Construção de um Modelo Realista e Controlo de um Robô Humanóide

José Luís Sousa de Magalhães Lima

Departamento de Engenharia Electrotécnica e de Computadores

Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para a obtenção do grau de

Doutor em Engenharia Electrotécnica e de Computadores

pela Faculdade de Engenharia da Universidade do Porto

Trabalho realizado sob a orientação do Professor Doutor

Paulo José Cerqueira Gomes da Costa

e do Professor Doutor

António Paulo Gomes Mendes Moreira

Novembro 2008.

Em memória do meu Pai.

Agradecimentos

Ao Professor Doutor Paulo José Cerqueira Gomes da Costa, orientador da presente tese, quero expressar o meu profundo agradecimento pela sua disponibilidade, incentivo, orientação, suporte e amizade.

Ao Professor Doutor António Paulo Moreira, co-orientador da presente tese, quero expressar a minha gratidão pela disponibilidade, interesse e dedicação assim como pela amizade demonstrada.

Ao colega José Gonçalves, pela amizade e companheirismo que sempre demonstrou.

À Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança pelas condições que me proporcionou.

À Luísa, sem a qual este trabalho não teria sido possível, quero expressar um agradecimento muito especial pela forma insuperável, pela paciência e pela compreensão como me apoiou nos momentos menos bons do meu trabalho.

Finalmente, um agradecimento muito especial à minha mãe pela forma incansável como me acompanhou e ajudou, e sem a qual este trabalho não teria sido possível.

"I wonder what makes us build inefficiently-shaped human robots instead of nice streamlined machines?"

"Pride, sir", said the robot.

Terry Pratchett (The Dark Side of the Sun)

Resumo

Com o objectivo de desenvolver e testar métodos inovadores, a simulação (desde que capture os aspectos essenciais da realidade), é uma excelente ferramenta que permite encurtar o tempo de desenvolvimento de software para além de não acarretar riscos em danificar o hardware, o que pode trazer custos elevados.

Com base no estudo e aperfeiçoamento de uma plataforma comercial de robô humanóide, o seu modelo e arquitectura de controlo deram origem ao desenvolvimento de um ambiente de simulação realista para robôs incluindo os humanóides. Este simulador, *SimTwo*, possui a capacidade de simular diversos tipos de robôs em simultâneo.

Depois de validado e comparado o comportamento com o robô real, o simulador está apto a desenvolver novas metodologias de controlo que posteriormente podem ser aplicadas no ambiente real.

Utilizando o *SimTwo*, foram desenvolvidos algoritmos de optimização de baixo nível tais como a minimização da aceleração, da velocidade e do consumo energético que visam o aperfeiçoamento da trajectória.

No seguimento da optimização do consumo energético, a diminuição de esforços é também um ponto-chave para aumentar a autonomia. É proposto um sistema mecânico passivo que aumenta a eficiência energética para qualquer tipo de manobra. Com base no simulador, são apresentados os resultados desta abordagem que evidenciam os ganhos na eficiência energética para o principal movimento de locomoção do robô (caminhar).

Abstract

In order to develop and test innovative methods, the simulation (as long it captures the essence of the reality), is an excellent tool that allows to reduce the development time of software without damaging the hardware that can bring high costs.

Based on the study and improvement of a commercial platform of humanoid robot, its design and control architecture led to the development of a realistic simulation environment that includes humanoid robots capability. This simulator, *SimTwo*, also has the ability to simulate several types of robots simultaneously.

Once validated and compared the behavior with real robot, the simulator is able to develop new methods of control that can be applied to the real environment.

Using the *SimTwo*, algorithms were developed to improve low level trajectories such as the minimization of acceleration, velocity and energy consumption optimization.

Following the optimization of energy consumption, the reduction of effort is also a key to autonomy increase. It is proposed a passive mechanical system that increases energy efficiency for any type of moving. Based on the simulator, the results of this approach, that shows the optimal gains for the energy efficiency, are presented for the main motion of the robot locomotion (walking).

Résumé

De façon à développer et à tester des méthodes innovateurs, le simulation (dés qu'elle capture les aspects essentiels de la réalité) est une excellente outillage qui permet raccourcir le temps de développement de software et en outre elle m'emporte pas de risques en endommager, l'hardware qui peut apporter des coûts élevés.

D'après l'étude et le perfectionnement d'une plateforme commerciale de robot humanoide, son modèle, et son architecture de contrôle ont conduit au développement d'une ambiance de simulation réaliste pour robots en incluant les humanoides.

Le simulateur, *SimTwo*, a la capacité de simuler de différents types de robots en simultané.

Une fois validé et comparé le comportement avec le robot réel, le simulateur est apte à développer de nouvelles méthodologies de contrôle que, plus tard, peuvent être appliquées dans l'environnement.

En utilisant le *SimTwo*, des algorithmes de optimisation de bas niveau ont été développés tels que la réduction de l'accélération, la vitesse et la consommation d'énergie qui visent le perfectionnement de la trajectoire.

Dans le suite de l'optimisation de la consommation d'énergie, la réduction de l'effort est aussi un point très important pour augmenter l'autonomie.

Ainsi, il est proposé un système mécanique passive qui augmente l'efficacité énergique pour n'importe quel type de manœuvre.

Basé sur le simulateur, c'est présenté pour le principal mouvement de locomotion du robot (marcher), les résultats de cette approche qui mettent en évidence les gains dans l'efficacité énergique.

Conteúdo

1 Introdução **1**

1.1 Contexto e motivação 1

1.2 Estado da arte 3

1.3 Objectivos e contribuições da tese 6

1.4 Estrutura da tese 8

2 Robô humanóide - Bioloid **10**

2.1 Introdução 10

2.2 Robôs humanóides 10

2.3 Descrição do protótipo do robô humanóide 12

2.3.1 Arquitectura geral 12

2.3.2 Configuração mecânica 15

2.3.3 Electrónica 17

2.3.4 Arquitectura de controlo 18

2.3.5 Modos de controlo dos servos AX-12 19

2.3.6 Protocolo de comunicação 23

2.4 Descrição do software: Aplicação ao nível do PC 25

2.4.1 Interface Bioloid - Simulador 26

2.5 Conclusão 27

3 Modelo de Simulação **29**

3.1 Introdução 29

3.1.1 Contexto e motivação 29

3.1.2 Simulação interactiva 31

3.2 Arquitectura do simulador 33

3.3 Modelo de simulação de corpos rígidos . . . 36
3.3.1 Linguagem de descrição da configuração mecânica . . . 39
3.4 Modelo do servo . . . 41
3.4.1 Modelo do motor DC . . . 42
3.4.2 Medição dos valores do modelo do motor . . . 44
3.4.3 Não-linearidades no modelo do motor: limitação de corrente e tensão . . . 45
3.4.4 Modelo das forças de atrito . . . 48
3.4.5 Modelo final do servo . . . 52
3.5 Controlador de baixo nível . . . 53
3.5.1 Resultado do modelo do servo com controlador . . . 54
3.6 Modelo físico: ODE - *Open Dynamics Engine* . . . 55
3.6.1 Erros numéricos na simulação . . . 56
3.6.2 Robô equivalente com uma perna . . . 60
3.6.3 Robô equivalente com meio tronco . . . 61
3.7 Visualização 3D . . . 62
3.8 Atrito entre diferentes corpos . . . 64
3.9 Editor de *Script* . . . 66
3.10 Simulação *versus* realidade . . . 67
3.11 Conclusão . . . 72

4 Optimização de trajectórias de baixo nível 73
4.1 Introdução . . . 73
4.2 Planeamento de trajectórias . . . 73
4.2.1 t_m intermédio . . . 77
4.2.2 Minimização da aceleração . . . 78
4.2.3 Minimização da velocidade . . . 80
4.2.4 Minimização da energia consumida . . . 81
4.3 Conclusão . . . 86

5 Teste e implementação dos diferentes métodos de optimização 87
5.1 Introdução . . . 87
5.2 Teste dos métodos de optimização em *Matlab* . . . 88
5.2.1 Posicionamento de t_m intermédio . . . 88

5.2.2 Minimização da aceleração 88
5.2.3 Minimização da velocidade 90
5.2.4 Minimização da energia consumida 90
5.3 Comparação dos métodos de cálculo do instante t_m 92
5.4 Teste do método optimização da aceleração mínima no simulador 95
5.5 Conclusão 96

6 Sistema passivo para aumento de eficiência do Robô 98
6.1 Introdução 98
6.2 Método proposto para optimização dos parâmetros 99
6.2.1 Joelho esquerdo 104
6.2.2 Joelho direito 105
6.2.3 Tornozelo esquerdo 107
6.2.4 Tornozelo direito 107
6.2.5 Anca esquerda 108
6.2.6 Anca direita 108
6.2.7 Tronco de equilíbrio 110
6.3 Análise dos resultados finais 110
6.4 Conclusão 111

7 Conclusões 113
7.1 Síntese e conclusões do trabalho 113
7.2 Trabalhos futuros 115

A Descrição em XML da estrutura do simulador de Humanóides 116

Referências Bibliográficas 133

Lista de Figuras

1.1 Estrutura de controlo genérica. 2
1.2 Estrutura de controlo genérica com optimização de trajectórias. . 9

2.1 Kits de robôs humanóides disponíveis no mercado: a) cortesia de i-Sobot; b) cortesia de KumoTek; c) cortesia de Tribotix; d) cortesia de LynxMotion; e) cortesia de Robophilo e f) cortesia de Robotis. 11
2.2 Robô humanóide real utilizado: *Bioloid*. 12
2.3 Placa de circuito impresso do módulo de controlo $CM-5$ do kit *Bioloid*: a) perspectiva frontal e b) perspectiva traseira. 13
2.4 Servos AX-12. 14
2.5 Gama de operação angular do servo AX-12. 15
2.6 Esboço de dois servos AX-12 acoplados ortogonalmente. 16
2.7 Modelo do robô da aplicação de controlo: a) vista frontal e b) vista traseira. 16
2.8 Fotografia do robô humanóide real com ARM acoplado. 17
2.9 Níveis da arquitectura de controlo e *hardware*. 19
2.10 Zona morta e ganhos de parametrização permitida do servo. . . . 22
2.11 Arquitectura de comunicação. 23
2.12 Esquema eléctrico da ligação de um módulo ao barramento. . . . 24
2.13 a) Screenshot da aplicação de controlo do robô real e b) fotografia do robô real. 25
2.14 Exemplo da aquisição de dados dos estados de diversos servos do Bioloid. 28

3.1 Validação da simulação. 30

3.2 Janela de controlo do simulador: a) modos de controlo, posições e ângulos das articulações; b) controlo de visualização (opções dos gráficos). 32
3.3 Arquitectura básica do simulador. 33
3.4 Arquitectura do simulador: Optimização de trajectórias. 34
3.5 Níveis de controlo do simulador. 35
3.6 Arquitectura de controlo do robô no simulador com o fluxo de informação. 35
3.7 Níveis do simulador: modelo físico. 36
3.8 Esquema de uma articulação do tipo *hinge*. 37
3.9 Esquema de uma articulação do tipo *Universal*. 38
3.10 Localização de: a) corpos e b) articulações no robô do simulador. 38
3.11 Arquitectura do simulador: Modelo do servo. 42
3.12 Esquema eléctrico do modelo de um motor DC. 43
3.13 Cálculo da constante K_s do servo. 44
3.14 Saturação das variáveis no modelo do motor. 45
3.15 Robô humanóide simulado: a) braço para baixo e b) braço erguido. 46
3.16 Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço sem limitação de binário. 46
3.17 Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço com limitação de binário sem atritos entre articulações. . . 47
3.18 Modelo da força de atrito viscosa. Função $F(\omega) = B\omega$. 48
3.19 Modelo da força de atrito estática. Função $F(\omega) = K_e sinal(\omega)$. . 49
3.20 Modelo da força de atrito final. 49
3.21 Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço com limitação de binário com atritos entre articulações. . . 50
3.22 Desvio dos ângulos do braço dos robôs real e simulado em queda. 51

3.23 Comparação de resultados do atrito para os robôs real e simulado. 51
3.24 Modelo final do servo. 52
3.25 Modelo do controlador de baixo nível. 53
3.26 Gráfico de superfície para obtenção dos ganhos K_θ^i e K_ω^i do controlador. 54
3.27 Comparação da resposta ao degrau entre braços do simulador e do Bioloid. 55
3.28 Braço de robô com uma articulação. 57
3.29 Referência e ângulo do a) robô completo e do b) modelo de um braço durante um degrau 0-90 graus. 58
3.30 Tensão aplicada ao servo do a) robô completo e do b) modelo de um braço durante um degrau 0-90 graus. 58
3.31 Corrente percorrida no servo do a) robô completo e no b) modelo de um braço durante um degrau 0-90 graus. 59
3.32 Potência consumida pelo servo do a) robô completo e pelo b) modelo de um braço durante um degrau 0-90 graus. 60
3.33 Imagem do robô simulado com uma perna. 61
3.34 Imagem do robô real com uma perna. 61
3.35 Imagem do robô simulado com duas pernas. 63
3.36 Imagem do robô real com duas pernas. 63
3.37 Robô humanóide do simulador em diversas posturas: a) em pé; b) em pé num plano inclinado; c) com braço na cabeça; d) postura de flexão; e) e f) postura de combate visto de diferentes perspectivas. 65
3.38 Deslocamento em função da força de atrito no simulador. 66
3.39 Janela do editor de *Script*. 68
3.40 Deslocamento dos pés e do robô durante um passo. 70
3.41 Ângulo do joelho dos robôs real e simulado. 71

4.1 Interpolação de θ_1 para θ_2. 74
4.2 Estado das articulações: a) Ângulo e b) Velocidade angular. . . . 75
4.3 Cálculo do ponto (t_m, ω_m) sobre a linha $L2$. 76
4.4 Minimização da aceleração. 78
4.5 Minimização da velocidade angular ω_{max}. 81

4.6 Cubo genérico para o cálculo do momento de inércia. 82
4.7 Paralelepípedo genérico para o cálculo do momento de inércia. . . 83
4.8 Valores mínimo a) e máximo b) permitidos para o instante t_m . . 85

5.1 Resultado do método t_m médio. 89
5.2 Resultado do método minimização da aceleração. 89
5.3 Resultado do método minimização da velocidade. 90
5.4 Resultado do método minimização da energia consumida. 91
5.5 Consumo de energia em função de t_m. 91
5.6 Consumo de energia em função de t_m para a nova condição. . . . 92
5.7 Resultado do método t_m médio com acelerações díspares. 94
5.8 Resultado do método de minimização da aceleração para o segundo exemplo. 94
5.9 Resultado do método optimização da aceleração aplicado ao simulador. 96

6.1 Postura do robô humanóide com as pernas flectidas. 99
6.2 Mola espiral. 100
6.3 Screenshot do robô equivalente com uma perna na posição equivalente de pé. 101
6.4 Screenshot do robô equivalente com uma perna na posição equivalente baixa. 102
6.5 Gráfico temporal com referências e ângulos dos eixos das três articulações da perna do robô. 102
6.6 Gráfico de superfície do consumo energético do joelho em função dos parâmetros da mola. 103
6.7 Gráfico de superfície do consumo energético do joelho em função dos parâmetros da mola (próximo do mínimo). 104
6.8 Gráfico de superfície do consumo energético do joelho esquerdo em função dos parâmetros da mola. 106
6.9 Gráfico de superfície do consumo energético do joelho direito em função dos parâmetros da mola. 106
6.10 Gráfico de superfície do consumo energético do tornozelo esquerdo em função dos parâmetros da mola. 107

6.11 Gráfico de superfície do consumo energético do tornozelo direito em função dos parâmetros da mola. 108
6.12 Gráfico de superfície do consumo energético da anca esquerda em função dos parâmetros da mola. 109
6.13 Gráfico de superfície do consumo energético da anca direita em função dos parâmetros da mola. 109
6.14 Gráfico de superfície do consumo energético do tronco de equilíbrio em função dos parâmetros da mola. 110
6.15 Gráfico de barras com consumos de cada articulação durante um passo com e sem elementos elásticos. 112

Lista de Tabelas

2.1 Tabela de configuração/programação dos servos AX-12 21

3.1 Posição e tamanho da estrutura do humanóide. 39
3.2 Posições das articulações do humanóide. 40
3.3 Posição e tamanho da estrutura do robô com uma perna. 60
3.4 Posições das articulações do robô com uma perna. 60
3.5 Posição e tamanho da estrutura do robô com duas pernas. 62
3.6 Posições das articulações do robô com duas pernas. 62
3.7 Deslocamento de um passo para diversas superfícies. 67
3.8 Passo: Robô real e simulador. 69

5.1 Comparação dos métodos de cálculo do instante t_m. 93
5.2 Referências de trajectórias desejadas. 95
5.3 Referências de trajectórias desejadas. 95

6.1 Ângulos de referência para baixar centro de massa do robô com uma perna. 101
6.2 Sequência de figuras do robô equivalente durante um passo. 105
6.3 Resultados da aplicação dos elementos elásticos óptimos nas articulações do simulador. 111

Capítulo 1

Introdução

Neste capítulo encontra-se apresentado o contexto e motivação para o desenvolvimento da tese, assim como os objectivos e contribuição da mesma para com a comunidade científica. No final, é apresentada a estrutura da tese.

1.1 Contexto e motivação

Uma possível solução para o problema da locomoção e interacção de um robô com um ambiente projectado para seres humanos é dotá-lo de uma forma e funcionalidades idênticas. Estes robôs humanóides têm vindo a mostrar-se cada vez mais viáveis e foco de investigação intensa. A investigação em robôs humanóides envolve várias áreas: mecânica, electrónica de potência, modelação de sistemas, controlo, locomoção, controlo de gestos, reconhecimento de padrões e inteligência artificial, interacção homem-robô, entre outros. A locomoção é, na sua essência, transportar o robô de um ponto A para um ponto B atravessando um terreno que poderá ser liso, rugoso, ondulado, irregular ou até mesmo possuir degraus. Para se mover, o robô humanóide tem de caminhar tal como acontece com os seres Humanos. Quando este está de pé, é fácil para as suas pernas suportar o tronco numa posição estável. Contudo, para este se mover horizontalmente, as pernas têm que se movimentar de acordo com uma sequência específica que origina a locomoção Xie *et al.* (2008). O movimento de caminhar num robô humanóide é uma área sob a qual um grande número investigadores se debruçam, e por conseguinte, existem vários métodos propostos na literatura como se pode observar na secção

1.2. Por outro lado, a simulação é uma excelente ferramenta que permite encurtar o tempo de desenvolvimento dos algoritmos de controlo uma vez que, cada vez mais, esta e a realidade se aproximam com a evolução dos modelos usados e a emergente capacidade de processamento disponível na computação. Mas há outras vantagens no recurso à simulação, quando esta se aproxima da realidade, uma é o facto de não haver riscos em danificar o hardware que, numa primeira fase de desenvolvimento, pode acarretar custos elevados, outra é a facilidade em reconfigurar o robot sem ser necessário um esforço de construção mecânica e finalmente a possibilidade de explorar o efeito de mudanças em parâmetros específicos que seriam muito difíceis de conseguir na prática.

Uma possível estrutura de controlo de um robô humanóide baseia-se na figura 1.1 onde o bloco planeamento de trajectórias é o responsável pela geração (em tempo real ou em modo off-line) das trajectórias que fazem o robô desempenhar um determinado movimento. No nível abaixo, encontram-se os servos cuja tarefa é controlar o movimento dos eixos de forma a que estes sigam a referência de estados. A seta que liga o bloco de controlo ao bloco de planeamento, a traço interrompido, indica o sinal de realimentação correspondente ao estado actual dos servos.

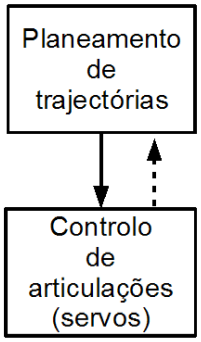


Figura 1.1: Estrutura de controlo genérica.

1.2 Estado da arte

Tem havido, nos últimos anos, um crescente aparecimento de robôs apresentando configurações complexas com um elevado número de graus de liberdade (tipicamente > 15) Neo *et al.* (2005). Os robôs testemunharam avanços importantes nas últimas décadas, desde aqueles baseados em animais, nos anos 80, até aos robôs humanóides que aparecem no virar do século Popovic *et al.* (2005). Esta é a configuração mais mítica. Se há alguns anos atrás, os robôs humanóides eram algo de ficção científica Suleiman *et al.* (2007), hoje em dia estão a tornar-se um tipo de robô comum na investigação. Um robô humanóide é um sistema que possui vários graus de liberdade Neo *et al.* (2005) com dois pés que usa para se locomover e que necessita de ser equilibrado Höhn *et al.* (2005). A presença de dois braços acrescenta graus de liberdade que podem ser usados para ajudar no equilíbrio estático e dinâmico, mas aumentam ainda mais a complexidade do sistema a controlar. Os principais progressos foram feitos a nível da construção e estabilização da locomoção. A locomoção em ambientes não estruturados implica que o robô bípede se adapte à situação presente, por exemplo evitar obstáculos e escolher o ponto de apoio. A investigação sobre o controlo deste tipo de robôs já existe há alguns anos onde Vukobratovic Vukobratovic *et al.* (1990), propôs um modelo matemático e o seu método de controlo, Zerrugh ZARRUGH & RAIXLIFFE (1979) estudou os estados das articulações num passo do robô humanóide recorrendo a dados da cinemática humana, Mcgeer McGeer (1990) apresentou a locomoção gerada por uma acção passiva da gravidade e de inércia numa rampa inclinada descendente, Roussel Roussel *et al.* (1998) propôs métodos de geração de estados das articulações com base na minimização da energia (*Energy Optimal Walking*), investigações sobre a estabilidade estática e dinâmica começaram a ser desenvolvidas por Zheng Zheng & Shen (1990) e Chevallerreau Chevallereau *et al.* (1998), respectivamente. Para assegurar a estabilidade dinâmica num robô bípede, Shin Shin *et al.* (1990) e Hirai Hirai *et al.* (1998) propuseram métodos de geração de estados das articulações baseado no ZMP - *Zero Moment Point* Vukobratovic & Juricic (1969), método já existente há alguns anos. Este método descreve a trajectória ZMP desejada com a correcção do movimento para concretizar a trajectória definida Kajita *et al.* (2006). Na realidade, nem todas as trajectórias

previstas por este método são, na prática, realizáveis Park & Kim (1998) Ferreira *et al.* (2005).

Hoje em dia, os robôs humanóides têm sido alvo de diversos projectos I&D, e têm sido apresentados protótipos de robôs muitas vezes focalizados na liga de futebol de humanóides Robocup (2008). O avanço tecnológico emergente das áreas de sensores e actuadores, comunicação, interface, mecatrónica e inteligência artificial tem encorajado cientistas e investigadores a desenvolverem robôs humanóides reais Knight (2008). Esta investigação é clara no Japão onde diversas plataformas de robôs humanóides têm sido criadas com o apoio de grandes marcas. Exemplos são o ASIMO da Honda Honda (2008), o QRIO T. Ishida (2004) e o SDR-3X da Sony Sony (2000), o HOAP-1 (Humanoid for Open Architecture Platform) da Fujitsu Limited (2001), entre outros. Nos Estados Unidos da América, universidades como o MIT investigam como os robôs podem aprender a interagir com o ambiente e com o ser Humano (European Robotics Research Network) Lee *et al.* (2008).

Na Europa, concretamente, alguns projectos estão orientados no sentido de seguirem a coordenação sensor-actuador baseado na biologia como é o caso dos projectos Brite-Euram Syneragh e o IST-FET Paloma. O projecto Baby-bot, da Universidade de Génova em Itália, baseia-se no controlo sensor-actuador semelhante à aprendizagem das crianças Knight (2008). Em Portugal, a investigação nesta área tem vindo a crescer uma vez que diferentes unidades de investigação têm apostado neste sentido Santos (2008) Teodoro (2008).

Em particular, o controlo de movimento de robôs humanóides tem vindo a ser desenvolvido com base em posições das articulações pré-definidas que originam uma sequência de movimentos Kim & Sankai (2005), como por exemplo: passo, rotação, baixar e levantar em malha aberta Buss *et al.* (2003), através de lógica Fuzzy Lei *et al.* (2005), com processamento central ou distribuído Tomokuni *et al.* (2005). Este facto deve-se à não linearidade e à elevada complexidade do sistema que não permite, por vezes, alcançar a solução em tempo real. Às posições pré-definidas, podem-se adicionar pequenos ajustes que melhoram a performance e equilíbrio do robô Huang *et al.* (2000). Há ainda autores que em vez de usarem trajectórias das articulações pré-definidas, usam regras simples na realimentação em tempo-real para controlarem o robô Silva *et al.* (2007).

Mas de um modo geral, a geração de estados das articulações tem sido baseada no controlo de um pêndulo invertido Suzuki & Ohnishi (2006) Kajita *et al.* (2002) e no ZMP (Zero Moment Point Kim *et al.* (2007) Kajita *et al.* (2006)), método utilizado pela maioria dos investigadores, por exemplo Eshghinejad & Keshmiri (2008). O modelo de um pêndulo invertido duplo consiste num pêndulo invertido que suporta um pêndulo normal. A parte superior do robô e a perna de suporte correspondem ao pêndulo invertido, enquanto que o pêndulo normal corresponde à perna que se encontra a dar o passo Wang *et al.* (2008). O método ZMP, é dos mais populares e úteis em termos de critério de estabilidade. Como princípio, se o ZMP está no interior da cobertura convexa dos pontos de apoio do robô, então o robô encontra-se estável. Ou seja, pretende-se cancelar todas as forças exercidas no robô com as forças de reacção do chão Teodoro (2007). O problema associado a este método é a quantidade de cálculos necessários que inviabiliza o tempo-real e é normalmente calculado num modo offline Xiong *et al.* (2008) Kuffner *et al.* (2003) embora em alguns casos seja possível Buschmann *et al.* (2007), após algumas simplificações.

Existe muita investigação nesta área de planeamento e geração de movimentos dos robôs humanóides, por exemplo, na optimização onde se pretende minimizar o erro entre o ZMP real e o ZMP desejado, minimizar o erro entre os ângulos das articulações e os ângulos de referência Xiong *et al.* (2008), entre outros.

Como de um modo geral, o estudo de modelos simples tende a dar uma melhor visão sobre os mecanismos subjacentes de sistemas complexos Silva & Santos (2005), investigadores baseiam-se na simplificação Chung (2008), por exemplo do modelo das pernas por massas, e do tronco por um ponto de massa. Neste caso a estabilização do tronco, que é de elevada importância nos robôs humanóides, é descartada. Um binário na anca permite alterar a posição do centro de massa, equilibrando o robô Maus *et al.* (2008).

Existe ainda investigação sobre como as pernas do robô humanóide devem ser segmentadas Chatzakos & Papadopoulos (2008).

Outro método proposto pela comunidade científica é, de um modo semelhante aos seres Humanos, ter um sistema nervoso central que cria movimentos, mesmo desligado do cérebro central. Este sistema nervoso tem como modelo osciladores não lineares de Pina Filho & Dutra (2008).

Para utilizar o método ZMP em tempo-real, torna-se necessário medir as forças que estão a cada momento a serem exercidas no robô. Para isso, a detecção do ZMP, pode ser baseada em sensores nos pés (colocação de uma matriz de pressão conforme Kagami *et al.* (2004) ou baseada em 4 interruptores estrategicamente colocados conforme Santos & Silva (2005)) ou com sensores de 6 eixos nos tornozelos, depois de uma transformação necessária Zhang *et al.* (2008).

A análise dos métodos de controlo de locomoção existentes de robôs humanóides poderá ser consultado no estudo de Bachar (2004).

Quanto à simulação, esta é, hoje em dia, indispensável para a segurança e eficiência no desenvolvimento das aplicações para a robótica. Mas, quando o ambiente de simulação tende a ser muito complexo, por exemplo, diversos robôs com diversos graus de liberdade, o tempo de processamento começa a ser um problema Huang *et al.* (2000). A simulação, que se baseia em abordagens matemáticas Casti (1998), aliada à realidade permite desenvolver rapidamente formas e métodos de controlo sem prejuízo do hardware Browning & Tryzelaar (2003). Neste contexto, o desenvolvimento de simuladores tem vindo a crescer e a aproximar-se da realidade como uma mais valia na evolução do desenvolvimento de metodologias e arquitecturas de controlo deste tipo de robôs.

1.3 Objectivos e contribuições da tese

Como objectivos principais deste trabalho, são pontos chave o desenvolvimento e apresentação de um modelo de simulação de um robô humanóide, baseado numa plataforma real, e a sua respectiva validação. Como segundo ponto, encontra-se a implementação de algoritmos de optimização de consumo energético através de escolha do método de interpolação no planeamento das trajectórias e através da implementação de um sistema passivo que contribui para o aumento da eficiência energética. Existem diversos simuladores no mercado com a capacidade de processar dados referentes a robôs humanóides tais como Simspark, Webots, MuRoSimF e Microsoft Robotics Studio, entre outros. O desenvolvimento de um novo simulador prende-se com o facto deste ser uma ferramenta personalizada e permitir assim o controlo e alteração da arquitectura internamente, como por exemplo ao nível mais baixo, tarefa que não é possível nos demais simuladores.

A implementação do modelo dos servos na simulação permite analisar formas de onda da corrente, tensão e potência associadas a cada servo, que possibilita o desenvolvimento de algoritmos sob ponto de vista da optimização da energia consumida.

Para além disso, este simulador permite criar ambientes com esteiras rolantes, robôs manipuladores, e robôs móveis omnidireccionais com os quais os robôs humanóides podem interagir. Assim torna-se possível desenvolver e testar algoritmos de controlo e optimização para problemas mais complexos. Posto isto, é possível utilizar o simulador em prototipagem de diferentes tipos e arquitecturas de robôs com a capacidade de cooperação. Uma vez validado o simulador, é possível efectuar o desenvolvimento dos algoritmos de optimização e a implementação de um sistema passivo que permite o aumento da eficiência energética, ambos apoiados no simulador permitindo assim não danificar o robô real nos estados iniciais de investigação.

Foi definida uma sequência de objectivos de forma a alcançar o desenvolvimento do simulador com a respectiva validação:

- Estudo da arquitectura física de controlo e comunicação de um robô real que servirá, posteriormente, de base ao desenvolvimento do simulador.
- Desenvolvimento do simulador que implica o conhecimento do modelo dinâmico deste. Para o efeito, é utilizado um motor de física que permite efectuar uma simulação após a construção do robô humanóide que contempla as articulações e os corpos rígidos.
- O modelo dos servos deve ser cuidadosamente desenvolvido tendo em conta o modelo dinâmico do servo real. A comparação de respostas dos servos real e simulado garante a sua validação.
- A validação do simulador no seu todo é realizada quando o robô real e o simulador são submetidos a sinais de referência iguais e o comportamento dos robôs é analisado.

As contribuições da tese estão relacionadas com novos métodos, algoritmos e estratégias de optimização de consumo e eficiência energética durante o movimento do robô. Resumidamente as contribuições estão descritas nos seguintes pontos:

- Como ponto inicial, o estudo e síntese dos trabalhos mais actuais e relevantes da área.
- Desenvolvimento de um simulador, disponível para a comunidade científica, que permite a aplicação e teste de técnicas de optimização.
- A optimização do mecanismo de interpolação no planeamento de trajectórias efectuada ao nível dos esforços dos servos e do consumo energético durante movimentos.
- Aplicação de um sistema passivo em algumas articulações que permite o aumento da eficiência energética contribuindo assim para uma maior autonomia do robô.

De uma forma resumida, a optimização pode ser analisada como um bloco colocado entre o nível de planeamento de trajectórias e o servo como ilustrado na figura 1.2.

1.4 Estrutura da tese

A tese está dividida em sete Capítulos.

No Capítulo 1 é abordado o tema da tese e as suas contribuições.

No Capítulo 2 descreve-se o robô humanóide real que serviu de base para o desenvolvimento do simulador. O robô real é inicialmente apresentado pela sua arquitectura, e seguidamente são descritos a electrónica, o controlo e o protocolo de comunicação. No final, é apresentada a aplicação de alto nível que controla o robô real.

No Capítulo 3 apresenta-se o modelo de simulação, a sua arquitectura, o modelo dos servos responsáveis por actuar as articulações, o controlo de baixo-nível, o modelo dinâmico implementado recorrendo ao ODE (*Open Dynamics*

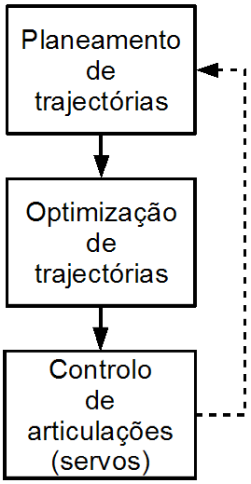


Figura 1.2: Estrutura de controlo genérica com optimização de trajectórias.

Engine), o editor de *Script* que permite compilar código durante a execução do simulador e por fim este é validado.

No Capítulo 4, os métodos de optimização das trajectórias (minimização da aceleração, da velocidade e da energia consumida) são apresentados.

No capítulo 5 são testados os métodos de optimização propostos no capítulo anterior. Por fim, a implementação de um dos métodos no simulador fecha o capítulo.

O Capítulo 6 aborda um sistema passivo que visa o aumento da eficiência energética do robô humanóide. A conclusão deste capítulo é efectuada com base nos resultados obtidos na simulação para diversos movimentos.

Finalmente no Capítulo 7 são apresentadas as conclusões e trabalhos futuros.

Capítulo 2

Robô humanóide - Bioloid

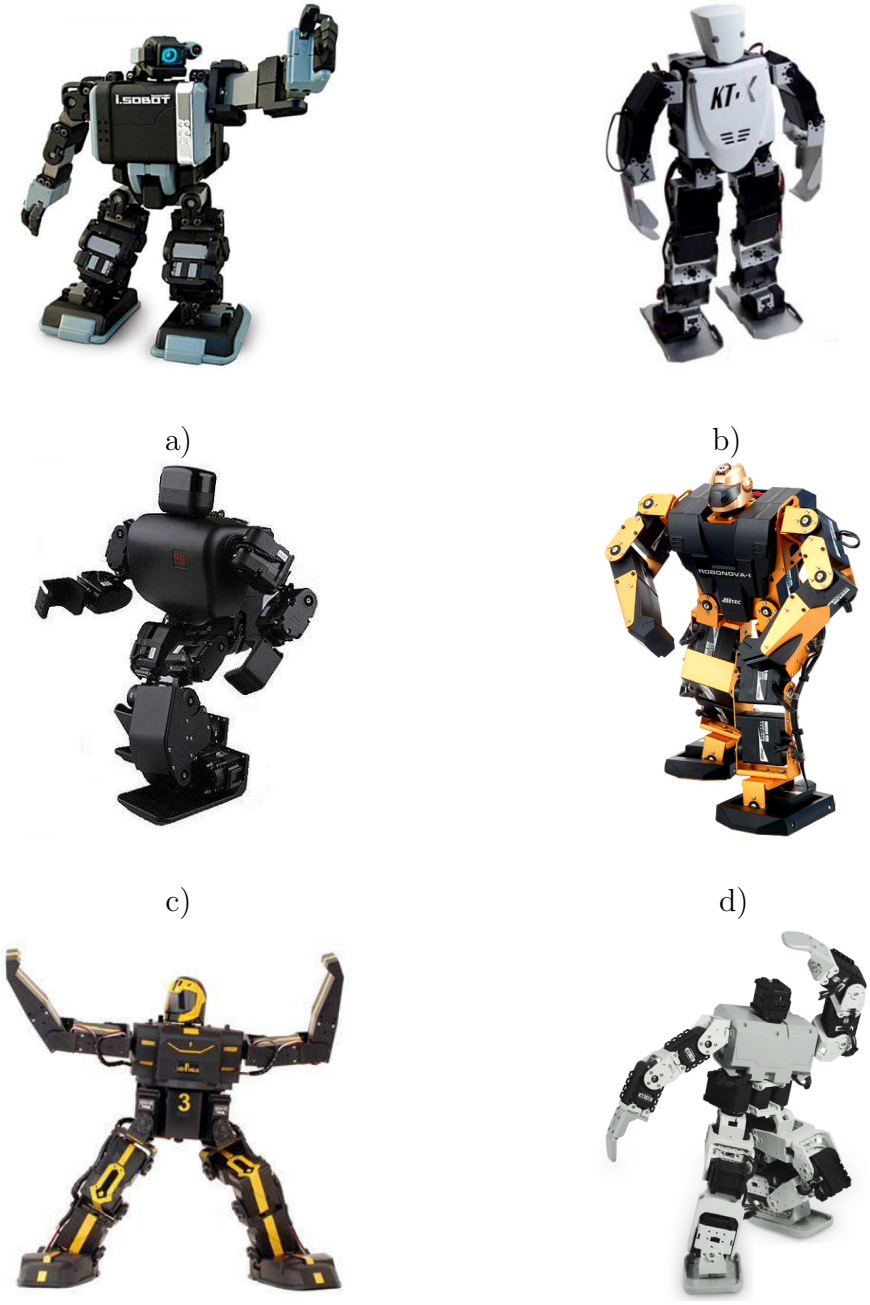
2.1 Introdução

Este capítulo apresenta a arquitectura física, o hardware e o protocolo de comunicação de um robô humanóide real, o qual serve de base para o desenvolvimento e validação do modelo de simulação desenvolvido.

