



Universidade de Aveiro
2011

Departamento de Electrónica, Telecomunicações
e Informática

WILLIAM ERAS LAGE **Algoritmos de Controlo do Movimento para um
Robô Humanóide**



WILLIAM ERAS LAGE **Algoritmos de Controlo do Movimento para um
Robô Humanóide**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. Filipe Miguel Teixeira Pereira da Silva, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática e do Dr. Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Júri

Presidente

Doutor Alexandre Manuel Moutela Nunes da Mota
Professor Associado da Universidade de Aveiro

Vogais

Doutor José António Tenreiro Machado
Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Doutor Vítor Manuel Ferreira dos Santos
Professor Associado da Universidade de Aveiro

Doutor Filipe Miguel Teixeira Pereira da Silva
Professor Auxiliar da Universidade de Aveiro

Agradecimentos

Em primeiro lugar gostaria de agradecer aos meus orientadores Prof. Dr. Filipe Silva e ao Prof. Dr. Vítor Santos que todas as semanas disponibilizavam algum do seu tempo para me guiar e, por vezes, ajudar a resolver diversos problemas com que me deparei ao longo deste trabalho.

Agradeço ao Miguel Antunes da Fonseca Ribeiro que concluiu o seu curso de mestrado em Engenharia Electrónica e Telecomunicações com a dissertação “Desenvolvimento dos Sistemas Sensorial e Motor Para Um Robô Humanóide” (2010) e ao Ricardo Godinho finalista do curso de Engenharia Mecânica pela grande ajuda prestada na fase inicial do trabalho. O Miguel ajudou na parte do software Eagle, do MPLab e disponibilizou todos os datasheets que foram utilizados por ele e o Ricardo na componente mecânica associada aos aspectos de montagem e afinação do robô humanóide.

Fica aqui também um agradecimento ao Departamento de Engenharia Mecânica e aos elementos do Laboratório de Automação e Robótica pelas condições oferecidas para a realização do trabalho.

Por último mas com a maior relevância, quero ainda agradecer com todo o meu coração aos meus pais, Manuel António Lage e Martine Joséphine Eras Lage, por me terem ajudado toda a minha vida, tanto em termos pessoais como monetários. Tenho orgulho de ser filho desta boa gente e como tal dedico esta dissertação aos meus pais e ao meu irmão, Yoann Eras Lage.

Palavras-chave

Robótica humanóide, locomoção bípede, análise cinemática, planeamento de trajetórias, controlo do movimento, realimentação de força

Resumo

O Projecto Humanóide na Universidade de Aveiro (PHUA) deu um passo em frente com o início da construção, em 2009, de uma nova plataforma humanóide de pequenas dimensões e baixo custo baseada num inovador sistema de actuação híbrida. O trabalho centrou-se em questões ligadas ao projecto mecânico e à selecção e integração de actuadores e sensores. No ano seguinte os esforços foram dedicados ao desenvolvimento e integração de componentes de hardware e software que culminaram com o desenvolvimento dos sistemas sensorial e motor. Tendo como ponto de partida os trabalhos anteriores, esta dissertação pretende abordar os problemas do planeamento e controlo do movimento. Para o efeito, procedeu-se à análise cinemática da estrutura robótica com a derivação das equações de cinemática directa e inversa para as sub-estruturas dos braços e pernas. Estas equações são um elemento chave para a implementação dos algoritmos de controlo do movimento que contemplam o movimento ponto-a-ponto e o movimento contínuo. Foram realizadas várias experiências para avaliar o comportamento destes algoritmos, dos servomotores utilizados e da própria estrutura mecânica.

O trabalho culminou com a inclusão de sensores de força na planta do pé que permitem estimar a localização do centro de pressão. Apresentam-se os resultados preliminares dos testes efectuados com um algoritmo de controlo baseado em realimentação de força que asseguram o equilíbrio da perna de suporte colocada numa superfície de inclinação variável.

Keywords

Humanoid robotics, biped locomotion, kinematics analysis, trajectory planning, motion control, force feedback.

Abstract

The Humanoid Project at the University of Aveiro (HPUA) made a major step forward with the design and development, in 2009, of a new small-size and low cost humanoid platform based on hybrid actuation. The focus was on the mechanical design and the selection of actuators and sensors. The next steps were dedicated to the development and integration of hardware and software components, giving rise to the sensorial and motor systems.

Based on these previous works, this dissertation aims to pursue the problems of motion planning and control. In line with this, the work was devoted to the kinematics analysis of the robotics structure by deriving the direct and the inverse kinematics equations for both the arms and legs. These equations are essential in the implementation of the motion control algorithms, namely, the point-to-point and the continuous motion planning. Several experiments were carried out in order to evaluate and improve the performance of these algorithms, as well the servomotors' response.

This work concludes with the integration of four force sensors attached in the extremities of the foot sole and the evaluation of the centre of pressure. Some preliminary results are included showing the application of a force feedback control algorithm that balance the support leg on the top of a moving platform.

Contéudo

CONTÉUDO.....	XIII
ÍNDICE DE FIGURAS	XV
ÍNDICE DE TABELAS.....	XIX
CAPÍTULO 1 INTRODUÇÃO	1
1.1 Motivação e objectivos	2
1.2 Organização da dissertação.....	2
CAPÍTULO 2 ENQUADRAMENTO DO TRABALHO	3
2.1 Introdução	3
2.2 Etapas Importantes da História dos Humanóides Bípedes.....	3
2.3 Projecto Humanóide na Universidade de Aveiro (PHUA)	6
CAPÍTULO 3 ANÁLISE CINEMÁTICA DO ROBÔ HUMANÓIDE.....	9
3.1 Introdução	9
3.2 O problema da cinemática e do planeamento.....	9
3.2.1 Cinemática de estruturas série	9
3.2.2 Planeamento do movimento.....	11
3.3 Modelação da perna de suporte.....	12
3.4 Modelação da perna livre.....	15
3.5 Modelação do tronco e braços	16
CAPÍTULO 4 CONTROLO DO MOVIMENTO EM POSIÇÃO E VELOCIDADE.....	21
4.1 Introdução	21
4.2 Arquitectura do sistema.....	22
4.2.1 Comunicação entre a unidade central e a unidade slave	22
4.3 Módulos básicos de software.....	23
4.4 Modos de controlo do movimento.....	29
4.4.1 Controlo individual das juntas.....	29
4.4.2 Movimento ponto-a-ponto.....	31
4.4.3 Movimento contínuo	34
4.5 Optimização dos algoritmos.....	40
4.5.1 Erros nas leituras sensoriais.....	40
4.5.2 Correção na velocidade dos servos.....	43
4.6 Resultados experimentais	45

4.6.1	Movimento dos braços.....	45
4.6.2	Perna livre.....	47
4.6.3	Perna de suporte.....	48
CAPÍTULO 5	CONTROLO BASEADO NAS FORÇAS DE REACÇÃO.....	51
5.1	Introdução.....	51
5.2	Células de carga.....	51
5.3	Cálculo do centro de pressão.....	55
5.4	Algoritmo de controlo baseado nas forças de reacção.....	56
5.5	Alterações introduzidas na unidade slave.....	58
CAPÍTULO 6	CONCLUSÕES.....	61
6.1	Discussão dos resultados.....	61
6.2	Perspectivas de trabalho futuro.....	62
APÊNDICE A	– DESCRIÇÃO DE FUNÇÕES RELEVANTES.....	63
APÊNDICE B	– LIMITES E CONVERSÕES ANGULARES.....	66
REFERÊNCIAS		72

Índice de Figuras

Figura 2.1: Os primeiros robôs WL-1 (esquerda) e WL-3 (direita) desenvolvidos pela Universidade de Waseda nos anos sessenta (Waseda University, “Biped Walking robot”).	4
Figura 2.2: Robôs desenvolvidos pela Universidade de Waseda (da esquerda para a direita): WABOT-1, WL-10, WL-10RD e Wabian-2R.	5
Figura 2.3: O robô WHL-11 desenvolvido pela Hitachi (esquerda) e o robô Manny (direita) desenvolvido pela Battelle’s Pacific Northwest Lab (Android World, “Historical Android Projects”).	5
Figura 2.4: Série de robôs humanóides desenvolvidos pela Honda (Honda, “History Robot Development Process”).	6
Figura 2.5: Robô QRIO (esquerda) da Sony e dois robôs (direita) ligados ao projecto HRP-AIST (Wikipédia, “Humanoid Robot”).	6
Figura 2.6: Vistas frontal, lateral e posterior do robô humanóide com 25-gdl, 65 cm de altura, 6 kg de peso e um sistema híbrido de actuação.	7
Figura 2.7: Arquitectura parcial para teste de uma slave; sistema constituído pela unidade central de processamento, a unidade slave, os servomotores digitais e os sensores de força.	8
Figura 3.1: Relação entre o espaço das juntas e o espaço Cartesiano: dado o vector das juntas q , o correspondente r é unicamente determinado (cinemática directa); para um dado r , o correspondente q pode não existir e, mesmo quando existe, pode não ser único (cinemática inversa).	9
Figura 3.2: Exemplos do movimento realizado por um robô 2R: movimento ponto-a-ponto (esquerda) e movimento contínuo (direita) passando por um conjunto de pontos intermédios.	12
Figura 3.3: Representação dos sistemas de coordenadas atribuídos à perna de suporte (perna esquerda - E e perna direita - D) em que o sistema de coordenadas de referência S_R está situado no solo e o elemento terminal corresponde à anca (origem do sistema de coordenadas S_3).	13
Figura 3.4: Simulador em Matlab usado para validar as equações de cinemática da perna de suporte e os algoritmos de planeamento do movimento.	13
Figura 3.5: Representação dos sistemas de coordenadas atribuídos à perna livre (perna esquerda - E e perna direita, D) em que o sistema de coordenadas de referência S_R está situado na anca e o elemento terminal corresponde à extremidade do pé (origem do sistemas de coordenadas S_6).	15
Figura 3.6: Simulador em Matlab do movimento da perna livre (6-gdl).	15
Figura 3.7: Modelo completo dos membros inferiores (esquerda) e animação de um padrão de locomoção (direita) adaptado de Ferreira (2008).	16
Figura 3.8: Representação dos sistemas de coordenadas atribuídos aos braços ao longo da estrutura de elos mecânicos suportados pelo tronco. O sistema de coordenadas de referência S_R está situado na parte inferior do tronco e o elemento terminal corresponde à extremidade de cada braço, O_4 .	17
Figura 3.9: Simulador em Matlab do movimento da estrutura tronco e braços com a implementação das equações de cinemática.	17

$l_{11} = 7.7$	17
$l_{12} = 4.9$	
Figura 3.10: Restrições aplicadas às juntas θ_2 e θ_3 de acordo com a coordenada y_{et}	18
Figura 4.1: Sistema de controlo de baixo-nível constituído pela unidade slave e o controlador interno do servomotor.....	22
Figura 4.2: Diagrama de blocos da função de controlo individual das juntas.....	30
Figura 4.3: Diagrama de blocos do algoritmo de movimento ponto-a-ponto.....	32
Figura 4.4: As posições e as velocidades angulares (médias) calculadas na unidade slave são os parâmetros de entrada do módulo de geração de trajectórias implementado na unidade de controlo dos servomotores que gera a sequência de referências para um controlador PID.....	34
Figura 4.5: Controlo em velocidade impondo incrementos de 0.3% a intervalos regulares.....	34
Figura 4.6: Diagrama de blocos do algoritmo de movimento contínuo.....	35
Figura 4.7: Evolução temporal da velocidade de um servomotor (esquerda) ao realizar um movimento ponto-a-ponto (asteriscos azuis) e um movimento contínuo (asteriscos verdes); e dos servomotores da perna de suporte (direita) ao realizar um movimento contínuo.....	37
Figura 4.8: Exemplos em que a evolução temporal das posições angulares dos servos é monótona e o movimento bem sucedido (esquerda) e em que existe uma passagem intermédia pela posição angular final que dá origem a erros de seguimento (direita). Os pontos verdes indicam a evolução angular necessária para a realização do movimento em linha recta, enquanto os asteriscos verdes ilustram os erros angulares introduzidos pelo algoritmo anterior.....	37
Figura 4.9: Movimento contínuo entre dois pontos: evolução temporal das posições angulares (esquerda) usando o algoritmo inicial e representação da posição do elemento terminal no espaço Cartesiano (direita)...	38
Figura 4.10: Movimento contínuo entre dois pontos: evolução temporal das posições angulares (esquerda) usando o algoritmo corrigido e representação da posição do elemento terminal no espaço Cartesiano (direita).	39
Figura 4.11: Movimento contínuo na forma de losango: evolução temporal das posições angulares (esquerda) usando o algoritmo inicial e representação da posição do elemento terminal no espaço Cartesiano (direita)...	39
Figura 4.12: Movimento contínuo em forma de losango: evolução temporal das posições angulares (esquerda) usando o algoritmo corrigido e representação da posição do elemento terminal no espaço Cartesiano (direita).	40
Figura 4.13: Sequência de movimentos da perna com a velocidade máxima a 20%.....	40
Figura 4.14: Sequência de movimentos da perna com a velocidade máxima a 100%.	41
Figura 4.15: Sequência de movimentos da perna com a velocidade máxima a 20%, mas agora com um tempo maior entre as leituras de posição exigidas aos servomotores.....	42
Figura 4.16: Eliminação dos erros de leitura (esquerda) e aplicação a uma sequência de movimentos (direita).....	43
Figura 4.17: Comportamento dos servomotores sem e com a correcção no valor das velocidades angulares...	44
Figura 4.18: Variação na posição angular do servomotor ao realizar um movimento entre os seus limites ($\pm 90^\circ$) usando uma tensão de alimentação de 6 V e 7.4 V.	44

Figura 4.19: Posição angular final e tempo de execução obtidos à máxima velocidade para a repetição do mesmo movimento desejado (deslocamento angular de 60°). (Ribeiro, 2010).....	45
Figura 4.20: Evolução temporal das posições angulares (esquerda) para um movimento contínuo do braço esquerdo (direita).....	45
Figura 4.21: Evolução temporal das posições angulares (esquerda) para um movimento contínuo dos dois braços (direita).	46
Figura 4.22: Evolução temporal das posições angulares (esquerda) para um movimento composto do braço esquerdo (direita).....	46
Figura 4.23: Evolução temporal das posições angulares (esquerda) para um movimento composto da perna livre (direita).	48
Figura 4.24: Evolução temporal das posições angulares (esquerda) para um movimento composto da perna de suporte (direita).	48
Figura 5.1: Células de carga miniatura de 5lbs usadas no robô humanóide.	52
Figura 5.2: Localização das células de carga no pé. (Ribeiro, 2010)	53
Figura 5.3: Testes efectuados com os quatro sensores de força colocados nas extremidades da placa de acrílico que suporta as duas massas.	54
Figura 5.4: Evolução temporal das quatro forças normais de reacção (comportamento estático).....	54
Figura 5.5: Evolução temporal das quatro forças normais de reacção (comportamento dinâmico).....	54
Figura 5.6: Evolução temporal das forças de reacção (em cima) e do centro de pressão (em baixo) em resultado de se premir com o dedo directamente na placa de acrílico.	55
Figura 5.7: Representação da trajectória realizada pelo centro de pressão à medida que as forças de reacção variam.....	56
Figura 5.8: Testes efectuados numa superfície de inclinação variável com os quatro sensores de força colocados nas extremidades da placa de acrílico que por sua vez suporta uma perna.	56
Figura 5.9: Inclinação da superfície móvel segundo o eixo dos yy: os gráficos da coluna à esquerda representam a evolução do centro de pressão debaixo do pé e a evolução temporal das posições e velocidades angulares; os gráficos da coluna à direita mostram a evolução temporal das forças em cada extremidade do pé, a evolução temporal da localização do centro de pressão segundo os eixos dos xx e yy e os sinais de erro usados pelo controlador.....	57
Figura 5.10: Inclinação da superfície móvel segundo o eixo dos xx: os gráficos da coluna à esquerda representam a evolução do centro de pressão debaixo do pé e a evolução temporal das posições e velocidades angulares; os gráficos da coluna à direita mostram a evolução temporal das forças em cada extremidade do pé, a evolução temporal da localização do centro de pressão segundo os eixos dos xx e yy e os sinais de erro usados pelo controlador.....	58
Figura 5.11: Disposição dos dois componentes que foram adicionados ao esquema do circuito da unidade slave (o MCP4352, o ISL28470 e resistências).....	59
Figura 5.12: Desenho final da unidade slave em Eagle: board antiga (esquerda) e board nova (direita).....	60
Figura 5.13: Esquema do circuito da nova placa.....	60
Figura 6.1: Movimentos da articulação do tornozelo e coluna. (Sabino, 2010)	66

Figura 6.2: Movimentos da articulação do joelho. (Sabino, 2010) 68

Índice de Tabelas

Tabela 3.1: Parâmetros de Denavit-Hartenberg.....	13
Tabela 3.2: Comprimentos e massas dos elos.....	13
Tabela 3.3: Equações de cinemática inversa da perna de suporte; a variável \mathcal{A} assume os valores $(x_{et} - l_0)/c_1$ ou $-y_{et}/s_1$ de acordo com o quadrante de θ_1 para evitar divisões por zero.....	14
Tabela 3.4: Parâmetros de Denavit-Hartenberg para o cálculo da transformação homogênea $T_{S_0 \rightarrow S_6}$	15
Tabela 3.5: Equações de cinemática inversa da perna livre para $\theta_4 = 0$	16
Tabela 3.6: Parâmetros de Denavit-Hartenberg para o cálculo da transformação homogênea $T_{S_0 \rightarrow S_4}$	17
Tabela 3.7: Comprimento dos elos mecânicos.....	17
Tabela 3.8: Equações de cinemática inversa obtidas a partir da expressão geral quando $\theta_3 = 0$	18
Tabela 3.9: Algoritmo de procura de uma solução para a cinemática inversa.....	19
Tabela 4.1: Parâmetros da comunicação RS-232.....	22
Tabela 4.2: Tramas usadas na comunicação entre a unidade central (UC) e a unidade slave (US).....	22
Tabela 4.3: Funções associadas ao controlo do movimento das três sub-estruturas: dois braços, perna de suporte e perna livre.....	26
Tabela 4.4: Funções associadas especificamente ao controlo do movimento dos dois braços.....	27
Tabela 4.5: Funções associadas especificamente ao controlo do movimento da perna livre.....	27
Tabela 4.6: Funções associadas especificamente ao controlo do movimento da perna de suporte.....	29
Tabela 4.7: Atribuições de Id's aos diferentes servomotores no caso do movimento dos dois braços, da perna livre e da perna de suporte.....	33
Tabela 4.8: Variação nos erros de leitura variando o número de servos no barramento de comunicação.....	41
Tabela 5.1: Características técnicas das células de carga LBS-5.....	52
Tabela 6.1: Parâmetros das funções de cinemática e planeamento de trajectória para os braços.....	63
Tabela 6.2: Parâmetros das funções de cinemática e planeamento de trajectória para a perna livre.....	64
Tabela 6.3: Parâmetros das funções de cinemática e planeamento de trajectória para a perna de suporte.....	65
Tabela 6.4: Limites angulares das articulações do corpo humano e respectivos ângulos de cada servomotor.....	67
Tabela 6.5: Ganhos das transmissões de uma perna.....	68
Tabela 6.6: Juntas que têm o sentido oposto às do simulador (com respectivos Id's).....	68
Tabela 6.7: Ajustes dos ângulos das juntas por forma a obter os ângulos dos servomotores.....	69
Tabela 6.8: Limites das posições Cartesianas para a função de cada membro.....	70

Capítulo 1

Introdução

O campo da Robótica Humanóide apresenta enormes desafios envolvendo áreas multidisciplinares do conhecimento. Os problemas que se colocam no desenvolvimento de um robô humanóide englobam os relacionados com a locomoção, a navegação, a manipulação e a percepção. Do ponto de vista de projecto, trata-se de um sistema completo com grande potencial face à múltipla funcionalidade e ao vasto espectro de aplicações (*e.g.*, industriais, médicas, espaciais ou militares). A complexidade envolvida levanta um outro desafio que passa pela necessidade de reunir competências em áreas tão diversas como a biomecânica, a física, a mecânica, a electrónica e a computação.

Apesar da complexidade associada à robótica humanóide, os avanços verificados nos últimos anos, quer ao nível da tecnologia como de novas metodologias, têm contribuído para um envolvimento crescente de universidades e centros de investigação em todo o mundo (Solis and Takanishi, 2010). Em grande medida, os esforços são dedicados ao projecto e concepção de pequenos robôs humanóides de baixo custo (Furuta *et al.*, 2001; Yamasaki *et al.*, 2001; Kim *et al.*, 2004; Benke *et al.*, 2008) ou, noutros casos, ao desenvolvimento de competências de alto nível a implementar em plataformas comerciais como os robôs HOAP da Fujitsu (Miyachi, “Miniature Humanoid Robots”), o robô NAO da Aldebaran Robotics (Gouaillier *et al.*, 2009), ou o robô DarWin-OP (Muecke *et al.*, 2006). Ao mesmo tempo, a organização de competições internacionais como o FIRA e o RoboCup, criadas em 1995 e 1997, respectivamente, tem impulsionado fortemente o interesse pelos robôs bípedes e, em particular, os robôs humanóides.

O Projecto Humanóide na Universidade de Aveiro (PHUA) representa um esforço conjunto dos departamentos de Engenharia Mecânica e de Electrónica, Telecomunicações e Informática da Universidade de Aveiro. Nos últimos anos, o trabalho centrou-se em questões ligadas ao projecto mecânico e ao desenvolvimento e integração de componentes de hardware e software. Este trabalho culminou com a construção de uma plataforma humanóide autónoma assente numa arquitectura de controlo distribuída com comunicação por rede CAN (Santos e Silva, 2006; Ruas, 2008; Rodrigues, 2008). Com base na experiência anterior, no ano lectivo de 2008/2009 deu-se início ao projecto, concepção e construção de uma nova plataforma mais compacta e leve, com uma estrutura cinemática simplificada e com a inclusão de elementos passivos de actuação. Este trabalho culminou com a construção mecânica dos membros inferiores de uma nova plataforma humanóide (Sabino, 2009; Ribeiro, 2010; Santos *et al.*, 2010) e, mais recentemente, da estrutura completa incluindo o tronco, os braços e a cabeça (Godinho, 2011). Tendo como ponto de partida a nova plataforma humanóide, este projecto pretende abordar novos desafios tais como a actuação híbrida integrando elementos activos e passivos, o controlo híbrido combinando capacidades reactivas e deliberativas e novos conceitos de programação e controlo baseados em aprendizagem.

1.1 Motivação e objectivos

O plano de trabalhos desta dissertação enquadra-se no âmbito do projecto PHUA visando dar continuidade aos últimos desenvolvimentos com a integração de tecnologia mais recente e o desenvolvimento de novas soluções de engenharia (Ribeiro, 2010). Assim, os principais objectivos traçados para este trabalho foram os seguintes:

- Proceder à análise cinemática do robô humanóide derivando as equações de cinemática directa e inversa para as sub-estruturas relativas aos membros inferiores e superiores.
- Estudar as características dos servomotores utilizados e avaliar as possibilidades de implementação de controlo em velocidade e de malhas externas de controlo com realimentação.
- Implementar o protocolo de comunicações entre a unidade central de processamento (PC) e a unidade slave (microcontroladora) tendo em vista o controlo do movimento das duas sub-estruturas mencionadas acima.
- Implementar e testar na unidade de controlo local (*slave*), os módulos de software necessários à realização de movimentos dos braços, perna de suporte e perna livre.
- Proceder à integração dos sensores de força e à realização de testes experimentais de controlo baseados em realimentação de força.
- Redesenhar e testar a unidade slave tendo em vista a integração de novas funcionalidades.

1.2 Organização da dissertação

Este trabalho de dissertação está organizado por capítulos, da seguinte forma: O Capítulo 2 faz um resumo da história da robótica humanóide e enquadra o trabalho no âmbito do projecto PHUA. O Capítulo 3 apresenta a análise cinemática da estrutura robótica subdividindo-a em duas partes: membros inferiores e membros superiores. Para além da derivação dos modelos matemáticos, descrevem-se os algoritmos associados à realização de movimentos ponto-a-ponto e movimentos contínuos. O Capítulo 4 é dedicado à descrição dos diversos módulos software implementados no robô físico e, em particular, à avaliação do desempenho dos algoritmos de planeamento e controlo do movimento. O Capítulo 5 aborda a utilização dos sensores de força colocados nos pés do robô e apresenta alguns resultados preliminares da implementação de um controlador baseado na realimentação de força. Finalmente, o Capítulo 6 discute os pontos mais importantes do trabalho, apresenta as conclusões principais e traça perspectivas de trabalho futuro.

Capítulo 2

Enquadramento do Trabalho

2.1 Introdução

Já toda gente sonhou, quando era mais nova, em ter um robô capaz de fazer tudo aquilo que quiséssemos, seja para executar tarefas mais complicadas e/ou repetitivas como para ajudar em certas tarefas ou, simplesmente, para brincar. Contudo, facilmente nos apercebemos que ainda não existem robôs tão “generalistas” dotados de inteligência artificial adequada ou com uma estrutura mecânica ágil e, ao mesmo tempo, robusta para sobreviver no ambiente do nosso dia-a-dia. Apesar disso, já existem muitos robôs capazes de realizarem tarefas mais específicas como por exemplo: limpar o chão da casa, cortar a relva, robôs capazes de levantar grandes pesos, auxiliar pessoas a caminhar, entre outros exemplos que poderiam ser mencionados. Aos poucos vamos caminhando para a concretização dos nossos sonhos, ou seja, criar um robô à nossa imagem e se possível melhor do que nós em certos aspectos.

A maioria dos robôs não se assemelha ao ser humano em grande medida porque a locomoção bípede é ainda um tópico de investigação em aberto. Desta forma, uma grande parte dos robôs actuais apresenta rodas que facilitam em muito o seu deslocamento em superfícies planas (com a desvantagem de não poderem ultrapassar certos obstáculos, como por exemplo, degraus). Por outro lado, mesmo as tarefas realizadas de forma aparentemente simples pelos humanos (*e.g.*, chutar uma bola ou correr), que crescem e aprendem ao longo do tempo com as experiências passadas, tornam-se muito difíceis de implementar num robô. Apesar de todas as dificuldades de nível tecnológico e científico, têm-se verificado avanços muito significativos no projecto, concepção e desenvolvimento de plataforma humanóides integradas.

Neste capítulo apresenta-se, em primeiro lugar, uma breve descrição da história da robótica humanóide destacando alguns dos trabalhos mais significativos. Em segundo lugar, faz-se o enquadramento deste trabalho de dissertação no âmbito do projecto PHUA.

2.2 Etapas Importantes da História dos Humanóides Bípedes

A robótica humanóide tem uma longa história ligada por completo à evolução da tecnologia e, em certa medida, aos filmes de ficção científica. Procura-se nesta subsecção realçar alguns marcos importantes neste caminho que nos leva aos dias de hoje. O primeiro destaque é dado ao primeiro autómato, que se parecia com um cavaleiro de armadura, desenhado pelo bem conhecido Leonardo da Vinci em 1495 e que ficou conhecido como o robô de Leonardo (Wikipédia, “Humanoid Robot”). Mais tarde, em 1942 Issac Asimov, autor de ficção científica, formula as três leis da robótica (*The Three Laws of Robotics*):

1. Um robô não pode ferir um ser humano ou, por noção, permitir que um ser humano sofra algum mal.

2. Um robô deve obedecer a todas as ordens dadas a ele por seres humanos, excepto quando essas ordens entram em conflito com a Primeira Lei.
3. Um robô deve proteger sua própria existência enquanto tal protecção não entre em conflito com a Primeira e Segunda Leis.

O primeiro modelo de uma perna robótica, o WL-1, foi criado em 1966 na Universidade de Waseda (Tóquio, Japão). Entretanto, vários estudos e progressos foram sendo realizados nesta universidade nos anos seguintes. Em 1968, o WL-3, foi uma máquina criada para caminhar usando um modo Master/Slave (ver Figura 2.1). De 1969 a 1971, foram criados o WAP-1, WAP-2 e WAP-3 (Waseda University, “Biped Walking Robot”), tendo sido testada a introdução de músculos artificiais. Ainda na mesma universidade, foi criado, em 1973, o primeiro robô antropomórfico em grande escala (o Wabot-1), sendo considerado um marco na história da robótica humanóide (Waseda University, “WABOT – WAseda roBOT”). Este robô bípede era constituído por um sistema de controlo dos membros, um sistema de visão e um sistema de conversação. Podem-se destacar as suas capacidades de locomoção, de manipulação, de transporte de objectos, com base em sensores tácteis nas mãos, de comunicação com pessoas em japonês e de medição de distâncias a objectos usando receptores externos (microfones, câmaras e altifalantes). Os desenvolvimentos seguintes deram origem ao caminhar quase-dinâmico (1980) com o robô WL-9DR, ao caminhar em superfícies planas (1983) com o WL-10R, à marcha dinâmica (1984) com o modelo WL-10RD e a padrões de locomoção mais próximos do humano (2004) com a geração mais actual Wabian-2 (ver Figura 2.2).

Criado no Japão em 1985, o WHL-11 (Waseda Hitachi Leg 11), um robô bípede desenvolvido pela Hitachi Ltd com base no WL-10R, estava equipado com um computador e uma bomba hidráulica instalada. Realizava um andar estático em superfícies planas demorando 13 segundos por passo (Figura 2.3). Nos Estados Unidos, entre 1986 e 1989, foi desenvolvido o Manny (Figura 2.3), um manequim de escala antropomórfica com 42 graus de liberdade (gdl) desenvolvido no Battelle’s Pacific Northwest Laboratories (Android World, “Historical Android Projects”).

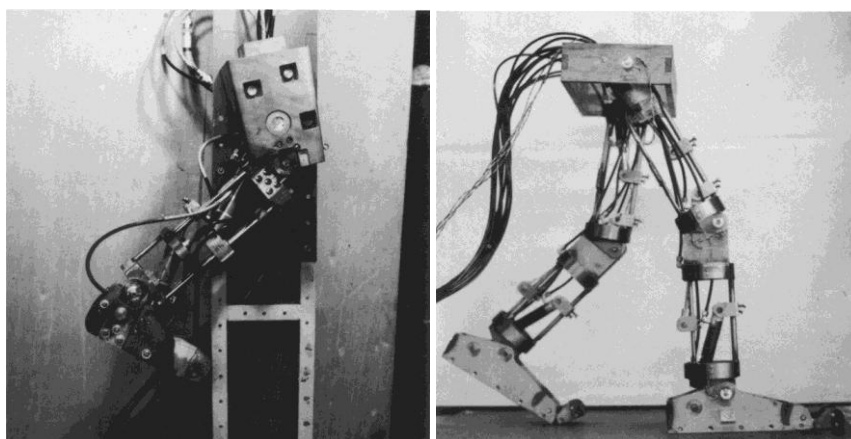


Figura 2.1: Os primeiros robôs WL-1 (esquerda) e WL-3 (direita) desenvolvidos pela Universidade de Waseda nos anos sessenta (Waseda University, “Biped Walking robot”).

