

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Desenvolvimento de um Controlador Dinâmico para Robôs Humanoides

NAO

Rui Edgar Vieira Monteiro

Mestrado de Engenharia Eletrotécnica e Computadores

Orientador: António Manuel Correia Pereira (PhD)

Co-orientador: Luís Paulo Reis (PhD)

Julho de 2012

Resumo

Nos últimos anos vários investigadores têm-se focado no desenvolvimento e controlo de robôs humanoides capazes de adquirir comportamentos e características humanas com o objetivo de permitir a interação com os humanos e trabalhar em cooperação com os mesmos ou até mesmo substituí-los em tarefas árduas, repetitivas ou perigosas. A queda de um robô pode provocar não só danos ao mesmo, o que implicará custos em reparação, como também riscos à saúde humana pelo que a garantia de estabilidade é uma questão inevitável em termos de custos e segurança.

Este documento apresenta a implementação de um controlador dinâmico para robôs, capaz de posicionar o centro de massa do robô numa determinada posição pretendida. O controlador pode funcionar como um módulo de estabilidade, caso o objetivo seja o posicionamento do centro de massa no centroide do polígono de suporte do robô, ou como método de avaliação de equilíbrio estático. Neste sentido serão apresentados os cálculos da cinemática direta e inversa do modelo do robô humanoide utilizado e foi desenvolvido um comportamento preciso de preparação de um remate através da colocação do pé de suporte do robô numa posição e orientação calculados de acordo com a posição do robô, da bola e o destino da bola, e através da transferência do centro de massa do robô para essa posição, constituindo o remate mais forte na liga de simulação 3D da RoboCup até à data.

Ao contrário dos remates dinâmicos e omnidirecionais que têm sido desenvolvidos nos últimos anos, este novo remate consiste num movimento constante resultante de um elevado processo de otimização, o qual necessita de um posicionamento preciso do pé de suporte do robô. Através da colocação dinâmica do pé de suporte é possível aliar a potência do chute desenvolvido à robustez e precisão do movimento de preparação de remate desenvolvido, garantindo uma capacidade de direcionamento do remate e uma maior abrangência na área admissível de execução do mesmo.

Abstract

In the last years several investigators have focused on the development and control of humanoid robots able to acquire human behaviours and features in order to be able to interact with humans and work together with them or even replace them in their arduous, repetitive or dangerous tasks. The fall of a robot can lead to damages, which entail costs for repair, or even human health risks so ensuring stability is a main concern in terms of cost and safety.

This document presents the implementation of a dynamic controller for robots, able to position the robot's center of mass in a certain location. The controller may function as a stability module, if the objective is to position the center of mass at the centroid of the robot's support polygon or work as a method of evaluation of static balance.

In this work both forward and inverse kinematics of the humanoid robot model were presented and it was also developed a precise behaviour of preparing a kick by placing the support foot of the robot in a precise position and orientation calculated in accordance with the robot's position, the ball and the target, and by transferring the robot's center of mass to the desired position. This is the greatest kick in the RoboCup's 3D Simulation League so far.

Unlike omnidirectional and dynamic kicks that have been recently developed, this new kick consists in a unique movement which results from a high optimization process and requires a precise positioning of the robot's support foot. By placing the support foot dynamically, is possible to combine the developed kick power with the robustness and accuracy provided by the prepare kick movement developed, ensuring a targeting capability and a larger coverage area of a possible kick.

“We should not let our fears hold us back from pursuing our hopes.”

John F. Kennedy

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo	2
1.3	Estrutura do Relatório	2
2	RoboCup	5
2.1	RoboCupJunior	5
2.2	RoboCupRescue	6
2.3	RoboCup@Home	7
2.4	RoboCupSoccer	7
2.4.1	Small Size Robot League (SSL)	7
2.4.2	Middle Size Robot League (MSL)	8
2.4.3	Humanoid League	9
2.4.4	Standard Platform League (SPL)	9
2.4.5	Simulation League	10
2.5	FC Portugal	11
2.5.1	FC Portugal	11
2.5.2	CAMBADA	12
2.5.3	5DPO	12
2.5.4	Portuguese Team	12
2.6	Conclusão	13
3	Robôs Humanoides	15
3.1	Introdução	15
3.2	História dos robôs	15
3.3	Projetos associados à robótica humanoide	19
3.3.1	Universidade de Waseda	19
3.3.2	Universidade de Tóquio	20
3.3.3	Boston Dynamics	20
3.3.4	Honda	21
3.3.5	HRP / AIST	21
3.3.6	Toyota	22
3.3.7	Aldebaran Robotics	22
3.4	Aldebaran NAO	23
3.5	Conclusão	24

4	Ambientes de simulação e agente FC Portugal	25
4.1	Introdução	25
4.2	Vantagens da simulação	25
4.3	SimSpark	26
4.3.1	Servidor	26
4.3.2	Monitor	28
4.3.3	Agente	29
4.4	Alternativas ao simulador	30
4.5	Ambiente e objetos do campo	31
4.5.1	Dimensões	31
4.5.2	Regras	31
4.6	Agente FC Portugal 3D	32
4.6.1	Arquitetura	32
4.6.2	Modelo	33
4.7	Conclusão	34
5	Análise de estabilidade	35
5.1	Configuração de locomoção de robôs	36
5.2	Polígono de suporte	37
5.3	Centro de Massa (CoM)	37
5.4	Centro de Pressão (CoP)	38
5.5	Zero Moment Point (ZMP)	39
5.6	Locomoção estática e dinâmica	39
5.7	Conclusão	40
6	Análise cinemática	41
6.1	Movimentos rígidos e transformações homogéneas	42
6.1.1	Representação de posições	42
6.1.2	Representação de rotações	43
6.1.3	Transformações homogéneas	44
6.2	Cinemática direta	44
6.3	Cinemática inversa	47
6.3.1	Braços	48
6.3.2	Pernas	49
6.4	Singularidades	52
6.5	Conclusão	54
7	Controlador do Centro de Massa	55
7.1	Controlo de malha aberta	55
7.1.1	Vantagens	56
7.1.2	Desvantagens	56
7.2	Controlo de malha fechada	56
7.2.1	Vantagens	58
7.2.2	Desvantagens	58
7.2.3	Controlo PID	58
7.3	O controlador	60
7.3.1	Braços	60
7.3.2	Pernas	61
7.4	Polígono de suporte	66

7.5	Centroide	68
7.6	RoboViz	69
7.6.1	O protocolo de desenho	70
7.6.2	Funções implementadas	71
7.7	Conclusão	71
8	Preparação de remate	73
8.1	Análise do mundo	74
8.2	Planeamento do movimento	75
8.2.1	Planeamento de trajetória	75
8.2.2	Transferência para suporte único	77
8.2.3	Colocação do pé de suporte	78
8.2.4	Abaixamento do corpo do robô	78
8.2.5	Transferência do centro de massa do robô	79
8.2.6	Colocação em pose de remate	80
8.3	Testes e resultados	80
8.4	Conclusão	85
9	Conclusões e trabalho futuro	87
9.1	Conclusões	87
9.2	Trabalhos futuros	88

Lista de Figuras

2.1	Ligas RoboCupJunior	6
2.2	RoboCupRescue Simulation league	6
2.3	RoboCup@Home league	7
2.4	Small Size Robot League	8
2.5	Middle Size Robot League	8
2.6	Humanoid League	9
2.7	Standard Platform League	10
2.8	Simulation League	10
3.1	Modelo baseado no robô de Leonardo Da Vinci	17
3.2	Os automata de Jacques Vaucanson	17
3.3	Steam Man e Electric Man	18
3.4	O robô humanoide Elektro	18
3.5	Waseda Wabian 2-R	20
3.6	Robô PetMan da Boston Dynamics	20
3.7	Evolução dos robôs humanoides Honda	21
3.8	Robô HRP-4	22
3.9	Toyota Motor Corporation partner robot	22
3.10	Aldebaran NAO	23
4.1	Diagrama de comunicação entre o servidor e o agente	27
4.2	Arquitetura do servidor SimSpark	28
4.3	Monitor genérico do SimSpark	28
4.4	NAO real vs modelo simulado	29
4.5	Disposição e orientação das juntas do NAO	30
4.6	Dimensões do campo simulado	31
4.7	Arquitetura do agente da equipa FC Portugal 3D	33
5.1	Planos anatómicos de um humanoide	36
5.2	Configuração bípede, quadrúpede e hexápode	36
5.3	Derivação do polígono de suporte no plano transversal	37
5.4	Projeção do centro de massa e polígono de suporte	38
5.5	Cálculo do centro de pressão	38
5.6	Equilíbrio estático e dinâmico	39
6.1	Junta rotativa e junta prismática	41
6.2	Representação de um ponto p em dois referenciais distintos	42
6.3	Rotação ao longo do eixo y	43
6.4	Atribuição de eixos segundo a convenção DH	47

6.5	Braço do NAO e esquemático das suas articulações	48
6.6	Dedução da cinemática inversa	49
6.7	Articulações das pernas do NAO	50
6.8	Referenciais utilizados no cálculo da cinemática inversa das pernas do NAO	50
7.1	Arquitetura típica de controlador em malha aberta	55
7.2	Arquitetura típica de controlador em malha fechada	57
7.3	Controlador PID	58
7.4	Resposta de controlador de juntas PID	59
7.5	Projeção do braço do robô segundo o eixo y	60
7.6	Resposta do CoM do robô	61
7.7	Fases da locomoção	61
7.8	Controlo do centro de massa do robô através das juntas pitch e roll da anca e dos pés em fase de duplo suporte	62
7.9	Posição do centro de massa do robô controlado em fase de duplo suporte	63
7.10	Controlo do centro de massa do robô através das juntas pitch e roll da anca e dos pés em fase de suporte único	63
7.11	Posição do centro de massa do robô controlado através das juntas da bacia e do pé em fase de suporte único	64
7.12	Sequência de controlo do centro de massa do robô controlado através das juntas da bacia e do pé em fase de suporte único	64
7.13	Controlo do centro de massa do robô através das juntas pitch e roll da anca em fase de suporte único	65
7.14	Posição do centro de massa do robô controlado através das juntas da bacia em fase de suporte único	65
7.15	Sequência de controlo do centro de massa do robô controlado através das juntas da bacia em fase de suporte único	66
7.16	Comparação da resposta dos controladores	66
7.17	Cálculo do polígono convexo	67
7.18	Atualização do polígono convexo	68
7.19	Polígono de 6 vértices	69
7.20	Comparação entre o RoboViz e o monitor genérico do SimSpark	70
7.21	Demonstração da funcionalidade de desenho do RoboViz	70
8.1	Cálculo da posição do pé de suporte do robô	75
8.2	Curvas de Bézier de 1º, 2º, 3º e 4º grau	76
8.3	Duas curvas cúbicas de Bézier com diferentes pontos de controlo.	77
8.4	Sequência de transferência do robô para suporte único	77
8.5	Volume de trabalho da perna do robô	78
8.6	Abaixamento do corpo do robô	78
8.7	Sequência de abaixamento e colocação do pé de suporte	79
8.8	Transferência do centro de massa do robô	79
8.9	Sequência de transferência do centro de massa do robô	79
8.10	Pose final após a transferência do centro de massa do robô e pose de remate	80
8.11	Sequência de colocação em pose de remate e conseqüente remate	80
8.12	Representação da distância do remate em relação à posição do robô orientado com o alvo	81
8.13	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 0º entre a sua orientação e o alvo	82

8.14	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 10° entre a sua orientação e o alvo	83
8.15	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 20° entre a sua orientação e o alvo	83
8.16	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 30° entre a sua orientação e o alvo	84
8.17	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 40° entre a sua orientação e o alvo	84
8.18	Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 50° entre a sua orientação e o alvo	85

Lista de Tabelas

3.1	Relação entre massa, altura, IMC e preço dos robôs humanoides funcionais . . .	19
6.1	Parâmetros associados à cinemática direta da cabeça segundo a convenção DH . .	46
6.2	Parâmetros associados à cinemática direta do braço esquerdo segundo a convenção DH	46
6.3	Parâmetros associados à cinemática direta do braço direito segundo a convenção DH	46
6.4	Parâmetros associados à cinemática direta da perna esquerda segundo a convenção DH	46
6.5	Parâmetros associados à cinemática direta da perna direita segundo a convenção DH	46

Abreviaturas e Símbolos

AIST	Advanced Industrial Science and Technology
API	Application Programming Interface
CoM	Center of Mass
CoP	Center of Pressure
CSV	Comma Separated Values
DARPA	Defense Advanced Research Projects Agency
DH	Denavit Hartenberg
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFA	Fédération Internationale de Football Association
HRP	Humanoid Robotics Project
IEETA	Instituto de Engenharia Eletrónica e Telemática de Aveiro
IMC	Índice de Massa Corporal
INESC	Instituto de Engenharia de Sistemas e Computadores
IP	Internet Protocol
LIACC	Laboratório de Inteligência Artificial e de Ciências de Computadores
Lisp	List Processing
MSL	Middle Size League
MSRS	Microsoft Robotics Studio
ODE	Open Dynamics Engine
SPADES	System for Parallel Agent Discrete Event Simulation
SPL	Standard Platform League
SSL	Small Size League
UA	Universidade de Aveiro
UDP	User Datagram Protocol
UP	Universidade do Porto
XML	Extensible Markup Language
ZMP	Zero Moment Point

Capítulo 1

Introdução

Os robôs que conhecemos nos dias de hoje têm origem em máquinas ancestrais chamadas *automata* cuja origem etimológica provém da mesma palavra grega que nos dá a palavra automático. Estes modelos iniciais não eram verdadeiros robôs pois não possuíam inteligência e não podiam ser instruídos a desempenhar tarefas diferentes daquelas para as quais haviam sido projetados. Os avanços da tecnologia elétrica e eletrônica no século XX possibilitaram aos engenheiros a construção de máquinas mais sofisticadas.

1.1 Motivação

Os robôs têm evidenciado inúmeros progressos, fruto da imaginação e da natureza inventiva do Homem. No futuro, robôs e humanos irão partilhar espaços comuns e trabalhar de forma cooperativa, pelo que a adoção de um corpo humanoide e a garantia da segurança do ser humano tornam-se elementos fundamentais para facilitar a adaptação dos humanos a estes novos parceiros.

A constante evolução da tecnologia permite que sejam atribuídas várias utilidades aos robôs. Desde a indústria à educação, ou apenas para fins lúdicos, são várias as finalidades para as quais os robôs são projetados. Contudo, o conhecimento científico atual pouco se reflete nas questões realmente importantes, como a segurança e a saúde do ser humano. Este trabalho surge da necessidade da constante evolução do controlo dos robôs humanoides, com especial ênfase no robô humanoide Aldebaran NAO, com vista à participação no campeonato mundial de futebol robótico RoboCup, uma prova onde vários investigadores estão envolvidos no desenvolvimento e melhoria das capacidades dos robôs humanoides. Este trabalho está integrado nos objetivos da equipa FC Portugal 3D e é capaz de fornecer uma vantagem à equipa em termos de jogo conferindo uma maior garantia de estabilidade e robustez na execução de comportamentos dos robôs.

1.2 Objetivo

O objetivo desta dissertação é o desenvolvimento de um controlador dinâmico para robôs humanoides NAO capaz de responder a perturbações externas que permita o posicionamento preciso do centro de massa do robô. Este controlador será implementado em linguagem C++, integrado no código do agente humanoide da equipa FC Portugal e testado no SimSpark, o ambiente de simulação oficial da liga de simulação 3D da RoboCup.

1.3 Estrutura do Relatório

Este documento está organizado em nove capítulos cujo conteúdo reflete todo o trabalho desenvolvido. Nesta secção está presente uma breve descrição de cada um dos capítulos:

- **Capítulo 1** - Além da presente secção informativa da estrutura do documento é possível encontrar referências à motivação e objetivos que sustentaram a realização desta dissertação.
- **Capítulo 2** - É efetuada uma descrição da iniciativa RoboCup, os seus objetivos, as ligas que a constituem e as equipas portuguesas presentes na mesma.
- **Capítulo 3** - É apresentada uma descrição geral dos robôs humanoides, introduzindo os mesmos numa perspetiva histórica, e dos projetos mais relevantes associados ao desenvolvimento da robótica humanoide.
- **Capítulo 4** - É analisada a utilização dos ambientes de simulação e os simuladores atualmente utilizados no domínio da robótica humanoide com especial ênfase no SimSpark, o simulador oficial da liga de simulação 3D da RoboCup, e é ainda efetuada uma descrição do agente da equipa FC Portugal.
- **Capítulo 5** - São introduzidos os planos anatómicos de análise de um humanoide e é efetuada uma análise aos métodos de avaliação e análise de estabilidade e a sua relação com a locomoção.
- **Capítulo 6** - São apresentados os métodos de estudo dos problemas de cinemática, como representação de pontos, rotações, transformações homogéneas e a convenção Denavit-Hartenberg para o cálculo da cinemática direta, e os cálculos da cinemática inversa do robô.
- **Capítulo 7** - É introduzido o conceito de controladores, é descrita a implementação de um módulo de comunicação com o monitor RoboViz e a implementação do controlador do posicionamento do centro de massa desenvolvido.
- **Capítulo 8** - É descrito a implementação do movimento de preparação de remate desenvolvido.

- **Capítulo 9** - São apresentadas as principais conclusões alcançadas na realização do projeto sendo também indicados os trabalhos futuros que possam dar continuidade ao trabalho desenvolvido.

Capítulo 2

RoboCup

A RoboCup é uma iniciativa científica internacional que teve início em 1997 com o intuito de promover o desenvolvimento do estado da arte dos robôs inteligentes. Esta iniciativa surge como forma de promover a investigação nas áreas de robótica e inteligência artificial oferecendo um desafio apelativo. O objetivo final desta iniciativa é definido como[1]:

"By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, comply with the official rule of the FIFA, against the winner of the most recent World Cup."[2]

Apesar deste objetivo final permanecer, a iniciativa RoboCup expandiu-se em vários domínios de aplicação baseados nas necessidades da sociedade atual.

2.1 RoboCupJunior

O RoboCupJunior é um projeto educacional para estudantes com idade inferior a 20 anos. Trata-se de um novo método para se compreender a ciência e a tecnologia de forma inovadora onde é permitida a realização de várias experiências ligadas à eletrónica, hardware e software. O projeto oferece ainda uma oportunidade para a aprendizagem de metodologias de trabalho em equipa pela partilha de ideias com os amigos. O objetivo deste projeto é o desenvolvimento de materiais de estudo e métodos de ensino inovadores incentivando a aprendizagem e desenvolvimento de competências na área da robótica e inteligência artificial[4]. A RoboCupJunior possui as seguintes ligas:

- Dance Challenge;
- Soccer Challenge;
- Rescue Challenge.



Figura 2.1: RoboCupJunior Dance, Soccer e Rescue Challenge[3]

2.2 RoboCupRescue



Figura 2.2: RoboCupRescue Simulation league[5]

Uma das grandes aplicações das tecnologias da RoboCup é a operação de busca e salvamento em situações de desastre massivo. A RoboCup iniciou o projeto RoboCupRescue especificamente para promover a pesquisa nesta área de importância vital para a sociedade. O objetivo fundamental deste projeto é o desenvolvimento de metodologias de combate a problemas sociais graves, a divulgação deste novo ramo de investigação na área da inteligência artificial e ainda promover a investigação colaborativa internacional[6]. Este projeto possui as ligas de robôs reais e de simulação.



Figura 2.3: RoboCup@Home league[7]

2.3 RoboCup@Home

A RoboCup@Home é uma nova liga dentro das competições RoboCup que se foca em aplicações do mundo real e na interação Homem-Máquina com robôs autônomos. O objetivo desta liga é fomentar o desenvolvimento de aplicações robóticas que possam ajudar os humanos nas suas vidas.

2.4 RoboCupSoccer

A RoboCupSoccer escolheu o futebol como tema central de pesquisa, esperando que as inovações resultantes sejam aplicadas em problemas importantes para a sociedade e indústria. A criação de uma equipa de robôs totalmente autônomos que atuem de forma cooperativa e exibam comportamentos e estratégias avançadas que permitam obter bons resultados nesta competição exige a investigação em várias áreas da robótica e a melhoria das tecnologias existentes.

2.4.1 Small Size Robot League (SSL)

A Small Size League foca-se no problema da cooperação inteligente multiagente e no controlo num ambiente altamente dinâmico com um sistema centralizado/distribuído híbrido.

Os jogos desta liga são efetuados entre duas equipas de 5 robôs cada. Cada robô deverá cumprir os requisitos definidos pelas regras F180:

"The robot must fit within a 180mm diameter circle and must be no higher than 15cm unless they use on-board vision. The robots play soccer on a green-carpeted field that is 4.9m long by 3.4m wide with an orange golf ball. Robots come in two flavors, those with local on-board vision sensors and those with global vision." [7]



Figura 2.4: Small Size Robot League[7]

O tipo de robôs mais comuns são os que utilizam visão global. Estes usam uma câmara aérea e um computador externo para identificar e localizar os robôs à medida que estes se deslocam no campo.

Um computador externo é utilizado para comunicar os comandos de arbitragem e, no caso de utilizarem visão aérea, a informação da posição aos robôs. Tipicamente este computador executa grande parte, se não todo, o processamento necessário para a coordenação e controlo dos robôs. A comunicação é efetuada sem fios e utiliza tipicamente equipamento transmissor/receptor FM comercial.

2.4.2 Middle Size Robot League (MSL)

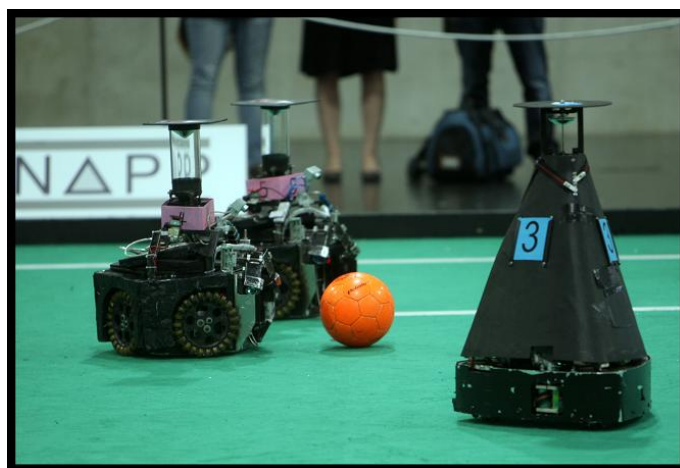


Figura 2.5: Middle Size Robot League[3]

Na Middle Size League, duas equipas de 6 robôs jogam futebol num campo de 18x12 metros. Cada robô está equipado com sensores e um computador de bordo para analisar a situação de jogo e praticar futebol com sucesso.

A cooperação em equipa pode ser estabelecida através de comunicação sem fios tal como a receção de comandos de arbitragem. No entanto, nenhuma intervenção externa por humanos é permitida à exceção das substituições.

Nos últimos anos a investigação nesta competição tem registado bons progressos que se focam na total autonomia e cooperação nas áreas de planeamento e perceção dos robôs.[8]

2.4.3 Humanoid League



Figura 2.6: Humanoid League[7]

Na Humanoid League, robôs autónomos com aparência e sensorização semelhante à humana competem entre si. Ao contrário dos robôs humanoides fora da Humanoid League, a tarefa da perceção e modelação do mundo não é simplificada pela utilização de sensores de distância não humanos.

São inúmeros os desafios técnicos levados a cabo para além da competição de futebol. Locomoção dinâmica, corrida, remate, perceção visual da bola, de outros jogadores e do campo, auto-localização e jogar em equipa estão entre os vários tópicos que são alvos de investigação nesta liga. Muitos dos melhores robôs autónomos humanoides do mundo competem nesta liga.

2.4.4 Standard Platform League (SPL)

Na Standard Platform League, todas as equipas utilizam robôs idênticos. Assim, as equipas concentram-se exclusivamente no desenvolvimento de software e utilizam os robôs da atualidade. As equipas são constituídas por quatro jogadores no total, sendo que um deles desempenha o papel de guarda-redes e os três restantes jogadores são jogadores de campo [9]. Os robôs operam de forma totalmente autónoma sem nenhuma intervenção externa quer por parte de humanos ou computadores. Em 2008 foi efetuada a transição entre o quadrúpede Sony Aibo para o humanoide Aldebaran NAO que continua a ser o robô oficial desta liga até à data [10].



Figura 2.7: Standard Platform League[7]

2.4.5 Simulation League



Figura 2.8: Simulation League[7]

A Simulation League foca-se essencialmente na inteligência artificial e na estratégia de equipa. Na liga de simulação 2D, duas equipas de 11 elementos de software autónomos (agentes), jogam futebol num campo virtual bidimensional representado por um servidor central, SoccerServer. Este servidor tem todas as informações do jogo, nomeadamente, a posição de todos os jogadores e da bola e as variáveis físicas. O jogo baseia-se na comunicação entre o servidor e cada um dos agentes. Por um lado, cada agente recebe os sinais dos seus sensores virtuais (visual, acústico e físico) e poderá, por outro lado, desempenhar comandos básicos como correr, virar e rematar para interagir com o mundo.

O maior desafio nesta liga é concluir a partir das informações que os agentes possuem do mundo a melhor ação a tomar. Uma vez que o jogo se encontra dividido em 6000 ciclos, esta tarefa deve ser concluída no intervalo de tempo de 100 ms, que corresponde à duração de cada ciclo.

A liga de simulação 3D aumenta o realismo do ambiente de simulação utilizado através da adição de uma dimensão extra e um conjunto de físicas mais complexas. No início, o único modelo de robô disponível era um agente esférico. Em 2006, um simples modelo do robô Fujitsu HOAP-2 foi disponibilizado sendo a primeira vez que um modelo humanoide foi utilizado na liga de simulação. Esta mudança alterou o objetivo da liga de simulação 3D do desenvolvimento de comportamentos estratégicos no jogo de futebol para o controlo de baixo nível de robôs humanoides e a criação de comportamentos básicos como caminhar, rematar, virar, levantar, entre outros.