2.2 Robôs humanóides

Existe uma gama diversificada de kits disponíveis no mercado com características razoavelmente idênticas. As diferenças entre eles centram-se, essencialmente, no tamanho e peso do robô, nos servos incluídos e nos módulos de controlo. As diferenças entre os servos estão relacionadas fundamentalmente com os binários disponíveis, com a quantidade de informação que estes permitem observar e alterar e com o protocolo de comunicação. Encontram-se apresentados na figura 2.1 diversos robôs disponíveis no mercado. Na figura 2.1 a) o $i-sobot$ i sobot (2008), b) o $kt-x$ Kumotek (2008), c) o *robobuilder* RoboBuilder.co. (2008), d) o Robonova Lynxmotion (2008), e) o RoboPhilo RoboPhilo (2008) e f), o Robotis Robotis (2008a) Robotis (2008b), entre outros.

A plataforma utilizada como base para o desenvolvimento do simulador foi o kit Bioloid Robotis (2008b), apresentado na figura 2.1 f). Para além de base,



a) b)

c) d)

e) f)

Figura 2.1: Kits de robôs humanóides disponíveis no mercado: a) cortesia de i-Sobot; b) cortesia de KumoTek; c) cortesia de Tribotix; d) cortesia de LynxMotion; e) cortesia de Robophilo e f) cortesia de Robotis.

este serviu também para validação do simulador no sentido em que se podem comparar respostas de ambos os robôs (real e simulado).

2.3 Descrição do protótipo do robô humanóide

2.3.1 Arquitectura geral

Este kit é composto por cerca de 19 servos, uma unidade de sensorização (com infravermelhos e microfone), uma unidade de controlo e diversas peças de plástico. A ligação entre peças é obtida a partir de alguns parafusos que apertam no plástico moldado em locais pré definidos. Desta forma, a ligação entre os servos e restantes componentes está assegurada. A sua modularidade permite ao utilizador assemblar diversos tipos de robôs desde os mais simples aos mais complexos, como é o caso do robô humanóide que se encontra apresentado na figura 2.2.

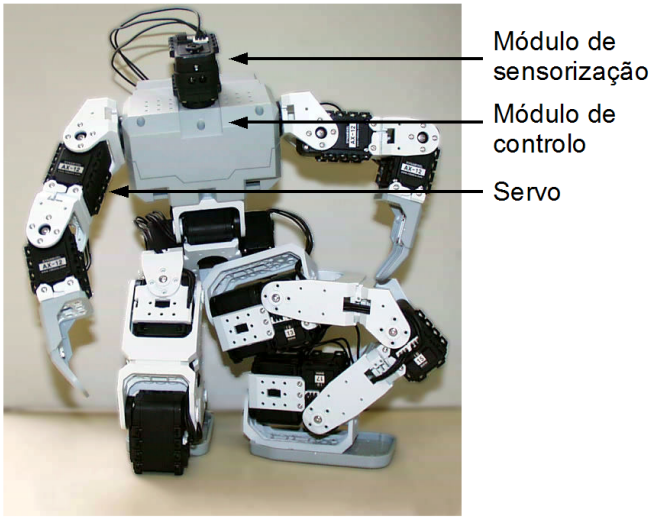
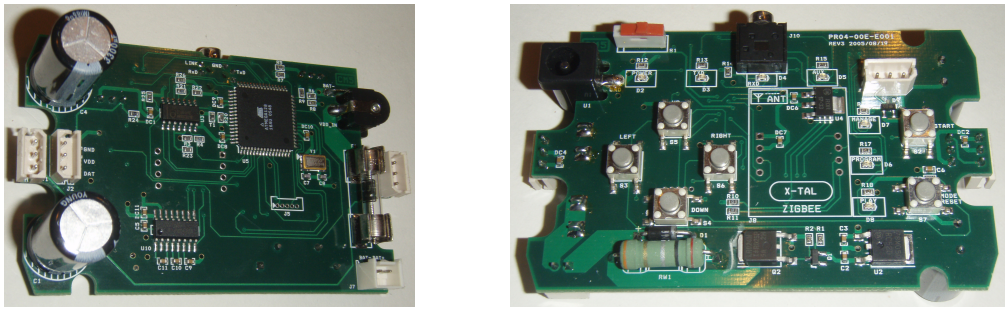


Figura 2.2: Robô humanóide real utilizado: *Bioloid*.

Esta e outras configurações são sugeridas pelos manuais do fabricante, e sendo a estrutura do robô modular, é possível introduzir as alterações desejadas, como por exemplo alterar dimensões, adicionar uma câmara, etc.

A unidade de controlo incluída no robô da Bioloid, encontra-se ilustrada na figura 2.3 e é designada por $CM-5$.



a) b)

Figura 2.3: Placa de circuito impresso do módulo de controlo $CM-5$ do kit *Bioloid*: a) perspectiva frontal e b) perspectiva traseira.

O fabricante do kit Bioloid sugere a reprogramação da unidade de controlo (firmware) para os utilizadores mais experientes fazendo desta plataforma robótica uma plataforma extremamente versátil em termos de investigação Robotis (2008b). Possui ainda uma bateria de 9.6 V incluída no módulo $CM-5$ que permite fornecer alguma autonomia ao robô.

Os servos da Bioloid (AX-12 Dynamixel), apesar do seu tamanho compacto, dispõem de um binário máximo de valor razoável (12 $kgf.cm$ que equivale a aproximadamente 1.177 $N.m$) tendo em conta o peso total do robô. Disponibilizam também a possibilidade de monitorizar a posição, velocidade, carga aplicada no motor, temperatura e tensão.

Estes servos são práticos e robustos e o protocolo digital de comunicação que eles implementam permite facilmente implementar uma arquitectura de controlo distribuída aproveitando a malha de realimentação local dos servos e a informação que eles disponibilizam para o nível de controlo superior. Esta arquitectura distribuída e hierárquica é proposta por vários autores Santos & Silva (2006) Behnke *et al.* (2005).

Para além disso, é possível ajustar parâmetros de controlo, como explicado posteriormente na secção 2.3.5, e é permitido definir limites de operação com base nos valores máximos e mínimos da corrente e temperatura através do barramento ao qual se encontram ligados, como apresentado nas secções 2.3.5 e 2.3.6.

A figura 2.4 ilustra os servos e a sua estrutura exterior em plástico que permite

a fixação de peças através de parafusos de uma forma bastante flexível. É ainda possível visualizar na mesma figura os conectores para ligação ao barramento e o eixo de rotação.

Figura 2.4: Servos AX-12.

Os servos têm um identificador (ID) guardado em cada servo numa memória não volátil ($EEPROM$). Todos estes se encontram ligados através de uma rede, como apresentado posteriormente na secção 2.3.6. Cada servo AX-12 (ilustrado na figura 2.4) incorpora uma caixa redutora, um motor DC e um sistema de controlo com capacidade de comunicação em rede. O servo AX-12 é capaz de detectar e actuar mediante condições internas, tais como sobreaquecimento, alteração da tensão de alimentação, etc. A posição e a velocidade são ambos controlados pela resolução de 1024 passos que corresponde a uma rotação de 300 graus e a uma velocidade máxima de 50 rotações por minuto (na realidade este valor depende da tensão de alimentação, e por conseguinte do estado das baterias). Na figura 2.5 encontra-se representado o ângulo coberto pelo servo. Para além disso, o AX-12 é também capaz de desligar ou ligar (binário) mediante uma temperatura limite definida, limitar valores máximos de binário, entre outros. A limitação dos valores máximos de binário foi implementada no simulador uma vez que é responsável por criar problemas no controlo.

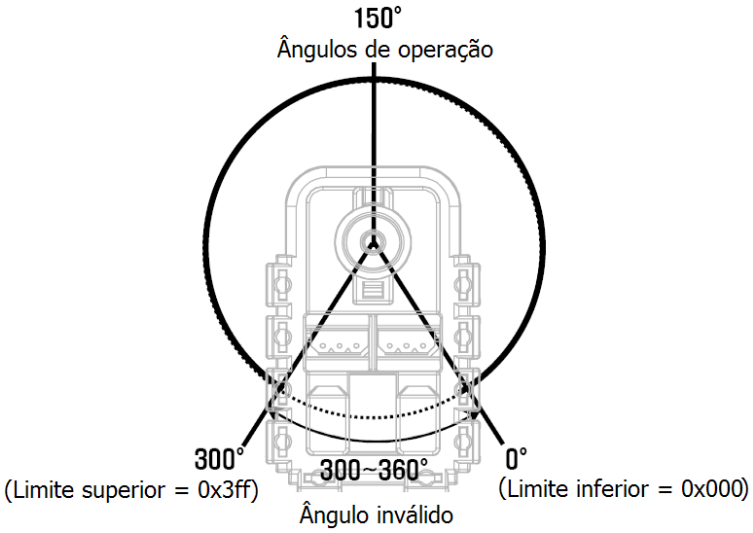


Figura 2.5: Gama de operação angular do servo AX-12.

Cortesia: Robotis.

2.3.2 Configuração mecânica

A arquitectura do robô humanóide é composta por 19 servos: 6 por perna, 3 em cada braço e um na cabeça. Há casos onde na mesma articulação se torna necessário mais do que um grau de liberdade. A solução é criar um bloco de dois servos (ortogonais) de forma a obter dois eixos de rotação como se pode verificar pela figura 2.6 que permite rotação dos dois eixos.

Três servos ortogonais são utilizados para construir os graus de liberdade da anca (para cada perna). Um servo actua na rotação de todo o membro inferior e dois ortogonais ocupam-se da abertura lateral e deslocamento para a frente e trás da perna. Dois servos ortogonais formam a articulação de 2 graus de liberdade no tornozelo. Um servo faz actuar a cabeça (futura câmara para processamento de imagem onde a localização e a detecção de obstáculos podem ser calculadas). As articulações dos ombros são baseadas em dois servos ortogonais que permitem 2 graus de liberdade em cada lado e um servo em cada cotovelo que permite obter um grau de liberdade. O peso total do robô é cerca de 2 kg e mede aproximadamente 38 cm. O modelo do robô humanóide, que está apresentado na figura

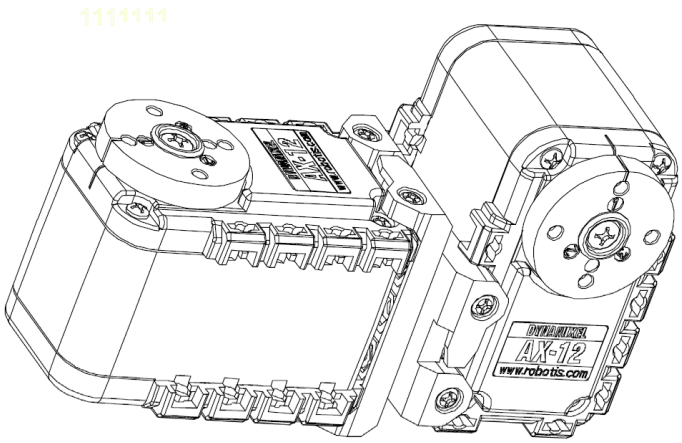
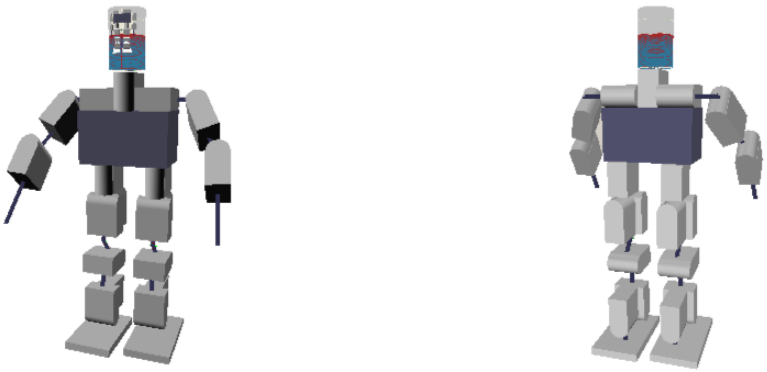


Figura 2.6: Esboço de dois servos AX-12 acoplados ortogonalmente.

Cortesia: Robotis.

2.7, serve de visualizador em tempo real da aplicação de controlo desenvolvida para o robô real Lopes (2007). Cada bloco representa um ou mais servos e as respectivas peças de conexão. Esta aplicação, apresentada na secção 2.4, permite ao utilizador controlar o robô real enquanto observa o estados das articulações e é independente do simulador Lima *et al.* (2008a).



a) b)

Figura 2.7: Modelo do robô da aplicação de controlo: a) vista frontal e b) vista traseira.

2.3.3 Electrónica

A unidade de controlo $CM-5$ é baseada no microcontrolador $ATMega128$ da família AVR da *Atmel* Atmel (2008) e pode ser facilmente programada recorrendo à linguagem C, uma vez que existe para esta família de microcontroladores um sistema livre de desenvolvimento para a linguagem C (avr-gcc). Tal como referido na secção 2.3.4, o $firmware$ incluído na versão original do kit foi alterado de modo a possibilitar a arquitectura desejada e uma maior performance (aumento da frequência de controlo) Lopes (2007). Uma vez que o fabricante disponibiliza o código fonte do $firmware$ original, o desenvolvimento e aperfeiçoamento de uma nova versão é simplificada. A versão original possibilitava um controlo básico e servia de interface com uma aplicação, também fornecida, que permitia a programação do robô. Assim, o controlo de alto nível passa a ser efectuado por um PC ou ARM embebido, com elevada capacidade de processamento e possibilidade para adicionar hardware externo (como por exemplo: câmara, acelerómetros, giroscópios, etc.). Na Figura 2.8, encontra-se uma fotografia do robô real e uma placa com um ARM embebido acoplado no tronco.

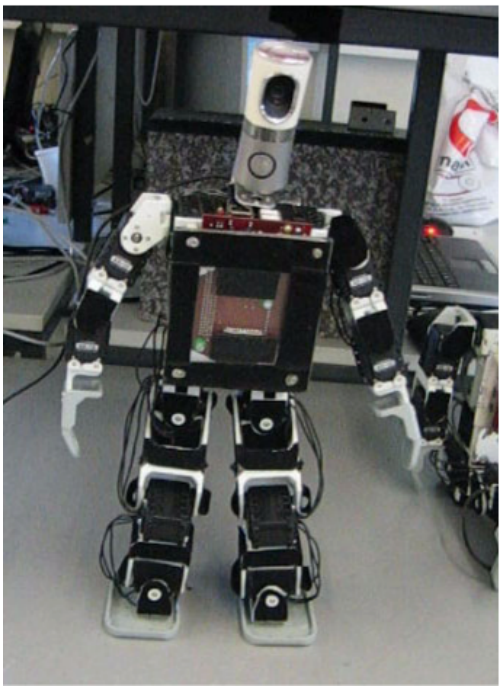


Figura 2.8: Fotografia do robô humanóide real com ARM acoplado.

O sistema de controlo e o interface de comunicação dos servos é baseado num microcontrolador ATMega8, também da Atmel Atmel (2008), o servo inclui ainda uma caixa redutora e os sensores de posição, temperatura e corrente.

2.3.4 Arquitectura de controlo

O controlo deste tipo de robôs encontra-se actualmente sob um intenso esforço de investigação (ver secção 1.2) e existem algumas arquitecturas de controlo possíveis de adoptar. De um modo resumido, algumas optam por controlar directamente o binário em cada um dos motores, noutros casos a utilização de servos remete o problema do controlo de binário para o servo esperando que este siga a referência de posição, deixando que a escolha da referência seja realizada por uma unidade a um nível superior. A arquitectura de controlo escolhida é uma arquitectura distribuída, em que o controlo de posição de cada articulação do robô é assegurado por cada servo individualmente. Os valores de referência de posição e velocidade que são enviados aos servos deverão ser determinados por uma unidade de controlo de nível superior que segue o método ZMP (Zero Moment Point) e tem pré-determinados os estados das articulações que permitem efectuar um determinado movimento. Assim, existem dois níveis de controlo: o nível mais baixo, que está implementado pelo fabricante nos servos (AX-12 da Dynamixel) e no topo encontra-se o nível de controlo mais alto que está responsável pelo planeamento da trajectória. Este nível tem como função gerar a sequência dos estados das articulações que definem um determinado movimento, por exemplo, caminhar, levantar, rodar, etc. O planeamento de trajectórias encontra-se documentado em diversa literatura (ver 1.2) e o método mais utilizado é o ZMP. Este nível está a cargo de uma aplicação que executa num PC ou numa placa embebida com um ARM. A interligação entre os dois níveis, encontra-se implementada na unidade $CM-5$.

Segundo o fabricante, no kit original a unidade de controlo implementaria também o controlo de movimentos, no entanto, a capacidade de processamento deste é bastante reduzida o que leva à solução de acrescentar um nível superior e fazer do módulo $CM-5$ apenas de interface. Esta operação acarreta a alteração do $firmware$ original dando assim a uma nova aplicação no módulo $CM-5$.

Uma vez que o fabricante do módulo de controlo $CM-5$ disponibiliza tanto o esquema eléctrico, que permite visualizar em que portas do microcontrolador estão ligados os periféricos (botões, leds e portas série), como o código fonte da aplicação, é possível alterar o *firmware* original de forma a transformar este microcontrolador num *gateway* entre os servos e o computador Lopes (2007).

O diagrama da Figura 2.9 ilustra os dois níveis utilizados.

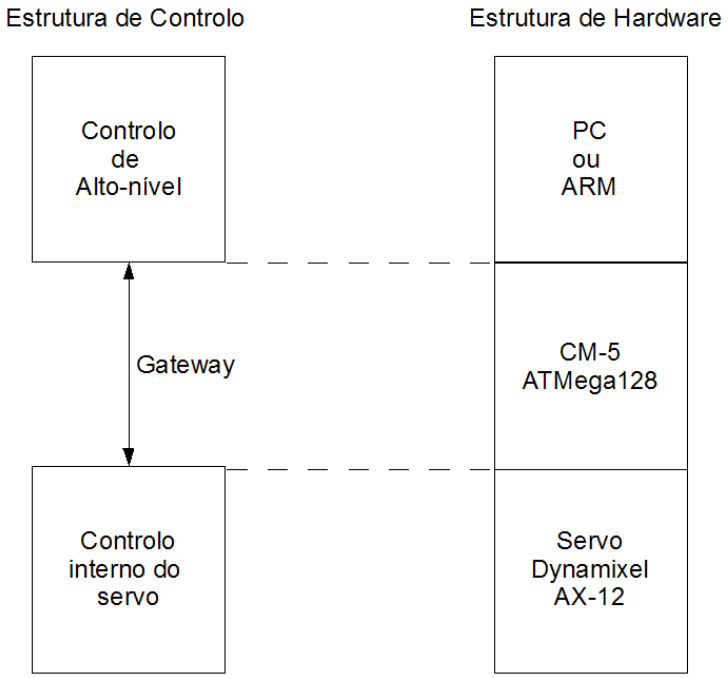


Figura 2.9: Níveis da arquitectura de controlo e *hardware*.

2.3.5 Modos de controlo dos servos AX-12

Genericamente, os servos utilizados permitem dois modos de controlo: posição e velocidade. O primeiro permite que seja indicado ao servo o ângulo desejado e este mover-se-à até esse ângulo. O segundo modo aceita uma velocidade de referência e mantém o eixo a rodar a essa velocidade. Obviamente que o modo escolhido dependerá da aplicação. No caso de um robô móvel com rodas o segundo modo é preferível, enquanto que para robôs onde o controlo de posição é importante, como é o caso de um robô humanóide, o primeiro modo é a melhor

opção. Posto isto, o modo utilizado é o controlo de posição, mais favorável à aplicação em causa. A forma de seleccionar e programar cada um dos modos é aceder a determinadas zonas de memória (endereços) que permitem efectuar a configuração do servo. Cada servo tem uma tabela de parâmetros que podem ser alterados e/ou lidos conforme ilustrado na tabela 2.1 que apresenta de uma forma resumida os parâmetros mais importantes sob ponto de vista de controlo do servo.

Esta tabela está dividida em 2 áreas: uma pertencente à EEPROM do microcontrolador interno do servo e outra pertencente à RAM. Os valores contidos na EEPROM (endereços 0 a 23), como memória não volátil, permanecem guardados e não são perdidos quando o controlador é desligado. Neste conjunto de valores encontram-se o valor do identificador ID do servo, o *baud rate* da ligação, os limites máximos e mínimos de binário, o limite máximo de temperatura e os limites de tensão aplicada para operação, entre outros. No final da presente secção é abordada a razão pela qual é vantajoso definir o binário mínimo (explicação do parâmetro E da figura 2.10). Os parâmetros contidos na RAM do microcontrolador (endereços 24 a 49) estão fundamentalmente relacionados com a operação do próprio servo: definem a resposta do controlador, os valores de posição e velocidade do servo, os valores de posição e velocidade de referência actuais, a carga aplicada e a tensão de alimentação. Desta forma, o acesso a esta tabela permite alterar os valores das variáveis que definem as referências a serem seguidas pelo servo. Para além disso, os servos utilizados permitem alterar parâmetros inerentes ao controlo. Esta propriedade torna-se útil uma vez que diferentes servos têm diferentes cargas associadas. Como se pode visualizar na figura 2.10, que representa o binário de saída de um servo em função do erro de posição, os ganhos de controlo e a zona morta podem ser configurados.

Os parâmetros B e C são os parâmetros que se encontram respectivamente nos endereços 26 (0x1A) e 27 (0x1B) da tabela 2.1 e definem qual a tolerância que o controlador deve ter em relação a um erro (*Compliance Margin*). Representam uma zona morta em função do erro que pode ser alargada ou eliminada conforme o pretendido. Numa situação em que o erro esteja a oscilar entre o valor nulo e a unidade, o controlador estará sempre a actuar e por vezes, a existência dessa margem de tolerância pode evitar um certo "nervosismo"por parte do servo. Essa

Tabela 2.1: Tabela de configuração/programação dos servos AX-12

Ender.	Descrição	N. (bytes)	Min	Max
3(0x03)	ID	1	0	253(0xfd)
4(0x04)	*Baud rate*	1	0	254(0xfe)
5(0x05)	Return delay time	1	0	254(0xfe)
6(0x06)	CW Angle limit	2	0	1023(0x3ff)
8(0x08)	CCW Angle limit	2	0	1023(0x3ff)
11(0x0B)	Highest limit temperature	1	0	150(0x96)
12(0x0C)	Lowest limit Voltage	1	50(0x32)	250(0xfa)
13(0x0D)	Highest limit Voltage	1	50(0x32)	250(0xfa)
14(0x0E)	Max torque	2	0	1023(0x3ff)
16(0x10)	Status return level	1	0	2
17(0x11)	Alarm led	1	0	127(0x7f)
18(0x12)	Alarm shutdown	1	0	127(0x7f)
19(0x13)	(Reserved)	1	0	1
24(0x18)	Torque enable	1	0	1
25(0x19)	Led	1	0	1
26(0x1A)	CW compliance margin	1	0	254(0xfe)
27(0x1B)	CCW compliance margin	1	0	254(0xfe)
28(0x1C)	CW compliance scope	1	1	254(0xfe)
29(0x1D)	CCW compliance scope	1	1	254(0xfe)
30(0x1E)	Goal position	2	0	1023(0x3ff)
32(0x20)	Moving speed	2	0	1023(0x3ff)
34(0x22)	Torque limit	2	0	1023(0x3ff)
44(0x2C)	Registered instruction	1	0	1
47(0x2F)	Lock	1	1	1
48(0x30)	Punch	2	0	1023(0x3ff)

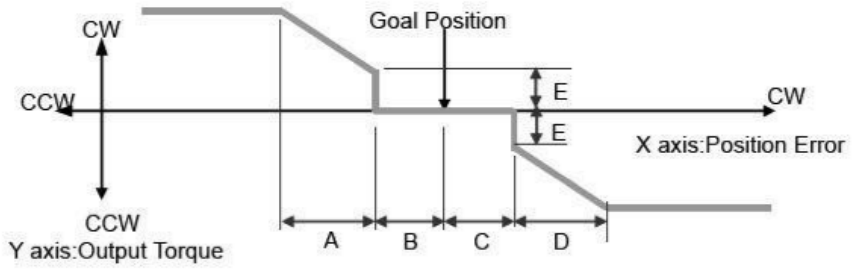


Figura 2.10: Zona morta e ganhos de parametrização permitida do servo.

Cortesia: Robotis.

margem de actuação pode ser definida com valores diferentes para os dois sentidos de movimento. Os parâmetros A e D são definidos nas posições de memória 28 (0x1C) e 29 (0x1D) respectivamente da tabela 2.1 e definem qual a taxa de evolução do binário em função do erro (*Compliance Slope*). Após o erro ter atravessado a zona morta definida pela margem de tolerância, o binário comporta-se de forma linear até atingir o valor máximo. Os parâmetros A e D vão definir o tamanho da gama de erro em que o binário irá crescer do valor mínimo até ao valor máximo, o que acaba por definir qual o declive dessa recta, ou seja qual a taxa instantânea de variação de binário. Estes parâmetros influenciam claramente qual o tipo de resposta que o servo irá ter. Uma vez que o binário se apresenta proporcional ao erro, estes parâmetros acabam por ser um ganho num simples controlador proporcional.

O parâmetro E está definido no endereço 48 (0x30) da tabela 2.1 e representa o valor mínimo de binário permitido (*Punch*). Este valor é interessante sobretudo para reduzir ruído audível durante o funcionamento do robô. Quando são pedidas correntes demasiado pequenas ao controlador, consegue-se ouvir ruído de alta-frequência provocado por comutações necessárias a originar esses valores de corrente. Para além disso, são valores de binário tão baixos que não conseguem vencer o atrito estático e portanto são inúteis estando só a consumir energia.

2.3.6 Protocolo de comunicação

A arquitectura de comunicações utilizada baseia-se na arquitectura de controlo previamente descrita. Como é possível observar na Figura 2.11 a unidade de controlo mais elevada (PC ou ARM) comunica com a unidade de controlo $CM-5$ via porta de série (RS-232) e as comunicações entre os servos e a unidade de controlo $CM-5$ estão asseguradas através de um barramento que circula pelos servos.

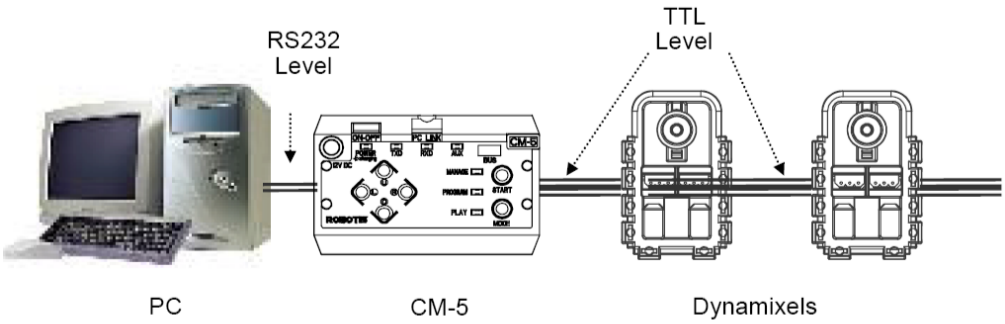


Figura 2.11: Arquitectura de comunicação.

Cortesia: Robotis.

Cada componente ligado ao barramento possui um identificador ID. Os servos podem ter ID desde 1 até 99. A unidade de controlo possui um ID = 200 enquanto o módulo de sensores possui um ID=100. Este módulo de sensorização (AX-S1), com o aspecto semelhante a um servo, é composto por um microfone, um receptor de infravermelhos, um medidor de distância por infravermelhos, um sensor de luminosidade e um besouro. Este módulo não foi utilizado uma vez que para o trabalho em causa não se tornava necessário.

A comunicação entre o PC e o controlador $CM-5$ do robô é uma simples ligação porta série Full-Duplex que funciona a um *baud rate* de 115200 *bps*. É esta ligação que acabará por ser o *bottleneck* das comunicações que irá limitar o valor máximo de velocidade de transmissão e consequentemente o valor do ciclo de controlo. A unidade $CM-5$ serve, portanto, como uma *gateway* entre a comunicação do PC para os servos. Cabe ao PC enviar os pacotes necessários

para colocar os servos em funcionamento, solicitar ou enviar dados a estes. A topologia da rede que liga os servos ao controlador $CM-5$ pode ser descrita como sendo uma topologia em barramento em que diversos nós são acrescentados em *daisy chain* Tucker (2004). O barramento passa por todos os servos, de modo que a comunicação seja efectuada através destes. Esta rede dos servos possui um Master, que é o controlador $CM-5$, o único capaz de despoletar a comunicação.

Ao nível físico, todos os servos utilizam três linhas comuns a todos eles: duas são responsáveis pela alimentação enquanto que uma outra é dedicada à comunicação de dados com níveis TTL. O tipo de ligação física que temos nesta rede é portanto multiponto. O protocolo ao nível da ligação física é um protocolo série Half-Duplex, assíncrono com 8 bits de dados e 1 *stop bit*. A velocidade de transmissão suportada e utilizada a este nível é de 1 $Mbps$. O protocolo de comunicação série Half-Duplex é um tipo de protocolo em que as funções de transmissão e recepção não podem ser usadas simultaneamente, uma vez que os pinos responsáveis pela transmissão e recepção, respectivamente TxD e RxD, se encontram ligados como ilustrado na figura 2.12.

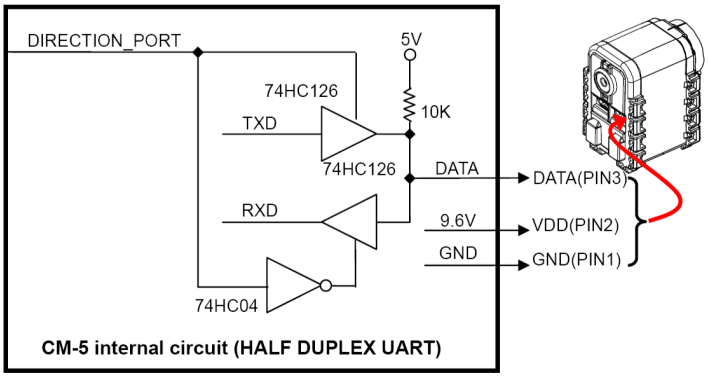


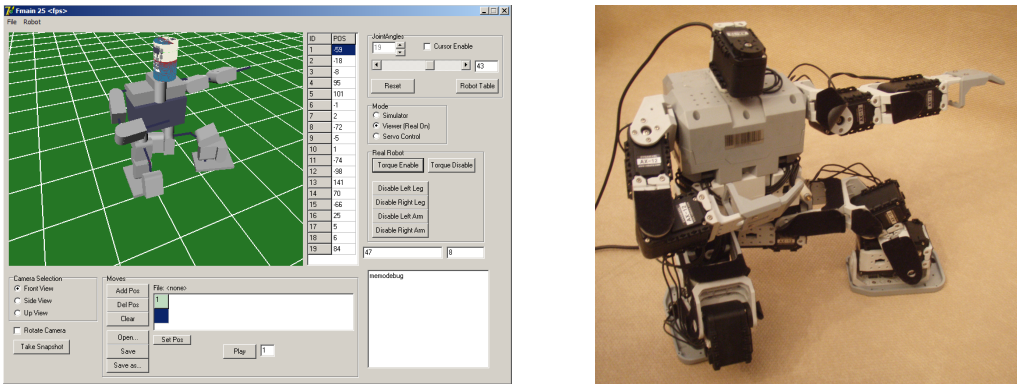
Figura 2.12: Esquema eléctrico da ligação de um módulo ao barramento.

Cortesia: Robotis.

Como há vários dispositivos ligados ao mesmo barramento, quando um dispositivo comunica os restantes têm que se encontrar em modo de escuta.

2.4 Descrição do software: Aplicação ao nível do PC

O interface entre o PC e o Bioloid é, normalmente, baseado numa aplicação gráfica para controlar e obter informações acerca do robô real Silva *et al.* (2007). A aplicação que corre ao nível mais alto (do PC) e que está encarregue de gerar as referências necessárias ao movimento do robô real (*gait pattern generator*), possui o interface gráfico apresentado na Figura 2.13 a). Para além disso, este interface inclui um modelo 3D em tempo real baseado na posição actual do robô real como se encontra ilustrado na figura 2.13 b). Desta forma, é possível visualizar e/ou controlar as articulações do robô real dando origem à sua postura.



a) b)

Figura 2.13: a) Screenshot da aplicação de controlo do robô real e b) fotografia do robô real.

A aplicação, para além de permitir o controlo do movimento do robô, através das referências dos estados de todos os seus servos, serve também de sistema de aquisição de dados de todas as variáveis das articulações Lopes (2007). Esta ferramenta torna-se útil para efectuar a análise da resposta dinâmica das articulações para posteriormente serem comparadas com a simulação. A aplicação possibilita também o armazenamento em ficheiro dos estados pré-definidos de forma a gerar sequências de movimentos e compartilhar com o simulador. Existe ainda um modo de desactivação dos binários dos servos. Neste modo, o robô permite que o

movimentem, enquanto a aplicação adquire os valores instantâneos dos ângulos de cada articulação e os guarda. Esta função é extremamente útil para obter referências e aplicá-las no simulador como explicado na secção 2.4.1.

