



INSTITUTO TECNOLÓGICO VALE



Programa de Pós Graduação em Instrumentação, Controle e Automação de  
Processos de Mineração - PROFICAM  
Universidade Federal de Ouro Preto - Escola de Minas  
Associação Instituto Tecnológico Vale - ITV

Dissertação

ANÁLISE DE MOBILIDADE DE UM DISPOSITIVO ROBÓTICO PARA  
INSPEÇÃO REMOTA DE CAVIDADES

Filipe Augusto Santos Rocha

Ouro Preto  
Fevereiro de 2018



**Filipe Augusto Santos Rocha**

**ANÁLISE DE MOBILIDADE DE UM DISPOSITIVO ROBÓTICO PARA  
INSPEÇÃO REMOTA DE CAVIDADES**

Dissertação apresentada ao curso de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Linha de Pesquisa: Robótica

Orientador: Prof. D.Sc. Gustavo Medeiros Freitas

Coorientador: Prof. D.Sc. Paulo Henrique Vieira Magalhães

Ouro Preto, MG – Brasil  
Fevereiro de 2018

R672d Rocha, Filipe Augusto Santos.  
Análise de mobilidade de um dispositivo robótico para  
Inspeção remota de cavidades [manuscrito] / Filipe Augusto Santos Rocha. –  
2018.  
118f. : il., color., tab.

Orientador: Prof. Dr. Gustavo Medeiros Freitas.  
Co-orientador: Dr. Paulo Henrique Vieira Magalhães.

Dissertação (Mestrado) – Universidade Federal de Ouro  
Preto/ Instituto Tecnológico Vale. Escola de Minas. Programa de Pós Graduação em  
Instrumentação, Controle e Automação de Processos de Mineração.  
Área de concentração: Engenharia de Controle e Automação de Processos  
Minerais.

1. Automação industrial. 2. Microcontroladores. 3. Minas e recursos minerais.  
4. Robos - Sistemas de controle. I. Universidade Federal de Ouro Preto. II. Título.

CDU: 681.5:622.2

Fonte de catalogação: [bibem@sisbin.ufop.br](mailto:bibem@sisbin.ufop.br)

Mestrado Profissional em Instrumentação, Controle e Automação de Processos  
de Mineração - PROFICAM

Análise de Mobilidade de um Dispositivo Robótico para Inspeção Remota  
de Cavidades

Filipe Augusto Santos Rocha

Dissertação defendida e aprovada em 19 de fevereiro de 2018 pela banca  
examinadora constituída pelos professores:



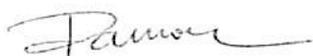
---

Ph.D. Gustavo Medeiros Freitas  
Orientador – Instituto Tecnológico Vale (ITV)



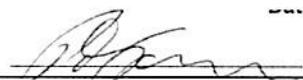
---

D.Sc. Paulo Henrique Vieira Magalhães  
Coorientador – Universidade Federal de Ouro Preto (UFOP)



---

D.Sc. Ramon Romankevicius Costa  
Membro externo – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia  
(COPEE)



---

Ph.D. Pål Johan From  
Membro externo – Norwegian University of Life Sciences

*Dedico este singelo trabalho a  
quem me dedicou toda uma vida  
com todo o coração: te amo  
muito, mamãe.*

# Agradecimentos

A Deus pelo ontem, hoje, e pelo amanhã.

Aquele que crê não necessitar da ajuda de outras pessoas demonstra, além de arrogância, o quão turva e restrita está sua visão sobre o mundo. A colaboração mútua se inicia antes mesmo de nascermos e perdura até nosso último sopro de vida. Segue então meus agradecimentos, singelos e de todo o coração, a todos que impactaram de alguma forma em minha vida.

Ao meu pai José e à minha mãe Maristela, por terem construído a fundação mais forte de onde posso edificar os meus sonhos.

Ao meu irmão Daniel, por nunca me dar o que quero, mas sim o que sempre preciso.

À minha segunda mãe Didi, por sempre estar ao nosso lado, garantindo nossa alegria e estômago cheio.

A toda a família, por sempre se mostrar presente na hora de compartilhar um sorriso ou uma lágrima, principalmente aos “primãos”: Allisson, Victor e Thulio.

À pequena gigante do meu coração, Renatinha, por me mostrar que, às vezes, basta apenas pintar os lados do cubo.

A todos os professores e guias acadêmico que ao longo da jornada dedicaram um minuto que fosse para ajudar no meu crescimento, em especial: Cocota, Gustavo, Adilson, Paulo Monteiro, Thiago Euzébio, Joca, Alan Kardek, Ronilson e Paulo Magalhães.

Aos grandes amigos do Laboratório de Controle e Robótica do ITV, que sempre me deram um apoio e um lanchinho: Marcão, Dudu, Leite, Hector, Ênio, Jhony, Cris, Luiz, André, Ângelo, Victória, Torga, Xandão e Zanini.

Aos companheiros do Instituto Tecnológico Vale pela confiança e pelas oportunidades de crescimento pessoal e profissional a mim proporcionados.

A toda a primeira turma do PROFICAM, pela rápida afinidade e forte companheirismo para enfrentar os inéditos desafios que nos sugeriram.

Por fim, à inoxidável República Tonteria e a todos os irmãos para a vida que aí forjei.

*“Se eu vi mais longe, foi por estar  
sobre ombros de gigantes.”*  
(Sir Isaac Newton).

Resumo da Dissertação apresentada à Escola de Minas/UFOP e ao ITV como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## ANÁLISE DE MOBILIDADE DE UM DISPOSITIVO ROBÓTICO PARA INSPEÇÃO REMOTA DE CAVIDADES

Filipe Augusto Santos Rocha

Fevereiro/2018

Orientadores: Gustavo Medeiros Freitas  
Paulo Henrique Vieira Magalhães

Esta dissertação aborda o estudo da mobilidade de dispositivos robóticos que se locomovem sobre terrenos não estruturados. O robô investigado aqui é projetado para a inspeção de cavidades naturais. Este pode utilizar diferentes modos de locomoção, como rodas, rodas tipo estrela, pernas e até configurações híbridas. A comutação entre os modos é possibilitada por um sistema de troca rápida dos mecanismos de locomoção. Métricas são usadas para quantificar o consumo energético, estabilidade, capacidade de transpor obstáculos e velocidade de translação. Um ambiente de simulação é desenvolvido integrando o simulador V-REP, *Robot Operating System* (ROS) e bibliotecas em Python. É possível comandar os elementos virtuais através do ROS e obter as métricas de mobilidade através das bibliotecas. Diferentes ensaios são propostos para a avaliação do desempenho dos modos. Em especial, um dos ensaios é realizado em ambiente representativo baseado no mapa tridimensional de uma caverna real. Informações são obtidas através do cálculo das métricas propostas. Testes de campo utilizando um protótipo do robô mostram o comportamento do mesmo em diversas situações. Os resultados e comparações obtidos permitem avaliar as características de cada modo de locomoção e inferir as vantagens e desvantagens do uso de cada um. Os códigos implementados são disponibilizados em anexo e em repositórios na internet.

**Palavras-chave:** Espeleologia, Métricas de Mobilidade, Análise de Mobilidade

Abstract of Dissertation presented to Escola de Minas/UFOP and ITV as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## MOBILITY ANALYSIS OF A ROBOTIC DEVICE FOR REMOTE CAVES INSPECTION

Filipe Augusto Santos Rocha

February/2018

Advisors: Gustavo Medeiros Freitas  
Paulo Henrique Vieira Magalhães

This dissertation treats the mobility of robotic devices that move over unstructured terrains. The robot studied here is designed for inspection of natural cavities. A mode of locomotion quick-change system allows the robot to use wheels, star wheels, legs and even hybrid configurations. Metrics enable quantifying and comparing energy consumption, stability, ability to overcome obstacles and speed of translation among the modes of locomotion. A simulation environment is developed by integrating the V-REP simulator, Robot Operating System (ROS) and Python libraries. It is possible to command the virtual elements using ROS nodes and obtain the mobility metrics through the libraries. Different tests are proposed for the evaluation of the performance of the modes. In particular, one of the analyses is performed in a representative environment based on the three-dimensional map of an real cave. Information is obtained by calculating the proposed metrics. Field tests using a prototype of the robot show the behavior of the robot in several situations. The results and comparisons obtained allow to evaluate the characteristics of each mode of locomotion and to infer the advantages and disadvantages of the use of each one. The implemented codes are available in an attachment and in repositories on the Internet.

**Keywords:** Speleology, Mobility Metrics, Mobility Analysis

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	O EspeleoRobô . . . . .	3
1.2	A robótica móvel na exploração ambiental . . . . .	5
1.3	Desafios da robótica em ambientes confinados . . . . .	7
1.3.1	Comunicação . . . . .	8
1.3.2	Mobilidade . . . . .	8
1.4	Objetivos . . . . .	9
1.5	Organização da Dissertação . . . . .	10
<b>2</b>	<b>MELHORIAS E ADAPTAÇÕES NO ESPELEOROBÔ</b>	<b>11</b>
2.1	Mecânica . . . . .	11
2.1.1	Sistema de troca rápida dos modos de locomoção . . . . .	14
2.2	Eletrônica . . . . .	15
2.3	<i>Software</i> . . . . .	16
2.3.1	Algoritmos de comando para locomoção . . . . .	17
2.3.2	Interface de operação . . . . .	18
2.4	Torre de instrumentação . . . . .	19
<b>3</b>	<b>MÉTRICAS DE MOBILIDADE</b>	<b>21</b>
3.1	Consumo energético . . . . .	21
3.2	Estabilidade . . . . .	22
3.2.1	Definições . . . . .	23
3.2.2	Margem de estabilidade estática . . . . .	24
3.2.3	Ângulo de tombamento . . . . .	25
3.2.4	Margem de estabilidade energética . . . . .	28
<b>4</b>	<b>CARACTERÍSTICAS DOS MECANISMOS DE LOCOMOÇÃO</b>	<b>29</b>
4.1	Robôs com pernas: hexápodes . . . . .	29
4.1.1	Marcha tripé com <i>Buehler's Clock</i> . . . . .	30
4.2	Robôs com rodas . . . . .	34
4.2.1	Cinemática de dispositivos <i>Skid-steering</i> . . . . .	35

4.3	Conclusões . . . . .	39
<b>5</b>	<b>SIMULAÇÕES</b>	<b>41</b>
5.1	Obtenção de parâmetros em simulação . . . . .	42
5.1.1	Métricas de consumo energético . . . . .	43
5.1.2	Métricas de estabilidade . . . . .	44
5.2	Ensaio realizados . . . . .	45
5.2.1	Estabilidade . . . . .	46
5.2.2	Máxima altura de transposição . . . . .	49
5.2.3	Locomoção em ambiente controlado . . . . .	51
5.2.4	Locomoção em ambiente representativo . . . . .	53
5.3	Discussões . . . . .	57
5.3.1	Comparação genérica dos modos . . . . .	57
5.3.2	Comparação dos modos em ambiente representativo . . . . .	60
<b>6</b>	<b>TESTES DE CAMPO</b>	<b>63</b>
6.1	Testes iniciais . . . . .	64
6.2	Teste em mina abandonada . . . . .	65
6.3	Teste em duto de dreno . . . . .	66
6.4	Testes em galerias de vertedouro de barragens . . . . .	66
6.5	Teste com a torre de mapeamento acoplada . . . . .	69
<b>7</b>	<b>CONCLUSÕES GERAIS E TRABALHOS FUTUROS</b>	<b>70</b>
7.1	Contribuições da dissertação . . . . .	71
7.1.1	Análise e comparação dos modos de locomoção . . . . .	71
7.1.2	Ambiente de simulação e algoritmos desenvolvidos . . . . .	72
7.1.3	Publicações . . . . .	72
7.2	Sugestões de trabalhos futuros . . . . .	73
	<b>Referências Bibliográficas</b>	<b>74</b>
<b>A</b>	<b>Códigos implementados</b>	<b>79</b>

# Lista de Figuras

1.1	Localização de cavidades naturais e zonas de proteção na área da mina de S11D, município de Canaã dos Carajás-PA (Arquivo da Vale). . . . .	2
1.2	Demonstração dos diversos riscos atrelados à inspeção de cavernas quando realizadas por pessoas (Arquivo público da internet). . . . .	3
1.3	Comparação entre as morfologias do RHex (à esquerda) e o EspeleoRobô (à direita). . . . .	3
1.4	Mapa tridimensional gerado a partir da fusão de dados de sensores referente a uma cavidade localizada em área de exploração ferrífera, Carajás-PA. . .	4
1.5	Modelo intermediário do Robô Ambiental Híbrido (SANTOS <i>et al.</i> ,2009). . .	5
1.6	Robô <i>Groundhog</i> , da Carnegie Mellon University (MORRIS,2005). . . . .	6
1.7	Modelos dos robôs: FRESE I - Robô com articulação no meio-corpo (a); FRESE II - Robô completamente articulado (b). Ambos os robôs utilizam o mesmo tipo de mecanismo de locomoção. (SILES e WALKER,2009) . . .	7
1.8	Robô CTER configurado para entrar no ambiente confinado através de um pequeno orifício (à esquerda) e para efetuar a inspeção do mesmo (à direita) (LARSON <i>et al.</i> ,2014). . . . .	8
1.9	Mostra de diferentes tipos de terrenos presentes em cavidades naturais, sendo trechos planos (à esquerda) e acidentados (à direita) (Arquivo público da internet). . . . .	9
2.1	Diagrama mecânico de montagem do EspeleoRobô. A descrição dos componentes enumerados pode ser consultada na Tabela 2.1. . . . .	12
2.2	Demonstração do funcionamento do sistema de engate rápido (COTA <i>et al.</i> ,2017). . . . .	14
2.3	Possíveis modos de locomoção do EspeleoRobô: hexápode original, apenas com rodas e híbrida, respectivamente . . . . .	15
2.4	Topologia dos periféricos instalados no EspeleoRobô (COTA <i>et al.</i> ,2017). . .	16
2.5	Diagrama hierárquico dos códigos de locomoção utilizados. Em vermelho encontram-se os tópicos, em verde os nós e, em azul, as bibliotecas implementadas. . . . .	17
2.6	Interface de comando do operador. . . . .	19

2.7	Modelo da torre de instrumentação que será acoplada ao EspeleoRobô. . . . .	20
3.1	Dispositivo móvel sobre um plano inclinado. . . . .	23
3.2	Demonstração do Polígono de Suporte para um dispositivo móvel qualquer. Em ambas as situações, todos os mecanismos de locomoção robô tocam o chão, configurando assim pontos de apoio. Note que $Pf_4$ não contribui com a área de PS na imagem da direita. . . . .	24
3.3	Demonstração do PS e da MEES para uma situação em que o robô se encontra inclinado. . . . .	25
3.4	Representação bidimensional de dois pontos de apoio do dispositivo e as variáveis de interesse relacionadas. . . . .	26
3.5	Representação tridimensional de dois pontos de apoio do dispositivo e as variáveis de interesse relacionadas. Adaptado de MESSURI e KLEIN (1985). . . . .	27
4.1	Demonstração do acoplamento entre a perna e o EspeleoRobô. Cada perna é rígida em formato de semi-arco e está ligada ao corpo do dispositivo através de uma junta rotacional. . . . .	30
4.2	Marcha tripé em um dispositivo hexápode. Adaptado de SICILIANO e KHATIB (2008). . . . .	31
4.3	Polígonos de suporte gerados durante a marcha tripé. PS1 e PS2 são os polígonos formado por cada tripé e PSI é a interseção destas duas formas. . . . .	31
4.4	Diagrama do <i>Buehler's Clock</i> aplicado em uma perna do EspeleoRobô. . . . .	32
4.5	Relação de $\phi_s$ e $\omega$ com a movimentação do robô. . . . .	33
4.6	Exemplo de diferentes perfis de pneu: liso (à esquerda) e rugoso (à direita). (Arquivo público da internet). . . . .	34
4.7	Comparação entre dispositivos com esteiras (a), quatro rodas fixas (b) e seis rodas fixas (c). Adaptado de MANDOW <i>et al.</i> (2007). . . . .	35
4.8	Pose de um dispositivo móvel com rodas no plano. . . . .	36
4.9	Localização dos CIR dos flancos e do veículo. . . . .	37
4.10	Trajectoria efetuada pelo dispositivo simulado e a calculada através do modelo cinemático (à esquerda). Módulo do erro entre a trajetória efetiva realizada e a estimada pelo modelo cinemático (à direita). . . . .	39
4.11	Comparação entre as cinemáticas do EspeleoRobô com 4 e 6 rodas para as mesmas velocidades. . . . .	40
5.1	Comparação entre as imagens geradas pelo protótipo e pelo dispositivo simulado. . . . .	42
5.2	Topologia de interação entre os <i>softwares</i> utilizados. . . . .	43

5.3	Modos de locomoção analisados nos ensaios de simulação. a) 6 rodas; b) 6 pernas; c) 4 rodas e 2 pernas (híbrido); d) 4 rodas e 2 “estrelas”; e) 6 “estrelas” f) exemplo de um modo de locomoção com a torre de mapeamento acoplada. . . . .	46
5.4	Plataforma utilizada para realizar os ensaios de estabilidade. . . . .	47
5.5	Ângulo de Tombamento para os modos de locomoção sem (à esquerda) e com a torre de mapeamento (à direita). . . . .	47
5.6	Margem de Estabilidade Energética para os modos de locomoção sem (à esquerda) e com a torre de mapeamento (à direita). . . . .	48
5.7	Os três tipos de PS apresentados pelos modos: a) 6R; b) 4R2P; c) 6R, 6E e 6R2E. Em azul, a MEES na situação de maior estabilidade. . . . .	49
5.8	Escadaria utilizada no ensaio de transposição de obstáculos. . . . .	49
5.9	Transposição de um obstáculo com o modo 4R2P: a) início do movimento apenas com rodas; b) dispositivo travado no obstáculo; c) utilização das pernas para sair do bloqueio; d) obstáculo transposto. . . . .	50
5.10	Plataformas utilizadas no ensaio. À esquerda, a plataforma livre e, à direita, trajeto com obstáculos. . . . .	51
5.11	Resultados referentes aos ensaios em ambiente controlado. . . . .	52
5.12	Detalhe do trajeto que cada modo de locomoção tem que realizar dentro do modelo tridimensional da N4E26, desde o ponto de partida até o ponto de chegada. . . . .	54
5.13	Resultados da potência consumida e resistência específica para o ensaio do trajeto em ambientes representativos. . . . .	55
5.14	Resultados do Ângulo de Tombamento e da Margem de Estabilidade Energética para o ensaio do trajeto em ambientes representativos. . . . .	55
5.15	Resultados da velocidade de translação para o ensaio do trajeto em ambientes representativos. . . . .	56
5.16	Comparação dos modos de locomoção em situação genérica através dos índices referentes às métricas de mobilidade. . . . .	58
5.17	Comparação do desempenho geral dos modos de locomoção, sem e com a carga acoplada. . . . .	59
5.18	Comparação dos modos de locomoção em ambiente representativo através das métricas de mobilidade. . . . .	60
5.19	Comparação do desempenho geral dos modos de locomoção em um ambiente representativo. Resultados para os modos sem e com a carga acoplada. . . . .	62
6.1	Primeiros testes realizados com o EspeleoRobô, sendo o teste na Minas da Passagem em Mariana-MG (à esquerda) e no campus da UFOP em Ouro Preto-MG (à direita). . . . .	63

6.2	Inspeção em mina abandonada no Bairro do Taquaral, em Ouro Preto-MG. Esta era utilizada para a extração de ouro na época colonial, porém hoje se encontra em zona residencial, tornando-se um risco em potencial para os moradores da região. . . . .	64
6.3	Área do duto inspecionado na Mina Córrego do Feijão. A última foto mostra o EspeleoRobô durante a inspeção. . . . .	65
6.4	EspeleoRobô no modo 6R ultrapassando um entulho de pedras. . . . .	66
6.5	Inspeção na galeria de barragem localizada no Rio do Peixe. EspeleoRobô no modo 6R se deslocando sobre um duto (à esquerda) e visualização das vigas que foram utilizadas como referência visual (à direita). . . . .	67
6.6	Inspeção na galeria da Barragem de Taquaras. À esquerda, EspeleoRobô iniciando o teste. À direita, momento em que um peixe foi flagrado circundando o robô. . . . .	68
6.7	Inspeção na galeria de barragem na Minas de Gongo Soco. As duas primeiras imagens mostram o momento em que o robô entram no duto. A terceira imagem mostra em detalhes as rugosidades da estrutura. . . . .	68
6.8	Testes realizados do EspeleoRobô com a torre de mapeamento acoplada. . . . .	69
A.1	Topologia de interação das bibliotecas em Python. . . . .	79

# Lista de Tabelas

2.1	Tabela descritiva dos elementos construtivos do EspeleoRobô de acordo com a Figura 2.1. . . . .	12
4.1	Tabela com valores de $\lambda$ e $Y_{CIR}$ para o EspeleoRobô obtidos em simulação. . . . .	38
5.1	Parâmetros dos motores e reduções utilizados no protótipo do EspeleoRobô. . . . .	44
5.2	Modos de locomoção utilizados e suas siglas. . . . .	46
5.3	Resultado da altura máxima transposta por cada modo de locomoção em ordem decrescente de altura superada. . . . .	50
5.4	As métricas de mobilidade e seus símbolos. . . . .	57
5.5	Taxa de queda de desempenho devido a adição de carga no índice geral em situação genérica. . . . .	59
5.6	Taxa de queda de desempenho devido a adição de carga no índice geral em ambientes representativos. . . . .	62

# Lista de Abreviaturas e Siglas

**ABS** Acrilonitrila Butadieno Estireno

**MEES** Análise da Margem de Estabilidade

**AT** Ângulo de Tombamento

**API** *Application Programming Interface*

**BIR** Instituto Brasileiro de Robótica

**BS** Borda de Suporte

**CIR** Centro Instantâneo de Rotação

**CMU** Carnegie Melon University

**CONAMA** Conselho Nacional do Meio Ambiente

**MEN** Margem de Estabilidade Energética

**FRESE** Flexible Robot for Exploration of Subterranean Environments

**GdL** Grau de Liberdade

**GNSS** Global Navigation Satellite System

**HD** *High Definition*

**IMU** *Inertial Measurement Unit*

**IROS** *International Conference on Intelligent Robots and Systems*

**ITV** Instituto Tecnológico Vale

**ITV-RoC** Laboratório de Robótica e Controle do Instituto Tecnológico Vale - Ouro Preto/MG

**LiDAR** *Light Detection And Ranging*

**NSA** Não se aplica

**PS** Polígono de Suporte

**ROS** Robot Operating System

**SS** *Skid-steering*

# Lista de Símbolos

$\chi_b$  Ângulo em que a perna deixa o chão

$\chi_a$  Ângulo em que a perna toca o chão

$\Omega$  Ângulo total rotacionado

$CG$  Posição do centro de gravidade

$d$  Distância total percorrida

$\lambda$  Fator de correção atrelado para o método de cálculo empírico da cinemática de dispositivos Skid-steering

$g$  Aceleração da gravidade

$m$  Massa do dispositivo

$A$  Matriz de cinemática obtida empiricamente para dispositivos Skid-Steering

$R$  Matriz de Rotação

$\widehat{MEES}$  Métrica normalizada: Margem de Estabilidade

$\widehat{AT}$  Métrica normalizada: Ângulo de Tombamento

$\widehat{MEN}$  Métrica normalizada: Margem de Estabilidade Energética

$\widehat{S}$  Métrica normalizada: Potência Elétrica Instantânea

$\widehat{\epsilon}$  Métrica normalizada: Resistência Específica

$\widehat{v}$  Métrica normalizada: Velocidade de Translação

$\Phi_{mr}^{\rightarrow}$  Orientação do robô com respeito ao sistema inercial em rotações de Euler-XYZ

$\phi_f$  Ângulo total da passada rápida

$\phi_s$  Ângulo total da passada lenta

$M_{hor}$  Plano horizontal da Figura 3.1

$M_{incl}$  Plano inclinado da Figura 3.1

$\vec{\xi}$  Pose do robô

$p_{mr}^{\vec{}}$  Vetor de posição do robô com respeito ao sistema inercial

$S$  Potência elétrica

$P_{cg}$  Vetor de posição da projeção no plano horizontal do centro de gravidade

$k_{red}$  Eficiência da redução

$k_{pr}$  Proporção da redução

$\varepsilon$  Resistência Específica

$O_0$  Sistema de coordenadas inercial

$O_1$  Sistema de coordenadas do robô

$T$  Tempo total da passada

$j_j$  Torque aplicado na junta do mecanismo de locomoção

$j_m$  Torque aplicado no eixo do motor

$\omega_f$  Velocidade de rotação na junta do mecanismo de locomoção durante a passada rápida do *Buehler's Clock*

$\omega_s$  Velocidade de rotação na junta do mecanismo de locomoção durante a passada lenta do *Buehler's Clock*

$\omega_m$  Velocidade de rotação no eixo do motor

$v$  Velocidade de translação

$\vec{N}$  Vetor normal ao plano

$\vec{Z}$  Vetor diretor do eixo Z do sistema de coordenadas

# Capítulo 1

## INTRODUÇÃO

A importância do ferro no desenvolvimento das civilizações é indiscutível. Há indícios de exploração e uso deste mineral já em 3.200 a.C. Apesar de todo o avanço tecnológico desde então, este elemento ainda é essencial para a fabricação da maioria dos artefatos modernos. Sua importância é tamanha, que a posse de grandes jazidas deste mineral favorece estrategicamente a nação detentora em vários campos, seja este geopolítico, econômico, tecnológico e/ou militar.

O Brasil possui uma das maiores jazidas de minério de ferro do mundo. Grande parte destes depósitos se encontra na Serra dos Carajás, no Estado do Pará. Somente nesta região, em 2016, a companhia VALE produziu 148 milhões de toneladas de minério de ferro, tendo gerado uma receita próxima de US\$4,5 bilhões na época. Este volume correspondeu a cerca de 40% de sua produção total neste ano (ROSAS,2017).

Devido a alta rentabilidade de suas minas, a região de Carajás recebe especial atenção no sentido de eficiência na produção do minério de ferro. O objetivo é maximizar os ganhos e diminuir as despesas de produção através de tecnologias e técnicas modernas. De acordo com a companhia, na mina de S11D, pertencente ao complexo Carajás, estão sendo empregados *drones* para inspeções diversas, sistema de transporte de minério *truckless*, sistemas modernos de automação, entre outros (VALE,2016).

Diversos desafios estão atrelados à produção mineral. Um destes é a exploração de zonas onde há a presença de cavidades naturais. Esta se trata de uma operação taxativamente regulamentada por órgãos legais e ambientais. O desafio é ainda maior, já que cavernas são recorrentes em áreas de formação ferrífera. Somente na mina S11D existem mais de 30 destas catalogadas (Figura 1.1).

Existem dois dispositivos na legislação brasileira que dissertam sobre cavidades subterrâneas naturais: a Resolução nº347 do Conselho Nacional do Meio Ambiente (CONAMA), que exalta os atributos sociais, ambientais, e culturais das cavidades; e o decreto nº99.556/1990, alterado posteriormente pelo decreto nº6.640/2008, instrumento este que enquadra as cavidades naturais em diferentes categorias, aferindo um nível de proteção a cada uma. Estes níveis podem variar desde preservação total, passando por extração com

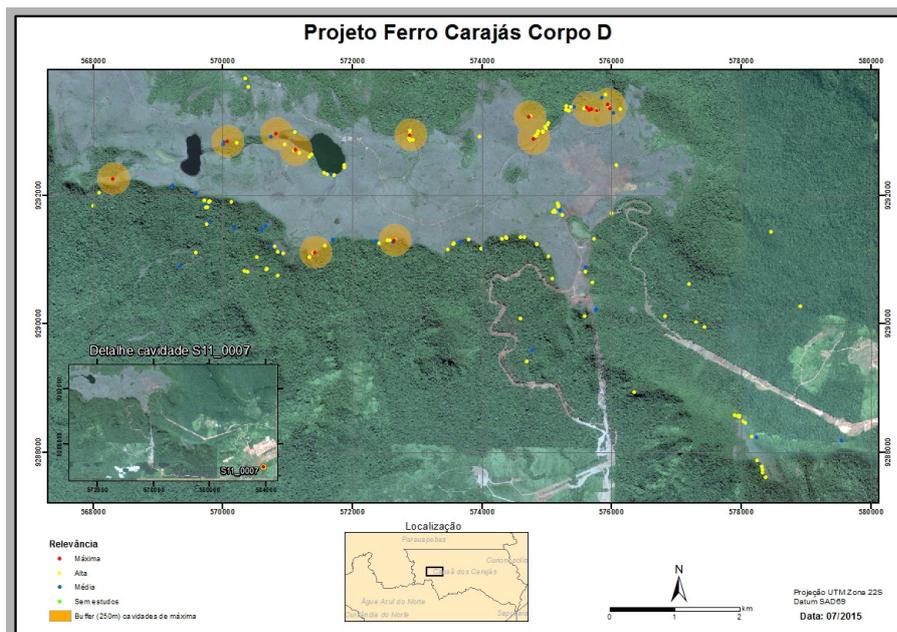


Figura 1.1: Localização de cavidades naturais e zonas de proteção na área da mina de S11D, município de Canaã dos Carajás-PA (Arquivo da Vale).

posterior recuperação, até permissão de supressão da cavidade (DE OLIVEIRA-GALVÃO e DA COSTA NETO,2013).

Certas condições podem levar à restrição exploratória permanente da cavidade e da região no entorno dela. Tal nível de proteção é alcançada em situações, como por exemplo, onde seja encontrada dentro da caverna uma espécie vegetal inédita ou até resquícios culturais de uma antiga civilização.

Afim de viabilizar a exploração, um laudo sobre as características e condições da cavidade deve ser enviado aos órgãos oficiais competentes. Neste documento devem constar informações sobre a geologia (topográfica, gênese, estrutural) e biologia (presença de vida animal, vegetal ou fungi) da cavidade.

O tempo de elaboração deste laudo é crucial, tendo em vista que o processo de lavra em regiões próximas a uma cavidade pode ser suspenso caso haja retardo nos trâmites legais permissivos. Levantar então as informações acerca da situação interna da caverna de forma rápida, eficiente e sem expor os colaboradores a riscos é de suma importância para a produção da mina.

A VALE possui a Gerência de Espeleologia, ligada à Diretoria de Planejamento e Desenvolvimento de Ferrosos, responsável pela espeleologia e assuntos relacionados. Uma de suas atribuições está em realizar a investigação presencial destas cavidades. Espeleólogos com alto grau de treinamento devem entrar nas cavernas para coletarem todas as informações relevantes do ambiente.

Entretanto, cavidades naturais são ambientes hostis ao homem, oferecendo vários riscos, como: desabamento; presença de animais selvagens; fungos ou bactérias que podem



Figura 1.2: Demonstração dos diversos riscos atrelados à inspeção de cavernas quando realizadas por pessoas (Arquivo público da internet).

transmitir doenças; presença de gases tóxicos; ou até regiões de difícil acesso (Figura 1.2). Independentemente do grau, a atuação humana direta no interior destes ambientes está sempre relacionada a riscos de saúde.

## 1.1 O EspeleoRobô

Visando a segurança humana, o grupo de espeleologia tomou em 2014 a iniciativa de adquirir um robô comercial com o intuito de realizar a inspeção de cavidades. Os operadores podem utilizar os dados coletados por este dispositivo afim de analisarem antecipadamente os riscos do ambiente. Sendo estes reduzidos, a inspeção direta pode então ser realizada com mais segurança.

O EspeleoRobô, como é conhecido, é originalmente um robô comercial adquirido de uma empresa brasileira especializada em robótica. Sua morfologia é baseada no robô hexápode RHex (SARANLI *et al.*,2001), da Boston Dynamics (Figura 1.3). Este dispositivo serviu de inspiração por conseguir aliar pequeno tamanho e alta mobilidade; pontos importantes para um robô espeleólogo.



Figura 1.3: Comparação entre as morfologias do RHex (à esquerda) e o EspeleoRobô (à direita).

A primeira versão do EspeleoRobô apresentava várias restrições de funcionamento. As pernas perdiam sincronização poucos passos após o início da operação, o que levava

o robô a se locomover de forma anômala, chocando por diversas vezes o chassi contra o solo. A comunicação era feita de forma analógica e unidirecional, utilizando um rádio de aeromodelismo para enviar comandos ao dispositivo e dois rádios no robô para retornar as imagens das câmeras. Era comum inclusive a queda de comunicação com poucos metros adentrados na cavidade. Dentre outras restrições.

Devido a isto, o dispositivo foi enviado ao Laboratório de Robótica e Controle (ITV-RoC) do Instituto Tecnológico Vale (ITV - Ouro Preto, MG) para receber melhorias em seus componentes. Em um ano de trabalho, praticamente todo o *hardware*, *software*, base de controle e sistema de comunicação foram substituídos.

Uma das implementações mais relevantes é o sistema intercambiável dos modos de locomoção. Este permite que o EspeleoRobô utilize outros mecanismos para se locomover além de pernas, como rodas, rodas tipo “estrela”, e até configurações híbridas.

Inúmeras montagens podem então ser realizadas, cada uma possuindo distintas vantagens e desvantagens. Conhecer as características de cada modo de locomoção disponível é importante afim de configurar o robô com a melhor opção para cada tipo de terreno.

Originalmente, os únicos dados fornecidos aos espeleólogos eram imagens de câmeras; uma localizada na parte frontal e outra na posterior da carcaça. LEDs de cor branca e/ou infravermelho podiam ser utilizados em auxílio às filmagens.

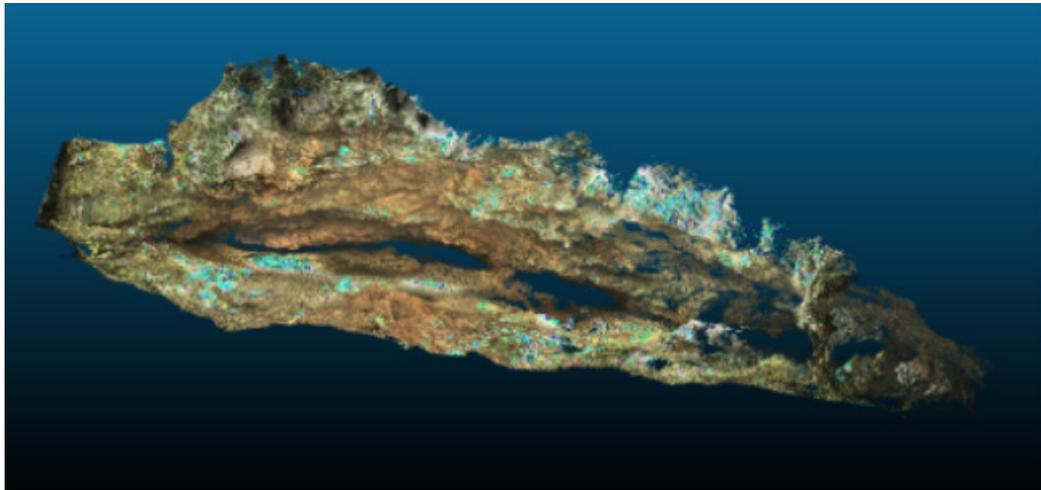


Figura 1.4: Mapa tridimensional gerado a partir da fusão de dados de sensores referente a uma cavidade localizada em área de exploração ferrífera, Carajás-PA.

Com o intuito de aumentar as capacidades sensoriais do robô, uma torre de mapeamento está sendo desenvolvida para levantar informações mais detalhadas sobre a caverna. O projeto é resultado de uma parceria entre ITV/VALE (Ouro Preto - MG) e o Instituto Brasileiro de Robótica (BIR), ligado ao SENAI/CIMATEC (Salvador - BA). O propósito é que seja gerado como produto da inspeção um mapa tridimensional colorido (Figura 1.4). Este mapa permitirá que os especialistas analisem diversas propriedades da cavidade, como: gênese; percolação de fluidos; geotecnia na parte de estabilização; pesquisa

mineral; entre outras.

Um dos maiores desafios na implementação de melhorias no EspeleoRobô está na sua condição de robô industrial ativo. Nenhuma modificação em suas características pode se estender por longos períodos de tempo, já que o mesmo é constantemente requisitado para realizar inspeções de campo e portanto deve estar sempre operacional.

Diferentes abordagens de trabalho têm que ser adotadas afim de respeitar esta restrição. Novos equipamentos e recursos são construídos e testados externamente, sendo implantados no robô apenas quando finalizados. Simuladores também são visados, já que permitem testar diversos conceitos em ambiente virtual, antes da implantação no protótipo.

## 1.2 A robótica móvel na exploração ambiental

Na área de investigação ambiental usando robôs móveis, diversos dispositivos já foram desenvolvidos. Como exemplo, o Robô Ambiental Híbrido (Figura 1.5) é um desenvolvimento brasileiro da PETROBRAS. Sua concepção foca na inspeção e monitoramento de áreas da floresta amazônica, especificamente por onde passa o gasoduto Coari-Manaus. Esta região possui terreno complexo, unindo trechos secos, florestas, rios e pântanos (SANTOS *et al.*,2009).



Figura 1.5: Modelo intermediário do Robô Ambiental Híbrido (SANTOS *et al.*,2009).

As rodas deste dispositivo são construídas em material de baixa densidade dotadas de aletas, todas conectadas ao chassi através de suspensões ativas. Quando em solo firme, as suspensões podem se reconfigurar para garantir a estabilidade do robô. Em água, estas aumentam a altura para que o robô fique mais distante do líquido. As rodas então submer-

gem, permitindo que as aletas sirvam de propulsores. Esta característica híbrida, unindo vantagens de diferentes sistemas de locomoção, permite ao Robô Ambiental Híbrido se locomover com sucesso através de terrenos com características antagônicas, o que seria impossível através de mecanismos convencionais.



Figura 1.6: Robô *Groundhog*, da Carnegie Mellon University (MORRIS,2005).

Tratando especificamente de robôs móveis desenvolvidos para a investigação de ambientes subterrâneos, pode-se citar o *Groundhog* (BAKER *et al.*,2006), desenvolvido na Carnegie Mellon University (CMU). Este se trata de um robô com rodas *off-road* e articulação ativa (Figura 1.6). Seu objetivo é entrar de forma autônoma em minas abandonadas para avaliar suas condições estruturais.