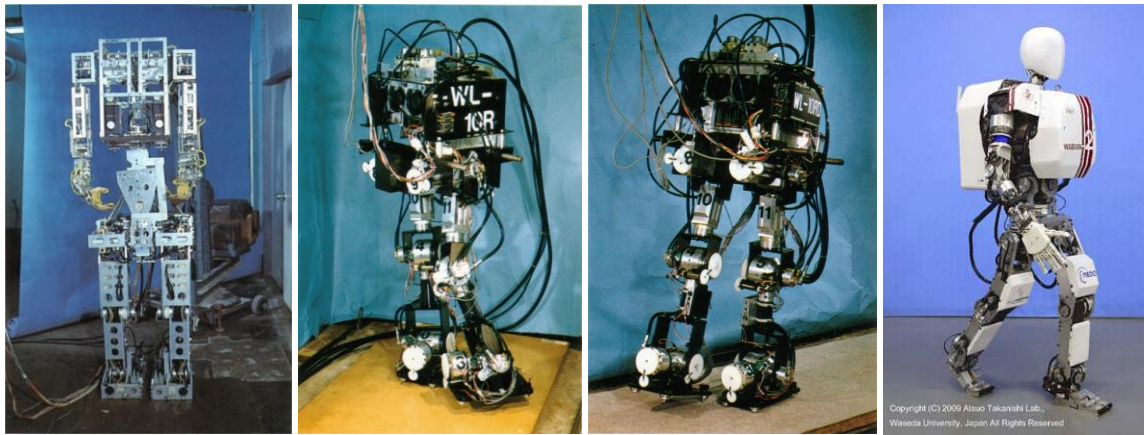


Figura 2.2: Robôs desenvolvidos pela Universidade de Waseda (da esquerda para a direita): WABOT-1, WL-10, WL-10RD e Wabian-2R.



Figura 2.3: O robô WHL-11 desenvolvido pela Hitachi (esquerda) e o robô Manny (direita) desenvolvido pela Battelle's Pacific Northwest Lab (Android World, "Historical Android Projects").

Em 1987, a Honda Motor Company iniciou um projecto de grande dimensão ligado ao desenvolvimento de robôs humanóides. Durante os primeiros sete anos foram desenvolvidos robôs bípedes designados por E0 a E6 (Honda, "History Robot Development Process"). Apenas em 1993 a Honda divulgou os primeiros resultados com a apresentação do robô P1. Este robô era autónomo utilizando baterias que asseguravam o seu funcionamento durante 15 minutos. Foi sem dúvida um grande avanço e um marco assinalável do ponto de vista tecnológico. Desde então a Honda deu a conhecer ao mundo os robôs P2, P3 e o ASIMO. A Figura 2.4 ilustra os vários modelos de robôs bípedes desenvolvidos pela Honda ao longo dos últimos anos. Já em 2011, a Honda exibiu o ASIMO com funcionalidades melhoradas do ponto de vista de adaptação a novas situações (Honda, "The Honda Humanoid Robot Asimo"). Para além das capacidades anteriores de reconhecer objectos e apreender os seus nomes, associar através da lógica a família à qual o objecto pertence, de andar e correr, foram incluídas novas funcionalidades tais como o saltar e equilibrar-se numa perna, chutar uma bola de futebol com bastante força, ou conseguir prever o percurso das pessoas para se desviar delas e evitar colisões.

Em 2001, outra companhia Japonesa, a Sony, revelava um pequeno humanóide de entretenimento chamado Sony Dream Robot (SDR) cuja evolução deu lugar, em 2003, ao Qrio (Figura 2.5). No mesmo ano, a Fujitsu lançava o seu primeiro robô humanóide comercial com o nome HOAP-1. Os sucessores, HOAP-2 e HOAP-3, foram anunciados em 2003 e 2005, respectivamente. Na mesma linha, é lançado em 2004 pela

companhia Kondo Kagaku o KHR-1, um pequeno robô humanóide programável. Finalmente, destaca-se o projecto HRP que envolve esforços da indústria (Kawada Industries Inc) e da academia (National Institute of Advanced Industrial Science and Technology) visando explorar o potencial de aplicação dos humanóides. O último robô apresentado em 2009, o HRP-4C, permite reproduzir movimentos faciais e da cabeça similares aos de um humano, para além do caminhar bípede.



Figura 2.4: Série de robôs humanóides desenvolvidos pela Honda (Honda, “History Robot Development Process”).



Figura 2.5: Robô QRIO (esquerda) da Sony e dois robôs (direita) ligados ao projecto HRP-AIST (Wikipédia, “Humanoid Robot”).

2.3 Projecto Humanóide na Universidade de Aveiro (PHUA)

O Projecto Humanóide na Universidade de Aveiro representa um esforço conjunto dos departamentos de Engenharia Mecânica e de Electrónica, Telecomunicações e Informática. Nos últimos anos, o trabalho centrou-se em questões ligadas ao projecto mecânico e ao desenvolvimento e integração de componentes de hardware e software. Este trabalho culminou com a construção de uma primeira plataforma humanóide autónoma assente numa arquitectura de controlo distribuída com comunicação por rede CAN (Santos e Silva, 2006; Ruas, 2008). Esta plataforma robótica permitiu validar varias soluções da engenharia e, ao mesmo tempo, avaliar os pontos francos do robô, tais como a existência de folgas mecânicas, a baixa eficiência energética e a baixa fiabilidade dos sensores de força. Com base na experiência anterior, Setembro de 2008 marcou o início da segunda fase do projecto visando abordar novos desafios, tais como a actuação híbrida

integrando elementos activos e passivos, o controlo híbrido combinando capacidades reactivas e deliberativas e conceitos de programação e controlo baseados em aprendizagem.

Neste sentido, deu-se início ao projecto, concepção e construção de uma nova plataforma mais compacta e leve, com uma estrutura cinemática simplificada e com a inclusão de elementos passivos de actuação (Sabino, 2009). A estrutura mecânica compreende um conjunto de juntas que copiam os graus de liberdade mais importantes do corpo humano. As soluções propostas incluem a utilização de juntas universais e mecanismos simples para ajustar as polias e a tensão das correias de transmissão. As proporções e gamas de variação angular antropomórficas servem de inspiração para esta plataforma. Do ponto de vista da tecnologia, importa realçar a substituição dos servomotores analógicos por digitais e a inclusão de células e carga para medição das forças de reacção no solo. A estrutura final do robô humanóide (ver Figura 2.6) tem 65 cm de altura, pesa cerca de 6 kg e apresenta um total de 25-gdl divididos da seguinte forma: 2×2-gdl no tornozelo, 2×1-gdl no joelho, 2×3-gdl na anca, 3-gdl no tronco, 2-gdl no pescoço, 2×3-gdl no ombro e 2×1-gdl no cotovelo. A maioria das juntas apresenta um sistema de actuação híbrida combinando um servomotor com elementos elásticos que proporcionam um mecanismo de armazenamento e recuperação de energia.

Este trabalho de dissertação enquadra-se no projecto PHUA e visa dar seguimento aos esforços de implementação e teste dos sistemas de comunicação e controlo realizados anteriormente (Ribeiro, 2010). A arquitectura global do sistema inclui a unidade central de processamento (para já é o PC a correr Matlab, uma unidade externa de interface e teste, mas no futuro será diferente, com um PC104), a unidade slave baseada num microcontrolador dsPIC, os servomotores digitais acoplados a cada junta usando transmissão por correias dentadas e os sensores de força baseados em células de carga e que fornecem medidas precisas das forças de reacção no solo.

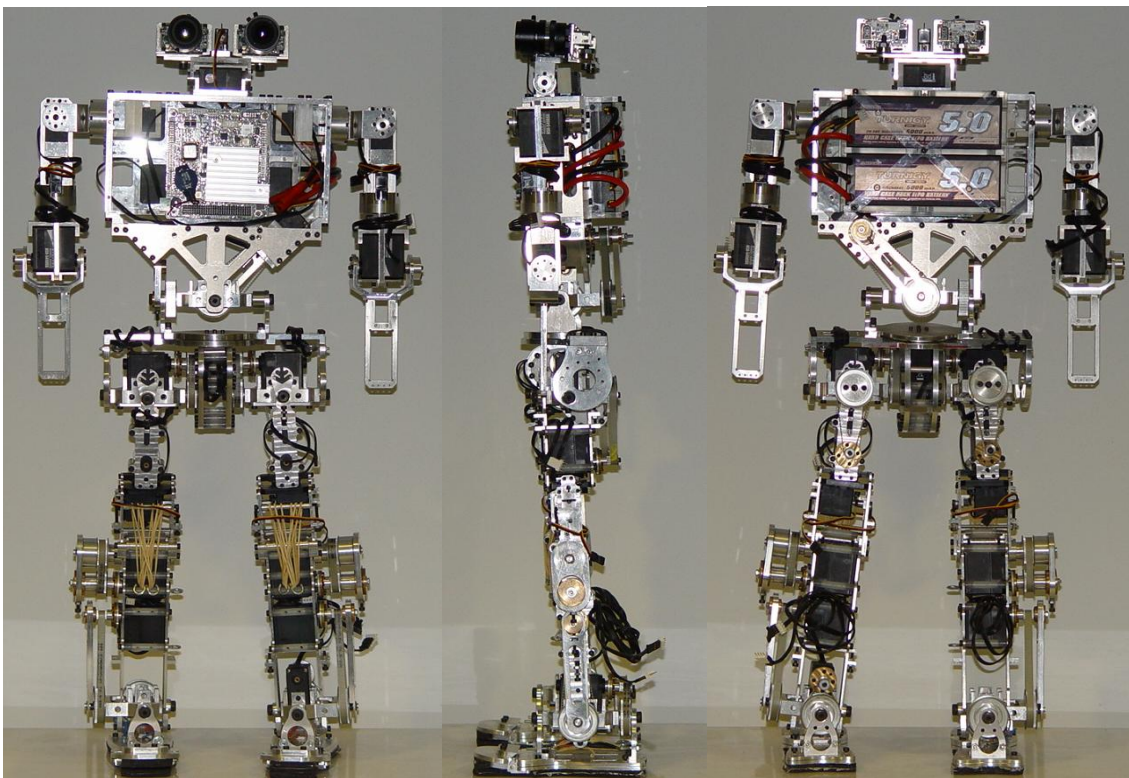


Figura 2.6: Vistas frontal, lateral e posterior do robô humanóide com 25-gdl, 65 cm de altura, 6 kg de peso e um sistema híbrido de actuação.

A Figura 2.7 ilustra os vários sub-sistemas que fazem parte deste trabalho, os processos de comunicação entre eles e o respectivo fluxo de informação. O objectivo central é desenvolver um conjunto de módulos software relacionados com a programação e o controlo do movimento do robô, assim como com a leitura sensorial a enviar para a unidade central tendo em vista a possibilidade de se estimar o estado do sistema em cada instante. Apesar da sua importância no âmbito do projecto PHUA, nesta dissertação não se pretendem abordar os problemas do controlo distribuído e da actuação híbrida. Assim, ao longo do trabalho será usada apenas uma unidade slave (a reformular) e a actuação será integralmente devida aos elementos activos (os servomotores digitais).

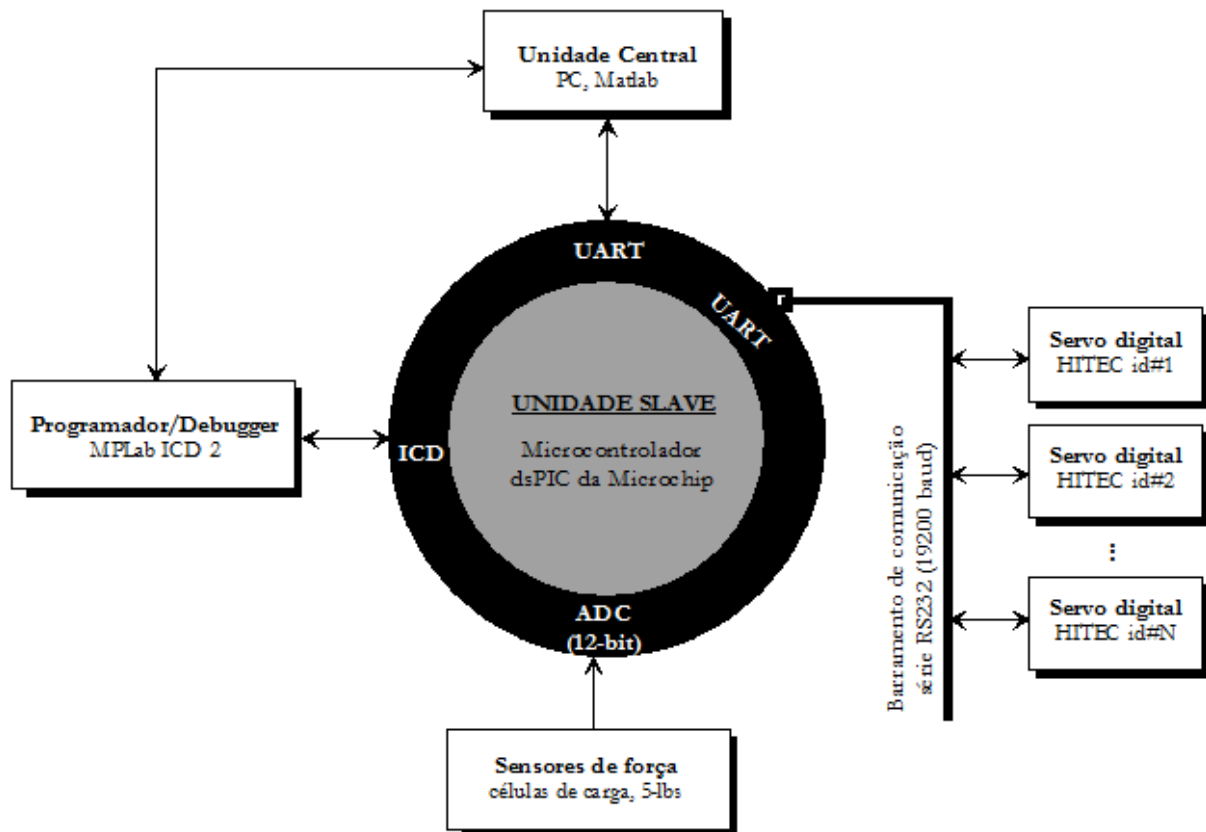


Figura 2.7: Arquitectura parcial para teste de uma slave; sistema constituído pela unidade central de processamento, a unidade slave, os servomotores digitais e os sensores de força.

Capítulo 3

Análise Cinemática do Robô Humanóide

3.1 Introdução

Este capítulo é dedicado à análise cinemática da estrutura humanóide que compreende o estudo geométrico do movimento dos elos mecânicos do robô em termos de posição, velocidade e aceleração. Em primeiro lugar, faz-se uma breve descrição dos problemas associados à análise cinemática de uma estrutura articulada e ao planeamento do movimento usando funções polinomiais. Em segundo lugar, derivam-se as equações de cinemática directa e inversa sub-dividindo o sistema completo em três estruturas: a perna de suporte, a perna livre e os dois braços suportados pelo tronco. A validação destas equações foi realizada com base em simuladores desenvolvidos em Matlab que permitem efectuar a animação do movimento individual destas três sub-estruturas. A integração das duas pernas levou à construção de um modelo com 12-gdl para o qual se definiu um padrão de locomoção baseado na especificação de um conjunto de parâmetros cinemáticos. As simulações efectuadas ajudaram na implementação dos vários módulos de software no microcontrolador tendo em conta a realização de experiências com o robô físico.

3.2 O problema da cinemática e do planeamento

3.2.1 Cinemática de estruturas série

O objectivo desta subsecção é descrever o método utilizado para derivar os modelos matemáticos que relacionam o espaço Cartesiano em que, tipicamente, uma tarefa é especificada com o espaço das juntas em que o sistema é actuado (ver Figura 3.1). Mais concretamente pretende-se estabelecer uma relação entre a posição e orientação do elemento terminal do robô e as variáveis das juntas. Para o efeito, considere-se o caso geral de uma estrutura composta por n -elos mecânicos rígidos interligados em série por n -juntas articuladas a operar no espaço 3-D.

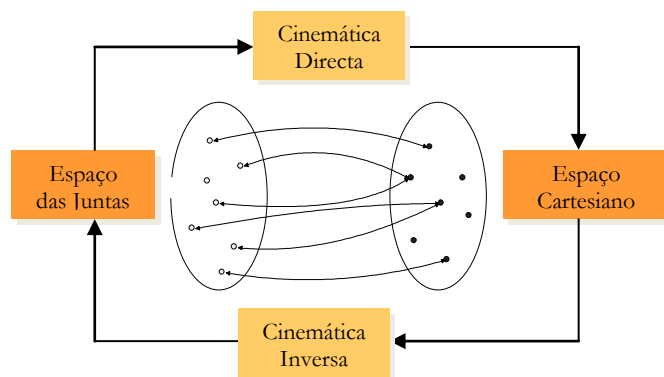


Figura 3.1: Relação entre o espaço das juntas e o espaço Cartesiano: dado o vector das juntas q , o correspondente r é unicamente determinado (cinemática directa); para um dado r , o correspondente q pode não existir e, mesmo quando existe, pode não ser único (cinemática inversa).

O vector das juntas é expresso por $q = [q_1, q_2, \dots, q_n]$, enquanto a posição e orientação do elemento terminal são descritas por um vector m -dimensional expresso por $r = [r_1, r_2, \dots, r_m]$. O problema de obter r para um dado q traduz-se na cinemática directa de posição:

$$r = f(q) \quad (3.1)$$

A função f é uma transformação não linear do espaço das juntas para o espaço Cartesiano que depende dos parâmetros cinemáticos do robô (*e.g.*, comprimento dos elos mecânicos). O problema inverso de obter q para um dado r é designado por problema da cinemática inversa e pode ser expresso simbolicamente por:

$$q = f^{-1}(r) \quad (3.2)$$

Enquanto o problema da cinemática directa é unicamente determinado, as equações de cinemática inversa podem não ter solução (*e.g.*, posição Cartesianas encontra-se fora dos limites do volume de trabalho) e, mesmo quando esta existe, pode não ser única. Outra questão que merece atenção é o problema da eliminação de redundâncias: no caso geral em que $n > m$ o sistema diz-se redundante uma vez que possui mais graus de liberdade do que os necessários para realizar uma dada tarefa desejada; mesmo na situação em que $n = m$ pode existir mais que uma configuração possível para o robô atingir uma dada posição e/ou orientação do elemento terminal.

Derivando a equação (3.1) em ordem ao tempo obtém-se as equações de cinemática directa relativas às velocidades e acelerações (cinemática directa diferencial) que expressam relações lineares com a ajuda da matriz do Jacobiano:

$$\begin{aligned} \dot{r} &= J(q)\dot{q} \\ \ddot{r} &= J(q)\ddot{q} + \dot{J}(q)\dot{q} \end{aligned} \quad (3.3)$$

As correspondentes equações de cinemática inversa são obtidas invertendo as relações anteriores:

$$\begin{aligned} \dot{q} &= J^{-1}(q)\dot{r} \\ \ddot{q} &= J^{-1}(q)\ddot{r} + \dot{J}^{-1}(q)\dot{r} \\ &= J^{-1}(q)[\ddot{r} + \dot{J}(q)\dot{r}] \end{aligned} \quad (3.4)$$

Apesar de, por exemplo, para um robô com 2-gdl no plano, a solução da cinemática directa e inversa se resumir a um problema de simples geometria, a derivação de uma solução torna-se cada vez mais complicada à medida que os gdl aumentam e se opera no espaço tridimensional. Uma forma de contornar o problema é atribuir um sistema de coordenadas apropriado a cada elo mecânico e descrever a relação entre elos pela relação entre estes sistemas de coordenadas. Desta forma, é possível derivar de forma sistemática a cinemática directa e obter analiticamente ou numericamente a solução da cinemática inversa. Neste capítulo foi usado o algoritmo de Denavit-Hartenberg (D-H) descrito em (Santos, 2003) para calcular a cinemática directa de forma recursiva. Para um sistema articulado com n -gdl serão atribuídos sistemas de coordenadas desde a base (sistema S_0) até ao elemento terminal (sistema S_n). A posição e a orientação do sistema de coordenadas S_i em relação ao sistema S_{i-1} pode ser estabelecida por um conjunto de quatro parâmetros e quatro

transformações geométricas (duas rotações e duas translações elementares) que, uma vez multiplicadas, dão origem a uma única matriz de transformação homogénea (transformação de coordenadas $T_{S_{i-1} \rightarrow S_i}$).

As equações de cinemática directa são obtidas pela multiplicação das matrizes de transformação homogéneas individuais:

$$T_{S_0 \rightarrow S_n}(q) = T_{S_0 \rightarrow S_1} \cdot T_{S_1 \rightarrow S_2} \cdots T_{S_{n-1} \rightarrow S_n} \quad (3.5)$$

Por sua vez, as equações de cinemática inversa foram derivadas analiticamente por manipulação das equações de cinemática directa e usando, regra geral, a solução da expressão $k_1 \sin \theta + k_2 \cos \theta = k_3$ dada por:

$$\theta = \operatorname{tg}^{-1} \left(\frac{k_1}{k_2} \right) \pm \operatorname{tg}^{-1} \left(\frac{\sqrt{k_1^2 + k_2^2 - k_3^2}}{k_3} \right) \quad (3.6)$$

em que k_1, k_2 e k_3 são constantes conhecidas que satisfazem a condição $k_1^2 + k_2^2 \geq k_3^2$.

3.2.2 Planeamento do movimento

O planeamento de trajectórias consiste na especificação, espacial e temporal, do movimento do robô em relação a um sistema de coordenadas de referência. Tipicamente, o planeamento é expresso em coordenadas Cartesianas, por ser mais intuitivo, e tem como principal função especificar um conjunto de variáveis desejadas (denominadas referências) como entrada para o controlador de baixo nível (*e.g.*, posição angular ou velocidade angular desejada de uma junta). Desta forma, as relações de cinemática inversa são necessárias para converter a especificação da tarefa no espaço Cartesiano para variáveis das juntas associadas às acções de controlo.

Ao longo deste trabalho foram implementados dois tipos de movimentos: (1) o movimento ponto-a-ponto que corresponde ao deslocamento do elemento terminal entre dois pontos no espaço operacional; e (2) o movimento contínuo que corresponde à execução de uma trajectória em linha recta entre dois pontos. Por um lado, no movimento ponto-a-ponto o objectivo é mover o elemento terminal de uma localização inicial para uma localização final. O movimento resultante do elemento terminal não é facilmente previsível dada os efeitos não lineares introduzidos pela cinemática directa. O problema pode ser articulado em dois sub-problemas: primeiro, a cinemática inversa é resolvida off-line para transformar requisitos do espaço Cartesiano (*i.e.*, as localizações inicial e final) nas respectivas variáveis do espaço das juntas. Segundo, um algoritmo de planeamento de trajectórias no espaço das juntas gera uma sequência temporal de valores para um determinado intervalo de tempo (*i.e.*, o movimento é planeado nas variáveis controladas). Como referido no Capítulo 2, os servomotores digitais implementam, internamente, uma função de geração de trajectórias baseado em perfis trapezoidais que geram as referências para o sistema de controlo. Por outro lado, no movimento contínuo o objectivo é mover o elemento terminal entre duas posições (orientações) ao longo de um caminho pré-estabelecido (*e.g.*, em linha recta). Mais uma vez, o problema pode ser articulado em dois sub-problemas: primeiro, um algoritmo de planeamento de trajectórias no espaço Cartesiano gera as sequências temporais de valores. Segundo, a cinemática diferencial inversa é calculada on-line com base no Jacobiano para se obter a sequência de valores no espaço das juntas. A Figura 3.2 ilustra o resultado da simulação dos dois tipos de movimentos num robô 2R a operar no plano.

O custo computacional introduzido pelo módulo de geração de trajectórias no espaço operacional e a inversão cinemática on-line é, em geral, bastante superior ao do movimento ponto-a-ponto, colocando limites na taxa de amostragem aquando da implementação no microcontrolador. Ao mesmo tempo, a implementação do movimento contínuo, assente na inversão on-line da cinemática diferencial, deve atender ao problema das singularidades e aos efeitos de drift. As singularidades cinemáticas correspondem a configurações das juntas angulares para as quais a matriz do Jacobiano é singular (não tem inversa). Nestas circunstâncias, o movimento do elemento terminal em certas direcções dá origem a velocidades nas juntas muito elevadas (em teoria infinitas). Ao mesmo tempo, os fenómenos de *drift*, em que a localização do elemento terminal se afasta da trajectória desejada, resulta do facto das variáveis do espaço Cartesiano serem controladas em malha aberta. Uma forma de contornar o problema é usar algoritmos de cinemática inversa em malha fechada que levam em conta o erro entre a posição/orientação actual e a desejada do elemento terminal (Sciavicco e Siciliano, 1996).

Qualquer que seja o tipo de movimento usado, é necessário encontrar algoritmos de planeamento que gerem trajectórias apropriadas, nomeadamente, que assegurem continuidade nas variáveis de velocidade e aceleração tendo em vista a realização de movimentos suaves. Neste trabalho foram usadas funções polinomiais de 3ª ordem que asseguram a continuidade nas velocidades iniciais e finais, e de 5ª ordem que acrescentam continuidade nas respectivas acelerações (Santos, 2003). O bloco de planeamento de trajectórias tem como entrada a especificação do movimento em termos do valor inicial/final e o tempo de execução, devolvendo a evolução temporal de cada variável.

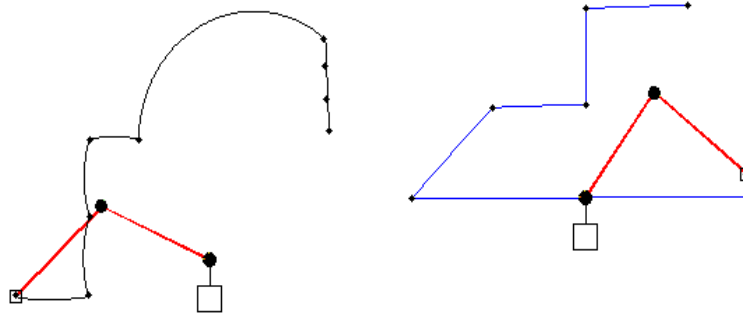


Figura 3.2: Exemplos do movimento realizado por um robô 2R: movimento ponto-a-ponto (esquerda) e movimento contínuo (direita) passando por um conjunto de pontos intermédios.

3.3 Modelação da perna de suporte

Nesta subsecção descrevem-se os modelos cinemáticos derivados no âmbito desta dissertação tendo em vista a simulação de uma perna similar à do robô humanóide. Neste estudo, a estrutura da perna consiste de uma série de elos mecânicos rígidos – pé-tornozelo, tornozelo Joelho e Joelho-anca – interligados pelas juntas rotacionais do tornozelo cujos eixos se intersectam num ponto e pela junta do Joelho. A Figura 3.3 ilustra o modelo da perna de suporte com a atribuição do sistema de coordenadas de referência (\mathcal{S}_R), o sistemas de coordenadas de base (\mathcal{S}_0) e dos restantes sistemas de eixos até ao elemento terminal que coincide com a anca (sistemas de coordenadas directos). Este modelo simples com 3-gdl é relevante em estudos (alguns fora do âmbito deste trabalho) relacionados com a relação entre o centro de gravidade (COG) e o centro de pressão (COP) e, em particular, na validação de algoritmos de controlo baseados nas forças de interacção com o solo (este assunto será tratado no Capítulo 5).

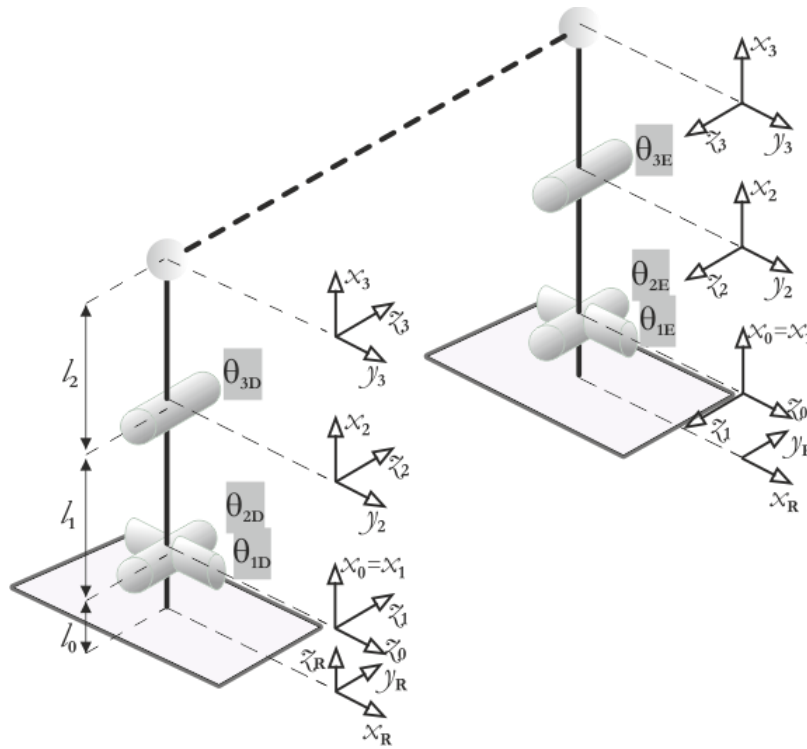


Figura 3.3: Representação dos sistemas de coordenadas atribuídos à perna de suporte (perna esquerda - *E* e perna direita - *D*) em que o sistema de coordenadas de referência S_R está situado no solo e o elemento terminal corresponde à anca (origem do sistema de coordenadas S_3).

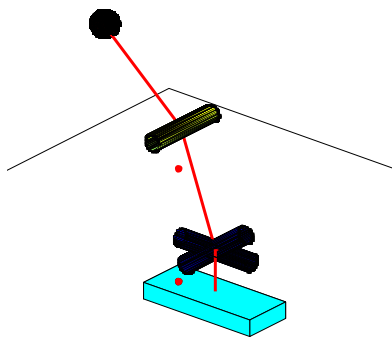


Figura 3.4: Simulador em Matlab usado para validar as equações de cinemática da perna de suporte e os algoritmos de planeamento do movimento.

Elo- <i>i</i>	θ_i	d_i	l_i	α_i
1	θ_1	0	0	$\mp\pi/2$
2	θ_2	0	l_1	0
3	θ_3	0	l_2	0

Tabela 3.1: Parâmetros de Denavit-Hartenberg.

