

Simulação de Robôs Quadrúpedes Utilizando o SimMechanics

Tomás Figueira Vieira Souto e Castro



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Outubro 2012

Simulação de Robôs Quadrúpedes Utilizando o SimMechanics

Tomás Figueira Vieira Souto e Castro

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Automação e Sistemas

Orientador: Manuel Fernando dos Santos Silva

Co-orientador: Ramiro de Sousa Barbosa

Aprovado em provas públicas pelo júri:

Presidente: José António Tenreiro Machado

Vogal Externo: António Manuel Ferreira Mendes Lopes

Orientador: Manuel Fernando dos Santos Silva

Outubro 2012

Agradecimentos

Agradeço a todas as pessoas que de alguma forma tornaram este projecto possível. Gostaria de expressar os meus sinceros agradecimentos a toda a comunidade que trabalha com Simmechanics™ e que indirectamente me ajudou a resolver inúmeros problemas que foram surgindo durante este trabalho.

Gostaria de agradecer especialmente aos meus orientadores, Professor Manuel Fernando dos Santos Silva e Professor Ramiro de Sousa Barbosa pelas suas contribuições directas neste trabalho. Muito obrigado pela motivação que me deram e pela paciência que demonstraram ter no desenrolar deste projecto.

Por último agradeço à minha família e amigos pelo incondicional apoio prestado.

A todos Muito Obrigado!

Tomás Castro

Resumo

Desde sempre o Homem procurou automatizar tarefas repetitivas. Tanto na indústria como no sector doméstico ou de comércio estão constantemente a surgir novos equipamentos procurando aumentar o nível de automação de diversas tarefas.

A necessidade de empresas criadoras de produtos inovadores se manterem concorrenciais obrigou à adopção de métodos de projecto e planeamento mais estruturados e eficientes que eliminem os desperdícios tanto de material como de tempo. Diversas restrições em termos orçamentais e o aumento da procura de produtos de alta qualidade a baixo custo deu origem a um novo tipo de programas, os programas de simulação virtual.

Tal como na indústria, também na investigação nas Instituições de Ensino Superior se procuram adoptar os métodos de trabalho mais eficientes possíveis. Uma parte dos programas de simulação robótica utilizados hoje em dia na investigação no Instituto Superior de Engenharia do Porto estão concebidos em linguagem C. Embora esta linguagem seja extremamente capaz, existem métodos mais adequados de simulação que aceleram o processo de modelação permitindo a visualização do movimento e simulação de diversas situações potencialmente críticas sem existir a necessidade de construção de um protótipo funcional do robô.

Durante o último ano, este projecto permitiu construir um modelo de um robô quadrúpede num programa de simulação virtual chamado Simmechanics™.

Embora haja já algum trabalho desenvolvido nesta área, constatou-se que a informação existente possuía algumas lacunas. Com o trabalho desenvolvido preparou-se esta dissertação que pretende clarificar algumas questões que surgem na modelação de um robô.

Neste documento explica-se como modelar um robô de corpo flexível, modelar as pernas e as respectivas juntas da anca e do joelho assim como também se apresenta a modelação de um contacto com o solo recorrendo a um modelo descrito na literatura.

Este documento foi escrito a pensar no leitor, procurando-se explicar de forma simples mas concisa os diversos passos necessários para construir um modelo virtual de um robô. O autor espera, desta forma, ajudar futuros colegas que pretendam trabalhar com o Simmechanics™ a mais facilmente se integrarem com o programa aumentando desta forma o declive da curva de aprendizagem e permitindo um maior aproveitamento desta ferramenta.

Palavras-chave

Locomoção, Matlab, Quadrúpede, Robótica, Simmechanics, Simulink.

Abstract

Man has always sought to automate repetitive tasks. In industry, homes or businesses are constantly emerging new equipments seeking to increase the level of automation of various tasks.

The need for companies creating innovative products to remain competitive is forcing them to adopt new methods of project planning, more structured and efficient to reduce material waste and time. Several constraints in terms of budget and the increasing demand for high quality products at low cost has led to a new kind of software for virtual simulation.

As in industry, also research laboratories in high schools are looking to adopt the most efficient ways to working. Part of robotics simulation software used today in research at Instituto Superior de Engenharia do Porto are designed in C language. Although this language is extremely capable, there are better methods that accelerate the modeling process enabling visualization of the movement and simulation of several potentially critical situations without the need to build a functional prototype of the robot.

During the past year this project allowed for the creation of a model of a quadruped robot in a virtual simulation program called SimmechanicsTM.

Although there is already some work in this area, it was found that existing information had some gaps. With the knowledge gained from this project this document was elaborated with the intent to clarify some doubts that arise during the creation of a SimmechanicsTM model of a robot.

This document explains how to model a flexible robot body, its legs and their joints of the hip and knee. The model of a feet-ground interaction was also modelled using a theoretic model described in the literature.

This document was written with the reader in mind, trying to explain in a simple but concise way the various steps needed to build a virtual model of a robot. The author hopes to help future colleagues who wish to work with SimmechanicsTM to more easily integrate with the software thereby increasing the slope of the learning curve and allowing a better use of this tool.

Keywords

Locomotion, Matlab, Quadruped, Robótica, Simmechanics, Simulink.

*“The great tragedy of science -
the slaying of a beautiful hypothesis
by an ugly fact.”*

Thomas Huxley

Conteúdo

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	ix
Lista de Figuras	xi
Lista de Tabelas	xv
Acrónimos	xvii
1 Introdução	1
1.1 Motivação	2
1.2 Objectivos	3
1.3 Porquê um robô quadrúpede?	3
1.4 Porquê Simmechanics™?	6
1.5 Calendarização	7
1.6 Estrutura da dissertação	8
2 Sistemas Robóticos	9
2.1 Breve resenha histórica	9
2.2 Definição de robô	10
2.3 Robôs com locomoção por pernas	13
2.4 Actualidade da robótica	15
3 Modelo do Robô Quadrúpede	19
3.1 Cinemática e dinâmica de robôs com pernas	19
3.2 Modelo cinemático	20
3.3 Modelo dinâmico	22
3.4 Modelo de contacto pé-solo	23
4 Simmechanics	27
4.1 Blocos básicos do Simmechanics	27
4.1.1 Corpo (<i>Body</i>)	28
4.1.2 Junta Rotacional (<i>Revolute Joint</i>)	31
4.1.3 Actuador de Junta (<i>Joint Actuator</i>)	33
4.1.4 Actuador de Corpo (<i>Body Actuator</i>)	34

CONTEÚDO

4.1.5	Bloco de Referência de Solo (<i>Ground</i>)	37
4.1.6	Bloco de parametrização do ambiente de simulação (<i>Machine Environment</i>)	37
4.1.7	Sensor de Corpo (<i>Body Sensor</i>)	40
4.1.8	Sensor de Junta (<i>Joint Sensor</i>)	40
4.2	Exemplos de sistemas mecânicos modelados em Simmechanics™	43
4.2.1	Pêndulo Simples	43
4.2.2	Bouncing ball (Bola saltitante)	43
4.2.3	Stewart platform	48
5	Trabalho desenvolvido	53
5.1	Modelação do robô quadrúpede	53
5.1.1	Modelação do corpo do robô	53
5.1.2	Modelação das pernas do robô	56
5.2	Sistema de controlo das juntas das ancas e dos joelhos	61
5.3	Modelo de contacto pé-solo	62
5.3.1	Força tangencial (F_x)	63
5.3.2	Força normal (F_y)	66
5.4	Looptune - Sintonização de controladores	67
5.5	Montagem final do robô	71
5.6	Problemas encontrados, soluções adoptadas e resultados obtidos	73
5.6.1	Teste de gerador central de padrões (CPG)	73
5.6.2	Padrão de locomoção gerado por aplicação externa	74
5.6.3	Problema de interação pé-solo	74
5.6.4	Problema de velocidade de simulação	76
5.6.5	Escolha do método de simulação (<i>solver</i>)	76
5.6.6	Optimização da velocidade de simulação	77
5.6.7	Análise do contacto pé-solo	79
5.6.8	Análise da complacência do corpo	79
5.6.9	Validação do caminhar do robô	79
6	Conclusões	91
6.1	Satisfação dos objectivos	91
6.2	Trabalho futuro	92
	Referências	95
A	Parametrização do robô	97

Lista de Figuras

Fig. 1.1	Robô Monópode - Uniroo.	4
Fig. 1.2	Robô Bípede - Honda Asimo.	4
Fig. 1.3	Robôs Quadrúpedes.	5
Fig. 2.1	Robô bípede projectado por Leonardo Da Vinci.	10
Fig. 2.2	Mecanismo de Chebyshev.	10
Fig. 2.3	Tipos de locomoção por rodas e por lagartas.	11
Fig. 2.4	Tipos de locomoção por pernas e imitando uma serpente por movimentos em zigue-zague.	12
Fig. 2.5	Representação de seis graus de liberdade, 3 prismáticos e 3 rotacionais.	12
Fig. 2.6	Robô que joga Ping-Pong - TOPIO.	14
Fig. 2.7	Imagem do que poderá vir a ser o robô Cheetah.	15
Fig. 2.8	Robô ASIMO no presente depois de 10 anos de trabalho.	16
Fig. 2.9	Robô TITAN.	17
Fig. 2.10	Dois exemplares do robô AIBO da Sony.	17
Fig. 3.1	Esquema das relações entre cinemática directa e inversa.	20
Fig. 3.2	Numeração de pernas robô.	21
Fig. 3.3	Disposição dos ângulos das juntas.	22
Fig. 3.4	Dinâmica directa e inversa.	23
Fig. 3.5	Esquema do corpo do robô apresentado neste trabalho.	24
Fig. 3.6	Força de contacto com solo.	24
Fig. 3.7	Modelo de contacto com o solo.	25
Fig. 4.1	Esquema da constituição da perna do robô.	28
Fig. 4.2	Bloco Corpo (<i>Body</i>).	28
Fig. 4.3	Configurações do bloco "Body", janela "Position".	29
Fig. 4.4	Exemplo de utilização dos sistemas de coordenadas.	30
Fig. 4.5	Janela de definição da orientação dos sistemas de coordenadas.	31
Fig. 4.6	Janela de definição das opções de visualização do corpo.	32
Fig. 4.7	Junta rotacional.	32
Fig. 4.8	Janela de definição das características da Junta Rotacional.	33
Fig. 4.9	Actuador de Junta.	34
Fig. 4.10	Janela inicial do actuador de junta.	34
Fig. 4.11	Parametrização do Actuador de Juntas.	35
Fig. 4.12	Actuador de Corpo.	36
Fig. 4.13	Parametrização do Actuador de Corpo.	36
Fig. 4.14	Bloco Solo (<i>Ground</i>).	37
Fig. 4.15	Parâmetros de configuração do bloco " <i>Ground</i> " (<i>Ground</i>).	37

LISTA DE FIGURAS

Fig. 4.16 Bloco <i>Machine Environment</i>	38
Fig. 4.17 Parâmetros <i>Machine Environment</i>	38
Fig. 4.18 Parâmetros da janela <i>Constraints</i>	39
Fig. 4.19 Blocos <i>Constraints & Drivers</i>	39
Fig. 4.20 Parâmetros da janela <i>Linearization</i>	40
Fig. 4.21 Parâmetros da janela <i>Visualization</i>	41
Fig. 4.22 Bloco <i>Body Sensor</i>	41
Fig. 4.23 Parâmetros do bloco <i>Body Sensor</i>	42
Fig. 4.24 Blocos de divisão de sinais de Simulink™	42
Fig. 4.25 Bloco <i>Joint Sensor</i>	42
Fig. 4.26 Parâmetros do bloco <i>Joint Sensor</i>	43
Fig. 4.27 Imagens do modelo do pêndulo simples.	44
Fig. 4.28 Gráficos de análise do movimento do pêndulo.	45
Fig. 4.29 Modelo da "Bola Saltitante".	46
Fig. 4.30 Gráficos de análise do movimento da bola saltitante.	47
Fig. 4.31 Modelo da plataforma de Stewart.	48
Fig. 4.32 Representação gráfica da plataforma.	49
Fig. 4.33 Controlador utilizado na plataforma de Stewart.	49
Fig. 4.34 Gráfico obtido pela simulação do modelo da plataforma de Stewart.	50
Fig. 5.1 Esquema do corpo do robô.	54
Fig. 5.2 Sistema mola-amortecedor do corpo.	55
Fig. 5.3 Parametrização do bloco 1 do corpo.	55
Fig. 5.4 Referenciação dos blocos do corpo.	57
Fig. 5.5 Modelo de uma perna do robô.	58
Fig. 5.6 Bloco junta rotacional da anca do robô.	59
Fig. 5.7 Configuração da perna superior.	60
Fig. 5.8 Configuração da perna inferior.	61
Fig. 5.9 Bloco do controlador.	61
Fig. 5.10 Controlador PD.	62
Fig. 5.11 Bloco de verificação de contacto com o solo.	62
Fig. 5.12 Detalhes da configuração do bloco de verificação do contacto com o solo.	63
Fig. 5.13 Esquema de blocos de cálculo da força de contacto tangencial.	64
Fig. 5.14 Bloco de cálculo da força de contacto tangencial.	65
Fig. 5.15 Bloco " <i>Fcn</i> " de cálculo da força tangencial.	65
Fig. 5.16 Esquema de blocos de cálculo da força de contacto normal.	66
Fig. 5.17 Bloco de cálculo da força de contacto normal.	66
Fig. 5.18 Bloco " <i>Fcn</i> " de cálculo da força normal.	67
Fig. 5.19 Esquema do sistema de controlo.	69
Fig. 5.20 Simulação do robô utilizando ganhos obtidos através da ferramenta looptune.	70
Fig. 5.21 Simulação do robô.	71
Fig. 5.22 Esquema do modelo do robô.	72
Fig. 5.23 Esquema de organização do modelo do robô.	73
Fig. 5.24 Subsistema 2.	74
Fig. 5.25 Robô com os pés a trespassar o solo.	75

LISTA DE FIGURAS

Fig. 5.26 Janela de desactivação de <i>Zero Crossing Detection</i>	77
Fig. 5.27 <i>Simulink Performance Advisor</i>	78
Fig. 5.28 Contacto pé-solo.	80
Fig. 5.29 Complacência do corpo em x e z	81
Fig. 5.30 Robô a caminhar com as pernas 2 e 3 na fase de transferência.	82
Fig. 5.31 Robô a caminhar com as pernas 1 e 4 na fase de transferência.	83
Fig. 5.32 Evolução da posição do robô no espaço.	84
Fig. 5.33 Estado do robô instantes antes da instabilidade.	85
Fig. 5.34 Estado do robô no momento em que o sistema se torna instável.	86
Fig. 5.35 Sistema instável com o robô a "voar".	87
Fig. 5.36 Forças de contacto da perna 3 no instante em que o sistema se torna instável.	88

LISTA DE FIGURAS

Lista de Tabelas

Tabela 1.1	Calendarização da Tese	7
Tabela 3.1	Variáveis de posição das juntas da anca e do joelho.	22
Tabela 5.1	Variáveis de configuração das pernas.	57
Tabela A.1	<i>Miscellaneous parameters</i>	97
Tabela A.2	<i>Feet/ground contact parameters</i>	98
Tabela A.3	<i>Body parameters</i>	99
Tabela A.4	<i>PD controller settings</i>	99
Tabela A.5	<i>Leg Inertia Tensors.</i>	100
Tabela A.6	<i>Loading locomotion pattern vectors.</i>	100

LISTA DE TABELAS

Acrónimos

CAD	<i>Computer-Aided Design</i>
CG	Centro de Gravidade
CPG	<i>Central Pattern Generator</i> (Gerador Central de Padrões)
CS	<i>Coordinate System</i>
GDL	Graus de Liberdade
ISEP	Instituto Superior de Engenharia do Porto
JIC	<i>Joint Initial Conditions</i>
PD	Proporcional e Derivativo
PID	Proporcional, Integral e Derivativo
SC	Sistema de Coordenadas
TOPIO	<i>TOSY Ping Pong Playing Robot</i>

Acrónimos

Capítulo 1

Introdução

O aumento da investigação na área da robótica tem sido muito notável. Os custos e o tempo investido no desenvolvimento de protótipos de equipamentos são muito elevados o que incentiva à pesquisa de novos métodos de trabalho que permitam o desenvolvimento e teste de máquinas de forma mais rápida, económica e eficaz.

A facilidade que hoje em dia se verifica em adquirir um computador com capacidades razoáveis de processamento a baixo custo e o actual nível de desenvolvimento de ferramentas de simulação em ambiente a três dimensões abrem portas para um novo método de trabalho.

Existem no mercado inúmeras ferramentas que permitem projectar, simular e testar sistemas mecânicos. Estes programas facilitam a fase de projecto dos sistemas mecânicos permitindo uma visualização em tempo real do equipamento que se pretende construir ajudando, desta forma, na detecção de eventuais problemas.

Para além de toda a facilidade que introduzem na fase de projecto, estes programas também auxiliam na fase de testes do equipamento sendo na maioria dos casos possível testá-los em ambiente de três dimensões virtual.

Embora a sua utilização não elimine por completo a necessidade de construção de um protótipo funcional do equipamento, a sua utilização diminui a probabilidade de erro devido aos inúmeros testes que permitem fazer em ambiente virtual. A sua correcta utilização tem os seguintes benefícios:

- Redução de desperdícios de matéria prima;
- Diminuição do número de horas necessário para a construção de protótipos;
- Diminuição do tempo de projecto;
- Visualização em tempo real do sistema enquanto é modelado;
- Simulação do sistema de controlo em ambiente virtual;

- Teste do equipamento em diversas condições sem necessidade de deslocação aos locais onde essas condições possam existir.

Os programas de simulação de sistemas dinâmicos são utilizados em diversos tipos de indústria, tendo particular interesse na robótica pelas possibilidades que apresentam de modelação e simulação de diferentes tipos de sistemas com diferentes graus de complexidade.

A qualidade e rapidez de simulação, assim como a capacidade de cálculo, são alguns dos parâmetros a ter em linha de conta quando se escolhe o simulador a utilizar para uma determinada função. O programa Matlab[®], faz parte do *curriculum* de diversas disciplinas leccionadas nos cursos de Licenciatura e Mestrado em Engenharia Electrotécnica e de Computadores do Instituto Superior de Engenharia do Porto. Este programa possui uma ferramenta de simulação de sistemas dinâmicos chamada Simulink[™]. Tratando-se de um programa com um enorme potencial, considerou-se oportuna a execução do presente trabalho utilizando a ferramenta Simmechanics[™], parte integrante do Simulink[™], através da criação de um modelo que permita a simulação de um robô quadrúpede.

Neste primeiro capítulo faz-se a introdução do tema que será abordado neste relatório de dissertação. Contém seis secções, na primeira explica-se a importância deste trabalho e as motivações para a sua realização, em seguida, na segunda secção são descritos os objetivos a que o autor se propõe neste trabalho, na terceira e quarta secção é, resumidamente explicado porque se optou pela utilização do Simmechanics[™] e o porquê da modelação de um robô quadrúpede, na secção 5 apresenta-se uma tabela com a calendarização do trabalho e, por último, na secção 6 é descrita a estrutura desta dissertação.

1.1 Motivação

A simulação dinâmica de equipamentos é, nos dias de hoje, uma ferramenta essencial ao desenvolvimento de um projecto.

Seja por questões financeiras ou puramente estratégicas, actualmente existe um consenso no que diz respeito à simulação computacional como forma de projecto e teste de sistemas mecânicos.

Um projecto que no passado, por falta de meios, obrigava à construção de diversos protótipos até chegar a um protótipo funcional pode, hoje em dia, ser feito sem sequer pegar numa ferramenta. Existe no mercado software que permite o desenho e construção virtual dos equipamentos e a execução de testes também em ambiente virtual. A rapidez com que diversas alterações podem ser implementadas e testadas, e a poupança de matéria prima que daí advém, são só duas das vantagens que estes novos métodos de projecto apresentam.

O potencial da ferramenta Simmechanics™ e a falta de informação a seu respeito foram também fontes de motivação para a execução deste trabalho. Os poucos trabalhos que existem relatados na literatura estão bastante incompletos, nunca demonstrando claramente como os modelos são implementados.

Procurou-se criar um projecto que permitisse a exploração do programa com um grau de profundidade suficiente que facilitasse no futuro o estudo de outros tipos de robôs quadrúpedes com pequenas, ou nenhuma, alterações ao modelo desenvolvido neste trabalho.

1.2 Objectivos

Este trabalho teve como principal objectivo a criação de um modelo virtual de simulação de robôs quadrúpedes utilizando a plataforma Simulink™ do Matlab®.

Como objectivos parciais foram definidos os seguintes:

1. estudo de projectos já existentes em Simmechanics™;
2. estudo dos conceitos básicos de cinemática e dinâmica de um robô quadrúpede;
3. criação de um modelo digital do robô quadrúpede;
4. implementação de flexibilidade no corpo do robô;
5. introdução de um padrão de locomoção no robô;
6. criação, afinação e teste do sistema de contacto com o solo;
7. afinação de controladores das juntas para que se verifique a locomoção do robô.

Pretende-se que pela leitura desta dissertação qualquer pessoa consiga implementar um robô com as mesmas características do projecto aqui implementado.

O modelo desenvolvido aceitará diferentes padrões de locomoção pela introdução directa das posições angulares nas juntas do joelho e anca. Também se pretende que seja relativamente fácil a alteração de algumas características físicas do robô, tais como tamanho do corpo e das pernas do modelo, posição inicial do robô, densidades de materiais constituintes, flexibilidade do corpo e parâmetros do solo.

1.3 Porquê um robô quadrúpede?

O acesso a diversos locais na Terra é facilitado através da utilização de pernas.

O potencial deste tipo de locomoção tem dado origem a diversos projectos de investigação nos mais diversos sectores, desde o sector residencial com o objectivo de desenvolver robôs autónomos para desempenharem as actividades rotineiras de uma casa até robôs industriais ou de entretenimento que poderão ser utilizados nas mais diversas circunstâncias como, por exemplo, a manipulação de produtos em fábricas ou utilizados

Introdução

para animar eventos sociais. No entanto, a complexidade da dinâmica e do controle deste tipo de locomoção tem criado diversos impasses ao seu desenvolvimento. A evolução da

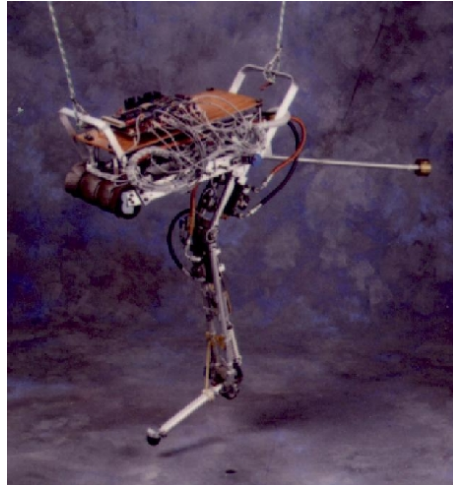


Figura 1.1: Robô Monópode - Uniroo.

ciência na área da robótica tem sido no sentido de criar robôs com diferente número de pernas, capazes de ultrapassar obstáculos de outra forma não ultrapassáveis (ver Figuras 1.1 e 1.2).

O que para os seres humanos são tarefas simples como correr e saltar, representam num robô uma infinidade de cálculos com vista a obter a melhor solução que permita ao robô deslocar-se para a posição pretendida. Dependendo da missão a desempenhar, a sua



Figura 1.2: Robô Bípede - Honda Asimo.

Introdução

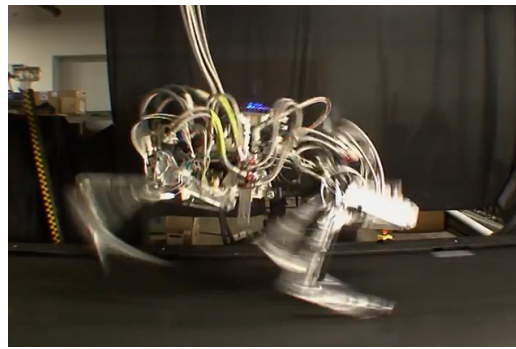
configuração pode ser muito variável, existindo robôs monópodes como o Uniuroo [Lab12] (Figura 1.1), bípedes como o Honda Asimo [Hon12] (Figura 1.2), quadrúpedes como o BigDog [Dyn12a] (Figura 1.3a), hexápodes como o RHex [DAR12], entre outros.

Nos últimos anos têm sido desenvolvidos diversos projectos de robôs adoptando a fisiologia mecânica de movimento de seres vivos. Os seus padrões de locomoção procuram "imitar" os padrões de locomoção de animais como aranhas, cães ou cavalos. O caminhar, que para o Ser Humano não apresenta grande desafio, é bastante complicado de replicar utilizando um robô. Os diversos estudos nesta área têm produzido muito conhecimento, sendo já possível observar alguns robôs a replicar o caminhar de um Humano, como o TITAN [Rob11], ou um cão, como o BigDog [MRP08] (Figura 1.3a).

O processo de modelação computacional da cinemática e dinâmica de robôs é bastante complexo devido à grande complexidade dos sistemas e à não linearidade do sistema introduzida pelas inúmeras variáveis que influenciam directamente a dinâmica do robô.



(a) BigDog



(b) Cheetah

Figura 1.3: Robôs Quadrúpedes.

Depois desta breve introdução e voltando à questão que dá início a esta secção, a opção pelo desenvolvimento de um modelo virtual de simulação de um robô quadrupede prende-se essencialmente com 3 factores:

- Actualidade;
- Complexidade;
- Disponibilidade de informação.

Os robôs quadrúpedes apresentam diversas vantagens em relação a robôs semelhantes com diferente número de pernas uma vez que existe uma relação directa entre estabilidade/complexidade e número de pernas de um robô. Dependendo do padrão de locomoção escolhido, do número de pernas e da sua colocação em relação ao corpo do robô, é possível aumentar a estabilidade dinâmica do robô garantindo que o centro de gravidade (CG) do robô se encontra sempre dentro de um triângulo formado pelas pernas que estão em contacto com o solo. No entanto, aumentando o número de pernas, também aumenta a

complexidade do sistema obrigando à implementação de um modelo de locomoção consistente com o número de pernas do robô. É portanto importante ponderar bem a relação estabilidade/complexidade que, satisfazendo os requisitos de mobilidade, não torne o seu desenvolvimento demasiado complexo para o tipo de trabalho pretendido.

Como neste trabalho se pretende desenvolver os conhecimentos da ferramenta de simulação, considerou-se que um robô com 4 pernas seria o indicado, introduzindo os desafios necessários no que diz respeito à mobilidade e coordenação do movimento das pernas e não tornando o modelo demasiado pesado, impossibilitando a sua simulação. Por último verificou-se a disponibilidade de informação, constatando-se que existe muita informação no que diz respeito ao projecto de robôs quadrúpedes.

1.4 Porquê Simmechanics™?

Existem no mercado inúmeras aplicações para simulação tridimensional de sistemas mecânicos como:

- Adams - Multibody Dynamics Simulation [Sof];
- Webots [Cyb];
- VisSim [Inc];
- Simpack [AG].

Qualquer um dos programas atrás mencionados permite fazer a análise de sistemas dinâmicos e sua simulação. Uns com uma interface gráfica mais atraente, outros recorrendo somente a programação, todos possuem capacidades semelhantes às do Simmechanics™.

A vantagem de utilização do Simmechanics™ prende-se com o seu interface bastante intuitivo, com a possibilidade de importação de detalhes do modelo directamente de desenhos CAD e da facilidade de importação/exportação de variáveis directamente do ambiente Matlab®. Trata-se de uma ferramenta bastante completa, de relativamente fácil funcionamento e, devido à possibilidade de interação com as restantes ferramentas do Simulink™, de fácil expansão. O Matlab® possui muitas funcionalidades não só ao nível de simulação (Simulink™), mas também como programa de cálculo numérico.

Esta ferramenta, embora exista desde 2001, é ainda pouco utilizada tanto no meio académico como no meio industrial. Existe pouca documentação de apoio e a sua comunidade é ainda pouco extensa, tornando este trabalho num desafio ainda mais interessante.

Espera-se que este trabalho sirva como guia de orientação para colegas que, no futuro, escolham esta plataforma para simulação dinâmica em ambiente de três dimensões.

O objectivo final deste trabalho é, no domínio da investigação, ganhar conhecimentos de uma ferramenta até agora pouco utilizada percebendo o seu potencial e a sua adaptabilidade às necessidades da simulação robótica.

1.5 Calendarização

Na Tabela 1.1 apresenta-se a calendarização da tese apresentada nesta dissertação. Este trabalho começou a ser executado em Outubro de 2011. Durante os primeiros 5 meses de trabalho foi recolhida informação sobre robôs com pernas, problemas na sua modelação e estudado o programa SimmechanicsTM. Em seguida começou-se o desenvolvimento do robô e a sua modelação, estas tarefas começaram em Março e foram desenvolvidas em paralelo com a escrita da dissertação. Por último tratou-se da sintonização dos controladores das juntas, tendo esta tarefa demorado mais tempo devido à tentativa de optimização automática dos controladores.

Tabela 1.1: Calendarização da Tese

	2011			2012										
	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maió	Junho	Julho	Agosto	Setembro	Outubro	Novembro
Estudo do estado da arte de robôs quadrúpedes	✓													
Identificação dos principais problemas de simulação		✓												
Definição concreta dos objectivos propostos		✓	✓											
Estudo das características cinemáticas e dinâmicas dos robôs quadrúpedes				✓	✓									
Introdução à ferramenta Simmechanics, execução de alguns testes			✓	✓	✓									
Estudo de tensores de Inércia					✓									
Elaboração do modelo de locomoção						✓								
Modelação da primeira perna e teste de movimento					✓	✓								
Introdução de controladores PID nas juntas da anca e joelho						✓	✓							
Estudo da ferramenta Looptune e sua implementação na perna								✓						
Construção do corpo flexível								✓						
Modelação do robô quadrúpede								✓	✓	✓	✓			
Sintonização de controladores											✓	✓	✓	✓
Elaboração da dissertação					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

1.6 Estrutura da dissertação

Para além da introdução, este relatório contém mais 5 capítulos. No Capítulo 2, é feita uma introdução ao estado da arte e são apresentados alguns robôs com pernas já existentes no mercado. No Capítulo 3 introduz-se o leitor a algumas questões relacionadas com a cinemática e dinâmica de robôs com pernas. No Capítulo 4 é descrito o funcionamento do programa de modelação SimmechanicsTM e são apresentados alguns exemplos de modelos já construídos utilizando esta ferramenta. No Capítulo 5, é explicado com o devido pormenor como foi modelado e testado todo o robô objecto desta dissertação. Por fim, no Capítulo 6 são apresentadas as conclusões do trabalho e ideias para futuros desenvolvimentos.

Capítulo 2

Sistemas Robóticos

Os robôs, com a sofisticação actual, são o resultado de muitos anos de experiências. Umhas bem sucedidas, outras não tão bem sucedidas, mas todas contribuíram com algo para o equipamento que actualmente se chama robô.

2.1 Breve resenha histórica

A existência de um equipamento que auxiliasse nos diversos trabalhos do dia-a-dia é, desde há muitos anos, uma ambição do Ser Humano. Já em 320 A.C. Aristóteles escreveu:

"If every tool, when ordered, or even of its own accord, could do the work that befits it... then there would be no need either of apprentices for the master workers or of slaves for the lords."

Muito mais tarde, em 1495 Leonardo Da Vinci idealizou o primeiro robô à semelhança de um cavaleiro que pela mecânica no seu interior deveria imitar uma pessoa a caminhar (Figura 2.1).

Posteriormente, em 1850 o matemático russo Chebyshev idealizou e criou um sistema de locomoção que permitia a um equipamento dotado de quatro pernas movimentar-se movendo duas pernas de cada vez de forma alternada, garantindo que mantinha o equilíbrio (Figura 2.2).

Entre o mecanismo de Chebyshev, desenvolvida no século XIX, e meados do século XX diversas máquinas foram idealizadas e concebidas. Verificou-se, no entanto, que com mais ou menos sofisticação as máquinas continuavam bastante limitadas em termos de utilidade devido ao facto de serem pouco ou nada adaptativas. Qualquer alteração nas condições previstas de funcionamento levariam a máquina a falhar.



Figura 2.1: Robô bípede projectado por Leonardo Da Vinci.



Figura 2.2: Mecanismo de Chebyshev.

A partir da década de cinquenta do século XX o interesse pela locomoção com recurso a pernas aumentou e começaram a aparecer laboratórios a desenvolver máquinas com diferentes níveis de sofisticação.

2.2 Definição de robô

Existem diversos tipos de robôs com diferentes finalidades, o que torna complicado definir um método de classificação que funcione em todos os casos.

Dependendo das funções que se pretende que o robô desempenhe este pode ser considerado um robô manipulador ou móvel. As funções de cada um são muito distintas sendo que o robô manipulador está usualmente estacionário e as suas funções consistem na movimentação de cargas ou, pelo uso de ferramentas apropriadas, execução de tarefas, por exemplo numa linha de montagem. Os robôs móveis são mais utilizados para transporte de cargas, inspeção de equipamentos de difícil acesso ou em ambientes hostis e segurança de perímetros.