2.4.1 Interface Bioloid - Simulador

Uma vez que o simulador necessita das referências a seguir para cada articulação num ficheiro do tipo XML, como será apresentado na secção 3.3.1, a aplicação de controlo do robô real gera-as a partir do estado de todos os servos que poderá servir de maquete. Um exemplo de uma posição estável em pé, está apresentada no excerto de código em XML a seguir transcrito, onde todos os servos se encontram parados com um ângulo de referência de 0 *graus*. Algumas articulações mais complexas, onde na realidade estão dois servos acoplados ortogonalmente, como já referido na secção 2.3.2, permitindo dois graus de liberdade em simultâneo, têm dois eixos de rotação conforme se pode visualizar na ID=8 e ID=9.

```
<jointwaypoints>
 <state ID='1' FinalTime='1'>
  <joint ID='0' theta='0' w='0'/>
  <joint ID='1' theta='0' w='0'/>
  <joint ID='2' theta='0' w='0'/>
  <joint ID='3' theta='0' w='0'/>
  <joint ID='4' theta='0' w='0'/>
  <joint ID='5' theta='0' w='0'/>
  <joint ID='6' theta='0' w='0'/>
  <joint ID='7' theta='0' w='0'/>
  <joint ID='8' axis='1' theta='0' w='0'/>
  <joint ID='8' axis='2' theta='0' w='0'/>
  <joint ID='9' axis='1' theta='0' w='0'/>
  <joint ID='9' axis='2' theta='0' w='0'/>
  <joint ID='10' theta='0' w='0'/>
  <joint ID='11' theta='0' w='0'/>
  <joint ID='12' theta='0' w='0'/>
  <joint ID='13' theta='0' w='0'/>
```

```
    <joint ID='14' theta='0' w='0'/>
    <joint ID='15' theta='0' w='0'/>
    <joint ID='16' theta='0' w='0'/>
  </state>
</jointwaypoints>
```

Desta forma é possível, quer controlar o robô real, quer obter os estados das articulações que servem para introduzir no simulador permitindo a comparação de resultados entre ambos.

Para além disso, esta aplicação permite visualizar, para cada articulação, um gráfico com os seguintes estados:

- Ângulo
- Ângulo de referência
- Velocidade
- Carga
- Tensão de alimentação
- Temperatura.

O gráfico da figura 2.14 ilustra um exemplo do ângulo e do ângulo de referência, durante o movimento de levantar, para diversas articulações do robô.

Posto isto, pode-se efectuar uma validação do simulador, comparando-o com a realidade.

2.5 Conclusão

Este capítulo apresentou o robô real Bioloid que serve de base à construção e desenvolvimento do simulador, apresentado na secção 3. Foram ilustrados diversos tipos de robôs comerciais disponíveis no mercado, e a plataforma escolhida foi apresentada sob ponto de vista da configuração mecânica, da electrónica, da arquitectura de controlo e do protocolo de comunicação entre a unidade de controlo e os servos.

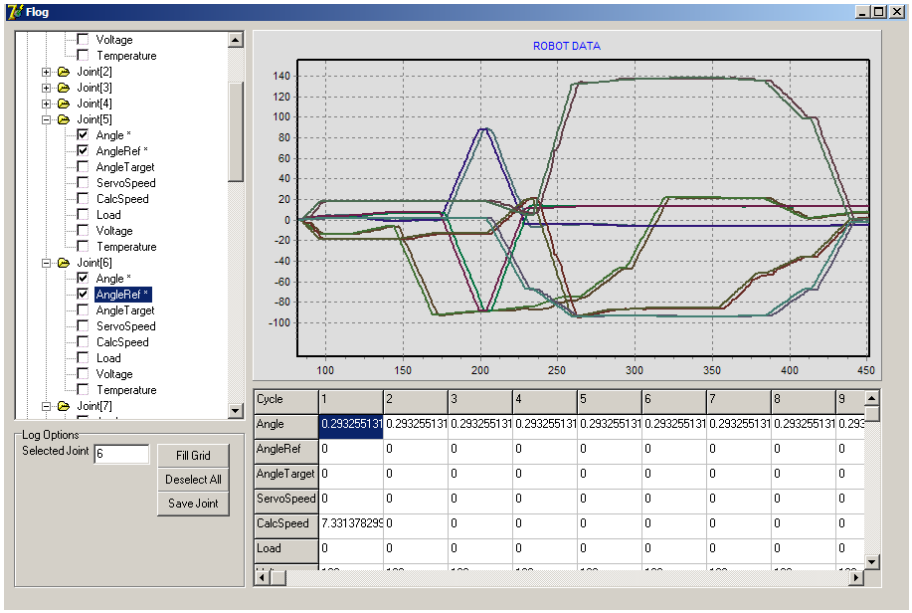


Figura 2.14: Exemplo da aquisição de dados dos estados de diversos servos do Bioloid.

Por fim, é realçada a aplicação de alto nível que controla o robô onde a visualização 3D deste em tempo real e a aquisição dos estados dos servos são ilustradas.

Capítulo 3

Modelo de Simulação

3.1 Introdução

Este capítulo apresenta o modelo de simulação desenvolvido e a sua implementação no *SimTwo* referindo o modelo de dinâmica utilizado e a construção do robô humanóide no simulador numa linguagem baseada em XML (*Extensible Markup Language*) Costa (2008). O modelo dos servos, o seu controlo (baixo nível) e a visualização 3D em tempo real é também apresentada. À semelhança do robô real, os mesmos níveis de controlo são elaborados dando ao simulador uma maior aproximação da realidade. No final, a validação do simulador é verificada.

3.1.1 Contexto e motivação

Desenvolver robôs autónomos é um processo complicado e caro devido à vasta interdisciplinaridade do problema e à necessidade de recorrer a equipamento e materiais tecnologicamente avançados e por isso caros. Normalmente, é uma combinação dos seguintes campos da engenharia: programação, electrónica, mecânica e bio-mecânica. Assim, o desenvolvimento desses robôs está reservado a grandes empresas e a universidades. Por outro lado, hoje em dia a capacidade de processamento tem vindo a aumentar e possibilita o desenvolvimento de aplicações e ferramentas de simulação. Para além da vantagem sob o ponto de vista económico da simulação face à utilização de robôs reais, o investigador pode desenvolver métodos e algoritmos sem se preocupar com o hardware, com as restrições que este

acarreta e com o prejuízo de danificar algum componente: um aspecto muito importante quando se trata de robôs que naturalmente caem Bachar (2004). Posto isto, a simulação é uma excelente ferramenta que permite acelerar o desenvolvimento de software na medida em que, cada vez mais, esta e a realidade se aproximam com a evolução de algoritmos e a emergente capacidade de processamento disponível na computação que permite processar modelos mais complexos.

Considere-se um robô genérico real R_{Real} e na simulação o mesmo sistema que tem um modelo aproximado R_{Sim}, ambos inseridos em ambientes idênticos. No ideal da simulação, deseja-se que para referências de entrada iguais se obtenham os mesmos resultados em ambos os robôs R_{Real} e R_{Sim} como ilustrado na figura 3.1. Um simulador será tanto mais fiel quanto mais próximos da realidade forem os resultados.

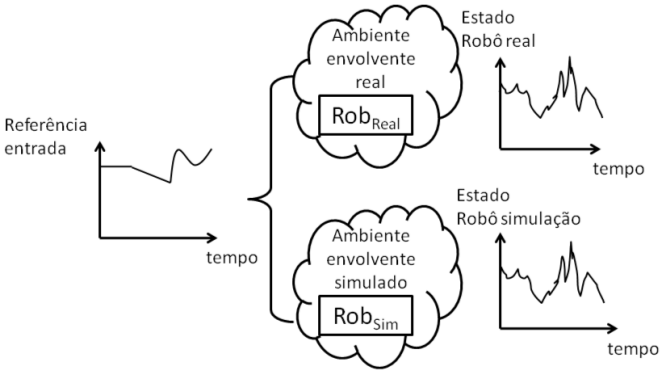


Figura 3.1: Validação da simulação.

Com uma boa simulação, consegue-se efectuar o desenvolvimento de algoritmos de controlo em simulação e transferir esses mesmos algoritmos directamente para o robô real em menos tempo e sem danificar o robô real. Contudo, desenvolver modelos 100 % fiéis com a realidade é um problema difícil Browning & Tryzelaar (2003). O simulador deverá ter as características importantes do ambiente. A simulação será tanto melhor quanto maior o número de características do ambiente real o simulador conseguir captar Lima *et al.* (2008a). No simulador desenvolvido, este aspecto foi considerado um objectivo de elevada importância.

O desenvolvimento de um novo simulador para além dos já existentes (exemplos na secção 1.3) prende-se com o facto deste ser uma ferramenta personalizada

e permitir o controlo e alteração da arquitectura internamente, como por exemplo ao nível das diferentes malhas de controlo. A visualização das formas de onda da corrente, tensão, potência, binário, posição e velocidade de cada servo é de elevada importância para o desenvolvimento de novos algoritmos e metodologias e não se encontra facilmente presente noutros simuladores disponíveis. Para além disso, este simulador permite criar ambientes com esteiras rolantes, robôs manipuladores, e robôs omnidireccionais com os quais os robôs humanóides podem interagir. Com o acesso aos vários níveis do sistema de controlo é possível desenvolver algoritmos de optimização e utilizar o simulador em prototipagem de diferentes tipos e arquitecturas de robôs com a capacidade de cooperação.

3.1.2 Simulação interactiva

Há alguns anos atrás, as ferramentas de simulação eram apenas baseadas em modo de texto. Hoje em dia, com a capacidade de processamento de placas gráficas cada vez maior, é possível criar excelentes cenários tridimensionais com texturas e mover a posição de observação cativando a atenção dos utilizadores Lima *et al.* (2006). Para além disso, a elevada capacidade de processamento permite a visualização em tempo-real e a inclusão de texturas, animações, iluminação controlada e sombras que permitem uma imediata avaliação do comportamento do sistema. O software de simulação desenvolvido permite ao utilizador, em tempo-real, obter a resposta do robô ao controlo seleccionado. É ainda possível desenvolver interfaces Homem-máquina acessível como um requisito básico de um sistema interactivo Kroumov *et al.* (2003).

O simulador tem uma janela de controlo que permite ao utilizador escolher diversas opções, entre os quais, seleccionar modos de controlo, actuar e visualizar os estados das articulações, alterar coeficientes de atrito e a frequência do passo da simulação e activar o robô que se pretende controlar, como ilustrado nas figuras 3.2 a) e b) que apresentam a janela com as diferentes abas de controlo.

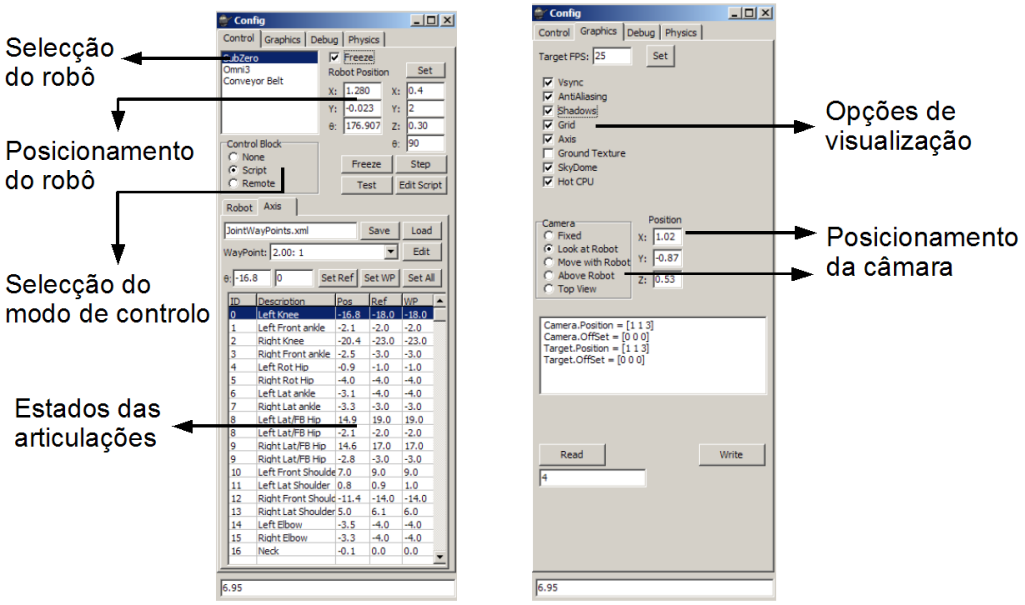


Figura 3.2: Janela de controlo do simulador: a) modos de controlo, posições e ângulos das articulações; b) controlo de visualização (opções dos gráficos).

3.2 Arquitectura do simulador

O simulador encontra-se estruturado com base no robô real. Assim, foram definidos os dois níveis de controlo idênticos a este (planeamento de trajectórias como alto nível e controlo dos servos como baixo nível), como ilustrado na figura 3.3. A simulação do comportamento dos corpos fica a cargo de um modelo físico que será apresentado na secção 3.6.

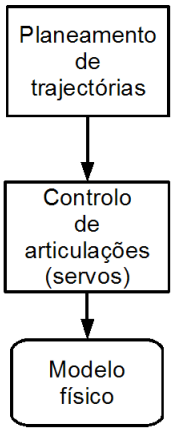


Figura 3.3: Arquitectura básica do simulador.

O bloco modelo físico simula a dinâmica do sistema e apenas fornece informação do estado das articulações (velocidades, posições e ângulos) aos níveis de controlo, como realimentação. É este o bloco responsável pela simulação e está detalhadamente descrito nas secções 3.3 e 3.6. A simulação sob ponto de vista do modelo físico é realizada com um período de 1 ms, embora seja parametrizável pelo utilizador. Para valores do passo da simulação superiores a este valor, começa a aparecer alguma instabilidade dos membros do robô, que se apresenta sob forma de oscilações indesejáveis. Para valores inferiores, a precisão aumenta mas a capacidade de processamento deixa de conseguir seguir o tempo real, mesmo com um processador actual, dado o elevado número de operações matemáticas exigidas pelo modelo físico. É o modelo físico que determina o comportamento de todos os objectos e efectua o tratamento das colisões.

Para além destes níveis, foi adicionado um outro intermédio que permite o estudo da optimização de trajectórias. Este nível, não obrigatório, é apresentado

e discutido posteriormente (secção 4) e encontra-se salientado na figura 3.4.

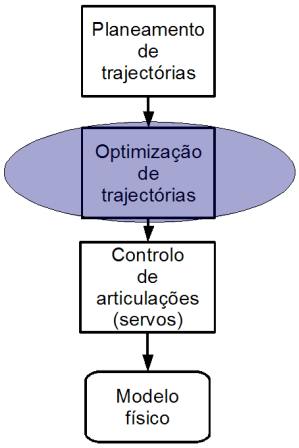


Figura 3.4: Arquitectura do simulador: Optimização de trajectórias.

A figura 3.5 ilustra, de uma forma geral, os níveis existentes no simulador assim como o trânsito de informação. Posteriormente, é apresentado cada nível: o nível mais baixo (modelo dos servos) encontra-se apresentado nas secções 3.4 e 3.5 e o bloco de optimização de trajectórias está apresentado na secção 4. O planeamento das trajectórias pode ser baseado em diversos métodos referenciados como explicado na secção 1.2. Este bloco é responsável pela geração dos estados de todas as articulações que fazem com que o robô se movimente.

No nível mais baixo, encontra-se o modelo dos servos que, sob ponto de vista da simulação, funcionam a uma frequência de 100 Hz. Esta frequência permite monitorizar, com alguma folga, a dinâmica do servo. O nível seguinte, optimização de trajectórias, permite obter uma melhoria no consumo, esforço ou velocidade do servo. O nível mais alto, planeamento de trajectórias, gera as referências para cada articulação de forma a que o robô execute os movimentos desejados. Este planeamento (controlo do robô humanóide) encontra-se descrito na literatura (ver secção 1.2).

Quanto à circulação de dados, o bloco modelo físico possui o estado de todo o mundo construído: posições, ângulos e velocidades de todos os objectos e articulações incluídos na simulação. Como o controlador de baixo-nível dos servos necessita da posição e velocidade de cada servo para actualizar a variável de actuação e o nível mais alto necessita do estado do robô e de todas as articulações

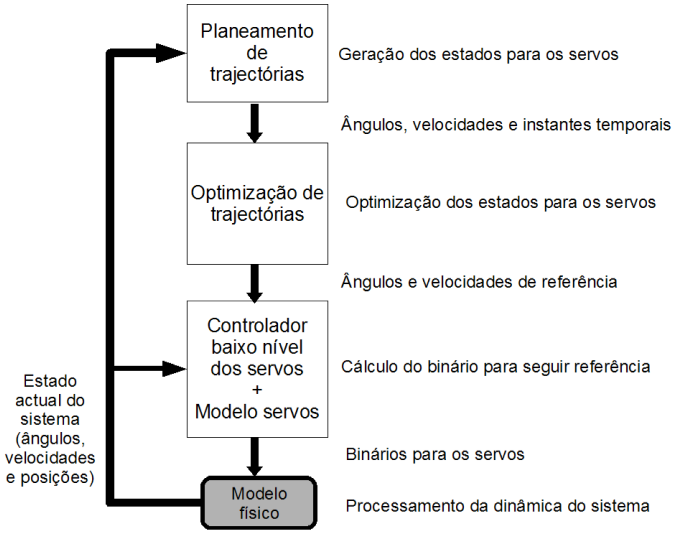


Figura 3.5: Níveis de controlo do simulador.

para planear as trajectórias, é o modelo físico que fornece todos estes dados aos blocos superiores.

De uma forma mais concreta e detalhada, a figura 3.6 representa a arquitectura de controlo de um servo i do robô implementada no simulador, onde ω^i_{ref} é a velocidade de referência e θ^i_{ref} o ângulo de referência impostos pelo planeamento de trajectórias. O ω^i e o θ^i representam a velocidade e o ângulo instantâneo estimado pelo modelo físico, o U^i_a é a alimentação de controlo do servo i e T^i_s o binário que o servo deve desenvolver.

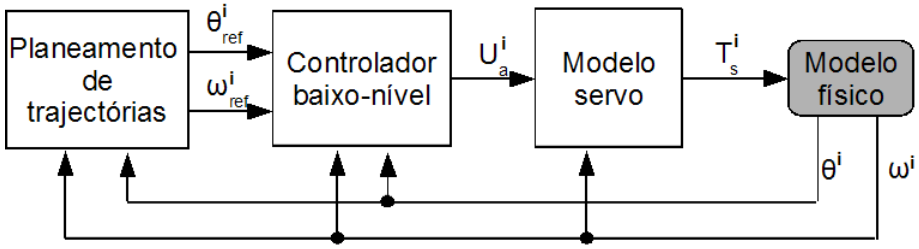


Figura 3.6: Arquitectura de controlo do robô no simulador com o fluxo de informação.

Para cada servo, é indicado ao motor de física qual o binário que este deve realizar. O motor de física devolve qual a posição e qual a velocidade angular

dependentemente da carga e posição. Este estado do servo é introduzido no planeamento de trajectórias e no controlador de baixo nível. Este último actua no motor que desenvolve um novo binário. Como o modelo do motor (secção 3.4) necessita de ter conhecimento da velocidade de rotação, usa essa informação proveniente do modelo físico.

3.3 Modelo de simulação de corpos rígidos

Este bloco, denominado como modelo físico ou modelo dinâmico e ilustrado na figura 3.7, é responsável pela dinâmica da simulação e pelo tratamento das colisões. O facto de se usar um motor de física permite que a introdução do modelo matemático possa ser evitada tornando-se apenas necessário desenhar a estrutura e as ligações do sistema a simular. O modelo matemático de um robô humanóide, sem ser simplificado, é bastante complexo e de dimensão elevada. Nesta secção é apresentada a forma como se pode construir um robô humanóide num modelo de simulação de corpos rígidos.

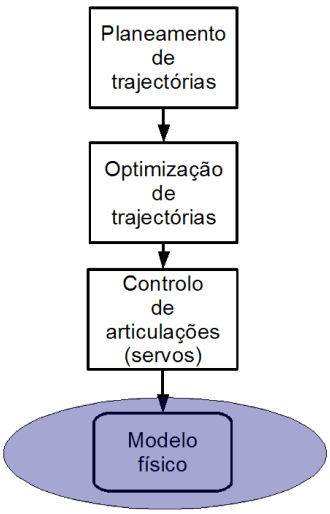


Figura 3.7: Níveis do simulador: modelo físico.

O modelo de simulação que tem de ser introduzido no motor de física, é diferente de um modelo matemático onde a dinâmica se encontra embebida. No modelo de simulação, apenas se torna necessário introduzir alguns dados tais como

a dimensão, a localização e a massa para cada objecto assim como as ligações (que funcionam como articulações do robô humanóide) entre cada objecto onde se tem que referir o seu tipo, posição e eixos da ligação. Existem diversos tipos de ligação comuns nos motores de física. Para a construção de um robô humanóide são necessárias as ligações entre objectos do tipo *hinge* e *universal* Smith (2008). O primeiro tipo permite que dois objectos se movam entre si através de um único eixo, como ilustrado na Figura 3.8.

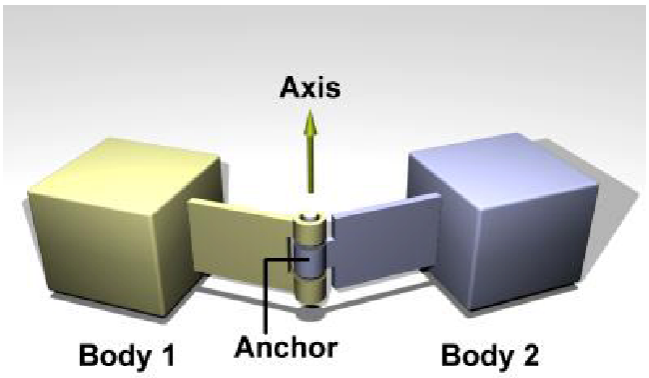


Figura 3.8: Esquema de uma articulação do tipo *hinge*.

Cortesia ODE

O segundo tipo permite que dois objectos se movam entre si através de dois eixos, como se encontra ilustrado na figura 3.9. Posto isto, um desenho dos corpos no simulador do robot humanóide encontra-se ilustrado na figura 3.10 a), enquanto que as articulações encontram-se ilustradas em b).

Assim, torna-se necessário conhecer os dados que dão origem à construção do robô humanóide e que se encontram apresentados nas tabelas 3.1 e 3.2 no motor do modelo dinâmico.

Na tabela 3.1 estão ilustrados os identificadores (ID) para cada objecto, a descrição, a massa, a posição do centro de massa e o tamanho. Cada peça tem como modelo o seu paralelepípedo envolvente. Há peças, denominadas como Ligação (Lig.) que servem apenas para ligar dois servos com uma certa distância entre si. Desta forma permite-se que o servo tenha um eixo com um braço maior o que difere de uma articulação *universal* em que os eixos têm um ponto em comum.

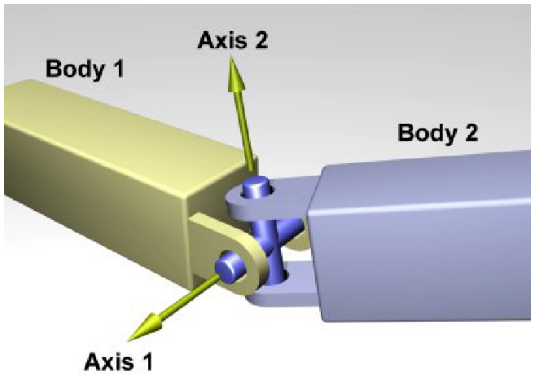
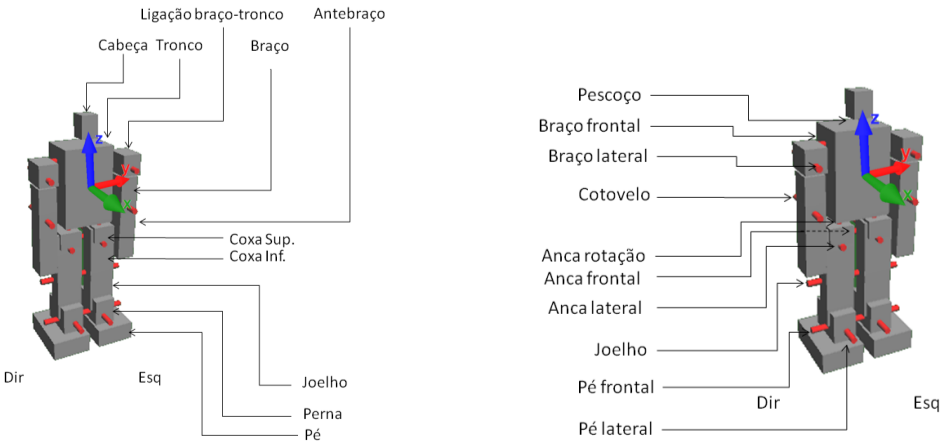


Figura 3.9: Esquema de uma articulação do tipo *Universal*.

Cortesia ODE



a) b)

Figura 3.10: Localização de: a) corpos e b) articulações no robô do simulador.

Tabela 3.1: Posição e tamanho da estrutura do humanóide.

Corpo	Descrição	Massa(kg)	Pos. CM (m)			Tamanho (m)		
ID			X	Y	Z	X	Y	Z
1	Perna esq	0.143	-0.035	0.033	-0.240	0.070	0.038	0.049
2	Pé esq	0.039	-0.035	0.045	-0.280	0.110	0.070	0.033
3	Coxa dir H	0.143	-0.035	-0.033	-0.085	0.050	0.038	0.029
4	Perna dta	0.143	-0.035	-0.033	-0.240	0.070	0.038	0.049
5	Pé dto	0.039	-0.035	-0.045	-0.280	0.110	0.070	0.033
6	Tronco	0.335	-0.040	0	0.010	0.072	0.09	0.14
7	Coxa esq L	0.173	-0.030	0.033	-0.120	0.030	0.048	0.070
8	Coxa dta L	0.173	0.030	-0.033	-0.120	0.030	0.048	0.070
11	Braço esq	0.080	-0.020	0.078	0	0.038	0.032	0.040
12	Braço dto	0.080	-0.020	-0.078	0	0.038	0.032	0.040
13	Lig. Br esq Tron	0.082	-0.020	0.078	0.043	0.050	0.034	0.038
14	Lig. Br dto Tron	0.082	-0.020	-0.078	0.043	0.050	0.034	0.038
15	Antebraço esq	0.102	-0.020	0.078	-0.074	0.037	0.032	0.126
16	Antebraço dto	0.102	-0.020	-0.078	-0.074	0.037	0.032	0.126
17	Joelho esq	0.077	-0.030	0.033	-0.200	0.029	0.047	0.08
18	Joelho dto	0.077	-0.030	-0.033	-0.200	0.029	0.047	0.08
19	Cabeça	0.100	-0.020	0	0.115	0.030	0.030	0.040
20	Coxa esq H	0.143	-0.035	0.033	-0.085	0.050	0.038	0.029

Na tabela 3.2 estão referidos os identificadores para cada articulação (ID), a sua descrição, a posição do eixo onde se dá a rotação, a respectiva direcção do eixo, os ID dos objectos que estão a ser ligados, o tipo de articulação e os limites mínimo e máximo de amplitude de cada servo de forma a que o simulador se pareça ao máximo com a realidade. Para além disso, é necessário que os servos do simulador tenham uma resposta idêntica aos do robô real. O modelo dos motores encontra-se apresentado na secção 3.4 onde coeficientes de atrito e ganhos são descritos pormenorizadamente. Quando o tipo de eixo é *Universal*, como é o caso das articulações Anca Esq Lat/FT e Anca Dta Lat/FT, torna-se necessário definir duas direcções (uma para cada eixo), assim como a respectiva posição e limites.

De forma a simplificar o processo de introdução dos valores descritos na tabela, foi desenvolvida uma interface para a linguagem de descrição XML, como apresentado na secção 3.3.1, que facilita a construção e modificação da estrutura do simulador do robô humanóide.

3.3.1 Linguagem de descrição da configuração mecânica

A utilização de um ficheiro onde estão descritas todas as configurações mecânicas (tamanhos, posições e massas inerciais entre outros) é carregado aquando da inicialização do simulador. Deste modo, torna-se possível criar e modificar, de uma

Tabela 3.2: Posições das articulações do humanóide.

Art.	Descrição	Posição eixo (m)			Direcção eixo			ID1	ID2	Tipo	Min.	Max
ID		X	Y	Z	X	Y	Z				(graus)	
0	Joelho esq	-0.039	0.030	-0.170	0	1	0	7	17	Hinge	-130	32
1	Pé esq frt	-0.027	0.030	-0.247	0	1	0	17	1	Hinge	-10	125
2	Joelho dto	-0.039	-0.030	-0.170	0	1	0	8	18	Hinge	-130	32
3	Pé dto frt	-0.027	-0.030	-0.247	0	1	0	18	4	Hinge	-10	125
4	Anca esq rot	-0.027	-0.030	-0.073	0	0	1	6	20	Hinge	-75	75
5	Anca dir rot	-0.027	0.030	-0.073	0	0	1	6	3	Hinge	-75	75
6	Pé esq lat	-0.022	0.030	-0.247	1	0	0	1	2	Hinge	-30	100
7	Pé dto lat	-0.022	-0.030	-0.247	1	0	0	4	5	Hinge	-30	100
8	Anca esq lat	-0.055	0.032	-0.103	1	0	0	20	7	Univ	-120	120
	Anca esq FT				0	1	0				-145	145
9	Anca dta lat	-0.055	-0.032	-0.103	1	0	0	3	8	Univ	-120	120
	Anca dta FT				0	1	0				-145	145
10	Braço esq frt	-0.043	0.033	0.053	0	1	0	6	13	Hinge	-120	120
11	Braço esq lat	-0.031	0.079	0.036	1	0	0	13	11	Hinge	-160	10
12	Braço dto frt	-0.043	-0.033	0.053	0	1	0	6	14	Hinge	-120	120
13	Braço dto lat	-0.031	-0.079	0.036	1	0	0	14	12	Hinge	-10	160
14	Cotovelo esq	-0.031	0.079	-0.038	1	0	0	11	15	Hinge	-90	90
15	Cotovelo dto	-0.031	-0.079	-0.038	1	0	0	12	16	Hinge	-90	90
16	Pescoço	-0.010	0	0.056	0	0	1	6	19	Hinge	-150	150

forma simples, o ou os robôs que se pretendem simular. Para isso, esse ficheiro de configuração da cena a simular baseia-se na linguagem XML. A linguagem XML (*Extensible Markup Language*) é uma linguagem geral de especificação que permite aos seus utilizadores a própria definição dos elementos. Por si só, a linguagem define uma sintaxe genérica para ser utilizada com etiquetas simples e perceptíveis com humanos Harold & Means (2004). Recorrendo a esta linguagem, o modelo físico do robô humanóide utilizado no simulador está descrito num ficheiro que serve de construtor do robô no arranque da aplicação. Para isso, posições, massas e tamanhos fazem a descrição dos ossos do robô enquanto que posições, eixos, limites de rotação e tipos fazem a descrição das articulações tal como apresentado no seguinte excerto de XML que mostra a descrição do tronco e do joelho esquerdo. No apêndice A encontra-se a descrição completa do robô. As variáveis encontram-se em unidades S.I.

```
<robot>
  <kind value='humanoid'/>
  <solids>
   <cuboid>
    <ID value='6'/>
    <pos x='-0.040' y='0' z='0.010'/>
    <size x='0.082' y='0.102' z='0.137'/>
```

```
    <mass value='0.635'/>
    <desc Pt='Tronco'/>
    <desc Eng='Trunk'/>
   </cuboid>
  </solids>
  <articulations>
   <joint>
    <ID value='0'/>
    <pos x='-0.039' y='0.030' z='-0.170'/>
    <axis x='0' y='1' z='0'/>
    <connect B1='7' B2='17'/>
    <limits Min='-98' Max='32'/>
    <type value='Hinge'/>
    <desc Pt='Joelho Esq'/>
    <desc Eng='Left Knee'/>
   </joint>
  </articulations>
 </robot>
```

Toda a estrutura do robô assim como o cenário envolvente são baseados na descrição XML e são carregados no arranque do simulador. Assim, torna-se fácil a tarefa de modificação da estrutura do robô uma vez que não é necessário compilar uma nova aplicação tornando-se assim um simulador utilizável por toda a comunidade científica para além do programador.

3.4 Modelo do servo

A filosofia de controlo distribuída pelas articulações é a implementada na maioria de robôs humanóides actuais. Nesta arquitectura, cada articulação tem o seu controlador individual e é controlada independentemente. Por isso, o estudo do modelo dinâmico do controlador apresentado baseia-se no modelo de um motor DC Kaynov *et al.* (2007). O bloco onde se encontra este modelo do motor está apresentado na figura 3.11.

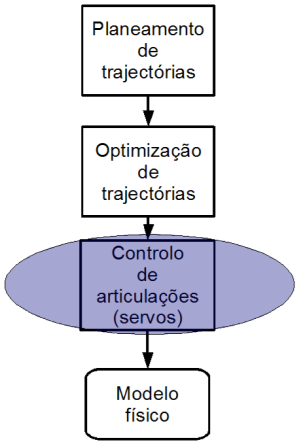


Figura 3.11: Arquitectura do simulador: Modelo do servo.

De realçar que, cada servo tem acoplado um binário de carga diferente e não constante. O binário de carga T_L e o momento de inércia J dependem da distribuição de massa, de articulação para articulação e da postura do robô. O momento de inércia afecta directamente a dinâmica do sistema enquanto que o binário de carga é uma variável não controlada que aparece como uma perturbação Kaynov *et al.* (2007).

3.4.1 Modelo do motor DC

Em cada articulação encontra-se combinada a dinâmica de um motor com uma massa inercial acoplada. A figura 3.12 ilustra o esquema eléctrico do modelo (baseado no motor DC) de uma articulação do robô humanóide com uma carga genérica.

Este modelo, alimentado por uma tensão $U_a(t)$ e percorrido por uma corrente $i(t)$, é composto por uma resistência R_a e uma indutância L_a ligadas em série com uma força contra electromotriz e (a medição destes valores encontra-se apresentada na secção 3.4.2). Por outro lado, quanto às características mecânicas do sistema, o motor actua uma carga correspondente a uma articulação que poderá ser composta por uma ou várias articulações dependentemente do membro que se encontra a modelar assim como a postura do robô. Considere-se um momento de inércia J e um binário de carga T_L como modelo da carga que o motor actua.

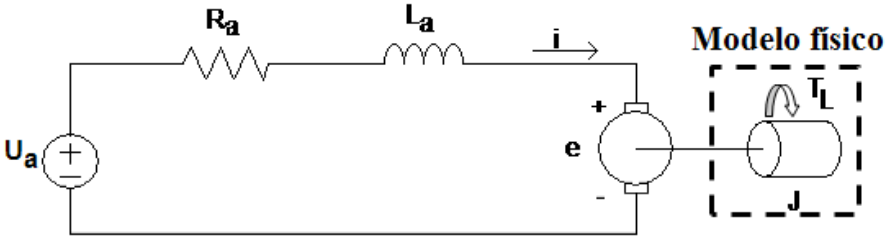


Figura 3.12: Esquema eléctrico do modelo de um motor DC.

