

## Otimização do controlo de um robô quadrúpede usando ferramentas do MATLAB Simulink

ITALO FERNANDO RODRIGUES DE OLIVEIRA

julho de 2017

# Otimização do controlo de um robô quadrúpede usando ferramentas do MATLAB<sup>®</sup> Simulink<sup>™</sup>

Italo Fernando Rodrigues de Oliveira



Mestrado em Engenharia Eletrotécnica e de Computadores  
Área de Especialização em Sistemas Autónomos  
Departamento de Engenharia Eletrotécnica  
Instituto Superior de Engenharia do Porto  
2016-2017



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação do 2º ano do Mestrado em Engenharia Electrotécnica e de Computadores.

Candidato: Italo Fernando Rodrigues de Oliveira, 1160384@isep.ipp.pt

Orientação científica: Ramiro de Sousa Barbosa, rsb@isep.ipp.pt

Coorientação científica: Manuel dos Santos Silva, mss@isep.ipp.pt

Coorientação científica (UTFPR): Roberto Ribeiro Neli, neli@utfpr.edu.br



Mestrado em Engenharia Eletrotécnica e de Computadores  
Área de Especialização em Sistemas Autónomos  
Departamento de Engenharia Eletrotécnica  
Instituto Superior de Engenharia do Porto  
2016-2017



---

# Agradecimentos

---

Agradeço à Universidade Tecnológica Federal do Paraná e ao Instituto Superior de Engenharia do Porto por terem celebrado, através da Dupla Titulação, um acordo tão importante para a internacionalização de seus alunos. Também incluo nos agradecimentos todos os professores e funcionários destas duas instituições, que são os verdadeiros responsáveis por torná-las grandes centros de pesquisa e de desenvolvimento humano e intelectual.

Agradeço também ao orientador Ramiro Barbosa e ao coorientador Manuel Silva pelas várias horas disponibilizadas durante os últimos meses para auxiliar no desenvolvimento técnico e científico desta obra, bem como o professor Roberto Neli pelo esclarecimento de dúvidas referentes ao programa de Dupla Titulação.

Finalmente, meu maior agradecimento vai aos meus familiares que muito se esforçaram para me ajudar durante este ano em Portugal, e principalmente, aos meus amados pais Márcio Roberto de Oliveira e Veronice Rodrigues da Silva que sempre me incentivaram a seguir os meus sonhos e foram minha fonte de inspiração durante toda a vida.



---

# Resumo

---

A robótica móvel desde seus primórdios procurou satisfazer as necessidades de locomoção do ser humano, e devido à eficiência do sistema opta-se pela locomoção por rodas em pisos regulares. Porém, com o passar do tempo as necessidades científicas e tecnológicas abriram espaço para a locomoção com pernas devido a esta ter facilidade em atravessar terrenos íngremes e/ou irregulares nos quais os robôs com rodas possuem óbvias restrições.

A fiabilidade dos sistemas robóticos também precisou ser melhorada devido ao aumento da complexidade das tarefas, e tal feito passa por um sistema de controlo eficiente. No campo de robôs com locomoção por pernas vários estudos concentram-se na otimização dos modelos, porém um número menor se atenta ao estudo do controlo das trajetórias planeadas.

Como é demasiado difícil testar diferentes tipos de controlo no modelo real, conclui-se que é de grande valia ter um modelo em ambiente de simulação que forneça uma boa estimativa do comportamento do robô no mundo físico. Porém, apesar de o ambiente de simulação apresentar uma grande versatilidade a testes, seu custo é a inviabilização da implementação de algumas variáveis externas, o que faz com que o projeto simulado leve em consideração apenas parâmetros essenciais.

Ao levar em consideração a importância do controlo adequado para uma boa reprodução de trajetórias a presente obra tem a finalidade de introduzir, via simulação computacional, algoritmos de controlo clássico em um modelo de robô quadrúpede, além de apresentar resultados e dar considerações sobre a aplicabilidade dos mesmos.

## **Palavras-Chave**

Robôs com pernas, Robôs quadrúpedes, Simulação, Simulink™, Controlo, Trajetória, Padrões de locomoção.





---

# Abstract

---

Since it's beginning the mobile robots tries to satisfy all human needs in terms of locomotion, and because of it's efficiency, the used locomotion systems in regular terrains are the wheeled ones. However, as the time passed by, the scientific and technological needs made room for locomotion with legs mainly because it's easier to cross steep and irregular terrains with this kind of robot.

Because of the increasing of complexity of tasks, the reliability of robotic systems needs to be improved constantly. That improvement passes through an efficient control system. In the field of legged robots there are many studies around model's optimization, but a few are concentrated in the control analysis of its trajectory planning.

Due to the difficulty of making control tests in the real world, is interesting to have a simulated model that provides a good approach of a real world model's behavior. Despite the simulation environment provides a good versatility in tests, its cost is a poor implementation of external variables, which makes feasible the simulated model to have only its essential parameters.

The importance of a good control system in trajectory planning inspired this work, which the purpose is to introduce, via computer simulation, classical control algorithms and show its results as well to make a comparison between them.

## Keywords

Legged robots, Quadruped robots, Simulation, Simulink<sup>™</sup>, Control, Trajectory, Locomotion patterns.



---

# Conteúdo

---

Conteúdo	xi
Lista de Figuras	xv
Lista de Tabelas	xix
Glossário	xxi
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	2
1.2 Problema . . . . .	2
1.3 Motivação . . . . .	2
1.4 Objetivos . . . . .	3
1.5 Etapas do projeto . . . . .	3
1.6 Estrutura da dissertação . . . . .	3
<b>2 Locomoção com pernas</b>	<b>5</b>
2.1 Robôs bípedes . . . . .	5
2.2 Robôs quadrúpedes . . . . .	7
2.2.1 ANYmal . . . . .	7
2.2.2 AIBO . . . . .	9
2.2.3 Super Mini Cheetah (SMC) . . . . .	9
2.2.4 Cheetah Cub . . . . .	10
2.3 Robôs hexápodes . . . . .	12
<b>3 Padrões de locomoção</b>	<b>15</b>
3.1 Introdução . . . . .	15
3.2 Padrões de locomoção periódicos . . . . .	17
3.2.1 Caminhar . . . . .	19
3.2.2 Trote . . . . .	20
3.2.3 Galope . . . . .	22

3.3	Padrões de locomoção não-periódicos . . . . .	22
<b>4</b>	<b>Modelo do Robô Quadrúpede</b>	<b>25</b>
4.1	Introdução . . . . .	25
4.2	Modelo matemático do robô . . . . .	25
4.3	Modelação com SimMechanics™ . . . . .	29
4.3.1	Corpo do robô . . . . .	29
4.3.2	Pernas do robô . . . . .	31
4.3.3	Contacto com o solo . . . . .	33
4.3.3.1	Força tangencial . . . . .	34
4.3.3.2	Força normal . . . . .	34
<b>5</b>	<b>Desenvolvimento do Trabalho</b>	<b>37</b>
5.1	Planeamento de trajetórias . . . . .	37
5.2	Estudo da cinemática . . . . .	39
5.2.1	Cinemática direta . . . . .	39
5.2.2	Cinemática inversa . . . . .	40
5.3	Linearização do modelo . . . . .	42
5.3.1	Análise linear . . . . .	43
5.3.2	Espaço de estados . . . . .	45
5.4	Sistema de controlo . . . . .	47
5.4.1	Implementação do sistema de controlo no Simulink™ . . . . .	48
5.4.2	Estimação de parâmetros via <i>Control System Designer</i> . . . . .	49
<b>6</b>	<b>Testes e resultados</b>	<b>53</b>
6.1	Metodologia dos testes . . . . .	53
6.1.1	Índices de erro . . . . .	54
6.1.2	Índices de rendimento . . . . .	55
6.2	Resultados obtidos . . . . .	55
6.2.1	Considerações gerais . . . . .	55
6.2.2	Resultados dos controladores . . . . .	56
6.2.3	Resultados da saturação . . . . .	59
6.2.4	Resultados do padrão de locomoção caminhar . . . . .	61
6.3	Discussão dos resultados . . . . .	65
<b>7</b>	<b>Conclusão e desenvolvimentos futuros</b>	<b>67</b>
7.1	Balanço da obra . . . . .	67
7.2	Conclusão . . . . .	68
7.3	Desenvolvimentos futuros . . . . .	68
	<b>Bibliografia</b>	<b>69</b>

<b>A</b>	<b>Equação obtida dos modelos das ancas e joelhos em espaço de estados</b>	<b>75</b>
A.1	Espaço de estados das ancas . . . . .	75
A.2	Espaço de estado dos joelhos . . . . .	77
<b>B</b>	<b>Código para obtenção dos índices resultantes da simulação</b>	<b>79</b>
<b>C</b>	<b>Código para inicialização de parâmetros do modelo em Simulink™</b>	<b>83</b>



---

# Lista de Figuras

---

2.1	Alguns modelos de robôs bípedes. . . . .	6
2.2	Exemplo de ZMP estável e instável. . . . .	7
2.3	Robô quadrúpede ANYmal [13]. . . . .	8
2.4	Diagrama de blocos do controlador PID do robô ANYmal [13]. . . . .	8
2.5	Robô quadrúpede AIBO da Sony [14]. . . . .	9
2.6	Robô quadrúpede Super Mini Cheetah do MIT [18]. . . . .	10
2.7	Esquema dos motores nas patas do SMC [18]. . . . .	10
2.8	Máquina de estados de cada pata do SMC [18]. . . . .	11
2.9	Robô quadrúpede Cheetah Cub [20]. . . . .	11
2.10	Robô hexápode DLR Crowler em equilíbrio com 5 patas [22]. . . . .	12
2.11	Alguns modelos de robôs hexápodes. . . . .	13
2.12	Robô hexápode Lauron V [23]. . . . .	14
2.13	Detalhe da pata do robô hexápode Lauron V [23]. . . . .	14
3.1	Estudo de Eadweard Muybridge sobre os padrões de locomoção de um cavalo [28]. . . . .	16
3.2	Sequência de movimentos possíveis do padrão de locomoção descontínuo para robôs quadrúpedes. As setas indicam a ordem de movimentação das patas [27]. . . . .	16
3.3	Definição da margem de estabilidade: a) Margem de estabilidade $S_m$ ; b) Margem longitudinal de estabilidade $S_{lm}$ [29]. . . . .	17
3.4	Disposição de camadas do processo de locomoção em um sistema robótico baseado em CPG. . . . .	18
3.5	Sequência de movimentos do caminhar de um cavalo [33]. . . . .	20
3.6	Gráfico de movimentos do padrão caminhar [27]. . . . .	20
3.7	Sequência de movimentos do trote de um cavalo [33]. . . . .	21
3.8	Gráfico de movimentos do padrão trote [27]. . . . .	21
3.9	Sequência de movimentos do galope de um cavalo [33]. . . . .	22
3.10	Gráfico de movimentos do padrão galope [27]. . . . .	23



3.11	Esquema de uma perna espacialmente desacoplada [36]. . . . .	24
4.1	Modelo de robô quadrúpede desenvolvido por Castro [37]. . . . .	26
4.2	Diagramas de fluxo da cinemática e dinâmica de um robô. . . . .	26
4.3	Problema de múltiplas soluções da cinemática inversa. . . . .	27
4.4	Decomposição de forças pé-solo no modelo. . . . .	28
4.5	Aparência dos blocos utilizados no modelo. . . . .	31
4.6	Diagrama simplificado do modelo do robô no SimMechanics™. . . . .	32
4.7	Diagrama do corpo do robô no SimMechanics™. . . . .	32
4.8	Diagrama da perna do robô no SimMechanics™. . . . .	33
4.9	Diagrama de contacto com o solo do robô no SimMechanics™. . . . .	34
4.10	Subsistema do bloco <i>friction_force</i> . . . . .	35
4.11	Bloco <i>Friction Force</i> da força tangencial. . . . .	35
4.12	Subsistema do bloco <i>normal_force</i> . . . . .	36
4.13	Bloco <i>Normal Force</i> da força normal. . . . .	36
5.1	Trajetória planeada aplicada ao modelo do robô. . . . .	39
5.2	Disposição dos ângulos para o cálculo da cinemática do modelo. . . . .	40
5.3	<i>Trim model</i> na ferramenta <i>Linear Analysis</i> . . . . .	43
5.4	Seleção dos estados das variáveis em <i>trim model</i> . . . . .	44
5.5	Especificação do <i>trim model</i> obtida com sucesso. . . . .	44
5.6	Seleção de <i>I/O Ports</i> em <i>Linear Analysis</i> . . . . .	45
5.7	Resposta ao degrau do sistema linearizado a partir da anca. . . . .	45
5.8	Resposta ao degrau do sistema linearizado a partir do joelho. . . . .	46
5.9	Diagrama completo do sistema de controlo. . . . .	48
5.10	Sistema de controlo implementado em Simulink™. . . . .	49
5.11	Implementação da arquitetura do controlador na simulação através do bloco <i>LTI System</i> . . . . .	50
5.12	Arquitetura do controlador selecionada em <i>Control System Designer</i> . . . . .	50
5.13	Seleção da ferramenta de sintonia em <i>Control System Designer</i> . . . . .	51
5.14	Ferramenta <i>PID Tuning</i> . . . . .	51
5.15	Janela da ferramenta <i>Compensator Editor</i> . . . . .	52
6.1	Valores de $E_{av}$ multiplicados por $\%I_{ae}$ para cada teste. . . . .	56
6.2	Valores de $E_{av}$ multiplicados por $\%I_{se}$ para cada teste. . . . .	57
6.3	Distâncias percorridas pelo robô para cada teste. . . . .	58
6.4	Medidas de $E_{av}$ do robô nos testes. . . . .	58
6.5	Medidas de $I_{ae}$ do robô nos testes. . . . .	58
6.6	Medidas de $I_{se}$ do robô nos testes. . . . .	59
6.7	Distância percorrida pelo robô durante simulação com $V_F = 0,05$ $\text{ms}^{-1}$ e $t = nT_0 = 20$ s. . . . .	60

6.8	Binários aplicados nas pernas do robô durante simulação para uma saturação de 10 Nm. . . . .	61
6.9	Trajetórias das juntas do robô para uma saturação de 10 Nm. . . . .	62
6.10	Simulação do robô no padrão de locomoção trote adaptado. . . . .	63
6.11	$E_{av}$ do sistema para diferentes valores de saturação. . . . .	63
6.12	$I_{ae}$ do sistema para diferentes valores de saturação. . . . .	64
6.13	Simulação do robô no padrão de locomoção caminhar. . . . .	65



---

# Lista de Tabelas

---

1.1	Calendarização da obra no ano de 2017. . . . .	4
2.1	Comparação de modelos de robôs bípedes . . . . .	6
3.1	Comparação entre as marchas contínua e descontínua dos padrões de locomoção periódicos . . . . .	19
4.1	Descrição dos blocos do modelo. . . . .	30
5.1	Parâmetros de linearização por <i>trimming</i> do modelo. . . . .	43
6.1	Parâmetros dos testes do controlador. . . . .	56
6.2	Valores das medidas realizadas no teste 5. . . . .	57
6.3	Índices de erros nas juntas do robô para vários valores de saturação. . . . .	59
6.4	Aspeto geral do sistema para vários valores de saturação. . . . .	59
6.5	Índices de erros para padrão de locomoção caminhar. . . . .	64
6.6	Aspeto geral do sistema para padrão de locomoção caminhar. . . . .	64



---

# Glossário

---

Abreviatura	Descrição	Definição
1G	<i>First Generation</i>	página 30
CPG	<i>Central Pattern Generator</i>	página 11
DARPA	<i>Defense Advanced Research Projects Agency</i>	página 9
EAV	<i>Mean Absolute Density of Energy per Travelled Distance</i>	página 55
EPFL	<i>École Polytechnique Federale de Lausanne</i>	página 7
ETH	<i>Eidgenössische Technische Hochschule</i>	página 7
GDL	Graus de Liberdade	página 6
GND	<i>Ground</i>	página 29
I/O	<i>Input/Output</i>	página 45
IAE	<i>Integral of Absolute Error</i>	página 54
ISE	<i>Integral of Squared Error</i>	página 54
ITAE	<i>Integral of Time Multiplied by Absolute Error</i>	página 54
ITSE	<i>Integral of Time Multiplied by Squared Error</i>	página 54
LED	<i>Light Emitter Diode</i>	página 9
LTI	<i>Linear Time Invariant</i>	página 49
MIT	<i>Massachussets Institute of Technology</i>	página 7
PCB	<i>Printed Circuit Board</i>	página 10
PD	Proporcional Derivativo	página 29
PDF	Proporcional Derivativo com Filtro	página 48
PID	Proporcional Integrativo Derivativo	página 8
ROS	<i>Robot Operational System</i>	página 8
SDK	<i>Software Development Kit</i>	página 9
SISO	<i>Single Input Single Output</i>	página 46
SMC	<i>Super Mini Cheetah</i>	página 9
ZMP	<i>Zero-Moment Point</i>	página 6



# Capítulo 1

---

## Introdução

---

A robótica móvel é uma importante ferramenta no panorama atual da sociedade, pois muitos campos da ciência dependem de instrumentos capazes de se locomover em lugares inóspitos para os seres humanos. Para isso, é necessário criar um meio de locomoção capaz de suportar diversos tipos de adversidades, tais como temperatura, pressão, terrenos íngremes e irregulares, etc. Os robôs móveis terrestres mais comuns se dividem em locomoção por rodas e locomoção por pernas. Embora a locomoção por rodas por vezes seja mais eficiente para terrenos regulares e ambientes interiores, a locomoção por pernas se torna mais robusta nas situações em que obstáculos surgem no trajeto [1].

Os robôs quadrúpedes são um sub-grupo dos robôs móveis de locomoção por pernas, biologicamente inspirados na anatomia de animais de quatro patas como chitas, cavalos, cães, etc. Embora fatores como a eficiência energética, estabilidade, velocidade, mobilidade ainda estejam muito aquém do ideal, os robôs quadrúpedes desempenham uma boa aproximação do modelo biológico destes animais quando os padrões de locomoção adequados são implementados [2].

Em comparação aos outros tipos de robôs móveis por pernas, os quadrúpedes levam vantagem em questão de estabilidade frente aos robôs bípedes, e em comparação aos robôs hexápodes e octópodes, possuem maior eficiência energética embora tenham menor estabilidade [11, 12, 13].



## 1.1 Contextualização

Esta obra abordará a modelação e implementação de diferentes técnicas de controlo de um robô quadrúpede em ambiente Simulink™, uma ferramenta associada ao MATLAB® responsável por simular parâmetros eletrónicos, lógicos e mecânicos de diversos tipos de projeto. O trabalho abordará as limitações do controlo quanto a eficiência energética, erro e desempenho do robô, bem como a implementação de padrões de locomoção inspirados na locomoção de animais quadrúpedes.

O robô utilizado é baseado em um modelo do autor Castro [37] contendo massa, densidade, inércia e tamanhos adequados a um robô real. O robô foi inteiramente modelado em SimMechanics™, uma *toolbox* do ambiente MATLAB®/Simulink™ que é responsável pela mecânica da simulação.

## 1.2 Problema

A problemática que envolve o controlo dos sistemas móveis está na aplicação adequada do mesmo de acordo com as especificações a serem alcançadas. Definir qual a taxa de amostragem, qual o controlador a ser utilizado e seus parâmetros, qual a fiabilidade que o padrão de locomoção requer do controlador, a eficiência energética e o dimensionamento dos atuadores e sensores são esforços necessários e que nem sempre são de solução trivial.

Integrar um sistema de controlo a um modelo pré-existente requer a total compreensão dos parâmetros mecânicos do mesmo, bem como a alteração de alguns desses parâmetros visando a adequação a um ambiente mais próximo da realidade. Além do mais, como é demasiado difícil a obtenção e controlo do modelo matemático completo do robô devido a este ter carácter não-linear, foi feita a linearização do sistema a fim de obter um sistema que seja fiel e que aceite a implementação das teorias clássicas de controlo.

## 1.3 Motivação

Embora um robô real difira do implementado na simulação e na maioria dos casos seja mais complexo, o ambiente computacional permite efetuar testes e dar uma boa estimativa de como devem ser dimensionadas suas características iniciais sem a necessidade de um modelo físico. Na parte de controlo geralmente é necessário fazer ajustes finais antes da implementação física, pois existem diversas variáveis que se alteram do modelo computacional para o modelo real que são difíceis de serem implementadas em *software*. Porém, como dito anteriormente, encontrar uma primeira aproximação e suas estatísticas de desempenho requer menos esforço nos testes de campo do modelo.

## 1.4 Objetivos

O objetivo principal da obra é apresentar a simulação de um robô quadrúpede em MATLAB<sup>®</sup>/Simulink<sup>™</sup> que satisfaça o padrão de locomoção implementado através de ferramentas de controlo disponíveis e comparar os resultados obtidos para diversos parâmetros do controlador.

Para que o objetivo principal fosse atingido, foi necessário desenvolver as seguintes etapas no projeto:

- Equacionar a cinemática direta e inversa referente ao movimento das pernas do robô;
- Desenvolver o planeamento de trajetórias baseado no algoritmo de uma ciclóide para a fase de transferência da perna;
- Estudar os padrões de locomoção a fim de especificar a sequência de movimento das pernas do robô para os diferentes tipos de padrão de locomoção;
- Encontrar uma maneira adequada de linearizar o modelo;
- Definir o tipo de controlador a ser utilizado em cada um dos atuadores considerando as limitações do modelo linear;
- Implementar o algoritmo de controlo utilizando ferramentas do MATLAB<sup>®</sup>/Simulink<sup>™</sup>;
- Obter medidas de desempenho do sistema de controlo projetado;
- Analisar e comparar os resultados para os diversos padrões de locomoção.

## 1.5 Etapas do projeto

Visando cumprir os prazos determinados e cumprir todos os tópicos propostos anteriormente, a calendarização da obra fica dividida em dez etapas no período que se refere de fevereiro a julho de 2017, como sugere a Tabela 1.1.

## 1.6 Estrutura da dissertação

Para uma melhor organização do conteúdo da obra optou-se por dividi-la em sete capítulos: Introdução, Locomoção com Pernas, Padrões de Locomoção, Modelo do Robô Quadrúpede, Desenvolvimento do Trabalho, Testes e Resultados e por último Conclusão e Desenvolvimentos Futuros.

Na introdução é abordado brevemente o panorama da área, o contexto no qual a obra se enquadra, a problemática, a motivação, os objetivos que se pretendem atingir, bem como o planeamento do projeto durante a sua fase de execução.

Tabela 1.1: Calendarização da obra no ano de 2017.

Calendarização da obra						
	Fev.	Mar.	Abr.	Mai.	Jun.	Jul.
Definição do tema e da proposta	<b>X</b>					
Revisão bibliográfica de sistemas robóticos multi-pernas		<b>X</b>	<b>X</b>			
Revisão e estudo do modelo mecânico em SimMechanics™			<b>X</b>	<b>X</b>		
Estudo cinemático, planeamento de trajetórias e padrão de locomoção do modelo				<b>X</b>		
Implementação das ferramentas de controlo do Simulink™				<b>X</b>		
Simulação e testes				<b>X</b>	<b>X</b>	
Aquisição de resultados e conclusão					<b>X</b>	
Elaboração do relatório						<b>X</b>

No capítulo de locomoção com pernas é abordado um panorama geral sobre robôs bípedes, quadrúpedes e hexápodes. Também são apresentados alguns exemplos comerciais desses tipos de robôs bem como suas principais características.

O terceiro capítulo faz a análise dos padrões de locomoção de animais visando sua implementação no modelo. É feito o estudo da locomoção de animais quadrúpedes, em especial o trote, galope e caminhar de um cavalo para padrões de locomoção periódicos, e uma abordagem breve e geral sobre os padrões de locomoção não-periódicos.

Uma descrição das propriedades mecânicas, equações matemáticas e parâmetros do modelo de Castro [37] utilizado na obra é feito no capítulo quatro. O capítulo divide o modelo em blocos que abordam o aspeto do corpo, das pernas e dos atuadores e a interação pé-solo do mesmo.

A teoria de controlo utilizada, formulações matemáticas, uso das ferramentas do MATLAB®/Simulink™, aquisição das funções de transferência, parâmetros dos controladores, cinemática direta e inversa e planeamento de trajetória serão abordados no capítulo cinco.

A descrição dos testes realizados na obra, gráficos, resultados, parâmetros e demais formulações utilizadas para medir o desempenho do modelo em diferentes cenários será apresentado no capítulo seis.

A conclusão será feita no capítulo sete mediante a análise dos resultados obtidos com a simulação, e face ao previamente apresentado propor-se-á como trabalhos futuros temas que complementem a obra.

## Capítulo 2

---

# Locomoção com pernas

---

*Neste capítulo serão apresentados os tipos de robôs móveis de acordo com sua quantidade de pernas e suas principais características físicas. Serão também apresentados os paradigmas atuais da área bem como alguns dos principais modelos existentes até à data de publicação desta obra.*

### 2.1 Robôs bípedes

Os robôs bípedes são uma gama de robôs que se movem e se equilibram sobre duas pernas. Existem vários modelos de robôs bípedes em desenvolvimento desde o século passado, e em sua grande maioria, são desenvolvidos modelos humanóides que simulam o comportamento humano. Este tipo de robô é tipicamente utilizado para interação com humanos, para a realização de tarefas em ambientes internos e para fins de entretenimento [3].

O projeto de um robô bípede humanóide geralmente conta com vários atuadores e controladores, podendo ter dezenas de graus de liberdade a fim de simular todas as articulações de um indivíduo, o que torna o sistema muito complexo, mas também capaz de realizar diversas tarefas humanas com alta precisão [3]. Além disso, para trabalhar em ambientes utilizados por humanos, o robô precisa ter sensores capazes de detectar obstáculos e planejar uma rota para evitar a colisão. Sensores de captura de som e imagem também são necessários para a comunicação com pessoas e para a identificação de objetos, respectivamente [4].

Alguns dos robôs bípedes atualmente utilizados para realizar tarefas humanas são o ASIMO da Honda, SDR4X-II da Sony e REEM-B da Pal-Robotics. Além disso existem os modelos humanóides realistas utilizados para demonstração e entretenimento, como exemplo o HRP-4C. A Figura 2.1 ilustra estes modelos.

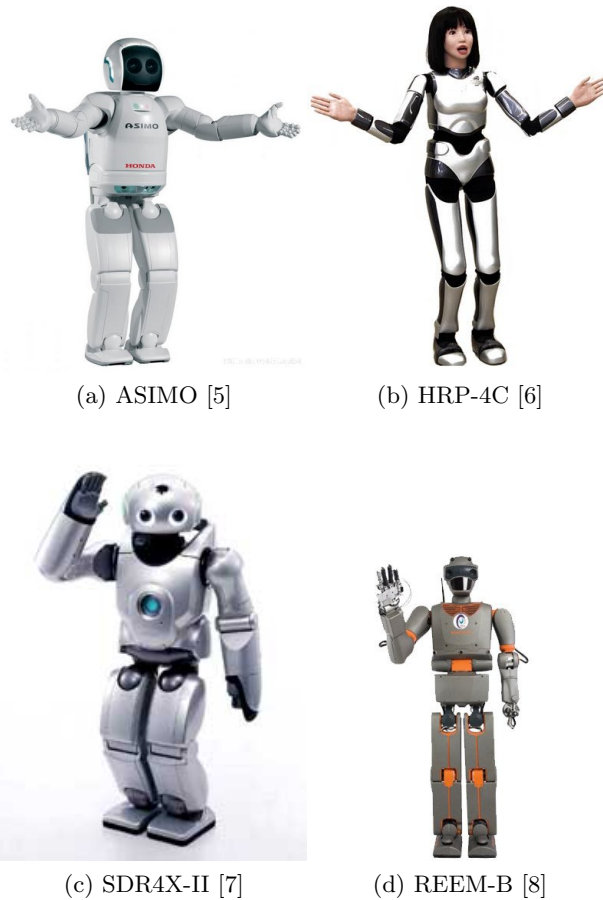


Figura 2.1: Alguns modelos de robôs bípedes.

De entre os modelos listados existem variações de tamanho, peso, *hardware* e autonomia de bateria, como mostrado na Tabela 2.1.

