



Pedro Miguel de Sá de Figueiredo Ferreira **Ajudas Computacionais para Aplicação a um Robô Humanóide**



Pedro Miguel de Sá de Figueiredo Ferreira **Ajudas Computacionais para Aplicação a um Robô Humanóide**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e de Telecomunicações, realizada sob a orientação científica do Dr. Filipe Silva, Professor auxiliar convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Dr. Vítor Santos, Professor associado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Uma nota particular de agradecimento, dirigida ao Departamento de Engenharia Mecânica, em especial às pessoas do Laboratório de Automação e Robótica pela ajuda, apoio e disponibilidade demonstrada ao longo de todo este trabalho.

o júri

presidente

Doutor José Fernando da Rocha Pereira
Professor Associado da Universidade de Aveiro

vogais

Doutor Urbano José Carreira Nunes
Professor Associado da Universidade de Coimbra

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

palavras-chave

robótica humanóide, locomoção bípede, cinemática inversa, interface gráfica.

resumo

Este trabalho insere-se no projecto robô humanóide da Universidade de Aveiro. É a evolução de uma ideia iniciada no projecto final de curso sobre a criação de uma interface gráfica para simular/operar o robô. Parte do pressuposto de uma base mecânica e electrónica sólida e com o intuito de desenvolver um ambiente integrado de programação, monitorização e simulação do robô.

O trabalho reúne um estudo de algumas das várias áreas ligadas à temática da robótica humanóide, onde se abordam questões de modelação humanóide, algoritmos de cinemática inversa, critérios de estabilidade, padrões de locomoção, entre outros. Os resultados desse estudo são depois aplicados na criação de um simulador cinemático integrado na interface gráfica implementada em Matlab.

As funcionalidades alcançadas na interface gráfica permitem a simulação, actuação e monitorização do robô humanóide. Inclui tarefas simples como andar e rodar para adição rápida na criação de movimentos mais complexos. Além da animação da estrutura com 22 graus de liberdade, é possível visualizar as trajectórias das juntas, do centro de massa e dos binários ao longo do movimento simulado. A interface oferece ainda a possibilidade de comparação entre a trajectória simulada e a obtida através de leitura sensorial do robô.

keywords

humanoid robot, bipedal locomotion, inverse kinematics, graphical user interface.

abstract

This work is part of the humanoid robot project of the University of Aveiro. It is the evolution of an idea initiated in the graduation project on the creation of a graphical user interface to simulate/command the robot. It begins with the assumption of a solid mechanical and electronic foundation aiming to develop an integrated programming, monitoring and simulation environment.

It brings together a study of some of the many topics related with humanoid robotics, including the humanoid modeling and simulation, inverse kinematics algorithms, criteria of stability, locomotion patterns, among others. The results are then applied to create a kinematic simulator integrated in a graphical user interface implemented in Matlab.

The functionalities achieved allow the simulation, command and monitoring of the humanoid robot. It includes simple tasks as walking and turning, ready for quickly integration in more complex movements. Besides the animation of the structure with 22 degrees of freedom, it shows the temporal evolution of the joints angles, the centre of gravity and the binary along the simulated movement. Finally, the graphical user interface offers the possibility of comparison between the simulated data and the real one obtained through sensory reading of the robot.

Índice de Conteúdos

1	Introdução	1
1.1	Objectivos	3
1.2	Organização da Dissertação.....	3
2	Robótica Humanóide: Enquadramento do Trabalho.....	5
2.1	Introdução	5
2.2	O Robô Humanóide da UA.....	5
2.2.1	Estrutura Mecânica	6
2.2.2	Arquitectura de Controlo	13
2.3	Problemas de Locomoção Bípede: Conceitos Fundamentais	15
2.3.1	O Ciclo de Locomoção	15
2.3.2	Algoritmos de Cinemática Inversa	17
2.3.3	Critérios de Estabilidade.....	22
2.4	Ferramentas Utilizadas	29
2.4.1	Matlab.....	29
2.4.2	Rotinas de Desenho	31
2.4.3	Protocolo de Comunicações	32
3	Estudo de Padrões de Locomoção	35
3.1	Introdução	35
3.2	Planeamento de Trajectórias	35
3.3	Modelação de uma perna com 5 DoF	39
3.3.1	Análise Cinemática.....	40
3.3.2	Análise Estática e Dinâmica	45
3.3.3	Resultados da Simulação	47
3.4	Modelação da Estrutura com 22 DoF	50
3.4.1	Análise Cinemática.....	50
3.5	Configurações Simples	55
3.5.1	Baixar Anca	55
3.5.2	Balançar Anca	56
3.5.3	Elevar Pé Livre	57
3.6	Avaliação de Tarefas Básicas	58
3.6.1	Andar	58
3.6.2	Mudança de Direcção	65
3.6.3	Pontapear uma Bola.....	67
3.7	Discussão de Resultados	68
4	Interface Gráfica do Utilizador	69
4.1	Introdução	69
4.2	Descrição das Funcionalidades	69
4.3	Organização da Interface Gráfica	71
4.3.1	Painel de Simulação.....	72
4.3.2	Painel de Monitorização	75

4.3.3	Painel de Controlo.....	77
4.3.4	<i>Plot</i> do Robô Humanóide.....	79
4.3.5	Envio de Tarefas	80
4.3.6	Análise de Dados.....	82
4.3.7	Configuração	85
4.3.8	Menus.....	86
4.4	Estruturação da Interface Gráfica	90
4.4.1	Interfaces	93
4.4.2	Processos.....	97
4.5	Discussão de Resultados.....	104
5	Conclusão	107
5.1	Discussão de Resultados.....	107
5.2	Perspectivas de Trabalho Futuro	108
	Referências	111
Apêndice A -	Modelo Virtual.....	113
Apêndice B -	Protocolo de Comunicações	117
Apêndice C -	Representação de Denavit-Hartenberg	123
Apêndice D -	Relação entre os Parâmetros de Locomoção.....	127

Índice de Figuras

Figura 1-1 - Robô ASIMO da HONDA	1
Figura 1-2 - Robô humanóide da Universidade de Aveiro	2
Figura 2-1 - Modelo Mecânico	5
Figura 2-2 - Sistema de eixos e origem do referencial usado (ref. sobre o pé direito – vista de trás)	6
Figura 2-3 - Centros de Massa com o referencial utilizado (vista de frente)	7
Figura 2-4 - Servomotor Hitec	9
Figura 2-5 - Conjunto servomotor, correia de transmissão e polia	9
Figura 2-6 - Junta do robô humanóide numeradas	9
Figura 2-7- Trajectória polinomial de grau 5	9
Figura 2-8 - Velocidade polinomial de grau 4	9
Figura 2-9 - Aceleração polinomial de grau 3	9
Figura 2-10 - Controlador PI implementado nas slaves	10
Figura 2-11 - Sentidos de rotação positiva das juntas no modelo real (ref. pé direito)	11
Figura 2-12 - Sentido de rotação positiva das juntas no modelo virtual (ref. pé direito)	11
Figura 2-13 - Arquitectura distribuída	14
Figura 2-14 - Ciclo de locomoção	15
Figura 2-15 - Movimento da anca durante um passo	16
Figura 2-16 - Exemplo de uma cadeia cinemática aberta	18
Figura 2-17 - Redundância 2R	20
Figura 2-18 - Fase de suporte singular (retirado de [4])	24
Figura 2-19 - Forças e momentos aplicados no pé de suporte (retirado de [4])	25
Figura 2-20 - Forças de reacção no pé (retirado de [8])	25
Figura 2-21 - Pormenor do pé (retirado de [8])	26
Figura 2-22 - Pé em rotação (retirado de [4])	27
Figura 2-23 - Dois exemplos do PVCoG, CoP e FRI (retirado de [4])	28
Figura 2-24 - GUIDE	29
Figura 2-25 - Workspace GUIDE	30
Figura 2-26 - Composição do desenho humanóide	32
Figura 3-1 - Simulador Cinemático	38
Figura 3-2 - Especificação de tarefas no espaço das juntas	39
Figura 3-3 - Planos do corpo humano	39
Figura 3-4 - Perna 5 DoF (vista de frente)	40
Figura 3-5 - Perna 5 DoF (vista de lado)	40
Figura 3-6 - Sistema de eixos para a perna 5 DoF (vista de frente)	41
Figura 3-7 - Exemplo da perna de suporte para análise cinemática (vista de trás)	42
Figura 3-8 - Plano frontal (vista de trás)	43
Figura 3-9 - Plano de funcionamento das juntas de acção "sagital" (vista de trás)	43
Figura 3-10 - Equivalente do problema no plano frontal	44
Figura 3-11 - Problema nas juntas "sagitais"	45
Figura 3-12 - 1º movimento – baixar	47
Figura 3-13 - 2º movimento	48
Figura 3-14 - 3º movimento	48
Figura 3-15 - 4º movimento	49
Figura 3-16 - 5º movimento	49
Figura 3-17 - Número de cada junta no simulador (vista de frente)	50
Figura 3-18 - Sistema de coordenadas de referência (vista de trás)	50
Figura 3-19 - Sistemas de eixos usados (vista de frente)	52
Figura 3-20 - Detalhe da perna (vista frontal)	53
Figura 3-21 - Detalhe da perna (vista lateral)	53
Figura 3-22 - Juntas de controlo da função de baixar a anca	56
Figura 3-23 - Movimento de lateral	56
Figura 3-24 – Mov. lateral direita (visto de trás)	57
Figura 3-25 - Mov. lateral esquerda (visto de trás)	57

Figura 3-26 - Elevar o pé livre	58
Figura 3-27 - Parâmetro Hh (Altura da anca).....	59
Figura 3-28 - Parâmetro Yg (posição lateral da anca).....	59
Figura 3-29 - Parâmetro Fc (elevação do pé).....	59
Figura 3-30 - Sequência de fases do movimento andar	59
Figura 3-31 - Parâmetros pé de suporte (plano frontal)	60
Figura 3-32 - Parâmetros do pé de suporte (plano sagital).....	60
Figura 3-33 - Parâmetros do pé livre (plano frontal).....	61
Figura 3-34 - Parâmetros do pé de livre (plano sagital)	61
Figura 3-35 - Parâmetros pé de suporte (plano frontal)	62
Figura 3-36 - Parâmetros do pé de suporte (plano sagital).....	62
Figura 3-37 - Parâmetros do pé livre (plano frontal).....	62
Figura 3-38 - Parâmetros do pé de livre (plano sagital)	62
Figura 3-39 - Parâmetros pé de suporte (plano frontal)	63
Figura 3-40 - Parâmetros do pé de suporte (plano sagital).....	63
Figura 3-41 - Parâmetros do pé livre (plano frontal).....	63
Figura 3-42 - Parâmetros do pé de livre (plano sagital)	63
Figura 3-43 - Parâmetros pé de suporte (plano frontal)	64
Figura 3-44 - Parâmetros do pé de suporte (plano sagital).....	64
Figura 3-45 - Parâmetros do pé livre (plano frontal).....	64
Figura 3-46 - Parâmetros do pé de suporte (plano sagital).....	64
Figura 3-47 - Algoritmo para a realização de vários passos	65
Figura 3-48 - Sequência do movimento de mudança de direcção	66
Figura 3-49 - Rotação do pé livre.....	67
Figura 3-50 - Deslocação do CoG para o novo pé de suporte.....	67
Figura 3-51 - Juntas em planos diferentes (vista de cima)	67
Figura 3-52 - Sequência de movimento de pontapear	67
Figura 4-1 - Painel de abertura do <i>TwoLegs_22dof</i>	72
Figura 4-2 - Painel de Simulação	73
Figura 4-3 - Barra de controlo da Fase.....	73
Figura 4-4 - Painel de Simulação	74
Figura 4-5 - Painel de Movimentos Especiais.....	74
Figura 4-6 - Painel de visualização de Binários.....	75
Figura 4-7 - Vista geral do painel de Monitorização.....	75
Figura 4-8 - Painel de Posição	76
Figura 4-9 - Painel do centro de pressão (CoP).....	76
Figura 4-10 - Painel dos Inclinómetros	77
Figura 4-11 - Vista geral do painel de controlo.....	77
Figura 4-12 - Painel de controlo 1º nível	78
Figura 4-13 - Painel de controlo 2º nível	79
Figura 4-14 - Área de Plot do Robô Humanóide.....	80
Figura 4-15 - GUI para envio de sequências de movimentos, sendGUI	80
Figura 4-16 - Utilização do sendGUI.....	81
Figura 4-17 - Introdução da etiqueta para os dados recolhidos	81
Figura 4-18 - sendGUI durante o envio de uma sequência	82
Figura 4-19 - Animação da sequência de movimentos	82
Figura 4-20 - Análise da trajectória de cada junta ao longo do movimento	83
Figura 4-21 - Representação do CoG ao longo do tempo	83
Figura 4-22 - Análise do binário em sentido estático ao longo do movimento	84
Figura 4-23 - Análise das posições lidas durante um movimento	84
Figura 4-24 - Painel de Configuração	86
Figura 4-25 - Configuração do <i>RoboPlot</i>	86
Figura 4-26 - Barra de menu	87
Figura 4-27 - Menu de contexto das juntas.....	89
Figura 4-28 - Estruturação da Interface em Matlab	91
Figura 4-29 - Representação esquemática da interface de dados	94
Figura 4-30 - Diagrama esquemático da interface de plot	95
Figura 4-31 - Diagrama esquemático da interface de comunicações	96

Figura 4-32 - Função isolada	97
Figura 4-33 - Função com continuidade	97
Figura 4-34 - Processo desencadeado pela alteração de uma junta	98
Figura 4-35 - Cálculo de dados para plot.....	99
Figura 4-36 - Fluxograma de envio de posições.....	101
Figura 4-37 - Fluxograma de envio de posições.....	102
Figura 4-38 - Processamento interno da função <i>adapt</i>	103
Figura A-1 - Definição do pé (pé direito – vista de trás).....	113
Figura A-2 - Definição da perna	114
Figura A-3 - Definição da anca	114
Figura A-4 - Definição do Tronco.....	115
Figura A-5 - Definição do braço	115
Figura B-1 - Controlador PI implementado nas slaves.....	122
Figura C-1 - Convenção para membros, juntas e sistema de eixos	123
Figura C-2 - Representação de Denavit-Hartenberg	125
Figura D-1 - Perna (vista de frente).....	127
Figura D-2 - Perna (vista de lado).....	127
Figura D-3 - Juntas sagitais alinhadas segundo xx	128
Figura D-4 - Parâmetros Hh' e h	128
Figura D-5 - Limitações - introdução de SI (Xc)	131
Figura D-6 - h' máximo no caso de rodar e chutar	132
Figura D-7 - h' mínimo no caso de rodar e chutar	132
Figura D-8 - Nuvem de pontos possíveis para uma cadeia 2R limitada.....	132
Figura D-9 - Limite superior dado (1).....	133
Figura D-10 - Limite inferior dado (1).....	133
Figura D-11 - Limite inferior e superior dado (2).....	134

Índice de Tabelas

Tabela 2-1 - Dimensões dos Elos	6
Tabela 2-2 - Massa dos Componentes	7
Tabela 2-3 - Centros de Massa dos Componentes	8
Tabela 2-4 - Relação de transmissão das juntas	10
Tabela 2-5 – Sentidos coincidentes e opostos, +1 e -1, respectivamente, entre o modelo real e o modelo virtual	11
Tabela 2-6 - Offset dos servos na posição zero vertical	12
Tabela 2-7 – Excursão máxima e mínima das juntas no modelo virtual	13
Tabela 2-8 - Endereços das Unidades Controladoras	14
Tabela 2-9 - Elementos do desenho humanóide	32
Tabela 2-10 - Device driver comunicações - Iniciar/Finalizar	33
Tabela 2-11 - Device driver comunicações – Teste	33
Tabela 2-12 - Device driver comunicações – Leitura	33
Tabela 2-13 - Device driver comunicações - Actuação	34
Tabela 3-1 – Dimensões dos elos para a perna 5 DoF	39
Tabela 3-2 - Parâmetros de DH para a perna 5 DoF	41
Tabela 3-3 – Parâmetros de DH para a estrutura 22 DoF	51
Tabela 4-1 - Estado da ligação	72
Tabela 4-2 - Tipo de suporte disponível	73
Tabela 4-3 - Lista de funções "intermédia"	98
Tabela 4-4 - Lista de funções "final"	98
Tabela 4-5 - Descrição do <i>timer</i>	100
Tabela 4-6 - Definição do timer (1º caso)	100
Tabela 4-7 - Definição do timer (2º caso)	101
Tabela A-1 - Dimensões do pé	113
Tabela A-2 - Dimensões da perna	113
Tabela A-3 - Dimensões da anca	114
Tabela A-4 - Dimensões do tronco	114
Tabela A-5 - Dimensões do braço	115
Tabela B-1 - Valores possíveis do parâmetro <i>param</i> da função <i>readjoint</i>	119
Tabela B-2 - Valores presentes no vector <i>state</i> retornado pela função <i>readjoint</i>	119
Tabela B-3 - Valores possíveis do parâmetro <i>param</i> na função <i>applyjoint</i>	121
Tabela B-4 - Valores possíveis do parâmetro <i>param</i> na função <i>applycontrol</i>	121
Tabela B-5 - Tipo de controladores de segundo nível	122

1 Introdução

Não se pode apontar com certeza os premonitores do robô humanóide. Já no século III a.c., Aristóteles referia máquinas capazes de substituir o engenho humano, mas só oito séculos depois Leonardo Da Vinci desenhou uma estrutura mecânica inspirada na armadura de um soldado. É certo que a tecnologia necessária para tais aspirações só apareceu muito recentemente, havendo ainda muitos cépticos quanto à possibilidade de um dia máquinas com características antropomórficas terem um uso generalizado. Para esse caso, visionários como Isacc Asimov, um famoso escritor de ficção científica, deixou no seu legado as três leis da robótica que estabelecem as fronteiras do comportamento de um robô.

As constantes inovações tecnológicas fornecem cada vez mais meios à robótica humanóide para concretizar os seus fins. Esta área não obstante o seu estado prematuro, começa a mostrar alguns sinais de maturidade, ao revelar a ritmo crescente novas abordagens para os diversos problemas inerentes ao seu estudo. Apesar do seu carácter ainda académico, é um ramo da robótica que pode ter grandes impactos socioeconómicos numa sociedade cada vez mais envelhecida e carente de, por exemplo alternativas médicas complementar. O desenvolvimento de capacidades humanas em dispositivos mecânicos cria alternativas reais de substituição de membros ou órgãos humanos danificados por homólogos mecânicos capazes de reproduzir as capacidades reais e suprir as falhas existentes. Esta vertente da robótica humanóide está longe de ser a única ou a mais explorada. O enorme desafio que representa o desenvolvimento de um mecanismo antropomórfico envolve vastas áreas científicas, produzindo por isso conhecimento que se estende desde a aplicação militar até a aplicação industrial.

Actualmente, pode-se enumerar um vasto número de robôs humanóides em estado experimental e alguns já em protótipo, como é o caso do ASIMO (**Figura 1-1**), talvez o mais mediático dos robôs humanóides. Este protótipo da HONDA, que já testou mais de 40 versões, é uma das referências de muitos projectos humanóides de menor dimensão. Este robô conta já com várias capacidades cognitivas como reconhecimento visual e de voz, também mostra um importante avanço nos padrões de locomoção ao correr, subir escadas, entre outras características. Por outro lado, um factor de grande importância é o custo astronómico de cada unidade que ronda um milhão de dólares. A SONY, talvez a concorrente mais



Figura 1-1 - Robô ASIMO da HONDA

directa da HONDA, é conhecida pelos *gadgets* inovadores que apresenta ao mercado. Depois do AIBO, um cão robótico, a SONY tem no mercado o seu mais avançado robô humanóide o QRIO. Este robô apresenta capacidades semelhantes ao ASIMO mas possui metade da altura.

O governo do Japão aposta na robótica humanóide, ao ter sob a alçada do ministério da economia um instituto de ciência e tecnologia onde se projecta e desenvolve o HRP-2, um robô humanóide com aptidão para a ajuda doméstica. Mas são muitos os projectos humanóides que centros académicos como, universidades e institutos, têm em desenvolvimento e surgem cada vez mais e com diferentes abordagens aos problemas da robótica humanóide. O MIT, um dos mais reputados institutos de investigação tecnológica do mundo, possui em desenvolvimento vários robôs para investigação e estudo em diferentes áreas, como reconhecimento visual, inteligência artificial, socialização, entre outros.

Para fomentar o estudo da robótica humanóide criaram-se diversas competições que organizam periodicamente encontros onde as várias equipas participantes tentam suplantar os desafios propostos. Estes encontros servem para uma troca de conhecimentos e de experiências entre os vários centros de investigação robótica em conferências e simpósios que decorrem paralelamente às competições. A *RoboCup* é uma dessas competições e possui o objectivo de *até ao ano 2050, desenvolver uma equipa de robôs humanóides totalmente autónomos capazes de derrotar a equipa campeã mundial de futebol humana*. Os desafios propostos pela *RoboCup* são espalhados pelos diversos ramos que esta competição apresenta, entre eles o *RobocupSoccer*.

Para não deixar escapar a oportunidade de consolidar marcas nesta área, a Universidade de Aveiro, numa associação entre o Departamento de Electrónica e Telecomunicações e o Departamento de Engenharia Mecânica iniciou, em 2004 a criação de um robô humanóide de baixo custo (**Figura 1-2**). Toda a concepção do projecto, desde o esboço mecânico ao protocolo de comunicações passando pela arquitectura de controlo foi protagonizada por alunos de projecto final de ambos os departamentos. Um projecto desta envergadura envolve grande complexidade e agrega múltiplas áreas de conhecimento, mecânica, materiais, percepção, controlo, electrónica, navegação são alguns dos conhecimentos aplicados no decorrer do seu desenvolvimento. Após a estabilização das bases de suporte,

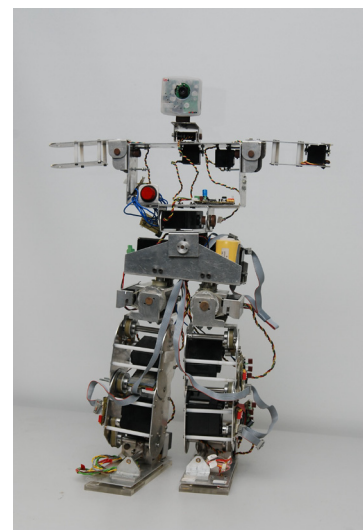


Figura 1-2 - Robô humanóide da Universidade de Aveiro

estrutura mecânica e electrónica de controlo, ao robô humanóide é chegada altura de pensar em problemas tais como controlo de alto nível, realização de movimentos coordenados, directivas de alto nível, problemas de estabilidade. É nesse contexto que surge a ideia de criar uma interface gráfica que permita facilitar a passagem para um patamar de desenvolvimento superior.

1.1 Objectivos

O objectivo central deste trabalho é o desenvolvimento de um ambiente gráfico de interacção com o robô humanóide, possibilitando a simulação, o controlo e a monitorização de toda a estrutura.

O projecto da interface gráfica envolveu o seguinte plano de trabalhos:

1. Compreensão dos principais problemas associados à locomoção bípede e estudo das questões do planeamento de trajectórias, da cinemática inversa e da estabilidade postural;
2. Desenvolvimento de um ambiente de simulação que permita ao utilizador o estudo e a avaliação de padrões de locomoção com base num modelo cinemático virtual;
3. Desenvolvimento de uma interface gráfica do utilizador (GUI) que permita a monitorização e a programação interactiva do robô real.

1.2 Organização da Dissertação

A dissertação apresenta-se dividida em 3 capítulos principais, encadeados por uma ordem crescente de ideias. Este fio condutor inicia-se com uma introdução alargada do contexto em que se insere este projecto.

Começa-se assim no Capítulo 2 pelo estudo das características actuais do robô humanóide da Universidade de Aveiro, um levantamento dos principais problemas do planeamento de movimento no robô humanóide e da locomoção bípede e a listagem fundamentada das ferramentas usadas para a execução do objectivo final.

Dedica-se o Capítulo 3 ao estudo dos padrões de locomoção a aplicar ao robô. Testam-se e aplicam-se os conhecimentos abordados no segundo capítulo. Explicam-se três padrões de locomoção envolvendo as seguintes tarefas: andar, mudar de direcção e pontapear uma bola.

Finalmente no Capítulo 4 é analisada a interface gráfica. É neste ponto que todos os tópicos anteriores convergem dando a respectiva forma ao *Graphical User Interface* (GUI). Há uma descrição das funcionalidades que a interface oferece ao utilizador e é exposta a sua organização interna.

Existe também uma secção de Apêndices onde é colocada diversa informação relativa ao projecto humanóide mas com uma relevância menor. Todos estes dados são devidamente apontados pelas secções que a eles façam referência.

2 Robótica Humanóide: Enquadramento do Trabalho

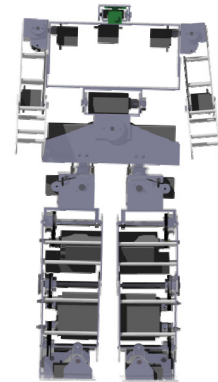
2.1 Introdução

A dissertação pressupõe a existência de um objecto alvo, que é o robô humanóide da UA. Neste capítulo pretende-se abordar todas as questões ligadas ao estado de desenvolvimento do projecto humanóide da Universidade de Aveiro (PhUA) e fazer o levantamento de dados para a criação de um modelo virtual que vai servir de base à interface. É também alvo de estudo as soluções usuais para os problemas de planeamento de movimento e locomoção bípede. Neste ponto discutem-se algoritmos de cinemática inversa e critérios de estabilidade. Por último, são referidas as ferramentas usadas para a construção da interface gráfica.

2.2 O Robô Humanóide da UA

O robô humanóide da UA foi projectado tendo em vista uma solução de baixo custo, baseado numa arquitectura de controlo distribuído, na agregação de múltiplos sensores e tendo em vista a participação em provas humanóides como o *RoboCup Humanoid League* (**Figura 2-1**). O robô assenta numa estrutura de alumínio provida de 22 graus de liberdade. Os veios são de aço de forma a assegurar a sua longevidade e eficácia. O peso total ronda os 6kg e possui uma altura de aproximadamente 60cm. Para o controlo do humanóide são usados servomotores Hitec distribuídos da seguinte forma:

- 2 em cada pé (2x2);
- 1 em cada joelho (1x2);
- 3 em cada anca (3x2);
- 2 no tronco (2x1);
- 3 em cada braço (3x2);
- 2 no suporte da câmara (cabeça) (2x1).



Posicionaram-se estrategicamente diversos sensores na estrutura humanóide, para medir o centro de pressão nos pés e a inclinação do tronco, sensores de força e inclinómetros, respectivamente. Para além destes sensores especiais, os servomotores têm o sensor de posição acessível ao programador e o robô é ainda munido de uma câmara CCD que o dota de percepção por visão artificial.

O estado actual do robô humanóide pode-se dividir em duas grandes áreas: mecânica e electrónica. Entre os parâmetros mecânicos mais relevantes estão as dimensões dos elos, as massas e respectivos centros de massa (CoG, *Center of Gravity*). A mecânica de controlo é o outro aspecto

importante e recai sobre os actuadores e as transmissões nas juntas. Por isso serão definidas as características dos servomotores e das respectivas polias de transmissão. Em último vai ser mapeado a electrónica de controlo, os endereços das *slaves* e quem são as suas dependências.

2.2.1 Estrutura Mecânica

Dimensões dos componentes

Para melhor estabelecer os comprimentos dos elos é necessário definir o referencial estabelecido para o robô humanóide, **Figura 2-2**. Os comprimentos dos elos mecânicos, obtidos através do CATIA¹ e relatórios da construção mecânica do humanóide ([1] e [3]) estão representados na tabela **Tabela 2-1**

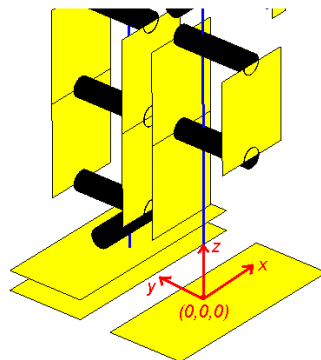


Figura 2-2 - Sistema de eixos e origem do referencial usado (ref. sobre o pé direito – vista de trás)

Elo		Comprimento (m)	Elo		Comprimento (m)
Pé – tornozelo	L(1)	0.0470	Tronco – tronco	L(10)	0.0380
Tornozelo – tornozelo	L(2)	0.0200	Tronco – tronco	L(11)	0.0230
Tornozelo – joelho	L(3)	0.0950	Secção do tronco (zz)	L(12)	0.0860
Joelho – anca	L(4)	0.0950	Secção do tronco (zz)	L(13)	0.0120
Anca – anca	L(5)	0.0555	Tronco – pescoço	L(14)	0.0147
Anca – anca	L(6)	0.0360	Pescoço – cabeça	L(15)	0.0270
Secção da anca (zz)	L(7)	0.0325	Secção do tronco (yy)	L(16)	0.0550
Secção da anca (yy)	L(8)	0.0840	Tronco – ombro	L(17)	0.0180
Anca – tronco	L(9)	0.0215	Ombro – pulso	L(18)	0.1151

Tabela 2-1 - Dimensões dos Elos

¹ CATIA (Computer Aided Three Dimensional Interactive Application) é um software de CAD/CAM/CAE para desenvolvimento de projectos mecânicos

No **Apêndice A** estão descritas as dimensões dos componentes de desenho virtual do robô humanóide. As massas apresentadas foram retiradas do CATIA e relatórios do trabalho mecânico [1], **Tabela 2-2**.

Componente	Massa (kg)	Componente	Massa (kg)
Pé	0.524	Secção da anca	1.212
Tornozelo	0.165	Cintura	0.246
Perna	0.410	Secção do tronco	0.457
Coxa	0.371	Pescoço/Cabeça	0.067
Anca baixa	0.239	Ombro	0.071
Anca cima	0.274	Braço	0.064

Tabela 2-2 - Massa dos Componentes

Os centros de massa foram obtidos recorrendo à ferramenta de modelação CATIA. Para isso, criou-se um referencial no canto superior direito do pé direito, **Figura 2-3**, com o eixo da abcissa perpendicular ao plano sagital, o eixo da ordenada perpendicular ao plano lateral e o eixo da cota perpendicular ao plano transversal.

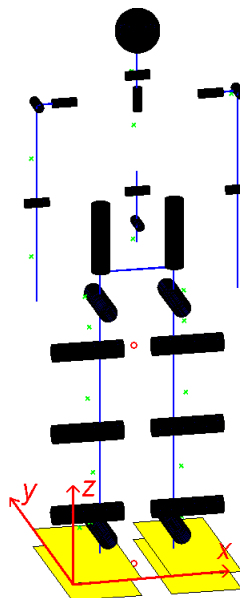


Figura 2-3 - Centros de Massa com o referencial utilizado (vista de frente)

Como a estrutura apresenta simetria em relação ao plano sagital, criando um referencial simétrico no pé esquerdo obtém-se para o lado esquerdo da estrutura os mesmos valores para o centro de

massa dos componentes do lado esquerdo. Por essa razão só são apresentados na **Tabela 2-3** os centros de massa relativos ao referencial do pé direito.

Componente	Centro de Massa (mm)		
	xx	yy	zz
Pé direito	40.00	142.00	17.00
Tornozelo direito	45.00	107.00	34.20
Perna direita	46.30	102.00	94.50
Coxa direita	41.00	103.30	180.70
Anca direita 1	42.80	103.70	264.30
Anca direita 2	40.50	115.50	295.80
Ombro direito	-12.24	109.47	524.20
Braço direito	-24.18	101.12	467.00
Barra anca	95.90	118.40	358.90
Cintura	93.90	114.00	413.40
Tronco	95.30	111.80	494.80
Pescoço	95.30	121.30	554.60

Tabela 2-3 - Centros de Massa dos Componentes

Características dos servos

Previamente foi referido que os servos usados eram Hitec (**Figura 2-4**), mas existem no robô humanóide servomotores com diferentes características, dependendo das exigências locais. Para a escolha dos servomotores efectuaram-se simulações de binários necessários na execução de diferentes tarefas e com base nesses binários escolheram-se servomotores entre duas categorias. Complementarmente, aplicaram-se nas juntas polias para ajustar às necessidades simuladas [1].

As características dos servomotores avaliadas para o modelo humanóide incluem a análise ao mecanismo de transmissão. A ligação do servomotor ao veio é obtida por intermédio de uma polia (**Figura 2-5**). Desta forma, consegue-se atingir um binário, na junta, superior ao binário fornecido pelo motor. A razão de proporcionalidade estabelecida entre os dois binários, na junta e no motor, é igual à relação entre os raios da polia da junta e da polia do motor, respectivamente. Na mesma proporcionalidade em que o binário aumenta, a velocidade de rotação e a excursão máxima dos servomotores diminui. Esta constante é denominada por relação de transmissão.

Nesta análise vão ser estudadas as relações de transmissão e, os limites e sentidos de rotação de cada junta. Na **Figura 2-6** está o número atribuído a cada junta no robô.



Figura 2-4 - Servomotor Hitec



Figura 2-5 - Conjunto servomotor, correia de transmissão e polia

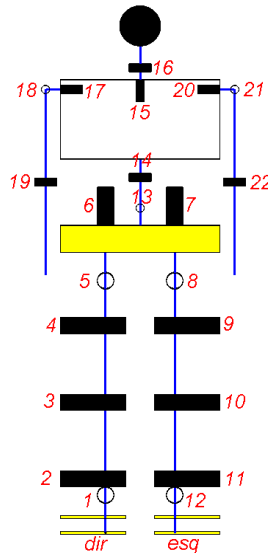


Figura 2-6 - Junta do robô humanóide numeradas

No robô humanóide os servomotores são actuados através de PWM (*Pulse Width Modulation*) com um período de 20ms. Está implementado nas unidades controladoras *slave* um algoritmo gerador de trajectórias. A trajectória é definida por um polinómio de grau 5, capaz de uma transição suave no arranque e na paragem e permite ter uma velocidade e uma aceleração inicial e final nulas. Nas **Figura 2-7**, **Figura 2-8** e **Figura 2-9**, estão representadas as curvas geradas para a posição, velocidade e aceleração, respectivamente.

$$pos = c_0 + c_1 \cdot t + c_2 \cdot t^2 + c_3 \cdot t^3 + c_4 \cdot t^4 + c_5 \cdot t^5$$

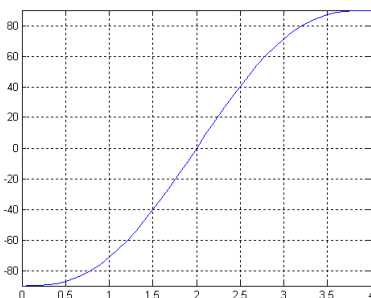


Figura 2-7- Trajectória polinomial de grau 5

$$vel = c_1 + 2 \cdot c_2 \cdot t + 3 \cdot c_3 \cdot t^2 + 4 \cdot c_4 \cdot t^3 + 5 \cdot c_5 \cdot t^4$$

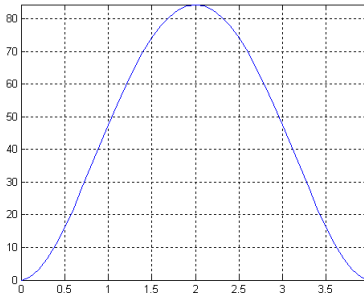


Figura 2-8 - Velocidade polinomial de grau 4

$$ace = 2 \cdot c_2 + 6 \cdot c_3 \cdot t + 12 \cdot c_4 \cdot t^2 + 20 \cdot c_5 \cdot t^3$$

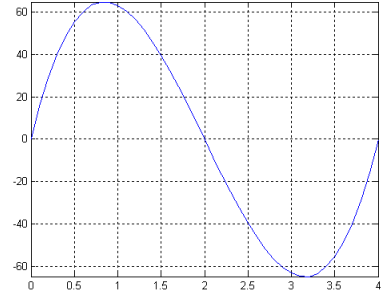


Figura 2-9 - Aceleração polinomial de grau 3

Além do controlador interno do servomotor, um proporcional derivativo (PD), está implementado nas unidades controladoras *slaves* um algoritmo de controlo proporcional integrador (PI). Este último está representado na **Figura 2-10**.

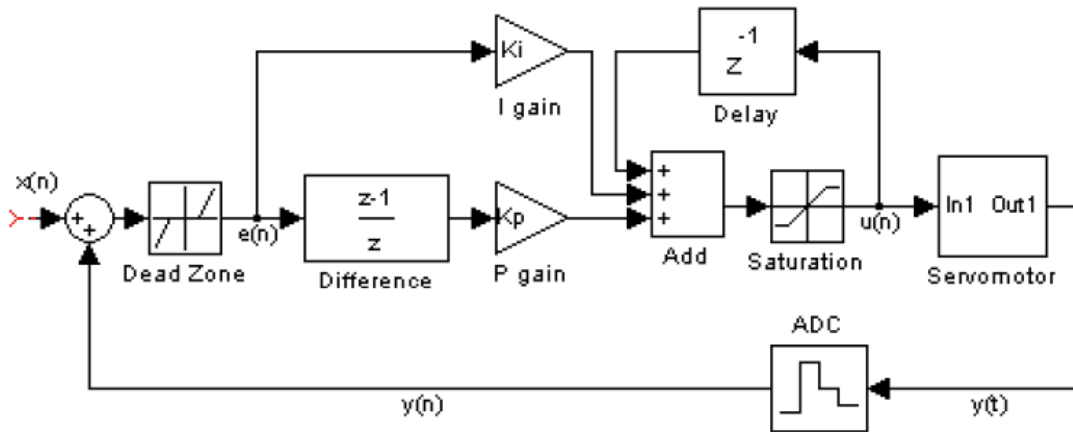


Figura 2-10 - Controlador PI implementado nas slaves

A relação de transmissão existente entre os servos e as juntas apresentam valores de 1, 2.5 ou 3.75, esses valores estão representados na **Tabela 2-4**.

Junta	Relação de Transmissão	Junta	Relação de Transmissão
1	2.50	12	2.50
2	2.50	13	1.00
3	2.50	14	1.00
4	2.50	15	1.00
5	3.75	16	1.00
6	1.00	17	1.00
7	1.00	18	1.00
8	3.75	19	1.00
9	2.50	20	1.00
10	2.50	21	1.00
11	2.50	22	1.00

Tabela 2-4 - Relação de transmissão das juntas

O sentido de rotação das juntas no modelo virtual do robô humanóide é diferente do modelo real. Esta diferença assenta sobretudo em pressupostos de coerência no sentido de rotação entre componentes homólogos do modelo humanóide, tais como o pé do lado direito e o pé do lado esquerdo. Estes pormenores técnicos facilmente conseguidos num simulador virtual são relegados para última prioridade no desenvolvimento do projecto mecânico. Dessa forma a distinção do sentido de rotação real e virtual é indicada através da seguinte matriz, **Tabela 2-5**. Na **Figura 2-11**

e **Figura 2-12** estão ilustrados os sentidos de rotação positiva real e virtual das juntas. Estes valores podem também ser interpretados como: positivo quando os sentidos de rotação coincidem entre as duas realidades e negativo quando são opostos.

Junta	Sentido de rotação
1	+1
2	+1
3	-1
4	-1
5	-1
6	+1
7	-1
8	+1
9	+1
10	+1
11	-1

Junta	Sentido de rotação
12	-1
13	+1
14	+1
15	+1
16	+1
17	+1
18	+1
19	+1
20	+1
21	+1
22	+1

Tabela 2-5 – Sentidos coincidentes e opostos, +1 e -1, respectivamente, entre o modelo real e o modelo virtual

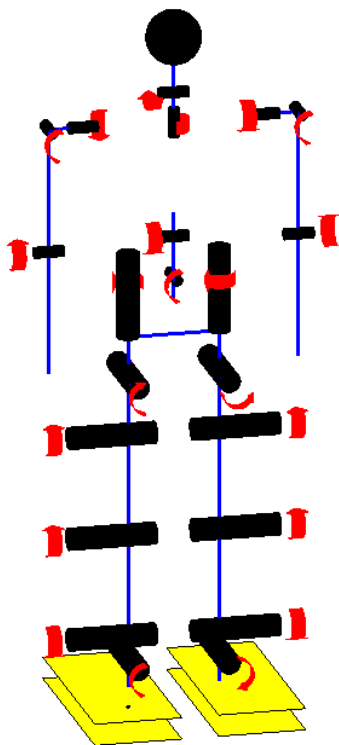


Figura 2-11 - Sentidos de rotação positiva das juntas no modelo real (ref. pé direito)

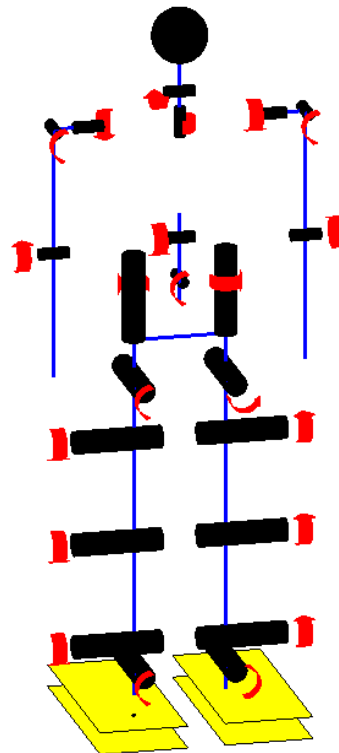


Figura 2-12 - Sentido de rotação positiva das juntas no modelo virtual (ref. pé direito)

Os servomotores usados possuem uma excursão máxima de 180 graus, estendendo-se entre os -90 e os 90 graus. Quando o robô está na posição vertical, tal como representado na **Figura 2-6**, designada por *zero vertical*, os servomotores apresentam um *offset* inicial projectado para aproveitar da melhor forma a excursão dos servomotores na execução de movimentos antropomórficos. Na **Tabela 2-6** apresenta-se esse *offset* existente nos servomotores, designado por *offset* estático. Analogamente aos seres humanos, as juntas dos joelhos no robô humanóide limitam um dos sentidos de rotação, havendo duas razões para esta exclusividade:

1. As outras juntas existentes, tal como num ser humano têm a mesma liberdade de movimentos em ambos os sentidos;
2. Juntas como os cotovelos, por exemplo, pelo facto de terem transmissão unitária, aproveitam ao máximo a excursão dos servomotores de -90 até 90 graus, não necessitando dessa forma de limitar um dos sentidos em benesse do outro.

