



**Milton Ruas da Silva**

**Sistemas de Comunicação e Controlo para uma  
Plataforma Humanóide**





**Milton Ruas da Silva**

**Sistemas de Comunicação e Controlo para uma  
Plataforma 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 Convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Dr. Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro



## **o júri**

presidente

**Doutor José Fernando da Rocha Pereira**

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 Convidado da Universidade de Aveiro



## **agradecimentos**

Agradeço aos meus orientadores, Filipe Silva e Vítor Santos, por todo o apoio e paciência que entregaram gratuitamente para que este trabalho desse os seus frutos.

Em igual medida, agradeço toda a cooperação por parte do Departamento de Mecânica pela disponibilidade de recursos e toda a ajuda fornecida para a realização deste trabalho.





**palavras-chave**

Robótica humanóide, arquitectura de controlo, comunicação CAN, controlo de baixo-nível, controlo ao nível da tarefa.

**resumo**

A área da robótica humanóide apresenta uma dinâmica crescente em resultado dos avanços tecnológicos verificados nos últimos anos. No seguimento destes desenvolvimentos, vários grupos de investigação têm iniciado a construção de robôs de pequenas dimensões e baixo custo no sentido de realizarem trabalho científico em áreas tão diversas como o controlo, a percepção, a navegação e a cooperação.

Com a realização deste trabalho pretendeu-se dar seguimento a actividades anteriores concentradas no projecto e concepção de uma plataforma humanóide assente numa arquitectura de controlo distribuída. Entre os maiores desafios colocados inicialmente assumiram particular importância o desempenho da arquitectura de controlo e as limitações dos elementos de actuação (servomotores).

Nesta linha de ideais, esta dissertação descreve o desenvolvimento e avaliação do sistema de comunicações, do sistema de controlo de baixo nível e de diversos algoritmos e estratégias de controlo ao nível da tarefa. Neste sentido, foram executadas um conjunto de actividades que permitiram desenvolver uma compreensão aprofundada dos problemas, desafios e compromissos inerentes ao trabalho, e clarificar as possibilidades de aplicação prática.



**keywords**

Humanoid robotics, control architecture, CAN-communications, low-level control, task-level control.

**abstract**

The field of humanoid robotics has been attracting the attention of a growing community, both from the industry and academia. The continuous progress in robotics technology opens up new possibilities for academic research on low-cost and easy-to-design humanoids. The main scope has been the development of humanoid platforms to carry out research on control, navigation and perception.

Beyond the ultimate goal of applicability and validation of the platform, the proposed work aimed the prosecution of ongoing developments covering the hardware components and the control issues that must be solved regarding the feasibility of upright biped walking. The main scope of the work beneath this dissertation has been the design and implementation of a distributed architecture for a humanoid robot where centralized and local controls co-exist and concur to provide a robust and versatile operation.

As the capabilities of the control system are at the core of the robot development, a research task specifically addressed the communications system and the control issues. One of the main issues is to investigate how centralised and local modules may co-exist and concur to provide robust and versatile operation. Research along the above development phases has been conducted with the purpose of clarifying the system's performance according to the initial ideas and requirements, and the possibilities of practical applications.



# ÍNDICE

<b>1 INTRODUÇÃO</b> .....	<b>1</b>
1.1 ENQUADRAMENTO DO TRABALHO .....	2
1.2 OBJECTIVOS E MOTIVAÇÕES .....	4
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO.....	5
<b>2 ARQUITECTURA DAS COMUNICAÇÕES</b> .....	<b>7</b>
2.1 INTRODUÇÃO .....	7
2.2 COMUNICAÇÃO RS-232.....	10
2.2.1 <i>Device Drivers</i> de Comunicação na Unidade Principal .....	10
2.2.2 Protocolo RS-232 .....	11
2.2.3 Unidade Principal .....	13
2.2.4 Unidade Distribuidora <i>MASTER</i> .....	16
2.3 COMUNICAÇÃO CAN .....	20
2.3.1 Protocolo CAN .....	22
2.3.2 Funcionamento do Barramento CAN.....	24
2.3.3 Unidade Master .....	29
2.3.4 Unidades Slave .....	30
2.4 DISCUSSÃO DOS RESULTADOS.....	32
<b>3 SISTEMA DE CONTROLO DE BAIXO-NÍVEL</b> .....	<b>33</b>
3.1 INTRODUÇÃO .....	33
3.2 CONTROLO DA PLATAFORMA HUMANÓIDE .....	34
3.2.1 A Unidade de Controlo Local.....	34
3.2.2 ACTUADORES: OS SERVOMOTORES.....	35
3.2.3 Setup Experimental .....	39
3.2.4 Actuação sobre o Servomotor .....	40
3.2.5 Leitura Sensorial do Servomotor .....	41
3.3 ESTUDO DO SERVOMOTOR EM MALHA ABERTA.....	47
3.3.1 Resposta ao Degrau em Malha Aberta .....	48
3.3.2 Controlo de Velocidade .....	49
3.4 ESTUDO DO SERVOMOTOR EM MALHA FECHADA.....	51
3.4.1 O Controlador .....	52
3.4.2 Controlo Integral (I).....	54
3.4.3 Controlo Proporcional+Integral (PI) .....	56
3.4.4 Controlo Integral+Derivativo (ID) .....	58
3.4.5 Controlo Proporcional+Integral+Derivativo (PID) .....	58
3.5 APLICAÇÃO DOS ALGORITMOS NO ROBOT HUMANÓIDE.....	60
3.5.1 Movimento de Flexão em Malha Aberta de uma Perna.....	62
3.5.2 Movimento de Flexão em Malha Fechada de uma Perna .....	65
3.5.3 Movimento das duas Pernas .....	66
3.6 DISCUSSÃO DOS RESULTADOS.....	67
<b>4 SISTEMA DE CONTROLO BASEADO NAS FORÇAS</b> .....	<b>69</b>
4.1 INTRODUÇÃO .....	69
4.2 ESTRUTURA DOS PÉS .....	69
4.2.1 Sensores de Força: os Extensómetros.....	69
4.2.2 O Pé como Elemento Sensitivo a Forças Externas .....	70
4.3 MEDIÇÃO DAS FORÇAS DE REACÇÃO .....	72

4.3.1	Interface entre os Extensómetros e o Microcontrolador .....	72
4.3.2	Resultados Experimentais .....	75
4.3.3	Amostragem dos Sensores .....	78
4.4	MEDIÇÃO DO CENTRO DE PRESSÃO .....	81
4.4.1	Representação do Centro de Pressão .....	81
4.4.2	Cálculo do Centro de Pressão.....	81
4.5	CONTROLO DE EQUILÍBRIO .....	83
4.5.1	Estrutura do Controlador .....	83
4.5.2	Controlo Proporcional .....	84
4.5.3	Controlo através da Matriz Jacobiana .....	84
4.5.4	Resultados Experimentais: Variação do Plano de Suporte .....	92
4.5.5	Resultados Experimentais: Resposta a Perturbações Externas .....	97
4.5.6	Resultados Experimentais: Resposta do Pé Livre.....	99
4.6	LOCOMOÇÃO ATRAVÉS DO CONTROLO DO COP .....	101
4.6.1	Controlo do CoP DE Referência.....	101
4.6.2	Análise Estática do CoP de Referência.....	102
4.6.3	Análise Dinâmica do CoP Referência.....	106
4.6.4	Execução de Trajectórias Rectilíneas .....	110
4.6.5	Execução de Trajectórias Elípticas.....	114
4.6.6	Execução de Trajectórias Rectangulares .....	115
4.7	DISCUSSÃO DOS RESULTADOS.....	118
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>121</b>
	<b>REFERÊNCIAS.....</b>	<b>121</b>

# LISTA DE FIGURAS

Fig. 1.1: QRIO (Sony), Asimo (Honda) e KHR 3 (Hubo Lab).	1
Fig. 1.2: Modelo 3D do robot e implementação actual.	2
Fig. 1.3: Arquitectura distribuída da plataforma.	3
Fig. 2.1: Exemplo de uma board PC-104.	7
Fig. 2.2: Placa de controlo Master/Slave.	7
Fig. 2.3: Rede completa de Microcontroladores.	7
Fig. 2.4: Algoritmo básico dos <i>device drivers</i> de actuação / leitura sensorial.	14
Fig. 2.5: Algoritmo básico da função <i>sendmessage</i> .	14
Fig. 2.6: Algoritmo detalhado da função <i>sendmessage</i> .	15
Fig. 2.7: Algoritmo de gestão das comunicações RS-232 na unidade <i>Master</i> .	16
Fig. 2.8: Recepção de byte por RS-232 e armazenamento no buffer de recepção.	17
Fig. 2.9: Envio do buffer de transmissão por RS-232.	18
Fig. 2.10: Processamento de uma mensagem na unidade <i>Master</i> .	18
Fig. 2.11: Formato <i>standard</i> das mensagens CAN.	20
Fig. 2.12: Arbitragem de mensagens segundo o sistema de colisões CSMA/BA.	21
Fig. 2.13: Codificação NRZ.	25
Fig. 2.14: Particionamento temporal de um bit.	25
Fig. 2.15: Registos associados à máscara e filtragem.	26
Fig. 2.16: Algoritmos de transmissão/recepção de mensagens CAN no <i>Master</i> .	29
Fig. 2.17: Algoritmo de troca de informação pelo CAN no <i>Slave</i> .	30
Fig. 3.1: Arquitectura da plataforma humanóide.	33
Fig. 3.2: Constituição de uma unidade <i>slave</i> .	34
Fig. 3.3: Imagens de uma unidade de controlo local.	35
Fig. 3.4: Servomotor da HITEC.	35
Fig. 3.5: Representação do interior de um servomotor.	35
Fig. 3.6: Sinal de PWM aplicado no servomotor.	36
Fig. 3.7: Correia de transmissão aplicada a um servo.	37
Fig. 3.8: Circuito do controlador de posição de um FUTABA S3003.	37
Fig. 3.9: Representação esquemática do controlador de posição interno.	38
Fig. 3.10: Setup experimental.	39
Fig. 3.11: Arquitectura das comunicações no <i>setup</i> .	39
Fig. 3.12: Geração de um sinal de PWM através de dois timers.	40
Fig. 3.13: Organização temporal das interrupções na geração do PWM.	41
Fig. 3.14: Impulso de tensão medido na presença de cargas/velocidades elevadas.	42
Fig. 3.15: Relação entre a posição e a largura de impulso.	42
Fig. 3.16: Configuração possível para medição da corrente.	43
Fig. 3.17: Organização das interrupções (setas) na medição sensorial.	44
Fig. 3.18: Algoritmo de leitura dos três servomotores.	44
Fig. 3.19: Multiplexagem na leitura dos servos.	44
Fig. 3.20: Algoritmo de processamento da tensão medida para cada servomotor.	46
Fig. 3.21: Processamento final (fim do período de PWM).	46
Fig. 3.22: Representação do servomotor por uma função de transferência $G(s)$ .	47
Fig. 3.23: Exemplo da resposta de um sistema por compensar com a indicação das suas características.	48
Fig. 3.24: Comparação das respostas ao degrau para duas cargas no percurso de $-45$ para $+45^\circ$ .	48
Fig. 3.25: Aplicação de um degrau de $-45^\circ$ para $+45^\circ$ no instante $t=1,8s$ .	49
Fig. 3.26: Aplicação de uma rampa de posição de velocidade média $50^\circ/s$ .	49
Fig. 3.27: Trajectória polinomial de terceira ordem.	50
Fig. 3.28: Comportamento da velocidade e da aceleração na trajectória polinomial.	50
Fig. 3.29: Resposta à rampa com duas cargas diferentes ( $\Delta p=5^\circ$ , $\Delta t=100ms$ ).	51
Fig. 3.30: Resposta à rampa com uma carga pesada ( $\Delta p=5^\circ$ , $\Delta t=100ms$ ).	51
Fig. 3.31: Resposta ao polinómio para duas cargas diferentes ( $T_{traj}=1s$ ).	51
Fig. 3.32: Controlo externo do servomotor.	52
Fig. 3.33: Compensador PID incremental.	52
Fig. 3.34: Resposta à rampa com uma carga de $675g$ ( $K_I=0.08$ ).	54
Fig. 3.35: Resposta à rampa com uma carga de $1129g$ ( $K_I=0.08$ ).	54
Fig. 3.36: Resposta à rampa com uma carga de $1129g$ ( $K_I=0.20$ ).	55
Fig. 3.37: Erro da resposta da Fig. 3.36.	55
Fig. 3.38: $K_I=0.02$ ( $m=924g$ ).	55
Fig. 3.39: $K_I=0.07$ ( $m=924g$ ).	55

Fig. 3.40: $K_I=0.15$ ( $m=924g$ ).....	56
Fig. 3.41: $K_I=0.30$ ( $m=924g$ ).....	56
Fig. 3.42: Fenómeno do overshoot para valores de $K_I$ elevados (carga de 924g).....	56
Fig. 3.43: Erro da resposta da Fig. 3.42.....	56
Fig. 3.44: Correção do overshoot com o aumento de $K_P$ (carga de 924g).....	57
Fig. 3.45: Erro da resposta da Fig. 3.44.....	57
Fig. 3.46: Variação de $K_P$ para $K_I=0.05$ (carga de 675g).....	57
Fig. 3.47: Variação de $K_P$ para $K_I=0.10$ (carga de 675g).....	57
Fig. 3.48: Resposta ao polinómio com $K_I=0.05$ e $K_P=0.00$ (carga de 675g).....	58
Fig. 3.49: Resposta ao polinómio com $K_I=0.10$ e $K_P=0.00$ (carga de 675g).....	58
Fig. 3.50: Comparação entre dois conjuntos de parâmetros durante o <i>tunning</i> ( $m=675g$ ). .....	59
Fig. 3.51: Situação de instabilidade ( $m=675g$ ). .....	59
Fig. 3.52: PID optimizado para uma carga de 675g. ....	60
Fig. 3.53: Repetição do ensaio com os parâmetros de compensação óptimos. ....	60
Fig. 3.54: Movimento de flexão de uma perna. ....	61
Fig. 3.55: Carga de 2 kg utilizada no topo da perna. ....	61
Fig. 3.56: Junta do pé.....	62
Fig. 3.57: Junta do Joelho. ....	62
Fig. 3.58: Junta da Anca. ....	63
Fig. 3.59: Junta do pé.....	63
Fig. 3.60: Junta do joelho. ....	64
Fig. 3.61: Junta da anca.....	64
Fig. 3.62: Junta do pé.....	65
Fig. 3.63: Junta do joelho. ....	65
Fig. 3.64: Junta da anca.....	65
Fig. 3.65: Movimento de flexão nas duas pernas.....	66
Fig. 3.66: Conjunto das duas pernas executando um movimento de flexão (sequência superior) e um movimento lateral (sequência inferior) com uma carga de 2.1 kg em completa sincronia de pernas. ....	66
Fig. 3.67: Resposta ao polinómio com um controlador PI. Imagens superiores e inferior esquerda: comportamento das três juntas envolvidas no movimento de flexão; imagem inferior direita: comportamento da junta lateral do pé na realização do deslocamento lateral. ....	67
Fig. 4.1: Exemplo de um extensómetro típico. ....	70
Fig. 4.2: Relação entre força imposta e deformação elástica. ....	70
Fig. 4.3: Estrutura completa de um pé. ....	71
Fig. 4.4: Colocação dos sensores na estrutura do pé. ....	71
Fig. 4.5: Peça de acrílico contendo o extensómetro para medição da sua deformação.....	71
Fig. 4.6: Pontos de contacto entre as 2 plataformas do pé.....	71
Fig. 4.7: Visão completa do pé.....	72
Fig. 4.8: Circuito de interface original dos extensómetros.....	72
Fig. 4.9: Circuito de calibração da ponte de Wheatstone.....	73
Fig. 4.10: Relação entre o potenciómetro e a resistência total de calibração. ....	74
Fig. 4.11: Circuito de acondicionamento de sinal com simetria completa.....	74
Fig. 4.12: Localização dos sensores no pé. ....	76
Fig. 4.13: Medição independente de cada sensor. ....	76
Fig. 4.14: Medição dos sensores em conjunto.....	76
Fig. 4.15: Relação entre a medição de cada sensor e o peso aplicado (experiência 1) .....	77
Fig. 4.16: Valores dos sensores com massas colocadas no centro da base (experiência 2) .....	78
Fig. 4.17: Projecção do centro de massa sobre o pé na situação de equilíbrio. ....	81
Fig. 4.18: Perna completa sobre um pé sensível a forças. ....	83
Fig. 4.19: Diagrama com os controladores aplicados para a compensação de equilíbrio. ....	84
Fig. 4.20: Modelo da perna humanóide com a associação das diversas variáveis de interesse. ....	86
Fig. 4.21: da inclinação do plano de suporte segundo o eixo yy (esquerda) e o eixo xx (direita). ....	92
Fig. 4.22: Variação do declive do plano segundo o eixo xx (controlo proporcional). ....	93
Fig. 4.23: Variação do declive do plano segundo o eixo xx - controlo pelo Jacobiano. ....	94
Fig. 4.24: Variação do declive do plano segundo o eixo yy (controlo proporcional). ....	95
Fig. 4.25: Variação do declive do plano segundo o eixo yy - controlo pelo Jacobiano. ....	95
Fig. 4.26: Inclinação do plano segundo os eixos xx e yy. ....	96
Fig. 4.27: Variação do declive do plano na diagonal dos eixos xx e yy (controlo proporcional).....	96
Fig. 4.28: Introdução de perturbações externas na perna (frente, trás, esquerda e direita) - controlo Proporcional.....	97
Fig. 4.29: Introdução de perturbações externas na perna (frente, trás, esquerda e direita) - controlo pelo Jacobiano.....	98
Fig. 4.30: Levantamento do pé no ar - controlo proporcional.....	100



Fig. 4.31: Levantamento do pé no ar - controlo pelo Jacobiano. ....	100
Fig. 4.32: Actuação sobre o Centro de Pressão referência.....	102
Fig. 4.33: Actuação sobre a altura desejada da anca. ....	102
Fig. 4.34: Relação entre o CoP medido e o actuado, para 30 passos ao longo do eixo xx. ....	103
Fig. 4.35: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo xx. .	104
Fig. 4.36: Relação entre o CoP medido e o actuado, para 32 passos ao longo do eixo yy. ....	104
Fig. 4.37: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo yy. .	105
Fig. 4.38: Relação entre o CoP medido e o actuado, para 25 passos ao longo do eixo xy. ....	105
Fig. 4.39: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo xy. .	106
Fig. 4.40: Resposta ao polinómio aplicado nos eixos xy com K=50. ....	107
Fig. 4.41: Resposta ao polinómio aplicado nos eixos xy com K=70. ....	108
Fig. 4.42: Resposta ao polinómio aplicado de duração 1s (K=50). ....	109
Fig. 4.43: Resposta ao polinómio aplicado de duração 4s (K=50). ....	110
Fig. 4.44: Trajectória rectilínea ao longo do eixo xx (Período=2s e K=50). ....	111
Fig. 4.45: Trajectória rectilínea ao longo do eixo yy (Período=2s e K=50). ....	112
Fig. 4.46: Flutuações do CoP e da cinemática directa na componente xx, na realização de uma trajectória sobre apenas o eixo yy. ....	113
Fig. 4.47: Flutuações do CoP e da cinemática directa na componente yy, na realização de uma trajectória sobre apenas o eixo xx. ....	113
Fig. 4.48: Variação do CoP ao longo do percurso elíptico - componentes x (azul) e y (verde) - em comparação com o real, e o respectivo erro. ....	114
Fig. 4.49: Cinemática directa do joelho e da anca (vista 3D). ....	114
Fig. 4.50: Cinemática directa do joelho e da anca (vista 2D de cima). ....	115
Fig. 4.51: Resposta a 10 voltas de uma trajectória rectangular para vários parâmetros do controlador local. ....	116
Fig. 4.52: Resposta a 10 voltas de uma trajectória rectangular sem e com controlo local ( $K_{jacobiano}=50$ ). ....	117
Fig. 4.53: Resposta a 10 voltas de uma trajectória rectangular para diferentes velocidades - $[K_I, K_P, K_{jacobiano}]=(5,40,30)$ . ....	117
Fig. 4.54: Duas experiências realizadas em diferentes momentos, com iguais parâmetros de controlo. ....	118



# LISTA DE TABELAS

Tabela 1: Características do Hardware e Software na unidade principal.....	8
Tabela 2: Lista de <i>device drivers</i> da unidade principal. ....	10
Tabela 3: Device drivers adicionais.....	10
Tabela 4: Tipos de mensagens USART (primeiro byte de cada frame).....	13
Tabela 5: Endereços atribuídos às diversas unidades de controlo. ....	21
Tabela 6: Campos do identificador de um pacote CAN. ....	22
Tabela 7: Campos do byte <i>Sensor Flags</i> . ....	24
Tabela 8: Resultado da filtragem para cada bit.....	26
Tabela 9: Configuração dos filtros para redireccionamento de pacotes para os dois buffers de recepção (padrão a3 a2 a1 a0 = endereço do SCU). ....	27
Tabela 10: Atribuição de prioridades entre cada buffer de transmissão. ....	27
Tabela 11: Especificações do servo da HITEC HS-805BB. ....	36
Tabela 12: Binários exigidos na simulação de um passo.....	36
Tabela 13: Lista de cargas utilizadas para teste. ....	39
Tabela 14: Erros em regime estacionário em diferentes posições para uma carga de 1138g.....	48
Tabela 15: Lista de pesos usados no teste dos sensores de força.....	75
Tabela 16: Resultados experimentais para cada sensor nas duas experiências.....	76
Tabela 17: Parâmetros das regressões lineares traçadas. ....	77
Tabela 18: Equações para o controlo proporcional.....	84
Tabela 19: Leis de controlo utilizando a matriz Jacobiana de centro de massa e de altura. ....	91
Tabela 20: Equações de controlo para o compensador proporcional, considerando o CoP de referência nulo. ....	92
Tabela 21: Equações de controlo para o compensador com o Jacobiano, considerando o CoP desejado nulo. ....	92



## Capítulo 1

# Introdução

A concepção de um robot Humanóide constitui um dos maiores desafios na área da robótica. Construir um ser artificial antropomórfico semelhante ao homem é um sonho inato do nosso engenho, e não é para menos, pois o ser humano é a forma de vida mais complexa existente à face da Terra. O século XX encheu a nossa imaginação com livros e filmes que demonstram esse sonho do ser artificial capaz de desempenhar tarefas, aprender coisas por si mesmo, interagir connosco, expressar emoções, possuir uma consciência própria; tudo características que, por enquanto, ainda consideramos como puramente humanas. Marcas como a Honda [1], a Sony [2] ou a Fujitsu (Fig. 1.1) já deram os primeiros passos no desenvolvimento de máquinas que imitam os comportamentos físicos dos seres humanos – caminhar, dançar ou pegar objectos. Outros passos também já foram dados no que respeita ao processamento de visão e som, com o objectivo de realizar tarefas e de interagir com o ser humano.



Fig. 1.1: QRIO (Sony), Asimo (Honda) e KHR 3 (Hubo Lab).

## 1.1 ENQUADRAMENTO DO TRABALHO

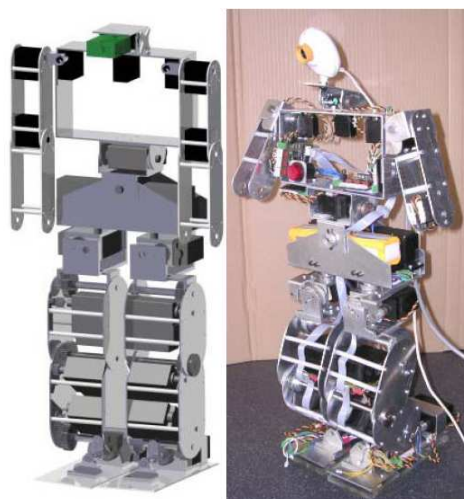
Muitos grupos de investigação iniciaram a construção de robôs de baixo custo no sentido de realizarem investigação em áreas tão diversas como o controlo, a percepção, a navegação, o comportamento ou a cooperação [3-7]. Este foi, também, o móbil principal que levou um grupo do Departamento de Engenharia Mecânica da Universidade de Aveiro a encetar a tarefa de construção de uma tal plataforma [8-9]. O estado actual de desenvolvimento perspectiva a abordagem de algoritmos eficientes ao nível do controlo, planeamento e percepção [10-11]. O projecto Humanóide na Universidade de Aveiro encontra a sua motivação em diversas vertentes, das quais se destacam as seguintes:

- A aposta nos robôs humanóides como a via mais promissora para chegar a sistemas de elevada mobilidade, versatilidade de operação e facilidade de interacção com os humanos;
- A criação de uma plataforma de investigação de grande valor pedagógico face aos enormes desafios científicos e técnicos, à diversidade de problemas, ferramentas e níveis de integração;
- A promoção do envolvimento de um grupo de estudantes da UA em competições robóticas internacionais. Por exemplo, o ROBOCUP e o FIRA são duas organizações internacionais que realizam anualmente competições na classe dos humanóides.

A plataforma humanóide possui um conjunto de 22 graus de liberdade, distribuídos da seguinte forma:

- 2 ortogonais em cada pé ( $2 \times 2$ );
- 1 em cada joelho ( $1 \times 2$ );
- 3 em cada anca, segundo os três eixos tridimensionais ( $3 \times 2$ );
- 2 no tronco ( $2 \times 1$ );
- 3 em cada braço ( $3 \times 2$ );
- 2 no suporte da câmara (cabeça) ( $2 \times 1$ ).

A estrutura é constituída essencialmente por alumínio e aço nos eixos e outros pequenos componentes, pesando um total de 6 kg com as baterias incluídas, e medindo cerca de 60 cm de altura. Estes valores foram estabelecidos de acordo com as regras impostas pelo *RoboCup*, baseando-se no pressuposto de que para valores superiores estes, o uso de servomotores de baixo custo poderia tornar-se inviável dada a impossibilidade de conciliar binários de motores e pesos dos equipamentos e acessórios como as baterias. Por razões de estética e de acomodação de componentes, foi adoptada uma estrutura em forma de exoesqueleto (carapaça) dotando assim o sistema de módulos ocultos onde são alojados os motores, sensores, cablagens, placas de controlo, etc (Fig. 1.2).



**Fig. 1.2: Modelo 3D do robot e implementação actual.**

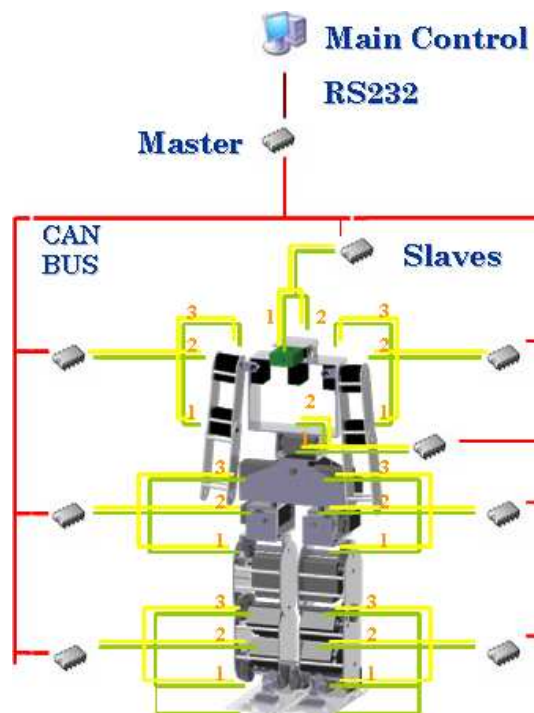
Preocupações com a autonomia energética do sistema, exigiram a escolha de baterias de elevada capacidade de corrente, pois será necessário alimentar 22 motores de binário apreciável (embora uns mais do que outros). Tendo em conta esses aspectos utilizaram-se duas baterias de 7.4V ligadas em paralelo com uma capacidade total máxima de 9600mAh.

No que respeita ao controlo, foram assumidas, desde logo, as vantagens de uma arquitectura distribuída e modular baseada num barramento CAN [12-14]. Este é responsável por permitir a troca de informação entre as diversas unidades locais de controlo a partir de uma unidade mestre que faz a gestão da rede e que está directamente ligada a uma unidade primária de decisão. Esta unidade central de processamento poderá ser um computador comum ou uma *embedded motherboard* do tipo PC-104 ou nano ITX de dimensões reduzidas.

Neste momento, a plataforma é constituída por 9 unidades de controlo, 8 de controlo local dos actuadores e sensores, e um de controlo de tráfego na rede CAN. As unidades de controlo local estão distribuídas de forma a agrupar conjuntos de três actuadores relativos a um determinado membro, como é o caso das pernas ou dos braços (Fig. 1.3). Com uma arquitectura deste género, pretende-se que a *hardware* que constitui estes módulos seja idêntico com um *software* muito similar entre eles. Implementando esta estratégia consegue-se um grau de fiabilidade superior, uma vez que os módulos são independentes permitindo que as anomalias sejam mais facilmente detectadas e corrigidas. Os módulos podem ser trocados facilmente, pois bastará programar o microprocessador com o *firmware* padrão que ele mesmo se adaptará à tarefa necessária, tanto através do endereço atribuído como pela gestão directa da unidade central de processamento.

Em resumo, apresentam-se as vantagens da arquitectura distribuída:

- Sistemas fiáveis (operação independente)
- Sistemas de controlo mais simples
- Mais fácil detecção de anomalias
- Actualização fácil do *firmware*



**Fig. 1.3: Arquitectura distribuída da plataforma.**

Para controlo da junta foram seleccionados servomotores de vários binários diferentes, de acordo com a exigência de cada junta. Cada microcontrolador pode controlar até três servomotores tal como indicado na Fig. 1.3. Relativamente à percepção destacam-se os seguintes sensores proprioceptivos e inerciais:

1. Potenciómetros de medição da posição das juntas, incluídas nos servomotores (é possível extrair deste sensor tanto a posição da junta como a corrente consumida);
2. Sensores de força, baseados em extensómetros, para medição das forças de reacção dos pés;
3. Inclínómetros e giroscópios para medição da aceleração gravítica e da velocidade angular respectivamente.

Cada microcontrolador poderá monitorar até três sinais de posição (retirados dos três servos), e até quatro sensores adicionais, podendo ser de força ou inerciais. Tal compromisso advém de limitações da gestão dos motores por parte do microcontrolador, bem como de questões relativas à organização da informação trocada pelo barramento de comunicações (protocolo).

## 1.2 OBJECTIVOS E MOTIVAÇÕES

O trabalho descrito nesta dissertação, envolve o estudo, desenvolvimento e implementação de um conjunto de estratégias e algoritmos de controlo com vista à locomoção básica de um robot humanóide. Do conjunto de estratégias investigadas, podemos distingui-las em três partes:

1. Controlo de posição e velocidade em malha aberta e fechada das juntas dos membros inferiores (pernas) por actuação directa sobre os respectivos servomotores. Para o controlo em malha fechada serão utilizados como sinais de retorno a posição do servo reportada pelo seu potenciómetro e/ou a corrente consumida pelo mesmo;
2. Controlo de equilíbrio da estrutura contra variações de declive do solo e perturbações externas, pela utilização de sensores de força presente sobre a base dos pés. Extensómetros de resistência dependente da força serão utilizados em vários pontos do pé para permitir a medição bidimensional do centro de pressão;
3. Realização da locomoção recorrendo ao controlador de equilíbrio especificado no ponto anterior, pela variação do centro de pressão desejado sobre a base de um dos pés. Por este processo, é possível, simultaneamente, a locomoção e a compensação contra irregularidades no solo e perturbações externas.

Dado que este robot possui uma estrutura modular, constituída por diversas unidades microcontroladoras espalhadas ao longo do corpo, é essencial uma comunicação robusta e fiável entre a unidade de processamento central, e cada uma das unidades microcontroladoras. Para tal trabalhou-se sobre o barramento CAN, conhecido pela sua robustez contra interferências eléctricas, dedicando uma parte do trabalho no desenvolvimento de um protocolo de comunicações capaz de conferir o máximo de fiabilidade para a configuração e utilização dos controladores ao nível da unidade central.

Finalmente, efectuou-se uma revisão ao *hardware* das unidades microcontroladoras, bem como à electrónica sensorial que realiza o acondicionamento de sinais externos, tal como é o caso da electrónica de processamentos dos sinais provenientes dos sensores de força localizados nos pés. Logicamente o processamento de sinal ao nível do *firmware* (pelo microcontrolador) foi tido em conta, pela implementação de filtros e de procedimentos de calibração, de modo a proporcionar o máximo de qualidade possível nas medições. Desta forma, aos algoritmos de controlo, que recorrem a estes sinais, é-lhes conferida fiabilidade adicional.

Como método para a verificação dos algoritmos implementados, começou-se por aplicar sobre motores aparte, como forma de garantir a funcionalidade dos mesmos. Posteriormente passou-se para a utilização de uma perna somente, para teste da robustez dos algoritmos; e finalmente utilizaram-se ambas as pernas, com um peso adicional no topo, para verificação da fiabilidade perante as variações de carga exercidas sobre cada junta.

Após a conclusão deste trabalho, pretende-se a locomoção dos membros inferiores, de modo a que o robot seja capaz de se deslocar entre dois pontos do espaço pela troca de comandos básicos com uma unidade central de processamento portátil (baseada em PC104 ou nano-ITX). Por isso, a unidade central apenas terá que fornecer as coordenadas de destino e a velocidade a que deverá realizar a trajectória, enquanto que os microcontroladores responsáveis pela locomoção das juntas de interesse, se responsabilizam pela implementação dos algoritmos de controlo tanto para realização das sequências de movimento de uma forma suave e contínua, como também a compensação contra irregularidades no solo e outras perturbações externas. Pretende-se por isso que todo o algoritmo de controlo de baixo-nível para locomoção/equilíbrio seja implementado ao nível do microcontrolador, minimizando assim a comunicação entre estas unidades e a central.



## 1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

A presente dissertação encontra-se organizada nos seguintes capítulos:

**Capítulo 2 - Arquitectura das Comunicações:** descreve a implementação do protocolo de comunicação entre a unidade central de processamento e os diversos microcontroladores.

**Capítulo 3 - Sistema de Controlo de Baixo Nível:** apresenta os algoritmos de controlo de baixo-nível aplicados sobre os actuadores do sistema.

**Capítulo 4 - Sistema de Controlo Baseado nas Forças:** descreve a implementação e teste dos algoritmos de controlo, baseados na informação das forças de reacção, responsáveis pelo equilíbrio estático da estrutura e pela realização de movimentos simples.

**Capítulo 5 - Conclusões:** faz-se uma avaliação dos resultados em termos de funcionalidade e desempenho do sistema e traçam-se as perspectivas de trabalho futuro.



## Capítulo 2

# Arquitectura das Comunicações

### 2.1 INTRODUÇÃO

O sistema de controlo implementado é baseado numa configuração distribuída *master/slave* e é constituída por três tipos de unidades ligadas em rede: a unidade central de processamento, a unidade *master* e as unidades *slave*. A unidade central de controlo é responsável pela gestão global dos procedimentos, efectuando o cálculo das configurações que as juntas tem de adoptar em função dos valores dos sensores. A unidade *Master* (mestre) tem como principal tarefa distribuir os comandos provenientes da unidade principal pelas diversas unidades locais (*slaves*), bem como direccionar os dados sensoriais provenientes das *slaves* para a unidade principal. As unidades *Slave* (escravo) têm como principais funções a geração da onda de pulso modulado (PWM) de controlo dos servomotores e a aquisição dos sinais dos diversos sensores.

A unidade central de controlo está definida como uma placa de controlo baseada no padrão PC104, tal como apresentada na Fig. 2.1, mas pode-se utilizar igualmente um PC externo ou um computador portátil com porta série RS-232. Para a presente dissertação utilizou-se como ferramentas um computador portátil com o software *MatLab*, dispondo de recursos para a troca de dados com a unidade *Master* e seu consequente processamento. Para as unidades de controlo local (*master/slave*), a escolha recaiu sobre os microcontroladores PIC da série 18F da *Microchip* – PIC18F258(0) – pelo seu baixo custo e por possuírem diversos periféricos que facilitam a actuação e as leituras sensoriais, bem como interfaces para redes de comunicações, incluindo a UART e o CAN (Fig. 2.3).

O meio de comunicação utilizado entre a unidade central e a unidade *Master* é a linha série ponto-a-ponto baseada na norma **RS-232** (acesso assíncrono byte a byte a um *baudrate* de 115200 bps). Entre a unidade *master* e as unidades *slave* é usado um barramento **CAN** (*Controller Area Network* - versão *fullCAN* 2.0A a uma taxa de transmissão/recepção de 1Mbps).



Fig. 2.1: Exemplo de uma board PC-104.

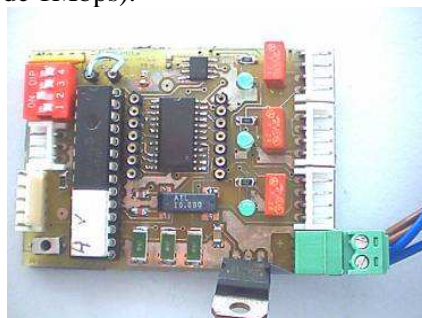


Fig. 2.2: Placa de controlo Master/Slave.

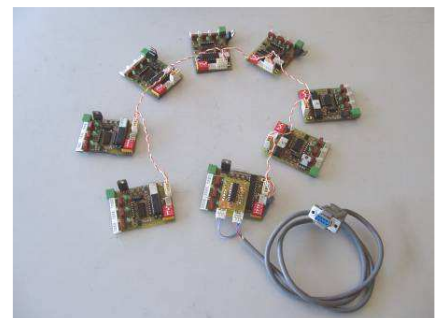


Fig. 2.3: Rede completa de Microcontroladores.

Até ao momento, a rede implementada é constituída por uma placa *master*, pelo qual designaremos por MCU (*Master Control Unit*), que efectua a interface entre a unidade principal e as unidades *slave*, e oito placas *slave*, designadas por SCU (*Slave Control Unit*), que efectuarão o controlo local até três actuadores através da geração de uma onda de pulso modulado em largura (PWM), e a aquisição de até 16 sinais analógicos usando um *multiplexer* (Fig. 2.2).

A organização enunciada tem como objectivo o controlo distribuído de secções distintas da plataforma humanóide, agrupando, para cada unidade, as juntas que estão directamente relacionadas. Um exemplo, é o caso das juntas do tornozelo e do joelho que possuem um controlador dedicado, que, por aquisição dos sinais analógicos dos sensores de força instalados nos pés, pode controlar o equilíbrio da perna associada, através de métodos de compensação em malha fechada, actuando sobre os respectivos actuadores. Desta forma, obtém-se um controlo localizado independente do resto do sistema sem que haja necessidade permanente de interagir com a unidade central de controlo.

Os sinais analógicos adquiridos actualmente podem ser de quatro origens diferentes:

- Potenciómetros internos aos servomotores, indicativos da sua posição;
- Extensómetros presentes na base de cada pé, para medição da força aplicada, para posterior cálculo do Centro de Pressão exercido;
- Acelerómetros/inclinómetros, que pela medição da aceleração da gravidade em duas componentes ortogonais, medem a inclinação do objecto onde se encontram localizadas;
- Giroscópios para medição da velocidade angular, para compensação das forças dinâmicas exercidas sobre o robot.

Estes valores são convertidos e registados localmente em cada unidade *slave* sendo depois enviados via CAN para o controlador *Master*. Os *Slaves* estão preparados também para receber mensagens via CAN. Estas mensagens consistem basicamente nos dados de actuação a aplicar sobre cada unidade local:

- Posições finais que os actuadores têm de tomar;
- Velocidade média a que têm que se mover até atingir a posição final;
- Tipo de controlador activo sobre as três juntas;
- Parâmetros de compensação para o algoritmo de compensação seleccionado;
- *Flags* booleanas de controlo (ex.: PWM activados).

O controlador *master* tem a tarefa de receber a informação enviada pelos *slaves* via CAN e registá-la para que esteja disponível para ser enviada para a unidade central de controlo quando solicitada. Este controlador mantém, por isso uma representação do estado actual das juntas (actuadores e sensores) que disponibilizará ao controlo central sempre que este o pedir. O processo é bidireccional e o controlador *master* também recebe as ordens da unidade central e despacha para o *slave* respectivo.

**Tabela 1: Características do Hardware e Software na unidade principal.**

Unidade Central de Controlo	<b>Computador com porta série RS-232</b> <ul style="list-style-type: none"> <li>● Software de suporte: MatLab 7.xx</li> <li>● Device Drivers de comunicação série: <i>mini-toolbox</i> cport v1.3</li> </ul>
Unidades de controlo local <i>Master/Slave</i>	<b>PIC18F258 da Microchip</b> <ul style="list-style-type: none"> <li>● Memória de programa: 2 MB</li> <li>● Memória de dados: 4 KB</li> <li>● Velocidade de processamento: 10 MIPS (<math>f_{osc}=40\text{MHz}</math> com a PLL activa)</li> <li>● Instruções de 16 bits e <i>datapath</i> de 8 bits</li> <li>● Definição de dupla prioridade nas interrupções.</li> <li>● Diversos periféricos: <i>timers</i>, módulos CCP, interfaces para redes de comunicação, ADC, etc...</li> </ul>

Desenvolveram-se dois protocolos de comunicação de modo a poder trocar dados sensoriais e de actuação entre o PC e as unidades *slave*, nomeadamente:

- A linha série RS-232 entre o PC e a unidade *Master*
- O CAN entre a unidade *Master* e os *Slaves*,

Entre os dados sensoriais podem-se enumerar (para um SCU):

- **Posição** dos três servomotores (em graus);
- **Velocidade** estimada de cada servomotor (em graus/s);
- **Corrente** consumida por cada servomotor;
- Valores dos **sensores de força** de cada pé (quatro sensores por pé);
- Saída dos **giroscópios** (em graus/s);
- Saída dos **inclinómetros** (em graus).

Em relação aos dados de actuação temos:

- **Posição referência** a atingir para cada componente (ângulos para os três servomotores ou as 3 componentes  $x$ ,  $y$ ,  $z$  do sistema de coordenadas cartesiana);
- **Velocidade** média para a realização da trajectória até à posição final;
- **Tipo de controlador** activo (centro de pressão, inclinação, ou velocidade angular);
- **Parâmetros de compensação** para o controlador activo ( $K$ ) e para o controlo local ( $K_P$ ,  $K_I$ );
- **Flags booleanas** de des/activação (ex.: PWMs).

Considera-se que o controlo de actuação deve ser feito de forma isolada a cada uma das três juntas (para cada unidade), e por isso as diversas juntas podem realizar movimentos com diferentes velocidades e diferentes parâmetros de compensação. Tal é útil no último caso, pois cada junta pode estar sujeita a diferentes esforços, e por isso, é rentável aplicar diferentes ganhos a cada uma.

Um dado importante a ter em conta, é a multiplicidade de significados que cada dado de actuação pode possuir, dependendo do tipo de controlador activo. Assim, para cada tipo de controlador, a posição referência diz respeito ao valor final que o respectivo controlador deve atingir. Quatro tipos de controladores de primeiro nível são implementados:

- **Sem controlador**: a posição referência diz respeito à posição angular a aplicar a cada junta, sendo directamente aplicada sobre o controlador local (ver abaixo);
- **Controlo de Centro de Pressão**: a posição referência diz respeito ao centro de pressão que a estrutura deve possuir no fim do movimento.
- **Controlo de Inclinação**: a posição referência especifica a inclinação que a estrutura deverá possuir.
- **Controlo de Velocidade Angular**: velocidade angular referência.

De igual modo, a velocidade indicada diz respeito ao controlador em causa, implementado-se, desta forma, trajectórias polinomiais que tanto podem ser de posição angular, centro de pressão, inclinação ou velocidade angular.

Com base nestes valores referência o controlador de primeiro nível determina as posições angulares a aplicar a cada servomotor, que posteriormente será entregue a um controlador local do tipo PI (Proporcional+Integrador) que se responsabilizará por garantir o alcance das posições solicitadas. O ajuste do controlador de primeiro nível é feito através do ganho  $K$ , enquanto que para o controlador local o ajuste é feito através dos ganhos  $K_I$  e  $K_P$ .

No que diz respeito aos dados sensoriais enviados à unidade *Master*, eles são de natureza constante e independentes do controlador activo.

## 2.2 COMUNICAÇÃO RS-232

### 2.2.1 DEVICE DRIVERS DE COMUNICAÇÃO NA UNIDADE PRINCIPAL

Para automatizar o processo de leitura/escrita dos dados sensoriais/actuadores, *device drivers* foram escritos na forma de funções em *MatLab*, tendo em vista o rápido desenvolvimento de rotinas que envolvam a leitura sensorial, processamento de dados, e consequente actuação para a realização de testes de avaliação de todo o sistema. A Tabela 2 apresenta os *device drivers* mais básicos para controlo e leitura directa dos actuadores e sensores respectivamente.

**Tabela 2: Lista de *device drivers* da unidade principal.**

<i>Ficheiro</i>	<i>Descrição</i>
<i>initcom.m</i>	Criação de uma nova ligação série entre a unidade central e a unidade Master.
<i>killcom.m</i>	Término da ligação série.
<i>testcom.m</i>	Envio de uma sequência de teste para a unidade Master, para avaliar a conectividade.
<i>readcanstat.m</i>	Consulta do estado do barramento CAN entre a unidade Master e os Slaves.
<i>readjoint.m</i>	Leitura dos sensores associados às juntas de um SCU.
<i>readspecial.m</i>	Leitura dos sensores adicionais conectados a um SCU.
<i>applyjoint.m</i>	Actuação nas juntas de um determinado SCU.
<i>applycontrol.m</i>	Seleção do algoritmo de compensação, e ajuste dos respectivos parâmetros de controlo para um SCU específico.

Outras funções estão também disponíveis, não sendo nada mais que operações de nível intermédio que fazem uso dos *device drivers* apresentados na secção anterior.

**Tabela 3: Device drivers adicionais.**

<i>Ficheiro</i>	<i>Descrição</i>
<i>inituc.m</i>	Activação do PWM de controlo sobre os actuadores de um SCU.
<i>killuc.m</i>	Desactivação do PWM para os actuadores de um SCU.
<i>waitpwmon.m</i>	Espera que o PWM seja activado nas três juntas de um SCU.
<i>sensor_reset.m</i>	Calibração dos sensores de um SCU.
<i>motionstart.m</i>	Espera que um SCU inicie a execução de uma trajectória.
<i>motionfin.m</i>	Espera que um SCU termine a execução de uma trajectória.
<i>waitmotionfin.m</i>	Espero que o processo de iniciação e término da trajectória seja executado.
<i>statmotionfin.m</i>	Consulta do estado da execução de trajectórias por parte de um SCU (se finalizou ou não).

## 2.2.2 PROTOCOLO RS-232

A comunicação RS-232 é aplicada entre o PC e a unidade *Master* e é efectuada assincronamente sendo orientada ao byte (1 *start* bit + 8 bits de dados + 1 *stop* bit), ou seja, é transmitido um byte de dados em cada transmissão/recepção. Pretende-se que numa única mensagem esteja contida toda a informação relativa a um parâmetro a ler/actuar nas três juntas de um SCU, o que implica que cada mensagem seja constituída por vários bytes. A estrutura das mensagens será apresentada a seguir.



*Estrutura de um pacote USART (transmissão/recepção de um byte)*

### Comandos PC→Master:

As mensagens no sentido PC→*master* são constituídas por seis bytes. O primeiro byte sinalizará a mensagem como sendo um comando de solicitação à unidade *master* – MESSAGE\_REQ (ver Tabela 4); o segundo byte conterá um código (*opcode*) indicativo da operação a realizar, do endereço do SCU alvo e de parâmetros adicionais; os três bytes seguintes conterão parâmetros a atribuir às três juntas no caso de um comando de actuação, e finalmente o byte BCC validará a integridade da mensagem.



*Mensagem USART PC→Master de actuação.*



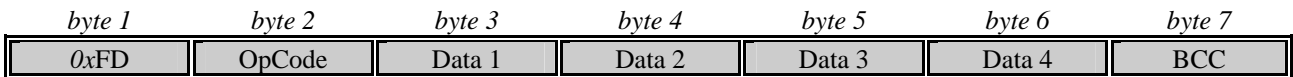
*Mensagem USART PC→Master de leitura sensorial.*

De notar, que a actuação é feita directamente às três juntas/componentes numa única mensagem, mas com a desvantagem de apenas se poder actuar num parâmetro cada vez (posição final, velocidade média ou um parâmetro do controlador).

### Respostas (aos comandos) Master→PC:

Na resposta, o *Master* responde com uma mensagem de 7 bytes, cuja estrutura é a seguinte:

- *SOF*: possui o valor MESSAGE\_SUCESS (0xFFB) no caso de uma resposta com sucesso;
- *OpCode*: *opcode* utilizado pela mensagem original PC→Master;
- *Data 1-4*: dados sensoriais no caso de um pedido de consulta sensorial;
- *BCC*: Validação da mensagem.



*Formato geral de uma mensagem de resposta Master→PC.*

Esta estrutura é geral e pode assumir várias formas de acordo com a operação envolvida:

- ➔ No caso de uma operação de actuação os bytes 2-5 possuem o mesmo valor que a mensagem original com o sexto byte nulo;
- ➔ Numa leitura sensorial, o byte *OpCode* é igual à da mensagem original com os bytes *Data 1-4* contendo a informação sensorial pedida. Se os dados sensoriais concernem aos servomotores, três bytes são utilizados para conter a informação relativa a cada um deles, e o último é utilizado para transmitir informação de *status* do SCU em causa. Se concernem aos sensores adicionais, são utilizados todos os quatro bytes para conter a informação de um dos conjuntos dos sensores – sensores de força (de um pé), inclinómetros ou giroscópios.

O formato das mensagens de resposta descrito é apresentado a seguir (== significa que o byte é o mesmo da mensagem de solicitação):

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFD	==	==	==	==	0	BCC

*Mensagem de actuação aplicada com sucesso*

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFD	==	Joint 1	Joint 2	Joint 3	Status*	BCC

*Mensagem de leitura sensorial das juntas(servomotores)*

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFD	==	Data 1	Data 2	Data 3	Data 4	BCC

*Mensagem de leitura dos sensores especiais*

Repare que, na leitura sensorial das juntas, na mesma mensagem é transportado os valores dos três servos. No entanto, apenas um parâmetro pode ser lido – posição, velocidade média ou corrente.

Quanto à leitura dos sensores adicionais, note também, a limitação de quatro valores, daí a limitação imposta anteriormente de apenas usar quatro linhas analógicas dedicadas a este género de sensores: ou o conjunto dos sensores de força, ou inclinómetros, ou giroscópios.

No entanto, podem ocorrer situações anómalas na recepção do comando por parte do *Master*, podendo-se discernir duas situações possíveis:

- ➔ Mensagem de pedido inválido: os parâmetros solicitados pelo PC não fazem sentido (ex.: SCU alvo não existente). Neste caso uma mensagem de SOF de código *MESSAGE\_INVREQ* (0xFC) (*invalid request*), com todos os restantes bytes iguais à da mensagem original, é retornada ao PC.
- ➔ Mensagem corrompida: o código BCC não está de acordo com a estrutura da mensagem. Neste caso uma mensagem de SOF igual a *MESSAGE\_INVALID* (0xFB) é retornado com os restantes bytes iguais à da mensagem recebida.

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFC	==	==	==	==	0	BCC

*Mensagem sinalizadora de um pedido inválido.*

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFB	==	==	==	==	0	BCC

*Mensagem sinalizadora de inconsistência no byte BCC.*

### Teste das comunicações USART entre PC e *Master*:

Para confirmação da correcta comunicação entre o PC e o *Master* pode ser enviada uma mensagem de teste igualmente de 6 bytes com o SOF como *MESSAGE\_TEST* (0xFE).

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
SOF: 0xFE	X2	X3	X4	X5	BCC

*Mensagem USART PC→Master de teste.*

Os bytes 2 a 5 podem possuir qualquer valor, desde que o byte BCC esteja de acordo com eles. Por sua vez, o *master* deverá responder com uma mensagem de 7 bytes possuindo os mesmos valores que a primeira:

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
SOF: 0xFE	X2	X3	X4	X5	0	BCC

*Resposta Master→PC a uma mensagem de teste.*

A Tabela 4 apresenta sintetizados os diversos tipos de mensagens apresentados nas secções anteriores, com a descrição do byte SOF utilizado para cada caso.



**Tabela 4: Tipos de mensagens USART (primeiro byte de cada frame).**

<i>Tipo de Mensagem</i>	<i>Designação</i>	<i>Código</i>	<i>Descrição</i>
Mensagem de solicitação	<i>MESSAGE_REQ</i>	<i>0xFF</i>	Solicitação de dados do PC para o <i>Master</i> .
Mensagem de teste	<i>MESSAGE_TEST</i>	<i>0xFE</i>	Envio de uma mensagem de teste das comunicações.
Mensagem de sucesso	<i>MESSAGE_SUCESS</i>	<i>0xFD</i>	Resposta a uma mensagem de solicitação ( <i>MESSAGE_REQ</i> ), indicando que o comando foi executado com sucesso.
Parâmetros inválidos	<i>MESSAGE_INVREQ</i>	<i>0xFC</i>	Pedido com parâmetros inválidos.
Mensagem inválida	<i>MESSAGE_INVALID</i>	<i>0xFB</i>	Mensagem de estrutura inválida (BCC incorrecto).

### 2.2.3 UNIDADE PRINCIPAL

#### Configuração da Unidade Principal

Como configurações gerais optou-se por efectuar as transacções à velocidade máxima permitida sem a aplicação de controlo de comunicações. Elas são:

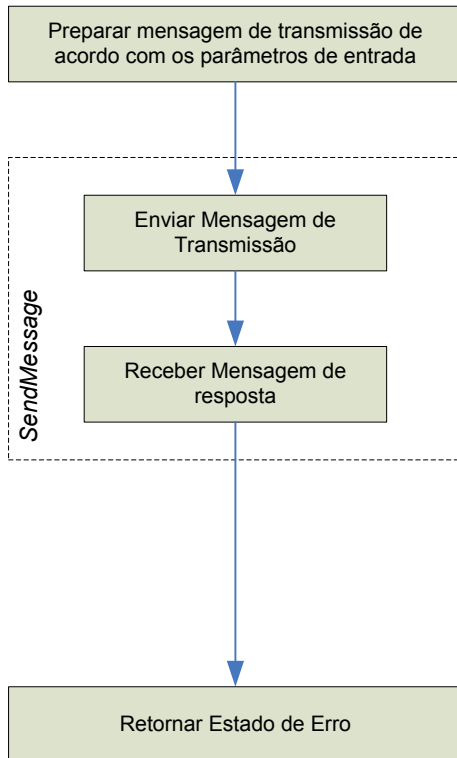
- *Baudrate*: 115200 bps
- Tamanho da palavra de dados: 8 bits
- Número de *stop bits*: 1
- Bit de paridade: desactivado
- Controlo de comunicações por *handshaking*: desactivado

Utilizou-se um computador Pentium com porta série RS-232 embutida, com o sistema operativo Microsoft® Windows XP, como unidade principal, para comunicar com a unidade *master*, embora seja perfeitamente compatível usar qualquer sistema desde que possua suporte a comunicações RS-232. O software utilizado foi o MatLab 7.0 através da mini-toolbox *cport* v1.3 que oferece *device-drivers* para troca directa por RS-232 tanto de caracteres como de *strings* e valores numéricos inteiros.

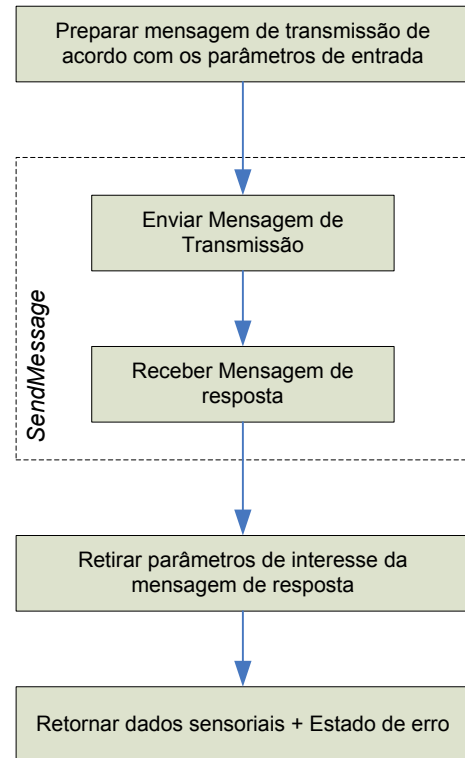
#### Algoritmos de Comunicação

As rotinas de comunicação básicas, já apresentadas na secção 2.2.1 (ver Tabela 2), constituem a forma de interface entre a unidade de controlo principal (normalmente um computador) e a unidade *master*. O seu algoritmo básico de funcionamento é apresentado na Fig. 2.4 e resume-se ao envio de uma mensagem à unidade *master* de acordo com os parâmetros de entrada fornecidos, retornando em seguida os dados sensoriais contidos na mensagem de resposta, caso seja o caso de um comando de leitura sensorial, mais o correspondente estado de erro (se ocorreu algum problema nas comunicações).

## ACTUAÇÃO

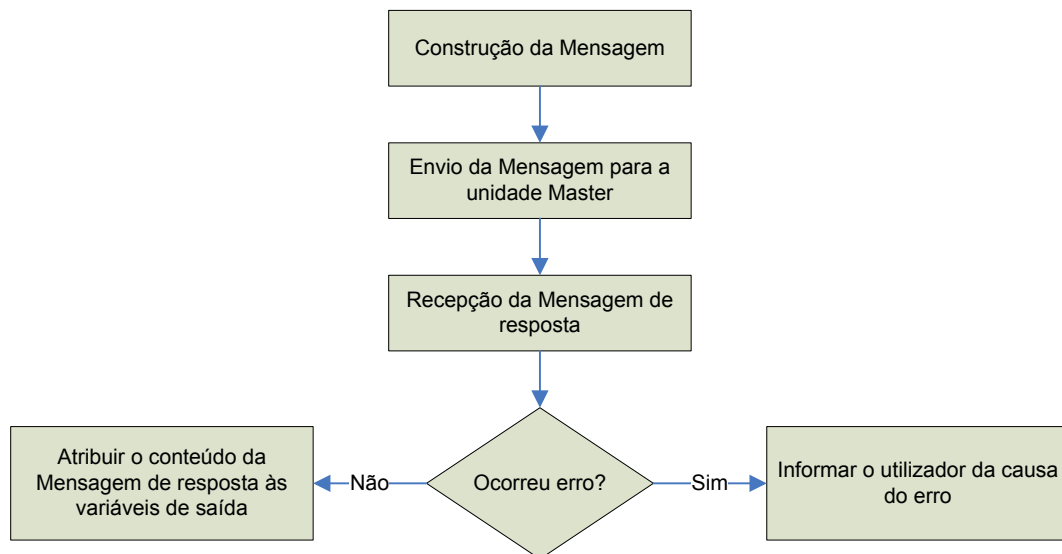


## LEITURA SENSORIAL

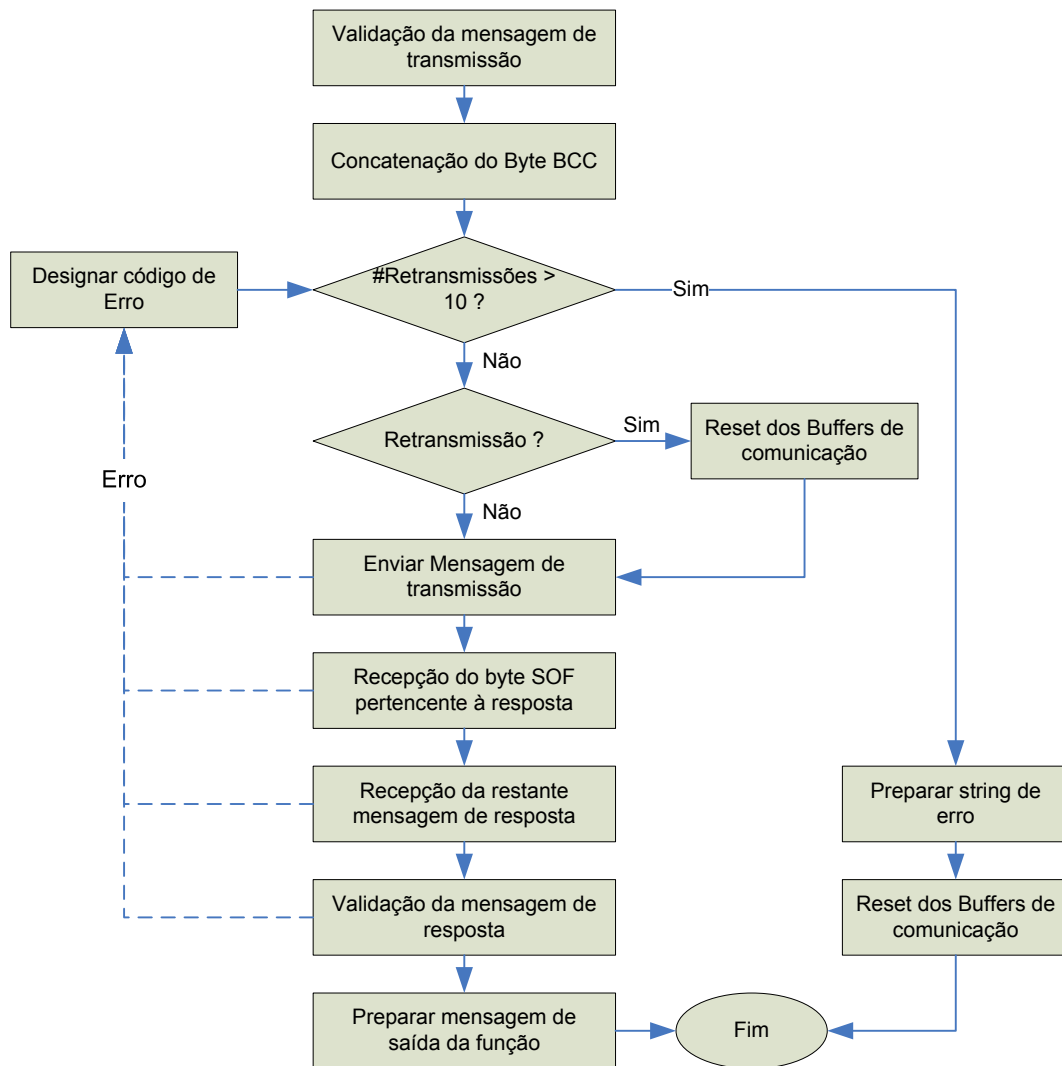


**Fig. 2.4:** Algoritmo básico dos *device drivers* de actuação / leitura sensorial.

Todos os *device drivers* apresentados utilizam uma função base que tem como responsabilidade a implementação do protocolo descrito na secção 2.2.2, ou seja, o envio de uma mensagem de 6 bytes à unidade *master* solicitando determinado dado sensorial ou com uma ordem de actuação, e depois o processamento da mensagem de resposta com a informação solicitada, ou confirmação da actuação, respectivamente. Esta função é denominada por *sendmessage*, e é constituída pelo conjunto de passos apresentado na Fig. 2.5 e Fig. 2.6



**Fig. 2.5:** Algoritmo básico da função *sendmessage*.



**Fig. 2.6: Algoritmo detalhado da função *sendmessage*.**

Um pormenor interessante que se pode extrair da Fig. 2.6 é a diferenciação dos passos de leitura do byte SOF e da restante mensagem de resposta. Observações experimentais indicam que por vezes a mensagem de resposta não se sincroniza com o envio do comando, começando a ocorrer deslocamentos (o primeiro byte recebido não corresponde ao primeiro byte da mensagem de resposta, mas sim a um byte intermédio da resposta interior. Para evitar este tipo de problemas, pesquisa-se durante algum tempo pelo byte SOF (de valor superior a *MESSAGE\_INVALID* – 0xFB) e só depois de recebido, lê-se a restante mensagem.