Em 2008, a introdução do modelo do robô NAO à simulação deu uma outra perspetiva à liga. O robô NAO da Aldebaran tem sido o robô oficial para a Standard Platform League desde 2008, e usando o mesmo modelo para as competições de simulação representa uma grande oportunidade para os investigadores que pretendem desenvolver e testar os seus algoritmos e ideias antes de tentar implementar as mesmas no robô real. O interesse na liga de simulação 3D tem crescido rapidamente e a investigação está lentamente a voltar ao desenvolvimento e implementação de comportamentos multiagente de alto nível baseados nas arquiteturas sólidas de comportamentos de baixo nível para as equipas de robôs humanoides realistas.

2.5 FC Portugal

2.5.1 FC Portugal

Criada em fevereiro de 2000, a equipa FC Portugal consiste num projeto de investigação colaborativo entre o Laboratório de Inteligência Artificial e de Ciências de Computadores (LIACC) da Universidade do Porto (UP) e o Instituto de Engenharia Eletrónica e Telemática de Aveiro (IEETA) da Universidade de Aveiro (UA) [11] [12] [13] [14].

A investigação da equipa é focada principalmente na criação de novas metodologias de coordenação, tais como, táticas, formações, posicionamento estratégico e alteração dinâmica de funções. A equipa tem apostado no desenvolvimento de soluções [13] [14] para posicionamento estratégico dos jogadores e na troca de posições entre jogadores para as ligas de simulação, permitindo um maior foco no desenvolvimento de algoritmos de controlo e métodos de coordenação de agentes de software.

A equipa já participou em diversas competições da RoboCup, incluindo simulação 2D, simulação 3D, competição de treinadores, simulação de salvamento e realidade mista, nas quais obteve bons resultados ao longo dos anos [15]:

- Campeã mundial na liga de simulação 2D em 2000
- Campeã europeia na liga de simulação 2D em 2000 e 2001
- Campeã mundial na liga de treinadores em 2002
- Campeã europeia na liga de simulação de resgate e salvamento em 2006
- Campeã mundial na liga de simulação 3D em 2006

- Campeã europeia na liga de simulação 3D em 2006, 2007 e 2012

Outras duas equipas, a atuar em competições diferentes, fazem parceria com a equipa FC Portugal: CAMBADA (liga de robôs médios) e 5DPO (liga de robôs pequenos e liga de robôs médios). Todos os investigadores de cada equipa contribuem entre si para o desenvolvimento e crescimento das equipas.

2.5.2 CAMBADA

O projeto CAMBADA, acrónimo de Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture, começou oficialmente em Outubro de 2003 e constitui a equipa de futebol robótico do IEETA/UA [16] [11] [12] [17]. Esta equipa é formada por investigadores que desenvolvem hardware e software para a participação da equipa na liga de robôs médios onde parte da investigação é efetuada nas áreas de coordenação, jogadas estudadas e posicionamento em campo [18] [19]. A equipa tem obtido excelentes resultados no festival nacional de robótica, tendo sido campeã entre 2007 e 2012, e RoboCup, onde em 2008 conquistou o título de campeã mundial.

2.5.3 5DPO

A 5DPO [11] [12] [20] é uma equipa formada por estudantes, professores e investigadores da FEUP¹, Departamento de Engenharia Eletrotécnica e de Computadores, em parceria com o Instituto de Engenharia de Sistemas e Computadores do Porto (INESC-P) [20]. Esta foi criada em 1997 para competir na liga de robôs pequenos e começou nos robôs médios em 2000. O desenvolvimento da equipa começou em 1998 competindo na liga de robôs pequenos e expandiu-se para a liga de robôs médios em 2001, com o nome 5DPO-2000, e mais tarde para a liga de robôs humanoides, com o nome 5DPO-Human. No conjunto obteve vários segundos e terceiros lugares em competições nacionais e internacionais mas os pontos altos foram as vitórias no campeonato europeu, na liga de robôs pequenos, em 2001, 2006 e 2007 [21].

As principais áreas de investigação da equipa são [20]:

- Auto-localização baseada em sistema de visão;
- Sensores e fusão de dados;
- Controlo em tempo-real;
- Decisão e Cooperação.

2.5.4 Portuguese Team

A equipa Portuguese Team [11] [12] [22] foi criada em 2011 e é composta por membros de três grandes equipas portuguesas (FC Portugal, CAMBADA e 5DPO. Baseada na experiência prévia

¹Faculdade de Engenharia da Universidade do Porto

dos seus membros, a Portuguese Team desenvolveu um agente ao desenvolvido pela equipa FC Portugal, da liga de simulação 3D, e utiliza uma arquitetura distribuída semelhante à utilizada pelos robôs da equipa CMBADA. A equipa foi criada com o intuito de participar na liga de robôs com plataforma normalizada (liga SPL) e sua investigação foca-se essencialmente na criação de novas metodologias de coordenação como táticas, formações, alteração dinâmica de funções e posicionamento estratégico [22].

2.6 Conclusão

Neste capítulo foi apresentada uma introdução à iniciativa RoboCup e aos seus objetivos, na qual os trabalhos desenvolvidos nesta dissertação se inserem. A RoboCup possui diversas ligas com o propósito de promover a investigação na área da robótica e da inteligência artificial quer por parte dos estudantes mais novos quer por parte dos investigadores. Foram ainda introduzidas as equipas portuguesas ligadas a esta iniciativa e as áreas de investigação às quais estão associadas. A prestação das equipas portuguesas tem sido extremamente positiva tendo as mesmas obtido vários troféus nas várias ligas da iniciativa.

Capítulo 3

Robôs Humanoides

Este capítulo apresenta uma descrição geral dos robôs humanoides, introduz os mesmos numa perspectiva histórica e descreve os projetos e robôs humanoides mais relevantes.

3.1 Introdução

Um robô humanoide é um robô cuja aparência global se baseia na de um corpo humano, permitindo a interação com ferramentas ou ambientes específicos para humanos. Genericamente os robôs humanoides possuem cabeça, tronco, duas pernas e dois braços apesar de que alguns possam modelar apenas algumas partes do corpo humano. Os robôs cuja aparência é esteticamente semelhante à dos humanos são designados por andróides. O robô humanoide é um robô autónomo que se adapta às alterações do mundo por forma a atingir o seu objetivo. Algumas das capacidades de um robô humanoide podem incluir:

- Auto-manutenção;
- Aprendizagem autónoma;
- Evitar situações potencialmente perigosas para as pessoas, bens ou para si mesmo;
- Interação segura com os humanos e o mundo.

3.2 História dos robôs

Contexto Cultural

O sonho dos investigadores de robótica tem sido alimentado pela ficção científica. Máquinas semelhantes a humanos ou humanos semelhantes a máquinas são usados muitas vezes como metáforas para os perigos da tecnologia e da hubris humana. A palavra "robot" foi utilizada pela primeira vez pelo escritor Karel Capek em 1920 na peça "*Rossum's Universal Robots*" estando

fortemente associada a um androide ou a um clone. A palavra Checa *robot* significa escravidão e a palavra *robotnik* significa agricultor ou escravo. Nesta peça, um cientista desenvolveu um conjunto de máquinas semelhantes aos humanos com características superiores às da raça humana. As máquinas começam a dominar a raça humana e ameaçam a sua extinção mas esta é salva no último momento.

O escritor de ficção científica Isaac Asimov, em 1941, utilizou pela primeira vez a palavra robótica referindo-se à tecnologia utilizada pelos robôs e previu uma poderosa indústria de robótica. No conto *Runaround*, publicado em 1942, Asimov deu a sua maior contribuição para o mundo da robótica escrevendo as três leis fundamentais da robótica, às quais veio posteriormente a juntar a lei zero. São elas:

- **Lei 0** — Um robô não pode fazer mal à humanidade e nem, por inação, permitir que ela sofra algum mal;
- **Lei 1** — Um robô não pode ferir um ser humano ou, por omissão, permitir que um ser humano sofra algum mal;
- **Lei 2** — Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens entrem em conflito com a Primeira Lei;
- **Lei 3** — Um robô deve proteger a sua própria existência desde que tal proteção não entre em conflito com a Primeira e/ou a Segunda Lei.

Isaac Asimov conseguiu transmitir a ideia de que um robô não teria de ser exatamente como um humano nem teriam de ser maus ou perigosos. A ficção científica deu um grande incentivo aos investigadores atuais em relação ao desenvolvimento da robótica, ainda assim grande parte do público geral vê os robôs como uma ameaça para os humanos ou possui expectativas irrealistas dos mesmos. Este contexto cultural é importante. O homem deve aceitar e possuir expectativas realistas em relação aos humanoides para que estes possam vir a ser utilizados na vida diária e ainda para que estes possam trabalhar de forma colaborativa com os humanos.

Contexto Tecnológico

No passado existiram várias fases distintas nas quais engenheiros e cientistas tentaram construir máquinas semelhantes a humanos ou animais. Cada fase corresponde a avanços significativos nas tecnologias dessa época. Na Grécia antiga existia um grande fascínio pelos *automata*, que eram máquinas que desempenhavam uma determinada função de forma automática e que muitas vezes imitavam ações humanas ou animais. Entre o século 3 A.C. e o século 13 D.C. estes mecanismos eram populares nos meios imperiais na China. Em 1495 Leonardo Da Vinci desenvolveu um dispositivo mecânico semelhante a um cavaleiro armado. Os mecanismos no interior do robô de Leonardo permitiam a este que se movesse como se de uma pessoa real se tratasse.

No século XVIII assistiu-se a um período muito produtivo no desenvolvimento de autómatos capazes de executar movimentos semelhantes aos dos seres humanos, no entanto foi apenas no século XIX que muito deste desenvolvimento se materializou. Durante esta era, um dos inventores



Figura 3.1: Modelo baseado no robô de Leonardo Da Vinci[23]

mais influentes foi Jacques de Vaucanson (1709-1782), que, em 1738, apresentou à Academia de Ciências três dos seus trabalhos. Estes trabalhos apresentados eram compostos por um tocador de flauta, um pato e um tocador de tambor. Ambos os músicos tocavam os instrumentos de forma semelhante ao ser humano, o pato era reconhecido por ser o primeiro automatismo capaz de simular a digestão da sua comida.

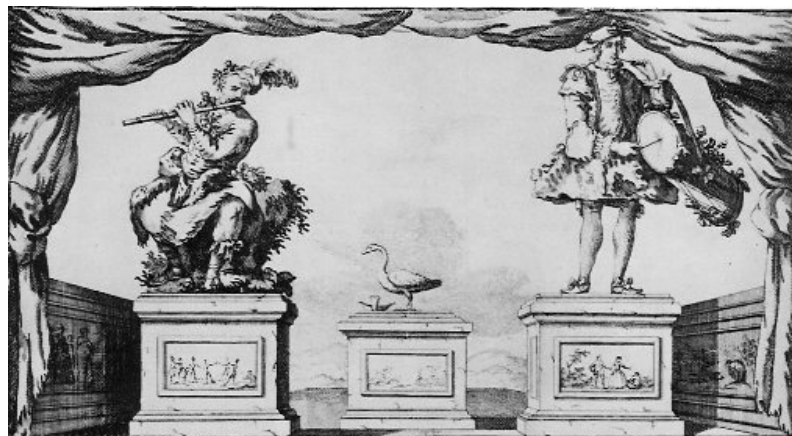


Figura 3.2: Os automata de Jacques Vaucanson[24]

A história dos humanoides no século XIX fica essencialmente marcada pelo aparecimento do Steam Man e do Electric Man. O Steam Man surge do trabalho de Zadoc P. Dederick e de Isaac Grass, que em 1868 desenvolveram e construíram um homem mecânico movido a vapor que puxava uma carroça. O Electric Man surgiu em 1885 pelas mãos de Frank Reade Junior, e trata-se da versão elétrica do Steam Man com algumas modificações.

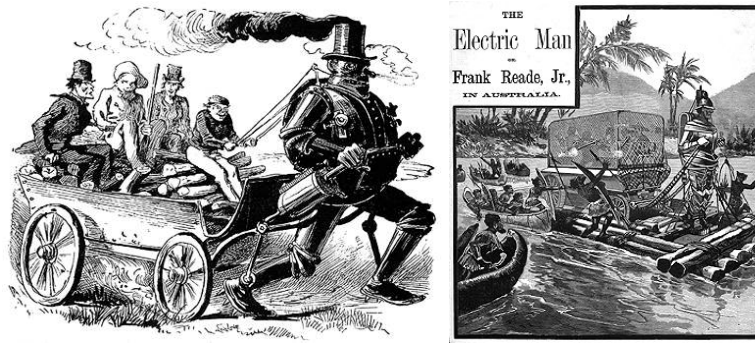


Figura 3.3: Steam Man[25] e Electric Man[26]

O primeiro robô humanoide da era moderna surgiu em 1938 e chamava-se Elektro. Este robô foi desenvolvido pela sociedade Westinghouse e simulava o andar, falar e fumar.

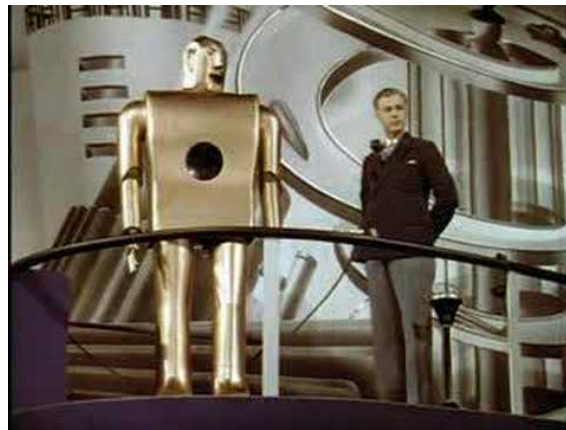


Figura 3.4: O robô humanoide Elektro na exposição de Nova York em 1939[27]

No entanto, verdadeiros robôs só conseguiram aparecer após as décadas de 1950 e 1960 com a invenção dos transistores e circuitos integrados. Em 1961 o braço robótico UniMate foi introduzido nas linhas de produção da General Motors por forma a desempenhar as tarefas árduas e repetitivas [28]. O desenvolvimento de robôs humanoides modernos teve início no final da década de 1960, guiados pelos avanços significativos dos sensores, atuadores e computadores que também permitiram a sofisticação da automação na indústria e a complexidade dos produtos eletrônicos.

Na construção de robôs autônomos é importante considerar uma panóplia de fatores cruciais ao seu desempenho, nomeadamente:

- Autonomia / eficiência energética;
- Poder de processamento;
- Dimensões físicas;
- Complexidade de hardware;

- Programabilidade;
- Estabilidade;
- Precisão sensorial;
- Custos.

De acordo com Gouaillier [29], a relação entre a altura e a massa de um robô fornece à partida uma boa estimativa do seu desempenho energético. Com efeito, um robô mais leve consegue produzir o mesmo movimento que um robô mais pesado, consumindo menos energia; os motores podem ser mais pequenos, e as perdas sob a forma de calor serão muito menos significativas. Robôs pequenos, além de uma melhor dinâmica, são tipicamente menos perigosos para o ser humano, e naturalmente mais baratos. A tabela 3.1 estabelece uma comparação entre alguns robôs humanoides funcionais, em função das suas massas, alturas, IMCs¹ e preços.

Tabela 3.1: Relação entre massa, altura, IMC e preço dos robôs humanoides funcionais

Robô	Massa (Kg)	Altura (m)	IMC	Preço
KHR-2HV	1.3	0.34	10.9	1 K\$
HOAP	7.0	0.50	28.0	50 K\$
NAO	4.5	0.57	13.5	10 K €
QRIO	6.5	0.58	19.0	N/A
ASIMO	54.0	1.30	32.0	N/A
REEM-A	40.0	1.40	20.4	N/A
HRP-2	58.0	1.54	24.5	400 K\$

3.3 Projetos associados à robótica humanoide

Nas últimas duas décadas, o campo da robótica humanoide tem evidenciado melhorias significativas. Esta evolução foi fruto do desenvolvimento de atuadores, computadores e outras tecnologias que permitem realizar a visão da construção de uma máquina com capacidade semelhante à dos humanos. Desde 1960 a indústria da robótica tem vindo a crescer e existem cada vez mais investigadores nesta área existindo vários projetos de investigação associados ao desenvolvimento de robôs humanoides.

3.3.1 Universidade de Waseda

Em 1973, o professor Kato iniciou a investigação moderna em robôs com locomoção bípede na universidade de Waseda com o desenvolvimento do Wabot-1, o primeiro robô humanoide à escala humana. Vários robôs foram subsequentemente desenvolvidos em Waseda sendo o último denominado Wabian-2. Este robô foi utilizado para testar conceitos inovadores de hardware incluindo dedos passivos e pés com estrutura arqueada semelhante à dos humanos.

¹Índice de Massa Corporal



Figura 3.5: Waseda Wabian 2-R[23]

3.3.2 Universidade de Tóquio

Em 1980, Miura e Shimoyama desenvolveram vários pequenos robôs bípedes como o Biper-3. Mais recentemente os robôs H6 e H7 foram desenvolvidos pelo grupo de investigação Inoue em colaboração com a Kawada Industries. Estes robôs possuem junções ativas nos dedos dos pés, permitindo-lhes caminhar mais rapidamente e subir escadas mais altas daquilo que seria possível com pés de um único segmento.

3.3.3 Boston Dynamics

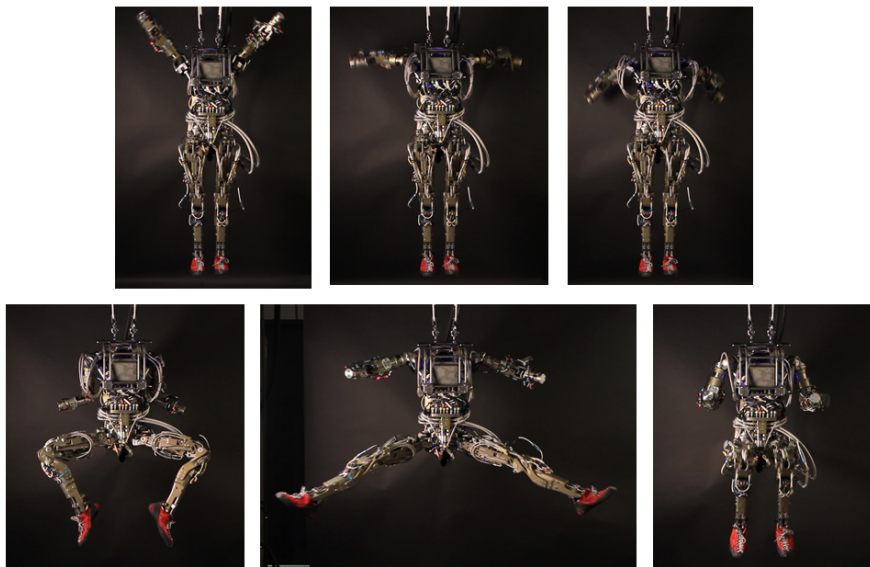


Figura 3.6: Robô PetMan da Boston Dynamics[30]

A Boston Dynamics apresentou recentemente um robô bípede capaz de caminhar a 7,08 km/h. O Projeto é financiado pela DARPA e existem poucas informações quer sobre o hardware utilizado quer sobre o sistema de controlo utilizado. A empresa afirma que o robô é baseado na tecnologia do Big Dog, um quadrúpede desenvolvido pela empresa que utiliza atuadores hidráulicos e está equipado com um motor de combustão interna.

3.3.4 Honda



Figura 3.7: Evolução dos robôs humanoides Honda[31]

A Honda obteve avanços significativos na tecnologia de locomoção bípede durante um programa de investigação e desenvolvimento iniciado em 1986 sendo desenvolvidos os protótipos E1 até ao E6. A abordagem da Honda na construção de robôs bípedes foi bem sucedida e serviu de inspiração para os trabalhos desenvolvidos nesta área. O robô Asimo, introduzido em 2000, é o modelo mais avançado da Honda. Desde a sua introdução o robô tem registado várias melhorias sendo que a versão de 2011 é capaz de correr de forma fluida a uma velocidade de 9 km/h.

3.3.5 HRP / AIST

O instituto de investigação Advanced Industrial Science and Technology (AIST) do Japão desenvolveu alguns robôs juntamente com a Kawada Industries e algumas universidades japonesas. O desenvolvimento começou com o "Humanoid Robotics Project"(HRP) financiado pelo ministro da economia Japonesa em 1997. O projeto começou com um robô P3 comprado à Honda no qual desenvolveram características únicas sendo posteriormente denominado HRP-1. O último robô, o HRP-4 foi desenvolvido com o intuito de ser utilizado como auxílio na vida diária das pessoas, possuindo um corpo leve e esguio e um preço mais baixo que os seus precedentes.



Figura 3.8: Robô HRP-4[32]

3.3.6 Toyota



Figura 3.9: Toyota Motor Corporation partner robot[33]

A Toyota apresentou vários "partner robots" que podem realizar várias tarefas como caminhar ou tocar instrumentos musicais. Um robô que teve um dos desenvolvimentos mais notáveis é o *Toyota Running Robot* do qual existem poucas informações mas que consegue correr a uma velocidade de 7 km/h através das suas junções ativas nos dedos dos pés.

3.3.7 Aldebaran Robotics

A Aldebaran Robotics é uma empresa francesa fundada em 2005 que se apercebeu que a era dos robôs pessoais se está a aproximar. A empresa dedica-se a desenvolver, produzir e comercializar robôs autónomos, o robô NAO.



Figura 3.10: Aldebaran NAO[34]

3.4 Aldebaran NAO

O NAO é um robô humanoide autônomo, programável e de tamanho médio desenvolvido pela empresa francesa Aldebaran Robotics tendo como principal motivação ao seu desenvolvimento a necessidade de fornecer ao mercado da investigação e ensino um robô humanoide a um preço mais reduzido que os robôs até à data existentes. Este robô foi projetado por forma a responder de forma rápida a um variado conjunto de estímulos, oferecendo igualmente um vasto conjunto de tarefas que este poderia executar, desde andar até apanhar objetos, ou mesmo recuperar autónoma e automaticamente de quedas.

Em 2008 o NAO substituiu o Sony Aibo na Standard Platform League. O mais recente modelo, o NAO H25 possui as seguintes características[35]:

- **Dimensões** — $573 \times 275 \times 311$ mm;
- **Massa** — 5,2 Kg;
- **Autonomia** — 60 a 90 min;
- **Graus de liberdade** — 25;
- **Processador** — ATOM Z530 1.6 GHz;
- **Visão** — 2 câmaras SOC 1288×968 ;
- **Conexão** — Wifi e Ethernet;
- **Unidade de Inércia** — 2 giroscópios de 1 eixo e um acelerómetro de 3 eixos.

3.5 Conclusão

Neste capítulo foi apresentada a noção de robôs humanoides e foi descrita a história do seu desenvolvimento. A palavra robô surgiu apenas em 1920 pelas mãos do escritor de ficção científica Karel Capek. No entanto, apenas após a década de 1950, com o aparecimento dos transístores e circuitos integrados é que o desenvolvimento de robôs e fundamentalmente do campo da robótica foi possível, o qual tem vindo a crescer nos últimos anos dada a sua importância. Devido à sua capacidade de interação com ambientes humanos, inúmeros projetos de desenvolvimento de robôs humanoides surgiram com o intuito de os mesmos trabalharem de forma cooperativa com os humanos e até mesmo de os substituir em ambientes perigosos. Vários projetos como o robô Asimo da Honda têm impulsionado este campo de investigação, no entanto a abordagem da Aldebaran Robotics, através da construção de um robô humanoide mais acessível, permitiu o acesso à investigação de robótica humanoide por parte de vários institutos de desenvolvimento e ensino. A existência de inúmeros ambientes de simulação com modelos precisos do robô fomentou ainda mais o interesse e desenvolvimento nesta área.

Capítulo 4

Ambientes de simulação e agente FC Portugal

4.1 Introdução

O objetivo fundamental da utilização de ambientes de simulação é o desenvolvimento de um agente virtual capaz de pensar e agir de acordo com a sua perceção do mundo. Desta forma, o conhecimento adquirido poderá ser transferido para o robô real. Não é razoável o teste e desenvolvimento de qualquer algoritmo diretamente num robô real visto que a aprendizagem necessita de várias iterações, os prejuízos adjacentes a quedas de robôs podem ser elevados e a depuração do código é realizada mais facilmente em ambientes de simulação. Os ambientes de simulação devem ser robustos e precisos, emulando o mundo real e o robô da melhor forma possível, permitindo desta forma uma melhor portabilidade entre o código do robô simulado e do robô real. No entanto, os simuladores físicos contêm algumas simplificações na modelação do mundo real pelo que o resultado numa simulação e no mundo real poderá diferir. Este problema é denominado *reality gap* no campo da robótica evolutiva[36].

4.2 Vantagens da simulação

As plataformas robóticas, quer sejam os mais simples braços articulados ou os mais complexos robôs humanoides, são usualmente muito caras. A utilização de ambientes de simulação para investigação, desenvolvimento e teste na robótica possui várias vantagens face à utilização de robôs reais[37]. As principais vantagens da utilização de ambientes de simulação são:

- Mais barato que robôs reais;
- Facilidade de desenvolvimento e teste de novos modelos de robôs;
- Facilidade de teste de novos algoritmos;

- Diminuição de tempo de desenvolvimento e teste;
- Facilita o estudo de métodos de coordenação multiagente;
- Todos os testes podem ser feitos sem danificar o robô real;
- Para otimização e testes repetitivos, a utilização de um modelo virtual reduz substancialmente o tempo entre cada iteração.

A maior desvantagem na utilização destes ambientes prende-se com o *reality gap*, pelo que os agentes de software desenvolvidos em simulação necessitam de pequenos ajustes para serem utilizados em ambiente real. O volume de alterações necessárias é proporcional à precisão da simulação.

4.3 SimSpark

O SimSpark é o simulador oficial da liga de simulação 3D da RoboCup. Este consiste numa plataforma de simulação genérica para simulações físicas multiagente. O simulador foi desenvolvido sobre uma framework de aplicação flexível (Zeitgeist) para ser um simulador genérico, capaz de simular qualquer coisa, desde o lançamento de um projétil a um grande jogo de futebol. A simulação consiste em três partes importantes: o servidor, o monitor e os agentes.