Elo- <i>i</i>	l_i (cm)	m_i (kg)
l_0	5.0	0.440
l_1	13.9	0.343
l_2	14.3	0.409

Tabela 3.2: Comprimentos e massas dos elos.

Apesar da simplicidade do modelo, uma análise geométrica da estrutura não permite facilmente derivar as equações de cinemática directa. Assim, optou-se pelo procedimento sistemático baseado na convenção de Denavit-Hartenberg. A Tabela 3.1 apresenta os parâmetros de elos e juntas que permitem partir do sistema S_0 e chegar à extremidade S_3 . Os símbolos $\{\pm, \mp\}$ são usados, sempre que necessário, de acordo com a seguinte regra: o sinal superior refere-se à perna esquerda e o inferior à direita. Os valores dos comprimentos e massas dos elos mecânicos estão descritos na Tabela 3.2. O desenvolvimento de um simulador cinemático permitiu o teste e a validação das equações obtidas, bem como proceder à animação do movimento (Figura 3.4).

O cálculo das equações de cinemática directa assenta num procedimento recursivo obtido a partir de produtos das contribuições de cada grau de liberdade. Neste caso concreto, é preciso incluir neste produto a matriz de transformação que relaciona o sistema de coordenadas de referência S_R com o sistema S_0 :

$$T_{S_R \rightarrow S_0} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

obtendo-se as seguintes expressões para a posição do elemento terminal (et):

$$\begin{aligned} x_{et} &= \mp l_1 s_2 \mp l_2 s_{23} \\ y_{et} &= -l_1 s_1 c_2 - l_2 s_1 c_{23} \\ z_{et} &= l_0 + l_1 c_1 c_2 + l_2 c_1 c_{23} \end{aligned} \quad (3.8)$$

em que a notação simplificada $s_i | s_{ij}$ e $c_i | c_{ij}$ representa o $\sin(\theta_1) | \sin(\theta_1 + \theta_2)$ e $\cos(\theta_1) | \cos(\theta_1 + \theta_2)$, respectivamente. Por sua vez, o problema da cinemática inversa consistiu na determinação do vector das juntas correspondente a uma dada posição do elemento terminal. Neste caso, foram obtidas equações na forma fechada por manipulação das equações de cinemática directa de acordo com a Tabela 3.3.

θ_i	k_1	k_2	k_3
θ_1	$z_{et} - l_0$	$\pm y_{et}$	0
θ_2	$\mp 2l_1 x_{et}$	$2l_1 A$	$x_{et}^2 + A^2 + l_1^2 - l_2^2$
θ_3	0	$2l_1 l_2$	$x_{et}^2 + A^2 - l_1^2 - l_2^2$

Tabela 3.3: Equações de cinemática inversa da perna de suporte; a variável A assume os valores $(z_{et} - l_0)/c_1$ ou $-y_{et}/s_1$ de acordo com o quadrante de θ_1 para evitar divisões por zero.

De notar que não se coloca o problema da redundância na junta do joelho uma vez que não é permitida a inversão do respectivo movimento. A derivação da cinemática diferencial, que estabelece a relação entre a velocidade nas juntas e a velocidade linear do elemento terminal, foi realizada com base na matriz do Jacobiano obtida pela diferenciação das equações de cinemática directa em ordem às variáveis das juntas:

$$J_{et} = \begin{bmatrix} 0 & l_1 c_2 + l_2 c_{23} & l_2 c_{23} \\ -l_1 c_1 c_2 - l_2 c_1 c_{23} & l_1 s_1 s_2 + l_2 s_1 s_{23} & l_2 s_1 s_{23} \\ -l_1 s_1 c_2 - l_2 s_1 c_{23} & -l_1 c_1 s_2 - l_2 c_1 s_{23} & -l_2 c_1 s_{23} \end{bmatrix} \quad (3.9)$$

O Jacobiano é uma ferramenta muito importante para caracterizar o sistema articulado e será usado posteriormente para obter uma solução para a cinemática inversa. Finalmente, calcularam-se as coordenadas do centro de massa da perna (equação 3.10) e a respectiva matriz do Jacobiano de COG em q (equação 3.11):

$$COG = \frac{1}{M_t} \begin{bmatrix} m_0 CM_{0x} + m_1 r_1 s_2 + m_2 (l_1 s_2 + r_2 s_{23}) \\ -m_1 (r_1 s_1 c_2 + CM_{1y} c_1) - m_2 (l_1 s_1 c_2 + r_2 s_1 c_{23}) \\ m_0 CM_{0z} + m_1 (l_0 + r_1 c_1 c_2 + CM_{1y} s_1) + m_2 (l_0 + l_1 c_1 c_2 + r_2 c_1 c_{23}) \end{bmatrix} \quad (3.10)$$

$$J_{COG} = \frac{1}{M_t} \begin{bmatrix} 0 & m_1 r_1 c_2 + m_2 (l_1 c_2 + r_2 c_{23}) & m_2 r_2 c_{23} \\ -m_1 (r_1 c_1 c_2 + CM_{1y} s_1) - m_2 (l_1 c_1 c_2 + r_2 c_1 c_{23}) & m_1 r_1 s_1 s_2 + m_2 (l_1 s_1 s_2 + r_2 s_1 s_{23}) & m_2 r_2 s_1 s_{23} \\ -m_1 (r_1 s_1 c_2 - CM_{1y} c_1) - m_2 (l_1 s_1 c_2 + r_2 s_1 c_{23}) & -m_1 r_1 c_1 s_2 - m_2 (l_1 c_1 s_2 + r_2 c_1 s_{23}) & -m_2 r_2 c_1 s_{23} \end{bmatrix} \quad (3.11)$$

3.4 Modelação da perna livre

Seguindo o esquema da subsecção anterior, procedeu-se à modelação da perna livre constituída pelos elos anca Joelho, joelho-tornozelo e tornozelo-pé e por seis juntas rotacionais. A atribuição dos sistemas de coordenadas está ilustrada na Figura 3.5 de acordo com a natureza de cada junta, juntamente com a identificação dos parâmetros geométricos que especificam o comprimento dos elos (descritos na Tabela 3.2). A origem do sistema de coordenadas de referência está localizada na anca e o elemento terminal corresponde à extremidade do pé. A Tabela 3.4 apresenta os respectivos parâmetros de Denavit-Hartenberg. De referir, que o modelo da perna livre é importante em termos das melhores estratégias de planeamento de movimento e posterior avaliação do seu comportamento nas fases de impacto (elevação) com o (do) solo.

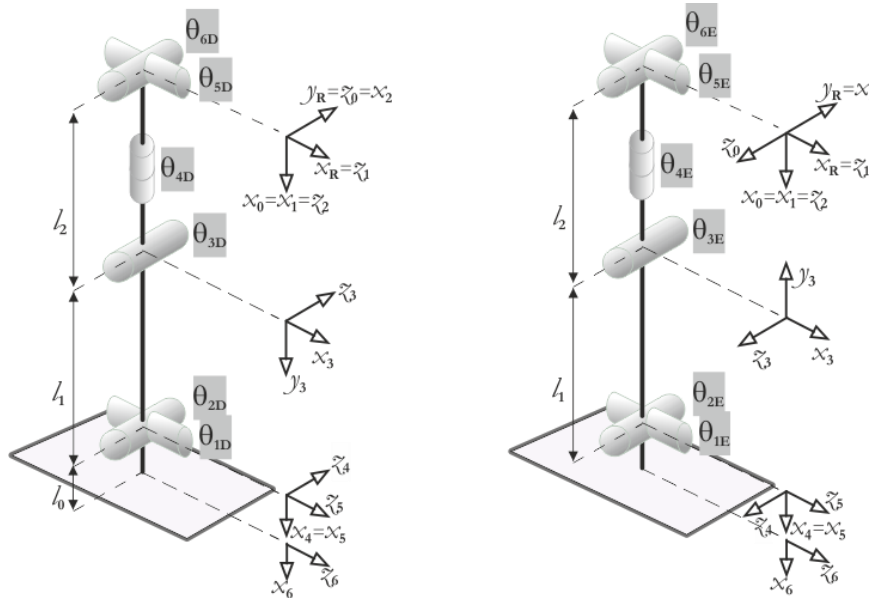


Figura 3.5: Representação dos sistemas de coordenadas atribuídos à perna livre (perna esquerda - E e perna direita, D) em que o sistema de coordenadas de referência S_R está situado na anca e o elemento terminal corresponde à extremidade do pé (origem do sistemas de coordenadas S_0).

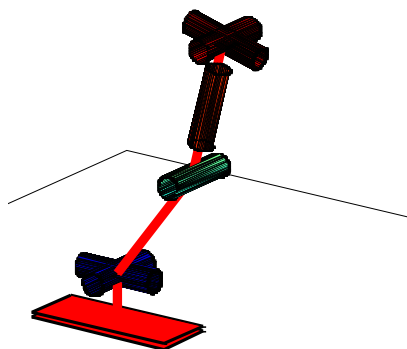


Figura 3.6: Simulador em Matlab do movimento da perna livre (6-gdl).

Elo- i	θ_i	d_i	l_i	α_i
1	θ_1	0	0	$\mp\pi/2$
2	$\theta_2 + \pi/2$	0	0	$\pi/2$
3	$\theta_3 + \pi/2$	l_2	0	$\mp\pi/2$
4	$\theta_4 - \pi/2$	0	l_1	0
5	θ_5	0	0	$\mp\pi/2$
6	θ_6	0	l_0	0

Tabela 3.4: Parâmetros de Denavit-Hartenberg para o cálculo da transformação homogênea $T_{S_0 \rightarrow S_6}$.

Por um lado, as equações de cinemática directa são obtidas a partir das seguintes expressões:

$$\begin{aligned}
 x_{et} &= \pm(s_4 s_5 s_6 \pm c_4 c_6) \cdot (l_0 c_1 s_{23} + l_1 s_3) + c_5 s_6 (l_0 c_1 c_{23} + l_1 c_3 + l_2) - l_0 s_1 (c_4 c_5 s_6 \mp s_4 c_6) \\
 y_{et} &= -s_4 c_5 (l_0 c_1 s_{23} + l_1 s_3) \pm s_5 (l_0 c_1 c_{23} + l_1 c_3 + l_2) \pm l_0 s_1 c_4 c_5 \\
 z_{et} &= \pm(s_4 s_5 c_6 \mp c_4 s_6) \cdot (l_0 c_1 s_{23} + l_1 s_3) + c_5 c_6 (l_0 c_1 c_{23} + l_1 c_3 + l_2) - l_0 s_1 (c_4 s_5 c_6 \pm s_4 s_6)
 \end{aligned}
 \tag{3.12}$$

Por outro lado, as equações de cinemática inversa foram obtidas seguindo a mesma aproximação algébrica da subsecção anterior. Para o efeito, assumem-se duas restrições que permitem chegar a uma solução na forma fechada. Primeiro, considera-se que a terceira junta da anca permanece imóvel ($\theta_4 = 0$). Em segundo lugar, aplica-se uma restrição na orientação do pé que mantém uma postura horizontal ao longo do movimento. A aplicação destas restrições permite simplificar as expressões (3.12) e obter a solução representada na Tabela 3.5. Os modelos matemáticos da perna de suporte e da perna livre foram incorporados e adaptados a um simulador já existente ligado ao estudo de padrões de locomoção (Ferreira, 2009). A Figura 3.7 ilustra o modelo desenvolvido em Matlab e a animação do movimento ao longo de vários passos simétricos.

θ_i	k_1	k_2	k_3
θ_3	0	$2l_1 l_2$	$x_{et}^2 + y_{et}^2 + z_{et}^2 - l_1^2 - l_2^2$
θ_1	x_{et}	$\pm y_{et}$	$-l_2 s_3$
θ_2	0	$(l_1 + l_2 c_4)$	$\pm x_{et} c_1 + y_{et} s_1$
θ_6	$\theta_6 = \arcsin(-c_1 s_2)$		
θ_5	$\theta_5 = -\theta_3 - \arcsin\left(\frac{s_1}{c_6}\right)$		

Tabela 3.5: Equações de cinemática inversa da perna livre para $\theta_4 = 0$.

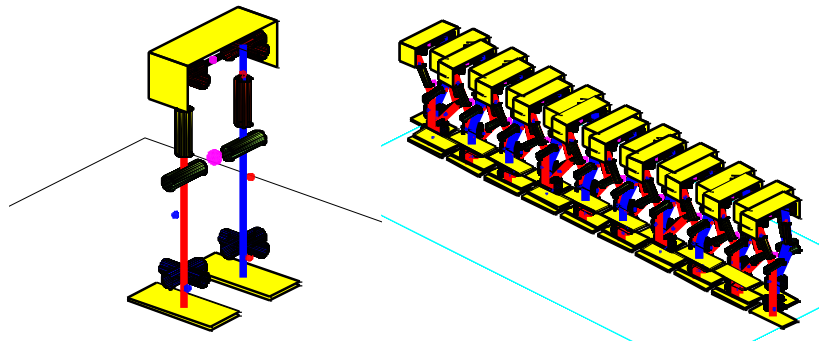


Figura 3.7: Modelo completo dos membros inferiores (esquerda) e animação de um padrão de locomoção (direita) adaptado de Ferreira (2008).

3.5 Modelação do tronco e braços

Nesta subsecção derivam-se as equações de cinemática para a subestrutura formada pelo tronco (postura vertical fixa) e pelos dois braços. Os braços do robô são constituídos pelos elos ombro-cotovelo e cotovelo-punho (3-gdl no ombro e uma junta rotacional no cotovelo). A Figura 3.8 ilustra a posição zero dos braços e a atribuição dos sistemas de coordenadas. O sistema de coordenadas de referência S_R está localizado no ponto de intersecção dos 3 eixos do tronco, enquanto o elemento terminal corresponde à extremidade dos braços.

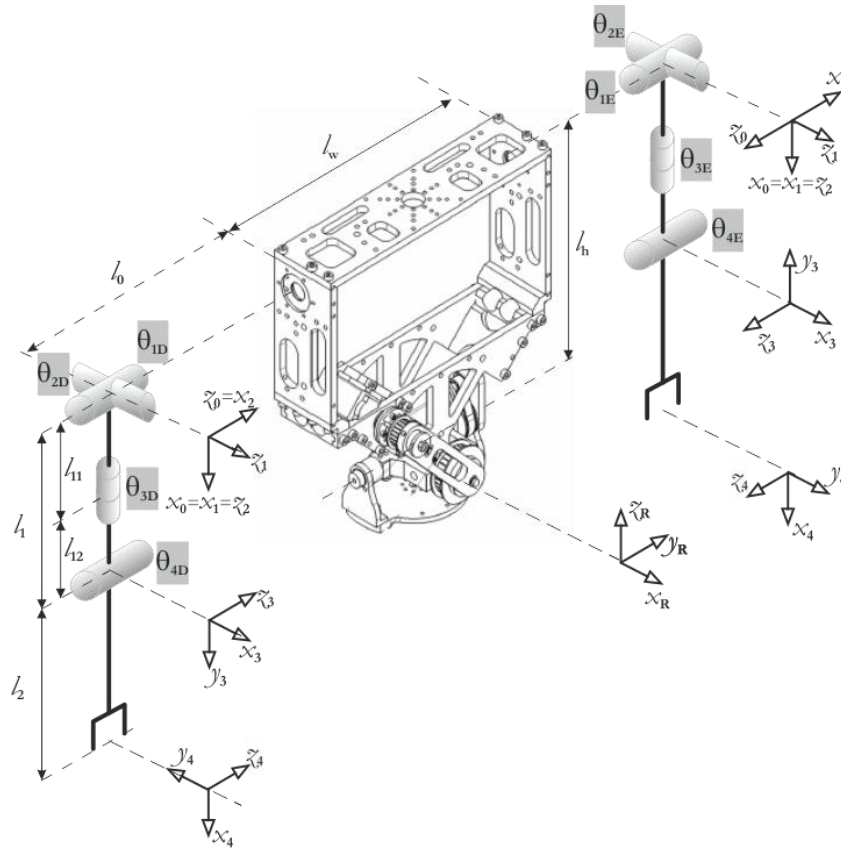


Figura 3.8: Representação dos sistemas de coordenadas atribuídos aos braços ao longo da estrutura de elos mecânicos suportados pelo tronco. O sistema de coordenadas de referência S_R está situado na parte inferior do tronco e o elemento terminal corresponde à extremidade de cada braço, O_4 .

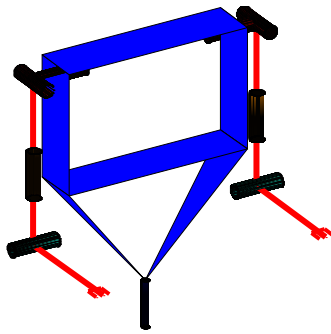


Figura 3.9: Simulador em Matlab do movimento da estrutura tronco e braços com a implementação das equações de cinemática.

Elo- i	θ_i	d_i	l_i	α_i
1	θ_1	0	0	$\mp\pi/2$
2	$\theta_2 + \pi/2$	0	0	$\pi/2$
3	θ_3	l_1	0	$\mp\pi/2$
4	$\theta_4 - \pi/2$	0	l_2	0

Tabela 3.6: Parâmetros de Denavit-Hartenberg para o cálculo da transformação homogênea $T_{S_0 \rightarrow S_4}$.

Elo- i	l_i (cm)
l_h	15
$l_w + l_0$	28
l_1	12.6
l_2	10.8

$l_{11} = 7.7$
 $l_{12} = 4.9$

Tabela 3.7: Comprimento dos elos mecânicos.

De notar que, neste estudo, a junta responsável pelo movimento de rotação medial/lateral (variável θ_3) assume particular relevância pois permite que os braços assumam uma postura em frente do tronco. Isto

significa que existe um grau de liberdade redundante para tarefas de posicionamento da mão no espaço tridimensional. Uma vez mais, a colocação do servomotor no cotovelo impede a inversão da sua configuração, pelo que os braços adoptam a postura designada de “cotovelo para baixo”. A Tabela 3.6 descreve os parâmetros de Denavit-Hartenberg relativos aos dois braços. Por sua vez, a Tabela 3.7 apresenta o valor dos comprimentos de cada elo mecânico.

A obtenção das equações de cinemática directa envolve a matriz de transformação homogênea:

$$T_{S_R \rightarrow S_0} = \begin{bmatrix} 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \mp 1 & \pm (l_w/2 + l_0) \\ -1 & 0 & 0 & l_b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

resultando nas seguintes expressões:

$$\begin{aligned} x_{et} &= l_2 s_4 (s_1 s_2 s_3 \pm c_1 c_3) \pm l_2 s_1 c_2 c_4 \pm l_1 s_1 c_2 \\ y_{et} &= \mp l_2 c_2 s_3 s_4 + l_2 s_2 c_4 + l_1 s_2 \pm l_w \pm l_0 \\ z_{et} &= \mp l_2 s_4 (c_1 s_2 s_3 \mp s_1 c_3) - l_2 c_1 c_2 c_4 - l_1 c_1 c_2 \end{aligned} \quad (3.14)$$

O problema da cinemática inversa foi dividido em duas partes de acordo com a posição desejada para o elemento terminal (Figura 3.10). Por um lado, assume-se que o ângulo da junta θ_3 é zero quando se realizam movimentos de abdução/adução ou rotação lateral/medial do ombro em que se verificam as condições $y_{et} \geq \frac{l_w}{2} + l_0$ e $y_{et} \leq -\frac{l_w}{2} - l_0$ no braço esquerdo e direito, respectivamente. Nestas circunstâncias, as equações de cinemática inversa podem ser obtidas a partir dos parâmetros definidos na Tabela 3.8.

θ_i	k_1	k_2	k_3
θ_4	0	$2l_1 l_2$	$x_{et}^2 + (y_{et} \mp l_w \mp l_0) + z_{et}^2 - l_1^2 - l_2^2$
θ_1	z_{et}	$\pm x_{et}$	$l_2 s_4$
θ_2	0	$(l_1 + l_2 c_4)$	$\pm x_{et} s_1 - z_{et} c_1$

Tabela 3.8: Equações de cinemática inversa obtidas a partir da expressão geral quando $\theta_3 = 0$.

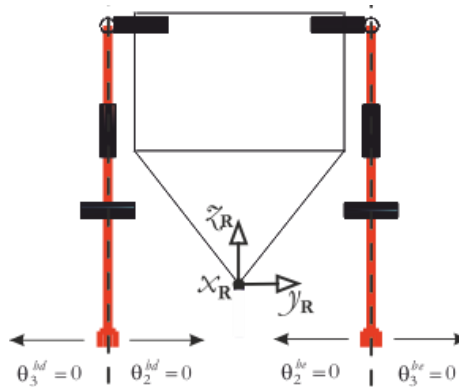


Figura 3.10: Restrições aplicadas às juntas θ_2 e θ_3 de acordo com a coordenada y_{et} .

A solução implementada no caso de movimentos que tragam o elemento terminal na direcção do tronco $\left(y_{et}^{be} < \frac{l_w}{2} + l_0 \text{ ou } y_{et}^{bd} > -\frac{l_w}{2} - l_0 \right)$ é baseada na procura do ângulo mínimo θ_3 que permita que a junta θ_2 se encontre dentro dos seus limites físicos (o algoritmo está descrito na Tabela 3.9).

1	Inicializar as variáveis $\theta_3^{\min} = 0$ e $\theta_3^{\max} = \frac{\pi}{2}$			
2	Definir o valor do ângulo θ_4 de acordo com os seguintes parâmetros:			
	θ_4	0	$2l_1l_2$	$x_{et}^2 + (y_{et} \mp l_w \mp l_0) + \tilde{x}_{et}^2 - l_1^2 - l_2^2$
3	WHILE $\theta_3^{\max} - \theta_3^{\min} > \text{resolução}$,			
	3.1	Definir o valor do ângulo $\theta_3 = \text{valormédio}(\theta_3^{\min}, \theta_3^{\max})$		
	3.2	Calcular os correspondentes ângulos θ_1 e θ_2 de acordo com os seguintes parâmetros:		
		θ_1	$\tilde{x}_{et} - l_b$	$\pm x_{et}$
	θ_2	$l_2 s_3 s_4$	$\pm (l_1 + l_2 c_4)$	$x_{et} s_1 \mp (\tilde{x}_{et} - l_b) c_1$
	3.3	Verificar se os limites físicos da junta θ_2 foram ultrapassados		
3.4	IF VERDADEIRO THEN: $\theta_3^{\min} = \theta_3$			
	ELSE: $\theta_3^{\max} = \frac{\pi}{2}$			
	NEXT LOOP			

Tabela 3.9: Algoritmo de procura de uma solução para a cinemática inversa.

Capítulo 4

Controlo do Movimento em Posição e Velocidade

4.1 Introdução

O objectivo deste capítulo é descrever os módulos de software implementados na unidade central de processamento, ou para ser mais preciso, unidade parcial de teste, e na unidade slave para permitir o controlo do movimento do robô. Tal como descrito no capítulo anterior, o movimento do robô será efectuado dividindo a estrutura completa em três partes distintas: braços que podem operar em conjunto (8-gdl), perna de suporte (3-gdl) e perna livre (6-gdl). Pretende-se analisar o comportamento individual destes sub-sistemas e avaliar o seu desempenho face a problemas próprios associados a cada um. Assim, em primeiro lugar, descreve-se a estrutura global do software desenvolvido quer na unidade central de processamento como na unidade slave. De seguida, procede-se à análise pormenorizada das funções relacionadas com o planeamento e controlo do movimento, incluindo o comando simultâneo de n servos de forma síncrona e a implementação dos movimentos ponto-a-ponto e movimentos contínuos. Nesta medida, procura-se apresentar quer os problemas encontrados como as soluções implementadas para assegurar os melhores resultados possíveis. No final do capítulo são então apresentados e discutidos os resultados finais obtidos. Para tal ser possível, foram realizadas diversas experiências com os diferentes membros do robô humanóide na tentativa de abranger e analisar diferentes situações.

O diagrama da Figura 4.1 ilustra a estrutura geral do sistema de controlo de baixo-nível (Ribeiro, 2010) que, preferencialmente, receberá como entrada variáveis de referência especificadas no espaço Cartesiano. Desta forma, as equações de cinemática inversa derivadas no Capítulo 3 foram implementadas na unidade slave para gerar as respectivas trajectórias no espaço das juntas, juntamente com os algoritmos de planeamento de trajectórias. Os servomotores utilizados estão vocacionados para o controlo de posição em que é fornecida uma posição angular final desejada e uma velocidade média relacionada com o tempo de execução do movimento. O módulo de planeamento de trajectória inserido no próprio servomotor gera perfis trapezoidais, sendo responsável pela geração de uma sequência de valores intermédios como referência para o controlador interno. Uma das hipóteses colocadas inicialmente e avaliada ao longo deste capítulo foi a possibilidade de implementar o controlo de velocidade nas diferentes juntas para realizar movimentos contínuos.

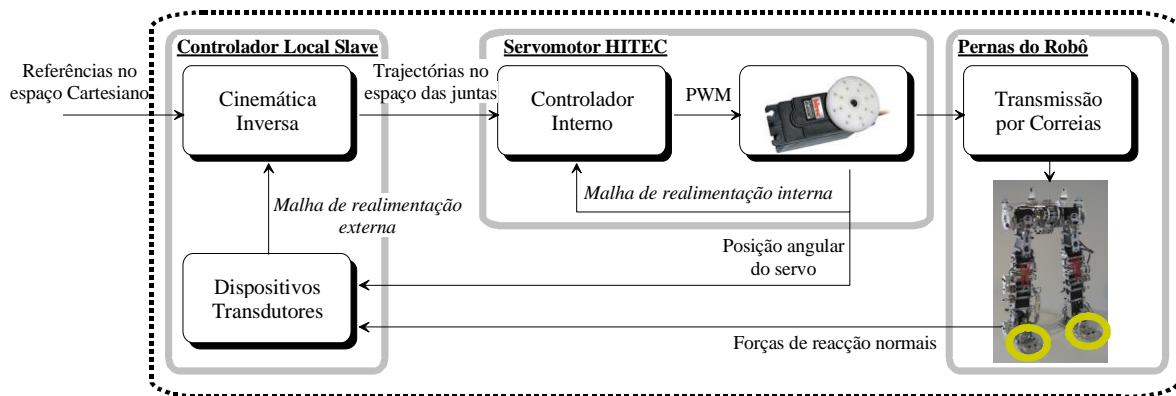


Figura 4.1: Sistema de controlo de baixo-nível constituído pela unidade slave e o controlador interno do servomotor.

4.2 Arquitectura do sistema

4.2.1 Comunicação entre a unidade central e a unidade slave

Os objectivos da comunicação entre a unidade central de processamento (UC) e a unidade slave (US) (Ribeiro, 2010) são, por um lado, o envio de comandos e/ou directivas de alto nível para o robô e, por outro lado, a recolha de dados que permitam avaliar o comportamento efectivo do sistema robótico. Entre a informação recolhida pela unidade central de processamento consta a evolução temporal das variáveis mais relevantes como sejam os ângulos das juntas ou as forças de reacção no solo. Esta comunicação é efectuada através do protocolo RS-232 usando os parâmetros identificados na Tabela 4.1 e uma trama de comunicação composta por 8 bytes (ver Tabela 4.2).

No caso da comunicação da unidade central para a unidade slave, o primeiro byte especifica o número de tramas que pretendemos enviar de cada vez. O segundo byte, o comando, representa a função que pretendemos invocar dentro da slave. Os restantes bytes são preenchidos com os parâmetros necessários para que a função invocada possa ser executada. Assim, na unidade central é criada uma matriz onde são colocados todos os dados que desejamos enviar para a unidade slave.

Parâmetros	Valor
Baudrate	57600
Databits	8
Stopbits	2
Parity	None
Handshake	None

Tabela 4.1: Parâmetros da comunicação RS-232.

byte	1	2	3	4	5	6	7	8
UC	num_ tramas	comando	param1 e param2		param3 e param4		param5 e param6	
US	num_ tramas	comando	return1		return2		return3	

Tabela 4.2: Tramas usadas na comunicação entre a unidade central (UC) e a unidade slave (US).

Como, por norma, apenas são colocados na trama três valores diferentes de cada vez (cada valor vai ocupar dois bytes da trama a enviar), esta matriz tem três colunas e um número de linhas que depende da quantidade de informação a enviar. Por exemplo, quando é utilizada a função de movimento contínuo para os dois braços é preciso enviar para a unidade slave as coordenadas cartesianas X, Y, Z do elemento terminal e o tempo de execução (T_x) para o movimento de cada braço e ainda a ordem polinomial (necessária para a função de Planeamento de Trajectória) para ambos os braços. Isto perfaz um total de nove valores que dão origem a uma matriz 3×3 no Matlab, que serve para organizar os dados e facilitar o envio dos mesmos para a unidade slave porque cada linha de dados é enviada numa trama (neste caso são necessárias 3 tramas).

Em relação à comunicação da unidade slave para a unidade central, existiam algumas limitações com a solução já implementada (Ribeiro, 2010) e que era baseada em dois passos: armazenamento da informação sensorial de interesse durante a execução de um dado movimento e posterior envio para a unidade central. Para o efeito, era criada na unidade slave uma matriz de dimensão fixa (posArray 3×200 , para três servos) onde era guardada toda a informação a enviar. No final do movimento os valores dessa matriz eram enviados para a unidade central que conhecia *a priori* a quantidade de tramas a receber. Apesar desta solução evitar maiores atrasos no ciclo de controlo, isso implicava algumas limitações na quantidade de dados recolhidos em resultado das limitações de memória do microprocessador dsPIC. Mais ainda, tornava mais difícil estabelecer a taxa de amostragem dos dados em função da duração estabelecida para os movimentos. Por exemplo, quando se tratava de um movimento com um pequeno tempo de execução a matriz era totalmente preenchida, se bem que o movimento terminava antes de preencher os 200 valores (os espaços que restavam continuavam a ser preenchidos sempre com a mesma posição, a posição final do movimento). Pelo contrário, num movimento com um tempo de execução maior a matriz iria ser completamente preenchida, mas correndo o risco de se perder informação porque os 200 valores iriam ser preenchidos antes do movimento terminar.

Desta forma, procederam-se a algumas alterações para permitir o envio dos dados para a unidade central de cada vez que é efectuada uma leitura de dados. Assim, em vez de se utilizar a memória do microcontrolador para armazenar todos os valores, estes vão ser logo enviados um a um, ou para ser mais preciso, de três em três valores porque uma trama pode enviar de uma só vez três valores (para aumentar a resolução são utilizados dois bytes para enviar um valor). Como à partida, não é conhecido o número de amostras que vão ser recolhidas, torna-se necessário enviar para a unidade central um código, “123” (basicamente é uma trama com o return1=1, return2=2 e return3=3), que sinaliza o fim da execução de um dado movimento. Juntamente com as leituras sensoriais são enviadas para a unidade central os instantes em que estes valores foram obtidos. Nesta medida, é de esperar um maior tempo de comunicação com o PC que se traduz num aumento do tempo de execução do ciclo de controlo. Esta última implementação foi a utilizada neste trabalho, mas fica em aberto a possibilidade de no futuro se utilizarem as duas formas de enviar informação num conceito misto, onde em casos mais extremos em que a velocidade do ciclo é importante seria utilizada a primeira forma e nos restantes casos a segunda solução.