Tabela 2.1: Comparação de modelos de robôs bípedes

	ASIMO	HRP-4C	REEM-B	SDR4X-II
<b>GDL</b>	57	42	40	38
<b>Peso</b>	50 kg	43 kg	60 kg	7,3 kg
<b>Altura</b>	1,3 m	1,58 m	1,47 m	0,6 m
<b>Autonomia</b>	1 h	20 min	2 h	1 h
<b>Sensor ZMP</b>	6 eixos	6 eixos	6 eixos	sensor de força x 4

Porém, uma das premissas fundamentais na construção de robôs bípedes é o controle de trajetória baseado em *Zero Moment Point* (ZMP), o que assegura seu equilíbrio e sua locomoção adequada [9]. O ZMP é o ponto em relação ao polígono de suporte do robô no qual o momento é zero. Para que o robô consiga realizar

a sua trajetória sem quedas, é necessário garantir que este ponto esteja sempre localizado em algum ponto do suporte do robô, preferencialmente no centro para que a estabilidade seja máxima [10]. A Figura 2.2 mostra um exemplo de ZMP estável e instável.



Figura 2.2: Exemplo de ZMP estável e instável.

## 2.2 Robôs quadrúpedes

Os robôs quadrúpedes por sua vez contam com quatro patas responsáveis pela sua locomoção. O grande diferencial frente aos bípedes é a estabilidade em terrenos irregulares, a possibilidade de realizar diferentes marchas e a sua eficiência energética [11, 12, 13]. Porém, a dificuldade em simular de maneira adequada as marchas é um desafio para a área de pesquisa desse tipo de robô. Muitos sistemas de controle são desenvolvidos na tentativa de aproximar os modelos ao tipo de marcha realizado por animais quadrúpedes, tais como cavalos, cães, onças, entre outros [14]. Devido à sua facilidade em atravessar obstáculos, uma das aplicações possíveis para os quadrúpedes é o patrulhamento de áreas onde ocorreram desastres naturais, bem como a monitorização de ambientes hostis [15].

Esforços são realizados constantemente pelas equipas de pesquisa para melhorar este tipo de robô e dar-lhe uma aplicação prática no mundo real. Alguns casos de sucesso são os robôs ANYmal da ETH Zurich, Cheetah-Cub da EPFL (École Polytechnique Federale de Lausanne), Super Mini Cheetah do MIT (Massachusetts Institute of Technology) e o robô AIBO da Sony. Todos estes modelos citados serão discutidos a seguir e terão suas principais características apresentadas.

### 2.2.1 ANYmal

Segundo Hutter et. al. [13], o robô ANYmal foi projetado para trabalhar por grandes períodos de tempo em ambientes nocivos. Seu foco é uma grande mobilidade, velocidade e dinâmica de locomoção, além de robustez, navegação segura e manutenção simples. Para a locomoção, cada pata conta com três graus de liberdade e um sensor de força na extremidade em contacto com o solo. Sua distância entre patas é de 250 mm e pesa aproximadamente 30 kg, o que é muito



### 2.2.2 AIBO

Em 1999 a Sony apresentou o robô AIBO (Figura 2.5), no qual foi o primeiro robô disponível comercialmente com autonomia considerável, uma grande gama de sensores, de uso fácil pelo consumidor, e *software* principal livre e programável [17]. Zhang et. al. [16] descreve o AIBO como um robô quadrúpede equipado com comunicação *wireless*, câmara a cores, LEDs, sensor de distância, sensor de aceleração, vários sensores de toque e 20 graus de liberdade para realizar as mais variadas tarefas voltadas ao entretenimento. O robô AIBO conta ainda com a plataforma multi-programável SDK (*Software development kit*) para que o usuário possa introduzir a marcha desejada e demais funcionalidades ao robô.

Desde seu lançamento a Sony buscou melhorar o *hardware* e o *design* do AIBO ao providenciar melhores soluções de *software* e versatilidade, o que por consequência gerou uma grande variedade de artigos de pesquisa na área de controle da locomoção com este robô [17].



Figura 2.5: Robô quadrúpede AIBO da Sony [14].

### 2.2.3 Super Mini Cheetah (SMC)

A busca do MIT por um robô quadrúpede de baixo custo resultou no desenvolvimento do Super Mini Cheetah (Figura 2.6), um protótipo leve com locomoção dinâmica capaz de realizar diversos tipos de marchas. Segundo Bosworth [18], o projeto foi desenvolvido para a *DARPA Robotics Challenge* e devido à proporção de seu custo, é o único capaz de realizar com precisão o controle de força e impedância entre cada pata e o solo, bem como providenciar força e movimento suficiente para diversos tipos de comportamentos dinâmicos.

Caminhadas, corridas, pulos, mudança de direção e frenagens são as diferentes tarefas que o SMC pode executar. Para isso, o robô conta com patas de dois motores em paralelo que efetuam uma abertura similar à de um losango, além de





Figura 2.6: Robô quadrúpede Super Mini Cheetah do MIT [18].

um sensor de força na extremidade, como ilustra a Figura 2.7. A configuração de motores é feita de maneira a minimizar o binário necessário para que o motor suporte sua carga eletrônica, que conta com uma PCB para seus oito controladores de corrente e duas baterias 12 células de 44,4 V nominal. O robô tem um peso final de aproximadamente 9 kg, comprimento de 30 cm, largura de 18 cm (entre as patas), e altura variável de 10-20 cm.

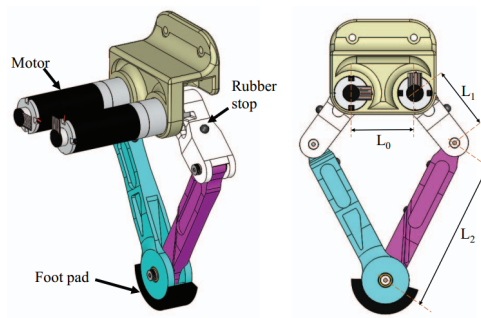


Figura 2.7: Esquema dos motores nas patas do SMC [18].

O controlo da impedância para caminhada e mudança de direção é feito utilizando máquinas de estado independentes para cada pata. A máquina de estados segue um ciclo pré-definido temporizado ou condicional, como sugere a Figura 2.8. O controlo da força vertical foi aplicado para gerar o pulo e a corrida do SMC, e a mudança de direção é feita ao gerar um binário quando se move as patas laterais em diferentes sentidos.

#### 2.2.4 Cheetah Cub

O Cheetah Cub (Figura 2.9) é um robô quadrúpede de pesquisa do tamanho de um gato de estimação, pesa aproximadamente 1 kg, tem 21 cm de comprimento

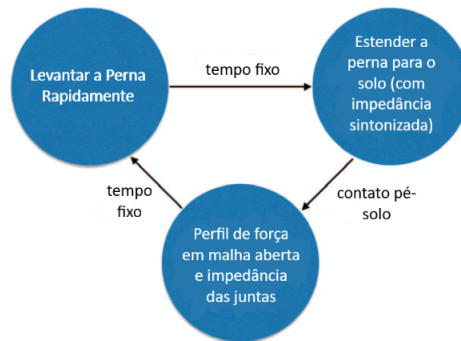


Figura 2.8: Máquina de estados de cada pata do SMC [18].

e pode alcançar a velocidade de 1,42 m/s, o que faz dele o robô mais rápido na categoria de menos de 30 kg. Conta com 8 graus de liberdade e tem como marcha característica o trote. Seus principais atributos são a estabilidade em uma variedade de velocidades utilizando controlo em malha aberta, robustez, custo e segurança no manuseio [20].

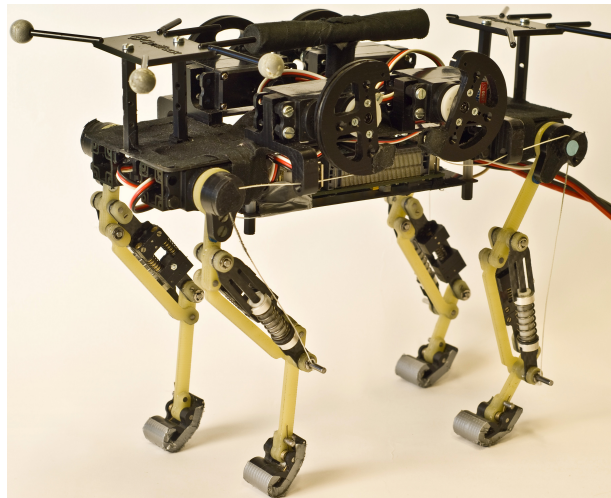


Figura 2.9: Robô quadrúpede Cheetah Cub [20].

Rutishauser et. al. [19] propõe um sistema de controlo para o Cheetah Cub baseado em CPG (*Central Pattern Generator*), uma técnica que simula comportamentos biológicos da espinha dorsal de animais quadrúpedes em robôs. A técnica apresenta vantagens interessantes quando comparada aos controladores convencionais, tais como:

- Troca de marchas mais suave;

- Facilidade na integração do *feedback* de controle;
- Perturbações são rapidamente corrigidas pelo controle;
- Criar diferentes marchas através da capacidade de sincronização dos motores pelo CPG;
- Permite a adaptação do robô ao ambiente;
- A simples troca de parâmetros permite ao robô modular suavemente sua locomoção.

### 2.3 Robôs hexápodes

Os robôs hexápodes são projetados para se moverem em terrenos desconhecidos e irregulares, e assim como os quadrúpedes têm como característica uma melhor estabilidade frente os robôs bípedes. Ao acrescentar um par de pernas, diminui-se a sua eficiência para tarefas que na maioria dos casos os quadrúpedes já realizam.

O grande diferencial dos robôs de 6 patas é justamente quando ocorre a falha em uma das patas, ou seja, para ambientes nocivos aos robôs e que se deseja uma operação contínua mesmo com avarias, os robôs hexápodes são uma boa solução. Neste caso, é importante garantir no sistema de controle que o padrão de locomoção mantém o centro de gravidade dentro do polígono de suporte para manter sua estabilidade [21]. Ao ocorrer uma falha de alguma pata, o sistema de controle do robô hexápode pode alterar o padrão de locomoção para continuar a andar, ao custo de perda de velocidade e desvio angular da trajetória a ser efetuada [22]. A Figura 2.10 mostra a simulação do robô DLR Crowler em equilíbrio com a pata dianteira esquerda desabilitada.

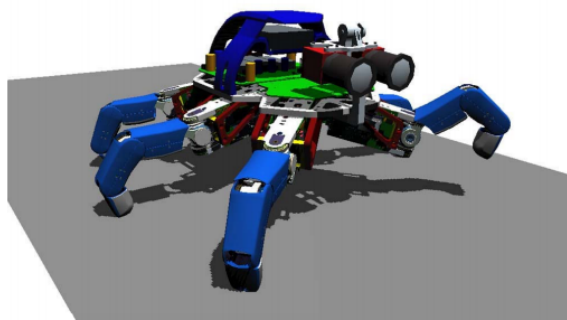
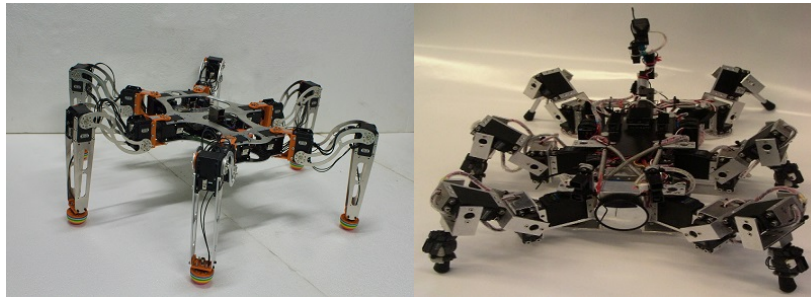


Figura 2.10: Robô hexápode DLR Crowler em equilíbrio com 5 patas [22].

Segundo Roennau et. al. [23], os padrões de locomoção e o modelo de um robô hexápode geralmente são baseados nas características biológicas de insetos. As patas geralmente contam com três graus de liberdade cada e seguem a configuração *yaw-pitch-pitch*, o que no total fornece 18 graus de liberdade para a locomoção do robô, o suficiente para a maioria dos terrenos. Messor II [24], Anton [25] e AMOS-WD06 [26] (Figura 2.11) são alguns exemplos citados na literatura.



(a) MessorII [24]

(b) AMOS-WD06 [26]



(c) Anton [25]

Figura 2.11: Alguns modelos de robôs hexápodes.

Adicionalmente, existem robôs com mais graus de liberdade nas patas com o objetivo de executar alguma tarefa extra, tal como manipulação de objetos. Raramente são projetados mais de três graus de liberdade para a locomoção. Caso à parte é o robô LAURON V (Figura 2.12), projetado com 4 graus de liberdade nas patas.

Além do tradicional *yaw-pitch-pitch*, é adicionado uma quarta junta rotacional no eixo *roll* do LAURON V que garante a diminuição do *stress* das juntas ao escalar algum obstáculo e o melhor balanceamento de cargas, e o deixa mais eficiente em termos de consumo, estabilidade e velocidade [23]. A Figura 2.13 mostra o esquema descrito.

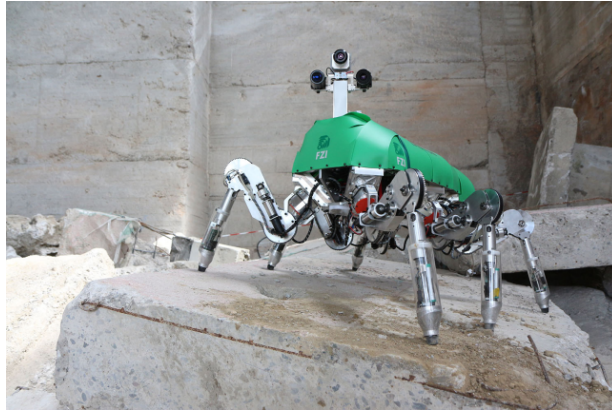


Figura 2.12: Robô hexápode Lauron V [23].

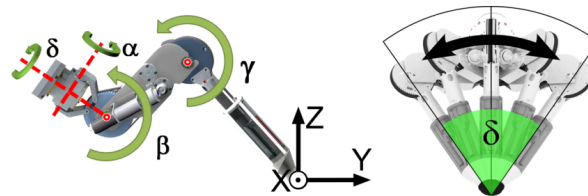


Figura 2.13: Detalhe da pata do robô hexápode Lauron V [23].

## Capítulo 3

---

# Padrões de locomoção

---

*Esse capítulo apresenta alguns dos principais padrões de locomoção para robôs quadrúpedes juntamente com suas principais vantagens, aplicações e modelos matemáticos.*

### 3.1 Introdução

Os primeiros estudos sobre os padrões de locomoção foram realizados por Eadward Muybridge ao observar o comportamento das marchas de mais de quarenta tipos de mamíferos, buscando reconhecer padrões e descrevê-los matematicamente para uma futura análise. Seu estudo inicial sobre os padrões de locomoção de um cavalo (Figura 3.1) foi originalmente publicado pela revista científica *Scientific American* em 1878. Porém, apenas a partir de 1961 os padrões de locomoção começaram efetivamente a fazer parte do escopo dos robôs móveis graças ao trabalho de Tomovic e Karplus [27].

A sequência das diferentes fases de movimento são descritas pelo padrão de locomoção. É o padrão de locomoção que descreve o movimento coletivo das pernas, definindo o instante de elevação e o local de colocação de cada pata, coordenando simultaneamente estas ações com o movimento do corpo nos seus graus de liberdade. Por consequência, os padrões de locomoção também determinam a velocidade, direção do movimento e mobilidade de um animal ou máquina locomotora.

Como já dito no capítulo anterior, os robôs quadrúpedes têm uma maior estabilidade frente aos bípedes. Por esse motivo, uma importante premissa para o projeto de padrões de locomoção é garantir a estabilidade do robô nos mais variados tipos de terrenos. Os quadrúpedes têm estabilidade estática para padrões

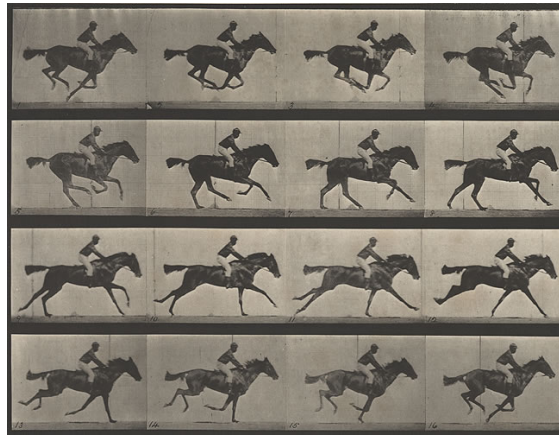


Figura 3.1: Estudo de Eadweard Muybridge sobre os padrões de locomoção de um cavalo [28].

de locomoção de baixa velocidade e estabilidade dinâmica para locomoção de média e alta velocidade.

Para garantir a estabilidade estática de um quadrúpede, é necessário que apenas uma pata esteja em trânsito por vez. Isso limita os padrões de locomoção periódicos de um robô a seis tipos de movimento [27], como ilustrado na Figura 3.2. Tal padrão de locomoção é denominado por Silva [30] como padrão de locomoção descontínuo, e apresenta grandes vantagens quanto à sua implementação em sistemas robóticos móveis.

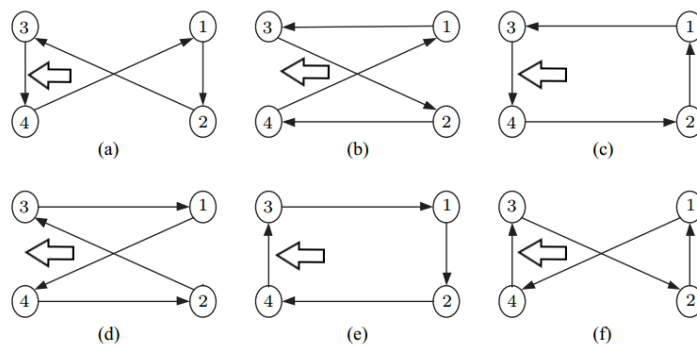


Figura 3.2: Sequência de movimentos possíveis do padrão de locomoção descontínuo para robôs quadrúpedes. As setas indicam a ordem de movimentação das patas [27].

Segundo Shuuji e Bernard [29], existem duas formas de se medir a estabilidade do robô baseado no estudo do seu centro de massa: Margem de estabilidade  $S_m$

e margem longitudinal de estabilidade  $S_{lm}$ . Ambas são definidas pelo autor da seguinte maneira:

- A margem de estabilidade  $S_m$ , definida como a menor distância entre o centro de massa e a  $i$ -ésima parte da borda do polígono;
- A margem longitudinal de estabilidade  $S_{lm}$  é a menor distância entre o centro de massa e a borda do polígono paralela à direção do movimento do robô.

A aplicação do controlo de estabilidade depende da necessidade do projeto. A Figura 3.3 mostra as diferenças entre os tipos de margem  $S_m$  e  $S_{lm}$ .

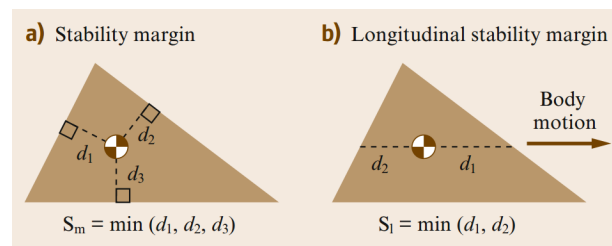


Figura 3.3: Definição da margem de estabilidade: a) Margem de estabilidade  $S_m$ ; b) Margem longitudinal de estabilidade  $S_{lm}$  [29].

Silva [30] e Barasuol [27] afirmam que existem dois grupos de padrões de locomoção: periódicos e não-periódicos. A ideia de diferenciar estes dois grupos partiu da necessidade de adaptação do controlo dos robôs ao tipo de terreno no qual serão sujeitos, e têm grande aplicação nos dias atuais. Baseado principalmente no estudo destas duas referências, será apresentada uma breve explanação sobre esses tipos de padrões nas próximas secções.

## 3.2 Padrões de locomoção periódicos

Os padrões de locomoção periódicos são assim denominados por apresentarem uma sequência ordenada e de igual período para a movimentação de cada uma das patas do robô. Esse tipo de padrão é inspirado em modelos biológicos de locomoção de mamíferos quadrúpedes, humanos e insetos, pois através de estudos notou-se que embora os padrões de locomoção possam variar muito entre as espécies, há um certo sincronismo no movimento de todos eles. Tais movimentos sincronizados dos animais são executados pelos seus respectivos sistemas nervosos centrais seguindo o estímulo de geradores centrais de padrão (sigla CPG em inglês). Os CPG's são osciladores neuronais agrupados em rede capazes de gerar estímulos repetitivos que podem ser descritos por frequência e fase.



Ainda que distante da agilidade em que os animais realizam as tarefas, os sistemas robóticos tendem a seguir certos aspetos cronológicos que estes animais apresentam. Todo o processo de locomoção do robô acontece em três camadas: Tomada de decisão, atuação, e retroação. Para garantir a locomoção, apenas as duas primeiras camadas apresentadas são necessárias, sendo a terceira apenas uma camada de otimização do modelo de controlo. A Figura 3.4 mostra a disposição das camadas em um diagrama de blocos.

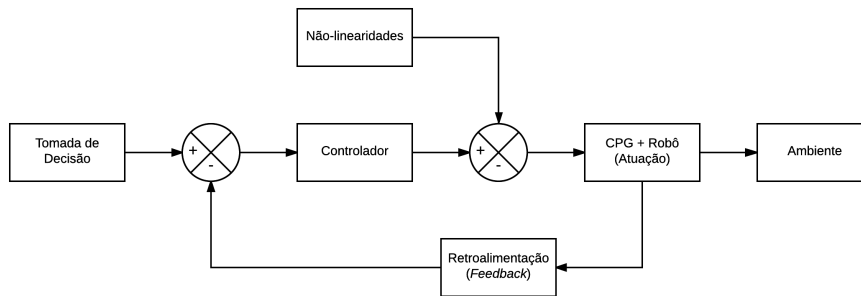


Figura 3.4: Disposição de camadas do processo de locomoção em um sistema robótico baseado em CPG.

A tomada de decisão do robô é crucial para garantir um bom padrão de locomoção. Essa camada assemelha-se com os estímulos neuronais enviados pelo cérebro em animais, e é simulada nos robôs por algoritmos de planeamento de trajetória. Para que os diferentes padrões de locomoção sejam corretamente implementados, é necessária uma boa escolha dos parâmetros do controlador.

A camada de atuação é onde está inserido o CPG. Essa, por sua vez, depende de um sinal rítmico proveniente dos comandos da primeira camada. Tais sinais variam de robô para robô e podem ser constituídos por equações diferenciais, como no caso de [31], ou ainda osciladores não lineares, como é o caso de [32].

A camada de retroação é responsável pelo *feedback* dos movimentos no sistema de controlo. Assim sendo, quando algum sensor de movimento não atinge o patamar desejado, ocorre a correção do movimento até que a leitura se estabilize em um padrão aceitável. Esse tipo de configuração torna o controlo muito mais robusto, adaptável e inteligente, pois o mesmo assume características de malha fechada. Tal sistema assemelha-se com a capacidade de reflexo que têm os animais.

Embora a estrutura de controlo baseada no modelo biológico de animais seja interessante, o facto é que ainda existem desafios a serem superados para esta tecnologia. O principal problema é o grande número de parâmetros a serem ajustados e a não-linearidade da dinâmica do modelo. Torna-se, portanto, di-

fácil correlacionar tais fatores a ponto de produzir um resultado satisfatório de oscilação periódica.

Existem dois subgrupos dentro dos padrões de locomoção periódicos que são largamente abordados pela literatura: A marcha contínua e a marcha descontínua. Ambos diferem um do outro devido à movimentação do corpo do robô.

As marchas contínuas são assim chamadas pois nesse tipo de locomoção o corpo do robô está sempre em movimento e suas pernas movem-se simultaneamente. Observa-se nesse tipo de marcha que a movimentação das patas formam um padrão ondulatório. Geralmente são indicadas para quando uma maior velocidade de locomoção é exigida, e o terreno não têm grandes depressões, vales ou obstáculos.

As marchas descontínuas, por sua vez, têm esse nome devido o corpo do robô permanecer em equilíbrio estático enquanto o robô posiciona sua perna em um novo local. Tal padrão garante ao robô uma maior estabilidade e é altamente indicado em terrenos aonde existem muitas irregularidades. Nesse tipo de locomoção o robô move apenas uma perna de cada vez, enquanto as outras três (no caso de um quadrúpede) permanecem em contacto com o solo com o objetivo de fazer o triângulo de suporte que estabiliza o robô.

A Tabela 3.1 faz a comparação entre os tipos de marcha contínua e descontínua de acordo com os principais requisitos de um sistema robótico móvel.

Tabela 3.1: Comparação entre as marchas contínua e descontínua dos padrões de locomoção periódicos

<b>Padrões de Locomoção Periódicos</b>		
	<b>Marcha Contínua</b>	<b>Marcha Descontínua</b>
<b>Tipo de Terreno ideal</b>	Não acidentado Sem obstáculos	Pequenos e Médios obstáculos
<b>Velocidade de Locomoção</b>	Alta velocidade e Média velocidade	Baixa velocidade
<b>Adaptabilidade</b>	Baixa	Média
<b>Estabilidade</b>	Média	Alta

Os padrões de locomoção abordados nesta secção têm como base o estudo de Muybridge [33] em animais, que apresenta três padrões de locomoção periódicos como sendo os mais frequentemente usados em quadrúpedes: O caminhar, o trote e o galope.

### 3.2.1 Caminhar

O caminhar do quadrúpede é uma marcha de quatro tempos que não apresenta suspensão do corpo do animal. É uma marcha lenta por característica, e muito

utilizada pelos mamíferos quadrúpedes. A Figura 3.5 mostra a imagem do caminhar de um cavalo.

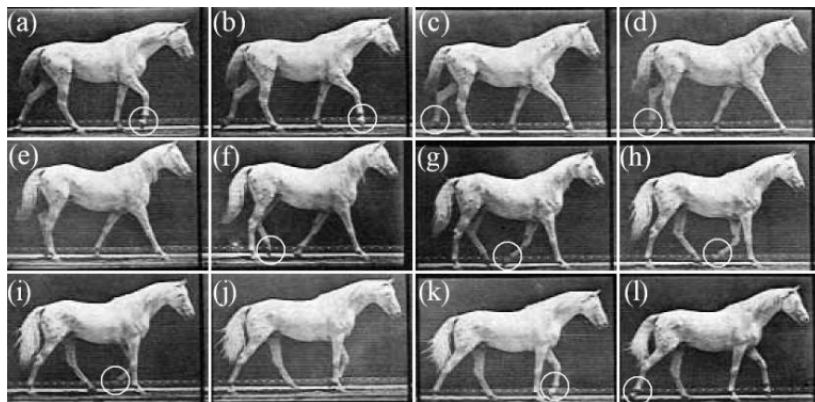


Figura 3.5: Sequência de movimentos do caminhar de um cavalo [33].

A melhor sequência de movimento das pernas para que a marcha garanta o máximo de estabilidade ao caminhar para frente ao longo do eixo longitudinal é 4-2-3-1 [29], com as pernas numeradas de acordo com a Figura 3.2 apresentada previamente. A Figura 3.6 mostra um gráfico que exemplifica a sequência otimizada do caminhar de um quadrúpede. De notar que o período da marcha está normalizado e as linhas contínuas mostram quando a respetiva pata está em contacto com o solo, enquanto as linhas tracejadas indicam que a perna indicada está em suspensão.

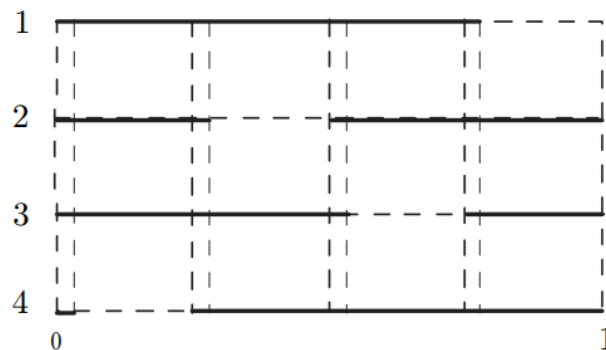


Figura 3.6: Gráfico de movimentos do padrão caminhar [27].