Minas subterrâneas são ambientes criados visando a facilidade de entrada e saída de pessoas e materiais. Por isto, estas possuem geralmente terreno estruturado e espaço para manobras. Restrições a estas somente aparecem em lugares onde tenha ocorrido um deslizamento ou abandono de materiais.

Aproveitando-se destas características do ambiente, o *Groundhog* não foi desenvolvido focando baixos peso ou tamanho do dispositivo. Suas dimensões são 1m de altura e 1,2m de largura, pesando cerca de 700kg. Devido a sua grande massa, os motores devem desenvolver maiores torques e ser acoplados a grandes reduções. Isto reduz a velocidade e eficiência energética do robô, fazendo com que este atinja apenas 0,15m/s a uma potência de 1kW.

Cavidades naturais são ambientes mais complexos de serem explorados. Robôs com grandes dimensões não conseguem passar por boa dos parte espaços confinados existentes. Peso elevado também é desaconselhável, já que causa vibrações no terreno durante a locomoção do robô, podendo danificar a estrutura da caverna. Devido a isto, dispositivos de peso e tamanho reduzidos são mais indicados para fins de inspeção espeleológica.

O robô FRESE (SILES e WALKER,2009) foi desenvolvido visando especificamente a exploração de cavidades naturais. Sua construção foca principalmente em conceder grande mobilidade ao dispositivo. O modo de locomoção se baseia em seis *whegs*, cada uma possuindo um atuador independente. Estes mecanismos proporcionam ao robô boa mobilidade em terreno acidentado; em contrapartida, há perda na capacidade de *payload* e velocidade máxima atingida pelo dispositivo.

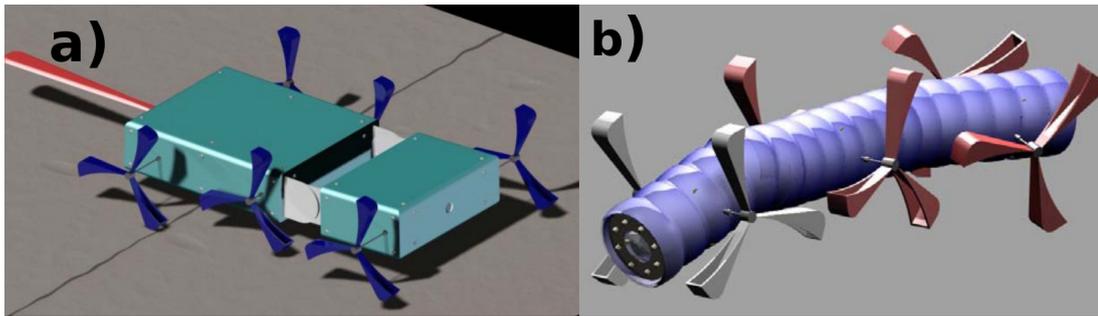


Figura 1.7: Modelos dos robôs: FRESE I - Robô com articulação no meio-corpo (a); FRESE II - Robô completamente articulado (b). Ambos os robôs utilizam o mesmo tipo de mecanismo de locomoção. (SILES e WALKER,2009)

O corpo do robô é formado por dois elos unidos por uma junta rotacional passiva (Figura 1.7-a). Esta junta tem liberdade de movimentação de  $\pm 90^\circ$  e proporciona maior flexibilidade ao dispositivo, permitindo que obstáculos maiores que suas dimensões sejam transponíveis. A segunda versão do FRESE, por sua vez, tem a carcaça baseada em um tubo flexível (Figura 1.7-b). Apesar disto, o ângulo máximo de curvatura do robô é de  $\pm 40^\circ$ . O *payload* do robô também é menor devido a sua estrutura maleável. Entretanto, suas dimensões reduzidas permitem que ambientes de menores dimensões possam ser explorados.

Os trabalhos de WHITTAKER *et al.* (2003) e LARSON *et al.* (2014) apresentam dispositivos móveis com corpo alongado ativamente reconfigurável (Fig. 1.8). O foco desta arquitetura está no acesso ao ambiente a ser inspecionado, permitindo ao robô passar através de pequenos orifícios. Apesar de versátil, esta montagem não permite que cargas de maior volume ou peso sejam adicionadas ao robô, como é o caso da torre de mapeamento proposta.

### 1.3 Desafios da robótica em ambientes confinados

O uso de robôs móveis no interior de cavernas é geralmente relacionada a diversos desafios. Esta seção demonstra os pontos mais relevantes da operação do EspeleoRobô nestes ambientes.



Figura 1.8: Robô CTER configurado para entrar no ambiente confinado através de um pequeno orifício (à esquerda) e para efetuar a inspeção do mesmo (à direita) (LARSON *et al.*,2014).

### 1.3.1 Comunicação

Dispositivos teleoperados necessitam manter constante comunicação com a base de controle. Uma primeira abordagem é a utilização de cabos para isto. A troca de dados é segura e o risco de erros no envio das informações é bem baixo. Todavia, o alcance do robô é restrito ao comprimento do cabo. Há também a possibilidade do mesmo se prender em algum obstáculo, conseqüentemente restringindo a mobilidade do robô e requisitando intervenção humana direta para efetuar o resgate do equipamento.

A comunicação sem fio proporciona maior flexibilidade. A restrição de locomoção se limita à distância máxima de transmissão dos aparelhos. A existência de galerias em cavidades comumente impede a visada direta das antenas, o que impacta na transmissão de dados. Além disto, cavernas de formações ferríferas podem causar reflexões do sinal. Estas restrições podem ser parcialmente contornadas utilizando redes *mesh* e protocolos robustos de troca de dados.

### 1.3.2 Mobilidade

A mobilidade é um fator crucial na concepção de robôs móveis para a exploração de cavidades naturais. O projeto deve considerar que estes ambientes podem apresentar diferentes tipos de terrenos, como trechos planos, acidentados, submersos, buracos, barrancos, entre outros (Figura 1.9).

Devido ao fato de grande parte dos terrenos de cavidades serem acidentados, a capacidade do dispositivo em transpor obstáculos deve ser alta. Neste caso, o uso de pernas e/ou mecanismos com maior poder de transposição de obstáculos é recomendado. Entretanto, geralmente há um compromisso entre mobilidade, capacidade de transportar carga útil (*payload*) e eficiência energética.

A autonomia energética também é um ponto importante. Cavernas podem se estender



Figura 1.9: Mostra de diferentes tipos de terrenos presentes em cavidades naturais, sendo trechos planos (à esquerda) e acidentados (à direita) (Arquivo público da internet).

por centenas de metros e o robô deve possuir energia o suficiente para realizar toda a sua missão e retornar à base. É importante então que a navegação seja feita levando em consideração este aspecto, dando preferência por exemplo à transposição de terrenos menos acidentados que requerem menor consumo energético.

A estabilidade também é de grande importância, devendo ser garantida a todo momento. Caso o robô capote e não possua meios próprios de se desvirar, a intervenção humana se faz necessária, o que nem sempre é possível por questões de segurança ou acesso. Além disto, um tombamento pode danificar equipamentos transportados pelo dispositivo.

## 1.4 Objetivos

Tendo em vista o sistema intercambiável dos mecanismos de locomoção do EspeleoRobô, o objetivo principal desta dissertação é analisar e comparar características de mobilidade de diferentes modos de locomoção especificamente disponíveis para este dispositivo.

Visando a operacionalização ininterrupta do robô em questão, um objetivo secundário desta dissertação é desenvolver um simulador que viabilize a realização destes estudos de mobilidade de maneira fiel à realidade.

Como objetivos específicos tem-se:

- selecionar as métricas que serão utilizadas para quantificar os quesitos de mobilidade dos modos de locomoção;
- selecionar os modos de locomoção mais relevantes para realizar estudos aprofundados sobre suas características;
- criar um ambiente de simulação completo integrando o *Robot Operating System* (ROS) com o simulador V-REP, MATLAB e programas em Python;

- implementar em simulação o modelo virtual do EspeleoRobô;
- criar cenas para a realização dos ensaios simulador, incluindo pelo menos uma em ambiente representativo;
- definir padrões de ensaios a serem utilizados para coleta sistemática de dados sobre os modos de locomoção;
- desenvolver bibliotecas em Python que permitam obter parâmetros do simulador e calcular as métricas de mobilidade em tempo de execução;
- realizar ensaios através de simulações com o dispositivo virtual;
- realizar testes de campo com o protótipo do robô;
- analisar e comparar os dados obtidos.

## 1.5 Organização da Dissertação

O Capítulo 2 é dedicado às melhorias e adaptações que foram realizadas no protótipo do EspeleoRobô. São vistas as implementações em sua estrutura mecânica, eletrônica e de *software*.

No Capítulo 3 é realizado um estudo bibliográfico sobre as métricas de mobilidade utilizadas na dissertação. São vistas metodologias referentes à aferição do consumo energético e a estabilidade de um dispositivo móvel.

As características dos principais mecanismos de locomoção utilizados são vistas no Capítulo 4. Um estudo sobre a utilização de pernas e rodas no EspeleoRobô é feito.

O Capítulo 5 dispõe sobre as simulações realizadas. Primeiramente é demonstrado como o simulador é implementado e como as métricas de mobilidade são obtidas e calculadas a partir do ambiente virtual. Os ensaios propostos e seus resultados são expostos, sendo por fim realizadas discussões acerca destes.

No Capítulo 6 são apresentados os ensaios de campo realizados com o protótipo.

Finalmente, o Capítulo 7 expõe as conclusões obtidas na dissertação, bem como as propostas de trabalhos futuros.

## Capítulo 2

# MELHORIAS E ADAPTAÇÕES NO ESPELEOROBÔ

Como descrito na Seção 1.1, o EspeleoRobô é originalmente um robô comercial e se encontra atualmente no laboratório ITV-RoC do ITV de Ouro Preto/MG. Em pouco mais de um ano de trabalho, várias melhorias foram realizadas no dispositivo em suas partes mecânicas, sistema de locomoção, eletrônica, comunicação e *software*.

Um dos maiores desafios, como já exposto, é que este se trata de um robô industrial ativo. Sendo assim, o processo de implementação não pode deixar o dispositivo inativo por longos períodos de tempo. Além disto, todas as melhorias devem ser feitas sobre a mesma base robótica, não sendo possível alterar drasticamente a arquitetura do mesmo.

A seguir estão detalhadas as melhorias implementadas no EspeleoRobô.

### 2.1 Mecânica

Esta seção trata das melhorias mecânicas implementadas no EspeleoRobô. O diagrama de montagem do dispositivo (Figura 2.1) e as respectivas legendas (Tabela 2.1) estão expostas a seguir.

A carcaça da primeira versão do protótipo era construída em sua maior parte em liga de magnésio, sendo este um material leve e resistente.

Certas modificações estruturais do dispositivo foram propostas. As novas peças foram usinadas inicialmente em alumínio, sendo a escolha deste material na fase de prototipagem justificada por seu baixo custo e facilidade de tratamento.

Assim que o projeto atingir a versão final, porém, as peças em alumínio serão substituídas novamente por equivalentes em liga de magnésio.

Tabela 2.1: Tabela descritiva dos elementos construtivos do EspeleoRobô de acordo com a Figura 2.1.

N <sup>o</sup>	Peça	Material constituinte
1	placa lateral externa	poliacetal
2	mancal	magnésio
3	eixo do mecanismo de locomoção	titânio
4	placa lateral interna	magnésio
5	engrenagem sincronizadora do motor	borracha
6	compartimento da bateria	NSA
7	compartimento da câmera e iluminação	poliacetal
8	compartimento do NUC	NSA
9	motores	NSA
10	compartimento do rádio	NSA
11	tampa central superior	poliacetal
12	tampas superiores aletadas	alumínio
13	luva de esticamento	magnésio
14	correia de transmissão	borracha com teia metálica
15	engrenagem sincronizadora movida	magnésio
16	placas internas dianteira e traseira	alumínio e poliacetal
17	redução planetária	NSA

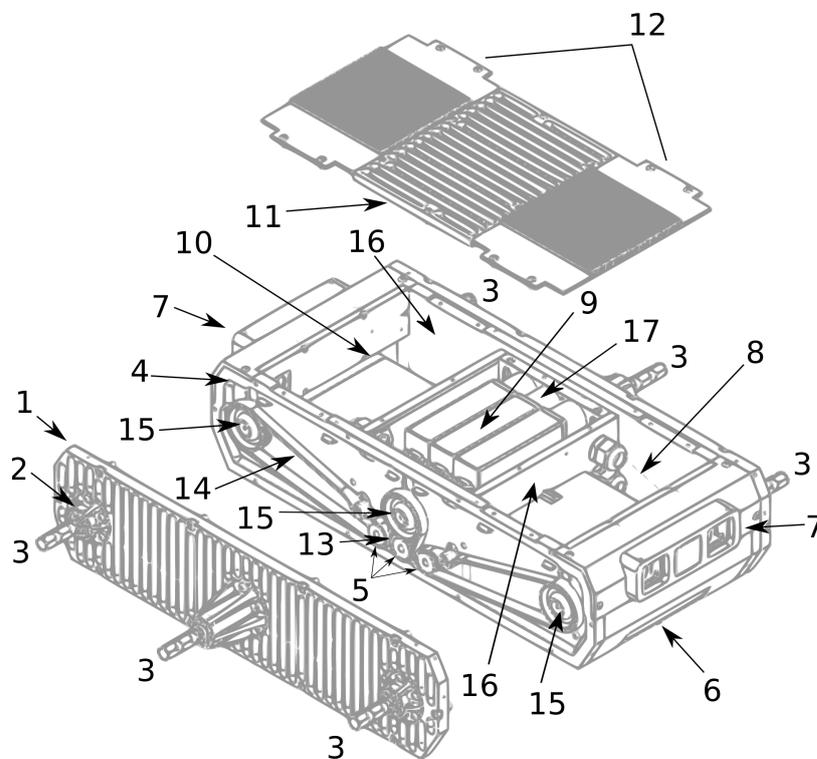


Figura 2.1: Diagrama mecânico de montagem do EspeleoRobô. A descrição dos componentes enumerados pode ser consultada na Tabela 2.1.

Partes do protótipo que não são submetidas a maior estresse mecânico e que não necessitam dissipar calor foram substituídas por placas em poliacetal. Este material é leve, possui boa resistência mecânica, é de relativo baixo custo e de fácil processamento (MELO *et al.*,2003). O poliacetal é cerca de 23% mais leve que o magnésio, reduzindo consideravelmente o peso total do conjunto. As peças usinadas neste material serão mantidas no protótipo final.

Dois para-choques de proteção para as câmeras (Figura 2.1-7) foram criados utilizando técnicas de manufatura aditiva. Suas composições são de Acrilonitrila Butadieno Estireno (ABS) da cor branca. Este material apresenta boa resistência contra impactos mecânicos.

Com as modificações estruturais propostas, o peso do robô diminuiu de 20kg para 17,7kg. Uma redução de 11,5% no valor total.

A distribuição de massa também foi alterada. As baterias (Figura 2.1-6) e câmeras (Figura 2.1-7) estavam instaladas em posições verticais opostas em relação à dianteira e à traseira do dispositivo, o que gerava desequilíbrio no conjunto. Tendo em vista o melhor arranjo, as câmeras foram instaladas na parte superior e o compartimento das baterias na inferior. Devido ao fato das baterias serem cerca de 5x mais pesadas que as câmeras, esta modificação diminuiu a altura do centro de massa em 4mm (FONTES,2017), aumentando a estabilidade do robô.

Em ensaios preliminares do protótipo, notou-se um desgaste anormal de uma das correias de transmissão (Figura 2.1-14). A correia central direita se abrasava contra um limitador devido a uma compressão desnecessariamente causada por sua luva de esticamento (Figura 2.1-13). A correção deste problema veio através do simples desbaste desta luva, o que disponibilizou mais espaço de movimentação para a correia.

Atualmente, duas limitações majoritárias ainda estão em processo de resolução. Cada motor (Figura 2.1-9) possui uma redução planetária (Figura 2.1-17) acoplada diretamente a seu eixo. A engrenagem sincronizadora do motor (Figura 2.1-5) está acoplada por uma luva ao eixo da redução. Há uma folga de  $\pm 15^\circ$  em duas destas engrenagens, o que inviabiliza diretamente a obtenção da posição angular exata dos eixos.

A segunda é sobre o torque máximo desenvolvido pelo dispositivo. Os motores disponibilizam diretamente 54mN.m. Existem duas reduções no sistema, sendo uma de 33:1 devido a redução planetária e uma de 2:1 proveniente da relação entre a engrenagem sincronizadora do motor (Figura 2.1-5) e a engrenagem sincronizadora movida (Figura 2.1-15), o que resulta em uma redução conjunta de 66:1.

Sendo a eficiência final da transmissão de 75%, o torque máximo disponível no eixos dos mecanismos de locomoção (Figura 2.1-3) é de 2,67N.m. Testes preliminares com o protótipo mostraram que este torque é insuficiente para o dispositivo executar com sucesso certos modos de locomoção, como o hexápode. A velocidade máxima atingida pelos eixos também é desnecessariamente elevada, tendo em vista a natureza das missões do EspeleoRobô.

A resolução de ambos os problemas já está em fase de implementação. As reduções acopladas aos motores serão substituídas por outras de maior proporção e a folga nas luvas será corrigida. A redução do conjunto final será de 222:1, gerando um torque final de aproximadamente 9N.m no eixo do mecanismo de locomoção.

### 2.1.1 Sistema de troca rápida dos modos de locomoção

Cavidades naturais podem possuir terrenos nas mais diversas configurações topográficas. Além disto, apesar do EspeleoRobô ter sido concebido visando a exploração de cavernas, suas características também o permitem explorar uma vasta gama de espaços confinados. Desta forma, diferentes perfis de terreno são deparados durante as missões.

Por ter sua morfologia baseada no robô hexápode RHex, o EspeleoRobô podia se locomover originalmente utilizando apenas o modo hexápode. Se comparada à locomoção genérica com rodas, esta é relativamente mais eficiente na transposição de terrenos acidentados. Entretanto, ao desenvolver geralmente baixas velocidades, o torque exigido dos motores se torna em média maior, sendo assim menos eficiente energeticamente (SI-EGWART *et al.*,2011).

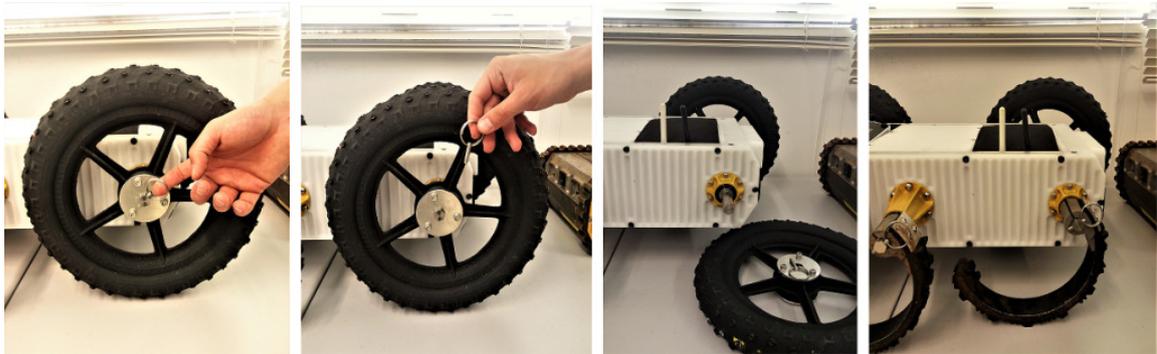


Figura 2.2: Demonstração do funcionamento do sistema de engate rápido (COTA *et al.*,2017).

Em tais condições, utilizar por exemplo o modo de locomoção hexápode em solos planos pode ser visto como desperdício de recursos, já que este tipo de terreno poderia ser transposto com sucesso utilizando rodas.

Visando adequar o robô ao terreno a ser transposto, é desenvolvido um sistema intercambiável dos modos de locomoção por meio de engate rápido (Figura 2.2). Este possibilita que os mecanismos de atuação sejam rapidamente trocado sem a utilização de ferramentas extras. É possível então utilizar: pernas; rodas comuns; rodas tipo estrela; esteiras; entre outros.

O sistema intercambiável proporciona capacidade de adaptação ao EspeleoRobô, permitindo que seja utilizado o modo de locomoção mais adequado de acordo com o tipo de terreno a ser transposto. A montagem é customizável, sendo possível configurar o robô

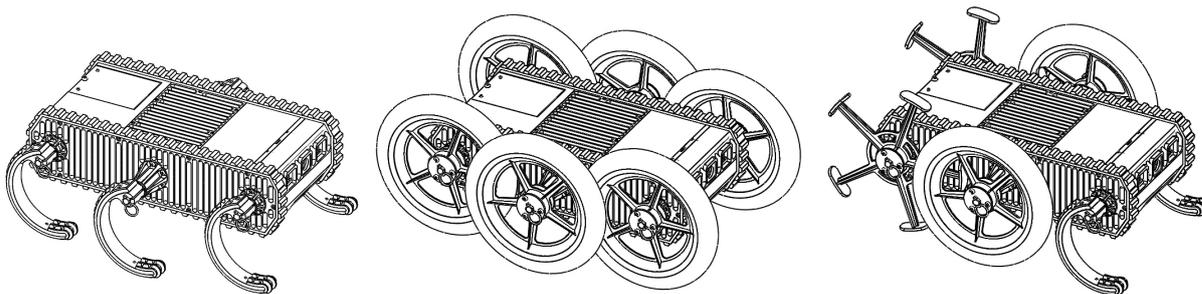


Figura 2.3: Possíveis modos de locomoção do EspeleoRobô: hexápode original, apenas com rodas e híbrida, respectivamente

com apenas um tipo de mecanismo ou também realizar montagens híbridas, utilizando diferentes tipos em conjunto (Figura 2.3).

Além disto, cavernas naturais geralmente se encontram em zonas não acessíveis a veículos. Sendo assim, a não necessidade de ferramental extra para realizar a troca reduz o peso a ser transportado a pé.

## 2.2 Eletrônica

Praticamente toda a arquitetura eletrônica foi revista. A topologia implementada pode ser consultada na Figura 2.4.

Um computador Intel® NUC de alto poder de processamento e dimensões reduzidas foi adicionado como computador central. Um circuito de interface foi desenvolvido para o Arduino® visando a aquisição dos sensores *Hall* e acionamento da iluminação pelo NUC. Câmeras full HD, sendo uma grande angular, foram instaladas nas partes frontal e traseira do dispositivo. Um circuito conversor foi implementado para distribuir a energia provinda das baterias para todos os componentes internos.

Para viabilizar a comunicação com a base de comando, um rádio *wireless* digital Ubiquiti Rocket M900 com duas antenas omnidirecionais foram incorporados ao robô. A base também possui um rádio similar, porém está equipada com uma antena direcional Ubiquiti Airmax Amy 9m16.

A combinação destes rádios e antenas consegue atingir uma distância de comunicação de cerca de 20km em visada direta (UBIQUITI,2012). Também é possível utilizar uma rede *mesh*, permitindo a instalação de repetidores de sinal em caso de cavidades mais extensas e ramificadas.

A base de operações do robô conta com um Dell® Latitude Rugged, um computador robusto que consegue operar sob temperaturas extremas, poeira, chuva e até suportar impactos mecânicos.

Os motores do robô são de corrente contínua com escovas de 60W e *driver* de acionamento MCD-EPOS da Maxon Motors. Há controladores PID internos aos *drivers* para

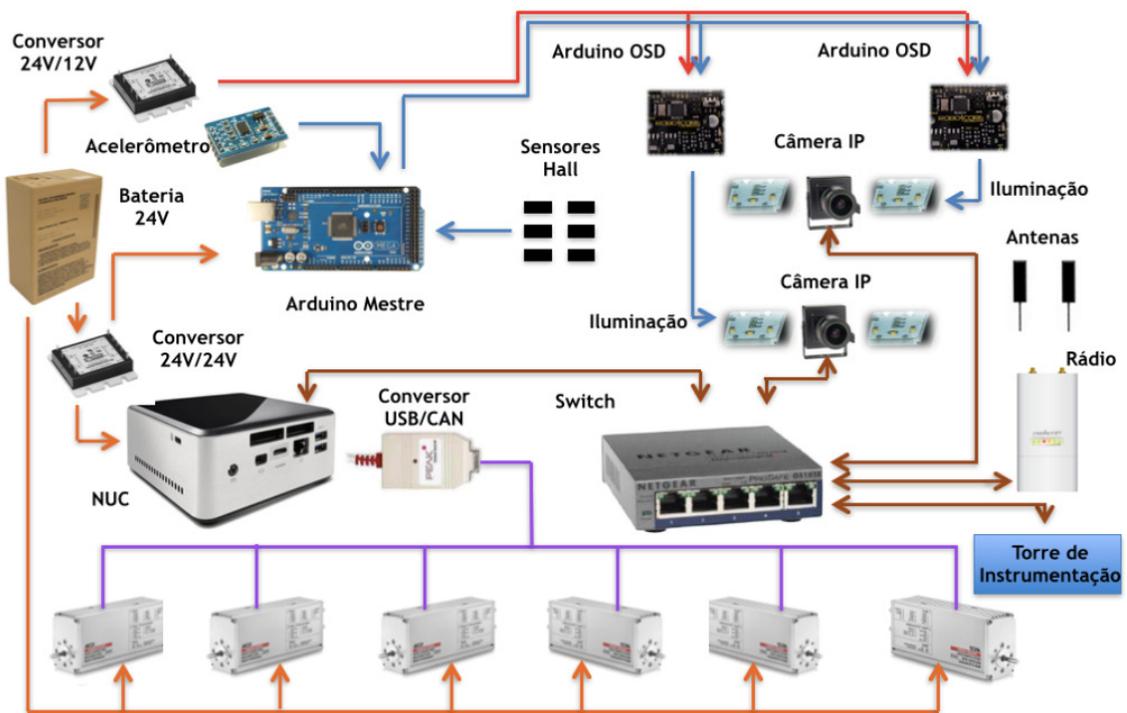


Figura 2.4: Topologia dos periféricos instalados no EspeleoRobô (COTA *et al.*,2017).

atuação em corrente, velocidade ou posição. A comunicação com o computador central é feita via barramento CAN.

## 2.3 Software

Toda a programação do EspeleoRobô foi implementada utilizando o *Robotic Operating System* (ROS). Este se trata de um pseudo sistema operacional, *open-source*, focado na programação de robôs. O ROS oferece serviços como: abstração de *hardware*; controle de dispositivos em baixo nível; implementação de funcionalidades comuns; transporte de mensagens entre processos e gerenciamento de pacotes. As diversas funcionalidades são processadas nos chamados nós, que podem receber ou transmitir dados entre si através de mensagens com formato pré-estabelecido de dados. Em suma, o ROS conecta o *hardware* ao *software*, permitindo uma programação avançada, porém simplificada, sobre dispositivos de baixo nível (O’KANE,2014).

Os nós do ROS trabalham em paralelo. Todo o gerenciamento de memória, tempo de execução e transmissão de mensagens entre os nós é controlada pelo *master* do ROS. Normalmente, cada nó trabalha com um *hardware* individual ou com alguma funcionalidade específica do robô, comunicando-se através da transmissão de mensagens em tópicos. Esta interligação simplificada permite a criação da estrutura lógica que controla todo o robô, desde a leitura dos sensores e acionamento dos atuadores, até o processamento em alto nível dos dados de entrada para uma ação cognitiva do dispositivo como um todo.

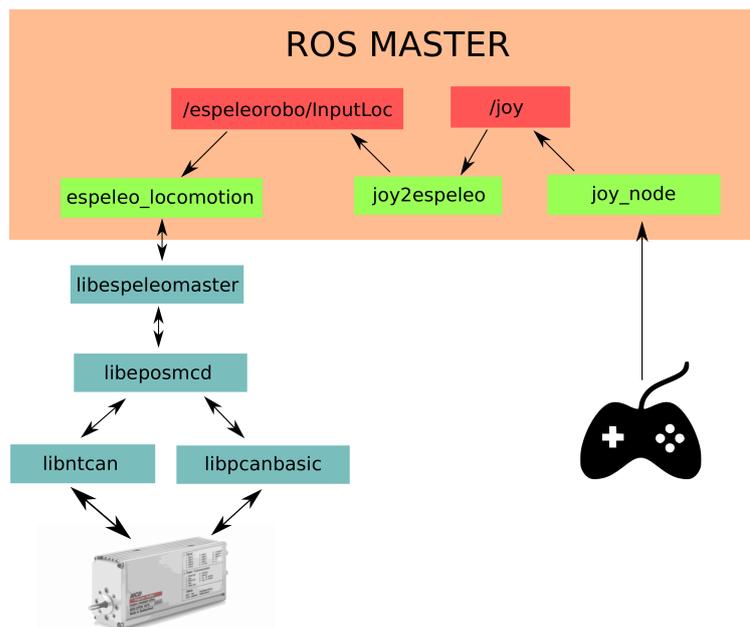


Figura 2.5: Diagrama hierárquico dos códigos de locomoção utilizados. Em vermelho encontram-se os tópicos, em verde os nós e, em azul, as bibliotecas implementadas.

### 2.3.1 Algoritmos de comando para locomoção

Como dito na Seção 2.2, a comunicação com os motores é feita via protocolo CAN. Por não possuir portas deste tipo, é necessário ligar ao NUC um conversor USB/CAN. Estão disponíveis para utilização no laboratório dois conversores, sendo o PCAN-USB, da Peak System e o CAN-USB/2, da ESD. O uso de um ou de outro é ditado pela demanda dos mesmos em outros projetos. Desta maneira, o código de comunicação dos motores foi implementado de forma modular. Isto torna possível a seleção de qual conversor está ligado ao NUC, fazendo com que o código se adapte para se comunicar com os motores. Assim, não são gerados conflitos ou problemas de compatibilidade com os demais códigos.

É proposto que o EspeleRobô seja tele-operado através de um *joystick*. Um botão de troca de modos de locomoção é necessário para que, quando acionado, indique ao ROS que o código de controle dos atuadores também deve ser alterado para o compatível.

Além disto, a navegação tem que ser simplificada para o operador. Este não deve ter que se preocupar com a movimentação específica de cada mecanismo de locomoção. Apenas o sentido de direção deve ser especificado através do controle e o comando de cada perna é executado automaticamente.

Códigos em *C++* foram implementados para viabilizar esta operação, tanto como bibliotecas quanto como nós do ROS. Foram escritas bibliotecas de comunicação com os motores; nós de interpretação dos comandos do joystick; códigos para gerar o comando exato dos motores; entre outros. Um diagrama representando a hierarquia dos códigos de comando do EspeleRobô está disponível na Figura 2.5. A descrição de cada item está explicitada logo em seguida.

- **Bibliotecas:**

- `libntcan`: Biblioteca do *driver* proprietário do conversor CAN-USB/2.
- `libpcanbasic`: Biblioteca do *driver* proprietário do conversor PCAN-USB.
- `libesposmcd`: Biblioteca própria. Fornece serviços de: abstração do *driver* USB-CAN utilizado; escrita de valores na memória do motor; leitura de variáveis de funcionamento do motor; envio de comandos padronizados para o motor. Sua função é basicamente abstrair a utilização do protocolo CAN ao programador, fornecendo uma ferramenta simplificada para acionamento dos motores.
- `libespeomaster`: Biblioteca própria. Esta utiliza os serviços da **`libesposmcd`** para gerar o comando lógico do EspeleoRobô e coordenação das pernas para movimentação. É possível selecionar qual o modo de locomoção está configurado no robô e o respectivo código de comando irá operar. Também fornece a telemetria do dispositivo, publicando valores a respeito do *status* de cada motor, sua velocidade e corrente consumida.

- **Nós do ROS:**

- `joy_node`: Nó que recebe os comandos da manete e publica estes valores no tópico `/joy`.
- `joy2espeleo`: Nó que converte os comandos brutos do *joystick*, provenientes do tópico `/joy`, em um comando genérico entendível pelo EspeleoRobô. Estes comandos são publicados no tópico `/espeleorobo/inputLoc`.
- `espeleo_locomotion`: Este nó basicamente invoca a biblioteca (**`libespeomaster`**), que faz o comando direto dos componentes físicos, e a conecta aos tópicos do nós.

### 2.3.2 Interface de operação

Uma interface de comando (Figura 2.6) é desenvolvida afim de facilitar a teleoperação. Esta se conecta ao *master* do ROS, sendo possível receber parâmetros e enviar comandos diretamente ao robô, sem a necessidade de *plug-ins* ou *sockets* de comunicação.

As informações de cada motor podem ser visualizadas separadamente, sendo disponíveis: corrente consumida; velocidade desenvolvida; status de operação e falhas. O dispositivo possui duas câmeras, uma grande angular em sua parte frontal e uma comum na posterior. É possível a visualização de ambas e a gravação dos vídeos em resolução HD.

O modelo tridimensional do EspeleoRobô também é mostrado, indicando a posição angular de todos os eixos dos motores. Esta informação é útil principalmente quando

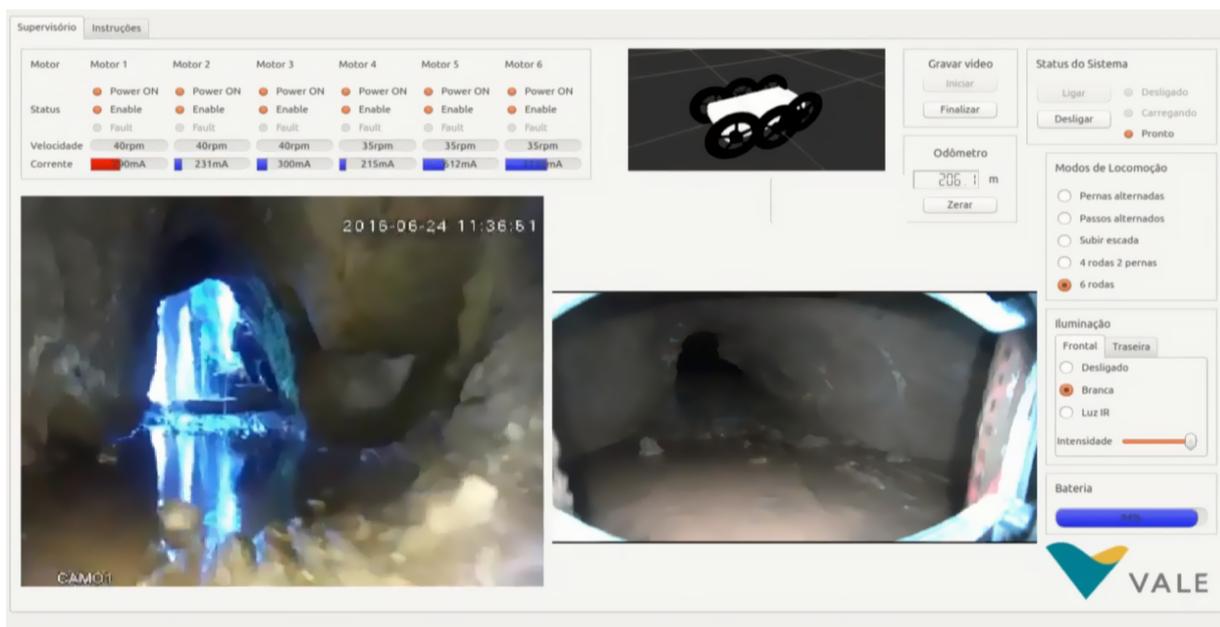


Figura 2.6: Interface de comando do operador.

o robô está no modo com pernas, já que é de grande importância ao operador saber a posição exata de cada mecanismo no instante de transposição de um obstáculo qualquer.

Ao trocar fisicamente o modo de locomoção do EspeleRobô, o operador também pode indicar na interface qual a configuração está sendo utilizada. Desta maneira, o dispositivo irá adequar a forma de acionamento dos motores de acordo com sua montagem.

É possível regular a intensidade das iluminações de luz branca e infravermelha separadamente através destas da tela. Também pode-se visualizar a carga da bateria e ligar/desligar o robô.

## 2.4 Torre de instrumentação

Como citado na Seção 1.1, o dispositivo conta atualmente com duas câmeras para gerar dados sobre a caverna. Uma torre de instrumentação (Figura 2.7) está sendo desenvolvida com o intuito de ampliar as capacidades sensoriais do conjunto. O objetivo é gerar a nuvem de pontos 3D colorida da cavidade (Figura 1.4). Esta unidade de mapeamento está sendo desenvolvida em parceria com o SENAI/CIMATEC (Salvador/BA).

Para realizar o mapeamento, a torre conta com diversos sensores e sistema de iluminação. São utilizados um LiDAR Velodyne VLP-16, um sensor inercial (*Inertial Measurement Unit*, IMU), duas câmeras RGB full-HD e quatro conjuntos de LED de alto brilho. O pináculo da torre rotaciona continuamente a 1rpm.

O sensor LiDAR é capaz de coletar 300 mil pontos por segundo a um alcance de até 100m (VELODYNE,2017). Aliando o alcance do Velodyne à rotação do pináculo, é viável coletar pontos de praticamente todo o entorno do robô. A fusão e tratamento dos dados

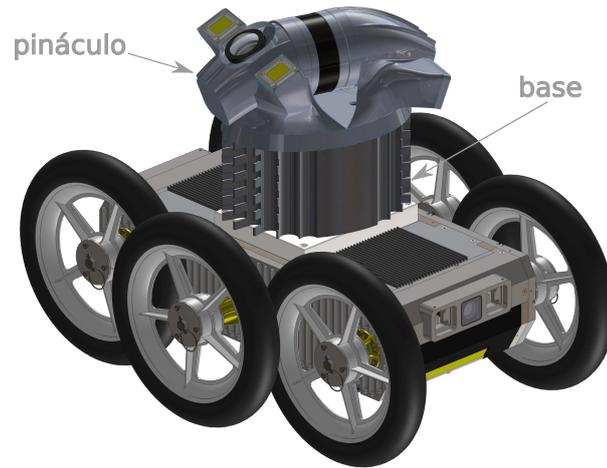


Figura 2.7: Modelo da torre de instrumentação que será acoplada ao EspeleoRobô.

gerados por estes sensores em conjunto possibilita a criação dos mapas tridimensionais coloridos.

O protótipo da torre está operacional e testes preliminares já foram realizados. A mesma pesa 6kg e pode ser facilmente acoplada sobre o EspeleoRobô.