No presente trabalho descreve-se a modelação de um robô móvel, importando por isso aprofundar um pouco as diversas opções dentro deste ramo. Um robô móvel é definido pelos parâmetros a seguir referidos.

Forma de locomoção

Dependendo do local de trabalho do robô deve ser escolhida a forma de locomoção mais apropriada.



(a) Rodas



(b) Lagartas

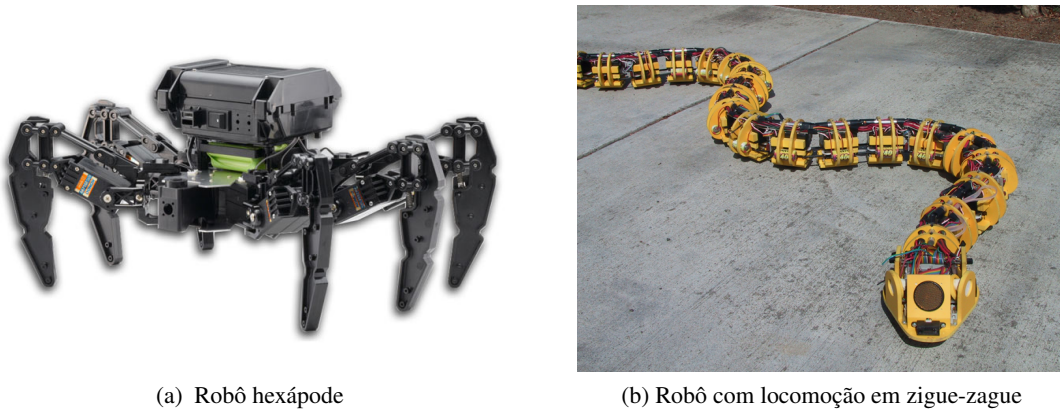
Figura 2.3: Tipos de locomoção por rodas e por lagartas.

Esta pode ser por meio de rodas (Figura 2.3a), lagartas (Figura 2.3b), pernas (Figura 2.4a) ou recorrendo às características do próprio corpo (Figura 2.4b), como seja a locomoção típica das serpentes em zigue-zague ou movimento de concertina.

Grau de liberdade

O grau de liberdade (gdl) de um determinado sistema mecânico indica de que forma é que o sistema se pode mover no espaço físico.

O número máximo de graus de liberdade de uma junta é seis, três graus de liberdade rotacionais e três graus de liberdade prismáticos (Figura 2.5). Graus de liberdade de um mecanismo são o número de movimentos independentes que esse mecanismo pode realizar no espaço tridimensional. Dependendo da aplicação a que se destina o robô,



(a) Robô hexápode

(b) Robô com locomoção em zigue-zague

Figura 2.4: Tipos de locomoção por pernas e imitando uma serpente por movimentos em zigue-zague.

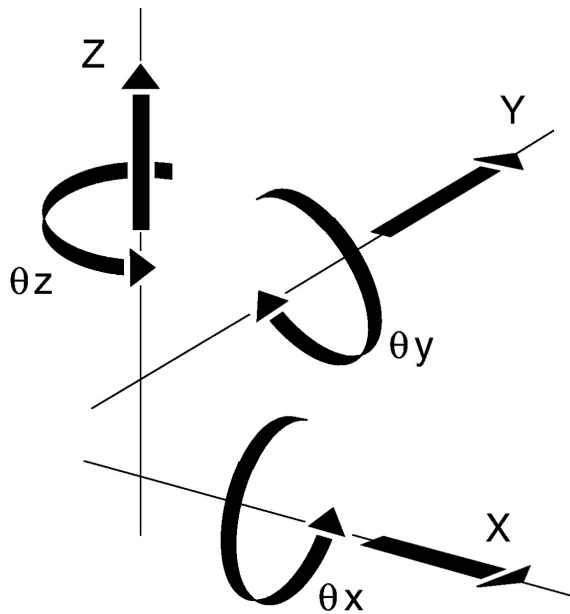


Figura 2.5: Representação de seis graus de liberdade, 3 prismáticos e 3 rotacionais.

devem ser definidos os gdl das suas juntas que irão definir quantos graus de liberdade o robô terá no espaço. Quanto mais graus de liberdade o robô tiver actuados mais complexo e dispendioso se torna.

O robô apresentado nesta dissertação possui dois graus de liberdade por perna. Embora possua liberdade para se movimentar com 6 gdl, por acção das juntas actuadas pelo sistema de controlo só lhe é possível deslocar-se em 4: 2 graus de liberdade prismáticos ao longo do eixo do x e do y e 2 graus de liberdade rotacionais em torno de x e z .

Padrão de locomoção

Este parâmetro só se aplica aos robôs com locomoção por pernas e define "a sequência

de movimentos dos membros e do corpo de um robô que ocorrem durante um ciclo de locomoção" [Sil05]. Os padrões de locomoção utilizados pelos robôs têm usualmente por base a locomoção de animais de igual número de membros. Para além dos tradicionais nomes de padrões de locomoção biológicos como o trote e o galope, existem muito outros padrões de locomoção com nomes e características bem definidas.

Com base no trabalho de M. Silva [Sil05] foram descritos diversos padrões de locomoção com diferentes características e finalidades. O robô descrito nesta dissertação utiliza um padrão de locomoção simétrico gerado através de uma aplicação desenvolvida pelo Professor Doutor Manuel Santos Silva para geração de padrões de locomoção com diferentes características.

Como a locomoção do robô desenvolvido nesta tese é conseguida pela introdução dos ângulos das juntas, é, teoricamente, possível utilizar diferentes padrões de locomoção para robôs quadrúpedes sendo, para esse efeito, somente necessário a sintonização dos controladores das juntas.

2.3 Robôs com locomoção por pernas

O recurso a pernas para implementação da locomoção em sistemas robóticos é relativamente recente.

A maior parte dos robôs terrestres já desenvolvidos têm como forma de locomoção rodas ou lagartas. No entanto, em determinadas situações, estas formas de locomoção tornam-se impraticáveis, apresentando dificuldades na transposição de obstáculos em terrenos irregulares.

Outro grande impulsionador do desenvolvimento da robótica com locomoção por pernas está intrinsecamente ligado com a fantasia do ser humano de procurar criar sistemas automáticos/autónomos que se assemelhem a seres existentes na natureza.

O robô TOPIO (Figura 2.6) é um robô construído por uma empresa Vietnamita, chamada TOSY, concebido especialmente para jogar Ping-Pong.

Embora um robô com pernas seja muito polivalente, o tamanho e configuração das suas pernas, a distância entre pernas, o número de graus de liberdade das pernas e o padrão de locomoção mais adequado são parâmetros obrigatórios, que nem sempre são de fácil definição.

O controlo das pernas é também um problema que tem sido alvo de muito estudo. A incerteza na colocação dos pés no chão obriga à criação de algoritmos de controlo suficientemente "maduros" que permitam lidar com súbitas mudanças nos apoios do robô.

Existem diversos robôs com locomoção por pernas que retratam casos de sucesso como o BigDog [Dyn12a], o Cheetah [Dyn12b], entre outros.

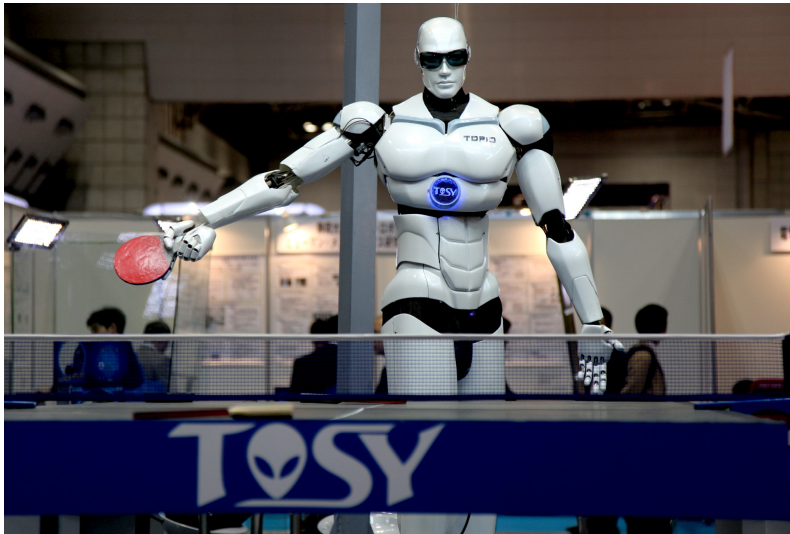


Figura 2.6: Robô que joga Ping-Pong - TOPIO.

A complexidade dos seus sistemas e os elevados custos de desenvolvimento e construção de protótipos estão cada vez mais a dar origem à utilização de sistemas informáticos em substituição dos testes iniciais de funcionamento. Desta forma é possível fazer uma primeira análise ao seu funcionamento, e em simultâneo realizar as afinações necessárias para, aquando da construção do modelo físico este esteja o mais "perfeito" possível.

Como qualquer tipo de sistema, é importante a sua correcta definição. Assim sendo, um robô com locomoção por pernas é definido pelos parâmetros a seguir referidos.

Objectivo

É primordial saber qual o propósito da construção do robô; o seu grau de sofisticação, assim como as suas características físicas irão sempre de encontro aos propósitos de utilização aqui definidos. Como exemplos de aplicações de robôs existe o BigDog [Dyn12a] criado com o objectivo de auxiliar as tropas Norte Americanas no transporte de equipamentos e mantimentos em cenários de guerra.

Número de pernas

É essencial saber qual o número de pernas pretendido para o sistema. Este valor está directamente relacionado com características de estabilidade do robô, assim como com o objectivo da sua criação.

Método de adesão

Os robôs com pernas podem, dependendo do objectivo a que se destinam estar equipados com ímanes ou dispositivos de sucção de forma a melhorar o desempenho das suas funções.

Graus de liberdade do robô

Tal como em qualquer outro sistema mecânico é importante definir, à partida, quantos graus de liberdade serão necessários para o correcto desempenho das funções a que o robô se destina. O robô quadrúpede apresentado nesta dissertação possui dois graus de liberdade por perna, conseguidos pela utilização de duas juntas rotacionais, uma no joelho e outra na anca. O corpo possui 8 graus de liberdade pela utilização de contactos mola-amortecedor entre os quatro blocos que constituem o corpo.

2.4 Actualidade da robótica

A robótica actual introduz-nos a um novo mundo onde cada vez mais os robôs farão parte do dia-a-dia. A tecnologia associada a esta disciplina e as suas inúmeras aplicações permitem o desenvolvimento de robôs como o Cheetah [Dyn12b], um robô quadrúpede cuja locomoção, tal como o nome indica, se assemelha à de uma chita (Figura 2.7). Este

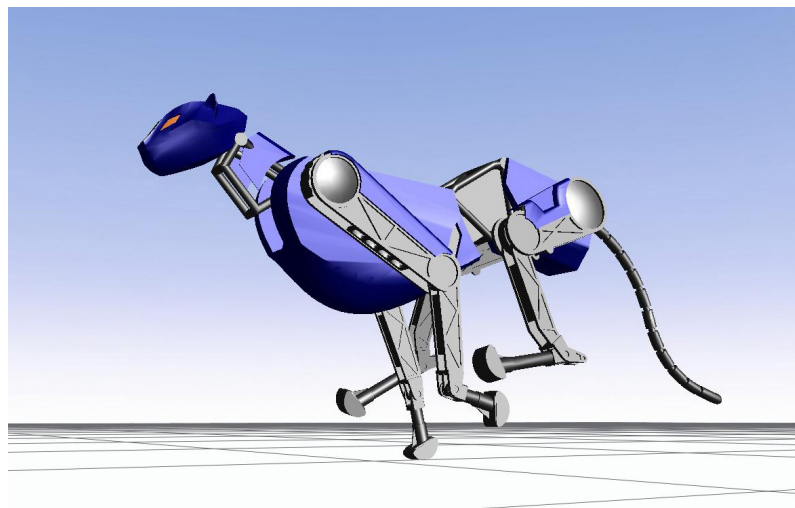


Figura 2.7: Imagem do que poderá vir a ser o robô Cheetah.

robô bateu recentemente o recorde de velocidade de um ser humano, alcançando uma velocidade de 45,54 km/h [Dyn12b].

Existem inúmeras finalidades que justificam o grande investimento na investigação nesta área. O robô ASIMO da Honda é já capaz de andar, correr e fazer operações mais complexas como, por exemplo, empurrar um carrinho, encher um copo com água ou até saltar só num pé.

Este robô, semelhante a um ser humano, mede 1,30 metros, pesa 48 kg e consegue correr a uma velocidade máxima de 9 km/h [Hon12]. Possui 6 gdl em cada perna, 2 gdl na anca, 13 gdl em cada mão, 7 gdl em cada braço e 3 na cabeça (Figura 2.8).

Um outro robô de entretenimento muito famoso é o TITAN. Criado exclusivamente para o sector do entretenimento, este robô é utilizado para animar todo o tipo de eventos

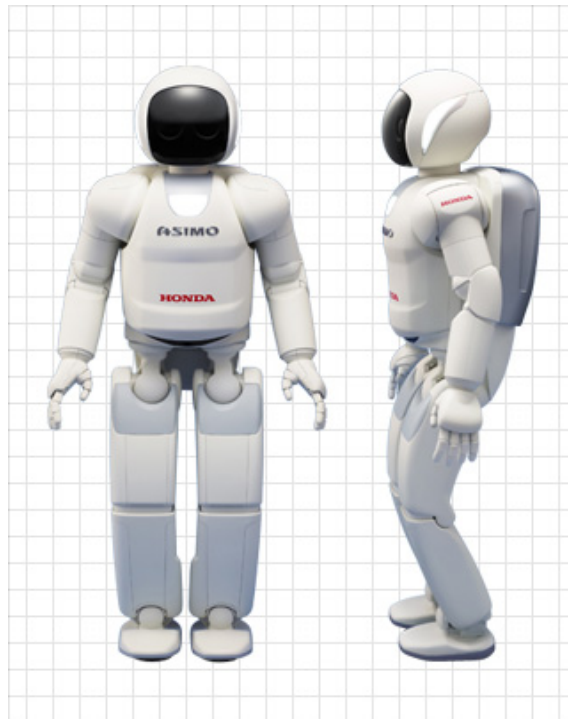


Figura 2.8: Robô ASIMO no presente depois de 10 anos de trabalho.

(Figura 2.9). Conta anedotas, dança e canta. Este robô já apareceu em programas como o *Big Brother*, *Paul O'Grady Show*, *The Gadget Show* e *Daybreak*, entre outros; contracenou com Hugh Jackman no filme *Real Steel* e apareceu nos *National Movie Awards* no Reino Unido [Rob11].

Um pouco diferente do robô ASIMO, que serve essencialmente como *marketing* da qualidade e tecnologia da Honda, o robô TITAN é utilizado como uma fonte de receitas por uma empresa que se dedica à sua exploração.

Como exemplo de robô quadrúpede é importante referir o AIBO™ da Sony®. Este robô, criado em 1999 e descontinuado em 2006, sofreu diversas fases de evolução existindo 14 modelos diferentes. Vendeu um total de 175.000 exemplares [Son12].

Na Figura 2.10a apresenta-se o primeiro modelo do AIBO. Estes robôs sofreram uma evolução muito grande ao longo dos 7 anos de produção, verificando-se a existência de uma comunidade *online* dedicada à partilha de programas e informação para melhorar o funcionamento do robô pela introdução de novas funcionalidades. Este robô, para além de brincar com os seus brinquedos, era capaz de detectar a sua plataforma de carregamento onde se deitava para recarregar as baterias e também possuía algumas funções que permitiam que o "dono" lhe ensinasse algumas tarefas tornando-o bastante interativo. O último modelo foi produzido em 2006, estando representado na Figura 2.10b.



Figura 2.9: Robô TITAN.



(a) Primeiro modelo (ERS-110)



(b) Último Modelo (ERS-7M3)

Figura 2.10: Dois exemplares do robô AIBO da Sony.

Capítulo 3

Modelo do Robô Quadrúpede

A utilização de pernas para a locomoção de um robô permite-lhe aceder a locais de outra forma inacessíveis a um veículo robótico. Se por um lado a utilização de pernas facilita o movimento do robô, permitindo a transposição de objectos de formatos diversos, por outro lado a sua implementação e controlo tornam-se bastante complexas.

Os modelos de robôs com pernas são usualmente concebidos com base em estudos da anatomia do animal que melhor se adequa às funções pretendidas para o robô. Desta forma se se quiser um robô quadrúpede veloz procurar-se-á desenhá-lo à semelhança de, por exemplo, uma chita [Dyn12b]. Para além das características anatómicas dos animais que costumam ser transpostas para a robótica, também os seus padrões de locomoção o são. Existem inúmeros padrões de locomoção, estando a sua escolha dependente da anatomia do robô, do tipo de função a executar e de outras variáveis, tais como a velocidade e estabilidade, entre outras características.

Na robótica os conceitos de cinemática e dinâmica estão presentes em todo o tipo de sistemas, desde os mais simples aos de maior complexidade.

Nos robôs com pernas estes estudos tornam-se mais complexos devido às diversas condições que devem ser impostas.

3.1 Cinemática e dinâmica de robôs com pernas

A cinemática e a dinâmica complementam-se no estudo do movimento dos robôs e das forças que lhe dão origem.

A cinemática é o ramo da física responsável pelo estudo e descrição do movimento de um corpo ignorando as suas causas. A dinâmica é responsável pelo estudo e descrição das forças necessárias para dar origem a um determinado movimento.

No caso da robótica, a cinemática define o movimento pretendido do robô, através da actuação das suas juntas. A dinâmica define a influência que cada componente do robô tem na sua locomoção assim como estabelece qual o efeito do contacto do robô com o meio circundante.

Inicialmente foram estudados os documentos [Sil05] e [Bar09] de forma a perceber exactamente qual a melhor abordagem ao problema apresentado.

3.2 Modelo cinemático

A cinemática permite definir matematicamente o movimento de uma determinada partícula ou conjunto delas. É utilizada em robótica como forma de determinação da posição das diversas juntas do robô de forma a alcançar um determinado ponto.

A análise cinemática de um determinado robô permite determinar as diferentes combinações angulares ou prismáticas que permitirão alcançar um determinado ponto previamente estabelecido.

Existem dois tipos de cinemática: directa e inversa (Figura 3.1). Na cinemática directa são utilizados os ângulos das juntas para calcular a posição final do pé do robô; na cinemática inversa calculam-se os ângulos necessários nas juntas do robô por forma a alcançar a posição pré-definida do pé do robô.

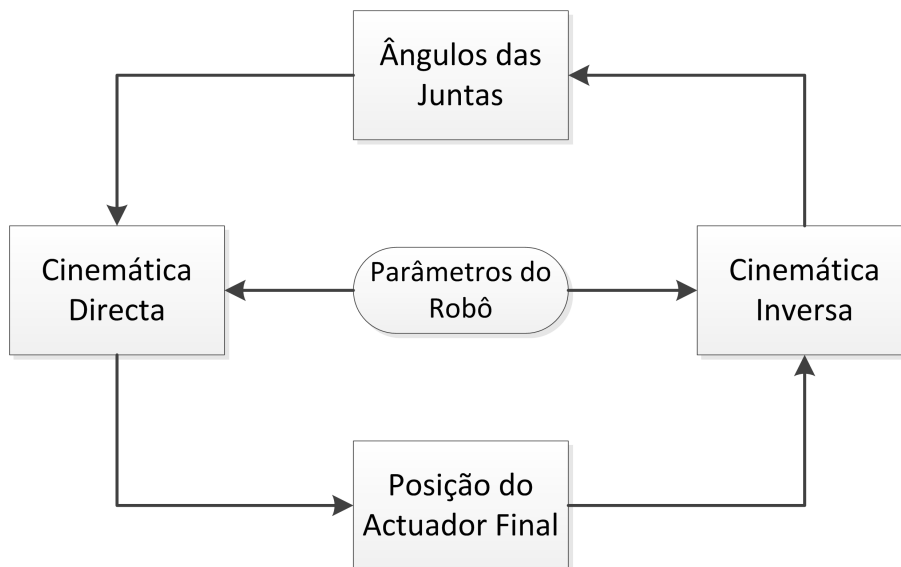


Figura 3.1: Esquema das relações entre cinemática directa e inversa.

A utilização de cinemática directa ou inversa depende somente das condições de resolução do movimento do robô. Dependendo do objectivo do estudo e dos dados que o programador possua deve-se optar pela utilização da cinemática directa ou inversa.

Os cálculos da cinemática directa são bastante mais simples do que os utilizados na cinemática inversa. Possuindo os ângulos das juntas e as dimensões da perna do robô com

alguma facilidade, por relações trigonométricas, se conseguem calcular as coordenadas do ponto onde se situa o pé do robô. No entanto, na maioria dos casos o utilizador sabe a posição que deseja para o pé e pretende calcular os ângulos necessários nas juntas para alcançar a posição desejada. A resolução da cinemática inversa é usualmente mais complexa do que a cinemática directa. Num braço robótico com dois graus de liberdade, para uma só posição existem, para a maioria dos pontos, duas formas de a alcançar; esta multiplicidade de soluções complica os cálculos para obtenção de uma solução. As equações envolvidas na resolução da cinemática inversa são extensas e complexas o que facilmente induz em erro.

Para a correcta execução deste trabalho foi necessário compreender os princípios da cinemática de um robô quadrúpede com dois graus de liberdade por perna.

Em primeiro lugar foram definidas as posições das pernas e estas foram numeradas como consta na Figura 3.2.

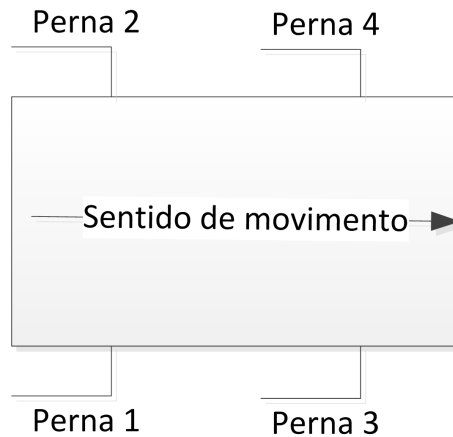


Figura 3.2: Numeração de pernas robô.

Em seguida definiu-se a orientação da junta que conecta a parte superior à parte inferior dos membros. Optou-se pela utilização de um modelo com a junta para a frente.

Um dos objectivos deste modelo era a possibilidade do mesmo funcionar com diferentes padrões de locomoção. Para isso foram colocados actuadores na junta do joelho e da anca. Estes actuadores recebem as posições angulares que permitem definir o padrão de locomoção do robô.

Embora este robô permita a introdução de diferentes trajectórias em cada um dos membros, no caso do exemplo criado para demonstrar o funcionamento do robô optou-se por definir o movimento da perna 4 igual ao da perna 1 e o da perna 3 igual ao da perna 2.

Também é importante definir que os ângulos das juntas sejam medidos uns em relação aos outros (Figura 3.3). Desta forma o ângulo da anca é medido em relação à posição horizontal do corpo e o ângulo do joelho é medido em relação a uma linha imaginária no prolongamento da parte superior da perna.

Esta relação angular é a mesma que o Simulink™ utiliza, sendo bastante comum em trabalhos de robótica por facilitar a visualização da posição dos membros [MFS04].

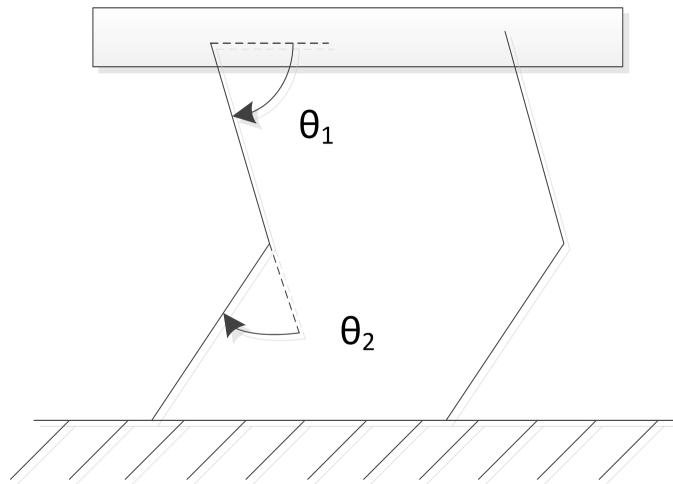


Figura 3.3: Disposição dos ângulos das juntas.

Com este trabalho pretende-se obter alguma flexibilidade em relação ao padrão de locomoção testado; desta forma foram definidas variáveis, utilizadas para a definição do movimento pretendido do robô (Tabela 3.1). Este método permite para um mesmo robô testar diversos padrões de locomoção verificando o que melhor se ajusta às necessidades do utilizador. As variáveis apresentadas na Tabela 3.1 são definidas como matrizes com 2 colunas, na primeira coluna são escritos os instantes de tempo, na segunda coluna são colocados os valores pretendidos para as juntas.

Tabela 3.1: Variáveis de posição das juntas da anca e do joelho.

	Perna 1	Perna 2	Perna 3	Perna 4
Posição Angular da Anca	<i>hip_1</i>	<i>hip_2</i>	<i>hip_3</i>	<i>hip_4</i>
Posição Angular do Joelho	<i>knee_1</i>	<i>knee_2</i>	<i>knee_3</i>	<i>knee_4</i>

3.3 Modelo dinâmico

A dinâmica permite calcular o efeito da aplicação de uma força/binário num determinado corpo ou a força/binário necessária para que um determinado corpo se mova como pretendido.

Tal como na cinemática, também existe dinâmica directa e dinâmica inversa. A dinâmica directa permite, com base na aplicação de uma força/binário, determinar o movimento que esta irá provocar num determinado corpo. No caso do robô documentado neste trabalho tem mais interesse a dinâmica inversa. O cálculo da dinâmica inversa de um determinado movimento resulta nas forças/binários que dão origem a esse movimento.

Modelo do Robô Quadrúpede

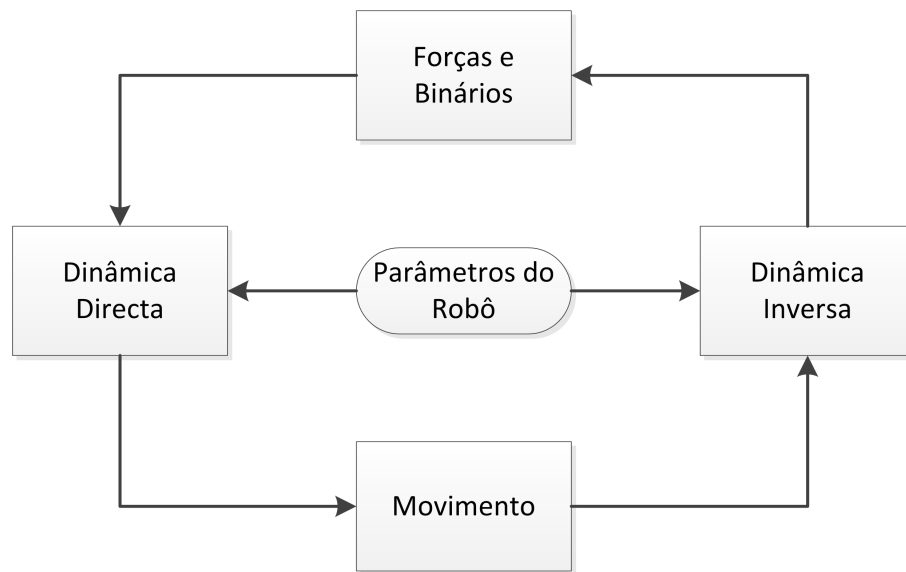


Figura 3.4: Dinâmica directa e inversa.

As expressões utilizadas no cálculo da dinâmica directa ou inversa estão fora do âmbito deste trabalho pelo que não serão aqui apresentadas. Para que o programa consiga calcular correctamente as forças necessárias para o movimento pretendido, têm que ser previamente definidas as matrizes de inércia dos diversos componentes utilizados no corpo.

Numa primeira abordagem, o robô foi concebido como tendo um corpo rígido e quatro membros de locomoção igualmente espaçados entre si. Posteriormente, com base em estudos prévios concluiu-se que, de forma a aumentar a eficiência e a estabilidade da locomoção, deveria-se optar pela modelação de um corpo complacente passando este a ser constituído por quatro blocos interligados por sistemas do tipo mola-amortecedor (Figura 3.5).

3.4 Modelo de contacto pé-solo

O modelo de contacto com o solo é uma parte fundamental do modelo do robô. Quando o pé do robô toca no solo exerce uma determinada força. De acordo com a terceira lei de Newton sabe-se que o solo aplica no pé uma força de igual intensidade e sentido oposto.

O movimento de um corpo numa determinada superfície é conseguido devido a este par acção-reacção.

Para simplificar o modelo considera-se um ponto de contacto na extremidade inferior da perna. Quando esse ponto entra em contacto com um determinado ponto no solo é exercida uma força tridimensional no solo e uma força com igual intensidade e sentido oposto no robô. Para um corpo em movimento rectilíneo, separando-se esta força nas

Modelo do Robô Quadrúpede

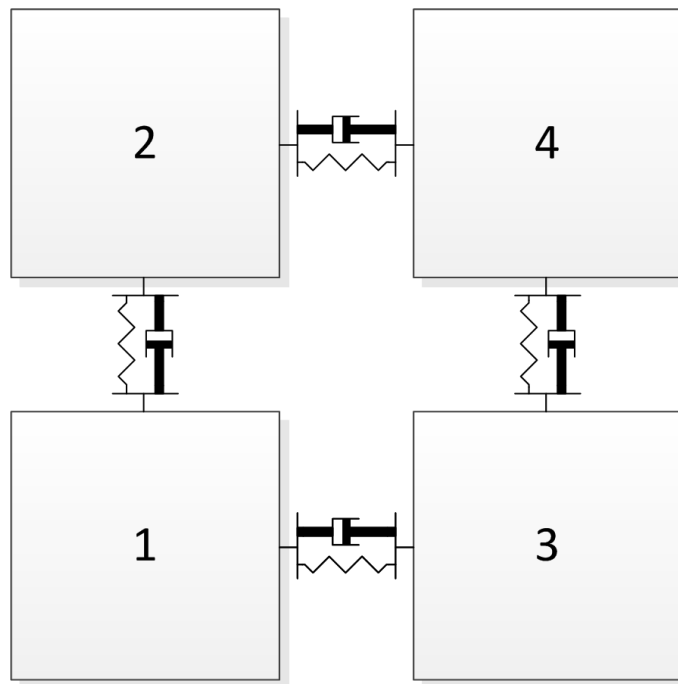


Figura 3.5: Esquema do corpo do robô apresentado neste trabalho.

suas componentes obtém-se uma componente segundo o eixo x e outra segundo o eixo y (Figura 3.6).

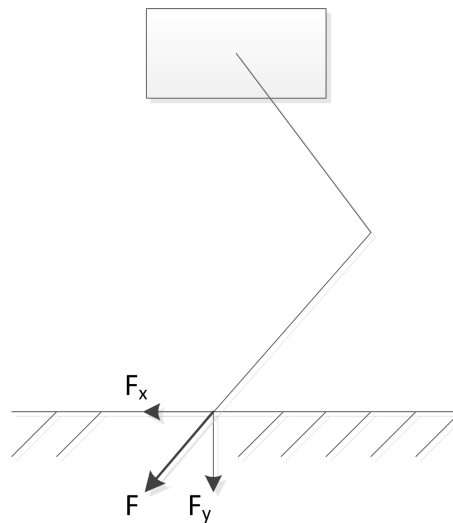


Figura 3.6: Força de contacto com solo.

Existem diversos modelos de análise do contacto dos pés com o solo. O método utilizado neste trabalho calcula as forças de contacto através de um modelo aproximado do solo com base em modelos obtidos pelo estudo da mecânica dos solos (Figura 3.7).

Os dois principais métodos de simulação do contacto com o solo são contactos rígidos ou contactos complacentes. Os contactos rígidos não possuem qualquer tipo de forma de

Modelo do Robô Quadrúpede

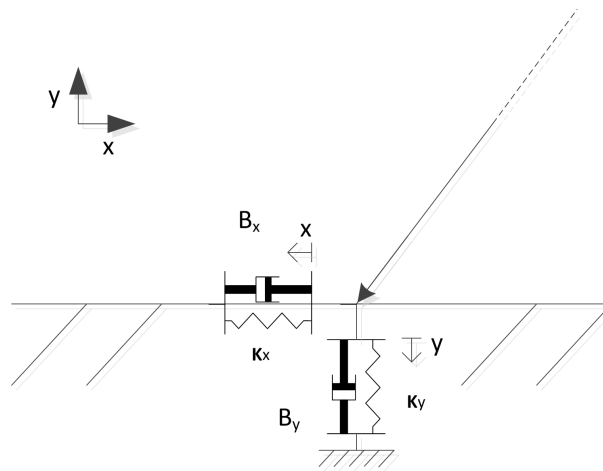


Figura 3.7: Modelo de contacto com o solo.

dissipação de energia do embate, podem ser utilizados como método simples de simulação do contacto entre dois corpos rígidos.