### 3.2.2 Trote

O trote é uma marcha caracterizada pela suspensão de duas patas em simultâneo. É uma marcha de média velocidade, e ao contrário do caminhar, o corpo do

quadrúpede fica levemente suspenso no ar. No trote tradicional estudado por Muybridge, o cavalo suspende uma pata dianteira e uma traseira de lados opostos, como sugere a Figura 3.7. Esse movimento é rítmico e garante a estabilidade do animal durante a execução do movimento, além de proporcionar maior velocidade de locomoção.

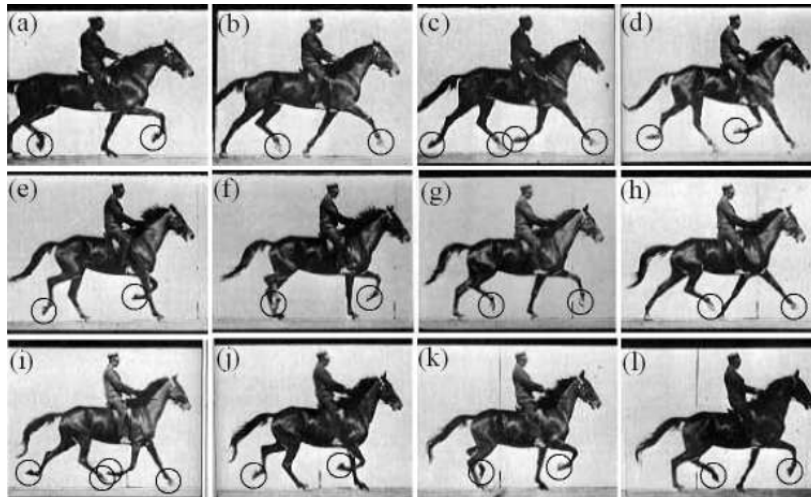


Figura 3.7: Sequência de movimentos do trote de um cavalo [33].

O diagrama de marcha do trote sem a fase de suspensão é mostrado na Figura 3.8. Para um período normalizado o movimento conta com duas etapas: na primeira o animal levanta as patas 1 e 4, e na etapa complementar as patas levantadas são as 2 e 3. Embora o levantar das pernas deve respeitar necessariamente o agrupamento descrito das patas levantadas, as etapas podem variar de ordem sem causar problemas na locomoção.

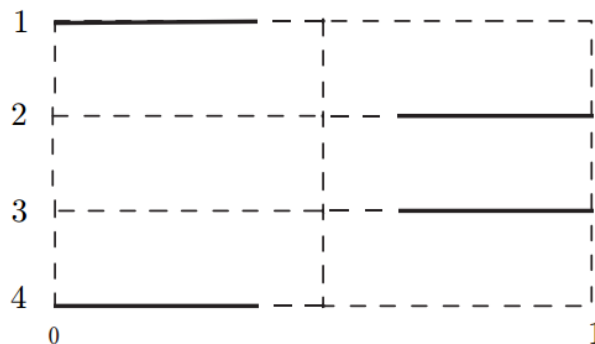


Figura 3.8: Gráfico de movimentos do padrão trote [27].

### 3.2.3 Galope

O galope por sua vez é uma marcha utilizada em altas velocidades. O corpo do quadrúpede fica grandes períodos de tempo suspenso e o contacto das patas com o solo é apenas para efetuar o impulso do movimento, como demonstra a Figura 3.9. A transição entre as pernas dianteiras e traseiras nesse tipo de marcha é feita de maneira suave.

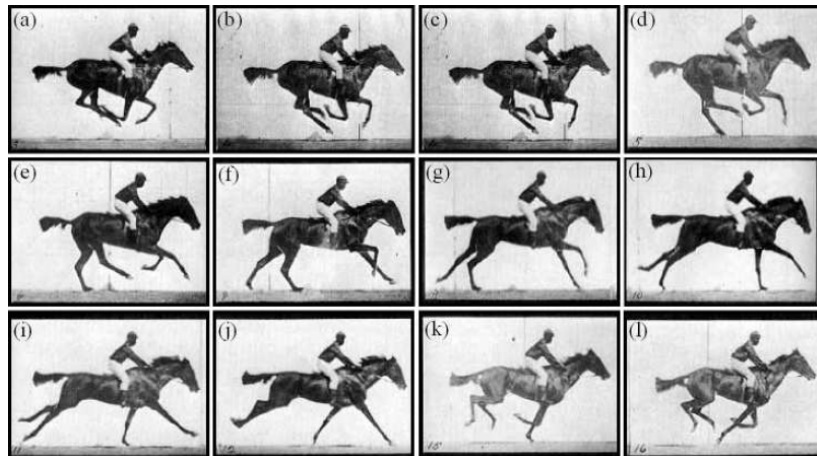


Figura 3.9: Sequência de movimentos do galope de um cavalo [33].

O diagrama de marcha do galope é ilustrado na Figura 3.10. Um período normalizado do galope mostra que as patas tocam o solo uma a uma, porém sem intervalos entre si. O equilíbrio do animal neste caso é dinâmico, o que garante uma elevada velocidade devido ao seu corpo estar sempre em movimento. Neste tipo de marcha as patas traseiras são apoiadas ao solo sempre antes das patas dianteiras, sendo que pouco antes da pata que está a apoiar o animal se levantar a próxima pata de apoio já toca o solo, causando assim o movimento ininterrupto do corpo.

## 3.3 Padrões de locomoção não-periódicos

A principal motivação para a implementação de um algoritmo de locomoção não periódico está no facto de que os algoritmos periódicos previamente mencionados têm pouca adaptabilidade frente a terrenos desconhecidos e bastante irregulares. A ideia para implementar um algoritmo de locomoção livre ou quase-periódico veio da complexidade ao qual a locomoção periódica exigia para executar trajetórias retilíneas e curvas simultaneamente.

A locomoção não periódica de um robô deve ser capaz de decidir sobre as ações frente aos obstáculos e mudar constantemente de padrão de locomoção.

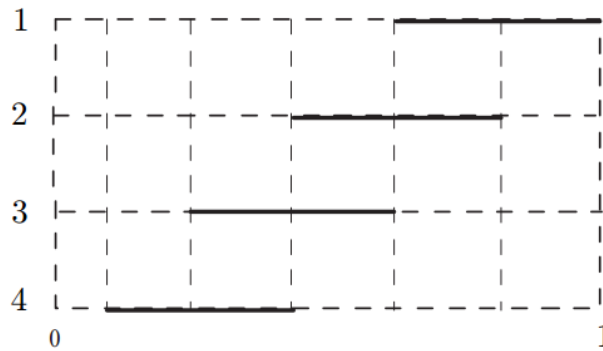


Figura 3.10: Gráfico de movimentos do padrão galope [27].

Isso exige uma grande autonomia e a implementação do algoritmo no robô torna-se muito mais complexa do que no caso em que existe periodicidade. Além disso, é interessante que o robô conte com sensores capazes de o auxiliar na tomada de decisões no planejamento de trajetória.

Ao passo que os padrões de locomoção periódicos ganham em velocidade, os não periódicos apesar de mais lentos são mais robustos e têm maior estabilidade.

Igarashi et. al. [34] propõem um modelo com a capacidade de alterar a postura do robô durante sua locomoção, cujo objetivo é o funcionamento em ambientes onde a adaptação rápida do robô é necessária. Adicionalmente, os autores desenvolvem uma marcha não-periódica utilizando processamento paralelo da postura do corpo e do posicionamento das pernas. A grande vantagem deste tipo de configuração é a troca de direção com a compensação do corpo do robô para garantir a estabilidade mais rapidamente, independente do movimento das pernas. E por sua vez, o robô decide qual perna fará o movimento baseado na sua própria condição ou condição do ambiente.

Huber et. al. [35] apresentam um modelo de marcha não periódica baseada em controle reativo e leituras de sensores do ambiente. O robô conta com uma arquitetura baseada em comportamento que garante maior flexibilidade e robustez ao modelo. A marcha é decidida com base nos dados lidos do ambiente no qual o robô está inserido e seu comportamento é construído através de um conjunto de sub-controladores base que representam os objetivos genéricos de controle. O modelo conta com dois controladores principais, um responsável pela rotação em seu próprio eixo e o outro para o caminhar. Ambos garantem a locomoção em terrenos desconhecidos ao mesmo tempo em que o modelo percebe e reage a presença de obstáculos.

Jeong et. al. [36] estudaram um método de locomoção quase-periódico para robôs quadrúpedes utilizando marchas não contínuas. O trabalho foca na adap-

tação dos padrões de locomoção de acordo com a direção no qual o robô pretende seguir. Tal modelo de locomoção une a eficiência dos padrões periódicos com a adaptabilidade exigida para o caso de mudança de trajetória do robô. O estudo sugere quatro tipos de movimentação para o caminhar retilíneo e dois tipos de movimentação em torno do próprio centro de massa. Além disso, é sugerido a implementação de um atuador linear vertical junto a um pantógrafo colocado no corpo do robô que permite um movimento da perna dissociado espacialmente, adequado para andar em terreno irregular. A Figura 3.11 ilustra o esquema proposto.

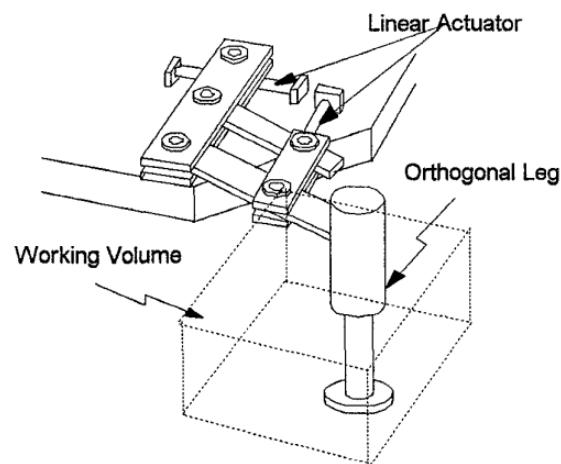


Figura 3.11: Esquema de uma perna espacialmente desacoplada [36].

## Capítulo 4

---

# Modelo do Robô Quadrúpede

---

*O objetivo do capítulo é mostrar o modelo de simulação do robô em MATLAB<sup>®</sup>/SimMechanics<sup>™</sup> que será utilizado para a implementação do algoritmo de controle desenvolvido nesta obra.*

### 4.1 Introdução

O modelo apresentado nesta seção, e que servirá de base para a implementação do algoritmo de controle, é o modelo desenvolvido em MATLAB<sup>®</sup>/SimMechanics<sup>™</sup> pelo autor Castro [37], cuja tese de mestrado deu oportunidade a este trabalho.

Basicamente, o modelo proposto é um quadrúpede com duas juntas rotativas por pata, sendo uma nas ancas para conectar o corpo do robô à parte superior da perna, e a outra nos joelhos, para conectar a parte superior à parte inferior da perna. Devido à escolha das pernas com juntas à frente, o modelo nunca apresenta um ângulo entre o eixo horizontal do robô e a parte superior da perna maior do que 90°. O modelo ainda apresenta um sistema de amortecimento que conecta seu corpo em 4 partes (cada parte conectada a uma perna) que proporciona melhor adaptabilidade e flexibilidade ao corpo. A Figura 4.1 mostra o aspeto do robô.

### 4.2 Modelo matemático do robô

Para a modelação de um robô são necessários definir parâmetros e equações referentes à sua cinemática e dinâmica. A cinemática é o estudo do movimento do robô sem se atentar às causas do mesmo, enquanto a dinâmica é a parte da modelação que estuda as forças e binários necessários para produzir tal movimento. Ambas as formulações têm a sua forma direta e inversa. Na cinemática direta a posição do atuador é obtida a partir do ângulo aplicado às juntas, enquanto na



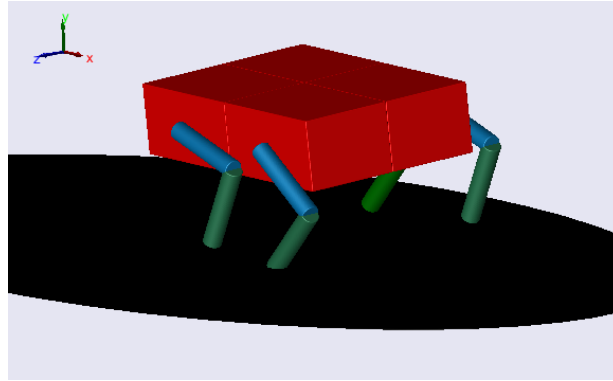


Figura 4.1: Modelo de robô quadrúpede desenvolvido por Castro [37].

cinemática inversa a posição desejada dá origem aos ângulos. Já no caso da dinâmica direta, as forças e binários aplicados dão origem ao movimento, enquanto na dinâmica inversa o binário e as forças são obtidos a partir do movimento desejado. O diagrama de fluxo da Figura 4.2 ilustra tais afirmações.

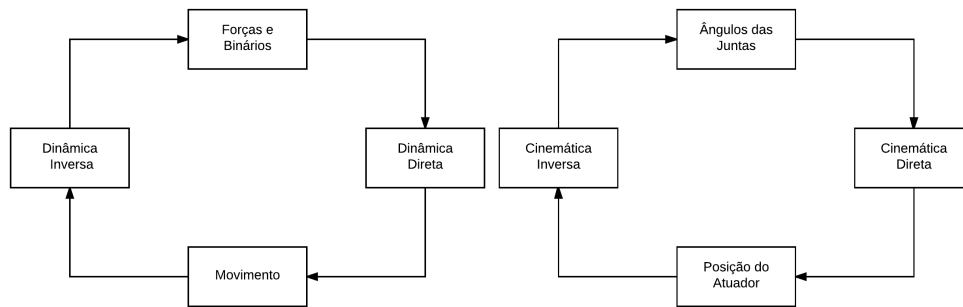


Figura 4.2: Diagramas de fluxo da cinemática e dinâmica de um robô.

Como pode se verificar, a forma direta de se calcular a cinemática e dinâmica exige cálculos muito mais simples do que a inversa, embora a inversa por vezes seja mais útil para o planejamento das ações dos robôs. Com relação à cinemática, para cada grau de liberdade de um robô aumenta-se muito as diferentes maneiras do mesmo alcançar certo ponto no espaço, e é aí que reside o principal problema dos cálculos inversos. Por exemplo, um robô com dois graus de liberdade tem duas maneiras de alcançar o mesmo ponto, como mostrado na Figura 4.3.

Para este trabalho foi necessário definir a matriz de inércia do modelo (Equação (4.1)), ou seja, a matriz que define como o modelo reage a partir do binário aplicado nos atuadores. Para facilitar os cálculos por parte do simulador, foi considerada a matriz de inércia simplificada paralela à origem dos eixos ortogonais

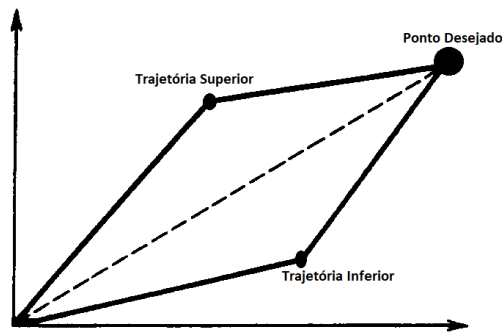


Figura 4.3: Problema de múltiplas soluções da cinemática inversa.

nas 3 dimensões, ou seja, a matriz de inércia não rotacionada, como sugere a Equação (4.2). Informações adicionais sobre matrizes de inércia e dinâmica não fazem parte do escopo deste trabalho, porém o estudo do autor Craig [38] é uma ótima recomendação para o aprofundamento no assunto.

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (4.1)$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (4.2)$$

Para garantir uma proximidade maior do modelo com o mundo real, o projeto levou em conta ainda a interação pé-solo para simular a aplicação de forças de ação e reação quando o robô se move. Porém, para efeito de simplificação, considerou-se que há apenas um ponto de contacto com o solo por pata do robô. A configuração de um contacto que simula um sistema mola-amortecedor foi usado na simulação. Como o ponto de contacto exerce uma força que não é paralela ao eixo vertical, é necessário dividir sua força em componente normal (eixo  $Y$ , força que garante o impulso do robô) e componente tangencial (eixo  $X$ , força que garante o movimento do robô). A Figura 4.4 mostra a decomposição das forças pé-solo no modelo.

Segundo Castro [37], o modelo apresenta melhores resultados quando a força normal segue o padrão não-linear do sistema massa-mola, enquanto a força tangencial não requer tal cuidado. Ambas as fórmulas utilizadas no modelo são apresentadas nas Equações (4.3) e (4.4).

$$F_x = -K_x(x - x_0) - B_x \dot{x} \quad (4.3)$$

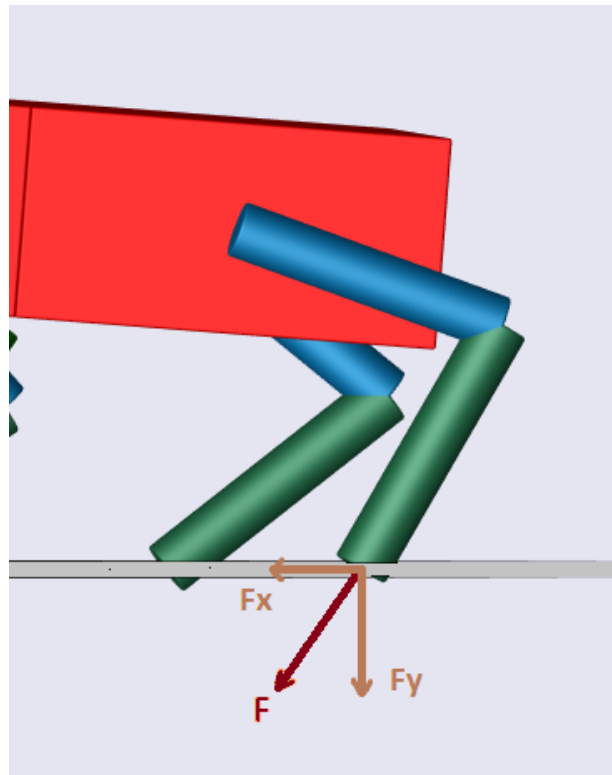


Figura 4.4: Decomposição de forças pé-solo no modelo.

$$F_y = -K_y y - B_y (-y)^m \dot{y} \quad (4.4)$$

Onde:

- $K_x$  e  $K_y$  são as constantes elásticas da mola para as forças decompostas;
- $x$  e  $y$  são as posições atuais da pata do robô no eixos correspondentes;
- $x_0$  é a última posição da pata do robô no eixo  $X$ ;
- $B_x$  e  $B_y$  são as constantes do amortecedor para as forças em  $X$  e  $Y$ , respectivamente;
- $\dot{x}$  e  $\dot{y}$  são as velocidades em  $X$  e  $Y$  da pata do robô, respectivamente;
- $m$  é a constante do amortecedor do robô, determinada experimentalmente.

### 4.3 Modelação com SimMechanics™

O modelo do robô foi feito com base na biblioteca SimMechanics™ do MATLAB®. Inicialmente, foi implementado para além do corpo, atuadores das juntas e sensores, um sistema de controlo PD. Porém, como a ênfase desse trabalho é justamente otimizar o controlo, esta secção abordará apenas os aspetos físicos do robô, sem se atentar ao sistema de controlo desenvolvido inicialmente. É importante salientar que as características dos blocos não serão abordadas a fundo, sendo que para uma melhor compreensão dos blocos é recomendado a leitura da obra original do modelo [37] e dos manuais de utilização da biblioteca SimMechanics™ do MATLAB® [39]. Porém, para um melhor entendimento das funções básicas que cada bloco exerce, é mostrado na Figura 4.5 a aparência dos blocos, e em seguida na Tabela 4.1, seus respectivos grupos, nomes e uma breve descrição sobre cada um deles.

O diagrama do modelo simplificado é mostrado na Figura 4.6. Como já dito anteriormente, o robô tem o corpo dividido em quatro partes conectadas com as respectivas pernas. Cada perna por sua vez conta com seus atuadores e sensores e ainda com um subsistema para calcular as forças resultantes do contacto com o solo. O sistema ainda apresenta um bloco GND que é indispensável para o robô saber sua posição em relação ao solo durante a simulação.

Para a melhor organização do trabalho foram criadas subsecções explicando detalhadamente o funcionamento de cada bloco do diagrama apresentado.

#### 4.3.1 Corpo do robô

O corpo do robô é interligado através de um sistema mola-amortecedor simples. Entre cada dois blocos existem duas juntas rotativas que garantem uma maior liberdade para esforços de torção quando o modelo se move em terrenos com obstáculos. Para cada bloco, as matrizes de inércia são as apresentadas no conjunto de Equações (4.5) e (4.6) e seu peso total é de aproximadamente 0,9 kg. As dimensões de largura ( $w$ ), profundidade ( $d$ ) e altura ( $h$ ) de cada bloco são 0,3 m x 0,2 m x 0,15 m, respetivamente. O diagrama de blocos implementado no SimMechanics™ é mostrado na Figura (4.7).

$$I_{corpo} = \frac{1}{12} \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \quad (4.5)$$

$$\begin{aligned} a_{11} &= m_{body}(h_{body}^2 + d_{body}^2) \\ a_{22} &= m_{body}(w_{body}^2 + d_{body}^2) \\ a_{33} &= m_{body}(w_{body}^2 + h_{body}^2) \end{aligned} \quad (4.6)$$

Tabela 4.1: Descrição dos blocos do modelo.

Grupo	Sub-grupo	Bloco	Nome	Descrição
M U L T I B O D Y  (1G)	Bodies	(1)	Body	Representação de um corpo rígido.
	Force Elements	(2)	Joint Spring & Damper	Representação de um oscilador linear amortecido.
	Joints	(3)	Revolute	Representação de um grau de liberdade rotacional.
	Joints	(4)	Universal	Representação de dois graus de liberdade rotacionais.
	Sensors & Actuators	(5)	Body Actuator	Aplicação de um binário ou força em um corpo rígido.
	Sensors & Actuators	(6)	Body Sensor	Medição do movimento em um corpo rígido.
	Sensors & Actuators	(7)	Joint Actuator	Aplicação de um binário ou força em uma junta qualquer.
	Sensors & Actuators	(8)	Joint Initial Condition	Definição da C.I. de binário, força ou velocidade em uma junta qualquer.
	Sensors & Actuators	(9)	Joint Sensor	Medição de posição, velocidade, aceleração, binário e binário reativo em uma junta qualquer.
S I M U L I N K	Discontinuities	(10)	Saturation	Saturação do resultado de uma função de entrada.
	Discrete	(11)	Memory	Memorização de um valor de entrada até que ocorra a troca da entrada.
	Math Operations	(12)	Sum	Soma de dois valores de entrada.
	Ports & Subsystems	(13)	If Block	Bloco condicional "sesenão".
	Ports & Subsystems	(14)	Inport	Conexão para entrada de valores de outros blocos.
	Ports & Subsystems	(15)	Outport	Conexão para saída de valores para outros blocos.
	Signal Routing	(16)	Merge	Fusão de duas ou mais entradas do bloco.
	Signal Routing	(17)	Selector	Seleciona uma entrada que será aplicada na saída.
	Sources	(18)	Constant	Assume um valor constante pré-definido.
	User Defined Functions	(19)	Fcn	Permite introduzir uma função baseada nos parâmetros de entrada do bloco.

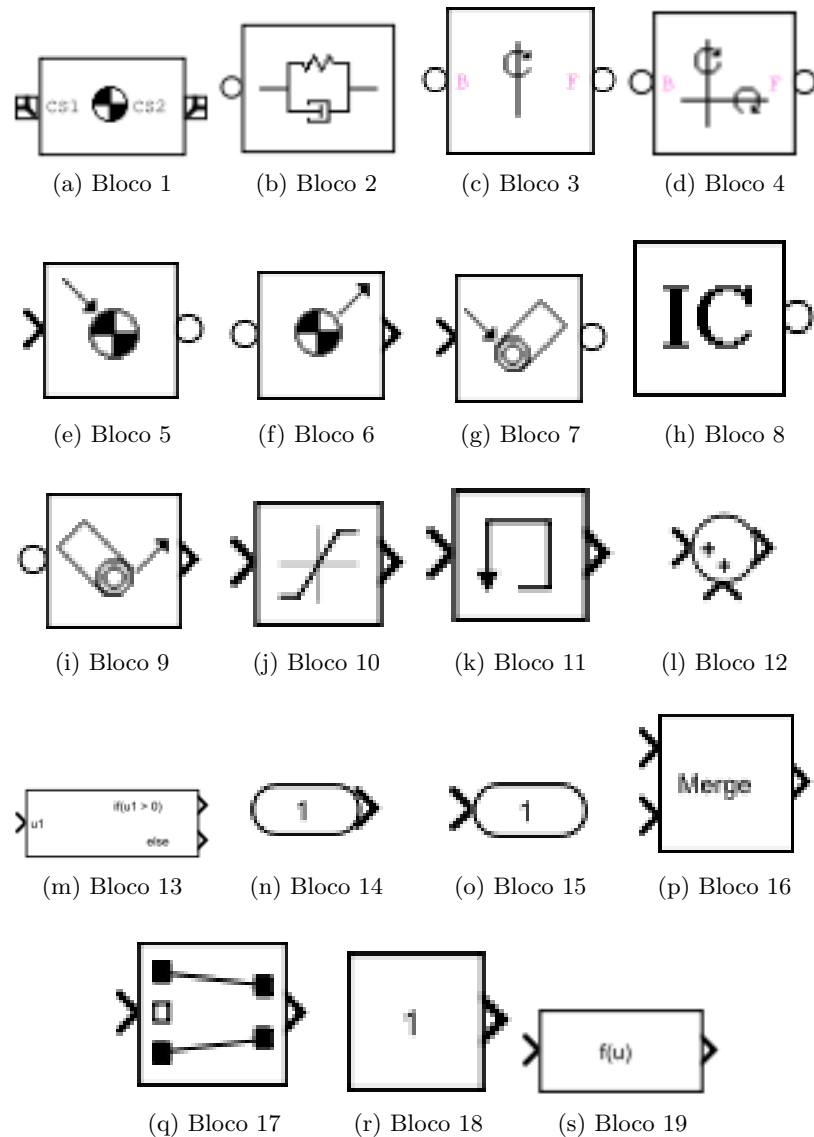


Figura 4.5: Aparência dos blocos utilizados no modelo.

### 4.3.2 Pernas do robô

As pernas do robô foram projetadas para serem cilíndricas, de raio  $r$  e comprimento  $L$ . Cada perna conta com duas juntas rotacionais com um grau de liberdade cada e são representadas pelos blocos *hip* e *knee* na Figura 4.8. Os corpos cilíndricos da perna são representados pelos blocos *leg\_sup* e *leg\_inf* que indicam a parte superior e a parte inferior, respectivamente. Os atuadores são representados pelos blocos *hip\_actuator* e *knee\_actuator* e os sensores por *hip\_joint\_sensor* e *knee\_joint\_sensor*. Para o sistema de controlo funcionar é atribuída a condi-

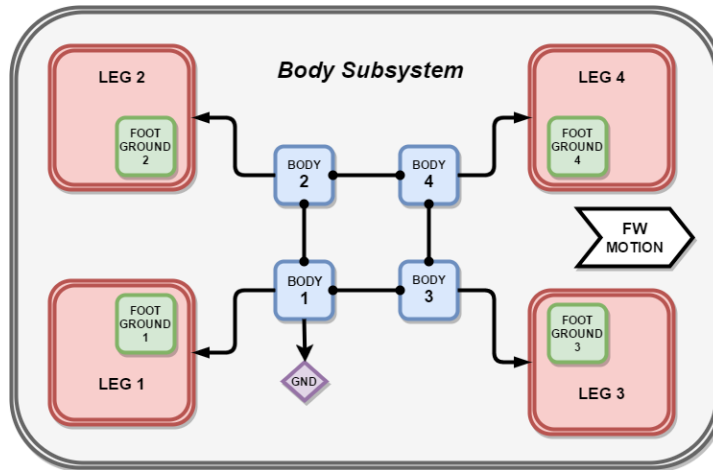


Figura 4.6: Diagrama simplificado do modelo do robô no SimMechanics™.