Junta	Posição Inicial	Junta	Posição Inicial
1	0	12	0
2	0	13	0
3	-70	14	0
4	0	15	0
5	0	16	0
6	0	17	0
7	0	18	0
8	0	19	0
9	0	20	0
10	70	21	0
11	0	22	0

Tabela 2-6 - Offset dos servos na posição zero vertical

Para a correcta simulação e validação de padrões de locomoção é necessário definir as limitações de cada elemento presente no robô humanóide. Essa análise foi efectuada tendo em conta unicamente as especificações dos servomotores e transmissão. Esta abordagem não inclui as limitações físicas inerentes a toda a estrutura mecânica, isto é, as barreiras impostas pelo exosqueleto do humanóide ao livre movimento de todos os componentes. A razão para a não inclusão das barreiras impostas pelo exosqueleto deve-se à cadeia de elos complexa que é um robô humanóide, onde existem vários limites variáveis dependentes da configuração actual das juntas, este facto é a principal dificuldade em modelar essas dependências. Dessa forma os limites apresentados na **Tabela 2-7** são os limites máximos e mínimos de excursão real de uma junta no modelo virtual, atendendo aos *offsets*, relações de transmissão, sinais de rotação e excursão máxima e mínima dos servomotores.

Junta	Inferior	Superior	Junta	Inferior	Superior
1	-36	36	12	-36	36
2	-36	36	13	-36	36
3	-64	8	14	-36	24
4	-36	36	15	-24	90
5	-24	24	16	-90	90
6	-36	36	17	-90	90
7	-36	36	18	-90	90
8	-24	24	19	-90	90
9	-36	36	20	-90	90
10	-64	8	21	-90	90
11	-36	36	22	-90	90

Tabela 2-7 – Excursão máxima e mínima das juntas no modelo virtual

2.2.2 Arquitectura de Controlo

A arquitectura de controlo distribuída era um dos pressupostos do projecto humanóide. Dessa forma o controlo baseia-se em pequenas unidades distribuídas por toda a estrutura (**Figura 2-13**). O sistema é constituído por três tipos de unidades:

- Unidade Central
- *Master Control Unit* (MCU)
- *Slave Control Unit* (SCU)

A Unidade Central é uma unidade singular responsável pela gestão global de toda a estrutura. Esta unidade envia comandos de controlo para todo o sistema. É também por ela que passa todo o processamento e gestão da visão. O *Master Control Unit* serve de interface entre a unidade central e as *slaves*, é um único dispositivo que serve de distribuidor dos comandos enviados pela Unidade Central para as *slaves*. É uma base de dados global dos parâmetros de actuação e sensoriais de todo o robô humanóide. As *Slave Control Unit* têm a função de controlo local, há ao todo 8 espalhadas pelo humanóide, cada uma com capacidade para actuação directa até 3 servomotores e também pela leitura dos sensores existentes na sua área de actuação.

Há que destacar nesta arquitectura a indiferenciação do *firmware* incluído em cada uma das *slaves*, sendo que, o único meio de distinção de cada unidade é produzido por um identificador, alterado através de um *switch* colocado em cada unidade. Cada uma das unidades controladoras permite a acoplação através do circuito *piggy-back* de 4 sensores especiais não impondo quaisquer restrições ao sensor usado, além do seu interface de leitura. Esta modularidade e adaptabilidade do *firmware*

às diferentes exigências locais é uma característica de grande utilidade num projecto com esta complexidade.

Apesar de o *firmware* ser comum a todas as *slaves*, há pequenos segmentos de código direccionados para determinados locais de controlo, por exemplo o controlador dos sensores de força só deve ser usado nos pés/perna do humanóide, porque é onde esses sensores residem. Mas esta informação provém de ordens superiores, quer isto dizer que esse mesmo segmento de código pode ser activado para outro qualquer ponto de controlo, sujeito a um qualquer sensor especial acoplado a essa unidade e com um comportamento imprevisível. É por isso que é necessário ter bem definido na unidade central os endereços de cada uma das *slaves*, a sua localização e os sensores especiais que estão ligados à mesma.

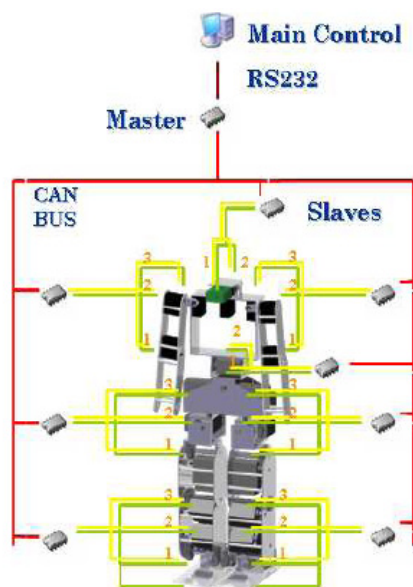


Figura 2-13 - Arquitectura distribuída

Na **Tabela 2-8** indicam-se os endereços de cada uma das unidades de controlo local, a sua localização e os sensores especiais que acondicionam.

ID	Unidade	Sensores Especiais
0	Master	-
1	Perna direito	Sensores de Força (CoP)
2	Perna esquerda	Sensores de Força (CoP)
3	Anca direita	Inclinómetros
4	Anca esquerda	Inclinómetros
5	Tronco	-
6	Braço direito	-
7	Braço esquerdo	-
8	Cabeça	-

Tabela 2-8 - Endereços das Unidades Controladoras

Para a comunicação entre as diferentes unidades, estão definidos dois tipos de ligação, uma linha série RS-232 que permite uma comunicação bidireccional ponto a ponto entre a Unidade Central e o *master* e um barramento CAN (*Controller Area Network*), que permite a comunicação multiponto entre o *master* e as *slaves* [2].

2.3 Problemas de Locomoção Bípede: Conceitos Fundamentais

2.3.1 O Ciclo de Locomoção

A locomoção bípede é um processo repetitivo, pode-se por isso caracterizar o ciclo da locomoção de acordo com diferentes parâmetros. Na **Figura 2-14**, considera-se o ciclo de locomoção constituído por dois passos e caracteriza-se o passo sob dois aspectos: o suporte em cada fase e o movimento da perna livre. O suporte no processo de locomoção, bem como em grande parte das tarefas executadas pelos seres bípedes, pode ser simples ou duplo, um pé ou dois pés, respectivamente. A caracterização do passo quanto ao suporte pode ser dividida em três fases [9]:

1. Duplo suporte (DS) - preparação do passo
2. Suporte simples (SS) - execução do passo
3. Duplo suporte - finalização do passo

Na primeira fase o pé de suporte da próxima fase ajusta-se para suportar todo o corpo durante a fase seguinte e o pé que vai executar o passo dá o impulso para o corpo balançar para a frente. Durante a fase de suporte simples o corpo pode ser visto como um pêndulo invertido, onde a perna de suporte é a haste do pêndulo. Na terceira fase o pé livre atinge o solo, amortece o movimento descendente do corpo e prepara-se para ser o pé de suporte da próxima fase.

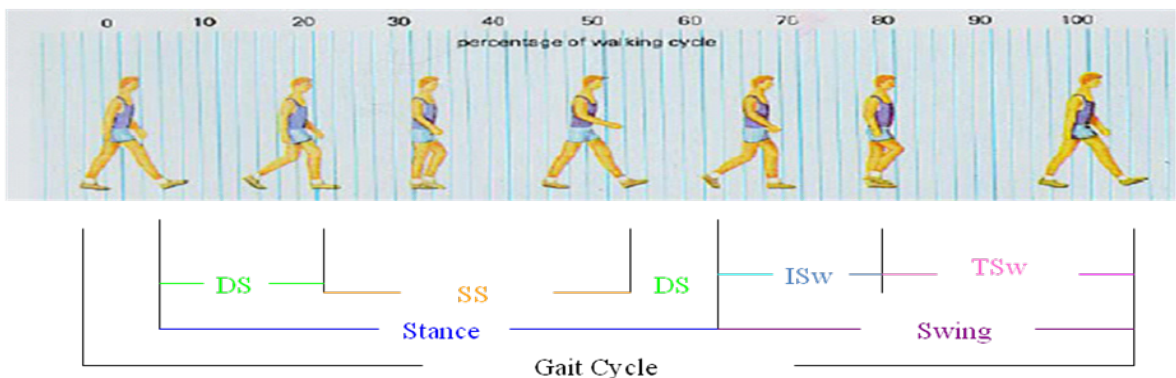


Figura 2-14 - Ciclo de locomoção

Do ponto de vista do balanço da perna livre são observadas duas fases [9]:

1. Início - A perna livre retrai-se e o pé impele o solo;
2. Balanço - A perna livre balança para a frente;
3. Fim - A perna livre distende-se e o pé atinge o solo.

O movimento da anca durante a locomoção evidencia bem o modelo de pêndulo invertido usado na descrição do processo de andar, **Figura 2-15**.

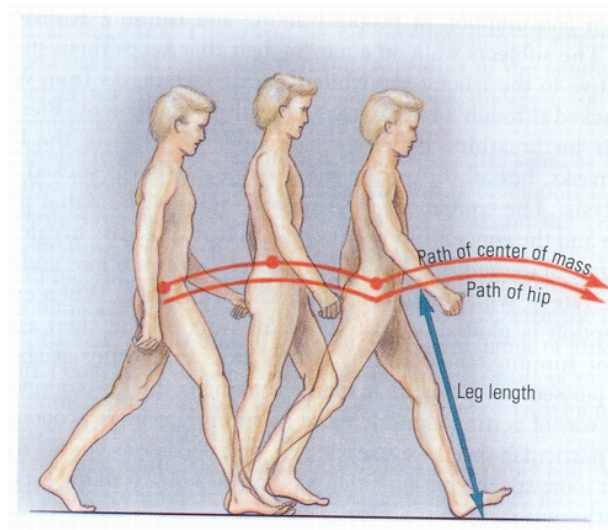


Figura 2-15 - Movimento da anca durante um passo

A locomoção bípede é um processo indubitavelmente de natureza instável, é por isso necessário um controlador do sistema que possibilite a sua estabilização. Um problema é a interacção entre o pé de suporte e o solo, porque ao contrário dos manipuladores onde existe uma base fixa, neste caso não há forças atractivas entre o solo e o pé. Este facto implica que a estabilidade do pé de suporte tem que ser garantida durante esta fase, sob pena de se cair. Na transição entre a fase de SS e a fase de DS, há um impacto do pé livre com o solo, esta chegada deve ser o mais suave possível, de forma a absorver a energia cinética adquirida durante o SS e projectá-la para o passo seguinte. Este efeito de mola do pé livre também é necessário na passagem entre o DS e o SS, ao ter que impulsionar o corpo para a frente.

2.3.2 Algoritmos de Cinemática Inversa

A cinemática inversa tem uma importância fulcral na simulação/animação de estruturas articuladas. Na verdade, torna-se quase impossível controlar a postura do robô ou executar uma determinada tarefa manipulando directamente cada junta. Tipicamente, o planeamento é expresso em coordenadas operacionais e tem como função principal especificar um conjunto de variáveis desejadas (referências) como entrada para um controlador. Um algoritmo de cinemática inversa permite que a localização de certos pontos seleccionados na estrutura sejam convertidos num conjunto de ângulos das juntas. Esta função, que relaciona os dois espaços, é uma função sobrejectiva, isto é, pode apresentar redundância múltipla. Isto implica, que a solução analítica da cinemática inversa só é possível através da inclusão de restrições ao sistema em análise. Estas restrições podem impossibilitar certas configurações estruturais, tornando o sistema indeterminado para esses casos, e a eliminação da redundância múltipla pode nem sempre acontecer. Este problema cresce com o aumento de juntas envolvidas na cadeia cinemática. É por estas razões que o problema da cinemática inversa é delicado e tem de ser analisado com muito cuidado, impondo restrições que assegurem uma solução possível e única. Se tal não for alcançado para todos os casos possíveis, deve-se garantir métodos condicionais de resolver essas situações particulares.

Definição

Seja $f : q \in \mathcal{R}^n \rightarrow SE(3)$, tal que q representa o valor das n juntas de uma cadeia cinemática, f é a função de cinemática directa que devolve a posição e orientação do último elo da cadeia e $SE(3)$ é o espaço operacional. O problema da cinemática inversa pode ser exposto da seguinte forma, dado $G \in SE(3)$, determinar q de tal forma que $f(q) = G$.

A cinemática inversa é um problema, que ao contrário da cinemática directa, não tem solução única. Existem portanto, diversos métodos alternativos para o cálculo desta cinemática, alguns de cariz analítico, outros de forma numérica [10] [11]. Ambas as abordagens apresentam vantagens e desvantagens que necessitam de uma análise mais aprofundada de forma a encontrar a mais adequada para aplicar no simulador cinemático.

Métodos analíticos

Os métodos analíticos são conhecidos como completos, porque determinam todas as soluções possíveis para determinado problema de cinemática inversa. Além desta vantagem, também a celeridade e fiabilidade destes métodos são factores de peso para a sua preferência mas o reverso da medalha é a dificuldade em parametrizar e restringir a cadeia cinemática em análise.

Existem basicamente dois subtipos de métodos analíticos, métodos de eliminação algébrica ou solução em forma fechada. A solução em forma fechada corresponde a um conjunto de equações que definem as variáveis das juntas. Este método só é aplicável, regra geral, para uma cadeia cinemática até 6 graus de liberdade (DoF, *Degree of Freedom*). Os métodos de eliminação algébrica definem as variáveis das juntas como solução de um polinómio multivariável ou alternativamente, a solução para o polinómio corresponde a uma única variável de junta e todas as outras variáveis de juntas são obtidas por uma solução em forma fechada.

Para exemplificar um método analítico considere-se o seguinte esquema:

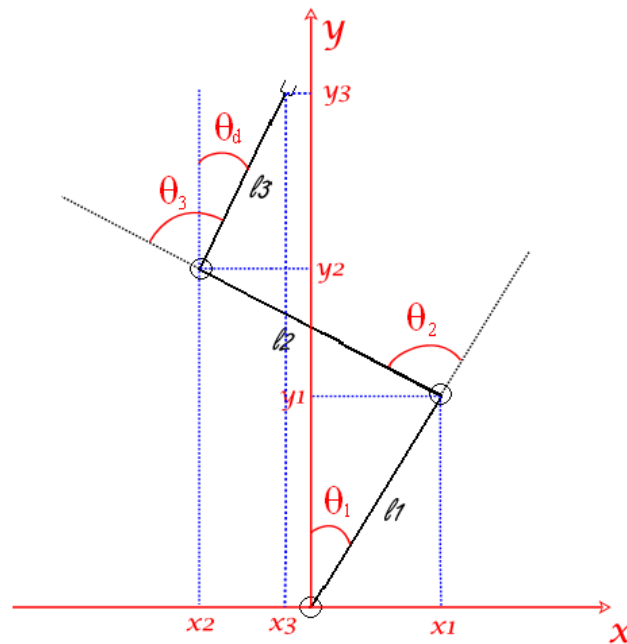


Figura 2-16 - Exemplo de uma cadeia cinemática aberta

A cinemática directa deste exemplo é dada por:

$$\begin{aligned} x_1 &= l_1 \cdot \text{sen}(\theta_1) \\ y_1 &= l_1 \cdot \text{cos}(\theta_1) \end{aligned} \quad (2-1)$$

$$\begin{aligned} x_2 &= l_1 \cdot \text{sen}(\theta_1) + l_2 \cdot \text{sen}(\theta_1 + \theta_2) \\ y_2 &= l_1 \cdot \text{cos}(\theta_1) + l_2 \cdot \text{cos}(\theta_1 + \theta_2) \end{aligned} \quad (2-2)$$

$$\begin{aligned} x_3 &= l_1 \cdot \text{sen}(\theta_1) + l_2 \cdot \text{sen}(\theta_1 + \theta_2) + l_3 \cdot \text{sen}(\theta_1 + \theta_2 + \theta_3) \\ y_3 &= l_1 \cdot \text{cos}(\theta_1) + l_2 \cdot \text{cos}(\theta_1 + \theta_2) + l_3 \cdot \text{cos}(\theta_1 + \theta_2 + \theta_3) \end{aligned} \quad (2-3)$$

Para determinar as equações de cinemática inversa, pode-se simplesmente inverter as equações da cinemática directa, ficando para um elo:

$$\begin{aligned} \theta_1 &= a \operatorname{sen}\left(\frac{x_1}{l_1}\right) \\ \theta_1 &= a \operatorname{cos}\left(\frac{y_1}{l_1}\right) \end{aligned} \quad (2-4)$$

Neste caso é possível determinar o θ através de uma das equações, ou seja em função de x_1 ou y_1 , isto porque há duas equações para apenas uma incógnita.

Para o caso de dois elos já se têm duas equações para duas incógnitas:

$$\begin{cases} [x_2 - l_1 \cdot \operatorname{sen}(\theta_1)]^2 = [l_2 \cdot \operatorname{sen}(\theta_1 + \theta_2)]^2 \\ [y_2 - l_1 \cdot \operatorname{cos}(\theta_1)]^2 = [l_2 \cdot \operatorname{cos}(\theta_1 + \theta_2)]^2 \end{cases} \quad (2-5)$$

Através da manipulação destas equações é possível chegar-se à seguinte solução [10],

$$\begin{aligned} k_1 &= 2 \cdot y \cdot l_1 \\ k_2 &= 2 \cdot x \cdot l_1 \\ k_3 &= x^2 + y^2 + l_1^2 - l_2^2 \end{aligned} \quad (2-6)$$

$$\theta_1 = \arctan\left(\frac{k_1}{k_2}\right) + c \cdot \arctan\left(\frac{\sqrt{k_1^2 + k_2^2 - k_3^2}}{k_3}\right)$$

$$\begin{aligned} k_1 &= 0 \\ k_2 &= 2 \cdot l_1 \cdot l_2 \\ k_3 &= x^2 + y^2 - l_1^2 - l_2^2 \end{aligned} \quad (2-7)$$

$$\theta_2 = \arctan\left(\frac{k_1}{k_2}\right) - c \cdot \arctan\left(\frac{\sqrt{k_1^2 + k_2^2 - k_3^2}}{k_3}\right)$$

A variável c determina a direcção de rotação do θ_2 , isto é o primeiro problema de redundância da cinemática inversa e pode ser ilustrado da seguinte forma:

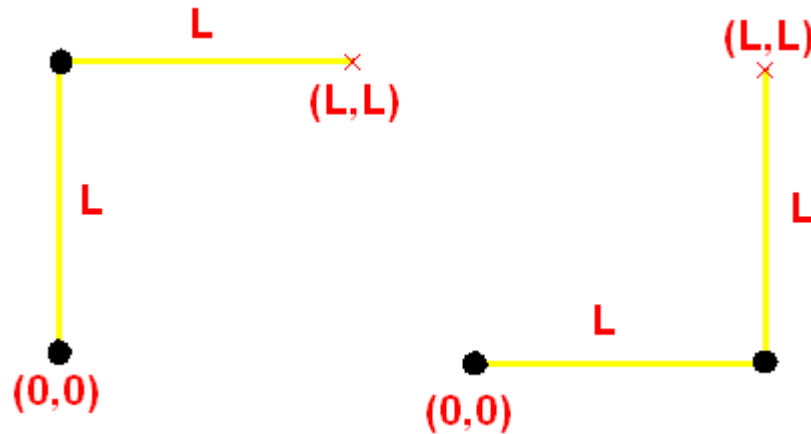


Figura 2-17 - Redundância 2R

O mesmo problema resolvido de duas formas distintas, o parâmetro c das equações em (2-6) e (2-7) serve para definir a configuração do “cotovelo”.

O caso dos três elos já não pode ser resolvido invertendo as equações da cinemática directa porque há três incógnitas e só duas equações. Pode-se então restringir o problema de cinemática 3R ao especificar um θ_d para o terceiro elo da cadeia e reduzindo o problema da cinemática 3R a um simples problema de cinemática 2R resolvido no ponto anterior. Dessa forma, depois de determinar θ_1 e θ_2 , θ_3 é dado por:

$$\theta_3 = \theta_d - (\theta_1 + \theta_2) \quad (2-8)$$

Métodos numéricos

Os métodos numéricos convergem iterativamente para uma solução do problema cinemático, tendo como ponto de partida um valor aleatório. Há 3 tipos usuais de métodos numéricos, a aplicação directa do algoritmo Newton-Raphson para a resolução de equações não lineares, a conversão do problema para uma equação diferencial em função de \mathbf{q} e $\dot{\mathbf{q}}$, e finalmente, uma terceira abordagem é converter o problema da cinemática inversa num problema de optimização não linear.

O método de Newton-Raphson pode-se expor da seguinte forma,

$$F(\mathbf{q}) \equiv f(\mathbf{q}) - \mathbf{g} = 0 \quad (2-9)$$

Onde, f é a equação de cinemática directa do sistema e g é a posição e orientação do elo terminal da cadeia cinemática. Este método converge para a solução do problema, apesar dessa conversão

poder ser lenta quando as equações são altamente não lineares. Há algumas variâncias do método de Newton-Raphson mais apropriadas para os problemas cinemáticos, nomeadamente o método de Piepers.

Os métodos baseados em equações diferenciais colocam o problema da cinemática inversa numa questão de variação da posição ou seja, velocidade. A velocidade pode então ser integrada entre 0 e t_f de forma a satisfazer a seguinte equação

$$q(t_f) = q_0 + \int_0^{t_f} \dot{q}(t) \cdot dt \quad (2-10)$$

Este método é muito idêntico ao método de Newton-Raphson, no entanto utiliza técnicas de integração que se revelam mais precisas e robustas.

A cinemática inversa pode ser vista como um problema de optimização não linear, isto é, seja $f(q)$ a solução da cinemática directa e x_d a posição desejada para o elo final, então pode-se definir o problema de acordo com a seguinte função:

$$P(q) = (f(q) - x_d)^T \cdot (f(q) - x_d). \quad (2-11)$$

A função, sempre positiva, apresenta um mínimo para qualquer q que resolva o problema Cartesiano.

O método da jacobiana inversa é um exemplo de aplicação de um método numérico. A matriz Jacobiana é formada pelas derivadas parciais de primeira ordem de uma função vectorial. Esta matriz permite relacionar grandezas em espaços distintos, isto é, o espaço Cartesiano e o espaço das juntas, e pode ser calculada da seguinte forma:

$$J(\mathbf{f}, \mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_M}{\partial x_1} & \dots & \dots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} \quad (2-12)$$

Considerando \mathbf{f} a função de cinemática directa, pode-se usar a inversa desta matriz para o cálculo da cinemática inversa, da seguinte forma:

$$\dot{\mathbf{r}} = J(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \Leftrightarrow \dot{\boldsymbol{\theta}} = J^{-1}(\boldsymbol{\theta}) \cdot \dot{\mathbf{r}} \quad (2-13)$$

Onde $\dot{\mathbf{r}}$ é o vector da variação da posição Cartesiana e $\dot{\boldsymbol{\theta}}$ é o vector da variação das posição nas juntas. Pressupondo o espaço Cartesiano (x,y) e duas juntas (θ_1, θ_2), a matriz jacobiana é dada por:

$$J(\mathbf{r}, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \quad (2-14)$$

A equação (2-13), pode ser aproximada por:

$$\dot{\boldsymbol{\theta}} = J^{-1}(\boldsymbol{\theta}) \cdot \dot{\mathbf{r}} \cong \Delta\boldsymbol{\theta} \approx J^{-1} \cdot \Delta\mathbf{r} \quad (2-15)$$

Considerando então um $\mathbf{e}=(\mathbf{e}_x, \mathbf{e}_y)$, como sendo o erro entre $\mathbf{r}_{\text{actual}}$ e o $\mathbf{r}_{\text{desejado}}$ e se se garantir que \mathbf{e} é pequeno, então pode-se aplicar a equação (2-15). Para assegurar um \mathbf{e} pequeno pode-se dividir o erro em pequenos segmentos. Desta forma define-se um k tal que, $0 < k < 1$. O algoritmo de cálculo é então,

```
while Δe
{
    calcular J(e,θ)           // para o θ actual
    Calcular J-1           // o jacobiano invertido
    Δe = k(ractual - rdesejado) // determinar o erro em pequenos incrementos
    Δθ = J-1 · Δe         // calcular as variações de θ
    θ = θ + Δθ              // calcular o θ real
}
```

2.3.3 Critérios de Estabilidade

Segundo a teoria do controlo, um sistema estável é aquele onde dada uma entrada limitada em amplitude, a sua resposta também é limitada em amplitude. Num sistema complexo como o robô humanóide, altamente não linear e multidimensional, a aplicação do conceito de estabilidade no sentido lato torna-se praticamente impossível e de difícil compreensão. É por isso que se

desenvolveram várias teorias ou critérios que garantem a estabilidade deste tipo de sistemas, nomeadamente em determinadas tarefas como a locomoção.

A modelação matemática de um sistema envolve o levantamento de todas as variáveis que influenciam o seu comportamento. Cria-se um paradoxo entre a complexidade do modelo matemático e a precisão com que este modela o sistema real. É preciso um sentido crítico por parte do projectista para criar um modelo que responda dentro da precisão com que pretende analisar o sistema. É à luz desta explicação que se pode falar de estabilidade estática, estabilidade quasi-estática e estabilidade dinâmica. Num sistema parado, em equilíbrio, pode-se analisar a sua estabilidade com base num modelo matemático estático, isto é, que não entra em linha de conta com as acelerações internas dos seus elementos. Usa-se uma análise dinâmica quando o sistema que se está a estudar apresenta movimento. O conceito de estabilidade quasi-estática deriva de uma análise estática aplicada a um sistema em movimento mas com a condição de uma velocidade reduzida.

Pode-se entender um robô humanóide como um mecanismo balístico que interage com o seu ambiente através do pé. Para o caso de um robô estacionário, ele será sempre estável desde que a projecção vertical do centro de massa (PVCoG) permaneça dentro do polígono de suporte. O polígono de suporte é a denominação técnica para a área de contacto entre o pé e o solo. Para um robô em movimento este critério deixa de ser suficiente para garantir a estabilidade, isto porque a PVCoG exclui os efeitos inerciais do movimento que podem afectar o equilíbrio. Tem então de se recorrer a outros critérios que incorporem a dinâmica do robô. Os mais usados são o centro de pressão (CoP), o *zero moment point* (ZMP) e *foot rotation indicator* (FRI).

O conceito de *zero moment point* (ZMP) foi introduzido no fim da década de 60 por Miomir Vukobratović. Desde então, a definição do ZMP tem sido largamente usada como critério de estabilidade para locomoção bípede [4] [6] [9] [12] [14].

Em 1999, Goswami atesta duas definições de ZMP como correctas e consistentes [4]:

- **Def1 (Hemani et Golliday [1977])** – ZMP é o ponto em que a força de reacção vertical intersecta o solo;
- **Def2 (Arakawa e Fukuda [1997])** – ZMP é o ponto no solo em que o momento $\vec{T} : (T_x, T_y, T_z)$ gerado pela força de reacção e o binário de reacção satisfazem $T_x = 0$ e $T_y = 0$.

Goswami prova depois que o CoP é igual ao ZMP, sendo pontos coincidentes mas definidos em diferentes termos. O CoP é definido em função da força de reacção do solo, enquanto o ZMP é definido em função das forças dinâmicas.

Considerando a **Figura 2-18**, pode-se observar as diversas forças que actuam sobre um elo genérico. As forças externas que actuam sobre o robô são a resultante da força e momento de reacção do solo, \vec{R} e \vec{M} e da força gravítica $m\vec{g}$. A equação de equilíbrio rotacional é obtida notando que a soma dos momentos das forças externas que actuam sobre o robô, calculados no CoG ou em qualquer ponto estacionário de referência, é igual à soma da taxa de variação do momento angular dos segmentos individuais em torno desse ponto. Calculando os momentos na origem O, fica,

$$\vec{M} + \vec{OP} \times \vec{R} + \sum_{i=1}^n \vec{OG}_i \times m_i \cdot \vec{g} = \sum_{i=1}^n \vec{H}_{G_i} + \sum_{i=1}^n \vec{OG}_i \times m_i \cdot \vec{a}_i \quad (2-16)$$

Onde m_i é a massa do elemento, \vec{G}_i a localização do CoG, \vec{a}_i a aceleração linear do CoG e \vec{H}_{G_i} o momento angular em torno do CoG_i.

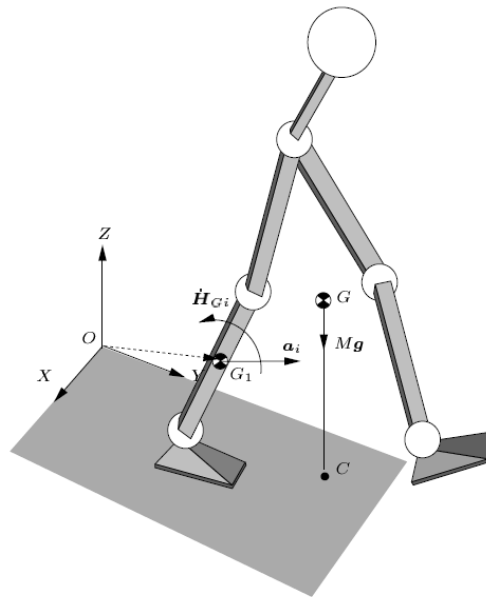


Figura 2-18 - Fase de suporte singular (retirado de [4])

A equação de equilíbrio dinâmico do pé de suporte obedece à expressão:

$$\vec{M} + \vec{OP} \times \vec{R} + \vec{OG}_1 \times m_1 \cdot \vec{g} - \vec{\tau}_1 - \vec{OO}_1 \times \vec{R}_1 = \vec{H}_{G_1} + \vec{OG}_1 \times m_1 \cdot \vec{a}_1 \quad (2-17)$$

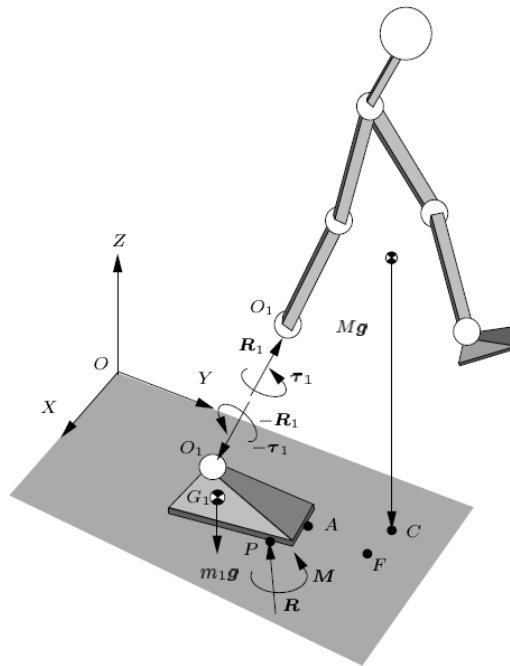


Figura 2-19 - Forças e momentos aplicados no pé de suporte (retirado de [4])

E retirando os termos dinâmicos, obtém-se a equação de equilíbrio estático do pé:

$$\vec{M} + \vec{OP} \times \vec{R} + \vec{OG}_1 \times m_1 \cdot \vec{g} - \vec{\tau}_1 - \vec{OO}_1 \times \vec{R}_1 = 0 \quad (2-18)$$

Onde \mathbf{R}_1 e $\boldsymbol{\tau}_1$ representam a força e binário do tornozelo, respectivamente, o que, visto a partir do pé de suporte descreve toda a dinâmica do resto do robô.

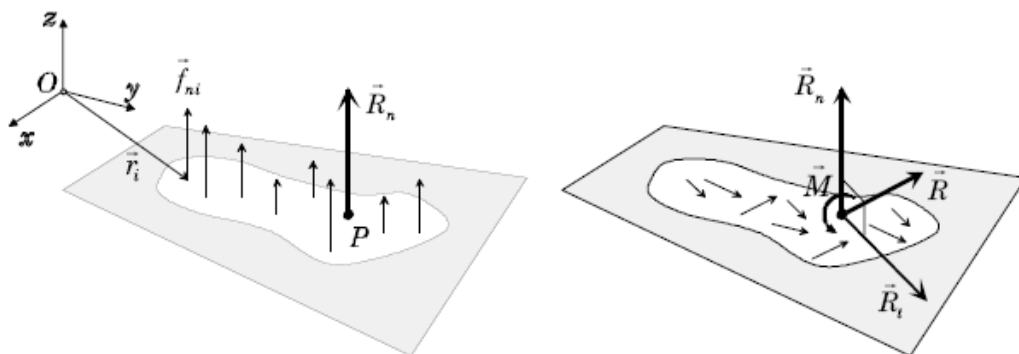


Figura 2-20 - Forças de reacção no pé (retirado de [8])

A força de reacção do solo \mathbf{R} combina a acção da gravidade sobre o sistema e as acelerações de todos os seus segmentos. Ou seja,

$$\vec{R} = \sum_{i=1}^n m_i \cdot (\vec{a}_i - \vec{g}) \quad (2-19)$$

Dado que a componente do momento resultante M , que representa a rotação do pé em torno da normal ao solo, não contribui para a perda de equilíbrio no plano sagital, pode-se ignorar, ficando a equação de equilíbrio estático do pé de suporte reduzida a:

$$\vec{PG}_1 \times m_1 \cdot \vec{g} - \vec{\tau}_1 - \vec{PO}_1 \times \vec{R}_1 = 0 \quad (2-20)$$

Nesta equação a referência passou a ser o ponto P e não o ponto O . Pode-se então derivar que segundo a componente xx , ao longo do plano sagital, o centro de pressão P é dado por:

$$P_x = \frac{\tau_1 - h \cdot R_t + G_{1x} \cdot m_1 \cdot g}{R_n} \quad (2-21)$$

Onde h é a distância que separa o plano do pé da junta do tornozelo.

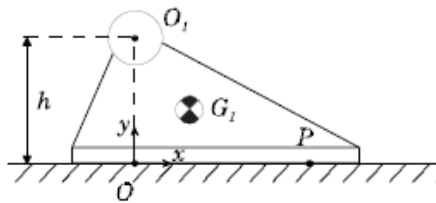


Figura 2-21 - Pormenor do pé (retirado de [8])

O CoP é ponto onde a resultante R_n actua. Como a resultante R_n é definida como somatório de todas as forças normais que actuaem no pé. Pode-se então afirmar que o CoP está delimitado pelo polígono de suporte.

Da mesma forma que o CoP, o ponto F correspondente ao *foot rotation indicator* (FRI), pode ser alcançado através da equação de equilíbrio estático do pé de suporte, ficando nesse caso coincidente com o ponto P :

$$\vec{FG}_1 \times m_1 \cdot \vec{g} - \vec{\tau}_1 - \vec{FO}_1 \times \vec{R}_1 = 0 \quad (2-22)$$

$$F_x = \frac{\tau_1 - h \cdot R_t + G_{1x} \cdot m_1 \cdot g}{R_n} \quad (2-23)$$

O FRI é definido como o ponto na superfície de contacto pé/solo, dentro ou fora do polígono de suporte, no qual o momento resultante da força/binário imposto no pé é normal à superfície. Quer isto dizer que se houver um momento não compensado sobre o pé, o CoP, por estar confinado ao polígono de suporte vai estar num dos limites deste polígono mas o FRI vai estar fora desses limites. Para a execução de movimentos que envolvam dinâmica rotacional do pé, como o caminhar realizado pelo ser humano, este ponto é de extrema importância ao indicar o sentido e o momento associado à rotação do pé [5].

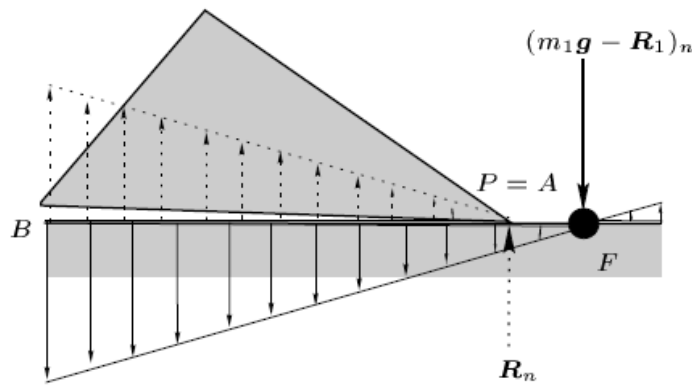


Figura 2-22 - Pé em rotação (retirado de [4])

Pode-se então enumerar algumas das propriedades do ponto FRI [8]:

- É um ponto que indica rotação no pé;
- A localização deste ponto indica a magnitude do momento não compensado no pé;
- Indica a direcção de rotação do pé;
- Indica a margem de estabilidade.

As coordenadas do CoP e do FRI são coincidentes nas situações em que $\vec{H}_{G1} + \vec{F}G_1 \times m_1 \cdot \vec{a}_1 = 0$, ou seja, nas situações em que:

- A aceleração angular do pé é zero $a_1=0$
- O pé tem massa e inércia nulas, $J_1=0$ e $m_1=0$.
- Os vectores $\vec{F}G_1$ e $m_1 \vec{a}_1$ são paralelos e $J_1=0$

Resumindo as características dos critérios de estabilidade [8]:

1. No caso de um robô estático e em equilíbrio, o CoP (ZMP), o FRI e o PVCoG encontram-se no polígono de suporte e são coincidentes;
2. Para configurações estacionários e instáveis, FRI e PVCoG encontram-se fora do polígono de suporte e são coincidentes;
3. Na presença de dinâmica e com o pé estacionário CoP (ZMP) e FRI são coincidentes, no interior do polígono de suporte e diferentes do PVCoG;
4. O CoP (ZMP) nunca pode deixar o polígono de suporte ao contrário do FRI que representa o ponto, dentro ou fora do polígono onde a força de reacção do solo deve actuar para manter o pé estacionário;
5. Com o FRI fora do pé de suporte indica que o pé está em rotação e quanto maior a distancia do FRI dos limites do pé maior o momento não balanceado.

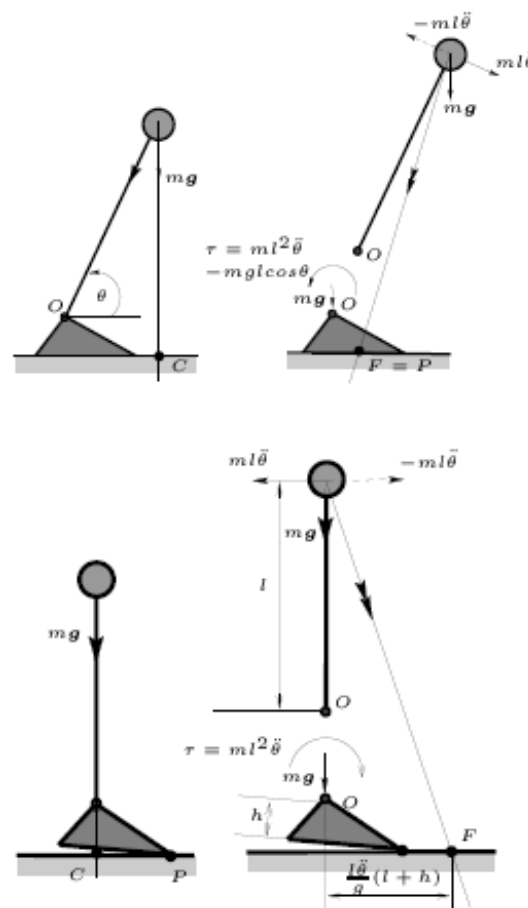


Figura 2-23 - Dois exemplos do PVCoG, CoP e FRI (retirado de [4])

2.4 Ferramentas Utilizadas

O projecto humanóide da UA (PhUA) está repleto de experiências efectuadas a um nível superior para validar, simular e muitas vezes actuar no robô humanóide. A criação da interface gráfica assenta, por isso, sobre muito trabalho desenvolvido durante os anos transactos. Como todo esse trabalho foi escrito e testado em *Matlab* e com o intuito de integrar todo esse esforço na interface gráfica decidiu-se criar essa mesma interface em *Matlab*. As razões para a escolha desta ferramenta de desenvolvimento recaem sobre as seguintes razões

- O protocolo de comunicações RS-232 estava desenvolvido em *Matlab* (**Apêndice B**);
- Muito do trabalho desenvolvido para este projecto estava em *Matlab*, entre os quais os algoritmos de padrões de locomoção, as definições e animações virtuais do humanóide, entre outros;

2.4.1 Matlab

O *Matlab* é um software de engenharia que permite efectuar cálculo numérico e possui uma linguagem de programação própria. O *Matlab* também fornece uma ferramenta para desenvolvimento de *Graphical User Interface (GUI)*, que se chama *GUIDE*.