4.3 Módulos básicos de software

O Matlab foi a plataforma utilizada para o desenvolvimento de uma interface básica de comunicação com o sistema robótico capaz de permitir a um utilizador comandar o movimento dos dois braços (8-gdl), da perna

de suporte (3-gdl) ou da perna livre (6-gdl). Nesta fase do projecto, a unidade central de processamento é baseada num PC que envia ordens ou comandos a serem executados na unidade slave com base no protocolo descrito anteriormente. Ao mesmo tempo, a unidade central permite ao utilizador recolher toda a informação sensorial, de forma a processar estes dados e efectuar a visualização gráfica de um conjunto de variáveis, que descrevem aquilo que se passa nos servomotores e sensores durante a execução de uma determinada acção. Nessa medida, a maioria dos algoritmos desenvolvidos e o seu processamento ocorre na unidade slave baseada num microcontrolador dsPIC.

Apesar de no início do trabalho existir uma interface funcional (Ribeiro, 2010), esta foi praticamente toda reconstruída de raiz, bem como a maioria das funções. Assim, o utilizador deve começar por executar uma rotina de inicialização, antes de escolher um dos modos de controlo do robô ou uma leitura sensorial. Estão disponíveis três modos de controlo: (1) movimento individual de n eixos em que o utilizador especifica a posição angular final da(s) junta(s), (2) movimento ponto-a-ponto e (3) movimento em linha recta em que o utilizador especifica as coordenadas Cartesianas finais do elemento terminal. Quanto às leituras sensoriais que podem ser efectuadas até ao momento resumem-se às posições angulares das juntas e às forças normais de reacção no solo.

A Tabela 4.3 apresenta a lista de funções desenvolvidas, implementadas e testadas com a unidade slave, os blocos internos à própria função, os parâmetros de entrada/saída e uma breve descrição do seu funcionamento. Estas funções são do Matlab mas correspondem directamente a uma função no PIC (as funções entre Matlab e microcontrolador estão interligadas com um número específico para cada, o command). As Tabela 4.4, Tabela 4.5 e Tabela 4.6 resumem um conjunto de funções especificamente ligadas ao movimento dos braços, da perna livre e da perna de suporte, respectivamente.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Initialization	Communication Matlab-PIC Communication PIC-Servos	H	void	Verifica se as comunicações entre Matlab-PIC e PIC-Servos estão a funcionar correctamente.
Set Servo ID Speed Read Position	Send Id and Speed Read Position	id, speed	pos	Indica ao servomotor com determinado id a velocidade pretendida (speed) e retorna a posição actual do mesmo.
Set Servo ID Target Position	Send Id and Position	id, pos	return	Indica a posição final (pos) do servomotor identificado por id; return é o estado da operação.
Read all Servos Positions and Set Speeds	Use of setSpeedReadPos function Display positions of each id	id, speed	pos	É feita a leitura de todas as posições e coloca se todos os servos com a mesma velocidade, 25%.
Release	-	Void	return	Liberta todos os servos.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Set Position to N ID's	Data processing Sending data Collecting data (time and position for the N Id's) Time course of joint angles	id's, Q's, time	t, posArray (angles)	Indica a N servos, escolhidos pelo usuário através dos id's, a posição final para cada um destes com o objectivo de terminarem todos ao mesmo tempo (time); o que retorna é o conjunto de dados necessário para visualização do gráfico das suas posições angulares em função do tempo.
Set Maximum Speed	-	maxSpeed	void	Estabelece a velocidade máxima a que um movimento deve ser executado.
Num Samples from last Move	-	Void	N	Leitura do número de amostras recolhidas no último movimento.
Direct Kinetic	Send angles Use of dirKin.m Display positions xyz	q1, q2, q3	void	Converte dentro da PIC através da cinemática directa de ângulos para posições cartesianas.
Inverse Kinematics	Send positions xyz Use of dirKin.m	posX, posY, posZ	return, qf1, qf2, qf3, qf4	Converte coordenadas Cartesianas em ângulos das juntas usando a cinemática inversa; no caso de não existir uma solução devolve mensagem indicativa.
Initial Position	Reading vector ini_pos Use of setPosId	ini_pos, id's	void	Todos os servos vão para uma posição inicial já definida para cada um.
Different Position	Reading vector out_pos Use of setPosId	out_pos, id's	void	Para observar que todos os servos estão operacionais.
Point to Point Move	Data processing Sending data Analysis of errors	XYZ for left and right arms, time	command, return1, return2	Faz um movimento ponto a ponto num determinado tempo (time) para as coordenadas cartesianas (XYZ) introduzidas para um ou dois braços; o command e os returns dão o alerta para eventuais erros ou limites.
Continuous Move	Data processing Sending data Analysis of errors	XYZ for left and right arms, time, ord	command, return1, return2	Faz um movimento contínuo num determinado tempo (time) para as coordenadas cartesianas (XYZ) introduzidas para um ou dois braços; o command e os returns dão o alerta para eventuais erros ou limites.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Continuous Move (with time and 3D XYZ)	<p>Data processing</p> <p>Sending data</p> <p>Analysis of erros</p> <p>Collecting data (time and position(angle))</p> <p>Convert angles to cartesian coordinates with DirKin_4dof.m function</p> <p>Presentations of graphics (Time vs Angles and Time vs Cartesian Coordinates)</p>	XYZ for left and right arms, time, ord	command, return1, return2, t, posArray (angles)	Faz um movimento contínuo num determinado tempo (time) para as coordenadas cartesianas (XYZ) introduzidas para um ou dois braços; retorna um conjunto de dados necessários para visualização dos gráficos das suas posições angulares e coordenadas cartesianas em cada instante de tempo (t).
Draw 3D Sequence of Continuous Moves	<p>Reading seq1 or seq2</p> <p>Use of Seq_ContMov.m for each move</p> <p>Save data for each move</p> <p>Convert angles to Cartesian coordinates with DirKin_4dof.m</p> <p>Presentations of time courses of angles & Cartesian coordinates</p>	seq1 or seq2	t, posArray (angles)	É necessário escolher uma das duas sequências de movimentos já pré-definidas, seq1 e seq2, dentro destas já tem as coordenadas, tempo e ordem; vão ser realizados 4 movimentos contínuos e no final de cada um dos movimentos os dados são guardados; os dados são tratados e é possível visualizar dois gráficos de toda a sequência (posições angulares e coordenadas cartesianas em cada instante de tempo)

Tabela 4.3: Funções associadas ao controlo do movimento das três sub-estruturas: dois braços, perna de suporte e perna livre.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Jacobian Move	<p>Data processing</p> <p>Sending data</p> <p>Analysis of errors</p> <p>Collecting data (time and position(angle))</p> <p>Time courses of joint angles</p>	XYZ for left and right arms, time	command, return1, return2, t, posArray (angles)	Faz um movimento contínuo com uma função Jacobiana num determinado tempo (time) para as coordenadas cartesianas (XYZ) introduzidas para um ou dois braços; retorna um conjunto de dados necessários para visualização do gráfico das suas posições angulares em cada instante de tempo (t).

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Draw Qvel given to the Servos	Collecting data (time and velocities) Time courses of joint velocities	Void	t, Qvel's (velocities)	É realizado um movimento contínuo para ser possível a recolha de dados sobre as mudanças das velocidades (Qvel) em cada instante de tempo (t).

Tabela 4.4: Funções associadas especificamente ao controlo do movimento dos dois braços.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Read Sensors	-	Void	valSensor	Valor lido em cada um dos sensores de força.
Calibration Sensors	-	Void	vaslSensor	Calibração dos quatros sensores de força, não sendo necessário voltar a chama-la depois de ter sido utilizada pelo menos uma vez.
Impact	Collecting data before impact Collecting data after impact	Void	t, XYZ_z, valSensor	É simulado um movimento com a perna onde o pé vai pousar no solo; leitura constante da posição dos motores e dos sensores de força antes do impacto; depois do impacto apenas leitura dos sensores de força uma vez que os motores tão parados; é no final apresentado um gráfico onde temos a posição do pé no eixo dos zz e valores dos sensores em cada instante de tempo.

Tabela 4.5: Funções associadas especificamente ao controlo do movimento da perna livre.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Constant Qvel vs Changing Qvel	First movement (const. Qvel) collecting data Second movement (chang. Qvel) collecting data Time course of joint velocities	void	t1, Qvel1, t2, Qvel2	Função criada apenas para comparação da velocidade calculada pelo servo (Qvel constante) com a velocidade calculada por nós para cada instante (Qvel variável)
Read Sensors	-	void	valSensor	Valor lido em cada um dos sensores de força.
Calibration Sensors	-	void	vaslSensor	Calibração dos 4 sensores de força; será invocada pelo menos uma vez.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
Continues Reading Sensors	Creation of graphics Constant read & update of values	void	valSensor, COPx, COPy	Leitura constante dos sensores de força; representação com dois gráficos em tempo real, um do Centro de Pressão (COPx em x e COPy em y) e outro com os valores dos sensores (valSensor).
Media of the Reading Sensors for Calibration	-	void	valSensor	Leitura dos sensores de força até obter 50 amostras, é de seguida feita a média para obter a Média do Total dos Sensores (sem calibração).
Simple Equilibrium Test	Sending data (Tx) Collecting data (COP reference, time, position(angle), value of the sensors, velocities(°/s and %), by this order) Calculation of COP for each time and th error of the COP Presentations of six graphics	Tx	t, posArray, COP_ref, valSensor, Qvel, Qvelp	Teste de equilíbrio da perna de suporte como o algoritmo de controlo baseado nas forças de reacção; as velocidades nas juntas calculadas são directamente proporcionais ao erro do COP. Leitura das posições angulares e dos sensores de força para cada instante de tempo com a representação de seis gráficos.
Equilibrium Test with Trans_J	Sending data (Tx) Collecting data (COP reference, time, position(angle), value of the sensors, velocities(°/s and %), by this order) Calculation of COP for each time and th error of the COP Presentations of ten graphics (some of them for comparisons)	Tx	t, posArray, COP_ref, valSensor, Qvel, Qvelp	Teste de equilíbrio da perna como o algoritmo de controlo baseado nas forças de reacção; as velocidades nas juntas são obtidas através da Transformada da matriz do Jacobiano. Leitura das posições angulares e dos sensores de força para cada instante de tempo com a representação de seis gráficos.
Equilibrium Test with Inv_J	Sending data (Tx) Collecting data (COP reference, time, position(angle), value of the sensors, velocities(°/s and %), by this order) Calculation of COP for each time and	Tx	t, posArray, COP_ref, valSensor, Qvel, Qvelp	Teste de equilíbrio da perna como o algoritmo de controlo baseado nas forças de reacção; as velocidades nas juntas são obtidas através da Inversa da matriz do Jacobiano. Leitura das posições angulares e dos sensores de força para cada instante de tempo com a representação de seis gráficos.

Função	Blocos dentro da Função	Parâmetros de Entrada	Parâmetros de Saída	Descrição
	the error of the COP Presentations of ten graphics (some of them for comparisons)			
Move following COP_ref	Sending data (Tx) Collecting data (COP reference, time, position(angle), value of the sensors, velocities(°/s and %), by this order) Calculation of COP for each time and the error of the COP Presentations of six graphics	T _x	t, posArray, COP_ref, valSensor, Qvel, Qvelp	É realizado um movimento da perna de suporte em linha recta através do deslocamento do COP de referência. O criada uma trajectória para o COP_ref, a perna tentar acompanhar o deslocamento deste por forma a ter um diferença entre o COP lido e o COP_ref próxima de zero e assim realiza o movimento desejado. Leitura das posições angulares e dos sensores de força para cada instante de tempo com a representação de seis gráficos.

Tabela 4.6: Funções associadas especificamente ao controlo do movimento da perna de suporte.

4.4 Modos de controlo do movimento

O operador do sistema pode optar entre os seguintes modos de controlo: (1) movimento individual de n eixos em que é especificada a posição angular final da(s) junta(s), (2) movimento ponto-a-ponto em que são especificadas as coordenadas (X, Y, Z) desejadas para o elemento terminal e (3) movimento contínuo que exige a mesma especificação do ponto (2), mas exibe um movimento do elemento terminal em linha recta. As próximas subsecções descrevem os algoritmos desenvolvidos para a unidade slave e que implementam cada um destes modos de controlo do movimento.

4.4.1 Controlo individual das juntas

O diagrama de blocos seguinte (Figura 4.2), resume a sequência de passos necessários para a execução do movimento individual de n eixos. Em primeiro lugar, a informação enviada para a unidade slave é colocada numa matriz A com um número de linhas que depende do número de servos a actuar e com 6 colunas onde são colocados os parâmetros associados a cada trama. O primeiro parâmetro está associado ao tipo de objecto com que estamos a lidar, podendo ser o Id do servomotor que queremos controlar ou um número (100 ou 200). Se o valor for 100 então o tipo de objecto é o tempo de execução e se o número for 200 quer dizer que não tem nenhum objecto não sendo necessário ler os parâmetros seguintes. O segundo e terceiro parâmetros têm o valor relacionado com o objecto anterior. No caso de o tipo de objecto ser um Id então o valor tem a ver com o ângulo para o qual queremos que determinado servomotor se mova.

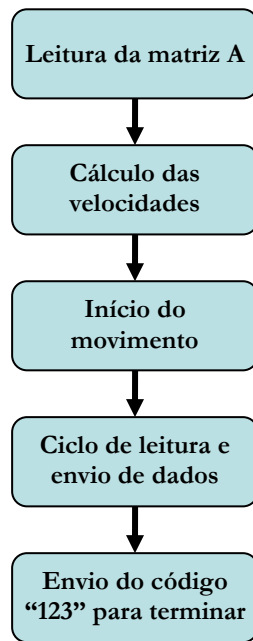


Figura 4.2: Diagrama de blocos da função de controlo individual das juntas.

No caso de ser o número 100 o valor tem a ver com o tempo de execução em segundos. Este tempo de execução é o mesmo para todos os motores o que quer dizer que todos devem terminar o movimento ao mesmo tempo. E no caso de ser o número 200 este apenas simboliza o fim dos dados, deixando os espaços vazios. O quarto parâmetro é novamente um tipo de objecto, o quinto e sexto parâmetros o valor relacionado com o objecto do parâmetro quatro.

Os Id's e respectivas posições angulares finais são guardados em vectores pela mesma ordem de chegada, permitindo usar ciclos que percorrem estes vectores, por exemplo, para dar a ordem de início do movimento a todos os servomotores. O tipo de objecto que simboliza o fim, o número 200, vai quase sempre aparecer no parâmetro 4. Isto acontece porque se o número de servomotores que queremos utilizar é ímpar, somando-lhe o valor do tempo de execução temos um número par de objectos, logo só serão necessárias $(n_objectos)/2$ tramas. Isto faz com que este símbolo 200 não seja utilizado neste caso. Quando queremos trabalhar com um número par de servomotores, juntando o tempo temos um número ímpar de objectos e serão necessárias $((n_objectos-1)/2)+1$ tramas. Nesta última trama vamos ter o tempo de execução (nos parâmetros 1, 2 e 3) e no parâmetro 4 o número 200 para indicar o fim.

Em segundo lugar, é necessário calcular a velocidade média de cada servomotor para assegurar o movimento síncrono dos eixos, isto é, garantir que todas as juntas terminam os seus deslocamentos ao mesmo tempo. Na verdade, o movimento dos servomotores HITEC é imposto fornecendo não só a posição angular desejada como um valor de velocidade de rotação (expresso em percentagem da velocidade máxima do servo) que, de certa forma, impõe um determinado intervalo de tempo para a execução do movimento. O cálculo da velocidade média a aplicar em cada junta requer o cálculo da diferença (Q_{dif}) entre a sua posição angular desejada e a posição angular actual (obtido através da função *setSpeedReadPos*).

No caso de o utilizador não especificar um tempo de execução, é estabelecida uma velocidade máxima de 50% (*maxVel*). Assim, o servo motor que tiver a maior diferença angular (*maiorDist*) vai ter a máxima velocidade a 50%, enquanto os restantes servos vão ter uma velocidade proporcionalmente inferior e

assegurando com que todos terminam ao mesmo tempo. Do ponto de vista de percentagem, a velocidade Q_{vel} para cada um dos actuadores (identificados pelo Id) é dada por:

$$Q_{vel} [i] = \frac{Q_{dif} [i]}{maiorDist} \times \max Vel(\%) \quad (4.1)$$

Nas situações em que o tempo de execução é especificado basta dividir a diferença entre a posição desejada e a actual (Q_{dif}) pelo tempo de execução (T_x):

$$Q_{vel} [i] = \frac{Q_{dif} [i]}{T_x} (\text{°} / s) \quad (4.2)$$

Para converter esta velocidade expressa em graus por segundo para percentagem são usados os dados do fabricante que indicam uma velocidade máxima de 340°/s. No caso de algum valor exceder os 100% este é truncado, enquanto todas as restantes juntas sofrem uma diminuição proporcional:

$$Q_{vel} [i] = \frac{Q_{vel} [i] \times 100}{\max Vel} (\%) \quad (4.3)$$

Em terceiro lugar, os comandos de movimento das juntas são enviados para os servomotores (através da função `setPosId`) dando-se início ao movimento. Antes desse passo, o tempo fornecido pelo relógio interno do PIC é guardado na variável `start`. Esta variável deve ser inicializada antes de se avançar para o ciclo de leituras/envio dos dados relativos à evolução angular das juntas. Este ciclo começa com a avaliação do tempo actual, $temp = (\text{clock}() - \text{start}) / (\text{FCY} / 1e3)$ em unidades de segundo, seguido do comando `setSpeedReadPos` que efectua o pedido de leitura da posição angular de cada servomotor. Desta forma, guarda-se em simultâneo quer os valores angulares das juntas como os instantes em que foram obtidos para enviar de imediato à unidade central de processamento de acordo com a trama de comunicação já descrita. O tempo entre leituras é de 19 ms. O ciclo termina logo que um dos servomotores atinja a sua posição desejada. Esta condição verifica-se sempre que a diferença entre a posição actual e a desejada for menor do que 1°. Mais tarde esta condição foi alterada, a variável de controlo deixou de ser a posição final desejada e passou a ser o tempo de execução atribuído a cada movimento. Finalmente, para que o Matlab reconheça que o ciclo de leitura terminou é enviado uma trama com o `return1=1`, `return2=2` e `return3=3`.

4.4.2 Movimento ponto-a-ponto

O algoritmo geral para implementação do movimento ponto-a-ponto foi descrito sucintamente no Capítulo 3. Nesta subsecção dá-se importância aos aspectos de implementação na unidade slave e ao facto de existirem algumas diferenças pontuais entre o algoritmo que controla os dois braços e aquele usado para cada perna (suporte ou livre). A forma como foi estruturado o software para a utilização dos dois braços é viável e poderá ser adoptada no futuro quando se pretender mover todos os membros e as respectivas juntas. Os passos mais importantes deste algoritmo estão representados na Figura 4.3 que se segue.

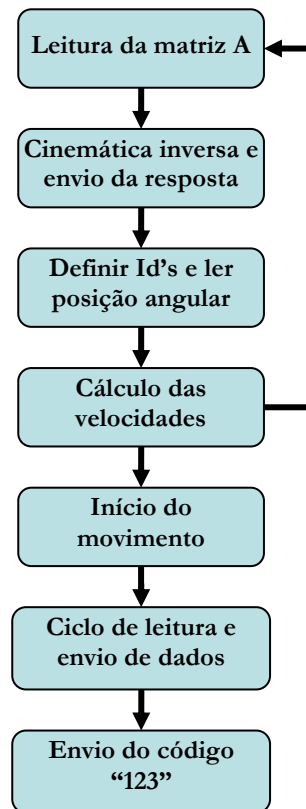


Figura 4.3: Diagrama de blocos do algoritmo de movimento ponto-a-ponto.

Do ponto de vista da leitura da matriz A, quando o objectivo é mover apenas um dos braços, esta é construída colocando na primeira linha as coordenadas cartesianas (XYZ) desejadas para o elemento terminal e, na segunda linha, o tempo de execução (T_x). Se porventura o objectivo é usar ambos os braços em simultâneo, na primeira linha da matriz são colocadas as coordenadas Cartesianas do braço esquerdo (XYZfL), na segunda linha as coordenadas Cartesianas do braço direito (XYZfR) e na terceira linha o tempo de execução do movimento de cada braço (T_{xL} e T_{xR}), juntamente com as flags useL e useR.

Estas flags servem para que o processador fique de imediato a saber se se pretende mexer ambos os braços ou apenas um dos braços e qual. Desta maneira, o processador é utilizado de uma forma mais eficiente evitando a leitura desnecessária de muitas linhas de código, caso o objectivo fosse apenas trabalhar com um membro. Se quisermos usar o braço esquerdo useL fica igual a 1 e se quisermos usar o braço direito useR fica igual a 1, caso contrário devem ficar a zero. Assim, usando um `if (useL == 1)` e um `if (useR == 1)` os quatro primeiros módulos serão ou não executados para cada um dos membros. Por exemplo se `useL` e `useR` forem verdade então serão executados os quatro primeiros módulos para o braço esquerdo, depois voltam a se repetir os quatro módulos para o direito e finalmente será iniciado o movimento de ambos os braços ao mesmo tempo. Ou seja, os restantes três módulos são executados apenas uma vez porque já englobam os dois membros. Este facto está representado no diagrama de blocos anterior por um ramo que foi acrescentado. No caso de apenas trabalharmos com um membro este ramo deixa de existir e os provedores de uso também deixam de ter significado.

Uma vez lidas as coordenadas Cartesianas desejadas para o elemento terminal, é invocada a função de cinemática inversa `invKin()` que devolve directamente os ângulos de cada servomotor. De notar que, para este

algoritmo, esta função devolve directamente os ângulos finais a aplicar aos servomotores. Estes ângulos diferem dos ângulos de cada junta de um valor proporcional às razões de transmissão (apenas nos membros inferiores). A função de cinemática inversa também devolve a flag *isVal* que indica a existência, ou não, de uma solução. Por um lado, se não existir uma solução, o valor da flag é 0 indicando que as coordenadas Cartesianas desejadas estão fora do espaço de trabalho, ou o valor da flag varia entre 2 e *m* indicando a junta que atinge os seus limites físicos. O valor da flag é de seguida enviado para o Matlab permitindo ao utilizador monitorar o estado da operação. Logo de seguida uma condição é apresentada onde o algoritmo apenas é executado if (*isVal* == 1) evitando, neste caso a execução de várias linhas de código. Por outro lado, sempre que a flag *isVal* assume o valor 1 o algoritmo prossegue para o passo seguinte: a definição dos Id's de cada um dos servomotores a actuar. A Tabela 4.7 apresenta as atribuições de Id's para o caso do movimento dos dois braços, da perna de suporte e da perna livre. Esta atribuição resulta do facto de se ter utilizado sempre como perna livre a perna direita e como perna de suporte a esquerda.

Os restantes passos do algoritmo são muito similares aos descritos na subsecção anterior. Novamente o tempo entre leituras é de 19 ms. No caso de se pretender realizar o movimento simultâneo dos dois braços é preciso atender ao maior dos dois tempos de execução (TxL ou TxR) para o sinalizar como o fim da execução dos movimentos. A Figura 4.4 ilustra a interligação entre a unidade slave e a unidade de controlo de cada servomotor baseada num protocolo de comunicação série (Ribeiro, 2010). Primeiro a unidade slave envia a informação relativa à velocidade de cada servomotor (função *setSpeedReadPos*) e, de seguida, procede ao envio das posições angulares de referência (função *setPosId*).

Software	Ordenação das juntas	Vector Id []	Id do servomotor
2× Braços			Braço esquerdo
	Junta sagital do ombro	Id[0]	8
	Junta lateral do ombro	Id[1]	7
	Junta vertical do braço	Id[2]	0
	Junta do cotovelo	Id[3]	6
			Braço direito
	Junta sagital do ombro	Id[4]	11
	Junta lateral do ombro	Id[5]	10
	Junta vertical do braço	Id[6]	0
	Junta do cotovelo	Id[7]	9
Perna livre			Perna direita
	Junta sagital da anca	Id[0]	3
	Junta lateral da anca	Id[1]	1
	Junta vertical da anca	Id[2]	0
	Junta do joelho	Id[3]	5
	Junta sagital do tornozelo	Id[4]	4
	Junta lateral do tornozelo	Id[5]	2
Perna de suporte			Perna esquerda
	Junta lateral do tornozelo	Id[0]	1
	Junta sagital do tornozelo	Id[1]	2
	Junta do joelho	Id[2]	3

Tabela 4.7: Atribuições de Id's aos diferentes servomotores no caso do movimento dos dois braços, da perna livre e da perna de suporte.

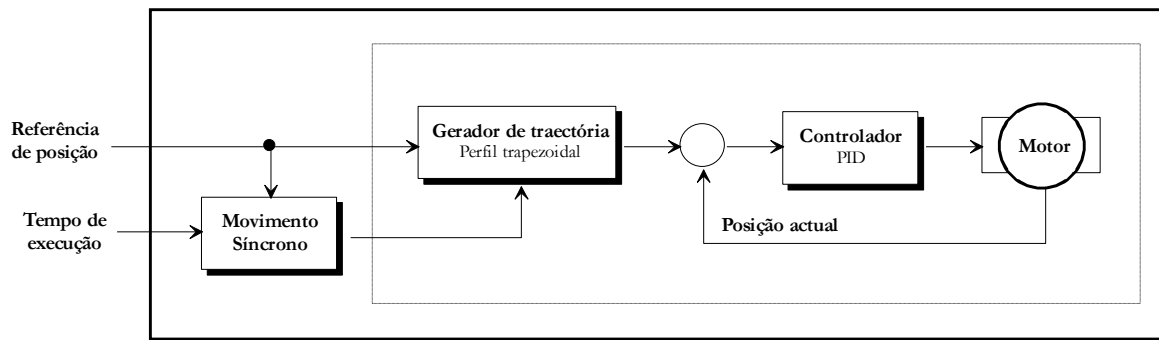


Figura 4.4: As posições e as velocidades angulares (médias) calculadas na unidade slave são os parâmetros de entrada do módulo de geração de trajetórias implementado na unidade de controlo dos servomotores que gera a sequência de referências para um controlador PID.

4.4.3 Movimento contínuo

Para se conseguir realizar o movimento contínuo, em que o elemento terminal (*e.g.*, a mão, o centro do pé, ou a extremidade da anca) executa um percurso em linha recta entre dois pontos, é importante controlar os servomotores em velocidade. Por outras palavras, pretende-se especificar uma posição angular final e variar continuamente o parâmetro velocidade média das juntas ao longo do movimento. A primeira ideia seria calcular a sequência temporal de coordenadas Cartesianas ao longo do percurso em linha recta e, de seguida, usar as equações de cinemática inversa para derivar a respectiva sequência de ângulos das juntas. Contudo, este processo indirecto de controlo em velocidade não pode ser realizado na prática porque sempre que o servo recebe uma nova referência angular o módulo de geração de trajetórias reinicia gerando um perfil trapezoidal com fases de aceleração e desaceleração que resultam em movimentos síncopados e com erros muito significativos na posição final alcançada pelo elemento terminal.

Numa primeira fase, foram realizados alguns testes experimentais para avaliar a possibilidade de implementar o controlo de velocidade nos servomotores HITEC. Na experiência inicial foi usado um servomotor que deveria deslocar-se desde a posição angular inicial (-90°) até à posição final ($+90^\circ$). A intervalos regulares o parâmetro velocidade média é incrementado de 0.3% dando origem ao gráfico de evolução temporal da posição angular (as velocidades são obtidas numericamente) ilustrado na Figura 4.5. É notório o crescimento linear da velocidade em percentagem, e o comportamento em relação a sua posição está de acordo com o desejado em que o deslocamento angular é maior a medida que a velocidade aumenta.

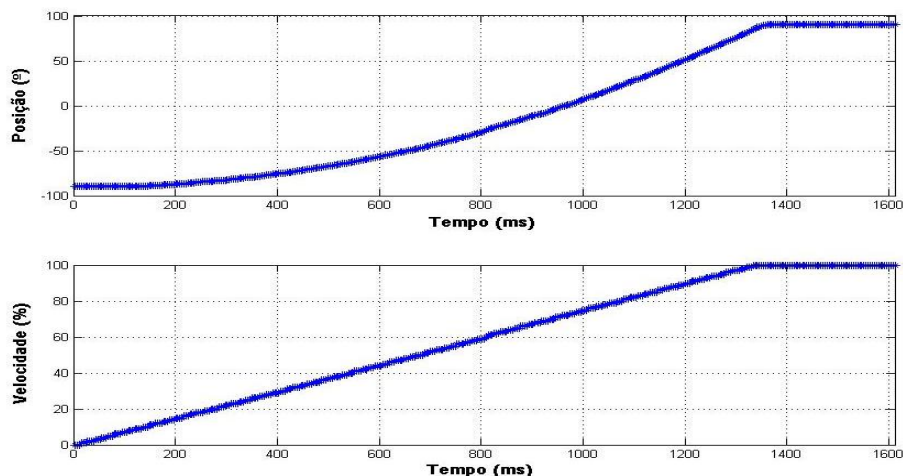


Figura 4.5: Controlo em velocidade impondo incrementos de 0.3% a intervalos regulares.

A próxima experiência resultou da implementação do algoritmo de controlo em velocidade (ver Figura 4.6). Em relação à leitura da matriz A , são extraídos os valores das coordenadas Cartesianas desejadas para o elemento terminal, o tempo de execução do movimento e , ainda, a ordem da função polinomial usada na geração das trajectórias em linha recta. A definição do vector de Id's e a leitura da posição angular actual dos servomotores resulta num algoritmo similar aos anteriores. Contudo, no movimento contínuo aproveita-se para colocar inicialmente a zero o parâmetro velocidade média, usando a mesma função de leitura da posição actual (*setSpeedReadPos*). O planeamento de trajectórias no espaço Cartesiano implica ainda o conhecimento da posição actual do elemento terminal, ou seja, o cálculo da cinemática directa (função *dirKin*) da sub-estrutura de interesse. A função de cinemática inversa é usada logo de seguida para indicar se a posição final do elemento terminal é alcançável através do valor da flag *isVal*. Antes de entrarmos no ciclo associado ao movimento contínuo, é preciso ainda inicializar a zero alguns parâmetros, tais como o tempo actual (t_a), o tempo antigo (t_o) e dois contadores (j e k), bem como inicializar o vector das juntas e o vector “posição angular antiga”.