No âmbito desta tese não faria sentido utilizar um contacto rígido porque se pretende simular o contacto do pé do robô com o solo onde este se desloca.

Optou-se pela utilização de um modelo de contacto que recorre a um sistema do tipo mola-amortecedor (Figura 3.7). Inicialmente procurou-se aplicar o modelo linear do sistema mola-amortecedor descrito através das equações (3.1) e (3.2).

$$F_x = -K_x(x - x_0) - B_x\dot{x} \quad (3.1)$$

$$F_y = -K_y(y - y_0) - B_y\dot{y} \quad (3.2)$$

Verificou-se que o sistema não funcionava correctamente porque, tal como apresentado no trabalho de O. Bruneau [OB97] e se pode constatar pela análise da equação (3.2), no instante em que a perna toca no solo, a componente da força normal de contacto da mola ($-K_y(y - y_0)$) era quase nula e a componente do amortecedor ($-B_y\dot{y}$) era positiva e muito elevada, funcionando como uma força que repelia o pé do chão. Como também se depreende da análise da equação (3.2) no momento em que o pé tenta abandonar o solo a velocidade é positiva dando origem a uma força de amortecimento muito elevada no sentido negativo. Como a força efectuada pela mola é muito pequena, a força resultante tem sentido negativo e uma intensidade suficiente para não deixar o pé sair do solo. Esta força desproporcional não corresponde à realidade, não sendo por isso um método fiável de simulação.

A solução passa então pela utilização de um sistema mola-amortecedor não linear que permita o aumento gradual das forças tornando o contacto com o solo mais suave. Encontrou-se na literatura [Sil05] o modelo descrito pela equação (3.3). Neste modelo

Modelo do Robô Quadrúpede

esse problema deixa de acontecer verificando-se uma maior progressividade da evolução da força de contacto tanto no contacto inicial com o solo como ao abandonar o mesmo.

$$F_y = -K_y y - B_y (-y)^m \dot{y} \quad (3.3)$$

Na equação (3.3) o expoente m permite ao utilizador ajustar com maior precisão as características do solo. No trabalho descrito nesta dissertação só se irão considerar duas componentes do contacto que daqui em diante irão ser chamadas força normal (exercida na direcção y) e força tangencial (exercida na direcção x). A componente normal da força é responsável por garantir que o robô se encontra sempre acima do solo e a componente tangencial impõe movimento ao robô.

Capítulo 4

Simmechanics

O Simmechanics™ é uma ferramenta de modelação tridimensional que permite simular modelos de vários corpos como robôs, suspensões de veículos e trens de aterragem de aviões [Matc]. O seu interface é relativamente simples, funcionando por meio de blocos que representam corpos, juntas, restrições de movimento e forças. Existem neste momento duas gerações de Simmechanics™. Este trabalho foi realizado recorrendo somente a elementos da primeira geração. Os diversos elementos são unidos utilizando linhas que representam os sinais transmitidos de uns blocos para os outros.

Para modelar um sistema físico qualquer é muito importante ter um grande conhecimento prévio do sistema. As suas características e limitações dão origem a problemas de difícil resolução quando o sistema ainda é relativamente desconhecido.

Embora o Simmechanics™ simplifique bastante o processo de modelação/simulação do sistema pretendido, a sua implementação pode-se tornar bastante complexa.

Neste trabalho procurou-se, numa primeira fase, estudar o manual do Simmechanics™. Embora o manual esteja bastante completo, a melhor forma de aprender a trabalhar com este programa é pela experimentação. Assim, numa fase inicial do trabalho, estudaram-se os diversos exemplos disponíveis na ajuda do programa.

Nesta fase é importante definir-se a constituição da perna (Figura 4.1), desta forma consideram-se pernas do robô todos os membros que irão permitir a sua locomoção. As pernas estão ligadas ao corpo do robô através de juntas às quais ir-se-à chamar "ancas". Cada perna é constituída por duas partes, a superior que neste trabalho se designa por "perna superior" e está ligada ao corpo do robô pela anca e na outra extremidade à "perna inferior" através de uma junta que será chamada de joelho.

4.1 Blocos básicos do Simmechanics

Os blocos necessários para a modelação de um robô são os que se discriminam em seguida.

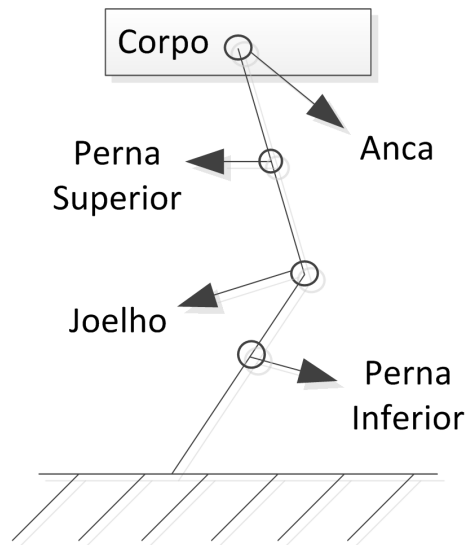


Figura 4.1: Esquema da constituição da perna do robô.

4.1.1 Corpo (*Body*)

O bloco do tipo corpo (Figura 4.2) permite modelar qualquer objecto rígido.

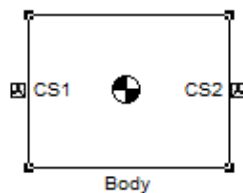


Figura 4.2: Bloco Corpo (*Body*).

No presente trabalho foi utilizado para modelar os diversos blocos que constituem o corpo do robô assim como ambas as secções das pernas.

Em qualquer modelo os diversos corpos constituintes devem ser os primeiros blocos a adicionar.

Quando se começa a modelar um qualquer equipamento já se tem uma ideia da aparência final do mesmo. Nesta primeira janela (Figura 4.3) devem ser introduzidas as informações respeitantes à massa do elemento, à sua inércia e os sistemas de coordenadas (SC) aos quais irá ser ligado. A massa deve ser introduzida directamente na janela ou através de uma variável que posteriormente será definida num ficheiro Matlab[®] podendo ser introduzida em quilogramas, gramas, miligramas, libras (lbm), onças (oz) e slug (equivalente a aproximadamente 14,59 kg).

A matriz de inércia de um determinado corpo permite ao Simmechanics[™] determinar de que forma o corpo se irá comportar quando animado com uma determinada força/binário num qualquer eixo. Esta matriz é de extrema importância visto ser essa informação

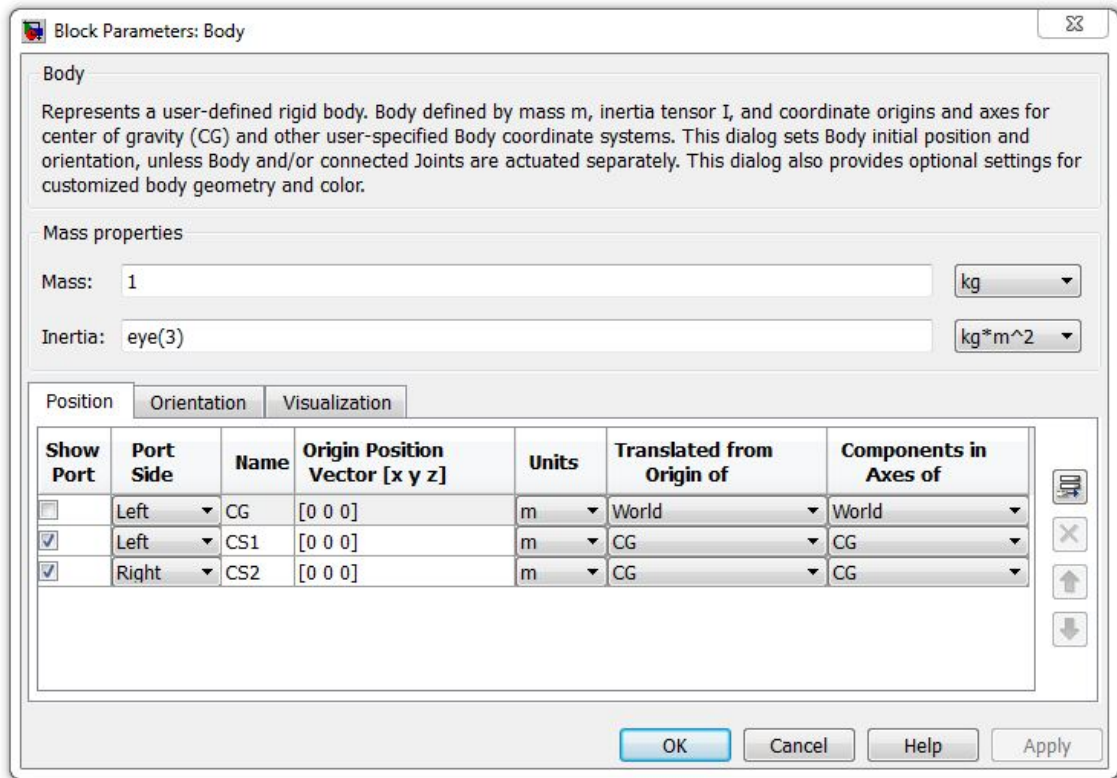


Figura 4.3: Configurações do bloco "Body", janela "Position".

que permite ao programa calcular de que forma os diversos componentes do sistema irão interagir.

Para corpos simétricos em torno dos eixos coordenados a matriz de inércia é definida por valores constantes na diagonal principal sendo os restantes valores nulos.

$$I = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix} \quad (4.1)$$

Em casos nos quais o corpo possui um formato mais complexo a sua matriz de inércia pode ser calculada com base em desenhos CAD. Nesses casos a matriz de inércia é composta não só pelos momentos de inércia que ocupam a diagonal principal, mas também pelos produtos de inércia representados pelos restantes valores.

$$I = \begin{bmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{bmatrix} \quad (4.2)$$

A compreensão do propósito dos produtos de inércia é bastante complexa. Uma explicação simples e suficiente, tendo em conta o âmbito deste trabalho, é que enquanto os momentos de inércia da diagonal principal descrevem o comportamento do corpo em

torno de três eixos coordenados em relação aos quais o corpo é simétrico, os produtos de inércia descrevem o comportamento do corpo em relação a eixos de rotação diferentes dos coordenados.

Ainda na janela de posição (Figura 4.3) é importante definir a posição dos diversos SC que permitirão a interação entre os diversos elementos do sistema. Ao definir o sistema de coordenadas é importante definir correctamente a sua origem. Um sistema de coordenadas novo é definido indicando qual a sua posição relativamente a uma determinada origem.

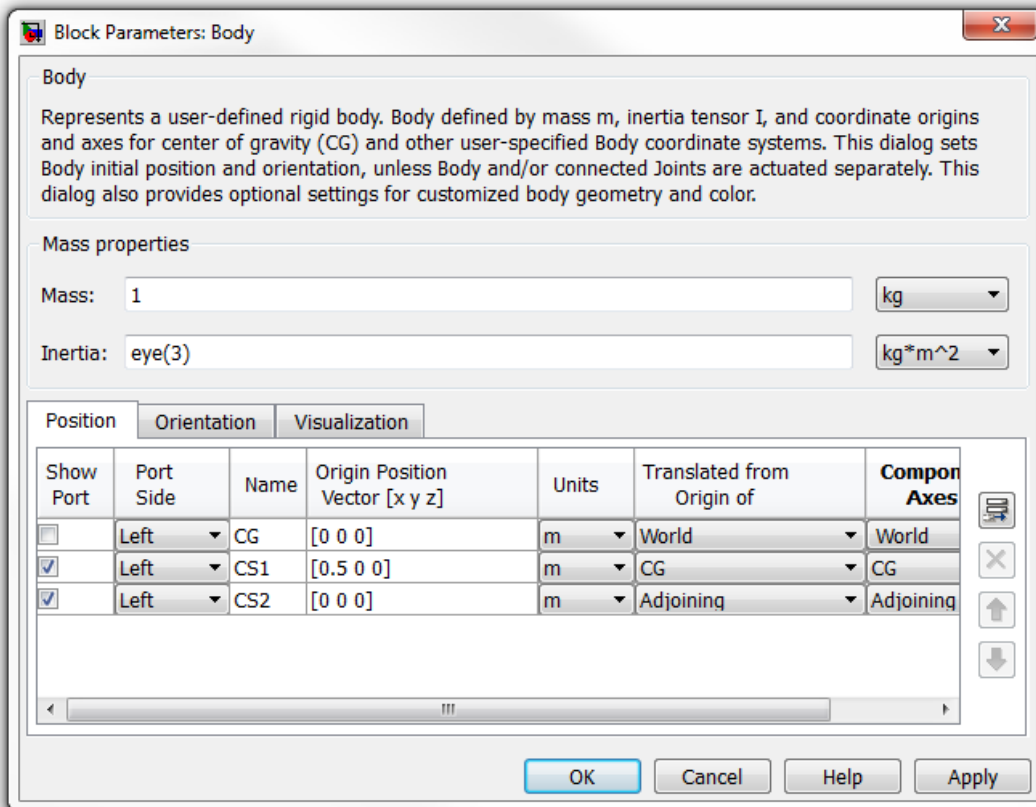


Figura 4.4: Exemplo de utilização dos sistemas de coordenadas.

No caso da Figura 4.4 observa-se que o CS1 (*Coordinate System 1*) se encontra no plano definido pelas coordenadas yy e zz do CG e deslocado 0,5 metros desse plano no eixo dos xx . Na janela "*Components in Axes of*" define-se qual a orientação do sistema de coordenadas. Não há uma forma mais correcta do que a outra de definir estes parâmetros desde que exista coerência suficiente para não causar confusão.

No caso do CS2 exemplifica-se a parametrização deste sistema de coordenadas com base, não no sistema de coordenadas do CG, mas através do sistema de coordenadas do bloco ao qual se irá conectar. Este tipo de parametrização é especialmente útil no caso de se pretender tornar o sistema flexível a alterações nos tamanhos dos corpos. Com esta opção pode-se definir somente o CG de um corpo com posição absoluta ao ambiente de simulação e depois utilizar esta opção para definir os SC relativamente uns aos outros.

Podem ser utilizadas diversas unidades de medida para definir estas condições como sejam metros, centímetros, milímetros, quilómetros, micrómetros, polegadas, pés, milhas e jardas.

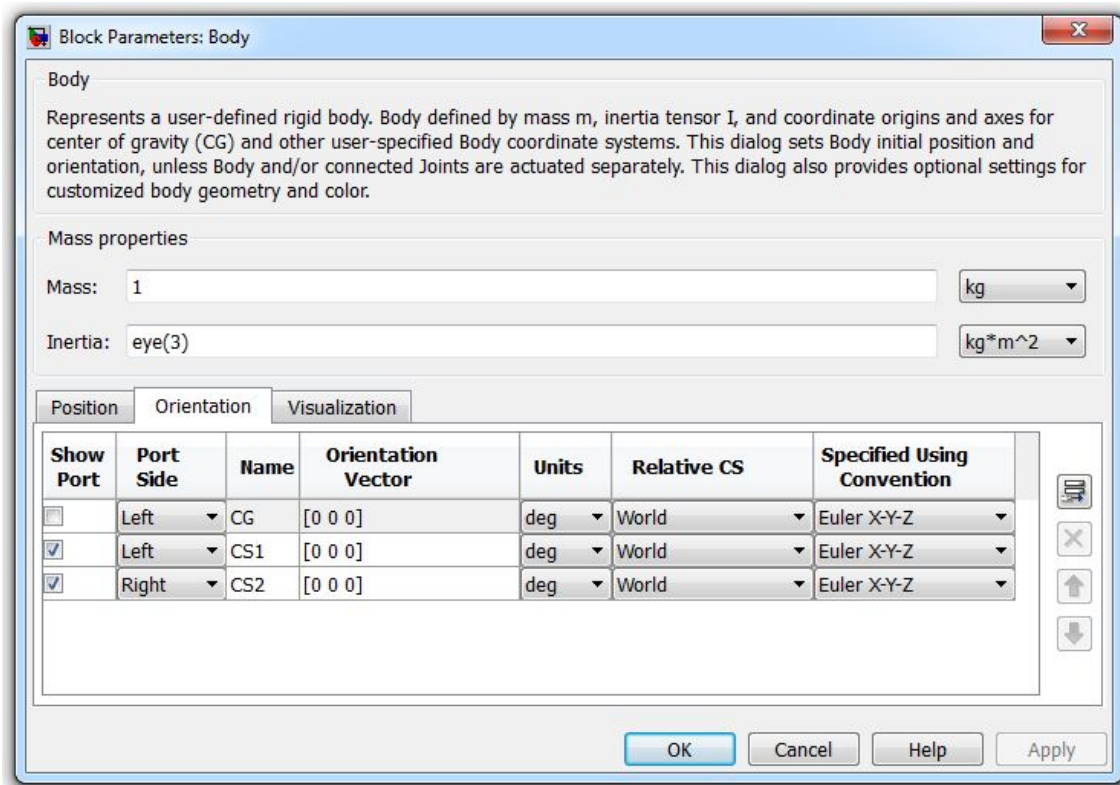


Figura 4.5: Janela de definição da orientação dos sistemas de coordenadas.

Ainda dentro da janela de parametrização do corpo existe a opção de alterar a orientação dos SC definidos previamente. Estes dados são definidos na sub-janela "*Orientation*" (Figura 4.5). Embora estas opções não tenham sido utilizadas no trabalho descrito nesta dissertação, estas opções poderão ser úteis noutro tipo de sistema. Existem muitos métodos de definição da orientação dos SC. Mais informação sobre os diversos métodos e a sua aplicação pode ser encontrada no manual de ajuda do programa (página 29 do manual de utilizador do Simmechanics™ intitulado mech_ug.pdf) [Matb].

Para total parametrização do bloco de corpo é também possível definir o tipo de visualização pretendida para este corpo, como se mostra na Figura 4.6, sendo inclusivamente possível utilizar desenhos de CAD na simulação tornando-a muito mais rica graficamente.

4.1.2 Junta Rotacional (*Revolute Joint*)

As juntas são usualmente utilizadas para definir a forma como uns corpos se movimentam em relação aos outros. A junta rotacional acrescenta um grau de liberdade ao robô permitindo rotação em torno de um eixo definido no próprio bloco.

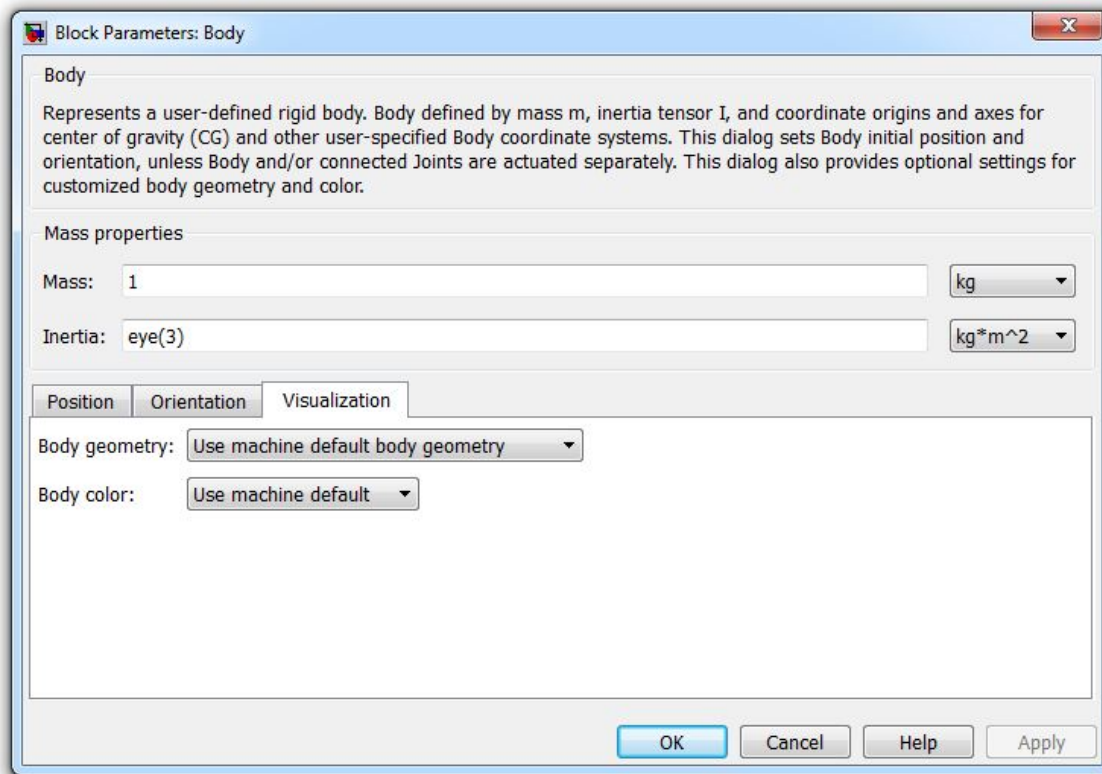


Figura 4.6: Janela de definição das opções de visualização do corpo.

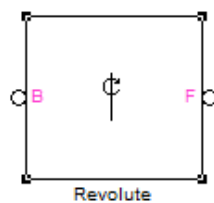


Figura 4.7: Junta rotacional.

Este bloco que se apresenta na Figura 4.7, embora seja muito importante na modelação do sistema, é relativamente fácil de configurar.

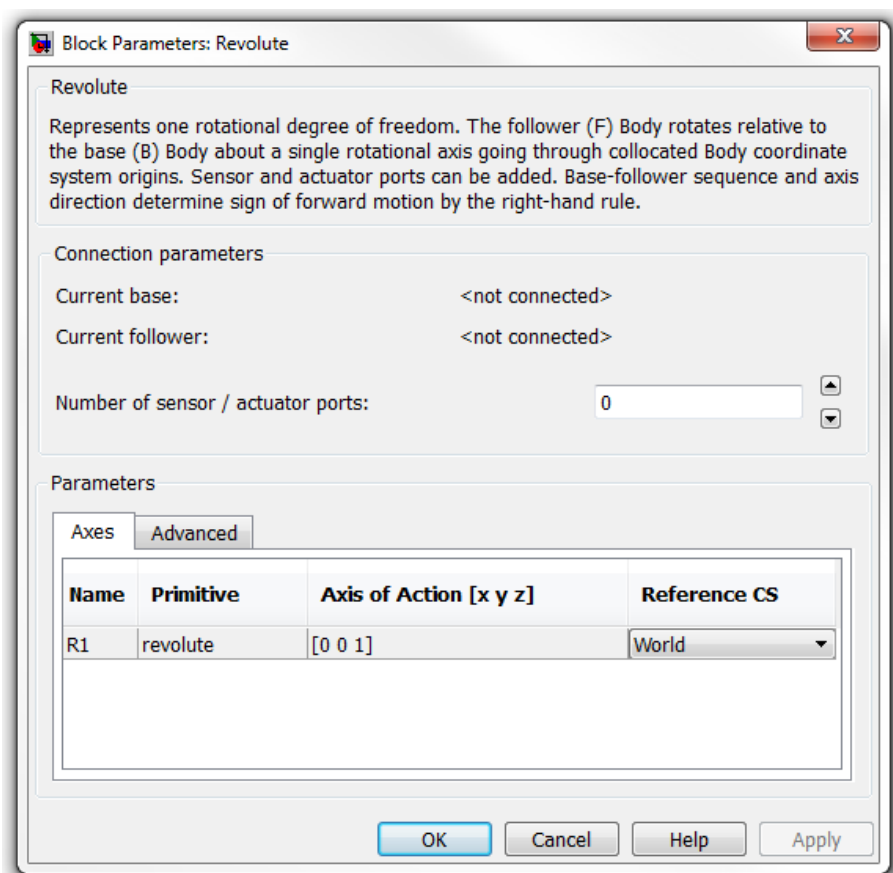


Figura 4.8: Janela de definição das características da Junta Rotacional.

No robô apresentado neste trabalho as juntas rotacionais são utilizadas na anca e nos joelhos. Esta junta permite às duas componentes das pernas rodarem em torno de um ponto de fixação num plano pré-definido na parametrização do bloco.

Na Figura 4.8 observam-se os parâmetros disponíveis para configuração do bloco. É possível aqui escolher o número de sensores e actuadores que irão ser conectados neste bloco. No caso do presente trabalho serão o *Joint Initial Conditions* (JIT), *Joint Sensor* e *Joint Actuator*.

4.1.3 Actuador de Junta (*Joint Actuator*)

O actuador de junta (cujo bloco se apresenta na Figura 4.9) permite actuar todo o tipo de juntas disponíveis no Simmechanics™.

Os parâmetros deste bloco adaptam-se automaticamente à junta à qual é acoplado (Figura 4.10).

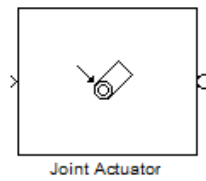


Figura 4.9: Actuador de Junta.

No caso da junta rotacional, por só permitir a actuação em torno de um eixo de rotação, os únicos parâmetros a definir são o tipo de actuação.

As juntas podem ser actuadas por forças/binários ou por sinais de posição (Figura 4.11). Dependendo da junta à qual o bloco actuador é conectado este automaticamente se modifica para apresentar as opções correctas. No caso de juntas rotacionais é possível a introdução de binários ou, no caso de movimento, pode-se introduzir a posição, velocidade e aceleração angulares.

4.1.4 Actuador de Corpo (*Body Actuator*)

O actuador de corpo, representado na Figura 4.12, permite actuar qualquer corpo. Este bloco aplica uma força tridimensional no ponto do corpo ao qual está conectado.

Pela definição de múltiplos SC num só corpo é possível actuar o mesmo em qualquer ponto fora do seu CG. A parametrização deste bloco é bastante simples sendo somente necessário definir três parâmetros, como se pode constatar pela observação da Figura 4.13.

O primeiro parâmetro indica em relação a que sistema de coordenadas o Simmechanics™ irá aplicar a força ou binário introduzido. O sistema de coordenadas do "World" é fixo, não variando no tempo, o sistema de coordenadas local pode ser fixo ou não dependendo do movimento dos corpos.

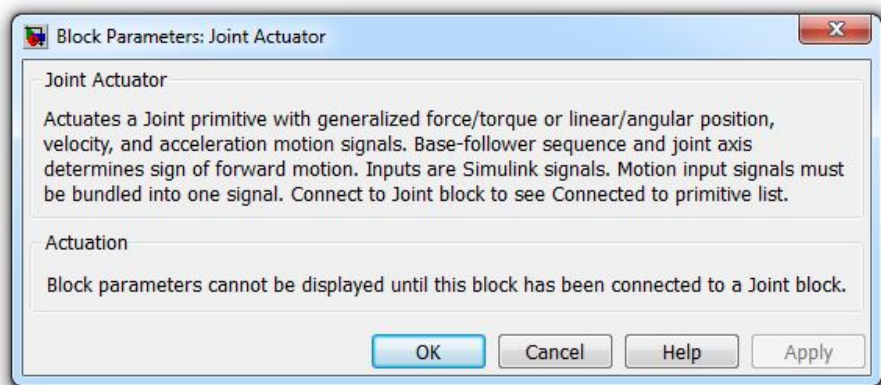
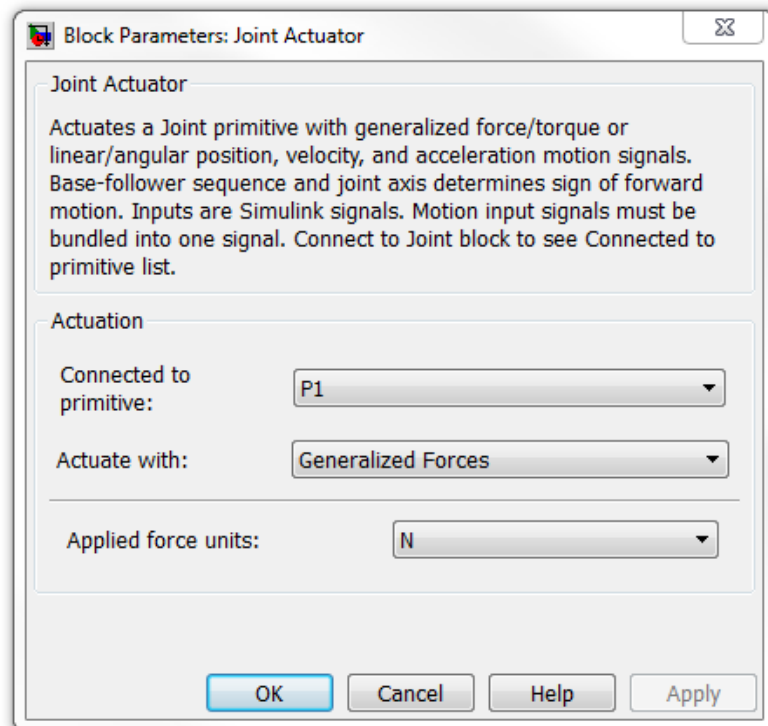
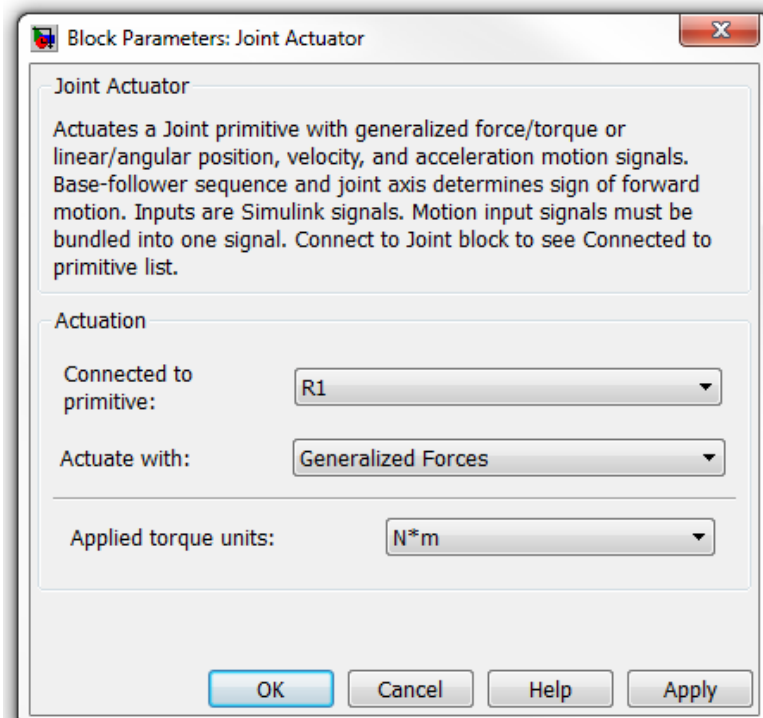


Figura 4.10: Janela inicial do actuador de junta.

Simmechanics



(a) Força



(b) Binário

Figura 4.11: Parametrização do Actuador de Juntas.

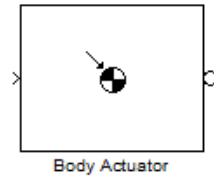


Figura 4.12: Actuador de Corpo.

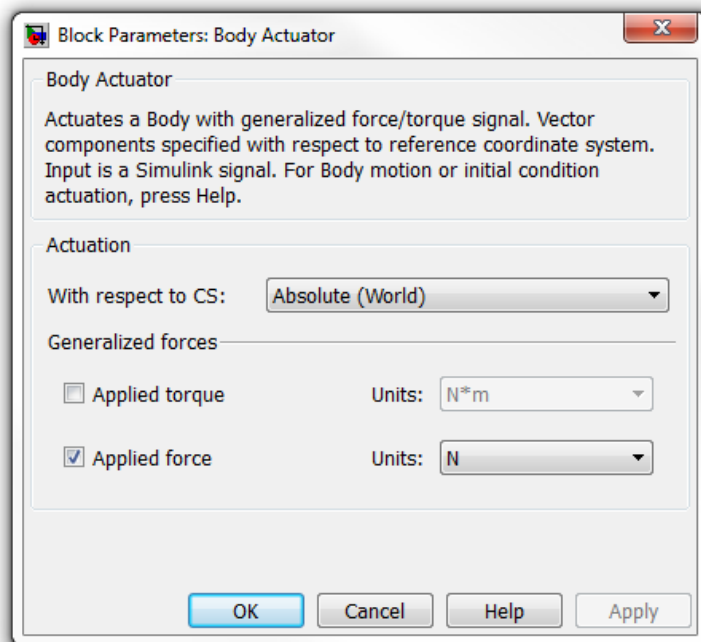


Figura 4.13: Parametrização do Actuador de Corpo.

O segundo parâmetro a introduzir é o tipo de actuação pretendida: esta pode ser um binário de rotação ou uma força prismática. Por último devem ser seleccionadas as unidades de actuação.

4.1.5 Bloco de Referência de Solo (*Ground*)

O bloco de solo, apresentado na Figura 4.14, permite estabelecer a relação entre o robô e o "Mundo".

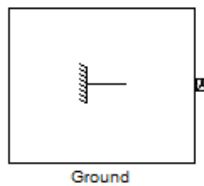


Figura 4.14: Bloco Solo (*Ground*).

Este bloco é essencial para o funcionamento de qualquer simulação. Todo o sistema tem que estar referenciado a, pelo menos, um ponto fixo no "Mundo" virtual (Figura 4.15); esse ponto fixo é definido pelo bloco *Ground*.