## 2.2.4 UNIDADE DISTRIBUIDORA MASTER

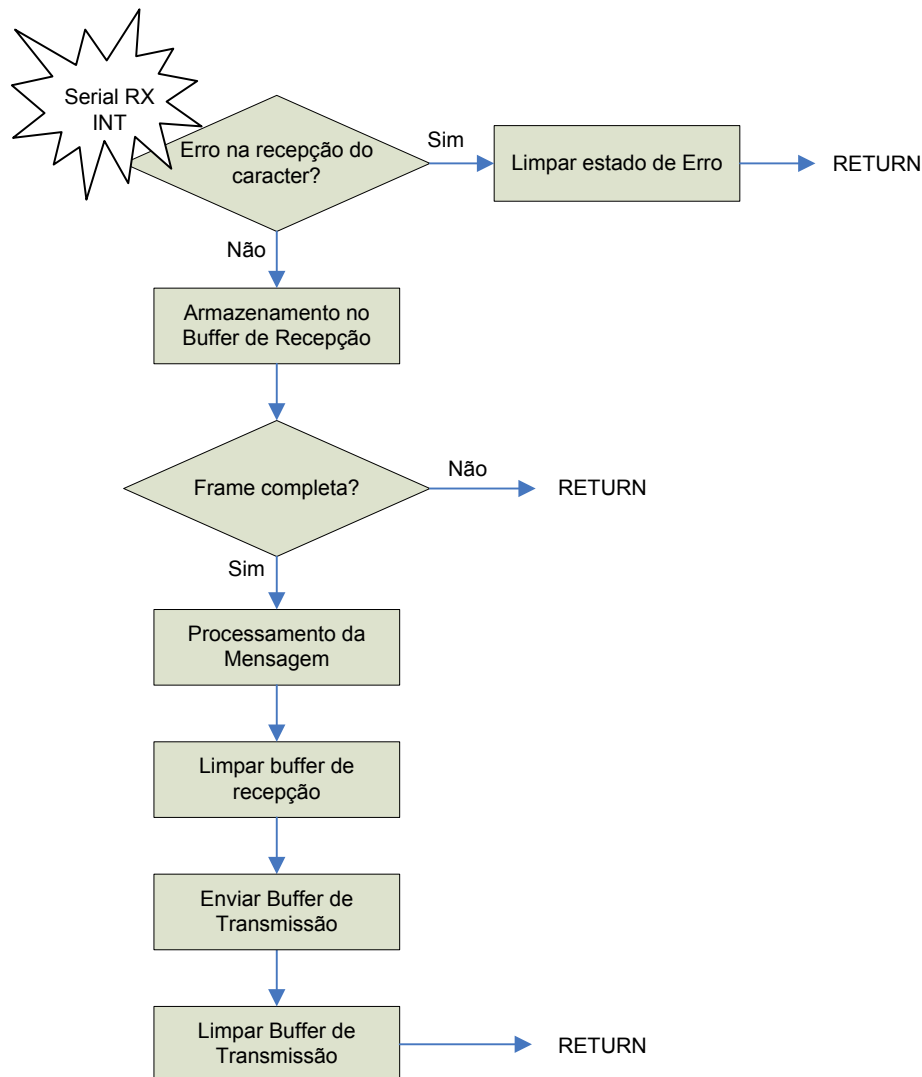
### Configuração da unidade Master

Para configurar a unidade microcontroladora a operar com a linha RS-232 é necessário efectuar um conjunto de passos que se passam a enunciar:

1. Configuração dos pinos I/O como entrada e saída referentes à recepção (RX) e à transmissão (TX) de dados respectivamente;
2. Definição do baudrate como 115200 bps;
3. Definição das restantes configurações como a ausência de paridade, apenas um *stop bit*, e a utilização de 8 bits para o campo de dados;
4. Activação do periférico UART;
5. Deinição das interrupções associadas.

### Algoritmos de Comunicação

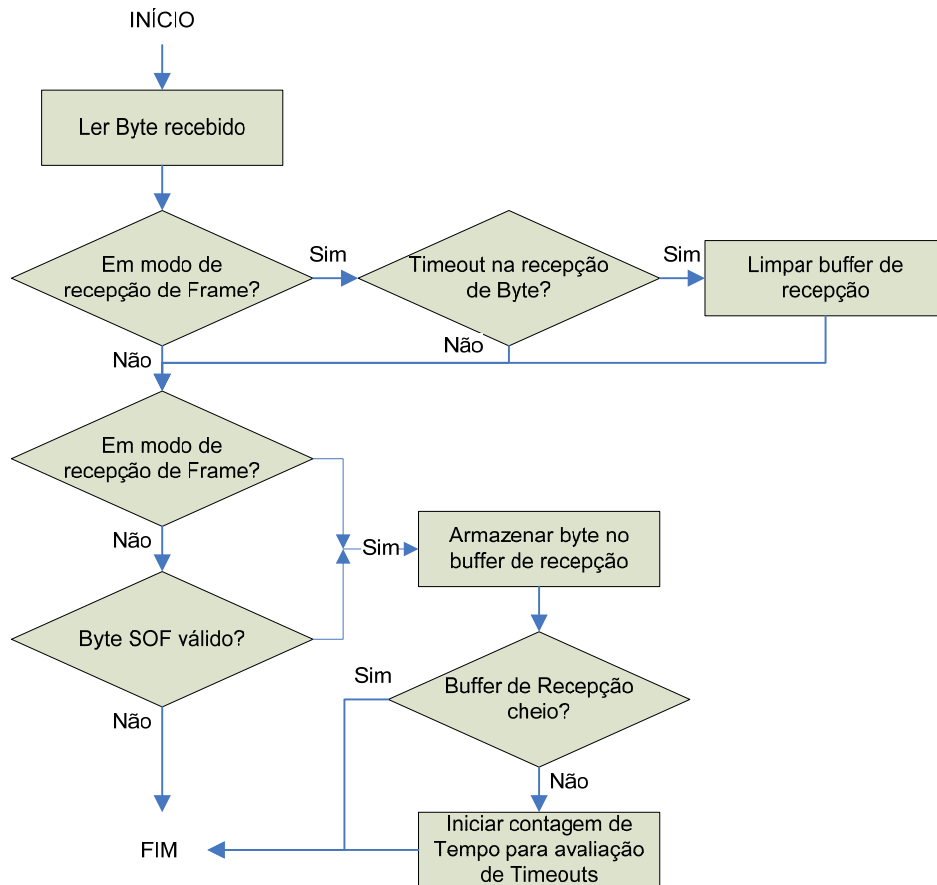
As Fig. 2.7 e Fig. 2.8 apresentam os algoritmos para troca de informação entre o *master* e a unidade principal.



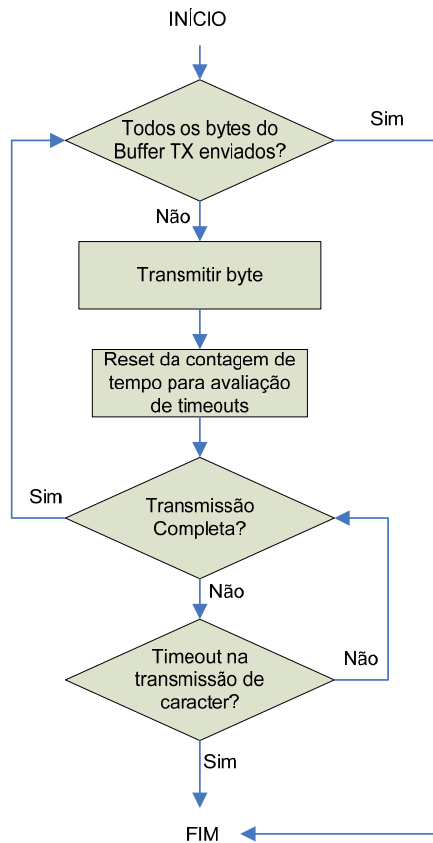
**Fig. 2.7: Algoritmo de gestão das comunicações RS-232 na unidade Master.**

Pela Fig. 2.7 podemos observar que o tratamento das mensagens é executado byte a byte a partir de interrupções de recepção pela linha série, com a utilização de *buffers* implementados em software para armazenar a mensagem recebida do PC (*buffer* de recepção) e a mensagem de resposta a enviar (*buffer* de transmissão). A implementação de cada um destes procedimentos está descrita na Fig. 2.8 e Fig. 2.9, correspondendo aos passos “armazenamento no *buffer* de recepção” e “enviar *buffer* de transmissão” apontados na Fig. 2.7.

Como detalhes importantes para a robustez deste algoritmo é a utilização de um *timer* para contagem de tempo, com vista a impor um tempo de *timeout* máximo para a conclusão de cada um dos procedimentos. Deste modo evitam-se bloqueios que poderiam comprometer o funcionamento do sistema.

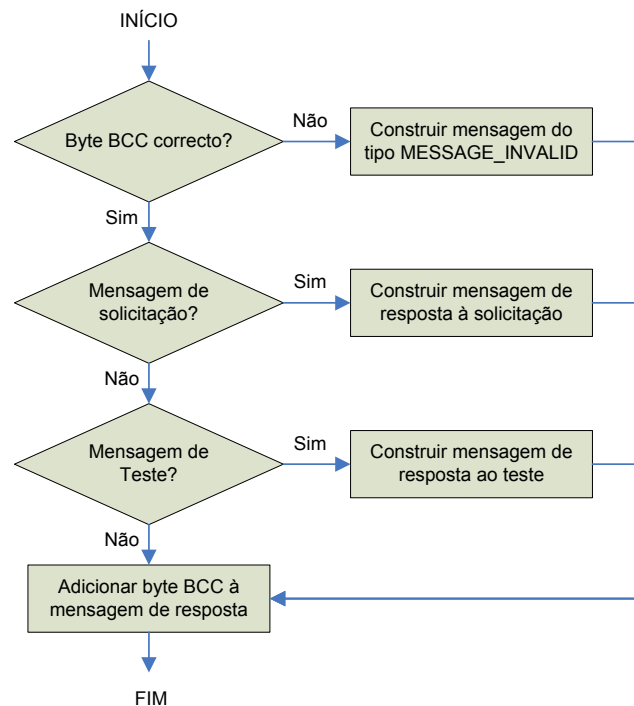


**Fig. 2.8: Recepção de byte por RS-232 e armazenamento no buffer de recepção.**



**Fig. 2.9: Envio do buffer de transmissão por RS-232.**

A função da tarefa “processamento da mensagem” apontada na Fig. 2.7 é a de ler a mensagem de solicitação presente no *buffer* de recepção, e construir a mensagem de resposta de acordo com o protocolo enunciado na secção 2.2.2, armazenando-a posteriormente no *buffer* de transmissão. O seu algoritmo é apresentado na Fig. 2.10, sendo capaz de validar a integridade da mensagem (por verificação do byte BCC) e construir as mensagens de resposta correspondentes a um comando de solicitação (actuação ou leitura sensorial) ou de teste das comunicações.



**Fig. 2.10: Processamento de uma mensagem na unidade Master.**

Um melhoramento importante a conferir a estes algoritmos, para evitar tempos mortos na largura de banda de CPU, sugere-se uma solução baseada na transmissão por interrupção, que em vez da transmissão dedicada a todos os bytes da mensagem de resposta, o CPU só deve ser utilizado para a transmissão de cada byte através da notificação por interrupção do envio do último byte. Desta forma, o período de tempo entre a transmissão de cada par de bytes pode ser utilizado para a execução de outras tarefas, como é o caso do serviço às comunicações CAN.

### **Base de Dados Global do Sistema Humanóide**

O processamento da mensagem de solicitação, proveniente da unidade principal, é baseado na análise do byte *opcode* inserido em cada *frame*. Uma vez decifrado o conteúdo da mensagem, uma base de dados global com a informação sensorial e de actuação de cada uma das unidades *slave* é acedida para consulta ou actualização, caso se trate de uma operação de consulta sensorial ou de actuação, respectivamente. A base de dados é dividida em dois conjuntos de dados principais: um para a informação sensorial e outro para a de actuação.

Esta base de dados, que pode ser visualizada na página seguinte, descreve o estado do sistema completo para todos os 8 *slaves*. A secção sensorial descreve o estado do sistema, as posições, velocidades estimadas e correntes drenadas fornecidas pelos potenciómetros dos três servomotores, bem como também a saída dos sensores que eventualmente possam estar conectados via *piggy-back* a cada unidade *slave*, como é o caso dos sensores de força e dos giroscópios.

Já a base de dados de actuação contém o estado de aplicação dos sinais de PWM sobre os servomotores e os parâmetros de compensação do controlador local e de primeiro nível, bem como os valores de referência dos mesmos compensadores.

Como já foi referido, cada comando de solicitação apenas consulta ou actualiza esta base de dados, não representando uma intervenção directa sobre as várias unidades *slave*. Tal procedimento permite que a troca de mensagens série opere à máxima velocidade, dado que não é necessário comunicar com mais nenhuma unidade. Quanto à entrega efectiva de dados entre *master* e *slaves*, é da responsabilidade da comunicação CAN que garante que os parâmetros de actuação são entregues às diversas unidades *slave* com um determinado ritmo periódico, bem como a actualização da base de dados com os dados sensoriais correctos para o instante.

Dado que o período de PWM aplicado aos servomotores é de 20ms (50 Hz), e que apenas tem interesse actualizar os parâmetros de actuação e sensoriais uma vez dentro deste período, para garantir alguma segurança na actualização dos mesmos, definiu-se um período de troca de informação *master-slave* de 1 ms, fazendo com que para 8 *slaves*, ao fim de 8 ms todas unidades locais possuam os seus parâmetros locais actualizados. Desta forma, garante-se que para cada período de PWM, a base de dados global no *master*, e a local em cada *slave* sejam actualizados duas vezes, de modo a incluir alguma redundância no caso de falha esporádica na entrega de mensagens.

### ***Estrutura sensorial***

```
// Estrutura descritiva dos sensores
typedef struct {
    byte sysStatus; // Estado do sistema
    struct_servo servo[N_SERVOS]; // Sensores dos servos
    unsigned char special[N_SPECIAL_SENSORS]; // Sensores de força dos pés
} struct_sensors;
#pragma varlocate 1 sensors
extern volatile struct_sensors sensors[N_SCU]; // Conjunto global dos sensores
```

## Estrutura de actuação

```
// Estrutura descritiva dos actuadores
typedef struct {
    // Estrutura de actuação de status para cada SCU
    struct {
        bool pwm; // PWM on/off
        bool calib; // Calib on/off
    } sysStatus;
    // Estrutura de Controlo para cada junta
    struct {
        byte ki, kp, kd;
        enum_controlType type;
    } control[N_SERVOS];
} struct_actuators;
#pragma varlocate 2 sensors
extern volatile struct_actuators actuators[N_SCU]; // Conjunto global dos actuadores

// Estrutura descritiva dos valores referência dos controladores
typedef struct {
    struct_paramControl servo[N_SERVOS];
    struct_paramControl cop[N_SERVOS];
    struct_paramControl inc[N_SERVOS];
    struct_paramControl giro[N_SERVOS];
} struct_refControl;
#pragma varlocate 3 sensors
extern volatile struct_refControl refControl[N_SCU];
```

## 2.3 COMUNICAÇÃO CAN

A comunicação CAN é implementada entre o *Master* e as diversas unidades *Slave* e tem como finalidade o redireccionamento dos dados de actuação, provenientes da unidade principal, para cada unidade *Slave*, e, no sentido oposto, o envio dos dados sensoriais para a unidade *Master*, de modo a que esta actualize a sua base de dados permitindo à unidade principal a sua consulta.

O CAN é um sistema de comunicações série multi-ponto que foi desenvolvido originalmente para a indústria automóvel para possibilitar as comunicações entre diversos componentes em ambientes extremamente ruidosos. Para tal, o sinal que serve de suporte à comunicação é definido em corrente e não em tensão. É um protocolo baseado na mensagem e não no endereço, o que significa que as mensagens são transmitidas na forma de *broadcasting* para todos os nós existentes na rede, cabendo a cada um a decisão de a aceitar ou não. As mensagens são constituídas por um identificador que pode possuir 11 ou 29 bits, de acordo com a versão 2.0A ou 2.0B respectivamente, e até 8 bytes de dados (Fig. 2.11).

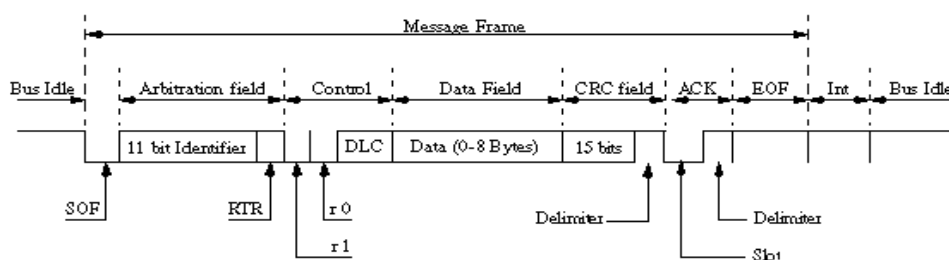


Fig. 2.11: Formato *standard* das mensagens CAN.

De modo a eliminar a hipótese de colisão destrutiva de pacotes (no caso de envio simultâneo de mensagens por diferentes nós) como acontece nas redes Ethernet, o CAN é dotado de um sistema de colisão determinístico bit a bit (CSMA/BA – Carrier Sense Multiple Access / Bit-wise Arbitration) que no caso de uma colisão de pacotes apenas um prevalece (o de maior prioridade), enquanto os restantes são destruídos. No último caso, os nós associados a esses pacotes devem tentar retransmitir mais tarde. Desta forma garante-se que a largura de banda oferecida pela rede não é desperdiçada, havendo sempre o transporte de mensagens desde que pelo menos um nó transmita.



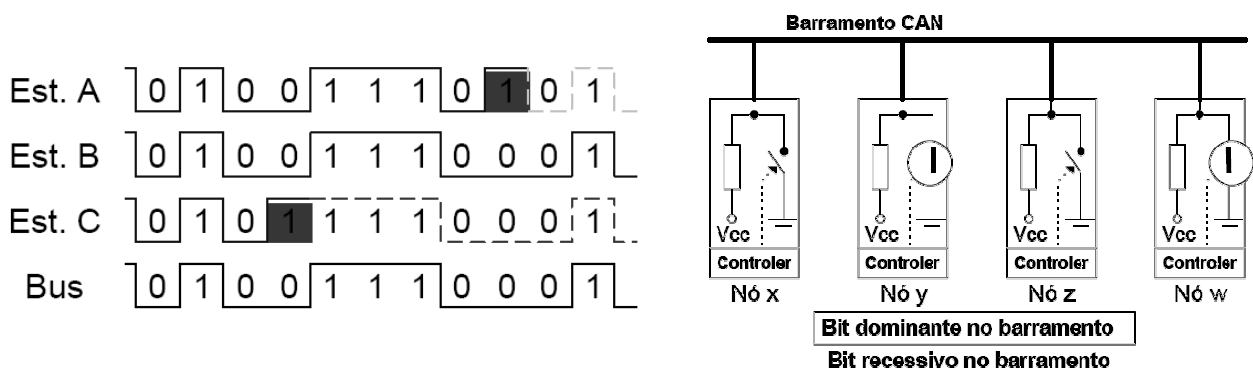


Fig. 2.12: Arbitragem de mensagens segundo o sistema de colisões CSMA/BA.

Para implementação deste modelo um dos bits é definido como dominante (bit 0) e o outro como recessivo (bit 1), de modo que quando dois bits colidem um com o outro “vence” o dominante. Esta estratégia é implementada bit a bit ao nível do identificador conferindo, deste modo, prioridades de acordo com o seu valor. Assim, quanto menor for o valor do identificador, maior é a prioridade da mensagem (Fig. 2.12). Segundo esta política é vantajoso atribuir um endereço de menor valor numérico à unidade *Master* de modo a conferir a máxima prioridade às mensagens de actuação *Master*→*Slave*, e às restantes unidades é-lhes atribuído endereços de valor crescente à medida que a sua ordem de importância na rede diminui. Desta forma, atribuiu-se o endereço *0b0000* ao *Master* e os endereços *0b0001* até *0b1000* para os oito *Slaves*, dando maior prioridade às unidades respeitantes aos membros inferiores dado que requerem maior atenção ao nível do controlo (Tabela 5).

<i>Unidade Controladora</i>	<i>Secção a que respeita</i>	<i>Endereço na Rede</i>
Master	Unidade mestre	<i>0b0000</i>
Slave 1	Perna direita	<i>0b0001</i>
Slave 2	Perna esquerda	<i>0b0010</i>
Slave 3	Anca direita	<i>0b0011</i>
Slave 4	Anca esquerda	<i>0b0100</i>
Slave 5	Tronco	<i>0b0101</i>
Slave 6	Braço direito	<i>0b0110</i>
Slave 7	Braço esquerdo	<i>0b0111</i>
Slave 8	Cabeça	<i>0b1000</i>

Tabela 5: Endereços atribuídos às diversas unidades de controlo.

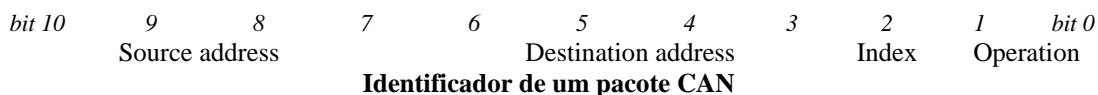
Outro benefício da comunicação baseada na mensagem é o facto de se poderem adicionar outros nós à rede sem haver necessidade de reprogramar todos os nós existentes. O novo nó adicionado receberá igualmente as mensagens que circulam na rede, decidindo por si só se deve ou não processar a informação nela contida. Esta foi uma das principais razões que motivaram a escolha do CAN, permitindo, desta forma que a rede possa ser modificada sem qualquer implicação ao nível do controlo local.

No microcontrolador em causa, é utilizada a versão *fullCAN 2.0A* cujas características principais são:

- Mensagens com um identificador de 11 bits e um máximo de 8 bytes de dados;
- Filtragem completa do conteúdo do identificador (todos os bits do identificador podem ser utilizados para configuração da filtragem);
- Associação do identificador a *buffers* de dados;
- Múltiplos *buffers* de transmissão/recepção: 3 de transmissão e 2 de recepção;
- Processamento e recuperação automático de erros.

## 2.3.1 PROTOCOLO CAN

De acordo com os dados a trocar entre o *Master* e cada *Slave*, e utilizando todos os 8 bytes de dados em cada mensagem, é possível efectuar trocas de informação, quer de actuação, quer sensoriais, recorrendo a apenas duas mensagens para cada instanciação (16 bytes). O protocolo é enunciado a seguir.



<i>Campo</i>	<i>Descrição</i>
<i>Source Address</i>	Endereço do nó emissor da mensagem
<i>Destination Address</i>	Endereço do nó destinatário da mensagem
<i>Index</i>	Índice do pacote dentro de uma mensagem: → 0: primeiro pacote; → 1: segundo pacote.
<i>Operation</i>	Operação associada à mensagem: → 0b00: actualização sensorial; → 0b01: actualização de actuadores.

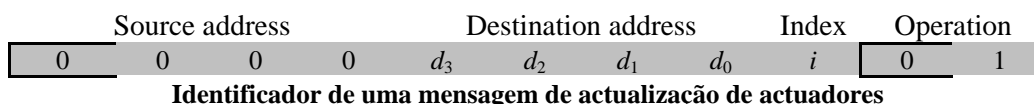
**Tabela 6: Campos do identificador de um pacote CAN.**

De notar que, do ponto de vista das unidades *Slave*, apenas o campo *destination address* é útil para a recepção de cada pacote, pois é ele quem define o seu destino. Já o campo *source address* está presente nos bits mais significativos com o propósito de definir a prioridade do pacote de acordo com endereço do remetente. Desta forma, mensagens enviadas pelo *master* possuem a máxima prioridade, e dentro dos *slaves*, os dos membros inferiores possuem maior prioridade de envio (ver Tabela 6). Como cada transacção é constituída por duas mensagens, o campo *index* identifica a sua ordem, e o campo *operation* refere a função dos dados transportados (*sensing/actuação*).

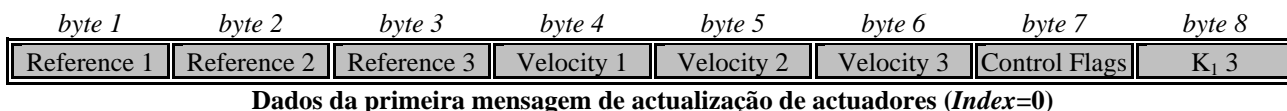
### Transacção Master→Slave

As mensagens trocadas do *master* para *slaves*, têm como função a actualização dos actuadores das juntas a que dizem respeito, pelo que transportará as seguintes informações para cada um dos três servomotores associados:

- Posição de referência a aplicar ao controlador de primeiro nível;
- Velocidade média do movimento;
- Flags de (des)activação dos controladores locais (ex: PWM);
- Parâmetros de compensação dos controladores local e de primeiro nível.



O identificador destas mensagens possuem o endereço 0b0000 no *source address* indicativo de que o pacote é proveniente da unidade *master*, e o campo *operation* contém o código 0b01 descrevendo que o pacote tem como intenção fazer uma actualização dos actuadores. O campo *destination address* identifica o SCU alvo.



byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
K <sub>I</sub> 1	K <sub>I</sub> 2	K <sub>I</sub> 3	K <sub>P</sub> 1	K <sub>P</sub> 2	K <sub>P</sub> 3	K <sub>I</sub> 1	K <sub>I</sub> 2

**Dados da segunda mensagem de actualização de actuadores (Index=1)**

Cada transacção é constituída por duas mensagens CAN em que tanto a posição referência final (*Reference i*), como a velocidade média (*Velocity i*) e os parâmetros de compensação ( $K_i + K_j$ ) para cada servomotor são actualizadas de uma só vez.

O byte *Control Flags* tem como função indicar qual o tipo de controlador de primeiro nível (associado aos parâmetros  $K_i$ ) a estar activo em cada um dos servomotores e se os sinais de controlo dos motores (PWM) devem estar activos (valores booleanos).

bit 7	6	5	4	3	2	1	bit 0
PWM	—	Controller 3		Controller 2		Controller 1	

**Byte Control Flags da primeira mensagem**

**Transacção Slave→Master**

Por outro lado é necessário actualizar a base de dados na unidade *master* pelo envio de mensagens CAN com os dados sensoriais de cada *slave*. Tal é executado logo após a recepção do pacote de actuação e com o retorno das seguintes informações:

- Posição angular dos servomotores;
- Velocidade estimada dos servos;
- Corrente consumida;
- Output dos sensores conectados via *piggy-back*;

Source address				Destination address				Index	Operation	
s <sub>3</sub>	s <sub>2</sub>	s <sub>1</sub>	s <sub>0</sub>	0	0	0	0	i	0	0

**Identificador de uma mensagem de actualização sensorial**

Os pacotes de actualização sensorial devem possuir no campo *destination address* o endereço da unidade *master* (0b0000), dado que a mensagem se destina a ele, e o campo *operation* a 0b00 indicativo de que a mensagem contém dados sensoriais. O *source address* contém o endereço do *slave* respectivo.

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
Sensor Flags	Position 1	Position 2	Position 3	Velocity 1	Velocity 2	Velocity 3	Slave Status

**Dados do primeiro pacote de actualização sensorial (Index=0)**

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
Sensor Flags	Current 1	Current 2	Current 3	Special 1	Special 2	Special 3	Special 4

**Dados do segundo pacote de actualização sensorial (Index=1)**

Os valores de saída dos sensores *piggy-back* estão descritos nos campos *Special* apenas podendo fazer parte apenas de uma das seguintes classes:

- Sensores de força de um dos pés (4 sensores);
- Inclínómetros e/ou Giroscópios (2 + 2 valores);

No entanto é indiferente de onde provém estes dados, dado que a unidade principal (PC) conhece *a priori* o género de sensores especiais conectados a cada unidade *slave*.

O byte *sensor flags* é puramente redundante, dado que a informação que contém também é transportado no identificador do pacote, e poderá ser usado no futuro para conter mais dados sensoriais.

bit 7	6	5	4	3	2	1	bit 0
Source Address				Index	0	0	0

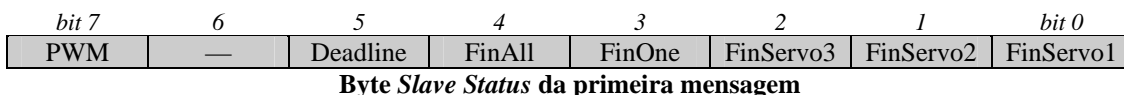
**Byte Sensor Flags**

<i>Campo</i>	<i>Descrição</i>
<i>Source Address</i>	Endereço do slave remetente.
<i>Índice</i>	Índice do pacote na mensagem de dados.

**Tabela 7: Campos do byte *Sensor Flags*.**

Já o byte *slave status* contém *flags* indicadoras do estado de funcionamento da unidade *slave*, informando se a unidade finalizou o movimento relativamente a cada uma das juntas (*FinServo i*), se pelo menos uma junta finalizou (*FinOne*), e se todas as juntas finalizaram (*FinAll*). PWM indica se o PWM está activo em todos os três servomotores, e a *flag deadline* apenas se destina para efeitos de *debugging*.

Estas informações são bastante úteis para a unidade principal gestora dos algoritmos de locomoção, servindo como marcações indicando se o movimento terminou para dar início a novas trajectórias.



## 2.3.2 FUNCIONAMENTO DO BARRAMENTO CAN

### Configuração CAN

#### *Inicializações*

As definições de funcionamento do CAN devem ser iguais entre a unidade *Master* e as unidades *Slave* de modo a permitir o correcto sincronismo entre todas as entidades envolvidas na rede, pelo que as opções tomadas devem ser implementadas em todas as unidades.

Como condições base seguiram-se as seguintes linhas:

- ✓ Escolha da versão CAN mais universal: *fullCAN 2.0A*;
- ✓ Configuração da velocidade na máxima possível (1 Mbps);
- ✓ Filtragem das mensagens apenas destinadas ao próprio SCU – apenas as mensagens com o endereço destino igual ao do próprio são aceites;
- ✓ Dadas as transacções serem constituídas por duas mensagens, aproveitou-se a presença de dois *buffers* de recepção para redireccionar cada uma para um *buffer* à parte – dupla filtragem;
- ✓ Igualmente, utilizaram-se dois *buffers* de transmissão para transmitir cada uma das duas mensagens.

A seguir são descritas as principais operações na configuração deste recurso.

#### *Modo de Configuração*

O PIC 18F258 possui seis modos de operação, dos quais apenas 3 nos são úteis:

- Modo *disable*: o PIC não transmite nem recebe quaisquer mensagens;
- Modo de configuração: necessário antes de activar as transmissões/recepções;
- Modo de operação normal: recepção e transmissão de mensagens válidas, via CAN.

Antes de colocar o PIC a processar mensagens via CAN, é necessário antes de mais, colocar o PIC em modo de configuração e definir os seus parâmetros. Apenas depois deste passo coloca-se o PIC no modo de operação normal.

#### *Baudrate*

Todos os nós na rede CAN devem ter o mesmo *bitrate* nominal (número de bits transmitidos por segundo) sem no entanto ser necessário que todos os nós tenham a mesma frequência de oscilação. No entanto é necessário o cuidado em programar o *baudrate prescaler* e o número de *time quanta* (explicado adiante) em cada segmento de bit, de modo a manter o *bitrate* nominal. O *bitrate* máximo depende da qualidade do transmissor e do oscilador, e da presença de resincronizações, mas poderá atingir o valor máximo de 1 Mbps numa situação ideal.

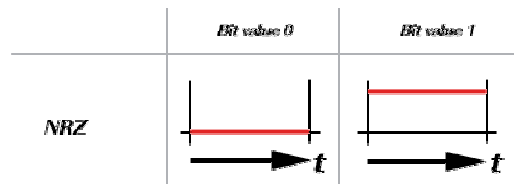


Fig. 2.13: Codificação NRZ.

Dado que a transmissão utiliza codificação NRZ (non return to zero) (Fig. 2.13) e que os osciladores e transmissores podem variar de nó para nó, é necessário introduzir *bit stuffing* para possibilitar a extracção de relógio para efeitos de sincronização. Tal é feito de modo a garantir uma alternância de bit pelo menos de 6 em 6 bits.

Uma unidade denominada por DPLL é utilizada para a sincronização dos dados recebidos e para garantir o *bitrate* nominal nos dados transmitidos. Funções de *bus timing* executadas dentro de cada *bit frame*, como a sincronização com o oscilador local, compensação do atraso introduzido pela rede e posicionamento de amostragem, implicam a particionamento de cada bit em vários segmentos definidos a partir de períodos de tempo mínimos chamados *Time Quanta* ( $T_Q$ ).

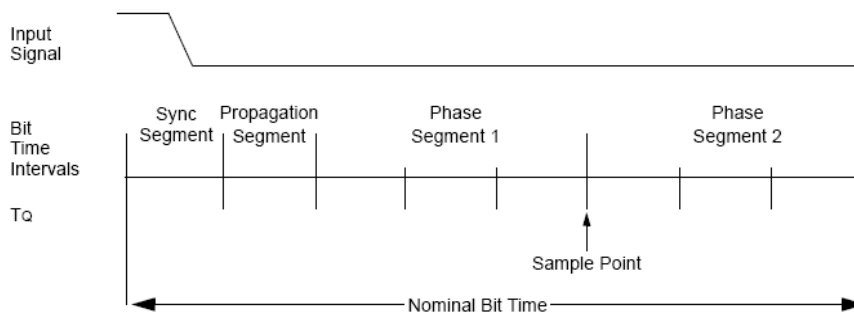


Fig. 2.14: Particionamento temporal de um bit.

Pela Fig. 2.14 podemos identificar 4 tipos de segmento:

1. Segmento de sincronização: utilizado para sincronizar os vários nós da rede – duração fixa de  $1 T_Q$ ;
2. Segmento de propagação: utilizado para compensar os atrasos introduzidos na rede – duração entre 1 e  $8 T_Q$ .
3. Segmento de fase 1: número de  $T_Q$  antes da amostragem do bit – duração entre 1 e  $8 T_Q$ .
4. Segmento de fase 2: fornece um atraso antes da mensagem seguinte – duração de 2 a  $8 T_Q$ .

A duração dos segmentos de fase 1 e 2 não são parâmetros estáticos e podem sofrer variações para efeitos de resincronização. O parâmetro SJW (*Synchronization Jump Width*) define a forma como se farão estas flutuações contendo o número máximo de  $T_Q$  que serão adicionados ao segmento de 1 ou subtraídos ao segmento 2.

Para definição das durações de cada segmento algumas regras deverão ser respeitadas. Elas são:

- ✓  $1 + \text{Seg. de propagação} + \text{Seg. de fase 1} + \text{Seg. de fase 2} \geq 8$
- ✓  $\text{Seg. de propagação} + \text{Seg. de fase 1} \geq \text{Seg. de fase 2}$
- ✓  $\text{Seg. de fase 2} \geq \text{SJW}$

Para o nosso caso, de forma a conseguir a máxima velocidade, definimos definiram-se os seguintes parâmetros:

- Segmento de sincronização:  $1 T_Q$  (não configurável);
- Segmento de propagação:  $3 T_Q$ ;
- Segmento de fase 1:  $3 T_Q$ ;
- Segmento de fase 2:  $3 T_Q$ ;
- SJW:  $2 T_Q$ .

Com base nestes parâmetros, e de modo a conseguir a máxima velocidade de  $F_{bit}=1\text{Mbps}$ , o *baudrate* a definir no registo BRP será de 1.

$$BRP = \frac{2 * F_{CPU}}{F_{bit} * N_{seg}} - 1, \text{ com } N_{seg} = Sync_{seg} + Propag_{seg} + Phase_1 + Phase_2$$

$$F_{bit} = \frac{2 * F_{CPU}}{(BRP + 1) * N_{seg}} = \frac{2 * 10 \text{ MHz}}{(1 + 1) * 10} = 1 \text{ Mbps}$$

### Máscaras e Filtros

Os filtros são utilizados para automatizar o processo de aceitação de mensagens. No caso das mensagens de actuação, apenas se destinam a um SCU particular não possuindo qualquer interesse para os restantes. Desta forma, na perspectiva de cada SCU, apenas algumas das muitas mensagens que passam pelo seu porto de entrada devem ser processadas, pelo que se a verificação do endereço destino fosse feita por *software*, a largura de banda de CPU atribuída a este processo seria tanto mais pesada quanto maior fosse o número de unidades *slave* na rede.

De modo a automatizar o processo de aceitação, uma máscara pode ser utilizada para indicar quais os bits do identificador que contém o padrão de interesse para a unidade em questão (como por exemplo o endereço destino) e filtros podem ser utilizados para fazer a selecção dos padrões e redireccionar para determinados *buffers*. Desta forma, apenas as mensagens cujos bits da secção do identificador programado coincide com o padrão especificado, são aceites e carregados para os *buffers* de recepção gerando uma interrupção para o processamento posterior.

Máscara do bit n	Filtro do bit n	Valor do bit n	Resultado
0	X	X	Rejeitado
1	0	0	Aceite
1	0	1	Rejeitado
1	1	0	Rejeitado
1	1	1	Aceite

Tabela 8: Resultado da filtragem para cada bit.

Pela Tabela 8 podemos verificar que apenas os bits definidos a 1 pela máscara serão considerados para filtragem. Todos os outros serão rejeitados *à priori*. Dos que são considerados para filtragem são comparados com um determinado padrão sendo apenas aceites os que coincidem com esse padrão. Apenas as mensagens em que todos os bits do filtro são aceites, são consideradas para processamento por parte do programador. Para o PIC 18F258 é possível utilizar todos os bits do identificador na máscara e filtragem e definir até dois padrões de filtragem para armazenamento imediato no primeiro *buffer* de recepção e até 4 padrões para armazenamento no segundo *buffer* (Fig. 2.15).

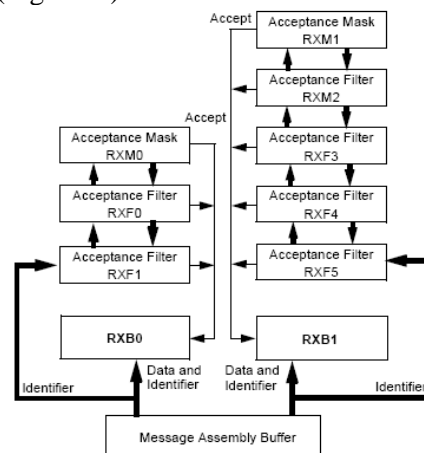


Fig. 2.15: Registos associados à máscara e filtragem.

Para o nosso caso em concreto, de acordo, com a estrutura do identificador enunciado anteriormente, a máscara a definir cobrirá os bits do endereço destinatário e do *index* do pacote, ou seja, possuirá o seguinte padrão:

bit 10	9	8	7	6	5	4	3	2	1	bit 0
Source address				Destination address				Index	Operation	
0	0	0	0	1	1	1	1	1	0	0

**Máscara a aplicar ao identificador**

De seguida definir-se-ão dois filtros, de modo a aceitar unicamente as mensagens cujo endereço destino corresponda ao próprio SCU, e destes, redireccionar os de *index* 0 (primeira mensagem) para o *buffer* de recepção 0, e os de *index* 1 (segunda mensagem) para o *buffer* de recepção 1 (Tabela 9).

<i>Redirecção dos pacotes</i>	<i>Registos a configurar</i>	<i>Padrão</i>
<i>Buffer 0</i>	RXF0	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 0 0 0
	RXF1	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 0 0 0
	RXF2	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 1 0 0
<i>Buffer 1</i>	RXF3	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 1 0 0
	RXF4	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 1 0 0
	RXF5	0 0 0 0 a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> 1 0 0

**Tabela 9: Configuração dos filtros para redireccionamento de pacotes para os dois buffers de recepção (padrão a<sub>3</sub> a<sub>2</sub> a<sub>1</sub> a<sub>0</sub> = endereço do SCU).**

Desta forma, redireccionando cada uma das duas mensagens para um *buffer* à parte, evita-se a perda de informação caso ocorra *overflow*, pois apenas o *buffer* cuja informação é equivalente à nova mensagem (*index* igual) é substituído.

### **Transmissão e Recepção de Mensagens**

Finalmente, é necessário configurar a forma como as mensagens serão transmitidas e recebidas. Para a recepção, é necessário definir a recepção de apenas pacotes válidos cujo identificador segue o formato da versão CAN 2.0A, ou seja de 11 bits, e deve-se assegurar que o *overflow* do *buffer* 0 para o *buffer* 1 está desactivado, dado que estamos a usar um *buffer* para cada índice de mensagem. Deve-se ainda colocar cada um dos dois buffers de recepção prontos para recepção, verificando também se o *interrupt flag* associado está desligado.

Para a transmissão, os identificadores das mensagens deverão estar de acordo com o formato *standard*, ou seja, deverão possuir 11 bits de comprimento, e é necessário definir prioridades para cada um dos três *buffers*, para definição da ordem de transmissão no caso de vários agendamentos simultâneos.

Por questões de similaridade de comportamento, também se reservará um *buffer* de transmissão para cada tipo de mensagem, fazendo apenas uso de dois *buffers*. O *buffer* 0 utilizado para as mensagens de *index* 0 possuirá a máxima prioridade e o *buffer* 1 possuirá a prioridade mínima (Tabela 10).

<i>Buffer</i>	<i>Pacotes associados</i>	<i>Prioridade</i>	<i>Registos</i>
<i>Buffer</i> 0	<i>Index</i> 0	Máxima	TXB0CON=0b11
<i>Buffer</i> 1	<i>Index</i> 1	Mínima	TXB1CON=0b10
<i>Buffer</i> 2	—	—	TXB2CON=0b01

**Tabela 10: Atribuição de prioridades entre cada buffer de transmissão.**

### ***Modo de Operação Normal***

Para colocar o CAN a funcionar só resta definir o modo de operação normal através do registo próprio para o efeito.

#### ***Resumindo o processo:***

1. Colocação do PIC no modo de configuração;
2. Definição da duração de cada segmento de bit;
3. Definição do *bitrate* para a velocidade pretendida;
4. Configuração da máscara de recepção;
5. Configuração dos dois filtros para cada um dos *buffers* de recepção;
6. Definição do formato *standard* para os identificadores dos pacotes a transmitir;
7. Definição de prioridades entre *buffers* de transmissão;
8. Configuração da recepção para apenas pacotes válidos com identificadores de 11 bits;
9. *Overflow* do *buffer* 0 para o *buffer* 1 desactivado;
10. *Clear* dos *interrupt flags* de recepção;
11. Abertura dos *buffers* para recepção;
12. Colocação do PIC no modo de operação normal.

(Os passos 6 a 11 devem ser aplicados a cada um dos buffers de recepção/transmissão.)

### **Troca de Mensagens CAN**

Esta secção apresenta os procedimentos a efectuar nos PICs 18F258 para a troca de pacotes utilizando a rede CAN. Estes microcontroladores oferecem uma gama diversificada de registos que permitem controlar diversos aspectos, como por exemplo o comprimento do identificador ou o conjunto de dados e, também, no domínio da recuperação de erros.

Na recepção de um pacote que passe a máscara e um dos dois filtros, o seu conteúdo é carregado no *buffer* associado ao seu *index* n. Os procedimentos a seguir descritos devem-se aplicar relativamente a esse *buffer*:

1. Verificação da recepção de uma mensagem inválida;
2. Confirmação da utilização do *buffer* n;
3. Reposição do *interrupt flag* associado ao *buffer* n a zero;
4. Transferência do *buffer* n para o *Acess Bank Area*, de modo a se poder aceder a ele a partir de qualquer banco de memória. Para tal deve-se definir a janela de endereçamento de acordo com o *buffer* em uso (*window address bits*);
5. Verificação da ocorrência de *overflow*;
6. Leitura do identificador do pacote;
7. Validação do comprimento de dados recebidos;
8. Leitura dos dados;
9. Libertação do *buffer* n para possibilitar a recepção de uma futura mensagem;
10. Reposição da janela de endereçamento para o seu valor original;

Transmissão de um pacote usando o *buffer* n:

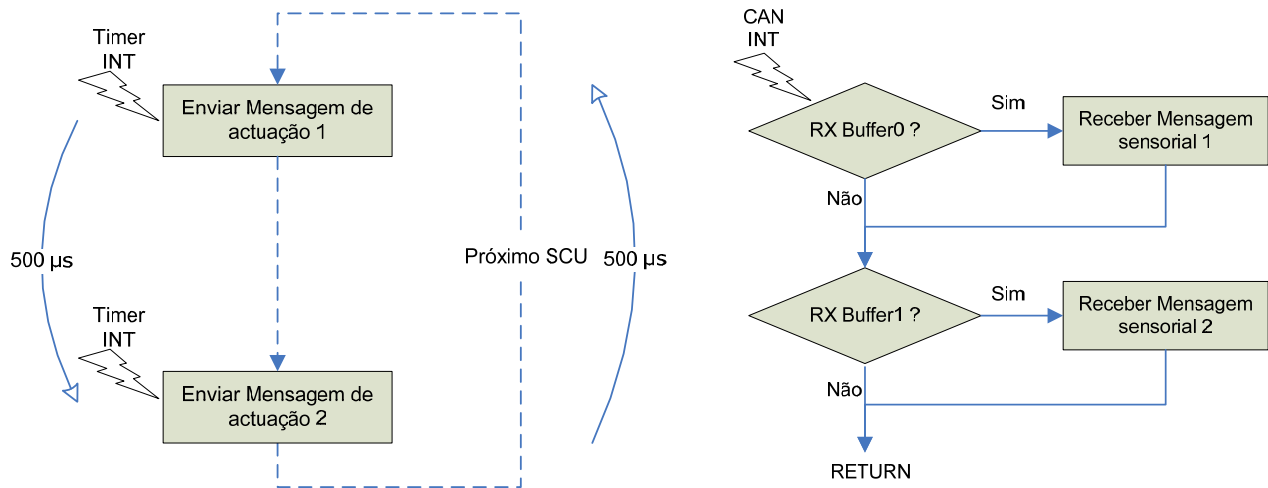
1. Verificar se o *buffer* n está pronto para transmissão;
2. *Clear* do *interrupt flag* associado ao *buffer* n;
3. Definição da janela de endereçamento para o *buffer* de transmissão n;
4. Definição do identificador;
5. Definição do comprimento do pacote (8 bytes);
6. Definição do conjunto de dados a enviar;
7. Activação da transmissão do *buffer* n;
8. Reposição da janela de endereçamento para o seu valor original;



### 2.3.3 UNIDADE MASTER

#### Algoritmo de Troca de Mensagens

As trocas de informação *Master-Slave* são desencadeadas a partir da unidade *Master*, estando os *slaves* programados para responder “imediatamente” após a recepção das mensagens oriundas do *Master*. A Fig. 2.16 apresenta os algoritmos utilizados na unidade *master* para gerir a troca de informação.



**Fig. 2.16: Algoritmos de transmissão/recepção de mensagens CAN no Master.**

As transmissões de mensagens de actuação são desencadeadas a partir de um *timer* programado para gerar uma interrupção de 500 em 500µs. Como duas mensagens tem de ser transmitidas a cada slave, cada um é servido durante 1 ms, correspondendo a um tempo de *turn-around* de 8 ms para 8 slaves.

Relativamente às recepções das mensagens sensoriais provenientes dos *slaves*, são processadas através de interrupções CAN segundo a política *event-triggered*: sempre que uma nova mensagem chega, uma interrupção CAN é gerada, e quando possível, é processada actualizando a base de dados sensorial.

Estas duas fontes de interrupções foram definidas como sendo de baixa prioridade, relativamente às interrupções da linha série (UART), pelo que o serviço CAN só pode ser atendido quando não há comunicação UART. Desta forma, garante-se uma satisfatória velocidade de resposta aos comandos enviados pelo PC, sem comprometer significativamente a performance CAN para o período de transmissão citado.

#### Transmissão e Recepção de Mensagens

Do ponto de vista do *Master*, todas as mensagens a enviar são de actuação, e todas as recebidas são de carácter sensorial, pelo que, no último caso, é escusado o processamento do identificador, excepto para o conhecimento do SCU fonte e do *index* do pacote. No caso do endereço destino e da operação apenas são verificados para efeitos de verificação do correcto funcionamento.

As mensagens de actuação a enviar vão buscar os valores a aplicar nos *slaves* à base de dados global que o PC periodicamente actualiza (por comunicação série). Já as mensagens de carácter sensorial que vão chegando dos *slaves*, actualizam constantemente as variáveis da mesma base de dados. É este mecanismo que garante a periódica actualização da base de dados sensorial e das unidades *slave* no que toca à actuação.

Procedimento de envio de uma mensagem de actuação:

1. Construção do identificador com base no endereço destino e no *index* do pacote (endereço origem e operação conhecidos);
2. Construir o *array* de dados a enviar (8 bytes), a partir da base de dados de actuação, de acordo com o *index* da mensagem;
3. Enviar o pacote (identificador + *array* de dados) pelo *buffer* de transmissão adequado ao *index*;
4. Actualização do estado de erro, se ocorreu algum.

Recepção de um pacote de carácter sensorial:

1. Obtenção do identificador e do *array* de dados do pacote recebido;
2. Actualização da base de dados sensorial, a partir do endereço origem e do *index* do pacote;
3. Actualização do estado de erro, se ocorreu algum.

## 2.3.4 UNIDADES SLAVE

### Algoritmo de Troca de Mensagens

Nas unidades *slave*, a comunicação CAN é efectuada segundo a política *event-triggered*, dado que apenas é accionada quando são recebidas mensagens do *master*. A Fig. 2.17 evidencia o algoritmo presente em cada unidade *slave*. Quando uma mensagem é recebida no *buffer* 0 (primeira mensagem de actuação) uma interrupção é gerada, e na Rotina de Serviço à Interrupção (RSI) os dados de actuação são actualizados e uma mensagem com informação sensorial é enviada (primeira mensagem de resposta). Por sua vez, se a interrupção corresponder à chegada de uma mensagem no *buffer* de recepção 1, o processo é equivalente mas respeitante à segunda mensagem de cada transacção. De notar ainda, que as interrupções CAN são de baixa prioridade, para não permitir interferência com as tarefas essenciais de controlo de PWM sobre os motores.

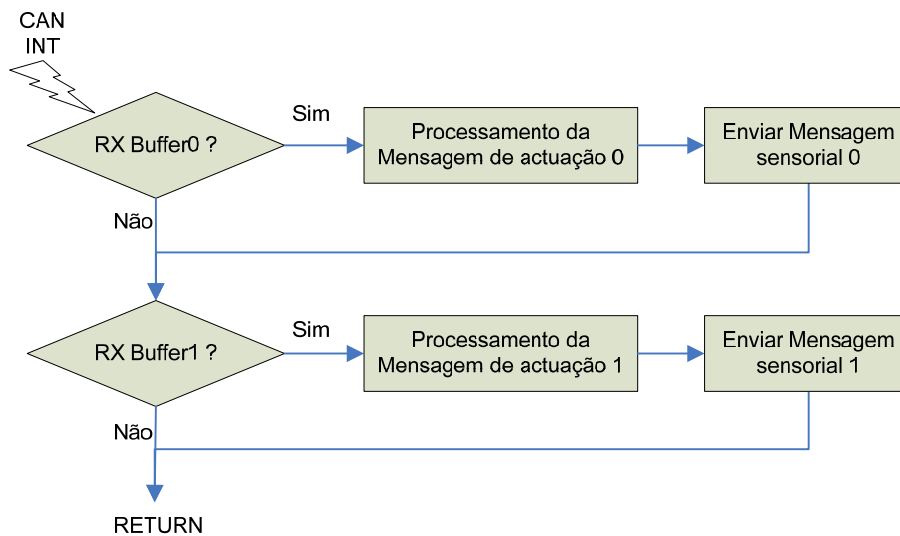


Fig. 2.17: Algoritmo de troca de informação pelo CAN no Slave.

### Transmissão e Recepção de Mensagens

Do ponto de vista de cada *slave*, todas as mensagens a enviar são de carácter sensorial, e todas as recebidas são de actuação, pelo que, no último caso, é escusado o processamento do identificador, excepto para o conhecimento do *index* da mensagem. O endereço origem (MCU), endereço destino (o próprio) e da operação apenas são verificados para efeitos de teste do correcto funcionamento.

Envio de uma mensagem de carácter sensorial:

1. Construção do identificador com base no *index* da mensagem (endereço origem, destino e operação conhecidos);
2. Construir o *array* de dados a enviar de acordo com o *index* da mensagem, a partir dos dados sensoriais obtidos directamente;
3. Enviar a mensagem (identificador + *array* de dados) pelo *buffer* de transmissão adequado ao *index*.

Recepção de um pacote de actuação:

1. Obtenção do identificador e do *array* de dados da mensagem recebida;
2. Actualização dos actuadores de acordo com o *index* da mensagem.

## **Base de Dados Local**

Como as novas informações de actuação nunca são aplicadas imediatamente, torna-se necessário armazenar estes valores numa base de dados local para posterior acesso pelas tarefas de controlo local. O mesmo se passa com a informação sensorial que quando é gerada tem que ser armazenada para que a tarefa de comunicação CAN possa acedê-la mais tarde para o envio das mensagens de resposta.

Esta base de dados local é muito semelhante à base de dados apresentada na unidade *master*, mas agora com a diferença de ser de carácter local, ou seja, apenas armazena a informação sensorial ou de actuação de um único SCU – o próprio.

```
// Tipo enumerado do género de controlador implementado em cada junta
typedef enum {
    NO_CONTROL    = 0b00,           // Funcionamento em malha aberta
    COP_CONTROL   = 0b01,           // Controlo de equilíbrio no tronco
    INC_CONTROL    = 0b10,           // Controlo das forças de reacção nos pés
    GIRO_CONTROL  = 0b11           // Controlo de posição e velocidade
} enum_controlType;

// Estrutura descritiva do estado de um servo
typedef struct {
    signed char position;           // Posição referêcia
    signed char velocity;          // Velocidade
    unsigned char current;         // Corrente consumida
} struct_servo;
```

## ***Base de dados sensorial***

```
// Estrutura descritiva dos sensores
typedef struct {
    struct {
        bool pwm;                  // Motores ligados/desligados
        bool calib;                // Calibração ligada/desligada
        bool deadlineError;        // Violação de deadline
        bool motionFinAll;         // Todos os motores terminaram o movimento
        bool motionFinOne;        // Um dos motores terminaram o movimento
        bool motionFin[N_SERVOS]; // Movimento terminado
    } sysStatus;                  // Estado sensorial do sistema
    struct_servo servo[N_SERVOS]; // Sensores dos servos
    unsigned char special[N_SPECIAL_SENSORS]; // Sensores especiais
} struct_sensors;
extern volatile struct_sensors sensors; // Sensores
```

## ***Base de dados de actuação***

```
// Estrutura descritiva dos actuadores
typedef struct {
    // Estrutura de actualização de status para cada SCU
    struct {
        bool pwm;                  // PWM on/off
        bool calib;                // Dynamic calibration on/off
    } sysStatus;                  // Estado da actuação do sistema
    struct {
        byte kp, kd, ki;           // Ganhos dos controladores
        byte type;                 // Tipo de Controlador de primeiro nivel
    } control[N_SERVOS];         // Controlador
    struct_servo servo[N_SERVOS]; // Informação de actuação
    // (posição, velocidade e corrente)
} struct_actuators;
extern volatile struct_actuators actuators; // Actuadores
```

## 2.4 DISCUSSÃO DOS RESULTADOS

A arquitectura de comunicações adoptada para a estrutura humanóide revelou-se bastante simples de implementar seguindo uma lógica bastante intuitiva. Esta arquitectura segue um formato distribuído, com a associação de cada trio de juntas (normalmente associadas a um membro físico) a uma unidade de controlo local baseada num PIC que se responsabiliza em aplicar os sinais de controlo e ler a informação sensorial. Estas unidades de controlo interligam-se umas às outras por meio de um único barramento que partilha as mensagens em circulação a todas as unidades de controlo. Este barramento também se interliga a uma unidade mestre (*master*) que por sua vez está conectada por ligação série à unidade principal, neste caso um PC, que é responsável por enviar os comandos de actuação e de leitura sensorial a cada unidade local. Deste modo, cada unidade tem uma função específica, deslocalizando assim, da unidade principal, todas as tarefas de controlo dos servomotores e de distribuição das mensagens entre os diversos *slaves*.

Implementando os protocolos de comunicação, tanto série entre o PC e o *master* como o CAN entre o *master* e os *slaves*, seguindo a metodologia de funcionamento enunciada, a fiabilidade das comunicações revelou-se bastante boa com a troca de mensagens à velocidade máxima possível e sem a detecção de corrupção de dados ou de atrasos excessivos na sua entrega.

Como interface, utiliza-se o *software MatLab* na unidade principal (PC) para fazer a interface uma vez que oferece uma linguagem de alto nível e intuitiva para programação das sequências de comandos a aplicar nas juntas e para monitorização dos sinais de saída.

## Capítulo 3

# Sistema de Controlo de Baixo-Nível

### 3.1 INTRODUÇÃO

Como já foi mencionado, o robot humanóide é constituído por oito unidades de controlo local, que denominámos por *slaves*, que podem controlar até 3 graus de liberdade dos 22 existentes na arquitectura actual, e que se interligam a um computador (unidade principal), através de uma unidade *master*, para o envio de comandos de actuação com base na informação sensorial medida (Fig. 3.1).

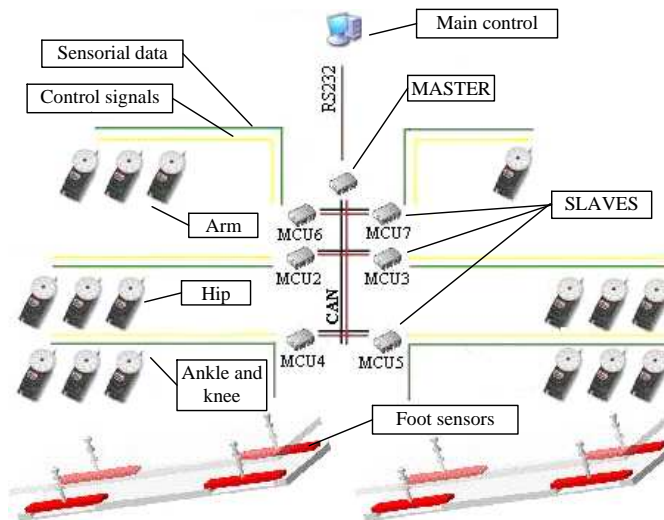


Fig. 3.1: Arquitectura da plataforma humanóide.

Como actuador para cada grau de liberdade são utilizados servomotores, que não são nada mais que motores DC adaptados para efectuar controlo de posição. Como vantagens, estes motores permitem-nos fazer medidas da sua posição e da corrente consumida o que nos possibilitará mais adiante fazer controlo externo para correcção de eventuais desvios ao comportamento ideal.

Para além do controlo sobre as juntas, existe um conjunto de sensores adicionais destinados ao equilíbrio da plataforma:

- Quatro sensores de força aplicados directamente sobre a base de cada pé, implementados a partir de extensómetros (resistências que variam o seu valor de acordo com a sua deformação).
- Inclinómetros para medição da verticalidade do tronco. Estes inclinómetros são basicamente acelerómetros que medem a aceleração da gravidade nos seus dois eixos ortogonais – na posição vertical o vector gravidade deve coincidir com o seu eixo vertical.

- Giroscópios para medição da velocidade angular em certos pontos do corpo.

A presente dissertação apenas irá cobrir a utilização dos sensores de força nos pés, no que respeita ao equilíbrio estático e dinâmico da estrutura. Este assunto será discutido em mais detalhe no Capítulo 4.

Neste capítulo apenas serão discutidos os actuadores e o seu controlo de forma a permitir:

- Fazer o controlo de posição de modo a garantir que o actuador atinge sempre a posição solicitada;
- Fazer o controlo de velocidade, variando o seu valor de acordo com as necessidades;
- Que os movimentos das juntas sejam os mais suaves possíveis, sem acelerações bruscas nem velocidades muito elevadas;

Embora os servomotores tenham sido escolhidos por possuírem um controlador interno de posição, contrariamente aos outros géneros de motores, veremos adiante que apresenta muitos problemas na concretização dos objectivos enunciados, principalmente na presença de elevadas cargas no seu eixo. Para tal, serão apresentadas algumas soluções já testadas de modo a resolver estes desvios, mais concretamente pela utilização de compensação via *firmware* nos módulos.

## 3.2 CONTROLO DA PLATAFORMA HUMANÓIDE

### 3.2.1 A UNIDADE DE CONTROLO LOCAL

A Fig. 3.2 apresenta a constituição de uma unidade de controlo local (*slave*) com o microcontrolador PIC18F258 da *Microchip* e a electrónica de interface para os servomotores e os diversos sensores. Uma foto dele pode ser visualizada na Fig. 2.2.

