



Giovanny Alberto Meneses Arboleda

**Projeto e Controle de Movimento de uma Bengala Robótica
Omnidirecional para Assistência de Deficiências Motoras**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica do Departamento de Engenharia Mecânica da PUC-Rio. Como requisito parcial para obtenção do título de Mestre em engenharia mecânica

Orientador: Prof. Marco Antonio Meggiolaro

Rio de Janeiro
setembro de 2015



Giovanny Alberto Meneses Arboleda

**Projeto e Controle de Movimento de uma Bengala Robótica
Omnidirecional para Assistência de Deficiências Motoras**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Mecânica da PUC-Rio. Aprovado pela comissão Examinadora abaixo assinada.

Prof. Marco Antonio Meggiolaro

Orientador

Departamento de Engenharia Mecânica-PUC-Rio

Prof. Mauro Speranza Neto

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Karla Tereza Figueireiro Leite

Departamento de Engenharia elétrica – PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 30 de setembro de 2015

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

Giovanny Alberto Meneses Arboleda

Autor graduado na Universidade Santo Tomas de Aquino. Bucaramanga,Colombia em Engenharia Mecatrônica.

Ficha Catalográfica

Arboleda, Giovanny Alberto Meneses

Projeto e controle de movimento de uma bengala robótica omnidirecional para assistência de deficiências motoras / Giovanny Alberto Meneses Arboleda ; orientador: Marco Antonio Meggiolaro. – 2016.

129 f. ; 30 cm

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

*À minha mãe, essa pessoa que me oferece amor incondicional.
Ao meu pai, exemplo de calma e sabedoria
Com todo o amor do meu mundo.*

Agradecimentos

Primeiramente agradeço à minha mãe que com sua doçura deu para mim a força necessária para continuar cada dia, meu pai que, não importando o que, está sempre cuidando de mim. Agradeço ao Alexis que no momento justo deu a última ajuda, a Capes Pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado e ao orientador Marco Meggiolaro.

Resumo

Meneses Arboleda, Giovanni Alberto; Meggiolaro, Marco Antonio. **Projeto e Controle de Movimento de uma Bengala Robótica Omnidirecional para Assistência de Deficiências Motoras**. Rio de Janeiro, 2015. 129p. Dissertação de Mestrado - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

A robótica já é parte importante da vida cotidiana, em especial na grande ajuda que pode promover para melhorar a qualidade de vida, mostrando-se como uma ótima opção, por exemplo, para a reabilitação física no corpo humano. O presente trabalho apresenta um projeto e um estudo de controle de movimento de uma plataforma robótica - uma bengala - controlada por meio de sensores de força nela localizados, fazendo com que o registro dos movimentos do usuário seja feito de forma não-invasiva. O protótipo se desloca por meio de três rodas omnidirecionais, acionadas por motores elétricos de corrente contínua. Estes últimos têm facilidade no controle e são de baixo custo. A bengala é fabricada em alumínio para facilitar o seu transporte devido à baixa densidade desse material, além de apresentar uma boa usinabilidade, facilitando a sua fabricação. O sistema eletrônico consiste de três etapas: (i) circuito de aquisição de ponte de *Wheatstone* para os extensômetros; (ii) amplificação e filtragem feitas com amplificadores de instrumentação e filtros passa-baixa *Butterworth*; e (iii) processamento e controle, implementado em dois microcontroladores PIC. O controle da bengala é baseado em admitância em paralelo com um controle PI linear, o qual pretende promover ao usuário uma sensação de naturalidade ao caminhar, sem esforços adicionais significativos e com rápida resposta. Em particular, o sistema pretende detectar situações de queda iminente do usuário, cenário não incomum no uso por idosos.

Palavras-Chave

Bengala robótica; Controle de admitância; detecção de queda; Redes neurais

Abstract

Meneses Arboleda, Giovanni Alberto; Meggiolaro, Marco Antonio (Advisor). **Design and Motion Control of an Omnidirectional Robotic Walking Cane for Assistance of Motor Disabilities**. Rio de Janeiro, 2016. 129p. MSc. Dissertation – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Robotics already is an important part of modern daily routine, with a quite unlimited potential for the improvement of life quality. For instance, robotics can be a very attractive technology for physical rehabilitation of the human body. The present work presents a design proposal and study of the stability control of an omnidirectional robotic walking cane for assistance of motor disabilities. Non-invasive force sensors are used to register the user's motions and to control the robotic cane. Three omnidirectional wheels, each of them driven by a continuous electrical current motor, move the prototype in all planar directions without the need for turning. The chosen electrical motors are characterized by their easy control and low cost. The stick is fabricated in aluminum, a low-density material with good machinability, in order to both facilitate the user in transporting the cane and to ease the manufacturing process. The electronic system is comprised of three stages: (i) a Wheatstone bridge circuit for the acquisition of strain-gage signals for force and torque sensing; (ii) amplification and filtering with instrumentation amplifiers and Butterworth-type low-pass filters; and (iii) processing and control, implemented on two PIC microcontrollers. The control of the robotics support is performed both by an admittance-based approach in parallel with a linear PI control. The quick response of this integrated control does not demand extra efforts from the user, thus providing a more natural sensation while walking. In particular, the system intends to detect whether the user is in the imminence of falling over, a likely scenario in eldercare.

Keywords

Robotic Cane; Admittance control; Fall Detection; Neural Network.

“Todos temos necessidades, mas como a maioria de nós geralmente não sabemos do que.”

Sr. Spock

Sumário

1. Introdução	20
1.1 Motivação	20
1.2 Revisão Bibliográfica	21
1.3 Objetivos do Trabalho	25
1.4 Estrutura da Dissertação	26
2. Fundamentos Teóricos	27
2.1 Modelo mecânico	27
2.1.1 Modelo cinemático de uma plataforma omnidirecional de 3 rodas	27
2.1.2 Modelo dinâmico de uma plataforma omnidirecional	31
2.2 Aquisição e Processamento	37
2.2.1 Decodificação das forças.	37
2.2.2 Extensômetro de Resistência Elétrica (ERE):	40
2.2.3 Ponte de Wheatstone.	43
2.2.4 Amplificação	46
2.2.5 Filtragem	47
2.3 Redes neurais.	49
2.4 Controle baseado em admitância	50
2.5 Controle PID	52
2.5.1 Controle Proporcional (P).	53
2.5.2 Controle Integral (I).	53

2.5.3 Controle Derivativo (D).	53
2.5.4 Método do lugar das raízes	54
2.5.5. Implementação no microcontrolador PIC	55
3. Projeto, simulação e controle do protótipo experimental	58
3.1 Projeto mecânico	58
3.2 Simulação do modelo cinemático	65
3.3 Simulação do modelo dinâmico	68
3.4 Controle	71
3.5 Controle de alto nível ou supervisor	72
3.6 Controle de baixo nível	76
4. Experimentos e Resultados	81
4.1. Modelagem do motor.	81
4.2 Controle baseado em admitância.	84
4.3 Base de dados	85
4.4. Controle de baixo nível.	92
4.5. Controle de alto nível ou supervisor	94
4.5.1 Primeira fase	96
4.5.2 Segunda fase	98
5. Conclusões e Recomendações	101
6. Referências Bibliográficas	104
Apêndice A	106
Apêndice B – Código computacional	114
B.1. Modelo cinemático - script.	114
B.2. Modelo dinâmico - script.	114

B.3. Interface da aquisição.	115
B.4. Treinamento da rede - função principal.	120
B.4.1. Treinamento rede funções secundárias.	122

Lista de Figuras

Figura 1 Andador robótico Walking Helper [1].....	22
Figura 2 SmartWalker [2]	22
Figura 3 Andador robótico I-go [3].....	22
Figura 4 Bengala robótica PAMM [4].....	23
Figura 5 Bengala com ITD [5]	24
Figura 6 Bengala omnidirecional [6].	24
Figura 7 Protótipo Intelligent Cane Robot [7].....	25
Figura 8 Plataforma omnidirecional de três rodas (vista superior)	27
Figura 9 Diagrama de forças da bengala	32
Figura 10 Vista superior da plataforma	34
Figura 11 Forças agindo em cada roda omnidirecional.....	35
Figura 12 Posição dos strain-gages na plataforma.	38
Figura 13 Planos locais.	40
Figura 14 Strain gage [8].....	41
Figura 15 Deformação do corpo de prova.....	41
Figura 16 Área transversal de um cilindro.....	42
Figura 17 Ponte de Wheatstone.....	43
Figura 18 Balanço de ponte de Wheatstone	45
Figura 19. Amplificador instrumental	46
Figura 20 Filtro Butterworth de 2º ordem.....	48
Figura 21 Resposta em Frequência do filtro adotado.....	49
Figura 22 Diagrama de um neurônio artificial.....	50
Figura 23 Diagrama de controle baseado em admitância.	51
Figura 24 Modelo massa-amortecedor utilizado.....	52
Figura 25 Sistema de segunda ordem	55

Figura 26 Diagrama de implementação de PID no PIC.....	56
Figura 27 Bengala de 4 pontas padrão, e versão cortada.....	59
Figura 28 Planos do corpo	60
Figura 29 Medições antropomórficas médias [21].....	60
Figura 30 Desenho da base de alumínio para os níveis 1 e 2.....	62
Figura 31 Desenho da base de alumínio para o nível 3.	62
Figura 32 Disposição dos strain gages na base da bengala.	63
Figura 33 Modelagem da plataforma omnidirecional.....	64
Figura 34 Protótipo completo da bengala robótica.	65
Figura 35 Função do modelo cinemático no Simulink.	66
Figura 36 Simulação do modelo cinemático da bengala robótica para os valores da Tabela 3.	67
Figura 37 Entradas e saídas do modelo cinemático da bengala.....	67
Figura 38 Função do modelo dinâmico no Simulink.....	68
Figura 39 Entradas do modelo dinâmico da bengala	69
Figura 40 Saídas do modelo dinâmico da bengala	70
Figura 41 Velocidades angulares das rodas.	70
Figura 42 Esquema geral de controle	71
Figura 43 Intenções do usuário	72
Figura 44 Resposta ao degrau de cada sistema motor-roda.	77
Figura 45 Lugar das raízes de cada sistema motor-roda.	77
Figura 46 Lugar das raízes junto com o compensador.	78
Figura 47 Resposta do controle como o lugar das raízes e tuning tool.	79
Figura 48 Resposta do controle como o lugar das raízes e tuning tool.	79
Figura 49 Diagrama de controle.....	80
Figura 50. Sistema para a modelagem dos motores das rodas.	81

Figura 51 Corrente do motor para resposta em degrau.	82
Figura 52 Rotação do motor para a resposta ao degrau.	83
Figura 53 Entrada e saída do controle baseado em admitância.	84
Figura 54 Testes com bengala.	85
Figura 55 Teste de caminhada para trás.	87
Figura 56 Teste de caminhada para a direita.	88
Figura 57 Teste de caminhada para frente.	88
Figura 58 Teste de caminhada para a esquerda.	89
Figura 59 Teste de queda para trás.	90
Figura 60 Teste de queda para a direita.	90
Figura 61 Teste de queda para frente.	91
Figura 62 Teste de queda para a esquerda.	91
Figura 63 Comparação de corrente entre o motor real e o simulado.	92
Figura 64 Comparação de erro de torque entre o motor real e o simulado.	93
Figura 65 Comparação experimental da resposta ao degrau do motor com o controlador e em malha aberta.	94
Figura 66. Diagrama de fluxo do controle de alto nível.	96
Figura 67. Gráfico com o desempenho da segunda fase do sistema supervisor.	100
Figura 68 Primeiro circuito elétrico (primeira camada).	107
Figura 69 Segundo circuito elétrico (segunda e terceira camadas).	108
Figura 70 Terceiro circuito elétrico (quarta camada).	109
Figura 71 Interface desenvolvida para aquisição de dados.	113

Lista de Tabelas

Tabela 1	Plano local de esforços	39
Tabela 2	Especificações do filtro Butterworth.....	47
Tabela 3	Propriedades físicas e mecânicas do alumínio ASTM 5052F ...	61
Tabela 4	Dados do modelo cinemático.	66
Tabela 5	Intenções do usuário	72
Tabela 6	Arquitetura do primeiro modelo	74
Tabela 7	Arquitetura do terceiro modelo	74
Tabela 8.	Codificação dos movimentos.....	75
Tabela 9	Arquitetura do terceiro modelo	75
Tabela 10.	Constantes identificadas do motor	83
Tabela 11.	Resumo da primeira fase do sistema supervisor.....	98
Tabela 12.	Desempenho da segunda fase do sistema supervisor.	99
Tabela 13	Elementos do circuito 1	107
Tabela 14	Elementos das camadas 2 e 3	108
Tabela 15	Elementos da camada 4.....	110
Tabela 16	Elementos da DAQ.....	112
Tabela 17	Componente mecânicos.....	113

Lista de Abreviações

ITD: Intentional direction, intenção do usuário.

ERE: Extensômetros de resistência elétrica.

USB: Universal Serial Bus.

PIC: Peripheral Interface Controller.

OPAM/AO: Operational Amplifier.

Lista de Variáveis

a : Aceleração linear/ constante.

A : Área transversal da bengala.

A_1 : Área Externa da bengala.

A_2 : Área interna da bengala.

D_i : Direção de avanço da roda i .

D : Diâmetro da bengala.

E : Módulo de Young.

F_x : Força de usuário no eixo x .

F_y : Força de usuário no eixo y .

F_z : Força de usuário no eixo z .

F_u : Vetor de forças do usuário.

F_{imx} : Força gerada pelo momento do usuário no eixo x na roda i .

F_{imy} : Força gerada pelo momento do usuário no eixo y na roda i .

F_{imz} : Força gerada pelo momento do usuário no eixo z na roda i .

F_{ri} : Força de atrito na roda i no sentido de avanço.

F_{Ai} : Força de atrito na roda i no sentido axial.

F_A : Vetor de forças de atrito das rodas no sentido axial.

f_d : Vetor de forças de atrito em sentido da direção de avanço e axial da roda.

g : Aceleração da gravidade.

I_A : Momento de inércia da área transversal.

I : Momento de inercia da bengala.

K : é fator de calibração do ERE.

L_i : Comprimento inicial do extensômetro.

L_f : Comprimento final do extensômetro.

L : Comprimento do braço desde o centro de massa da base até o centro de massa da roda.

m : Massa.

M : Matriz de inercia da bengala.

M_x : Momento de usuário no eixo x .

M_y : Momento de usuário no eixo y .

M_z : Momento de usuário no eixo z .

M_u : Vetor de momentos de usuário.

N : Força normal.

N_i : Força normal da roda i .

p : Força axial.

\vec{P}_{0i} : Vetor posição do centro de massa da roda i em relação ao ponto 0.

P_o : Vetor posição do centro de massa da base em relação com a origem do marco de referência.

\ddot{q}_c : Vetor de acelerações desejadas.

$R(\theta)$: Matriz de rotação.

R_i : Resistência inicial do extensômetro.

R_f : Resistência final do extensômetro.

ΔR : Variação da resistência do extensômetro.

r : Raio da roda.

r_i : Vetor posição da roda i até a origem do marco de referência.

t : Espessura da bengala.

v_i : Vetor velocidade da roda i até a origem do marco de referência.

\dot{x} : Velocidade no eixo x

\dot{y} : Velocidade no eixo y .

z : Vetor de velocidades angulares das rodas.

ε : Deformação do extensômetro.

$\dot{\theta}$: Velocidade angular no eixo z .

θ : Ângulo entre o eixo x do marco de referência inercia e $\overrightarrow{P_{01}}$.

$\ddot{\theta}_c$: Aceleração angular da bengala.

μ_m : Coeficiente de atrito.

τ_i : Torque da roda i .

τ : Vetor de torque das rodas.

ω_i : Velocidade angular da roda i .

1. Introdução

1.1 Motivação

Há um número crescente de idosos e pacientes com problemas de mobilidade, com debilidade muscular ou óssea nas pernas, ou ainda com doenças neurológicas. Nesses casos, pode ser perdida a capacidade de se deslocar em diferentes ambientes, como em suas próprias residências e lugares públicos, diminuindo assim a qualidade de vida das pessoas que são atingidas por essas doenças, o que pode gerar problemas para si mesmos e para as pessoas ao seu redor.

Para ajudar essas pessoas, é necessário fornecer um apoio temporário para reabilitação, ou um apoio permanente com o qual possam desenvolver suas rotinas de fisioterapia e/ou trabalho diário, sem ter que ser dependentes de enfermeiros 100% do tempo. Cientistas têm criado uma quantidade considerável de dispositivos mecânicos, tais como bengalas ou andadores, que proporcionam suporte mecânico, mas com possibilidade de situações de instabilidade, nas quais esses dispositivos precisam de uma força extra do usuário. Já os dispositivos eletrônicos oferecem ajuda através de diferentes sensores, como por exemplo os ultrassônicos, que avisam os usuários sobre possíveis obstáculos para assim evitá-los.

Em uma outra classificação estão os sistemas robóticos ativos, os quais ajudam em diferentes setores de incapacidades, proporcionando uma união de diferentes dispositivos, tanto mecânicos quanto eletrônicos. A maioria desses dispositivos robóticos pode ser agrupada em cadeiras de rodas automatizadas, exoesqueletos, andadores e bengalas. Cadeiras de rodas automatizadas são muito populares, no entanto possuem restrição de acesso em degraus, e podem ser evitadas se o usuário ainda possuir alguma mobilidade nas pernas. Exoesqueletos são mais difícil no seu uso, necessitando que o usuário acople o dispositivo ao seu corpo, além de possuir um alto custo. Por outro lado, andadores robóticos necessitam que o usuário utilize ambos os braços para se apoiar, recomendado para casos mais extremos de deficiência antes da adoção

de cadeiras de rodas. Finalmente, bengalas robóticas permitem um auxílio não-invasivo de menores dimensões que os andadores, além de custo mais baixo que exoesqueletos, e permitindo que o usuário permaneça em pé, ao contrário da maioria das cadeiras de rodas automatizadas.

As pesquisas voltadas para o desenvolvimento da Bengala Robótica Omnidirecional desta dissertação se concentram em três etapas: (i) projeto da parte mecânica e elétrico do equipamento; (ii) aquisição-processamento de sinais; e (iii) implementação do controle.

O projeto mecânico compreende a seleção de materiais a serem usados na fabricação da plataforma omnidirecional. Neste caso, deseja-se um material de baixa densidade e boa usinabilidade, trabalhando este, de tal maneira que o dispositivo com seu tamanho não atrapalhe o movimento do usuário, e que possa suportar as forças aplicadas com boa estabilidade. No projeto mecânico há também a seleção dos motores e das caixas redutoras, a seleção das rodas utilizadas, a forma ideal para poder acomodar todos os equipamentos, a alimentação dos motores, a posição certa dos sensores para a aquisição dos dados das forças aplicadas na bengala pelo usuário, além da análise dos modelos cinemático e dinâmico da estrutura.

A aquisição-processamento relaciona os tipos de sensores a serem utilizados, a maneira de adquirir seus sinais, amplificação do sinal e filtragem caso necessário. Além disso, esta etapa requer escolher os elementos eletrônicos que devem ser utilizados, o desenho da placa de circuito impresso, o processo de fabricação, e as dimensões da placa.

A terceira etapa do projeto envolve o sistema de controle, que visa oferecer ao usuário uma boa interação com uma movimentação sem perturbações, com pouco esforço. E principalmente, deseja-se que o sistema seja capaz de ajudar na caminhada e detectar situações de queda do usuário.

Na próxima seção, os principais dispositivos andadores e bengalas robóticas são revisados.

1.2 Revisão Bibliográfica

Diversos sistemas robóticos foram desenvolvidos para o auxílio na locomoção de pessoas com deficiências motoras. Dentre os similares a andadores, há por exemplo o *Walking Helper* [1], o *SmartWalker* [2] e o *I-go* [3],

apresentados nas Figura 1 a 3. No entanto, esses dispositivos são de grandes dimensões, dificultando o deslocamento em lugares fechados.

Diferentemente, os dispositivo *PAMM "Helping-Hand"* [4] e *cane robot* [5], apresentados nas Figuras 4 e 5, apresentam uma maior liberdade de movimento e menor peso, facilitando assim o transporte em longas distâncias, mas diminuindo a capacidade de peso que conseguem suportar.



Figura 1 Andador robótico Walking Helper [1].

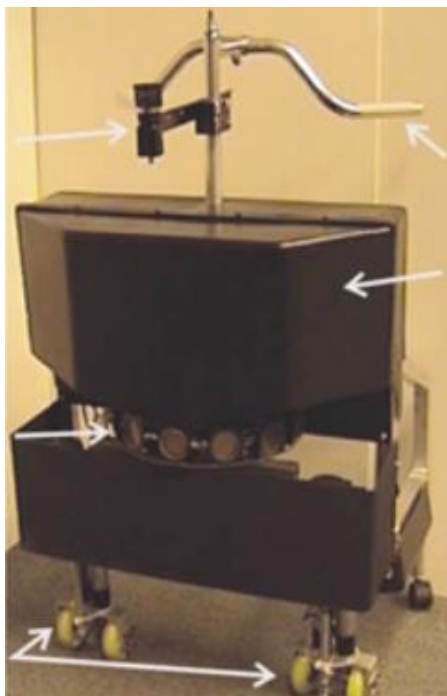


Figura 2 SmartWalker [2].



Figura 3 Andador robótico I-go [3].

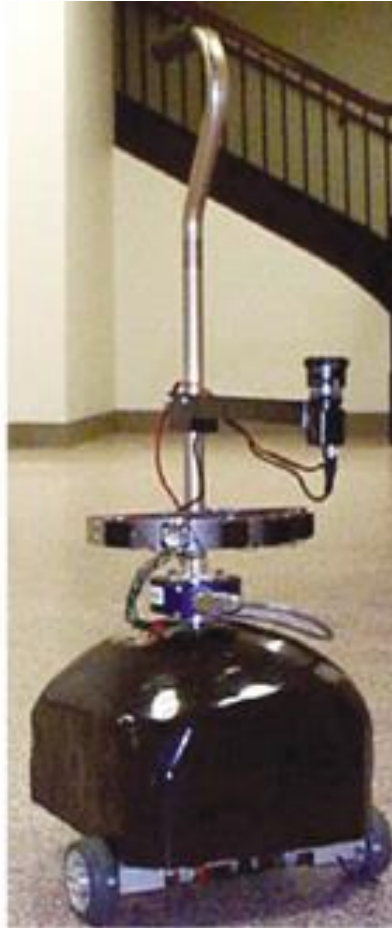


Figura 4 Bengala robótica PAMM [4].

Em [4], o robô denominado PAMM foi projetado com uma base quadrada e três rodas, das quais uma é um rodízio com encoders para avaliar a velocidade dos motores. Esse robô também tem um arranjo de sensores acústicos (ultrassom) para medir distância, identificar e assim permitir evitar objetos que possam se tornar obstáculos. Adicionalmente, PAMM tem uma câmera para que seja gerado um histórico de posições do usuário no interior de um apartamento, com a ajuda de marcadores colados no teto. Esse sistema, com massa de 15kg, consegue atingir uma velocidade máxima de 0.5 m/s.

O trabalho desenvolvido em [5], mostrado na Figura 5, é um robô com plataforma feita em alumínio com forma hexagonal e rodas omnidirecionais, que permitem translações em ambas as direções no plano sem necessidade de rotação, que é independente. O sistema, de três graus de liberdade, possui um sensor de força posicionado na parte inferior da bengala, e um sensor a laser de proximidade para saber a posição da bengala em relação ao usuário. Este projeto também utiliza o algoritmo chamado de ITD (Intentional Direction), o qual permite identificar a intenção do usuário para então acionar o controle.

O trabalho desenvolvido em [6], vide Figura 6, é uma bengala que usa três rodas omnidirecionais em uma plataforma com forma hexagonal, com sensor de força no extremo superior da bengala, dois sensores a laser para medir a distância a possíveis obstáculos, e um sensor de inclinação para medir quando a bengala estiver sobre um desnível. Este dispositivo é acionado através de um controle baseado em admitância, usando um algoritmo para saber a intenção do usuário.



Figura 5 Bengala com ITD [5].

Figura 6 Bengala omnidirecional [6].

O protótipo em [7], vide Figura 7, é feito em alumínio com forma hexagonal e rodas omnidirecionais, e conta com um grupo de sensores que compreende um arranjo a laser para medir proximidade, um sensor de torque e força de 6 eixos na parte superior da bengala, e uma junta universal na parte inferior, onde se une à base metálica com a barra da bengala para facilitar a postura do usuário no momento de utilizá-lo.

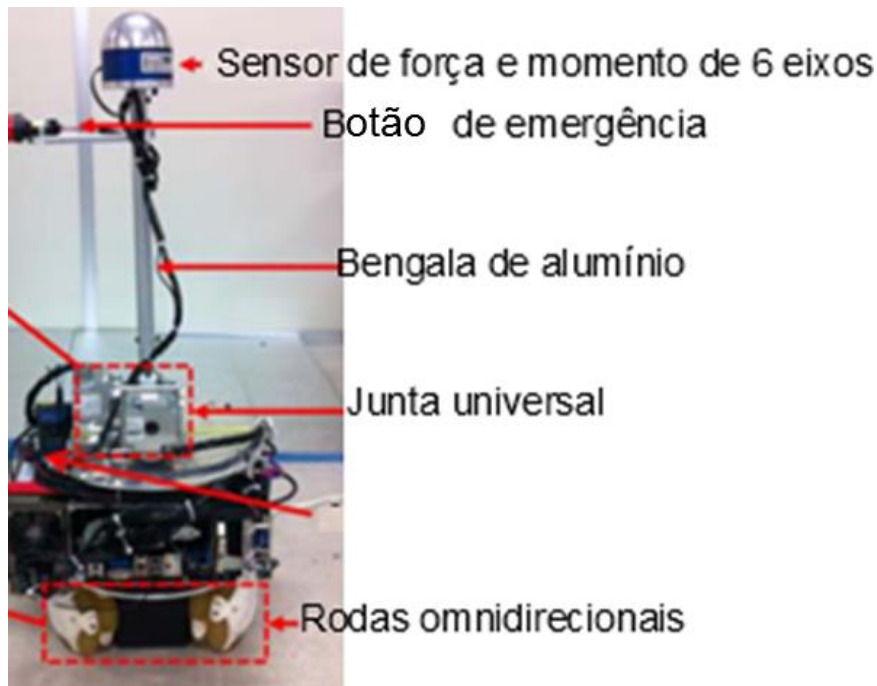


Figura 7 Protótipo Intelligent Cane Robot [7].

Indistintamente do protótipo, os desafios a serem enfrentados são os mesmos em cada sistema, ou seja, oferecer uma sensação de conforto na condução, ergonomia no uso, e dar suporte necessário ao usuário.

1.3 Objetivos do Trabalho

O objetivo principal deste trabalho é desenvolver o projeto de um protótipo funcional de uma bengala omnidirecional robótica, e implementar nela um controle de estabilidade, além de oferecer suporte para pessoas com problemas motores.

Para atingir o objetivo do controle, utiliza-se um sistema de redes neurais, o que pretende reconhecer a intenção do usuário, além de implementar um controle baseado em admitância para garantir um deslocamento suave que seja confortável para o usuário, tentando sempre implementar este com componentes de baixo custo para facilitar o processo de fabricação, principalmente nos sensores de deformação.

1.4 Estrutura da Dissertação

Este trabalho está dividido em 4 capítulos começando desde aqui. O primeiro capítulo numerado como 2 faz a revisão dos fundamentos teóricos, dividida em duas partes: a primeira parte faz uma abordagem sobre o projeto mecânico, o modelo cinemático, e o modelo dinâmico da plataforma; e a segunda parte compreende a aquisição e processamento dos sinais. Este processamento pode ser ainda dividido em 4 partes: (i) descrição do sensor para medir a deformação; (ii) circuito de adaptação (ponte de Wheatstone) e seu balanceamento; (iii) amplificação; e (iv) filtragem do sinal dos sensores de deformação.

O segundo capítulo numerado como o 3 contém as especificações físicas da bengala, tais como dimensões, material, forma da plataforma, a disposição dos motores, e as rodas a serem utilizadas. Na parte eletrônica apresentam-se as dimensões das placas de circuito e suas funções de amplificação, filtragem e processamento. Este capítulo também contém as especificações e implementação do controle baseado em admitância, e os controles de baixo e alto nível. Além disso, apresenta a aquisição e a apresentação da base de dados e o pré-processamento dos dados adquiridos. Por fim o capítulo descreve as estruturas da rede neural.

O terceiro capítulo numerado com o 4 descreve os testes realizados, a criação do controle de alto nível com redes neurais, e a verificação do controle baseado em admitância com os seus respectivos resultados.

No quarto capítulo numerado como 5 se conclui o presente trabalho, onde se enumeraram as conclusões da pesquisa e os possíveis trabalhos a serem desenvolvidos no futuro.

2. Fundamentos Teóricos

Este capítulo aborda os diferentes conceitos necessários para a construção do protótipo da bengala robótica proposta.

2.1 Modelo mecânico

2.1.1 Modelo cinemático de uma plataforma omnidirecional de 3 rodas

O modelo cinemático da bengala desenvolvida neste dissertação é baseado em uma plataforma omnidirecional com três rodas. A separação angular entre as rodas, representadas na vista superior da Figura 8 por retângulos enumerados de 1 a 3, é de 120° .

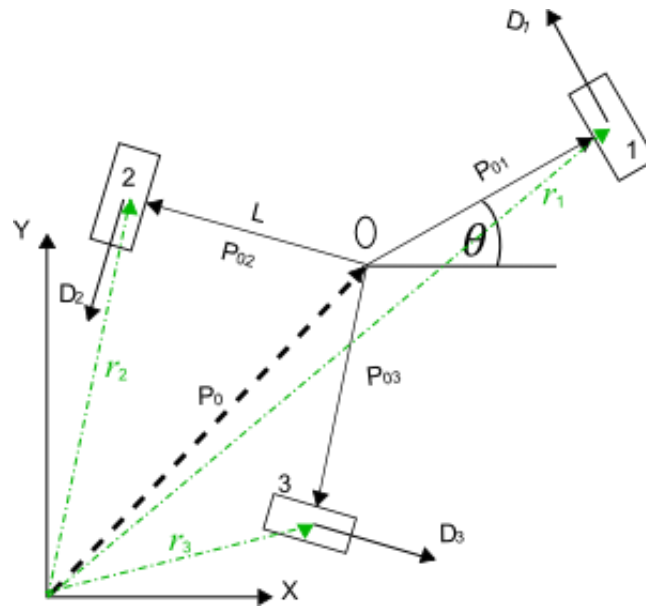


Figura 8 Plataforma omnidirecional de três rodas (vista superior)

Para encontrar a posição das rodas da base omnidirecional com respeito a seu centro de massa (ponto O), utilizou-se uma matriz de rotação em relação ao eixo z [8-9]:

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

onde a variável θ é o ângulo entre o eixo x do sistema de referência inercial e o vetor posição do centro de massa da roda 1 em relação ao ponto 0 (\vec{P}_{01}), e L é o comprimento de cada um dos braços de alavanca desde o centro de massa da base até o centro de massa da roda.