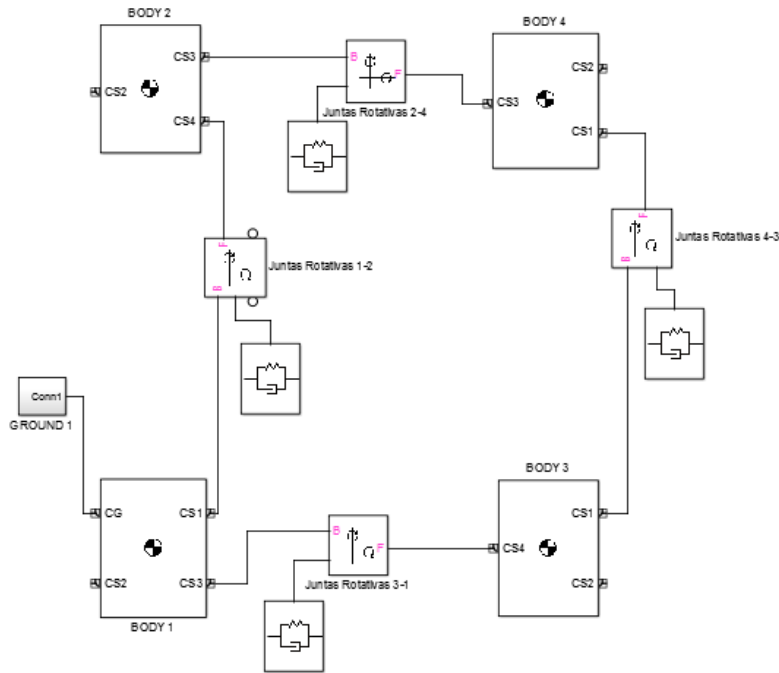


Figura 4.7: Diagrama do corpo do robô no SimMechanics™.

ção inicial das juntas através dos blocos *JIT\_hip* e *JIT\_knee*. O bloco da perna conecta-se ao corpo do robô pelas ancas e ao sistema de contacto com o solo através do bloco *leg\_inf*.

As matrizes de inércia das pernas utilizadas no modelo são definidas pelo conjunto de Equações (4.7) e (4.8). A variável  $m_{leg}$  é referente à massa de cada

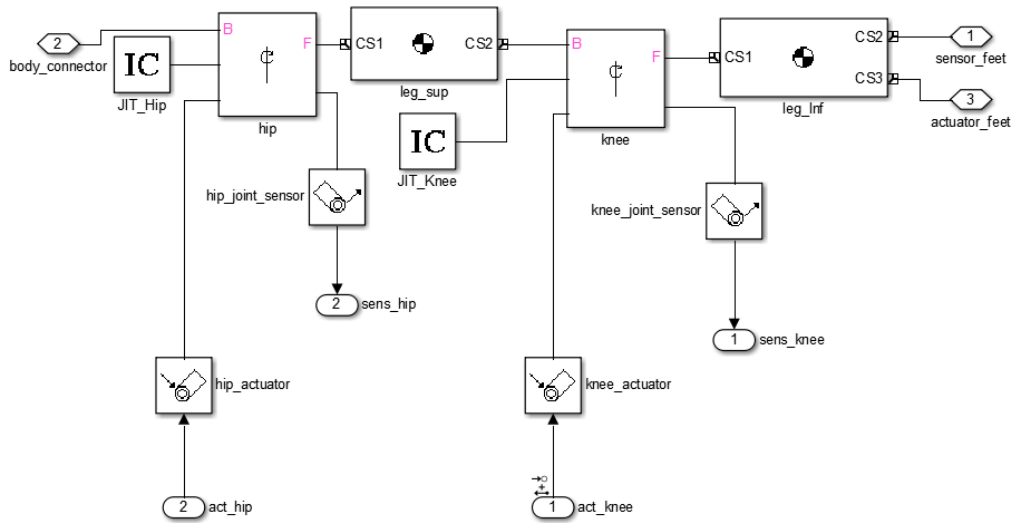


Figura 4.8: Diagrama da perna do robô no SimMechanics™.

uma das pernas, que para este modelo totaliza 0,1 kg.

$$I_{perna} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (4.7)$$

$$I_{xx} = \frac{1}{2}m_{leg}r^2$$

$$I_{yy} = \frac{1}{12}m_{leg}(3r^2 + L^2) \quad (4.8)$$

$$I_{zz} = \frac{1}{12}m_{leg}(3r^2 + L^2)$$

### 4.3.3 Contacto com o solo

O modelo de contacto com o solo foi feito com base em um único ponto de contacto por perna do robô e seu esquema geral é mostrado na Figura 4.9. Assume-se para a simulação o chão como um plano  $XZ$  na origem do eixo  $Y$ . Para que o bloco das forças normais e tangenciais seja acionado na perna em questão, é necessário que a extremidade da perna esteja em contacto com o solo. Essa condição é verificada dentro dos blocos *friction\_force* e *normal\_force* pelo bloco *Contact Test* (Figuras 4.10 e 4.12). Os blocos de contacto com o solo se tornam ativos quando o valor no eixo  $Y$  chega a zero. Como o robô se desloca somente em duas dimensões, a componente da força aplicada no eixo  $Z$  é idealmente nula. Antes de atingir o atuador (bloco *Body Actuator*) as componentes normal e tangencial da força são multiplexadas em um único sinal.



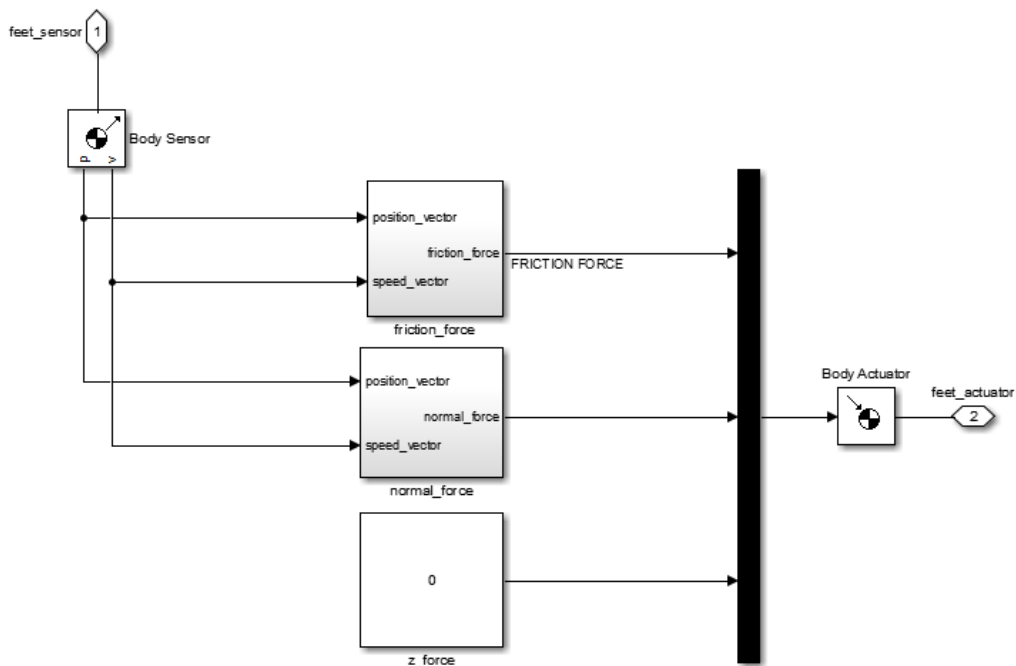


Figura 4.9: Diagrama de contacto com o solo do robô no SimMechanics™.

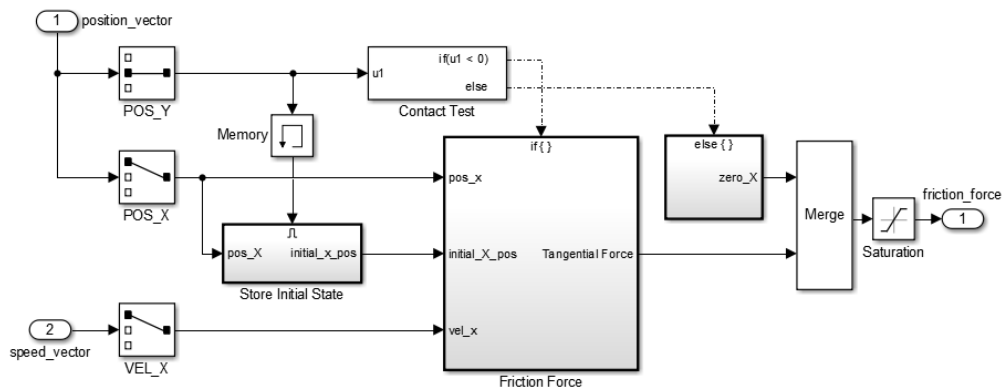
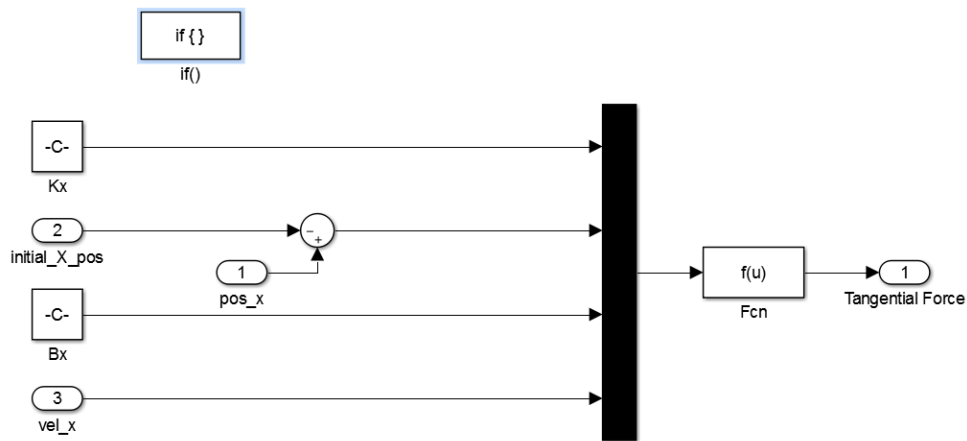
#### 4.3.3.1 Força tangencial

A componente tangencial da força que o robô aplica no solo é responsável pelo seu avanço no terreno, e seu modelo é baseado em um sistema mola-amortecedor simples já apresentado na Equação (4.3). A Figura 4.10 mostra os subsistemas do bloco *friction\_force*. Os dados de velocidade e posição tangencial são lidos pelo sensor e dão entrada a um segundo subsistema denominado *Friction Force* que faz o cálculo da força necessária para o deslocamento do robô. Além disso, é necessário saber a posição inicial relativa ao último movimento do robô (variável  $x_0$  na equação (4.3)), e para isso é inserido um elemento de memória através do bloco *Memory* e o referido valor é armazenado no bloco *Store Initial State*.

O subsistema *Friction Force* é apresentado na Figura 4.11. Esse subsistema é o responsável por multiplexar os dados necessários para o cálculo da força e aplicar a fórmula (4.3) através do bloco *Fcn*.

#### 4.3.3.2 Força normal

O bloco de força normal do sistema (apresentado na Figura 4.12) é similar ao bloco de força tangencial, porém o objetivo desse bloco é gerar o impulso e garantir que o robô esteja sempre acima do solo. A Equação (4.4) apresenta o comportamento de um sistema mola-amortecedor não linear que será usado para os cálculos da

Figura 4.10: Subsistema do bloco *friction\_force*.Figura 4.11: Bloco *Friction Force* da força tangencial.

força normal. Os dados lidos pelos sensores para este bloco são a velocidade e posição em  $Y$ , que dão entrada no bloco *Normal Force*. Um bloco de saturação ao final do sistema dá a liberdade de limitação aos valores do solo.

O subsistema *Normal Force* da Figura 4.13 tem o mesmo princípio que o subsistema *Friction Force* da componente tangencial, ou seja, multiplexar as constantes com os valores recebidos pelos sensores e aplicar, através do bloco *Fcn*, a equação apresentada em (4.4).

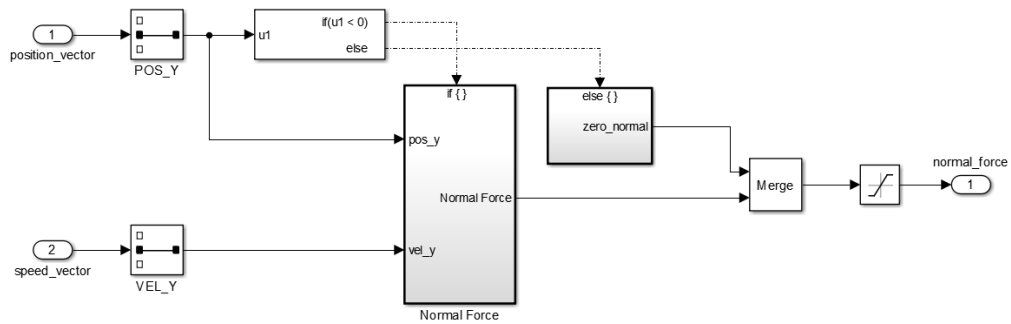


Figura 4.12: Subsistema do bloco *normal\_force*.

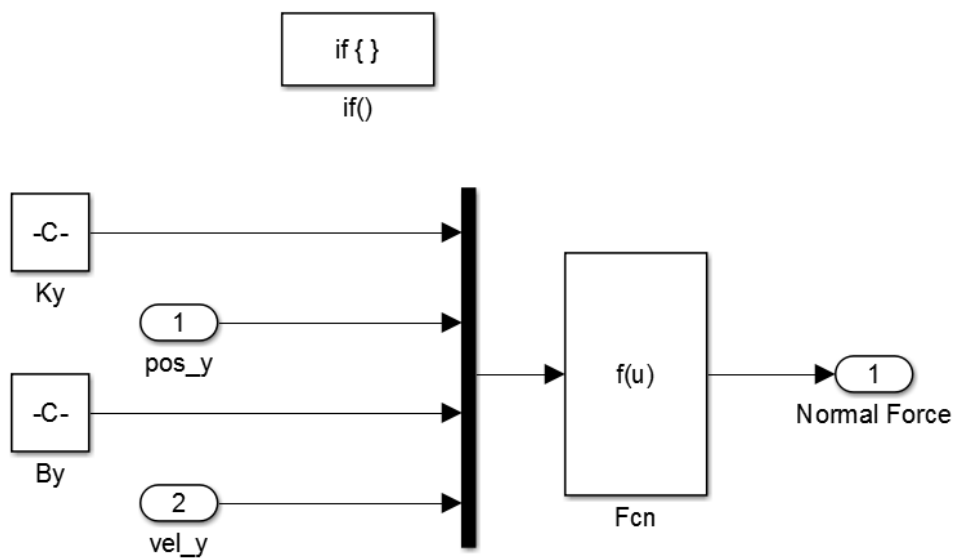


Figura 4.13: Bloco *Normal Force* da força normal.

## Capítulo 5

---

# Desenvolvimento do Trabalho

---

*Neste capítulo será apresentado o desenvolvimento do trabalho no software MATLAB<sup>®</sup> e a fundamentação matemática por detrás do planeamento de trajetórias, cinemática e implementação do controlo.*

### 5.1 Planeamento de trajetórias

O planeamento de trajetórias é o que define como será o caminhar do robô. Como já mencionado, este passa pela definição do padrão de locomoção adotado do robô, seja ele de estabilidade dinâmica ou estática. Porém, um fator que é comum a todos os tipos de padrões de locomoção periódicos é a trajetória da perna durante o movimento. Uma boa solução neste caso é o uso da cicloide durante a fase de transporte, sendo o tempo de transporte  $T_t = (1 - \beta)T_0$ , onde  $\beta$  é o fator de ocupação da perna. O corpo, por sua vez, precisa ter um deslocamento constante no tempo a uma velocidade frontal definida por  $V_F$ , e também ter seu centro de gravidade alinhado com a extremidade da perna quando a cicloide atinge seu valor máximo. Para um período  $T_0$ , as equações do movimento do corpo são descritas em (5.1) e as equações para o movimento das pernas são dadas em (5.2).

$$\begin{cases} X_{CORPO}(t) = V_F t + L_S \frac{\beta}{2} \\ Y_{CORPO}(t) = H_{RB} \end{cases} \quad (5.1)$$

$$\begin{cases} X_{T_{PERNA}}(t) = V_F \left[ t - \frac{T_0 \text{sen}(\frac{2\pi t}{T_0})}{2\pi} \right], & 0 < t \leq (1 - \beta)T_0 \\ Y_{T_{PERNA}}(t) = \frac{F_C}{2} \left[ 1 - \cos(\frac{2\pi t}{T_0}) \right], & 0 < t \leq (1 - \beta)T_0 \\ X_{S_{PERNA}}(t) = X_{T_{PERNA}}((1 - \beta)T_0), & (1 - \beta)T_0 < t < T_0 \\ Y_{S_{PERNA}}(t) = 0, & (1 - \beta)T_0 < t < T_0 \end{cases} \quad (5.2)$$

Onde:

- $X_{CORPO}(t)$  é a coordenada no eixo  $X$  para o movimento do corpo do robô;
- $Y_{CORPO}(t)$  é a coordenada no eixo  $Y$  para o movimento do corpo do robô;
- $L_S \frac{\beta}{2}$  é a relação que garante o alinhamento do corpo e da extremidade da perna quando  $Y_{T_{PERNA}}(t)$  atinge o valor máximo.
- $H_{RB}$  é a altura do centro de massa do corpo do robô em relação ao solo;
- $X_{T_{PERNA}}(t)$  é a coordenada no eixo  $X$  para a fase de transferência da perna;
- $Y_{T_{PERNA}}(t)$  é a coordenada no eixo  $Y$  para a fase de transferência da perna;
- $X_{S_{PERNA}}(t)$  é a coordenada no eixo  $X$  para a fase de suporte da perna;
- $Y_{S_{PERNA}}(t)$  é a coordenada no eixo  $Y$  para a fase de suporte da perna;
- $T$  é o período do ciclo completo da perna;
- $F_C$  é a altura da cicloide implementada;
- $t$  é a variável de tempo.

No caso da trajetória completa da perna  $[X_{PERNA}, Y_{PERNA}]^T$ , é necessário concatenar os resultados obtidos para os dois casos existentes durante o período  $T_0$ , ou seja:

$$\begin{cases} X_{PERNA} = [X_{T_{PERNA}}(t) & X_{S_{PERNA}}(t)] \\ Y_{PERNA} = [Y_{T_{PERNA}}(t) & Y_{S_{PERNA}}(t)] \end{cases} \quad (5.3)$$

O referencial da perna é sempre a junta da anca. Com isso em mente, é necessário trocar as coordenadas do referencial absoluto (origem do sistema) para o referencial relativo (junta da anca), como demonstra a Equação (5.4).

$$\begin{cases} X_{cin} = X_{PERNA}(t) - X_{CORPO}(t) \\ Y_{cin} = Y_{PERNA}(t) - Y_{CORPO}(t) \end{cases} \quad (5.4)$$

O planeamento implementado no robô segue as equações previamente mencionadas, e é demonstrado para um único ciclo através da Figura 5.1.

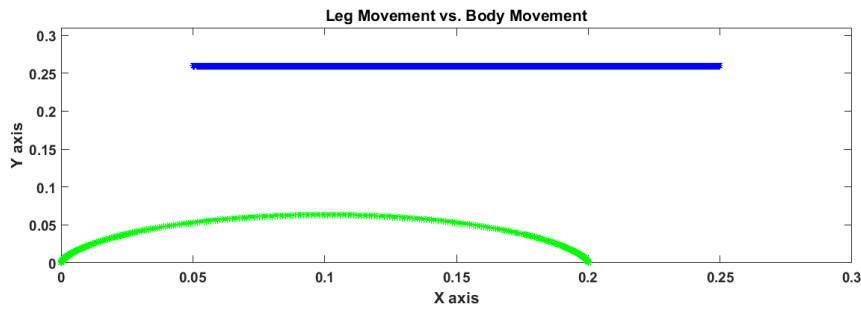


Figura 5.1: Trajetória planejada aplicada ao modelo do robô.

## 5.2 Estudo da cinemática

De acordo com Craig [38] a cinemática é o campo da mecânica que estuda o movimento de um corpo sem se atentar ao que o causa. Tendo em vista tal consideração, a cinemática é um passo importante para o desenvolvimento correto da trajetória do robô, pois é devido à relação entre o espaço cartesiano e o espaço de juntas que é possível transmitir o comportamento desejado ao atuador. Assim sendo, propôs-se a cinemática do robô, utilizando para isso apenas uma das patas do robô devido à redundância entre as quatro presentes no modelo. A análise dos ângulos e comprimentos é ilustrada na Figura 5.2.

### 5.2.1 Cinemática direta

A cinemática direta é o estudo responsável por definir no plano cartesiano a posição da extremidade de um robô de acordo com o ângulo que é aplicado em suas juntas. Através de relações trigonométricas e utilizando o teorema de Pitágoras aplicado na Figura 5.2, é extraída a posição correspondente do pé do robô nos eixos  $X$  e  $Y$  através das Equações (5.5) e (5.6), respetivamente.

$$X_{cin} = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (5.5)$$

$$Y_{cin} = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \quad (5.6)$$

Onde:

- $X_{cin}$  é a posição em  $X$  onde se encontra a extremidade inferior da pata do robô em determinado instante em relação ao referencial da anca;

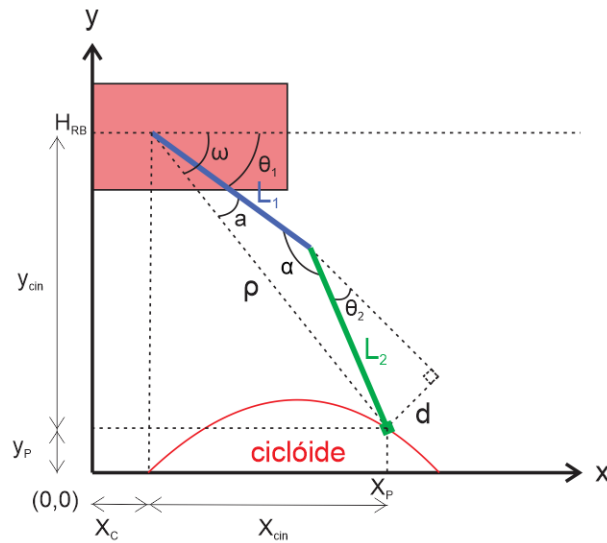


Figura 5.2: Disposição dos ângulos para o cálculo da cinemática do modelo.

- $Y_{cin}$  é a distância, medida no eixo  $Y$ , entre a altura  $H_{RB}$  da junta da anca à extremidade inferior da pata do robô;
- $L_1$  é o comprimento da parte superior da perna do robô;
- $L_2$  é o comprimento da parte inferior da perna do robô;
- $\theta_1$  é o valor angular da junta das ancas;
- $\theta_2$  corresponde ao valor angular da junta dos joelhos.

### 5.2.2 Cinemática inversa

A cinemática inversa é o estudo que faz o caminho inverso da cinemática direta, ou seja, a partir da posição desejada para o robô obtêm-se os ângulos necessários para o robô alcançá-lo. Para o caso da Figura 5.2, as relações iniciais necessárias para a obtenção do modelo são descritas pelas expressões em (5.7).

$$\begin{cases} Y_{cin} = Y_P - H_{RB} \\ X_{cin} = X_P - X_C \\ \alpha = 180^\circ - \theta_2 \\ d = L_2 \text{sen}(\theta_2) \end{cases} \quad (5.7)$$

Onde:

- $H_{RB}$  é a altura do centro de massa do corpo do robô;

- $Y_c$  é a altura no eixo  $Y$  da trajetória planeada em um determinado instante de tempo.

Utilizando a relação de Pitágoras, o comprimento da hipotenusa  $\rho$  pode ser representado por:

$$\rho = \sqrt{X_{cin}^2 + Y_{cin}^2} \quad (5.8)$$

Uma outra relação para  $\rho$  é encontrada ao aplicar a lei dos cossenos, resultando na Equação (5.9) que relaciona também os comprimentos  $L_1$  e  $L_2$ .

$$\rho^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(\alpha) \quad (5.9)$$

Logo, ao agrupar as Equações (5.8) e (5.9), uma relação entre a posição da trajetória  $[X_{cin}, Y_{cin}]^T$ , o comprimento das pernas  $L_1$  e  $L_2$  e o ângulo dos joelhos  $\theta_2$  é obtida por:

$$X_{cin}^2 + Y_{cin}^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(180^\circ - \theta_2) \quad (5.10)$$

Devido à característica de função par dos cossenos, a Equação (5.10) pode ser reduzida na forma:

$$X_{cin}^2 + Y_{cin}^2 = L_1^2 + L_2^2 + 2L_1L_2\cos(\theta_2) \quad (5.11)$$

Ao isolar  $\theta_2$  é obtida a Equação (5.12) que corresponde ao ângulo das juntas dos joelhos.

$$\theta_2 = \text{acos} \left( \frac{X_{cin}^2 + Y_{cin}^2 - L_1^2 - L_2^2}{2L_1L_2} \right) \quad (5.12)$$

Já para o ângulo  $\theta_1$ , a relação inicial pode ser fornecida por:

$$\theta_1 = \omega - a \quad (5.13)$$

Através da relação das tangentes, o conjunto de equações (5.14) descreve os valores de  $\omega$  e  $a$  em termos das dimensões do robô, coordenadas da trajetória e ângulo  $\theta_2$ .

$$\begin{cases} \omega = \text{atan}_2 \left( \frac{Y_{cin}}{X_{cin}} \right) \\ a = \text{atan}_2 \left( \frac{L_2 \text{sen}(\theta_2)}{L_1 + L_2 \text{cos}(\theta_2)} \right) \end{cases} \quad (5.14)$$



Substituindo  $\omega$  e  $a$  por suas respectivas relações, obtém-se o ângulo  $\theta_1$  das juntas das ancas, descrito por:

$$\theta_1 = \text{atan}_2\left(\frac{Y_{cin}}{X_{cin}}\right) - \text{atan}_2\left(\frac{L_2 \text{sen}(\theta_2)}{L_1 + L_2 \text{cos}(\theta_2)}\right) \quad (5.15)$$

A função  $\text{atan}_2$  é utilizada nos cálculos por retornar o valor correto do ângulo desejado de acordo com sinal do par ordenado inserido. Como exemplo, para o par ordenado genérico  $(arg_1, arg_2)$ , a função  $\text{atan}_2$  pode ser representada por:

$$\text{atan}_2(arg_2, arg_1) = \begin{cases} \text{atan}\left(\frac{arg_2}{arg_1}\right) & arg_1 > 0 \\ \text{atan}\left(\frac{arg_2}{arg_1}\right) + \pi & arg_2 \geq 0, arg_1 < 0 \\ \text{atan}\left(\frac{arg_2}{arg_1}\right) - \pi & arg_2 < 0, arg_1 < 0 \\ +\frac{\pi}{2} & arg_2 > 0, arg_1 = 0 \\ -\frac{\pi}{2} & arg_2 < 0, arg_1 = 0 \\ \text{indeterminado} & arg_2 = 0, arg_1 = 0 \end{cases} \quad (5.16)$$

As equações (5.12) e (5.15) referentes aos ângulos  $\theta_1$  e  $\theta_2$  respectivamente, são utilizadas tanto para a fase de transferência quanto para a fase de sustentação das pernas do robô.

### 5.3 Linearização do modelo

Devido à dificuldade de se obter um modelo matemático completo do robô quadrúpede implementado, buscou-se alternativas para obter um modelo simplificado do mesmo, o que facilitaria também a obtenção dos parâmetros do controlador. O modelo linear, no entanto, deve ser projetado no intuito de conter as principais características do modelo real, bem como se aproximar do seu comportamento.

