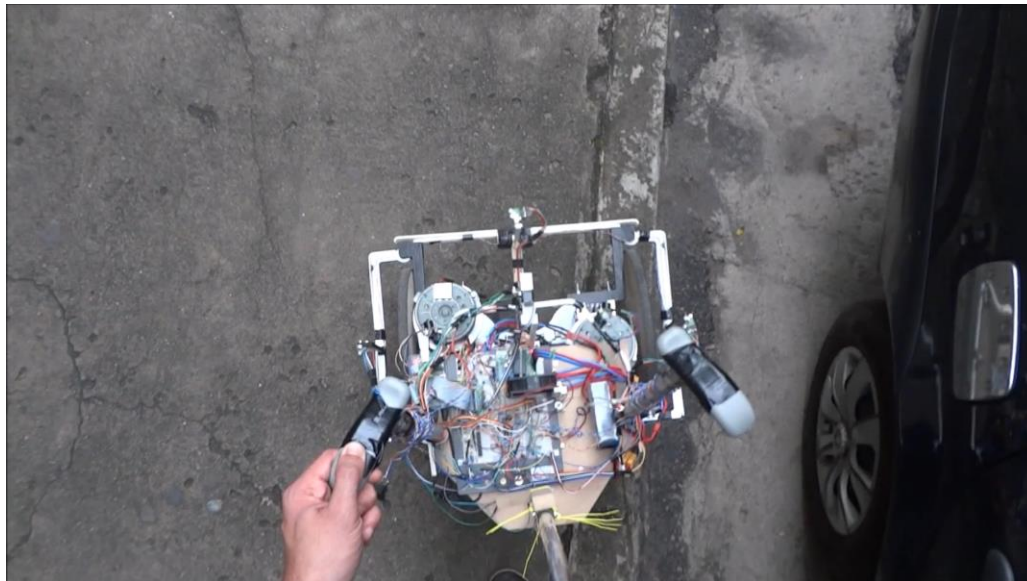


Figura 155 - Posição de parada devido a ruído do sensor de altura frente

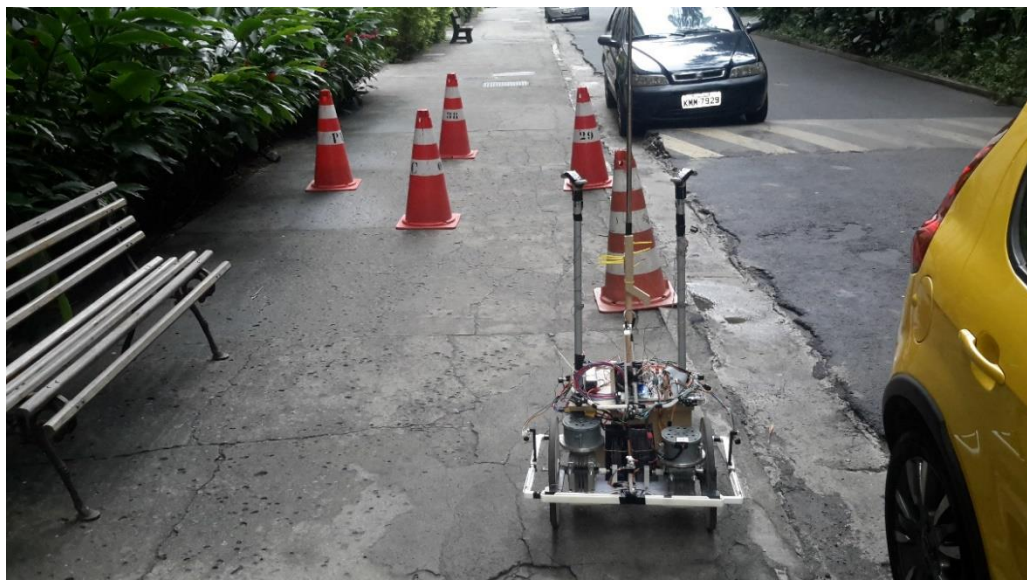


Figura 156 - Posição final do percurso 1

### 5.2.2. Percurso 2

O segundo percurso (Figura 157) inicia de uma posição próxima ao término do percurso 1. No percurso 2 a configuração inicial do robô faz um ângulo de  $-90$  graus com o objetivo, ou seja, a direção paralela à calçada no sentido do fundo da Figura 157.

Na Figura 157, assim como na Figura 149, está desenhada em vermelho uma aproximação do trajeto realizado pelo robô durante esse percurso de teste.



Figura 157 - Posição inicial do percurso 2

O robô, assim como no trajeto anterior, realiza sua navegação de maneira suave em direção ao objeto até que no instante  $t = 37,5$  s, após desviar do último cone, o robô se direciona para o meio-fio, atingindo uma condição não segura (Figura 158, Figura 159 e Figura 164). Essa condição, caracterizada por um desnível na parte da frente e em uma de suas laterais, tem como saída no SIF um valor nulo (Figura 164 instante  $t = 37,5$  s), exigindo uma atuação manual do usuário nos motores. Nesse momento o teste é encerrado e o robô é reposicionado com a atuação manual. Os dados dos sensores e atuadores durante a realização desse percurso podem ser observados na Figura 160 a Figura 164.



Figura 158 - Condição de parada do percurso 2 devido a configuração de risco

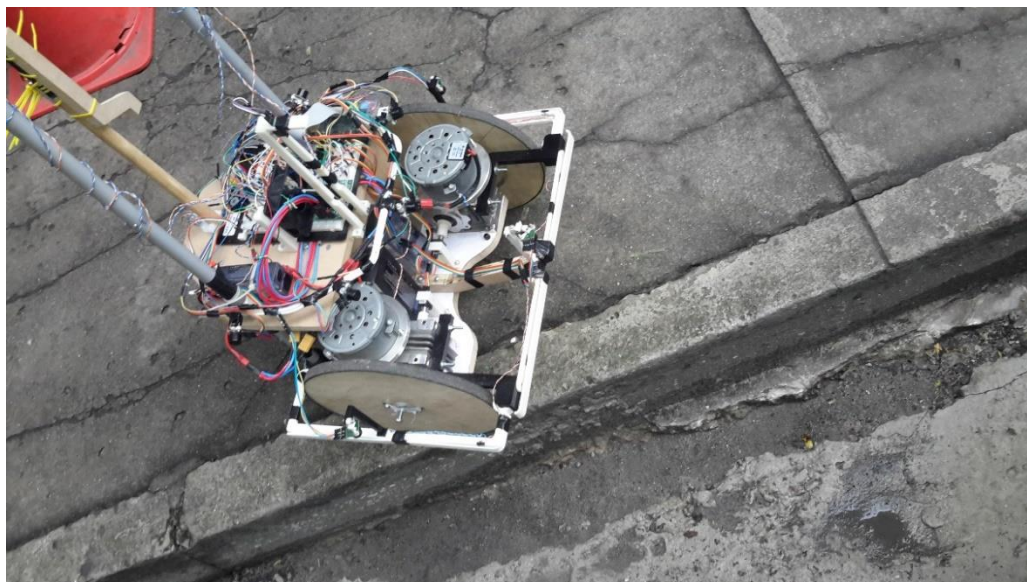


Figura 159 - Condição de parada do percurso 2 devido a configuração de risco, em detalhes



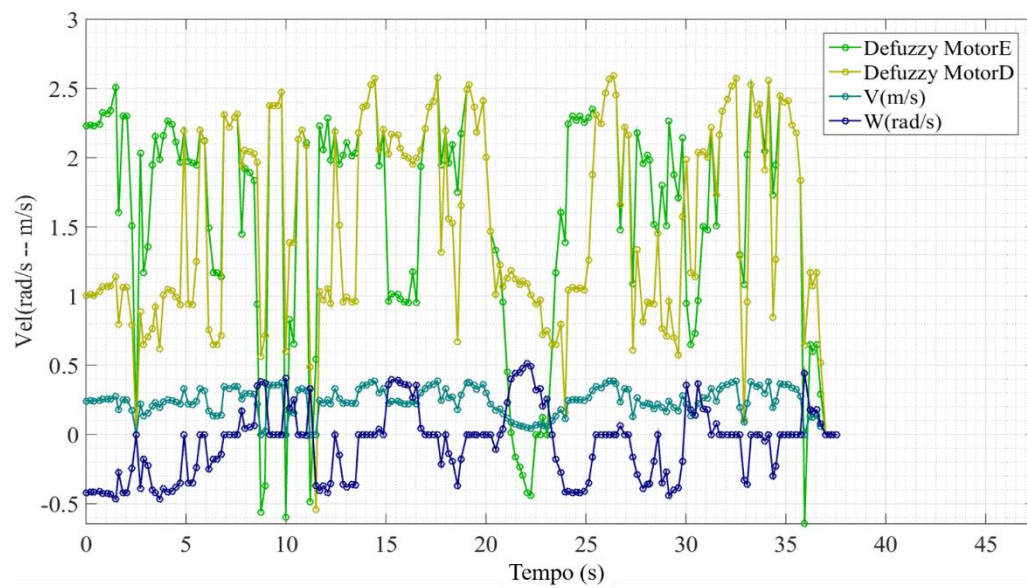


Figura 160 - Velocidades aplicadas durante a realização do percurso 2

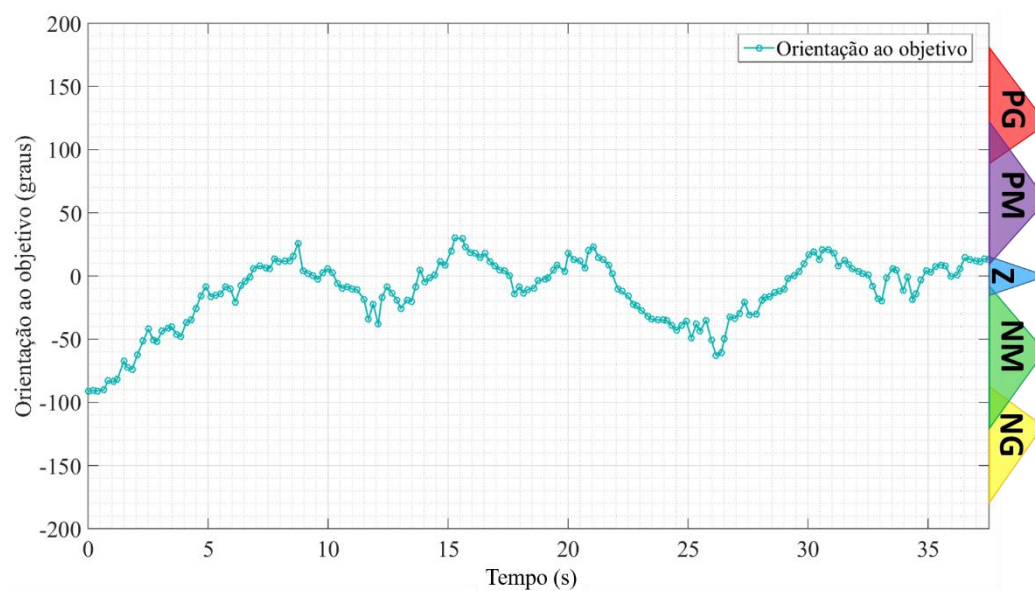


Figura 161 - Dados de orientação ao objetivo durante a realização do percurso 2

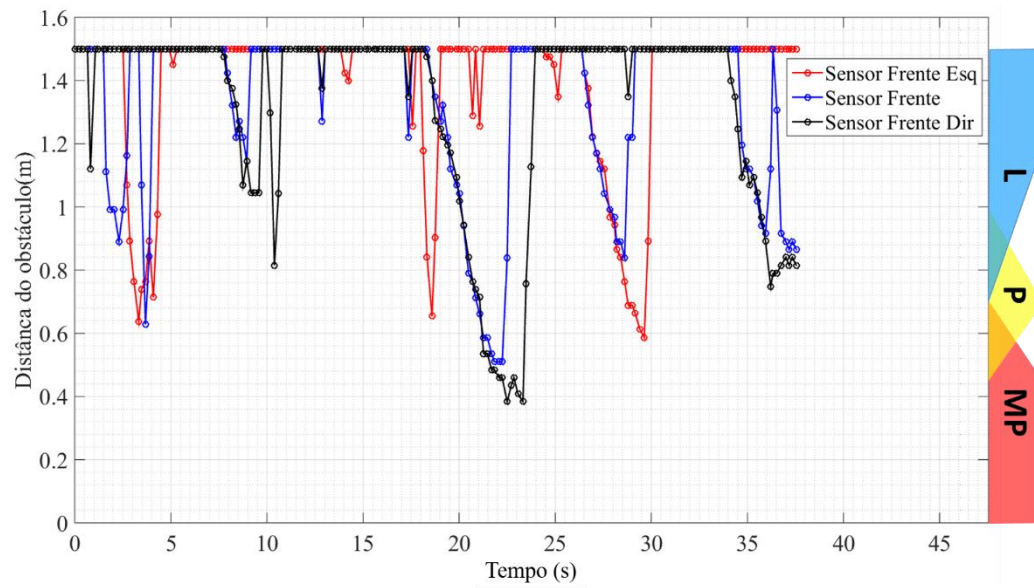


Figura 162 - Dados dos sensores de distâncias frontais durante a realização do percurso 2

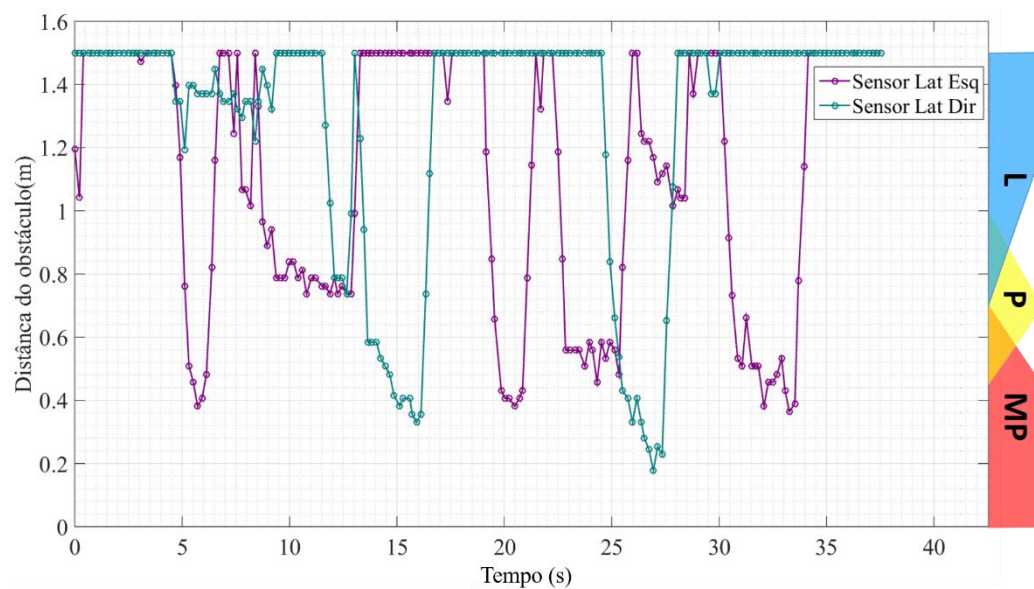


Figura 163 - Dados dos sensores de distâncias laterais durante a realização do percurso 2