Usando o vetor \vec{P}_{01} como eixo \vec{OX} de um sistema de referência embarcado (X, Y) como mostra a Figura 8, tem-se

$$P_{01} = L \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

Com isso, pré multiplicando este valor pela matriz de rotação de Eq. (1), e com $\theta = \frac{2 \cdot \pi}{3} = 120^\circ$, tem-se que

$$P_{02} = R\left(\frac{2 \cdot \pi}{3}\right) \cdot P_{01} = \frac{L}{2} \begin{bmatrix} -1 \\ \sqrt{3} \\ 0 \end{bmatrix} \quad (3)$$

Repetindo o processo da Eq. (3) mas agora com $\theta = \frac{4 \cdot \pi}{3} = 240^\circ$, tem-se

$$P_{03} = R\left(\frac{4 \cdot \pi}{3}\right) \cdot P_{01} = -\frac{L}{2} \begin{bmatrix} 1 \\ \sqrt{3} \\ 0 \end{bmatrix} \quad (4)$$

A direção de avanço da roda D_i (Figura 8) é determinada por

$$D_i = \frac{1}{L} \cdot R(\theta) \cdot P_{oi} \quad i = 0 \dots 3 \quad (5)$$

A partir da Eq. (5), temos que a direção de avanço da roda 1 é

$$D_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (6)$$

Igualmente, tem-se a direção de avanço da roda 2

$$D_2 = -\frac{1}{2} \begin{bmatrix} \sqrt{3} \\ 1 \\ 0 \end{bmatrix} \quad (7)$$

E por último, podemos obter a direção de avanço da roda 3

$$D_3 = \frac{1}{2} \begin{bmatrix} \sqrt{3} \\ -1 \\ 0 \end{bmatrix} \quad (8)$$

A posição e a velocidade das rodas são dadas por

$$r_i = P_o + R(\theta) \cdot P_{oi} \quad (9)$$

$$v_i = \dot{P}_o + R(\theta) \cdot \dot{P}_{oi} \quad (10)$$

onde $P_o = [x \ y]^T$ é o vetor posição do centro de massa da base em relação ao sistema referencial inercial. A velocidade de cada uma das rodas em relação ao ponto 0, v_d , é dada por

$$v_d = v_i^T \cdot (R(\theta) \cdot D_i) \quad (11)$$

Substituindo Eq. (10) em Eq. (11), tem-se

$$v_{di} = \dot{P}_0 \cdot R(\theta) \cdot D_i + P_{0i}^T \cdot \dot{R}(\theta)^T \cdot R(\theta) \cdot D_i. \quad (12)$$

onde o segundo termo da equação é a velocidade tangencial do robô

$$P_{0i}^T \cdot \dot{R}(\theta)^T \cdot R(\theta) \cdot D_i = L \cdot \dot{\theta}. \quad (13)$$

A partir das Eqs. (12) e (13), a velocidade da roda 1 é dada por

$$v_{d1} = -\dot{x} \cdot \sin(\theta) + \dot{y} \cdot \cos(\theta) + L \cdot \dot{\theta} \quad (14)$$

Da mesma forma, pode-se encontrar a velocidade da roda 2

$$v_{d2} = \dot{x} \cdot \frac{\sin(\theta) - \sqrt{3} \cdot \cos(\theta)}{2} - \dot{y} \cdot \frac{\cos(\theta) + \sqrt{3} \cdot \sin(\theta)}{2} + L \cdot \dot{\theta} \quad (15)$$

e a velocidade da roda 3

$$v_{d3} = \dot{x} \cdot \frac{\sin(\theta) + \sqrt{3} \cdot \cos(\theta)}{2} - \dot{y} \cdot \frac{\cos(\theta) - \sqrt{3} \cdot \sin(\theta)}{2} + L \cdot \dot{\theta} \quad (16)$$

Logo, rescrevendo as Eqs (14-16) em forma matricial, obtém-se

$$\begin{bmatrix} v_{d1} \\ v_{d2} \\ v_{d3} \end{bmatrix} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ \frac{\sin(\theta) - \sqrt{3} \cdot \cos(\theta)}{2} & -\frac{\cos(\theta) + \sqrt{3} \cdot \sin(\theta)}{2} & L \\ \frac{\sin(\theta) + \sqrt{3} \cdot \cos(\theta)}{2} & -\frac{\cos(\theta) - \sqrt{3} \cdot \sin(\theta)}{2} & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (17)$$

Assumindo que não há deslizamento das rodas, sabe-se que a velocidade angular de cada roda ω_i é dada por

$$v_{di} = \omega_i \cdot r \quad (18)$$

onde r é o seu raio. Com isso, pode-se rescrever a Eq. (17) como

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ \frac{\sin(\theta) - \sqrt{3} \cdot \cos(\theta)}{2} & -\frac{\cos(\theta) + \sqrt{3} \cdot \sin(\theta)}{2} & L \\ \frac{\sin(\theta) + \sqrt{3} \cdot \cos(\theta)}{2} & -\frac{\cos(\theta) - \sqrt{3} \cdot \sin(\theta)}{2} & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (19)$$

ou

$$z = \frac{1}{r} \cdot H^{-1} \cdot \dot{q}_c \quad (20)$$

onde $z = [\omega_1 \ \omega_2 \ \omega_3]^T$ é o vetor de velocidades angulares e $\dot{q}_c = [\dot{x} \ \dot{y} \ \dot{\theta}]^T$ é o vetor de velocidades generalizadas.

2.1.2 Modelo dinâmico de uma plataforma omnidirecional

Objetivando o desenvolvimento do modelo dinâmico da plataforma, assume-se que está se desloca em uma superfície lisa, plana, sem inclinação, na qual os motores estarão livres para se movimentar. Na Figura 9 tem-se o diagrama de forças atuantes na plataforma omnidirecional, onde se mostram as forças de usuário ($F_x, F_y, F_z, M_x, M_y, M_z$) que agem na parte superior na alça da bengala, e as reações das rodas com o chão.

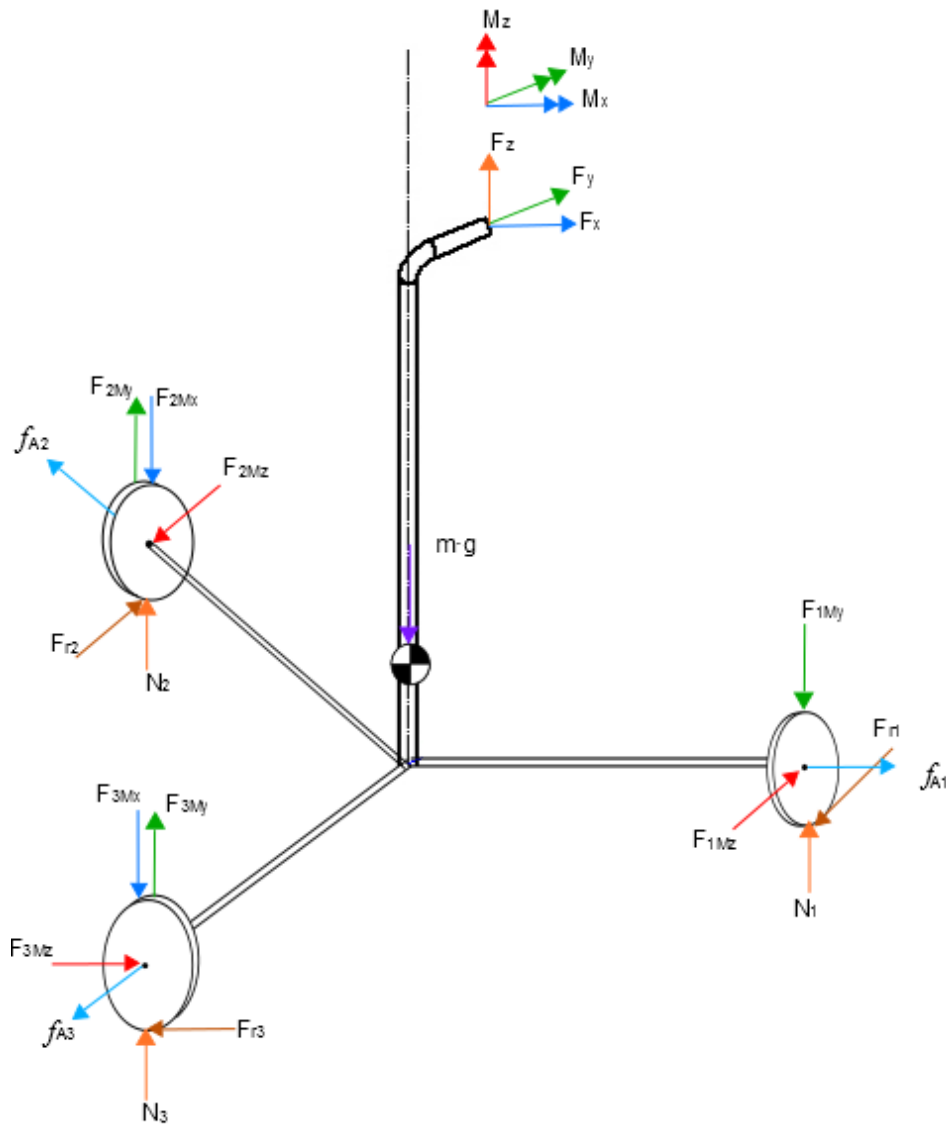


Figura 9 Diagrama de forças da bengala

De acordo como a segunda lei de Newton, podem-se escrever as equações de equilíbrio de forças como [10]

$$\sum F = m \cdot a \quad (21)$$

Na direção z, tem-se que

$$F_z + m \cdot g = N \quad (22)$$

onde m é a massa da bengala, g é a aceleração da gravidade, e N é a força normal da bengala. Assumindo que esta força é suportada pelas 3 rodas omnidirecionais, então pode-se dizer que

$$N = N_1 + N_2 + N_3 \quad (23)$$

Contudo, a força normal N não depende unicamente do peso da bengala, mas também da força exercida pelo usuário, logo

$$N = \frac{(F_{zi} + m \cdot g)}{3} \quad (24)$$

onde a força F_{zi} é a componente da força do usuário no eixo z atuante na roda i .

Então, pelo somatório de forças no plano $x - y$, tem-se que

$$\sum_{i=1}^3 (F_i - F_{ri}) \cdot R(\theta) \cdot D_i - F_A + F_u = m \cdot P_c, \quad (25)$$

onde F_i é a força exercida pela roda i , e F_{ri} é a força de atrito na direção do avanço, descrita como

$$F_{ri} \leq \mu_m \cdot N_i \quad i=1...3 \quad (26)$$

e

$$F_u = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \quad (27)$$

A força do usuário nos eixos x e y é mostrado na Figura 10.

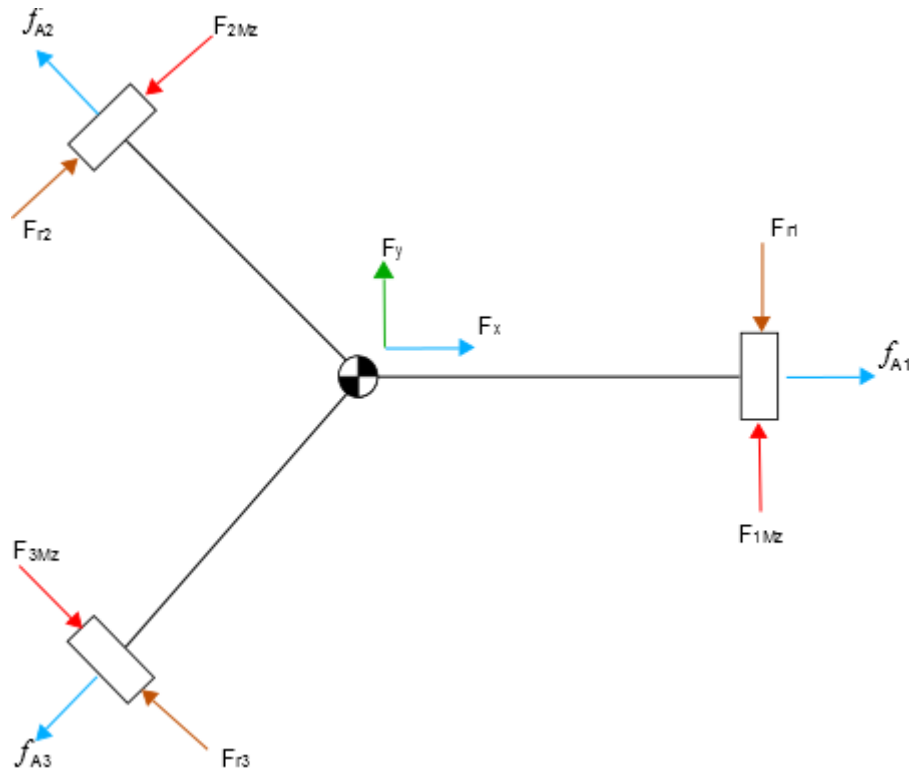


Figura 10 Vista superior da plataforma

Agora, faz-se o somatório de momentos

$$\sum \tau = I \cdot \ddot{\theta}_c \quad (28)$$

Então, tem-se

$$L \cdot \sum_{i=1}^3 (F_i - F_{ri}) + \tau_u = I \cdot \ddot{\theta}_c, \quad (29)$$

onde τ_u é o momento de força do usuário no eixo z , I é o momento de inércia da bengala na direção vertical, e $\ddot{\theta}_c$ é a aceleração angular da bengala no plano da Figura 10. A força de atrito na direção axial em forma vetorial pode ser escrita como

$$F_A = \begin{bmatrix} f_{A1} \cdot \cos(\theta) + f_{A2} \cdot \cos\left(\frac{2 \cdot \pi}{3} + \theta\right) + f_{A3} \cdot \cos\left(\frac{4 \cdot \pi}{3} + \theta\right) \\ f_{A1} \cdot \sin(\theta) + f_{A2} \cdot \sin\left(\frac{2 \cdot \pi}{3} + \theta\right) + f_{A3} \cdot \sin\left(\frac{4 \cdot \pi}{3} + \theta\right) \end{bmatrix} \quad (30)$$

Sabendo que o momento da força M_u gera uma força no sentido do avanço das rodas, então pode-se escrever que

$$F_i = \frac{\tau_i}{r} \quad (31)$$

onde r é o raio da roda, como ilustrado na Figura 11. Define-se como m a massa da base omnidirecional, é I o momento de inércia no centroide em relação à vertical.

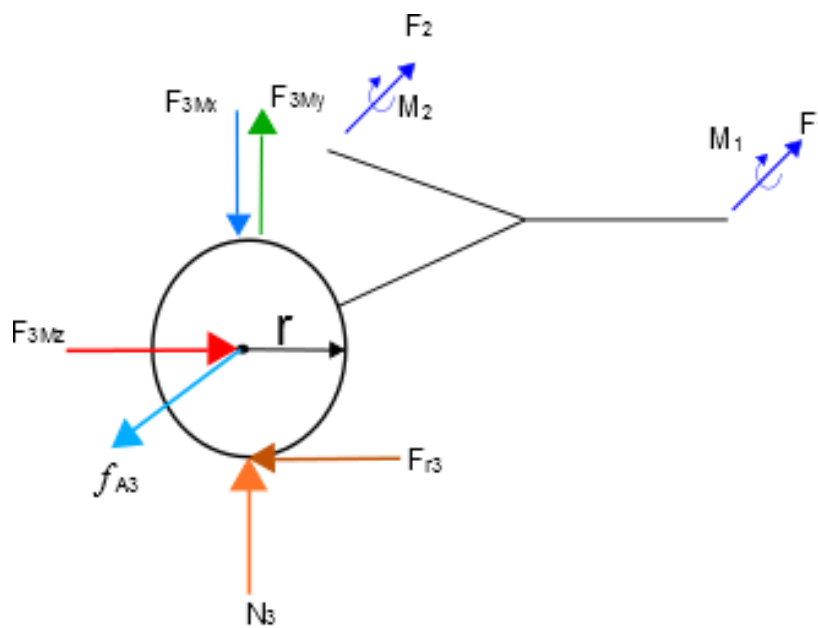


Figura 11 Forças agindo em cada roda omnidirecional

A partir das equações da dinâmica Eqs. (21) e (28), tem-se

$$M \cdot \ddot{q}_c = \frac{1}{r} \cdot H^{-T} \cdot \tau - F_d \quad (32)$$

onde: M é a matriz de inércia

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (33)$$

τ é o vetor de torque das rodas

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (34)$$

\ddot{q}_c são as acelerações desejadas em x, y e a aceleração angular desejada no eixo z

$$\ddot{q}_c = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{\theta}_c \end{bmatrix} \quad (35)$$

e F_d são as forças de atrito (na direção do avanço) e de deslizamento (direção perpendicular à roda)

$$F_d = \begin{bmatrix} f_{1d} \\ f_{2d} \\ f_{3d} \end{bmatrix} \quad (36)$$

Então, reescrevendo as equações acima, tem-se

$$\begin{aligned} f_{1d} = & -f_{r1} \cdot \sin(\theta) + f_{r2} \cdot \frac{\sin(\theta) - \sqrt{3} \cdot \cos(\theta)}{2} \\ & + f_{r3} \cdot \frac{\sin(\theta) + \sqrt{3} \cdot \cos(\theta)}{2} + f_{A1} \cdot \cos(\theta) \\ & + f_{A2} \cdot \cos\left(\frac{2\pi}{3} + \theta\right) + f_{A3} \cdot \cos\left(\frac{4\pi}{3} + \theta\right) - F_{ux} \end{aligned} \quad (37)$$

$$\begin{aligned}
f_{2d} = & f_{r1} \cdot \cos(\theta) - f_{r2} \cdot \frac{\cos(\theta) + \sqrt{3} \cdot \sin(\theta)}{2} \\
& - f_{r3} \cdot \frac{\cos(\theta) - \sqrt{3} \cdot \sin(\theta)}{2} + f_{A1} \cdot \sin(\theta) \\
& + f_{A2} \cdot \sin\left(\frac{2 \cdot \pi}{3} + \theta\right) + f_{A3} \cdot \sin\left(\frac{4 \cdot \pi}{3} + \theta\right) - F_{uy}
\end{aligned} \tag{38}$$

$$f_{3d} = L(F_{r1} + F_{r2} + F_{r3}) - \tau_u \tag{39}$$

2.2 Aquisição e Processamento

2.2.1 Decodificação das forças.

Esta decodificação é utilizada para saber a direção, a magnitude e o sentido da força que é aplicada na bengala pelo usuário, é necessário saber cada um dos esforços que atuam em cada um dos strain gages. Para isto, é necessário fazer análises separadamente destes. A Figura 12 mostra a plataforma omnidirecional na vista superior, onde no círculo central da bengala se encontram os strain gages colados, os quais são representados por pequenos retângulos denominados A, B, C e D. A plataforma está em um plano (x,y).

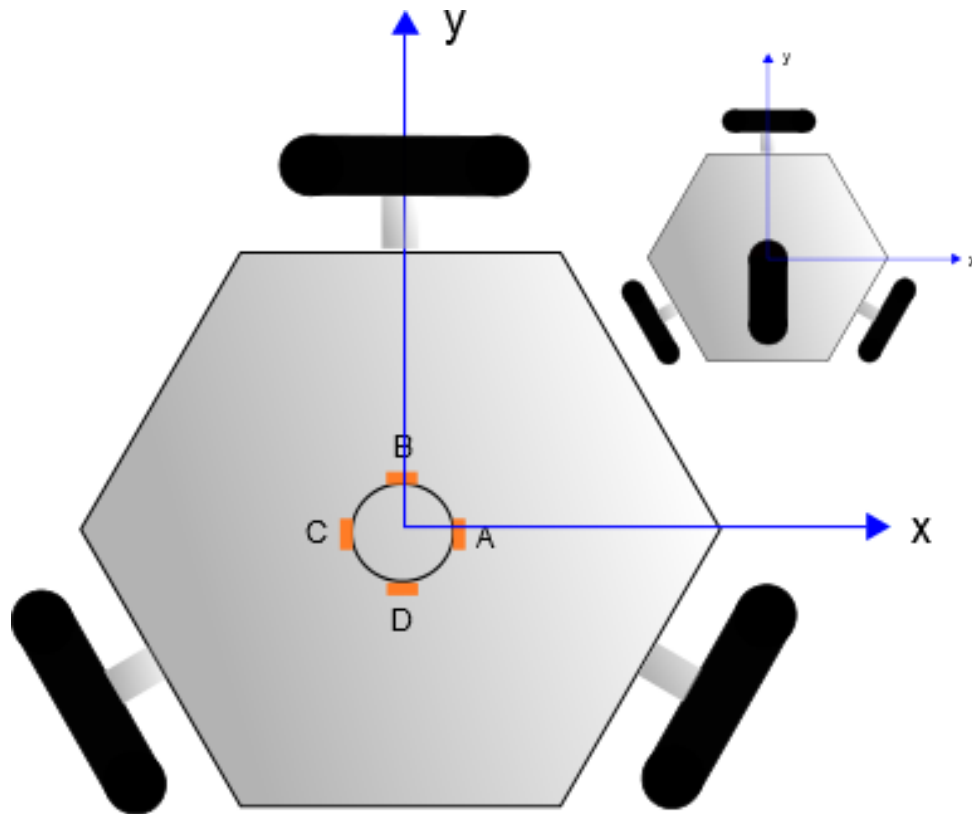


Figura 12 Posição dos strain-gages na plataforma.

Sabendo-se que o diâmetro externo D da haste da bengala é 20 mm, e sua espessura t vale 1 mm, pode-se assumir a hipótese de um tubo de paredes finas [13], já que a razão $\frac{D}{t} \geq 20$. Com isso, as equações de tensão podem ser simplificadas para

$$\sigma_x = \frac{E}{1-\mu^2}(\varepsilon_x + \mu \cdot \varepsilon_y) \quad (40)$$

com tensões normais no eixo y

$$\sigma_y = \frac{E}{1-\mu^2}(\varepsilon_y + \mu \cdot \varepsilon_x) \quad (41)$$

e tensão cisalhante causada pelo esforço cortante como

$$\tau_{xy} = \frac{E \cdot \gamma_{xy}}{2(1+\mu)} \quad (42)$$

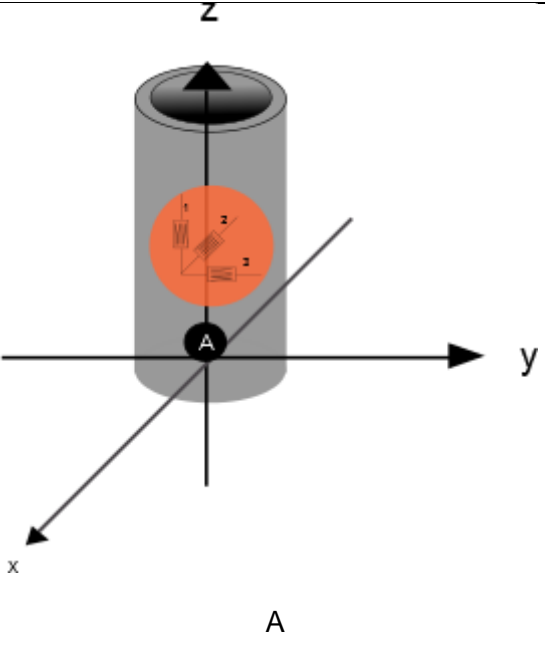
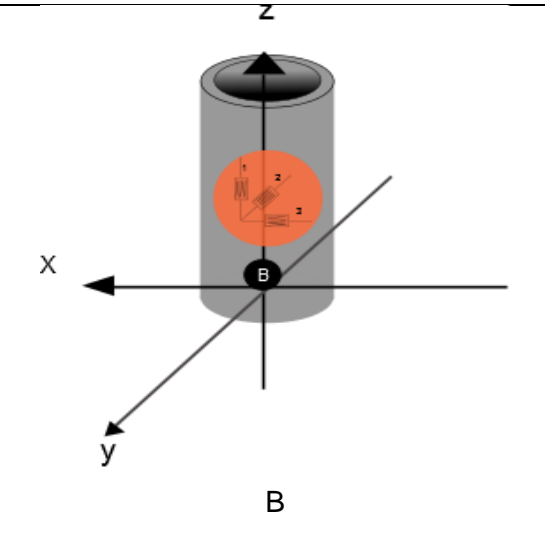
onde define-se

$$a = \frac{E}{1 - \mu^2}, \quad (43)$$

e, para uma roseta retangular com strain gages a 45° , em x e em y, obtém-se

$$\gamma_{xy} = 2 \cdot \varepsilon_{45^\circ} - \varepsilon_x - \varepsilon_y \quad (44)$$

Tabela 1 Plano local de esforços

Plano local dos esforços nos extensômetros	Equações dos esforços
	$\sigma_y = a \cdot (\varepsilon_{A3} + \mu \cdot \varepsilon_{A1})$
	$\sigma_z = a \cdot (\varepsilon_{A1} + \mu \cdot \varepsilon_{A3})$
	$\tau_{yz} = \frac{E}{2 \cdot (1 + \mu)} \cdot \gamma_{yz}$
	$\gamma_{yz} = 2 \cdot \varepsilon_{A2} - \varepsilon_{A3} - \varepsilon_{A1}$
	$\sigma_x = -a \cdot (\varepsilon_{B3} + \mu \cdot \varepsilon_{B1})$
	$\sigma_z = a \cdot (\varepsilon_{B1} + \mu \cdot \varepsilon_{B3})$
	$\tau_{xz} = \frac{E}{2 \cdot (1 + \mu)} \cdot \gamma_{xz}$
	$\gamma_{xz} = 2 \cdot \varepsilon_{B2} - \varepsilon_{B3} - \varepsilon_{B1}$

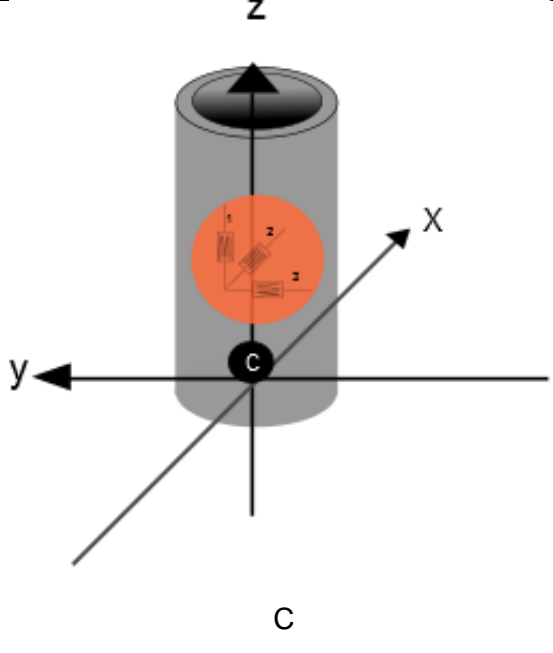
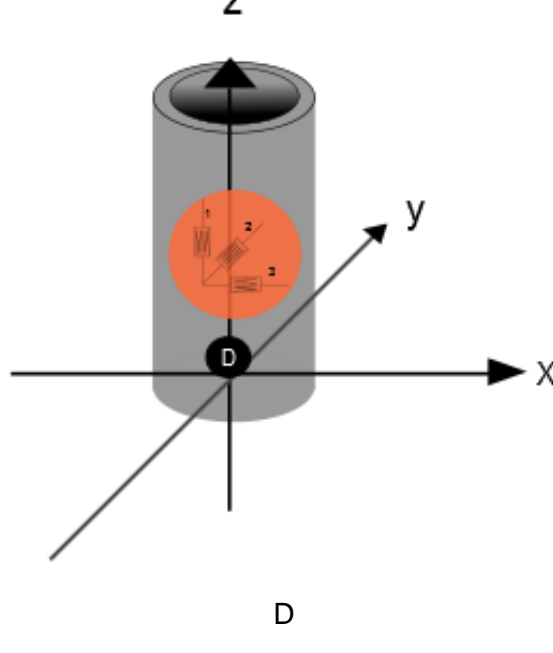
 <p style="text-align: center;">C</p>	$\sigma_y = -a \cdot (\varepsilon_{C3} + \mu \cdot \varepsilon_{C1})$
	$\sigma_z = a \cdot (\varepsilon_{C1} + \mu \cdot \varepsilon_{C3})$
	$\tau_{yz} = \frac{E}{2 \cdot (1 + \mu)} \cdot \gamma_{yz}$
	$\gamma_{yz} = 2 \cdot \varepsilon_{C2} - \varepsilon_{C3} - \varepsilon_{C1}$
 <p style="text-align: center;">D</p>	$\sigma_x = a \cdot (\varepsilon_{D3} + \mu \cdot \varepsilon_{D1})$
	$\sigma_z = a \cdot (\varepsilon_{D1} + \mu \cdot \varepsilon_{D3})$
	$\tau_{xz} = \frac{E}{2 \cdot (1 + \mu)} \cdot \gamma_{xz}$
	$\gamma_{xz} = 2 \cdot \varepsilon_{D2} - \varepsilon_{D3} - \varepsilon_{D1}$

Figura 13 Plano local

2.2.2

Extensômetro de Resistência Elétrica (ERE):

Extensômetros são sensores feitos de fios finos condutores que sofrem variações do seu comprimento, e dão como resposta uma variação da resistência elétrica, vide Figura 12. Estes sensores são colados em corpos submetidos a esforços, com o objetivo de medir deformações. Os comprimentos

do extensômetro inicial e final são nomeados como L_i e L_f , e as resistências inicial e final como R_i e R_f , vide Figura 13.

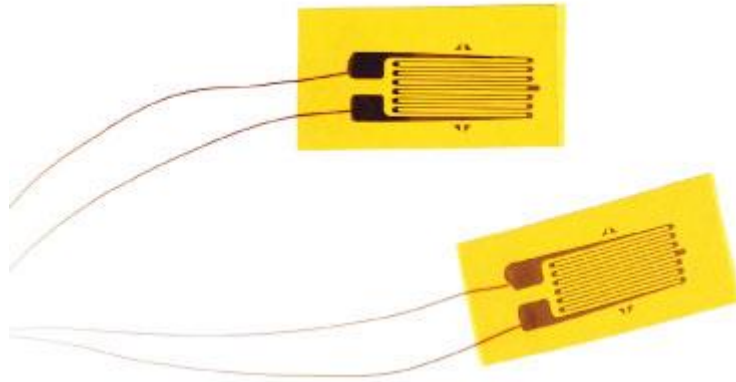


Figura 14 Strain gage [8]

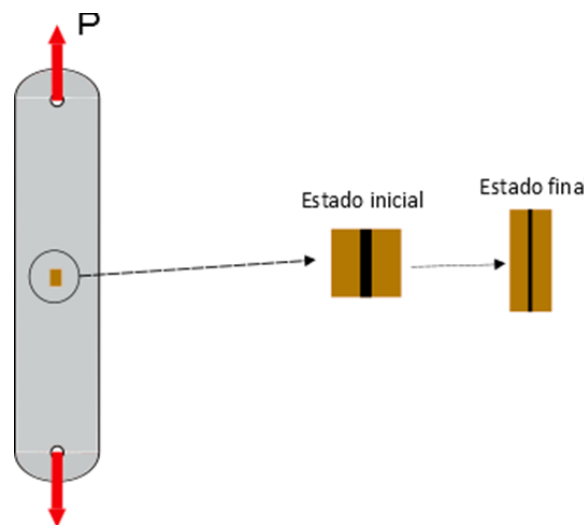


Figura 15 Deformação do corpo de prova.

Então, as variações ΔR e ΔL podem ser escritas como

$$\frac{R_f - R_i}{R_i} = K \cdot \frac{L_f - L_i}{L_i} \quad (45)$$

$$\frac{\Delta R}{R} = K \cdot \varepsilon \quad (46)$$

onde K é fator de calibração do ERE (ou o *strain gage*) dado pelo fabricante e ε é a deformação do sensor.

Para tensões uniaxiais [9], tem-se

$$\varepsilon = \frac{\sigma}{E} \quad (47)$$

onde E é o módulo de Young, σ é a tensão e ε é a deformação atuantes na posição do strain gage. Em um sistema com uma força axial, o esforço normal é dado por

$$\sigma = \frac{P}{A} \quad (48)$$

Para a área de um tubo, como a utilizada na haste da bengala, tem-se

$$A = A_1 - A_2 \quad (49)$$

onde as áreas externa e interna são A_1 e A_2 , respectivamente, vide Figura 16.

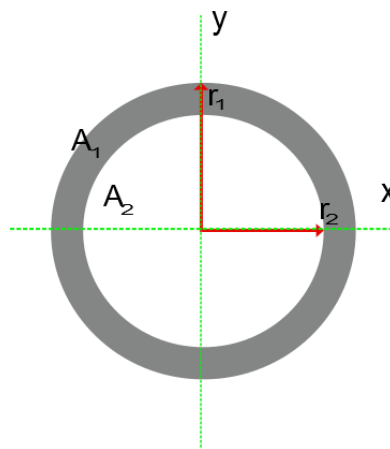


Figura 16 Área transversal de um cilindro.

Analisando a atuação de um momento fletor, tem-se

$$\sigma = \frac{M \cdot y}{I_A}, \quad (50)$$

onde o I_A é o momento de inércia de flexão (ou o segundo momento) da seção transversal com respeito ao eixo perpendicular ao momento M. Para uma seção transversal circular, tem-se

$$I_A = \frac{1}{4} \cdot \pi \cdot (r_2 - r_1)^4 \quad (51)$$

2.2.3 Ponte de Wheatstone.

A ponte de Wheatstone é uma ferramenta usada para medir variações de resistência elétrica em cada um dos braços do circuito elétrico. Chama-se de braço um conjunto de 2 resistências das 4 que compõe a ponte (ver Figura 17). As resistências $R1 + \Delta R1$ e $R2$ formam o braço número 1, $R3$ e $R4$ formam o braço número 2. Estas quatro resistências da ponte de Wheatstone podem ser de valor fixo ou podem ser os strain gages, dependendo da configuração implementada.