A linearização de um modelo matemático é feita em torno de uma condição de operação no qual deseja-se que o sistema opere. É importante que o sistema seja linearizado de maneira a aproximar-se da condição de operação real do sistema para que o modelo dimensionado tenha validade. Uma boa opção para a linearização é a utilização da técnica de *trimming*, que atribui um estado estacionário às variáveis que têm pouca influência na dinâmica do modelo. Essa técnica elimina polos e zeros indesejáveis e torna a sintonia dos controladores mais simples.

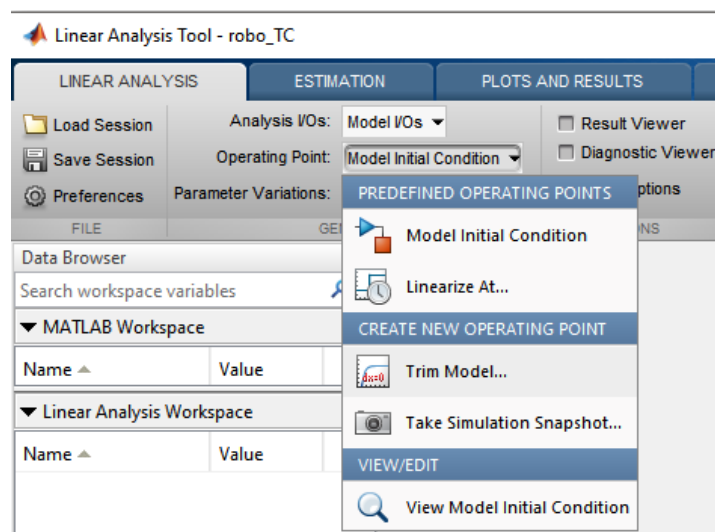
No caso do robô quadrúpede desenvolvido foi definido como estados estacionários todas as variáveis do modelo que não têm influência direta na locomoção do robô, tais como as variáveis do sistema massa-mola-amortecedor do corpo, dos sensores de binário e velocidade angular e de outros tipos de sensores utilizados para a medição de parâmetros. A lista completa com as variáveis de estado utilizadas é demonstrada na Tabela 5.1.

Tabela 5.1: Parâmetros de linearização por *trimming* do modelo.

Variável de Estado	Estado no Modelo Linear	Influência na Locomoção
Velocidade das Juntas do Corpo	Estado Estacionário	Baixa
Posição das Juntas do Corpo	Estado Estacionário	Baixa
Velocidade da Referência	Estado Estacionário	Baixa
Posição da Referência	Estado não Estacionário	Média
Quaternião da Referência	Estado Estacionário	Baixa
Velocidade Angular da Referência	Estado Estacionário	Baixa
Velocidade da Junta das Ancas	Estado não Estacionário	Alta
Posição da Junta das Ancas	Estado não Estacionário	Alta
Velocidade da Junta dos Joelhos	Estado não Estacionário	Alta
Posição da Junta dos Joelhos	Estado não Estacionário	Alta

### 5.3.1 Análise linear

Para obter o sistema linear do robô foi utilizada a ferramenta *Linear Analysis* do Simulink™. Primeiramente, dentro do ambiente desta ferramenta, obteve-se a condição de operação que será utilizada no modelo ao aceder à configuração de *Operating Point > Trim Model*, como mostra a Figura 5.3.

Figura 5.3: *Trim model* na ferramenta *Linear Analysis*.

Após selecionada a linearização por *Trim Model*, a janela de definição das variáveis de estados que permanecerão estacionárias é mostrada na Figura 5.4. A configuração da coluna *Steady State* ocorre conforme o especificado na Tabela 5.1, e todos os campos da coluna *Known* permanecem desmarcados.

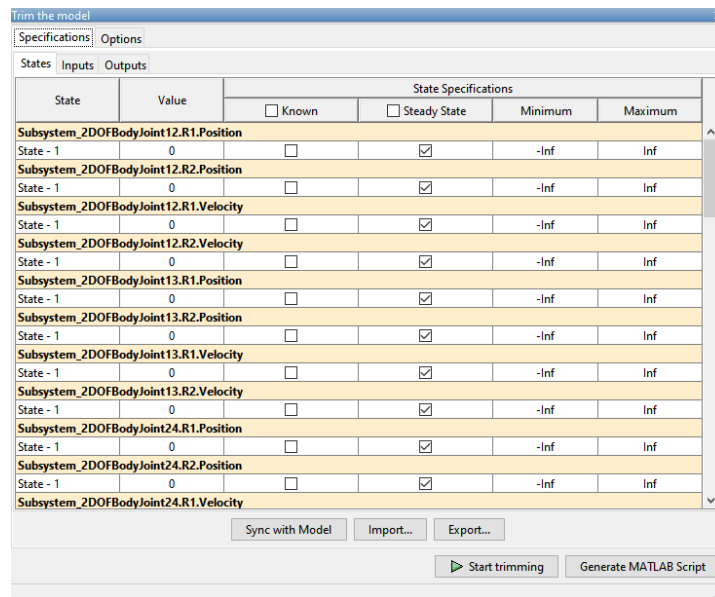


Figura 5.4: Seleção dos estados das variáveis em *trim model*.

Após a definição de estados e o *trimming* do modelo, a janela da Figura 5.5 é apresentada, indicando que a condição de operação foi obtida com sucesso.

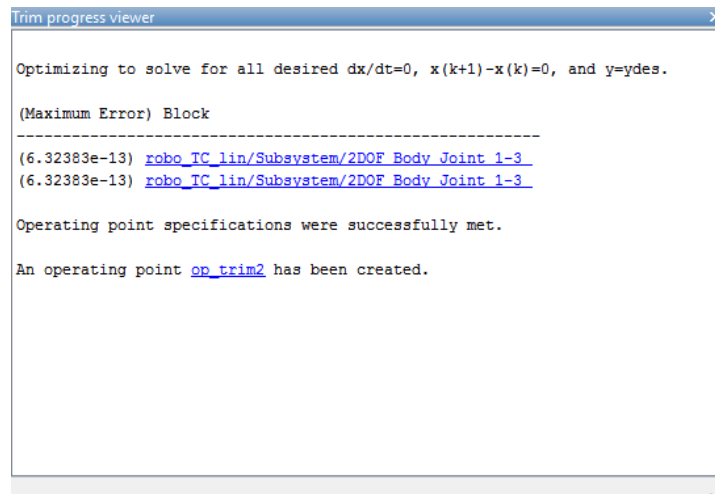


Figura 5.5: Especificação do *trim model* obtida com sucesso.

Uma vez definida a condição de operação, é necessário obter o modelo linearizado do sistema de acordo com a entrada e saída selecionada. Para tal deve-se inserir o ponto de perturbação *Input Perturbation* na entrada do atuador da junta selecionada, e um ponto de medida em malha aberta *Open-loop Output* na saída

do sensor desta mesma junta. Esse tipo de análise permite ver o sistema completo do robô a partir da junta selecionada. Para a configuração das entradas e saídas dentro do ambiente *Linear Analysis* deve-se aceder a *Analysis I/Os > Create New Linearization I/Os*. As entradas e saídas podem ser selecionadas através da janela *Analysis I/Os > Edit Model I/Os*, conforme exemplificado na Figura 5.6.

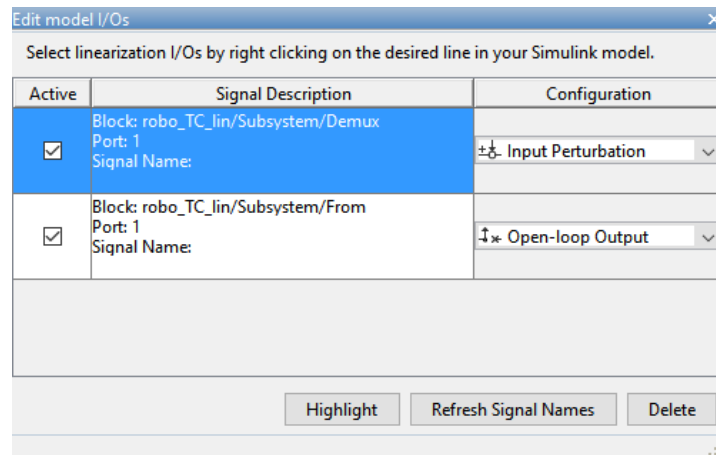


Figura 5.6: Seleção de *I/O Ports* em *Linear Analysis*.

No caso aplicado ao robô, obteve-se o modelo do sistema a partir das juntas das ancas e joelhos da perna 1, o que resultou em dois sistemas lineares distintos. A resposta ao degrau do sistema linearizado para as ancas e para os joelhos é demonstrado nas Figuras 5.7 e 5.8, respetivamente. Como se pode notar, os sistemas apresentados são instáveis, o que justifica a implementação do controlo para estabilizar a planta.

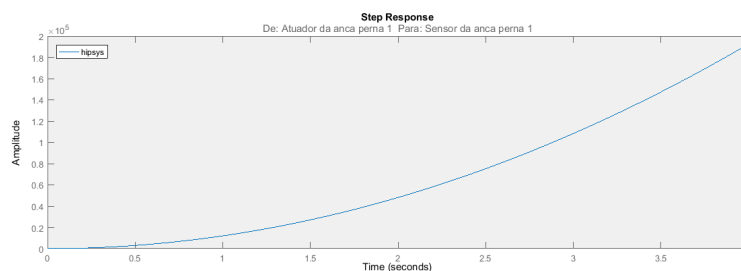


Figura 5.7: Resposta ao degrau do sistema linearizado a partir da anca.

### 5.3.2 Espaço de estados

A representação do modelo linear pela ferramenta *Linear Analysis* é dada por equação de espaço de estados, genericamente representada em sistemas *Single-*

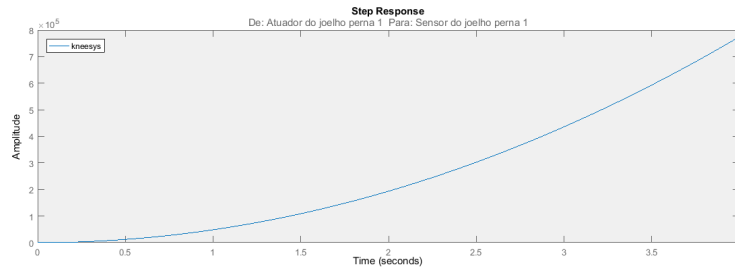


Figura 5.8: Resposta ao degrau do sistema linearizado a partir do joelho.

*Input-Single-Output* (SISO) no domínio do tempo por:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Br(t) \\ y(t) &= Cx(t) + Dr(t) \end{aligned} \quad (5.17)$$

Dado um número  $n$  de estados do sistema, as matrizes  $A, B, C$  e  $D$  são definidas por:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$C = [c_1 \cdots c_n] \quad D = d_0$$

Onde:

- $\dot{x}(t)$  é o vetor das derivadas dos estados do sistema;
- $x(t)$  é o vetor de estados do sistema;
- $y(t)$  é a saída do sistema;
- $r(t)$  é a entrada aplicada ao sistema;
- $A$  é a matriz constante que correlaciona os estados do sistema entre si;
- $B$  é a matriz constante que correlaciona os estados com a entrada do sistema;
- $C$  é a matriz constante que relaciona os estados com a saída do sistema;
- $D$  é a matriz constante que relaciona entradas e saídas do sistema.

Porém, para a aplicação do sistema na ferramenta de controlo do Simulink™, é necessário analisar a equação de espaço de estados no domínio da frequência  $s$ . Após a aplicação da Transformada de Laplace em (5.17), o espaço de estados no domínio da frequência é dado por:

$$\begin{aligned} sX(s) &= AX(s) + BR(s) \\ Y(s) &= CX(s) + DR(s) \end{aligned} \quad (5.18)$$

Usualmente, o sistema linearizado é descrito pela forma de função de transferência  $G(s)$  na arquitetura de controlo. Tal forma é obtida ao arranjar a equação (5.18) relacionando a saída  $Y(s)$  com a entrada  $R(s)$ , como mostra a equação (5.19).

$$G(s) = \frac{Y(s)}{R(s)} = C(sI - A)^{-1}B + D \quad (5.19)$$

Para o caso do robô implementado, as equações de estado das ancas e dos joelhos contêm inicialmente 32 estados cada. No propósito de cancelar polos e zeros desnecessários do espaço de estados, aplicou-se a minimização das equações através do comando *minreal* no *prompt* do MATLAB®. Após a minimização, o modelo visto a partir da anca obteve um total de 17 estados, enquanto que para os joelhos o total de estados foi 5. É importante salientar que o modelo linearizado levou em consideração um valor de saturação de  $\pm 20\text{Nm}^{-1}$  para o binário das juntas. Os valores das matrizes constantes de ambos os modelos estão apresentadas no Apêndice A.

## 5.4 Sistema de controlo

O sistema de controlo para a planta linearizada  $G(s)$  das ancas e dos joelhos do robô foi modelado através da ferramenta *Control System Designer* do Simulink™. Foram implementados dois tipos diferentes de controlador no sistema: um para as juntas das ancas e outro para as juntas dos joelhos. Devido à similaridade das pernas do robô, a análise foi feita utilizando apenas o modelo visto a partir das juntas da perna 1 e replicado para a junta das outras pernas.

Ao todo o sistema de controlo do robô apresenta oito controladores independentes, sendo quatro para as ancas e quatro para os joelhos. Apesar de o sistema utilizado para a aplicação das leis de controlo ser um sistema linearizado, a atuação do controlador será feita no modelo não-linear do robô devido a este apresentar uma melhor estimativa do seu comportamento real. O objetivo do controlador é minimizar a diferença entre o valor do ângulo que é lido no sensor da junta e o ângulo de referência gerado pela cinemática inversa. Tal correção é

feita aplicando o binário necessário no atuador das juntas. O diagrama de controlo implementado no modelo pode ser representado pelo diagrama de blocos da Figura 5.9.

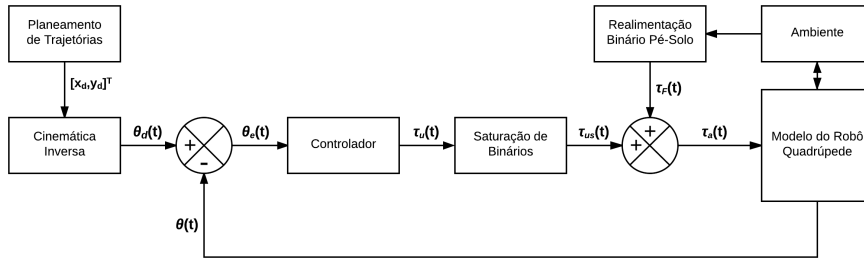


Figura 5.9: Diagrama completo do sistema de controlo.

Onde:

- $[x_d(t), y_d(t)]^T$  é a matriz das posições desejadas na perna do robô;
- $\theta_d(t)$  é vetor de ângulo desejado para as juntas do robô;
- $\theta(t)$  é o vetor de ângulos medidos nos sensores das juntas;
- $\theta_e(t)$  é o vetor da diferença entre os ângulos  $\theta_d(t)$  e  $\theta(t)$  ;
- $\tau_u(t)$  é o vetor de binário aplicado pelo controlador no robô;
- $\tau_{us}(t)$  é o vetor de binário saturado aplicado pelo controlador no robô;
- $\tau_F(t)$  é o vetor de binário de reação do sistema pé-solo do robô;
- $\tau_a(t)$  é o vetor de binário total que é introduzido no sistema.

O tipo de controlador é uma premissa importante a ser definida, e deve levar em consideração o tipo de resposta que se deseja. No caso de controlo de trajetórias, devido ao fato de a referência ser variável, é usual aplicar um controlador do tipo Proporcional-Derivativo com Filtro, também conhecido por PDF. Nestes casos, o fator integrativo do controlo tem pouca influência na resposta final.

#### 5.4.1 Implementação do sistema de controlo no Simulink™

O sistema de controlo implementado no modelo é demonstrado na Figura 5.10. A entrada do sistema de controlo recebe o vetor  $\theta(t)$  contendo a medida dos sensores de cada uma das juntas. Os ângulos  $\theta_d(t)$  são aplicados à entrada de referência do bloco. Na saída do controlador existe um bloco de saturação que limita os

valores de binário aplicados às juntas com o objetivo de aproximar o modelo de uma situação real.

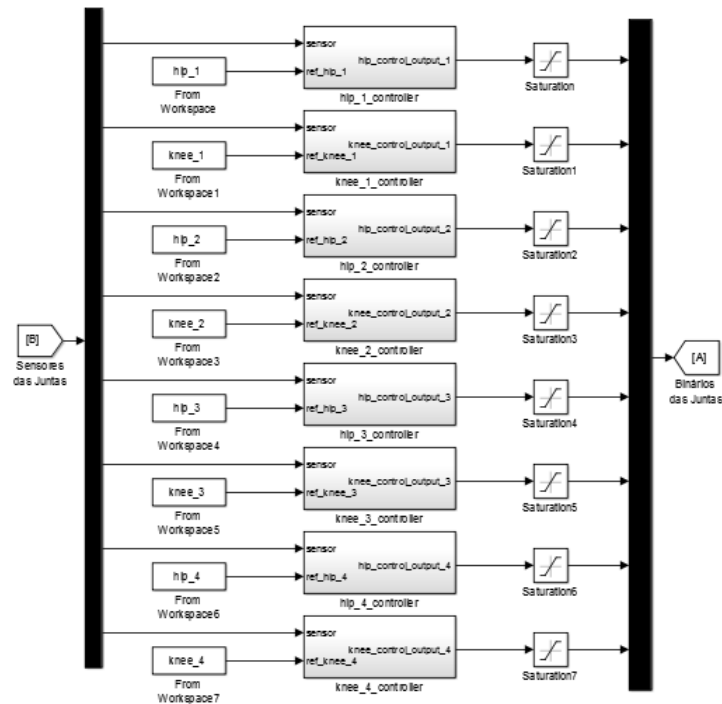


Figura 5.10: Sistema de controlo implementado em Simulink™.

Dentro do bloco principal de cada controlador é implementada a função de transferência do compensador, ou simplesmente  $C(s)$ , através do bloco *LTI System* da biblioteca do Simulink™. Basicamente, esse bloco importa a lei de controlo projetada pelo *Control System Designer* para dentro do ambiente de simulação, o que torna mais rápido a mudança de parâmetros durante os testes do controlador. A Figura 5.11 mostra o esquema de implementação do bloco na simulação.

#### 5.4.2 Estimação de parâmetros via Control System Designer

Para a sintonia dos controladores  $C_{hip}(s)$  e  $C_{knee}(s)$ , a ferramenta utilizada é o *Control System Designer* devido a esta permitir tanto o ajuste manual como o ajuste automático dos parâmetros, bem como a visualização em tempo real do comportamento da resposta do sistema através de gráficos de Bode, respostas ao degrau, respostas ao impulso, diagramas de Nyquist e mapas de zeros e polos.

Inicialmente, a arquitetura para o controlo foi selecionada em *Edit Architecture*. A configuração utilizada é a mostrada na Figura 5.12 devido à semelhança com a arquitetura do modelo de controlo proposto.



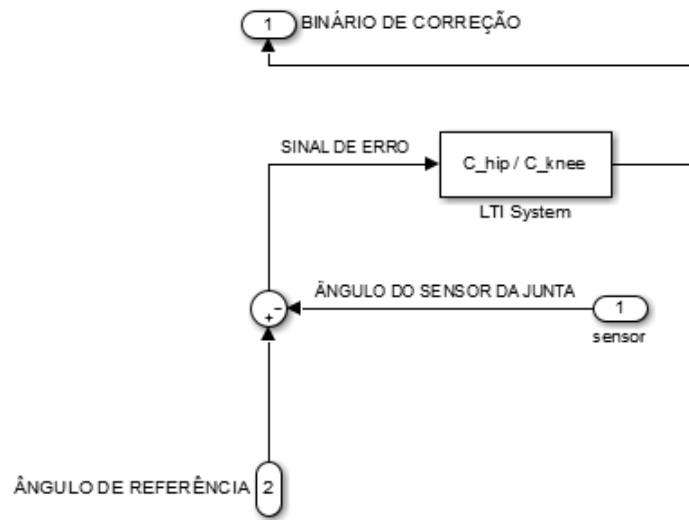


Figura 5.11: Implementação da arquitetura do controlador na simulação através do bloco *LTI System*.

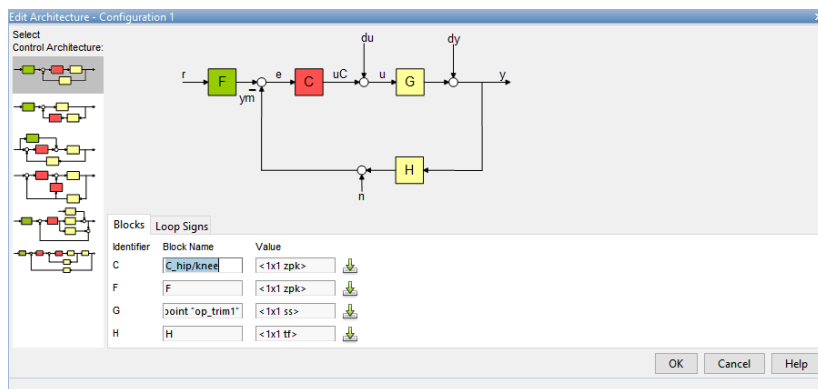


Figura 5.12: Arquitetura do controlador selecionada em *Control System Designer*.

Após definida a arquitetura, a sintonia foi feita no ambiente *PID Tuning*, que é acessado dentro da ferramenta *Control System Designer* através do caminho *Tuning Methods > PID Tuning*, conforme mostra a Figura 5.13.

Para o caso dos controladores propostos, a sintonia baseia-se na variação do comportamento transitório (*Transient Behaviour*) e no tempo de resposta (*Response Time*) desejado para o sistema. O método de sintonia é o *Robust Response Time* e conforme justificado anteriormente, o tipo de controlador selecionado é o Proporcional-Derivativo com Filtro. Para aplicar a função de transferência fornecida pela ferramenta nos gráficos das respostas é necessário atualizar o com-

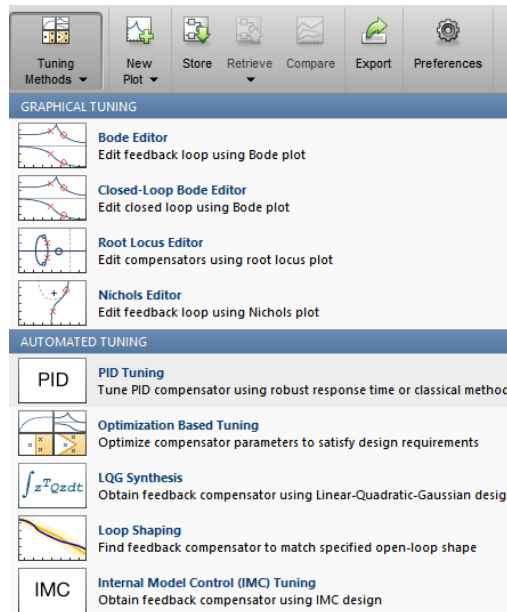


Figura 5.13: Seleção da ferramenta de sintonia em *Control System Designer*.

pensador através do *Update Compensator*.

A configuração selecionada no *PID Tuning* é demonstrada pelo retângulo da Figura 5.14.

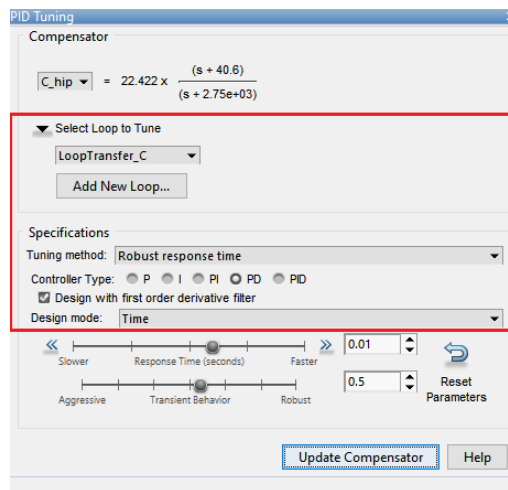


Figura 5.14: Ferramenta *PID Tuning*.

A lei de controlo  $C_{ij}(s)$  fornecida pela ferramenta *Control System Designer* tem o formato de compensador em avanço de primeira ordem, uma notação mais

geral do controlador PD com filtro. Genericamente, o compensador em avanço  $C_{ij}(s)$  de primeira ordem é definido pela Equação (5.20).

$$C_{ij}(s) = K_{ij} \frac{s + Z_{ij}}{s + P_{ij}}, \quad Z_{ij} < P_{ij} \quad (5.20)$$

Onde:

- $i$  e  $j$  são a perna e a junta correspondente do controlador, respetivamente;
- $K_{ij}$  é o ganho do controlador;
- $Z_{ij}$  é o zero do controlador;
- $P_{ij}$  é o polo do controlador;

Para o ajuste manual dos valores de  $K_{ij}$ ,  $Z_{ij}$  e  $P_{ij}$  no *Control System Designer* foi utilizado o *Compensator Editor* demonstrado na Figura 5.15. A ferramenta é acessada ao selecionar com duplo clique a variável de controlo  $C(s)$  no ambiente de análise do *Control System Designer*.

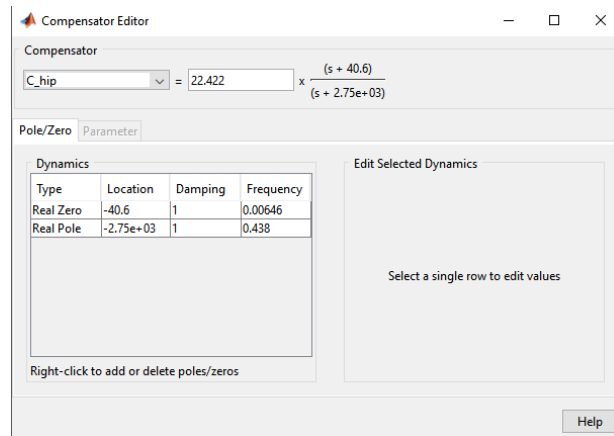


Figura 5.15: Janela da ferramenta *Compensator Editor*.

## Capítulo 6

---

# Testes e resultados

---

*Neste capítulo serão apresentados os testes realizados, parâmetros de desempenho, metodologia utilizada e os resultados obtidos na obra.*

### 6.1 Metodologia dos testes

Para uma melhor organização do trabalho, foram definidas várias etapas a serem seguidas para a aquisição dos resultados do modelo. As etapas estão descritas na lista abaixo e foram executadas de maneira sequencial.

- a) Selecionar os pontos de entrada e saída do sistema para a linearização;
- b) Selecionar os parâmetros que permanecerão em estado estacionário durante a linearização;
- c) Obter os espaços de estados da planta para os diferentes pontos de entrada utilizando o *trim model*;
- d) Encontrar os parâmetros de controle adequados para as plantas linearizadas utilizando o *PID Tuning*;
- e) Testar os parâmetros de controle no modelo não linear do robô e analisar os resultados;
- f) Caso necessário, fazer ajuste fino nos parâmetros do controlador utilizando o *Compensator Editor*;
- g) Realizar o teste para diferentes velocidades, padrões de locomoção e valores de saturação no controlador.