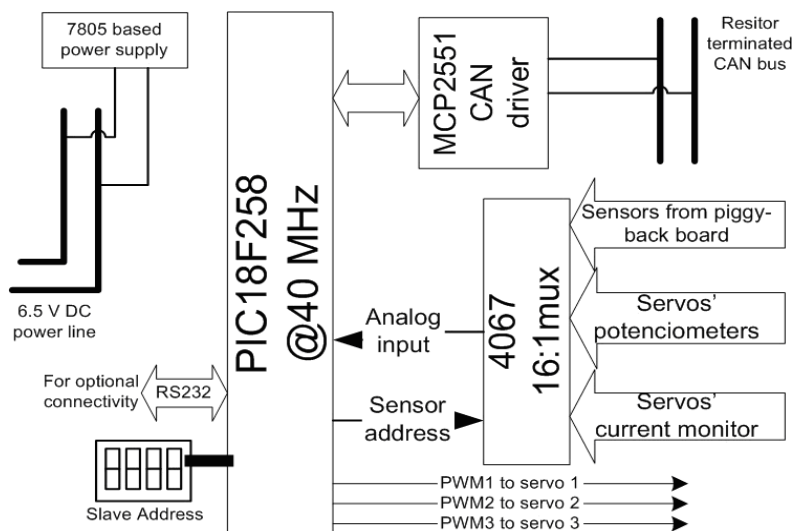


Fig. 3.2: Constituição de uma unidade *slave*.

Como se pode observar, cada *slave*, possui uma interface para o barramento CAN constituído pelo *driver* do barramento CAN MCP2551, um módulo UART para comunicação via RS-232, e um *DIP switch* que permite, via *hardware*, configurar o endereço da unidade (*slave address*). Para a actuação, é possível controlar até três servomotores com a disponibilidade de um pino dedicado a cada um para aplicação directa de um sinal modulado em *duty-cycle* para o controlo de posição (PWM). Adicionalmente, um *multiplexer* de 16 canais é utilizado para fazer a aquisição sensorial dos servomotores e dos sensores inerciais. A Fig. 3.3 descreve os compontes apresentados no próprio módulo.

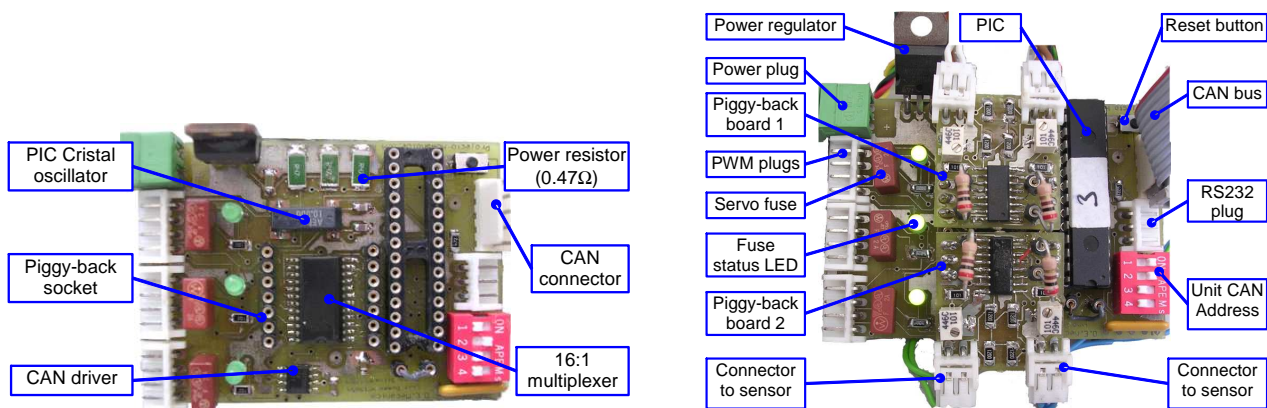


Fig. 3.3: Imagens de uma unidade de controlo local.

### 3.2.2 ACTUADORES: OS SERVOMOTORES

Para a actuação sobre as juntas é essencial tanto o controlo de posição como de velocidade, e dado que, em média, cada junta não possui uma excursão de movimento superior a 180°, uma solução baseada em servomotores foi a mais imediata (Fig. 3.4). Podem-se enunciar as seguintes vantagens e desvantagens para esta escolha:

- ✓ Excursão de posição de 180°;
- ✓ Controlador de posição incluído;
- ✓ Dispositivo relativamente pequeno e compacto;
- ✓ Relativamente barato;
- \* Não oferece externamente a possibilidade de controlo de velocidade.



Fig. 3.4: Servomotor da HITEC.

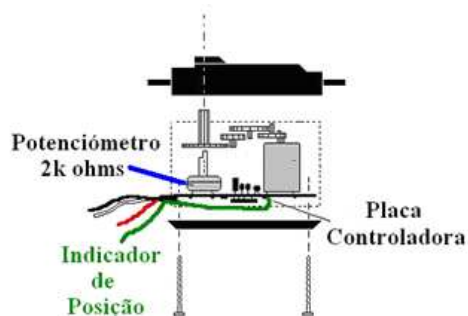
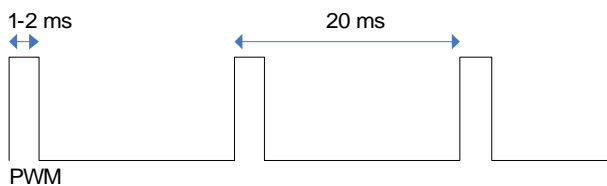


Fig. 3.5: Representação do interior de um servomotor.

Olhando para o interior deste actuador (Fig. 3.5) podemos discernir os seguintes componentes:

- Motor DC;
- Caixa redutora;
- Mecanismo de *feedback* da posição angular do motor (potenciómetro);
- Electrónica de controlo do motor a partir de um sinal digital;
- Electrónica de controlo de posição.

Pode-se, por isso, concluir que estes servos, não são nada mais do que simples motores DC que incluem electrónica interna responsável por implementar o controlo de posição em malha fechada a partir de um sinal externo de referência. Este sinal externo define a posição a ser atingida, e é definido através de uma onda quadrada modulada em largura de impulso – PWM (*Pulse Width Modulation*) –, cuja largura define a posição final. Desta forma, a posição do eixo do servo é controlada a partir do *duty-cycle* de uma onda quadrada de formato digital (Fig. 3.6).



**Fig. 3.6: Sinal de PWM aplicado no servomotor.**

Spec	Values	
Control system	Pulse Width Control 1.5 ms neutral	
Voltage range	4.8V to 6.0V	
Teat voltage	@ 4.8V	@ 6.0V
Speed (no load)	60%/0.19 s	60%/0.14 s
Stall torque	1.94 Nm	2.42 Nm
Operating angle	45° /one side pulse traveling 400μs	
Direction	clockwise/pulse traveling 1.5 to 1.9 ms	
Current drain	8mA (idle); 700mA (no load running)	
Dead bandwidth	8 μs	
Dimensions	66 x 30 x 57.6 mm	
Weight	152g	

**Tabela 11: Especificações do servo da HITEC HS-805BB.**

Para cada modelo de servomotor este sinal deve assumir determinadas especificações, mas para um dos modelos utilizados (HITEC HS-805BB<sup>1</sup>) estas devem ser as características do sinal de PWM:

- Frequência de 50 Hz (Período de 20 ms);
- *Duty-cycle* variável entre 1 (0°) e 2 ms (180°).

Mais dados podem ser consultados na Tabela 11 com indicações do *duty-cycle* correspondente à posição central (90°) (*control system*), gama das tensões do sinal de PWM (*voltage range*), em que se observa que aceita sinais digitais de 5V provenientes do PIC. Outros dados também estão presentes como é o caso do binário máximo, corrente consumida, etc.

A escolha do modelo HITEC HS-805BB para as juntas mais exigentes, recaiu essencialmente pelo seu binário máximo de 2.42 N.m. Simulações em CATIA (software de simulação de estruturas mecânicas) da plataforma humanóide realizando um passo (ano lectivo 2004/05) demonstram que no pior cenário as juntas podem estar submetidas a binários de cerca de 2.6 N.m (Tabela 12). O modelo HS-805BB destacou-se por possuir o máximo binário da gama disponível, que mesmo assim, mostra ser insuficiente para o nosso caso. Para resolver este problema, introduziram-se correias de transmissão que multiplicam o binário máximo pela relação entre número de dentes do eixo do servo e da junta. Nas juntas dos pés e dos joelhos utilizaram-se relações de transmissão de 2:1 e de 2.2:1 assegurando um binário máximo de 5.3 N.m assumindo como 2.42 o binário máximo do servomotor. Nas juntas das ancas aumentou-se a margem para 1:3.75 dada a sua exigente natureza (máximo de 9.1 N.m).

Motor / Junta	$\Theta_1$ [°]	$T_1$ [N.m]	$\Theta_2$ [°]	$T_2$ [N.m]	$\Theta_3$ [°]	$T_3$ [N.m]
Pé 1 lateral	0.0	<b>2.37</b>	7.1	0.98	7.1	0.96
Pé 1 frente	4.7	0.30	10.1	0.20	10.1	0.04
Joelho 1	10.1	0.76	21.8	1.17	21.8	1.01
Anca 1 frente	5.4	0.35	11.7	0.30	11.7	0.14
Anca 1 lateral	0.0	<b>2.26</b>	7.1	<b>2.57</b>	7.1	<b>2.55</b>
Pé 2 lateral	0.0	0.00	7.1	0.00	7.1	0.00
Pé 2 frente	4.7	0.12	10.1	0.12	16.4	0.12
Joelho 2	10.1	0.17	21.8	0.23	41.9	0.30
Anca 2 frente	5.4	0.07	11.7	0.02	25.6	0.14
Anca 2 lateral	0.0	0.01	7.1	0.30	7.1	0.29

**Tabela 12: Binários exigidos na simulação de um passo.**

<sup>1</sup> Estes dispositivos podem ser adquiridos no site <http://www.maxxprod.com>.



$$N = \frac{\text{N.º dentes da polia da junta}}{\text{N.º de dentes do eixo do servo}}$$

$$\tau_{max} = N * \tau_{servo}$$



Fig. 3.7: Correia de transmissão aplicada a um servo.

Olhando para o interior do servo na tentativa de perceber melhor a sua electrónica interna, a Fig. 3.8 apresenta o circuito de controlo de um servo semelhante aos da HITEC – o FUTABA S3003 – cujo sinal de PWM (*input pulse*) apresenta as mesmas especificações que as descritas na Tabela 11. Estes dispositivos utilizam como *feedback* tanto o sinal de posição como de velocidade que, além de fornecer um controlo preciso de posição, permite estabilizá-lo de modo a conduzi-lo para a posição desejada com o mínimo de oscilação.

Em funcionamento normal, o sinal de PWM da entrada é comparado com o sinal resultante de um “gerador de impulso linear” (*linear pulse generator*) controlado a partir da posição obtida pelo potenciómetro e da velocidade medida a partir da força contra-electromotriz do motor (tensão gerada entre impulsos de potência). O sinal gerado é da mesma forma que o de entrada e, para baixas velocidades, a sua largura de impulso deve corresponder à posição efectiva do motor.

A diferença de largura de impulso entre estes dois sinais, produzido pelo “comparador de largura de impulso” (*pulse width comparator*), conhecido como sinal de erro, é em seguida amplificada através de um “amplificador de largura de impulso” (*pulse stretcher*) que é depois aplicado numa “ponte H” (BAL6686 na Fig. 3.8) que controla o motor. Facilmente se percebe que quando a largura de impulso do sinal de entrada é igual à resultante pelo gerador de impulso (erro nulo) a diferença é de largura nula e nenhum sinal é aplicado ao motor deixando-o em repouso – note que o motor em si é controlado em velocidade pelo que se nenhum sinal for aplicado ele tende para o repouso.

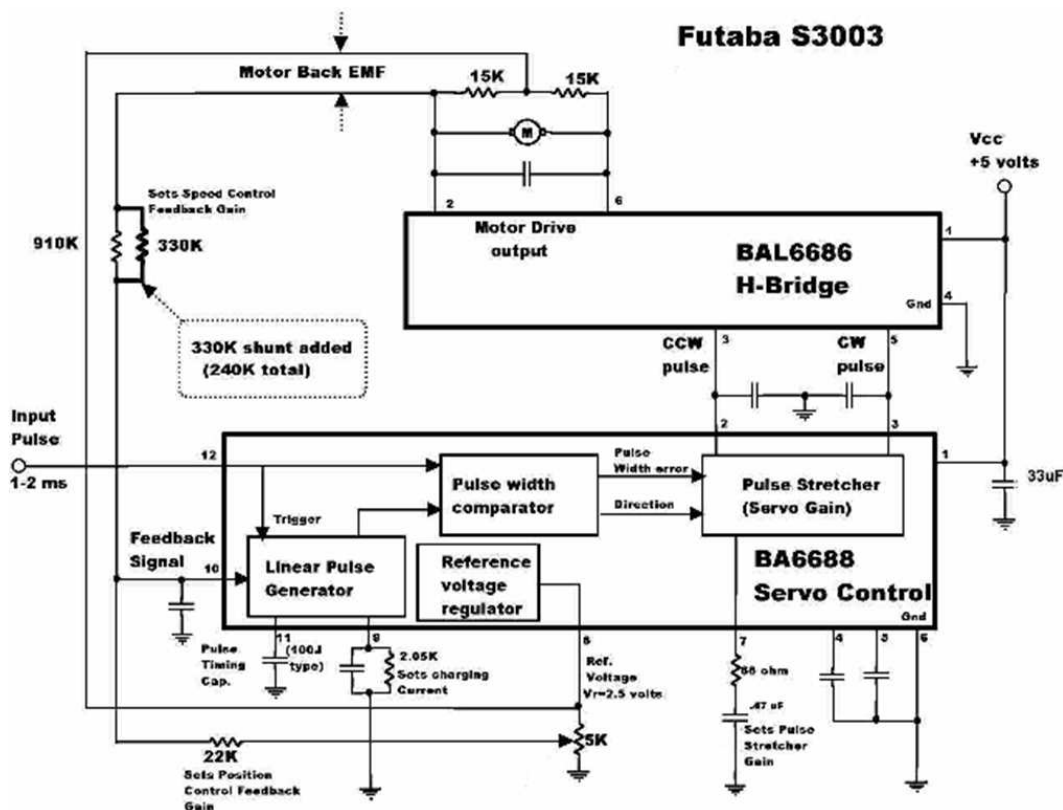


Fig. 3.8: Circuito do controlador de posição de um FUTABA S3003.

Representando matematicamente as operações envolvidas, verificamos que:

- À posição medida pelo potenciômetro  $p(t)$  é aplicado um ganho  $K_P$  representado pela resistência de  $22K\Omega$  que é somada à velocidade resultante da força contra-electromotriz multiplicado pelo ganho  $K_D$  representado pelo paralelo das resistências de  $910K$  e de  $330K\Omega$ . Este é o sinal de *feedback*:

$$F(s) = K_P \cdot P(s) + K_D \cdot V(s)$$

$$F(s) = (K_P + s \cdot K_D) \cdot P(s)$$

- O resultado da soma  $f(t)$  é introduzido no “gerador de impulso linear” que produz um sinal de PWM para comparação (em termos de largura de impulso) com o sinal de entrada de referência  $r(t)$  através do “comparador de largura de impulso” produzindo o sinal de erro  $e(t)$  também do formato PWM.

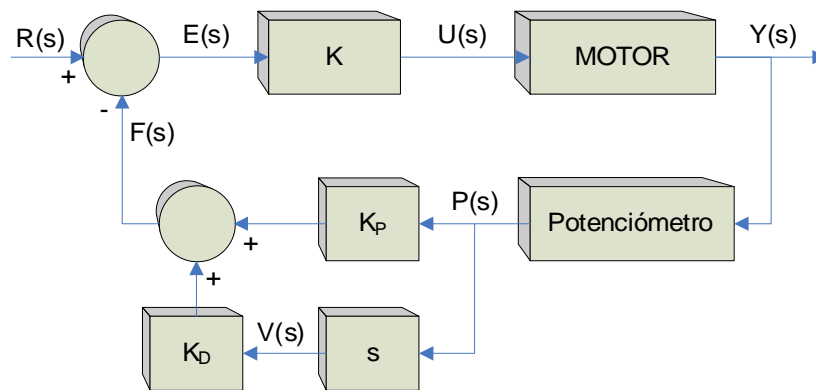
$$E(s) = R(s) - F(s)$$

- O sinal de erro é amplificado num factor de  $K$  através de um “amplificador de largura de impulso” e é aplicado ao motor através de uma “ponte H”.

$$U(s) = K \cdot E(s)$$

Logo, concluindo:

$$U(s) = K \cdot [R(s) - P(s) \cdot (K_P + s \cdot K_D)]$$



**Fig. 3.9: Representação esquemática do controlador de posição interno.**

Comparando a Fig. 3.8 e a Fig. 3.9 podemos relacionar o elemento  $s$  como o sensor de velocidade, a diferença  $R(s) - F(s)$  como o “comparador de largura de impulso” e o ganho  $K$  como o “amplificador de largura de impulso”. Verifica-se então dois tipos de compensação presentes: a compensação série executada pelo ganho proporcional  $K$  e a compensação paralela realizada na realimentação através de um ganho PD (proporcional+derivativo) à posição medida. Note-se que o sinal proveniente da realimentação  $f(t)$  para posterior comparação não depende exclusivamente da posição medida, mas também da velocidade actual o que confere uma maior estabilidade na realização do percurso para a posição desejada. Um pormenor que vale a pena salientar é a ausência da componente integral no controlo, o que poderá originar erros em regime estacionário quando aplicadas cargas elevadas sobre o eixo.

### 3.2.3 SETUP EXPERIMENTAL

De forma a poder avaliar o desempenho do servomotor foi montada o setup experimental ilustrado na Fig. 3.10. Pretende-se monitorar a posição efectiva do actuador ao longo do tempo e a corrente consumida, tendo em vista a variação dos seguintes parâmetros:

- Excursão do movimento (posição inicial e final);
- Velocidade do movimento<sup>2</sup>;
- Carga aplicada no eixo.



Fig. 3.10: Setup experimental.

Load	Mass (g)	Torque (N.m)
0	9	0.009
1	258	0.253
2	463	0.454
4	675	0.662
1+4	924	0.906
2+4	1129	1.108
1+2+4	1378	1.352

Tabela 13: Lista de cargas utilizadas para teste.

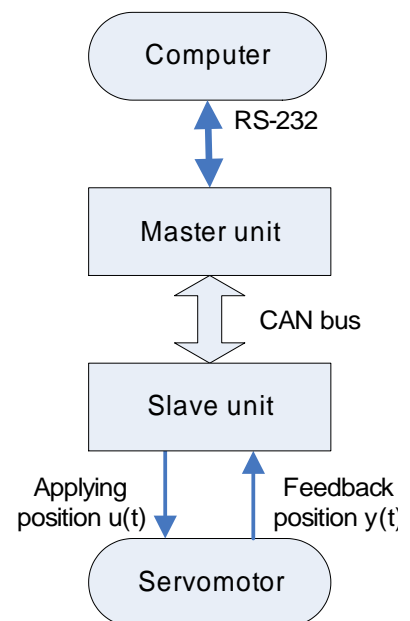


Fig. 3.11: Arquitectura das comunicações no setup.

Pela Fig. 3.10 podemos visualizar as ligações com a unidade principal (PC). Como se pretende testar um único actuador, será necessária uma unidade *slave* que liga ao PC segundo a rede de comunicações utilizada na plataforma humanóide.

O servomotor, por sua vez, encontra-se fixo num aparelho de fixação para poder mover cargas de uma forma segura. As cargas são presas ao servo através de um braço de massa desprezável e de 10 cm de comprimento que percorre a excursão dos 180° desde a posição vertical inferior até ao ponto vertical superior. Esta opção é muito útil dado que nos permite estimar o valor do binário resultante da força gravítica através da seguinte fórmula:

$$\tau = F_G \cdot L \cdot \cos \theta = M \cdot g \cdot L \cdot \cos \theta$$

em que  $M$  é a massa da carga,  $g$  a aceleração gravítica ( $9.81 \text{ m/s}^2$ ),  $L$  é o comprimento do braço (0.1m) e  $\theta$  é a posição angular do braço, com  $0^\circ$  na posição central (perpendicular ao vector gravítico),  $+90^\circ$  no extremo superior e  $-90^\circ$  no extremo inferior. A partir dela sabemos que nos extremos o binário gravítico é nulo e no ponto central (braço e  $F_G$  perpendiculares) é máximo.

<sup>2</sup> Embora não seja possível fazer controlo directo de velocidade, é possível a partir do controlo de posição induzir uma determinada velocidade ao dispositivo. Tal é explicado na secção seguinte.

A Tabela 13 indica a lista de massas utilizadas durante os testes. Embora as especificações destes servos indiquem um binário máximo de 2.42 N.m, na prática verificou-se que este estava muito abaixo deste valor deixando de responder para cargas superiores a 1.5 kg (binário  $\tau=1.47$  N.m) talvez devendo-se a desgaste destes actuadores. Tal justifica o facto da massa mais elevada da lista ser de 1.35 kg.

### 3.2.4 ACTUAÇÃO SOBRE O SERVOMOTOR

Tal como foi descrito na secção 3.2.1 cada unidade de controlo local é capaz de gerar três sinais de PWM independentes entre si, a partir do microcontrolador PIC disponibilizando-os através de três pinos reservados que se podem ligar directamente aos servomotores.

Tendo em conta que o microprocessador não dispõe de recursos suficientes para o controlo autónomo de três sinais de impulso modulado, foi necessário adoptar um algoritmo baseado em interrupções e algumas variáveis globais para controlar a largura do impulso de PWM para os três servomotores.

Pela Fig. 3.12 utilizou-se um *timer* para gerar uma interrupção periódica de 50 Hz, necessária para elevar o sinal de PWM a “1” para os três servomotores. A partir deste ponto, um segundo *timer* é programado com um determinado período entre 1 e 2 ms, responsabilizando-se assim por retornar o sinal de PWM a “0”. Desta forma temos um meio de controlar o duty-cycle do PWM para definição da posição dos servos.

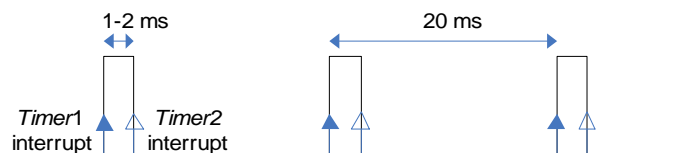


Fig. 3.12: Geração de um sinal de PWM através de dois timers.

Contudo, para ser possível definir um duty-cycle diferente para cada um dos três servomotores, dado que podem possuir posições diferentes, torna-se necessário recorrer a um estratégia diferente, se quisermos continuar com a utilização de apenas dois *timers* (para a arquitectura PIC 18F258 apenas possuímos no máximo quatro *timers*).

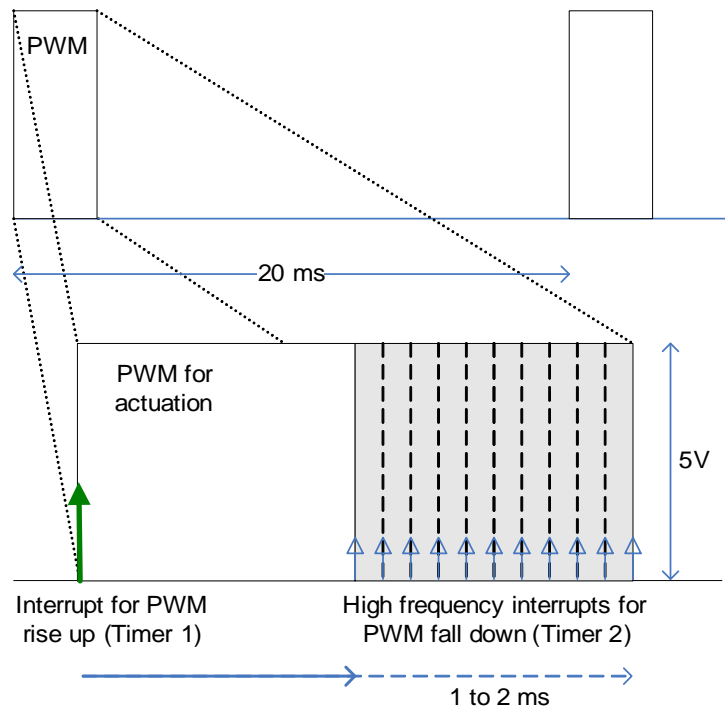
Para tal, utilizou-se uma variável global para cada servo, indicativa do *duty-cycle* correspondente à sua posição. O segundo *timer* iniciado, após a interrupção gerada pelo primeiro, é programado para gerar uma segunda interrupção após um intervalo de tempo correspondente ao *duty-cycle* mínimo: 1 ms.

A partir daqui, este segundo *timer* é reprogramado para gerar interrupções de alta-frequência de periodicidade correspondente à variação mínima de posição do servo  $\Delta_{pos}$ . Para um passo de  $1^\circ$ , o período será:

$$T_{HF} = \frac{duty_{max} - duty_{min}}{180^\circ / \Delta_{pos}} = \frac{2000\mu s - 1000\mu s}{180^\circ / 1^\circ} = \frac{1000\mu s}{180^\circ} = 5.56\mu s$$

A cada instanciação da interrupção de alta-frequência (passo de  $5.56\mu s$ ), cada servo é verificado se o PWM correspondente deve baixar a “0” nessa iteração, através da respectiva variável global. Em caso negativo, nada é feito e a próxima iteração é aguardada para uma nova avaliação.

Na última iteração (correspondente ao *duty-cycle* máximo) todos os sinais de PWM devem estar a “0” e o *timer* desligado. O processo é repetido na próxima interrupção do *timer* de 50 Hz.



**Fig. 3.13: Organização temporal das interrupções na geração do PWM.**

A Fig. 3.13 ilustra bastante bem este processo. A cada servomotor é atribuído uma variável indicadora da largura de impulso a aplicar, que basicamente é o número de iterações a aplicar durante as interrupções de alta-frequência a manter o sinal a “1”. Após a primeira interrupção um contador é utilizado para contagem das iterações que vão decorrendo entre o período de 1 a 2 ms e em cada uma delas é comparado com a variável atribuída a cada servomotor da duração do impulso. Quando o contador for igual a essa variável, o sinal de PWM correspondente é baixado a “0”.

### 3.2.5 LEITURA SENSORIAL DO SERVOMOTOR

Embora, por defeito, os servomotores não disponibilizem externamente os sinais de posição e de velocidade para *feedback*, é possível extrair o sinal de posição dado o fácil acesso ao potenciômetro interno. Este potenciômetro encontra-se solidário com o eixo do motor, e por isso, a medição da sua resistência resulta consequentemente na medição da posição angular do motor. Desta forma, abriram-se todos os servomotores e adicionou-se um fio ligado ao terminal do potenciômetro indicador da posição angular do eixo (Fig. 3.5). Já quanto à velocidade desconhece-se a forma de aceder a este dado. Tendo acesso à informação da posição, é possível monitorar constantemente a posição do servo a partir da unidade de controlo local e transmiti-la ao PC através da rede de comunicações, podendo avaliar continuamente a performance de cada actuador.

Deslocando o servo, na ausência de carga, verifica-se uma variação da tensão de saída entre 0.8 e 1.8 V ao longo dos 180° de excursão, podendo ser amostrado pelo PIC através da ADC. Utilizando as tensões de referência *standard* para a ADC (0 e +5V) precisamos de pelo menos 10 bits para o quantificador, tendo em conta que necessitamos de uma resolução que permita distinguir 180 posições possíveis (resolução de 1°).

$$bits_{quant} = \lceil \log_2(\# \text{níveis de quantificação}) \rceil \quad \# \text{níveis de quantificação} = \frac{5V}{\frac{1.8V - 0.8V}{180^\circ}} = 900$$

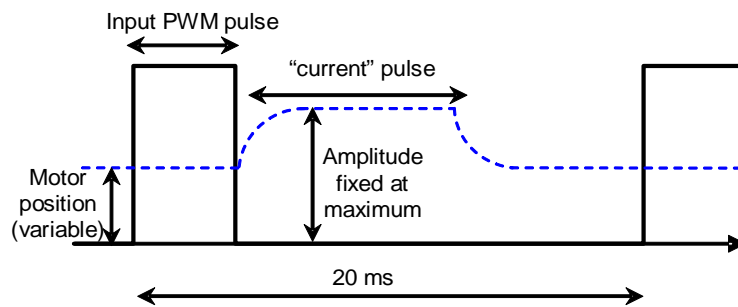
$$bits_{quant} = \lceil \log_2(900) \rceil = 10 \text{ bits}$$

Como a ADC do PIC oferece a opção de quantificação a 10 bits tal é conveniente aproveitar. No entanto, a medição não é tão simples como amostrar a tensão de saída quando desejado, pois na presença de cargas no

eixo e/ou de velocidades elevadas surge um estranho impulso acima do nível de tensão correspondente à posição, pelo que se a medição for executada no momento do impulso o resultado será falso. Tal motivo é devido ao facto de nos modelos da HITEC a tensão de referência do potenciómetro não ser a mesma que a da ADC do PIC, pelo que, embora para a electrónica de controlo interna este sinal corresponderá sempre à posição do servo, para a perspectiva da ADC, este sinal possuirá oscilações na forma de impulsos ao longo do tempo, mesmo sem variar a posição do servo.

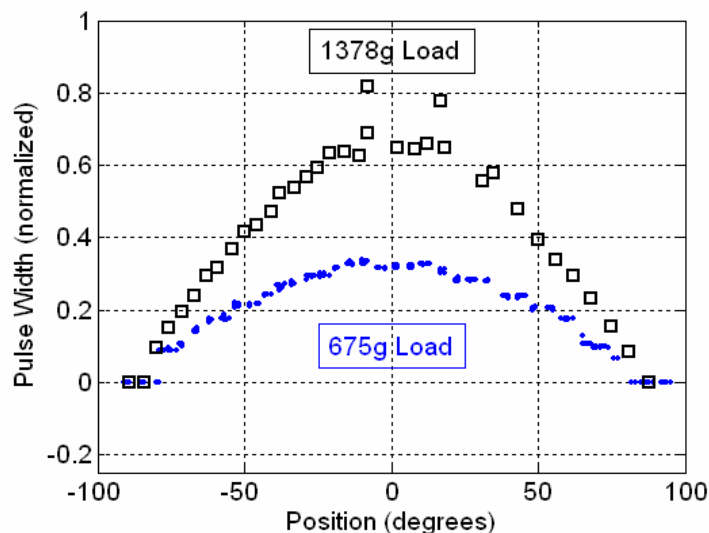
Testando diversas situações através do osciloscópio concluímos que estes impulsos possuíam propriedades não casuais:

- O impulso ocorre sempre acima da tensão indicadora da posição;
- A amplitude do impulso é constante e apenas varia com a tensão de alimentação;
- Período de repetição coincidente com o PWM aplicado (50Hz);
- Ponto de início sincronizado com o fim do impulso de PWM;
- Largura dependente da carga e da velocidade.



**Fig. 3.14: Impulso de tensão medido na presença de cargas/velocidades elevadas.**

O último aspecto foi o que se revelou mais interessante: quando se aumentava a carga ou a velocidade, a largura deste impulso aumentava. Tentando relacionar a largura deste impulso com a massa da carga efectuou-se o seguinte teste: fazia-se deslocar o servo para um conjunto de posições entre  $-90^\circ$  e  $+90^\circ$ , e para cada uma delas o actuador era deixado em repouso e media-se a largura do impulso (Fig. 3.15). Desta forma, assegurávamos que a largura medida apenas era devida ao binário gravítico (sem a interferência da velocidade). Esta experiência foi executada para duas cargas de massas diferentes, uma aproximadamente dupla da da outra.



**Fig. 3.15: Relação entre a posição e a largura de impulso.**

Analisando cada uma das duas cargas à parte, verifica-se um comportamento sinusoidal ao longo das diversas posições sugerindo a relação com a função *seno/coseno*. Ora, pela equação do binário gravítico, se pode concluir que a largura do impulso de tensão está directamente relacionada com o binário gravítico. Relacionando os valores entre as duas cargas, tal fundamento é reforçado dada a relação de multiplicação

entre os dois conjuntos – aproximadamente de dois!

Ora como se sabe a corrente está directamente relacionada com o binário:

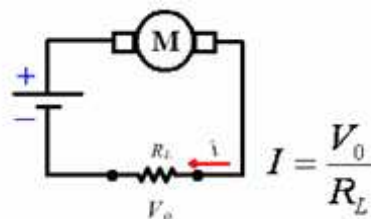
$$I = K \cdot \tau$$

pelo que a largura de impulso é um indicador da corrente drenada pelo servo. Por esta razão, de agora em diante, este impulso será denominado por impulso de corrente. Este comportamento confirma a actuação sobre o motor DC de uma forma digital, também utilizando sinais no formato de PWM através da ponte H.

Note que, pelas características do sinal de posição, é possível medir tanto a posição como a corrente a partir desta única fonte, sem a necessidade de qualquer electrónica adicional para a medição de corrente!

### Medição de Corrente

Inicialmente, usou-se um método para a medição de corrente baseado no uso de uma resistência de baixo valor em série com a alimentação do servo.



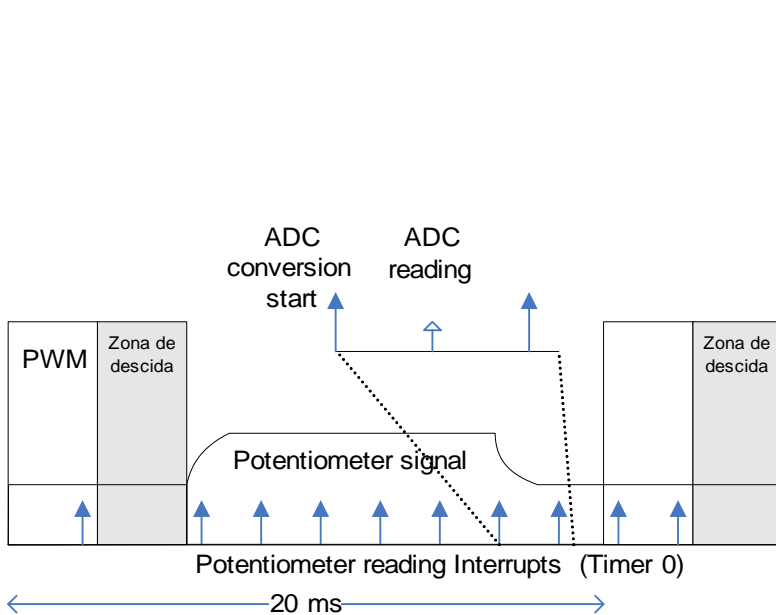
**Fig. 3.16: Configuração possível para medição da corrente.**

Embora este método permita uma leitura de corrente mais simplificada e de resultado sempre disponível (sem a necessidade de algoritmos de medição de largura de impulsos) introduz outros inconvenientes:

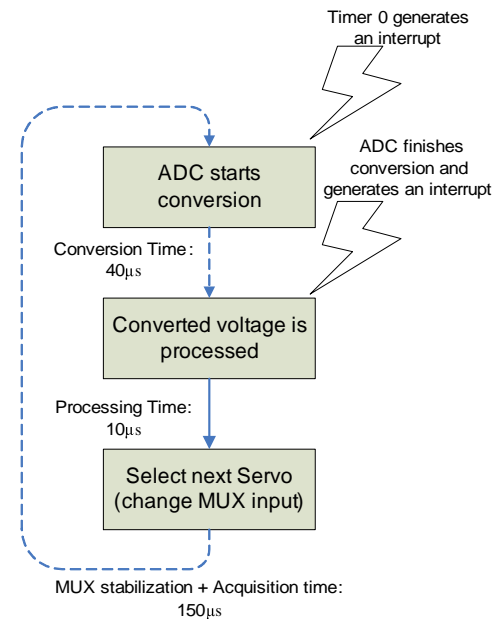
- \* Emissão excessiva de calor nas resistências nas situações de maior consumo de corrente.
- \* A presença da resistência provoca um aumento da corrente consumida pelo servo. Tal é explicado pelo facto de na resistência se verificar uma queda de tensão o que resulta numa diminuição da tensão de alimentação do servo, que será tanto maior, quanto maior for a corrente exigida. Com esta diminuição, o servo tem de realizar um maior esforço o que equivale a aumentar a corrente drenada.

Dados estes inconvenientes, achou-se proveitoso, aproveitar o sinal de posição para também ler o consumo de corrente com base na largura do impulso de corrente. Desta forma, o *hardware* é simplificado dispensando a utilização de resistências em série com a alimentação dos servos, passando o trabalho para o lado do *firmware*, responsabilizando-se por determinar a posição e a corrente consumida unicamente através do sinal de posição, pela medição do seu valor mínimo e da largura de impulso respectivamente.

A Fig. 3.17 descreve a estratégia adoptada: uma interrupção baseada num *timer* é gerada durante todo o período de PWM com uma periodicidade de 200µs. Apenas na zona de descida do PWM, compreendida entre 1 e 2ms do *duty-cycle*, ela é inibida não se pretendendo interferir com as interrupções de alta-frequência. Sempre que este *timer* gera uma interrupção, a tensão de saída do potenciómetro é medido e é comparado com um determinado limiar logo após a leitura da ADC. Se estiver abaixo desse limiar considera-se a não ocorrência do impulso de corrente, mas caso esteja acima, uma variável contadora da largura de impulso é incrementada. No fim do período de PWM (considerado logo após a zona de descida de PWM) esta variável contadora indicará a largura do impulso medido no período de PWM anterior. De notar que a resolução de medição da largura de impulso é tanto maior quanto menor for a periodicidades das interrupções. No entanto não convém definir períodos muito curtos sob pena de não atribuir suficiente largura de banda de CPU para o programa principal ser executado completamente num período de PWM. A periodicidade de 200µs foi o melhor compromisso encontrado e permite executar no total 95 medições, ou 31 para cada servo, ao longo de um período de PWM (excluindo a zona de descida de PWM).



**Fig. 3.17: Organização das interrupções (setas) na medição sensorial.**



**Fig. 3.18: Algoritmo de leitura dos três servomotores.**

De referir que estas medições são feitas aos três servos de forma alternada. Para cada instanciação de 200  $\mu$ s, apenas um servo é medido, mas irá alternando, até que ao fim de 600  $\mu$ s todos os três servos são lidos (Fig. 3.19).



**Fig. 3.19: Multiplexagem na leitura dos servos.**

O valor do limiar a considerar depende da amplitude do impulso, mas sabendo que este apenas varia com a tensão de alimentação, mantendo uma amplitude sempre superior a 1.0V, utilizando baterias de 7.4V, definiu-se o limiar como 1.0V. Note que este limiar encontra-se acima da tensão DC pelo que é necessário em cada período de PWM medir a tensão mínima e utilizá-la para o período seguinte como sendo a tensão base dos impulsos.

Contudo este método possui uma desvantagem. Em situações de elevada exigência, em que a corrente consumida é próxima da máxima, a largura do impulso de corrente pode ocupar praticamente todo o período de PWM inibindo a capacidade de leitura da posição e da largura do impulso, dado que o valor mínimo corresponde ao topo do impulso de corrente. É, por isso, de evitar que esta situação ocorra, quer impondo limites físicos, quer pela adopção de estratégias de controlo que minimizem a corrente a consumir – uma delas é a limitação de velocidade.

### **Medição da Posição**

Já foi referido anteriormente que a componente DC da saída do potenciómetro está relacionada com a posição do servo, pelo que a tarefa de guardar o valor mínimo deste sinal, executada para efeitos de detecção do impulso de corrente, é aproveitada para o cálculo da posição angular do servo.

Convencionando a tensão mínima como correspondendo à posição  $+90^\circ$  e a tensão máxima como  $-90^\circ$ , recorrendo a uma relação linear podemos determinar a posição através relação entre a leitura da ADC com um ganho  $m$ , com  $b$  correspondente à posição angular do servo quando o resultado da ADC é nulo.



$$\boxed{pos = m * ADC + b}$$

$$m = \frac{+90^\circ - (-90^\circ)}{ADC_{(v=0.8V)} - ADC_{(v=1.8V)}} = \frac{180^\circ}{163 - 368} = -0.878$$

$$b = pos - m * ADC = 90^\circ - m * ADC_{(pos=90^\circ)} = 90^\circ + 0.878 * 163 = 233$$

em que  $ADC$  correspondente à leitura da ADC para uma determinada posição ( $pos$ ) / tensão ( $v$ ) do potenciómetro:

$$ADC(v) = \frac{v}{5V} \cdot (2^{bits_{quant}} - 1) = v \cdot \frac{2^{10} - 1}{5V} = v \cdot \frac{1023}{5V}$$

Logo,

$$\boxed{pos(ADC) = 233 - ADC \cdot 0.878}$$

Como esta medição também têm de ser feita para a medição do consumo de corrente, ela é aplicada após as interrupções do *timer* de 200  $\mu$ s (Fig. 3.17), seguindo o modelo referido de alternância entre os três servomotores (Fig. 3.18). Ao fim de um período de PWM (19 ms excluindo o intervalo das interrupções de alta-frequência), a tensão mínima é considerada como correspondendo à posição do servomotor. Como já foi referido na secção anterior, algum cuidado é necessário para limitar o consumo de corrente, de modo a evitar a monopolização do impulso de corrente em todo o período de PWM, o que geraria uma falsa leitura da posição. A secção 3.3.2 detalha um método de limitação da velocidade o que irá necessariamente impor algum controlo no consumo de corrente.

Um pormenor importante é a impossibilidade de garantir que a gama da tensão de saída do potenciómetro esteja compreendida entre 0.8 e 1.8V para qualquer servomotor, podendo-se verificar variações até cerca de 0.2V. Tal é problemático para o cálculo da posição angular, pois estamos a considerar uma relação de correspondência constante quando tal é difícil de se garantir entre dois actuadores. No entanto, note que esta diferença é mais notória para o parâmetro  $b$  do que para  $m$ . O valor de  $b$  exige que os extremos de posição possuam exactamente os valores enunciados, mas para o valor de  $m$  só é exigido uma excursão entre extremos de 1.0V o que já é mais frequente. Este pequeno detalhe permite-nos assim a utilização de uma rotina de calibração simples de modo a acertar o parâmetro  $b$  de acordo com o servo a lidar.

Considerando o parâmetro  $m$  constante, o que é uma aproximação aceitável, sempre que o sistema é ligado, se assegurarmos que cada servo está numa posição conhecida *à priori*, é fácil calcular o valor de  $b$  através da equação de cálculo deste parâmetro e utilizá-lo nas medições consequentes.

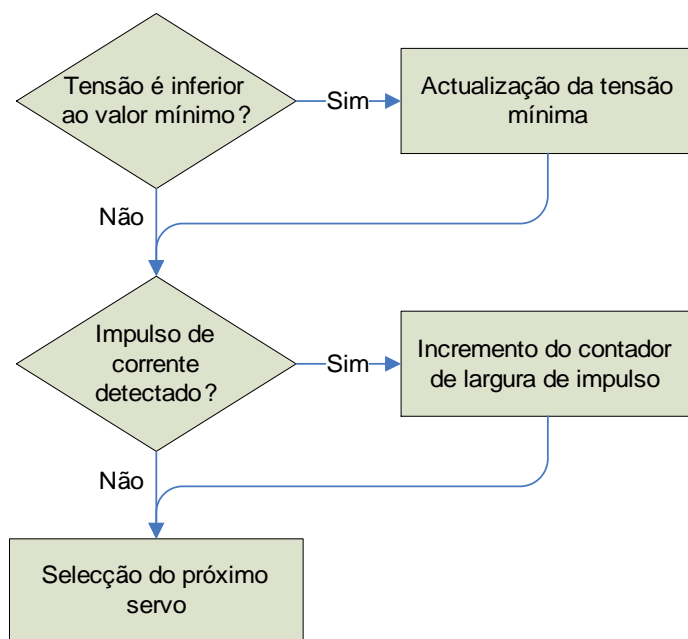
A rotina de calibração é executada sempre que o sistema arranca e segue o seguinte algoritmo:

1. A primeira mensagem de actuação já chegou? Só passar para o passo seguinte, quando afirmativo (é o *master* que fornece as posições iniciais);
2. Após a recepção da posição inicial, actuação sobre os servomotores de forma a cumprir a posição solicitada pela unidade *master*;
3. Esperar dois segundos para a realização do movimento e estabilização do sinal de posição;
4. Amostragem de 25 medidas da saída do servo (25 períodos de PWM);
5. Cálculo da média aritmética do valor medido;
6. A partir da seguinte equação determinar o valor de  $b$  a partir da posição inicial fornecida pelo *master* e do valor médio  $ADC$ :
$$b = pos_{Master} + ADC_{medio} \cdot m$$
7. Activação dos filtros de medição sensorial.

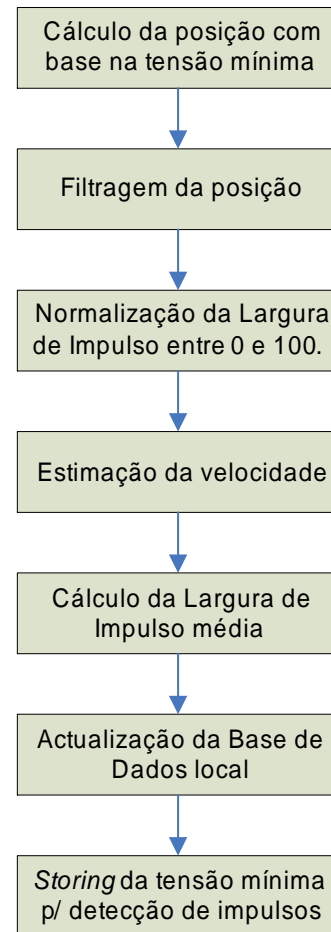
Desta forma, a medição da posição é adaptado a cada servomotor de uma forma personalizada com mínimos erros de cálculo (apenas dependem do erro de  $m$ ).

## Integração dos Algoritmos de Medição Sensorial

A Fig. 3.20 apresenta o algoritmo final que será executado de 200 em 200  $\mu$ s ao longo de um período de PWM, até que no fim deste, a posição do servo e a sua corrente é determinada, pela avaliação do valor mínimo e pela largura do impulso respectivamente. A posição será dada em graus (-90 a +90°), e a corrente será um valor normalizado entre 0 e 100 indicativo da percentagem da largura do impulso de corrente ao longo do período de PWM.



**Fig. 3.20:** Algoritmo de processamento da tensão medida para cada servomotor.



**Fig. 3.21:** Processamento final (fim do período de PWM).

No fim do período de PWM também são realizadas operações de filtragem da posição e da corrente, bem como também a estimação da velocidade média dos últimos 100ms pela utilização directa das posições medidas (Fig. 3.21). Após a salvaguarda destes resultados na base de dados local do microcontrolador, estes dados ficam disponíveis para envio para a unidade *master* através do barramento CAN.

### 3.3 ESTUDO DO SERVOMOTOR EM MALHA ABERTA

Agora que os detalhes de funcionamento do microcontrolador foram apresentados, vamos estudar o comportamento do servomotor sob determinadas condições de modo a avaliar a seu desempenho. O estudo será feito do ponto de vista de um sistema no qual aplicaremos uma entrada e queremos observar a sua resposta (Fig. 3.22).

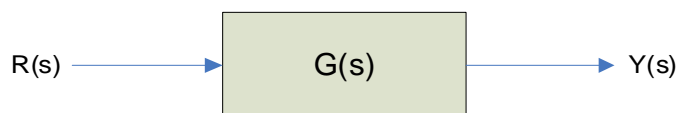


Fig. 3.22: Representação do servomotor por uma função de transferência  $G(s)$ .

Ora, tendo sido discutido o funcionamento interno do servomotor na secção 3.2.2 (Fig. 3.8), infelizmente apenas era uma aproximação ao nosso modelo e além que não conhecemos os valores dos ganhos enunciados. Por tudo isto iremos estudar o comportamento do sistema  $G(s)$ , desconhecendo a sua função de transferência.

Embora, pudéssemos tentar encontrar esta expressão experimentalmente através de métodos bastante conhecidos como é o caso das regras de *Ziegler-Nichols*, tal não nos leva muito longe, pois como é sabido, as juntas da plataforma humanoíde estão sujeitas a constantes variações de inércia pelo que o comportamento dos servos torna-se bastante não linear. Teríamos, por isso, uma função de transferência diferente para cada carga aplicada.

Esta secção tem como objectivo estudar primeiramente a resposta em malha aberta do servo, ou seja, aplicando um sinal (ou uma sequência de sinais) de PWM na entrada  $r(t)$  e analisar a resposta pela leitura da posição angular  $y(t)$ . Rotinas em *MatLab* para uso pela unidade principal foram especialmente construídas para este efeito enviando para o *slave* respectivo uma ordem de actuação, e monitorar a posição do servo o tempo suficiente para permitir captar o percurso realizado até atingir a posição solicitada.

Algum vocabulário específico será utilizado nas secções que se seguem, pelo que convém, antes de tudo, esclarecer o seu significado:

- **Tempo de crescimento/subida**: tempo que a resposta demora a crescer entre 10% e 90% da distância prevista a percorrer;
- **Tempo de atraso**: instante que a resposta atinge 50% da distância total prevista;
- **Tempo de pico**: instante em que a resposta passa pela posição máxima/mínima (depende do sentido de deslocamento);
- **Tempo de estabelecimento**: tempo necessário para que a resposta entre, sem voltar a sair, numa determinada vizinhança, previamente especificada, do valor final da resposta. Dois a cinco por cento é normalmente a margem especificada.
- **Overshoot**: oscilação verificada em torno do valor final no fim da resposta. O seu valor corresponde normalmente à relação entre a margem máxima de oscilação e a distância total percorrida (percentagem).
- **Erro em regime estacionário**: diferença entre o valor final e o valor desejado após estabilização da resposta.

Estas definições correspondem a características da curva de resposta do sistema a testar e normalmente são usados para avaliar a sua *performance* (Fig. 3.23). No caso de um actuador ideal, todas estas características deviam ser nulas, mas infelizmente tal não existe na realidade: os sistemas levam tempo a reagir e a atingir o seu valor final, e por vezes podem entrar em oscilação (*overshoot*) quando a entrada é demasiado exigente. Este capítulo procurará perceber o quanto os servomotores se desviam da resposta ideal, e posteriormente tentaremos encontrar soluções para a sua melhoria.

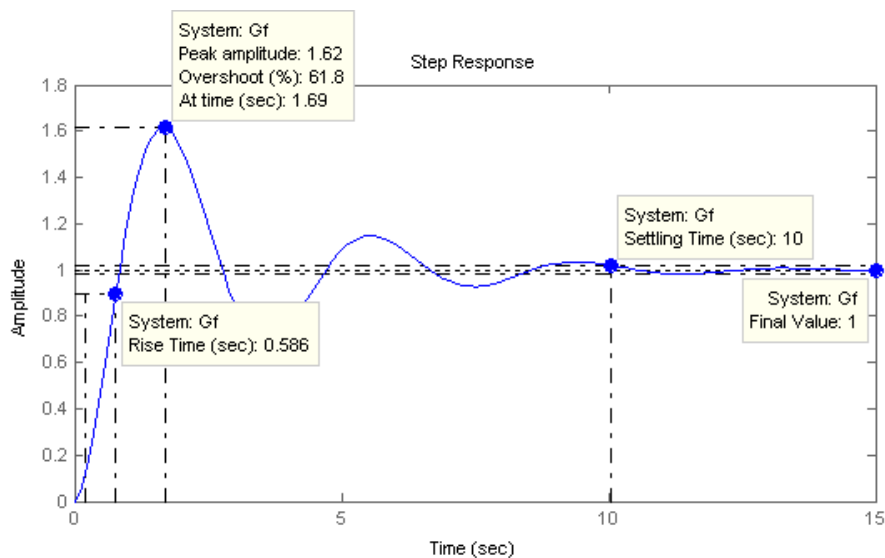


Fig. 3.23: Exemplo da resposta de um sistema por compensar com a indicação das suas características.

### 3.3.1 RESPOSTA AO DEGRAU EM MALHA ABERTA

Com uma carga aplicada no eixo, de acordo com a Fig. 3.10, inicializou-se o servo na posição inicial de  $-45^\circ$  e enviou-se um comando de actuação para a posição de  $+45^\circ$  – um degrau é aplicado. Monitorizou-se o percurso percorrido através do comando de leitura de posição durante 1.5s para duas cargas diferentes: 258 e 1138g (Fig. 3.24).

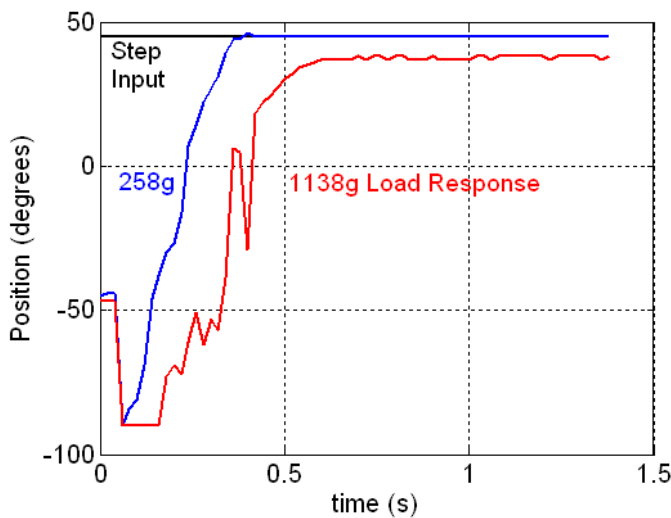


Fig. 3.24: Comparação das respostas ao degrau para duas cargas no percurso de  $-45$  para  $+45^\circ$ .

Posição desejada ( $^\circ$ )	Posição medida ( $^\circ$ )	Erro ( $^\circ$ )	Binário (Nm)
-80	-80	0	0.198
-60	-62	2	0.569
-40	-45	5	0.872
-20	-28	8	1.069
0	-9	9	1.138
+20	+11	9	1.069
+40	+33	7	0.872
+60	+55	5	0.569
+80	+80	0	0.197

Tabela 14: Erros em regime estacionário em diferentes posições para uma carga de 1138g.

Comparando as respostas da carga de 258g com a de 1138g, constata-se uma diferença dos tempos de subida evidenciando um atraso maior para cargas elevadas, o que é compreensível uma vez que o esforço dispendido é maior. Um segundo aspecto é a diferença no erro em regime estacionário: para a massa leve o erro é praticamente nulo, mas para a mais pesada já é mensurável um erro de cerca de  $9^\circ$ .

Adicionalmente, fez-se uma experiência tendo em vista o estudo da relação do erro em regime estacionário com o binário resultante da força gravítica. Deslocando o servo para um conjunto de posições conhecidas, para cada uma delas, esperou-se pela finalização do movimento e pela estabilização do sinal de posição, anotando de seguida a posição medida pelo microcontrolador (Tabela 14). Comparando a posição solicitada com a efectiva observa-se que o erro aumenta à medida que a posição se aproxima do ponto  $0^\circ$ , o que demonstra que quanto maior é o binário gravítico maior é a dificuldade em atingir a posição final, resultando num erro em regime estacionário não nulo.

Outro aspecto a salientar é a visível instabilidade durante a realização do trajecto, o que é mais notório para cargas elevadas do que para baixas. Além disso observa-se um “salto” no início da trajectória para posições inferiores a  $-45^\circ$  seguidamente com inversão de velocidade em direcção ao valor final. No entanto este comportamento não corresponde ao observado, visualizando-se um movimento rápido e sem oscilações durante todo o percurso. Tal sugere que a posição lida a partir do potenciómetro esteja a ser perturbada por algum efeito anómalo.

A presença de acelerações bruscas, como acontece no arranque da trajectória, pode provocar picos de corrente, que em termos sensoriais, correspondem a impulsos de corrente que podem ocupar todo o período de PWM. Nestas circunstâncias a posição considerada como sendo a tensão mínima deixa de poder ser medida, sofrendo um aumento em tensão para o topo do impulso, que, em termos de posição angular, corresponde a uma descida brusca, tal como os gráficos nos mostram.

Neste sentido, pode-se dizer que a presença do impulso de corrente está a mascarar as medições de posição do servo agravando os resultados na presença de cargas elevadas. No entanto, tal deixa de se verificar nos últimos instantes ainda antes da finalização da trajectória e durante a fase estacionária, demonstrando que não só a massa da carga aumenta as exigências de corrente, como também a velocidade e as acelerações bruscas.

### 3.3.2 CONTROLO DE VELOCIDADE

Como se pode concluir, os servomotores são muito sensíveis a variações bruscas de velocidade e de aceleração provocando muito facilmente picos de corrente que impedem a correcta leitura da sua posição. De modo a prevenir este efeito, tentou-se introduzir algum controlo de velocidade, para que também pudéssemos regular a velocidade segundo as nossas necessidades, possibilitando movimentos mais suaves. No entanto, como não temos possibilidade de fazer o controlo directo de velocidade, utilizar-se-á o controlo de posição para executar trajectórias que no seu todo definem uma velocidade média que pode ser configurável.

Até agora temos vindo a aplicar degraus de posição aos servos tal como exemplifica a Fig. 3.24. Se aplicarmos uma sucessão de degraus de variação de amplitude e intervalo de tempo o mais pequenos possíveis, cuja amplitude final de cada degrau aumenta proporcionalmente até atingir a posição desejada, temos a aplicação de uma rampa de posições cujos extremos de posição e duração total definem a velocidade média do movimento. A Fig. 3.26 apresenta um exemplo de uma trajectória em rampa desde  $-45^\circ$  até  $+45^\circ$  com uma duração de 1.8s, o que corresponde a uma velocidade média de  $50^\circ/s$ .

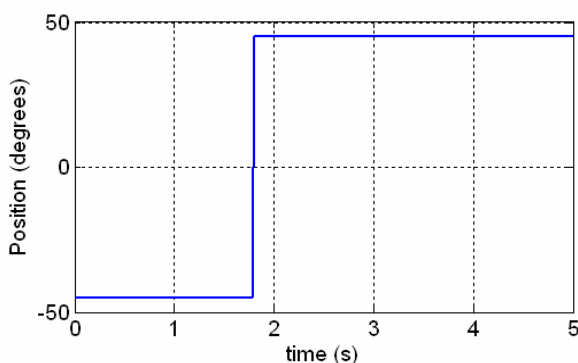


Fig. 3.25: Aplicação de um degrau de  $-45^\circ$  para  $+45^\circ$  no instante  $t=1,8s$ .

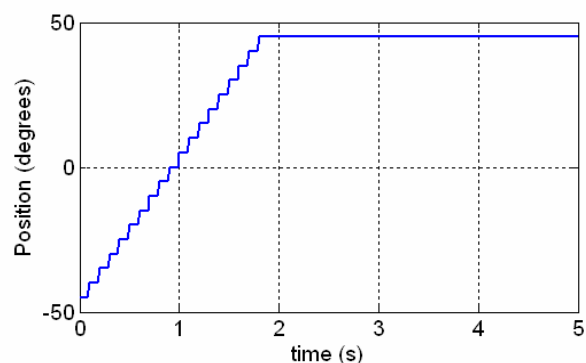


Fig. 3.26: Aplicação de uma rampa de posição de velocidade média  $50^\circ/s$ .

Enquanto que a aplicação de um degrau introduz um delta de *Dirac* na velocidade e na aceleração provocando facilmente picos de corrente, a trajectória em forma de rampa contém essas variações bruscas pela aplicação sucessiva de pequenos degraus o que limita a velocidade do servo e também a sua necessidade de consumo de corrente.

Esta trajectória pode ser implementada através da seguinte equação:

$$pos = pos_0 + v_{med} \cdot t = pos_0 + v_{med} \cdot n \cdot T_a \quad \text{com } T_a=20\text{ms (período de PWM)}$$

$$vel = v_{med} \quad accel = 0$$

No entanto, se quisermos otimizar ainda mais o consumo de corrente, podemos implementar trajectórias, de modo a limitar os deltas de *Dirac* na aceleração que continuam a verificar-se no caso da rampa. Se além de variarmos a posição, também variarmos a velocidade de modo a ser nula no início e no fim da trajectória, a necessidade de consumo de corrente decai ainda mais. Tal é exequível através de uma equação polinomial de terceira ordem que introduz velocidade zero no início e no fim de cada trajectória (Fig. 3.27).

$$pos = c_0 + c_1 \cdot t + c_2 \cdot t^2 + c_3 \cdot t^3 \quad vel = c_1 + 2 \cdot c_2 \cdot t + 3 \cdot c_3 \cdot t^2 \quad accel = 2 \cdot c_2 + 6 \cdot c_3 \cdot t$$

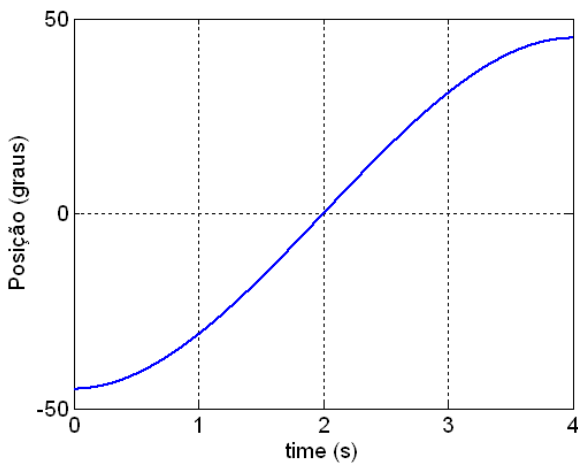


Fig. 3.27: Trajectory polynomial of third order.

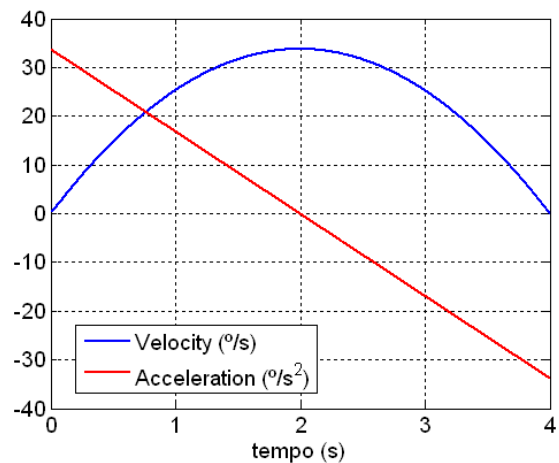


Fig. 3.28: Behavior of the velocity and acceleration on the third-order trajectory.

Como se pode observar, pela Fig. 3.28, deixa de se verificar deltas de *Dirac* até à segunda derivação da posição (aceleração). Se aumentarmos mais ainda a ordem do polinómio para 5, podemos garantir aceleração nula no início e no fim. Contudo, à medida que aumentamos a ordem do polinómio, maior é a velocidade instantânea a meio do trajecto, pelo que uma ordem superior a 5 não é recomendável.

Voltando a amostrar a resposta dos servos em malha aberta, agora com a implementação de trajectórias, podemos observar, para o caso da rampa (Fig. 3.29 e Fig. 3.30) a estabilidade acrescida nas respostas para qualquer carga aplicada. Os efeitos de picos de posição e as oscilações durante o percurso praticamente desapareceram demonstrando o consumo controlado de corrente com esta solução.