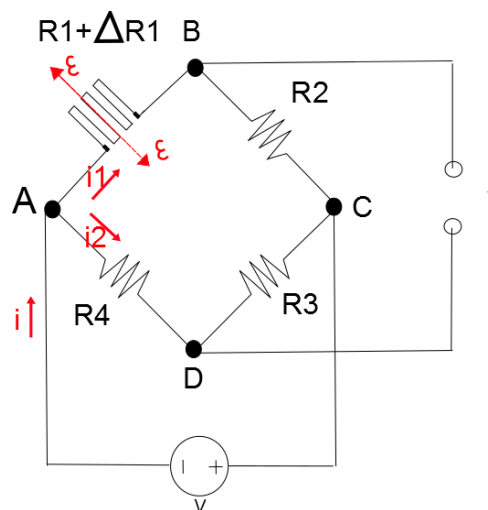


Figura 17 Ponte de Wheatstone

Para computar a variação da resistência do extensômetro, mede-se a variação de tensão na ponte nos pontos B e D. Definindo i como o fluxo elétrico que sai de uma fonte de tensão e sendo i_1 e i_2 a corrente nos braços 1 e 2, respectivamente, é possível escrever que

$$i = i_1 + i_2 \quad (52)$$

A corrente do braço 1, i_1 , não considerando a variação ΔR_1 , é dada por

$$V_{AC} = V = i_1 \cdot (R_1 + R_2) \rightarrow i_1 = \frac{V}{R_1 + R_2} \quad (53)$$

Da mesma forma, a corrente do braço 2 i_2 é

$$V_{AC} = V = i_2 \cdot (R_3 + R_4) \rightarrow i_2 = \frac{V}{R_3 + R_4} \quad (54)$$

Definindo T como a tensão entre os pontos B e D e usando as Eqs. (53) e (54), tem-se

$$T = \frac{V \cdot R_1}{R_1 + R_2} - \frac{V \cdot R_4}{R_3 + R_4}, \quad (55)$$

e, fatorando a equação acima, encontra-se que

$$T = V \cdot \frac{R_1 \cdot R_3 - R_2 \cdot R_4}{(R_1 + R_2) \cdot (R_3 + R_4)} \quad (56)$$

Quando ocorre uma variação de resistência do extensômetro ΔR_1 , ou seja

$$R_1 := R_1 + \Delta R_1, \quad (57)$$

e substituindo a Eq. (57) na Eq. (56), temos que ΔR_1 é dado por

$$\Delta R1 = \frac{V \cdot R2 \cdot R4 + T \cdot R2 \cdot (R2 + R4)}{V \cdot R3 - T \cdot (R3 + R4)} - R1 \quad (58)$$

Com isso, podem-se combinar as Eqs. (46), (47) e (48), e tem-se que

$$P = \varepsilon \cdot E \cdot A \quad (59)$$

A Eq. (59) representa a força que é aplicada no corpo onde o extensômetro foi fixado

2.2.3.1 Balanço da ponte de Wheatstone

Com o objetivo de igualar as tensões nos dois pontos *B* e *D* (Figura 17), é necessário o balanço da ponte de Wheatstone. Para tanto, utiliza-se o circuito [10], como mostrado na Figura 18.

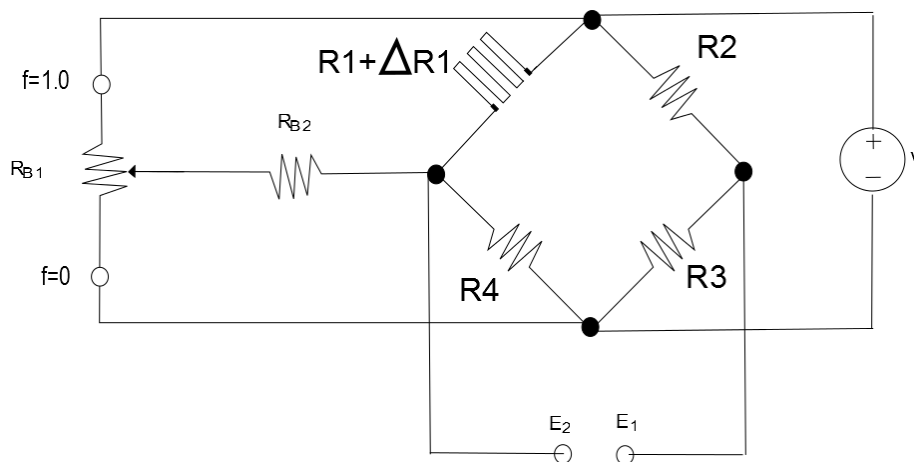


Figura 18 Balanço de ponte de Wheatstone

O potenciômetro R_{B1} tem que ser 10 vezes maior que a resistência $R1 + \Delta R1$, e a resistência R_{B2} tem que ser 10 vezes maior que R_{B1} . Tais condições garantem que R_{B2} não altere o divisor de tensão. O coeficiente de porcentagem do potenciômetro, f , determina a máxima corrente que será adicionada ou retirada da ponte de Wheatstone. Com $f > 0,5$, uma pequena variação de corrente é injetada no circuito pelo nó E_2 , e se $f < 0,5$ a corrente

elétrica será extraída da resistência fazendo com que a tensão no nó E_2 seja menor.

2.2.4 Amplificação

Para fazer a amplificação do sinal da ponte de Wheatstone, usa-se um amplificador de tipo instrumentação, o qual é feito de três amplificadores operacionais (AO) e sete resistências dispostas, como ilustrado na Figura 19. O amplificador pode ser visto como uma união de um amplificador isolador, ao lado esquerdo da linha vermelha, e um amplificador diferencial básico com ganho unitário, ao lado direito da linha. Este amplificador fornece as características de resistência na entrada muito elevadas e um ganho ajustável. Além disso, possui um *offset* que varia a tensão na saída (variando a resistência r), objetivando encontrar uma nova referência para a saída do amplificador [10].

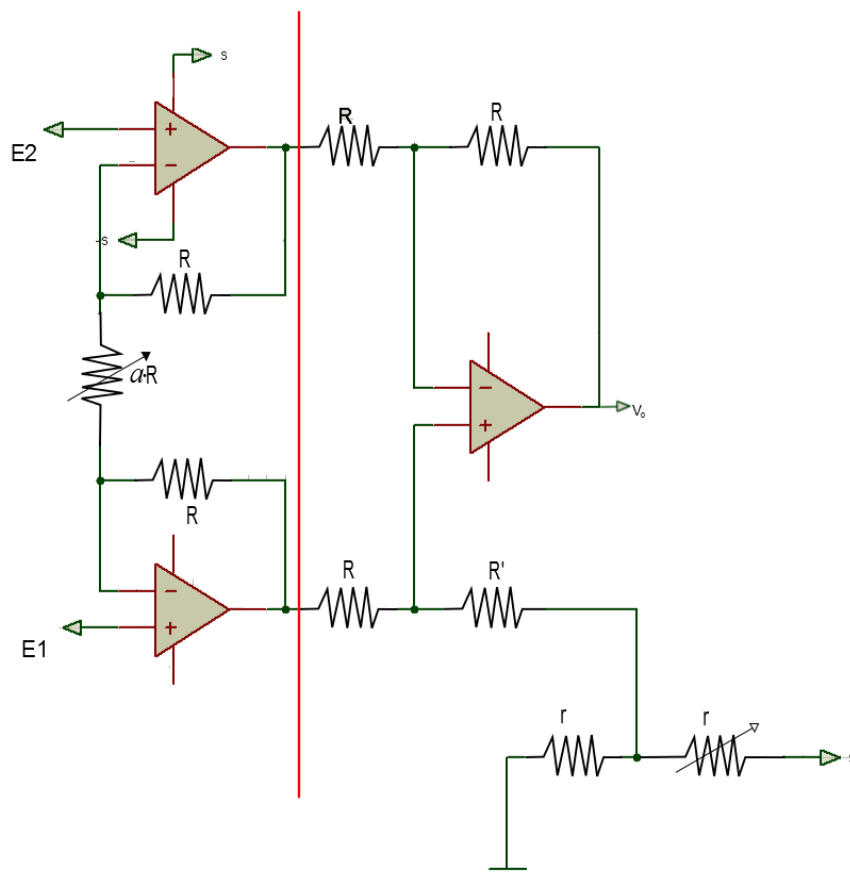


Figura 19. Amplificador instrumental

A variação do ganho é regida por a

$$\frac{V_o}{E_1 - E_2} = 1 + \frac{2}{a}, \quad (60)$$

onde

$$a = \frac{a_R}{R}. \quad (61)$$

e a_R é uma resistência variável.

2.2.5 Filtragem

Para a filtragem do sinal, foi utilizado um filtro Butterworth de segunda ordem, pois apresenta uma resposta com pouca perturbação na banda passante, ou seja, não modifica o sinal e nem induz ondulação na banda rejeitada.

A Figura 20 mostra o filtro Butterworth citado acima, com frequência de corte em 20 Hz. O filtro ativo foi calculado com a aplicação **FILTERLAB** na versão 2.0.0.0 da Microchip, um *software* especializado para projetos de filtros ativos. O *software* oferece uma grande facilidade na criação de filtros, uma vez que apenas as especificações do filtro são requeridas. Tais especificação estão detalhadas na Tabela 2.

Tabela 2 Especificações do filtro Butterworth.

Especificações.	
Tipo de filtro (<i>Approximation</i>).	Butterworth.
Seletividade (<i>Selectivity</i>).	Passa-baixa. (Low-pass).
Ganho (<i>Overall Filter Gain</i>).	1 (V/V).
Ordem do filtro (<i>Filter Order</i>).	2.
Atenuação da frequência de corte. (<i>PassBand Attenuation</i>).	-3 dB.
Frequência de corte. (<i>Passband Frequency</i>).	20 Hz.

Topologia do circuito (<i>Topology</i>).	<i>Sallen-Key</i> .
--	---------------------

Com isso, os resultados obtidos para o filtro foram $R_{11}=8,87K\Omega$ e $R_{22}=15K\Omega$, e 2 capacitores, o primeiro $C_{11}=1\mu F$ e o segundo $C_{12}=0,47\mu F$.

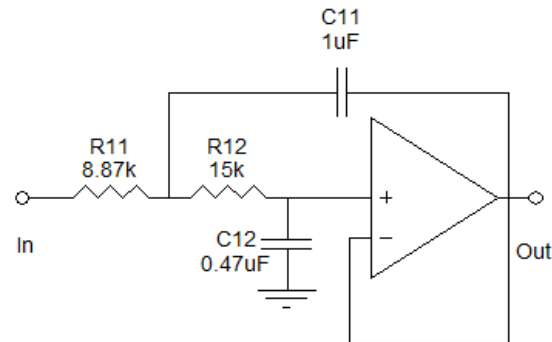


Figura 20 Filtro Butterworth de 2º ordem

Segundo [11], um circuito passa-baixa de segunda ordem de topologia Sallen-Key Butterworth tem como constantes $Q = 0.707$, que é dado por

$$Q = 0,5 \cdot \sqrt{\frac{C_{11}}{C_{12}}} \quad (62)$$

Substituindo pelos dados obtidos com o *software*, tem-se que

$$Q = 0,5 \cdot \sqrt{\frac{1}{0.47}} = 0,729 \quad (63)$$

Assim, a frequência de corte foi encontrada como sendo

$$f_p = \frac{1}{2 \cdot \pi \cdot \sqrt{R_1 \cdot R_2 \cdot C_{12} \cdot C_{11}}} = 20,126Hz \quad (64)$$

Na Figura 21 mostra-se a simulação da resposta em frequência do filtro ativo, onde é possível verificar a atenuação na frequência de corte em -3 dB e a atenuação a 60 Hz (frequência da rede elétrica) em -20dB.

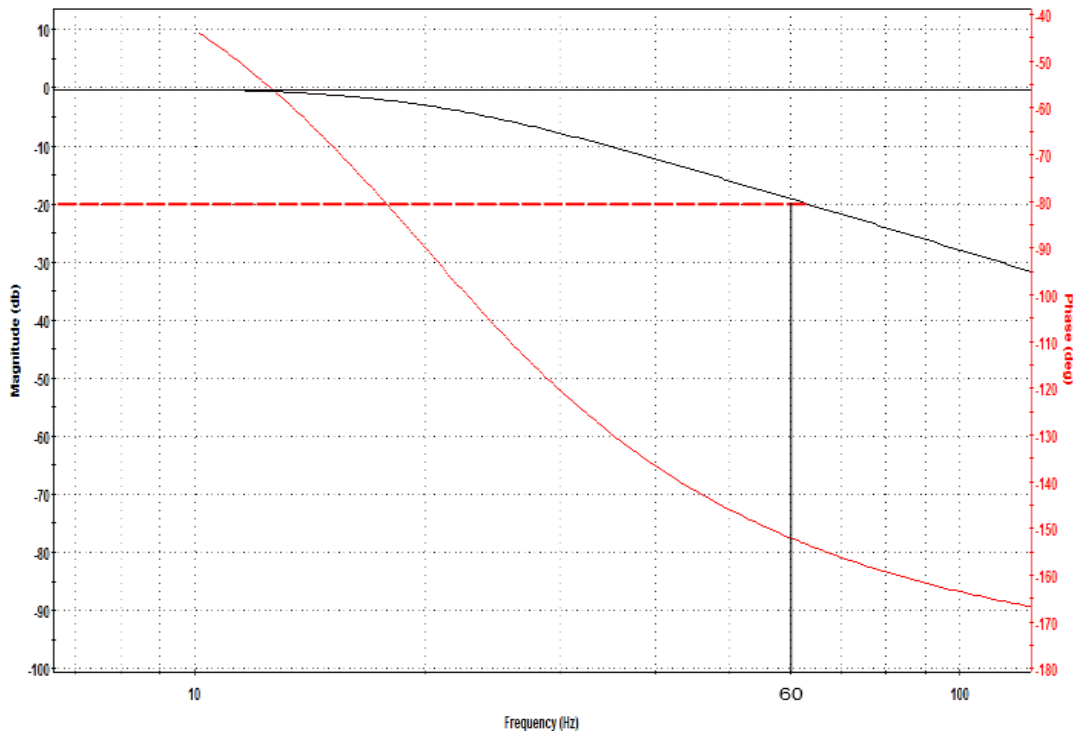


Figura 21 Resposta em Frequência do filtro adotado.

2.3 Redes neurais.

As redes neurais (ou *Neural Networks*) estão baseadas nas redes neurais biológicas que estão no cérebro humano. São constituídas por elementos que se comportam de maneira similar a um neurônio biológico nas suas funções mais comuns, apresentando características similares, como a aprendizagem por experiência, generalização de exemplos antigos com novos, e abstração de propriedades principais [12].

Os neurônios são unidades de processamento que recebem e combinam sinais de um neurônio para outro. Se esta combinação de sinais na entrada da saída é suficientemente forte, esta ativará a sua respectiva saída. Ele tem normalmente diferentes entradas, as quais são modificadas pela função de transferência, cujo valor é usado diretamente na saída do neurônio. Estas saídas podem estar conectadas a outros neurônios mediante conexões ponderadas, dependendo do seu estado de sinapses [13]. Na Figura 22 mostra-se o esquema de um neurônio artificial.

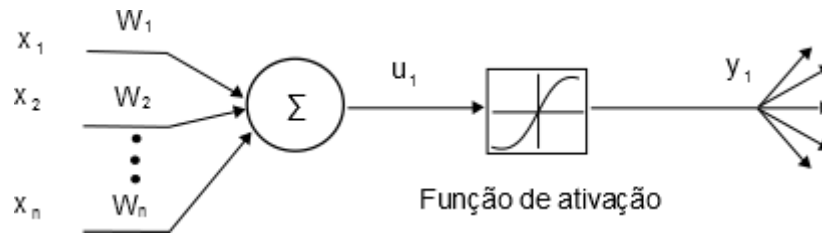


Figura 22 Diagrama de um neurônio artificial.

A saída de cada neurônio é dada por três funções, a primeira delas e chamada função de propagação ou excitação a qual é normalmente uma somatória das entradas multiplicada pelo seu valor de interconexão, o segundo é chamada como função de ativação, esta faz uma modificação da função de propagação e a sua existência é decidida pelo critério do arquiteto da rede, a terceira função é a função de transferência esta função e dada segundo a interpretação desejada na saída do sistema. As funções mas utilizadas são a função sigmoide que tem um intervalo de 0-1 e a função tangente hiperbólica que da valores entre -1 e 1.

Multilayer perceptron

Esta arquitetura chamada mapeamento de padrões (*pattern-mapping*) tem a qualidade de aprender a classificar padrões com a ajuda de um algoritmo de aprendizagem supervisionado. O modelo geralmente tem como entradas valores previamente normalizados, e as categorias de saída, as categorias de classificação, normalmente são feitas com vetores binários; estas, ao terem várias camadas, permitem que sejam resolvidos problemas linearmente não separáveis, permitindo uma grande variedade de situações onde pode ser aplicado.

2.4 Controle baseado em admitância

O controle baseado em admitância utiliza uma força medida para gerar uma resposta de controle proporcional a ela, propiciando assim uma resposta suave, associada a um conforto no movimento se bem calibrada. Ou seja, esse controle gera uma resposta que proporciona uma sensação de naturalidade do movimento. O objetivo do controlador baseado em admitancia não é a regulação de uma força em cambio se quer fazer uma relação entre uma força, no caso a mesurada, e uma velocidade de deslocamento adequada para o usuário.

O modelo pode ser definido como uma função de transferência $G(s)$ que tem como entrada as forças e os torques do usuário $F(s)$, e como saída a velocidade da plataforma omnidirecional $V(s)$, i.e.

$$G(s) = \frac{V(s)}{F(s)} = \frac{1}{M \cdot s + B} \quad (65)$$

onde M e B são os parâmetros de massa e de amortecimento adotados os quais variam de acordo com a necessidade. A resposta da plataforma omnidirecional é obtida através da velocidade de saída da equação (65). Deriva-se a velocidade para obter a aceleração comandada, a qual será utilizada na resolução do modelo Dinâmico em tempo real, e dizer a força entra no modelo de admitância, sai deste a velocidade calculada e deriva-se para obter a aceleração correspondente para resolver o modelo dinâmico e, assim, obter o torque necessário nas rodas (ver Figura 23).

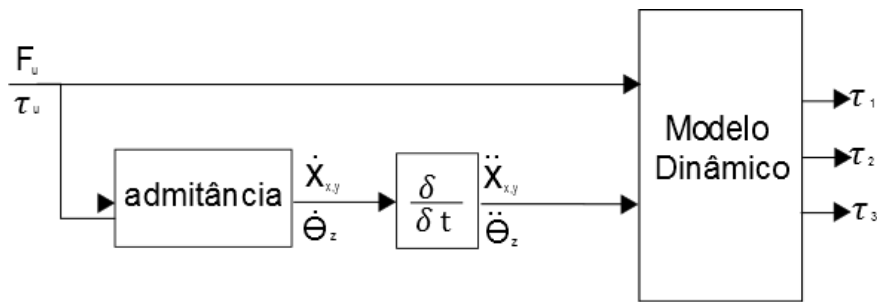


Figura 23 Diagrama de controle baseado em admitância.

Como a plataforma omnidirecional possui 3 graus de liberdade, deslocamento em x e y e uma rotação no eixo z , o modelo massa-amortecedor implementado na plataforma resulta na equação vetorial:

$$\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\omega}_z \end{bmatrix} + \begin{bmatrix} B_x & 0 & 0 \\ 0 & B_y & 0 \\ 0 & 0 & B_z \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ \tau_z \end{bmatrix}, \quad (66)$$

onde M é a massa, I_z é o momento de inércia da plataforma no eixo z . e B_x, B_y, B_z são cada um dos valores de constantes de amortecimento desejados em cada um dos eixos. Essa modelagem é feita para gerar uma resposta

diferente em cada um dos eixos. Com diferentes valores de B_x, B_y, B_z é possível minimizar o impacto das forças indesejadas na trajetória final, já que este trabalha como um filtro de alta frequência. O sistema descrito na Eq. (66) está representado esquematicamente na Figura 24.

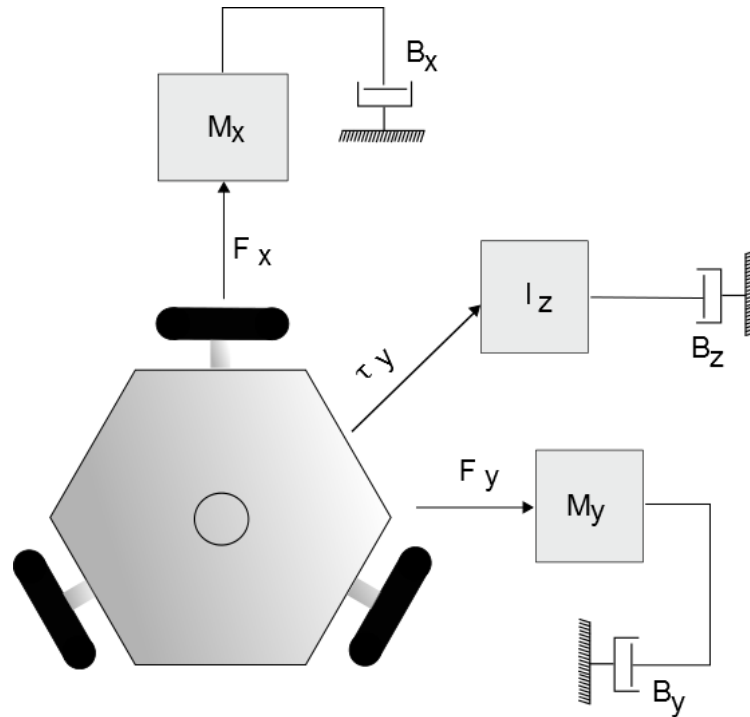


Figura 24 Modelo massa-amortecedor utilizado.

Analisando a resposta do sistema acima submetido a uma entrada em degrau, no domínio do tempo tem-se

$$v(t) = \frac{F}{B} \left(1 - e^{-\frac{t}{\tau}} \right), \quad (67)$$

onde $\tau = \frac{M}{B}$.

2.5 Controle PID

O controlador PID é um dos mais usados na indústria [14], devido à sua facilidade de utilização e calibragem, e sua versatilidade em diferentes sistemas. Ele é composto por 3 tipos de controle: Proporcional (P), Integral (I) e Derivativo (D).

2.5.1 Controle Proporcional (P).

O sinal de saída $u(t)$ neste controlador é calculado utilizando o erro $e(t)$ medido junto com o ganho proporcional K_p . Este erro é a diferença entre a medida da variável de controle e seu valor desejada. Matematicamente, este controlador pode ser escrito como

$$u(t) = K_p \cdot e(t), \quad (68)$$

Ele tem a vantagem de diminuir o tempo de resposta, mas na presença de perturbações não conseguirá atingir erro nulo no regime permanente. Além disso, para valores altos de K_p , pode-se tornar instável.

2.5.2 Controle Integral (I).

A resposta deste depende da acumulação do erro na entrada, fazendo que com o passar do tempo o controlador consiga chegar a um erro nulo em regime permanente. O controlador integral pode ser escrito como

$$u(t) = K_i \cdot \int_0^t e(t) \cdot dt, \quad (69)$$

2.5.3 Controle Derivativo (D).

Este controlador é um diferenciador do sinal de entrada, capaz de gerar uma resposta rápida para um erro grande de velocidade. À medida que o erro diminui, ele reduz a saída do controlador, e assim evita que se ultrapasse o valor desejado. Ele pode ser expresso por

$$u(t) = K_d \cdot \frac{de(t)}{dt}, \quad (70)$$

Este termo tem a desvantagem de funcionar como um filtro passa-alta, susceptível a ruídos de alta frequência, gerando distúrbios no processo de controle.

No controlador PID, são utilizados os três controladores apresentados acima, obtendo a lei de controle combinada

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) \cdot dt + K_d \cdot \frac{de(t)}{dt}, \quad (71)$$

2.5.4 Método do lugar das raízes

O método do lugar das raízes é uma representação gráfica dos pólos e zeros de um sistema de malha fechada, quando se varia um parâmetro do sistema. Este é um método rápido e eficaz de análise e projeto de estabilidade e resposta transiente [15]. A vantagem do método implica na facilidade de fornecer soluções para sistemas de ordem superior.

Define-se

$$G(s) = \frac{\omega_n^2}{s^2 + 2 \cdot \xi \omega_n s + \omega_n^2}, \quad (72)$$

a função de transferência de um sistema de malha fechada de segunda ordem, como mostrado na Figura 25. Nesta equação, o fator de amortecimento relativo (ξ) [14] é dado por:

$$\xi = \frac{\ln(Mp)}{\sqrt{\pi^2 + \ln(Mp)^2}}, \quad (73)$$

onde o valor Mp é o fator máximo de *overshoot* desejado no sistema de controle, com frequência natural sem amortecimento dada por:

$$\omega_n = \frac{4}{\xi \cdot T_s}, \quad (74)$$

onde T_s é o tempo de estabilização. Os pólos relativos à Eq. (88) são dados por:

$$Pd = -\xi \cdot \omega_n \pm j \cdot \omega_n \cdot \sqrt{1 - \xi^2}, \quad (75)$$

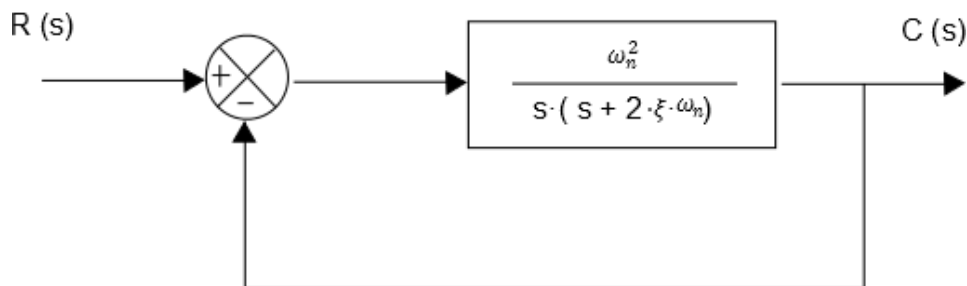


Figura 25 Sistema de segunda ordem

Pode-se fazer o projeto do sistema de controle através do método do lugar das raízes, tendo como entrada um sinal em degrau (C). Satisfazendo a condição de ângulo dada por:

$$\angle G(s) \cdot c(s) = \pm 180^\circ \cdot (2 \cdot k + 1) \quad (76)$$

e com a condição de magnitude dada por

$$|G(s)H(s)| = 1, \quad (77)$$

podem-se achar os valores do compensador desejado.

2.5.5. Implementação no microcontrolador PIC

Um controlador digital PID, pode ser expresso por

$$\frac{U(z)}{E(z)} = K_p \left[a + \frac{b}{1-z^{-1}} + c(1-z^{-1}) \right], \quad (78)$$

onde as respectivas constantes podem ser obtidas a partir dos ganhos PID usando

$$a = K_p, \quad b = \frac{K_p T}{T_i}, \quad c = \frac{K_p T_d}{T}$$

Para a implementação do controle no microcontrolador *PIC*, escolhe-se um ponto como referência, denominado *SetPoint*, mede-se o sinal a ser controlado, e calcula-se o erro entre ambos. Em seguida, calculam-se os valores dos termos do controlador PID para gerar a saída para o atuador. A Figura 26 mostra os detalhes da implementação do PID no PIC.

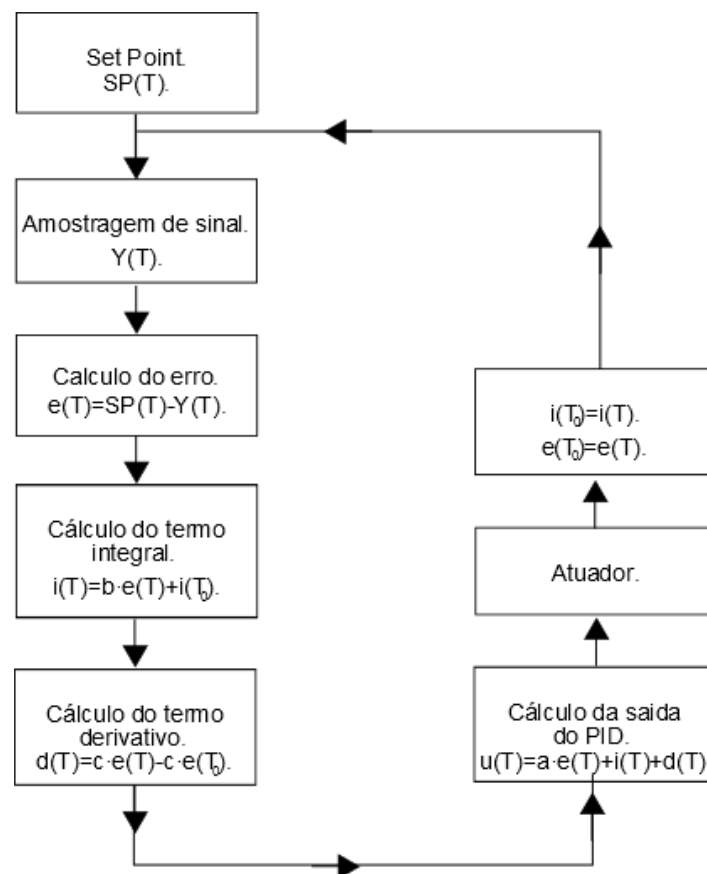


Figura 26 Diagrama de implementação de PID no PIC

A partir da fundamentação teórica abordada neste capítulo, a construção do protótipo da bengala robótica omnidirecional pode ser inicializada. O próximo capítulo descreve os procedimentos adotados para a confecção da bengala.

3. Projeto, simulação e controle do protótipo experimental

Neste capítulo será descrito o projeto da bengala robótica desenvolvida nessa dissertação, que consta de 3 partes. Na primeira, a parte mecânica onde serão tratadas as questões relacionadas ao projeto, construção, localização dos *strain gages*, assim como as considerações tomadas com base nas medições do compor humano. Além disso, será abordada a simulação do modelo cinemático da plataforma omnidirecional, e também seu modelo dinâmico.

A segunda parte deste capítulo abordará a eletrônica do sistema, detalhando as partes do circuito desenvolvido. Serão brevemente apresentados os diagramas dos planos elétricos para a construção das placas, assim como uma explicação do funcionamento de cada grupo de componentes.

Na terceira parte será abordado o controle utilizado na bengala, incluindo o controle baseado em admitância, o controle de baixo nível (PID), e o uso de redes neurais, onde será descrito todo o processo para aquisição de dados, pré-processamento e treinamento da rede.

3.1 Projeto mecânico

O projeto mecânico do sistema foi baseado em uma bengala de 4 pontas padrão, que foi comprada em uma loja especializada em implementos de ortopedia, feita de tubo de alumínio com diâmetro 20mm e espessura de parede 1mm. As 4 pontas da bengala foram cortadas, como mostrado na Figura 27, deixando assim uma pequena superfície para parafusar a plataforma de alumínio desenvolvida, com o objetivo de conter com mais espaço físico para fazer um melhor suporte.

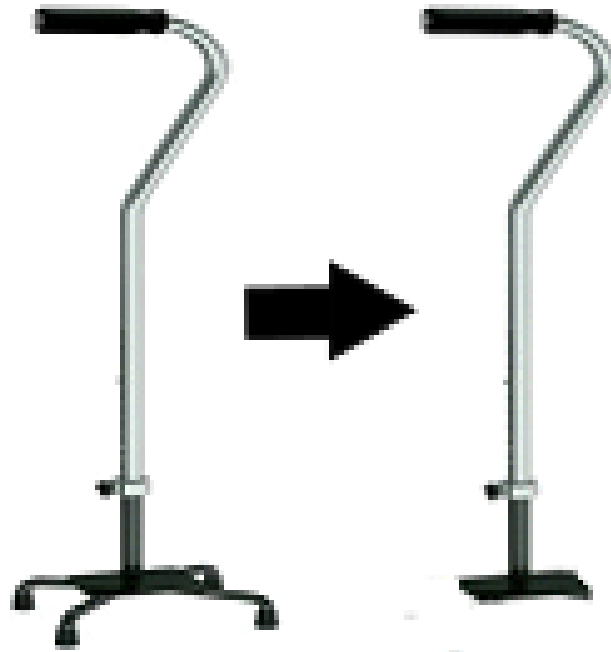


Figura 27 Bengala de 4 pontas padrão, e versão cortada.

A plataforma é composta por três chapas de alumínio 5052F com forma hexagonal regular com 16 cm de lado e espessura de 5 mm. A forma hexagonal foi escolhida pela facilidade na localização e orientação dos motores em uma configuração de três rodas omnidirecionais. Além disso, a forma hexagonal (ao invés de circular) da plataforma ajuda, já que dificuldade ao usuário em bater-se e machucar-se com a mesma, oferecendo um pouco de espaço para que o seu pé não interfira com a plataforma. As dimensões do usuário padrão foram estimadas a partir de dados médios descritos em [16]. A distância média que pode ser atingida na caminhada pela ponta dos pés de uma pessoa é de aproximadamente 44 cm [17] desde o plano coronal Figura 28 em direção ventral/anterior, como pode ser visto na Figura 29. O comprimento médio do braço da pessoa, desde o mesmo plano até o punho, é de cerca de 88 cm em média, como detalha a Figura 29, deixando um comprimento livre de $88 - 44 = 44$ cm. A base foi feita de alumínio ASTM 5052F, uma vez que apresenta um baixo custo, é relativamente fácil de usinar, possui boa resistência à corrosão, e tem rigidez suficiente para a tarefa desejada.