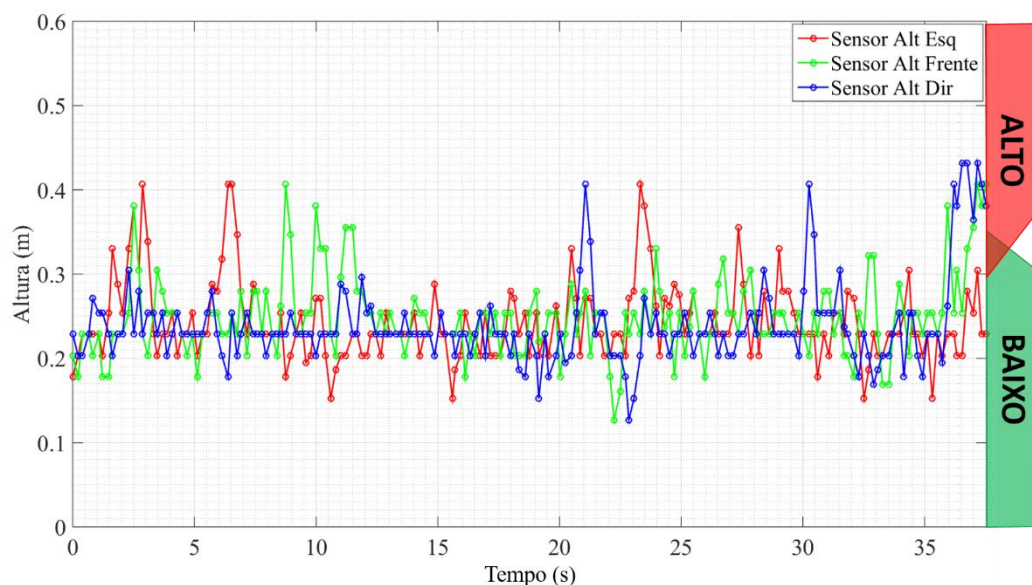


Figura 164 - Dados dos sensores de altura durante a realização do percurso 2

### 5.2.3. Percurso 3

O terceiro e último percurso possui o mesmo objetivo do percurso 1, contudo, nessa configuração o robô inicia alinhado com o objetivo e um obstáculo, conforme Figura 165. Assim como nos percursos anteriores, a linha vermelha desenhada na Figura 165 representa uma aproximação do trajeto realizado durante o percurso de teste. Na Figura 166 a Figura 170 são apresentados os dados dos sensores e atuadores durante a execução do percurso 3.

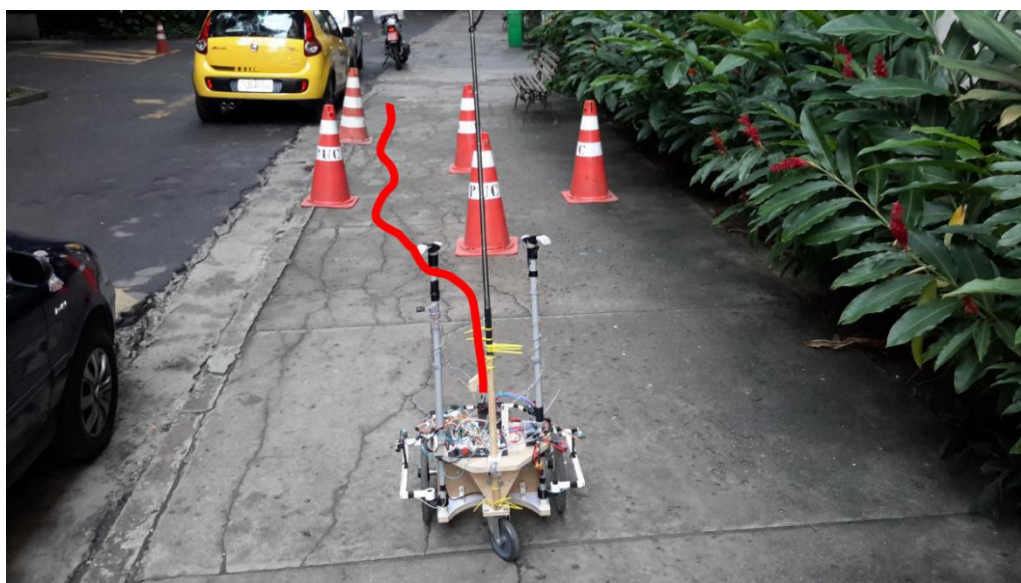


Figura 165 - Início do percurso 3

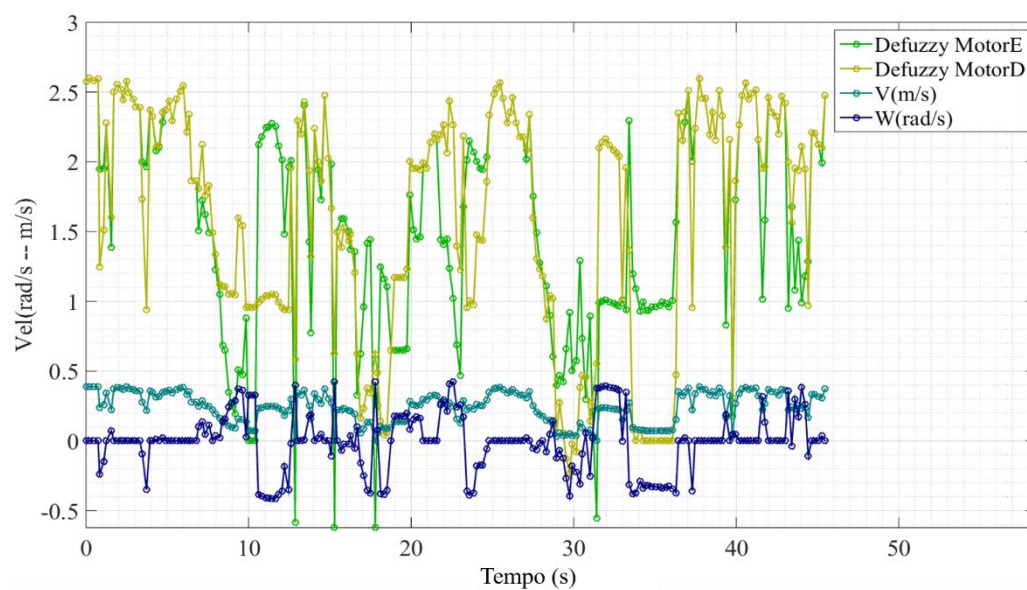


Figura 166 - Velocidades aplicadas durante a realização do percurso 3

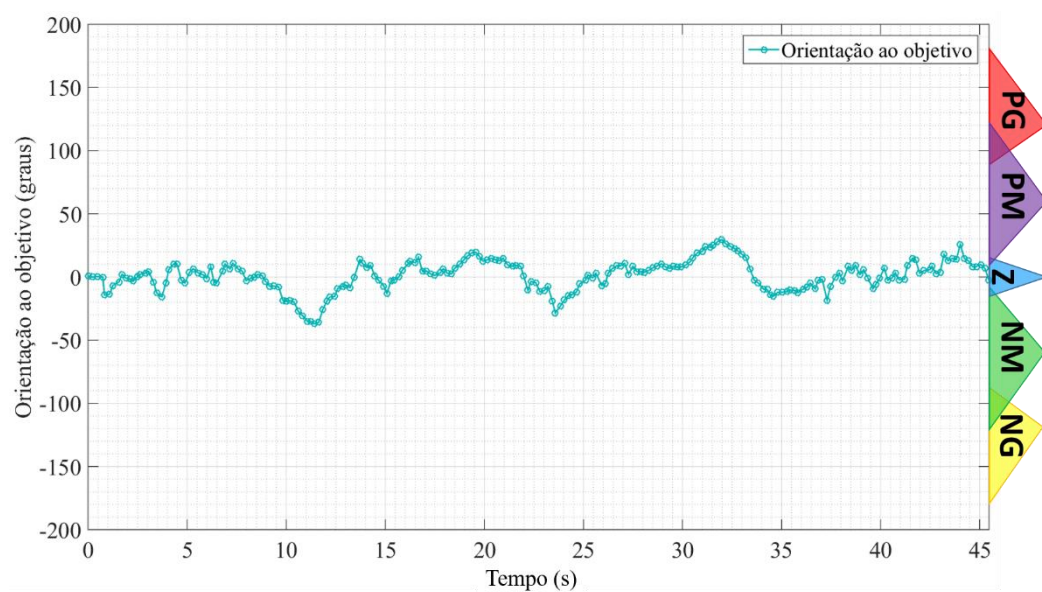


Figura 167 - Dados de orientação ao objetivo durante a realização do percurso 3

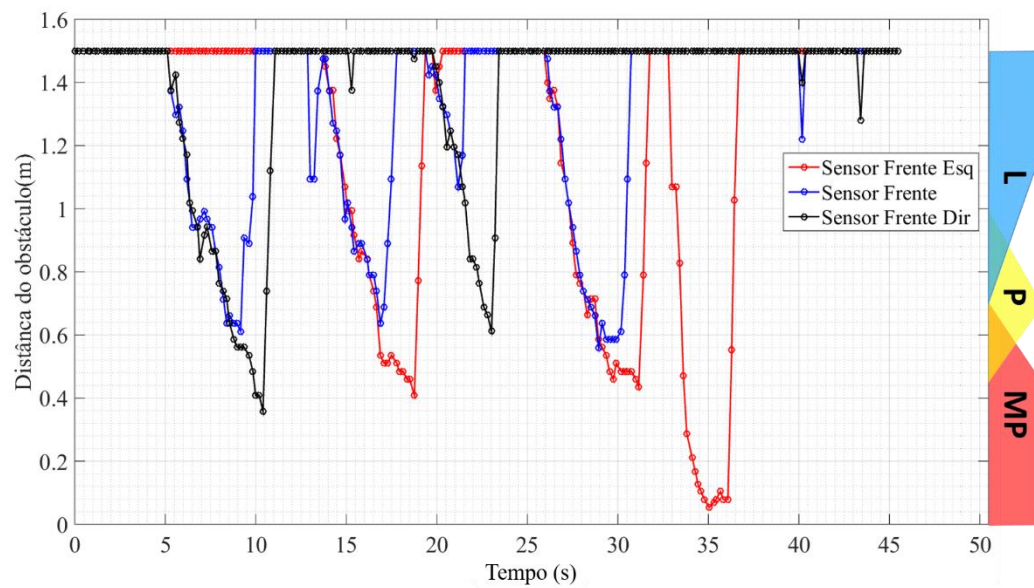


Figura 168 - Dados dos sensores de distâncias frontais durante a realização do percurso 3

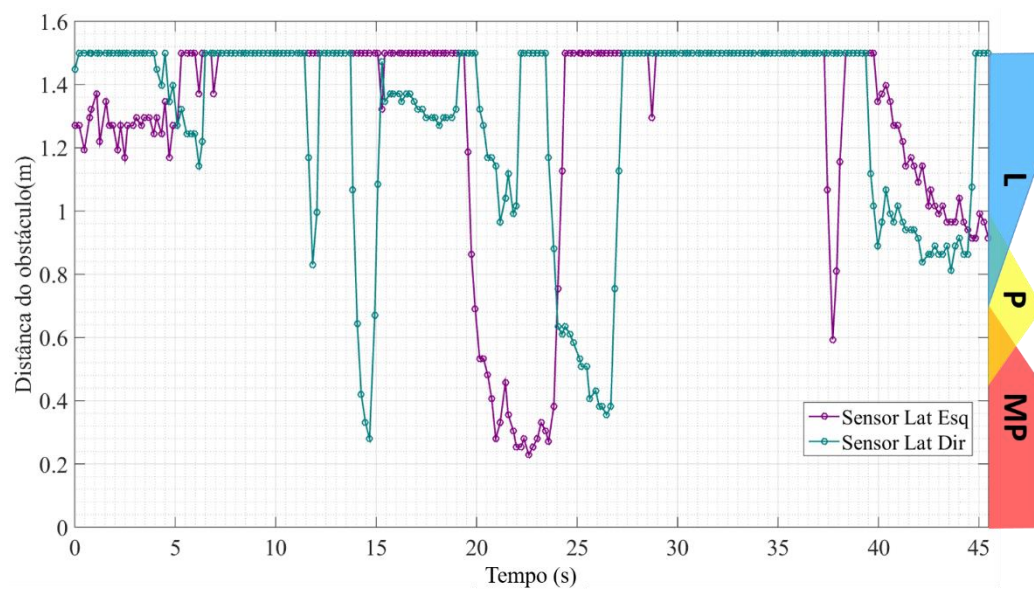


Figura 169 - Dados dos sensores de distâncias laterais durante a realização do percurso 3



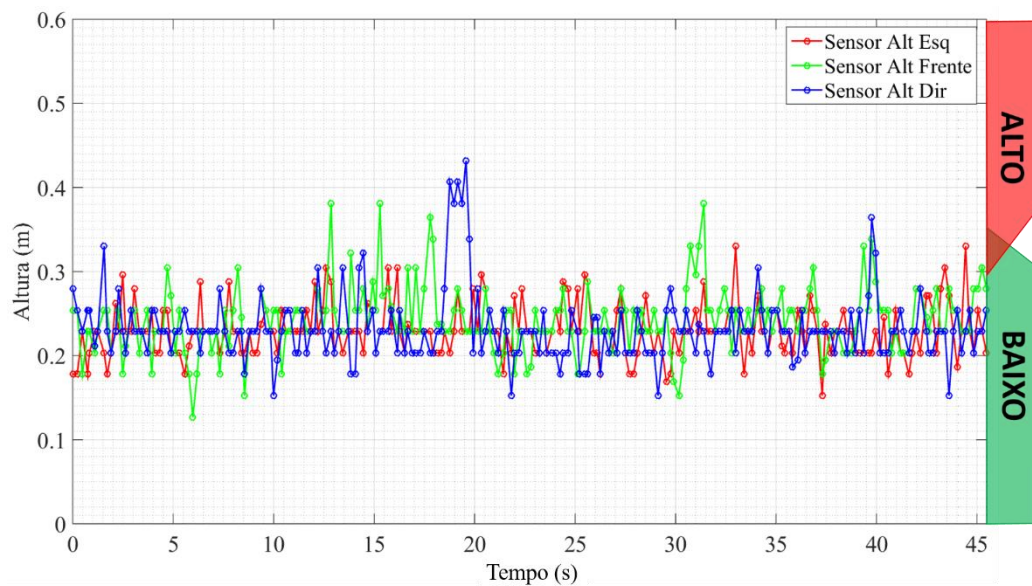


Figura 170 - Dados dos sensores de altura durante a realização do percurso 3

No início desse percurso, é possível observar que o robô vai se aproximando do obstáculo em prol de manter-se alinhado com o objetivo até que entra na configuração de distância ao obstáculo *Muito Perto* nos sensores de distância *Frente* e *Frente Direita* (instante  $t = 8$  s da Figura 168). No momento em que esses sensores de começam a entrar na região Fuzzy *Muito Perto*, o comportamento do robô passa a ser desviar do obstáculo ( $t > 8$  s e  $t < 14$  s na Figura 166). Esse comportamento é esperado, pois como o robô já estava alinhado com o objetivo, a tendência de SIF é ir reduzindo sua velocidade até alcançar a configuração de desvio.