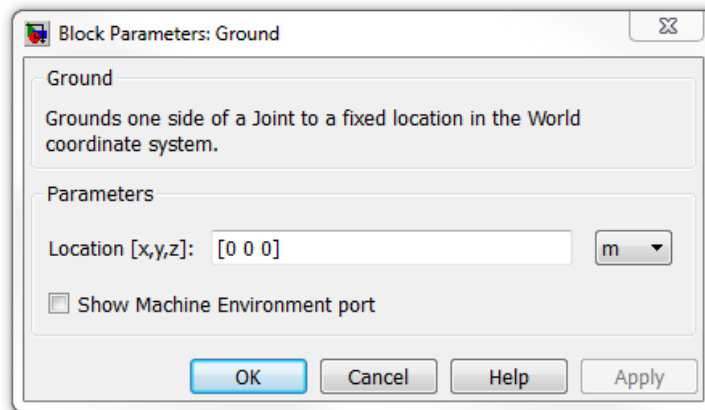


Figura 4.15: Parâmetros de configuração do bloco "*Ground*" (*Ground*).

Outra função deste bloco é a sua conexão ao bloco "*Machine Environment*", explicado de seguida.

4.1.6 Bloco de parametrização do ambiente de simulação (*Machine Environment*)

Neste bloco, representado pela Figura 4.16, são ajustadas diversas propriedades da simulação, tais como: vector de força de gravidade, método de análise (Cinemática, Dinâmica Directa, Dinâmica Inversa e *Trimming*), tolerâncias e método de visualização.

Simmechanics

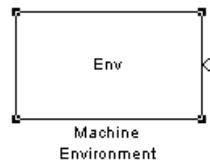


Figura 4.16: Bloco *Machine Environment*.

Na primeira janela (Figura 4.17) é importante definir correctamente as tolerâncias mecânicas tanto prismáticas como de rotação. Este valor indica o erro máximo permitido durante a simulação; quanto mais preciso o modelo mais tempo demorará a simular.

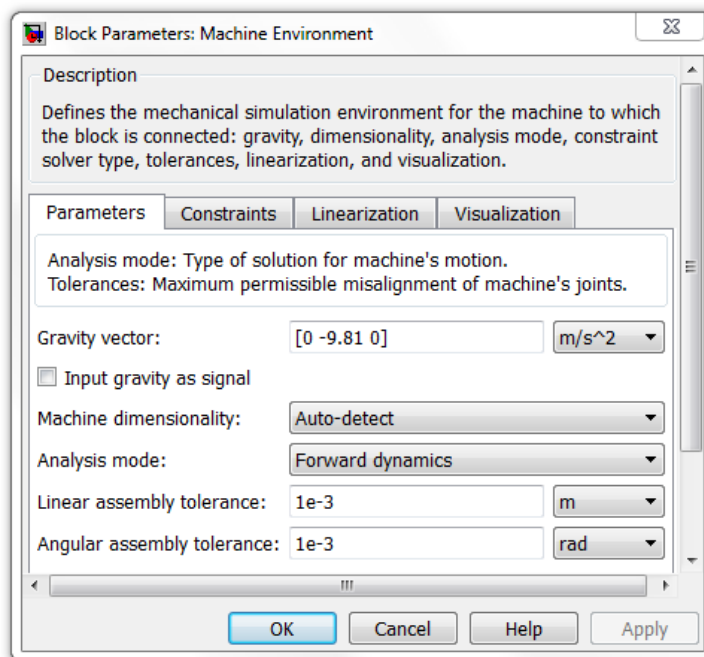


Figura 4.17: Parâmetros *Machine Environment*.

Nesta janela também é importante definir a dimensionalidade do modelo. Os modelos mais complexos são mais "pesados" demorando mais tempo a simular. Se a máquina modelada só se movimentar em duas dimensões deve ser utilizado o método de simulação 2D para que a simulação demore o menor tempo possível.

Na janela *Constraints* define-se o método e limites que o computador deve utilizar para resolver as limitações mecânicas impostas por blocos da biblioteca *Constraints & Drivers* (Figuras 4.18 e 4.19).

É possível configurar o método de linearização do modelo mecânico assim como indicar o tipo de perturbação desejado de forma a tornar o sistema mais ou menos preciso. Esta configuração faz-se na janela "*Linearization*" apresentada na Figura 4.20.

A última janela de parametrização deste bloco permite a definição do método de visualização desejado. Podem ser introduzidas imagens de CAD das diversas peças do sistema

Simmechanics

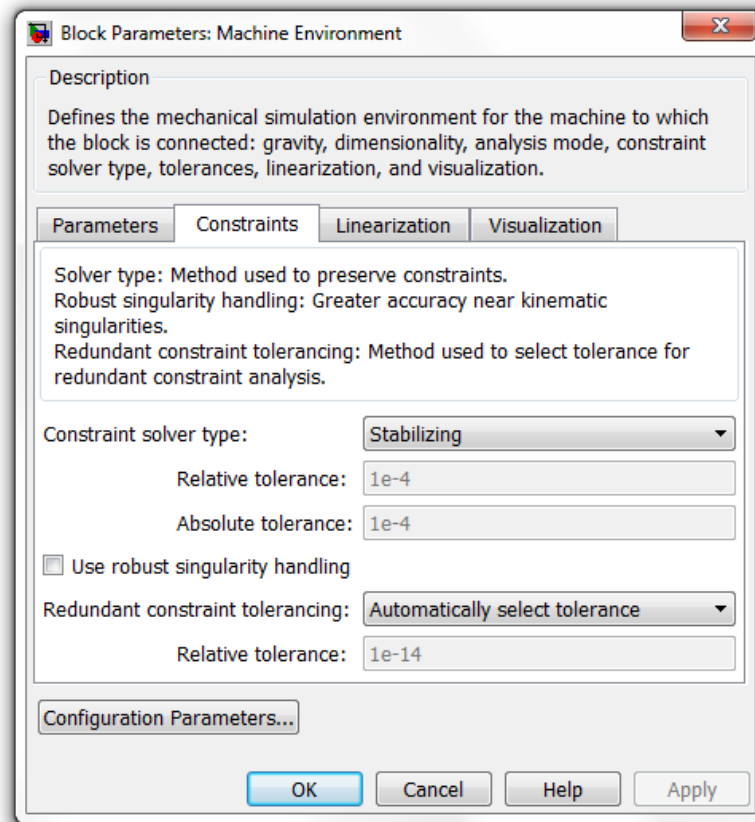


Figura 4.18: Parâmetros da janela *Constraints*.

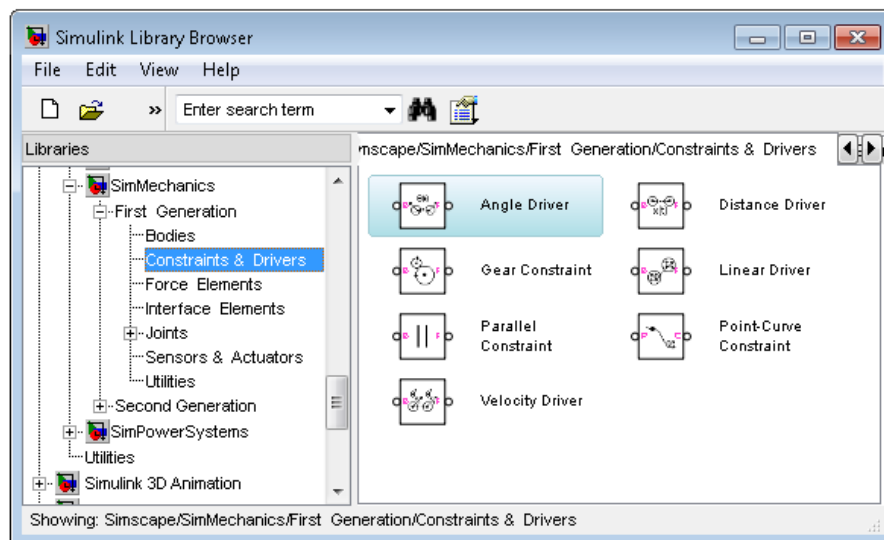


Figura 4.19: Blocos *Constraints & Drivers*

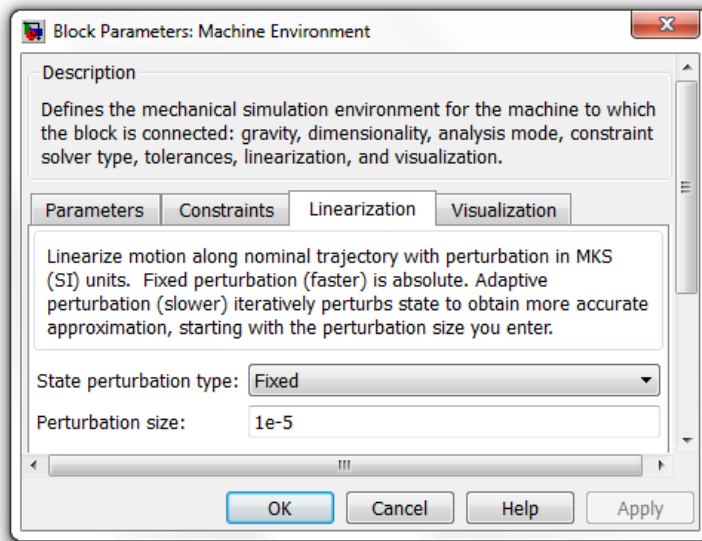


Figura 4.20: Parâmetros da janela *Linearization*.

de forma a melhorar a sua visualização (Figura 4.21).

4.1.7 Sensor de Corpo (*Body Sensor*)

O sensor de corpo, representado na Figura 4.22, permite ler alguns parâmetros de um sistema de coordenadas num corpo. Estes parâmetros podem ser: posição, velocidade linear, velocidade angular, matriz de rotação, aceleração linear e aceleração angular.

Para cada um destes parâmetros podem ser definidas diferentes unidades de medida (Figura 4.23).

O sensor de corpo permite seleccionar qual o referencial pretendido para o cálculo dos diversos parâmetros, tais como posição, aceleração, força, entre outros. A saída do bloco pode ser feita através de uma só porta, ou através de uma porta para cada sinal. Seja qual for o sinal, este traz a referência aos três eixos sendo, em seguida, necessário utilizar um demultiplexador ou um selector de sinal (Figura 4.24) para obter somente a componente do sinal pretendido.

Os sinais obtidos a partir da leitura dos sensores podem ser posteriormente utilizados para cálculos internos, exportados para o ambiente de trabalho do Matlab[®] ou apresentados em gráficos.

4.1.8 Sensor de Junta (*Joint Sensor*)

Os parâmetros lidos por este sensor (Figura 4.25) dependem do bloco ao qual é ligado.

Quando é conectado a uma junta prismática permite ler a posição, velocidade e acelerações lineares; quando ligado a uma junta rotacional permite ler a posição, velocidades e

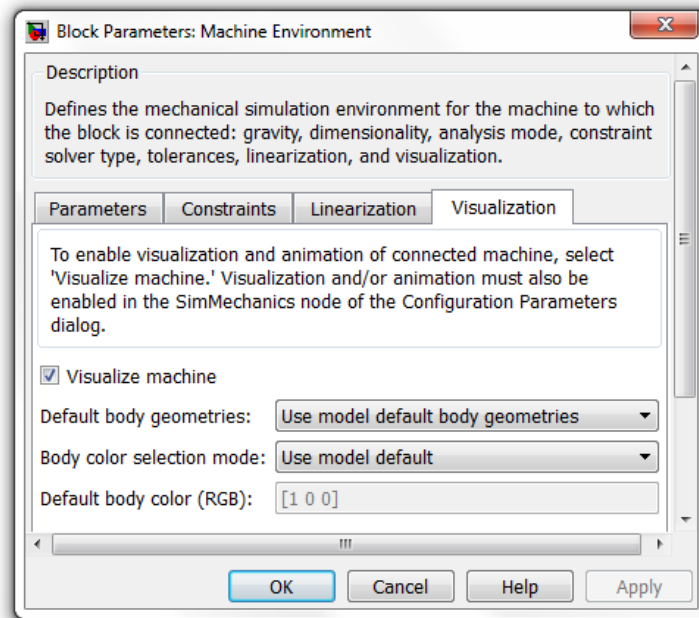


Figura 4.21: Parâmetros da janela *Visualization*.

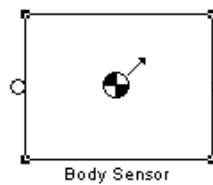


Figura 4.22: Bloco *Body Sensor*.

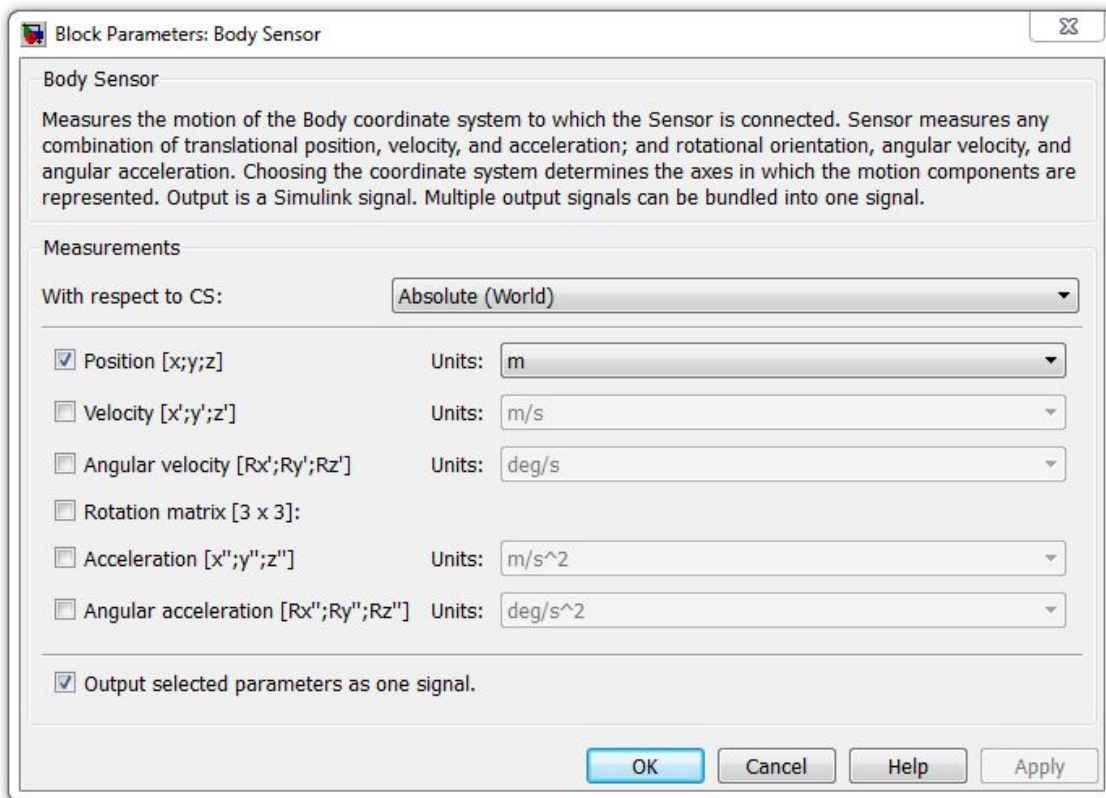


Figura 4.23: Parâmetros do bloco *Body Sensor*.

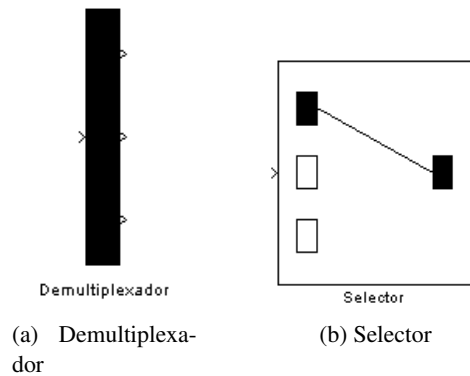


Figura 4.24: Blocos de divisão de sinais de Simulink™.

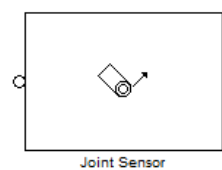


Figura 4.25: Bloco *Joint Sensor*.

acelerações angulares; se ligado a uma junta esférica é possível obter o quaterniões assim como a sua primeira e segunda derivadas.

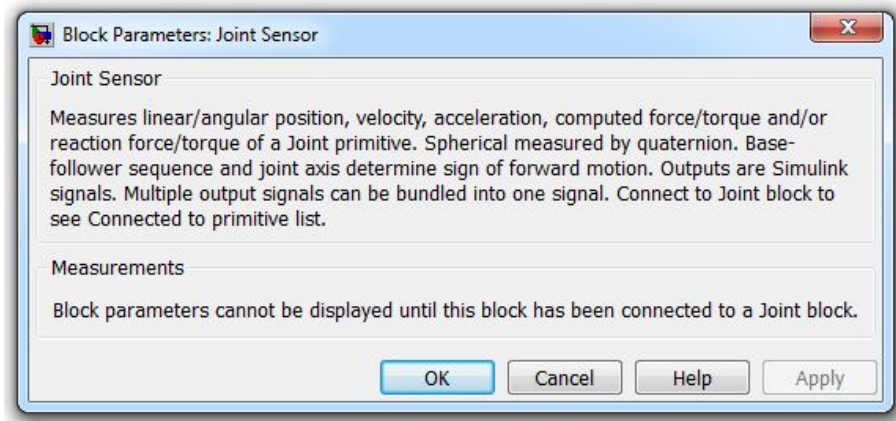


Figura 4.26: Parâmetros do bloco *Joint Sensor*.

Tal como o sensor de corpo, os dados exportados por este bloco (Figura 4.26) podem ser utilizados para construção de gráficos, cálculos internos ou exportados para o ambiente de trabalho do Matlab®.

4.2 Exemplos de sistemas mecânicos modelados em Simmechanics™

4.2.1 Pêndulo Simples

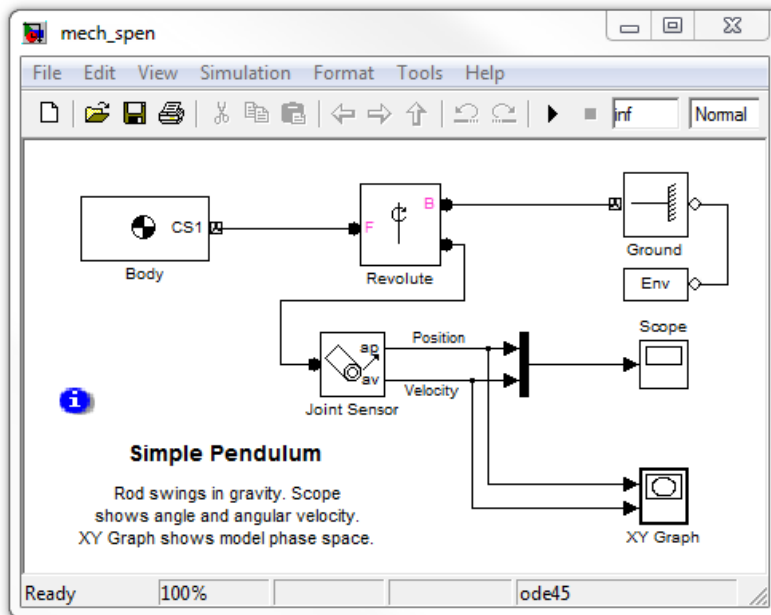
Um exemplo muito interessante e fácil para uma primeira abordagem é o do pêndulo simples (Figura 4.27) (digitar 'mech_spen' na linha de comandos do Matlab®). Este modelo permite compreender o funcionamento dos componentes básicos do Simmechanics™, tais como a introdução da força de gravidade, alterações ao peso e inércia dos modelos e a forma como os diversos componentes se ligam no modelo.

Na Figura 4.28 observam-se os gráficos obtidos após simulação do modelo do pêndulo. No gráfico da esquerda apresenta-se, a cor-de-rosa, a velocidade angular do pêndulo e, a amarelo, a variação angular de posição do pêndulo. O gráfico da direita apresenta a relação entre a posição angular em relação ao eixo dos xx e a velocidade angular em relação ao eixo dos yy , usualmente conhecido como plano de fase.

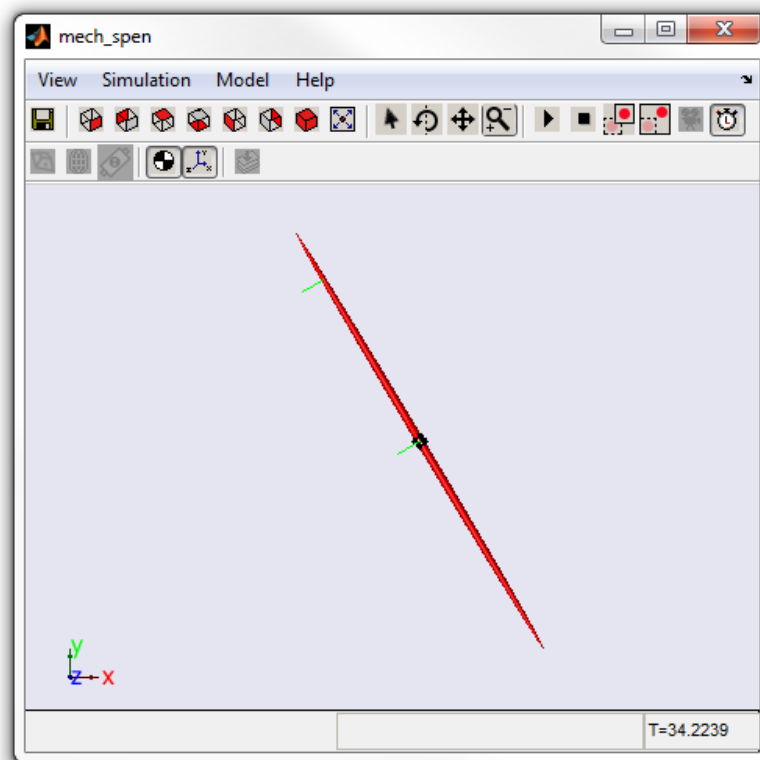
4.2.2 Bouncing ball (Bola saltitante)

Um módulo um pouco mais avançado que o anterior é o modelo da "Bouncing ball" (Figura 4.29) (digitar 'mech_bouncing_ball' na linha de comandos do Matlab®) que simula a queda de uma bola no chão. Neste exemplo pode-se observar a forma de actuação

Simmechanics



(a) Modelo



(b) Representação gráfica

Figura 4.27: Imagens do modelo do pêndulo simples.

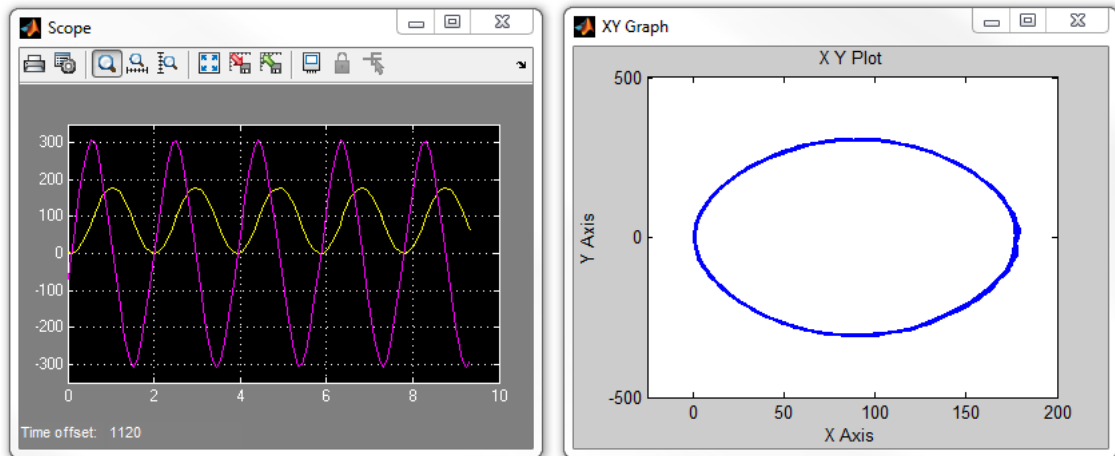


Figura 4.28: Gráficos de análise do movimento do pêndulo.

de forças nos corpos e a implementação de um modelo linear de contacto com o solo do tipo:

$$Fy = k \times y - b \times \dot{y} \quad (4.3)$$

Este modelo (Figura 4.29) não possui nenhuma força a actuar directamente na bola com excepção da força da gravidade, neste caso no sentido negativo do eixo dos zz .

Neste exemplo a bola está ligada ao solo por uma junta do tipo prismática/translacional. O bloco chamado "plane" representa o solo, estando fixo ao ambiente virtual num ponto definido pelo bloco "Ground" (Figura 4.29).

A bola tem o seu CG ligado ao bloco de simulação de contacto com o solo chamado, neste caso, de "Contact Force". Como este modelo pretende somente simular uma bola a cair ao chão, definem-se condições iniciais para estabelecer a altura da qual a bola irá cair. Para esse efeito é utilizado o bloco "Initial Conditions".

Os gráficos da Figura 4.30 apresentam a evolução ao longo do tempo da oscilação da bola. Consegue-se visualizar nestes gráficos o amortecimento da bola no embate no chão e também a força de reacção do solo quando a bola entra em contacto com este.

A azul está representada a posição, velocidade e força de contacto no eixo dos yy e a cor-de-rosa no eixo dos xx . O facto de não haver qualquer tipo de indicação no eixo dos xx justifica-se por se tratar de um modelo unidimensional uma vez que a bola só oscila na direção do eixo dos yy .

Simmechanics

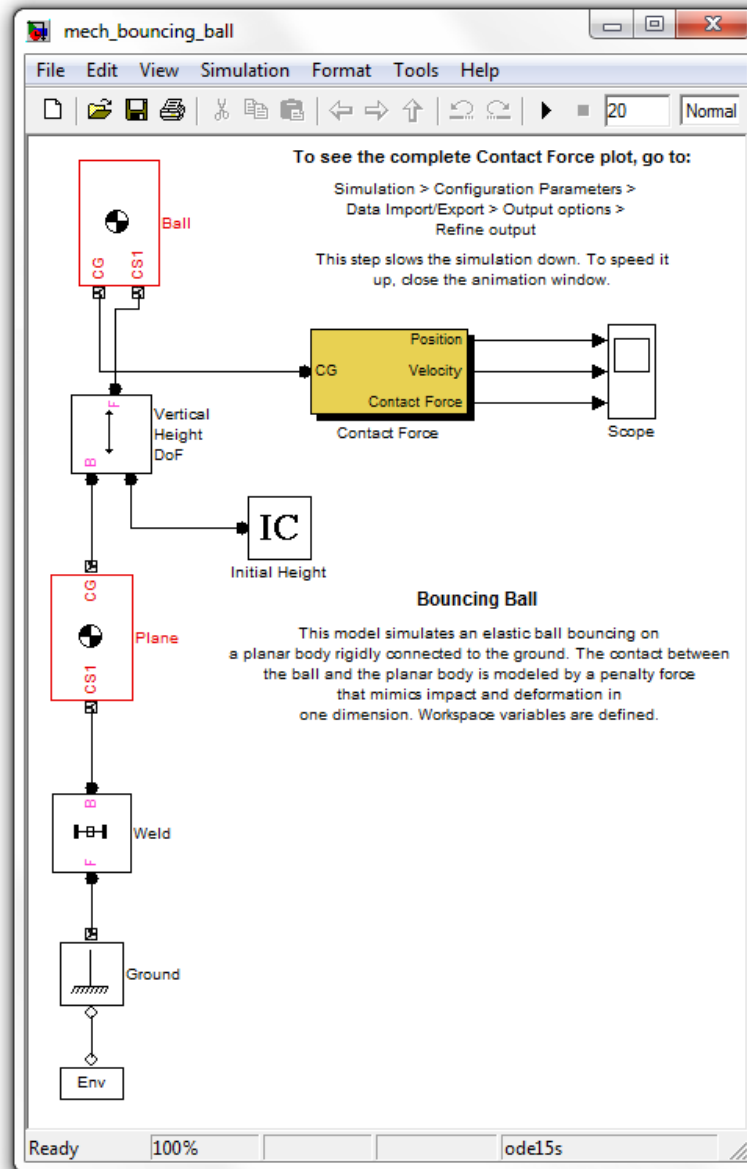


Figura 4.29: Modelo da "Bola Saltitante".

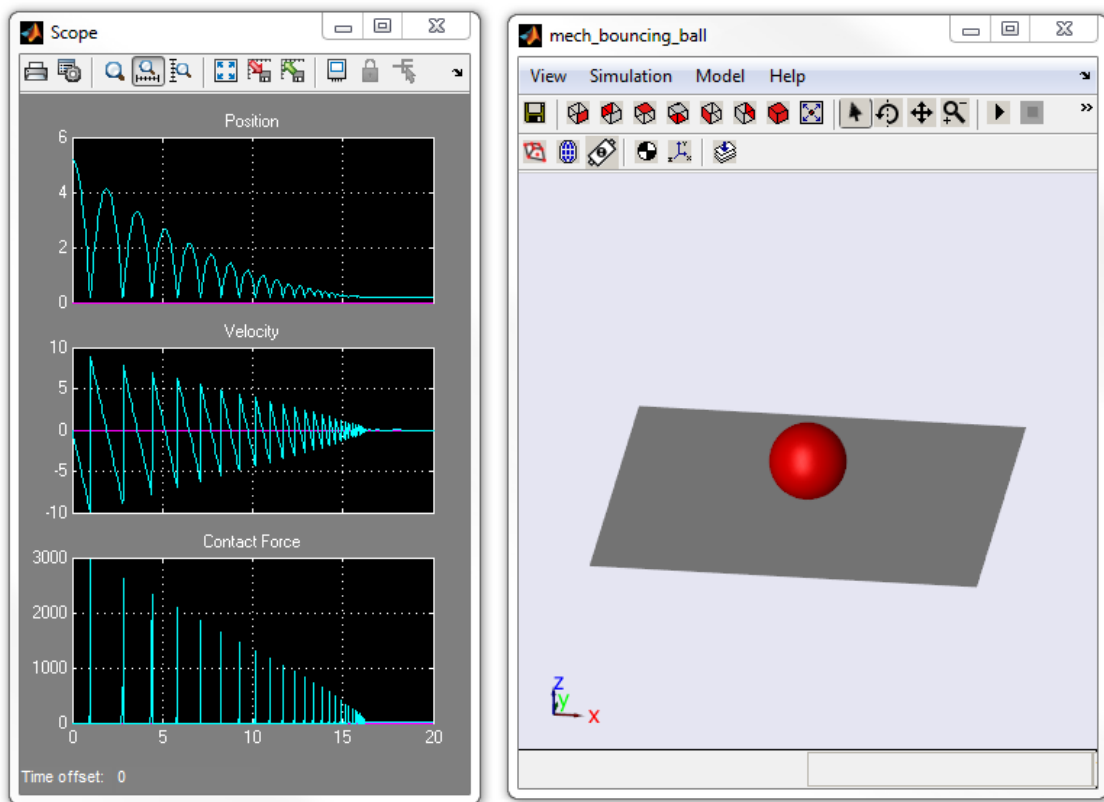


Figura 4.30: Gráficos de análise do movimento da bola saltitante.

4.2.3 Stewart platform (plataforma de Stewart¹)

Por último, enquanto introdução às capacidades do Simmechanics™, aconselha-se a observação do modelo de uma plataforma de Stewart. Este modelo pode ser aberto digitando na linha de comandos do Matlab® 'stewart_demo' (Figura 4.31).

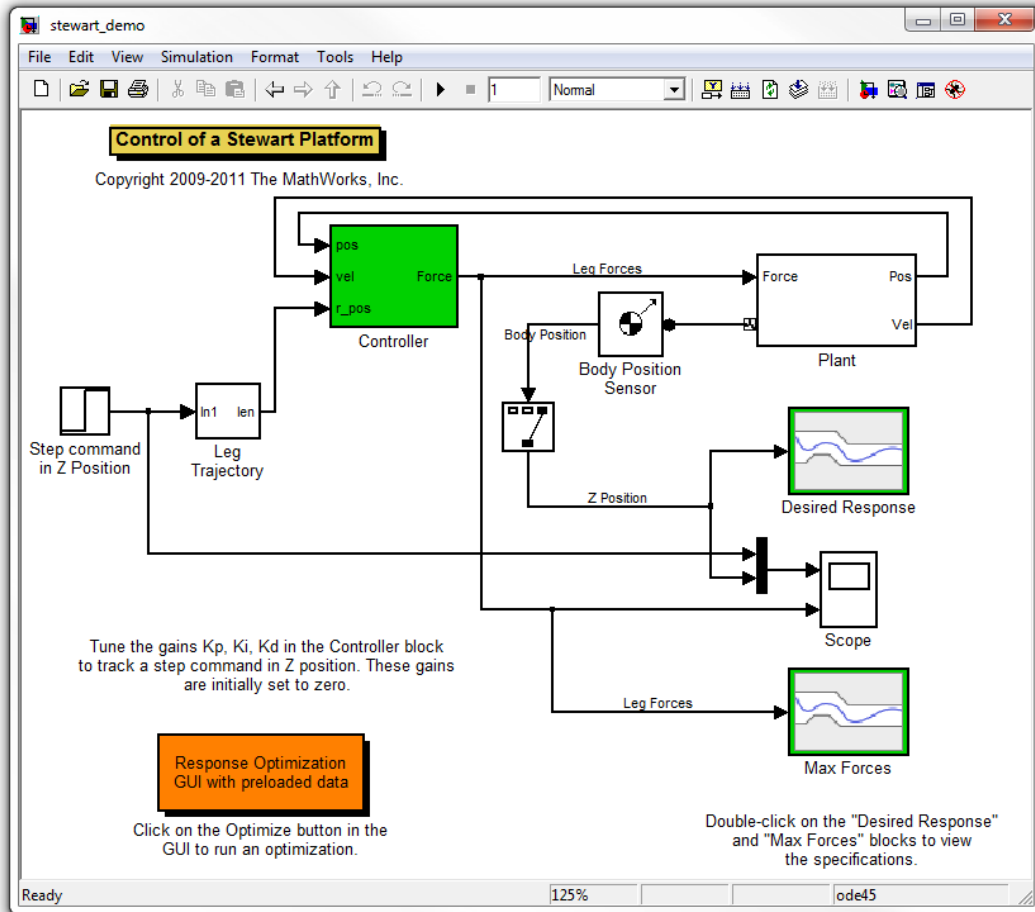


Figura 4.31: Modelo da plataforma de Stewart.