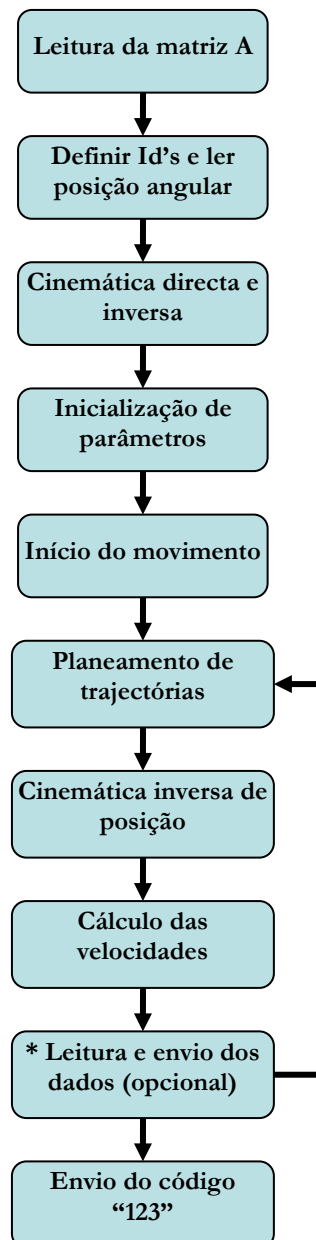


Figura 4.6: Diagrama de blocos do algoritmo de movimento contínuo.

Ao mesmo tempo, é preciso salvar o tempo actual em *start* e lançar a função *setPosId()* para dar início ao movimento e indicar aos servos a posição angular final desejada. Como as velocidades dos servos foram, inicialmente, colocadas a zero o movimento será suave evoluindo de acordo com o planeamento adoptado. Neste momento, é iniciado um ciclo que se repete enquanto o tempo actual (t_a) for menor que o tempo de execução (T_x) fornecido pelo utilizador. Nos movimentos contínuos a especificação do tempo de execução é essencial para o correcto funcionamento do ciclo. No caso de o utilizador não escolher esse tempo de execução é usado um valor por defeito pré-definido no Matlab. A taxa máxima a que este ciclo se repete vai depender da sequência de instruções a processar e terá, certamente, influência no desempenho final. As funções incluídas são o planeamento de trajectória que calcula as coordenadas cartesianas do elemento terminal, a cinemática inversa para transformar essas coordenadas em posições angulares, o cálculo numérico das velocidades das juntas com base nas posições angulares actual/anterior e no tempo actual/anterior, e o envio da informação para os servomotores. Pelos resultados obtidos um ciclo demora 23 ms (para um membro).

A função de planeamento de trajectórias *CartTrajPlanning* tem como parâmetros de entrada as coordenadas Cartesianas iniciais e finais, o tempo de execução do movimento, o tempo actual e a ordem da polinomial (3ª ou 5ª ordem). Durante as várias repetições deste ciclo o único valor que se vai alterando é o tempo actual que está a ser constantemente actualizado no início do ciclo. Para permitir a visualização dos dados experimentais, é possível incluir o bloco associado à leitura e envio de dados. Este bloco está representado no diagrama com um asterisco (*) porque só é utilizado nestas circunstâncias. De notar que se aproveita a função *setSpeedReadPos()* para enviar os valores das velocidades angulares de referência e, ao mesmo tempo, proceder à leitura das posições angulares de cada servomotor. Isto significa que apenas se acrescenta ao tempo de ciclo o período necessário para enviar os dados para a unidade central. Este aumento traduz-se em maiores intervalos de actualização das velocidades o que torna o movimento menos preciso e/ou mais irregular.

Uma vez implementado o algoritmo, realizaram-se várias experiências práticas para demonstrar a sua efectividade. Os perfis de velocidades vão ser necessariamente diferentes daqueles obtidos no movimento ponto-a-ponto. E mesmo no movimento contínuo, a evolução das velocidades entre os diferentes servos é muito variada. A Figura 4.7 compara a evolução da velocidade de um determinado servomotor durante a realização do movimento ponto-a-ponto (asteriscos azuis) e do movimento contínuo (asteriscos verdes), especificando as mesmas coordenadas Cartesianas finais e o mesmo tempo de execução. No movimento ponto-a-ponto, a velocidade permanece constante e as fases de aceleração e desaceleração típicas nos perfis trapezoidais não são visíveis (seriam necessários mais pontos para reconstruir estas rampas). Por sua vez, no movimento contínuo estas fases são bem visíveis, mas com a forma de uma parábola (um aumento e diminuição mais gradual) onde a velocidade máxima aumenta significativamente.

Numa outra experiência, foi realizado um movimento contínuo usando os 3-gdl da perna de suporte. Os resultados da evolução temporal da velocidade dos servomotores estão ilustrados na Figura 4.7. É notória a diferença entre os perfis de velocidade obtidos para cada um dos servos de forma a assegurar a realização do trajecto desejado em linha recta. Uma característica que é igual em todos os servos é que existe sempre uma fase de aceleração e de desaceleração gradual, por vezes mais ou menos acentuadas. Outra característica importante é o facto de, regra geral, os servomotores terminarem o movimento numa configuração muito próxima da desejada com diferenças de um grau ou por vezes até menos.

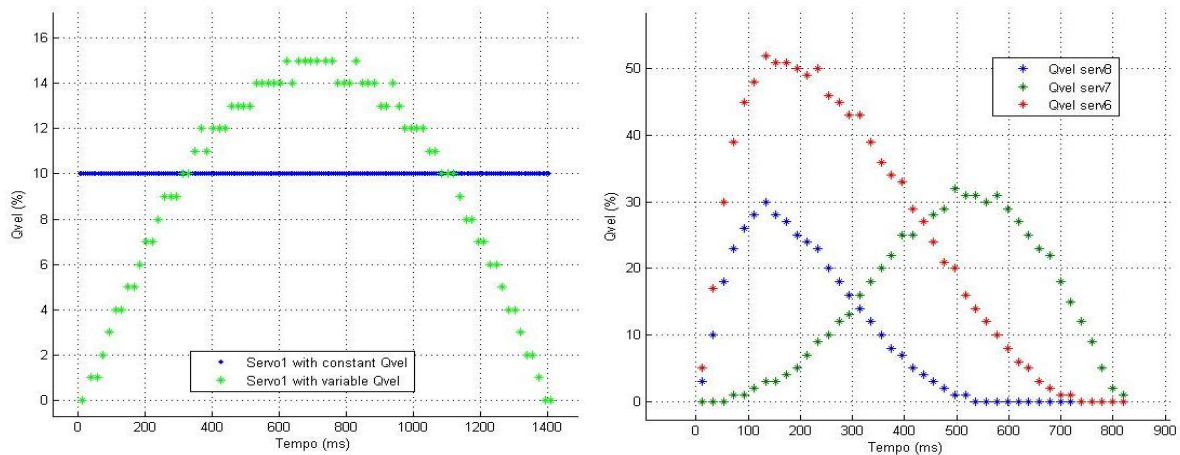


Figura 4.7: Evolução temporal da velocidade de um servomotor (esquerda) ao realizar um movimento ponto-a-ponto (asteriscos azuis) e um movimento contínuo (asteriscos verdes); e dos servomotores da perna de suporte (direita) ao realizar um movimento contínuo.

Com o decorrer do trabalho verificou-se que, em algumas situações, o algoritmo de controlo em velocidade dava origem a movimentos que não seguiam uma linha recta, ainda que os servomotores terminassem o movimento nas posições finais desejadas. O que acontecia é que alguns movimentos exigiam a um, ou mais, servomotores um deslocamento desde um ângulo inicial até um ângulo final, mas com uma passagem intermédia nesse ângulo final. Este facto resulta do movimento de inversão de uma junta durante a execução de um movimento. No algoritmo anterior, os servomotores eram comandados para se deslocarem para a posição final de referência. O facto de ocorrer uma inversão no movimento significa que algures no intervalo de execução há uma passagem pela posição angular final. Desta forma, sempre que a posição angular do servo atinge o ângulo final o servomotor termina a execução do movimento (mesmo se o parâmetro velocidade é diferente de zero). A Figura 4.8 ilustra, para diferentes servomotores, situações em que a evolução angular é monótona e um caso particular de uma junta em que tal não se verifica dando origem a erros na execução do movimento contínuo.

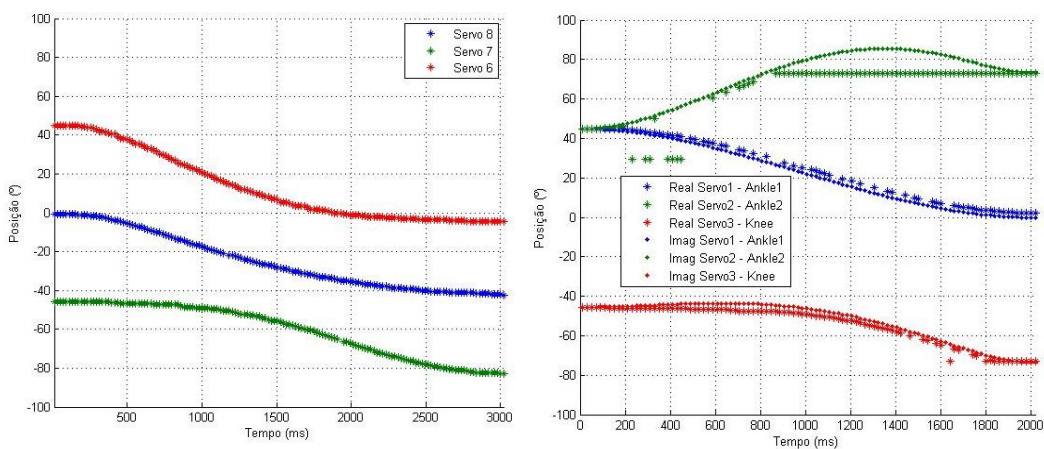


Figura 4.8: Exemplos em que a evolução temporal das posições angulares dos servos é monótona e o movimento bem sucedido (esquerda) e em que a existe uma passagem intermédia pela posição angular final que dá origem a erros de seguimento (direita). Os pontos verdes indicam a evolução angular necessária para a realização do movimento em linha recta, enquanto os asteriscos verdes ilustram os erros angulares introduzidos pelo algoritmo anterior.

No gráfico da esquerda estamos perante um movimento simples em que, para cumprir o percurso desejado, nenhum dos servos tem a necessidade de inverter o seu próprio movimento. Em relação ao gráfico da direita, é possível constatar que as posições angulares reais, representadas por asteriscos, e as posições angulares desejadas, representadas com pontos, não coincidem. O problema surge quando o servomotor atinge a posição angular final aos 840 ms e permanece nessa posição até ao final do movimento. Para corrigir este problema, o algoritmo de controlo em velocidade foi modificado de maneira a entrar em linha de conta com fenómeno de inversão no sentido de rotação. A estrutura do algoritmo mantém-se, à parte alguns pormenores. No essencial, deixa-se de enviar cada servomotor para a posição angular final e passa-se a usar os limites físicos de $\pm 90^\circ$. Ao mesmo tempo, sempre que ocorrem variações no sinal da velocidade angular é preciso actualizar a posição angular final considerando o limite oposto.

Em seguida são apresentados e discutidos alguns resultados experimentais que mostram a realização do mesmo movimento da perna de suporte sem e com as correcções no algoritmo de movimento contínuo. Os movimentos realizados têm, precisamente, as mesmas características, ou seja, um movimento contínuo com as mesmas coordenadas iniciais e finais e com o mesmo tempo de execução de 3 segundos. A Figura 4.9 apresenta as posições angulares obtidas usando o algoritmo inicial, sem correcção (à esquerda), e o movimento realizado pelo elemento terminal (à direita). As curvas com a mesma cor representam, respectivamente, as posições angulares de referência (pontos) e as medidas pela leitura dos servomotores (asteriscos). O problema ocorre com a junta do joelho (servo 3 a vermelho) em que o valor angular inicial e final são iguais. As velocidades calculadas para este não são nulas, mas como o seu objectivo de chegar a posição angular final é cumprido logo à partida, este permanece nessa posição, o que faz com que a trajectória do elemento terminal não segue o trajecto em linha recta. A correcção introduzida no algoritmo permite eliminar o problema e assegurar a realização de um movimento muito próximo do desejado. A Figura 4.10 apresenta os resultados obtidos com a mesma experiência, mas agora com a correcção no algoritmo. O gráfico da direita mostra como a evolução da posição do elemento terminal aproxima a trajectória em linha recta especificada.

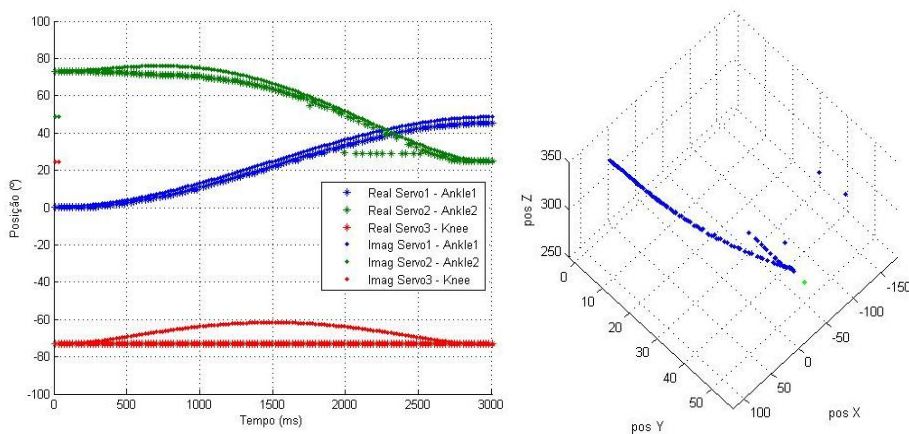


Figura 4.9: Movimento contínuo entre dois pontos: evolução temporal das posições angulares (esquerda) usando o algoritmo inicial e representação da posição do elemento terminal no espaço Cartesiano (direita).

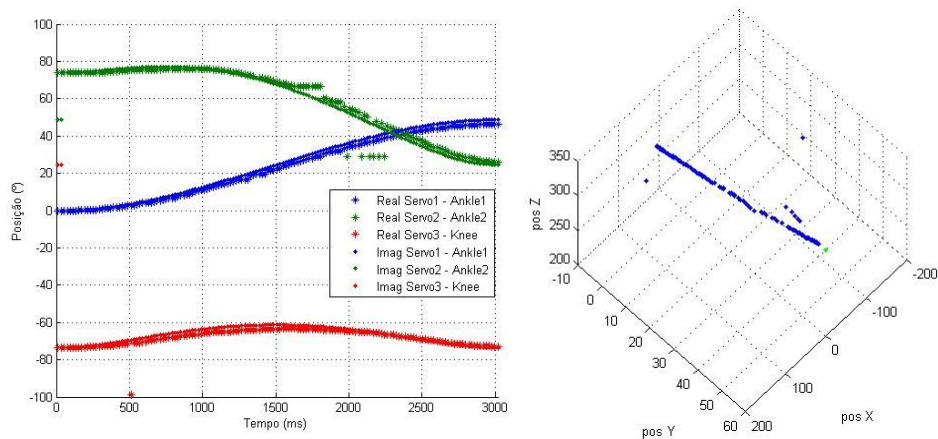


Figura 4.10: Movimento contínuo entre dois pontos: evolução temporal das posições angulares (esquerda) usando o algoritmo corrigido e representação da posição do elemento terminal no espaço Cartesiano (direita).

Numa outra experiência, compara-se o resultado dos dois algoritmos ao executar uma sequência de movimentos contínuos, com um tempo de execução de 2 segundos cada, que no seu conjunto formam um percurso fechado na forma de um losango (Figura 4.11 e Figura 4.12). Para além das posições reais do elemento terminal, representadas por pontos azuis, é sobreposta uma linha tracejada a verde indicando a trajetória ideal. Em qualquer das experiências anteriores é possível verificar a existência de erros nas leituras das posições angulares fornecidas pelos servomotores (e nas respectivas coordenadas Cartesianas). Porém, estes erros não interferem no desempenho dos algoritmos de movimento implementados na unidade slave uma vez que não existe, para já, uma malha de realimentação fechada em torno destas variáveis (a cinemática inversa é implementada em malha aberta).

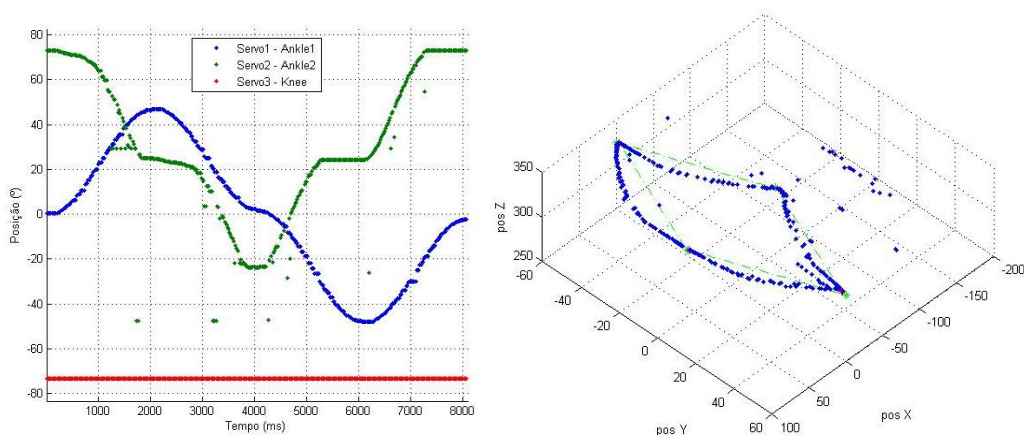


Figura 4.11: Movimento contínuo na forma de losango: evolução temporal das posições angulares (esquerda) usando o algoritmo inicial e representação da posição do elemento terminal no espaço Cartesiano (direita).

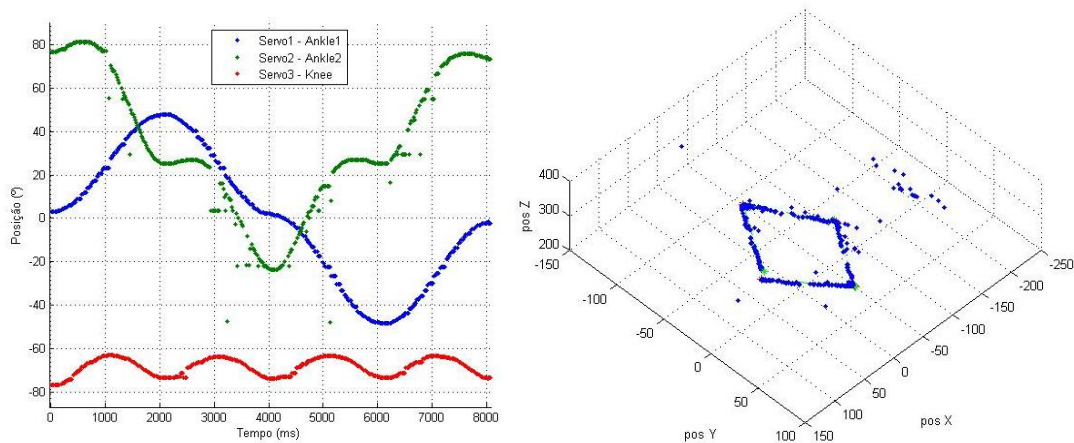


Figura 4.12: Movimento contínuo em forma de losango: evolução temporal das posições angulares (esquerda) usando o algoritmo corrigido e representação da posição do elemento terminal no espaço Cartesiano (direita).

4.5 Optimização dos algoritmos

Nesta secção abordam-se duas questões relacionadas com a optimização do algoritmo de planeamento e controlo do movimento contínuo. Primeiro, analisa-se o problema dos erros nas leituras sensoriais das posições angulares dos servos e avalia-se em que medida é possível usar uma estimativa dessas posições. Segundo, é feita uma correcção no valor das velocidades angulares calculadas na unidade central face à influência no comportamento do servomotor de se efectuarem, em paralelo com a execução do movimento, leituras sucessivas dos sensores de posição.

4.5.1 Erros nas leituras sensoriais

A existência de erros nas leituras da posição angular dos servomotores já tinha sido mencionada e analisada em trabalhos anteriores (Ribeiro, 2010). O problema pode ser ilustrado através da representação das leituras das posições angulares de três servomotores usados na realização de uma sequência de movimentos com velocidade angular máxima de 20% (Figura 4.13) e de 100% (Figura 4.14).

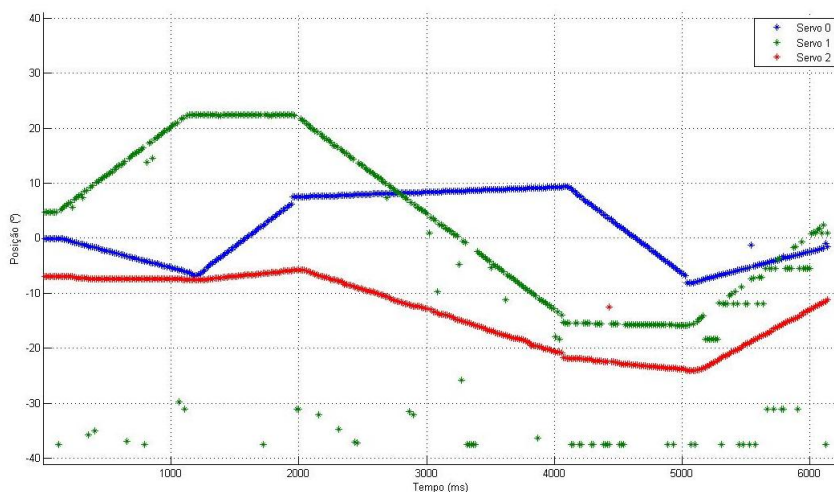


Figura 4.13: Sequência de movimentos da perna com a velocidade máxima a 20%.

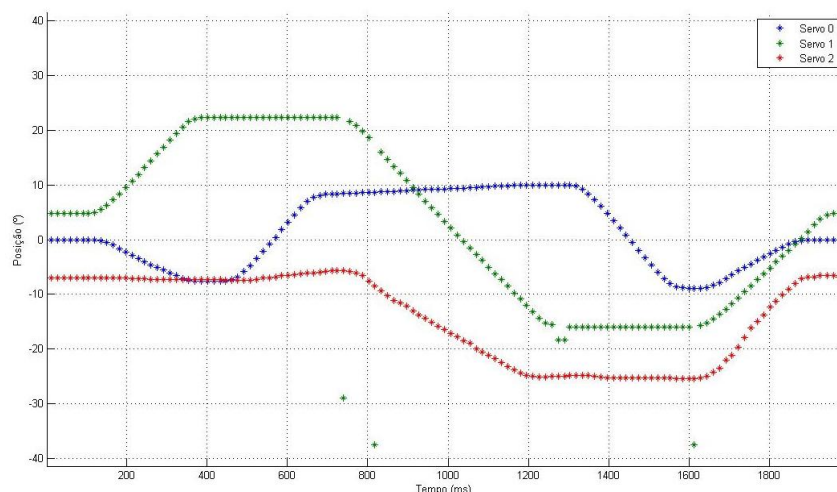


Figura 4.14: Sequência de movimentos da perna com a velocidade máxima a 100%.

As experiências foram realizadas com a mesma sequência de movimentos de maneira a facilitar a análise o que sucede ao variar a velocidade máxima e o tempo entre cada leitura. A Figura 4.13 e a Figura 4.14 descrevem os valores das leituras de posição angular de três servomotores usando uma velocidade máxima, respectivamente, de 20% e 100% para aquele com maior variação angular. Isto significa que o tempo que necessário para executar a mesma sequência de movimentos é de, aproximadamente, 6.3 e 2.0 segundos.

Uma questão importante é a influência do número de servos ligados ao barramento de comunicação (ver Figura 2.7). As experiências descritas na dissertação do Miguel Ribeiro (Ribeiro, 2010) mostram, contudo, que não existe uma relação directa entre os erros nas leituras e o número de servos no barramento (Tabela 4.8). Um outro aspecto curioso é o facto de estes erros não aparecerem nas experiências realizadas com os braços, mesmo para uma grande quantidade de leituras. Uma vez que os motores digitais nos braços e pernas são iguais, a única diferença está na existência de correias de transmissão nas pernas que originam uma redução na velocidade real das juntas. Nesta linha de ideias, foram realizadas algumas experiências adicionais com as pernas para identificar o problema.

Analisando a percentagem de erros de leituras com o aumento da velocidade verifica-se que o número de ocorrências é aproximadamente igual. A Figura 4.15 mostra os resultados obtidos para a mesma sequência de movimentos usando como velocidade máxima os 20%. A grande diferença reside na introdução de um maior tempo entre as leituras de cada servo. Assim, entre as leituras dos servos 0 e 1 foi introduzido um espaçamento de 3 ms e entre as leituras dos servos 1 e 2 outro espaçamento de 3 milissegundos.

Servos no Barramento	Pedidos de Leitura (de um servo)	Erros na Leitura (de um servo)	Percentagem de Sucesso
1	871	190	78.19
2	109	109	75.17
3	70	70	76.03

Tabela 4.8: Variação nos erros de leitura variando o número de servos no barramento de comunicação.

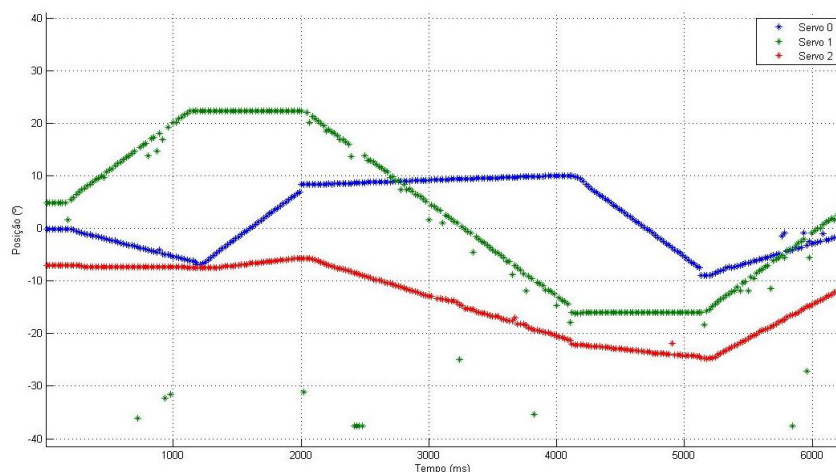


Figura 4.15: Sequência de movimentos da perna com a velocidade máxima a 20%, mas agora com um tempo maior entre as leituras de posição exigidas aos servomotores.

Verifica-se uma melhoria significativa com uma menor ocorrência de erros que, parece indicar, que o parâmetro mais importante é o intervalo entre leituras sucessivas. Uma justificação para explicar estes erros é o facto do controlador interno do servomotor funcionar com prioridades, dando prioridade ao mecanismo de controlo de posição em detrimento da resposta ao pedido externo.

Apesar de não ter sido possível determinar a origem do problema, este foi analisado de várias vertentes permitindo uma melhor compreensão do mesmo. Ao mesmo tempo, esta dificuldade motivou avanços noutra direcção, ligada à substituição da leitura errada da posição angular por um valor estimado (ou teórico). Mais concretamente, as leituras das posições angulares recebidas pela unidade slave são comparadas com os valores estimados a partir das posições e das velocidades angulares derivadas no instante anterior. Sempre que a diferença é maior que um valor limite, assume-se que a leitura está errada e o valor da posição angular é substituído pela sua estimativa.

Foram realizadas duas experiências para avaliar o efeito desta alteração sobre as leituras sensoriais efectuadas (Figura 4.16). No gráfico à esquerda encontram-se sobrepostos os gráficos da evolução temporal da posição angular de um servomotor obtidos pela leitura e pela estimação. O vector final combina as leituras efectuadas correctamente com alguns valores estimados sempre que ocorre um erro de leitura (a diferença entre o valor real e o estimado é maior que 6°). Como já foi referido, os servomotores operam na gama de -90° a $+90^\circ$. Assim, sempre que o valor estimado ultrapasse estes limites, será truncado aos limites (este pormenor não é visível no gráfico propositadamente de maneira a se perceber facilmente a diferença). O gráfico à direita ilustra o resultado de uma sequência de movimentos da perna de suporte em que são utilizados os 6 servomotores acoplados. A correcção introduzida permite eliminar de forma efectiva as leituras inconsistentes, abrindo desde já a possibilidade de implementação de algoritmos de cinemática inversa em malha fechada ou de malhas externas de controlo com realimentação baseada nas posições angulares das juntas.

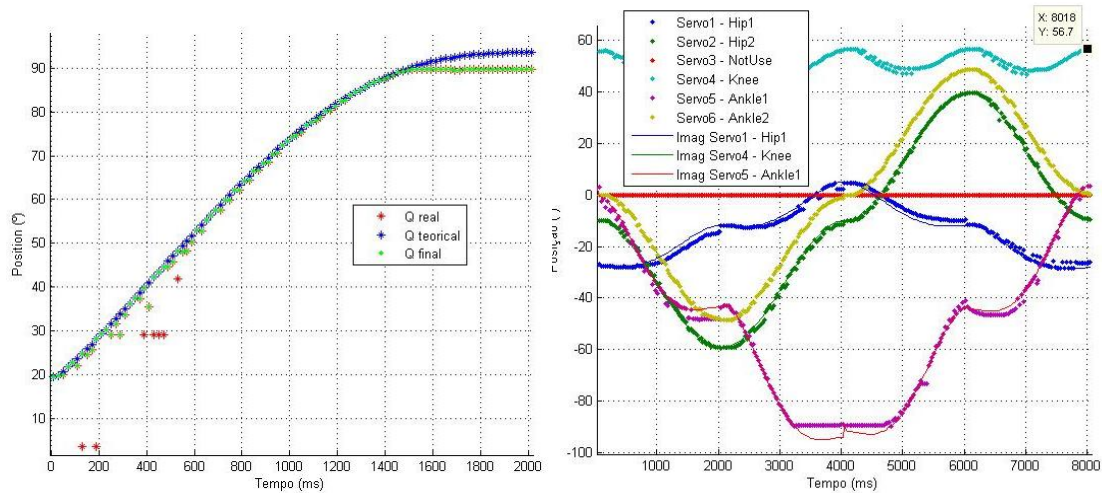


Figura 4.16: Eliminação dos erros de leitura (esquerda) e aplicação a uma sequência de movimentos (direita).