### 6.1.1 Índices de erro

Os índices de erro utilizados para medir a qualidade de resposta do controlador são o *Integral Absolute Error* ( $I_{ae}$ ), *Integral of Time Multiplied by Absolute Error* ( $I_{tae}$ ), *Integral Square Error* ( $I_{se}$ ), o *Integral of Time Multiplied by Square Error* ( $I_{tse}$ ) e as porcentagens de  $I_{ae}$  e  $I_{se}$  em relação à área do ângulo  $\theta_d(t)$  da referência, denominados  $\%I_{ae}$  e  $\%I_{se}$ , respectivamente. As equações dos índices de erros são apresentados nas expressões (6.1)-(6.6).

$$I_{ae_{hip}} = \sum_{i=1}^4 \int_0^{nT_0} |\theta_{d_{i1}} - \theta_{i1}| dt \quad , \quad I_{ae_{knee}} = \sum_{i=1}^4 \int_0^{nT_0} |\theta_{d_{i2}} - \theta_{i2}| dt \quad (6.1)$$

$$I_{tae_{hip}} = \sum_{i=1}^4 \int_0^{nT_0} t |\theta_{d_{i1}} - \theta_{i1}| dt \quad , \quad I_{tae_{knee}} = \sum_{i=1}^4 \int_0^{nT_0} t |\theta_{d_{i2}} - \theta_{i2}| dt \quad (6.2)$$

$$I_{se_{hip}} = \sum_{i=1}^4 \int_0^{nT_0} |\theta_{d_{i1}} - \theta_{i1}|^2 dt \quad , \quad I_{se_{knee}} = \sum_{i=1}^4 \int_0^{nT_0} |\theta_{d_{i2}} - \theta_{i2}|^2 dt \quad (6.3)$$

$$I_{tse_{hip}} = \sum_{i=1}^4 \int_0^{nT_0} t |\theta_{d_{i1}} - \theta_{i1}|^2 dt \quad , \quad I_{tse_{knee}} = \sum_{i=1}^4 \int_0^{nT_0} t |\theta_{d_{i2}} - \theta_{i2}|^2 dt \quad (6.4)$$

$$\%I_{ae} = \sum_{i=1}^4 \sum_{j=1}^2 \frac{\int_0^{nT_0} |\theta_{d_{ij}} - \theta_{ij}| dt}{\int_0^{nT_0} |\theta_{d_{ij}}| dt} * 100 \quad (6.5)$$

$$\%I_{se} = \sum_{i=1}^4 \sum_{j=1}^2 \frac{\int_0^{nT_0} |\theta_{d_{ij}} - \theta_{ij}|^2 dt}{\int_0^{nT_0} |\theta_{d_{ij}}| dt} * 100 \quad (6.6)$$

Onde:

- $i$  e  $j$  são o número da perna e da junta do robô, respectivamente;
- $n$  é o número de ciclos de locomoção do robô durante a simulação;
- $T_0$  é o período do padrão de locomoção adotado pelo robô na simulação;
- $\theta_{d_{ij}} - \theta_{ij}$  pode ser entendido como o valor de  $\theta_{e_{ij}}$ , a diferença entre a referência desejada e o valor atual de cada uma das juntas.

Enquanto os índices  $I_{ae}$  fornecem o somatório do valor  $\theta_e$  no tempo de maneira linear, o  $I_{se}$  soma o quadrado do erro  $\theta_e$  no tempo, o que enfatiza os erros de maior grandeza do sistema. Por sua vez, os erros  $I_{tae}$  e  $I_{tse}$ , além das características em comum com  $I_{ae}$  e  $I_{se}$ , respetivamente, também inserem o tempo como um fator de medida ao atribuir maior peso aos erros conforme  $t$  avança. A medição de  $\%I_{ae}$  e  $\%I_{se}$  foi implementada no intuito de comparar os erros do sistema para diferentes tempos de simulação, onde  $t = n * T_0$ .

O código em MATLAB<sup>®</sup> para o cálculo dos erros da simulação é apresentado no Apêndice B.

### 6.1.2 Índices de rendimento

Os índices de rendimento energético do robô são analisados através da saturação dos atuadores e do cálculo da média de energia gasta por distância percorrida, ou apenas  $E_{av}$  do sistema [40]. O binário máximo aplicado ao sistema é especificado pelos valores de saturação, o que também indica o binário necessário nos motores para que o robô realize a locomoção. O cálculo do  $E_{av}$  é feito através da equação (6.7).

$$E_{av} = \frac{1}{d} \sum_{i=1}^4 \sum_{j=1}^2 \int_0^{T_0} |\tau_{a_{ij}}(t) \dot{\theta}_{ij}(t)| dt \quad (6.7)$$

Onde:

- $\tau_{a_{ij}}(t)$  é o binário aplicado na junta  $j$  da perna  $i$ ;
- $\dot{\theta}_{ij}(t)$  é a velocidade angular da junta  $j$  da perna  $i$ ;
- $T_0$  é o período de locomoção do robô;
- $d$  é a distância percorrida pelo robô durante a simulação.

## 6.2 Resultados obtidos

### 6.2.1 Considerações gerais

Primeiramente, foram realizados dez testes variando os parâmetros  $K_{ij}$ ,  $Z_{ij}$  e  $P_{ij}$  do controlador para o padrão de locomoção do trote adaptado a um beta de  $\beta = 0,5$  seguindo a metodologia proposta. Cada teste executado levou em consideração cinco velocidades diferentes e notou-se para todos os casos, exceto o último, estabilidade no caminhar do robô. Os parâmetros do controlador tiveram uma primeira análise feita em um intervalo de *Response Time* de 0,001 s a 0,05 s

e um *Transient Behaviour* que variou de 0,3 a 0,7, seguidos de ajustes manuais. A Tabela 6.1 mostra os parâmetros utilizados no compensador em cada teste.

Devido a questões temporais o trabalho limitou-se a analisar a saturação dos controladores e o caminhar do robô apenas no melhor caso obtido, sendo todos os outros casos a saturação fixada em  $\pm 20$  Nm. Foram realizados 5 testes de saturação que variaram de  $\pm 2$  Nm até  $\pm 20$  Nm em todas as juntas.

Tabela 6.1: Parâmetros dos testes do controlador.

		Parâmetros					
		$K_{i1}$	$Z_{i1}$	$P_{i1}$	$K_{i2}$	$Z_{i2}$	$P_{i2}$
T e s t e s	1	892,77	-195,4	-21910	558,16	-195,4	21910
	2	2221,8	-409,8	-27270	4614	-409,8	-27270
	3	157	-95,58	-9749	222,28	-95,58	-9749
	4	200	-90	-5500	150	-90	-5500
	5	77	-20,88	-150	204	-20,88	-150
	6	50	-15,88	-150	100	-15,88	-150
	7	20	-15	-120	50	-15	-120
	8	9	-8,12	550,3	20,3	-8,116	-550,3
	9	5	-8	-90	15	-8	-90
	10	15	-20	-1365	20	-20	-1365

## 6.2.2 Resultados dos controladores

No objetivo de obter o melhor controlador em termos de erro e desempenho, foi feita a análise da relação entre  $E_{av}$  multiplicado por  $\%I_{ae}$  e  $E_{av}$  multiplicado por  $\%I_{se}$ . Como se deseja o valor para ambos os parâmetros, o resultado pretendido foi o que apresentou um menor valor para as relações propostas. Para melhor ilustrar os resultados, os gráficos das Figuras 6.1 e 6.2 mostram a comparação entre os testes realizados para diversas velocidades.

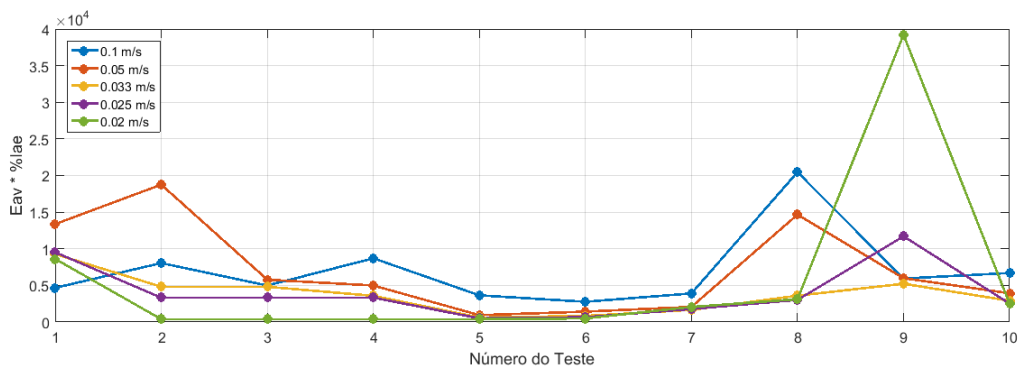


Figura 6.1: Valores de  $E_{av}$  multiplicados por  $\%I_{ae}$  para cada teste.

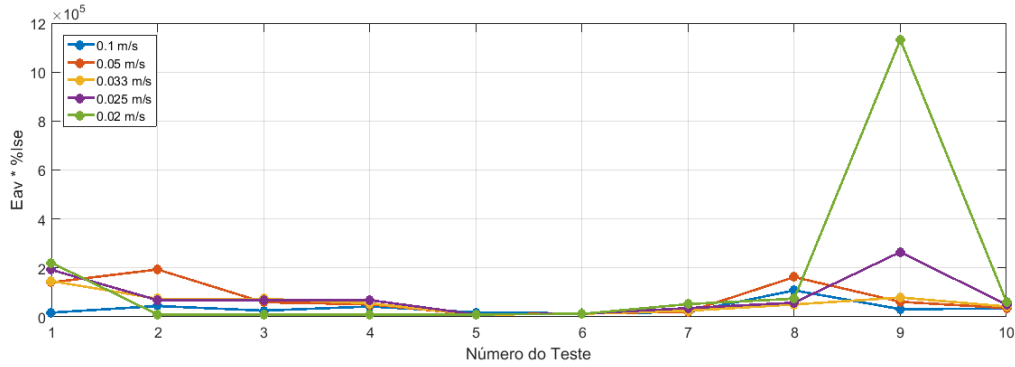


Figura 6.2: Valores de  $E_{av}$  multiplicados por  $\%I_{se}$  para cada teste.

Conclui-se, portanto, que o controlador que obteve o melhor desempenho é o do teste 5, que minimiza a relação entre a média de energia por distância percorrida  $E_{av}$  e os erros  $I_{ae}$  e  $I_{se}$ . A Tabela 6.2 mostra os valores das medidas do sistema para o controlador do teste referido.

Tabela 6.2: Valores das medidas realizadas no teste 5.

$V_F$ ( $\text{ms}^{-1}$ )	$I_{ae_{hip}}$ (graus)	$I_{ae_{knee}}$ (graus)	$I_{se_{hip}}$ (graus)	$I_{se_{knee}}$ (graus)	$\%I_{ae}$	$\%I_{se}$	$d$ (m)	$E_{av}$ ( $\text{Jm}^{-1}$ )
0,1	6,176	2,931	2,170	0,401	0,161	0,045	0,745	22681,668
0,05	6,571	4,344	1,548	0,414	0,096	0,017	1,126	9756,350
0,033	8,110	5,993	1,347	0,441	0,083	0,011	1,030	7425,555
0,025	11,249	7,892	4,216	0,631	0,084	0,021	0,981	5942,543
0,02	13,717	9,802	5,039	0,810	0,083	0,021	1,000	4645,231

A distância percorrida pelo robô também é influenciada pelo controlador, sendo que em todos os testes o ideal é o robô caminhar a distância de 1 m. Porém, diferenças na distância total percorrida foram percebidas, o que acontece devido ao erro do controlador que altera o planejamento proposto pela referência. Em todos os testes o tempo de simulação foi de  $t = n * T_0$  e o comprimento do passo foi de  $L_s = 0,2$  m. A Figura 6.3 mostra os resultados das distâncias percorridas pelo robô.

Percebe-se que, para os parâmetros da Tabela 6.2, o melhor resultado de distância percorrida é no caso em que a velocidade é  $0,05 \text{ ms}^{-1}$ , que apresenta um período de  $T_0 = 4$  s.

Como análise complementar, as medidas isoladas do  $E_{av}$  e percentagem de erro  $\%I_{ae}$  e  $\%I_{se}$  são mostradas nas Figuras 6.4 - 6.6.

Como já esperado, os testes que apresentam menores índices de erro são os que mais exigem energia do sistema. Por sua vez, nos testes que apresentam



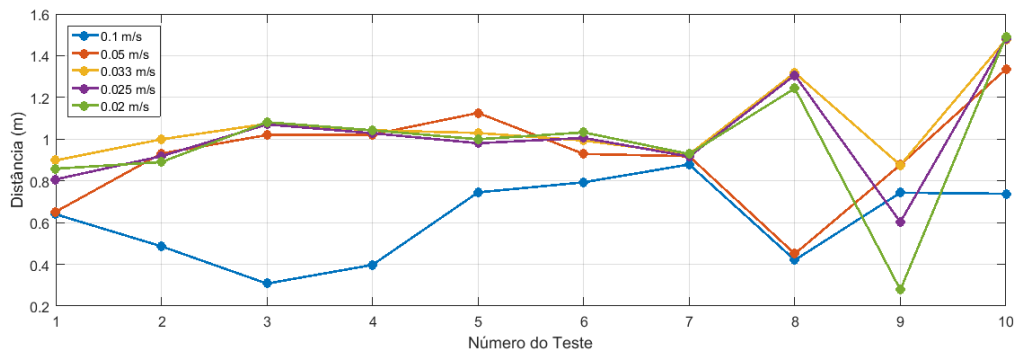


Figura 6.3: Distâncias percorridas pelo robô para cada teste.

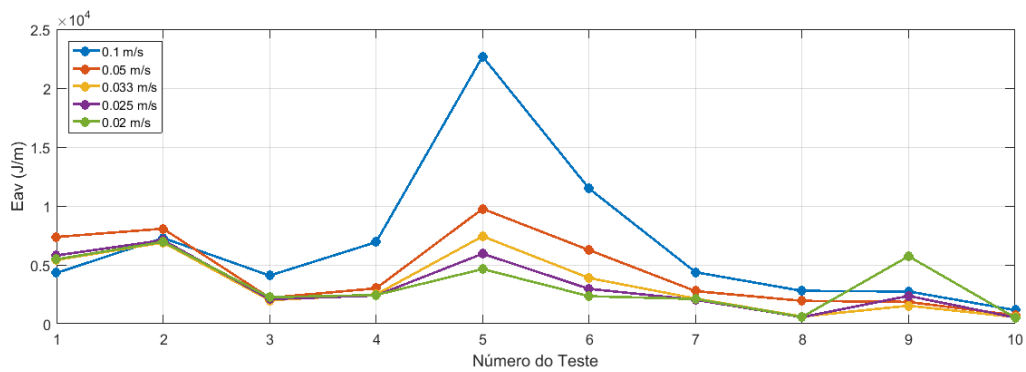


Figura 6.4: Medidas de  $E_{av}$  do robô nos testes.

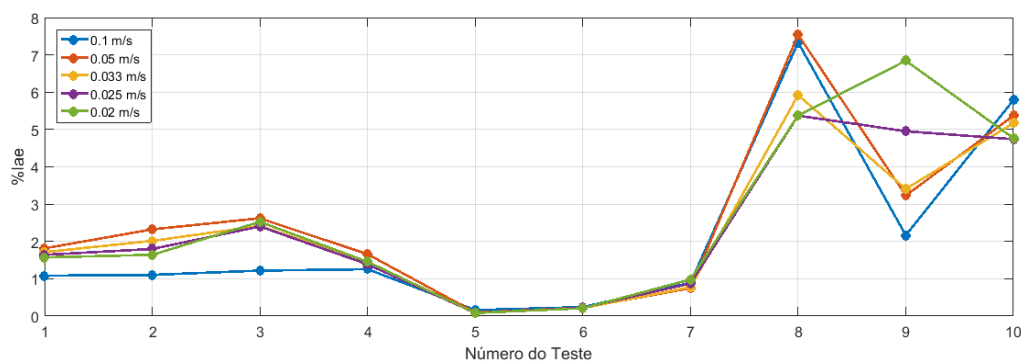


Figura 6.5: Medidas de  $I_{ae}$  do robô nos testes.

menor consumo de energia o sistema torna-se imprevisível a ponto de alterar consideravelmente a sua dinâmica conforme se altera a velocidade de locomoção.

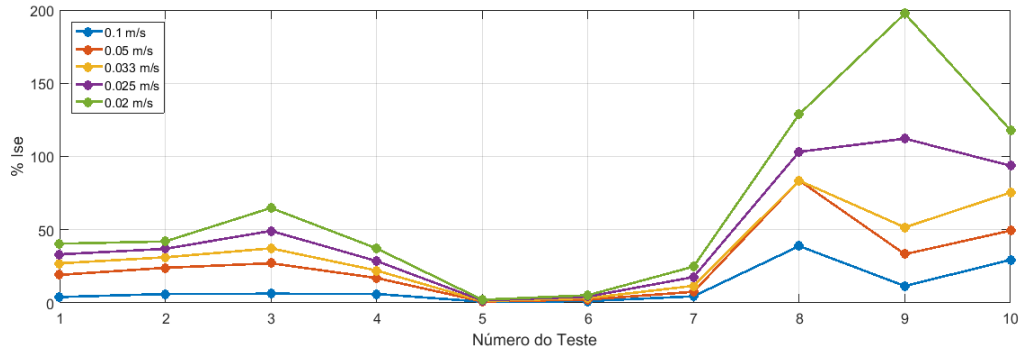


Figura 6.6: Medidas de  $I_{se}$  do robô nos testes.

### 6.2.3 Resultados da saturação

Após definido o controlador do sistema, foi analisado o resultado para diferentes saturações no binário fornecido pelo controlador. A proposta dessa etapa é verificar o menor binário teórico com que o robô consegue executar a trajetória, e qual o impacto no sistema que isso causa. A Tabela 6.3 mostra detalhadamente os índices de erros nas juntas do robô para cada caso de saturação testado.

Tabela 6.3: Índices de erros nas juntas do robô para vários valores de saturação.

Saturação (Nm)	$I_{ae\_anca}$ (graus)	$I_{ae\_joelho}$ (graus)	$I_{tae\_anca}$ (graus)	$I_{tae\_joelho}$ (graus)	$I_{se\_anca}$ (graus)	$I_{se\_joelho}$ (graus)	$I_{tse\_anca}$ (graus)	$I_{tse\_joelho}$ (graus)
20	6,5	4,3	64,1	43,0	1,5	0,4	10,3	3,3
15	6,8	4,8	65,6	47,0	2,3	0,6	10,9	4,0
10	8,8	5,4	83,2	52,2	5,4	1,1	20,6	5,2
5	73,6	11,7	743,1	116,2	652,6	8,9	6396,6	68,3
4	103,7	14,7	1066,9	145,0	1292,4	17,4	13246,5	134,6

Por sua vez, em uma abordagem mais geral, a Tabela 6.4 mostra o aspecto do sistema para as diferentes limitações dos binários.

Tabela 6.4: Aspecto geral do sistema para vários valores de saturação.

Saturação (Nm)	$\%I_{ae\_total}$	$\%I_{se\_total}$	$d$ (m)	$E_{av}$ ( $Jm^{-1}$ )
20	0,096	0,017	1,126	9756,350
15	0,103	0,027	1,071	9869,309
10	0,125	0,058	0,827	9789,381
5	0,753	5,836	0,862	5179,345
4	1,045	11,555	0,905	3974,033

Nota-se que a saturação, como esperado, diminui o  $E_{av}$  do sistema ao custo do

aumento do erro. A Figura 6.7 apresenta a posição do corpo durante a simulação com  $t = nT_0 = 20$  s e uma saturação de 10 Nm.

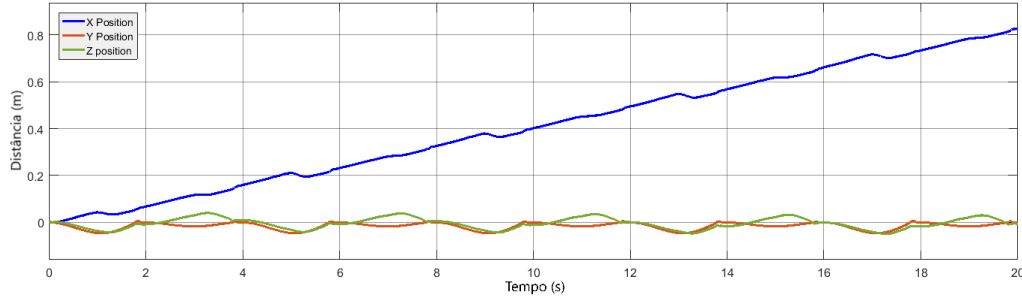


Figura 6.7: Distância percorrida pelo robô durante simulação com  $V_F = 0,05$   $\text{ms}^{-1}$  e  $t = nT_0 = 20$  s.

Para analisar o comportamento dos binários das juntas do robô durante toda a simulação, foram obtidos através das Figuras 6.8a - 6.8g os gráficos do modelo saturado em 10 Nm. Nota-se que o binário neste caso só é efetivamente saturado no instante em que é necessário um movimento mais brusco nas pernas.

No intuito de validar a experiência na questão de fiabilidade do controlador para todas as pernas, as Figuras 6.9a - 6.9d apresentam a trajetória de todas as juntas do robô durante a simulação. Ao analisar as trajetórias, conclui-se que o estudo de apenas uma perna é suficiente para os casos em que os robôs apresentem modelos de pernas idênticos.

Depois da demonstração numérica dos resultados obtidos, a Figura 6.10 mostra em seis imagens o comportamento do robô adotando o padrão de locomoção do trote adaptado para um  $\beta = 0,5$ .

Embora o robô cumpra o objetivo de percorrer a trajetória planeada, o movimento do robô não é feito de maneira totalmente suave. Na opinião do autor, tal fato deve-se a trajetória implementada não apresentar a etapa de suspensão característica do trote animal, o que ocasiona diferenças na sua estabilidade.

Como era de se esperar, as Figuras 6.11 e 6.12 mostram que uma mudança na saturação do controlador acarreta em uma diferença de rendimento e de erro. Quando deseja-se um binário menor nos motores, conseqüentemente o erro do sistema aumenta, porém o  $E_{av}$  diminui. Essa condição é verificada desde que os motores tenham binários suficientes para movimentar o robô para uma distância pretendida. No entanto, para maiores saturações dos binários o oposto acontece: ao passo em que os erros diminuem, o sistema passa a consumir mais energia.

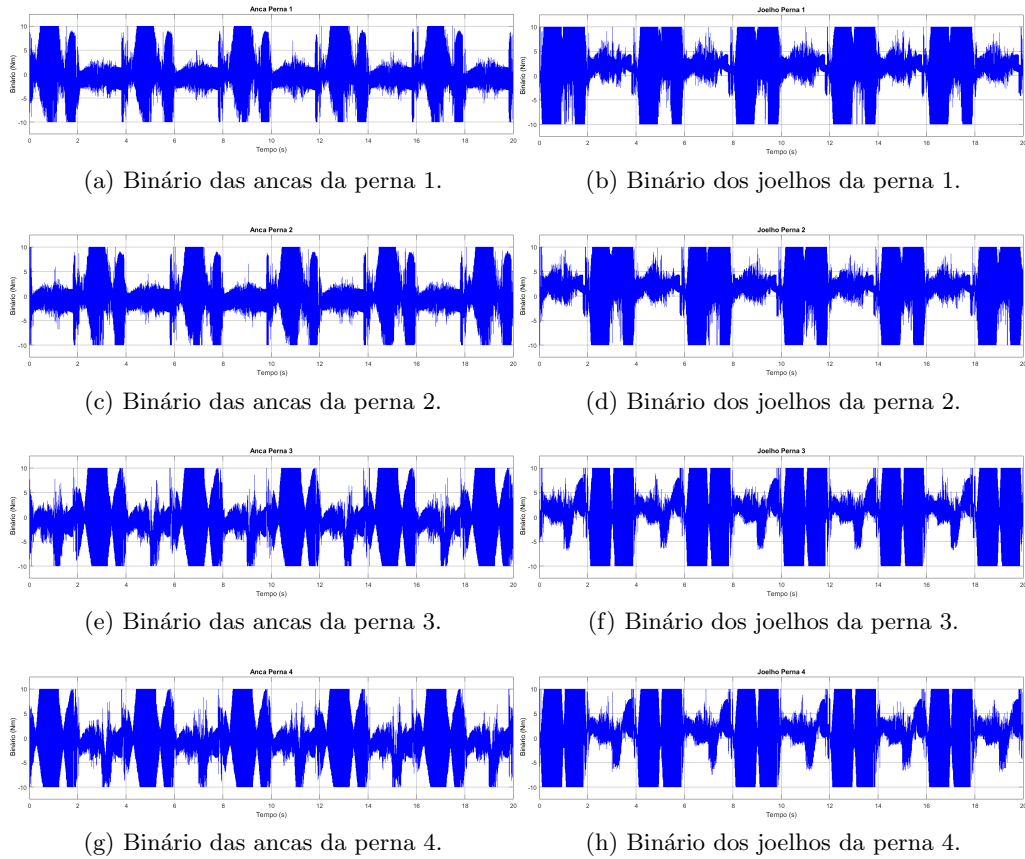
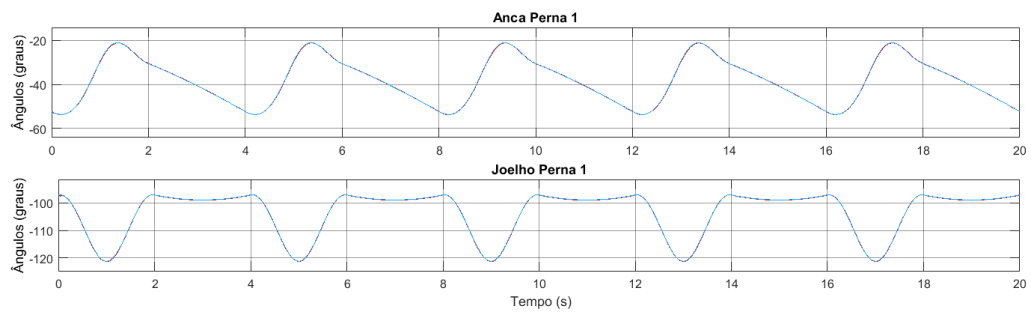


Figura 6.8: Binários aplicados nas pernas do robô durante simulação para uma saturação de 10 Nm.

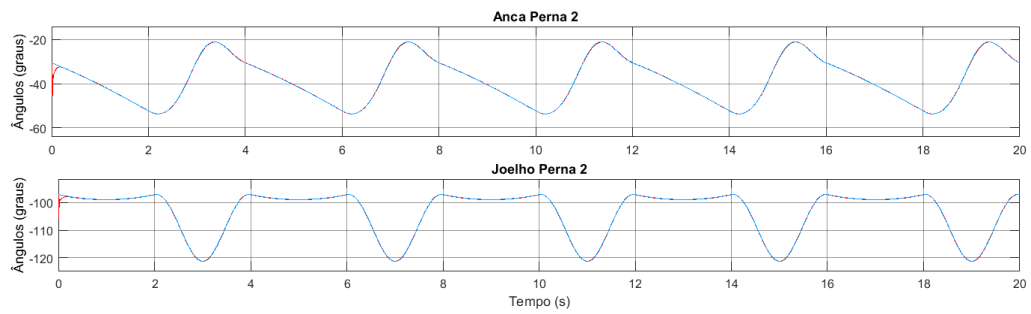
#### 6.2.4 Resultados do padrão de locomoção caminhar

Após feita a análise dos casos do controlador e da saturação, iniciou-se a aplicação dos dois melhores parâmetros de controlo no padrão de locomoção caminhar, com  $\beta = 0,75$  e cuja velocidade nos testes foi diminuída pela metade. Essa etapa foi feita no objetivo de validar a implementação do planeamento de trajetórias para esse caso, e analisar possíveis diferenças entre o rendimento dos padrões de locomoção. Os parâmetros do controlador implementados foram dos testes 5 e 6 da Tabela 6.1. A Tabela 6.5 mostra os valores dos erros detalhados e a Tabela 6.6 mostra o aspeto geral do modelo para ambos os parâmetros do controlador.

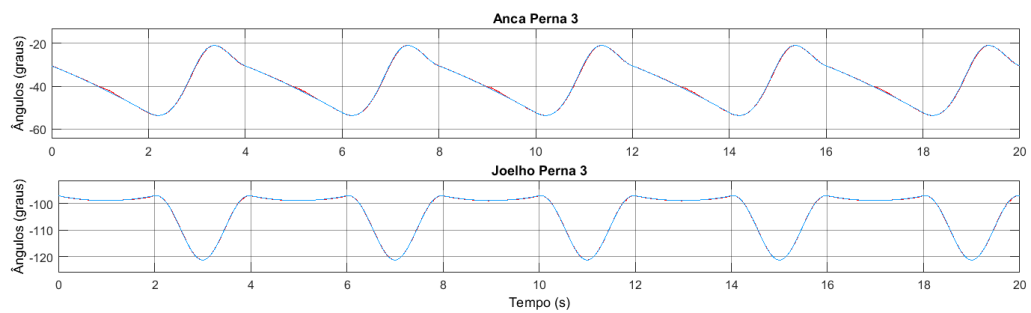
Pela análise das tabelas conclui-se que para os controladores testados o sistema não se comporta bem na velocidade de  $0,05 \text{ ms}^{-1}$ , além de no aspeto geral o padrão de locomoção com  $\beta = 0,5$  ter um melhor rendimento e um menor erro. Porém, como visto na literatura, este tipo de locomoção garante uma maior estabilidade do robô e é útil para terrenos irregulares.