Os cálculos da dinâmica (momento de inércia e binário de carga) estão a cargo do modelo dinâmico, denominado na figura como o bloco modelo físico. As características eléctricas do modelo e a segunda lei de $Kirchoff$ permitem escrever a equação 3.1 e estão a cargo do modelo do servo.

$$u_a = e + R_a i_a + L_a \frac{\partial i_a}{\partial t}. \tag{3.1}$$

Pela lei de $Faraday$, a força electromotriz induzida (e) é descrita pela equação 3.2 onde K_s é uma constante do motor e $\omega(t)$ a velocidade do motor (a medição da constante K_s encontra-se apresentada na secção 3.4.2).

$$e(t) = K_s \omega(t). \tag{3.2}$$

A mesma constante K_s permite relacionar a corrente percorrida $i(t)$ com o binário desenvolvido T_d através da equação 3.3.

$$T_d(t) = K_s i(t). \tag{3.3}$$

Quanto às características mecânicas do modelo, a equação de equilíbrio está apresentada na equação 3.4 onde T_J é o binário necessário para vencer o momento de inércia (equação 3.5), T_a a força de atrito resistente discutida posteriormente, e T_L o binário resistente da carga.

$$T_d = T_J + T_a + T_L. \tag{3.4}$$

$$T_J(t) = J \frac{\partial \omega(t)}{\partial t}. \tag{3.5}$$

Substituindo as equações anteriores e, neste caso, desprezando a indutância L_a, obtém-se a equação 3.6 que representa o modelo final do motor DC.

$$K_s i(t) = J\frac{\partial\omega(t)}{\partial t} + T_a(t) + T_L(t). \tag{3.6}$$

3.4.2 Medição dos valores do modelo do motor

Os valores R_a, L_a e K_s podem ser obtidos através de algumas medições directas ou indirectas no motor Bishop (2002). A resistência R_a, pode ser medida directamente obtendo-se o valor de 8 Ω, assim como a indutância L_a obtendo-se um valor de 5 mH. Quanto à constante K_s, pode-se obter este valor através de alguns ensaios experimentais. Uma vez que a força electromotriz induzida é proporcional à rotação com o ganho K_s, conforme equação 3.2, podem-se medir vários pontos de funcionamento (para diferentes velocidades em rad/s) obtendo-se o gráfico ilustrado na figura 3.13, assumindo que nesta medida a corrente é desprezável e, por isso, a queda de tensão em R_a é nula.

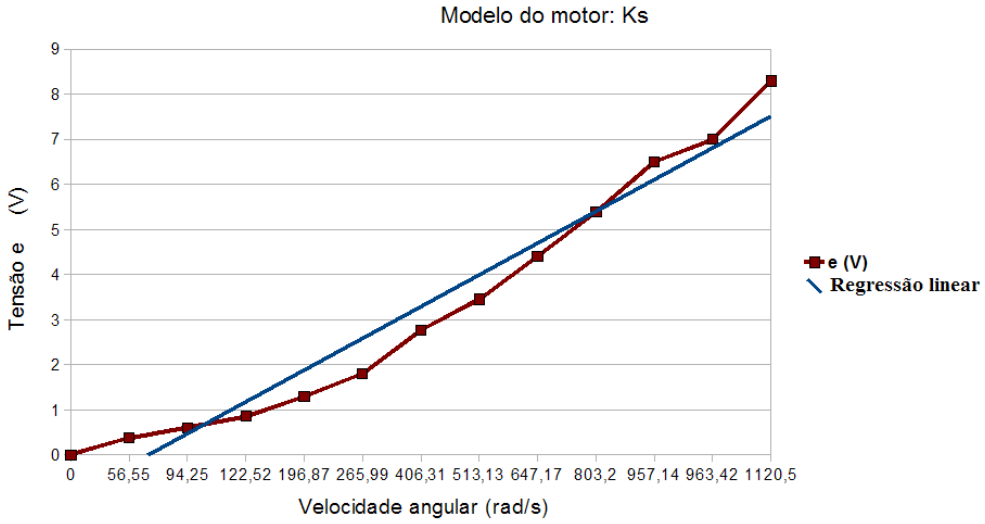


Figura 3.13: Cálculo da constante K_s do servo.

O declive da recta aproximada indica assim o valor numérico para K_s e vale 0.006705 $V.s/rad$.

3.4.3 Não-linearidades no modelo do motor: limitação de corrente e tensão

De forma a tornar o modelo do motor mais realista, onde as variáveis não podem tomar qualquer valor (principalmente valores acima de um determinado nível), ao modelo descrito anteriormente, são ainda adicionadas algumas restrições. Em primeiro lugar, é limitada a tensão de alimentação U_a e a corrente $i(t)$. Tal como na realidade, o gradiente de corrente é comandado pela indutância L_a que previne variações instantâneas de $i(t)$. Assim, a referida indutância L_a dá origem à limitação do gradiente da corrente $\frac{\partial i(t)}{\partial t}$ que já se encontra implementada na equação 3.1. Na implementação desta indutância no modelo do motor do simulador, esta foi substituída por um bloco que limita a variação $\frac{\partial i(t)}{\partial t}$. Este bloco substitui o efeito da indutância uma vez que tem como vantagem ser mais estável numericamente, especialmente quanto o período do passo da integração é muito maior que a dinâmica eléctrica. A figura 3.14, ilustra as limitações presentes no modelo do motor.

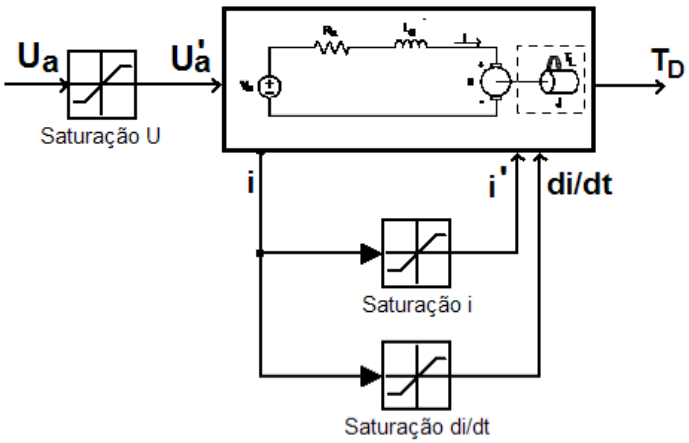
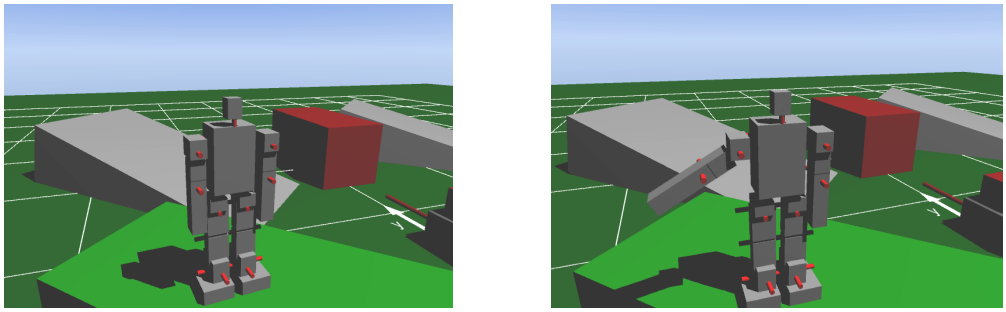


Figura 3.14: Saturação das variáveis no modelo do motor.

Como a limitação da corrente $i(t)$ acarreta directamente a limitação do binário (equação 3.3), torna-se necessário efectuar uma análise ao comportamento das articulações com este efeito.

A limitação de binário nas articulações pode, em alguns casos, provocar oscilações caso não exista força de atrito. Por exemplo, no movimento de erguer

um braço (figura 3.15 a) e b)) acontece que se o binário não for limitado, este desloca-se para o ângulo desejado (60 graus).



a) b)

Figura 3.15: Robô humanóide simulado: a) braço para baixo e b) braço erguido.

Na figura 3.16, que ilustra o gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis), é possível observar que o ângulo do braço vai de zero até 60 graus e permanece nesse ângulo até que a referência seja novamente zero.

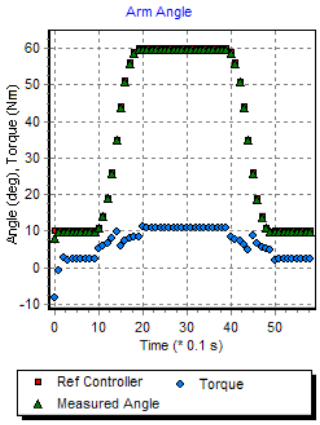


Figura 3.16: Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço sem limitação de binário.

Com a limitação de binário, acontece que nos movimentos em que o binário

necessário para seguir a referência depende do ângulo (como é o caso de levantar um braço), o sistema permanecerá a oscilar. Como se pode observar pela figura 3.17, o ângulo tende a oscilar uma vez que quando o braço se encontra a zero graus, o peso provoca uma força resistente nula, mas à medida que este se vai levantando (o ângulo aumenta) o peso começa a exercer uma força que é descrita pela equação 3.7 em que F_p é a força resultante do peso P provocado pela força da gravidade com um ângulo do braço α.

$$F_p = Psin\left(\alpha\right). \tag{3.7}$$

Há um ângulo de equilíbrio entre o binário máximo e a F_p. Mas o braço atinge esse ângulo com uma velocidade angular diferente de zero o que faz com que este o ultrapasse. A figura 3.17 ilustra a referida oscilação onde a articulação não possui força de atrito.

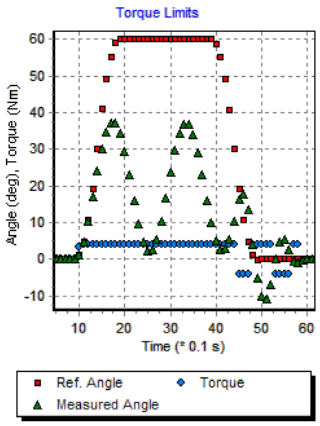


Figura 3.17: Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço com limitação de binário sem atritos entre articulações.

Na secção 3.4.4, encontra-se apresentado um modelo para as forças de atrito e no final o resultado da implementação do atrito ao problema da oscilação na articulação em causa.

3.4.4 Modelo das forças de atrito

A força de atrito resistente T_a é o resultado da soma do atrito viscoso T_{av} com o atrito estático T_{ae} como ilustrado na equação 3.8. O atrito viscoso é descrito pela equação 3.9, onde B é um coeficiente de viscosidade, e pode ser representado pela função linear ilustrada na figura 3.18. O atrito estático é descrito pela equação 3.10 (função *sinal*) e pode ser representado pela função ilustrada na figura 3.19. Estas forças de atrito encontram-se desactivadas no motor de física e estão incluídas no modelo do motor.

$$T_a = T_{av} + T_{ae}. \tag{3.8}$$

$$T_{av}(t) = B\omega(t). \tag{3.9}$$

$$T_{ae}(t) = K_e sign\left(\omega(t)\right). \tag{3.10}$$

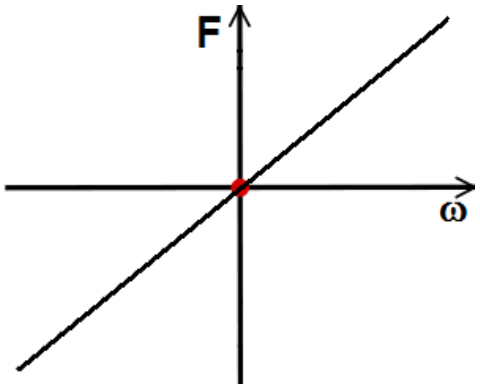


Figura 3.18: Modelo da força de atrito viscosa. Função $F(\omega) = B\omega$.

Como resultado, a força de atrito resume-se à função ilustrada na figura 3.20.

Assim, o servo exerce o binário T_d menos a soma dos atritos, como descrito na equação 3.11, onde T_s é o binário útil desenvolvido pelo servo a aplicar à carga.

$$T_s = T_d - \left(T_{av} + T_{ae}\right). \tag{3.11}$$

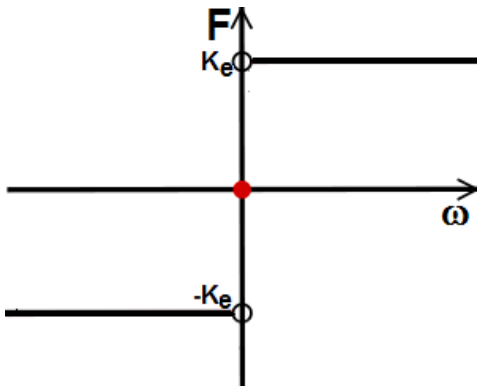


Figura 3.19: Modelo da força de atrito estática. Função $F(\omega) = K_e sinal(\omega)$.

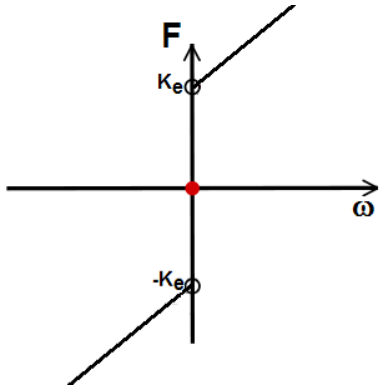


Figura 3.20: Modelo da força de atrito final.

Voltando ao exemplo da secção 3.4.3, onde o braço permanecia a oscilar com a limitação da corrente e consequentemente a limitação do binário, a aplicação de uma força de atrito faz com que o braço estabilize numa determinada posição. A figura 3.21 ilustra um gráfico temporal com o resultado.

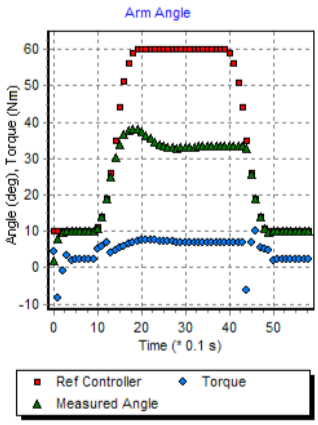


Figura 3.21: Gráfico temporal com a referência do ângulo (quadrados vermelhos), o ângulo da articulação do ombro (triângulos verdes) e o binário aplicado (círculos azuis) durante o movimento erguer o braço com limitação de binário com atritos entre articulações.

Posto isto, o atrito entre articulações deverá ser introduzido de forma a que o simulador tenha um comportamento mais semelhante com a realidade onde os atritos existem, por exemplo, no interior da caixa redutora do servo.

O cálculo das constantes de atrito poderá ser determinado com base num teste experimental. Teste esse que é composto pela libertação de um braço do robô real com um ângulo de 90 *graus* e adquirida a evolução temporal dessa articulação. Com base nesses dados, o simulador varre várias constantes de atrito até obter um erro mínimo comparado com a realidade. A figura 3.22 apresenta um gráfico de superfície com o desvio dos ângulos do braço dos robôs real e simulado.

O ponto de mínimo desvio acontece para K_e=0.000258 (no XML do simulador esta constante é denominada F_c) e para B=0.01264 (no XML do simulador esta constante é denominada B_v) com um erro de aproximadamente 11 $graus^2$. A figura 3.23 mostra a evolução temporal dos dois robôs durante a queda do braço. Como se pode verificar, ambas são bastante aproximadas.

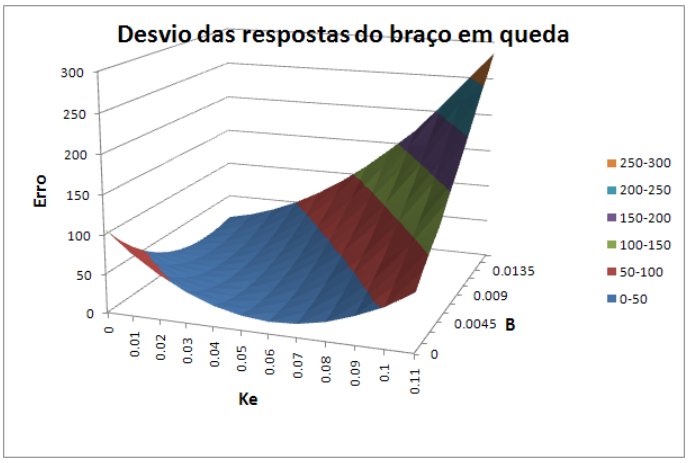


Figura 3.22: Desvio dos ângulos do braço dos robôs real e simulado em queda.

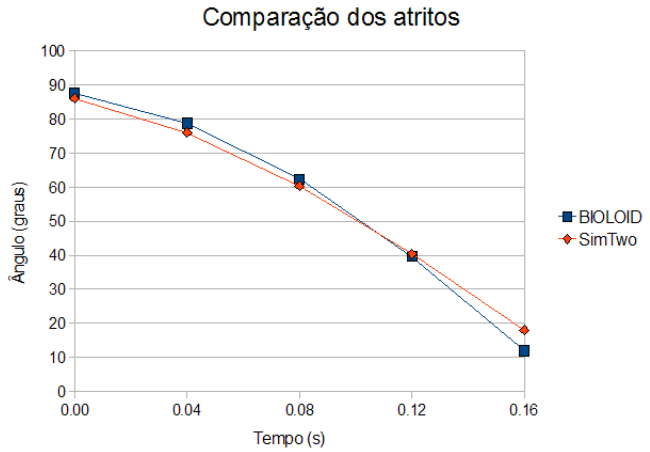


Figura 3.23: Comparação de resultados do atrito para os robôs real e simulado.

O facto de não serem incluídas as forças de atrito entre articulações do robô humanóide no modelo dinâmico (uma vez que o modelo dinâmico utilizado não o permite, pelo menos até à versão actual), obriga a que o cálculo destas tenha que ser embebido no modelo do motor. Por outro lado, as forças de atrito entre os pés e o chão estão a cargo do modelo da dinâmica e podem ser configuradas no simulador pelo utilizador, como se encontra apresentado na secção 3.8.

3.4.5 Modelo final do servo

Como modelo final, o servo recebe uma tensão como entrada e fornece um binário útil a aplicar a uma carga. Carga essa que faz parte do modelo dinâmico, onde estão contemplados o binário resistente e a inércia que dependem da articulação em causa e da postura do robô. O modelo do servo apenas necessita de ter conhecimento da velocidade e que lhe é indicada pelo modelo dinâmico. Assim, a figura 3.24 apresenta, de uma forma conclusiva, o modelo do servo que contempla o modelo do motor, o modelo das não-linearidades (saturações) e o modelo das forças de atrito.

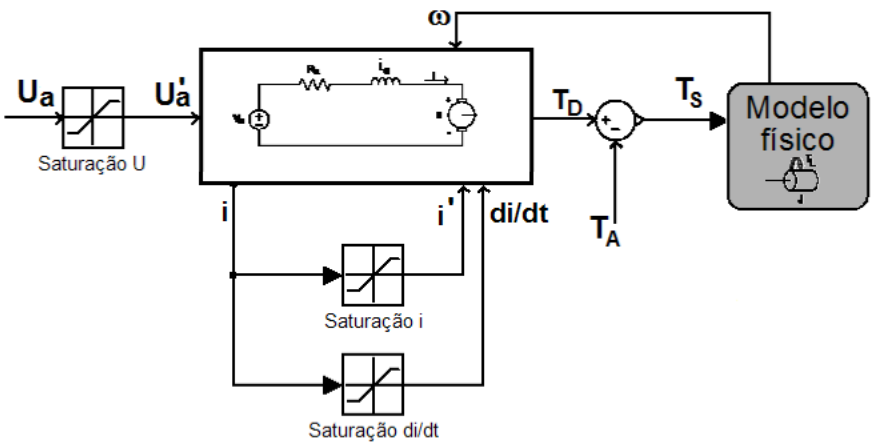


Figura 3.24: Modelo final do servo.

3.5 Controlador de baixo nível

O controlador de baixo nível do simulador assemelha-se, no caso do robô real, à malha fechada implementada internamente nos servos pelo fabricante. Assim, é necessário que este controlador receba, de um nível superior, as referências de ângulo e velocidades angulares desejadas para definir uma certa trajectória. Assumindo o ganho K_θ^i para o erro da posição e o ganho K_ω^i para o erro da velocidade angular de cada articulação i, pode-se escrever a equação 3.12 que define o sinal de controlo $U_a^i(t)$ que o servo da articulação i deve efectuar para manter as condições ângulo e velocidade angular, respectivamente θ_{ref}^i e ω_{ref}^i impostas Lima *et al.* (2008b).

$$u_a^i(t) = K_\theta^i(\theta_{ref}^i(t) - \theta^i(t)) + K_\omega^i(\omega_{ref}^i(t) - \omega^i(t)). \tag{3.12}$$

A figura 3.25 ilustra este controlador como um diagrama de blocos.

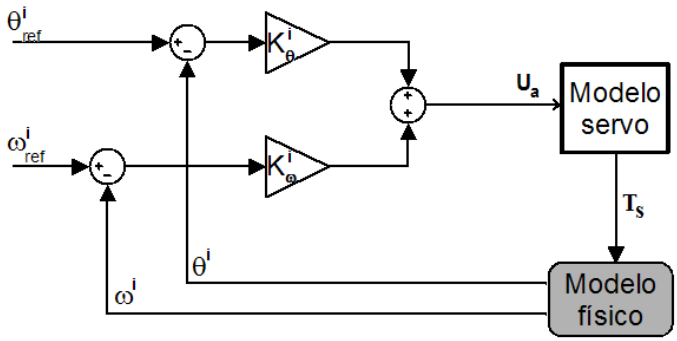


Figura 3.25: Modelo do controlador de baixo nível.

Os sinais θ_{ref}^i e ω_{ref}^i são indicados pelo nível superior (planeamento de trajectórias ou optimização de trajectórias, caso esteja implementado), os sinais θ^i e ω^i são indicados pelo motor de física como sendo o estado actual dos servos e o sinal de controlo U_a^i é introduzido no modelo do servo i que por sua vez introduz um binário no modelo dinâmico com a diferença da força de atrito. O modelo do servo, para resposta em velocidade, encontra-se apresentado na literatura, tais como Teodoro (2007) e Lopes (2007).

Os ganhos K_θ^i e K_ω^i podem ser obtidos através da execução de diversas simulações para um varrimento de ganhos calculando o erro (soma dos quadrados dos desvios) entre a resposta do simulador e do robô real. A figura 3.26 apresenta o gráfico de superfície para as simulações referidas onde o ponto de mínimo erro (30 para K_θ^i e -1.2 para K_ω^i) é o eleito para os ganhos do controlador com um erro mínimo de 0.66 $graus^2$. A resposta deste controlador e a comparação com o robô real encontra-se apresentada na secção 3.5.1.

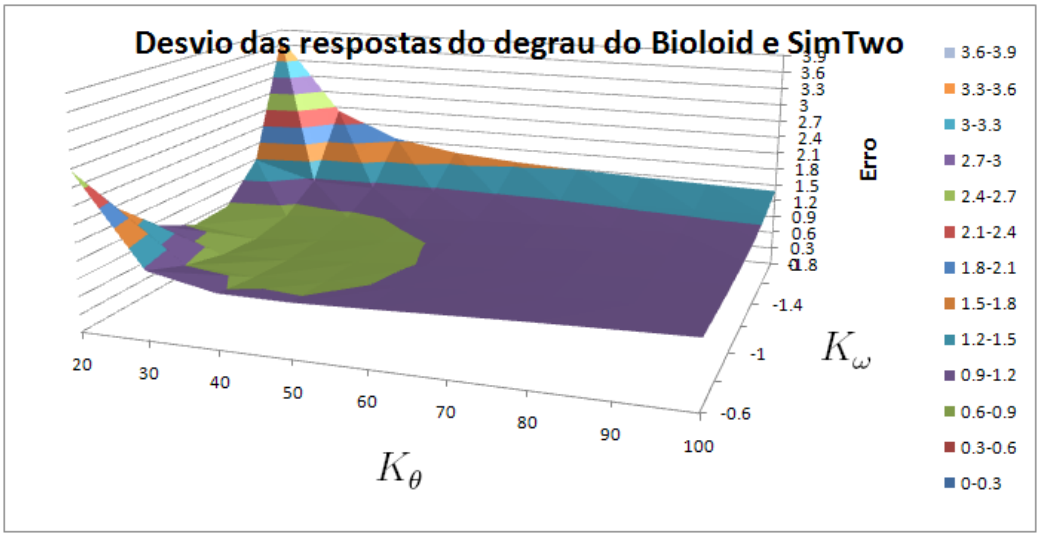


Figura 3.26: Gráfico de superfície para obtenção dos ganhos K_θ^i e K_ω^i do controlador.

3.5.1 Resultado do modelo do servo com controlador

Como resultado do modelo implementado no simulador, a mesma referência (degrau de 0 a 90 graus) é aplicada nos robôs simulado e Bioloid. O teste foi efectuado na articulação de levantar um braço lateralmente (com a respectiva carga inercial da massa do braço) e o resultado está apresentado no gráfico temporal ilustrado pela figura 3.27, onde a referência é o sinal indicado por losangos cor de laranja, a resposta do simulador está indicada por quadrados azuis e a resposta do Bioloid está indicada por triângulos amarelos.

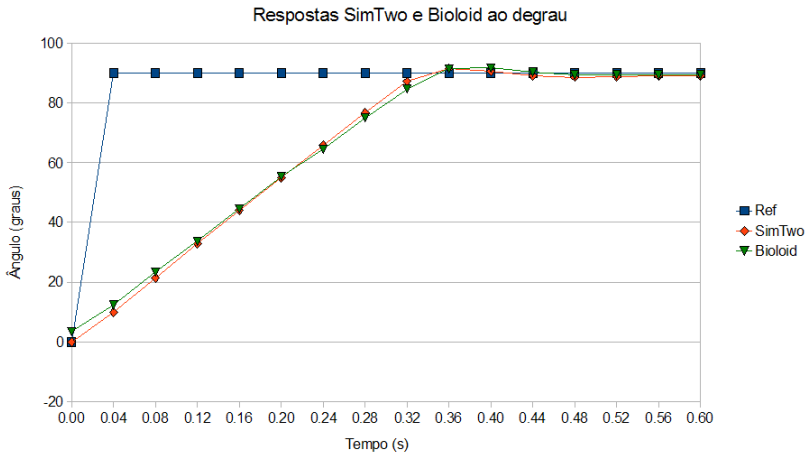


Figura 3.27: Comparação da resposta ao degrau entre braços do simulador e do Bioloid.

Como análise do gráfico temporal, pode-se constatar que as respostas são coincidentes validando assim o modelo dos servos e respectivo controlador implementados no simulador. Como confirmação adicional, e segundo dados do fabricante do servo, este atinge uma velocidade (quando alimentado a 10 $Volt$) expressa por 0.196 $segundos/60\ graus$, o que equivale a 5.34 rad/s, ou seja 0.3 $segundos$ para percorrer um ângulo de 90 graus, valores coincidentes com os dados da aquisição.

3.6 Modelo físico: ODE - *Open Dynamics Engine*

Desenhar comportamentos reais sem hardware é hoje em dia possível recorrendo a um motor de física implementado num simulador. O motor de física é o ponto chave para tornar a simulação com utilidade em termos de controlo de robôs de alta performance. Contudo, existe um largo número de motores de física que apoiam a simulação: desde comerciais, gratuitos ou até mesmo $open-source$. A maior parte deles, dá ênfase à produção de código rápido que é óptima para jogos de computador. Estes simuladores são rápidos a processar mas produzem movimentos visualmente agradáveis em vez de serem precisos. Por outro lado, existem diversos motores de física para simulação de corpos rígidos que não permitem as suas interacções mecânicas Browning & Tryzelaar (2003). Para uma simulação em tempo-real deve ser usado um motor de física rápido e preciso. O

ODE - Open Dynamics Engine garante esses requisitos. Desenvolvido por Russell Smith Smith (2008) como um projecto de motor de física de corpos rígidos *opensource*, encontra-se num estado de maturidade tal que o código produzido é estável. É utilizado hoje em dia em mais de uma centena de projectos como uma ferramenta na investigação ou no desenvolvimento de jogos de computador fornecendo comportamentos reais.

Na realidade, o ODE é essencialmente uma biblioteca de funções de simulação de alto nível que suporta o movimento e ligação de corpos rígidos, inércia rotacional e tratamento de colisões. Assim, apenas se torna necessário construir o mundo que se pretende simular. Uma ferramenta que permite o estudo do comportamento de sistemas é o *Matlab*. Esta é uma ferramenta potente de cálculo nas áreas da Engenharia que pode simular diversos tipos de sistemas Lima *et al.* (2006), mas o modelo matemático tem que ser conhecido e em alguns casos este pode ser bastante complexo como é o presente caso de um robô humanóide. O facto de ser usado um motor de física, a introdução do modelo matemático pode ser evitado tornando-se apenas necessário desenhar a estrutura e as ligações do sistema.

3.6.1 Erros numéricos na simulação

A forma como o modelo de dinâmica garante (dentro de certos limites) que dois corpos ligados através de uma articulação se mantém unidos baseia-se na aplicação de forças e binários calculados internamente. À frequência do passo da simulação, as forças que são aplicadas a cada articulação são calculadas Smith (2008). Se existe um corpo fixo A e um outro corpo B que se encontra ligado ao primeiro através de uma articulação, as forças que garantem a conexão dos corpos apenas são aplicadas no corpo B. No caso de ter uma articulação que liga um corpo C ao corpo B que por sua vez está ligado ao corpo A, levanta a necessidade de calcular forças e binários para os dois corpos B e C. Claro que ao aplicar as forças no corpo B este se vai mover, mas ao aplicar a força ao corpo C vai alterar o corpo B que terá de ser corrigido no passo seguinte da simulação. No caso de um robô humanóide, e supondo que os pés estão fixos, cada corpo poderá depender de várias articulações, como por exemplo os antebraços que estão

ligados ao braços, que por sua vez estão ligados ao tronco, que por sua vez está ligado à anca, esta liga à perna, ao joelho, tornozelo e finalmente ao pé. Todas as articulações provocam um erro de simulação que no final não é desprezável e que na prática vai depender do passo da simulação. Claro que o passo não pode ser tão pequeno o quanto seria desejável, uma vez que a capacidade de processamento é limitada e pretende-se seguir o tempo real. Para um robô humanóide com 19 articulações (mais o respectivo controlador para cada articulação), um período de passo de 0.5 ms é o limite para um processador de 2.2 GHz. Na figura 3.28, encontra-se o modelo de um braço de um robô (sem cotovelo) com apenas uma articulação que, para além de necessitar de menor capacidade de processamento, possui menos erros de simulação e oscilações indesejáveis nas formas de onda.

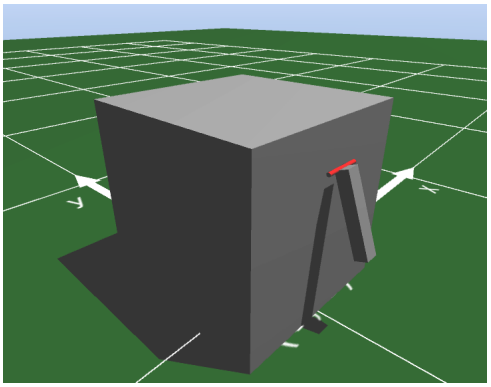
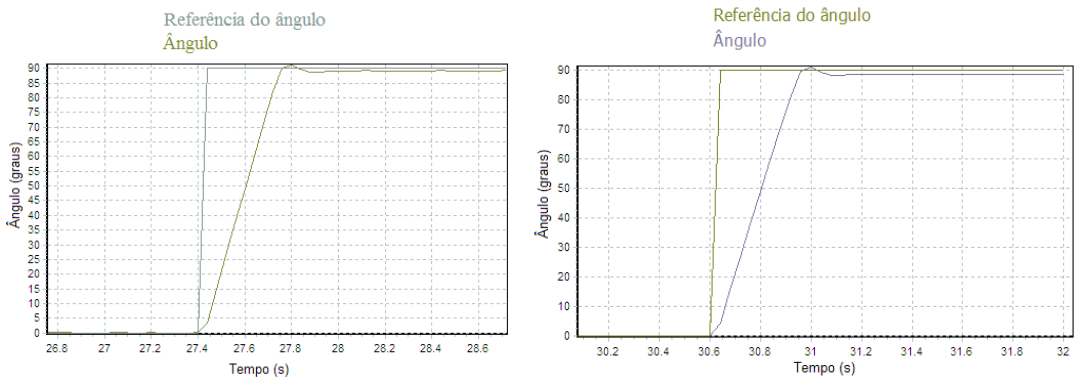


Figura 3.28: Braço de robô com uma articulação.

Para o teste realizado, foi introduzido um degrau na referência do ângulo de zero para noventa graus, como se pode verificar nas figuras 3.29 a) e b), respectivamente para o modelo de uma articulação e robô completo. As respostas são semelhantes.

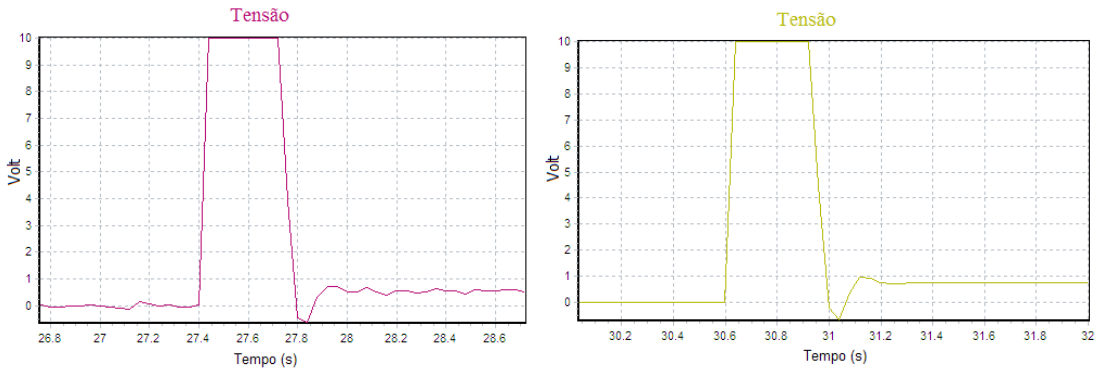
Nas figuras 3.30 a) e b) é possível visualizar que a forma de onda da tensão aplicada aos servos do robô completo e do modelo com uma articulação apenas, durante o mesmo período de tempo, são semelhantes.

É a corrente que mais difere entre o robô completo e o modelo de uma articulação. Como se pode verificar pelas figuras 3.31 a) e b), que ilustram respectivamente a corrente percorrida no servo do robô completo e no modelo de uma



a) b)

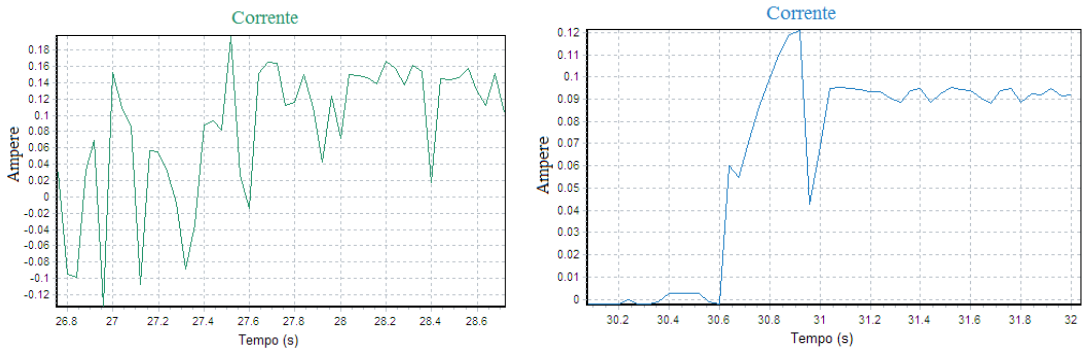
Figura 3.29: Referência e ângulo do a) robô completo e do b) modelo de um braço durante um degrau 0-90 graus.



a) b)

Figura 3.30: Tensão aplicada ao servo do a) robô completo e do b) modelo de um braço durante um degrau 0-90 graus.

articulação, existe uma discrepância na forma de onda, embora os valores sejam aproximadamente iguais.