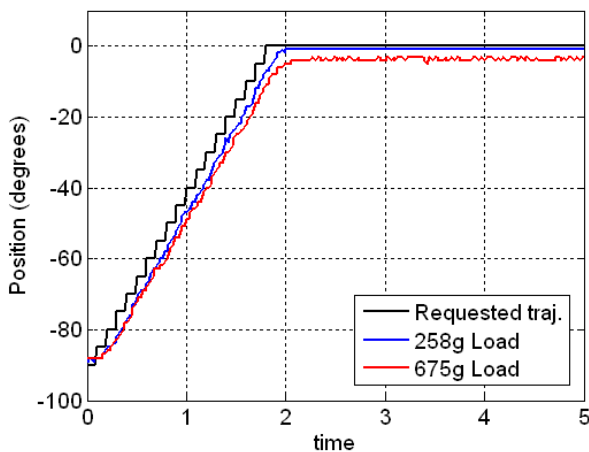


Fig. 3.29: Resposta à rampa com duas cargas diferentes ( $\Delta p=5^\circ$ ,  $\Delta t=100\text{ms}$ ).

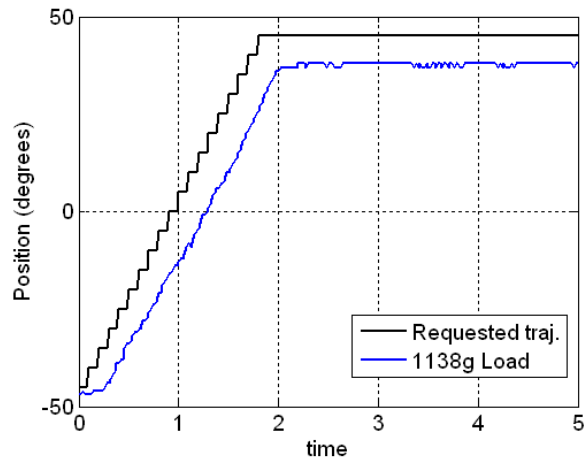


Fig. 3.30: Resposta à rampa com uma carga pesada ( $\Delta p=5^\circ$ ,  $\Delta t=100\text{ms}$ ).

A Fig. 3.31 apresenta as respostas correspondentes às trajectórias polinomiais com a observação dos mesmos resultados que os da rampa.

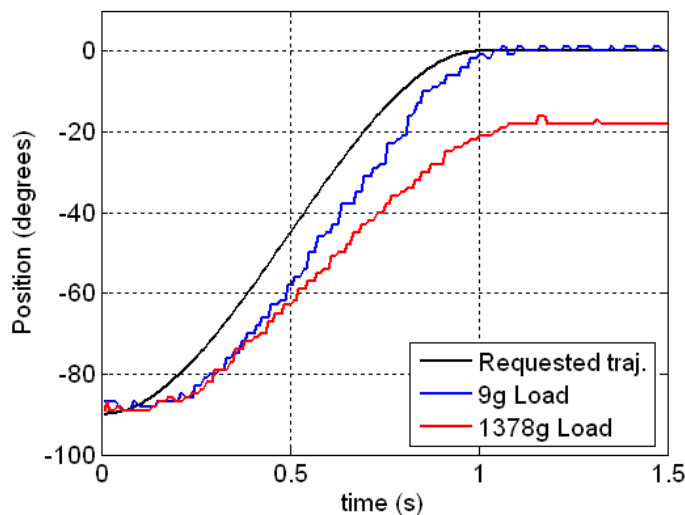


Fig. 3.31: Resposta ao polinómio para duas cargas diferentes ( $T_{\text{traj}}=1\text{s}$ ).

No entanto, continua-se a verificar o crescente tempo de atraso e erro em regime estacionário relativamente à trajectória esperada para cargas elevadas. Para corrigir estas características torna-se importante realizar alguma espécie de controlo adicional ou ao nível do controlador interno do servo, ou externamente, usando o sinal de posição como *feedback* e o sinal de PWM para controlo da posição.

### 3.4 ESTUDO DO SERVOMOTOR EM MALHA FECHADA

Esta secção descreve algumas estratégias para a melhoria da resposta dos servos, relativamente ao observado em malha aberta, em termos de:

- Tempo de subida;
- Tempo de estabelecimento;
- Erro em regime estacionário.

O objectivo é, basicamente, a minimização destes parâmetros de modo a aproximar ao máximo a resposta à trajectória solicitada. Várias metodologias podem ser seguidas para compensação, como por exemplo a substituição da electrónica de controlo dos actuadores. No entanto, têm-se em mente a procura de métodos

que não modifiquem o *hardware* destas unidades, de forma a permitir a fácil substituição em caso de necessidade. Por razões de simplicidade realizar-se-á o controlo externamente ao servomotor usando para isso o microcontrolador para implementar a lei de controlo mais adequada. Com a implementação do controlador por *firmware*, é possível alterar os parâmetros ou a estrutura do controlador em tempo real, quer autonomamente ao nível da unidade *slave* em questão, ou pela troca de informação entre o PC e o respectivo módulo. Desta forma, evita-se intervenções ao nível do *hardware*, e torna viável a implementação de algoritmos de controlo dinâmicos que se auto regulam de acordo com as necessidades do sistema.

### 3.4.1 O CONTROLADOR

A Fig. 3.32 descreve a metodologia a usar: o controlador representado pelo bloco  $G_C(z)$  é implementado ao nível do microcontrolador que fará uso do sinal de posição  $y(t)$  como *feedback* para comparação com a posição desejada  $r(t)$ , para melhor decidir a largura de PWM  $u(t)$  a aplicar ao servo.

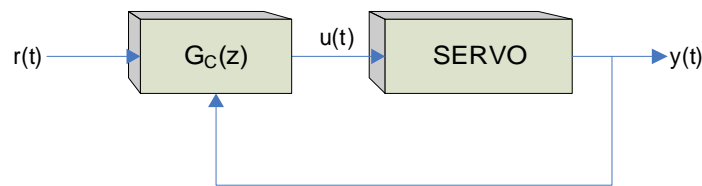


Fig. 3.32: Controlo externo do servomotor.

Note que não se pode afirmar que a cada *duty-cycle* de PWM do sinal de actuação corresponde a uma determinada posição, pois tal depende muito da inércia aplicada sobre o eixo, mas sim que para uma determinada carga corresponde a uma determinada posição. Para tal, a lei de controlo  $G_C(z)$  seguirá um método para procurar o sinal de PWM mais adequado para que o sinal de posição medido  $y(t)$  coincida com a posição solicitada  $r(t)$ . Esse método é apresentado a seguir na Fig. 3.33 e é baseado num compensador clássico do tipo PID [15-17].

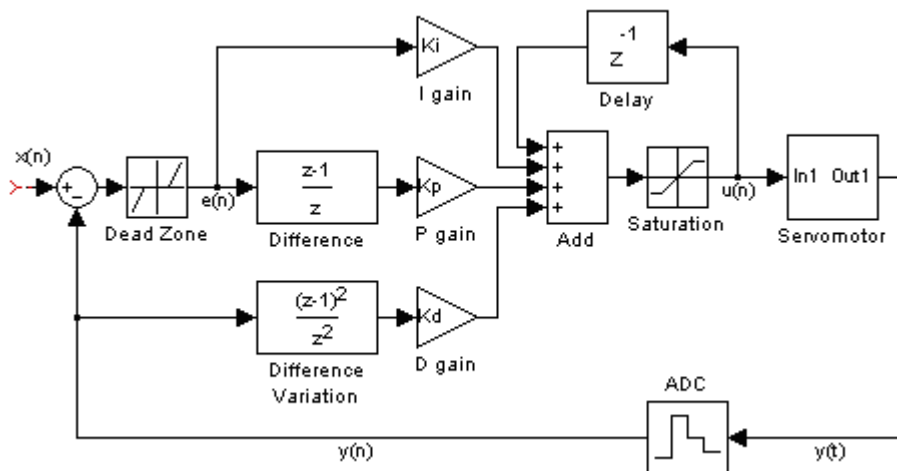


Fig. 3.33: Compensador PID incremental.

Este compensador é baseado na equação de compensação PI-D no domínio de *Laplace*, sendo convenientemente adaptado para o caso digital:

$$U(s) = \left[ k_i \cdot \frac{1}{s} + k_p + k_d \cdot s \right] \cdot E(s)$$

No domínio digital  $z$  temos:

$$U(z) = \left[ k_i \cdot \frac{T_s}{1 - z^{-1}} + k_p + k_d \cdot \frac{1 - z^{-1}}{T_s} \right] \cdot E(z), \quad \text{com } T_s \text{ correspondente ao período de amostragem.}$$

Fazendo  $K_I = k_i \cdot T_s$ ,  $K_P = k_p$ , e  $K_D = k_d / T_s$ , com  $T_s = 20$  ms (período de PWM), temos então:



$$U(z) = \left[ K_I \cdot \frac{1}{1-z^{-1}} + K_P + K_D \cdot (1-z^{-1}) \right] \cdot E(z)$$

Colocando a equação na forma incremental:

$$U(z) \cdot (1-z^{-1}) = K_I \cdot E(z) + K_P \cdot E(z) \cdot (1-z^{-1}) + K_D \cdot E(z) \cdot (1-z^{-1})^2$$

E finalmente, na forma numérica:

$$\boxed{u(n) - u(n-1) = \Delta u = K_I \cdot e(n) + K_P \cdot [e(n) - e(n-1)] + K_D \cdot [e(n) - 2 \cdot e(n-1) + e(n-2)]}$$

$$\boxed{\Delta u = (K_I + K_P + K_D) \cdot e(n) - (K_P + 2 \cdot K_D) \cdot e(n-1) - 2 \cdot K_D \cdot e(n-2)}$$

em que  $\Delta u$  corresponde a uma variação relativamente à amostra anterior.

Dado que o termo de segunda ordem é muito susceptível à instabilização da resposta, é conveniente fazer com que este termo não dependa das variações do sinal de referência  $r(t)$  que contém a posição desejada, conferindo à lei de controlo maior robustez contra oscilações.

Substituindo no termo derivativo o sinal de erro pela diferença entre a referência  $r(t)$  e a posição efectiva  $y(t)$ , e considerando  $r(t)$  constante ao longo do tempo, então temos a seguinte expressão:

$$\boxed{\Delta u = K_I \cdot e(n) + K_P \cdot [e(n) - e(n-1)] - K_D \cdot [y(n) - 2 \cdot y(n-1) + y(n-2)]}$$

Esta é a lei de controlo implementada em cada microcontrolador, cujos parâmetros  $K_I$ ,  $K_P$  e  $K_D$  são passados pelo PC através da rede de comunicações RS-232 e CAN. Prevê-se que a componente integral resolva o problema do erro em regime estacionário e as restantes componentes lidem com a velocidade do sistema, mas note que não se devem fazer suposições sobre os seus efeitos com base no controlo clássico, uma vez que nem sequer iremos lidar com entradas em degrau, mas sim numa sucessão de degraus, o que pode resultar em efeitos diferentes.

Repare que o controlo é realizado de forma incremental (repare no elemento  $z^{-1}$  que introduz na compensação a última saída  $u(n-1)$ ), calculando em cada iteração o incremento a dar ao sinal de controlo  $u(t)$  a fornecer ao servo. Optou-se por esta solução dadas as vantagens que oferece:

- ✓ Não são necessárias variáveis de elevada resolução para armazenar o resultado de somas (normalmente necessário para a componente integral);
- ✓ Protecção *wind-up*;
- ✓ Transferência *bumpless* simplificada.

A alternativa ao algoritmo incremental exigiria a actualização de um somatório a cada iteração (elemento  $1/s$ ) o que implicaria o recurso a variáveis de elevada dimensão (32 bits) para armazenar o resultado, o que não é favorável em arquitecturas baseadas em microcontroladores.

Além disso poderia ocorrer a requisição de uma posição fora dos extremos do servo ( $-90$  e  $+90^\circ$ ) sem que a integração fosse capaz de inverter a tendência do sinal de controlo uma vez que o seu incremento apenas se limita ao valor do extremo. Só ao fim de algum tempo, a soma pode ser suficiente para inverter a tendência, resultando numa perda significativa na reactividade da resposta. Este fenómeno denomina-se por *wind-up* e é devida à saturação do actuador. Embora haja bastantes soluções para este problema, a mais simples é a do algoritmo incremental, pois não existe qualquer integrador, fazendo com que numa situação de saturação o sinal de controlo  $u(n)$  deixe automaticamente de aumentar resultando imediatamente na inversão da tendência.

Um outro pormenor são as transferências *bumpless*, que não são nada mais do que a activação e desactivação do controlador em pleno funcionamento do sistema. Pretende-se que quando o controlador é ligado ou desligado, o actuador não sofra qualquer variação brusca de posição. Embora não haja nenhum problema na

situação de desactivação do controlador, o problema surge na reactivação. Na solução com integrador, se a integração parar de funcionar durante a desactivação, quando reactivada, o sinal de controlo  $u(n)$  não corresponderá ao sinal  $r(t)$  que anteriormente era aplicado directamente, pois o resultado da integração deixou de ser actualizado, o que resulta num deslocamento brusco para uma posição imprevisível.

Por outro lado, se nunca se parar a integração corremos o risco da soma atingir valores excessivamente elevados ou mesmo de sofrer *overflow*, bastando para isso que os pedidos de actuação nunca correspondam ao valor de *feedback* – tal é frequente na presença de cargas. O resultado reflectir-se-ia em movimentos bastante oscilatórios na reactivação, acabando por provocar o fenómeno de *wind-up*!

Uma possível solução seria, em todas as reactivações, calcular o valor da soma de modo a que o sinal de controlo  $u(t)$  correspondesse ao valor de  $r(t)$  e inicializar a soma com esse valor. No caso do algoritmo incremental, se definirmos o valor  $u(n-1)$  como o último valor aplicado no servo, quer com o controlador ligado, quer desligado, escusamo-nos de qualquer preocupação com este procedimento.

No entanto há uma desvantagem com este algoritmo, resultante do aumento da ordem de um para dois na remoção do integrador. Na presença de ruído, este compensador torna-se mais sensível podendo levar o sistema mais facilmente à instabilidade. Daí a necessidade do formato PI-D em que é o sinal de saída e não o de erro o utilizado na componente derivativa, eliminando qualquer possibilidade de instabilidade resultante de variações do sinal referência  $r(t)$ . Mesmo assim recomenda-se o uso de valores baixos para o parâmetro  $K_D$ .

O elemento “banda morta” presente no compensador, após o cálculo do sinal de erro, também tem como finalidade a prevenção de respostas oscilatórias, resultantes de imprecisões na medição da posição do servomotor. Observou-se que a digitalização deste parâmetro resultava em oscilações até dois graus, daí a introdução deste elemento, para evitar que estas flutuações se transferissem para o compensador.

### 3.4.2 CONTROLO INTEGRAL (I)

Vamos agora testar o controlador para várias cargas e percursos utilizando as trajectórias em forma de rampa e de polinómio de terceira ordem, usando como referência as respostas em malha aberta para a sua avaliação.

Comecemos por utilizar a componente integral, definindo os parâmetros  $K_P$  e  $K_D$  a zero. Experimentando o valor de 0.08 para  $K_I$  para duas massas de elevado valor (Fig. 3.34 e Fig. 3.35) pode-se observar em ambos os casos a eliminação do erro em regime estacionário. No caso da Fig. 3.35 a diferença de dois graus é devida ao efeito da banda morta presente na entrada do controlador. Repare no sinal de saída do controlador solicitando ao servo uma posição mais elevada do que a desejada (curva a rosa), de modo a que ela seja cumprida na presença da carga eliminando assim o erro em regime estacionário.

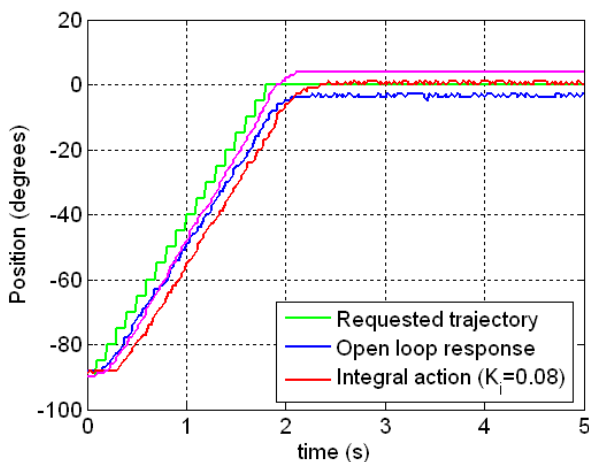


Fig. 3.34: Resposta à rampa com uma carga de 675g ( $K_I=0.08$ ).

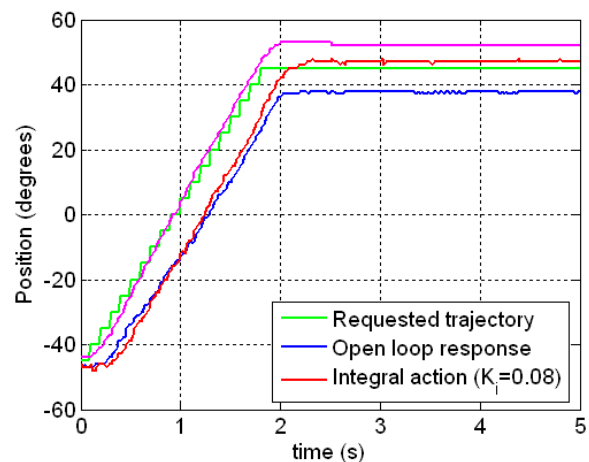


Fig. 3.35: Resposta à rampa com uma carga de 1129g ( $K_I=0.08$ ).

Aumentando o valor de  $K_I$  para 0.20 e realizando o trajecto de  $-45$  para  $+45^\circ$  com uma carga de 1129g (Fig. 3.36 e Fig. 3.37), além da ausência do erro em regime estacionário, o tempo de atraso da resposta

relativamente à trajectória solicitada é melhorado na presença do controlador o que beneficia o tempo de estabelecimento.

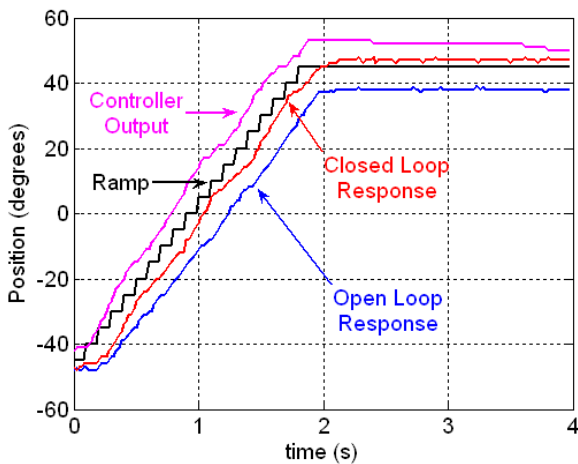


Fig. 3.36: Resposta à rampa com uma carga de 1129g ( $K_I=0.20$ ).

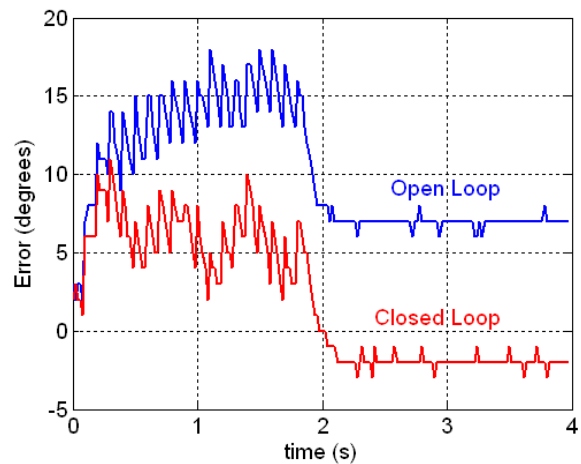


Fig. 3.37: Erro da resposta da Fig. 3.36.

De modo a estudar os efeitos do parâmetro  $K_I$  na resposta do servo, realizou-se uma experiência na qual para uma massa elevada de 924g, efectuando um percurso fixo (-45 a +45°), experimentaram-se vários valores de  $K_I$ . As respostas podem ser visualizadas da Fig. 3.38 à Fig. 3.41.

Por observação, constata-se que para baixos valores, a resposta tende a atrasar-se demasiado, mas à medida que vai aumentando, o tempo de atraso diminui com melhorias significativas relativamente à resposta em malha aberta. O caso da Fig. 3.41 evidencia a resposta típica para valores excessivos do  $K_I$ : durante a fase transitória, tanto a posição medida como o sinal de controlo apresenta-se bastante oscilatório provocando o fenómeno de *overshoot*. No entanto seria desejável conter este *overshoot* de modo a melhorar ainda mais o atraso de cerca de 150ms medido pouco antes de se verificar as oscilações.

Quanto ao erro em regime estacionário, em todos os casos apresentados ele é eliminado, o que sugere que é suficiente a presença do integrador, independentemente do parâmetro  $K_I$ , exceptuando-se, obviamente, o valor nulo.

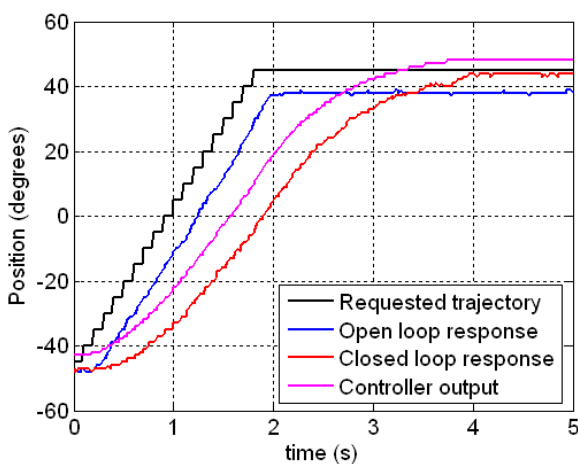


Fig. 3.38:  $K_I=0.02$  ( $m=924g$ ).

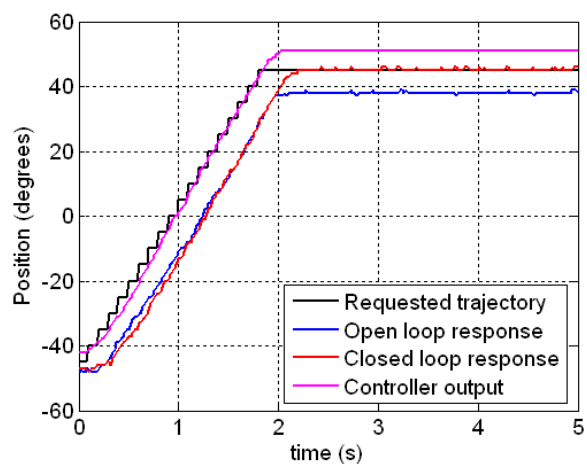


Fig. 3.39:  $K_I=0.07$  ( $m=924g$ ).

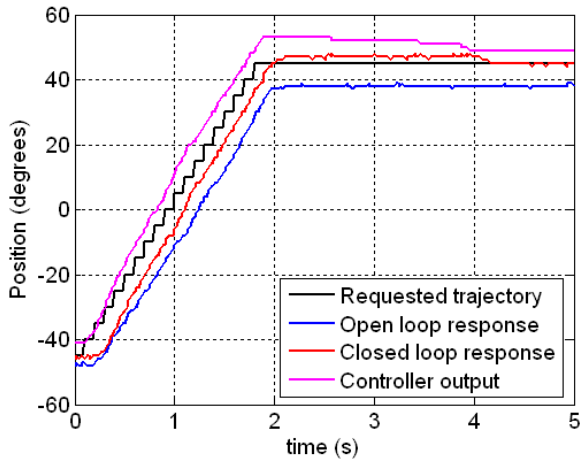


Fig. 3.40:  $K_I=0.15$  ( $m=924g$ ).

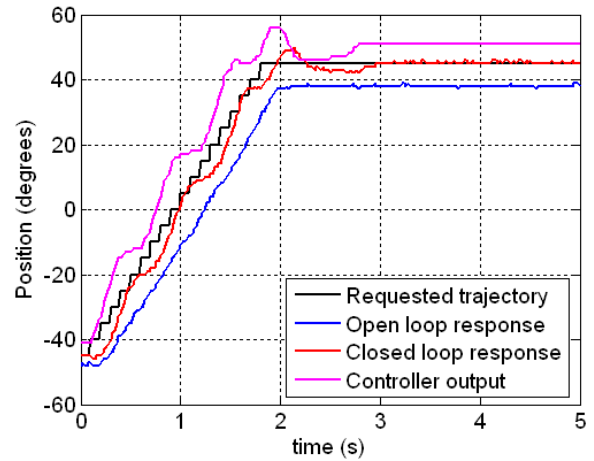


Fig. 3.41:  $K_I=0.30$  ( $m=924g$ ).

### 3.4.3 CONTROLO PROPORCIONAL+INTEGRAL (PI)

Vamos agora adicionar a componente proporcional, mantendo a integral pois é fundamental para a eliminação do erro em regime estacionário. Para melhor percebermos as vantagens da componente proporcional, primeiramente definamos um  $K_I$  de modo a provocar um ligeiro *overshoot* na resposta do servo. A Fig. 3.42 e Fig. 3.43 apresentam um exemplo para uma carga de 924g, para  $K_I=0.10$  e  $K_P=0.04$ .

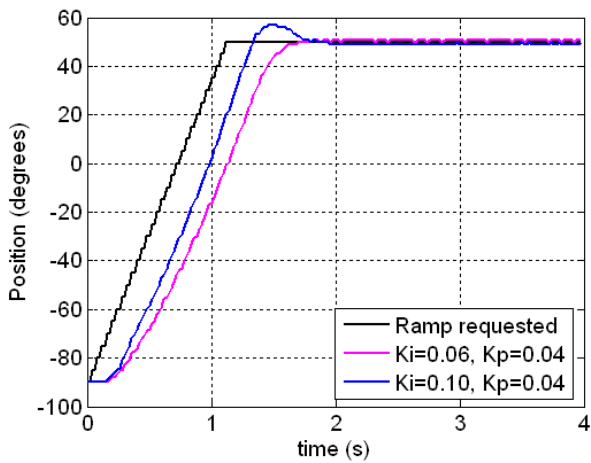


Fig. 3.42: Fenómeno do overshoot para valores de  $K_I$  elevados (carga de 924g)

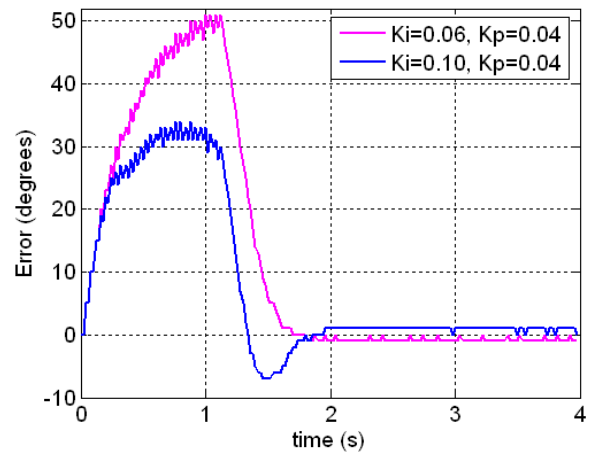


Fig. 3.43: Erro da resposta da Fig. 3.42.

Mantendo fixo o valor de  $K_I$  em 0.10 vamos aumentar o parâmetro  $K_P$  para 0.30 (Fig. 3.44 e Fig. 3.45).

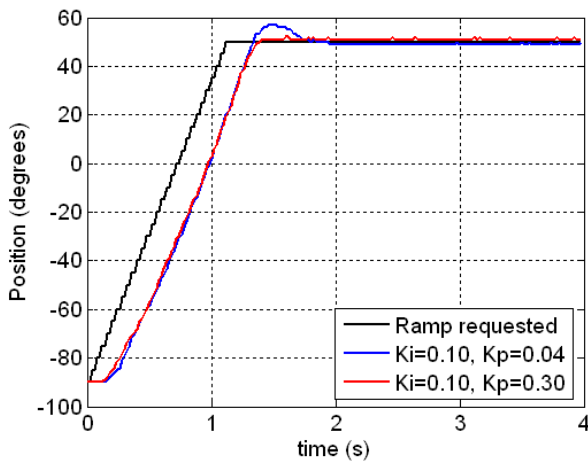


Fig. 3.44: Correção do overshoot com o aumento de  $K_P$  (carga de 924g).

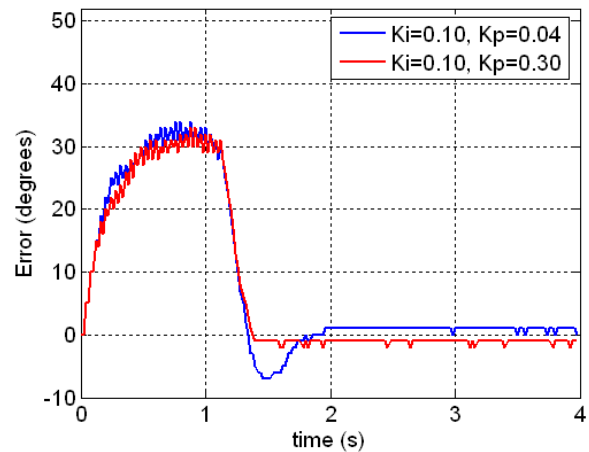


Fig. 3.45: Erro da resposta da Fig. 3.44

Pode-se constatar que o *overshoot* desapareceu sem afectar significativamente o tempo de atraso. De modo a melhor perceber estes resultados registaram-se várias respostas para diferentes valores de  $K_P$  mantendo fixas a trajectória solicitada e a carga. A Fig. 3.46 e a Fig. 3.47 apresentam os resultados para dois valores diferentes de  $K_I$  agora utilizando trajectórias polinomiais.

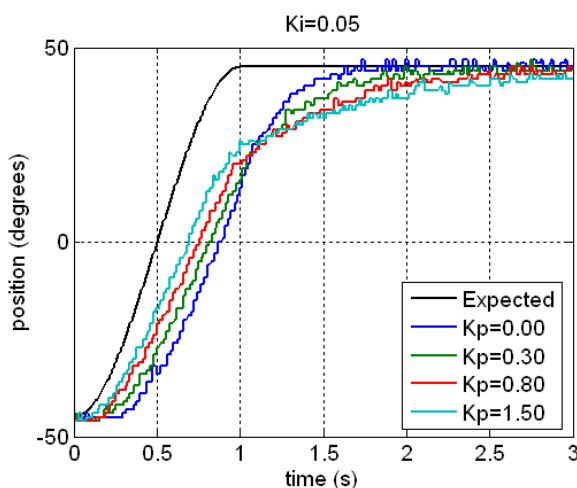


Fig. 3.46: Variação de  $K_P$  para  $K_I=0.05$  (carga de 675g).

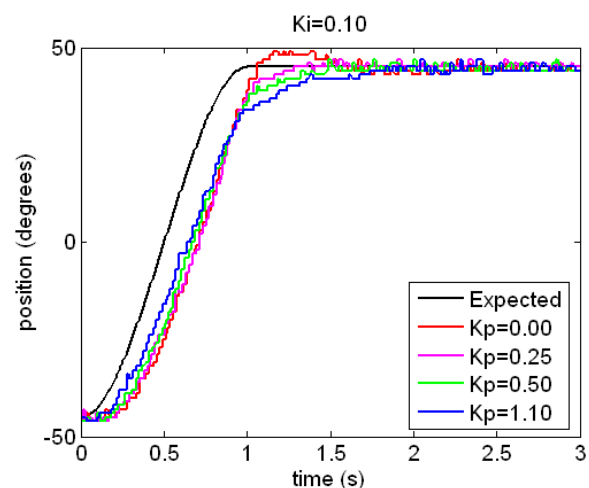


Fig. 3.47: Variação de  $K_P$  para  $K_I=0.10$  (carga de 675g).

Analisando o gráfico da Fig. 3.47 com  $K_I=0.10$  confirma-se o que se havia dito sobre os efeitos no *overshoot*: o *overshoot* é reduzido chegando mesmo a deteriorar o tempo de estabelecimento caso este valor seja excessivo, sem no entanto alterar significativamente o tempo de atraso.

Para um  $K_I$  mais pequeno – 0.05 – (Fig. 3.46), pode-se evidenciar mais claramente os efeitos de um  $K_P$  excessivo em valor: o tempo de estabelecimento é claramente prejudicado levando muito mais tempo a atingir o valor final, mas o tempo de atraso é melhorado acelerando o seguimento da trajectória no seu início. Ocorre, por isso, um ponto de cruzamento entre as diversas respostas quase coincidente com o término da trajectória.

Logo, conclui-se que deve haver um compromisso entre o tempo de atraso e o tempo de estabelecimento de modo a não perder demasiado num dos lados. O parâmetro  $K_P$  deve, por isso, ser escolhido tendo em conta estes aspectos.

### 3.4.4 CONTROLO INTEGRAL+DERIVATIVO (ID)

Substituindo a componente proporcional pela derivativa podemos observar que praticamente não afecta nem a acção integral de eliminação do erro em regime estacionário, nem a acção proporcional de redução do *overshoot*, tal como se observa na Fig. 3.48 e na Fig. 3.49. Dada a não influência na resposta do sistema, não se achou utilidade para este tipo de controlo sem a componente proporcional.

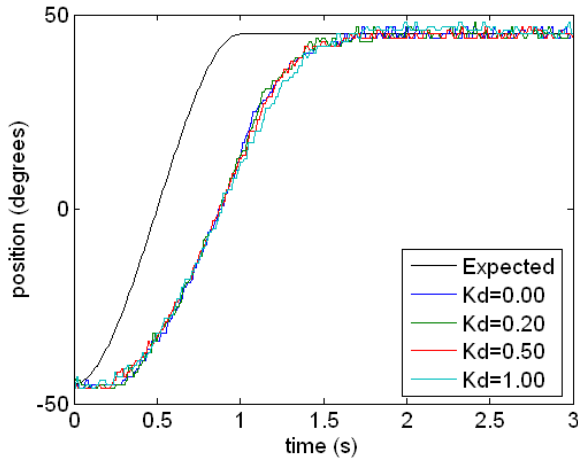


Fig. 3.48: Resposta ao polinómio com  $K_I=0.05$  e  $K_P=0.00$  (carga de 675g).

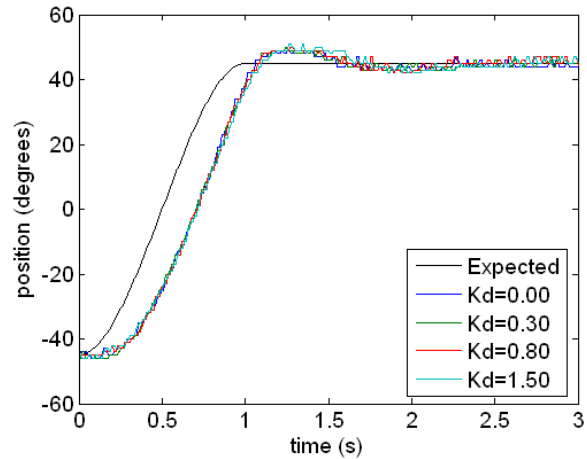


Fig. 3.49: Resposta ao polinómio com  $K_I=0.10$  e  $K_P=0.00$  (carga de 675g).

### 3.4.5 CONTROLO PROPORCIONAL+INTEGRAL+DERIVATIVO (PID)

Introduzindo agora todas as três componentes podemos otimizar todos os parâmetros de modo a encontrar a melhor relação entre:

- Tempo de atraso;
- Tempo de estabelecimento;
- *Overshoot* (estático)
- Oscilação durante a fase transitória (*overshoot* transiente).

Por análise de cada componente do controlador, apresentadas nas secções anteriores, já podemos relacionar estes parâmetros com cada um deles:

- A **componente integral** está directamente relacionada com o tempo de atraso, sendo tanto menor quanto maior for o parâmetro  $K_I$ . No entanto para valores excessivos a resposta instabiliza com a ocorrência de *overshoot*.

Ou seja:

- Diminui o tempo de atraso;
- Aumenta o *overshoot* para valores de  $K_I$  elevados.

- A **componente proporcional** é utilizada para contenção do *overshoot*, sendo capaz também de melhorar o tempo de atraso se  $K_P$  possuir valores elevados, sob pena do tempo de estabelecimento se deteriorar.

Logo, temos:

- Redução do *overshoot*;
- Para valores elevados de  $K_P$ :
  - O tempo de atraso é melhorado;
  - O tempo de estabelecimento deteriora-se.

- Embora ainda não tenha sido demonstrado a utilidade da **componente derivativa**, verificou-se experimentalmente que esta componente confere alguma vantagem na suavidade da resposta transiente.

Combinando estas três componentes deveremos conseguir uma resposta muito próxima da ideal, com um tempo de atraso muito pequeno, um tempo de estabelecimento próxima da duração da trajectória, e um

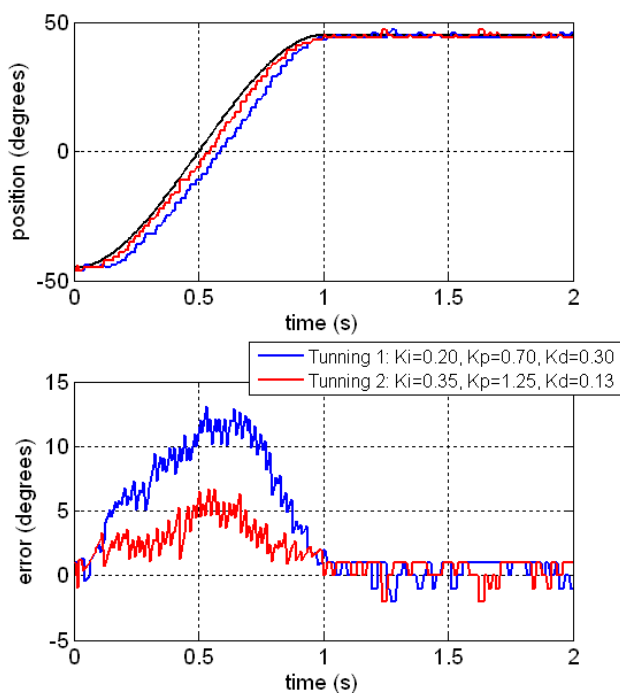
comportamento suave durante a fase transitória.

Como procedimento para fazer o *tunning* da compensação seguiram-se os passos seguintes:

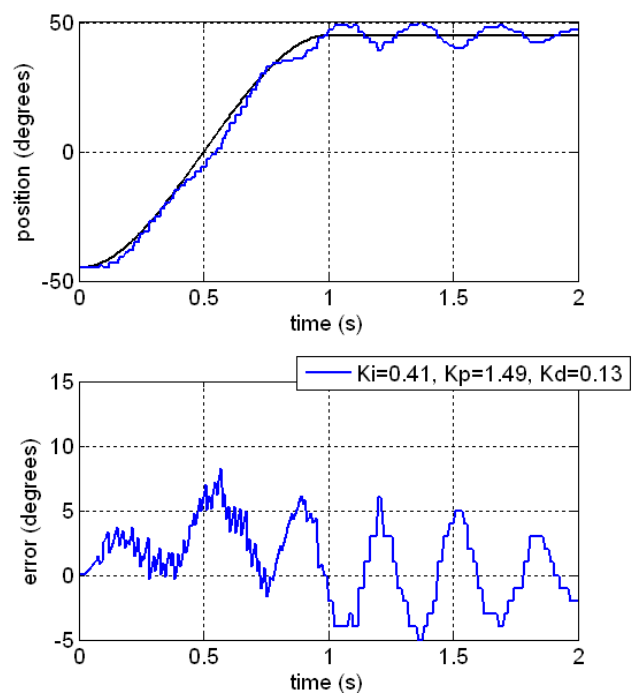
1. Aumentar  $K_I$ , de modo a otimizar o tempo de atraso, até começar a ocorrer *overshoot*;
2. Aumentar o valor de  $K_P$  o suficiente para eliminar o *overshoot*. Não convém utilizar este parâmetro para otimizar o tempo de atraso, uma vez que o tempo de estabelecimento é, ao mesmo tempo, agravado. Deixemos, por isso, essa tarefa à acção integral;
3. Se a resposta transitória ainda não for demasiado oscilante, voltar ao passo 1 para melhorar ainda mais o tempo de atraso;
4. Se a resposta começa a oscilar durante a fase transitória, aumentar o parâmetro  $K_D$  de modo a conferir suavidade durante o percurso;
5. Voltar ao passo 1.

Na Fig. 3.50 comparam-se dois ensaios executados durante o *tunning* correspondentes a um ajuste inicial e a outro final, verificando-se uma melhoria de cerca de  $6^\circ$  no erro máximo em regime transitório. A Fig. 3.51 apresenta um caso de exagero nos parâmetros de compensação levando à instabilidade. De modo a evitar estas situações convém executar o algoritmo de *tunning* em pequenos passos, permitindo assim encontrar mais facilmente os parâmetros óptimos.

A partir de certo ponto, ao qual chamaremos de compensação limite, a melhoria já começa a ser mais exigente com variações muito mais pequenas dos parâmetros de compensação. Quando tal começa a ocorrer considere o *tunning* como terminado com os parâmetros óptimos correspondentes ao ajuste anterior.



**Fig. 3.50:** Comparação entre dois conjuntos de parâmetros durante o *tunning* ( $m=675g$ ).



**Fig. 3.51:** Situação de instabilidade ( $m=675g$ ).

Para a carga de  $675g$  encontraram-se como parâmetros óptimos o conjunto  $K_I=0.39$ ,  $K_P=1.46$  e  $K_D=0.15$  com um erro máximo de menos de  $5^\circ$  (Fig. 3.52). No entanto, note que nos encontramos numa situação em que o sistema torna-se muito susceptível à instabilidade face a perturbações externas. A Fig. 3.53 exemplifica este caso, em que o ensaio com os parâmetros óptimos é repetido duas vezes verificando-se, na última, alguma instabilidade.

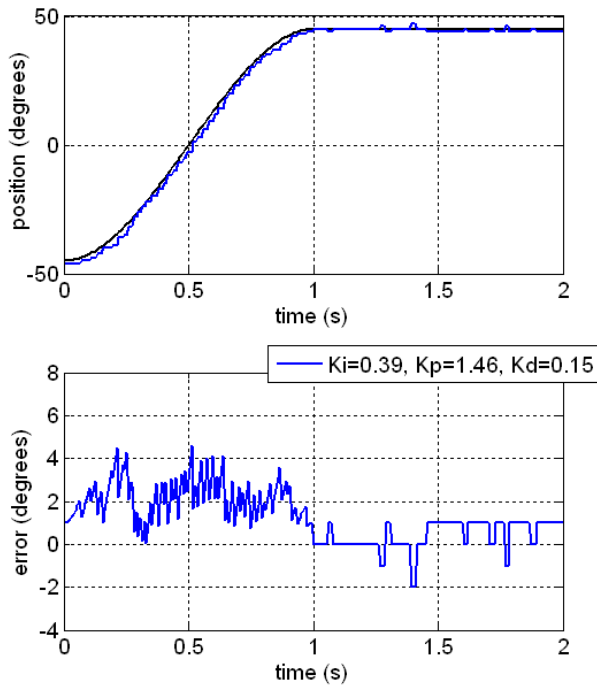


Fig. 3.52: PID otimizado para uma carga de 675g.

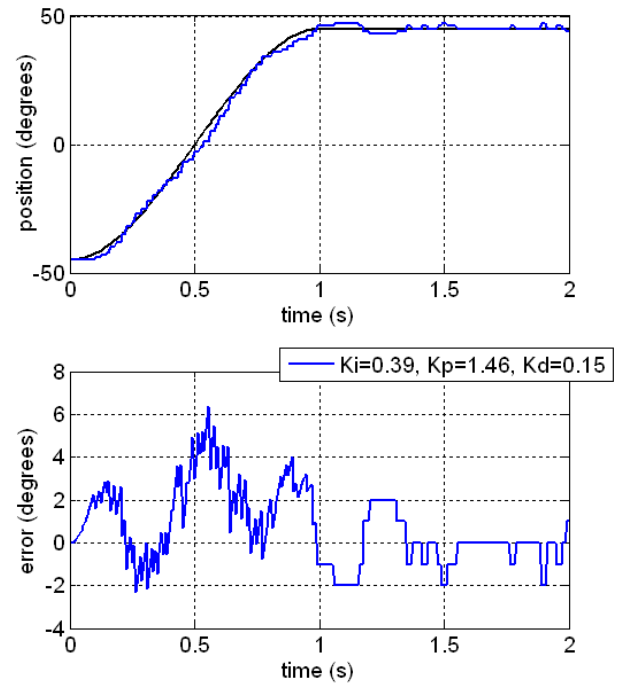


Fig. 3.53: Repetição do ensaio com os parâmetros de compensação óptimos.

### 3.5 APLICAÇÃO DOS ALGORITMOS NO ROBOT HUMANÓIDE

Embora a solução de controlo baseada num PID aparente ter bons resultados, um determinado conjunto de parâmetros de compensação apenas se pode considerar óptimo nas condições em que foi feito o *tuning*:

- Massa da carga;
- Posição inicial e final, ou melhor, variação do binário ao longo da travessia;
- Período do trajecto;
- Tensão de alimentação e corrente máxima fornecida (dependente da fonte de alimentação).

Mudando qualquer um destes parâmetros resulta invariavelmente na alteração da resposta do servo podendo conferir-lhe maior estabilidade ou torná-lo instável:

- O aumento da massa da carga induz à instabilidade, uma vez que o servo tem de fazer um “esforço” maior para a mover, no sentido que tem de aplicar maior binário no motor, e por isso, o tempo de atraso será acrescido. Numa situação destas em que o sinal de *feedback* tem dificuldade em acompanhar a “velocidade” do controlador (definido pelo factor de integração  $K_I$ ), o sinal de controlo tende a saturar facilmente provocando fenómenos de *overshoot*. Por outro lado, se a carga diminuir de massa, o sinal de *feedback* consegue acompanhar perfeitamente o sinal de controlo conferindo uma maior reactividade à sua correcção.
- Se o percurso da trajectória for alterado, a trajectória de binário também muda fazendo com que o servo tenha de despende mais ou menos energia, de acordo com o tipo de percurso a realizar. Esta situação é equivalente à da modificação da carga, na medida que quando o binário a que está sujeito aumenta (aproxima-se do ponto  $0^\circ$ ) é equivalente a aumentar a massa da carga, e vice-versa.
- Além da carga influenciar a resposta do servo, também a velocidade influencia. Tal é devido ao facto, de não só a força gravítica fazer parte do binário exercido no servo, mas também a velocidade e a sua variação (aceleração). Desta forma, diminuindo o período da trajectória, estamos a aumentar a velocidade, o que interfere no binário exercido, induzindo, por isso, numa resposta mais instável.



- Um outro detalhe importante é as condições de alimentação eléctrica dos servomotores. Caso a corrente máxima a fornecer seja limitada ou a tensão de alimentação é baixa, o binário a aplicar no motor para executar um determinado movimento aumenta, introduzindo atraso na resposta e logo maior instabilidade. Para minimizar estes problemas, duas baterias de Lítio de 7.4V são ligadas em paralelo de modo a fornecer uma corrente de 9600 mAh ao sistema.

Como se pode constatar, estamos a lidar com um processo altamente não linear em que as condições iniciais aplicadas nos servomotores estão sempre a mudar, o que é um problema, uma vez que o controlo clássico baseado num PID não entra em consideração com as condições iniciais. Uma forma de dar a volta a esta questão corresponde em actualizar, em tempo real, o valor dos parâmetros de compensação de modo a adaptar o controlador a cada situação específica. Foi pensando nesta questão que se decidiu que a compensação via *firmware* seria a melhor opção, uma vez que é muito fácil mudar os parâmetros de controlo sem intervenções a nível de *hardware* como acontecia se o controlador estivesse implementado fisicamente. De modo a evitar a recompilação do código na modificação destes valores, os parâmetros de controlo são passados a cada *slave* via barramento CAN, sendo a unidade principal, o PC, a responsável por atribuir os valores de compensação mais apropriados a cada acção.

No entanto, outro problema surge: como é que se detectam as situações mais ou menos exigentes em cada junta; e caso consigamos detectá-las, que lei de controlo seguirão os parâmetros de compensação? Para já tentaremos responder à primeira questão. Muito embora, no teste de um só servo, haja uma relação estreita entre binário aplicado e posição, tal deixa de acontecer na presença de várias juntas que se interligam em série por meio de elos, como é o que acontece em cada perna. Além disso, a velocidade da junta também afecta o binário pelo que é preciso discernir cada uma destas fontes. Uma das formas de estimação do binário aplicado baseia-se na medição da corrente consumida por cada servo: quanto maior for o binário aplicado, maior é a corrente consumida pelo que a medição desta grandeza pode ajudar na detecção de situações de elevado ou baixo *stress* sobre as juntas.

Nesta fase, considere-se a implementação destes algoritmos de controlo no microcontrolador conectado a três servomotores correspondentes às seguintes juntas de uma perna humanóide (Fig. 3.54):

- Servo 1: junta do pé (de rotação dianteira);
- Servo 2: junta do joelho;
- Servo 3: junta da anca (de rotação dianteira).

Foram executados vários movimentos, sendo apresentado neste capítulo apenas o de flexão (Fig. 3.54), dada a sua relevância na realização de um passo. As primeiras experiências foram realizadas sem carga e, posteriormente, foi adicionada uma carga de cerca de 2 kg no topo da perna (Fig. 3.55).

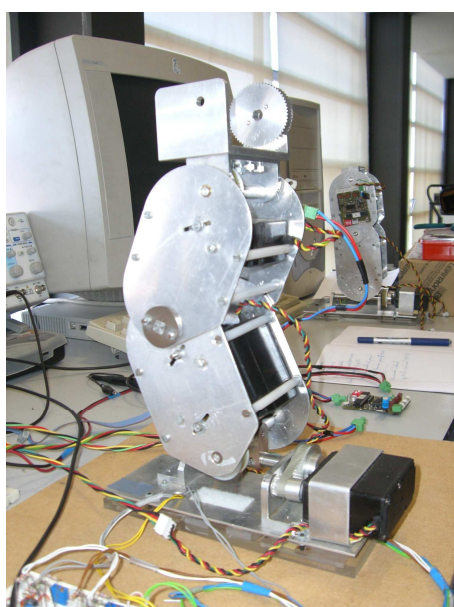


Fig. 3.54: Movimento de flexão de uma perna.



Fig. 3.55: Carga de 2 kg utilizada no topo da perna.

### 3.5.1 MOVIMENTO DE FLEXÃO EM MALHA ABERTA DE UMA PERNA

#### Na ausência de carga

Como se pode visualizar nas figuras seguintes (Fig. 3.56, Fig. 3.57 e Fig. 3.58), na ausência de carga, o erro em regime estacionário é praticamente nulo, praticamente sem a necessidade de praticar controlo externo. Note-se na natureza errática da estimação da velocidade. Tal deve-se ao processo de medição baseada na variação de posição, que infelizmente introduz bastante erro.

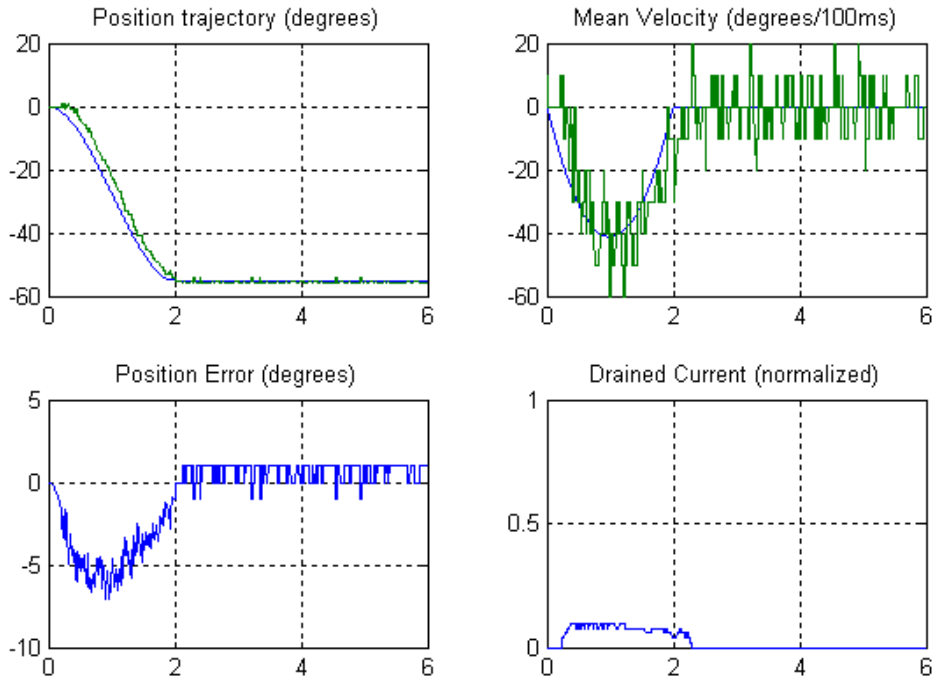


Fig. 3.56: Junta do pé.

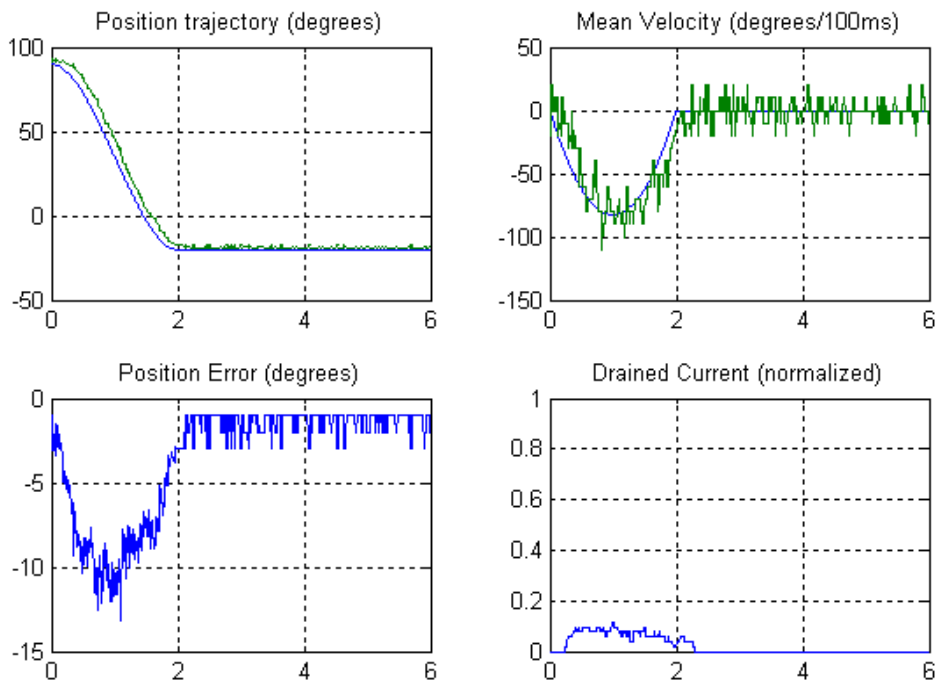


Fig. 3.57: Junta do Joelho.

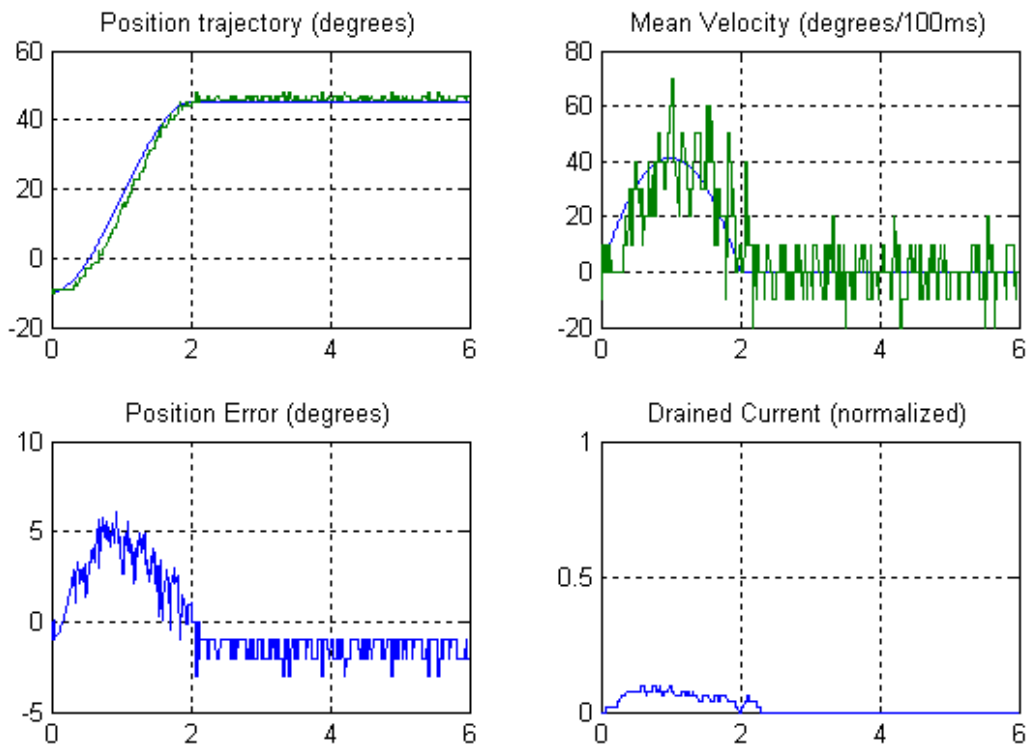


Fig. 3.58: Junta da Anca.

**Na presença de uma carga de cerca de 2 kg**

Com a carga de 2 kg (Fig. 3.59, Fig. 3.60 e Fig. 3.61), surpreendentemente verifica-se pouca variação do erro em regime estacionário, com um baixo consumo de corrente. Note-se que agora o esforço é distribuído pelas três juntas, pelo que as condições não são tão exigentes como as experimentadas no caso anterior.

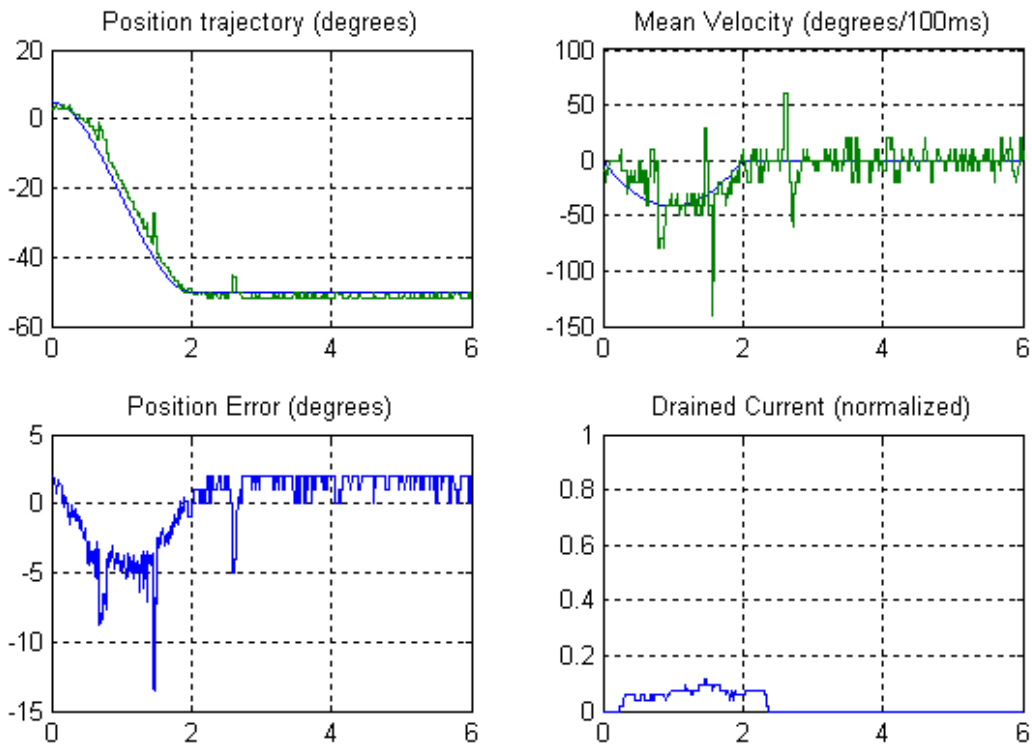
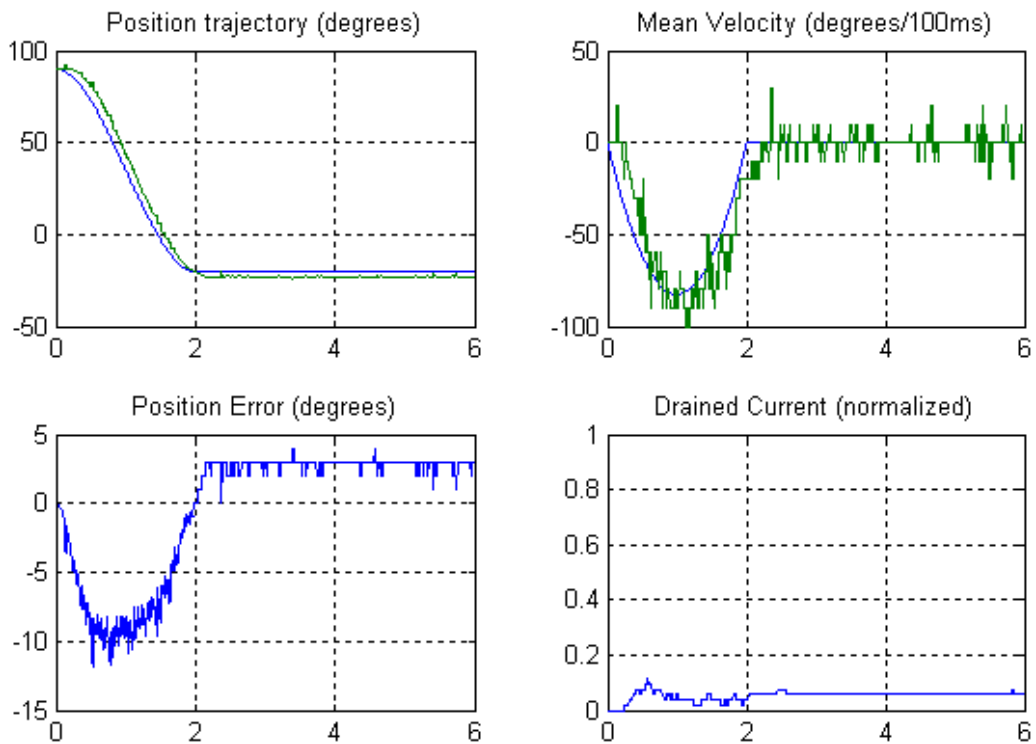
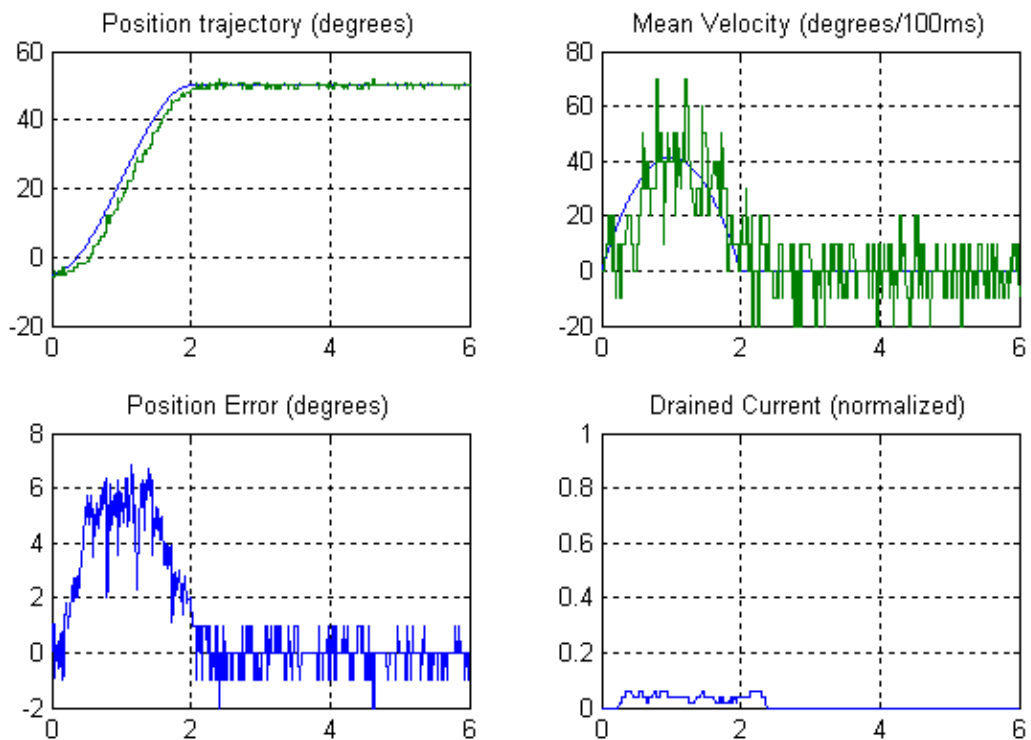


Fig. 3.59: Junta do pé.