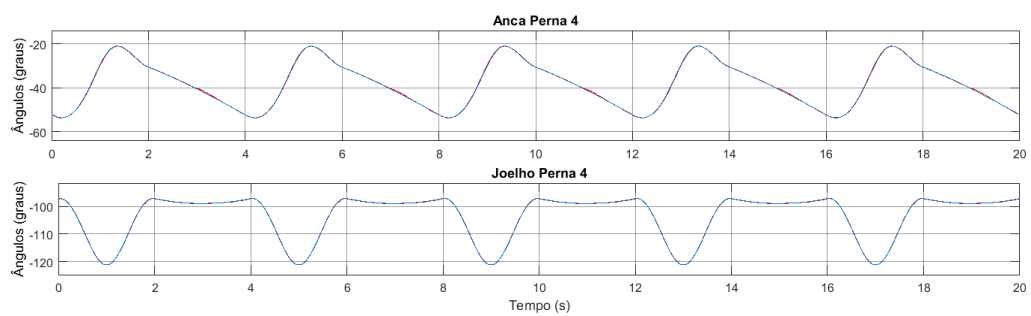
(a) Trajetórias das juntas na perna 1.



(b) Trajetórias das juntas na perna 2.



(c) Trajetórias das juntas na perna 3.



(d) Trajetórias das juntas na perna 4.

Figura 6.9: Trajetórias das juntas do robô para uma saturação de 10 Nm.

A Figura 6.13 mostra em seis imagens a implementação do padrão de locomoção caminhar para o robô. Neste caso, nota-se um bom comportamento para o

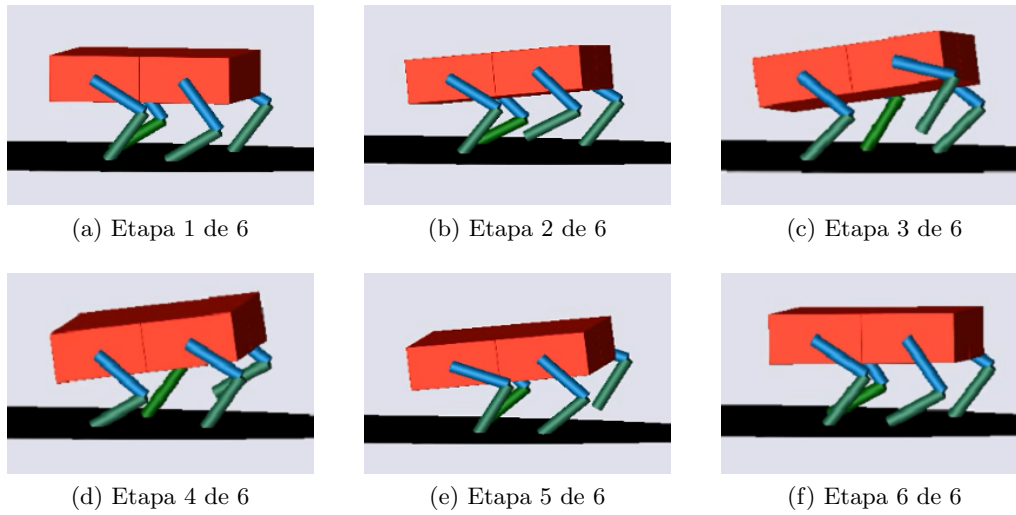


Figura 6.10: Simulação do robô no padrão de locomoção trote adaptado.

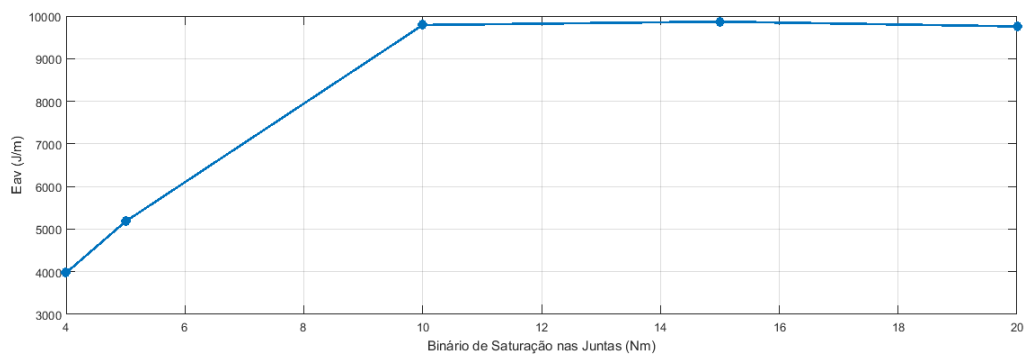


Figura 6.11:  $E_{av}$  do sistema para diferentes valores de saturação.

movimento das patas dianteiras, porém no movimento das patas traseiras o robô acaba por empinar à frente e o movimento das ancas acontece com o pé encostado ao solo durante sua locomoção.

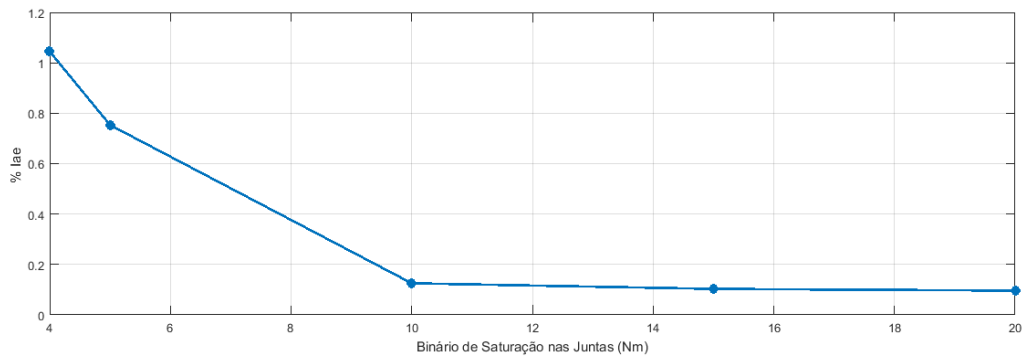


Figura 6.12:  $I_{ae}$  do sistema para diferentes valores de saturação.

Tabela 6.5: Índices de erros para padrão de locomoção caminhar.

Teste	$V_F$ ( $\text{ms}^{-1}$ )	$I_{ae_{anca}}$ (graus)	$I_{ae_{joelho}}$ (graus)	$I_{se_{anca}}$ (graus)	$I_{se_{joelho}}$ (graus)
5	0,05	9,0	8,4	2,8	3,3
	0,025	11,2	11,6	1,9	1,3
	0,016	27,5	18,0	31,7	2,4
	0,0125	37,8	24,1	55,3	3,1
	0,01	51,0	30,2	97,8	3,9
6	0,05	12,9	12,8	5,4	3,8
	0,025	36,2	24,5	53,1	6,4
	0,016	67,9	39,8	147,8	13,3
	0,0125	103,4	53,7	315,9	18,2
	0,01	145,3	68,2	559,0	23,0

Tabela 6.6: Aspetto geral do sistema para padrão de locomoção caminhar.

Teste	$V_F$ ( $\text{ms}^{-1}$ )	$\%I_{ae_{total}}$	$\%I_{se_{total}}$	$d$ (m)	$E_{av}$ ( $\text{Jm}^{-1}$ )
5	0.05	0,155	0,054	0,048	326538,835
	0.025	0,102	0,014	0,504	18590,824
	0.016	0,135	0,101	0,791	8318,740
	0.0125	0,138	0,131	0,862	5792,893
	0.01	0,145	0,182	0,894	4718,898
6	0.05	0,229	0,082	0,257	35003,370
	0.025	0,271	0,266	0,750	6846,190
	0.016	0,320	0,480	0,820	4581,175
	0.0125	0,351	0,746	0,843	3813,654
	0.01	0,381	1,039	0,879	3425,707

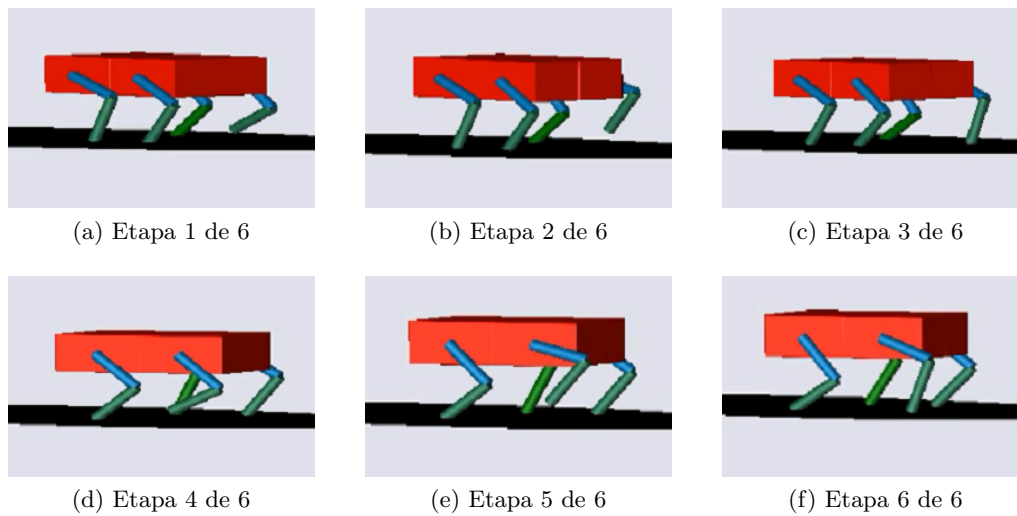


Figura 6.13: Simulação do robô no padrão de locomoção caminhar.

### 6.3 Discussão dos resultados

Por questões temporais, limitou-se a realização dos testes para pouco mais de 60 casos, visto que o MATLAB<sup>®</sup> demora cerca de quatro horas para executar uma simulação de apenas 15 s. Isso acontece devido à complexidade dos cálculos do modelo e às limitações de processamento do computador que realiza os testes, um modelo Dell Inspiron 14 com processador Intel Core i5 5200-U @2,2 GHz, memória RAM de 8 GB, placa de vídeo NVidia GeForce 820M de 2 GB e sistema operacional Windows 10.

Muitos fatores ainda podem ser alterados para a obtenção de um modelo mais fiel, tais como dimensão do robô, peso, parâmetros do solo e sistemas massa-mola-amortecedor. Isso justificou-se durante a fase de execução dos testes, pois se percebeu que vários parâmetros alheios ao controle também influenciavam a resposta final.

O tempo de amostragem é um importante fator para a qualidade da resposta, porém também é um agravante no quesito tempo de simulação. A escolha do tempo de amostragem 0,005 s apresentou uma boa relação entre o rendimento e a qualidade dos testes.

A linearização do modelo apresentou uma boa perspectiva do modelo, gerando bons resultados no *Control System Designer* para um tempo de resposta de aproximadamente 0,001 s e um comportamento transitório de 0.5. Porém devido a evidente limitação de um modelo linear, foram necessários ajustes finos no controlador para minimizar as diferenças. O parâmetro fornecido pela ferramenta *Control System Designer* que menos se aproximou de uma boa resposta no mo-



delo real foi o ganho do controlador, que na maioria casos teve de ser corrigido para um valor diferente.

Quanto aos parâmetros dos controladores, é importante notar que para a condição imposta, os valores que apresentaram melhores resultados são os que estavam entre  $50 < K_{ij} < 100$ ,  $10 < Z_{ij} < 30$  e  $P_{ij} \approx 10 * Z_{ij}$ . Além do mais, a saturação imposta nos testes garante uma resposta satisfatória do controlador para motores com binários a partir de 4 Nm.

O código com os parâmetros do robô utilizado neste trabalho, cuja inicialização é necessária para o funcionamento do modelo em Simulink™/SimMechanics™, encontra-se no Apêndice C.

## Capítulo 7

---

# Conclusão e desenvolvimentos futuros

---

*Neste capítulo é feita uma conclusão geral acerca do trabalho desenvolvido e das competências adquiridas, além de sugerir tópicos para desenvolvimento futuro desta obra.*

### 7.1 Balanço da obra

Durante os meses de trabalho na obra as metas alcançadas pelo trabalho foram:

- Implementação de um sistema de controlo através de ferramentas do MATLAB<sup>®</sup>/Simulink<sup>™</sup>;
- Implementação do planeamento de trajetórias e dos padrões de locomoção trote adaptado e caminhar no modelo do robô;
- Implementação de um modelo cinemático para o robô;
- Análise da eficiência e demais parâmetros dos controladores do robô.

Porém, principalmente devido a questões temporais e limitações de *hardware*, os objetivos propostos que não constam no escopo do trabalho são:

- Visualização da locomoção do robô em tempo real em Simulink<sup>™</sup>;
- Implementação do padrão de locomoção galope;
- Estudo detalhado e melhoramento de variáveis do modelo que influenciaram na dinâmica da simulação, tais como o sistema pé-solo, tempo de amostragem e parâmetros de configuração da simulação no Simulink<sup>™</sup>;

- Estudo da margem de estabilidade do modelo para os diferentes tipos de padrões de locomoção.

## 7.2 Conclusão

A robótica móvel e em especial os robôs de locomoção por pernas são sistemas que necessitam de vários testes e definições antes de serem implementados fisicamente. Ao abordar um tema que envolva a simulação desse tipo de robô, abre-se a oportunidade para uma variedade de diferentes tipos de sistema serem testados. Com o estudo foi concluído que a versatilidade das simulações é demasiado importante para encontrar bons parâmetros de funcionamento e assim definir melhor as métricas do projeto de um robô físico.

Através da análise do controlo foi possível encontrar uma boa resposta com um binário relativamente baixo no sistema. Além do mais, o sistema linearizado forneceu uma aproximação inicial para encontrar tais valores desejados. Consoante o fator energético, o padrão implementado do trote forneceu ao sistema uma melhor resposta frente ao caminhar para os casos analisados. Por fim, a implementação de um planeamento de trajetórias juntamente com a cinemática inversa no modelo deu maior versatilidade às experiências se comparado ao modelo existente em Castro [37].

## 7.3 Desenvolvimentos futuros

Para o presente modelo, pontos chave que requerem melhorias e podem resultar em mais trabalhos são:

- A implementação de diferentes padrões de locomoção no robô, tais como o galope;
- Um sistema de controlo de binário e velocidade no sistema;
- O estudo do rendimento do robô alterando parâmetros físicos como tamanho e peso;
- Implementação de um sistema de leitura e correção de estabilidade;
- Implementação de algoritmos de padrão de locomoção não periódicos.

Os casos acima mencionados foram previamente encontrados em referências da área e, na opinião do autor, têm grande valia para posterior implementação no modelo.

---

# Bibliografia

---

- [1] RAIBERT, Marc H. "Legged robots that balance". MIT press, 1986. [citado na p. 1]
- [2] S. P. O. Carvalho, "Optimização dos parâmetros de um robô hexápode através de algoritmos genéticos". Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2008. [citado na p. 1]
- [3] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka and S. Kajita, "Cybernetic human HRP-4C,"2009 9th IEEE-RAS International Conference on Humanoid Robots, Paris, 2009, pp. 7-14. DOI: 10.1109/ICHR.2009.5379537 [citado na p. 5]
- [4] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki and K. Fujimura, "The intelligent ASIMO: system overview and integration,"IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 2478-2483 vol.3. DOI: 10.1109/IRDS.2002.1041641 [citado na p. 5]
- [5] Proctor Honda. "ASIMO by Honda: The World's Most Advanced Humanoid Robot". [Online]. Disponível em: <http://www.proctorhonda.com/blog/2013/august/6/asimo-the-honda-robot.htm>. Acesso em: 14/04/2017. [citado na p. 6]
- [6] ROBOManiac. "Singer Japanese Robot HRP-4". [Online]. Disponível em: [http://robomaniac.com.ua/news/humanoids/japanese\\_singing\\_robot\\_hrp4\\_news.html](http://robomaniac.com.ua/news/humanoids/japanese_singing_robot_hrp4_news.html). Acesso em: 14/04/2017. [citado na p. 6]
- [7] Wiki.NUS "Current uses of robots". [Online]. Disponível em: <https://wiki.nus.edu.sg/display/cs1105groupreports/Current+uses+of+robots>. Acesso em: 14/04/2017. [citado na p. 6]
- [8] R. Tellez et al., "Reem-B: An autonomous lightweight human-size humanoid robot,"Humanoids 2008 - 8th IEEE-RAS International Conference on Huma-

- noid Robots, Daejeon, 2008, pp. 462-468. DOI: 10.1109/ICHR.2008.4755995 [citado na p. 6]
- [9] T. Ishida, "Development of a small biped entertainment robot QRIO," *Micro-Nanomechatronics and Human Science*, 2004 and *The Fourth Symposium Micro-Nanomechatronics for Information-Based Society*, 2004., 2004, pp. 23-28. DOI: 10.1109/MHS.2004.1421265 [citado na p. 6]
- [10] A. S. Baskoro and M. G. Priyono, "Design of humanoid robot stable walking using inverse kinematics and zero moment point," *2016 International Electronics Symposium (IES)*, Denpasar, Indonesia, 2016, pp. 335-339. DOI: 10.1109/ELECSYM.2016.7861027 [citado na p. 7]
- [11] Q. Cao, A. T. van Rijn and I. Poulakakis, "On the control of gait transitions in quadrupedal running," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 5136-5141. DOI: 10.1109/IROS.2015.7354100 [citado na p. 1, 7]
- [12] F. Gao, C. Qi, Q. Sun, X. Chen and X. Tian, "A quadruped robot with parallel mechanism legs," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 2566-2566. DOI: 10.1109/ICRA.2014.6907223 [citado na p. 1, 7]
- [13] M. Hutter et al., "ANYmal - a highly mobile and dynamic quadrupedal robot," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, 2016, pp. 38-44. DOI: 10.1109/IROS.2016.7758092 [citado na p. xv, 1, 7, 8]
- [14] H. Suzuki, H. Nishi, A. Aburadani and S. Inoue, "Animal Gait Generation for Quadrupedal Robot," *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, Kumamoto, 2007, pp. 20-20. DOI: 10.1109/ICICIC.2007.169 [citado na p. xv, 7, 9]
- [15] T. Kamioka, T. Watabe, M. Kanazawa, H. Kaneko and T. Yoshiike, "Dynamic gait transition between bipedal and quadrupedal locomotion," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 2195-2201. DOI: 10.1109/IROS.2015.7353671 [citado na p. 7]
- [16] Jiaqi Zhang and Qijun Chen, "Learning based gaits evolution for an AIBO dog," *2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 1523-1526. DOI: 10.1109/CEC.2007.4424653 [citado na p. 9]
- [17] N. Suzuki and Y. Yamamoto, "Pursuing entertainment aspects of SONY AIBO quadruped robots," *2011 Fourth International Conference on Modeling*,

- Simulation and Applied Optimization, Kuala Lumpur, 2011, pp. 1-5. DOI: 10.1109/ICMSAO.2011.5775611 [citado na p. 9]
- [18] W. Bosworth, S. Kim and N. Hogan, "The MIT super mini cheetah: A small, low-cost quadrupedal robot for dynamic locomotion," 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), West Lafayette, IN, 2015, pp. 1-8. DOI: 10.1109/SSRR.2015.7443018 [citado na p. xv, 9, 10, 11]
- [19] S. Rutishauser, A. Sprowitz, L. Righetti and A. J. Ijspeert, "Passive compliant quadruped robot using Central Pattern Generators for locomotion control," 2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, AZ, 2008, pp. 710-715. DOI: 10.1109/BIOROB.2008.4762878 [citado na p. 11]
- [20] EPFL. "Cheetah-Cub - a compliant quadruped robot". [Online]. Available at: <http://biorob.epfl.ch/cheetah>. Accessed on: 03/16/2017. [citado na p. xv, 11]
- [21] de Oliveira, Luiz Fernando Pinto. Modelação, Simulação e Implementação de Padroes de Locomoção para Robôs Hexápodes. Diss. Instituto Superior de Engenharia do Porto, 2016. [citado na p. 12]
- [22] M. Görner and G. Hirzinger, "Analysis and Evaluation of the Stability of a Biologically Inspired, Leg Loss Tolerant Gait for Six- and Eight-Legged Walking Robots", in IEEE 2010 International Conference on Robotics and Automation, 2010. [citado na p. xv, 12]
- [23] A. Roennau, G. Heppner, M. Nowicki and R. Dillmann, "LAURON V: A versatile six-legged walking robot with advanced maneuverability," 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon, 2014, pp. 82-87. DOI: 10.1109/AIM.2014.6878051 [citado na p. xv, 13, 14]
- [24] D. Belter, K. Walas, "A Compact Walking Robot – Flexible Research and Development Platform," in Advances in Intelligent Systems and Computing, J.Kacprzyk (Ed.), Springer, 2014. [citado na p. 13]
- [25] M. Konyev et al., "Walking robot ANTON: Design, simulation, experiments" in Proc. of 11th Int. Conf. on Climbing and Walking Robots (CLAWAR), 2008. [citado na p. 13]
- [26] P. Manoonpong, F. Woergoetter, "Biologically-Inspired Reactive Walking Machine AMOS-WD06," in Proc. of Int. Symposium on Adaptive Motion of Animals and Machines (AMAM), 2008. [citado na p. 13]
- [27] Barasuol, Victor. "Um sistema de controle reativo para locomoção de robôs quadrúpedes". Tese de Doutorado. Universidade Federal de Santa Catarina, 2013. [citado na p. xv, 15, 16, 17, 20, 21, 23]

- [28] Eadweard Muybridge. "Horse in Motion, ca. 1886". [Online]. Available at: [http://www.hrc.utexas.edu/exhibitions/permanent/windows/southeast/eadweard\\_muybridge.html](http://www.hrc.utexas.edu/exhibitions/permanent/windows/southeast/eadweard_muybridge.html). Accessed on: 03/22/2017. [citado na p. xv, 16]
- [29] Kajita, Shuuji, and Bernard Espiau. "Legged robots." Springer handbook of robotics. Springer Berlin Heidelberg, 2008. 361-389. [citado na p. xv, 16, 17, 20]
- [30] dos Santos Silva, Manuel Fernando. "Sistemas robóticos de locomoção multipernas". Tese de Doutorado. Faculdade de Engenharia da Universidade do Porto, 2005. [citado na p. 16, 17]
- [31] Matsuoka, Kiyotoshi. "Sustained oscillations generated by mutually inhibiting neurons with adaptation." Biological cybernetics, 1985. [citado na p. 18]
- [32] Zielińska, Teresa. "Coupled oscillators utilised as gait rhythm generators of a two-legged walking machine." Biological Cybernetics, 1996. [citado na p. 18]
- [33] Muybridge, Eadweard. "Animals in motion". Courier Corporation, 2012. [citado na p. xv, 19, 20, 21, 22]
- [34] H. Igarashi, T. Machida, F. Harashima and M. Kakikura. "Free gait for quadruped robots with posture control,". 9th IEEE International Workshop on Advanced Motion Control. 2006. DOI: 10.1109/AMC.2006.1631698 [citado na p. 23]
- [35] M. Huber, W. S. MacDonald and R. A. Grupen, "A control basis for multilegged walking," Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, 1996, pp. 2988-2993 vol.4. DOI: 10.1109/ROBOT.1996.509166 [citado na p. 23]
- [36] Kyung-Min Jeong, Tae-Seok Yang and Jun-Ho Oh, "A study on the support pattern of a quadruped walking robot for aperiodic motion," Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, Pittsburgh, PA, 1995, pp. 308-313 vol.3. DOI: 10.1109/IROS.1995.525901 [citado na p. xvi, 23, 24]
- [37] Castro, Tomás Figueira Vieira Souto. "Simulação de robôs quadrúpedes utilizando o SimMechanics". Diss. 2012. [citado na p. xvi, 2, 4, 25, 26, 27, 29, 68]
- [38] Craig, John J. "Introduction to robotics: mechanics and control". Vol. 3. Upper Saddle River: Pearson Prentice Hall, 2005. [citado na p. 27, 39]
- [39] Mathworks. "Simscape™ Multibody™ Reference". Version R2016b. Manual de usuário, 2016. [citado na p. 29]

- [40] Machado, JA Tenreiro, and Manuel F. Silva. "An overview of legged robots." International symposium on mathematical methods in engineering. Ankara, Turkey: MME Press, 2006. [citado na p. 55]