4.3.1 Servidor

O servidor do SimSpark efetua a gestão da simulação, modificando o estado de simulação num ciclo infinito de execução. Toda a simulação é efetuada pelo servidor que recebe as ações dos agentes e lhes fornece o estado e as informações dos seus sensores. O SimSpark utiliza o motor físico Open Dynamics Engine (ODE) para simular toda a dinâmica dos corpos e objetos. Uma outra responsabilidade do servidor é manter o registo dos agentes conectados e a cada ciclo de simulação ele envia para os agentes a informação dos seus sensores, e recebe sequências de ações dos agentes para alterar os seus atuadores. O SimSpark implementa um modelo simples de simulação baseado em eventos que executa imediatamente cada ação recebida pelos agentes. Ele não tenta compensar latência na rede ou alterações nos recursos de computação durante a simulação. A figura 4.1 representa a possibilidade de dessincronização entre o agente e o servidor: uma ação recebida num ciclo n apenas é executada no ciclo $n+1$ [38].

A implementação do servidor segue a seguinte arquitetura, ilustrada na figura 4.2:

- **Open Dynamics Engine (ODE)** - Corresponde ao motor da física do jogo. Permite simular a dinâmica do sistema e as propriedades físicas dos objetos simulados fornecendo tipos de juntas avançados e integração e cálculo de deteção de colisões e fricção. A ODE é particularmente útil para simular objetos em ambientes de realidade virtual;

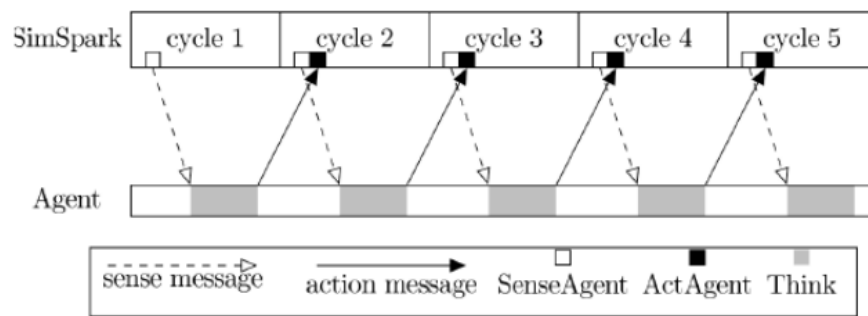


Figura 4.1: Diagrama de comunicação entre o servidor e o agente[38]

- **Zeitgeist** - É a framework utilizada para o tratamento de objetos de dados e componentes funcionais de um sistema de forma uniforme e que segue estritamente o paradigma de programação orientada a objetos utilizando C++;
- **Motor de simulação** - Representa o núcleo do simulador. Recebe mensagens com ações dos agentes, executa as operações de simulação e envia uma mensagem de resposta ao agente com as informações do ambiente;
- **System for Parallel Agent Discrete Event Simulation(SPADES)** - Fornece uma camada middleware que pode estar presente entre o agente e o motor de simulação para gerir a distribuição da simulação entre máquinas e é suficientemente robusto para variações na rede e sobrecargas nas máquinas. O SPADES não requer que os agentes sejam escritos numa linguagem específica.

Os agentes podem interagir diretamente com o motor de simulação ou através da camada de middleware providenciada pelo SPADES. O motor de simulação funciona como servidor, tratando as mensagens dos agentes e respondendo com outras mensagens. No caso da simulação de futebol humanoide, em cada ciclo de simulação o agente envia a mensagem ao servidor contendo informação sobre os atuadores como as juntas. A mensagem do servidor para o agente contém informação temporal e informação específica do domínio da aplicação. Esta informação inclui o estado do jogo, dos sensores do robô nomeadamente as juntas, giroscópios, sensores dos pés e informação da visão. As mensagens são construídas utilizando um formato semelhante a Lisp.

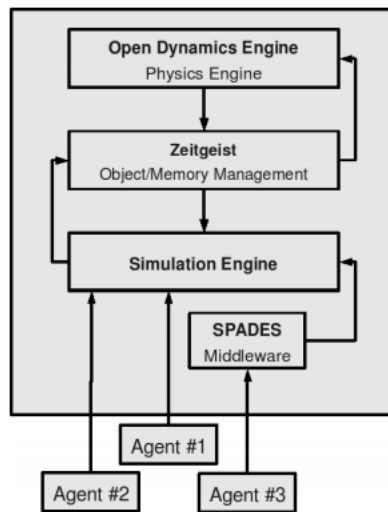


Figura 4.2: Arquitetura do servidor SimSpark[39]

4.3.2 Monitor

O monitor do SimSpark é o elemento responsável pela renderização da simulação. Ele liga-se a uma instância do servidor da qual recebe continuamente um fluxo de dados que descreve o estado da simulação. No caso particular da simulação humanoide, este disponibiliza informação adicional como o nome das equipas, o tempo de jogo, o estado do jogo e o resultado. O monitor pode ser configurado para ler ficheiros de registo de jogos anteriores de modo a repetir um jogo previamente gravado. O formato do ficheiro de *log* é idêntico ao protocolo utilizado na rede entre o servidor e o monitor.



Figura 4.3: Monitor genérico do SimSpark



Figura 4.4: NAO real vs modelo simulado[40]

4.3.3 Agente

O SimSpark utiliza um modelo virtual do robô NAO da Aldebaran com 57 cm de altura e 4,5 Kg de massa. A sua estrutura bípede com 22 graus de liberdade permite uma multiplicidade de movimentos do mesmo, aliado a uma grande mobilidade. Tal como se pode observar na figura 4.4 o modelo simulado é bastante semelhante ao robô real. O modelo simulado está equipado com um poderoso conjunto de sensores e atuadores que fornecem um vasto conjunto de informação para o desenvolvimento do agente:

- O robô NAO possui um giroscópio e um acelerómetro para manter registo dos seus movimentos radiais e axiais no espaço tridimensional;
- Para detetar contacto com o chão ou outros objetos na simulação, um sensor de força em cada pé indica a pressão atual neste;
- Para obter informação visual sobre diferentes objetos no seu ambiente o robô NAO possui um sensor de visão no centro da sua cabeça;
- Para efeitos de comunicação, o robô está equipado com um atuador de fala e o correspondente sensor;
- O sensor de estado de jogo é utilizado para informar sobre o tempo de jogo e sobre o estado do jogo;
- A posição de cada junta é representada por um sensor e manipulada pelo respetivo atuador. A lista de todas as juntas e a sua disposição e orientação está esquematizada na figura 4.5.

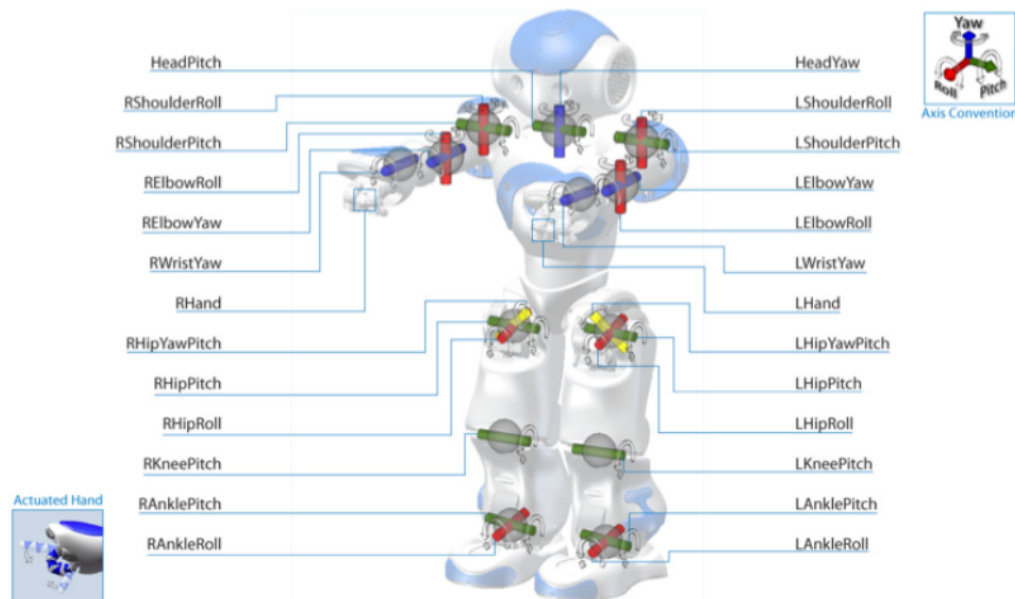


Figura 4.5: Disposição e orientação das juntas do NAO[41]

4.4 Alternativas ao simulador

Existem vários simuladores disponíveis que podem ser utilizados para o desenvolvimento e teste de comportamentos em robôs humanoides. Entre eles existem alguns simuladores com características semelhantes ao SimSpark tais como:

- **Microsoft Robotics Studio (MSRS)** - Aplicação freeware baseada em ambiente Windows para criar aplicações robóticas para propósitos académicos e comerciais. Uma grande variedade de plataformas hardware são suportadas;
- **Webots** - Ferramenta de simulação desenvolvida pela Cyberbotics. Permite ao utilizador modelar, programar e simular robôs móveis. As bibliotecas de robôs incluídas permitem ao utilizador transferir os programas de controlo a vários robôs reais disponíveis comercialmente;
- **Player/Stage/Gazebo** - Projeto open-source que corre em ambientes UNIX. Player é um servidor de rede para o controlo de robôs. Ele fornece uma interface limpa e simples dos sensores e atuadores do robô através de uma rede IP. Stage e Gazebo fornecem ambientes de simulação 2D e 3D respetivamente, e são capazes de simular uma população de robôs a interagir com o ambiente. Além disso, o Gazebo permite a simulação das propriedades físicas dos objetos simulados.

4.5 Ambiente e objetos do campo

4.5.1 Dimensões

As dimensões do campo simulado são de 30 por 20 metros. Um campo da FIFA tem aproximadamente 110 por 70 metros, fazendo com que as dimensões do campo simulado sejam 35% do campo real. A área central tem um raio de 2 metros, cada baliza tem 2,1 por 0,6 metros e uma altura de 0,8 metros, a grande área tem 3,9 por 1,8 metros e por fim a bola tem 0,04 metros de raio e um peso de 26 gramas [42].

Em cada canto existe uma bandeira distinta para que os agentes possam localizar-se no campo. A posição de cada bandeira está acessível ao agente, o qual as distingue através do seu código como mostrado na figura 4.6.

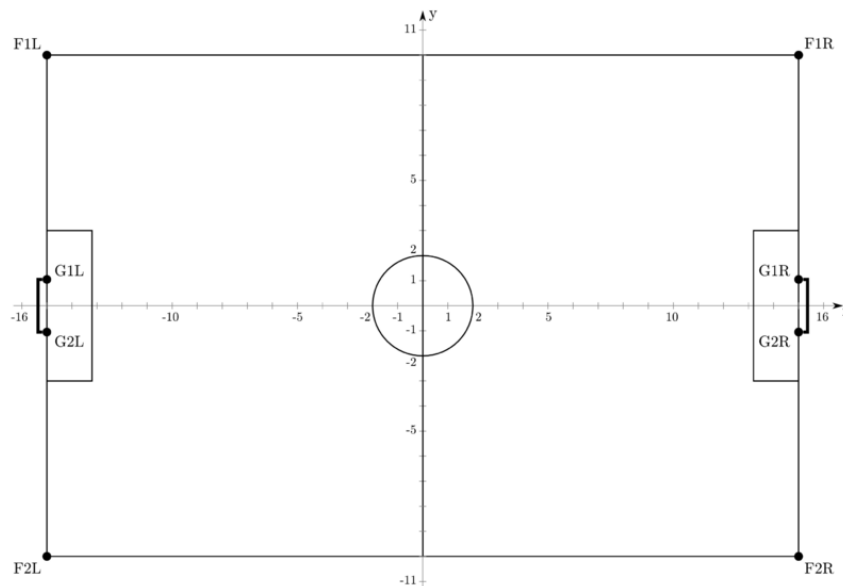


Figura 4.6: Dimensões do campo simulado [42]

4.5.2 Regras

As regras de jogo serão processadas por dois tipos de árbitros [43], um árbitro automático e um humano.

O árbitro automático está responsável por gerir o tempo de cada parte do jogo, guardar a identificação do último jogador a tocar na bola, detetar e atualizar os golos e detetar a saída da bola da área limite do jogo. O árbitro humano é responsável apenas por iniciar a partida.

4.6 Agente FC Portugal 3D

4.6.1 Arquitetura

O agente da equipa FC Portugal 3D [44] está dividido em vários pacotes, cada um com uma função específica. A maior parte do código foi desenvolvido na linguagem C++ e segue totalmente o paradigma orientado a objetos. A descrição desses pacotes é [45] [44]:

- **WorldState:** O pacote WorldState contém as classes para gerir a informação do ambiente. Este ambiente inclui objetos presentes no campo, estado do jogo, e características do campo;
- **Physics:** Pacote que contém classes responsáveis pelos dados do agente. Inclui a estrutura do robô, a interface de cinemática, controlo de baixo nível das juntas e módulos de planeamento de trajetória;
- **Geometry:** Contém classes para ajudar nas necessidades de geometria, entre as quais o processamento de vetores, matrizes, polígonos, linhas, retângulos e círculos;
- **Skills:** É neste pacote que são definidas as classes relacionadas com habilidades do robô, tais como andar, chutar, levantar, rodar, etc;
- **Utils:** Contém classes que, não estando relacionadas diretamente com o agente, devem existir para este funcionar, tais como comunicação entre agentes, comunicação entre o agente e o simulador e processamento de API's;
- **Logger:** Usado essencialmente para guardar todo o tipo de informação em ficheiro. Apesar de podermos guardar informação arbitrária, contém métodos específicos para guardar informação relativa a posições e sensores;
- **Optimization:** Contém classes usadas no processo de otimização;
- **Strategy:** Contém classes com as funções de alto nível do agente.

A figura 4.7 mostra a estrutura da arquitetura do agente.

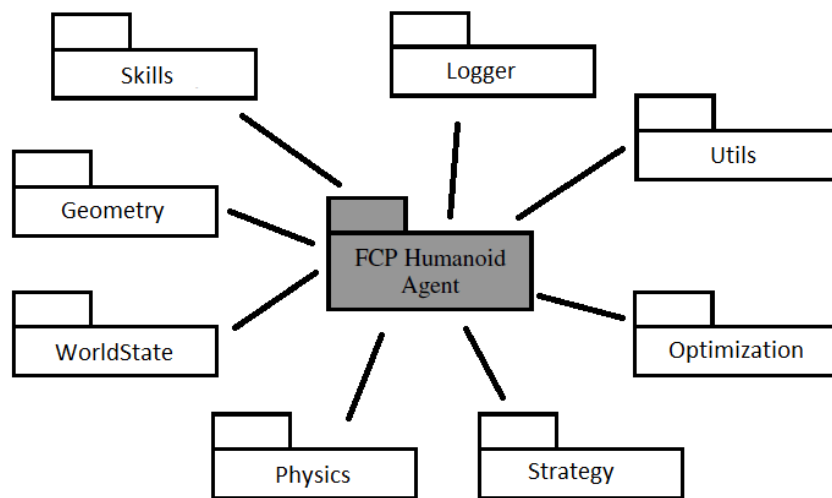


Figura 4.7: Arquitetura do agente da equipa FC Portugal 3D

4.6.2 Modelo

A construção do modelo do agente tem como base a leitura de um ficheiro XML que contém a estrutura do robô. Neste ficheiro são definidos os membros do corpo, articulações e sensores e o seu posicionamento no corpo. A seguir é mostrado um excerto de um ficheiro de configuração usado:

Listagem 4.1: Excerto de um ficheiro XML usado na configuração do modelo do robô NAO.

```
<robot type="humanoid" rsgfile="nao/nao.rsg">
  <bodypart name="torso" mass="1.2171" />
  <bodypart name="lhip1" mass="0.09" />
  <bodypart name="rhip1" mass="0.09" />
  ...

  <joint name="lleg2" per="llj2" eff="lle2"
    axis="0,1,0" min="-25" max="45">
    <anchor part="lhip1" x="0" y="0" z="0" />
    <anchor part="lhip2" x="0" y="0" z="0" />
  </joint>
  <joint name="rleg2" per="rlj2" eff="rle2"
    axis="0,1,0" min="-45" max="25">
    <anchor part="rhip1" x="0" y="0" z="0" />
    <anchor part="rhip2" x="0" y="0" z="0" />
  </joint>
  ...
</robot>
```

Uma vez que a informação sobre a estrutura do robô é guardada num ficheiro XML, o código do agente da equipa torna-se mais permitindo, com o mesmo código, definir e controlar outro tipo de robôs. Várias propriedades e ferramentas têm base na informação presente neste ficheiro, como é o caso das matrizes de transformação utilizadas na cinemática.

4.7 Conclusão

Neste capítulo foram introduzidos vários ambientes de simulação. As principais vantagens da utilização dos ambientes de simulação centram-se na redução de custos e na facilidade de teste e desenvolvimento de novos comportamentos e algoritmos. No entanto, é necessário que o ambiente de simulação possua um modelo do robô real o mais preciso possível para que os resultados obtidos sejam válidos. Foi ainda apresentada uma descrição da arquitetura do agente utilizado pela equipa FC Portugal no qual será desenvolvido o controlador de posicionamento do centro de massa por forma a assegurar a estabilidade do robô.

Capítulo 5

Análise de estabilidade

Um humanoide é descrito como sendo um robô cuja construção e movimentos se aproximam das características apresentados por um ser humano. Neste sentido, a locomoção humanoide e a análise da sua estabilidade constituem um elemento essencial ao desenvolvimento do robô. Esta análise é tipicamente definida de acordo com três planos:

- **Sagital** - Define as características de locomoção no eixo x do robô, i.e. andar para a frente e para trás;
- **Frontal** - Define as características de locomoção no eixo y do robô, i.e. andar para o lado;
- **Transverso** - Define as características de locomoção em torno do eixo z, i.e. rodar sobre si.

O plano sagital, é o plano que atravessa longitudinalmente o corpo dividindo-o nas partes esquerda e direita. O plano frontal divide o corpo nas partes anterior e posterior. O plano transverso é o plano ortogonal aos dois planos anteriores. Este plano divide o corpo nas partes superior e inferior. A figura 5.1 esquematiza os planos e lados referidos.

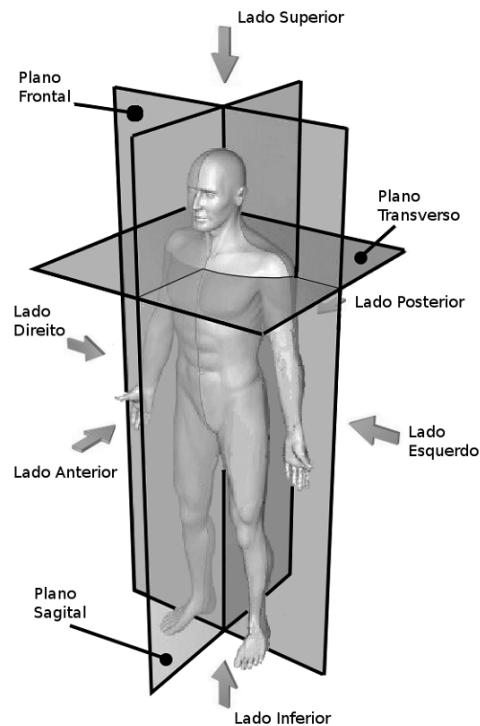


Figura 5.1: Planos anatômicos de um humanoide[46]

5.1 Configuração de locomoção de robôs

A locomoção bípede garante uma maior agilidade dos robôs em evitar obstáculos e permite uma maior facilidade na interação robô humano e na utilização dos robôs em ambientes humanos tendo como compromisso a eficiência e a complexidade do método de locomoção. Quanto menor o número de pernas, mais complexo se torna o desenvolvimento da locomoção devido aos problemas que surgem com a estabilidade. É por esta razão que tipicamente animais hexápodes sabem caminhar à nascença enquanto que os humanos apenas conseguem caminhar a partir dos 11 meses de idade.

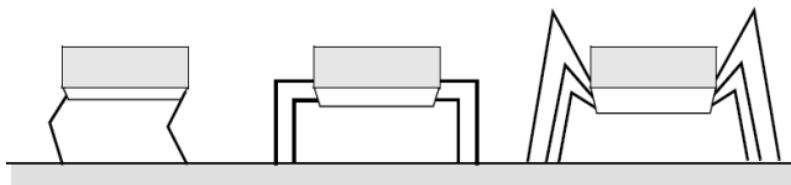


Figura 5.2: Configuração bípede, quadrúpede e hexápode[47]

5.2 Polígono de suporte

O polígono de suporte corresponde à superfície de contacto entre o pé e o chão quando apenas um pé está no chão. Quando ambos os pés estão no chão, o polígono de suporte é designado pelo polígono convexo que contém ambos os pés. Este polígono pode ser encontrado desenhando linhas retas que unem as partes exteriores de cada superfície de contacto. A figura 5.3 representa no plano transversal a derivação acima descrita.

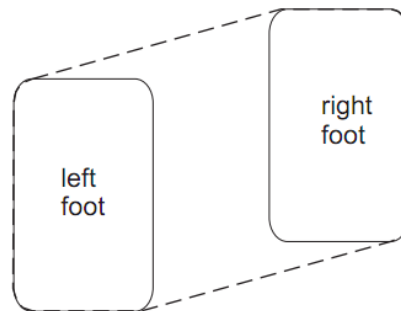


Figura 5.3: Derivação do polígono de suporte no plano transversal [48]

5.3 Centro de Massa (CoM)

Em física, o centro de massa corresponde à média pesada da localização de todas as massas num ou mais corpos. Inúmeros cálculos em mecânica são simplificados quando as quantidades são referidas à posição do centro de massa. O centro de massa CoM de um sistema de partículas com massa total M vem definido em relação às suas posições r_i e massas m_i como:

$$CoM = \frac{1}{M} \sum m_i r_i \quad (5.1)$$

É possível calcular várias propriedades interessantes a partir do centro de massa do corpo tal como o momento total de um sistema de partículas que é dado por:

$$p = Mv_{cm}, \quad (5.2)$$

onde M indica a massa total e v_{cm} a velocidade do centro de massa. A velocidade do centro de massa pode ser calculada através da derivada temporal da posição do centro de massa.

O centro de massa do corpo é um critério de estabilidade largamente utilizado para o controlo do equilíbrio estático. Um robô é considerado estaticamente estável quando a projeção no solo do seu centro de massa segundo a força da gravidade está contida no interior do seu polígono de suporte 5.4.

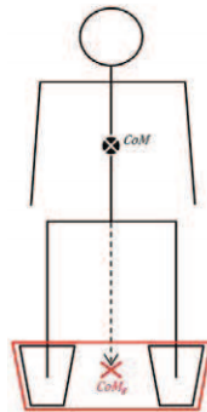


Figura 5.4: Projeção do centro de massa e polígono de suporte[49]

5.4 Centro de Pressão (CoP)

O centro de pressão é o ponto no plano de suporte do corpo onde atua a soma total das forças de contacto, causando uma força mas não um momento. Quando o robô está em pé, a parte do corpo em que são exercidas as forças de contacto são os pés, onde as componentes das forças de contacto normais da superfície do pé de suporte são chamadas forças de pressão. Um exemplo da localização do centro de pressão no plano sagital está presente na figura 5.5. Nesta podemos ver que a força resultante é igual à soma das forças de pressão F_{p1} e F_{p2} que atuam no centro de pressão.

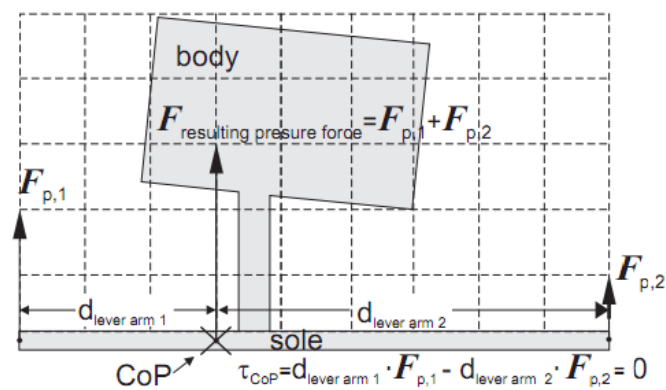


Figura 5.5: Cálculo do centro de pressão[48]

A formulação geral do centro de pressão é dada por:

$$CoP = \frac{z \wedge M_{gr}}{F_{gr} \cdot z}, \quad (5.3)$$

onde M_{gr} e F_{gr} representam respetivamente os momentos e forças de reação do solo em relação ao centro de pressão e z é o versor segundo a direção z , perpendicular ao plano do solo.

5.5 Zero Moment Point (ZMP)

Enquanto o Centro de Pressão está relacionado com as forças de contacto, o ZMP está relacionado com as forças que atuam no robô causadas pela gravidade e pelos seus atuadores. O ZMP é o ponto na superfície onde a soma de todos os momentos relacionados com as forças geradas pela gravidade e pelos atuadores são iguais a zero, momentos esses que atuam de forma tangencial à superfície de contacto. Se a superfície de contacto for horizontal, apenas são considerados os momentos resultantes de movimento de pitch¹ e roll², uma vez que a componente yaw³ é coplanar ao plano de suporte. Desde que as forças de contacto apareçam numa única superfície planar, o ZMP e o CoP coincidem[50]. Este resultado acontece devido ao facto da componente vertical da força resultante causada pela gravidade e os atuadores atuarem no ZMP. Assim, o ZMP deve coincidir com o CoP para que o robô esteja em equilíbrio. De uma outra forma a força de pressão resultante e a resultante da componente vertical das forças gravitacionais e do atuador aplicariam momento ao pé que por sua vez iria rodar. O ZMP é dado pela equação

$$ZMP = \frac{z \wedge M_{gi}}{F_{gi} \cdot z}, \quad (5.4)$$

onde M_{gi} e F_{gi} representam respetivamente os momentos e forças devido à gravidade e inércia e z é o versor segundo a direção z , perpendicular ao plano do solo.