4.5.2 Correção na velocidade dos servos

Sempre que, em simultâneo com a execução de um dado movimento, são enviados comandos para o servomotor (*e.g.*, pedido da leitura dos sensores) verifica-se que a sua velocidade tende a diminuir. O próprio fabricante (Hitec RCD, 2011) indica uma relação empírica entre a velocidade e a frequência dos comandos enviados dada por:

$$v = A_s(1 - Br) \quad (4.4)$$

em que v é a velocidade real do servo (em tics/s), s é o parâmetro velocidade média desejada (em percentagem), r é a frequência dos comandos enviados ao servo (s^{-1}), A e B são constantes definidas como sendo $A = 37.46$ tics/s e $B = 0.001106$ s. O fabricante indica (Hitec, “General Specification of HSR-5980SG Coreless Digital Robot Servo”) que a velocidade máxima (100%) do servo alimentado a 6 V (0.17s @ 60°) e a 7.4 V (0.14s @ 60°) é, respectivamente, de 353°/s (3529 tics/s) e 428°/s (4286 tics/s).

Estimando a frequência média a que as leituras sensoriais são efectuadas (tempo de execução de cada ciclo de movimento) é possível determinar um valor corrigido para a velocidade angular. Para analisarmos as diferenças produzidas pela introdução desta correcção, foram realizados dois movimentos contínuos usando as mesmas coordenadas Cartesianas iniciais/finais e o mesmo tempo de execução. Num dos movimentos o cálculo das velocidades é feito através da forma usada até aqui, enquanto no outro movimento as velocidades são calculadas utilizando a correcção anterior. Estas experiências foram realizadas com a perna de suporte cujo elemento terminal inicia o movimento nas coordenadas $XYZ_i = [-60 \ 0 \ 325]mm$ e tem como objectivo chegar às coordenadas $XYZ_f = [60 \ 0 \ 325]mm$, como podemos observar pelos gráficos da evolução temporal das posições e velocidades angulares (Figura 4.17).

Um outro problema relacionado, é a influência da tensão de alimentação no valor da velocidade máxima atingida pelos servomotores. Este efeito pode ser observado na Figura 4.18 comparando as respostas do servomotor ao realizar um movimento entre os seus limites físicos ($\pm 90^\circ$) com uma velocidade de 100%, usando uma tensão de alimentação de 6 V e 7.4 V.

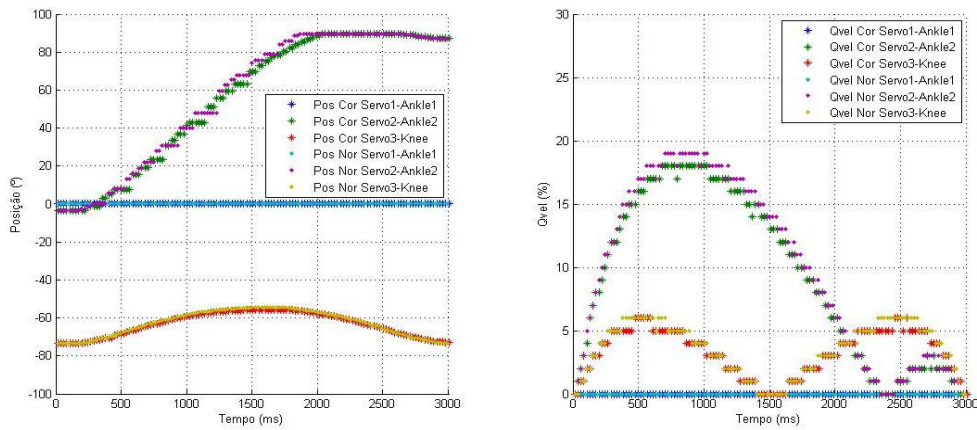


Figura 4.17: Comportamento dos servomotores sem e com a correcção no valor das velocidades angulares.

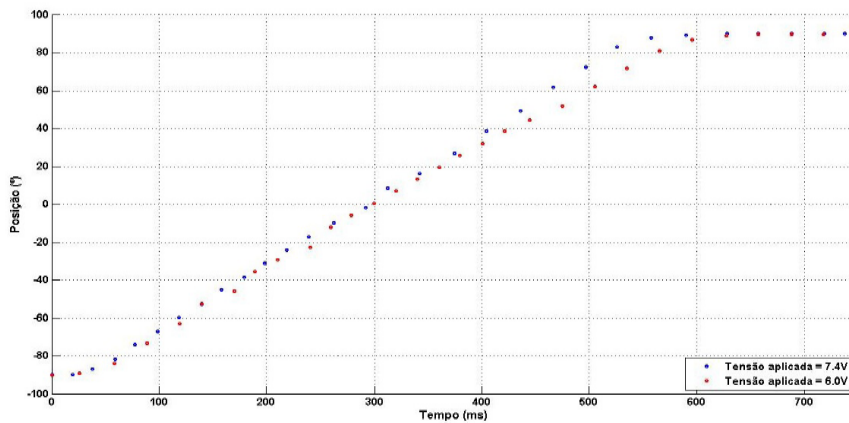


Figura 4.18: Variação na posição angular do servomotor ao realizar um movimento entre os seus limites ($\pm 90^\circ$) usando uma tensão de alimentação de 6 V e 7.4 V.

Como seria de esperar, uma diminuição na tensão de alimentação provoca uma diminuição da velocidade e, por conseguinte, o tempo de execução do movimento aumenta. O fabricante assume que a velocidade máxima do servomotor é de $352^\circ/\text{s}$ para 6 V e $428^\circ/\text{s}$ para 7.4 V. Contudo, os resultados obtidos (Figura 4.19) com um servomotor alimentado a 7.4 V, a operar à velocidade máxima, revelam comportamentos diferentes quando repetido sucessivamente. Este gráfico evidencia que os servomotores, na realidade, não cumprem com o que é estipulado pelo fabricante. Podemos verificar que a maior concentração de pontos está situada em torno dos 185 ms e com uma posição angular final em entre os 62° e os 64° .

Comparando estes valores com os do fabricante verifica-se que o menor erro é de cerca de 30 ms. Isto significa que a velocidade máxima efectiva rondará, aproximadamente, os $340.5^\circ/\text{s}$. Todas as experiências realizadas até este momento assumiram um valor para a velocidade máxima (100%) do servomotor de $340^\circ/\text{s}$ (e não os $352^\circ/\text{s}$). Apesar de não se ter feito uma avaliação exaustiva do impacto das leituras sensoriais e da tensão de alimentação sobre os movimentos realizados, é de prever que se possa otimizar o desempenho global do sistema levando em linha de conta as correcções no valor da velocidade máxima dos servomotores.

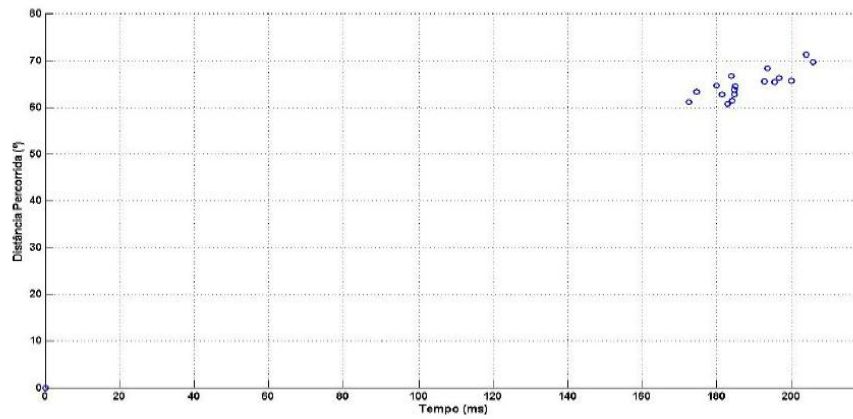


Figura 4.19: Posição angular final e tempo de execução obtidos à máxima velocidade para a repetição do mesmo movimento desejado (deslocamento angular de 60°). (Ribeiro, 2010)

4.6 Resultados experimentais

Uma vez resolvidos alguns problemas associados à implementação dos diversos algoritmos de controlo do movimento, procedeu-se à realização de diversas experiências, incidindo sobre o movimento contínuo e envolvendo os dois braços, a perna de suporte e a perna livre, e cujos resultados são descritos nas próximas subsecções.

4.6.1 Movimento dos braços

Na primeira experiência pretendia-se deslocar o elemento terminal do braço esquerdo das coordenadas iniciais $XYZ_i = [91 \ 218 \ 31]mm$ para as coordenadas finais $XYZ_f = [91 \ 140 \ -47]mm$ num tempo de execução de 3 segundos. Como podemos observar pelos gráficos da Figura 4.20, trata-se de um movimento contínuo realizado sem que se verifiquem erros nas leituras das posições angulares. Ao mesmo tempo, verifica-se que existe um bom seguimento da trajectória desejada, com todas as juntas a atingirem as posições finais desejadas no tempo previsto. Este movimento simples vai ser útil para fazermos uma comparação com os gráficos que se seguem: o uso de apenas um braço versus o uso dos dois braços ao mesmo tempo.

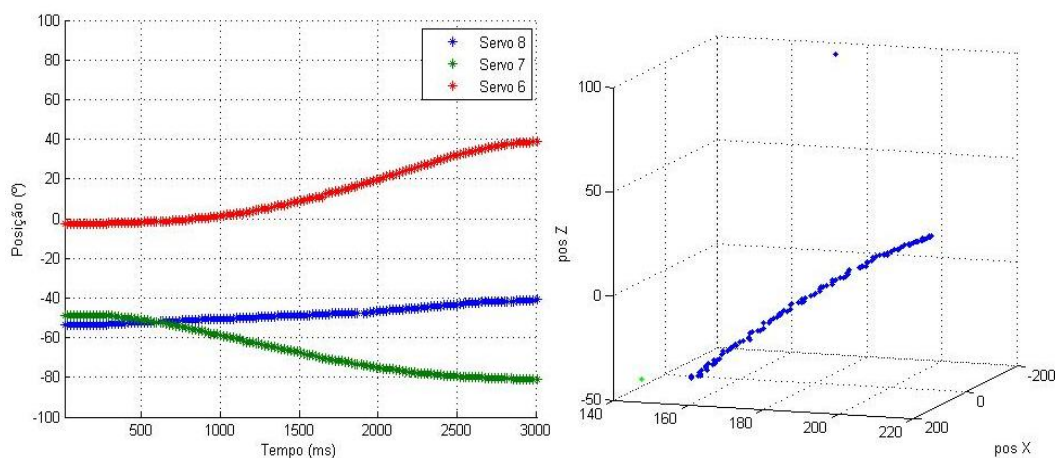


Figura 4.20: Evolução temporal das posições angulares (esquerda) para um movimento contínuo do braço esquerdo (direita).

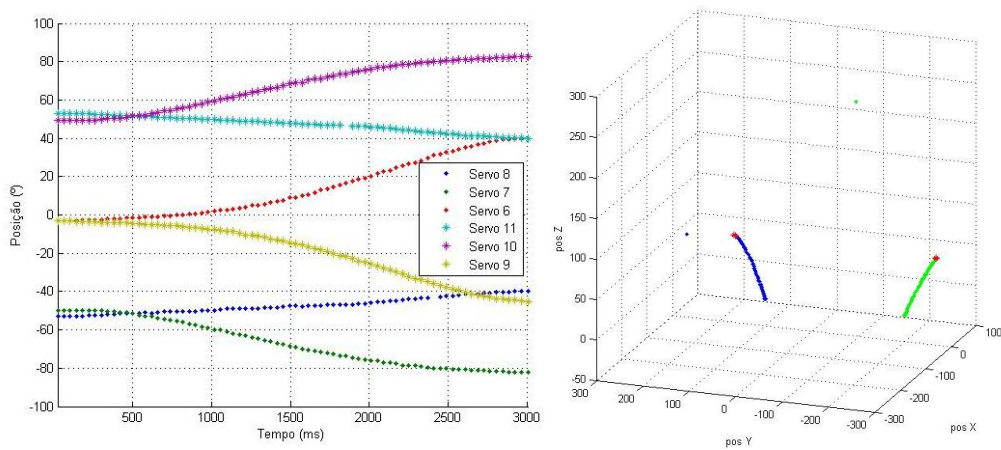


Figura 4.21: Evolução temporal das posições angulares (esquerda) para um movimento contínuo dos dois braços (direita).

A Figura 4.21 ilustra esta última situação em que os dois braços são controlados em simultâneo para realizarem um movimento contínuo especificando trajetórias simétricas e com o mesmo tempo de execução de 3 segundos. A principal diferença e a razão da comparação entre estas duas experiências, reside no tempo do ciclo que envolve o cálculo e envio dos valores de velocidade angular para os servomotores, bem como o envio para a unidade central das leituras das posições angulares e respectivos instantes de amostragem. O tempo do ciclo de actualização é de aproximadamente 20 ms no caso de um braço e de cerca de 40 ms para os dois braços, sem que este aumento no tempo de actualização afecte o bom comportamento global do sistema.

A Figura 4.22 mostra os resultados obtidos para uma sequência de seis movimentos contínuos realizados pelo braço esquerdo de maneira a criar a forma de um quadrado de 78 mm de lado no plano yz (intersecta o eixo dos xx nos 91mm) e de um triângulo no plano xz (intersecta o eixo dos yy nos 140mm). Para realizar esta sequência de movimentos é criada uma matriz que agrupa estes movimentos, um em cada linha, definidos pelas coordenadas Cartesianas desejadas para o braço esquerdo e o respectivo tempo de execução, as coordenadas Cartesianas desejadas para o braço direito (em milímetros) e o seu tempo de execução (em segundos), os valores que indicam se o braço esquerdo, direito ou ambos vão ser usados e, finalmente, a ordem da função polinomial usada na geração de trajetórias (ver a matriz na página seguinte).

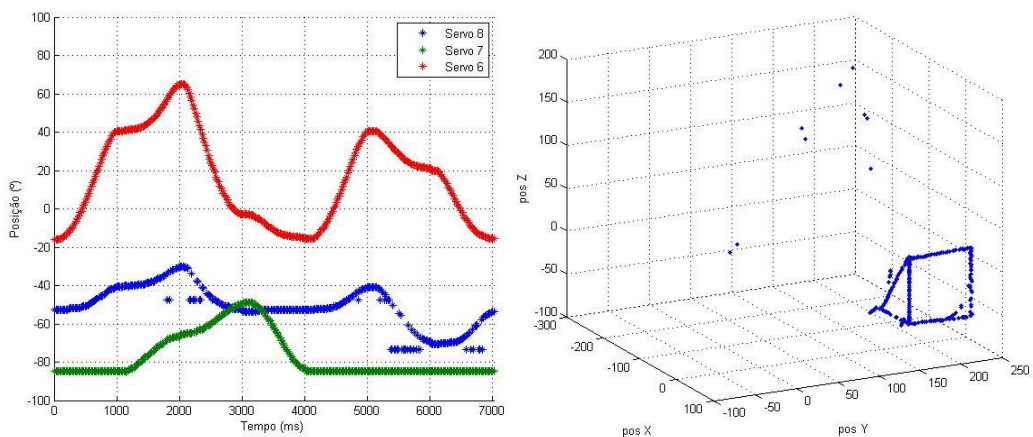


Figura 4.22: Evolução temporal das posições angulares (esquerda) para um movimento composto do braço esquerdo (direita).

X(mm)	Y(mm)	Z(mm)	TxL(s)	X(mm)	Y(mm)	Z(mm)	TxR(s)	useLuseR	ord	
91	140	-47	1	0	0	0	0	1	0	3
91	218	-47	1	0	0	0	0	1	0	3
91	218	31	1	0	0	0	0	1	0	3
91	140	31	1	0	0	0	0	1	0	3
91	140	-47	1	0	0	0	0	1	0	3
91	140	31	1	0	0	0	0	1	0	3

O gráfico relativo à evolução temporal das posições angulares (esquerda) permite identificar cada um dos seis movimentos programados e a forma como planeamento no espaço Cartesiano é traduzido em variações angulares. Do ponto de vista do elemento terminal (mão esquerda), os movimentos desejados foram bem executados em ambos os planos, com trajectórias rectilíneas e sem descontinuidades.

4.6.2 *Perna livre*

A avaliação do comportamento da perna livre foi realizada programando uma sequência de movimentos em linha recta em que a extremidade do pé define um percurso fechado em forma de quadrado no plano xy. Esta sequência de movimentos é realizada invocando o movimento contínuo e inclui as leituras sensoriais. Nestes movimentos com a perna livre são realizados os cálculos das velocidades para os seis servos que constituem este membro e, por sua vez, as leituras das posições angulares dos mesmos. A matriz utilizada para especificar os movimentos usa as posições Cartesianas desejadas, o tempo de execução e a ordem da função polinomial:

X(mm)	Y(mm)	Z(mm)	Tx(s)	Ord
0	60	-16	2	3
-60	0	-16	2	3
0	-60	-16	2	3
60	0	-16	2	3

As curvas da evolução temporal da posição angular dos seis servomotores são apresentadas na Figura 4.23 (esquerda). De notar que, durante a realização desta sequência de movimentos o servomotor 3, associado à junta vertical do ombro, permanece fixo na posição 0°. Sobrepostas a estas curvas, encontram-se as posições angulares teóricas (as linhas) obtidas na unidade slave com os parâmetros de saída da função de cinemática inversa. Primeiro, é notória a ocorrência de muitos erros de leitura onde o corrector tem de intervir, substituindo o valor lido pelo estimado. Verifica-se ainda que estes valores teóricos estimados introduzidos estão ligeiramente desalinhados em relação aqueles que formam o percurso original das posições angulares. Estes pequenos desfasamentos angulares vão reflectir-se em desvios nas posições Cartesianas do elemento terminal, tal como se mostra no lado direito da Figura 4.23. Todavia, deve referir-se que o desvio entre a trajectória desejada (tracejado verde) e a trajectória obtida a partir dos ângulos das juntas (com correcção das leituras sensoriais) resulta do próprio processo de leitura. Na realidade, estes desvios não ocorrem, mantendo-se um movimento suave com boa capacidade de seguimento da trajectória desejada. Esta sequência de movimentos comprova a existência de alguns pontos críticos relacionados com o fim de um movimento e o início do seguinte.

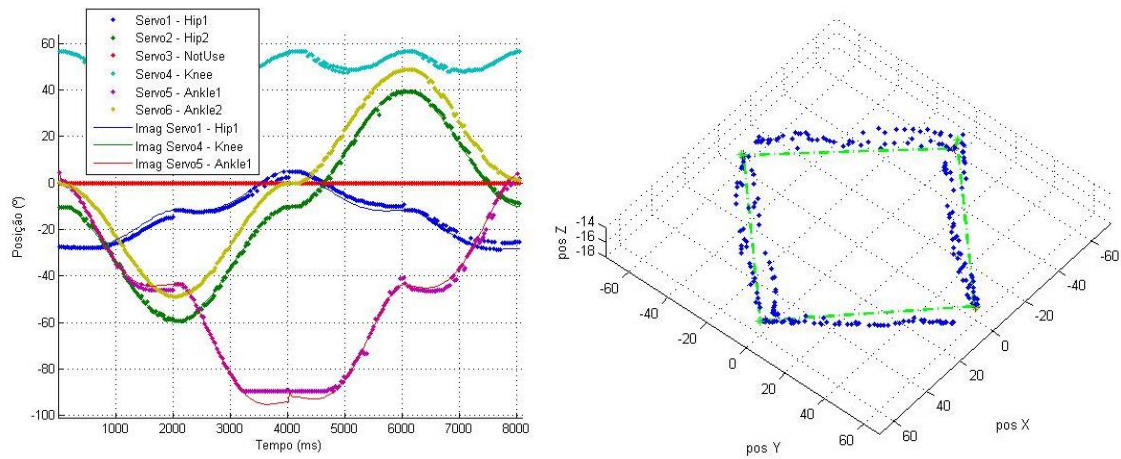


Figura 4.23: Evolução temporal das posições angulares (esquerda) para um movimento composto da perna livre (direita).

4.6.3 Perna de suporte

Finalmente, a perna de suporte (3-gdl) foi programada de forma a realizar uma sequência de movimentos da sua extremidade (a anca). Neste caso, foi realizada uma sequência de quatro movimentos contínuos no plano xy com o objectivo de mover a anca ao longo de um percurso fechado em forma de quadrado num intervalo de 2 segundos para cada movimento. Os três servomotores em causa são os do tornozelo (servo 1 – lateral e servo 2 - sagital) e um do joelho (servo 3). A matriz abaixo indica as coordenadas Cartesianas finais da anca (em mm), o tempo de execução (em segundos) e a ordem da função polinomial:

X(mm)	Y(mm)	Z(mm)	Tx(s)	Ord
0	60	325	2	3
-60	0	325	2	3
0	-60	325	2	3
60	0	325	2	3

A maioria dos comentários apresentados para os braços e a perna livre podem ser directamente aplicados à sequência de movimentos da perna de suporte (Figura 4.24). Mais uma vez, a execução desta sequência de movimentos foi bem sucedida, com bons resultados no seguimento do percurso desejado em linha recta.

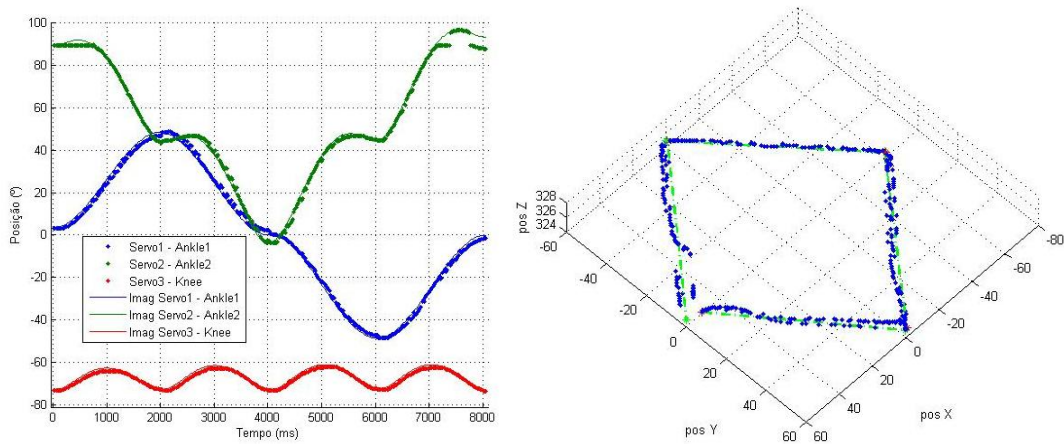


Figura 4.24: Evolução temporal das posições angulares (esquerda) para um movimento composto da perna de suporte (direita).

Por fim, deve-se destacar que o movimento contínuo foi implementado de tal forma que podem ocorrer situações onde o elemento terminal se afasta da trajectória desejada (pelo menos em certos períodos) porque alguma junta atinge o seu limite físico ou situações em que o tempo de execução não é cumprido porque a velocidade angular requerida ultrapassa o valor máximo. Em qualquer das circunstâncias, o movimento não é interrompido (retornando logo que possível à trajectória correcta), mas isto significa que nem todas as especificações fornecidas pelo utilizador são sempre cumpridas.

Capítulo 5

Controlo Baseado nas Forças de Reacção

5.1 Introdução

No ser humano, a manutenção do equilíbrio e da orientação corporal é essencial para a execução de qualquer actividade da vida quotidiana e, principalmente, na prática de actividades desportivas. O equilíbrio é uma acção complexa que o humano realiza sem aparente dificuldades, mas que encerra um conjunto de características difíceis de replicar num robô. No caso de o robô adoptar uma postura estática ou realizar movimentos lentos é comum usar-se a projecção vertical do centro de gravidade (COG) como medida de estabilidade. Por sua vez, a análise da estabilidade dinâmica de padrões de locomoção bípede aparece associada a critérios como o *zero moment point* (ZMP) ou ao centro de pressão (COP). O centro de pressão é definido como o ponto de actuação da resultante das forças normais de reacção no solo, sendo possível obter uma estimativa dessa localização com base em sensores de força colocados na planta do pé.

O robô humanóide utilizado neste trabalho está dotado de dois tipos de sensores úteis na perspectiva do equilíbrio: células de carga e sensores inerciais. As quatro células de carga colocadas nas extremidades de cada pé permitem estimar com boa precisão as forças normais de reacção no solo e, desta forma, o centro de pressão. Por sua vez, os sensores inerciais visam emular o sistema vestibular humano e dotar o robô de mecanismos de inspiração biológica. O trabalho realizado centrou-se nos processos de calibração dos sensores, na determinação do COP e na implementação de um controlador baseado nas forças de interacção com o solo. Foram obtidos alguns resultados preliminares em que a perna de suporte (3-gdl) mantém o seu equilíbrio estático quando colocada sob uma superfície plana de inclinação variável. A parte final deste capítulo descreve os esforços no projecto, concepção e teste de uma nova placa de circuito impresso com alterações na unidade slave.

5.2 Células de carga

As células de carga utilizadas são o modelo LBS-5 da Interface com dimensões muito reduzidas (aproximadamente do tamanho de uma moeda de 5 cêntimos), como é possível constatar a partir da Figura 5.1. Para além do tamanho, estas células de carga tem a particularidade do seu ponto de contacto ser curvo, maximizando a sensibilidade. A Tabela 5.1 resume as principais características técnicas deste componente (Interface, “Advanced Force Measurement”), com destaque para sobrecarga permitida de 150% e o facto de incluir compensação de temperatura.



Figura 5.1: Células de carga miniatura de 5lbs usadas no robô humanóide.

Características		Valor
Altura	Mm	3.00
Diâmetro	Mm	9.60
Tensão de Alimentação	V	[5.0; 7.0]
Sensibilidade de Saída	mV/V	2.0
Capacidade	Kg,f	2.27
Sobrecarga permitida	%	150
Resistência	Ω	350
Tem compensação de temperatura		

Tabela 5.1: Características técnicas das células de carga LBS-5.

Os sensores integram uma ponte de Wheatstone com a resistência de entrada de 350Ω e quando alimentados a 5V apresentam uma sensibilidade de 2mV/V . Deste modo, torna-se necessário amplificar o sinal fornecido pelas células de carga tendo sido utilizados amplificadores de instrumentação (INA129), juntamente com potenciômetros para dimensionar o ganho de acordo com a gama pretendida. O modelo escolhido resultou de um compromisso entre uma boa resolução e a necessidade de não se excederem os limites de carga permitidos. Se o caminhar realizado pelo robô humanóide for semelhante ao dos humanos, existem duas fases do movimento em que apenas alguns dos sensores suportam o peso total do humanóide. Um primeiro momento em que o pé descola do solo e os dois sensores da frente do pé suportam todo o peso da estrutura e, uma segunda fase, em que o pé entra em contacto com o solo e os dois sensores de trás suportam todo o peso. Considerando estes períodos como as situações extremas (no caso de um salto provavelmente iriam ser exercidas forças ainda maiores), podemos assumir que cada um dos sensores tem que suportar pelo menos metade do peso total do robô que ronda os 6 kg.

Os sensores de força serão colocados nas extremidades do pé, como mostra a Figura 5.2, sendo integrados entre duas placas metálicas que constituem a sola do pé. A placa inferior tem pequenas aberturas com as dimensões exactas dos sensores para os fixar, enquanto um sistema em rosca aperta os sensores e une as duas placas. Esta última parte do processo de colocação dos sensores é complicada dado que as placas tem de estar bem apertadas para que os sensores nunca saiam da sua posição e para que o sistema de aperto nunca se desenrosque com o tempo (caso isso aconteça é preciso voltar a apertar e calibrar). Por outro lado, também não podem estar demasiado apertadas porque isso afectaria as leituras efectuadas, podendo mesmo levar à danificação dos sensores.

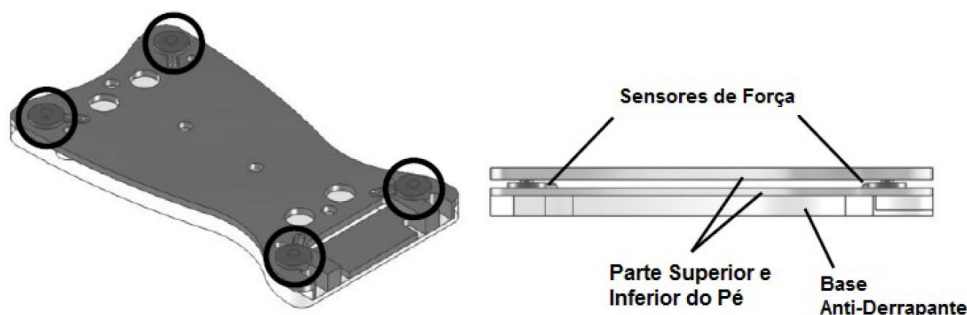


Figura 5.2: Localização das células de carga no pé. (Ribeiro, 2010)

O sistema em rosca que prende as duas placas faz com que exista uma certa resistência a variações mecânicas que provêm das forças exercidas pelo humanóide, diminuído a variação das forças lidas. Para tentar suavizar este problema, o ideal seria colocar os parafusos o mais afastado possível dos sensores, no centro do pé, de maneira a permitir que a placa metálica dobre ou oscile mais nas suas extremidades onde estão localizados os sensores.

Para além destes problemas, importa garantir que o aperto das roscas é homogéneo e que, periodicamente, se proceda a um processo de revisão de maneira a ter os quatro sensores correctamente calibrados. Nesta fase de testes, optou-se por não montar os sensores de força no pé, mas colocá-los numa superfície plana (*e.g.*, na mesa do laboratório) sobre os quais se coloca uma placa de acrílico que suportará as massas de teste ou a própria perna de suporte (Figura 5.3). Nesta medida, tanto a superfície onde são colocados os sensores como a placa de acrílico devem ser perfeitamente planas para evitar valores inconsistentes. Nos testes realizados, no decorrer deste projecto, foi possível observar que a placa de acrílico utilizada tinha uma pequena curvatura. Apesar de esta não ser visível a olho nu, analisando as forças lidas pelos sensores com uma massa centrada na placa, era possível ver que o peso não era distribuído igualmente pelos quatro sensores. Dois destes, situados nas extremidades opostas (na diagonal), estavam a suportar uma parte maior da massa. Como veremos mais à frente, esta situação não é um problema porque os cálculos que são realizados para obter o centro de pressão anulam este erro. Mesmo assim tentou-se minimizar esta diferença colocando uma ou duas camadas de fita-cola por baixo dos sensores que estavam a receber uma carga menor, de modo a obter os resultados pretendidos, ou seja, aproximadamente a mesma força nos quatro sensores.

Foram realizados alguns testes para avaliar o desempenho dos sensores de força usando massas conhecidas que eram, manualmente, colocadas sobre a placa de acrílico. A Figura 5.4 pretende mostrar o bom comportamento estático dos sensores, nomeadamente o facto de não existir histerese. Depois de uma colocação sucessiva de massas (descontinuidades visíveis no gráfico), verifica-se que os valores lidos regressam ao patamar anterior quando estas são retiradas. Apesar de as massas estarem bem centradas na placa de acrílico, percebe-se existir uma pequena deformação na placa que faz com que o valor de um dos sensores seja menor que todos os restantes. Por sua vez, a Figura 5.5 ilustra o comportamento dinâmico dos sensores quando ocorre a remoção de uma das massas. Os sensores apresentam uma resposta bastante rápida com um declive de aproximadamente 17.2mV/ms. As diferenças entre as forças de cada sensor, durante a remoção da massa, parecem menos importantes uma vez que este processo é efectuado manualmente.