Existem diversas variantes deste modelo na Internet, sendo o modelo apresentado o que vem no pacote original do Simmechanics™ (Figura 4.32).

Neste modelo é utilizado um controlador do tipo Proporcional, Integral e Derivativo (PID) (Figura 4.33) para garantir que a plataforma se comporta como pretendido. Existe um sistema complexo que gera o movimento pretendido para cada uma das "pernas".

Este modelo tem a possibilidade de apresentar uma visualização com recurso a imagens de CAD. Apresenta também inúmeras formas de simular, podendo o utilizador alterar a entrada do sistema, configurar os controladores manualmente ou com recurso a

¹Usualmente conhecida por plataforma de Gough Stewart em homenagem aos seus criadores D. Stewart e Eric Gough. Publicada pela primeira vez por D. Stewart na revista *Proceedings of the Institution of Mechanical Engineers*, Volume 180 (Pt 1, No 15)

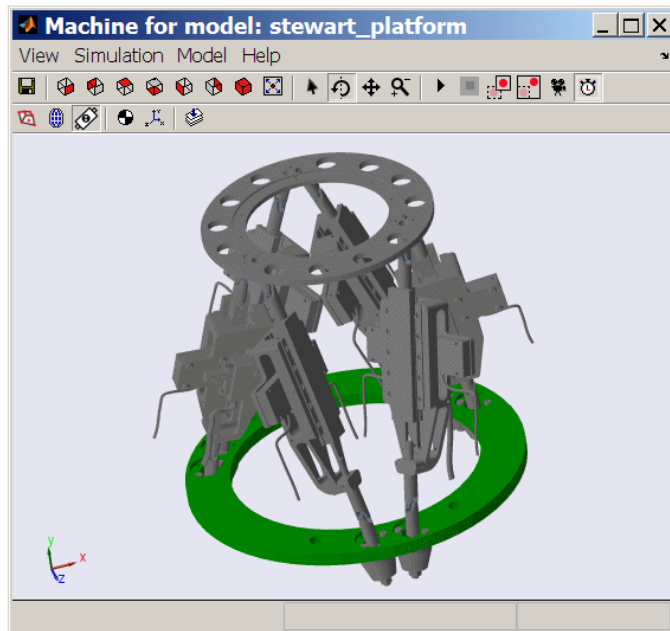


Figura 4.32: Representação gráfica da plataforma.

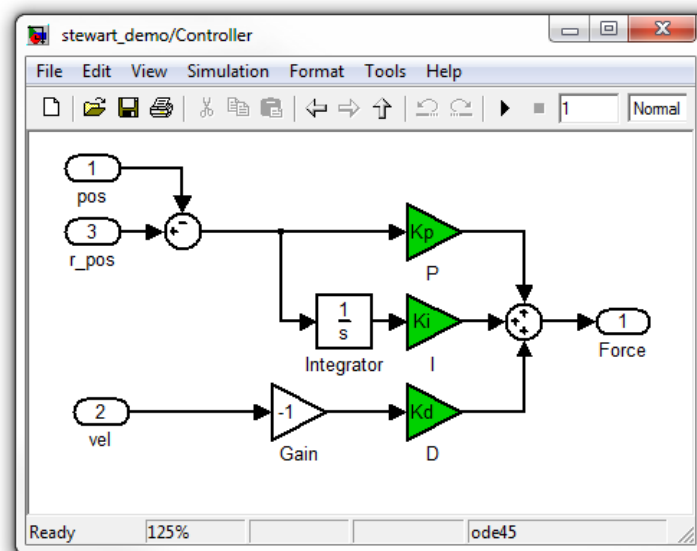


Figura 4.33: Controlador utilizado na plataforma de Stewart.

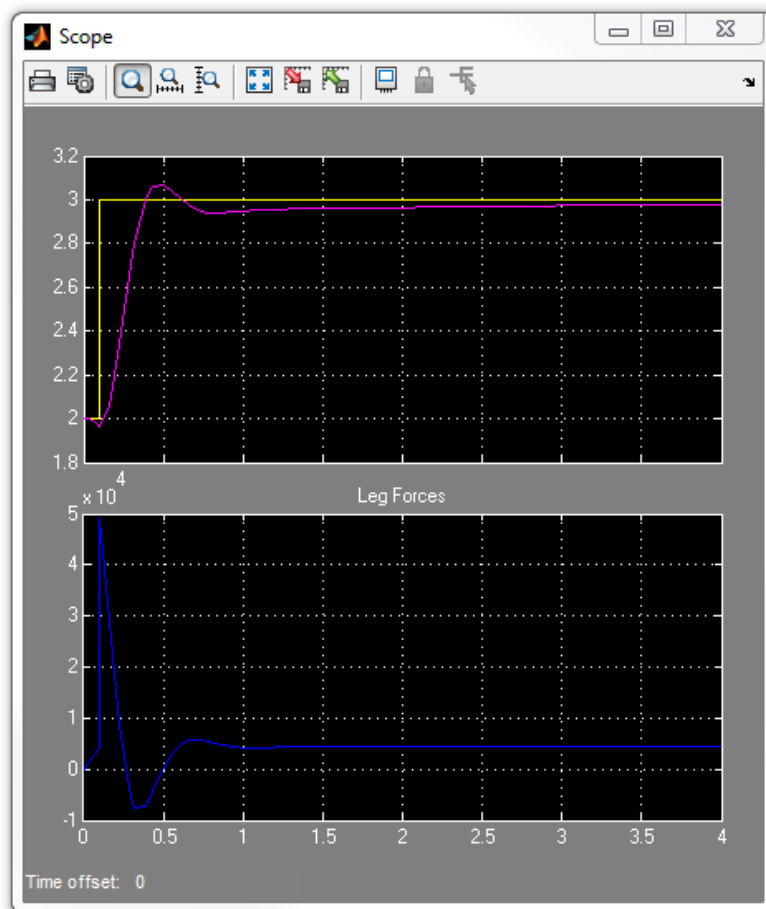


Figura 4.34: Gráfico obtido pela simulação do modelo da plataforma de Stewart.

um modo de optimização automático. Na Figura 4.34 apresentam-se os resultados da resposta da plataforma a um degrau de amplitude 3. Na parte superior do gráfico observa-se o degrau pretendido a amarelo e a resposta do sistema em cor-de-rosa. Na parte inferior da mesma figura pode-se verificar o valor da força aplicada na perna de forma a deslocar a plataforma para a posição pretendida.

Capítulo 5

Trabalho desenvolvido

Neste capítulo são explicados todos os passos necessários para a modelação de um robô quadrúpede, tal como descrito no Capítulo 3. No final do capítulo são apresentados alguns testes e resultados obtidos de forma a demonstrar a validade deste modelo.

5.1 Modelação do robô quadrúpede

Nesta secção explica-se, passo a passo, como foi implementado o modelo do robô, sendo incluídas diversas imagens para melhor ilustrar os detalhes da configuração. Uma correcta definição do robô é essencial para o seu correcto funcionamento. É muito importante, antes de começar a introduzir os blocos necessários e interligá-los, ter uma ideia muito clara de quais os blocos que serão necessários e como eles irão interagir. Um bom procedimento é fazer um esquema em papel do modelo do robô, onde se devem representar todos os corpos e juntas, indicando sempre o tipo de junta e o número de graus de liberdade que irá acrescentar ao robô.

5.1.1 Modelação do corpo do robô

O corpo do robô é constituído por quatro blocos iguais, interligados por uma mola e um amortecedor dispostos em paralelo de forma a simular uma coluna vertebral, tornando o caminhar do robô mais suave e amortecendo os contactos mais bruscos (Figura 5.1).

Cada bloco tem uma massa de 9 kg e um tamanho de 0,3 m por 0,2 m com 0,15 m de altura (consultar anexo A). A massa dos diversos blocos é calculada com base no seu volume multiplicado pela densidade do material utilizado, 2.800 kg/m^3 para as pernas e 1.000 kg/m^3 . Através do ficheiro de parametrização que acompanha o robô é possível alterar tanto as dimensões como a sua densidade, de forma a verificar a sua influência na simulação.

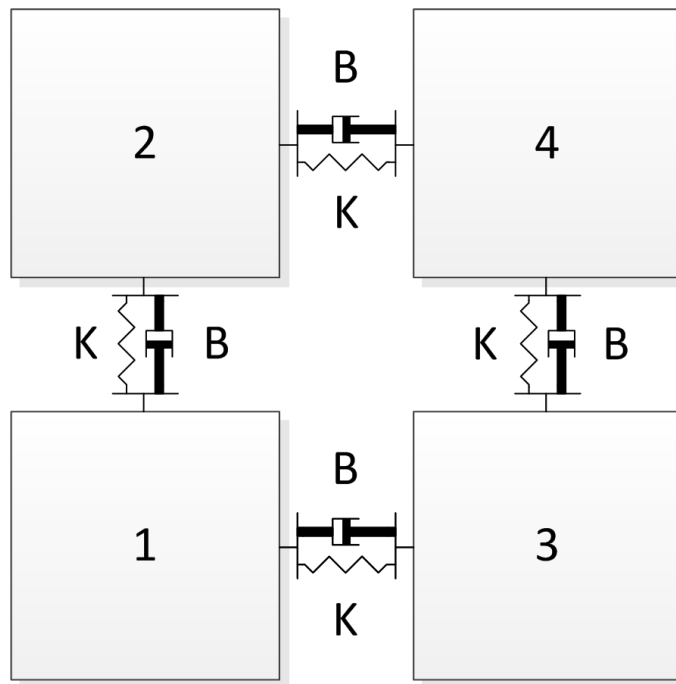


Figura 5.1: Esquema do corpo do robô.

O corpo é modelado no Simmechanics™ pela utilização de quatro blocos de corpo (*Body Blocks*). Estes blocos são interligados por uma junta do tipo universal actuada por um bloco do tipo "*Joint Spring and Damper*" (Figura 5.2).

Como apresentado na Figura 5.3, no bloco de corpo é preciso introduzir a sua massa, a sua matriz de inércia e os diversos SC que irão permitir ligar a estes blocos as pernas e também as juntas que ligam os quatro blocos constituintes do corpo do robô.

Foi criada uma variável que contém o valor da massa do corpo: essa variável é m_{body} e é definida pela expressão (5.1). As variáveis h_{body} , d_{body} e w_{body} correspondem respectivamente à altura, profundidade e largura de cada bloco do corpo. A variável ρ_{body} representa a densidade do material utilizado na modelação deste robô.

$$m_{body} = \rho_{body} \times (h_{body} \times d_{body} \times w_{body}) \quad (5.1)$$

Em seguida deve ser introduzida a matriz de Inércia. Esta irá permitir ao programa calcular o efeito que as diversas forças/binários têm no corpo. No Matlab® a matriz de inércia de um determinado corpo é sempre definida em relação ao seu centro de massa.

Os blocos constituintes do corpo possuem uma matriz de inércia da forma apresentada na equação (5.2).

$$inertia_{body} = \frac{1}{12} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \quad (5.2)$$

Trabalho desenvolvido

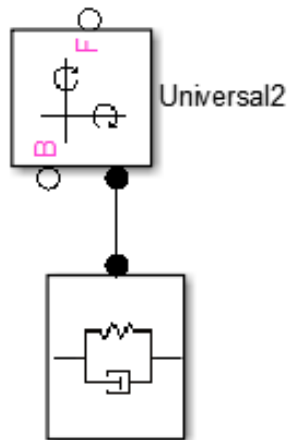


Figura 5.2: Sistema mola-amortecedor do corpo.

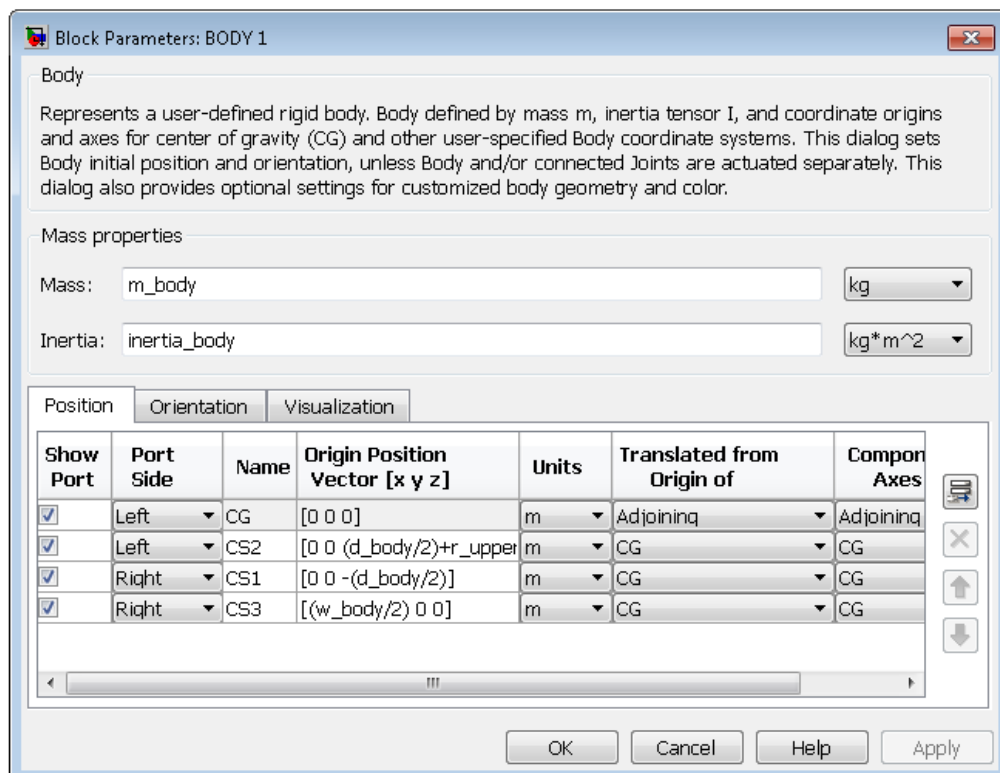


Figura 5.3: Parametrização do bloco 1 do corpo.

$$a_{11} = m_body \times (h_body^2 + d_body^2)$$

$$a_{22} = m_body \times (w_body^2 + d_body^2)$$

$$a_{33} = m_body \times (w_body^2 + h_body^2)$$

Uma vez mais é utilizada uma variável para facilitar a alteração das diversas características do robô a partir de um só ficheiro.

Nesta fase é necessário unir os diversos blocos e definir os SC para que o corpo assuma a forma previamente estabelecida. Um SC, tal como explicado na subsecção 4.1.1, é nada mais do que um ponto dentro ou fora de um corpo, no qual se definem os eixos xx , yy e zz com características que poderão ser diferentes das do SC do "mundo". Estes SC são utilizados como referência para ligar uma junta ou actuador/sensor de corpo.

No Simmechanics™ existem diversos métodos para definir as posições de uns corpos em relação aos outros. Esta referenciação pode ser absoluta, quando é feita em relação à origem do "mundo", ou relativa, quando é feita em relação a um outro elemento constituinte do sistema. Qualquer que seja o sistema de referenciação dos diversos elementos do robô, é muito importante manter uma certa coerência na sua definição. Quando se dá início à simulação, o Matlab® verifica se todos os corpos e juntas estão dentro de determinados limites de tolerância ("*Assembly Tolerances*"). Dependendo do modelo que se pretende simular, estes limites devem ser ajustados no menu *parameters* do bloco "*Machine Environment*". É importante ter em atenção que, no caso da referenciação relativa, é sempre necessário colocar um dos blocos ligado ao bloco "*Ground*" cuja referenciação terá que ser obrigatoriamente absoluta.

No robô apresentado nesta dissertação utilizou-se referenciação relativa para todo o robô. Utilizando uma junta do tipo "*6 Degrees Of Freedom*" foi conectado um bloco de "*Ground*" num dos SC do bloco 1 do corpo; esta montagem permite ao robô movimentar-se no espaço tridimensional. Em seguida foi definida a posição dos restantes blocos do corpo com base na opção "*adjoining*" dos SC. A opção "*adjoining*" indica ao SC que a sua posição está referenciada ao SC ao qual se encontra conectado no bloco adjacente. Desta forma simplifica-se muito o sistema sendo necessários menos blocos e reduzindo a probabilidade de erro ao definir a posição de algum bloco. Estabeleceu-se que os blocos 2 e 3 se referenciam com base no bloco 1 e que o bloco 4 vai buscar a referência ao bloco 2 (Figura 5.4).

5.1.2 Modelação das pernas do robô

As pernas do robô são constituídas por duas partes, sendo ambas representadas neste modelo por cilindros rígidos. No ficheiro de configuração do sistema é possível definir diferentes raios e alturas para os elementos das pernas permitindo alguma personalização. Também é possível definir pernas em diferentes materiais bastando, para esse efeito, alterar o valor da densidade para a do material pretendido. As variáveis que foram definidas,

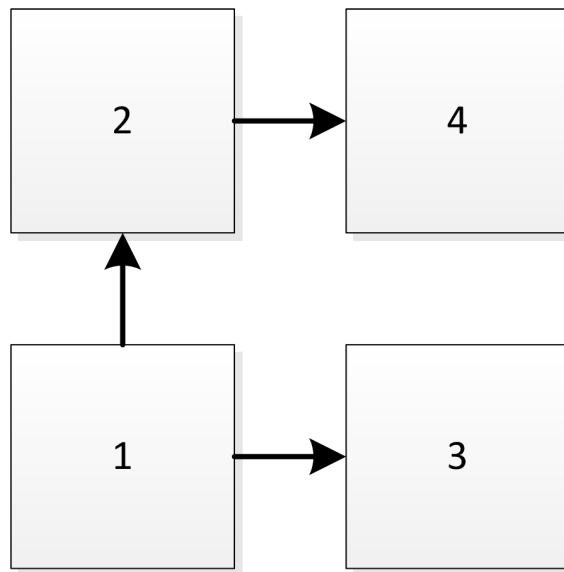


Figura 5.4: Referência dos blocos do corpo.

e cujos valores podem ser alterados no ficheiro de Matlab[®] de configuração do sistema, são respectivamente o comprimento, o raio e a densidade, tanto da parte superior como da parte inferior de cada uma das pernas. As variáveis são as apresentadas na Tabela 5.1.

Tabela 5.1: Variáveis de configuração das pernas.

Segmento	Comprimento	Densidade	Massa	Raio
Perna 1 superior	$h_{upper_leg_1}$	ρ	$m_{upper_leg_1}$	$r_{upper_leg_1}$
Perna 1 inferior	$h_{lower_leg_1}$	ρ	$m_{lower_leg_1}$	$r_{lower_leg_1}$
Perna 2 superior	$h_{upper_leg_2}$	ρ	$m_{upper_leg_2}$	$r_{upper_leg_2}$
Perna 2 inferior	$h_{lower_leg_2}$	ρ	$m_{lower_leg_2}$	$r_{lower_leg_2}$
Perna 3 superior	$h_{upper_leg_3}$	ρ	$m_{upper_leg_3}$	$r_{upper_leg_3}$
Perna 3 inferior	$h_{lower_leg_3}$	ρ	$m_{lower_leg_3}$	$r_{lower_leg_3}$
Perna 4 superior	$h_{upper_leg_4}$	ρ	$m_{upper_leg_4}$	$r_{upper_leg_4}$
Perna 4 inferior	$h_{lower_leg_4}$	ρ	$m_{lower_leg_4}$	$r_{lower_leg_4}$

As pernas possuem duas juntas rotativas com um grau de liberdade cada. A junta superior pretende simular a rotação da anca e a junta inferior pretende simular o joelho. Na Figura 5.5 apresenta-se o modelo de Simmechanics[™] utilizado para modelar as pernas do robô.

De forma a simplificar a definição da matriz de inércia das duas partes que constituem os membros optou-se por defini-los inicialmente como alinhados no eixo dos xx . As suas

Trabalho desenvolvido

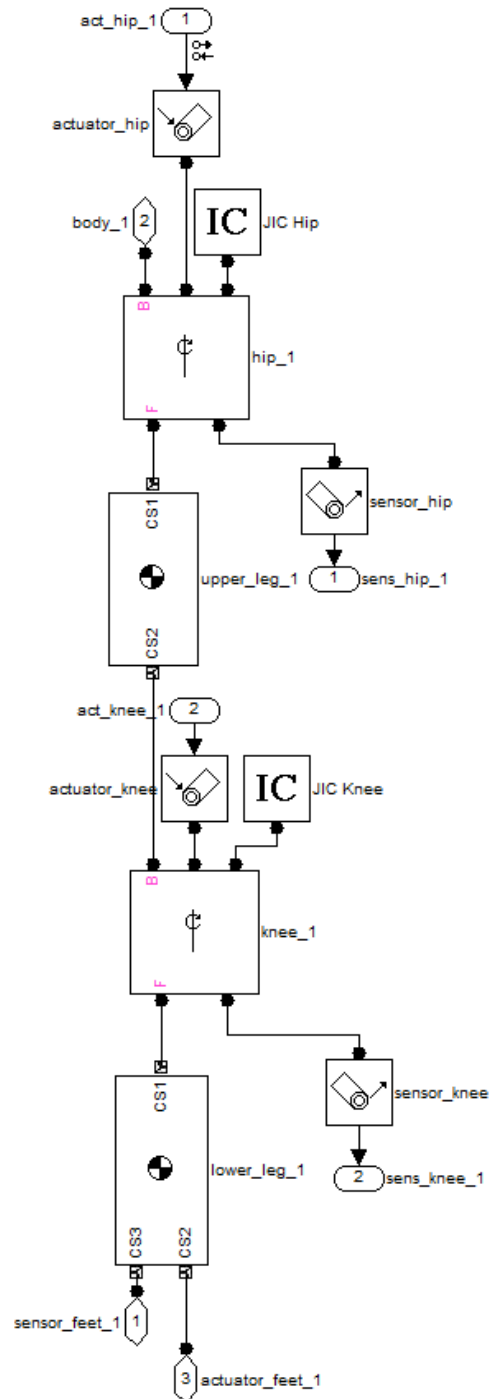


Figura 5.5: Modelo de uma perna do robô.

matrizes de inércia são definidas pela equação 5.3.

$$I_{perna} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (5.3)$$

$$I_{xx} = \frac{1}{2}mr^2$$

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

Na equação 5.3 o parâmetro L corresponde ao comprimento do segmento e o r corresponde ao raio do cilindro.

O bloco com o nome *body_1* está conectado de um lado ao SC do bloco 1 do corpo e do outro lado, como se pode ver na Figura 5.5, a uma junta do tipo rotacional. Esta junta (*hip_1*) modela a anca do robô servindo como ligação entre o corpo e a perna. A sua configuração apresenta-se na Figura 5.6.

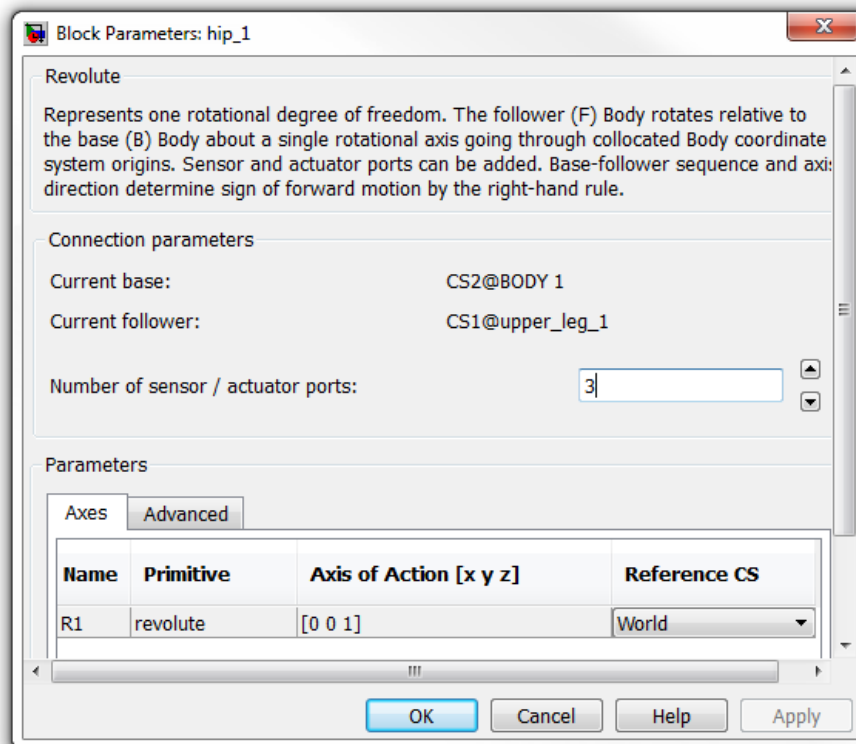


Figura 5.6: Bloco junta rotacional da anca do robô.

A rotação da anca é efectuada em torno do eixo dos zz , como se pode observar na Figura 5.6 no campo "Axis of Action [x y z]".

A parte superior da perna está representada na Figura 5.5 com o nome "*upper_leg_1*". O SC da perna superior que liga a perna ao corpo é definido com base no SC do corpo.

Tanto o CG como o SC2 da perna superior são definidos com base no SC1, como se pode observar na Figura 5.7.

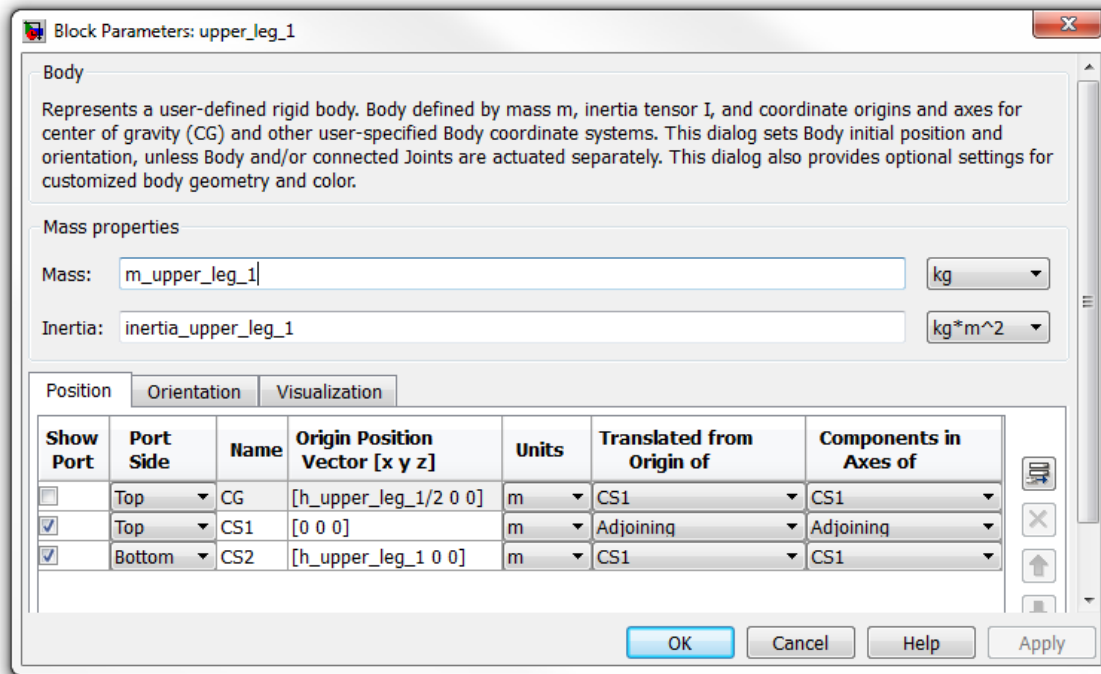


Figura 5.7: Configuração da perna superior.

A configuração da junta rotacional do joelho é igual à da anca, estando este bloco identificado na Figura 5.5 com o nome "*knee_1*". A parte inferior da perna é muito semelhante à superior na forma como é definido o seu CG. Para além do SC utilizado para ligar a perna inferior à perna superior também têm que ser definidos dois SC no seu extremo inferior. Estes SC (CS2 e CS3) permitirão ao modelo de solo "ler" a posição e velocidade do pé e também actuá-lo com as forças de reacção do solo, como se pode verificar pela análise da Figura 5.8.

A actuação dos membros é efectuada directamente nas juntas utilizando-se para o efeito actuadores de juntas. Os controladores PD estão ligados a estes actuadores permitindo o controlo do movimento das juntas. Também existem sensores ligados a estas juntas das pernas. Estes sensores recolhem o valor da posição real da junta permitindo aos controladores em seguida calcular o erro existente entre a posição desejada e a posição real e desta forma ajustar o sinal de controlo.

Os blocos JIC permitem a introdução das condições iniciais de simulação. A introdução destes valores é de extrema importância. Caso isso não aconteça, na transição da posição na qual o robô foi definido (com as pernas alinhadas com o eixo dos *xx*) para a primeira posição desejada, o erro detectado pelos controladores é muito elevado dando origem a uma resposta incorrecta dos controladores. As condições iniciais introduzidas

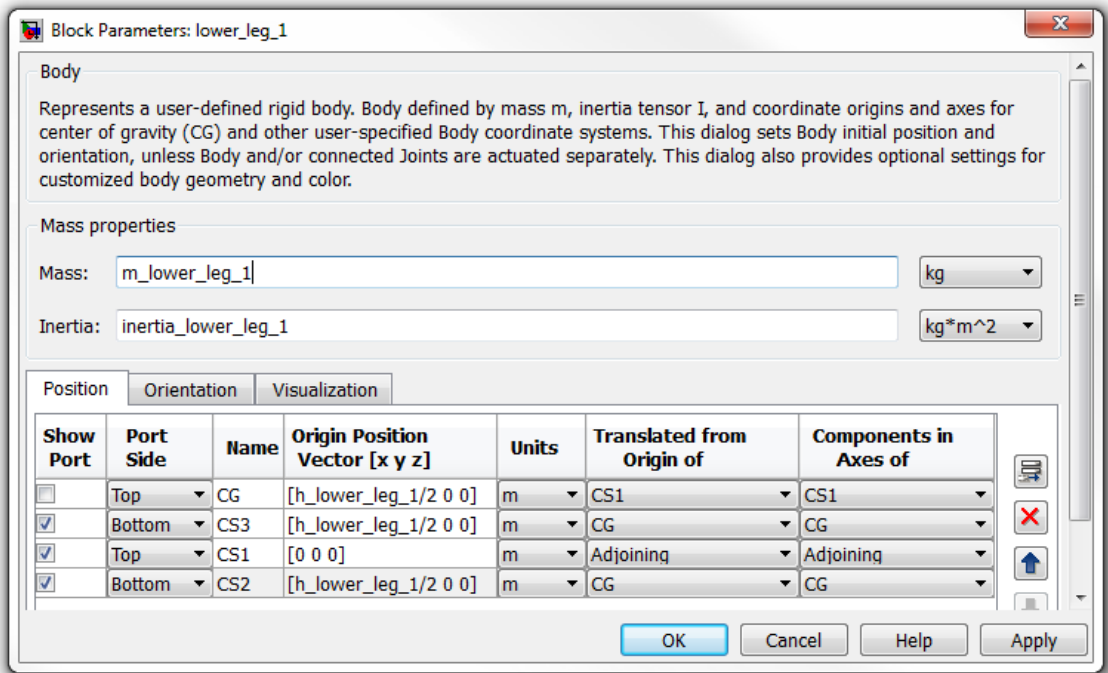


Figura 5.8: Configuração da perna inferior.

no bloco JIC correspondem à primeira posição dos vectores de posição da anca e do joelho $hip_1(1,2)$ e $knee_1(1,2)$.

5.2 Sistema de controlo das juntas das ancas e dos joelhos

O controlador escolhido para o controlo das juntas das ancas e dos joelhos do robô é do tipo proporcional e derivativo (PD). A sua implementação foi feita através do bloco PID, parte integrante da ferramenta Simulink™. Os blocos de controlo, como o apresentado na Figura 5.9, recebem dois sinais de entrada e colocam um sinal na saída.