5.6 Locomoção estática e dinâmica

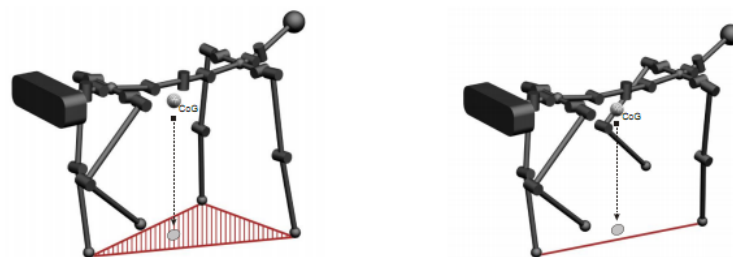


Figura 5.6: Equilíbrio estático e dinâmico[47]

A diferença entre a estabilidade das diferentes configurações apresentadas em 5.1 pode ser explicada com os conceitos de equilíbrio estático e dinâmico. A locomoção estática assume que o robô está sempre numa posição estaticamente equilibrada pelo que, mesmo que esta pare em

¹Rotação em torno do eixo y

²Rotação em torno do eixo x

³Rotação em torno do eixo z

qualquer momento, o robô permanecerá numa posição estável. A velocidade de locomoção estática é baixa, o que significa que a dinâmica do sistema pode ser ignorada. Quando isto é feito, a projeção do centro de massa no plano horizontal é igual ao centro de pressão, e para que o robô se mantenha estaticamente equilibrado, o centro de massa e o centro de pressão devem estar contidos no polígono de suporte. A locomoção dinâmica é usualmente mais rápida em relação à estática. Nesta, o centro de massa pode estar fora do polígono de suporte durante um tempo limitado, o que significa que o robô estaria estaticamente desequilibrado durante a fase de suporte único.

5.7 Conclusão

Neste capítulo foram introduzidos os métodos mais utilizados de análise de estabilidade e foi apresentado o conceito de locomoção estática e dinâmica. O método de análise da posição do centro de massa é um método bastante útil na análise de estabilidade para métodos de locomoção estática garantindo assim que o robô está estaticamente equilibrado. Por outro lado, os métodos de análise do CoP e do ZMP permitem uma análise da estabilidade do robô durante a locomoção dinâmica do robô. Estes métodos, juntamente com a análise da cinemática do robô, são peças fundamentais para o desenvolvimento do controlador do centro de massa do robô.

Capítulo 6

Análise cinemática

A cinemática é o ramo da física que estuda o movimento dos corpos. Os manipuladores robóticos são compostos por segmentos ligados por juntas formando uma cadeia cinemática. As juntas são tipicamente rotativas, permitindo uma rotação relativa entre dois segmentos, ou prismáticas, quando permitem um movimento linear entre dois segmentos. Estas juntas encontram-se representadas na figura 6.1.

A utilização de transformações geométricas, também chamadas de transformações rígidas, para descrever o movimento dos corpos de um sistema mecânico simplifica a derivação das duas equações de movimento, e é fundamental para a análise dinâmica do sistema.

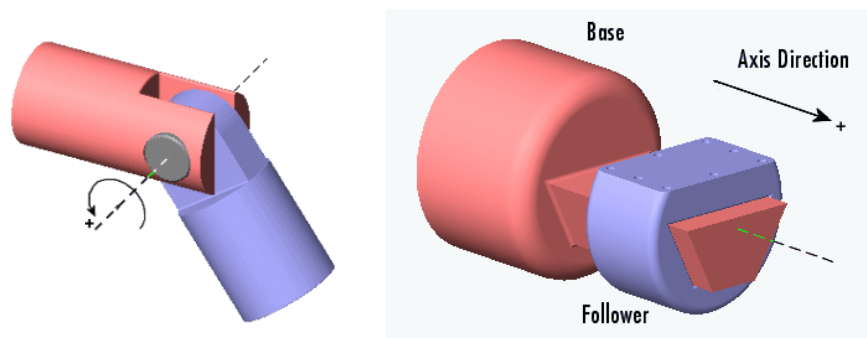


Figura 6.1: Junta rotativa e junta prismática[51]

6.1 Movimentos rígidos e transformações homogêneas

Para o desenvolvimento das equações cinemáticas de um robô há necessidade de estabelecer vários sistemas de coordenadas para representar as posições e orientações dos seus corpos rígidos constituintes. Também é necessário conhecer as transformações de coordenadas entre esses sistemas, de modo a possibilitar que vetores representativos de posições, velocidades e acelerações, dados num determinado sistema de coordenadas, possam ser representados noutros sistemas de coordenadas. Neste sentido serão consideradas as operações de rotação e de translação de um sistema em relação a um outro sistema e apresentada a noção de transformação homogênea, muito usuais na área da robótica.

6.1.1 Representação de posições

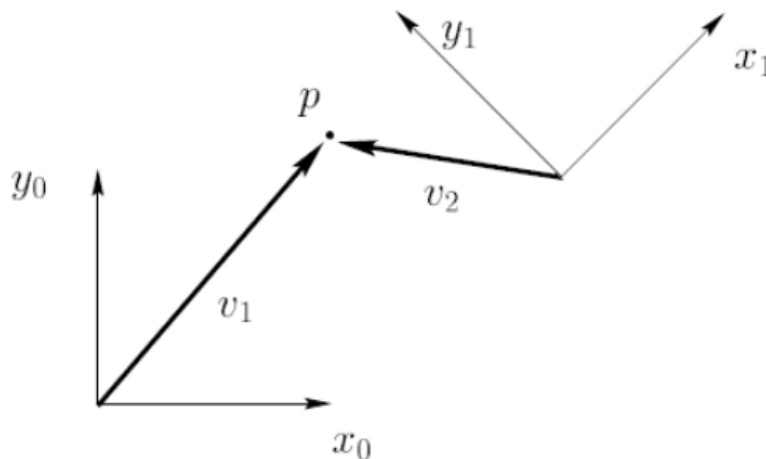


Figura 6.2: Representação de um ponto p em dois referenciais distintos[52]

A figura 6.2 apresenta dois referenciais $o_0x_0y_0$ e $o_1x_1y_1$, em que este último está rodado de 45° em relação ao primeiro. Pode-se verificar que o ponto p pode ser expresso em coordenadas relativas a ambos os referenciais pelo que para a distinção dos mesmos será adotado um sobrescrito. Assim, as coordenadas de p em relação a ambos os referenciais poderiam ser representadas por:

$$p^0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}, p^1 = \begin{bmatrix} -2,8 \\ 4,2 \end{bmatrix} \quad (6.1)$$

Dado que a origem de um referencial de coordenadas é apenas um ponto no espaço, pode-se atribuir coordenadas que refletem a posição de um referencial de coordenadas em relação ao outro.

Na figura 6.2 poderíamos ter:

$$o_1^0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, o_0^1 = \begin{bmatrix} -10,6 \\ 3,5 \end{bmatrix} \quad (6.2)$$

Este método permite a translação de coordenadas entre referenciais. No entanto, é necessário ter em conta a orientação dos referenciais para que um ponto possa ser corretamente definido em relação a diferentes referenciais.

6.1.2 Representação de rotações

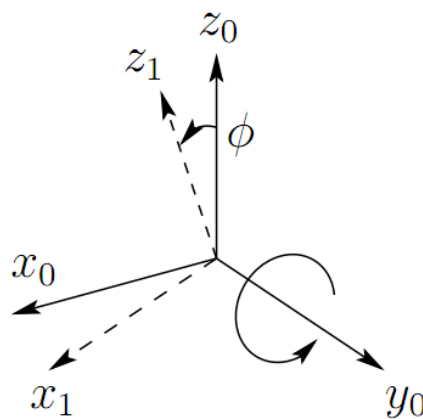


Figura 6.3: Rotação do referencial em y_0 de um ângulo ϕ [52]

Uma matriz de rotação é uma transformação linear que quando multiplicada por um qualquer vetor provoca uma rotação desse vetor segundo um eixo mantendo a sua norma (comprimento). Quando aplicada num referencial, permite-nos representar a orientação do mesmo em relação a outro referencial. Assim, é possível, recorrendo a uma matriz deste tipo, representar a orientação de um corpo rígido em relação a um referencial. Uma matriz que efetua a rotação de um referencial $o_1x_1y_1z_1$ para um referencial $o_0x_0y_0z_0$ é definida em 6.3 onde cada coluna representa a projeção de cada um dos eixos do referencial $o_1x_1y_1z_1$ em $o_0x_0y_0z_0$.

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \quad (6.3)$$

As matrizes de rotação são matrizes quadradas onde as colunas e linhas têm norma unitária. O conjunto destas matrizes de tamanho $n \times n$ forma o grupo conhecido como Grupo Especial Ortogonal $SO(n)$, e goza das seguintes propriedades:

- $R^T = R^{-1}$;

- As linhas e colunas de R são mutuamente ortogonais;
- Cada linha e coluna de R é um vetor unitário;
- $\det R = 1$.

6.1.3 Transformações homogêneas

As transformações homogêneas combinam as operações de rotação e translação e são utilizadas para derivar as equações de cinemática direta de um manipulador ou de uma cadeia cinemática.

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, R \in SO(3), d \in \mathfrak{R}^3 \quad (6.4)$$

O conjunto de matrizes da forma definida em 6.4, em que R e d correspondem a uma matriz de rotação e translação, representam uma transformação homogênea. Estas podem ser usadas na transformação entre eixos de coordenadas que difiram em translação e rotação. A composição de transformações pode ser efetuada por

$$H_0^2 = H_1^0 H \quad (6.5)$$

no caso em que a segunda transformação, H , é efetuada relativa ao eixo de coordenadas e

$$H_0^2 = H H_1^0 \quad (6.6)$$

no caso em que a segunda transformação, H , é efetuada relativa a um eixo de coordenadas fixo.

6.2 Cinemática direta

O objetivo da cinemática direta é determinar a posição e orientação do punho do robô dados os valores das variáveis das juntas do mesmo. Anteriormente foi descrito que uma cadeia cinemática é composta por um conjunto de segmentos ligados por juntas. Estas podem ser muito simples, como uma junta prismática ou uma junta rotacional, ou um pouco mais complexas, como uma junta esférica. A diferença entre estas prende-se com os graus de liberdade que cada uma possui. Neste trabalho assume-se que todas as juntas possuem apenas um grau de liberdade sem perda de generalidade, visto que uma junta complexa pode ser decomposta numa sucessão de juntas com um grau de liberdade conectadas por segmentos de comprimento nulo. Para a modelização do robô são convencionadas as seguintes regras:

- Um robô possui n articulações e $n + 1$ segmentos;
- Uma articulação i une os segmentos $i - 1$ e i ;
- Atuar na articulação i faz o segmento i mover-se;

- O segmento 0 é a base, e não se move;
- A cada articulação i está associado um referencial $o_i x_i y_i z_i$.

A convenção Denavit-Hartenberg (DH)

Em 1955, Jacques Denavit e Richard Hartenberg introduziram uma convenção para a definição das matrizes de juntas e segmentos por forma a normalizar os eixos de coordenadas para ligações espaciais. Esta convenção proporciona um procedimento sistemático para executar a análise da cinemática direta, simplificando a mesma de forma considerável. Nesta, cada transformação homogénea H_i é representada como o produto de quatro transformações básicas:

$$H_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (6.7)$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde os parâmetros $\theta_i, a_i, d_i, \alpha_i$ estão associados ao segmento e junta i e representam:

- θ_i : ângulo entre o eixo x atual e o seguinte;
- α_i : ângulo entre o eixo z atual e o seguinte;
- d_i : distância entre o eixo x atual e o seguinte;
- a_i : distância entre o eixo z atual e o seguinte.

Este método define referenciais associados a cada articulação seguindo dois princípios básicos, tal como representado na figura 6.4:

1. O eixo Z_n está na direção do eixo da articulação a_n .
2. O eixo x_n é perpendicular ao plano definido pelos eixos Z_n e Z_{n-1} .

Uma vez que o NAO é um robô humanoide, a análise da cinemática deste pode ser decomposta nas suas várias cadeias distintas, cabeça e os seus quatro membros, unidas por um corpo fixo, o torso. Após ter sido efetuada a análise cinemática do robô NAO foram obtidas as tabelas de parâmetros DH 6.1, 6.2, 6.3, 6.4 e 6.5.

Tabela 6.1: Parâmetros associados à cinemática direta da cabeça segundo a convenção DH

Joint Name	α	a	θ	d
HeadYaw	0.0	0.0	0.0	0.0
HeadPitch	$-\pi/2$	0.0	$-\pi/2$	0.0

Tabela 6.2: Parâmetros associados à cinemática direta do braço esquerdo segundo a convenção DH

Joint Name	α	a	θ	d
LShoulderPitch	$-\pi/2$	0.0	0.0	0.0
LShoulderRoll	$\pi/2$	0.0	$\pi/2$	0.0
LElbowYaw	$-\pi/2$	0.0	0.0	UpperArmLength
LElbowRoll	$-\pi/2$	0.0	0.0	0.0
LWristYaw	$\pi/2$	0.0	0.0	LowerArmLength

Tabela 6.3: Parâmetros associados à cinemática direta do braço direito segundo a convenção DH

Joint Name	α	a	θ	d
RShoulderPitch	$-\pi/2$	0.0	0.0	0.0
RShoulderRoll	$\pi/2$	0.0	$\pi/2$	0.0
RElbowYaw	$\pi/2$	0.0	0.0	UpperArmLength
RElbowRoll	$-\pi/2$	0.0	0.0	0.0
RWristYaw	$\pi/2$	0.0	0.0	LowerArmLength

Tabela 6.4: Parâmetros associados à cinemática direta da perna esquerda segundo a convenção DH

Joint Name	α	a	θ	d
LHipYawPitch	$-3/4 \times \pi$	0.0	$-\pi/2$	0.0
LHipRoll	$-\pi/2$	0.0	$\pi/4$	0.0
LHipPitch	$\pi/2$	0.0	0.0	0.0
LKneePitch	0.0	TighLength	0.0	0.0
LAnklePitch	0.0	TibiaLength	0.0	0.0
LAnkleRoll	$-\pi/2$	0.0	0.0	0.0

Tabela 6.5: Parâmetros associados à cinemática direta da perna direita segundo a convenção DH

Joint Name	α	a	θ	d
RHipYawPitch	$-\pi/4$	0.0	$-\pi/2$	0.0
RHipRoll	$-\pi/2$	0.0	$-\pi/4$	0.0
RHipPitch	$\pi/2$	0.0	0.0	0.0
RKneePitch	0.0	TighLength	0.0	0.0
RAnklePitch	0.0	TibiaLength	0.0	0.0
RAnkleRoll	$-\pi/2$	0.0	0.0	0.0

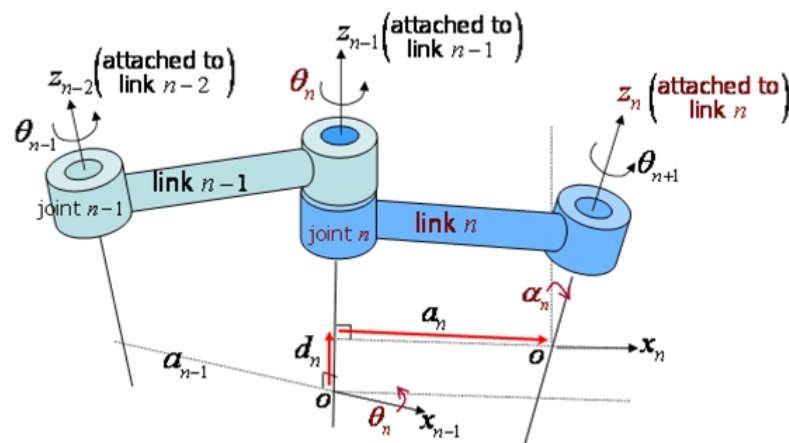


Figura 6.4: Atribuição de eixos segundo a convenção DH[53]

6.3 Cinemática inversa

Enquanto a cinemática direta permite conhecer as coordenadas das articulações no espaço cartesiano de acordo com os respectivos ângulos e comprimentos dos seus segmentos associados, a cinemática inversa permite-nos calcular os valores das coordenadas articulares com base na posição da ponta do objeto. Algumas características associadas à mecânica do nosso objeto, tais como o número de articulações, graus de liberdade e limites de rotação e translação de cada articulação, podem tornar o cálculo complexo e muitas vezes complicado de se obter uma única solução. De facto, múltiplas combinações de ângulos podem permitir ao robô atingir a mesma posição e orientação da extremidade. A determinação da configuração correta das articulações passa portanto pela resolução de um sistema altamente não linear, baseado nas restrições e limitações físicas do robô. O processo de resolução de um problema tradicional de cinemática inversa parte das coordenadas dos pontos inicial e final, dos comprimentos de todos os segmentos envolvidos, e do tipo de articulações, sendo o conhecimento destas variáveis necessário e suficiente para a descoberta de pelo menos uma solução. No entanto, dois fenómenos podem conduzir à existência de infinitas soluções:

- Redundância, quando o número de articulações excede o número de graus de liberdade. A complexidade da cinemática inversa é tanto maior quanto maior for o número de articulações. Seis é o número mínimo de articulações necessário para atingir seis graus de liberdade e, portanto, possibilitar ao robô atingir, dentro do volume de trabalho, qualquer posição e orientação do segmento da extremidade;
- Singularidade, quando várias combinações de uma ou mais articulações sobrepostas colocam a articulação seguinte na mesma posição. Uma singularidade pode ser identificada através da análise do Jacobiano do manipulador, e da determinação do conjunto de valores das articulações que anula o seu determinante.

6.3.1 Braços

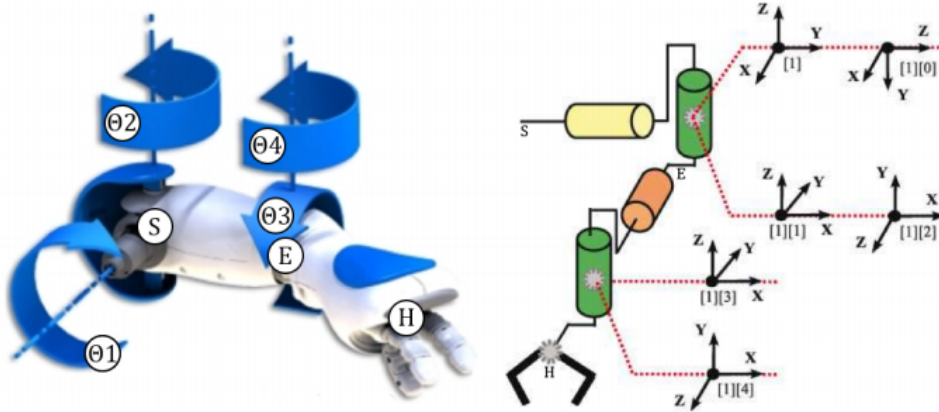


Figura 6.5: Braço do NAO e esquemático das suas articulações[54]

Os ombros do NAO correspondem às articulações que se encontram imediatamente acopladas ao torso, que possui um referencial fixo, e são modelizados pelas duas articulações rotativas correspondentes aos ângulos θ_1 e θ_2 da figura 6.5. Uma vez que os referenciais do torso e do ombro possuem a mesma orientação, a transformação das coordenadas de um ponto entre os referenciais pode ser efetuada por uma simples translação.

Definindo inicialmente a articulação θ_3 a 0, é possível calcular θ_1 através da altura do ponto desejado H_Z e da distância do ombro à sua projeção $distance$ no plano xy do referencial do ombro do robô através de:

$$\theta_1 = \arcsin(H_Z/distance) \quad (6.8)$$

O ângulo θ_4 , definido como o ângulo Yaw do cotovelo, pode ser obtido através da lei dos cossenos definida como:

$$a^2 = b^2 + c^2 - 2.b.c.\cos\hat{A} \quad (6.9)$$

Aplicando a equação 6.9 ao triângulo formado pelo ombro, braço e cotovelo representado na figura 6.6, θ_4 é dado por:

$$\theta_4 = \begin{cases} \arccos\left(\frac{distance^2 - armLength^2 - forearmLength^2}{-2. forearmLength. armLength}\right) - \pi & \text{se braço esquerdo} \\ -\arccos\left(\frac{distance^2 - armLength^2 - forearmLength^2}{-2. forearmLength. armLength}\right) - \pi & \text{se braço direito} \end{cases} \quad (6.10)$$

Por fim, o ângulo θ_2 pode ser obtido pela soma do ângulo θ_{2_1} formado entre o segmento *head direction* e *distance* da figura 6.6 e o ângulo θ_{2_2} formado entre este e o segmento *armLength*. O

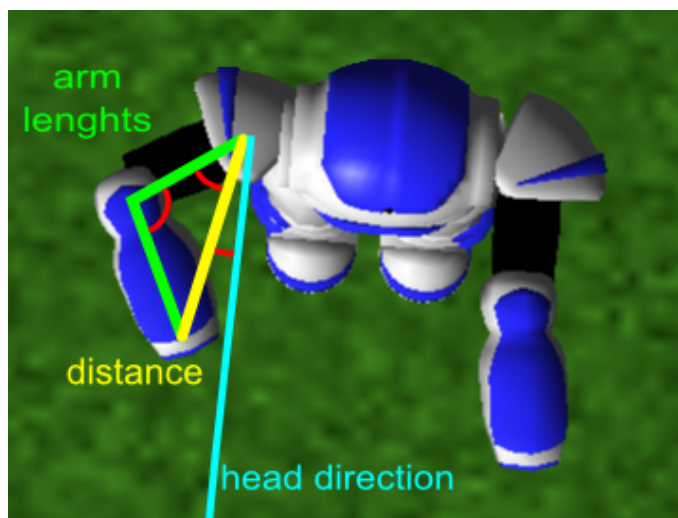


Figura 6.6: Dedução da cinemática inversa

primeiro ângulo pode ser calculado por:

$$\theta_{21} = \begin{cases} \arctan(H_y/H_x) & \text{se braço esquerdo} \\ -\arctan(H_y/H_x) & \text{se braço direito} \end{cases} \quad (6.11)$$

O segundo ângulo é obtido através da aplicação da lei dos cossenos, representada pela equação 6.9, ao triângulo formado pelo ombro, braço e cotovelo representado na figura 6.6:

$$\theta_{22} = \arccos\left(\frac{\text{forearmLength}^2 - \text{armLength}^2 - \text{distance}^2}{-2 \cdot \text{distance} \cdot \text{armLength}}\right) \quad (6.12)$$

6.3.2 Pernas

O cálculo da cinemática inversa das pernas do robô NAO são uma componente central para o controlo do robô dado que qualquer módulo de locomoção e posicionamento dinâmico necessita do mesmo. A cinemática inversa é uma ferramenta muito útil para a criação de movimentos, trajetórias e a colocação do tornozelo do robô em pontos precisos. No entanto, o cálculo da cinemática inversa analiticamente não é simples uma vez que as juntas *yaw* da bacia estão rodadas de 45°. O ponto desejado do pé é dado através de uma matriz que representa uma transformação homogénea com a posição e orientação do pé em relação à bacia do robô [11] [12].