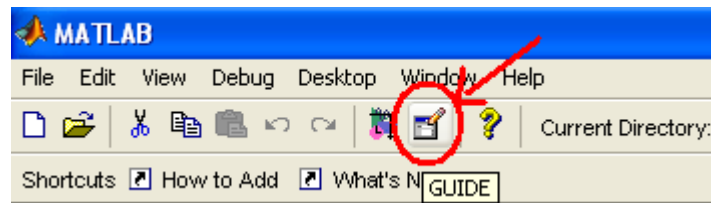


Figura 2-24 - GUIDE

Uma das desvantagens da ferramenta *GUIDE* consiste no pequeno leque de objectos de desenvolvimento que oferece mas é também este um dos factores que torna o desenvolvimento da aplicação gráfica fácil. Além dos objectos de desenvolvimento presentes na **Figura 2-25**, é também possível desenvolver uma barra de menu e/ou menus de contexto. O aspecto gráfico da ferramenta *GUIDE* está representado na seguinte ilustração:

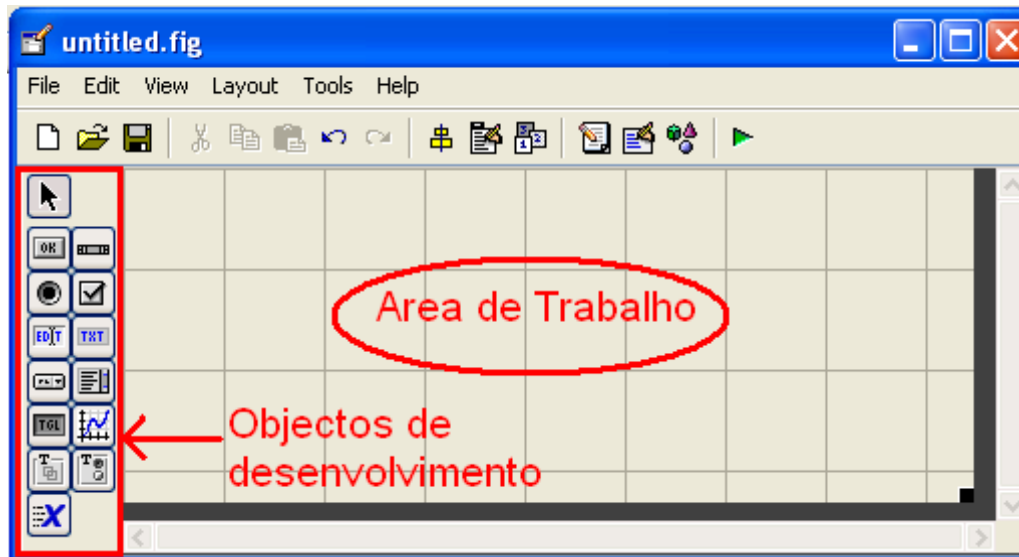


Figura 2-25 - Workspace GUIDE

O ambiente de desenvolvimento *GUIDE* permite modelar o aspecto visual da aplicação, mas não permite associar tarefas a objectos. Para isso, simultaneamente à criação visual, existe um *script* Matlab que alberga todas as funções relativas aos objectos da aplicação. Cada tipo de objecto tem um conjunto de chamadas ao sistema diferentes, isto é, a função que é despoletada por determinada acção. Por exemplo, clicar com o botão esquerdo do rato ou com o botão direito, são acções diferentes que podem encetar uma função diferente. Resumindo, há um *script Matlab* que alberga todas as funções desencadeadas por uma qualquer acção executada sobre o *GUI*.

A estrutura do *script*, inicia-se com código associado à iniciação e encerramento do *GUI*. Sempre que se cria uma nova função de *callback* esta tem presente nos seus parâmetros de entrada uma estrutura designada por *handles* e que contém o identificador de todos os objectos presentes no *GUI*. Através deste identificador e dos comandos *get* e *set*, pode-se aceder ou redefinir, respectivamente, parâmetros desse objecto:

- `get(h,'PropertyName')`
- `set(h,'PropertyName',PropertyValue,...)`

O *handles* é uma estrutura associada a uma figura, neste caso, essa figura é onde se apresenta o *GUI*. Há um conjunto de comandos que permitem aceder e trabalhar com esta estrutura, são eles:

- `guidata` – permite guardar ou ler dados associados a uma figura;
- `getappdata` – obtêm todos os dados ligados a uma figura;

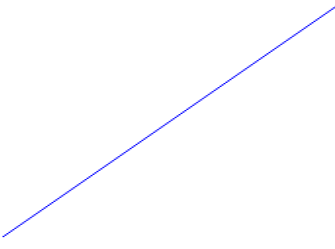
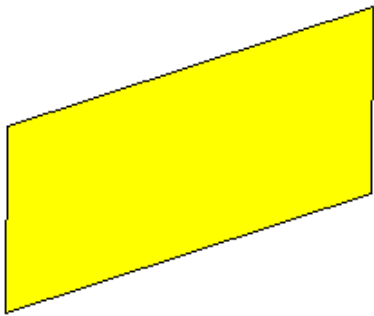
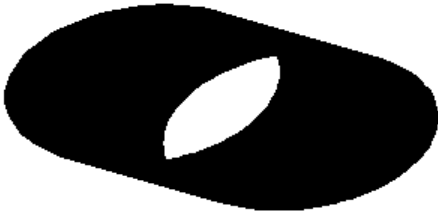
- *setappdata* – guarda os dados associados a uma figura.

Mais informação sobre o Matlab, com especial ênfase para o GUIDE e a criação de interfaces gráficas, pode-se consultar os documentos [15] [16] [17].

2.4.2 Rotinas de Desenho

O desenho do robô humanóide no Matlab é constituído por componentes elementares que resultam das funções *patch*, *surface* e *plot3*. Numa explicação superficial de cada uma das funções poder-se-á dizer que o *patch* é uma função de base do Matlab e serve para criar polígonos, sendo bastante versátil quanto à forma como se define o polígono. A função *surface* permite a criação de objectos 3D e a função *plot3* faz a conexão entre pontos vectoriais no espaço.

Está a seguir descrito cada um dos objectos elementares do desenho humanóide:

<p><u>LINHA:</u></p> <p>Var: x, y, z</p> <p>Func: <i>plot3(x, y, z)</i></p>	
<p><u>CHAPA:</u></p> <p>Var: $\begin{bmatrix} x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \\ z1 & z2 & z3 & z4 \end{bmatrix}$</p> <p>Func: <i>patch(var(1,:), var(2,:), var(3,:))</i></p>	
<p><u>CILINDRO</u></p> <p>Var: $[X,Y,Z] = \text{cylinder}(1,100);$</p> <p>Func: <i>surface(X, Y, Z);</i></p>	

<p><u>ESFERA</u></p> <p>Var: $[X,Y,Z] = sphere(100);$</p> <p>Func: $surface(X, Y, Z);$</p>	
---	--

Tabela 2-9 - Elementos do desenho humanóide

Todo o desenho do humanóide está assente nestes quatro elementos, que são replicados de forma a preencherem todas as necessidades e é aplicado a cada um deles a matriz de transformação relativa ao elo da cadeia cinemática humanóide onde ele está inserido. As especificações das dimensões de cada componente estão definidas no **Apêndice A**.

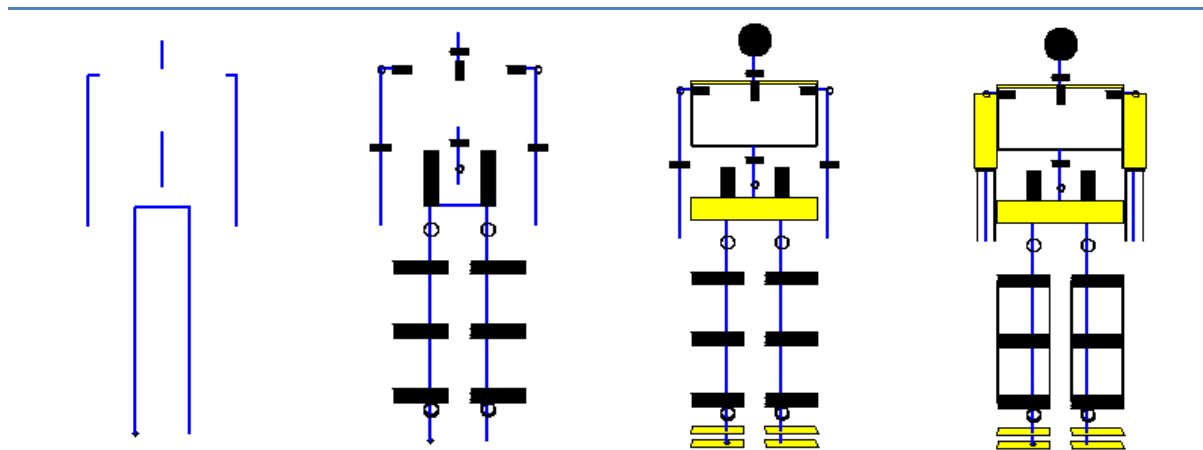


Figura 2-26 - Composição do desenho humanóide

2.4.3 Protocolo de Comunicações

O protocolo de comunicações entre o PC e o *master* baseia-se numa arquitectura cliente/servidor, implementada num barramento série RS232 [3]. Estão definidos em Matlab os *device drivers* que permitem a interacção com o robô, sem entrar em pormenores sobre a implementação do protocolo, indicam-se em seguida as funções disponibilizadas. As definições pormenorizadas encontram-se no **Apêndice B**.

Funções de iniciação e finalização de uma ligação	
<code>[handler, state]=initcom(gate, rate)</code>	Inicia uma comunicação através da porta série <i>gate</i> com um <i>baudrate</i> de <i>rate</i>
<code>stat=killcom(handler)</code>	Finaliza a comunicação identificada pelo <i>handler</i>

Tabela 2-10 - Device driver comunicações - Iniciar/Finalizar

Funções de teste	
<code>[error, errorstr]=testcom(H)</code>	Testa a validade da ligação identificada por <i>H</i>

Tabela 2-11 - Device driver comunicações – Teste

Funções de leitura	
<code>[servos, state, rx, error, errorstr]=readjoint(H, scu_id, param)</code>	Lê através da ligação identificada por <i>H</i> , o parâmetro <i>param</i> do <i>slave scu_id</i>
<code>[special, rx, error, errorstr]=readspecial(H, scu_id)</code>	Faz a leitura dos sensores especiais do <i>slave scu_id</i>

Tabela 2-12 - Device driver comunicações – Leitura

Funções de Actuação	
<pre>[rx,error,errorstr]=applyjoint(H,scu_id,param,servos)</pre>	Envia um parâmetro de actuação identificado por <i>param</i> com o valor <i>servos</i> para a <i>slave scu_id</i>
<pre>[rx,error,errorstr]=applycontrol(H,scu_id,param,servos)</pre>	Aplica um parâmetro do controlador identificado por <i>param</i> , com o valor <i>servos</i> para a <i>slave scu_id</i>

Tabela 2-13 - Device driver comunicações - Actuação

3 Estudo de Padrões de Locomoção

3.1 Introdução

Depois de em trabalhos anteriores se ter desenvolvido a estrutura mecânica, implementado os protocolos de comunicações aliados aos actuadores e sensores, de ter o robô humanóide completamente funcional, o passo seguinte consiste na aplicação de conceitos de coordenação. Um padrão de locomoção descreve o movimento coordenado das pernas através da definição de uma sequência de diferentes fases.

Os padrões de locomoção estudados para o robô humanóide são inspirados no ser humano. O processo de análise e modelação matemática do movimento praticado pelo ser humano é de cariz subjectivo, possibilitando a existência de várias abordagens ao mesmo problema. A descrição matemática desse padrão de locomoção é adaptada às características mecânicas do robô. Para o robô humanóide da UA estudaram-se um conjunto restrito de padrões de locomoção que permitem efectuar as tarefas mais elementares. A deslocação em linha recta e a mudança de direcção são as duas principais tarefas desenvolvidas. O movimento de pontapear uma bola, apesar do seu teor específico de uma modalidade desportiva, insere-se no conjunto de tarefas geralmente incluídas numa competição humanóide.

Neste capítulo pretende-se aplicar os vários conceitos abordados anteriormente ao robô humanóide no estudo dos padrões de locomoção. Para tornar o estudo progressivo, inicia-se uma análise a uma perna com 5 DoF, aprofundando alguns dos problemas de estabilidade de forma a extrapolar resultados para o robô humanóide com 22 DoF. Para o robô humanóide serão exemplificados alguns movimentos especificados quer no espaço das juntas como no espaço Cartesiano. Essa introdução servirá para depois expor as tarefas básicas desenvolvidas para o robô.

3.2 Planeamento de Trajectórias

Um robô humanóide, pela sua natureza, possui um grande número de graus de liberdade, tornando exigentes os problemas fundamentais associados com a sua modelação, programação, planeamento e controlo. Do ponto de vista da locomoção bípede, as questões fundamentais centram-se no planeamento de trajectórias e no controlo. Em termos gerais, pode-se dizer que os processos de planeamento geram os objectivos ou intenções de movimento, enquanto os processos de controlo asseguram que todos os aspectos do movimento planeado sejam levadas a cabo através da coordenação de um conjunto de acções.

Todavia, para um robô com estrutura similar ao humano, a especificação das tarefas não está limitada à posição e orientação do elemento terminal (caso dos manipuladores). Para estes robôs, a descrição da tarefa pode envolver a combinação de coordenadas associadas com os braços, as pernas, a cabeça e o tronco. Um dos tópicos de investigação na área da robótica humanóide é a coordenação do movimento destas estruturas no contexto da redundância cinemática¹. Considere-se portanto os seguintes pressupostos:

- Depois de saber como relacionar o espaço das juntas com o espaço operacional, é necessário proceder ao que se chama planeamento de trajectórias: definir a forma de variação das diversas juntas de forma que a estrutura cumpra os objectivos de movimento esperados.
- O planeamento de trajectórias é acompanhado pela especificação das trajectórias Cartesianas da secção da anca, da secção do tronco e do pé livre. Para eliminar a redundância do sistema impõem-se as seguintes restrições ao movimento: (i) a secção da anca deve manter uma postura fixa; (ii) a secção do tronco deve manter uma postura vertical; e (iii) o pé livre deve manter uma postura horizontal.
- Cada uma das fases necessárias para a realização de uma dada tarefa e a transferência entre elas apresentam dificuldades próprias que devem ser levadas em consideração. Durante a fase de suporte simples o voo do pé da perna livre deve apresentar uma trajectória tal que proporcione o adequado afastamento e aproximação ao solo. O problema do impacto é resolvido pelo planeamento apropriado da trajectória do pé livre. Ao mesmo tempo, assegura-se uma transição contínua entre as fases de SS e DS.
- É preciso impor um conjunto de condições sobre as velocidades e acelerações fronteiriças, de forma a assegurar a respectiva continuidade durante as transições entre fases. Os instantes de transição correspondem às situações em que as pernas alternam da fase de suporte para a fase aérea e vice-versa. Para uma transição e suporte suave é preciso garantir que as velocidades e acelerações angulares, antes e depois, sejam idênticas. Por outras palavras, o pé deve levantar e aterrar com velocidade e aceleração nula.
- O planeamento de trajectórias é baseado no movimento discreto entre dois pontos (movimento ponto-a-ponto) no espaço Cartesiano, exigindo um planeamento de trajectórias no espaço das juntas. Desde que as coordenadas Cartesianas desejadas para a anca, tronco e pé livre sejam conhecidas, então, todas as variáveis das juntas podem ser determinadas. Cabe ao gerador de trajectórias a tarefa de sincronizar e coordenar as pernas tendo por base a utilização de funções polinomiais.

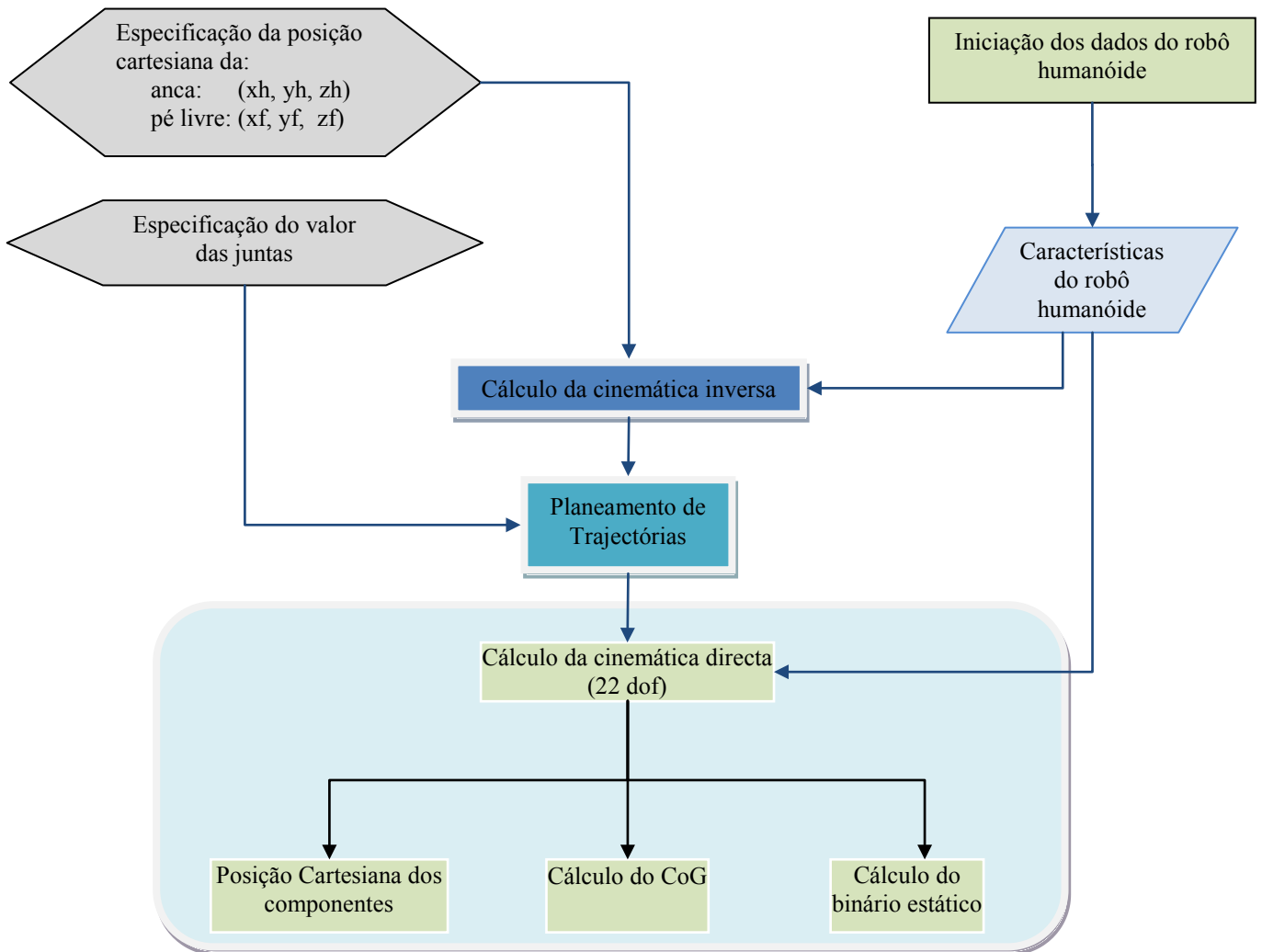
¹ Redundância é o termo usado em engenharia para descrever a existência de mais que uma solução para o mesmo problema

Para além da questão da coordenação de múltiplos graus de liberdade, o planeamento de trajectórias deve assegurar a estabilidade postural do robô. Na verdade, existe uma relação difícil entre o planeamento e a estabilidade devido às características de interacção que se estabelece entre o pé e o solo: equivalente a uma junta unilateral e não actuada [4]. Esta questão tem motivado a utilização de vários critérios de estabilidade quer na avaliação de trajectórias pré-planeadas como no controlo do processo de locomoção. Neste trabalho, assume-se uma velocidade de execução do movimento reduzida pelo que o critério de estabilidade usado será a PVCoG. Conhecidos os parâmetros de locomoção, uma questão que envolve o planeamento de trajectórias é a determinação daquelas que são fisicamente realizáveis:

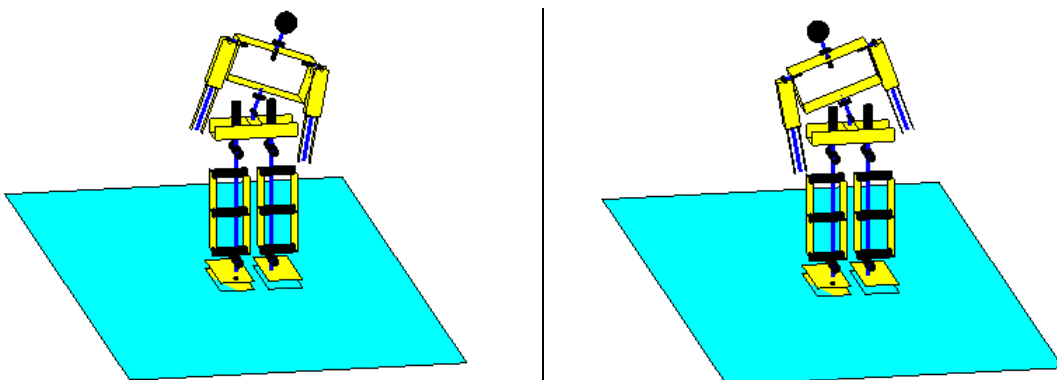
- Considera-se que durante todas as fases do movimento o sistema se encontra em equilíbrio estático (PVCoG dentro dos limites de suporte).
- Durante o período de tempo de duplo suporte o PVCoG deve mover-se de uma perna para a outra.

Partindo desta análise criou-se o simulador cinemático representativo do robô humanóide. O simulador cinemático é um modelo computacional capaz de calcular/analisar a resposta de um sistema. Para o robô humanóide as entradas podem ser especificadas como coordenadas Cartesianas ou posições angulares das juntas, o movimento executado é calculado e é avaliada a sua estabilidade.

Na **Figura 3-1** está representado o diagrama que descreve o simulador implementado. Há um momento inicial de criação dos parâmetros que definem o robô humanóide. Esta informação fica armazenada em memória para uso das rotinas de cinemática inversa e directa. O simulador é iniciado pela especificação de coordenadas Cartesianas relativas à anca e/ou ao pé livre ou então pela introdução de posições angulares finais das juntas. O movimento é traçado no espaço das juntas através do planeamento de trajectórias, onde é gerada uma matriz temporal das posições das juntas em cada instante. Esta matriz é enviada para o algoritmo de cinemática directa, que calcula as matrizes homogéneas relativas a essas posições. Estas matrizes, uma por cada junta do robô, são usadas para o cálculo Cartesiano de cada um dos componentes do robô, para o cálculo do CoG e para o cálculo do binário estático.



Na especificação de tarefas no espaço das juntas é definido directamente o valor final que cada junta deve tomar, sendo o movimento de todas as juntas executado de forma síncrona. As figuras seguintes ilustram algumas configurações executadas com aplicação directa do valor da junta.



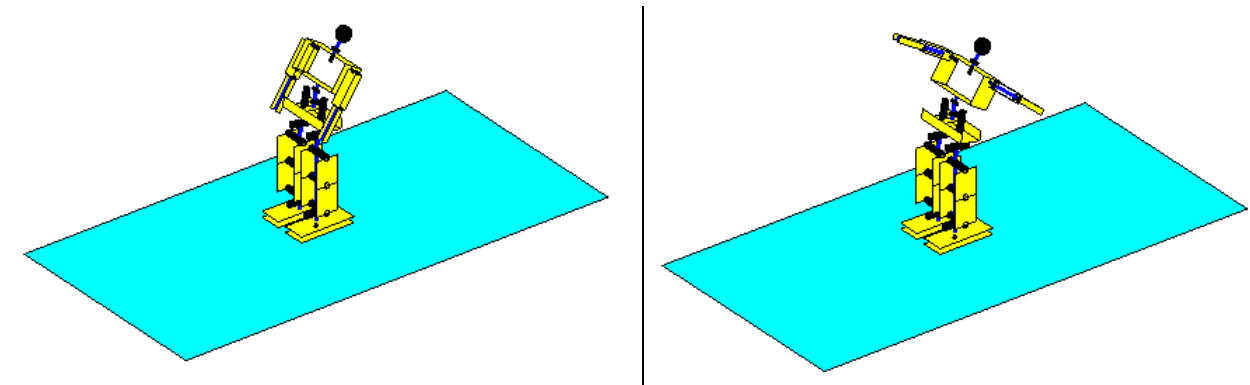


Figura 3-2 - Especificação de tarefas no espaço das juntas

Para entender a problemática explicada nesta secção convém introduzir os termos usadas para designar os planos do corpo humano: plano sagital, plano frontal e plano transversal (Figura 3-3).

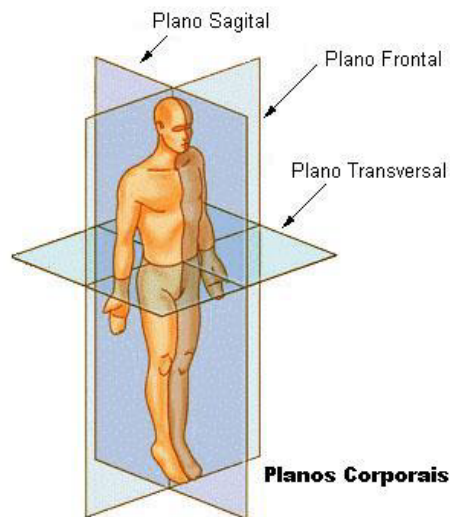


Figura 3-3 - Planos do corpo humano

3.3 Modelação de uma perna com 5 DoF

Considere-se uma perna com 5 DoF, genericamente similar às pernas do robô humanóide, tal como se ilustra nas Figura 3-4 e Figura 3-5. Os comprimentos dos elos e respectivas massas estão na Tabela 3-1.

Componente	Comprimento (m)	Massa (kg)
Pé-tornozelo	0.020	0.28
Tornozelo-tornozelo	0.022	0.20
Tornozelo-joelho	0.160	0.63
Joelho-anca1	0.120	0.26
Anca1-anca2	0.053	0.21
Anca2-Anca3	0.175	5.50

Tabela 3-1 – Dimensões dos elos para a perna 5 DoF

Considere-se que o centro de massa de cada componente se situa a meio do seu elo.

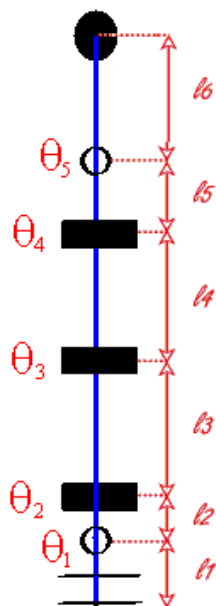


Figura 3-4 - Perna 5 DoF (vista de frente)

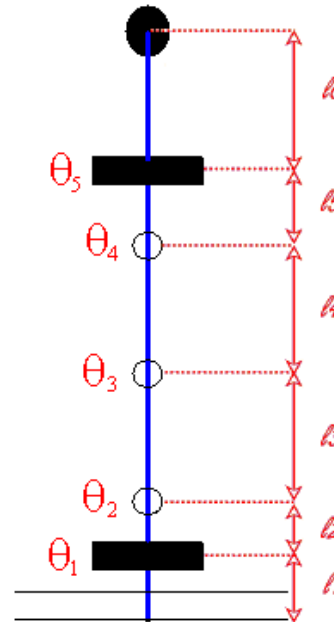


Figura 3-5 - Perna 5 DoF (vista de lado)

3.3.1 Análise Cinemática

Cinemática Directa

A cinemática directa assenta num cálculo iterativo que se inicia na base da estrutura (pé de suporte) e passa por todos os elos até chegar ao fim da cadeia, órgão terminal. É possível encontrar as equações de cinemática directa de qualquer estrutura semi-rígida, utilizando somente conceitos de trigonometria e geometria mas torna-se uma tarefa entediante e complexa para um sistema com muitos graus de liberdade. A obtenção das equações de cinemática usadas neste trabalho baseou-se no algoritmo de Denavit-Hartenberg (*DH*). O uso da representação de Denavit-Hartenberg permite obter as equações de cinemática directa de qualquer cadeia cinemática de forma simples e sistemática, sendo bastante utilizada em robótica. No caso do robô humanóide PhUA, só existem juntas rotacionais, as quais permitem apenas um grau de liberdade, assim, a acção de cada junta é descrita pelo seu ângulo de rotação. A cinemática directa determina o efeito cumulativo do conjunto das variáveis das juntas reflectido no elo final.

Na tabela seguinte está exposta a lista extensiva dos parâmetros de DH¹ aplicados à perna 5 DoF e os referenciais obtidos para o caso de todas as juntas estarem a zero na **Figura 3-6**.

Elemento	Número	θ	a	d	α
Pé	2	θ_1	0	l_2	90
Tornozelo	3	θ_2	0	l_3	0
Joelho	4	θ_3	0	l_4	0
Anca1	5	θ_4	0	l_5	-90
Anca2	6	θ_5	0	l_6	0

Tabela 3-2 - Parâmetros de DH para a perna 5 DoF

O sistema de eixos número θ é o sistema de referência global, externo ao robô. O sistema de eixos número 1, como não é influenciado por nenhuma junta, é obtido através da seguinte matriz homogênea:

$$A(:, :, 1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-1)$$

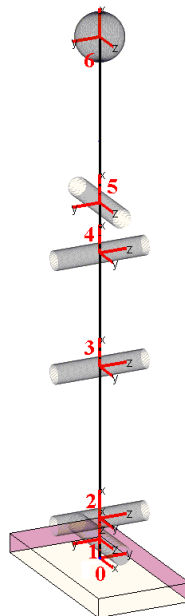


Figura 3-6 - Sistema de eixos para a perna 5 DoF (vista de frente)

¹ O algoritmo de Denavit-Hartenberg está explicado no Apêndice C.

Cinemática Inversa

Depois de introduzidos na **secção 2.3.2** os métodos possíveis para o cálculo da cinemática inversa, é altura de aplicar as soluções apresentadas. A perna 5 DoF envolve algumas peculiaridades, que não se encontram, por exemplo, num braço robótico. A não existência de uma base fixa e múltiplos pontos de apoio possíveis, são características relevantes na avaliação e escolha de um algoritmo de cinemática inversa, que se quer fiável e consistente. Como para este caso particular só há um pé de suporte possível, define-se dessa forma a coordenada Cartesiana do ponto terminal, a anca, como sendo o objectivo a atingir:

1. H_h - Altura da anca (posição em zz);
2. Y_g - Posição lateral da anca (posição em yy);
3. X_g - Posição sagital da anca (posição em xx).

Supondo a seguinte configuração da perna, observada de trás, define-se a vermelho o plano sagital do pé de suporte e a verde o plano frontal.

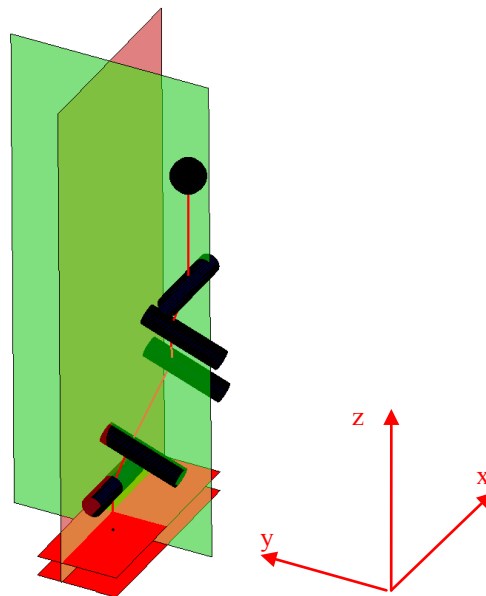


Figura 3-7 - Exemplo da perna de suporte para análise cinemática (vista de trás)

A primeira abordagem é encontrar os planos funcionais de cada junta. Dessa forma, nas **Figura 3-8** e **Figura 3-9**, pode observar-se o plano frontal e o plano de acção das juntas “sagitais”.

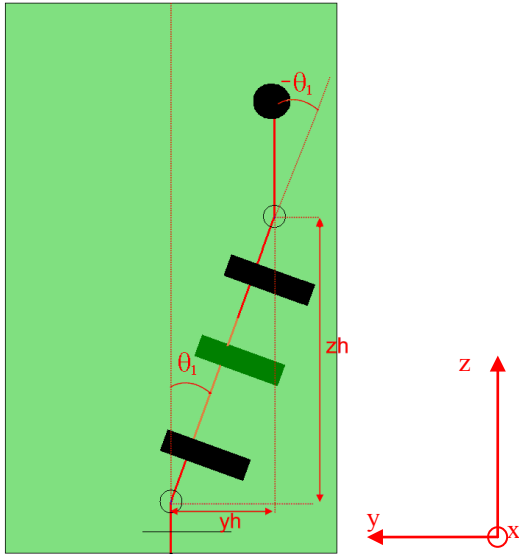


Figura 3-8 - Plano frontal (vista de trás)

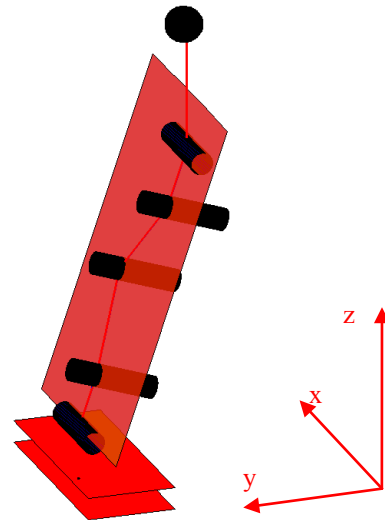


Figura 3-9 - Plano de funcionamento das juntas de ação "sagital" (vista de trás)

Fica assim presente a ideia de que a junta 1 e 5 actuam no plano frontal, **Figura 3-8**, enquanto as juntas 2, 3 e 4 actuam no plano sagital (adaptado ao referencial do tornozelo), **Figura 3-9**.

A primeira e única restrição a apontar é a postura do tronco e do pé livre. Dessa forma, pela análise da **Figura 3-8**, têm-se um zh e um yh que se pode derivar directamente da posição desejada para a anca.

$$\begin{aligned} yh &= Yg \\ zh &= Hh - (l_1 + l_6) \end{aligned} \quad (3-2)$$

Para determinar θ_1 ,

$$\theta_1 = \text{atan}\left(\frac{yh}{zh}\right) \quad (3-3)$$

Como θ_5 é obrigado a corrigir a inclinação do tronco, introduzida por θ_1 , então,

$$\theta_5 = -\theta_1 \quad (3-4)$$

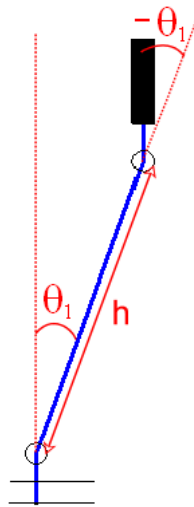


Figura 3-10 - Equivalente do problema no plano frontal

Na **Figura 3-10**, está clarificado o problema já resolvido, resta agora voltar atenções para o h que é dado por,

$$h = \sqrt{zh^2 + yh^2} \quad (3-5)$$

E definido no plano das juntas sagitais, tal como demonstrado na **Figura 3-11**. Pode-se reparar que este é o típico problema 3R resolvido pelo método de cinemática inversa introduzido anteriormente, com um θ_d zero, para garantir o tronco na vertical. A redundância do cotovelo é definida de acordo com a direcção em que o joelho dobra. O ponto (x_1, y_1) representado na figura é dado por:

$$\begin{aligned} x_1 &= Xg \\ y_1 &= h - (l_2 + l_5) \end{aligned} \quad (3-6)$$

Estes valores têm entrada directa no algoritmo de cálculo 3R obtendo-se assim θ_2 , θ_3 e θ_4 .

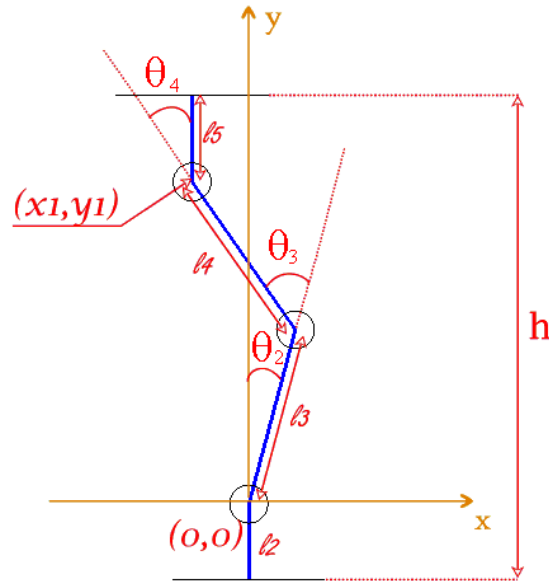


Figura 3-11 - Problema nas juntas "sagitais"

Resumindo, o algoritmo de cinemática inversa usado pode ser descrito, especificando \$(X_g, Y_g, H_h)\$, nos seguintes passos:

1. Com base na imposição da postura vertical do tronco e pé livre calcula-se \$y_h\$ e \$z_h\$;
2. Através destes dois parâmetros (\$y_h\$ e \$z_h\$) calcula-se \$\theta_1\$ e \$\theta_5\$;
3. Adicionando a abcissa (\$x_h=X_g\$), usa-se o algoritmo de cinemática inversa 3R para calcular \$\theta_2, \theta_3\$ e \$\theta_4\$.

3.3.2 Análise Estática e Dinâmica

Conhecidas as coordenadas no referencial local (**subsecção 2.2.1**), usa-se a multiplicação sucessiva das matrizes de transformação homogênea para obter as coordenadas no referencial global. O cálculo do CoG global da perna 5 DoF corresponde a

$$\mathbf{CoG} = \frac{1}{M_T} \cdot \sum_{i=1}^5 m_i \cdot \mathbf{CoG}_i = \begin{cases} CoGx = \frac{1}{M_T} \cdot \sum_{i=1}^5 m_i \cdot CoGx_i \\ CoGy = \frac{1}{M_T} \cdot \sum_{i=1}^5 m_i \cdot CoGy_i \\ CoGz = \frac{1}{M_T} \cdot \sum_{i=1}^5 m_i \cdot CoGz_i \end{cases} \quad (3-7)$$

Onde M_T corresponde à massa total da estrutura, m_i é a massa de cada um dos componentes, CoG_i é o centro de massa de cada componente, e o CoG é o vector do centro de massa global, (CoG_x , CoG_y , CoG_z).

A força de reacção do solo é dada pela seguinte fórmula

$$\mathbf{R} = \sum_{i=k}^5 m_i \cdot (\mathbf{a}_i + \mathbf{g}) \quad (3-8)$$

Onde \mathbf{a}_i é a aceleração linear do CoG do elo i . O binário em cada junta é dado pela expressão geral (3-9) ([10]).

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{B}(\dot{\mathbf{q}}) \quad (3-9)$$

- $\boldsymbol{\tau}$ é o vector coluna(n-dimensional) dos binários fornecidos pelos actuadores;
- $\mathbf{H}(\mathbf{q})$ é a matriz quadrada (de ordem n) dos coeficientes inerciais (simétrica e definida positiva);
- $\mathbf{C}(\mathbf{q}, d\mathbf{q})$ é o vector coluna (n-dimensional) das forças Coriolis/centríptas;
- $\mathbf{G}(\mathbf{q})$ é o vector coluna (n-dimensional) das forças gravitacionais;
- $\mathbf{B}(d\mathbf{q})$ é o vector que descreve os efeitos de atrito o espaço das juntas.