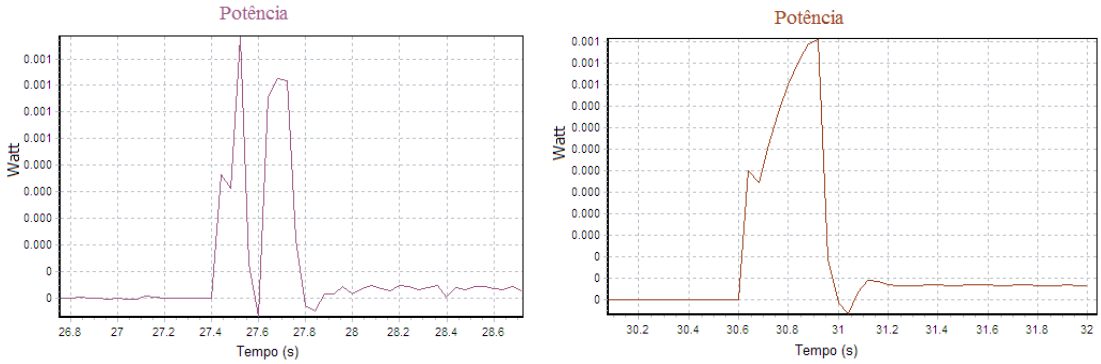
a) b)

Figura 3.31: Corrente percorrida no servo do a) robô completo e no b) modelo de um braço durante um degrau 0-90 graus.

Neste caso, o erro de simulação no modelo simplificado é bastante menor do que o do simulador completo. Aqui a corrente apresenta um *ripple* de aproximadamente 0.007 A, enquanto que no simulador completo do robô humanóide este *ripple* é drasticamente aumentado para um valor de aproximadamente 0.12 A. O consumo energético do servo do robô real responsável por manter o braço levantado é de aproximadamente 0.1 A, valor muito aproximado da simulação.

Para finalizar, a forma de onda da potência consumida nos dois robôs é também apresentada nas figuras 3.32. Uma vez que esta é o produto da tensão pela corrente, uma vez que esta última é bastante diferente, a potência também o será.

Desta forma, para algumas simulações onde se pretendam visualizar com algum rigor as formas de onda dos motores, é preferível optar por um modelo simplificado sob o ponto de vista do número de articulações. É o preço a pagar por um simulador que tem internamente o modelo completo dos motores e que permite a visualização de formas de onda da corrente, tensão, potência e binário. Mediante a simulação, foram desenvolvidos modelos simplificados de robôs que mais se adequam caso a caso minimizando o número de articulações. Nas secções



a) b)

Figura 3.32: Potência consumida pelo servo do a) robô completo e pelo b) modelo de um braço durante um degrau 0-90 graus.

3.6.2 e 3.6.3, apresentam-se os modelos simplificados que foram desenvolvidos para cada caso.

3.6.2 Robô equivalente com uma perna

Para o estudo de consumos energéticos do ponto de vista das articulações frontais (as responsáveis pela subida e descida do centro de massa do robô), foi desenvolvido um robô na simulação, que é apenas constituído por uma perna e um tronco, conforme se apresenta a estrutura na tabela 3.3 e as articulações na tabela 3.4.

Tabela 3.3: Posição e tamanho da estrutura do robô com uma perna.

Corpo	Descrição	Massa(kg)	Pos. CM (m)			Tamanho (m)		
ID			X	Y	Z	X	Y	Z
2	Pé	0.139	-0.035	0.0	-0.280	0.45	0.20	0.033
6	Tronco	0.635	-0.040	0	-0.05	0.082	0.099	0.1
7	Coxa	0.173	-0.030	0.0	-0.120	0.030	0.048	0.090
17	Joelho	0.077	-0.030	0.0	-0.210	0.049	0.047	0.101

Tabela 3.4: Posições das articulações do robô com uma perna.

Art.	Descrição	Posição eixo (m)			Direcção eixo			ID1	ID2	Tipo	Min.	Max
ID		X	Y	Z	X	Y	Z				(graus)	
0	Joelho	-0.039	0.0	-0.170	0	1	0	7	17	Hinge	-98	32
1	Pé frt	-0.027	0.0	-0.26	0	1	0	17	2	Hinge	-10	125
8	Tronco frt	-0.03	0.0	-0.08	0	1	0	6	7	Hinge	-75	75

Assim, foram retiradas as articulações laterais das pernas restando apenas as articulações tornozelo frontal, joelho e anca frontal. Nas figuras 3.33 e 3.34 encontram-se ilustrados o robô no simulador e o robô real, respectivamente. Na secção 6.2 é apresentada a sua utilidade e são discutidos os resultados.

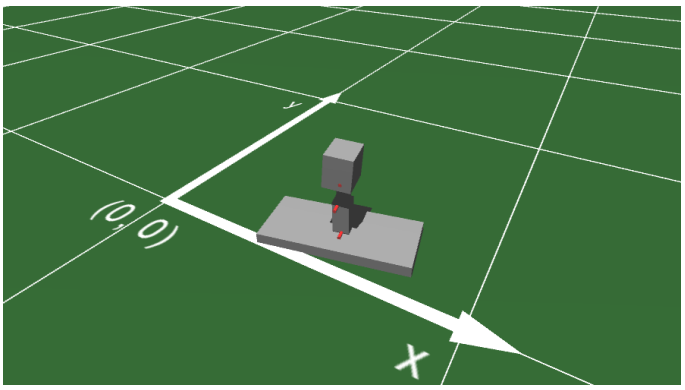


Figura 3.33: Imagem do robô simulado com uma perna.

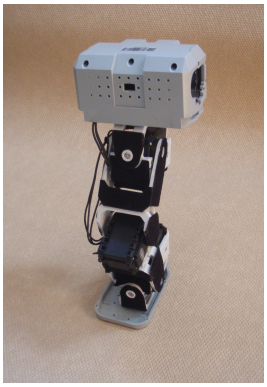


Figura 3.34: Imagem do robô real com uma perna.

3.6.3 Robô equivalente com meio tronco

À semelhança do robô equivalente com uma perna, e também com o intuito de estudar consumos energéticos mas durante o movimento de caminhar, foi desenvolvido um robô que é constituído por duas pernas (apenas com as articulações frontais), e um tronco que se divide em duas partes. À parte de cima é permitido

rodar de um lado para o outro de forma a equilibrar o robô sobre a perna que pretende manter o contacto no chão. Na realidade, o que esta massa faz é alternar a projecção da posição do centro de massa do robô para o interior da superfície de apoio de um pé para que outro pé possa ser levantado. Nas tabelas 3.5 e 3.6 encontra-se apresentada a estrutura deste robô com base nos ossos e articulações, respectivamente.

Tabela 3.5: Posição e tamanho da estrutura do robô com duas pernas.

Corpo	Descrição	Massa(kg)	Pos. CM (m)			Tamanho (m)		
ID			X	Y	Z	X	Y	Z
2	Pé esq	0.139	-0.035	0.0	-0.280	0.25	0.13	0.033
5	Pé dto	0.139	-0.035	-0.15	-0.280	0.25	0.13	0.033
6	Tronco	0.335	-0.040	-0.075	-0.05	0.082	0.22	0.08
7	Coxa esq	0.173	-0.030	0.0	-0.120	0.030	0.048	0.090
8	Coxa dta	0.173	-0.030	-0.15	-0.120	0.030	0.048	0.090
17	Joelho esq	0.077	-0.030	0.0	-0.210	0.049	0.047	0.101
18	Joelho dto	0.077	-0.030	-0.15	-0.210	0.049	0.047	0.101
22	Tronco 2	0.635	-0.040	-0.075	-0.08	0.082	0.11	0.20

Tabela 3.6: Posições das articulações do robô com duas pernas.

Art.	Descrição	Posição eixo (m)			Direcção eixo			ID1	ID2	Tipo	Min.	Max
ID		X	Y	Z	X	Y	Z				(graus)	
0	Joelho esq	-0.039	0.0	-0.170	0	1	0	7	17	Hinge	-98	32
1	Pé esq	-0.027	0.0	-0.26	0	1	0	17	2	Hinge	-10	125
2	Joelho dto	-0.039	-0.15	-0.170	0	1	0	8	18	Hinge	-98	32
3	Pé dto	-0.027	-0.15	-0.26	0	1	0	18	5	Hinge	-10	125
8	Anca esq	-0.03	0.0	-0.08	0	1	0	6	7	Hinge	-75	75
9	Anca dta	-0.03	-0.15	-0.08	0	1	0	6	8	Hinge	-75	75
22	Tronco 2	-0.04	-0.075	-0.02	1	0	0	22	6	Hinge	-90	90

Nas figuras 3.35 e 3.36 encontram-se ilustrados os robôs simulado e real, respectivamente. Na secção 6.3 é apresentada a sua utilidade e discutidos os resultados.

3.7 Visualização 3D

Pode-se afirmar que a visualização é um processo de transformação de dados num conjunto de gráficos Schroeder *et al.* (1996) e a transformação de dados é na prática um conjunto de algoritmos Johnson & Hansen (2004). De forma a visualizar os dados da simulação (posições de cada objecto) que o modelo de física detém, torna-se necessário desenhar imagens que ilustrem o mundo da simulação. Há alguns anos atrás, estas imagens eram apenas desenhos geométricos a 2D sem

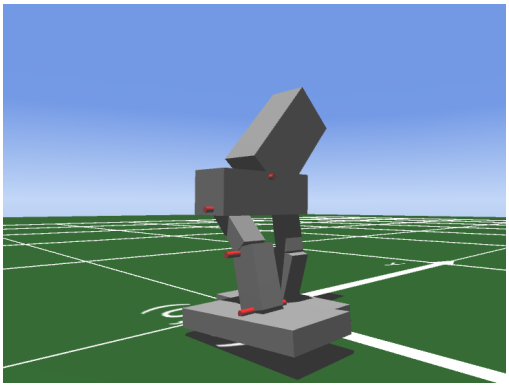


Figura 3.35: Imagem do robô simulado com duas pernas.

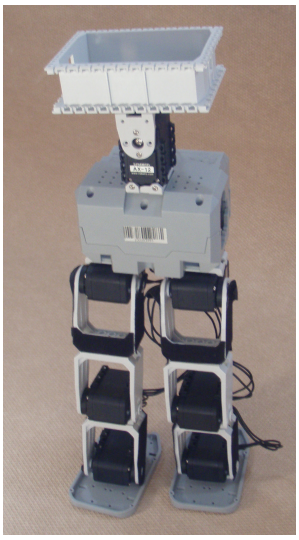


Figura 3.36: Imagem do robô real com duas pernas.

texturas. A dificuldade situava-se na capacidade de processamento insuficiente uma vez que um dos problemas comuns na renderização (processo de gerar imagens computadorizadas Schroeder *et al.* (1996)) de cenas 3D é a eliminação de superfícies escondidas, isto é, superfícies que não se encontram visíveis do ponto de vista da posição da câmara. A evolução de algoritmos e as especificações de hardware hoje em dia (a capacidade de processamento actual das placas gráficas) permitem atingir uma elevada capacidade de processamento gráfico, que possibilita o desenvolvimento de aplicações com imagens tridimensionais baseadas em texturas, animações, iluminação controlada e sombras que dão ênfase à visualização.

De forma a visualizar o mundo onde corre a simulação de uma forma intuitiva foi implementada, no simulador, uma visualização tridimensional, baseada numa biblioteca gráfica de funções que fornece as rotinas para renderização de ambientes 3D (OpenGL). A biblioteca utilizada, *GLScene* GLScene (2008), possui alguns componentes e objectos visuais que permitem a descrição e renderização de cenas 3D como se encontra ilustrado nas figuras 3.37 que apresentam algumas posturas ilustrativas do robô. Nas figuras 3.37 e) e f) é possível observar diferentes ângulos de visualização para a mesma postura. Para além disso, fornece ainda funções de zoom, movimentação da câmara e iluminação, entre outros. Esta biblioteca permite uma visualização em tempo real do mundo criado pelo modelo dinâmico, uma mais valia no simulador.

3.8 Atrito entre diferentes corpos

A força de atrito entre articulações está embebida no modelo do motor, tal como já foi apresentado na secção 3.4.4. No entanto, o atrito entre os pés do robô humanóide e o chão existe e não deverá ser desprezado. No caso de o robô se encontrar a caminhar (secção 3.10), a distância percorrida por cada passo dependerá do coeficiente de atrito (configurável pelo utilizador), tal como o robô real.

O deslocamento de um passo depende do coeficiente de atrito e da dureza da superfície. No robô humanóide real, este deslocamento poderá variar entre 0.04 m na alcatifa, cuja dureza impede os movimentos, até 0.065 m quando colocado

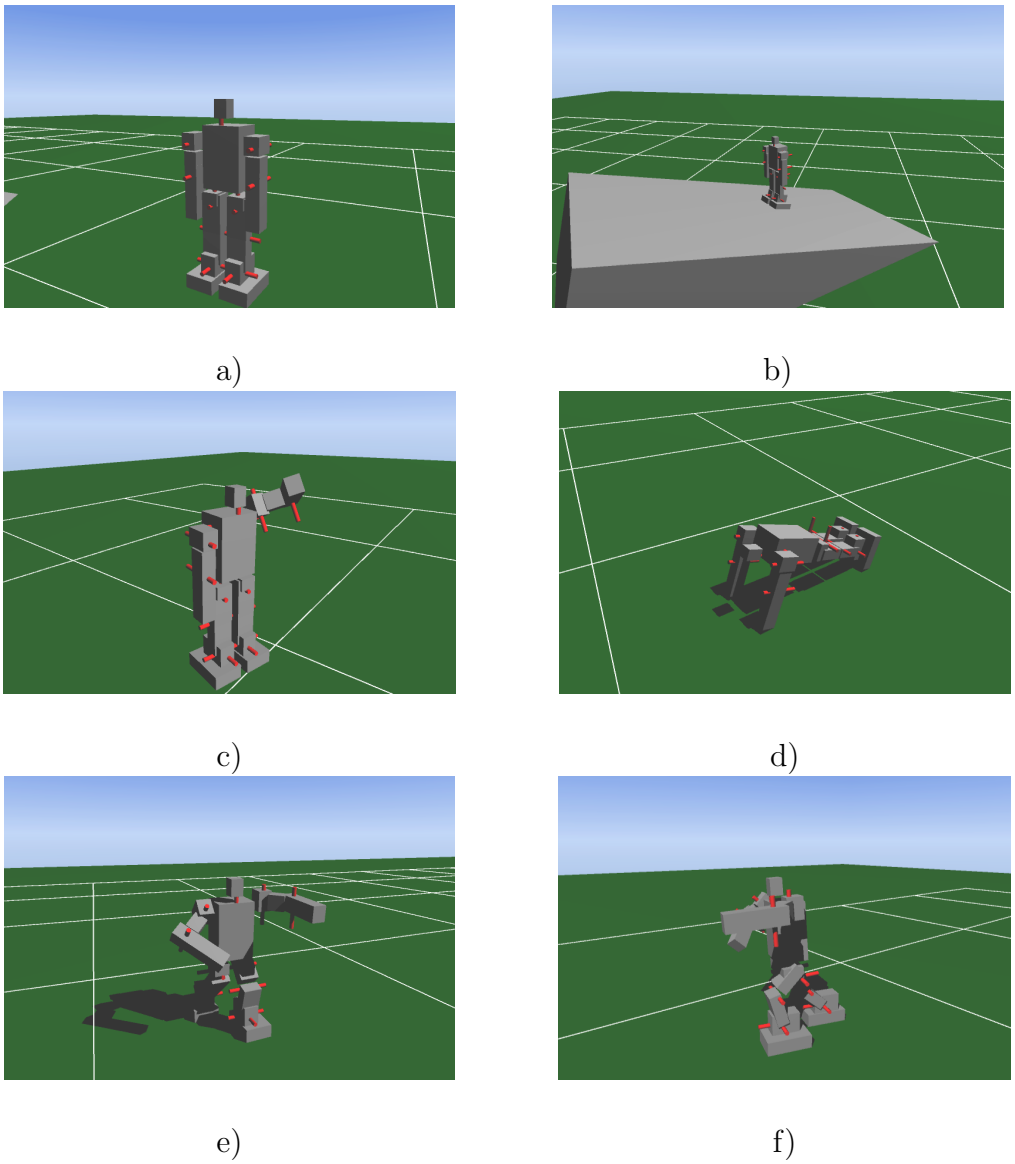


Figura 3.37: Robô humanóide do simulador em diversas posturas: a) em pé; b) em pé num plano inclinado; c) com braço na cabeça; d) postura de flexão; e) e f) postura de combate visto de diferentes perspectivas.

sobre papel (com uma superfície dura por baixo deste), tal como apresentado posteriormente na tabela 3.7. Por este motivo, deve ser utilizado um coeficiente de atrito entre os pés do robô humanóide e o chão na simulação tal como na realidade. Assim foi criado um coeficiente genérico de atrito entre os pés e o chão. O gráfico da figura 3.38 mostra a função entre o coeficiente de atrito e o deslocamento no simulador.

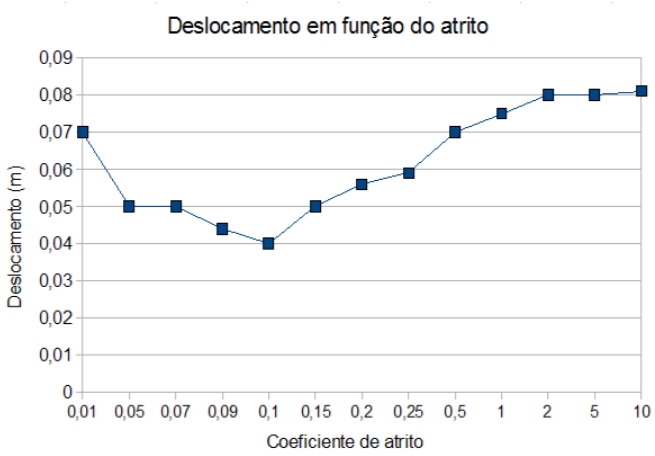


Figura 3.38: Deslocamento em função da força de atrito no simulador.

Para coeficientes de atrito baixos (inferiores a 0.1), o robô parece deslizar ou escorregar no chão não se assemelhando com a realidade (em condições normais). Para valores muito altos, o deslocamento tende para um valor constante, o que faz sentido uma vez que o robô levanta o pé e não o faz deslizar.

Por fim, foram medidas as deslocações do robô real para algumas superfícies e com base na deslocação pode-se estimar qual o atrito equivalente no simulador como apresentado na tabela 3.7 a qual apresenta as superfícies, o deslocamento e o atrito estimado, com base no simulador.

3.9 Editor de *Script*

Um editor de *Script* é uma ferramenta que permite usar e compilar código escrito pelo utilizador durante a execução do simulador. É composto por um conjunto de funções Costa (2008) que permitem aceder e controlar variáveis durante a

Tabela 3.7: Deslocamento de um passo para diversas superfícies.

Superfície	Deslocamento (m)	Coeficiente de atrito estimado
Alcatifa	0.04	0.1
Papel mole	0.05	0.15
Tijoleira lisa	0.055	0.18
Plástico	0.055	0.18
Madeira	0.059	0.19
Papel plastificado	0.06	0.2
Tijoleira rugosa	0.061	0.21
Papel duro	0.065	0.3

execução do simulador sem ter a necessidade de distribuir ficheiros adicionais para além do executável Software (2008). Esta opção de controlo tem a vantagem de não necessitar de um ambiente externo de desenvolvimento. Na prática é uma janela de programação onde o código pode ser alterado e compilado durante a execução do simulador. Desta forma é permitido desenvolver métodos de controlo e testá-los no ambiente de simulação à semelhança da realidade mas com a vantagem de ter um sistema de *debugging* incorporado. Na figura 3.39, encontra-se ilustrada a referida janela de programação *script*, onde na parte superior está o código e na inferior estão os valores das variáveis seleccionadas para *debugging*.

3.10 Simulação *versus* realidade

Uma forma para testar o simulador é aplicar o mesmo sinal de controlo a ambos os robôs, real e simulado, e analisar a diferença entre os comportamentos. O caso do movimento de caminhar, em que a geração dos estados das articulações se encontra descrita na literatura (ZMP Kajita *et al.* (2006) Zhang *et al.* (2008), por exemplo), permite obter a validação do simulador.

Assim, foram adquiridas as sequências que permitem ao robô real caminhar e posteriormente aplicadas ao simulador. Obviamente que no simulador a estrutura mecânica do robô deverá ser cuidadosamente construída e idêntica ao Bioloid real.

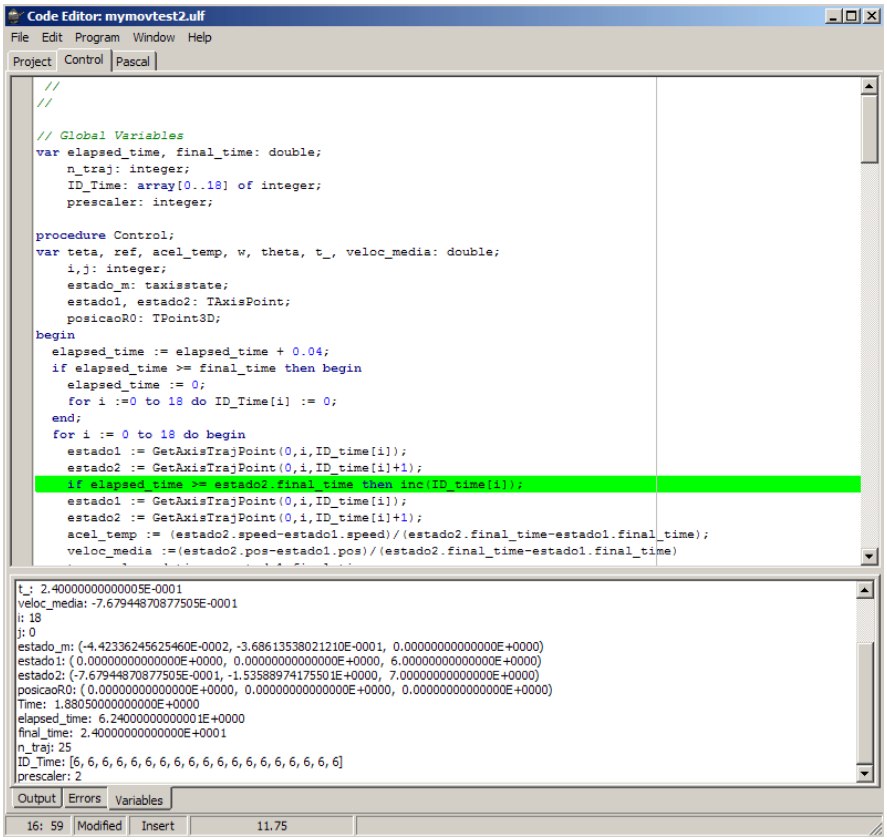
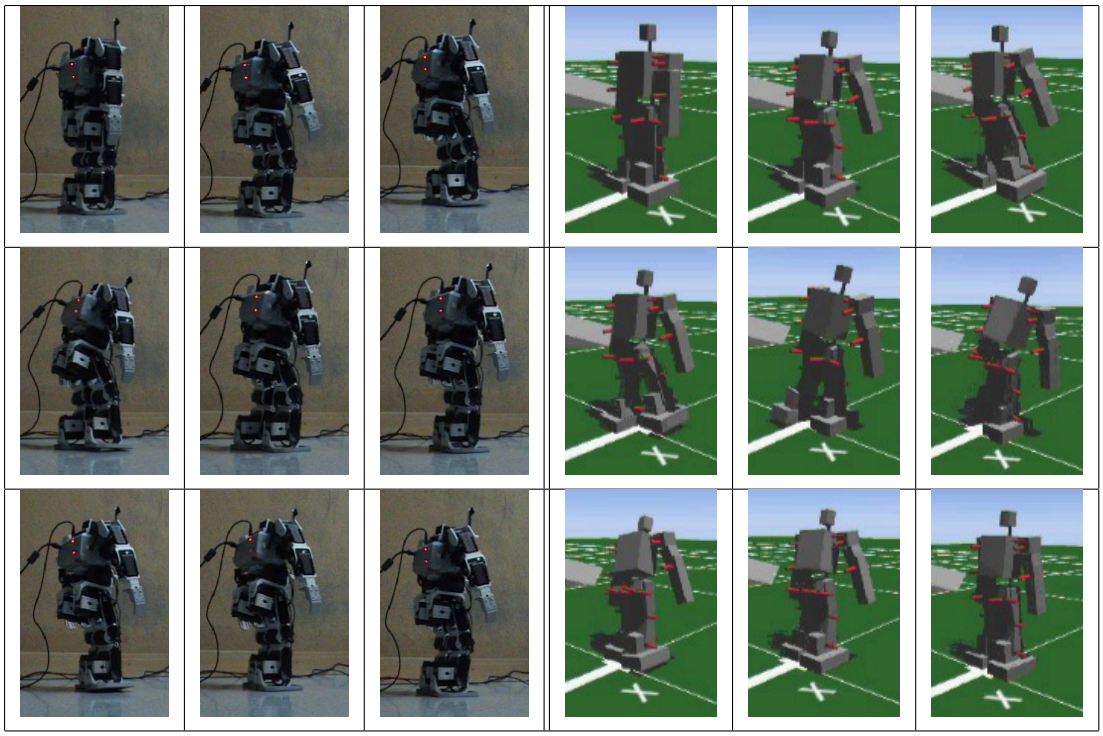


Figura 3.39: Janela do editor de *Script*.

Como resultado final, as figuras apresentadas na tabela 3.8 mostram à esquerda o robô real Bioloid e à direita o simulador a darem um passo com os mesmos estados nas articulações com um comportamento muito similar o que permite validar o simulador desenvolvido.

Tabela 3.8: Passo: Robô real e simulador.



É possível observar, no simulador, a posição do pé esquerdo, direito e corpo do robô em relação a um referencial colocado no chão durante um passo. Assim é possível confirmar que o robô simulado, tal como o real, levanta o pé enquanto o move para a frente o que implica que o outro pé se mantém estático como ilustra a figura 3.40. Enquanto o pé direito se move (do instante 2 até ao instante 3.6 seg.), o pé esquerdo permanece estático na posição 1 m. E por outro lado, quando o pé esquerdo se move (do instante 6 até 7.2 seg.) o pé direito mantém-se estático na posição 1.07 m.

O movimento final de recuo do tronco (de 1.13 para 1.05 m no instante 8.8 a 10.4 seg.) apenas se torna indispensável caso se necessite de colocar o robô

na posição completamente vertical, caso contrário, pode continuar com o passo seguinte. Note-se que os pés já se encontravam estáveis.

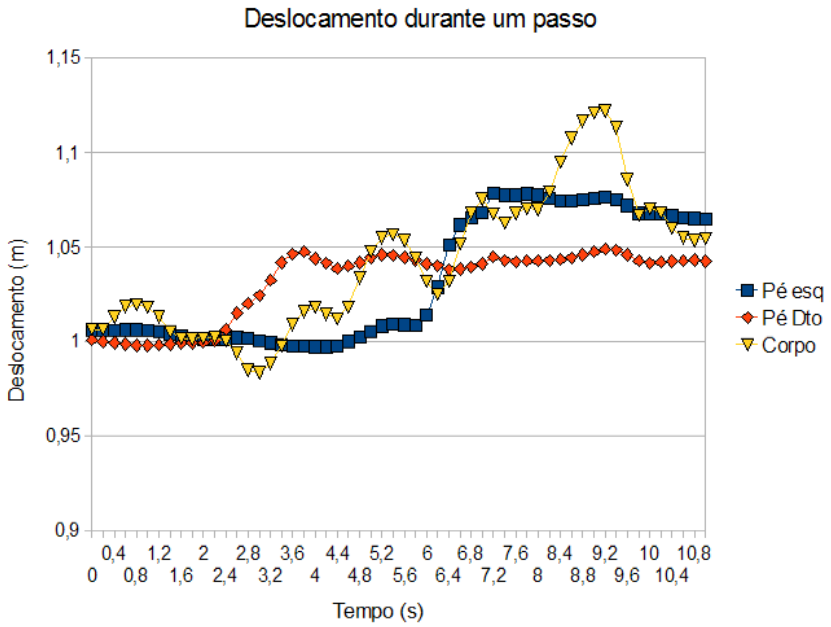


Figura 3.40: Deslocamento dos pés e do robô durante um passo.

Outra forma de validar o simulador durante um passo é analisar os ângulos das articulações entre os robôs real e simulado. O gráfico da figura 3.41 mostra os ângulos de referência de um joelho em função do tempo, para o robô real e para o robô simulado. Como se pode verificar, as curvas estão muito aproximadas, o que vem validar o simulador.

Desta forma, o simulador encontra-se validado, e tem um comportamento muito parecido ao robô real. Assim, os algoritmos de optimização podem ser testados na simulação à semelhança da realidade.

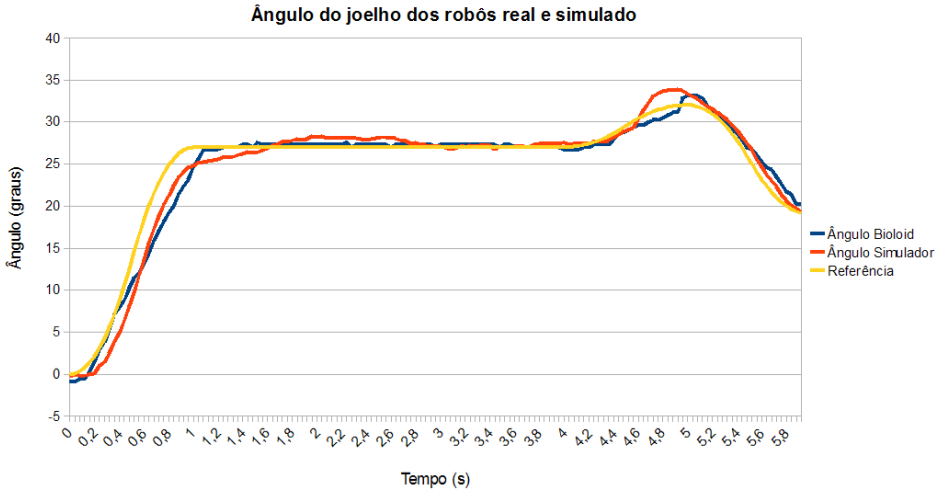


Figura 3.41: Ângulo do joelho dos robôs real e simulado.

3.11 Conclusão

Este capítulo apresentou o simulador desenvolvido referindo como é construído o robô sob ponto de vista da simulação. A linguagem XML é usada para permitir construir e alterar, de uma forma simples, a estrutura do robô. O modelo dos servos foi apresentado e validado onde a força de atrito entre articulações se encontra embebida. Por fim, o simulador é validado tendo como comparação o robô real Bioloid quando ambos os robôs se comportam de uma forma idêntica, se excitados da mesma forma (por exemplo a caminhar).

Capítulo 4

Optimização de trajectórias de baixo nível

4.1 Introdução

Este capítulo descreve o bloco ilustrado na figura 3.4 e apresenta várias formas como se podem optimizar trajectórias já planeadas. O controlador descrito neste capítulo encontra-se no nível intermédio o que permite receber as referências desejadas das articulações do nível superior e optimizar, utilizando diferentes metodologias, a trajectória de forma a minimizar velocidades, acelerações e energia consumida. A trajectória agora planeada é enviada ao nível mais baixo onde se encontra o controlador dos servos como ilustrado na figura 3.4. Deste modo, o nível da optimização de trajectórias recebe o próximo estado (ângulo e velocidade angular) assim como o instante desejado e desenha a trajectória a seguir até esse mesmo instante sob ponto de vista de trajectória óptima.

4.2 Planeamento de trajectórias

Se o nível alto do controlador do robô humanóide gera os estados para as articulações, o optimizador das trajectórias calcula a trajectória intermédia levando-as para o estado desejado e fornecendo-as ao nível inferior (servos) para que este as siga, de forma a minimizar a velocidade de ponta, a aceleração ou a ener-

gia consumida. A forma mais simples seria a interpolação entre estados (como ilustrado na figura 4.1) mas nesse caso o controlo da velocidade perder-se-ia e o deslocamento era efectuado, desde t_1 até t_2 com uma velocidade constante indicada pela equação 4.1. Isto acarretaria a mudanças bruscas na velocidade w_k nos instantes de comutação (entre t_2 e o t_1 do próximo período), para além de serem trajectórias impossíveis de o servo acompanhar.

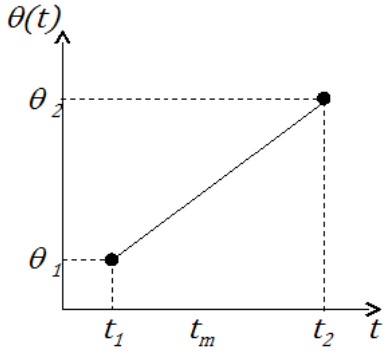


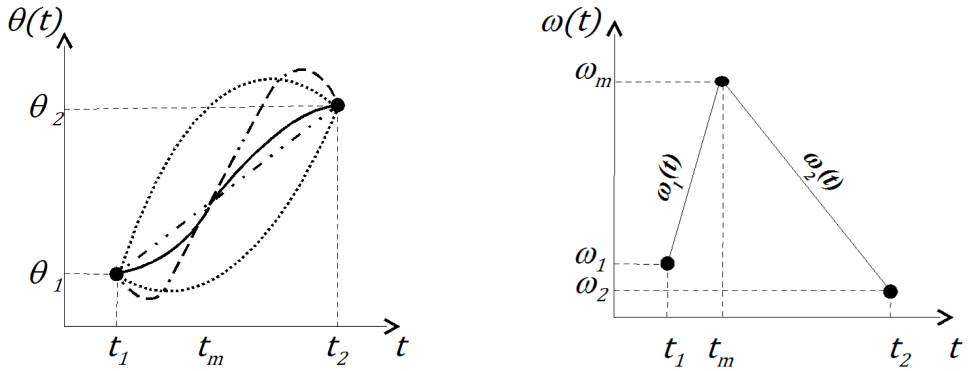
Figura 4.1: Interpolação de θ_1 para θ_2.

$$\omega_k = \frac{\theta_2 - \theta_1}{t_2 - t_1}. \tag{4.1}$$

Supondo que para $t = t_1$ (instante actual) é medido o ângulo θ_1 e uma velocidade angular ω_1, e para $t = t_2$ (próximo instante periódico) é desejado que a articulação esteja na posição θ_2 com uma velocidade angular ω_2, tal como ilustrado na figura 4.2 a) e b), que apresenta a simbologia usada neste capítulo assim como alguns exemplos de trajectórias possíveis assumindo uma aceleração constante nos intervalos $[t_1, t_m]$ e $[t_m, t_2]$, para uma articulação. Para o primeiro intervalo, a velocidade angular está descrita na equação 4.2 enquanto que para o segundo intervalo a velocidade angular está descrita na equação 4.3. A optimização da trajectória, calculada independentemente para cada articulação, resume-se ao cálculo do instante t_m onde se dá a mudança de trajectória, e obrigatoriamente o ω_m que garante o ângulo percorrido desejado $(\theta_2 - \theta_1)$, supondo apenas um instante temporal de mudança de velocidade.

$$\omega_1(t) = \frac{\omega_m - \omega_1}{t_m - t_1} t + \omega_1 - \frac{\omega_m - \omega_1}{t_m - t_1} t_1. \tag{4.2}$$

$$\omega_2(t) = \frac{\omega_2 - \omega_m}{t_2 - t_m}t + \omega_2 - \frac{\omega_2 - \omega_m}{t_2 - t_m}t_2. \tag{4.3}$$



a) b)

Figura 4.2: Estado das articulações: a) Ângulo e b) Velocidade angular.

Assim, torna-se necessário calcular a equação temporal do ângulo e velocidade angular que permitem obter as condições desejadas.