## Capítulo 3

# MÉTRICAS DE MOBILIDADE

A mobilidade é considerada neste texto como a capacidade de um dispositivo em se movimentar livremente por uma área ou espaço quaisquer. Métricas são utilizadas para gerar informações quantizadas acerca de certos parâmetros da mobilidade. Nesta dissertação são tratados especificamente sobre o consumo energético, estabilidade, altura de obstáculo transponível e velocidade de translação. Os resultados obtidos destes métodos permitem avaliar e comparar numericamente os desempenhos de diferentes modos de locomoção.

A altura de obstáculo transponível e velocidade de translação são métricas que podem ser diretamente mensuradas. Por sua vez, as métricas de consumo energético e estabilidade requerem métodos mais sofisticados para sua obtenção, sendo este capítulo dedicado a demonstrá-las.

### 3.1 Consumo energético

A autonomia energética é um tópico de especial importância na robótica móvel. É imperativo que o dispositivo possua uma fonte de energia que o permita se deslocar até o lugar da missão, realizá-la e retornar à base. Nem todos os robôs podem ser continuamente alimentados por uma fonte externa, seja através de cabos, fonte solar ou outra técnica qualquer. Sendo assim, estudar o consumo energético afim de otimizá-lo é crucial para maximizar a autonomia do sistema.

Uma abordagem de análise da eficiência energética é através da potência elétrica instantânea  $S$  consumida pelo dispositivo. Esta indica a quantidade de energia por unidade de tempo fornecida pela bateria.

Mensurar  $S$  durante um teste possibilita a obtenção de informações detalhadas sobre o consumo pontual do dispositivo nas situações encontradas. Além do mais, é possível utilizar os valores de  $S$  coletados para todo um ensaio afim de realizar inferências estatísticas, como obtenção de valor médio, entre outras.

A alteração de massa ou dimensões do EspeleoRobô quando há a troca dos mecanismos é irrelevante. Por isto, a eficiência energética dos diferentes modos pode ser diretamente

comparada com a dos demais através de  $S$ .

Entretanto, dispositivos de diferentes tamanhos e massa possuem necessidades energéticas distintas. Isto faz que o valor de  $S$  não seja válido para uma comparação direta da eficiência energética neste caso.

Uma métrica que permite esta comparação é a Resistência Específica  $\varepsilon$  (VON KARMAN e GABRIELLI,1950). Esta é formalizada por:

$$\varepsilon = S/(mgv), \quad (3.1)$$

que basicamente normaliza a potência elétrica instantânea  $S$  consumida pelo peso  $mg$  do dispositivo a uma dada velocidade de translação  $v$ .

A Resistência Específica é calculada para cada instante de tempo. Assim como para  $S$ , o valor de  $\varepsilon$  pode ser continuamente mensurado afim de realizar inferências estatísticas.

Seu valor adimensional permite comparar diretamente a eficiência energética entre dispositivos de diferentes dimensões e massa. Quanto mais eficiente energeticamente um modo de locomoção for em movimentar o dispositivo móvel, menor será o valor de  $\varepsilon$ .

## 3.2 Estabilidade

Neste trabalho, a estabilidade é tratada como a capacidade do dispositivo em manter a sustentação por meios próprios, sem tombar. Tombamento é uma situação indesejada onde um ou mais mecanismos de atuação perdem efetividade na locomoção e/ou o chassi do robô toca o terreno.

Certas missões do EspeleoRobô são realizadas em ambientes restritos ao acesso humano. Caso haja um tombamento irreversível, poderá não haver maneira de resgatá-lo. Além disto, ao acoplar a torre de instrumentação (como já visto na Figura 2.7), o centro de gravidade do dispositivo desloca 5,19cm para cima, tornando-o menos estável e consequentemente mais propenso a tombar. Desta maneira, analisar a todo momento a estabilidade do conjunto é de suma importância.

Diferentes métricas são consideradas para isto. Estas indicam o quão próximo do tombamento um corpo se encontra em uma situação qualquer. Diversas métricas são propostas na literatura, sendo que cada uma leva em consideração características como: configuração geométrica; forças atuantes; morfologia; altura do centro de gravidade; entre outras.

As métricas de estabilidade podem ser avaliadas de forma dinâmica ou estática. As dinâmicas consideram as forças atuantes no corpo como gravidade, força centrípeta, torque e etc. As estáticas, por sua vez, consideram apenas a força da gravidade e, na prática, são válidas em dispositivos à baixas velocidades e sofrendo a ação de distúrbios irrelevantes.

Visando operações de inspeção em ambientes desconhecidos e não estruturados, apenas

velocidades menores que 1m/s são comumente desenvolvidas. Sendo assim, apenas as métricas estáticas serão utilizadas para a análise da estabilidade.

### 3.2.1 Definições

Algumas definições devem ser feitas antes de expor as métricas. Para tal, considere o exemplo da Figura 3.1, onde o EspeleoRobô se encontra sobre um plano inclinado  $M_{incl}$  (em verde), possuindo simultaneamente  $Q = 6$  pontos de apoio.

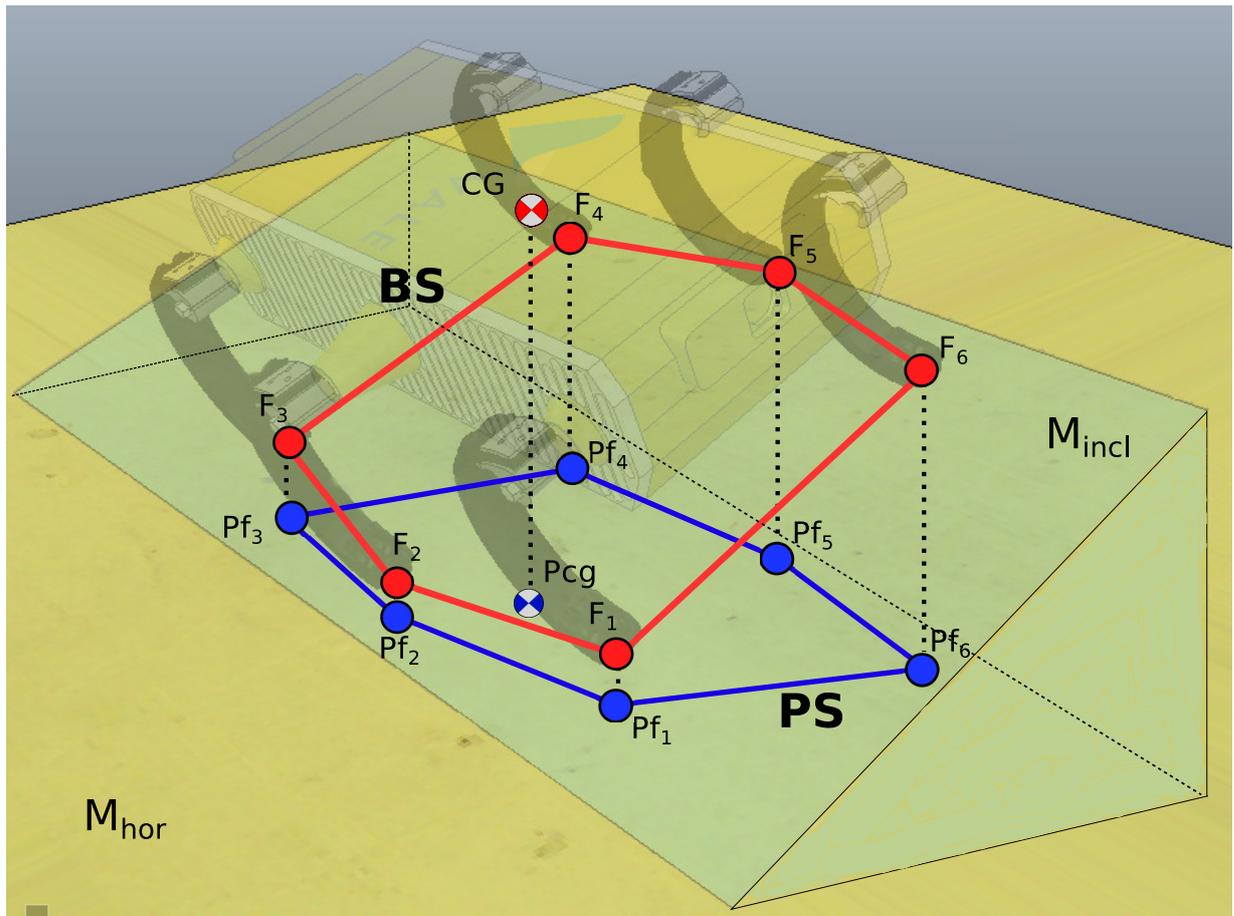


Figura 3.1: Dispositivo móvel sobre um plano inclinado.

O valor de  $Q$  representa a quantidade instantânea de pontos de contato entre o robô e o plano  $M_{incl}$ . O mesmo não deve ser confundido com a quantidade de mecanismos de locomoção acoplados ao dispositivo.

O ponto  $CG$  representa a posição do centro de gravidade. Fazendo a projeção vertical de  $CG$  sobre um plano horizontal  $M_{hor}$  (em amarelo), obtemos o ponto  $Pcg$ . Os pontos de apoio do dispositivo com o terreno (em  $M_{incl}$ ) são representados por  $F_n$ . Suas respectivas projeções verticais também em  $M_{hor}$  são denominadas  $Pf_n$ .

Define-se então como Borda de Suporte (BS) o polígono gerado pela união dos pontos  $F_n$  (MESSURI e KLEIN,1985). Por sua vez, o Polígono de Suporte (PS) é definido como o invólucro convexo do polígono formado pelas projeções  $Pf_n$  (FREITAS,2014).

O termo “invólucro” é utilizado pois, caso alguma das projeções se encontre dentro de PS, significa que esta não ajuda para aumentar a área do mesmo. Consequentemente, o mecanismo não contribui na estabilidade do robô, sendo então irrelevante na análise. A Figura 3.2 demonstra este conceito.

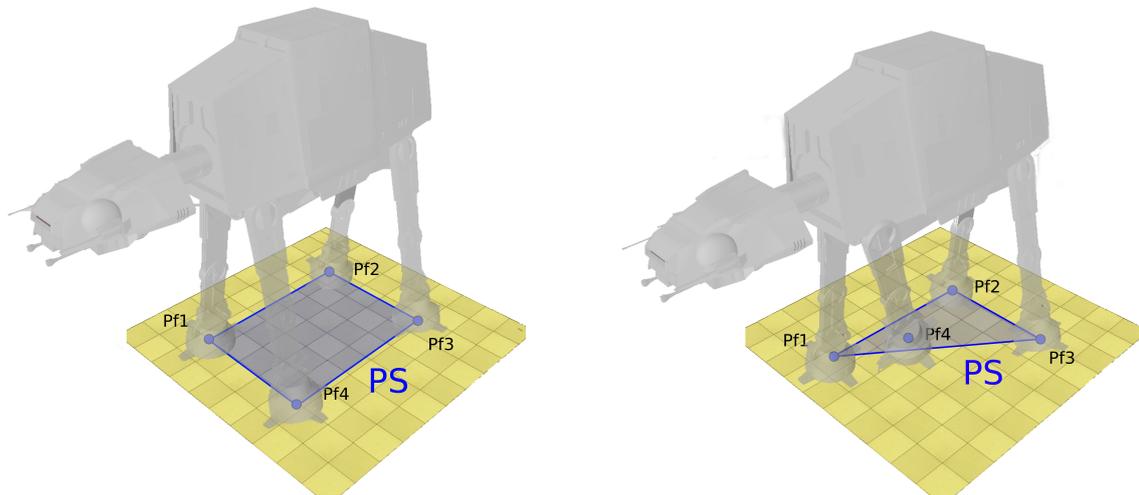


Figura 3.2: Demonstração do Polígono de Suporte para um dispositivo móvel qualquer. Em ambas as situações, todos os mecanismos de locomoção robô tocam o chão, configurando assim pontos de apoio. Note que  $Pf_4$  não contribui com a área de PS na imagem da direita.

Nota-se então que a Borda de Suporte se encontra no plano descrito pelos pontos de contato entre o dispositivo móvel e o chão, representado na Figura 3.1 por  $M_{incl}$ . Já o Polígono de Suporte está sempre representado em um plano horizontal, não precisando necessariamente se tratar de um plano real.

### 3.2.2 Margem de estabilidade estática

A primeira métrica estudada é a Margem de Estabilidade Estática (MEES). Esta leva em consideração para seus cálculos apenas o Polígono de Suporte e  $Pcg$ .

Um exemplo ilustrativo da MEES está representado na Figura 3.3. O polígono azul é o PS, sendo a linha tracejada um indicador do sentido da frontal do robô. Os losangos vermelhos e o círculo vermelho representam  $Pf_n$  e  $Pcg$ , respectivamente.

O valor da MEES equivale à menor distância entre  $Pcg$  e PS, equivalendo então ao vetor do módulo verde no exemplo dado.

Considerando que o PS é formado pela união dos  $Pf_n$  que contribuem na estabilidade do dispositivo, podemos formalizar MEES sendo:

$$MEES = \min_{\forall n \in Q^*} \mathbf{dist}(Pcg, \overrightarrow{Pf_n Pf_{n+1}}), \quad (3.2)$$

onde **dist** é a função que retorna a menor distância entre um ponto e uma reta e  $Q^*$  representa o conjunto dos  $Pf_n$  que fazem parte do PS. Desta forma, é calculada a distância entre  $Pcg$  e todos os segmentos de reta que formam o PS, sendo tomado como valor da MEES a menor distância encontrada.

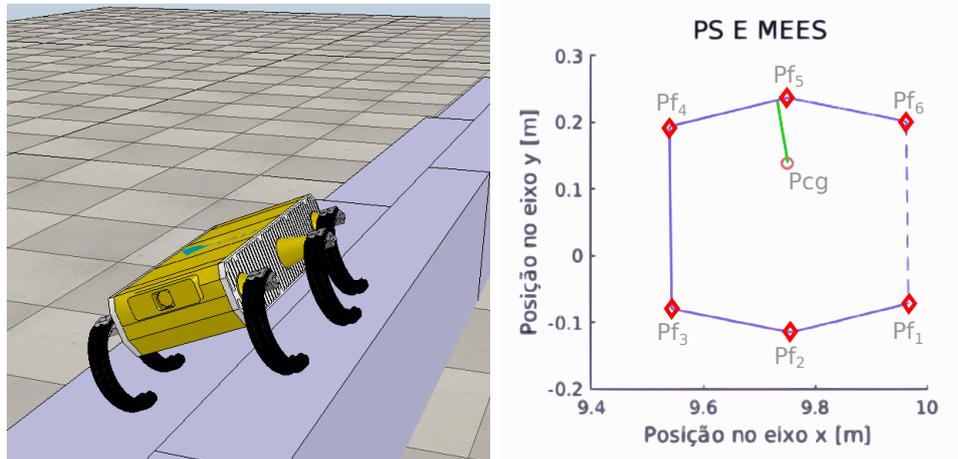


Figura 3.3: Demonstração do PS e da MEES para uma situação em que o robô se encontra inclinado.

Quando  $Pcg$  está dentro do PS, o corpo se encontra em situação estaticamente estável e a MEES assume valores positivos. Quando  $Pcg$  se encontra fora do PS significa que o robô está capotado, fazendo com que a MEES possua valores negativos. Sendo assim, quanto maior o valor encontrado, mais estável se encontra o corpo.

A limitação deste método está em sua análise puramente geométrica através de projeções dos pontos de interesse em um plano horizontal. Não são levadas em conta a altura do  $CG$  ou a massa do corpo. Isto permite que o método consiga avaliar apenas se o corpo se encontra estável ou não. É difícil precisar o quanto ainda falta para um eventual capotamento.

### 3.2.3 Ângulo de tombamento

O Ângulo de Tombamento (AT) é uma métrica que indica diretamente qual é o menor ângulo que o dispositivo pode ser inclinado até capotar. Para tal, calcula-se o ângulo  $\Theta_n$  necessário para rotacionar o corpo até leva-lo ao limiar da estabilidade sobre todas as arestas de sua Borda de Suporte. O valor de AT será então o menor valor  $\Theta_n$  encontrado.

Considere a Figura 3.4, onde há uma ilustração bidimensional do EspeleoRobô sobre uma rampa. Para cada ponto de apoio  $F_n$  neste exemplo, traça-se um vetor  $\vec{G}_n$  que o liga a  $CG$ . Ainda a partir de cada  $F_n$ , descreve-se os vetores  $\vec{G}'_n$  que possuem módulos iguais a seus respectivos  $\vec{G}_n$ , porém apontam no mesmo sentido que  $\vec{Z}$ . O vetor  $\vec{Z} = [0, 0, 1]$  é o vetor diretor do eixo Z do sistema de coordenadas atrelado ao sistema inercial considerado. O valor de  $\Theta_n$  equivale então ao menor ângulo entre os vetores  $\vec{G}_n$  e  $\vec{G}'_n$ .

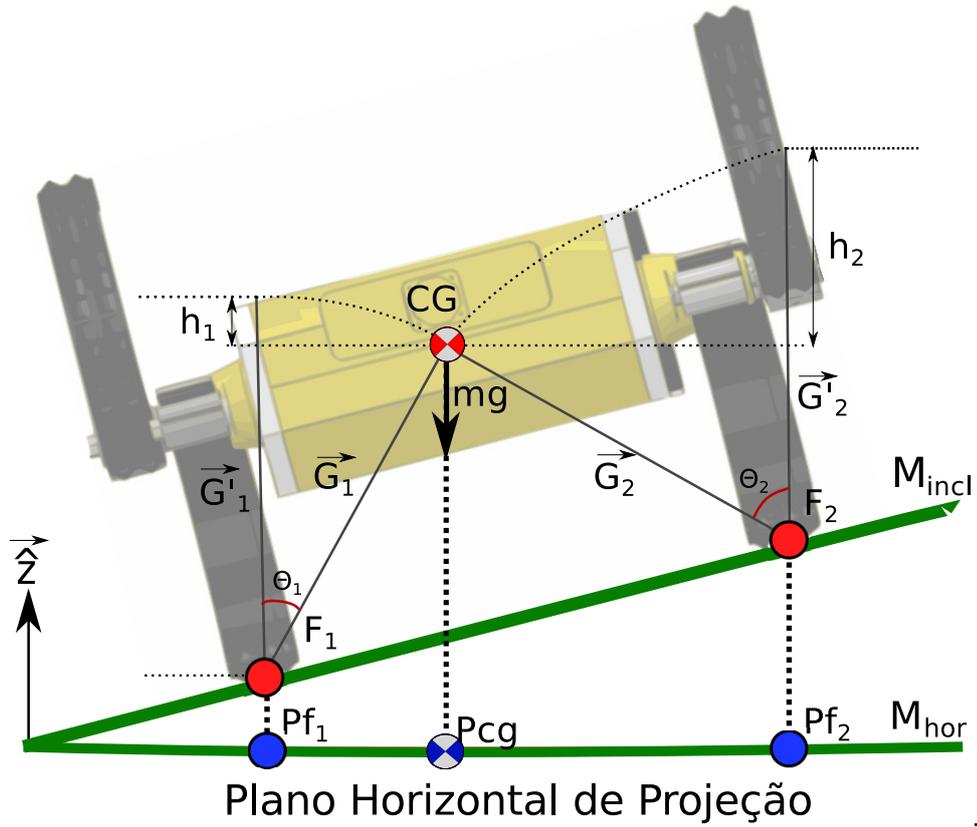


Figura 3.4: Representação bidimensional de dois pontos de apoio do dispositivo e as variáveis de interesse relacionadas.

Na situação do exemplo, nota-se que o corpo se encontra mais propenso a tombar rampa abaixo do que rampa acima. Como discutido na Seção 3.2.2, o corpo se torna instável quando  $Pcg$  se encontra sobre ou fora do PS. Neste exemplo bidimensional, esta situação ocorre quando  $Pcg$  se encontra sobre, ou além, de  $Pf_1$  ou  $Pf_2$ . O fato de  $\Theta_1$  ser menor que  $\Theta_2$  indica a maior facilidade do capotamento ocorrer sobre o ponto de apoio  $F_1$ , sentido morro abaixo.

De forma mais geral, analisa-se agora o caso de  $CG$  e dois pontos de apoio genéricos quaisquer ( $F_1$  e  $F_2$ ) no espaço tridimensional. Para isto, considere a Figura 3.5.

A reta que liga  $F_1$  a  $F_2$  é uma das arestas da BS, considerando-a aqui como  $\overrightarrow{F_1F_2}$ . O *Plano1* pode ser descrito por  $\overrightarrow{F_1F_2}$  e o vetor diretor  $\vec{Z}$ . O vetor normal ao plano pode ser encontrado através de:

$$\vec{N}_{plano1} = \frac{\vec{Z} \times \overrightarrow{F_1F_2}}{|\vec{Z} \times \overrightarrow{F_1F_2}|}. \quad (3.3)$$

O vetor  $\vec{G}_n$  descreve a menor distância entre  $\overrightarrow{F_1F_2}$  e  $CG$ . Cabe ressaltar que, na Figura 3.5,  $\vec{G}_n$  se encontra fora do *Plano1*.

A situação de capotamento é quando  $CG$  está contido em *Plano1*. Faz-se isto rotacionando  $\vec{G}_n$  em torno de  $\overrightarrow{F_1F_2}$ , até que este intercepte *Plano1*, obtendo-se assim o vetor

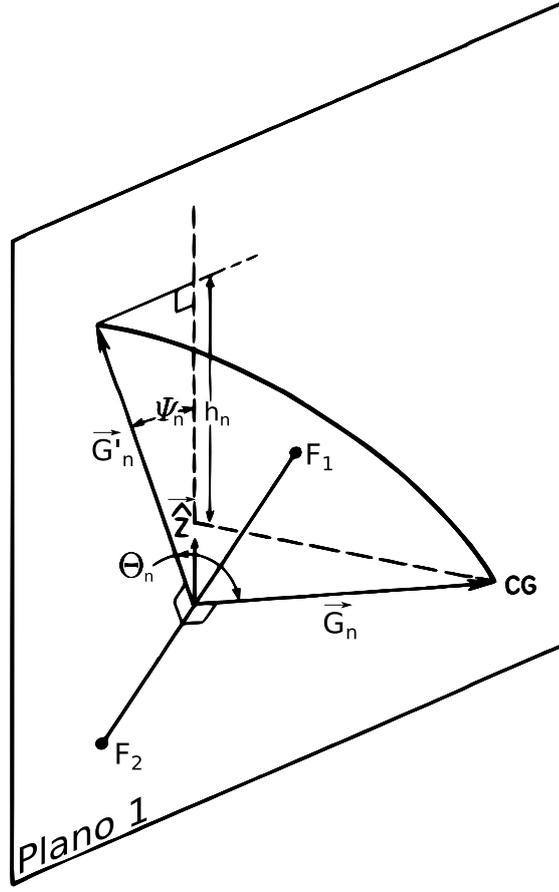


Figura 3.5: Representação tridimensional de dois pontos de apoio do dispositivo e as variáveis de interesse relacionadas. Adaptado de MESSURI e KLEIN (1985).

$\vec{G}'_n$ . O ângulo  $\Theta_n$  para esta aresta específica da BS é dado então pelo menor ângulo entre  $\vec{G}_n$  e  $\vec{G}'_n$ . De maneira genérica, pode-se calcular este ângulo através de:

$$\Theta_n = \arccos \left( \frac{\langle \vec{G}_n, \vec{G}'_n \rangle}{|\vec{G}_n| \cdot |\vec{G}'_n|} \right). \quad (3.4)$$

O vetor  $\vec{G}'_n$  pode ser encontrado pela fórmula:

$$\vec{G}'_n = \vec{G}_n - \left( \vec{G}_n \cdot \vec{N}_{plano} \right) \times \vec{N}_{plano}. \quad (3.5)$$

Tendo então calculado  $\Theta_n$  para todas as arestas da BS, o valor do Ângulo de Tombamento para a pose calculada será igual ao menor valor encontrado. Formalizando:

$$AT = \min_{\forall n \in Q^*} \Theta_n. \quad (3.6)$$

Na prática, qualquer rotação maior ou igual a  $AT$  sobre a aresta de menor  $\Theta_n$  da BS levará o dispositivo ao capotamento. O Ângulo de Tombamento fornece informação mais descritiva sobre a situação atual de estabilidade do dispositivo. Mais importante ainda,

também é possível prever uma iminente situação de capotamento, já que conhecemos o quanto o corpo pode rotacionar sobre sua BS sem tombar.

### 3.2.4 Margem de estabilidade energética

Segundo MESSURI e KLEIN (1985), a métrica da Margem de Estabilidade Energética (MEN) quantifica a energia necessária para capotar o robô a partir de uma situação arbitrária. Basicamente, para o Ângulo de Tombamento de uma pose qualquer, a MEN calcula qual é o trabalho necessário a ser realizado para efetuar esta rotação. Desta maneira, a métrica indica quantitativamente o impacto energético que pode ser suportado pelo dispositivo, sem capotar.

O cálculo se baseia na energia potencial do  $CG$  da inclinação atual até uma pose de iminente capotamento. Analisando ainda o esquema da Figura 3.5, vemos que esta energia é representada por  $mgh_n$ , onde  $h_n$  representa a diferença de altura entre as posições inicial e final do  $CG$ . Esta pode ser encontrada calculando-se primeiramente o ângulo  $\Psi_n$  entre  $\vec{G}'$  e  $\vec{Z}$ :

$$\Psi_n = \arccos\left(\frac{\langle \vec{G}'_n, \vec{Z} \rangle}{|\vec{G}'_n| \cdot |\vec{Z}|}\right); \quad (3.7)$$

e então através da equação (MESSURI e KLEIN,1985):

$$h_n = |\vec{G}'_n| \times (1 - \cos(\Theta_n)) \times \cos(\Psi_n). \quad (3.8)$$

Em posse destes valores, a Margem de Estabilidade Energética  $MEN_n$  relacionada a uma aresta da BS qualquer é dada por:

$$MEN_n = mgh_n, \quad (3.9)$$

onde  $m$  é a massa do dispositivo e  $g$  é a aceleração da gravidade. A MEN global, referente à pose atual do dispositivo, é dada então pelo menor valor de  $MEN_n$  encontrado, sendo:

$$MEN = \min_{\forall n \in Q^*} MEN_n \quad (3.10)$$

Traçando então uma comparação entre o Ângulo de Tombamento e a Energia de Tombamento, o primeiro fornece uma informação mais direta do quanto é possível, a partir de uma pose qualquer, rotacionar o dispositivo até que este capote. A MEN, por sua vez, fornece o trabalho que deve ser realizado para efetuar este movimento; logo, esta é a menor energia capaz de tombar o dispositivo pela aresta da BS que possui o menor  $\Theta_n$ .

## Capítulo 4

# CARACTERÍSTICAS DOS MECANISMOS DE LOCOMOÇÃO

Existem atualmente inúmeras maneiras de locomoção para dispositivos móveis. Podem ser utilizados desde rodas, pernas, asas, hélices; até outros mais inusitados, como ventosas (BRIONES *et al.*,1994) ou imãs (SATO *et al.*,2004). Independente do mecanismo utilizado, sempre haverá vantagens e limitações atreladas.

Por serem os mais utilizados nesta dissertação, este capítulo trata das características do uso de rodas e pernas em dispositivos móveis. O estudo é direcionado para o uso destes mecanismos especificamente no EspeleoRobô.

### 4.1 Robôs com pernas: hexápodes

A utilização de pernas em robôs móveis é um assunto extenso; isto porque existem vários tipos de pernas, esquemas de montagem, métodos de controle, morfologia, entre outras características. Para não se estender, esta seção trata então especificamente das características do dispositivo no qual o EspeleoRobô foi baseado, o robô hexápode RHex (Figura 1.3) da Boston Dynamics (SARANLI *et al.*,2001).

Morfologicamente, as duas maiores inspirações do RHex são o robô SCOUT (BUEHLER *et al.*,1998) e a barata-americana (*Periplaneta americana*), que é comprovadamente a espécie do reino animal que, proporcionalmente ao tamanho de seu corpo, realiza o caminhar mais rápido e mais estável (SPONBERG e FULL,2008).

Mesmo em ambiente hostil, o RHex desenvolve um caminhar ágil e estável. Se os métodos de controle e negociação com obstáculos forem bem implementados, este robô consegue demonstrar excelente mobilidade em terrenos acidentados (BILLAH *et al.*,2008).

O Rhex é concebido visando a incorporação do menor número possível de sensores e atuadores. Suas pernas são semiarcos simples, acopladas ao corpo do robô por juntas rotacionais (Figura 4.1); o que configura um Grau de Liberdade (GdL) por mecanismo

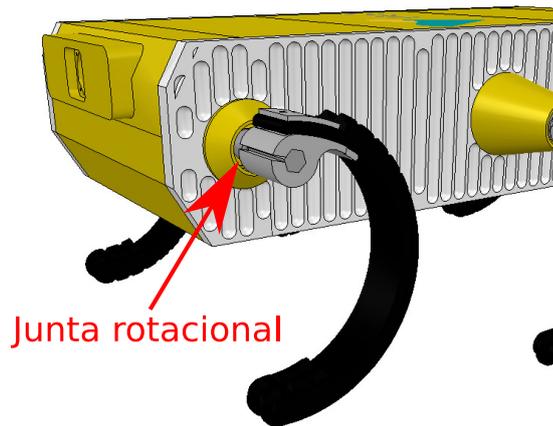


Figura 4.1: Demonstração do acoplamento entre a perna e o EspeleoRobô. Cada perna é rígida em formato de semi-arco e está ligada ao corpo do dispositivo através de uma junta rotacional.

de locomoção. Esta montagem faz com que o movimento das pernas durante a passada se faça acima do plano do dispositivo, evitando a colisão em obstáculos rentes ao solo (JUN e CLARK,2009). Aliando isto a sua construção espelhada com respeito a seu plano sagital, torna-se possível que o robô se locomova de cabeça para baixo.

Esta simplicidade na montagem leva a alta robustez e maior economia de energia (SARANLI *et al.*,2000). Além disto, o robô consegue realizar variados tipos de movimento, indo desde um simples andar estático, até outros mais complexos, como correr, pular, andar com duas pernas ou até subir escadas (MOORE *et al.*,2002).

#### 4.1.1 Marcha tripé com *Buehler's Clock*

Para que dispositivos com pernas se movimentem, uma estratégia de comando coordenado dos mecanismos de locomoção deve ser adotada. Entre várias abordagens existentes, a utilização de marchas se mostra efetiva e simplificada (HAYNES e RIZZI,2006). Em suma, estas são sequências pré-definidas de acionamento dos atuadores. Cada marcha é geralmente concebida focando em um aspecto de locomoção específico: seja na melhor transposição de um tipo de terreno; para privilegiar a estabilidade; para fazer o dispositivo se mover mais rápido; ou até vários destes objetivos em conjunto.

Afim de se manter estaticamente estável no espaço tridimensional, um corpo precisa de no mínimo três pontos de apoio. Por possuir três pernas a mais do que o mínimo necessário, os hexápodes podem utilizar as restantes para realizar manipulações simples ou procurar novos pontos de apoio (TAPPEINER *et al.*,2009). Durante a locomoção, esta característica permite ao dispositivo manter uma pose estável enquanto leva as pernas livres para os pontos de apoio da próxima passada, permitindo assim que uma marcha estática seja efetuada.

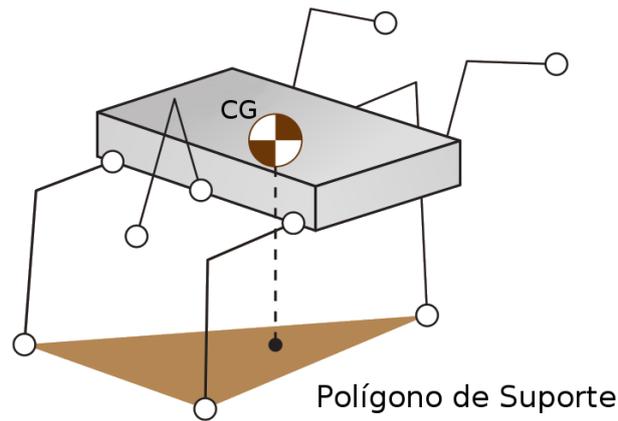


Figura 4.2: Marcha tripé em um dispositivo hexápode. Adaptado de SICILIANO e KHATIB (2008).

Inspirada na biomimética, a marcha estática mais popular em robôs hexápodes é baseada na locomoção de animais que também possuem seis pernas: a tripé alternada (FULL e TU,1990). As seis pernas são divididas em dois grupos que são acionados com um mesmo sinal de controle cíclico, porém defasado entre os grupos (Figura 4.2). Desta maneira, enquanto um tripé realiza uma passada, o outro mantém a sustentação do corpo.

Tendo isto em vista, uma consideração neste tipo de marcha é a existência de dois Polígonos de Suporte que se alternam durante o caminhar. A projeção de  $CG$  deve ser mantida na zona de interseção destes dois polígonos afim de garantir a estabilidade em ambas as passadas, inclusive na transição entre elas. Na Figura 4.3 podem ser observados os dois polígonos formados e a zona de interseção entre eles.

A marcha estática tripé requer uma estratégia de controle da posição angular de cada junta separadamente. Uma muito popular é a de *Buehler's Clock* (SARANLI *et al.*,2001). Este descreve uma trajetória de referência que as pernas devem seguir. Ba-

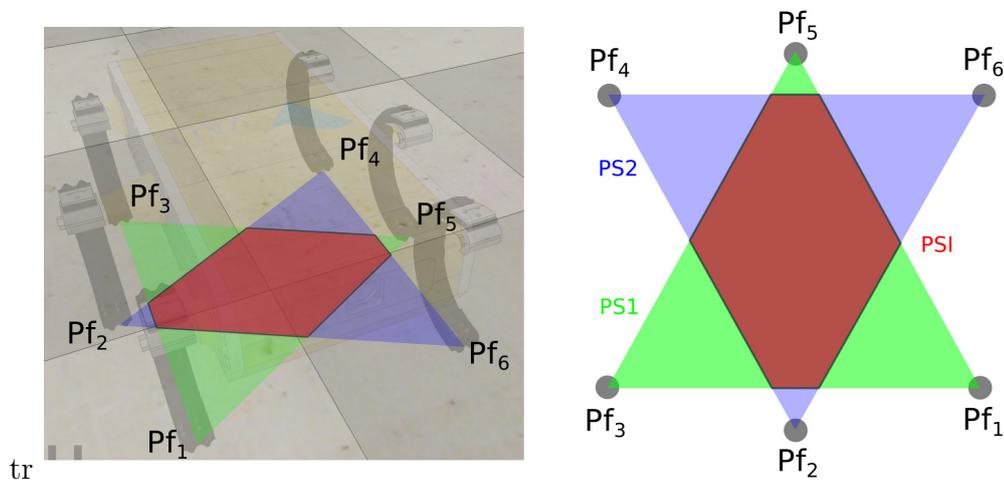


Figura 4.3: Polígonos de suporte gerados durante a marcha tripé. PS1 e PS2 são os polígonos formado por cada tripé e PSI é a interseção destas duas formas.

sicamente, o método divide uma passada em duas etapas, sendo uma lenta, enquanto a perna está tocando o chão e uma rápida, enquanto a perna está rotacionando no ar (JUN e CLARK,2009).

Mesmo em terreno acidentado, a estratégia de *Buehler's Clock* proporciona boas velocidades de translação e estabilidade se comparado a outros tipos de marcha para dispositivos hexápodes.

Na Figura 4.4 está demonstrado este esquema aplicado no EspeleoRobô. O ângulo de referência zero da perna é considerado no sentido para baixo do eixo perpendicular ao plano sagital do robô.

Uma passada completa dura um período  $T$  arbitrário. A fase lenta se inicia quando a perna toca o solo no ângulo  $\chi_a$  e termina quando ela deixa o terreno em  $\chi_b$ ; sua duração é igual a um tempo  $t_s < T$ . A fase rápida então é quando a perna deixa o solo e rotaciona no ar até encontrar novamente o terreno em  $\chi_a$ , levando para isto um tempo  $t_f = T - t_s$ .

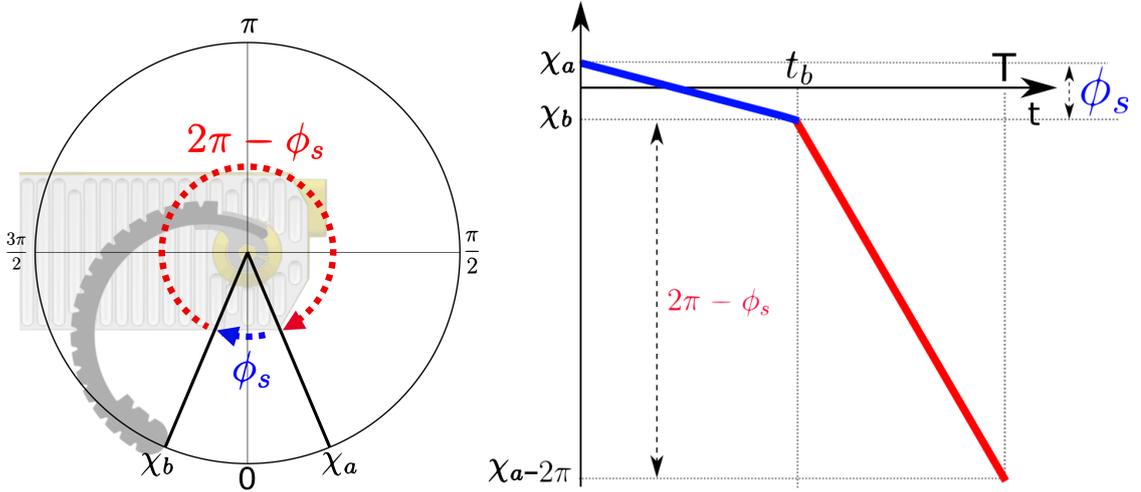


Figura 4.4: Diagrama do *Buehler's Clock* aplicado em uma perna do EspeleoRobô.

Fazendo  $t_s = t_f$ , garantimos que um tripe inicializará a fase rápida no mesmo momento em que o outro inicia a lenta. Isto proporciona maior eficiência na locomoção, já que quando os dois tripés executam simultaneamente a passada lenta, não há garantia que todas as pernas estejam realmente tocando o chão, ou seja, não contribuindo na locomoção.