Outro ponto a ser observado nesse percurso é o intervalo de tempo entre os instantes  $t = 34$  s e  $t = 37$  s. Nesse intervalo o robô passa bem próximo de um obstáculo em sua lateral esquerda. Essa aproximação ocorreu, pois, ao desviar de um dos obstáculos, o cone seguinte não está na região de atuação dos sensores, sendo ele percebido apenas quando este já está muito próximo ao robô (Figura 171). Apesar de passar bem próximo ao obstáculo, o robô consegue contorná-lo, terminando assim seu trajeto.

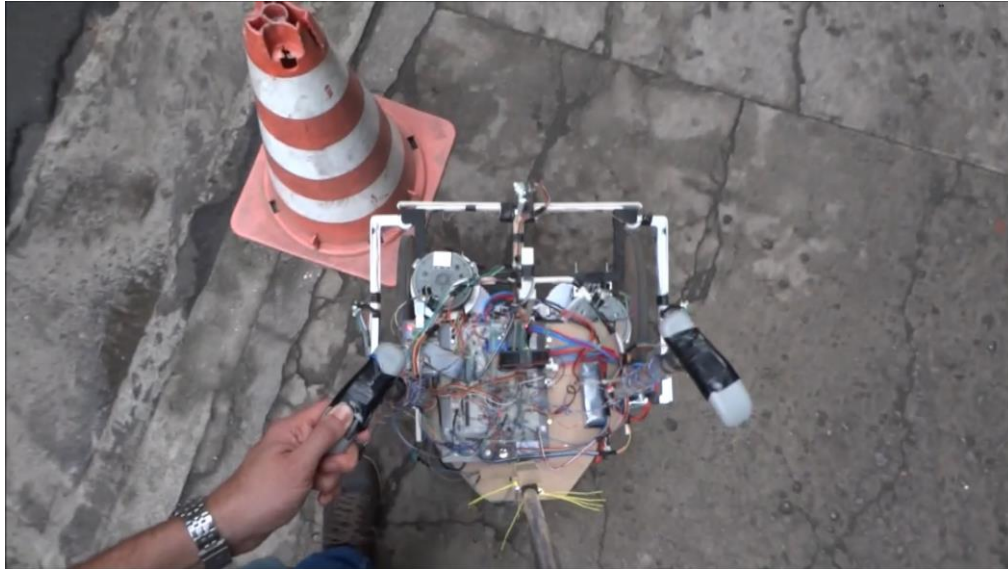


Figura 171 - Instante  $t = 34s$  do percurso 3

Assim como nos trajetos anteriores, esse percurso foi executado com manobras suaves e sempre direcionado ao objetivo. A posição do robô ao terminar o percurso 3 pode ser observada na Figura 172.



Figura 172 - Posição final do percurso 3

Com a conclusão dos testes do modelo real e virtual, pode-se concluir que o robô consegue realizar o objetivo proposto, sendo capaz de executar a navegação com a segurança necessária para a tarefa proposta. Na comparação entre os testes do modelo real e virtual, eles apresentaram respostas parecidas, sendo que, nas configurações em que o robô possuía pouco espaço de manobra, o modelo real

apresentou menos oscilações para conseguir desviar do obstáculo. Essa melhor resposta é justificada pela inclusão do regulador externo no controle fuzzy do modelo real, apresentado na seção 4.3.2.

Com o término da apresentação dos testes e resultados, a seguir, no capítulo 6, serão apresentadas a conclusão e as indicações para trabalhos futuros.

## 6 Conclusão e trabalhos futuros

### 6.1 Conclusão

Conforme apresentado no capítulo 1, essa dissertação de mestrado teve como objetivo o desenvolvimento do protótipo de um dispositivo de assistência à locomoção para pessoas idosas. O protótipo criado, embarcado com sensores, atuadores e hardware processador, foi capaz de se locomover de maneira autônoma em um ambiente desconhecido, desviando dos obstáculos de maneira suave e sempre direcionado ao objetivo, fornecido pelo aplicativo de navegação.

Ao se deparar com uma configuração em que o conjunto de informações fornecidas pelos sensores não é suficiente para garantir que o SIF atue de maneira segura para o usuário, o SIF responde com uma atuação nula. Portanto, o SIF implementado apresentou a robustez desejada para conduzir o usuário de maneira segura e, consequentemente, as funcionalidades de assistência a locomoção, desvio de obstáculos e posicionamento, apresentadas no capítulo 3, foram atendidas. Além da navegação autônoma, o dispositivo possui mecanismos para atuação direta do usuário, que sempre tem prioridade sobre a automação do dispositivo. A atuação direta é essencial para a locomoção em ambientes fechados ou sem rotas de navegação. Outra característica do deslocamento do protótipo é que ele só é habilitado, seja de maneira autônoma ou manual, enquanto o usuário segura a pega da bengala.

Em relação à funcionalidade de prevenção de quedas, conforme apresentado no capítulo 3, ela foi tratada de maneira passiva, ou seja, apenas pelo apoio fornecido pela estrutura. Contudo, apesar desse caráter passivo adotado para essa funcionalidade, devido ao fato do dispositivo conseguir identificar e evitar potenciais fontes de desequilíbrio e/ou queda, como buracos, desníveis e obstáculos, a contribuição para a funcionalidade de prevenção de quedas foi muito além de um simples apoio fornecido por uma estrutura passiva.

Com relação aos objetivos secundários, apresentados no capítulo 1, o protótipo desenvolvido também obteve êxito. A aplicação da lógica Fuzzy, como ferramenta para a criação do algoritmo de navegação, se mostrou uma escolha adequada. Por meio da lógica Fuzzy foi possível criar um modelo de fácil interpretação, devido ao conjunto de regras obtido, cuja robustez foi capaz de orientar e conduzir o usuário de maneira segura, realizando manobras suaves em um ambiente desconhecido e ao auxílio de uma tabela com filtros, que facilitou a apuração do sistema para diferentes configurações de suas entradas e saídas Fuzzy. Todo o hardware utilizado foi integrado com o framework de robótica Player e, portando, suas informações podem ser acessadas por qualquer programa cliente que esteja na mesma rede do protótipo. E, por último, os tipos e modelos de sensores, atuadores e hardware processador utilizados conseguiram atender às demandas do protótipo. Em relação a esse último item, cabe ressaltar que a utilização dos sensores de ultrassom para identificar os desníveis não se mostrou a escolha mais adequada. O problema observado na utilização desse tipo de sensor é o fato dele não permitir realizar leituras muito afastadas do dispositivo, devido ao seu posicionamento, fazendo com que a identificação do desnível não ocorra com a antecedência ideal. Essa característica fez com que o dispositivo precisasse chegar bem próximo ao meio-fio para conseguir detectá-lo.

Por último, essa dissertação teve como grande desafio a multidisciplinaridade da tarefa proposta. A construção do dispositivo envolveu diversas áreas do conhecimento como: eletrônica, computação e mecânica, além dos trabalhos realizados por Rivero (2015), com as pesquisas de funcionalidade e condições de contorno, e a construção da estrutura para teste, realizada por terceiros. Algumas das etapas do desenvolvimento dessa dissertação que exemplificam alguns dos desafios encontrados estão listadas a seguir:

- *Especificação do hardware:* foi necessário realizar um estudo sobre diversos tipos de sensores, atuadores e hardware processadores com o intuito de especificar os tipos e modelos mais adequados, além do embasamento necessário para realizar a calibração, tratamento e leitura/atuação de cada um deles.
- *Projeto eletrônico:* para cada um dos diferentes tipos de hardware utilizados, foi necessária a montagem de circuitos eletrônicos para a integração com o hardware processador.



- *Ferramentas de software:* foram criados diversos programas, *drivers*, para integrar todo o hardware utilizado com o servidor Player, além da criação do ambiente virtual para testes, por meio do simulador de robôs Stage. A complexidade dessa etapa é destacada pela necessidade de um conhecimento mais aprofundado da ferramenta Player, para a criação dos *drivers*, além de toda a programação envolvida para o acesso dos diferentes tipos de hardware utilizados e por diferentes protocolos de comunicação como, por exemplo, I2C, SPI, TCP/IP e PWM, e da integração dessas ferramentas com o sistema operacional Linux.
- *Algoritmos de controle:* A criação do algoritmo de navegação, baseado em lógica Fuzzy, capaz de orientar e guiar o usuário de maneira segura, com conversões suaves e poucas oscilações, além da programação da biblioteca Fuzzy e da implementação de um controle PID para regular a velocidade angular de cada motor.

Apesar do trabalho desenvolvido nessa dissertação ser apenas um protótipo inicial, com base nos resultados apresentados pode-se concluir que essa dissertação obteve grande êxito na tarefa proposta. Além dos resultados obtidos nos testes, todos os objetivos listados no capítulo 1 foram atendidos.

A seguir, na seção 6.2, serão apresentadas algumas propostas de melhorias e pontos a serem desenvolvidos para trabalhos futuros.

## **6.2**

### **Trabalhos futuros**

Em relação aos trabalhos futuros, o protótipo possui alguns pontos para serem desenvolvidos e melhorados. Com relação aos pontos a serem desenvolvidos, pode-se destacar a parte de design, estrutura, prevenção de quedas e identificação de falhas de hardware e/ou software. Já quanto a pontos a serem melhorados, pode-se destacar o refinamento das regras, a atualização do hardware processador e a mudança nos sensores de altura.

Para a melhoria do design, deve-se realizar um estudo mais aprofundado da ergonomia do dispositivo para a tarefa proposta, tendo um desenho final mais

adequado para interação com usuários idosos. Com o desenho estabelecido, o próximo passo seria o estudo dos materiais mais adequados para a construção do protótipo, de maneira que a estrutura final consiga conjugar leveza e resistência.

Para a prevenção de quedas, propõe-se a aplicação de um controle ativo no qual apenas a haste da bengala, por meio de atuadores ligados à sua base, atue de modo a reestabelecer a postura de equilíbrio do usuário, conforme Di et al. (2012), por exemplo.

Para aumentar a robustez do sistema, principalmente devido à fragilidade do público alvo, propõe-se a implementação de um sistema para monitorar possíveis falhas nos sensores, atuadores e/ou software. Esse sistema pode ser implementado com a adição e combinação de sensores redundantes, para estimar a falha de um dado sensor ou hardware, por exemplo. Atualmente a redundância implementada apenas monitora se o servidor Player e o programa cliente, responsável por operar o dispositivo robótico, estão sendo executados, portando está mais relacionada a uma redundância de software.

Para o refinamento do SIF, propõe-se a remoção de regras redundantes, tornando o SIF mais simples e rápido. Muitas das regras atualmente empregadas já foram identificadas como sendo redundantes, contudo elas não foram removidas para facilitar a análise das transições das regras durante as etapas de teste. Outra proposta, com relação ao refinamento do conjunto de regras, seria a avaliação de uma abordagem hierárquica para a reorganização desse conjunto.

Para melhoria do hardware processador, propõe-se a atualização do Raspberry Pi para uma versão mais atual, tendo como vantagens maior capacidade de processamento e número de portas de entrada e saída, o que diminuiria a necessidade de chips de expansão. Com o aumento da capacidade de processamento seria interessante avaliar a viabilidade de adicionar um sensor de visão para contribuir com as informações já existentes.

Para melhorar a identificação dos desníveis, propõe-se a utilização de sensores infravermelho do tipo PSD juntamente com os sensores de ultrassom. A combinação da informação desses sensores de diferentes naturezas tem como objetivo reduzir as limitações existentes em cada sensor. O sensor de ultrassom operando sozinho se mostrou uma escolha não adequada para essa tarefa, pois não permite um horizonte de medição muito grande. Esse fato ocorre porque, ao se inclinar o sensor para aumentar seu alcance, uma grande quantidade de ruído é

inserida, diminuindo sua exatidão. Já os sensores infravermelhos são sensíveis a ruídos gerados pela incidência do sol ou a baixa reflexibilidade da região de contato. Contudo, a fusão dos dados desses sensores, de diferentes naturezas, proporcionaria uma melhor estimativa da informação de desnível.

Por último, com a atualização da parte estrutural e do sistema de monitoração de falhas, tem-se como trabalhos futuros a realização de testes reais com usuários idosos, a fim de evoluir o protótipo a partir desses novos resultados.

## 7

## Referências bibliográficas

A MEMSNET. What is MEMS Technology? **MEMSnet**. Disponível em: <[http://www.memsnet.org/mems/what\\_is.html](http://www.memsnet.org/mems/what_is.html)>. Acesso em: 01 17 2016.

ALMEIDA, ; ZHANG, M.; LIU, J.-C. Dynamic Fall Detection and Pace Measurement in Walking Sticks. **High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP. Joint Workshop on**, 25-27 jun. 2007. 204-206.