**Fig. 3.60: Junta do joelho.**



**Fig. 3.61: Junta da anca.**

Contudo, observa-se na junta do pé (Fig. 3.59), uns picos estranhos na leitura de posição, sinalizando momentos em que o impulso de corrente poderá ter ocupado todo período de PWM. Tal não é óbvio no registo de corrente, uma vez que este representa um conjunto de leituras esporádicas ao longo deste período. Tal indica que a junta do pé é a mais exigente em termos de consumo de corrente.

### 3.5.2 MOVIMENTO DE FLEXÃO EM MALHA FECHADA DE UMA PERNA

As figuras seguintes apresentam o comportamento das juntas, agora com o controlador ligado (Fig. 3.62, Fig. 3.63 e Fig. 3.64). Mais uma vez, o caso mais exigente verifica-se para o caso da junta do pé (Fig. 3.62), com erro em regime transitório de  $14^\circ$  e em regime estacionário de  $5^\circ$ . Com o controlador ligado, a resposta melhorou bastante reduzindo o erro máximo em regime transitório para menos de metade, e eliminando o erro em regime estacionário.

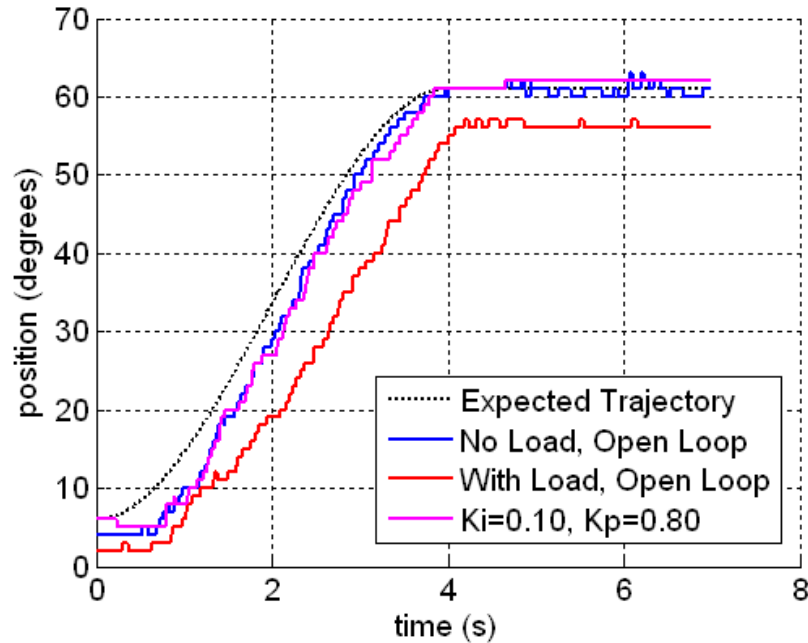


Fig. 3.62: Junta do pé.

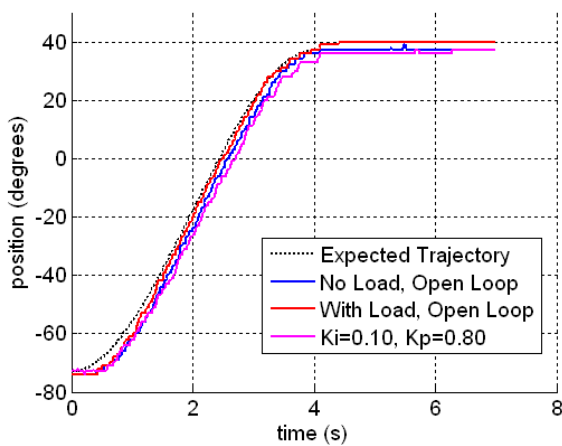


Fig. 3.63: Junta do joelho.

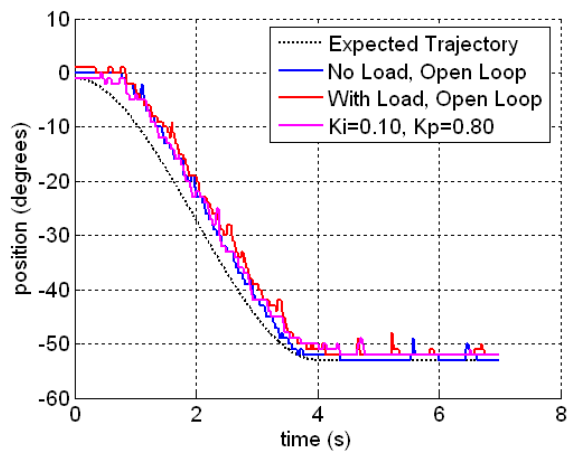
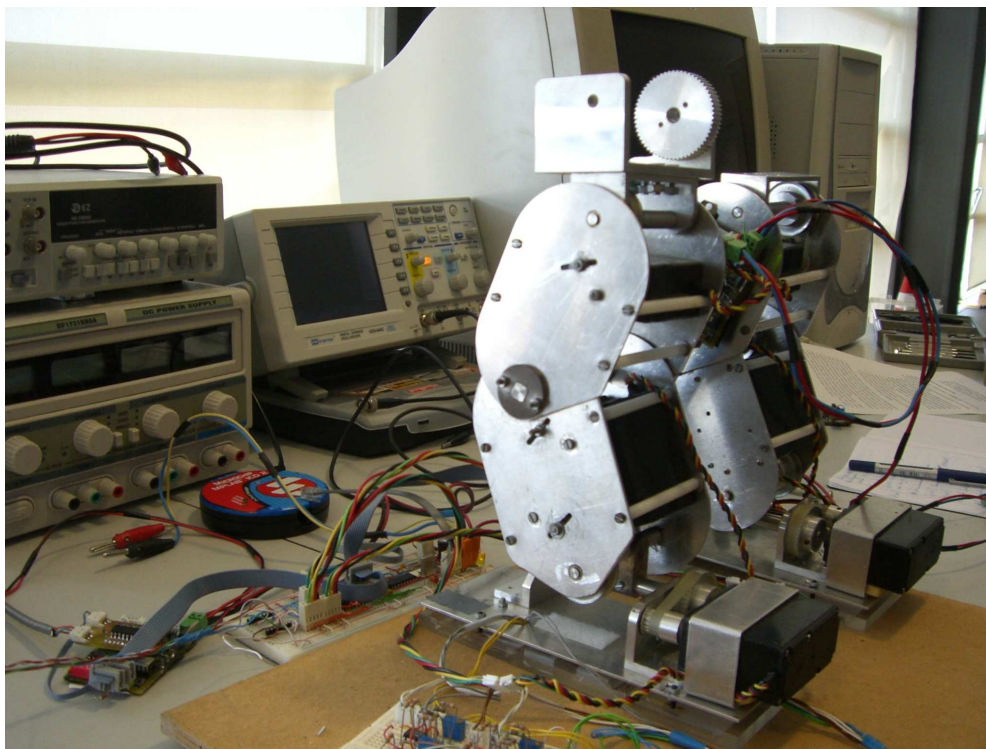


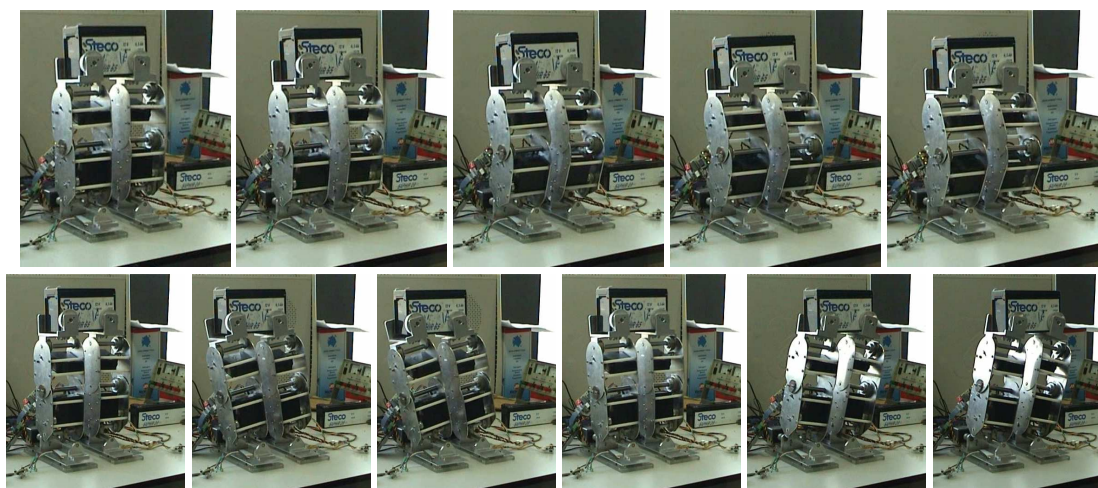
Fig. 3.64: Junta da anca.

### 3.5.3 MOVIMENTO DAS DUAS PERNAS

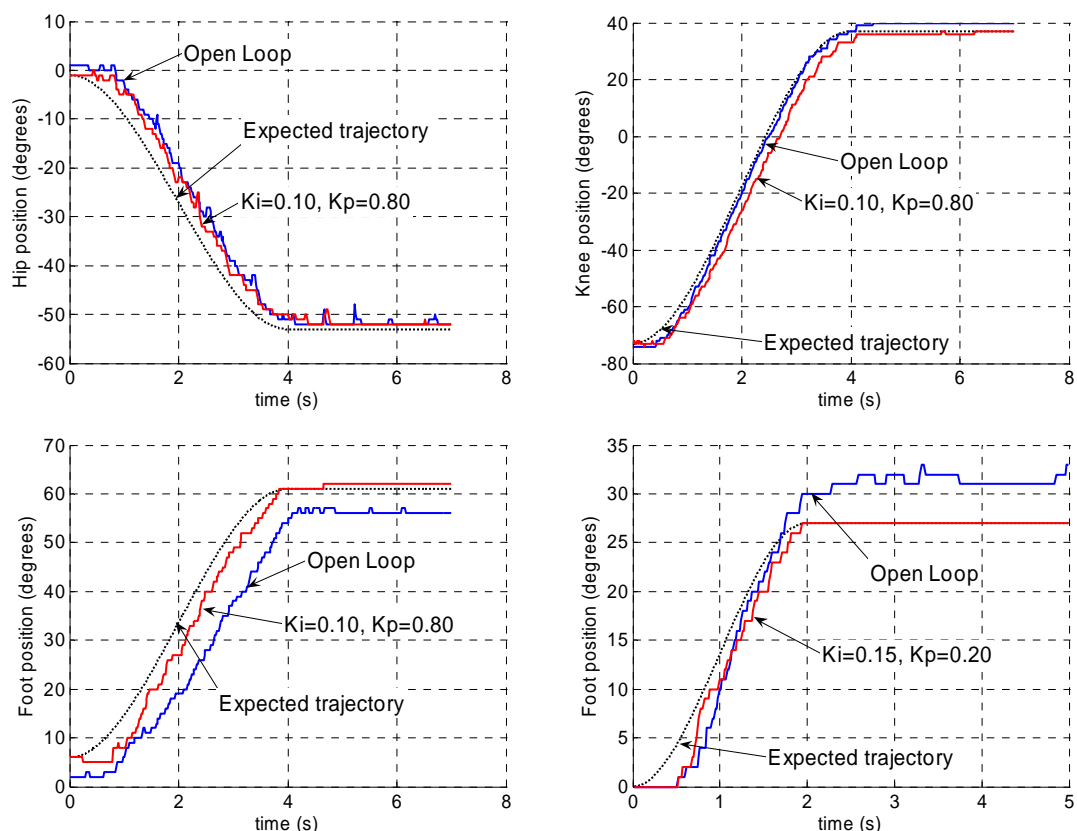
Utilizando agora, as duas pernas juntas (Fig. 3.65), com a aplicação de uma carga de massa partilhada entre os dois membros, executou-se o mesmo movimento de flexão para teste das juntas do pé, do joelho e da anca. Adicionalmente, de modo a testar a junta lateral do tornozelo do pé, realizou-se um deslocamento para o lado. As figuras seguintes (Fig. 3.66) demonstram os resultados referentes a apenas uma das pernas, dado que os da outra perna são bastante semelhantes.



**Fig. 3.65: Movimento de flexão nas duas pernas.**



**Fig. 3.66: Conjunto das duas pernas executando um movimento de flexão (seqüência superior) e um movimento lateral (seqüência inferior) com uma carga de 2.1 kg em completa sincronia de pernas.**



**Fig. 3.67: Resposta ao polinómio com um controlador PI. Imagens superiores e inferior esquerda: comportamento das três juntas envolvidas no movimento de flexão; imagem inferior direita: comportamento da junta lateral do pé na realização do deslocamento lateral.**

Como se pode observar na Fig. 3.67, só com a presença do controlador interno dos servos, a resposta das juntas mais exigentes (juntas do pé) apresentam um comportamento com um apreciável tempo de atraso e de erro em regime estacionário, que é deveras melhorado com a introdução da compensação PI externa. Demonstra-se assim, com estes dados as vantagens da implementação do controlador.

## 3.6 DISCUSSÃO DOS RESULTADOS

Este capítulo teve como objecto de estudo, os servomotores utilizados para actuação. Como são dispositivos relativamente pequenos e baratos que já oferecem controlo de posição, foram escolhidos dada a sua simplicidade de interface com um microcontrolador – podem-se ligar directamente – podendo controlar a sua posição simplesmente através de um sinal digital de PWM cuja largura de pulso define a posição do servo. Os modelos da HITEC, nomeadamente o modelo HS-805BB para as juntas de grande esforço, foram escolhidos dado o seu elevado binário (2.42 N.m @6V) comparativamente a outras marcas.

No entanto, cedo nos apercebemos que estes dispositivos não são perfeitos, não oferecendo qualquer tipo de controlo de velocidade – simplesmente se deslocam para a posição final à sua velocidade máxima – algo nada favorável quando pretendemos controlar a posição e a velocidade das juntas para que realizem movimentos suaves de velocidade limitada. Contudo, pela aplicação de um conjunto crescente (ou decrescente) de pequenos degraus de posição em forma de rampa, conseguimos implementar uma espécie de controlo de velocidade indirecto, na medida em que os extremos de posição da rampa e o intervalo de tempo da sua duração impõem uma velocidade média ao movimento. Uma forma de trajectória mais exigente é o polinómio de terceiro/quinto grau que além de permitir a definição de uma velocidade média garante outros aspectos como é o caso de velocidade nula e acelerações limitadas no início e no fim da trajectória, algo importante para prevenir eventuais picos de corrente e limitar o consumo de corrente durante a realização destas trajectórias.

Mais tarde verificou-se que embora estes actuadores apresentem respostas praticamente ideais (resposta semelhante à trajectória solicitada) na ausência de cargas, na sua presença tal deixa de ser verídico verificando um acentuado tempo de atraso no acompanhamento da trajectória solicitada e uma dificuldade em atingir o valor final devido à presença da força gravítica no binário aplicado (erro em regime estacionário não nulo). Usando o sinal de *feedback* de posição que o servo disponibiliza, por comparação com a posição desejada, tentou-se implementar uma lei de controlo que permitisse corrigir estes desvios. Esta lei de controlo baseou-se num compensador clássico do tipo PID (Proporcional+Integrador+Derivador). Por meio de diversos ensaios, conseguiu-se associar cada componente às características da resposta de modo a permitir escrever um procedimento para rapidamente se encontrar um conjunto adequado de parâmetros.

No entanto um conjunto de parâmetros do controlador só se adequa a uma situação específica de carga, trajectória e velocidade específicos, pelo que seria importante detectar os vários cenários possíveis para que o controlador se adaptasse. Tal seria possível pela medição do binário aplicado no motor que varia de acordo com a inércia presente. Usando o sinal de saída do potenciômetro interno do servo, conhecido por estar relacionado com a sua posição, descobrimos que além de nos fornecer a posição também nos providencia a corrente consumida pelo dispositivo, que por sua vez está relacionada com o binário aplicado ao motor. Desta forma poderíamos aplicar um controlador de força que adaptaria os parâmetros de controlo de acordo com a corrente drenada pelo motor. No entanto este sinal apresenta uma natureza bastante oscilatória, ainda mais com o controlo de posição activado, estando relacionada não só com o binário resultante da força gravítica como também do que resulta da velocidade e da aceleração. Para já desconhece-se qual seria o formato da lei de controlo deste adaptador, nem que parâmetros deverá assumir pelo que mais investigação será necessária sobre este tópico.

Mesmo assim, nas experiências realizadas sobre as pernas do humanóide são utilizados parâmetros fixos do controlador de posição, que claro, são bastante limitados para garantir a estabilidade em qualquer uma das juntas, devendo, estes valores, corresponder ao pior caso. Infelizmente ficamos sempre a perder, dado que os parâmetros não são os mais optimizados introduzindo atrasos adicionais que por vezes podem ser piores do que se o controlador estivesse desligado (em malha aberta). Efectuando ensaios sobre as pernas acabámos por verificar que as exigências em termos de binário são bastante inferiores às testadas com o servo isolado, com respostas de bastante qualidade mesmo em malha aberta. Apenas a estrutura mecânica das pernas se revelou incapaz de executar movimentos com as cargas mais pesadas provocando folgas, que embora algumas fossem ajustáveis, outras revelaram-se sem solução com a necessidade de substituir peças. Esperemos que no trabalho futuro que se segue, estas contrariedades sejam levadas a sério não só no sentido de eliminar quaisquer folgas existentes, como também em aplicar novas técnicas e materiais que atribuam robustez à estrutura, independentemente da solicitação imposta.

Outros aspectos que precisam de ser limados prendem-se com a alimentação dos actuadores e das unidades de controlo local. Mesmo com o uso de duas baterias de elevado fornecimento de corrente em paralelo, verifica-se esporadicamente um aparente *reset* de alguns microcontroladores, na medida que por vezes alguns servos deslocam-se para a posição original a alta velocidade. Por outro lado, muito embora o comportamento dos servomotores possam ser satisfatórios, em termos mecânicos ainda há muitos aspectos que precisam de ser melhorados sob pena de tornar ineficaz os algoritmos de controlo de posição e velocidade. Aconselha-se, por isso, uma séria revisão à estrutura das pernas, uma vez que são estes membros que terão de suportar os esforços mais exigentes, nomeadamente:

- No aperto das correias de transmissão: estas são as principais causas de folgas, aumentando à medida que se verifica um esforço, correndo o risco de provocar saltos de dentes. Uma solução que já foi indicada corresponde a efectuar o aperto de forma dinâmica com o recurso a uma espécie de mola, tornando automático o processo de ajuste.
- Ao nível das ligações entre os veios e os elos tornando-os mais robustos quer pelo aumento da resistência dos materiais quer pelo aumento do número de ligações. Tal convêm ser estudado;



## Capítulo 4

# Sistema de Controlo Baseado nas Forças

### 4.1 INTRODUÇÃO

O Capítulo 3 descreveu a implementação do sistema de controlo de baixo nível e as técnicas de compensação relativas à posição e velocidade dos servomotores de modo a corrigir desvios introduzidos por estes dispositivos na realização de trajectórias. Neste capítulo apresentam-se vários algoritmos e estratégias visando a implementação de um nível de controlo intermédio baseado nas forças de interacção entre o pé e o solo. Este é um tópico em que tem sido realizada intensa investigação no sentido de dotar os sistemas de locomoção bípede com maior robustez e versatilidade de operação [18-19].

Tendo em conta os objectivos traçados inicialmente, pretende-se explorar a importância e as possibilidades que resultam da utilização de informação de força para executar duas tarefas elementares. Em primeiro lugar, pretende-se desenvolver um controlador que assegure o equilíbrio de uma perna assente num plano de inclinação variável. A estratégia proposta segue o comportamento do corpo humano que, pela sensação das forças aplicadas sobre os pés, procura o equilíbrio do corpo pelo movimento das juntas das pernas e do tronco. Em segundo lugar, serão avaliadas as possibilidades de realização de movimentos simples com uma perna tendo por base um conjunto mínimo de variáveis de referência. O método de controlo proposto é baseado na medição da informação de vários sensores de força localizados na base de cada pé, que permitem determinar o ponto onde se localiza o centro de pressão. Para baixas velocidades, este valor deverá coincidir com a projecção do centro de massa da estrutura sobre os pés.

Ao longo deste trabalho foram estudados dois tipos de controladores: o primeiro baseado numa abordagem bastante simplista seguindo uma lei proporcional e, o segundo, um tanto mais complexo, que procura relacionar com mais precisão as forças aplicadas sobre cada pé com a velocidade angular a aplicar nas juntas, usando para tal uma matriz Jacobiana [20]. Nesta fase do trabalho apenas se testaram os algoritmos desenvolvidos para uma das pernas, dado que se confia que estes algoritmos podem ser aplicados de forma igual a cada perna sem perda de generalidade.

### 4.2 ESTRUTURA DOS PÉS

#### 4.2.1 SENSORES DE FORÇA: OS EXTENSÓMETROS

Pretende-se desenvolver um pé que seja capaz de medir as forças de reacção aplicadas ao longo da sua área, de modo a permitir avaliar a posição do centro de pressão aplicado. Para medição das forças de reacção, a solução mais económica e viável consiste na utilização de extensómetros (Fig. 4.1) que são basicamente resistências cujo valor óhmico varia com a sua deformação.



**Fig. 4.1: Exemplo de um extensómetro típico.**

Aproveitando esta propriedade elástica que relaciona proporcionalmente a deformação com a força aplicada, podemos colocar vários extensómetros ao longo da área de cada pé, e, pela medição da sua deformação, medir a força aplicada sobre cada um deles. No entanto, note que esta relação proporcional apenas é válida para uma determinada zona de funcionamento denominada por zona elástica (Fig. 4.2). Apenas nesta zona se pode garantir que o material recue sempre ao seu estado inicial após uma deformação, e que assegure que o valor desta seja sempre igual para uma determinada força imposta.



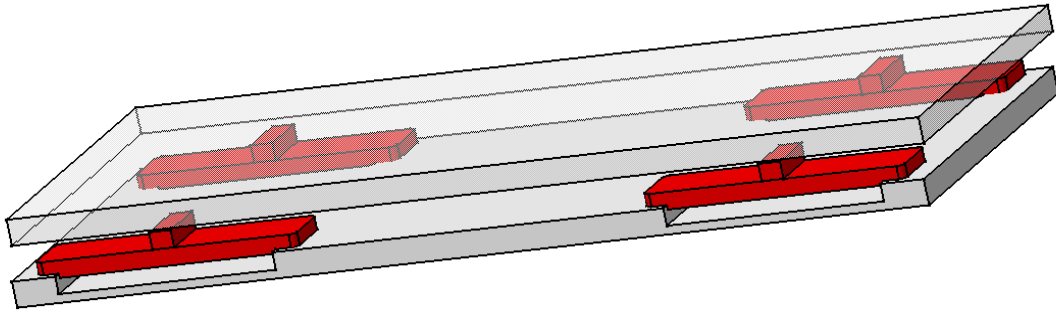
**Fig. 4.2: Relação entre força imposta e deformação elástica.**

Para medição da deformação, pelo seu valor óhmico, uma ponte de *Wheatstone* foi utilizada, onde duas das resistências foram substituídas por extensómetros. A ponte permite medir uma diferença de potencial aplicada nos seus extremos que varia com o valor resistivo. De notar que os extensómetros normalmente deformam-se em cerca de 2 a 5% do seu valor, estando assim limitados a medir pequenas deformações, pelo que se torna necessário amplificar a saída com o recurso a um amplificador de instrumentação para acomodar este sinal à ADC do microprocessador.

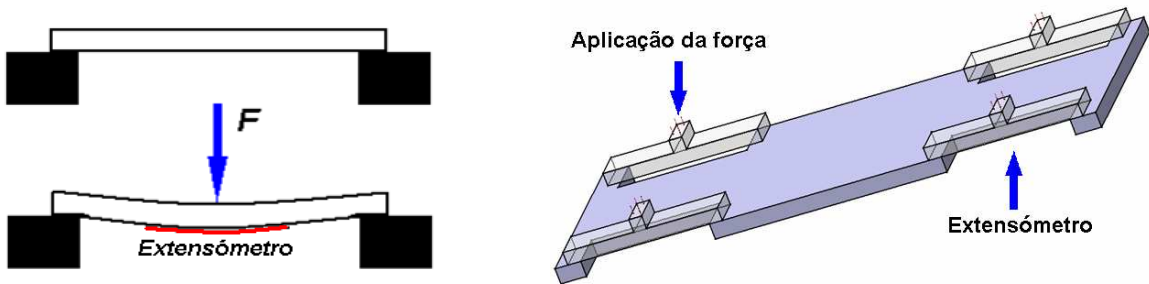
## **4.2.2 O PÉ COMO ELEMENTO SENSITIVO A FORÇAS EXTERNAS**

A ideia essencial é tornar cada pé sensível a forças externas, de modo a poder determinar parâmetros como a inclinação do pé ou a presença de perturbações externas na perna, de modo a accionar os mecanismos necessários para compensar esses desvios. Para tal utilizou-se a estrutura do pé desenhada no ano lectivo 2004/05 em que são usados quatro sensores de força, um em cada extremidade do pé.

A estrutura do pé baseia-se na presença de duas plataformas paralelas entre si: uma inferior que interage com o solo, e uma superior que faz a ligação ao tornozelo (Fig. 4.3). Na plataforma inferior quatro aberturas em formato rectangular foram feitas, uma em cada canto da plataforma, que servem de apoio a pequenas peças de igual formato, mas de comprimento ligeiramente superior, de modo a não penetrar nas aberturas. A Fig. 4.4 exemplifica a colocação destas pequenas peças que deverão possuir propriedades elásticas de modo a que, na presença de uma força externa, sejam capazes de se deformar usando todo o espaço livre da abertura.

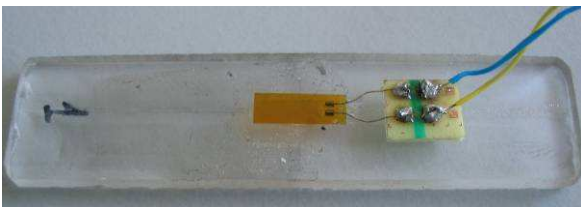


**Fig. 4.3: Estrutura completa de um pé.**

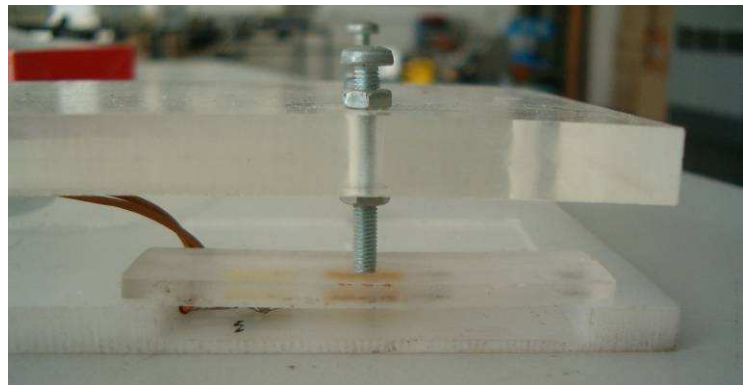


**Fig. 4.4: Colocação dos sensores na estrutura do pé.**

Cada extensómetro é colocado na parte inferior de cada peça, medindo, deste modo, a sua deformação, ou seja, a força aplicada. A Fig. 4.5 apresenta o plano inferior das peças utilizadas, já com o extensómetro colado. O acrílico foi o material escolhido para estas peças dado o seu elevado grau de elasticidade.



**Fig. 4.5: Peça de acrílico contendo o extensómetro para medição da sua deformação.**



**Fig. 4.6: Pontos de contacto entre as 2 plataformas do pé.**

Estas peças deformáveis são os únicos pontos de contacto da plataforma superior, usando para isso parafusos que assentam directamente sobre o centro destas peças. Desta forma, as forças aplicadas sobre a plataforma superior distribuem-se unicamente pelas peças de acrílico, com os extensómetros a medir a força total aplicada. A Fig. 4.6 apresenta um ponto de contacto, em que se observa o parafuso directamente assente sobre a localização do extensómetro de modo a poder medir a força aplicada nesse ponto.

Toda a estrutura está visualizada na Fig. 4.7 com cada extensómetro a medir a força aplicada em cada canto do pé, permitindo, posteriormente, efectuar uma estimativa do centro de pressão a partir da distribuição das quatro forças.

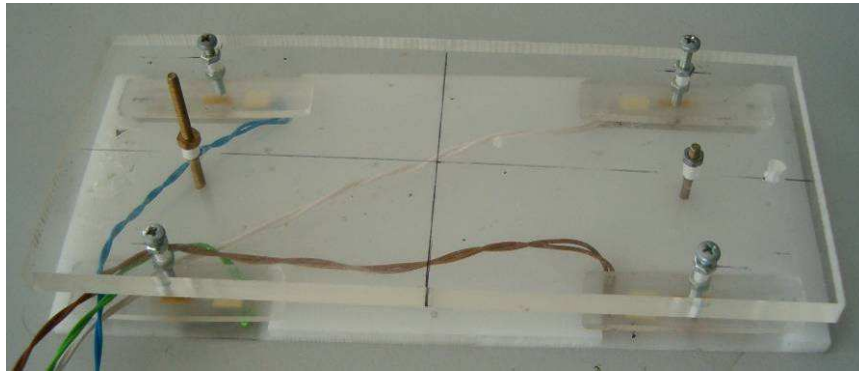


Fig. 4.7: Visão completa do pé.

## 4.3 MEDIÇÃO DAS FORÇAS DE REACÇÃO

### 4.3.1 INTERFACE ENTRE OS EXTENSÓMETROS E O MICROCONTROLADOR

Agora que é possível medir as forças aplicadas ao longo da base de um pé é necessário amostrar estes sinais através do microcontrolador fazendo uso destes sensores.

#### Versão original do circuito de acondicionamento de sinal

A Fig. 4.8 apresenta o circuito de acondicionamento de sinal para cada extensómetro, utilizado no ano lectivo 2004/05 para aplicação numa entrada analógica do microcontrolador.

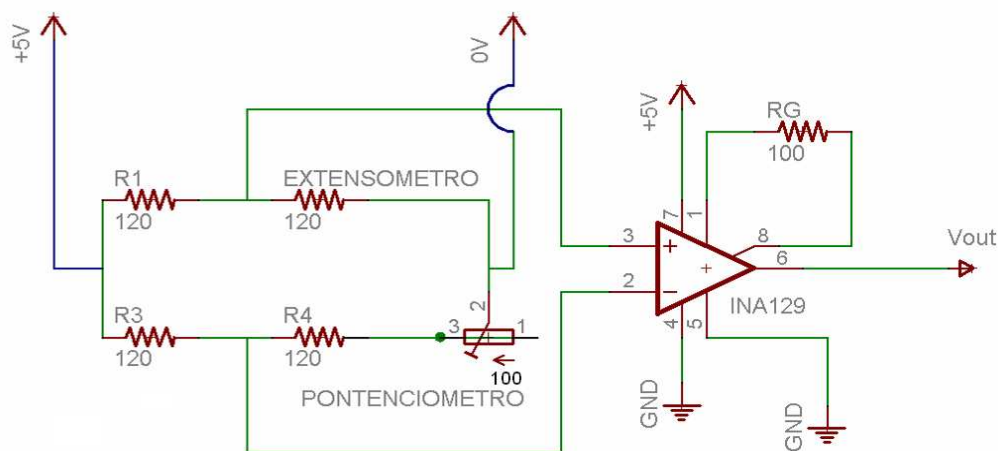


Fig. 4.8: Circuito de interface original dos extensómetros.

Uma ponte de *Wheatstone* é utilizada para medir a resistência óhmica do extensómetro pela sua queda de tensão. Numa situação de equilíbrio, a tensão nos terminais 2 e 3 do amplificador devem ser iguais, pelo que...

$$v_3 = \frac{EXT}{R_1 + EXT} \cdot V_{DD}$$

$$v_4 = \frac{R_4 + POT}{R_3 + R_4 + POT} \cdot V_{DD} = \frac{R_{CAL}}{R_3 + R_{CAL}}, \quad \text{com } R_{CAL} = R_4 + POT$$

$$\begin{array}{l}
 v_3 = v_4 \\
 \frac{EXT}{R_1 + EXT} = \frac{R_{CAL}}{R_3 + R_{CAL}} \\
 EXT \cdot R_3 = R_1 \cdot R_{CAL}
 \end{array}$$

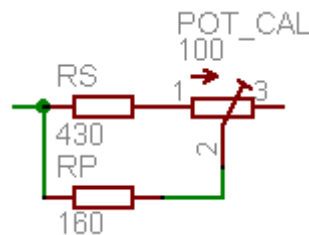
em que  $EXT$  correspondente ao valor ôhmico do extensómetro e  $R_{CAL}$  relacionado com o potenciómetro de calibração.  $R_3$  e  $R_4$  referem-se a outras resistências de valor fixo, e  $v_3$  e  $v_4$  é a tensão presente à saída da ponte.

Contudo, este equilíbrio de que falamos pode não corresponder necessariamente à ausência de força aplicada sobre os sensores, uma vez que pode existir já uma pré-carga, dada por exemplo pelo peso da plataforma superior do pé e do restante corpo. No entanto, tal não é problemático, uma vez que apenas nos interessa a distribuição de forças pelos sensores. Dada a existência desta pré-carga, torna-se necessário calibrar a ponte de *Wheatstone* através do potenciómetro  $POT$  de modo a que a diferença de potencial na saída da ponte seja nula durante equilíbrio.

A presença do amplificador de instrumentação INA129P justifica-se pelo facto da gama de excursão do sinal de saída ser demasiado pequena para processamento directo por parte do microprocessador PIC. Repare que a resistência do extensómetro apenas varia até um máximo de 5% do seu valor standard ( $120\Omega$ ), pelo que um ganho de amplificação é indispensável para acomodar o sinal à excursão 0-5V da ADC do microprocessador.

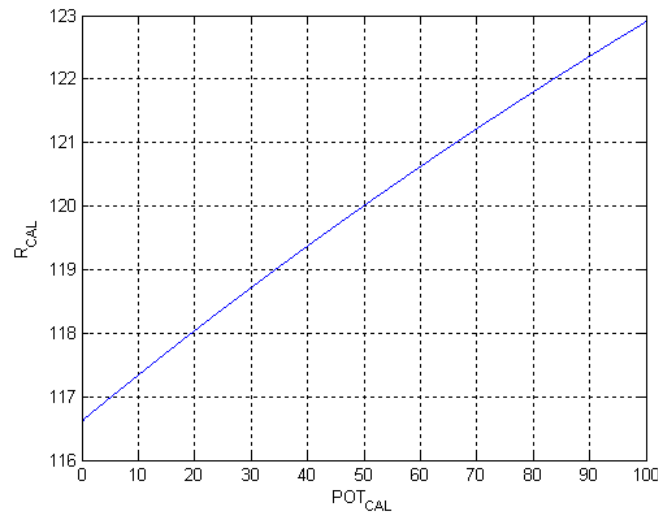
### Versão melhorada do circuito de acondicionamento de sinal

No entanto, a configuração da Fig. 4.8 apresenta vários problemas quando aplicada na prática. Um dos problemas está relacionado com o potenciómetro  $POT$  usado na calibração: permite variar a resistência de calibração  $R_{CAL}=R_4+POT$  entre 120 e  $220\Omega$  o que representa uma enormidade quando comparado com a variação que o extensómetro pode apresentar – 114 a  $126\Omega$  ( $\pm 5\%$  de  $120\Omega$ ). Isto confere ao potenciómetro uma resolução muito baixa, mesmo sendo multi-volta, além de que não é possível calibrar para valores inferiores a  $120\Omega$ . O mesmo problema se apresenta se retirarmos a resistência  $R_4$ , não sendo possível, agora, calibrações acima dos  $120\Omega$ . Para melhorar o procedimento de calibração substituiu-se o conjunto  $R_4+POT$  pelo circuito da Fig. 4.9, em que  $R_S$  se refere a uma resistência em série com o potencómetro  $POT$ , e  $R_P$  encontra-se em paralelo com o conjunto  $R_S+POT$ .



**Fig. 4.9: Circuito de calibração da ponte de Wheatstone.**

A Fig. 4.10 revela a excursão da resistência equivalente deste conjunto quando se varia  $POT\_CAL$  em toda a sua gama. Podemos assim observar que a calibração do ponto de equilíbrio é em muito facilitada, dado que percorre a gama dos  $\pm 5\%$  do valor nominal dos extensómetros.

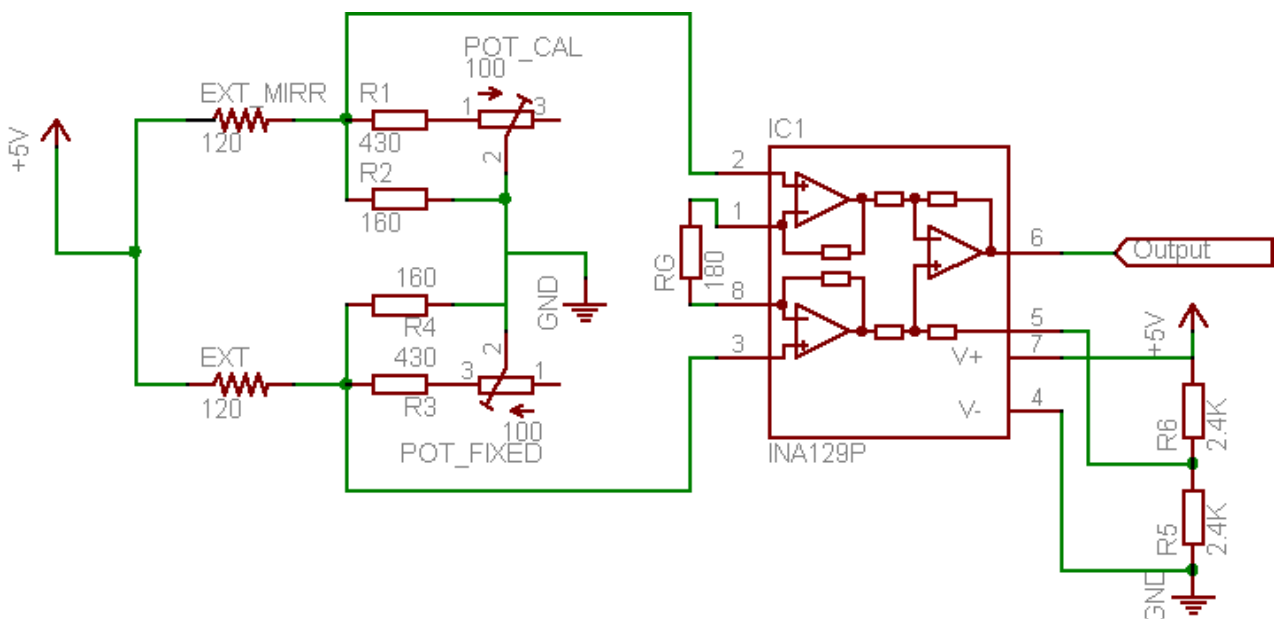


**Fig. 4.10: Relação entre o potenciômetro e a resistência total de calibração.**

Outro problema que se evidencia é a ausência de simetria na ponte de Wheatstone, por parte dos componentes utilizados, mais concretamente no que respeita aos coeficientes de temperatura das resistências, originando assim, flutuações na diferença de tensão à saída devido a variações de temperatura.

Repare-se que, pela expressão de equilíbrio calculada atrás da expressão ( $EXT \cdot R_3 = R_1 \cdot R_{CAL}$ ), se o produto  $EXT \times R_3$  não variar na mesma proporção que o produto  $R_1 \times R_{CAL}$  após uma modificação de temperatura, o sinal de saída é alterado. Tal é provocado por diferentes coeficientes de temperatura nas resistências. Embora estes coeficientes sejam, por natureza muito baixos, não nos podemos esquecer que a ordem de grandeza do sinal de saída da ponte é igualmente baixo o que pode tornar significativo este desvio.

Experimentalmente comprovou-se que a presença do potenciômetro era a causa destas flutuações, o que sugere um coeficiente de temperatura deste componente muito diferente dos restantes. De forma a resolver este problema, definiu-se uma nova configuração da ponte, de modo a que, na presença de qualquer interferência externa, os diversos componentes compensassem o desvio anulando-o.



**Fig. 4.11: Circuito de acondicionamento de sinal com simetria completa.**

A Fig. 4.9 apresenta a nova geometria com a adição de um segundo potenciômetro (potenciômetro fixo *POT\_FIXED* que se pretende não regulável pelo utilizador) e de um segundo extensômetro (extensômetro espelho *EXT\_MIRR* igual ao aplicado para medição) para compensar eventuais desvios introduzidos aos primeiros.

### 4.3.2 RESULTADOS EXPERIMENTAIS

Utilizou-se a estrutura do pé indicada na Fig. 4.7 com os seus quatro sensores ligados a circuitos de acondicionamento de sinal usando o circuito da Fig. 4.9. Por sua vez, as suas saídas estão ligadas a quatro entradas analógicas da placa de controlo local (Fig. 3.2). A leitura digital segue o algoritmo mais simples: conversão analógico-digital para um valor de 8 bits e envio dos resultados dos quatro sensores para a unidade de controlo principal (PC).

Dois tipos de experiências foram realizadas:

1. A primeira, colocando vários pesos, previamente calibrados, sobre cada sensor: cada sensor foi testado de forma isolada (Fig. 4.13);
2. E a segunda, colocando os pesos sobre o ponto central da base do pé: os quatro sensores foram medidos simultaneamente (Fig. 4.14).

A lista de pesos calibrados é indicada a seguir na Tabela 15: os pesos 1 a 4 correspondem a cilindros de metálicos e o peso B corresponde a uma bateria pesando aproximadamente 2 kg. Usaram-se estes pesos de forma isolada e combinados uns com os outros de modo a obter uma massa máxima de cerca de 4 kg.

<i>Carga</i>	<i>Massa (g)</i>	<i>Peso (N)</i>
0	0	0.00
1	249	2.44
2	454	4.45
3	713	6.99
4	666	6.53
B	1946	19.1

**Tabela 15: Lista de pesos usados no teste dos sensores de força.**

A Tabela 16 indica os valores registados em cada uma das experiências para cada sensor, com o azul e o verde localizados na parte dianteira e o branco e castanho na parte traseira do pé (Fig. 4.12). Todos os sensores foram calibrados com o valor 40 na ausência de carga.



Fig. 4.12: Localização dos sensores no pé.

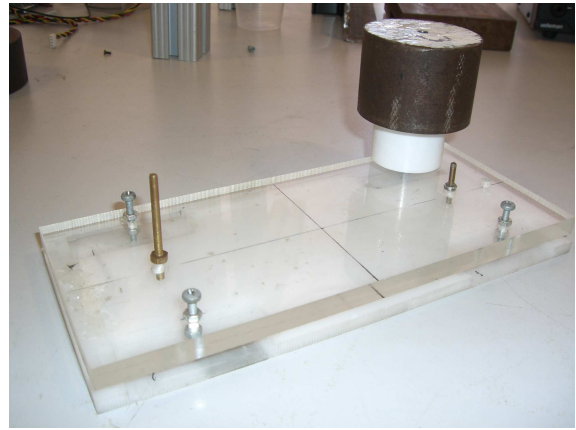


Fig. 4.13: Medição independente de cada sensor.

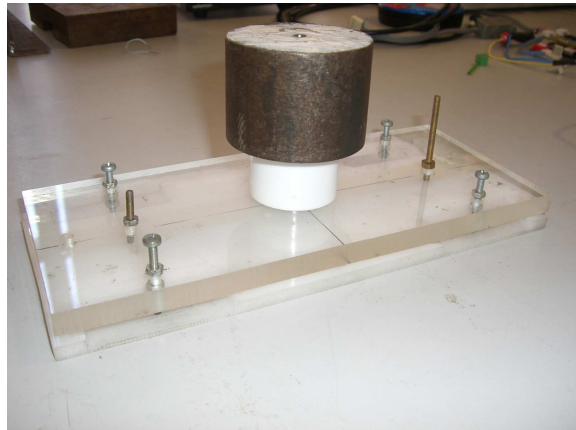
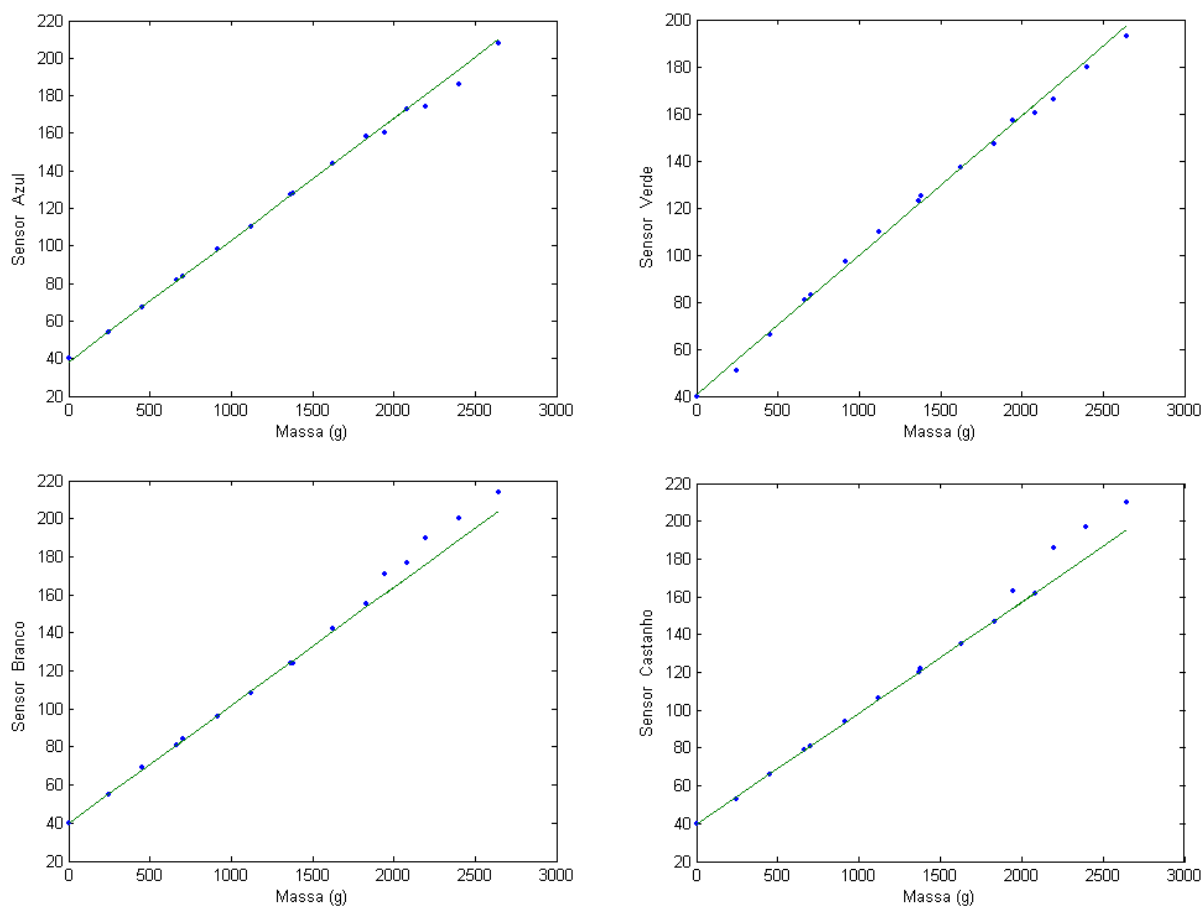


Fig. 4.14: Medição dos sensores em conjunto.

Tabela 16: Resultados experimentais para cada sensor nas duas experiências.

Cargas	Peso (N)	Teste de cada sensor isolado (experiência 1)				Teste conjunto: pesos colocados no centro do pé (experiência 2)			
		Azul	Verde	Branco	Castanho	Azul	Verde	Branco	Castanho
0	0.00	40	40	40	40	40	40	41	40
1	2.44	54	51	55	53				
2	4.45	67	66	69	66	48	40	42	48
1+2	6.90	84	83	84	81				
4	6.53	82	81	81	79	55	46	46	56
4+1	8.98	98	97	96	94				
4+2	11.0	110	110	108	106	64	55	54	64
4+1+2	13.4	127	123	124	120				
4+3	13.5	128	125	124	122	69	60	59	69
4+3+1	16.0	144	137	142	135				
4+3+2	18.0	158	147	155	147	77	69	67	75
4+3+2+1	20.4	173	160	177	162				
B	19.1	160	157	171	163	85	65	64	86
B+1	21.5	174	166	190	186				
B+2	23.5	186	180	200	197	95	71	71	95
B+2+1	26.0	208	193	214	210				
B+4	25.6					100	75	75	102
B+4+2	30.1					110	82	83	111
B+4+3	32.6					114	85	87	116
B+4+3+2	37.1					125	91	92	124
B+432+1	39.5					130	94	97	129





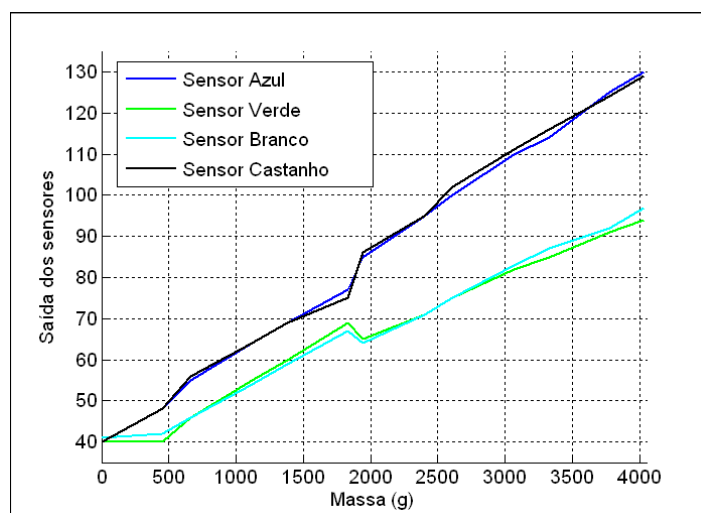
**Fig. 4.15: Relação entre a medição de cada sensor e o peso aplicado (experiência 1)**

A Fig. 4.15 apresenta graficamente os dados registados para cada sensor testado isoladamente (experiência 1), e confirma-nos a relação esperada entre a força gravítica aplicada sobre cada extensómetro e a deformação medida: trata-se sem dúvida de uma relação proporcional. As rectas traçadas correspondem à regressão linear “robusta” baseada nos pontos de massa inferior a 2000g, e pode-se observar que para cargas inferiores a este valor, a recta estimada coincide praticamente nos pontos medidos com coeficientes de correlação muito próximos de 1. Veja-se a Tabela 17 em que até os declives e as ordenadas na origem são bastante próximos entre as várias respostas.

<i>Parâmetro da regressão</i> $y = m \times x + b$	<i>Sensor Azul</i>	<i>Sensor Verde</i>	<i>Sensor Branco</i>	<i>Sensor Castanho</i>
Declive m	38.378	40.607	39.720	39.734
Ordenada na origem b	0.0648	0.0592	0.0621	0.0588
Coefficiente de correlação $R^2$	0.9999	0.9999	0.9999	0.9999

**Tabela 17: Parâmetros das regressões lineares traçadas.**

Para cargas superiores a este valor verifica-se um pequeno desvio principalmente no caso dos sensores branco e castanho. Este comportamento relaciona-se pelo facto de até à carga de 2000g as massas foram adicionadas de forma acumulativa, e após este valor, todas elas foram removidas para a colocação da massa B, o que provocou um retorno dos extensómetros à posição original seguida de nova deformação. O facto de não voltarmos a obter o mesmo comportamento linear sugere que ou a massa B não estava devidamente centrada sobre o sensor (difícil dado o formato da bateria), ou os sensores estavam no limite da zona elástica de funcionamento, ou então o processo regressão+deformação alterou de alguma forma o comportamento inicial.



**Fig. 4.16: Valores dos sensores com massas colocadas no centro da base (experiência 2)**

Analisando os dados obtidos colocando vários pesos no centro do pé e medindo os quatro sensores (experiência 2) podemos observar o mesmo efeito evidenciado atrás para cargas superior a 2 kg (Fig. 4.16). Outro resultado interessante é a assimetria das respostas quando os sensores funcionam em conjunto: os sensores “azul” e “castanho” apresentam valores superiores em relação aos restantes, e, para cargas superiores a 2 kg, a diferença ainda é maior. Como já se comprovou que os sensores estão a funcionar correctamente pela experiência 1, tal nos leva a concluir que a plataforma superior do pé não é completamente homogénea com distribuições de força diferentes ao longo da estrutura.

### 4.3.3 AMOSTRAGEM DOS SENSORES

O procedimento de leitura utilizado na secção anterior não efectuava qualquer pós-processamento via *firmware*, lendo apenas o resultado da ADC e passando-o directamente à unidade *Master* pelo barramento CAN. No entanto tal não é favorável, pois verificou-se experimentalmente que a saída dos extensómetros varia ao longo do tempo transmitindo-se essa variação para o circuito de acondicionamento de sinal. Ora, embora se possa ajustar o potenciómetro de calibração presente na ponte de *Wheatstone* é muito difícil garantir que após algum tempo, as saídas permaneçam constantes e iguais entre os quatro sensores. Tal pode ser motivado por variações de pré-carga ou simplesmente pelo ruído presente no circuito de acondicionamento.

Para ultrapassar este problema incluiu-se um segundo procedimento de calibração em *firmware* que considera os valores iniciais como correspondendo à posição de equilíbrio<sup>3</sup> da perna/corpo. A partir daqui todas as medições consequentes são feitas a partir dos valores originais pela adição de um *offset*.

A calibração segue o seguinte procedimento para cada um dos 4 sensores:

1. Amostragem e armazenamento de 25 valores para cada sensor de força;
2. Cálculo da média aritmética dos 25 valores para cada sensor;
3. Armazenamento do valor médio de cada sensor (valor original) para correcção de futuras medições.

No passo 1, a leitura dos sensores de força é realizada na forma pura, ou seja, sem a adição de qualquer *offset* (resultado efectivo da ADC), e só depois do procedimento de calibração, é introduzido o factor de correcção determinado pelo passo 3. Tal correcção é baseada na seguinte expressão:

$$Resultado = Sensor_{output} - Sensor_{original}$$

<sup>3</sup> Considera-se como equilíbrio a situação de perna/corpo vertical: o centro de pressão assenta sobre o centro da base do pé.

De modo a atribuir ao Resultado um valor de 8 bits, reduziu-me a resolução aos parâmetros  $Sensor_{output}$  e  $Sensor_{original}$  (8 para 7 bits). Desta forma, o resultado encontrar-se-á na gama -128 a +127. Contudo, a representação de valores negativos não faz muito sentido, uma vez que nunca temos forças negativas aplicadas sobre cada sensor, mas sim forças abaixo do valor inicial. Forças negativas implicariam que as placas de acrílico, onde estão fixos os extensómetros, se deformassem no sentido inverso, o que nunca acontece, uma vez que os parafusos de apoio só estão aplicados numa face das placas. Contudo, há sempre uma força de pré-carga não nula com o sistema em equilíbrio, e por isso, podemos ter forças acima ou abaixo dessa força de pré-carga.

Interessa-nos por isso obter sempre resultados positivos, mas para tal precisaríamos de conhecer *à priori* a força aplicada inicialmente, o que não sabemos, nem temos meios de saber. Para resolver este pormenor, em vez de atribuímos zero ao resultado (da expressão) inicial iremos atribuir o valor 127, como correspondendo ao ponto intermédio entre 0 e 255 relativo à utilização de valores de 8 bits. Uma condição deste género pressupõe que a força inicial situa-se a meia escala do valor máximo, o que nem sempre é verdade, mas pelo menos permite-nos dar a garantia de resultados sempre positivos. De qualquer maneira, a unidade principal de controlo (PC) sempre pode corrigir este resultado pela adição de um segundo *offset*.

$$Resultado = (Sensor_{output} - Sensor_{original}) + 127$$

Resumindo, o processo de leitura da força aplicado segue o seguinte processo:

1. Com a aplicação de uma força, a placa de acrílico em conjunto com o extensómetro colado, deformam-se;
2. A resistência óhmica do extensómetro varia em torno do valor nominal (120Ω);
3. A ponte de Wheatstone, previamente calibrada (pelo potenciómetro), regista a variação pelo desequilíbrio de corrente que circula entre os dois ramos, resultando numa variação da tensão à saída;
4. O amplificador instrumental amplifica esta tensão;
5. A ADC do microcontrolador digitaliza esta tensão num valor de 8 bits (0-255);
6. O valor estimado da força aplicada, é determinado com base numa calibração inicial (por *software*) presumindo a postura vertical da perna nessa circunstância.

Como se pode constatar, a força estimada não corresponde ao seu valor absoluto, dado que ele depende de inúmeros factores:

- Desgaste na deformação do acrílico (resultante das quebras nas ligações poliméricas);
- Precisão dos extensómetros e resistências envolvidas na ponte de Wheatstone: pode variar para os diversos sensores;
- Precisão da ADC;
- **Calibração física sobre o potenciómetro;**
- **Calibração por *software*.**

Embora os motivos resultantes de desgaste e precisão possam ser resolvidos pela adopção de materiais e componentes de boa qualidade, o problema agrava-se no que respeita aos procedimentos de calibração. Como se sabe, no momento da inicialização do sistema, os sensores de força dos pés já possuem uma determinada força aplicada sobre eles, resultante da massa da estrutura superior do pé, da perna e do restante corpo, se presente. Estas condições iniciais são desconhecidas, e ao fixar este valor em 127 está-se a pressupor que um acréscimo de 10 unidades na força estimada corresponde a um acréscimo de 7.8% na força absoluta, o que pode não ser verdade. Pelo contrário, se definirmos o valor referência como 50, o mesmo acréscimo de 10 unidades já corresponde a um acréscimo de 20% na força absoluta. Tudo depende da sensibilidade das placas de acrílico e do peso aplicado inicialmente. Embora a sensibilidade seja praticamente constante entre cada placa individual, no que respeita à operação em conjunto com uma carga inicial tal já não é verdade: repare-se que com a postura vertical da perna, mais peso é aplicado no par de sensores traseiros dada a presença de um servomotor nessa posição.

O mesmo também se verifica relativamente a um dos pares de sensores laterais, devido à assimetria do eixo de rotação lateral do pé em relação ao centro do mesmo. Ou seja, existem dois problemas: (i) o valor inicial da força absoluta é desconhecido; e (ii) a força absoluta inicial difere de sensor para sensor.

A resolução do problema da força inicial, por enquanto é baseada em métodos empíricos, pela seu produto com uma constante pré-definida. Relativamente à variação de sensor para sensor, ainda não se tornou suficientemente importante quando usando apenas a perna, pois o seu reduzido peso torna estas variações desprezáveis. Contudo, experiências subsequentes com cargas aplicadas sobre a perna, tornaram este problema evidente, requerendo a aplicação de ganhos de amplificação independentes para cada sensor.

Um aspecto importante que se deve ter em conta, é que nunca se conhece a força absoluta aplicada sobre cada sensor de força. Aquilo que se conhece é o que se pode medir: uma variação de força aplicada em cada sensor. No que respeita aos parâmetros absolutos, como a força aplicada, eles dependem das seguintes condições iniciais:

- Deformação inicial das placas de acrílico que sustentam os extensómetros;
- Força de pré-carga introduzida pelo peso da plataforma humanóide na situação de equilíbrio;
- Amplificação introduzida pelo circuito de acondicionamento de sinal;
- *Offset* introduzido na calibração da ponte de Wheatstone;
- *Offset* introduzido pela calibração via *software*.

Nesta linha de ideais, a única coisa que temos conhecimento é que a variação do sinal medido está relacionada proporcionalmente com a variação da força aplicada e que o sinal medido está relacionado com a força através de um *offset*.

Embora se possa conhecer o *offset* pela análise dos dados experimentais (Tabela 17) não nos podemos esquecer que esse valor apenas se aplica a condições iniciais específicas. O que se sabe, é que as condições de pré-carga da plataforma humanóide, bem como a deformação inicial das placas de acrílico não são sempre constantes o que dificulta obviamente esta tarefa. Para piorar mais, as saídas iniciais dos extensómetros estão sempre em mudança pelo que os *offsets* de calibração nunca são estáticos.