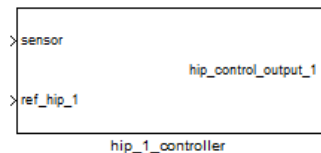


Figura 5.9: Bloco do controlador.

Os sinais de entrada são o sinal de referência e o sinal de erro. Estes sinais são posteriormente subtraídos de forma a calcular o sinal de erro (Figura 5.10).

Com o sinal de erro o controlador determina a saída necessária para corrigir a posição da junta.

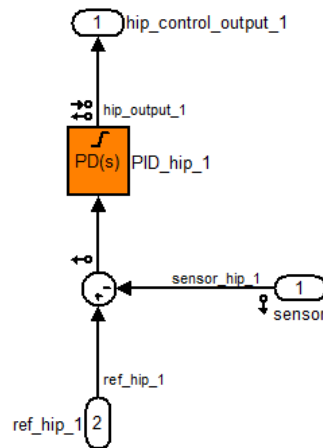


Figura 5.10: Controlador PD.

5.3 Modelo de contacto pé-solo

Na secção 3.4 foi referida e explicada a forma utilizada para modelar o contacto do pé com o solo. Nesta secção explica-se como é que esse modelo teórico foi implementado em SimmechanicsTM.

Definiu-se que o contacto com o solo se daria somente num ponto na extremidade inferior da perna.

Nesse ponto foram criados dois SC, utilizados para conectar o sensor que permite ao modelo de solo saber, em todas as amostras, a posição, velocidade e aceleração do pé e também para conectar o actuador que permite ao modelo do solo aplicar a força necessária ao robô, simulando o contacto da perna com o solo.

Como o programa SimmechanicsTM não detecta colisões entre objectos, foi necessário definir uma condição que permita ao sistema saber em que instante é que o corpo entra em contacto com o solo.

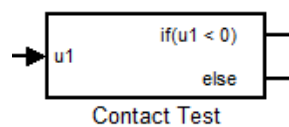


Figura 5.11: Bloco de verificação de contacto com o solo.

Optou-se pela utilização de um bloco "if" do SimulinkTM ao qual foi dado o nome de *Contact Test* (Figura 5.11). No modelo apresentado nesta dissertação considerou-se o solo como sendo o plano composto pelos eixos de coordenadas xz situado na coordenada 0 do eixo dos y . Este bloco verifica se a extremidade inferior da perna se encontra numa posição em y inferior a 0, executando a condição $u1 < 0$. A configuração deste bloco

(Figura 5.12) passa pela definição do número de entradas, neste caso somente uma, e da condição de verificação.

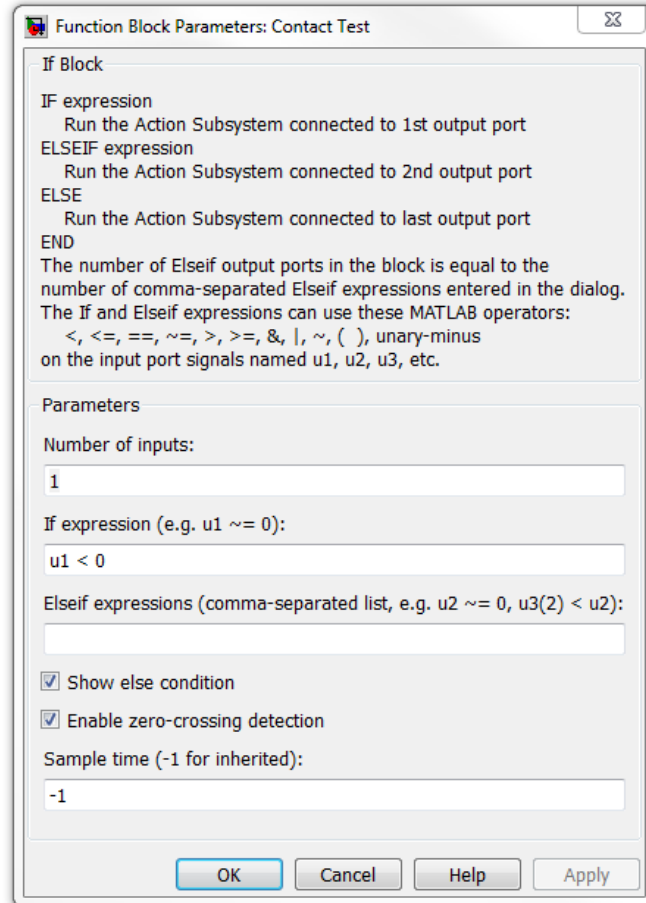


Figura 5.12: Detalhes da configuração do bloco de verificação do contacto com o solo.

Após o sistema detectar que o pé se encontra em contacto com o solo, são activados dois subsistemas que irão calcular as forças de contacto tangenciais e normais.

As forças resultantes desses cálculos são posteriormente juntas em um só sinal utilizando como valor de força no eixo dos z o valor zero.

5.3.1 Força tangencial (F_x)

Como referido na secção 3.4, o modelo matemático utilizado para modelar o contacto entre o pé e o solo está dividido em duas forças. A força tangencial permite ao robô avançar no terreno. Esta força é calculada com base na expressão (5.4).

$$F_x = -K_x(x - x_0) - B_x\dot{x} \quad (5.4)$$

A força tangencial é determinada pelo modelo representado na Figura 5.13. Este modelo recebe como parâmetros de entrada a posição e a velocidade do pé. Em seguida esses vectores são divididos e existe um bloco responsável por determinar se o pé já se encontra em contacto com o solo ou não. Se o pé ainda não se encontrar em contacto com o solo a força colocada na saída deste bloco é zero. Caso o pé se encontre em contacto com o solo é activado o bloco representado na Figura 5.13 por "*Friction Force*".

Um dos parâmetros da fórmula de cálculo da força de contacto tangencial é a posição inicial de contacto do pé com o solo no eixo dos x . Esta posição deve ser memorizada durante todo o período de contacto do pé com o solo para que seja possível calcular a diferença entre a posição de contacto e a posição actual. O bloco responsável por esta função é o bloco "*Store Initial State*". Este bloco só está activo enquanto o valor colocado na sua porta de "*Enable*" (entrada na parte superior do bloco) é positiva. Quando essa entrada passa a negativa o bloco fica imediatamente desactivado e mantém na sua saída a última posição registada na sua porta de entrada. Aproveitou-se essa característica deste bloco e introduziu-se na sua porta de controlo o sinal de posição de y ; na entrada desde subsistema (porta do lado esquerdo do bloco) foi colocada a posição em x . Quando o pé se encontra fora do solo este subsistema coloca na saída o mesmo valor que lhe é colocado na entrada. Quando o pé entra em contacto com o solo, o sinal de controlo passa a negativo, e o sistema fica desactivado. Neste estado o subsistema coloca na porta de saída (porta do lado direito do bloco) a última posição válida mantendo esse valor até o seu sinal de controlo voltar a ser superior a zero.

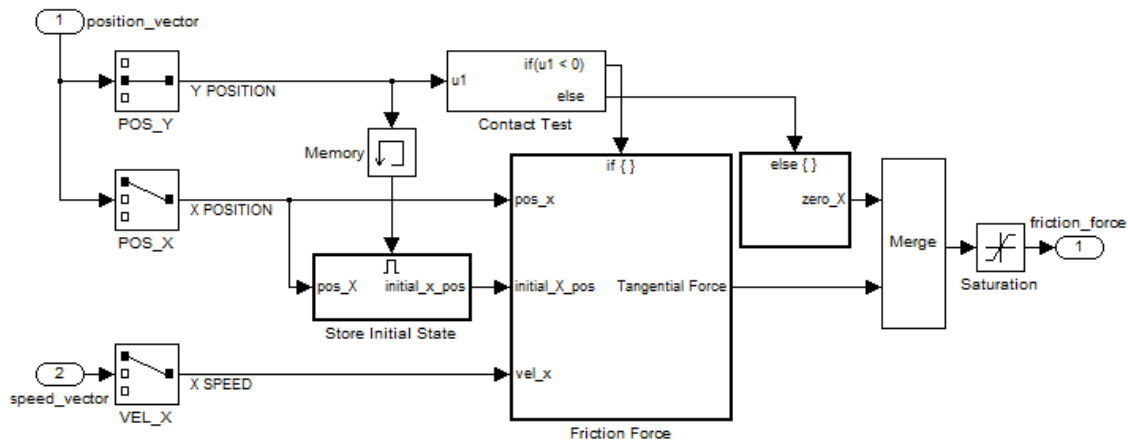


Figura 5.13: Esquema de blocos de cálculo da força de contacto tangencial.

Dentro do subsistema "*Friction Force*" encontram-se os blocos representados na Figura 5.14. Estes blocos servem como entradas de variáveis para o coeficiente de elasticidade da mola e para o coeficiente de atrito viscoso do amortecedor. É também nesta fase que é calculada a diferença entre a posição actual do pé em x e a posição de contacto do pé com o chão no eixo dos x .

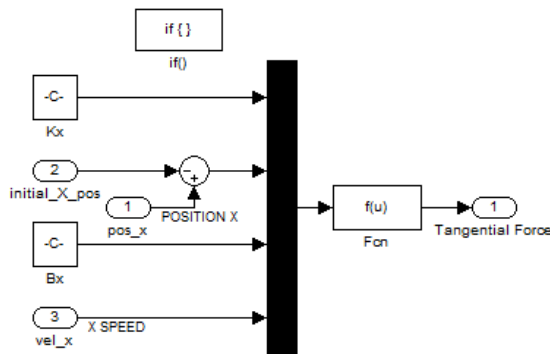


Figura 5.14: Bloco de cálculo da força de contacto tangencial.

Em seguida é utilizado o bloco "*Fcn*" (Figura 5.15) do Simulink™ que calcula o resultado de uma determinada expressão introduzida pelo utilizador.

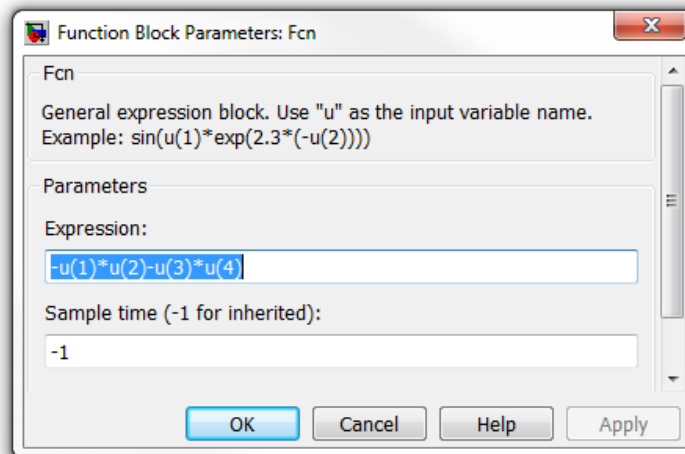


Figura 5.15: Bloco "*Fcn*" de cálculo da força tangencial.

Na saída deste bloco o sinal de força tangencial passa por um bloco de saturação que permite ao utilizador limitar os valores de força do solo. Estes valores podem ser definidos no ficheiro de configuração do sistema através das variáveis "*upper_friction_saturation*" e "*lower_friction_saturation*". Na configuração padrão apresentada no anexo A estes parâmetros são definidos como, respectivamente, *inf* e *-inf*. Considerou-se que esta saturação poderá ser interessante para, se necessário, limitar a força tangencial exercida pelo solo no pé do robô.

5.3.2 Força normal (F_y)

A força normal garante que durante a simulação o robô se irá manter sempre acima do solo. Esta força é calculada com base na expressão (5.5).

$$F_y = -K_y y - B_y (-y)^m \dot{y} \quad (5.5)$$

A força normal é determinada pelo modelo representado na Figura 5.16. Este modelo recebe como parâmetros de entrada a posição e a velocidade do pé. Em seguida esses vectores são divididos e, tal como acontece no subsistema de cálculo da força tangencial, existe um bloco responsável por determinar se o pé já se encontra em contacto com o solo ou não. Se o pé ainda não se encontrar em contacto com o solo a força colocada na saída deste bloco é zero. Caso o pé se encontre em contacto com o solo é activado o bloco representado na Figura 5.16 por "Normal Force".

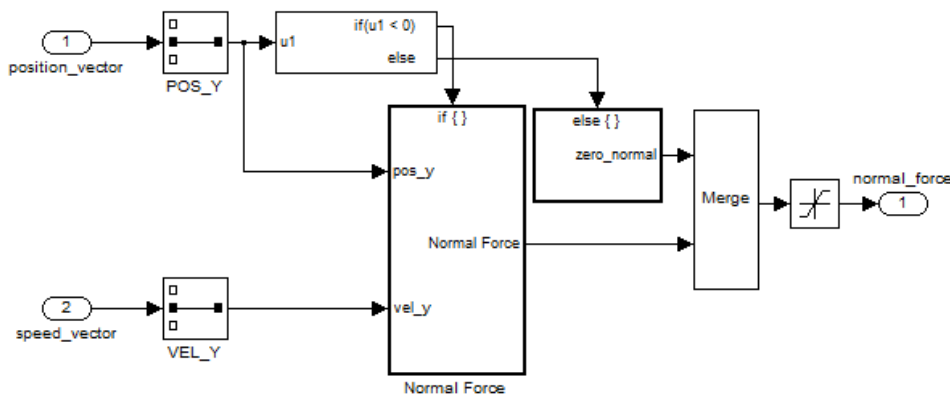


Figura 5.16: Esquema de blocos de cálculo da força de contacto normal.

Dentro do subsistema "Normal Force" encontram-se os blocos representados na Figura 5.17. Estes blocos servem como entradas de variáveis para o coeficiente de elasticidade da mola e para o coeficiente de atrito viscoso do amortecedor.

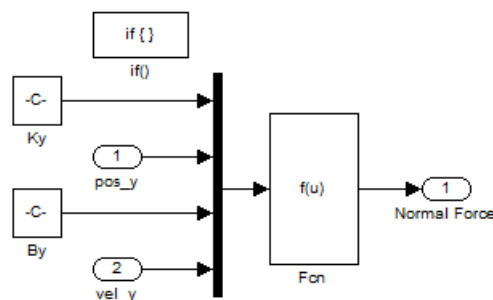


Figura 5.17: Bloco de cálculo da força de contacto normal.

Em seguida é utilizado o bloco "Fcn" (Figura 5.18) do Simulink™ que calcula o resultado de uma determinada expressão introduzida pelo utilizador.

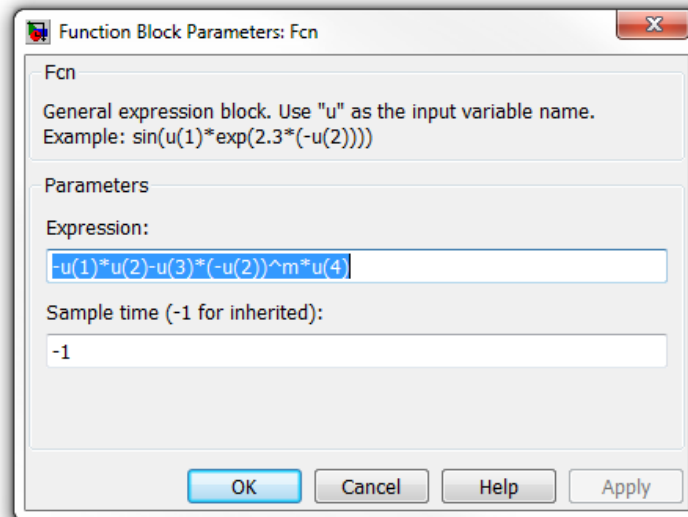


Figura 5.18: Bloco "Fcn" de cálculo da força normal.

Na saída deste bloco o sinal de força normal passa por um bloco de saturação que permite ao utilizador limitar os valores de força do solo. Estes valores podem ser definidos no ficheiro de configuração do sistema através das variáveis "*upper_normal_saturation*" e "*lower_normal_saturation*". Na configuração padrão disponível no anexo A o limite inferior desta saturação é definido como 0 e o valor superior da saturação como infinito (inf). Esta implementação garante que a força exercida pelo modelo de solo no pé do robô, em instante algum será negativa.

5.4 Looptune - Sintonização de controladores

Depois de montado o robô foi necessário sintonizar os controladores, tentou-se sintonizar os controladores utilizando o método de sintonização automático, parte integrante do bloco de PID do Simulink™. Acontece que esse método de sintonização "vê" cada controlador de forma isolada. Devido à complexidade deste sistema os controladores não podem ser sintonizados isoladamente porque a actuação de um controlador influencia directamente o funcionamento do seguinte. Nesta altura encontrou-se uma função do Simulink™ que, de acordo com as especificações apresentadas na sua documentação, permitiria sintonizar os ganhos dos controladores exactamente como pretendido.

O looptune [Matd] é uma função do Simulink™ que permite a sintonização de controladores PID em modelos de sistemas complexos nos quais a sintonização manual dos controladores PID se torna muito complexa. Com base em manuais disponíveis na página

de Internet da ferramenta foi construído um código para sintonização dos controladores das ancas e dos joelhos do robô. O código utilizado foi o seguinte:

```
open_system('robo_v7');
ST0 = slTunable('robo_v7',{ 'PID_hip_1', 'PID_knee_1', 'PID_hip_2', 'PID_knee_2', 'PID_hip_3',
    'PID_knee_3', 'PID_hip_4', 'PID_knee_4'});
addControl(ST0, 'actuator_system');
addMeasurement(ST0, 'sensor_system');
addIO(ST0, { 'ref_hip_1', 'ref_knee_1', 'ref_hip_2', 'ref_knee_2', 'ref_hip_3', 'ref_knee_3',
    'ref_hip_4', 'ref_knee_4'}, 'in');
addIO(ST0, { 'sensor_hip_1', 'sensor_knee_1', 'sensor_hip_2', 'sensor_knee_2', 'sensor_hip_3',
    'sensor_knee_3', 'sensor_hip_4', 'sensor_knee_4'}, 'inout');
rng('default');
Options = looptuneOptions('Display', 'off', 'RandomStart', 5, 'MaxIter', 5);
TrackReq = TuningGoal.Tracking({ 'ref_hip_1', 'ref_knee_1', 'ref_hip_2', 'ref_knee_2',
    'ref_hip_3', 'ref_knee_3', 'ref_hip_4', 'ref_knee_4'}, { 'sensor_hip_1', 'sensor_knee_1',
    'sensor_hip_2', 'sensor_knee_2', 'sensor_hip_3', 'sensor_knee_3', 'sensor_hip_4',
    'sensor_knee_4'}, 1);
[ST, gam, Info] = looptune(ST0, [1000, 1200], TrackReq, Options);
T = ST.getIOTransfer({ 'ref_hip_1', 'ref_knee_1', 'ref_hip_2', 'ref_knee_2', 'ref_hip_3',
    'ref_knee_3', 'ref_hip_4', 'ref_knee_4'}, { 'sensor_hip_1', 'sensor_knee_1',
    'sensor_hip_2', 'sensor_knee_2', 'sensor_hip_3', 'sensor_knee_3', 'sensor_hip_4',
    'sensor_knee_4'});
clf, step(T), grid;
ST.showBlockValue();
ST.writeBlockValue();
```

A função `open_system('robo_v7')` dá instrução ao Matlab[®] para abrir o modelo do robô modelado, em seguida são indicados os controladores que se pretende que o `looptune` sintonize. A função `slTunable` aceita como primeiro parâmetro o nome do modelo onde se encontram os blocos que se pretende sintonizar e como segundo parâmetro os nomes dos diversos blocos de controlo (ver esquema de sistema de controlo na Figura 5.19). Numa segunda fase, são indicados os nomes dos sinais onde o Simulink[™] deverá ir buscar as informações de posição das juntas (`addMeasurement(ST0, 'sensor_system')`) e também o sinal de controlo que será utilizado para actuar o sistema (`addControl(ST0, 'actuator_system')`). Como último passo na definição do sistema é importante indicar ao programa os elementos necessários para que este possa calcular o erro a introduzir no controlador, estes são os sinais de referência para cada uma das juntas e os sinais de posição actual, retirados da demultiplexação do sinal `sensor_system`. Para esse efeito utiliza-se a função `addIO`.

Nesta fase do código o sistema de controlo já se encontra totalmente definido, tendo sido introduzidos os sinais que limitam o sistema de controlo: sinal de referência, sinal de posição real da junta e sinal de controlo. A variável `Options` é nesta altura definida como uma estrutura onde são definidas as diversas opções pretendidas para o `looptune`. Antes de se proceder à sintonização dos controladores deve ser definida uma variável com os nomes dos sinais de referência que se pretende que o `looptune` procure seguir e respectivos sinais reais de posição para que seja possível o cálculo do erro.

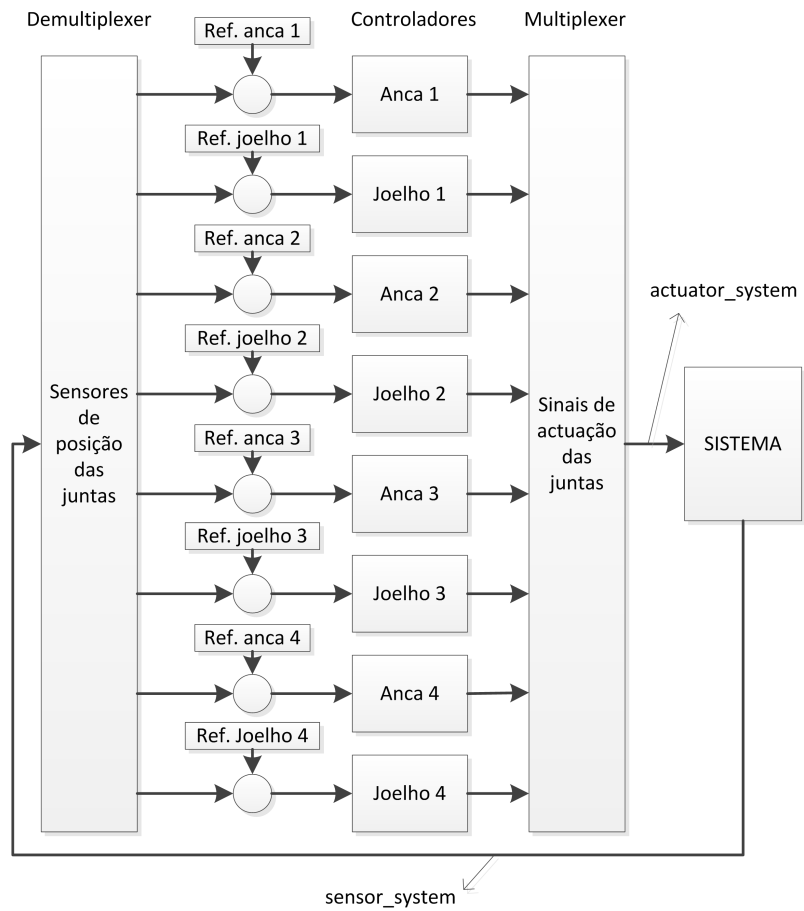


Figura 5.19: Esquema do sistema de controlo.

Na linha seguinte de código é executado o looptune introduzindo os nomes dos diversos sinais e blocos que constituem o sistema de controlo, a frequência de corte dos controladores e as restantes informações de optimização do sistema de controlo tais como as referências que se pretende que o controlador siga e as opções do sistema de controlo como definidas pela variável *Options*. Depois de correr a ferramenta de sintonização dos controladores basta executar a instrução `ST.WRITEBLOCKVALUE` para que os ganhos dos controladores sejam automaticamente escritos para os respectivos blocos.

O restante código apresentado permite a verificação da qualidade da resposta do sistema de controlo pela aplicação de um degrau ao sistema e verificação da resposta obtida.

Embora tivesse sido possível ver o robô em movimento com os ganhos calculados por este sistema, por razões desconhecidas o robô "perdia o controlo" nos instantes iniciais e começava a cair para a frente (Figura 5.20). Mais tarde verificou-se que esta reacção era devida aos baixos ganhos que a ferramenta looptune estava a atribuir aos controladores.

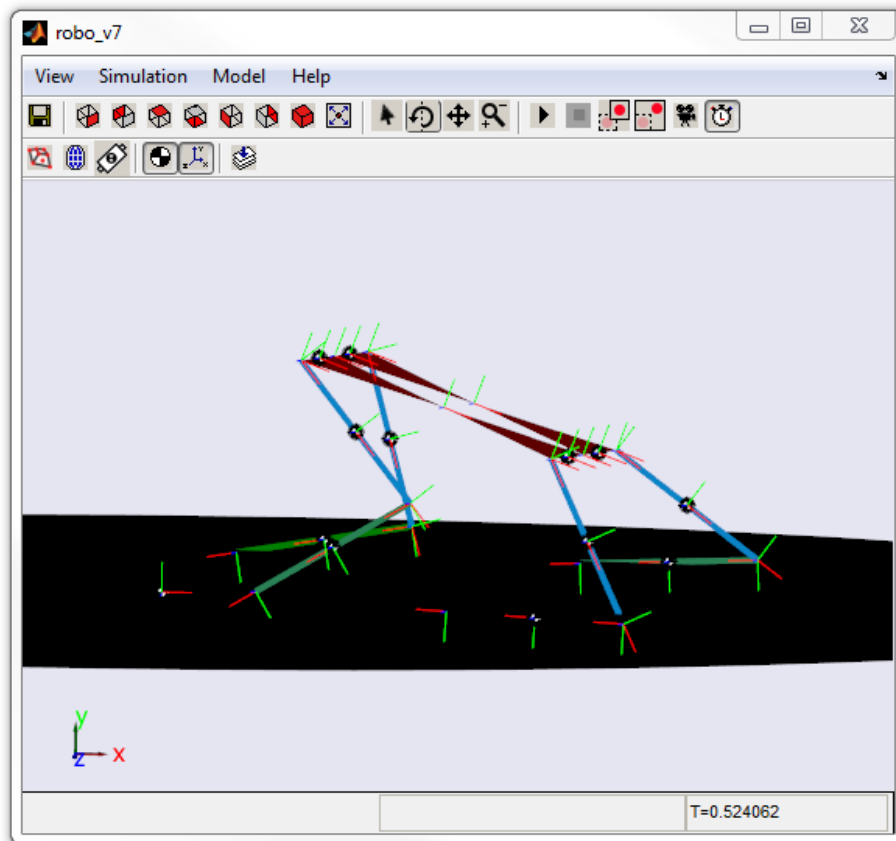


Figura 5.20: Simulação do robô utilizando ganhos obtidos através da ferramenta looptune.

Por restrições temporais não foi possível diagnosticar com maior detalhe e corrigir a razão pela qual esta ferramenta não estaria a funcionar correctamente.

5.5 Montagem final do robô

O valor do peso do corpo teve que ser alterado porque um corpo sólido com 0,3 m de largura, 0,15 m de altura e 0,2 m de profundidade em alumínio (densidade de 2.800 kg/m^3) pesava 25,2 kg, o que multiplicado pelos quatro constituintes do corpo daria um total de 100,8 kg. Considerou-se esse peso excessivo tendo em conta as dimensões definidas para o robô. Optou-se pela alteração da densidade do corpo, passando a 1.000 kg/m^3 para o corpo, em vez dos 2.800 kg/m^3 inicialmente previstos, traduzindo-se num peso total do corpo de 36 kg (Figura 5.21).

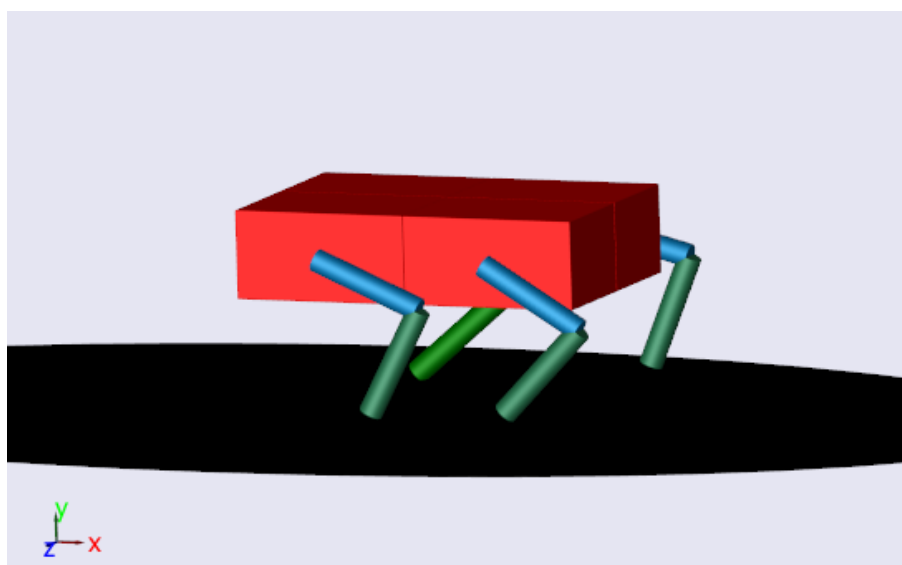


Figura 5.21: Simulação do robô.

Definiu-se que tanto a perna superior do robô como a inferior teriam um comprimento de 0,2 m e um raio de 0,02 m. Em relação às pernas manteve-se a densidade do alumínio obtendo-se um peso final de 1,4 kg por perna.

Assim, o peso total do robô é de 41,6296 kg.

Numa fase intermédia do trabalho, devido ao elevado número de blocos presentes no modelo, foi importante criar subsistemas. Na "raiz" do modelo foram colocados os controladores das juntas, foi criado um subsistema para alojar os blocos constituintes do corpo e os subsistemas das pernas (Figura 5.22) e também foi adicionado um bloco para permitir a visualização do solo durante a simulação.

Por razões de espaço não é possível apresentar neste documento uma imagem do subsistema onde se encontram os blocos do corpo do robô e os subsistemas das pernas, podendo os mesmos ser consultados no CD anexo a este documento. Apresenta-se no entanto um esquema da organização do modelo para facilitar a sua compreensão (Figura 5.23).

Trabalho desenvolvido

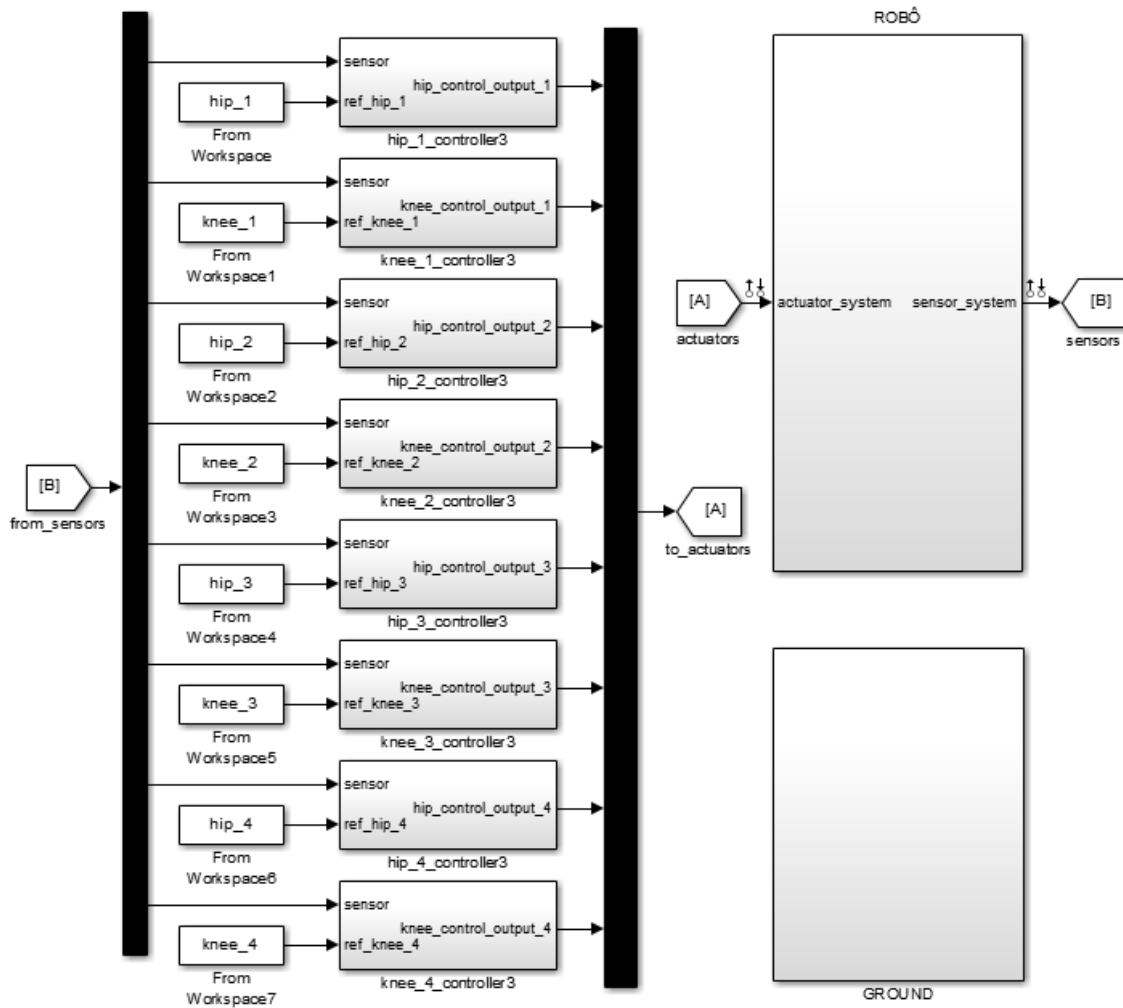


Figura 5.22: Esquema do modelo do robô.