Dado que a articulação situada na anca do robô se encontra rodada de 45° em torno do eixo *x* é necessário calcular a posição em relação a este referencial, *AncaOrtogonal*, obtendo-se assim a transformação homogénea:

$$H_{Pe}^{AncaOrtogonal} = Rot_x\left(\frac{\pi}{4}\right) \cdot H_{Pe}^{Anca} \quad (6.13)$$

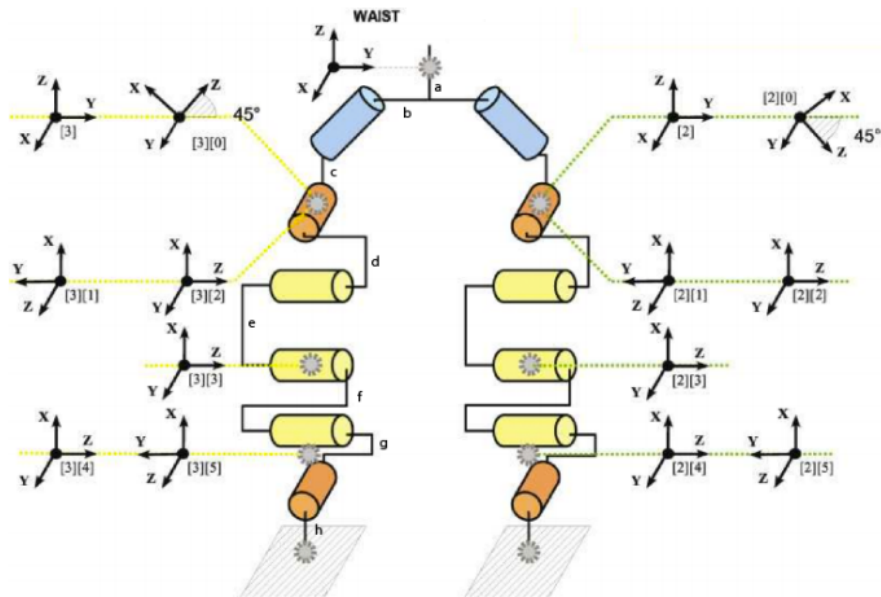


Figura 6.7: Articulações das pernas do NAO

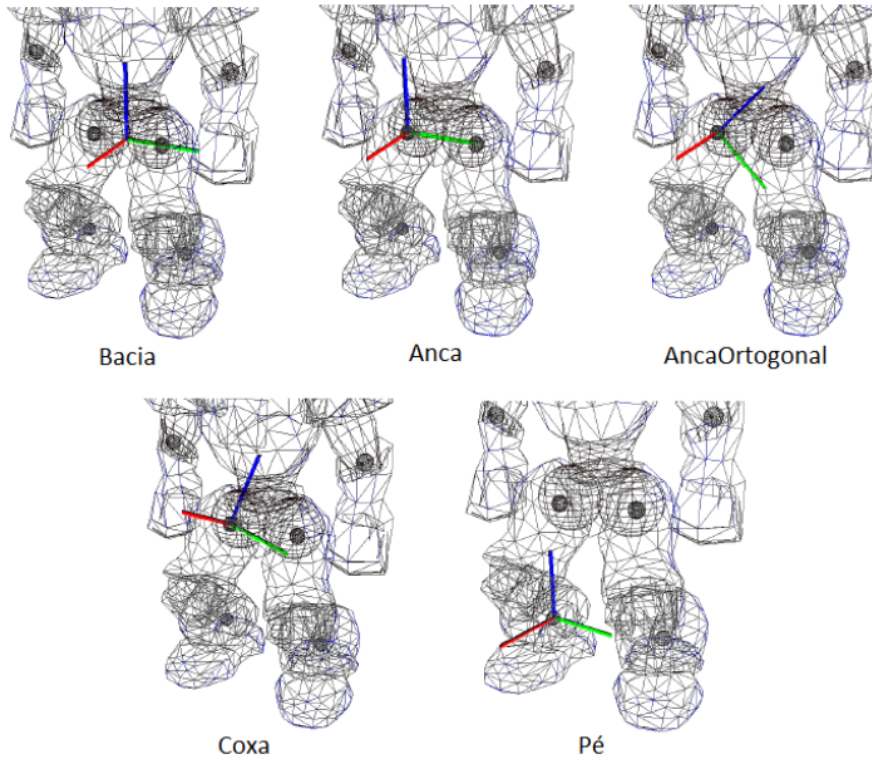


Figura 6.8: Referenciais utilizados no cálculo da cinemática inversa das pernas do NAO

Como é possível observar na figura 6.8, o joelho e os membros da perna formam um triângulo onde todos os lados são conhecidos. Um dos lados, l_{trans} , é dado pelo módulo do vetor de translação da matriz $H_{Pe}^{AncaOrtogonal}$, os outros dois, (l_{coxa} e l_{tibia}), são parâmetros físicos do robô NAO. Usando o teorema dos cossenos é possível determinar os ângulos do triângulo. O ângulo γ entre l_{coxa} e l_{tibia} é dado pela equação:

$$c^2 = a^2 + b^2 - 2.a.b.\cos \gamma \quad (6.14)$$

$$\gamma = \arccos \frac{l_{coxa}^2 + l_{tibia}^2 - l_{trans}^2}{2.l_{coxa}.l_{tibia}} \quad (6.15)$$

Uma vez que o ângulo do joelho δ_{joelho} é zero quando a perna está esticada, este é definido através de:

$$\delta_{joelho} = \pi - \gamma \quad (6.16)$$

O ângulo oposto a l_{coxa} também tem que ser calculado visto corresponder ao ângulo pitch do pé:

$$\delta_{ppitch1} = \arccos \frac{l_{tibia}^2 + l_{trans}^2 - l_{coxa}^2}{2.l_{tibia}.l_{trans}} \quad (6.17)$$

Se combinarmos o triângulo e a articulação do pé veremos que formam uma espécie de unidade pan-tilt¹. O ângulo pitch e roll dessa articulação pode ser calculado usando atan2:

$$\delta_{ppitch2} = \text{atan2}(x, \sqrt{y^2 + z^2}) \quad (6.18)$$

$$\delta_{peRoll} = \text{atan2}(y, z) \quad (6.19)$$

onde x, y, z são as componentes do vetor translação da matriz $H_{Pe}^{AncaOrtogonal}$. O ângulo pitch do pé é dado pela soma das suas duas partes constituintes:

$$\delta_{pePitch} = \delta_{ppitch1} + \delta_{ppitch2} \quad (6.20)$$

Com os ângulos do joelho e do pé calculados podemos calcular os ângulos correspondentes à articulação situada na anca. Para calcular estes ângulos adicionamos um referencial auxiliar (chamado de Coxa na figura 6.8) e extraímos esses ângulos diretamente da matriz de transformação

¹Sistema usado, por exemplo, pelas câmaras de vigilância.

homogénea $H_{AncaOrtogonal}^{Coxa}$, que pode ser calculada por:

$$H_{AncaOrtogonal}^{Coxa} = (H_{Coxa}^{Pe})^{-1} \cdot H_{AncaOrtogonal}^{Pe} \quad (6.21)$$

onde

$$H_{AncaOrtogonal}^{Pe} = (H_{Pe}^{AncaOrtogonal})^{-1} \quad (6.22)$$

e a matriz H_{Coxa}^{Pe} pode ser calculada seguindo a cadeia do pé até à coxa:

$$H_{Coxa}^{Pe} = Rot_x(\delta_{peRoll}) \cdot Rot_y(\delta_{pePitch}) \cdot Trans_z(l_{tibia}) \cdot Rot_y(\delta_{joelho}) \cdot Trans_z(l_{coxa}) \quad (6.23)$$

Para entendermos o cálculo dos ângulos da articulação da anca a matriz $H_{AncaOrtogonal}^{Coxa}$ é representada em função da rotação em torno dos diferentes eixos (a matriz está numa notação abreviada, exemplo, c_x significa $\cos\delta_x$).

$$H_{AncaOrtogonal}^{Coxa} = Rot_z(\delta_z) \cdot Rot_x(\delta_x) \cdot Rot_y(\delta_y) = \begin{bmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z \\ c_z s_x s_y + c_y s_z & c_x c_z & -c_y c_z s_x + s_y s_z \\ -c_x s_y & s_x & c_x c_y \end{bmatrix} \quad (6.24)$$

de onde concluímos que:

$$\delta_x = \arcsin(r_{21})^2 \quad (6.25)$$

O ângulo δ_z retiramos de :

$$\frac{-r_{01}}{r_{11}} = \tan\delta_z \quad (6.26)$$

$$\delta_{ancaYaw} = \delta_z = \text{atan2}(-r_{01}, r_{11}) \quad (6.27)$$

e seguindo o mesmo raciocínio:

$$\delta_{ancaPitch} = \delta_y = \text{atan2}(-r_{20}, r_{22}) \quad (6.28)$$

Só nos falta compensar a rotação de 45 graus referente ao referencial AncaOrtogonal:

$$\delta_{ancaRoll} = \delta_x - \frac{\pi}{4} = \arcsin(r_{21}) - \frac{\pi}{4} \quad (6.29)$$

6.4 Singularidades

Qualquer que seja a posição dos braços do robô existe uma e uma só combinação das quatro articulações, à exceção de dois casos[54]. De facto, $\theta_2 = \pm 90^\circ$ faz com que o cotovelo atinja sempre a mesma posição, independentemente do valor aplicado em θ_1 . Quando $\theta_4 = 0^\circ$, uma singularidade surge por se estar a atingir um ponto limite do volume de trabalho. A matriz Jacobiana

²O primeiro índice representa a linha, e o segundo a coluna da matriz. Os índices começam em zero.

consiste numa representação matricial de 6 linhas e N colunas, em que N é o número articulações de um dado manipulador. Esta matriz é composta pelas derivadas parciais das equações de cinemática, em ordem à variável a que corresponde cada articulação, e descreve a relação entre as velocidades linear e angular da extremidade e as velocidades lineares e angulares de cada uma das articulações, e que são conhecidas. Destas seis linhas, as três primeiras refletem a relação entre as velocidades lineares nos três eixos, enquanto as restantes refletem a relação entre as velocidades angulares. Entre os vários aspetos em que esta matriz é útil, ela permite ainda determinar combinações de articulações que, quando aplicadas no manipulador, originam singularidades (como os dois casos acima descritos). Na matriz Jacobiana, uma singularidade é detetada quando uma combinação particular de valores reduz a característica da matriz, e estes pontos são caracterizados por geralmente neles existir uma infinidade de soluções.

A equação 6.30, abaixo determinada, consiste no Jacobiano do braço da figura 6.5. Nessa mesma figura, a representação à direita explicita a configuração obtida quando é aplicado o valor de 0° a cada articulação, sendo a base da matriz abaixo calculada. Foi também tida em consideração a representação dos referenciais associados a cada articulação segundo o método D-H. Por outro lado, a matriz Jacobiana de manipuladores com menos de 6 articulações não é quadrada, tornando impossível determinar as combinações singulares através do cálculo do seu determinante.

$$\begin{bmatrix} J_V \\ J_\omega \end{bmatrix} = \begin{bmatrix} J_{v1} & J_{v2} & J_{v3} & J_{v4} \\ J_{\omega1} & J_{\omega2} & J_{\omega3} & J_{\omega4} \end{bmatrix} \quad (6.30)$$

Genericamente, uma singularidade pode ser verificada através do Jacobiano, quando uma dada combinação de valores implica a existência de colunas dependentes, tornando o sistema indeterminado. Para qualquer manipulador, independentemente do número e tipo de articulações que o constituem, é possível encontrar tais combinações recorrendo à matriz pseudo inversa do Jacobiano ($J^+ = J^T (JJ^T)^{-1}$), e verificando para que condições o determinante de JJ^T se anula [52].

O cálculo do Jacobiano foi assim feito recorrendo à ferramenta MatLab. Dado o tamanho das matrizes calculadas ser demasiado grande, assim como as equações para o determinante de JJ^T , resultado da presença de quatro articulações rotativas, estes resultados não foram incluídos nesta secção. Todavia, substituindo na equação $\det(JJ^T)$ o valor $\theta_4 = 0^\circ$, este determinante anula-se, confirmando a existência de uma singularidade neste ponto. Da mesma forma, substituir $\theta_2 = \pm 90^\circ$ anula igualmente este determinante e corresponde, portanto, à segunda singularidade deste manipulador. Efetivamente, colocar $\theta_2 = \pm 90^\circ$ corresponde à abertura dos braços para os lados, paralelamente ao chão, alinhando as articulações associadas a θ_1 e θ_2 , e originando a perda de independência de colunas do Jacobiano deste manipulador e, portanto, a diminuição da sua característica. Assim, foram adicionadas as condições 6.31 e 6.32 com o objetivo de eliminar a instabilidade dos ângulos θ_1 e θ_3 . Note-se que, embora $\theta_2 = -90^\circ$ origine igualmente uma singularidade, esta não foi considerada, uma vez que esta articulação do robô está fisicamente limitada,

sendo impossível atingir este valor.

$$\theta_1 = \begin{cases} \theta_1 & \text{se } |\theta_2 - 90^\circ| > 5^\circ \\ 0^\circ & \text{se } |\theta_2 - 90^\circ| \leq 5^\circ \end{cases} \quad (6.31)$$

$$\theta_3 = \begin{cases} \theta_1 & \text{se } |\theta_4| > 5^\circ \\ 0^\circ & \text{se } |\theta_4| \leq 5^\circ \end{cases} \quad (6.32)$$

6.5 Conclusão

Neste capítulo foi apresentada a análise da cinemática direta e inversa do robô que permitem calcular a posição das extremidades do robô de acordo com os ângulos das juntas do robô e o inverso, respectivamente. Esta análise é essencial para qualquer módulo que implique a movimentação de um robô, tal como acontece com o controlador do centro de massa desenvolvido.

Capítulo 7

Controlador do Centro de Massa

A teoria de controlo é um ramo da engenharia com o objetivo de controlar um sistema dinâmico através das suas entradas. A aplicação destes controladores estende-se a imensas áreas, entre as quais a robótica, onde são muito utilizados no controlo dos servo motores e em módulos de estabilidade. Em engenharia, um sistema de controlo tem pelo menos dois módulos principais, o próprio controlador e o sistema a ser controlado[55]. Para que o robô se mantenha estável é imprescindível um módulo de estabilidade que utilize um controlador dinâmico para atingir os objetivos de controlo especificados. O controlador desenvolvido, cujo objetivo é manter o robô estável em meios de locomoção estática, utiliza o posicionamento do centro de massa para garantir que o robô se mantém estável durante movimentos precisos.

7.1 Controlo de malha aberta

Um método de controlo em malha aberta não possui nenhuma noção do estado do sistema, pelo que não sabe se a saída do sistema está ou não de acordo com a pretendida. Assim, utilizando este tipo de controladores não existe forma de corrigir uma diferença entre o valor de saída esperado e o real (chamado de erro), uma vez que nenhuma informação está disponível em relação a esta diferença. A figura 7.1 mostra a arquitetura genérica de um controlador em malha aberta para um valor discreto no tempo n onde $x[n]$ representa a entrada, $u[n]$ representa a saída do controlador e $y[n]$ representa a saída do sistema controlado.

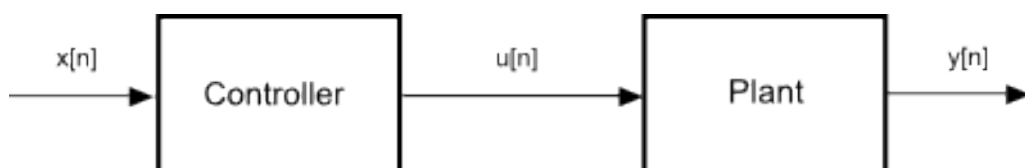


Figura 7.1: Arquitetura típica de controlador em malha aberta[51]

Para uma utilização correta deste tipo de controlador é necessário um conhecimento muito preciso do sistema a controlar. O controlo em malha aberta é tipicamente utilizado em sistemas cuja relação entrada-saída pode ser modelada por uma equação matemática. Se o comportamento do sistema é previsível, o controlo em malha-aberta é suficiente para controlar o sistema. Por outro lado, o sistema pode necessitar de ser realimentado. Sendo K_c o ganho proporcional do controlador e K_p o ganho proporcional do sistema, a saída do sistema será dada por:

$$y[n] = K_p u[n] = K_c (K_p x[n]) \quad (7.1)$$

Assumindo que o sistema corresponde a um sistema linear, que poderá não ser o caso, qualquer perturbação externa produzirá um erro no sistema e o controlador não estará consciente da perturbação pelo que não corrigirá o erro. Se o modelo matemático da planta não é preciso, as consequências podem ser catastróficas, dependendo da necessidade do sistema. Para sistemas não lineares, este problema torna-se ainda mais complexo uma vez que os sistemas não lineares são difíceis de modelizar e as equações não lineares são difíceis de resolver.

7.1.1 Vantagens

As principais vantagens do controlo em malha aberta são:

- Simples de compreender;
- Simples de implementar;
- Implementação e manutenção de baixo custo.

7.1.2 Desvantagens

As principais desvantagens do controlo em malha aberta são:

- É necessário um modelo preciso do sistema;
- O sistema não pode ser realimentado;
- É sensível a perturbações externas.

7.2 Controlo de malha fechada

Como alternativa para o controlo em malha aberta pode ser utilizado um controlo em malha fechada. Tipicamente não é possível obter um modelo preciso do sistema a controlar. Nesse caso é melhor utilizar um controlador capaz de se adaptar à entrada do sistema, tendo em consideração não só a saída desejada, mas também a saída efetiva do sistema. Quando a saída do sistema é realimentada, fazendo parte das suas entradas, temos o chamado controlo em malha fechada. A figura 7.2 representa a arquitetura genérica de um sistema de controlo em malha fechada.

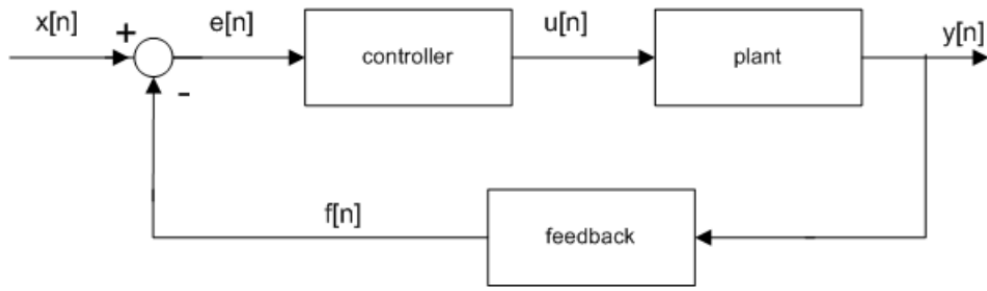


Figura 7.2: Arquitetura típica de controlador em malha fechada[51]

Para um valor discreto no tempo n , $x[n]$ representa a entrada, $u[n]$ representa a saída do controlador e $y[n]$ representa a saída do sistema controlado. Adicionalmente, $f[n]$ representa a informação de realimentação disponibilizada por um sensor.

Assumindo um controlo de realimentação negativa, a informação será subtraída da saída desejada para gerar o chamado erro, que é representado na figura por $e[n]$. Este erro, que representa a diferença entre a saída desejada e a saída efetiva, será a nova entrada do controlador. Sejam K_c , K_p e K_f os ganhos proporcionais do controlador, sistema e sensor respetivamente, a saída do sistema será dada em função do seguinte sistema de equações [56]:

$$\begin{cases} y[n] = K_p u[n] \\ u[n] = K_c (x[n] - f[n]) \\ f[n] = K_f y[n] \end{cases} \quad (7.2)$$

Resolvendo este sistema de equações o sistema é dado por:

$$y[n] = \frac{K_p K_c}{1 + K_p K_c K_f} x[n] \quad (7.3)$$

Dividindo o numerador e o denominador da equação por $K_p K_c$ temos:

$$y[n] = \frac{1}{\frac{1}{K_p K_c} + K_f} x[n] \quad (7.4)$$

Assumindo que o ganho do controlador, K_c , é suficientemente grande para descartar o primeiro termo do denominador, uma simplificação da equação será:

$$y[n] \simeq \frac{1}{K_f} x[n] \quad (7.5)$$

A equação 7.5 significa que, se o ganho do controlador for suficientemente grande, a saída torna-se menos sensível a perturbações externas. Visto que os sistemas reais são sempre afetados

por algum ruído, com principal destaque na robótica, este tipo de controlador tem várias vantagens em relação ao controlador em malha aberta descrito na secção anterior.

7.2.1 Vantagens

As principais vantagens de um controlo em malha fechada são:

- O sistema é realimentado para correção de erros;
- Menos sensível a perturbações externas;
- Não existe necessidade de um modelo preciso do sistema.

7.2.2 Desvantagens

As principais desvantagens de um controlo em malha fechada são:

- A arquitetura é mais complexa do que em malha aberta;
- Necessita de sensorização do sistema;
- A sua estrutura inerentemente reativa não é adequada para atuar em antecipação.

7.2.3 Controlo PID

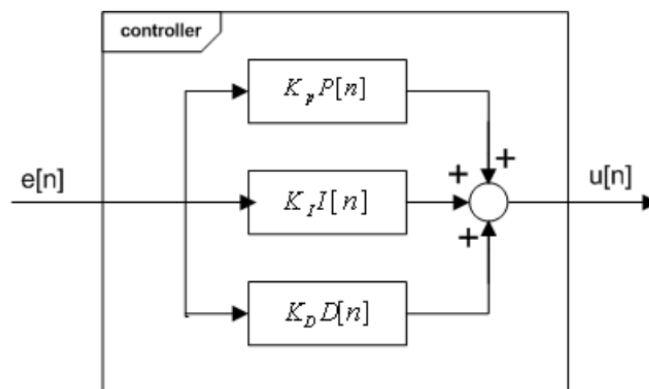


Figura 7.3: Controlador PID[51]

PID significa Proporcional-Integral-Derivativo e é uma implementação particular de um controlador em malha fechada, vastamente utilizado em processos industriais e na robótica [57] [58] [59] [55]. Também é conhecido como o controlador de três termos uma vez que utiliza três ganhos para gerar a saída. A saída do controlador como sinal contínuo é descrita por:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (7.6)$$

O equivalente discreto da equação é dado pelo seguinte sistema de equações:

$$\begin{cases} u[n] = K_P P[n] + K_I I[n] + K_D D[n] \\ P[n] = e[n] \\ I[n] = I[n-1] + T e[n] \\ I[n] = \frac{e[n] - e[n-1]}{T} \end{cases} \quad (7.7)$$

Onde T é o intervalo entre amostragens discretas. A arquitetura genérica é a mesma da figura 7.2. O que faz este controlo especial é a sua configuração interna do bloco controlador. A figura 7.3 representa uma visão detalhada do bloco do controlador no caso de um controlador PID.

O controlador pode aparecer em diferentes variantes: controlador Proporcional (P) (o mais simples, utiliza apenas o termo proporcional), Proporcional-Integral (PI), Proporcional-Derivativo (PD) e finalmente controladores PID (todos os termos são usados).

Tomando por exemplo uma junta de um humanoide, se se quiser mover a junta em 1 radiano, a resposta da junta não será imediata, mas haverá algum tempo de subida. O objetivo do controlador é aumentar a subida o mais rápido possível sem produzir instabilidade. A figura 7.4 representa quatro situações da junta controlada.

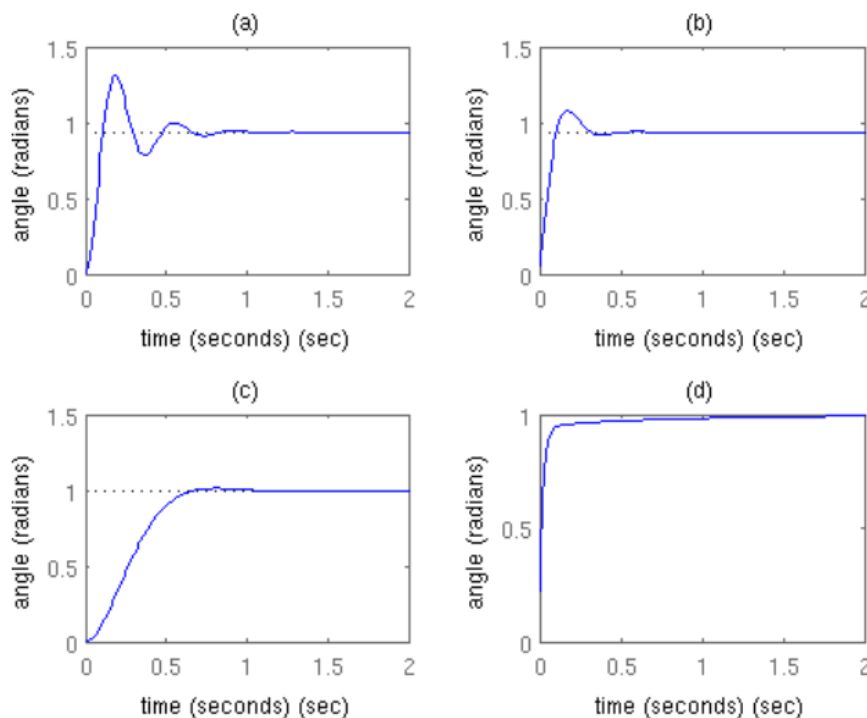


Figura 7.4: Resposta de controlador de juntas PID (a)controlador P (b)controlador PD (c)controlador PI (d)controlador PID[51]

A figura 7.4 mostra algumas consequências dos diferentes termos. Um ganho proporcional

elevado pode produzir uma sobre-elongação. O termo integral reduz o erro em regime estacionário mas, tal como o termo proporcional, reduz o tempo de subida e pode produzir uma sobre-elongação, pelo que pode ser necessário diminuir o ganho proporcional para somar o termo integral. O termo derivativo reduz a sobre-elongação e tem um pequeno efeito no erro em regime estacionário. Existem várias aproximações para definir os ganhos do controlador PID embora a mais popular seja o método de Ziegler-Nichols [60].

7.3 O controlador

O controlador desenvolvido permite a utilização de várias arquiteturas para o controlo da projeção do centro de massa no solo através do controlo do tipo PID. Todos os métodos de controlo desenvolvidos foram ajustados de acordo com o método heurístico de Ziegler Nichols.

7.3.1 Braços

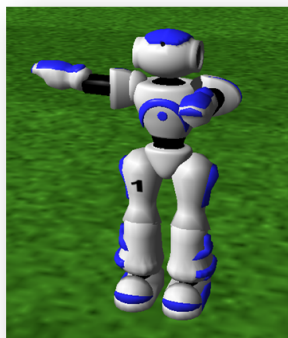


Figura 7.5: Projeção do braço do robô segundo o eixo y

Dado que os braços do robô durante uma partida de futebol são raramente utilizados, inicialmente foi desenvolvido um controlador que utiliza os braços para colocar o centro de massa do corpo numa determinada posição definida, visto que o volume de trabalho dos braços do robô corresponde a uma semiesfera que assenta no plano frontal do robô. O controlador utilizado, do tipo PID, recebe o erro da posição do centro de massa em coordenadas (x,y) referentes ao torso do robô e calcula a posição desejada para a colocação dos braços do robô em (x,y) . Dado que para efeitos de estabilidade, a posição do braço deverá ser o mais próxima possível do plano de suporte, a coordenada z dos braços é calculada através do comprimento do braço l_{arm} e pela posição (x,y) deste, por:

$$z = -\sqrt{l_{arm}^2 + x^2 + y^2} \quad (7.8)$$

As coordenadas obtidas após a equação 7.8 são aplicadas ao módulo de cinemática inversa desenvolvido para os braços que posteriormente ajusta a posição dos mesmos de acordo com a posição do centro de massa do robô.

Aplicando a pose representada na figura 7.5 a posição do centro de massa do robô teve a resposta apresentada na figura 7.6.

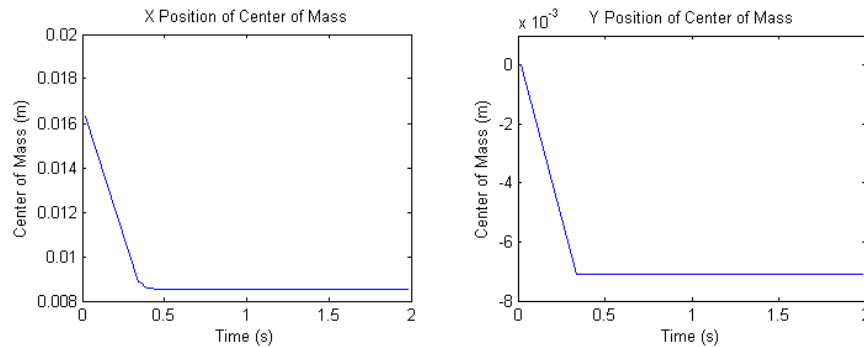


Figura 7.6: Resposta do CoM do robô em relação à projeção do braço do robô segundo o eixo x e y

Este método revelou ser pouco eficaz no controlo da posição do centro de massa do robô uma vez que a massa dos braços é demasiado pequena em relação à massa total do robô pelo que a manipulabilidade do centro de massa de acordo com este método é muito baixa. Na figura 7.6 pode verificar-se que um braço consegue apenas alterar a posição do centro de massa do robô em cerca de 9mm, quer na sua projeção no eixo x , quer na sua projeção no eixo y .