Para isso, depois de t_1 e antes de t_2 deverá haver um instante chave (t_m) com uma velocidade ω_m que devem ser calculados de forma a garantirem as condições desejadas supondo que a velocidade angular tem uma evolução linear nos intervalos de tempo $[t_1, t_m]$ e $[t_m, t_2]$ e a aceleração angular é portanto constante nesses mesmos intervalos de tempo. A forma de calcular o valor ω_m (para o instante t_m) dependerá do método escolhido: t_m a meio, minimização de velocidade, aceleração ou energia consumida como explicado nas secções seguintes. Para além disso, a referência angular e a velocidade angular devem ser calculadas de forma a proporcionarem movimentos suaves mas seguindo as condições desejadas. Quanto menos suave for o movimento a maior esforço estarão sujeitos os servos. Dentro do intervalo temporal t_1 e t_2 existe uma gama alargada de soluções de t_m para as mesmas condições. O ângulo percorrido $(\theta_2 - \theta_1)$ pode ser calculado de duas formas: como o integral das velocidades nos dois intervalos indicados na equação 4.4, ou como a diferença das áreas A_I, A_{II} e A_{III} ilustradas na figura 4.3 e na equação 4.5 (na realidade a área indicada é o integral da velocidade angular den-

tro de cada intervalo como apresentado na equação 4.4). A equação 4.6 permite calcular a função linear $\omega_m = f(t_m)$ representada pela linha $L2$ na figura 4.3.

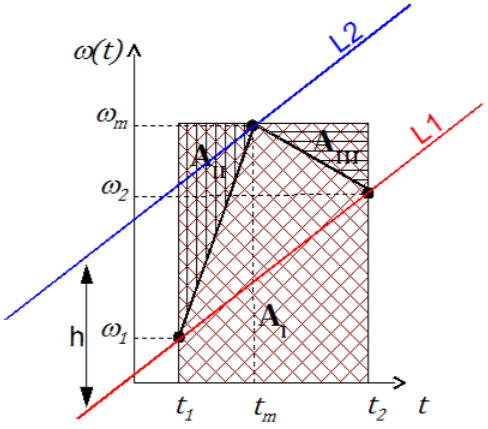


Figura 4.3: Cálculo do ponto (t_m, ω_m) sobre a linha $L2$.

$$\theta_2 - \theta_1 = \int_{t_1}^{t_m} \omega_1(t)dt + \int_{t_m}^{t_2} \omega_2(t)dt. \tag{4.4}$$

$$\begin{aligned} \theta_2 - \theta_1 = A_I - A_{II} - A_{III} = \\ = \omega_m(t_2 - t_1) - (\omega_m - \omega_1)(\frac{t_m - t_1}{2}) - (\omega_m - \omega_2)(\frac{t_2 - t_m}{2}). \end{aligned} \tag{4.5}$$

$$\omega\left(t_m\right) = \frac{2(\theta_2 - \theta_1 - \frac{t_m - t_1}{2}\omega_1 - \frac{t_2 - t_m}{2}\omega_2)}{t_2 - t_1}. \tag{4.6}$$

De facto, a equação 4.6 é uma função linear em t_m (descrita como a linha L2) e o seu declive (derivada) pode ser encontrado como $(\omega_2 - \omega_1)/(t_2 - t_1)$, ou seja, o mesmo declive que a linha $L1$ que une os pontos (t_1, ω_1) e (t_2, ω_2), como apresentado na equação 4.7.

$$\omega_m = \frac{\omega_2 - \omega_1}{t_2 - t_1} t_m + \frac{2\theta_2 - 2\theta_1 + t_1\omega_1 - t_2\omega_2}{t_2 - t_1}. \tag{4.7}$$

As linhas $L1$ e $L2$ estão afastadas de h que depende do ângulo percorrido e pode ser calculado através da equação 4.8.

$$h = \omega_m(t_1) - \omega_1. \tag{4.8}$$

Logo, h pode ser determinado através da equação 4.9.

$$h = 2\frac{\theta_2 - \theta_1}{t_2 - t_1} - \omega_2 - \omega_1. \tag{4.9}$$

O cálculo do ângulo $\theta_{ref}(t)$ pode ser encontrado recorrendo à formula do movimento uniformemente acelerado dentro de cada intervalo temporal $[t_1, t_m]$ e $[t_m, t_2]$ tal como está ilustrado na equação 4.10. Deste modo, t_m pode ser colocado dentro do intervalo $[t_1, t_2]$, de acordo com e equação 4.6 que garante o ângulo percorrido, $\theta_2 - \theta_1$, desejado. O cálculo do instante t_m dependerá do método de optimização escolhido e permite comutar a variável θ_{ref} para o intervalo respectivo da 4.10.

$$\begin{aligned} \theta^{Ref}(t) &= \theta_1 + \omega_1(t - t_1) + \frac{1}{2}\frac{\omega_m - \omega_1}{t_m - t_1}(t - t_1)^2 \ , t_1 <= t <= t_m \\ \theta^{Ref}(t) &= \theta^{Ref}(t_m) + \omega_m(t - t_m) + \frac{1}{2}\frac{\omega_2 - \omega_m}{t_2 - t_m}(t - t_m)t^2 \ , t_m < t <= t_2. \end{aligned} \tag{4.10}$$

As próximas secções discutem o cálculo de t_m tendo em conta a minimização da aceleração, da velocidade e da energia consumida. O método de t_m intermédio é também apresentado, uma vez que é o mais simples.

4.2.1 t_m intermédio

Como primeira abordagem, pode-se colocar o instante t_m a meio de t_1 e t_2 como indicado na equação 4.11.

$$t_m = \frac{t_1 + t_2}{2}. \tag{4.11}$$

Este é o cálculo mais simples mas que pode, por vezes, não apresentar um comportamento aceitável sob o ponto de vista de optimização, como é ilustrado nas próximas secções, mas que necessita de menor poder de cálculo. O cálculo de ω_m para o instante t_m pode ser obtido através da equação 4.6 em que resulta a equação 4.12.

$$\omega_m = \frac{(\omega_1 + \omega_2)(t_1 + t_2) + 4(\theta_2 - \theta_1)}{2(t_2 - t_1)}. \tag{4.12}$$

4.2.2 Minimização da aceleração

Uma vez que a aceleração está directamente associada ao binário exercido pelos servos, a minimização desta permite obter um decréscimo do esforço aumentando assim a longevidade dos servos. Para além disso, acelerações elevadas consomem mais energia e podem provocar desequilíbrios no robô aumentando assim a possibilidade de queda. Durante o intervalo $[t_1, t_m]$, a aceleração das articulações a_1 pode ser descrita pela equação 4.13, enquanto que para o intervalo $[t_m, t_2]$, a aceleração a_2 pode ser descrita pela equação 4.14 como ilustrado na figura 4.4.

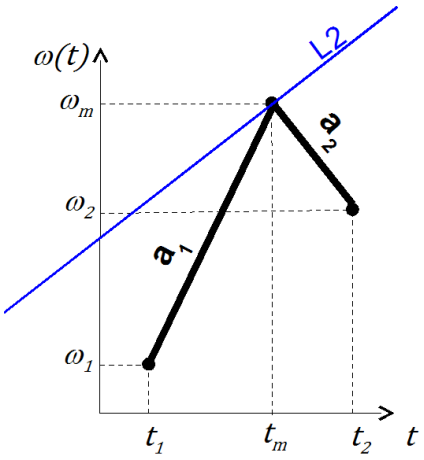


Figura 4.4: Minimização da aceleração.

$$a_1 = \frac{\omega_m - \omega_1}{t_m - t_1}. \tag{4.13}$$

$$a_2 = \frac{\omega_2 - \omega_m}{t_2 - t_m}. \tag{4.14}$$

Por uma questão de simplificação na demonstração que se t_m se move de t_1 para t_2, a_1 torna-se menor e por outro lado a_2 maior, considere-se a equação 4.15 que define a recta $L2$ indicada na figura 4.3 com um declive genérico m.

$$\omega(t) = mt + b. \tag{4.15}$$

Nesse caso, as acelerações a_1 e a_2 podem ser escritas como apresentado nas equações 4.16 e 4.17, respectivamente.

$$a_1(t_m) = \frac{mt_m + b - \omega_1}{t_m - t1}. \tag{4.16}$$

$$a_2(t_m) = \frac{-m_t m - b + \omega_2}{t_2 - t_m}. \tag{4.17}$$

Para a demonstração em causa, consideram-se as derivadas de a_1 e a_2 em ordem a t_m como calculadas pelas equações 4.18 e 4.19, respectivamente, para t_m dentro do intervalo $[t_1, t_2]$.

$$\frac{\partial a_1}{\partial t_m} = \frac{-mt_1 - b + \omega_1}{(t_m - t_1)^2}. \tag{4.18}$$

$$\frac{\partial a_2}{\partial t_m} = \frac{-mt_2 - b + \omega_2}{(t_2 - t_m)^2}. \tag{4.19}$$

Considerando que a recta $L2$ se encontra acima dos pontos (t_1, ω_1) e (t_2, ω_2) (para outros casos a demonstração será idêntica), vêm as derivadas de a_1 e a_2 negativas, o que demonstra que para a_1 se t_m diminui, então a_1 aumenta e vice-versa. Para a_2, e como esta aceleração neste caso é negativa, se t_m se aproxima de t_2, a_2 será mais negativo.

Como foi demonstrado (se t_m se move de t_1 para t_2, a_1 torna-se menor e por outro lado a_2 maior), o ponto t_m óptimo, sob o ponto de vista da minimização da aceleração, pode ser encontrado quando ambas as acelerações em módulo se igualam como apresentado na equação 4.20.

$$|a_1| = |a_2|\,. \tag{4.20}$$

A equação 4.20, que na realidade é traduzida pela equação 4.21, permite encontrar a função $t_m = f(\omega_m(t))$ para o ω_m desejado que está ilustrada na equação 4.22, tendo em atenção a minimização da aceleração durante a trajectória.

$$a_1 = -a_2. \tag{4.21}$$

$$t_m^{Amin} = \frac{\omega_m(-t_2 - t_1) + \omega_1 t_2 + \omega_2 t_1}{\omega_1 + \omega_2 - 2\omega_m}. \tag{4.22}$$

O ponto (t_m, ω_m) pode ser calculado substituindo a equação 4.22 na equação 4.6 de onde resulta 4.23.

$$\omega_m = \frac{1}{2}(\omega_2 + \omega_1 + h) \pm \sqrt{\omega_2^2 - 2\omega_2\omega_1 + \omega_1^2 + h^2}. \tag{4.23}$$

O $w(t_m)$ válido é aquele que origina um valor de t_m pertencente ao intervalo $[t_1, t_2]$. A referência de trajectória $\theta(t)$ e $w(t)$, que minimiza as acelerações a_1 e a_2 está assim encontrada.

4.2.3 Minimização da velocidade

Há casos em que algumas articulações do robô podem estar a descrever uma trajectória de um ponto A para um ponto B e num determinado instante a rota de referência seja alterada (exemplo disso é o robô estar a deslocar para a bola e esta sofrer uma colisão alterando a sua posição). Nessas circunstâncias o robô deverá, o mais rapidamente possível, alterar a sua trajectória. Note-se que quanto maior velocidade este se encontrar a descrever os movimentos, maior será a desaceleração necessária e mais tempo demorará até conseguir alterá-la. A forma de minimizar a velocidade angular é atingir a linha $L2$ o mais rápido possível mantendo sempre respeitado o limite máximo de aceleração permitido por cada servo. Assim, para todas as soluções do instante t_m, a que minimiza a velocidade angular está apresentada na figura 4.5 onde a_{max} é a aceleração máxima do servo. Esta pode ser facilmente determinada através do cálculo da segunda derivada da resposta do servo a um degrau na referência. Isto porque, o fabricante não define a aceleração máxima do servo mas sim o binário máximo que este consegue exercer e que poderá ser traduzido em aceleração dependentemente da carga a ele acoplada.

O instante t_m pode ser determinado através da intersecção das linhas $L2$ e $L3$ como apresentado na equação 4.24.

$$t_m = \frac{h}{a_{max} - \frac{\omega_2 - \omega_1}{t_2 - t_1}} + t_1. \tag{4.24}$$

O valor de ω_m para o instante t_m desejado pode ser determinado através da equação 4.6 conhecendo o t_m, ou directamente através da equação 4.25.

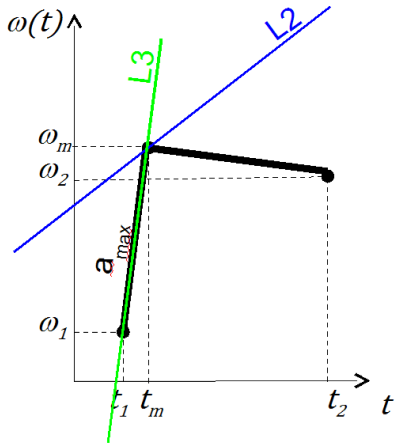


Figura 4.5: Minimização da velocidade angular ω_{max}.

$$\omega_m = -\frac{\omega_2 a_{max} t_1 + 2\theta_2 a_{max} - 2\theta_1 a_{max} - \omega_1\omega_2 + \omega_1^2 - \omega_2 t_2 a_{max}}{-a_{max} t_2 + a_{max} t_1 + \omega_2 - \omega_1}. \tag{4.25}$$

4.2.4 Minimização da energia consumida

Como tipicamente os robôs são alimentados por baterias abordo, o consumo da energia deve ser reduzido ao mínimo possível de forma a aumentar a autonomia. O desenho de trajectórias das articulações pode ser pensado tendo em conta o consumo de energia e a sua minimização. Considerando apenas a energia mecânica, e assumindo que o consumo instantâneo de potência por um servo pode ser determinado pelo produto do seu binário com a sua velocidade angular w e que o binário pode ser considerado proporcional à aceleração angular (desprezando-se os atritos), pode-se calcular a potência através da equação 4.26, onde J é o momento de inércia e a a aceleração angular. Assim torna-se possível encontrar o instante t_m que minimiza a energia consumida tal como apresentada nesta secção.

$$P_{Mec} = Ja\omega. \tag{4.26}$$

O momento de inércia J, depende da articulação e da postura do robô, mas por simplicidade pode ser considerada constante para os cálculos necessários da minimização da energia. O seu cálculo poderá ser efectuado através da equação

genérica do momento de inércia de um corpo, equação 4.27 onde m é a massa e r a distância do eixo de rotação ao corpo. A massa m poderá ser escrita como o produto da sua densidade ρ pelo volume V (altura, comprimento e largura, respectivamente A, C e L), como ilustrado na equação 4.28.

$$J = \sum m_i r_i^2. \tag{4.27}$$

$$m = \rho ACL. \tag{4.28}$$

Considerando a figura 4.6, a distância r poderá ser calculada a partir to teorema de *Pythagoras*, como descrito na equação 4.29.

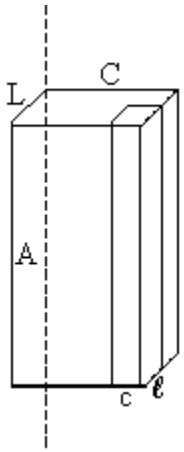


Figura 4.6: Cubo genérico para o cálculo do momento de inércia.

$$r = \sqrt{l^2 + c^2}. \tag{4.29}$$

Um braço do robô humanóide pode ser aproximado pela figura 4.7, e o momento de inércia pode então ser calculado a partir do integral apresentado na equação 4.30.

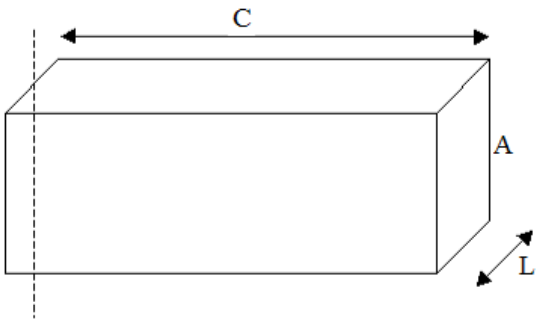


Figura 4.7: Paralelepípedo genérico para o cálculo do momento de inércia.

$$\begin{aligned} J &= \rho A \int_{-L/2}^{L/2} \int_{0}^{C} \left(l^2 + c^2\right)\ dc\ dl \\ &= \rho A \int_{-L/2}^{L/2} \frac{1}{3}C^3 + Cl^2\ dl \\ &= \rho A \frac{1}{3}\left[C^3 L + \frac{1}{3}C\frac{L^3}{4}\right] \\ = \frac{1}{3}\rho ACL\left(C^2 + \frac{L^2}{4}\right) &= \frac{1}{3}m\left(C^2 + \frac{L^2}{4}\right). \end{aligned} \tag{4.30}$$

Por exemplo, um braço estendido que mede aproximadamente 0.2 m de comprimento, 0.045 m de largura e pesa 0.174 kg tem um momento de inércia de aproximadamente 0.0023 $kg.m^2$.

Para o primeiro intervalo (onde $t \in [t_1, t_m]$) existe uma velocidade angular de $\omega_1(t)$ com uma aceleração constante a_1 e para o segundo intervalo (onde $t \in [t_m, t_2]$) existe uma velocidade angular $\omega_2(t)$ com uma aceleração a_2. O consumo instantâneo de potência está descrito na equação 4.31 onde os valores $\omega(t)$, a_1 e a_2 dependem do instante t_m e a energia total consumida durante o período t_1 a t_2 pode ser calculada através do integral da potência como indicado na equação 4.32.

$$P_1(t) = Ja_1\omega(t) \qquad P_2(t) = Ja_2\omega(t). \tag{4.31}$$

$$E_{TOT} = \int_{t_1}^{t_m} P_1(t) + \int_{t_m}^{t_2} P_2(t). \tag{4.32}$$

Substituindo a equação 4.32 pelas variáveis $\omega_1(t)$, $\omega_2(t)$, a_1 e a_2 obtém-se a equação 4.33.

$$\begin{aligned} E_{TOT} = J(&-10w_2^2t_2t_1 - 8w_1t_1\theta_1 - 4w_2t_m^2w_1 - 8w_2^2t_mt_2 \\ &-4w_1^2t_mt_1 + 2w_1^2t_1t_2 + 8w_2t_m\theta_2 - 8w_2t_m\theta 1 \\ &-8w_1t_m\theta_2 + 8w_1t_m\theta_1 + 8w_1t_1\theta_2 - 16\theta_2w_2t_2 \\ &+16\theta_1w_2t_2 + 2w_2^2t_m^2 + 2w_1^2t_m^2 +_w 1^2t_1^2 + 8\theta_2^2 \\ &-16\theta_2\theta_1 + 8\theta_1^2 + 9w_2^2t_2^2 - w_1^2t_2^2 + 4w_2^2t_1t_m \\ &+4w_2t_1^2w_1 + 8w_2t_1\theta_2 - 8w_2t_1\theta_1 + 8w_2t_mw_1t_2 \\ &-8w_1t_1w_2t_2 + 3w_2^2t_1^2)/(2(-t_2 + t_1)^2). \end{aligned} \tag{4.33}$$

O instante t_m que minimiza a energia consumida pode ser encontrado quando a primeira derivada da equação 4.33 é igual a zero (equação 4.34).

$$\frac{dE_{Tot}}{dt} = 0. \tag{4.34}$$

Isto é verdade porque a função $E_{TOT} = f(tm)$ é uma polinómio de segunda ordem do tipo apresentado pela equação 4.35 com a concavidade voltada para cima. Desta forma é possível determinar o mínimo quando a derivada é nula.

$$E_{TOT} = at_m^2 + bt_m + c. \tag{4.35}$$

A derivada da equação da energia está apresentada na equação 4.36.

$$\begin{aligned} \frac{dE_{Tot}}{dt} = J(&-8w_2t_mw_1 - 8w_2^2t_2 - 4t_1w_1^2 + 8\theta_2w_2 \\ &-8\theta_1w_2 - 8w_1\theta_2 + 8w_1\theta_1 + 4w_2^2t_m + 4w_1^2t_m \\ &+4w_2^2t_1 + 8w_2w_1t_2)/(2(-t_2 + t_1)^2). \end{aligned} \tag{4.36}$$

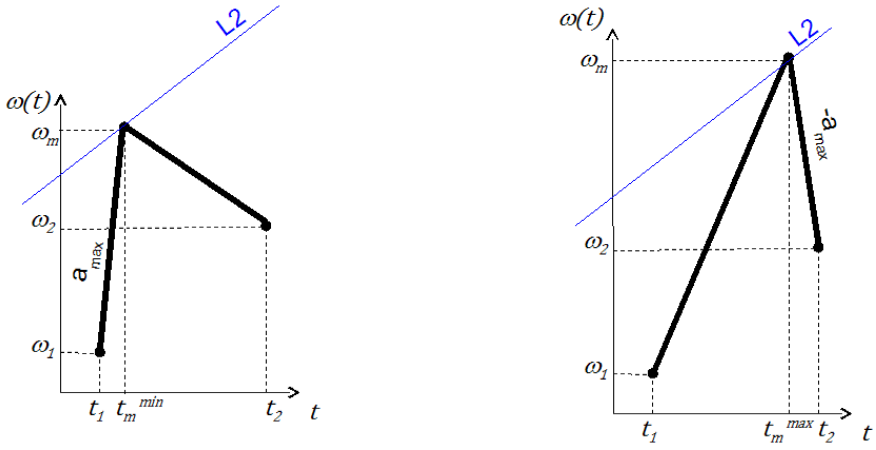
Assim cálculo do instante t_m está ilustrado na equação 4.37 e a velocidade angular ω_m pode ser calculada uma vez mais pela equação 4.6 com o conhecimento de t_m, ou directamente através da equação 4.38 que simplificada resulta na equação 4.39.

$$t_m = \frac{\omega_1 t_1 - 2\omega_2 t_2 - 2\theta_1 + 2\theta_2 + \omega_2 t_1}{-\omega_2 + \omega_1}. \tag{4.37}$$

$$\omega_m = \frac{2\theta_2 - 2\theta_1 - \omega_1 \left(\frac{\omega_1 t_1 - 2\omega_2 t_2 - 2\theta_1 + 2\theta_2 + \omega_2 t_1}{-\omega_2 + \omega_1} - t_1 \right) - \omega_2 \left(t_2 - \frac{\omega_1 t_1 - 2\omega_2 t_2 - 2\theta_1 + 2\theta_2 + \omega_2 t_1}{-\omega_2 + \omega_1} \right)}{t_2 - t_1}. \tag{4.38}$$

$$\omega_m = \omega_2. \tag{4.39}$$

De facto estes cálculos só são válidos quando o instante t_m se encontra no intervalo $[t_1, t_2]$, teoricamente. Na realidade, o intervalo é ainda mais apertado devido à aceleração máxima permitida pelo servo não ser infinita na prática como ilustrado na figura 4.8 a) e b).



a) b)

Figura 4.8: Valores mínimo a) e máximo b) permitidos para o instante t_m .

Desta forma, o instante t_m óptimo para o método optimização do consumo de energia deve ser limitado pelos instantes t_m^{min} ou t_m^{max}, fazendo coincidir com o método de mínima velocidade (secção 4.2.3). Tal como já mencionado este método apresenta algumas desvantagens e assim é preferível comutar para o método minimização da aceleração.

4.3 Conclusão

Este capítulo iniciou com uma abordagem ao desenho da trajectória pretendida pelo nível superior. Após algumas manipulações matemáticas, é apresentada a trajectória de cada servo que pode ser desenhada tendo em conta a optimização da velocidade, aceleração e energia consumida. A implementação e comparação dos métodos é efectuada na secção 5 em *Matlab*. Uma vez que o método aceleração mínima produz bons resultados, quer ao nível do consumo energético como ao nível de esforços, este será o método escolhido para ser implementado no simulador.

Capítulo 5

Teste e implementação dos diferentes métodos de optimização

5.1 Introdução

Neste capítulo é apresentada uma implementação em *Matlab* que permite validar os métodos previamente apresentados. Posteriormente será ilustrada a aplicação do método aceleração mínima no simulador uma vez que é o que proporciona tanto a economia da energia como a redução de esforços em comparação com os outros métodos. A implementação deste ou destes métodos no robô real, implica que o servo tenha a capacidade de seguir referências com rampas na velocidade, pelo que se torna necessário um novo controlador nos servos, tal como indicado em trabalho futuro na secção 7.2.

A interpolação entre estados (como ilustrado na figura 4.1) teria a vantagem de economizar energia durante a trajectória, uma vez que esta era efectuada a velocidade constante, e por conseguinte o binário necessário era apenas para vencer as forças de atrito. No entanto, nos instantes de comutação (entre t_2 e o t_1 do próximo período), as mudanças bruscas de velocidade seriam inevitáveis e o servo não conseguiria seguir a referência desejada. Para além de que este método não permitiria obter o ângulo final θ_2 com a velocidade desejada ω_2. O consumo

de energia paga o preço pelo desenho de trajectórias possíveis nas mudanças de intervalos.

5.2 Teste dos métodos de optimização em *Matlab*

Para o efeito, foram consideradas as seguintes condições iniciais e finais:

- $\theta_1 = 0$ *graus*
- $\theta_2 = 27$ *graus*
- $w_1 = 20$ *graus/s*
- $w_2 = 30$ *graus/s*
- $t_1 = 3$ *s*
- $t_2 = 4$ *s*.

5.2.1 Posicionamento de t_m intermédio

Utilizando este método, a aceleração poderá ser bastante diferente entre o primeiro e segundo intervalos. Para as condições apresentadas, a aceleração no primeiro intervalo é de 18 $graus/s^2$ e no segundo é de 2 $graus/s^2$. A velocidade angular, w_m, é de 29 $graus/s$ para o instante t_m de 3.5 *seg*. A energia consumida estimada é de 0.1774 mJ. A figura 5.1 apresenta a evolução das referências do ângulo e da velocidade angular para as condições definidas.

5.2.2 Minimização da aceleração

Neste caso, as acelerações durante os dois intervalos são iguais, em módulo, a 14.77 $graus/s^2$. A velocidade angular, t_m é de 32.39 $graus/s$ para o instante t_m de 3.838 *seg*. O consumo de energia estimado é de 0.1807 mJ. A figura 5.2 apresenta a evolução das referências do ângulo e da velocidade angular para as condições definidas.

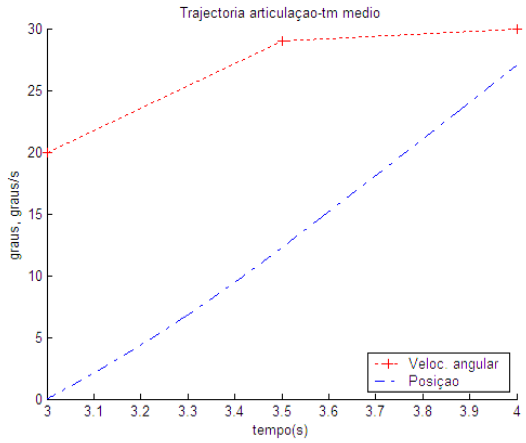


Figura 5.1: Resultado do método t_m médio.

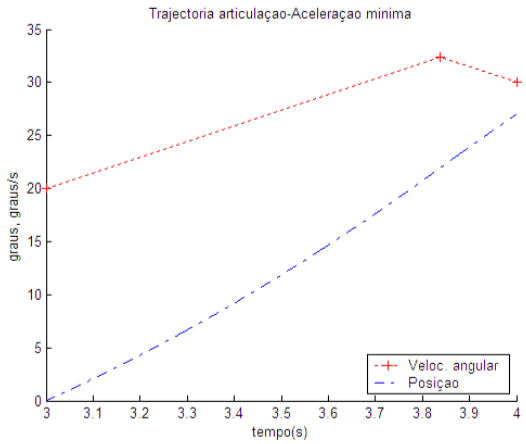


Figura 5.2: Resultado do método minimização da aceleração.

5.2.3 Minimização da velocidade

Para este método, o servo-motor arranca com a aceleração máxima de 200 $graus/s^2$ atingindo a velocidade angular w_m de 24.21 $graus/s$ no instante t_m de 3.02 s. A energia consumida estimada é de 0.2004 mJ. A figura 5.3 apresenta a evolução das referências do ângulo e da velocidade angular para as condições definidas.

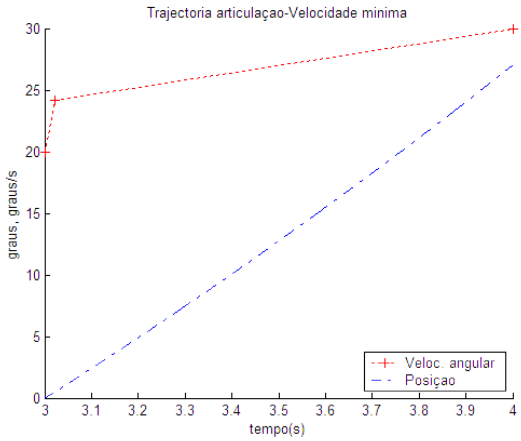


Figura 5.3: Resultado do método minimização da velocidade.

5.2.4 Minimização da energia consumida

A utilização deste método, garante uma maior conservação de energia para além de promover um movimento suave das articulações mas para apenas os casos em que o instante óptimo coincide com o intervalo temporal $[t_1, t_2]$, o que é raro acontecer na prática. Para o primeiro intervalo, a aceleração é de 16.67 $graus/s^2$ enquanto que o segundo intervalo é executado com uma velocidade constante, ou seja a aceleração é de 0 $graus/s^2$. No instante t_m de 3.6 s é atingida uma velocidade de 30 $graus/s$ e o consumo estimado de energia é de 0.1767 mJ. Uma vez mais, a figura 5.4 apresenta a evolução das referências do ângulo e da velocidade angular para as condições definidas.

É ainda possível visualizar na figura 5.5 a energia consumida em função do tempo (dentro do intervalo $[t_1, t_2]$). Como confirmação, o instante de t_m escolhido para este método (3.6 s) apresenta o mínimo da energia consumida.

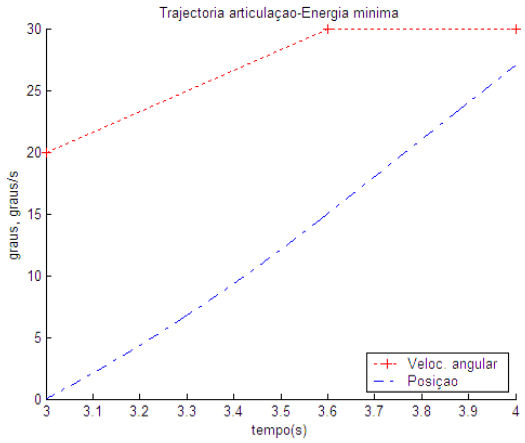


Figura 5.4: Resultado do método minimização da energia consumida.

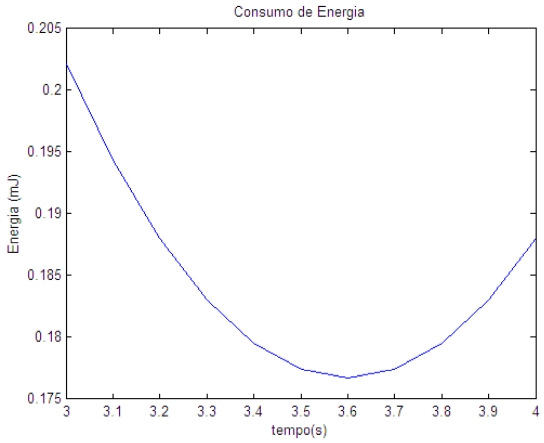


Figura 5.5: Consumo de energia em função de t_m.

O facto de ser pouco provável a aplicação deste método resultar num valor de t_m viável, e por consequência degenerar no método de velocidade mínima, acarreta o aparecimento de acelerações elevadas durante um curto espaço de tempo, tal como já referido. Claro que as condições iniciais deste teste foram cuidadosamente escolhidas para que o t_m óptimo sob ponto de vista da minimização da energia estivesse contido em $[t_1, t_2]$. Na realidade, se alterar o ângulo final ω_2 de 27 para 25 graus, o instante óptimo já seria 4 s o que leva a ter uma minimização com restrições activas. Neste caso, o valor óptimo estará numa fronteira, encostado a t_2. O gráfico da energia consumida em função de t_m para este novo t_2 encontra-se apresentado na figura 5.6.

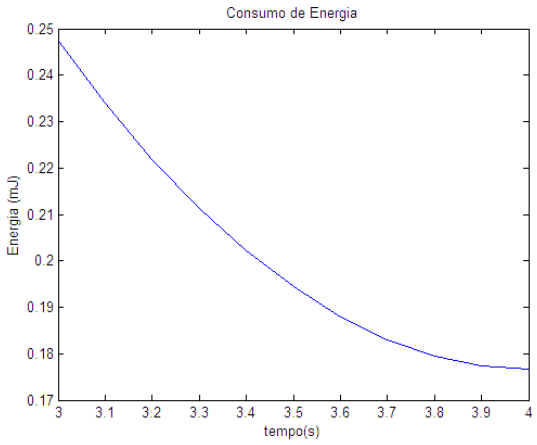


Figura 5.6: Consumo de energia em função de t_m para a nova condição.

5.3 Comparação dos métodos de cálculo do instante t_m

De modo a comparar os vários métodos de cálculo do instante t_m, estão apresentados na tabela 5.1 o valor de w_m, o instante t_m, a aceleração máxima a_{max} e a energia consumida E_{cons} para cada método apresentado (t_m médio - centrado em t_1 e t_2; A_{min} - minimização da aceleração; ω_{min} - minimização da velocidade angular; E_{min} - minimização da energia consumida).

Tabela 5.1: Comparação dos métodos de cálculo do instante t_m.

Método	t_m (s)	ω_m (graus/s)	$\lvert a_{max}\rvert$(graus/s^2)	E_{cons} (mJ)	E_{cons} (%)
t_m médio	3.5	29	18	0.1774	0.4
A_{min}	3.84	32.39	14.77	0.1807	2.26
ω_{min}	3.02	24.21 (30)	200.0	0.2004	13.41
E_{min}	3.6	30	16.67	0.1767	-

Aparentemente, o método t_m médio é o que necessita de menores cálculos e com resultados bastante razoáveis comparativamente aos outros métodos. Mas podem existir situações em que a aceleração num dos intervalos seja baixa obrigando assim a uma aceleração alta no outro o que leva a um esforço maior dos servos e consequentemente mais vibrações. Como exemplo, considerem-se as condições seguintes:

- $\theta_1 = 10\ graus$
- $\theta_2 = -20\ graus$
- $w_1 = -10\ graus/s$
- $w_2 = -100\ graus/s$
- $t_1 = 3\ s$
- $t_2 = 4\ s$.

Neste caso, a aceleração a_1 é de 10 $graus/s^2$ enquanto que a aceleração a_2 será de -190 $graus/s^2$ como se pode verificar no gráfico da figura 5.7.

No caso de optar pelo método de minimização da aceleração, esta será menor no segundo intervalo mas maior no primeiro. O módulo da aceleração neste caso é de 152.96 $graus/s^2$, como se pode verificar pela figura 5.8 que apresenta a trajectória para o mesmo exemplo, mas com o método de aceleração mínima implementado.

Assim, o método que promove a aceleração mínima parece interessante, sob o ponto de vista de esforço e vibrações provocadas no robô, e será adoptado como método preferido e utilizado no simulador.

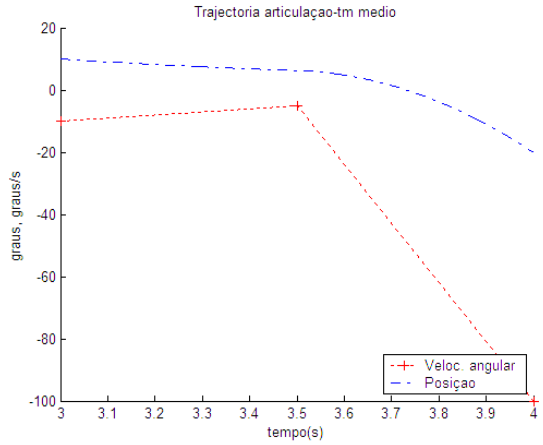


Figura 5.7: Resultado do método t_m médio com acelerações díspares.