O cálculo do CoP (ZMP), discutido na **subsecção 2.3.3**, é efectuado através das seguintes fórmulas:

$$P_x = \frac{-(M_{01})_y - h \cdot R_x - m_0 \cdot CoG_0 \cdot g}{R_z} \quad (3-10)$$

$$P_y = \frac{(M_{01})_x - h \cdot R_y - m_0 \cdot CoG_0 \cdot g}{R_z} \quad (3-11)$$

$$\mathbf{M}_{01} = \sum_{k=1}^5 m_k \cdot (\mathbf{r}_k \times (\mathbf{a}_k + \mathbf{g})) \quad (3-12)$$

Onde P_x e P_y são a abcissa e ordenada, respectivamente do CoP , \mathbf{M}_{01} é o momento no pé, h a distância do polígono de suporte ao tornozelo, R_x , R_y e R_z as componentes da força de reacção do solo, m_i a massa do elemento i , \mathbf{g} a aceleração da gravidade, \mathbf{r}_k a posição da junta k e \mathbf{a}_k a aceleração linear do CoG do elo k .

3.3.3 Resultados da Simulação

Realizou-se um estudo com a especificação de alguns movimentos para a perna. O planeamento da trajetória foi definido no espaço das juntas, usando o mesmo polinómio de grau 5 que é aplicado aos servomotores do robô humanóide e apresentado na **subsecção 2.2.1**. O cálculo dos binários foi realizado recorrendo à *toolbox* simbólica do Matlab, exigindo bastante tempo de processamento e impossibilitando qualquer tentativa de fazer esta simulação *on the fly*.

Com esta experiência pretendia-se averiguar o grau de disparidade entre o PVCoG e o CoP para um movimento com uma velocidade média da anca de 1cm/s. A definição de velocidade é dada em termos médios porque ela obedece a uma variação descrita por um polinómio de grau 4.

O primeiro movimento está ilustrado na **Figura 3-12**, observando-se aí os resultados simulados. Pode verificar-se que o PVCoG e CoP são coincidentes ao longo de todo o movimento.

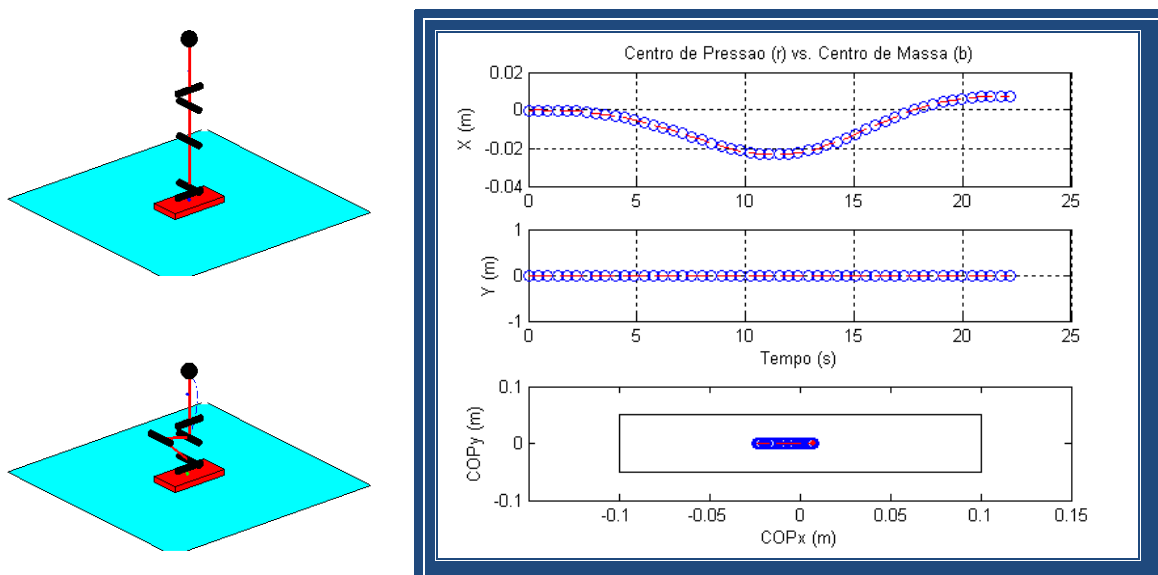


Figura 3-12 - 1º movimento – baixar

O resultado anterior foi obtido em praticamente todas as situações simuladas. Em seguida mostram-se todos os outros movimentos e os seus resultados, **Figura 3-13**, **Figura 3-14**, **Figura 3-15** e **Figura 3-16**.

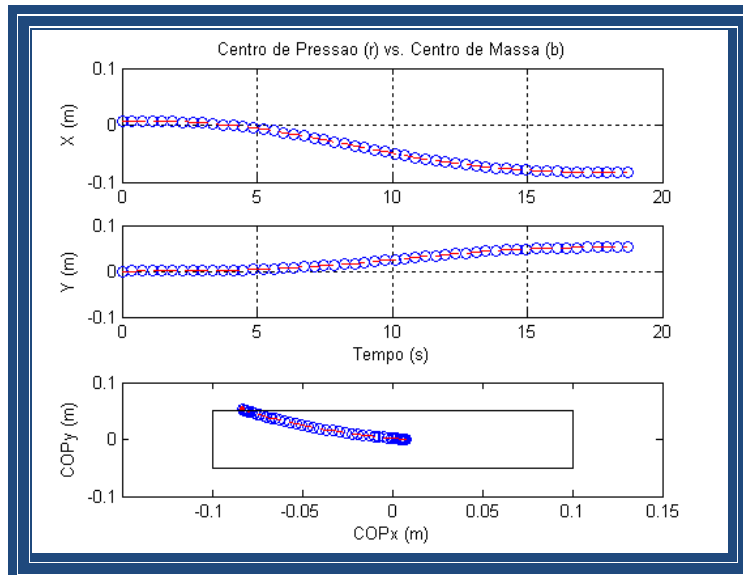
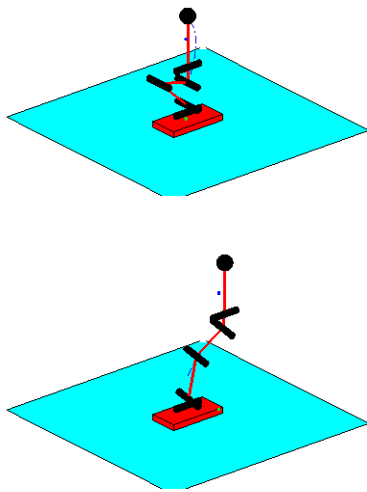


Figura 3-13 - 2º movimento

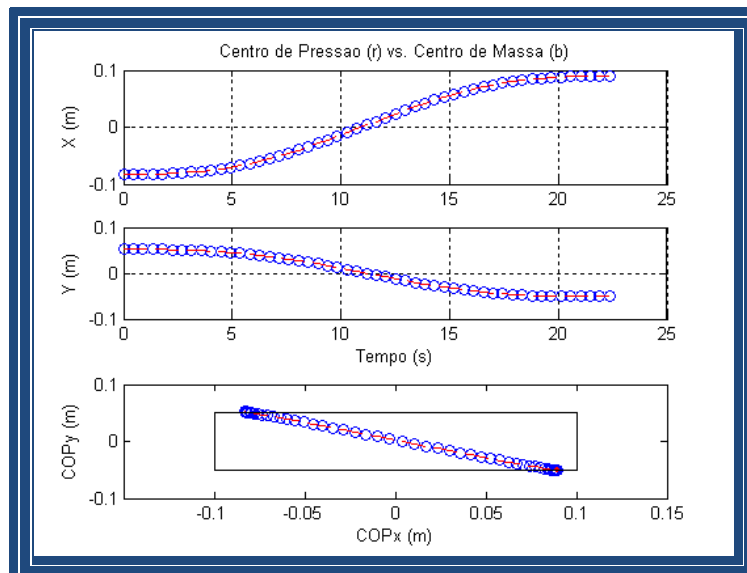
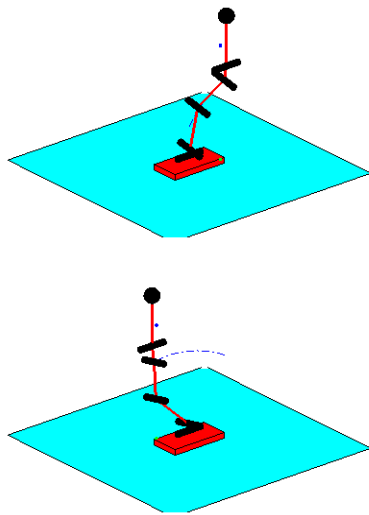


Figura 3-14 - 3º movimento

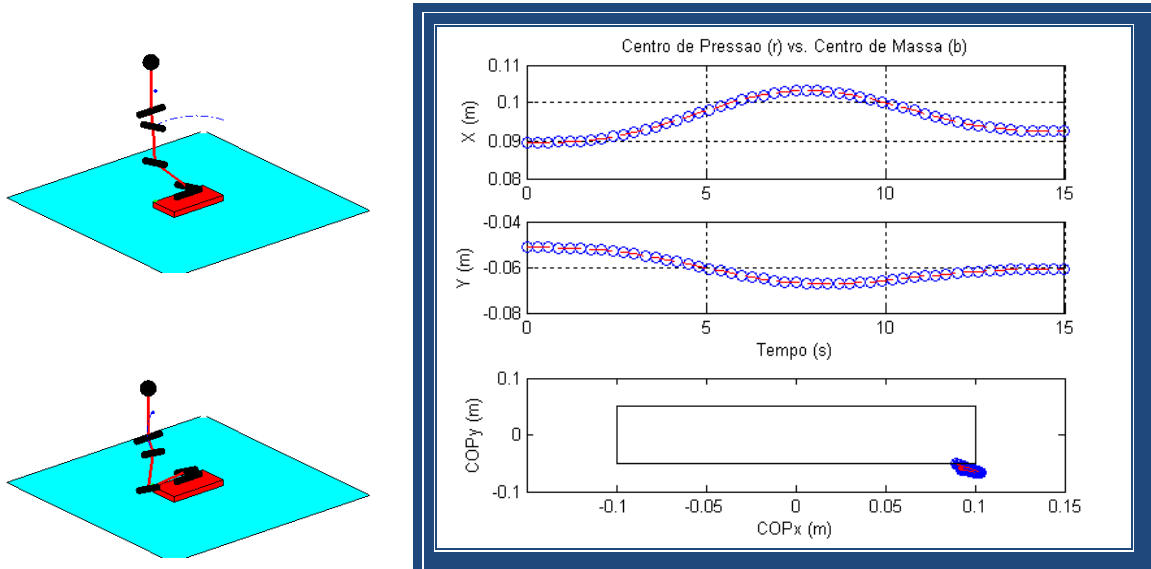


Figura 3-15 - 4º movimento

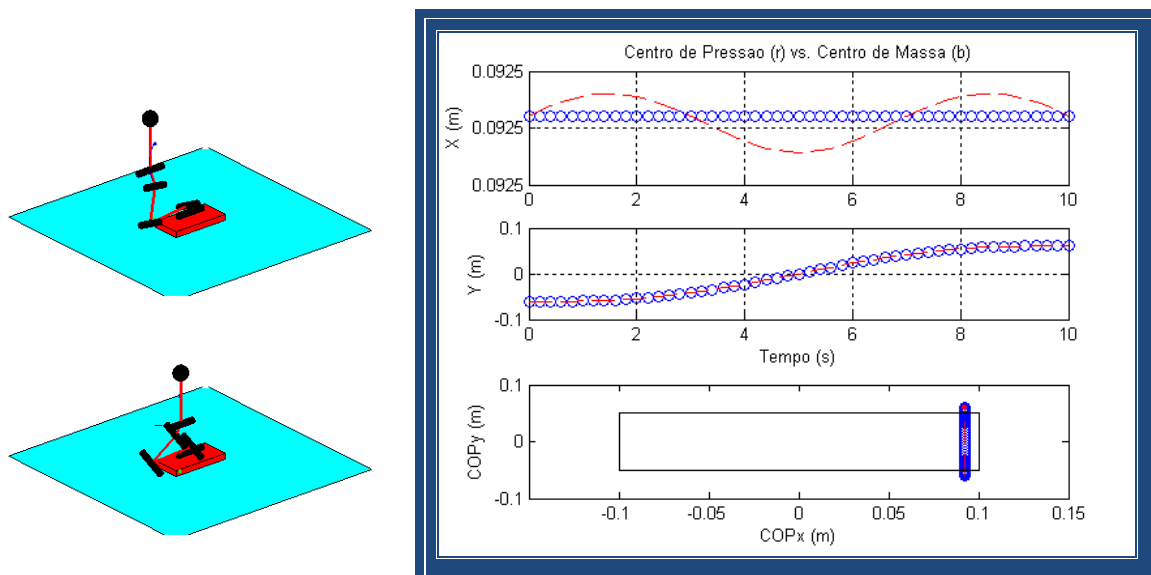


Figura 3-16 - 5º movimento

Esta experiência serviu para mostrar que o uso do PVCoG como critério de estabilidade para movimentos lentos é uma boa aproximação ao CoP. Portanto, dada a dificuldade em calcular o CoP em tempo-real para toda a estrutura humanóide é possível, desde que se garanta um movimento lento, o uso do PVCoG como critério de estabilidade. Esta parece ser uma boa aproximação do que se passaria se estivesse a ser usado o robô completo.

3.4 Modelação da Estrutura com 22 DoF

3.4.1 Análise Cinemática

Antes de entrar nos detalhes dos algoritmos cinemáticos aplicados ao robô 22 DoF convém relembrar a numeração das juntas do robô humanóide e o sistema de coordenadas de referência (Figura 3-17 e Figura 3-18).

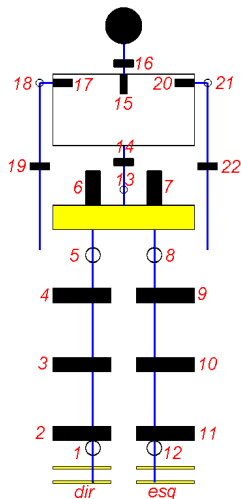


Figura 3-17 - Número de cada junta no simulador (vista de frente)

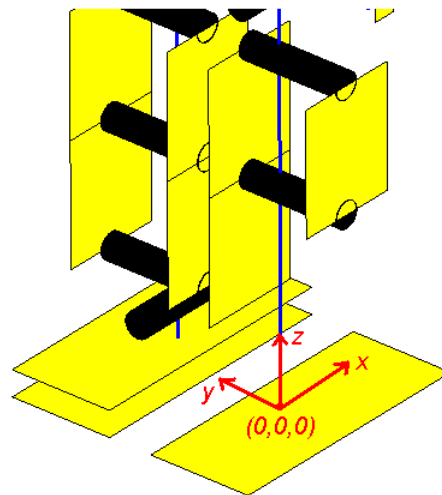


Figura 3-18 - Sistema de coordenadas de referência (vista de trás)

Cinemática Directa

O robô humanóide da Universidade de Aveiro, possui 22 graus de liberdade (DoF) mas dadas as condições impostas pela representação de DH (**Apêndice C**), criaram-se 30 sistemas de eixos, os quais estão representados na **Figura 3-22**. Na tabela seguinte está exposta a lista extensiva dos parâmetros de DH aplicados ao robô humanóide:

Elemento	Número	θ	a	d	α
Pé direito	2	θ_1	0	L(2)	90
Tornozelo direito	3	θ_2	0	L(3)	0
Joelho direito	4	θ_3	0	L(4)	0
Anca1 direita	5	θ_4	0	L(5)	-90
Anca2 direita	6	θ_5	0	L(6)	0
	7	-90	0	0	-90
Anca3 direita	8	θ_6-180	0	-L(8)	0
Anca3 esquerda	9	θ_7	-L(6)	0	-90
Anca2 esquerda	10	θ_8+90	0	L(5)	90

Elemento	Número	θ	a	d	α
Anca1 esquerda	11	θ_9	0	L(4)	0
Joelho esquerdo	12	θ_{10}	0	L(3)	0
Tornozelo esquerdo	13	θ_{11}	0	L(2)	-90
Pé esquerdo	14	θ_{12}	0	L(1)	0
	15	θ_6-180	L(7)	-L(8)/2	-90
	16	-90	0	L(9)	0
Cinta	17	θ_{13}	0	L(10)	90
Tronco	18	θ_{14}	0	L(11)	0
	19	0	0	L(12)+L(13)	0
	20	90	0	0	90
Pescoço	21	θ_{15}	L(14)	0	-90
Cabeça	22	$\theta_{16}-90$	0	L(15)	0
	23	θ_{14}	-L(8)/2-L(16)	L(11)+L(12)	180
Ombro direito	24	θ_{17}	L(17)	0	90
Braço direito	25	$\theta_{18}-180$	0	L(18)	90
Antebraço direito	26	θ_{19}	0	L(18)	0
	27	θ_{14}	L(8)/2+L(16)	L(11)+L(12)	0
Ombro esquerdo	28	θ_{20}	L(17)	0	-90
Braço esquerdo	29	$\theta_{21}-180$	0	L(18)	-90
Antebraço esquerdo	30	θ_{22}	0	L(18)	0

Tabela 3-3 – Parâmetros de DH para a estrutura 22 DoF

Podem-se observar os eixos de ligação entre os diferentes componentes do robô, representados pelo campo “Elemento” vazio. Os θ_i representam o valor da junta i , correspondendo i às juntas da **Figura 3-19**. Os comprimentos correspondem aos elos obtidos na **secção 2.2.1**.

Da mesma forma que para a análise da perna 5DoF, também neste caso o sistema de eixos número 0 é o sistema de referência global e o sistema número 1 é obtido através da seguinte matriz homogênea:

$$A(:, :, 1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & L(1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-13)$$

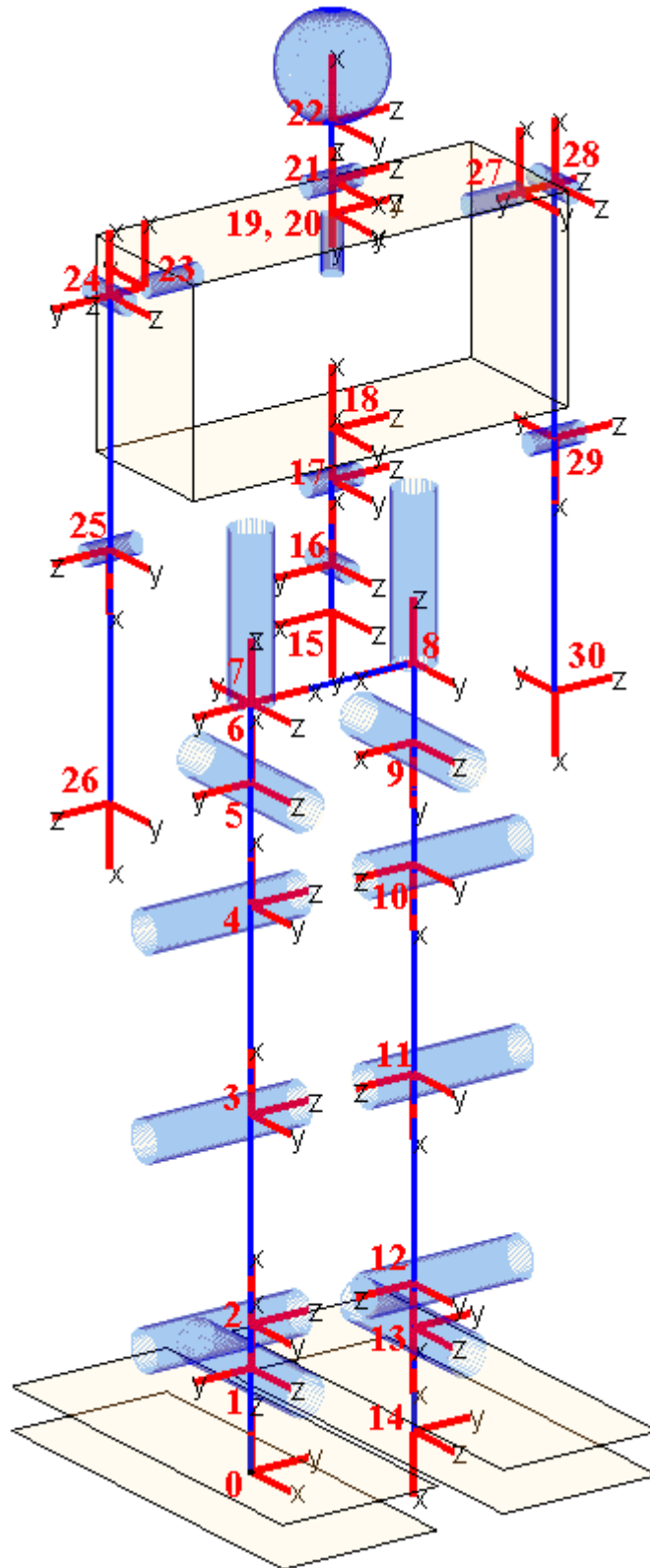


Figura 3-19 - Sistemas de eixos usados (vista de frente)

Cinemática Inversa

Como já foi referido os métodos analíticos apresentam vantagens quando se pretende um cálculo rápido e directo, mas a parametrização do problema torna a tarefa complicada numa estrutura humanóide com 22 DoF. O robô humanóide pode ser dividido em várias sub-cadeias cinemáticas, tais como a da perna direita, da perna esquerda, do braço direito, etc. Como a aplicação da cinemática inversa se resume aos padrões de locomoção analisados mais à frente nesta dissertação e estes, por sua vez, aplicam-se exclusivamente às pernas do robô humanóide, a aplicação da cinemática inversa assenta sobretudo nas pernas.

O problema da cinemática inversa para uma perna já foi apresentado na secção anterior. Extrapolando os resultados dessa análise para o caso de duas pernas, pode-se criar dois algoritmos de cinemática inversa, um para a perna de suporte, e outro para a perna livre. Observa-se, pela **Figura 3-17** que as juntas 1, 5, 8 e 12 operam no plano frontal e as juntas 2, 3, 4 e 9, 10, 11, no plano sagital. No caso de se efectuar um cálculo de cinemática inversa para uma das pernas, as juntas que se movem no plano frontal só interferem com o movimento das juntas que se movem no plano sagital segundo zz , quer isto dizer, que quando há uma especificação Cartesiana para uma perna tem-se dois graus de liberdade para controlar o movimento em yy (juntas do plano frontal) e 3 juntas controlam a posição segundo xx (juntas do plano sagital).

Pode-se observar em detalhe nas **Figura 3-20** e **Figura 3-21** a configuração das pernas.

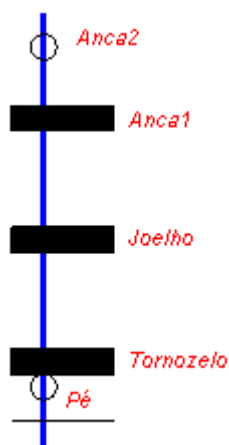


Figura 3-20 - Detalhe da perna (vista frontal)

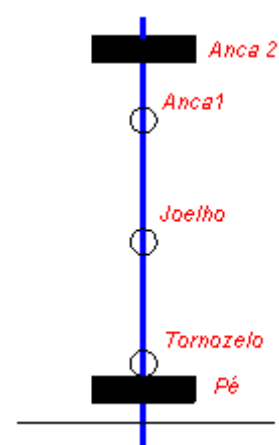


Figura 3-21 - Detalhe da perna (vista lateral)

A cinemática inversa da perna de suporte é invocada quando se impõe uma posição Cartesiana para a anca. Os parâmetros de entrada são dados pelas seguintes variáveis:

1. Hh - Altura da anca;
2. Yg - Posição lateral da anca;
3. Xg - Posição sagital da anca.

O processo de cálculo cinemático para a perna livre é em tudo idêntico ao usado para a perna de suporte. As diferenças residem na referência usada, já que quando se fala de perna de suporte, o ponto fixo é o pé e no caso da perna livre o ponto fixo será a anca. Como a junta 6 não é utilizado nestes cálculos pode-se aproveitar a simetria vertical e fazer uma inversão vertical dos esquemas usados, adaptar os comprimentos dos elos obtendo a cinemática inversa para a perna livre. Da mesma forma que para a perna de suporte, definem-se os mesmos parâmetros de entrada para a perna livre, mas com uma interpretação ligeiramente diferente,

1. Hh - Distância da anca ao pé livre;
2. Yg - Posição lateral do pé livre;
3. Xg - Posição sagital do pé livre.

Análise Estática

Para a determinação do CoG da estrutura humanóide usa-se a mesma fórmula do caso da perna 5 DoF, mas expandida para os 22 DoF. Dessa forma,

$$\mathbf{CoG} = \frac{1}{M_T} \cdot \sum_{i=1}^{22} m_i \cdot \mathbf{CoG}_i = \begin{cases} CoGx = \frac{1}{M_T} \cdot \sum_{i=1}^{22} m_i \cdot CoGx_i \\ CoGy = \frac{1}{M_T} \cdot \sum_{i=1}^{22} m_i \cdot CoGy_i \\ CoGz = \frac{1}{M_T} \cdot \sum_{i=1}^{22} m_i \cdot CoGz_i \end{cases} \quad (3-14)$$

Onde M_T corresponde à massa total da estrutura, m_i é a massa de cada um dos componentes, \mathbf{CoG}_i é o centro de massa de cada componente, e o \mathbf{CoG} é o ponto do centro de massa global, (CoG_x , CoG_y , CoG_z).

Neste caso a análise dinâmica é reduzida ao cálculo dos termos gravitacionais de acordo com a seguinte expressão:

$$\tau_k = \sum_{i=k}^{22} m_i \cdot (\text{CoG}_i - \mathbf{r}_k) \cdot \mathbf{R}_k \times \mathbf{g} \quad (3-15)$$

Onde m_i é a massa individual de cada elo, CoG_i é o centro de massa de cada elo, \mathbf{r}_k é a posição da junta k, \mathbf{R}_k é o vector que descreve a orientação da junta k e \mathbf{g} a aceleração da gravidade.

3.5 Configurações Simples

Entra-se agora no estudo de movimentos simples e recorrentes na definição de outros mais complexos. Nesta análise não se vai entrar em linha de conta com as equações de cinemática inversa estudadas, porque dada a simplicidade destas tarefas é possível definir estes movimentos no espaço das juntas. É claro que a assunção desta afirmação é baseada na própria cinemática inversa. Assume-se que $l_3=l_4$ e que há simetria sagital na estrutura.

3.5.1 Baixar Anca

Passando à prática, baixar a anca significa actuar nas juntas que actuam no plano sagital. Como neste movimento não se pretende interferir com a postura do tronco, o abaixamento torna-se a tarefa de encolher o h que separa a junta 2 da junta 4. Como o robô humanóide possui o l_3 igual ao l_4 , então quer dizer que para anular qualquer deslocação de θ_2, θ_3 precisa de andar o dobro em sentido oposto. Nesta fase da explicação há que deixar claro que o deslocamento da anca não é definido no plano Cartesiano mas sim definido como um incremento a aplicar às juntas. A **Figura 3-22** ilustra a situação e assumindo que se vai dar um incremento θ à junta 2, então a função de baixar é dada por:

$$\begin{aligned} \theta_2 &= \theta_2 + \theta \\ \theta_3 &= \theta_3 - 2 \cdot \theta \\ \theta_4 &= \theta_4 + \theta \end{aligned} \quad (3-16)$$

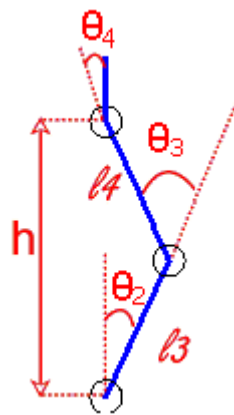


Figura 3-22 - Juntas de controlo da função de baixar a anca

O movimento executado por este algoritmo garante a postura do tronco mas no caso de as juntas 2 e 4 não estarem alinhadas segundo o eixo zz , a abcissa e ordenada da anca (e tronco), pode sofrer alteração durante o movimento. Isto é, se o eixo que liga as duas juntas não for perpendicular a xx ou yy a abcissa e/ou a ordenada, respectivamente, vão ser alteradas com o movimento.

Como as juntas são incrementais com um incremento mínimo de 1 grau, então, sendo θ_3 a junta que sofre maior deslocação vai ter que se limitar o valor mínimo do incremento deste movimento com base no incremento desta junta. Quer isto dizer que o θ mínimo que se pode aplicar é de 2 graus.

3.5.2 Balançar Anca

O movimento lateral da anca, **Figura 3-23**, envolve a deslocação da posição da anca em yy , mantendo a sua posição em xx fixa. É pois um movimento realizado no plano frontal, para isso são usadas as juntas de acção no plano frontal, 1, 5, 8, 12.

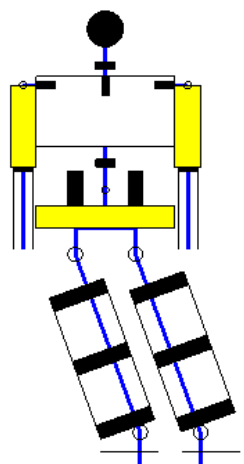


Figura 3-23 - Movimento de lateral

Nas figuras seguintes está ilustrado o sentido de rotação das juntas no caso do pé de suporte ser o direito. Ter em atenção, que para a compreensão desses sentidos se começa a definir o movimento a partir do pé de suporte (direito neste caso).

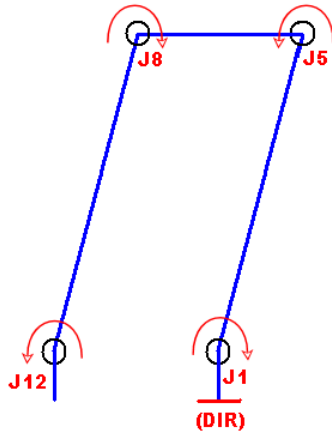


Figura 3-24 – Mov. lateral direita (visto de trás)

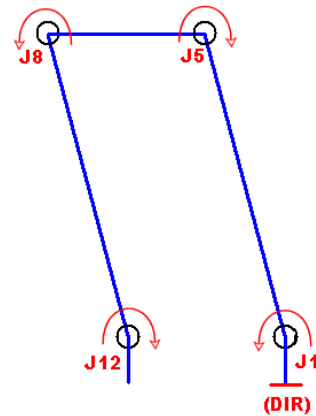


Figura 3-25 - Mov. lateral esquerda (visto de trás)

Portanto, pode-se inferir desta ilustração o seguinte, dado um incremento θ :

$$\begin{aligned}
 \theta_1 &= \theta_1 + \theta \\
 \theta_5 &= \theta_5 - \theta \\
 \theta_8 &= \theta_8 + \theta \\
 \theta_{12} &= \theta_{12} - \theta
 \end{aligned}
 \tag{3-17}$$

3.5.3 Elevar Pé Livre

A elevação do pé livre é uma réplica do movimento de baixar a anca, aplicado exclusivamente à perna livre. Isto porque a elevação do pé livre é efectuada pelas juntas sagitais. Portanto, caso o pé livre seja o direito a fórmula aplicada para elevar o pé é a seguinte:

$$\begin{aligned}
 \theta_2 &= \theta_2 + \theta \\
 \theta_3 &= \theta_3 - 2 \cdot \theta \\
 \theta_4 &= \theta_4 + \theta
 \end{aligned}
 \tag{3-18}$$

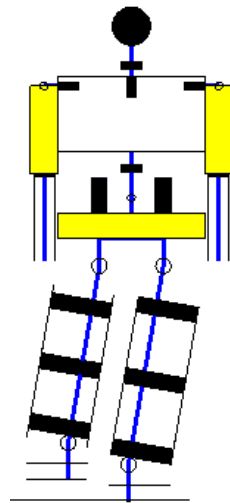


Figura 3-26 - Elevar o pé livre

Da mesma forma que o abaixamento da anca, a elevação do pé é realizada pelas juntas 2, 3 e 4, logo o movimento não garante uma abcissa constante para o pé livre, só no caso das juntas 2 e 4 estarem alinhadas segundo xx é que tal é verificado. Da mesma forma, a ordenada também é alterada se essas duas juntas não estiverem alinhadas segundo yy .

3.6 Avaliação de Tarefas Básicas

Estudaram-se três tarefas básicas para o robô humanóide, andar, mudar de direção e chutar uma bola. Estas tarefas envolvem especificações no espaço Cartesiano e no espaço das juntas.

3.6.1 Andar

O movimento de andar foi dividido em três fases,

1. Transferência do CoG para o pé de suporte;
2. Deslocamento e elevação do pé livre até perfazer metade do passo;
3. Deslocamento e abaixamento do pé livre até atingir o solo e perfazer o comprimento do passo.

Para o cálculo deste movimento são definidos 4 parâmetros Cartesianos de referência:

- Sl – Metade do comprimento do passo;
- Hh – Altura da anca;
- Yg – Posição lateral do centro da anca;
- Fc – Máxima elevação do pé.

A altura da anca reflecte a flexão das pernas durante o movimento, isto é, determina que no início do movimento, com o intuito de baixar o centro de gravidade e dar liberdade de movimento às pernas, a anca deve estar a determinada posição do solo. A posição lateral do centro da anca é definida para se garantir que o centro de gravidade se encontra sobre o pé de suporte, dessa forma e, por razões de simetria sagital da estrutura, o centro da anca deverá estar aproximadamente coincidente com o centro do pé de suporte, qualquer que ele seja. A máxima elevação do pé define o ponto intermédio por onde o pé livre terá de passar. Este ponto é muito importante porque, como as trajectórias locais são definidas no espaço das juntas, se só fosse enviado para o robô humanóide o valor das posições iniciais e finais do passo, iria ser calculada uma trajectória em que o pé livre avançaria por baixo do solo (na realidade cairia!). Com base nestes parâmetros e recorrendo a equações de cinemática inversa calculam-se as posições angulares das juntas ao longo do tempo.

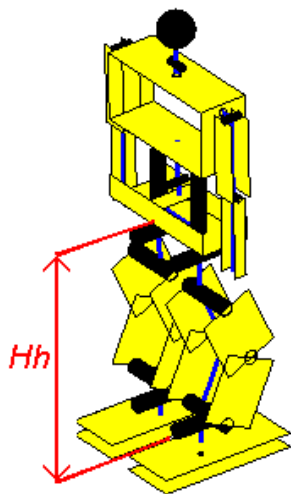


Figura 3-27 - Parâmetro Hh
(Altura da anca)

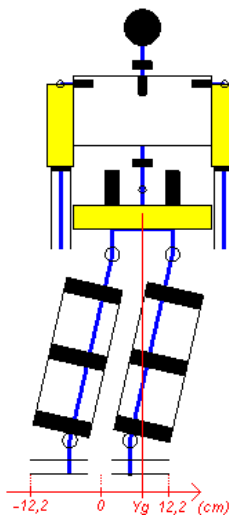


Figura 3-28 - Parâmetro Yg
(posição lateral da anca)

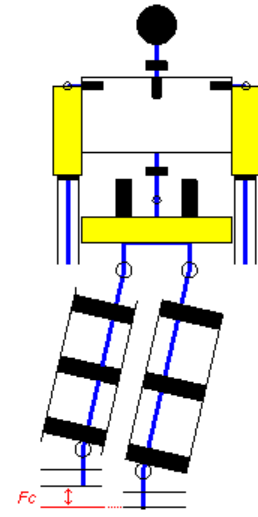


Figura 3-29 - Parâmetro Fc
(elevação do pé)

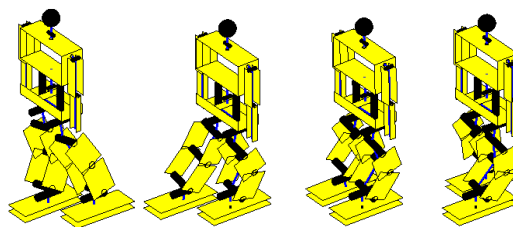


Figura 3-30 – Sequência de fases do movimento andar

Este movimento é integralmente definido no espaço Cartesiano. É fornecido para um ciclo iterativo, sob a forma de uma matriz dividida no número de fases que compõe o movimento, a especificação das coordenadas Cartesianas para a anca e para o pé livre. A análise pormenorizada do movimento é realizada em seguida como exemplo do estudo exaustivo dos padrões de locomoção.

Na fase inicial de duplo suporte, a projecção do *CoG* é deslocada para a parte de trás do pé direito, no caso ser este o pé de suporte durante o próximo passo, por exemplo. Depois do centro da anca ser posicionado, segundo o eixo dos *yy*, sobre o pé de suporte, há um avanço da anca de $\frac{3}{4}$ do comprimento do passo na direcção do movimento. Este movimento da anca correspondente a $\frac{1}{2}$ do comprimento do passo e acompanha o voo do pé livre, imitando o modelo do pêndulo invertido aplicado à locomoção bípede.

Em resultado da análise efectuada no estudo da cinemática inversa aplicada ao robô humanóide, podem-se ilustrar os parâmetros desta fase da seguinte forma:

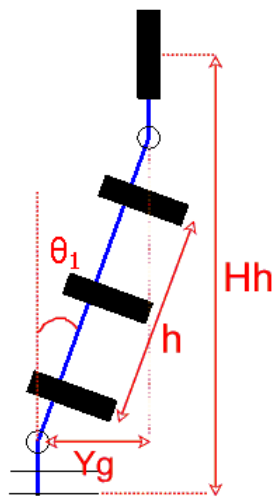


Figura 3-31 - Parâmetros pé de suporte (plano frontal)

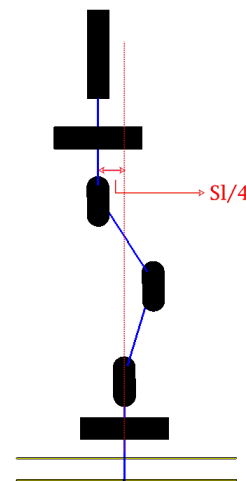


Figura 3-32 - Parâmetros do pé de suporte (plano sagital)

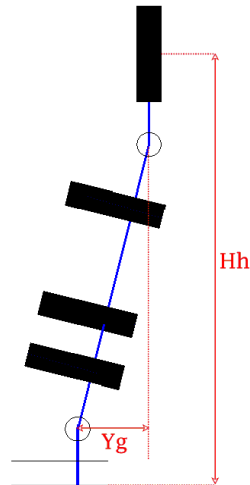


Figura 3-33 - Parâmetros do pé livre (plano frontal)

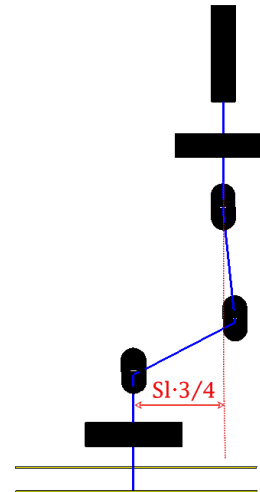


Figura 3-34 - Parâmetros do pé livre (plano sagital)

Onde, θ_1 é o ângulo da junta do pé, que garante uma posição da anca em yy igual a Yg , h corresponde à distância, no plano frontal, que separa as juntas 2 e 3 mais o comprimento dos dois elos l_2 e l_5 . Como em relação ao pé de suporte, a anca está recuada $\frac{1}{4}$ de Sl e o pé livre está recuado $\frac{3}{4}$ de Sl em relação à anca, tem-se uma distância total do pé livre em relação ao pé de suporte de Sl . Para cumprir com estas especificações são fornecidos os seguintes parâmetros para o cálculo da cinemática inversa:

Parâmetros para o pé de suporte:

$$\begin{aligned} Hh &= Hh \\ Yg &= Yg \\ Xg &= -\frac{Sl}{4} \end{aligned} \quad (3-19)$$

E para o pé livre:

$$\begin{aligned} Hh &= Hh \\ Yg &= Yg \\ Xg &= -Sl \end{aligned} \quad (3-20)$$

Depois de estar no ponto intermédio o pé livre deve ser enviado para a sua posição final, ou seja, no início do passo o pé livre desloca-se o comprimento de um passo para a frente, mas fica parado no ar a uma altura Fc . Esta fase intermédia serve para assegurar que no planeamento da trajectória (espaço das juntas), o pé livre não executasse o movimento sob o piso! Se esse problema não existisse esta fase seria dispensável.

A anca acompanha sempre o movimento do pé. No final, o centro da anca encontra-se exactamente alinhado segundo o eixo dos xx com o pé de suporte e a posição segundo o eixo dos yy não se altera.