AMERICANAS. Bengala de Alumínio - Dilepe. **Americanas**, 2016. Disponível em: <[http://www.americanas.com.br/produto/5993940/bengala-de-aluminio-dilepe?opn=YSMESP&loja=02&WT.srch=1&epar=bp\\_pl\\_00\\_go\\_pla-bs-todas#informacoes-tecnicas](http://www.americanas.com.br/produto/5993940/bengala-de-aluminio-dilepe?opn=YSMESP&loja=02&WT.srch=1&epar=bp_pl_00_go_pla-bs-todas#informacoes-tecnicas)>. Acesso em: 10 fev. 2019.

ANDRADE, A. S. D. Notas de Aula - Elementos Orgânicos de Máquinas II. **Universidade Federal do Paraná**. Disponível em: <<http://www.madeira.ufpr.br/disciplinasalan/AT102-Aula04.pdf>>. Acesso em: 2015 nov. 19.

ANDREONI, G. et al. Wearable Monitoring Devices for Assistive Technology: Case Studies in Post-Polio Syndrome. **Sensors**, v. 2, p. 2012-2027, 24 jan. 2014. ISSN ISSN 1424-8220.

ARAÚJO, F. M. U. D. Controle Inteligente. **Departamento de Engenharia de Computação e Automação - UFRN**, Natal, p. 75, nov. 2004. Disponível em: <<http://www.dca.ufrn.br/~meneghet/FTP/IA/contrint.pdf>>. Acesso em: 10 jan. 2016.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 9050 - Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos**. ABNT. Rio de Janeiro, p. 148. 2015. (ISBN 978-85-07-05706-2).

BANANA PI. Banana Pi - A Highend Single-Board Computer. **Banana Pi**, 2014. Disponível em: <<http://www.bananapi.org/p/product.html>>. Acesso em: 10 dez. 2015.

BANCROFT, J. B.; LACHAPPELLE, G. Data Fusion Algorithms for Multiple Inertial Measurement Units. **Sensors**, v. 11, n. 12, p. 6771–6798, jun. 2011. Disponível em: <<http://dx.doi.org/10.3390/s110706771>>.

BEAGLEBOARD. **BeagleBone Black**. Disponível em: <<http://beagleboard.org/black>>. Acesso em: 05 dez. 2014.

BIASI, S. C. D.; GATTASS, M. **Utilização de quatérnios para representação de rotações em 3D**. Instituto Tecgraf/PUC-Rio. Rio de Janeiro, p. 35. 2002.

BORENSTEIN, J.; EVERETT, H. R.; FENG, L. **Where am I? Sensors and Methods for Mobile Robot Positioning**. 1ª. ed. Ann Arbor: Universidade de Michigan, 1996.

BORENSTEIN, J.; ULRICH, I. The GuideCane — A Computerized Travel Aid for the Active Guidance of Blind Pedestrians. **Proceedings of the IEEE International Conference on Robotics and Automation**, 21-27 Abril 1997. 1283-1288.

BOURNS. Datasheet Encoder. **Bourns**, dez. 2015. Disponível em: <<https://www.bourns.com/data/global/pdfs/enc1j.pdf>>. Acesso em: 01 jan. 2016.

BROADCOM CORPORATION. **BCM2835 ARM Peripherals**. Broadcom. Cambridge, p. 205. 2012.

BROWN, K. **Reflections on Relativity**. Softcover. ed. [S.l.]: MathPages, 2015.

CARON, F. et al. GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. **Information Fusion**, 2004. 221-230.

CONDIT, R. Brushed DC Motor Fundamentals - AN905. **Microchip**, p. 10, 26 jan. 2004. Disponível em: <<http://ww1.microchip.com/downloads/en/appnotes/00905a.pdf>>. Acesso em: 01 dez. 2015.

COSTA, F. N. D. Cidadania & Cultura. **Fernando Nogueiracosta**, 16 dez. 2010. Disponível em: <<https://fernandonogueiracosta.wordpress.com/2010/12/16/piramide-etaria-brasileira/>>. Acesso em: 23 set. 2015.

CUBIEBOARD. Cubieboard Docs. **CubieBoard**. Disponível em: <<http://docs.cubieboard.org/tutorials/cubietruck/start>>. Acesso em: 02 dez. 2015.

DANAHER INDUSTRIAL CONTROLS. Danaher Handbook - Encoder Aplicatoin Handbook. **Dynapar**, 2003. Disponível em: <[https://www.dynapar.com/uploadedFiles/Products/Danaher\\_Encoder\\_Handbook.pdf](https://www.dynapar.com/uploadedFiles/Products/Danaher_Encoder_Handbook.pdf)>. Acesso em: 11 jan. 2016.

DHAOUADI, R.; HATAB, A. A. Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework. **Advances in Robotics & Automation**, 2013.

DI, P. et al. Optimal posture control for stability of intelligent cane robot. **RO-MAN, 2012 IEEE**, 9-13 Set 2012. 725 - 730.

DPT LABORATORIES. What is Quality by Design (QbD) – And why should your care? **DPT LABs**, San Antonio, p. 5, 2013. Disponível em: <<http://www.dptlabs.com/wp-content/uploads/2013/05/What-is-Quality-by-Design-QbD-and-Why-Should-You-Care.pdf>>. Acesso em: 16 set. 2014.

DUBOWSKY, S. et al. PAMM - a robotic aid to the elderly for mobility assistance and monitoring: a “helping-hand” for the elderly. **Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on (Volume:1)**, 24-28 Abril 2000. 570 - 576 vol.1.

DULLERUD, G. E.; PAGANINI, F. **A Course in Robust Control Theory: A Convex Approach**. 1ª. ed. [S.l.]: Springer, 2005.

EATON, J. W. GNU Octave. **GNU**, 2016. Disponível em: <[https://www.gnu.org/software/octave/John W. Eaton](https://www.gnu.org/software/octave/John%20W.%20Eaton)>. Acesso em: 05 fev. 2016.

EITEL, E. Basics of Rotary Encoders: Overview and New Technologies. **Machine Design**, 07 maio 2014. Disponível em: <<http://machinedesign.com/sensors/basics-rotary-encoders-overview-and-new-technologies-0>>. Acesso em: 01 dez. 2015.

ELECTRICAL4U. Types of DC Motor Separately Excited Shunt Series Compound DC Motor. **Electrical4u - ONLINE ELECTRICAL ENGINEERING STUDY SITE**. Disponível em: <<http://www.electrical4u.com/types-of-dc-motor-separately-excited-shunt-series-compound-dc-motor/>>. Acesso em: 19 nov. 2015.

FAIRCHILD SEMICONDUCTOR. Datasheet: LM78XX, LM78XXA - 3-Terminal 1 A Positive Voltage Regulator. **FairchildSemi**, set. 2014. Disponível em: <<https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf>>. Acesso em: 01 fev. 2016.

FAPESP. Uma política para o bem-envelhecer. **Pesquisa FAPESP**, São Paulo, n. 145, p. 14-19, Março 2008. Disponível em: <<http://revistapesquisa.fapesp.br/wp-content/uploads/2008/03/14-19-Entrevista-145.pdf?e3082f>>. Acesso em: 2015.

FIGUEIREDO, L. C.; JOTA, F. G. Introdução ao controle de sistemas não-holonômicos. **Sba Controle & Automação**, Campinas, v. 15, n. 3, p. 243-268, set. 2004. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-17592004000300002&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592004000300002&lng=en&nrm=iso)>.

FRADEN, J. **Handbook of Modern Sensor: Physics, Designs, and Applications**. 4º. ed. New York: Springer, 2010.

GERKEY, B. P.; VAUGHAN, R. T.; HOWARD, A. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. **In Proceedings of the International Conference on Advanced Robotics (ICAR 2003)**, Coimbra, 30-03 06-07 2003. 317-323.

GERKEY, B.; CONTRIBUTORS. The Player Robot Device Interface. **PlayerStage**, 2010. Disponível em: <<http://playerstage.sourceforge.net/doc/Player-3.0.2/player/>>. Acesso em: 01 dez. 2014.

GERKEY, B.; STOY, K.; CONTRIBUTORS. The Player Robot Device Interface. **PlayerStage**, 2010. Disponível em: <<http://playerstage.sourceforge.net/doc/Player-3.0.2/player/>>. Acesso em: 01 dez. 2014.

GRAY, F. **Pulse code communication**. US2632058 A, 17 mar. 1953.

HANSELMAN, D. D. **Brushless Permanent Magnet Motor Design**. 2<sup>a</sup>. ed. Lebanon: Magna Physics Publishing, 2006.

HARDKERNEL. ODROID-U3 - Technical Detail. **Hardkernel**. Disponível em: <[http://www.hardkernel.com/main/products/prdt\\_info.php?g\\_code=G138745696275&tab\\_idx=2](http://www.hardkernel.com/main/products/prdt_info.php?g_code=G138745696275&tab_idx=2)>. Acesso em: 01 dez. 2015.

HENDERSONS, G. Wiring Pi - GPIO Interface library for the Raspberry Pi. **Wiringpi**, 2015. Disponível em: <<http://wiringpi.com/>>. Acesso em: 10 fev. 2016.

HIRATA, Y.; KOMATSUDA, S.; KOSUGE, K. Fall Prevention Control of Passive Intelligent Walker Based on Human Model. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, 22-26 Set 2008. 1222 - 1228.

HUANG, J. et al. Motion control of omni-directional type cane robot based on human intention. **Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on**, 22-26 set. 2008. 273-278.

IBGE. Sala de Imprensa. **IBGE**, 2008. Disponível em: <<http://saladeimprensa.ibge.gov.br/noticias?view=noticia&id=1&idnoticia=1272&busca=1&t=ibge-populacao-brasileira-envelhece-ritmo-acelerado>>. Acesso em: 23 set. 2015.

IBGE. Projeção da população por sexo e idade: Brasil 2000-2060. **IBGE**, 29 ago. 2013. Disponível em: <<http://www.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000014425608112013563329137649.pdf>>. Acesso em: 23 set. 2015.

INTERLINK ELECTRONICS. FSR 400. **Interlink Electronics**, 2015. Disponível em: <<http://www.interlinkelectronics.com/FSR400.php>>. Acesso em: 20 fev. 2016.

JAVACLIENT. Javacient Player/Stage. **Source Forge**, 2010. Disponível em: <<http://java-player.sourceforge.net/index.php>>. Acesso em: 01 fev. 2016.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the ASME--Journal of Basic Engineering**, v. 82, n. Series D, p. 35-45, 1960.

KARIM NICE, T. V. W. G. G. How Digital Cameras Work. **How Stuff Works**. Disponível em: <<http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera2.htm>>. Acesso em: 17 dez. 2015.

KIONIX. AN 006 - Handheld Electronic Compass Applications Using a Kionix MEMS Tri-Axis Accelerometer. **Kionix**, Ithaca, p. 16, 09 jul. 2015. Disponível em: <<http://kionixfs.kionix.com/en/document/AN006%20Handheld%20Electronic%20Compass%20Applications%20Using%20a%20Kionix%20MEMS%20Tri-Axis%20Accelerometer.pdf>>. Acesso em: 25 jan. 2016.

KIONIX. AN 012 - Accelerometer Errors. **Kionix**, Ithaca, p. 10, 09 jul. 2015. Disponível em: <<http://kionixfs.kionix.com/en/document/AN012%20Accelerometer%20Errors.pdf>>. Acesso em: 01 fev. 2016.

KONVALIN, C. Motion/Velocity/Displacement Compensating for Tilt, Hard-Iron, and Soft-Iron Effects. **Sensor Online**, 01 dez. 2009. Disponível em: <<http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475>>. Acesso em: 01 jun. 2015.

KUIPERS, J. B. Quaternions and Rotation Sequences. **Geometry, Integrability and Quantization**, 1-10 set. 2000. 127-143.

LACEY, G.; DAWSON-HOWE, K. M. Personal Adaptive Mobility Aid (PAM-AID) for the Infirm and Elderly Blind. In: MITTAL, V. O., et al. **Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing**. [S.l.]: Springer Berlin Heidelberg, v. 1458, 1996. p. 211-220.

LEHRBAUM, R.; BROWN, E. LinuxGizmos: Top 10 hacker SBCs. **HackerBoards.com**, 2014. Disponível em: <<http://linuxgizmos.com/top-10-hacker-sbcs-survey-results/>>. Acesso em: 10 ago. 2014.

LIONBERGER, R. A. et al. Quality by Design: Concepts for ANDAs. **The AAPS Journal**, v. 10, n. 2, p. 268–276, jun. 2008. ISSN 10.1208/s12248-008-9026-7.

LOUVISON, M. P.; ROSA, T. E. D. C. **Vigilância e prevenção de quedas em idosos**. São Paulo: São Paulo (ESTADO) Secretaria da Saúde, 2010.

MADGWICK, S. O. H. **An efficient orientation filter for inertial and inertial/magnetic sensor arrays**. Universidade de Bristol, Departamento de Engenharia Mecânica. Bristol, p. 32. 2010.

MAGNUSSON, N.; ODENMAN, T. **Improving absolute position estimates of an automotive vehicle using GPS in sensor fusion**. Dissertação de Mestrado, CHALMERS UNIVERSITY OF TECHNOLOGY. Göteborg, p. 72. 2012.



MARQUES, L. S. B.; SAMBAQUI, A. B. K.; DUARTE, J. **Apostila de Máquinas Elétricas**. Instituto Federal de Santa Catarina, Campus Joinvill. Joinville, p. 63. 2013.

MARTINS, M. M. et al. Assistive mobility devices focusing on Smart Walkers: Classification and Review. **Robotics and Autonomus Systems**, Guimarães, Portugal, v. 6, p. 548-562, 7 December 2011.

MATHWORKS, INC. Simulink Design Optimization. **MathWorks**, 2016b. Disponível em: <<http://www.mathworks.com/products/sl-design-optimization/>>. Acesso em: 01 fev. 2016.

MATHWORKS, INC. PID Controller Tuning. **MathWorks**, 2016c. Disponível em: <<http://www.mathworks.com/help/control/pid-controller-design.html>>. Acesso em: 10 fev. 2016.

MATHWORKS, INC. Simulink. **MathWorks**, 2016a. Disponível em: <<http://www.mathworks.com/products/simulink/>>. Acesso em: 20 jan. 2016.

MAXBOTIX INC. LV-MaxSonar-EZ Datasheet. **MaxBotix**, p. 11, 2015. Disponível em: <[http://www.maxbotix.com/documents/LV-MaxSonar-EZ\\_Datasheet.pdf](http://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf)>. Acesso em: 19 nov. 2015.