Por  $\chi_b$  e  $\chi_a$  serem isométricos com respeito ao ângulo zero, o tamanho total da passada lenta ( $\phi_s$ ) e da passada rápida ( $\phi_f$ ) valem:

$$\begin{aligned}\phi_s &= 2\chi_a, \\ \phi_f &= 2\pi - \phi_s.\end{aligned}\tag{4.1}$$

Sendo assim, as velocidades angulares  $\omega_s$  que devem ser aplicadas nas juntas do robô para a fase lenta e  $\omega_f$  para a fase rápida se dão por:

$$\begin{aligned}\omega_s &= \frac{\phi_s}{t_s}, \\ \omega_f &= \frac{2\pi - \phi_s}{T - t_s}.\end{aligned}\tag{4.2}$$

A estratégia de *Buehler's Clock* permite ao dispositivo se locomover em linha reta, andar de ré, fazer curva enquanto anda e rotacionar sobre o próprio eixo quando parado. Estes comportamentos são alcançados regulando os parâmetros de controle  $\phi_s$  para cada mecanismo de locomoção.

Também é possível regular a velocidade de translação do robô. De acordo com a Equação 4.2, a velocidade de rotação da perna - e conseqüentemente do deslocamento do dispositivo - pode ser variada através do tempo total da passada ( $T$ ). Quanto menor  $T$ , mais rápida será efetuada uma passada, e vice-versa. O valor  $T$  deve ser igual para todas as pernas de um mesmo flanco do dispositivo.

Em relação ao sentido em que o robô está se locomovendo, devem ser regulados os ângulos  $\phi_s$  de cada lado do robô e o sentido de rotação das pernas. É demonstrada na Figura 4.5 um esquema com as relações entre estas variáveis e qual o comportamento esperado do dispositivo. Considere que o lado esquerdo do robô é o lado  $a$  e o direito,  $b$ . Considere também  $\omega$  como uma velocidade arbitrária qualquer aplicada nas juntas.

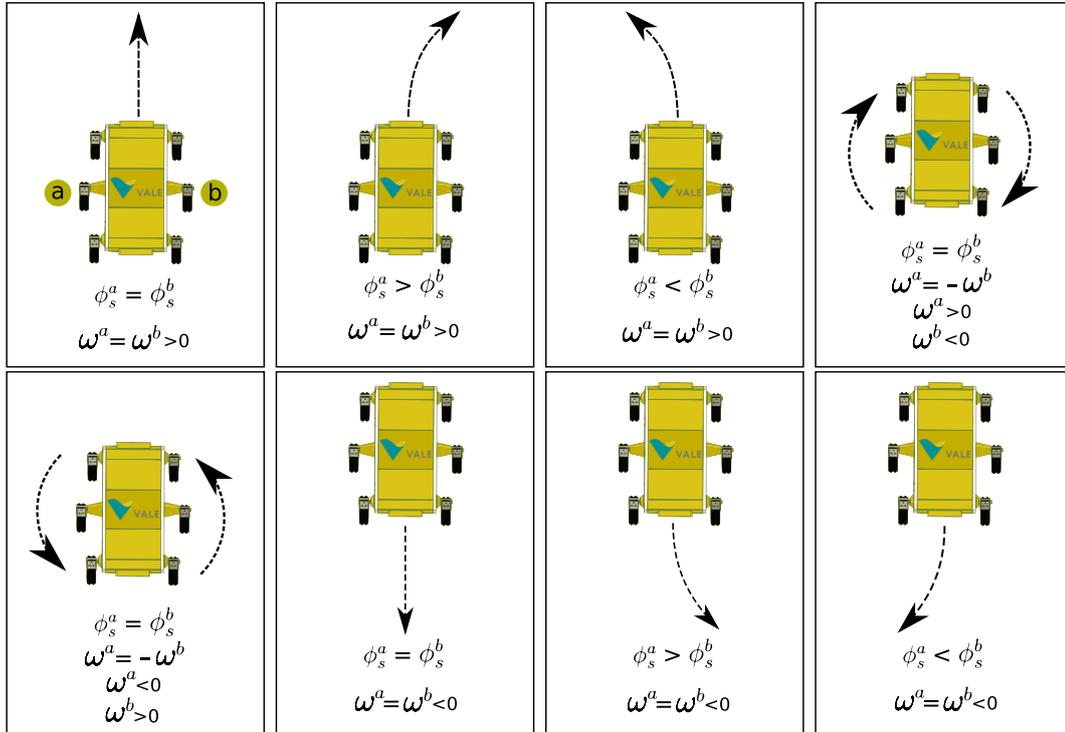


Figura 4.5: Relação de  $\phi_s$  e  $\omega$  com a movimentação do robô.

Caso a marcha e os métodos de negociação de obstáculos forem bem implementados, estes robôs podem apresentar vantagem de mobilidade sobre dispositivos com rodas. As desvantagens residem na maior complexidade do controle de caminhar, em menores

velocidades de deslocamento e na menor capacidade de *payload*.

## 4.2 Robôs com rodas

A utilização de rodas na locomoção de dispositivos móveis leva a vantagens em relação ao consumo energético, velocidade de translação e *payload* (GRAND *et al.*,2004).

Em condições normais, as rodas mantêm continuamente o contato com o terreno, o que proporciona um PS constante em torno do *CG*, aumentando a estabilidade do dispositivo (SICILIANO e KHATIB,2008). Esta mesma característica limita porém a capacidade de transpor obstáculos já que, quando atuando sozinhas, as rodas são capazes de vencer barreiras de tamanho menor que seu próprio raio.

O perfil do pneu é de grande impacto na mobilidade do dispositivo. Em seus experimentos com um robô móvel para fins de exploração lunar, WETTERGREEN *et al.* (2010) tentam relacionar vários aspectos construtivos do robô com seu consumo energético e tração em relação ao terreno. Ao fim, chegam à conclusão que a característica mais impactante nestes dois pontos é o perfil do pneu utilizado.



Figura 4.6: Exemplo de diferentes perfis de pneu: liso (à esquerda) e rugoso (à direita). (Arquivo público da internet).

Não existe um perfil que atenda a todos os terrenos; cada um se comporta melhor em um tipo específico. Perfis lisos garantem alta aderência em terrenos estruturados, permitindo ao dispositivo realizar curvas de menor raio, em alta velocidade, sem perder a aderência. Em terrenos não estruturados, perfis rugosos proporcionam maior tração com o solo. No geral, quanto maior o atrito com o chão, maior é o consumo energético. Estão ilustrados na Figura 4.6 diferentes tipos de pneus.

Em questão de consumo energético, pneus geralmente apresentam maior economia quando comparados às pernas; isto se deve à natureza movimento contínuo que possuem. Durante a locomoção, o dispositivo mantêm suas energias cinética e potencial constantes. Há perdas nas juntas e no contato dos atuadores com o solo, porém estas podem ser minimizadas.

### 4.2.1 Cinemática de dispositivos *Skid-steering*

Um ponto importante no projeto de um dispositivo com rodas é o sistema de direção utilizado. Este conceito remete a qual será a disposição das rodas com respeito ao corpo do dispositivo, à definição da roda ser ativa ou passiva, e a quais serão fixas ou orientáveis. Esta configuração impacta diretamente na mobilidade do robô e na sua cinemática.

A avaliação da cinemática permite relacionar as velocidades dos atuadores com as velocidades de translação e rotação do corpo do robô. Dispositivos móveis que têm rodas alinhadas e não esterçáveis possuem cinemática similar aos *Skid-steering* (SS). Veículos famosos deste tipo são os tanques militares com esteiras.

Ambas as configurações com rodas mais utilizadas no EspeleoRobô, com 4 e 6 rodas, podem ter a cinemática aproximada à dos SS (Figura 4.7).

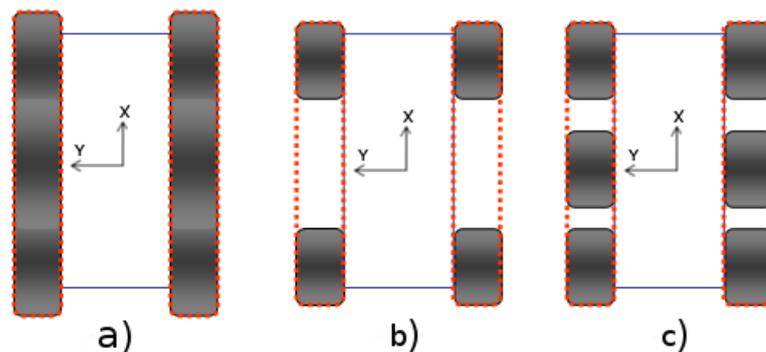


Figura 4.7: Comparação entre dispositivos com esteiras (a), quatro rodas fixas (b) e seis rodas fixas (c). Adaptado de MANDOW *et al.* (2007).

Veículos SS possuem controle parecido aos de direção diferencial, sendo também governados pelas velocidades lineares de seus dois lados. Basicamente, velocidades idênticas proporcionam uma translação linear e um desbalanço entre estas faz com que o dispositivo realize curvas. Rotações sobre o próprio eixo também são factíveis ao se aplicar velocidades de sinais inversos nos lados do robô.

Apesar desta semelhança no controle, é equivocado relacionar a cinemática dos SS com a de veículos de direção diferencial. ANOUSAKI e KYRIAKOPOULOS (2004) demonstram que esta diferença é principalmente devida à significativa derrapagem que ocorre nos SS durante sua locomoção, principalmente durante as curvas.

Justamente devido ao derrapamento, modelos cinemáticos não garantem boa predição do comportamento real de dispositivos SS.

Modelos dinâmicos utilizando informações sobre as forças de interação entre pneu e terreno conseguem obter melhores resultados (HONG e HU, 2017). Entretanto, não estão disponíveis no EspeleoRobô sensores que viabilizem a obtenção deste tipo de modelo, sendo a adotada a modelagem cinemática demonstrada a seguir.

Considere um sistema de coordenadas  $O_1$  convenientemente fixo ao dispositivo móvel, que por sua vez se encontra sobre um plano. A pose do robô (posição e orientação) pode ser descrita com respeito a um sistema inercial  $O_0$  qualquer como:

$$\vec{\xi} = \begin{bmatrix} x \\ y \\ \eta \end{bmatrix}, \quad (4.3)$$

onde  $x$  e  $y$  são as coordenadas cartesianas da origem de  $O_1$  e  $\eta$  é o menor ângulo entre os vetores diretor  $\vec{X}1$  de  $O_1$  e  $\vec{X}0$  de  $O_0$  (Figura 4.8).

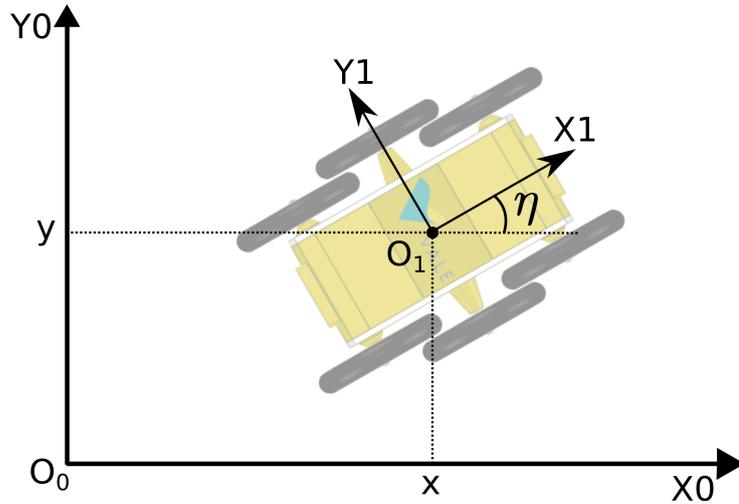


Figura 4.8: Pose de um dispositivo móvel com rodas no plano.

Por estar no plano, todas as rotações realizadas pelo dispositivo serão em torno de um vetor diretor ortogonal a este plano. Desta maneira, a matriz de rotação para este sistema será sempre da forma:

$$R(\eta) = \begin{bmatrix} \cos(\eta) & -\sin(\eta) & 0 \\ \sin(\eta) & \cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

As velocidade de translação e rotação do dispositivo com respeito ao sistema inercial podem ser relacionadas através da equação

$$\dot{\vec{\xi}} = R(\eta) \begin{bmatrix} v_{x1}(t) \\ v_{y1}(t) \\ \omega_{z1}(t) \end{bmatrix}, \quad (4.5)$$

sendo  $v_{x1}$  e  $v_{y1}$  as velocidades lineares do dispositivo longitudinal aos eixos  $\vec{X}1$  e  $\vec{Y}1$ , respectivamente;  $\omega_{z1}$  é a velocidade angular de  $O_1$  em torno de seu próprio eixo Z.

Através desta definição, uma abordagem para modelar a cinemática de dispositivos SS é fazer:

$$\dot{\xi}(t) = R(\eta) \begin{bmatrix} v_{x1}(t) \\ v_{y1}(t) \\ \omega_{z1}(t) \end{bmatrix} = R(\eta)A \begin{bmatrix} V_l(t) \\ V_r(t) \end{bmatrix} = R(\eta)A \begin{bmatrix} r\omega_{j_l}(t) \\ r\omega_{j_r}(t) \end{bmatrix}, \quad (4.6)$$

onde  $A$  é uma matriz que depende das coordenadas do Centro Instantâneo de Rotação (CIR) (MARTÍNEZ *et al.*,2005).  $V_l(t)$  e  $V_r(t)$  são as velocidades lineares dos flancos esquerdo e direito,  $\omega_{j_l}(t)$  e  $\omega_{j_r}(t)$  são as velocidades angulares das juntas dos flancos esquerdo e direito e  $r$  é o tamanho do raio das rodas.

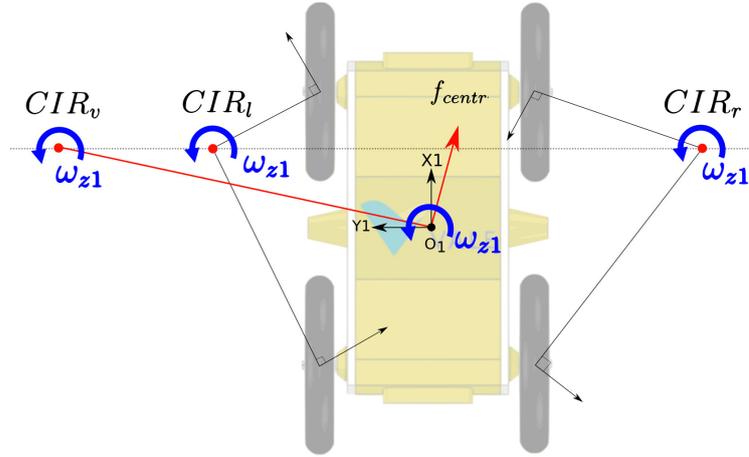


Figura 4.9: Localização dos CIR dos flancos e do veículo.

É ilustrado na Figura 4.9 a localização dos CIR dos flancos esquerdo ( $CIR_l$ ), direito ( $CIR_r$ ) e do veículo como um todo ( $CIR_v$ ) para uma situação arbitrária. Estas podem ser referenciados com respeito a  $O_1$  através das coordenadas  $CIR_v = [X_{CIR_v} \quad Y_{CIR_v}]^T$ ,  $CIR_l = [X_{CIR_l} \quad Y_{CIR_l}]^T$  e  $CIR_r = [X_{CIR_r} \quad Y_{CIR_r}]^T$ . Estes pontos podem ser determinados através das equações:

$$Y_{CIR_v} = \frac{-v_{x1}}{\omega_z}; \quad (4.7)$$

$$Y_{CIR_l} = \frac{\lambda_l V_l - v_{x1}}{\omega_{z1}}; \quad (4.8)$$

$$Y_{CIR_r} = \frac{\lambda_r V_r - v_{x1}}{\omega_{z1}}; \quad (4.9)$$

$$X_{CIR_v} = X_{CIR_l} = X_{CIR_r} = \frac{v_{y1}}{\omega_{z1}}, \quad (4.10)$$

onde  $\lambda_l$  e  $\lambda_r$  são fatores de correção dos flancos devido à mecânica do robô, ao tipo de terreno e outros fatores. Note que  $Y_{CIR_v}$  pode variar de  $-\infty$  a  $+\infty$  dependendo da rotação que o veículo faz sobre seu próprio eixo ( $\omega_{z1}$ ). Entretanto, os valores  $Y_{CIR_l}$ ,  $Y_{CIR_r}$

e  $X_{CIR_v}$  se mantém em uma região restrita e finita, já que o numerador e denominador das equações 4.8 a 4.10 mantém suas ordem de grandeza relativamente próximas.

Relacionando as equações 4.7 a 4.10 com 4.6, temos que a matriz  $A$  vale:

$$A = \frac{1}{Y_{CIR_l} - Y_{CIR_r}} \begin{bmatrix} -Y_{CIR_r} \alpha_r & -Y_{CIR_r} \alpha_r \\ X_{CIR_v} \alpha_l & -X_{CIR_v} \alpha_l \\ \lambda_l & -\lambda_r \end{bmatrix}. \quad (4.11)$$

Em dispositivos com modelo cinemático simétrico, a matriz  $A$  pode ser simplificada para:

$$A = \frac{\lambda}{-2Y_{CIR}} \begin{bmatrix} -Y_{CIR} & -Y_{CIR} \\ 0 & 0 \\ 1 & -1 \end{bmatrix}. \quad (4.12)$$

Para a equação 4.12, os valores de  $\lambda$  e  $Y_{CIR}$  podem ser encontrados de forma experimental através de um método proposto por MANDOW *et al.* (2007).

A determinação de  $\lambda$  se dá através de um teste de locomoção em linha reta com duração  $T$  qualquer. Devem ser coletadas a distância  $d$  total percorrida pelo dispositivo no percurso e as velocidades  $V_l$  e  $V_r$  desenvolvidas. A partir destes dados, temos que:

$$\lambda = \frac{2d}{\int_0^T V_r(t)dt + \int_0^T V_l(t)dt}. \quad (4.13)$$

O segundo parâmetro  $Y_{CIR}$  é encontrado através de um teste rotação pura também com duração  $T$  arbitrária. Devem ser aplicadas velocidades  $V_l$  e  $V_r$  de módulos iguais mas sinais inversos. Devem ser mensuradas as velocidades dos flancos ao longo do ensaio e o ângulo total rotacionado ( $\Omega$ ) pelo dispositivo, assim:

$$Y_{CIR} = \frac{\int_0^T V_r(t)dt - \int_0^T V_l(t)dt}{2\Omega}. \quad (4.14)$$

Com a matriz  $A$  computada, obtem-se o modelo cinemático proposto na Equação 4.6.

Os parâmetros  $\alpha$  e  $Y_{CIR}$  foram encontrados para o EspeleoRobô através de simulação. Estão disponíveis na Tabela 4.1 os valores obtidos.

Tabela 4.1: Tabela com valores de  $\lambda$  e  $Y_{CIR}$  para o EspeleoRobô obtidos em simulação.

Modo	$Y_{CIR}$	$\lambda$
4 rodas	0.531	0.965
6 rodas	0.400	0.965

Utilizando ainda o mesmo ambiente de simulação para comparar a trajetória estimada pelo modelo cinemático e a realizada pelo dispositivo virtual, obtém-se os resultados exibidos na Figura 4.10.

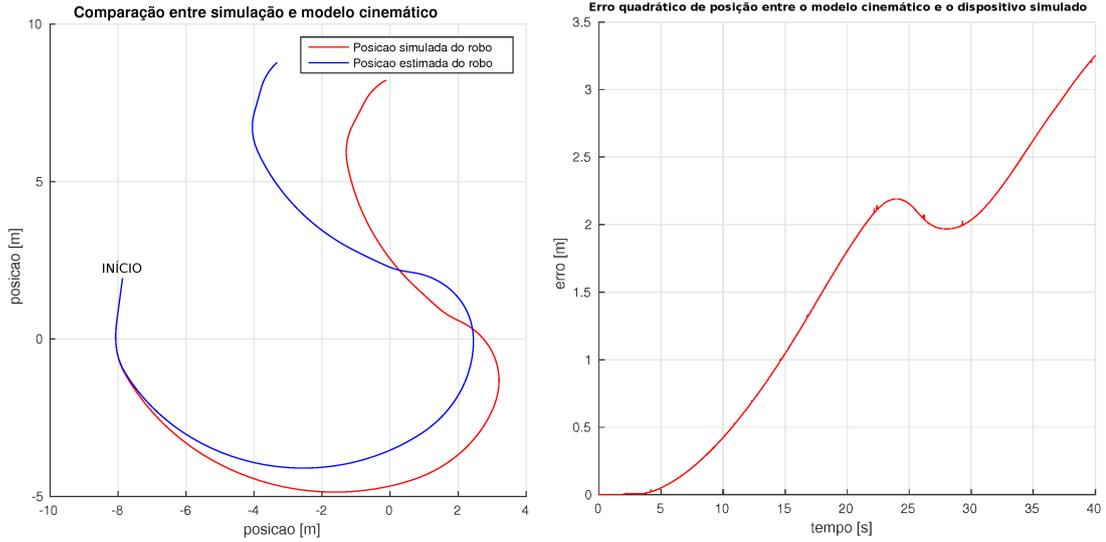


Figura 4.10: Trajetória efetuada pelo dispositivo simulado e a calculada através do modelo cinemático (à esquerda). Módulo do erro entre a trajetória efetiva realizada e a estimada pelo modelo cinemático (à direita).

O erro quadrático de posição entre modelo cinemático e dispositivo virtual tende a crescer à medida que a trajetória é efetuada. Isto demonstra a não representatividade do modelo com o comportamento real do robô. Considerando ainda os parâmetros relevantes para a obtenção do modelo, ao se alterar características do robô ou do solo, a desconexão entre estes aumenta ainda mais.

Tendo em vista porém que o robô atualmente é teleoperado, este modelo se torna aceitável. Eventuais correções entre trajetória efetuada e desejada podem ser realizadas diretamente pelo operador.

A influência das rodas centrais na realização de curvas também é avaliada através deste modelo cinemático. Velocidades constantes de  $\omega_{j_l} = 2rad/s$  e  $\omega_{j_r} = 3rad/s$  dos flancos são aplicadas nos modelos cinemáticos do robô nas configurações com 4 e 6 rodas.

A trajetória descrita pelo modo com 6 rodas possui menor raio se comparada a do modo com 4 rodas, para um mesmo comando de entrada (Figura 4.11). Isso mostra a importância da contribuição que um atuador central tem na realização de curvas em um dispositivo SS.

### 4.3 Conclusões

Em geral, as rodas são mais adequadas a terrenos planos ou pouco acidentados, desenvolvendo locomoção mais eficiente neste tipo de ambiente. Já as pernas provém melhor mobilidade em terrenos mais acidentados, permitindo o alcance de pontos de apoios impossíveis às rodas.

Na locomoção com pernas, há constante alteração nas energias cinética e potencial do

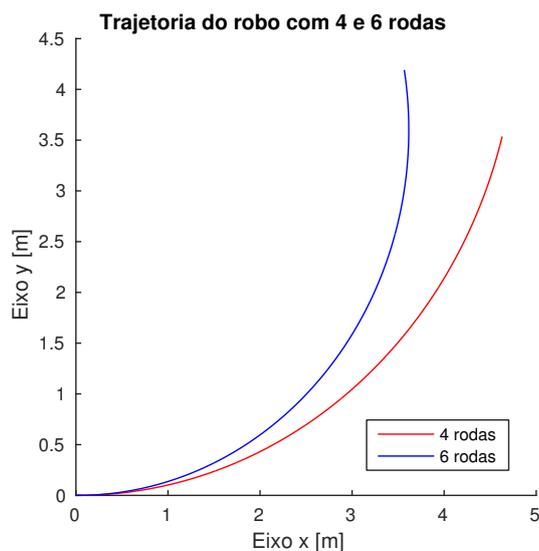


Figura 4.11: Comparação entre as cinemáticas do EspeleoRobô com 4 e 6 rodas para as mesmas velocidades.

dispositivo, o que gera maior consumo energético. As rodas, por sua vez, mantêm mais constantes estas energias, devendo em maior parte apenas repor as perdas por atrito afim de manter o movimento (SIEGWART *et al.*,2011).

Durante condições normais de locomoção, as rodas mantêm contato constante com o terreno, enquanto pernas alternam entre diferentes pontos de apoio sem tocar o terreno. Devido a isto, rodas sofrem mais com o atrito do solo, podendo chegar ao ponto de pernas serem mais vantajosas energeticamente (SIEGWART *et al.*,2011).

Por possuírem vantagens e desvantagens distintas, cada modo de locomoção se torna ideal para um tipo de terreno. Tradicionalmente no projeto de um robô, deve-se conhecer antecipadamente o tipo de terreno onde suas missões se darão para assim escolher qual o melhor modo de locomoção e ser utilizado. Devido ao sistema de troca rápida dos mecanismos de locomoção, esta escolha pode ser feita no momento da missão para o EspeleoRobô.

Além disto, esta flexibilidade permite a utilização de não apenas um tipo de mecanismo, mas também de tipos diferentes atuando em conjunto. Estas configurações híbridas permitem extrair o que cada mecanismo possui de vantajoso na transposição de vários tipos de terrenos (EIJI e SEI,1993).

# Capítulo 5

## SIMULAÇÕES

O uso de simuladores é largamente difundido na robótica. Isto se deve ao fato destes *softwares* permitirem que as características de um robô sejam testadas de maneira rápida e barata, sem necessitar da construção física do mesmo (MICHEL,2004).

O simulador utilizado nesta dissertação é o *Virtual Robot Experimentation Platform* (V-REP). Este possibilita que a cinemática e dinâmica dos corpos sejam abstraídas, permitindo ao programador focar especificamente na modelagem e controle do robô. Um de seus pontos mais poderosos está na possibilidade do uso de diversas abordagens para interagir com os objetos em cena, como *scripts* em linguagem LUA, APIs remotas, *plug-ins* e até mesmo o ROS (ROHMER *et al.*,2013).

Afim de representar o EspeleoRobô no V-REP, sua modelagem foi primeiramente realizada no SolidWorks. Não apenas uma representação tridimensional foi implementada, mas também sua descrição física detalhada. A partir deste modelo é possível obter dados como: localização do CG; peso; distribuição de massa e matriz de inércia. Os pesos das peças estimados pelo programa foram 15,5kg para o chassi, 0,354kg para cada perna e 0,368kg para cada roda.

O chassi e os mecanismos de locomoção são incluídos separadamente, sendo unidos através de juntas rotacionais. Estas podem ser configuradas com os mesmos parâmetros dos motores reais. Dados físicos e geométricos extraídos do dispositivo real e do SolidWorks também são inseridos no simulador afim de aumentar a representatividade do mesmo.

A integração do V-REP com o ROS é implementada via interface, permitindo que o simulador interaja diretamente com os tópicos publicados. Os mesmos códigos de comando do protótipo podem então ser usados para acionar o dispositivo virtual. Além de responder aos mesmos comandos via *joystick*, o robô simulado também gera imagens de câmeras parecidas com as fornecidas pelo EspeleoRobô (Figura 5.1).

Esta integração agiliza a programação, uma vez que se torna possível testar os códigos de acionamento antes mesmo de rodá-los nos protótipo. Verifica-se não apenas a presença de falhas, mas também a viabilidade e eficiência dos modos de locomoção implementados.

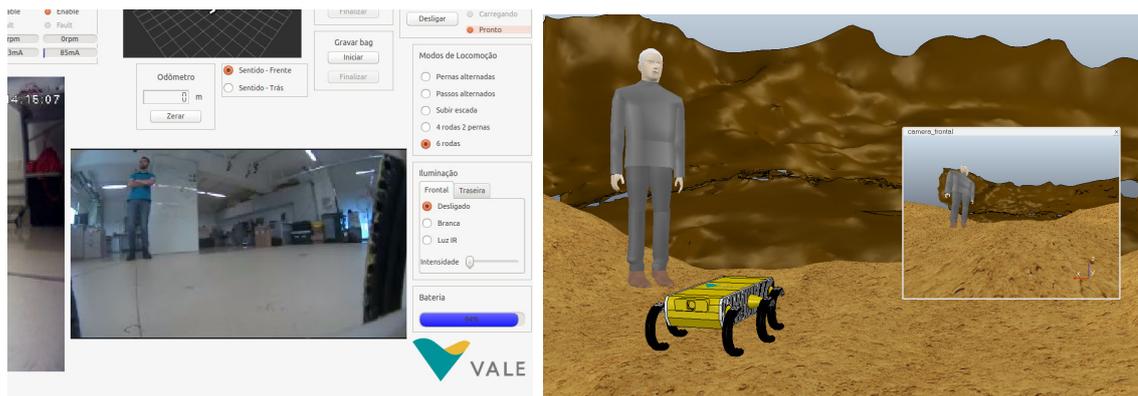


Figura 5.1: Comparação entre as imagens geradas pelo protótipo e pelo dispositivo simulado.

Esta fidelidade permite também a utilização do simulador no treinamento de novos operadores, que podem experimentar antecipadamente o comando do EspeleoRobô em um ambiente virtual. Estes treinamentos podem reduzir as chances de danos ao protótipo por motivo de má condução.

Além do mais, possuindo as juntas rotacionais simuladas os mesmos parâmetros dos motores reais, é possível avaliar o torque aplicado por estes afim de realizar a locomoção com os diversos modos. Isto ajuda no projeto mecânico do robô e na escolha de motores e redutores para o mesmo.

## 5.1 Obtenção de parâmetros em simulação

Inicialmente, códigos em MATLAB foram implementados para interagirem com o simulador V-REP. O objetivo é coletar dados da simulação e calcular as métricas de mobilidade. Apesar de didático, este *software* é proprietário e requer a aquisição de licença para uso. Visando então o uso de código aberto e gratuito, todo o conceito foi portado para Python.

Bibliotecas foram implementadas para de gerar informações acerca dos ensaios realizados. A interação com o V-REP é feita através de sua API remota, viabilizando a coleta dados em tempo de simulação. Estes códigos calculam e plotam as métricas relevantes em tempo de execução, salvando automaticamente os dados após o fim do ensaio.

A topologia de interação entre os *softwares* utilizados pode ser conferida na Figura 5.2. Os códigos das bibliotecas implementadas estão disponíveis no Apêndice A.

A métrica velocidade de translação pode ser diretamente obtida via API. Porém, as métricas de consumo energético e estabilidade devem ser estimadas a partir dos dados fornecidos pelo simulador. Está exposto a seguir como isto é realizado.

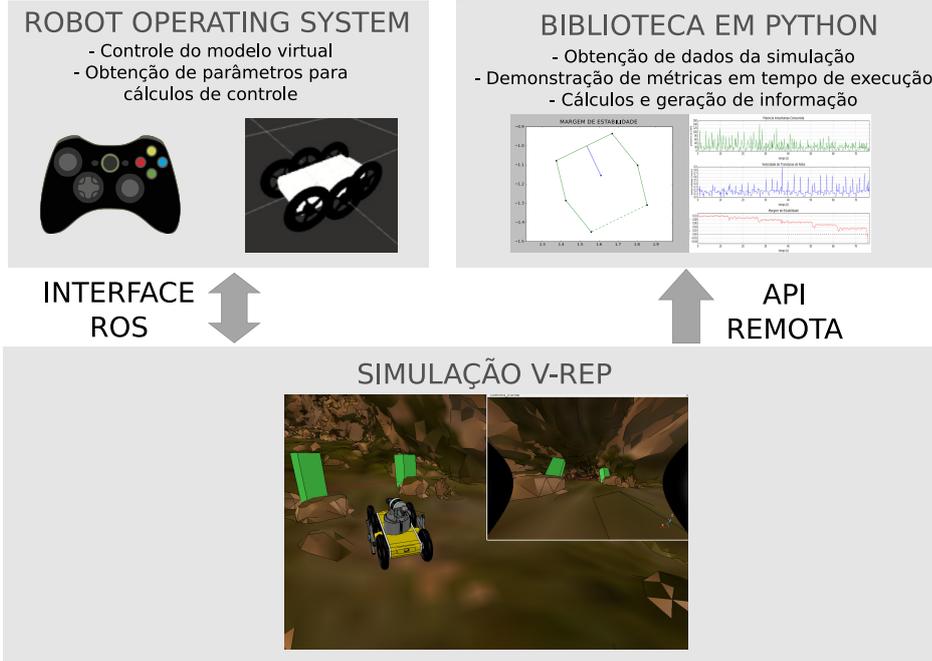


Figura 5.2: Topologia de interação entre os *softwares* utilizados.

### 5.1.1 Métricas de consumo energético

No protótipo, a potência elétrica instantânea é obtida através da tensão aplicada e corrente consumida pelos motores. Além disto, no caso da Resistência Específica  $\varepsilon$ , também devem ser conhecidos o peso do dispositivo e sua velocidade de translação. Destes, o peso é uma constante conhecida e a velocidade de translação é diretamente obtida do simulador.

Não há como obter em simulação as grandezas elétricas afim de encontrar a potência consumida pelos motores. Felizmente, esta também pode ser estimada a partir da velocidade de rotação e do torque aplicado no eixo da junta (RASHID,2015). Estes dois parâmetros são diretamente obteníveis do V-REP.

No protótipo há reduções entre o eixo da junta e o eixo dos motores (descritas na Seção 2.2). Logo, suas proporções de redução e eficiência devem ser levadas em conta para os cálculos.

A velocidade de rotação no eixo do motor  $\omega_m$  pode ser obtida através da velocidade rotacional no eixo da junta  $\omega_j$  multiplicada pela proporção da redução  $k_{pr}$ :

$$\omega_m = \omega_j k_{pr}. \quad (5.1)$$

Para calcular o torque  $j_m$  aplicado no motor a partir do torque  $j_j$  na junta, deve-se levar em consideração a proporção da redução e a eficiência  $k_{red}$  desta:

$$j_m = \frac{j_j}{k_{pr} k_{red}}. \quad (5.2)$$

Calculado  $\omega_m$  e  $j_m$  para cada motor, a potência elétrica instantânea  $S$  consumida por

todo o robô se dá por:

$$S = \sum_{n=1}^Q \left( \frac{j_{mn}}{k_t} \frac{\omega_{mn}}{k_v} \right), \quad (5.3)$$

onde  $Q$  é a quantidade de juntas ativas do robô. O valor destas constantes para o EspeleoRobô podem ser vistas na Tabela 5.1.

A Resistência Específica  $\varepsilon$  pode então ser obtida através da Equação 3.1, considerando que a massa do dispositivo e a aceleração da gravidade são constantes conhecidas e que  $v$  é disponibilizada.

Tabela 5.1: Parâmetros dos motores e reduções utilizados no protótipo do EspeleoRobô.

Parâmetro	Símbolo	Valor	Unidade
constante de torque do motor	$k_t$	$24,3 \times 10^{-3}$	Nm/A
constante de velocidade do motor	$k_v$	41	rad.V/s
proporção das reduções	$k_{pr}$	66 : 1	-
eficiência total das reduções	$k_{red}$	75	%

### 5.1.2 Métricas de estabilidade

Para os cálculos da Margem de Estabilidade Estática (MEES) e do Ângulo de Tombamento (AT), é necessário obter a posição  $F$  dos pontos de apoio do dispositivo com respeito a um sistema de coordenadas inercial arbitrário. Para a Margem de Estabilidade Energética (MEN), além de  $F$ , também é necessário conhecer o peso do robô, que é uma constante conhecida.

É possível coletar do simulador a pose (posição + orientação) do sistema de referência do robô com respeito ao sistema inercial e a posição angular de suas juntas. O sistema inercial é fixado na origem da cena do V-REP.

A posição do robô com respeito ao sistema inercial ( $p_{mr}^{\vec{}}$ ) é obtida do simulador em coordenadas cartesianas tridimensionais. A orientação ( $\Phi_{mr}^{\vec{}}$ ) é em ângulos de Euler-XYZ, sendo representadas respectivamente como:

$$p_{mr}^{\vec{}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (5.4)$$

$$\Phi_{mr}^{\vec{}} = \begin{bmatrix} \alpha_x \\ \beta_y \\ \gamma_z \end{bmatrix}. \quad (5.5)$$

Conhecendo-se as características construtivas do dispositivo e a posição angular de

suas juntas, é possível encontrar os vetores de translação  $p_{ra}^{\vec{n}}$  que relaciona a posição de um ponto de apoio  $F_n$  qualquer com respeito ao sistema de coordenadas do robô.

Com estas informações,  $F_n$  pode ser referenciado com respeito ao sistema inercial da seguinte maneira:

$$F_n = R_{mr}(\Upsilon)p_{mr}^{\vec{n}} + p_{ra}^{\vec{n}}, \quad (5.6)$$

onde  $R_{mr}(\Upsilon)$  é a matriz de rotação do robô com respeito ao sistema inercial.

Esta matriz pode ser calculada multiplicando-se as matrizes de rotação elementares em torno dos eixos  $X$ ,  $Y$  e  $Z$  do sistema inercial com as rotações de Euler obtidas:

$$R_{mr}(\Upsilon) = R_x(\alpha_x)R_y(\beta_y)R_z(\gamma_z). \quad (5.7)$$

Conhecendo-se então as posições dos vários  $F_n$  com respeito ao sistema inercial, basta aplicar os métodos descritos na Seção 3.2 para encontrar os valores de MEES, AT e MEN.

## 5.2 Ensaios realizados

Apesar de haver um protótipo do EspeleoRobô, o simulador é usado exclusivamente para realizar a análise de mobilidade proposta pela dissertação devido aos seguintes motivos: a) é possível configurar maiores torques nas juntas, de maneira que todos os modos possam operar corretamente; b) testar mecanismos que ainda não estão disponíveis, como a roda tipo “estrela”; c) o protótipo não possui os sensores necessário afim de calcular todos os parâmetros das métricas de mobilidade visadas.

Afim de facilitar a discussão que se segue, cada modo de locomoção é associado a uma sigla. Estas podem ser consultadas na Tabela 5.2. Atrélado às siglas podem aparecer sufixos, indicando variações do modo ou condição específica de ensaio. Estão ilustrados na Figura 5.3 os modos de locomoção que serão analisados e comparados, bem como a variação de um deles com a torre de mapeamento acoplada.