Substituindo o acrílico utilizado nas placas que sustentam os extensómetros por outros materiais pode melhorar o desempenho dos sensores, mas as variações inerciais inerentes à locomoção continuam presentes, pelo que não existe uma solução simples para a resolução deste problema. Contudo, garante-se que a relação de variação entre a deformação e a força é estática pelo que se pode conhecer a variação de força aplicada. Pela Tabela 17 sabe-se que a constante que relaciona a deformação com a força aplicada é de cerca de 4:

$$\frac{\Delta \text{deformação}}{\Delta \text{massa}} = 39.6 \quad \Rightarrow \quad \frac{\Delta \text{deformação}}{\Delta \text{força}} = 4.04, \quad \text{com } g = 9.81 \text{ m/s}^2$$

Ou seja, sabe-se que uma variação na deformação de uma unidade corresponde a uma variação de força na ordem dos 0.25N. Apenas o valor absoluto é que não se pode conhecer!

## 4.4 MEDIÇÃO DO CENTRO DE PRESSÃO

### 4.4.1 REPRESENTAÇÃO DO CENTRO DE PRESSÃO

O centro de pressão é dado pela projecção da força gravitacional da plataforma humanóide combinada com as forças inerciais inerentes ao próprio movimento sobre cada pé. Para baixas velocidades as forças inerciais podem ser desprezadas só entrando em linha de conta a força gravitacional da estrutura que cada pé suporta. Como esta força está relacionada com o centro de massa, podemos dizer que, para baixas velocidades, o centro de pressão corresponde à projecção do centro de massa da estrutura sobre cada pé. No caso de uma só perna, diz-se que o centro de pressão corresponde ao ponto geométrico localizado na base do pé que corresponde à projecção do centro de massa da perna.

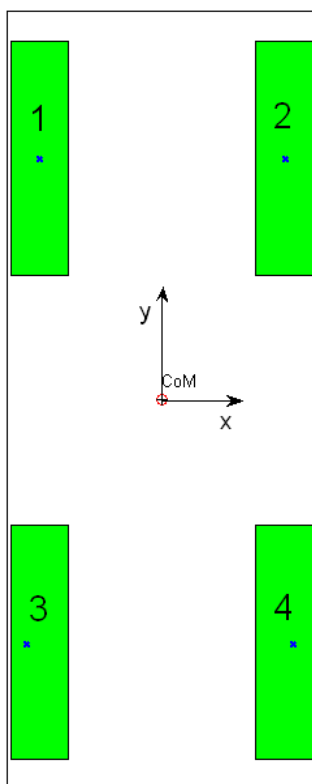


Fig. 4.17: Projecção do centro de massa sobre o pé na situação de equilíbrio.

$$C\bar{o}P = (CoP_x, CoP_y)$$

A Fig. 4.17 representa a projecção do centro de massa sobre um pé no caso de equilíbrio de apenas uma perna. Quando a perna se encontra na posição vertical, o seu centro de massa é projectado sobre o centro geométrico da base do pé, ou seja, nas coordenadas  $(x,y)=(0,0)$ .

### 4.4.2 CÁLCULO DO CENTRO DE PRESSÃO

Como já foi evidenciado, o centro de pressão é dado em coordenadas cartesianas bidimensionais  $(x,y)$  segundo o plano do pé. Contudo, considerando a hipótese da utilização de baixas velocidades, não podemos determinar este parâmetro de uma forma analítica com base no cálculo do centro de massa, uma vez que este método implica a utilização de matemática bastante pesada para um microcontrolador. Como alternativa podemos medir experimentalmente esta grandeza utilizando as forças de reacção obtidas a partir dos extensómetros.

Tal pode ser feito pela análise da distribuição das forças ao longo dos quatro sensores:

- Se os quatro sensores indicarem a mesma força, o centro de pressão localiza-se sobre o centro do pé;
- Se um sensor apresenta uma força não nula, com as restantes nulas, o centro de pressão desloca-se completamente para a localização do sensor de valor não nulo.

O cálculo do centro de pressão baseia-se por isso, numa média ponderada das quatro forças usando a localização dos sensores. Seja a localização dos sensores dada pelos vectores  $\vec{d}_1$ ,  $\vec{d}_2$ ,  $\vec{d}_3$  e  $\vec{d}_4$  segundo os eixos cartesianos indicados na Fig. 4.17:

$$\begin{aligned}\vec{d}_1 &= (x_1, y_1) & \vec{d}_2 &= (x_2, y_2) \\ \vec{d}_3 &= (x_3, y_3) & \vec{d}_4 &= (x_4, y_4)\end{aligned}$$

O centro de pressão é dado pelas expressões:

$$\boxed{C\vec{o}P = \frac{\sum_{i=1}^4 F_i \cdot \vec{d}_i}{\sum_{i=1}^4 F_i} \quad \Rightarrow \quad CoP_x = \frac{\sum_{i=1}^4 F_i \cdot x_i}{\sum_{i=1}^4 F_i} \quad CoP_y = \frac{\sum_{i=1}^4 F_i \cdot y_i}{\sum_{i=1}^4 F_i}}$$

Estas são as expressões usadas para a determinação da localização do centro de pressão nos dois eixos cartesianos. De notar que o parâmetro  $F_i$  indicado corresponde à força absoluta para o sensor  $i$ . No entanto, tal como já foi indicado, esse valor não é mensurável: apenas conseguimos medir um valor que está relacionado com a força segundo uma constante e depende de um valor inicial definido durante a calibração. Para resolver este problema é necessário estimar a força absoluta inicial, o que não é possível fazê-lo com precisão. A utilização do valor 127 como resultado inicial na situação de equilíbrio já prevê esta força inicial, e deve ser alterado caso a variação da força medida não corresponda à real.

$$C\vec{o}P = \frac{\sum_{i=1}^4 (F_i^{med} + F_i^0) \cdot \vec{d}_i}{\sum_{i=1}^4 (F_i^{med} + F_i^0)} = \frac{\sum_{i=1}^4 (F_i^{med} + 127) \cdot \vec{d}_i}{\sum_{i=1}^4 (F_i^{med} + 127)}, \quad \text{com } F_i^{med}=0 \text{ em equilíbrio!}$$

A força medida apresentada na expressão anterior corresponde à diferença entre o resultado da ADC em cada momento e o seu valor original no momento da calibração (em equilíbrio) (ver secção 4.3.3), o que resulta no valor nulo nesta posição. Considerando apenas o resultado estimado da força calculado na secção 4.3.3 ( $F_i^{calc}$ ), ou seja, entrando em linha de conta com um *offset*, e usando o ponto 127 como valor neutro, a equação anterior pode ser aproximada da seguinte forma (aproximada se o *offset* considerado for semelhante à força inicial):

$$CoP_x = \frac{\sum_{i=0}^4 (F_i^{calc} * x_i)}{\sum_{i=0}^4 F_i^{calc}} \quad CoP_y = \frac{\sum_{i=0}^4 (F_i^{calc} * y_i)}{\sum_{i=0}^4 F_i^{calc}}$$

Convém referir também que o formato não simétrico dos pés (rectangular), faz com que a medição do CoP na componente  $xx$  tenha uma maior sensibilidade que na componente ortogonal  $yy$ . Para compensar esta assimetria, adicionou-se um segundo *offset* (subtractivo) à força medida, relativo a cada componente ortogonal. Este segundo *offset* está expresso nas equações seguintes, possuindo um valor superior para a componente  $yy$ :

$$CoP_x = \frac{\sum_{i=0}^4 [(F_i^{calc} - offset_x) * x_i]}{\sum_{i=0}^4 (F_i^{calc} - offset_x)} \quad CoP_y = \frac{\sum_{i=0}^4 [(F_i^{calc} - offset_y) * y_i]}{\sum_{i=0}^4 (F_i^{calc} - offset_y)}$$

Por observação experimental, convencionou-se  $offset_x=0$  e  $offset_y=50$  – dado que a força normalizada é 127, os *offsets* devem ser inferiores a 128 de modo a evitar resultados negativos.

## 4.5 CONTROLO DE EQUILÍBRIO

Nesta secção procurar-se-á construir um controlador de equilíbrio que, baseando-se na medição do centro de pressão aplicado nos pés, procure o equilíbrio da plataforma humanóide. Este controlador poderá ser utilizado na perna de suporte na realização da marcha: enquanto uma perna é responsável pelo movimento correspondente a um passo, a outra terá de garantir o equilíbrio da plataforma de modo a evitar qualquer tipo de instabilidade.

No caso de uso de uma só perna, o estado de equilíbrio corresponderá à posição vertical, dado que é a única configuração em que o centro de massa coincide com o ponto central do pé. Nos testes realizados no corrente trabalho apenas se teve a preocupação do desenvolvimento dos algoritmos de controlo tendo apenas utilizado uma perna para os testes. Testes a realizar no futuro deverão incluir os restantes componentes da estrutura humanóide para análise de comportamentos tanto na presença de irregularidades no solo, como de perturbações externas aplicadas ao longo do corpo.

Uma visão global da perna assente sobre o pé sensível a forças, com a respectiva electrónica de condicionamento, pode ser visualizada na Fig. 4.18.

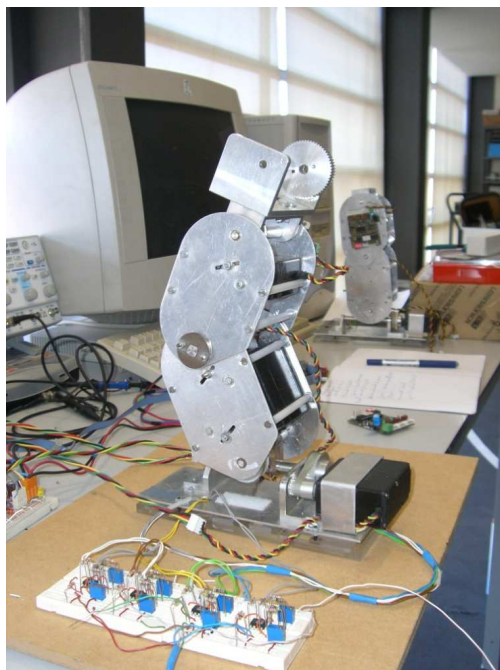


Fig. 4.18: Perna completa sobre um pé sensível a forças.

### 4.5.1 ESTRUTURA DO CONTROLADOR

Uma vez que já possuímos um meio de calcular o centro de pressão em cada instante, apenas nos falta apresentar o controlador de equilíbrio que tem como função o movimento das juntas com o objectivo de atingir um determinado Centro de Pressão (CoP) referência, que tanto pode ser o centro do pé, se apenas pretendemos garantir a postura vertical, ou então outros valores de modo a conferir alguma dinâmica ao sistema.

A Fig. 4.19 apresenta um diagrama com a descrição dos controladores envolvidos. Após a leitura dos sensores de força, o cálculo do Centro de Pressão é executado, que por comparação com o Centro de Pressão desejado ( $CoP_{ref}$ ), a variação de posição  $v(n)$  a aplicar a cada junta do pé e da perna é determinado. Esta variação  $v(n)$  é adicionada à posição actual de cada junta  $p(n)$  obtendo assim o valor absoluto da posição  $u(n)$ . Finalmente o controlador local PID é usado para assegurar a aplicação da posição solicitada.

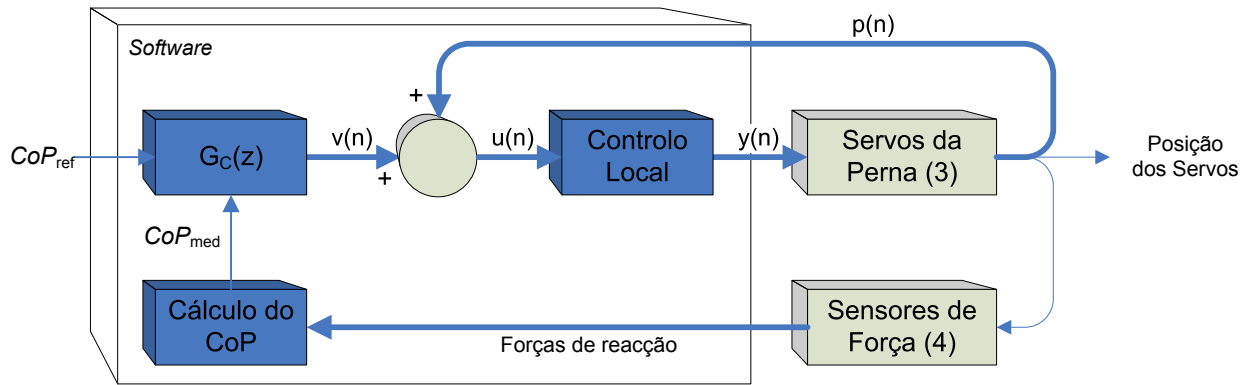


Fig. 4.19: Diagrama com os controladores aplicados para a compensação de equilíbrio.

As secções seguintes centram-se na implementação do bloco  $G_c(z)$ , ou seja, na implementação de um controlador, que, com base nos sensores de força, atinja o estado de equilíbrio.

## 4.5.2 CONTROLO PROPORCIONAL

O tipo de controlo mais simples de implementar, é o controlo proporcional, cuja velocidade a aplicar depende proporcionalmente do erro do centro de pressão:

$$\vec{e}_{CoP} = CoP^{[ref]} - CoP^{[med]} \Leftrightarrow \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \end{bmatrix} = \begin{bmatrix} CoP_x^{[ref]} \\ CoP_y^{[ref]} \end{bmatrix} - \begin{bmatrix} CoP_x^{[med]} \\ CoP_y^{[med]} \end{bmatrix}$$

$$\Delta \vec{\theta} = \vec{K}_P \cdot \vec{e}_{CoP} \Leftrightarrow \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \end{bmatrix} = \begin{bmatrix} K_{Px} \\ K_{Py} \end{bmatrix} \cdot \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \end{bmatrix}$$

A lei de controlo é indicada na última equação verificando-se a relação proporcional entre o erro do centro de pressão  $e_{CoP}$  e a velocidade aplicada aos motores  $\Delta \theta$  pela constante  $K_P$  (o produto das últimas duas grandezas é escalar) (Tabela 18). No entanto note-se que a perna a equilibrar possui três motores: o motor lateral do pé, o motor dianteiro do pé e o motor do joelho.

O parâmetro de velocidade  $\Delta \theta$  apenas controla dois motores. Como estes dois motores devem estar associados à deslocação  $(x,y)$  da perna sobre o solo de modo a corrigir o centro de pressão  $(CoP_x, CoP_y)$  medido, estes motores deverão corresponder às duas juntas ortogonais do pé (1 e 2) (ver Fig. 4.20). A junta do joelho não é, por isso, utilizada para a compensação.

Junta lateral do pé:	$\Delta \theta_1 = K_{Px} \cdot e_{CoPx}$
Junta dianteira do pé:	$\Delta \theta_2 = K_{Py} \cdot e_{CoPy}$

Tabela 18: Equações para o controlo proporcional.

## 4.5.3 CONTROLO ATRAVÉS DA MATRIZ JACOBIANA

Contudo gostaríamos de utilizar todas as juntas para efectuar compensação, incluindo a do joelho quando as duas ortogonais do pé não fossem suficientes. Um exemplo claro da utilidade desta junta é dado quando a junta dianteira do pé ( $\theta_2$ ) satura num dos seus extremos, utilizando-se a partir daí a junta do joelho para continuar a compensação.



Outro pormenor a salientar, relaciona-se com a falta de rigor apresentada na lei de controlo proporcional apresentada: estamos a relacionar o erro do centro de pressão, que é uma grandeza linear de distância, com um parâmetro angular – velocidade em graus por segundo – só através de uma constante! Sempre que se pretende relacionar uma grandeza linear com outra angular, uma matriz, denominada por Jacobiano, é utilizada. Por isso, a lei de controlo será modificada de modo a incluir este novo parâmetro:

$$\vec{v} = (\vec{K} \cdot \vec{J})^{-1} \times \vec{e}$$

A matriz Jacobiana representa a relação entre a derivada de um parâmetro linear com a derivada de um parâmetro angular. Para o nosso caso pretendemos relacionar centro de pressão (parâmetro linear) com a posição (parâmetro angular) pelo que:

$$\vec{J}_{CoP} = \frac{\partial \vec{CoP}}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial CoP_x}{\partial \theta_1} & \frac{\partial CoP_x}{\partial \theta_2} & \frac{\partial CoP_x}{\partial \theta_3} \\ \frac{\partial CoP_y}{\partial \theta_1} & \frac{\partial CoP_y}{\partial \theta_2} & \frac{\partial CoP_y}{\partial \theta_3} \\ \frac{\partial CoP_z}{\partial \theta_1} & \frac{\partial CoP_z}{\partial \theta_2} & \frac{\partial CoP_z}{\partial \theta_3} \end{bmatrix}$$

cujo centro de pressão **CoP** é um valor ternário representado nas coordenadas espaciais (x,y,z), e  $\theta$  corresponde às posições  $\theta_1$ ,  $\theta_2$ , e  $\theta_3$  das juntas lateral e dianteira do pé, e do joelho respectivamente (Fig. 4.20). Note que nesta matriz já está incluída a posição do joelho o que prevê a controlabilidade desta junta. Embora a coordenada z de **CoP** não tenha significado, optou-se por incluí-lo, pois confere o formato quadrado à matriz Jacobiana, o que facilita o cálculo da sua inversa.

Considerando que para baixas velocidades o centro de massa coincide com o centro de pressão, podemos estimar a matriz Jacobiana do centro de pressão a partir do Centro de Massa da estrutura de uma perna:

$$\vec{J}_{CoP} = \frac{\partial \vec{CoP}}{\partial \vec{\theta}} \approx \frac{\partial \vec{CoM}}{\partial \vec{\theta}} = \vec{J}_{CoM} = \begin{bmatrix} \frac{\partial CoM_x}{\partial \theta_1} & \frac{\partial CoM_x}{\partial \theta_2} & \frac{\partial CoM_x}{\partial \theta_3} \\ \frac{\partial CoM_y}{\partial \theta_1} & \frac{\partial CoM_y}{\partial \theta_2} & \frac{\partial CoM_y}{\partial \theta_3} \\ \frac{\partial CoM_z}{\partial \theta_1} & \frac{\partial CoM_z}{\partial \theta_2} & \frac{\partial CoM_z}{\partial \theta_3} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}$$

Desenvolvendo a relação entre a variação angular das juntas e o erro do Centro de Pressão:

$$\begin{aligned} \partial \vec{\theta} &= \vec{J}_{CoP}^{-1} \times \partial \vec{CoP} \approx \vec{J}_{CoM}^{-1} \times \partial \vec{CoP} \\ \Delta \vec{\theta} &\approx \vec{J}_{CoM}^{-1} \times \Delta \vec{CoP} \end{aligned}$$

Com base nesta equação, facilmente se pode deduzir a lei de controlo a partir da matriz Jacobiano do Centro de Massa:

$$\Delta \vec{\theta} \approx (\vec{k} \cdot \vec{J}_{CoM})^{-1} \times \vec{e}_{CoP}, \quad \vec{e}_{CoP} = \vec{CoP}^{[ref]} - \vec{CoP}^{[med]}$$

A matriz  $k$  será necessária para regular os ganhos de compensação para cada elemento da matriz Jacobiano. Por questões de simplificação, em vez da inversa do Jacobiano, podemos usar o Jacobiano transposto, dada a necessidade de simplificação dos cálculos matemáticos ao nível do microcontrolador:

$$\Delta \vec{\theta} \approx (\vec{k} \cdot \vec{J}_{CoM})^T \times \vec{e}_{CoP} = (\vec{k}^T \cdot \vec{J}_{CoM}^T) \times \vec{e}_{CoP}$$

Expandindo os parâmetros vectoriais, obtém-se:

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = \left( \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} J_{11} & J_{21} & J_{31} \\ J_{12} & J_{22} & J_{32} \\ J_{13} & J_{23} & J_{33} \end{bmatrix} \right) \times \begin{bmatrix} CoP_x^{[ref]} - CoP_x^{[med]} \\ CoP_y^{[ref]} - CoP_y^{[med]} \\ CoP_z^{[ref]} - CoP_z^{[med]} \end{bmatrix}$$

As variáveis  $\theta_i$  estão indicadas no modelo da Fig. 4.20 descritiva da perna e do pé.

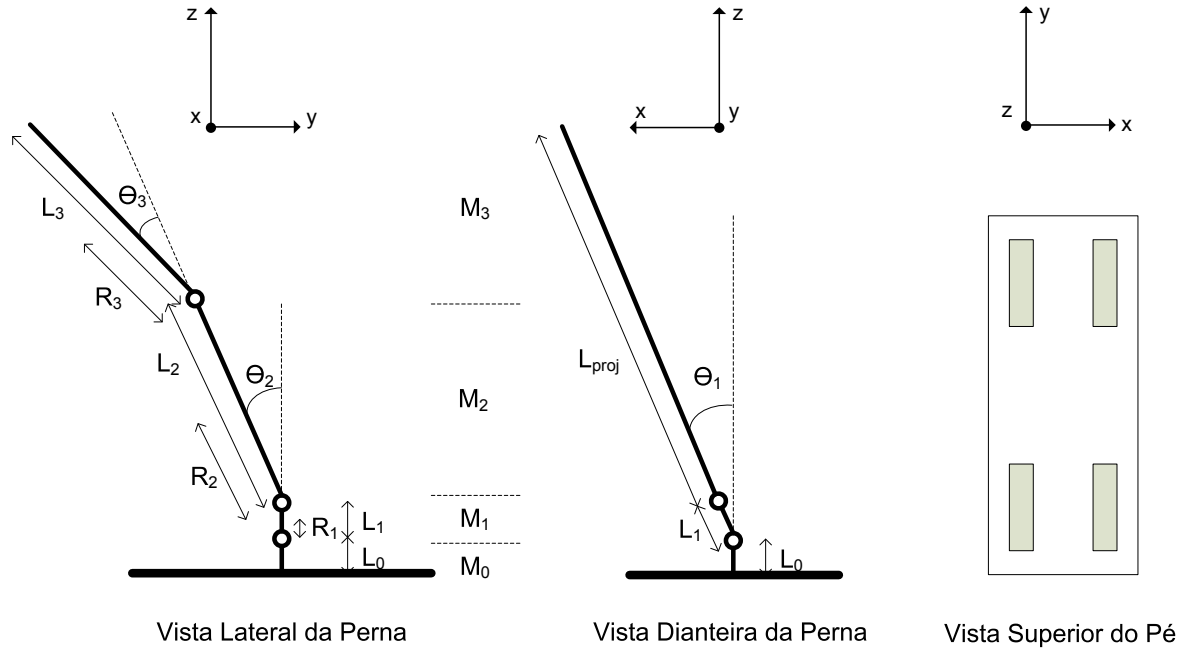


Fig. 4.20: Modelo da perna humanóide com a associação das diversas variáveis de interesse.

### Lei de Controle do Centro de Pressão

Para determinar a lei que regula o centro de pressão, precisamos de obter primeiramente a matriz Jacobiana do Centro de Pressão ( $J_{CoP}$ ). Para tal teremos de conhecer a relação entre o centro de pressão e cada uma das juntas  $\theta$ . Dado que não conhecemos nenhuma expressão analítica que nos forneça tal informação, faremos a suposição do centro de pressão coincidir com o centro de massa, o que até verdade se nos restringirmos a baixas velocidades.

$$\vec{J}_{CoP} \approx \frac{\partial \vec{CoM}}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial CoM_x}{\partial \theta_1} & \frac{\partial CoM_x}{\partial \theta_2} & \frac{\partial CoM_x}{\partial \theta_3} \\ \frac{\partial CoM_y}{\partial \theta_1} & \frac{\partial CoM_y}{\partial \theta_2} & \frac{\partial CoM_y}{\partial \theta_3} \\ \frac{\partial CoM_z}{\partial \theta_1} & \frac{\partial CoM_z}{\partial \theta_2} & \frac{\partial CoM_z}{\partial \theta_3} \end{bmatrix}$$

Já o centro de massa, podemos calculá-lo através da expressão analítica:

$$\vec{CoM} = \frac{\sum M_i \cdot \vec{R}_i}{\sum M_i}$$

em que  $M_i$  representa a massa de cada elo, e  $\vec{R}_i$  a distância do início de cada elo ao seu centro de massa.

A Fig. 4.20 representa uma perna sob duas perspectivas diferentes – de lado e de frente. A partir deste diagrama, as expressões dos centros de massa para as diversas componentes espaciais podem ser deduzidas.

Seja:

$$\begin{cases} s_1 = \sin \theta_1 \\ s_2 = \sin \theta_2 \\ s_{23} = \sin(\theta_2 + \theta_3) \end{cases} \quad \begin{cases} c_1 = \cos \theta_1 \\ c_2 = \cos \theta_2 \\ c_{23} = \cos(\theta_2 + \theta_3) \end{cases}$$

$$\bar{CoM} = \begin{cases} CoM_x = \frac{M_1 R_1 s_1 + M_2 (L_1 + R_2 c_2) s_1 + M_3 (L_1 + L_2 c_2 + R_3 c_{23}) s_1}{M_1 + M_2 + M_3} \\ CoM_y = \frac{-M_2 R_2 s_2 - M_3 (L_2 s_2 + R_3 s_{23})}{M_2 + M_3} \\ CoM_z = \frac{M_0 R_0 + M_1 (L_0 + R_1 c_1) + M_2 [L_0 + (L_1 + R_2 c_2) c_1] + M_3 [L_0 + (L_1 + L_2 c_2 + R_3 c_{23}) c_1]}{M_0 + M_1 + M_2 + M_3} \end{cases}$$

Calculando as suas derivadas parciais em ordem a  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  para obtenção dos elementos da matriz, obtém-se:

$$\frac{\partial \bar{CoM}}{\partial \theta} : \begin{cases} J_{11} = \frac{\partial CoM_x}{\partial \theta_1} = \frac{M_1 R_1 c_1 + M_2 (L_1 + R_2 c_2) c_1 + M_3 (L_1 + L_2 c_2 + R_3 c_{23}) c_1}{M_1 + M_2 + M_3} \\ J_{12} = \frac{\partial CoM_x}{\partial \theta_2} = \frac{-M_2 R_2 s_2 s_1 - M_3 (L_2 s_2 + R_3 s_{23}) s_1}{M_1 + M_2 + M_3} \\ J_{13} = \frac{\partial CoM_x}{\partial \theta_3} = \frac{-M_3 R_3 s_{23} s_1}{M_1 + M_2 + M_3} \\ J_{21} = \frac{\partial CoM_y}{\partial \theta_1} = 0 \\ J_{22} = \frac{\partial CoM_y}{\partial \theta_2} = \frac{-M_2 R_2 c_2 - M_3 (L_2 c_2 + R_3 c_{23})}{M_2 + M_3} \\ J_{23} = \frac{\partial CoM_y}{\partial \theta_3} = \frac{M_3 R_3 c_{23}}{M_2 + M_3} \\ J_{31} = \frac{\partial CoM_z}{\partial \theta_1} = \frac{-M_1 R_1 s_1 - M_2 (L_1 + R_2 c_2) s_1 - M_3 (L_1 + L_2 c_2 + R_3 c_{23}) s_1}{M_0 + M_1 + M_2 + M_3} \\ J_{32} = \frac{\partial CoM_z}{\partial \theta_2} = \frac{-M_2 R_2 s_2 c_1 - M_3 [(L_2 s_2 + R_3 s_{23}) c_1]}{M_0 + M_1 + M_2 + M_3} \\ J_{33} = \frac{\partial CoM_z}{\partial \theta_3} = \frac{-M_3 R_3 s_{23} c_1}{M_0 + M_1 + M_2 + M_3} \end{cases}$$

De modo a simplificar estas expressões consideremos que  $M_0 \approx M_1 \approx 0$  e que  $L_1 \approx 0$ :

$$\frac{\partial \bar{C}oM}{\partial \bar{\theta}} : \left\{ \begin{array}{l} J_{11} = \frac{\partial CoM_x}{\partial \theta_1} = \frac{M_2 R_2 c_2 + M_3 (L_2 c_2 + R_3 c_{23})}{M_2 + M_3} \cdot c_1 \\ J_{12} = \frac{\partial CoM_x}{\partial \theta_2} = -\frac{M_2 R_2 s_2 + M_3 (L_2 s_2 + R_3 s_{23})}{M_2 + M_3} \cdot s_1 \\ J_{13} = \frac{\partial CoM_x}{\partial \theta_3} = -\frac{M_3 R_3 s_{23}}{M_2 + M_3} \cdot s_1 \\ \\ J_{21} = \frac{\partial CoM_y}{\partial \theta_1} = 0 \\ J_{22} = \frac{\partial CoM_y}{\partial \theta_2} = -\frac{M_2 R_2 c_2 + M_3 (L_2 c_2 + R_3 c_{23})}{M_2 + M_3} \\ J_{23} = \frac{\partial CoM_y}{\partial \theta_3} = \frac{M_3 R_3 c_{23}}{M_2 + M_3} \\ \\ J_{31} = \frac{\partial CoM_z}{\partial \theta_1} = -\frac{M_2 R_2 c_2 + M_3 (L_2 c_2 + R_3 c_{23})}{M_2 + M_3} \cdot s_1 \\ J_{32} = \frac{\partial CoM_z}{\partial \theta_2} = -\frac{M_2 R_2 s_2 + M_3 (L_2 s_2 + R_3 s_{23})}{M_2 + M_3} \cdot c_1 \\ J_{33} = \frac{\partial CoM_z}{\partial \theta_3} = -\frac{M_3 R_3 s_{23}}{M_2 + M_3} \cdot c_1 \end{array} \right.$$

Fazendo:

$$\begin{array}{ll} S_{23} = \frac{M_3 \cdot R_3 \cdot s_{23}}{M_2 + M_3} & SS_{23} = \frac{M_2 \cdot R_2 \cdot s_2 + M_3 (L_2 \cdot s_2 + R_3 \cdot s_{23})}{M_2 + M_3} \\ C_{23} = \frac{M_3 \cdot R_3 \cdot c_{23}}{M_2 + M_3} & CC_{23} = \frac{M_2 \cdot R_2 \cdot c_2 + M_3 (L_2 \cdot c_2 + R_3 \cdot c_{23})}{M_2 + M_3} \end{array}$$

Então as diversas derivadas parciais podem ser descritas de uma forma mais simples:

$J_{11} = CC_{23} \cdot c_1$	$J_{21} = 0$	$J_{31} = -CC_{23} \cdot s_1$
$J_{12} = -SS_{23} \cdot s_1$	$J_{22} = -CC_{23}$	$J_{23} = -SS_{23} \cdot c_1$
$J_{13} = -S_{23} \cdot s_1$	$J_{23} = C_{23}$	$J_{33} = -S_{23} \cdot c_1$

Desta forma, a matriz Jacobiana transposta, relativa ao centro de pressão, é dada por:

$\bar{J}_{CoP}^T = \left( \frac{\partial \bar{C}oM}{\partial \bar{\theta}} \right)^T = \begin{bmatrix} J_{11} & J_{21} & J_{31} \\ J_{12} & J_{22} & J_{32} \\ J_{13} & J_{23} & J_{33} \end{bmatrix} = \begin{bmatrix} CC_{23} \cdot c_1 & 0 & -CC_{23} \cdot s_1 \\ -SS_{23} \cdot s_1 & -CC_{23} & -SS_{23} \cdot c_1 \\ -S_{23} \cdot s_1 & C_{23} & -S_{23} \cdot c_1 \end{bmatrix}$
---

Concluindo, as variações angulares a aplicar são determinadas da seguinte forma:

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = \left( \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} J_{11} & J_{21} & J_{31} \\ J_{12} & J_{22} & J_{32} \\ J_{13} & J_{23} & J_{33} \end{bmatrix} \right) \times \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \\ e_{CoPz} \end{bmatrix}$$

$$\begin{cases} \Delta\theta_1 = k_{11} \cdot J_{11} \cdot e_{CoPx} + k_{21} \cdot J_{21} \cdot e_{CoPy} + k_{31} \cdot J_{31} \cdot e_{CoPz} \\ \Delta\theta_2 = k_{12} \cdot J_{12} \cdot e_{CoPx} + k_{22} \cdot J_{22} \cdot e_{CoPy} + k_{32} \cdot J_{32} \cdot e_{CoPz} \\ \Delta\theta_3 = k_{13} \cdot J_{13} \cdot e_{CoPx} + k_{23} \cdot J_{23} \cdot e_{CoPy} + k_{33} \cdot J_{33} \cdot e_{CoPz} \end{cases}$$

Dado que o erro do CoP na componente  $zz$  não tem significado, este parâmetro não será considerado, pelo que:

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = \left( \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} J_{11} & J_{21} & J_{31} \\ J_{12} & J_{22} & J_{32} \\ J_{13} & J_{23} & J_{33} \end{bmatrix} \right) \times \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \\ 0 \end{bmatrix}$$

$$\begin{cases} \Delta\theta_1 = k_{11} \cdot J_{11} \cdot e_{CoPx} + k_{21} \cdot J_{21} \cdot e_{CoPy} \\ \Delta\theta_2 = k_{12} \cdot J_{12} \cdot e_{CoPx} + k_{22} \cdot J_{22} \cdot e_{CoPy} \\ \Delta\theta_3 = k_{13} \cdot J_{13} \cdot e_{CoPx} + k_{23} \cdot J_{23} \cdot e_{CoPy} \end{cases}$$

Pela análise a estas expressões, a terceira coluna do Jacobiano transposto deixa de ser utilizada, pelo que a matriz é suficiente na forma:

$$\vec{J}_{CoP}^T = \begin{bmatrix} J_{11} & J_{21} & 0 \\ J_{12} & J_{22} & 0 \\ J_{13} & J_{23} & 0 \end{bmatrix} = \begin{bmatrix} CC_{23} \cdot c_1 & 0 & 0 \\ -SS_{23} \cdot s_1 & -CC_{23} & 0 \\ -S_{23} \cdot s_1 & C_{23} & 0 \end{bmatrix}$$

### Lei de Controlo da Altura da Anca

No entanto, além do centro de pressão, torna-se igualmente importante o controlo da altura da anca, para efeitos de estabilização da cintura ao longo dos processos de locomoção. Por isso, para as coordenadas  $x$  e  $y$  seria utilizado o controlador de CoP, enquanto para a coordenada  $z$  utilizaríamos um controlador de altura baseado na cinemática directa.

Utilizando igualmente uma matriz Jacobiana, mas relativa às coordenadas do topo da perna (anca):

$$\vec{J}_{topo} = \frac{\partial \vec{p\ddot{o}s}_{topo}}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial x_{topo}}{\partial \theta_1} & \frac{\partial x_{topo}}{\partial \theta_2} & \frac{\partial x_{topo}}{\partial \theta_3} \\ \frac{\partial y_{topo}}{\partial \theta_1} & \frac{\partial y_{topo}}{\partial \theta_2} & \frac{\partial y_{topo}}{\partial \theta_3} \\ \frac{\partial z_{topo}}{\partial \theta_1} & \frac{\partial z_{topo}}{\partial \theta_2} & \frac{\partial z_{topo}}{\partial \theta_3} \end{bmatrix} \Rightarrow \Delta\vec{\theta} = \left( \vec{k}_{topo} \cdot \vec{J}_{topo} \right)^{-1} \times \vec{e}_{pos(topo)}$$

Apenas considerando a coordenada  $z$  da anca, como a altura  $h$ , a matriz é simplificada da seguinte forma:

$$\vec{J}_h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial h}{\partial \theta_1} & \frac{\partial h}{\partial \theta_2} & \frac{\partial h}{\partial \theta_3} \end{bmatrix} \Rightarrow \vec{J}_h^T = \begin{bmatrix} 0 & 0 & \frac{\partial h}{\partial \theta_1} \\ 0 & 0 & \frac{\partial h}{\partial \theta_2} \\ 0 & 0 & \frac{\partial h}{\partial \theta_3} \end{bmatrix}$$

Sabendo que, pela cinemática directa:

$$\boxed{h = L_0 + (L_1 + L_2 \cdot c_2 + L_3 \cdot c_{23}) \cdot c_1} \quad \begin{cases} \frac{\partial h}{\partial \theta_1} = -(L_1 + L_2 \cdot c_2 + L_3 \cdot c_{23}) \cdot s_1 \\ \frac{\partial h}{\partial \theta_2} = -(L_2 \cdot s_2 + L_3 \cdot s_{23}) \cdot c_1 \\ \frac{\partial h}{\partial \theta_3} = -L_3 \cdot s_{23} \cdot c_1 \end{cases}$$

E fazendo com que:

$$\begin{cases} CC'_{23} = L_1 + L_2 \cdot c_2 + L_3 \cdot c_{23} \\ SS'_{23} = L_2 \cdot s_2 + L_3 \cdot s_{23} \\ S'_{23} = L_3 \cdot s_{23} \end{cases}$$

Então, o Jacobiano transposto relativo à altura da anca é dado pela seguinte matriz:

$$\vec{J}_h^T = \begin{bmatrix} 0 & 0 & -CC'_{23} \cdot s_1 \\ 0 & 0 & -SS'_{23} \cdot c_1 \\ 0 & 0 & -S'_{23} \cdot c_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & J'_{31} \\ 0 & 0 & J'_{32} \\ 0 & 0 & J'_{33} \end{bmatrix}$$

A partir deste Jacobiano, a compensação angular das juntas devida ao erro de altura, será dada por:

$$\Delta \vec{\theta} = (\vec{k}^T \cdot \vec{J}_h^T) \times \vec{e}_h \Rightarrow \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{bmatrix} = \left( \begin{bmatrix} 0 & 0 & k'_{31} \\ 0 & 0 & k'_{32} \\ 0 & 0 & k'_{33} \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & J'_{31} \\ 0 & 0 & J'_{32} \\ 0 & 0 & J'_{33} \end{bmatrix} \right) \times \begin{bmatrix} 0 \\ 0 \\ e_h \end{bmatrix}$$

$$\begin{cases} \Delta \theta_1 = -k'_{31} \cdot J'_{31} \cdot e_h \\ \Delta \theta_2 = -k'_{32} \cdot J'_{32} \cdot e_h \\ \Delta \theta_3 = -k'_{33} \cdot J'_{33} \cdot e_h \end{cases}$$

## Lei de Controlo Integrada

Integrando as matrizes Jacobiano relativa ao centro de pressão e à altura da anca, obtemos as seguintes equações de controlo definitivas:

$$\vec{J}^T = \begin{bmatrix} J_{11} & J_{21} & J'_{31} \\ J_{12} & J_{22} & J'_{32} \\ J_{13} & J_{23} & J'_{33} \end{bmatrix} = \begin{bmatrix} CC_{23} \cdot c_1 & 0 & -CC'_{23} \cdot s_1 \\ -SS_{23} \cdot s_1 & -CC_{23} & -SS'_{23} \cdot c_1 \\ -S_{23} \cdot s_1 & C_{23} & -S'_{23} \cdot c_1 \end{bmatrix}$$

$$\Delta \vec{\theta} = (\vec{K}^T \cdot \vec{J}^T) \times \vec{e} \quad \Rightarrow \quad \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{21} & k'_{31} \\ k_{12} & k_{22} & k'_{32} \\ k_{13} & k_{23} & k'_{33} \end{bmatrix} \cdot \begin{bmatrix} J_{11} & J_{21} & J'_{31} \\ J_{12} & J_{22} & J'_{32} \\ J_{13} & J_{23} & J'_{33} \end{bmatrix} \times \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \\ e_h \end{bmatrix}$$

$$\begin{cases} \Delta \theta_1 = k_{11} \cdot J_{11} \cdot e_{CoPx} + k_{21} \cdot J_{21} \cdot e_{CoPy} + k'_{31} \cdot J'_{31} \cdot e_h \\ \Delta \theta_2 = k_{12} \cdot J_{12} \cdot e_{CoPx} + k_{22} \cdot J_{22} \cdot e_{CoPy} + k'_{32} \cdot J'_{32} \cdot e_h \\ \Delta \theta_3 = k_{13} \cdot J_{13} \cdot e_{CoPx} + k_{23} \cdot J_{23} \cdot e_{CoPy} + k'_{33} \cdot J'_{33} \cdot e_h \end{cases}$$

Estas três equações, constituem assim a lei de controlo do centro de pressão e da altura, com base nos respectivos sinais de erro. O erro do centro de pressão é calculado a partir dos sensores de força, pela comparação entre o CoP desejado ( $CoP^{ref}$ ) e o medido ( $CoP^{med}$ ):

$$e_{CoP} = CoP^{ref} - CoP^{med}$$

Já o erro de altura é calculado pela comparação entre a altura desejada ( $h^{ref}$ ) e a estimada através da cinemática directa das juntas ( $h^{med}$ ):

$$e_h = h^{ref} - h^{med}$$

$$h = L_0 + (L_1 + L_2 \cdot c_2 + L_3 \cdot c_{23}) \cdot c_1$$

Por questões de simplicidade definiu-se  $k_{1i}=k_{2i}=k_i$  (considere  $k'_{3i} = k'_i \neq k_i$ ):

$$\Delta \vec{\theta} = (\vec{K}^T \cdot \vec{J}^T) \times \vec{e} \quad \Rightarrow \quad \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{bmatrix} = \begin{bmatrix} k_1 & k_1 & k'_1 \\ k_2 & k_2 & k'_2 \\ k_3 & k_3 & k'_3 \end{bmatrix} \cdot \begin{bmatrix} J_{11} & J_{21} & J'_{31} \\ J_{12} & J_{22} & J'_{32} \\ J_{13} & J_{23} & J'_{33} \end{bmatrix} \times \begin{bmatrix} e_{CoPx} \\ e_{CoPy} \\ e_h \end{bmatrix}$$

As equações finais de controlo são, por isso, as descritas na Tabela 19:

Junta lateral do pé:	$\Delta \theta_1 = k_1 \cdot (J_{11} \cdot e_{CoPx} + J_{21} \cdot e_{CoPy}) + k'_1 \cdot J'_{31} \cdot e_h$
Junta dianteira do pé:	$\Delta \theta_2 = k_2 \cdot (J_{12} \cdot e_{CoPx} + J_{22} \cdot e_{CoPy}) + k'_2 \cdot J'_{32} \cdot e_h$
Junta do Joelho	$\Delta \theta_3 = k_3 \cdot (J_{13} \cdot e_{CoPx} + J_{23} \cdot e_{CoPy}) + k'_3 \cdot J'_{33} \cdot e_h$

**Tabela 19: Leis de controlo utilizando a matriz Jacobiana de centro de massa e de altura.**

Vamos agora pôr em prática os algoritmos de controlo descritos e observar os seus resultados. Os controladores foram configurados de modo a procurar a posição de equilíbrio, ou seja, o ponto, cujo centro de pressão, coincide com o centro geométrico do pé: coordenadas  $(x,y)=(0,0)$ .

$$e_{CoPx} = CoP_x^{[ref]} - CoP_x^{[med]} = -CoP_x^{[med]} \qquad e_{CoPy} = CoP_y^{[ref]} - CoP_y^{[med]} = -CoP_y^{[med]}$$

Considerando  $CoP_x$  e  $CoP_y$  como os centros de pressão medidos em cada componente espacial  $(x,y)$ , as leis de controlo aplicadas foram as seguintes, tanto para o caso do controlador proporcional (Tabela 20), como para o caso com controlador Jacobinao (Tabela 21):

Junta lateral do pé:	$\Delta\theta_1 = -K_{Px} \cdot CoP_x$
Junta dianteira do pé:	$\Delta\theta_2 = -K_{Py} \cdot CoP_y$

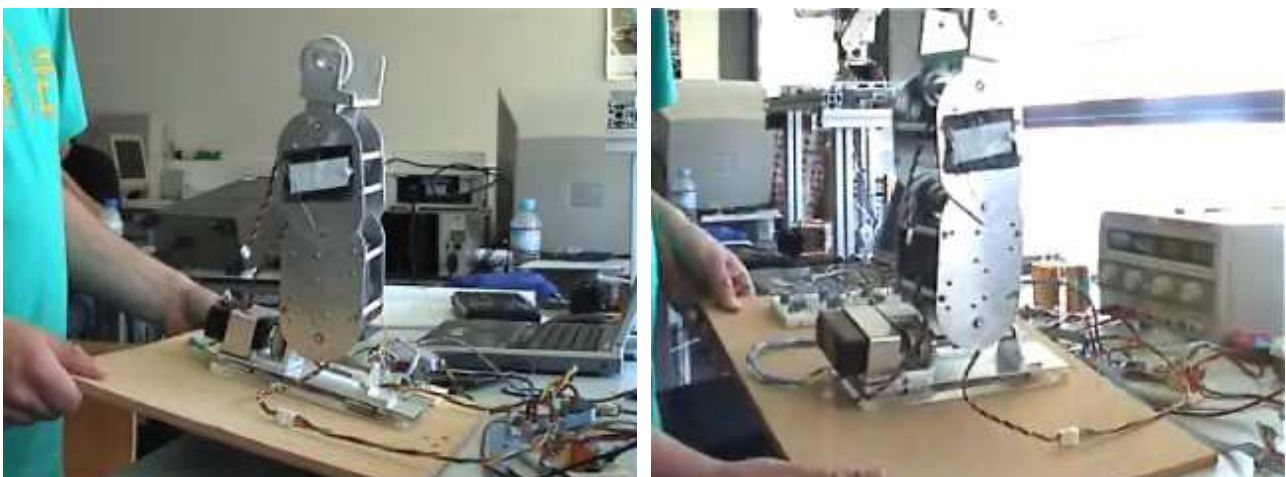
**Tabela 20: Equações de controlo para o compensador proporcional, considerando o CoP de referência nulo.**

Junta lateral do pé:	$\Delta\theta_1 = -k_1 \cdot (J_{11}(\vec{\theta}) \cdot CoP_x + J_{21}(\vec{\theta}) \cdot CoP_y)$
Junta dianteira do pé:	$\Delta\theta_2 = -k_2 \cdot (J_{12}(\vec{\theta}) \cdot CoP_x + J_{22}(\vec{\theta}) \cdot CoP_y)$
Junta do joelho:	$\Delta\theta_3 = -k_3 \cdot (J_{13}(\vec{\theta}) \cdot CoP_x + J_{23}(\vec{\theta}) \cdot CoP_y)$

**Tabela 21: Equações de controlo para o compensador com o Jacobiano, considerando o CoP desejado nulo.**

#### 4.5.4 RESULTADOS EXPERIMENTAIS: VARIAÇÃO DO PLANO DE SUPORTE

Esta experiência põe em prática a ideia sugerida sobre a tarefa da perna de suporte na realização da locomoção, ou seja, na presença de irregularidades do solo, tanto pela variação do seu declive ou pela presença de corpos estranhos no solo, a perna deve mover-se de modo a garantir o equilíbrio da estrutura. Aplicaram-se variações na inclinação do plano de suporte a uma perna, cujo controlador deverá procurar a posição de equilíbrio, ou seja, a posição vertical. A Fig. 4.21 apresenta a experiência realizada: segundo o sistema de eixos apresentado na Fig. 4.20, efectuaram-se variações na inclinação do plano, primeiramente segundo o eixo dos  $yy$  e depois segundo o eixo  $xx$ .



**Fig. 4.21: da inclinação do plano de suporte segundo o eixo  $yy$  (esquerda) e o eixo  $xx$  (direita).**



Os resultados estão expressos nos gráficos seguintes onde se visualiza a posição do centro de pressão segundo os eixos  $xx$  e  $yy$  (da Fig. 4.20). As referências a cada servo dizem respeito aos descritos a seguir:

- Servo 0: junta lateral do pé (usada para compensar o  $CoP$  segundo o eixo  $xx$ ) –  $\theta_1$ ;
- Servo 1: junta dianteira do pé (usada para compensar o  $CoP$  segundo o eixo  $yy$ ) –  $\theta_2$ ;
- Servo 2: junta do joelho (usado para compensação adicional do  $CoP$  segundo o eixo  $yy$ ) –  $\theta_3$ .

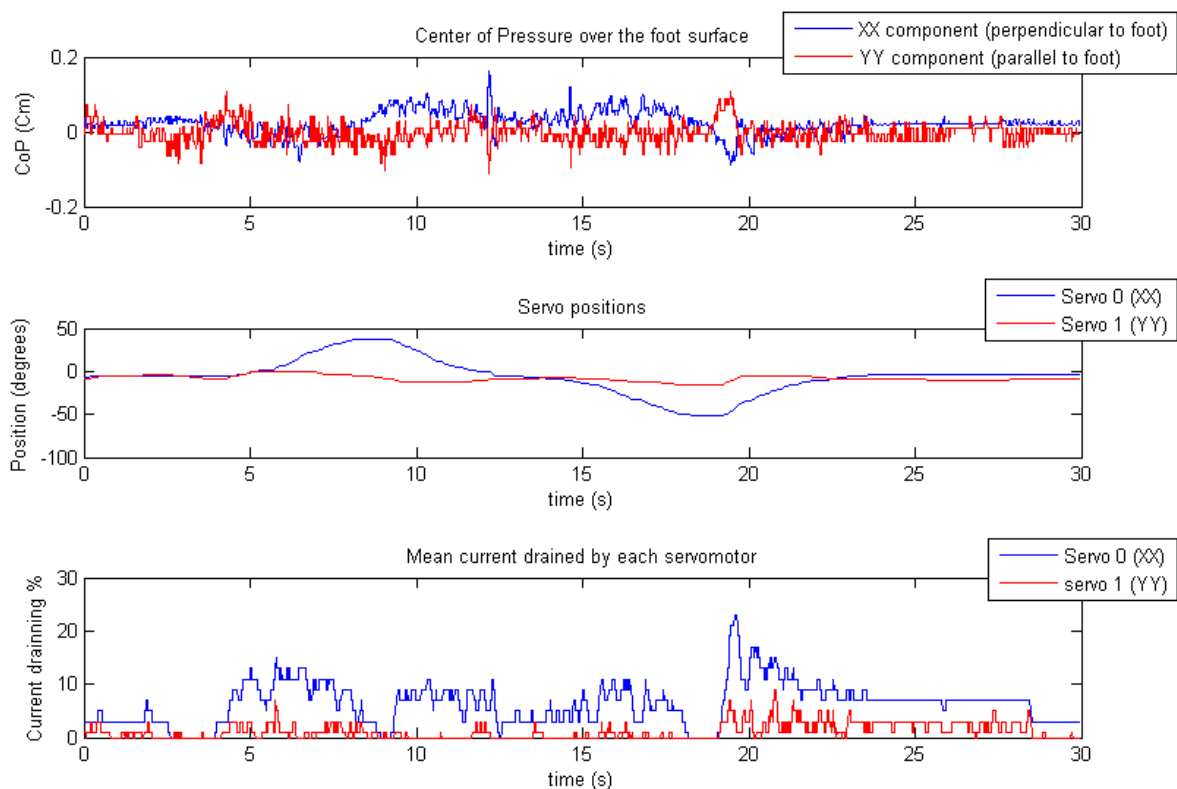
Relativamente à corrente consumida por cada um, apresentam-se em unidades normalizadas de 0 a 100 (percentagem), para um tempo de medição de 30s. Os resultados são comparados entre os dois tipos de controlador apresentados.

### Variação da inclinação do plano segundo o eixo $xx$

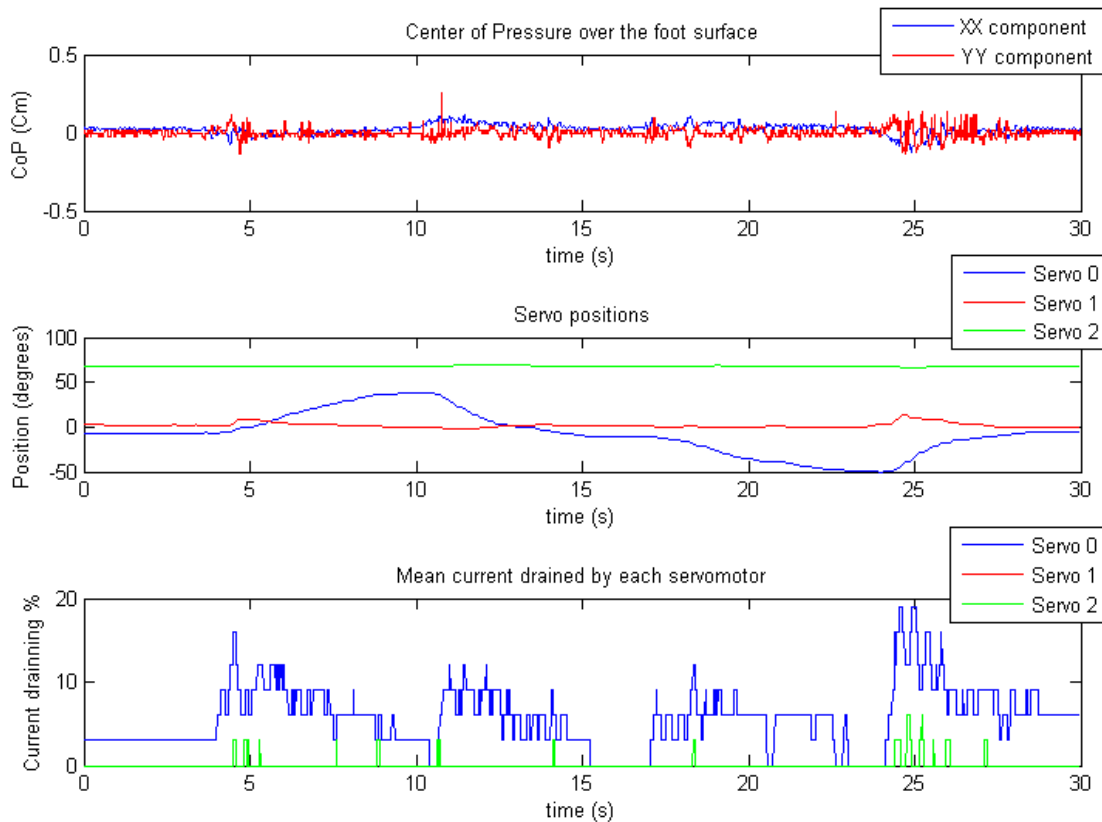
Neste primeiro ensaio apenas se variou a componente  $xx$  do plano, verificando-se, segundo os gráficos da Fig. 4.22 e da Fig. 4.23, algumas variações na componente  $xx$  do  $CoP$  durante a compensação. Por análise, pode-se observar que, embora se incline o plano até cerca de  $45^\circ$ , o controlador não permite erros do  $CoP$  superiores a 2 mm do ponto de equilíbrio.

Usando o controlador proporcional pode-se constatar que o servomotor responsável por compensar esta componente (servo 0 – junta lateral do pé) foi o único a ser utilizado atingindo um deslocamento máximo de cerca de  $45^\circ$ , por forma a compensar a inclinação de  $45^\circ$  introduzida.

Comparando agora com o controlador que utiliza o Jacobiano, pode-se observar um comportamento muito semelhante, também com um deslocamento do servo 0 na ordem dos  $45^\circ$ . Como a junta do joelho pouco interfere com a componente  $xx$  do  $CoP$ , esta manteve-se praticamente imóvel.



**Fig. 4.22: Variação do declive do plano segundo o eixo  $xx$  (controlo proporcional).**

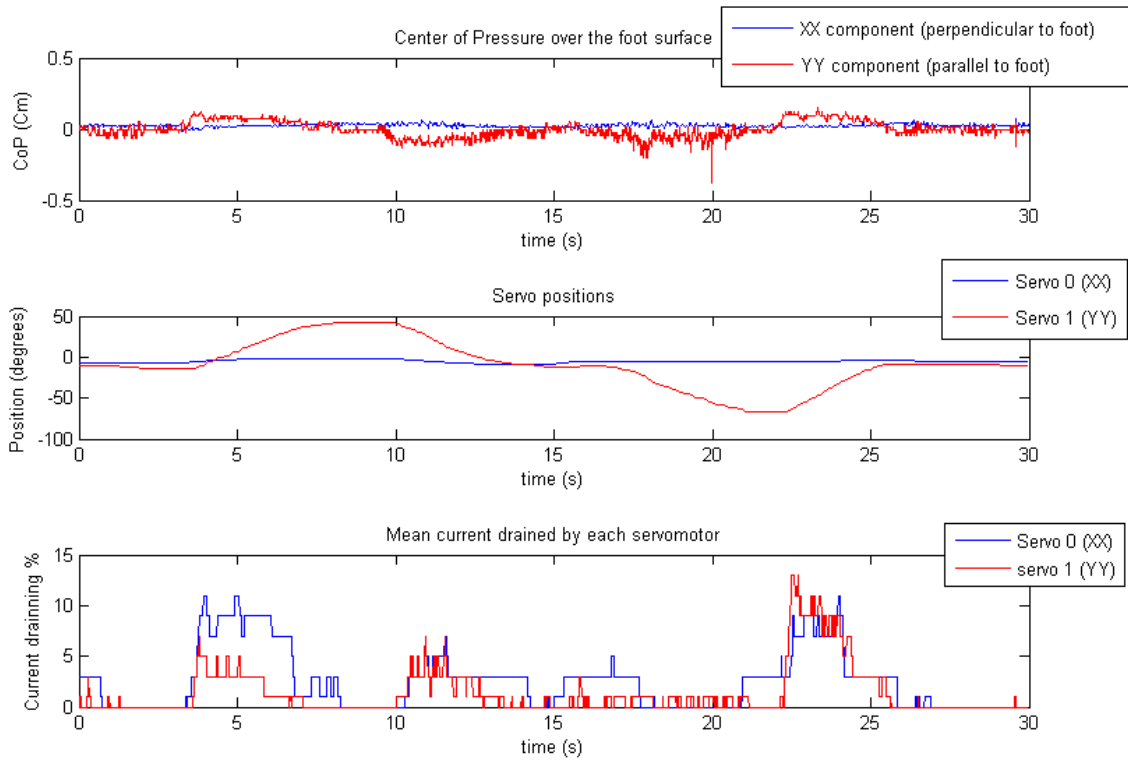


**Fig. 4.23: Variação do declive do plano segundo o eixo xx - controlo pelo Jacobiano.**

### Variação da inclinação do plano segundo o eixo yy

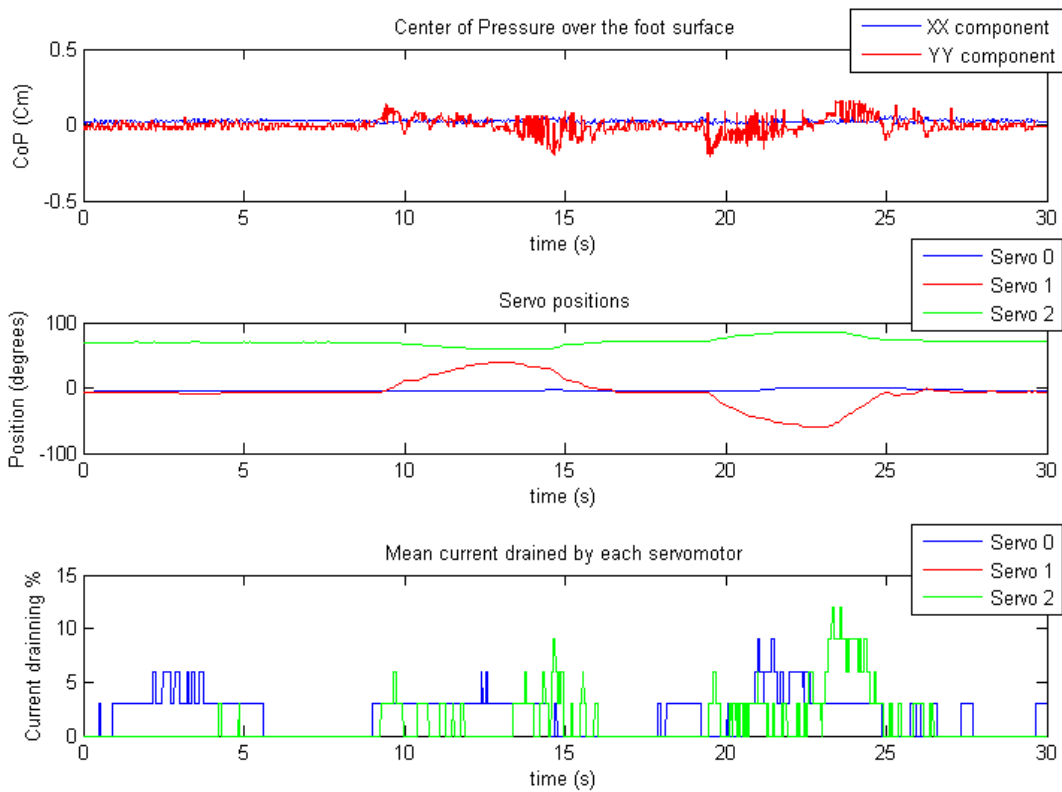
Efectuando agora variações da inclinação do plano segundo o eixo yy, pode-se verificar pela Fig. 4.24, um comportamento análogo ao caso anterior, mas com a diferença de apenas ser utilizados o servo associado ao eixo yy (servo 1) para compensar a variação do CoP no eixo yy.

As primeiras experiências efectuadas (não visualizadas nestes gráficos) evidenciaram uma inferior sensibilidade na compensação desta componente apresentando uma reacção bastante lenta no controlo. Tal é explicado pelo formato rectangular do pé: dado que a distância entre o par dianteiro e traseiro dos sensores é muito maior que entre os conjuntos laterais, é de esperar que a sensibilidade na medição do  $CoP_y$  seja bastante menor que na medição do  $CoP_x$ . Para colmatar este problema diminuiu-se o *offset* usado para a adaptação dos valores fornecidos pela ADC que mede estes sensores (ver secção 4.4.2), traduzindo-se numa menor força de pré-carga, e conseqüentemente, maior sensibilidade. Este procedimento apenas se efectuou para o eixo yy (*offset<sub>y</sub>*) permitindo aumentar a sensibilidade da compensação na componente yy sem afectar a componente xx, apresentando como único inconveniente o aparecimento de alguma oscilação devida a ruído.



**Fig. 4.24: Variação do declive do plano segundo o eixo yy (controlo proporcional).**

Utilizando o algoritmo de controlo incluindo a matriz Jacobiana (Fig. 4.25), podemos observar a novidade do uso da junta do joelho na compensação, introduzindo maior eficiência na compensação.