7.3.2 Pernas

Para controlarmos a posição do centro de massa do robô há que ter em conta a fase de locomoção em que o mesmo se encontra. A figura 7.7 representa as fases de locomoção humanoide.

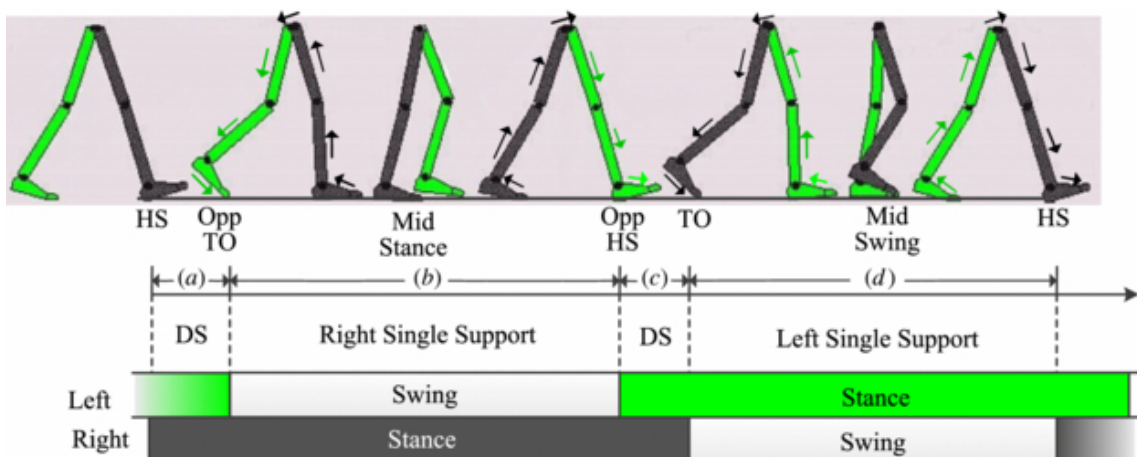


Figura 7.7: Fases da locomoção

Tal como se pode observar na figura 7.7, as fases *a* e *c* correspondem às fases de duplo suporte, em que ambos os pés se encontram em contacto com o solo, e as fases *b* e *d* correspondem às fases de suporte único nas quais apenas um dos pés se encontra em contacto com o solo.

7.3.2.1 Suporte duplo

Na fase de suporte duplo são utilizadas as juntas *pitch* e *roll* do pé e da anca. Deste modo, aplicando um ângulo igual entre cada par de juntas anca e pé com sinal oposto, o corpo do robô é transladado mantendo-se paralelo ao chão. Desta modo é possível controlar a posição em *x* do robô através do ângulo aplicado às juntas *pitch* do pé e da bacia do robô, e é possível controlar a posição *y* do robô através do ângulo aplicado às juntas *roll* do pé e da bacia do robô. Este método de controlo está esquematizado na figura 7.8.

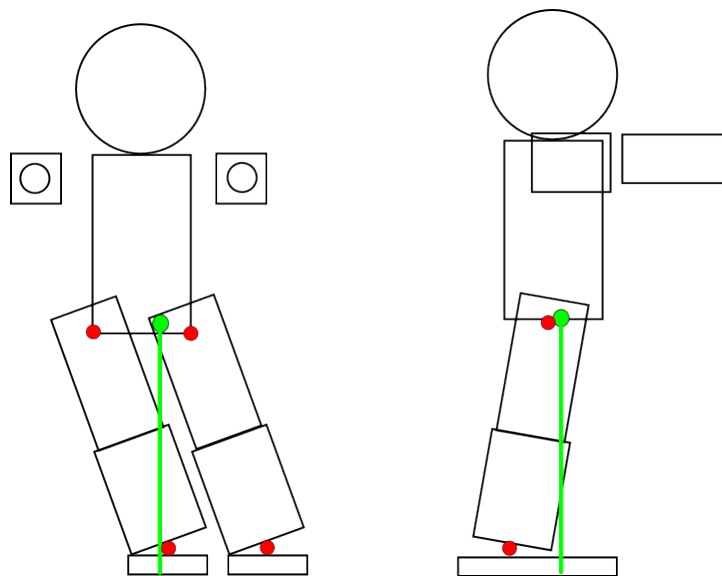


Figura 7.8: Controlo do centro de massa do robô através das juntas *pitch* e *roll* da anca e dos pés em fase de duplo suporte

A figura 7.9 apresenta a resposta do controlador após a aplicação de uma entrada em onda quadrada que varia entre a posição (0,0) e (-0.05 , -0.02), em metros, e com um período de 6 segundos. Através da análise da mesma, é possível verificar que o sistema apresenta uma resposta subamortecida, com um tempo de subida e de assentamento um pouco elevados. Esta característica demonstra que o controlador é preciso e ideal para situações que requerem um posicionamento exato do centro de massa do robô.

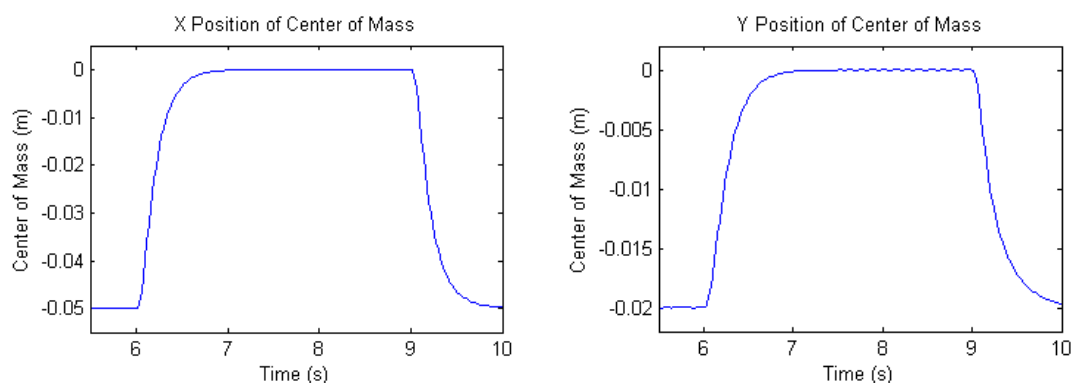


Figura 7.9: Posição do centro de massa do robô controlado em fase de duplo suporte

7.3.2.2 Suporte único

Na fase de suporte único podem ser utilizadas apenas as juntas da bacia para controlar o centro de massa do robô, o que provoca uma alteração na orientação do torso do robô, ou podem ser utilizadas as juntas dos pés do robô juntamente com a bacia de modo a que a orientação do torso se mantenha igual.

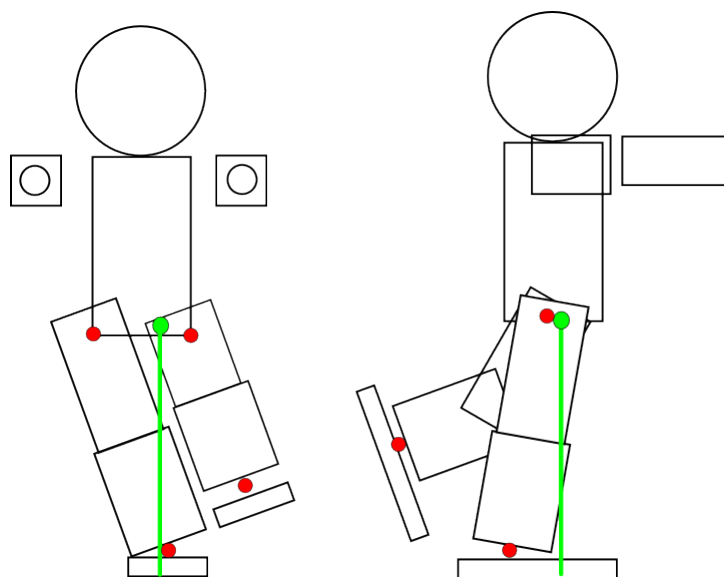


Figura 7.10: Controlo do centro de massa do robô através das juntas pitch e roll da anca e dos pés em fase de suporte único

Como se pode observar na figura 7.10, através da aplicação de um ângulo determinado pelo controlador PID ao pé e à anca do robô, com este último tendo sinal oposto ao primeiro, é possível deslocar o centro de massa do robô mantendo a orientação do tronco.

A figura 7.11 apresenta a resposta do controlador após a aplicação de uma entrada em onda quadrada que varia entre a posição (0,0) e (-0.05 , -0.02), em metros, e com um período de 6

segundos. Através da análise da mesma, é possível verificar que o sistema apresenta uma resposta subamortecida, com um tempo de subida e de assentamento um pouco elevados. Esta característica, tal como esperado semelhante à da figura 7.9, demonstra que o controlador é preciso e ideal para situações que requerem um posicionamento exato do centro de massa do robô.

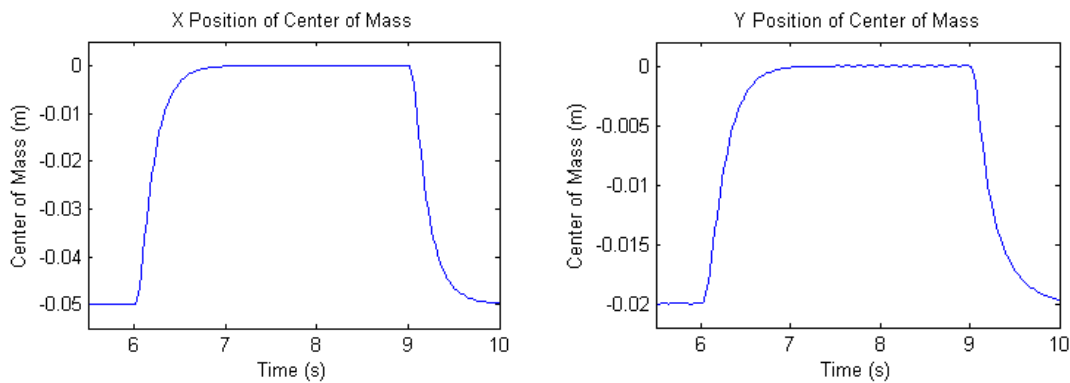


Figura 7.11: Posição do centro de massa do robô controlado através das juntas da bacia e do pé em fase de suporte único

A sequência de controlo do centro de massa do controlador anteriormente referido está representada na figura 7.12.

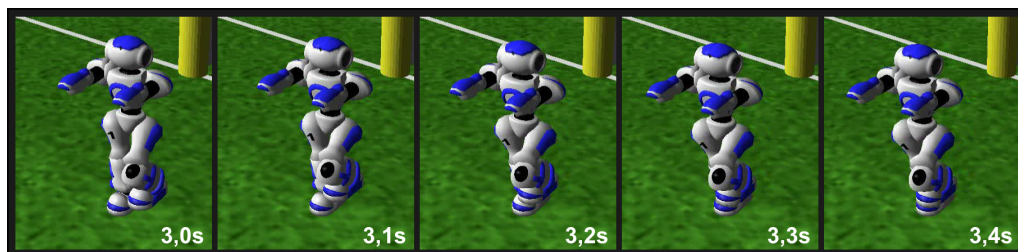


Figura 7.12: Sequência de controlo do centro de massa do robô controlado através das juntas da bacia e do pé em fase de suporte único

Como se pode observar na figura 7.13, através da aplicação de um ângulo determinado pelo controlador PID à anca do robô é possível deslocar o centro de massa do robô. Desta forma, as restantes articulações do robô estão disponíveis para a execução de qualquer tarefa que o robô necessite.

A figura 7.14 apresenta a resposta do controlador após a aplicação de uma entrada em onda quadrada que varia entre a posição (0,0) e (-0.05, -0.02), em metros, e com um período de 6 segundos. Através da análise da mesma, é possível verificar que o sistema apresenta uma resposta subamortecida, com um tempo de subida e de assentamento baixos. Esta característica demonstra que o controlador possui uma resposta rápida e precisa, ideal para situações que requerem um posicionamento rápido do centro de massa do robô como em situações de equilíbrio.

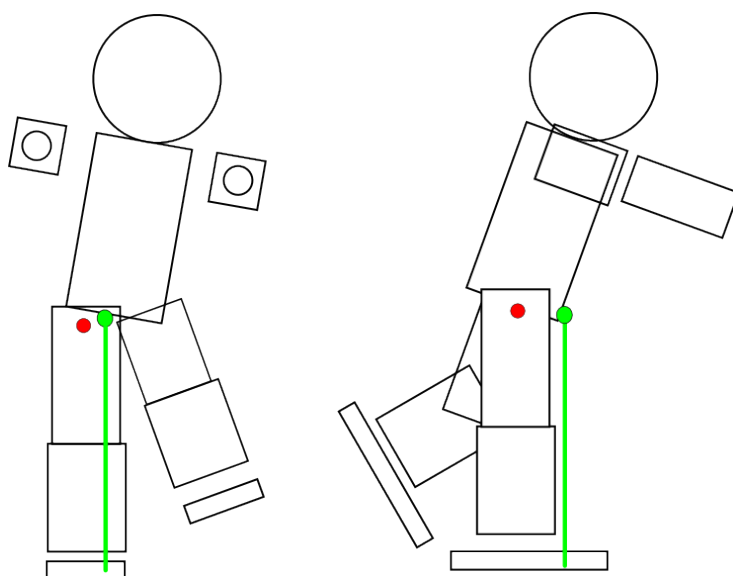


Figura 7.13: Controle do centro de massa do robô através das juntas pitch e roll da anca em fase de suporte único

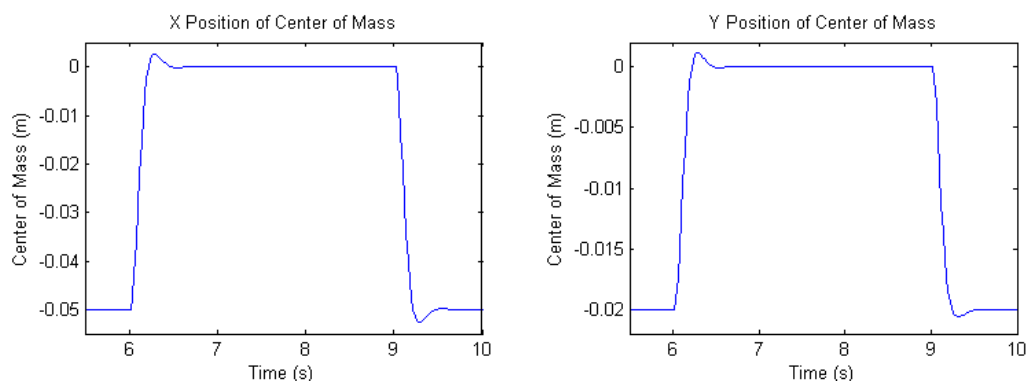


Figura 7.14: Posição do centro de massa do robô controlado através das juntas da bacia em fase de suporte único

A sequência de controle do centro de massa do controlador anteriormente referido está representada na figura 7.15.

Através da sobreposição da resposta do controlador que utiliza as juntas da bacia e do pé, da figura 7.11, e da resposta do controlador que utiliza apenas as juntas da bacia, presente na figura 7.14, obteve-se o gráfico presente na figura 7.16. Neste é possível denotar que através da utilização do segundo controlador é possível obter-se uma resposta mais rápida que o primeiro, apresentando contudo uma sobre-elongação de cerca de 5%. Assim, o controlador que utiliza as juntas da bacia e do pé deverá ser utilizado em situações na qual a orientação do tronco deverá permanecer constante durante o controle ou em aplicações onde não seja permitida sobre-elongações, como no caso do posicionamento preciso do centro de massa do robô próximo dos extremos do polígono de suporte do mesmo. A utilização de apenas as juntas da bacia por parte do controlador deverá ser utilizada

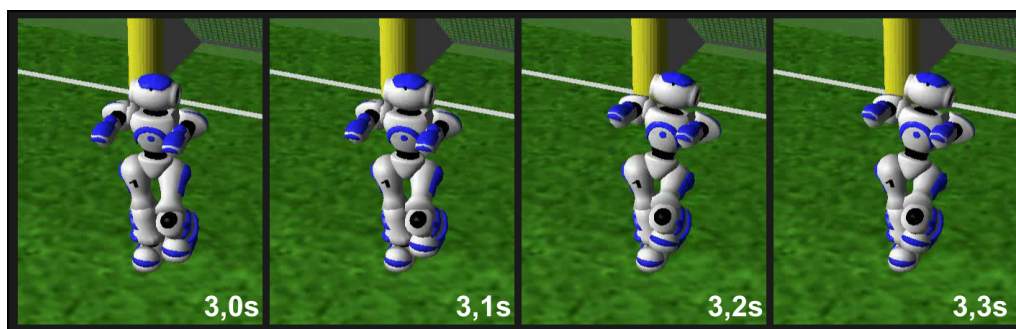


Figura 7.15: Sequência de controlo do centro de massa do robô controlado através das juntas da bacia em fase de suporte único

em situações de estabilização do robô devido à sua resposta rápida, à possibilidade de existência de sobre-elongação e à sua abordagem minimalista que permite a utilização das restantes juntas do robô para a execução de um determinado comportamento.

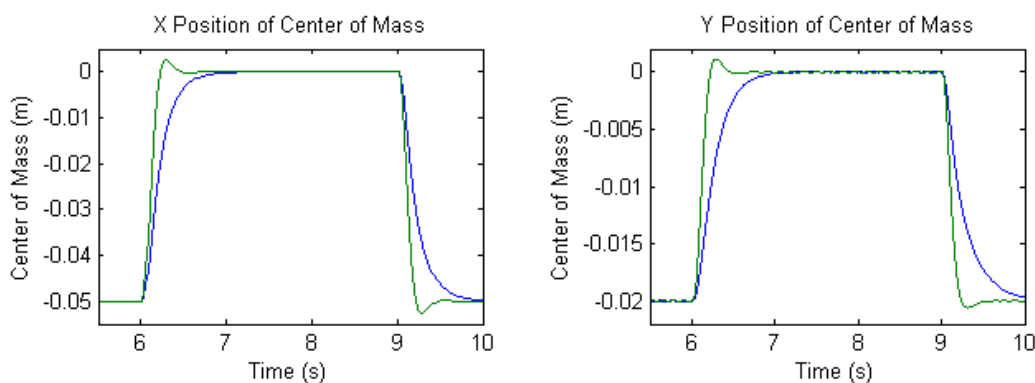


Figura 7.16: Comparação da resposta entre o controlador que utiliza as juntas da bacia e do pé(azul) e o controlador que utiliza apenas as juntas da bacia(verde)

7.4 Polígono de suporte

Através da análise da cinemática direta presente no capítulo 6, é possível calcular a posição dos pés do robô em relação ao torso. Através do cálculo do polígono de suporte, e analisando a posição da projeção do centro de massa do robô no solo, é possível saber se o robô está estaticamente estável ou não. Por forma a tornar o método robusto e adaptável, o polígono de suporte foi calculado através da utilização do algoritmo de Melkman que é capaz de calcular o polígono convexo de uma cadeia poligonal simples em tempo linear. A ideia chave deste algoritmo é manter uma versão atualizada do polígono convexo, e para cada novo ponto encontrado na entrada da cadeia poligonal é testado se o polígono convexo necessita de ser atualizado.

A saída a cada passo do algoritmo é um polígono convexo com orientação anti-horária, cujo primeiro e último elemento são o mesmo ponto, que corresponde ao último ponto a ser adicionado

ao polígono. Assim, o algoritmo começa pelo envio dos primeiros três pontos com a devida orientação.

A cada passo do algoritmo testamos se o novo ponto está fora do polígono convexo. Caso esteja, o polígono convexo tem de ser atualizado com o novo ponto, caso contrário, o novo ponto é descartado e a saída permanece a mesma.

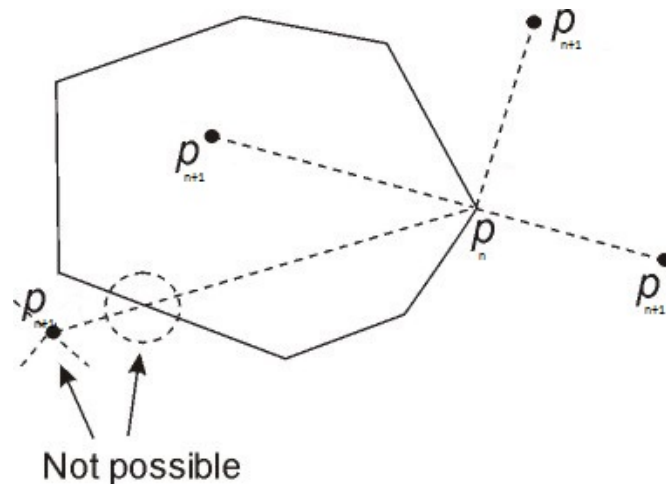


Figura 7.17: Cálculo do polígono convexo

A figura 7.17 mostra um exemplo da saída do método, mostrando os dois primeiros pontos bem como várias possibilidades para a localização do novo ponto (P_{n+1}). Uma vez que o algoritmo deve funcionar em tempo linear, o teste para determinar se o novo ponto está dentro do polígono deverá ser efetuado em tempo constante. Assim, Melkman apresentou um método de determinar em tempo constante se um ponto está dentro do polígono convexo atual:

Se o novo ponto se encontra à direita do segmento orientado $P_2 - P_1$ e à esquerda do segmento $P_{n-1} - P_n$, então é certo de que o ponto está dentro do polígono convexo, uma vez que a linha ou polígono que vai de P_n para P_{n+1} (que faz parte do polígono) teria de intersestar o polígono convexo para que o ponto P_{n+1} estivesse de fora. Se ele intersestar o polígono convexo, então ele intersestará o próprio polígono de entrada. Contudo, uma vez que o polígono de entrada é simples, então sabemos que não existem interseções nos seus cantos.

Assim, testando a orientação para estes dois tripletes de pontos, é possível determinar se o novo ponto deverá ser descartado ou não. Se estiver dentro, é descartado. Se estiver fora, então teremos de o juntar a ambas as extremidades do polígono de saída, e será agora necessário reajustar o polígono de saída removendo qualquer vértice que tenha ficado côncavo após a inserção deste novo ponto.

Todas estas novas atualizações envolvem apenas testes de orientação de tripletes de pontos, uma vez que o polígono de saída estará em sentido anti-horário, o que significa que uma curva à esquerda signifique um vértice convexo, e uma curva à direita representa um vértice côncavo.

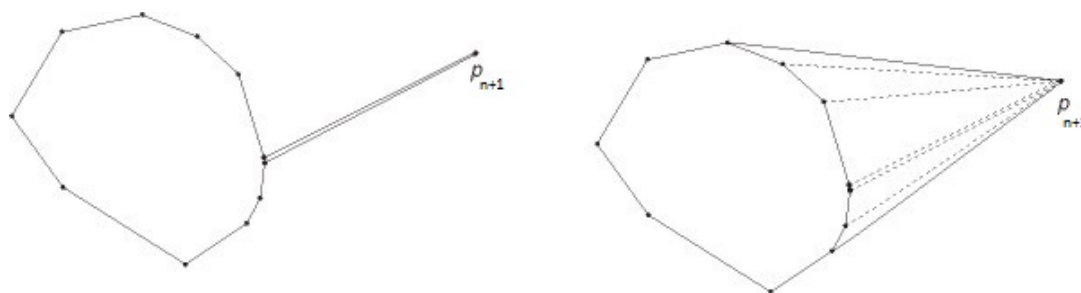


Figura 7.18: Atualização do polígono convexo

A figura 7.18 mostra um exemplo deste processo de atualização. Apesar do facto de um passo (após $O(n)$ inserções) possa envolver $O(n)$ remoções, o algoritmo funciona em tempo linear. Cada ponto do polígono de entrada pode ser inserido no máximo uma vez na saída, e pode ser removido, no máximo uma vez. Assim, temos no máximo $4 \times N$ operações, uma vez que as inserções e remoções podem ser efetuadas em ambas as extremidades. Dado que o teste que verifica se um ponto deverá ser descartado ou não funciona em tempo constante, o algoritmo funciona em tempo real.

Para analisar a estabilidade do robô adiciona-se o ponto correspondente à projeção do centro de massa no solo, ao polígono de suporte e procede-se ao processo de cálculo do polígono convexo anteriormente descrito. Caso o ponto que representa a projeção do centro de massa do robô no solo seja descartado, significa que este se encontra no interior do polígono de suporte e que o robô se encontra estável. Caso contrário o robô não se encontra estaticamente estável.

7.5 Centroide

Para o controlo estável de um robô, a posição ideal para definir como *setpoint* para a posição pretendida do centro de massa do robô é dada pelo centroide do polígono de suporte do mesmo. Para o cálculo do centroide de um polígono é necessário primeiro calcular a área do mesmo.

Considerando um polígono constituído por um conjunto de segmentos retos entre N vértices (x_i, y_i) , com $i = 0$ até $N - 1$. O último vértice (x_N, y_N) é assumido como sendo também o primeiro, i.e., o polígono é fechado. A sua área é dada por:

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i) \quad (7.9)$$

O sinal da expressão acima pode ser utilizado para determinar a ordem dos vértices do polígono. Se o sinal for positivo os vértices do polígono estão por ordem anti-horária, caso contrário estão ordenados de forma horária.