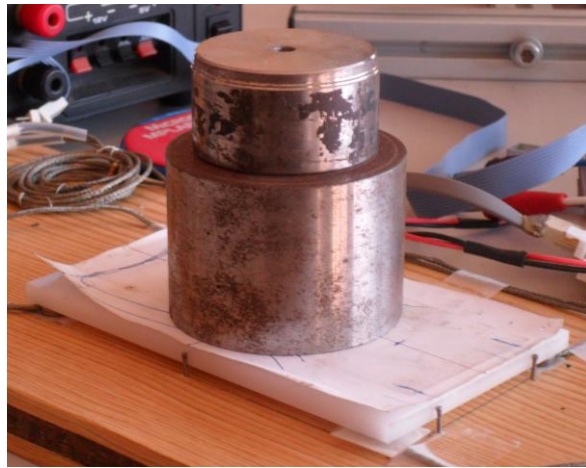


Figura 5.3: Testes efectuados com os quatro sensores de força colocados nas extremidades da placa de acrílico que suporta as duas massas.

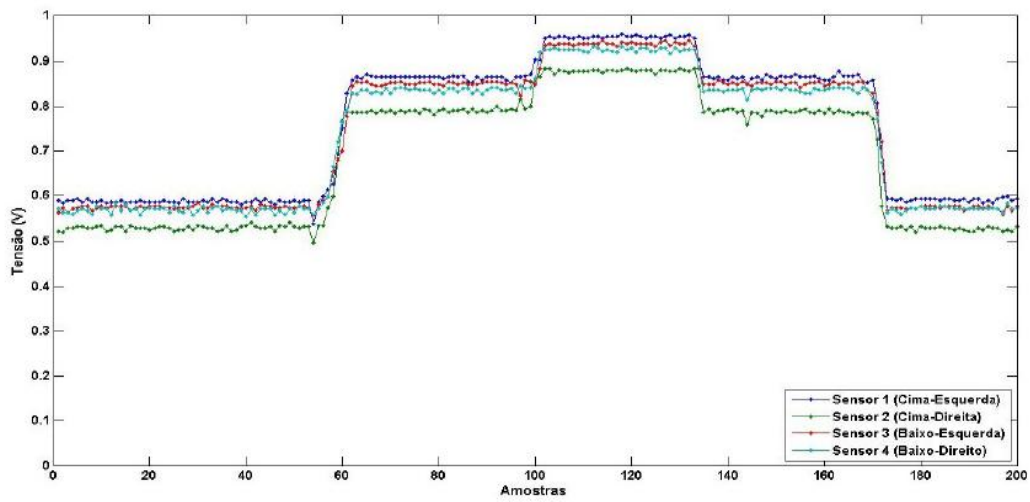


Figura 5.4: Evolução temporal das quatro forças normais de reacção (comportamento estático).

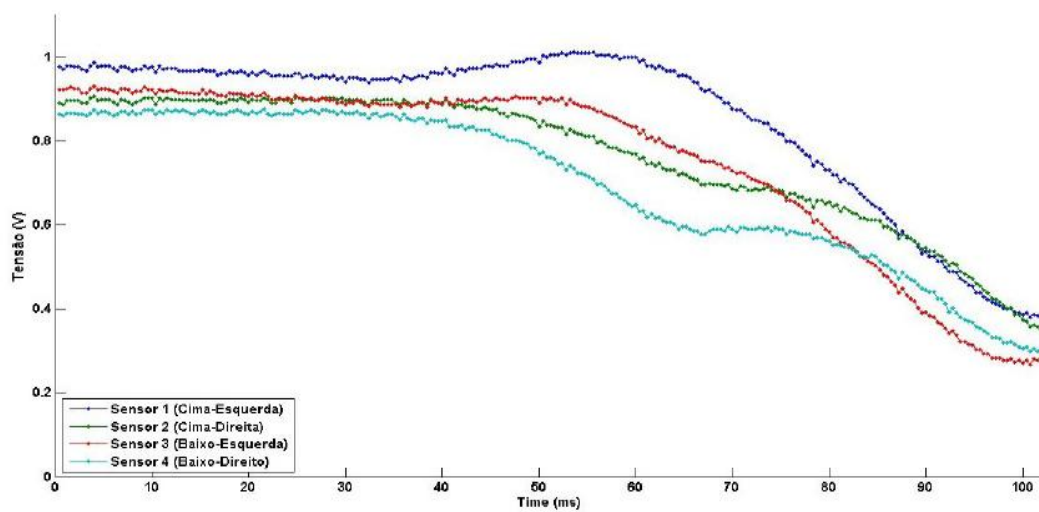


Figura 5.5: Evolução temporal das quatro forças normais de reacção (comportamento dinâmico).

5.3 Cálculo do centro de pressão

O centro de pressão pode ser calculado com base nas leituras sensoriais das forças normais de reacção e na disposição dos próprios sensores. A equação (5.1) representa o centro de pressão definido em relação a um sistema de coordenadas colocado no centro da base de acrílico:

$$\vec{COP} = \frac{\sum_i \vec{r}_i f_i}{\sum_i f_i} \quad (5.1)$$

em que f_i são os valores algébricos das forças e \vec{r}_i os vectores posição das origens. Na unidade slave, os valores obtidos pela ADC do microcontrolador são chamados de $valSensor[i]$, sendo i o respectivo sensor. Estes valores passam por um filtro que elimina as pequenas oscilações aleatórias e/ou o ruído (estes valores são chamados de $valSensorCalib[i]$). É com base neste vector que se calculam as coordenadas do COP em relação ao centro da placa segundo o eixo xx (d direcção do movimento) e o eixo yy (d direcção lateral):

(5.2)

Foram realizados alguns ensaios para testar a fiabilidade destas equações usando uma placa em acrílico rectangular com aproximadamente o tamanho do pé do humanóide de 140 por 80 mm. Desta forma, os sensores colocados nos vértices da parte inferior da placa ficam aproximadamente alinhados com as aberturas criadas na sola dos pés. O objectivo era visualizar em tempo real as variações no centro de pressão através da aplicação de diferentes forças em diferentes pontos da placa de acrílico (ver Figura 5.7 e Figura 5.7).

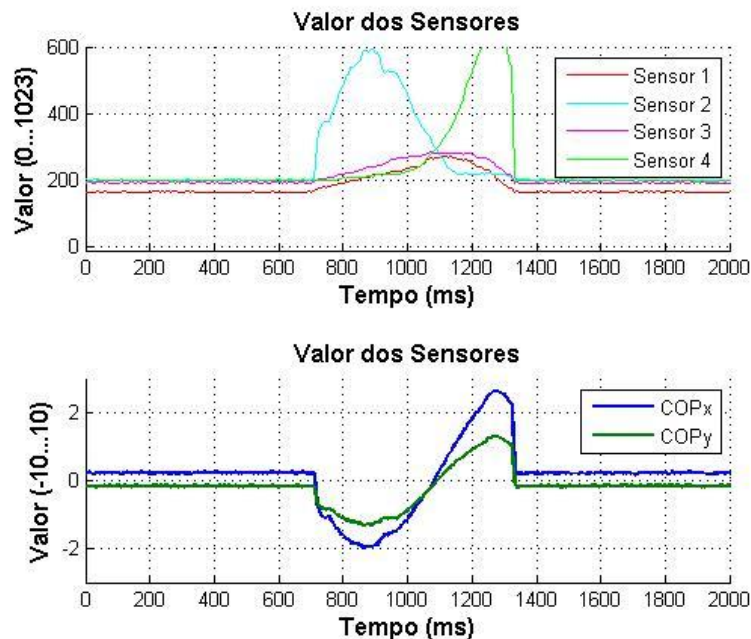


Figura 5.6: Evolução temporal das forças de reacção (em cima) e do centro de pressão (em baixo) em resultado de se premir com o dedo directamente na placa de acrílico.

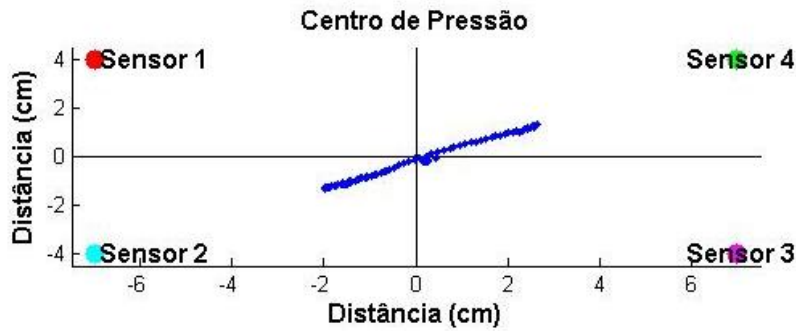


Figura 5.7: Representação da trajetória realizada pelo centro de pressão à medida que as forças de reacção variam.

5.4 Algoritmo de controlo baseado nas forças de reacção

As experiências descritas nesta subsecção resultam da colocação da perna de suporte (3-gdl) sobre a placa de acrílico que está em contacto directo com os sensores força. Este conjunto assenta sobre uma superfície plana cuja inclinação é alterada pela acção de um utilizador (Figura 5.8). O objectivo do controlador é assegurar o equilíbrio estático da perna em resposta a alterações da inclinação desta superfície. Por outras palavras, pretende-se mover as juntas do tornozelo de forma a manter-se a posição do COP no centro do pé. A lei de controlo utilizada consistiu em desacoplar os eixos considerando que as velocidades das juntas do tornozelo são directamente proporcionais ao erro do COP em cada direcção (a junta do joelho permanece fixa):

$$\begin{aligned} \dot{q}_t^{xx} &= k_p^{xx} \cdot (COP_x^{ref} - COP_x) \\ \dot{q}_t^{yy} &= k_p^{yy} \cdot (COP_y^{ref} - COP_y) \end{aligned} \quad (5.3)$$

em que a notação xx indica o eixo de variação do COP e os parâmetros k_p são os ganhos do termo proporcional. O COP de referência é obtido antes de ser executado o algoritmo, quando a perna se encontra em equilíbrio (COP^{ref} próximo de zero).

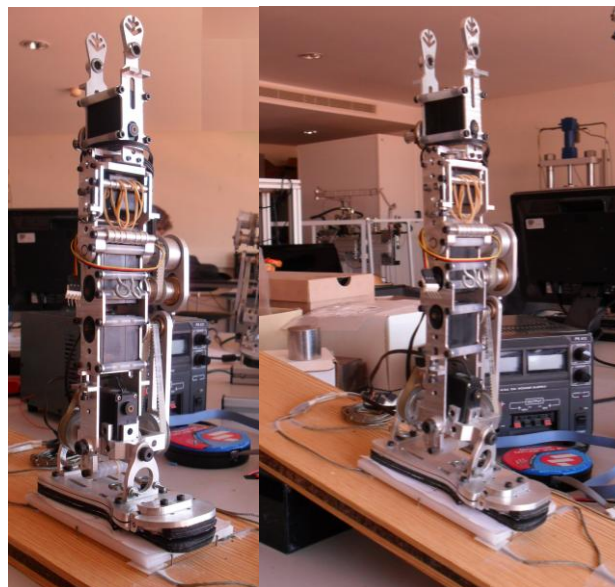


Figura 5.8: Testes efectuados numa superfície de inclinação variável com os quatro sensores de força colocados nas extremidades da placa de acrílico que por sua vez suporta uma perna.

O deslocamento do COP em relação à referência resulta da elevação da superfície segundo um dos eixos principais (eixo dos xx ou yy). Na primeira experiência a superfície móvel é elevada manualmente de forma a explorar variações do centro de pressão segundo o eixo dos xx. Os gráficos da Figura 5.9 permitem identificar o instante em que a superfície é elevada (cerca dos 2.5 s) e o instante em que volta à posição inicial (cerca dos 5.2 s). Com este algoritmo cada ciclo demora 43 ms a ser executado. Como é possível observar a partir deste conjunto de gráficos, o objectivo principal foi cumprido com o COP próximo dos valores de referência. Seria possível obter um erro menor com este controlador, mas à custa de movimentos mais bruscos que facilmente criavam instabilidade. O sistema reage rapidamente aos estímulos produzidos pelo utilizador humano e assegura o equilíbrio da perna de forma eficaz.

Na segunda experiência exploram-se as compensações a introduzir ao explorar variações no centro de pressão segundo o eixo dos yy. Os resultados estão ilustrados na Figura 5.10 que segue a mesma ordem de apresentação dos gráficos. Mais uma vez, o COP real desloca-se ligeiramente em relação ao valor de referência ainda que os erros sejam ligeiramente superiores, em parte, devido ao facto de se ter reduzido o valor do ganho. Apesar dos resultados obtidos serem satisfatórios, existem alguns aspectos que não foram integralmente esclarecidos e merecem um trabalho mais exaustivo no futuro. Em primeiro lugar, o desempenho do controlador de força deve ser analisado em função das duas malhas de realimentação: a malha externa fechada em torno da unidade slave e o controlador interno dos servomotores. Ao contrário do Capítulo 4 em que as velocidades angulares de referência resultavam de um algoritmo de cinemática inversa em malha aberta, agora as velocidades angulares derivam de uma lei de controlo proporcional. Em segundo lugar, o algoritmo actual baseado no desacoplamento entre eixos deve ser substituído por uma lei de controlo baseada na matriz do Jacobiano, tal como descrito em (Ruas, 2008).

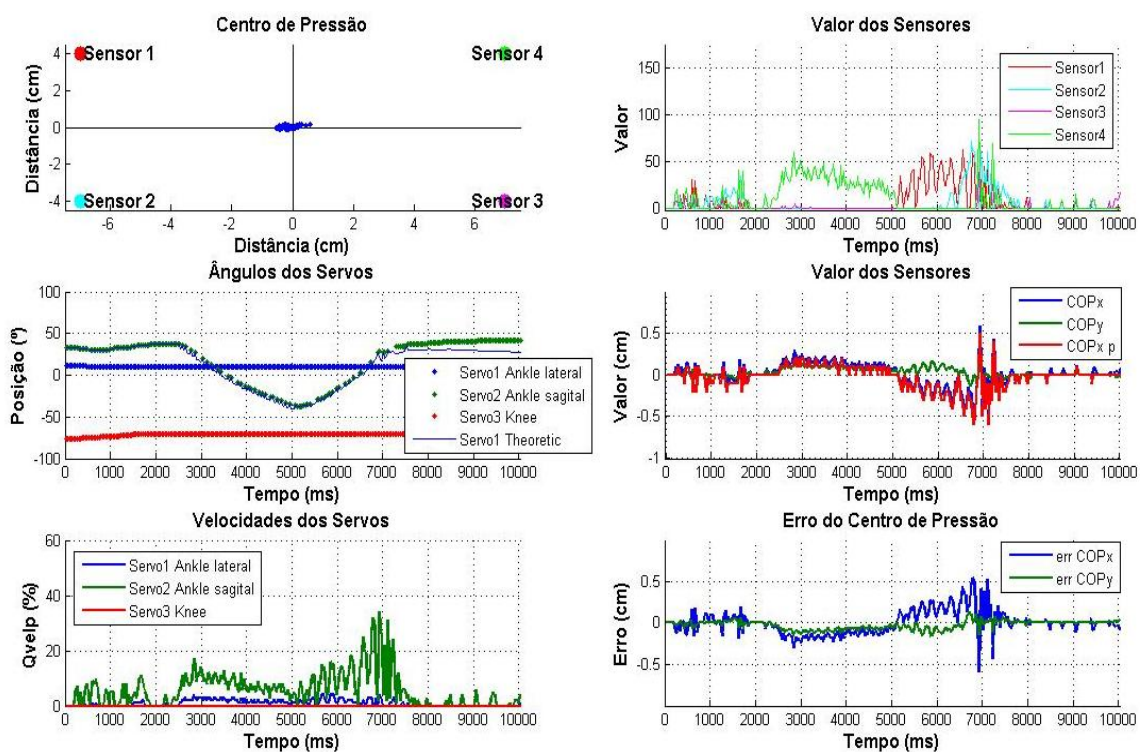


Figura 5.9: Inclinação da superfície móvel segundo o eixo dos yy: os gráficos da coluna à esquerda representam a evolução do centro de pressão debaixo do pé e a evolução temporal das posições e velocidades angulares; os gráficos da coluna à direita mostram a evolução temporal das forças em

cada extremidade do pé, a evolução temporal da localização do centro de pressão segundo os eixos dos xx e yy e os sinais de erro usados pelo controlador.

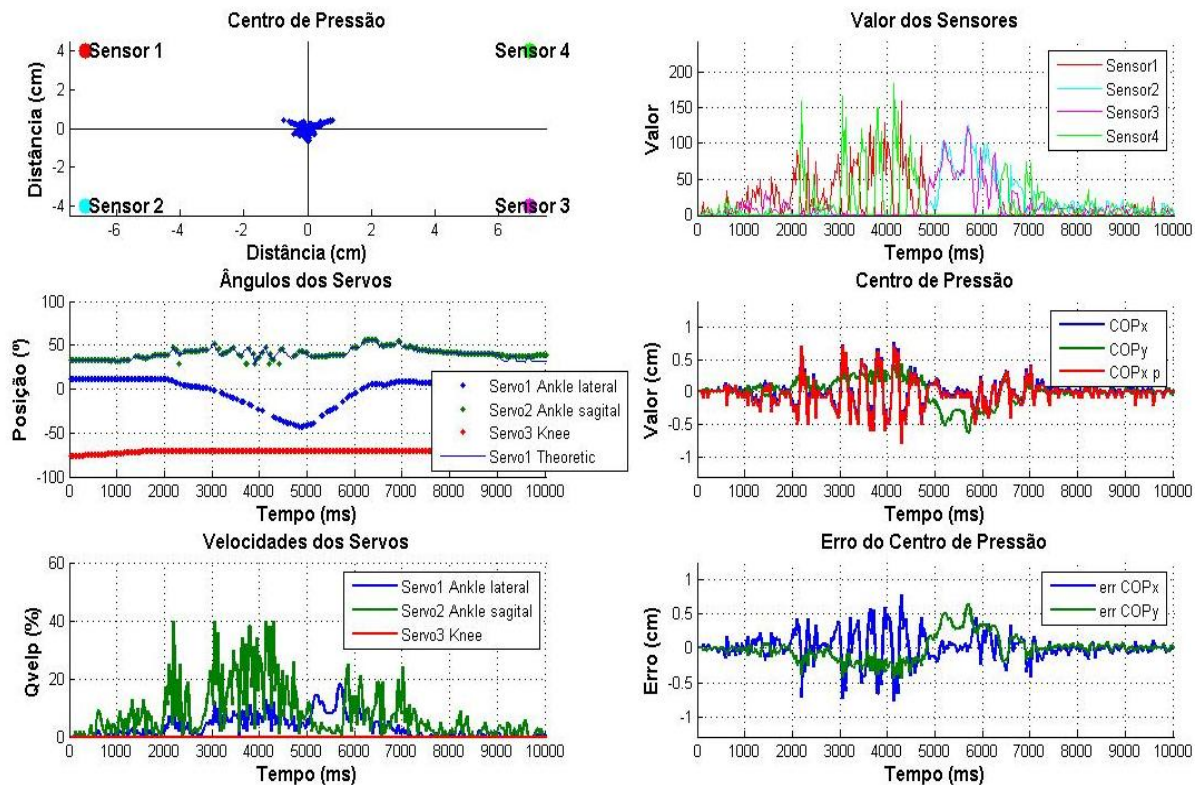


Figura 5.10: Inclinação da superfície móvel segundo o eixo dos xx: os gráficos da coluna à esquerda representam a evolução do centro de pressão debaixo do pé e a evolução temporal das posições e velocidades angulares; os gráficos da coluna à direita mostram a evolução temporal das forças em cada extremidade do pé, a evolução temporal da localização do centro de pressão segundo os eixos dos xx e yy e os sinais de erro usados pelo controlador.

5.5 Alterações introduzidas na unidade slave

O hardware de testes usado ao longo deste trabalho era constituído pela unidade slave concebida em placa de circuito impresso e um conjunto de componentes em placa branca, incluindo os circuitos de alimentação, os circuitos de amplificação dos sensores de força e os circuitos de comunicação com os servomotores (Ribeiro, 2010). Os circuitos de alimentação contêm reguladores de tensão que disponibilizam três tensões de alimentação: 3.3V para a placa de circuito impresso, 5V para os sensores de força e 7.4V para alimentação dos servomotores. Um dos objectivos do trabalho era projectar e conceber uma nova unidade slave capaz de integrar na placa de circuito impresso o sistema de amplificação dos sinais vindos dos sensores de força. O sistema de aquisição é constituído por um amplificador de instrumentação com quatro canais e por um potenciômetro digital também com quatro canais. A placa foi concebida como circuito impresso porque o microcontrolador escolhido apenas existia no package TQFP, sendo novamente desenhada no software grátis Eagle (CadSoft, “Eagle”). A placa de circuito impresso redesenhada é constituída por:

- 1 microcontrolador – dsPIC33FJ128GP706;
- 1 transceiver para os dois módulos UART – MAX3223;
- 2 transceivers para as duas redes ECAN – SN65HVD232;

- 1 amplificador instrumental – ISL28470;
- 1 potenciômetro digital – MCP4352;
- conectores para outras interfaces.

Para redesenhar a placa foi utilizada a mesma biblioteca que o Miguel Ribeiro usou (Ribeiro, 2010) que já incluía alguns destes elementos disponíveis (*e.g.*, o MCP4352). No caso do ISL28470 foi necessário desenhar o símbolo para o esquema do circuito e o package para a board. Posteriormente, foram introduzidos no esquema do circuito estes novos componentes (amplificador, potenciômetro e resistências para regular ganhos) antes de se avançar para o desenho da board. Optou-se por uma organização manual dos componentes (a opção automática requer licença) que, apesar de muito mais trabalhosa, tem a vantagem de permitir dispor os componentes de uma maneira lógica por forma a facilitar as ligações e, dessa maneira, um melhor aproveitamento do espaço para se obter uma placa de dimensões reduzidas. Para além disso, o processo automático podia originar situações com dificuldade acrescida na montagem manual.

O resultado final é apresentado na Figura 5.11 e Figura 5.12. Apesar de não se ter validado integralmente o funcionamento desta placa, esta foi usada durante alguns meses, enquanto eram testados os algoritmos de controlo em velocidade. A única parte que não foi testada está relacionada com os sensores de força, nomeadamente, a programação necessária para configurar os potenciômetros digitais.

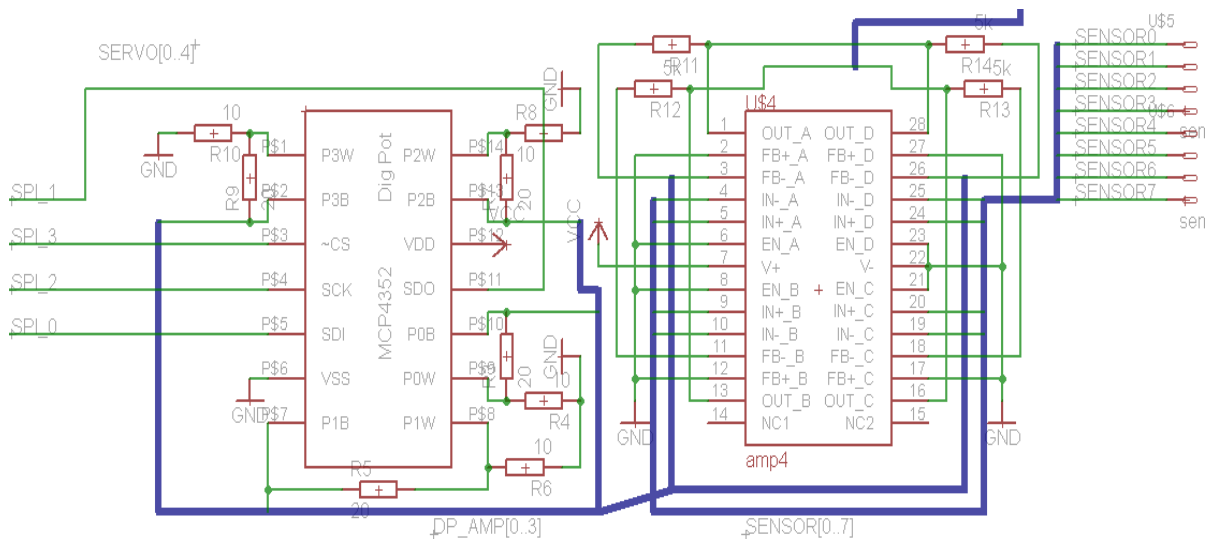


Figura 5.11: Disposição dos dois componentes que foram adicionados ao esquema do circuito da unidade slave (o MCP4352, o ISL28470 e resistências).

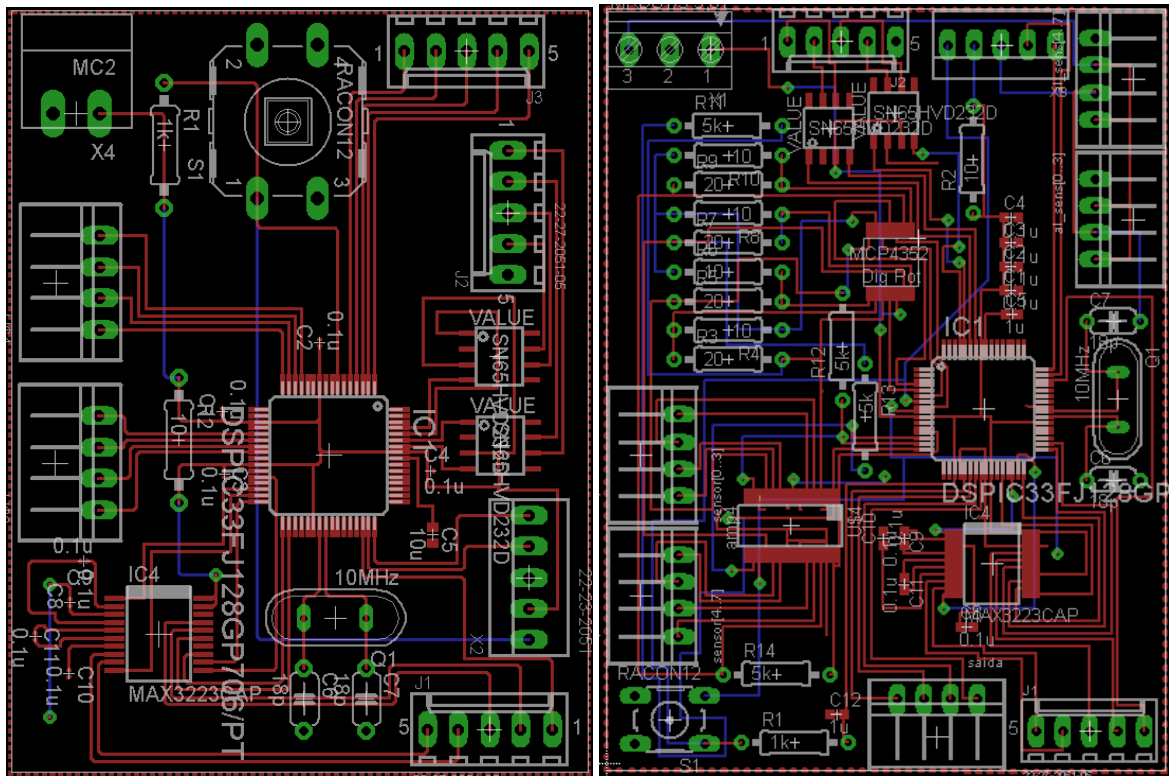


Figura 5.12: Desenho final da unidade slave em Eagle: board antiga (esquerda) e board nova (direita).

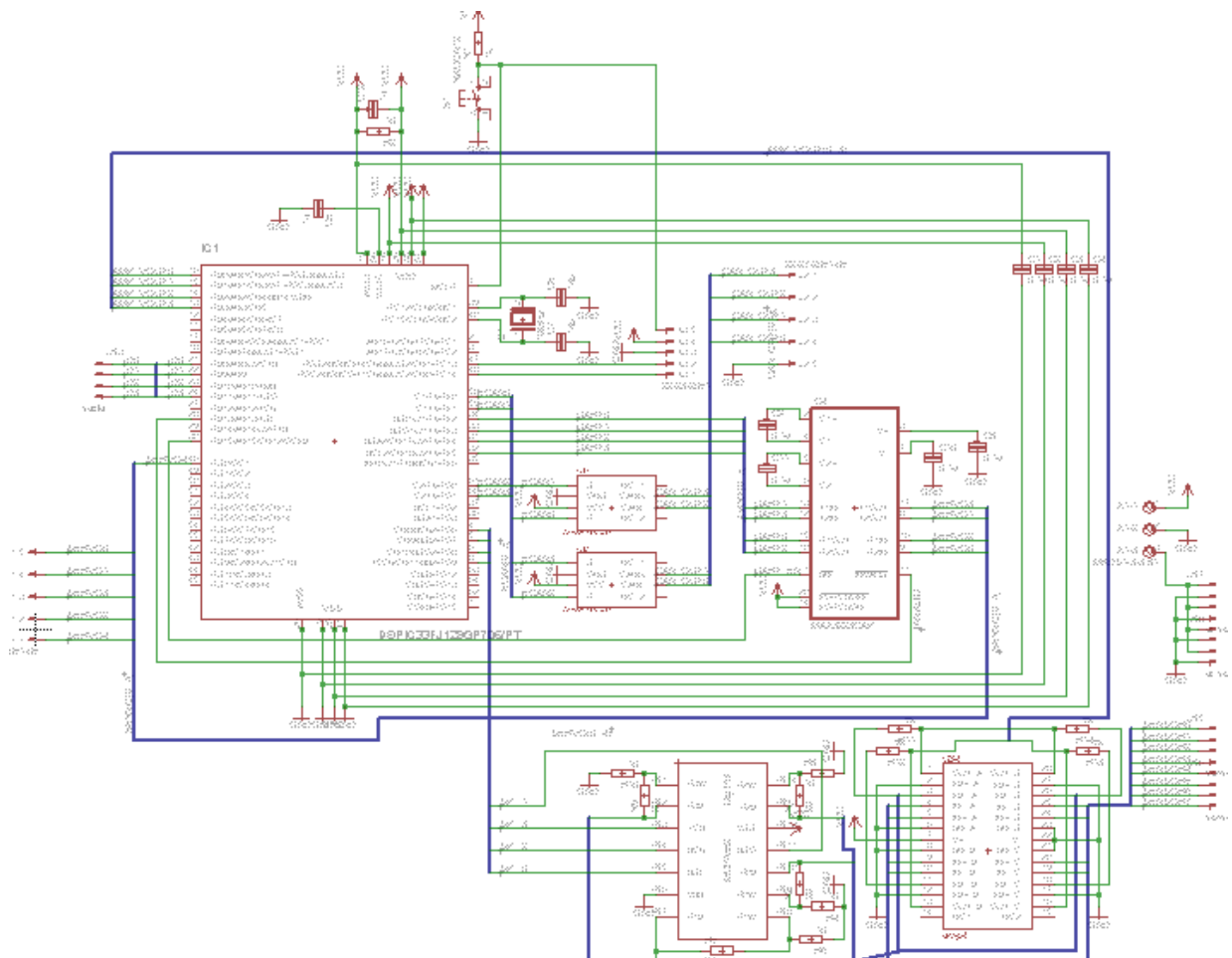


Figura 5.13: Esquema do circuito da nova placa

Capítulo 6

Conclusões

6.1 Discussão dos resultados

Este trabalho de dissertação permitiu introduzir alguns avanços que estão, sobretudo, relacionados com os algoritmos de planeamento e controlo do movimento, nomeadamente com a implementação e teste dos movimentos ponto-a-ponto e dos movimentos contínuos. As diversas experiências realizadas ao longo do trabalho permitiram confirmar, por um lado, o bom desempenho das unidades microcontroladoras. Atendendo ao peso computacional dos algoritmos implementados, foram obtidos tempos de ciclo a rondar os 20 ms com uma unidade slave a comandar em simultâneo o movimento de três servomotores. Nessa medida, uma vez implementada a arquitectura distribuída de controlo, o sistema ficará dotado de um poder de cálculo assinalável tendo em vista um controlo híbrido capaz de combinar capacidades reactivas e deliberativas. Por outro lado, foram implementados com sucesso algoritmos de controlo em velocidade necessários à implementação dos movimentos contínuos e à implementação da malha de realimentação de força.