MENDEL, J. M. Fuzzy logic systems for engineering: a tutorial. **Proceedings of the IEEE**, v. 83, n. 3, p. 345-377, mar. 1995. ISSN 10.1109/5.364485.

MICROCHIP TECHNOLOGY INC. DataSheet MCP23017/MCP23S17. **Microchip**, 2007. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf>>. Acesso em: 30 jan. 2016.

MICROCHIP TECHNOLOGY INC. DataSheet MCP3004/3008. **Microchip**, 2008. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>>. Acesso em: 30 jan. 2016.

MICROCHIP TECHNOLOGY INC. Overview and Use of the PICmicro Serial Peripheral Interface. **Microchip**, p. 46. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf>>. Acesso em: 20 jan. 2016.

MOHAN, N.; UNDELAND, T. M.; ROBBINS, W. P. **Power Eletronics - Converts, Aplications and Design**. 3ª. ed. [S.l.]: John Wiley & Sons, 2003.

MOVELLAN, J. R. **DC Motors**. Universidade de San Diego. San Diego, p. 17. 2010.

NADPARA, N. P. et al. Quality by Design (QbD): A complete review. **International Journal of Pharmaceutical Sciences Review and Research**, 17(2), 02 out. 2012. 20-28. Disponível em: <www.globalresearchonline.net>.

NEDELKOVSKI, D. MEMS Accelerometer Gyroscope Magnetometer & Arduino. **How to Mechatronics Works**, 19 nov. 2015. Disponível em: <<http://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>>. Acesso em: 18 jan. 2016.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6ª. ed. [S.l.]: LTC, 2012.

NOURY, N. et al. **A Smart Sensor Based on Rules and its Evaluation in Daily Routines**. 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Cancún, México: [s.n.]. 2003. p. 3286-3289.

NXP SEMICONDUCTORS. **Product data sheet - 1N4148; 1N4448 High-speed diodes**. NXP Semiconductors. [S.l.], p. 9. 2004. (9397 750 13541).

NXP SEMICONDUCTORS. UM10204 - I2C bus specification and user manual. **NXP Semiconductors**, 04 abr. 2014. ISSN UM10204. Disponível em: <[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)>. Acesso em: 01 fev. 2016.

OGATA, K. **Engenharia de Controle Moderno**. 5ª. ed. São Paulo: Pearson, 2010.

OHNUMA, T.; LEE, G.; CHONG, N. Y. Particle Filter Based Feedback Control of JAIST Active Robotic Walker. **RO-MAN, 2011 IEEE**, 31 - 3 Jul - Agos 2011. 264-269.

OMS. **Guia Global das Cidades Amigas das Pessoas Idosas**. 1. ed. [S.l.]: [s.n.], 2007.

ON SEMICONDUCTOR. **DC Motor Driver Fundamentals**. Literature Distribution Center for ON Semiconductor. Denver, p. 9. 2014. (TND6041/D).

PASSINO, K. M.; YURKOVICH, S. **Fuzzy Control**. 1ª. ed. California: Addison Wesley Longman, 1997.

PAVANI, A. M. B. Notas de Aula de Controles e Servomecanismos (VOLUME 4). **Maxwell - PUC-Rio**, 06 maio 2011. Disponível em: <<http://www.maxwell.vrac.puc-rio.br/eletricaonline/serieConsulta.php?strSecao=resultado&nrSeq=17420@1>>. Acesso em: 10 jan. 2016.

PETROV, Y. Ellipsoid fit. **MathWorks**, 04 out. 2015. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit>>. Acesso em: 20 fev. 2016.

PHIDGETS. Compass Primer. **Phidgets**, 14 out. 2014. Disponível em: <[http://www.phidgets.com/docs/Compass\\_Primer](http://www.phidgets.com/docs/Compass_Primer)>. Acesso em: 01 fev. 2015.

POLOLU. Pololu Dual VNH5019 Motor Driver Shield for Arduino (ash02a). **Pololu**, 2011. Disponível em: <<https://www.pololu.com/product/2502>>. Acesso em: 01 out. 2015.

POLOLU. MinIMU-9 v2 Gyro, Accelerometer, and Compass (L3GD20 and LSM303DLHC Carrier). **Pololu**, 2014. Disponível em: <<https://www.pololu.com/product/1268>>. Acesso em: 01 out. 2015.

RASPBERRY PI FOUNDATION. RASPBERRY PI. **Raspberry Pi**. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 01 dez. 2015.

RENTSCHLER, A. J. et al. Intelligent walkers for the elderly: Performance and safety testing of VA-PAMAID robotic walker. **Department of Veterans Affairs**, v. 40, n. 3, p. 423-432, 2003.

RIBEIRO, A. et al. A influência das quedas na qualidade de vida de idosos. **Ciência e Saúde Coletiva**, v. 13, n. 4, p. 1265-1273., Jul-Ago 2008.

RICHARDS-TECH. RTIMULib. **github**, 2014a. Disponível em: <<https://github.com/richards-tech/RTIMULib>>. Acesso em: 10 out. 2015.

RICHARDS-TECH. RTIMULib Calibration. **github**, 13 out. 2014b. Disponível em: <<https://github.com/richards-tech/RTIMULib/blob/master/Calibration.pdf>>. Acesso em: 2 fev. 2016.

RIOS, J. A.; WHITE, E. Fusion filter algorithm enhancements for a MEMS GPS/IMU. **Proc. ION NTM**, 2002. 126-137.

RIVERO, A. M. **Novo método Quality by Design Fuzzy no desenvolvimento de tecnologia biomédica**. Dissertação de Mestrado, PUC-RIO. Rio de Janeiro. 2015.

RIVERO, A. M. et al. **Abordagem da Qualidade no Desenvolvimento de Tecnologia Robótica Assistiva**. Proceedings of the 1st International Workshop on Assistive Technologies. Vitoria, Espírito Santo: UFES. 2015a. p. 22-25.

RIVERO, A. M. et al. **Implementing first stages of Quality by Design approach in the development of an assistive robotics technology**. XXI IMEKO World Congress: Measurement in Research and Industry. Prague: [s.n.]. 2015b.

ROMERO, R. A. F. et al. **Robótica Móvel**. 1ª. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 2014.

SABATINI, A. M. Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing. **Biomedical Engineering, IEEE Transactions on (Volume:53 , Issue: 7 )**, 19 jul. 2006. 1346 - 1356.

SABATINI, A. M. Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation. **Sensor**, v. 11, n. 12, p. 9182–9206, 27 set. 2011. Disponível em: <<http://www.mdpi.com/1424-8220/11/10/9182>>. Acesso em: 26 jan. 25.

SAGNAC, G. Sur la preuve de la realitet de l’ether lumineaux par l’experience de l’interferographe tournant. **C. R. Acad. Sci.**, v. 157, p. 1410-1013, 1913a.

SAGNAC, G. L’ether lumineux demontre par l’effect du vent relatif d’ether dans un interferometre en rotation uniforme. **C. R. Acad. Sci**, v. 157, p. 708-710, 1913b.

SENSORWIKI. Gyroscope. **SensorWiki**, 26 abr. 2013a. Disponível em: <<http://www.sensorwiki.org/doku.php/sensors/gyroscope>>. Acesso em: 09 jan. 2016.

SHIM, I.; YOON, J.; YOH, M. A Human Robot Interactive System “RoJi”. **International Journal of Control, Automation, and Systems**, Set 2004. 398-405.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. 1<sup>a</sup>. ed. Berlin: Springer-Verlag, 2008.

SIEGWART, R.; NOURBAKHSI, I. R. **Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)**. 1<sup>a</sup>. ed. [S.l.]: Massachusetts Institute of Technology, 2004.

SIEMENS. Manual de Engenharia de Acionamentos Padrão. **Siemens**, out. 2008. Disponível em: <[http://w3.siemens.com.br/drives/br/pt/conversores/conversores-bt/sinamicsv/sinamics-v20/documentacao/treinamentos/Documents/Manual%20de%20engenharia%20standar%20drives%20\(PT\).pdf](http://w3.siemens.com.br/drives/br/pt/conversores/conversores-bt/sinamicsv/sinamics-v20/documentacao/treinamentos/Documents/Manual%20de%20engenharia%20standar%20drives%20(PT).pdf)>. Acesso em: 2015 nov. 19.

SIEMENS. Siemens Basics of DC drives. **Siemens**. Disponível em: <[https://www.industry.usa.siemens.com/services/us/en/industry-services/training/self-study-courses/quick-step-courses/Documents/dc\\_drives.pdf](https://www.industry.usa.siemens.com/services/us/en/industry-services/training/self-study-courses/quick-step-courses/Documents/dc_drives.pdf)>. Acesso em: 10 dez. 2015.

TAGHVAEI, S.; HIRATA, Y.; KOSUGE, K. Control of a Passive Walker Using a Depth Sensor for User State Estimation. **Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on**, 7-11 Dez 2011. 1639 - 1645.

TANSCHKEIT, R. Sistemas Fuzzy. **ICA - PUC-Rio**. Disponível em: <<http://www2.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas%20Fuzzy.pdf>>. Acesso em: 01 fev. 2016.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. Cambridge: The MIT Press. , 2005.

TRINDADE, R. H. **Estudo de Máquinas Elétricas Não-Convencionais: Motor Brushless DC**. TCC, Universidade de São Paulo. São Carlos, p. 30. 2009.

UPTON, E.; HALFACREE, G. **Raspberry Pi User Guide**. 1<sup>a</sup>. ed. [S.l.]: A John Wiley and Sons, Ltd., Publication, 2012.

VAUGHAN, R. Massively multi-robot simulation in stage. **Swarm Intelligence**, v. 2, n. 2<sup>a</sup>, p. 189-208, 2-4 fev. 2008. ISSN DOI 10.1007/s11721-008-0014-4.

VAUGHAN, R.; CONTRIBUTORS. The Stage Robot Simulator. **github**, 2011. Disponível em: <<http://rtv.github.io/Stage/>>. Acesso em: 01 dez. 2015.

WASSON, G. et al. An Assistive Robotic Agent for Pedestrian Mobility. **Int. Conf. Autonomous Agents AGENTS**, jan. 2001. 169-173.

WEG INDÚSTRIAS S.A. Características e Especificações de Motores de Corrente Contínua e Conversores CA/CC. **Ecatalog**. Disponível em: <<http://ecatalog.weg.net/files/wegnet/WEG-curso-dt-3-caracteristicas-e-especificacoes-de-motores-de-corrente-continua-conversores-ca-cc-artigo-tecnico-portugues-br.pdf>>. Acesso em: 01 out. 2015.

WIELENBERG, S. Corner Reflectors Can Cause Surprise Clutter. **MaxBotix**, 12 mar. 2015. Disponível em: <<http://www.maxbotix.com/articles/103.htm>>. Acesso em: 05 dez. 2015.

YU, X. Approaches and Principles of Fall Detection for elderly and patient. **e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on**, Singapore, 7-9 jul. 2008.

ZADEH, L. A. Fuzzy sets \*. **Information and Control**, v. 8, n. 3, p. 338–353, 1965.

ZHANG, P. et al. Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter. **Mechatronics and Automation, 2005 IEEE International Conference (Volume:3 )**, jul. 2005. 1497 - 1502 Vol. 3.

ZHOU, W.; XU, L.; YANG, J. An intent-based control approach for an intelligent mobility aid. **Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on**, 6-7 Mar 2010. 54-57.

## 8 Apêndices

### 8.1 Tabela de inferência do modelo virtual

Nº Regra	Ang	Ang	Ang	Ang	Sensor Alt. Esq.	Sensor Alt. Frente	Sensor Alt. Dir.	Sensor Lateral Esq.	Sensor Frente Esq.	Sensor Frente	Sensor Frente Dir.	Sensor Lateral Dir.	Dist. Obj.	Mot. E	Mot. D
0					BX	BX	BX	MP	MP	MP	MP	MP	L	NB	PM
1					BX	BX	BX	MP	MP	MP	MP	P	L	PM	NB
2					BX	BX	BX	MP	MP	MP	MP	L	L	PM	NB
3					BX	BX	BX	P	MP	MP	MP	MP	L	NB	PM
4	Z	PM	PG		BX	BX	BX	P	MP	MP	MP	P	L	NB	PM
5	NM	NG			BX	BX	BX	P	MP	MP	MP	P	L	PM	NB
6	Z	PM	PG		BX	BX	BX	P	MP	MP	MP	L	L	NB	PM
7	NM	NG			BX	BX	BX	P	MP	MP	MP	L	L	PM	NB
8					BX	BX	BX	L	MP	MP	MP	MP	L	NB	PM
9	Z	PM	PG		BX	BX	BX	L	MP	MP	MP	P	L	NB	PM
10	NM	NG			BX	BX	BX	L	MP	MP	MP	P	L	PM	NB
11	Z	PM	PG		BX	BX	BX	L	MP	MP	MP	L	L	NB	PM
12	NM	NG			BX	BX	BX	L	MP	MP	MP	L	L	PM	NB
13					BX	BX	BX	MP	MP	MP	P	MP	L	PB	Z
14					BX	BX	BX	MP	MP	MP	P	P	L	PM	NB
15					BX	BX	BX	MP	MP	MP	P	L	L	PM	NB
16					BX	BX	BX	P	MP	MP	P	MP	L	PB	Z
17					BX	BX	BX	P	MP	MP	P	P	L	PM	NB
18					BX	BX	BX	P	MP	MP	P	L	L	PM	NB
19	Z	PM	PG		BX	BX	BX	L	MP	MP	P	MP	L	NB	PM
20	NM	NG			BX	BX	BX	L	MP	MP	P	MP	L	PB	Z
21					BX	BX	BX	L	MP	MP	P	P	L	PM	NB
22					BX	BX	BX	L	MP	MP	P	L	L	PM	NB
23					BX	BX	BX	MP	MP	MP	L	MP	L	PB	Z
24					BX	BX	BX	MP	MP	MP	L	P	L	PM	NB
25					BX	BX	BX	MP	MP	MP	L	L	L	PM	NB
26					BX	BX	BX	P	MP	MP	L	MP	L	PB	Z
27					BX	BX	BX	P	MP	MP	L	P	L	PM	NB
28					BX	BX	BX	P	MP	MP	L	L	L	PM	NB
29	Z	PM	PG		BX	BX	BX	L	MP	MP	L	MP	L	NB	PM
30	NM	NG			BX	BX	BX	L	MP	MP	L	MP	L	PB	Z
31					BX	BX	BX	L	MP	MP	L	P	L	PM	NB
32					BX	BX	BX	L	MP	MP	L	L	L	PM	NB