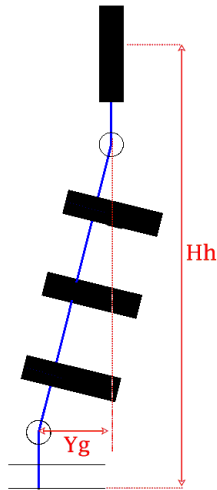


Figura 3-35 - Parâmetros pé de suporte (plano frontal)

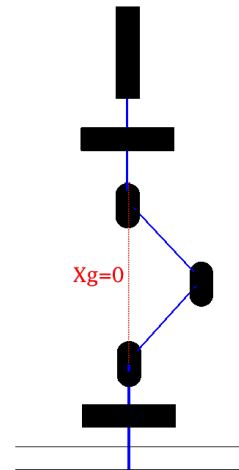


Figura 3-36 - Parâmetros do pé de suporte (plano sagital)

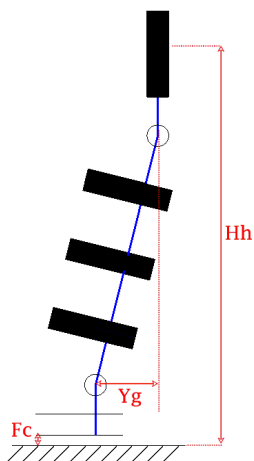


Figura 3-37 - Parâmetros do pé livre (plano frontal)

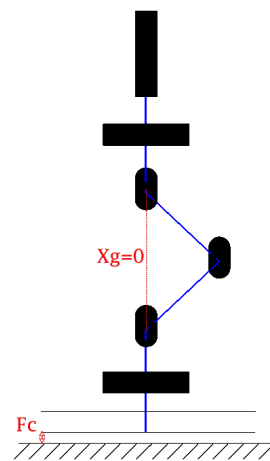


Figura 3-38 - Parâmetros do pé de livre (plano sagital)

Parâmetros para o pé de suporte:

$$\begin{aligned} Hh &= Hh \\ Yg &= Yg \\ Xg &= 0 \end{aligned} \quad (3-21)$$

E para o pé livre:

$$\begin{aligned} Hh &= Hh - Fc \\ Yg &= Yg \\ Xg &= 0 \end{aligned} \quad (3-22)$$

Depois de estar no ponto intermédio o pé livre deve ser enviado para a sua posição final, ou seja terá de percorrer o segundo passo que falta. No fim desta fase a anca avançou $\frac{1}{4}$ do passo segundo o eixo dos xx e a posição em yy não se altera.

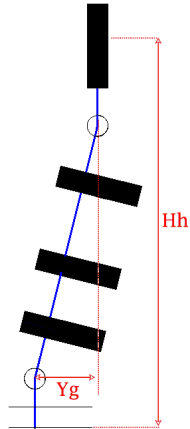


Figura 3-39 - Parâmetros pé de suporte (plano frontal)

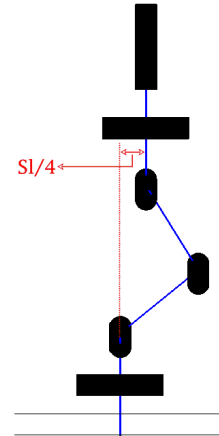


Figura 3-40 - Parâmetros do pé de suporte (plano sagital)

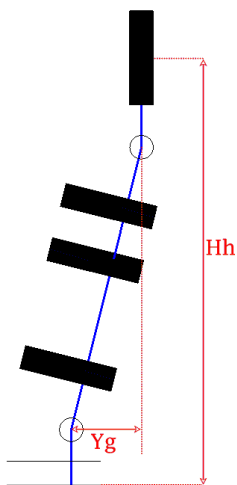


Figura 3-41 - Parâmetros do pé livre (plano frontal)

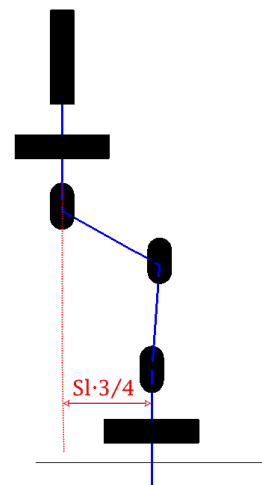


Figura 3-42 - Parâmetros do pé de livre (plano sagital)

Parâmetros Cartesianos para o pé de suporte:

$$\begin{aligned} Hh &= Hh \\ Yg &= Yg \\ Xg &= \frac{Sl}{4} \end{aligned} \quad (3-23)$$

E para o pé livre:

$$\begin{aligned} Hh &= Hh \\ Yg &= Yg \\ Xg &= Sl \end{aligned} \quad (3-24)$$

Como há simetria segundo o plano sagital a execução de múltiplos passos envolve a repetição desta sequência mas convertida para as juntas simetricamente opostas no plano sagital. Dessa forma, exemplificando a mesma configuração da primeira fase mas desta vez com referência ao pé de suporte anterior.

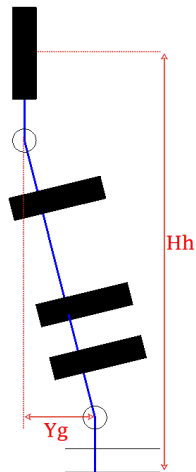


Figura 3-43 - Parâmetros pé de suporte (plano frontal)

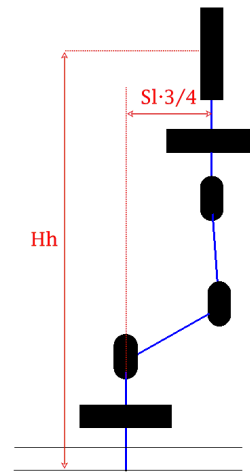


Figura 3-44 - Parâmetros do pé de suporte (plano sagital)

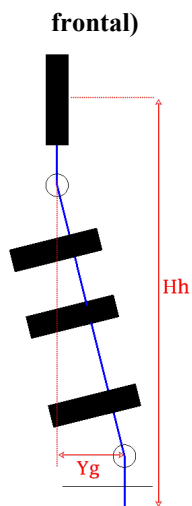


Figura 3-45 - Parâmetros do pé livre (plano frontal)

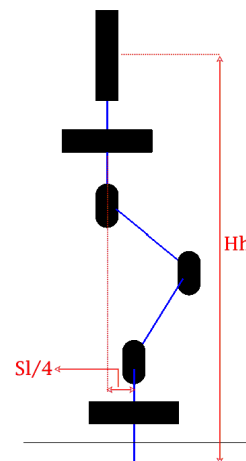


Figura 3-46 - Parâmetros do pé de suporte (plano sagital)

A criação de vários passos recorre à simetria no plano sagital. Dessa forma é possível replicar o passo para o outro pé de suporte aplicando as mesmas posições adaptadas a esta perna. É através desse método de cálculo do movimento de um passo para o pé de suporte e replicação simétrica do passo para o outro pé que se obtém vários passos seguidos. O algoritmo usado é então o representado na **Figura 3-47**.

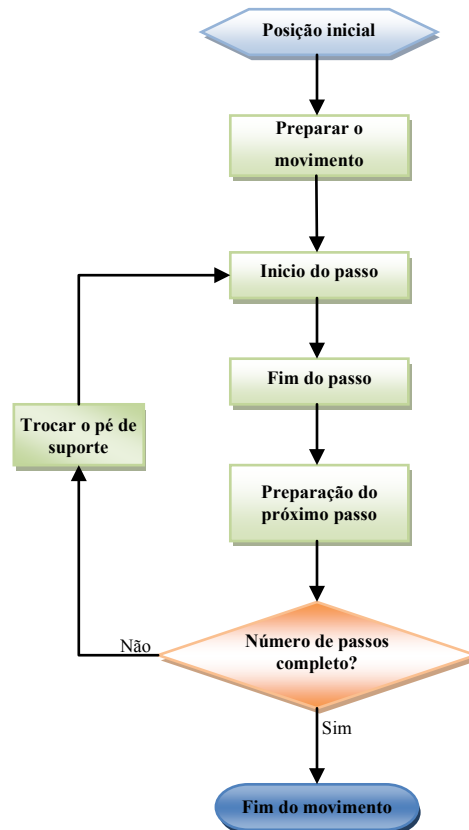


Figura 3-47 - Algoritmo para a realização de vários passos

3.6.2 Mudança de Direcção

O movimento de rotação sobre si próprio possibilita ao robô humanóide mudar a sua direcção, por isso é um movimento de enorme importância no desenvolvimento de padrões de locomoção. Este movimento apresenta contudo uma dificuldade intrínseca à própria concepção da estrutura, isto porque, apesar dos 12 graus de liberdade existentes da cintura para baixo neste robô, não são comparáveis aos graus de liberdade que um ser humano apresenta. Daí que para o ser humano este movimento de rotação seja relativamente simples mas projectá-lo para este robô humanóide não seja tão trivial assim.

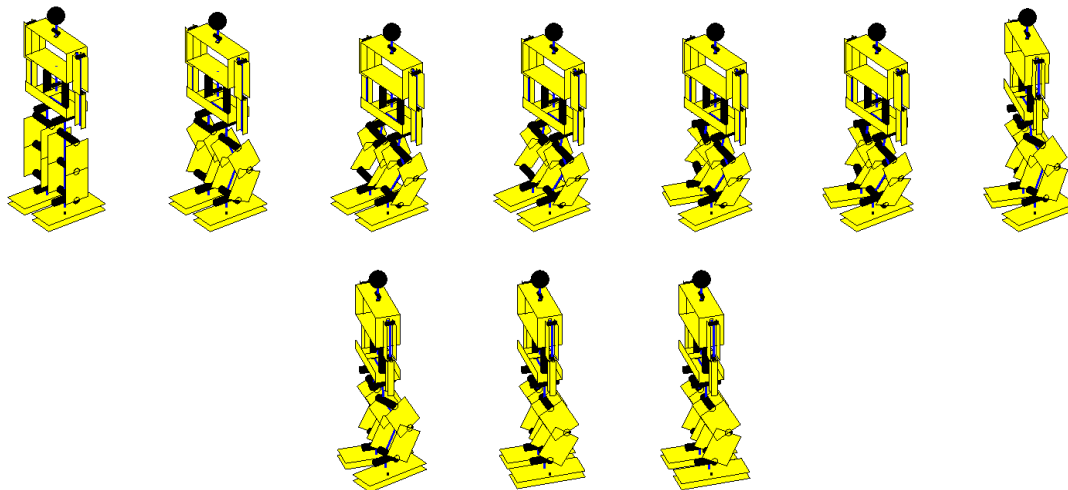


Figura 3-48 - Sequência do movimento de mudança de direcção

A sequência do movimento de rotação é bastante extensa, como se pode ver pela **Figura 3-48**. Este movimento está dividido em 9 fases, mas poderá ainda ser encurtada com a junção de algumas fases, sem que isso afecte a sua correcta execução. A razão para não se terem encurtado as fases deve-se ao actual processo de estudo que ainda decorre sobre este movimento e para explicar separadamente cada uma das sequências envolvidas nesta tarefa.

Há vários parâmetros que definem este movimento, são eles:

- ang_r – Ângulo de rotação desejado;
- Hh – Altura da anca;
- Yg – Posição lateral do centro da anca;
- Fc – Altura do pé durante o movimento de rotação.

O cálculo das fases é executado no espaço Cartesiano e no espaço das juntas. Este movimento pode ser explicado em quatro fases, estando as várias fases reais encapsuladas em cada uma destas quatro. A primeira pode ser designada por fase de preparação, é quando o humanoide usa os parâmetros Hh e Yg para flectir as pernas e incliná-las de forma a alinhar o CoG com o pé de suporte. A segunda fase envolve os passos de levantar, rodar e pousar o pé livre e representa o movimento de rotação deste. Como terceira fase está o movimento complicado de passar o CoG para o novo pé de suporte. Esta fase representa um único movimento bastante complexo, que consiste num sincronismo de juntas preciso, calculado através de cinemática inversa aplicada a ambas as pernas. Como última fase designa-se o movimento de levantar o novo pé livre, rodá-lo e voltar a pousá-lo no solo.

Na **Figura 3-49**, ilustra-se o problema da rotação do pé livre, para o pé rodar, têm que se rodar toda a perna segundo uma circunferência. O eixo dessa circunferência provém da junta que efectua o movimento de rotação. O problema que isto provoca, representado na **Figura 3-50** e **Figura 3-52**, é o desalinhamento entre as juntas responsáveis pelo movimento segundo xx.

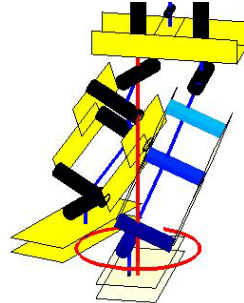


Figura 3-49 - Rotação do pé livre

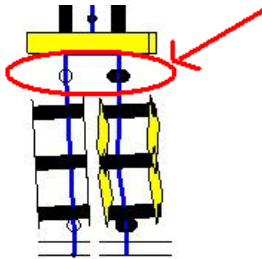


Figura 3-50 - Deslocação do CoG para o novo pé de suporte

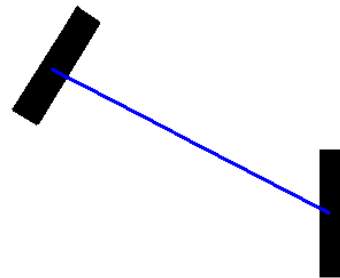


Figura 3-51 - Juntas em planos diferentes (vista de cima)

3.6.3 Pontapear uma Bola

Este é das três, a tarefa mais simples que está implementada. A sua utilidade prática incide basicamente sobre as competições de robótica humanóide que existem, em que um dos testes é o remate de uma bola.

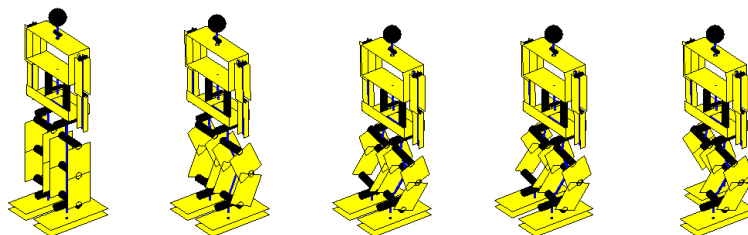


Figura 3-52 - Sequência de movimento de pontapear

Os parâmetros que definem este movimento são os seguintes:

- H_h – Altura da anca;
- Y_g – Posição lateral do centro da anca;
- F_c – Altura do pé aquando do remate.

Este movimento tem as mesmas características iniciais do movimento de rotação, ou seja há uma fase de preparação onde as pernas são flectidas e o CoG é transferido para o pé de suporte. Depois há uma fase de elevação do pé livre a uma altura F_c e seguidamente, o pé livre avança para frente no sentido de fazer um remate. A altura F_c é também o ponto segundo o eixo dos zz em que o pé vai embater com a bola, por isso, ao contrário do que se passa com o movimento de rotação esta altura têm que ser especificada tendo em vista o ponto de impacto do remate.

O avanço final do pé livre é executado unicamente pela junta do joelho. Este movimento consiste em espelhar o ângulo que a junta faz com a vertical.

3.7 Discussão de Resultados

O algoritmo de cinemática inversa aplicado, apesar de restringir alguns dos graus de liberdade existentes no humanóide ao impor uma postura sempre vertical, tem a enorme vantagem de ser um método analítico. Este facto permite obter facilmente, de forma directa, as posições angulares que satisfazem determinada posição Cartesiana da anca ou do pé livre. A subdivisão da cadeia cinemática em duas cadeias mais simples e uma terceira não actuada é suficiente para a execução das tarefas básicas criadas mas terá de se pensar em soluções mais abrangentes quando se quiser aplicar movimentos definidos no espaço Cartesiano para a cabeça ou as mãos do robô.

Mostrou-se que, para baixas velocidades, o CoP e $PVCoG$ são pontos quase coincidentes. A aplicação do CoP envolve o conhecimento preciso dos binários existentes em cada junta e na presença de dinâmica torna-se matematicamente complicado determinar esse valor. Estes factos serviram de base para a aplicação exclusiva do $PVCoG$ como critério de estabilidade ao robô humanóide.

As tarefas básicas definidas estão ainda em fase de simulação necessitando, principalmente no caso da mudança de direcção, de alguns testes adicionais para poderem ser aplicadas ao robô com segurança.

4 Interface Gráfica do Utilizador

4.1 Introdução

A criação de uma interface gráfica de simulação e controlo de alto-nível para aplicação a um robô humanóide aponta para um patamar de desenvolvimento superior onde se analisam problemas de navegação, de estabilidade, de coordenação e planeamento de trajectórias [7] [13] [14]. Esta evolução na abordagem ao robô pressupõe uma base mecânica e electrónica sólida capaz de suportar os novos desafios propostos.

No âmbito desta dissertação, pretende-se desenvolver um conjunto de ajudas computacionais que, por um lado, permitam estudar padrões de locomoção associados a tarefas elementares e, por outro lado, facilitem a programação e monitorização do robô. São especificados nos objectivos deste trabalho que um utilizador da interface poderá seleccionar, sempre que considere relevante, um conjunto de ferramentas que permitam inicializar, configurar e monitorizar o robô humanóide, actuar individualmente as juntas, estudar e validar uma sequência de movimentos e efectuar leituras sensoriais.

Este capítulo apresenta a interface gráfica criada, analisando as suas funcionalidades, o conteúdo e aspecto visual dos seus comandos, a sua organização e uma visão geral da sua programação em Matlab.

4.2 Descrição das Funcionalidades

O *graphical user interface* (GUI) denominado *TwoLegs_22dof* encaminhou-se no sentido dos objectivos estabelecidos para esta dissertação. Recordando as três áreas abrangidas nos objectivos, simulação, monitorização e controlo da plataforma humanóide. Através desta enumeração é possível traçar a evolução e as funcionalidades do *TwoLegs_22dof*.

A simulação da estrutura humanóide está assegurada por algoritmos de cinemática directa e rotinas de *plot*. O ambiente oferecido pela interface permite actuar independentemente em cada junta ou invocar qualquer uma das rotinas dos padrões de locomoção. Depois de definido, ou calculado no caso de se aplicar um dos padrões de locomoção, as posições angulares das juntas em cada fase do movimento, são invocadas as funções de cinemática directa que retornam a postura e orientação de cada um dos componentes do robô humanóide. Através desta informação é possível animar o modelo humanóide correspondente à configuração introduzida, assim como elaborar cálculos de CoG e binários. A junção das fases é calculada pelo algoritmo de planeamento de trajectórias,

elaborando dessa forma uma estrutura de dados com posições angulares, posições Cartesianas, CoG e binários num espaço temporal segmentado em períodos de 20ms. Estes dados são processados de forma a gerar gráficos com a evolução das posições angulares, do CoG ou do binário. É também possível visualizar uma animação completa do movimento.

A monitorização da estrutura humanóide é obtida por intermédio do protocolo de comunicações, logo está sujeita as características imposta pelo protocolo. Este aspecto é relevante para explicar mais à frente alguns inconvenientes da monitorização humanóide. Há duas formas possíveis de visualizar a informação proveniente do robô,

1. Através de mostradores embutidos na interface gráfica, com uma actualização accionada manualmente ou automaticamente com uma taxa de actualização de 1/100ms mas com o inconveniente de tornar todo o GUI muito mais lento.
2. Aquando do envio de um movimento para o humanóide através de uma interface gráfica própria designada por sendGUI e que permite recolher à taxa de 1/100ms, a variação das posições das juntas ao longo do tempo de execução do movimento.

O protocolo de comunicações, tendo em conta a arquitectura cliente/servido que apresenta, necessita que se peça ao *master* sempre que se quer ler um valor relativo a uma qualquer *slave*. Ora, como cada mensagem enviada contém 5 bytes e 6 bytes cada mensagem recebida, a uma taxa de 115200bps, corresponde a um tempo total de pedido/resposta por parâmetro e por *slave* de 0.76ms. Ou seja, a leitura das posições de oito *slaves* implica um tempo total de 6,1ms, sem contar com períodos de latência, erros de comunicação, atrasos ou processamento quer por parte da PIC ou do PC (desprezável). Este valor pode ser duplicado se adicionar às posições a leitura dos sensores especiais. O protocolo, do lado do PC, fica em *pooling* até obter a resposta do *master*, quer isto dizer que se o processo de pedido/resposta dura 12.2ms, o PC fica esses 12ms preso a essa rotina. Se houvesse uma aplicação dedicada a executar esta leitura, não haveria qualquer problema, mas a dificuldade reside na integração de mais este gasto temporal em todo o GUI. Para mais, quando se fala numa actualização a cada 100ms, isto implica que mais de 12% do processamento é dedicado à leitura dos valores sensoriais.

Quando se fala em inconvenientes do protocolo de comunicação está-se a referir à necessidade de efectuar um pedido ao *master* e ficar em *pooling* à espera da resposta. Também se refere ao facto de se ter que efectuar pedidos *slave a slave* quando se pretende ler todo o conjunto. Como estas alterações ao protocolo necessitariam de bastante tempo dedicado ao problema e fugiam ligeiramente ao cerne dos objectivos optou-se por manter o protocolo de comunicações intacto

acarretando o efeito inercial introduzido no GUI. Nas versões mais recentes do Matlab é possível especificar processos *batch* que correm em paralelo com outros processos (a sua gestão é efectuada ao nível do processador), esta seria uma solução parcial ao problema indicado, porque pelo menos tornava independente o processamento do GUI e do protocolo de comunicações, podendo até aumentar-se a taxa de actualização.

A outra funcionalidade indicada é o controlo do robô. Mais uma vez o protocolo de comunicações desempenha um papel crucial para a execução desta funcionalidade. O controlo do robô consiste em enviar parâmetros de configuração/actuação de todas as variáveis possíveis existentes nas *slaves*. Posições, velocidades, ganhos e definição do tipo de controlo dos servomotores encontram-se nos parâmetros directamente modificados pelo utilizador. No protocolo de comunicações estão descritas as variáveis configuráveis do robô (**Apêndice B**). Em relação à actuação das posições das juntas existem três alternativas,

1. Envio das posições relativas a cada *slave* (ex.: *slave* do pé – envio da junta do pé, tornozelo e joelho);
2. Envio do conjunto das oito *slaves*, ou seja das 22 juntas existentes;
3. *Graphical User Interface sendGUI* criado especificamente para o envio de movimentos compostos (com várias fases) para o robô humanóide.

Sempre que se actua numa junta é necessário ter activo o *PWM* respectivo, isto implica que sempre que se usa o método indicado em 2 só são realmente actuadas as *slaves* que possuem *PWM* activo, o mesmo se passa quando se usa a interface *sendGUI*.

4.3 Organização da Interface Gráfica

A introdução de uma interface gráfica pressupõe obviamente a criação de facilidades na interacção com um ambiente complexo. No caso do robô humanóide interessa criar um ambiente que permita de uma forma fácil, acessível e intuitiva trabalhar sobre o robô humanóide, seja em simulação ou em controlo. Na **Figura 4-1** pode-se observar a organização global do *TwoLegs_22dof*. Possui um menu geral, composto pelas opções das diversas vocações do GUI, depois está dividido em três blocos, o painel de simulação, o painel de monitorização e controlo e por último a janela de *plot* do robô humanóide.

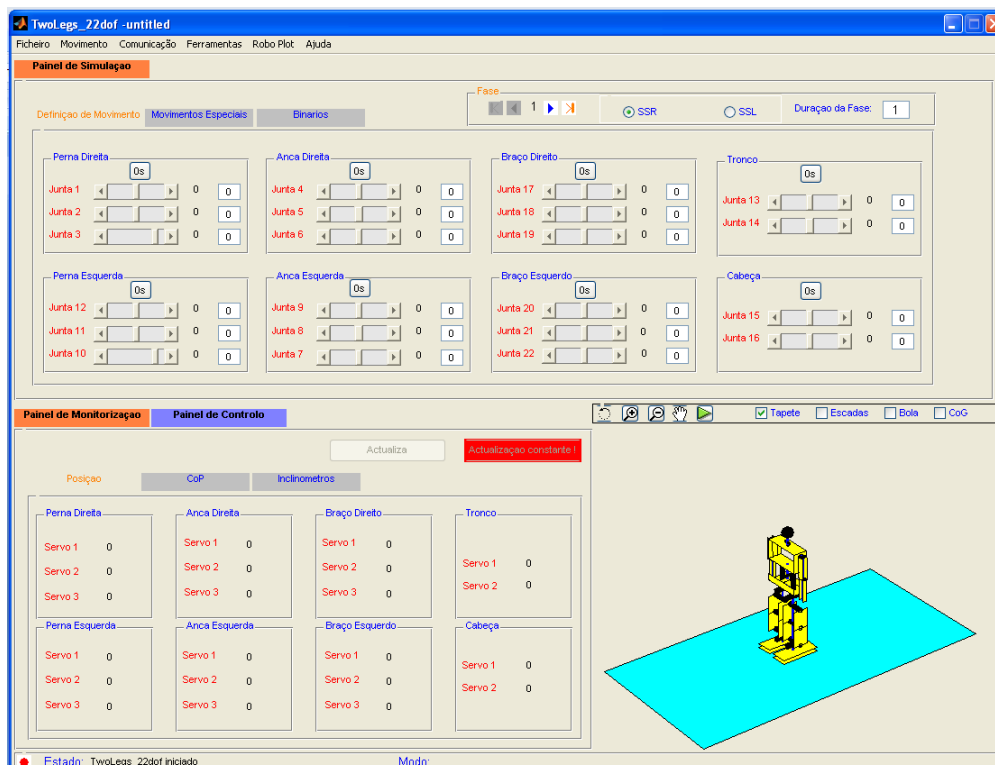


Figura 4-1 - Painel de abertura do *TwoLegs_22dof*

Este GUI apresenta como componentes adicionais a barra de status no rodapé da janela, indicativo do tipo de operação que está a ser executada e um botão tricolor que informa sobre o actual estado das comunicações,

Indicador	Estado
●	Desligado
●	A ligar ou desligar
●	Ligado

Tabela 4-1 - Estado da ligação

4.3.1 Painel de Simulação

O painel de simulação é o responsável pela introdução de posições nas juntas e composição de movimentos complexos. Possibilita também a visualização da informação relativa ao binário em sentido estático que as juntas suportam em cada uma das fases. Na **Figura 4-2** tem-se uma visualização geral da organização deste painel.

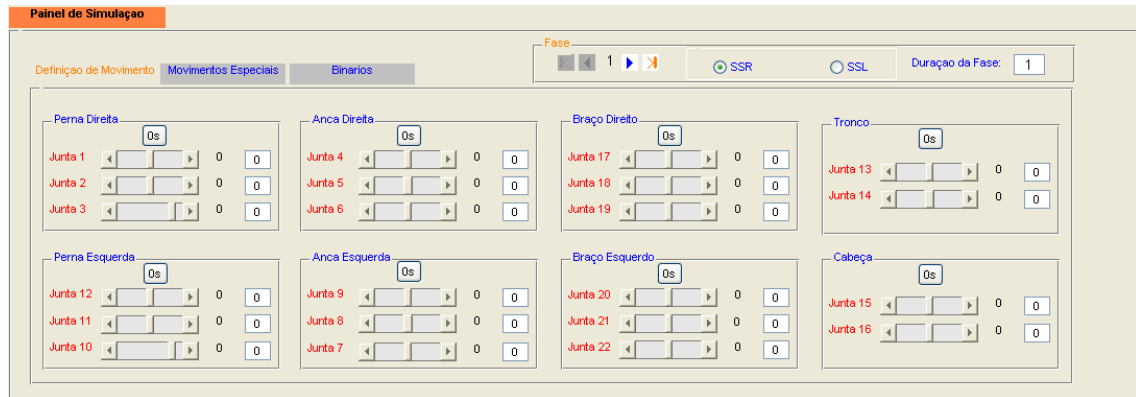


Figura 4-2 - Painel de Simulação

Encontra-se uma barra denominada por **Fase** onde é possível saber a actual fase de movimento, o tipo de suporte e a duração do movimento para essa fase. Na **Figura 4-3** pode-se observar com mais pormenor a estrutura desta zona de controlo da fase. Através de botões de avanço e recuo é possível controlar a fase actual. Na **Tabela 4-2**, indica-se o tipo de suporte disponível para a simulação do movimento humanóide, que pode ser suporte simples da perna direita ou da perna esquerda.

SSR	<i>Single Support Right</i>
SSL	<i>Single Support Left</i>

Tabela 4-2 - Tipo de suporte disponível

Por razões inerentes à implementação dos cálculos cinemáticos, não é possível definir duplo suporte para determinada fase. O campo **duração da fase** representa o tempo total da trajectória que deverá decorrer entre a fase anterior e a fase actual.

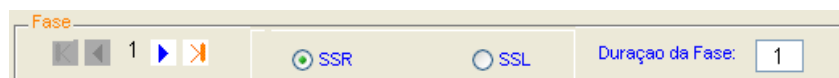


Figura 4-3 - Barra de controlo da Fase

Em formato de *tabPanels* encontram-se o painel de definição de movimento, o painel com os movimentos especiais e o painel de visualização de binários. Na **Figura 4-4** apresenta-se o painel de definição de movimento, este é dividido em oito espaços de actuação, representando as oito *slaves* que controlam toda a estrutura humanóide. Em cada uma dessas oito secções encontram-se barras de controlo da posição de cada junta, uma caixa de texto com o respectivo valor angular da junta e uma caixa de texto editável designada por **dependência**. Esta última *editBox* representa possíveis dependências que podem existir entre juntas, quer isto dizer que por exemplo para a

execução de determinado movimento a junta 12 deve deslocar-se o mesmo que a junta 1, então na *editBox* dependência da junta 12 indica-se o valor 1 e todos os movimentos que a junta 1 fizer a junta 12 também vai executar, a menos que os limites de uma ou outra sejam ultrapassados. Se em vez de 1 se indicar -1, a junta 12 vai-se deslocar o oposto da junta 1.

Existe também em cada uma das secções um botão com o valor 0s que permite colocar a zeros o valor de todas as juntas dessa secção.

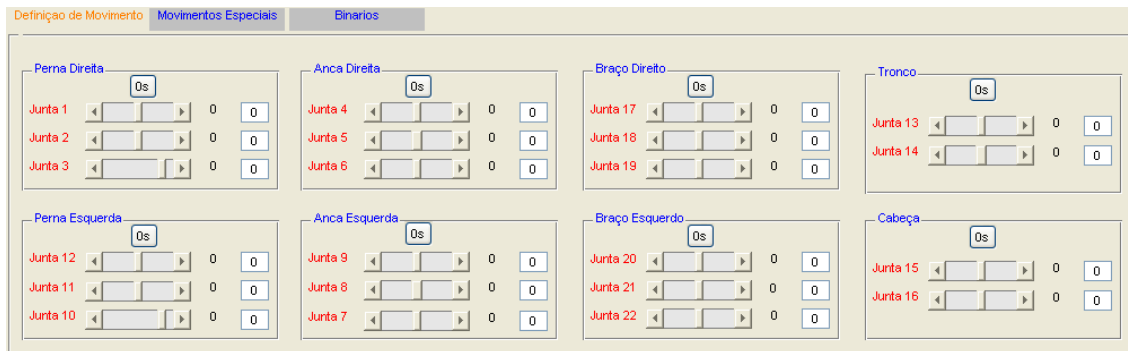


Figura 4-4 - Painel de Simulação

Na **Figura 4-5**, está indicado o painel com os movimentos especiais, quer isto dizer, as configurações básicas e tarefas simples definidas na **secção 3.5** e **3.6**, respectivamente, desta dissertação. Cada uma das tarefas básicas possui um conjunto de variáveis que as definem e para cada uma o utilizador tem limites impostos pela cadeia cinemática humanóide. É por isso que se pousar o ponteiro do rato sobre a caixa de texto respectiva serão apresentados os limites máximos e mínimo que essa variável poderá ter (**Apêndice D**). Esta função revela-se bastante útil porque por vezes os limites destas variáveis encontram-se muito próximos e seria penoso para o utilizador fazer tentativas até encontrar o valor que melhor se ajusta às suas necessidades e que estivesse dentro dos limites admissíveis pelo simulador cinemático.

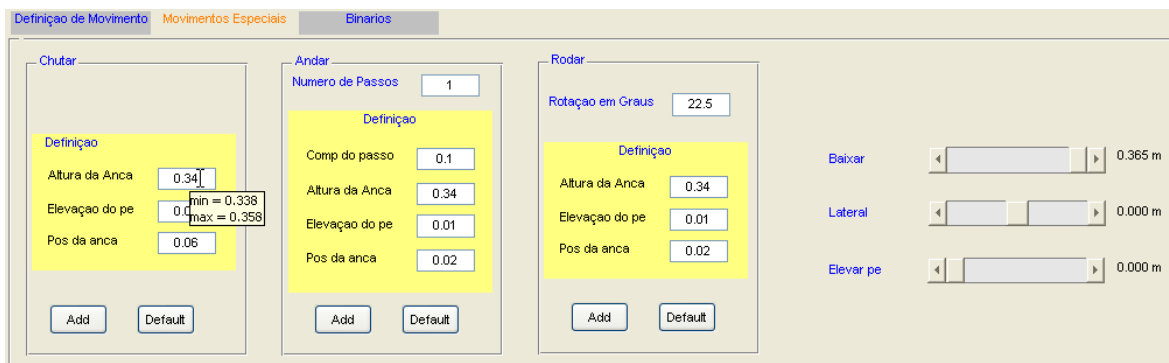


Figura 4-5 - Painel de Movimentos Especiais

Por último, na **Figura 4-6** está o painel que fornece informação sobre o binário necessário para os servos suportarem determinada configuração do robô humanóide. Mais uma vez atenção, porque esta definição de binário só considera o efeito da gravidade no robô, designado binário estático e não contempla as acelerações presentes no humanóide quando este está em movimento.

Definição de Movimento		Movimentos Especiais		Binários			
Perna Direita		Anca Direita		Braço Direito		Tronco	
Junta 1	2.01	Junta 4	-0.65	Junta 17	-0.07	Junta 13	-1.01
Junta 2	-0.66	Junta 5	2.13	Junta 18	-0.00	Junta 14	-0.13
Junta 3	-0.66	Junta 6	0.00	Junta 19	0.00		
Perna Esquerda		Anca Esquerda		Braço Esquerdo		Cabeça	
Junta 12	0.08	Junta 9	-0.22	Junta 20	-0.00	Junta 15	0.00
Junta 11	-0.21	Junta 8	0.21	Junta 21	0.00	Junta 16	0.00
Junta 10	-0.21	Junta 7	0.00	Junta 22	0.00		

Figura 4-6 - Painel de visualização de Binários

4.3.2 Painel de Monitorização

Algumas das variáveis de monitorização que estão disponíveis nas *slaves* podem ser visualizadas no *TwoLegs_22dof*. Uma vista geral do painel de monitorização é dada na **Figura 4-7**, onde se pode conhecer a organização deste “mostrador”. Existem dois botões de actualização no canto superior direito que permitem actualizar manualmente ou activar a actualização automática. Já foram referidos (**secção 4.2**) os problemas associados a esta forma de actualização e deriva daí o aspecto choque que este botão apresenta. Sempre que se efectua uma actualização todas as variáveis existentes neste painel são actualizadas, este modo de funcionamento não permite uma actualização selectiva.

Painel de Monitorização		Painel de Controlo	
<input type="button" value="Actualiza"/> <input style="background-color: red; color: white;" type="button" value="Actualização constante!"/>			
Posição		CoP	
Inclinómetros			
Perna Direita		Anca Direita	
Servo 1	0	Servo 1	0
Servo 2	0	Servo 2	0
Servo 3	-28	Servo 3	0
Perna Esquerda		Anca Esquerda	
Servo 1	0	Servo 1	0
Servo 2	0	Servo 2	0
Servo 3	-28	Servo 3	0
Braço Direito		Braço Esquerdo	
Servo 1	0	Servo 1	0
Servo 2	0	Servo 2	0
Servo 3	0	Servo 3	0
Tronco		Cabeça	
Servo 1	0	Servo 1	0
Servo 2	0	Servo 2	0

Figura 4-7 - Vista geral do painel de Monitorização

O primeiro painel que se apresenta é o indicador da posição das juntas, representado na **Figura 4-8**. Como se afirmou na **secção 2.2.1**, os servomotores não tem um zero coincidente com a posição vertical do humanóide mas, a apresentação das posições no painel posição já possui a correcção desse valor, tanto do *offset* estático como dinâmico. A correcção estática refere-se aos *offsets* predefinidos no robô, por exemplo, o servomotor do joelho está próximo de um dos limites, pois tal como no ser humano o joelho só se dobra em determinado sentido, logo o θ do joelho na interface gráfica corresponde a um dos extremos do servomotor do joelho. O *offset* dinâmico representa os ajustes que se vão fazendo no robô de forma a garantir a sua verticalidade na posição zero do simulador. Os valores dos *offsets*, tanto estáticos como dinâmicos são completamente transparentes para o utilizador, resultando unicamente de conversões automáticas entre as diferentes realidades, virtual no simulador e real no robô humanóide.



Figura 4-8 - Painel de Posição

O centro de pressão é representado de acordo com a **Figura 4-9**, isto é, através de uma marca no pé. Indica-se também a posição em *cm* mas este valor não é absoluto e é resultado de um ganho *k* indicado pelo utilizador e que se aplica ao valor lido nos sensores.



Figura 4-9 - Painel do centro de pressão (CoP)

Os inclinómetros medem duas direcções perpendiculares e apesar de a sua localização ser variável e dependente das necessidades do utilizador a sua aplicação está indicada para medir a inclinação do tronco no plano sagital e frontal. Dessa forma apresenta-se na **Figura 4-10** duas barras por inclinómetro que demonstram a inclinação do tronco em cada uma das direcções medidas pelos sensores de inclinação para ambos os lados do humanóide.

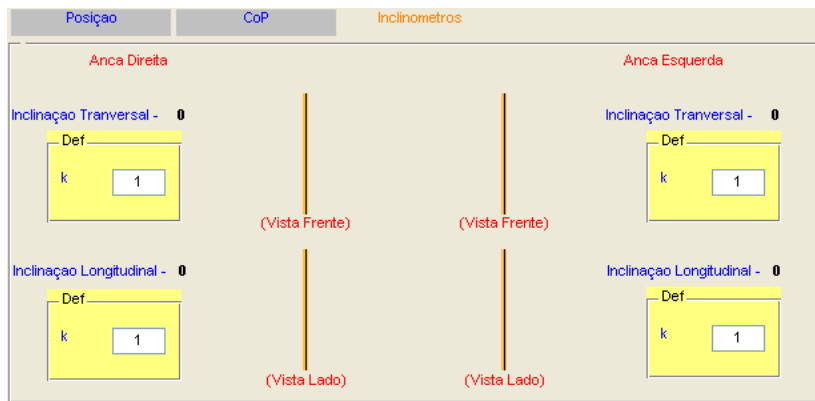


Figura 4-10 - Painel dos Inclinómetros

4.3.3 Painel de Controlo

Já foram referidos os tipos de controlo disponíveis para o robô (**secção 2.2.1 e 2.2.2**) mas no protocolo de comunicações (**Apêndice B**) compreende-se melhor a sua utilização e hierarquia. Na **Figura 4-11** apresenta-se a organização geral do painel de controlo. Existe um único botão que envia para o robô humanóide toda a estrutura de controlo configurada nestes painéis. O resto do painel consiste em dois *tabPanels* com as especificidades do controlo de primeiro e segundo nível.

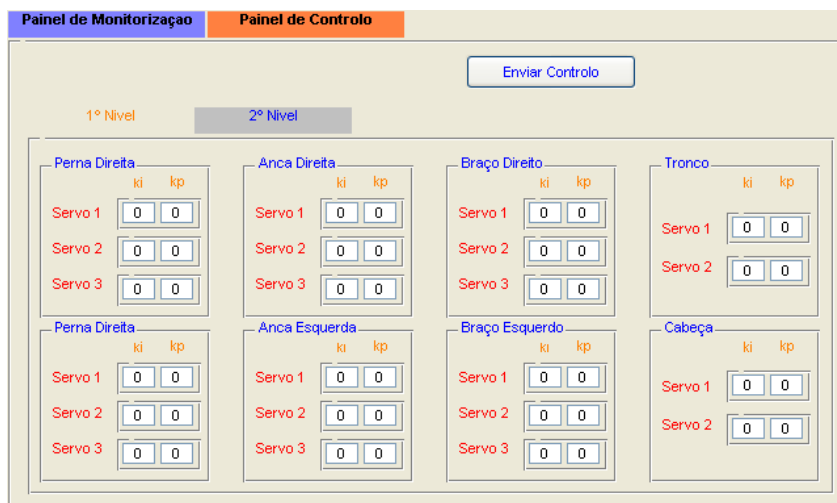


Figura 4-11 - Vista geral do painel de controlo

Na **Figura 4-12**, está em pormenor o painel de especificação do controlador de primeiro nível, encontram-se pois, dois parâmetros, são eles o ki e o kp , a aplicação deste controlador está dependente do protocolo de comunicações (**Apêndice B**). Em termos gerais, o motor é precedido por um controlador PD encapsulado no próprio servomotor mas, existe um segundo controlador PI integrado nas unidades controladoras *slaves*. Os parâmetros do controlador PD não podem ser alterados pelo utilizador porque o circuito está integrado directamente no servomotor e não têm acesso exterior. Dessa forma, os parâmetros ki e kp destinam-se exclusivamente ao controlador PI implementado nas *slaves*.

Sempre que se colocam a zero os dois parâmetros está-se a desligar o controlador PI e a usar somente o controlador dos servomotores.