Ao longo do trabalho não foi valorizada uma avaliação quantitativa do desempenho do sistema do ponto de vista do seguimento das trajectórias desejadas nem da capacidade do elemento terminal em causa atingir a posição desejada no instante pré-definido. Para além das dificuldades que resultam do processo de leitura das posições angulares de cada servomotor, existem os problemas associados ao facto da especificação dos movimentos ser feita no espaço Cartesiano, enquanto o controlo se faz ao nível das juntas. Os resultados obtidos são, de qualquer forma, muito satisfatórios tendo em vista a realização de movimentos mais complexos envolvendo as diferentes sub-estruturas estudadas, incluindo a própria marcha. Na mesma linha de ideias, este trabalho permitiu perceber a dificuldade associada à programação de um robô humanóide com tantos graus de liberdade capaz de gerar acções coordenadas na realização de uma dada tarefa. Sendo certo que as ajudas computacionais desenvolvidas ao longo do trabalho são úteis na tentativa de avançar para a definição de um padrão de locomoção, parece cada vez mais necessário avançar para novos conceitos de programação e controlo baseados em aprendizagem.

Em relação às células de carga, o estudo efectuado mostra claramente que estes componentes são uma mais valia para o planeamento e o controlo de movimentos suaves e fisicamente realizáveis. Os resultados preliminares obtidos a partir da leitura dos sensores de força indicam que a determinação da localização do centro de pressão e a avaliação da distribuição de forças na planta dos pés podem ser realizadas com boa qualidade. Desta forma é possível antever, desde já, que a implementação de algoritmos de controlo com realimentação de força serão tópicos a explorar no futuro. Finalmente, uma referência para o desempenho mecânico da estrutura humanóide cujo exoesqueleto mostrou ser robusto ao longo das múltiplas experiências.

Contudo, o uso das correias de transmissão actualmente integradas deve ser reavaliado, pois verificaram-se fenómenos de folgas quando a estrutura foi sujeita a maiores esforços (*e.g.*, a perna de suporte).

6.2 Perspectivas de trabalho futuro

De certa forma, a discussão de resultados já deixou em aberto algumas das inúmeras possibilidades de trabalho futuro no que toca ao sistema de controlo de baixo nível. Aqui, destacam-se alguns dos pontos mais importantes:

- Efectuar testes exaustivos com a perna livre nas fase de elevação (impacto) do (com o) solo com e sem elásticos usando a possibilidade de *release* dos servomotores (função que liberta os servos, ou seja, pára o mecanismo interno de controlo de posição);
- Realizar testes com os sensores de força inseridos nas solas dos pés e avançar para a implementação de algoritmos de controlo baseados nas forças de reacção usando a matriz do Jacobiano;
- Integrar o software desenvolvido para os braços e pernas no sentido de planear o movimento completo do robô conducente à realização de várias tarefas, tais como andar, rodar sobre si, chutar uma bola, etc;
- Terminar a validação da nova placa de teste de circuito impresso e efectuar a programação dos potenciómetros digitais e a comunicação com o microprocessador;
- Implementar a arquitectura de controlo distribuido substituindo o PC actual por uma unidade de processamento baseada em PC-104 e a actual unidade slave por múltiplas unidades distribuídas ao longo de toda a estrutura e comunicando por barramento ECAN.

Apêndice A – Descrição de Funções Relevantes

Software 2 Arms		
Função	Parâmetros de Entrada	Parâmetros de Saída
invKin_4dof ()	xyz [] – coordenadas cartesianas da mão xyz[0] – x xyz[1] – y xyz[2] – z warm – qual dos braços L ou R	qf [] – ângulos dos servomotores qf[0] – junta sagital do ombro qf[1] – junta lateral do ombro qf[2] – junta vertical do braço qf[3] – junta do cotovelo
dirKin_4dof ()	q [] – ângulos dos servomotores q[0] – junta sagital do ombro q[1] – junta lateral do ombro q[2] – junta vertical do braço q[3] – junta do cotovelo warm – qual dos braços L ou R	P [] – coordenadas cartesianas da mão P[0] – x P[1] – y P[2] – z
CartTrajPlanning ()	t – tempo actual tf – tempo de execução XYZ_i [] – coordenadas cartesianas da posição inicial da mão XYZ_i[0] – x XYZ_i[1] – y XYZ_i[2] – z XYZ_f [] – coordenadas cartesianas da posição final da mão XYZ_f[0] – x XYZ_f[1] – y XYZ_f[2] – z N – ordem polinomial	Pee [] – posição da mão no instante actual Pee[0] – x Pee[1] – y Pee[2] – z dPee [] – velocidade da mão no instante actual

Unidades: coordenadas cartesianas – mm; tempos – ms; ângulos – graus

Tabela 6.1: Parâmetros das funções de cinemática e planeamento de trajectória para os braços

Software Free Leg		
Função	Parâmetros de Entrada	Parâmetros de Saída
invKin_6dof ()	xyz [] – coordenadas cartesianas do centro do pé xyz[0] – x xyz[1] – y xyz[2] – z	qf [] – ângulos dos servomotores qf[0] – junta sagital da anca qf[1] – junta lateral da anca qf[2] – junta vertical da anca qf[3] – junta do joelho qf[4] – junta sagital do tornozelo qf[6] – junta lateral do tornozelo
dirKin_6dof ()	q [] – ângulos dos servomotores	P [] – coordenadas cartesianas do

	$q[0]$ – junta sagital da anca $q[1]$ – junta lateral da anca $q[2]$ – junta vertical da anca $q[3]$ – junta do joelho $q[4]$ – junta sagital do tornozelo $q[6]$ – junta lateral do tornozelo	centro do pé $P[0]$ – x $P[1]$ – y $P[2]$ – z
CartTrajPlanning ()	t – tempo actual tf – tempo de execução XYZ_i [] – coordenadas cartesianas da posição inicial do centro do pé $XYZ_i[0]$ – x $XYZ_i[1]$ – y $XYZ_i[2]$ – z XYZ_f [] – coordenadas cartesianas da posição final do centro do pé $XYZ_f[0]$ – x $XYZ_f[1]$ – y $XYZ_f[2]$ – z N – ordem polinomial	Pee [] – posição do centro do pé no instante actual $Pee[0]$ – x $Pee[1]$ – y $Pee[2]$ – z dPee [] – velocidade do centro do pé no instante actual

Unidades: coordenadas cartesianas – mm; tempos – ms; ângulos – graus

Tabela 6.2: Parâmetros das funções de cinemática e planeamento de trajectória para a perna livre

Software Support Leg		
Função	Parâmetros de Entrada	Parâmetros de Saída
invKin_3dof ()	xyz [] – coordenadas cartesianas do centro da anca $xyz[0]$ – x $xyz[1]$ – y $xyz[2]$ – z	qf [] – ângulos dos servomotores $qf[0]$ – junta lateral do tornozelo $qf[1]$ – junta sagital do tornozelo $qf[2]$ – junta do joelho
dirKin_3dof ()	q [] – ângulos dos servomotores $q[0]$ – junta lateral do tornozelo $q[1]$ – junta sagital do tornozelo $q[2]$ – junta do joelho	P [] – coordenadas cartesianas do centro da anca $P[0]$ – x $P[1]$ – y $P[2]$ – z
CartTrajPlanning ()	t – tempo actual tf – tempo de execução XYZ_i [] – coordenadas cartesianas da posição inicial do centro da anca $XYZ_i[0]$ – x $XYZ_i[1]$ – y $XYZ_i[2]$ – z XYZ_f [] – coordenadas cartesianas da posição final do centro da anca $XYZ_f[0]$ – x $XYZ_f[1]$ – y	Pee [] – posição do centro da anca no instante actual $Pee[0]$ – x $Pee[1]$ – y $Pee[2]$ – z dPee [] – velocidade do centro da anca no instante actual

	$XYZ_f[2] - z$ N – ordem polinomial	
Unidades: coordenadas cartesianas – mm; tempos – ms; ângulos – graus		

Tabela 6.3: Parâmetros das funções de cinemática e planeamento de trajetória para a perna de suporte

Apêndice B – Limites e Conversões Angulares

Limites angulares de cada servo (importantes para os cálculos realizados nas cinemáticas)

Todos os limites angulares são impostos pelos limites mecânicos do nosso humanóide. O objectivo do Rémi Sabino foi tornar o humanóide fisicamente o mais parecido possível com um humano (Sabino, 2010), ou seja, isso inclui as limitações físicas de um humano. Por exemplo o cotovelo, joelho, tornozelo ou coluna são zonas que apresentam limitações características de um humano e também estas estão presentes no nosso humanóide.

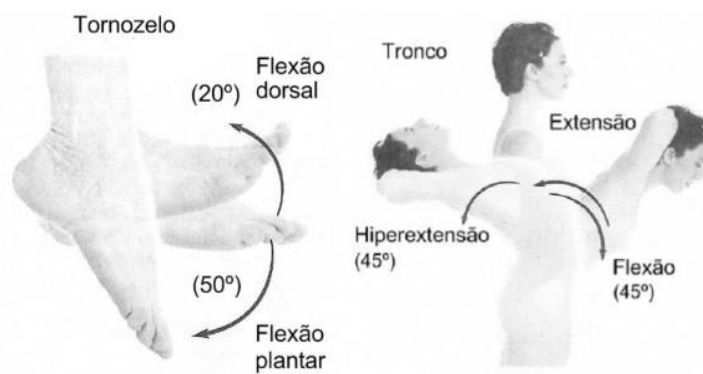


Figura 6.1: Movimentos da articulação do tornozelo e coluna. (Sabino, 2010)

Junta	Limitação angular Lado - Esquerdo/Direito	Junta	Limitação angular Lado - Esquerdo/Direito
Cabeça sagital	Humano: $-90^\circ \leftrightarrow 45^\circ$ $-90^\circ \leftrightarrow 45^\circ$	Cabeça lateral	Humano: $-90^\circ \leftrightarrow 90^\circ$ $-90^\circ \leftrightarrow 90^\circ$
Ombro sagital	Humano: $-45^\circ \leftrightarrow 180^\circ$ $-90^\circ \leftrightarrow 90^\circ / 90^\circ \leftrightarrow -90^\circ$	Anca sagital	Humano: $-30^\circ \leftrightarrow 120^\circ$ $-74^\circ \leftrightarrow 40^\circ / 40^\circ \leftrightarrow -74^\circ$
Ombro lateral	Humano: $-180^\circ \leftrightarrow 0^\circ$ $90^\circ \leftrightarrow -90^\circ / -90^\circ \leftrightarrow 90^\circ$	Anca lateral	Humano: $-40^\circ \leftrightarrow 45^\circ$ $90^\circ \leftrightarrow -90^\circ / -90^\circ \leftrightarrow 90^\circ$
Ombro vertical	Humano: $-90^\circ \leftrightarrow 45^\circ$ $90^\circ \leftrightarrow -45^\circ / -90^\circ \leftrightarrow 45^\circ$	Anca vertical	Humano: $-45^\circ \leftrightarrow 45^\circ$ $45^\circ \leftrightarrow -45^\circ / -45^\circ \leftrightarrow 45^\circ$
Cotovelo	Humano: $0^\circ \leftrightarrow 160^\circ$ $85^\circ \leftrightarrow -75^\circ / -85^\circ \leftrightarrow 75^\circ$	Joelho	Humano: $-130^\circ \leftrightarrow 0^\circ$ $90^\circ \leftrightarrow -90^\circ / -90^\circ \leftrightarrow 80^\circ$
Tronco sagital	Humano: $-45^\circ \leftrightarrow 45^\circ$ $-45^\circ \leftrightarrow 45^\circ$	Tornozelo sagital	Humano: $-50^\circ \leftrightarrow 20^\circ$ $90^\circ \leftrightarrow -90^\circ / 90^\circ \leftrightarrow 90^\circ$
Tronco lateral	Humano: $-45^\circ \leftrightarrow 45^\circ$ $-45^\circ \leftrightarrow 45^\circ$	Tornozelo lateral	Humano: $-45^\circ \leftrightarrow 30^\circ$ $90^\circ \leftrightarrow -90^\circ / -90^\circ \leftrightarrow 90^\circ$

Tronco vertical	Humano: $-90^\circ \leftrightarrow 90^\circ$ $-90^\circ \leftrightarrow 90^\circ$	-	-
------------------------	--	---	---

Tabela 6.4: Limites angulares das articulações do corpo humano e respectivos ângulos de cada servomotor.

Validação dos ângulos

Estas limitações são impostas logo a partida, no Matlab, onde a introdução de um ângulo só é aceite se este valor se encontrar dentro dos limites angulares do servo que se pretende usar. Quando estamos a tentar executar algum movimento utilizando coordenadas Cartesianas, esta verificação com os limites angulares apenas pode ser realizada depois de convertida a posição das coordenadas cartesianas para ângulos. Esta conversão é feita, na unidade slave, através da cinemática inversa e é dentro desta, depois de obtidos os ângulos para cada um dos servos, que a comparação com os limites angulares deve ser feita. O mesmo deve ser feito com a cinemática directa, mas para este caso, o comparador deve ficar logo no início da cinemática porque já temos os valores angulares com parâmetros de entradas.

Conversão dos ângulos das juntas para ângulos dos servos (na cinemática inversa)

Quando é utilizada a função $\text{invKin}()$ de cinemática inversa temos como valores de entrada as posições cartesianas e valores de saída os ângulos de cada servomotor pertencentes ao membro. No interior desta função os cálculos são todos realizados em radianos e próximo do fim os valores são convertidos para ângulos: $qf[i] = q[i] \times 180/\pi$; com o $q[i]$ em radianos e o $qf[i]$ em graus.

Neste momento, os ângulos representam os ângulos das juntas mas é preciso converter estes para ângulos dos servos. Esta transformação tem três etapas: a primeira tem a ver com ganhos devido às transmissões, de seguida é preciso saber se o deslocamento angular da junta tem o mesmo sentido ao do servomotor, caso não tenha, é preciso inverter o ângulo. No final é feito um ajuste aos ângulos para os calibrar, uns devido a colocação do motor num dos limites angulares 90 ou -90 (como é o caso do joelho), e outros devido aos motores não estarem bem centrados (podemos chamar a isto uma calibração digital, mas no futuro isto tem de ser ajustado mecanicamente).

Ganho

Como já foi dito os ganhos são aplicados quando existem transmissões, ou seja, estas multiplicações dos ângulos das juntas com os ganhos de cada transmissão são apenas realizadas nas cinemáticas das pernas (livre e suporte) porque nos braços não existem transmissões. Estes ganhos variam consoante o diâmetro das polias construídas, sendo assim possível a regulação do comportamento do sistema de transmissão. Na seguinte tabela estão presentes os ganhos das juntas de uma perna porque como as pernas são simétricas as transmissões são iguais nos dois lados.

Junta		Ganho (mm/mm)
Anca sagital	Gain0	30/22
Anca lateral	Gain1	32/8

Anca vertical	Gain2	1
Joelho	Gain3	30/22
Tornozelo sagital	Gain4	32/8
Tornozelo lateral	Gain5	32/8

Tabela 6.5: Ganhos das transmissões de uma perna

Sentido

As funções de cinemáticas foram criadas primeiro no Matlab para simulação e posteriormente traduzidas para linguagem C. Como não existiram alterações, o sentido do deslocamento angular de cada junta criada no simulador pode não corresponder ao sentido da junta do servomotor correspondente na realidade. Foi então necessário encontrar quais as juntas que têm o sentido oposto às do simulador. Tendo o conhecimento destas, o processo é simples, basta apenas trocar o sinal das juntas que têm sentido oposto.

Servos com inversão	Id
Braço Esquerdo (Arms)	
Cotovelo	6
Braço Direito (Arms)	
Cotovelo	9
Perna Direita (Free Leg)	
Anca sagital	3
Anca lateral	1
Tornozelo sagital	4
Perna Esquerda (Support Leg)	
Tornozelo lateral	1
Joelho	3

Tabela 6.6: Juntas que têm o sentido oposto às do simulador (com respectivos Id's).

Calibração

Algumas das juntas necessitam de uma calibração, como é o caso do cotovelo ou do joelho que na realidade depois de esticados apenas podem dobrar para um dos lados e têm para além desta característica a possibilidade de fazer um grande deslocamento de valor superior a 90°, um deslocamento de aproximadamente 160° no cotovelo e de 130° no joelho.



Figura 6.2: Movimentos da articulação do joelho. (Sabino, 2010)

A única maneira de isto ser concebível utilizando estes motores digitais é de coloca-los num dos limites angulares (90 ou -90 graus dependendo do lado) quando o membro se encontrar completamente esticado, disponibilizando assim um deslocamento de 180°, todo para um lado (com a transmissão o deslocamento de 180° no servo passam para 120° na junta). Nos cálculos realizados com a cinemática do simulador, estas juntas do joelho e cotovelo têm a possibilidade de realizar um movimento dos 0° até aos 180° e como tal, quando o membro se encontra nos 0° ele está completamente esticado, por isso é preciso fazer este ajuste, neste caso, de -90° para obtermos o valor real do servomotor com a possibilidade de se deslocar de -90 a 90 graus.

Analisando a tabela seguinte este tipo de ajustes são realizados nas juntas do ombro sagital e lateral, do cotovelo e do joelho para ambos os lados esquerdo e direito.

Junta	Id	Ângulo de ajuste
Arms		Braço esquerdo
Ombro sagital	8	-45
Ombro lateral	7	-85
Ombro vertical (não é usado)	0	0
Cotovelo	6	+85
		Braço direito
Ombro sagital	11	+45
Ombro lateral	10	+85
Ombro vertical (não é usado)	0	0
Cotovelo	9	-90
Free Leg		Perna direita
Anca sagital	3	0
Anca lateral	1	-10
Anca vertical (não é usado)	0	0
Joelho	5	+80
Tornozelo sagital	4	-10
Tornozelo lateral	2	0
Support Leg		Perna esquerda
Tornozelo lateral	1	0
Tornozelo sagital	2	+20
Joelho	3	-90

Tabela 6.7: Ajustes dos ângulos das juntas por forma a obter os ângulos dos servomotores

Para além dos ajustes mencionados anteriormente ainda são necessários mais alguns, pequenos mas que não deixam de ser relevantes. Infelizmente nem todos os motores se encontram perfeitamente centrados mas esta pequena calibração digital coloca-os todos alinhados.

O cotovelo do braço direito deveria estar nos -85° com o braço esticado, como tal foi feito um acréscimo de -5° para ficar alinhado. A anca lateral e o tornozelo sagital da perna direita deviam estar alinhados nos 0° mas para isso foi preciso fazer um acréscimo de -10°. O mesmo acontece com o joelho da perna direita que devia estar alinhado nos -90° e precisou de um acréscimo de -10°. O tornozelo sagital da perna esquerda que se apresentava com um grande desvio teve de levar com um acréscimo de +20° para ficar alinhado.

Conversão dos ângulos dos servos para ângulos das juntas (na cinemática directa)

Todos estes processos há pouco explicados vão no sentido da conversão de ângulos de juntas para ângulos dos servos que é realizado na cinemática inversa. Quando utilizamos a cinemática directa, esta conversão vai no sentido oposto de ângulos de servos para juntas. Portanto a ordem dos processos é invertida: calibração, sentido e ganho.

Na calibração os valores dos ajustes angulares que eram a somar passam a subtrair e os que estavam a subtrair passam a somar. Nas alterações de sentido, as juntas que tinham o sentido diferente das juntas do simulador continuam a ter de ser invertidas, sendo assim realizado uma troca de sinal a estas. Na zona dos ganhos, todos os ângulos, que eram na cinemática inversa multiplicados pelos ganhos respectivos passam aqui a ser divididos pelos ganhos.

Depois destes três processos, os valores representam agora os ângulos das juntas e estes são de seguida convertidos de graus para radianos, multiplicando os valores angulares por $(\pi/180)$, para continuar com os cálculos da cinemática.

Limites das posições cartesianas

Cada um dos membros tem limites físicos, dependendo do tamanho deste, que torna possível apenas alguns movimentos. Quando introduzimos as coordenadas Cartesianas para onde queremos que o membro se mova, estas devem estar dentro destes limites. Se tal não for verdade, a cinemática inversa utilizada, antes de ser iniciado o movimento, apresenta uma mensagem de erro mencionado que as coordenadas introduzidas estão fora do alcance, tornando o $isVal$ diferente de 1, o que faz com que o movimento não seja executado.

	Braço Esquerdo	Braço Direito	Perna Livre	Perna de suporte
X (mm)	-234 ↔ 234	-234 ↔ 234	-56 ↔ 185	-165 ↔ 21
Y (mm)	140 ↔ 374	-374 ↔ 140	-95 ↔ 95	-109 ↔ 109
Z (mm)	-84 ↔ 384	-84 ↔ 384	-25 ↔ 60	66 ↔ 333

Tabela 6.8: Limites das posições Cartesianas para a função de cada membro

É preciso prestar atenção ao facto que os limites colocados nesta tabela têm a origem do sistema cartesiano em diferentes pontos do humanóide para cada membro. No caso dos braços esse ponto é o mesmo e situa-se na cintura do humanóide. Isto faz com que as coordenadas em Y sejam simétricas como é possível ver. No caso da perna de suporte a origem situa-se na superfície do chão (com a perna esticada), no centro do pé que está a ser utilizado, o pé esquerdo. No caso da perna livre a origem situa-se num plano imaginário paralelo ao chão, no centro do pé direito. Esta origem foi escolhida com a perna flectida o que justifica o facto de termos valores negativos nas coordenadas Z com a perna livre porque se esta estiver completamente estica o pé pode se situar nos -25.6 mm em Z.

Para além destes pormenores temos ainda um que, à primeira vista, pode não ser de fácil compreensão. Se olharmos para as coordenadas em X, da perna livre e da perna de suporte, estas não têm nada a ver porque a perna livre está fixa pela anca tendo o maior deslocamento possível, sempre com o pé paralelo ao chão, no

sentido positivo do eixo xx. Já a perna de suporte tem o pé fixo ao chão tendo o maior deslocamento possível, sempre com o pé em completo contacto com o chão, no sentido negativo do eixo xx.

Todos estes limites apresentados não levam em conta o equilíbrio do humanóide ou do membro, como é o caso da perna de suporte que se encontra separada do resto do corpo. Uma grande parte destas coordenadas podem levar ao desequilíbrio.

Obter os limites angulares das pernas de suporte e livre não foi fácil. Todas as posições passam sempre por uma das cinemáticas, estas cinemáticas têm sempre como objectivo manter o pé plano, paralelo ao chão, com a frente do pé sempre apontada para a frente no sentido do eixo xx. Isto faz com que as posições cartesianas possíveis sejam muito limitadas. Estes limites angulares apresentados nesta tabela são valores aproximados, recolhidos através de vários cálculos teóricos (foi utilizada a cinemática directa e inversa no Matlab) de maneira a chegar a estes valores sem exceder os limites angulares dos servomotores.

Para além disso, muitas das posições não são possíveis de serem executadas na realidade devido a inconvenientes mecânicos. Algumas das correias, quando estão a realizar certos movimentos que tentam se aproximar mais dos limites saltam das suas transmissões e em alguns casos chegam mesmo a quebrar. Elas têm de ser esticadas e mesmo as que já estão esticadas tem algumas dificuldades de realizar estes movimentos. Por exemplo, elevar a perna esticada obriga o servo da anca a um grande esforço e a partir dos 45 graus por vezes a correia salta.

De modo a diminuir a introdução de coordenadas incorrectas que se encontrem fora dos limites apresentados, foram adicionadas algumas linhas de código logo à partida, no Matlab.

Referências

- Android World, “Historical Android Projects”. Disponível online em: <http://www.androidworld.com/prod06.htm> (acedido em Novembro de 2011).
- Behnke, S. and Stückler, J. (2008). Hierarchical Reactive Control for Humanoid Soccer Robots. *International Journal of Humanoid Robots*, 5(3): 375-396.
- CadSoft, “Eagle”. Disponível online em: <http://www.cadsoftusa.com/> (acedido em Novembro de 2011).
- Ferreira, P. (2008). “Ajudas Computacionais para Aplicação a um Robô Humanóide”, Dissertação de Mestrado, Universidade de Aveiro, 2008.
- Furuta, T., Tawara, T., Okumura, Y. & Shimizu, M.. (2001). Design and Construction of a Series of Compact Humanoid Robots and Development of Biped Walk Control Strategies. *Robotics and Autonomous Systems*, 37, 81-100.
- Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J. and Maisonnier, B. (2009). Mechatronic Design of NAO Humanoid. *Proceedings of the IEEE International Conference on Robotics and Automation*, 769-774.
- Godinho, R. (2011). Desenvolvimento do tronco e membros de uma plataforma humanóide híbrida. Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro.
- Hitec RCD Korea Inc., “General Specification of HSR-5980SG Coreless Digital Robot Servo”. Disponível online em: <http://www.hitecrd.co.jp/RC/servo/pdf/HSR5980SG.pdf> (acedido em Novembro de 2011).
- Honda, “History Robot Development Process”. Disponível online em: <http://world.honda.com/ASIMO/history/> (acedido em Novembro de 2011).
- Honda, “The Honda Humanoid Robot Asimo”. Disponível online em: <http://world.honda.com/ASIMO/> (acedido em Novembro de 2011).
- Idaho National Laboratory, U. S. Department of Energy, “Humanoid Robotics - Introduction”. Disponível online em: <https://inlportal.inl.gov/portal/server.pt/community/introduction/536> (acedido em Novembro de 2011).
- Interface, “Advanced Force Measurement”. Disponível online em: <http://www.interfaceforce.com/lbs-miniature-compression-load-button-p-40.html> (acedido em Novembro de 2011).
- Kim, J.-H., Kim, D.-H., Kim, Y.-J., Park, K.-H., Park, J.-H., Moon, C.-K., Ryu, J.-H., Seow, K.T. & Koh, K.-C.(2004). Humanoid Robot HanSaRam: Recent Progress and Developments. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8(1), 45-55.
- Miyachi Systems Corp, “Miniature Humanoid Robots” (em Japonês). Disponível online em: <http://www.miyachi-sys.com/services/robot/index.html> (acedido em Novembro de 2011).
- Muecke, K., Cox, P., Hong, D. (2006). DARwIn: Dynamic Anthropomorphic Robot with Intelligence, Part 1 – Concept & General Overview. *Servo Magazine*, 4(12).
- Ribeiro, M. (2010). Desenvolvimento dos Sistemas Sensorial e Motor para um Robô Humanóide. Dissertação de Mestrado, Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro.

- Rodrigues, M. (2008). Unidade de Processamento e Sistema de Visão para um Robô Humanóide. Dissertação, de Mestrado, Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro.
- Ruas, M. (2008). Sistemas de Comunicação e Controlo para uma Plataforma Humanóide. Dissertação de Mestrado, Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro.
- Sabino, R. (2009). Estrutura Híbrida de Locomoção para um Robô Humanóide. Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Aveiro.
- Santos, V. and Silva, F. (2006). Design and Low-Level Control of a Humanoid Robot Using a Distributed Architecture Approach. *Journal of Vibration and Control*, Special Issue on Modelling and Control of Artificial Locomotion Systems, 12(12): 1431-1456.
- Santos, V., Moreira, R., Ribeiro, M. and Silva, F. (2010). Development of a hybrid Humanoid Platform and Incorporation of the Passive Actuators. *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 679-684.
- Santos, V. (2003). Robótica Industrial: apontamentos teóricos, exercícios para aulas práticas e problemas de exame resolvidos. Departamento de Engenharia Mecânica, Universidade de Aveiro.
- Sciavicco, L. and Siciliano, B. (1996). *Modeling and Control of Robot Manipulators*, McGraw-Hill.
- Silva, F., Santos, V. (2007). Multipurpose Small-Cost Humanoid Platform and Modular Control Software Development. In *Humanoid Robots: Human-like Machines*, [ISBN: 978-3-902613-07-3], edited by Matthias Hanckel.
- Solis, J. and Takanishi, T. (2010). Recent Trends in Humanoid Robotics Research: Scientific Background, Applications, and Implications. *Accountability in Research*, 17: 278-298.
- Waseda University, Humanoid Research Institute, “Biped Walking Robot”. Disponível online em: http://www.humanoid.waseda.ac.jp/booklet/kato_4.html (acedido em Novembro de 2011).
- Waseda University, Humanoid Research Institute, “WABOT -WAseda roBOT”. Disponível online em: http://www.humanoid.waseda.ac.jp/booklet/kato_2.html (acedido em Novembro em 2011).
- Wikipédia, “Humanoid robot” - Online, Disponível em: http://en.wikipedia.org/wiki/Humanoid_robot [Último acesso: Novembro, 2011].
- Yamasaki, F., Matsui, T., Miyashita, T. & Kitano, Y. (2000). PINO the Humanoid: A Basic Architecture. RoboCup2000, volume 2019 of *Lecture Notes in Artificial Intelligence*, pp. 269-279, Springer.