Trabalho desenvolvido

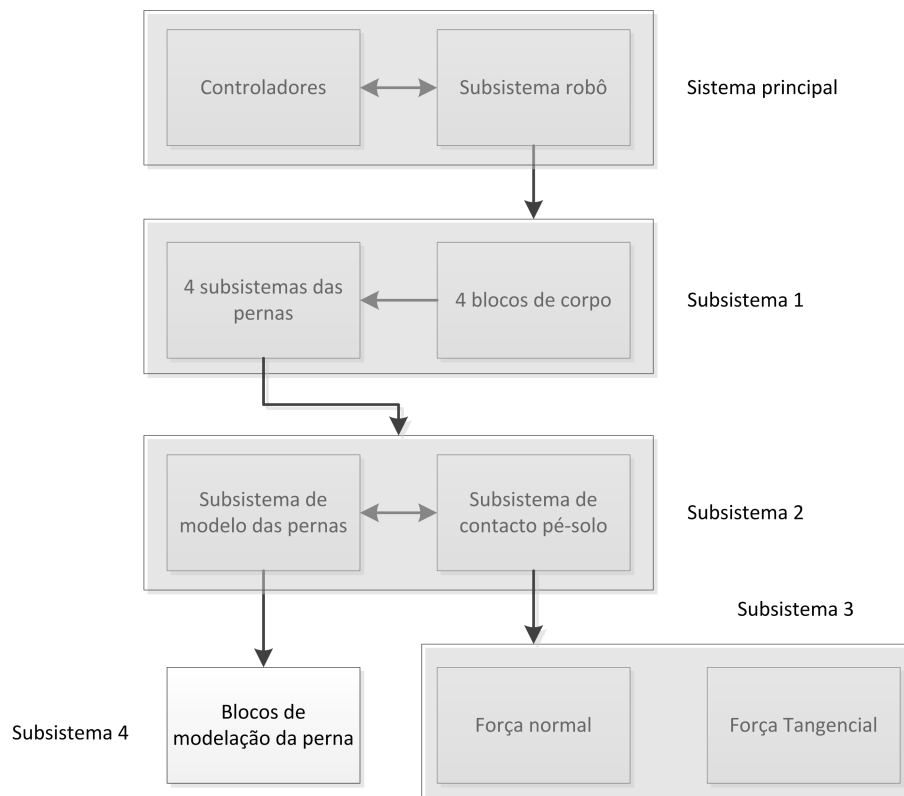


Figura 5.23: Esquema de organização do modelo do robô.

Dentro do *subsistema 1* encontram-se os 4 blocos que constituem o corpo assim como as juntas universais que os unem. A cada bloco de corpo encontra-se ligado um subsistema identificado na Figura 5.23 como *subsistema 2* onde estão definidos os subsistemas do contacto com o solo e o subsistema com os blocos de modelação da perna (Figura 5.24).

Esta organização do modelo facilita a visualização do robô pela forma como estão dispostos os diversos blocos e subsistemas.

5.6 Problemas encontrados, soluções adoptadas e resultados obtidos

Foram diversos os problemas encontrados durante o trabalho descrito nesta dissertação.

5.6.1 Teste de gerador central de padrões (CPG)

Numa fase inicial de simulação, depois da modelação da perna, utilizou-se um gerador central de padrões (CPG), baseado num artigo de Roy e Demiris [PR05], para gerar os ângulos que seriam utilizados nas juntas das ancas e dos joelhos. O CPG que foi testado tinha um problema: os ângulos fornecidos não tinham em conta o movimento do robô

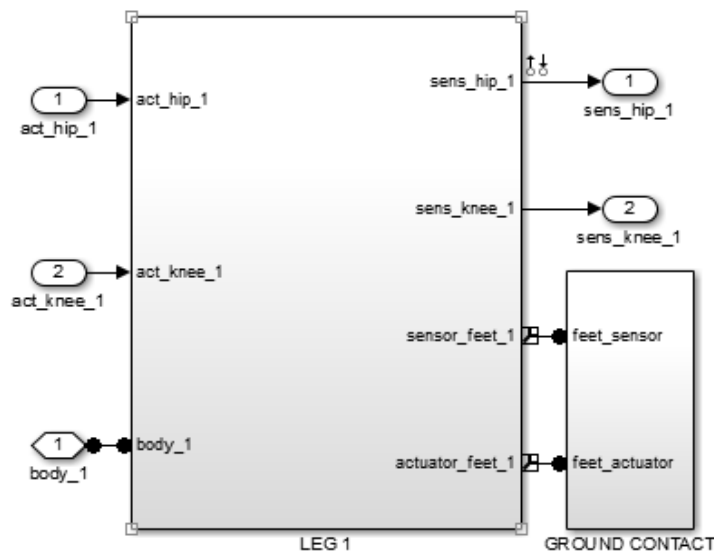


Figura 5.24: Subsistema 2.

sendo sempre relativos ao eixo horizontal dos x . No Simmechanics™ os ângulos introduzidos num actuador de junta são sempre relativos à posição actual do SC ao qual a junta se encontra conectada. Este método foi abandonado porque a sua alteração obrigaria a um estudo profundo do seu funcionamento, considerando-se que estava fora do âmbito deste trabalho.

5.6.2 Padrão de locomoção gerado por aplicação externa

Posto isto, optou-se pela geração dos vectores com as posições angulares das ancas e dos joelhos através de um programa, desenvolvido em linguagem C, pelo Professor Manuel Fernando dos Santos Silva. Este programa permite a geração de diversos tipos de padrões de locomoção recorrendo à cinemática inversa.

O programa cria 3 ficheiros de texto contendo os vectores com as posições angulares das ancas *teta1.txt*, os vectores com as posições angulares dos joelhos *teta2.txt* e o vector com os instantes de tempo *t.txt*. Estes vectores são processados dentro do ficheiro de configuração do modelo extraíndo-se a informação relativa a cada uma das juntas para cada uma das pernas.

Testou-se o movimento da perna, verificando-se um movimento em tudo semelhante ao de uma perna de um robô.

5.6.3 Problema de interação pé-solo

Com os vectores de posição das juntas definidos testou-se o contacto dos pés do robô com o solo. Verificou-se que quando o pé entrava em contacto com o solo surgiam forças

de reacção do solo, no entanto essas forças não tinham qualquer efeito na trajectória do pé (Figura 5.25) verificando-se que estes penetravam no solo.

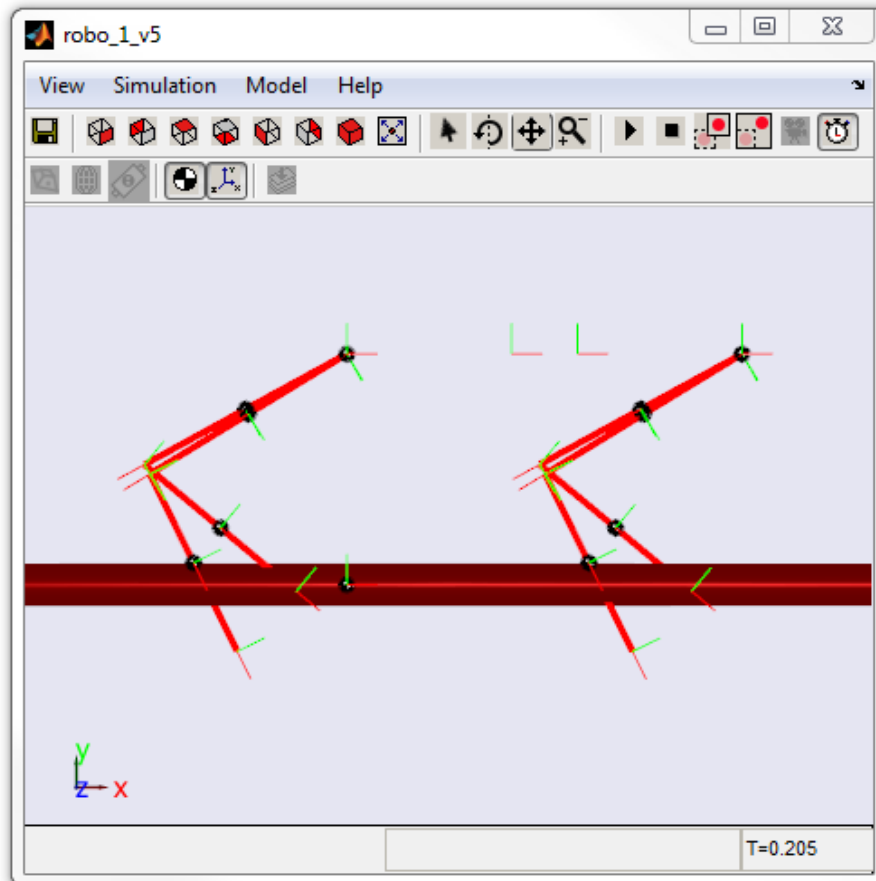


Figura 5.25: Robô com os pés a trespassar o solo.

Dos vários testes efectuados concluiu-se que o Simmechanics™ estaria a assumir o movimento introduzido nas juntas como prioritário, aplicando nas mesmas um binário suficientemente elevado para ultrapassar qualquer tipo de força imposta pelo solo.

Resolveu-se este problema introduzindo controladores do tipo PD nas juntas das ancas e joelhos. Desta forma conseguiu-se diminuir o binário aplicado nas juntas, simulando correctamente a interacção entre o solo e o pé do robô. Este método funcionou de imediato verificando-se que o pé já não conseguia trespassar o solo. Foram efectuados diversos testes para garantir que as forças de reacção do solo tinham a intensidade e sentido desejados.

Nesta altura, com o modelo de interacção pé-solo a funcionar, replicou-se o modelo da perna e montou-se o robô.

Com o robô fixo ao "mundo" e afastado do chão, de forma a nunca entrar em contacto com o solo, testou-se o padrão de locomoção, verificando se os controladores das juntas conseguiam seguir as referências introduzidas. Constatou-se que o modelo não tinha qualquer problema em seguir as trajectórias introduzidas nas juntas.

5.6.4 Problema de velocidade de simulação

Tendo-se, com sucesso, observado o contacto de um pé com o solo e estando o robô completamente montado decidiu-se testar o movimento do robô no espaço tridimensional. Introduziu-se uma junta do tipo 6-dof entre o robô e o "mundo" e iniciou-se a simulação.

Embora desse para perceber que o robô caía por acção da gravidade e que quando entrava em contacto com o solo não o trespassava, o modelo estava muito lento não sendo possível observar um passo completo.

Inicialmente considerou-se que seria um problema relacionado com o modelo de solo, tendo sido efectuadas diversas experiências, procurando identificar a origem do problema que tornava o sistema tão lento. Depois de inúmeras tentativas de alteração do modelo de solo, os problemas de velocidade de simulação mantinham-se. Procurou-se então verificar se a origem do problema estaria nos ganhos dos controladores. Constatou-se que por maiores que fossem as alterações aos ganhos dos controladores, ou o sistema se tornava completamente instável acabando por bloquear o programa de simulação ou então o robô caía por acção da gravidade e no instante que entrava em contacto com o solo a simulação começa a ficar mais lenta acabando sempre por bloquear.

Depois de algumas tentativas de resolução deste problema, verificou-se que havia duas razões principais que estariam a provocar este atraso na simulação e, conseqüentemente, os bloqueios do programa. Estas eram o método de solução (*solver*) e a detecção de passagem por zero (*Zero Crossing Detection*). A forma como estes problemas foram resolvidos são apresentados nas duas subsecções seguintes.

5.6.5 Escolha do método de simulação (*solver*)

Um factor desde cedo identificado como muito influente na velocidade de simulação foi o tipo de *solver* a utilizar. Numa fase inicial do trabalho utilizou-se um período de amostragem fixo. Cedo se percebeu que essa solução não funcionava: a dinâmica do sistema dava origem a muitos problemas de singularidades tornando impossível a simulação do modelo. Optou-se então pela utilização de um método de solução com período de amostragem variável. Foram considerados diversos métodos de simulação, verificando-se que o *solver ode15s* é o que apresenta melhor relação estabilidade/velocidade de simulação para o modelo descrito neste trabalho.

Depois de ultrapassada a dificuldade de selecção do *solver* o sistema já começava a apresentar uma resposta mais aceitável, mas a lentidão de simulação tornava difícil observar, em tempo útil, o contacto do robô com o solo e o efeito que este teria na locomoção do robô.

5.6.6 Optimização da velocidade de simulação

Procurou-se na literatura existente e foi encontrado um endereço na Internet onde são apresentados parâmetros indispensáveis para a optimização da velocidade de simulação [Mata].

Foram-se efectuando as diversas alterações indicadas neste documento, verificando-se a cada passo a influência que as mesmas teriam na velocidade de simulação. Embora todas as alterações descritas contribuam para a optimização da velocidade de simulação, verificou-se que a opção que mais impacto tinha na velocidade de simulação era a de "*Zero Crossing Detection*". Antes de desactivar esta opção verificou-se qual o efeito que a mesma teria no sistema. Esta opção permite ao simulador verificar se, entre instantes de tempo consecutivos, o parâmetro a ser calculado passou por zero o que, em diversas ocasiões, aumenta significativamente a precisão e rigor da simulação (Figura 5.26).

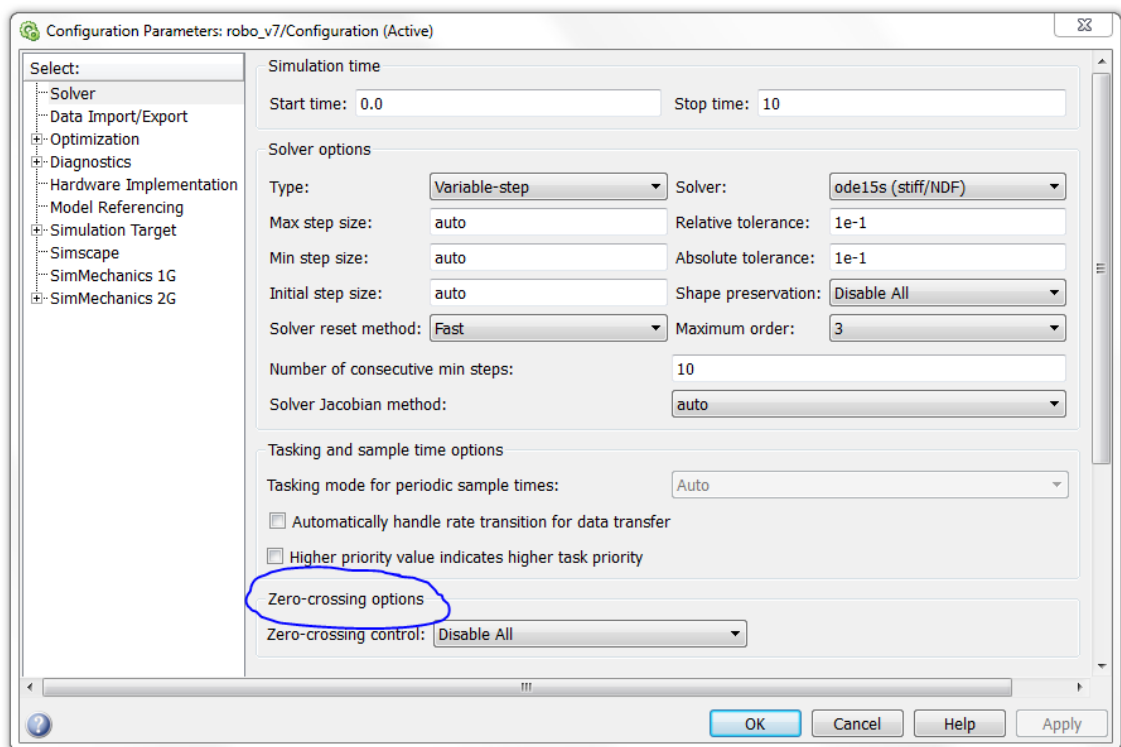


Figura 5.26: Janela de desactivação de *Zero Crossing Detection*.

Esta opção, embora dê origem a uma simulação mais real do modelo, provoca um atraso muito grande na simulação. Este atraso era devido a algumas vibrações detectadas no pé do robô que o levavam a cruzar a cota zero do solo diversas vezes consecutivas aumentando significativamente a quantidade de informação a processar dando, por vezes, origem a erros no Matlab®.

Procurando na literatura, encontrou-se um endereço na Internet [Mate] onde era referido que no caso de existirem problemas de *chattering* na simulação deve ser desac-

tivada a detecção de passagem por zero ("*Zero crossing detection*"). Procedeu-se dessa forma, desligando essa opção, e verificou-se um aumento significativo da velocidade de simulação conseguindo-se identificar um movimento que já se começava a parecer com o caminhar pretendido.

Posteriormente verificou-se que na versão mais recente do Matlab® existe uma ferramenta chamada *Simulink Performance Advisor*. Esta ferramenta detecta quais as fases ou componentes do modelo que estão a causar mais atrasos na simulação e sugere alterações. Utilizou-se esta ferramenta para otimizar o desempenho do modelo (Figura 5.27).

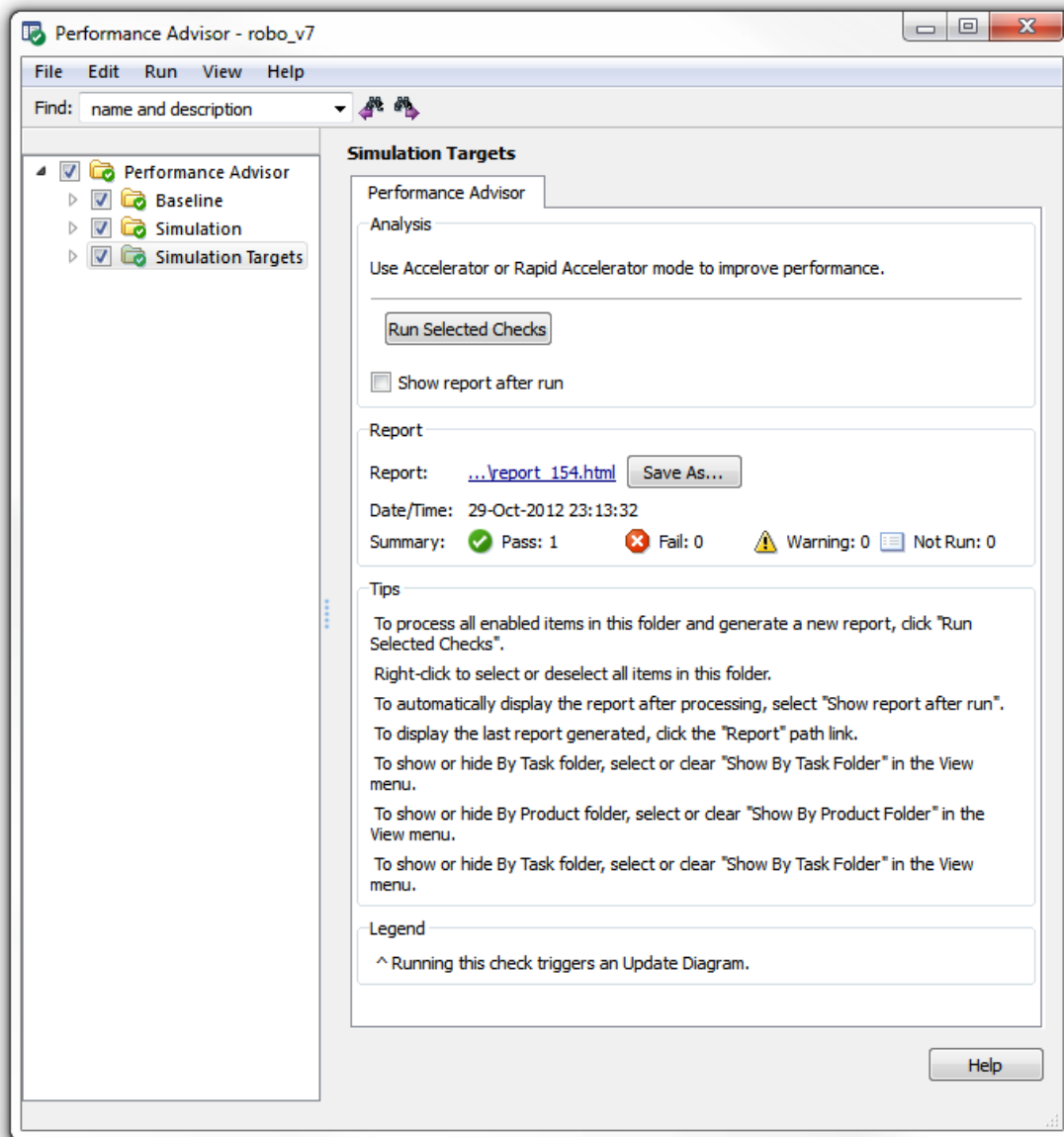


Figura 5.27: *Simulink Performance Advisor*.

Depois de otimizado o modelo, verificou-se que o sistema demorava 51 segundos a simular 1 segundo de movimento.

5.6.7 Análise do contacto pé-solo

Com o modelo otimizado e uma velocidade de simulação aceitável, procedeu-se à optimização dos seus controladores. Após algumas tentativas de implementação da ferramenta looptune do Simulink™ optou-se pela sintonização manual dos controladores. Verificou-se tratar de uma tarefa muito morosa, no entanto obtendo-se resultados positivos.

Pela análise da Figura 5.28 observa-se a evolução das forças à medida que se dá o contacto do pé com o solo. O contacto estabelece-se aos 0,06 segundos e nesse instante as forças de contacto começam gradualmente a subir. Observe-se a coerência das forças. No caso da força normal, à medida que o pé penetra no solo a força vai aumentando observando-se o máximo aos 0,07 segundos; neste instante o movimento do pé é invertido começando a sair do solo. No instante de tempo 0,1 segundos o pé abandona o contacto com o solo e a força torna-se nula.

Quando se estabelece o contacto inicial o pé encontra-se com velocidade no eixo dos x negativa, a força tangencial é inicialmente positiva ocorrendo um pico no instante de contacto devido à elevada velocidade. À medida que o pé vai deslizando para trás a força tangencial aumenta, tornando-se nula no instante em que o pé abandona o solo.

5.6.8 Análise da complacência do corpo

Foi analisada a complacência do corpo pela introdução de um sensor de junta na junta universal que une o bloco 1 do corpo ao bloco 2.

Na Figura 5.29 observam-se as oscilações induzidas no corpo pelo contacto da perna com o solo. Seria interessante, num trabalho futuro, testar diferentes parâmetros no sistema mola-amortecedor que modela a complacência do corpo, de forma a perceber qual a sua influência no movimento do robô.

5.6.9 Validação do caminhar do robô

Depois de sintonizar os controladores do robô procedeu-se à análise do seu movimento, tal como representado nas Figuras 5.30 e 5.31.

Foi possível observar o robô a dar diversos passos durante aproximadamente 7 segundos (Figura 5.32).

Aos 7 segundos, possivelmente devido a uma transição rápida nos valores de referência das juntas que obriga a uma velocidade excessiva das pernas, o sistema torna-se instável (Figuras 5.33, 5.34 e 5.35).

Esta instabilidade deve-se ao aumento súbito das forças de contacto das pernas 3 e 4 com o solo, tal como se pode verificar no gráfico da Figura 5.36 retirado dos sensores da perna 3.

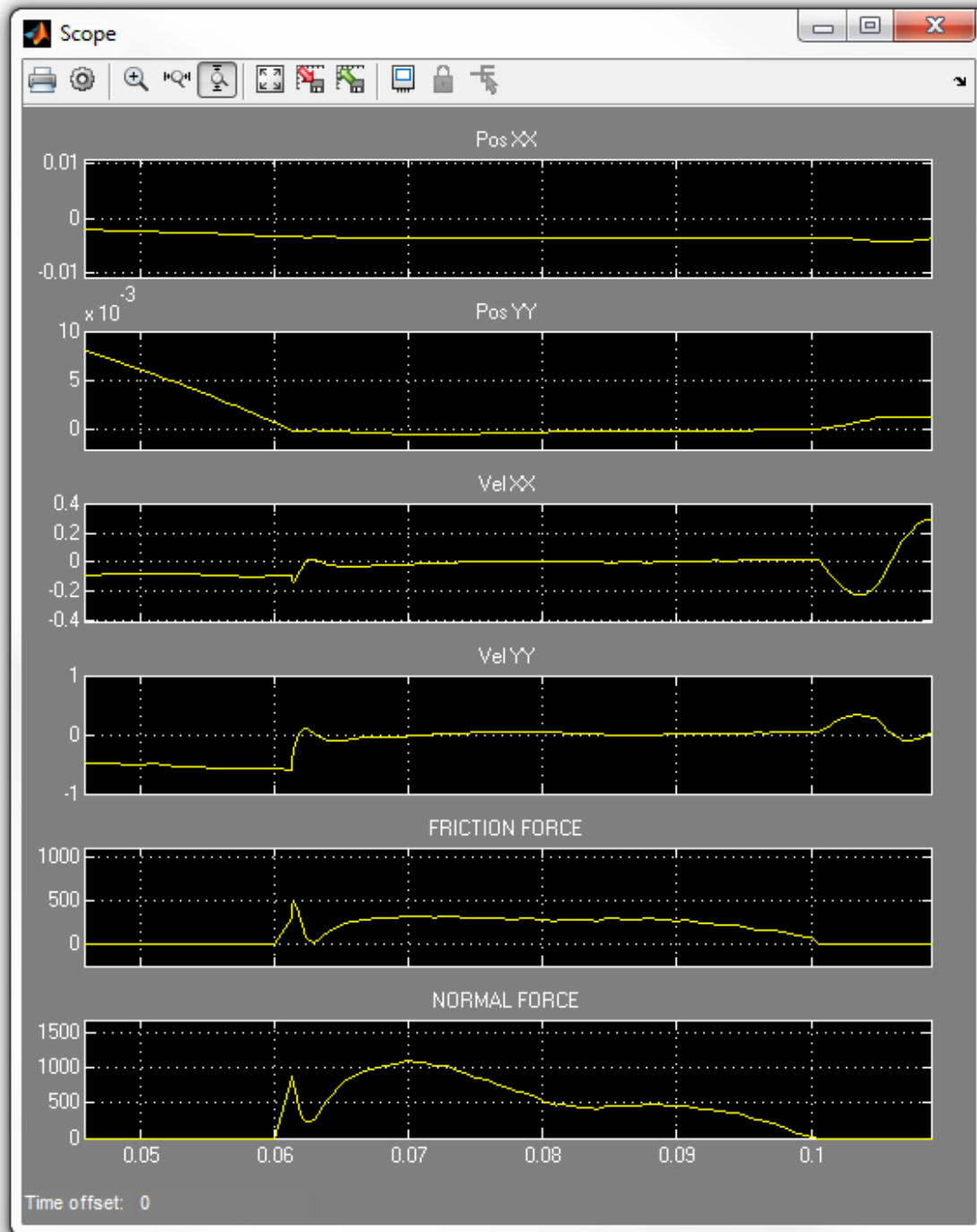


Figura 5.28: Contacto pé-solo.

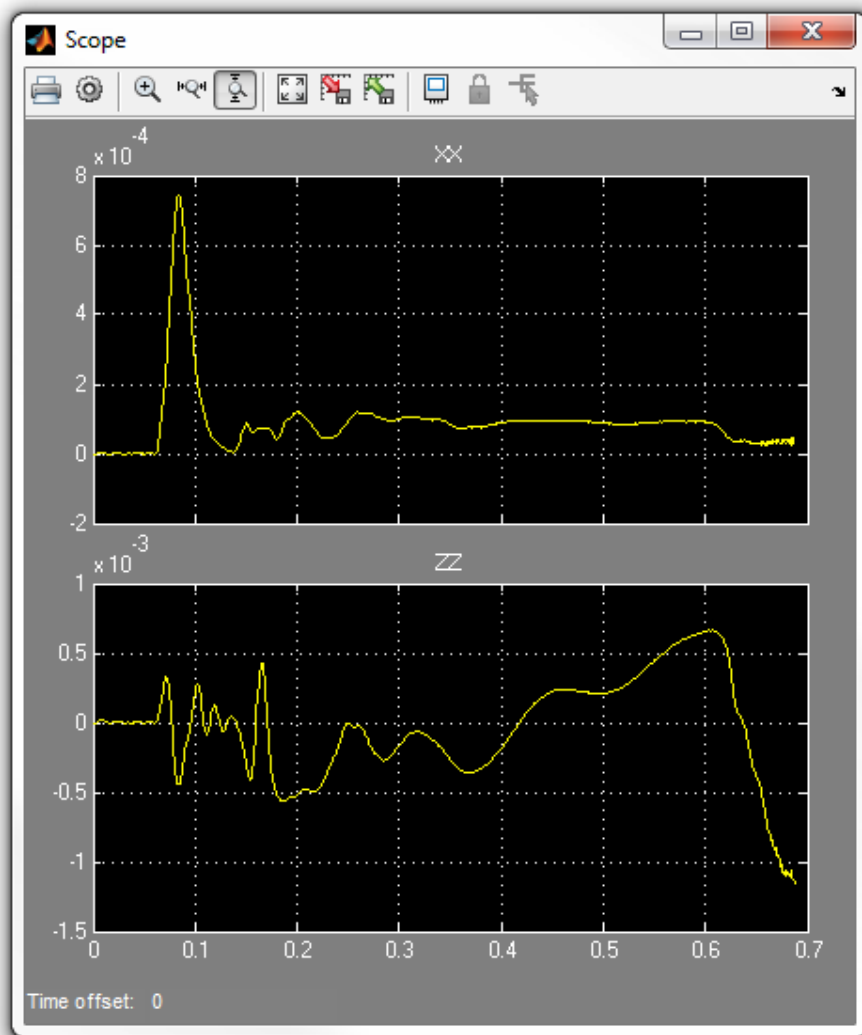


Figura 5.29: Complacência do corpo em x e z .

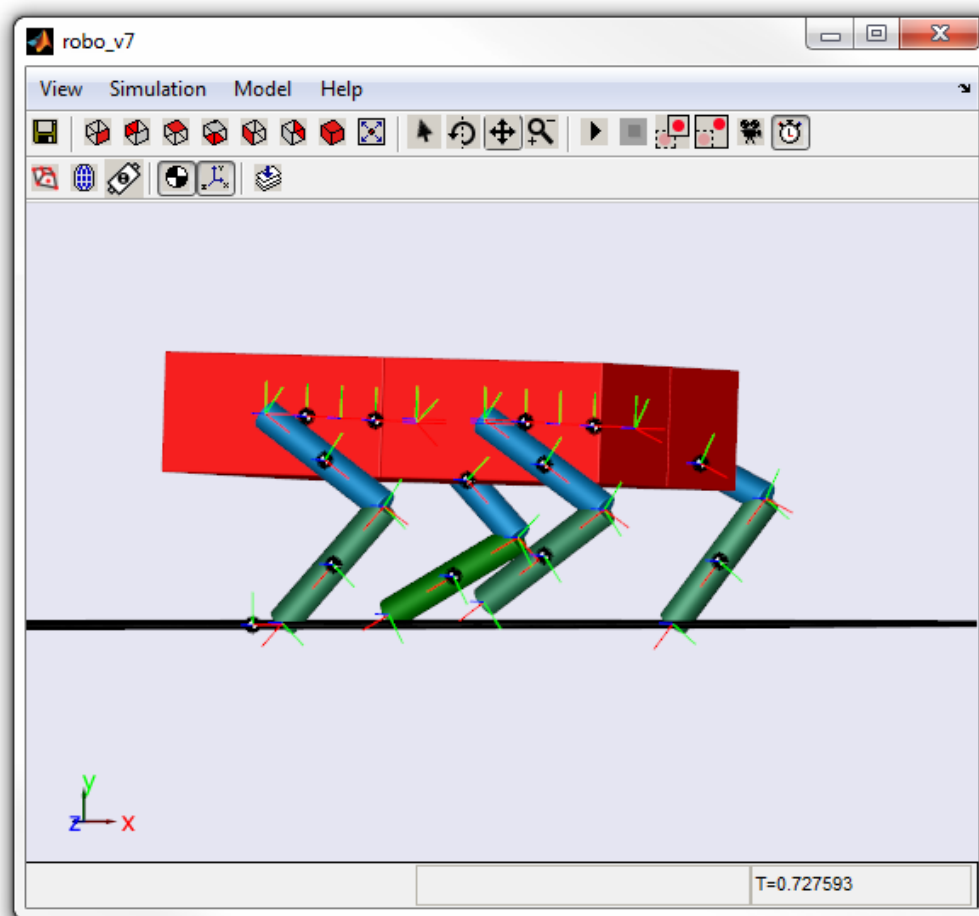


Figura 5.30: Robô a caminhar com as pernas 2 e 3 na fase de transferência.