Figura 4-12 - Painel de controlo 1º nível

Para o 2º nível de controlo são disponibilizadas actualmente 3 hipóteses:

1. Tipo 0 - Sem controlo de 2º nível;
2. Tipo 1 - Controlador de CoP;
3. Tipo 2 - Controlador de inclinação.

Cada um destes controladores e o seu respectivo ganho podem ser definidos no painel representado na **Figura 4-13**.



Figura 4-13 - Painel de controlo 2º nível

4.3.4 Plot do Robô Humanóide

O *plot* do robô é acompanhado por uma *toolbar* para configuração dessa visualização. A **Figura 4-14** ilustra essa zona de visualização do robô humanóide.

A *toolbar* possibilita a introdução de alguns elementos visuais auxiliares de uma simulação, são eles:

1. Tapete – piso sobre o qual o robô se movimenta;
2. Escadas – Elemento que auxilia na criação de movimentos envolvendo escadas (subir ou descer escadas);
3. Bola – Apesar de não estar habilitado de qualquer tipo de reacção, este elemento poderá ser útil na criação de trajectórias e remate de uma bola (muito usual em competições de robótica humanóide);
4. CoG – Este é talvez o elemento visual mais útil na criação de qualquer movimento, além do CoG é também indicada a sua projecção no solo, o que permite obter uma ideia sobre a estabilidade de determinadas posições.

Além dos elementos visuais é também possível ajustar a visualização do *plot* humanóide através de botões de *zoom*, de *pan* e de rotação 3D. Está também acessível nesta *toolbar* um acesso rápido ao comando de animação do movimento criado.

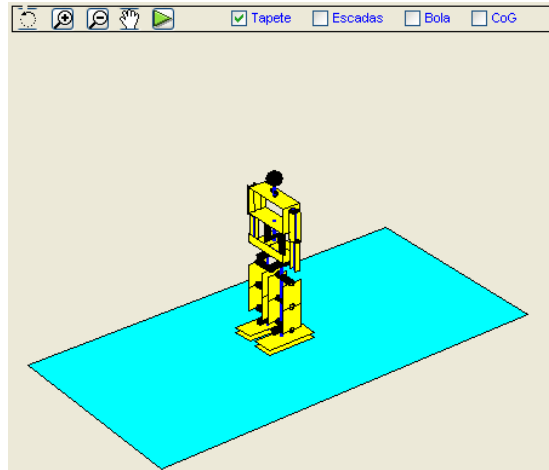


Figura 4-14 - Área de Plot do Robô Humanóide

4.3.5 Envio de Tarefas

Quando se trabalha com um robô humanóide, em fase de testes, raramente o movimento simulado coincide com o movimento que o robô desempenha. Este foi o motivo para a criação de uma ferramenta independente para o envio de movimentos para o robô, onde é possível repetir várias vezes o envio de determinado movimento, inclusive alterando a sequência de movimento.



Figura 4-15 - GUI para envio de sequências de movimentos, sendGUI

Como característica principal, esta ferramenta dá ao utilizador a capacidade para recolher dados do robô humanóide enquanto este executa o movimento enviado. Estes dados podem depois ser

analisados e comparados com as simulações. Outra das características deste GUI é a possibilidade de se alterar a sequência das fases do movimento ou só enviar algumas das fases do movimento criado, tal como exemplificado na **Figura 4-16**. Outra das opções disponíveis é a estipulação de uma duração fixa para cada fase, em vez de respeitar a duração programada durante a criação da sequência de movimento.

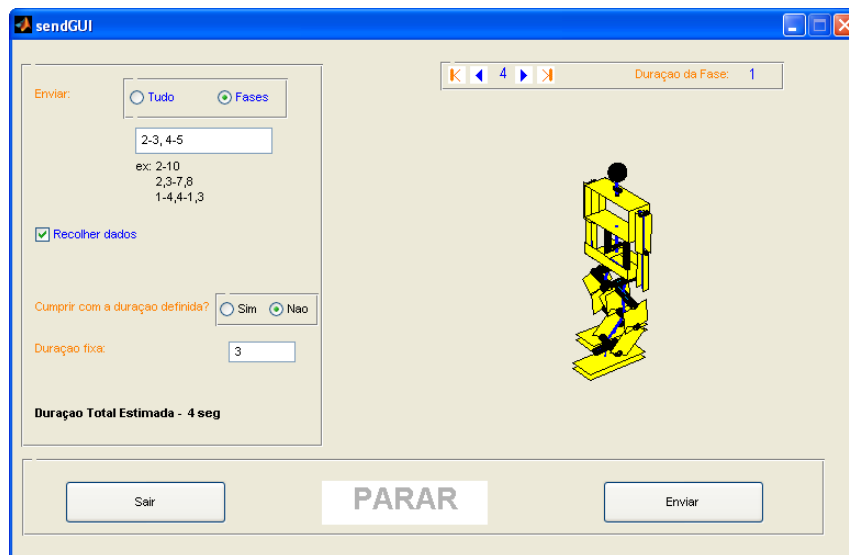


Figura 4-16 - Utilização do sendGUI

Depois de ter a sequência de movimentos pronta para ser enviada e em caso de estar seleccionada a opção de recolher dados, aparece uma janela para a introdução de uma etiqueta para os dados que vão ser recolhidos, **Figura 4-17**.

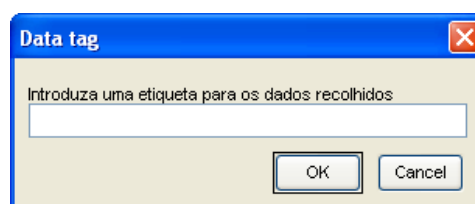


Figura 4-17 - Introdução da etiqueta para os dados recolhidos

Durante o envio do movimento está sempre disponível um botão para parar o envio do movimento, **Figura 4-18**.



Figura 4-18 - sendGUI durante o envio de uma sequência

4.3.6 Análise de Dados

A visualização de toda a informação gerada durante a simulação ou a leitura de valores do humanóide através do *sendGUI* é efectuada através de gráficos dedicados a cada área de análise. A simulação permite gerar dados das posições das juntas, do CoG e do binário ao longo do tempo. Uma das principais funcionalidades na análise de dados é a animação do movimento criado, podendo seguir visualmente as trajetórias executadas pelos componentes do humanóide. Na **Figura 4-19** mostra-se a janela de simulação, onde é possível controlar ao nível da frame cada sequência do movimento. Esta janela disponibiliza uma *toolbar* onde é possível controlar a animação executada e introduzir no cenário os elementos auxiliares já falados (tapete, escadas, etc.).

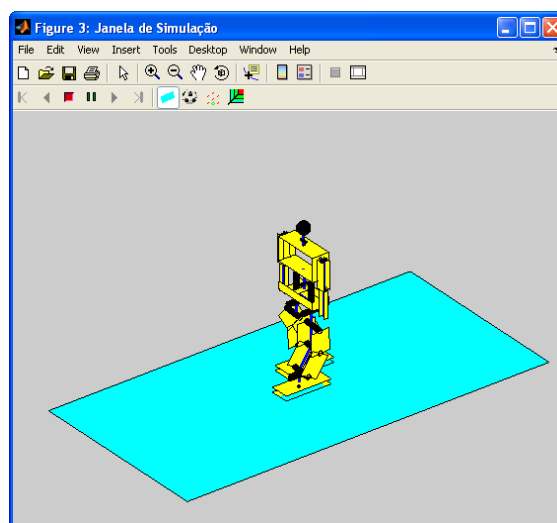


Figura 4-19 - Animação da sequência de movimentos

O *plot* das posições ao longo do tempo é adornado com faixas indicadoras de cada fase do movimento identificadas pelo seu número, **Figura 4-20**. É disponibilizado na barra de menus da janela de *plot*, sob o nome de “Servos”, a escolha das juntas que estão a ser analisadas.



Figura 4-20 - Análise da trajetória de cada junta ao longo do movimento

A análise do CoG ao longo do tempo, dado a sua utilidade como critério de estabilidade, é acompanhada por duas faixas relativas às posições do pé de suporte e do pé livre. No GUI *TwoLegs_22dof* não é possível escolher suporte duplo para nenhuma fase, dessa forma, mesmo movimentos com os dois pés assentes no solo são sempre relativos a um dos pés como sendo o pé de suporte. É por essa razão que se disponibiliza no *plot* do CoG a projecção de ambos os pés, sendo o pé de suporte indicado com uma faixa verde e o pé livre com uma faixa cinzenta mais transparente. Cabe ao utilizador fazer a distinção na análise às fases com suporte duplo.

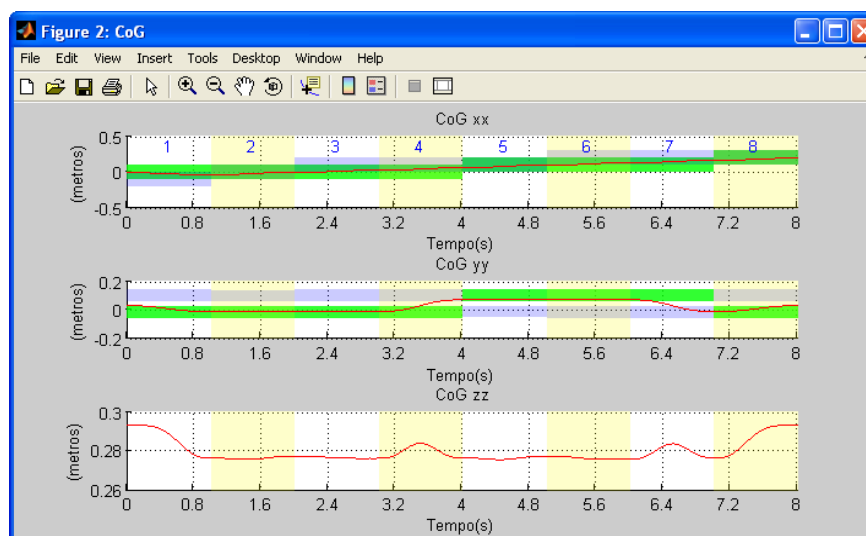


Figura 4-21 - Representação do CoG ao longo do tempo

O binário, à semelhança da representação das posições também conta com um menu, “Servos”, onde se pode seleccionar as juntas que se quer visualizar, **Figura 4-22**.

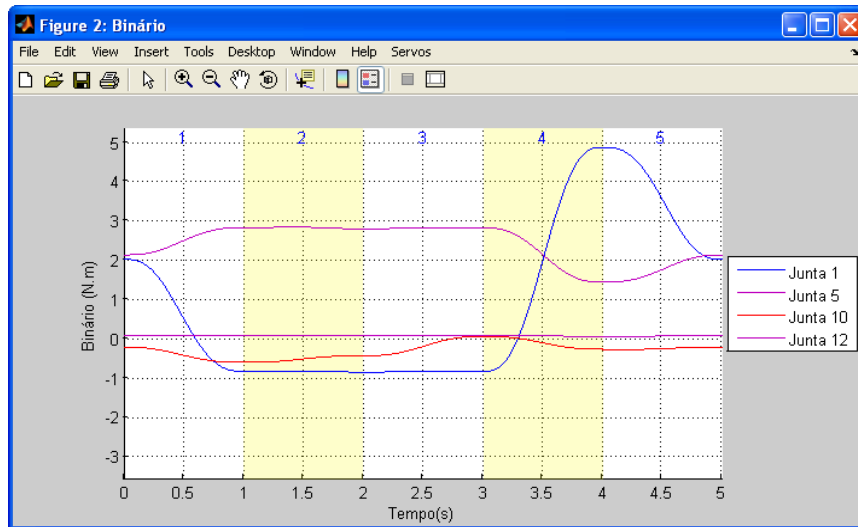


Figura 4-22 - Análise do binário em sentido estático ao longo do movimento

Na **Figura 4-23**, está um caso de comparação entre a posição lida no robô humanóide durante a aplicação de determinado movimento e a posição simulada. Na barra de menus há dois itens dedicados a este *plot*, o primeiro é o menu “Servos” onde se indica qual a junta que se quer analisar, o segundo, é o menu “Dados em Análise”, onde se selecciona a etiqueta dos dados que se recolheram no *GUI sendGUI*.

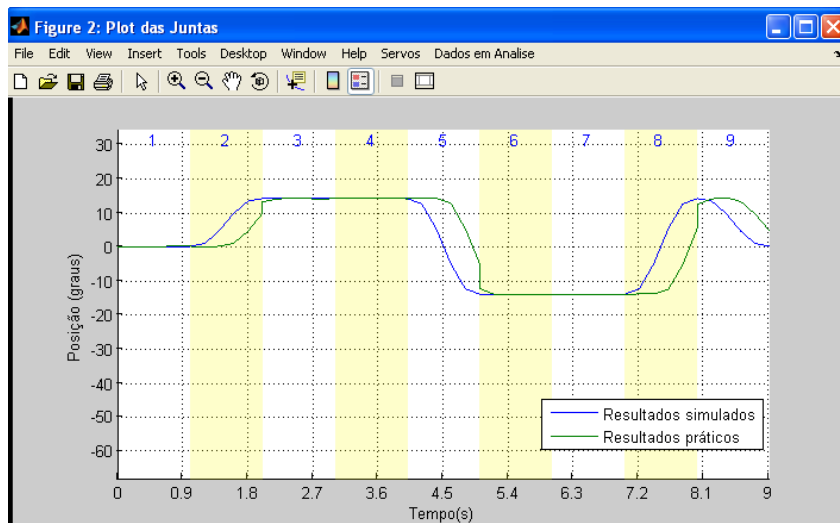


Figura 4-23 - Análise das posições lidas durante um movimento

4.3.7 Configuração

É essencial que uma aplicação suportada pela modulação de elementos reais permita a alteração desta informação facilmente, sem ter que alterar o código de desenvolvimento. Foi a pensar nas alterações que se podem fazer no robô humanóide, servomotores ou mesmo ao protocolo de comunicações que se criou uma janela de configuração de alguns dos parâmetros associados a estes três componentes.

Na **Figura 4-24** está o aspecto geral da janela de configuração. Mais à frente irá ser apresentada a organização estrutural da interface gráfica mas convém referir neste ponto que esta janela de configuração é integralmente construída por algoritmos de disposição de elementos *GUI*. A razão por detrás desta escolha centra-se na dificuldade de trabalhar a quantidade de elementos *GUI* que se apresenta nesta janela na ferramenta de construção de *GUIs Matlab, GUIDE*. Outra das razões está na contínua procura de soluções automatizadas de apresentação e gestão de dados. A solução encontrada ainda padece de especificidades direccionadas para casos particulares fugindo um pouco à ideia por detrás da automatização mas está em boas condições de garantir num futuro, a análise aos dados a apresentar e automaticamente apresentá-los ao utilizador na melhor disposição possível e usando os elementos visuais mais adequados para cada tipo de dados.

Actualmente a interface dispõe dos seguintes parâmetros configuráveis:

1. Dimensões - Parâmetros dimensionais do humanóide
 - a. Massas - Massas dos componentes (pé, tornozelo, ...);
 - b. Comprimentos - Comprimentos dos elos;
 - c. CoG Estático - CoG medido de acordo com o referencial explicado na **secção 2.2.1**.
2. Servos - Parâmetros relativos aos servomotores
 - a. Limites - Limites máximo e mínimo de actuação;
 - b. Nrel - Relação de transmissão das polias;
 - c. OffSet - *Offsets* “estáticos” dos servomotores;
 - d. Sinal - Sentido de rotação.
3. Comunicação - Parâmetros da comunicação *RS232*.

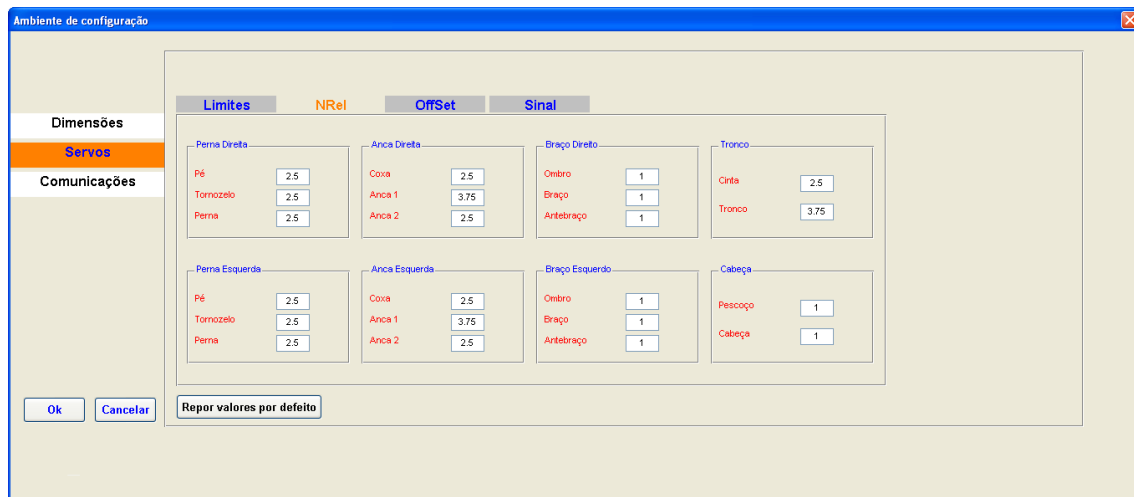


Figura 4-24 - Painel de Configuração

Além do menu de configuração dos dados de suporte base do simulador cinemático, existe uma segunda janela de configuração dedicada aos elementos visuais do cenário do *plot* humanóide, Figura 4-25.



Figura 4-25 - Configuração do *RoboPlot*

4.3.8 Menus

Quando se fala em menus refere-se à panóplia de comandos à distância de um clique organizados segundo determinado critério. Há que distinguir desde já dois tipos diferentes de menu, são eles os menus tradicionais expostos numa barra de menus e de utilidade e acesso geral e os menus de contexto, geralmente invocados através do clique com o botão direito de rato em determinada área da interface gráfica. O *TwoLegs_22dof* integra ambos os tipos de menus, em primeiro lugar irão ser mostrados os comandos da barra de menus.

Barra de Menu

A barra de menus pormenorizada na **Figura 4-26**, revela quatro áreas de funcionamento distintas, mais o menu “Ficheiro” e o menu “Ajuda”.



Figura 4-26 - Barra de menu

Ficheiro

1. Novo - Criação de um novo movimento;
2. Abrir - Abrir um movimento previamente guardado sob o formato “2ls”;
3. Guardar - Guardar o movimento actual (formato “2ls”);
4. Sair - Sair da interface gráfica.

Movimento

1. Inserir - Inserir uma nova fase na posição actual do movimento;
2. Remover - Remover a fase actual;
3. Copiar - Copiar a fase actual;
4. Colar - Colar as posições em memória.

Comunicação

1. Ligar - Iniciar uma linha de comunicação com o robô humanóide;
2. Porta de comunicação - Definir a porta série para comunicação RS232;
3. BaudRate - Definir o *BaudRate* para a comunicação;
4. Enviar Fase Actual - Enviar a fase actual do movimento para o humanóide;
5. PWM - Alterar o estado do *PWM* das *slaves*;
 - a. Activar Tudo;
 - b. Desactivar Tudo;

- c. Perna Direita;
- d. Anca Direita;
- e. ...

Ferramentas

1. Gerador de Plots - Visualização dos resultados simulados;
 - a. Juntas - *Plot* das juntas;
 - b. CoG - *Plot* do CoG;
 - c. Binário - *Plot* do Binário;
2. Adapt - Permite alterar as matrizes de *offset* de modo a definir um novo zero;
3. Robot Sender - Invoca a interface sendGUI;
4. Analisar Dados - *Plot* comparativo das posições lidas com as posições simuladas;
5. Simulação - Animação do movimento criado;
6. Configuração - Janela de configuração de algumas variáveis residentes.

RoboPlot

1. Vistas - Permite alterar o ângulo de observação do robô humanóide;
 - a. Por defeito;
 - b. De frente;
 - c. De trás;
 - d. ...
2. Captura - Guarda a imagem actual do robô humanóide;
3. Componentes - Permite seleccionar os elementos visuais que se pretende visualizar no cenário do *plot* humanóide;
 - a. Tapete;
 - b. Bola;
 - c. ...
4. Robô - Permite seleccionar os componentes do robô que se pretende visualizar no *plot* humanóide;

- a. Activar tudo;
- b. Desactivar tudo;
- c. Endoesqueleto;
- d. Exosqueleto;
- e. ...

5. Configuração - Entra na janela de configuração dos elementos visuais do cenário do *plot* humanóide (tapete, bola, ...).

Ajuda

1. Debug - Dá acesso à estrutura de controlo de toda a interface gráfica (*handles*).

Menus de Contexto

O menu de contexto das juntas foi criado especialmente para ter um controlo independente de cada uma das *slaves* existentes no humanóide e representadas na interface gráfica pelo nome do componente actuado por essa *slave*. Este menu pode ser acedido com o botão direito do cursor em qualquer um dos painéis internos do painel de “Definição de Movimento”. Neste menu podem-se distinguir duas vertentes, uma pertencente à criação de movimento e outra de envio de comandos para o humanóide.

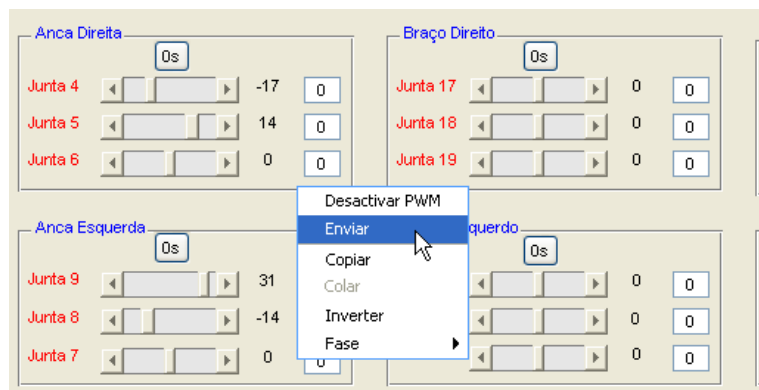


Figura 4-27 - Menu de contexto das juntas

- Activar/Desactivar PWM – Altera o estado do PWM na *slave* seleccionada;
- Enviar – Enviar as posições para *slave* a *slave* seleccionada;

- Copiar – permite copiar o conjunto de valores das juntas da *slave* actual;
- Inverter – realiza a inversão dos valores presentes nas juntas da *slave* actual;
- Fase – serve para colocar na fase actual as posições das juntas que pertencem a outra fase de movimento desta *slave*.

4.4 Estruturação da Interface Gráfica

A estruturação de uma interface gráfica com estas funcionalidades, requer um nível de abstracção e de modularidade significativo para tornar a sua programação e compreensão proporcionalmente melhores. Esta interface gráfica, espera-se, será sempre alvo de melhoramentos, introdução de ferramentas e estará sempre a acompanhar o desenvolvimento futuro do projecto *PhUA*. Por isso, apesar de o desenvolvimento até ao seu estado actual ter sido consumado de forma singular, no futuro mais pessoas poderão estar envolvidas na sua actualização e o requisito da compreensão por outros torna-se indispensável. Na secção anterior mostrou-se a apresentação gráfica da interface, os elementos visuais que o utilizador terá de enfrentar para produzir resultados. Nesta secção falar-se-á de como os elementos são actuados e de como toda a interface se articula no sentido de dar resposta às necessidades do utilizador.

Os objectivos fornecidos para a criação do *TwoLegs_22dof* dão logo o mote para duas áreas distintas, a simulação e a comunicação com o robô humanóide, onde a comunicação envolve o envio de posições, parâmetros de controlo e leitura sensorial do humanóide. A simulação envolve todos os processos de cálculo cinemático, padrões de locomoção, CoG e binários, que produzem a animação virtual e o *plot* de dados. A comunicação com o humanóide envolve o grande alicerce do protocolo de comunicações. Na **Figura 4-28** está representado numa perspectiva muito abrangente da estruturação da interface gráfica. Foram criados 3 interfaces, um interface de comunicações, que pretende não só abstrair a interface gráfica das funções de comunicação mas também enfrentar situações de conflito e gerir pequenas anomalias na comunicação. Um outro interface, que faz a ponte com os dados relativos a tudo o que rodeia a interface gráfica e um terceiro, responsável pelo *plot* dos dados simulados e recebidos do humanóide. Este último pode ser visto mais como um utilitário, uma ferramenta do *TwoLegs_22dof* que mostra ao utilizador em janelas próprias os dados em questão.

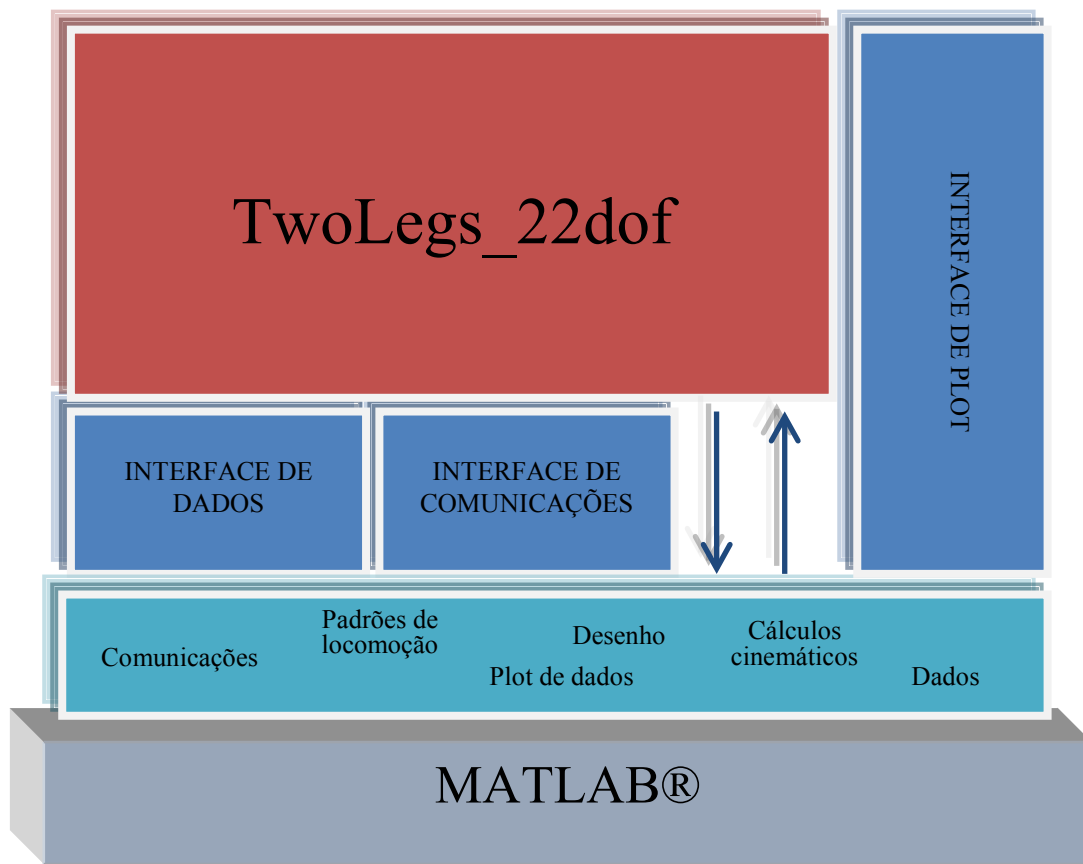


Figura 4-28 - Estruturação da Interface em Matlab

Há todo um conjunto de funções, representadas no segundo nível da **Figura 4-28** que perfazem o resto das bases em que assenta o TwoLegs_22dof.

O *layout* da interface foi realizado recorrendo à ferramenta GUIDE do Matlab, ferramenta essa que complementarmente ao diagrama visual cria um script pronto a acolher os diversos *callback* invocados pelos elementos visuais da interface gráfica. Quando é iniciado o TwoLegs_22dof, é anexada uma estrutura de dados com todos os *handlers* internos, quer isto dizer que todos os elementos inseridos na interface, além de uma etiqueta legível para identificação do elemento têm um número que, para o Matlab, identifica aquele objecto. Essa estrutura, *handles*, contém portanto a referência de todos os objectos criados pela interface gráfica e também quaisquer dados que o programador quiser vincular à aplicação. (Quando no TwoLegs_22dof se activa o comando *Help/Debug* o que acontece é o Matlab dar acesso directo a essa estrutura, *handles*.) Todos os dados analisados na **secção 2.2** estão incluídos em campos do *handles*, assim como outra informação adicional, relativa por exemplo aos padrões de locomoção ou às dimensões dos

elementos visuais do *plot* humanóide. Pode-se então observar na estrutura *handles* os seguintes campos:

- paineis - O GUIDE/Matlab não possui o elemento visual *tabPanel*, daí que de forma a contornar o problema se tenha criado esta estrutura que guarda células com as *tags* (etiquetas) dos painéis e das caixas de texto que os seleccionam;
- data - Estrutura gerada pelo *datInterface* e que possui todas informações já referidas;
 - dim - dados dimensionais do humanóide;
 - ser - informações sobre os servomotores;
 - com - estrutura de apoio à comunicação;
 - mov - padrões de locomoção;
 - des - informações dos elementos visuais de ornamento do *plot* humanóide (tapete, bola, escadas);
 - sim - modelo para a estrutura de simulação;
 - plot - modelo para a estrutura de plo;
- cart - Os dados Cartesianos de todos os elementos que estão a ser visualizados no *plot* humanóide;
- gui - No decorrer da aplicação e de forma a facilitar a sua gestão é necessário ter *handlers* de objectos que partilham algo em comum, nesta estrutura estão inseridas essas relações;
- mix - Mistura de dados ligados a diferentes áreas, por exemplo a função de copiar guarda nesta estrutura os dados copiados, assim como por exemplo o endereço do directório que guarda imagens que utilizador captura também está aqui inserido;
- sim - Estrutura de simulação, que serve de suporte à informação de cada fase do movimento;
 - fase - indica a fase actual;
 - q - matriz com todas as posições de todas as fases do movimento;
 - Ref - matriz de transformação relativa ao pé suporte;
 - pe - pé de suporte para cada fase (1 ou 0);
 - duracao - duração de cada uma das fases;
 - dependencia - matriz 22x22 com as dependências entre juntas;
- mon - Estrutura de monitorização;
- mov - Os dados dos padrões de locomoção em execução estão aqui descritos;
- plo - Estrutura com dados simulados, prontos a serem exibidos nos *plots* gerados pelo *plotInterface*.

Estão listados todos os campos de 1º nível inseridos na estrutura *handles* e alguns campos mais pertinentes de 2º nível. A estrutura “data” deve ser entendida como uma base de dados estáticos, de suporte à aplicação e a fonte primária dos dados dinâmicos. Quer isto dizer que está na estrutura “data” definida a estrutura de simulação com os valores por defeito e sempre que o *TwoLegs_22dof* inicia uma nova folha de simulação carrega desta estrutura a informação para a estrutura “sim”.

4.4.1 Interfaces

Foram apresentadas na **Figura 4-28**, três interfaces usadas pelo *TwoLegs_22dof*, são elas:

- Interface de dados, *datInterface*;
- Interface de plot, *plotInterface*;
- Interface de comunicações, *comInterface*.

A interface de dados tem a função de carregar para a aplicação toda a informação existente na estrutura “data”. O modelo humanóide, os padrões de locomoção, os elementos visuais são todos definidos por dados carregados pelo *datInterface*. Além desta função de abertura, esta interface é responsável pela rotina de alteração desta informação, *confGUI*. E por último faz a salvaguarda dos campos relevantes para a próxima utilização. A operação desta interface é executada através de três funções disponibilizadas para o exterior, são elas:

- *init* - devolve a estrutura de dados inicializada;
- *conf* - inicia e gere o *confGUI*;
- *close* - guarda as informações relevantes para a próxima inicialização.

Chamadas através do seguinte comando:

```
>>[data, task] = datInterface([func], varargin),
```

Onde *varargin* depende da função especificada em *func*.

Esta interface tem a peculiaridade de passar para os *scripts* de dados responsabilidade de gestão. Isto é, esses *scripts*, quando são criados devem fornecer indicações do tipo de dados possíveis para os seus campos e/ou um algoritmo para criação e gestão de um painel de configuração dos seus próprios dados. Desta forma, quando a interface de dados executa o *confGUI*, invoca esses algoritmos definidos nos próprios *scripts* de dados. Para o funcionamento deste processo de configuração, cada *script* deve possuir três campos reservados na sua estrutura de dados, são eles:

- *name* , onde se especifica um nome para o tipo de dados;
- *chkFcn* , *handler* de uma função de verificação da validade dos dados internos;
- *guiFcn* , *handler* da função de criação e gestão de painel de configuração dos dados.

Estes campos não são obrigatórios e só os *scripts* com dados que se pretende sejam editáveis/configuráveis é que os devem possuir. Esta é uma visão abrangente do modo de funcionamento da interface de dados e serve para ter uma ideia de como está implementada. Na **Figura 4-29** está esquematizado o modo de funcionamento desta interface.

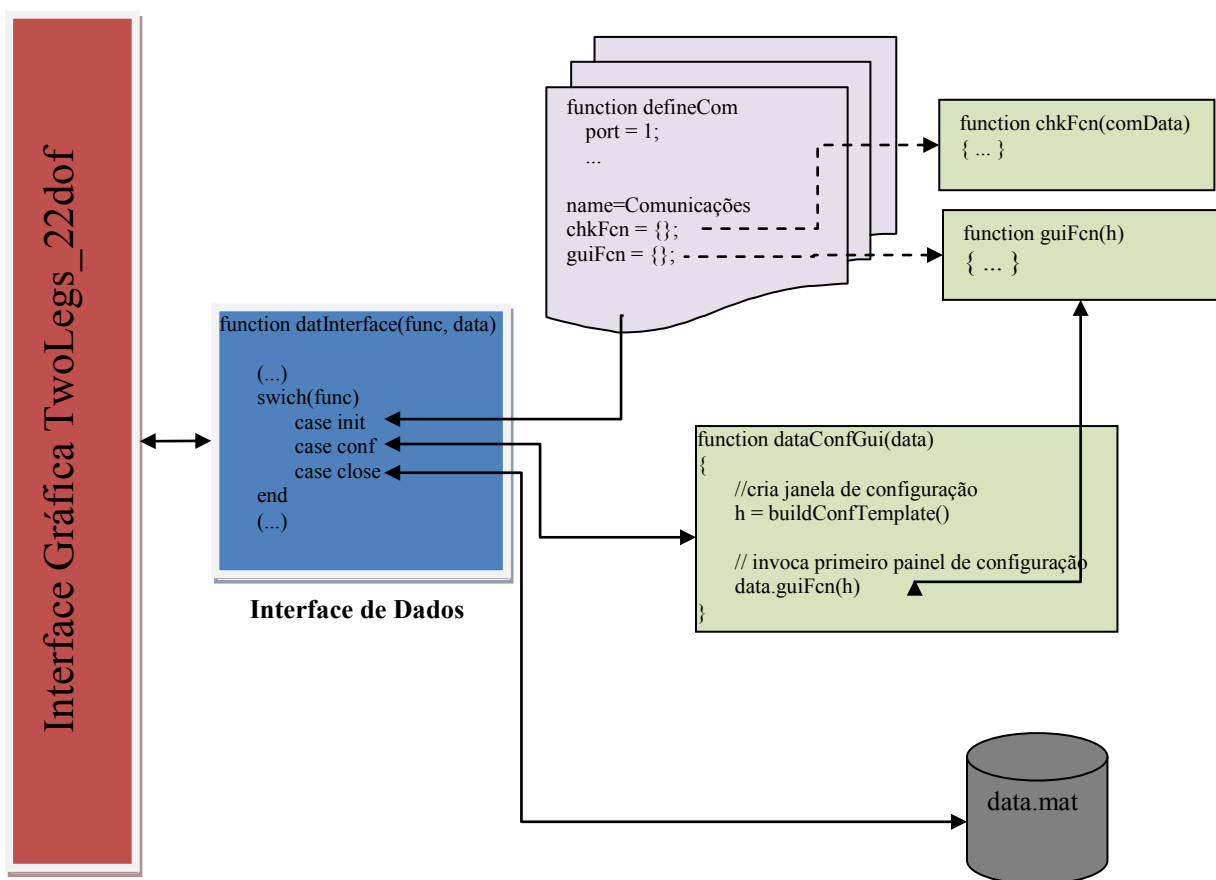


Figura 4-29 - Representação esquemática da interface de dados

A *interface de plot* (**Figura 4-30**) desempenha um papel crucial na funcionalidade de simulação, porque é esta interface a responsável pela criação de trajetórias das juntas entre as fases e subsequentes cálculos cinemáticos e binários para futura visualização dos mesmos por parte do utilizador. Disponibiliza dessa forma as seguintes funções:

- *pos* , *plot* das juntas ao longo do tempo;

- bin , *plot* do binário ao longo do tempo;
- cog , *plot* do CoG ao longo do tempo;
- cmpFcn , análise dos dados recebidos, comparando as posições recebidas e as simuladas.

Esta interface realiza todos os cálculos temporais, e guarda os valores gerados numa estrutura designada por “plo”. Esses valores podem depois ser visualizados através das funções disponibilizadas pela interface, mas existe uma função externa à interface que no entanto faz uso dos valores calculados por esta. É a função de animação de movimento. A forma de actuar a interface obedece ao seguinte comando:

```
>>[out, task] = plotInterface(data, [func], varargin);
```

Onde, *data* representa o campo de dados da estrutura *handles*, *func* é opcional e designa o *plot* que se pretende gerar e em *varargin* definem-se variáveis dependentes da função invocada.

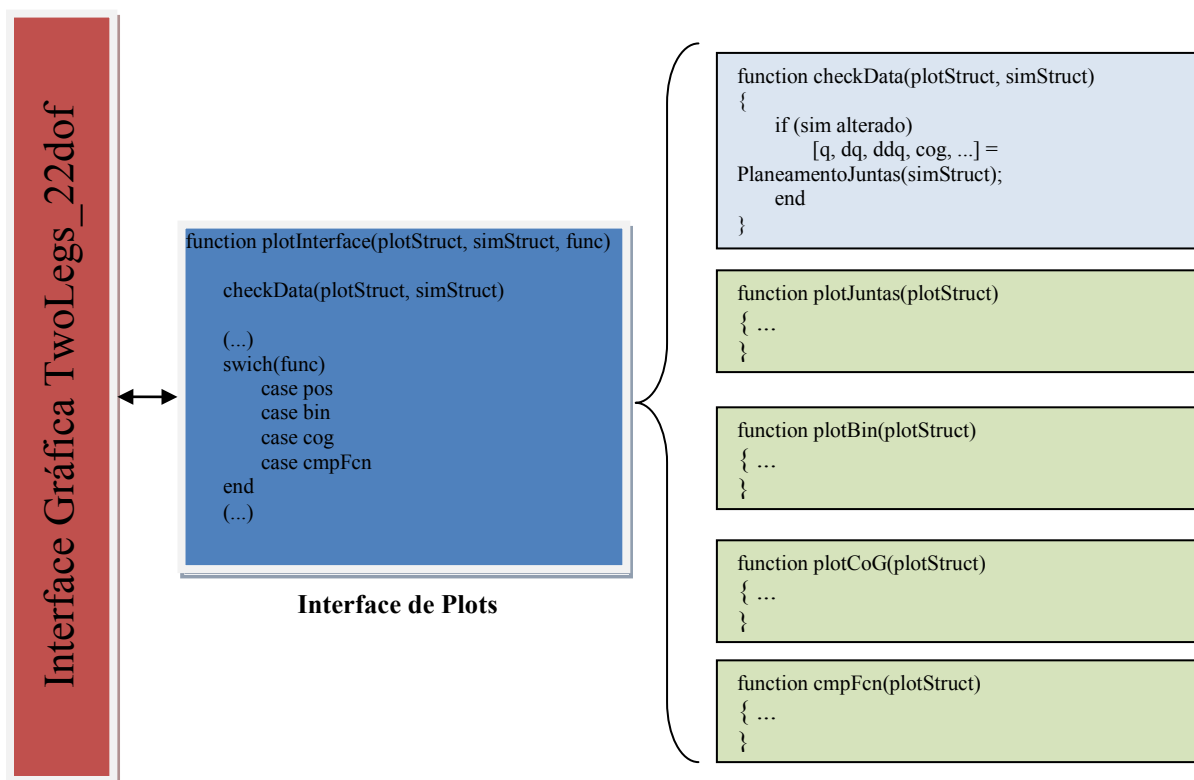


Figura 4-30 - Diagrama esquemático da interface de plot

Por último encontra-se a interface de comunicações (**Figura 4-31**). Esta pode ser vista como um agregado dos *device drivers* implementados no protocolo de comunicação (**Apêndice B**). A forma de actuar desta interface não difere muito das anteriores,

```
>>[ out, task ] = comInterface( varargin );
```

Neste caso o *varargin* é mais complexo porque torna opcional a passagem da estrutura de comunicações e também porque as funções invocadas têm ainda mais especificidades muito próprias. A opção da passagem ou não da estrutura de comunicações é possível porque esta interface possui essa estrutura numa variável do tipo *persistent*, correspondente ao tipo *static* do C e que mantém em memória a variável mesmo depois da interface já ter sido finalizada. Mas para garantir a correcta execução desta interface convém passar sempre como parâmetro a estrutura de comunicações, salvo algumas excepções.