33	Z	PM	PG		BX	BX	BX	MP	MP	P	MP	MP	L	Z	PB
34	NM	NG			BX	BX	BX	MP	MP	P	MP	MP	L	PB	Z
35					BX	BX	BX	MP	MP	P	MP	P	L	PB	Z
36					BX	BX	BX	MP	MP	P	MP	L	L	PM	NB
37					BX	BX	BX	P	MP	P	MP	MP	L	Z	PB
38	Z	PM	PG		BX	BX	BX	P	MP	P	MP	P	L	Z	PB
39	NM	NG			BX	BX	BX	P	MP	P	MP	P	L	PB	Z
40	Z	PM	PG		BX	BX	BX	P	MP	P	MP	L	L	Z	PB
41	NM	NG			BX	BX	BX	P	MP	P	MP	L	L	PM	NB
42					BX	BX	BX	L	MP	P	MP	MP	L	NB	PM
43	Z	PM	PG		BX	BX	BX	L	MP	P	MP	P	L	NB	PM
44	NM	NG			BX	BX	BX	L	MP	P	MP	P	L	PB	Z
45	Z	PM	PG		BX	BX	BX	L	MP	P	MP	L	L	NB	PM
46	NM	NG			BX	BX	BX	L	MP	P	MP	L	L	PM	NB
47	Z	PM	PG		BX	BX	BX	MP	MP	L	MP	MP	L	Z	PB
48	NM	NG			BX	BX	BX	MP	MP	L	MP	MP	L	PB	Z
49					BX	BX	BX	MP	MP	L	MP	P	L	PB	Z
50					BX	BX	BX	MP	MP	L	MP	L	L	PM	NB
51					BX	BX	BX	P	MP	L	MP	MP	L	Z	PB
52	Z	PM	PG		BX	BX	BX	P	MP	L	MP	P	L	Z	PB
53	NM	NG			BX	BX	BX	P	MP	L	MP	P	L	PB	Z
54	Z	PM	PG		BX	BX	BX	P	MP	L	MP	L	L	Z	PB
55	NM	NG			BX	BX	BX	P	MP	L	MP	L	L	PM	NB
56					BX	BX	BX	L	MP	L	MP	MP	L	NB	PM
57	Z	PM	PG		BX	BX	BX	L	MP	L	MP	P	L	NB	PM
58	NM	NG			BX	BX	BX	L	MP	L	MP	P	L	PB	Z
59	Z	PM	PG		BX	BX	BX	L	MP	L	MP	L	L	NB	PM
60	NM	NG			BX	BX	BX	L	MP	L	MP	L	L	PM	NB
61					BX	BX	BX	MP	P	MP	MP	MP	L	Z	PB
62					BX	BX	BX	MP	P	MP	MP	P	L	Z	PB
63	Z	PM	PG		BX	BX	BX	MP	P	MP	MP	L	L	Z	PB
64	NM	NG			BX	BX	BX	MP	P	MP	MP	L	L	PM	NB
65					BX	BX	BX	P	P	MP	MP	MP	L	NB	PM
66					BX	BX	BX	P	P	MP	MP	P	L	NB	PM
67					BX	BX	BX	P	P	MP	MP	L	L	NB	PM
68					BX	BX	BX	L	P	MP	MP	MP	L	NB	PM
69					BX	BX	BX	L	P	MP	MP	P	L	NB	PM
70					BX	BX	BX	L	P	MP	MP	L	L	NB	PM
71					BX	BX	BX	MP	L	MP	MP	MP	L	Z	PB
72					BX	BX	BX	MP	L	MP	MP	P	L	Z	PB
73	Z	PM	PG		BX	BX	BX	MP	L	MP	MP	L	L	Z	PB
74	NM	NG			BX	BX	BX	MP	L	MP	MP	L	L	PM	NB
75					BX	BX	BX	P	L	MP	MP	MP	L	NB	PM
76					BX	BX	BX	P	L	MP	MP	P	L	NB	PM
77					BX	BX	BX	P	L	MP	MP	L	L	NB	PM
78					BX	BX	BX	L	L	MP	MP	MP	L	NB	PM

79					BX	BX	BX	L	L	MP	MP	P	L	NB	PM
80					BX	BX	BX	L	L	MP	MP	L	L	NB	PM
81					BX	BX	BX	MP	P	P	P	MP	L	PM	PM
82	Z	PM	PG		BX	BX	BX	MP	P	P	P	P	L	PM	PM
83	NM				BX	BX	BX	MP	P	P	P	P	L	PM	PB
84	NG				BX	BX	BX	MP	P	P	P	P	L	PM	Z
85	Z	PM	PG		BX	BX	BX	MP	P	P	P	L	L	PM	PM
86	NM				BX	BX	BX	MP	P	P	P	L	L	PM	PB
87	NG				BX	BX	BX	MP	P	P	P	L	L	PM	Z
88	Z	NM	NG		BX	BX	BX	P	P	P	P	MP	L	PM	PM
89	PM				BX	BX	BX	P	P	P	P	MP	L	PB	PM
90	PG				BX	BX	BX	P	P	P	P	MP	L	Z	PM
91	Z				BX	BX	BX		P	P	P		L	PM	PM
92	PM				BX	BX	BX	P	P	P	P	P	L	PB	PM
93	PG				BX	BX	BX	P	P	P	P	P	L	Z	PM
94	NM				BX	BX	BX	P	P	P	P	P	L	PM	PB
95	NG				BX	BX	BX	P	P	P	P	P	L	PM	Z
96	PM				BX	BX	BX	P	P	P	P	L	L	PB	PM
97	PG				BX	BX	BX	P	P	P	P	L	L	Z	PM
98	NM				BX	BX	BX	P	P	P	P	L	L	PM	PB
99	NG				BX	BX	BX	P	P	P	P	L	L	PM	Z
100	NM	NG			BX	BX	BX	L	P	P	P	MP	L	PM	PM
101	PM				BX	BX	BX	L	P	P	P	MP	L	PB	PM
102	PG				BX	BX	BX	L	P	P	P	MP	L	Z	PM
103	PM				BX	BX	BX	L	P	P	P	P	L	PB	PM
104	PG				BX	BX	BX	L	P	P	P	P	L	Z	PM
105	NM				BX	BX	BX	L	P	P	P	P	L	PM	PB
106	NG				BX	BX	BX	L	P	P	P	P	L	PM	Z
107	PM				BX	BX	BX	L	P	P	P	L	L	PB	PM
108	PG				BX	BX	BX	L	P	P	P	L	L	Z	PM
109	NM				BX	BX	BX	L	P	P	P	L	L	PM	PB
110	NG				BX	BX	BX	L	P	P	P	L	L	PM	Z
111					BX	BX	BX	MP	P	P	MP	MP	L	Z	PB
112					BX	BX	BX	MP	P	P	MP	P	L	Z	PB
113					BX	BX	BX	MP	P	P	MP	L	L	Z	PB
114	Z	NM	NG	PM	BX	BX	BX	P	P	P	MP	MP	L	Z	PM
115	PG				BX	BX	BX	P	P	P	MP	MP	L	NB	PM
116	Z	NM	NG	PM	BX	BX	BX	P	P	P	MP	P	L	Z	PM
117	PG				BX	BX	BX	P	P	P	MP	P	L	NB	PM
118	Z	NM	NG	PM	BX	BX	BX	P	P	P	MP	L	L	Z	PM
119	PG				BX	BX	BX	P	P	P	MP	L	L	NB	PM
120	Z	NM	NG	PM	BX	BX	BX	L	P	P	MP	MP	L	Z	PM
121	PG				BX	BX	BX	L	P	P	MP	MP	L	NB	PM
122	Z	NM	NG	PM	BX	BX	BX	L	P	P	MP	P	L	Z	PM
123	PG				BX	BX	BX	L	P	P	MP	P	L	NB	PM
124	Z	NM	NG	PM	BX	BX	BX	L	P	P	MP	L	L	Z	PM



125	PG				BX	BX	BX	L	P	P	MP	L	L	NB	PM
126					BX	BX	BX	MP	P	P	L	MP	L	PM	PB
127	PM	PG			BX	BX	BX	MP	P	P	L	P	L	PM	PB
128	NM				BX	BX	BX	MP	P	P	L	P	L	PM	PB
129	NG				BX	BX	BX	MP	P	P	L	P	L	PM	Z
130	PM	PG			BX	BX	BX	MP	P	P	L	L	L	PM	PM
131	NM				BX	BX	BX	MP	P	P	L	L	L	PA	PM
132	NG				BX	BX	BX	MP	P	P	L	L	L	PM	Z
133	NM	NG			BX	BX	BX	P	P	P	L	MP	L	PM	PM
134	PM				BX	BX	BX	P	P	P	L	MP	L	PB	PM
135	PG				BX	BX	BX	P	P	P	L	MP	L	Z	PM
136	Z				BX	BX	BX		P	P	L		L	PM	PM
137	PM				BX	BX	BX	P	P	P	L	P	L	PB	PM
138	PG				BX	BX	BX	P	P	P	L	P	L	Z	PM
139	NM				BX	BX	BX	P	P	P	L	P	L	PM	PB
140	NG				BX	BX	BX	P	P	P	L	P	L	PM	Z
141	PM				BX	BX	BX	P	P	P	L	L	L	PB	PM
142	PG				BX	BX	BX	P	P	P	L	L	L	Z	PM
143	NM				BX	BX	BX	P	P	P	L	L	L	PA	PM
144	NG				BX	BX	BX	P	P	P	L	L	L	PM	Z
145	NM	NG			BX	BX	BX	L	P	P	L	MP	L	PM	PM
146	PM				BX	BX	BX	L	P	P	L	MP	L	PB	PM
147	PG				BX	BX	BX	L	P	P	L	MP	L	Z	PM
148	PM				BX	BX	BX	L	P	P	L	P	L	PB	PM
149	PG				BX	BX	BX	L	P	P	L	P	L	Z	PM
150	NM				BX	BX	BX	L	P	P	L	P	L	PM	PB
151	NG				BX	BX	BX	L	P	P	L	P	L	PM	Z
152	PM				BX	BX	BX	L	P	P	L	L	L	PB	PM
153	PG				BX	BX	BX	L	P	P	L	L	L	Z	PM
154	NM				BX	BX	BX	L	P	P	L	L	L	PA	PM
155	NG				BX	BX	BX	L	P	P	L	L	L	PM	Z
156	Z	PM	PG		BX	BX	BX	MP	P	MP	P	MP	L	Z	PB
157	NM	NG			BX	BX	BX	MP	P	MP	P	MP	L	PB	Z
158					BX	BX	BX	MP	P	MP	P	P	L	PM	NB
159					BX	BX	BX	MP	P	MP	P	L	L	PM	NB
160					BX	BX	BX	P	P	MP	P	MP	L	NB	PM
161	Z	PM	PG		BX	BX	BX	P	P	MP	P	P	L	NB	PM
162	NM	NG			BX	BX	BX	P	P	MP	P	P	L	PM	NB
163	Z	PM	PG		BX	BX	BX	P	P	MP	P	L	L	NB	PM
164	NM	NG			BX	BX	BX	P	P	MP	P	L	L	PM	NB
165					BX	BX	BX	L	P	MP	P	MP	L	NB	PM
166	Z	PM	PG		BX	BX	BX	L	P	MP	P	P	L	NB	PM
167	NM	NG			BX	BX	BX	L	P	MP	P	P	L	PM	NB
168	Z	PM	PG		BX	BX	BX	L	P	MP	P	L	L	NB	PM
169	NM	NG			BX	BX	BX	L	P	MP	P	L	L	PM	NB
170					BX	BX	BX	MP	P	L	P	MP	L	PM	PM

171	PM	PG			BX	BX	BX	MP	P	L	P	P	L	PM	PM
172	NM				BX	BX	BX	MP	P	L	P	P	L	PM	PB
173	NG				BX	BX	BX	MP	P	L	P	P	L	PM	Z
174	PM	PG			BX	BX	BX	MP	P	L	P	L	L	PM	PM
175	NM				BX	BX	BX	MP	P	L	P	L	L	PM	PB
176	NG				BX	BX	BX	MP	P	L	P	L	L	PM	Z
177	NM	NG			BX	BX	BX	P	P	L	P	MP	L	PM	PM
178	PM				BX	BX	BX	P	P	L	P	MP	L	PB	PM
179	PG				BX	BX	BX	P	P	L	P	MP	L	Z	PM
180	Z				BX	BX	BX		P	L	P		L	PM	PM
181	PM				BX	BX	BX	P	P	L	P	P	L	PB	PM
182	PG				BX	BX	BX	P	P	L	P	P	L	Z	PM
183	NM				BX	BX	BX	P	P	L	P	P	L	PM	PB
184	NG				BX	BX	BX	P	P	L	P	P	L	PM	Z
185	PM				BX	BX	BX	P	P	L	P	L	L	PB	PM
186	PG				BX	BX	BX	P	P	L	P	L	L	Z	PM
187	NM				BX	BX	BX	P	P	L	P	L	L	PM	PB
188	NG				BX	BX	BX	P	P	L	P	L	L	PM	Z
189	NM	NG			BX	BX	BX	L	P	L	P	MP	L	PM	PM
190	PM				BX	BX	BX	L	P	L	P	MP	L	PB	PM
191	PG				BX	BX	BX	L	P	L	P	MP	L	Z	PM
192	PM				BX	BX	BX	L	P	L	P	P	L	PB	PM
193	PG				BX	BX	BX	L	P	L	P	P	L	Z	PM
194	NM				BX	BX	BX	L	P	L	P	P	L	PM	PB
195	NG				BX	BX	BX	L	P	L	P	P	L	PM	Z
196	PM				BX	BX	BX	L	P	L	P	L	L	PB	PM
197	PG				BX	BX	BX	L	P	L	P	L	L	Z	PM
198	NM				BX	BX	BX	L	P	L	P	L	L	PM	PB
199	NG				BX	BX	BX	L	P	L	P	L	L	PM	Z
200					BX	BX	BX	MP	MP	P	P	MP	L	PB	Z
201	Z	PM	PG	NM	BX	BX	BX	MP	MP	P	P	P	L	PM	Z
202	NG				BX	BX	BX	MP	MP	P	P	P	L	PM	NB
203	Z	PM	PG	NM	BX	BX	BX	MP	MP	P	P	L	L	PM	Z
204	NG				BX	BX	BX	MP	MP	P	P	L	L	PM	NB
205					BX	BX	BX	P	MP	P	P	MP	L	PB	Z
206	Z	PM	PG	NM	BX	BX	BX	P	MP	P	P	P	L	PM	Z
207	NG				BX	BX	BX	P	MP	P	P	P	L	PM	NB
208	Z	PM	PG	NM	BX	BX	BX	P	MP	P	P	L	L	PM	Z
209	NG				BX	BX	BX	P	MP	P	P	L	L	PM	NB
210					BX	BX	BX	L	MP	P	P	MP	L	PB	Z
211	Z	PM	PG	NM	BX	BX	BX	L	MP	P	P	P	L	PM	Z
212	NG				BX	BX	BX	L	MP	P	P	P	L	PM	NB
213	Z	PM	PG	NM	BX	BX	BX	L	MP	P	P	L	L	PM	Z
214	NG				BX	BX	BX	L	MP	P	P	L	L	PM	NB
215					BX	BX	BX	MP	L	P	P	MP	L	PB	PM
216	PM	PG			BX	BX	BX	MP	L	P	P	P	L	PM	PM