O centroide é também conhecido como o centro de gravidade ou o centro de massa. A posição do centroide pode ser calculada através das seguintes equações:

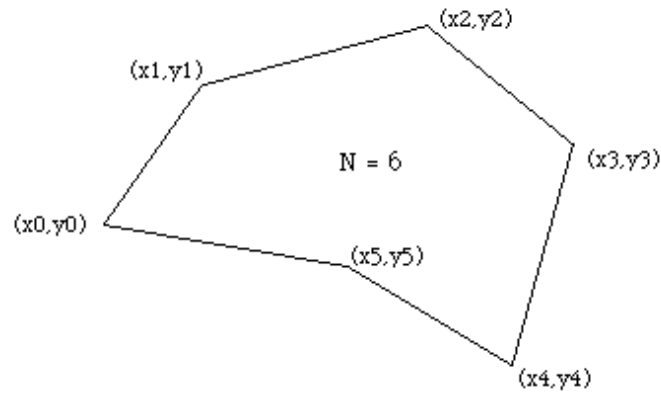


Figura 7.19: Polígono de 6 vértices

$$c_x = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (7.10)$$

$$c_y = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (7.11)$$

7.6 RoboViz

Para permitir a visualização em tempo real das informações do agente e do controlador desenvolvido foi implementada uma interface de comunicação entre o mesmo e o RoboViz. O RoboViz é uma aplicação desenhada para avaliar e desenvolver comportamentos de agentes em ambientes multiagente, mais concretamente na liga de simulação 3D da RoboCup. A aplicação trata-se de um monitor que processa a informação do estado do mundo e do agente num cenário tridimensional. Ao contrário do monitor genérico do SimSpark, o RoboViz fornece funcionalidades de depuração e desenho aos agentes através de comunicação em rede. A ferramenta facilita a visualização em tempo real dos agentes que se encontram a correr no simulador SimSpark e fornece uma análise de visualização de alto nível que não é possível com o monitor genérico do SimSpark. Este conjunto de funcionalidades pode ser observado na figura 7.21.

O RoboViz apresenta ainda uma melhoria significativa na qualidade dos gráficos apresentados em relação ao monitor, tal como se pode verificar na figura 7.20.



Figura 7.20: Comparação entre o RoboViz (à esquerda) e o monitor genérico do SimSpark (à direita)[61]

7.6.1 O protocolo de desenho

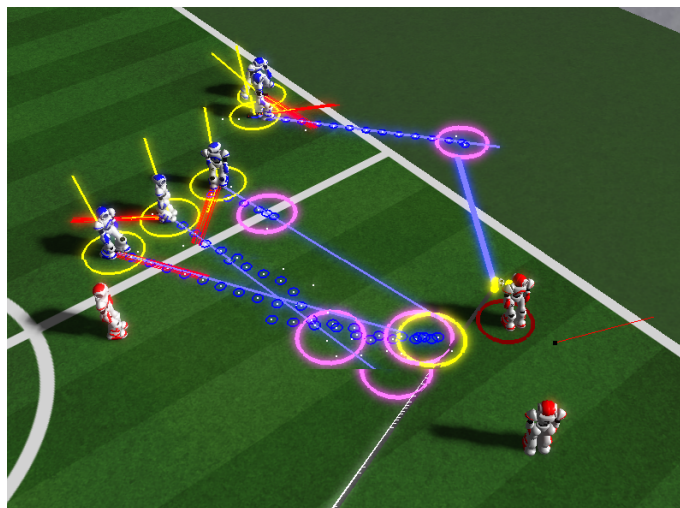


Figura 7.21: Demonstração da funcionalidade de desenho do RoboViz[61]

Os clientes interagem com o RoboViz através da emissão de comandos. Para efeitos de desenho, um exemplo de um comando pode ser desenhar uma linha ou uma esfera. Cada comando é formatado de acordo com uma forma específica adequada para ser enviada através de pacotes UDP através de uma rede.

Para que a funcionalidade de desenho remoto seja simples e flexível, tudo é decomposto num pequeno conjunto de formas. Formas mais complexas devem ser construídas utilizando combinações destas formas. A descrição de uma forma possui um conjunto de propriedades como posição, cor e escala. Cada forma também faz parte de um conjunto de formas identificado por um nome único. Desta forma é possível filtrar no RoboViz os conjuntos de formas que o mesmo deve renderizar em tempo real.

Os tipos de formas adicionados ao RoboViz podem pertencer a dois tipos: estáticos e animados. Formas estáticas são aquelas que persistem após terem sido adicionadas e não precisam ser atualizadas. Por exemplo, uma grelha no topo do campo ou eixos de coordenadas 3D não requerem que as suas formas mudem com o tempo. Formas animadas são aquelas que precisam de ter as suas propriedades atualizadas, como um vetor representando a orientação de um robô.

O sistema de coordenadas utilizado pelo RoboViz é o mesmo que o sistema de coordenadas utilizado pelo SimSpark. Para estes o terreno de jogo encontra-se no plano xy sendo que a direção z é perpendicular a este plano em direção ao céu.

7.6.2 Funções implementadas

Para permitir uma fácil representação do estado do robô foram desenvolvidas várias funções que servem de interface com o RoboViz como:

- **displaySupportPolygon** - Desenha o polígono de suporte do robô;
- **displayRelativeSupportPolygon** - Desenha o polígono de suporte do robô em coordenadas relativas à cabeça do robô;
- **displayPoint** - Desenha um ponto;
- **displayLine** - Desenha uma linha;
- **displayAxis** - Desenha o referencial de um *BodyObject* do agente como uma parte do corpo ou uma junta;
- **displayRelativeAxis** - Desenha o referencial de um *BodyObject* do agente como uma parte do corpo ou uma junta em coordenadas relativas à cabeça do corpo;
- **displayBody** - Desenha todos os elementos corporais do robô, o polígono de suporte, o seu centro de massa e a sua projeção no solo;
- **displayRelativeBody** - Desenha todos os elementos corporais do robô, o polígono de suporte, o seu centro de massa e a sua projeção no solo. Todos os elementos referidos são desenhados em coordenadas relativas à cabeça do robô.

7.7 Conclusão

Neste capítulo foi apresentado o controlador desenvolvido. Este é baseado num controlo em malha fechada do tipo PID que permite controlar a posição do centro de massa através da atuação de um determinado conjunto de juntas. De acordo com o conjunto de juntas utilizados foram definidos dois métodos de controlo para a fase de suporte único e um método de controlo para a fase de suporte duplo do robô. A alteração de métodos permite adequar as características de controlo à situação em que o robo se encontra. Por forma a garantir a estabilidade do robô foi

elaborado um método de cálculo do polígono de suporte através da utilização do algoritmo de Melkman que permite uma utilização genérica do controlador desenvolvido em diferentes tipos de configurações dos pés do robô, permitindo uma adaptação mais fácil do controlador a um robô real, e foi calculado o centroide deste polígono, para definir como ponto desejado para a posição do centro de massa. Durante o desenvolvimento do controlador foi necessário desenvolver uma interface do robô com o monitor RoboViz que permite visualizar em tempo real informações sobre o estado do robô e do mundo. O controlador desenvolvido tornou-se bastante útil quer no equilíbrio do robô quer no desenvolvimento de novos comportamentos baseados na transferência do centro de massa do robô.

Capítulo 8

Preparação de remate

O remate é a ação por excelência do futebol, e que na maioria das vezes, decide o resultado de um jogo. O remate é um gesto técnico de grande complexidade motora que necessita de estar bem ajustado às variáveis do jogo, devendo realizar-se no local e momento adequados, além de exigir um nível elevado de força explosiva. Este deve ser executado com potência e precisão e necessita de uma grande perfeição na sua execução. O rematador deve ser rápido em julgar a situação própria para o remate, quer na decisão, quer na execução, a qual deve ser tão explosiva quanto instantânea.

Num jogo de futebol entre robôs é necessário ter em conta todo o conjunto destes fatores complexos. Para o mesmo ser bem sucedido é necessário criar um conjunto de comportamentos rápidos, robustos e flexíveis, capazes de providenciar uma elevada capacidade de remate, e ter em conta todos os aspectos de alto nível que validem a possibilidade de execução e a sua utilidade em tempo real.

A potência de um remate é um dos fatores preponderantes do seu desempenho, no entanto, a sua viabilidade, robustez e possibilidade de execução são pontos essenciais para a sua utilização. Assim, torna-se necessário garantir com um elevado grau de confiança que o comportamento de remate é viável antes de o mesmo ser executado.

Os primeiros movimentos de chuto consistiam no uso de *keyframes* (quadros-chave) para a trajetória do pé. Este conceito define o movimento como uma série estática dos valores dos ângulos das articulações. Os valores são consecutivamente interpolados para executar o movimento. As principais desvantagens desta abordagem são a inflexibilidade e a adição de uma fase de preparação no movimento, na qual o robô se posiciona para executar o movimento.

Logo de seguida surgiram ideias da utilização de um chuto omnidirecional fazendo com que o chuto se torne mais flexível e eliminando assim fases de preparação rudimentares que inicialmente existiam, as quais consistiam em pequenos movimentos e rotações do jogador para que este se situe na posição e orientação pretendidas.

O novo chute desenvolvido consiste na utilização de um comportamento híbrido que alia a performance de um movimento repetitivo altamente otimizado com a precisão, flexibilidade e robustez do novo movimento de preparação de remate desenvolvido.

8.1 Análise do mundo

Por forma a possibilitar a preparação de um remate e permitir um grau elevado de dinamismo e flexibilidade do mesmo é necessário efetuar uma análise do mundo por forma a calcular a posição exata da colocação do pé para a execução do movimento único de remate. Para esta análise é necessário considerar três objetos do mundo:

- Jogador;
- Bola;
- Alvo.

Através deste conjunto de objetos é possível calcular a posição e orientação do pé em relação ao jogador, a partir de um conjunto de parâmetros definidos através da otimização da melhor posição de início de remate e que relacionam os três objetos referidos. Estes parâmetros são:

- $dist$ - Distância da posição do pé à bola;
- θ - Ângulo formado entre o alvo e a posição do pé;
- α - Orientação do pé em relação ao alvo.

Inicialmente as coordenadas do alvo e da bola são corrigidas de acordo com a orientação do robô para que as mesmas venham referidas em relação ao seu referencial. A figura 8.1 esquematiza o cálculo da posição onde o robô deve colocar o pé de suporte.

Neste cálculo é aplicada uma rotação ao vetor que vai desde o alvo até à bola, representado a azul, de θ e é normalizado o seu comprimento para $dist$. Assim, obtém-se o ponto em que o robô deve colocar o pé através da soma do vetor que vai desde o robô até à bola com o vetor que vai desde a bola ao ponto de suporte, representados a vermelho. O ângulo de orientação do pé em relação à pose inicial do robô é dado pela soma do ângulo formado entre a bola e o alvo, com o ângulo α fornecido como parâmetro.

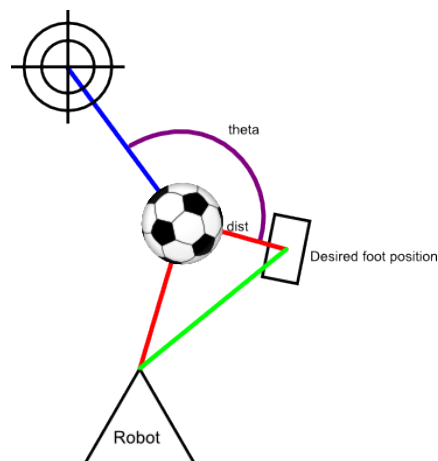


Figura 8.1: Cálculo da posição do pé de suporte do robô

8.2 Planeamento do movimento

O movimento implementado pode ser decomposto em cinco fases sendo elas, a passagem de duplo suporte para suporte único, o posicionamento do pé de suporte, o abaixamento do corpo, a transferência do centro de massa e o erguer do corpo para a pose de remate. Entre estas, a fase de posicionamento do pé de suporte e de abaixamento do corpo são executadas em paralelo. Para descrever cada uma destas fases será necessário introduzir o conceito de planeamento de trajetória.

8.2.1 Planeamento de trajetória

O módulo de planeamento de trajetória tem como função determinar um caminho entre dois pontos no espaço, usando um determinado número de pontos intermédios, de modo a determinar a trajetória que o pé deve seguir para conseguir chutar a bola[11][12].

Existem vários algoritmos capazes de produzir um caminho planar ou tridimensional, usando parâmetros para calcular o caminho mais curto e/ou evitar obstáculos, mas o seu grau de complexidade aumenta exponencialmente com o número de graus de liberdade do robô, tornando-se assim algoritmos com elevada exigência de processamento.

Neste módulo é determinada uma função que nos devolve uma curva representativa da trajetória que o pé deve seguir. Para o efeito serão utilizadas curvas de Bézier para determinar essa função.

Curvas de Bézier são assim chamadas devido ao seu inventor, Dr. Pierre Bézier. Bézier trabalhou como engenheiro na empresa de carros Renault e propôs-se, no início do anos 60, a desenvolver uma formulação de uma curva que lhe ajudasse no desenho de formas [62], tendo-a utilizado posteriormente para o design de automóveis. Esta foi desenvolvida como resultado do Algoritmo de De Casteljaun em 1957 e formalizada na década de 60.

A curva de Bézier é uma curva polinomial expressa como a interpolação linear entre alguns pontos representativos, chamados de pontos de controle. É uma curva utilizada em diversas aplicações gráficas e formatos de imagem vetorial como o SVG. Esse tipo de curva também é bastante utilizada em modelagem tridimensional e animações. Esta interpolação pode ser de grau 1 (linear), de grau 2 (quadrática), de grau 3 (cúbica) ou de grau maior que 3 [63]. Através da equação 8.1 [64] [65] [62] podemos determinar uma curva de Bézier de grau n que permite determinar uma trajetória com elevado grau de complexidade.

$$b(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad t \in [0, 1] \quad (8.1)$$

onde n é o grau da curva e P_i são os pontos de controle.

Na figura 8.2 pode ver-se exemplos de curvas de Bézier de diferentes graus.

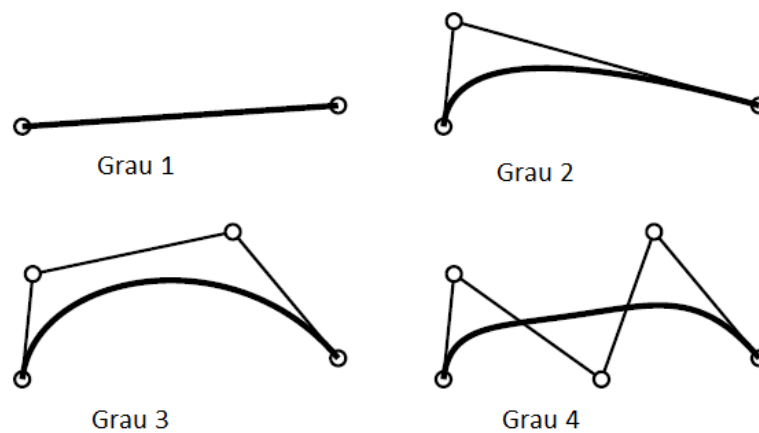


Figura 8.2: Curvas de Bézier de 1º, 2º, 3º e 4º grau [62].

Outra vantagem do uso de curvas de Bézier consiste no facto de serem acessíveis a todo o tipo de utilizadores, evitando assim toda a complexidade matemática na base do cálculo. Algumas características são [66]:

- **Intuitiva** – Todos os passos e algoritmos têm uma interpretação geométrica e intuitiva;
- **Flexível** – O sistema permite algum tipo de controlo para edição da forma da curva;
- **Abordagem unificada** – A maneira de representar, criar e editar diferentes tipos de curvas (por exemplo, retas, secções cónicas e curvas cúbicas) deve ser a mesma, isto é, não exige técnicas diferentes para manipulação de diferentes curvas;
- **Invariante** – A curva representada não sofrerá alterações na sua geometria sob transformações geométricas, tais como translações e rotações;
- **Eficiente e numericamente estável** – O cálculo da curva é rápido e preciso (eficiente). Além disso, uma grande quantidade de cálculos não distorce a forma da curva (numericamente estável).

8.2.1.1 Construção

Para a construção de uma curva de Bézier de grau n , é necessário definir $n + 1$ pontos de controlo, $P_0, P_1, P_2, \dots, P_{n-1}, P_n$. Cada um destes pontos afeta a curva como podemos ver na figura 8.3. Após determinar o grau da curva (n) e os pontos de controlo, utiliza-se a equação 8.1 para determinar $b(t)$, que corresponde à localização do ponto no instante t .

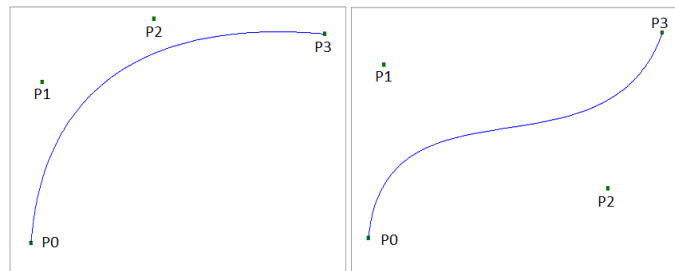


Figura 8.3: Duas curvas cúbicas de Bézier com diferentes pontos de controlo.

Relativamente ao problema, pretende-se uma curva de Bézier de grau 3, ou curva cúbica de Bézier, com $n + 1$ pontos de controlo, P_0, P_1, P_2 e P_3 , e após aplicar a equação 8.1 obtém-se uma curva de Bézier paramétrica de grau 3 (cúbica) que fornece a localização de um ponto num determinado instante. Essa curva é dada por:

$$b(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \quad t \in [0, 1] \quad (8.2)$$

8.2.2 Transferência para suporte único

O movimento de preparação de remate tem início partindo do pressuposto que o robô se encontra numa posição estável em fase de suporte duplo. Esta garantia pode ser obtida através da análise da posição de centro de massa em relação ao polígono de suporte, tal como explicitado no capítulo 7. Por forma a efetuar um passo, é necessário colocar o robô suportado apenas por um pé, sendo que para isso a primeira fase da preparação do remate consiste na transferência do centro de massa do robô para a superfície de contacto entre o pé com que o robô irá rematar e o solo, seguida do levantamento do pé contrário.

A figura 8.4 representa a transferência do robô para suporte único.



Figura 8.4: Sequência de transferência do robô para suporte único

8.2.3 Colocação do pé de suporte

Dado que o volume de trabalho da perna do robô, representado na figura 8.5 pelo semicírculo a vermelho, não contém o ponto final do pé de suporte desejado, representado a azul, o robô deverá colocar o pé no ponto cuja projeção em z interseste o semicírculo representativo do volume de trabalho. Deste modo, através do abaixamento do pé contrário, o robô colocará o pé de suporte na posição pretendida. Assim, o pé de suporte segue uma curva Bezier de grau 3 que levará o pé até esta posição numa trajetória circular.

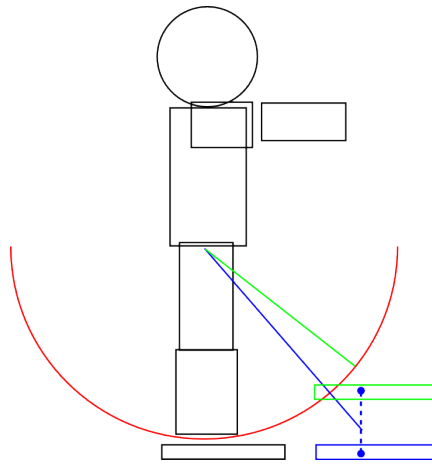


Figura 8.5: Volume de trabalho da perna do robô

8.2.4 Abaixamento do corpo do robô

Para colocar o pé na posição pretendida o robô deverá baixar o corpo de acordo com a elevação do pé definida na subsecção anterior e representada na figura 8.6. O pé deverá seguir uma trajetória linear descendente que termina quando o robô regressar à posição de duplo suporte.

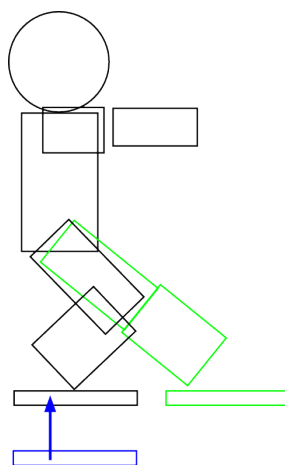


Figura 8.6: Abaixamento do corpo do robô

A figura 8.7 representa a execução em paralelo do abaixamento do corpo do robô e da colocação do pé de suporte.



Figura 8.7: Sequência de abaixamento e colocação do pé de suporte

8.2.5 Transferência do centro de massa do robô

Após o abaixamento do robô é necessário efetuar a transferência do centro de massa do robô de um pé para o outro para que seja possível que o robô consiga suportar o seu peso neste pé. Dado que a posição da bacia do robô se mantém dentro do volume de trabalho de ambas as pernas, é possível efetuar uma translação linear da mesma para o pé de suporte desejado, tal como representado na figura 8.8.

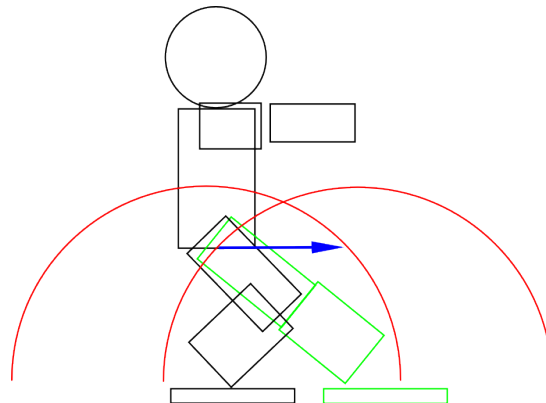


Figura 8.8: Transferência do centro de massa do robô

A figura 8.9 representa a transferência do centro de massa do robô.



Figura 8.9: Sequência de transferência do centro de massa do robô

8.2.6 Colocação em pose de remate

Após a transferência do centro de massa para o pé desejado é possível colocar o robô na pose desejada de remate, tal como ilustrado na figura 8.10.

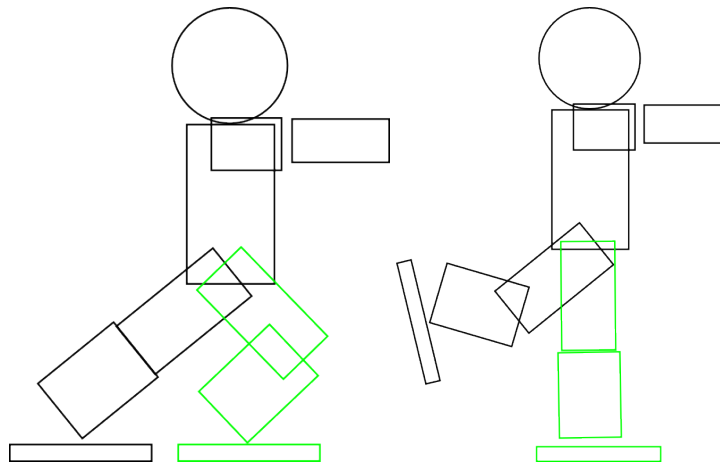


Figura 8.10: Pose final após a transferência do centro de massa do robô e pose de remate

A figura 8.11 representa a colocação em pose de remate seguida do remate.

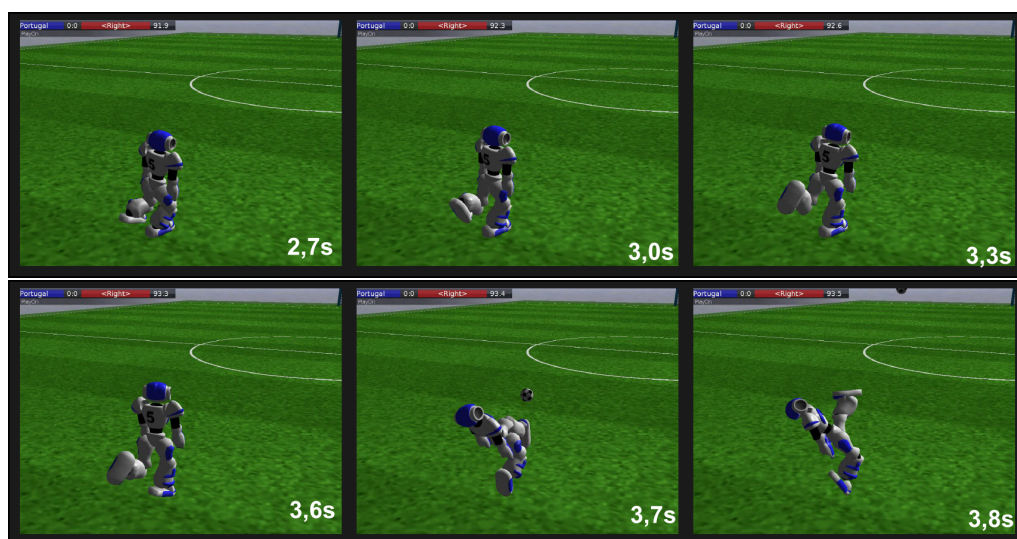


Figura 8.11: Sequência de colocação em pose de remate e consequente remate

8.3 Testes e resultados

Para verificar a área de viabilidade de remate foi efetuado um conjunto de testes ao movimento de preparação do remate. Os testes consistiram no posicionamento do robô num determinado conjunto de posições e orientações predeterminadas em relação à bola e na verificação dos resultados

do chute em termos de distância percorrida e a direção que a mesma seguiu. Apesar dos resultados obtidos serem sujeitos a alterações de acordo com os parâmetros definidos na secção 8.1, uma vez que estes podem ser alterados através de processos de otimização, a capacidade de efetuar o remate não é tão susceptível a estas alterações. A área anterior à bola foi decomposta numa matriz de resolução de 5 cm, na qual o robô foi posicionado. Esta decomposição foi efetuada para uma variação de x entre (0,25 e 0) e uma variação em y entre (-0,25 e 0,25) em metros. Em cada ponto da matriz foram testados os ângulos de -50° a 50° com intervalos de 10° e para efeitos de confirmação de resultados foram efetuadas duas amostras por teste. No total foram efetuados mais de 1300 testes que confirmaram as condições que garantem a capacidade do robô executar o movimento de remate com o pé direito. Os resultados do teste foram guardados num ficheiro csv, por simplicidade de criação e de acesso aos seus valores.

Através da ferramenta MatLab procedeu-se à análise dos resultados obtidos. Para isso foram lidos os resultados e foi efetuada uma transformação dos mesmos numa imagem em escala de cinza que nos permite obter uma função que relaciona a distância percorrida pela bola em relação à pose inicial do robô. A figura 8.12 representa o resultado obtido após a aplicação da referida transformação ao conjunto de posições em que o robô se encontra alinhado com o alvo, sendo que no gráfico o branco representa uma distância nula e o preto representa uma distância percorrida pela bola de 18 metros.