Devido ao baixo torque disponível pelos motores no protótipo (Seção 2.1), o correto funcionamento de certos modos de locomoção é comprometido, principalmente os que utilizam pernas. Como o objetivo desta dissertação é avaliar os isoladamente os modos, o torque máximo disponível nos ensaios simulados é arbitrariamente superdimensionado para 30N.m afim de evitar que uma eventual falta de torque comprometesse os resultados.

Para igualizar as condições de ensaio, a velocidade máxima que pode ser desenvolvida pelo eixo do mecanismo é de 5rad/s. A condução do robô durante os ensaios busca levar o dispositivo à maior velocidade de translação possível pelo modo de locomoção.

Visando as métricas de estabilidade, é necessário conhecer quais mecanismos de locomoção estão tocando o chão e qual é exatamente sua posição com respeito ao  $CG$  do robô. Para viabilizar os ensaios, é feita a simplificação de que cada mecanismo toca o solo

Tabela 5.2: Modos de locomoção utilizados e suas siglas.

Modo de Locomoção	Sigla
6 rodas	6R
6 pernas (hexápode) com marcha tripé	6P
4 rodas 2 pernas	4R2P
4 rodas 2 “estrela”	4R2E
6 “estrela”	6E

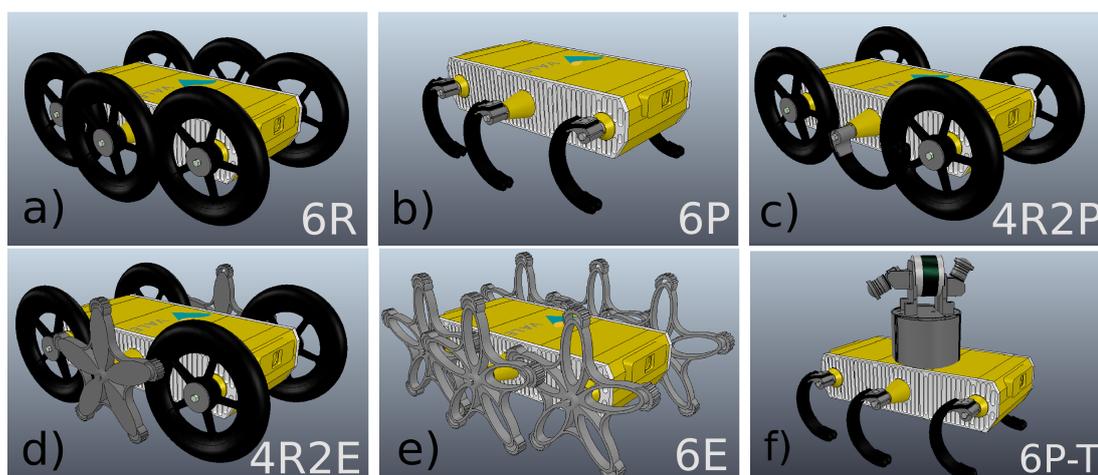


Figura 5.3: Modos de locomoção analisados nos ensaios de simulação. a) 6 rodas; b) 6 pernas; c) 4 rodas e 2 pernas (híbrido); d) 4 rodas e 2 “estrelas”; e) 6 “estrelas” f) exemplo de um modo de locomoção com a torre de mapeamento acoplada.

em um ponto diretamente abaixo do seu eixo de encaixe. Especificamente no modo 6P, a Borda de Suporte é considerada como um triângulo devido à marcha tripé utilizada.

A seguir estão expostos os ensaios realizados em simulação e seus resultados.

### 5.2.1 Estabilidade

Este ensaio tem como objetivo mensurar a estabilidade de modo de locomoção, desde a situação mais estável até o capotamento. O teste consiste no deslocamento do dispositivo sobre uma plataforma de 80cm de altura que, a cada 2m, tem apenas o piso sob o lado esquerdo decrescido em 5cm (Figura 5.4). Eventualmente, à medida que percorre a plataforma, o ângulo de inclinação lateral do robô aumenta até alcançar o ponto de tombamento. Para todos os modos de locomoção, são realizados ensaios com e sem a torre de mapeamento acoplada. O objetivo é também mensurar a influência da adição de uma carga à estabilidade do conjunto.

O modo 4R2P foi testado utilizando como apoio apenas suas quatro rodas. As pernas não auxiliaram em momento algum na locomoção, já que seu objetivo é apenas auxiliar na transposição de obstáculos.

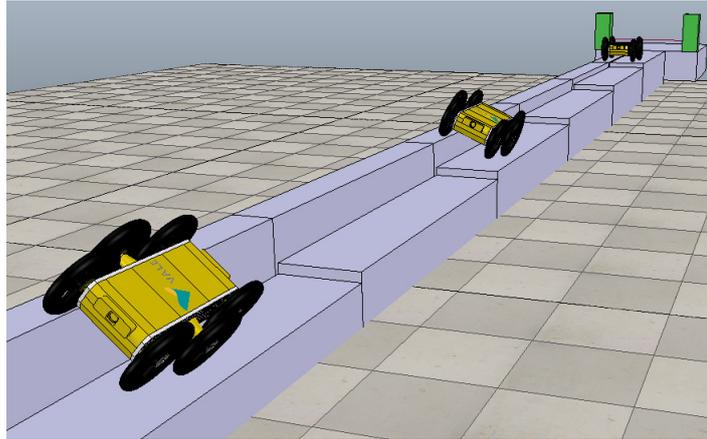


Figura 5.4: Plataforma utilizada para realizar os ensaios de estabilidade.

São calculados em tempo de execução as métricas de estabilidade Ângulo de Tombamento e Margem de Estabilidade Energética. Os resultados obtidos podem ser consultados nas Figuras 5.5 e 5.6.

Ao início do percurso, o robô se encontra sobre um plano sem inclinação, sendo esta a circunstância de maior estabilidade. Nesta situação, os modos 6R, 6E e 4R2E são os mais estáveis, sendo seguidos pelo 4R2P e, finalmente, o 6P. Isto é explicado pela maior área de PS gerado pelos modos que obtêm os melhores resultados (Figura 5.7).

Pela natureza do ensaio estar relacionada a descer uma escada, as métricas de estabilidade decrescem discretamente à medida que o declínio entre as laterais do robô aumenta. O tombamento do robô ocorre quando o limiar zero é atingido.

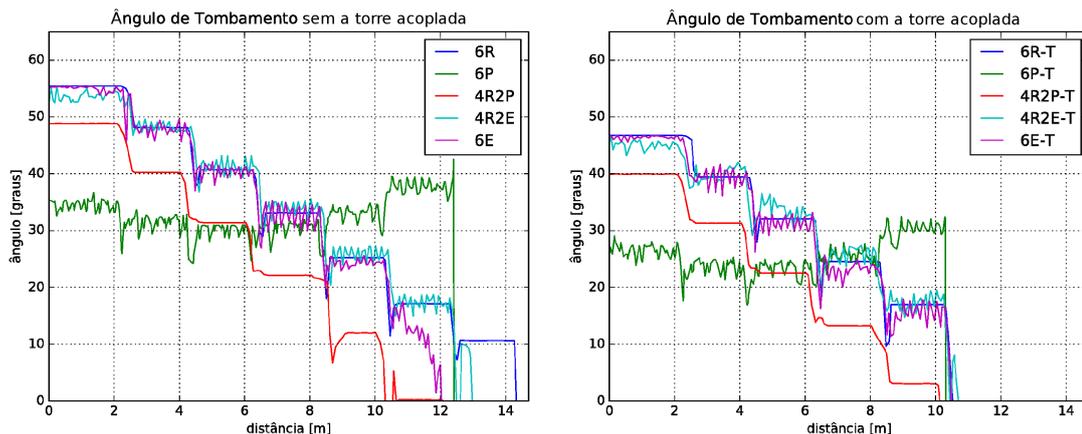


Figura 5.5: Ângulo de Tombamento para os modos de locomoção sem (à esquerda) e com a torre de mapeamento (à direita).

Os modos que utilizam rodas como único tipo de mecanismo, como o 6R e 4R2P, desenvolvem uma locomoção mais suave devido ao tipo de interação deste mecanismo de locomoção com o terreno. Este comportamento se reflete nas curvas de seus gráficos de estabilidade. Por sua vez, os modos que possuem rodas tipo “estrela” descrevem

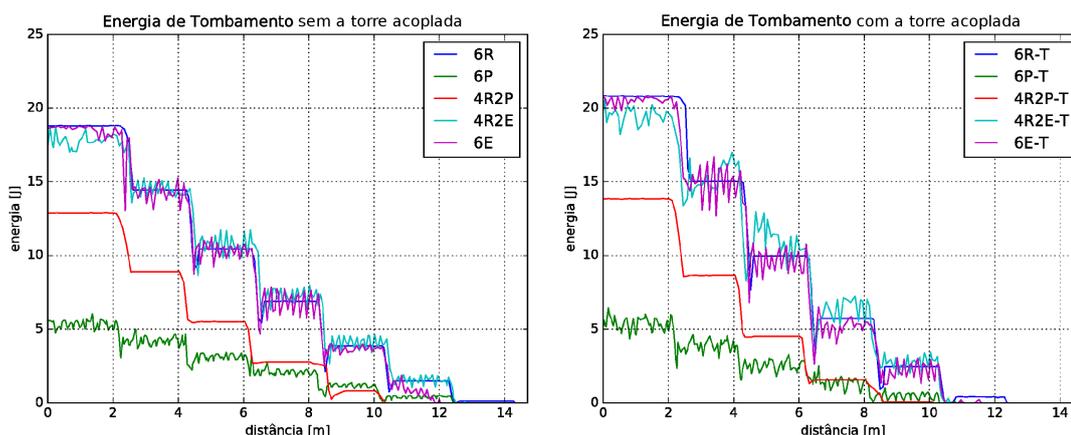


Figura 5.6: Margem de Estabilidade Energética para os modos de locomoção sem (à esquerda) e com a torre de mapeamento (à direita).

curvas similares, porem apresentando vibrações introduzidas por este tipo de mecanismo de locomoção.

O modo 6P também apresenta trepidações, mesmo quando se locomove sobre um plano nivelado. Entretanto, diferentemente dos demais modos de locomoção, nota-se que o AT não decresce discretamente à medida que os degraus são passados. Este se mantém a um nível relativamente constante e, abruptamente, despenca para zero quando ocorre o capotamento.

Isto é um comportamento não desejado, já que se torna difícil prever através do AT quando o robô se encontra em situação iminente de tombar. A MEN, por sua vez, é sensível ao aumento da inclinação do corpo, indicando este através de valores discretamente decrescentes ao longo do ensaio.

Quando acoplada ao robô, a torre de mapeamento impacta na estabilidade de todos os modos de locomoção. O Ângulo de Tombamento se torna em média menor para uma mesma inclinação devido ao aumento da altura do centro de gravidade do conjunto, mesmo quando nivelado. Isto pode ser confirmado pelo capotamento precoce dos modos de locomoção quando estão com a torre acoplada se comparado aos resultados sem a mesma.

Por sua vez, a MEN no geral aumenta. Isto ocorre devido à maior massa do conjunto quando comparada a do dispositivo isolado.

Na prática, isto indica que quando a torre está acoplada, o dispositivo tende a capotar com menores ângulos de inclinação, entretanto se torna mais trabalhoso levar o robô até a situação de instabilidade; há assim uma espécie de compensação.

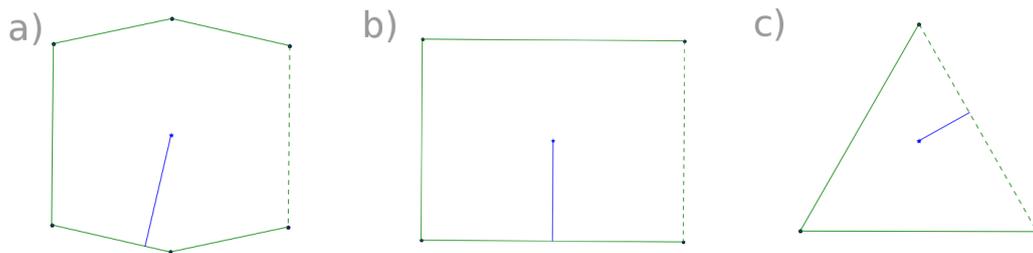


Figura 5.7: Os três tipos de PS apresentados pelos modos: a) 6R; b) 4R2P; c) 6R, 6E e 6R2E. Em azul, a MEES na situação de maior estabilidade.

## 5.2.2 Máxima altura de transposição

Este ensaio visa descobrir a altura máxima transponível por cada modo de locomoção. É requerido ao robô subir uma escada (Figura 5.8) cuja altura dos degraus cresce gradativamente a uma taxa de 1,125cm por degrau.

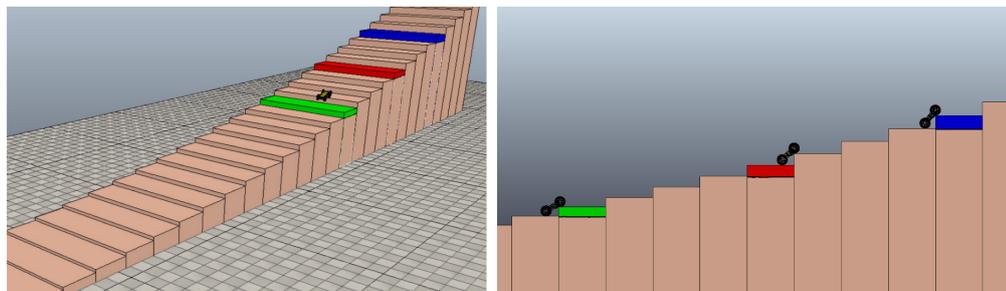


Figura 5.8: Escadaria utilizada no ensaio de transposição de obstáculos.

Obviamente que degraus são obstáculos bem estruturados, diferindo muito dos que são comumente encontrados pelo dispositivo em suas missões rotineiras. Entretanto, a metodologia deste ensaio permite analisar a potencialidade de transposição de cada modo de locomoção. É possível também constatar a função de cada mecanismo no momento da subida, bem como os efeitos que a torre de mapeamento causa nesta habilidade do conjunto.

O modo 6P desenvolve neste ensaio, assim como em todos os outros, a marcha tripé com *Buehler's Clock*. Existem marchas próprias para subir escadas, como por exemplo a demonstrada em MOORE *et al.* (2002). Entretanto, ensaios preliminares mostram que esta marcha não se comporta bem em escadas com degraus de altura variável e largura maior que o comprimento do robô, como é o caso da escadaria do ensaio. Devido a isto, marchas específicas na subida de escadarias não foram utilizadas nesta etapa.

Dois fatores importantes notados durante as transposições são: o quanto os mecanismos de locomoção fazem o robô vibrar; a inclinação que o dispositivo apresenta no momento da subida. Estes dois fatores geralmente causam o capotamento do robô para trás.

Os resultados obtidos neste ensaio estão discriminados na Tabela 5.3. Considerando

Tabela 5.3: Resultado da altura máxima transposta por cada modo de locomoção em ordem decrescente de altura superada.

Modo	Degrau atingido	Altura transposta [cm]
4R2P	25	31,25
6P	21	26,25
4R2E	19	23,75
6P-T	19	23,75
6R	17	21,25
6R-T	15	18,75
6E	15	18,75
6E-T	12	15
4R2E-T	11	13,75
4R2P-T	11	13,75

os modos sem a torre, o que obteve o melhor resultado foi do 4R2P, 31,25cm; o pior resultado veio do modo 6E, com uma altura máxima transposta de 18,75cm.

Sobre o modo 4R2P, a utilização apenas das quatro rodas proporciona baixa capacidade de transposição; porém, enquanto bloqueado em um degrau, as pernas laterais podem ser acionadas afim de completar a subida (Figura 5.9). As pernas funcionam então como “ganchos”, que se ancoram no andar superior e puxam o corpo para cima.

Esta estratégia é a mais eficiente neste ensaio, fazendo o 4R2P apresentar os melhores resultados quando não há carga. Entretanto, este também é o modo mais suscetível à adição da torre de mapeamento, apresentando queda de 56% em seu desempenho. Com a carga acoplada, há uma desestabilização do conjunto no momento em que as pernas são acionadas, o que leva a um prematuro capotamento.

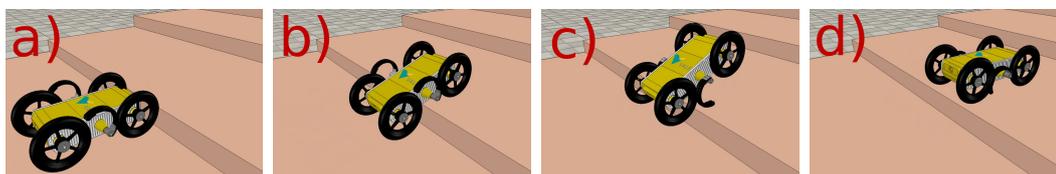


Figura 5.9: Transposição de um obstáculo com o modo 4R2P: a) início do movimento apenas com rodas; b) dispositivo travado no obstáculo; c) utilização das pernas para sair do bloqueio; d) obstáculo transposto.

O segundo modo mais sensível à adição de carga é o 4R2E, com perda de 42%. Os modos que menos sofrem com a adição da carga são o 6P e o 6R, sofrendo perdas de apenas 9,52% e 11,76%, respectivamente.

O modo 6R conseguiu ultrapassar obstáculos com altura maior que o raio de suas rodas, cujo valor é de 14cm. Isto se deve ao trabalho em conjunto realizado por todos estes

mecanismos. Enquanto as rodas dianteiras escalam o obstáculo, as traseiras proporcionam tração para apoiar este movimento. Assim que as rodas centrais atingem o topo do degrau, estas puxam todo o corpo do robô para cima, finalizando a transposição.

### 5.2.3 Locomoção em ambiente controlado

O objetivo deste ensaio é analisar o comportamento das métricas de consumo energético, estabilidade e velocidade de translação, durante a locomoção em ambiente controlado. O dispositivo deve realizar um percurso em linha reta sobre uma plataforma de 20m de comprimento.

Duas variações do mesmo ensaio são realizadas, sendo uma em pista livre e outra com pequenos obstáculos de 120mm de altura dispostos à distâncias arbitrárias entre si (Figura 5.10). Com isto, pode-se analisar o impacto de pequenos obstáculos no comportamento dos modos de locomoção.

Além disto, os modos de locomoção são ensaiados com e sem a torre de mapeamento acoplada em cada variação. Isto permite avaliar também a influência da adição de carga em locomoções desta natureza.

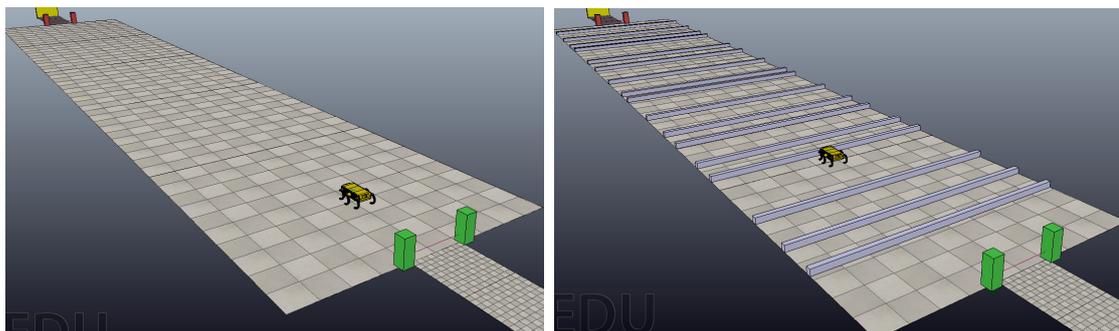


Figura 5.10: Plataformas utilizadas no ensaio. À esquerda, a plataforma livre e, à direita, trajeto com obstáculos.

As métricas são calculadas para cada instante de tempo do ensaio e apresentados em *boxplot* na Figura 5.11. Este formato é escolhido pois permite representar diferentes informações estatísticas em um mesmo gráfico, como mediana, circunscrição dos primeiro e terceiro quartis e os limites inferior e superior dos dados obtidos (WILLIAMSON *et al.*,1989).

Para a apresentação dos resultados, os modos de locomoção recebem os sufixos “-T” quando o teste é realizado com a torre de mapeamento e “-O” quando se trata do ensaio com obstáculos. A combinação destes dois também é realizada.

#### Análise da eficiência energética

Analisando a eficiência energética através da Resistência Específica  $\varepsilon$ , nota-se que os modos com rodas são os mais eficientes. Além do mais, a adição da carga não influencia

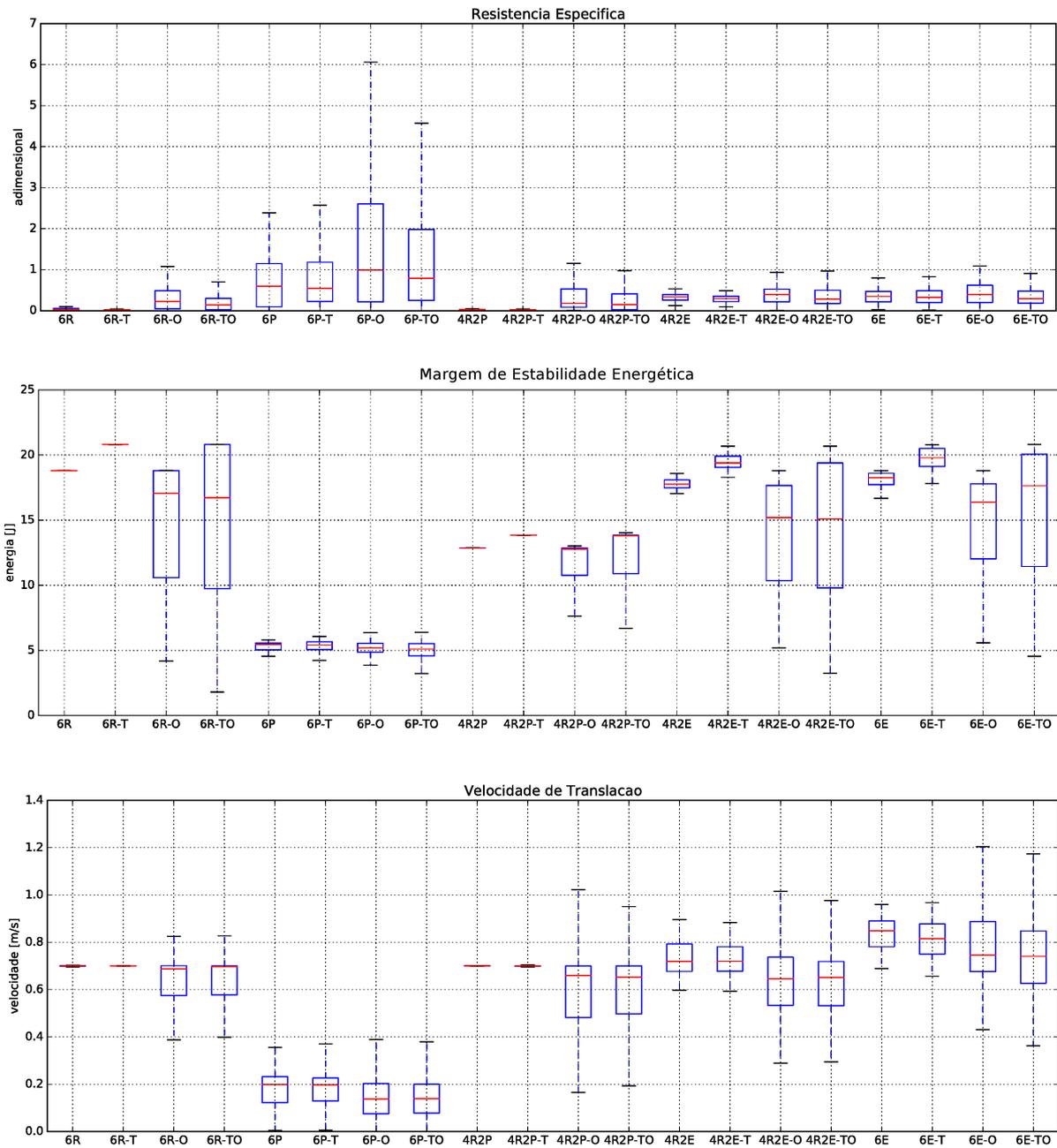


Figura 5.11: Resultados referentes aos ensaios em ambiente controlado.

consideravelmente neste parâmetro. Quando os obstáculos são adicionados, porém, há uma pequena queda na eficiência. Aliás, este comportamento é constatado na maioria dos modos de locomoção.

O modo 6P apresenta os piores resultados para esta métrica. Além de obter a pior mediana, a variação em seu desempenho energético é maior do que a dos demais.

### **Análise da estabilidade**

Com respeito à estabilidade, o modo 6R é o melhor em terreno completamente plano. Sua Margem de Estabilidade Energética (MEN) se mantém relativamente alta e constante durante o percurso. Os modos 4R2E e 6E demonstram desempenho similar ao 6R, possuindo porém maior variação de estabilidade.

No geral, a MEN dos modos aumenta quando a carga é adicionada e o dispositivo se encontra no plano. Isto se deve ao aumento da massa do conjunto, compensando o aumento da altura de *CG*. Quando percorrem porém o trajeto com obstáculos, esta métrica sofre uma queda e aumenta sua variação.

O modo 6P apresenta resultado distinto. Apesar de ser o menos estável sob a ótica da MEN, não há considerável mudança deste parâmetro quando é adicionada a torre ou os obstáculos são enfrentados.

### **Análise da velocidade de translação**

O modo 6E é o mais rápido, apresentando a mediana da métrica de Velocidade de Translação próxima dos 0,85m/s. Este resultado é 4,25x melhor que o do último colocado, o modo 6P.

À parte o 6P, os modos de locomoção apresentam velocidades de translação relativamente próximas. Nota-se também que a adição da carga não altera sensivelmente este parâmetro. Por sua vez, a presença de obstáculos gera sensível queda no resultado desta métrica.

O modo 6R e 4R2P possuem medianas próximas e não variam muito o desempenho quando a carga é considerada. No percurso com obstáculos porém, o modo 4R2P sofre queda de velocidade mediana 2,5x maior que o 6R. Isto acontece pois ambos se locomovem da mesma maneira no plano, porém seus métodos de negociação de obstáculos se diferem.

## **5.2.4 Locomoção em ambiente representativo**

Este ensaio pode ser visto como uma extensão do anterior (Seção 5.2.3). Os mesmos parâmetros dos modos de locomoção são analisados, o teste porém é realizado em ambiente representativo, similar ao que será enfrentado em missões reais.

O modelo de caverna utilizado se baseia na N4E26, uma paleotoca localizada na Serra dos Carajás - PA (6°02'32" S - 50°09'35" W). Devido às suas características topográficas e à

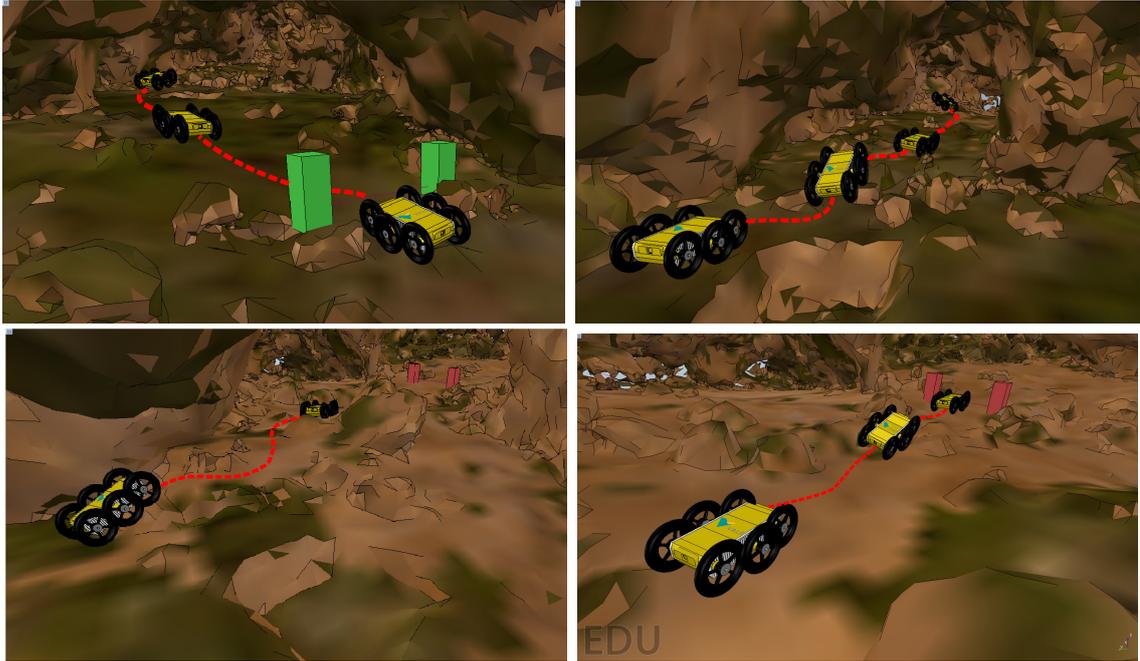


Figura 5.12: Detalhe do trajeto que cada modo de locomoção tem que realizar dentro do modelo tridimensional da N4E26, desde o ponto de partida até o ponto de chegada.

presença de espécimes singulares em seu interior, esta é considerada uma cavidade de alta relevância e é então preservada em sua totalidade. Devido à existência de extração mineral em suas proximidades, um monitoramento constante de sua estrutura é necessário, sendo este então um ambiente típico onde os serviços do EspeleoRobô serão requisitados.

Por ser uma das cavernas mais bem catalogadas no mundo, o interior da N4E26 foi mapeada tridimensionalmente. Este mapa foi tratado e inserido no V-REP. As características do solo foram configuradas com padrões pré-estabelecidos pelo simulador de terrenos com baixo atrito. Testar então o modelo do EspeleoRobô neste ambiente permite que seu comportamento em situações bem próximas às reais sejam verificadas.

O ensaio consiste na realização de um trajeto preestabelecido na cavidade. O percurso (Figura 5.12) possui cerca de 50m e o tipo de terreno é uma mescla entre trechos estruturados e partes acidentadas, inclusive com a presença de obstáculos consideráveis. Um marco verde indica o ponto inicial e um vermelho, o final.

Novamente, cada modo é testado sem e com a torre de mapeamento fixada. São mensurados o tempo de realização do trajeto, a potência elétrica ( $S$ ), Resistência Específica ( $\varepsilon$ ), Margem de Estabilidade Energética (MEN), Ângulo de Tombamento (AT) e a velocidade de translação do dispositivo ( $v$ ).

Vários ensaios foram realizados para cada modo, sendo registrado para fins de análise o de menor tempo. A seguir estão discriminados os resultados obtidos.

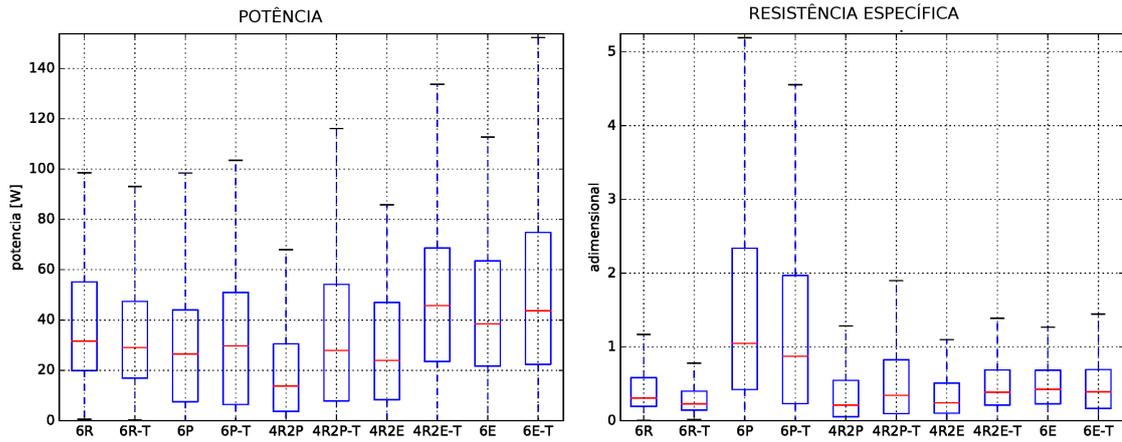


Figura 5.13: Resultados da potência consumida e resistência específica para o ensaio do trajeto em ambientes representativos.

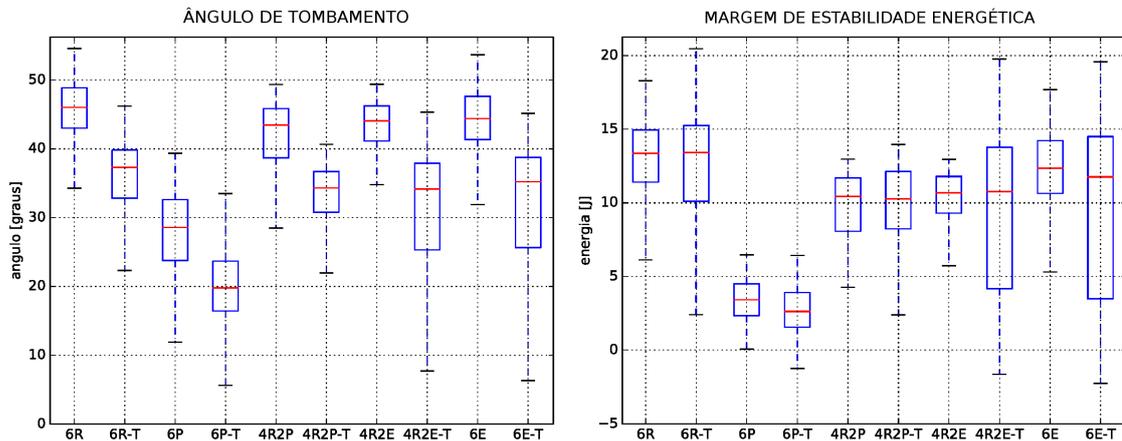


Figura 5.14: Resultados do Ângulo de Tombamento e da Margem de Estabilidade Energética para o ensaio do trajeto em ambientes representativos.

### Análise da eficiência energética

As métricas energéticas obtidas podem ser visualizadas na Figura 5.13. Tem-se à esquerda os valores da Potência Instantânea desenvolvida ( $S$ ) e à direita da Resistência Específica ( $\varepsilon$ ).

Nota-se que alguns modos obtém bons resultados em uma métrica, mas ficam aquém na outra. Isto ocorre pois  $S$  avalia apenas a potência instantânea, enquanto  $\varepsilon$  (Equação 3.1) também considera o peso e a velocidade de translação do dispositivo.

Como exemplo, considerando apenas  $S$ , os modos com rodas tipo estrela possuem maior consumo. Avaliando porém  $\varepsilon$ , estes ficam no mesmo nível que os demais. O contrário ocorre com o modo 6P, que possui menor eficiência energética se avaliado através da Resistência Específica.

O modo 4R2P apresenta os melhores resultados para ambas as métricas. Isto muito se

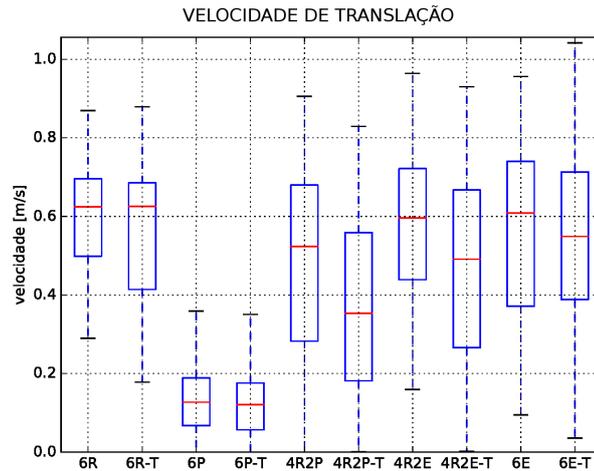


Figura 5.15: Resultados da velocidade de translação para o ensaio do trajeto em ambientes representativos.

deve à utilização de apenas quatro rodas para realizar a locomoção, sendo que as pernas laterais são acionadas apenas nas transposições de obstáculos.

### Análise da estabilidade

Os resultados para as métricas de estabilidade estão mostrados na Figura 5.14.

Considerando o Ângulo de Tombamento (AT), os resultados são similares para todos os modos de locomoção, salvo o 6P. Avaliando porém a Análise da Margem de Estabilidade (MEES), nota-se que o 6R é ligeiramente mais estável que os demais.

A métrica AT é no geral fortemente afetada pela adição da torre de mapeamento. Já a MEES não altera muito a mediana; o impacto maior está em sua variação. Novamente, isto se explica pela maior estabilidade do conjunto devido ao aumento do peso, mesmo que o CG tenha aumentado de altura.

### Análise da velocidade de translação

A velocidade desenvolvida por cada modo está demonstrada na Figura 5.15.

Em ambiente representativo, o modo 6R é um pouco mais rápido que os demais. Este é cerca de 4,5x mais rápido que o modo mais lento, o 6P.

Estes dois também são os modos que menos sofrem com a adição de carga. Suas medianas e variações não fortemente impactadas nesta situação. O modo 4R2P, por sua vez, é o que mais perde em desempenho com a presença da torre, com uma queda de desempenho de 33%.

## 5.3 Discussões

Esta seção faz uma síntese dos resultados obtidos nas simulações realizadas. O objetivo é realizar uma análise comparativa dos modos de locomoção de forma quantitativa baseada nos dados obtidos dos ensaios apresentados neste capítulo.

Os resultados das métricas são normalizados afim de serem inseridos em um mesmo gráfico. Os índices normalizados obtidos variam então de zero a um sendo que, quanto maior seu valor, melhor é o desempenho do modo naquele respectivo parâmetro. A simbologia das métricas normalizadas pode ser consultada na Tabela 5.4.

Tabela 5.4: As métricas de mobilidade e seus símbolos.

Métrica	Métrica Normalizada
Resistência Específica	$\hat{\epsilon}$
Potência Consumida	$\hat{S}$
Velocidade de Translação	$\hat{v}$
Margem de Estabilidade	$\widehat{MEES}$
Ângulo de Tombamento	$\widehat{AT}$
Margem de Estabilidade Energética	$\widehat{MEN}$

Deve haver um cuidado na interpretação dos resultados normalizados de maneira que não se confunda métricas de mobilidade naturalmente inversas. Um exemplo é a Potência Instantânea  $S$ ; quanto menor seu valor, melhor é o consumo energético do modo de locomoção. Entretanto, os resultados de sua métrica normalizada são inversos. Quanto mais próximo de um for o valor de  $\hat{S}$ , melhor é seu resultado.