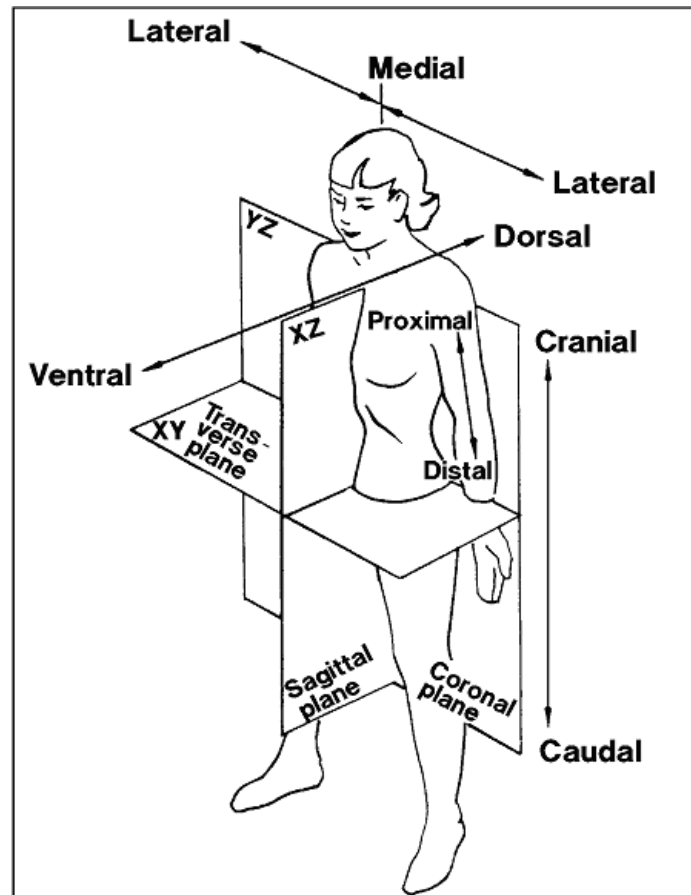


Figura 28 Planos do corpo

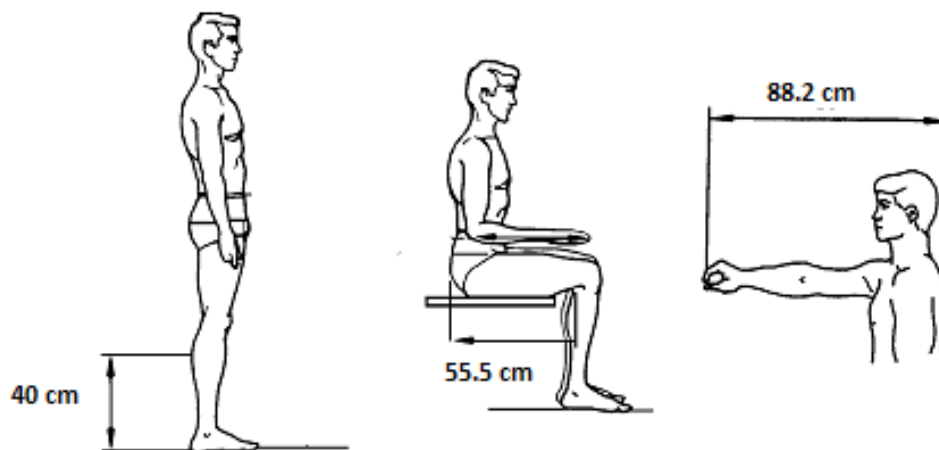


Figura 29 Medições antropométricas médias [21].

A Tabela 3 apresenta as características físicas e mecânicas da chapa de alumínio utilizada na construção da plataforma omnidirecional, o qual mostra um baixo peso, baixa dureza para uma fácil usinagem e limite de escoamento ótimo para seu uso no projeto.

Tabela 3 Propriedades físicas e mecânicas do alumínio ASTM 5052F

Propriedades físicas	Propriedades Mecânicas
<ul style="list-style-type: none"> • Peso específico = 2,68 (g/cm³). • Temperatura de Fusão = 595-650 °C. • Condutibilidade Térmica (IACS)% = 34. 	<ul style="list-style-type: none"> • Limite de escoamento = 65MPa. • Limite de resistência = 170MPa.

A plataforma móvel é formada por três níveis, separados pelas placas de alumínio: no primeiro nível está a parte de potência mecânica, que é constituída por três motores de 20W (Matrix Pro Lathe 55T) com a sua respectiva caixa de redução (P60 gearbox), tendo 120 graus de separação entre eles. Cada um dos motores possui uma roda omnidirecional dupla de 100mm de diâmetro, espessura de 16 mm e 290 gramas de peso, feita de alumínio e borracha. Escolheram-se rodas omnidirecionais pela necessidade de deslocamento rápido em qualquer direção, e sua quantidade reduzida a 3 (ao invés de 4 rodas, como em muitos sistemas omnidirecionais) para diminuir o número de variáveis a serem controladas e o consumo de energia, além de diminuir o custo computacional.

No segundo nível da plataforma está a parte de potência elétrica e a alimentação, composta por dois controladores de velocidade *Sabertooth 2x60* e duas baterias de lítio tipo 5S com capacidade 4000mAh. No terceiro nível estão os demais circuitos eletrônicos e os sensores (ERE ou *strain gages*), incluindo o sistema de amplificação e filtragem dos sinais e o microcontrolador PIC para o processo de controle.

A placa de alumínio do primeiro nível (Figura 30) é responsável por fixar as caixas redutoras, e possui furos para ser conectada aos demais níveis da plataforma. A placa de alumínio do nível 2 é a mesma usada no nível 1, pois esta também é necessária para prender os motoredutores. Este nível serve de base para fixar os circuitos de potência, assim como os 2 controladores de velocidade Sabertooth e as baterias, sempre com o objetivo de conservar a simetria no arranjo e tentar manter o centro de massa do sistema no eixo da bengala. A placa de alumínio para o nível 3, onde fica a parte eletrônica do sistema assim como os sensores, as pontes de Wheatstone, o sistema de amplificação, e a parte de filtragem e processamento do controle, pode ser observada na Figura 31.

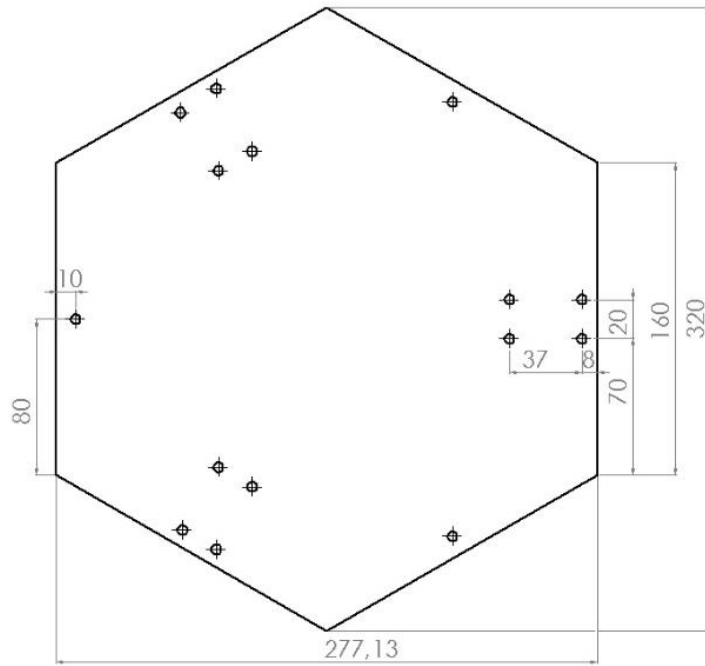


Figura 30 Desenho da base de alumínio para os níveis 1 e 2.

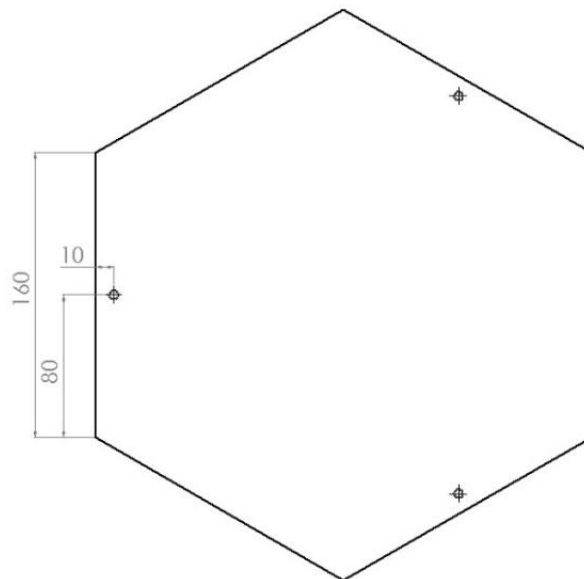


Figura 31 Desenho da base de alumínio para o nível 3.

Os 3 níveis estão unidos por três barras roscadas de 5mm de diâmetro, de passo fino e com um comprimento de 150mm, fixando cada um dos níveis com seis porcas e seis arruelas (2 porcas e 2 arruelas por barra), garantindo assim uma boa rigidez estrutural.

O conjunto de sensores é formado por 12 ERE (*strain gages*) da empresa *Excel Sensor*, dispostos em grupos de quatro rosetas com três ERE cada, separadas de 90 graus entre si e coladas o mais próximo possível da superfície do terceiro nível da plataforma. Com essa localização, como mostra a Figura 32, pode-se medir com melhor precisão as forças e torques que o usuário irá exercer na bengala.

Adicionalmente a esse conjunto, há três sensores de efeito hall ACS712ELCTR-20A-T da Allegro Microsystems, usados para medir as correntes elétricas enviadas aos motores, utilizadas para estimar os torques efetivamente aplicados às rodas.

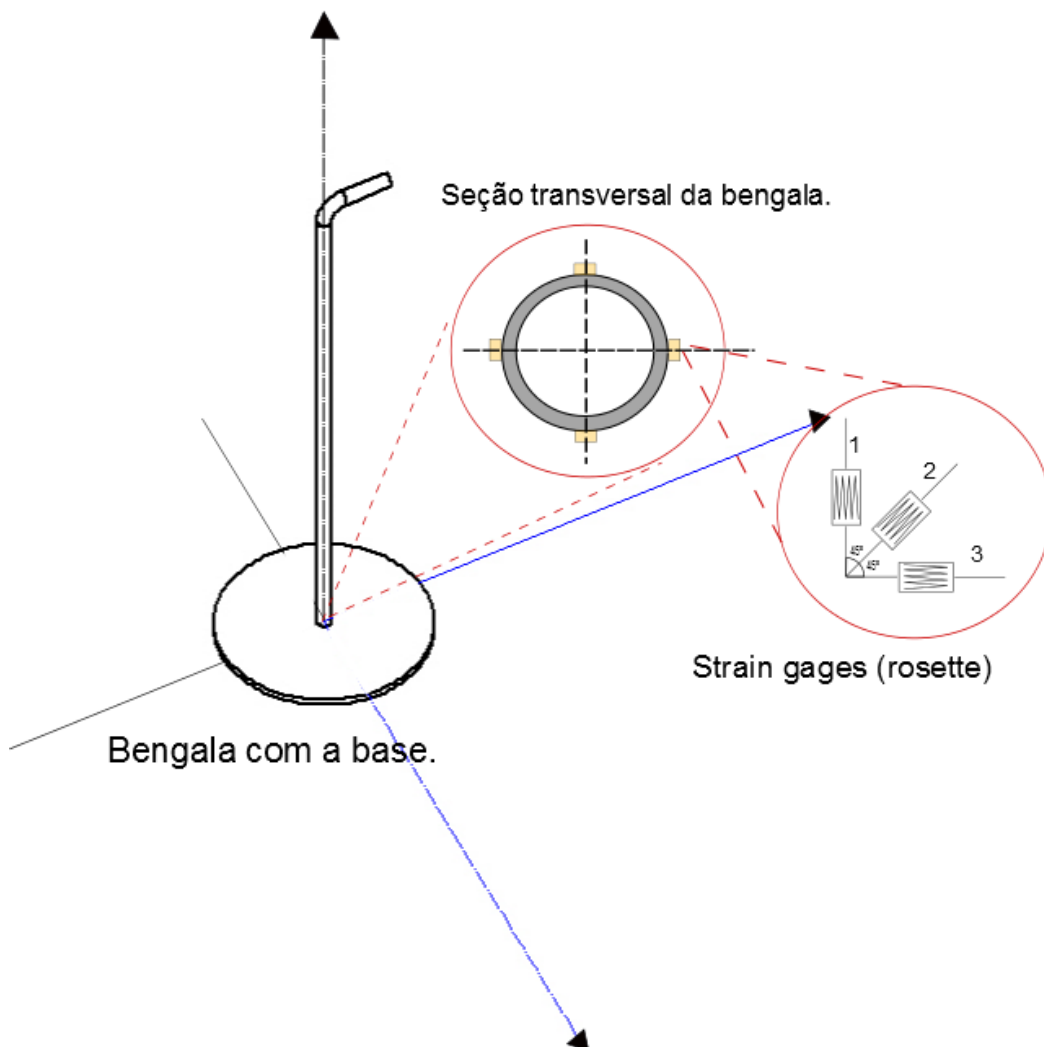


Figura 32 Disposição dos strain gages na base da bengala.

A Figura 33 mostra a disposição da haste e das 3 bases de alumínio da bengala robótica, incluindo rodas e motoredutores, desenhadas no *software* de CAD 3D *Solidworks*. A Figura 34 mostra o protótipo omnidirecional completo, constituído das partes mecânica e eletrônica.

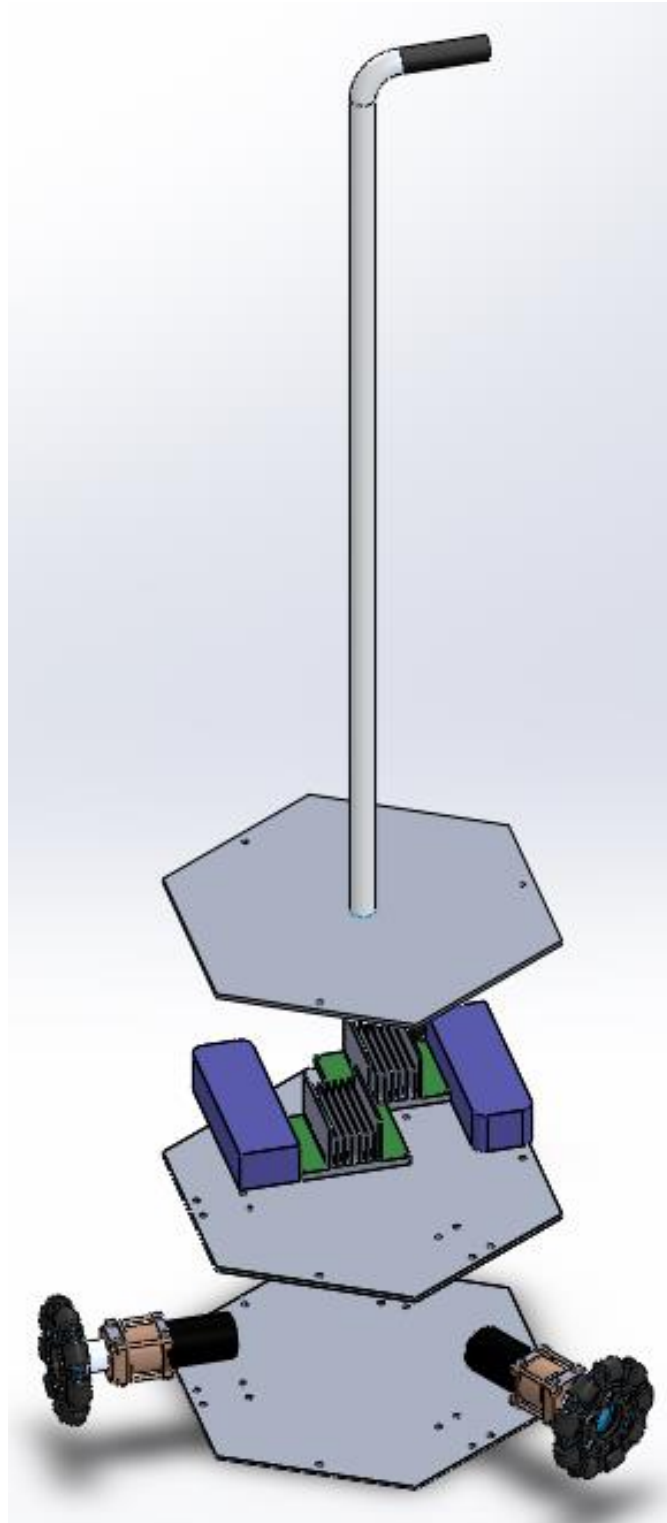


Figura 33 Modelagem da plataforma omnidirecional.



Figura 34 Protótipo completo da bengala robótica.

3.2 Simulação do modelo cinemático

O modelo cinemático apresentado no item 2.1.1 foi implementado em *Simulink*, ferramenta do *software Matlab*. Nesta ferramenta, o bloco de nome *MATLAB Function* (Figura 35) permite ao usuário gerar um script (Apêndice B.1). Neste projeto, o nome criado para o bloco foi **FunModeloCinematicoDir**.

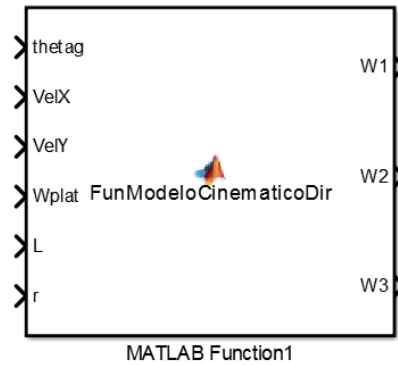


Figura 35 Função do modelo cinemático no Simulink.

Este bloco tem como entradas as dimensões da plataforma omnidirecional: distância do centro de massa da plataforma até o centro de massa das rodas (L) e o raio das rodas (r); velocidades desejadas em x ($VelX$) e em Y ($VelY$), e velocidade angular do eixo z ($Wplat$); e o ângulo de rotação no eixo z ($thetag$). As únicas saídas do modelo são as velocidades angulares das 3 rodas ($W1$, $W2$ e $W3$).

Tabela 4 Dados do modelo cinemático.

Entrada	Valores
$VelX$	$2 \cdot \sin(t)$ (em m/s)
$VelY$	1 m/s
$Wplat$	0 rad/s
t	10s
L	170mm
r	50mm

A Figura 36 apresenta o resultado de uma simulação (a trajetória percorrida pela plataforma omnidirecional) usando o modelo cinemático, contido no arquivo **RunCinematic.m**, com os parâmetros de entrada da Tabela 4, nesta pode-se enxergar a linha de cor vermelho a qual é a trajetória percorrida pela plataforma desde as coordenadas (0,0) até a coordenada (4,10), e a plataforma é representada pelo desenho de 3 “L” juntas no médio. Os gráficos da Figura 37 mostram as respectivas velocidades lineares e angulares das rodas ($W1$, $W2$ e $W3$), da trajetória mostrada na Figura 36.

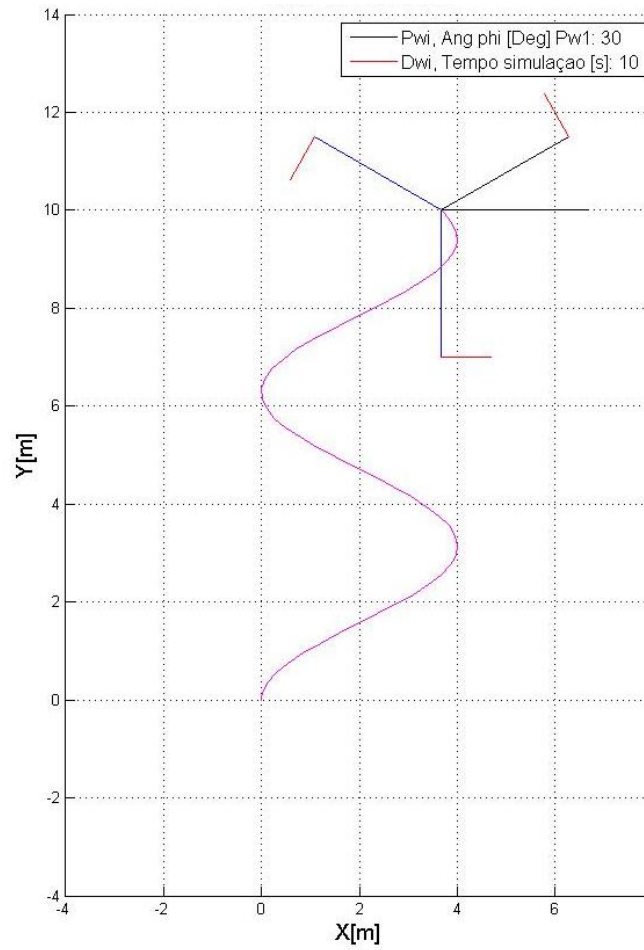


Figura 36 Simulação do modelo cinemático da bengala robótica para os valores da Tabela 3.

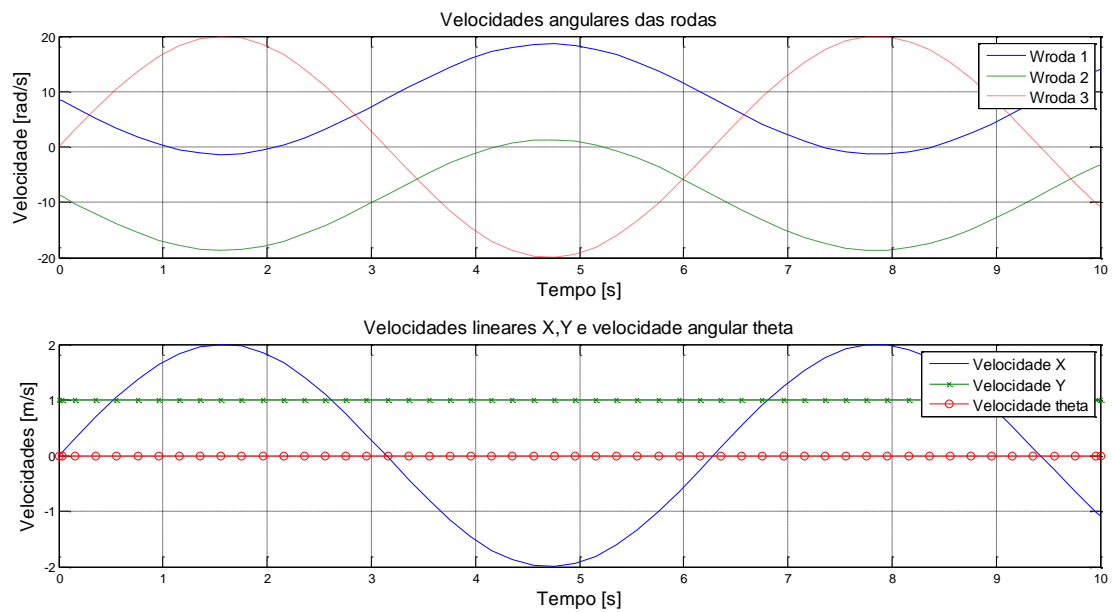


Figura 37 Entradas e saídas do modelo cinemático da bengala.

3.3 Simulação do modelo dinâmico

O modelo dinâmico apresentado no item 2.1.2 também foi implementado em *Simulink / Matlab*. Nesta ferramenta, o bloco de nome *MATLAB Function* (Figura 38) permite ao usuário gerar um script (Apêndice B.2). Para este projeto, o nome criado para o bloco foi **FunModelodinamicoDir**.

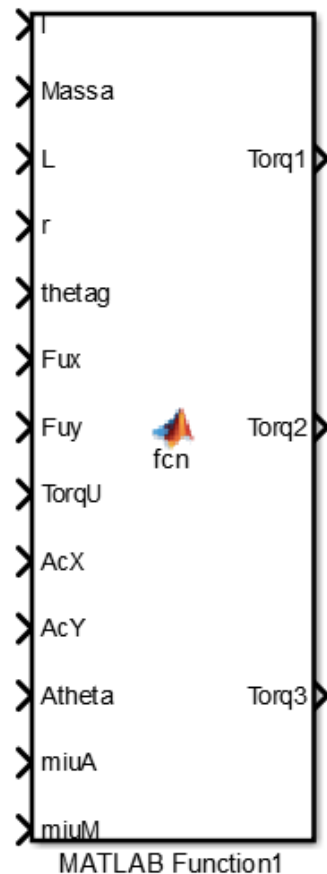


Figura 38 Função do modelo dinâmico no Simulink

Este bloco tem como entradas as propriedades físicas inerciais no eixo vertical (I) a qual foi obtida utilizando com *SolidWorks* e a massa total da bengala ($Massa$); as dimensões da plataforma omnidirecional: a distância do centro de massa do sistema até o centro de massa das rodas (L), o raio das rodas (r), e o ângulo de rotação da bengala no eixo z ($thetag$); as forças do usuário: força no eixo X (Fux), força em Y (Fuy) e o torque no eixo Z ($TorqueU$); as acelerações desejadas: aceleração linear no eixo X (AcX), no eixo Y (AcY), e aceleração angular no eixo z ($Atheta$); e os coeficientes de atrito na direção axial da roda omnidirecional ($miuA$, associada às perdas no rolamento dos

roletes da roda, que em um caso ideal seriam nulos) e na direção do movimento da roda (miuM, associada à tração do sistema, lembrando que estas rodas omnidirecionais são em teoria capazes de promover tração apenas na direção tangencial). As únicas saídas do modelo são os torques a serem aplicados em cada uma das rodas (Torq1, Torq2 e Torq3).

A Figura 39 apresenta o perfil de forças aplicadas pelo usuário nas direções do plano horizontal (X e Y), e as velocidades resultantes obtidas das simulações dinâmicas incluindo o controle baseado em admitância. Na Figura 40 são apresentados os torques calculados pelas leis de controle para os 3 motores, assim como as duas acelerações lineares e angular. A Figura 41 apresenta as velocidades angulares das rodas como resultado da simulação do modelo cinemático, assumindo que não há deslizamento.

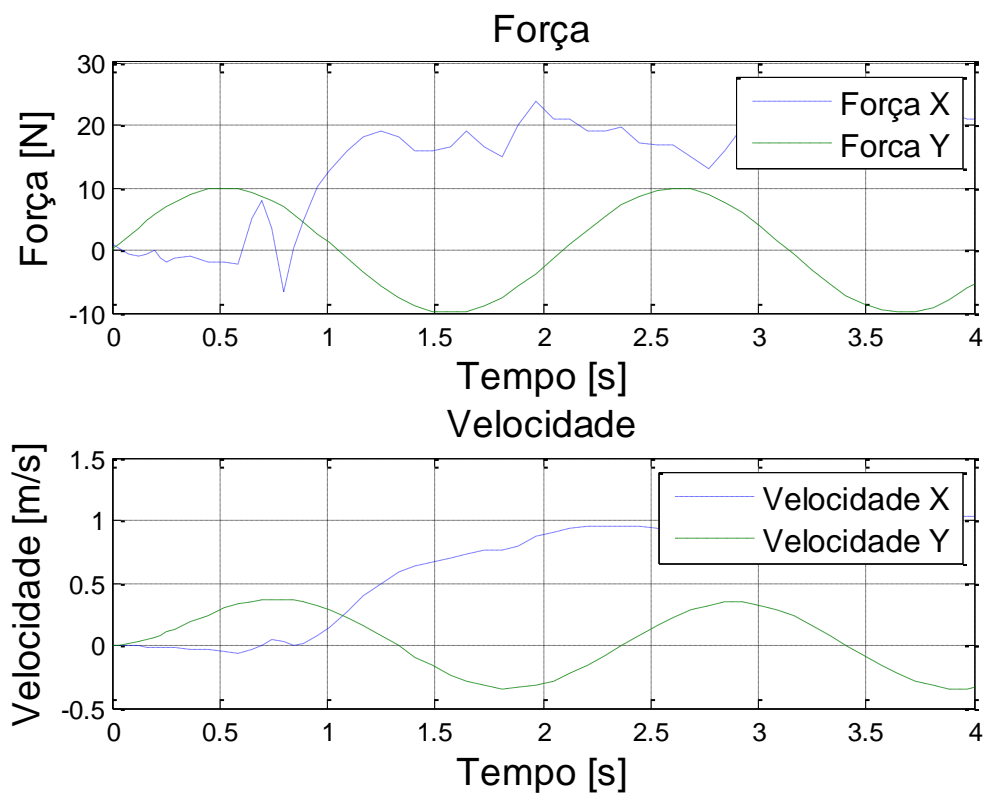


Figura 39 Entradas do modelo dinâmico da bengala

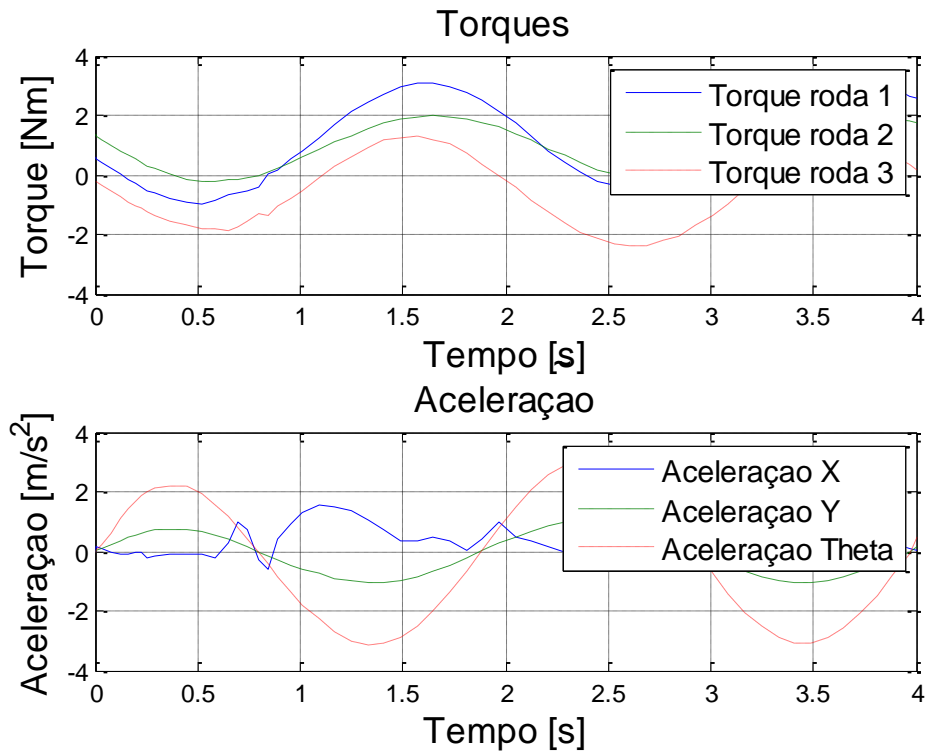


Figura 40 Saídas do modelo dinâmico da bengala

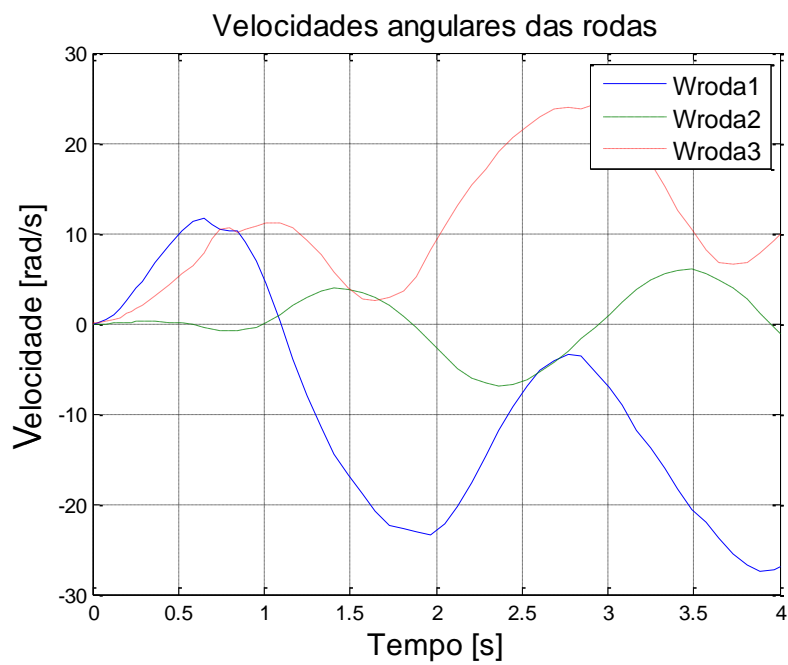


Figura 41 Velocidades angulares das rodas.

3.4 Controle

O controle da bengala robótica consta de 3 unidades básicas: a primeira é um controle baseado em admitância, para permitir o controle de força pelo usuário; a segunda unidade é um controle de alto nível ou supervisor, para lidar com situações de queda iminente; e a terceira é um controle *PID*, que é o controle de baixo nível para movimentação das rodas. Estas 3 unidades têm como finalidade gerar um melhor conforto no manuseio da bengala, decidir entre um estado de caminhada ou queda iminente, e o controle dos motores, respectivamente. O conjunto de todas as etapas de controle é mostrado na Figura 42. As 3 unidades de controle são descritas nas seções seguintes.