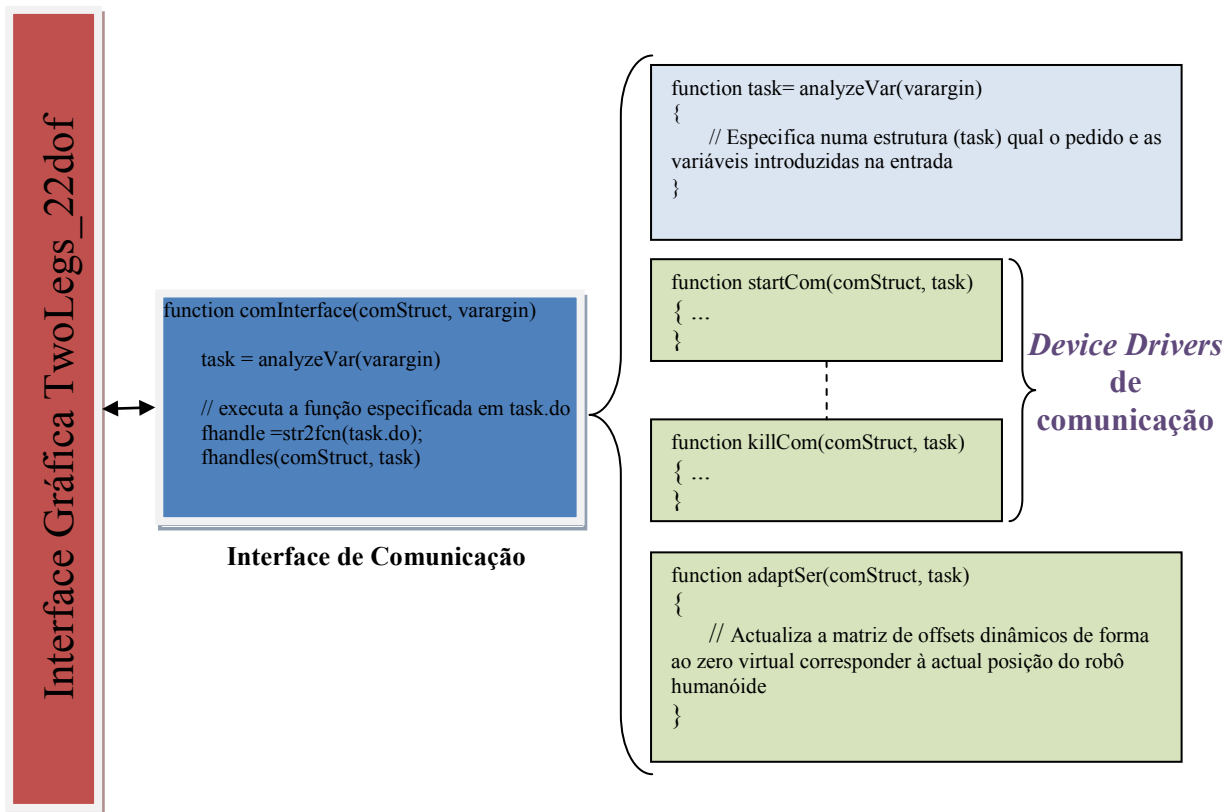


Figura 4-31 - Diagrama esquemático da interface de comunicações

Esta abordagem global das interfaces dá uma ideia da sua utilidade, do seu funcionamento sem no entanto entrar em grandes pormenores, mas consultando o *help* de cada uma delas obtém-se uma descrição detalhada de como as operar (>>*help [nome da interface]*).

4.4.2 Processos

Já se determinaram quais as funcionalidades da interface gráfica e qual o *layout* disponível, assim como que campos de dados estão presentes no decorrer da aplicação. Falta por isso mostrar de que forma as principais características da interface são desencadeadas, processadas e propagadas.

No *script* principal há funções denominadas “isoladas” e outras “continuadas”, quer isto dizer que há determinadas funções de *callbacks* independentes, enquanto outras tem repercussões em diferentes áreas. Estas funções distinguem-se pelo destino final que dão à informação produzida. Isto é, o Matlab possui uma função especial para guardar a estrutura *handles* do GUI,

```
>>guidata(hObject, handles),
```

Onde *hObject* é qualquer objecto integrado na interface e *handles* a estrutura que se quer guardar.

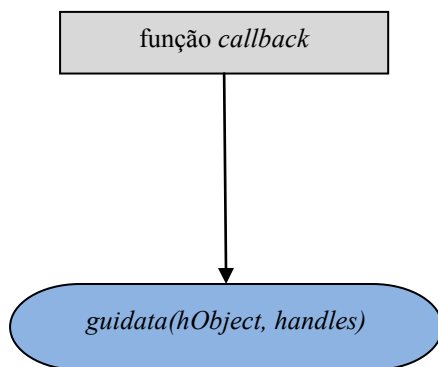


Figura 4-32 - Função isolada

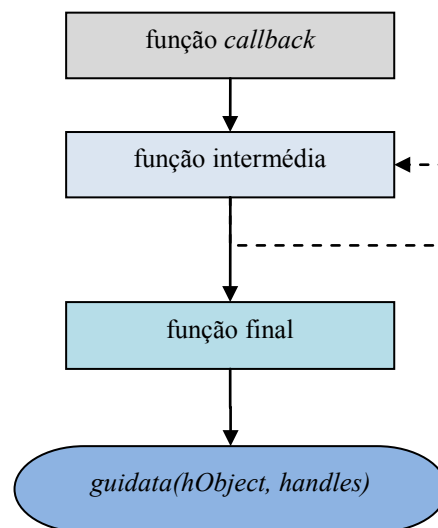


Figura 4-33 - Função com continuidade

Atenção, porque existe um terceiro tipo de funções que não produz informação, apenas actua em elementos visuais do GUI, não necessitando por isso de executar a função *guidata* no seu final.

Em seguida dá-se uma lista extensiva das funções intermédias e final disponíveis:

Função intermédia	Descrição
novo_movimento(handles)	Cria ou limpa as estruturas associadas à simulação e monitorização. Resumindo, cria uma “nova folha” de simulação
correct_sliders(handles)	Coloca os <i>sliders</i> das juntas de acordo com a fase actual.
actualiza_painelFase(handles)	Faz a actualização dos valores mostrados no painel de controlo de fase.

Tabela 4-3 - Lista de funções "intermédia"

Função final	Descrição
ActualizaMonitor(handles)	Lê os últimos valores sensoriais recebidos pela interface de comunicações e actualiza todos os mostradores dos dados monitorizados
actualiza_cart_coord(handles)	Faz os cálculos cinemáticos e binários para a actual fase do movimento. Actualiza o painel de binários, o <i>plot</i> do humanóide e os sliders dos valores Cartesianos no painel de “Movimentos Especiais”

Tabela 4-4 - Lista de funções "final"

É de referir que a função final *actualiza_cart_coord* efectua os cálculos cinemáticos só para a configuração que a fase apresenta, quer isto dizer, que o planeamento de trajectória e os cálculos cinemáticos associados a essa trajectória ficam a cargo da interface de *plot*, que idealmente deveria ser uma *thread* independente do GUI principal. Quando se altera o valor de uma junta, como representado na **Figura 4-34**, é recalculada a cinemática para a fase actual mas não é executado o planeamento de trajectórias.

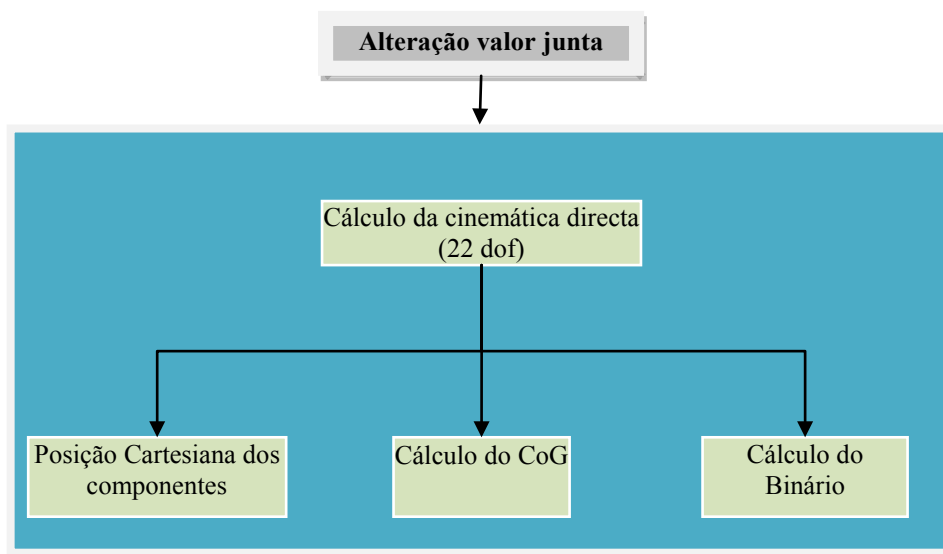


Figura 4-34 - Processo desencadeado pela alteração de uma junta

Só quando se muda a fase do movimento em que se está a trabalhar é que é invocada a interface de *plot* para recalculer o planeamento de trajectórias e cálculos anexos. Este cálculo, idealmente só seria efectuado quando se chamasse alguma das funções de *plot* de dados mas como os cálculos do CoG e do binário são computacionalmente pesados, para um movimento com mais de 20 fases seria maçante para o utilizador ter de esperar pela finalização dos cálculos. Desta forma, esta rotina é chamada a cada fase que se adiciona, progredindo de uma forma quase imperceptível para o utilizador na criação dos dados que mais tarde poderão vir a ser pedidos.

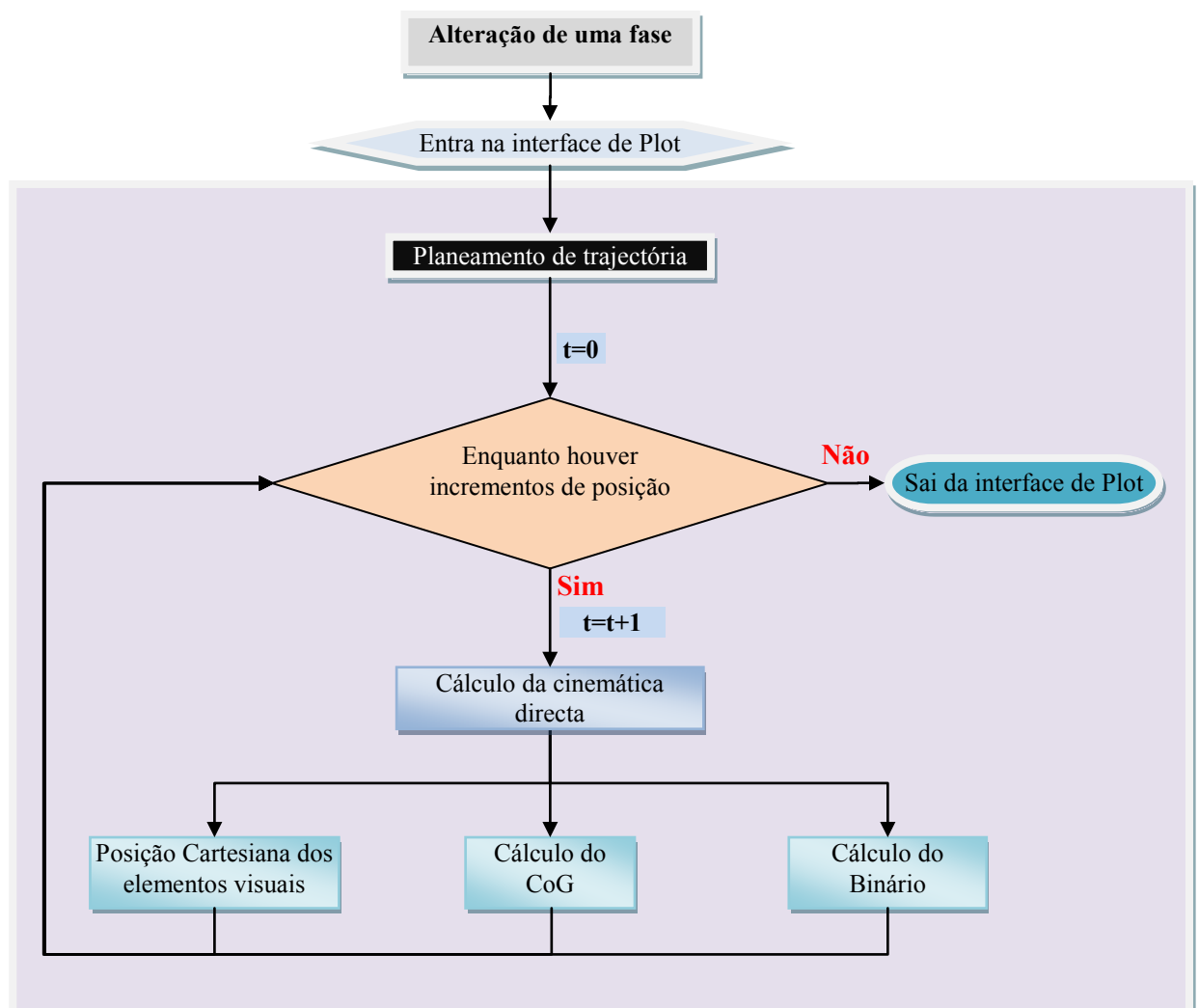


Figura 4-35 - Cálculo de dados para plot

Outro dos processos relevantes e que merece uma análise mais cuidada é o envio e leitura de dados através do *sendGUI*. Já foi mostrada a apresentação deste GUI na secção anterior. Quando se ordena nesta interface o envio de determinado movimento, é criado um objecto *timer* para controlo do tempo de amostragem e de envio. Antes de avançar vai-se mostrar em seguida uma lista de

propriedades deste tipo de objecto. À semelhança dos outros elementos no Matlab, este objecto é identificado por um *handler* e possui a grosso modo as seguintes características:

Campo	Descrição
Execution Mode	Modo de funcionamento – permite especificar um modo periódico ou um modo <i>single-shot</i>
Period	Período da interrupção
TasksToExecute	Número de tarefas a cumprir (só disponível em modo de funcionamento periódico)
TimerFcn	Função de <i>callback</i> do <i>timer</i>
StartFcn	Função que executa ao iniciar
StopFcn	Função que executa quando é parado

Tabela 4-5 - Descrição do *timer*

O *timer* funciona à base de interrupções, quando definido com as funções todas, gera uma interrupção quando inicia, outra de cada vez que atinge o período especificado e uma última quando pára. Pode ser parado por chegar ao fim o número de interrupções pedidas ou através da invocação da função *stop(timer)* por alguma outra função.

Depois desta introdução ao *timer*, é mais fácil perceber como é que funciona a rotina de envio e leitura no *sendGUI*. Existe portanto, um campo do *sendGUI* que especifica se se pretende fazer leituras enquanto se envia o movimento. Assumindo que não, define-se o *timer* da seguinte forma:

Campo	Valor
Execution Mode	<i>Single shot</i>
Period	Duração da 1ª fase do movimento
TasksToExecute	- vazio -
TimerFcn	- vazio -
StartFcn	Envia a 1ª fase do movimento
StopFcn	Verifica se há mais fases e programa o relógio para mais uma fase

Tabela 4-6 - Definição do *timer* (1º caso)

E passa a obedecer ao seguinte fluxograma, **Figura 4-36**. As linhas a tracejado indicam que não há um fio condutor real entre as funções, já que elas são desencadeadas por interrupções.

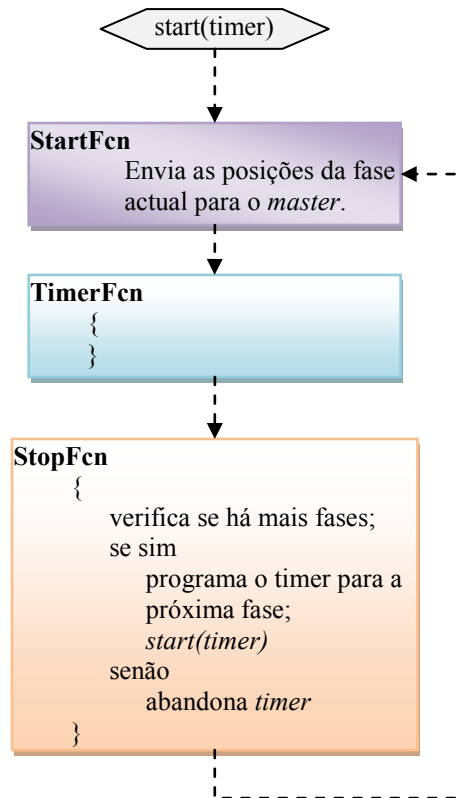


Figura 4-36 - Fluxograma de envio de posições

Para o caso de estar seleccionado o campo para leitura de valores sensoriais, a configuração do *timer* passa a ser a seguinte:

Campo	Valor
Execution Mode	fixedRate – periódico
Period	100ms (período de leitura)
TasksToExecute	Duração da 1ª fase do movimento multiplicado pela frequência de leitura ($t_f \cdot 10$)
TimerFcn	Faz uma leitura sensorial
StartFcn	Envia a 1ª fase do movimento
StopFcn	Verifica se há mais fases. Caso afirmativo programa o <i>timer</i> para a próxima fase

Tabela 4-7 - Definição do timer (2º caso)

E o fluxograma sofre uma pequena alteração (**Figura 4-37**).

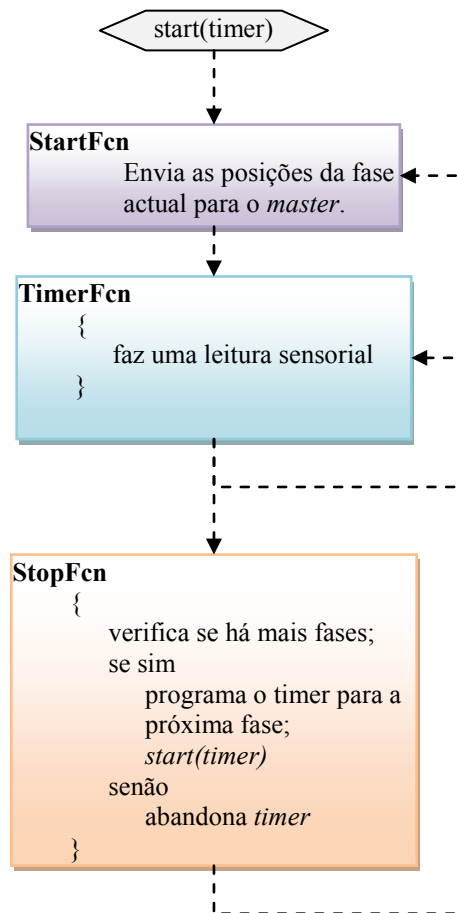


Figura 4-37 - Fluxograma de envio de posições

De reparar que a diferença entre os dois modos possíveis é que o segundo obriga o *timer* a executar interrupções periódicas geradas à taxa de 10 por segundo (de 100 em 100ms), para efectuar a leitura sensorial.

Um dos processos de grande importância no que concerne à comunicação é a função *adapt*. Esta é uma função de grande utilidade quando se trata de aplicar movimentos simulados ao robô real ou transformar posições lidas do robô real em posições do simulador. O que é executado no interior desta função é uma adaptação de posições associadas à estrutura virtual em posições aplicáveis à estrutura real. Como parâmetros indispensáveis desta função estão as relações de transmissão, os sentidos de rotação, os *offsets* estáticos, *offsets* dinâmicos e a excursão máxima e mínima das juntas. Quando se trata de enviar uma posição para o robô a fórmula de cálculo é a seguinte:

$$posRobo = posVirtual \cdot \sin alRotac\tilde{a}o \cdot relTransmiss\tilde{a}o + offsetEst\tilde{a}tico + offsetDin\tilde{a}mico \quad (4-1)$$

No caso de se efectuar uma leitura do robô humanóide, para convertê-la no sistema usado no simulador usa-se a seguinte fórmula:

$$posVirtual = \frac{(posRobo - offsetEst\tilde{a}tico - offsetDin\tilde{a}mico)}{relTransmiss\tilde{a}o} \cdot \sin alRotac\tilde{a}o \quad (4-2)$$

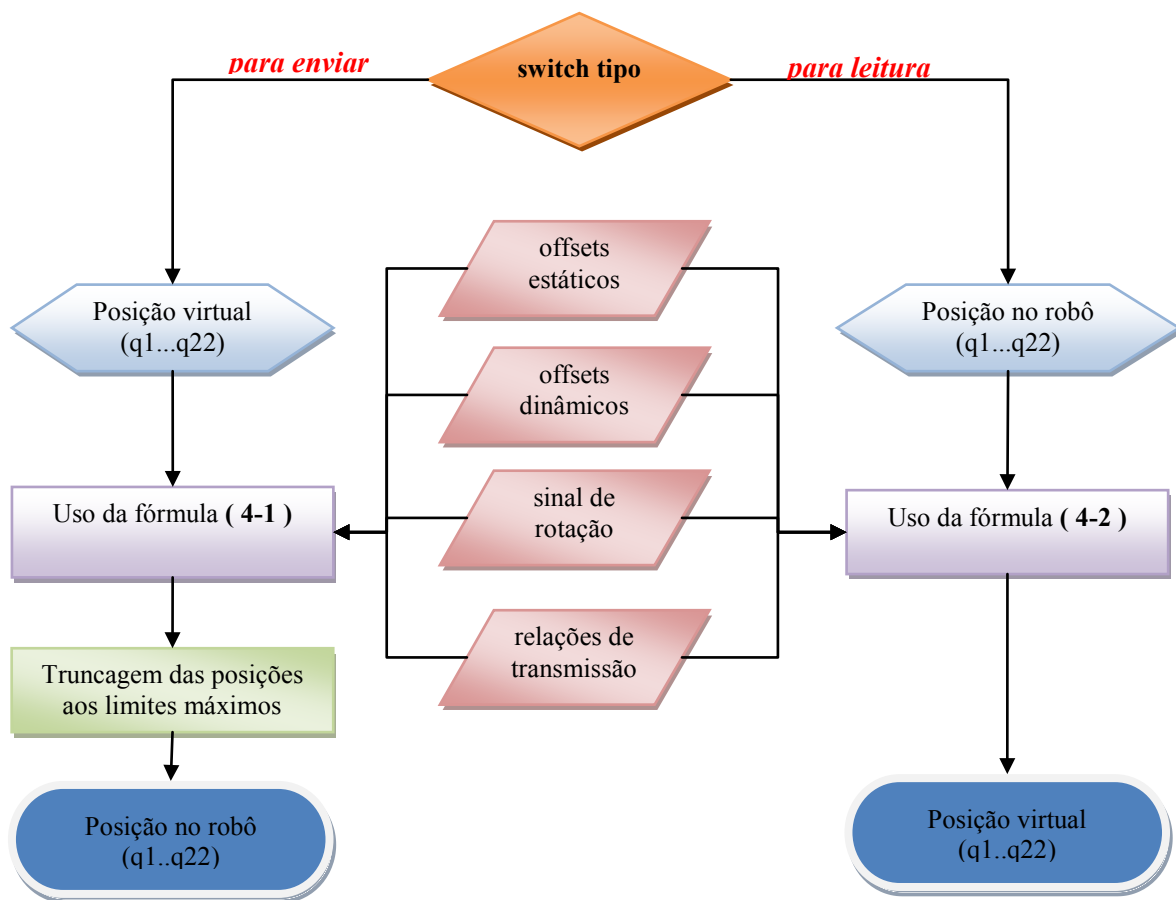


Figura 4-38 - Processamento interno da função *adapt*

A utilidade da função *adapt* ao converter posições entre as duas realidades também serve para calcular os limites dos *sliders* de posição das juntas na interface gráfica. Estes valores estão especificados em termos do robô real, ou seja na excursão máxima dos servomotores, sendo necessário contemplar *offsets* estáticos e dinâmicos, sinais de rotação e relações de transmissão. Dessa forma, ao introduzir na função *adapt* como “posição no robô” (variável de entrada) a excursão máxima, por exemplo, na saída obter-se-á a “posição virtual” que corresponde aos limites máximos dos *sliders* das juntas na interface gráfica.

4.5 Discussão de Resultados

As especificações definidas nos objectivos do trabalho para a interface gráfica foram satisfeitas. De facto, a interface permite ir um pouco mais além ao introduzir ferramentas redundantes como o *sendGUI* (interface gráfica de envio de movimentos para o robô). A avaliação dos resultados centra-se por isso na implementação prática.

A monitorização do robô humanóide está limitada a uma frequência máxima de 10 actualizações por segundo e ainda assim, compromete a fluidez da interface ao ocupar 12% do processamento em trocas de informação com o *master* do robô. Este ponto merece especial atenção pela importância que representa a monitorização e pela necessidade de se ter uma interface célere e reactiva. Apontaram-se duas soluções: (i) incluir o processo de comunicações com um robô numa *thread*; ou (ii) alterar o protocolo de comunicações para permitir o envio de toda a informação de todas as *slaves* de uma só vez eliminando o *overhead* de pedido *slave* a *slave*, parâmetro a parâmetro. A primeira das soluções não é possível na versão do Matlab usada para a implementação da interface gráfica e a segunda necessitaria de algum tempo para alterar a implementação do protocolo de comunicações no PC e no *master*.

Outro inconveniente existente é o processo de planeamento de trajectórias e respectivo cálculo cinemático. Durante a especificação das fases do movimento não há necessidade de calcular o planeamento de trajectórias porque o utilizador não precisa nem se apercebe da sua existência mas, no momento em que o utilizador pretenda observar os *plots* da trajectória, do CoG, do binário ou mesmo a animação do movimento criado, a trajectória planeada já faz falta. Se só se efectuasse o planeamento de trajectórias quando o utilizador invocasse essas funções e se o movimento fosse composto por mais de 20 fases haveria muito tempo de espera durante esse cálculo. A implementação usada evita isso, ao forçar o planeamento de trajectórias sempre que o utilizador insere uma nova fase e dessa forma a trajectória é calculada gradualmente tornando-se quase imperceptível para o utilizador. Mas no caso de se alterar a especificação de uma fase do movimento passada, no meio de dezenas de fases, vai ter de se calcular o planeamento de tudo, desvirtuando a solução implementada. Mais uma vez o uso de uma *thread* resolveria o problema, mas na impossibilidade de se implementar uma criou-se um algoritmo que aproveita os cálculos do planeamento já efectuados em fases que não foram alteradas cingindo o novo cálculo só às fases que foram realmente modificadas. Ainda assim este algoritmo carece de uma procura no caso de haver um *shift* das fases.

A configuração dos parâmetros da interface é feita numa janela própria criada pela interface de dados. O método usado distribui a responsabilidade de apresentação dos dados pelos próprios *scripts* fonte desses dados. Quer isto dizer que sempre que o programador introduz novas variáveis no modelo humanóide ou na interface e se pretender que elas sejam configuráveis têm de especificar também um algoritmo que apresente e faça a gestão do painel de configurações dos seus próprios dados. Esta forma de delegar funções para os *scripts* de dados tem as suas vantagens e desvantagens. Primeiro, a ideia é a interface gráfica oferecer um espaço para configuração de dados mas é da responsabilidade do *script* que possui esses parâmetros criar um *layout* de apresentação ao utilizador. Esta abordagem faz todo o sentido ao possibilitar que, de uma forma *plug&play*, novos dados introduzidos estejam ao dispor do utilizador sem ter que se alterar o código da interface gráfica mas, por outro lado, obriga a que o painel de configuração seja construído através de um algoritmo ao invés de uma ferramenta visual. Este inconveniente torna esta funcionalidade complicada de usar.

5 Conclusão

5.1 Discussão de Resultados

O projecto de uma interface de controlo de um sistema complexo como o robô humanóide envolve a cobertura de diferentes áreas de conhecimento científico. Este trabalho começou com um enquadramento geral do projecto em que se inseriu, uma abordagem simples à estrutura mecânica, aos actuadores, à arquitectura de controlo e à electrónica do robô humanóide. Neste ponto de grande importância, já que se estão a desenhar as bases do projecto, há que estabelecer a fronteira entre a realidade e a parametrização dessa realidade.

Analisou-se o estado da arte do planeamento do movimento e apresentaram-se algumas soluções para os problemas que a cinemática inversa coloca. A discussão neste ponto passa pela escolha entre um método analítico, que é directo e total, porque expõe todas as soluções através de cálculo directo, ou um método numérico, que procura uma solução aleatória através de aproximação, mas cuja parametrização é mais simples que no caso analítico, que por vezes nem é possível realizar. Além da cinemática introduziu-se um pouco da dinâmica numa análise relativa a critérios de estabilidade.

O estudo do planeamento de trajectórias foi depois aplicado num simulador cinemático, começando por um modelo simplificado composto só por 5 DoF. Este modelo simplificado serviu para efectuar um teste comparativo entre o PVCoG e o CoP, confirmando-se que a baixas velocidades quase que se elimina os factores dinâmicos da equação de estabilidade, tornando praticamente coincidentes os dois pontos comparados. Na aplicação das soluções de cinemática inversa à estrutura completa, optou-se por simplificar o problema reduzindo a cadeia cinemática humanóide de 22 DoF para dois problemas de 3 DoF mais restrições. Partindo desta simplificação, criaram-se tarefas simples como caminhar, rodar (mudar de direcção) e pontapear uma bola.

Ainda neste ponto, foram gerados movimentos mais simples, concebidos no espaço de aplicação, espaço das juntas. A sua facilidade de implementação explica algumas das suas fragilidades e dos casos particulares em que eles realmente se inserem. Entre estes movimentos estão o baixar a anca, balançar a anca ou elevar o pé livre. Há ainda algum estudo a realizar nos algoritmos de padrões de locomoção criados, nomeadamente no movimento de rotação que apresenta alguns problemas de aplicação prática.

Finalizou-se o projecto com a criação da interface gráfica. Esta ferramenta de trabalho oferece três grandes funcionalidades para um utilizador do robô humanóide: simulação, controlo e

monitorização. O primeiro assenta sobretudo no simulador cinemático, enquanto os outros dois recorrem ao protocolo de comunicação com o robô. Pode-se definir sequências de movimento, explorar as suas características através de gráficos exclusivos, enviar e guardar os resultados da monitorização da execução desse movimento e novamente analisar, comparando os resultados esperados com os obtidos. A definição do movimento conta com a especificação de posições ao nível da junta, a adição de qualquer um dos padrões de locomoção apresentados, o auxílio dos movimentos mais simples para alguns ajustes e a visualização do binário estático empregue por cada uma das juntas. Estas simulações e especificações são sempre delimitadas pelas limitações da própria estrutura humanóide garantindo dessa forma um movimento realizável na prática. A ferramenta *TwoLegs_22dof* apresenta-se assim como uma mais-valia na simulação de movimentos a aplicar ao robô e permite um estudo aprofundado dos padrões de locomoção, criando assim condições para adaptar os movimentos ao humanóide em tempo-real. É também reconhecida a perda de eficiência causada pelo desenvolvimento da interface em Matlab.

Resumindo as características alcançadas pelo projecto da interface gráfica, podem-se enumerar as seguintes conclusões:

- O ambiente de simulação proporciona a geração de padrões de movimento que poderão ser testados e optimizados, mesmo antes da sua aplicação ao robô real;
- Os padrões de locomoção estudados e as ferramentas de simulação implementadas representam um primeiro passo no sentido do desenvolvimento de um sistema versátil associado ao planeamento do movimento;
- A interface gráfica do utilizador revela-se uma ferramenta fundamental quer na monitorização e visualização da informação sensorial como na programação e comando do robô humanóide;
- A integração de um conjunto de menus e painéis de controlo, assim como de uma vista dinâmica 3D do robô, ajudam o utilizador final a familiarizar-se com as funções e comandos básicos do sistema.

5.2 Perspectivas de Trabalho Futuro

A interface gráfica não pode ser fechada enquanto o projecto humanóide da UA continuar em evolução. Espera-se que haja sempre um *upgrade* das suas ferramentas, melhoramento de alguns algoritmos e introdução de novas funcionalidades. Por outro lado, a avaliação final da sua utilidade dependerá da sua apreciação e teste pelos utilizadores finais.

A migração da interface gráfica para outro ambiente de desenvolvimento é uma proposta aliciante mas complicada. Aliciante porque dá mais liberdade à programação da interface, eliminando algumas das barreiras impostas pelo *Matlab*. Complicada porque o *Matlab* fornece um ambiente de desenvolvimento amigável, intuitivo e recheado de funções úteis que não são facilmente exportáveis para outro ambiente de desenvolvimento.

No entanto, existe um vasto leque de melhoramentos e/ou ferramentas que se podem criar para esta interface, como por exemplo:

- Mecanismo interactivo de criação de padrões de locomoção (com especificações no espaço Cartesiano);
- Abordar a questão da transição entre diferentes padrões de locomoção e o problema do planeamento do movimento baseado em directivas de alto nível (e.g., simular situações reais da competição RoboCup);
- Permitir a transferência de dados obtidos a partir de sistemas de captura do movimento humano (ou outros) para aplicação directa no robô humanóide;
- Introduzir novos gráficos de análise, e.g., para os dados recebidos pelos sensores especiais (centro de pressão e/ou inclinómetros);
- Alterar o critério de estabilidade usado, incorporando a dinâmica do robô para o uso de critérios de estabilidade dinâmica (CoP, FRI);
- Definir algoritmos de detecção de colisões com base em polígonos simples representativos dos componentes do robô humanóide.
- Dotar a interface de capacidade de realimentação visual passível de permitir o seu uso para tele-operar o robô humanóide.

Referências

Robô Humanóide da UA

- [1]. CARDOSO, Ângelo, BEÇA, Nuno; *Desenvolvimento e Integração das Subestruturas Inferior e Superior para a Locomoção de uma plataforma humanoide – Relatório Final de Projecto*; DEM, Universidade de Aveiro; 2004/05.
- [2]. SILVA, Milton Ruas; *Desenvolvimento de Algoritmos de Controlo para Locomoção de um Robot Humanóide – Relatório Final de Projecto*; DEM, Universidade de Aveiro; 2005/06.
- [3]. GOMES, Luís; SILVA, Mauro; *Concepção e Desenvolvimento de Unidades de Percepção e Controlo para um Robot Humanóide – Relatório Final de Projecto*; DEM, Universidade de Aveiro; 2004/05.

Locomoção bípede e critérios de estabilidade

- [4]. GOSWAMI, Ambarish; *Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots*; University of Pennsylvania – USA; 1999.
- [5]. CHOI, Jun Ho; GRIZZLE, J. W.; *Planar Bipedal Walking with Foot Rotation*; American Control Conference; USA; 2008.
- [6]. CARPIN, Stefano; PAGELLO, Enrico; *The challenge of motion planning for humanoid robots playing soccer*; IEEE-RAS Intenation Conference on Humanoid Robots; Italy; Dezembro 2006.
- [7]. KUFFNER, James; NISHIWAKI, Koichi; KAGAMI, Satoshi; INABA, Masayuki; INOUE, Hirochika; *Motion Planning for Humanoid Robots*; International Symposium of Robotics Research; 2003.
- [8]. SILVA, Filipe M., *Análise Dinâmica e Controlo de Sistemas Robóticos de Locomoção Bípede*, Tese de Doutoramento; Universidade do Porto; Fevereiro de 2002.
- [9]. DENK J.; SCHMIDT, G.; *Synthesis of a Walking Primitive Database for a Humanoid Robot using Optimal Control Techniques*; IEEE-RAS Intenation Conference on Humanoid Robots; Tóquio, Japão; Novembro de 2001

Cinemática Inversa

- [10]. SILVA, Filipe; *Sistemas de Manipulação e Locomoção – Mestrado em Engenharia Mecânica*; 2002-2003.
- [11]. TOLANI, Deepak, GOSWAMI, Ambarish, BADLER, Norman I.; *Real-Time Inverse Kinematics Techniques for Antropomorphic Limbs*; Agosto 1999.

- [12].SIAN, Neo Ee, YOKOI, Kazuhito, KAJITA, Shuuji, TANIE, Kazuo; *A Framework for Remote Execution of Whole Body Motions for Humanoid Robots*; International Journal of Humanoid Robotics

Interface Gráfica

- [13].KUFFNER, James J.; KAGAMI, Satoshi; INABA, Masayuki; INOUE, Hirochika; *Graphical Simulation and High-Level Control of Humanoid Robots*; IEEE RSJ International Conference on Intelligent Robots and Systems; 2000;
- [14].KANEHIRO, Fumio, FUJIWARA, Kiyoshi, KAJITA, Shuuji, YOKOI, Kazuhito, KANEKI, Kenji, HIRUKAWA, Hirohisa, NAKAMURA, Yoshihiko, YAMANE, Katsu; *Open Architecture Humanoid Robotics Platform*; IEEE International Conference on Robotics Automation; USA; Maio 2002.
- [15].Mathworks; *Matlab 7 – Creating Graphical User Interfaces; last revision*; March 2007;
- [16].Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro; *Curso de Matlab 5.1 – Introdução à Solução de Problemas de Engenharia*; 2ª Edição;
- [17].MARCHAND, Patrick, HOLLAND, O. Thomas; *Graphics and GUIs with Matlab*; Chapman & Hall/CRC; 3ª Edição; USA; 2003

Apêndice A - Modelo Virtual

Para ter o modelo dimensional completo é necessário abordar cada secção da estrutura individualmente. As dimensões fornecidas em seguida são uma aproximação muito pobre do robô humanóide, isto porque estes valores só serão úteis para o desenho do robô humanóide e animação de movimentos virtuais. Por esse motivo, para facilitar esse desenho e aliviar o processador de grandes cálculos estas aproximações cingem-se a figuras geométricas simples.

A.1 Secção do Pé

As dimensões do pé são acompanhadas por dois *offsets* que indicam a posição de ligação do pé com a junta. Estes valores estão descritos na seguinte tabela e seguem a ilustração da **Figura A-1**:

Dimensão	Comprimento (m)
Comprimento	0.200
Largura	0.080
Altura	0.020
X_offset	0.102
Y_offset	0.055

Tabela A-1 - Dimensões do pé

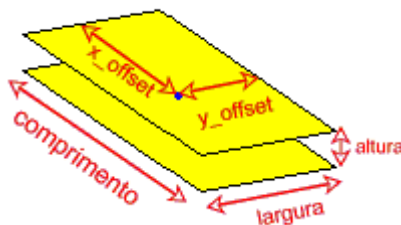


Figura A-1 - Definição do pé (pé direito – vista de trás)

A.2 Secção da Perna

Dimensão	Comprimento (m)
Comprimento	0.080
Largura	0.080
Altura	(dependente do comprimento dos elos)

Tabela A-2 - Dimensões da perna

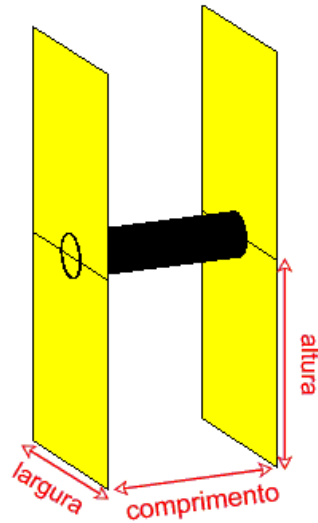


Figura A-2 - Definição da perna

A.3 Secção da Anca

Dimensão	Comprimento (m)
Comprimento	0.194
Largura	0.072
Altura	(dependente do comprimento dos elos)

Tabela A-3 - Dimensões da anca

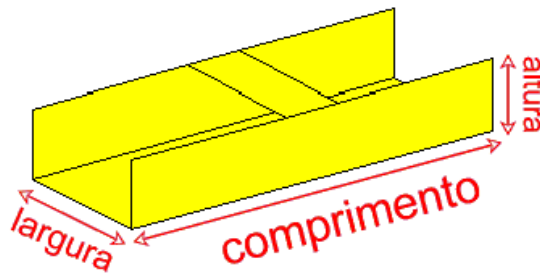


Figura A-3 - Definição da anca

A.4 Secção do Tronco

Dimensão	Comprimento (m)
Comprimento	0.194
Largura	0.072
Altura	(dependente do comprimento dos elos)

Tabela A-4 - Dimensões do tronco

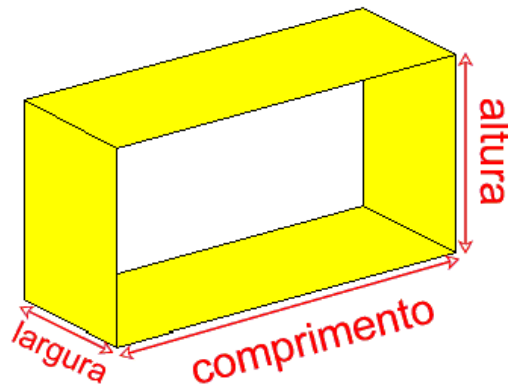


Figura A-4 - Definição do Tronco

A.5 Secção do Braço

Dimensão	Comprimento (m)
Comprimento (L1)	0.027
Largura (W1)	0.036
Comprimento (L2)	0.027
Largura (W2)	0.018
Altura	(dependente do comprimento dos elos)

Tabela A-5 - Dimensões do braço

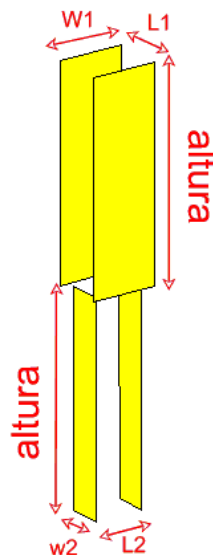


Figura A-5 - Definição do braço

Apêndice B - Protocolo de Comunicações

O PC comunica com o *master* através de uma comunicação série *RS232*. Este protocolo assenta numa arquitectura cliente/servidor. O *master* é uma base de dados global do robô humanóide, nesta base de dados constam os parâmetros de actuação e das leituras de todos os sensores existentes na estrutura. O protocolo de comunicação do *master* com as *slaves* permite manter esta base de dados actualizada com uma periodicidade de 8ms. ¹

Para a comunicação série usa-se a *toolbox* *cport_v1.3*. A comunicação é definida por um *handler* que serve para identificar a porta e as configurações especificadas para a comunicação. Entre os parâmetros de configuração de uma ligação *RS232*, estão os seguintes:

Parâmetro	Valor
BaudRate	115200
StopBits	1
ByteSize	8
DtrControl	Off
Parity	Off
RtsControl	Off

Para a gestão da comunicação série são disponibilizadas as funções *initcom* e *killcom*.