**Fig. 4.25: Variação do declive do plano segundo o eixo yy - controlo pelo Jacobiano.**

## Varição da inclinação do plano simultaneamente segundo o eixo $xx$ e $yy$

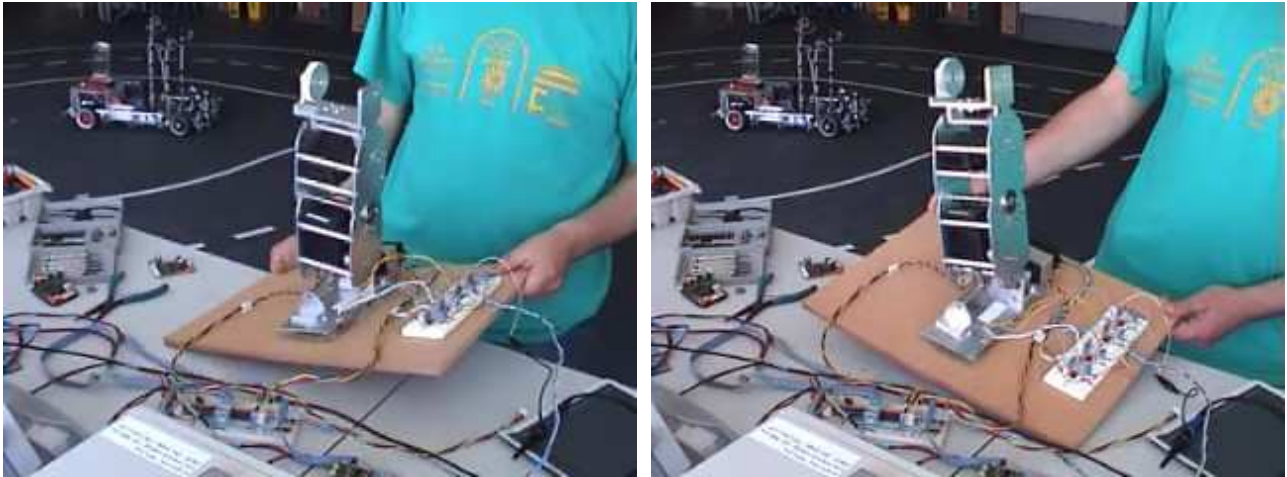


Fig. 4.26: Inclinação do plano segundo os eixos  $xx$  e  $yy$ .

Os gráficos da Fig. 4.27 apresentam os resultados para a variação do plano simultaneamente nos eixos  $xx$  e  $yy$ , observando-se deslocamentos simultâneos por parte dos dois servomotores do pé, de modo a corrigir as variações do  $CoP$  nas duas componentes espaciais.

Embora não estejam presentes os resultados para o compensador com a matriz Jacobiana, os resultados são bastante semelhantes, com a exceção da junta do joelho que participa também no processo de compensação.

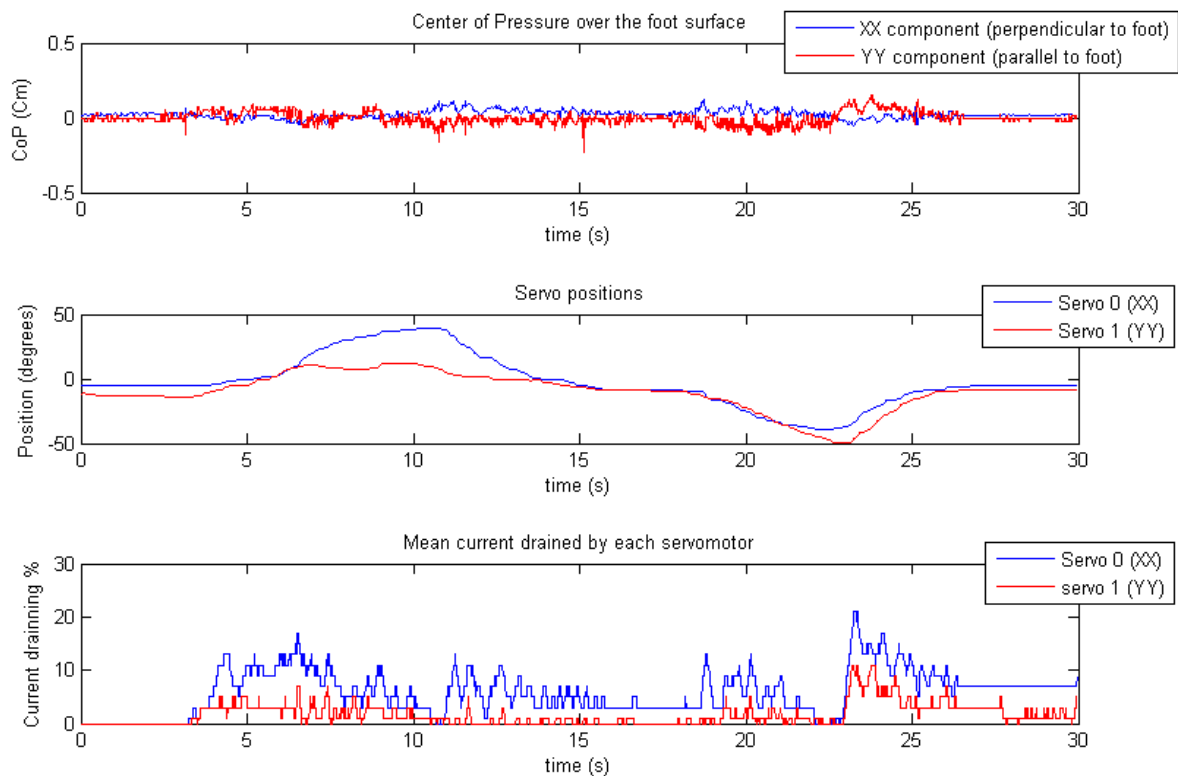


Fig. 4.27: Variação do declive do plano na diagonal dos eixos  $xx$  e  $yy$  (controlo proporcional).

#### 4.5.5 RESULTADOS EXPERIMENTAIS: RESPOSTA A PERTURBAÇÕES EXTERNAS

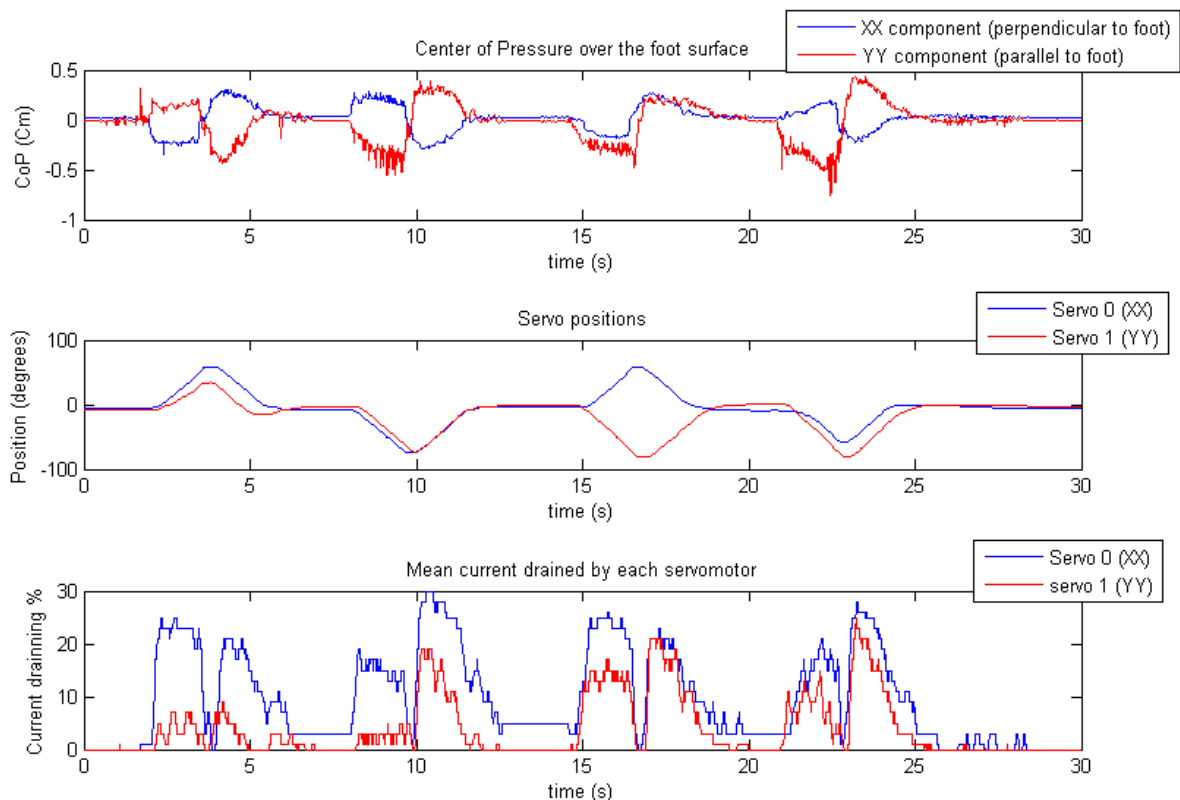
A segunda experiência explora os efeitos da presença de perturbações externas. Entende-se por perturbações externas, as forças que não são aplicadas directamente sobre os pés, mas que de forma indirecta são sentidas pelos sensores de força. Um exemplo é a aplicação de forças sobre uma perna ou sobre a estrutura da plataforma, como toques ou empurrões, em que inevitavelmente os sensores de força detectarão estas anomalias e o controlador responderá da mesma forma como se se tratasse de uma aplicação directa da força sobre o pé.

Seria interessante fazer com que o controlador distinguísse estes tipos de aplicação, podendo eventualmente aplicar diferentes tipos de controlo para melhor se adequar à situação em causa, mas infelizmente, não existem meios de detectar a origem das forças pelo que ficaremos limitados a um único tipo de controlo.

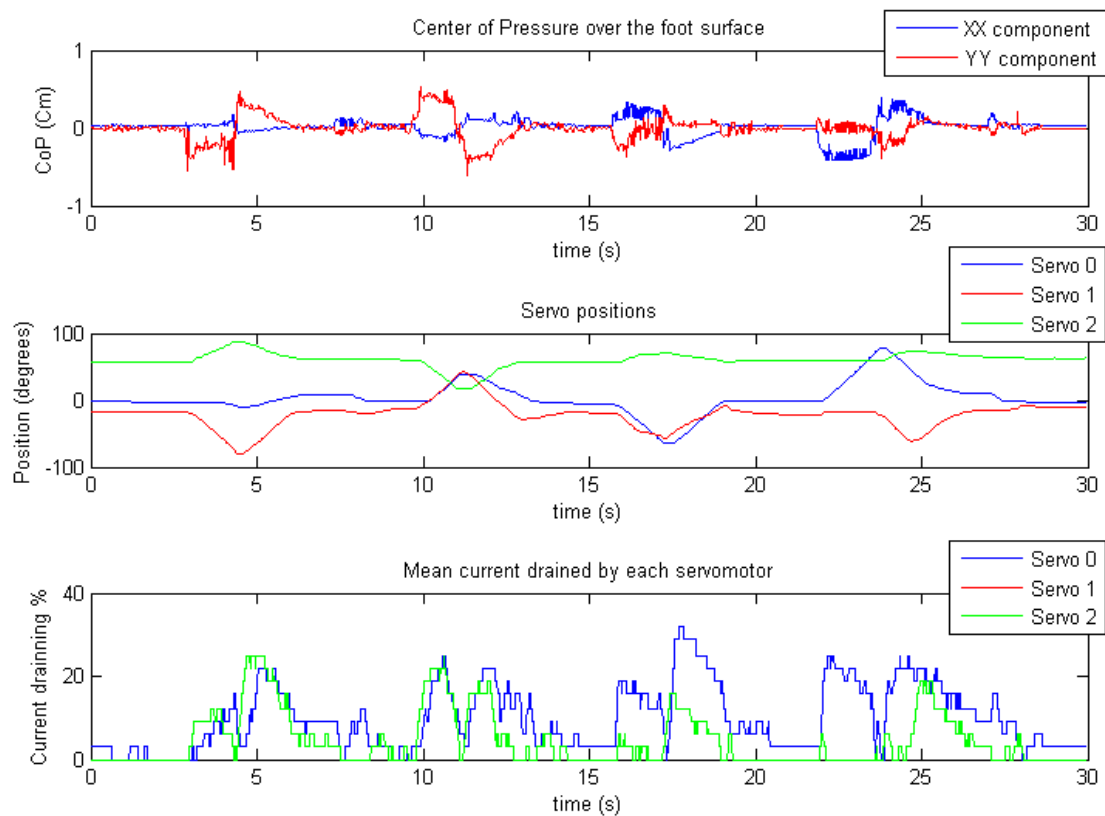
Aplicaram-se perturbações, na forma de “empurrões”, sobre a perna em teste na seguinte sequência, considerando a perspectiva traseira da perna:

1. Frente;
2. Trás;
3. Esquerda;
4. Direita.

“Empurrões” suficientemente bruscos foram aplicados de forma a provocar o levantamento do pé sobre o solo, de modo a analisar o comportamento do controlador no caso em que o centro de massa da perna deixa de coincidir com o centro de pressão.



**Fig. 4.28: Introdução de perturbações externas na perna (frente, trás, esquerda e direita) - controlo Proporcional.**



**Fig. 4.29: Introdução de perturbações externas na perna (frente, trás, esquerda e direita) - controlo pelo Jacobiano.**

Um controlador baseado no centro de massa como o implementado, procura fazer o deslocamento oposto ao introduzido pela nova força: se uma força externa for aplicada sobre a perna, o controlador assumirá que o centro de massa se deslocou para o extremo oposto aonde foi aplicada a força, e, por isso, efectuará o deslocamento oposto ao aplicado, de modo a repor o centro de massa na posição de equilíbrio. Este efeito acontece nas experiências traduzidas pelos gráficos da Fig. 4.28 e da Fig. 4.29.

Embora este efeito seja desejável, uma vez que procura contrariar a presença de forças externas, verifica-se o problema de se deixar de garantir a postura vertical da perna na situação de equilíbrio, uma vez que a presença da força estranha faz divergir o *CoP* medido do *CoM* da estrutura. Mais ainda, se a base do pé deixar de estar em completo contacto com o solo, o *CoP* tende a deslocar-se e a manter-se na extremidade oposta à aplicação da força, independentemente da postura da perna, resultando na saturação das juntas. Tal situação corresponde à máxima divergência entre *CoP* e *CoM*.

Este fenómeno demonstra que nem sempre é necessária compensação negativa (ou oposta) no controlo do centro de massa. Tal também é verdade no caso da adaptação dos pés a pisos irregulares ou não horizontais. Para este caso as juntas dos pés devem adaptar-se até toda a sua base assentar no piso. Para tal, também pode ser usado como *feedback* a posição do centro de pressão, mas agora com compensação positiva no controlo – deslocamento dos servos no sentido oposto. Apenas depois de este procedimento estar completo (base do pé em contacto com o solo) o algoritmo tradicional é que pode entrar em funcionamento. Este controlador de compensação oposta poderá ser utilizado na perna de locomoção até esta se encontrar completamente assente sobre o solo, e só a partir deste ponto, o algoritmo de equilíbrio com base no centro de massa entraria em funcionamento, passando a atribuir a função de suporte a esta perna.

Uma outra estratégia, é simplesmente desligar o controlador de equilíbrio, bem como também o sinal de PWM aplicado aos motores do pé. Desta forma, encontrar-se-iam em movimento livre, adaptando-se naturalmente ao solo, à medida que a perna baixa. Só quando totalmente em contacto, os motores se reactivariam, preservando esta nova posição adquirida.

#### 4.5.6 RESULTADOS EXPERIMENTAIS: RESPOSTA DO PÉ LIVRE

Nesta experiência adicional, levantou-se a perna, de modo a suspender o pé, e observou-se o comportamento do controlador. Segundo a Fig. 4.30 e a Fig. 4.31, quando o pé está suspenso, o valor do centro de pressão afasta-se do ponto de equilíbrio, o que provoca o deslocamento dos servomotores. Como este deslocamento em nada afecta a medição de forças nos sensores (dada a ausência de forças de reacção), o centro de pressão continuará desequilibrado, independentemente da posição dos servos, levando à saturação na actuação dos servos.

Estes resultados sugerem que a distribuição de forças sobre a base do pé não é homogénea. Embora após activação todos os sensores sejam inicializados num determinado valor inicial, a força absoluta inicial (pré-carga) aplicada em cada um é diferente, o que faz com que na ausência do plano de suporte as forças medidas sejam diferentes de sensor para sensor resultando num valor do *CoP* diferente do valor original. É este *offset* que provoca o deslocamento dos servos até à saturação.

Para esta falta de homogeneidade contribuem os seguintes factores;

- Diferente natureza das placas de acrílico de suporte aos extensómetros: como os sensores da frente são mais sensíveis que os traseiros, o *CoP* medido é deslocado para a traseira do pé quando levantado.
- Plataforma superior do pé não homogénea;

Mesmo resolvidos estes dois aspectos, ocorre um novo problema: se no momento da calibração, o *CoM* do robot não coincide com a posição de equilíbrio, como é o caso em que a perna não está exactamente na vertical, verifica-se uma diferença nas forças absolutas aplicadas inicialmente. Esta diferença revelar-se-á, na ausência do solo, resultando no desequilíbrio permanente dos sensores. Como condições para que tal não aconteça é necessário que:

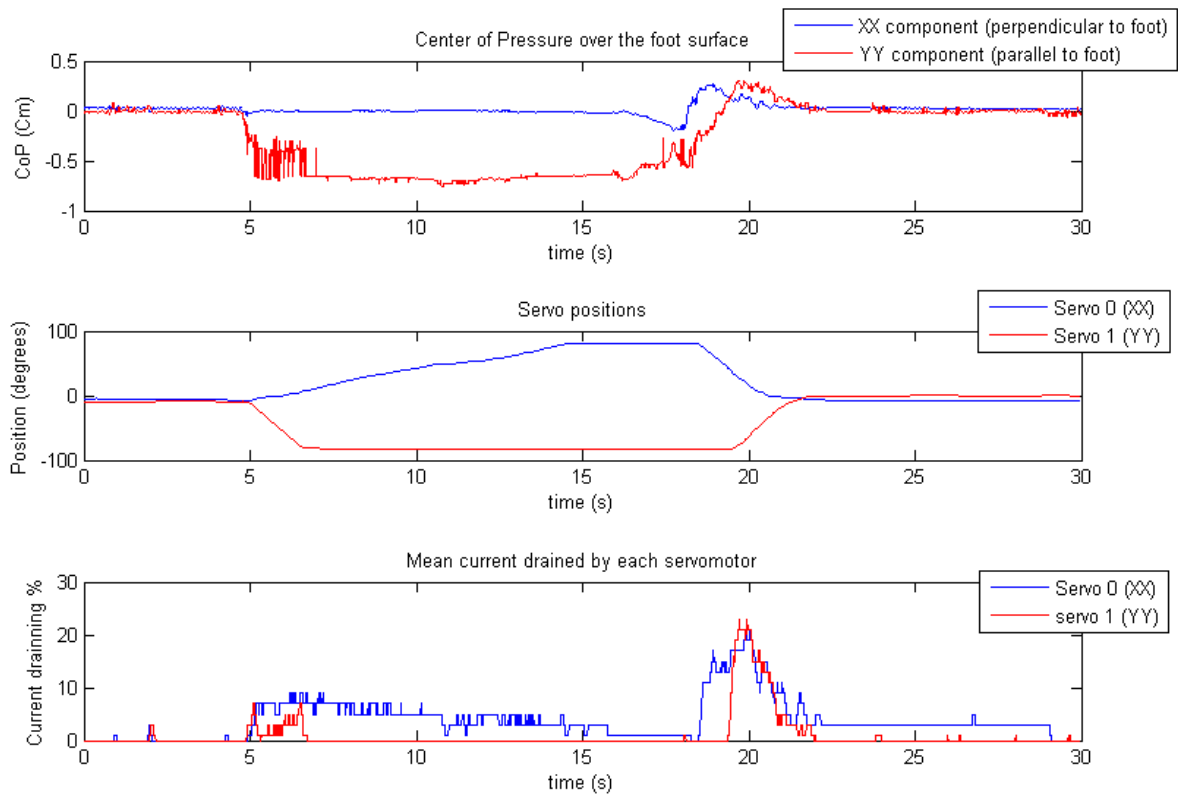
- A plataforma superior do pé seja completamente homogénea;
- As quatro placas de acrílico sejam exactamente iguais, e estejam fixas à plataforma superior de forma semelhante entre si.
- O centro de massa da estrutura do robot coincida com o centro de pressão nulo no momento da calibração.

Estas são condições muito difíceis de garantir, pelo que ainda não existem soluções em vista para resolver este problema.

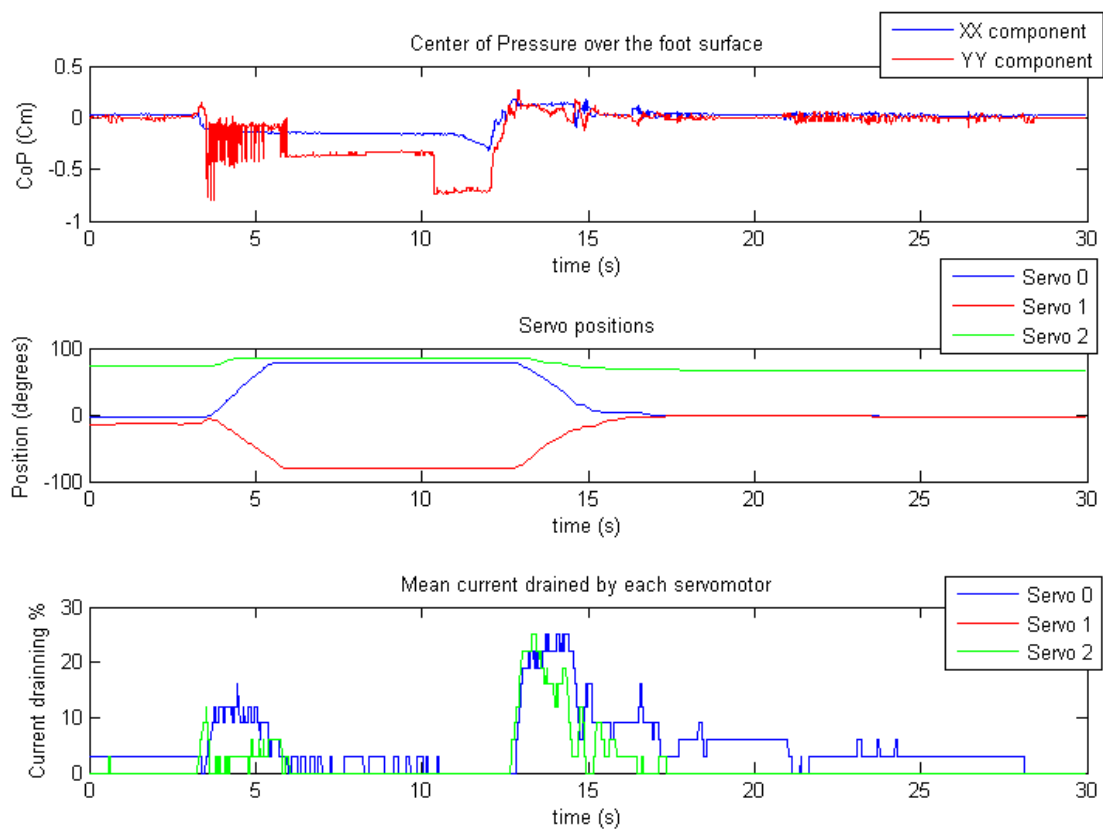
A Fig. 4.30 e a Fig. 4.31, revelam os dados numéricos do centro de pressão medido e da posição angular actuada para o controlador proporcional e para o Jacobiano respectivamente, para duas circunstâncias distintas:

- Levantamento do pé: os servomotores saturam numa das posições extremas, provocado pelo permanente desequilíbrio do centro de pressão medido;
- Reposição do pé no solo: o controlador converge novamente para o valor original e os motores estabilizam numa posição próxima à inicial.

Para a situação descrita na secção anterior, de adaptação de um pé ao solo, esta experiência demonstra claramente que o método de compensação positiva não resulta quando este se encontra suspenso acima do solo, dada a inabilidade deste controlador de funcionar convenientemente na ausência de forças de reacção. Por este motivo, o método de adaptação livre, por desactivação do PWM aplicado aos motores, é o caminho a seguir para este tipo de eventos.



**Fig. 4.30: Levantamento do pé no ar - controlo proporcional.**



**Fig. 4.31: Levantamento do pé no ar - controlo pelo Jacobiano.**



## 4.6 LOCOMOÇÃO ATRAVÉS DO CONTROLO DO COP

A secção anterior, demonstrou que é possível utilizar o controlador de equilíbrio para sustentar as pernas na postura vertical, face à inclinação do plano de suporte. Dois possíveis controladores para esta tarefa foram desenvolvidos para alcançar esse objectivo:

- Controlo proporcional de cada junta ortogonal do pé (nos eixos  $xx$  e  $yy$ );
- Controlo usando a matriz Jacobiana.

Dado que o controlador usando a matriz Jacobiana permite controlar todas as três juntas da perna, e revelando-se pouco sensível a imprecisões no cálculo desta matriz, este foi o controlador adoptado, obtendo-se resultados bastante satisfatórios.

Até aqui o único objectivo era o de equilibrar a perna face a variações e irregularidades do solo. Contudo, também poderemos, com o mesmo controlador, realizar trajectórias, pela variação do CoP desejado. Se em vez de utilizarmos como referência o  $\text{CoP}=(0,0)$  correspondente à postura vertical, usarmos outros valores, a perna desloca-se para outras posturas efectuando movimento. Ora, é do interesse que quando o robot se encontra sobre a perna de suporte durante a locomoção, além de fornecer o equilíbrio também seja capaz de a mover ajudando a realizar as sequências de locomoção. Para tal, e como já foi enunciado, ir-se-á actuar no valor referência do CoP ( $\text{CoP}_{\text{ref}}$ ) no controlador de equilíbrio para cumprir este objectivo (Fig. 4.19).

Nesta secção, vários estudos serão feitos sobre os sensores de força, de modo a averiguar os seguintes aspectos:

- Relação entre o Centro de Pressão medido e o Centro de Gravidade, em circunstâncias estáticas;
- Relação entre o Centro de Pressão medido e o Centro de Gravidade, durante a realização de trajectórias;
- Execução de trajectórias de vários formatos, e comparação dos trajectos efectuados com os previstos.

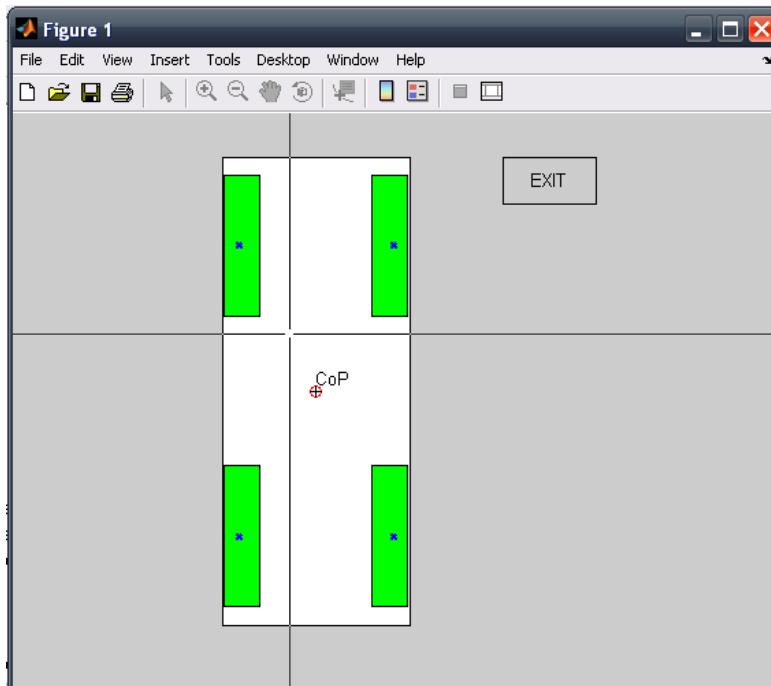
### 4.6.1 CONTROLO DO COP DE REFERÊNCIA

#### Funções de Controlo

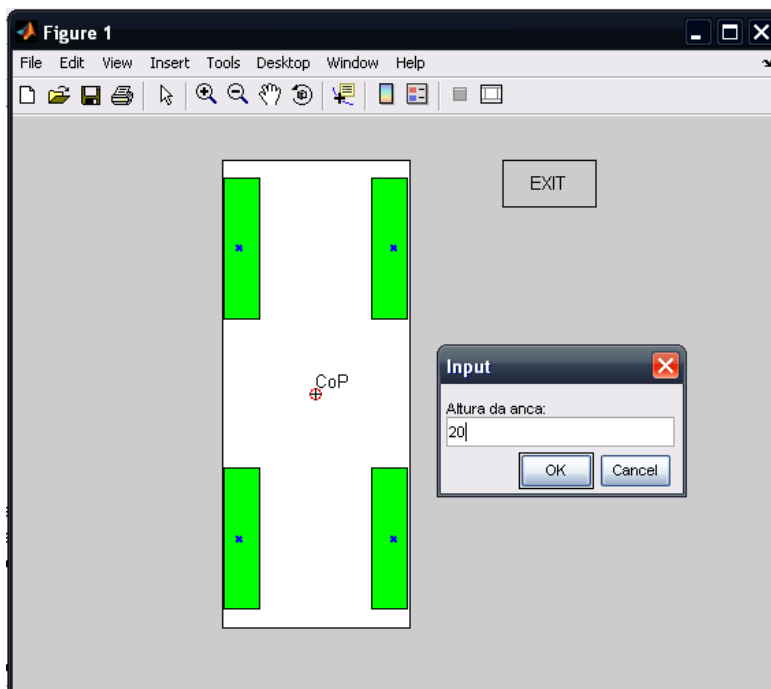
Com base nas rotinas de controlo do robot, descritas na secção 2.2.1, foi desenvolvida uma aplicação gráfica em Matlab, para manipulação do CoP de referência, utilizando o rato como interface para introdução das coordenadas.

A Fig. 4.32 apresenta um exemplo de como as coordenadas bidimensionais do centro de pressão desejado são introduzidas: a mira que se observa é controlada pelo rato que selecciona o ponto espacial desejado. Além disso também é possível controlar a altura desejada da anca: para tal, utiliza-se o botão direito do rato que fará aparecer uma caixa de diálogo para introduzir numericamente a altura a atribuir à anca (Fig. 4.33).

Esta aplicação, e outras semelhantes, serão úteis adiante para a análise do comportamento estático dos sensores de força quando o centro de pressão referência é alterado.



**Fig. 4.32: Actuação sobre o Centro de Pressão referência.**



**Fig. 4.33: Actuação sobre a altura desejada da anca.**

## 4.6.2 ANÁLISE ESTÁTICA DO COP DE REFERÊNCIA

A análise estática quando o CoP referência é alterado, é importante para verificar se o centro de pressão medido corresponde ao actuado, analisando assim se existe algum erro associado. Também pela comparação do CoP de referência com a cinemática directa das juntas do joelho e da anca, é possível obter uma relação entre os valores actuados do CoP e as coordenadas reais das juntas, permitindo assim controlar a posição destas juntas em termos de coordenadas espaciais.

As experiências realizadas a seguir utilizam a aplicação gráfica descrita atrás fazendo a actuação do CoP para vários pontos específicos. De igual forma é realizada a leitura sensorial:

1. Actualização do CoP referência;
2. Esperar que o controlador atinja o equilíbrio;
3. Esperar alguns segundos, para assegurar a paragem dos servos;
4. Leitura dos dados sensoriais;
5. Voltar a 1.

Os dados sensoriais lidos são os seguintes:

- Sensores de força para cálculo do centro de pressão actual;
- Ângulos das diversas juntas da perna, para cálculo da cinemática directa.

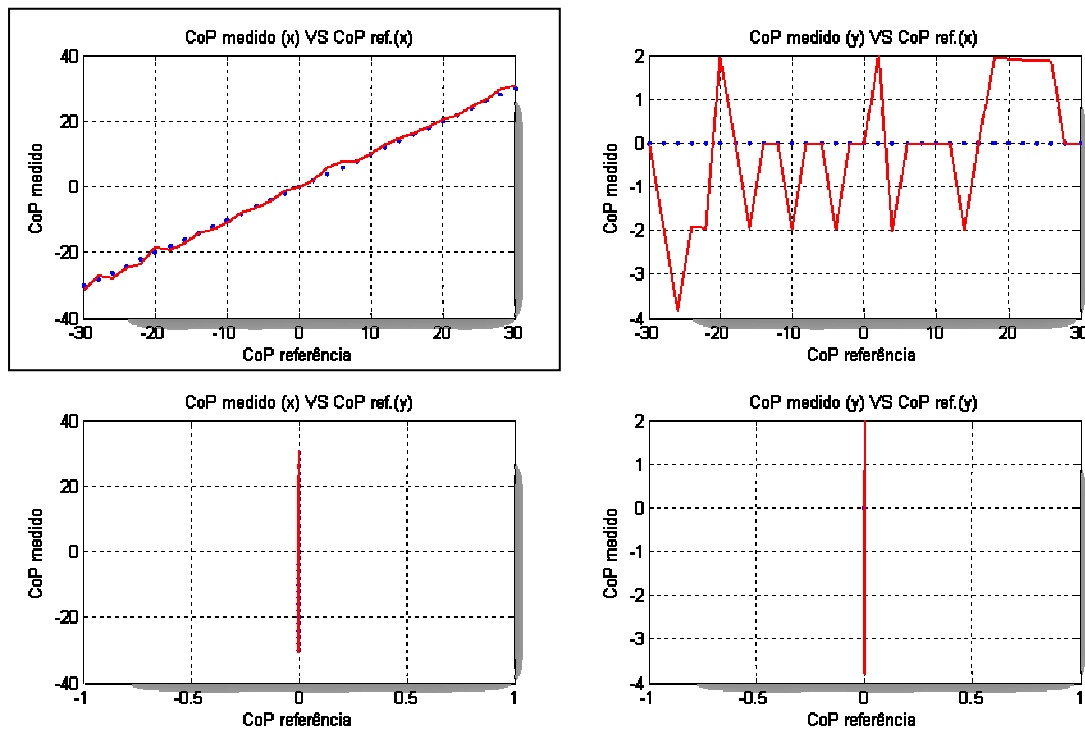
As direcções dos eixos cartesianos aplicados seguem a convenção da Fig. 4.20.

### Variacão do CoP no eixo xx

Efectuando 30 passos desde as coordenadas  $\text{CoP}=(30,0,0)$  até  $\text{CoP}=(-30,0,0)$ , obtiveram-se os dados dos sensores de força e da cinemática directa do joelho e da anca apontados na Fig. 4.34 e na Fig. 4.35. Note que estas unidades, por enquanto, não têm qualquer significado, dependendo da sensibilidade dos sensores de força, bem como da estrutura da perna em teste.

Pela análise à Fig. 4.34 (coordenada  $x$ ), podemos concluir que o controlador de CoP está a funcionar regularmente cumprindo todos os pontos de referência atribuídos. Já o CoP na componente  $y$  apresenta variações desprezáveis, tal como esperado, dado que o percurso do CoP referência é apenas feito no eixo  $xx$ .

No que respeita à cinemática directa, as juntas do joelho e da anca apresentam percursos aproximadamente lineares perto da coordenada nula (postura vertical), mas para pontos mais distantes (perna com maior inclinação) a curva arqueia assemelhando-se à função arco-seno. Tal comportamento deve-se à gravidade que atribui uma maior inércia ao movimento nos pontos mais inclinados. A componente  $yy$  apresenta igualmente flutuações desprezáveis tal como o esperado.



**Fig. 4.34: Relação entre o CoP medido e o actuado, para 30 passos ao longo do eixo xx.**

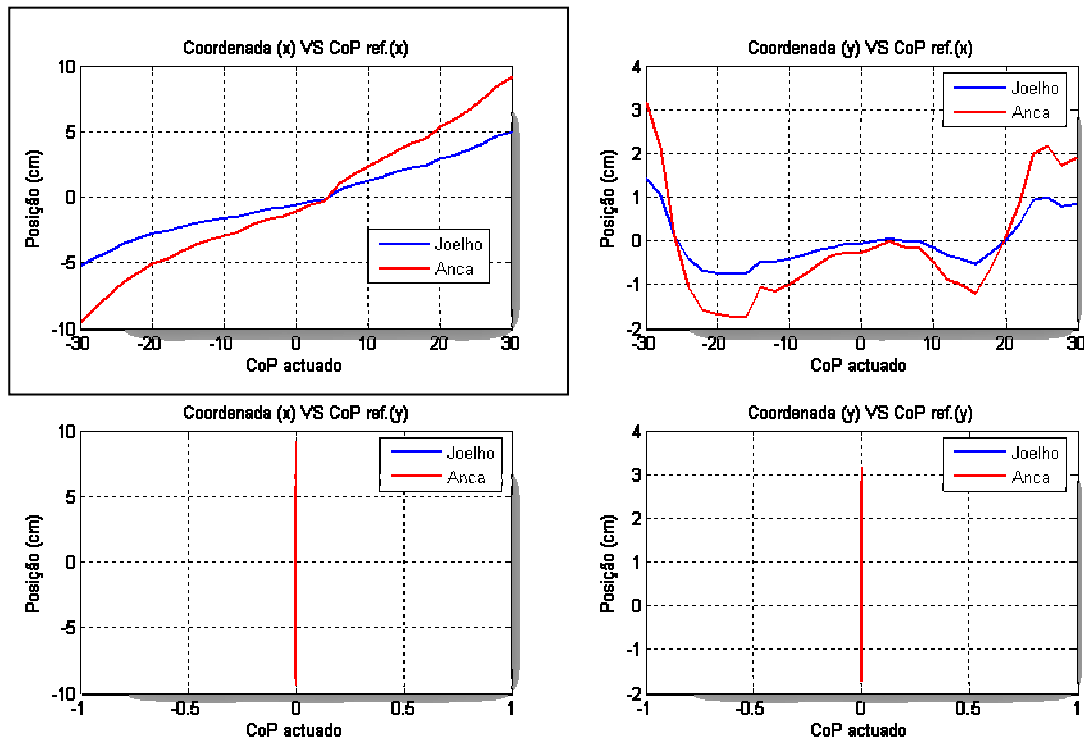


Fig. 4.35: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo xx.

### Variação do CoP no Eixo yy

Efectuando a mesma experiência, mas agora realizando 32 passos ao longo do eixo yy:  $CoP=(0,-30,0) \rightarrow (0,34,0)$ , obtiveram-se os dados da Fig. 4.36 e Fig. 4.37. Como se pode comprovar, o comportamento é muito semelhante ao observado na experiência anterior.

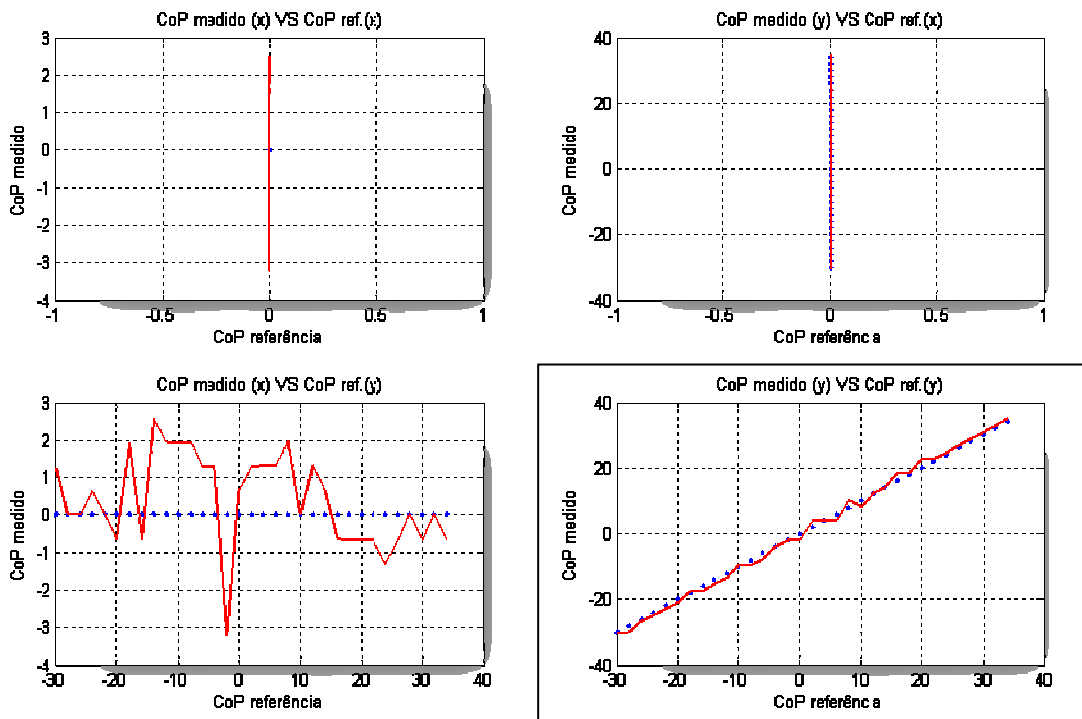


Fig. 4.36: Relação entre o CoP medido e o actuado, para 32 passos ao longo do eixo yy.

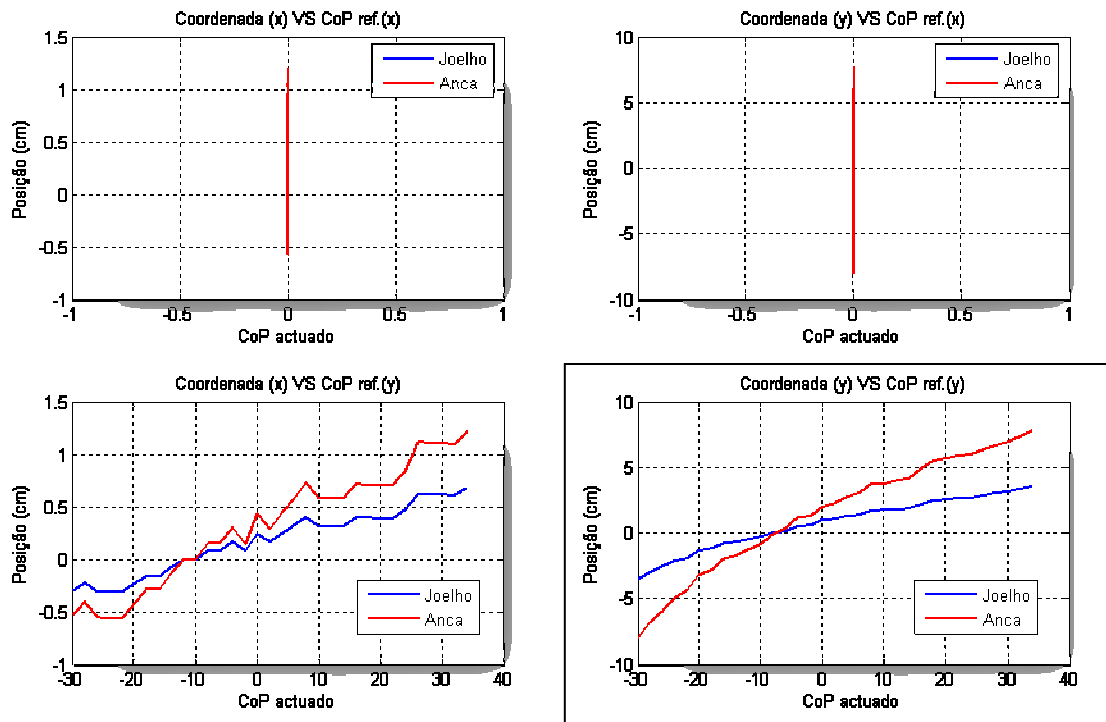


Fig. 4.37: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo yy.

### Variação do CoP no eixo xy

Efectuando agora 25 passos ao longo da diagonal entre os eixos xx e yy –  $\text{CoP}=(25,-25,0)\rightarrow(-25,25,0)$  – amostraram-se os dados apresentados na Fig. 4.38 e Fig. 4.39.

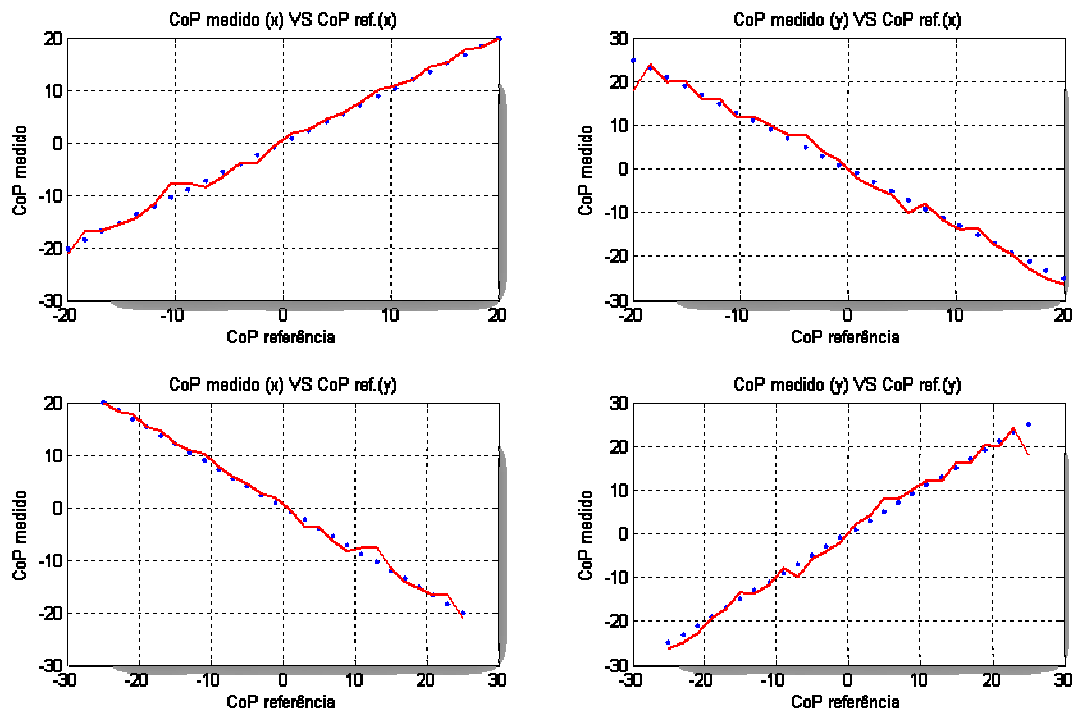


Fig. 4.38: Relação entre o CoP medido e o actuado, para 25 passos ao longo do eixo xy.

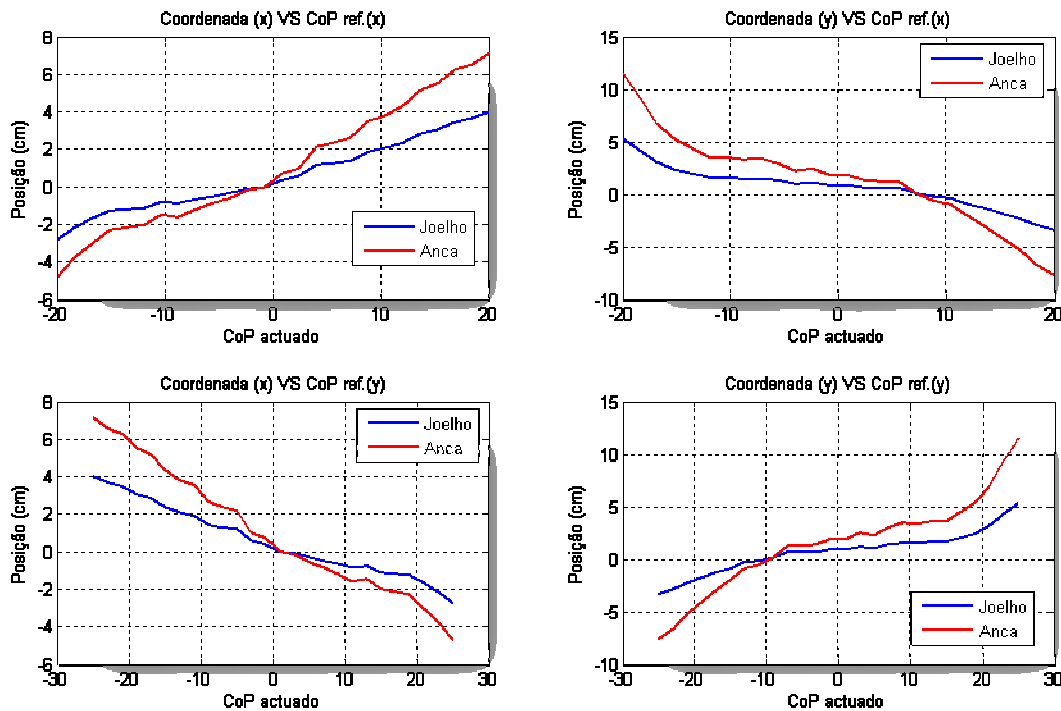


Fig. 4.39: Relação entre o CoP medido e a cinemática directa do joelho e da anca, ao longo do eixo  $xy$ .

### 4.6.3 ANÁLISE DINÂMICA DO COP REFERÊNCIA

A análise dinâmica é entendida como a amostragem da resposta durante a aplicação de uma trajectória às diversas juntas do sistema. Na situação em concreto aplicar-se-á uma trajectória polinomial de quinta ordem (velocidade e aceleração inicial e final nulas) ao centro de pressão de referência ( $CoP_{ref}$ ) e a saída dos sensores de força, bem como a posição angular das várias juntas, serão amostrados durante a execução desta trajectória.

O procedimento das experiências executadas está descrito a seguir:

1. Inicialização do  $CoP_{ref}$  para a execução de uma trajectória polinomial de 5ª ordem;
2. Activação do controlador de equilíbrio;
3. Definição do período da trajectória e do ganho do controlador de equilíbrio;
4. Execução da trajectória para um determinado  $CoP_{ref}$  final;
5. Ao longo da trajectória registar a saída dos sensores de pressão e de posição dos servos;
6. No fim da trajectória, guardar os resultados sensoriais, e reposicionar a perna para o  $CoP_{ref}$  inicial;
7. Inicializar ganhos de compensação, desligar os controladores e sinais de PWM das juntas.

Executando uma trajectória polinomial desde as coordenadas (25,-25,0) até à posição (-25,45,0) no que respeita ao centro de pressão, iremos efectuar um movimento ao longo dos eixos  $xy$ . O ganho escolhido para o controlador de equilíbrio ( $k_i$ ) é de 50, e o período da trajectória é de 2s. Por questões de análise dos dados, foi feita uma amostragem de 4 segundos tendo em vista o estudo da resposta em regime transiente e estacionário. A Fig. 4.40 apresenta os resultados da experiência.

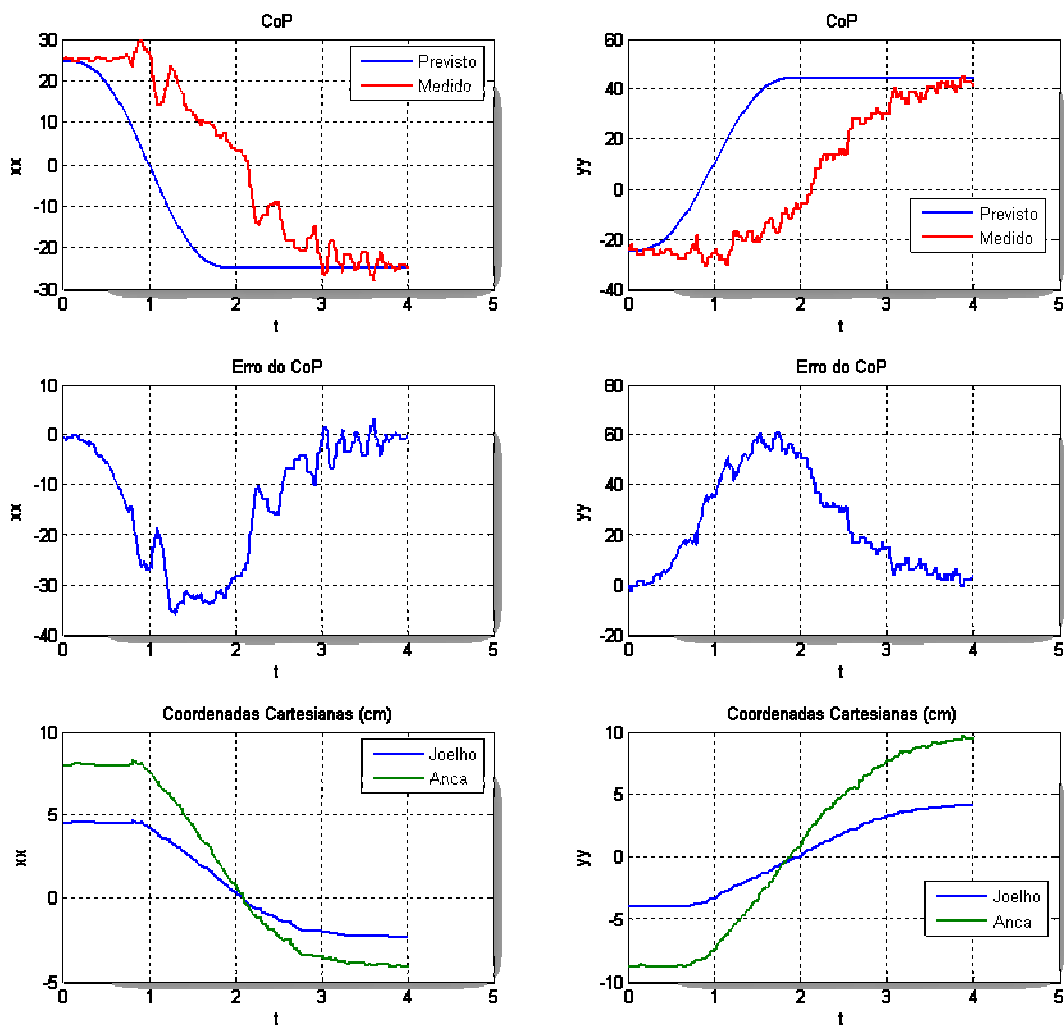


Fig. 4.40: Resposta ao polinómio aplicado nos eixos  $xy$  com  $K=50$ .

Como se pode observar na Fig. 4.40, a resposta ao polinómio, segundo o eixo  $xx$ , apresenta um elevado atraso, com um erro máximo de 35 unidades para  $t=1.25s$ . O tempo de estabelecimento também apresenta um valor elevado, apenas atingindo o valor final para  $t=3s$ . Relativamente ao eixo  $yy$ , obteve-se também um elevado tempo de atraso, com um erro máximo de 60 unidades e um tempo de estabelecimento de 4 segundos.

Tendo em vista a redução do sinal de erro, modificaram-se alguns parâmetros como é o caso do ganho do controlador de equilíbrio e o período da trajectória.

Aumentando o ganho do controlador  $k_i$  de 50 para 70 (Fig. 4.41) verifica-se imediatamente uma diminuição dos tempos de estabelecimentos – 3 segundos para a componente  $xx$  e 3.5 segundos para  $yy$  – e o erro de *tracking* também tende a ficar mais pequeno: 30 unidades para  $xx$ , e 58 unidades para  $yy$ .

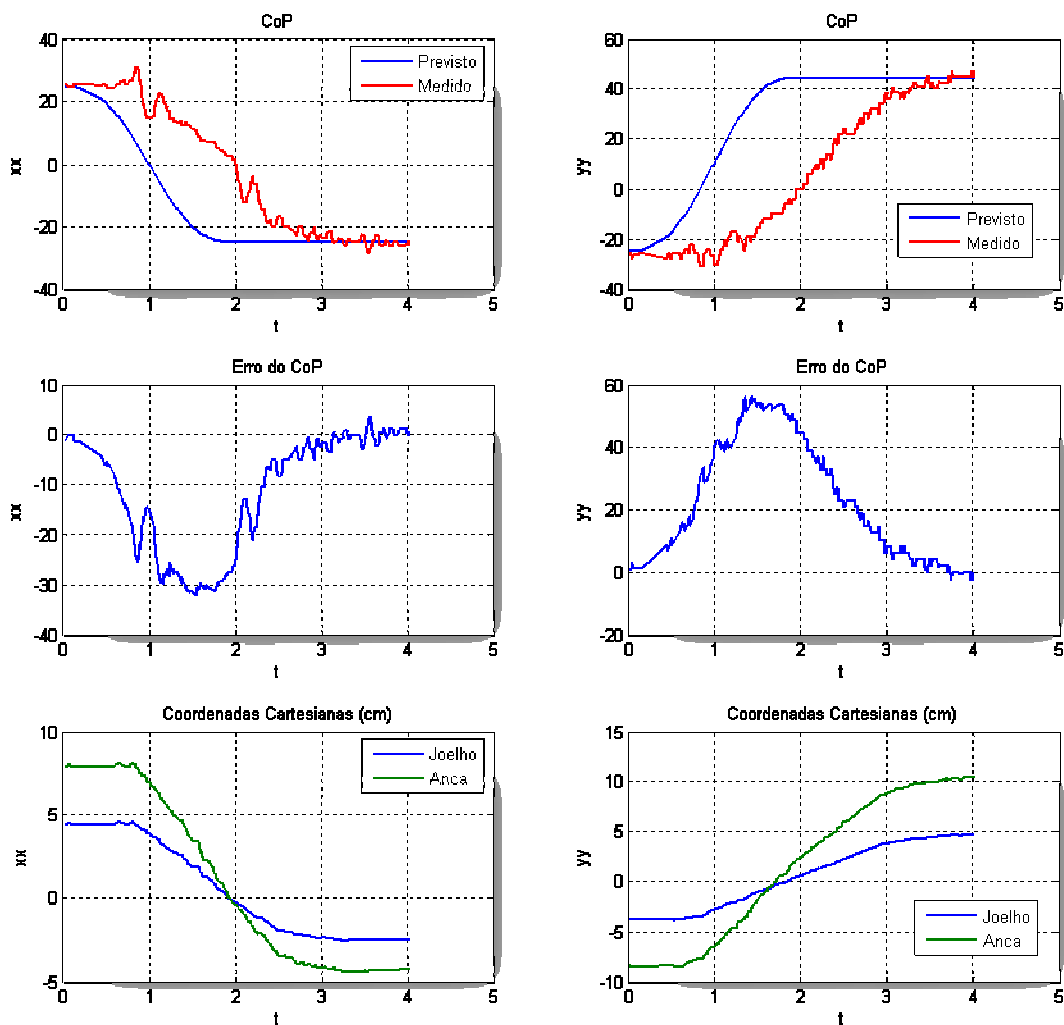


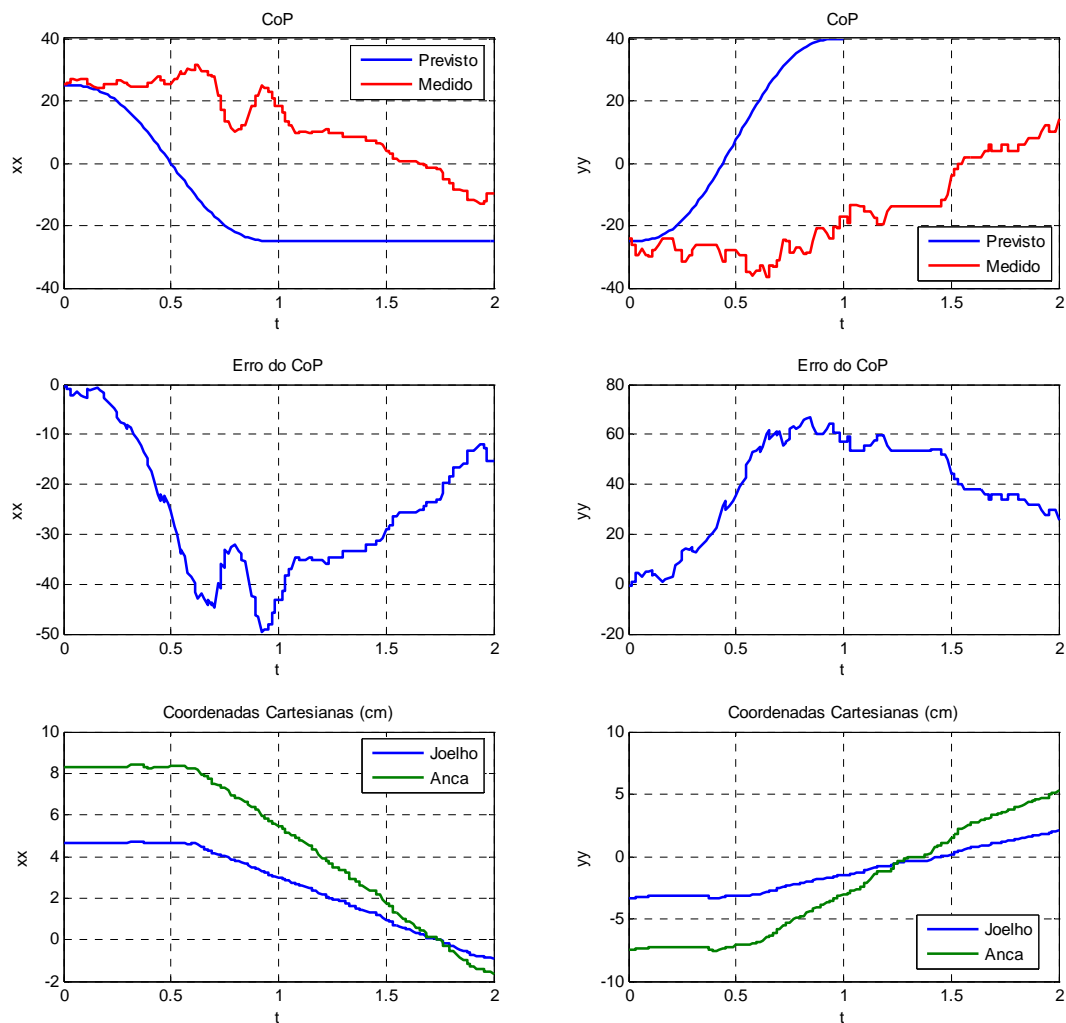
Fig. 4.41: Resposta ao polinómio aplicado nos eixos  $xy$  com  $K=70$ .

Conclui-se assim que o aumento do ganho de compensação do controlador de equilíbrio tende a melhorar o tempo de estabelecimento e o erro de *tracking*. Contudo, existe um limite para o qual o sistema começa a instabilizar observando-se significativas vibrações. Este fenómeno começa a ser observado precisamente para o último ganho testado ( $K=70$ ), pelo que poucas possibilidades se apresentam para melhorar o erro que em si ainda se apresenta um tanto longe para possuir valores satisfatórios.

Se em vez de manipular o ganho de compensação, variar a velocidade da trajectória, ou por outras palavras, o seu período, é possível que o erro possa ser melhorado. A Fig. 4.42 e a Fig. 4.43 exploram esta possibilidade.