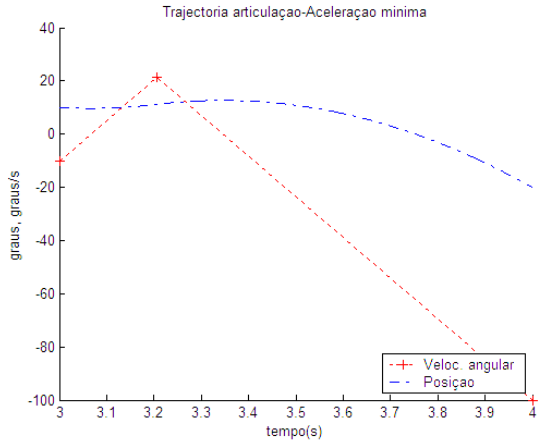


Figura 5.8: Resultado do método de minimização da aceleração para o segundo exemplo.

Na realidade, há casos em que é necessário comutar o método dependendo dos estados iniciais e finais, tal como proposto em trabalho futuro na secção 7.2.

De realçar que, apenas com um planeamento de trajectórias correcto se pode obter um decréscimo de consumo energético (no exemplo apresentado 13.4%), apenas com algumas manipulações matemáticas.

5.4 Teste do método optimização da aceleração mínima no simulador

As mesmas condições apresentadas na secção 5.2 (para t_1=3 s e t_2=4 s) foram introduzidas num braço do robô no simulador. Para além dos instantes 3 e 4 segundos, de forma a dar uma sequência coerente à articulação (imediatamente antes e depois) foram introduzidas as referências indicadas na tabela 5.2 que mostra a trajectória desejada desde o arranque até ao instante $t = 5$ segundos.

Tabela 5.2: Referências de trajectórias desejadas.

t(s)	1	3	4	5
θ ($graus$)	0	0	27	20
ω ($graus/s$)	0	20	30	0

Utilizando as equações 4.23 e 4.22, o simulador calcula os valores t_m e ω_m intermédios que resultam na tabela 5.3 de forma a que este possa definir as equações das trajectórias a seguir como se pode comprovar pela figura 5.9.

Tabela 5.3: Referências de trajectórias desejadas.

t(s)	1	1.586	3	3.839	4	4.654	5
θ ($graus$)	0	-4.131	0	21.946	27	25.821	20
ω ($graus/s$)	0	-14.142	20	32.3852	30	-33.627	0

A figura 5.9 apresenta o gráfico temporal com as curvas de referência do ângulo e velocidade angular assim como o estado da articulação, não só para as condições previamente testadas, mas também para os instantes referidos na tabela 5.2.

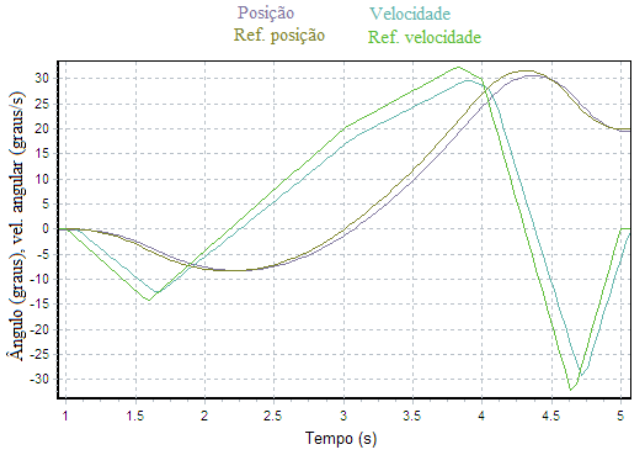


Figura 5.9: Resultado do método optimização da aceleração aplicado ao simulador.

Como se pode verificar, a referência é seguida pela articulação, embora com algum atraso. Note-se que nos instantes $t = 3$ e $t = 4$ segundos, os valores da referência do ângulo e da velocidade angular coincidem com o esperado (idênticos à figura 5.2), o que valida a implementação do método de minimização da aceleração no simulador.

O facto de existirem atrasos entre a velocidade real e a pretendida (referência) introduz erros, uma vez que no planeamento das trajectórias estes não se estavam a supor. O desenvolvimento de um controlador mais complexo, com erro nulo a referências de velocidade em rampa é proposto como trabalho futuro na secção 7.2.

5.5 Conclusão

Este capítulo apresentou a implementação e os resultados dos diferentes métodos propostos para optimizar a trajectória de cada servo. O método que minimiza a aceleração parece ser o mais interessante uma vez que não provoca grandes

esforços e acelerações no robô desequilibrando-o. A aplicação deste método no simulador fecha este capítulo.

Capítulo 6

Sistema passivo para aumento de eficiência do Robô

Este capítulo descreve a abordagem da aplicação de um sistema passivo que permite o aumento da eficiência do robô durante o movimento de caminhar.

6.1 Introdução

Hoje em dia, a maior parte dos robôs com pernas usa actuadores eléctricos ao contrário daqueles robôs de dinâmica passiva (PWD - *Passive Dynamic Walkers*) que são dispositivos sem apoio eléctrico que conseguem caminhar através de um plano que desce uma vez que estes usam a energia potencial gravítica para o alimentarem Versluys *et al.* (2008).

Tendo por base a energia potencial gravítica (E_p), expressa pela equação 6.1 em que m é a massa de um corpo, g a aceleração da gravidade e h a altura a que se encontra um determinado corpo, é possível observar que se o centro de massa do robô desce (supondo a postura de pé inicialmente), este está a perder energia potencial que não poderá ser reutilizada uma vez que os servos não têm a capacidade de regenerar esta energia para as baterias, ou fonte de alimentação.

$$E_p = mgh. \tag{6.1}$$

Para além disso, o facto dos enrolamentos do servo terem uma resistência

R_a (apresentada na secção 3.4.1) acarreta perdas por efeito de *Joule*, de modo algum desprezáveis, que são proporcionais ao esforço do servo, como se pode ver na equação 6.2, onde P_d é a potência dissipada e T_h é o binário necessário para manter a posição actual do robô.

$$P_d = R_a \frac{(T_h - T_{ae})^2}{K_s^2}. \tag{6.2}$$

Assim, enquanto o robô se mantém numa postura baixa e para que o robô volte a subir, a energia necessária para esse efeito provém na totalidade das baterias. Na secção 6.2, encontra-se apresentada e discutida uma forma que permite diminuir o consumo energético durante alguns movimentos.

6.2 Método proposto para optimização dos parâmetros

Posto o problema de outra forma, para que o centro de massa do robô desça, é necessário que as articulações tornozelo, joelho e anca aumentem o ângulo ficando em esforço, como ilustrado na figura 6.1.

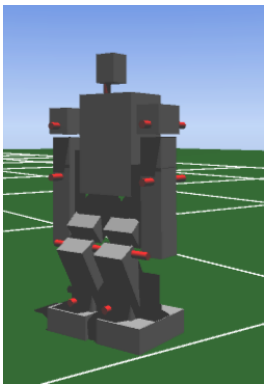


Figura 6.1: Postura do robô humanóide com as pernas flectidas.

Para voltar à posição inicial (de pé), os servos têm de efectuar um binário bastante elevado, dependendo da velocidade de subida e da postura. No entanto, durante a descida parte da energia potencial é perdida, para além da energia consumida para manter a posição baixa do robô e que não é de modo nenhum desprezável (potência de 0.9 W). Uma forma de armazenar temporariamente

esta energia é acoplar à articulação um elemento elástico (mola), cuidadosamente calculado, que enquanto o robô se encontra numa postura baixa (com os ângulos das articulações diferentes de zero) esta auxilia o servo no binário e quando este se encontra no movimento de levantar, a mola dissipa a energia acumulada para a energia potencial gravitacional do centro de massa do robô. A força exercida pela mola poderá ser calculada como uma função linear da sua extensão, através da lei de *Hooke*. Mas, uma vez que se pretende armazenar energia com a alteração de um ângulo de uma articulação, a opção de utilizar molas em espiral, como ilustrado na figura 6.2, é a mais conveniente.

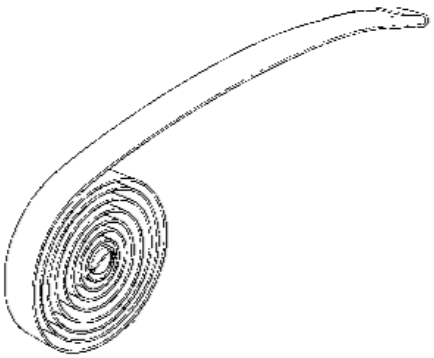


Figura 6.2: Mola espiral.

Esta mola exerce um binário T_m proporcional ao ângulo de torção α_m, tal como indicado pela equação 6.3.

$$F = k_m \Delta \alpha_m. \tag{6.3}$$

Como é claro, para uma mola mal dimensionada (demasiadamente forte) o servo tem ainda que efectuar um binário para comprimir a mola. Assim, esta situação não é de modo algum aconselhada. Para efectuar a análise do consumo energético (integral da potência consumida ao longo de um período), com base na simulação e com o mínimo de ruído possível (utilização do menor número possível de articulações, como explicado na secção 3.6.1), foi desenvolvido um robô que apenas é constituído por uma perna e um tronco. As articulações desnecessárias, tais como rotação da perna e abertura lateral, foram desactivadas restando assim três articulações básicas: o joelho, tornozelo e anca frontais, respectivamente as

articulações com ID 0, 1 e 8 (de acordo com a tabela de construção apresentada na secção 3.3). Este robô protótipo está apresentado na figura 6.3 e encontra-se na posição equivalente ao robô humanóide de pé.

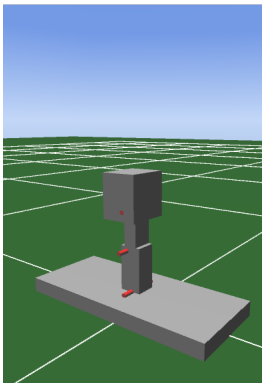


Figura 6.3: Screenshot do robô equivalente com uma perna na posição equivalente de pé.

Para verificar a vantagem do uso deste componente, optou-se inicialmente por simular o movimento de caminhar e para isso torna-se necessário baixar o centro de massa (à semelhança com o robô humanóide). Para esse efeito, os ângulos das articulações referidas têm que aumentar. Na posição mais baixa de teste, com as articulações a seguirem as referências indicadas na tabela 6.1, o robô encontra-se com a postura ilustrada na figura 6.4.

Tabela 6.1: Ângulos de referência para baixar centro de massa do robô com uma perna.

ID	Descrição	Ângulo ref. (*graus*)
0	Joelho	-50
1	Tornozelo frt	40
8	Anca frt.	10

Desta forma, o movimento das articulações utilizado pela perna do robô que servirá de teste encontra-se ilustrado pelo gráfico temporal da figura 6.5 que mostra, para cada articulação, a referência e o ângulo do eixo que a segue.

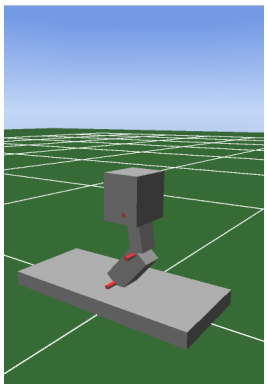


Figura 6.4: Screenshot do robô equivalente com uma perna na posição equivalente baixa.

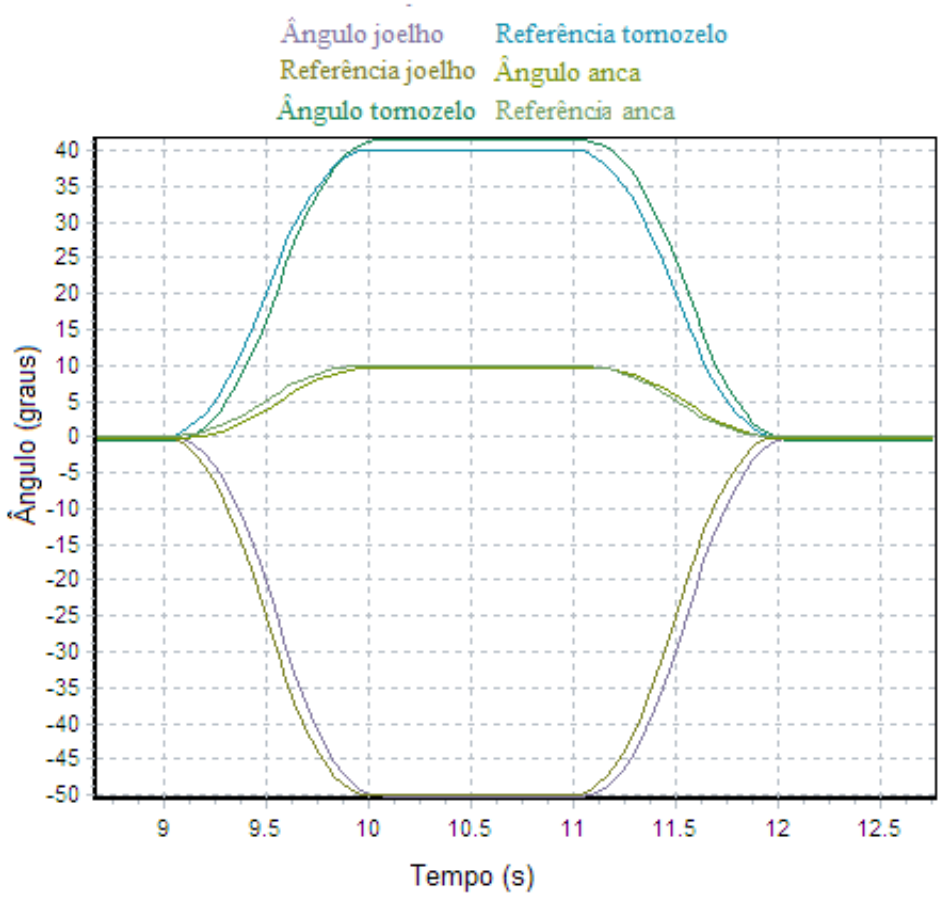


Figura 6.5: Gráfico temporal com referências e ângulos dos eixos das três articulações da perna do robô.

O elemento elástico a adicionar tem como propriedades, segundo a lei de *Hooke* (equação 6.3), uma constante denominada característica constante da mola (k_m) e a posição de zero, $offset$ (a partir da qual o valor $\Delta\alpha_m$ começa a ser diferente de zero). O gráfico da figura 6.6 apresenta o consumo energético durante o mesmo ciclo mas com a variação das duas variáveis (k_m e $offset$). Para k=0, ou seja, onde a acção da mola é nula, o consumo energético é de 0.35 J.

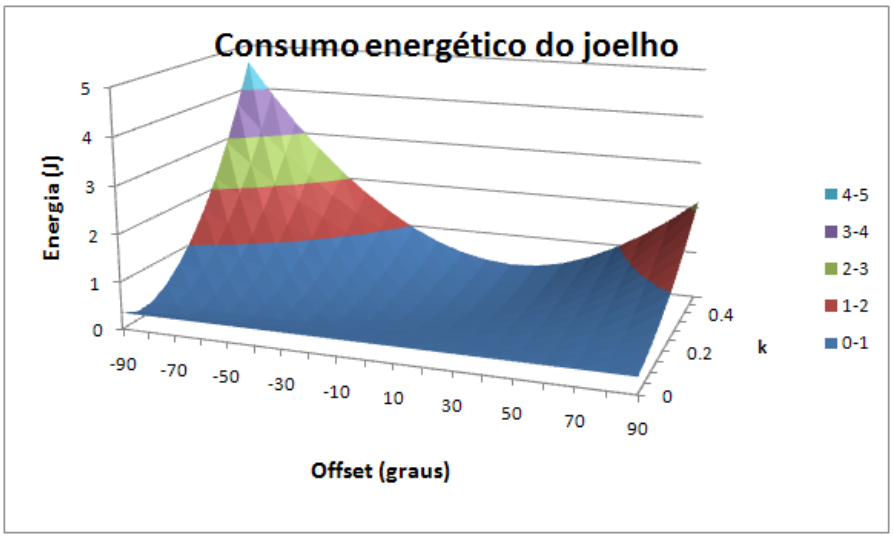


Figura 6.6: Gráfico de superfície do consumo energético do joelho em função dos parâmetros da mola.

Pela figura, é possível observar que existem pontos onde o consumo é inferior ao necessário em k=0. A figura 6.7 apresenta, com maior detalhe, o ponto onde o consumo é mínimo.

Para o ponto K=0.15 e $offset$=-20 $graus$ o consumo é de 0.143 J, o que significa que a acção da mola é benéfica e permite baixar o consumo energético para cerca de 40% neste caso. Claro que parâmetros da mola incorrectos implicam um consumo superior. O cálculo destes parâmetros, k_m e $offset$, depende de articulação para articulação e da trajectória planeada. Uma forma de os determinar, é utilizar o simulador com o planeamento de trajectória desejado, para cada articulação, e varrer uma determinada gama de k_m e $offset$ possíveis. O cálculo de cada ponto é realizado em tempo real com o movimento desejado. O número de simulações dependerá da gama e da sua resolução. Esta abordagem permite

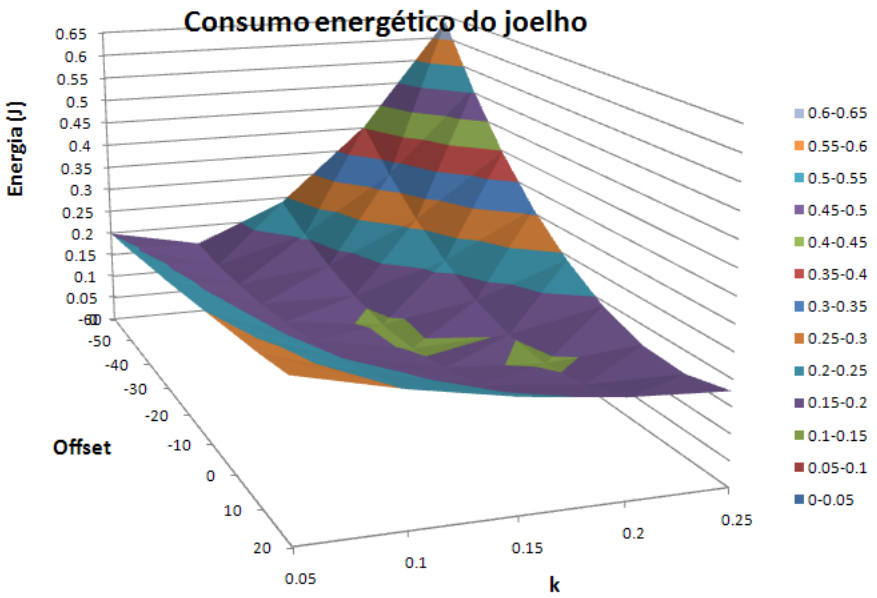


Figura 6.7: Gráfico de superfície do consumo energético do joelho em função dos parâmetros da mola (próximo do mínimo).

visualizar a forma da função custo. A alternativa de usar um algoritmo de minimização poderia ser mais eficiente mas não permitiria conseguir este género de informação. Um dos aspectos importantes é a convexidade da função custo e a verificação da quase ausência de mínimos relativos.

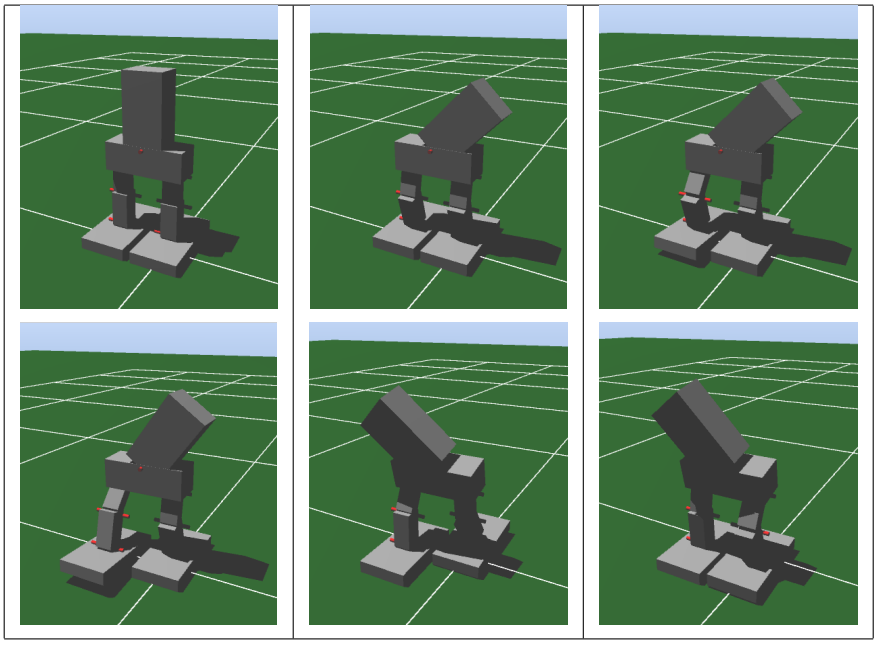
Para cada parâmetro, deverá ser calculada a energia consumida por articulação e no final escolhido o ponto de consumo mínimo. Nas próximas secções são aplicados diversos parâmetros a cada uma das articulações joelhos, tornozelos e anca. As figuras da tabela 6.2 ilustram a sequência deste robô durante um passo.

Para os testes foi utilizado o robô simplificado, apresentado na secção 3.6.3, uma vez que possui menor erro numérico como já referido na secção 3.6.1.

6.2.1 Joelho esquerdo

O consumo energético da articulação joelho esquerdo em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.8.

Tabela 6.2: Sequência de figuras do robô equivalente durante um passo.



O ponto de consumo mínimo está posicionado em k_m de 0.05 Nm^{-1} e um $offset$ de -80 graus. Com este elemento elástico, o consumo passou de 0.942 para 0.847 J, o que representa 10.1 % de decréscimo no consumo energético desta articulação.

6.2.2 Joelho direito

O consumo energético da articulação joelho direito em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.9.

O ponto de consumo mínimo está posicionado em k_m de 0.10 Nm^{-1} e um $offset$ de 40 graus. Com este elemento elástico, o consumo passou de 1.094 para 0.878 J, o que representa 19.7 % de decréscimo no consumo energético desta articulação.

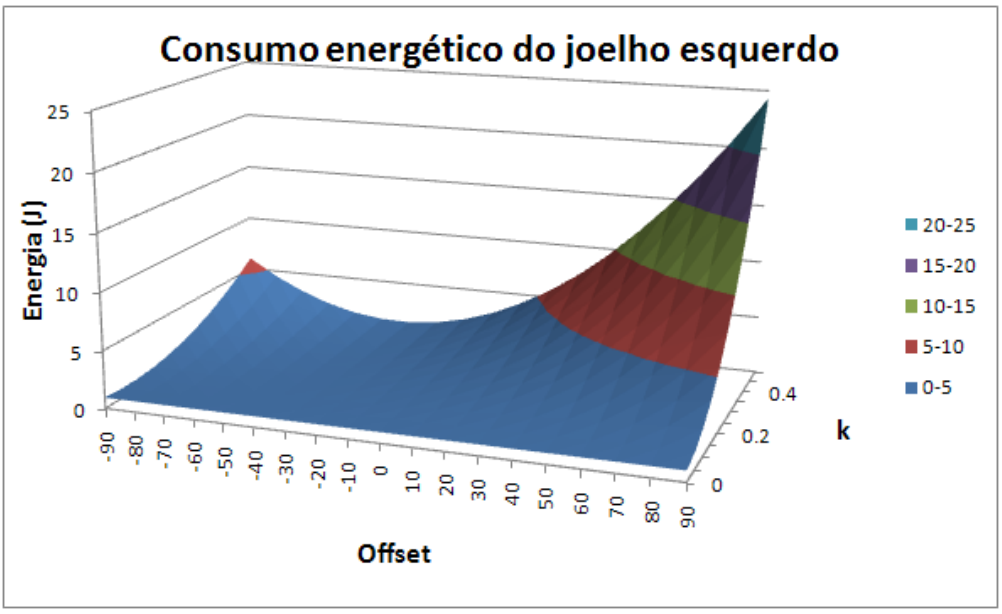


Figura 6.8: Gráfico de superfície do consumo energético do joelho esquerdo em função dos parâmetros da mola.

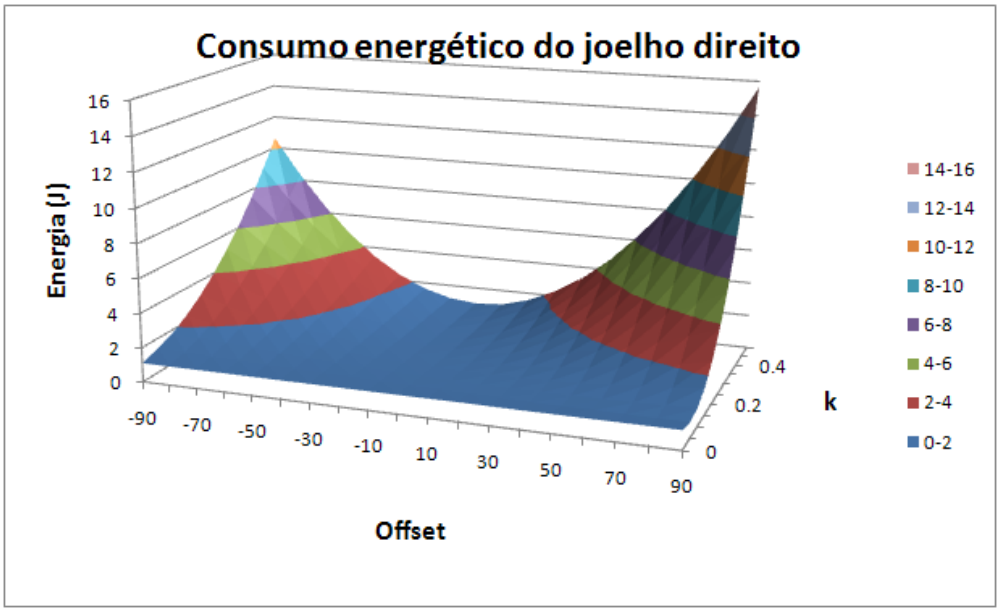


Figura 6.9: Gráfico de superfície do consumo energético do joelho direito em função dos parâmetros da mola.

6.2.3 Tornozelo esquerdo

O consumo energético da articulação tornozelo esquerdo em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.10.

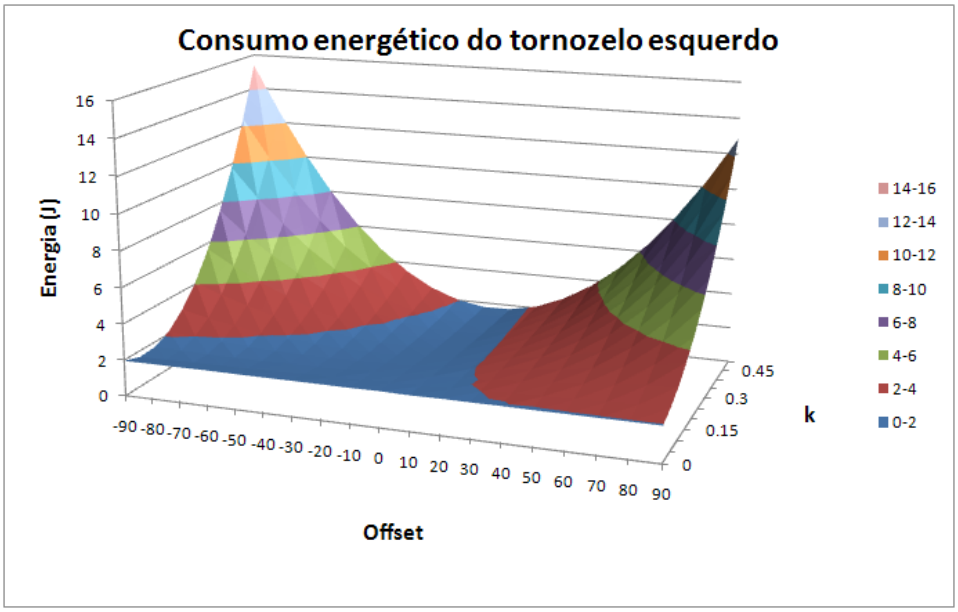


Figura 6.10: Gráfico de superfície do consumo energético do tornozelo esquerdo em função dos parâmetros da mola.

O ponto de consumo mínimo está posicionado em k_m de 0.25 Nm^{-1} e um $offset$ de -10 graus. Com este elemento elástico, o consumo passou de 1.968 para 1.629 J, o que representa 17.2 % de decréscimo no consumo energético desta articulação.

6.2.4 Tornozelo direito

O consumo energético da articulação tornozelo direito em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.11.

O ponto de consumo mínimo está posicionado em k_m de 0.05 Nm^{-1} e um $offset$ de 150 graus. Com este elemento elástico, o consumo passou de 1.903 para 1.45 J, o que representa 23.8 % de decréscimo no consumo energético desta articulação.

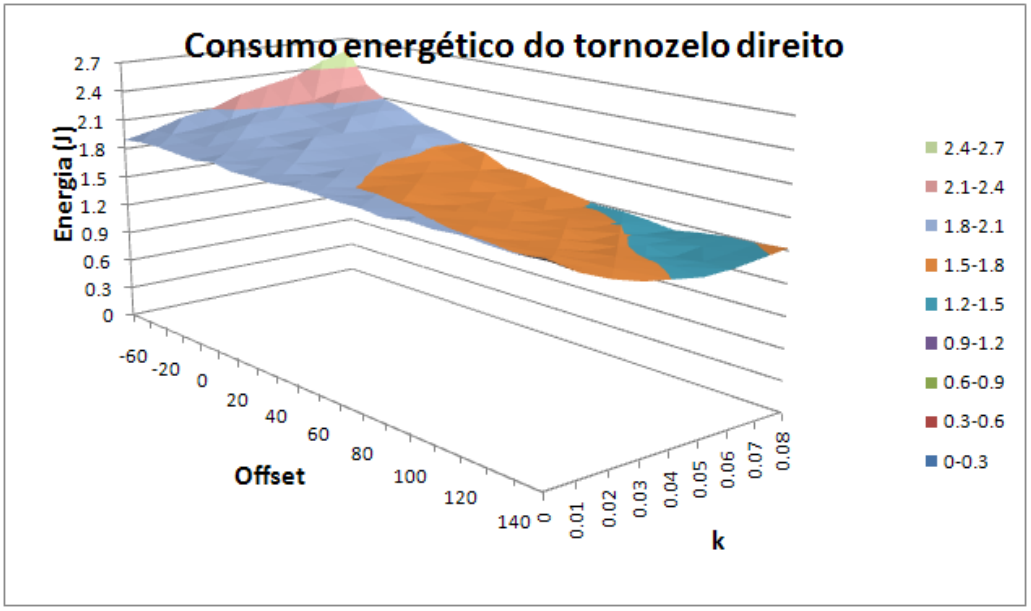


Figura 6.11: Gráfico de superfície do consumo energético do tornozelo direito em função dos parâmetros da mola.

6.2.5 Anca esquerda

O consumo energético da articulação anca esquerda em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.12.

O ponto de consumo mínimo está posicionado em k_m de 0.05 Nm^{-1} e um $offset$ de -60 graus. Com este elemento elástico, o consumo passou de 0.785 para 0.687 J, o que representa 12.5 % de decréscimo no consumo energético desta articulação.

6.2.6 Anca direita

O consumo energético da articulação anca direita em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.13.

O ponto de consumo mínimo está posicionado em k_m de 0.05 Nm^{-1} e um $offset$ de 80 graus. Com este elemento elástico, o consumo passou de 0.853 para 0.793 J, o que representa 7 % de decréscimo no consumo energético desta articulação.

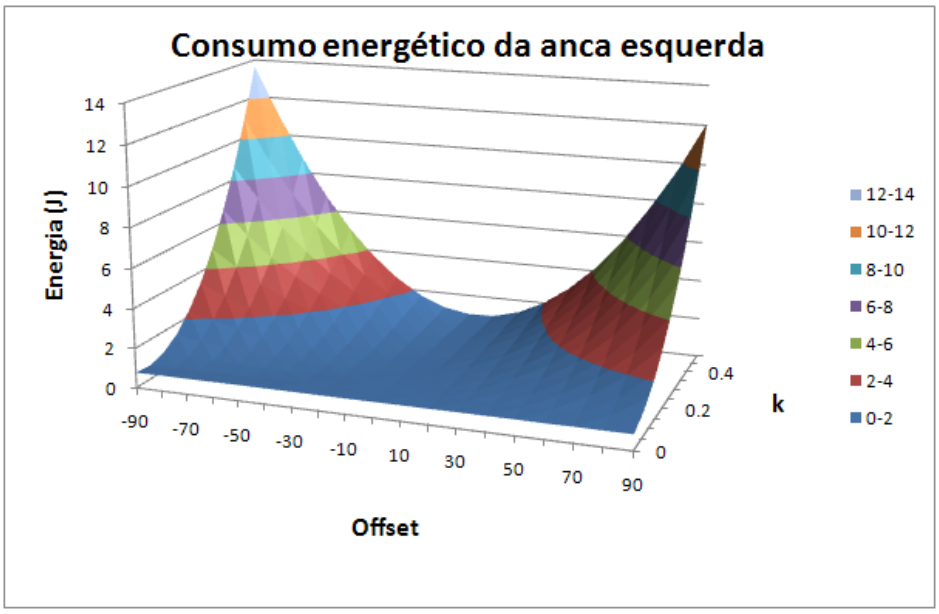


Figura 6.12: Gráfico de superfície do consumo energético da anca esquerda em função dos parâmetros da mola.

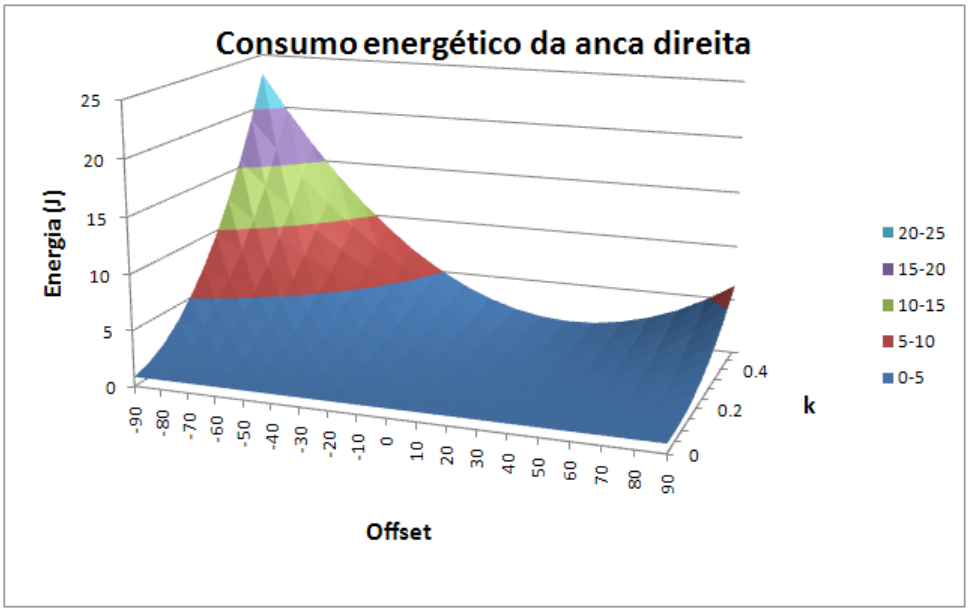


Figura 6.13: Gráfico de superfície do consumo energético da anca direita em função dos parâmetros da mola.