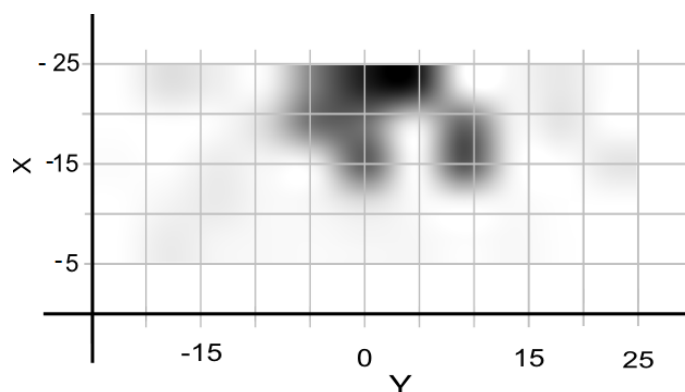


Figura 8.12: Representação da distância do remate em relação à posição do robô orientado com o alvo

Como se pode observar na figura 8.12, o desempenho do remate do robô é afetado pela posição de onde parte o movimento. Definindo-se como 7 metros a distância mínima para considerar que o robô é capaz de efetuar um remate, através da aplicação de uma binarização da imagem referida com o *threshold* adequado, obteve-se assim a área de capacidade de remate dada pelo movimento para o pé direito. Para a obtenção da área de capacidade de remate com ambos os pés fez-se a união entre a imagem representativa dos resultados para uma dada orientação, com a imagem espelhada representativa dos resultados para a orientação oposta.

Deste resultado obteve-se o seguinte conjunto de zonas admissíveis de remate de acordo com o ângulo entre o alvo e a orientação do jogador:

Orientação 0°

A figura 8.13 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 0° entre a sua orientação e o alvo. Quando o robô se encontra orientado com o alvo, a área admissível é vasta e possui um conjunto de resultados muito bons, com um máximo de 16,16 m registado na posição (-25, 5)cm. As duas pequenas áreas de inadmissibilidade de remate registadas na parte inferior do gráfico devem-se à sobreposição dos pés para o remate o que o torna impossível de ser concretizado. Esta orientação é extremamente favorável ao remate e possui uma vasta área em que o mesmo é passível de ser executado.

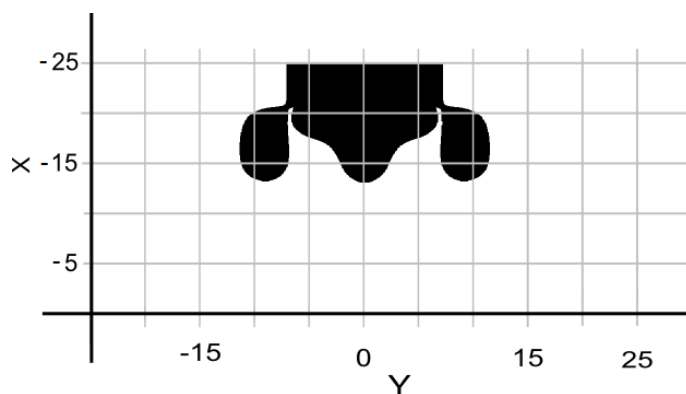


Figura 8.13: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 0° entre a sua orientação e o alvo

Orientação 10°

A figura 8.14 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 10° entre a sua orientação e o alvo. Quando o robô se encontra com uma orientação de 10° em relação ao alvo, a área admissível continua extensa em tudo semelhante à anterior. Esta orientação é bastante favorável ao remate e possui uma vasta área em que o mesmo é passível de ser executado. Para esta orientação, a preparação de remate possui um conjunto de resultados muito bons com um máximo de 16,69 m registado na posição (-25, 0) cm.

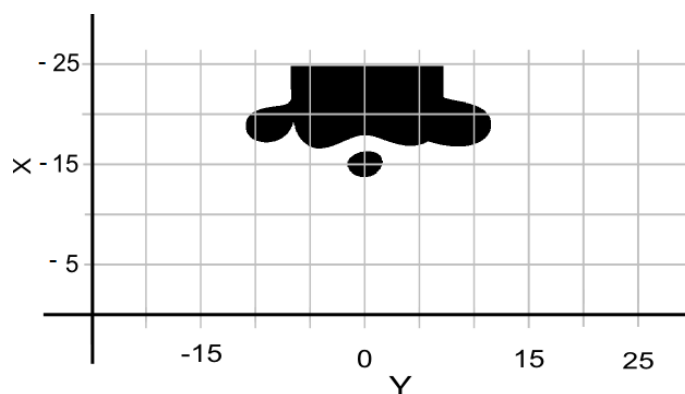


Figura 8.14: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 10° entre a sua orientação e o alvo

Orientação 20°

A figura 8.15 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 20° entre a sua orientação e o alvo. Quando o robô se encontra com uma orientação de 20° em relação ao alvo, a área admissível é a maior do conjunto de testes efetuados com mais de 80cm^2 . Esta orientação é a mais favorável para a execução do remate e possui um conjunto de resultados excelentes com um máximo de 17,03 m registado na posição (-20, 5) cm.

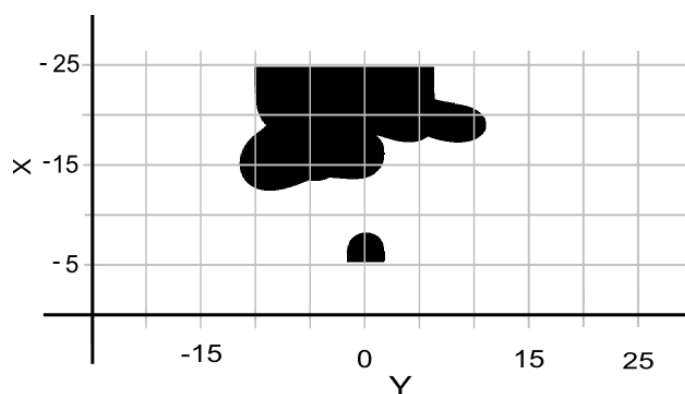


Figura 8.15: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 20° entre a sua orientação e o alvo

Orientação 30°

A figura 8.16 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 30° entre a sua orientação e o alvo. Quando o robô se encontra com uma orientação de 30° em relação ao alvo a sua área admissível é um pouco pior face aos

resultados anteriores. Apesar de tudo esta orientação possui um conjunto de resultados médios com um máximo de 12,96 m registado na posição (-20, 0) cm.

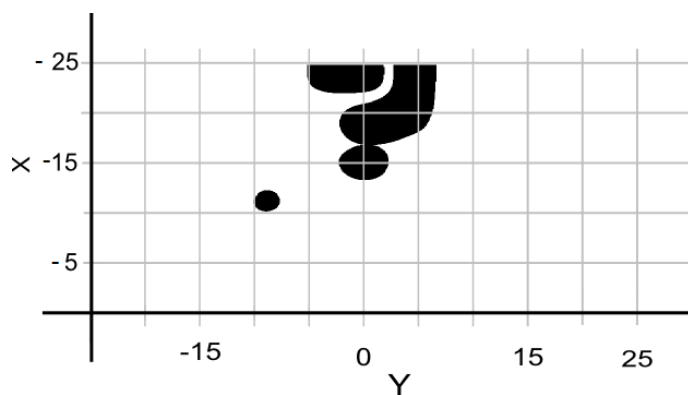


Figura 8.16: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 30° entre a sua orientação e o alvo

Orientação 40°

A figura 8.17 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 40° entre a sua orientação e o alvo. Quando o robô se encontra com uma orientação de 40° em relação ao alvo a sua área admissível é um pouco pior em comparação com os resultados anteriores sendo bastante reduzida. Apesar de tudo esta orientação possui um conjunto de resultados médios com um máximo de 12,74 m registado na posição (-20, 0) cm.

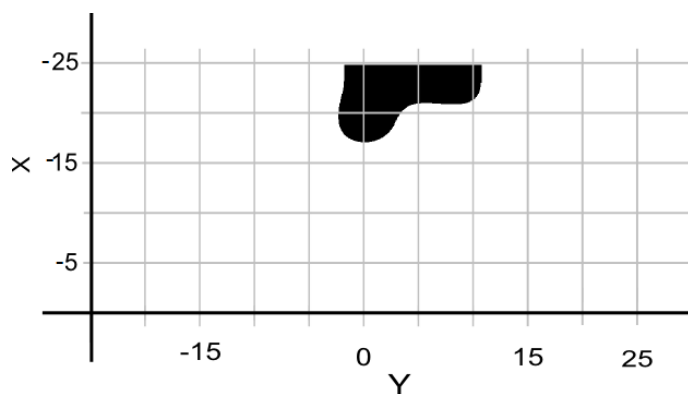


Figura 8.17: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 40° entre a sua orientação e o alvo

Orientação 50°

A figura 8.18 representa a área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 50° entre a sua orientação e o alvo. Quando o robô se encontra

com uma orientação de 50° em relação ao alvo a sua área admissível de preparação de remate é extremamente reduzida. Ainda assim, dadas as condições extremamente desfavoráveis em termos de orientação do robô, esta configuração possui um conjunto de resultados aceitáveis com um máximo de 13,06 m registado na posição (-25, 10) cm.

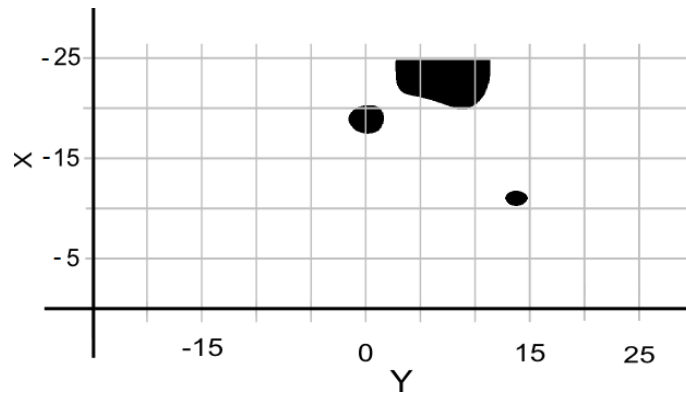


Figura 8.18: Representação da área admissível de remate em relação à posição do robô quando este se encontra com um ângulo de 50° entre a sua orientação e o alvo

8.4 Conclusão

Neste capítulo foi apresentado o desenvolvimento do comportamento de preparação de remate. Este comportamento permite um grau elevado de dinamismo e flexibilidade de acordo com a posição e orientação do robô em relação ao alvo, dotando o movimento de remate único de uma preparação omnidirecional. Os resultados obtidos através do conjunto de testes efetuados foram bastante satisfatórios e estão sujeitos a melhorias através da otimização dos parâmetros que definem o comportamento.

Capítulo 9

Conclusões e trabalho futuro

Este capítulo sumariza as principais conclusões obtidas, e apresenta um conjunto de melhorias do sistema projetado. Por fim, é feito um levantamento de trabalhos futuros passíveis de ser realizados no âmbito de dissertações ou outros projetos relacionados com o trabalho desenvolvido.

9.1 Conclusões

Neste trabalho foi desenvolvido um módulo de estabilidade para permitir a avaliação do estado de equilíbrio de um robô humanoide NAO e ainda o controlo da posição do centro de massa do mesmo. Este comportamento permite uma melhoria significativa no desempenho do robô, nomeadamente na avaliação do estado de equilíbrio do robô nas fases de transição entre comportamentos e como ferramenta de avaliação e otimização de comportamentos.

Dada a modularidade do controlador desenvolvido, este está apto a ser facilmente incorporado na arquitetura de comportamentos da equipa FC Portugal.

O controlador desenvolvido permite a adequação do tipo de controlo a utilizar, através da atuação das juntas da bacia ou através da atuação destas juntas juntamente com as juntas dos pés, por forma a permitir escolher um controlador com respostas mais velozes ou controladores que garantam uma inalteração da orientação do tronco do robô. Este controlador foi devidamente testado e foi comprovado que a utilização de um controlador do tipo PID para o controlo da posição do centro de massa permite uma resposta rápida e precisa.

Devido à dificuldade de análise em tempo real do estado do agente fornecido pelas ferramentas utilizadas pela equipa revelou-se necessário a utilização do monitor RoboViz e da criação de uma interface entre este e o agente da equipa. Esta interface revelou ser uma ajuda preciosa no desenvolvimento do controlador através do desenho das características do robô e dos cálculos efetuados pelo mesmo, tais como o polígono de suporte e o centro de massa.

Após o desenvolvimento do controlador referido decidiu-se utilizar o trabalho efetuado para o desenvolvimento de um comportamento de preparação de remate. O movimento de preparação de

remate desenvolvido consiste num movimento em que o robô se encontra sempre numa posição estaticamente equilibrada através do qual é possível posicionar o pé do robô numa posição e orientação extremamente precisas, permitindo dotar um movimento sequencial predefinido através de *keyframes* num remate com uma preparação que lhe garante um grau de omnidirecionalidade.

Para a análise de resultados e a verificação de condição de remate foi efetuado um conjunto de testes intensivo que permitiu obter um grau de confiança elevado para os resultados do mesmo.

9.2 Trabalhos futuros

A estabilidade de robôs humanoides tem sido um campo de investigação importante na área da robótica e assim continuará a ser no futuro dada a sua importância e complexidade inerentes. Apesar da obtenção de um conjunto de resultados interessantes e satisfatórios, existem um conjunto de melhorias e experiências que podem ser efetuadas para a complementação do mesmo, tais como:

- Utilização de outros métodos de análise de estabilidade como CoP e ZMP;
- Utilização e análise de diferentes arquiteturas de controladores como controladores difusos ou redes neuronais;
- Otimização dos parâmetros do comportamento de preparação de remate para diversos conjuntos de remates. Desta forma seria possível selecionar um tipo de remate como remate rasteiro ou remate em altura com base num conjunto de parâmetros que permitam o desempenho e robustez dos mesmos;
- Aumento do número de testes do controlador do centro de massa e dos movimento de preparação do remate para garantir de forma mais precisa as regiões de utilização dos mesmos;
- Desenvolvimento de um método de locomoção omnidirecional baseado no comportamento de locomoção definido. Dado que o movimento consiste na realização de um passo seria interessante criar um método de locomoção baseado neste, garantindo um método de locomoção extremamente robusto dado que o mesmo seria estaticamente equilibrado;
- Incorporar o comportamento num movimento de passo, isto é, executar o movimento em andamento sem que o andamento seja interrompido. Após o desenvolvimento do método de locomoção, bastaria colocar o pé no ponto predeterminado na última passada, garantir o equilíbrio do robô através do controlador de centro de massa desenvolvido e por fim executar o movimento de remate.

Bibliografia

- [1] RoboCup, “RoboCup Mexico 2012.” Disponível em <http://www.robocup2012.org>, acessado a última vez em Fevereiro de 2012.
- [2] H. Kitano, “RoboCup Humanoid Challenge : That ’ s One Small Step for A Robot , One Giant Leap for Mankind,” *Power*, no. October, 1998.
- [3] RoboCup, “RoboCup 2011 - Istanbul.” Disponível em <http://www.robocup2011.org/en/>, acessado a última vez em Maio de 2012.
- [4] E. Sklar, A. Eguchi, and J. Johnson, “RoboCupJunior : Learning with Educational Robotics,” *History*, pp. 238–253.
- [5] RoboCup, “RoboCup Rescue.” Disponível em <http://www.robocuprescue.org>, acessado a última vez em Junho de 2012.
- [6] S. Tadokoro, H. Kitano, T. Takahashi, I. Noda, H. Matsubara, A. Shin, J. O. H. Tetsuo, I. Takeuchi, H. Takahashi, F. Matsuno, M. Hatayamm, J. Nobe, and S. Shimada, “The RoboCup-Rescue Project : A Robotic Approach to the Disaster Mitigation Problem,” *Simulation*, 2000.
- [7] RoboCup, “RoboCup 2011.” Disponível em <http://www.robocup2011.org>, acessado a última vez em Fevereiro de 2012.
- [8] RoboCup, “Middle Size Robocup.” Disponível em <http://www.robocup.org/robocup-soccer/middle-size/>, acessado a última vez em Fevereiro de 2012.
- [9] R. T. Committee, *RoboCup Standard Platform League (Nao) Rule Book*. 2012.
- [10] RoboCup, “Standard Platform League History.” Disponível em <http://www.tzi.de/spl/bin/view/Website/History>, acessado a última vez em Maio de 2012.
- [11] R. P. G. Ferreira, “Desenvolvimento de um chute omnidirecional para um robô humanoide,” *Tese Mestrado, Faculdade de Engenharia da Universidade do Porto*, 2012.
- [12] R. Ferreira, L. P. Reis, and A. P. Moreira, “Development of an omnidirectional kick for a humanoid robot,” *CISTI 2012, Atas da Conferencia Ibérica de Sistemas e Tecnologias de Informação*, pp.235-240 Madrid, Espanha, 2012.
- [13] L. P. Reis and N. Lau, “Fc portugal team description: Robocup 2000 simulation league champion,” in *RoboCup 2000: Robot Soccer World Cup IV* (P. Stone, T. Balch, and G. Kraetzschmar, eds.), vol. 2019 of Lecture Notes in Computer Science, pp. 29–40, Universidade do Minho, 2001.
- [14] L. P. Reis, N. Lau, and L. Mota, “Fc portugal 2d simulation: Team description paper,” in *RoboCup 2010: Robot Soccer World Cup XIV*, 2010.

- [15] L. P. Reis, N. Lau, A. J. Neves, and A. P. Moreira, “Fc portugal spl: A new portuguese mixed team for the standard platform league with scientific contributions of fc portugal, cambada and 5dpo teams,” pp. 1 – 3, 2009.
- [16] C. Team, “Cambada robotic soccer.” Disponível em <http://www.ieeta.pt/atri/cambada/index.htm>, acessado pela última vez em Janeiro de 2012.
- [17] A. J. R. Neves, J. L. Azevedo, B. Cunha, J. Cunha, R. Dias, P. Fonseca, N. Lau, J. Silva, T. Angelo, C. Cruz, G. Corrente, L. S. Lopes, P. Pedreiras, A. Pereira, A. Pinho, J. Rodrigues, F. Santos, and J. Vieira, “Cambada’2011: Team description paper,” 2011.
- [18] N. Lau, L. S. Lopes, G. Corrente, and N. Filipe, “Multi-robot team coordination through roles, positionings and coordinated procedures,” in *2009 IEEE/RSJ international conference on Intelligent robots and systems, IROS ’09*, pp. 5841 – 5848, IEEE Press, 2009.
- [19] N. Lau, L. S. Lopes, N. Filipe, and G. Corrente, “Roles, positionings and set plays to coordinate a robocup msl team,” in *14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence, EPIA ’09*, pp. 323 – 337, 2009.
- [20] D. Team, “5dpo.” Disponível em <http://paginas.fe.up.pt/~robosoc/en/doku.php>, acessado pela última vez em Janeiro de 2012.
- [21] T. P. Nascimento, M. A. Pinto, H. M. Sobreira, F. Guedes, A. Castro, P. Malheiros, A. Pinto, H. P. Alves, M. Ferreira, P. Costa, P. G. Costa, A. Souza, L. Almeida, L. P. Reis, and A. P. Moreira, “5dpo’2011: Team description paper,” 2011.
- [22] A. Neves, N. Lau, L. Reis, A. Moreira, A. Trifan, B. Pimentel, C. Sobrinho, and E. Domingues, “Spl portuguese team: Team description paper for robocup 2011,” 2011.
- [23] Takanishi Laboratory, “Biped Humanoid Robot Wabian-2R.” Disponível em <http://www.takanishi.mech.waseda.ac.jp/top/research/wabian/index.htm>, acessado a última vez em Fevereiro de 2012.
- [24] F. Junod, “Jaques Vaucanson.” Disponível em http://www.francoisjunod.com/automates/eightennth/vaucanson_uk.htm, acessado a última vez em Junho de 2012.
- [25] P. Guinan, “The Steam Man.” Disponível em <http://www.bigredhair.com/steamman/>, acessado a última vez em Junho de 2012.
- [26] P. Guinan, “The Electric Man.” Disponível em <http://www.bigredhair.com/electricman/>, acessado a última vez em Junho de 2012.
- [27] W. News, “1939 New York World’s Fair.” Disponível em http://wn.com/Category:1939_New_York_World's_Fair, acessado a última vez em Junho de 2012.
- [28] J. Isom, “A Brief History of Robotics.” Disponível em <http://robotics.megagiant.com/history.html>, acessado a última vez em Fevereiro de 2012.
- [29] V. David Gouaillier, “Mechatronic design of nao humanoid,” *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference*, 2009.
- [30] Boston Dynamics, “PetMan.” Disponível em http://www.bostondynamics.com/robot_petman.html, acessado a última vez em Fevereiro de 2012.

- [31] Honda Motor, “Asimo History.” Disponível em <http://world.honda.com/ASIMO/history/>, acessado a última vez em Fevereiro de 2012.
- [32] Kawada Industries, “HRP-4.” Disponível em <http://www.kawada.co.jp/mechs/hrp-4/index.html>, acessado a última vez em Fevereiro de 2012.
- [33] T. B. Globe, “The Big Picture - Robots.” Disponível em <http://www.boston.com/bigpicture/2009/03/robots.html>, acessado a última vez em Junho de 2012.
- [34] Aldebaran Robotics, “Aldebaran Robotics Gallery.” Disponível em <http://www.aldebaran-robotics.com/en/Discover-NAO/images-gallery.html>, acessado a última vez em Fevereiro de 2012.
- [35] Aldebaran Robotics, “Nao’s Specifications.” Disponível em <http://developer.aldebaran-robotics.com/nao/>, acessado a última vez em Fevereiro de 2012.
- [36] N. Shafi and L. Paulo Reis, “Two Humanoid Simulators,” 2011.
- [37] H. Lund and O. Miglino, “From simulated to real robots,” 1996.
- [38] M. Y. F. M. e. H. Joschka Boedecker, Klaus Dorer, “SimSpark User’s Manual,” 2010.
- [39] H. Picado, “Desenvolvimento de Comportamentos para um Robô Humanoide Simulado D,” 2008.
- [40] Simspark, “Models.” Disponível em <http://simspark.sourceforge.net/wiki/index.php/Models>, acessado a última vez em Fevereiro de 2012.
- [41] A. Robotics, “Nao Datasheet,” 2008.
- [42] J. Boedecker, K. Dorer, M. Rollmann, Y. Xu, F. Xue, M. Buchta, and H. Vatankhah, *SimSpark - Soccer Simulation*. Disponível em http://simspark.sourceforge.net/wiki/index.php/Soccer_Simulation, acessado a última vez em Junho de 2012.
- [43] R. Community, “Robocup soccer server 3d users manual,” January 2010.
- [44] H. Marques, N. Lau, and L. P. Reis, “Fc portugal 3d simulation team: Architecture, low-level skills and team behaviour optimized for the new robocup 3d simulator,” *Scientific Meeting of the Portuguese Robotics Open 2004*, vol. 14, pp. 31 –37, April 2004.
- [45] H. Picado, “Fcportugal3d - a team of robocup 3d simulation league,” p. 9, January 2009.
- [46] H. W. e. a. Gray, *Gray’s Anatomy The Anatomical Basis of Clinical Practice*. Churchill Livingstone Elsevier, 2008.
- [47] R. Arkin, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.
- [48] J. Helbo, “Modelling and Control of a Biped Robot,” 2007.
- [49] S. Cotton, “Modélisation, dynamique et estimation du centre de masse de robots humanoïdes,” 2010.
- [50] T. F. Yik, “Locomotion of bipedal humanoid robots: Planning and learning to walk,” *Tese PhD*, 2007.

- [51] MathWorks, “MathWorks.” Disponível em <http://www.mathworks.com>, acessado a última vez em Maio de 2012.
- [52] M. W. Spong and H. Carter, *Robot Modeling and Control*. John Wiley and Sons, Inc., 2006.
- [53] O. R. Project, “Open Robot Project.” Disponível em <http://orobot.net/wiki/index.php>, acessado a última vez em Maio de 2012.
- [54] E. Mota, “Movimentação e desenvolvimento de um Robô NAO,” *Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto*, 2011.
- [55] H. Ozbay, *Introduction to Feedback Control Theory*. CRC Press, 1999.
- [56] J. Silva, *Sensor fusion and behaviours for the CAMBADA robotic soccer team*. Universidade de Aveiro, 2008.
- [57] V. Santos and F. Silva, “Journal of vibration and control,” in *Design and low-level control of a humanoid robot using a distributed architecture approach*, ch. 12, p. 1431–1456, 2006.
- [58] K. H. Ang, G. Chong, and Y. Li, “Ieee transactions on control systems technology,” in *PID control system analysis, design and technology*, ch. 13, p. 559–576, 2005.
- [59] A. Isidori, *Nonlinear Control Systems (Communications and Control Engineering)*. Springer, 1995.
- [60] J. Ziegler and N. Nichols, “Transactions of asme,” in *Optimum settings for automatic controllers*, ch. 64, p. 759–768, 1942.
- [61] J. Stoecker, “The roboviz project.” Disponível em <https://sites.google.com/site/umroboviz>, acessado a última vez em Maio de 2012.
- [62] T. Sederberg, “Byu bézier curves. chapter 2..” Disponível em http://www.tsplines.com/resources/class_notes/Bezier_curves.pdf, acessado pela última vez em Janeiro de 2012.
- [63] J. C. Bartels, R. H.; Beatty and B. A. Barsky, “Bézier curves,” *Ch. 10 in An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, pp. 211 – 245, 1998.
- [64] J. Muller, T. Laue, and T. Rofer, “Kicking a ball- modeling complex dynamic motions for humanoid robots,” *RoboCup 2010: Robot Soccer World Cup XIV, volume 6556 of Lecture Notes in Artificial Intelligence*, pp. 119 – 120, Springer 2011.
- [65] E. W. Weisstein, ““bézier curve.” from mathworld—a wolfram web resource..” Disponível em <http://mathworld.wolfram.com/BezierCurve.html>, acessado pela última vez em Janeiro de 2012.
- [66] C. Shene, “Introduction to computing with geometry notes. unit 5: Bézier curves..” Disponível em <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/>, acessado pela última vez em Janeiro de 2012.