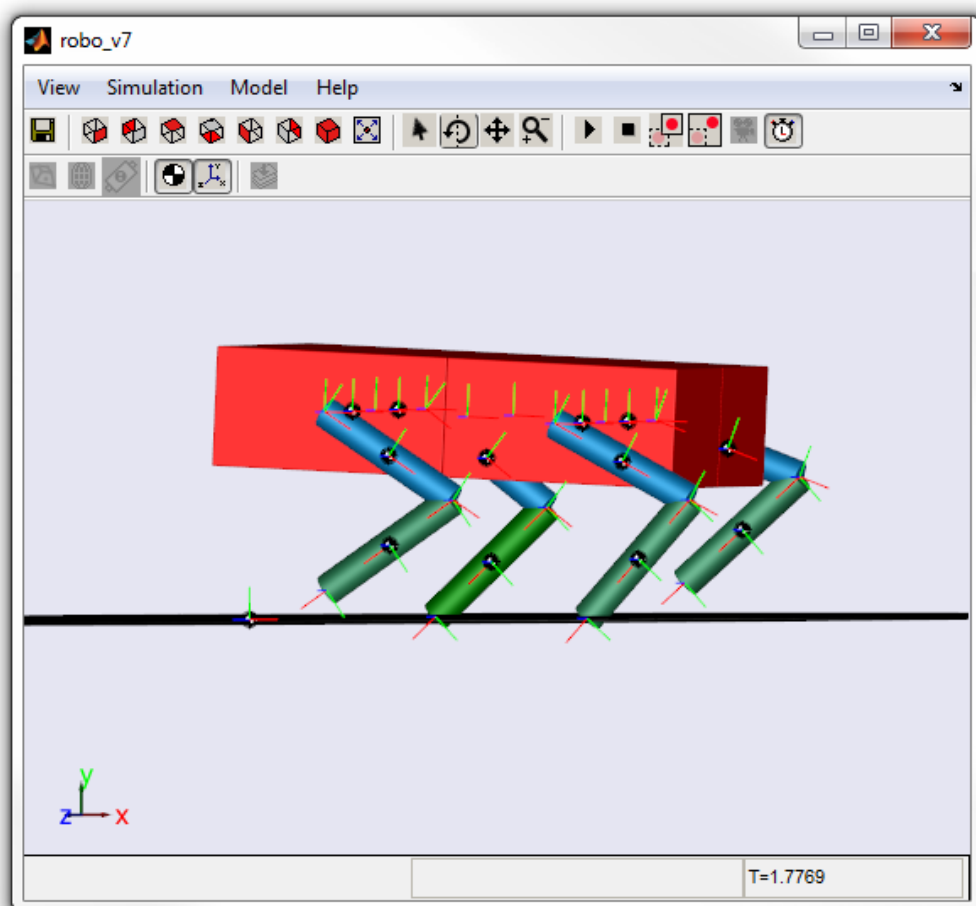


Figura 5.31: Robô a caminhar com as pernas 1 e 4 na fase de transferência.

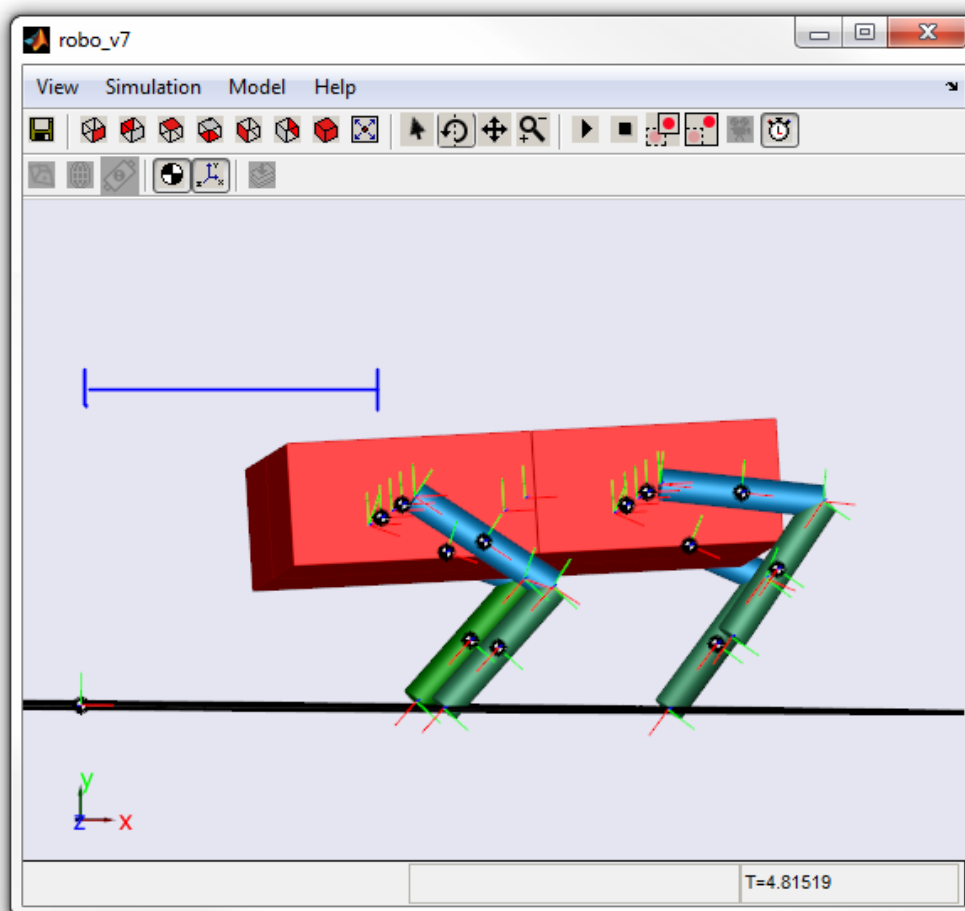


Figura 5.32: Evolução da posição do robô no espaço.

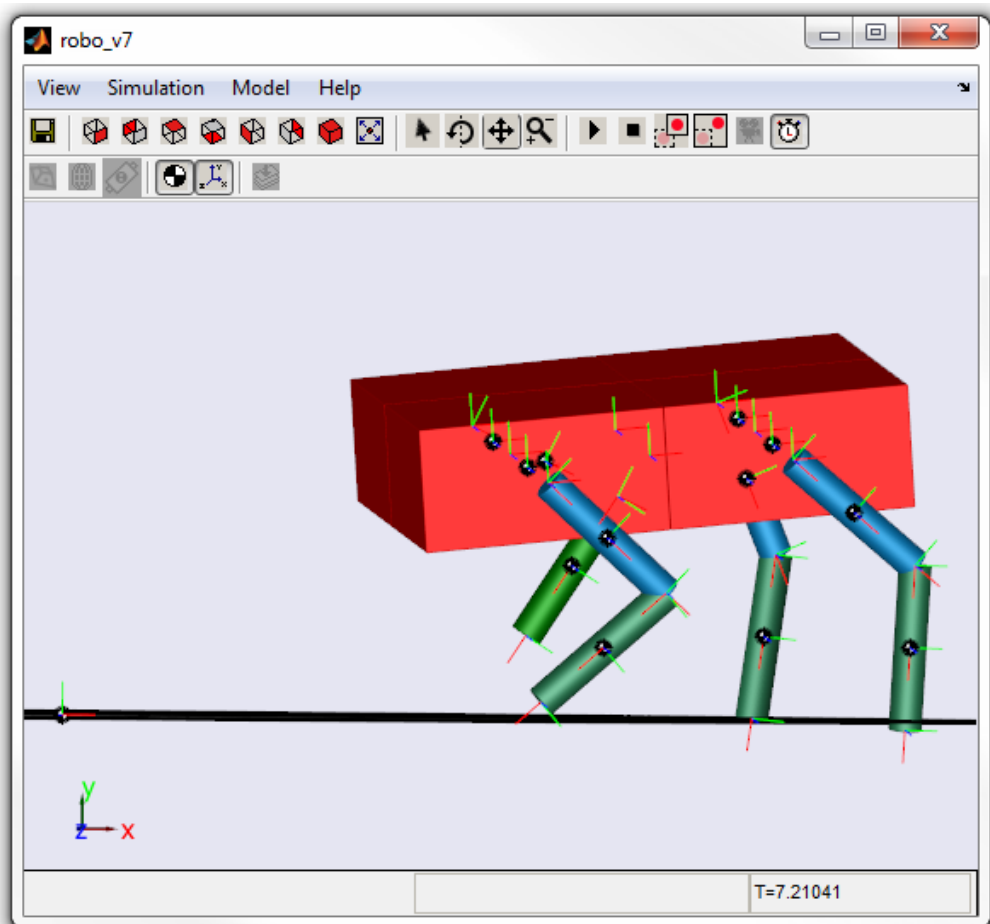


Figura 5.33: Estado do robô instantes antes da instabilidade.

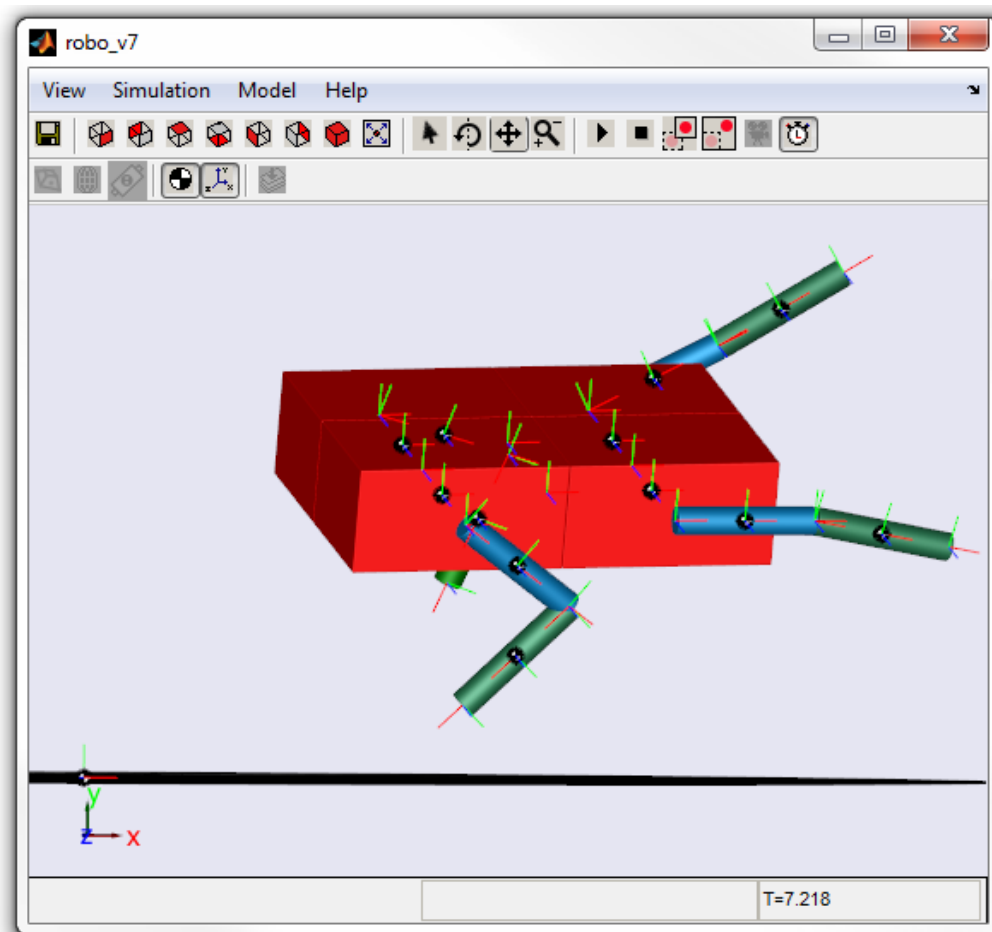


Figura 5.34: Estado do robô no momento em que o sistema se torna instável.

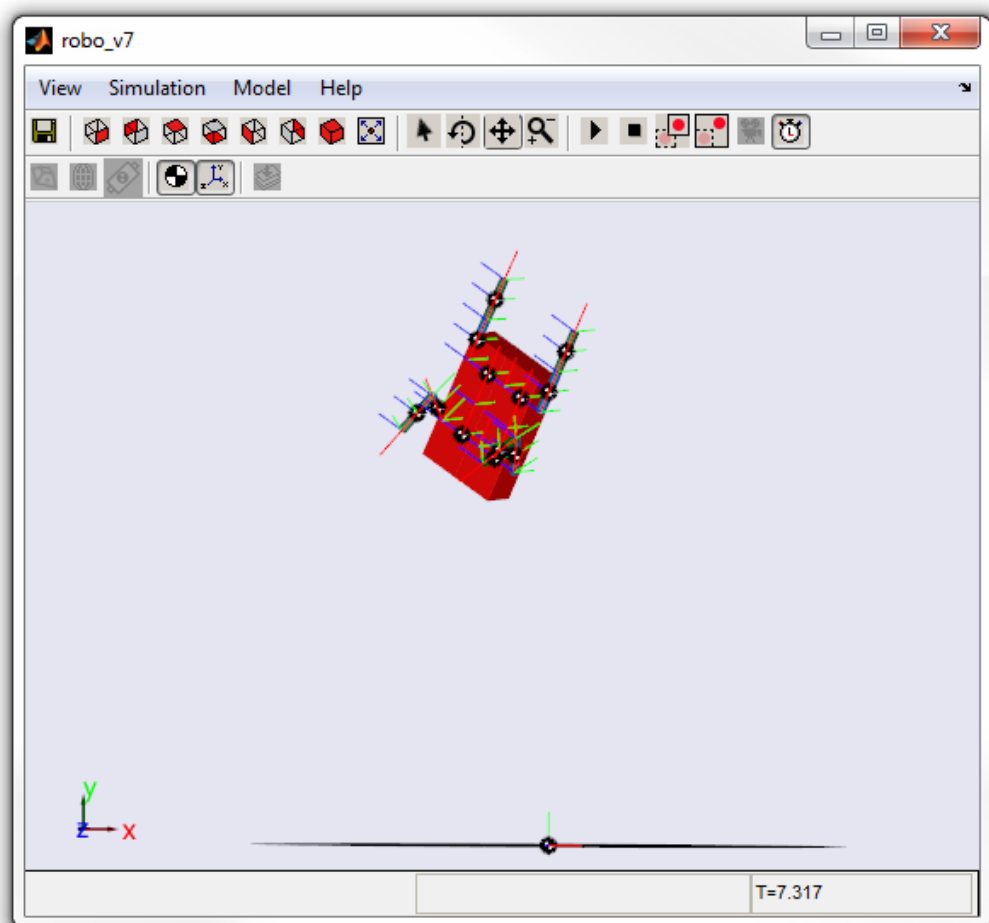


Figura 5.35: Sistema instável com o robô a "voar".

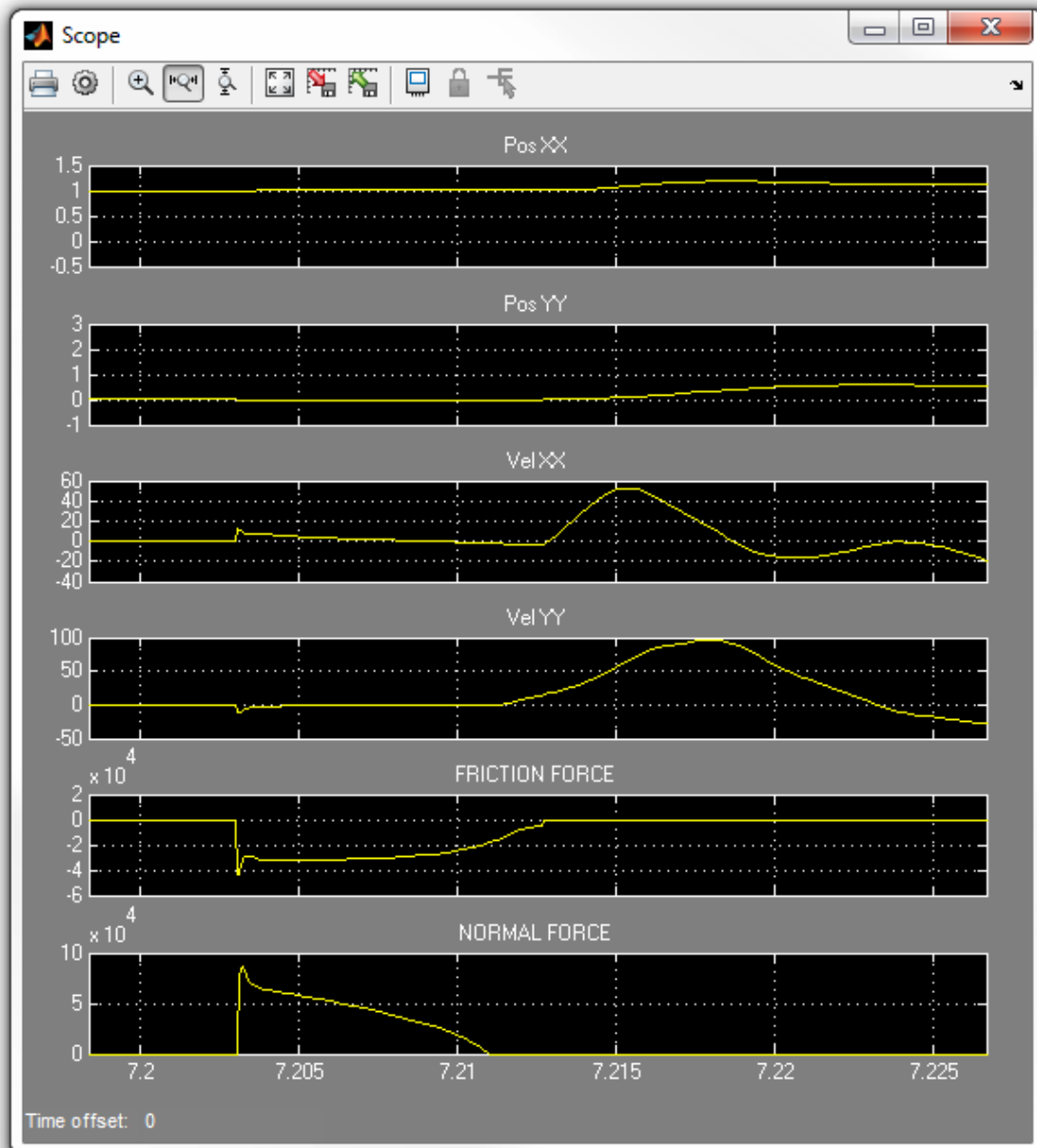


Figura 5.36: Forças de contacto da perna 3 no instante em que o sistema se torna instável.

Trabalho desenvolvido

De notar na Figura 5.36 o aumento súbito das forças de contacto, usualmente com valores absolutos abaixo de 1.000, neste momento aumentando em 40 vezes para a força tangencial e 80 vezes para a força normal.

Na opinião do autor, a utilização de um padrão de locomoção diferente e a sintonização mais fina dos controladores das juntas, deverão resolver esta situação. Não foi possível testar esta teoria por questões de tempo.

Trabalho desenvolvido

Capítulo 6

Conclusões

A área da simulação robótica está já num estado de desenvolvimento bastante avançado existindo no mercado diversas aplicações para modelação e teste de sistemas robóticos. Os conceitos associados à modelação robótica são bastante complexos obrigando a um estudo aprofundado das questões mecânicas associadas.

Com o desenrolar deste trabalho foram surgindo novas questões que inicialmente não tinham sido previstas. Estas questões foram resolvidas e as suas soluções explicadas nesta dissertação.

Inicialmente considerou-se a possibilidade de, através das posições desejadas para o pé, determinar-se os ângulos desejados para as ancas e joelhos. Este modelo não funcionava porque, para além de introduzir demasiados cálculos em cada instante de tempo, dava origem a muitos erros de singularidades. Nessa altura percebeu-se que não seria possível simular o modelo em modo de *Inverse Dynamics*. Optou-se então por simular o modelo em *Forward Dynamics* e através de diferentes sensores ler os valores pretendidos. Neste modo de simulação são colocadas menos limitações, sendo na mesma possível a introdução de movimento nas juntas de forma a obter as forças/binários pretendidos.

Verificaram-se diversas características interessantes do Simmechanics™ tal como descrito na secção 5.6.

6.1 Satisfação dos objectivos

O objectivo de criação de um modelo de simulação de um robô quadrúpede em Simmechanics™ foi alcançado na totalidade.

Dos objectivos parciais definidos na secção 1.2, o autor considera que a sintonização dos controladores das juntas deveria ser melhor trabalhada porque, embora se consiga identificar perfeitamente o caminhar do robô, pela optimização do sistema de controlo e melhoria da parametrização do padrão de locomoção, dever-se-ia conseguir obter um caminhar mais estável.

Com a experiência adquirida durante a execução deste trabalho, o autor considera que deve, inicialmente, ser implementado um padrão de locomoção relativamente lento, evitando grandes mudanças angulares em períodos curtos de tempo, facilitando assim a sintonização dos controladores e identificação de eventuais problemas de modelação.

O modelo de solo funciona sem qualquer tipo de problema apresentando valores coerentes com o movimento do robô, e pela utilização de um modelo não linear de amortecedor na componente normal da força de contacto conseguiu-se eliminar as forças que repeliam o pé do solo no contacto inicial e que o impediam de abandonar o solo no final do contacto.

O modelo de complacência utilizado no corpo também está operacional. verificando-se efectivamente a torção entre corpos apresentado na subsecção 5.6.8.

6.2 Trabalho futuro

A enorme complexidade desta ferramenta permite a modelação de sistemas com um grau de precisão e de rigor muito elevados. Existem inúmeras possibilidades de desenvolvimento deste trabalho.

Em relação ao modelo, seria muito interessante estudar as propriedades físicas de um robô quadrúpede e implementá-las neste modelo obtendo-se um modelo cujas características fossem o mais reais possível.

Os parâmetros utilizados para criação do padrão de locomoção utilizado neste trabalho não foram otimizados para o tamanho do robô. Como trabalho futuro seria interessante otimizar esses parâmetros.

Na continuação deste trabalho dever-se-ia testar diferentes padrões de locomoção, com diferentes velocidades, para ver de que forma o sistema modelado é flexível e de fácil adaptação aos diferentes padrões de locomoção.

O estudo mais aprofundado do modelo de complacência do corpo poderia permitir encontrar uma relação mais otimizada entre a velocidade de locomoção, o padrão de locomoção e os valores de parametrização do sistema mola-amortecedor utilizados no corpo.

Em relação ao modelo poder-se-ia introduzir mais um grau de liberdade em cada perna que iria permitir ao robô deslocar-se lateralmente, tornando-o mais completo em termos de simulação de locomoção.

Para melhorar a simulação também deve ser introduzida uma terceira componente no modelo de solo correspondente ao atrito lateral do contacto do pé com o solo.

Como trabalho futuro também seria importante a optimização do funcionamento da ferramenta Looptune; esta função tem um potencial enorme, substituindo a necessidade de sintonização manual dos controladores. A sintonização manual dos controladores, para

Conclusões

além de muito morosa, não garante a melhor resposta do sistema. A utilização do LO-optune poderá ajudar a sintonizar os controladores com maior facilidade permitindo a introdução e teste de diversos padrões de locomoção e características físicas do robô num menor espaço de tempo.

Conclusões

Referências

- [AG] SimPack AG. Programa de simulação dinâmica multi-corpo SIMPACK. <http://www.simpack.com/>. Acedido em Outubro, 2012.
- [Bar09] Ramiro Sousa Barbosa. Apontamentos da cadeira de sistemas electromecânicos, Setembro 2008/2009. <http://www.isep.ipp.pt>.
- [Cyb] Cyberbotics. Programa de simulação dinâmica multi-corpo WEBOTS. <http://www.cyberbotics.com>. Acedido em Setembro, 2012.
- [DAR12] DARPA. Rhex, Acedido em Maio 2012. <http://kodlab.seas.upenn.edu/RHex/Home>.
- [Dyn12a] Boston Dynamics. Bigdog, Acedido em Maio 2012. http://www.bostondynamics.com/robot_bigdog.html.
- [Dyn12b] Boston Dynamics. Cheetah, Acedido em Agosto 2012. http://www.bostondynamics.com/robot_cheetah.html.
- [Hon12] Honda. Asimo, Acedido em Abril 2012. <http://asimo.honda.com>.
- [Inc] Visual Solutions Incorporated. Programa de modelação e simulação de sistemas dinâmicos VISSIM. <http://www.vissim.com/products/vissim.html>. Acedido em Outubro, 2012.
- [Lab12] MIT Leg Laboratory. Uniroo, Acedido em Janeiro 2012. <http://www.ai.mit.edu/projects/leglab/robots/robots.html>.
- [Mata] Mathworks. Improving simulation performance in simulink™. <http://www.mathworks.com/company/newsletters/articles/improving-simulation-performance-in-simulink.html>. Acedido em Outubro, 2012.
- [Matb] Mathworks. Manual de utilizador da ferramenta simmechanics™. http://www.mathworks.cn/help/pdf_doc/physmod/smlink/smlink_ug.pdf. Acedido em Outubro, 2012.
- [Matc] Mathworks. Programa de simulação dinâmica simmechanics™. <http://www.mathworks.com/products/simmechanics/>. Acedido em Novembro, 2012.
- [Matd] Mathworks. Website da ferramenta looptune. <http://www.mathworks.com/help/robust/ref/looptune.html>. Acedido em Outubro, 2012.

REFERÊNCIAS

- [Mate] Mathworks. When to enable zero-crossing detection. <http://www.mathworks.com/help/stateflow/ug/when-to-enable-zero-crossing-detection.html>. Acedido em Outubro, 2012.
- [MFS04] António M. Lopes Manuel F. Silva, J. A. Tenreiro Machado. Modelling and simulation of artificial locomotion systems. *Robotica 2005*, page 12, Outubro 2004.
- [MRP08] Gabriel Nelson Marc Raibert, Kevin Blankespoor e Rob Playter. Bigdog, the rough-terrain quadruped robot. *Proceedings of the 17th World IFAC Congress*, page 4, Julho 2008.
- [OB97] F. Ben Ouezdou O. Bruneau. Compliant contact of walking robot feet. *Proc. of the third ECPD International Conference*, page 7, Setembro 1997.
- [PR05] Yiannis Demeris Pantha Roy. Analysis of biped gait patterns generated by van der pol and rayleigh oscillators under feedback. *Imperial College London, SW7 2BT*, page 4, 2005.
- [Rob11] Cyberstein Robots. Titan, Acedido em Dezembro 2011. <http://www.cyberstein.co.uk>.
- [Sil05] Manuel F. Silva. Sistemas robóticos de locomoção multipernas, Fevereiro 2005. <http://ave.dee.isep.ipp.pt/~mss/>.
- [Sof] MSC Software. Programa de simulação dinâmica multi-corpo ADAMS. <http://www.mscsoftware.com/Products/CAE-Tools/Adams.aspx>. Acedido em Setembro, 2012.
- [Son12] Sony. Aibo, Acedido em Agosto 2012. <http://en.wikipedia.org/wiki/AIBO>.

Anexo A

Parametrização do robô

Tabela A.1: *Miscellaneous parameters*

Simulation Parameters		
<i>g_force</i>	9.81	Gravity force
<i>sample_time</i>	0.003	Sample Time
Density		
<i>rho</i>	2800	Leg Density kg/m^3
<i>rho_body</i>	1000	Body Density kg/m^3
Robot Initial Position		
<i>body_x_ic</i>	0	Initial Condition XX Axis
<i>body_y_ic</i>	0.28	Initial Condition YY Axis
<i>body_z_ic</i>	0	Initial Condition ZZ Axis

Parametrização do robô

Tabela A.2: *Feet/ground contact parameters*

Ground Contact Forces Saturation		
<i>upper_friction_saturation</i>	inf	Maximum Friction Force
<i>lower_friction_saturation</i>	-inf	Minimum Friction Force
<i>upper_normal_saturation</i>	inf	Maximum Normal Force
<i>lower_normal_saturation</i>	0	Minimum Normal Force
Soil Parameters		
<i>friction_damper</i>	3423	Damping Coefficient of Friction Force
<i>friction_spring</i>	1302152	Spring Coefficient of Friction Force
<i>normal_damper</i>	2700000	Damping Coefficient of Friction Force
<i>normal_spring</i>	1700000	Spring Coefficient of Friction Force
Ground contact non-linear settings		
<i>n</i>	0.9	
<i>m</i>	0.9	
Parameters		
<i>m_ground</i>	400	Ground Mass
<i>h_ground</i>	0.01	Ground Height YY
<i>w_ground</i>	2	Ground Length XX
<i>d_ground</i>	1	Ground Depth ZZ
<i>inertia_ground</i>	$\frac{1}{12} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}$ $a_{11} = m_ground * (h_ground^2 + d_ground^2)$ $a_{22} = m_ground * (w_ground^2 + d_ground^2)$ $a_{33} = m_ground * (w_ground^2 + h_ground^2)$	Ground inertia

Parametrização do robô

Tabela A.3: *Body parameters*

Body spring - damper system settings		
<i>spring_body_x</i>	100000	Body Spring Coefficient XX
<i>damper_body_x</i>	1000	Body Damper Coefficient XX
<i>spring_body_z</i>	10000	Body Spring Coefficient ZZ
<i>damper_body_z</i>	100	Body Damper Coefficient ZZ
Body Parameters		
<i>h_body</i>	0.15	Body Height YY
<i>w_body</i>	0.3	Body Length XX
<i>d_body</i>	0.2	Body Depth ZZ
<i>m_body</i>	$\rho_{body} * (h_{body} * d_{body} * w_{body})$	Body Mass
<i>body_inertia</i>	$\frac{1}{12} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}$ $a_{11} = m_{body} * (h_{body}^2 + d_{body}^2)$ $a_{22} = m_{body} * (w_{body}^2 + d_{body}^2)$ $a_{33} = m_{body} * (w_{body}^2 + h_{body}^2)$	Body Inertia

Tabela A.4: *PD controller settings*

PD Controller Parameters		
<i>hip_p</i>	50	Hip Controller Proportional Gain
<i>hip_i</i>	0	Hip Controller Integrative Gain
<i>hip_d</i>	5	Hip Controller Derivative Gain
<i>hip_n</i>	100	Hip Controller Filter Coefficient
<i>knee_p</i>	50	Knee Controller Proportional Gain
<i>knee_i</i>	0	Knee Controller Integrative Gain
<i>knee_d</i>	5	Knee Controller Derivative Gain
<i>knee_n</i>	100	Knee Controller Filter Coefficient
PD Controller Saturation		
<i>sat_control_hip_sup</i>	inf	Maximum value for the hip
<i>sat_control_hip_inf</i>	-inf	Minimum value for the hip
<i>sat_control_knee_sup</i>	inf	Maximum value for the knee
<i>sat_control_knee_inf</i>	-inf	Minimum value for the knee

Parametrização do robô

Tabela A.5: *Leg Inertia Tensors.*

Leg Inertia		
Leg 1		
<i>inertia_upper_leg_1</i>	$\frac{1}{12} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}$ $a_{11} = m_{upper_leg_1} * r_{upper_leg_1}^2$ $a_{22} = m_{upper_leg_1} * (3 * r_{upper_leg_1}^2 + h_{upper_leg_1}^2)$ $a_{33} = m_{upper_leg_1} * (3 * r_{upper_leg_1}^2 + h_{upper_leg_1}^2)$	Upper Leg 1 Inertia
<i>inertia_lower_leg_1</i>	$\frac{1}{12} \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}$ $a_{11} = m_{lower_leg_1} * r_{lower_leg_1}^2$ $a_{22} = m_{lower_leg_1} * (3 * r_{lower_leg_1}^2 + h_{lower_leg_1}^2)$ $a_{33} = m_{lower_leg_1} * (3 * r_{lower_leg_1}^2 + h_{lower_leg_1}^2)$	Lower Leg 1 Inertia

Tabela A.6: *Loading locomotion pattern vectors.*

Locomotion Pattern		
load t.txt		Load time samples
load teta1.txt		Load hip values
load teta2.txt		Load knee values
Leg 1		
<i>knee_1(:,2)</i>	teta2(1:6000,1)	Knee
<i>knee_1(:,1)</i>	t(1:6000)	Time vector
<i>hip_1(:,2)</i>	teta1(1:6000,1)	Hip
<i>hip_1(:,1)</i>	t(1:6000)	Time vector
Leg 2		
<i>knee_2(:,2)</i>	teta2(1:6000,3)	Knee
<i>knee_2(:,1)</i>	t(1:6000)	Time vector
<i>hip_2(:,2)</i>	teta1(1:6000,3)	Hip
<i>hip_2(:,1)</i>	t(1:6000)	Time vector
Leg 3		
<i>knee_3(:,2)</i>	knee_2(:,2)	Knee
<i>knee_3(:,1)</i>	t(1:6000)	Time vector
<i>hip_3(:,2)</i>	hip_2(:,2)	Hip
<i>hip_3(:,1)</i>	t(1:6000)	Time vector
Leg 4		
<i>knee_4(:,2)</i>	knee_1(:,4)	Knee
<i>knee_4(:,1)</i>	t(1:6000)	Time vector
<i>hip_4(:,2)</i>	hip_1(:,4)	Hip
<i>hip_4(:,1)</i>	t(1:6000)	Time vector