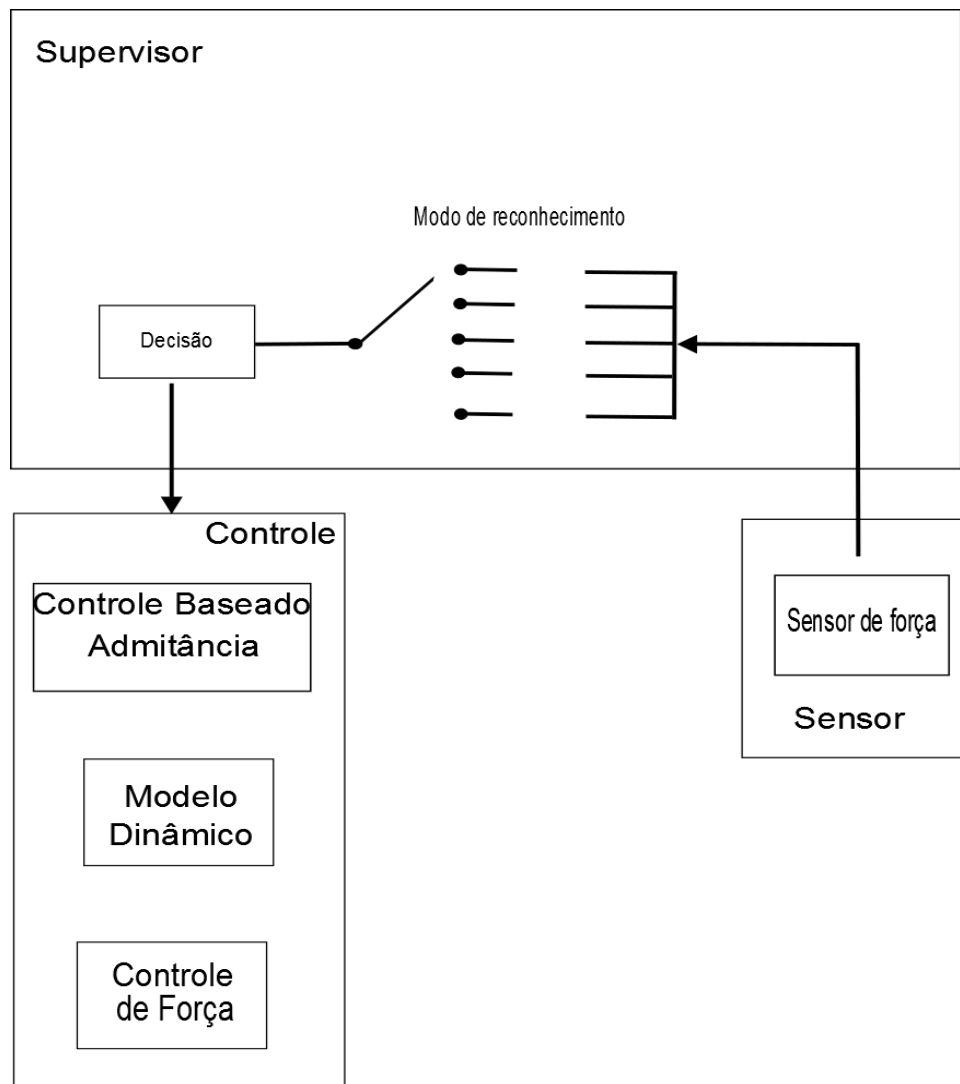


Figura 42 Esquema geral de controle

3.5 Controle de alto nível ou supervisor

O controle de alto nível, ou supervisor, é aquele que irá identificar a intenção do usuário, e a partir disso ajudará na decisão de controle da bengala, evitando possíveis condições indesejadas, tais como queda iminente. O controle baseado em ITD tem 3 modos de caminhada, os quais são mostrados na Tabela 5. Qualquer modo pode ser mudado para qualquer outro e em qualquer ordem, como apresentado no diagrama da Figura 43, dependendo apenas das decisões do controle de alto nível. Este controle de alto nível é comandado por uma rede neural *Multi Layer Perceptron* (MLP), que foi modelada no ambiente *Matlab*.

Tabela 5 Intenções do usuário

Modos de caminhada.		
Modo	Descrição.	
I	Parada	
II	IIa	Direto frente
	IIb	Linha reta em qualquer direção.
III	IIIa	Direita
	IIIb	Esquerda

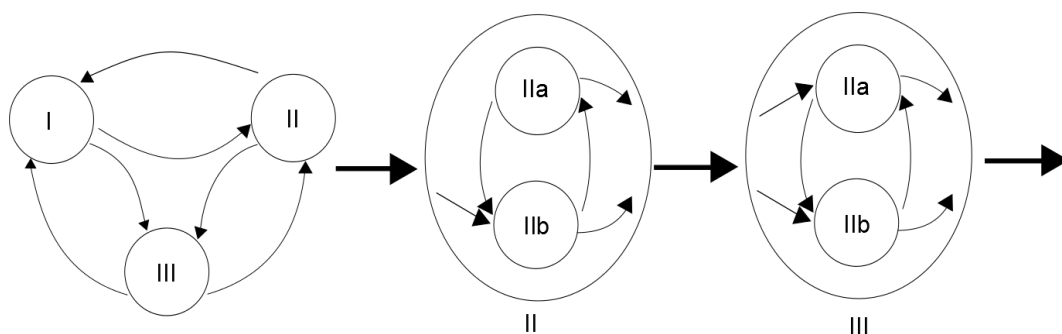


Figura 43 Intenções do usuário

A aquisição de dados dos *strain-gages* foi feita em formato de 8 bits (0 - 255), com uma frequência de 20Hz. Foram feitos 10 testes de queda e 10 testes

de caminhada para cada uma das 4 direções (frente, trás, direita, esquerda), sendo um total de 80 testes. Estes dados são descritos no Capítulo 4, na seção “base de dados”.

Para testes de queda, os dados relativos aos instantes posterior e anterior ao evento da queda foram eliminados, deixando só um intervalo que corresponde à janela limite para que o sistema consiga identificar o movimento. Para facilitar o processamento pela rede, um *offset* foi ajustado aos dados subtraindo os valores medidos imediatamente antes da janela considerada. Desse modo, o evento de caminhada ou queda foi identificado a partir de sinais associados a forças e torques relativos.

Para encontrar o modelo mais adequado para treinar a rede neural, treinamentos divididos em duas fases foram feitos. Na primeira fase, tentou-se identificar padrões de queda e caminhada e, na segunda fase, pretendia-se encontrar padrões de direção. Tudo isto é feito na busca de uma arquitetura base das redes neurais a serem adotadas.

3.5.1 Primeira fase

Esta fase tem como objetivo identificar os padrões dos estados de caminhada e queda, e que tem como entradas os dados dos 10 sensores de deformação tendo este um rango de valores de 0 a 255, onde os valores acima de 128 representam uma situação de tensão no corpo da bengala e os valores embaixo deste número representam uma situação de compressão, igualmente está-se trabalhando com uma saída codificada binária de 1 bit, onde o número 1 está associado a queda, e 0 a uma caminhada normal.

- **Primeiro Modelo.**

Utilizou-se a diferença entre dois valores consecutivos da amostra de pontos, pois o objetivo era avaliar a diferença das leituras do sensor entre duas amostras consecutivas:

$$y(t) - y(t-1) \quad (79)$$

onde $y(t)$ é o valor do sensor no instante t , e $y(t-1)$ é o valor no instante de aquisição imediatamente anterior.

Foram analisadas as seguintes arquiteturas de redes neurais:

Tabela 6 Arquitetura do primeiro modelo

Camadas	1	2
Neuronios	10	-
	10	10
	20	-
	20	20

- **Segundo Modelo**

Para uma segunda proposta de modelo, foram utilizados os mesmos dados dos sensores utilizados no primeiro modelo, mas estes foram normalizados entre $[0,1]$.

- **Terceiro Modelo**

Neste modelo, a métrica foi trocada conforme Eq.(80):

$$\frac{y(t) + y(t-1)}{2} \quad (80)$$

Para este modelo foram analisadas as seguintes arquiteturas de redes neurais:

Tabela 7 Arquitetura do terceiro modelo

Camadas	1	2
Neurônios	10	-
	10	10
	20	-

3.5.2 Segunda fase

O objetivo dessa fase foi identificar os padrões de direção da força exercida na bengala, ou seja, deseja-se indicar quando a bengala foi submetida a uma força que indica deslocamento para frente, para trás, para direita ou para a esquerda. Nesse experimento, foram utilizadas as mesmas características do modelo 3 da fase 1. Além disso, fez-se uma codificação binária da saída em 4 dígitos, como apresentado na Tabela 8.

Tabela 8. Codificação dos movimentos

Codificação		
Classe	Movimento	Binária
1	Frente	[0 0 0 1]
2	Trás	[0 0 1 0]
3	Direita	[0 1 0 0]
4	Esquerda	[1 0 0 0]

Nesta fase, foram alterados os valores dos números de neurônios nas camadas escondidas e a quantidade de camadas, como descrito abaixo.

Tabela 9 Arquitetura do terceiro modelo

Camadas	1	2	3
Neurônios	10	-	-
	10	10	
	10	10	10
	11	-	-
	11	11	-
	11	11	11
	15	-	-

	15	15	-
	15	15	15
	20	-	-
	20	20	-
	20	20	20
	28	-	-
	28	28	-
	28	28	28

3.6 Controle de baixo nível

Para avaliar o controlador de baixo nível, responsável pelo controle dos motores das rodas da bengala robótica, uma aquisição de dados experimentais foi feita e aplicada ao módulo de identificação do *Matlab*. Nele escolheu-se uma função de transferência de 1 zero e 2 pólos [18] para representar o comportamento de cada sistema motor + roda:

$$g(s) = \frac{5,979 \cdot s + 0,7933}{s^2 + 3,457 \cdot s + 2,848} \quad (81)$$

Utilizando o script do *Matlab* nomeado como **Controlpidtorque** (vide Apêndice A), obteve-se a resposta ao degrau do sistema e o desenho do lugar das raízes nas Figura 44 e Figura 45, respectivamente. Para se projetar um compensador para um *overshoot* de no máximo 30%, e um tempo de estabilização de 1s, obtêm-se os valores $\xi = 0,3579$ e $\omega_n = 11,1776$, os quais são utilizados para obter o pólo dominante em $PD = -4 \pm j \cdot 10,43741$. Para cumprir a condição de ângulo, precisa-se adicionar um $\angle 25,1643^\circ$, obtendo assim o compensador dado por (ver Figura 46 para o lugar das raízes resultante):

$$C(s) = \frac{84,01 \cdot s + 2202}{s} \quad (82)$$

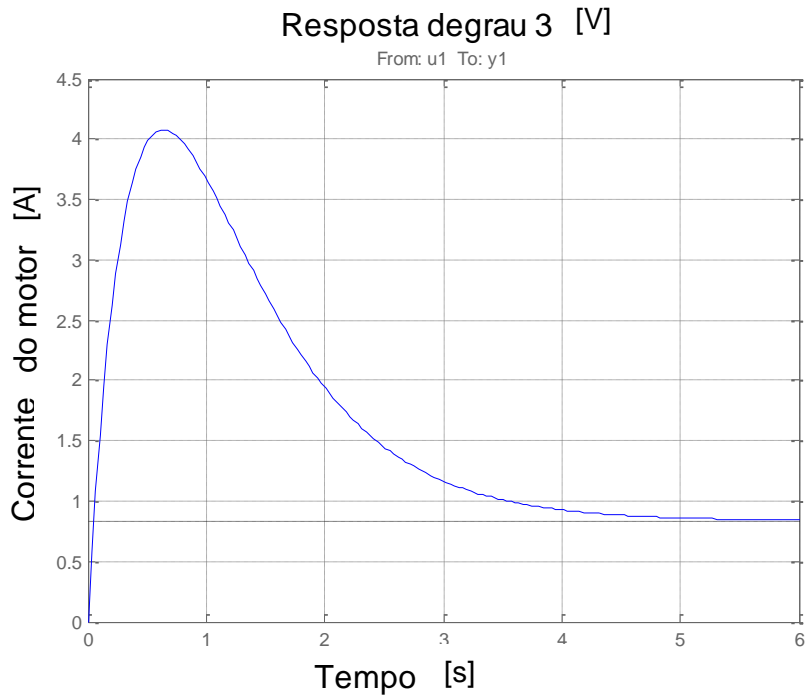


Figura 44 Resposta ao degrau de cada sistema motor-roda.

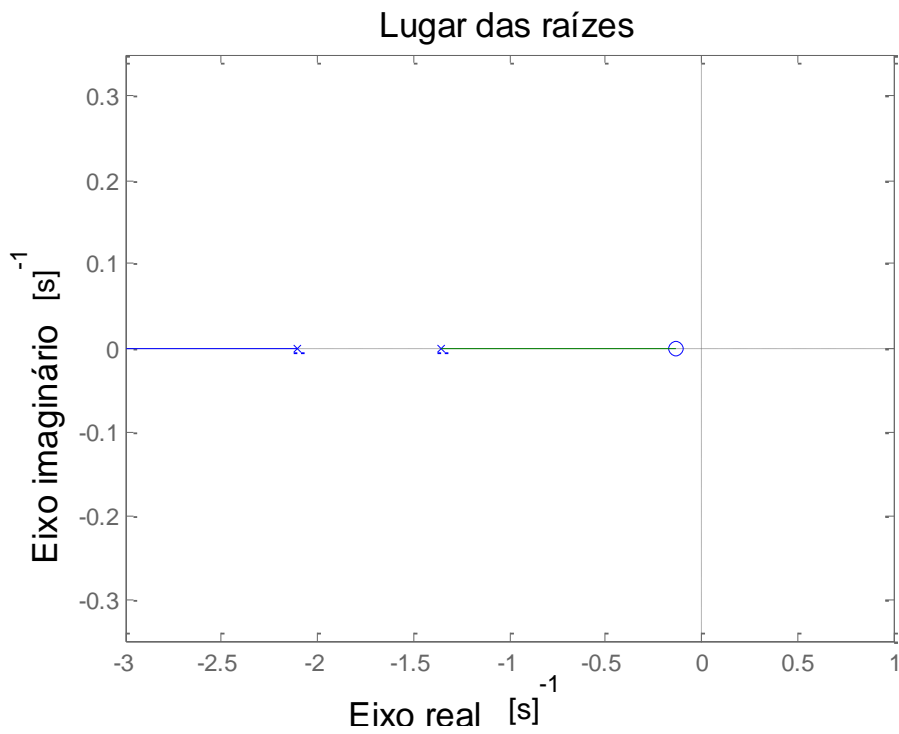


Figura 45 Lugar das raízes de cada sistema motor-roda.

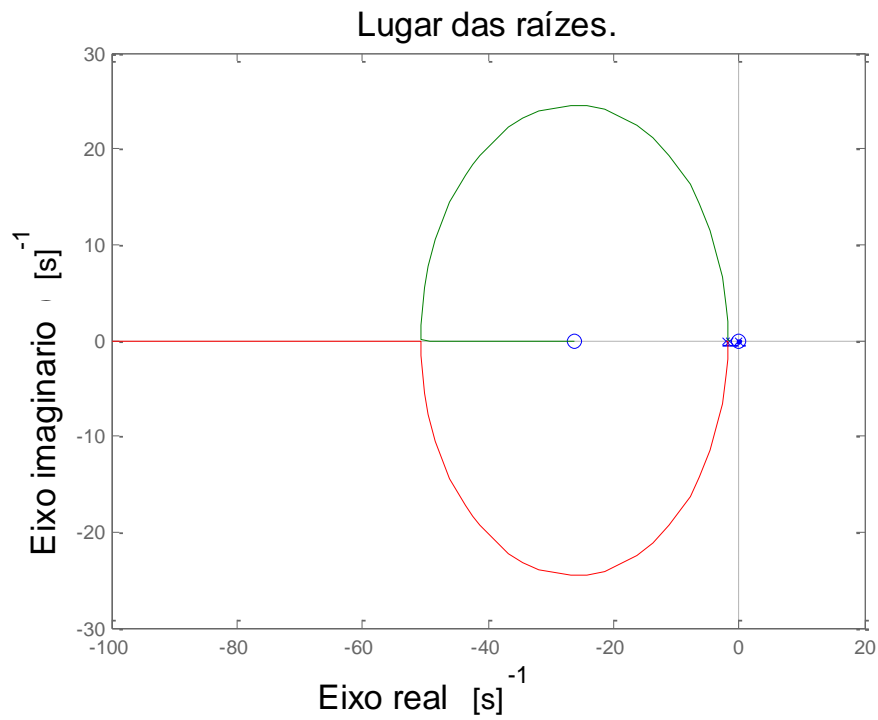


Figura 46 Lugar das raízes junto com o compensador.

O compensador projetado está associado a ganhos PI iguais a $K_p = 174,68$ e $K_i = 2269,22$, obtidos pela ferramenta *tuning tool* do Matlab. A Figura 47 mostra a corrente do motor para um degrau de torque de 4Nm, onde pode-se enxergar uma estabilização perto de 4,7A. A Figura 48 mostra os torques associados a essa resposta ao degrau, com estabilização para uma entrada de cerca de 4Nm.

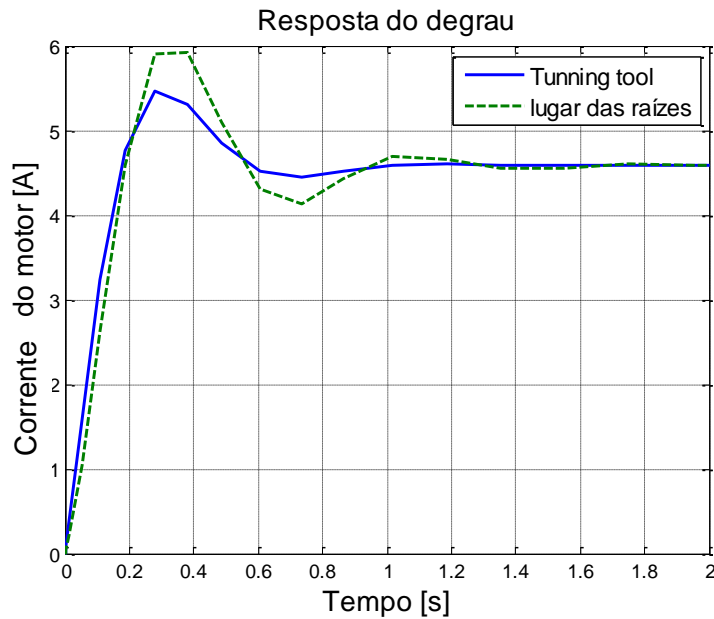


Figura 47 Resposta do controle como o lugar das raízes e tuning tool.

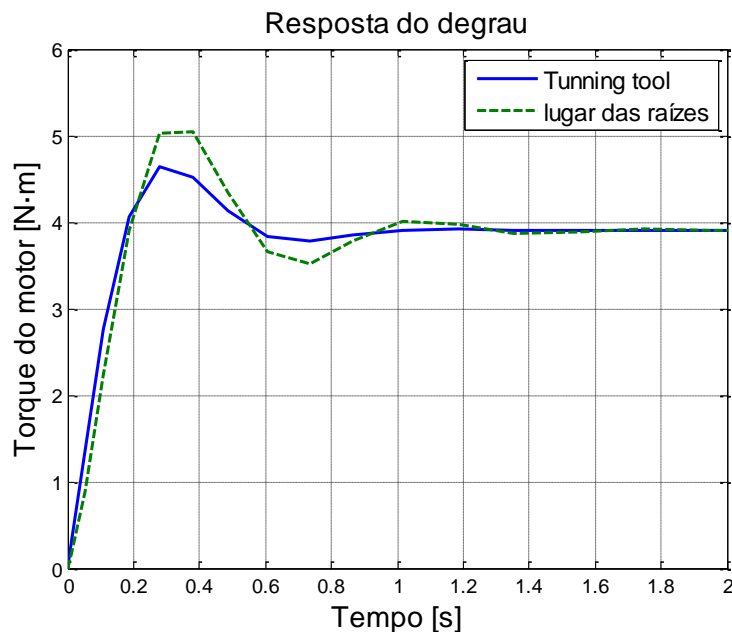


Figura 48 Resposta do controle como o lugar das raízes e tuning tool.

O diagrama de controle da bengala robótica é mostrado na Figura 49, onde tem-se o conjunto do controle baseado em admitância e o controle PI de baixo nível. Primeiro, a força do usuário é inserida no controle baseado em admitância, o qual gera uma velocidade, que é derivada para obter-se uma aceleração desejada que entrará no modelo dinâmico, que fará o cálculo do torque desejado dos motores. O torque desejado então entra no sistema de controle PI de baixo nível, e o sinal de entrada é somado ao sinal dos sensores de corrente pré-

multiplicados pela constante do motor K_e , que tem a função de converter a corrente em torque a partir do modelo do motor. Em seguida, essa soma é multiplicada pela constante K_v , que converte o sinal do erro em uma tensão elétrica aciona cada motor.

No próximo capítulo, o controle de alto nível por redes neurais é integrado na bengala robótica ao controlador mostrado na Figura 49, e os resultados experimentais são avaliados.

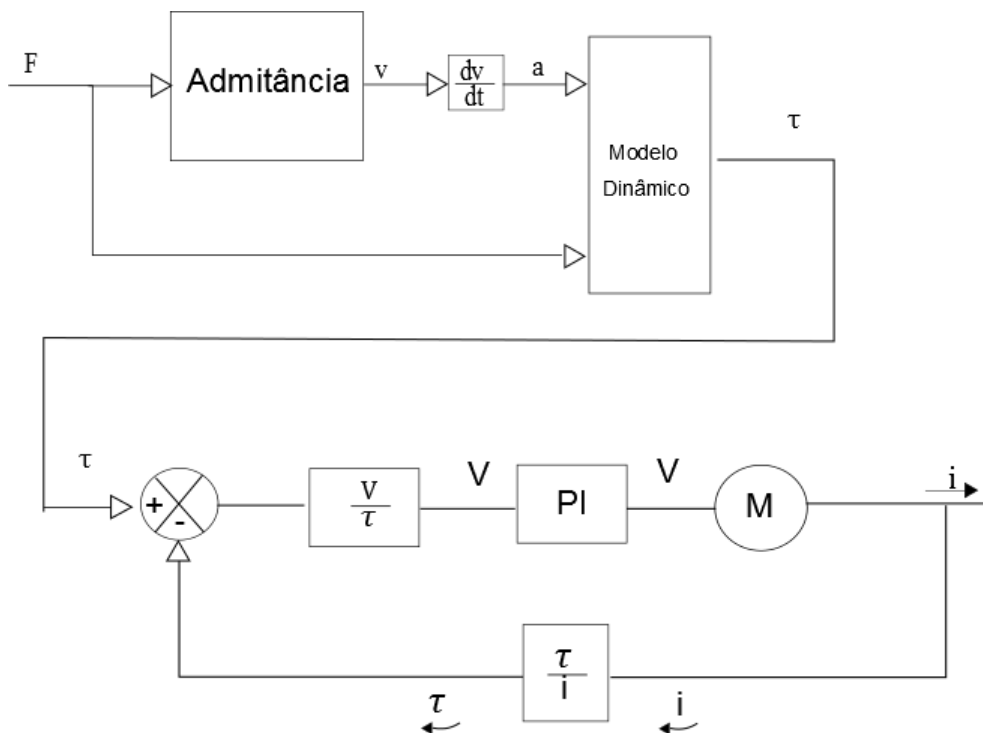


Figura 49 Diagrama de controle

4. Experimentos e Resultados

Neste capítulo são apresentados os resultados dos experimentos realizados no controle da bengala, para cada uma das etapas de aquisição, calibragem, controle de baixo nível, e controle de supervisor.

4.1. Modelagem do motor.

A modelagem do motor foi feita no Laboratório de LDC da PUC-Rio. A montagem do processo de aquisição de dados está mostrada na Figura 50. Nela, o motor está destacado com o número 1; o sistema de aquisição (DAQ) foi usado para obter dados de corrente e rotação (em RPM) do motor, destacado na figura com o número 2; e um volante de inércia de madeira de peso 210 g e de raio 5 cm y espessura 2 cm foi usado como carga, representado pelo número 3.

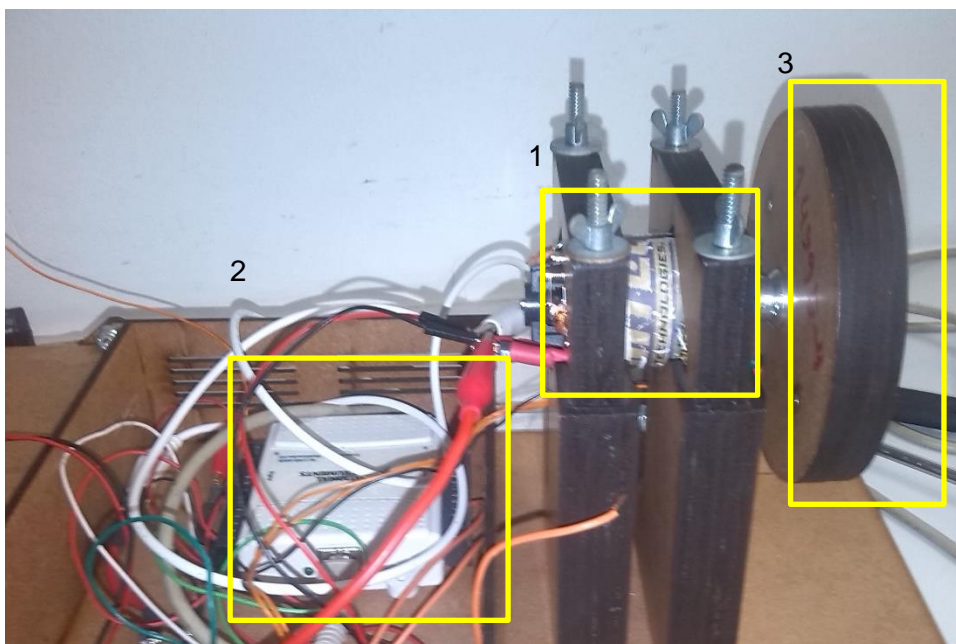


Figura 50. Sistema para a modelagem dos motores das rodas.

O processo foi realizado excitando o motor com um degrau de 3V. A Figura 51 mostra a curva da corrente do motor deste teste, com um *overshoot* de 4A,

valor muito alto se comparado com a corrente estabilizada média de 0,8A, e com um alto tempo de estabilização de 5 segundos.

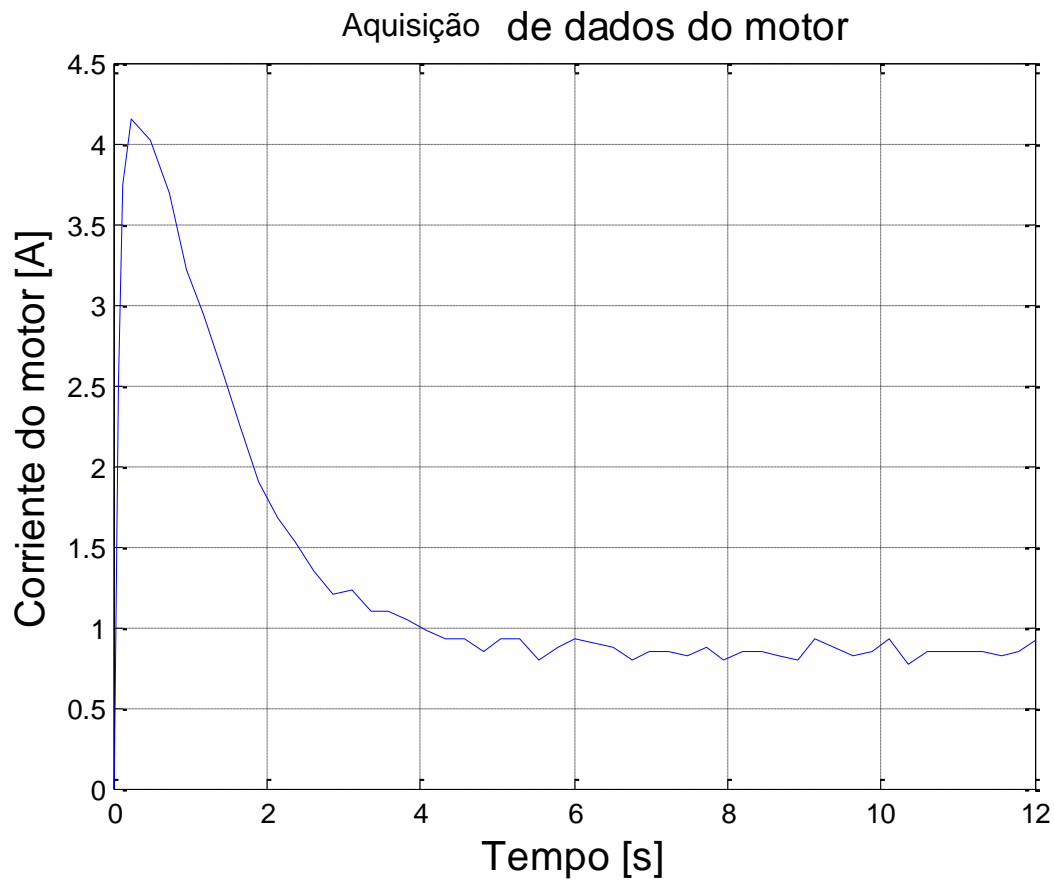


Figura 51 Corrente do motor para resposta em degrau.

A Figura 52 mostra a velocidade angular do motor, em RPM, para uma entrada em degrau de 3V. Na figura pode-se observar um valor máximo de 2600 RPM para a velocidade angular, atingido em um tempo de estabilização de 5 segundos.

Através desta aquisição, foi possível calibrar os parâmetros do motor, como mostrados na Tabela 10.

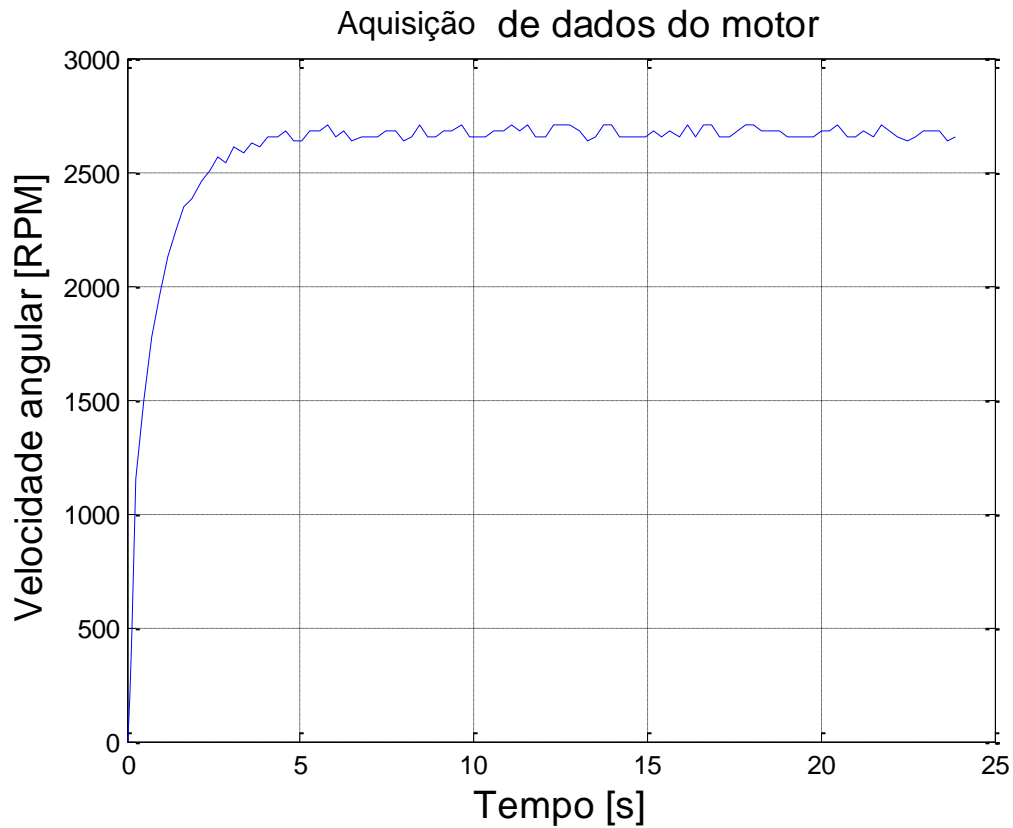


Figura 52 Rotação do motor para a resposta ao degrau.

Os seguintes dados foram o resultado do teste mencionado, onde K_v e a capacidade de geração de torque a partir da tensão em V/A , K_w representa as perdas internas eletromecânicas em $N \cdot m / rad / s$, R é a resistência interna combinada dos enrolamentos do rotor e das escovas e em Ω .

Tabela 10. Constantes identificadas do motor

Constante	Valor
K_t	$4,3 \cdot 10^1 \cdot V/A$
R	$9,5 \cdot 10^{-1} \Omega$
K_v	$1,4 \cdot 10^{-2} N \cdot m/V$
K_w	$8,6 \cdot 10^{-5} N \cdot m / rad / s$
K_i	$3,3 \cdot 10^{-1} A/V$

4.2 Controle baseado em admitância.

Utilizando o desenvolvimento do item 2.4 Controle baseado em admitância, os dados disponíveis em [19], i.e. $M = 14kg$, $B = 40 N/m$, $V = 0,25 m/s$ e $F = 12N$, além do modelo dinâmico do sistema, obteve-se a resposta ao controle por admitância apresentada na Figura 53. Nesta figura, observa-se a força aplicada pelo usuário (a entrada), e a velocidade obtida com o controle de admitância (a saída), ambas em função do tempo.