São apresentadas análises referentes a duas situações, sendo a primeira com o dispositivo sem carga alguma, denominado como “modo base”, e a segunda com a torre de mapeamento acoplada, nomeado “modo com a torre”. Tendo em vista que as demais condições de ensaio são exatamente as mesmas, torna-se possível analisar o impacto gerado pela adição de carga em cada modo de locomoção.

### 5.3.1 Comparação genérica dos modos

Esta primeira comparação avalia o comportamento dos modos de locomoção em ambiente estruturado. Os resultados podem ser analisados na Figura 5.16.

Basicamente, são analisados o consumo energético (através de  $\epsilon$ ), a estabilidade (através de  $AT$ ) e a velocidade de translação ( $v$ ) dos modos em uma movimentação frontal sobre um terreno plano sem obstáculos. Esta é a situação onde os modos apresentam seus melhores desempenhos para estes parâmetros. Por sua vez, o quesito altura de obstáculo transponível provém dos resultados obtidos no ensaio da Seção 5.2.2.

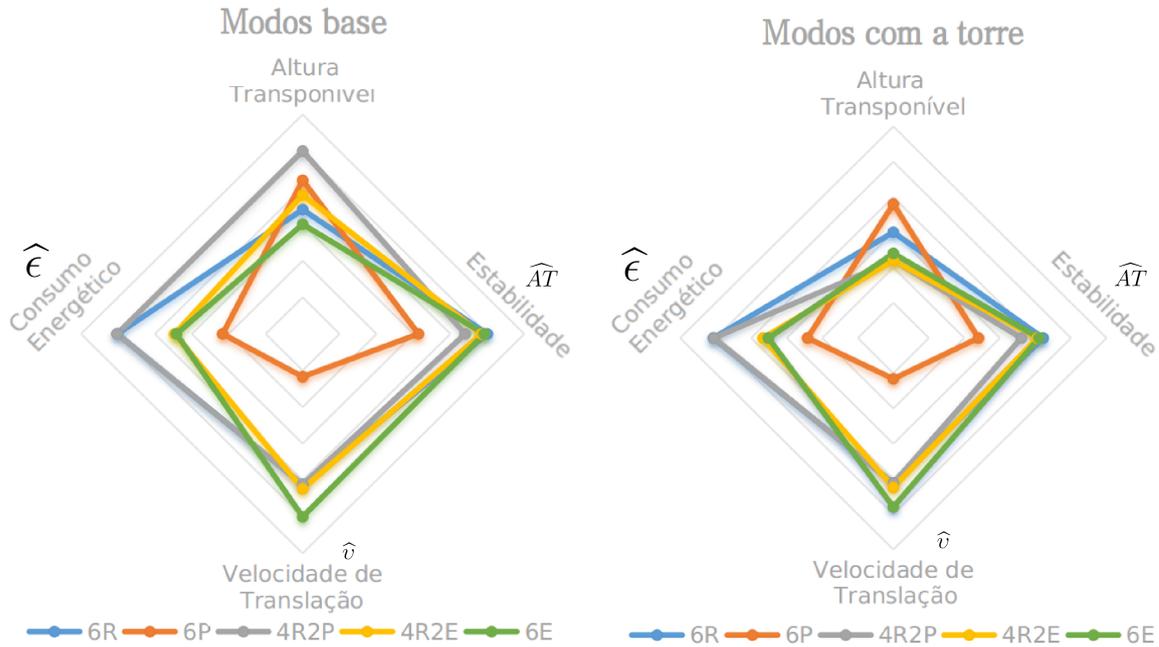


Figura 5.16: Comparação dos modos de locomoção em situação genérica através dos índices referentes às métricas de mobilidade.

Analisando primeiramente a altura transponível, o modo 4R2P obteve o melhor resultado. Como visto, esta realização é graças a utilização das pernas laterais em auxílio durante a transposição. Entretanto, a nota do modo cai consideravelmente ao se adicionar a carga: cerca de 56%.

O modo 6P aparece em segunda posição, porém não sofre tanto com a adição de *payload*, sendo o melhor modo em questão de altura transponível quando a torre esta acoplada.

No quesito eficiência energética, os modos com rodas (6R e 4R2P) são os melhores. Em segunda posição aparecem os modos com rodas tipo “estrela” (6E e 4R2E) e, com a menor eficiência energética, aparece o modo 6P.

Cabe lembrar que a métrica Resistência Específica ( $\varepsilon$ ) considera a velocidade de translação no cálculo da eficiência energética. O resultado ruim do modo 6P para este parâmetro se deve em grande parte à sua baixa velocidade de translação desenvolvida, como pode ser visto neste respectivo atributo.

Analisando os demais modos em questão de velocidade, o 6E se mostra o mais rápido. Intermediariamente aparecem os modos 4R2E, 4R2P e 6R. Este quesito não sofre grande impacto relacionado à adição da carga.

Quanto a Estabilidade, novamente devido ao Polígono de Suporte descrito, os modos 6R, 6E e 4R2E ficam na primeira posição. O modo 4R2P fica em segundo e o modo 6P aparece em último. Neste quesito, a adição da carga impacta de maneira similar em todos os modos.

É possível gerar uma nota de desempenho global de cada modo através da área dos

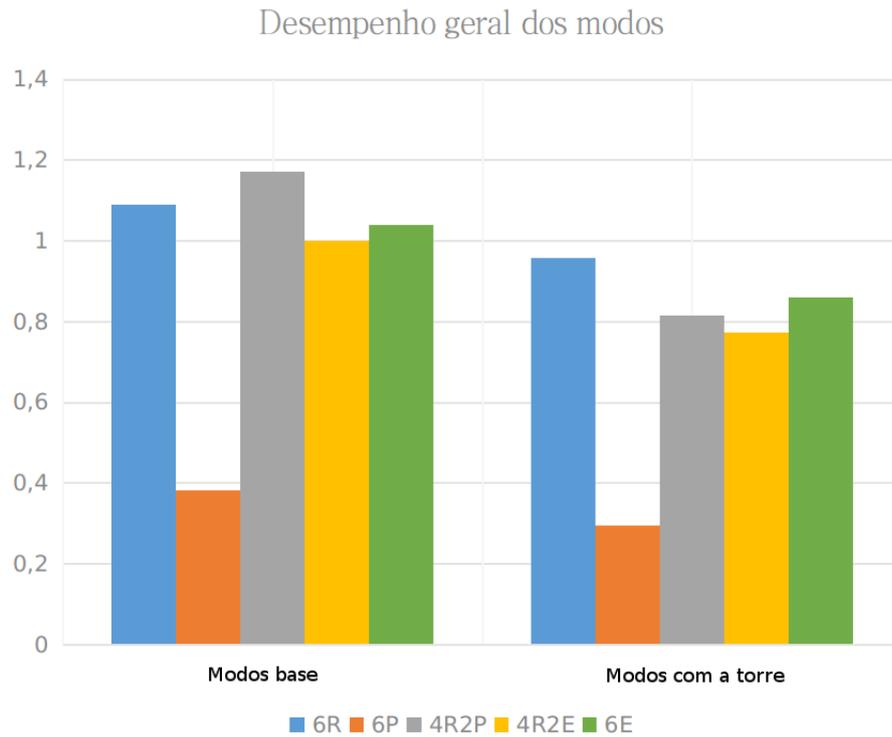


Figura 5.17: Comparação do desempenho geral dos modos de locomoção, sem e com a carga acoplada.

polígonos formados na Figura 5.16. Quanto maior seu valor, melhor o modo é avaliado em uma média geral dos resultados de suas métricas. Estas notas podem ser consultadas na Figura 5.19.

Tabela 5.5: Taxa de queda de desempenho devido a adição de carga no índice geral em situação genérica.

Modo de Locomoção	Taxa de Queda no Desempenho
4R2P	-30%
4R2E	-23%
6P	-22%
6E	-17%
6R	-12%

Em ambiente estruturado, o modo base 4R2P fica em primeiro lugar com nota 1,17. Em último está o modo 6P, obtendo 0,38.

A adição de carga impacta no desempenho de todos os modos. A Tabela 5.5 indica a taxa de queda no desempenho geral dos modos de locomoção quando a torre de mapeamento é acoplada.

O modo 4R2P é o mais afetado, com 30% de queda. Isto reflete no seu índice de desempenho geral, caindo da primeira para a terceira colocação. O modo 6R é o menos afetado, sendo que assume a primeira posição quando a torre está fixada.

### 5.3.2 Comparação dos modos em ambiente representativo

Diferentemente da primeira, a segunda análise visa comparar os modos de locomoção enquanto operando em ambiente representativo. Basicamente, os modos de locomoção são aqui comparados através dos resultados obtidos na Seção 5.2.4. A análise pode ser consultada na Figura 5.18.

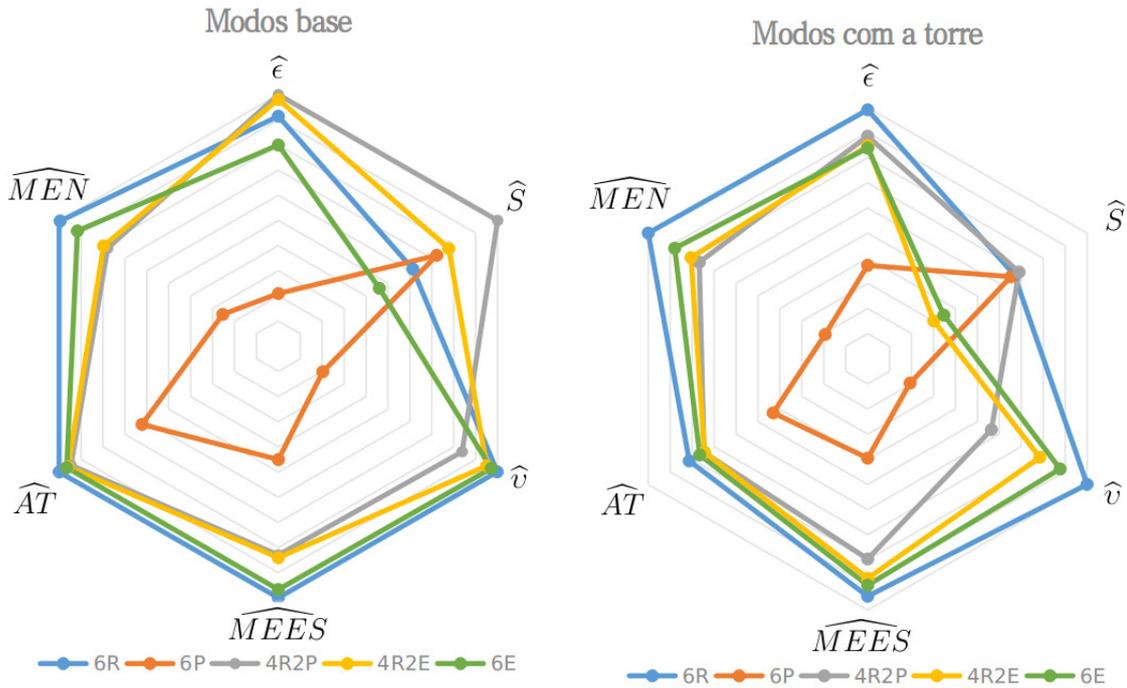


Figura 5.18: Comparação dos modos de locomoção em ambiente representativo através das métricas de mobilidade.

#### Análise da estabilidade

Em questão de estabilidade, os principais índices a serem analisados são  $\widehat{AT}$  e  $\widehat{MEN}$ , já que estes levam em consideração a altura do  $CG$  do robô. Relembrando os conceitos da Seção 3.2, o Ângulo de Tombamento (atrelado a  $\widehat{AT}$ ) representa o quanto o dispositivo pode ser rotacionado até capotar; a Margem de Estabilidade Energética (relacionada  $\widehat{MEN}$ ) quantifica o trabalho que o dispositivo pode sofrer até atingir o capotamento.

Os resultados de  $\widehat{AT}$  foram próximos para a maioria dos modos de locomoção base. O modo 6R possui o melhor índice, sendo acompanhado de perto pelos demais. A exceção fica com o modo 6P, que possui desempenho bem abaixo dos outros neste quesito. A explicação se dá pelo tamanho do PS descrito por este modo de locomoção, como já demonstrado na Figura 5.7.

Quando a torre de mapeamento é adicionada, vê-se que  $\widehat{AT}$  cai para todos os modos; isto se deve pelo aumento da altura do  $CG$ . O valor de  $\widehat{MEN}$  entretanto se mantém

praticamente inalterado, o que é explicado pelo aumento da inércia do conjunto em compensação.

### **Análise do consumo energético**

Em questão de consumo energético, o modo 4R2P possui na comparação dos modos base os melhores resultados, tanto para  $\widehat{S}$  quanto para  $\widehat{\epsilon}$ . Isto acontece pois este modo aciona, em maior parte do tempo, apenas quatro mecanismos para realizar a locomoção; as pernas são acionadas esporadicamente durante a transposição de obstáculos. Ao ser adicionado entretanto a torre de mapeamento, seu índice  $\widehat{\epsilon}$  cai, ficando em segunda posição logo atrás do modo 6R.

Uma análise interessante é a comparação de  $\widehat{S}$ , relacionado à potência instantânea consumida, e  $\widehat{\epsilon}$ , atrelado à Resistência Específica (Seção 3.1). A potência instantânea  $S$  se trata apenas do consumo energético, enquanto a Resistência Específica  $\epsilon$  leva em consideração, além de  $S$ , a massa do dispositivo e a velocidade translacional que este desenvolve.

O modo de locomoção 6P apresenta o valor de  $\widehat{\epsilon}$  bem abaixo de  $\widehat{S}$ . Isto indica que, apesar de possuir consumo de potência próximo dos demais modos, sua eficiência energética fica comprometida devido à menor velocidade translacional desenvolvida.

O modo 6P apresenta o valor de  $\widehat{S}$  maior que o modo 6R. Isto se explica pelo fato do 6R acionar simultaneamente todas as suas seis juntas para realizar a locomoção. Por sua vez, o modo 6P aciona apenas três juntas ao mesmo tempo para realizar a locomoção, enquanto as demais rotacionam livres no ar. Comparando finalmente os modos 6P e 6R através de  $\widehat{\epsilon}$ , o modo 6R se mostra superior em eficiência energética, devido principalmente a sua velocidade em média 3,6x maior.

Os modos de locomoção que utilizam rodas tipo “estrelas” apresentam resultados abaixo da média em questão  $S$ . O modo 6E possui o pior resultado para  $\widehat{S}$ , sendo também acompanhado pelo 4R2E quando a torre é acoplada. Isto é devido à natureza de operação deste tipo de mecanismo que, apesar de manter um movimento rotacional contínuo, requer pequenos picos de torque afim de realizar a passada entre as pontas do mecanismo.

### **Comparação conjunta dos índices**

Na Figura 5.19 está demonstrada uma comparação global das métricas avaliadas, similarmente ao que foi feito na seção anterior.

O modo de locomoção 6R é o melhor avaliado, atingindo a nota 0,79. Os modos 4R2P, 6E e 4R2E estão praticamente empatados em segundo, atingindo em média a nota 0,69. O modo 6P, por sua vez, aparece em último com nota 0,17.

Todos os modos de locomoção sofrem perda de desempenho devido a adição da torre de mapeamento. A Tabela 5.6 indica a taxa de perda apresentada por cada modo separa-

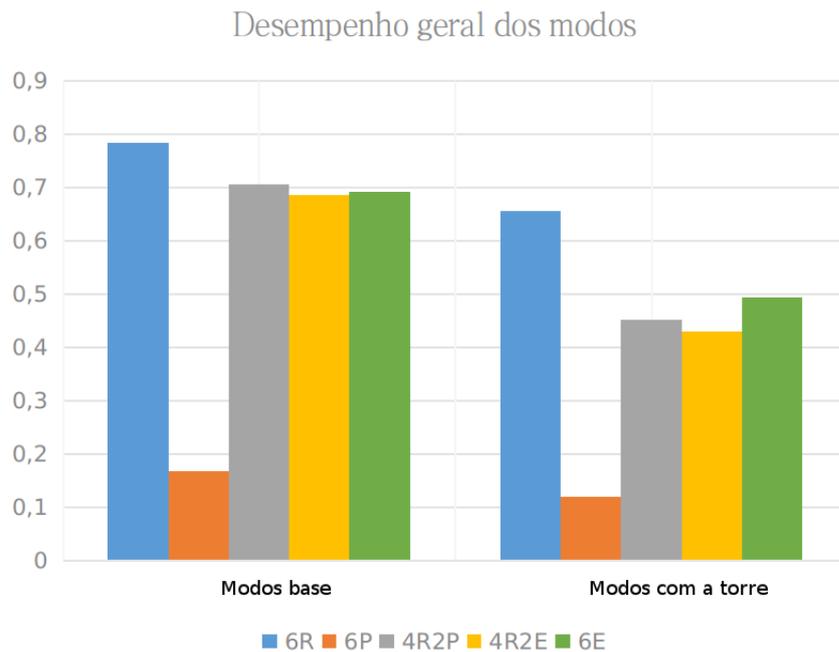


Figura 5.19: Comparação do desempenho geral dos modos de locomoção em um ambiente representativo. Resultados para os modos sem e com a carga acoplada.

damente. O modo 4R2E aparece em primeiro, com uma queda de 37%, enquanto o modo 6R é o que menos perde em desempenho com a adição de *payload*, sendo esta de apenas 16%.

O modo base 4R2P aparece em segundo lugar, ligeiramente melhor que o modo 6E. Ao se adicionar a carga, entretanto, este cai para terceiro lugar com resultado próximo ao do modo 4R2E.

Tabela 5.6: Taxa de queda de desempenho devido a adição de carga no índice geral em ambientes representativos.

Modo de Locomoção	Taxa de Queda no Desempenho
4R2E	-37%
4R2P	-36%
6P	-29%
6E	-29%
6R	-16%

# Capítulo 6

## TESTES DE CAMPO

Os modos de locomoção, bem como outros aspectos operacionais do EspeleoRobô, foram analisados também em testes de campo.

Apesar de ter sido concebido com o intuito de explorar cavidades naturais, o robô se mostra muito versátil em explorar diversos outros tipos de ambientes. Missões também foram realizadas em minas subterrâneas, áreas de risco, galerias de vertedouro de barragens, drenos de estradas, entre outros.



Figura 6.1: Primeiros testes realizados com o EspeleoRobô, sendo o teste na Minas da Passagem em Mariana-MG (à esquerda) e no campus da UFOP em Ouro Preto-MG (à direita).

Diferentemente da simulação, obter todos os parâmetros necessários para se calcular as métricas de mobilidade propostas não é tão trivial em situações reais. Exemplos são: a velocidade de cada motor é facilmente obtível, porém, escorregamentos na transmissão fazem com que a estimação da velocidade e posição angulares dos atuadores não seja acurada; a orientação e acelerações do dispositivo podem ser obtidas através de uma IMU, estimar entretanto a velocidade e posição do corpo utilizando-se apenas destas informações gera intrinsecamente um grande erro quando  $t \rightarrow \infty$ ; também sobre a obtenção da posição do robô, a maioria dos ambientes inspecionados não possui serviço de GNSS ou qualquer outra infraestrutura que dê suporte a isto; a odometria não pode ser corre-



Figura 6.2: Inspeção em mina abandonada no Bairro do Taquaral, em Ouro Preto-MG. Esta era utilizada para a extração de ouro na época colonial, porém hoje se encontra em zona residencial, tornando-se um risco em potencial para os moradores da região.

tamente calculada através apenas das velocidades dos atuadores devido a natureza dos dispositivos *Skid-steering* e aos recorrentes escorregamentos sobre o solo dos ambientes inspecionados; entre outras questões.

Devido a estes pontos, não foram obtidos em campo as informações detalhadas da mobilidade como realizado em simulação. Estes ensaios são porém de grande valia na análise qualitativa do comportamento do robô em ambientes reais.

Durante as missões, geralmente foi acoplada ao robô uma corda de segurança. Esta não possuía qualquer outra função senão a de garantir o içamento do protótipo caso este perdesse a comunicação com a base de comando. Não foram incluídos neste texto quaisquer relatos de situações onde este dispositivo teve de ser utilizado em auxílio à mobilidade do robô.

Outra questão sobre os testes de campo é sobre quais modos de locomoção foram utilizados. Devido ao baixo torque entregue atualmente pelos motores (descrito detalhadamente na Seção 2.1), o modo 6P não consegue desenvolver eficientemente a marcha tripé. O 4R2P também é penalizado por esta característica; as pernas laterais não têm força suficiente para auxiliar nas transposições de obstáculos. Além disso, as rodas tipo “estrela” ainda não estão disponíveis. Considerando isto, o modo 6R foi majoritariamente utilizado durante as inspeções de campo.

## 6.1 Testes iniciais

Os primeiros testes foram realizados na Mina da Passagem, em Mariana-MG e no campus da Universidade Federal de Ouro Preto. O enfoque nesta etapa inicial está em avaliar a comunicação, a interface de comando, a resposta aos comandos, a mecânica, entre outros aspectos estruturais e funcionais do protótipo.



Figura 6.3: Área do duto inspecionado na Mina Córrego do Feijão. A última foto mostra o EspeleoRobô durante a inspeção.

Um importante comportamento na locomoção com rodas foi notado. Sendo estas provenientes de bicicletas infantis, vários parafusos foram fixados às rodas com o intuito de aumento da tração. Ocorreu porém o revés, já que estas causaram um maior derrapamento em solo compacto, que é o tipo comumente encontrado em cavidades.

Um novo perfil foi implementado, retirando estas travas e confeccionando vários sulcos horizontais ao longo do comprimento da roda. Esta simples mudança aumentou consideravelmente a tração do dispositivo, sendo que investidas foram feitas contra encostas irregulares de considerável inclinação e o dispositivo se comportou bem na maior parte dos casos. Diversos furos laterais também foram feitos afim de aumentar a flexibilidade da roda. Estes dois perfis podem ser vistos na Figura 6.1.

O modo 4R2P apresenta boa mobilidade, entretanto é notado que a falta de rodas centrais torna difícil as tarefas de rotacionar sobre o próprio eixo e até mesmo fazer curvas enquanto deslocando-se frontalmente. As pernas laterais foram utilizadas para superar pequenos obstáculos e percebe-se que estas, de fato, auxiliam na transposição. Contudo, devido ao baixo torque, as pernas não conseguem efetuar o movimento quando operadas isoladamente; há sempre a necessidade das rodas atuarem em auxílio.

## 6.2 Teste em mina abandonada

À pedido da Prefeitura Municipal de Ouro Preto, o segundo teste de campo ocorreu em uma mina abandonada situada em zona de risco (Figura 6.2). A entrada é interdita devido aos riscos envolvidos; a inspeção então é de grande importância, já que auxilia na análise estrutural da mina sem colocar a vida de alguma pessoa em risco. A relevância é ainda maior pois a fenda está localizada em zona habitacional, sendo que o colapso da mesma pode causar diversos prejuízos à vida dos locais.

O dia do teste estava muito chuvoso, de forma que a área se encontrava parcialmente alagada. Apesar da presença de filetes de água e lama, o modo 6R conseguiu se locomover com certa facilidade. Cerca de 20 metros foram inspecionados, sendo que o dispositivo não conseguiu penetrar mais devido a um estreitamento no vão da mina.



Figura 6.4: EspeleoRobô no modo 6R ultrapassando um entulho de pedras.

### 6.3 Teste em duto de dreno

Um outro teste aconteceu na Mina Córrego do Feijão, situada no município de Brumadinho-MG. Devido às regras de segurança da Vale, o acesso de pessoas em áreas confinadas apenas pode ocorrer se o operador possuir treinamento e equipamentos adequados para tal; mesmo assim, alguns setores são totalmente interditados devido a alta periculosidade do ambiente.

O EspeleoRobô foi então levado para inspecionar um local desta categoria, sendo o duto de dreno situado debaixo de uma estrada (Figura 6.3). Por esta via passam diariamente dezenas de caminhões fora-de-estrada pesando mais de 300 toneladas; daí o risco intrínseco do local e a importância em se realizar a inspeção.

O duto não ofereceu grandes desafios à mobilidade. Todo o setor percorrido dispõe de área circular e chão delineado. Há porém na entrada da estrutura vários blocos empilhados que formam um obstáculo relativamente alto. Utilizando o modo 6R, o EspeleoRobô ultrapassou sem grandes dificuldades esta barreira. Está ilustrado na Figura 6.4 este momento.

### 6.4 Testes em galerias de vertedouro de barragens

Três ensaios foram realizados em galerias de vertedouro de barragens. Apesar de artificiais, estes ambientes geralmente apresentam dificuldades quanto à locomoção do dispositivo móvel, como por exemplo presença de água e lama.

A primeira inspeção neste tipo de ambiente foi realizada na barragem do Rio de Peixe, em Itabira-MG. No momento da inspeção, havia no chão da galeria uma camada de 15cm de lama, inviabilizando a movimentação do robô. A solução foi locomover o dispositivo



Figura 6.5: Inspeção na galeria de barragem localizada no Rio do Peixe. EspeleoRobô no modo 6R se deslocando sobre um duto (à esquerda) e visualização das vigas que foram utilizadas como referência visual (à direita).

sobre um duto que passa pelo interior estrutura (Figura 6.5).

A particularidade deste teste está na restrição da trajetória a ser realizada a uma estreita faixa de piso. Uma pequena fuga do objetivo e o robô poderia cair. Um eventual tombo seria catastrófico, já que o ambiente é totalmente interdito ao acesso humano e a lama provavelmente impediria o resgate do dispositivo via cabo.

O modo 6R efetuou eficientemente o desafio proposto. A estabilidade oferecida por este modo permitiu que o operador mantivesse o robô sobre o duto durante toda a inspeção. Modos que naturalmente causam trepidações na locomoção, como o 6E ou 6P, provavelmente teriam problemas em realizar este tipo de trajetória.

Outro desafio desta inspeção consistiu em manter o dispositivo nivelado lateralmente. A situação escura e confinada do ambiente não oferece muitos pontos de referência, isto retira do operador parte da percepção de orientação. A solução foi se inferir visualmente a inclinação do robô através das vigas de suporte de um duto superior.

Esta mesma estrutura impediu que o EspeleoRobô continuasse a inspeção devido a baixa altura. Ao fim, foram geradas imagens de cerca de 20m do interior da estrutura.

A segunda galeria inspecionada faz parte da Barragem de Taquaras, localizada na Mina da Mutuca, Nova Lima-MG (Figura 6.6). Havia considerável fluxo de água no interior da estrutura.

A locomoção do EspeleoRobô foi relativamente simples, sem grandes desafios. Um empecilho neste ensaio porém foi a comunicação entre o robô e a base de comando. Há uma curva acentuada à meia altura da extensão da galeria e, após passar por este ponto, pequenas falhas de comunicação ocorreram. Todavia, o final da estrutura foi alcançado, atingindo assim a marca de 80m inspecionados.

Uma situação inusitada ocorreu durante este teste. A barragem faz a contenção de uma lagoa populada por peixes, sendo que estes também estavam presentes na galeria. Em alguns momentos foi possível presencia-los nadando paralelamente ao robô enquanto



Figura 6.6: Inspeção na galeria da Barragem de Taquaras. À esquerda, EspeleoRobô iniciando o teste. À direita, momento em que um peixe foi flagrado circundando o robô.



Figura 6.7: Inspeção na galeria de barragem na Minas de Gongo Soco. As duas primeiras imagens mostram o momento em que o robô entram no duto. A terceira imagem mostra em detalhes as rugosidades da estrutura.

este se locomovia. Isto exemplifica o significativo volume de água presente no momento da inspeção.

A última galeria de vertedouro examinada está localizada na Mina de Gongo Soco, no município de Barão de Cocais-MG. Esta foi a que apresentou o maior desafio em questão de mobilidade. Havia um considerável fluxo de água, a profundidade do filete estava na altura do chassi do robô, muita lama se encontrava em toda a extensão da estrutura e o perfil do tubo da galeria é enrugado (Figura 6.7).

Devido ao volume de água, a estratégia visada foi locomover o dispositivo sobre as laterais do setor trafegável do duto. Desta forma, apenas uma roda se mantém na água, deixando porém o robô em situação muito inclinada.

A presença de lama também foi um fator dificultante, já que por várias vezes o robô derrapou até o ponto central da estrutura, ou simplesmente derrapava sobre as costeletas, sem conseguir se locomover.

Apesar das condições, o EspeleoRobô conseguiu inspecionar toda a extensão da galeria. Ao todo foram percorridos 230m, sendo esta a maior marca até então atingida em uma única inspeção.

## 6.5 Teste com a torre de mapeamento acoplada

Mais recentemente houveram dois testes de campo com a torre de mapeamento acoplada (Figura 6.8). O primeiro aconteceu na Mina do Chico Rei, em Ouro Preto-MG, e o segundo na Mina da Passagem, em Mariana-MG. Ambos são ambientes turísticos, sendo que o terreno oferece baixo desafio à mobilidade em sua maior parte. O modo de locomoção 6R foi o único utilizado para estes ensaios.

A tração das rodas com o solo aumentou devido ao maior peso do conjunto. Entretanto, houve um impacto do peso na força de tração devido ao baixo torque dos motores. Subir maiores inclinações e rotacionar sobre o próprio eixo se tornaram ações mais difíceis de se realizar.



Figura 6.8: Testes realizados do EspeleoRobô com a torre de mapeamento acoplada.

Em um dado momento na Mina da Passagem foi requisitado ao EspeleoRobô transpor um trecho mais acidentado, com vários blocos. A topografia deste local é similar à encontrada na Figura 6.4. Os obstáculos ofereceram relevante resistência à passagem do dispositivo.

A transposição em baixas velocidades se mostrou difícil, por vezes não factível. Várias vezes foi necessário aumentar a velocidade do robô afim de facilitar a transposição. Além disso, rotações sobre o próprio eixo se tornaram impraticáveis nesta situação.

Por fim, toda a extensão visada de ambas as minas foram mapeadas com êxito. O tipo de terreno enfrentado pelo conjunto robô+torre foi em sua maior parte plano. A transposição porém de terrenos não estruturados se mostrou penosa.

## Capítulo 7

# CONCLUSÕES GERAIS E TRABALHOS FUTUROS

Esta dissertação aborda a análise de mobilidade do EspeleoRobô, um dispositivo robótico concebido para a inspeção remota de cavidades. O robô em questão possui um sistema intercambiável dos mecanismos de locomoção, sendo possível configurar diferentes modos de locomoção no mesmo dispositivo. Desta maneira, uma análise de cada modo é feita e, posteriormente, a comparação entre estes é realizada.

A mobilidade é vista sob os aspectos de eficiência energética, estabilidade, altura de transposição e velocidade de translação. Afim de viabilizar a comparação quantitativa entre os modos, métricas de mobilidade são utilizadas.

Por ser um robô industrial em atividade, o mesmo não pode ficar grandes períodos de tempo disponibilizado para a realização de estudos ou implementação de melhorias. Devido a isto, um ambiente virtual de testes é implementado utilizando o simulador V-REP, ROS, MATLAB e códigos em Python. Este permite que a mobilidade seja analisada sem necessitar do protótipo real.

Diferentes testes são propostos em simulação. Para todos, os modos de locomoção são testados com e sem uma carga acoplada. A ideia é analisar a influência da adição da torre de mapeamento ao comportamento do robô.

Dentre os ensaios propostos, um destes ocorre em ambiente representativo. Para tal, o mapa tridimensional de uma caverna real foi importada no simulador. Desta forma, se torna possível avaliar o desempenho do robô em um ambiente similar ao que será encontrado nas missões reais.

As informações obtidas no ambiente virtual são utilizadas para analisar o desempenho dos modos. Os resultados de cada ensaio, para cada modo, são discriminados separadamente. As métricas normalizadas de todos os modos também são agrupadas em um mesmo gráfico, viabilizando a comparação direta de suas eficiências.

Testes de campo com um protótipo também são realizados. Apesar de ter sido concebido para a inspeção de cavernas, outros tipos de ambientes de também são inspecionados,

como dutos de dreno de estradas, minas abandonadas e galerias de vertedouro de barragens. Estes testes possuem apenas caráter qualitativo, não sendo obtidos daí dados que viabilizem uma criteriosa análise quantitativa. Testes de campo com a torre de mapeamento acoplada também foram realizados.

## 7.1 Contribuições da dissertação

A seguir estão descritas as contribuições desta dissertação.

### 7.1.1 Análise e comparação dos modos de locomoção

A contribuição primária desta dissertação está na análise e comparação do desempenho de diversos modos de locomoção aplicados especificamente no dispositivo robótico EspeleoRobô.

Os modos de locomoção avaliados são: 6 rodas (6R); 6 pernas com marcha tripé (6P); 4 rodas 2 pernas (4R2P); 4 rodas 2 rodas “estrela” (4R2E); 6 rodas tipo “estrela” (6E).

Sintetizando os resultados obtidos, o modo 6R é o mais balanceado, apresentando em média os melhores resultados em comparação aos demais modos de locomoção. Este apresenta boa relação entre consumo energético, estabilidade, velocidade de translação e capacidade de transpor terrenos acidentados. A adição da torre de mapeamento não afeta significativamente seu desempenho geral.

O modo 4R2P demonstra bons resultados pontuais. Seu consumo energético e capacidade de transpor obstáculos é no geral melhor que a dos demais. Entretanto, a adição de carga causa considerável perda de desempenho, principalmente em sua estabilidade e capacidade de transpor obstáculos.

Os modos que utilizam rodas tipo “estrela” se destacam pela maior velocidade de translação. No entanto, sua capacidade de transpor obstáculos é sensivelmente impactada pela adição da torre de mapeamento.

O modo 6P apresenta no geral um desempenho bem abaixo dos demais. Apesar de possuir o consumo energético instantâneo similar a dos outros modos, a baixa velocidade translacional faz com que sua eficiência energética fique bem aquém.

Cabe ressaltar que esta discussão possui um caráter generalizado do comportamento dos modos de locomoção propostos quando aplicados especificamente no EspeleoRobô.

Além do mais, por ser uma comparação generalizada baseada nos dados provenientes dos ensaios realizados, alguns modos podem possuir vantagens de locomoção em ambientes específicos que supera o desempenho dos demais. Como exemplo, o modo 6P é altamente indicado para terrenos de alto atrito, como áreas com lama, por manter poucos pontos de contato com o terreno (SIEGWART *et al.*, 2011).

Outro ponto a ser observado é em relação a metodologia aplicada. Incontáveis tipos

de robôs móveis terrestres são descritos na literatura. Cada um possui suas características de locomoção, unindo vantagens e desvantagens. Apesar de vários autores apresentarem resultados de desempenho referentes a seus sistemas robóticos, a bibliografia no geral carece de uma análise sistemática que permita comparar diretamente a performance de diferentes dispositivos entre si.

Sendo assim, outra contribuição deste trabalho é propor ensaios que, aliados à utilização de métricas sob certas condições, permitem a comparação de diferentes sistemas robóticos entre si, independente de suas dimensões e/ou mecanismos de locomoção utilizados.

### **7.1.2 Ambiente de simulação e algoritmos desenvolvidos**

O ambiente de simulação implementado também é uma importante contribuição. Além de servir como ferramenta para avaliação dos modos de locomoção, outras funcionalidades são disponibilizadas.

Por considerar vários parâmetros físicos do EspeleoRobô, a simulação permite uma avaliação do comportamento do mesmo. Isto auxilia inclusive no projeto mecânico do protótipo. Como exemplo, em simulação foi previsto que o torque máximo disponível pelos motores não seria suficiente para desenvolver a marcha tripé. Esta situação foi confirmada posteriormente com o protótipo. A simulação foi então usada também para avaliar qual seria o valor mínimo de torque necessário para fazer o robô andar corretamente no modo 6P.

Não se restringindo apenas ao EspeleoRobô, o ambiente de simulação criado pode ser aplicado a diversos outros dispositivos robóticos móveis. É possível implementar o novo modelo no V-REP e utilizar a biblioteca em Python desenvolvida para calcular as métricas de mobilidade vistas neste trabalho. Os códigos de comando escritos em ROS também podem ser facilmente acoplados para controlar o robô virtual. Todos os códigos e bibliotecas desenvolvidos no decorrer desta dissertação estão listados no Anexo A.

Além disto, devido à fidelidade de interação que o robô simulado possui se comparado ao protótipo, o simulador pode ser usado como plataforma de treinamento para novos operadores do EspeleoRobô. A forma de comandar o dispositivo, bem como sua resposta e a visualização das imagens das câmeras podem ser emulados de maneira fiel à realidade. Este treinamento diminui a probabilidade de haver danos em campo no robô devido a motivos de má pilotagem.

### **7.1.3 Publicações**

Os estudos sobre a mobilidade do EspeleoRobô com diversos modos de locomoção foram publicados em formato de artigo nos anais do Simpósio Brasileiro de Automação

Inteligente-SBAI (ROCHA *et al.*,2017b) e como resumo estendido nos anais do *International Conference on Intelligent Robots and Systems-IROS* (ROCHA *et al.*,2017a).

As características do EspeleoRobô, o sistema de troca rápida dos modos de locomoção e suas aplicações na mineração aparecem em artigos publicados nos anais da ABM Week (COTA *et al.*,2017) e novamente no Simpósio Brasileiro de Automação Inteligente-SBAI (COTA *et al.*,2017).

Expandindo o escopo desta pesquisa, dois pedidos provisório de patente relacionados ao robô foram depositados em fevereiro de 2017 (BRANDI *et al.*,2017a) e (BRANDI *et al.*,2017b). No dia 09 de fevereiro de 2018, um pedido de patente relacionando também às análises de mobilidade desenvolvidas nesta dissertação foi depositado no Escritório Americano de Patentes e Marcas (FREITAS *et al.*,2018).

## 7.2 Sugestões de trabalhos futuros

Como proposta de desenvolvimento futuro, tem-se a implementação de um *software* que, baseado no conhecimento sobre os modos de locomoção do EspeleoRobô gerados nesta dissertação, proponha qual modo deve ser utilizado em cada situação. O operador dirá ao programa características relevantes do terreno a ser inspecionado e o mesmo retornará o modo de locomoção indicado.