A Fig. 4.42 apresenta o caso, em que o período da trajectória foi reduzido para apenas 1s – duplicou-se a velocidade. As consequências traduzem-se numa variação demasiado elevada para o sistema acompanhar, e por isso os tempos de estabelecimento e os erros máximos pioram para valores incontroláveis.





**Fig. 4.42: Resposta ao polinómio aplicado de duração 1s ( $K=50$ ).**

Se, ao invés, aumentarmos o período da trajectória para o dobro – 4 segundos – (Fig. 4.43), as características da resposta melhoram em muito: o tempo de estabelecimento diminui para 4 segundos para as duas componentes (corresponde ao instante final da trajectória), e o erro diminui para 20 e 32 unidades para as componentes  $xx$  e  $yy$  respectivamente.

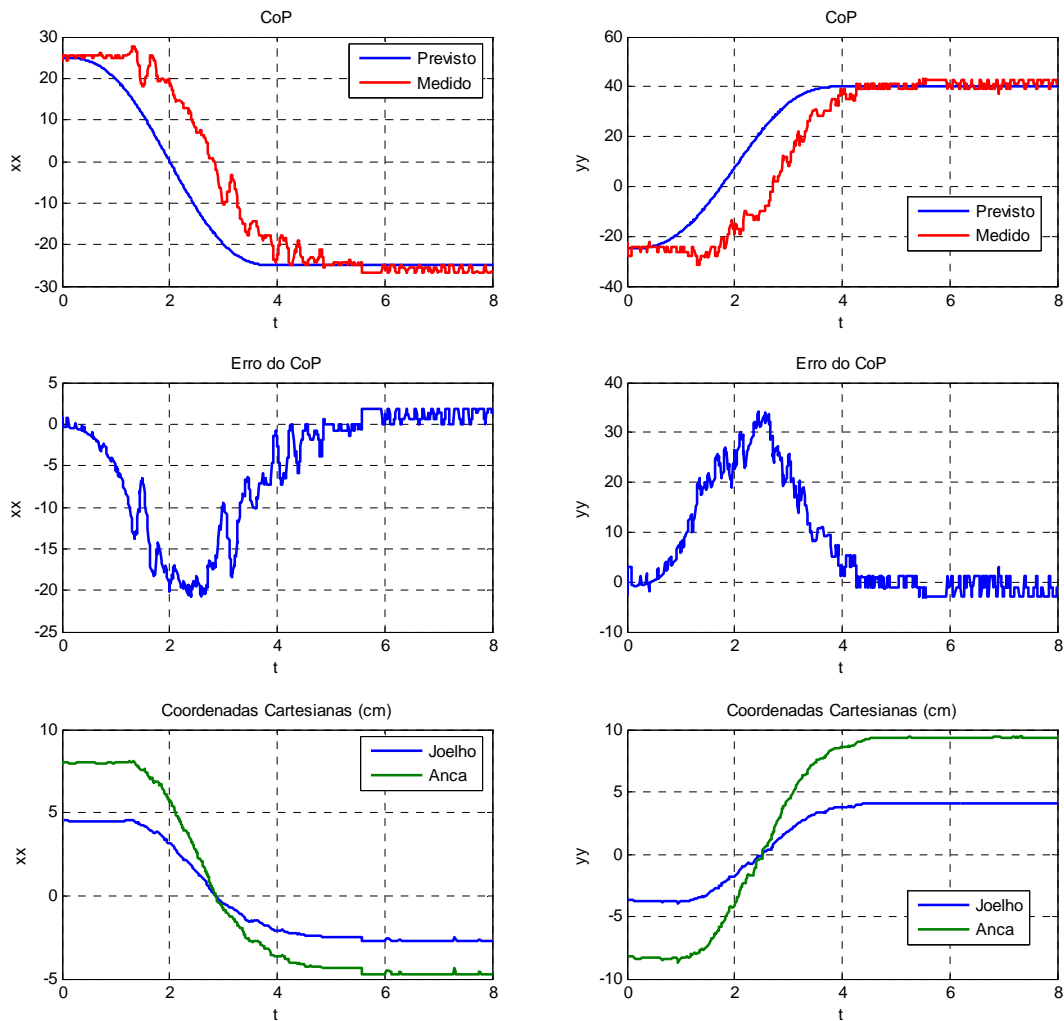


Fig. 4.43: Resposta ao polinómio aplicado de duração 4s ( $K=50$ ).

Como conclusão, pode-se afirmar que a melhor estratégia para otimizar a resposta do sistema é usar ganhos de compensação no controlador de equilíbrio próximos do limite (sem provocar instabilidade) e aplicar velocidades reduzidas. Como se pretende, numa fase inicial, que o robot se desloque a baixa velocidade, tais exigências não deverão constituir um problema para a implementação do controlador.

#### 4.6.4 EXECUÇÃO DE TRAJECTÓRIAS RECTILÍNEAS

Como complemento ao estudo dinâmico da resposta do sistema a um polinómio de quinta ordem, foram executadas experiências similares, mas com a repetição da mesma trajectória várias vezes para análise do comportamento do sinal do erro ao longo do tempo. Cada experiência seguiu o seguinte procedimento:

1. Inicialização do  $\text{CoP}_{\text{ref}}$  para a execução de uma trajectória polinomial de 5ª ordem;
2. Activação do controlador de equilíbrio;
3. Definição do período da trajectória e do ganho do controlador de equilíbrio;
4. Execução da trajectória para um determinado  $\text{CoP}_{\text{ref}}$  final;
5. Esperar que o sistema termine a trajectória;
6. Execução de uma nova trajectória para o  $\text{CoP}_{\text{ref}}$  inicial;
7. Esperar que o sistema termine a trajectória;
8. Repetir os passos 4 a 7 mais nove vezes.
9. No final, inicializar os ganhos de compensação e desligar os controladores, bem como os sinais de PWM.

Ao longo da execução das trajectórias, os sensores de pressão e as posições angulares dos servomotores são registados para um ficheiro para posterior análise. De modo a fazer uma comparação válida com as trajectórias previstas, foram também registados os instantes de tempo inicial e final de cada trajecto, de modo a que, aquando da comparação, os polinómios de quinta ordem previstos sejam mapeados no tempo de forma correcta.

Finalmente, os resultados foram dispostos de forma a analisar o CoP medido e a correspondente cinemática directa do joelho e da anca ao longo do tempo, comparando os valores esperados com os registados.

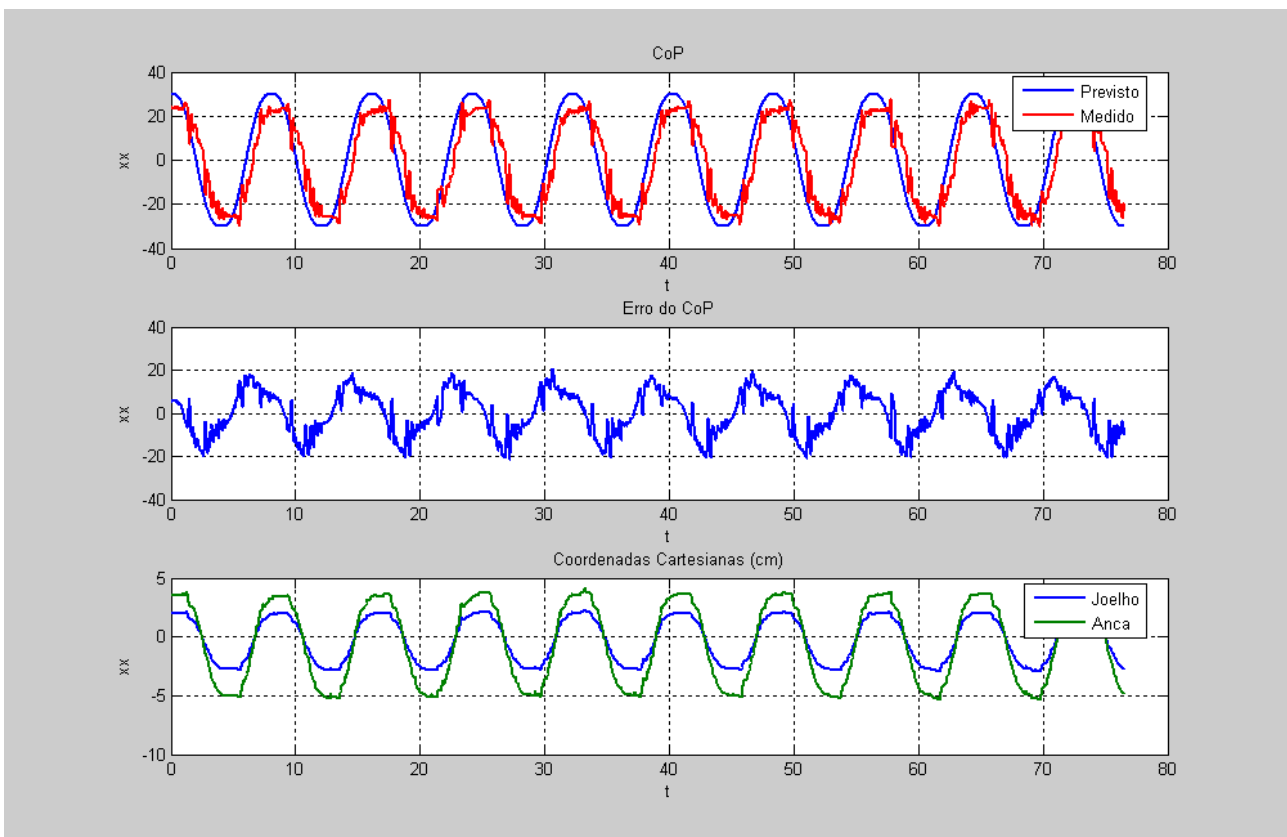
As trajectórias aplicadas obedeceram às seguintes condições de funcionamento:

- Controlador de equilíbrio activado, e com ganho de compensação igual a 50;
- Controlador local PID desactivado;
- Período das trajectórias igual a 2 segundos.

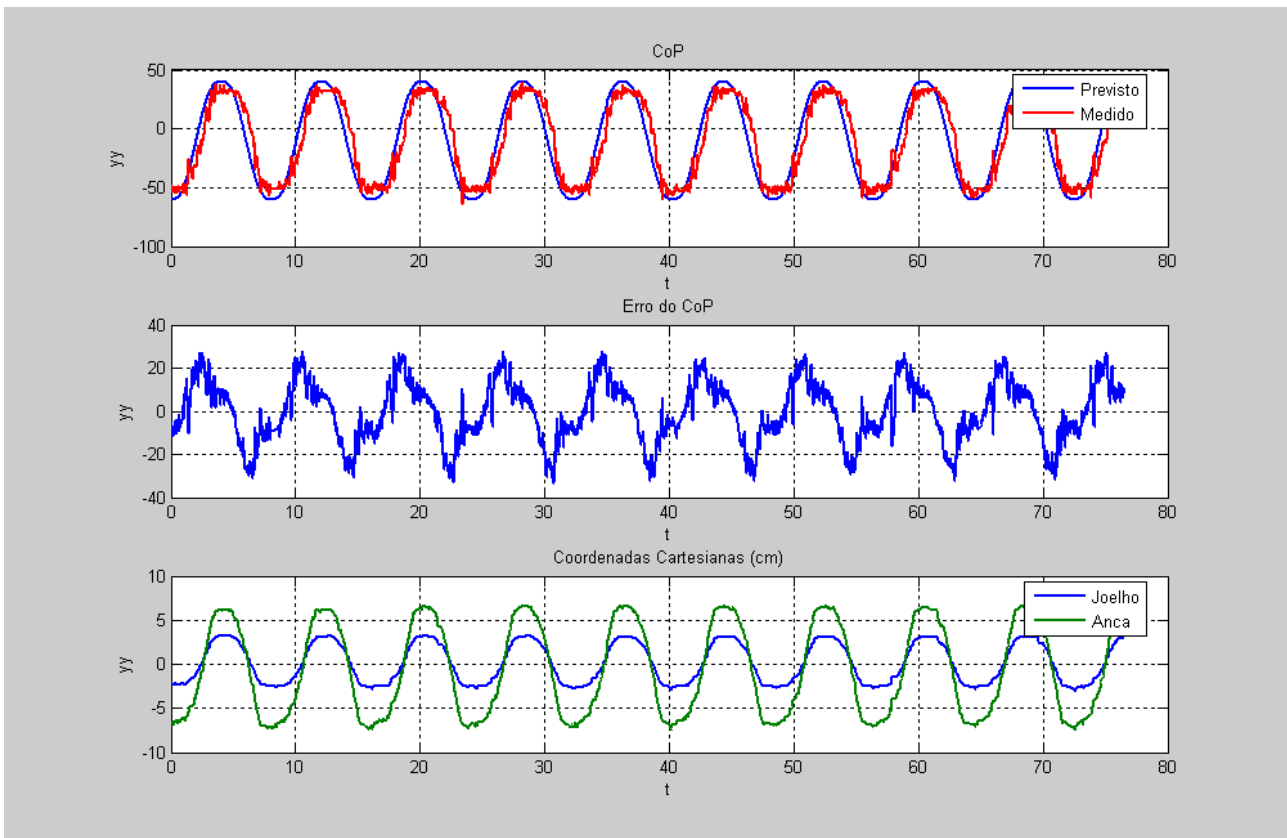
A Fig. 4.44 e a Fig. 4.45 visualizam o resultado da execução de uma trajectória rectilínea ao longo dos eixos  $xy$ , repetida dez vezes – CoP=(30,-60,10) até (-30,40,10).

Pela Fig. 4.44, referente ao percurso pelo eixo  $xx$ , pode-se observar que o sinal de erro mantém-se estabilizado na gama entre +20 e -20 unidades do centro de pressão medido, de acordo se trata do sentido positivo ou negativo da trajectória, respectivamente. A cinemática directa também se apresenta estabilizada, sem a presença de acumulação de erros, contida entre  $\pm 3$ cm para o joelho, e -5 a +4cm para a anca.

Relativamente ao eixo  $yy$  (Fig. 4.45) podemos observar erros de *tracking* na ordem das 30 unidades. Por análise a estes dados, podemos concluir que é perfeitamente possível aplicar trajectórias contínuas, sem a presença de erros acumulativos.



**Fig. 4.44: Trajectória rectilínea ao longo do eixo  $xx$  (Período=2s e K=50).**



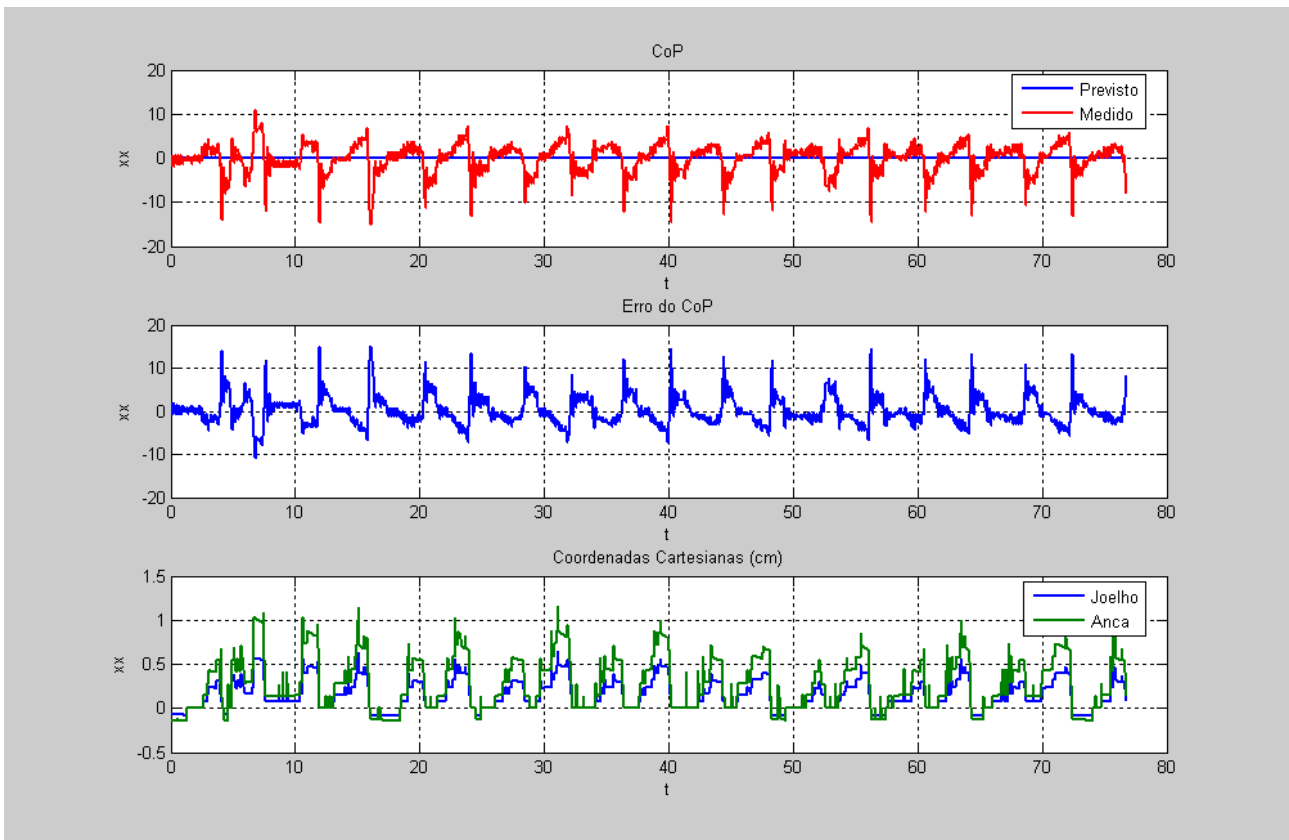
**Fig. 4.45: Trajectória rectilínea ao longo do eixo yy (Período=2s e K=50).**

Realizando a mesma experiência, mas executando trajectórias sobre apenas um dos eixos, podemos fazer algumas inferências, relativamente à interferência que um movimento num determinado eixo pode ter sobre a componente ortogonal (Fig. 4.46 e Fig. 4.47).

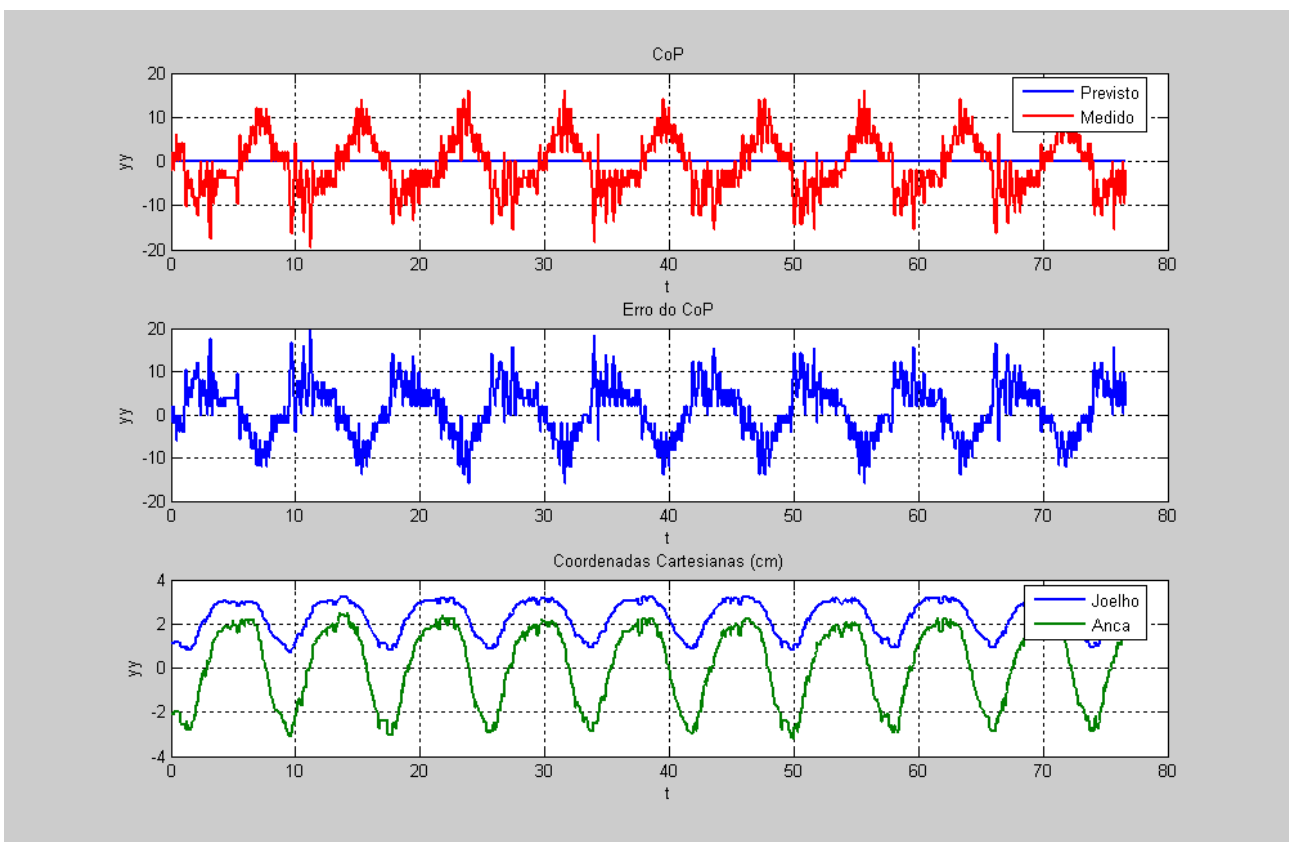
Movendo o sistema apenas segundo o eixo yy, entre as coordenadas  $CoP_{ref}=(0,-60,10)$  a  $(0,40,10)$ , podemos observar flutuações no eixo ortogonal a este que podem atingir as 15 unidades. Contudo em termos de cinemática directa, a perna apenas se desloca no máximo 1 cm, o que representa uma baixa interferência que os movimentos segundo o eixo yy induzem no eixo xx.

Para a trajectória ao longo do eixo xx –  $CoP_{ref}=(-30,0,10)$  até  $(+30,0,10)$  –, também se verificam algumas flutuações no eixo yy, mas contidas entre  $\pm 15$  unidades do centro de pressão. A cinemática directa também apresenta flutuações compreendidas entre 1 e 3cm para o joelho, e entre -3 e +2cm para a anca. Estas flutuações caracterizam-se por uma forte dependência com o movimento no eixo ortogonal xx, e não podem de forma nenhuma ser ignorados.

Como que resumindo, os dados apontados pelos movimentos em cada um dos eixos mostram uma sensibilidade muito maior no eixo yy no que respeita às flutuações sensoriais. Tal pode ser induzido pela introdução dos *offsets* no cálculo do centro de pressão a partir das saídas dos sensores de força (ver secção 4.4.2). Por um lado possuem a vantagem de assemelhar as sensibilidades de cada componente do centro de pressão, mas trazem consigo a desvantagem de introduzir maiores flutuações no sinal de erro de *tracking*. Infelizmente estamos perante um dilema: quanto maior for este *offset*, maior é a reactividade do controlador, sob pena de maiores oscilações do sinal de erro; e quanto menor for, maior é a oposição às flutuações, mas mais lenta é a resposta do controlador. Uma solução de compromisso torna-se necessária. Percursos com ganho de compensação de valor 50, e períodos de trajectória superiores a 2 segundos são recomendados para minimização do sinal de erro.



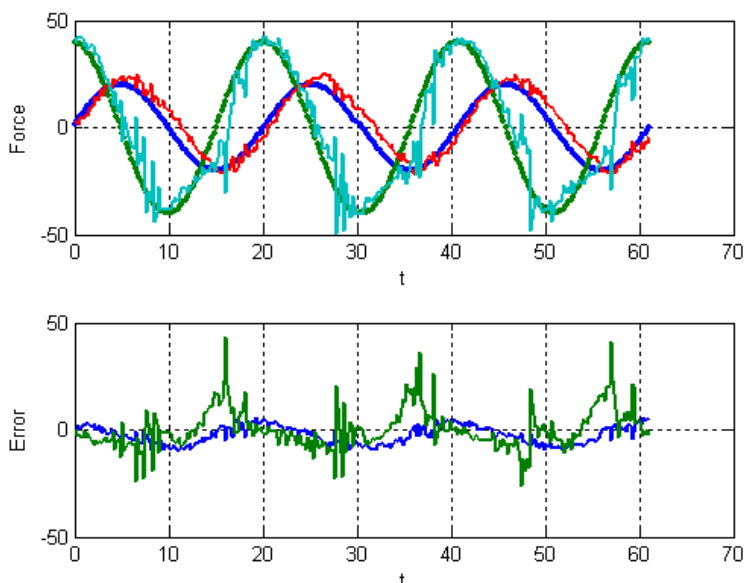
**Fig. 4.46:** Flutuações do CoP e da cinemática directa na componente  $xx$ , na realização de uma trajectória sobre apenas o eixo  $yy$ .



**Fig. 4.47:** Flutuações do CoP e da cinemática directa na componente  $yy$ , na realização de uma trajectória sobre apenas o eixo  $xx$ .

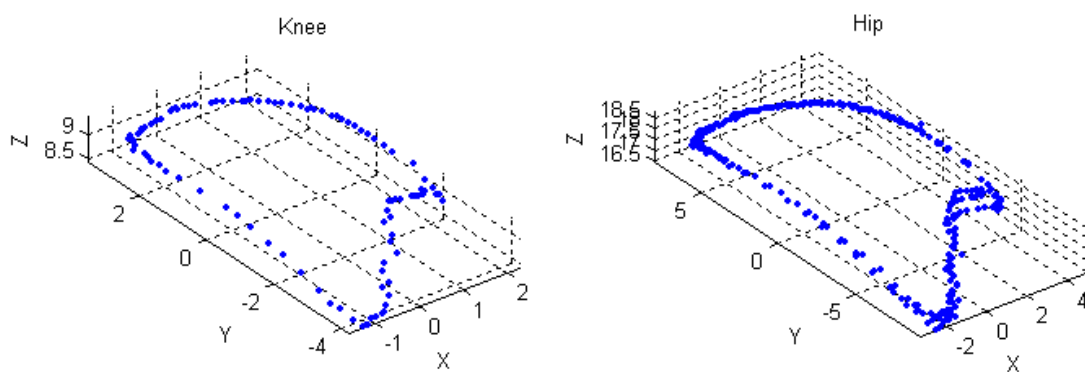
### 4.6.5 EXECUÇÃO DE TRAJECTÓRIAS ELÍPTICAS

Executando trajectórias elípticas, especificando vários pontos de passagem segundo um formato elíptico, e com uma duração temporal entre cada par, podemos visualizar a resposta do sistema na Fig. 4.48. Pode-se confirmar mais uma vez a superior sensibilidade ao ruído por parte da componente  $yy$ .

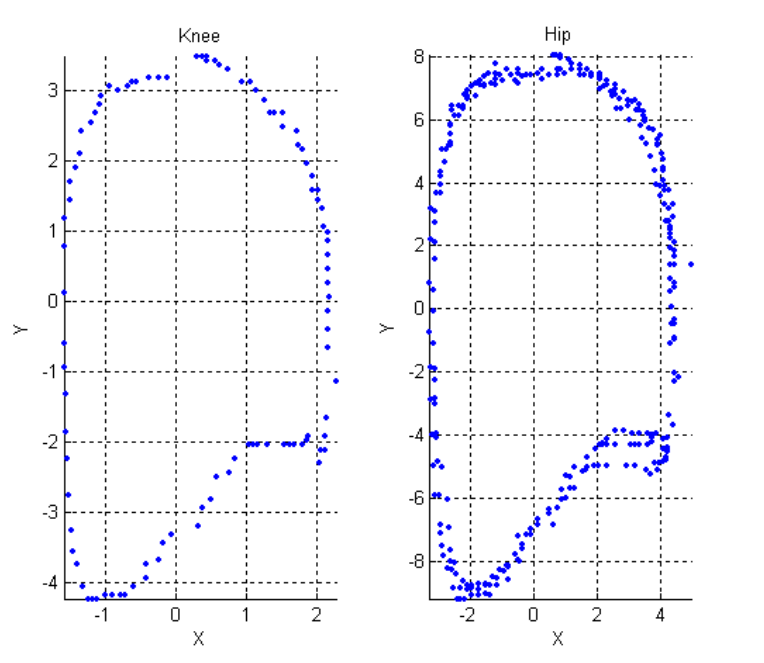


**Fig. 4.48:** Variação do CoP ao longo do percurso elíptico - componentes  $x$  (azul) e  $y$  (verde) - em comparação com o real, e o respectivo erro.

Um pormenor interessante a observar na Fig. 4.48 é o facto do erro na componente  $xx$  ser máximo quando o  $CoP_{ref}$  se encontra nas extremidades do pé (componente  $yy$  nos extremos). Efectuando uma análise espacial, em vez de temporal, podemos ver esta anomalia mais evidentemente na Fig. 4.49 e na Fig. 4.50, segundo a perspectiva 3D e usando o eixo  $zz$  como referência, respectivamente.



**Fig. 4.49:** Cinemática directa do joelho e da anca (vista 3D).



**Fig. 4.50: Cinemática directa do joelho e da anca (vista 2D de cima).**

De facto, a trajectória real do centro de pressão afasta-se da elipse projectada, no canto inferior direito. Tal anomalia pode-se dever a diversos factores, sobretudo físicos, tais como a excessiva sensibilidade do respectivo sensor de força, podendo ser causada pelo próprio sensor, ou pela placa de acrílico subjacente com um nível de rigidez inferior ao normal, o que minimiza a necessidade da estrutura se deslocar para a direcção negativa do eixo  $yy$ .

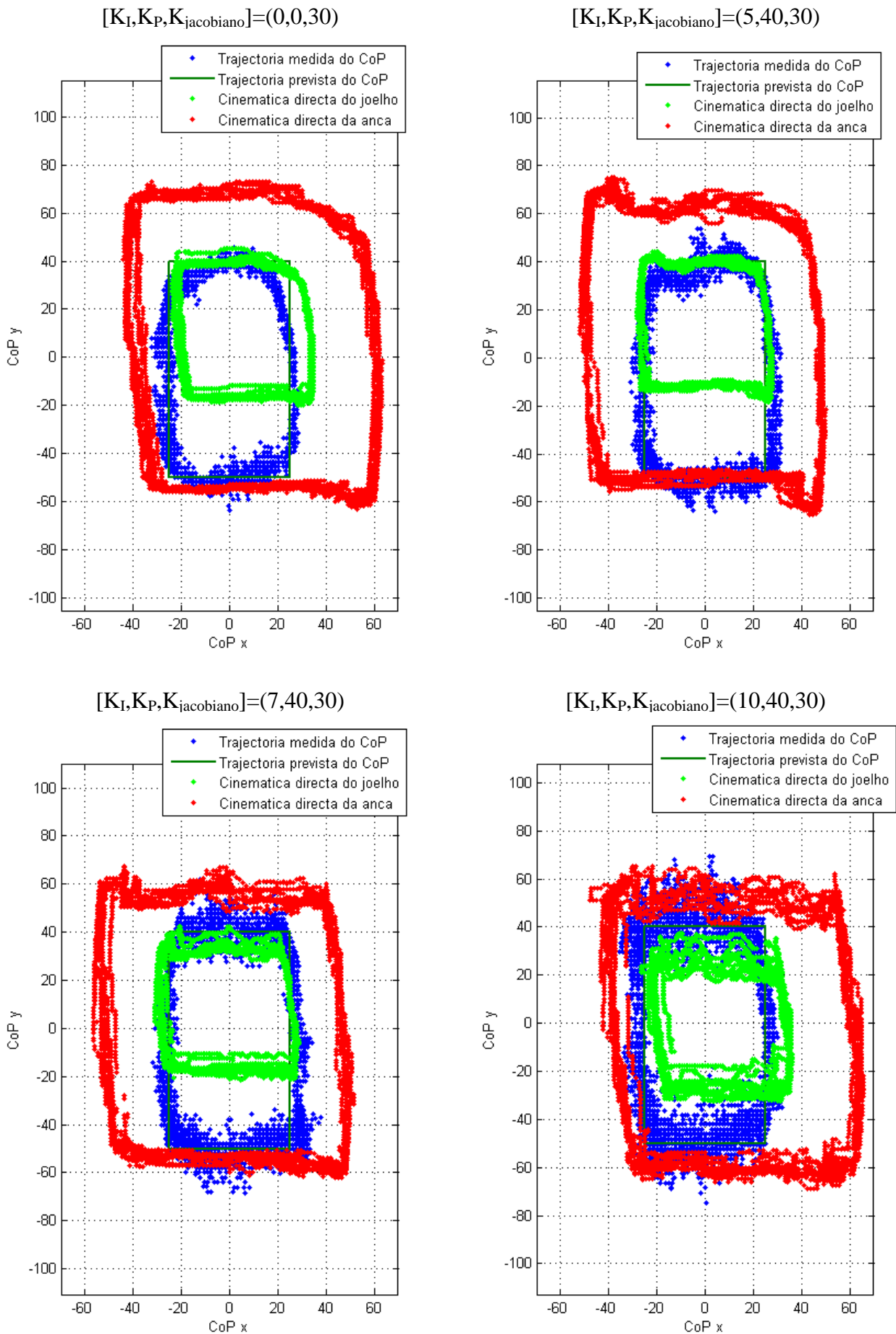
#### 4.6.6 EXECUÇÃO DE TRAJECTÓRIAS RECTANGULARES

Experimentando agora o formato rectangular, definiram-se quatro pontos de referência do centro de pressão correspondentes aos quatro vértices, e aplicaram-se sucessivamente em instantes de tempo periódicos, até completarem 10 voltas. A Fig. 4.51, apresenta a execução de 10 trajectos rectangulares, cujas arestas são compostas por trajectórias polinomiais de quinta ordem com um período de execução de 4 segundos. A visualização é do tipo espacial, podendo observar o trajecto do CoP esperado (a preto), o CoP medido (pontos a azul), e a cinemática directa do joelho (verde) e da anca (vermelho) vista do eixo  $zz$ . Os diferentes gráficos da Fig. 4.51 mostram o efeito do controlador local variando os seus parâmetros de controlo – ganho de integração  $K_I$  e proporcional  $K_P$  – para um ganho fixo do controlador de equilíbrio ( $K_{\text{jacobiano}}=30$ ).

Começando pelo controlo em malha aberta (sem controlo local –  $K_I$  e  $K_P$  nulos), podemos observar o seguimento do trajecto proposto com alguma exactidão, mas ao nível dos vértices a dificuldade de seguimento aumenta conferindo ao trajecto medido um formato semelhante ao elíptico. Ligando o controlo local e aumentando os seus parâmetros, observa-se um seguimento mais exacto, mas ao mesmo tempo uma densidade de pontos (de CoP) muito mais intensa em torno do trajecto aplicado. Tal só pode evidenciar a presença de oscilações resultante da instabilidade do controlador: quanto mais elevados os parâmetros de controlo, maior a instabilidade.

Efectuando os mesmos trajectos, mas agora com um aumentado ganho no controlador de equilíbrio ( $K_{\text{jacobiano}}=50$ ) (ver Fig. 4.52), pode-se observar as mesmas características evidenciadas, mas com o surgimento mais precoce das oscilações. Tal é consistente com o esperado, pois o aumento do ganho do controlador de equilíbrio, tende a aumentar a instabilidade.

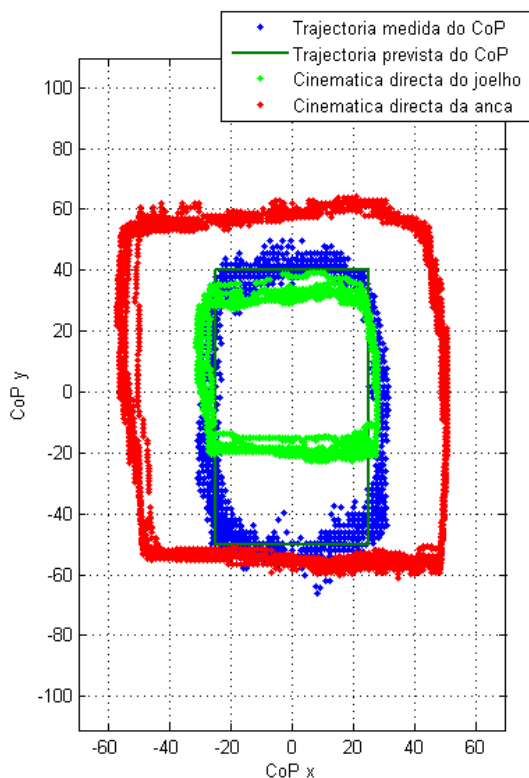
A Fig. 4.54 propõe variar a velocidade para um conjunto fixo de parâmetros de compensação. Para  $[K_I, K_P, K_{\text{jacobiano}}]=(5,40,30)$ , observa-se que quanto maior é a velocidade, mais instável é o seguimento de trajectória. Por observação o período de 4 segundos aparenta ser o melhor para a execução de trajectórias rectangulares.



**Fig. 4.51: Resposta a 10 voltas de uma trajectória rectangular para vários parâmetros do controlador local.**



Sem Controlo Local:  $[K_I, K_P, K_{\text{jacobiano}}] = (0, 0, 50)$



Com Controlo Local:  $[K_I, K_P, K_{\text{jacobiano}}] = (5, 40, 50)$

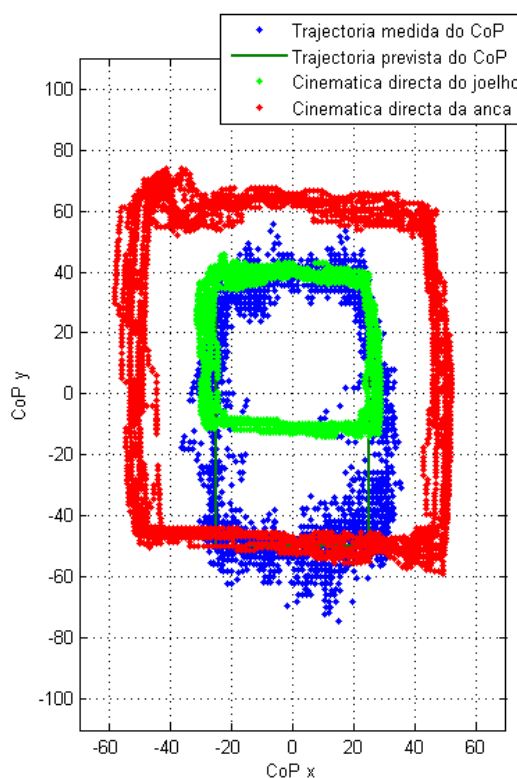
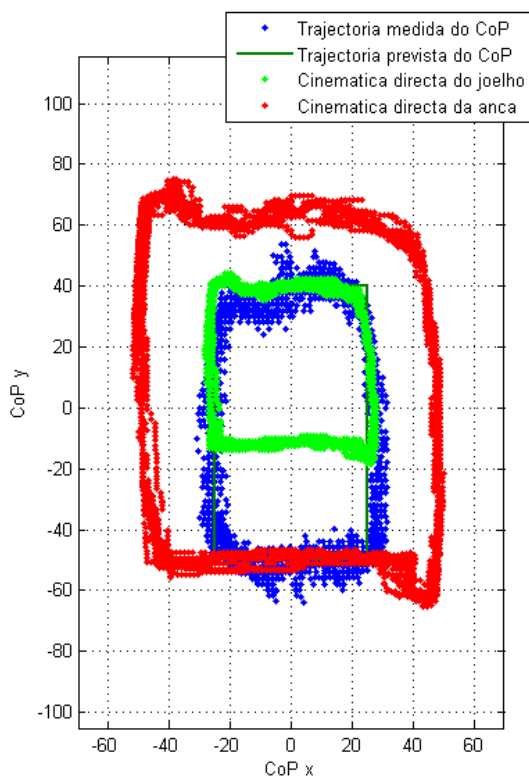


Fig. 4.52: Resposta a 10 volts de uma trajectória rectangular sem e com controlo local ( $K_{\text{jacobiano}}=50$ ).

Período=4s



Período=2s

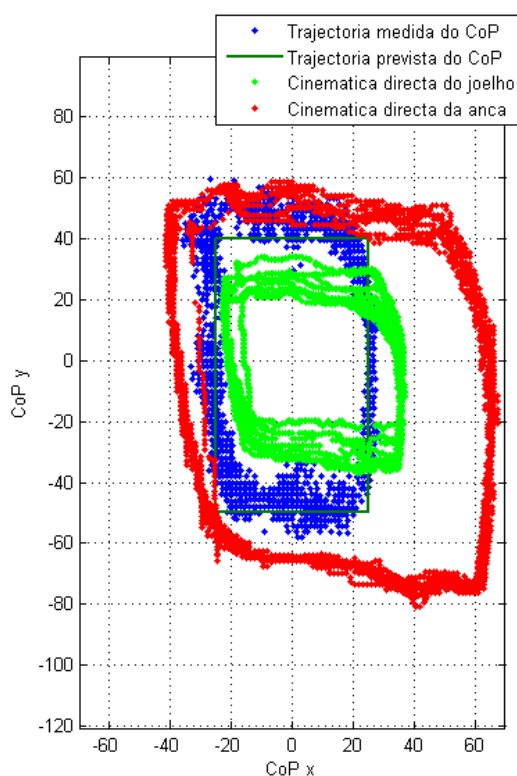
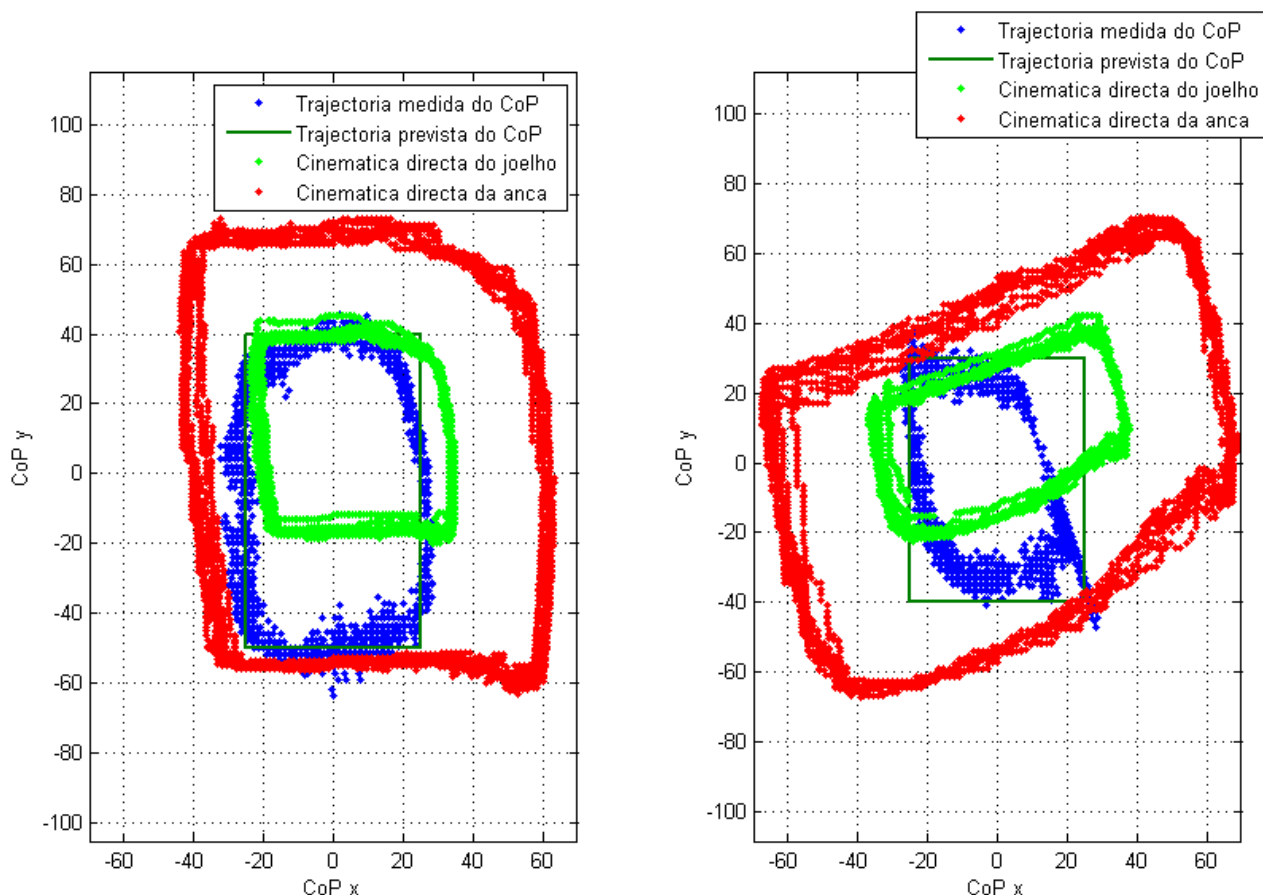


Fig. 4.53: Resposta a 10 volts de uma trajectória rectangular para diferentes velocidades -  $[K_I, K_P, K_{\text{jacobiano}}] = (5, 40, 30)$ .

Apesar dos resultados animadores observados nas figuras anteriores, mostrando bons seguimentos de trajectória, desde que os parâmetros de compensação e a velocidade estejam devidamente ajustados, mesmo assim os resultados estão excessivamente dependentes das condições físicas do sistema, mais em concreto dos elementos adjacentes aos sensores de força. Diferenças na sensibilidade destes sensores, rigidez desigual nas placas de acrílico ligadas a estes, e a necessidade constante de ajuste dos potenciômetros que controlam o nível de tensão à saída dos sensores, resultam em significativas diferenças no cumprimento das trajectórias propostas, além que os valores solicitados ao controlador de equilíbrio respeitante a uma determinada localização física estão continuamente em mudança.

A Fig. 4.54 ilustra o problema descrito para duas experiências executadas em instantes de tempo diferentes, mantendo iguais os ganhos de compensação, coordenadas referência do CoP e período de execução.



**Fig. 4.54: Duas experiências realizadas em diferentes momentos, com iguais parâmetros de controlo.**

Esta variabilidade de comportamento dificulta em grande parte a construção de um algoritmo que permita fazer a correspondência da localização espacial do centro de gravidade com as coordenadas do centro de pressão calculadas. Por estas razões, ainda não é possível indicar uma lei de correspondência entre estas duas grandezas, o que possibilitaria, um dia mais tarde, indicar as coordenadas espaciais desejadas da anca, e o *firmware* presente nos microcontroladores calcularia o centro de pressão referência a aplicar ao respectivo controlador.

## 4.7 DISCUSSÃO DOS RESULTADOS

Dois controladores de equilíbrio foram propostos neste capítulo, nomeadamente o proporcional que determina a velocidade das juntas ortogonais do pé com base no centro de pressão medido a partir dos sensores de força, através de uma constante, e o Jacobiano, que faz uso de uma matriz que relaciona o erro

do centro de pressão com a variação angular a aplicar às duas juntas ortogonais do pé mais a do joelho. Este último compensador fornece a vantagem de controlar também a junta do joelho, tornando o controlo mais eficaz, mas contudo, depende das características físicas da estrutura, como a massa e comprimento dos elos. Utilizando estes dois algoritmos realizaram-se vários testes perante a situação de variação de declive do plano de suporte e a de presença de interferências externas à estrutura, revelando a satisfatória eficiência destes controladores, especialmente a do Jacobiano em que o controlo da junta do joelho representa uma mais valia para a sua eficiência.

Vários estudos no que concerne à execução de trajectórias por parte da perna, também foram realizados, pela manipulação do CoP referência do controlador de equilíbrio. Três estudos principais foram feitos:

- Análise estática do controlador de equilíbrio;
- Análise dinâmica do controlador;
- Execução de multi-trajectórias e análise do desempenho.

A análise estática conseguiu demonstrar o correcto funcionamento do controlador de equilíbrio com os cumprimentos das diversas posições solicitadas. A análise dinâmica demonstrou a relação entre os parâmetros de compensação do controlador de equilíbrio e do período com a resposta, no que respeita ao erro de seguimento de trajectória e tempo de estabelecimento. Concluiu-se assim que aumentando o ganho de compensação, o erro tende a diminuir bem como o tempo de estabelecimento, mas desde que dentro de certos limites de modo a evitar instabilidade. De igual forma o período da trajectória também contribui para a diminuição do erro, recomendando-se, por isso, a aplicação de trajectórias com um período superior ou igual a quatro segundos para a minimização do sinal de erro.

Em seguida realizaram-se vários tipos de trajectórias multi-volta, nomeadamente rectilíneas, elípticas e rectangulares. As trajectórias rectilíneas permitiram observar a interferência que o movimento em cada eixo introduz no eixo perpendicular, mais concretamente, oscilações de carácter não desprezável e relacionáveis com o movimento aplicado no primeiro eixo, concluindo, desta forma, que as trajectórias aplicadas em cada eixo não são independentes entre si. Verificou-se a susceptibilidade ao ruído que a componente *yy* apresenta, devido ao atribuído aumento de sensibilidade no cálculo do centro de pressão no mesmo eixo (*offset* subtraído à saída dos sensores de força segundo a secção 4.4.2).

Finalmente com as trajectórias elípticas e rectangulares, pudemos conhecer a limitação que os factores físicos inerentes à estrutura dos pés introduzem no cumprimento das trajectórias solicitadas. Verificou-se que quando a trajectória aplicada passa sobre um dos sensores, por vezes, o controlador é incapaz de garantir o cumprimento do CoP solicitado. O motivo deste comportamento não foi completamente esclarecido, mas discutem-se algumas das causas prováveis:

1. As assimetrias de conexões entre o pé e a perna da estrutura humanóide podem interferir de alguma forma, ao fazer com que o centro de pressão não possa cobrir toda a área do pé. Lembra-se que os dois eixos que ligam estas duas partes não estão centrados sobre o pé, pelo que a trajectória realizada pode apresentar anomalias à medida que o centro de gravidade viaja ao longo do plano do pé.
2. As assimetrias de conexão entre o pé e o tornozelo, também atribuem diferenças de sensibilidade a cada sensor, fazendo com que para certas extremidades, onde os sensores são mais sensíveis, a estrutura da perna não necessita de se deslocar tanto.
3. Também é possível que a pré-carga aplicada sobre um sensor coloque-o perto da zona de saturação (ver Fig. 4.2), fazendo com que atinja facilmente o valor óhmico máximo, impedindo que o centro de pressão possa ser medido com rigor. Tal é perfeitamente possível de provocar se os parafusos de fixação das duas plataformas do pé estiverem excessivamente apertados. Além disso, existem pontos de fixação destas plataformas onde os parafusos enroscam directamente sobre a plataforma superior, limitando em grande parte a excursão de deformação das placas. Por isso, recomenda-se a utilização do sistema parafuso+*porca* para evitar este problema.

Além destes problemas, também se observam respostas variáveis com o tempo causados com tendência a deteriorar-se ao longo do tempo, necessitando de reajustes no ganho do controlador, bem como na deformação inicial dos sensores. As causas podem ser identificadas como as seguintes:

- Postura da perna no momento da calibração inicial dos sensores de força: como é posicionada manualmente (por intervenção humana, nunca se garante duas posturas exactamente iguais, pelo que ocorrem sempre diferenças no centro de gravidade inicial). Mesmo efectuando o posicionamento através dos servomotores, os esforços envolvidos em cada experiência fazem com que a correia de transmissão sofra fadiga, ou o próprio eixo do servo adquira desfasamentos, pelo que este problema ocorrerá sempre;
- Assimetria ao nível da electrónica de instrumentação: note que para cada extensómetro, um circuito de acondicionamento é utilizado, pelo que se aplicarem componentes de baixa qualidade, podem-se registar variações no seu comportamento à medida que a temperatura vai aumentando. Além disso, se existirem ligações não suficientemente seguras no PCB, o movimento da perna, pode provocar oscilações nos sinais eléctricos;
- A baixa qualidade dos potenciómetros multi-volta usados na electrónica de instrumentação para ajuste dos *offsets* na tensão de saída, requer o seu reajuste constante o que constitui um sério problema tendo em vista a necessidade de autonomia do robot;
- Modificação accidental da deformação inicial das placas de acrílico, motivado por alguns acidentes de trabalho que forcem o deslocamento dos parafusos de fixação;
- É possível também que a relação deformação/força aplicada varie ao longo do tempo. Várias observações registaram diferenças de comportamento por parte dos sensores após várias horas de funcionamento, mas no dia seguinte, as condições iniciais voltavam a ser repostas. Note que isto acontecia quando movimentos intensivos sobre o pé eram executados, pelo que pode ocorrer alguma fadiga por parte da estrutura molecular do acrílico. Infelizmente esta suposição não pode ser verificada, uma vez que a realização de trajectórias durante um período de amostragem suficientemente extenso implica a presença constante de um observador dada a instabilidade inerente ao sistema.

Tendo em conta estes problemas, apontam-se algumas soluções/recomendações a levar em consideração nos desenvolvimentos futuros:

- Adopção de extensómetros de elevada qualidade, com um baixo coeficiente de temperatura. O mesmo também se aplica às resistências e potenciómetros presentes na ponte de *Wheatstone*;
- Uso de PCB's de qualidade com ligações bastante seguras para garantir a estabilidade dos sinais eléctricos face a vibrações;
- Utilização de acrílico ou outro material com características bastante homogéneas, e resistente à fadiga, de modo a minimizar a variação do seu comportamento ao longo do tempo;
- Utilização de montagens parafuso+porca, em todos os pontos de fixação das plataformas do pé, para homogeneização da resposta sensorial;
- Minimização das assimetrias na ligação pé/perna: procurar que os eixos de ligação entre estas duas partes se aproximem do centro do pé, de modo a conferir ao centro de pressão medido, uma relação bastante consistente com o centro de gravidade.

## Capítulo 5

# Conclusões

Este trabalho permitiu aprofundar os conhecimentos práticos sobre barramentos de comunicação, mais concretamente a comunicação CAN muito usada em ambientes industriais, tendo-se desenvolvido vários protocolos de forma a conferir a máxima robustez e maximização da largura de banda disponível. Relativamente ao último protocolo implementado, os resultados observados são bastante animadores, sendo possível observar um funcionamento contínuo sem quaisquer falhas aparentes nas comunicações. Estas afirmações são baseadas em testes exaustivos que foram realizados durante períodos de funcionamento contínuos que nalguns casos atingiam dez horas, pelo que é possível confirmar a elevada robustez nas comunicações.

Apenas algumas correcções são apontadas ao nível do hardware para minimização dos efeitos colaterais na comunicação CAN quando o módulo RS-232 da unidade *Master* é removido. Sugere-se por isso a adição de uma resistência na entrada série RX, de forma a evitar a ocupação total da largura de banda de CPU pelo módulo de comunicação RS-232, que sendo de alta prioridade, impediria o normal funcionamento do módulo CAN.

Relativamente ao sistema de controlo deu-se ênfase à implementação do sistema de controlo de baixo-nível e à avaliação de leis de controlo que usam a informação da força de interacção entre o pé e o solo. Por um lado, o controlo local nas juntas apresenta resultados bastante satisfatórios, permitindo o controlo de posição e velocidade. Por outro lado, no que respeita aos sensores de força, foram implementados controladores que asseguram o equilíbrio de uma perna do robot quando colocada num plano de inclinação variável. Para além disso, foi avaliada a possibilidade de utilização deste controlador na realização de movimentos pela variação das referências. Apesar dos resultados serem promissores, observam-se problemas de calibrações por parte dos sensores de força, bem como uma grande sensibilidade a interferências externas. Várias causas foram apontadas em detalhe bem como possíveis soluções para melhoria do funcionamento do controlador de equilíbrio, pelo que se espera que, uma vez seguidas, melhorem o desempenho e a estabilidade dos algoritmos de controlo implementados.

Apesar de algumas dificuldades no que respeita ao equilíbrio, todo o *firmware* desenvolvido para o controlo local já se encontra numa fase suficientemente madura para permitir, a curto prazo, a locomoção e a realização de outras tarefas pela unidade central de processamento de modo a podermos participar na competição mundial de robots humanóides RoboCup.

Nesta linha de ideias, considero que os principais objectivos propostos para a dissertação foram alcançados com sucesso, tendo sido analisadas estratégias e apontados novos caminhos que permitirão a maturidade deste projecto e a iminente concretização do objectivo último: a criação de uma estrutura humanóide de carácter pedagógico capaz de gerar várias sequências de locomoção de uma forma autónoma e recorrendo unicamente a materiais e componentes de baixo custo.



# Referências

- [1] Y. Sakagami, *et al.*, “The Intelligent ASIMO: System Overview and Integration”, in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, September 30 - October 5, pp. 2478-2483, 2002.
- [2] K. Nagasaka, , *et al.*, “Integrated Motion Control for Walking, Jumping and Running on a Small Bipedal Entertainment Robot”, in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, USA, April 26 - May 1, pp. 3189-3194, 2004.
- [3] F. Yamasaki, T. Matsui, T. Miyashita, H. Kitano, “PINO - the Humanoid that Walk”, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Boston, Massachusetts, USA, September 7-8, 2000.
- [4] T. Furuta, T. Tawara, Y. Okumura, M. Shimizu, K. Tomiyama, “Design and Construction of a Series of Compact Humanoid Robots and Development of Biped Walk Control Strategies”, *Robotics and Autonomous Systems* **37**, pp. 81-100, 2001.
- [5] J-H. Kim, *et al.*, “Humanoid Robot HanSaRam: Recent Progress and Developments”, *Journal of Computational Intelligence* **8(1)**, pp. 45-55, 2004.
- [6] L. Hu, C. Zhou, “Locomotion planning of humanoid robot Robo-Erectus Senior (RESr-1)”, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, USA, Nov 29-Dez 01, 2007.
- [7] S. Behnke, J. Stückler, H. Strasdat and M. Schreiber, “Hierarchical Reactive Control for Soccer Playing Humanoid Robots”, book chapter in the publication *Humanoid Robots: Human-like Machines*, [ISBN: 978-3-902613-07-3], edited by Matthias Hackel, pp. 625-642, 2007.
- [8] Luís Gomes, Mauro Silva, “Concepção e Desenvolvimento da Unidade de Percepção e Controlo para um Robot Humanóide”, Relatório final de projecto 2004/05 – Departamento de Engenharia Mecânica.
- [9] Nuno Beça, Ângelo Cardoso, “Desenvolvimento e Integração das Sub-estruturas Inferior e Superior para a Locomoção de uma Plataforma Humanóide”, Relatório final de projecto 2004/05 – Departamento de Engenharia Mecânica.
- [10] Vítor M. F. Santos, Filipe M. T. Silva, “Engineering Solutions to Built an Inexpensive Humanoid Robot Based on a Distributed Control Architecture”, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Humanoids2005, 5-7 Dezembro, Tsukuba, Japão, 2005.
- [11] Vítor M. F. Santos, Filipe M. T. Silva, “Development of a Low-Cost Humanoid Robot: Components and Technological Solutions”, *Proceedings of the 8<sup>th</sup> International Conference on Climbing and Walking Robots*, CLAWAR’2005, 13-15 Setembro, Londres, UK, 2005.
- [12] M. Albero, F. Blanes, G. Benet, P. Pérez, J. Simó, J. Coronel, “Distributed Real Time Architecture for Small Biped Robot YABIRO”, in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, June 27-30, 2005.
- [13] Y.-J. Cho, *et al.*, “A Compact/Open Network-Based Controller Incorporating Modular Software Architecture for a Humanoid Robot”, *Journal of Intelligent and Robotic Systems* **25**, 341-355, 1999.
- [14] N. Tomokuni, M. Saiga, T. Yabuta, “Distributed Controller for Cooperative Joint Torque Control of Compact Humanoid Robots”, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, December 5-7, pp. 259-264, 2005.
- [15] Karl Johan Åström, *Control System Design*, 2002.
- [16] The PID Algorithm: [http://members.aol.com/pidcontrol/pid\\_algorithm.html](http://members.aol.com/pidcontrol/pid_algorithm.html), 2006.
- [17] Youbin Peng, Damir Vrancic, and Raymond Hanus, “Anti-Windup, Bumpless, and Conditioned Transfer Techniques for PID Controllers”, *IEEE Control Systems*, 1996.
- [18] Michael Hardt and Oskar von Stryk, “The Role of Motion Dynamics in the Design, Control and

Stability of Bipedal and Quadrupedal Robots”, *RoboCup 2002 International Symposium*, June 24-25, Fukuoka, Japan, 2002.

- [19] Dusko Katić and Miomir Vukobratović, “Survey of Intelligent Control Techniques for Humanoid Robots”, *Journal of Intelligent and Robotic Systems*, 37: 117–141, 2003.
- [20] T.Sugihara, Y.Nakamura, “Whole-body Cooperative Balancing of Humanoid Robot using COG Jacobian”, *Proceedings of the 2002 IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland, 2002.
- [21] Milton Ruas, Filipe M. T. Silva, Vítor M. F. Santos, “Parameter Measurement for Speed and Torque Control of RC Servomotors on a Small-Size Humanoid Robot”, *nas Actas do Encontro Científico Robótica 2006, Festival Nacional de Robótica*, pp. 71-79, 28 Abril, Guimarães, Portugal, 2006.
- [22] Milton Ruas, Filipe M. T. Silva, Vítor M. F. Santos, “Techniques for Velocity and Torque Control of RC Servomotors for a Humanoid Robot”, *Proceedings of the 9<sup>th</sup> International Conference on Climbing and Walking Robots*, CLAWAR’06, 12-14 Setembro, Brussels, Belgium, 2006.
- [23] Milton Ruas, Filipe M. T. Silva, Vítor M. F. Santos, "Development of Control Algorithms for a Humanoid Robot", *Revista do Departamento de Electrónica Telecomunicações e Informática da Universidade de Aveiro*, PT: Universidade de Aveiro, ISSN: 1645-04, vol. 4, n.º 8, pp. 946 - 959, 2007.
- [24] Milton R. Silva, Pedro M. Ferreira, Filipe M. Silva, Vítor M. Santos, “Local-level Control of a Humanoid Robot Prototype with Force-Driven Balance“, *Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots*, HUMANOIDS’07, Nov. 29 - Dec. 1, Pittsburgh, Pennsylvania, USA, 2007.
- [25] Milton R. Silva, “Desenvolvimento de Algoritmos de Controlo para Locomoção de um Robot Humanóide”, Relatório final de projecto 2005/06 – Departamento de Electrónica, Telecomunicações e Informática.
- [26] Milton R. Silva, “Desenvolvimento de Algoritmos de Controlo para Locomoção de um Robot Humanóide”, Manual do Programador – projecto 2005/06 – Departamento de Electrónica, Telecomunicações e Informática.
- [27] Milton R. Silva, “Desenvolvimento de Algoritmos de Controlo para Locomoção de um Robot Humanóide”, Manual do Utilizador – projecto 2005/06 – Departamento de Electrónica, Telecomunicações e Informática.