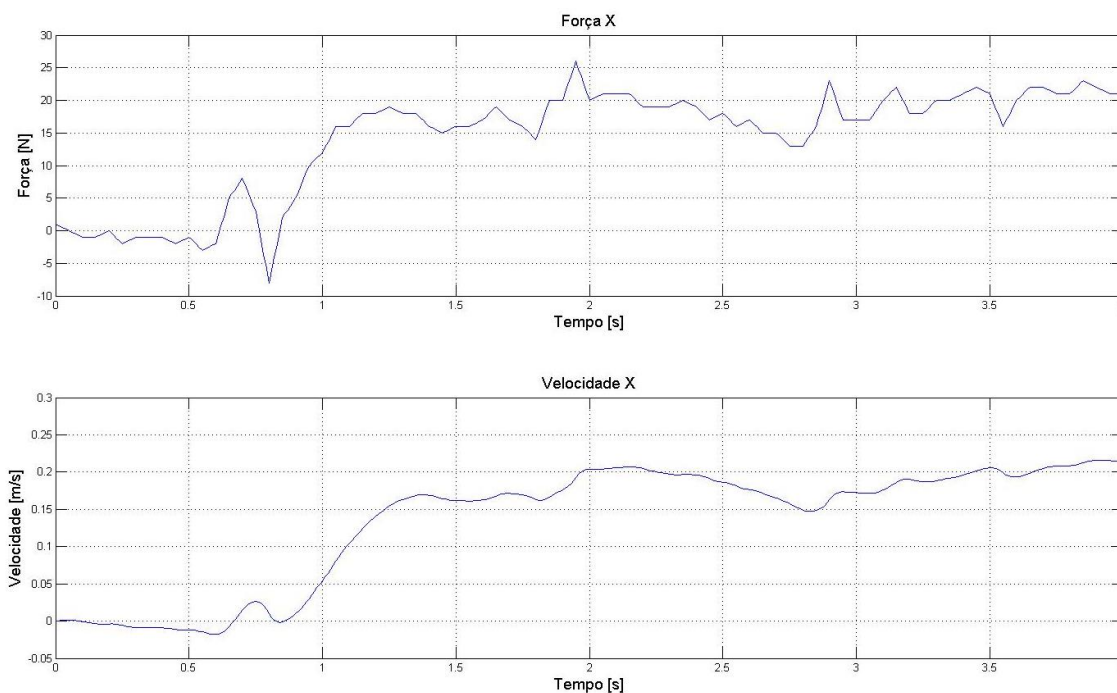


Figura 53 Entrada e saída do controle baseado em admitância.

Pode-se observar na figura que o controle baseado em admitância consegue acompanhar o perfil de forças aplicado, além de gerar uma resposta suave e pouco sensível a perturbações, atuando como um filtro passa-baixa para perturbações.

4.3 Base de dados

Cada teste executado tem uma duração de 4 segundos, com uma frequência de amostragem de 20Hz. Logo, a base de dados tem um total de 80 dados por sensor por teste. Com os 80 testes foram obtidos então 6400 entradas de dados por sensor. Desta forma, com os 10 sensores funcionais no momento dos o total da base de dados é de 64000 amostras.

Os testes de queda e caminhada foram realizados por diferentes usuários. Os usuários reproduziram as diferentes condições de caminhada, puxando ou empurrando o punho da bengala com força suficiente para movê-la em cada uma das direções. A seguir, os testes de queda foram feitos com mais força e mais velocidade, procurando uma aproximação às condições de uso. A Figura 54 mostra um teste de caminhada, onde o usuário está segurando o punho da bengala.



Figura 54 Testes com bengala.

As seguintes figuras mostram um exemplo dos dados capturados pelos sensores de deformação que foram colados na base da bengala, onde o nível de 2,5V em cada sensor significa carregamento zero, valores maiores indicam compressão e menores tração. Estes testes foram feitos com 10 sensores só, já que 2 destes estavam com defeito, Na Figura 55 mostram-se os dados que representam uma situação de caminhada para trás; a Figura 56 mostra um segundo teste de caminhada para a direita; a Figura 57 é um teste de caminhada para frente; e a Figura 58 mostra um teste de caminhada para a esquerda. Estas figuras têm no canto direito a nomenclatura mostrada na Figura 12, começando pela roseta de *strain gage* A e seguindo no sentido anti-horário até a roseta D, e em cada uma o número 1 é o *strain gage* vertical, o número 2 é o que está a 45°, e o número 3 é o horizontal.

Na Figura 55, que mostra o teste de caminhada para trás, nota-se elevada excitação dos sensores B1 e B2, como esperado pela disposição dos *strain-gages* na base da bengala. Estes se encontram portanto abaixo do nível de *offset* de 2,5V, o que indica um esforço de tração, e os sensores D2 e D1 acima de 2,5V, indicando um esforço de compressão, embora tenham uma variação um pouco menor.

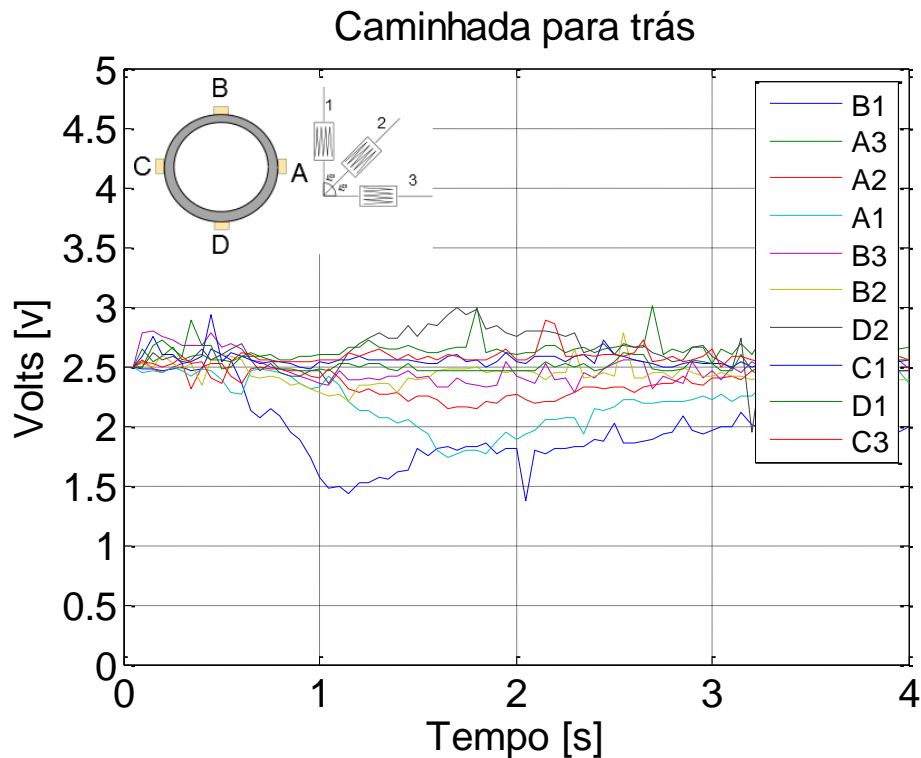


Figura 55 Teste de caminhada para trás.

Na Figura 56 percebe-se que os sensores ativados acima de 2,5 V são A1, A2 e A3, e os sensores C1 e C3 encontram-se abaixo dos 2,5V, como esperado para um movimento para a direita, indicando que a parte da direita da haste da bengala sofre um esforço de compressão, e à esquerda tração.

Na Figura 57 os valores de B1, B2 e B3 estão acima de 2,5V, e os sensores D1 e D2 abaixo de 2,5V, com variação menor. Mais uma vez, estes valores são esperados para um movimento para a frente, com a frente da haste sofrendo um esforço de compressão, e a parte posterior da bengala submetida a um esforço de tração.

Na Figura 58 pode-se observar que os sensores A1 e A2 apresentam valores abaixo dos 2,5V, significando tração. Já os sensores C2 e C3 apresentam valores acima do mesmo valor de tensão, associados a um esforço de compressão, indicando corretamente os esforços necessários para uma caminhada para a esquerda. Nota-se que o sensor B3 apresenta uma excitação, com módulo menor, mas indesejada, à qual o sistema de controle deve ser robusto.

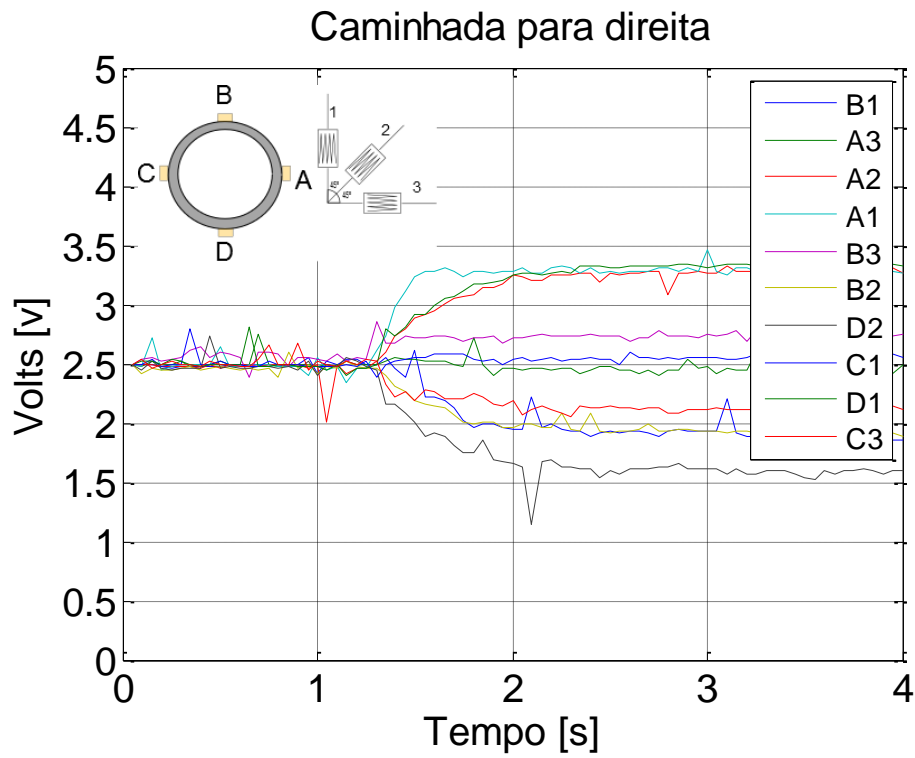


Figura 56 Teste de caminhada para a direita

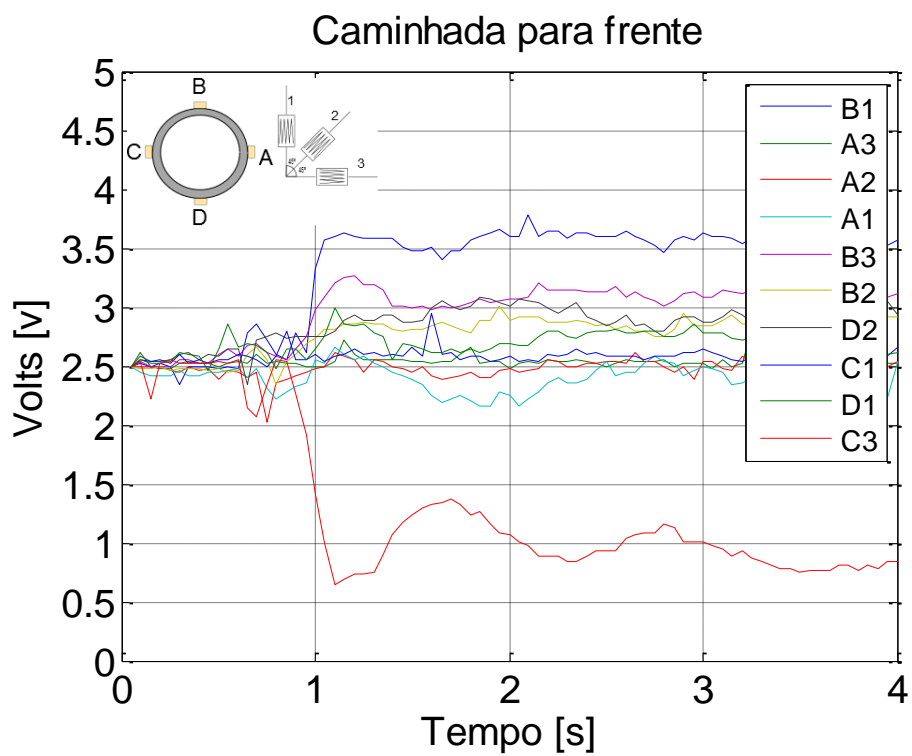


Figura 57 Teste de caminhada para frente

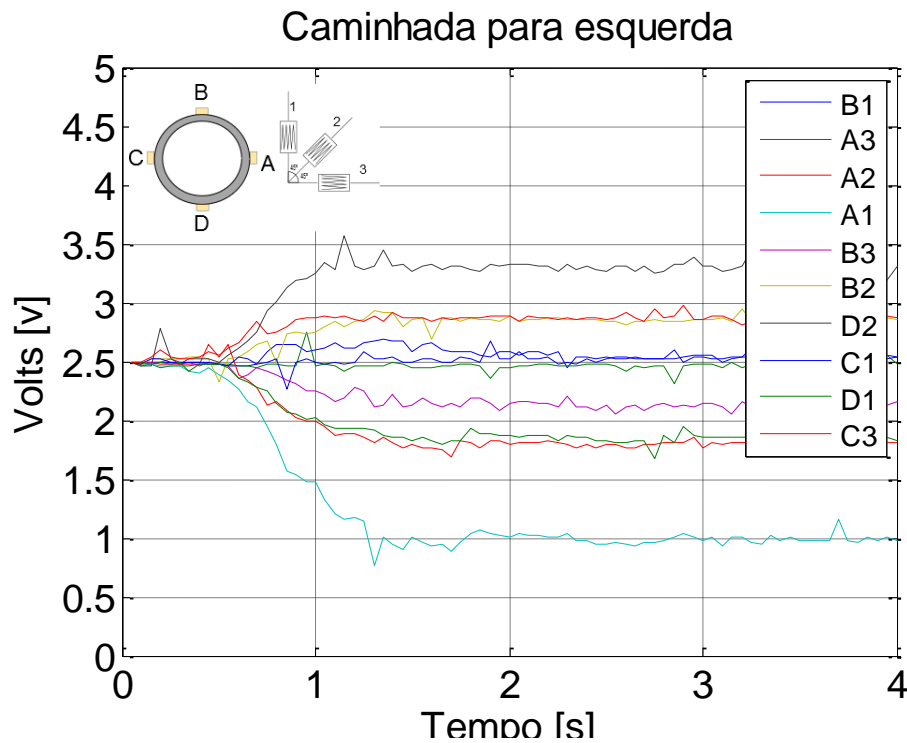


Figura 58 Teste de caminhada para a esquerda

As próximas figuras mostram os testes de queda. Na Figura 59, tem-se um teste de queda para trás; na Figura 60 para a direita; na Figura 61 para frente; e, por último, na Figura 62 se mostra um teste de queda para a esquerda.

A Figura 59 mostra os valores dos sensores B1, B2 e B3 com uma forte resposta de excitação abaixo dos 2,5V, o que representa um esforço muito grande de tração, indicando uma queda iminente para trás. Na Figura 60, observam-se os valores obtidos pelos sensores A1 e A2, ambos acima de 2,5V, com uma forte perturbação, e os valores de C1 e C3 abaixo dos 2,5V, correspondendo a uma queda iminente para a direita. Na Figura 61 observa-se uma forte perturbação nos sensores B1 e B3, com valores muito acima de 2,5V, associados a uma queda iminente para a frente. E, finalmente, a Figura 62 mostra os valores para os sensores C3 e C1 com tensões acima de 2,5V, mostrando que estão sob esforço de compressão, e os sensores A1, A2 e A3 com valores bem menores que 2,5V, mostrando que estão sob um esforço de tração, todos coerentes com uma queda iminente para a esquerda. Apesar dos ruídos, característicos dos *strain-gages*, todas as diferentes condições de caminhada e iminente queda possuem padrões e intensidades bem diferentes, que podem ser identificados em um sistema supervisor.

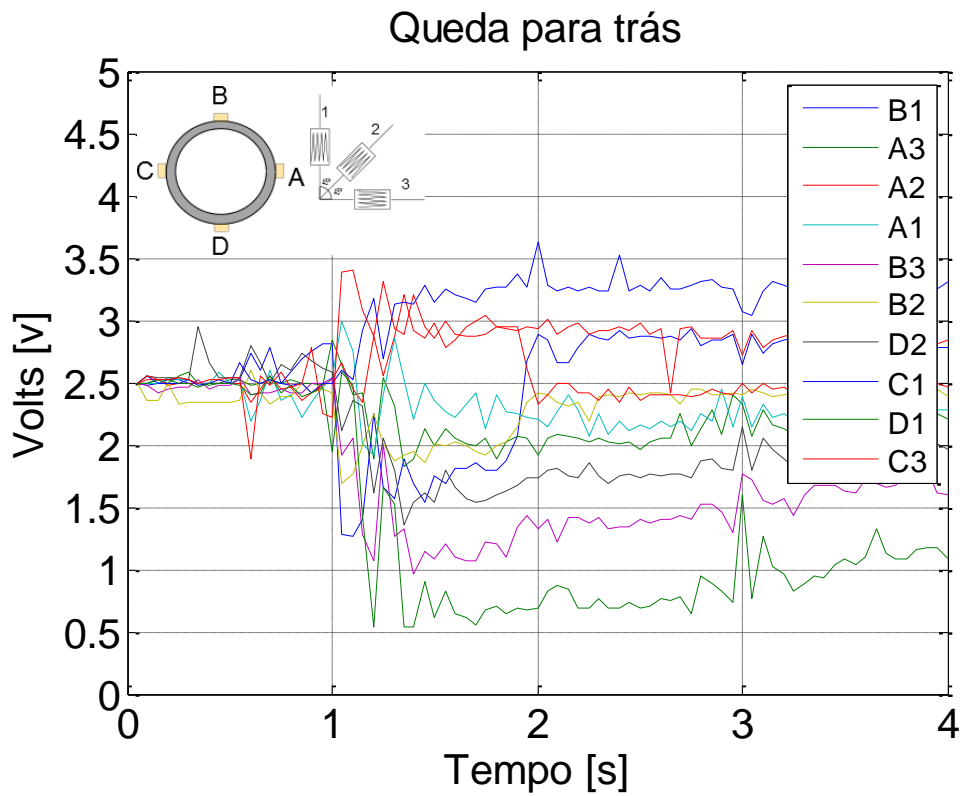


Figura 59 Teste de queda para trás.

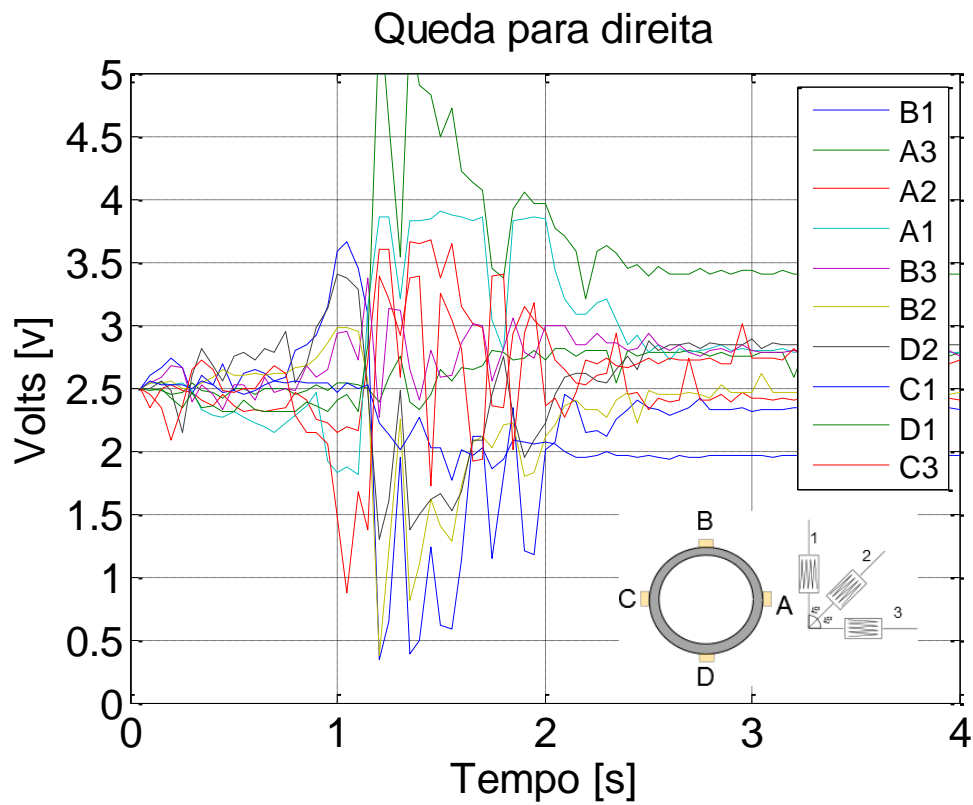


Figura 60 Teste de queda para a direita.

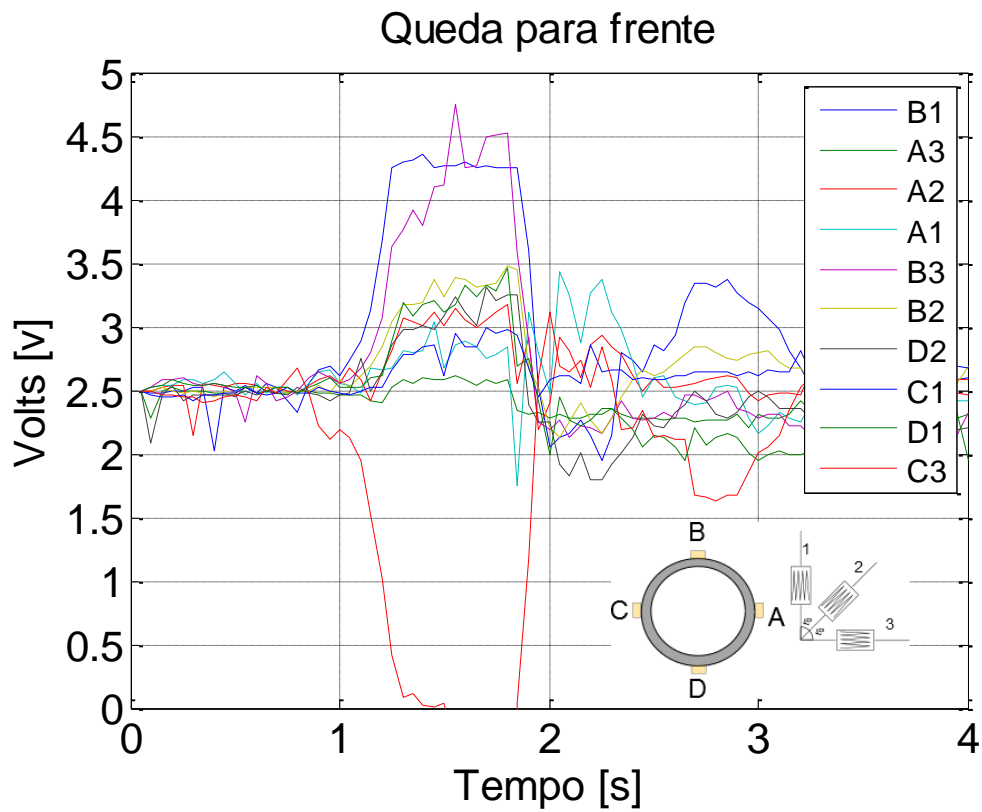


Figura 61 Teste de queda para frente.

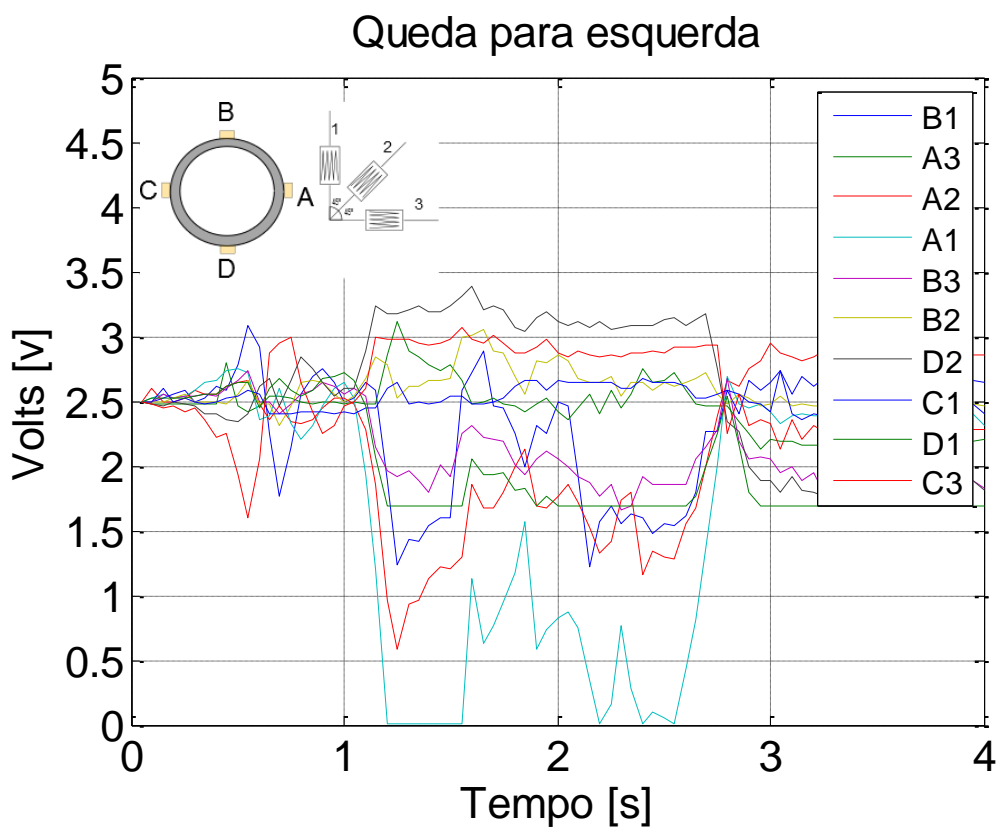


Figura 62 Teste de queda para a esquerda.

4.4. Controle de baixo nível.

Com os valores dos ganhos PID obtidos no capítulo anterior, foram feitos testes de controle nos motores. Nestes testes, coloca-se a bengala no chão e impõe-se um torque inicial de 4Nm para um dos motores, gerando a resposta da Figura 63. Nesta figura são comparados os valores reais da resposta do motor e os valores simulados com a ferramenta *tuning tool* do Matlab. Pode-se observar que a resposta do real do motor tem um *overshoot* um pouco maior, provavelmente devido à frequência relativamente baixa do controlador discreto adotado, porém com um tempo de estabilização similar.

A Figura 64 mostra a comparação dos erros de torque para o controlador real e para o simulado com a ferramenta *tuning tool* do Matlab. Nota-se mais uma vez um erro do motor real um pouco maior que o da simulação, porém com tempo de estabilização similar e erro residual próximo de zero em regime permanente.

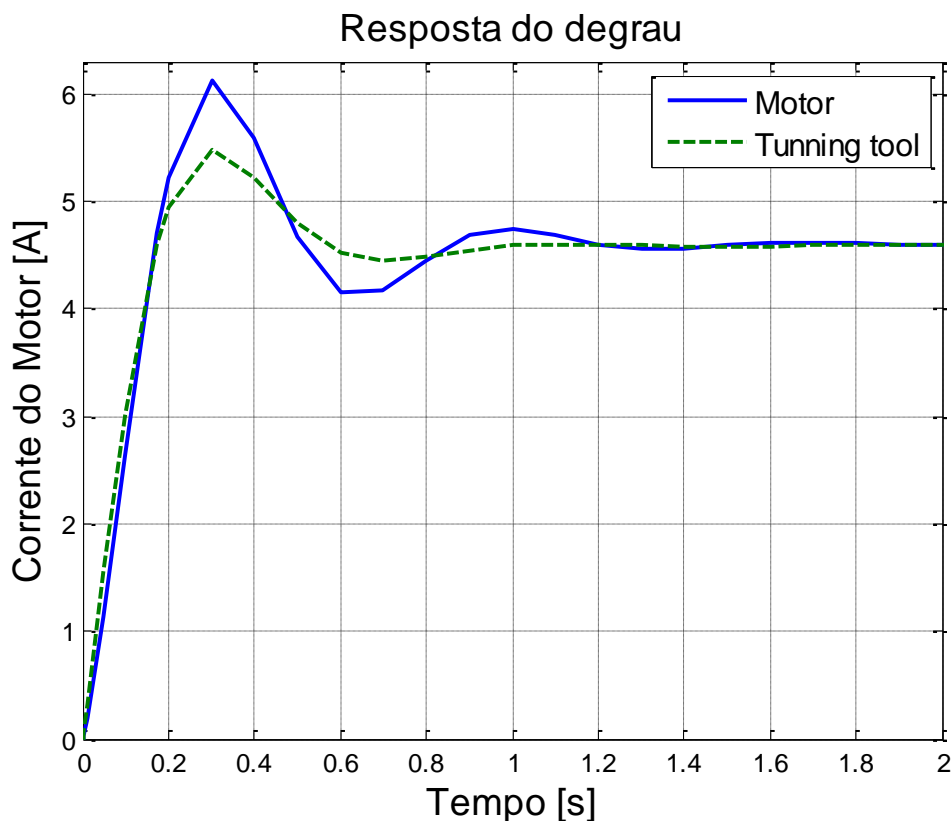


Figura 63 Comparação de corrente entre o motor real e o simulado.

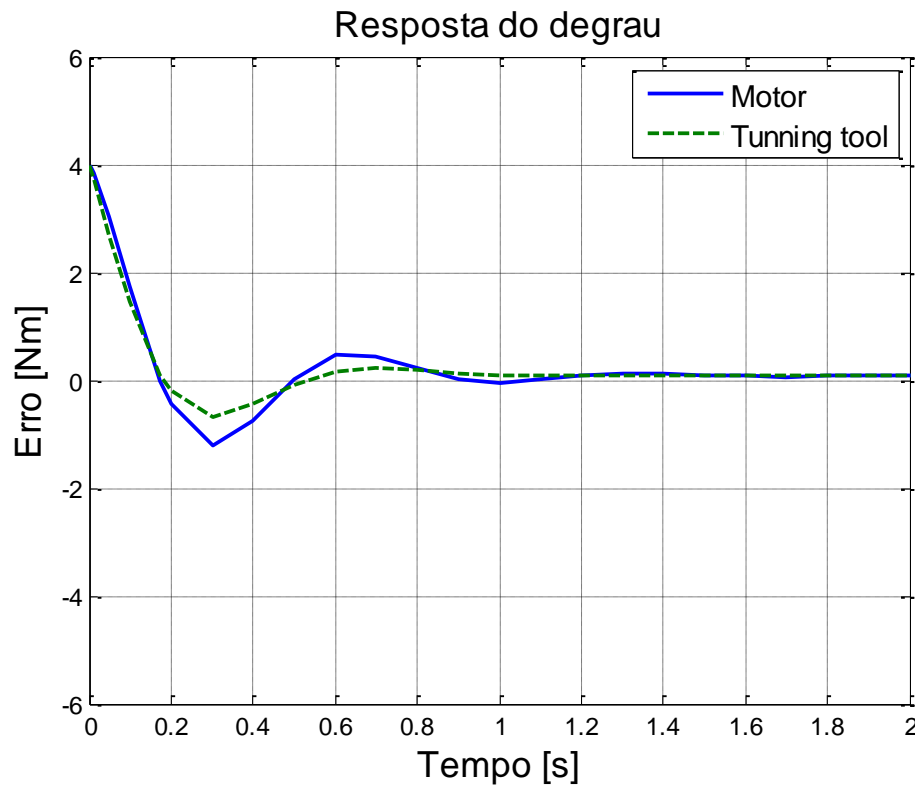


Figura 64 Comparação de erro de torque entre o motor real e o simulado.

A Figura 65 mostra uma comparação de corrente entre o motor em malha aberta e com o controlador. Pode-se observar uma diferença de mais de 70% na corrente de pico do motor ante uma excitação em degrau de cerca de 4Nm, além de mais de 3s de diferença de tempo de estabilização, justificando a necessidade do controlador para um desempenho adequado.