Além do mais, como descrito na Seção 2.4, uma torre de mapeamento está sendo desenvolvida para ser acoplada ao EspeleoRobô. Esta será capaz de gerar um mapa tridimensional com informações de cor do ambiente inspecionado. Uma próxima etapa do projeto consiste na implantação de algoritmos de navegação autônoma no robô utilizando-se primariamente destes mapas tridimensionais.

Propõem-se então a utilização da metodologia descrita no presente trabalho de maneira a auxiliar a navegação autônoma do EspeleoRobô. Como exemplos, mas não restrito apenas a estes, pode-se utilizar as métricas de estabilidade vistas para avaliar a topografia do terreno e evitar que o robô transponha trechos com risco de capotamento. As métricas de consumo energético também podem auxiliar o algoritmo a escolher sempre o caminho de menor consumo.

O conhecimento acerca dos modos de locomoção gerados por este trabalho também podem ser incorporados ao algoritmo de navegação autônoma. Desta maneira, através da avaliação do terreno, o *software* saberá avaliar quais trechos são melhor transpostos pelo modo de locomoção atual; para os trechos não transponíveis, será possível propor qual o melhor modo de locomoção que possa ser utilizado para transpô-lo.

# Referências Bibliográficas

- ANOUSAKI, G., KYRIAKOPOULOS, K. J., 2004, “A dead-reckoning scheme for skid-steered vehicles in outdoor environments”. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, v. 1, pp. 580–585. IEEE.
- BAKER, C., OMOHUNDRO, Z., THAYER, S., et al., 2006, “A case study in robotic mapping of abandoned mines”. In: *Field and Service Robotics*, pp. 487–495. Springer.
- BILLAH, M. M., AHMED, M., FARHANA, S., 2008, “Walking hexapod robot in disaster recovery: developing algorithm for terrain negotiation and navigation”. In: *Proceedings of World Academy of Science, Engineering and Technology*, v. 42, pp. 328–333.
- BRANDI, I. V., GOMES, M. F. M., ARAUJO, R. N., et al., 2017a. “Terrestrial mobile platform”. a. United States Patent and Trademark Office, Registro 62/458463, Deposito 13/02/2017.
- BRANDI, I. V., GOMES, M. F. M., ARAUJO, R. N., et al., 2017b. “Cave inspection and remote monitoring device”. b. United States Patent and Trademark Office, Registro 62/458311, Deposito 13/02/2017.
- BRIONES, L., BUSTAMANTE, P., SERNA, M. A., 1994, “Wall-climbing robot for inspection in nuclear power plants”. In: *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1409–1414. IEEE.
- BUEHLER, M., BATTAGLIA, R., COCOSCO, A., et al., 1998, “SCOUT: A simple quadruped that walks, climbs, and runs”. In: *ICRA*, pp. 1707–1712.
- COTA, E., TORRE, M. P., ROCHA, F. A. S., et al., 2017, “Dispositivo de Monitoramento Remoto de Cavidades - EspeleoRobô”. In: *SBAI'17. Proceedings., Simpósio Brasileiro de Automação Inteligente, 2017.*, pp. 1759–1766. SBA.

- COTA, E., TORRE, M. P., FERREIRA, J. A. T., et al., 2017, “ROBÓTICA NA MINERAÇÃO”. In: *Anais da ABM Week 2017, São Paulo-SP*. Associação Brasileira de Metalurgia, Materiais e Mineração-ABM.
- DE OLIVEIRA-GALVÃO, A. L. C., DA COSTA NETO, J. F., 2013, “PROPOSTA DE PROCEDIMENTO METODOLÓGICO PARA AVALIAÇÃO DO NÍVEL DE RELEVÂNCIA DE CAVIDADES NATURAIS SUBTERRÂNEAS”, *Revista Brasileira de Espeleologia*, v. 1, n. 3, pp. 19–34.
- EIJI, N., SEI, N., 1993, “Leg-wheel robot: A futuristic mobile platform for forestry industry”, pp. 109–112.
- FONTES, A. F. J., 2017. “Modelagem, adaptações e fabricação de componentes mecânicos de um robô para inspeção de cavidades”. Graduação em Engenharia Mecânica, Escola de Minas - Universidade Federal de Ouro Preto.
- FREITAS, G. M., 2014, *Reconfiguração de robôs móveis com articulação ativa navegando em terrenos irregulares*. Tese de Doutorado, COPPE-UFRJ.
- FREITAS, G. M., ROCHA, F. A. S., TORRE, M. P., et al., 2018. “Multi-terrain inspection robotic device and methods for configuring and guiding the same”. United States Patent and Trademark Office, WIPO - No PCT/BR2018/050025.
- FULL, R. J., TU, M. S., 1990, “Mechanics of six-legged runners”, *Journal of experimental biology*, v. 148, n. 1, pp. 129–146.
- GRAND, C., BENAMAR, F., PLUMET, F., et al., 2004, “Stability and traction optimization of a reconfigurable wheel-legged robot”, *The International Journal of Robotics Research*, v. 23, n. 10-11, pp. 1041–1058.
- HAYNES, G. C., RIZZI, A. A., 2006, “Gaits and gait transitions for legged robots”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1117–1122. IEEE.
- HONG, R., HU, G.-D., 2017, “The Direct Yaw Moment Control of the 6WD Skid-steering Vehicle Based on the Tire Longitudinal Forces Optimization Distribution”, Disponível em: <<http://www.dpi-proceedings.com/index.php/dtetr/article/viewFile/19472/18963>>.
- JUN, J. Y., CLARK, J. E., 2009, “Dynamic stability of variable stiffness running”. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 1756–1761. IEEE.

- LARSON, J., OKORN, B., PASTORE, T., et al., 2014, “Counter tunnel exploration, mapping, and localization with an unmanned ground vehicle”. v. 9084, p. 90840Q. International Society for Optics and Photonics, jun. doi: 10.1117/12.2054340. Disponível em: <<http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2054340>>.
- MANDOW, A., MARTINEZ, J. L., MORALES, J., et al., 2007, “Experimental kinematics for wheeled skid-steer mobile robots”. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1222–1227. IEEE.
- MARTÍNEZ, J. L., MANDOW, A., MORALES, J., et al., 2005, “Approximating kinematics for tracked mobile robots”, *The International Journal of Robotics Research*, v. 24, n. 10, pp. 867–878.
- MELO, L. C., FERREIRA, M., COSTA, C. A., 2003, “Desenvolvimento de revestimentos poliméricos para linhas flexíveis submersas”, *Anais do congresso brasileiro de PeD em petróleo e gas*, v. 2.
- MESSURI, D., KLEIN, C., 1985, “Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion”, *IEEE Journal on Robotics and Automation*, v. 1, n. 3, pp. 132–141.
- MICHEL, O., 2004, “Cyberbotics Ltd. Webots™: professional mobile robot simulation”, *International Journal of Advanced Robotic Systems*, v. 1, n. 1, pp. 5.
- MOORE, E., CAMPBELL, D., GRIMMINGER, F., et al., 2002, “Reliable stair climbing in the simple hexapod ‘RHex’”. In: *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, v. 3, pp. 2222–2227. IEEE.
- MORRIS, A., 2005. “Subterranean Robotics - Groundhog”. Disponível em: <[http://www.cs.cmu.edu/~./groundhog/robots\\_ghog.html](http://www.cs.cmu.edu/~./groundhog/robots_ghog.html)>.
- O’KANE, J. M., 2014, *A gentle Introduction to ROS*. University of South Carolina.
- RASHID, M. H., 2015. “Eletrônica de Potência: Dispositivos Circuitos e Aplicações”. .
- ROCHA, F. A. S., FREITAS, G. M., MAGALHÃES, P., et al., 2017a, “Mobility Analysis of a speleologist robot with flexible locomotion system”. In: *International Conference on Intelligent Robots and Systems, Vancouver-CA*. IEEE/RSJ, a.
- ROCHA, F. A. S., FREITAS, G. M., MAGALHÃES, P., et al., 2017b, “Análise de mobilidade de um dispositivo robótico para inspeção remota de cavidades”. In: *Simpósio Brasileiro de Automação Inteligente-SBAI, Porto Alegre-RS, Anais do SBAI 2017*. Sociedade Brasileira de Automática-SBA, b.

- ROHMER, E., SINGH, S. P., FREESE, M., 2013, “V-REP: A versatile and scalable robot simulation framework”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1321–1326. IEEE.
- ROSAS, RAFAEL; MACHADO, J., 2017. “Produção de minério de ferro da Vale bate recorde em 2016”. Disponível em: <http://www.valor.com.br/empresas/4871504/producao-de-minerio-de-ferro-da-vale-bate-recorde-em-2016>.
- SANTOS, A. V., DE GÓES, E., MIRANDA, F. P., et al., 2009, “Robô Ambiental Híbrido: um novo conceito em locomoção e monitoramento de áreas inóspitas na floresta amazônica”, *Anais do 2º Simpósio de Geotecnologias do Pantanal*.
- SARANLI, U., BUEHLER, M., KODITSCHKEK, D. E., 2000, “Design, modeling and preliminary control of a compliant hexapod robot”. In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, v. 3, pp. 2589–2596. IEEE.
- SARANLI, U., BUEHLER, M., KODITSCHKEK, D. E., 2001, “Rhex: A simple and highly mobile hexapod robot”, *The International Journal of Robotics Research*, v. 20, n. 7, pp. 616–631.
- SATO, T., FUKUI, R., MORISHITA, H., et al., 2004, “Construction of ceiling adsorbed mobile robots platform utilizing permanent magnet inductive traction method”. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, v. 1, pp. 552–558. IEEE.
- SICILIANO, B., KHATIB, O., 2008, *Springer handbook of robotics*. Springer Science & Business Media.
- SIEGWART, R., NOURBAKHSH, I. R., SCARAMUZZA, D., 2011, *Introduction to autonomous mobile robots*. MIT press.
- SILES, I., WALKER, I. D., 2009, “Design, construction, and testing of a new class of mobile robots for cave exploration”. In: *Mechatronics, 2009. ICM 2009. IEEE International Conference on*, pp. 1–6. IEEE.
- SPONBERG, S., FULL, R., 2008, “Neuromechanical response of musculo-skeletal structures in cockroaches during rapid running on rough terrain”, *Journal of Experimental Biology*, v. 211, n. 3, pp. 433–446.
- TAPPEINER, H., SKAFF, S., SZABO, T., et al., 2009, “Remote haptic feedback from a dynamic running machine”. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2368–2373. IEEE.

- UBIQUITI, 2012, *airMAX Yagi Antenna*. Ubiquiti Networks.
- VALE, 2016. “S11D Eliezer Batista Complex - Technological Advance”. Disponível em: <<http://www.vale.com/brasil/EN/initiatives/innovation/s11d/Pages/technological-progress.aspx>>.
- VELODYNE, 2017, *LiDAR PUCK - Real-time 3D LiDAR Sensor - VLP-16*. Velodyne.
- VON KARMAN, T., GABRIELLI, G., 1950, “What price speed? Specific power required for propulsion of vehicles”, *Mechanical Engineering*, v. 72, pp. 775–781.
- WETTERGREEN, D., JONAK, D., KOHANBASH, D., et al., 2010, “Field experiments in mobility and navigation with a lunar rover prototype”. In: *Field and service robotics*, pp. 489–498. Springer.
- WHITTAKER, W. L., WHITTAKER, W. C., THAYER, S. M., et al., 2003. “Robotic modeling of voids”. oct. Disponível em: <<https://patents.google.com/patent/US7069124B1/en>>.
- WILLIAMSON, D. F., PARKER, R. A., KENDRICK, J. S., 1989, “The box plot: a simple visual method to interpret data”, *Annals of internal medicine*, v. 110, n. 11, pp. 916–921.

# Apêndice A

## Códigos implementados

Os algoritmos desenvolvidos durante a pesquisa estão listados a seguir.

### Comando do protótipo

Os códigos de locomoção do EspeleRobô podem ser encontrados em repositório do Github: [https://github.com/fillshad/espeleo\\_src.git](https://github.com/fillshad/espeleo_src.git). Estão contidos neste *link* as bibliotecas de comunicação com o *driver* EPOS-MCD através de protocolo CAN, de comando dos mecanismos de locomoção e de gerenciamento do funcionamento do protótipo. O repositório é um pacote do ROS Indigo contendo inclusive os nós implementados.

### Ensaio simulados

Como visto no texto (Figura 5.2), um ambiente virtual foi criado com o simulador V-REP, o *Robot Operating System* e bibliotecas em Python. O ROS é utilizado para comandar o robô virtual. As bibliotecas conectam diretamente ao simulador afim de calcular e plotar para o usuário, em tempo de execução, as métricas de mobilidade. Outra possibilidade é salvar automaticamente os gráficos e informações geradas para posterior uso.

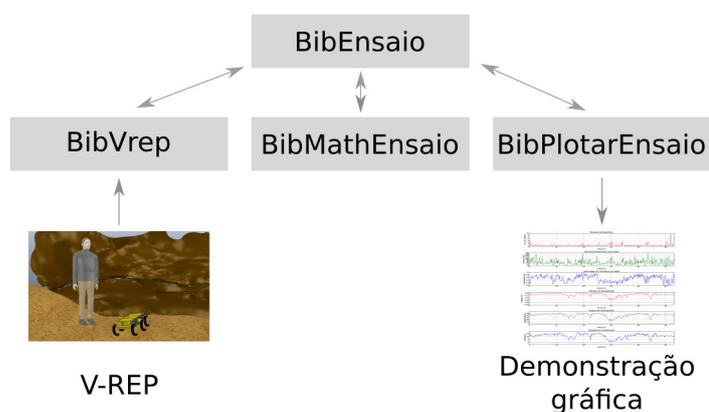


Figura A.1: Topologia de interação das bibliotecas em Python.

A organização das bibliotecas em Python pode ser consultada na Figura A.1. Sua implementação é feita visando a modularidade e compatibilidade, de modo que possam ser facilmente usadas em outros códigos. A seguir são expostas cada biblioteca, sua funcionalidade e respectivo código.

- **BibEnsaio:** Biblioteca principal. Faz a chamada e coordena a troca de informações entre as demais bibliotecas. Possui o valor dos parâmetros do EspeleoRobô. Coordena a execução dos códigos do ensaio.

```

1 # Biblioteca que realiza o ensaio do EspeleoRobo
2 import time
3 # import math
4 # import matplotlib.pyplot as plt
5 # import numpy as np
6 from BibVrep import bib_vrep
7 from BibMathEnsaio import bib_mathEnsaio
8 from BibPlotarEnsaio import bib_plotarEnsaio
9
10 class bib_ensaio:
11     # ===== DEFINICAO DOS PARAMETROS PARA CALCULOS =====
12
13     # Aceleracao da gravidade
14     gravidade = 9.80665
15
16     # Constante de torque do motor (em Nm/A)
17     const_torque = 24.3e-3
18
19     # Constante de velocidade do motor (em rad/s /V)
20     const_velocidade = 41
21
22     # Reducao total do motor - reducao:1
23     reducao_proporcao = 66
24
25     # Eficiencia da reducao (valor aplicavel ao torque) (em porcentagem)
26     reducao_eficiencia = 0.75
27
28     # Massa do robo (em kg)
29     massa_espeleo = 17.71
30
31     # Massa da torre (em kg)
32     massa_torre = 6.116
33
34     # Tensao da bateria (em Volts)
35     bateria_tensao = 32
36
37     # Capacidade da bateria (em Ampere/hora)
38     bateria_cap_ah = 9.9
39
40     # Maxima corrente de saida da bateria
41     bateria_maxcorrente_output = 10
42
43     # Maxima corrente de pico de saida da bateria
44     bateria_maxcorrente_outputPico = 18
45
46     # Capacidade total em coulombs da bateria
47     bateria_cap_coulomb = bateria_cap_ah * 3600
48
49     # Fator C da bateria
50     bateria_CFactor = bateria_maxcorrente_outputPico / bateria_cap_ah
51
52     # Fator E da bateria
53     bateria_EFactor = bateria_maxcorrente_output / bateria_cap_ah
54
55     # Quantidade de pontos a serem inseridos no plot dinamico
56     pltDin_qtdPts = 300
57
58     # ===== CONSTRUTOR DA CLASSE =====
59     def __init__(self):
60         print('Codigo iniciado')
61
62         # Inicia variaveis necessarias
63         self.nivelBateria = self.bateria_cap_coulomb # Nivel atual da bateria
64
65         #Instanciando as classes de apoio

```

```

66 self.vrep = bib_vrep()
67 self.bibMath = bib_mathEnsaio()
68 self.bibPlotar = bib_plotarEnsaio(False)
69
70 # Recebe o handle dos sensores de prox
71 # resp, self.handle_sensprox_ini = vrep.simxGetObjectHandle(self.ClientID, 'laser_inicial',
72 # resp, self.handle_sensprox_fim = vrep.simxGetObjectHandle(self.ClientID, 'laser_final', vrep
73 # .simx_opmode_blocking)
74
75 # ===== FUNCAO QUE REALIZA O ENSAIO =====
76 def realizarEnsaio(self, num_apoios):
77
78     # Pergunta ao usuario qual eh o modo de locomocao em uso
79     '''mod_loc = raw_input("Qual o modo de locomocao utilizado? ")
80     while not mod_loc:
81         mod_loc = raw_input("Por favor, digite um nome valido: ")'''
82     # mod_loc = 'teste'
83
84     # ---- Recebendo valores do usuario
85     # Pergunta ao usuario qual eh o modo de locomocao em uso
86     # Apenas passa para a execucao do ensaio quando o usuario entrar algo valido
87     flag_bem_recebido = False
88     while not flag_bem_recebido:
89
90         # Recebe um valor do usuario
91         num_apoios = raw_input("Quantos pontos de apoio o dispositivo possui? (3, 4 ou 6 pontos):
92         ")
93
94         # Testa se o valor esta de acordo
95         if num_apoios:
96             num_apoios = int(num_apoios)
97             if num_apoios == 3 or num_apoios == 4 or num_apoios == 6:
98                 flag_bem_recebido = True
99             else:
100                 print("O valor entrado e invalido. Vamos tentar novamente...")
101         else:
102             print("O valor entrado e invalido. Vamos tentar novamente...")
103
104     # Pergunta ao usuario se a torre esta fixada ao robo
105     # Apenas passa para a execucao do ensaio quando o usuario entrar algo valido
106     flag_bem_recebido = False
107     while not flag_bem_recebido:
108
109         # Recebe um valor do usuario
110         aux = raw_input("A torre esta fixada ao robo? ('s' ou 'n'): ")
111
112         # Testa se o valor esta de acordo
113         if aux:
114
115             # Verifica se a entrada do usuario foi sim
116             if aux == 's':
117                 torre_presente = True
118                 self.massa_espeleo = self.massa_espeleo + self.massa_torre
119                 flag_bem_recebido = True
120
121             # Verifica se a entrada do usuario foi nao
122             if aux == 'n':
123                 torre_presente = False
124                 flag_bem_recebido = True
125
126         # Envia uma mensagem ao usuario caso a entrada nao tenha sido valida
127         if not flag_bem_recebido:
128             print("O valor entrado e invalido. Vamos tentar novamente...")
129
130         else:
131             print("O valor entrado e invalido. Vamos tentar novamente...")
132
133     # ----- Iniciando de fato os testes -----
134     # Iniciando variaveis necessarias
135     flag_first_read = True
136
137     # Inicializacao de variaveis necessarias ao ensaio
138     vel_transl = []
139     potencia_instantanea_robo = []
140     corrente_instantanea_robo = []
141     resistencia_esp_instantanea_robo = []
142     potencia_instantanea_pmotor = []
143     base_tempo = []
144     bateria_carga_atual = []

```

```

143     maxima_tensao_bateria = []
144     delta_gasto_energetico = []
145     angulo_cap = []
146     energ_est = []
147     marg_est = []
148
149     # Trava nesta parte ate o sensor inicial ser acionado
150     print "Ensaio aguardando o acionamento do sensor inicial"
151     while not self.vrep.VREP_receberSensoresPresenca()[0]:
152         time.sleep(0.001)
153     print('Ensaio Iniciado...')
154
155     # Loop do ensaio. Este ira rodar indeterminadamente ate que algo o pare
156     flag_fim_ensaio = False
157     while not flag_fim_ensaio:
158
159         # ----- Recebendo os valores do VREP -----
160         vel = []
161         torque = []
162         for i in range(0, 6):
163
164             # Le o torque do motor i no VREP
165             ret_torque = self.vrep.VREP_receberTorque(i)
166
167             # Le a velocidade aplicada no motor i VREP
168             ret_vel = self.vrep.VREP_receberVelocidadeMotor(i)
169
170             # Salva os valores recebidos em uma lista
171             vel.append(ret_vel)
172             torque.append(ret_torque)
173
174         # Recebe a velocidade de translacao do robo nos tres eixos - Ignora o recebimento da
175         # velocidade angular
176         vel_linear, _ = self.vrep.VREP_receberVelocidadeTranslacional()
177
178         # Recebe a orientacao do robo em angulos de Euler
179         eul_ang = self.vrep.VREP_receberOrientacaoRoboEuler()
180
181         # Recebe o vetor de posicao do robo no espaco
182         pos = self.vrep.VREP_receberPosicaoRoboEspaco()
183
184         # ----- Tratamento do tempo do ensaio -----
185         # Recebe o tempo de simulacao quando a ultima requisicao ao VREP foi feita
186         # O valor eh dividido por 1000 pois o mesmo retorna em milisegundos
187
188         # Primeiro tempo a ser lido
189         if flag_first_read:
190
191             # Recebe o tempo atual do VREP
192             tempo_atual = self.vrep.VREP_receberTempoEnsaioUltimaRequisicao()
193
194             # Atualiza os tempos dos calculos
195             tempo_first = tempo_atual
196             tempo_atual = 0
197
198         else:
199             # Recebe o tempo do VREP e regulariza com o tempo inicial
200             tempo_atual = self.vrep.VREP_receberTempoEnsaioUltimaRequisicao() - tempo_first
201
202         # ----- Calculo da Estabilidade -----
203
204         # Calculando o vetor de sequencia de ligacao e o vetor com a posicao dos apoios com
205         # respeito ao robo
206         vet_seq_lig, p_cg_f = self.bibMath.estruturaRobo(num_apoios, torre_presente)
207
208         # Calculando a posicao do robo e de seus apoios com respeito ao mundo
209         p_mundo_robo, p_mundo_f = self.bibMath.calculoCoordenadasRobo(pos, eul_ang, num_apoios,
210         p_cg_f)
211
212         # Funcao que retorna a mag da Margem de estabilidadee, os pontos inicial e final da reta
213         ret_marg_est, ret_marg_est_ini, ret_marg_est_fim = self.bibMath.calculaMagMargEstabilidade
214         (vet_seq_lig, p_mundo_f, p_mundo_robo, eul_ang)
215
216         # Funcao que retorna a Energia de estabilidade e o angulo de capotamento
217         # return min(angulo_cap), min(energia), p_cg, p_inter[energia_min_index]
218         ret_angulo_cap, ret_energ_est, ret_energ_est_ini, ret_energ_est_fim = self.bibMath.
219         calculaEnergiaEstabilidade(vet_seq_lig,
220         p_mundo_f,
221         p_mundo_robo,
222         eul_ang,

```

```

218                                     self.massa_espeleo ,
219                                     self.gravidade)
220
221 # ----- Calculo do consumo energetico -----
222 ret_velTrans , ret_resEsp , ret_Tensao ,\
223 ret_PotPMotor , ret_PotTot , ret_CorTot = self.bibMath.calculaEnergetico(vel_linear , torque ,
224                                     vel , self.reducao_eficiencia ,
225                                     self.reducao_proporcao ,
226                                     self.massa_espeleo ,
227                                     self.gravidade ,
228                                     self.const_velocidade ,
229                                     self.const_torque)
230
231 # Salvando os retornos de estabilidade em listas
232 angulo_cap.append(ret_angulo_cap)
233 energ_est.append(ret_energ_est)
234 marg_est.append(ret_marg_est)
235
236 # Salvando os retornos de energia em listas
237 vel_transl.append(ret_velTrans)
238 resistencia_esp_instantanea_roboto.append(ret_resEsp)
239 # maxima_tensao_bateria.append(ret_Tensao)
240 # potencia_instantanea_pmotor.append(ret_PotPMotor)
241 potencia_instantanea_roboto.append(ret_PotTot)
242 corrente_instantanea_roboto.append(ret_CorTot)
243
244
245 # Atualizando o nivel da bateria
246 if flag_first_read :
247     delta_gasto_energetico.append(0)
248 else :
249     # Calculando a quantidade de energia retirada da bateria na ultima contagem de tempo
250     delta_gasto_energetico.append((tempo_atual - base_tempo[-1]) * corrente_instantanea_roboto
251 [-2])
252
253     self.nivelBateria -= delta_gasto_energetico[-1]
254
255 # ----- Salvando o tempo atual -----
256 base_tempo.append(tempo_atual)
257
258 # ----- Plotando os resultados -----
259 ''' if len(base_tempo) > self.pltDin_qtdPts:
260
261     # DEVEM SER ENVIADOS DOIS PONTOS PARA A PLOTAGEM, no caso os dois ultimos calculados
262
263     # Plotando a resistencia especifica
264     self.bibPlotar.plotLinha_genDyn('res_esp', '-r', base_tempo[-self.pltDin_qtdPts:],
265 resistencia_esp_instantanea_roboto[-self.pltDin_qtdPts:])
266
267     # Plotando a velocidade de translacao
268     self.bibPlotar.plotLinha_genDyn('velocidade', '-b', base_tempo[-self.pltDin_qtdPts:],
269 vel_transl[-self.pltDin_qtdPts:])
270
271     # Plotando a energia de estabilidade
272     self.bibPlotar.plotLinha_genDyn('energ_estabilidade', '-b', base_tempo[-self.
273 pltDin_qtdPts:], energ_est[-self.pltDin_qtdPts:]) '''
274
275 # Plota a margem de estabildiade
276 self.bibPlotar.plotPoligonoSuporte(vet_seq_lig , p_mundo_roboto , p_mundo_f , ret_marg_est_ini ,
277 ret_marg_est_fim)
278
279 # -----Indicando ao programa que ja passou da primeira leitura -----
280 if flag_first_read :
281     flag_first_read = False
282
283 # ----- Testa se o sensor final foi acionado -----
284 if self.vrep.VREP_receberSensoresPresenca()[1]:
285     flag_fim_ensaio = True
286
287 # TESTE TEMPORARIO PARA TERMINAR O ENSAIO
288 """ if base_tempo[-1] > 5:
289     flag_fim_ensaio = True"""
290
291 # Trava um pouco o ensaio para nao dar problema
292 # self.bibPlotar.pausaPrograma(0.2)
293
294 # ----- Fim do ensaio -----
295

```

```

292 # Avisa ao usuario o fim do ensaio
293 print "Fim do ensaio! Duracao: %0.1f segundos " % (base_tempo[-1]-base_tempo[0])
294
295 # Fecha o plot dinamico
296 # self.bibPlotar.fecharPlotDinamico()
297
298 # Criando a pasta onde serao salvos os resultados
299 caminho_ensaio = self.bibPlotar.criarPastaEnsaio()
300
301 # Enviando uma mensagem ao usuario
302 print "Os arquivos do seu ensaio serao salvos em "+caminho_ensaio+"/"
303
304 # Salvando os dados do ensaio em um arquivo excel
305 self.bibPlotar.salvarDadosEnsaioExcel(caminho_ensaio, base_tempo,
306                                     resistencia_esp_instantanea_roboto,
307                                     potencia_instantanea_roboto,
308                                     vel_transl,
309                                     marg_est,
310                                     angulo_cap,
311                                     energ_est)
312
313 # Printando na tela os graficos finais referentes ao ensaio
314 self.bibPlotar.plotGrafico.final(True, caminho_ensaio, base_tempo,
315                                 resistencia_esp_instantanea_roboto,
316                                 potencia_instantanea_roboto,
317                                 vel_transl,
318                                 marg_est,
319                                 angulo_cap,
320                                 energ_est)
321
322 # Travar o programa no fim
323 # self.bibPlotar.travarPrograma()
324
325 # ===== Teste de chamada da biblioteca =====
326
327 chamou = bib_ensaio()
328 chamou.realizarEnsaio(6)
329
330 '''if __name__ == '__main__':
331
332     pass'''

```

- **BibVrep:** Realiza a conexão com o V-REP através de uma *Application Programming Interface* (API). Recebe os *handlers* dos objetos relevantes na cena do V-REP. Requer ao simulador o início do *streaming* dos dados desejados. Fornece serviços de obtenção sob demanda de variáveis.

```

1 # BIBLIOTECA DE COMUNICACAO DO PYTHON COM O VREP
2 # Antes de instanciar um objeto daqui, verifique qual eh a porta que o VREP esta utilizando para
  a API remota em sua inicializacao
3
4 # Tenta importar a biblioteca do VREP
5 try:
6     import vrep
7 except:
8     print('-----')
9     print('"vrep.py" could not be imported. This means very probably that')
10    print('either "vrep.py" or the remoteApi library could not be found.')
11    print('Make sure both are in the same folder as this file,')
12    print('or appropriately adjust the file "vrep.py"')
13    print('-----')
14    print('')
15
16 import time
17
18
19 # Definicao da Classe
20 class bib_vrep:
21     def __init__(self):
22
23         # Inicializacao de variaveis globais que serao utilizados pela biblioteca
24         self.ClientID = []
25         self.handle_motores = []
26         self.handle_espeleo = []

```

```

27
28     self.VREP_conexao(19997)
29     self.VREP_iniciarStreamings()
30     self.VREP_enviarMensagemConsole("Biblioteca V-REP Python se conectou remotamente - FILIPE
    AUGUSTO SANTOS ROCHA")
31
32 # ===== METODO DE CONEXAO COM O VREP E GETTER DOS HANDLERS NECESSARIOS
33 # INPUT
34 # -porta: deve ser passada a porta para conexao com API remoto. Verifique esta no terminal
    do VREP enquanto inicializa
35 def VREP_conexao(self, porta):
36
37     # Fecha qualquer conexao ja existente com o VREP
38     vrep.simxFinish(-1)
39
40     # Se conecta ao VREP
41     self.ClientID = vrep.simxStart('127.0.0.1', porta, True, True, 5000, 5)
42
43     # Testa a conexao com o VREP
44     if self.ClientID != -1:
45         print('Programa conectado com sucesso ao VREP')
46     else:
47         print('A conexao com o VREP falhou')
48         return -1
49
50     # Envia uma mensagem ao VREP
51     vrep.simxAddStatusBarMessage(self.ClientID,
52                                 'Biblioteca V-REP Python se conectou remotamente',
53                                 vrep.simx_opmode_oneshot)
54
55     # Recebe os handlers dos motores
56     for i in range(0, 6):
57         _, aux_handle = vrep.simxGetObjectHandle(self.ClientID, 'motor' + str(i), vrep.
    simx_opmode_blocking)
58         self.handle_motores.append(aux_handle)
59
60     # Recebe o handle do espeleo
61     _, self.handle_espeleo = vrep.simxGetObjectHandle(self.ClientID, 'EspeleoRobo', vrep.
    simx_opmode_blocking)
62
63
64     # Retorna 1 caso a conexao tenha obtido resultado positivo
65     return 1
66
67 # ===== FUNCAO QUE INICIA OS STREAMINGS REQUISITADOS =====
68 def VREP_iniciarStreamings(self):
69
70     # streaming das informacoes dos motores
71     for i in range(0, 6):
72         vrep.simxGetJointForce(self.ClientID, self.handle_motores[i], vrep.simx_opmode_streaming)
73         vrep.simxGetJointPosition(self.ClientID, self.handle_motores[i], vrep.
    simx_opmode_streaming)
74         vrep.simxCallScriptFunction(self.ClientID, 'EspeleoRobo', vrep.sim_scripttype_childscript,
75                                     ('getM' + str(i + 1) + 'Velocity'),
76                                     [], [], [], bytearray(), vrep.simx_opmode_streaming)
77
78     # Orientacao do robo
79     vrep.simxGetObjectOrientation(self.ClientID, self.handle_espeleo, -1, vrep.
    simx_opmode_streaming)
80
81     # Posicao do robot
82     vrep.simxGetObjectPosition(self.ClientID, self.handle_espeleo, -1, vrep.
    simx_opmode_streaming)
83
84     # Velocidades angular e linear
85     vrep.simxGetObjectVelocity(self.ClientID, self.handle_espeleo, vrep.simx_opmode_streaming)
86
87     # Pega os sensores de proximidade
88     # a leitura
89     vrep.simxGetObjectGroupData(self.ClientID, vrep.sim_object_proximitysensor_type, 13, vrep.
    simx_opmode_streaming)
90
91
92     # Espera 1s para iniciar a coleta dos valores
93     time.sleep(0.3)
94
95 # ===== FUNCAO QUE ENVIA UMA MENSAGEM AO CONSOLE DO VREP =====
96 def VREP_enviarMensagemConsole(self, mensagem):
97     vrep.simxAddStatusBarMessage(self.ClientID, str(mensagem), vrep.simx_opmode_oneshot)
98

```

```

99 # ===== FUNCOES PARA REALIZAR LEITURAS NO VREP =====
100 # IMPORTANTE: ESTES METODOS APENAS FUNCIONAM QUANDO O METODO "INICIAR STREAMING" JA HOVER
    SIDO CHAMADO
101
102 # Recebe o torque de um motor especifico PS: O indice do motor eh o COBID dele -1
103 def VREP_receberTorque(self, indice_motor):
104
105     # Recebe o torque do motor
106     _, retTorque = vrep.simxGetJointForce(self.ClientID, self.handle_motores[indice_motor], vrep
        .simx_opmode_buffer)
107
108     # Adequa o torque ao lado correto do robo
109     if indice_motor < 3:
110         retorno = -retTorque
111     else:
112         retorno = retTorque
113
114     return retorno
115
116 # Funcao que retorna os valores dos sensores inicial e final do ensaio
117 def VREP_receberSensoresPresenca(self):
118
119     # Recebe o valor dos sensores na cena
120     _, ret_hand, ret_int, _, _ = vrep.simxGetObjectGroupData(self.ClientID, vrep.
        sim_object_proximitysensor_type, 13, vrep.simx_opmode_streaming)
121
122     # Retorna os sensores de presenca, inicial e final
123     return ret_int[0], ret_int[2]
124
125 # Recebe a velocidade de um motor especifico PS: o indice do motor eh o COBID dele -1
126 def VREP_receberVelocidadeMotor(self, indice_motor):
127
128     # Recebe a velocidade aplicada em um motor no VREP
129     _, _, retFloats, _, _ = vrep.simxCallScriptFunction(self.ClientID, 'EspeleoRobo',
        vrep.sim_scripttype_childscript,
130         ('getM' + str(indice_motor + 1) + 'Velocity'),
131         [], [], [], bytearray(),
132         vrep.simx_opmode_buffer)
133
134     # Adequa as velocidades recebidas aos lados corretos do robo
135     if indice_motor < 3:
136         retorno = -1*retFloats[0]
137     else:
138         retorno = retFloats[0]
139
140     return retorno
141
142
143 # Recebe a velocidade translacional nos tres eixos do centro do robo
144 def VREP_receberVelocidadeTranslacional(self):
145
146     # Le a velocidade translacional do robo
147     _, vel_linear, vel_angular = vrep.simxGetObjectVelocity(self.ClientID, self.handle_espeleo,
        vrep.simx_opmode_buffer)
148
149     return vel_linear, vel_angular
150
151
152 # Metodo que recebe a orientacao do sistema de coordenadas do robo
153 def VREP_receberOrientacaoRoboEuler(self):
154
155     _, eul_ang = vrep.simxGetObjectOrientation(self.ClientID, self.handle_espeleo, -1,
        vrep.simx_opmode_buffer)
156
157     return eul_ang
158
159
160 # Metodo que recebe a posicao do robo no espaco euclidiano
161 def VREP_receberPosicaoRoboEspaco(self):
162
163     _, pos = vrep.simxGetObjectPosition(self.ClientID, self.handle_espeleo, -1,
        vrep.simx_opmode_buffer)
164
165     return pos
166
167
168 # Metodo que recebe o tempo no simulador quando a ultima requisicao foi realizada
169 # O tempo retornado esta em segundos
170 def VREP_receberTempoEnsaioUltimaRequisicao(self):
171
172     return vrep.simxGetLastCmdTime(self.ClientID)/1000.0
173
174
175

```

```

176 # ===== FIM DA CLASSE =====
177 #chamou = bib_vrep()
178 #chamou.VREP_conexao(19997)
179 #chamou.VREP_iniciarStreamings()

```

- **BibMathEnsaio:** Realiza todos os cálculos relativos para a obtenção das métricas de mobilidade. Fornece ferramentas matemáticas para conversão de ângulos de Euler em Matriz de Rotação.

```

1 # BIBLIOTECA QUE CONTEM A MATEMATICA NECESSARIA AOS ENSAIOS
2 import math
3 import numpy as np
4 import matplotlib.path as mplPath
5
6
7 class bib_mathEnsaio:
8     # Construtor da Classe
9     def __init__(self):
10         a = 0
11
12     # ===== FUNCAO QUE, BASEADO NO NUMERO DE APOIOS DO ROBO, RETORNA SUA ESTRUTURA
13     # INPUTS
14     # -num_apoios: numero de apoios do robo (por exemplo, 6 caso esteja com 6 rodas)
15     @staticmethod
16     def estruturaRobo(num_apoios, presenca_torre):
17
18         # Define a altura de CG em Z de acordo com a presenca, ou nao, da torre
19         if presenca_torre:
20             altura_z = -0.1936
21         else:
22             altura_z = -0.14176
23
24         # Define a estrutura do robo de acordo com o numero de apoios do mesmo e da presenca ou nao
25         # da torre
26         # Estas variaveis sao utilizadas pelo calculo da estabilidade
27         # condicao quando esta sendo utilizado o robo com 4 rodas
28         if num_apoios == 4:
29
30             # Vetor guia para plotar o PS do robo
31             vet_seq_lig = [1, 2, 4, 3]
32
33             p_cg_f = np.matrix([[0.21156, -0.16388, altura_z],
34                                [-0.21244, -0.16386, altura_z],
35                                [0.21156, 0.16407, altura_z],
36                                [-0.21244, 0.16407, altura_z]])
37
38         # Condicao quando esta sendo utilizado o robo com 6 rodas
39         elif num_apoios == 6:
40
41             # Vetor guia para plotar o PS do robo
42             vet_seq_lig = [1, 2, 3, 6, 5, 4]
43
44             p_cg_f = np.matrix([[0.21156, -0.16388, altura_z],
45                                [0.00044, -0.21109, altura_z],
46                                [-0.21244, -0.16386, altura_z],
47                                [0.21156, 0.16407, altura_z],
48                                [-0.00044, 0.21127, altura_z],
49                                [-0.21244, 0.16407, altura_z]])
50
51         elif num_apoios == 3:
52
53             # Vetor guia para plotar o PS do robo
54             vet_seq_lig = [1, 2, 3]
55
56             p_cg_f = np.matrix([[0.21156, -0.16388, altura_z],
57                                [-0.21244, -0.16386, altura_z],
58                                [-0.00044, 0.21127, altura_z]])
59
60         return vet_seq_lig, p_cg_f
61
62     # ===== FUNCAO QUE CALCULA AS COORDENADAS DO ROBO COM RESPEITO AO MUNDO =====
63     # INPUTS
64     # -pos: vetor com a posicao no espaco do centro de gravidade do robo
65     # -eul_ang: rotacao do sistema de coordenada do robo com respeito ao mundo em angulo de
66     # euler ZYX
67     # -num_apoios: quantidade de apoios que o robo possui (por exemplo, com 6 rodas ou 4)
68     # OUTPUTS