Initcom

Estabelecimento de uma nova ligação via RS-232

```
[handler, state]=initcom(gate, rate)
```

Entradas:

gate -> Porta a utilizar (1,2,...)

rate -> baudrate a definir (bits/s)

Saídas:

handler -> ID da linha de comunicações

state -> Configurações da linha

¹ Para uma descrição completa da comunicação master/slaves do humanóide ver [2]

Killcom

Término de uma ligação RS-232 existente.

```
stat=killcom(handler)
```

Entradas:

handler -> ID da linha série

Saídas:

stat -> retorna 0 em caso de sucesso

O *master* foi programado para responder a uma sequência de teste de forma a validar a comunicação *RS232*. A função *testcom* envia essa sequência estipulada e verifica a autenticidade da resposta. Esta função responde com 0 no caso de não haver erros na comunicação.

Testcom

Pedido de envio de uma sequência de teste.

```
[error,errorstr]=testcom(H)
```

Entradas:

handler -> ID da linha série

Saídas:

error -> Código de erro

errorstr -> String descritiva do erro

Existem dois tipos de sensores em cada *slave*, os sensores de posição e os sensores, designados especiais, devido à sua aplicação específica, tais como os sensores de força ou de inclinação. O sensor de posição além de permitir determinar a posição, por derivação permite determinar a velocidade e através de uma análise especial permite determinar a corrente consumida pelo servo.

Existem duas funções para leitura, *readjoint* e *readspecial*. A primeira permite a leitura de valores associados ao sensor de posição e recorre a parâmetros especificados na **Tabela B-1**. A segunda função permite a leitura do valor dos sensores especiais. As funções de leitura especificam sempre a *slave* que se pretende questionar.

Readjoint

Leitura de um parâmetro sensorial dos servos de um SCU

```
[servos, state, rx, error, errorstr]=readjoint (H, scu_id, param)
```

Entradas:

H => Handler para comunicar com o Master
scu_id => Identificador do SCU alvo
param => Parâmetro a ler (**Tabela B-1**)

Saídas:

servos => Parâmetro de saída [servo1, servo2, servo3]
state => Bits de estado dos servos (**Tabela B-2**)
rx => Mensagem de baixo nível recebida
error => Código de erro, se existente
errorstr => String descritiva do erro

Campo param		Descrição	Campo servos	Unidades
PARAM_POSITION	0	Posição angular de cada junta.	[pos ₁ , pos ₂ , pos ₃]	Graus
PARAM_VELOCITY	1	Velocidade estimada de cada junta.	[vel ₁ , vel ₂ , vel ₃]	Graus/100ms
PARAM_CURRENT	2	Corrente drenada por cada servo.	[curr ₁ , curr ₂ , curr ₃]	% do T de PWM

Tabela B-1 - Valores possíveis do parâmetro param da função readjoint

A variável *state* retornada pela função *readjoint* permite saber alguma informação adicional sobre os servos acoplados à *slave* questionada. A descrição de cada bit, do MSB até ao LSB, está especificada na seguinte tabela:

Elemento	Campo	Descrição
1	PWM	Activo se todos os motores possuem o PWM ligado.
2	Calib	Calibração dinâmica da posição dos servos activada.
3	Deadline	Ocorrência de um erro de violação da largura de banda disponível.
4	FinAll	Todos as juntas terminaram a trajectória.
5	FinOne	Pelo menos uma das juntas terminou a trajectória.
6	FinServo3	A junta 3 terminou a trajectória.
7	FinServo2	A junta 2 terminou a trajectória.
8	FinServo1	A junta 1 terminou a trajectória.

Tabela B-2 - Valores presentes no vector state retornado pela função readjoint

A escolha do sensor que se pretende ler está dependente do *scu_id* (*slave*) a que se está a aceder, pois uma *slave* só tem um tipo de sensor especial acoplado ao seu circuito de *piggy-back*.

Readspecial

Leitura dos sensores especiais (sensores de força, giroscópio ou inclinómetro).

```
[special,rx,error,errorstr]=readspecial(H,scu_id)
```

Entradas:

H => Handler das comunicações com o Master
scu_id => Identificador do SCU alvo

Saídas:

special => Valores dos sensores especiais
rx => Mensagem de baixo nível recebida
error => Código de erro, se existente
errorstr => String descritiva do erro

Para a actuação dos servomotores, há dois tipos de comando, o *applyjoint* que permite o envio de posições, velocidade ou simplesmente de activação do PWM nos servomotores e o *applycontrol* que permite designar o tipo de controlador activo e os parâmetros que definem esse controlador.

A função *applyjoint* tem os tipos de parâmetro de actuação descrito na **Tabela B-3**. Como característica peculiar está o parâmetro velocidade, como a aplicação de uma posição envolve o cálculo de uma trajectória de polinomial 5, como já foi dito anteriormente, logo a velocidade descreve uma polinomial de grau 4, por isso, o parâmetro velocidade define não a velocidade do movimento mas o tempo de duração da trajectória em incrementos de 20ms. Exemplificando, um movimento com duração de 2s deve especificar no parâmetro velocidade o valor 100

$$\left(\frac{2s}{20ms} = 100\right).$$

Applyjoint

Aplicação de uma ordem de actuação a cada componente de um SCU.

```
[rx,error,errorstr]=applyjoint(H,scu_id,param,servos)
```

Entradas:

H => Handler para comunicar com o Master
scu_id => Identificador do SCU alvo
param => Parâmetro a aplicar (**Tabela B-3**)
servos => Dados a aplicar [servo1,servo2,servo3]

Saídas:

rx => Mensagem de baixo nível recebida
error => Código de erro, se existente
errorstr => String descritiva do erro

<i>Campo param</i>		<i>Descrição</i>	<i>Campo servos</i>	<i>Unidades</i>
<i>PARAM_POSITION</i>	0	Posição referência a ser atingida.	[<i>ref</i> ₁ , <i>ref</i> ₂ , <i>ref</i> ₃]	Graus ou outro
<i>PARAM_VELOCITY</i>	1	Duração do movimento a efectuar.	[<i>vel</i> ₁ , <i>vel</i> ₂ , <i>vel</i> ₃]	Ciclos de 20ms
<i>PARAM_SPECIAL</i>	3	Activação/desactivação do PWM aplicado aos motores.	[0/1, 0, 0]	(Valor booleano)

Tabela B-3 - Valores possíveis do parâmetro param na função *applyjoint*

O controlo do servomotor é composto por dois níveis, o primeiro é o controlador de posição, que pode ser em malha aberta ou um controlador PI, **Figura B-1**. O segundo nível de controlo refere-se ao controlador com realimentação proveniente dos sensores especiais e cujo algoritmo de controlo é específico de determinado sensor. O firmware existente tem definido dois algoritmos possíveis de controlo de segundo nível e contempla a existência de um terceiro. Os parâmetros de controlo estão descritos na **Tabela B-4** e os tipos de controlador de segundo nível na **Tabela B-5**.

Applycontrol

Seleção do controlador de primeiro nível e ajuste dos seus parâmetros bem como dos parâmetros do controlador local

```
[rx,error,errorstr]=applycontrol(H,scu_id,param,servos)
```

Entradas:

H => Handler para comunicar com o Master

scu_id => Identificador do SCU alvo

param => Parâmetro a modificar (**Tabela B-4**)

servos => Dados a aplicar [*servo*₁, *servo*₂, *servo*₃]

Saídas:

rx => Mensagem de baixo nível recebida

error => Código de erro, se existente

errorstr => String descritiva do erro

<i>Campo param</i>		<i>Descrição</i>	<i>Campo servos</i>
<i>PARAM_KI</i>	0	Ganho da componente integral (<i>K_i</i>) do controlador local.	[<i>K_i</i> ₁ , <i>K_i</i> ₂ , <i>K_i</i> ₃]
<i>PARAM_KP</i>	1	Ganho da componente proporcional (<i>K_p</i>) do controlo local.	[<i>K_p</i> ₁ , <i>K_p</i> ₂ , <i>K_p</i> ₃]
<i>PARAM_K</i>	2	Ganho (<i>K</i>) do controlador de primeiro nível.	[<i>K</i> ₁ , <i>K</i> ₂ , <i>K</i> ₃]
<i>PARAM_CONTROLON</i>	3	Tipo de controlo de primeiro nível (<i>Type</i>) a aplicar em cada junta (Tabela B-5).	[<i>Type</i> ₁ , <i>Type</i> ₂ , <i>Type</i> ₃]

Tabela B-4 - Valores possíveis do parâmetro param na função *applycontrol*

Tipo de Controlo (Type)		Descrição
NO_CONTROL	0	Sem Controlo de primeiro nível
COP_CONTROL	1	Controlo de Centro de Pressão
INC_CONTROL	2	Controlo de Inclinação
GIRO_CONTROL	3	Controlo de velocidade angular

Tabela B-5 - Tipo de controladores de segundo nível

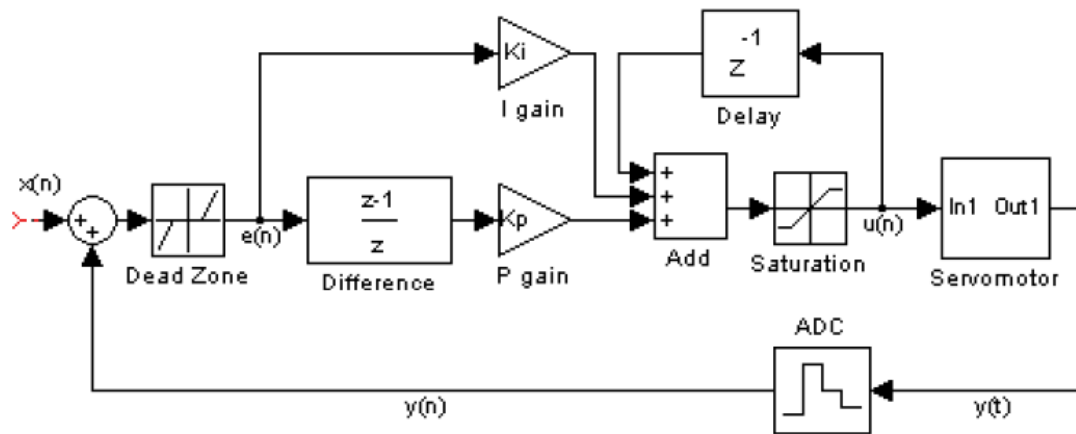


Figura B-1 - Controlador PI implementado nas slaves

Apêndice C - Representação de Denavit-Hartenberg

Considera-se, para uma cadeia cinemática de $n+1$ elementos conectados por n juntas. A cada membro da cadeia é associado um sistema de coordenadas Cartesianas. Assim, o sistema O_0, x_0, y_0, z_0 é associado à base 0 da cadeia, o sistema O_1, x_1, y_1, z_1 , ao membro 1 da cadeia e assim por diante.

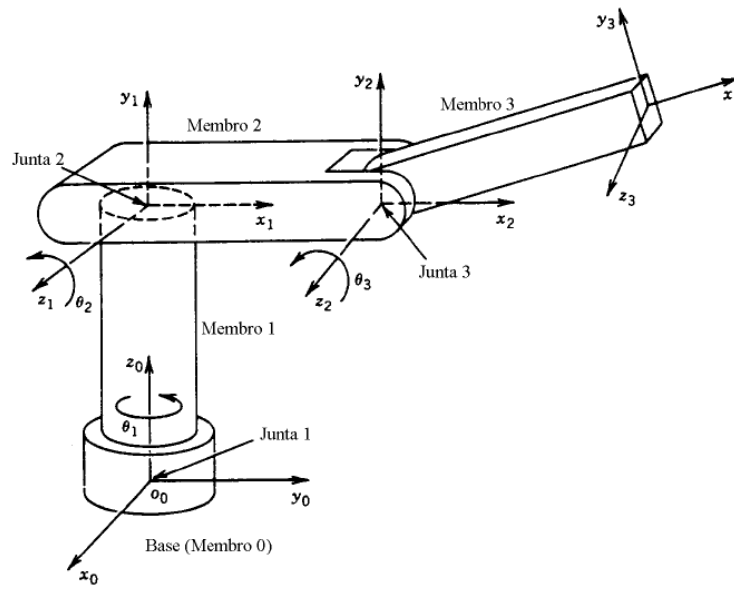


Figura C-1 - Convenção para membros, juntas e sistema de eixos

Seja A_{i-1}^i a matriz de transformação do sistema do membro i para o sistema do membro $i-1$. Como cada junta possui apenas 1 grau de liberdade, a matriz de transformação depende apenas da variável da junta q_i .

$$A_{i-1}^i = R_{z,\theta} \cdot T_{z,d} \cdot T_{x,a} \cdot R_{x,\alpha} \quad (C-1)$$

Onde:

$R_{z,\theta}$ representa a rotação θ em torno do eixo z (sinal positivo dado pela regra da mão direita);

$T_{z,d}$ representa a translação d ao longo do eixo z (sinal positivo quando a translação concorda com o sentido do eixo);

$T_{x,a}$ representa a translação a ao longo do eixo x (sinal positivo quando a translação concorda com o sentido do eixo);

$R_{x,\alpha}$ representa a rotação α em torno do eixo x (sinal positivo dado pela regra da mão direita).

Os parâmetros θ , d , a e α são o ângulo de rotação, a excentricidade, o comprimento e a torção, respectivamente e são designados por parâmetros de DH.

Desenvolvendo (C-1), fica:

$$A_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 & 0 \\ \text{sen}(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\text{sen}(\alpha_i) & 0 \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C-2})$$

Resolvendo fica,

$$A_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) \cdot \cos(\alpha_i) & \text{sen}(\theta_i) \cdot \text{sen}(\alpha_i) & a_i \cdot \cos(\theta_i) \\ \text{sen}(\theta_i) & \cos(\theta_i) \cdot \cos(\alpha_i) & -\cos(\theta_i) \cdot \text{sen}(\alpha_i) & a_i \cdot \text{sen}(\theta_i) \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C-3})$$

Sendo que,

a_i distancia, ao longo de x_i , de O_i à intersecção dos eixos x_i e z_{i-1} (ou a distancia mais curta entre os eixos z_{i-1} e z_i);

d_i distancia, ao longo de z_{i-1} de O_{i-1} à intersecção dos eixos x_i e z_{i-1} ;

α_i ângulo do eixo z_{i-1} para o eixo z_i , medido em torno de x_i (sinal dado pela regra da mão direita);

θ_i ângulo do eixo x_{i-1} para o eixo x_i , medido em torno de z_{i-1} (sinal dado pela regra da mão direita).

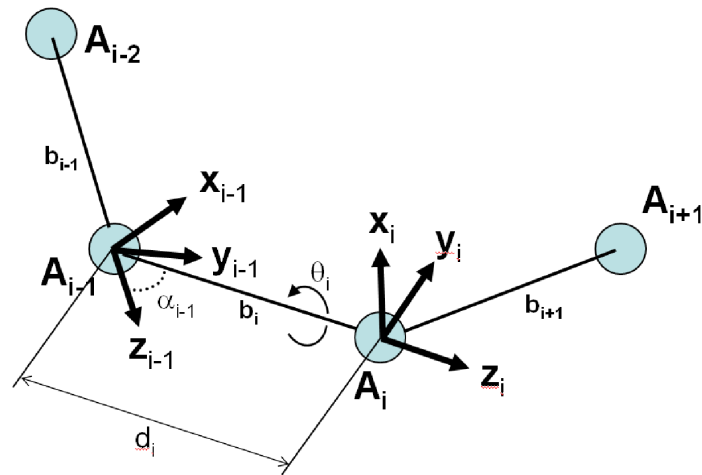


Figura C-2 - Representação de Denavit-Hartenberg

A representação de DH impõe duas condições:

- O eixo z_{i-1} é o eixo da junta i ;
- O eixo x_i é perpendicular ao eixo z_{i-1} , apontando no sentido do afastamento desse último e intercepta o eixo z_{i-1}

Com estas variáveis definidas, forma-se a matriz homogênea para cada um dos elementos da cadeia de elos e, depois de definidas todas as matrizes de transformação cria-se a matriz de transformação directa, que é dada da seguinte forma,

$$H_0^n = \begin{bmatrix} \cos(x_n, x_0) & \cos(y_n, x_0) & \cos(z_n, x_0) & x_0 \\ \cos(x_n, y_0) & \cos(y_n, y_0) & \cos(z_n, y_0) & y_0 \\ \cos(x_n, z_0) & \cos(y_n, z_0) & \cos(z_n, z_0) & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C-4})$$

Esta matriz representa a transformação entre o sistema de coordenadas do elemento n e o sistema de coordenadas de referencia e é obtido através de,

$$H_0^n = A_0^1 \cdot A_1^2 \cdot \dots \cdot A_{n-1}^n \quad (\text{C-5})$$

Apêndice D - Relação entre os Parâmetros de Locomoção

Quando se pretende definir os parâmetros de entrada dos padrões de locomoção surge a questão de quais é que são os limites para cada parâmetro? Esta pergunta é de extrema importância porque o robô humanóide é uma cadeia cinemática complexa com vários tipos de limitações e por isso não é qualquer valor que o calculador cinemático aceita como válido.

Há uma forma de responder a essa questão mas parte do princípio de que o operador do algoritmo de padrão de locomoção tem em mãos uma sequência de valores *standard* válidos para o movimento que pretende. Estabelece-se então uma relação entre os limites que cada parâmetro pode ter se se mantiverem todos os outros inalterados.

Esta secção aborda o problema da relação entre parâmetros focando-se exclusivamente na configuração de uma perna. Considere-se, para o estudo realizado nesta secção, as dimensões da perna dadas pela **Figura D-1** e **Figura D-2**.

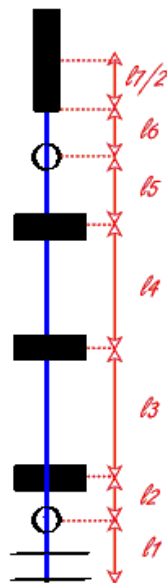


Figura D-1 - Perna (vista de frente)

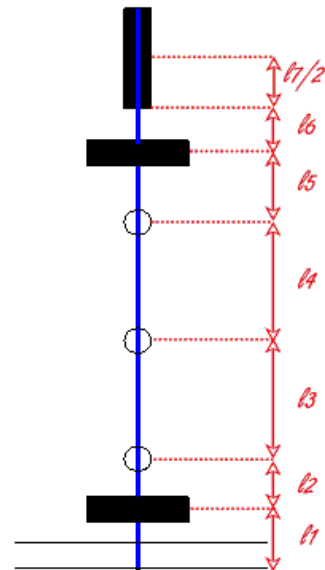


Figura D-2 - Perna (vista de lado)

D.1 Rodar e Chutar

Para os movimentos de rodar e chutar o trabalho realizado pelas juntas sagitais não envolve qualquer deslocação em xx , como representado na **Figura D-3**, para qualquer uma das pernas.

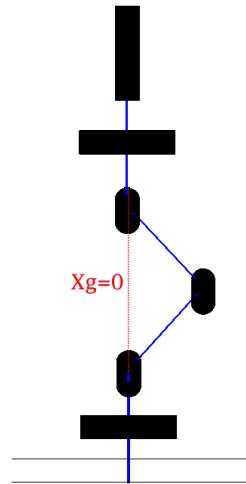


Figura D-3 - Juntas sagitais alinhadas segundo xx

Portanto, tendo em conta a **Figura D-4**, podem-se derivar as seguintes equações:

$$Hh' = Hh - (l_1 + l_6 + \frac{l_7}{2}) \quad (\text{D-1})$$

$$h_{\max} = l_2 + l_3 + l_4 + l_5 \quad (\text{D-2})$$

$$h_{\min} = l_2 + (l_3 + l_4) \cdot \cos(\theta_{S \max}) + l_5 \quad (\text{D-3})$$

Onde:

$$\theta_{S \max} = \min\left(\text{abs}\left(\theta_{2 \max}, \frac{\theta_{3 \min}}{2}, \theta_{4 \max}\right)\right) \quad (\text{D-4})$$

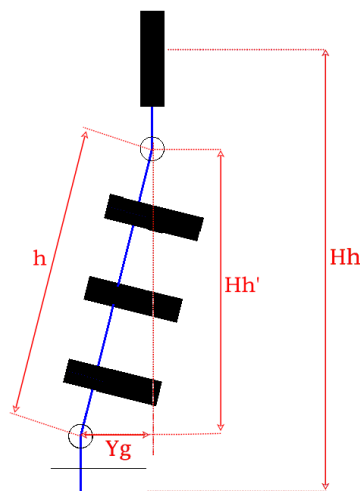


Figura D-4 - Parâmetros Hh' e h

Repare-se que nestes cálculos está-se sempre a assumir que não há movimento da anca em xx , e que todos os critérios de postura do tronco e tornozelo estão a ser cumpridos. É também assumido que θ_F é o ângulo da junta de acção frontal, ou seja das juntas 1 e 5, onde

$$\theta_1 = \theta_F = -\theta_5 \quad (\text{D-5})$$

Para o caso das juntas do plano sagital 2, 3 e 4, definem-se como θ_S e devido às condições já impostas e aos comprimentos dos elos l_3, l_4 , pode-se afirmar que:

$$\theta_2 = \theta_S = -2 \cdot \theta_3 = \theta_4 \quad (\text{D-6})$$

Derivando as dependências entre os parâmetros Yg, Hh, Fc , fica o seguinte:

Para os limites de Yg ,

máximo

$$\text{Sabendo que } \theta_{F \max} = \min(\text{abs}(\theta_{1 \max}, \theta_{5 \min}))$$

$$\begin{cases} Yg_{\max} = \sqrt{h_{\max}^2 - Hh'^2} \\ \theta_F = a \cos\left(\frac{Hh'}{h_{\max}}\right) \end{cases} \quad (\text{D-7})$$

Se $\theta_F > \theta_{F \max}$,

$$\begin{cases} Yg_{\max} = \frac{Hh}{\tan(\theta_{F \max})} \\ \theta_F = \theta_{F \max} \end{cases} \quad (\text{D-8})$$

mínimo

Se $h_{\min} < Hh'$ então,

$$\begin{cases} Yg_{\min} = 0 \\ \theta_F = 0 \end{cases} \quad (\text{D-9})$$

senão,

$$\begin{cases} Yg_{\min} = \sqrt{h_{\min}^2 - Hh'^2} \\ \theta_F = a \cos\left(\frac{Hh'}{h_{\min}}\right) \end{cases} \quad (\text{D-10})$$

Para definir os limites de Hh

máximo

$$Hh'_{\max} = \sqrt{h_{\max}^2 - Yg^2} \quad (\text{D-11})$$

$$Hh_{\max} = Hh'_{\max} + (l_1 + l_6 + l_7/2) \quad (\text{D-12})$$

mínimo

$$Hh'_{\min} = \sqrt{h_{\min}^2 - Yg^2} \quad (\text{D-13})$$

$$Hh_{\min} = Hh'_{\min} + (l_1 + l_6 + l_7/2) + Fc \quad (\text{D-14})$$

Para definir os limites de Fc

máximo

$$\theta_F = a \tan\left(\frac{Yg}{Hh'}\right) \quad (\text{D-15})$$

$$h = \sqrt{Yg^2 + Hh'^2} \quad (\text{D-16})$$

$$Fc_{\max} = (h - h_{\min}) \cdot \cos(\theta_F) \quad (\text{D-17})$$

mínimo

$$Fc_{\min} = 0 \quad (\text{D-18})$$

D.2 Andar

O movimento de andar introduz um parâmetro em xx que é o Sl . Dessa forma os cálculos de h_{\max} e h_{\min} já não são válidos para este caso mas todas as outras definições e limites obedecem ao mesmo cálculo. Portanto, é necessário encontrar uma função que calcule o h_{\min} e h_{\max} tendo em conta o novo parâmetro. Na **Figura D-5** ilustra-se o problema, Xc é a nova variável introduzida.

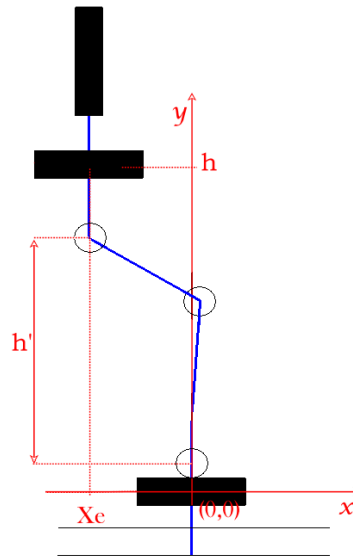


Figura D-5 - Limitações - introdução de Sl (Xc)

O problema de cinemática inversa apresentado conta com um Xc fixo e com as limitações das juntas, portanto

$$h' = h - (l_2 + l_3) \quad (\text{D-19})$$

Para os casos de rodar e chutar, considerou-se que não havia deslocamento segundo o eixo das abcissas, nas **Figura D-6** e **Figura D-7** está representado o limite máximo e mínimo do h' para essa situação. O limite máximo de h' ocorre quando ambos os elos estão alinhados com o eixo dos yy , segundo o eixo dos xx , logo para θ_2 e θ_3 nulos. O limite mínimo de h' sucede pela inclinação máxima que a junta 2, 3 e 4 podem ter garantindo sempre duas condições:

1. Qualquer deslocamento em xx provocado pela junta 2, têm de ser compensado pela junta 3, logo $\theta_3 = -2\theta_2$ (considerando $l_3=l_4$);
2. O tronco tem que estar vertical, logo $\theta_4 = \theta_2$.

Dessa forma, o $\theta_{S \max}$, como já foi referido na secção anterior é dado por

$$\theta_{S \max} = \min \left(\text{abs} \left(\theta_{2 \max}, \frac{\theta_{3 \min}}{2}, \theta_{4 \max} \right) \right) \quad (\text{D-20})$$

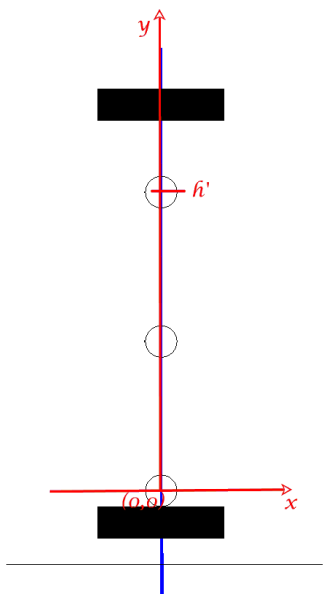


Figura D-6 - h' máximo no caso de rodar e chutar

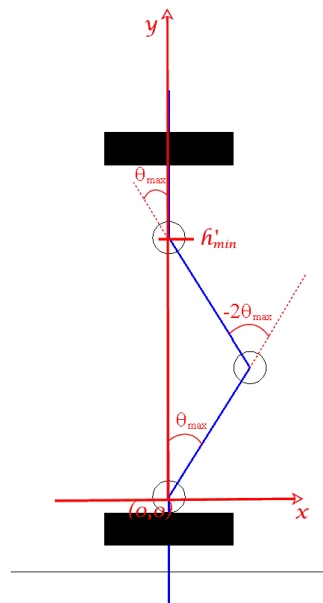


Figura D-7 - h' mínimo no caso de rodar e chutar

Devido à não linearidade das equações de cinemática inversa 2R, vai ter de se encontrar outras formas de calcular o limite máximo e mínimo de h' quando há liberdade de movimentos segundo o eixo dos xx . Interessa portanto encontrar todas as ordenadas possíveis para uma abscissa fixa.

Definiu-se no modelo humanóide que a junta do joelho, tal como no ser humano, só se desloca num sentido, logo é válido que se limite a junta entre $[\theta_{3min} 0]$. Segundo esta condição e, elaborando um mapa de todos os pontos atingíveis pelo elo final da cadeia cinemática 2R (**Figura D-8**), podem-se definir duas fases para o limite superior e outras duas para o limite inferior.

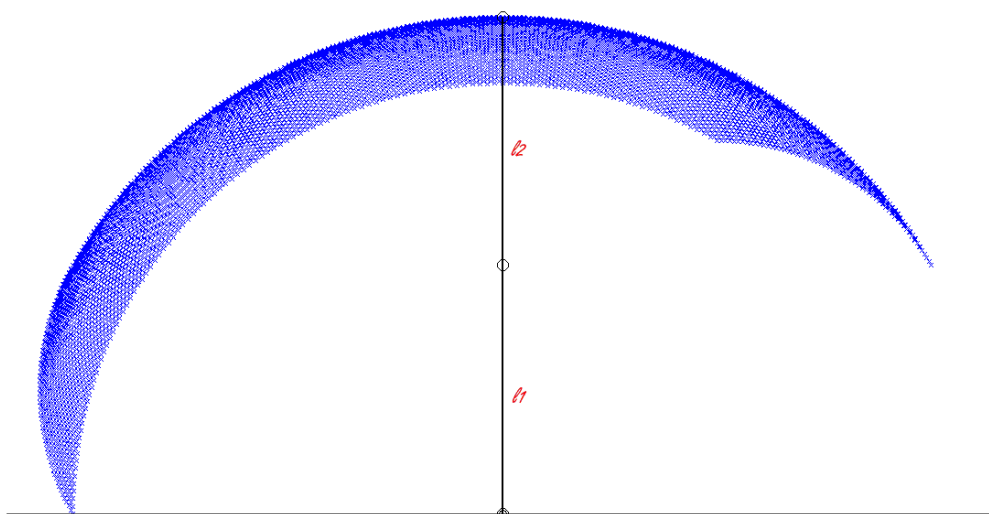


Figura D-8 - Nuvem de pontos possíveis para uma cadeia 2R limitada

Como se está a operar exclusivamente no primeiro e segundo quadrante do círculo unitário, o limite superior é garantido quando ambas as juntas estão alinhadas, ou seja quando θ_3 é nulo, **Figura D-9**.

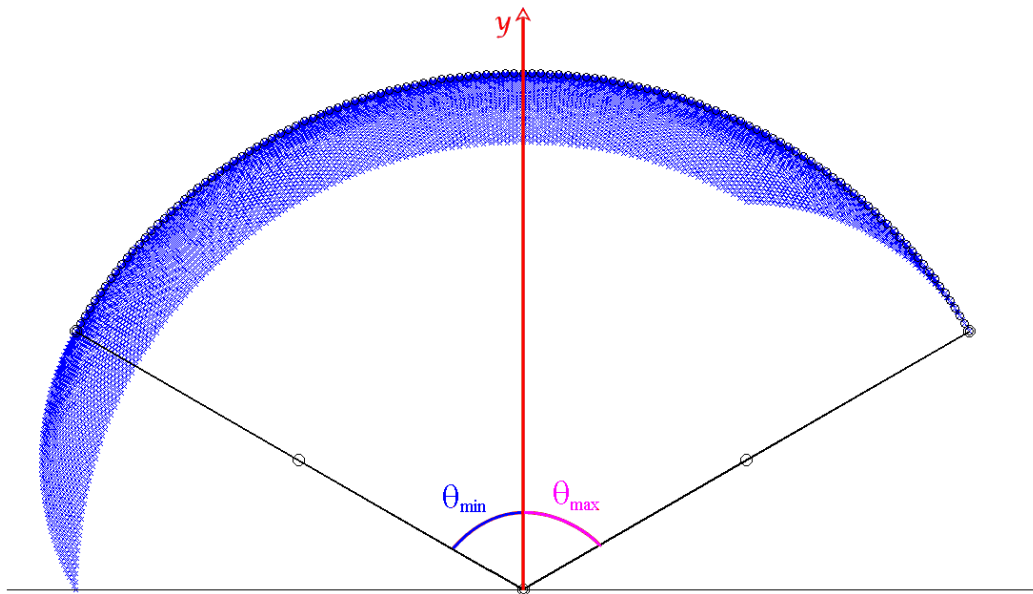


Figura D-9 - Limite superior dado (1)

O limite inferior, é obtido quando a distância entre os extremos da cadeia é mínimo, ou seja quando θ_3 é mínimo, **Figura D-10**.

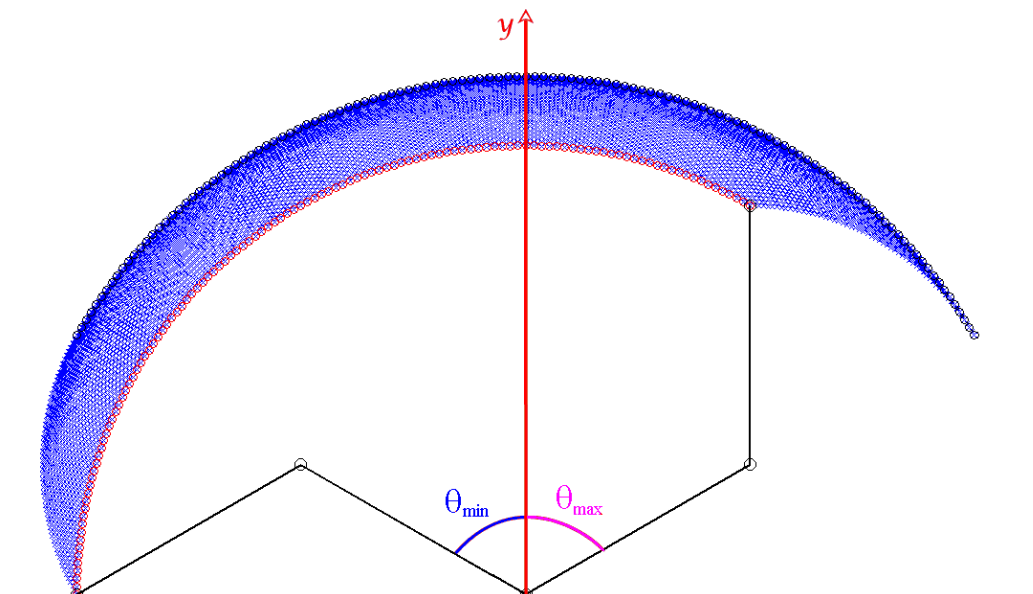


Figura D-10 - Limite inferior dado (1)

Por último há uma situação especial, quando θ_2 atinge o seu limite superior, a faixa de valores que pode ser varrida variando o θ_3 estabelece a segunda fase do limite inferior. Por outro lado, quando

θ_2 atinge o limite mínimo, a faixa de valores que pode ser varrida variando o θ_3 estabelece a segunda fase do limite superior, **Figura D-11**.

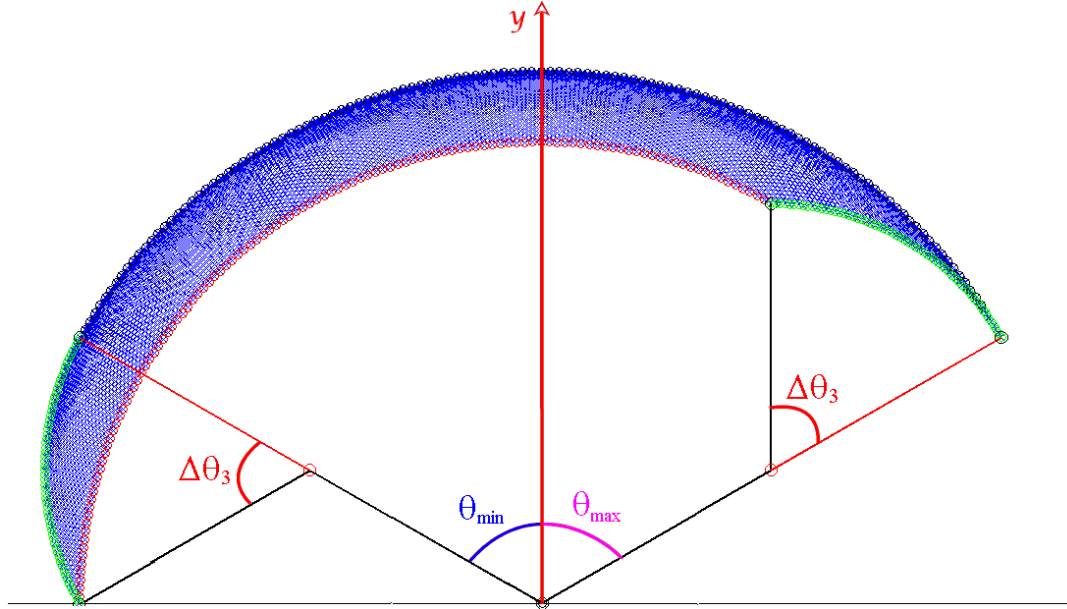


Figura D-11 - Limite inferior e superior dado (2)

Resta então elaborar as equações que definem os limites fronteiros dos pontos possíveis para a cadeia cinemática 2R.

$$\begin{cases} y = \sqrt{(l_1 + l_2)^2 - x^2} & , \text{para } x > (l_1 + l_2) \cdot \sin(\theta_{\min}) \\ y = \sqrt{(l_1 + l_2 \cdot \cos(\theta_{3\min}))^2 - x^2} & , \text{para } x < (l_1 + l_2 \cdot \cos(\theta_{3\min})) \cdot \sin(\theta_{\max}) \\ y = l_1 \cdot \cos(\theta_{\min}) + \sqrt{(l_2)^2 - (x - l_1 \cdot \sin(\theta_{\min}))^2} & , \text{para } x < \max[l_2 \cdot \sin(\theta_{\min} + \theta_{3\min}), l_2 \cdot \sin(\theta_{\min})] - l_1 \cdot \sin(\theta_{\min}) \\ y = l_1 \cdot \cos(\theta_{\max}) + \sqrt{(l_2)^2 - (x - l_1 \cdot \sin(\theta_{\max}))^2} & \text{para } x > \min[l_2 \cdot \sin(\theta_{\max} + \theta_{3\min}), l_2 \cdot \sin(\theta_{\max})] + l_1 \cdot \sin(\theta_{\max}) \end{cases}$$

Considerando a função contínua, pode ser reescrita da seguinte forma:

$$\begin{cases} y_{\text{sup}} = \sqrt{(l_1 + l_2)^2 - x^2} & , \text{para } x \geq (l_1 + l_2) \cdot \sin(\theta_{\min}) \\ y_{\text{sup}} = l_1 \cdot \cos(\theta_{\min}) + \sqrt{(l_2)^2 - (x - l_1 \cdot \sin(\theta_{\min}))^2} & , \text{para } x < (l_1 + l_2) \cdot \sin(\theta_{\min}) \\ y_{\text{inf}} = \sqrt{(l_1 + l_2 \cdot \cos(\theta_{3\min}))^2 - x^2} & , \text{para } x \leq (l_1 + l_2 \cdot \cos(\theta_{3\min})) \cdot \sin(\theta_{\max}) \\ y_{\text{inf}} = l_1 \cdot \cos(\theta_{\max}) + \sqrt{(l_2)^2 - (x - l_1 \cdot \sin(\theta_{\max}))^2} & , \text{para } x > (l_1 + l_2 \cdot \cos(\theta_{3\min})) \cdot \sin(\theta_{\max}) \end{cases} \quad (\text{D-21})$$

Resolvendo as equações anteriores para determinado x , é possível obter os valores fronteiros de y e, correspondendo o y ao h , é possível assim determinar o h_{\max} e o h_{\min} :

$$\begin{aligned} h_{\min} &= \min(y_{\sup}, y_{\inf}) \\ h_{\max} &= \max(y_{\sup}, y_{\inf}) \end{aligned} \quad (\text{D-22})$$