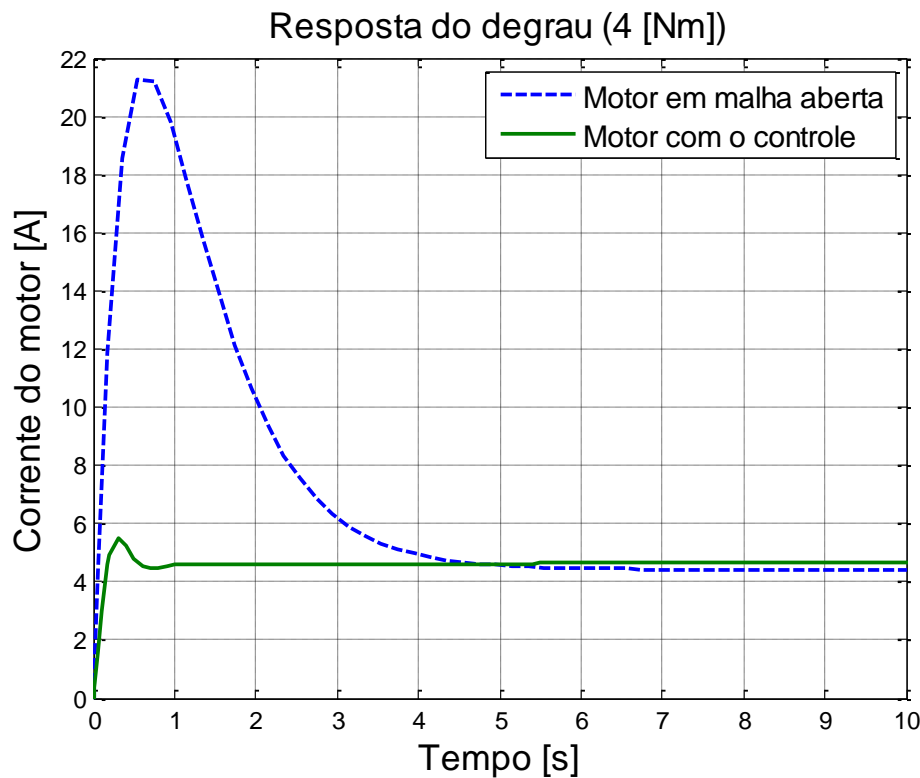


Figura 65 Comparação experimental da resposta ao degrau do motor com o controlador e em malha aberta.

4.5. Controle de alto nível ou supervisor

No Apêndice B encontra-se o arquivo em *Matlab* desenvolvido para a realização do teste de controle de alto nível. Primeiramente, os dados são importados usando a função *importados*. Esta traz os dados dos arquivos de extensão *.txt* e os organiza em uma matriz de três dimensões. Há duas matrizes “camina” e “caída” (caminhada e queda, respectivamente), de 80x10x40 elementos, as quais incluem 80 dados ao longo do tempo para 10 sensores, para os 40 testes.

O segundo passo consiste em gerar um *offset* para inicialmente zerar os dados. No apêndice é apresentada a função *zera*, a qual usa os dados do primeiro instante da janela de aquisição e o toma como referência inicial igual a zero. Este procedimento foi realizado para cada uma das matrizes descritas.

O terceiro passo consiste em ordenar as duas matrizes anteriores em duas estruturas. No apêndice são apresentadas as funções *arreglacamina* e

arreglada, que cuidam deste procedimento. Além disso, a função *arreglada* usa somente os dados que ocorrem no evento da queda.

No quarto passo, há a função *pend* (vide apêndice), a qual é destinada a aplicar a (79)), que pode ser trocada pela função *promedio*. Esta função tem por objetivo calcular a média seguindo a (80)), dependendo do modelo a ser utilizado. No quinto passo, a função *target* cria o objetivo dando um número de 1 até 4, dependendo da classe, como visto na Tabela 8. No sexto passo, organiza-se os dados dos grupos de treinamento, teste e validação em porcentagens. Neste caso, foram 80%, 10% e 10%, respectivamente, deixando assim 2560 dados para teste, 320 para teste e 320 para validação. As funções *organiza1* e *organiza1camina* realizam este procedimento, vide Apêndice B.

O sétimo passo tem por objetivo unir cada um dos grupos de treinamento, teste e validação de queda e de caminhada na entrada, para em seguida criar os mesmos grupos para o *target*. No oitavo passo, a função *nvectoriza* faz a codificação binária, como se pode observar na Tabela 8. Além disso, com esta função normaliza-se os valores dos dados. Esta função é aplicada a todos os grupos de entrada para a rede neural. No nono passo, os dados são renomeados para poder apresentá-los à rede neural. Neste passo, utiliza-se a função *newff* do *Matlab*, estes passos são mostrados no diagrama de fluxo na Figura 66

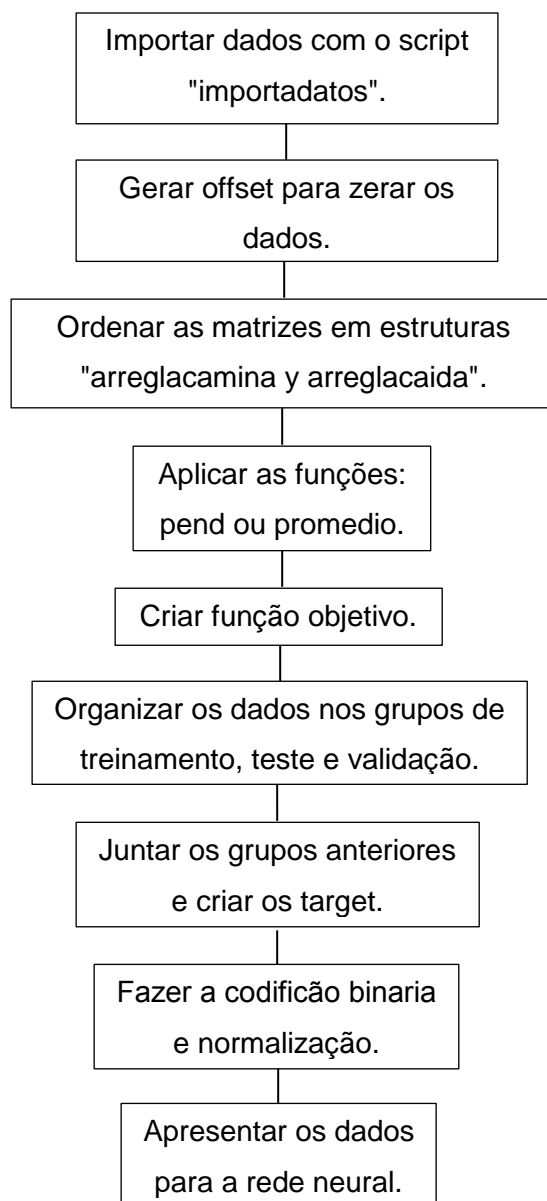


Figura 66. Diagrama de fluxo do controle de alto nível.

4.5.1 Primeira fase

A seguir, mostram-se os resultados dos modelos das redes neurais para a primeira fase, descrita no Capítulo 3. A definição das redes adotadas envolve o tipo de arquitetura, a quantidade de neurônios e de suas camadas ocultas, a porcentagem de acertos com as amostras de validação, e o valor de erro do MSE, calculado levando em consideração a média de 50 experimentos para uma mesma arquitetura de redes neurais.

Os valores de neurônios nas camadas ocultas utilizadas nos modelos foram determinados com procedimentos heurísticos, iniciando-se com valores comuns na área, tais como:

$$N_h = \frac{\#_{in} + \#_{out}}{2} = 6 \quad (83)$$

$$N_h = \#_{in} = 10 \quad (84)$$

$$N_h = \#_{out} = 2 \quad (85)$$

$$N_h = \#_{in} + \#_{out} = 12 \quad (86)$$

$$N_h = 2 \cdot \#_{in} = 20 \quad (87)$$

onde N_h é o número de camadas ocultas, $\#_{in}$ é o número de entradas da rede, e $\#_{out}$ é o número de saídas da rede. Dependendo dos resultados, aqueles que apresentaram melhores valores foram explorados na sua vizinhança, visando escolher os melhores para continuar nos testes seguintes.

Tomando como referência a resposta do modelo 1, que é a rede que tem uma quantidade duas camadas de 10, nota-se que a porcentagem de acerto das redes com 6 e 10 neurônios é menor (85,2 e 87%, respectivamente), mostrando uma má generalização da rede neural. Da mesma maneira, observam-se os valores da resposta das redes neurais com duas camadas de 20 neurônios. O modelo número 2 segue a mesma tendência que se obteve no modelo 1: um bom desempenho da rede com duas camadas de 10 neurônios, e desempenho pior nas redes com menor número de neurônios, diminuindo a porcentagem de acerto nas redes maiores.

O modelo número 3 foi o de melhor resultado dos três. Este apresenta a porcentagem de acerto de 100% em cada uma das redes testadas. Foram selecionados os valores dos neurônios seguindo a tendência dos anteriores (modelos 1 e 2), desprezando as redes de uma camada com 6 e de duas camadas de 20 neurônios pelo seu baixo desempenho. Estes valores

apresentam uma configuração para inferir o estado de queda ou caminhada. Os testes anteriores foram utilizados para ter uma ideia geral do comportamento da rede neural para os dados apresentados.

Na Tabela 11 apresenta-se o resultado da primeira fase do sistema supervisor. Observa-se que o modelo 3 apresentou um bom desempenho, assim escolheu-se a configuração com 10 neurônios por apresentar um menor custo computacional.

Tabela 11. Resumo da primeira fase do sistema supervisor.

Resumo				
Modelo	Neurônios camada 1	Neurônios camada 2	% Acerto validação	Mse
1	6	-	85,2%	0,1121
	10	-	87%	0,1111
	10	10	89,1%	0,0927
	20	-	82,6%	0,1521
	20	20	84,8%	0,1180
2	6	-	83,4%	0,1312
	10	-	84,8%	0,1136
	10	10	89,1%	0,0857
	20	-	84,8%	0,1239
	20	20	87%	0,1087
3	10	-	100%	$7,2896 \times 10^{-9}$
	10	10	100%	$2,5843 \times 10^{-4}$
	20	20	100%	0,0034

4.5.2 Segunda fase

A Tabela 12 apresenta o resultado da segunda fase do sistema supervisor, descrita no Capítulo 3. Esta fase tinha como objetivo achar uma configuração para inferir a direção do estado de caminhada ou queda. Nesta fase, utilizou-se de igual forma a mesma linha de raciocínio para estabelecer o número de neurônios nas camadas ocultas. Neste caso, contando com 10 valores de entradas e 4 valores de saída. De forma similar, os valores de MSE,

apresentados na Tabela 12, foram calculados tomando em conta a média de 50 experimentos para a mesma arquitetura de redes neurais, sendo estas as configurações de arquitetura mais representativas.

Observa-se que a rede número 15, que tem 28 neurônios nas 3 camadas, apresentou um ótimo desempenho, atingindo um 85,3% de acerto, e um MSE de 0,0959. Da mesma maneira, tem-se um bom desempenho na rede neural número 13, a qual atingiu uma percentagem de acerto de 81,3% e um MSE de 0,1007. A rede número 3, com quantidade de neurônios igual a 10 nas 3 camadas, apresenta uma percentagem 73,3% de acerto e um valor de MSE de 0,1010. Procurando perto desse tamanho de rede, encontrou-se a rede 5, com o tamanho de 11 neurônios nas duas camadas, o qual atingiu 77,3% de acerto e um valor de MSE 0,0957. Desta forma escolheu-se rede número 5, cuja configuração apresenta um menor custo computacional, compensando o valor de percentagem de acerto com a velocidade de computação.

Tabela 12. Desempenho da segunda fase do sistema supervisor.

Resumo					
Nº	Neurônios camada 1	Neurônios camada 2	Neurônios camada 3	% Acerto validação	Mse
1	10	-	-	70,3%	0,1068
2	10	10	-	69,3%	0,1125
3	10	10	10	73,3%	0,1010
4	11	-	-	66,7%	0,1020
5	11	11	-	77,3%	0,0957
6	11	11	11	76,0%	0,0988
7	15	-	-	65,3%	0,0991
8	15	15	-	76,0%	0,0911
9	15	15	15	73,3%	0,0915
10	20	-	-	64%	0,1068
11	20	20	-	69,3%	0,1083
12	20	20	20	69,3%	0,1010
13	28	-	-	81,3%	0,1007
14	28	28	-	72,0%	0,0950
15	28	28	28	85,3%	0,0959

A Figura 67 apresenta os valores estudados acima de forma gráfica, além de mostrar alguns outros valores que, pelo seu baixo desempenho, não foram incluídos na Tabela 12.

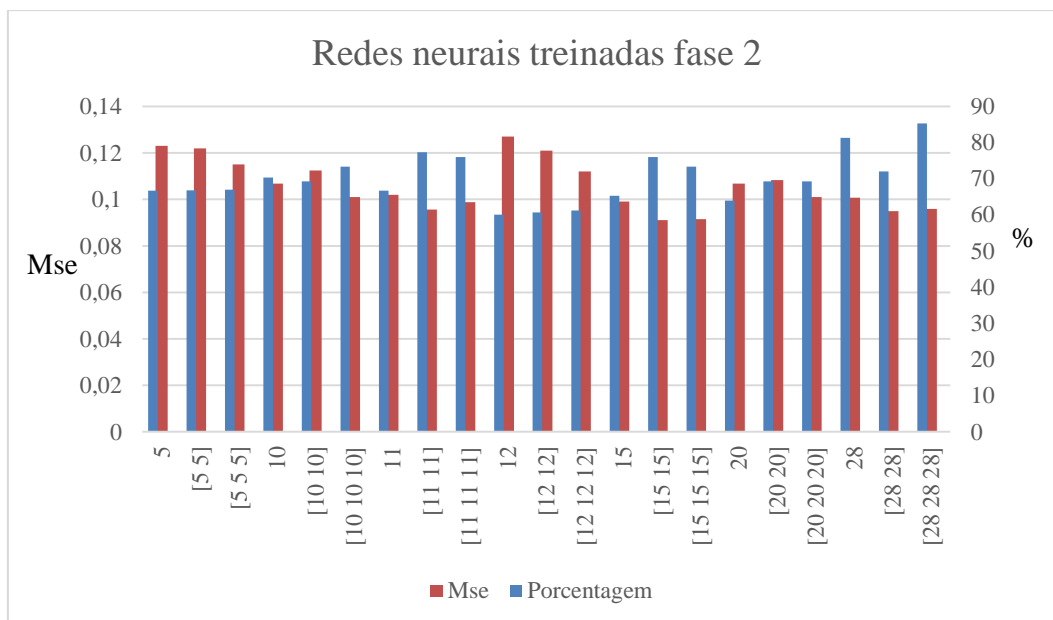


Figura 67. Gráfico com o desempenho da segunda fase do sistema supervisor.

A partir das análises anteriores, foi possível escolher redes neurais adequadas para a identificação do estado do usuário. Quando integrado com os controles por admitância e PID, este sistema supervisor permite que a bengala robótica desempenhe seu papel de suportar o usuário durante a caminhada, e identificar situações de queda. No entanto, o controle para compensação de queda não fez parte do escopo dessa dissertação.

5. Conclusões e Recomendações

No presente trabalho foi realizado o projeto de um protótipo funcional de uma bengala omnidirecional robótica. Com esse fim, implementou-se o controle baseado em admitância para suavizar os movimentos a partir de entradas de força e torque do usuário, e um controle de torque nos motores para garantir a estabilização do movimento nas velocidades desejadas. Neste trabalho utilizou-se uma rede neural para reconhecer a intenção de movimento do usuário, e identificar situações de caminhada normal e de queda iminente no seu deslocamento.

As principais conclusões derivadas do projeto da bengala robótica omnidirecional são:

1. Construiu-se um protótipo de bengala omnidirecional robótica funcional, utilizando sensores de baixo custo. Isto foi possível pois não foi necessário utilizar sensores de força e torque multiaxiais comerciais, que são aproximadamente 100 vezes mais caros que a solução adotada, perto dos 8,000 dólares. Em seu lugar, foi usado um arranjo de *strain gages* (extensômetros) para identificar forças e torques impostos pelo usuário. Na construção da bengala, foram usados materiais de baixo custo como chapas de alumínio naval, e acessórios de fácil acesso, cumprindo com o objetivo de oferecer suporte para pessoas com deficiências motoras através de uma boa relação custo-benefício.
2. Os resultados com as redes neurais adotadas mostraram que este método pode ser usado para inferir a direção e sentido das forças aplicadas em uma bengala, mostrando uma porcentagem de acerto nas decisões em torno de 80%. Contudo, devido ao tamanho da rede que se treinou, propõe-se em trabalhos futuros o uso de um melhor processador que o PIC para a realização dos cálculos, já que o custo computacional é relativamente alto.
3. Ao implementar o controle baseado em admitância, e após a devida análise da resposta, percebe-se que este trabalha como um filtro passa-baixa. A frequência de filtragem pode ser modificada aumentando o valor do amortecedor, o qual pode ser usado para fazer calibragem para diferentes usuários com diferentes

intensidades de deficiências motoras, evitando movimentos indesejados e proporcionando uma sensação de suavidade e conforto no movimento.

4. O controle PI de baixo nível implementado consegue estabilizar o movimento na situação de caminhada. O sistema poderia alcançar uma resposta mais rápida se uma frequência de controle maior for utilizada, porém maiores ganhos poderiam saturar os motores.
5. Utilizando as redes neurais, corroborou-se que não é necessário utilizar os 12 *strain gage* inicialmente proposto para inferir a direção e sentido da força do usuário ao interagir com a bengala. Isto porque apenas com os dados adquiridos por 10 sensores no momento do treinamento foi possível inferir direção e sentido com ajuda das redes neurais.
6. O modelo matemático utilizado no motor precisa ser reavaliado, já que o apresentado no capítulo 4.1 está sujeito à perturbações nos testes experimentais que não foram consideradas. Recomenda-se fazer uma maior quantidade de testes, de acordo com um procedimento bem definido e um tratamento mais rigoroso dos resultados obtidos, de modo a ter um modelo mais representativo e confiável desse componente.
7. Verificou-se na implementação do sistema de aquisição de dados que é importante uma boa etapa de condicionamento para evitar erros induzidos na leitura e processamento.
8. Observando a resposta do controle, percebe-se que foi diminuído o consumo de corrente pico do motor em 72% aproximadamente, diminuindo o consumo de potência e evitando possíveis danos ao motor, aumentando assim a vida útil, além de evitar um possível dano às baterias.

Propõem-se para futuros trabalhos as seguintes ações:

1. O uso de um dispositivo móvel, e.g. miniPcs ou até telefones celulares (*smartphones*), para monitorar de forma remota o estado da bengala, o histórico de esforços dos motores, e para contabilizar o estado de carga da bateria. Além disso, o dispositivo poderia monitorar o ritmo cardíaco da pessoa e alertar sobre acidentes do usuário.

2. Usar sensores de distância (e.g. sensores de ultrassom ou infravermelhos) para monitorar possíveis obstáculos e evitar colisões.
3. Propõe-se utilizar outros métodos diferentes para inferir a intensão do usuário, tais como logica fuzzy ou programação genética.

6. Referências Bibliográficas

- [1] HIRATA,Y.;BABA,T.;KOSUGE,K., Motion control of omni-directional type walking support system “Walking Helper,” in *The 12th IEEE International Workshop on Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003.*, 2003, 85–90p. .
- [2] YU,Haoyong.;SPENKO,Matthew.;DUBOWSKY,Steven., Omni-Directional Mobility Using Active Split Offset Castors, *J. Mech. Des.*, vol. 126, no. 5, 2004, 822p. .
- [3] YANG,Yi-che Huang Hsiang-ping.;YOUNG,Chun-hsu Ko Kuu-young., Human Intention Recognition for Robot Walking Helper Using ANFIS, no. 3, 2011, 311–316p. .
- [4] DUBOWSKY,S. et al., PAMM - a robotic aid to the elderly for mobility assistance and monitoring: a “helping-hand” for the elderly, *Proc. 2000 ICRA. Millenn. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. (Cat. No.00CH37065)*, vol. 1, 570–576p. .
- [5] HUANG,Jian. et al., Motion control of omni-directional type cane robot based on human intention, in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2008*, 273–278p. .
- [6] WAKITA,Kohei. et al., Human-Walking-Intention-Based Motion Control of an Omnidirectional-Type Cane Robot, *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 1, Feb. 2013, 285–296p. .
- [7] DI,Pei. et al., Optimal posture control for stability of intelligent cane robot, *2012 IEEE RO-MAN 21st IEEE Int. Symp. Robot Hum. Interact. Commun.*, Sep. 2012, 725–730p. .
- [8] strainpic1.jpg (383x219). [Online]. Available: <http://www.omega.com/literature/transactions/volume3/images/strainpic1.jpg>. [Accessed: 03-Aug-2015].
- [9] FERDINAND P,Beer.;E. RUSSELL,Johnston .Jr.;DEWOLF T,John., *MECÁNICA DE MATERIALES*. 2004.
- [10] DRISCOLL,Frederick F., *Amplificadores operacionales y circuitos integrados lineales*. Pearson educacion., 1999.
- [11] ALBERT,Paul Malvino., *Principios De Electrónica*. McGraw-Hill, 2000.
- [12] OLABE,Xabier Basogain., *Redes Neurnales Artificiales y sus Aplicaciones*. Escuela Superion de ingenieria de Bilbao, 2014.
- [13] HAYLIN,Simon., *Neural Networks*, Second Edi. Upper Saddle River,New Jersey: Prentice-Hall, Inc., 1999.
- [14] OGATA,Katsuhiko., *Ingeniería de control moderna*. 2003.
- [15] NORMAN S,Nise., *Sistemas De Control Para Ingenieria*. Grupo PATRIA CULTURAL , S.A DE C.V, 2006.
- [16] Man-Systems Integration Standards (MSIS). [Online]. Available: <http://msis.jsc.nasa.gov/>. [Accessed: 02-Mar-2015].
- [17] ANTHROPOMETRY AND BIOMECHANICS. [Online]. Available: <http://msis.jsc.nasa.gov/sections/section03.htm>. [Accessed: 25-Jul-2015].
- [18] CORP,Electro-Craft., *DC Motors · Speed Controls · Servo Systems*, Fifth. Minesota: Electro-Craft Corporation, 1980.

- [19] YU,Haoyong.;SPENKO,Matthew.;DUBOWSKY,Steven., An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly, 2003, 53–66p. .
- [20] DIMENSION ENGINEERING., Sabertooth 2x60 User’s Guide. 25p. .

Apêndice A

Neste apêndice, são apresentados os detalhes dos circuitos eletrônicos implementados na bengala robótica, além de conter uma tabela com o resumo dos componentes mecânicos da bengala.

Os circuitos estão divididos em quatro placas de 100x200mm, as quais estão divididas de acordo com o papel a desenvolver no projeto final. Estes circuitos foram divididos para facilitar a revisão na procura de erros no funcionamento, e para o fácil entendimento do mesmo. As placas foram dispostas uma acima da outra formando múltiplas camadas, onde cada camada desenvolve uma atividade diferente. O projeto do circuito foi iniciado no software *Proteus*. Com o software *Ispis* foram desenhados e simulados os testes para cada um dos processos, fizeram-se testes na ponte de *Wheatstone*, nos valores de alimentação elétrica dos amplificadores, nos valores de ganho, e testes dos filtros. Também foi utilizado o software *Ares* para escolher o posicionamento e a orientação dos elementos eletrônicos.

A primeira placa de circuito elétrico, como mostra a Figura 68, é a parte de aquisição onde estão todas as pontes de *Wheatstone* junto com o circuito de balanço, onde as resistências P_{x01} , P_{x02} e P_{x04} são os braços da ponte do sinal x , ou seja, $P101$ é a primeira resistência da ponte 1, $P201$ é a primeira resistência da ponte 2, e assim por diante, onde todas são de valor 120Ω e de alta precisão. Ao total, são 36 unidades.

A Tabela 13 mostra um resumo da quantidade de itens contidos no circuito 1. R38-R49 e RV13-RV24 são as resistências de balanço das 12 pontes, respectivamente. A resistência R38-R49 são resistências fixas de $100K\Omega$ e as resistência RV13-RV24 são *trimpots* de $1K\Omega$.

O segundo circuito elétrico (Figura A.2) encontra-se na segunda e terceira camadas, constituído por 4 amplificadores operacionais TL084 ($U1-U4$), os quais têm 4 amplificadores nomeados de A até D, onde os amplificadores A, B e D formam o amplificador de instrumentação descrito na seção 2.2.3. Há também seis resistências de $100K\Omega$ ($R_{x01}-R_{x06}$) por sinal, uma resistência variável de 100Ω (R_{x07}), a qual varia o ganho do amplificador operacional, e duas resistências de $1K\Omega$ (R_{x08} e RV_x variável), as quais mudam o *offset* do sinal de saída. Na filtragem, tem-se uma resistência de $15K\Omega$ (R_{x09}), um *trimpot* de $10k\Omega$ (R_{x10}), 1 capacitor de $1\mu F$ (C_{x01}) e 1 capacitor de $0,47\mu F$ (C_{x02}) por

signal, onde x varia de 1-4 sinais. Há também uma porta de entrada (*SI1*) do sinal sem amplificar, e uma porta de saída (*ASO*) dos sinais amplificados e filtrados. Na Tabela 14 é mostrado um resumo da quantidade de unidades no circuito.

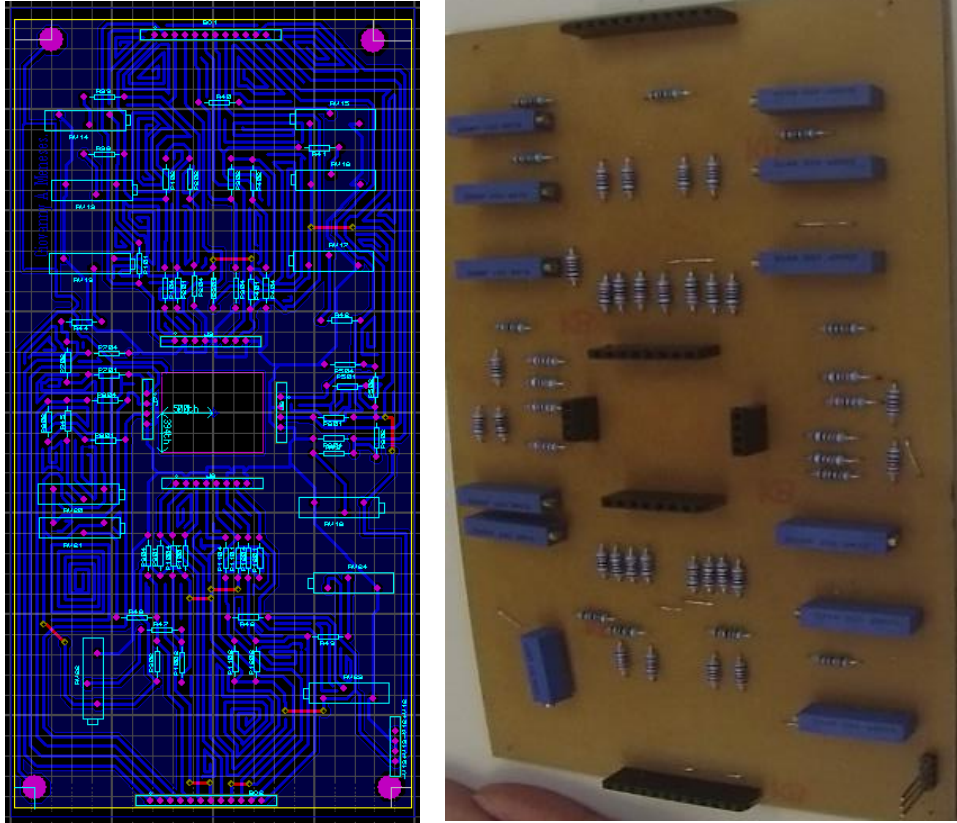


Figura 68 Primeiro circuito elétrico (primeira camada).

Tabela 13 Elementos do circuito 1

Objeto	Código	Valor	Quantidade
Resistências	$Px01, Px02, Px04$	120 Ω	36
	R38-R49	100K Ω	12
	RV13-Rv24	1K Ω	12
Conectores	J3, J8	8 pinos	2
	J6, J7	4 pinos	2
	SO1, SO2	12 pinos	2
	Alimentação	4 pinos	1

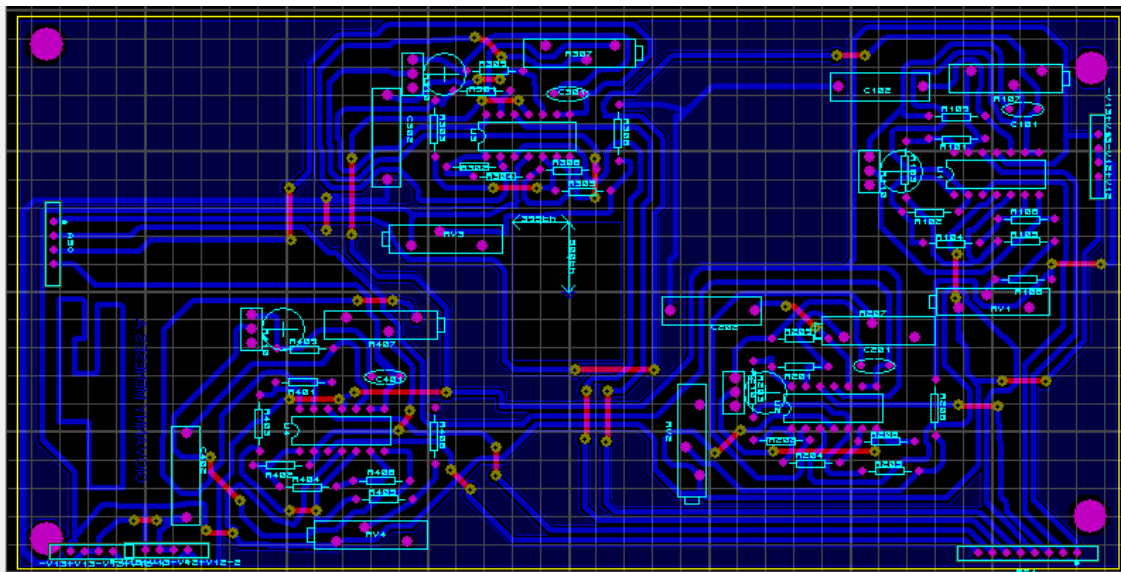


Figura 69 Segundo circuito elétrico (segunda e terceira camadas).

Tabela 14 Elementos das camadas 2 e 3

Objeto	Código	Valor	Quantidade
Resistências	<i>Rx07</i>	100Ω	4
	<i>Rx01 – Rx06</i>	100KΩ	24
	<i>Rx08</i>	1kΩ	4
	<i>RVx</i>	1KΩ	4
	<i>Rx09</i>	15KΩ	4
	<i>Rx10</i>	10kΩ	4
Capacitores	<i>Cx01</i>	1μF	4
	<i>Cx02</i>	0,47μF	4
Amplificadores TL084	U1-U4	/	4
Conectores	<i>SI1</i>	8 pinos	1
	<i>ASO</i>	4 pinos	1
	Alimentação	4 pinos	3

O terceiro circuito elétrico, apresentado na Figura 70, encontra-se na quarta camada. Ele é um híbrido entre uma parte de amplificação e filtragem e a parte de processamento e controle.

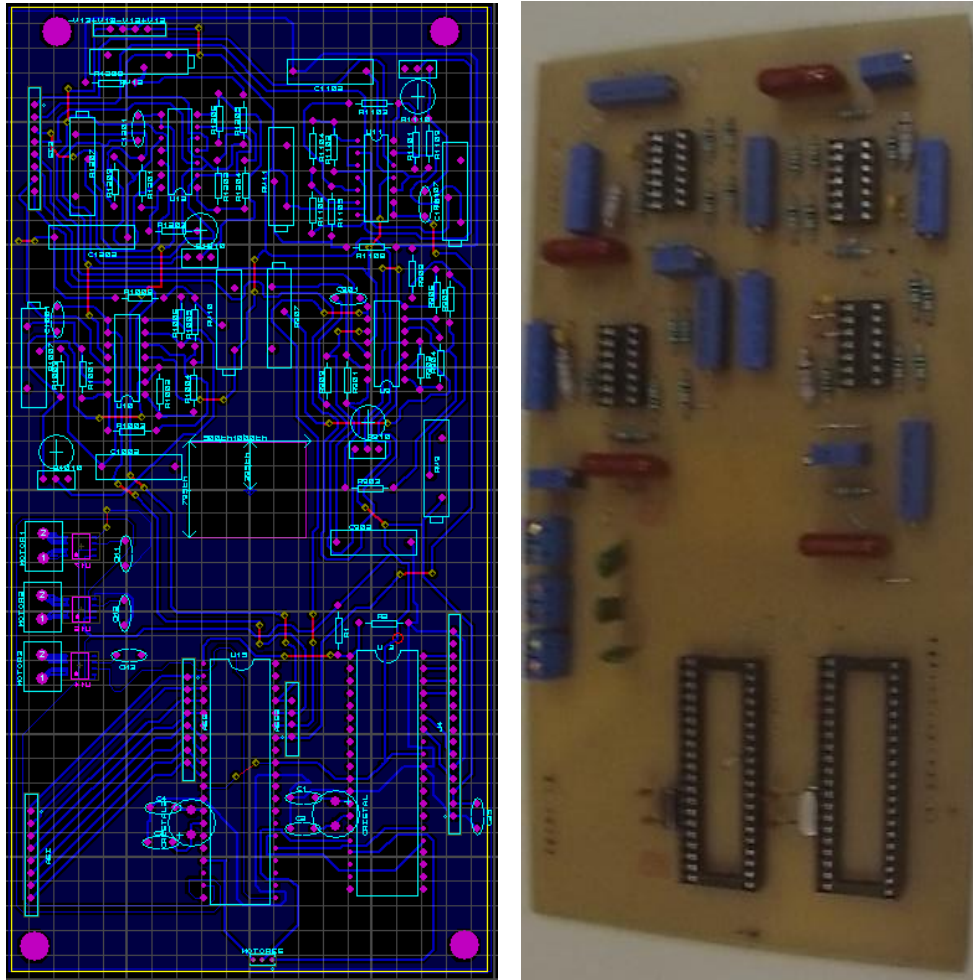


Figura 70 Terceiro circuito elétrico (quarta camada).