```

```

67 # -p_mundo_roboto: vetor com as coordenadas do centro do roboto
68 # -p_mundo_f: lista de vetores com as coordenadas dos pontos de apoio do roboto com respeito a
    ele mesmo
69 def calculoCoordenadasRoboto(self, pos, eul_ang, num_apoios, p_cg_f):
70
71     # Cria o vetor para calculo de pos do roboto no mundo
72     p_mundo_roboto = np.array([pos[0], pos[1], pos[2]]).reshape(3, 1)
73
74     # Convertendo os angulos de euler em matriz de rotacao
75     R_mundo_roboto = self.eulerAnglesToRotationMatrix(eul_ang)
76
77     # Calcula a posicao dos pontos de apoio com respeito ao mundo
78     p_mundo_f = []
79     for i in range(0, num_apoios):
80         aux = np.dot(R_mundo_roboto, p_cg_f[i].transpose())
81         p_mundo_f.append(p_mundo_roboto + np.array(aux))
82
83     return p_mundo_roboto, p_mundo_f
84
85 # ===== FUNCAO QUE CONVERTE VETOR DE ANGULO DE EULER EM MATRIZ DE ROTACAO
86 # INPUT
87 # -vetor com os angulos de euler (radianos, por favor)
88 # OUTPUT
89 # -Matriz de rotacao (SE3)
90 @staticmethod
91 def eulerAnglesToRotationMatrix(theta):
92
93     R_x = np.array([[1, 0, 0],
94                    [0, math.cos(theta[0]), -math.sin(theta[0])],
95                    [0, math.sin(theta[0]), math.cos(theta[0])]
96                    ])
97
98     R_y = np.array([[math.cos(theta[1]), 0, math.sin(theta[1])],
99                    [0, 1, 0],
100                   [-math.sin(theta[1]), 0, math.cos(theta[1])]
101                   ])
102
103     R_z = np.array([[math.cos(theta[2]), -math.sin(theta[2]), 0],
104                    [math.sin(theta[2]), math.cos(theta[2]), 0],
105                    [0, 0, 1]
106                    ])
107
108     # Multiplicando as matrizes de rotacao em RxRyRz
109     aux = np.dot(R_x, R_y)
110     R = np.dot(aux, R_z)
111
112     return R
113
114 # ===== FUNCAO QUE RECEBE OS PONTOS DE APOIO DO ROBO E RETORNA A =====
115 # ===== MAGNITUDE DA MARGEM DE ESTABILIDADE, SEUS PONTOS INICIAL E FINAL =====
116 # INPUTS
117 # -vet_seq_lig: vetor indicando a sequencia de ligacao dos pontos para desenhar a Borda de
    Suporte ou Poligono
118 # -p_apoios: coordenadas dos pontos de apoio com respeito ao mundo
119 # -p_cg: coordenada do centro de gravidade do roboto com respeito ao mundo
120 # OUTPUTS
121 # -Modulo da margem de estabilidade
122 # -Coordenada do inicio da reta que liga o CG ao ponto mais proximo da BS
123 # -Coordenada de fim da reta que liga o CG ao ponto mais proximo da BS
124 def calculaMagMargEstabilidade(self, vet_seq_lig, p_apoios, p_centro_gravidade, euler):
125
126     # Extraindo a coordenada 2D do centro de gravidade do roboto
127     p_cg = np.array([p_centro_gravidade[0], p_centro_gravidade[1]])
128
129     # Extraindo as coordenadas extremas de uma linha do padrao de suporte
130     q1 = []
131     q2 = []
132     mag = []
133
134     # Calcula a margem de estabilidade com respeito a todos os vertices do PS
135     for i in range(len(p_apoios)):
136
137         if i < len(p_apoios) - 1:
138             q1.append(np.array([p_apoios[vet_seq_lig[i] - 1][0], p_apoios[vet_seq_lig[i] - 1][1]]))
139             q2.append(np.array([p_apoios[vet_seq_lig[i + 1] - 1][0], p_apoios[vet_seq_lig[i + 1] -
140             1][1]]))
141         else:
142             q1.append(np.array([p_apoios[vet_seq_lig[i] - 1][0], p_apoios[vet_seq_lig[i] - 1][1]]))
143             q2.append(np.array([p_apoios[vet_seq_lig[0] - 1][0], p_apoios[vet_seq_lig[0] - 1][1]]))

```

```

144 # Encontrando a norma da magnitude para este vertice especifico do PS
145 aux_cnt = np.hstack([np.array(q2[-1]) - np.array(q1[-1]), np.array(p_cg) - np.array(q1
146 [-1])])
147 aux_num = math.fabs(np.linalg.det(aux_cnt))
148 aux_den = np.linalg.norm(np.array(q2[-1]) - np.array(q1[-1]))
149 mag.append(aux_num / aux_den)
150
151 # Descobrimo a menor magnitude e seu index, que eh o valor da margem
152 mag_min_index = mag.index(min(mag))
153
154 # -- Encontrando a reta que liga cg ao vertice de menor magnitude
155
156 # Vetor perpendicular a linha entre pontos de apoio
157 v = np.array([q2[mag_min_index][1] - q1[mag_min_index][1], -(q2[mag_min_index][0] - q1[
158 mag_min_index][0])])
159
160 # Slope do vetor perpendicular
161 slope = v[1] / v[0]
162
163 # Encontrando o ponto intermediario - Considera-se que o x = 0
164 p_cg_reta = np.array([0, (-slope * p_cg[0] + p_cg[1])])
165
166 # linha 1
167 p1 = np.array([q1[mag_min_index][0][0], q1[mag_min_index][1][0]]) .T
168 p2 = np.array([q2[mag_min_index][0][0], q2[mag_min_index][1][0]]) .T
169
170 # linha 2
171 p3 = np.array([p_cg[0][0], p_cg[1][0]]) .T
172 p4 = np.array([p_cg_reta[0], p_cg_reta[1]]) .T
173
174 # Calculando a intersecao
175 intersecao = self.seg_intersect(p1, p2, p3, p4)
176
177 # Chama a funcao que notifica se o ponto esta dentro ou fora do PS
178 # Esta enviando q1 para ja ter os pontos organizados de acordo com a ligacao no PS
179 capotado = self.estaCapotado(vet_seq_lig, p_apoios, p_cg, euler)
180
181 # Retornando os valores
182 # - Valor da menor magnitude, que eh a margem
183 # - Ponto inicial da reta da margem
184 # - Ponto final da reta da margem
185 # - Booleano indicando se p_cg esta dentro do PS
186
187 if not capotado:
188     return min(mag), np.array([p_cg[0], p_cg[1]]) .T[0], intersecao
189 else:
190     return -min(mag), np.array([p_cg[0], p_cg[1]]) .T[0], intersecao
191
192 # ===== FUNCOES PARA CALCULAR A INTERSECAO ENTRE DOIS PONTOS
193 @staticmethod
194 def perp(a):
195     b = np.empty_like(a)
196     b[0] = -a[1]
197     b[1] = a[0]
198     return b
199
200 # line segment a given by endpoints a1, a2
201 # line segment b given by endpoints b1, b2
202 # return
203 def seg_intersect(self, a1, a2, b1, b2):
204     da = a2 - a1
205     db = b2 - b1
206     dp = a1 - b1
207     dap = self.perp(da)
208     denom = np.dot(dap, db)
209     num = np.dot(dap, dp)
210     return (num / denom.astype(float)) * db + b1
211
212 # ===== FUNCAO QUE CALCULA O ANGULO DE TOMBAMENTO E A ENERGIA =====
213 # INPUTS
214 # -vet_seq_lig: vetor indicando a sequencia de ligacao dos pontos para desenhar a Borda de
215 # Suporte ou Poligono
216 # -p_apoios: coordenadas dos pontos de apoio com respeito ao mundo
217 # -p_cg: coordenada do centro de gravidade do robo com respeito ao mundo
218 # OUTPUTS
219 # -Angulo de capotamento para a pose dada
220 # -Energia de tombamento para a pose dada
221 # -Coordenada do inicio da reta que liga o CG ao ponto mais proximo da BS
222 # -Coordenada de fim da reta que liga o CG ao ponto mais proximo da BS
223 def calculaEnergiaEstabilidade(self, vet_seq_lig, p_apoios, p_cg, euler, massa, gravidade):

```

```

221
222 # Extrairdo as coordenadas extremas de uma linha do padrao de suporte
223 v_R = []
224 p_inter = []
225 h = []
226 energia = []
227 angulo_cap = []
228
229 # Descobre se o ponto esta dentro, ou fora, do PS
230 capotado = self.estaCapotado(vet_seq_lig, p_apoios, p_cg, euler)
231
232 # Calcula a margem de estabilidade com respeito a todos os vertices do PS
233 for i in range(len(p_apoios)):
234
235     # Recebendo as coordenadas dos ptos de apoio necessarios
236     if i < len(p_apoios) - 1:
237         f1 = p_apoios[vet_seq_lig[i] - 1]
238         f2 = p_apoios[vet_seq_lig[i + 1] - 1]
239     else:
240         f1 = p_apoios[vet_seq_lig[i] - 1]
241         f2 = p_apoios[vet_seq_lig[0] - 1]
242
243     # Vetores ligando f1 a f2 e f1 ao cg
244     v_f1f2 = (f2 - f1).T[0]
245     v_f1cg = (p_cg - f1).T[0]
246
247     # Vetor projecao de v_f1cg em v_f1f2
248     v_proj_vf1cg_vf1f2 = ((np.dot(v_f1cg, v_f1f2)) / (np.linalg.norm(v_f1f2) ** 2)) * v_f1f2
249
250     # Ponto final da projecao (que eh a intersecao das duas linhas)
251     p_inter.append(f1.T[0] + v_proj_vf1cg_vf1f2)
252
253     # Vetor ligando cg a intersecao
254     v_R.append((p_cg.T - p_inter[-1].T)[0])
255
256     # Vetor normal ao plano unitario
257     z = np.array([0, 0, 1])
258     v_NPlano = np.cross(z, v_f1f2)
259     v_NPlano = v_NPlano / np.linalg.norm(v_NPlano)
260
261     # Projecao de v_R no plano para encontrar v_Rlinha
262     v_Rlinha = v_R[-1] - (((np.dot(v_R[-1], v_NPlano)) / np.linalg.norm(v_NPlano) ** 2) *
v_NPlano)
263
264     # Angulo entre v_R e v_Rlinha
265     theta_vR_vRlinha = np.arctan2(np.linalg.norm(np.cross(v_R[-1], v_Rlinha)), np.dot(v_R[-1],
v_Rlinha))
266
267     # Angulo entre v_Rlinha e z_chapeu
268     psi_vRlinha_zchapeu = np.arctan2(np.linalg.norm(np.cross(v_Rlinha, z)), np.dot(v_Rlinha, z
))
269
270     # Finalmente, calculo altura de deslocamento do cg
271     h.append(np.linalg.norm(v_R[-1]) * (1 - math.cos(theta_vR_vRlinha)) * (math.cos(
psi_vRlinha_zchapeu)))
272
273     # Calculo da energia potencial para o h encontrado
274     # Eh feito o calculo pelo absoluto para evitar problemas com cosseno negativo
275     energia.append(massa * gravidade * math.fabs(h[-1]))
276
277     # Angulo de capotamento
278     # Este eh o angulo entre -v_R e -z_chapeu
279     angulo_cap.append(
280         180 - math.degrees(math.atan2(np.linalg.norm(np.cross(v_R[-1], -z)), np.dot(v_R[-1], -z)
)))
281
282 # Descobrimo o index da menor energia
283 energia_min_index = energia.index(min(energia))
284
285 if not capotado:
286     return min(angulo_cap), min(energia), p_cg, p_inter[energia_min_index]
287 else:
288     return -min(angulo_cap), -min(energia), p_cg, p_inter[energia_min_index]
289
290 # ===== FUNCAO DESCOBRE SE UM DADO PONTO QUALQUER ESTA OU NAO DENTRO DE UM POLIGONO
=====
291 # INPUTS
292 # -q1: array com os pontos iniciais das arestas 2D
293 # -q2: array com os pontos finais das arestas 2D
294 # -p_cg: array com as coordenadas 2D de da projecao do centro de gravidade

```

```

295 # -alturas: alturas dos triangulos
296 # OUTPUT
297 # - True se esta dentro; False se estiver fora
298 def estaDentroPoligonoSuporte(self, vet_seq_lig, p_apoios, p_cg):
299
300 # Organizando os pontos da maneira correta como devem ser ligados
301 # Calcula a margem de estabilidade com respeito a todos os vertices do PS
302 apoios = []
303 for i in range(len(p_apoios)):
304     apoios.append(np.array([[p_apoios[vet_seq_lig[i] - 1][0], p_apoios[vet_seq_lig[i] -
305     1][1]]]))
306
307 # Criando uma matriz numpy com as coordenadas x e y dos pontos de apoio
308 flag_first = True
309 for i in range(len(apoios)):
310
311     if flag_first:
312         poligono_coord = np.array([[apoios[i][0][0], apoios[i][1][0]])]
313         flag_first = False
314     else:
315         poligono_coord = np.append(poligono_coord, [[apoios[i][0][0], apoios[i][1][0]], axis=0)
316
317 # Criando um path com as coordenadas do poligono
318 poligono = mplPath.Path(poligono_coord)
319
320 # Retornando o booleano que indica se o o ponto esta, ou nao, dentro de PS
321 return poligono.contains_point((p_cg[0][0], p_cg[1][0]))
322
323 # ===== FUNCAO QUE CALCULA A AREA DE UM POLIGONO QUALQUER, DADOS SEUS VERTICES
324 @staticmethod
325 def PolyArea(x, y):
326     return 0.5 * np.abs(np.dot(x, np.roll(y, 1)) - np.dot(y, np.roll(x, 1)))
327
328 # ===== FUNCAO RETORNA SE O ROBO SE ENCONTRA CAPOTADO OU NAO =====
329 # O calculo eh feito a partir da afericao da projecao de CG fora de PS ou se algum dos angulos
330 # de Roll and Pitch
331 # estao < -90 ou > 90.
332 #
333 # INPUTS
334 # -p_apoios: coordenadas dos pontos de apoio com respeito ao mundo
335 # -p_cg: coordenada do centro de gravidade do robo com respeito ao mundo
336 # OUTPUT
337 # True se o robo esta capotado, False caso negativo
338 def estaCapotado(self, vet_seq_lig, p_apoios, p_cg, euler):
339
340 # Condicao para dizer se o robo esta capotado ou nao
341 if self.estaDentroPoligonoSuporte(vet_seq_lig, p_apoios, p_cg) and -np.pi/2 < euler[0] < np.
342 pi/2 and -np.pi/2 < euler[0] < np.pi/2:
343     return False
344 else:
345     return True
346
347 # ===== FUNCAO QUE REALIZA OS CALCULOS ENERGETICOS DO ENSAIO =====
348 # INPUTS
349 # -vel_linear: vetor com a velocidade linear em cada eixo do centro do robo
350 # -torque: vetor de 6 pos com os torques aplicados por motor
351 # -vel: vetor de 6 pos com a velocidade aplicada por motor
352 # OUTPUTS
353 # -vel_transl: modulo de velocidade translacional do centro do robo (m/s)
354 # -resistencia_esp_instantanea_robo: resistencia especifica instantanea calculada
355 # -maxima_tensao_bateria: Maxima tensao retirada da bateria
356 # -aux_potencia_instantanea_pmotor: vetor com P instantanea retirada por motor
357 # -aux_potencia_instantanea_total: soma das potencias gastas por motor
358 # -aux_corrente_instantanea_robo: corrente consumida no total pelo robo
359 @staticmethod
360 def calculaEnergetico(vel_linear, torque, vel, reducao_eficiencia, reducao_proporcao, massa,
361 gravidade, const_velocidade, const_torque):
362
363 # Variaveis auxiliares necessarias
364 aux_potencia_instantanea_pmotor = []
365 aux_tensao_instantanea_pmotor = []
366 aux_potencia_instantanea_total = 0
367 aux_corrente_instantanea_robo = 0
368
369 # Calcula a velocidade de translacao no espaco do robo
370 # Converte as velocidades transl nos eixos na total do corpo
371 vel_transl = (vel_linear[0] ** 2 + vel_linear[1] ** 2 + vel_linear[2] ** 2) ** 0.5

```

```

372 for i in range(0, 6):
373     # Torque desenvolvido no motor, antes da reducao e considerando a eficiencia desta
374     Td = (torque[i] / reducao_proporcao) * (2 - reducao_eficiencia)
375
376     # Velocidade desenvolvido no motor antes da reducao
377     Vd = vel[i] * reducao_proporcao
378
379     # Calculo da energia gasta por motor
380     aux_potencia_instantanea_pmotor.append(
381         (math.fabs(Td) / const_torque) * (math.fabs(Vd) / const_velocidade))
382
383     # Somatorio da energia total (em Watts)
384     aux_potencia_instantanea_total += math.fabs(aux_potencia_instantanea_pmotor[-1])
385
386     # Corrente instantanea retirada da bateria (em Amperes)
387     aux_corrente_instantanea_robo += math.fabs(Td) / const_torque
388
389     # Tensao instantanea retirada da bateria
390     aux_tensao_instantanea_pmotor.append(math.fabs(Vd) / const_velocidade)
391
392     # Calculo da resistencia especifica instantanea do robo
393     resistencia_esp_instantanea_robo = aux_potencia_instantanea_total / (massa * gravidade *
394         vel_transl)
395
396     # Maxima tensao retirada da bateria neste instante de tempo
397     maxima_tensao_bateria = max(aux_tensao_instantanea_pmotor)
398
399     # Salvando valores calculados em lista
400     potencia_instantanea_robo = aux_potencia_instantanea_total
401     corrente_instantanea_robo = aux_corrente_instantanea_robo
402     potencia_instantanea_pmotor = aux_potencia_instantanea_pmotor
403
404     # Retornar
405     # vel_transl
406     return vel_transl, resistencia_esp_instantanea_robo, maxima_tensao_bateria, \
407         aux_potencia_instantanea_pmotor, \
408         aux_potencia_instantanea_total, aux_corrente_instantanea_robo

```

- **BibPlotarEnsaio:** Recebe o valor das métricas do ensaio e mostra em tempo de execução seus gráficos.

```

1 # Biblioteca com as funcionalidades para plotar o ensaio
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import time
5 import xlswriter
6 import os
7 import getpass
8 import datetime
9
10 class bib_plotarEnsaio:
11
12     # Construtor da Classe
13     def __init__(self, flag_plot_dinamico):
14
15         # Fechando todos os graficos eventualmente abertos
16         plt.close('all')
17         plt.ion()
18
19         if flag_plot_dinamico:
20             # --- Preparando a figura que recebera as plotagens dinamicas da linhas
21             # Criando a figura que recebera os valores
22             self.fig, self.axarr = plt.subplots(nrows=3, sharex=True)
23
24             # Devemos desenhar antes de comecar a animacao
25             self.fig.show()
26             self.fig.canvas.draw()
27
28             self.styles = ['r-', 'g-', 'y-']
29
30             # Captura as linhas das figuras
31             self.lines = [self.plotar(1, 1, ax, style) for ax, style in zip(self.axarr, self.styles)]
32
33             # Captura os backgrounds das figuras
34             self.backgrounds = [self.fig.canvas.copy_from_bbox(ax.bbox) for ax in self.axarr]
35
36             # --- Preparando a figura que recebera a plotagem da margem de estabilidade
37             self.figMargem, self.axMargem = plt.subplots()

```

```

38     self.figMargem.show()
39
40
41 # Funcao para plotar de inicio
42 @staticmethod
43 def plotar(x, y, ax, style):
44     return ax.plot(x, y, style, animated=True)[0]
45
46 # ===== Retorna um int dependendo da string desejada para o subindex
47
48 @staticmethod
49 def getIndiceSubplot(nome_subplot):
50     if nome_subplot == 'res_esp':
51         return 0
52     elif nome_subplot == 'potencia':
53         return 1
54     elif nome_subplot == 'velocidade':
55         return 2
56     elif nome_subplot == 'margem_estabilidade':
57         return 3
58     elif nome_subplot == 'angulo_capotamento':
59         return 4
60     elif nome_subplot == 'energ_estabilidade':
61         return 5
62
63 @staticmethod
64 def getIndiceSubplotDyn(nome_subplot):
65
66     if nome_subplot == 'res_esp':
67         return 0
68     elif nome_subplot == 'velocidade':
69         return 1
70     elif nome_subplot == 'energ_estabilidade':
71         return 2
72
73 # ===== FUNCAO QUE PLOTA OS GRAFICOS FINAIS DO ENSAIO
74 @staticmethod
75 def plotGrafico_final(salvar_graficos, caminho, tempo, res_esp, potencia, velocidade, marg_est
76     , ang_cap, ener_est):
77
78     # Enviando uma mensagem ao usuario
79     if salvar_graficos:
80         print "Plotando e salvando em arquivos os graficos do ensaio..."
81     else:
82         print "Plotando os graficos do ensaio..."
83
84     # Criando a Figura para colocar os graficos finais
85     figura, axis_arr = plt.subplots(nrows=6)
86     plt.pause(0.01)
87     figura.set_size_inches(15, 15)
88     plt.pause(0.01)
89
90     # ---Configurando os plots ---
91     # Resistencia especifica
92     axis_arr[0].set_xlabel('tempo [s]')
93     axis_arr[0].set_ylabel('res. esp. [adim]')
94     axis_arr[0].set_title('Resistencia Especifica')
95     axis_arr[0].plot(tempo, res_esp, '-r')
96     axis_arr[0].grid(True)
97     axis_arr[0].set_xlim(tempo[0], tempo[-1])
98
99     # Potencia Consumida
100    axis_arr[1].set_xlabel('tempo [s]')
101    axis_arr[1].set_ylabel('potencia [W]')
102    axis_arr[1].set_title('Potencia Instantanea Consumida')
103    axis_arr[1].plot(tempo, potencia, '-g')
104    axis_arr[1].grid(True)
105    axis_arr[1].set_xlim(tempo[0], tempo[-1])
106
107    # Velocidade de translacao
108    axis_arr[2].set_xlabel('tempo [s]')
109    axis_arr[2].set_ylabel('velocidade [m/s]')
110    axis_arr[2].set_title('Velocidade de Translacao do Robo')
111    axis_arr[2].plot(tempo, velocidade, '-b')
112    axis_arr[2].grid(True)
113    axis_arr[2].set_xlim(tempo[0], tempo[-1])
114
115    # Margem de estabilidade
116    axis_arr[3].set_xlabel('tempo [s]')

```

```

116 axis_arr[3].set_ylabel('margem [m]')
117 axis_arr[3].set_title('Margem de Estabilidade')
118 axis_arr[3].plot(tempo, marg_est, '-r')
119 axis_arr[3].plot(tempo, [0]*len(tempo), '--k')
120 axis_arr[3].grid(True)
121 axis_arr[3].set_xlim(tempo[0], tempo[-1])
122 axis_arr[3].set_ylim(-0.05, max(marg_est) * 1.1)
123
124 # Angulo de capotamento
125 axis_arr[4].set_xlabel('tempo [s]')
126 axis_arr[4].set_ylabel('angulo [graus]')
127 axis_arr[4].set_title('Angulo de Tombamento')
128 axis_arr[4].plot(tempo, ang_cap, '-g')
129 axis_arr[4].plot(tempo, [0] * len(tempo), '--k')
130 axis_arr[4].grid(True)
131 axis_arr[4].set_xlim(tempo[0], tempo[-1])
132 axis_arr[4].set_ylim(-5, max(ang_cap) * 1.1)
133
134 # Energia de estabilidade
135 axis_arr[5].set_xlabel('tempo [s]')
136 axis_arr[5].set_ylabel('energia [J]')
137 axis_arr[5].set_title('Energia de Tombamento')
138 axis_arr[5].plot(tempo, ener_est, '-b')
139 axis_arr[5].plot(tempo, [0] * len(tempo), '--k')
140 axis_arr[5].grid(True)
141 axis_arr[5].set_xlim(tempo[0], tempo[-1])
142 axis_arr[5].set_ylim(-3, max(ener_est) * 1.1)
143
144 plt.tight_layout()
145 plt.pause(0.01)
146
147 plt.draw()
148 plt.pause(0.01)
149
150 # Salva os graficos, caso o usuario peca
151 if salvar_graficos:
152
153     # Salva as figuras completas
154     figura.savefig(caminho + '/graf00_completo.png')
155     figura.savefig(caminho + '/graf00_completo.eps')
156
157     # Salva especificamente a resistencia especifica
158     extent = axis_arr[0].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
159     figura.savefig(caminho + '/graf01_res_esp.png', bbox_inches=extent.expanded(1.2, 1.3))
160     figura.savefig(caminho + '/graf01_res_esp.eps', bbox_inches=extent.expanded(1.2, 1.3))
161
162     # Salva especificamente a resistencia especifica
163     extent = axis_arr[1].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
164     figura.savefig(caminho + '/graf01_potencia.png', bbox_inches=extent.expanded(1.2, 1.3))
165     figura.savefig(caminho + '/graf01_potencia.eps', bbox_inches=extent.expanded(1.2, 1.3))
166
167     # Salva especificamente a resistencia especifica
168     extent = axis_arr[2].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
169     figura.savefig(caminho + '/graf01_velocidade.png', bbox_inches=extent.expanded(1.2, 1.3))
170     figura.savefig(caminho + '/graf01_velocidade.eps', bbox_inches=extent.expanded(1.2, 1.3))
171
172     # Salva especificamente a resistencia especifica
173     extent = axis_arr[3].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
174     figura.savefig(caminho + '/graf01_margem.png', bbox_inches=extent.expanded(1.2, 1.3))
175     figura.savefig(caminho + '/graf01_margem.eps', bbox_inches=extent.expanded(1.2, 1.3))
176
177     # Salva especificamente a resistencia especifica
178     extent = axis_arr[4].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
179     figura.savefig(caminho + '/graf01_angulo_tombamento.png', bbox_inches=extent.expanded(1.2,
180     1.3))
181     figura.savefig(caminho + '/graf01_angulo_tombamento.eps', bbox_inches=extent.expanded(1.2,
182     1.3))
183
184     # Salva especificamente a resistencia especifica
185     extent = axis_arr[5].get_window_extent().transformed(figura.dpi_scale_trans.inverted())
186     figura.savefig(caminho + '/graf01_energia.png', bbox_inches=extent.expanded(1.2, 1.3))
187     figura.savefig(caminho + '/graf01_energia.eps', bbox_inches=extent.expanded(1.2, 1.3))
188
189 # ===== Funcao que atualiza um grafico qualquer baseado no indice enviado
190
191 def plotLinha_genDyn(self, subplot, estilo, vx, vy):
192
193     # Testa o tipo da mensagem enviado
194     if isinstance(subplot, str):

```

```

193 # Se for string, descobre o indice do subplot baseado na string e plota neste
194 # self.axarr[self.getIndiceSubplot(subplot)].plot(vx, vy, estilo)
195 index = self.getIndiceSubplotDyn(subplot)
196
197 elif isinstance(subplot, int):
198 # Se for int, plot sobre o indice indicado
199 # Se for string, descobre o indice do subplot baseado na string e plota neste
200 index = subplot
201
202 else:
203 print "ERRO: a funcao getIndiceSubplot (BibPlotarEnsaio) recebeu um parametro erroneo"
204 index = -1
205
206 # Atualizando o grafico de modo dinamico e rapido
207 self.fig.canvas.restore_region(self.backgrounds[index])
208 self.lines[index].set_ydata(vy)
209 self.lines[index].set_xdata(vx)
210 self.axarr[index].set_ylim(min(vy) - min(vy) * 0.1, max(vy) + max(vy) * 0.1)
211 self.axarr[index].set_xlim(vx[0], vx[-1])
212 self.axarr[index].draw_artist(self.lines[index])
213 self.fig.canvas.blit(self.axarr[index].bbox)
214
215 # Gera a modificacao no grafico
216 self.fig.canvas.flush_events()
217
218 # ===== FUNCAO QUE PLOTA A MARGEM DE SUPORTE =====
219 # self.bibPlotar.plotPoligonoSuporte(vet_seq_lig, p_mundo_rob, p_mundo_f, ret_marg_est_ini,
220 # ret_marg_est_fim)
221 def plotPoligonoSuporte(self, vet_seq_lig, p_centro_gravidade, p_apoios, marg_ini, marg_fim):
222 # Sequenciando os pontos de apoio de acordo com a maneira como serao ligados
223 q1 = []
224 q2 = []
225 for i in range(len(p_apoios)):
226
227 if i < len(p_apoios) - 1:
228 q1.append(np.array([p_apoios[vet_seq_lig[i] - 1][0][0], p_apoios[vet_seq_lig[i] -
229 1][1][0]]))
230 q2.append(np.array([p_apoios[vet_seq_lig[i + 1] - 1][0][0], p_apoios[vet_seq_lig[i + 1]
231 - 1][1][0]]))
232 else:
233 q1.append(np.array([p_apoios[vet_seq_lig[i] - 1][0][0], p_apoios[vet_seq_lig[i] -
234 1][1][0]]))
235 q2.append(np.array([p_apoios[vet_seq_lig[0] - 1][0][0], p_apoios[vet_seq_lig[0] -
236 1][1][0]]))
237
238 # Limpando o array
239 self.axMargem.clear()
240
241 # Organizando as coordenadas dos pontos de apoio
242 x = []
243 y = []
244 for apoio in p_apoios:
245 x.append(apoio[0])
246 y.append(apoio[1])
247
248 # Desenhando o ponto de apoio
249 self.axMargem.scatter(x, y, marker=(5, 0))
250
251 # Desenhando a posicao do centro de gravidade
252 self.axMargem.scatter(p_centro_gravidade[0], p_centro_gravidade[1], marker=(5, 2))
253
254 # Desenhando o PS
255 for i in range(len(q1)):
256 if i < len(q1) - 1:
257 self.axMargem.plot([q1[i][0], q2[i][0]], [q1[i][1], q2[i][1]], '-g')
258 else:
259 self.axMargem.plot([q1[i][0], q2[i][0]], [q1[i][1], q2[i][1]], '--g')
260
261 # Desenhando a margem de estabilidade
262 self.axMargem.plot([marg_ini[0], marg_fim[0]], [marg_ini[1], marg_fim[1]], '-b')
263
264 # Ajusta a proporcao
265 self.axMargem.set_aspect('equal', 'datalim')
266
267 # Desenha a imagem
268 self.figMargem.canvas.draw()
269
270 # Funcao para fechar a figura do plot dinamico
271 def fecharPlotDinamico(self):

```

```

268     plt.close(self.fig)
269
270 # Funcao para travar o programa, porem permitir redraws
271 @staticmethod
272 def travarPrograma():
273     print "Fim do programa. Pressione qualquer tecla para finalizar tudo..."
274     plt.waitforbuttonpress()
275
276 # Funcao para pausar o programa por um certo tempo
277 @staticmethod
278 def pausaPrograma(tempo):
279     plt.pause(tempo)
280
281 # Funcao que cria a pasta onde sera salvo os dados do ensaio
282 # Recebe o nome do modo de locomocao em uso
283 @staticmethod
284 def criarPastaEnsaio():
285
286     mod_loc = "teste"
287
288     # Descobrimo o nome do usuario da maquina
289     usuario = getpass.getuser()
290
291     # Recolhendo a data e horario atuais
292     agora = datetime.datetime.now()
293
294     # Montando o caminho onde sera criada a pasta com os resultados
295     caminho_pasta = "/home/" + usuario + "/Desktop/Teste_" + str(agora.year) + "_" + str(agora.
296     month) + "_" + str(
297         agora.day)
298
299     # Detecta se a pasta onde sera enviado os resultados ja existe
300     if not os.path.exists(caminho_pasta):
301         # Caso nao exista, cria a pasta
302         os.mkdir(caminho_pasta)
303
304     # Cria a subpasta do ensaio especifico
305     caminho_subpasta = caminho_pasta + '/Ensaio_' + mod_loc + "_" + str(agora.hour) + '_' + str(
306     agora.minute) + '_' + str(agora.second)
307     os.mkdir(caminho_subpasta)
308
309     return caminho_subpasta
310
311 # Funcao para salvar os dados do ensaio em um arquivo excel
312 @staticmethod
313 def salvarDadosEnsaioExcel(caminho_subpasta, tempo, res_esp, potencia, velocidade, marg_est,
314     ang_cap, ener_est):
315
316     # Enviando uma mensagem ao usuario
317     print "Salvando os dados em arquivo excel..."
318
319     # Criando o arquivo basico
320     workbook = xlswriter.Workbook(caminho_subpasta+'Dados_Ensaio.xls')
321
322     # ----- Cria a primeira worksheet que ira salvar os dados do ensaio
323     worksheet1 = workbook.add_worksheet()
324
325     # Criando o cabecalho da tabela
326     worksheet1.write_row(0, 0, ["Tempo [s]", "Res Esp", "Potencia [W]", "Velocidade [m/s]", "
327     Marg Est [m]",
328     "Angulo Capot [graus]", "Energia Estabilidade"])
329
330     # Montando o array que sera escrito com os resultados dos ensaios
331     array1 = [tempo, res_esp, potencia, velocidade, marg_est, ang_cap, ener_est]
332
333     # Escrevendo os dados no worksheet
334     row = 1
335     for col, data in enumerate(array1):
336         worksheet1.write_column(row, col, data)
337
338     # ----- Cria um segundo worksheet para salvar as estatisticas
339     worksheet2 = workbook.add_worksheet()
340
341     worksheet2.write_column(0, 0, ["Tempo Ensaio [s]",
342     "Res Esp Media ", "Res Esp Desv Padrao", "Res Esp Mix", "Res Esp Man",
343     "Pot Media [W]", "Pot Desv Padrao [W]", "Pot Min [W]", "Pot Max [W]",
344     "Vel Media [m/s]", "Vel Desv Padrao [m/s]", "Vel Min [m/s]", "Vel Max [m/s]

```

```

341         "AT Media [graus]", "AT Desv Padrao [graus]", "AT Min [graus]", "AT Max [
    graus]",
342         "ET Media [J]", "ET Desv Padrao [J]", "ET Min [J]", "ET Max [J]")
343
344 # >> Calculando os valores a serem salvos
345
346 # Cria um array numpy com a resistencia especifica
347 arr_res_esp = np.array(res_esp)
348 arr_pot = np.array(potencia)
349 arr_vel = np.array(velocidade)
350 arr_ang = np.array(ang_cap)
351 arr_enr = np.array(ener_est)
352
353 res_esp_media = np.mean(arr_res_esp) # Resistencia especifica media
354 res_esp_desv = np.std(arr_res_esp) # Desvio padrao da resistencia especifica
355 res_esp_max = np.amax(arr_res_esp) # Pico da resistencia especifica
356 res_esp_min = np.amin(arr_res_esp) # Menor valor da resistencia especifica
357
358 pot_media = np.mean(arr_pot) # Potencia media consumida
359 pot_desv = np.std(arr_pot) # Desvio padrao da potencia
360 pot_max = np.amax(arr_pot) # Pico de potencia
361 pot_min = np.amin(arr_pot) # Minimo de potencia
362
363 vel_media = np.mean(arr_vel) # Velocidade media
364 vel_desv = np.std(arr_vel) # Desvio padrao da velocidade
365 vel_min = np.amin(arr_vel) # Minimo de velocidade
366 vel_max = np.amax(arr_vel) # Pico de velocidade
367
368 arr_ang_positivo = [i for i in arr_ang if i > 0] # Extrai apenas os valores positivos do
    array
369 ang_media = np.mean(arr_ang_positivo) # Angulo de Tombamento media
370 ang_desv = np.std(arr_ang_positivo) # Desvio padrao da Angulo de Tombamento
371 ang_min = np.amin(arr_ang_positivo) # Minimo de Angulo de Tombamento
372 ang_max = np.amax(arr_ang_positivo) # Pico de Angulo de Tombamento
373
374 arr_enr_positivo = [i for i in arr_enr if i > 0] # Extrai apenas os valores positivos do
    array
375 enr_media = np.mean(arr_enr_positivo) # Energia de Capotamento media
376 enr_desv = np.std(arr_enr_positivo) # Desvio padrao da Energia de Capotamento
377 enr_min = np.amin(arr_enr_positivo) # Minimo de Energia de Capotamento
378 enr_max = np.amax(arr_enr_positivo) # Pico de Energia de Capotamento
379
380 # Escrevendo na planilha
381 worksheet2.write_column(0, 1, [tempo[-1], res_esp_media, res_esp_desv, res_esp_min,
    res_esp_max,
382         pot_media, pot_desv, pot_min, pot_max,
383         vel_media, vel_desv, vel_min, vel_max,
384         ang_media, ang_desv, ang_min, ang_max,
385         enr_media, enr_desv, enr_min, enr_max])
386
387 # ----- Salva propriamente o arquivo
388 workbook.close()

```