## Apêndice A

---

# Equação obtida dos modelos das ancas e joelhos em espaço de estados

---

### A.1 Espaço de estados das ancas

$$A_{17 \times (1-8)} = \begin{bmatrix} -1.101e^7 & 3.415e^6 & 3.673e^6 & 6.686e^6 & -2.846e^6 & 3.955e^6 & -2.059e^6 & 4.512e^5 \\ 1.022e^6 & -1.148e^7 & 4.294e^6 & -4.615e^6 & -1.318e^6 & -2.47e^6 & 1.872e^6 & 4.788e^6 \\ 1.839e^6 & 3.222e^6 & -1.349e^7 & -1.909e^6 & 2.93e^4 & -1.912e^6 & 3.423e^5 & 1.458e^6 \\ 5.426e^6 & -2.966e^6 & -3.725e^6 & -9.981e^6 & 5.572e^6 & -5.267e^6 & 3.202e^6 & -3.054e^6 \\ -4.837e^6 & 1.298e^6 & 3.405e^6 & 6.496e^6 & -3.75e^6 & 3.517e^6 & -2.034e^6 & 2.145e^6 \\ 4.789e^5 & -3.772e^6 & -4.067e^6 & -6.158e^6 & 1.945e^6 & -3.425e^6 & 2.058e^6 & 5.636e^5 \\ 6.68e^5 & 3.819e^6 & -2.634e^6 & 6.531e^6 & -3.787e^6 & 2.983e^6 & -2.274e^6 & 2.425e^6 \\ 2.939e^6 & -2.938e^6 & 2.028e^6 & -3.872e^6 & 2.066e^6 & -1.877e^6 & 1.322e^6 & -1.106e^6 \\ -2.575e^5 & 1.79e^6 & 3.045e^6 & 1.683e^6 & 3.233e^5 & 1.152e^6 & -5.609e^5 & -1.607e^6 \dots \\ -3.661e^5 & 1.817e^6 & 4.33e^5 & 1.565e^6 & -1.922e^5 & 8.775e^5 & -5.514e^5 & -6.262e^5 \\ -2.535e^6 & 3.676e^6 & 2.482e^6 & 3.165e^6 & -3.988e^4 & 1.972e^6 & -1.069e^6 & -1.907e^6 \\ 9.729e^5 & -1.399e^6 & -1.047e^6 & -1.311e^6 & 7.515e^4 & -8.085e^5 & 4.4e^5 & 6.93e^5 \\ -3.775e^6 & 5.655e^6 & 3.7e^6 & 5.05e^6 & -2.309e^5 & 3.103e^6 & -1.707e^6 & -2.758e^6 \\ -1.986e^6 & 2.867e^6 & 1.88e^6 & 2.581e^6 & -1.343e^5 & 1.585e^6 & -8.709e^5 & -1.383e^6 \\ -3.423e^5 & 4.791e^5 & 3.104e^5 & 4.312e^5 & -2.335e^4 & 2.649e^5 & -1.454e^5 & -2.297e^5 \\ -6.5e^5 & 9.841e^5 & 6.542e^5 & 8.826e^5 & -4.063e^4 & 5.424e^5 & -2.982e^5 & -4.816e^5 \\ 7.096e^5 & -1.075e^6 & -7.145e^5 & -9.636e^5 & 4.383e^4 & -5.922e^5 & 3.256e^5 & 5.266e^5 \end{bmatrix}$$

$$\begin{array}{cccccccc}
 & & & & & & & A_{17X(9-16)} = \\
 \dots & \left[ \begin{array}{cccccccc}
 -1.644e^6 & -3.294e^5 & -2.039e^6 & 9.229e^5 & -3.113e^6 & -1.587e^6 & -2.681e^5 & -5.462e^5 \\
 4.569e^6 & 8.17e^5 & 6.289e^6 & -2.198e^6 & 8.653e^6 & 4.385e^6 & 7.597e^5 & 1.48e^6 \\
 2.754e^6 & 9.519e^5 & 4.074e^6 & -1.446e^6 & 5.794e^6 & 2.944e^6 & 5.023e^5 & 9.98e^5 \\
 -2.17e^5 & -3.76e^5 & -9.683e^5 & -3.728e^4 & -9.2e^5 & -4.56e^5 & -8.502e^4 & -1.377e^5 \\
 1.321e^5 & 1.992e^5 & 5.942e^5 & 4.197e^4 & 5.269e^5 & 2.596e^5 & 4.97e^4 & 7.651e^4 \\
 1.937e^6 & 2.852e^5 & 2.431e^6 & -1.014e^6 & 3.555e^6 & 1.807e^6 & 3.091e^5 & 6.176e^5 \\
 1.112e^6 & 6.741e^5 & 2.15e^6 & -4.973e^5 & 2.764e^6 & 1.397e^6 & 2.424e^5 & 4.625e^5 \\
 -2.599e^5 & -2.635e^5 & -6.712e^5 & 9.244e^4 & -7.983e^5 & -4.019e^5 & -7.056e^4 & -1.302e^5 \\
 -1.916e^6 & -4.727e^5 & -2.718e^6 & 9.558e^5 & -3.793e^6 & -1.924e^6 & -3.312e^5 & -6.506e^5 \\
 -8.623e^5 & -1.532e^5 & -1.153e^6 & 4.312e^5 & -1.626e^6 & -8.25e^5 & -1.421e^5 & -2.798e^5 \\
 -2.499e^6 & -5.634e^5 & -3.47e^6 & 1.253e^6 & -4.872e^6 & -2.472e^6 & -4.253e^5 & -8.372e^5 \\
 9.513e^5 & 2.119e^5 & 1.314e^6 & -4.789e^5 & 1.85e^6 & 9.391e^5 & 1.615e^5 & 3.182e^5 \\
 -3.713e^6 & -8.205e^5 & -5.13e^6 & 1.866e^6 & -7.215e^6 & -3.662e^6 & -6.298e^5 & -1.241e^6 \\
 -1.878e^6 & -4.149e^5 & -2.593e^6 & 9.441e^5 & -3.648e^6 & -1.852e^6 & -3.184e^5 & -6.273e^5 \\
 -3.127e^5 & -6.912e^4 & -4.317e^5 & 1.572e^5 & -6.075e^5 & -3.083e^5 & -5.302e^4 & -1.045e^5 \\
 -6.488e^5 & -1.434e^5 & -8.964e^5 & 3.26e^5 & -1.261e^6 & -6.398e^5 & -1.1e^5 & -2.168e^5 \\
 7.091e^5 & 1.567e^5 & 9.797e^5 & -3.563e^5 & 1.378e^6 & 6.993e^5 & 1.203e^5 & 2.369e^5
 \end{array} \right] \dots
 \end{array}$$

$$A_{17X17} = \dots \begin{bmatrix} 5.985e^5 \\ -1.626e^6 \\ -1.093e^6 \\ 1.516e^5 \\ -8.455e^4 \\ -6.779e^5 \\ -5.068e^5 \\ 1.426e^5 \\ 7.14e^5 \\ 3.072e^5 \\ 9.188e^5 \\ -3.492e^5 \\ 1.361e^6 \\ 6.885e^5 \\ 1.147e^5 \\ 2.379e^5 \\ -2.6e^5 \end{bmatrix} ;$$

$$B_{17X1} = \begin{bmatrix} -509.4 \\ 1359 \\ -482.4 \\ 252.6 \\ -184.1 \\ 378.5 \\ -378.4 \\ 808.7 \\ -43.48 \\ -244.3 \\ -377.4 \\ 148.1 \\ -612.2 \\ -313.6 \\ -52.77 \\ -106.1 \\ 115.8 \end{bmatrix} ;$$

$$C_{1X(1-9)} = \begin{bmatrix} 7.322 & 0.1517 & 1.419 & 11.34 & 5.136 & -10.74 & 2.875 & 5.481 \end{bmatrix} \dots$$

$$C_{1X(10-17)} = \dots \begin{bmatrix} -2.048 & 1.917 & -30.2 & -3.263 & 23.87 & -22.57 & -17.75 & -8.627 & -15.91 \end{bmatrix} ;$$

$$D_{1X1} = \begin{bmatrix} 0 \end{bmatrix} ;$$

## A.2 Espaço de estado dos joelhos

$$A_{5X5} = \begin{bmatrix} -2.761e^7 & 6.24e^4 & 4.063e^6 & -1.976e^6 & 3.751e^6 \\ 7.885e^5 & -6.824e^6 & -1.432e^6 & -1.587e^6 & -1.305e^6 \\ 8.668e^6 & 6244 & -2.142e^6 & 8.11e^6 & -2.031e^6 \\ -7.317e^6 & 4123 & 3.043e^6 & -1.743e^7 & 2.929e^6 \\ 8.005e^6 & -2702 & -1.988e^6 & 7.556e^6 & -1.885e^6 \end{bmatrix} ;$$

$$B_{5X1} = \begin{bmatrix} 239.2 \\ -595.1 \\ 1256 \\ 111.3 \\ 1146 \end{bmatrix} ;$$

$$C_{1X5} = \begin{bmatrix} -0.0319 & 0.05099 & 38.19 & -0.3406 & -41.78 \end{bmatrix};$$

$$D_{1X1} = \begin{bmatrix} 0 \end{bmatrix};$$

## Apêndice B

---

# Código para obtenção dos índices resultantes da simulação

---

```
%-----%
% obter os binarios maximos nas juntas
max_h_1 = max(abs(torque_hip_1));

max_h_2 = max(abs(torque_hip_2));

max_h_3 = max(abs(torque_hip_3));

max_h_4 = max(abs(torque_hip_4));

max_k_1 = max(abs(torque_knee_1));

max_k_2 = max(abs(torque_knee_2));

max_k_3 = max(abs(torque_knee_3));

max_k_4 = max(abs(torque_knee_4));

torque_hip = max([max_h_1,max_h_2,max_h_3,max_h_4]);

torque_knee = max([max_k_1,max_k_2,max_k_3,max_k_4]);

%-----%
```

```

area_total = area_hip_1(end) + area_hip_2(end) + area_hip_3(end) ...
            + area_hip_4(end) + area_knee_1(end) + area_knee_2(end)...
            + area_knee_3(end) + area_knee_4(end) ;

```

```

% calculo dos erros nas ancas e nos joelhos

```

```

Iae_hip = Iae_hip_1(end) + Iae_hip_2(end) + ...
          Iae_hip_3(end) + Iae_hip_4(end);

```

```

Iae_knee = Iae_knee_1(end) + Iae_knee_2(end) + ...
           Iae_knee_3(end) + Iae_knee_4(end);

```

```

Itae_hip = Itae_hip_1(end) + Itae_hip_2(end) + ...
           Itae_hip_3(end) + Itae_hip_4(end);

```

```

Itae_knee = Itae_knee_1(end) + Itae_knee_2(end) + ...
            Itae_knee_3(end) + Itae_knee_4(end);

```

```

Ise_hip = Ise_hip_1(end) + Ise_hip_2(end) + ...
          Ise_hip_3(end) + Ise_hip_4(end);

```

```

Ise_knee = Ise_knee_1(end) + Ise_knee_2(end) + ...
           Ise_knee_3(end) + Ise_knee_4(end);

```

```

Itse_hip = Itse_hip_1(end) + Itse_hip_2(end) + ...
           Itse_hip_3(end) + Itse_hip_4(end);

```

```

Itse_knee = Itse_knee_1(end) + Itse_knee_2(end) + ...
            Itse_knee_3(end) + Itse_knee_4(end);

```

```

%-----%

```

```

% calculo dos indices de erro totais

```

```

Iae_total = Iae_hip + Iae_knee;

```

```

Itae_total = Itae_hip + Itae_knee;

```

```

Ise_total = Ise_hip + Ise_knee;

```

```

Itse_total = Itse_hip + Itse_knee;

```

```

perc_Iae = (Iae_total / area_total)*100;

```

```
perc_Itae = (Itae_total / area_total)*100;

perc_Ise = (Ise_total / area_total)*100;

perc_Itse = (Itse_total / area_total)*100;

%-----%
% calculo do rendimento do sistema
Eav_total = Eav_hip_1(end) + Eav_hip_2(end) + Eav_hip_3(end) ...
            + Eav_hip_4(end) + Eav_knee_1(end) + Eav_knee_2(end) ...
            + Eav_knee_3(end) + Eav_knee_4(end) ;

Eav_ciclo = Eav_total / nc ;

Distancia = distance_travelled(end,1);

Eav_final = Eav_ciclo / abs(distance_travelled(end,1));

% fim do codigo
%-----%
```





## Apêndice C

---

# Código para inicialização de parâmetros do modelo em Simulink<sup>TM</sup>

---

```
clear all
close all
clc

%-----%
% Definicao da Caixa de Dialogo:

promptdlg1 = 'algoritmo (1 - Italo / 2 - Thomas)';
defaultans1 = '1';

promptdlg2 = 'Gait (1 - Caminhar / 2 - Trotar)(Valido p/ algoritmo 1)';
defaultans2 = '1';

promptdlg3 = 'Comprimento do Passo (Valido p/ algoritmo 1)';
defaultans3 = '0.2';

promptdlg4 = 'Periodo do Ciclo (Valido p/ algoritmo 1)';
defaultans4 = '4';

promptdlg5 = 'Quantidade de Ciclos (Valido p/ algoritmo 1)';
defaultans5 = '5';
```

```

namedlg = 'Parametros de Simulacao';

promptdlg = {promptdlg1,promptdlg2,promptdlg3,...
             promptdlg4,promptdlg5};

defaultans = {defaultans1, defaultans2, defaultans3,...
             defaultans4, defaultans5};

answer = inputdlg(promptdlg,namedlg,1,defaultans,'on');

%-----%

%-----%
% atribuicao das opcoes nas variaveis:

    algor = str2double(answer{1}); % algoritmo utilizado
    gait  = str2double(answer{2}); % padrao de locomocao
    LS    = str2double(answer{3}); % comprimento do passo
    T     = str2double(answer{4}); % periodo de ciclo
    nc    = str2double(answer{5}); % numero de ciclos

%-----%

%-----%
% Parametros Fisicos do modelo:

body_x_ic = 0;      % Posicao CG Corpo X e condicao inicial
body_y_ic = 0.26;  % Posicao CG Corpo Y
body_z_ic = 0;     % Posicao CG Corpo Z

n = 0.9; % Peso da mola (forca normal nao linear)
m = 0.9; % Peso do amortecedor (forca normal nao linear)

g_force = 9.81;    % Forca da gravidade
sample_time = 0.003; % Sample time Thomas
sample_time_measurements = -1;

```

```
spring_body_x = 10^5; % Param. em X da mola do corpo
damper_body_x = 10^3; % Param. em X do amortecedor do corpo

spring_body_z = 10^4; % Param. em Z da mola do corpo
damper_body_z = 10^2; % Param. em Z do amortecedor do corpo

friction_spring = 1.3*10^6; % Param. de friccao da mola do solo
friction_damper = 3.4*10^3; % Param. de friccao do amortecedor do solo

normal_spring = 1.7*10^6; % Param. normal do amortecedor do solo
normal_damper = 2.7*10^6; % Param. normal da mola do solo

upper_friction_saturation = 2*10^6; % Saturacao positiva Friccao
lower_friction_saturation = -2*10^6; % Saturacao negativa Friccao

upper_normal_saturation = 4*10^6; % Saturacao positiva Normal
lower_normal_saturation = -2*10^6; % Saturacao negativa Normal

rho = 200; % Densidade da perna
rho_body = 100; % Densidade do corpo

h_legs = 0.2; % comprimento das pernas
r_legs = 0.02; % raio das pernas

h_upper_leg_1 = h_legs; % Comprimento perna 1 superior
r_upper_leg_1 = r_legs; % Raio perna 1 superior

h_lower_leg_1 = h_legs; % Comprimento perna 1 inferior
r_lower_leg_1 = r_legs; % Raio perna 1 inferior

h_upper_leg_2 = h_legs; % Comprimento perna 2 superior
r_upper_leg_2 = r_legs; % Raio perna 2 superior

h_lower_leg_2 = h_legs; % Comprimento perna 2 inferior
r_lower_leg_2 = r_legs; % Raio perna 2 inferior

h_upper_leg_3 = h_legs; % Comprimento perna 3 superior
r_upper_leg_3 = r_legs; % Raio perna 3 superior
```

```

h_lower_leg_3 = h_legs; % Comprimento perna 3 inferior
r_lower_leg_3 = r_legs; % Raio perna 3 inferior

h_upper_leg_4 = h_legs; % Comprimento perna 4 superior
r_upper_leg_4 = r_legs; % Raio perna 4 superior

h_lower_leg_4 = h_legs; % Comprimento perna 4 inferior
r_lower_leg_4 = r_legs; % Raio perna 4 inferior

h_ground = 0.01; %Altura solo Y
w_ground = 2; %Largura solo X
d_ground = 1; %Profundidade solo Z

h_body = 0.15; %Altura corpo Y
w_body = 0.3; %Largura corpo X
d_body = 0.2; %Profundidade corpo Z

% Massa perna 1:
m_upper_leg_1 = (pi*r_upper_leg_1^2*h_upper_leg_1*rho);
m_lower_leg_1 = (pi*r_lower_leg_1^2*h_lower_leg_1*rho);

% Massa perna 2:
m_upper_leg_2 = (pi*r_upper_leg_2^2*h_upper_leg_2*rho);
m_lower_leg_2 = (pi*r_lower_leg_2^2*h_lower_leg_2*rho);

% Massa perna 3:
m_upper_leg_3 = (pi*r_upper_leg_3^2*h_upper_leg_3*rho);
m_lower_leg_3 = (pi*r_lower_leg_3^2*h_lower_leg_3*rho);

% Massa perna 4:
m_upper_leg_4 = (pi*r_upper_leg_4^2*h_upper_leg_4*rho);
m_lower_leg_4 = (pi*r_lower_leg_4^2*h_lower_leg_4*rho);

m_ground = 400; % Massa do solo

m_body = rho_body*(h_body*d_body*w_body); % Massa do corpo

inertia_upper_leg_1 = ... % Inercia Sup. Perna 1
    [((1/2)*m_upper_leg_1*r_upper_leg_1^2) 0 0; ...
    0 (1/12)*(m_upper_leg_1*(3*r_upper_leg_1^2+h_upper_leg_1^2)) 0; ...

```

```

0 0 (1/12)*(m_upper_leg_1*(3*r_upper_leg_1^2+h_upper_leg_1^2));

inertia_lower_leg_1 = ... % Inercia Inf. Perna 1
  [((1/2)*m_lower_leg_1*r_lower_leg_1^2) 0 0; ...
  0 (1/12)*(m_lower_leg_1*(3*r_lower_leg_1^2+h_lower_leg_1^2)) 0; ...
  0 0 (1/12)*(m_lower_leg_1*(3*r_lower_leg_1^2+h_lower_leg_1^2))];

inertia_upper_leg_2 = ... % Inercia Sup. Perna 2
  [((1/2)*m_upper_leg_2*r_upper_leg_2^2) 0 0; ...
  0 (1/12)*(m_upper_leg_2*(3*r_upper_leg_2^2+h_upper_leg_2^2)) 0; ...
  0 0 (1/12)*(m_upper_leg_2*(3*r_upper_leg_2^2+h_upper_leg_2^2))];

inertia_lower_leg_2 = ... % Inercia Inf. Perna 2
  [((1/2)*m_lower_leg_2*r_lower_leg_2^2) 0 0; ...
  0 (1/12)*(m_lower_leg_2*(3*r_lower_leg_2^2+h_lower_leg_2^2)) 0; ...
  0 0 (1/12)*(m_lower_leg_2*(3*r_lower_leg_2^2+h_lower_leg_2^2))];

inertia_upper_leg_3 = ... % Inercia Sup. Perna 3
  [((1/2)*m_upper_leg_3*r_upper_leg_3^2) 0 0; ...
  0 (1/12)*(m_upper_leg_3*(3*r_upper_leg_3^2+h_upper_leg_3^2)) 0; ...
  0 0 (1/12)*(m_upper_leg_3*(3*r_upper_leg_3^2+h_upper_leg_3^2))];

inertia_lower_leg_3 = ... % Inercia Inf. Perna 3
  [((1/2)*m_lower_leg_3*r_lower_leg_3^2) 0 0; ...
  0 (1/12)*(m_lower_leg_3*(3*r_lower_leg_3^2+h_lower_leg_3^2)) 0; ...
  0 0 (1/12)*(m_lower_leg_3*(3*r_lower_leg_3^2+h_lower_leg_3^2))];

inertia_upper_leg_4 = ... % Inercia Sup. Perna 4
  [((1/2)*m_upper_leg_4*r_upper_leg_4^2) 0 0; ...
  0 (1/12)*(m_upper_leg_4*(3*r_upper_leg_4^2+h_upper_leg_4^2)) 0; ...
  0 0 (1/12)*(m_upper_leg_4*(3*r_upper_leg_4^2+h_upper_leg_4^2))];

inertia_lower_leg_4 = ... % Inercia Inf. Perna 4
  [((1/2)*m_lower_leg_4*r_lower_leg_4^2) 0 0; ...
  0 (1/12)*(m_lower_leg_4*(3*r_lower_leg_4^2+h_lower_leg_4^2)) 0; ...
  0 0 (1/12)*(m_lower_leg_4*(3*r_lower_leg_4^2+h_lower_leg_4^2))];

inertia_ground = ... % Inercia Solo
  [(1/12)*m_ground*(h_ground^2+d_ground^2) 0 0; ...
  0 (1/12)*m_ground*(w_ground^2+d_ground^2) 0; ...
  0 0 (1/12)*m_ground*(w_ground^2+h_ground^2)];

```

```

inertia_body = ... % Inercia Corpo
    (1/12)*[m_body*(h_body^2+d_body^2) 0 0; ...
    0 (1/12)*m_body*(w_body^2+d_body^2) 0; ...
    0 0 (1/12)*m_body*(w_body^2+h_body^2)];

%-----%
% Parametros dos controladores do modelo:

% Parametros das ancas
ganho_hip = 77;
zero_hip = -20.88;
polo_hip = -150;

% Parametros dos joelhos
ganho_knee = 204;
zero_knee = -20.88 ;
polo_knee = -150;

% FT do controlador das ancas
C_hip = zpk(zero_hip,polo_hip,ganho_hip);

% FT do controlador dos joelhos
C_knee = zpk(zero_knee,polo_knee,ganho_knee);

upper_torque_hip = 20; % saturacao superior ancas
lower_torque_hip = -20; % saturacao inferior ancas

upper_torque_knee = 20; % saturacao superior joelhos
lower_torque_knee = -20; % saturacao inferior joelhos

%-----%

% Padroes de Locomocao:

switch algor,

%-----%
% Plameamento de Trajectorias:
    case 1

```

```

FC = LS / pi; % altura do passo
L = h_upper_leg_1; % comprimento da perna.

switch gait,

    case 1

        beta = 0.75; % fracao de sustentacao

    case 2

        beta = 0.5; % fracao de sustentacao

end

% tempo de amostragem do sistema
sample_time = 0.005;

% tempo de amostragem das medidas
sample_time_measurements = -1;

final_time = nc*T;
t0 = 0:sample_time:T; % vetor de tempo de um periodo
t = 0:sample_time:final_time; % vetor de tempo total

TS = (beta)*T; % tempo de suporte
ts = 0:sample_time:TS;

TT = (1-beta)*T; % tempo de transferencia
tt = 0:sample_time:TT;

Hrb = 0.26; % altura do robo
VF = LS/T; % velocidade frontal
VL = VF/(1-beta); % velocidade da perna

% inicializacao dos vetores
theta1 = [];
theta2 = [];
X_leg = [];

```



```

Y_leg = [];
Z_leg = [];
X = [];
Y = [];

% fase de transferencia da perna
X_leg_s = VL*(tt - TT*sin((2*pi*tt)/TT)/(2*pi));
Y_leg_s = (FC/2)*(1 - cos(2*pi*tt/TT));
Z_leg_s(1:length(X_leg)) = 0;

% fase de suporte da perna
X_leg_p = repmat(X_leg_s(end),1,length(ts));
Y_leg_p = repmat(Y_leg_s(end),1,length(ts));
Z_leg_p = zeros(1,length(ts));

% concatenacao das fases da perna
X_leg = [X_leg_s X_leg_p];
Y_leg = [Y_leg_s Y_leg_p];
Z_leg = [Z_leg_s Z_leg_p];

% movimento do corpo
X_body = (VF)*t0 + LS*beta/2;
Y_body(1,1:length(t0)) = Hrb;
Z_body(1:length(X_body)) = 0;

% ajuste do referencial
for i = 1:1:length(t0)
    X = [X X_leg(i) - X_body(i)];
    Y = [Y Y_leg(i) - Y_body(i)];
end

% Cinematica Inversa
for i = 1:1:length(t0),

    Theta2 = - acos((X(1,i)^2 + Y(1,i)^2 - 2*L^2)/(2*L^2));

    Theta1 = atan2(Y(1,i),X(1,i)) - ...
        atan2(sin(Theta2),((1 + cos(Theta2)))) ;

```

```

        theta1 = [theta1 ((180/pi)*Theta1) ];
        theta2 = [theta2 ((180/pi)*Theta2) ];

    end

    % Replicacao de trajetoria para n ciclos

    theta1 = horzcat(theta1);
    theta2 = horzcat(theta2);

    auxtheta1 = theta1;
    auxtheta2 = theta2;

    X = horzcat(X_leg, X_body(1,2:end));
    Y = horzcat(Y_leg, Y_body(1,2:end));
    Z = horzcat(Z_leg, Z_body(1,2:end));

    auxX = X;
    auxY = Y;
    auxZ = Z;

    if nc>1
        for i = 1:1:nc-1 ,

            theta1 = horzcat(auxtheta1,theta1(1,2:end));
            theta2 = horzcat(auxtheta2,theta2(1,2:end));

            X = horzcat(auxX,X(1,2:end));
            Y = horzcat(auxY,Y(1,2:end));
            Z = horzcat(auxZ,Z(1,2:end));

        end
    end

    % defasamento dos angulos para os padroes de locomocao
    switch gait
    case 1
        % Caminhar
        a4 = 0/nc;
        a2 = 0.25/nc;

```

```

        a3 = 0.5/nc;
        a1 = 0.75/nc;

    case 2
        % Trotar
        a1 = 0/nc;
        a2 = 0.5/nc;
        a3 = 0.5/nc;
        a4 = 0/nc;

    end

F1 = round(a1*length(t));
F2 = round(a2*length(t));
F3 = round(a3*length(t));
F4 = round(a4*length(t));

shift = 1:1:length(t);

dt1 = circshift(shift,[0,F1]);
dt2 = circshift(shift,[0,F2]);
dt3 = circshift(shift,[0,F3]);
dt4 = circshift(shift,[0,F4]);

hip_1(:,2) = theta1(1,dt1); % angulos anca 1
hip_1(:,1) = t(1,:);      % Tempos anca 1

knee_1(:,2) = theta2(1,dt1); % angulos joelho 1
knee_1(:,1) = t(1,:);      % Tempos joelho 1

hip_2(:,2) = theta1(1,dt2); % angulos anca 2
hip_2(:,1) = t(1,:);      % Tempos anca 2

knee_2(:,2) = theta2(1,dt2); % angulos joelho 2
knee_2(:,1) = t(1,:);      % Tempos joelho 2

hip_3(:,2) = theta1(1,dt3); % angulos anca 3
hip_3(:,1) = t(1,:);      % Tempos anca 3

knee_3(:,2) = theta2(1,dt3); % angulos joelho 3

```

```
knee_3(:,1) = t(1,:);           % Tempos joelho 3
```

```
hip_4(:,2) = theta1(1,dt4);    % angulos anca 4
```

```
hip_4(:,1) = t(1,:);          % Tempos anca 4
```

```
knee_4(:,2) = theta2(1,dt4);  % angulos joelho 4
```

```
knee_4(:,1) = t(1,:);          % Tempos joelho 4
```

```
%-----%
% angulos pre-definidos em arquivos .txt:
```

```
case 2
```

```
load t.txt           % Carregar o vector tempo
```

```
load teta1.txt      % Carrega valores das juntas da anca
```

```
load teta2.txt      % Carrega valores das juntas do joelho
```

```
%Perna 1
```

```
hip_1(:,2) = teta1(1:6000,1); % angulos anca 1
```

```
hip_1(:,1) = t(1:6000);       % Tempos anca 1
```

```
hip_1(:,2) = hip_1(:,2)-10;
```

```
knee_1(:,2) = teta2(1:6000,1); % angulos joelho 1
```

```
knee_1(:,1) = t(1:6000);       % Tempos joelho 1
```

```
%Perna 2
```

```
hip_2(:,2) = teta1(1:6000,2); % angulos anca 2
```

```
hip_2(:,1) = t(1:6000);       % Tempos anca 2
```

```
hip_2(:,2) = hip_2(:,2)- 10;
```

```
knee_2(:,2) = teta2(1:6000,2); % angulos joelho 2
```

```
knee_2(:,1) = t(1:6000);       % Tempos joelho 2
```

```
%Perna 3
```

```
hip_3(:,2) = teta1(1:6000,3); % angulos anca 3
```

```
hip_3(:,1) = t(1:6000);       % Tempos anca 3
```

```
hip_3(:,2) = hip_3(:,2)- 10;
```

```
knee_3(:,2) = teta2(1:6000,3); % angulos joelho 3
```

```
knee_3(:,1) = t(1:6000);       % Tempos joelho 3
```

```

    %Perna 4
    hip_4(:,2) = teta1(1:6000,4); % angulos anca 4
    hip_4(:,1) = t(1:6000);      % Tempos anca 4
    hip_4(:,2) = hip_4(:,2)-10;

    knee_4(:,2) = teta2(1:6000,4); % angulos joelho 4
    knee_4(:,1) = t(1:6000);      % Tempos joelho 4

end

%-----%
%-----%
% Graficos:
    if algor == 1,

        figure(1); % Grafico dos angulos
        subplot(2,4,1);
        plot(hip_1(:,1),hip_1(:,2));
        title('Hip 1');
        subplot(2,4,5);
        plot(knee_1(:,1),knee_1(:,2));
        title('Knee 1');

        subplot(2,4,2);
        plot(hip_2(:,1),hip_2(:,2));
        title('Hip 2');
        subplot(2,4,6);
        plot(knee_2(:,1),knee_2(:,2));
        title('Knee 2');

        subplot(2,4,3);
        plot(hip_3(:,1),hip_3(:,2));
        title('Hip 3');
        subplot(2,4,7);
        plot(knee_3(:,1),knee_3(:,2));
        title('Knee 3');

        subplot(2,4,4);

```

```

plot(hip_4(:,1),hip_4(:,2));
title('Hip 4');
subplot(2,4,8);
plot(knee_4(:,1),knee_4(:,2));
title('Knee 4');

figure(2); % Grafico da trajetoria planeada
subplot(2,1,1);
plot(X_leg,Y_leg);
title('Leg Movement');
subplot(2,1,2);
plot(X_body,Y_body);
title('Body Movement');

figure(3); % Grafico animado da trajetoria planeada
for i = 1:length(X_body)
axis([-2*LS 2*LS, -0 Hrb+0.05]);
plot(X_body(i),Y_body(i),'b*');
pause(0.001)
hold on
plot(X_leg(i),Y_leg(i),'g*');
pause(0.001)
title('Leg Movement vs. Body Movement');
end

else

figure(2); % graficos do algoritmo de Castro

plot(hip_1(:,1),hip_1(:,2),'r');
hold on
plot(hip_2(:,1),hip_2(:,2),'b');
hold on
plot(hip_3(:,1),hip_3(:,2),'g');
hold on
plot(hip_4(:,1),hip_4(:,2),'d');

end

```

*% fim do código*

*%-----%*