Neste circuito, podem-se encontrar quatro amplificadores operacionais igualmente como nos dois níveis anteriormente descritos. Encontram-se também 2 microcontroladores PIC 18f4550 (U_{13}, U_{15}), 3 sensores de efeito hall ACS712ELCTR-20A-T, 2 osciladores de cristal de 20MHz para o *clock* dos PICs, 4 capacitores de $22pF$ (C_1-C_4), 3 capacitores de $0.1\mu F$ (CMF_1-CMF_3), 3 capacitores de $1nF$ (CFM_1-CFM_3) necessários para os sensores de efeito hall (U_{14}, U_{16}, U_{17}) ACS712ELCTR-20A-T da Allegro Microsystems, e 3 conectores para os motores ($MOTOR_1-MOTOR_3$). Há também 3 capacitores de $0.1\mu F$ (CMS_1-CMS_3) para a entrada do controlador de velocidade *Sabertooth*, junto com as suas 3 resistências de $10K\Omega$ (RM_1-RM_3), os quais promovem um filtro para fazer o controle com PWM (*Pulse Width Modulation*) mais suave, de acordo com o manual do fabricante *Sabertooth* [20]. Neste circuito há 2 botões, o

primeiro (R2) é o RESET do sistema, e o segundo (R5) é uma tecla para acionar modificações na visualização no display LCD de 16x2 que está conectado no PIC U13. Tem-se um *trimpot* (TR1) para a variação de contraste no display LCD, além de uma porta de entrada dos sinais não amplificados (*SI3*) que vão para o conjunto de amplificadores. Há uma porta de entrada (*ASI*) para os sinais amplificados do circuito da camada 2 e 3, que vão para o PIC U15. Na Tabela 15 é mostrado um resumo da quantidade de unidades que se tem no circuito.

Tabela 15 Elementos da camada 4

Objeto	Código	Valor	Quantidade
Resistências	RM1-RM3	10K Ω	3
	<i>Rx07</i>	100 Ω	4
	<i>Rx01 – Rx06</i>	100K Ω	24
	<i>Rx08</i>	1k Ω	4
	<i>RVx</i>	1K Ω	4
	<i>Rx09</i>	15K Ω	4
	<i>Rx10</i>	10k Ω	4
Capacitores	C1-C4	22pF	4
	CMF1-CMF3	0.1 μ F	3
	CFM1-CFM3	1nF	3
	CMS1-CMS3	0.1 μ F	3
	<i>Cx01</i>	1 μ F	4
		0,47 μ F	4
Sensor de efeito hall ACS712 ELCTR-20A-T	U14,U16,U17	-	3
Amplificadores TL084	U9-U12	-	4
PIC18f4550	U13, U15	-	2

Botões	R2,R5	-	2
Display Lcd	-	-	1
Cristal		2	2
Conectores	SI3	8 pinos	1
	ASI	8 pinos	1
	MOTOR1-MOTOR3	2 pinos	3
	MOTORES	3 pinos	1
	J4	16 pinos	1
	Alimentação	4 pinos	1

Por não se dispor de um dispositivo com 12 entradas analógicas de pequenas dimensões que coubesse na base da bengala robótica, decidiu-se construir um dispositivo próprio. O esquema do dispositivo foi desenhado com a ferramenta Proteus-Isis. O software Proteus-Ares foi utilizado para fazer o desenho do *layout*, para assim poder construir de forma física.

O dispositivo USB para a aquisição de dados conta com um total de 3 portas, das quais duas delas (PORTO1 e PORTO2) são as entradas analógicas, com um total de 12 pinos. A terceira porta (PORTO3) é constituída de 10 pinos, onde 9 deles são entradas ou saídas digitais. Além disso, o dispositivo possui um oscilador de cristal de 20MHz, 2 capacitores cerâmicos C3 e C4 para o oscilador de cristal com valor de $22pF$, e 2 capacitores eletrolíticos para eliminar ruídos (um deles C25 na alimentação, e o outro C5 na comunicação USB). O dispositivo ainda conta com 3 diodos leds: um led (D1) de cor verde, o qual anunciará o momento da transmissão de dados; um led (D2) de cor vermelha, o qual avisará que está capturando os dados das portas analógicas, neste caso os *strain gages*; e, por último, um led (D3) de cor verde que se liga no momento em que o PIC recebe alimentação elétrica. Adicionalmente, o dispositivo conta com dois botões: um deles é o botão chamado PULSA, o qual quando acionado inicia ou parada da coleta de dados; e o outro botão, chamado RESET, reinicia o microcontrolador PIC. Um conector USB (J2) port B também faz parte do dispositivo. Na Tabela 16 é mostrado um resumo da quantidade de unidades que se tem neste circuito de entradas analógicas.

Tabela 16 Elementos da DAQ

Objeto	Código	Valor	Quantidade
Resistências	R1-R5	1K Ω	5
Capacitores	C3 e C4	22 pF	2
	C5	22 μ F	
	C25	220 μ F	
Led	D1-D3	-	3
Cristal	Cristal2	20MHz/	1
Botão	Pulsa-Reset	-	2
Conectores	PORTO1	4 pinos	1
	PORTO2	8 pinos	1
	PORTO3	10 pinos	1
	J2	-	1

A interface para coleta desses dados foi feita em *Borland Delphi for Microsoft Windows version 10.0*. Esta interface foi desenvolvida com a ajuda de rotinas do portal *Project Jedi* para desenvolver o *USB Human Interface Device (HID)*, a qual é utilizada para adquirir os dados com o circuito anterior. O seu arquivo de programação encontra-se no Apêndice B.3.

A Figura 71 apresenta a interface do programa. Nela, a aba do *software* conta com uma série de botões *TradioButton*, os quais têm por função serem selecionados dependendo da ação que se tem como objetivo. Por exemplo, em uma caminhada para frente, escolhe-se “Caminhada” como o estado e “frente” como direção; em seguida, escreve-se o nome do usuário do teste, e finalmente a condição selecionada será salva pressionando o botão “Salvar” após ter recebido os dados. O programa salvará automaticamente um arquivo *.txt* com as condições escolhidas.

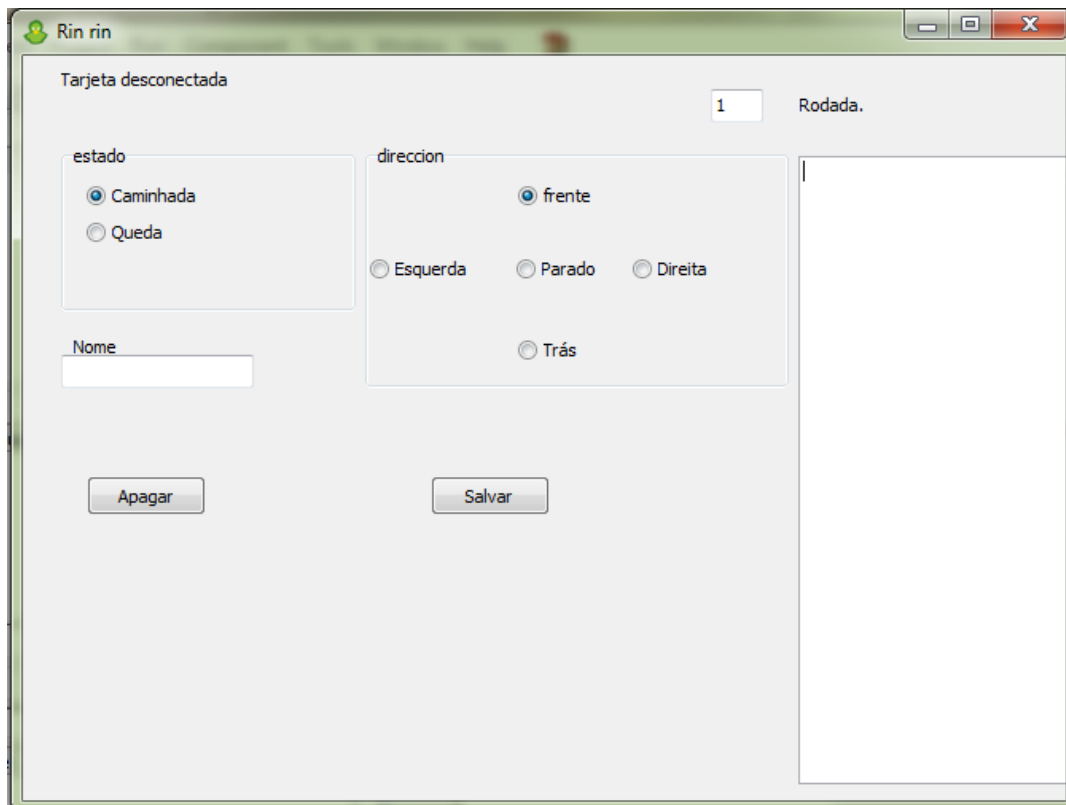


Figura 71 Interface desenvolvida para aquisição de dados.

Tabela 17 Componentes mecânicos

Componentes mecânicos	Quantidade
Motor20W (Matrix Pro Lathe 55T)	3
Caixa de redução (P60 gearbox)	3
Baterias lítio tipo 5S	2
Bengala de alumínio padrão	1
Bases de alumínio	3
Rodas omnidireccionais duplas	3

Apêndice B – Código computacional

Neste apêndice, são apresentadas as rotinas em Matlab / Simulink implementadas para a simulação do sistema.

B.1.

Modelo cinemático - script.

```
function [W1, W2, W3] = FunModeloCinematicoDir(phig, VelX,
Vely, Wplat, L, r)
%#codegen

phi=phig*pi/180;

Hinv=[-sin(phi)          cos(phi)          L;
      -sin((pi/3)-phi)  -cos((pi/3)-phi) L;
       sin((pi/3)+phi)  -cos((pi/3)+phi) L];

Qcpto=[VelX; Vely; Wplat];

z=(1/r)*(Hinv)*(Qcpto);
W1=z(1);
W2=z(2);
W3=z(3);
```

B.2.

Modelo dinâmico - script.

```
function [Torq1, Torq2, Torq3] = fcn(I, Massa, L, r, phig,
FX, FY, Fux, Fuy, TorqU, AcX, AcY, Aphi, miuA, miuM)
%#codegen

g=9.81;
N=Massa*g;
FA1=miuA*N/3;
FA2=miuA*N/3;
FA3=miuA*N/3;
FM1=miuM*N/3;
FM2=miuM*N/3;
FM3=miuM*N/3;

phi=phig*pi/180;
FA=[FA1*cos(phi)+FA2*cos((2*pi/3)+phi)+FA3*cos((4*pi/3)+phi)
;
;
```

```

FA1*sin(phi)+FA2*sin((2*pi/3)+phi)+FA3*sin((4*pi/3)+phi)];

F1d=-FM1*sin(phi) +FM2*(1/2)*(sin(pi) -sqrt(3)*cos(phi))...
    +FM3*(1/2)*(sin(pi) -sqrt(3)*cos(phi)) +FA1*cos(phi)...
    +FA2*cos((2*pi/3)+phi) +FA3*cos((4*pi/3)+phi) -Fux;

F2d=-FM1*cos(phi) -FM2*(1/2)*(cos(pi) +sqrt(3)*sin(phi))...
    -FM3*(1/2)*(-cos((pi)-sqrt(3)*sin(phi)))
+FA1*sin(phi)...
    +FA2*sin((2*pi/3)+phi) +FA3*sin((4*pi/3)+phi) -Fuy;

F3d=L*(FM1+FM2+FM3)-TorqU;

Hinv=[-sin(phi)          cos(phi)          L;
      -sin((pi/3)-phi)  -cos((pi/3)-phi)  L;
      sin((pi/3)+phi)   -cos((pi/3)+phi)  L];

H=inv(Hinv);

Fd=[F1d; F2d; F3d];
M=[Massa 0 0; 0 Massa 0; 0 0 I];

Qcpto=[AcX; AcY; Aphi];
Torq=(r*H')*(M*Qcpto+Fd);
Torq1=Torq(1);
Torq2=Torq(2);
Torq3=Torq(3);

```

B.3. Interface da aquisição.

```

unit osCHID;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, CPort, xpman, ComCtrls, ExtCtrls,
  JvHidControllerClass{para apariciencia bonita};

const
  MiVID = $04d8;
  MiPID = $1234;

type
  TForm1 = class(TForm)
    Memo1: TMemo;
    JvHidDeviceController1: TJvHidDeviceController;
    Label2: TLabel;
    RichEdit1: TRichEdit;
    SaveDialog1: TSaveDialog;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;

```

```

GroupBox1: TGroupBox;
caida: TRadioButton;
camina: TRadioButton;
GroupBox2: TGroupBox;
frente: TRadioButton;
atras: TRadioButton;
izquierda: TRadioButton;
derecha: TRadioButton;
Label1: TLabel;
Edit2: TEdit;
quieto: TRadioButton;
Label3: TLabel;
Label4: TLabel;
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure FormCreate(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure JvHidDeviceController1DeviceChange(Sender: TObject);
function JvHidDeviceController1Enumerate(HidDev: TJvHidDevice;
  const Idx: Integer): Boolean;
procedure JvHidDeviceController1Removal(HidDev: TJvHidDevice);
procedure JvHidDeviceController1DeviceData(HidDev: TJvHidDevice;
  ReportID: Byte; const Data: Pointer; Size: Word);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
  procedure Lectura(HidDev: TJvHidDevice; ReportID: Byte;
    const Data: Pointer; Size: Word);
end;

```

```

var
  Form1: TForm1;
  Txtenviar:string;
  Txtrecibido:string;
  letra:Char;
  a:double;
  a1:double; // Para las dimensiones de la imagen
  a2:double;
  b1:double;
  b2:double;
  c:Integer; //Para ir corriendo en el tiempo
  numero:Integer;
  Conectada : Boolean;
  Midispositivo : TJvHidDevice;
  envia : array[0..2] of string;
  Cenvia : cardinal;
  texto:string;
  conta: Integer;
  ct:integer;
  estado: string;
  direccion:string;
  cd:integer;

```

implementation

```
{ $R *.dfm }
```

```
procedure TForm1.Lectura(HidDev: TJvHidDevice; ReportID: Byte; const
Data: Pointer; Size: Word);
```

```
var
```

```
  I: Integer;
```

```
  Str: string;
```

```
  Recibe: array [0..10] of int64;
```

```
begin
```

```
  Str := ('RPM : ');
```

```
  for I := 0 to Size - 1 do
```

```
    begin
```

```
      Recibe[I] := Cardinal(PChar(Data)[I]);
```

```
      label1.Caption:=inttostr(Recibe[I]);
```

```
      Str := Format('%0.2d', [Cardinal(PChar(Data)[I])]);
```

```
    end;
```

```
    texto:=Str;
```

```
    cd:=cd+1;
```

```
    Memo1.Text:=memo1.Text+texto+',';
```

```
    conta:=conta+1;
```

```
    richedit1.Text:=richedit1.Text+texto+chr(9);
```

```
    label1.Caption:=inttostr(cd);
```

```
    if(conta=12) then
```

```
      begin
```

```
        memo1.Lines.add("");
```

```
        richedit1.Lines.Add("");
```

```
        conta:=0;
```

```
      end;
```

```
{ if CheckBox1.Checked then
```

```
  Begin
```

```
    if c>400 then
```

```
      begin
```

```
        c:=0;
```

```
        Image1.Picture:=nil;
```

```
      end;
```

```
    Image1.Canvas.MoveTo(c-1,numero);
```

```
    a:=strtofloat(texto);
```

```
    numero:=Round(((b2-b1)/(a2-a1))*(a-a1)+b1);
```

```
    label6.Caption:=inttostr(conta);
```

```
{ if conta=1 then
```

```
  begin
```

```
    label4.Caption:=Str;
```

```
    conta:=0;
```

```
  end;
```

```
  label6.Caption:=inttostr(conta);
```

```

    if conta=0 then
    begin
    label5.Caption:=Str;
    conta:=1;
    end; }

{   if c>0 then
    begin
    Image1.Canvas.LineTo(c,numero);
    Memo1.Text:=currtoStr(floattocurr((a/255)*5));

    end;
    c:=c+1; }

end;

{
procedure TMainForm.ShowRead(HidDev: TJsHidDevice; ReportID: Byte;
  const Data: Pointer; Size: Word);
var
  I: Integer;
  Str: string;
begin
  Str := Format('RD %.2x ', [ReportID]);
  for I := 0 to Size - 1 do
    Str := Str + Format('%.2x ', [Cardinal(PChar(Data)[I])]);
  AddToHistory(Str);
end;
}

procedure TForm1.Button1Click(Sender: TObject);
begin
  ForceCurrentDirectory := True;
  ct:=strtoint(edit2.Text);
  if(strtoint(booltostr(caida.Checked))=-1) then
  begin
    estado:='caida';
  end
  else
  begin
    estado:='camina';
  end;
  if(strtoint(booltostr(frente.Checked))=-1) then
  begin
    direccion:='frente';
  end;
  if(strtoint(booltostr(atras.Checked))=-1) then
  begin
    direccion:='atras';
  end;
  if(strtoint(booltostr(izquierda.Checked))=-1) then
  begin
    direccion:='izquierda';
  end;
  if(strtoint(booltostr(derecha.Checked))=-1) then

```

```

begin
  direccion:='derecha';
end;

if(strtoint(booltostr(quieto.Checked))=-1) then
begin
  direccion:='quieto';
end;
richedit1.Lines.SaveToFile(edit2.Text+' - '+edit1.Text+' '+kilos+' '+estado
+' '+direccion+'.txt');
ct:=ct+1;
edit2.Text:=inttostr(ct);

end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  richedit1.Text:="";
  memo1.Text:="";
  conta:=0;
  cd:=0;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  numero:=byte('@');
end;

{ Txtenviar:=Edit2.Text+' dice: '+Edit1.Text;
  Comport1.WriteStr(Txtenviar);
  Memo1.Font.Color:=clLime;
  Memo1.Lines.Add(Txtenviar);      {Para enter chr(13)+chr(10)}
  {RichEdit1.SelAttributes.Color:=clLime;
  RichEdit1.Lines.Add(Txtenviar);
  Edit1.Text:="";}

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
begin
  {Cuenta cuantos caracteres llegan. Es un hilo porque lo hace cuando
ocurre}

end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  c:=0;
  //Image1.Canvas.MoveTo(10,105);
  {cuando en ascii vale 0 es el pixel y=200 => Vale 255 es y=10
  0*a+b=200 y 255*a+b=10 a=-190/255}
  a:=-190/255;
  a1:= 0;
  a2:=255;
  b1:=190;
  b2:=10;
  Conectada := false;
  Label2.Caption := 'Tarjeta desconectada';

```

```

        conta:=0;

    end;

    procedure          TForm1.JvHidDeviceController1DeviceChange(Sender:
TObject);
    begin
        JvHidDeviceController1.Enumerate;
    end;

    procedure          TForm1.JvHidDeviceController1DeviceData(HidDev:
TJvHidDevice;
        ReportID: Byte; const Data: Pointer; Size: Word);
    begin
        conta:=conta+1;

    end;

    function           TForm1.JvHidDeviceController1Enumerate(HidDev:
TJvHidDevice;
        const Idx: Integer): Boolean;
    begin
        if (JvHidDeviceController1.CheckOutByID(HidDev,MiVID,MiPID)= true)
then
        begin
            Label2.Caption :='Tarjeta conectada';
            Conectada := true;
            Midispositivo := HidDev;
            Midispositivo.OnData := Lectura;
        end;
    end;

    procedure          TForm1.JvHidDeviceController1Removal(HidDev:
TJvHidDevice);
    begin
        if (Hiddev.Attributes.VendorID = MiVID)and (Hiddev.Attributes.ProductID =
MiPID) then
        begin
            label2.Caption := 'Tarjeta desconectada';
            Conectada := false;
            JvHidDeviceController1.CheckIn(HidDev);
            Midispositivo := nil;
        end;
    end;
end.

```

B.4. Treinamento da rede - função principal.

```

clear all
clc
[caida, caminadas]=importardatos(40);

```



```

frec=20;
%% pega o primeiro dado e zera como referência para os
demais
[caida]=zera(caida);
[camina]=zera(caminadas);
%%
%%faz uma estrutura com os dados, só os dados
%do evento da queda.
[cc]=areglacaida(caida);
[cm]=areglacamina(caminadas);

%%
% cria declive
% [valorpend]=pend(cc,frec);
% [valorpendcm]=pend(cm,frec);

%%
% cria média
[valorpend]=promedio(cc,frec);
[valorpendcm]=promedio(cm,frec);
%%
% cria o target da rede.
[target_caida, target_camina]=target(valorpend,valorpendcm);

%%
% datos de entrada, % treinamento, %teste, %validacion
[tre_caida,
test_caida,val_caida]=organiza(target_caida,0.8,0.1,0.1);
[tre_camina,
test_camina,val_camina]=organizacamina(target_camina,0.8,0.
1,0.1);

dtreinamento=[tre_caida(:,1:10);tre_camina(:,1:10)];
dteste=[test_caida(:,1:10);test_camina(:,1:10)];
dval=[val_caida(:,1:10); val_camina(:,1:10)];

target_tr_caida=tre_caida(:,11);
target_t_caida=test_caida(:,11);
target_v_caida=val_caida(:,11);

target_tr_camina=tre_camina(:,11);
target_t_camina=test_camina(:,11);
target_v_camina=val_camina(:,11);

%converte a matriz em um vetor, normaliza ele e retorna
matriz original normalizada.
[ndtreinamento]=nvectoriza(dtreinamento,0,1);
[ndtest]=nvectoriza(dteste,0,1);
[ndval]=nvectoriza(dval,0,1);
[sndtreinamento]=vectoriza(dtreinamento,0,1);

padroes_tr_pu=ndtreinamento';
padroes_t_pu=ndtest';
padroes_v_pu=ndval';
target_tr=codifica([target_tr_caida; target_tr_camina])';
target_t=codifica([target_t_caida; target_t_camina])';

```

```

target_v=codifica([target_v_caida; target_v_camina]');

for i=1:50; % número de redes a testar

P=[padroes_tr_pu padroes_v_pu padroes_t_pu];
T=[target_tr target_v target_t];

net=newff(P,T,[10 10 10]);
net.trainParam.epochs = 1000;
net.trainParam.max_fail = 50;
net.initFcn='initlay';

net.divideFcn='divideind';
net.divideParam.trainInd=1:size(padroes_tr_pu,2);
net.divideParam.valInd=size(padroes_tr_pu,2)+1:size(padroes_
tr_pu,2)+size(padroes_v_pu,2);
net.divideParam.testInd=size(padroes_tr_pu,2)+size(padroes_v
_pu,2)+1:size(padroes_tr_pu,2)+size(padroes_v_pu,2)+size(pad
roes_t_pu,2);

net.trainParam.showWindow=1;
net_T=trainlm(net,P,T);

outputs=sim(net_T,padroes_v_pu);
perf(i) = mse(net_T,target_v,outputs);

pesos(:,i)=getwb(net_T);
i
end

%%
menor=min(perf);
[x]=find(perf==menor);
net_T=setwb(net_T,pesos(:,x));
outputs=sim(net_T,padroes_v_pu);
mse(net_T,target_v,outputs)
figure
plotconfusion(target_v,outputs)

```

B.4.1. Treinamento rede funções secundárias.

B.4.1.1. importardados

```

function [caidas caminas]=importardatos(i)

carpeta='caminada';
estado='camina';
persona='Giovanny';
direccion='frente';

for n = 1:i

```

```

    if (n==11)
        direcion='atras';
    end

    if (n==21)
        direcion='derecha';
    end

    if (n==31)
        direcion='izquierda';
    end

    camina(:, :, n) =
import (carpeta, int2str(n), estado, persona, direcion);
    caminas(:, :, n)=[camina(:, 1:5, n) camina(:, 7, n)
camina(:, 9:12, n)];
end

carpeta='caida';
estado='caida';
persona='Giovanny';
direcion='atras';

for n = 1:i
    if (n==11)
        direcion='frente';
    end

    if (n==21)
        direcion='izquierda';
    end

    if (n==31)
        direcion='derecha';
    end

    caida(:, :, n) =
import (carpeta, int2str(n), estado, persona, direcion);
    caidas(:, :, n)=[caida(:, 1:5, n) caida(:, 7, n)
caida(:, 9:12, n)];
end

```

B.4.1.2. Zera

```

function [out]=zera(in)

out=in;
for j=1:size(in, 3)
    for i=1:size(in, 2)
        out(:, i, j)=in(:, i, j)-in(1, i, j);
    end
end
end

```

B.4.1.3. Ajustar o caminhar

```
function [caminare]=areglacamina(camina)
    for l=1:40

caminare.(genvarname(strcat('caida',num2str(l))))=camina(:, :
,l);
    end
```

B.4.1.4. Ajustar a queda

```
function [caidasare]=areglacaida(caida)
for l=1:40
    caidasare.(genvarname(strcat('caida',num2str(l))))=[];
end
```

```
caidasare.caida1=caida(19:25, :, 1);
caidasare.caida2=caida(25:36, :, 2);
caidasare.caida3=caida(22:36, :, 3);
caidasare.caida4=caida(19:29, :, 4);
caidasare.caida5=caida(14:25, :, 5);
caidasare.caida6=caida(17:26, :, 6);
caidasare.caida7=caida(20:32, :, 7);
caidasare.caida8=caida(8:23, :, 8);
caidasare.caida9=caida(15:29, :, 9);
caidasare.caida10=caida(15:28, :, 10);
caidasare.caida11=caida(21:26, :, 11);
caidasare.caida12=caida(21:30, :, 12);
caidasare.caida13=caida(11:20, :, 13);
caidasare.caida14=caida(25:34, :, 14);
caidasare.caida15=caida(15:22, :, 15);
caidasare.caida16=caida(21:25, :, 16);
caidasare.caida17=caida(9:18, :, 17);
caidasare.caida18=caida(12:20, :, 18);
caidasare.caida19=caida(14:22, :, 19);
caidasare.caida20=caida(12:18, :, 20);
caidasare.caida21=caida(20:25, :, 21);
caidasare.caida22=caida(21:25, :, 22);
caidasare.caida23=caida(11:18, :, 23);
caidasare.caida24=caida(11:16, :, 24);
caidasare.caida25=caida(10:17, :, 25);
caidasare.caida26=caida(18:24, :, 26);
caidasare.caida27=caida(31:39, :, 27);
caidasare.caida28=caida(17:27, :, 28);
caidasare.caida29=caida(28:30, :, 29);
caidasare.caida30=caida(25:27, :, 30);
caidasare.caida31=caida(16:22, :, 31);
caidasare.caida32=caida(23:27, :, 32);
caidasare.caida33=caida(19:23, :, 33);
caidasare.caida34=caida(33:37, :, 34);
caidasare.caida35=caida(34:37, :, 35);
```

```
caidasare.caida36=caida(30:34, :, 36);
caidasare.caida37=caida(21:24, :, 37);
caidasare.caida38=caida(15:18, :, 38);
caidasare.caida39=caida(19:27, :, 39);
caidasare.caida40=caida(23:28, :, 40);
```

B.4.1.5. Pend

```
function [valorpend]=pend(valor,frec)
%%
for l=1:40;
    for
j=1:size(valor.(genvarname(strcat('caida',num2str(l)))),1)-
1;

valorpend.(genvarname(strcat('caida',num2str(l))))(j,:)=(val
or.(genvarname(strcat('caida',num2str(l))))(j+1,:)-
valor.(genvarname(strcat('caida',num2str(l))))(j,:))*frec;
        end
    end
end
```

B.4.1.6. Média

```
function [valorpend]=promedio(valor,frec)
%%
for l=1:40;
    for
j=1:size(valor.(genvarname(strcat('caida',num2str(l))))(j,:)-
1;

valorpend.(genvarname(strcat('caida',num2str(l))))(j,:)=(val
or.(genvarname(strcat('caida',num2str(l))))(j+1,:)+valor.(ge
nvarname(strcat('caida',num2str(l))))(j,:))/2;
        end
    end
end
```

B.4.1.7. Alvo

```
function [out1 out2]=target(in1, in2)
% crea targets en las estructura principal.
% in1 caida          in2 caminada
out1=[];
out2=[];
x=1;
    for j=1:40
        if (j==11)
            x=2;
```

```

        end
        if (j==21)
            x=3;
        end
        if (j==31)
            x=4;
        end
    end

in1.(genvarname(strcat('caida',num2str(j))))(:,11)=x;
end

    x=2;
    for j=1:40
        if (j==11)
            x=1;
        end
        if (j==21)
            x=4;
        end
        if (j==31)
            x=3;
        end
    end
in2.(genvarname(strcat('caida',num2str(j))))(:,11)=x;
end
out1=in1;
out2=in2;

```

B.4.1.7. Organiza1

```

function [tre, test,val]=organiza1(dadoin,ptre,pptest,pval)
% dado in=dados a organizar,
% ptre ptest pval porcentagem de entrada do treinamento
% teste y validacao.

out1=[];
out2=[];
out3=[];
out4=[];

    for j=1:10
        out1=[out1;
dadoin.(genvarname(strcat('caida',num2str(j))))];
    end

    for j=11:20
        out2=[out2;
dadoin.(genvarname(strcat('caida',num2str(j))))];
    end

    for j=21:30
        out3=[out3;
dadoin.(genvarname(strcat('caida',num2str(j))))];
    end
end

```

```

    for j=31:40
        out4=[out4;
dadoin.(genvarname(strcat('caida',num2str(j))))];
    end

tamanho=size(out1,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre1=out1(1:t1,:);
test1=out1(t1+1:t1+t2,:);
val1=out1(t1+t2:t1+t2+t3,:);

tamanho=size(out2,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre2=out2(1:t1,:);
test2=out2(t1+1:t1+t2,:);
val2=out2(t1+t2:t1+t2+t3,:);

tamanho=size(out3,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre3=out3(1:t1,:);
test3=out3(t1+1:t1+t2,:);
val3=out3(t1+t2:t1+t2+t3,:);

tamanho=size(out4,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre4=out4(1:t1,:);
test4=out4(t1+1:t1+t2,:);
val4=out4(t1+t2:t1+t2+t3,:);

% saidas
% grupos de treinamento, teste e validação.
tre=[tre1;tre2;tre3;tre4];
test=[test1;test2;test3;test4];
val=[val1;val2;val3;val4];

```

B.4.1.8. Organiza1camina

```

function [tre,
test,val]=organiza1camina(dadoin,ptre,ptest,pval)
% datos a organizar target intrada,
% ptre ptest pval porcentaje de entrada de treinamento
% teste y validação.
% dadoin=target_caida;
out1=[];

```

```

out2=[];
out3=[];
out4=[];

    for j=1:10
        out1=[out1;
dadoin.(genvarname(strcat('caida',num2str(j))))(1:10,:)]];
    end

    for j=11:20
        out2=[out2;
dadoin.(genvarname(strcat('caida',num2str(j))))(1:10,:)]];
    end

    for j=21:30
        out3=[out3;
dadoin.(genvarname(strcat('caida',num2str(j))))(1:10,:)]];
    end

    for j=31:40
        out4=[out4;
dadoin.(genvarname(strcat('caida',num2str(j))))(1:10,:)]];
    end

tamanho=size(out1,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre1=out1(1:t1,:);
test1=out1(t1+1:t1+t2,:);
val1=out1(t1+t2:t1+t2+t3,:);

tamanho=size(out2,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre2=out2(1:t1,:);
test2=out2(t1+1:t1+t2,:);
val2=out2(t1+t2:t1+t2+t3,:);

tamanho=size(out3,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre3=out3(1:t1,:);
test3=out3(t1+1:t1+t2,:);
val3=out3(t1+t2:t1+t2+t3,:);

tamanho=size(out4,1);
t1=tamanho*ptre;
t2=tamanho*ptest;
t3=tamanho*pval;
tre4=out4(1:t1,:);
test4=out4(t1+1:t1+t2,:);
val4=out4(t1+t2:t1+t2+t3,:);

```



```
tre=[tre1;tre2;tre3;tre4];  
test=[test1;test2;test3;test4];  
val=[val1;val2;val3;val4];
```

B.4.1.9. **Nvectoriza**

```
function [out]=nvectoriza(in,min,max)  
out=[];  
m=size(in,1);  
vector=[];  
for i=1:m  
    vector=[vector in(i,:)];  
end  
  
nvect=mapminmax(vector,min,max);  
  
out=nvect(1:10);  
for i=1:m-1  
    out=[out;nvect(1,i*10+1:i*10+10)];  
end
```