217	NM				BX	BX	BX	MP	L	P	P	P	L	PM	PB
218	NG				BX	BX	BX	MP	L	P	P	P	L	PM	Z
219	PM	PG			BX	BX	BX	MP	L	P	P	L	L	PM	PM
220	NM				BX	BX	BX	MP	L	P	P	L	L	PM	PB
221	NG				BX	BX	BX	MP	L	P	P	L	L	PM	Z
222	NM	NG			BX	BX	BX	P	L	P	P	MP	L	PM	PM
223	PM				BX	BX	BX	P	L	P	P	MP	L	PB	PM
224	PG				BX	BX	BX	P	L	P	P	MP	L	Z	PM
225	Z				BX	BX	BX		L	P	P		L	PM	PM
226	PM				BX	BX	BX	P	L	P	P	P	L	PB	PM
227	PG				BX	BX	BX	P	L	P	P	P	L	Z	PM
228	NM				BX	BX	BX	P	L	P	P	P	L	PM	PB
229	NG				BX	BX	BX	P	L	P	P	P	L	PM	Z
230	PM				BX	BX	BX	P	L	P	P	L	L	PB	PM
231	PG				BX	BX	BX	P	L	P	P	L	L	Z	PM
232	NM				BX	BX	BX	P	L	P	P	L	L	PM	PB
233	NG				BX	BX	BX	P	L	P	P	L	L	PM	Z
234	NM	NG			BX	BX	BX	L	L	P	P	MP	L	PM	PM
235	PM				BX	BX	BX	L	L	P	P	MP	L	PM	PA
236	PG				BX	BX	BX	L	L	P	P	MP	L	Z	PM
237	PM				BX	BX	BX	L	L	P	P	P	L	PM	PA
238	PG				BX	BX	BX	L	L	P	P	P	L	Z	PM
239	NM				BX	BX	BX	L	L	P	P	P	L	PM	PB
240	NG				BX	BX	BX	L	L	P	P	P	L	PM	Z
241	PM				BX	BX	BX	L	L	P	P	L	L	PM	PA
242	PG				BX	BX	BX	L	L	P	P	L	L	PB	PA
243	NM				BX	BX	BX	L	L	P	P	L	L	PM	PB
244	NG				BX	BX	BX	L	L	P	P	L	L	PM	Z
245					BX	BX	BX	MP	L	L	L	MP	L	PA	PA
246	PM	PG			BX	BX	BX	MP	L	L	L	P	L	PA	PA
247	NM				BX	BX	BX	MP	L	L	L	P	L	PA	PM
248	NG				BX	BX	BX	MP	L	L	L	P	L	PA	PB
249	PM	PG			BX	BX	BX	MP	L	L	L	L	L	PA	PA
250	NM				BX	BX	BX	MP	L	L	L	L	L	PA	PM
251	NG				BX	BX	BX	MP	L	L	L	L	L	PA	PB
252	NM	NG			BX	BX	BX	P	L	L	L	MP	L	PA	PA
253	PM				BX	BX	BX	P	L	L	L	MP	L	PM	PA
254	PG				BX	BX	BX	P	L	L	L	MP	L	PB	PA
255	Z				BX	BX	BX		L	L	L		L	PA	PA
256	PM				BX	BX	BX	P	L	L	L	P	L	PM	PA
257	PG				BX	BX	BX	P	L	L	L	P	L	PB	PA
258	NM				BX	BX	BX	P	L	L	L	P	L	PA	PM
259	NG				BX	BX	BX	P	L	L	L	P	L	PA	PB
260	PM				BX	BX	BX	P	L	L	L	L	L	PM	PA
261	PG				BX	BX	BX	P	L	L	L	L	L	PB	PA
262	NM				BX	BX	BX	P	L	L	L	L	L	PA	PM

263	NG				BX	BX	BX	P	L	L	L	L	L	PA	PB
264	NM	NG			BX	BX	BX	L	L	L	L	MP	L	PA	PA
265	PM				BX	BX	BX	L	L	L	L	MP	L	PM	PA
266	PG				BX	BX	BX	L	L	L	L	MP	L	PB	PA
267	PM				BX	BX	BX	L	L	L	L	P	L	PM	PA
268	PG				BX	BX	BX	L	L	L	L	P	L	PB	PA
269	NM				BX	BX	BX	L	L	L	L	P	L	PA	PM
270	NG				BX	BX	BX	L	L	L	L	P	L	PA	PB
271	PM				BX	BX	BX	L	L	L	L	L	L	PM	PA
272	PG				BX	BX	BX	L	L	L	L	L	L	PB	PA
273	NM				BX	BX	BX	L	L	L	L	L	L	PA	PM
274	NG				BX	BX	BX	L	L	L	L	L	L	PA	PB
275					BX	BX	BX	MP	L	L	MP	MP	L	Z	PB
276					BX	BX	BX	MP	L	L	MP	P	L	Z	PB
277					BX	BX	BX	MP	L	L	MP	L	L	Z	PB
278	Z	NM	NG	PM	BX	BX	BX	P	L	L	MP	MP	L	Z	PM
279	PG				BX	BX	BX	P	L	L	MP	MP	L	NB	PM
280	Z	NM	NG	PM	BX	BX	BX	P	L	L	MP	P	L	Z	PM
281	PG				BX	BX	BX	P	L	L	MP	P	L	NB	PM
282	Z	NM	NG	PM	BX	BX	BX	P	L	L	MP	L	L	Z	PM
283	PG				BX	BX	BX	P	L	L	MP	L	L	NB	PM
284	Z	NM	NG	PM	BX	BX	BX	L	L	L	MP	MP	L	Z	PM
285	PG				BX	BX	BX	L	L	L	MP	MP	L	NB	PM
286	Z	NM	NG	PM	BX	BX	BX	L	L	L	MP	P	L	Z	PM
287	PG				BX	BX	BX	L	L	L	MP	P	L	NB	PM
288	Z	NM	NG	PM	BX	BX	BX	L	L	L	MP	L	L	Z	PM
289	PG				BX	BX	BX	L	L	L	MP	L	L	NB	PM
290					BX	BX	BX	MP	L	L	P	MP	L	PM	PA
291	PM	PG			BX	BX	BX	MP	L	L	P	P	L	PM	PM
292	NM				BX	BX	BX	MP	L	L	P	P	L	PM	PB
293	NG				BX	BX	BX	MP	L	L	P	P	L	PM	Z
294	PM	PG			BX	BX	BX	MP	L	L	P	L	L	PM	PM
295	NM				BX	BX	BX	MP	L	L	P	L	L	PA	PM
296	NG				BX	BX	BX	MP	L	L	P	L	L	PA	PB
297	NM	NG			BX	BX	BX	P	L	L	P	MP	L	PM	PM
298	PM				BX	BX	BX	P	L	L	P	MP	L	PM	PA
299	PG				BX	BX	BX	P	L	L	P	MP	L	PB	PA
300	Z				BX	BX	BX		L	L	P		L	PM	PA
301	PM				BX	BX	BX	P	L	L	P	P	L	PM	PA
302	PG				BX	BX	BX	P	L	L	P	P	L	Z	PM
303	NM				BX	BX	BX	P	L	L	P	P	L	PM	PB
304	NG				BX	BX	BX	P	L	L	P	P	L	PM	Z
305	PM				BX	BX	BX	P	L	L	P	L	L	PM	PA
306	PG				BX	BX	BX	P	L	L	P	L	L	PB	PA
307	NM				BX	BX	BX	P	L	L	P	L	L	PA	PM
308	NG				BX	BX	BX	P	L	L	P	L	L	PA	PB

309	NM	NG			BX	BX	BX	L	L	L	P	MP	L	PM	PM
310	PM				BX	BX	BX	L	L	L	P	MP	L	PM	PA
311	PG				BX	BX	BX	L	L	L	P	MP	L	PB	PA
312	PM				BX	BX	BX	L	L	L	P	P	L	PM	PA
313	PG				BX	BX	BX	L	L	L	P	P	L	PB	PA
314	NM				BX	BX	BX	L	L	L	P	P	L	PM	PB
315	NG				BX	BX	BX	L	L	L	P	P	L	PM	Z
316	PM				BX	BX	BX	L	L	L	P	L	L	PM	PA
317	PG				BX	BX	BX	L	L	L	P	L	L	PB	PA
318	NM				BX	BX	BX	L	L	L	P	L	L	PA	PM
319	NG				BX	BX	BX	L	L	L	P	L	L	PA	PB
320	Z	PM	PG		BX	BX	BX	MP	L	MP	L	MP	L	Z	PB
321	NM	NG			BX	BX	BX	MP	L	MP	L	MP	L	PB	Z
322					BX	BX	BX	MP	L	MP	L	P	L	PM	NB
323					BX	BX	BX	MP	L	MP	L	L	L	PM	NB
324					BX	BX	BX	P	L	MP	L	MP	L	NB	PM
325	PM	PG			BX	BX	BX	P	L	MP	L	P	L	NB	PM
326	PM	PG			BX	BX	BX	P	L	MP	L	P	L	NB	PM
327	Z	PM	PG		BX	BX	BX	P	L	MP	L	P	L	NB	PM
328	PM	PG			BX	BX	BX	P	L	MP	L	P	L	NB	PM
329	Z	PM	PG		BX	BX	BX	P	L	MP	L	L	L	NB	PM
330	NM	NG			BX	BX	BX	P	L	MP	L	L	L	PM	NB
331					BX	BX	BX	L	L	MP	L	MP	L	NB	PM
332	Z	PM	PG		BX	BX	BX	L	L	MP	L	P	L	NB	PM
333	PM	PG			BX	BX	BX	L	L	MP	L	P	L	NB	PM
334	NM	NG			BX	BX	BX	L	L	MP	L	P	L	PM	NB
335	NM	NG			BX	BX	BX	L	L	MP	L	P	L	PM	NB
336	Z	PM	PG		BX	BX	BX	L	L	MP	L	L	L	NB	PM
337	PM	PG			BX	BX	BX	L	L	MP	L	L	L	NB	PM
338	NM	NG			BX	BX	BX	L	L	MP	L	L	L	PM	NB
339	NM	NG			BX	BX	BX	L	L	MP	L	L	L	PM	NB
340					BX	BX	BX	MP	L	P	L	MP	L	PM	PM
341	PM	PG			BX	BX	BX	MP	L	P	L	P	L	PM	PB
342	NM				BX	BX	BX	MP	L	P	L	P	L	PM	PB
343	NG				BX	BX	BX	MP	L	P	L	P	L	PM	Z
344	PM	PG			BX	BX	BX	MP	L	P	L	L	L	PM	PM
345	NM				BX	BX	BX	MP	L	P	L	L	L	PA	PM
346	NG				BX	BX	BX	MP	L	P	L	L	L	PA	PB
347	NM	NG			BX	BX	BX	P	L	P	L	MP	L	PM	PB
348	PM				BX	BX	BX	P	L	P	L	MP	L	PB	PM
349	PG				BX	BX	BX	P	L	P	L	MP	L	Z	PM
350	Z				BX	BX	BX		L	P	L		L	PM	PM
351	PM				BX	BX	BX	P	L	P	L	P	L	PB	PM
352	PG				BX	BX	BX	P	L	P	L	P	L	Z	PM
353	NM				BX	BX	BX	P	L	P	L	P	L	PM	PB
354	NG				BX	BX	BX	P	L	P	L	P	L	PM	Z

355	PM				BX	BX	BX	P	L	P	L	L	L	PB	PM
356	PG				BX	BX	BX	P	L	P	L	L	L	Z	PM
357	NM				BX	BX	BX	P	L	P	L	L	L	PA	PM
358	NG				BX	BX	BX	P	L	P	L	L	L	PA	PB
359	NM	NG			BX	BX	BX	L	L	P	L	MP	L	PM	PB
360	PM				BX	BX	BX	L	L	P	L	MP	L	PM	PA
361	PG				BX	BX	BX	L	L	P	L	MP	L	PB	PA
362	PM				BX	BX	BX	L	L	P	L	P	L	PM	PA
363	PG				BX	BX	BX	L	L	P	L	P	L	PB	PA
364	NM				BX	BX	BX	L	L	P	L	P	L	PA	PM
365	NG				BX	BX	BX	L	L	P	L	P	L	PA	PB
366	PM				BX	BX	BX	L	L	P	L	L	L	PM	PA
367	PG				BX	BX	BX	L	L	P	L	L	L	PB	PA
368	NM				BX	BX	BX	L	L	P	L	L	L	PA	PM
369	NG				BX	BX	BX	L	L	P	L	L	L	PA	PB
370					BX	BX	BX	MP	MP	L	L	MP	L	PB	Z
371	Z	PM	PG	NM	BX	BX	BX	MP	MP	L	L	P	L	PM	Z
372	NG				BX	BX	BX	MP	MP	L	L	P	L	PM	NB
373	Z	PM	PG	NM	BX	BX	BX	MP	MP	L	L	L	L	PM	Z
374	NG				BX	BX	BX	MP	MP	L	L	L	L	PM	NB
375					BX	BX	BX	P	MP	L	L	MP	L	PB	Z
376	Z	PM	PG	NM	BX	BX	BX	P	MP	L	L	P	L	PM	Z
377	NG				BX	BX	BX	P	MP	L	L	P	L	PM	NB
378	Z	PM	PG	NM	BX	BX	BX	P	MP	L	L	L	L	PM	Z
379	NG				BX	BX	BX	P	MP	L	L	L	L	PM	NB
380					BX	BX	BX	L	MP	L	L	MP	L	PB	Z
381	Z	PM	PG	NM	BX	BX	BX	L	MP	L	L	P	L	PM	Z
382	NG				BX	BX	BX	L	MP	L	L	P	L	PM	NB
383	Z	PM	PG	NM	BX	BX	BX	L	MP	L	L	L	L	PM	Z
384	NG				BX	BX	BX	L	MP	L	L	L	L	PM	NB
385					BX	BX	BX	MP	P	L	L	MP	L	PA	PM
386	PM	PG			BX	BX	BX	MP	P	L	L	P	L	PA	PM
387	NM				BX	BX	BX	MP	P	L	L	P	L	PA	PM
388	NG				BX	BX	BX	MP	P	L	L	P	L	PA	PB
389	PM	PG			BX	BX	BX	MP	P	L	L	L	L	PM	PM
390	NM				BX	BX	BX	MP	P	L	L	L	L	PA	PM
391	NG				BX	BX	BX	MP	P	L	L	L	L	PA	PB
392	NM	NG			BX	BX	BX	P	P	L	L	MP	L	PA	PM
393	PM				BX	BX	BX	P	P	L	L	MP	L	PB	PM
394	PG				BX	BX	BX	P	P	L	L	MP	L	Z	PM
395	Z				BX	BX	BX		P	L	L		L	PA	PM
396	PM				BX	BX	BX	P	P	L	L	P	L	PB	PM
397	PG				BX	BX	BX	P	P	L	L	P	L	Z	PM
398	NM				BX	BX	BX	P	P	L	L	P	L	PA	PM
399	NG				BX	BX	BX	P	P	L	L	P	L	PA	PB
400	PM				BX	BX	BX	P	P	L	L	L	L	PB	PM