6.2.7 Tronco de equilíbrio

O consumo energético da articulação tronco de equilíbrio em função dos parâmetros k_m e $offset$ do elemento elástico a acoplar está indicado na figura 6.14.

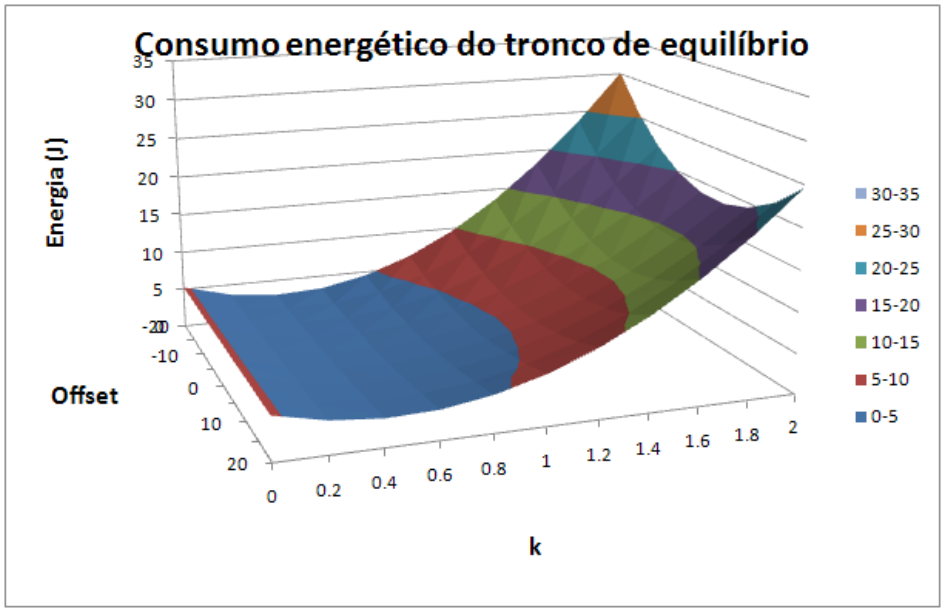


Figura 6.14: Gráfico de superfície do consumo energético do tronco de equilíbrio em função dos parâmetros da mola.

O ponto de consumo mínimo está posicionado em k_m de 0.6 Nm^{-1} e um $offset$ de 0 graus. Com este elemento elástico, o consumo passou de 5.255 para 2.609 J, o que representa 50.4 % de decréscimo no consumo energético desta articulação.

6.3 Análise dos resultados finais

A tabela 6.3 apresenta de uma forma resumida o consumo para cada articulação nos modos com e sem elementos elásticos e o ganho obtido.

Como é possível constatar, a inclusão de elementos elásticos óptimos nas articulações levou a um aumento de 30.86 % na eficiência no robô durante o mesmo movimento. O maior ganho foi obtido na articulação responsável pela anca de equilíbrio. Se esta não for considerada no cálculo, o aumento da eficiência do

Tabela 6.3: Resultados da aplicação dos elementos elásticos óptimos nas articulações do simulador.

Articulação	En. s/ molas (J)	En. c/ mola óptima (J)	Ganho (%)
Joelho esq	0.942	0.85	9.77
Joelho dto	1.094	0.81	25.96
Tornoz esq	1.968	1.63	17.17
Tornoz dto	1.903	1.45	23.8
Anca esq	0.785	0.69	12.1
Anca dta	0.853	0.82	3.87
Anca equil.	5.255	2.6	50.52
Total	12.800	8.85	30.86

robô não será tão alto mas atinge perto de 20 %, que mesmo assim confirma a vantagem da utilização deste método.

6.4 Conclusão

A figura 6.15 ilustra, de uma forma conclusiva, um gráfico de barras que apresenta o consumo para cada articulação para um passo com e sem elementos elásticos.

Desta forma, está mostrado que a utilização dos elementos elásticos nas articulações permite um decréscimo no consumo de energia. Neste caso, o consumo foi diminuído de 12.8 para 8.85 J, ou seja 30.8 %.

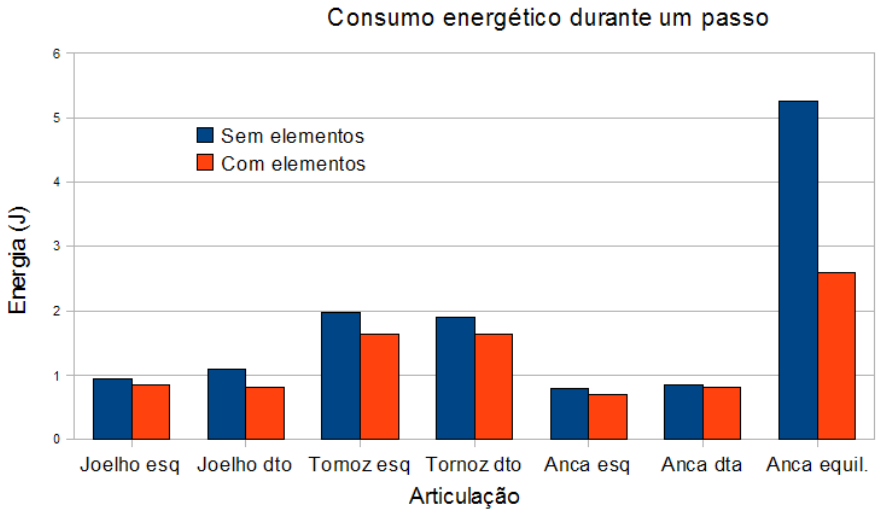


Figura 6.15: Gráfico de barras com consumos de cada articulação durante um passo com e sem elementos elásticos.

Capítulo 7

Conclusões

7.1 Síntese e conclusões do trabalho

O objectivo principal deste trabalho foi o desenvolvimento de novas metodologias para optimização do consumo energético de um robô humanóide. Para esse efeito foi desenvolvido um modelo realista de simulação, com base num modelo de física, que permite a implementação e teste dos métodos propostos. A optimização do planeamento de trajectórias poderá ser usada noutras áreas, uma vez que o problema foi abordado de uma forma genérica. Ao contrário dos simuladores existentes actualmente no mercado, que possuem a capacidade de processar dados referentes a robôs humanóides, o simulador desenvolvido é uma ferramenta personalizada e permite assim o controlo e alteração da arquitectura interna, como por exemplo ao nível mais baixo (modelo dos servos), tarefa que não é possível nos demais simuladores. Para além disso, o simulador permite criar ambientes com esteiras rolantes, robôs manipuladores, e robôs omnidireccionais com os quais os robôs humanóides podem interagir, sendo possível utilizar o simulador em prototipagem de diferentes tipos e arquitecturas de robôs com a capacidade de cooperação.

O trabalho foi planeado em três etapas principais: inicialmente obteve-se informação importante a respeito das características dinâmicas do robô real que serviu de base ao desenvolvimento do modelo de simulação para a construção de um modelo o mais realista possível. A implementação do modelo dos servos na simulação permite analisar formas de onda da corrente, tensão, potência e binário

associadas a cada servo, que possibilita o desenvolvimento dos algoritmos sob ponto de vista da optimização da energia consumida. Os parâmetros necessários à simulação foram obtidos quer por medidas directas (exemplo dos parâmetros do motor), quer por exploração sistemática apoiada no simulador, do valor óptimo para as constantes em função da resposta real (exemplo das constantes de atrito e ganhos do controlador). Uma vez apresentado o modelo de simulação de um robô humanóide, baseado numa plataforma real e a sua respectiva validação, foram desenvolvidos métodos de optimização de trajectórias que podem ser aplicados quer ao robô em causa, quer noutras áreas onde a optimização de trajectórias seja necessária. Numa fase de desenvolvimento deste tipo de novas metodologias, o recurso à simulação é um ponto-chave que permite não danificar o robô real nos estados iniciais de investigação.

Na etapa seguinte promoveu-se a optimização de trajectórias de baixo nível. Esta optimização tem por base calcular a evolução da trajectória durante um intervalo de tempo, respeitando as condições impostas pelo planeamento. Assim, foi possível garantir resultados onde se pode escolher a minimização da aceleração, velocidade ou energia consumida. A trajectória planeada e optimizada é enviada ao nível mais baixo onde se encontra o controlador dos servos.

Na última etapa, foi apresentada a implementação de um sistema passivo que aumenta a eficiência energética do robô durante a execução dos seus movimentos. Este sistema, que consiste na utilização de elementos elásticos nas articulações, permite obter menores consumos energéticos durante os movimentos do robô. Os parâmetros óptimos são determinados com base na exploração de soluções possíveis recorrendo à simulação. São apresentados exemplos que com a utilização do sistema passivo se consegue reduzir o consumo na ordem dos 30 %.

Uma vez que a robótica é uma área multidisciplinar, sendo necessário a aplicação de um vasto conjunto de conceitos e técnicas, este trabalho disponibiliza uma importante contribuição através de um conjunto de técnicas e algoritmos, testados e implementados, que poderão inclusive ser usados noutro tipo de problemas.

7.2 Trabalhos futuros

O desenvolvimento e pesquisa para o presente trabalho proporcionou novas perspectivas de desenvolvimento, tais como:

- A opção entre métodos de optimização poderá ser estudada para que, com base no estado do robô e na trajectória futura, dê origem a uma solução mais adequada. A escolha correcta do método de optimização, entre os apresentados neste trabalho, poderá dar origem a um novo estudo.
- Expandir o estudo dos métodos de optimização para casos com mais segmentos e com a aceleração descrita por funções que levem a trajectórias ainda mais suaves.
- A frequência de amostragem poderá depender da dinâmica da articulação. Este ponto tornar-se-ia vantajoso quando aplicado no robô real que permitiria partilhar recursos de uma forma melhor ao nível da comunicação com os servos.
- Projectar controladores dos servos com melhor resposta, sem atraso e sem erro em rampa na velocidade seria uma mais-valia para a implementação dos métodos propostos na realidade, por exemplo, implementar uma realimentação de estado nos servos. Este ponto requeria reprogramar o $firmware$ de cada servo.
- Para terminar, a implementação na prática dos conceitos de optimização desenvolvidos com base na simulação, permitiria obter robôs que apresentassem menor consumo energético e consequentemente maior autonomia.

Apêndice A

Descrição em XML da estrutura do simulador de Humanóides

```
<?xml version=”1.0”?>
    <robot>
     <kind value='humanoid'/>
      <solids>
       <cuboid> <!– bone–>
         <ID value='6'/>
         <pos x='-0.040' y='0' z='0.010'/>
         <pos x='0.082' y='0.102' z='0.100'/>
         <mass value='0.635'/>
         <desc Pt='Tronco'/>
         <desc Eng='Trunk'/>
       </cuboid>
       <cuboid>
         <ID value='20'/>
         <pos x='-0.035' y='0.033' z='-0.085'/>
         <size x='0.070' y='0.038' z='0.049'/>
         <mass value='0.143'/>
         <desc Pt='Coxa Esq L'/>
         <desc Eng='Left Thig L'/>
       </cuboid>
       <cuboid>
         <ID value='1'/>
```

```
      <pos x='-0.035' y='0.033' z='-0.240'/>
      <size x='0.070' y='0.038' z='0.049'/>
      <mass value='0.143'/>
      <desc Pt='Perna Esq'/>
      <desc Eng='Left Leg'/>
    </cuboid>
    <cuboid>
      <ID value='2'/>
      <pos x='-0.035' y='0.040' z='-0.280'/>
      <size x='0.100' y='0.060' z='0.033'/>
      <mass value='0.039'/>
      <desc Pt='Pe Esq'/>
      <desc Eng='Left Foot'/>
</cuboid>
    <cuboid>
      <ID value='3'/>
      <pos x='-0.035' y='-0.033' z='-0.085'/>
      <size x='0.070' y='0.038' z='0.049'/>
      <mass value='0.143'/>
      <desc Pt='Coxa Dir H'/>
      <desc Eng='Right Thig H'/>
    </cuboid>
    <cuboid>
      <ID value='4'/>
      <pos x='-0.035' y='-0.033' z='-0.240'/>
      <size x='0.070' y='0.038' z='0.049'/>
      <mass value='0.143'/>
      <desc Pt='Perna Dta'/>
      <desc Eng='Right Leg'/>
    </cuboid>
    <cuboid>
      <ID value='5'/>
      <pos x='-0.035' y='-0.040' z='-0.280'/>
      <size x='0.100' y='0.060' z='0.033'/>
      <mass value='0.039'/>
      <desc Pt='Pe Dto'/>
```

```
  <desc Eng='Right Foot'/>
</cuboid>
<cuboid>
  <ID value='7'/>
  <pos x='-0.030' y='0.033' z='-0.120'/>
  <size x='0.030' y='0.048' z='0.090'/>
  <mass value='0.173'/>
  <desc Pt='Coxa Esq L'/>
  <desc Eng='Left L Thigh'/>
</cuboid>
<cuboid>
  <ID value='8'/>
  <pos x='-0.030' y='-0.033' z='-0.120'/>
  <size x='0.030' y='0.048' z='0.090'/>
  <mass value='0.173'/>
  <desc Pt='Coxa Dta L'/>
  <desc Eng='Right L Thigh'/>
</cuboid>
<cuboid>
  <ID value='11'/>
  <pos x='-0.020' y='0.078' z='0'/>
  <size x='0.038' y='0.032' z='0.040'/>
  <mass value='0.080'/>
  <desc Pt='Braco Esq'/>
  <desc Eng='Left arm'/>
</cuboid>
<cuboid>
  <ID value='12'/>
  <pos x='-0.020' y='-0.078' z='0'/>
  <size x='0.038' y='0.032' z='0.040'/>
  <mass value='0.080'/>
  <desc Pt='Braco Dto'/>
  <desc Eng='Right Arm'/>
</cuboid>
<cuboid>
  <ID value='13'/>
```

```
  <pos x='-0.020' y='0.078' z='0.043'/>
  <size x='0.050' y='0.034' z='0.038'/>
  <mass value='0.082'/>
  <desc Pt='Dumm Braco Esq'/>
  <desc Eng='Dummy Left Arm'/>
</cuboid>
<cuboid>
  <ID value='14'/>
  <pos x='-0.020' y='-0.078' z='0.043'/>
  <size x='0.050' y='0.034' z='0.038'/>
  <mass value='0.082'/>
  <desc Pt='Dumm Braco Dto'/>
  <desc Eng='Dummy Roght Arm'/>
</cuboid>
<cuboid>
  <ID value='15'/>
  <pos x='-0.020' y='0.078' z='-0.074'/>
  <size x='0.037' y='0.032' z='0.126'/>
  <mass value='0.102'/>
  <desc Pt='AnteBraco Esq'/>
  <desc Eng='Left Forearm'/>
</cuboid>
<cuboid>
  <ID value='16'/>
  <pos x='-0.020' y='-0.078' z='-0.074'/>
  <size x='0.037' y='0.032' z='0.126'/>
  <mass value='0.102'/>
  <desc Pt='AnteBraco Dto'/>
  <desc Eng='Right Forearm'/>
</cuboid>
<cuboid>
  <ID value='17'/>
  <pos x='-0.030' y='0.033' z='-0.200'/>
  <size x='0.049' y='0.047' z='0.101'/>
  <mass value='0.077'/>
  <desc Pt='Joelho Esq'/>
```

```
      <desc Eng='Left Knee'/>
    </cuboid>
    <cuboid>
      <ID value='18'/>
      <pos x='-0.030' y='-0.033' z='-0.200'/>
      <size x='0.049' y='0.047' z='0.101'/>
      <mass value='0.087'/>
      <desc Pt='Joelho Dto'/>
      <desc Eng='Right Knee'/>
    </cuboid>
    <cuboid>
      <ID value='19'/>
      <pos x='-0.020' y='0' z='0.115'/>
      <size x='0.030' y='0.030' z='0.040'/>
      <mass value='0.100'/>
      <desc Pt='Cabeca'/>
      <desc Eng='Head'/>
    </cuboid>
  </solids>
  <articulations >
    <default >
      <draw radius='0.005' height='0.1' rgb24='8F0000'/ >
      <motor ri='8' ki='0.0075' vmax='10' imax='0.9' active='1'/ >
      <gear ratio='254'/ >
      <friction bv='0.01264' fc='0.000258' coulomblimit='-1'/ >
      <encoder ppr='1000' mean='0' stdev='0'/ >
      <controller mode='state' kp='30' ki='0' kd='-1.2' kf='0.05' active='1' period='10'/ >
      <spring k='0' zeropos='0'/ >
    </default >
    <joint >
      <ID value='0'/ >
      <pos x='-0.039' y='0.030' z='-0.170'/ >
      <axis x='0' y='1' z='0'/ >
      <connect B1='7' B2='17'/ >
      <limits Min='-98' Max='32'/ >
```

```
  <type value='Hinge'/ >
  <desc Pt='Joelho Esq'/ >
  <desc Eng='Left Knee'/ >
</joint >
<joint >
  <ID value='1'/ >
  <pos x='-0.027' y='0.030' z='-0.247'/ >
  <axis x='0' y='1' z='0'/ >
  <connect B1='17' B2='1'/ >
  <limits Min='-10' Max='125'/ >
  <type value='Hinge'/ >
  <desc Pt='Pe Esq Frt'/ >
  <desc Eng='Left Front ankle'/ >
</joint >
<joint >
  <ID value='2'/ >
  <pos x='-0.039' y='-0.030' z='-0.170'/ >
  <axis x='0' y='1' z='0'/ >
  <connect B1='8' B2='18'/ >
  <limits Min='-98' Max='32'/ >
  <type value='Hinge'/ >
  <desc Pt='Joelho Dto'/ >
  <desc Eng='Right Knee'/ >
</joint >
<joint >
  <ID value='3'/ >
  <pos x='-0.027' y='-0.030' z='-0.247'/ >
  <axis x='0' y='1' z='0'/ >
  <connect B1='18' B2='4'/ >
  <limits Min='-10' Max='125'/ >
  <type value='Hinge'/ >
  <desc Pt='Pe Dto Frt'/ >
  <desc Eng='Right Front ankle'/ >
</joint >
<joint >
  <ID value='4'/ >
```

```
    <pos x='-0.027' y='-0.030' z='-0.073'/ >
    <axis x='0' y='0' z='1'/ >
    <connect B1='6' B2='20'/ >
    <limits Min='-75' Max='75'/ >
    <type value='Hinge'/ >
    <desc Pt='Anca Esq Rot'/ >
    <desc Eng='Left Rot Hip'/ >
  </joint >
  <joint >
    <ID value='5'/ >
    <pos x='-0.027' y='0.030' z='-0.073'/ >
    <axis x='0' y='0' z='1'/ >
    <connect B1='6' B2='3'/ >
    <limits Min='-75' Max='75'/ >
    <type value='Hinge'/ >
    <desc Pt='Anca Dir Rot'/ >
    <desc Eng='Right Rot Hip'/ >
  </joint >
  <joint >
    <ID value='6'/ >
    <pos x='-0.022' y='0.030' z='-0.247'/ >
    <axis x='1' y='0' z='0'/ >
    <connect B1='1' B2='2'/ >
    <limits Min='-30' Max='100'/ >
    <type value='Hinge'/ >
    <desc Pt='Pe Esq Lat'/ >
    <desc Eng='Left Lat ankle'/ >
  </joint >
  <joint >
    <ID value='7'/ >
    <pos x='-0.022' y='-0.030' z='-0.247'/ >
    <axis x='1' y='0' z='0'/ >
    <connect B1='4' B2='5'/ >
    <limits Min='-30' Max='100'/ >
    <type value='Hinge'/ >
    <desc Pt='Pe Dto Lat'/ >
```

```
  <desc Eng='Right Lat ankle'/ >
</joint >
<joint >
  <ID value='8'/ >
  <pos x='-0.055' y='0.032' z='-0.103'/ >
  <axis x='1' y='0' z='0'/ >        <!– lateral– >
  <axis2 x='0' y='1' z='0'/ >        <!– rotação– >
  <connect B1='20' B2='7'/ >
  <limits Min='-90' Max='90'/ >        <!– lateral– >
  <limits2 Min='-75' Max='75'/ >        <!– rotação– >
  <type value='Universal'/ >
  <desc Pt='Anca Esq Lat/FT'/ >
  <desc Eng='Left Lat/FB Hip'/ >
</joint >
<joint >
  <ID value='9'/ >
  <pos x='-0.055' y='-0.032' z='-0.103'/ >
  <axis x='1' y='0' z='0'/ >        <!– lateral– >
  <axis2 x='0' y='1' z='0'/ >        <!– rotação– >
  <connect B1='3' B2='8'/ >
  <limits Min='-90' Max='90'/ >        <!– lateral– >
  <limits2 Min='-75' Max='75'/ >        <!– rotação– >
  <type value='Universal'/ >
  <desc Pt='Anca Dta Lat/FT'/ >
  <desc Eng='Right Lat/FB Hip'/ >
</joint >
<joint >
  <ID value='10'/ >
  <pos x='-0.043' y='0.033' z='0.053'/ >
  <axis x='0' y='1' z='0'/ >
  <connect B1='6' B2='13'/ >
  <limits Min='-120' Max='120'/ >
  <type value='Hinge'/ >
  <desc Pt='Braco Esq Frt'/ >
  <desc Eng='Left Front Shoulder'/ >
</joint >
```

```
<joint >
  <ID value='11'/ >
  <pos x='-0.031' y='0.079' z='0.036'/ >
  <axis x='1' y='0' z='0'/ >
  <connect B1='13' B2='11'/ >
  <limits Min='-160' Max='10'/ >
  <type value='Hinge'/ >
  <desc Pt='Braco Esq Lat'/ >
  <desc Eng='Left Lat Shoulder'/ >
</joint >
<joint >
  <ID value='12'/ >
  <pos x='-0.043' y='-0.033' z='0.053'/ >
  <axis x='0' y='1' z='0'/ >
  <connect B1='6' B2='14'/ >
  <limits Min='-120' Max='120'/ >
  <type value='Hinge'/ >
  <desc Pt='Braco Dto Frt'/ >
  <desc Eng='Right Front Shoulder'/ >
</joint >
<joint >
  <ID value='13'/ >
  <pos x='-0.031' y='-0.079' z='0.036'/ >
  <axis x='1' y='0' z='0'/ >
  <connect B1='14' B2='12'/ >
  <limits Min='-10' Max='160'/ >
  <type value='Hinge'/ >
  <desc Pt='Braco Dto Lat'/ >
  <desc Eng='Right Lat Shoulder'/ >
</joint >
<joint >
  <ID value='14'/ >
  <pos x='-0.031' y='0.079' z='-0.038'/ >
  <axis x='1' y='0' z='0'/ >
  <connect B1='11' B2='15'/ >
  <limits Min='-90' Max='90'/ >
```

```
    <type value='Hinge'/ >
    <desc Pt='Cotovelo Esq'/ >
    <desc Eng='Left Elbow'/ >
  </joint >
  <joint >
    <ID value='15'/ >
    <pos x='-0.031' y='-0.079' z='-0.038'/ >
    <axis x='1' y='0' z='0'/ >
    <connect B1='12' B2='16'/ >
    <limits Min='-90' Max='90'/ >
    <type value='Hinge'/ >
    <desc Pt='Cotovelo Dto'/ >
    <desc Eng='Right Elbow'/ >
  </joint >
  <joint >
    <ID value='16'/ >
    <pos x='-0.010' y='0' z='0.056'/ >
    <axis x='0' y='0' z='1'/ >
    <connect B1='6' B2='19'/ >
    <limits Min='-150' Max='150'/ >
    <type value='Hinge'/ >
    <desc Pt='Pescoco'/ >
    <desc Eng='Neck'/ >
  </joint >
</articulations >
```

Referências Bibliográficas

Atmel (2008). Atmel webpage. http://atmel.com. 17, 18

Bachar, Y. (2004). Developing controllers for biped humanoid locomotion. In *Master of Science, School of Informatics of University of Edinburgh*. 6, 30

Behnke, S., Mayer, N., Müller, J. & Schreiber, M. (2005). Nimbro 2004 team description. In *8th RoboCup International Symposium*. 13

Bishop, R.H. (2002). *The Mechatronics Handbook*. CRC Press. 44

Browning, B. & Tryzelaar, E. (2003). Übersim: A realistic simulation engine for robotsoccer. In *Proceedings of Autonomous Agents and Multi-Agent Systems, AAMAS'03*. 6, 30, 55

Buschmann, T., Lohmeier, S., Bachmayer, M., Ulbrich, H. & Pfeiffer, F. (2007). A collocation method for real-time walking pattern generation. In *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots*. 5

Buss, M., Hardt, M., Kiener, J., Sobotka, M., Stelzer, M., von Stryk, O. & Wollherr, D. (2003). Towards an autonomous, humanoid, and dynamically walking robot: Modeling, optimal trajectory planning, hardware architecture, and experiments. In *Proceedings of the 3rd International Conference on Humanoid Robots*. 4

Casti, J.L. (1998). *Would-Be Worlds: How Simulation is Changing the Frontiers of Science*. Wiley. 6

Chatzakos, P. & Papadopoulos, E. (2008). The effect of leg segmental proportions on the energetic cost of robotic locomotion. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 662–670. 5

Chevallereau, C., Formalsky, A. & Perrin, B. (1998). Low energy cost reference trajectories for a biped robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1398–1404. 3

Chung, G. (2008). Simple intuitive method for a planar biped robot to walk. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 687–694. 5

Costa, P.G. (2008). Sistema de simulação realista - simtwo. http://paginas.fe.up.pt/˜paco/wiki/index.php?n=Main.SimTwo. 29, 66

de Pina Filho, A.C. & Dutra, M.S. (2008). Using nonlinear oscillators to create a pattern generator of bipedal locomotion. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 937–944. 5

Eshghinejad, A. & Keshmiri, M. (2008). Devloping fast bipedal locomotion method for inclined floors. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 728–735. 5

Ferreira, J.P., Crisóstomo, M. & Coimbra, A.P. (2005). Controlo de um robô bípede com base em sensores de força nos pés. In *Proceedings of 9th Spanish-Portuguese Congress on Electrical Engineering*. 4

GLScene (2008). Glscene. http://glscene.sourceforge.net. 64

Harold, E.R. & Means, W.S. (2004). *XML in a Nutshell*. O'Reilly. 40

Höhn, O., Gacnik, J. & Gerth, W. (2005). Detection and classification of posture instabilities of bipedal robots. In *Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 409–416. 3

Hirai, K., Hirose, M., Haikawa, Y. & Takenaka, T. (1998). The development of honda humanoid robot. In *Proceedings of International Conference on Robotics and Automation*, 1321–1326. 3

Honda (2008). Honda asimo. http://world.honda.com/ASIMO/. 4

Huang, Q., Nakamura, Y., Arai, H. & Tanie, K. (2000). Development of a biped humanoid simulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1936 – 1942. 4, 6

i sobot (2008). http://www.isobotrobot.com/. 10

Johnson, C.R. & Hansen, C. (2004). *Visualization Handbook*. Elsevier Academic Press. 62

Kagami, S., Takahashi, Y., Nishiwaki, K., Mochimaru, M. & Mizoguchi, H. (2004). High-speed matrix pressure sensor for humanoid robot by using thin force sensing resistance rubber sheet. In *Proceedings of IEEE Sensors Conference*, 1534–1537. 6

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K. & Hirukawa, H. (2002). A realtiine pattern generator for biped walking. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 31–37. 5

Kajita, S., Morisawa, M., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K. & Hirukawa, H. (2006). Biped walking pattern generator allowing auxiliary zmp control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2994–2999. 3, 5, 67

Kaynov, D., Hernandez, A.A. & Balaguer, C. (2007). Joint control of a humanoid robot. In *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots*. 41, 42

Kim, J., Park, I. & Oh, J. (2007). Walking control algorithm of biped humanoid robot on uneven and inclined floor. *Journal of Intelligent and Robotic Systems*. 5

Kim, S.H. & Sankai, Y. (2005). Stair climbing task of humanoid robot by phase composition and phase sequence. In *IEEE International Workshop on Robot and Human Interactive Communication*, 531–536. 4

Knight, W. (2008). State-of-the-art robotics on display. http://www.newscientist.com. 4

Kroumov, V., Shibayama, K. & Inoue, A. (2003). Interactive learning tools for enhancing the education in control systems. In *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference.*. 31

Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M. & Inoue, H. (2003). Motion planning for humanoid robots. In *Proceedings of the 11th International Symposium of Robotics Research*. 5

Kumotek (2008). http://www.kumotek.com/. 10

Lee, C.S.G., Hirukawa, H. & Sugan, S. (2008). Ieee-ras technical committee on humanoid robots. http://www.humanoidrobots.org/. 4

Lei, X.S., Pan, J. & Su, J.B. (2005). Humanoid robot locomotion. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 882–887. 4

Lima, J.L., Gonçalves, J.C., Costa, P.G. & Moreira, A.P. (2006). Inverted pendulum virtual control laboratory. In *Proceedings of 7th Portuguese Conference on Automatic Control*. 31, 56

Lima, J.L., Gonçalves, J.C., Costa, P.G. & Moreira, A.P. (2008a). Humanoid robot simulation with a joint trajectory optimized controller. In *Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation*. 16, 30

Lima, J.L., Gonçalves, J.C., Costa, P.G. & Moreira, A.P. (2008b). Realistic behaviour simulation of a humanoid robot. In *Proceedings of 8th Conference on Autonomous Robot Systems and Competitions*. 53

Limited, F.A. (2001). Miniature humanoid robot, hoap-1. http://pr.fujitsu.com/en/news/2001/09/10.html#1. 4

Lopes, F.D. (2007). Projecto humanóide. In *Projecto final de curso, Departamento Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto*. 16, 17, 19, 25, 53

Lynxmotion (2008). http://www.lynxmotion.com/. 10

Maus, H.M., Rummel, J. & Seyfarth, A. (2008). Stable upright walking and running using a simple pendulum based control scheme. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 623–628. 5

McGeer, T. (1990). Passive walking with knees. In *Proceedings of the 1990 IEEE Robotics & Automation Conference*, 1640–1645. 3

Neo, E.S., Yokoi, K., Kajita, S., Kanehiro, F. & Tanie, K. (2005). A switching command-based whole-body operation method for humanoid robots. *IEEE/ASME Transactions on Mechatronics*. 3

Park, J.H. & Kim, K.D. (1998). Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *IEEE International Conference on Robotics and Automation*. 4

Popovic, M., Goswami, A. & Herr, H. (2005). Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *Int. Journal of Robotics Research*. 3

RoboBuilder.co., L. (2008). Robobuilder. http://www.robobuilder.net/eng/. 10

Robocup (2008). Robocup humanoid league. www.humanoidsoccer.org. 4

RoboPhilo (2008). Robophilo-programable humanoid in lifelike operation. http://www.robophilo.com/. 10

Robotis (2008a). http://www.robotis.com/. 10

Robotis, T. (2008b). Bioloid robotic kit. http://www.tribotix.com/. 10, 13

Roussel, L., de Wit, C.C. & Goswami, A. (1998). Generation of energy optimal complete gait cycles for biped robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2036–2041. 3

Santos, V. (2008). Humanoid project of university of aveiro. http://www2.mec.ua.pt/robotics/humanoid/. 4

Santos, V. & Silva, F. (2005). Engineering solutions to build an inexpensive humanoid robot based on a distributed control architecture. In *Proceedings of the IEEE/RSJ International Conference on Humanoid Robots*, 86–91. 6

Santos, V.M.F. & Silva, F.M.T. (2006). Design and low-level control of a humanoid robot using a distributed architecture approach. *Journal of Vibration and Control*. 13

Schroeder, W., Schroeder, W., Martin, K.W., Martin, K. & Lorensen, B. (1996). *The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics*. Prentice Hall. 62, 64

Shin, C.L., Li, Y.Z., Churng, S., Lee, T.T. & Cruver, W.A. (1990). Trajectory synthesis and physical admissibility for a biped robot during the single-support phase. In *Proceedings of International Conference on Robotics and Automation*. 3

Silva, F. & Santos, V. (2005). Towards an autonomous small-size humanoid robot: Design issues and control strategies. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 87–92. 5

Silva, M., Ferreira, P., Silva, F. & Santos, V. (2007). Local-level control of a humanoid robot prototype with force-driven balance. In *Proceedings of the IEEE 7th International Conference on Humanoid Robots*. 4, 25

Smith, R. (2008). Open dynamics engine. http://www.ode.org. 37, 56

Software, R. (2008). Pascal script 3.0. http://www.remobjects.com/ps. 67

Sony (2000). Small biped entertainment robot sdr-3x. http://www.sony.net/SonyInfo/News/Press_Archive/200011/00-057E2/. 4

Suleiman, W., Yoshida, E., Laumond, J.P. & Monin, A. (2007). On humanoid motion optimization. In *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots*. 3

Suzuki, T. & Ohnishi, K. (2006). Trajectory planning of biped robot with two kinds of inverted pendulums. In *Proceedings of 12th International Power Electronics and Motion Control Conference*, 396–401. 5

T. Ishida, E.R.S.C. (2004). Development of a small biped entertainment robot qrio. In *Proceedings of the 2004 International Symposium on Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society*, 23– 28. 4

Teodoro, P.D.D. (2007). Development of a simulation environment of an entertainment humanoid robot. In *Dissertação de Mestrado em Engenharia Mecânica, Instituto Superior Técnico, Lisboa*, 31–32. 5, 53

Teodoro, P.D.D. (2008). Humanoid robot: Development of a simulation environment of an entertainment humanoid robot. http://humanoids.dem.ist.utl.pt/. 4

Tomokuni, N., Saiga, M. & Yabuta, T. (2005). Distributed controller for cooperative joint torque control of compact humanoid robots. In *Proceedings of 5th IEEE-RAS International Conference on Humanoid Robots*, 259–264. 4

Tucker, A.B. (2004). *Computer science handbook*. CRC Press. 24

Versluys, R., Vanderborght, B., Ham, R.V., Beyl, P., Cherelle, P. & Lefeber, D. (2008). Energy-efficient humanoid walking with ankle actuation: Learning from biomechanics. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 1216–1224. 98

VUKOBRATOVIC, M. & JURICIC, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*. 3

VUKOBRATOVIC, M., BOROVAC, B., SURLA, D. & STOKIC, D. (1990). *Biped locomotion: Dynamics, Stability, Control and Aplication*. Springer-Verlag. 3

WANG, L., XIE, M., ZHONG, W., WANG, C. & ZHANG, L. (2008). Power analysis and structure optimization in the design of a humanoid robot. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 55–62. 5

XIE, M., ZHONG, Z.W., ZHANG, L., XIAN, L.B., WANG, L. & YANG, H.J. (2008). A deterministic way of planning and controlling biped walking of loch humanoid robot. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 1181–1190. 1

XIONG, R., ZHOU, C. & ZHANG, L. (2008). A convex optimization approach for online walking pattern generation. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 1207–1215. 5

ZARRUGH, M.Y. & RAIXLIFFE, C.W. (1979). Computer generation of human gait kinematics. *Journal of Biomechanics*. 3

ZHANG, L., ZHOU, C. & XIONG, R. (2008). A lie group formulation for realtime zmp detection using force/torque sensor. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 1250–1257. 6, 67

ZHENG, Y.F. & SHEN, J. (1990). Gait synthesis for the sd-2 biped robot to climb sloping surface. *IEEE Transactions on Robotics and Automation*. 3