401	PG				BX	BX	BX	P	P	L	L	L	L	Z	PM
402	NM				BX	BX	BX	P	P	L	L	L	L	PA	PM
403	NG				BX	BX	BX	P	P	L	L	L	L	PA	PB
404	NM	NG			BX	BX	BX	L	P	L	L	MP	L	PA	PM
405	PM				BX	BX	BX	L	P	L	L	MP	L	PB	PM
406	PG				BX	BX	BX	L	P	L	L	MP	L	Z	PM
407	PM				BX	BX	BX	L	P	L	L	P	L	PM	PA
408	PG				BX	BX	BX	L	P	L	L	P	L	PB	PA
409	NM				BX	BX	BX	L	P	L	L	P	L	PA	PM
410	NG				BX	BX	BX	L	P	L	L	P	L	PA	PB
411	PM				BX	BX	BX	L	P	L	L	L	L	PB	PM
412	PG				BX	BX	BX	L	P	L	L	L	L	Z	PM
413	NM				BX	BX	BX	L	P	L	L	L	L	PA	PM
414	NG				BX	BX	BX	L	P	L	L	L	L	PA	PB
415													P	Z	Z
416					BX	BX	BX	MP	MP	P	L	MP	L	PB	Z
417	Z	PM	PG	NM	BX	BX	BX	MP	MP	P	L	P	L	PM	Z
418	NG				BX	BX	BX	MP	MP	P	L	P	L	PM	NB
419	Z	PM	PG	NM	BX	BX	BX	MP	MP	P	L	L	L	PM	Z
420	NG				BX	BX	BX	MP	MP	P	L	L	L	PM	NB
421					BX	BX	BX	P	MP	P	L	MP	L	PB	Z
422	Z	PM	PG	NM	BX	BX	BX	P	MP	P	L	P	L	PM	Z
423	NG				BX	BX	BX	P	MP	P	L	P	L	PM	NB
424	Z	PM	PG	NM	BX	BX	BX	P	MP	P	L	L	L	PM	Z
425	NG				BX	BX	BX	P	MP	P	L	L	L	PM	NB
426					BX	BX	BX	L	MP	P	L	MP	L	PB	Z
427	Z	PM	PG	NM	BX	BX	BX	L	MP	P	L	P	L	PM	Z
428	NG				BX	BX	BX	L	MP	P	L	P	L	PM	NB
429	Z	PM	PG	NM	BX	BX	BX	L	MP	P	L	L	L	PM	Z
430	NG				BX	BX	BX	L	MP	P	L	L	L	PM	NB
431					BX	BX	BX	MP	MP	L	P	MP	L	PB	Z
432	Z	PM	PG	NM	BX	BX	BX	MP	MP	L	P	P	L	PM	Z
433	NG				BX	BX	BX	MP	MP	L	P	P	L	PM	NB
434	Z	PM	PG	NM	BX	BX	BX	MP	MP	L	P	L	L	PM	Z
435	NG				BX	BX	BX	MP	MP	L	P	L	L	PM	NB
436					BX	BX	BX	P	MP	L	P	MP	L	PB	Z
437	Z	PM	PG	NM	BX	BX	BX	P	MP	L	P	P	L	PM	Z
438	NG				BX	BX	BX	P	MP	L	P	P	L	PM	NB
439	Z	PM	PG	NM	BX	BX	BX	P	MP	L	P	L	L	PM	Z
440	NG				BX	BX	BX	P	MP	L	P	L	L	PM	NB
441					BX	BX	BX	L	MP	L	P	MP	L	PB	Z
442	Z	PM	PG	NM	BX	BX	BX	L	MP	L	P	P	L	PM	Z
443	NG				BX	BX	BX	L	MP	L	P	P	L	PM	NB
444	Z	PM	PG	NM	BX	BX	BX	L	MP	L	P	L	L	PM	Z
445	NG				BX	BX	BX	L	MP	L	P	L	L	PM	NB
446	Z	PM	PG		BX	BX	BX	MP	P	MP	L	MP	L	Z	PB

447	NM	NG			BX	BX	BX	MP	P	MP	L	MP	L	PB	Z
448					BX	BX	BX	MP	P	MP	L	P	L	PM	NB
449					BX	BX	BX	MP	P	MP	L	L	L	PM	NB
450					BX	BX	BX	P	P	MP	L	MP	L	NB	PM
451	Z	PM	PG		BX	BX	BX	P	P	MP	L	P	L	NB	PM
452	NM	NG			BX	BX	BX	P	P	MP	L	P	L	PM	NB
453	Z	PM	PG		BX	BX	BX	P	P	MP	L	L	L	NB	PM
454	NM	NG			BX	BX	BX	P	P	MP	L	L	L	PM	NB
455	NM	NG			BX	BX	BX	P	P	MP	L	L	L	PM	NB
456					BX	BX	BX	L	P	MP	L	MP	L	NB	PM
457	Z	PM	PG		BX	BX	BX	L	P	MP	L	P	L	NB	PM
458	PM	PG			BX	BX	BX	L	P	MP	L	P	L	NB	PM
459	NM	NG			BX	BX	BX	L	P	MP	L	P	L	PM	NB
460	Z	PM	PG		BX	BX	BX	L	P	MP	L	L	L	NB	PM
461	PM	PG			BX	BX	BX	L	P	MP	L	L	L	NB	PM
462	NM	NG			BX	BX	BX	L	P	MP	L	L	L	PM	NB
463	NM	NG			BX	BX	BX	L	P	MP	L	L	L	PM	NB
464					BX	BX	BX	MP	P	L	MP	MP	L	Z	PB
465					BX	BX	BX	MP	P	L	MP	P	L	Z	PB
466					BX	BX	BX	MP	P	L	MP	L	L	Z	PB
467	Z	NM	NG	PM	BX	BX	BX	P	P	L	MP	MP	L	Z	PM
468	PG				BX	BX	BX	P	P	L	MP	MP	L	NB	PM
469	Z	NM	NG	PM	BX	BX	BX	P	P	L	MP	P	L	Z	PM
470	PG				BX	BX	BX	P	P	L	MP	P	L	NB	PM
471	Z	NM	NG	PM	BX	BX	BX	P	P	L	MP	L	L	Z	PM
472	PG				BX	BX	BX	P	P	L	MP	L	L	NB	PM
473	Z	NM	NG	PM	BX	BX	BX	L	P	L	MP	MP	L	Z	PM
474	PG				BX	BX	BX	L	P	L	MP	MP	L	NB	PM
475	Z	NM	NG	PM	BX	BX	BX	L	P	L	MP	P	L	Z	PM
476	PG				BX	BX	BX	L	P	L	MP	P	L	NB	PM
477	Z	NM	NG	PM	BX	BX	BX	L	P	L	MP	L	L	Z	PM
478	PG				BX	BX	BX	L	P	L	MP	L	L	NB	PM
479	Z	PM	PG		BX	BX	BX	MP	L	MP	P	MP	L	Z	PB
480	NM	NG			BX	BX	BX	MP	L	MP	P	MP	L	PB	Z
481					BX	BX	BX	MP	L	MP	P	P	L	PM	NB
482					BX	BX	BX	MP	L	MP	P	L	L	PM	NB
483					BX	BX	BX	P	L	MP	P	MP	L	NB	PM
484	Z	PM	PG		BX	BX	BX	P	L	MP	P	P	L	NB	PM
485	NM	NG			BX	BX	BX	P	L	MP	P	P	L	PM	NB
486	Z	PM	PG		BX	BX	BX	P	L	MP	P	L	L	NB	PM
487	PM	PG			BX	BX	BX	P	L	MP	P	L	L	NB	PM
488	NM	NG			BX	BX	BX	P	L	MP	P	L	L	PM	NB
489					BX	BX	BX	L	L	MP	P	MP	L	NB	PM
490	Z	PM	PG		BX	BX	BX	L	L	MP	P	P	L	NB	PM
491	NM	NG			BX	BX	BX	L	L	MP	P	P	L	PM	NB
492	NM	NG			BX	BX	BX	L	L	MP	P	P	L	PM	NB



493	Z	PM	PG		BX	BX	BX	L	L	MP	P	L	L	NB	PM
494	PM	PG			BX	BX	BX	L	L	MP	P	L	L	NB	PM
495	NM	NG			BX	BX	BX	L	L	MP	P	L	L	PM	NB
496	NM	NG			BX	BX	BX	L	L	MP	P	L	L	PM	NB
497					BX	BX	BX	MP	L	P	MP	MP	L	Z	PB
498					BX	BX	BX	MP	L	P	MP	P	L	Z	PB
499					BX	BX	BX	MP	L	P	MP	L	L	Z	PB
500	Z	NM	NG	PM	BX	BX	BX	P	L	P	MP	MP	L	Z	PM
501	PG				BX	BX	BX	P	L	P	MP	MP	L	NB	PM
502	Z	NM	NG	PM	BX	BX	BX	P	L	P	MP	P	L	Z	PM
503	PG				BX	BX	BX	P	L	P	MP	P	L	NB	PM
504	Z	NM	NG	PM	BX	BX	BX	P	L	P	MP	L	L	Z	PM
505	PG				BX	BX	BX	P	L	P	MP	L	L	NB	PM
506	Z	NM	NG	PM	BX	BX	BX	L	L	P	MP	MP	L	Z	PM
507	PG				BX	BX	BX	L	L	P	MP	MP	L	NB	PM
508	Z	NM	NG	PM	BX	BX	BX	L	L	P	MP	P	L	Z	PM
509	PG				BX	BX	BX	L	L	P	MP	P	L	NB	PM
510	Z	NM	NG	PM	BX	BX	BX	L	L	P	MP	L	L	Z	PM
511	PG				BX	BX	BX	L	L	P	MP	L	L	NB	PM
512					AL	BX	BX		nMP	nMP	nMP		L	PM	PB
513					AL	BX	AL						L	PB	PB
514					AL	AL								Z	Z
515						AL	AL							Z	Z
516					BX	BX	AL		nMP	nMP	nMP		L	PB	PM
517					BX	AL	BX	nMP				MP	L	Z	PB

## 8.2

## Trecho da tabela de inferência do modelo real

Nº regra	Ang.	Ang.	Ang.	Ang.	Sensor Alt. Esq.	Sensor Alt. Frente	Sensor Alt. Dir.	BumperEsq.	BumperFrente	BumperDir.	Sensor Lateral Esq.	Sensor Frente Esq.	Sensor Frente	Sensor Frente Dir.	Sensor Lateral Dir.	Dist. Obj.	MotorE	MotorD
512					BX	BX	BX	ON	OFF	OFF						L	Z	NB
513								ON		ON							Z	Z
514					BX	BX	BX	ON	ON	OFF						L	Z	NB
515					BX	BX	BX	OFF	ON	ON						L	NB	Z
516					BX	BX	BX	OFF	OFF	ON						L	NB	Z
517					BX	BX	BX	OFF	ON	OFF						L	NB	Z
518					AL	BX	BX	OFF	OFF	OFF		nMP	nMP	nMP		L	PM	PB
519					AL	BX	BX	OFF	OFF	OFF		nMP	nMP	MP		L	PB	PB
520					AL	BX	BX	OFF	OFF	OFF		MP	nMP	nMP	MP	L	PB	Z
521					AL	BX	BX	OFF	OFF	OFF		MP	nMP	nMP	nMP	L	PM	Z
522					AL	BX	BX	OFF	OFF	OFF			MP		nMP	L	PM	NB
523					AL	BX	BX	OFF	OFF	OFF			MP	nMP	MP	L	PB	Z
524					AL	BX	AL	OFF	OFF	OFF						L	PB	PB
525					AL	AL											Z	Z
526						AL	AL										Z	Z
527					BX	BX	AL	OFF	OFF	OFF		nMP	nMP	nMP		L	PB	PM
528					BX	BX	AL	OFF	OFF	OFF		MP	nMP	nMP		L	PB	PB
529					BX	BX	AL	OFF	OFF	OFF	MP	nMP	nMP	MP		L	Z	PB
530					BX	BX	AL	OFF	OFF	OFF	nMP	nMP	nMP	MP		L	Z	PM
531					BX	BX	AL	OFF	OFF	OFF	nMP		MP			L	NB	PM
532					BX	BX	AL	OFF	OFF	OFF	MP	nMP	MP			L	Z	PB
533					BX	AL	BX	OFF	OFF	OFF	nMP				MP	L	Z	PB
534					BX	AL	BX	OFF	OFF	OFF	MP				nMP	L	PB	Z
535	Z	PM	PG		BX	AL	BX	OFF	OFF	OFF	nMP				nMP	L	NB	PB
536	NM	NG			BX	AL	BX	OFF	OFF	OFF	nMP				nMP	L	PB	NB